

BASIC 553

rel. 3.1 e successive

INTRODUZIONE

Il **BASIC 52** é un potente tool software sviluppato dalla **grifo**®, che consente la programmazione ad alto livello (BASIC), su tutte le schede basate sulla famiglia 51 Intel. Tale BASIC viene eseguito da EPROM e genera un codice "romabile" che viene eseguito dall' eeprom parallela di bordo; si riduce così la necessità di hardware esterno (in circuit emulator, EPROM programmer, etc.) e allo stesso tempo si velocizza la fase di debug del programma applicativo. **BASIC 52** é il riferimento a pacchetti software generici, ma per ciascuna scheda esiste una relativa versione di software implementata per gestire le differenti periferiche di bordo; perciò per ogni scheda il nome **BASIC 52** diventa BASIC e di seguito la parte finale del nome della scheda. Questa documentazione deve essere utilizzata come "addendum" al manuale d'uso del **BASIC 52** e perciò include soltanto le differenze rispetto alla documentazione originale (addizioni e eliminazioni di comandi e istruzioni).

BASIC 52 PER GPC® 553 = BASIC 553

Di seguito vengono elencate le differenze tra il **BASIC 52** originale e il **BASIC 553**:

Comandi eliminati

LIST#
FPROG
FPROG1
FPROG2
FPROG3
FPROG4
FPROG5
FPROG6

Istruzioni eliminate

BAUD
PRINT#
PH0.#
PH1.#
PWM

Operatori eliminati

Nessuno

Comandi aggiunti

ERASE

->

Cancella tutto il contenuto della EEPROM ossia elimina i programmi salvati tramite i comandi PROG, PROG1...PROG6.

Funzione

Istruzioni aggiunte

GES_RTC
ALARM
A_D
SET_PWM
BY_EE
BL_EE

->

Gestisce la lettura o la inizializzazione dell'RTC.

->

Gestisce le varie modalità di allarme dell'RTC.

->

Gestisce l' acquisizione dell' A/D di bordo.

->

Gestisce le linee di PWM della CPU.

->

Gestisce la scrittura o la lettura di un byte sulla EEPROM seriale e sul RAM-RTC seriale di bordo.

->

Gestisce la scrittura o la lettura di un blocco sulla EEPROM seriale e sul RAM-RTC seriale di bordo.

Istruzioni aggiunte**Funzione**

RW_SFR	->	Gestisce la lettura/scrittura in un SFR.
COM2	->	Gestisce l'INPUT sulla linea seriale software.
PRINT@	->	Gestisce l'OUTPUT sulla linea seriale software.
DISPLAY	->	Gestisce l'inizializzazione del display selezionato.
KEYB	->	Gestisce tutte le operazioni sulla tastiera operatore.
Ridirezione console output	->	Gestisce l'output sul display selezionato con istruzioni BASIC ad alto livello.

Operatori aggiunti**Funzione**

Nessuno	->	-
---------	----	---

Autorun - Debug mode.

Per lanciare un programma in AUTORUN (partenza automatica dopo il power on o un reset), é necessario muovere il DIP 8 del dip-switch DSW1 di bordo nella posizione **ON**. Se é presente un programma di AUTORUN ma si intende eseguire lo stesso il **BASIC 553**, allora bisogna muovere il DIP 8 nella posizione **OFF**. Riassumendo:

DSW1.8 in OFF:	Modalità DEBUG.
DSW1.8 in ON:	Modalità RUN.

La modalità di DEBUG é molto interessante quando per esempio, bisogna modificare un sistema già operativo. L'utente non deve fare altro che connettere la scheda ad un P.C., selezionare la modalità di DEBUG, tramite il **BASIC 553** modificare l'applicativo e successivamente, dopo gli opportuni salvataggi, ripristinare la modalità di RUN. Come si può facilmente notare non bisogna apportare nessuna modifica hardware alla scheda perciò é possibile intervenire direttamente sul campo. Per programmare in AUTO-RUN un applicativo seguire i seguenti passi.

- Entrare in modalità DEBUG.
- digitare: ERASE <invio>.
- Caricare il programma in memoria.
- digitare: PROG <invio>
- digitare: PROG4 <invio>

Descrizione e sintassi delle procedure aggiunte.

Per poter utilizzare ad alto livello tutte le periferiche della scheda, si è ampliata la gamma di istruzioni del BASIC creando procedure specifiche. Per l'implementazione di queste nuove istruzioni si sono utilizzate alcune locazioni in RAM ESTERNA con cui si effettua l'interscambio delle informazioni necessarie. Questa area di interscambio é allocata nel range **07E00H÷07FFFH** (RAM = 32 Kbytes). Prima di scrivere un programma che utilizza tali nuovi token é necessario digitare il comando **MTOP=07DFFH**, questo per evitare l'utilizzo da parte del BASIC dell'area di memoria riservata. Si ricorda che le locazioni di ram interna da 018H a 021H non sono più disponibili per l'utente, in quanto esse sono state utilizzate per l'implementazione dei nuovi token.

Di seguito vengono elencate e descritte tutte le nuove istruzioni.

PROCEDURA DI GESTIONE A/D

Sintassi: A_D <expr>

Descrizione istruzione:

Restituisce il valore convertito (10 Bits --> 0÷1023) sul canale specificato dell'A/D di bordo.

Descrizione parametri:

<expr> --> Numero del canale (0÷7).

Descrizione output (valore da estrarre con l'istruzione POP ...):

OUTPUT: 0÷1023 --> Valore di conversione.
 65535 --> Parametri non corretti.

Esempio:

```
0  REM ***** Esempio di conversione su un canale dell' A/D.
10 A_D 3 : REM Conversione sul canale 3.
20 POP A : REM Lettura del valore di conversione e del risultato dell' operazione.
30 IF A<>65535 THEN PRINT A : REM Stampa del valore convertito.
```

PROCEDURA DI GESTIONE DELLE LINEE DI PWM

Sintassi: SET_PWM <expr1>, <expr2>, <expr3>

Descrizione istruzione:

Setta le 2 linee di PWM presenti a bordo della CPU. Si ricorda che per il settaggio della frequenza e del duty-cycle si sono utilizzate le seguenti formule:

$$fpwm = 22118400 / (2 \times (1 + PWMP) \times 255).$$

$$\text{Low/high ratio of /PWMn} = \text{PWMn} / 255 - \text{PWMn}$$

Descrizione parametri:

<expr1> --> Selezione della linea di PWM (0÷1).
<expr2> --> Frequenza (169÷43369 Hz con quarzo da 22118400 Hz).
<expr3> --> Duty-Cycle (0÷100%).

Descrizione output (valore da estrarre con l' istruzione POP ...):

OUTPUT: Nessuno.

Esempi:

```
0  REM ***** Esempio di settaggio della linea 0 di PWM.
10  SET_PWM 0, 180, 50 : REM Settaggio PWM0; 180 Hz; 50%.

0  REM ***** Esempio di settaggio della linea 1 di PWM.
10  SET_PWM 1, 180, 50 : REM Settaggio PWM1; 180 Hz; 50%.

0  REM ***** Esempio di settaggio della linea 0 di PWM al valore logico "0".
10  SET_PWM 0, 0, 0 : REM Settaggio linea PWM0 a "0";

0  REM ***** Esempio di settaggio della linea 0 di PWM al valore logico "1".
10  SET_PWM 0, 1, 1 : REM Settaggio linea PWM0 a "1";

0  REM ***** Esempio di settaggio della linea 1 di PWM al valore logico "0".
10  SET_PWM 1, 0, 0 : REM Settaggio linea PWM1 a "0";

0  REM ***** Esempio di settaggio della linea 1 di PWM al valore logico "1".
10  SET_PWM 1, 1, 1 : REM Settaggio linea PWM1 a "1";
```

PROCEDURA DI GESTIONE DELL'OROLOGIO

Sintassi: GES_RTC <expr1>, <expr2>, <expr3>, <expr4>, <expr5>, <expr6>, <expr7>, <expr8>

Descrizione istruzione:

Inizializza l' orologio o restituisce la DATA o l'ORA del medesimo. Si ricorda che anche in lettura bisogna indicare tutti i parametri anche se non significativi (<expr2>, ..., <expr8>).

