

BASIC 52

8052 MICROCONTROLLER BASIC

GUIDA RAPIDA

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>

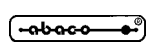
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



BASIC 52

Edizione 3.00

Rel. 11 Gennaio 1999

, GPC[®], grifo[®], sono marchi registrati della ditta grifo[®]

BASIC 52

8052 MICROCONTROLLER BASIC

GUIDA RAPIDA

MCS BASIC 52 é un potente tool software 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 o EPROM 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. **MCS 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 **MCS BASIC 52** diventa **BASIC** e di seguito la **parte finale del nome della scheda**.

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>

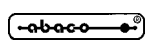
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



BASIC 52

Edizione 3.00

Rel. 11 Gennaio 1999

, GPC[®], grifo[®], sono marchi registrati della ditta grifo[®]

Vincoli sulla documentazione grifo® Tutti i Diritti Riservati

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 nel 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:

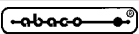


Attenzione: Pericolo generico



Attenzione: Pericolo di alta tensione

Marchi Registrati

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

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

INDICE GENERALE

GUIDA RAPIDA AL BASIC 52	1
CONVENZIONI	1
ELENCO OPERATORI	1
ELENCO ISTRUZIONI	2
VARIAZIONI DEL BASIC 52 PER SCHEDE GRIFO®	12
COMANDI, ISTRUZIONI, OPERATORI ELIMINATI	12
COMANDI AGGIUNTI.....	12
OPERATORI AGGIUNTI.....	12
ISTRUZIONI AGGIUNTE.....	12
GESTIONE SECONDA SERIALE (SERIALE SOFTWARE)	13
GESTIONE CONVERTITORE A/D	14
GESTIONE INTERRUPT REAL TIME CLOCK.....	14
LETTURA/SCRITTURA BLOCCO SU EEPROM E RAM RTC SERIALI	15
LETTURA/SCRITTURA BYTE SU EEPROM E RAM RTC SERIALI	15
GESTIONE DELLA TASTIERA OPERATORE	15
SELEZIONE E INIZIALIZZAZIONE DEL DISPLAY	16
INIZIALIZZAZIONE 82C55 PER GESTIONE RIDIREZIONAMENTO CONSOLE .	16
LETTURA/SCRITTURA DI UN SFR (SPECIAL FUNCTION REGISTER)	17
GESTIONE DEL REAL TIME CLOCK	18
GESTIONE DELLE LINEE DI PWM.....	18

INDICE DELLE FIGURE

FIGURA 1: TABELLA ISTRUZIONI AGGIUNTE	13
FIGURA A-1: SCHEMA ELETTRICO KDX x24	A-1
FIGURA A-2: SCHEMA ELETTRICO QTP 24P PARTE 1	A-2
FIGURA A-3: SCHEMA ELETTRICO QTP 24P PARTE 2	A-3
FIGURA A-4: SCHEMA ELETTRICO QTP 16P	A-4
FIGURA A-5: SCHEMA ELETTRICO PPI 82C55	A-5



GUIDA RAPIDA AL BASIC 52

Questa guida rapida al linguaggio di programmazione MCS BASIC 52 elenca le parole chiave dando per ciascuna una breve descrizione di uso e funzione.

CONVENZIONI

La guida usa le seguenti convenzioni tipografiche:

PAROLE CHIAVE (grassetto maiuscolo)

Parola chiave del BASIC 52.

referimento (italico)

Variabili, espressioni, costanti, o altre informazioni che vanno sostituite a seconda delle esigenze.

[*valori opzionali*] (racchiusi in parentesi quadre)

Valori che non sono obbligatoriamente richiesti.

elementi ripetitivi... (seguiti da tre puntini)

Bisogna aggiungere altri valori con la stessa forma di quello precedente.

ELENCO OPERATORI

C = command mode

R = run mode

variabile = espressione

Assegna un valore ad una variabile

C,R

espressione = espressione

Test di equivalenza (operatore relazionale)

C,R

espressione + espressione

Aggiunge

C,R

espressione - espressione

Sottrae

C,R

*espressione * espressione*

Moltiplica

C,R

espressione / espressione

Divide

C,R

*espressione ** espressione*

Eleva la prima espressione al valore della seconda espressione

C,R

espressione <> espressione

Test di diseuguaglianza (operatore relazionale)

C,R

<i>espressione < espressione</i> Test di minoranza (operatore relazionale)	C,R
<i>espressione > espressione</i> Test di maggioranza (operatore relazionale)	C,R
<i>espressione <= espressione</i> Test di minoranza o uguaglianza (operatore relazionale)	C,R
<i>espressione >= espressione</i> Test di maggioranza o uguaglianza (operatore relazionale)	C,R

ELENCO ISTRUZIONI

? Uguale al PRINT	
ABS (<i>espressione</i>) Restituisce il valore assoluto dell'espressione	C,R
<i>espressione .AND. espressione</i> AND logico	C,R
ASC (<i>carattere</i>) Restituisce il valore ASCII del carattere	C,R
ATN (<i>espressione</i>) Restituisce l'arcotangente dell'espressione	C,R
BAUD <i>espressione</i> Setta il baud rate per LPT. Per questa operazione, XTAL deve essere settato con la frequenza del quarzo del sistema.	C,R
CALL <i>intero</i> Chiama una routine in linguaggio assembly all'indirizzo specificato nella memoria programma	C,R
CBY (<i>espressione</i>) Restituisce il valore dell'espressione letto in memoria programma o codice. Questo operatore non può assegnare valori ma soltanto leggerli.	C,R
CHR (<i>expression</i>) Converte l'espressione nel carattere ASCII corrispondente.	C,R
CLEAR Setta tutte le variabili a 0, resetta tutto lo stack e gli interrupt chiamati dal BASIC.	C,R

