

# CAN AVR

CAN - grifo® Mini Modulo AVR

## MANUALE TECNICO



**grifo®**

ITALIAN TECHNOLOGY

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

E-mail: [grifo@grifo.it](mailto:grifo@grifo.it)

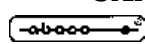
<http://www.grifo.it>

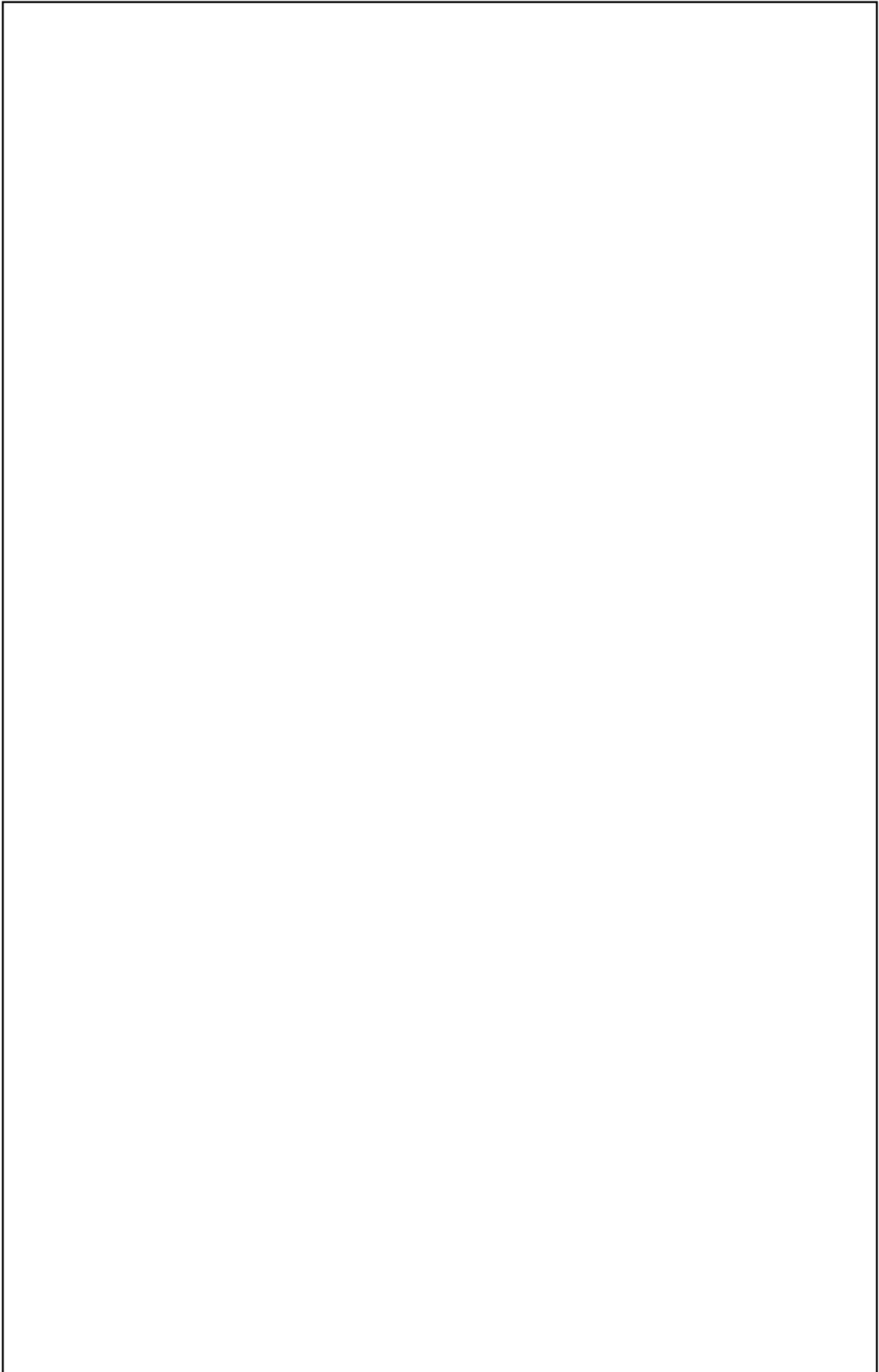
<http://www.grifo.com>

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



CAN AVR Rel. 3.00 Edizione 17 Gennaio 2006

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



# CAN AVR

CAN - grifo® Mini Modulo AVR

## MANUALE TECNICO

**Zoccolo** maschio **28** piedini dual in line a passo **100** mils, largo 600 mils; ridottissimo ingombro: 42 x 25 x 15 mm; Unica alimentazione a **5 Vdc** (l'assorbimento puo' variare in base ai collegamenti del modulo); disponibilita' di **Idle Mode** e **Power Down Mode**; microcontrollore **Atmel AT90CAN128** (AVR RISC) con quarzo da **12 Mhz**; **128K** bytes **FLASH** per codice, fino a **8K bytes FLASH** riservabile per **Boot Loader**, **4K bytes SRAM** per dati, **4K bytes EEPROM** per dati; **A/D Converter** da **5** canali con **10 bit** di risoluzione a **6 µsec** per conversione; 2 canali differenziali con amplificatore a guadagno programmabile che permettono di effettuare facilmente conversioni **A/D** di tipo bipolare; 37 sorgenti di **Interrupt**; **4 Timer Counter** a 8 o 16 bits con funzionalita' di **PWM**, comparazione, cattura, ecc.; **24** linee di **I/O** digitale; una linee seriale hardware con Baud Rate programmabile fino a 115.200 Baud, bufferata