Descrizione parametri:

<expr1> --> 0 = Lettura ORE, MINUTI, SECONDI.
 --> 1 = Lettura GIORNO DELLA SETTIMANA, GIORNO, MESE, ANNO.
 --> 2 = Inizializzazione dell'orologio.
 <expr2> --> Valore che esprime le ore (0÷23).
 <expr3> --> Valore che esprime i minuti (0÷59).
 <expr4> --> Valore che esprime i secondi (0÷59).
 <expr5> --> Valore che esprime il giorno della settimana (0÷6).
 <expr6> --> Valore che esprime il giorno del mese (1÷31).
 <expr7> --> Valore che esprime il mese (1÷12).
 <expr8> --> Valore che esprime l'anno (0÷3).

Descrizione output (valore da estrarre con l' istruzione POP ...):

OUTPUT INIZIALIZZAZIONE: 0 --> Inizializzazione OK.
 <>0 --> Dispositivo non funzionante.

OUTPUT LETTURA: 0÷255 --> Lettura OK.
 >255 --> Dispositivo non funzionante.

Esempio:

```

0   REM ***** Esempio di SETTAGGIO e di LETTURA dell' orologio di bordo.
10  GES_RTC 2,23,59,30,6,31,12,2 : REM Inizializzazione orologio.
20  POP A : REM Lettura del risultato dell' operazione.
30  IF A<>0 THEN PRINT "ERRORE nella INIZIALIZZAZIONE" : END
40  GES_RTC 1,0,0,0,0,0,0,0 : REM Lettura della DATA.
50  POP SET : POP GIO : POP MES : POP ANN
60  IF (SET>255.OR.GIO>255.OR.MES>255.OR.ANN>255) THEN 120
70  ? SET, : ? CHR(45), : ? GIO, : ? CHR(45), : ? MES, : ? CHR(45), : ? ANN+1900,
80  GES_RTC 0,0,0,0,0,0,0,0 : REM Lettura dell' ORA.
90  POP A : POP B : POP C
100 IF (A>255.OR.B>255.OR.C>255) THEN 120
110 ? " " : ? A, : ? CHR(58), : ? B, : ? CHR(58), : ? C, : ? CHR(13), : GOTO 40
120 PRINT "ERRORE nella LETTURA dell' orologio." : END
  
```

LETTURA/SCRITTURA DI UN BYTE SULLA EEPROM SERIALE E SULLA RAM-RTC SERIALE

Sintassi: BY_EE <expr1>, <expr2>, <expr3>

Descrizione istruzione:

Effettua la scrittura o la lettura di un byte all' indirizzo specificato, sulla EEPROM seriale (IC16) o sul PCF-8583 (IC17).

Si ricorda che anche in lettura bisogna indicare tutti i parametri anche se non significativi (<expr3>).

Descrizione parametri:

<expr1> --> 0 = Lettura di un byte.
--> 1 = Scrittura di un byte.
<expr2> --> Indirizzo della locazione da scrivere o leggere (0÷<ultimo indirizzo dispositivo>).
<expr3> --> Valore da scrivere (0÷255).

I dispositivi (EEPROM e RAM-RTC) sono mappati secondo la seguente tabella:

0000H÷00FFH	->	selezione RAM-RTC	(IC17)
0430H÷04FFH	->	selezione EEPROM 24c02	(IC16)
0430H÷05FFH	->	selezione EEPROM 24c04	(IC16)
0430H÷07FFH	->	selezione EEPROM 24c08	(IC16)

Descrizione output (valore da estrarre con l'istruzione POP ...):

OUTPUT SCRITTURA:	0	-->	Scrittura OK.
	1	-->	Parametri non corretti.
	2	-->	Dispositivo non funzionante.

OUTPUT LETTURA:	0÷255	-->	Letture OK.
	256	-->	Parametri non corretti.
	257	-->	Dispositivo non funzionante.

Esempi:

```
0   REM ***** Esempio di SCRITTURA del byte 85 all' indirizzo 100 della RAM-RTC.
10  BY_EE 1,100,85 : REM Si effettua la scrittura di un byte.
20  POP A : REM Lettura del risultato dell' operazione.

0   REM ***** Esempio di LETTURA all' indirizzo 100 della RAM-RTC
10  BY_EE 0,100,0 : REM Si effettua la lettura di un byte.
20  POP A : REM Lettura del valore e del risultato dell' operazione.
```

LETTURA/SCRITTURA DI UN BLOCCO SULLA EEPROM SERIALE E SULLA RAM-RTC SERIALE

Sintassi: BL_EE <expr1>, <expr2>, <expr3>

Descrizione istruzione:

Effettua la scrittura o la lettura di un blocco di bytes dall'indirizzo specificato, sulla EEPROM seriale (IC16) o sul PCF-8583 (IC17). Il buffer di scambio dei dati é allocato in RAM ESTERNA e in dettaglio nel range 07E00H÷07EFFH.

Descrizione parametri:

<expr1> --> 0 = Lettura di un blocco.
 --> 1 = Scrittura di un blocco.
 <expr2> --> Indirizzo della locazione da cui scrivere o leggere (0÷<ultimo indirizzo dispositivo).
 <expr3> --> Numero di bytes da scrivere o leggere (1÷255).

I dispositivi (EEPROM e RAM-RTC) sono mappati secondo la seguente tabella:

0000H÷00FFH	->	selezione RAM-RTC	(IC17)
0430H÷04FFH	->	selezione EEPROM 24c02	(IC16)
0430H÷05FFH	->	selezione EEPROM 24c04	(IC16)
0430H÷07FFH	->	selezione EEPROM 24c08	(IC16)

Descrizione output (valore da estrarre con l'istruzione POP ...):

OUTPUT:	0	-->	Scrittura/Lettura OK.
	1	-->	Parametri non corretti.
	2	-->	Dispositivo non funzionante.

Esempi:

```

0   REM ***** Esempio di SCRITTURA di 3 bytes dall' indirizzo 100 della RAM-RTC.
10  XBY(07E00H)=1
20  XBY(07E01H)=2
30  XBY(07E02H)=3
10  BL_EE 1,100,3 : REM Si effettua la scrittura del blocco.
20  POP A : REM Lettura del risultato dell' operazione.

0   REM ***** Esempio di LETTURA di 3 bytes dall' indirizzo 100 della RAM-RTC
10  XBY(07E00H)=0 : REM Si azzera il buffer di ricezione.
20  XBY(07E01H)=0
30  XBY(07E02H)=0
40  BL_EE 0,100,3 : REM Si effettua la lettura di 3 bytes.
50  POP A : REM Lettura del risultato dell' operazione.
60  FOR I=07E00H TO 07E02H : REM Stampa del buffer di ricezione.
70  PRINT XBY(I),
80  NEXT I
  
```

LETTURA/SCRITTURA IN UN SFR (Special Function Register)

Sintassi: RW_SFR <expr1>, <expr2>, <expr3>

Descrizione istruzione:

Effettua la lettura o la scrittura in alcuni Special Function Register. Si ricorda che anche in lettura bisogna indicare tutti i parametri anche se non significativi (<expr3>). Di seguito viene riportata una tabella in cui a ciascun Special Function Register viene associato un CODICE:

NOME REGISTRO	CODICE DI IDENTIFICAZIONE	
CTCON	0	
CTH3	1	
CTH2	2	
CTH1	3	
CTH0	4	
CMH2	5	
CMH1	6	
CMH0	7	
CTL3	8	
CTL2	9	
CTL1	10	
CTL0	11	
CML2	12	
CML1	13	
CML0	14	
IEN1	15	
IP1	16	
RTE	17	
S1ADR	18	
S1DAT	19	
S1STA	20	
S1CON	21	
STE	22	
TMH2	23	
TML2	24	
TM2CON	25	
TM2IR	26	
T3	27	
P4	28	
P5	29	(Si può solo leggerlo)

Descrizione parametri:

<expr1> --> 0÷1 = Selezione dell' operazione (0=Lettura; 1=Scrittura).
 <expr2> --> 0÷29 = Codice di identificazione del registro.
 <expr3> --> 0÷255 = Byte da scrivere (in lettura deve essere comunque fornito anche se non viene utilizzato.).

Descrizione output (valore da estrarre con l'istruzione POP ...):

OUTPUT in SCRITTURA: 0 = Scrittura OK.
 65535 = Errore nel passaggio parametri.

OUTPUT in LETTURA: 0÷255 = Valore letto.
 65535 = Errore nel passaggio parametri.