CLEARI	C,R
Resetta tutti gli interrupt chiamati dal BASIC. Disabilita ONTIME, ONEX1.	
CLEARs	C,R
Resetta tutto lo stack. Setta il control stack = 0FEh, argument stack = 1FEh, internal stack = valore contenuto nella locazione 3Eh in RAM interna	
CLOCK0	C,R
Disabilita il temporizzatore hardware software.	
CLOCK1	C,R
Abilita il temporizzatore hardware software.	
CONT	C
Continua l'esecuzione del programma dopo uno STOP o CONTROL+C.	
COS(espressione)	C,R
Restituisce il coseno dell' <i>espressione</i> .	
CR	
Opzione di PRINT . Genera un carriage return ma non un line feed sulla console collegata.	
DATA espressione [...,espressione]	R
Specifica espressioni che saranno restituite dal comando READ.	
DBY(espressione)	C,R
Restituisce o assegna un valore all'espressione nella memoria dati interna.	
DIM nome array [(dimensione)] [...nome array(dimensione)]	C,R
Riserva spazio per l'immagazzinamento di un array.La dimensione di default é 11 (0-10). I limiti sono 0-254.	
DO: [comandi del programma]: UNTIL espressione relazionale	R
Esegue tutti i comandi tra DO e UNTIL finché l'espressione relazionale é vera.	
DO: [comandi del programma]: WHILE espressione relazionale	R
Esegue tutti i comandi tra DO e UNTIL finché l'espressione relazionale é falsa.	
END	R
Termina il programma in esecuzione.	
EXP (espressione)	C,R
Eleva e (2.7182818) alla potenza dell'espressione.	

<p>FOR <i>variabile di conteggio</i> = <i>espressione di inizio conteggio</i> TO <i>espressione di fine conteggio</i> [STEP <i>espressione di incremento</i>]: [<i>comandi del programma</i>]: NEXT [<i>variabile di conteggio</i>]</p> <p>Esegue tutti i comandi tra FOR e NEXT per un numero di volte specificato dall'<i>espressione</i> di conteggio e di incremento.</p>	C,R
<p>FPROG, FPROG1-FPROG6</p> <p>Come PROG, PROG1÷PROG6, ma usando un algoritmo di programmazione intelligente.</p>	C
<p>FREE</p> <p>Restituisce il numero di byte non usati in RAM dati esterna.</p>	C,R
<p>GET</p> <p>Contiene il codice ASCII di un carattere ricevuto dalla tastiera di un PC remoto. Dopo che il programma legge il valore di GET (per esempio, G=GET), GET restituisce 0 finché non arriva un nuovo carattere.</p>	R
<p>GOSUB <i>numero di linea</i></p> <p>Genera un trasferimento del controllo del programma ad una subroutine che inizia all'indirizzo specificato nel <i>numero di linea</i>. Il comando RETURN restituisce il controllo del programma alla linea successiva al comando GOSUB.</p>	R
<p>GOTO <i>numero di linea</i></p> <p>Genera un salto al <i>numero di linea</i> nel programma corrente.</p>	C,R
<p>IDLE</p> <p>Forza l'MCS BASIC 52 ad aspettare l'interrupt di ONTIME o ONEX1.</p>	R
<p>IE</p> <p>Restituisce o assegna un valore al registro speciale delle funzioni IE dell'8052.</p>	C,R
<p>IF <i>espressione relazionale</i> THEN <i>comandi del programma</i> [ELSE] [<i>comandi del programma</i>]</p> <p>Se l'espressione relazionale é vera, esegue i comandi seguenti lo THEN. Se l'espressione relazionale é falsa, esegue i comandi seguenti l'ELSE se usato.</p>	R
<p>INPUT [<i>“Messaggio del Prompt”</i>][,] <i>variabile</i> [,<i>variabile</i>] [,...<i>variabile</i>]</p> <p>Visualizza un punto interrogativo ed un messaggio di prompt opzionale sul pc remoto e attende un input da tastiera. Immagazzina l'input in una variabile. Una virgola antecedente la prima variabile elimina il punto interrogativo.</p>	R
<p>INT(<i>espressione</i>)</p> <p>Restituisce l'intero dell'espressione.</p>	C,R

IP	C,R
Restituisce o assegna un valore al registro funzioni speciali IP dell'8052.	
LD@ <i>expression</i>	C,R
Restituisce un numero di 6 byte floating point che era stato salvato con <i>-ST@ espressione-</i> . Il numero é posizionato nella argument stack. L'espressione punta al byte più significativo del numero.	
LEN	C,R
Restituisce il numero di byte del programma corrente.	
[LET] <i>variabile = espressione</i>	C,R
Assegna ad una variabile il valore dell'espressione. L'uso di LET é opzionale.	
LIST[<i>numero di linea</i>][-<i>numero di linea</i>]	C,R
Visualizza il programma corrente nel pc remoto.	
LIST# [<i>numero di linea</i>][-<i>numero di linea</i>]	C,R
Scriva il programma corrente per LPT.	
LIST@ [<i>numero di linea</i>][-<i>numero di linea</i>]	C,R
Questo comando fa la stessa cosa di LIST# ma con l'eccezione che l'uscita é ridirezionata verso un driver definito dall'utente. In particolare l'utente deve allocare all'indirizzo 403Ch della memoria codice esterna una routine in linguaggio assembly. Per abilitare tale routine l'utente deve settare il bit 7 nella locazione 27h nella memoria dati esterni.	
LOG(<i>espressione</i>)	C,R
Restituisce il logaritmo naturale dell'espressione.	
MTOP [=<i>indirizzo più alto in RAM</i>]	C,R
Assegna o legge l'indirizzo più alto che l'MCS BASIC 52 userà per immagazzinare variabili, stringhe e programma in RAM.	
NEW	C
Cancella il programma corrente in RAM; resetta tutte le variabili.	
NOT (<i>espressione</i>)	C,R
Restituisce il complemento a 1 (inverso) dell'espressione.	
NULL [<i>intero</i>]	C
Setta il numero (0-255) di caratteri NULL (ASCII 00) che l'MCS BASIC 52 spedisce automaticamente dopo un carriage return. Solo stampanti e terminali molto lenti necessitano di questo extra NULL.	
ON <i>espressione</i> GOSUB <i>numero di linea</i> [,<i>numero di linea</i>] [,...,<i>numero di linea</i>]	R
Trasferisce il controllo del programma ad una subroutine che inizia ad uno dei numeri di linea della lista. Il valore dell'espressione confronta la posizione del numero di linea selezionato con il primo numero di linea alla posizione zero.	