in **RS 232 o TTL**; una linea seriale differenziale **CAN** hardware totalmente compatibile con le specifiche **CAN 2.0** parti **A** e **B**; interfaccia **JTAG** per il debugging remoto in circuit; interfaccia **I<sup>2</sup>C BUS**; interfaccia **SPI**; circuiteria di **Reset**; **Watch Dog** programmabile da 16 msec fino a 2 sec; due dip switch di configurazione a 8 vie e a 4 vie; **Real Time Clock** in grado di gestire giorno, mese, anno, giorno della settimana, ore, minuti, secondi e di generare interrupt periodici; **240** bytes di SRAM per parametri di configurazione; **RTC** e **SRAM** tamponati con batteria al **Litio** di bordo; **2 LEDs** di segnalazione gestiti via software tramite due linee di **I/O** digitale; **Boot Loader** preinstallato che permette di programmare la FLASH e la EEPROM tramite la porta seriale **RS 232**; possibilita' di gestione della FLASH ed EEPROM interna in modalita' **In System Programming**, ovvero con modulo gia' montato, sfruttando la modalita' di programmazione **ISP**; vasta disponibilita' di software di sviluppo quali: **Assemblatori**; compilatori **C** (ICC AVR, DDS Micro C); **Compilatori BASIC** (BASCOM AVR); ecc.; vasta serie di programmi dimostrativi ed esempi di utilizzo forniti sotto forma di sorgenti ampiamente commentati, per i vari ambienti di sviluppo.

**grifo**®

ITALIAN TECHNOLOGY

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

E-mail: grifo@grifo.it

<http://www.grifo.it>

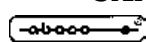
Tel. +39 051 892.052 (r.a.)