Esempi:

```
0  REM ***** Esempio di SCRITTURA byte 85 in P4
10 RW_SFR 1,28,85 : POP A
```

```
0  REM ***** Esempio di LETTURA di P4
10 RW_SFR 0,28,0 : POP A
20 IF A<>65535 THEN ? "P4= ",A
```

PROCEDURA DI GESTIONE DELLA TASTIERA OPERATORE

Sintassi: KEYB <expr>

Descrizione istruzione:

Abilita o disabilita la gestione con debouncing della tastiera a matrice 4*6 per la **QTP 24P** o restituisce nello stack il codice dell' ultimo tasto premuto (0 se nessun tasto é premuto).

Nella tabella seguente, sono riportati i codici restituiti dal BASIC in relazione allo scanning della tastiera a matrice 4*6, collegata al connettore **CN5** della **GPC® 553**:

	CN5-PIN 11 (P1.4)	CN5-PIN 12 (P1.5)	CN5-PIN 9 (P1.6)	CN5-PIN 10 (P1.7)
CN5-PIN 5 (P4.5)	68	67	66	65
CN5-PIN 6 (P4.4)	72	71	70	69
CN5-PIN 3 (P4.3)	76	75	74	73
CN5-PIN 4 (P4.2)	54	52	51	50
CN5-PIN 1 (P4.1)	13	48	57	56
CN5-PIN 2 (P4.0)	55	27	53	49

Descrizione parametri:

<expr> --> 0 = Disabilitazione scanning della tastiera.
 --> 1 = Attivazione scanning tastiera.
 --> 2 = Restituisce nello stack il codice dell' ultimo tasto premuto (0 se nessun tasto é premuto). Se lo scanning non era attivo gli si fornisce lo start.

Descrizione output (valore da estrarre con l' istruzione POP ...):

OUTPUT: 0 = Nessun tasto premuto.
 nn = Codice dell' ultimo tasto premuto (fare riferimento alla relativa tabella).
 65535 = Errore nel passaggio parametri.

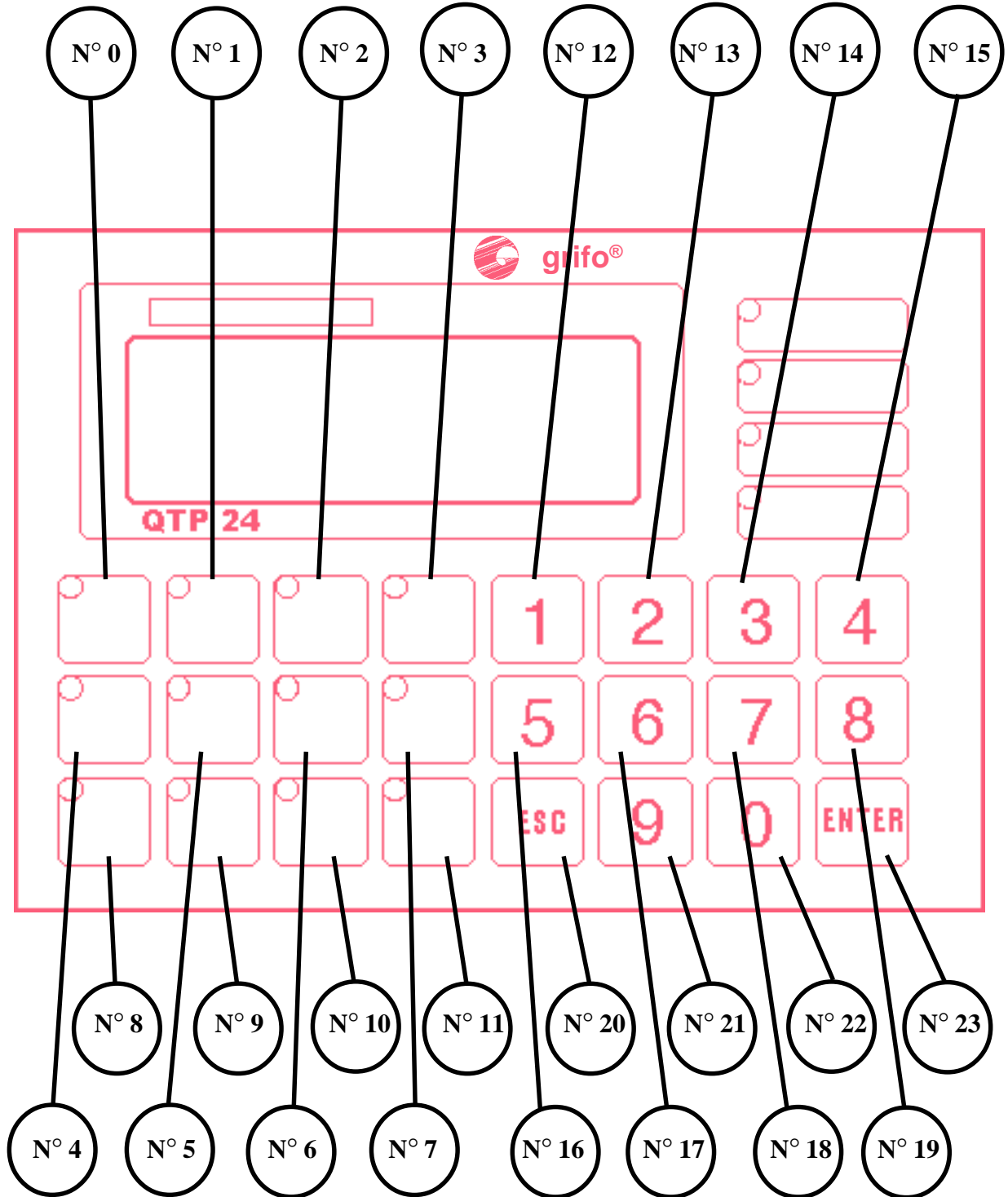
Esempio:

Viene abilitata la gestione della tastiera a matrice e si attende che l' utente prema il tasto associato al codice 48, quando ciò avviene si fa terminare lo scanning della tastiera.

```

0   REM ***** Esempio di gestione della tastiera esterna.
10  KEYB 1 : POP A : REM Abilitazione scanning tastiera.
20  KEYB 2 : POP A : REM Richiesta codice dell' ultimo tasto premuto.
30  IF A<>48 THEN 20
40  KEYB 0 : POP A : REM Disabilitazione scanning della tastiera.
50  END
    
```