<p>ON <i>espressione</i> GOTO <i>numero di linea</i> [,numero di linea] [...,<i>numero di linea</i>]</p> <p>Trasferisce il controllo del programma ad uno dei numeri di linea della lista. Il valore dell'espressione confronta la posizione del numero di linea selezionato con il primo numero di linea alla posizione zero.</p>	R
<p>ONERR <i>numero di linea</i></p> <p>Passa il controllo al numero di linea seguente un errore aritmetico. Errori aritmetici includono ARITH. OVERFLOW, ARITH. UNDERFLOW, DIVIDE BY ZERO, e BAD ARGUMENT.</p>	R
<p>ONEX1 <i>numero di linea</i></p> <p>Con l'interrupt 1 (pin 13), l'MCS BASIC 52 finisce di eseguire il comando corrente e dopo passa il controllo all'interrupt routine che inizia al numero di linea specificato. L'interrupt routine deve terminare con RETI.</p>	R
<p>ONTIME <i>numero di secondi, numero di linea</i></p> <p>Quando TIME = numero di secondi, l'MCS BASIC 52 passa il controllo alla routine di interrupt che inizia al numero di linea. La routine di interrupt deve terminare con RETI. CLOCK.1 fa partire il timer.</p>	R
<p><i>espressione</i> .OR. <i>espressione</i></p> <p>OR logico.</p>	C,R
<p>P.</p> <p>Uguale al PRINT.</p>	
<p>PCON</p> <p>Restituisce o assegna un valore al registro funzioni speciali PCON dell'8052.</p>	C,R
<p>PGM</p> <p>Programma la EPROM, EEPROM o NV RAM con i dati della memoria. I seguenti dati devono essere immagazzinati nella memoria dati interna nelle locazioni qui elencate:</p> <p>1Bh,19h Byte alto, byte basso del primo indirizzo dati</p> <p>1Ah,18h Byte alto, byte basso del primo indirizzo del programma -1</p> <p>1Fh,1Eh Byte alto, byte basso indicante il numero di byte del programma</p> <p>40h,41h Byte alto, byte basso indicante le dimensioni dell'impulso. byte alto = ((65536 - dimensioni impulso in secondi * XTAL/12) / 256. byte basso = ((65536 - dimensioni impulso in secondi * XTAL/12) .AND. 0FFh.</p> <p>26h Per la programmazione intelligente, settare il bit 3. Per un tempo fisso di 50 millisecondi, resettare il bit 3.</p>	C,R
<p>PH0.</p> <p>Uguale al PRINT, ma visualizza i valori in formato esadecimale. Usa due cifre per visualizzare i valori minori di 0FFh.</p>	C,R
<p>PH0.#</p> <p>Uguale al PRINT#, ma visualizza il valore in formato esadecimale (2 cifre).</p>	C,R

PH0.@ Uguale al PRINT@, ma visualizza il valore in formato esadecimale (2 cifre).	C,R
PH1. Uguale al PRINT, ma visualizza i valori in formato esadecimale. Sempre visualizza 4 cifre.	C,R
PH1.# Uguale al PRINT#, ma visualizza in formato esadecimale (4 cifre).	C,R
PH1.@ Uguale al PRINT@, ma visualizza valori in PH1 in formato esadecimale (4 cifre).	C,R
PI Costante uguale a 3.1415926.	C,R
POP <i>variabile</i> [... <i>variabile</i>] Assegna il valore della cima dello argument stack alla variabile.	C,R
PORT1 Restituisce o assegna un valore al PORT1.	C,R
PRINT [<i>espressione</i>] [... <i>espressione</i>] [,] Visualizza il valore dell'espressione sulla console. Una virgola alla fine del comando elimina il CARRIAGE RETURN/LINEFEED. I valori sono separati da due spazi. Opzioni aggiuntive al PRINT sono CR, SPC, TAB, USING.	C,R
PRINT# Uguale al PRINT, ma l'output é ridirezionato verso LPT. I valori BAUD e XTAL modificano la velocità di PRINT#.	C,R
PRINT@ Uguale al PRINT, ma l'output é ridirezionato verso un driver definito dall'utente. Richiede una routine in linguaggio assembly all'indirizzo 403Ch della memoria esterna. Settando il bit 7 della locazione 24h della memoria interna dati si abilita la routine di output.	C,R
PROG Memorizza in EPROM il programma corrente residente in RAM.	C
PROG1 Salva il baud rate della porta seriale. All'accensione o reset, l'MCS BASIC 52 si avvia senza ricevere un carattere di spazio. Il baud rate del sistema collegato in seriale deve corrispondere a quello salvato.	C
PROG2 Come PROG1, inoltre all'accensione o reset, l'MCS BASIC 52 inizia ad eseguire la prima istruzione in EPROM.	C

PROG3	C
Come PROG1, inoltre salva MTOP. All'accensione o reset, l'MCS BASIC 52 resetta la memoria fino a MTOP.	
PROG4	C
E' l'insieme dei comandi PROG1, PROG2, PROG3.	
PROG5	C
Come PROG3, inoltre legge la locazione 5Fh in memoria dati esterna sul accensione o sul reset. Se tale locazione contiene 0A5h, l'MCS BASIC 52 non resetta la memoria dati esterna. Se la locazione di memoria dati 5Eh contiene 34h, l'MCS BASIC 52 automaticamente inizierà ad eseguire il programma in memoria dati esterna.	
PROG6	C
Come PROG5, ma se la locazione di memoria dati esterna contiene 5Fh, l'MCS BASIC 52 chiama una routine di reset scritta dall'utente in linguaggio assembly che inizia alla locazione 4039h della memoria programma.	
PUSH <i>espressione</i> [... <i>espressione</i>]	C,R
Posiziona il valore dell'espressione sequenzialmente all'interno dello argument stack.	
PWM <i>espressione1</i> , <i>espressione2</i> , <i>espressione3</i>	C,R
Genera sul pin 3 una sequenza di impulsi modulata (PWM). L' <i>espressione 1</i> indica la durata di ogni impulso, espressa in cicli di clock. L' <i>espressione 2</i> é la durata di ogni impulso a livello basso, espresso in cicli di clock. L' <i>espressione 3</i> indica il numero di cicli di output di PWM. Un ciclo di clock = 12/XTAL. Un ciclo di PWM = un impulso alto e un impulso basso. L' <i>espressione1</i> e l' <i>espressione 2</i> devono essere minimo 25. Il valore massimo per ogni espressione é 65535.	
RAM	C
Seleziona il programma corrente in RAM.	
RCAP2	C,R
Restituisce o assegna un valore al registro speciale funzioni RCAP2H e RCAP2L dell'8052.	
READ <i>variabile</i> [..., <i>variabile</i>]	R
Restituisce l'espressione del comando DATA e assegna ogni espressione ad una variabile.	
REM	C,R
Introduce un commento o una nota. Nella linea di programma tutto il testo dopo il REM verrà ignorato dall'MCS BASIC 52.	
RESTORE	R
Resetta il puntatore a READ sulla la prima espressione del comando DATA.	