<http://www.grifo.com>

FAX: +39 051 893.661



CAN AVR Rel. 3.00 Edizione 17 Gennaio 2006

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

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

## IMPORTANTE

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

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

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

## LEGENDA SIMBOLI

Nel presente manuale possono comparire i seguenti simboli:



Attenzione: Pericolo generico

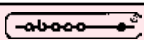


Attenzione: Pericolo di alta tensione



Attenzione: Dispositivo sensibile alle cariche elettrostatiche

## Marchi Registrati



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

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

# INDICE GENERALE

INTRODUZIONE .....	1
VERSIONE SCHEDA .....	3
INFORMAZIONI GENERALI .....	4
LINEE DI I/O DIGITALE .....	6
COMUNICAZIONE SERIALE .....	6
CLOCK .....	6
DISPOSITIVI DI MEMORIA .....	8
LINEA I <sup>2</sup> C BUS .....	8
LINEA SPI .....	8
LINEA CAN .....	9
DIP SWITCH .....	9
A/D CONVERTER .....	10
COMPARATORE ANALOGICO .....	10
TIMER COUNTER E PWM .....	10
WATCH DOG .....	10
REAL TIME CLOCK .....	10
SPECIFICHE TECNICHE .....	12
CARATTERISTICHE GENERALI .....	12
CARATTERISTICHE FISICHE .....	13
CARATTERISTICHE ELETTRICHE .....	13
INSTALLAZIONE .....	14
SEGNALAZIONI VISIVE .....	14
CONNESSIONI .....	14
CN1 - CONNETTORE CON SEGNALI DEL MINI MODULO .....	14
CONFIGURAZIONE MINI MODULO .....	16
SELEZIONE COMUNICAZIONE SERIALE .....	18
INTERFACCIAMENTO CONNETTORI CON IL CAMPO .....	20
ALIMENTAZIONE .....	20
INTERRUPTS .....	21
INTERFACCIA JTAG .....	21
SCHEDE DI SUPPORTO .....	22
UTILIZZO CON IL MODULO BLOCK GMB HR84 .....	22
UTILIZZO CON LA SCHEDA CAN GMT .....	24
COME INIZIARE .....	26
RICOMPILAZIONE CON BASCOM AVR. ....	32
RICOMPILAZIONE CON ICC AVR. ....	32

DESCRIZIONE SOFTWARE .....	34
DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO .....	36
LED DI ATTIVITA' .....	36
DIP SWITCH .....	36
SRAM TAMPONATA + RTC SERIALE .....	37
PERIFERICHE DELLA CPU .....	37
BIBLIOGRAFIA .....	38
APPENDICE A: DESCRIZIONE COMPONENTI DI BORDO .....	A-1
AT90CAN128 .....	A-1
PCF 8583.....	A-2
APPENDICE B: PROGRAMMAZIONE ISP CON GMM TST 2 .....	B-1
OPERAZIONI COMUNI PER TUTTI I PROGRAMMATORI .....	B-1
PROGRAMMAZIONE CON <b>GRIFO</b> <sup>®</sup> MP-AVR 51+ ; <b>GRIFO</b> <sup>®</sup> UEP 49 .....	B-4
PROGRAMMAZIONE CON EQUINOX EPSILON5 .....	B-7
PROGRAMMAZIONE CON ATMEL AVR ISP .....	B-9
APPENDICE C: SCHEMA ELETTRICO CAN GMT .....	C-1
APPENDICE D: INDICE ANALITICO .....	D-1