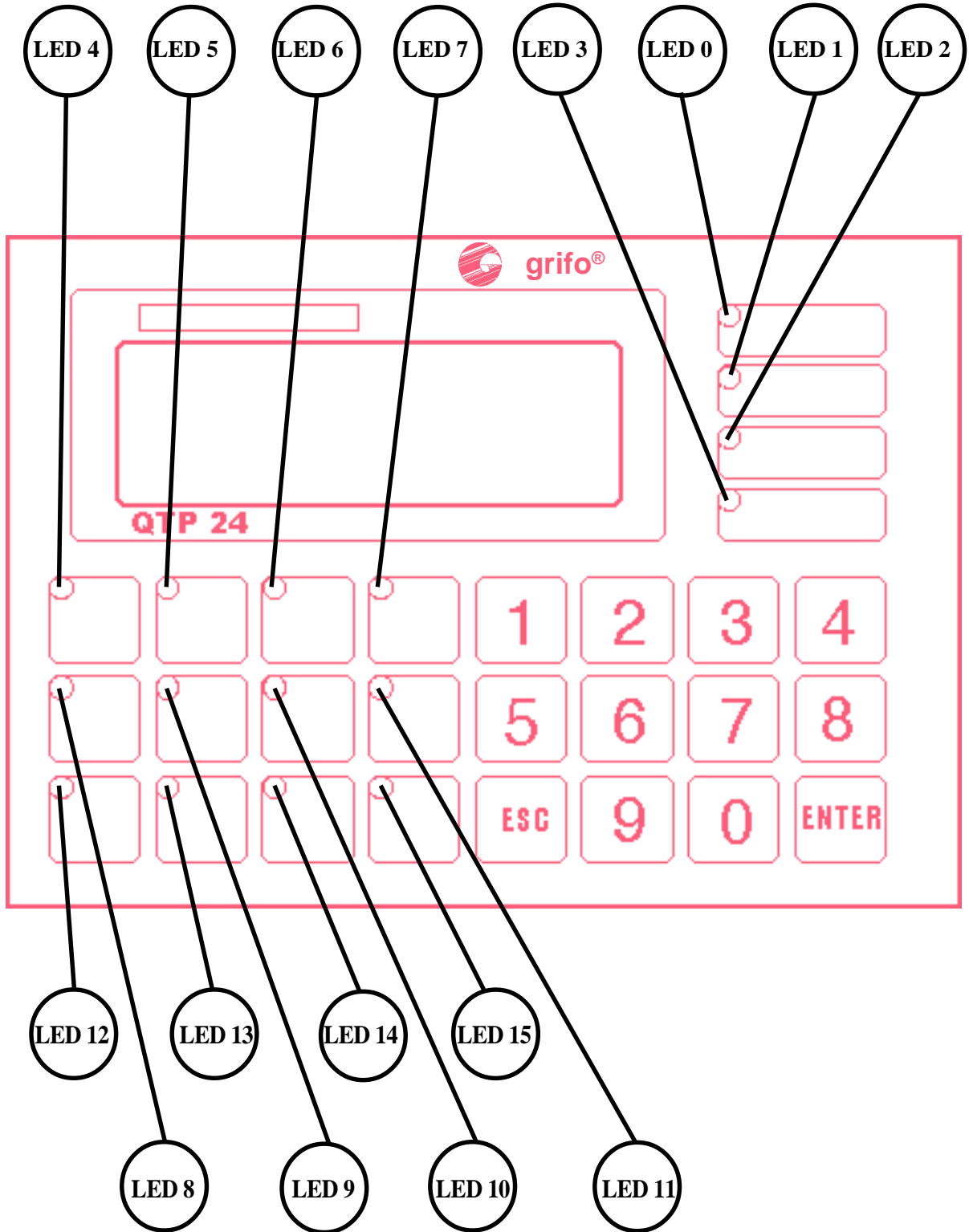
MAPPA DEI TASTI PRESENTI SULLA QTP 24P



CODICI DEI TASTI SULLA QTP 24P

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

MAPPA DEI LEDS PRESENTI SULLA QTP 24P



PROCEDURA DI SELEZIONE E INIZIALIZZAZIONE DEL DISPLAY

Sintassi: DISPLAY <expr>

Descrizione istruzione:

Inizializza il display selezionato. Si ricorda che prima di utilizzare la ridirezione dell' output (UO1) l' utente deve prima utilizzare questa funzione per scegliere il display da utilizzare.

Descrizione parametri:

<expr> --> 0 = FUTABA 20x2
1 = FUTABA 40x1
2 = FUTABA 40x2
3 = FUTABA 40x4
4 = LCD 20x2
5 = LCD 20x4
6 = LCD 40x1
7 = LCD 40x2
8 = LCD 40x4

Descrizione output (valore da estrarre con l' istruzione POP ...):

OUTPUT: Nessuno.

Esempio:

```
10 DISPLAY 5 : REM Selezione display LCD20x4.  
20 UO1 : REM Abilitazione ridirezione OUTPUT  
30 PRINT "GRIFO", : REM Stampa di una stringa sul display  
40 UO0 : REM Disabilitazione ridirezione OUTPUT
```

PROCEDURA DI GESTIONE DELLA LINEA SERIALE SOFTWARE

Sintassi: COM2 <expr>

Descrizione istruzione:

Gestisce tutte le operazioni sulla linea seriale software (TXB=PIN 2 di CN3B; RXB=PIN 5 di CN3B). Per la trasmissione sulla seconda seriale utilizzare l'istruzione PRINT@

Il buffer di ricezione (64 caratteri) é allocato in RAM ESTERNA e più precisamente nel range 07F00H÷07F3FH.

Si ricorda che se la gestione della seconda seriale é attiva, l'utente non può usare le istruzioni per la gestione del TIMER 0 in quanto esso é utilizzato come Baud-Rate generator.

Descrizione parametri:

<expr> --> 0 = Disabilita la gestione della seconda seriale.
 1 = Abilita la gestione della seconda seriale a 1200 BAUD.
 2 = Abilita la gestione della seconda seriale a 2400 BAUD.
 3 = Abilita la gestione della seconda seriale a 4800 BAUD.
 4 = Richiesta del numero di caratteri presenti nel buffer di ricezione.
 5 = Reset del buffer di ricezione.

Descrizione output (valore da estrarre con l'istruzione POP ...):

OUTPUT: 0 = Operazione OK se <expr>=0, 1, 2, 3, 5.
 0÷63 = Numero di caratteri ricevuti se <expr>=4.
 65535 = Errore nel passaggio parametri.

Esempio:

```

0   REM ***** Esempio di gestione della seconda seriale: viene effettuato l' echo sulla
1   REM ***** seconda seriale dei caratteri ricevuti dalla seriale primaria e viceversa; si esce
2   REM ***** quando si preme "F".
10  COM2 1 : POP B : REM Attivazione della gestione della seconda seriale a 1200 BAUD.
20  A=GET
30  IF A=0 THEN GOTO 70 : REM Si é ricevuto un carattere dalla seriale A ?
40  IF A=70 THEN GOTO 140 : REM Il carattere ricevuto é "F" ?
50  PRINT@ CHR(A), : REM Si effettua l' echo sulla seconda seriale.
60  GOTO 20
70  COM2 4 : POP A : REM Richiesta del numero di caratteri ricevuti dalla seriale B.
80  IF A<>1 THEN 20
90  FOR I=0 TO A-1
100 PRINT CHR(XBY(07F00H+I)),
110 NEXT I
120 COM2 5 : POP B : REM Reset del buffer di ricezione.
130 GOTO 20
140 COM2 0 : POP B : REM Disattivazione della gestione della seconda seriale.
150 END
  
```

PROCEDURA DI GESTIONE INTERRUPT RTC

Sintassi: ALARM <expr1>, <expr2>, <expr3>, <expr4>, <expr5>, <expr6>, <expr7>, <expr8>

Descrizione istruzione:

Tramite questa procedura si abilita l' interrupt dell' RTC (IC17), é possibile quindi utilizzare questa funzione per generare basi temporali o gestire allarmi. Per chiarimenti sulla funzionalità dell' RTC, fare riferimento alla relativa documentazione.

Si ricorda che é possibile gestire l' interrupt proveniente dall' RTC, connettendo opportunamente il jumper J13 della **GPC® 553** (posizione 1-2). In questo modo tale segnale risulterà collegato alla linea /INT1 della CPU, che viene gestita dalla funzione **ONEX1** del **BASIC 52**.

Descrizione parametri:

<expr1> --> 0 = Programmazione NO CLOCK ALARM (ALARM MODE).
 1 = Programmazione DAILY ALARM (ALARM MODE).
 2 = Programmazione WEEKDAY ALARM (ALARM MODE).
 3 = Programmazione DATED ALARM (ALARM MODE).
 4 = Programmazione TIMER (TIMER MODE).
 5 = Reset flag di ALARM.

TIMER MODE

<expr2> --> 0÷99 = Valore di conteggio.
 <expr3> --> 0 = No timer.
 1 = Conta i "CENTESIMI DI SECONDO".
 2 = Conta i "SECONDI".
 3 = Conta i "MINUTI".
 2 = Conta le "ORE".
 2 = Conta i "GIORNI".

ALARM MODE

<expr2> --> Valore che esprime le ore (0÷23).
 <expr3> --> Valore che esprime i minuti (0÷59).
 <expr4> --> Valore che esprime i secondi (0÷59).
 <expr5> --> Valore che esprime il giorno della settimana (0÷6).
 <expr6> --> Valore che esprime il giorno del mese (1÷31).
 <expr7> --> Valore che esprime il mese (1÷12).
 <expr8> --> Valore che esprime l' anno (0÷3).

Descrizione output (valore da estrarre con l' istruzione POP ...):

OUTPUT: 0 = OK
 1 = ERRORE durante l' accesso all' RTC.
 2 = ERRORE nel passaggio parametri.

