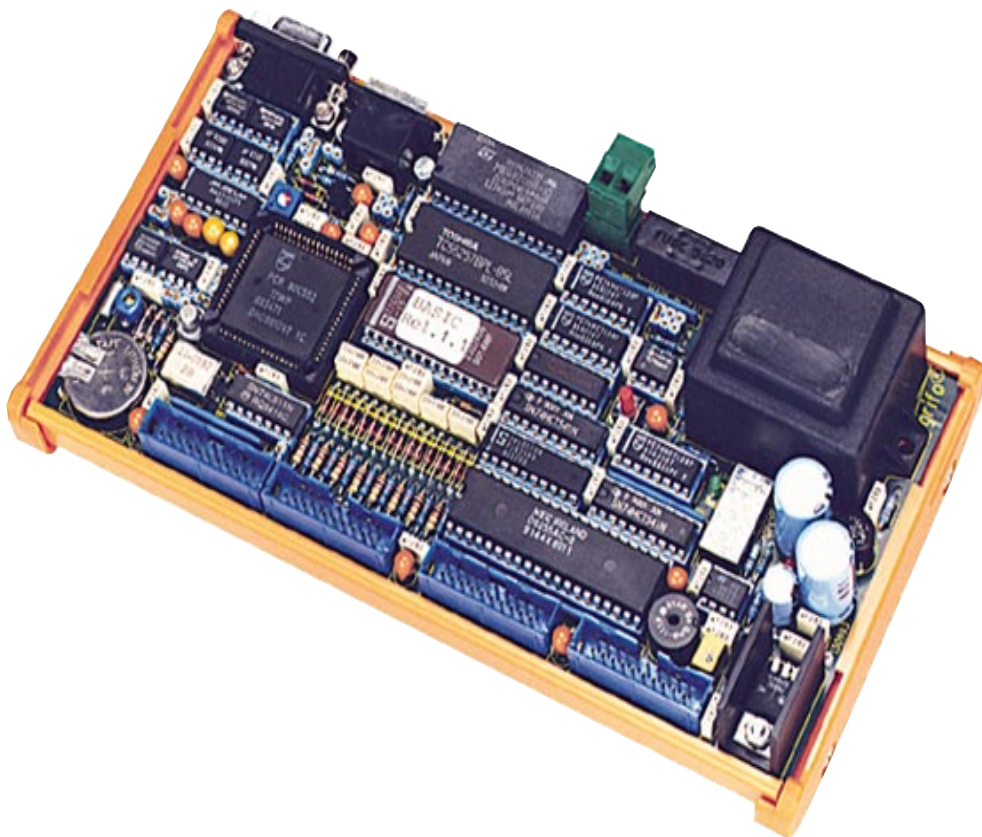


GPC[®] 552

General Purpose Controller 80C552 Philips

MANUALE TECNICO



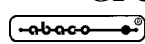
grifo[®]
ITALIAN TECHNOLOGY

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

GPC[®] 552 Edizione 3.30 Rel. 26 Gennaio 2000

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

GPC[®] 552

General Purpose Controller 80C552 Philips

MANUALE TECNICO

CPU 80C552, 87C552, 80C562, 87C562 da 22.1184 o da 29.4912 MHz;
1 linea seriale settabile in RS 232, RS 422-485 o Current Loop;
Funzionamento in IDLE-MODE o POWER-DOWN MODE; 44 I/O
TTL; Timer-Counter da 16 bits con 4 registri di Capture e 3 di
comparazione; 6 uscite Set-Reset legate al comparatore T2, più 2 uscite
di Toggle; Watch Dog; Dip Switch; Buzzer; 1 LED di diagnostica ; 1 LED
di attività; 32K EPROM, 32K RAM/EEPROM, 32K RAM/EEPROM o
EPROM; EEPROM seriale, opzionale, da 512 a 2048 Bytes; Orologio
RTC, opzionale, con batteria al LITIO e 256 Bytes di RAM; 2 linee di
PWM da 8 bits; 8 linee di A/D converter da 10 Bits; I²C.BUS; 3 connettori
standard di I/O **Abaco**[®] da 20 vie; 1 connettore standard di A/D **Abaco**[®]
da 20 vie; Alimentatore da rete a bordo oppure alimentazione a bassa
tensione; Scheda a basso consumo per guide omega DIN 46277-1 e 3.

grifo[®]

ITALIAN TECHNOLOGY

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

E-mail: grifo@grifo.it

<http://www.grifo.it>

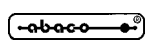
<http://www.grifo.com>

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



GPC[®] 552

Edizione 3.30 Rel. 26 Gennaio 2000

, 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 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:

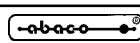


Attenzione: Pericolo generico



Attenzione: Pericolo di alta tensione

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 SCHEDA	1
CARATTERISTICHE GENERALI	2
PROCESSORE DI BORDO	3
DISPOSITIVI DI CLOCK	3
ALIMENTAZIONE DI BORDO	3
COMUNICAZIONE SERIALE	4
DISPOSITIVI DI MEMORIA	4
DISPOSITIVI PERIFERICI DI BORDO	6
LOGICA DI CONTROLLO	7
TASTO DI RESET	7
SPECIFICHE TECNICHE	9
CARATTERISTICHE GENERALI	9
CARATTERISTICHE FISICHE	9
CARATTERISTICHE ELETTRICHE	10
INSTALLAZIONE	11
CONNESSIONI CON IL MONDO ESTERNO	11
CN3 - CONNETTORE DI ALIMENTAZIONE	11
CN1 - CONNETTORE PER I/O: PORT B PPI 82C55, PWM, OUT GENERICI.	12
CN2 - CONNETTORE PER I/O DEL PPI 82C55 PER PORT A E C	13
CN4 - CONNETTORE PER PORT 5 CPU, INGRESSI A/D CONVERTER.	14
CN5 - CONNETTORE A MORSETTIERA PER I2C BUS, SECONDA SERIALE. .	16
CN8 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP	17
CN6 - CONNETTORE PER I/O DEL LA CPU.	18
CN7 - CONNETTORE PER LA COMUNICAZIONE SERIALE.....	20
SEGNALAZIONI VISIVE	22
INTERFACCIAMENTO DEGLI I/O CON IL CAMPO	24
TASTO DI RESET	24
TEST POINT	24
INPUT DI BORDO	24
SELEZIONE TENSIONI DI ALIMENTAZIONE	25
SELEZIONE TIPO INGRESSI ANALOGICI	25
TRIMMERS E TARATURE.....	26
INTERFACCIE OPERATORE LOCALI	26
JUMPERS	27
JUMPERS A 2 VIE	28
JUMPERS A 3 VIE	30
JUMPERS A 4 VIE	31
JUMPERS A 5 VIE	31
NOTE	31
SELEZIONE MODO E ALIMENTAZIONE DELLE RETI RESISTIVE	31
SELEZIONE DEL TIPO DI COMUNICAZIONE SERIALE	32

SELEZIONE MEMORIE	33
BACK UP	33
DESCRIZIONE SOFTWARE	34
DESCRIZIONE HARDWARE	35
INTRODUZIONE	35
MAPPAGGIO DELLE RISORSE DI BORDO	35
MAPPAGGIO DELL'I/O	36
MAPPAGGIO DELLE MEMORIE	36
MAPPAGGIO 1	37
MAPPAGGIO 2	38
MAPPAGGIO 3	39
MAPPAGGIO 4	40
DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO	42
BUZZER	42
USCITE GENERICHE	42
LED DI ATTIVITÀ	42
DIP SWITCH DSW1	43
DIREZIONALITÀ DELLA COMUNICAZIONE IN RS 422-485	43
SELEZIONE PULL UP/DOWN SULLE LINEE DI I/O DELLA SCHEDA	43
EEPROM SERIALE	44
RAM TAMPONATA + RTC SERIALE	44
PPI 82C55	44
PWM	45
UART	45
A/D CONVERTER	45
TIMER/COUNTER	45
WATCH-DOG INTERNO	45
SCHEDE ESTERNE PER GPC® 552	46
BIBLIOGRAFIA	50
APPENDICE A: DISPOSIZIONE JUMPERS	A-1
APPENDICE B: DESCRIZIONE COMPONENTI DI BORDO	B-1
APPENDICE C: INDICE ANALITICO	C-1

INDICE DELLE FIGURE

FIGURA 1: SCHEMA A BLOCCHI	5
FIGURA 2: PIANTA COMPONENTI.....	8
FIGURA 3: CN3 - CONNETTORE DI ALIMENTAZIONE	11
FIGURA 4: CN1 - CONNETTORE PER I/O DEL PORT B DEL PPI 82C55, PWM, OUT GENERICI ...	12
FIGURA 5: CN2 - CONNETTORE PER I/O DEL PPI 82C55 PER PORT A E C	13
FIGURA 6: CN 4- CONNETTORE PER PORT 5 DELLA CPU, INGRESSI A/D CONVERTER	14
FIGURA 7: SCHEMA D' INGRESSO A/D CONVERTER	15
FIGURA 8: CN 5- CONNETTORE A MORSETTIERA PER I2C BUS E SECONDA SERIALE	16
FIGURA 9: CN8 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP	17
FIGURA 10: CN6 - CONNETTORE PER I/O DELLA CPU	18
FIGURA 11: SCHEMA COLLEGAMENTO LINEE DI I/O	19
FIGURA 12: CN7-CONNETTORE PER COMUNICAZIONE SERIALE	20
FIGURA 13: SCHEMA DI COMUNICAZIONE SERIALE	21
FIGURA 14: TABELLA DELLE SEGNALAZIONI VISIVE	22
FIGURA 15: DISPOSIZIONE LEDs, CONNETTORI, DIP-SWITCH, ECC.	23
FIGURA 16: TABELLA RIASSUNTIVA JUMPERS	27
FIGURA 17: TABELLA JUMPERS A 2 VIE	28
FIGURA 18: DISPOSIZIONE JUMPERS	29
FIGURA 19: TABELLA JUMPERS A 3 VIE	30
FIGURA 20: TABELLA JUMPERS A 4 VIE	31
FIGURA 21: TABELLA JUMPERS A 5 VIE	31
FIGURA 22: TABELLA DI SELEZIONE MEMORIE	33
FIGURA 23: TABELLA INDIRIZZAMENTO I/O	36
FIGURA 24: MAPPAGGIO DELLE MEMORIE IN MODO 1	37
FIGURA 25: MAPPAGGIO DELLE MEMORIE IN MODO 2	38
FIGURA 26: MAPPAGGIO DELLE MEMORIE IN MODO 3	39
FIGURA 27: MAPPAGGIO DELLE MEMORIE IN MODO 4	40
FIGURA 28: FOTO DELLA SCHEDA	41
FIGURA 29: SCHEMA DELLE POSSIBILI ESPANSIONI	51
FIGURA A1: DISPOSIZIONE JUMPERS PER COMUNICAZIONE SERIALE	A-1
FIGURA A2: DISPOSIZIONE JUMPERS PER SELEZIONE MEMORIE	A-2



INTRODUZIONE

L'uso di questa scheda é rivolto - **IN VIA ESCLUSIVA** - a personale specializzato.

Scopo di questo manuale é la trasmissione delle informazioni necessarie all'uso competente e sicuro del prodotto. 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 il prodotto 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.

Particolare attenzione deve essere prestata dall'utenza nella fase di installazione ed eventuale manutenzione del modulo, in particolare per quanto riguarda gli accorgimenti relativi alla presenza di una tensione di rete.

Il dispositivo non può essere utilizzato all'aperto. Si deve sempre provvedere ad inserire il modulo 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 col prodotto, é 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, é 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 SCHEDA

Il presente manuale é riferito alla scheda **GPC® 552** versione **180796** e successive. La validità delle informazioni riportate é quindi subordinata al numero di versione della scheda in uso e l'utente deve quindi sempre verificare la giusta corrispondenza tra le due indicazioni. Sulla scheda il numero di versione é riportato in più punti sia a livello di serigrafia che di stampato (ad esempio nell'angolo in alto a destra del lato componenti).

CARATTERISTICHE GENERALI

La **GPC® 552** è un nuovo potente modulo di controllo, che ha un ottimo rapporto prezzo/prestazioni. La sua modularità la rende il componente ideale per poter costruire architetture con logica distribuita, con ottime risorse in termini elaborativi e di I/O verso il campo. La scheda è in grado di risolvere autonomamente molti problemi del settore industriale e, quando necessario, è possibile aumentarne la potenza ed i campi di applicazione, tramite l'**I/O BUS** di bordo. La **GPC® 552** dispone di una serie di connettori normalizzati che facilitano l'operazione di installazione del sistema.

La **GPC® 552** è provvista di un supporto isolante con attacco rapido per guide ad omega del tipo DIN 46277-1 e DIN 46277-3. Dispone di varie soluzioni per l'alimentazione tra cui un alimentatore a tensione di rete. In questo modo è così possibile porre l'elettronica assieme alle strutture elettromeccaniche del quadro elettrico, eliminando tutti i tipici costi di messa in campo del carteggio tradizionale.

Tramite i potenti pacchetti software di cui la scheda può essere dotata, l'utente può sviluppare con rapidità e comodità tutti gli applicativi richiesti anche con linguaggi di programmazione ad alto livello. È inoltre possibile gestire le varie risorse di bordo con istruzioni evolute, eliminando il problema dell'interfacciamento diretto con l'hardware.

- Formato da **100x195** mm per guide **DIN 46277-1** e **DIN 46277-3**;
- CPU **80C552, 87C552, 80C562, 87C562**, con quarzo da **22.1184** a **29.4912 MHz**, codice compatibile **8051 INTEL**;
- Indirizzamento massimo dell' area di bordo, **64KBytes**;
- Zoccoli per **32K EPROM, 32K RAM o EEPROM, 32K RAM, EEPROM o EPROM**;
- **EEPROM** seriale opzionale da **512** a **2048 Bytes**;
- Orologio **RTC** opzionale con **256 Bytes** di **RAM**, con batteria al Litio
- Massima **EEPROM** seriale in abbinamento ad **RTC, 1024 Bytes**.
- **8** linee di **A/D** converter da **10 Bits**, +2,49 V fs o +5 V fs, tempo di conversione **50µs** a **12MHz**;
- Connettore standard per **I²C bus**;
- **44** linee **TTL** di **I/O**, settabili da software;
- **2** linee indipendenti di **PWM** da **8 bits**
- **1** dip-switch da **8** vie, leggibile da software;
- Circuiteria con **BUZZER** attivo a bordo scheda;
- **Watch-Dog** settabile da software;
- **Timer Counter** da **16 bits** con **4** registri di **Capture** e **3** di **Comparazione**;
- **6** uscite **Set-Reset** legate al comparatore **T2**, più **2** uscite di **Toggle**.
- Registro di **Timer-Counter** standard da **16 bits**
- **1** linea seriale settabile in **RS232, RS422, RS485** o **Current-Loop**;
- **1** linea seriale software settabile in **RS232**;
- **1** LED di diagnostica;
- **1** LED di attività gestibile da software;
- **3** connettori standard di **I/O ABACO®**, da **20** vie;
- **1** connettore standard di **A/D ABACO®**
- Possibilità di funzionamento in **Idle-Mode** o **Power-Down Mode**
- Alimentatore da rete incorporato oppure alimentazione a bassa tensione
- Vasta disponibilità di software di sviluppo quali **Monitor, Debugger, Assemblatori, BASIC Interpretato, BASIC Compiler, FORTH, Compilatore C, Compilatore PASCAL**, ecc.

PROCESSORE DI BORDO

La scheda **GPC® 552** é predisposta per accettare i processori: 80C552, 87C552, 80C562, 87C562. Tali processori ad 8 bit sono codice compatibile con l' 8051 della INTEL e sono quindi caratterizzati da un esteso set di istruzioni, da un'alta velocità di esecuzione e di manipolazione dati e da un'efficiente gestione vettorizzata degli interrupts.

Di seguito viene riportato un elenco di tutte le caratteristiche principali delle varie CPU che tale scheda é in grado di montare:

- 80C552:
- 8k bytes EPROM, 256 bytes RAM
 - 6 ports di I/O ad 8 bits;
 - 2 Timer/Counters da 16 bits
 - 1 Timer/Counters da 16 bits con funzioni di Capture e Compare;
 - 2 livelli di priorità per gli Interrupt;
 - 8 linee di A/D converter da 10 bits;
 - 2 linee indipendenti di PWM da 8 bits;
 - 1 linea seriale UART;
 - 1 linea per I²C bus;
 - Watch Dog Timer;
 - Funzionamento in IDLE-MODE o POWER-DOWN MODE;

Per maggiori informazioni a riguardo di questo componente si faccia riferimento all'apposita documentazione della casa costruttrice.

DISPOSITIVI DI CLOCK

Sulla **GPC® 552** sono presenti due circuiti separati che provvedono a generare rispettivamente la frequenza di clock per la CPU (da 22.1184 a 29.4912 MHz) e la frequenza per REAL TIME CLOCK di IC 25 (32768 Hz). La scelta di utilizzare due circuiti e quindi due quarzi indipendenti, é legata alla possibilità di poter variare in modo indipendente la frequenza di lavoro della CPU.

ALIMENTAZIONE DI BORDO



Una delle caratteristiche peculiari della **GPC® 552** é quella di essere provvista di una sezione alimentatrice a bordo scheda che provvede a generare l'unica tensione di alimentazione necessaria di +5 Vdc. Sono disponibili quattro diverse sezioni alimentatrici: quella da rete che richiede una tensione di 230 Vac, quella lineare che richiede una tensione 6÷10 Vac (+12 Vdc), quella switching che necessita di una tensione 8÷26 Vac e senza sezione alimentatrice che richiede una tensione di +5 Vdc (per maggiori informazioni vedere apposito paragrafo "TENSIONI DI ALIMENTAZIONE"). La tensione di alimentazione può essere fornita tramite un apposito connettore standardizzato di facile ed immediata installazione. Sulla scheda sono state adottate tutte le scelte circuitali e componentistiche che tendono a ridurre i consumi, compresa la possibilità di far lavorare il microcontrollore in power down ed idle mode.

Il tipo di alimentazione della scheda non può essere variato dall'utente e deve quindi essere specificato in fase di ordine.

COMUNICAZIONE SERIALE

La comunicazione seriale per quanto riguarda la SIO 0 é completamente settabile via software per quanto riguarda sia il protocollo sia la velocità di comunicazione.

Tali settaggi avvengono tramite la programmazione dei registri della SIO 0 interni all' 80C552 di cui la scheda é provvista, quindi per ulteriori informazioni si faccia riferimento alla documentazione tecnica della casa costruttrice o all'appendice B di questo manuale.

Dal punto di vista hardware é invece possibile selezionare, tramite una serie di comodi jumpers, il protocollo fisico di comunicazione. In particolare la linea per la SIO 0 può essere bufferata in RS 232, Current Loop o RS 422-485; in quest'ultimo caso é definibile anche se la comunicazione avviene in Full Duplex o Half Duplex.

Tale CPU ha anche una seconda seriale denominata SIO 1 utilizzata per la comunicazione con dispositivi che supportano la comunicazione con il protocollo fisico I²C bus.

Per ulteriori informazioni sulla SIO 1 si faccia riferimento alla documentazione tecnica della casa costruttrice o all'appendice B di questo manuale.

DISPOSITIVI DI MEMORIA

E' possibile dotare la scheda di un massimo di 98K e 256 bytes di memoria variamente suddivisi con un massimo di 32K EPROM, 32K RAM/EEPROM, 32K RAM/EEPROM/EPROM, 256 bytes di RAM+RTC seriale ed infine 2K EEPROM seriale. La scelta della configurazione delle memorie presenti sulla scheda può avvenire in relazione all'applicazione da risolvere e quindi in relazione alle esigenze dell'utente. Da questo punto di vista si ricorda che la scheda viene normalmente fornita con 32K RAM di lavoro e che tutte le rimanenti memorie devono essere quindi opportunamente specificate in fase di ordine della scheda.

Tramite la circuiteria di back up presente a bordo scheda c'è inoltre la possibilità di tamponare fino ad un massimo di 32K e 256 bytes di RAM aggiungendo quindi la possibilità 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. La circuiteria di back up é basata su una batteria al Litio presente a bordo scheda e da una batteria esterna collegabile tramite un apposito connettore. Qualora la quantità di RAM tamponata risulti insufficiente (ad esempio per sistemi di data loghin) si possono sempre utilizzare i moduli di RAM tamponata e/ o di EEPROM.

Il modulo di RAM da montare su IC 25 può essere inoltre provvisto di un real time clock interno in grado di gestire via software l'orario (ore, minuti, secondi) e la data (giorno, mese, anno, giorno della settimana).

Il mappaggio delle risorse di memoria avviene tramite una opportuna circuiteria di bordo, che provvede ad allocare i dispositivi all'interno dello spazio d'indirizzamento del microprocessore; tale logica di controllo provvede a gestire in modo completamente automatico diversi tipi di mappaggi che si adattano ai diversi pacchetti software disponibili per la **GPC® 552**.

Per maggiori informazioni fare riferimento al capitolo "DESCRIZIONE HARDWARE" e "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO". Per una descrizione più approfondita sui dispositivi di memoria, sugli zoccoli da utilizzare e sullo strappaggio della scheda, fare riferimento al paragrafo "SELEZIONE MEMORIE".

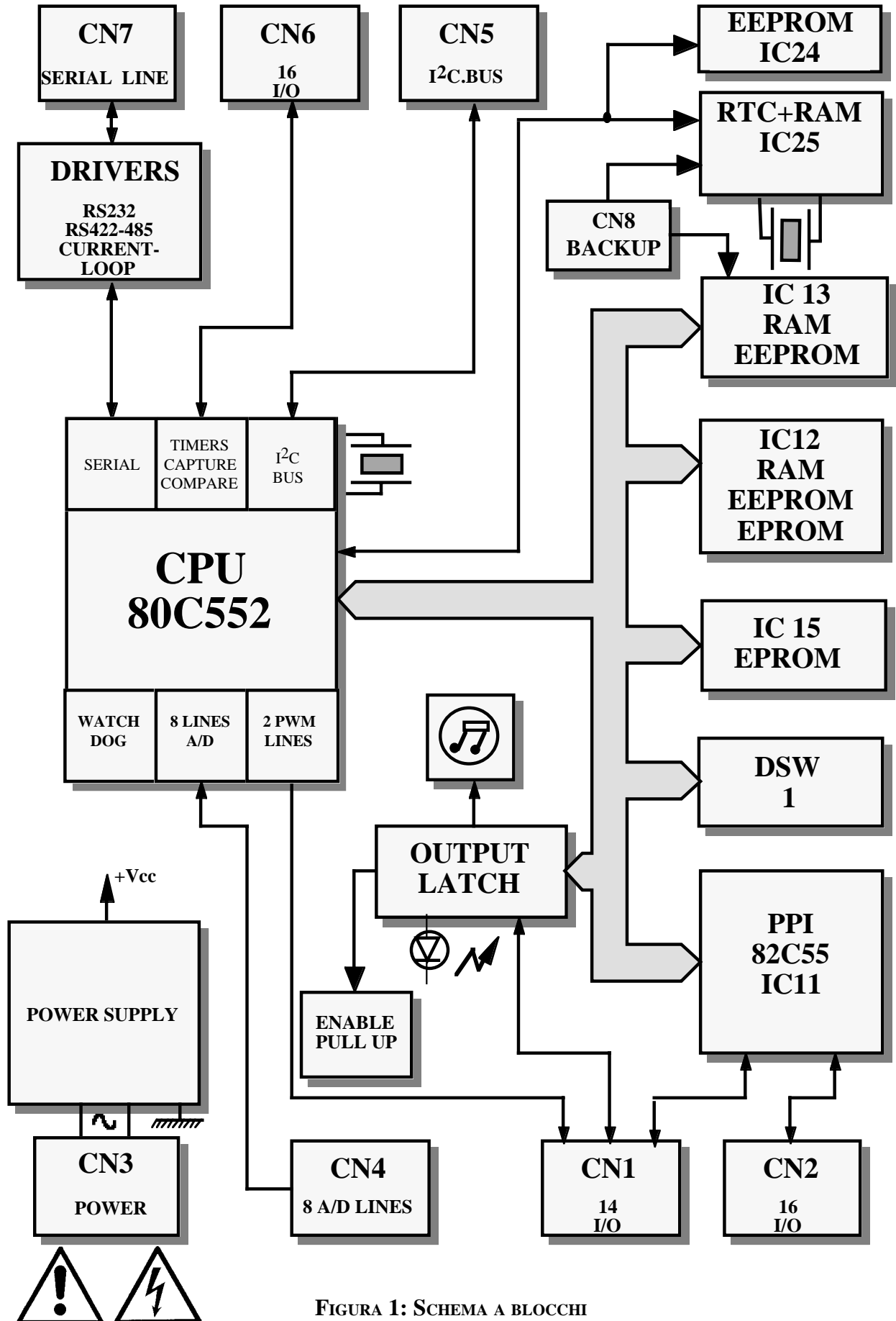


FIGURA 1: SCHEMA A BLOCCHI

DISPOSITIVI PERIFERICI DI BORDO

La scheda **GPC® 552**, nata per risolvere molteplici problemi di controllo e comando di automatismi, é dotata di alcuni componenti periferici che si occupano dell'interfacciamento con il mondo esterno. In particolare:

- **SIO 0**: periferica interna alla CPU in grado di gestire una linea per la comunicazione seriale. Il dispositivo può essere utilizzato per la comunicazione con tutti i sistemi provvisti di una linea seriale bufferata in RS 232 o RS 422-485 o Current Loop. Dal punto di vista software é infatti definibile la velocità di comunicazione, la lunghezza della parola, il numero di stop bit, la parità e lo stato dei segnali di handshake hardware. Il tutto avviene tramite una semplice programmazione di registri interni alla CPU.
- **SIO 1**: periferica interna in grado di gestire la comunicazione seriale con il protocollo fisico I²C bus.
- **A/D converter**: periferica interna alla CPU in grado di acquisire 8 canali con una risoluzione massima di 10 bits. Dal punto di vista software é possibile definire quali canali attivare, dare lo start o lo stop all' acquisizione ecc.
- **PWM**: sono disponibili due linee indipendenti di PWM. Dal punto di vista software é definibile la frequenza e il duty-cycle del segnale.
- **WATCH-DOG**: periferica interna alla CPU che permette il reset della medesima ad intervalli di tempo prefissati se non viene effettuato un retrigger. Dal punto di vista software é definibile il tempo di intervento, l' abilitazione e il retrigger.
- **Configurazione scheda**: allo scopo di rendere configurabile la scheda ed in particolare il programma applicativo sviluppato per questa, é stato previsto un dip switch ad 8 vie. Tale dip_switch ha una duplice funzione di configurazione: selezionare la modalità RUN/DEBUG e configurare il software di controllo. La possibilità di acquisire via software lo stato di parte del dip_switch, fornisce all'utente la possibilità di gestire diverse condizioni tramite un unico programma, senza dover rinunciare ad altre linee d'ingresso (le applicazioni caratteristiche sono: selezione della lingua di rappresentazione, definizione parametri del programma, selezione delle modalità operative, ecc.). Sempre in merito alla configurazione della scheda, sulla **GPC® 552** é stato previsto un LED di attività, gestito via software con cui l'utente può segnalare visivamente lo stato di tutto il sistema.
- **Real Time Clock**: il modulo di RAM da montare sullo zoccolo IC 25 può essere provvisto di un completo Real Time Clock in grado di gestire ore, minuti, secondi, giorno del mese, mese, anno e giorno della settimana in modo completamente autonomo. Il componente é opzionale e viene fornito con batteria di Back-Up.
- **EEPROM seriale**: il modulo di EEPROM seriale da montare sullo zoccolo IC 24 é necessario quando si devono mantenere delle informazioni anche in assenza di alimentazione, senza ricorrere al back up della RAM, con una sicurezza estrema sulla validità dei dati. Tale modulo può avere un size variabile nel range 512÷2048 bytes, é opzionale e deve essere quindi esplicitamente ordinato.
- **BUZZER**: sulla **GPC® 552** è presente una circuiteria in grado di emettere un suono costante, basata su un buzzer capacitivo. Questa circuiteria viene abilitata e/o disabilitata via software tramite la logica di controllo e può essere utilizzata per generare allarmi acustici, feed back sonori, ecc.

- **PPI 82C55**: periferica in grado di gestire tre port paralleli da 8 bit per un totale di 24 linee di I/O logico a livello TTL, con direzionalità settabile a livello di byte.
Il chip PPI 82C55 viene completamente gestito via software tramite la programmazione di 4 registri situati nello spazio di I/O della CPU da un'apposita logica di controllo.

Per ulteriori informazioni a riguardo dei dispositivi periferici descritti, si faccia riferimento alla documentazione tecnica della casa costruttrice o all'appendice B di questo manuale.

LOGICA DI CONTROLLO

Il mappaggio di tutti i registri delle periferiche presenti sulla scheda e dei dispositivi di memoria, é affidata ad un'opportuna logica di controllo che si occupa di allocare tali dispositivi nello spazio d'indirizzamento della CPU. Per maggiori informazioni fare riferimento al paragrafo "MAPPAGGIO DELL'I/O".

TASTO DI RESET

Sulla **GPC® 552** é presente un comodo pulsante di reset che una volta premuto fà ripartire la scheda da una condizione di azzeramento generale. La funzione principale di questo tasto é quella di uscire da condizioni di loop infinito, soprattutto durante la fase di debug o di garantire uno stato certo di partenza.

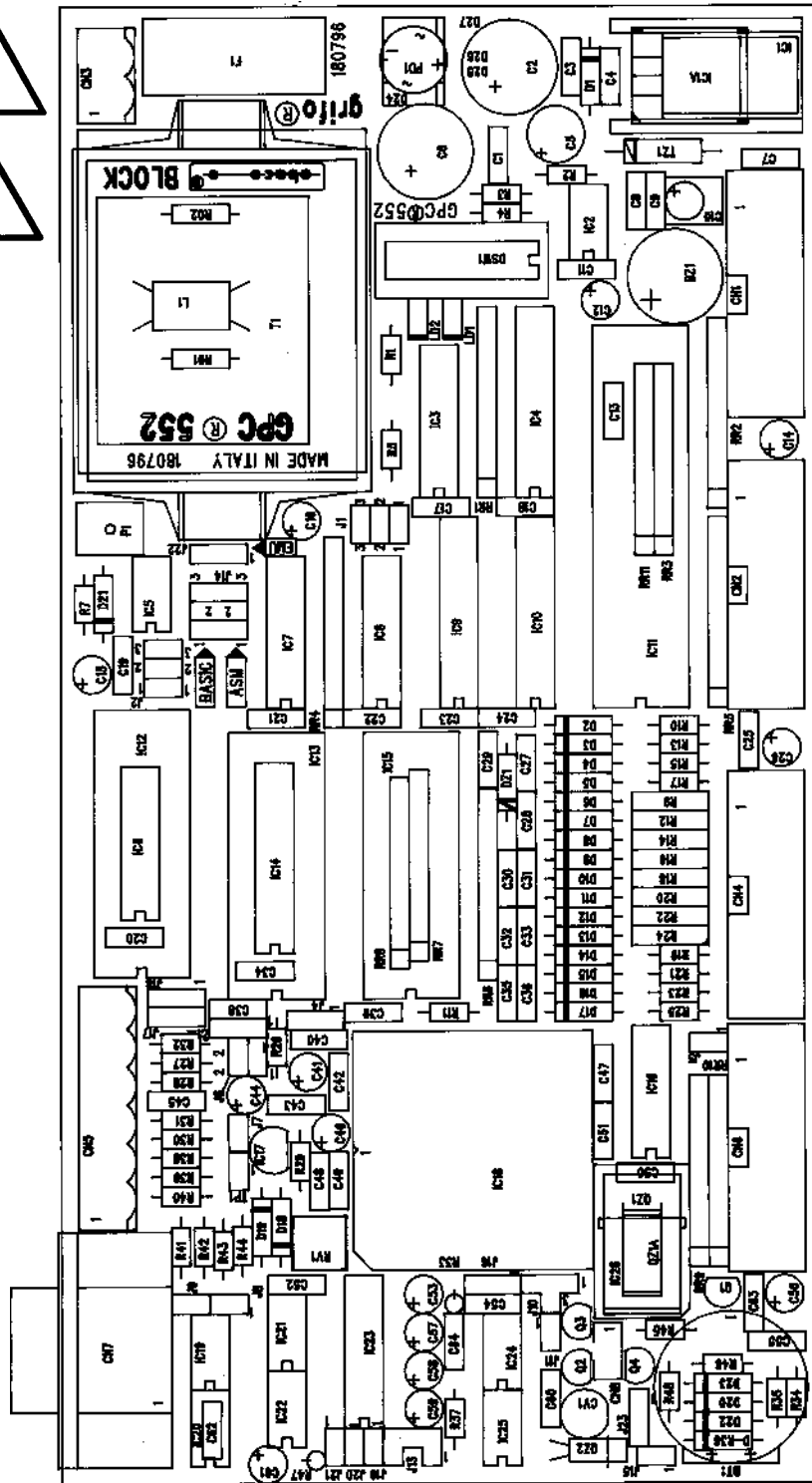


FIGURA 2: PIANTA COMPONENTI



SPECIFICHE TECNICHE


CARATTERISTICHE GENERALI

Risorse di bordo	44 Input/Output programmabili TTL 3 Timer Counter a 16 bit 1 Linea bidirezionale RS 232 o RS 422-485 o Current Loop 1 Linea per I ² C bus 1 Linea bidirezionale RS 232 (seriale software) 1 Watch Dog 8 Linee di A/D converter 1 Tasto locale di reset 1 Real Time Clock 1 Buzzer 2 LEDs 1 Dip switch per un totale di 8 dip 2 Linee di PWM da 8 bits
Memoria indirizzabile	IC 15:EPROM da 32K x 8 IC 13:RAM/EEPROM da 8K x 8 a 32K x 8 IC 12:RAM/EEPROM/EPROM da 8K x 8 a 32K x 8 IC 24:EEPROM seriale da 512 bytes a 2048 bytes IC 25:RAM+RTC seriale da 256 bytes
CPU di bordo	PHILIPS 80C552 da 22.1184 a 29.4912 MHz
Risoluzione A/D	10 bits
Tempo conversione A/D	27.126 µs a 22.1184 MHz 20.345 µs a 29.4912 MHz

CARATTERISTICHE FISICHE

Dimensioni stampato	Formato 100 x 195 mm
Peso	570 g.
Connettori	CN1: 20 vie scatolino verticale M CN2: 20 vie scatolino verticale M CN3: 2 vie rapida estrazione CN4: 20 vie scatolino verticale M CN5: 6 vie morsettiera CN6: 20 vie scatolino verticale M CN7: Vaschetta D femmina 9 vie CN8: 2 vie scatolino verticale M
Range di temperatura	da 0 a 70 gradi Centigradi
Umidità relativa	20% fino a 90% (senza condensa)

CARATTERISTICHE ELETTRICHE

Fusibile F2	100 mA; 250 V di tipo rapido	
Tensione di alimentazione	+5 Vdc	(senza sezione alimentatrice)
	230 Vac; 50 Hz	(sezione alimentatrice da rete)
	6÷10 Vac	(sez. alimentatrice a bassa tensione lineare) *
	8÷26 Vac	(sez. alimentatrice a bassa tensione switching)
Corrente assorbita sui 5 Vdc	100÷140 mA	
Corrente fornita sui +5 Vdc per carichi esterni	450÷500 mA	(rete) *
	300÷260 mA	(switching) *
	900÷860 mA	(lineare) *
Batteria esterna di back up	3,6÷5 Vdc	15 µA corrente di back up
Ingressi analogici in tensione	0÷2,49 Vcc o 0÷5 Vcc	
Ingressi analogici in corrente	0÷20 mA	

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

INSTALLAZIONE

In questo capitolo saranno illustrate tutte le operazioni da effettuare per il corretto utilizzo della scheda. A questo scopo viene riportata l'ubicazione e la funzione degli strips, dei connettori e dei LEDs presenti sulla **GPC® 552**.

CONNESSIONI CON IL MONDO ESTERNO

Il modulo **GPC®552** è provvisto di 8 connettori con cui vengono effettuate tutte le connessioni con il campo e con le altre schede 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 15, 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.

CN3 - CONNETTORE DI ALIMENTAZIONE

CN3 é un connettore a morsetti per rapida estrazione, composto da 2 contatti. Tramite CN3 deve essere fornita alla scheda la tensione di alimentazione selezionata.