# INDICE DELLE FIGURE

<b>FIGURA 1: POSIZIONE DEL NUMERO DI REVISIONE .....</b>	<b>3</b>
<b>FIGURA 2: SCHEMA A BLOCCHI .....</b>	<b>7</b>
<b>FIGURA 3: FOTO CAN AVR .....</b>	<b>9</b>
<b>FIGURA 4: PIANTA COMPONENTI (LATO COMPONENTI) .....</b>	<b>11</b>
<b>FIGURA 5: PIANTA COMPONENTI (LATO SALDATURE).....</b>	<b>11</b>
<b>FIGURA 6: TABELLA DELLE SEGNALAZIONI VISIVE .....</b>	<b>14</b>
<b>FIGURA 7: CN1 - ZOCCOLO CON SEGNALI DEL MINI MODULO.....</b>	<b>15</b>
<b>FIGURA 8: TABELLA DIP SWITCH DSW2 A 4 VIE .....</b>	<b>16</b>
<b>FIGURA 9: TABELLA DIP SWITCH DSW1 AD 8 VIE .....</b>	<b>17</b>
<b>FIGURA 10: ESEMPIO COLLEGAMENTO SERIALE RS 232.....</b>	<b>19</b>
<b>FIGURA 11: ESEMPIO COLLEGAMENTO SERIALE TTL .....</b>	<b>19</b>
<b>FIGURA 12: DISPOSIZIONE LEDs, DIP SWITCHES, ECC. ....</b>	<b>21</b>
<b>FIGURA 13: FOTO DI GMB HR84 E CAN AVR .....</b>	<b>23</b>
<b>FIGURA 14: ESEMPIO DI COLLEGAMENTO DI CAN GMT CON CAN AVR.....</b>	<b>25</b>
<b>FIGURA 15: COLLEGAMENTO SERIALE RS 232 TRA CAN AVR E PC .....</b>	<b>26</b>
<b>FIGURA 16: TABELLA ESEMPI.....</b>	<b>27</b>
<b>FIGURA 17: BOOT LOADER UTILITY .....</b>	<b>29</b>
<b>FIGURA 18: CARICAMENTO SORGENTE CON BASCOM AVR .....</b>	<b>30</b>
<b>FIGURA 19: CONFIGURAZIONE COMPILATORE BASCOM AVR .....</b>	<b>31</b>
<b>FIGURA 20: COMPILAZIONE CON BASCOM AVR .....</b>	<b>31</b>
<b>FIGURA 21: CARICAMENTO FILE DI PROGETTO CON ICC AVR.....</b>	<b>32</b>
<b>FIGURA 22: COMPILAZIONE CON ICC AVR .....</b>	<b>32</b>
<b>FIGURA 23: VISTA DALL'ALTO E DAL BASSO DELLA CAN AVR .....</b>	<b>35</b>
<b>FIGURA 24: SCHEMA DELLE POSSIBILI CONNESSIONI .....</b>	<b>39</b>
<b>FIGURA B-1: SCHEMA DELL'INTERFACCIA TRA GMM TST 2 E MP-AVR 51+ .....</b>	<b>B-3</b>
<b>FIGURA B-2: SCHEMA DELL'INTERFACCIA TRA GMM TST 2 E UEP 49 .....</b>	<b>B-3</b>
<b>FIGURA B-3: SELEZIONE DEL COMPONENTE CON PG4UW .....</b>	<b>B-4</b>
<b>FIGURA B-4: ZOCCOLO DI PROGRAMMAZIONE ISP CAN AVR PER GMM TST 2 .....</b>	<b>B-5</b>
<b>FIGURA B-5: CONFIGURAZIONE DEL COMPONENTE CON PG4UW .....</b>	<b>B-6</b>
<b>FIGURA B-6: CONFIGURAZIONE DEL PROGRAMMATORE PG4UW .....</b>	<b>B-6</b>
<b>FIGURA B-7: CONFIGURAZIONE PROGETTO CON EQTOOLS .....</b>	<b>B-8</b>
<b>FIGURA B-8: SELEZIONE CPU CON AVR STUDIO .....</b>	<b>B-10</b>
<b>FIGURA B-9: CONFIGURAZIONE CPU CON AVR STUDIO .....</b>	<b>B-10</b>
<b>FIGURA B-10: CONFIGURAZIONE AVR ISP .....</b>	<b>B-10</b>
<b>FIGURA C-1: SCHEMA ELETTRICO CAN GMT .....</b>	<b>C-1</b>





## INTRODUZIONE

L'uso di questi dispositivi è rivolto - **IN VIA ESCLUSIVA** - a personale specializzato.  
Questo prodotto non è un **componente di sicurezza** così come definito dalla direttiva **98-73/CE**.



I pin del Mini Modulo non sono dotati di protezione contro le cariche elettrostatiche. Esiste un collegamento diretto tra i pin del Mini Modulo e i rispettivi pin del microcontrollore. Il Mini Modulo è sensibile ai fenomeni ESD.