Esempi:

```
0   REM ***** Esempio di generazione di una base temporale di un secondo.
10  MTOP=07DFFH
20  B=1 : REM Inizializzazione contatore del numero di interrupt.
30  ALARM 4, 1, 2, 0, 0, 0, 0, 0 : POP C : REM Attivazione interrupt ogni secondo.
40  ONEX1 100
50  GOTO 50
100 ? "Numero di interrupt avvenuti= ", B
110 B=B+1
120 ALARM 5, 0, 0, 0, 0, 0, 0, 0 : POP C : REM Reset flag di alarm.
130 RETI
```

```
0   REM ***** Esempio di generazione interrupt ad una certa DATA.
10  MTOP=07DFFH
20  ALARM 3, 12, 30, 0, 3, 25, 12, 1 : POP C : REM Interrupt il 25 Dicembre 1997 12:30:00.
30  ONEX1 100
40  GOTO 40
100 ? "Allarme avvenuto."
110 ALARM 5, 0, 0, 0, 0, 0, 0, 0 : POP C : REM Reset flag di alarm.
120 RETI
```

PROCEDURA DI GESTIONE DELLA RIDIREZIONE OUTPUT UTENTE

Il **BASIC 553** include specifiche procedure che permettono la ridirezione dell' OUTPUT mediante le istruzioni UO1 e UO0. L' utente perciò può ridirezionare l' OUTPUT in modo da poter utilizzare l' istruzione PRINT per visualizzare informazioni sul display selezionato e per spedire alcuni comandi spiegati in seguito:

Esempio:

```
10 DISPLAY 5 : REM Selezione display LCD 20x4.
20 UO1 : REM Abilitazione ridirezione OUTPUT
30 PRINT "Hello", : REM Stampa di una stringa sul display
40 PRINT "Grifo", : REM Stampa di una stringa sul display
50 PRINT CHR(1), : REM Si posiziona il cursore nella posizione di HOME.
60 PRINT CHR(27), : PRINT CHR(80), : REM Si disattiva il cursore.
70 UO0 : REM Disabilitazione ridirezione OUTPUT
```

E' possibile visualizzare sul display di bordo tutti i caratteri aventi un codice compreso nel range **32÷255 (20÷FF Hex)**; se viene inviato un codice al di fuori di questo range, e questo non é un comando, viene ignorato. I caratteri relativi ai codici compresi nel range **32÷127 (20÷7F Hex)**, corrispondono a quelli della tabella ASCII standard, mentre i caratteri associati ai codici **128÷255 (80÷FF Hex)**, variano in funzione del tipo di display installato. Per questo motivo si faccia riferimento alle varie tabelle alla fine di questa documentazione. Il carattere viene visualizzato nella posizione attuale del cursore, e quest'ultimo avanzerà nella posizione successiva; se si trova nell'ultimo carattere, in basso a destra, del display, verrà posizionato nella posizione di Home.

COMANDI PER IL POSIZIONAMENTO DEL CURSORE

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

CURSOR LEFT

Codice: **21** **(15 Hex)**
Mnemonic: **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.

CURSOR RIGHT

Codice: **06**
Mnemonic: **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.

CURSOR DOWN

Codice: **10** **(0A Hex)**
Mnemonic: **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.

CURSOR UP

Codice: **26** **(1A Hex)**
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 del display.

HOME

Codice: 01
Mnemonico: SOH

Il cursore viene posto nella posizione di Home, corrispondente alla prima riga ed alla prima colonna del display.

CARRIAGE RETURN

Codice: 13 (0D Hex)
Mnemonico: CR

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

CARRIAGE RETURN + LINE FEED

Codice: 29 (1D Hex)
Mnemonico: 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 CON OFFSET 20H

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

Il cursore viene posizionato nel punto assoluto, indicato tramite "r" e "c". Questi codice esprimono i valori di riga e colonna del display, a cui va aggiunto un offset di **32 (20 Hex)**. Se, per esempio, si desidera posizionare il cursore nella posizione di Home (riga 0, colonna 0), sarà necessario inviare la seguente sequenza: **27 89 32 32**. 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 dal display.

BACKSPACE

Codice: 08
Mnemonic: *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.

CLEAR PAGE

Codice: 12 (0C Hex)
Mnemonic: *FF*

Viene cancellato l'intero display ed il cursore va nella posizione di Home.

CLEAR LINE

Codice: 25 (19 Hex)
Mnemonic: *EM*

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

CLEAR END OF LINE

Codice: 27 75 (1B 4B Hex)
Mnemonic: *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 **Clear End Of Line**. Se, per esempio, il cursore si trova all'inizio di una riga del display, verrà cancellata l'intera linea.

CLEAR END OF PAGE

Codice: 27 107 (1B 6B Hex)
Mnemonic: *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 **Clear End Of Page**. Se, per esempio, il cursore si trova nella posizione di home, verrà cancellato l'intero display.

COMANDI PER LA GESTIONE DEGLI ATTRIBUTI DEL CURSORE

Sono riportati di seguito, i comandi relativi ai vari attributi del cursore, disponibili in funzione dei vari display installabili.

DISATTIVAZIONE DEL CURSORE

Codice: **27 80** (**1B 50 Hex**)
Mnemonic: **ESC P**

Il cursore viene disattivato e non é più visibile.

ATTIVAZIONE DEL CURSORE FISSO

Codice: **27 79** (**1B 4F Hex**)
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.

Questo comando non é disponibile quando é installato il display **Futaba 40x4**.

ATTIVAZIONE DEL CURSORE LAMPEGGIANTE

Codice: **27 77** (**1B 4D Hex**)
Mnemonic: **ESC M**

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

N.B.

Questo comando é disponibile solo con i display **Futaba** del tipo: **20x2** e **40x1**; se viene inviato quando é montato un diverso tipo di display, viene ignorato.

ATTIVAZIONE DEL CURSORE “BLOCCO” LAMPEGGIANTE

Codice: **27 81** (**1B 51 Hex**)
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.

N.B.

Questo comando é disponibile solo per i display **LCD**, in caso contrario viene ignorato.

COMANDI PER LA GESTIONE DEI LEDS

Sono riportati di seguito, i comandi per la gestione dei LEDs di bordo della **QTP 24P**.

ATTIVAZIONE DI UN LED

Codice: *27 50 n.LED Attr. (1B 32 n.LED Attr. Hex)*

Mnemonic: *ESC 2 ASCII(n.LED) ASCII(Attr.)*

Viene attivato il LED indicato in “n.LED”, con l’attributo specificato in “Attr.”.

Se si utilizza una scheda **QTP 24P** i numeri dei LED sono compresi nel range **0÷15**, come rappresentato nella figura a pagina 11.

Gli attributi disponibili sono i seguenti:

<i>0</i>	<i>LED disattivato</i>
<i>255 (FF Hex)</i>	<i>LED attivato</i>
<i>85 (55 Hex)</i>	<i>LED lampeggiante (Blinking)</i>

Se, per esempio, si vuole attivare il LED 5 con l’attributo di Blinking, sarà necessario inviare la seguente sequenza: **27 50 5 85**. Se il parametro con il numero del LED, o quello con l’attributo, non é valido, il comando viene ignorato.

ATTIVAZIONE MASCHERA DI LED

Codice: *27 52 byte1 byte2 byte3 (1B 34 byte1 byte2 byte3 Hex)*

Mnemonic: *ESC 4 ASCII(byte1) ASCII(byte2) ASCII(byte3)*

Vengono gestiti contemporaneamente tutti i LEDs presenti sulla **QTP 24P** come indicato in “byte1”, “byte2” e “byte3”, secondo il seguente codice:

<i>byte1 (bit 0 ...7)</i>	<i>LED 0 ... LED 7</i>
<i>byte2 (bit 0 ...7)</i>	<i>LED 8 ... LED 15</i>
<i>byte3</i>	<i>Non significativo (si deve comunque spedire per compatibilità)</i>

Se un bit é posto a 0, il LED relativo risulterà spento, viceversa questo sarà acceso, se il bit in questione é posto a 1. Se dei LEDs possiedono l’attributo di Blinking, questo viene disattivato.

COMANDI PER LA GESTIONE DELLA TASTIERA

RICONFIGURAZIONE DI UN TASTO

Codice: *27 55 n.tasto codice* (*1B 37 n.tasto codice Hex*)

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

Il tasto indicato viene riconfigurato, ed ogni volta che questo verrà premuto, la scheda invierà in seriale il nuovo codice specificato.

Il numero del tasto da riconfigurare, che si può ricavare dalla figura a pagina 9, deve essere compreso nel range **0÷23 (0÷17 Hex)**, se questo non avviene il comando viene ignorato.

Il valore del codice può variare nel range **0÷254 (0÷FE Hex)**, in quanto il valore **255 (FF Hex)**, indica che il tasto in questione deve essere disattivato, e quindi, quando lo si preme non si riceverà nessun codice.

N.B.

Tale comando comporta la scrittura di un dato nella EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che essa sia presente a bordo scheda.

NOTE PER IL COMPILATORE BXC51 REL. 5.0

Tramite tale compilatore, è possibile velocizzare notevolmente l'esecuzione di un programma scritto in **BASIC 52** (INTERPRETE) infatti il codice generato non viene più interpretato, ma viene eseguito direttamente dal microprocessore.