RETI	R
Restituisce il controllo del programma al numero di linea seguente la più recente esecuzione del comando ONEX1 o ONTIME.	
RETURN	R
Restituisce il controllo del programma al numero di linea seguente la più recente esecuzione del comando GOSUB.	
RND	C,R
Restituisce un numero casuale compreso tra 0 e 1.	
ROM [<i>program number</i>]	C
Seleziona un programma in EPROM (inizio da 8000h). Il programma di default é 1.	
RROM [<i>program number</i>]	C,R
Cambia il ROM mode e lancia il programma specifico. Il programma di default é 1.	
RUN	C
Esegue il programma corrente. Resetta le variabili.	
SGN (<i>espressione</i>)	C,R
Restituisce +1 se l'espressione ≥ 0 , zero se l'espressione =0, e -1 se l'espressione < 0 .	
SIN (<i>espressione</i>)	C,R
Restituisce il seno dell'espressione.	
SPC (<i>espressione</i>)	
Opzione di PRINT. Nella visualizzazione genera l'aggiunta di spazi (altrimenti il minimo é 2) tra i valori del comando PRINT.	
SQR (<i>espressione</i>)	C,R
Restituisce la radice quadrata dell'espressione	
ST@ <i>espressione</i>	C,R
Copia un numero di 6 byte floating point dallo stack alla memoria dati esterna. L'espressione punta al byte più significativo del numero.	
STOP	
Ferma l'esecuzione del programma .	
STRING <i>espressione1, espressione2</i>	C,R
<i>Alloca memoria per stringhe (ogni variabile consiste in una serie di caratteri di testo)</i>	
<i>Espressione1 = (Espressione2 * numero di stringhe) + 1.</i>	
<i>Espressione2 = numero massimo di byte (caratteri) per stringa + 1. Esegue il reset di tutte le variabili STRING. Il numero massimo di stringhe é 255.</i>	

T2CON	C,R
Restituisce o assegna un valore al registro speciale funzioni T2CON dell'8052.	
TAB (<i>espressione</i>)	
Opzione del PRINT. Specifica la posizione (numero di spazi) di inizio visualizzazione del valore del comando PRINT.	
TAN (<i>espressione</i>)	C,R
Restituisce la tangente dell'espressione.	
TCON	C,R
Restituisce o assegna un valore al registro speciale funzioni TCON dell'8052.	
TIME	C,R
Restituisce o assegna un valore, in secondi, per il temporizzatore hardware e software dell'MCS BASIC 52.	
TIMER0	C,R
Restituisce o assegna un valore ai registri TH0 e TL0 dell'8052.	
TIMER1	C,R
Restituisce o assegna un valore ai registri TH1 e TL1 dell'8052.	
TIMER2	C,R
Restituisce o assegna un valore ai registri TH2 e TL2 dell'8052.	
TMOD	C,R
Restituisce o assegna un valore al registro TMOD dell'8052.	
U.	
Opzione del PRINT. Uguale a USING.	
UI0	C,R
Ridireziona l'input da console sulla procedura standard dell'MCS BASIC 52, dopo l'uso di UI1.	
UI1	C,R
Permette all'utente di ridirezionare l'input da consolle su una procedura in linguaggio assembly. La locazione 4033h di memoria programma esterna deve contenere il salto alla procedura assembly.	
UO0	C,R
Ridireziona l'output su console sulla procedura standard dell'MCS BASIC 52, dopo l'uso di UO1.	
UO1	C,R
Permette all'utente di ridirezionare l'output su consolle ad una procedura in linguaggio assembly. La locazione 4033h di memoria programma esterna deve contenere il salto alla procedura assembly.	

USING (FN)

Opzione del PRINT. Genera in uscita un numero in formato esponenziale con N cifre significative. L'MCS BASIC 52 sempre assegna minimo 3 cifre significative. Il valore massimo per N é 8.

USING(0)

Opzione del PRINT. Genera in uscita un numero compreso tra ± 99999999 a ± 0.1 . Numeri fuori da questo range vengono visualizzati in formato USING(FN). USING(0) é il formato di default.

USING (#[...#][.]#[...#])

Opzione del PRINT. Genera in uscita un valore usando i decimali, con # vengono rappresentati il numero di cifre significativi prima e dopo la virgola.

XBY(espressione)

Restituisce o assegna un valore alla memoria dati esterna.

C,R

XFER

Copia il programma corrente dalla EPROM (inizio 8010h per il programma 1) alla RAM (inizio 200h) e seleziona RAM mode.

C

espressione .XOR. espressione

OR logico esclusivo

C,R

XTAL

Assegna un valore uguale alla frequenza del quarzo del sistema. L'MCS BASIC 52 utilizza questo valore per il temporizzatore hardware e software, per la programmazione e per il baud rate della porta seriale.

C,R

VARIAZIONI DEL BASIC 52 PER SCHEDE GRIFO®

Segue una breve descrizione delle variazioni all'MCS BASIC 52=**BASIC xxx** effettuate dalla **grifo®** per soddisfare meglio le esigenze degli utenti.

COMANDI, ISTRUZIONI, OPERATORI ELIMINATI

Comandi eliminati

LIST#
FPROG
FPROG1
FPROG2
FPROG3
FPROG4
FPROG5
FPROG6

Istruzioni eliminate

BAUD
PRINT#
PH0.#
PH1.#
PWM

Operatori eliminati

Nessuno

COMANDI AGGIUNTI

Comandi aggiunti

ERASE ->

Funzione

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

OPERATORI AGGIUNTI

Operatori aggiunti

Nessuno ->

Funzione

-

ISTRUZIONI AGGIUNTE

Viene di seguito riportata una descrizione tabellare delle istruzioni e comandi aggiunti rispetto all'MCS BASIC 52 originale. Tali aggiunte servono per gestire ad alto livello le risorse hardware della scheda in uso e quindi semplificare lo sviluppo del programma applicativo.