Il personale che maneggia i Mini Moduli è invitato a prendere tutte le precauzioni necessarie per evitare i possibili danni che potrebbero derivare dalle cariche elettrostatiche.

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

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

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

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

Per un corretto rapporto coi prodotti, 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 è riferito alla scheda **CAN AVR** revisione **220205**.

La validità delle informazioni riportate è quindi subordinata al numero di revisione della scheda in uso e l'utente deve quindi sempre verificarne la giusta corrispondenza.

Tali versioni sono sempre riportate sullo stampato in più punti e la seguente figura illustra la posizione più facilmente accessibile.

## NUMERO DI REVISIONE DELLO STAMPATO

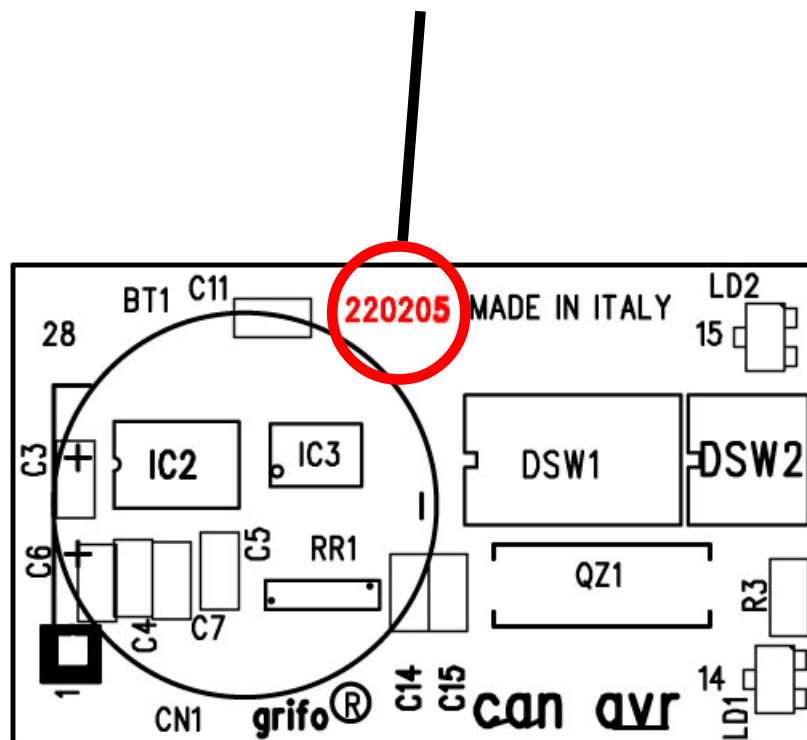


FIGURA 1: POSIZIONE DEL NUMERO DI REVISIONE

## INFORMAZIONI GENERALI

Il modulo **CAN AVR** (CAN - **grifo**<sup>®</sup> Mini Modulo AVR), è basato sul microcontrollore **Atmel AT90CAN128**, un potente e completo sistema on-chip dotato di **CPU**, **memoria integrata** sia per il codice da eseguire sia per i dati, **A/D converter**, **watch dog**, **interrupts**, linee di I/O digitali TTL, una linea seriale hardware, **timer/counter** dedicati con capacità di capture/compare e **PWM**, ecc. Il modulo ha già montati nella sua ridottissima area alcuni componenti che servono a valorizzare le principali caratteristiche del microcontrollore e, oltre a questo, monta ulteriori circuiterie che ne integrano le funzionalità, come quella che genera il segnale di reset.

Le **possibili applicazioni** del modulo **CAN AVR** sono innumerevoli.