Per la compilazione è necessario digitare nella linea di comando alcune opzioni come mostrato di seguito:

BXC51 -2 -b<BAUD> -c7F00 -u7DFF -bCRUNxxyy <Nome file> (RAM=32K)

Il file CRUNxxyy.BXL dove xx esprime la frequenza del quarzo e yy il numero di versione, è il RUN-TIME per il compilatore e viene fornito dalla grifo® quando si acquista il BXC51.

L'opzione -b<BAUD> indica al compilatore che non deve generare il codice per la determinazione automatica del Baud-Rate per la seriale, ma deve inizializzare la seriale con un Baud-Rate=BAUD dove BAUD può assumere i seguenti valori: 1200, 2400, 4800, 9600, 19200.

Si ricorda che se il quarzo a bordo scheda è 22118400 Hz, il Baud-Rate effettivo è doppio rispetto a quello indicato nella riga di comando, per esempio se si imposta -b19200, il baud_rate effettivo sarà di 38400 BAUD.

Se il quarzo a bordo scheda non è un multiplo di 11059200 Hz, l'opzione -b<BAUD> ha solo il compito di eliminare la generazione del codice per la determinazione automatica del BAUD-RATE. Per l'inizializzazione di quest'ultimo è necessario modificare il source BASIC, inserendo direttamente i valori corretti negli appositi registri della CPU. Come esempio di seguito vengono riportate le due versioni di uno stesso programma che deve essere eseguito per un quarzo da 14745600 Hz:

```
0   REM ***** Source ORIGINALE eseguito con l' INTERPRETE.
10  PRINT "Hello word"
```

```
0   REM ***** Source MODIFICATO per la compilazione con il BXC51
5   TIMER1=0FCFCH : REM Baud=19200.
10  PRINT "Hello word"
```

Per ulteriori informazioni sulle restanti opzioni mostrate nelle linee di comando fare riferimento al manuale specifico del BXC51.

TABELLA RIASSUNTIVA DEI COMANDI QTP 24P

<i>COMANDO</i>	<i>CODICE</i>	<i>CODICE HEX</i>	<i>MNEMONICO</i>
HOME	01	01	SOH
CURSOR LEFT	21	15	NACK
CURSOR RIGHT	06	06	ACK
CURSOR DOWN	10	0A	LF
CURSOR UP	26	1A	SUB
CARRIAGE RETURN	13	0D	CR
CR+LF	29	1D	GS
Posiz. assoluto cursore con offset 20H	27 89 r c	1B 59 r c	ESC Y ASCII(r) ASCII(c)
BACKSPACE	08	08	BS
CLEAR PAGE	12	0C	FF
CLEAR LINE	25	19	EM
CLEAR END OF LINE	27 75	1B 4B	ESC K
CLEAR END OF PAGE	27 107	1B 6B	ESC k
Disattivazione cursore	27 80	1B 50	ESC P
Attivazione cursore Fisso	27 79	1B 4F	ESC O
Attivazione cursore Lampeggiante	27 77	1B 4D	ESC M
Attivazione cursore "Blocco" Lampeggiante	27 81	1B 51	ESC Q
Attivazione LEDS	27 50 n.LED Attr.	1B 32 n.LED Attr.	ESC 2 ASCII(n.LED) ASCII(Attr.)
Attivazione maschera LEDS	27 52 byte1 byte2 byte3	1B 34 byte1 byte2 byte3	ESC 4 ASCII(byte1) ASCII(byte2) ASCII(byte3)
Riconfigurazione tasto	27 55 n.Tasto cod.	1B 37 n.Tasto cod.	ESC 7 ASCII(n.Tasto) ASCII(cod.)

TABELLE CON CARATTERI DEI DISPLAY

		Higher 4-bit (D4 to D7) of Character Code (Hexadecimal)															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Lower 4-bit (D0 to D3) of Character Code (Hexadecimal)	0	CG RAM (1)	!	"	#	\$	%	&	'	()	*	+	,	-	.	:
	1	CG RAM (2)	;	<	=	>	?	@	A	B	C	D	E	F	G	H	I
	2	CG RAM (3)	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	3	CG RAM (4)	Y	Z	[]	^	_	`	{	}	~					
	4	CG RAM (5)															
	5	CG RAM (6)															
	6	CG RAM (7)															
	7	CG RAM (8)															
	8	CG RAM (1)															
	9	CG RAM (2)															
	A	CG RAM (3)															
	B	CG RAM (4)															
	C	CG RAM (5)															
	D	CG RAM (6)															
	E	CG RAM (7)															
	F	CG RAM (8)															

TABELLA CON CARATTERI DISPLAY LCD 20x2



LOWER 4-BIT	MSB	0000	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
LSB XXXX0000	(1)	RAM	0	1	2	3	4	5	6	7	8	9	A	B	C	D
XXXX0001	(2)	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
XXXX0010	(3)	T	U	V	W	X	Y	Z	[]	^	_	~	0	1	2
XXXX0011	(4)	3	4	5	6	7	8	9	:	;	<	=	>	?	@	A
XXXX0100	(5)	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
XXXX0101	(6)	Q	R	S	T	U	V	W	X	Y	Z	[]	^	_	~
XXXX0110	(7)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>
XXXX0111	(8)	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J
XXXX1000	(9)	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
XXXX1001	(10)	Z	[]	^	_	~	0	1	2	3	4	5	6	7	8
XXXX1010	(11)	9	:	;	<	=	>	?	@	A	B	C	D	E	F	G
XXXX1011	(12)	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
XXXX1100	(13)	W	X	Y	Z	[]	^	_	~	0	1	2	3	4	5
XXXX1101	(14)	6	7	8	9	:	;	<	=	>	?	@	A	B	C	D
XXXX1110	(15)	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
XXXX1111	(16)	T	U	V	W	X	Y	Z	[]	^	_	~	0	1	2

TABELLA CON CARATTERI DISPLAY LCD 20x4 E 40x2



LOWER 4-BIT HEXADECIMAL

Upper 4 bits Lower 4 bits	0000 (0)	0010 (2)	0011 (3)	0100 (4)	0101 (5)	0110 (6)	0111 (7)	1010 (A)	1011 (B)	1100 (C)	1101 (D)	1110 (E)	1111 (F)
xxxx0000 (0)	CG RAM (1)		0	1	2	3	4	5	6	7	8	9	A
xxxx0001 (1)	(2)	.	1	A	0	a	4	u	7	+	4	ä	9
xxxx0010 (2)	(3)	"	2	B	R	b	r	r	7	u	x	æ	0
xxxx0011 (3)	(4)	#	3	C	S	c	s	j	9	t	e	æ	0
xxxx0100 (4)	(5)	\$	4	D	T	d	t	\	I	A	+	W	0
xxxx0101 (5)	(6)	%	5	E	U	e	u	.	+	+	1	æ	0
xxxx0110 (6)	(7)	&	6	F	V	v	v	9	n	2	3	æ	Σ
xxxx0111 (7)	(8)	'	7	G	W	w	w	7	+	x	7	9	π
xxxx1000 (8)	(1)	(8	H	X	x	x	4	0	*	U	7	x
xxxx1001 (9)	(2))	9	I	Y	y	y	6	7	U	W	.	U
xxxx1010 (A)	(3)	*	:	J	Z	z	z	æ	0	n	v	J	+
xxxx1011 (B)	(4)	+	:	K	L	k	l	+	7	e	0	7	π
xxxx1100 (C)	(5)	.	<	L	*	l	l	7	5	0	7	+	π
xxxx1101 (D)	(6)	-	=	n	I	n	3	u	z	\	0	+	÷
xxxx1110 (E)	(7)	.	>	N	^	n	+	a	e	+	.	7	
xxxx1111 (F)	(8)	/	?	O	_	o	+	u	v	7	8	5	■

TABELLA CON CARATTERI DISPLAY LCD 40x1 E 40x4



	D7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1				
	D6	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1				
	D5	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1				
	D4	0	0	1	0	1	0	1	0	1	0	0	1	0	1	1				
D D D D 3 2 1 0		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
0 0 0 0	0		DP		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0 1	1			!	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0 0 1 0	2			"	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0 0 1 1	3			#	3	4	5	6	7	8	9	A	B	C	D	E	F			
0 1 0 0	4	DIM	CU1	\$	4	5	6	7	8	9	A	B	C	D	E	F				
0 1 0 1	5		CU2	%	5	6	7	8	9	A	B	C	D	E	F					
0 1 1 0	6		CU3	&	6	7	8	9	A	B	C	D	E	F						
0 1 1 1	7		DC	'	7	8	9	A	B	C	D	E	F							
1 0 0 0	8	BS		(8	9	A	B	C	D	E	F								
1 0 0 1	9	HT)	9	A	B	C	D	E	F									
1 0 1 0	A			*	A	B	C	D	E	F										
1 0 1 1	B			+	B	C	D	E	F											
1 1 0 0	C			,	C	D	E	F												
1 1 0 1	D	CLR		-	D	E	F													
1 1 1 0	E			.	E	F														
1 1 1 1	F	ALD	RST	/	F															