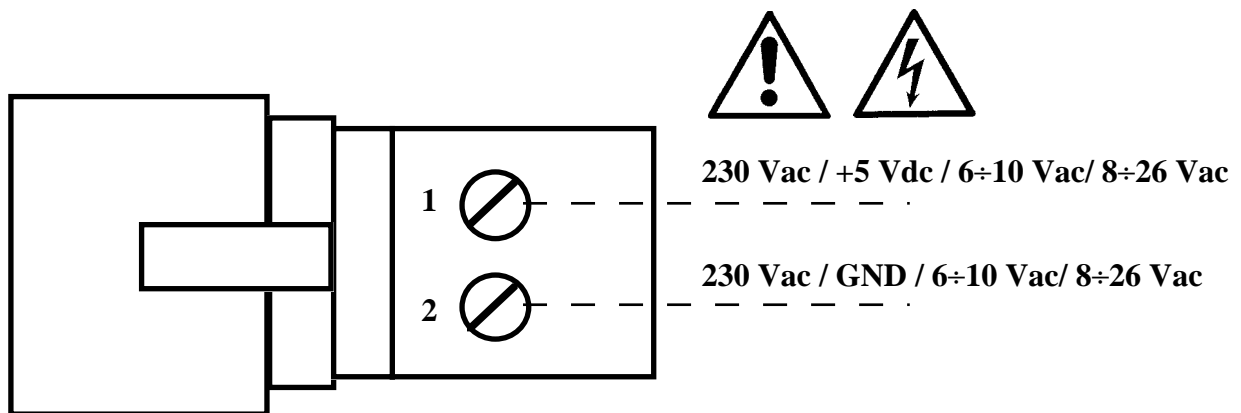


FIGURA 3: CN3 - CONNETTORE DI ALIMENTAZIONE

Legenda:

230 Vac / +5 Vdc / 6÷10 Vac / 8÷26 Vac =

I - Linee per alimentazione da rete 230 Vac.

I - Linee per alimentazione +5 Vdc.

I - Linee per l'alimentazione 6÷10 Vac e +12 Vdc.

I - Linee per l'alimentazione 8÷26 Vac.

GND



=

- Linea di massa.

CN1 - CONNETTORE PER I/O: PORT B PPI 82C55, PWM, OUT GENERICI.

CN1 è un connettore a scatolino verticale con passo 2.54 mm a 20 piedini. Tramite CN1 si effettua la connessione tra il port B del PPI 82C55 e l'ambiente esterno.

Su tale connettore inoltre si possono trovare due uscite PWM e 4 uscite TTL generiche.

I segnali presenti su questo connettore coincidono con segnali logici a livello TTL.

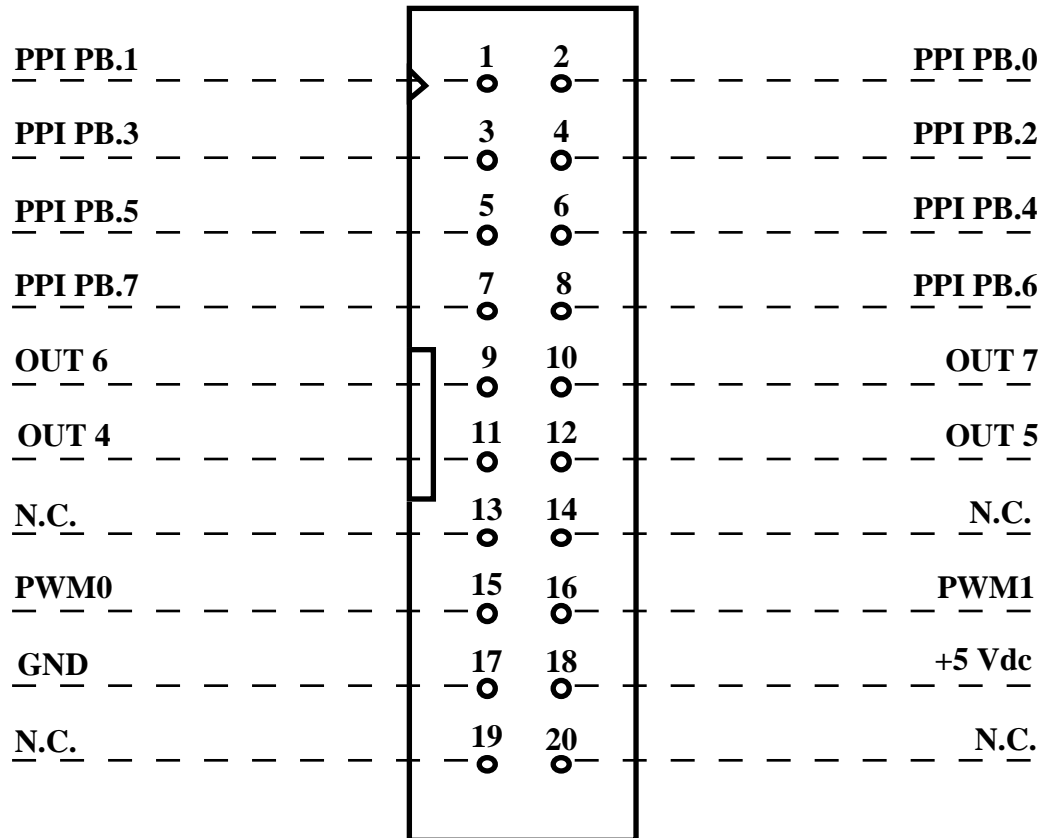


FIGURA 4: CN1 - CONNETTORE PER I/O DEL PORT B DEL PPI 82C55, PWM, OUT GENERICI

Legenda:

- PPI PB.n** = I/O - Linea digitale n del port B del PPI 82C55.
PWM0 = O - Linea di PWM n. 0 della CPU.
PWM1 = O - Linea di PWM n.1 della CPU.
GND = - Linea di massa.
+5 Vdc = - Linea di alimentazione a +5 Vdc.
N.C. = - Non Collegato.
OUT n = O - Linea digitale generica n.

CN2 - CONNETTORE PER I/O DEL PPI 82C55 PER PORT A E C

CN2 è un connettore a scatolino verticale con passo 2.54 mm a 20 piedini. Tramite CN2 si effettua la connessione tra l'interfaccia periferica programmabile PPI 82C55 e l'ambiente esterno, utilizzando due dei tre port paralleli ad 8 bit. I segnali presenti su questo connettore coincidono con segnali logici a livello TTL.

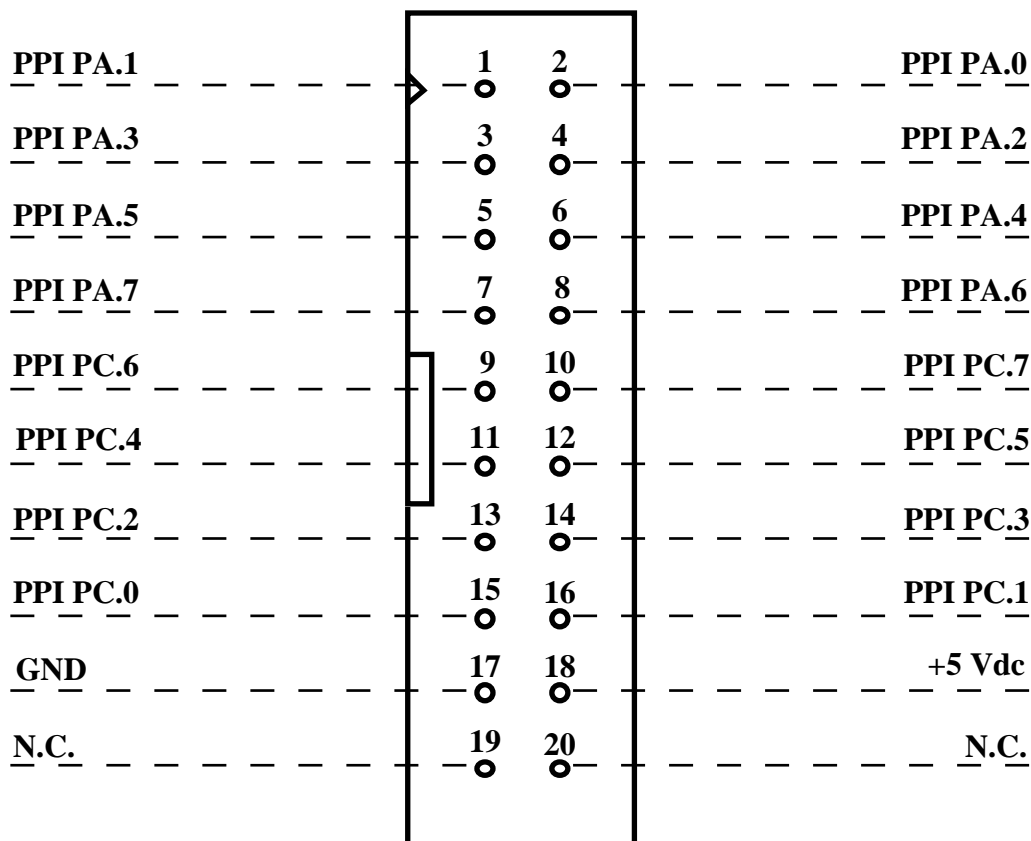


FIGURA 5: CN2 - CONNETTORE PER I/O DEL PPI 82C55 PER PORT A E C

Legenda:

- PPI PA.n** = I/O - Linea digitale n del port A del PPI 82C55.
- PPI PC.n** = I/O - Linea digitale n del port C del PPI 82C55.
- GND** = - Linea di massa.
- +5 Vdc** = - Linea di alimentazione a +5 Vdc.
- N.C.** = - Non Collegato.

Di particolare interesse é la possibilità di collegare direttamente al connettore CN2 una serie di interfacce con cui risolvere molti problemi caratteristici del settore industriale. Per i moduli di interfaccia operatore locale (**QTP 24P**, **KDL x24**, **KDF 224**, ecc.), oltre alla facilità d'installazione si ricorda anche la facilità di gestione software, supportata da linguaggi di programmazione ad alto livello. Per maggiori informazioni si faccia riferimento al paragrafo "INTERFACCIE OPERATORE LOCALI".

CN4 - CONNETTORE PER PORT 5 CPU, INGRESSI A/D CONVERTER.

CN4 è un connettore a scatolino verticale con passo 2.54 mm a 20 piedini. Tramite CN4 si interfacciano le 8 linee di input per la sezione di A/D converter della CPU con il campo esterno. Da ricordare che il port 5 della CPU ha una doppia funzione ossia le 8 linee possono essere o ingressi per l' A/D converter.

Tutti i segnali descritti sono riportati sul connettore CN4 tramite un filtro passa basso descritto in figura 7.

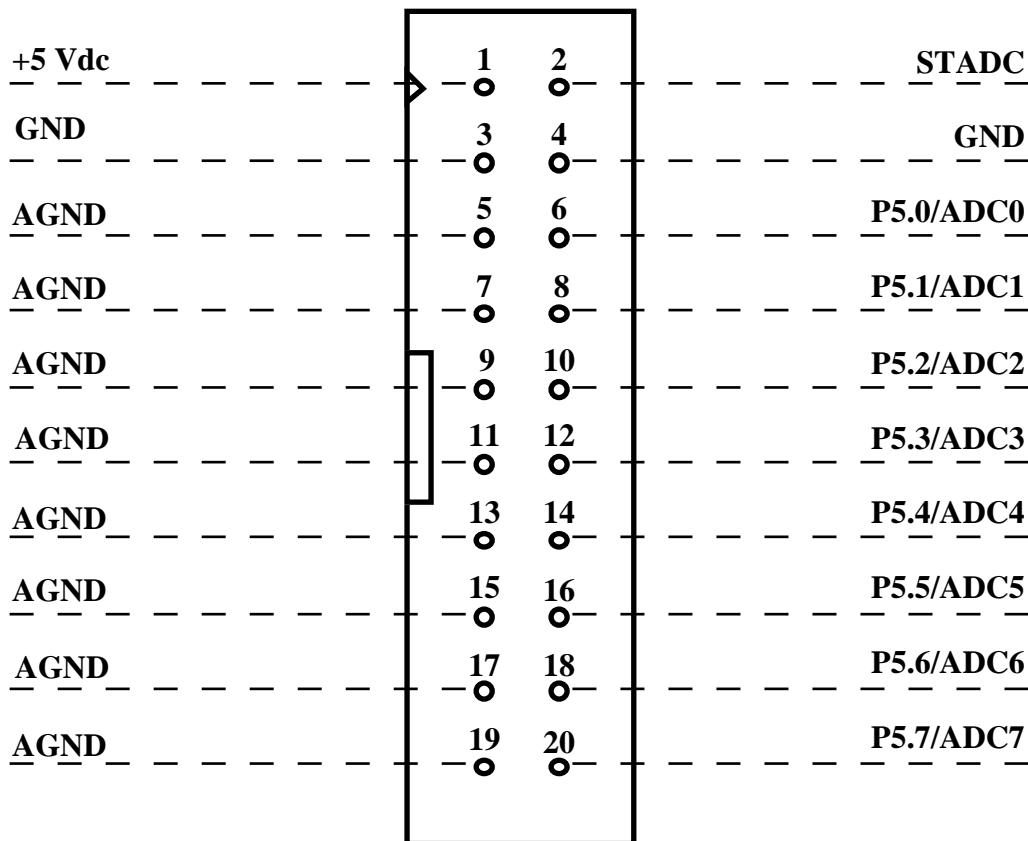


FIGURA 6: CN 4- CONNETTORE PER PORT 5 DELLA CPU, INGRESSI A/D CONVERTER

Legenda:

- STADC** = I - Linea digitale di start per la conversione analogica/digitale.
P5.n/ADCn = I - Linea digitale n o ingresso canale n dell' A/D della CPU.
GND = - Linea di massa digitale.
AGND = - Linea di massa analogica.
+5 Vdc = - Linea di alimentazione a +5 Vdc.

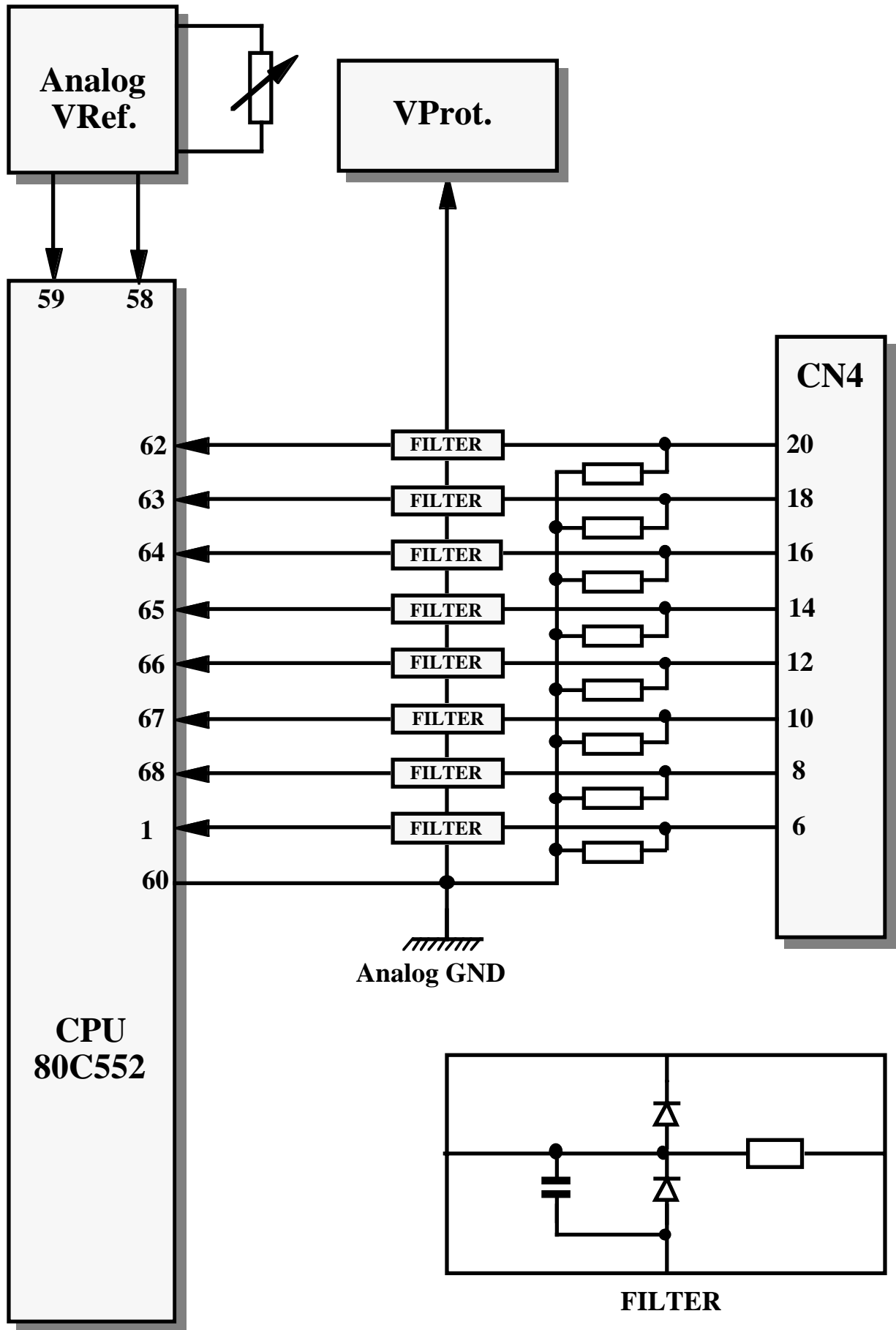


FIGURA 7: SCHEMA D' INGRESSO A/D CONVERTER

CN5 - CONNETTORE A MORSETTIERA PER I²C BUS, SECONDA SERIALE.

CN5 é un connettore a morsettiere composto da 6 pins.

Tramite CN5 é possibile comunicare con qualsiasi dispositivo in grado di supportare la comunicazione tramite il protocollo standard I²C bus, inoltre é presente il collegamento per la seconda seriale software in RS 232.

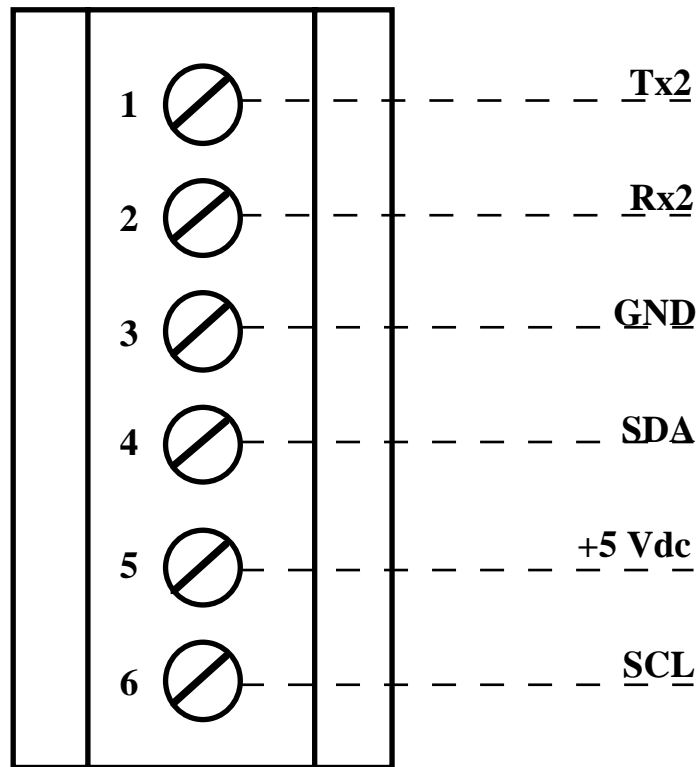


FIGURA 8: CN 5- CONNETTORE A MORSETTIERA PER I²C BUS E SECONDA SERIALE

Legenda:

Tx2	= O	- Linea di trasmissione della seconda seriale software in RS 232.
Rx2	= I	- Linea di ricezione della seconda seriale software in RS 232.
SDA	= I/O	- Linea DATA seriale per I ² C bus.
SCL	= I/O	- Linea CLOCK seriale per I ² C bus.
GND	=	- Linea di massa.
+5 Vdc	=	- Linea di alimentazione a +5 Vdc.

CN8 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP

CN8 é un connettore a scatolino, verticale, maschio, con passo 2,54 mm a 2 vie. Tramite CN8 deve essere collegata una batteria esterna che provvede a mantenere i dati delle RAM di bordo ed a garantire il funzionamento dell'eventuale real time clock, anche in assenza di tensione di alimentazione (per maggiori informazioni fare riferimento al paragrafo "BACK UP").

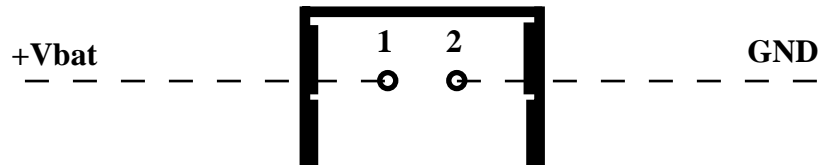


FIGURA 9: CN8 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP

Legenda:

+Vbat	=	I	-	Positivo della batteria esterna di back up
GND	=	-	-	Negativo della batteria esterna di back up

CN6 - CONNETTORE PER I/O DEL LA CPU.

CN6 è un connettore a scatolino verticale con passo 2.54 mm a 20 piedini. Tramite CN6 si effettua la connessione tra i port 1, 3 e 4 della CPU e l'ambiente esterno.

I segnali presenti su questo connettore coincidono con segnali logici a livello TTL.

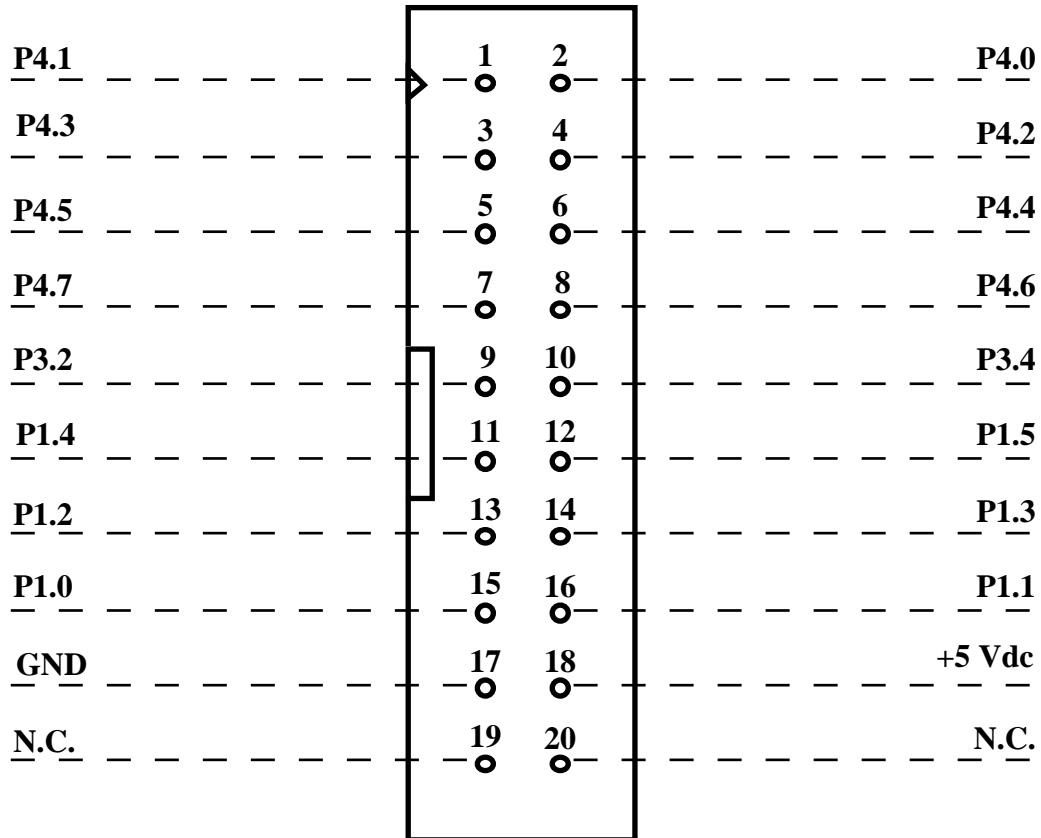


FIGURA 10: CN6 - CONNETTORE PER I/O DELLA CPU

Legenda:

- P1.n** = I/O - Linea digitale n del port 1 della CPU.
- P3.n** = I/O - Linea digitale n del port 3 della CPU.
- P4.n** = I/O - Linea digitale n del port 4 della CPU.
- GND** = - Linea di massa.
- +5 Vdc** = - Linea di alimentazione a +5 Vdc.
- N.C.** = - Non Collegato.

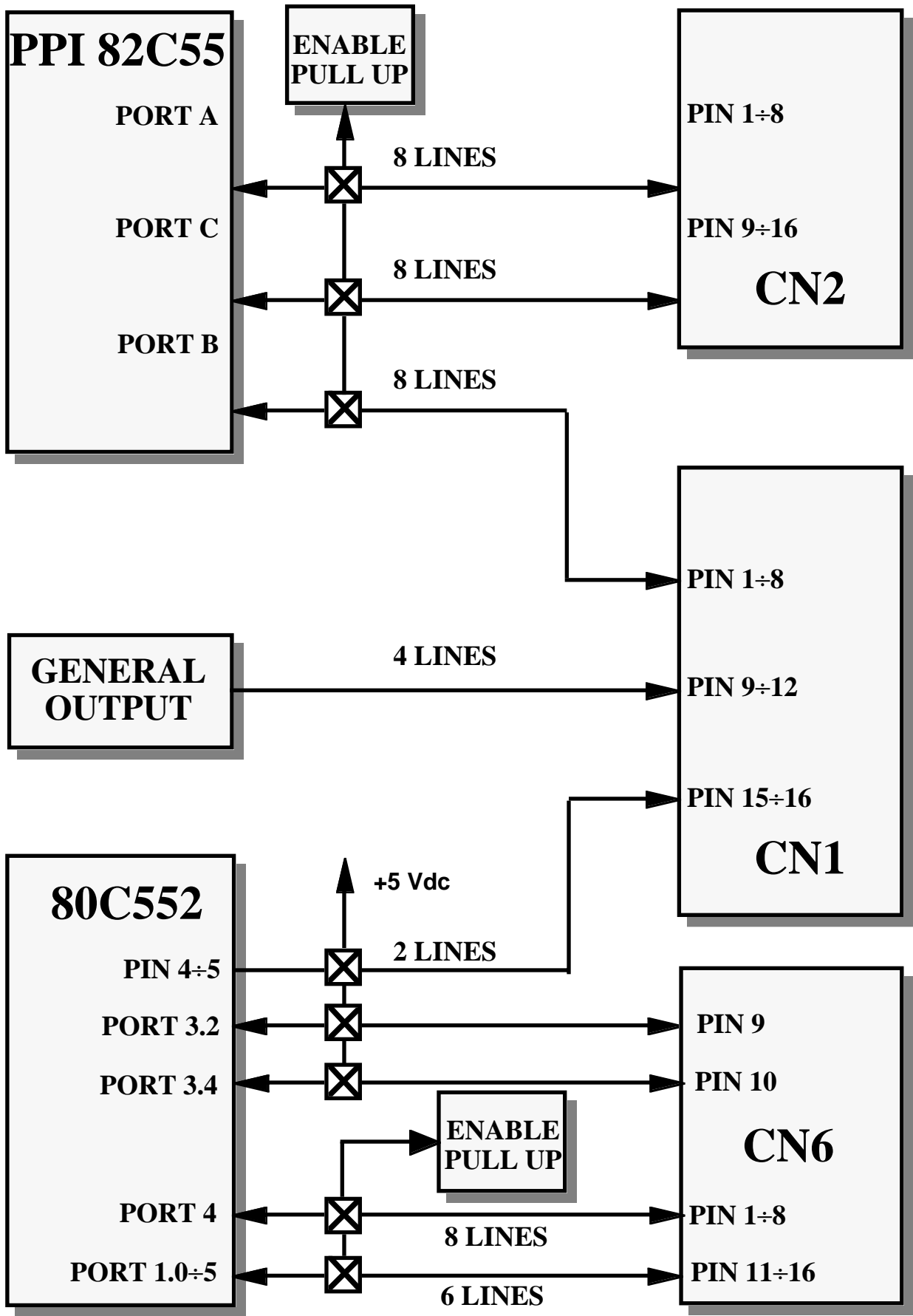


FIGURA 11: SCHEMA COLLEGAMENTO LINEE DI I/O

CN7 - CONNETTORE PER LA COMUNICAZIONE SERIALE.

Il connettore per la comunicazione seriale, in RS 232, RS 422-485 o Current Loop, denominato CN7 sulla scheda, é del tipo a vaschetta D a 9 vie femmina a 90°. La disposizione di tali 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.

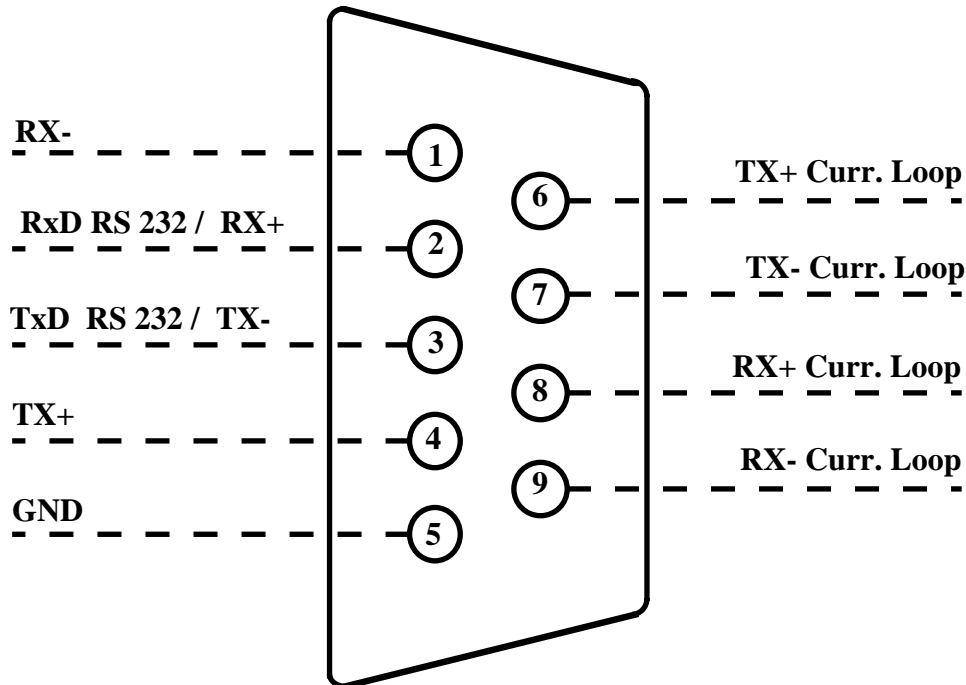


FIGURA 12: CN7-CONNETTORE PER COMUNICAZIONE SERIALE

Legenda:

RX-	=	I	-	Receive Data Negative: linea bipolare negativa per ricezione seriale differenziale in RS 422-485
RX+	=	I	-	Receive Data Positive: linea bipolare positiva per ricezione seriale differenziale in RS 422-485
TX-	=	O	-	Transmit Data Negative: linea bipolare negativa per trasmissione seriale differenziale in RS 422-485
TX+	=	O	-	Transmit Data Positive: linea bipolare positiva per trasmissione seriale differenziale in RS 422-485
RxD RS 232	=	I	-	Receive Data: linea di ricezione in RS 232 della linea seriale. Tale pin corrisponde con il pin RX+
TxD RS 232	=	O	-	Transmit Data: linea di trasmissione in RS 232 della linea seriale. Tale pin corrisponde con il pin TX-
RX- Curr. Loop	=	I	-	Receive Data Negative: linea bipolare negativa per ricezione seriale Current Loop
RX+ Curr. Loop	=	I	-	Receive Data Positive: linea bipolare positiva per ricezione seriale Current Loop
TX- Curr. Loop	=	O	-	Transmit Data Negative: linea bipolare negativa per trasmissione seriale Current Loop
TX+ Curr. Loop	=	O	-	Transmit Data Positive: linea bipolare positiva per trasmissione seriale Current Loop
GND	=			Linea di massa

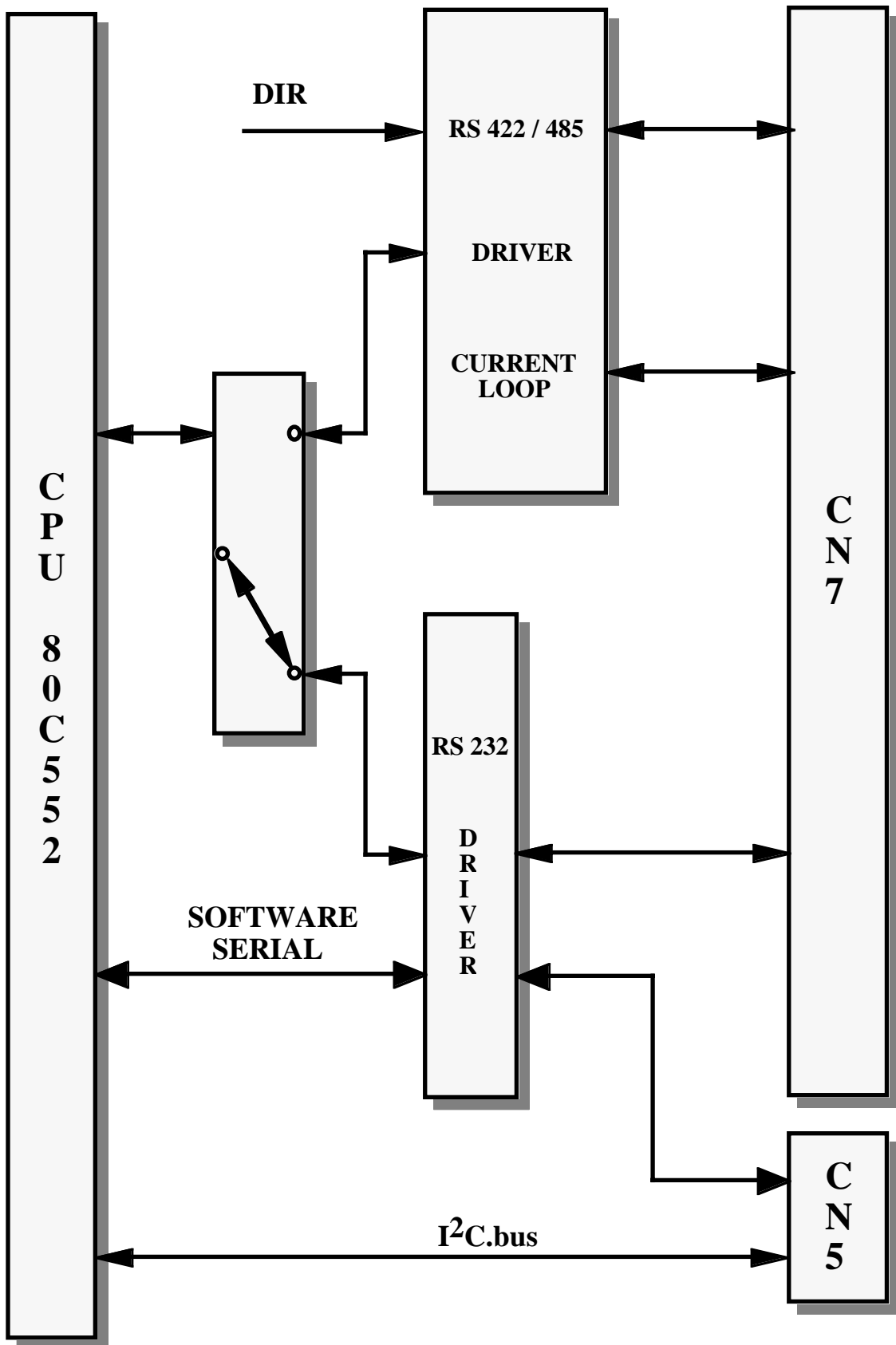


FIGURA 13: SCHEMA DI COMUNICAZIONE SERIALE

SEGNALAZIONI VISIVE

La scheda **GPC® 552** é dotata di 2 LEDs con cui segnala alcune condizioni di stato, come descritto nelle seguenti tabelle:

LEDs	COLORE	FUNZIONE
LD1	Rosso	Segnala, quando attivo, la presenza della tensione di alimentazione a +5Vdc
LD2	Verde	LED di attività gestito via software

FIGURA 14: TABELLA DELLE SEGNALAZIONI VISIVE

La funzione principale di questi LEDs é quella di fornire un'indicazione visiva dello stato della scheda, facilitando quindi le operazioni di debug e di verifica di funzionamento di tutto il sistema. Per una più facile individuazione di tali segnalazioni visive, si faccia riferimento alla figura 15.