Comandi e istruzioni	BASIC 52 per								
	GPC® F2	GPC® 51	GPC® 552	GPC® 553	GPC® 554	GPC® 323	GPC® 324	GPC® R63	GPC® T63
ERASE	•	•	•	•	•	•	•	•	•
A_D			•	•	•	•			
ALARM			•	•		•			
BL_EE BL_EE2			•	•	•	•	•		
BY_EE			•	•	•	•	•		
COM2			•	•	•	•	•		
DISPLAY			•	•	•	•	•		
GES_RTC			•	•		•			
KEYB			•	•	•	•	•		
P82C55			•			•	•		
RW_SFR			•	•	•	•	•		
SET_PWM			•	•	•				

FIGURA 1: TABELLA ISTRUZIONI AGGIUNTE

GESTIONE SECONDA SERIALE (SERIALE SOFTWARE)

Sintassi: COM2 <expr>

Descrizione istruzione:

Gestisce tutte le operazioni sulla seconda seriale (seriale software gestita con linee di I/O della CPU). 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 generatore di baud rate.

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.

GESTIONE CONVERTITORE A/D

Sintassi: A_D <expr>

Descrizione istruzione:

Restituisce il valore convertito dal canale specificato dell'A/D di bordo.

Descrizione parametri:

<expr> —> Numero del canale da convertire.

GESTIONE INTERRUPT REAL TIME CLOCK

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

Descrizione istruzione:

Tramite questa procedura si abilita l'interrupt dell'RTC, é possibile quindi utilizzare l'istruzione ONEX1 per generare basi temporali o gestire allarmi. Per chiarimenti sulla funzionalità del RTC, fare riferimento alla relativa documentazione.

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 ALARTIMER 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).

LETTURA/SCRITTURA BLOCCO SU EEPROM E RAM RTC SERIALI

Sintassi: BL_EE <expr1>, <expr2>, <expr3>
BL_EE2 <expr1>, <expr2>, <expr3>

Descrizione istruzione:

Effettua la scrittura o la lettura di un blocco di bytes dall'indirizzo specificato, sulla EEPROM seriale o sulla RAM RTC della scheda. 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).

LETTURA/SCRITTURA BYTE SU EEPROM E RAM RTC SERIALI

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

Descrizione istruzione:

Effettua la scrittura o la lettura di un byte all'indirizzo specificato, sulla eeprom seriale o sulla RAM RTC della scheda.

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).

GESTIONE DELLA TASTIERA OPERATORE

Sintassi: KEYB <expr>

Descrizione istruzione:

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

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.

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

INIZIALIZZAZIONE 82C55 PER GESTIONE RIDIREZIONAMENTO CONSOLE

Sintassi: P8255 <expr>

Descrizione istruzione:

Tramite questa istruzione é possibile inizializzare l'82c55 per la gestione del pannello operatore collegato. Si ricorda che é necessario chiamarla una volta sola, prima di utilizzare le istruzioni di gestione del pannello operatore (KEYB, DISPLAY, UO1).

Descrizione parametri:

<expr> —> 0 = PORTB in INPUT.
 —> 1 = PORTB in OUTPUT.

LETTURA/SCRITTURA DI 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 SFR	CODICE SFR per GPC® 552,553,554	NOME SFR	CODICE SFR per GPC® 323D,324D
CTCON	0	DPL1	0
CTH3	1	DPH1	1
CTH2	2	DPS	2
CTH1	3	CKCON	3
CTH0	4	EXIF	4
CMH2	5	SCON1	5
CMH1	6	SBUF1	6
CMH0	7	TA	7
CTL3	8	WDCON	8
CTL2	9	EIE	9
CTL1	10	EIP	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 (di sola lettura)		

Descrizione parametri:

- <expr1> —> 0 = Lettura del SFR.
- > 1 = Scrittura del SFR.
- <expr2> —> Codice identificazione SFR da scrivere o leggere (0÷29).
- <expr3> —> Valore da scrivere (0÷255).

GESTIONE DEL REAL TIME CLOCK

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).

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:

$$\text{Frequenza pwm} = F_{\text{OSC}} / (2 \times (1 + \text{PWM}_F) \times 255)$$

$$\text{Duty cycle pwm} = \text{PWM}_{\text{DT}} / 255 - \text{PWM}_{\text{DT}}$$

dove PWM_F e PWM_{DT} sono valori programmabili ad 8 bit, variabili nel range 0÷255 e F_{OSC} è il valore della frequenza di lavoro della CPU in Hz. Sostituendo il valore minimo (0) e massimo (255) nelle precedenti formule si ottengono i range di variazione per i parametri <expr2> ed <expr3> della procedura.

Descrizione parametri:

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

APPENDICE A: SCHEMI ELETTRICI INTERFACCIE OPERATORE

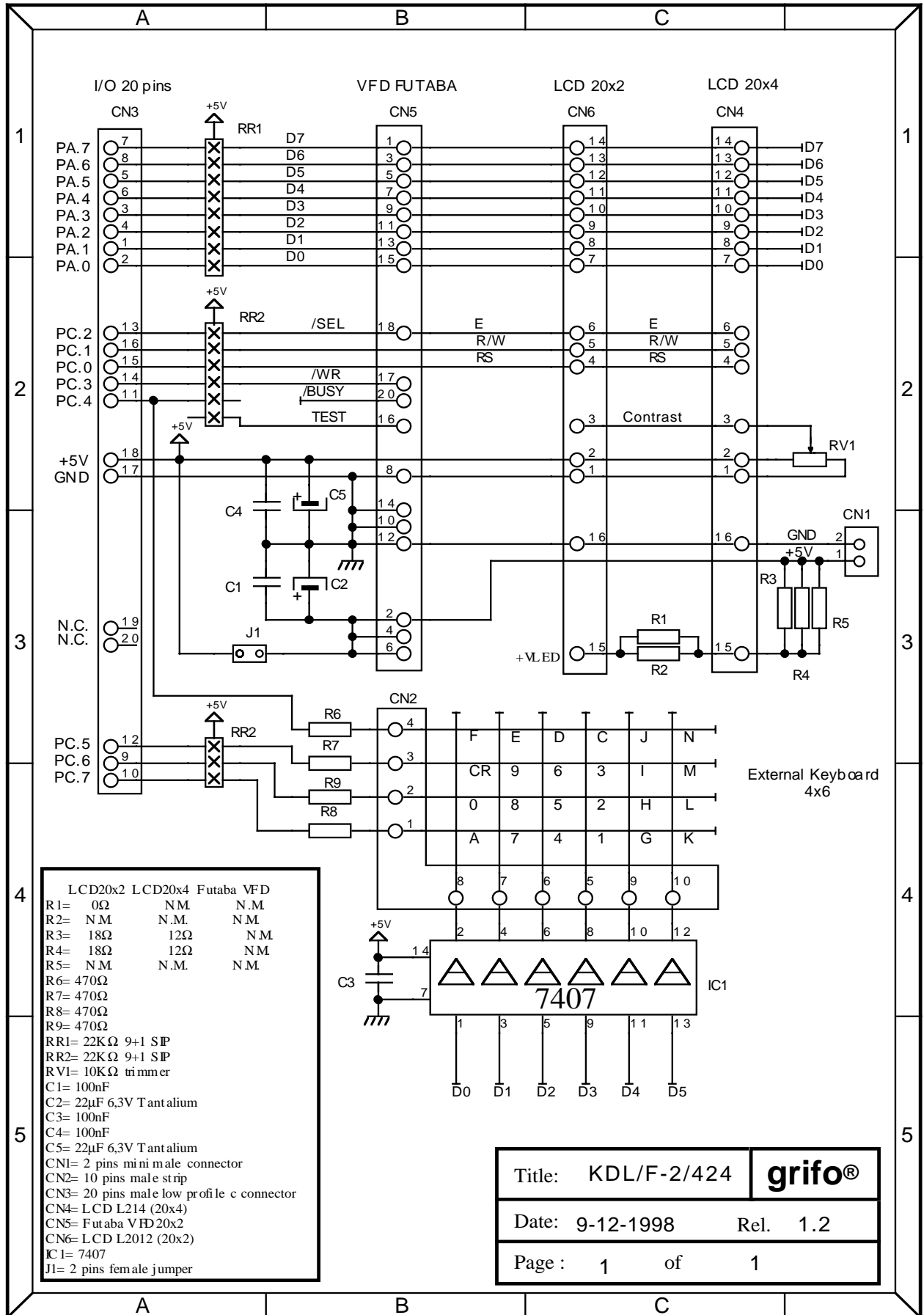