TABELLA CON CARATTERI DISPLAY FLUORESCENTE 20x2



		D7	D6	D5	D4												
		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
		0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	
		0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	
		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
3210	DDDD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			0	a	P	\	F	A	E		-	7	E	e	P	
0001	1			1	A	Q	a	a	A	e	e	7	7	4	a	a	
0010	2			"	2	B	R	b	r	A	E	T	4	7	e	e	
0011	3			#	3	C	S	c	s	A	R	U	7	T	E	e	
0100	4			\$	4	D	T	d	t	a	e	#	\	I	T	P	
0101	5			%	5	E	U	e	u	E	O	.	7	7	1	e	
0110	6			&	6	F	U	f	u	O	*	7	7	2	e	P	
0111	7			'	7	G	V	g	v	O	*	7	7	7	7	e	
1000	8			(8	H	X	h	x	O	1	7	7	7	7	e	
1001	9)	9	I	Y	i	y	O	7	7	7	7	7	e	
1010	A			*	*	J	Z	j	z	U	4	e	7	7	7	e	
1011	B			+	+	K	[k	[U	4	7	7	7	7	e	
1100	C			<	<	L]	l]	\	7	7	7	7	7	e	
1101	D			=	=	M	^	m	^	7	7	7	7	7	7	e	
1110	E			>	>	N	^	n	^	7	7	7	7	7	7	e	
1111	F			?	?	O	_	o	_	7	7	7	7	7	7	e	

TABELLA CON CARATTERI DISPLAY FLUORESCENTE 20x4



	D7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
	D6	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	
	D5	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	
	D4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
D3 D2 D1 D0		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0 0	0		DP		0	0	0	0	0	0	0	0	0	0	0	0	0
0 0 0 1	1		DC1	:	1	0	0	0	0	0	0	0	0	0	0	0	0
0 0 1 0	2		DC2	"	2	0	0	0	0	0	0	0	0	0	0	0	0
0 0 1 1	3	DEF		#	3	0	0	0	0	0	0	0	0	0	0	0	0
0 1 0 0	4	DIM		\$	4	0	0	0	0	0	0	0	0	0	0	0	0
0 1 0 1	5			%	5	0	0	0	0	0	0	0	0	0	0	0	0
0 1 1 0	6			&	6	0	0	0	0	0	0	0	0	0	0	0	0
0 1 1 1	7			'	7	0	0	0	0	0	0	0	0	0	0	0	0
1 0 0 0	8	BS		(8	0	0	0	0	0	0	0	0	0	0	0	0
1 0 0 1	9	HT)	9	0	0	0	0	0	0	0	0	0	0	0	0
1 0 1 0	A	LF		*	A	0	0	0	0	0	0	0	0	0	0	0	0
1 0 1 1	B			+	B	0	0	0	0	0	0	0	0	0	0	0	0
1 1 0 0	C			,	C	0	0	0	0	0	0	0	0	0	0	0	0
1 1 0 1	D	CR		-	D	0	0	0	0	0	0	0	0	0	UF0	0	0
1 1 1 0	E			.	E	0	0	0	0	0	0	0	0	0	UF1	0	0
1 1 1 1	F		RST	/	F	0	0	0	0	0	0	0	0	0	UF2	0	0