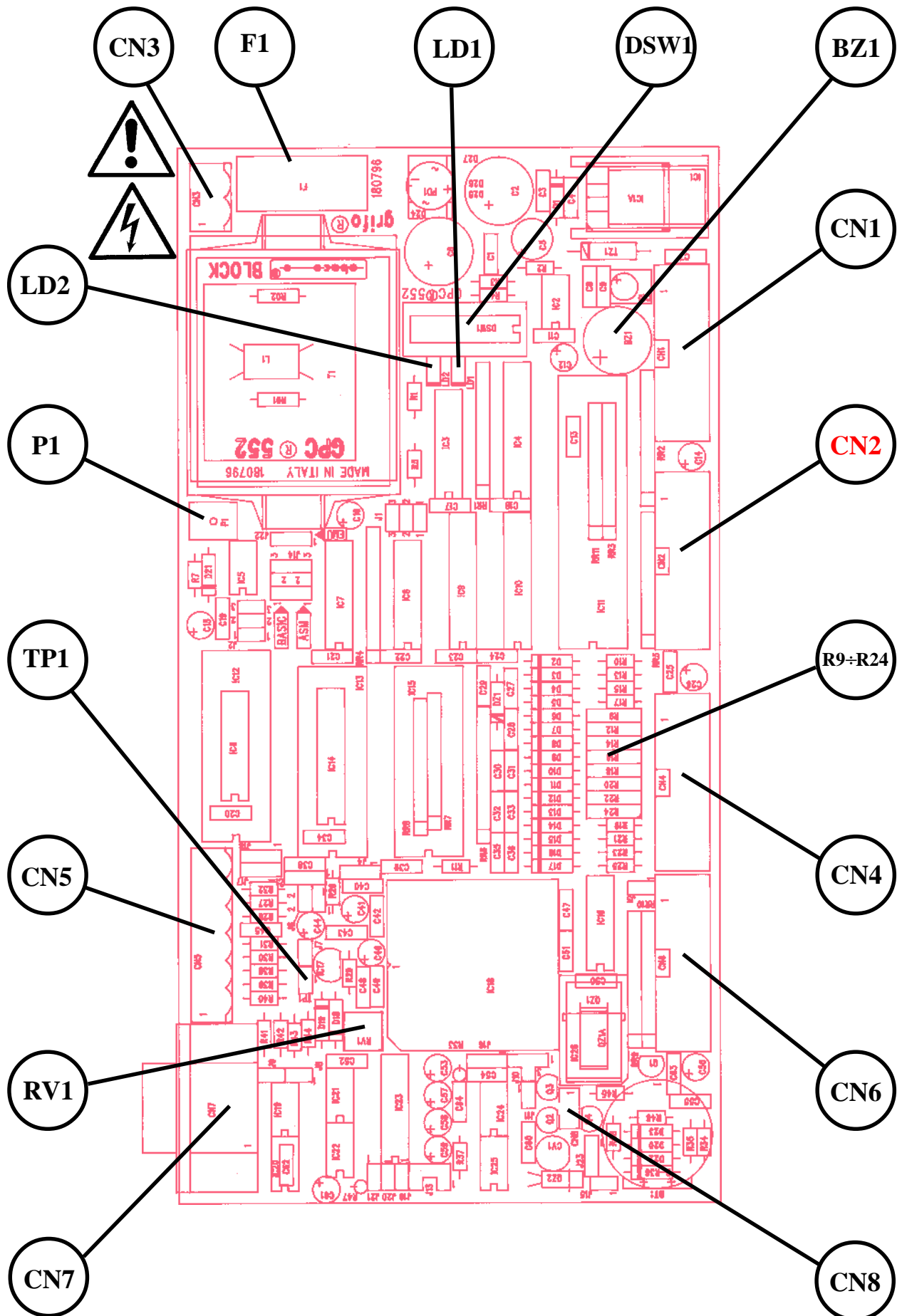


FIGURA 15: DISPOSIZIONE LEDS, CONNETTORI, DIP-SWITCH, ECC.

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 la **GPC® 552** si deve interfacciare, si devono seguire le informazioni riportate nei precedenti paragrafi e le relative figure che illustrano le modalità interne di connessione.

- Per tutti i segnali che riguardano la comunicazione seriale con i protocolli RS 232, RS 422-485 o Current Loop, fare riferimento alle specifiche standard di ognuno di questi protocolli.
- Tutti i segnali a livello TTL devono 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.
- I segnali d'ingresso alla sezione A/D devono essere collegati a segnali analogici a bassa impedenza che rispettino il range di variazione ammesso che può essere $0 \div +2,49 V_{cc}$, $0 \div +5 V_{cc}$ o $0 \div 20 mA$ a seconda della configurazione. Per maggiori informazioni si veda paragrafo "A/D CONVERTER".

TASTO DI RESET

Con il tasto P1 presente sulla **GPC® 552** si ha la possibilità di attivare la linea di /RESET della scheda. Una volta premuto il tasto P1, la scheda riprende l'esecuzione del programma in EPROM, partendo da una condizione di azzeramento generale. La funzione principale di questo tasto è quella di uscire da condizioni di loop infinito, soprattutto durante la fase di debug. Per una facile individuazione di tale pulsante a bordo scheda, si faccia riferimento alla figura 15.

TEST POINT

La scheda è provvista di un test point denominato TP1, che permette la lettura attraverso un multimetro galvanicamente isolato, della tensione di riferimento che viene regolata in laboratorio a $V_{ref}=2.4900 V$ (default) o $V_{ref}=5.0000 V$ (su richiesta). Il TP1 è composto da due contatti con la seguente corrispondenza:

pin 1	->	Vref
pin 2	->	GND

Per una facile individuazione di tale test point a bordo scheda, si faccia riferimento alla figura 15, mentre per ulteriori informazioni sul segnale Vref fare riferimento ai paragrafi "SELEZIONE TIPO INGRESSI ANALOGICI" e "TRIMMERS E TARATURE".

INPUT DI BORDO

La scheda **GPC® 552** è provvista di 1 dip switch a 8 vie (DSW1), tipicamente utilizzato per la configurazione del sistema, i cui valori sono acquisibili via software. Le applicazioni più immediate possono essere quelle destinate al settaggio delle condizioni di lavoro o alla selezione di parametri relativi al firmware di bordo. La lettura della combinazione fissata sul dip switch avviene in logica negata (0 -> dip in ON ed 1 -> dip in OFF) effettuando un'operazione di input all'indirizzo di I/O dedicatogli dalla logica di controllo della scheda. Per ulteriori informazioni si faccia riferimento ai paragrafi "MAPPAGGIO DELL'I/O" e "MAPPAGGIO DELLE MEMORIE", mentre per una facile individuazione della loro posizione si veda la figura 15.

SELEZIONE TENSIONI DI ALIMENTAZIONE



La scheda **GPC® 552** dispone di una efficiente circuiteria di alimentazione che si presta a risolvere in modo comodo ed efficace il problema dell'alimentazione della scheda in qualsiasi condizione di utilizzo. Di seguito vengono riportate le possibili configurazioni della sezione alimentatrice:

- Alimentazione da rete standard: in questa configurazione la scheda deve essere alimentata dalla tensione di rete a 230 Vac che viene fornita sui pins 1 e 2 di CN3.
- Alimentazione senza sezione alimentatrice: in questa configurazione la scheda deve essere alimentata con +5 Vdc che viene fornita sui pins 1 e 2 di CN3.
- Alimentazione a bassa tensione lineare: in questa configurazione la scheda deve essere alimentata da una tensione di 6÷10 Vac o corrispondente tensione continua (es. 12 Vdc), che deve essere fornita sui pins 1 e 2 di CN3.
- Alimentazione a bassa tensione switching: in questa configurazione la scheda deve essere alimentata da una tensione di 8÷26 Vac o corrispondente tensione continua, che deve essere fornita sui pins 1 e 2 di CN3.

Indipendentemente dalla sezione alimentatrice scelta la **GPC® 552** é sempre dotata di un'efficace circuiteria di protezione che si preoccupa di proteggere la scheda da sovratensioni o dal rumore del campo. Si ricorda che la selezione del tipo di sezione alimentatrice della scheda, deve avvenire in fase di ordine della stessa; infatti questa scelta implica una diversa configurazione hardware, che deve essere effettuata dal personale addetto.

SELEZIONE TIPO INGRESSI ANALOGICI

La scheda **GPC® 552**, può avere ingressi analogici in tensione e/o corrente, come descritto nei precedenti paragrafi e capitoli. La selezione del tipo d'ingresso dei canali analogici é effettuata in fase di ordine della scheda. Per quanto riguarda il fondo scala degli ingressi analogici é possibile selezionare una tensione di +2.490 V o di +5 V. La selezione tra corrente-tensione é effettuata montando un apposito modulo di conversione basato su semplici resistenze di caduta. In particolare vale la corrispondenza:

R9	->	canale 0
R12	->	canale 1
R14	->	canale 2
R16	->	canale 3
R18	->	canale 4
R20	->	canale 5
R22	->	canale 6
R24	->	canale 7

Nel caso il modulo corrente-tensione non sia montato (default) il corrispondente canale accetta un ingresso in tensione nei range 0÷2,49 V (default) o 0÷5 V (da specificare in fase di ordine); viceversa un ingresso in corrente. Il valore della resistenza, su cui si basa il convertitore corrente-tensione, si ottiene dalla seguente formula:

$$R = +2,49 \text{ V} / I_{max} \quad \text{o} \quad R = +5 \text{ V} / I_{max}$$

Normalmente i moduli di conversione tensione-corrente, si basano su resistenze di precisione da **124Ω** o **248Ω**, relative ad ingressi 4÷20 mA o 0÷20 mA. Per una facile individuazione del modulo descritto e delle relative resistenze componenti, fare riferimento alle figure 15.

TRIMMERS E TARATURE

Sulla **GPC® 552** é presente un trimmer da utilizzare per la taratura della scheda. In particolare con il trimmer RV1 si può fissare il valore della tensione di riferimento su cui si basa l'eventuale sezione di A/D converter. Per una facile individuazione del trimmer a bordo scheda, si faccia riferimento alla figura 15.

La scheda viene sottoposta ad un accurato test di collaudo che provvede a verificare la funzionalità della stessa ed allo stesso tempo a tararla in tutte le sue parti. La taratura viene effettuata in laboratorio a temperatura costante di +20 gradi centigradi, seguendo la procedura di seguito descritta:

- Si effettua la taratura di precisione della Vref della sezione A/D tramite la regolazione del trimmer RV1, tramite un multimetro galvanicamente isolato a 5 cifre ad un valore di 2,4900 V o 5,0000 V.
- Si verifica la corrispondenza tra segnale analogico fornito in ingresso e combinazione letta dalla sezione A/D converter. La verifica viene effettuata fornendo il segnale di verifica con un calibratore campione e controllando che la combinazione determinata dalla scheda e quella determinata in modo teorico non differiscano di quella che é la somma degli errori della sezione A/D della scheda.
- Si blocca il trimmer della scheda, opportunamente tarato, tramite vernice.

Le sezioni d'interfaccia analogica utilizzano componenti di alta precisione che vengono addirittura scelti in fase di montaggio, proprio per evitare lunghe e complicate procedure di taratura. Per questo una volta completato il test di collaudo e quindi la taratura, tutti i trimmer della scheda vengono bloccati, in modo da garantire una immunità della taratura anche ad eventuali sollecitazioni meccaniche (vibrazioni, spostamenti, ecc.).

L'utente di norma non deve intervenire sulla taratura della scheda, ma se lo dovesse fare (a causa di derive termiche, derive del tempo, ecc.) deve rigorosamente seguire la procedura sopra illustrata.

INTERFACCIE OPERATORE LOCALI

Tramite CN2 (connettore standard di I/O **Abaco®**) si può collegare la **GPC® 552** ai numerosi moduli del carteggio **Grifo®** che riportano lo stesso pin out. Di particolare interesse é la possibilità di collegare direttamente una serie di moduli come la **QTP 24P**, **KDL x24**, **KDF 224**, ecc. con cui risolvere tutti i problemi di interfacciamento operatore locale. Questi moduli sono già dotati delle risorse necessarie per gestire un buon livello di colloquio uomo-macchina (includono infatti display alfanumerici, tastiera a matrice e LEDs di visualizzazione) ad una breve distanza dalla **GPC® 552**. Dal punto di vista dell'installazione, queste interfacce operatore locali richiedono un solo flat a 20 vie con cui é possibile portare anche le alimentazioni, mentre dal punto di vista software la gestione é altrettanto semplice ed immediata, infatti i pacchetti software disponibili per la **GPC® 552** sono provvisti di tutte le procedure necessarie. Quest'ultime per la maggioranza dei pacchetti software disponibili, coincidono con dei "driver software" aggiunti al linguaggio di programmazione, che consentono di utilizzare direttamente le istruzioni di "console" dello stesso linguaggio di programmazione e quindi tutta la loro potenza. Per maggiori informazioni relative alle interfacce operatore locali si veda il capitolo "SCHEDE ESTERNE" e la documentazione del software utilizzato.

JUMPERS

Esistono a bordo della **GPC® 552** 22 jumpers, con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento della stessa. Di seguito ne é riportato l'elenco, l'ubicazione e la loro funzione nelle varie modalit  di connessione. Per riconoscere tali connessioni sulla scheda si faccia riferimento alla serigrafia della stessa o alla figura 2 . Per l'individuazione dei jumpers a bordo della scheda, si utilizzi invece la figura 18.

JUMPERS	N. VIE	UTILIZZO
J1	2 Triplo	Seleziona size IC 15 tra 8, 16, 32 KBytes.
J2	2 Triplo	Seleziona size IC 13 tra 16, 24, 32 KByte.
J3	3	Predisporre IC12 per RAM/EEPROM/EPROM da 8 o 32 KBytes.
J4	3	Predisporre IC 13 per RAM/EEPROM da 8 o 32 KByte.
J5	2	Seleziona la lettura codice dalla ROM interna o dalla ROM esterna.
J6	2 Doppio	Gestisce il PULL-UP delle linee SCL e SDA.
J7	2	Collega a Vdc il pin 5 di CN 5.
J8, J9	2	Collegano la circuiteria di terminazione e di forzatura alla linea di comunicazione seriale in RS 422-485.
J10	4	Seleziona il segnale da collegare al pin INT0 della CPU.
J11	2	Gestisce l' abilitazione hardware del WATCH-DOG.
J13	5	Seleziona direzionalit� e modalit� di attivazione della linea seriale in RS 422-485.
J14	3 Triplo	Seleziona la configurazione per il funzionamento in BASIC o in ASSEMBLER della scheda.
J15	2	Configura la circuiteria di back up per RAM e RTC.
J16	3	Seleziona il segnale da collegare al pin T0 della CPU.
J17	3	Seleziona il size e il dispositivo di memoria su IC 12.
J18	3	Seleziona il dispositivo di memoria su IC 12.
J19, J20	2	Collegano il driver RS 232 della seriale primaria al CN7.
J21	2	Abilita la ricezione della seriale primaria in RS232.
J22	3	Abilita l' utilizzo di mappaggi di memoria speciali per usi futuri.
J23	3	Gestisce l' alimentazione delle reti resistive utilizzate dai PORT 4 e 1 della CPU e dai PORT A, B e C dell' 82C55.

FIGURA 16: TABELLA RIASSUNTIVA JUMPERS

JUMPERS A 2 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J1	connesso pos. 1	Definisce un size di 32 KBytes su IC 15	*
	connesso pos. 2	Definisce un size di 16 KBytes su IC 15	
	connesso pos. 3	Definisce un size di 8 KBytes su IC 15	
J2	connesso pos. 1	Definisce un size di 32 KBytes su IC 13	*
	connesso pos. 2	Definisce un size di 16 KBytes su IC 13	
	connesso pos. 3	Definisce un size di 24 KBytes su IC 13	
J5	non connesso	Abilita la lettura codice dalla ROM interna (tagliare la pista nel lato saldature)	*
	connesso	Abilita la lettura codice dalla ROM esterna	
J6	non connesso pos. 1	Resistenza pull-up scollegata alla linea SCL	*
	connesso pos. 1	Resistenza pull-up collegata alla linea SCL	
	non connesso pos. 2	Resistenza pull-up scollegata alla linea SDA	*
	connesso pos. 2	Resistenza pull-up collegata alla linea SDA	
J7	non connesso	Il pin 5 di CN 5 é libero	*
	connesso	Collega il pin 5 di CN 5 a Vdc	
J8, J9	non connessi	Non collegano la circuiteria di terminazione e di forzatura alla linea di comunicazione seriale primaria in RS 422-485	*
	connessi	Collegano la circuiteria di terminazione e di forzatura alla linea di comunicazione seriale primaria in RS 422-485	
J11	non connesso	Disabilitazione hardware del watch-dog	*
	connesso	Il watch-dog é sempre abilitato	
J15	non connesso	RAM di IC 13 tamponata solo dall' eventuale batteria esterna. RAM+RTC di IC 25 tamponata solo dalla batteria di bordo BT1	*
	connesso	RAM di IC 13 tamponata dall' eventuale batteria esterna e dall' eventuale batteria di bordo BT1. RAM+RTC di IC 25 tamponata dalla batteria di bordo BT1 e dall' eventuale batteria esterna	
J19, J20	non connessi	Non collegano il driver RS 232 della linea seriale primaria al CN7	*
	connessi	Collegano il driver RS 232 della linea seriale primaria al CN7	
J21	non connesso	Disabilita ricezione della seriale primaria in RS232	

FIGURA 17: TABELLA JUMPERS A 2 VIE

L' * indica la connessione di default, ovvero la connessione con cui la scheda viene fornita.

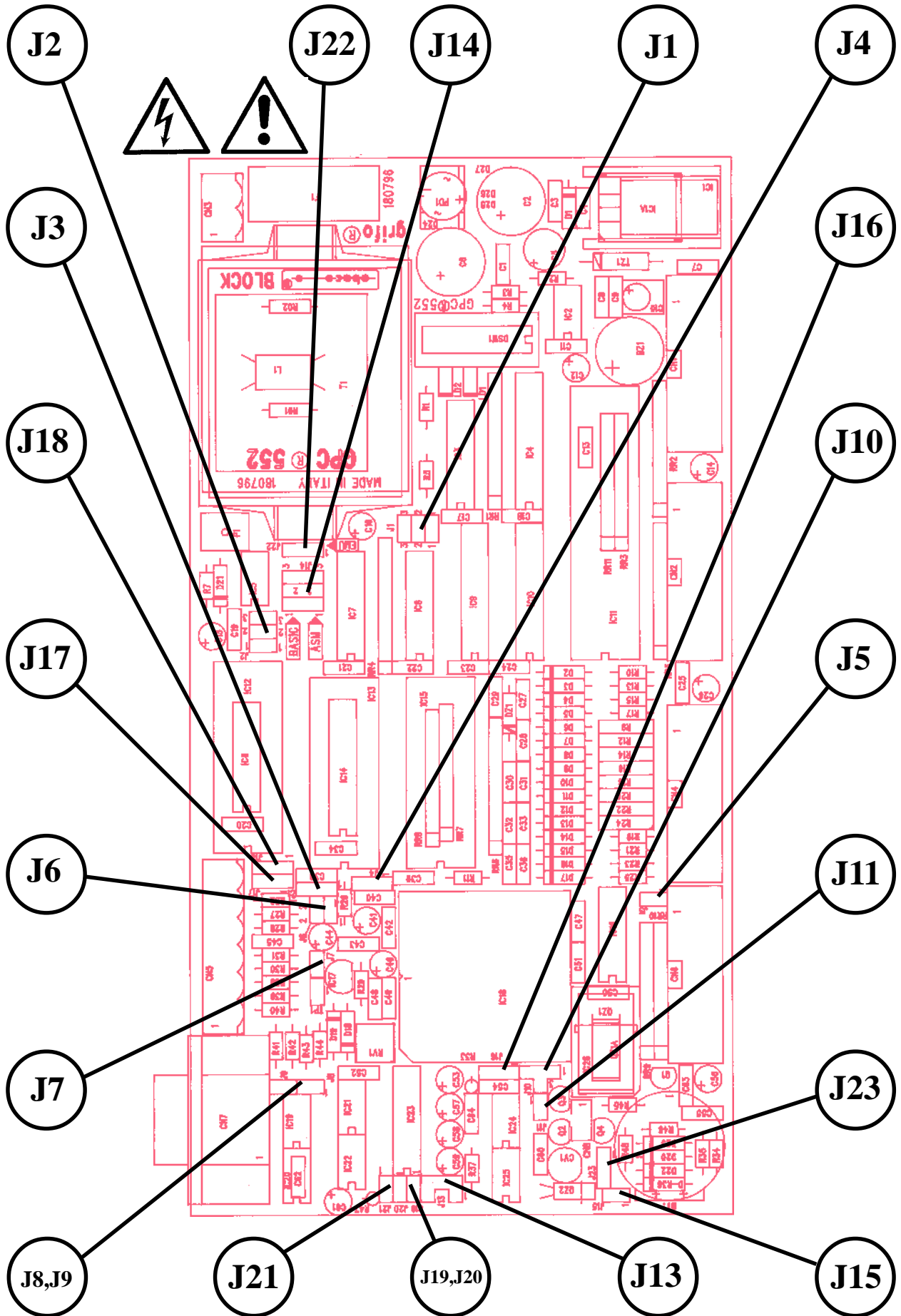


FIGURA 18: DISPOSIZIONE JUMPERS

JUMPERS A 3 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J3	posizione 1-2	Predisporre IC12 per RAM/EEPROM/EPROM da 32 KBytes	*
	posizione 2-3	Predisporre IC12 per RAM/EEPROM/EPROM da 8 KBytes	
J4	posizione 1-2	Predisp. IC13 per RAM/EEPROM da 32 KBytes	*
	posizione 2-3	Predisp. IC13 per RAM/EEPROM da 8 KBytes	
J14 (1,2,3)	posizione BASIC	Si configura la scheda per il funzionamento in BASIC	*
	posizione ASM	Si configura la scheda per il funzionamento in ASSEMBLER	
J16	posizione 1-2	Collega il segnale T0 della CPU al Tx2 della seconda linea seriale in RS 232 (seriale software)	*
	posizione 2-3	Collega il segnale T0 della CPU al pin 10 di CN 6	
J17	posizione 1-2	Predisporre IC12 per EPROM	*
	posizione 2-3	Predisporre IC12 per RAM/EEPROM da 32 K	
	Non connesso	Predisporre IC12 per RAM/EEPROM da 8 K	
J18	posizione 1-2	Predisporre IC12 per EPROM	*
	posizione 2-3	Predisporre IC12 per RAM/EEPROM	
J22	posizione 1-2	Modifica i mappaggi per usi futuri	*
	posizione 2-3	Mantiene i mappaggi di memoria già definiti	
J23	posizione 1-2	Mantiene abilitato il pull up sulle linee dei PORT 4 e 1 della CPU e dei PORT A, B e C dell' 82C55	*
	posizione 2-3	Gestione software del pull up o pull down sulle linee dei PORT 4 e 1 della CPU e dei PORT A, B e C dell' 82C55	

FIGURA 19: TABELLA JUMPERS A 3 VIE

L' * indica la connessione di default, ovvero la connessione impostata in fase di collaudo, con cui la scheda viene fornita.

JUMPERS A 4 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J10	posizione 1-2	Collega il segnale INT0 della CPU al Rx2 della seconda linea seriale in RS 232 (seriale software)	*
	posizione 2-3	Collega il segnale INT0 della CPU al pin 9 di CN6	
	posizione 2-4	Collega il segnale INT0 della CPU al pin 7 di IC25 (interrupt del Real Time Clock)	

FIGURA 20: TABELLA JUMPERS A 4 VIE

L' * indica la connessione di default, ovvero la connessione impostata in fase di collaudo, con cui la scheda viene fornita.

JUMPERS A 5 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J13	posizione 1-2 e 3-4	Seleziona la comunicazione su linea seriale in RS 485 (half duplex a 2 fili)	*
	posizione 2-3 e 4-5	Seleziona la comunicazione su linea seriale in RS 422-485 (full duplex o half duplex a 4 fili)	

FIGURA 21: TABELLA JUMPERS A 5 VIE

L' * indica la connessione di default, ovvero la connessione impostata in fase di collaudo, con cui la scheda viene fornita.

NOTE

Vengono di seguito riportate una serie di indicazioni con cui descrivere in modo più dettagliato quali sono le operazioni da eseguire per configurare correttamente la scheda.

SELEZIONE MODO E ALIMENTAZIONE DELLE RETI RESISTIVE

Come già detto nella tabella di figura 19, J23 può essere strappato in due posizioni:

- posizione 1-2: é sempre abilitato il pull up sulle linee dei PORT 4 e 1 della CPU e dei PORT A, B, C dell' 82C55. Per questo all' atto di un power on o di un reset, bisogna prestare particolare attenzione alle relative uscite, in quanto verranno poste allo stato logico "1" con conseguente attivazione dei dispositivi di tipo NPN ad esse collegate.
- posizione 2-3: gestione software del pull up e del pull down sulle linee dei PORT 4 e 1 della CPU e dei PORT A, B, C dell' 82C55. Al power on o al reset, viene garantito lo stato logico "0" alle uscite (si evita così l' attivazione dei dispositivi di tipo NPN ad esse collegate). In seguito l' utente setterà lo stato iniziale delle uscite e infine attiverà il pull up agendo sul registro EPUL settando D0=1 (vedere para.grafo "SELEZIONE PULL UP/DOWN SULLE LINEE DI I/O DELLA SCHEDA").

SELEZIONE DEL TIPO DI COMUNICAZIONE SERIALE

La linea di comunicazione seriale A della scheda GPC® 552 può essere bufferata in RS 232, in RS 422-485 o in Current Loop. La selezione del tipo d'interfacciamento avviene via hardware e viene effettuata tramite un opportuno strappaggio dei jumpers di bordo, come può essere desunto dalla lettura delle precedenti tabelle. Dal punto di vista software sono invece definibili tutti i parametri del protocollo fisico di comunicazione tramite la programmazione dei registri interni della CPU.

Vengono di seguito riportate le possibili configurazioni che possono essere effettuate; da notare che i jumpers non menzionati nella successiva descrizione, non hanno alcuna influenza ai fini della comunicazione, qualunque posizione essi occupino.

- LINEA SERIALE A SETTATA IN RS 232

Su IC23 deve essere montato il driver MAX 232 mentre su IC19, IC20, IC21, IC22 non deve essere montato nessun driver. I jumpers J19, J20 e J21 devono essere connessi. La posizione dei tre jumpers J8, J9, J13 é ininfluente,

- LINEA SERIALE A SETTATA IN CURRENT LOOP

Su IC21 deve essere montato il driver HCPL 4100, su IC22 deve essere montato il driver HCPL 4200 mentre su IC19 e IC20 non deve essere montato nessun driver. I jumpers J19, J20 e J21 devono essere non connessi. La posizione dei tre jumpers J8, J9, J13 é ininfluente,

- LINEA SERIALE A SETTATA IN RS 485

Su IC 20 deve essere montato il driver SN75176 mentre su IC 21, IC 22, non deve essere montato nessun driver. Il jumper J13 deve essere in posizione 1-2 e 3-4, J19, J20 e J21 non connessi. In questa modalità le linee da utilizzare sono i pins 1 e 2 di CN7, che quindi diventano le linee di trasmissione o ricezione a seconda dello stato del segnale DIR gestito via software. Questa comunicazione la si utilizza per comunicazioni su sistemi multipunto, infatti il driver su IC 20 può essere settato in ricezione o in trasmissione, tramite la gestione del segnale DIR (0=basso= ricezione, 1=alto=trasmissione). Sempre in questa modalità é possibile ricevere 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.

- LINEA SERIALE A SETTATA IN RS 422

Su IC 20 e IC 19 devono essere montati i drivers SN75176 mentre su IC 21, IC 22, non deve essere montato nessun driver. Il jumper J13 deve essere in posizione 2-3 e 4-5, J19, J20 e J21 non connessi. Per sistemi punto punto, la linea DIR può essere mantenuta sempre alta (mantenendo il trasmettitore sempre attivo), mentre per sistemi multi_punto con la linea DIR si deve attivare il trasmettitore solo in corrispondenza della trasmissione sempre tramite la linea DIR(1=alto=trasmettitore attivo e viceversa).

Nel caso si utilizzi la linea seriale in RS 422-485, con i jumpers J8 e J9 é 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 multi_punto, deve essere collegata solo sulle schede che risultano essere alla maggior distanza, ovvero ai capi della linea di comunicazione.

SELEZIONE MEMORIE

La **GPC® 552** può montare fino ad un massimo di 98K e 256 bytes di memoria variamente suddivisa. In particolare valgono le informazioni riportate nella seguente tabella:

IC	DISPOSITIVO	DIMENSIONE	STRIPPAGGIO
12	RAM/EEPROM	8K Bytes	J18 in 2-3; J17 non connesso; J3 in 2-3
	RAM/EEPROM	32K Bytes	J18 in 2-3; J17 in 2-3; J3 in 1-2
	EPROM	8K Bytes	J18 in 1-2; J17 in 1-2; J3 non connesso
	EPROM	32K Bytes	J18 in 1-2; J17 in 1-2; J3 in 1-2
13	RAM/EEPROM	8K Bytes	J4 in posizione 2-3
	RAM/EEPROM	32K Bytes	J4 in posizione 1-2
15	EPROM	32K Bytes	
24	EEPROM	512÷ 2048 Bytes	
25	RAM+RTC	256 Bytes	

FIGURA 22: TABELLA DI SELEZIONE MEMORIE

Tutti i dispositivi sopra descritti devono essere con pin out di tipo JEDEC a parte l'EEPROM seriale di IC 24 e la RAM+RTC seriale IC 25 che devono essere richiesti alla **GRIFO®** in fase di ordine della scheda. Per quanto riguarda le sigle dei vari dispositivi di memoria che possono essere montati sulla scheda, fare riferimento alla documentazione della casa costruttrice. Da ricordare che nel caso sia presente il componente di IC 25, le dimensioni massime di IC 24 sono pari a 1024 bytes. I moduli di RAM per IC 12 ed IC 13, possono, su richiesta, essere del tipo tamponato.

BACK UP

La **GPC® 552** può montare una batteria di back up al litio che permette di mantenere le informazioni salvate nella RAM del componente PCF 8583 (IC25) ed il settaggio dell' orologio di bordo, anche quando la scheda non é alimentata. Premettendo che sulla **GPC® 552** la batteria al litio BT1 é opzionale e viene fornita assieme all'opzione di RAM+RTC di IC25:

- Se l'utente deve tamponare la RAM di IC13 deve obbligatoriamente collegare una batteria esterna al CN8. Con il jumper J15 può tamponare anche l'eventuale circuiteria di RAM+RTC di IC 25, salvaguardando quindi la durata della batteria di bordo BT1.
- Se la scheda é provvista della batteria di bordo BT1, con il jumper J15 l'utente può tamponare anche la RAM di IC13, ma solo per brevi intervalli di tempo, come ad esempio quello di sostituzione della batteria esterna.

Tutte le altre condizioni non sono previste e quindi non possono essere usate.

DESCRIZIONE SOFTWARE

Questa scheda ha la possibilità di usufruire di una ricca serie di strutture software che consentono di utilizzarne al meglio le caratteristiche. In generale la scheda 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. Tra questi ricordiamo:

MDP: monitor debugger in grado di caricare e debuggare un qualsiasi file HEX con codice 'I51. Dispone di tutti i comandi normalmente disponibili con un emulatore e fornisce quindi all'utente la possibilità di operare comodamente con tutte le risorse di bordo. Per questo pacchetto software è sufficiente disporre di un P.C. che effettua le sole operazioni di console nei confronti dell'utente.

FORTH: completa struttura di sviluppo che consente di programmare la scheda in FORTH. Richiede un P.C. per l'interfaccia utente e rende disponibili strutture dati e di programmazione ad alto livello, che velocizzano lo sviluppo dell'applicativo con ottime caratteristiche in termini di codice sviluppato e velocità di esecuzione.

DESIGN-51: Emulatore per 8051, 80C51, 8052, 80C52, 8031, 80C31, 8032, 80C32, 8751, 8051, 80C451, 80C552, 80C562, 80C652, 87C751, 87C752. Potente struttura a basso costo che comprende un In Circuit Emulator, un Symbolic Debugger ed un cross assembler.

BASIC 323: completa struttura di sviluppo che consente di programmare la scheda con un BASIC interpretato adatto alle applicazioni industriali. Per operare è sufficiente un P.C. che svolge le funzioni di console nei confronti della scheda su cui viene invece sviluppato, debuggato, provato e salvato il programma da realizzare. La programmazione è ad alto livello ed interessa la maggioranza dei dispositivi a bordo scheda di cui vengono già forniti i driver software di facile utilizzo.

BXC51: Cross compilatore per files sorgenti scritti in BASIC-552. Disponibile in ambiente MS-DOS, permette un notevole incremento in termini di velocità di esecuzione rispetto all'equivalente programma in BASIC interpretato.

MICRO/ASM-51: Macro Cross Assembler. Disponibile in ambiente MS-DOS e nella versione "ASSOLUTA" o "RILOCABILE", permette una facile ed efficiente programmazione in assembler, dei microcontrollori basati sull'8051. In versione "RILOCABILE", viene anche fornito un LINKER ed un GESTORE DI LIBRERIE.

MICRO/C-51: Integer Cross Compiler per files sorgenti scritti in linguaggio "C". Disponibile in ambiente MS-DOS, genera un source assembly compatibile con il MICRO/ASM-51 o con il Macro Assembler rilocabile dell'Intel (MCS-51).

MICRO/SLD-51: Simulatore e Debugger a livello source. Simulatore/Debugger in grado di simulare i microcontrollori della famiglia I51 e di monitorare lo stato di esecuzione di un programma. Permette tramite un PC e senza l'aggiunta di emulatori o hardware addizionale, il caricamento o il salvataggio di file HEX o SIMBOLICI, il settaggio di breakpoints, l'esecuzione in modalità TRACE di istruzioni "C" e/o "ASSEMBLER", la visualizzazione di qualsiasi registro o variabile, ecc.

HI-TECH C: Cross compilatore per file sorgenti scritti in linguaggio "C". E' un potente pacchetto software che tramite un comodo I.D.E. permette di utilizzare un editor, un compilatore "C" (floating-point), un assembler, un linker e un remote debugger. Sono inoltre inclusi i source delle librerie.

XPAS51: Cross compilatore per files sorgenti scritti in Pascal, disponibile in ambiente MS-DOS.

DDS C: E' un comodo pacchetto software, a basso costo, che tramite un completo I.D.E. permette di utilizzare un editor, un compilatore "C" (integer), un assembler, un linker e un remote debugger abbinato ad un monitor. Sono inclusi i sorgenti delle librerie ed una serie di utility.

RSD 552: Potente struttura che comprende un Remote Symbolic Debugger ed un cross assembler. E' necessario un P.C. su cui gira il pacchetto che si collega in RS 232 alla scheda e con cui l'utente può assemblare, caricare, esaminare e debuggare il programma applicativo. L'interfaccia utente è evoluta e completa di tutto lo stato del microprocessore.

DESCRIZIONE HARDWARE

INTRODUZIONE

In questo capitolo ci occuperemo di fornire tutte le informazioni relative all'utilizzo della scheda, dal punto di vista della programmazione via software. Tra queste si trovano le informazioni riguardanti il mappaggio della scheda e la gestione software delle sezioni componenti.

MAPPAGGIO DELLE RISORSE DI BORDO

La gestione delle risorse della scheda é affidata ad una logica di controllo completamente realizzata con porte CMOS. Essa si occupa, con un minimo assorbimento di corrente, del mappaggio delle zone di RAM ed EPROM e di tutte le periferiche di bordo.

La logica di controllo é realizzata in modo da gestire separatamente il mappaggio delle memorie di bordo ed il mappaggio delle periferiche viste in Input/Output. Complessivamente la CPU 80C552 indirizza direttamente 64K Byte di memoria, quindi alla logica di controllo è assegnato il compito di allocare lo spazio logico d'indirizzamento delle memorie nello spazio fisico massimo di 64K Byte. Questa gestione è effettuata via hardware tramite lo strippaggio di alcuni jumpers (J1, J2, J3, J4, J14) con cui si può definire quali memorie utilizzare e il range di indirizzamento per ciascuna di esse. Per quanto riguarda il mappaggio dell'I/O si deve invece ricordare che la logica di controllo provvede naturalmente a non utilizzare le locazioni riservate per le periferiche interne della CPU, in modo da evitare ogni problema di conflittualità.

Per l'I/O si sono utilizzati gli ultimi 8 bytes dei 64K disponibili per l'indirizzamento.

Riassumendo i dispositivi mappati sulla scheda sono essenzialmente:

- Fino a 32K Bytes di EPROM su IC 15
- Fino a 32K Bytes di RAM/EEPROM su IC 13 (meno gli ultimi 8 bytes utilizzati per l'I/O)
- Fino a 32K Bytes di EEPROM/RAM/EPROM su IC 12 (meno gli ultimi 8 bytes utilizzati per l'I/O)
- PPI 82C55
- Dip switch di configurazione DSW1
- Settaggio linea di EPUL (gestione del pull-up o pull-down delle linee TTL)
- Settaggio linea DIR (direzionalità driver nella comunicazione in RS 422-485)
- Buzzer
- LED di attività (e linea EMU utilizzata in future espansioni)
- Settaggio 4 uscite generiche
- Lettura dello stato delle linee utilizzate per la gestione dei 5 punti precedenti

Questi occupano gli indirizzi riportati nei paragrafi seguenti e non possono essere riallocati in nessun altro indirizzo.