FIGURA A-1: SCHEMA ELETTRICO KDX x24

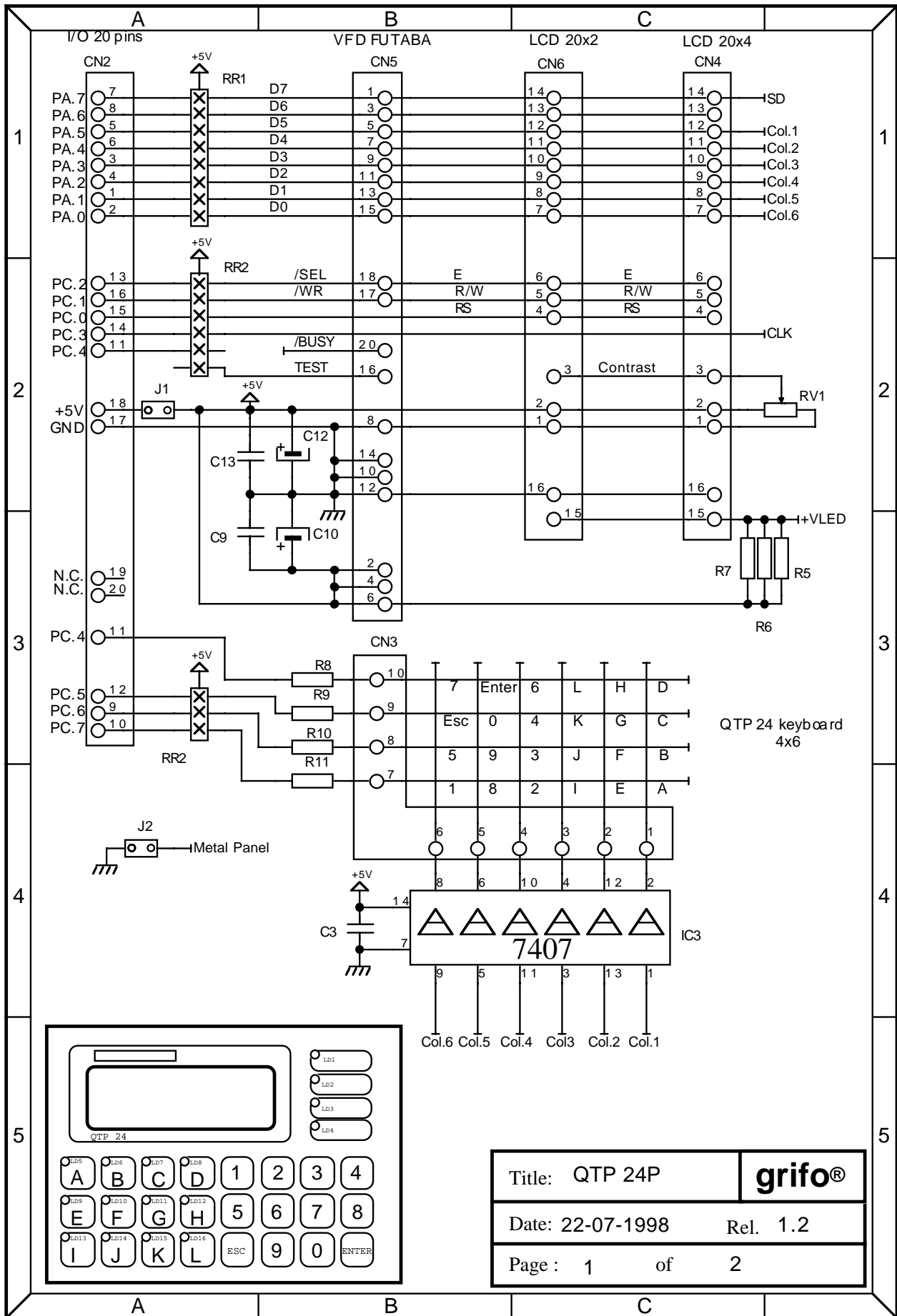
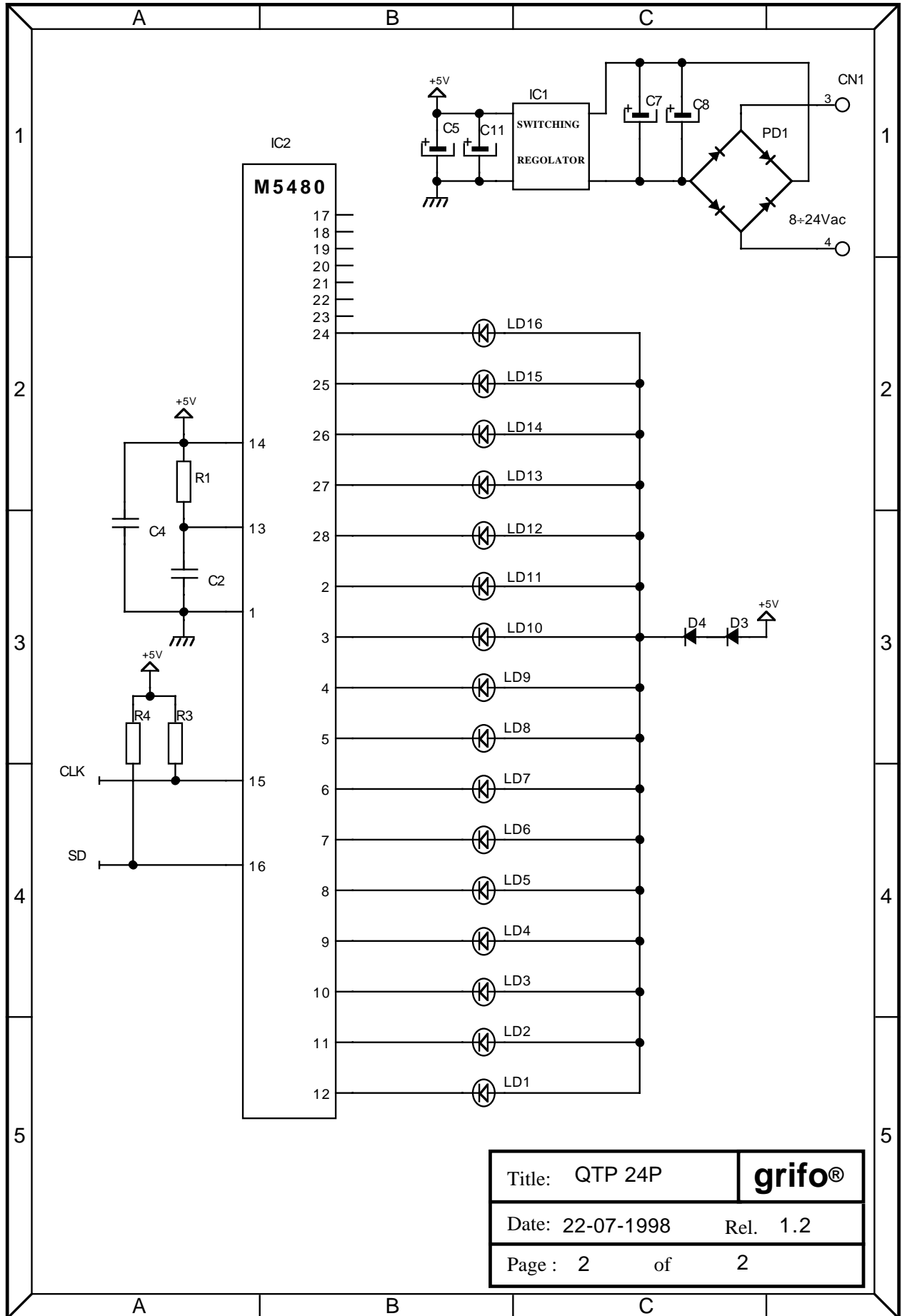