Ci sono le applicazioni native del CAN, cioè l'**automatizzazione** del **controllo** nel settore **automobilistico** (accensione e spegnimento luci, controllo riscaldamento e condizionamento, supervisione servizi elettrici, sistemi antifurto, diagnostica di funzionamento, ecc.). Si possono citare anche il **collegamento** sulle reti **CAN** con **protocolli proprietari** o con protocolli standard (CANopen, DeviceNet, SDS, CAN Kingdom ecc.). Si può avere anche il **funzionamento** come piccoli **nodi intelligenti** con funzionalità locali come il controllo con algoritmi PID di temperature, motori, valvole o come **sistemi a logica distribuita** tipo robot, automazioni su macchine di produzione in linea, automazioni di fabbriche di grosse dimensioni. Infine la **teleacquisizione** e il **telecontrollo** su medio brevi distanze, la **conversione** tra **CAN** e **seriale asincrona** o linea **I<sup>2</sup>C BUS** e l'**automazione domestica** (accensione e spegnimento luci, controllo riscaldamento e condizionamento, supervisione elettrodomestici e servizi elettrici, sistemi di sorveglianza e controllo accesso).

Da non dimenticare il **settore didattico**; infatti la **CAN AVR** offre la possibilità di apprendere il funzionamento di un microcontrollore **RISC** con core famiglia AVR e di sviluppare le sue applicazioni canoniche ad un costo veramente basso.

A questo scopo è ideale la scheda di supporto **CAN GMT**, che permette di collegare immediatamente una porta seriale RS 232 per il collegamento con un PC, una tensione di riferimento per l'A/D Converter e la linea CAN.

Il modulo **CAN AVR** viene fornito con un programma di Boot Loader preinstallato, che permette di riprogrammare la FLASH e la EEPROM tramite una semplice porta seriale RS 232 (ad esempio, quella del PC).

L'elettronica del microcontrollore di bordo ha inoltre il pieno supporto alla diffusa interfaccia **JTAG**, che permette di controllarne il funzionamento direttamente sull'applicativo mentre è in funzione.

In questo caso il debugging viene controllato tramite il PC ed una apposita interfaccia da connettere a sei segnali del Mini Modulo comodamente disponibili sullo zoccolo.

Il programma PC usato per effettuare questo controllo è **AVR Studio**, il famoso IDE creato da ATMEL e distribuito gratuitamente tramite il loro sito Internet.

In tutti i casi di scarso tempo di sviluppo **l'utente puo' avere il suo prototipo** o addirittura il prodotto finito **nel giro di una settimana**.

Le caratteristiche di massima del modulo **CAN AVR** sono:

- **Zoccolo** maschio **28** piedini dual in line a passo **100** mils, largo 600 mils
- Ridottissimo ingombro: 42 x 25 x 15 mm
- Unica alimentazione a **+5,0 Vdc** (l'assorbimento puo' variare in base ai collegamenti del modulo)
- Disponibilita' di **idle mode** e **power down mode**
- Microcontrollore **Atmel AT90CAN128** (AVR RISC) con quarzo da **12 Mhz**

- **128K bytes FLASH** per codice, **fino a 8K bytes FLASH riservabile per boot loader**, **4K bytes SRAM** per dati, **4K bytes EEPROM** per dati
- **A/D converter da 5 canali** con **10 bit** di risoluzione a **6 µsec** per conversione
- 2 canali differenziali con amplificatore a guadagno programmabile che permettono di effettuare facilmente conversioni A/D di tipo bipolare
- 37 sorgenti di **interrupt**
- **4 Timer Counter** a 8 o 16 bits con funzionalità di PWM, comparazione, cattura, ecc.
- **24 linee di I/O** digitale
- Una linea seriale hardware con Baud Rate programmabile fino a 115200 Baud, bufferata in **RS 232** o TTL
- Una linea seriale differenziale **CAN** hardware totalmente compatibile con le specifiche **CAN 2.0** parti A e B
- Interfaccia **JTAG** per il debugging remoto in circuit
- Interfaccia **I<sup>2</sup>C BUS**
- Interfaccia **SPI**
- Circuiteria di **Reset**
- **Watch dog** programmabile da 16 msec fino a 2 sec
- Due dip switch di configurazione a 8 vie e a 4 vie
- **Real Time Clock** in grado di gestire giorno, mese, anno, giorno della settimana, ore, minuti, secondi e di generare interrupt periodici
- **240 bytes** di SRAM per parametri di configurazione
- **RTC** e **SRAM** tamponati con batteria al **Litio** di bordo
- **2 LEDs** di segnalazione gestiti via software tramite due linee di I/O digitale
- **Boot Loader** preinstallato che permette di programmare la FLASH e la EEPROM tramite una delle porte seriali RS 232
- Possibilità di gestione della FLASH ed EEPROM interna in modalità **In System Programming**, ovvero con modulo già montato, sfruttando la modalità di programmazione **ISP**
- Vasta disponibilità di software di sviluppo quali: **Assemblatori**; compilatori **C** (ICC AVR, DDS Micro C); **compilatori BASIC** (BASCOS AVR); ambienti intergrati (**AVR Studio**) ecc.
- Ricca serie di programmi dimostrativi ed esempi di utilizzo forniti sotto forma di sorgenti ampiamente commentati, per i vari ambienti di sviluppo.