Altri dispositivi come la EEPROM di IC 24 e la RAM+RTC di IC 25 sono sempre gestiti dalla logica di controllo, ma effettivamente non occupano spazio d'indirizzamento in quanto sfruttano una comunicazione seriale sincrona gestita tramite linee di I/O della CPU.

MAPPAGGIO DELL'I/O

Come detto precedentemente, per l'I/O si sono utilizzati gli ultimi 8 indirizzi (6 utilizzati, 2 riservati per future espansioni) dei 64K Bytes massimi di indirizzamento utilizzabili. Naturalmente in corrispondenza degli indirizzi usati per i dispositivi di I/O non è possibile gestire alcun dispositivo di memoria infatti la logica di controllo seleziona sempre solo un dispositivo, in modo da eliminare ogni tipo di conflittualità. Viene di seguito riportato l'indirizzamento delle periferiche di bordo, escluse quelle interne alla CPU; per maggior chiarezza si riporta il nome del registro, il suo indirizzo, il tipo di accesso ed una breve descrizione del loro significato:

DISP.	REG.	IND.	R/W	SIGNIFICATO
PPI 82C55	PDA	FFF8H	R/W	Registro dati del port A
	PDB	FFF9H	R/W	Registro dati del port B
	PDC	FFFAH	R/W	Registro dati del port C
	CNT	FFFBH	R/W	Registro di controllo e comando
LED ATT.	LD2	FFFCH	R/W	Registro di gestione del LED di attività
BUZZER	BUZ	FFFCH	R/W	Registro di gestione del buzzer
ALIMEN. RETI RESISTIVE	EPUL	FFFCH	R/W	Registro di selezione modo e alimentazione delle reti resistive.
DIREZ. DRIVER RS 422-485	DIR	FFFCH	R/W	Registro di gestione direzione driver per RS 422-485
OUT GENERICI	OUTG	FFFCH	R/W	Registro di gestione delle 4 uscite generiche
DSW1	DSW1	FFFDH	R	Registro di lettura del DSW1

FIGURA 23: TABELLA INDIRIZZAMENTO I/O

Per quanto riguarda la descrizione del significato dei registri qui sopra riportati, si faccia riferimento al capitolo successivo "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO".

MAPPAGGIO DELLE MEMORIE

Per quanto riguarda il mappaggio delle memorie, la scheda può essere configurata in 4 modi. Di seguito viene riportata una schematizzazione di questi indirizzamenti, con le indicazioni di come devono essere strappati i jumpers che riguardano la gestione delle stesse memorie. La selezione della configurazione delle memorie viene effettuata in base all'applicazione da realizzare, alle specifiche esigenze dell'utente ed al pacchetto software in uso. Ad esempio alcune configurazioni tipiche sono:

BAS_552 --> MAPPAGGIO 4; IC 13 = 32 K RAM; IC 12 = 8 o 32 K EEPROM
 MDP --> MAPPAGGIO 2; IC 13 = 32 K RAM; IC 12 = 32 K RAM
 HI_TECH C --> MAPPAGGIO 1; IC 13 = 32 K RAM; IC 12 = 32 K RAM

MAPPAGGIO 1

CODE and DATA AREA

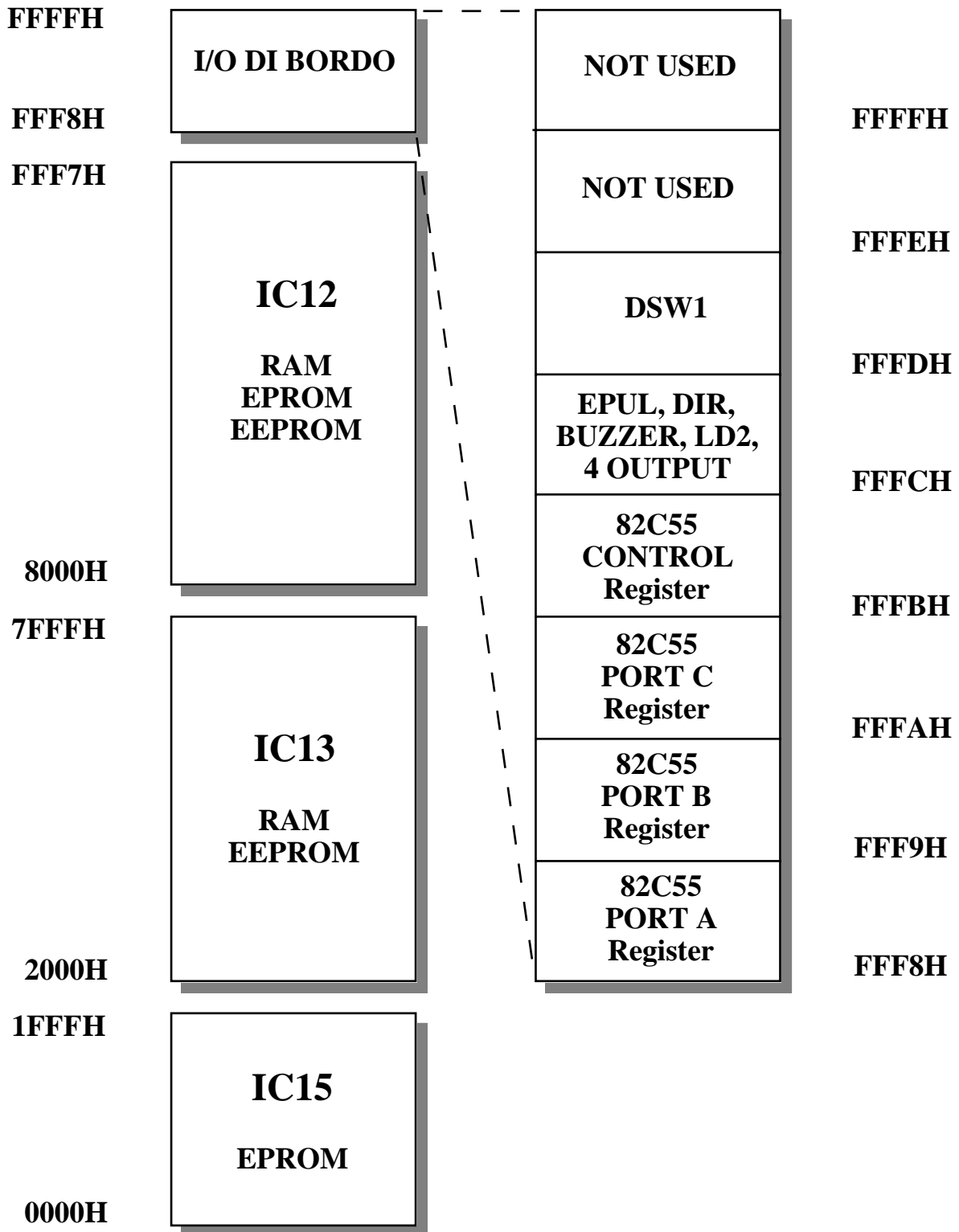


FIGURA 24: MAPPAGGIO DELLE MEMORIE IN MODO 1

Jumpers: J1 in posizione 3; J2 in posizione 3; J14(1,2,3) in posizione ASM
 Usato da pacchetti software come: HI TECH C; DDS C; ecc.



MAPPAGGIO 2

CODE and DATA AREA

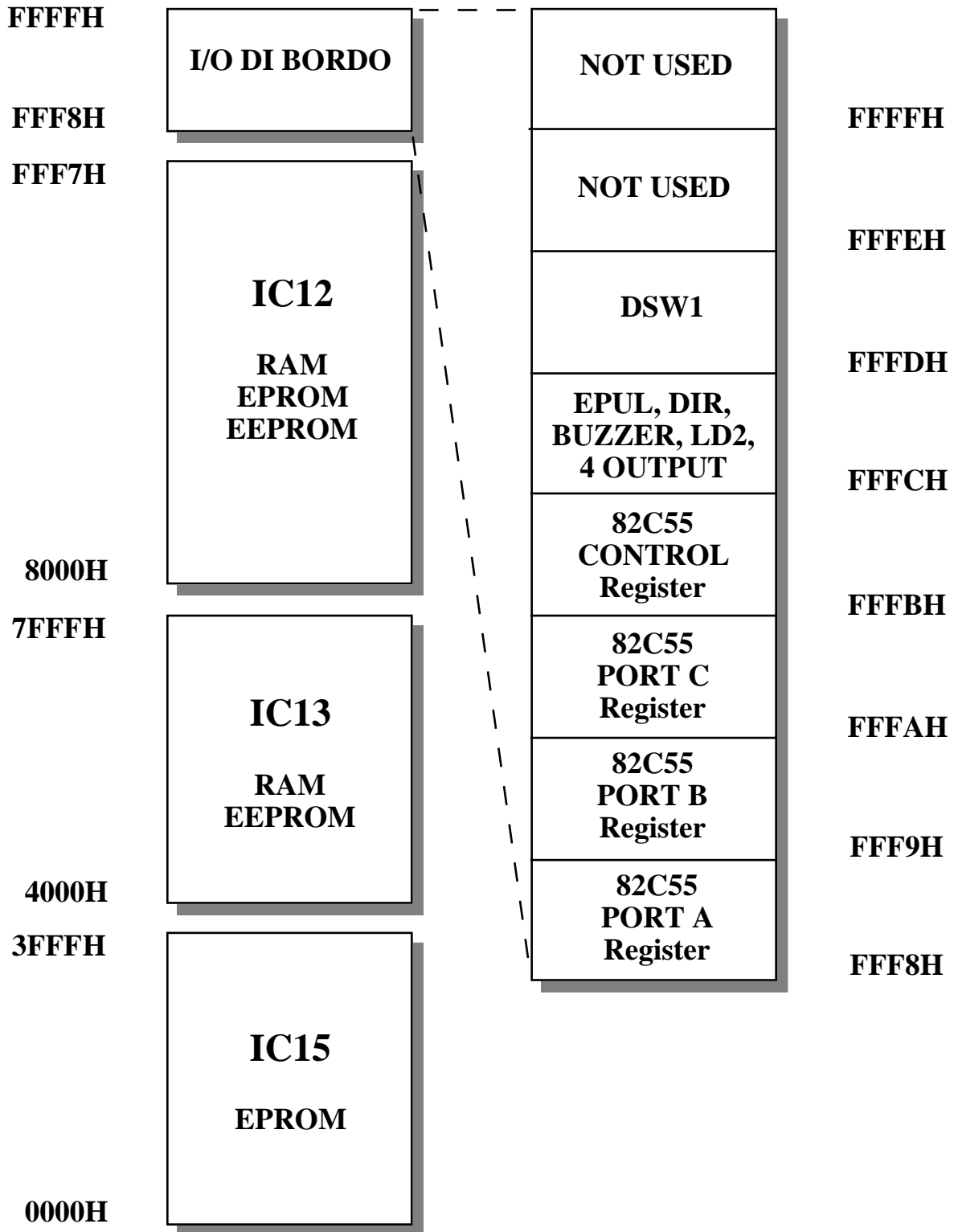


FIGURA 25: MAPPAGGIO DELLE MEMORIE IN MODO 2

Jumpers: J1 in posizione 2; J2 in posizione 2; J14(1,2,3) in posizione ASM
 Usato da pacchetti software come: MDP; HI TECH C; DDS C; ecc.



MAPPAGGIO 3

CODE and DATA AREA

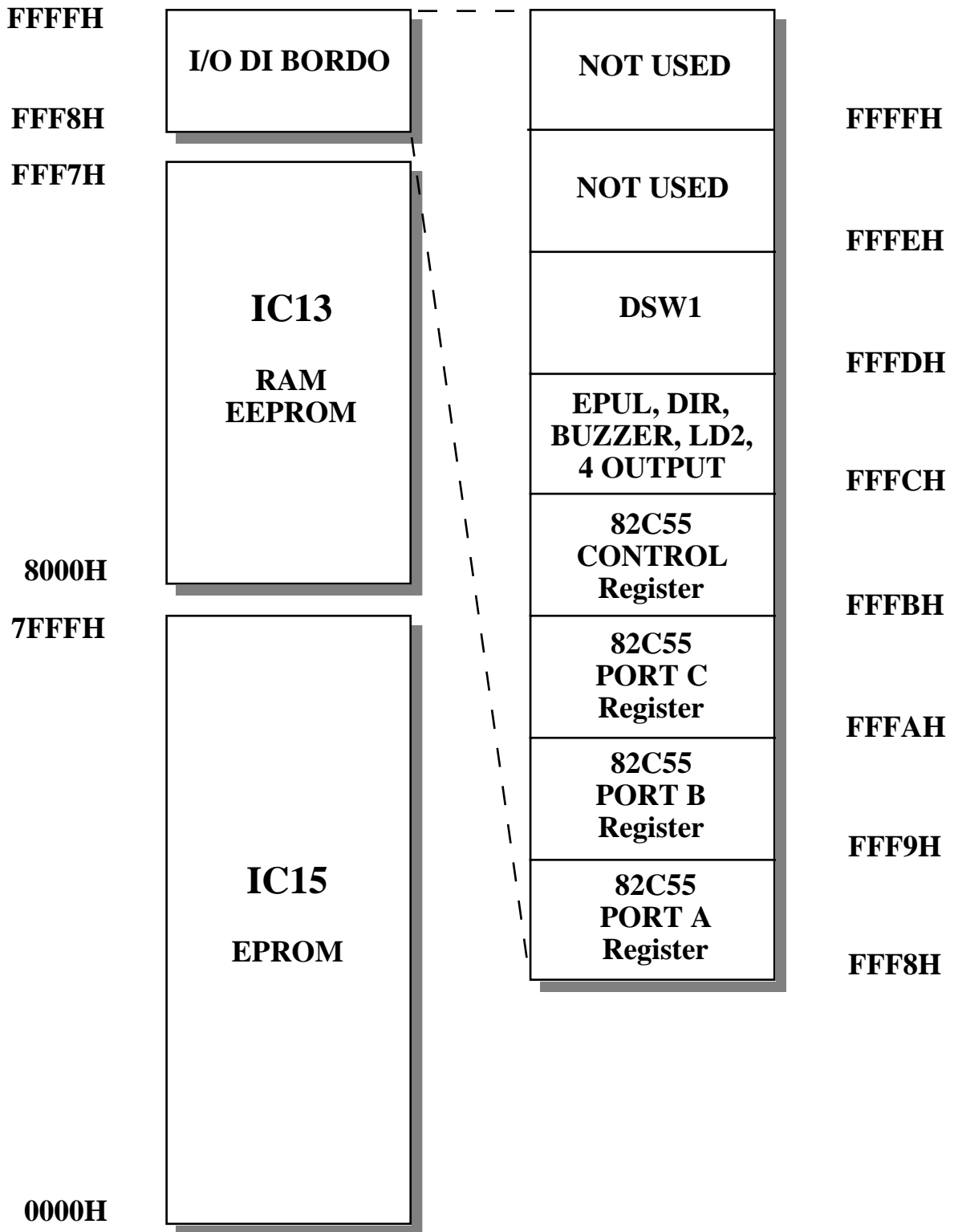


FIGURA 26: MAPPAGGIO DELLE MEMORIE IN MODO 3

Jumpers: J1 in posizione 1; J2 in posizione 1; J14(1,2,3) in posizione ASM
 Usato da pacchetti software come: HI TECH C; DDS C; ecc.

MAPPAGGIO 4

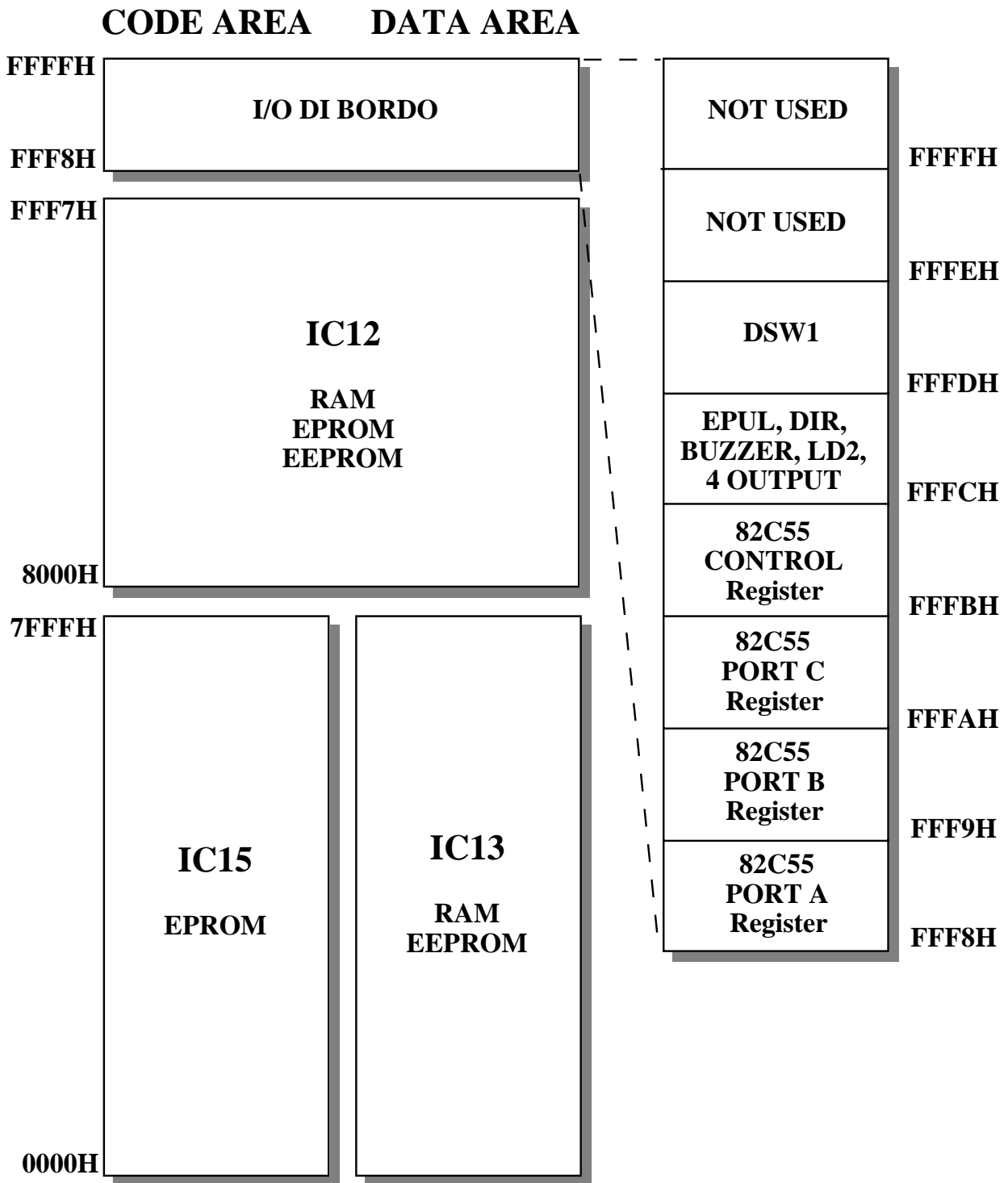


FIGURA 27: MAPPAGGIO DELLE MEMORIE IN MODO 4

Jumpers: J1 in posizione 1; J2 in posizione 1; J14(1,2,3) in posizione BASIC
 Usato da pacchetti software come: MCS BASIC 323; BXC51; HI TECH C; DDS C; ecc.



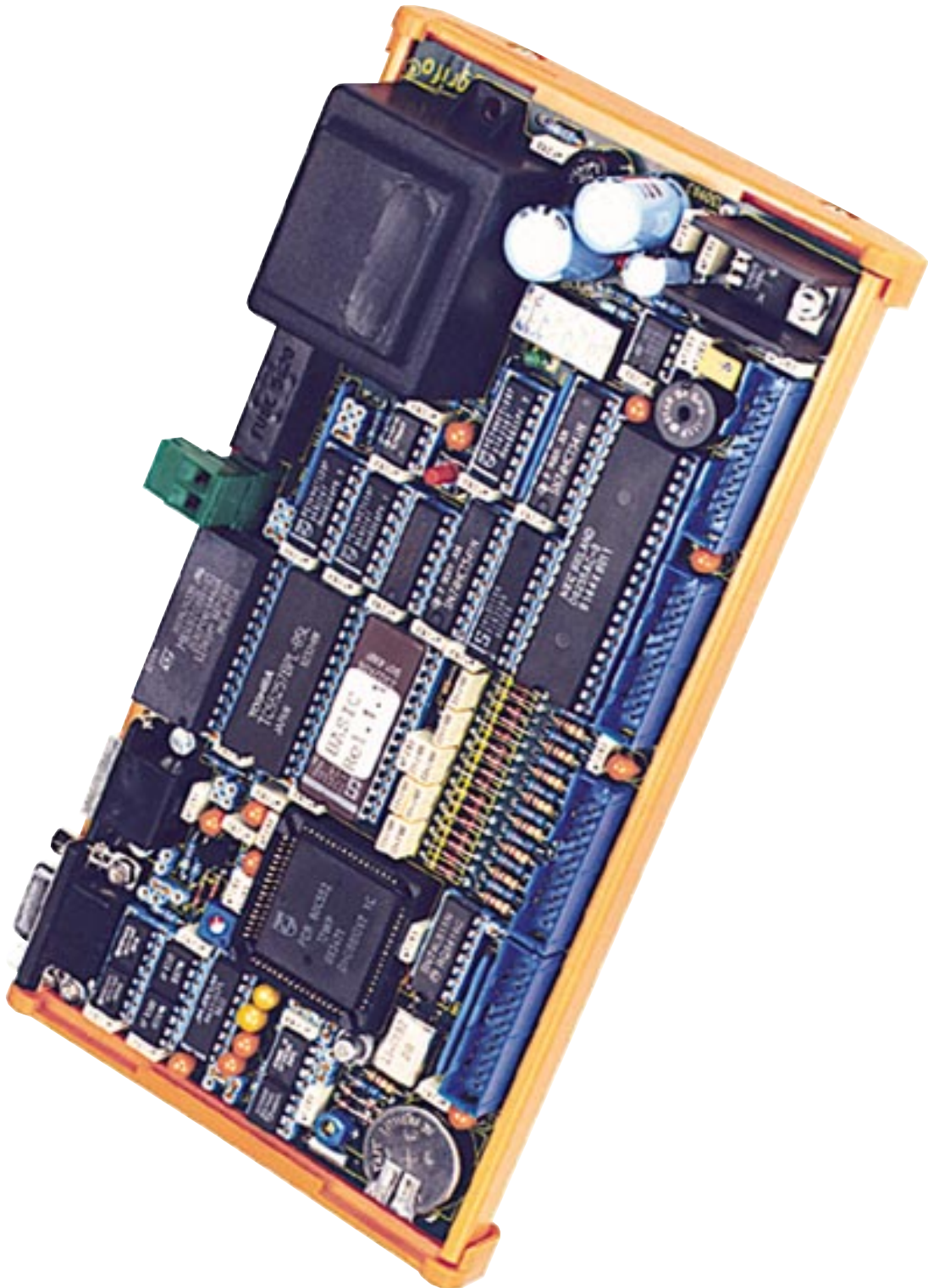


FIGURA 28: FOTO DELLA SCHEDA

DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO

Nel paragrafo precedente sono stati riportati gli indirizzi di allocazione di tutte le periferiche e di seguito viene riportata una descrizione dettagliata della funzione e del significato dei relativi registri (al fine di comprendere le successive informazioni, fare sempre riferimento alla tabella di indirizzamento I/O). Qualora la documentazione riportata fosse insufficiente fare riferimento direttamente alla documentazione tecnica della casa costruttrice del componente. Per quanto riguarda la programmazione delle sezioni della CPU si faccia riferimento all'appendice B di questo manuale.

BUZZER

Il buzzer, si attiva effettuando una operazione di output con D2=1, all'indirizzo di allocazione del registro BUZ. Logicamente la dissattivazione avviene tramite un'analogica operazione di output ma con il bit D2 resettato a 0. I rimanenti sette bits del registro BUZ devono essere settati tenendo conto della programmazione fornita loro precedentemente, onde evitare settaggi o reset di altre risorse associate ai bits di tale registro (si ricorda che i registri denominati EPUL, LD2, BUZ, DIR, OUTG in realtà corrispondono ad uno stesso indirizzo). Il registro BUZ è azzerato (tutti i bits a 0) in fase di Reset o power on, di conseguenza in seguito ad una di queste fasi il buzzer è disattivo.

Da ricordare che é possibile sapere se il buzzer é attivo o disattivo, effettuando una lettura del registro BUZ e mascherando il bit D2.

USCITE GENERICHE

La gestione delle 4 uscite generiche presenti sulla **GPC® 552** è effettuata tramite un registro di input/output denominato OUTG. I 4 bits più significativi che compongono tale registro, servono appunto per gestire le 4 uscite generiche (bit a 1= uscita a livello logico alto e viceversa):

D7 -> OUT GENERICA SUL PIN 10 DI CN1
D6 -> OUT GENERICA SUL PIN 9 DI CN1
D5 -> OUT GENERICA SUL PIN 12 DI CN1
D4 -> OUT GENERICA SUL PIN 11 DI CN1

I rimanenti 4 bits del registro OUTG devono essere settati tenendo conto della programmazione fornita loro precedentemente, onde evitare settaggi o reset di altre risorse associate ai bits di tale registro (si ricorda che i registri denominati EPUL, LD2, BUZ, DIR, OUTG in realtà corrispondono ad uno stesso indirizzo). Il registro OUTG è azzerato (tutti i bits a 0) in fase di Reset o power on, di conseguenza in seguito ad una di queste fasi le quattro uscite generiche sono disattive.

Da ricordare che é possibile sapere lo stato delle quattro uscite, effettuando una lettura del registro OUTG e mascherando i 4 bits più significativi.

LED DI ATTIVITÀ

Il LED di attività LD2, si attiva effettuando una operazione di output con D3=1, all'indirizzo di allocazione del registro LD2. Logicamente la dissattivazione avviene tramite un'analogica operazione di output ma con il bit D3 resettato a 0. I rimanenti sette bits del registro LD2 devono essere settati tenendo conto della programmazione fornita loro precedentemente, onde evitare settaggi o reset di altre risorse associate ai bits di tale registro (si ricorda che i registri denominati EPUL, LD2, BUZ, DIR, OUTG in realtà corrispondono ad uno stesso indirizzo). Il registro LD2 è azzerato (tutti i bits a 0) in fase di Reset o power on, di conseguenza in seguito ad una di queste fasi il led è disattivo. Da ricordare che é possibile sapere se il LED é attivo o disattivo, effettuando una lettura del registro LD2 e mascherando il bit D3.

DIP SWITCH DSW1

Il dip switch DSW1 montato a bordo della GPC®552 può essere acquisito via software, effettuando una semplice operazione di input all'indirizzo di allocazione del registro DSW1. La corrispondenza tra i bits del registro e le linee del dip switch è la seguente:

D7	->	DSW1.8
D6	->	DSW1.7
D5	->	DSW1.6
D4	->	DSW1.5
D3	->	DSW1.4
D2	->	DSW1.3
D1	->	DSW1.2
D0	->	DSW1.1

La combinazione è in logica negata, ovvero il dip in **ON** fornisce lo stato logico **0** al corrispondente bit, mentre il dip in **OFF** fornisce lo stato logico **1**.

DIREZIONALITÀ DELLA COMUNICAZIONE IN RS 422-485

La trasmissione si attiva effettuando una operazione di output con D1=1, all'indirizzo di allocazione del registro DIR. Logicamente la ricezione avviene tramite un'analogica operazione di output ma con il bit D1 resettato a 0. I rimanenti sette bits del registro DIR devono essere settati tenendo conto della programmazione fornita loro precedentemente, onde evitare settaggi o reset di altre risorse associate ai bits di tale registro (si ricorda che i registri denominati EPUL, LD2, BUZ, DIR, OUTG in realtà corrispondono ad uno stesso indirizzo). Il registro DIR è azzerato (tutti i bits a 0) in fase di Reset o power on, di conseguenza in seguito ad una di queste fasi il driver è in ricezione.

Da ricordare che è possibile sapere la direzione settata, effettuando una lettura del registro DIR e mascherando il bit D1.

SELEZIONE PULL UP/DOWN SULLE LINEE DI I/O DELLA SCHEDA

La selezione del pull up o pull down sulle linee di I/O si effettua con una operazione di output, utilizzando D0, all'indirizzo di allocazione del registro EPUL (J23 deve essere in posizione 2-3). Più specificatamente:

D0=0	-->	PULL DOWN attivo
D0=1	-->	PULL UP attivo

I rimanenti sette bits del registro EPUL devono essere settati tenendo conto della programmazione fornita loro precedentemente, onde evitare settaggi o reset di altre risorse associate ai bits di tale registro (si ricorda che i registri denominati EPUL, LD2, BUZ, DIR, OUTG in realtà corrispondono ad uno stesso indirizzo). Il registro EPUL è azzerato (tutti i bits a 0) in fase di Reset o power on, di conseguenza in seguito ad una di queste fasi le reti resistive si comportano come delle resistenze di pull-down.

Da ricordare che è possibile acquisire il tipo di settaggio delle reti resistive, effettuando una lettura del registro EPUL e mascherando il bit D0.

EEPROM SERIALE

Per quanto riguarda la gestione dell'eventuale modulo di EEPROM seriale (IC 24), si faccia riferimento alla documentazione specifica del componente. In questo manuale tecnico non viene riportata alcuna informazione software in quanto la modalità di gestione è articolata e prevede una conoscenza approfondita del componente e comunque l'utente può usare le apposite procedure ad alto livello fornite nel pacchetto di programmazione. Si ricorda solo che i primi 30 bytes (0...29) sono riservati e perciò si deve evitare la modifica dei medesimi.

Dal punto di vista elettrico la logica di controllo effettua le seguenti connessioni:

linea DATI (SDA)	-->	pin 27 (P3.3) della CPU
linea CLOCK (SCL)	-->	pin 29 (P3.5) della CPU

Data l'implementazione hardware della circuiteria di gestione del modulo di EEPROM seriale, si ricorda che di tale dispositivo i segnali A2,A1,A0 dello slave address sono rispettivamente posti a 1,0,0.

RAM TAMPONATA + RTC SERIALE

Per quanto riguarda la gestione dell'eventuale modulo di RAM+RTC seriale (IC 25), si faccia riferimento alla documentazione specifica del componente. In questo manuale tecnico non viene riportata alcuna informazione software in quanto la modalità di gestione è articolata e prevede una conoscenza approfondita del componente e comunque l'utente può usare le apposite procedure ad alto livello fornite nel pacchetto di programmazione. Si ricorda che in caso di montaggio di questo componente in IC 24 deve esserci al massimo una eeprom di 1024 bytes (24C08).

Dal punto di vista elettrico la logica di controllo effettua le seguenti connessioni:

linea DATI (SDA)	-->	pin 27 (P3.3) della CPU
linea CLOCK (SCL)	-->	pin 29 (P3.5) della CPU

Data l'implementazione hardware della circuiteria di gestione del modulo di RAM+RTC seriale, si ricorda che di tale dispositivo il segnale A0 dello slave address è posto a 0.

PPI 82C55

Questa periferica è vista in 4 registri: uno di stato (CNT) e tre dei dati (PDA, PDB, PDC) con cui si effettua la programmazione ed il comando della stessa. I registri dati sono utilizzati sia per operazioni di input (acquisizione linee dei port) che per quelle di output (settaggio linee dei port) ed ognuno di tali registri riporta i dati di I/O del corrispondente port. La periferica può operare in tre modi diversi: MODO 0 = Prevede due port bidirezionali da 8 bit (A,B) e due port bidirezionali da 4 bit (C LOW, C HIGH); gli ingressi non sono latched, mentre le uscite lo sono; nessun segnale di handshaking. MODO 1 = Prevede due port da 12 bit (A+C LOW, B+C HIGH) dove gli 8 bit dei port A e B costituiscono le linee di I/O, mentre i 4 bit del port C costituiscono le linee di handshaking. Gli ingressi e le uscite sono latched.

MODO 2 = Prevede un port da 13 bit (A+C3-7) dove gli 8 bit del port A costituiscono le linee di I/O, mentre i rimanenti 5 bit del port C costituiscono le linee di controllo. Un port da 11 bit (B+C0-2) dove gli 8 bit del port B costituiscono le linee di I/O ed i rimanenti 3 bit del port C costituiscono le linee di controllo. Sia gli ingressi che le uscite sono latched.

La programmazione della periferica avviene scrivendo una parola a 8 bit nel registro di controllo CNT, quando:

CNT = SF M1 M2 A CH M3 B CL

dove

SF = Se attivo (1) abilita il comando della periferica
M1 M2 = Selezionano il modo di funzionamento
0 0 = Selezione del modo 0
0 1 = Selezione del modo 1
1 X = Selezione del modo 2
A = Se attivo (1) setta il port A in input e viceversa
CH = Se attivo setta il nibble più significativo del port C in input e viceversa
M3 = Se attivo (1) seleziona modo 1, viceversa seleziona modo 0
B = Se attivo setta il port B in input e viceversa
CL = Se attivo setta il nibble meno significativo del port C in input e viceversa.

Dopo una fase di Reset o di power on il PPI 82C55 viene settato in modo 0 con tutti i port settati in input.

PWM

Fare riferimento all'apposita documentazione tecnica dell'appendice B.

UART

Fare riferimento all'apposita documentazione tecnica dell'appendice B.

A/D CONVERTER

Fare riferimento all'apposita documentazione tecnica dell'appendice B.

TIMER/COUNTER

Fare riferimento all'apposita documentazione tecnica dell'appendice B.

WATCH-DOG INTERNO

Fare riferimento all'apposita documentazione tecnica dell'appendice B.

SCHEDE ESTERNE PER GPC® 552

La scheda **GPC® 552** si interfaccia a buona parte dei moduli della serie BLOCK e di interfaccia utente. A titolo di esempio ne riportiamo un elenco con una breve descrizione delle caratteristiche di massima, per maggiori informazioni, richiedere la documentazione specifica:

OBI 01 - OBI 02

Opto BLOCK Input NPN-PNP

Interfaccia per 16 input optoisolati e visualizzati tipo NPN, PNP, connettore a morsettiera, connettore normalizzato **ABACO®** di I/O a 20 vie; sezione alimentatrice; attacco rapido per guide DIN 46277-1 e 3.

OBI N8 - OBI P8

Opto BLOCK Input NPN-PNP

Interfaccia per 8 input optoisolati e visualizzati tipo NPN, PNP, connettore a morsettiera, connettore normalizzato **ABACO®** di I/O a 20 vie; sezione alimentatrice; attacco rapido per guide DIN 46277-1 e 3.

TBO 01 - TBO 08

Transistor BLOCK Output

Interfaccia per 16 connettore normalizzato **ABACO®** di I/O a 20 vie; 16 o 8 output a transistor in Open Collector da 45 Vcc 3 A su connettore a morsettiera. Uscite optoisolate e visualizzate; attacco rapido per guide DIN 46277-1 e 3.

RBO 01

Relé BLOCK Output

Interfaccia per connettore normalizzato **ABACO®** di I/O a 20 vie; 8 output visualizzati con relé da 5 o 10 A (connettore a morsettiera); contatti in scambio (N.O. e N.C.); attacco rapido per guide DIN 46277-1 e 3.

RBO 08 - RBO 16

Relé BLOCK Output

Interfaccia per connettore normalizzato **ABACO®** di I/O a 20 vie; 8 o 16 output visualizzati con relé da 3 A con MOV; connettore a morsettiera; attacco rapido per guide DIN 46277-1 e 3.

XBI 01

miXed BLOCK Input-Output

Interfaccia tra 8 input + 8 output TTL (connettore normalizzato **ABACO®** di I/O a 20 vie), con 8 output a transistor in Open Collector da 45 Vcc 3 A + 8 input con filtro a Pi-Greco (connettore a morsettiera). I/O optoisolati e visualizzati; attacco rapido per guide DIN 46277-1 e 3.

XBI R4 - XBI T4

miXed BLOCK Input-Output

Interfaccia per connettore normalizzato **ABACO®** di I/O a 20 vie; 4 relé da 3 A con MOV o 4 transistor open collectors da 3 A optoisolati; 4 linee di input optoisolate; linee di I/O visualizzate; connettore a morsettiera; attacco rapido per guide DIN tipo C e guide Ω .

FBC 20 - FBC 120

Flat Block Contact 20 vie

Interfaccia tra 2 o 1 connettori a perforazione di isolante (scatolino da 20 vie maschi) e la filatura da campo (morsettiere a rapida estrazione). Attacco rapido per guide tipo DIN 46277-1 e 3.

IBC 01

Interface Block Comunication

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.

IAC 01

Interface Adapter Centronics

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**® e connettore a vaschetta D 25 vie femmina con pin out standard Centronics per la gestione di una stampante parallela.

IAF 42

Interface Adapter Futaba

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**® e connettore a scatolino a 20 vie con pin out standard per la gestione dei display fluorescenti della FUTABA.