FIGURA A-2: SCHEMA ELETTRICO QTP 24P PARTE 1

Title: QTP 24P	grifo®
Date: 22-07-1998	Rel. 1.2
Page : 1	of 2





Title: QTP 24P	grifo®
Date: 22-07-1998	Rel. 1.2
Page : 2	of 2

FIGURA A-3: SCHEMA ELETTRICO QTP 24P PARTE 2



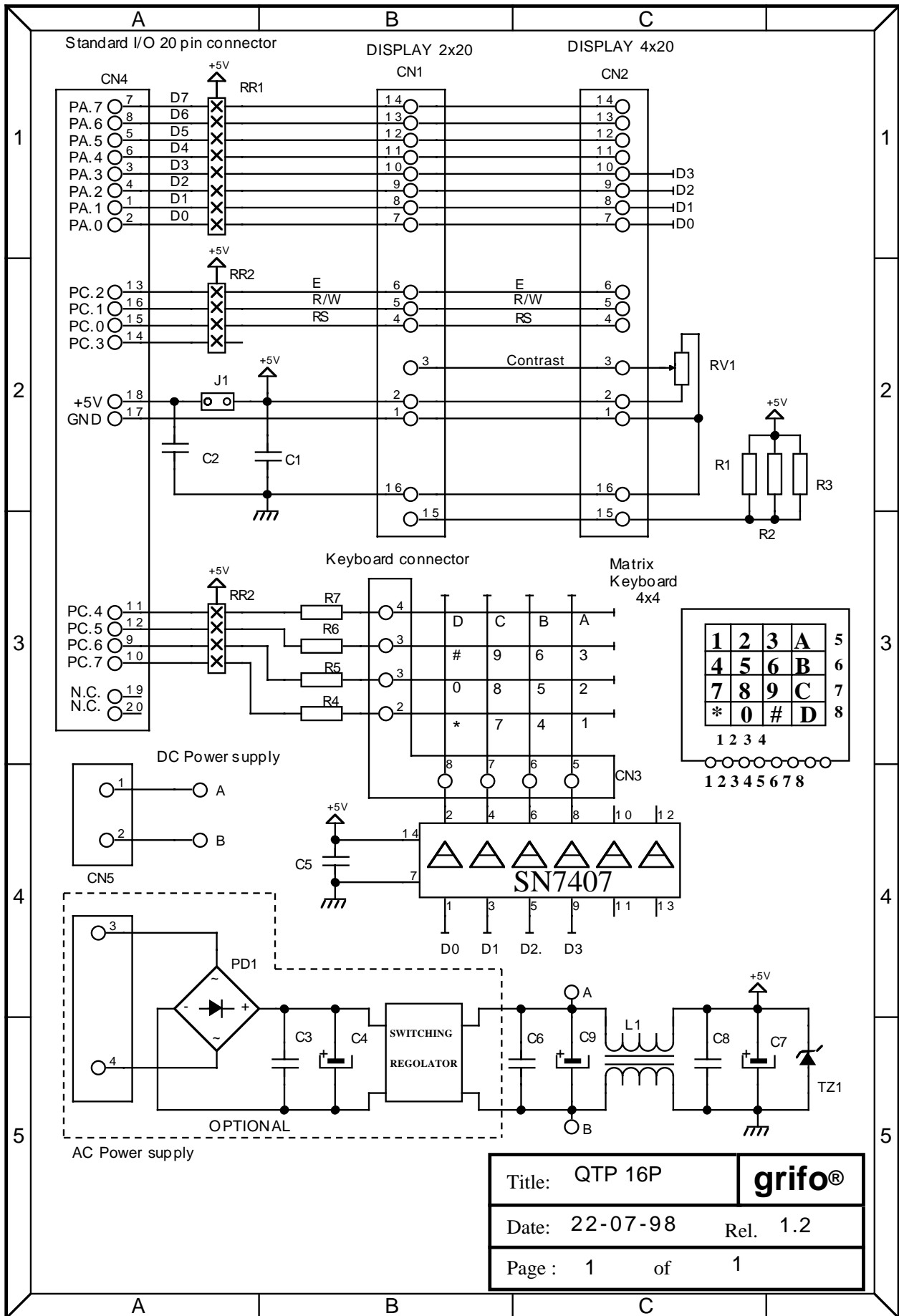
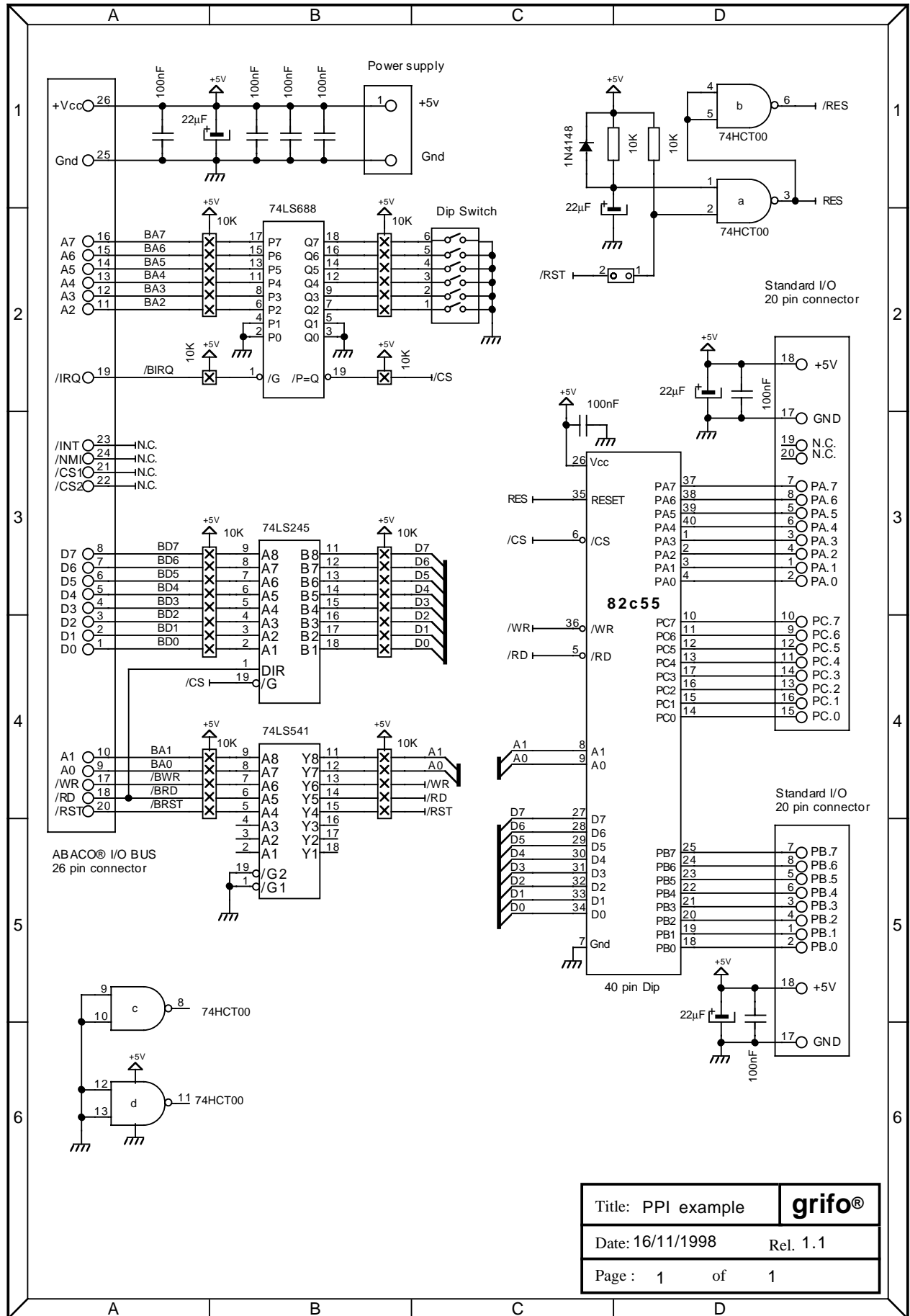


FIGURA A-4: SCHEMA ELETTRICO QTP 16P



Title: QTP 16P	grifo®
Date: 22-07-98	Rel. 1.2
Page : 1	of 1



Title: PPI example	grifo®
Date: 16/11/1998	Rel. 1.1
Page : 1	of 1

FIGURA A-5: SCHEMA ELETTRICO PPI 82c55