TABELLA CON CARATTERI DISPLAY FLUORESCENTI 40x1 E 40x2

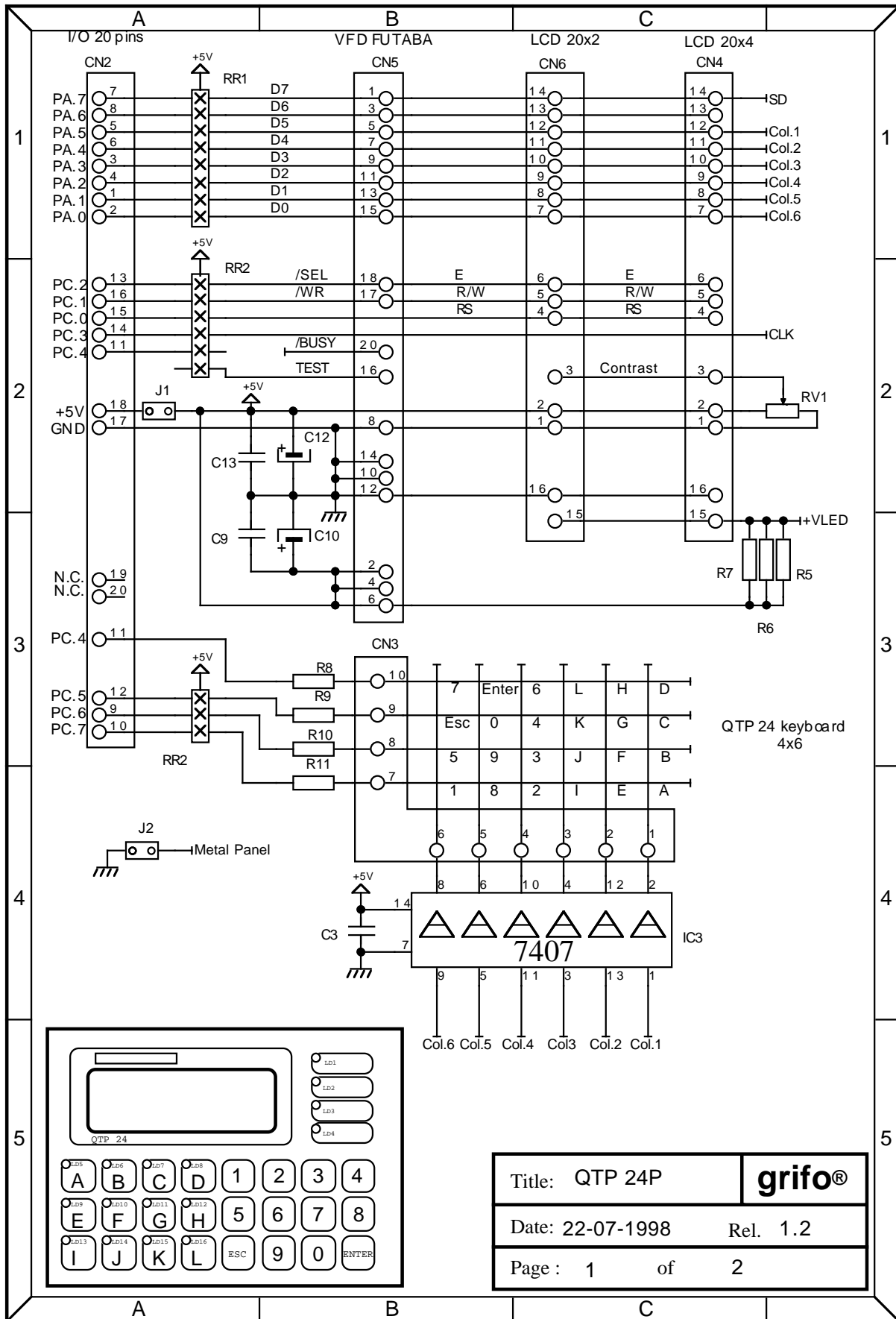


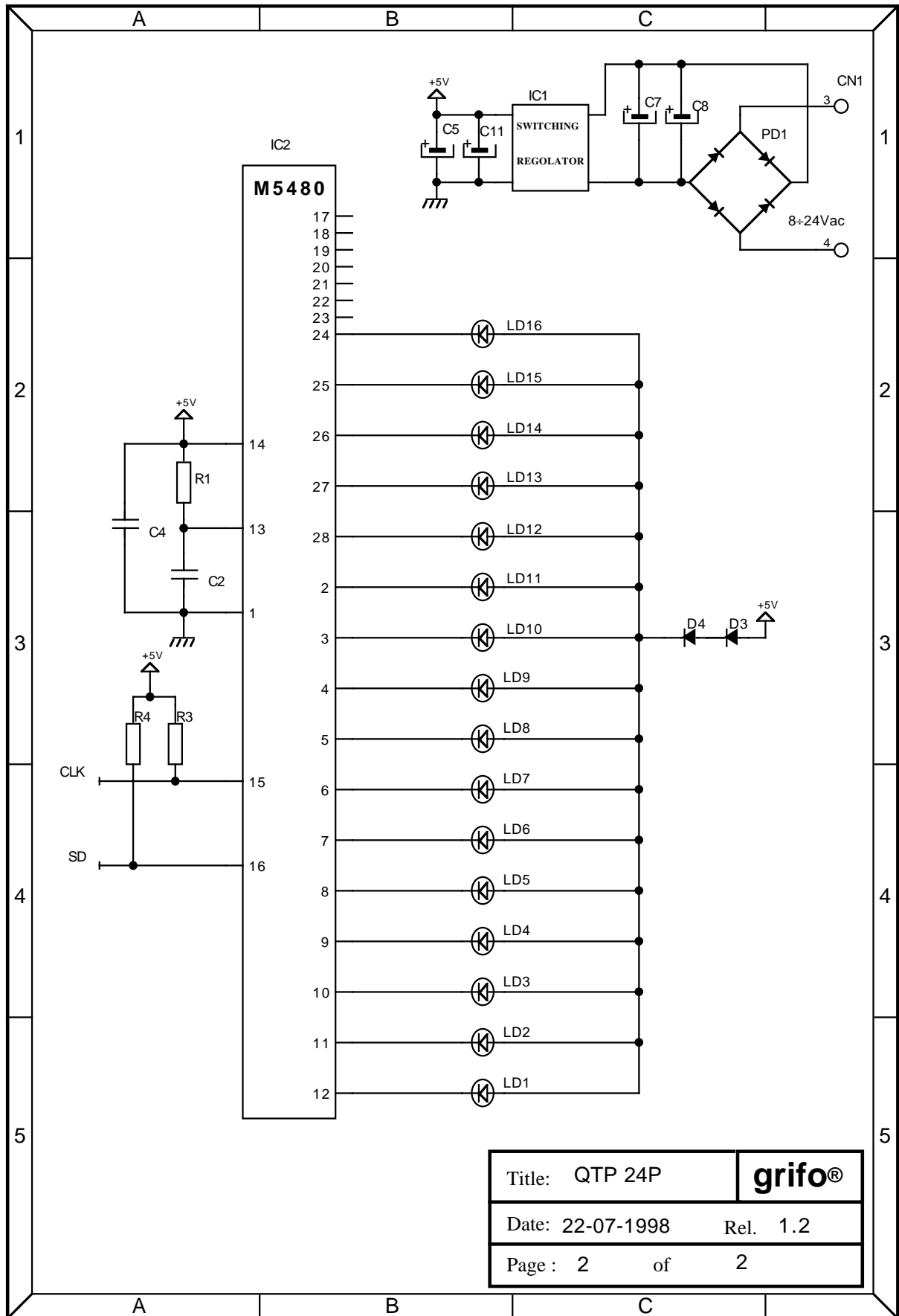
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000		DP		0	1	2	3	4	5	6	7	8	9	-	.	/
xxxx0001		DC1	:	1	A	Q	W	F	B	E	=	T	F	Δ	↓	↘
xxxx0010		DC2	"	2	R	E	T	E	T	T	∞	∞	∞	∞	∞	∞
xxxx0011	DEF	DC3	#	3	S	C	S	Δ	R	∞	∞	∞	∞	∞	∞	∞
xxxx0100	DIM	DC4	⊕	4	O	T	d	t	e	.	.	I	T	+	+	∞
xxxx0101		DC5	⊗	5	E	U	a	u	∞	X	=	∞	∞	∞	∞	∞
xxxx0110			⊙	6	F	V	t	v	∞	∞	∞	∞	∞	∞	∞	∞
xxxx0111			∞	7	G	W	a	w	∞	-	∞	∞	∞	∞	∞	∞
xxxx1000	BS		∞	8	X	H	X	∞	∞	∞	∞	∞	∞	∞	∞	∞
xxxx1001			∞	9	V	v	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
xxxx1010		UP	∞	J	Z	z	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
xxxx1011	HM CLR	DWN	∞	K	K	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
xxxx1100		RT	∞	L	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	UF0
xxxx1101	CR + LF	LT	∞	M	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	UF1
xxxx1110			∞	N	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	UF2
xxxx1111		RST	∞	O	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	

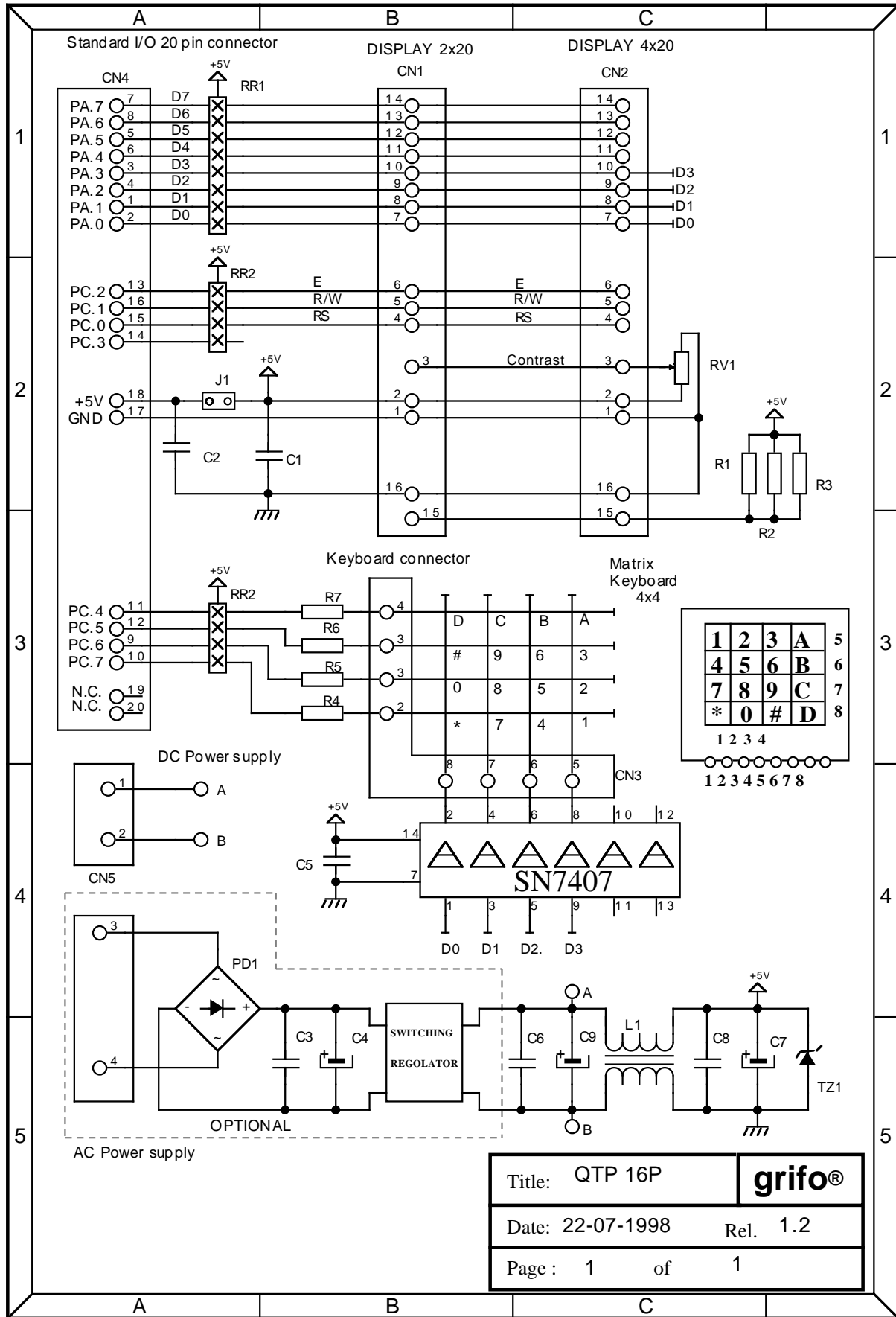
TABELLA CON CARATTERI DISPLAY FLUORESCENTE 40x4



SCHEMI ELETTRICI







Title: QTP 16P	grifo®
Date: 22-07-1998	Rel. 1.2
Page: 1	of 1