IAF N42

Interface Adapter NEC

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**® e connettori a scatolino a 26 e 34 vie con pin out standard per la gestione dei display fluorescenti della NEC.

IAL 42

Interface Adapter LCD

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**® e connettore a scatolino a 14 vie con pin out standard per la gestione dei display fluorescenti della FUTABA.

DEB 01

Didactis Experimental Board

Scheda di supporto per l'utilizzo di 16 linee di I/O TTL. Comprende: 16 tasti; 16 LED; 4 digits; tastiera a matrice da 16 tasti; interfaccia per stampante Centronics, display LCD, display Fluorescente, connettore I/O **GPC**® 68; collegamento con il campo.

MCI 64

Memory Cards Interfaces 64 MBytes

Interfaccia per la gestione di Memory cards PCMCIA a 68 pins tramite un connettore normalizzato **ABACO**® ;sono disponibili linguaggi ad alto livello.

KDI L32 - KDI F32

Keyboard Display Interface 32 Key

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**® e tastiera a matrice da 32 tasti; 8 LED; Buzzer; display alfanumerico fluorescente FUTABA o LCD. Tasti a bassa corsa per pannelli a membrana.

KDI L33 + KDI F33

Keyboard Display Interface LCD TOSHIBA +

Keyboard Display Interface Fluorescent FUTABA

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**[®] e tastiera a matrice da 32 tasti; 8 LED; Buzzer; display LCD alfanumerico e grafico TOSHIBA (L33) o display alfanumerico fluorescente FUTABA (F33). Tasti a lunga corsa per utilizzo diretto; frame metallico.

KDI LT - KDI FF

Keyboard Display Interface LCD TOSHIBA e Fluorescent FUTABA

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**[®] e tastiera a matrice da 32 tasti; 8 LED; Buzzer; display LCD alfanumerico e grafico TOSHIBA; display fluorescente FUTABA. Tastiera e LED esterni.

APT 100

Analog PT 100

Interfaccia per 8 sensori PT100 a 3 fili su connettore normalizzato **ABACO**[®] di input analogico.

JKT 07 - JKT PTC

J, K Thermocoupled interface + PT 100

Interfaccia per 7 o 3 termocoppie J, K, S + 4 sensori PT100 a 3 fili su connettore normalizzato **ABACO**[®] di input analogico.

CBT 420

Current Block Transmitter 4÷20mA

Interfaccia tra 4 linee di input 0÷5, 0÷10 Vdc e 4 canali di output in corrente 4÷20mA; segnali su connettori a rapida estrazione; 14 bits di risoluzione; attacco rapido per guide tipo DIN 247277-1 e 3.

KDL 224 - KDL 424

Keyboard Display LCD

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**[®] e tastiera a matrice da 24 tasti; display LCD alfanumerico retroilluminato da 20 caratteri per 2 o per 4. Possibilità di collegare tastiere di tipo telefonico con matrice 3x4 e 4x4.

KDF 224

Keyboard Display FUTABA

Interfaccia tra 16 I/O TTL su connettore normalizzato **ABACO**[®] e tastiera a matrice da 24 tasti; display fluorescente alfanumerico da 20 caratteri per 2. Possibilità di collegare tastiere di tipo telefonico con matrice 3x4 e 4x4.

UAR 24

Universal Analog Regulator

Scheda periferica intelligente per l'acquisizione ed il controllo di 2 sensori di temperatura PT100 o termocoppie di tipo J, K, S, T; 4 relé di uscita da 3A; 2 uscite D/A da 12 bits da 0÷10 Vdc ciascuna; interfacciamento tramite BUS o tramite linea seriale (RS 232, RS 422-485 o Current Loop); sezione A/D con risoluzione da 16 bits + segno.

IPC 51

Intelligent Peripheral Controller

Scheda periferica intelligente per l'acquisizione di 8 sensori di temperatura PT100 o termocoppie di tipo J, K, S, T; interfacciamento tramite BUS o tramite linea seriale (RS 232, RS 422-485 o C. L.); sezione A/D con risoluzione 16 bits + segno; 5 o 8 conversioni per secondo; 0,1 gradi di risoluzione.

QTP 22

Quick Terminal Panel

Pannello utente intelligente equipaggiato con display LCD alfanumerico retroilluminato a LEDs da 40x1, 40x2 o 40x4 caratteri o da display fluorescente alfanumerico da 40x1, 40x2 o 40x4 caratteri; linea seriale settabile in RS 232, RS 422-485 o Current Loop; EEPROM seriale per il set-up e la gestione dei messaggi; possibilità di riconfigurare i 22 tasti e di cambiare il relativo nome sul pannello, inserendo la nuova etichetta nella relativa tasca di personalizzazione; 22 LEDs con l'attributo di blinking e Buzzer gestibili da software; alimentazione a bordo scheda (24 Vac); come opzione la scheda può disporre di un RTC, un lettore di carte magnetiche e di relé.

QTP 24

Quick Terminal Panel - 24 keys

Pannello utente intelligente equipaggiato con display LCD alfanumerico retroilluminato a LEDs da 20x2 o 20x4 caratteri o da display fluorescente alfanumerico da 20x2 caratteri; linea seriale settabile in RS 232, RS 422-485 o Current Loop; EEPROM seriale per il set-up e la gestione dei messaggi; possibilità di riconfigurare i 24 tasti e di cambiare il relativo nome sul pannello, inserendo la nuova etichetta nella relativa tasca di personalizzazione; 16 LEDs con l'attributo di blinking e Buzzer gestibili da software; alimentazione a bordo scheda; come opzione la scheda può disporre di un RTC, un lettore di carte magnetiche e di relé.

KDL X24 - KDF 224

Keyboard Display LCD 2,4 righe 24 tasti - Keyboard Display Fluorescent 2 righe 24 tasti
Interfaccia tra 16 I/O TTL su connettore normalizzato I/O ABACO® a 20 vie e tastiera a matrice esterna da 24 tasti; display alfanumerico fluorescente 20x 2 o LCD 20x2, 20x4 retroilluminato a LEDs. Predisposizione per collegamento a tastiera telefonica.

QTP 24P

Quick Terminal Panel 24 tasti con interfaccia Parallela

Interfaccia operatore provvista di display alfanumerico fluorescente 20x 2 o LCD 20x2, 20x4 retroilluminato a LEDs; tastiera a membrana da 24 tasti di cui 12 configurabili dall'utente; 16 LEDs di stato; alimentatore a bordo scheda in grado di pilotare anche carichi esterni; interdaccia parallela basata su 16 I/O TTL di un connettore normalizzato I/O ABACO® a 20 vie. Tasti ed etichette personalizzabili dall'utente tramite serigrafie da inserire in apposite tasche; opzione di contenitore metallico e plastico.

QTP G26

Quick Terminal Panel 26 tasti con LCD grafico

Interfaccia operatore provvista di display grafico da 240x128 pixel retroilluminato a LEDs; tastiera a membrana da 26 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.

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 **GPC 552**.

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 TEXAS INSTRUMENTS:	<i>Linear Circuits Data Book - Volumi 1 e 3</i>
Manuale NEC:	<i>Microprocessors and Peripherals - Volume 3</i>
Manuale NEC:	<i>Memory Products</i>
Manuale HEWLETT PACKARD:	<i>Optoelectronics Designer' s Catalog</i>
Manuale MAXIM:	<i>New Releases Data Book - Volume 4</i>
Manuale XICOR:	<i>Data Book</i>
Manuale PHILIPS:	<i>80C51 - Based 8-Bit Microcontrollers</i>
Manuale PHILIPS:	<i>IC12 - I²C bus.</i>
Manuale NATIONAL SEMICONDUCTOR:	<i>Linear Databook - Volume 1</i>

Per avere tutti gli aggiornamenti di tali manuali e di tutti i data-sheet fare riferimento ai siti in INTERNET delle case madri costruttrici.

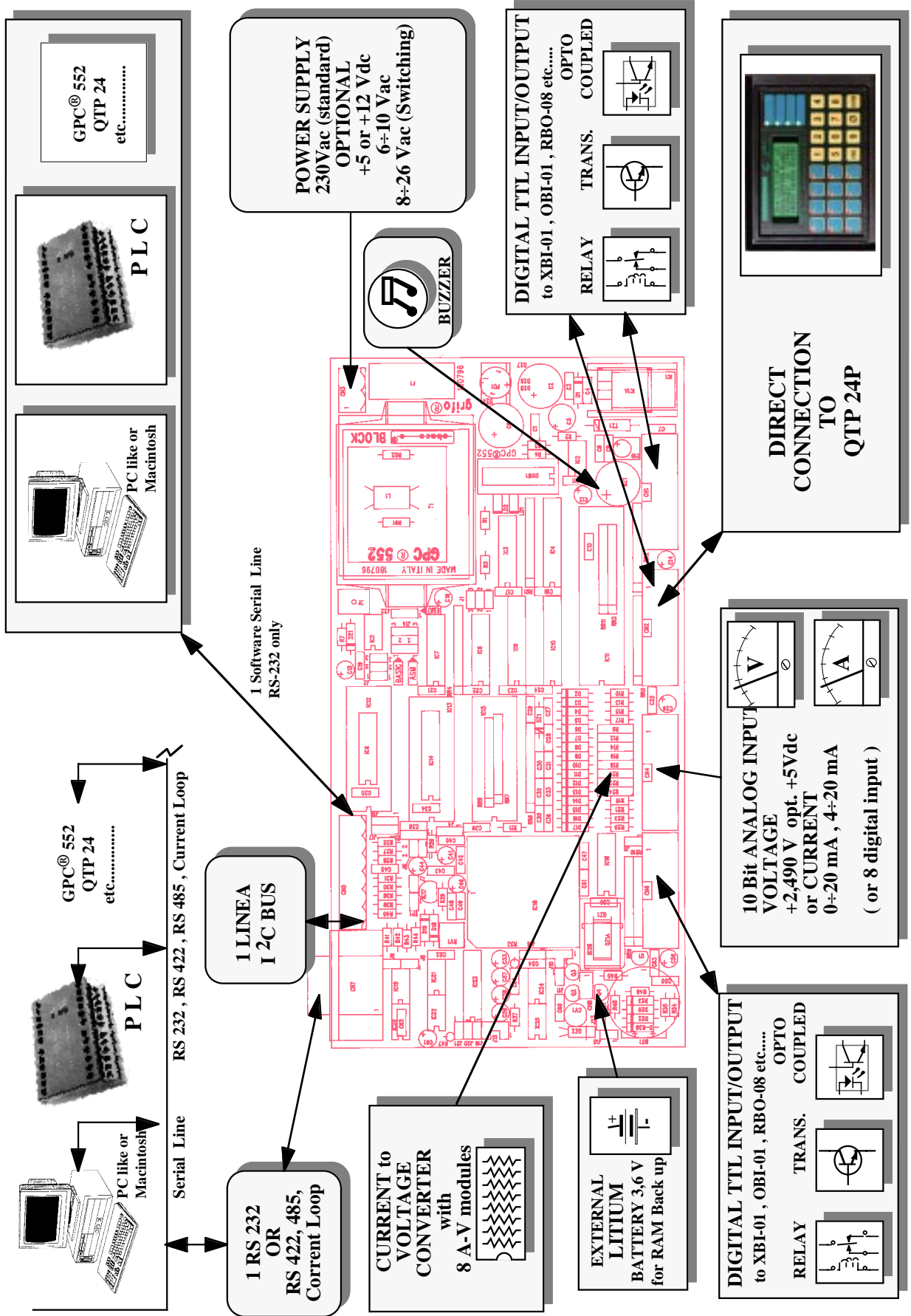


FIGURA 29: SCHEMA DELLE POSSIBILI ESPANSIONI



APPENDICE A: DISPOSIZIONE JUMPERS

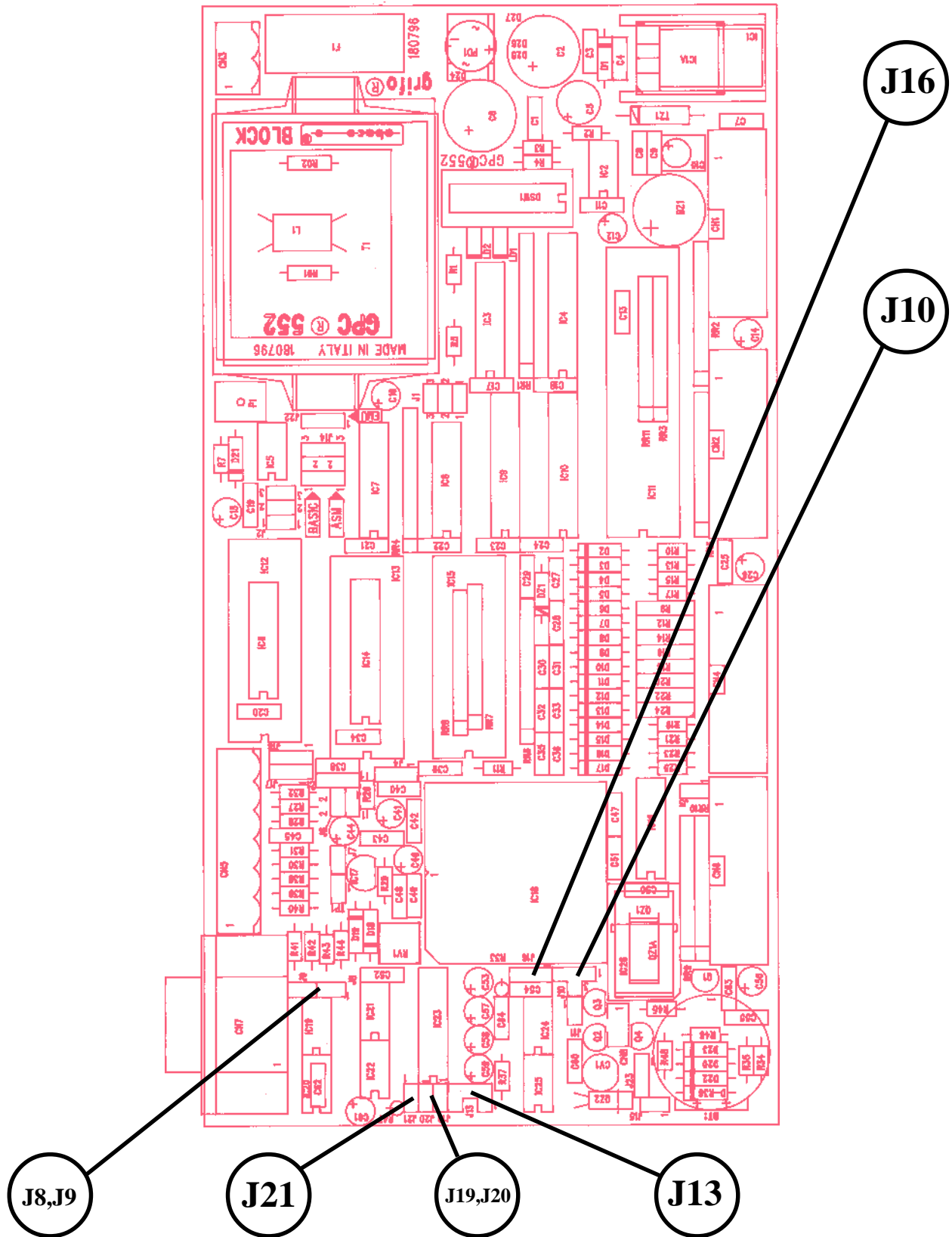


FIGURA A1: DISPOSIZIONE JUMPERS PER COMUNICAZIONE SERIALE



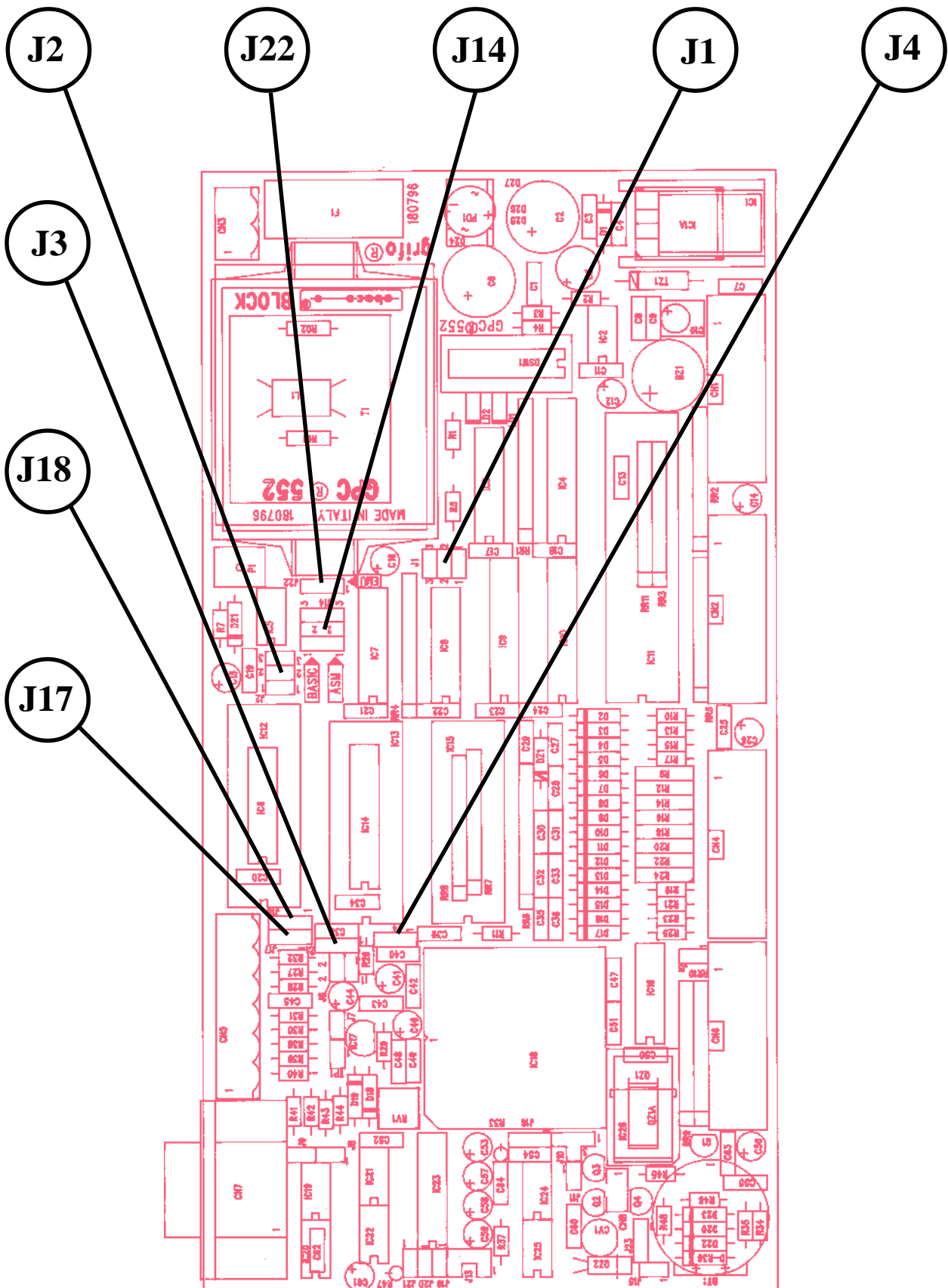


FIGURA A2: DISPOSIZIONE JUMPERS PER SELEZIONE MEMORIE

APPENDICE B: DESCRIZIONE COMPONENTI DI BORDO

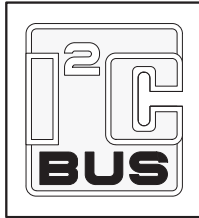
Philips Semiconductors

Product specification

Single-chip 8-bit microcontroller

80C552/83C552

Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM



DESCRIPTION

The 80C552/83C552 (hereafter generically referred to as 8XC552) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC552 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83C552—8k bytes mask programmable ROM
- 80C552—ROMless version of the 83C552
- 87C552—8k bytes EPROM (described in a separate chapter)

The 8XC552 contains a non-volatile 8k × 8 read-only program memory (83C552), a volatile 256 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I²C-bus), a “watchdog” timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC552 can be expanded using standard TTL compatible memories and logic.

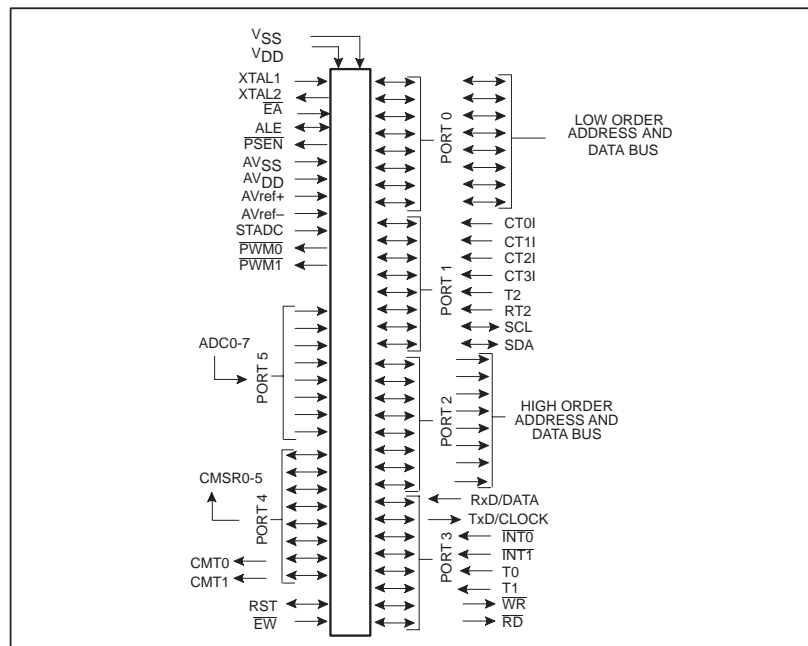
In addition, the 8XC552 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

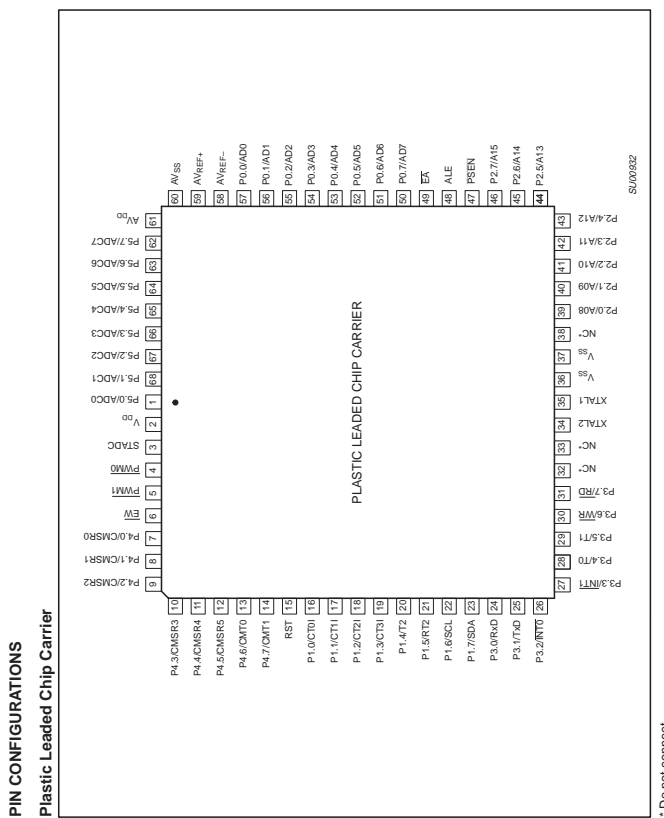
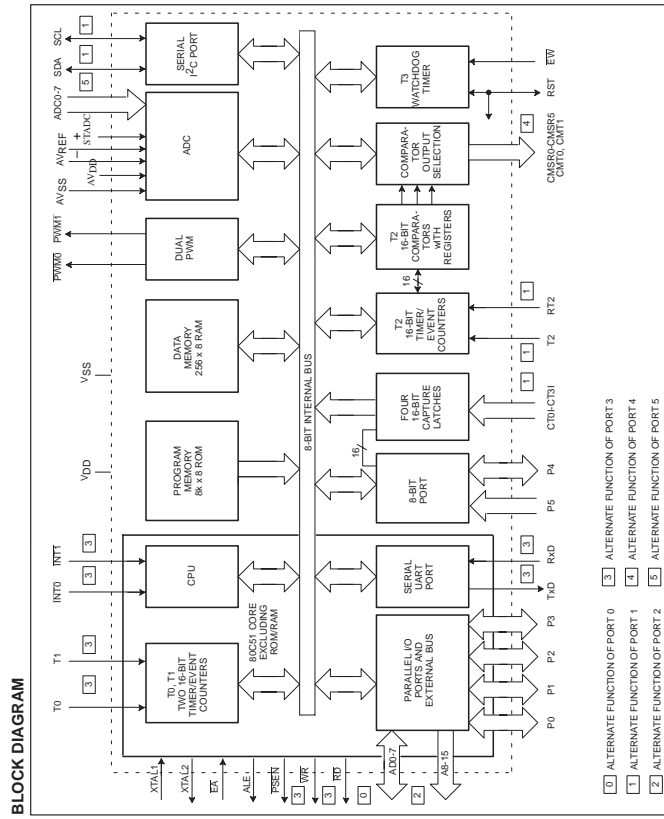
The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16MHz (24MHz) crystal, 58% of the instructions are executed in 0.75µs (0.5µs) and 40% in 1.5µs (1µs). Multiply and divide instructions require 3µs (2µs).

FEATURES

- 80C51 central processing unit
- 8k × 8 ROM expandable externally to 64k bytes
- ROM code protection
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- Two standard 16-bit timer/counters
- 256 × 8 RAM, expandable externally to 64k bytes
- Capable of producing eight synchronized, timed outputs
- A 10-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I²C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- Three speed ranges:
 - 3.5 to 16MHz
 - 3.5 to 24MHz (ROM, ROMless only)
 - 3.5 to 30MHz (ROM, ROMless only)
- Three operating ambient temperature ranges:
 - P83C552xBx: 0°C to +70°C
 - P83C552xFx: -40°C to +85°C (XTAL frequency max. 24 MHz)
 - P83C552xHx: -40°C to +125°C (XTAL frequency max. 16 MHz)

LOGIC SYMBOL





PIN CONFIGURATIONS

Plastic Leaded Chip Carrier

* Do not connect.



80C51 Family Derivatives

8XC552/562 overview

80C51 Family Derivatives

8XC552/562 overview

8XC552 OVERVIEW

The 8XC552 is a stand-alone high-performance microcontroller designed for use in real-time applications such as instrumentation, industrial control, and automotive control applications such as engine management and transmission control. The device provides, in addition to the 80C51 standard functions, a number of dedicated hardware functions for these applications.

The 8XC552 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC552 uses the powerful instruction set of the 80C51. Additional special function registers are incorporated to control the on-chip peripherals. Three versions of the derivative exist although the generic term '8XC552' is used to refer to family members:

- 83C552: 8k bytes mask-programmable ROM, 256 bytes RAM
- 87C552: 8k bytes EPROM, 256 bytes RAM
- 80C552: ROMless version of the 83C552

The 8XC552 contains a nonvolatile 8k x 8 read-only program memory, a volatile 256 x 8 read/write data memory, five 8-bit I/O ports and one 8-bit input port, two 16-bit time/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a fifteen-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I²C bus), a "watchdog" timer, and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC552 can be expanded using standard TTL compatible memories and logic.

The 8XC552 has two software selectable modes of reduced activity for further power reduction—Idle and Power-down. The Idle mode freezes the CPU and resets Timer T2 and the ADC and PWM circuitry but allows the other timers, RAM, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to become inoperative.

83C562 OVERVIEW

The 83C562 has been derived from the 8XC552 with the following changes:

- The SIO1 (I²C) interface has been omitted.
- The output of port lines P1.6 and P1.7 have a standard configuration instead of open drain.
- The resolution of the A/D converter is decreased from 10 bits to 8 bits.
- The resolution of the A/D converter is decreased from 50 machine cycles to 24 machine cycles.

All other functions, pinning and packaging are unchanged.

This chapter of the users' guide can be used for the 83C562 by omitting or changing the following:

- Disregard the description of SIO1 (I²C).
- The SFRs for the interfaces: S1ADR, S1DAT, S1STA, and S1CON are not implemented. The two SIO1 related flags ES1 in SFR1EN0 and PS1 in SFR1P0 are also not implemented. These two

flag locations are undefined after RESET. The interrupt vector for SIO1 is not used.

- Port lines P1.6 and P1.7 are not open drain but have the same standard configuration and electrical characteristics as P1.0-P1.5. Port lines P1.6 and P1.7 have alternative functions.
- The A/D converter has a resolution of 8 bits instead of 10 bits and consequently the two high-order bits 6 and 7 of SFR_ADCON are not implemented. These two locations are undefined after RESET. The 8-bit result of an A/D conversion is present in SFR_AIDCH. The result can always be calculated from the formula:

$$256 \frac{V_{IN}}{AV_{ref}} \frac{AV_{ref}}{AV_{ref}}$$

The A/D conversion time is 24 machine cycles instead of 50 machine cycles, and the sampling time is 6 machine cycles instead of 8 machine cycles. The conversion time takes 3 machine cycles per bit.

- The serial I/O function SIO0 and its SFRs SIOBUF and SIOCON are renamed to SIO, SBUF, and SCON. The interrupt related flags ES0 and PS0 are renamed ES and PS. Interrupt source S0 is renamed S. The serial I/O function remains the same.

Differences From the 80C51

Program Memory

The 8XC552 contains 8k bytes of on-chip program memory which can be extended to 64k bytes with external memories (see Figure 1). When the EA pin is held high, the 8XC552 fetches instructions from internal ROM unless the address exceeds 1FFFFH. Locations 2000H to FFFFH are fetched from external program memory. When the EA pin is held low, all instruction fetches are from external memory. ROM locations 0003H to 0073H are used by interrupt service routines.

Data Memory

The internal data memory is divided into 3 sections: the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128-byte special function register areas. The lower 128 bytes of RAM are directly and indirectly addressable. While RAM locations 128 to 255 and the special function register area share the same address space, they are accessed through different addressing modes. RAM locations 128 to 255 are only indirectly addressable, and the special function registers are only directly addressable. All other aspects of the internal RAM are identical to the 80C51.

The stack may be located anywhere in the internal RAM by loading the 8-bit stack pointer. Stack depth is 256 bytes maximum.

Special Function Registers

The special function registers (directly addressable only) contain all of the 8XC552 registers except the program counter and the four register banks. Most of the 56 special function registers are used to control the on-chip peripheral hardware. Other registers include arithmetic registers (ACC, B, PSW), stack pointer (SP), and data pointer registers (DHP, DPL). Sixteen of the SFRs contain 128 directly addressable bit locations. Table 1 lists the 8XC552's special function registers.

The standard 80C51 SFRs are present and function identically in the 8XC552 except where noted in the following sections.

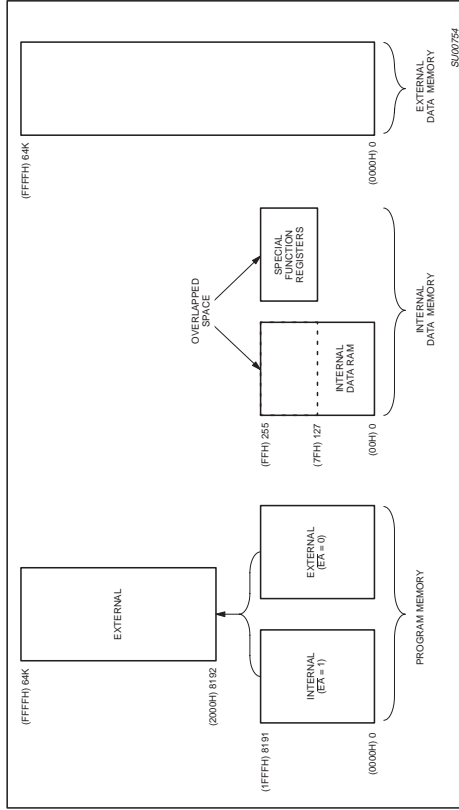


Figure 1. Memory Map

Timer T2

Timer T2 is a 16-bit timer consisting of two registers TM2H (HIGH byte) and TM2L (LOW byte). The 16-bit timer/counter can be switched off or clocked via a prescaler from one of two sources: f_{osc}/12 or an external signal. When Timer T2 is configured as a counter, the prescaler is clocked by an external signal on T2 (P1.4). A rising edge on T2 increments the prescaler, and the maximum repetition rate is one count per machine cycle (1MHz with a 12MHz oscillator).

The maximum repetition rate for Timer T2 is twice the maximum repetition rate for Timer 0 and Timer 1. T2 (P1.4) is sampled at f_{osc}/12 and again at SSP1 (i.e., twice per machine cycle). A rising edge is detected when T2 is LOW during one sample and HIGH during the next sample. To ensure that rising edge is detected, the input signal must be LOW for at least 12 cycles and then HIGH for at least 12 cycles. If a rising edge is detected before the end of SSP1, the timer will be incremented during the following cycle; otherwise it will be incremented one cycle later. The prescaler has a programmable division factor of 1, 2, 4, or 8 and is cleared if its division factor or input source is changed, or if the timer/counter is reset.

Timer T2 may be read "on the fly" but possesses no extra read latches, and software precautions may have to be taken to avoid misinterpretation in the event of an overflow from least to most significant byte while Timer T2 is being read. Timer T2 is not loadable and is reset by the RST signal or by a rising edge on the

input signal RT2. If enabled, RT2 is enabled by setting bit T2ER (TM2CON.5).

When the least significant byte of the timer overflows or when a 16-bit overflow occurs, an interrupt request may be generated. Either or both of these overflows can be programmed to request an interrupt. In both cases, the interrupt vector will be the same. When the lower byte (TM2L) overflows, flag T2BO (TM2CON) is set and flag T2OV (TM2IF) is set when TM2H overflows. These flags are set one cycle after an overflow occurs. Note that when T2OV is set, T2BO will also be set. To enable the byte overflow interrupt, bits ET2 (EN1.7) enable overflow interrupt, see Figure 2) and T2IS0 (TM2CON.6, byte overflow interrupt select) must be set. Bit TWB0 (TM2CON.4) is the Timer T2 byte overflow flag.

To enable the 16-bit overflow interrupt, bits ET2 (IE1.7, enable overflow interrupt) and T2S1 (TM2CON.7, 16-bit overflow interrupt select) must be set. Bit T2OV (TM2IF.7) is the Timer T2 16-bit overflow flag. All interrupt flags must be reset by software. To enable both byte and 16-bit overflow, T2IS0 and T2S1 must be set and two interrupt service routines are required. A test on the overflow flags indicates which routine must be executed. For each routine, only the corresponding overflow flag must be cleared.

Timer T2 may be reset by a rising edge on RT2 (P1.5) if the Timer T2 external reset enable bit (T2ER) in T2CON is set. This reset also clears the prescaler. In the Idle mode, the timer/counter and TM2CON special function register (see Figure 3).



Table 1. 8XC552 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION	LSB	RESET VALUE
ACC*	Accumulator	E0H	E7 E6 E5 E4 E3 E2 E1 E0	E0	00H
ADCH#	A/D converter high	C6H			xxxxxxxB
ADCON#	A/D control	C5H	ADC.1 ADC.0 ADEX ADC1 ADC2 ADC3 AADR2 AADR1 AADRO		xx000000B
B*	B register	F0H	F7 F6 F5 F4 F3 F2 F1 F0	F0	00H
CTCON#	Capture control	EBH	CTN3 CTN2 CTN1 CTP1 CTN0 CTP0		00H
CTH3#	Capture high 3	CFH			xxxxxxxB
CTH2#	Capture high 2	CEH			xxxxxxxB
CTH1#	Capture high 1	CDH			xxxxxxxB
CTH0#	Capture high 0	CCH			xxxxxxxB
CMH2#	Compare high 2	CBH			00H
CMH1#	Compare high 1	CAH			00H
CMH0#	Compare high 0	C9H			00H
CTL3#	Capture low 3	AFH			xxxxxxxB
CTL2#	Capture low 2	AEH			xxxxxxxB
CTL1#	Capture low 1	ADH			xxxxxxxB
CTL0#	Capture low 0	ACH			xxxxxxxB
CMIL2#	Compare low 2	ABH			00H
CMIL1#	Compare low 1	A9H			00H
CMIL0#	Compare low 0	A8H			00H
DPTR:	Data pointer				
DPH	Data pointer high (2 bytes)	83H			00H
DPL	Data pointer low	82H			00H
IE0#	Interrupt enable 0	A8H	AF AE AD AC AB AA A9 A8	A8	00H
IE1#	Interrupt enable 1	E8H	EE ED EC EB EA E9 E8	E8	00H
IP0#	Interrupt priority 0	B8H	BF BE BD BC BB BA B9 B8	B8	00H
IP1#	Interrupt priority 1	F8H	FF FE FD FC FB FA F9 F8	F8	00H
P5#	Port 5	C4H	ADC7 ADC6 ADC5 ADC4 ADC3 ADC2 ADC1 ADC0		xxxxxxxB
P4#	Port 4	C0H	CMT1 CMT0 CMSR5 CMSR4 CMSR3 CMSR2 CMSR1 CMSR0		FFH
P3#	Port 3	B0H	RD WR T1 T0 INTT INT0 TXD RXD		FFH
P2#	Port 2	A0H	A15 A14 A13 A12 A11 A10 A9 A8		FFH
P1#	Port 1	90H	97 96 95 94 93 92 91 90		FFH
P0#	Port 0	80H	87 86 85 84 83 82 81 80		FFH
PCON#	Power control	87H	AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0		00x0000B
PSW*	Program status word	D0H	SMOD - - - WALE GF1 GF0 PD IDL		00H

* SFRs are bit addressable.
SFRs are modified from or added to the 80C51 SFRs.

Table 1. 8XC552 Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION	LSB	RESET VALUE
PWMP#	PWM prescaler	FEH			00H
PWM1#	PWM register 1	FDH			00H
PWM0#	PWM register 0	FCH			00H
RTE#	Resettoggle enable	EFH	TP47 TP46 TP45 TP44 RP44 RP43 RP42 RP41		00H
SP	Stack pointer	81H			07H
SUBUF	Serial 0 data buffer	99H	9F 9E 9D 9C 9B 9A 99 98		xxxxxxxB
SOCN*	Serial 0 control	98H	SM0 SM1 SM2 REN TB8 RB8 TI RI		00H
S1ADR#	Serial 1 address	DBH			GC
S1DAT#	Serial 1 data	DAH			GC
S1STA#	Serial 1 status	D9H	SC4 SC3 SC2 SC1 SC0 0 0 0		0 F8H
S1CON#	Serial 1 control	D8H	DF DE DD DC DB DA D9 D8		00H
STE#	Set enable	EEH	CR2 EN51 STA ST0 SI AA CR1 CR0		00H
TH1	Timer high 1	8DH	TG47 TG46 SP45 SP44 SP43 SP42 SP41 SP40		00H
TH0	Timer high 0	8CH	GATE C/T M1 M0 GATE C/T M1 M0		00H
TL1	Timer low 1	8BH	8F 8E 8D 8C 8B 8A 89 88		00H
TL0	Timer low 0	8AH	TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0		00H
TMH2#	Timer high 2	8EH	T2S1 T2S0 T2ER T2B0 T2P1 T2P0 T2MS1 T2MS0		00H
TML2#	Timer low 2	EDH	CF CE CD CC CB CA C9 C8		00H
TMOD	Timer mode	89H	T20V CM2 CM1 CM0 CT13 CT12 CT11 CT10		00H
TOON*	Timer control	88H	GATE C/T M1 M0 GATE C/T M1 M0		00H
TM2CON#	Timer 2 control	EAH	8F 8E 8D 8C 8B 8A 89 88		00H
TM2IR#	Timer 2 int flag reg	C8H	TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0		00H
T3#	Timer 3	FFH	T2S1 T2S0 T2ER T2B0 T2P1 T2P0 T2MS1 T2MS0		00H

* SFRs are bit addressable.
SFRs are modified from or added to the 80C51 SFRs.

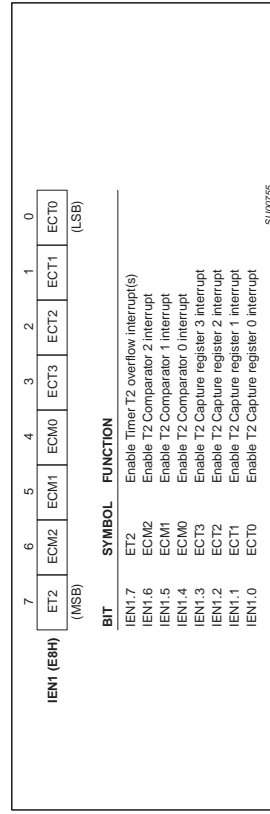
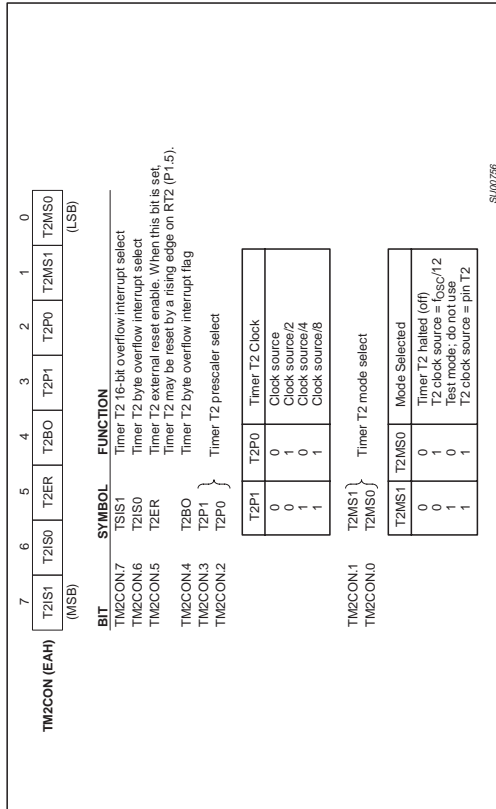


Figure 2. Timer T2 Interrupt Enable Register (IEN1)





SJ00756

Figure 3. T2 Control Register (TM2CON)

Timer T2 Extension: When a 12MHz oscillator is used, a 16-bit overflow on Timer T2 occurs every 65.5, 131, 262, or 524 ms, depending on the prescaler division ratio; i.e., the maximum cycle time is approximately 0.5 seconds, in applications where cycle times are greater than 0.5 seconds, it is necessary to extend Timer T2. This is achieved by selecting f_{osc}/12 as the clock source (set T2MS0, reset T2MS1), setting the prescaler division ratio to 1/8 (set T2P0, set T2P1), disabling the byte overflow interrupt (reset T2IS0) and enabling the 16-bit overflow interrupt (set T2IS1). The following software routine is written for a three-byte extension which gives a maximum cycle time of approximately 2400 hours.

```
OVINT: PUSH ACC ;save accumulator
      PUSH PSW ;save status
      INC R0 ;increment first byte (low order)
      :of extended timer
MOV A, TMEX1
JNZ INTEX ;jump to INTEX if there is no overflow
INC A
MOV A, TMEX2
JNZ INTEX ;jump to INTEX if there is no overflow
INC A
MOV A, TMEX3
JNZ INTEX ;increment third byte (high order)
INC A
INTEX: CLR T2OV ;reset interrupt flag
      PSW ;restore status
      POP ACC ;restore accumulator
      RETI ;return from interrupt
```

Timer T2, Capture and Compare Logic: Timer T2 is connected to four 16-bit capture registers and three 16-bit compare registers. A capture register may be used to capture the contents of Timer T2 when a transition occurs on its corresponding input pin. A compare register may be used to set, reset, or toggle port-4 output pins at certain pre-programmable time intervals.

The combination of Timer T2 and the capture and compare logic is very powerful in applications involving rotating machinery, automotive injection systems, etc. Timer T2 and the capture and compare logic are shown in Figure 4.

Capture Logic: The four 16-bit capture registers that Timer T2 is connected to are: CT0, CT1, CT2, and CT3. These registers are loaded with the contents of Timer T2, and an interrupt is requested upon receipt of the input signals CT0, CT1, CT2, or CT3. These input signals are shared with port 4. The four interrupt flags are in the Timer T2 interrupt register (TM2IR special function register). If the capture facility is not required, these inputs can be regarded as additional external interrupt inputs.

Using the capture control register CTCON (see Figure 5), these inputs may capture on a rising edge, a falling edge, or on either a rising or falling edge. The inputs are sampled during S1P1 of each cycle. When a selected edge is detected, the contents of Timer T2 are captured at the end of the cycle.

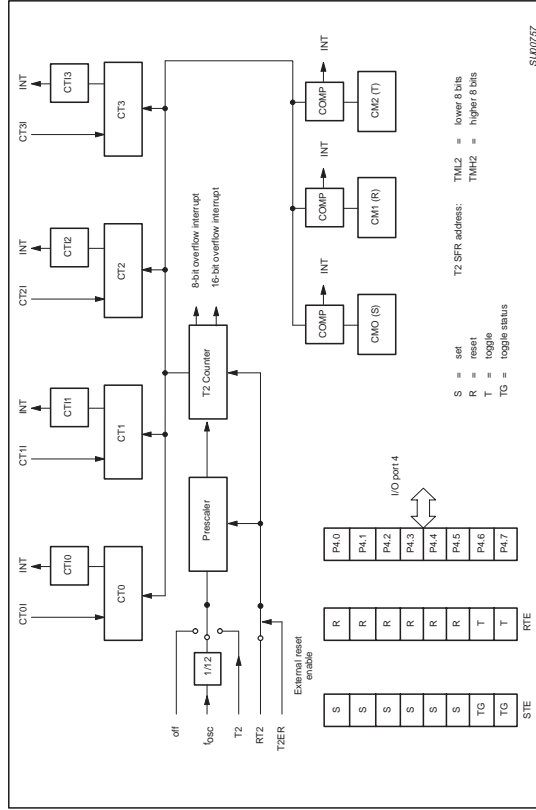


Figure 4. Block Diagram of Timer 2

Measuring Time Intervals Using Capture Registers: When a recurring external event is represented in the form of rising or falling edges on one of the four capture pins, the time between two events can be measured using Timer T2 and a capture register. When an event occurs, the contents of Timer T2 are copied into the relevant capture register and an interrupt request is generated. The interrupt service routine may then compute the interval time if it knows the previous contents of Timer T2 when the last event occurred. With a 12MHz oscillator, Timer T2 can be programmed to overflow every 524ms. When event interval times are shorter than this, computing the interval time is simple, and the interrupt service routine is short. For longer interval times, the Timer T2 extension routine may be used.

Compare Logic: Each time Timer T2 is incremented, the contents of the three 16-bit compare registers CM0, CM1, and CM2 are compared with the new counter value of Timer T2. When a match is found, the corresponding interrupt flag in TM2IR is set at the end of the following cycle. When a match with CM0 occurs, the controller sets bits 0-5 of port 4 if the corresponding bits of the set enable register STE are at logic 1.

When a match with CM1 occurs, the controller resets bits 0-5 of port 4 if the corresponding bits of the reset/toggle enable register RTE are at logic 1 (see Figure 6 for RTE register function). If RTE is "0", then P4.n is not affected by a match between CM1 or CM2 and Timer 2. When a match with CM2 occurs, the controller "toggles" bits 6 and 7 of port 4 if the corresponding bits of the RTE are at logic 1. The port latches of bits 6 and 7 are not toggled.

Two additional flip-flops store the last operation, and it is these flip-flops that are toggled.

Thus, if the current operation is "set," the next operation will be "reset" even if the port latch is set by software before the "reset" operation occurs. The first "toggle" after a chip RESET will set the port latch. The contents of these two flip-flops can be read at STE 6 and STE 7 (corresponding to P4.6 and P4.7, respectively). Bits STE 6 and STE 7 are read only (see Figure 7 for STE register function). A logic 0 indicates that the next toggle will set the port latch; a logic 1 indicates that the next toggle will reset the port latch. CM0, CM1, and CM2 are reset by the RST signal.

The modified port latch information appears at the port pin during SSP1 of the cycle following the cycle in which a match occurred. If the port is modified by software, the outputs change during S1P1 of the following cycle. Each port 4 bit can be set or reset by software at any time. A hardware modification resulting from a comparator match takes precedence over a software modification in the same cycle. When the comparator results require a "set" and a "reset" at the same time, the port latch will be reset.

Timer T2 Interrupt Flag Register TM2IR: Eight of the nine Timer T2 interrupt flags are located in special function register TM2IR (see Figure 8). The ninth flag is TM2CON.4.

The CT0 and CT1 flags are set during S4 of the cycle in which the contents of Timer T2 are captured. CT0 is scanned by the interrupt logic during S2, and CT1 is scanned during S3. CT2 and CT3 are set during S6 and are scanned during S4 and S5. The associated



80C51 Family Derivatives

8XC552/562 overview

interrupt requests are recognized during the following cycle. If these flags are polled, a transition at CT01 or CT11 will be recognized one cycle before a transition on CT21 or CT31 since registers are read during S5. The CM0, CM1, and CM2 flags are set during S6 of the cycle following a match. CM0 is scanned by the interrupt logic during S2; CM1 and CM2 are scanned during S3 and S4. A match will be recognized by the interrupt logic (or by polling the flags) two cycles after the match takes place.

CTCON (EBH)		CAPTURE/INTERRUPT ON:								CTP0		
(MSB)		7	6	5	4	3	2	1	0	(LSB)		
BIT	SYMBOL	FUNCTION										
CTCON.7	CTN3	Capture Register 3 triggered by a falling edge on CT31										
CTCON.6	CTN3	Capture Register 3 triggered by a rising edge on CT31										
CTCON.5	CTN2	Capture Register 2 triggered by a falling edge on CT21										
CTCON.4	CTP2	Capture Register 2 triggered by a rising edge on CT21										
CTCON.3	CTN1	Capture Register 1 triggered by a falling edge on CT11										
CTCON.2	CTP1	Capture Register 1 triggered by a rising edge on CT11										
CTCON.1	CTN0	Capture Register 0 triggered by a falling edge on CT01										
CTCON.0	CTP0	Capture Register 0 triggered by a rising edge on CT01										

Figure 5. Capture Control Register (CTCON)

RTE (EFH)		RESET/TOGGLE ENABLE REGISTER (RTE)								RTE0		
(MSB)		7	6	5	4	3	2	1	0	(LSB)		
BIT	SYMBOL	FUNCTION										
RTE.7	TP47	If "1" then P4.7 toggles on a match between CM1 and Timer T2										
RTE.6	TP46	If "1" then P4.6 toggles on a match between CM1 and Timer T2										
RTE.5	RP45	If "1" then P4.5 is reset on a match between CM1 and Timer T2										
RTE.4	RP44	If "1" then P4.4 is reset on a match between CM1 and Timer T2										
RTE.3	RP43	If "1" then P4.3 is reset on a match between CM1 and Timer T2										
RTE.2	RP42	If "1" then P4.2 is reset on a match between CM1 and Timer T2										
RTE.1	RP41	If "1" then P4.1 is reset on a match between CM1 and Timer T2										
RTE.0	RP40	If "1" then P4.0 is reset on a match between CM1 and Timer T2										

Figure 6. Reset/Toggle Enable Register (RTE)

STE (EBH)		SET ENABLE REGISTER (STE)								STE0		
(MSB)		7	6	5	4	3	2	1	0	(LSB)		
BIT	SYMBOL	FUNCTION										
STE.7	TG47	Toggle flip-flops										
STE.6	TG46	Toggle flip-flops										
STE.5	SP45	If "1" then P4.5 is set on a match between CM0 and Timer T2										
STE.4	SP44	If "1" then P4.4 is set on a match between CM0 and Timer T2										
STE.3	SP43	If "1" then P4.3 is set on a match between CM0 and Timer T2										
STE.2	SP42	If "1" then P4.2 is set on a match between CM0 and Timer T2										
STE.1	SP41	If "1" then P4.1 is set on a match between CM0 and Timer T2										
STE.0	SP40	If "1" then P4.0 is set on a match between CM0 and Timer T2										

Figure 7. Set Enable Register (STE)

80C51 Family Derivatives

8XC552/562 overview

TM2IR (C8H)		Interrupt Flag Register (TM2IR)								CTI0		
(MSB)		7	6	5	4	3	2	1	0	(LSB)		
BIT	SYMBOL	FUNCTION										
TM2IR.7	TZOV	Timer T2 16-bit overflow interrupt flag										
TM2IR.6	CM2	CM2 interrupt flag										
TM2IR.5	CM1	CM1 interrupt flag										
TM2IR.4	CM0	CM0 interrupt flag										
TM2IR.3	CTI3	CT3 interrupt flag										
TM2IR.2	CTI2	CT2 interrupt flag										
TM2IR.1	CTI1	CT1 interrupt flag										
TM2IR.0	CTI0	CT0 interrupt flag										

IP1 (FBH)		Timer T2 Interrupt Priority Register (IP1)								PCT0		
(MSB)		7	6	5	4	3	2	1	0	(LSB)		
BIT	SYMBOL	FUNCTION										
IP1.7	PT2	Timer T2 overflow interrupt(s) priority level										
IP1.6	PCM2	Timer T2 comparator 2 interrupt priority level										
IP1.5	PCM1	Timer T2 comparator 1 interrupt priority level										
IP1.4	PCM0	Timer T2 comparator 0 interrupt priority level										
IP1.3	PCT3	Timer T2 capture register 3 interrupt priority level										
IP1.2	PCT2	Timer T2 capture register 2 interrupt priority level										
IP1.1	PCT1	Timer T2 capture register 1 interrupt priority level										
IP1.0	PCT0	Timer T2 capture register 0 interrupt priority level										

Figure 8. Interrupt Flag Register (TM2IR) and Timer T2 Interrupt Priority Register (IP1)

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the microcontroller if it enters erroneous processor states (possibly caused by electrical noise or RFI) within a reasonable period of time. An analogy is the "dead man's handle" in railway locomotives. When enabled, the watchdog circuitry will generate a system reset if the user program fails to reload the watchdog timer within a specified length of time known as the "watchdog interval."

Watchdog Circuit Description: The watchdog timer (Timer T3) consists of an 8-bit timer with an 11-bit prescaler as shown in Figure 9. The prescaler is fed with a signal whose frequency is 1/12 the oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "1" seconds, where:

$$t = 12 \times 2048 \times 1/f_{osc}$$

$$(t = 1.5ms \text{ at } f_{osc} = 16MHz, = 1ms \text{ at } f_{osc} = 24MHz)$$

If the 8-bit timer overflows, a short internal reset pulse is generated which will reset the 8XC552. A short output reset pulse is also generated at the RST pin. This short output pulse (3 machine cycles) may be destroyed if the RST pin is connected to a capacitor. This would not, however, affect the internal reset operation.

Watchdog operation is activated when external pin EW0 is tied low. When EW0 is tied low, it is impossible to disable the watchdog operation by software.

How to Operate the Watchdog Timer: The watchdog timer has to be reloaded within periods that are shorter than the programmed watchdog interval, otherwise the watchdog timer will overflow and a system reset will be generated. The user program must therefore continually execute sections of code which reload the watchdog timer. The period of time elapsed between execution of these sections of code must never exceed the watchdog interval. When using a 16MHz oscillator, the watchdog interval is programmable between 1.5ms and 392ms. When using a 24MHz oscillator, the watchdog interval is programmable between 1ms and 255ms.

In order to prepare software for watchdog operation, a programmer should first determine how long his system can sustain an erroneous processor state. The result will be the maximum watchdog interval. As the maximum watchdog interval becomes shorter, it becomes more difficult for the programmer to reload the watchdog interval, and thus it becomes more difficult to implement watchdog operation.

The programmer must now partition the software in such a way that reloading of the watchdog is carried out in accordance with the above requirements. The programmer must determine the execution times of all software modules. The effect of possible conditional branches, subroutines, external and internal interrupts must all be taken into account. Since it may be very difficult to evaluate the execution times of some sections of code, the programmer should use worst case estimations. In any event, the programmer must make sure that the watchdog is not activated during normal operation.



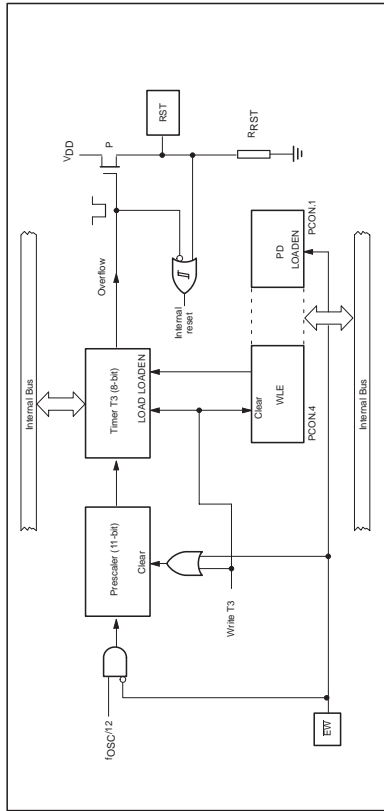


Figure 9. Watchdog Timer

The watchdog timer is reloaded in two stages in order to prevent erroneous software from reloading the watchdog. First PCON.4 (WLE) must be set. The T3 may be loaded. When T3 is loaded, PCON.4 (WLE) is automatically reset. T3 cannot be loaded if it is called frequently. Since Timer T3 is an up-counter, a reload value of 00H gives the maximum watchdog interval (510ms with a 12MHz oscillator), and a reload value of 0FFH gives the minimum watchdog interval (2ms with a 12MHz oscillator).

In the idle mode, the watchdog circuitry remains active. When watchdog operation is implemented, the power-down mode cannot be used since both states are contradictory. Thus, when watchdog operation is enabled by tying external pin EW low, it is impossible to enter the power-down mode, and an attempt to set the power-down bit (PCON.1) will have no effect. PCON.1 will remain at logic 0.

During the early stages of software development/debugging, the watchdog may be disabled by tying the EW pin high. At a later stage, EW may be tied low to complete the debugging process.

Watchdog Software Example: The following example shows how watchdog operation might be handled in a user program.

```

at the program start:
T3      EQU 0FFH ;address of watchdog timer T3
PCON    EQU 087H ;address of PCON SFR
WATCHH-INTV EQU 156 ;watchdog interval (e.g., ~2x100ms)
;to be inserted at each watchdog reload location within
;the user program:
LCALL WATCHDOG
;watchdog service routine:
WATCHDOG: ORL  PCON,#00H ;load condition flag (PCON.4)
           MOV  T3,WATCHH-INTV ;load T3 with watchdog interval
           RET
    
```

The CPU interfaces to the I²C logic via the following four special function registers: SICON (SIO1 control register), SISTA (SIO1 status register), S1DAT (SIO1 data register), and S1ADR (SIO1 slave address register). The SIO1 logic interfaces to the external I²C bus via two port 1 pins: P1.6/SCL (serial clock line) and P1.7/SDA (serial data line).

A typical I²C bus configuration is shown in Figure 10, and Figure 11 shows how a data transfer is accomplished on the bus. Depending on the state of the direction bit (RW), two types of data transfers are possible:

1. Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.
2. Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a "not acknowledge" is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the I²C bus will not be released.

Modes of Operation: The on-chip SIO1 logic may operate in the following four modes:

1. Master Transmitter Mode:
 - Serial data output through P1.7/SDA while P1.6/SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this case the data direction bit (RW) will be logic 0, and we say that a "W" is transmitted. Thus the first byte transmitted is S1A+W. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.
2. Master Receiver Mode:
 - The first byte transmitted contains the slave address of the transmitting device (7 bits) and the data direction bit. In this case the data direction bit (RW) will be logic 1, and we say that an "R" is transmitted. Thus the first byte transmitted is S1A+R. Serial data is received via P1.7/SDA while P1.6/SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.
3. Slave Receiver Mode:
 - Serial data and the serial clock are received through P1.7/SDA and P1.6/SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

4. Slave Transmitter Mode:

The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via P1.7/SDA while the serial clock is input through P1.6/SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

In a given application, SIO1 may operate as a master and as a slave. In the slave mode, the SIO1 hardware looks for its own slave address and the general call address. If one of these addresses is detected, an interrupt is requested. When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the master mode, SIO1 switches to the slave mode immediately and can detect its own slave address in the same serial transfer.

SIO1 Implementation and Operation: Figure 12 shows how the on-chip I²C bus interface is implemented, and the following text describes the individual blocks.

INPUT FILTERS AND OUTPUT STAGES

The input filters have I²C compatible input levels. If the input voltage is less than 1.5V, the input logic level is interpreted as 0; if the input voltage is greater than 3.0V, the input logic level is interpreted as 1. Input signals are synchronized with the internal clock (f_{OSC/4}), and spikes shorter than three oscillator periods are filtered out.

The output stages consist of open drain transistors that can sink 3mA at V_{OUT} < 0.4V. These open drain outputs do not have clamping diodes to V_{DD}. Thus, if the device is connected to the I²C bus and V_{DD} is switched off, the I²C bus is not affected.

ADDRESS REGISTER, S1ADR

This 8-bit special function register may be loaded with the 7-bit slave address (7 most significant bits) to which SIO1 will respond when programmed as a slave transmitter or receiver. The LSB (GC) is used to enable general call address (00H) recognition.

COMPARATOR

The comparator compares the received 7-bit slave address with its own slave address (7 most significant bits in S1ADR). It also compares the first received 8-bit byte with the general call address (00H). If an equality is found, the appropriate status bits are set and an interrupt is requested.

SHIFT REGISTER, S1DAT

This 8-bit special function register contains a byte of serial data to be transmitted or a byte which has just been received. Data in S1DAT is always shifted from right to left; the first bit to be transmitted is the MSB (bit 7) and after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.



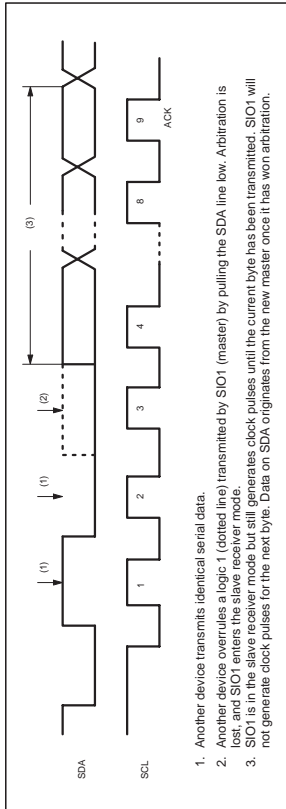


Figure 13. Arbitration Procedure

1. Another device transmits identical serial data.
2. Another device overrides a logic 1 (dotted line) transmitted by SIO1 (master) by pulling the SDA line low. Arbitration is lost and SIO1 enters the slave receiver mode.
3. SIO1 and SIO0 generate clock pulses until the current byte has been transmitted. SIO1 will not generate clock pulses for the next byte. Data on SDA originates from the new master once it has won arbitration.

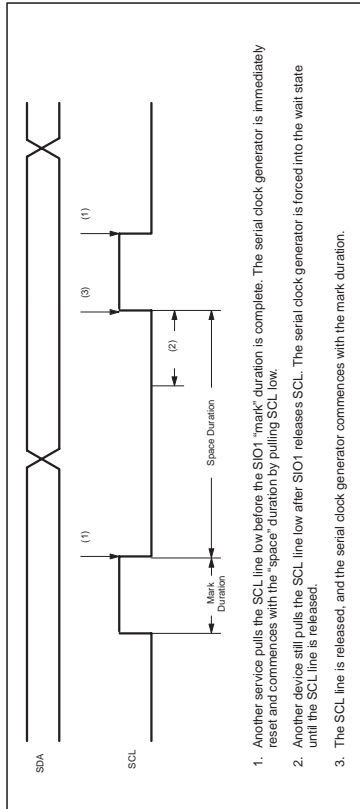


Figure 14. Serial Clock Synchronization

1. Another service pulls the SCL line low before the SIO1 "mark" duration is complete. The serial clock generator is immediately reset and commences with the "space" duration by pulling SCL low.
2. Another device still pulls the SCL line low after SIO1 releases SCL. The serial clock generator is forced into the wait state until the SCL line is released.
3. The SCL line is released, and the serial clock generator commences with the mark duration.

ARBITRATION AND SYNCHRONIZATION LOGIC
 In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I²C bus. If another device on the bus overrides a logic 1 and pulls the SDA line low, arbitration is lost, and SIO1 immediately changes from master transmitter to slave receiver. SIO1 will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while SIO1 is returning a "not acknowledge" (logic 1) to the bus. Arbitration is lost when another device on the bus pulls this signal LOW. Since this can occur only at the end of a serial byte, SIO1 generates no further clock pulses. Figure 13 shows the arbitration procedure.

The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the "mark" duration is determined by the device that generates the shortest "marks," and the "space" duration is determined by the device that generates the longest "spaces." Figure 14 shows the synchronization procedure.

A slave may stretch the space duration to slow down the bus master. The space duration may also be stretched for handshaking purposes. This can be done after each bit or after a complete byte transfer. SIO1 will stretch the SCL space duration after a byte has been transmitted or received and the acknowledge bit has been transferred. The serial interrupt flag (SI) is set, and the stretching continues until the serial interrupt flag is cleared.

SERIAL CLOCK GENERATOR

This programmable clock pulse generator provides the SCL clock pulses when SIO1 is in the master transmitter or master receiver mode. It is switched off when SIO1 is in a slave mode. The programmable output clock frequencies are: f_{osc}/20, f_{osc}/9600, and the Timer 1 overflow rate divided by eight. The output clock

pulses have a 50% duty cycle unless the clock generator is synchronized with other SCL clock sources as described above.

TIMING AND CONTROL

The timing and control logic generates the timing and control signals for serial byte handling. This logic block provides the shift pulses for S1DAT, enables the comparator, generates and detects start and stop conditions, receives and transmits acknowledge bits, controls the master and slave modes, contains interrupt request logic, and monitors the I²C bus status.

CONTROL REGISTER, S1CON

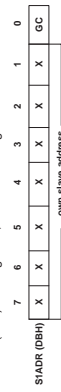
This 7-bit special function register is used by the microcontroller to control the following SIO1 functions: start and restart of a serial transfer, termination of a serial transfer, bit rate, address recognition, and acknowledgment.

STATUS DECODER AND STATUS REGISTER

The status decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I²C bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines. Each service routine processes a particular bus status. There are 26 possible bus states if all four modes of SIO1 are used. The 5-bit status code is latched into the five most significant bits of the status register when the serial interrupt flag is set (by hardware) and remains stable until the interrupt flag is cleared by software. The three least significant bits of the status register are always zero. If the status code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code is sufficient for most of the service routines (see the software example in this section).

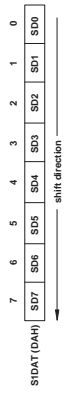
The Four SIO1 Special Function Registers: The microcontroller interfaces to SIO1 via four special function registers. These four SFRs (S1ADR, S1DAT, S1CON, and S1STA) are described individually in the following sections.

The Address Register, S1ADR: The CPU can read from and write to this 8-bit, directly addressable SFR. S1ADR is not affected by the SIO1 hardware. The contents of this register are irrelevant when SIO1 is in a master mode. In the slave modes, the seven most significant bits must be loaded with the microcontroller's own slave address, and, if the least significant bit is set, the general call address (00H) is recognized; otherwise it is ignored.



The most significant bit corresponds to the first bit received from the I²C bus after a start condition. A logic 1 in S1ADR corresponds to a high level on the I²C bus, and a logic 0 corresponds to a low level on the bus.

The Data Register, S1DAT: S1DAT contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from and write to this 8-bit, directly addressable SFR while it is not in the process of shifting a byte. This occurs when SIO1 is in a defined state and the serial interrupt flag is set. Data in S1DAT remains stable as long as SI is set. Data in S1DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7), and after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last data byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.



S1D7 - SD0:

Eight bits to be transmitted or just received. A logic 1 in S1DAT corresponds to a high level on the I²C bus, and a logic 0 corresponds to a low level on the bus. Serial data shifts through S1DAT from right to left. Figure 15 shows how data in S1DAT is serially transferred to and from the SDA line.

S1DAT and the ACK flag form a 9-bit shift register which shifts in or shifts out an 8-bit byte, followed by an acknowledge bit. The ACK flag is controlled by the SIO1 hardware and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into S1DAT on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into S1DAT, the serial data is available in S1DAT, and the acknowledge bit is returned by the control logic during the ninth clock pulse. Serial data is shifted out from S1DAT via a buffer (BSD7) on the falling edges of clock pulses on the SCL line.

When the CPU writes to S1DAT, BSD7 is loaded with the content of S1DAT7, which is the first bit to be transmitted to the SDA line (see Figure 16). After nine serial clock pulses, the eight bits in S1DAT will have been transmitted to the SDA line, and the acknowledge bit will be present in ACK. Note that the eight transmitted bits are shifted back into S1DAT.

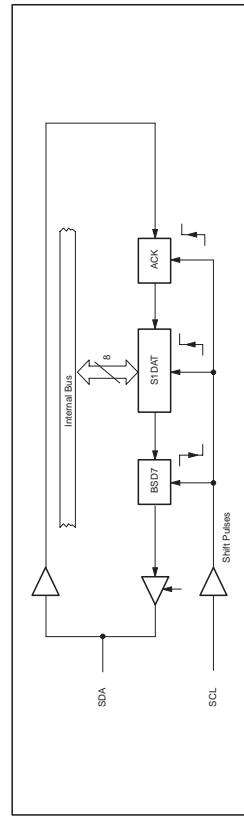


Figure 15. Serial Input/Output Configuration



80C51 Family Derivatives

8XC552/562 overview

The Control Register, S1CON: The CPU can read from, and write to this 8-bit, directly addressable SFR. Two bits are affected by the SIO1 hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I²C bus. The STO bit is also cleared when ENS1 = "0".

S1CON (08h)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
7	6	5	4	3	2	1	0	

ENS1: THE SIO1 ENABLE BIT
 ENS1 is "0". When ENS1 is "0", the SDA and SCL outputs are in a high impedance state. SDA and SCL input signals are ignored. SIO1 is in the "not addressed" slave state, and the STO bit in S1CON is forced to "0". No other bits are affected. P1.6 and P1.7 may be used as open drain I/O ports.

ENS1 = "1": When ENS1 is "1", SIO1 is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

ENS1 should not be used to temporarily release SIO1 from the I²C bus since, when ENS1 is reset, the I²C bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

In the following text, it is assumed that ENS1 = "1".

STA: THE START FLAG
 STA = "1": When the STA bit is set to enter a master mode, the SIO1 hardware checks the status of the I²C bus and generates a START condition if the bus is free. If the bus is not free, then SIO1 waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal serial clock generator.

If STA is set while SIO1 is already in a master mode and one or more bytes are transmitted or received, SIO1 transmits a repeated START condition. STA may be set at any time. STA may also be set when SIO1 is an addressed slave.

STA = "0": When the STA bit is reset, no START condition or repeated START condition will be generated.

STO: THE STOP FLAG
 STO = "1": When the STO bit is set while SIO1 is in a master mode, a STOP condition is transmitted to the I²C bus. When the STOP condition is detected on the bus, the SIO1 hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from an error condition. In this case, no STOP condition is transmitted to the I²C bus. However, the SIO1 hardware behaves as if a STOP condition has been received and switches to the defined "not addressed" slave receiver mode. The STO flag is automatically cleared by hardware.

If the STA and STO bits are both set, the a STOP condition is transmitted to the I²C bus if SIO1 is in a master mode (in a slave mode, SIO1 generates an internal STOP condition which is not transmitted). SIO1 then transmits a START condition.
 STO = "0": When the STO bit is reset, no STOP condition will be generated.

SI: THE SERIAL INTERRUPT FLAG

SI = "1": When the SI flag is set, then, if the EA and ES1 (interrupt enable register) bits are also set, a serial interrupt is requested. SI is set by hardware when one of 25 of the 26 possible SIO1 states is

entered. The only state that does not cause SI to be set is state FBH, which indicates that no relevant state information is available. While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI must be reset by software.

SI = "0": When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

AA: THE ASSERT ACKNOWLEDGE FLAG

AA = "1": If the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- The "own slave address" has been received
- The general call address has been received while the general call bit (GC) in S1ADR is set
- A data byte has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

AA = "0": If the AA flag is reset, a not acknowledge (high level to SDA) will be returned during the acknowledge clock pulse on SCL when:

- A data byte has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

When SIO1 is in the addressed slave transmitter mode, state CBH will be entered after the last serial data is transmitted (see Figure 20).

When SI is cleared, SIO1 leaves state CBH, enters the not addressed slave receiver mode, and the SDA line remains at a high level. In state CBH, the AA flag can be set again for future address recognition.

When SIO1 is in the not addressed slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO1 can be temporarily released from the I²C bus while the bus status is monitored. While SIO1 is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If the AA flag is set when the part's own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

CR0, CR1, AND CR2: THE CLOCK RATE BITS

These three bits determine the serial clock frequency when SIO1 is in a master mode. The various serial rates are shown in Table 2. A 12.5kHz bit rate may be used by devices that interface to the I²C bus via standard I/O port lines which are software driven and slow. 100kHz is usually the maximum bit rate and can be derived from a 1.0MHz, 12MHz, or a 6MHz oscillator. A variable bit rate (0.5kHz to 62.5kHz) may also be used if Timer 1 is not required for any other purpose while SIO1 is in a master mode.

The frequencies shown in Table 2 are unimportant when SIO1 is in a slave mode. In the slave modes, SIO1 will automatically synchronize with any clock frequency up to 100kHz.

80C51 Family Derivatives

8XC552/562 overview

The Status Register, S1STA: S1STA is an 8-bit read-only special function register. The three least significant bits are always zero.

The five most significant bits contain the status code. There are 26 possible status codes. When S1STA contains FBH, no relevant state information is available and no serial interrupt is requested. All other S1STA values correspond to defined SIO1 states. When each of these states is entered, a serial interrupt is requested (SI = "1"). A valid status code is present in S1STA one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software.

More information on SIO1 Operating Modes: The four operating modes are:

- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Data transfers in each mode of operation are shown in Figures 17-37. These figures contain the following abbreviations:

Abbreviation

- S Start condition
- SLA 7-bit slave address
- R Read bit (high level at SDA)
- W Write bit (low level at SDA)
- A Acknowledge bit (low level at SDA)
- A Not acknowledge bit (high level at SDA)
- Data 8-bit data byte
- P Stop condition

In Figures 17-37, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in the S1STA register. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in S1STA is set by branch to the appropriate service routine. For each status

code, the required software action and details of the following serial transfer are given in Tables 3-7.

Master Transmitter Mode: In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 17). Before the master transmitter mode can be entered, S1CON must be initialized as follows:

S1CON (08h)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
7	6	5	4	3	2	1	0	

CR0, CR1, and CR2 define the serial bit rate. ENS1 must be set to logic 1 to enable SIO1. If the AA bit is reset, SIO1 will not acknowledge its own slave address or the general call address in the event of another device becoming master of the bus. In other words, if AA is reset, SIO1 cannot enter a slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit using the SETB instruction. The SIO1 logic will now test the I²C bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (S1STA) will be 0BH. This status code must be used to vector to an interrupt service routine that loads S1DAT with the slave address and the data direction bit (SLA-W). The SI bit in S1CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. There are 19H, 20H, or 38H for the master mode and also 68H, 78H, or 80H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 3. After a repeated start condition (state 10H), SIO1 may switch to the master receiver mode by loading S1DAT with SLA-R).

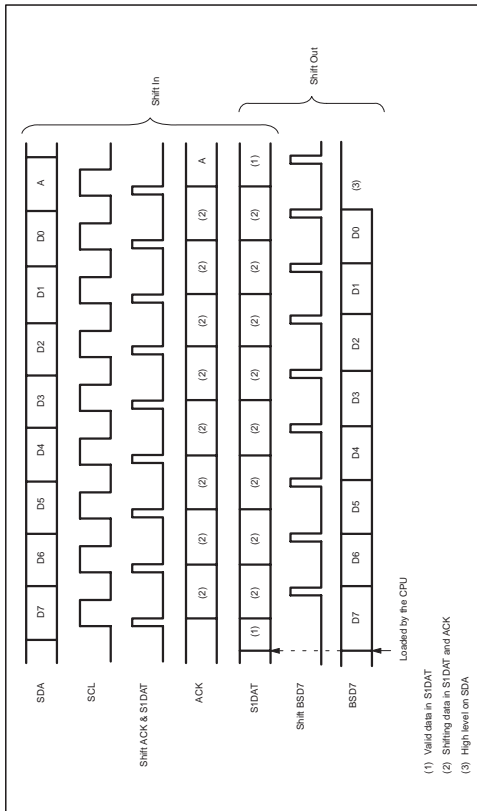


Figure 16. Shift-in and Shift-out Timing

Table 2. Serial Clock Rates

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f _{osc}			f _{osc} DIVIDED BY
			6MHz	12MHz	16MHz	
0	0	0	23	47	63	256
0	0	1	27	54	71	224
0	1	0	31	63	83	192
0	1	1	37	75	100	160
1	0	0	6.25	12.5	17	960
1	0	1	50	100	120	60
1	1	0	100	—	—	96 × (256 - reload value Timer 1) (Reload value range: 0 - 254 in mode 2)
1	1	1	0.25 - 62.5	0.5 - 62.5	0.67 - 56	—

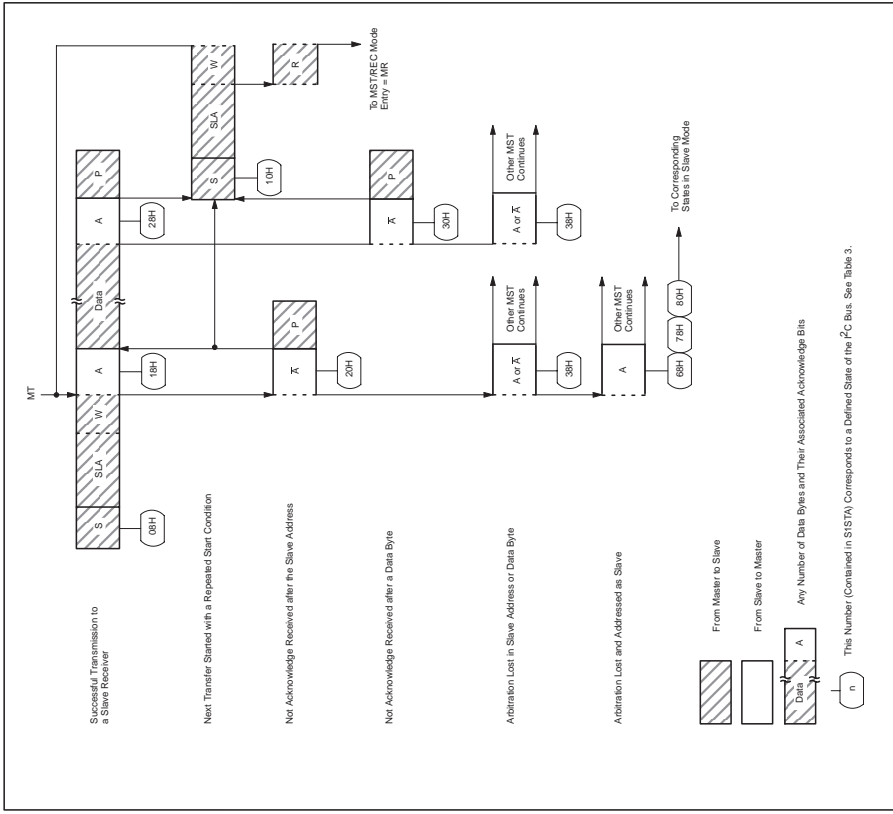


Figure 17. Format and States in the Master Transmitter Mode

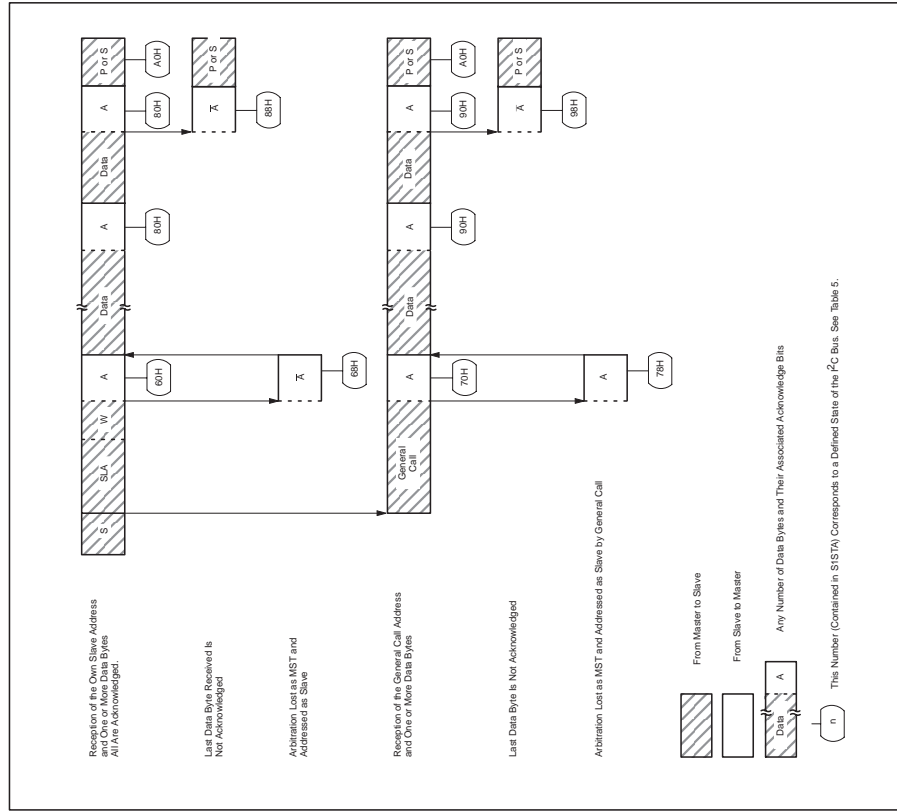


Figure 19. Format and States in the Slave Receiver Mode

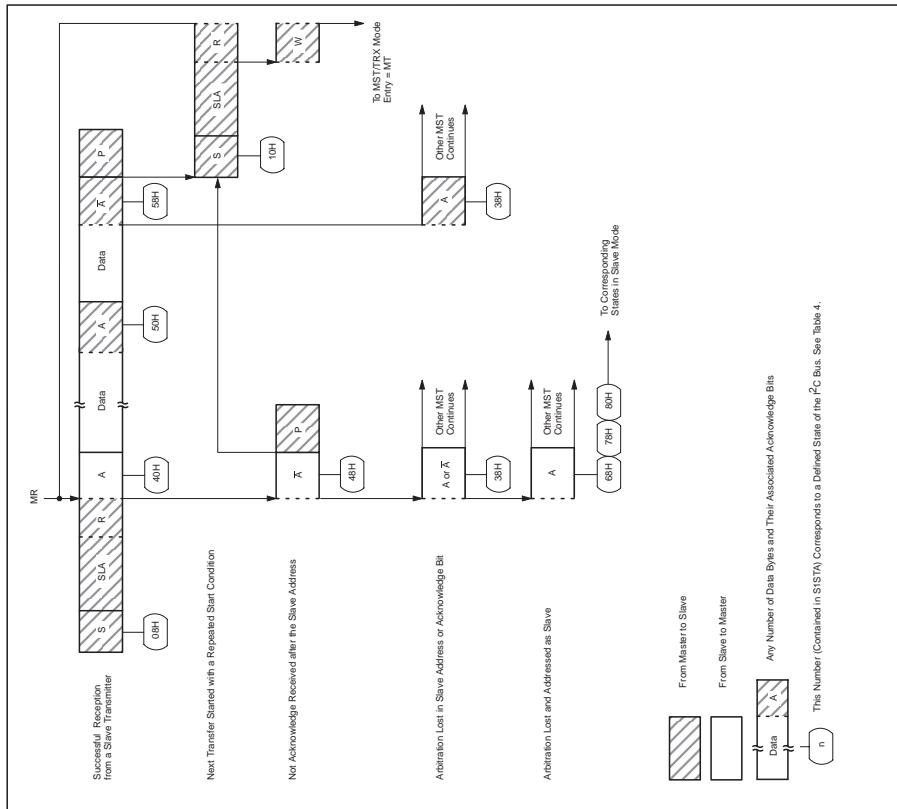


Figure 18. Format and States in the Master Receiver Mode

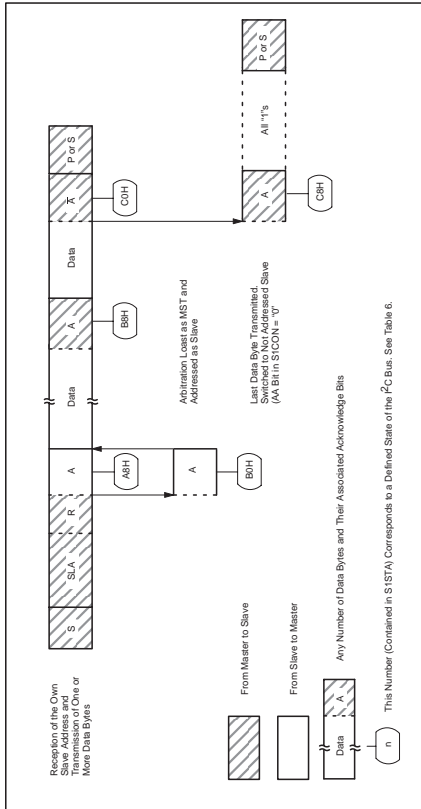


Figure 19. Format and States of the Slave Transmitter Mode

Master Receiver Mode: In the master receiver mode, a number of data bytes are received from a slave transmitter (see Figure 19). The transfer is initialized as in the master transmitter mode. When the start condition has been transmitted, the interrupt service routine must load S'DAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in STCON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S'ISTA are possible. These are 40H, 48H, or 38H for the master mode and also 68H, 78H, or 80H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 4. ENS1, CR1, and CR0 are not affected by the serial transfer and are not referred to in Table 4. After a repeated start condition (state 10H), SIO1 may switch to the master transmitter mode by loading S'DAT with SLA+W.

Slave Receiver Mode: In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 19). To initiate the slave receiver mode, STADR and STCON must be loaded as follows:

STADR (DBH)	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	gc

own slave address

The upper 7 bits are the address to which SIO1 will respond when addressed by a master. If the LSB (GC) is set, SIO1 will respond to

Table 3. Master Transmitter Mode

STATUS CODE (S'ISTA)	STATUS OF THE I ² C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE				NEXT ACTION TAKEN BY SIO1 HARDWARE	
		TO/FROM S'DAT	STA	STO	SI AA		
08H	A START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+W or Load SLA+R	X	0	0	X	As above SLA+W will be transmitted; SIO1 will be switched to MST/REC mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte or no S'DAT action or no S'DAT action or no S'DAT action	0	0	0	X	Data byte will be transmitted; ACK bit will be received Repeated START will be transmitted; SIO1 will be reset STOP condition will be transmitted; SIO1 flag will be reset STOP condition followed by a START condition will be transmitted; SIO1 flag will be reset
20H	SLA+W has been transmitted; NOT ACK has been received	Load data byte or no S'DAT action or no S'DAT action or no S'DAT action	0	0	0	X	Data byte will be transmitted; ACK bit will be received Repeated START will be transmitted; SIO1 will be reset STOP condition will be transmitted; SIO1 flag will be reset STOP condition followed by a START condition will be transmitted; SIO1 flag will be reset
28H	Data byte in S'DAT has been transmitted; ACK has been received	Load data byte or no S'DAT action or no S'DAT action or no S'DAT action	0	0	0	X	Data byte will be transmitted; ACK bit will be received Repeated START will be transmitted; SIO1 flag will be reset STOP condition will be transmitted; SIO1 flag will be reset STOP condition followed by a START condition will be transmitted; SIO1 flag will be reset
30H	Data byte in S'DAT has been transmitted; NOT ACK has been received	Load data byte or no S'DAT action or no S'DAT action or no S'DAT action	0	0	0	X	Data byte will be transmitted; ACK bit will be received Repeated START will be transmitted; SIO1 flag will be reset STOP condition will be transmitted; SIO1 flag will be reset STOP condition followed by a START condition will be transmitted; SIO1 flag will be reset
38H	Arbitration lost in SLA+RW or Data bytes	No S'DAT action or No S'DAT action	0	0	0	X	I ² C bus will be released; not addressed slave will be entered A START condition will be transmitted when the bus becomes free



Table 5. Slave Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I ² C BUS AND SIO1 HARDWARE	TO/FROM S1DAT	APPLICATION SOFTWARE RESPONSE				NEXT ACTION TAKEN BY SIO1 HARDWARE	
			TO S1CON	TO S1CON	TO S1CON	TO S1CON		
			STA	STO	SI	AA		
60H	Own SLA+W has been received; ACK has been returned	No S1DAT action or no S1DAT action	X	0	0	0	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
68H	Arbitration lost in SLA+R/W as master; Own SLA+W has been received; ACK has been returned	No S1DAT action or no S1DAT action	X	0	0	0	1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
70H	General call address (00H) has been received; ACK has been returned	No S1DAT action or no S1DAT action	X	0	0	0	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
78H	Arbitration lost in SLA+R/W as master; General call address has been received; ACK has been returned	No S1DAT action or no S1DAT action	X	0	0	0	1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
80H	Previously addressed with own SLV address; DATA has been received; ACK has been returned	Read data byte or read data byte	X	0	0	0	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
88H	Previously addressed with own SLA; DATA byte has been received; NOT ACK has been returned	Read data byte or read data byte or read data byte or read data byte	0	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1 Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
90H	Previously addressed with General Call; DATA byte has been received; ACK has been returned	Read data byte or read data byte	X	0	0	0	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
98H	Previously addressed with General Call; DATA byte has been received; NOT ACK has been returned	Read data byte or read data byte or read data byte or read data byte	0	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1 Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Table 4. Master Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I ² C BUS AND SIO1 HARDWARE	TO/FROM S1DAT	APPLICATION SOFTWARE RESPONSE				NEXT ACTION TAKEN BY SIO1 HARDWARE	
			TO S1CON	TO S1CON	TO S1CON	TO S1CON		
			STA	STO	SI	AA		
08H	A START condition has been transmitted	Load SLA+R	X	0	0	0	X	SLA+R will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+R or Load SLA+W	X	0	0	0	X	As above SLA+R will be transmitted; SIO1 will be switched to MSTR/TRX mode
38H	Arbitration lost in NOT ACK bit	No S1DAT action or No S1DAT action	0	0	0	0	X	I ² C bus will be released; SIO1 will enter a slave mode A START condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	No S1DAT action or no S1DAT action	0	0	0	0	0	Data byte will be received; NOT ACK bit will be returned Data byte will be received; ACK bit will be returned
48H	SLA+R has been transmitted; NOT ACK has been received	No S1DAT action or no S1DAT action or no S1DAT action	1	0	0	0	X	Repeated START condition will be transmitted STOP condition will be transmitted; S1TO flag will be reset STOP condition followed by a START condition will be transmitted; S1TO flag will be reset
50H	Data byte has been received; ACK has been returned	Read data byte or read data byte	0	0	0	0	0	Data byte will be received; NOT ACK bit will be returned Data byte will be received; ACK bit will be returned
58H	Data byte has been received; NOT ACK has been returned	Read data byte or read data byte or read data byte	1	0	0	0	X	Repeated START condition will be transmitted STOP condition will be transmitted; S1TO flag will be reset STOP condition followed by a START condition will be transmitted; S1TO flag will be reset

Table 5. Slave Receiver Mode (Continued)

STATUS CODE (S1STA)	STATUS OF THE I ² C BUS AND S1OT HARDWARE	APPLICATION SOFTWARE RESPONSE				NEXT ACTION TAKEN BY S1OT HARDWARE					
		TO/FROM SIDAT	STA	STO	SI	AA	STA	STO	SI	AA	
A0H	A STOP condition or a repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No STDAT action or	0	0	0	0	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		No STDAT action or	0	0	0	0	0	0	1	0	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		No STDAT action or	1	0	0	0	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		No STDAT action	1	0	0	0	0	0	1	0	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Table 6. Slave Transmitter Mode

STATUS CODE (S1STA)	STATUS OF THE I ² C BUS AND S1OT HARDWARE	APPLICATION SOFTWARE RESPONSE				NEXT ACTION TAKEN BY S1OT HARDWARE					
		TO/FROM SIDAT	STA	STO	SI	AA	STA	STO	SI	AA	
A6H	Own SLA/R has been received; ACK has been returned	Load data byte or	X	0	0	0	0	0	0	0	Last data byte will be transmitted and ACK bit will be received
B0H	Arbitration lost in SLA/R/W as master; Own SLA/R has been received; ACK has been returned	Load data byte or	X	0	0	0	0	0	0	0	Last data byte will be transmitted and ACK bit will be received
B8H	Data byte in SIDAT has been transmitted; ACK has been received	Load data byte or	X	0	0	0	0	0	0	0	Last data byte will be transmitted and ACK bit will be received
C0H	Data byte in SIDAT has been transmitted; NOT ACK has been received	No STDAT action or	0	0	0	0	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. Own SLA/R will be recognized if S1ADR.0 = logic 1
		No STDAT action or	0	0	0	0	0	0	1	0	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
		No STDAT action	1	0	0	0	0	0	1	0	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
C8H	Last data byte in SIDAT has been transmitted (AA = 0); NOT ACK has been received	No STDAT action or	0	0	0	0	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. Own SLA will be recognized if S1ADR.0 = logic 1
		No STDAT action or	0	0	0	0	0	0	1	0	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
		No STDAT action or	1	0	0	0	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		No STDAT action	1	0	0	0	0	0	1	0	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Slave Transmitter Mode: In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 20). Data transfer is initialized as in the slave receiver mode. When S1ADR and S1CON have been initialized, S1OT waits until it is addressed by its own slave address followed by the data direction bit which must be "1" (R) for S1OT to operate in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 6. The slave transmitter mode may also be entered if arbitration is lost while S1OT is in the master mode (see state B0H).

If the AA bit is reset during a transfer, S1OT will transmit the last byte of the transfer and enter state C0H or C8H. S1OT is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all its own slave data. While AA is reset, S1OT does not respond to its own slave address or a general call address. However, the I²C bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate S1OT from the I²C bus.

Miscellaneous States: There are two S1STA codes that do not correspond to a defined S1OT hardware state (see Table 7). These are discussed below.

S1STA = FBH: This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs between other states and when S1OT is not involved in a serial transfer.

S1STA = 00H: This status code indicates that a bus error has occurred during an S1OT serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal S1OT signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This causes S1OT to enter the "not addressed" slave mode (a defined state) and SDA and SCL lines are released (a STOP condition is not transmitted).

Some Special Cases: The S1OT hardware has facilities to handle the following special cases that may occur during a serial transfer:

Simultaneous Repeated START Conditions from Two Masters
A repeated START condition may be generated in the master transmitter or master receiver modes. A special case occurs if another master simultaneously generates a repeated START condition (see Figure 21). Until this occurs, arbitration is not lost by either master since they were both transmitting the same data.

If the S1OT hardware detects a repeated START condition on the I²C bus before generating a repeated START condition itself, it will release the bus, and no interrupt request is generated. If another master frees the bus by generating a STOP condition, S1OT will transmit a normal START condition (state 08H), and a retry of the total serial data transfer can commence.

DATA TRANSFER AFTER LOSS OF ARBITRATION
Arbitration may be lost in the master transmitter and master receiver modes (see Figure 13). Loss of arbitration is indicated by the following states in S1STA: 38H, 68H, 78H, and B0H (see Figures 17 and 18).

If the STA flag in S1CON is set by the routines which service these states, then, if the bus is free again, a START condition (state 08H) is transmitted without intervention by the CPU, and a retry of the total serial transfer can commence.

Forced Access to the I²C Bus
In some applications, it may be possible for an uncontrolled source to cause a bus hang-up. In such situations, the problem may be caused by interference, temporary interruption of the bus or a temporary short-circuit between SDA and SCL.

If an uncontrolled source generates a superfluous START or masks a STOP condition, then the I²C bus stays busy indefinitely. If the STA flag is set and bus access is not obtained within a reasonable amount of time, then a forced access to the I²C bus is possible. This is achieved by setting the STA flag while the I²C bus is still set. No STOP condition is transmitted. The S1OT hardware behaves as if a STOP condition was received and is able to transmit a START condition. The STO flag is cleared by hardware (see Figure 22).

I²C BUS OBSTRUCTED BY A LOW LEVEL ON SCL OR SDA
An I²C bus hang-up occurs if SDA or SCL is pulled LOW by an uncontrolled source. If the SCL line is obstructed (pulled LOW) by a device on the bus, no further serial transfer is possible, and the S1OT hardware cannot resolve this type of problem. When this occurs, the problem must be resolved by the device that is pulling the SCL bus line LOW.

If the SDA line is obstructed by another device on the bus (e.g., a slave device out of bit synchronization), the problem can be solved by transmitting additional clock pulses on the SCL line (see Figure 23). The S1OT hardware transmits additional clock pulses when the STA flag is set, but no START condition can be generated because the SDA line is pulled LOW while the I²C bus is considered free. The S1OT hardware attempts to generate a START condition after every two additional clock pulses on the SCL line. When the SDA line is eventually released, a normal START condition is transmitted, state 08H is entered, and the serial transfer continues.

If a forced bus access occurs or a repeated START condition is transmitted while SDA is obstructed (pulled LOW), the S1OT hardware performs the same action as described above. In each case, state 08H is entered after a successful START condition is transmitted and normal serial transfer continues. Note that the CPU is not involved in solving these bus hang-up problems.

Bus Error

A bus error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data or an acknowledge bit.

The S1OT hardware only reacts to a bus error when it is involved in a serial transfer either as a master or an addressed slave. When a bus error is detected, S1OT immediately switches to the not addressed slave mode, releases the SDA and SCL lines, sets the interrupt flag, and loads the status register with 00H. This status code may be used to vector to a service routine which either attempts the aborted serial transfer again or simply recovers from the error condition as shown in Table 7.



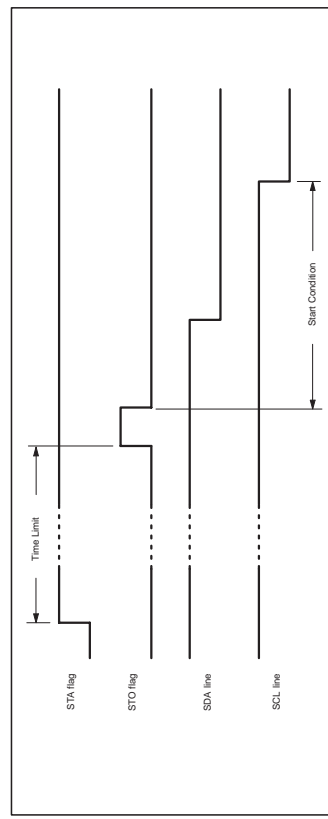


Figure 22. Forced Access to a Busy I²C Bus

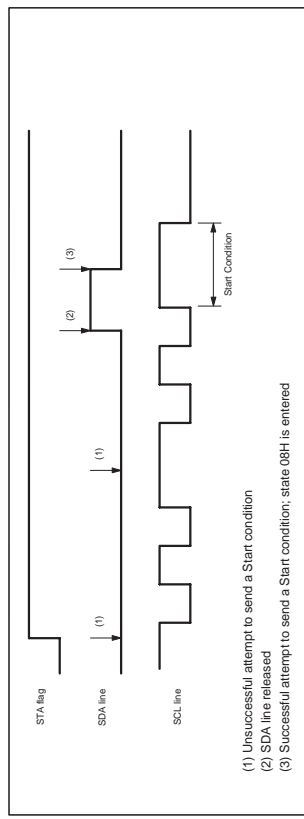


Figure 23. Recovering from a Bus Obstruction Caused by a Low Level on SDA

Table 7. Miscellaneous States

STATUS CODE (S1STA)	STATUS OF THE I²C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE				NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM STDAT	TO S1CON	SI	AA	
FBH	No relevant state information available; SI = 0	No STDAT action	No S1CON action	No S1CON action	AA	Wait or proceed current transfer
00H	Bus error during MST or selected slave modes, due to an illegal START or STOP condition. State 00H can also occur when interference causes SIO1 to enter an undefined state.	No STDAT action	0	1	0	X Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and SIO1 is switched to the not addressed SLV mode. STO is reset.

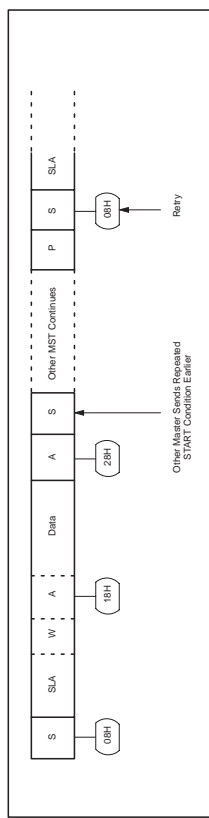


Figure 21. Simultaneous Repeated START Conditions from 2 Masters

Interrupts

The 8XC552 has fifteen interrupt sources, each of which can be assigned one of two priority levels, as shown in Figure 27. The five interrupt sources common to the 80C51 are the external interrupts (INT0 and INT1), the timer 0 and timer 1 interrupts (T0 and T1), and the serial I/O interrupt (RI or TI). In the 8XC552, the standard serial interrupt is called SIO0. Since the subsystems which create these interrupts are identical on both parts, their functionality is likewise identical. The only differences are the locations of the enable and priority register configurations and the priority structure. This is detailed below along with the specifics of the interrupts unique to the 8XC552.

The eight Timer T2 interrupts are generated by flags CTI0-CTI3, CM10-CM12, and by the logical OR of flags T2OV and T2BO. Flags CTI0 to CTI3 are set by input signals CT01 to CT31. Flags CM10 to CM12 are set when a match occurs between Timer T2 and the compare registers CM0, CM1, and CM2. When an 8-bit or 16-bit overflow occurs, flags T2BO and T2OV are set, respectively. These nine flags are not cleared by hardware and must be reset by software to avoid recurring interrupts.

The ADC interrupt is generated by the ADCl flag in the ADC control register (ADCON). This flag is set when an ADC conversion result is ready to be read. ADCl is not cleared by hardware and must be reset by software to avoid recurring interrupts.

The SIO1 (I²C) interrupt is generated by the SI flag in the SIO1 control register (SIOCON). This flag is set when STSTA is loaded with a valid status code.

The ADCl flag may be reset by software. It cannot be set by software. All other flags that generate interrupts may be set or cleared by software, and the effect is the same as setting or resetting the flags by hardware. Thus, interrupts may be generated by software and pending interrupts can be cancelled by software.

Interrupt Enable Registers: Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable special function registers IEN0 and IEN1. All interrupt sources can also be globally enabled or disabled by setting or clearing bit EA in IEN0. The interrupt enable registers are described in Figures 28 and 29.

Interrupt Priority Structure: Each interrupt source can be assigned one of two priority levels. Interrupt priority levels are defined by the interrupt priority special function registers IP0 and IP1. IP0 and IP1 are described in Figures 30 and 31.

Interrupt priority levels are as follows:

- "0"—low priority
- "1"—high priority

A low priority interrupt may be interrupted by a high priority interrupt. A high priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority occur simultaneously, the

high priority level request is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus, within each priority level, there is a second priority structure determined by the polling sequence. This second priority structure is shown in Table 8.

The above Priority Within Level structure is only used when there are simultaneous requests of the same priority level.

Interrupt Handling: The interrupt sources are sampled at SSP2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at SSP2 of the previous machine cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of higher or equal priority level is already in progress.
2. The current machine cycle is not the final cycle in the execution of the instruction in progress. (No interrupt request will be serviced until the instruction in progress is completed.)

3. The instruction in progress is RETI or any access to the interrupt priority or interrupt enable registers. (No interrupt will be serviced after RETI or after a read or write to IP0, IP1, IEO, or IE1 until at least one other instruction has been subsequently executed.)

The polling cycle is repeated with every machine cycle, and the values polled are the values present at SSP2 of the previous machine cycle. Note that if an interrupt flag is active but is not being responded to because of one of the above conditions, and if the flag is inactive when the blocking condition is removed, then the blocked interrupt will not be serviced. Thus, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate service routine. In some cases it also clears the flag which generated the interrupt, and in others it does not. It clears the Timer 0, Timer 1, and external interrupt flags. An external interrupt flag (IE0 or IE1) is cleared only if it was transition-activated. All other interrupt flags are not cleared by hardware and must be cleared by the software. The LCALL pushes the contents of the program counter on to the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to as shown in Table 9.

Execution proceeds from the vector address until the RETI instruction is encountered. The RETI instruction clears the "priority level active" flip-flop that was set when this interrupt was acknowledged. It then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from where it was interrupted.

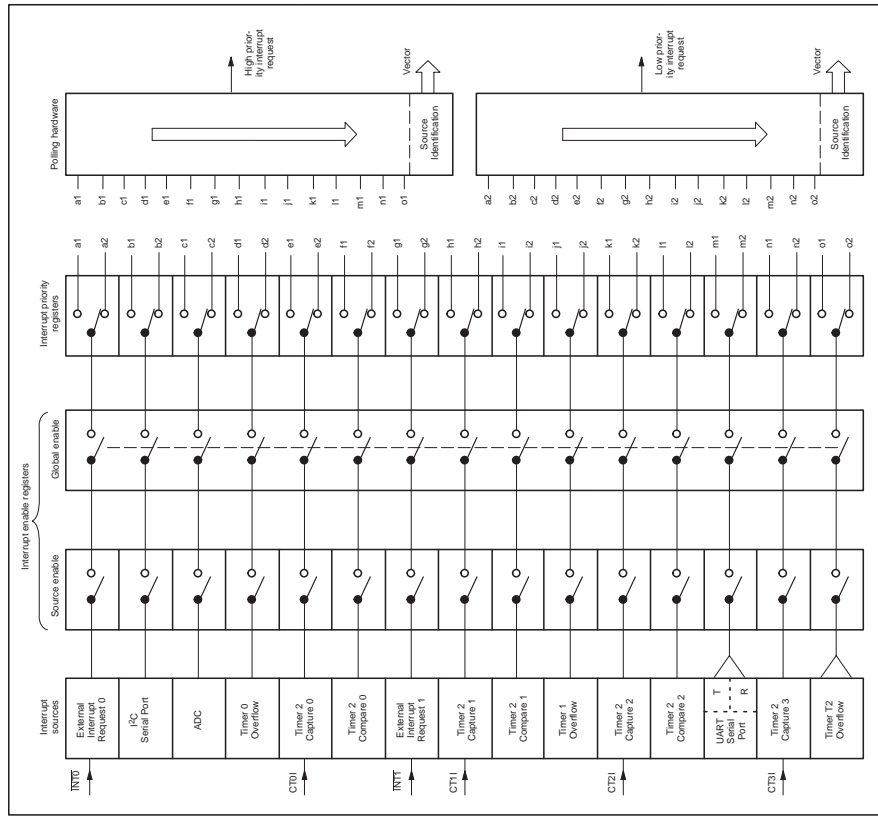


Figure 27. The Interrupt System



IP1 (F8H)										PCT0 (LSB)	
		(MSB)									
BIT	SYMBOL	FUNCTION	7	6	5	4	3	2	1	0	
IP1.7	PT2	T2 overflow interrupt(s) priority level									
IP1.6	PCM2	T2 comparator 2 interrupt priority level									
IP1.5	PCM1	T2 comparator 1 interrupt priority level									
IP1.4	PCM0	T2 comparator 0 interrupt priority level									
IP1.3	PCT3	T2 capture register 3 interrupt priority level									
IP1.2	PCT2	T2 capture register 2 interrupt priority level									
IP1.1	PCT1	T2 capture register 1 interrupt priority level									
IP1.0	PCT0	T2 capture register 0 interrupt priority level									

S1J00764

Figure 31. Interrupt Priority Register (IP1)

Table 8. Interrupt Priority Structure			
SOURCE	NAME	PRIORITY WITHIN LEVEL	
External interrupt 0	X0	(highest)	
SIO1 (I ² C)	S1		
ADC completion	ADC		
Timer 0 overflow	T0		
T2 capture 0	CT0		
External interrupt 1	X1		
T2 capture 1	CT1		
T2 compare 1	CM1		
Timer 1 overflow	T1		
T2 capture 2	CT2		
T2 compare 2	CM2		
SIO0 (UART)	S0		
T2 capture 3	CT3		
Timer T2 overflow	T2	(lowest)	

Table 9. Interrupt Vector Addresses			
SOURCE	NAME	VECTOR ADDRESS	
External interrupt 0	X0	0003H	
Timer 0 overflow	T0	000BH	
External interrupt 1	X1	0013H	
Timer 1 overflow	T1	001BH	
SIO0 (UART)	S0	0023H	
SIO1 (I ² C)	S1	002BH	
T2 capture 0	CT0	0033H	
T2 capture 1	CT1	003BH	
T2 capture 2	CT2	0043H	
T2 capture 3	CT3	004BH	
ADC completion	ADC	0053H	
T2 compare 0	CM0	005BH	
T2 compare 1	CM1	0063H	
T2 compare 2	CM2	006BH	
T2 overflow	T2	0073H	

IEN0 (A8H)										EX0 (LSB)	
		(MSB)									
BIT	SYMBOL	FUNCTION	7	6	5	4	3	2	1	0	
IEN0.7	EA	Global enable/disable control 0 = No interrupt is enabled 1 = Any individually enabled interrupt will be accepted									
IEN0.6	EAD	Enable ADC interrupt									
IEN0.5	ES1	Enable SIO1 (I ² C) interrupt									
IEN0.4	ES0	Enable SIO0 (UART) interrupt									
IEN0.3	ET1	Enable Timer 1 interrupt									
IEN0.2	EX1	Enable External interrupt 1									
IEN0.1	ET0	Enable Timer 0 interrupt									
IEN0.0	EX0	Enable External interrupt 0									

S1J00762

Figure 28. Interrupt Enable Register (IEN0)

IEN1 (E8H)										ECT0 (LSB)	
		(MSB)									
BIT	SYMBOL	FUNCTION	7	6	5	4	3	2	1	0	
IEN1.7	ET2	Enable Timer T2 overflow interrupt(s)									
IEN1.6	ECM2	Enable T2 Comparator 2 interrupt									
IEN1.5	ECM1	Enable T2 Comparator 1 interrupt									
IEN1.4	ECM0	Enable T2 Comparator 0 interrupt									
IEN1.3	ECT3	Enable T2 Capture register 3 interrupt									
IEN1.2	ECT2	Enable T2 Capture register 2 interrupt									
IEN1.1	ECT1	Enable T2 Capture register 1 interrupt									
IEN1.0	ECT0	Enable T2 Capture register 0 interrupt									

S1J00765

In all cases, if the enable bit is 0, then the interrupt is disabled, and if the enable bit is 1, then the interrupt is enabled.

Figure 29. Interrupt Enable Register (IEN1)

IP0 (B8H)										PX0 (LSB)	
		(MSB)									
BIT	SYMBOL	FUNCTION	7	6	5	4	3	2	1	0	
IP0.7	-	Unused									
IP0.6	PAD	ADC interrupt priority level									
IP0.5	PS1	SIO1 (I ² C) interrupt priority level									
IP0.4	PS0	SIO0 (UART) interrupt priority level									
IP0.3	PT1	Timer 1 interrupt priority level									
IP0.2	PX1	External interrupt 1 priority level									
IP0.1	PT0	Timer 0 interrupt priority level									
IP0.0	PX0	External interrupt 0 priority level									

S1J00763

Figure 30. Interrupt Priority Register (IP0)

I/O Port Structure
The 8XC552 has six 8-bit ports. Each port consists of a latch (special function registers P0 to P5), an input buffer, and an output driver (port 0 to 4 only). Ports 0-3 are the same as in the 80C51, with the exception of the additional functions of port 1. The parallel I/O function of port 4 is equal to that of ports 1, 2, and 3. Port 5 may be used as an input port only.

Figure 32 shows the bit latch and I/O buffer functional diagrams of the unique 8XC552 ports. A bit latch corresponds to one bit in a port's SFR and is represented as a D type flip-flop. A "write to latch" signal from the CPU latches a bit from the internal bus and a "read latch" signal from the CPU places the Q output of the flip-flop on the internal bus. A "read pin" signal from the CPU places the actual port pin level on the internal bus. Some instructions that read a port read the actual port pin levels, and other instructions read the latch (SFR) contents.

Port 1 Operation

Port 1 operates the same as it does in the 8051, with the exception of port lines P1.6 and P1.7, which may be selected as the SCL and SDA lines of serial port SIO1 (I²C). Because the I²C bus may be active while the device is disconnected from V_{DD}, these pins are provided with open drain drivers. Therefore pins P1.6 and P1.7 do not have internal pull-ups.

Port 5 Operation

Port 5 may be used to input up to 8 analog signals to the ADC. Unused ADC inputs may be used to input digital inputs. These inputs have an inherent hysteresis to prevent the input logic from drawing excessive current from the power lines when driven by analog signals. Channel to channel crosstalk (CI) should be taken into consideration when both analog and digital signals are simultaneously input to Port 5 (see, D.C. characteristics in data sheet).

Port 5 is not bidirectional and may not be configured as an output port. All six ports are multifunctional, and their alternate functions are listed in Table 10. A more detailed description of these features can be found in the relevant parts of this section.

Pulse Width Modulated Outputs

The 8XC552 contains two pulse width modulated output channels (see Figure 33). These channels generate pulses of programmable length and interval. The repetition frequency is defined by an 8-bit prescaler PWM_P, which supplies the clock for the counter. The prescaler counter and counter are common to both PWM channels. The 8-bit counter counts modulo 255, i.e., from 0 to 254 inclusive. The value of the 8-bit counter is compared to the contents of two registers: PWM0 and PWM1. Provided the contents of either of these registers is greater than the counter value, the corresponding PWM0 or PWM1 output is set LOW. If the contents of these registers are equal to, or less than the counter value, the output will be HIGH. The pulse-width-ratio is therefore defined by the contents of the registers

PWM0 and PWM1. The pulse-width-ratio is in the range of 0 to 1 and may be programmed in increments of 1/255.

Buffered PWM outputs may be used to drive DC motors. The rotation speed of the motor would be proportional to the contents of PWMn. The PWM outputs may also be configured as a dual DAC. In this application, the PWM outputs must be integrated using conventional operational amplifier circuitry. If the resulting output voltages have to be accurate, external buffers with their own analog supply should be used to buffer the PWM outputs before they are integrated. The repetition frequency f_{PWM} at the PWMn outputs is given by:

$$f_{PWM} = \frac{f_{OSC}}{2 \cdot (1 + PWMn) \cdot 255}$$

This gives a repetition frequency range of 123Hz to 31.4kHz (f_{OSC} = 16MHz). At f_{OSC} = 24MHz, the frequency range is 184Hz to 47.1Hz. By loading the PWM registers with either 00H or FFH, the PWM channels will output a constant HIGH or LOW level, respectively. Since the 8-bit counter counts modulo 255, it can never actually reach the value of the PWM registers when they are loaded with FFH.

When a compare register (PWM0 or PWM1) is loaded with a new value, the associated output is updated immediately. It does not have to wait until the end of the current counter period. Both PWMn output pins are driven by push-pull drivers. These pins are not used for any other purpose.

Prescaler frequency control register PWMP

PWMP (FEH)	7	6	5	4	3	2	1	0	
	MSB							LSB	

PWMP-0-7 Prescaler division factor = PWMP + 1.

Reading PWMP gives the current reload value. The actual count of the prescaler cannot be read.

PWMP (FCH)	7	6	5	4	3	2	1	0	
PWMP (FPH)	MSB							LSB	

$$PWM0/1 (0-7) \text{ Low/high ratio of } \frac{PWMn}{255} \text{ (PWMn)}$$

Analog-to-Digital Converter

The analog input circuitry consists of an 8-input analog multiplexer and a 10-bit, straight binary, successive approximation ADC. The analog reference voltage and analog power supplies are connected via separate input pins. The conversion takes 50 machine cycles, i.e., 37.5µs at an oscillator frequency of 16MHz, 25µs at an oscillator frequency of 24MHz. Input voltage swing is from 0V to +5V. Because the internal DAC employs a ratio-metric potentiometer, there are no discontinuities in the converter characteristic. Figure 34 shows a functional diagram of the analog input circuitry.

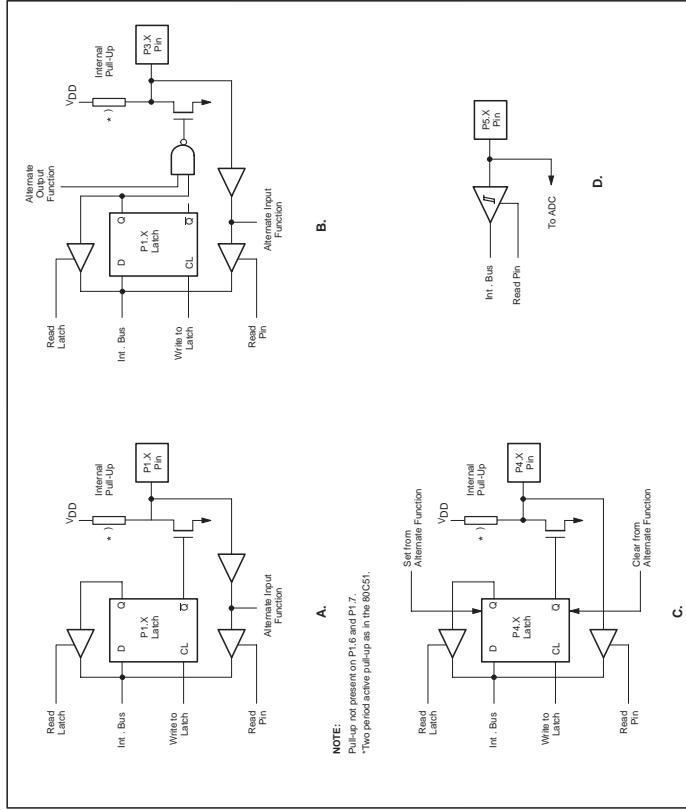


Figure 32. Port Bit Latches and I/O Buffers



Table 10. Input/Output Ports

PORT PIN	ALTERNATE FUNCTION
P0.0 P0.1 P0.2 P0.3 P0.4 P0.5 P0.6 P0.7	AD0 AD1 AD2 AD3 AD4 AD5 AD6 AD7 Multiplexed lower order address/data bus used during external memory accesses
P1.0 P1.1 P1.2 P1.3 P1.4 P1.5 P1.6 P1.7	CT0l CT1l CT2l CT3l RTZ RTZ SCL SDA Capture timer input signals for timer T2 T2 timer reset signal. Rising edge triggered Serial port clock line. I ² C bus Serial port data line. I ² C bus
P2.0 P2.1 P2.2 P2.3 P2.4 P2.5 P2.6 P2.7	A8 A9 A10 A11 A12 A13 A14 A15 High order address byte used during external memory accesses
P3.0 P3.1 P3.2 P3.3 P3.4 P3.5 P3.6 P3.7	RxD TxD INT0 INT1 T0 T1 WR RD Serial input port (UART) Serial output port (UART) External interrupt 0 External interrupt 1 Timer 0 external input Timer 1 external input External data memory write strobe External data memory read strobe
P4.0 P4.1 P4.3 P4.4 P4.5 P4.6 P4.7	CMSR0 CMSR1 CMSR2 CMSR3 CMSR4 CMSR5 CMT0 CMT1 Timer T2; compare and set/reset outputs on a match with timer T2 Timer T2; compare and toggle outputs on a match with timer T2
P5.0 P5.1 P5.2 P5.3 P5.4 P5.5 P5.6 P5.7	ADC0 ADC1 ADC2 ADC3 ADC4 ADC5 ADC6 ADC7 Eight analogue ADC inputs

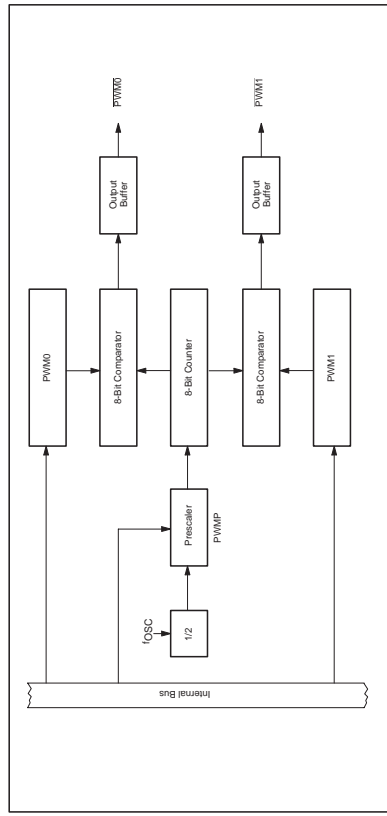


Figure 33. Functional Diagram of Pulse Width Modulated Outputs

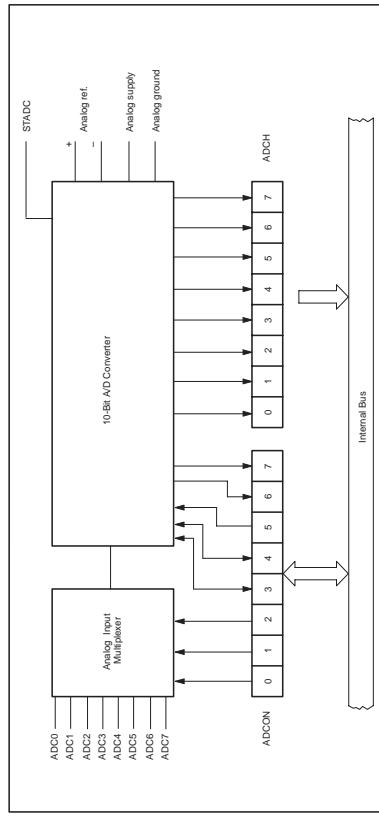


Figure 34. Functional Diagram of Analog Input Circuitry

Analog-to-Digital Conversion: Figure 35 shows the elements of a successive approximation (SA) ADC. The ADC contains a DAC which converts the contents of a successive approximation register to a voltage (VDAC) which is compared to the analog input voltage (VIN). The output of the comparator is fed to the successive approximation control logic which controls the successive approximation register. A conversion is initiated by setting ADCS in the ADCON register. ADCS can be set by software only or by either hardware or software.

The software only start mode is selected when control bit ADCON.5 (ADEX) = 0. A conversion is then started by setting control bit ADCON.3 (ADCS). The hardware or software start mode is selected when ADCON.5 = 1, and a conversion may be started by setting ADCON.3 as above or by applying a rising edge to external pin STADC. When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

The low-to-high transition of STADC is recognized at the end of a machine cycle, and the conversion commences at the beginning of the next cycle. When a conversion is initiated by software, the conversion starts at the beginning of the machine cycle which follows the instruction that sets ADCS. ADCS is actually implemented with two flip-flops: a command flip-flop which is affected by set operations, and a status flag which is accessed during read operations.

The next two machine cycles are used to initiate the converter. At the end of the first cycle, the ADCS status flag is set and a value of "1" will be returned if the ADCS flag is read while the conversion is in progress. Sampling of the analog input commences at the end of the second cycle.

During the next eight machine cycles, the voltage at the previously selected pin or port is sampled, and this input voltage should be stable in order to obtain a useful sample. In any event, the input

voltage slew rate must be less than 10V/ms in order to prevent an undefined result.

The successive approximation control logic first sets the most significant bit and clears all other bits in the successive approximation register (10 0000 0000B). The output of the DAC (50% full scale) is compared to the input voltage VIN. If the input voltage is greater than VDAC, then the bit remains set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (11 0000 0000B or 01 0000 0000B, depending on the previous result), and VDAC is compared to VIN again. If the input voltage is greater than VDAC, then the bit being tested remains set; otherwise, the bit being tested is cleared. This process is repeated until all ten bits have been tested, at which stage the result of the conversion is held in the successive approximation register. Figure 36 shows a conversion flow chart. The bit pointer identifies the bit under test. The conversion takes four machine cycles per bit.

The end of the 10-bit conversion is flagged by control bit ADCON.4 (ADCI). The upper 8 bits of the result are held in special function register ADCH, and the two remaining bits are held in ADCON.7 (ADC.1) and ADCON.6 (ADC.0). The user may ignore the two least significant bits in ADCON and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 50 machine cycles for the 8XC552 or 24 machine cycles for the 8XC562. ADCI will be set and the ADCS status flag will be reset 50 (or 24) cycles after the command flip-flop (ADCS) is set.

Control bits ADCON.0, ADCON.1, and ADCON.2 are used to control an analog multiplexer which selects one of eight analog channels (see Figure 37). An ADC conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1; a new ADC conversion already in progress is aborted when the idle or power-down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode.

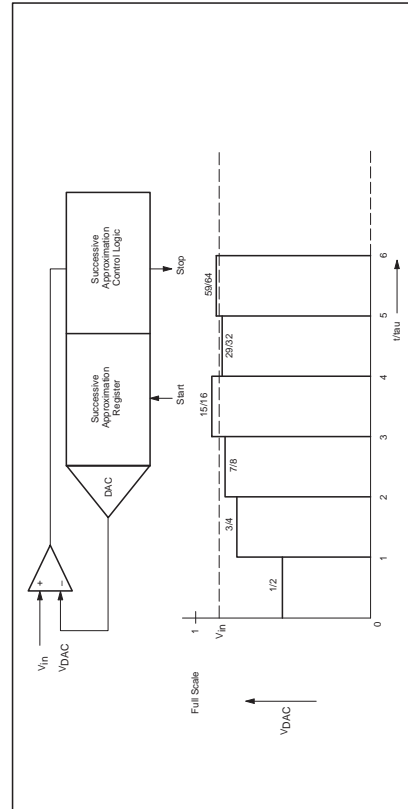


Figure 35. Successive Approximation ADC

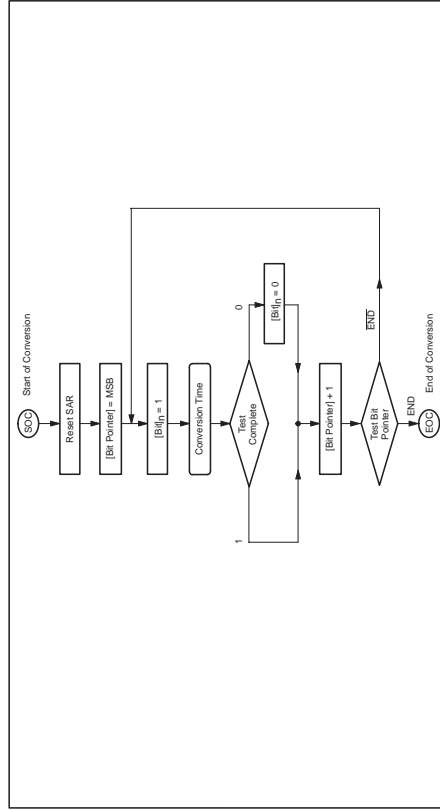


Figure 36. A/D Conversion Flowchart

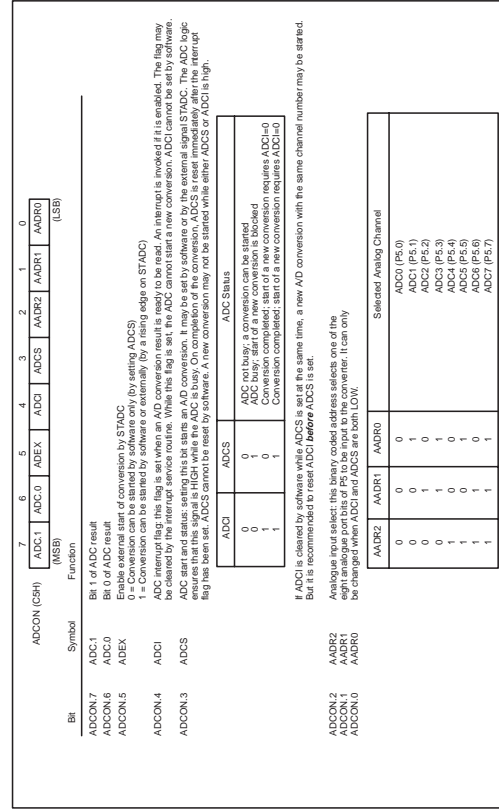


Figure 37. ADC Control Register (ADCON)



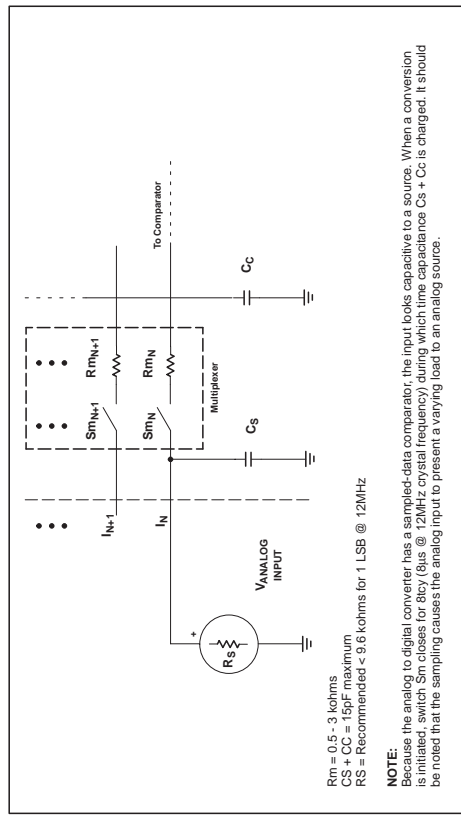


Figure 39. AD Input: Equivalent Circuit

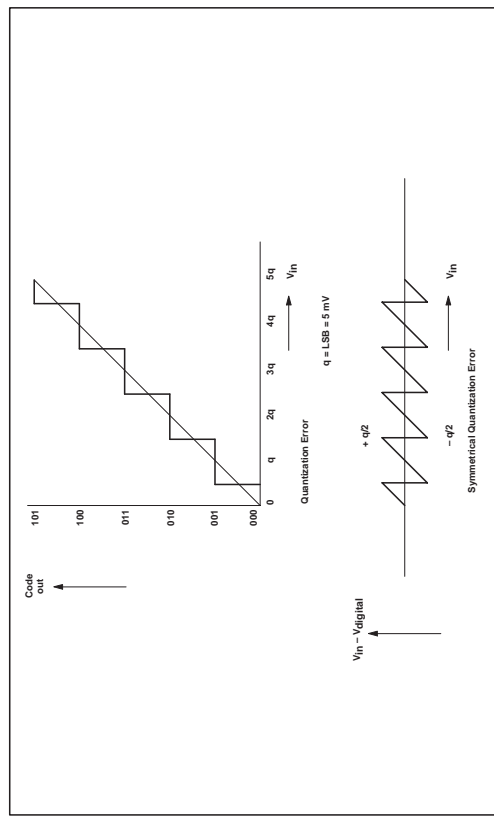


Figure 40. Effective Conversion Characteristic

Power Reduction Modes
The 8XC552 has two reduced power modes of operation: the idle mode and the power-down mode. These modes are entered by setting bits in the PCON special function register. When the 8XC552 enters the idle mode, the following functions are disabled:

- CPU (halted)
- Timer T2 (halted and reset)
- PWM0, PWM1 (reset; outputs are high) (conversion aborted if in progress).

In idle mode, the following functions remain active:

- Timer 0
- Timer 1
- Timer T3
- SIC0/SIO1
- External interrupts

When the 8XC552 enters the power-down mode, the oscillator is stopped. The power-down mode is entered by setting the PD bit in the PCON register. The PD bit can only be set if the ETW input is tied HIGH.

ADC Resolution and Analog Supply: Figure 38 shows how the ADC is realized. The ADC has its own supply pins (AV_{DD} and AV_{SS}) and two pins (V_{ref+} and V_{ref-}) connected to each end of the DAC's resistance-ladder. The ladder has 1023 equally spaced taps, separated by a resistance of R. The first tap is located 0.5 × R above V_{ref-}, and the last tap is located 1.5 × R below V_{ref+}. This gives a total ladder resistance of 1024 × R. This structure ensures that the DAC is monotonic and results in a symmetrical quantization error as shown in Figure 40.

For input voltages between V_{ref-} and (V_{ref+} + 1/2 LSB), the 10-bit result of an AD conversion will be 00 0000 0000B = 000H. For input voltages between (V_{ref+} - 3/2 LSB and V_{ref+}, the result of a conversion will be 11 1111 1111B = 3FFH. AV_{ref+} and AV_{ref-} may be between AV_{DD} + 0.2V and AV_{SS} - 0.2V. AV_{ref+} should be positive with respect to AV_{ref-}, and the input voltage (V_{in}) should be between AV_{ref+} and AV_{ref-}. If the analog input voltage range is from 2V to 4V, then 10-bit resolution can be obtained over this range if AV_{ref+} = 4V and AV_{ref-} = 2V.

The result can always be calculated from the following formula:

$$\text{Result} = 1024 \frac{V_{in} - AV_{ref-}}{AV_{ref+} - AV_{ref-}}$$

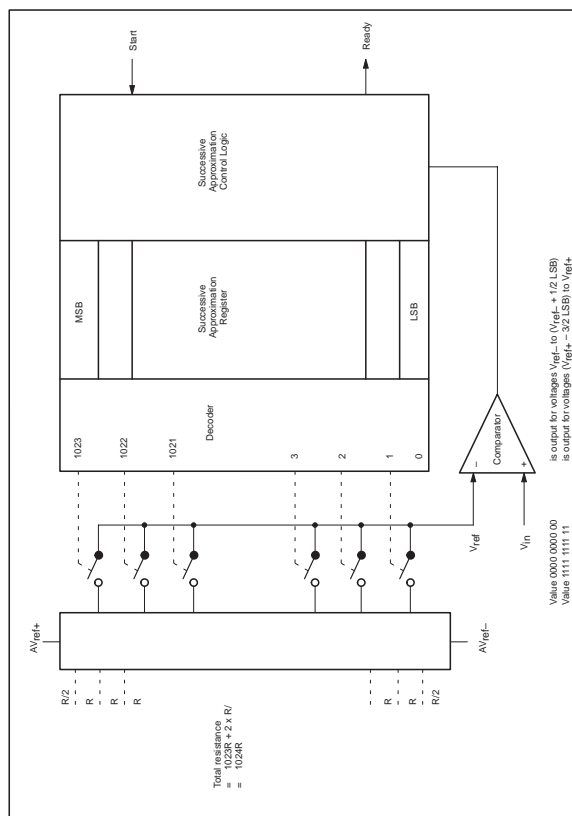


Figure 38. ADC Realization

Power-Down Mode: The instruction that sets PCON.1 will be the last instruction executed in the normal operating mode before the power-down mode is entered. In the power-down mode, the on-chip oscillator is stopped. This freezes all functions; only the on-chip RAM and special function registers are held. The ports pins output the contents of their respective special function registers. A hardware reset is the only way to terminate the power-down mode. Reset re-defines all the special function registers, but does not change the on-chip RAM.

In the power-down mode, V_{DD} and AV_{DD} can be reduced to minimize power consumption. V_{DD} and AV_{DD} must not be reduced before the power-down mode is entered and must be restored to the normal operating voltage before the power-down mode is terminated. The reset that terminates the power-down mode also freezes the oscillator. The reset should not be activated before V_{DD} and AV_{DD} are restored to their normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10ms).

The status of the external pins during power-down is shown in Table 11. If the power-down mode is entered while the 8XC552 is executing out of external program memory, the port data that is held in the P2 special function register is restored to port 2. If a port latch contains a "1", the port pin is held HIGH during the power-down mode by the strong pull-up transistor.

Power Control Register PCON: The idle and power-down modes are entered by writing 0 bits in PCON. PCON is not bit addressable. See Figure 41.

Memory Organization

The memory organization of the 8XC552 is the same as in the 80C51, with the exception that the 8XC552 has 8k ROM, 256 bytes RAM, and additional SFRs. Addressing modes are the same in the 8XC552 and the 80C51. Details of the differences are given in the following paragraphs.

In the 8XC552, the lower 8k of the 64k program memory address space is filled by internal ROM. By tying the EA pin high, the

processor fetches instructions from internal program ROM. Bus expansion for accessing program memory from 8k upwards is automatic since external instruction fetches occur automatically when the program counter exceeds 8191. If the EA pin is tied low, all program memory fetches are from external memory. The execution speed of the 8XC552 is the same regardless of whether fetches are from external or internal program memory. If all storage is on-chip, then byte location 8191 should be left vacant to prevent an undesired pre-fetch from external program memory address 8192.

Certain locations in program memory are reserved for specific programs. Locations 0000H to 0002H are reserved for the initialization program. Following reset, the CPU always begins execution at locations 0000H. Locations 0003H to 0075H are reserved for the fifteen interrupt request service routines.

Functionally, the internal data memory is the most flexible of the address spaces. The internal data memory space is subdivided into a 256-byte internal data RAM address space and a 128-byte special function register (SFR) address space, as shown in Figure 42.

The internal data RAM address space is 0 to 255. Four 8-bit register banks occupy locations 0 to 31, 128 bit locations of the internal data RAM are accessible through direct addressing. These 128 reside in 16 bytes of internal data RAM at locations 200 to 2FFH. The stack can be located anywhere in the internal data RAM address space by loading the 8-bit stack pointer. The stack depth may be 256 bytes maximum.

The SFR address space is 128 to 255. All registers except the program counter and the four 8-bit register banks reside in this address space. Memory mapping the SFRs allows them to be accessed as easily as internal RAM, and as such, they can be operated on by most instructions. The 56 SFRs are listed in Figure 43, and their mapping in the SFR address space is shown in Figures 44 and 45. RAM bit addresses are the same as in the 80C51 and are summarized in Figure 46. The special function bit addresses are summarized in Figure 47.

Table 11. External Pin Status During Idle and Power-Down Modes

MODE	MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	PWM0/PWM1
Idle (I)	Internal	1	1	Port data	Port data	Port data	Port data	Port data	HIGH
Idle (I)	External	1	1	Floating	Port data	Address	Port data	Port data	HIGH
Power-down	Internal	0	0	Port data	Port data	Port data	Port data	Port data	HIGH
Power-down	External	0	0	Floating	Port data	Port data	Port data	Port data	HIGH

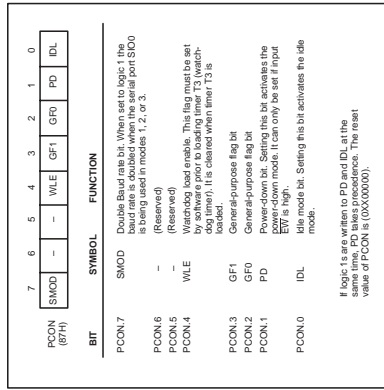


Figure 41. Power Control Register (PCON)

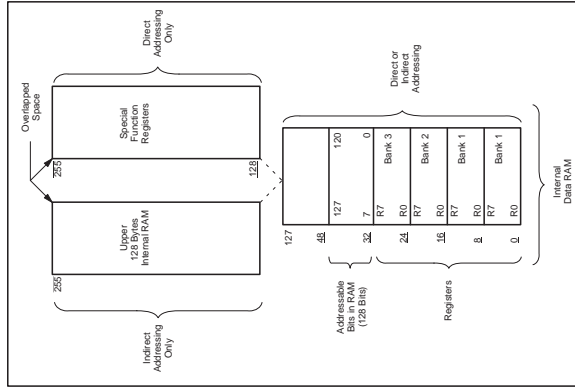


Figure 42. Internal Data Memory Address Space

- ARITHMETIC REGISTERS:
 - Accumulator
 - Program Status Word
 - POINTERS:
 - Stack Pointer
 - Data Pointer (High and Low)
 - PARALLEL I/O PORTS:
 - Port 5: Port 4, Port 3, Port 2, Port 1, Port 0
 - INTERRUPT SYSTEM:
 - Interrupt Enable 1*
 - Interrupt Enable 0*
 - Interrupt Enable 1*
 - Interrupt Enable 0*
 - Timer High 2; Timer T3
 - Timer High 0; Timer High 1
 - Timer Low; Timer Low 1
 - Timer Low; Timer Low 2
 - Timer High 2; Timer T3
 - TIMEBASES:
 - Timer Mode; Timer Control*
 - Timer Low 0; Timer High 0
 - Timer Low; Timer Low 1
 - Timer Low; Timer Low 2
 - Timer High 2; Timer T3
 - CAPTURE AND COMPARE LOGIC:
 - Timer T2 Interrupt Register
 - Capture Low 0; Capture High 0
 - Capture Low 1; Capture High 1
 - Capture Low 2; Capture High 2
 - Capture Low 3; Capture High 3
 - Compare Low 0; Compare High 0
 - Compare Low 1; Compare High 1
 - Compare Low 2; Compare High 2
 - Set Enable; Reset Enable
 - ADC:
 - ADC Control; ADC High byte
 - ADC Low byte
- *NOTE: Bit and byte addressable

Figure 43. Special Function Registers



80C51 Family Derivatives

8XC552/562 overview

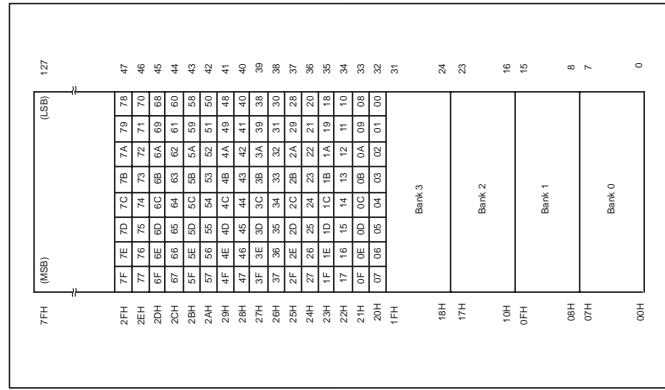


Figure 46. RAM Bit Addresses

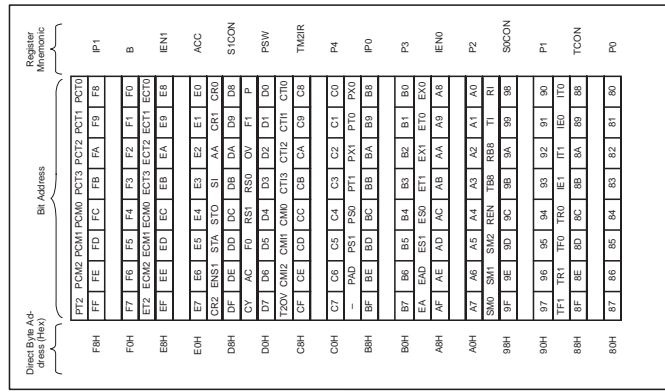


Figure 47. Special Function Register Bit Address

80C51 Family Derivatives

8XC552/562 overview

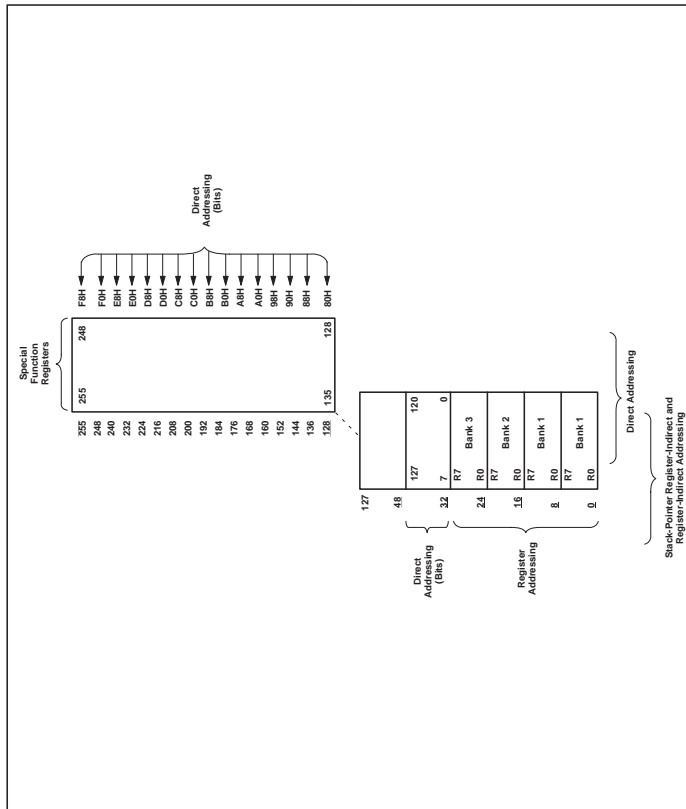


Figure 45. Bit and Byte Addressing Overview of Internal Data Memory



APPENDICE C: INDICE ANALITICO

A

A/D 2, 6, 9, 14, 15, 24, 26, 45

Alimentazione 2, 3, 10, 25

B

Back Up 17, 33

Batteria 10

Bibliografia 50

Buzzer 2, 6, 9, 42

C

Caratteristiche elettriche 10

Caratteristiche generali 2, 9

Comunicazione seriale 4

Connettori 9, 23

CN1 9, 12

CN2 9, 13

CN3 9, 11

CN4 9, 14

CN5 9, 16

CN6 9, 18

CN7 9, 20

CN8 9, 17

Current Loop 2, 4, 9, 32

D

Descrizione hardware 35

Descrizione software 34

Dip switch 6, 9, 23, 43

Dispositivi di clock 3

Dispositivi di memoria 4

Dispositivi periferici di bordo 6

DSW1 24, 43

E

EEPROM 2, 4, 6, 9

EEPROM seriale 44

EPROM 2, 4, 9

F

Formato 2

I

I/O ABACO® 2
I²C bus 2, 9, 16
Idle-Mode 2
Input di bordo 24
Installazione 11
Interfacce operatore locali 26
Introduzione 1

J

Jumpers 27, 29
2 vie 28
3 vie 30
4 vie 31
5 vie 31

L

LED di attività 42
LEDs 9, 22, 23
Logica di controllo 7

M

MAPPAGGIO 1 37
MAPPAGGIO 2 38
MAPPAGGIO 3 39
MAPPAGGIO 4 40
Mappaggio delle memorie 36
Mappaggio dell'I/O 36
Memorie 54

N

Note 31

P

Pianta Componenti 8
Power-Down Mode 2
PPI 82C55 6, 12, 13, 31, 44
Processore di bordo 3
Pull Up 43
PWM 2, 6, 9, 12, 45

Q

Quarzo 2

R

RAM 2, 4, 9
RS 232 2, 4, 9, 32
RS 422 2, 32
RS 422-485 4, 9, 43
RS 485 2, 32
RTC 2, 9, 44

S

Schede esterne per GPC® 552 46
Schema a blocchi 5
Seconda seriale 16
Segnalazioni visive 22
Selezione memorie 33
Selezione modo e alimentazione delle reti resistiv 31
Selezione tipo ingressi analogici 25
Seriale 20, 21, 32, 53
SIO 0 6
SIO 1 6
Software 2
Specifiche tecniche 9

T

Tasto di Reset 7, 24
Temperatura 9
Test Point 24
Trimmers e tarature 26

U

UART 45
Uscite generiche 42

V

Versione scheda 1

W

Watch Dog 2, 6, 9, 45

Z

Zoccoli 2