Viene di seguito riportata una descrizione dei blocchi funzionali della scheda, con indicate le operazioni effettuate da ciascuno di essi. Per una più facile individuazione di tali blocchi e per una verifica delle loro connessioni, fare riferimento alla figura 2.

## LINEE DI I/O DIGITALE

Il Mini Modulo **CAN AVR** mette a disposizione 24 linee di I/O digitale TTL del microcontrollore Atmel AT90CAN128, ovvero tutti i segnali dei Port PB, i segnali da 0 a 6 di PE, i segnali 0, 1, 4 e 7 di PD ed i segnali 0 e da 4 a 7 di PF.

Tali linee sono collegate direttamente al connettore a 28 vie con pin out standard **grifo**<sup>®</sup> Mini Modulo ed hanno quindi la possibilità di essere direttamente collegate a numerose schede d'interfaccia.

Via software è definibile ed acquisibile la funzionalità e lo stato di queste linee, con possibilità di associarle anche alle periferiche della scheda (Timer Counter, Interrupt, I<sup>2</sup>C BUS, SPI, ecc.), tramite una semplice programmazione di alcuni registri interni della CPU.

Per maggiori informazioni fare riferimento ai paragrafi **CONNESSIONI** e **DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO**.

## COMUNICAZIONE SERIALE

La scheda dispone di una linea seriale hardware in cui il protocollo fisico (baud rate, stop bit, bit x chr, ecc.) è completamente settabile via software tramite la programmazione dei registri interni.

Quindi per ulteriori informazioni si faccia riferimento alla documentazione tecnica del microcontrollore della casa costruttrice o alle appendici di questo manuale.

La linea seriale è collegata al connettore CN1 a livello TTL o RS 232, grazie alla configurazione di alcuni dip switch di bordo quindi, quando la scheda deve essere collegata in una rete, collegata a distanza, o collegata ad altri dispositivi che usano diversi protocolli elettrici, si deve interporre un apposito driver seriale esterno (RS 422, RS 485, Current loop, ecc.).

Sul connettore CN1 oltre alle linee di ricezione e trasmissione sono disponibili anche altre linee di I/O gestibili via software che possono essere usate per definire la direzione della linea in caso di RS 485, per abilitare il driver di trasmissione in caso di RS 422 oppure come handshake hardware in caso di RS 232. Ad esempio può essere utilizzato il modulo **MSI 01** che è in grado di convertire la linea seriale TTL in qualsiasi altro standard elettrico in modo comodo ed economico.

Per maggiori informazioni contattare direttamente la **grifo**<sup>®</sup> e leggere il paragrafo **SELEZIONE COMUNICAZIONE SERIALE**.

## CLOCK

Nel modulo **CAN AVR** vi sono due circuiterie separate ed indipendenti basate su due quarzi, che si occupano della generazione dei segnali di clock per il microcontrollore e per l'RTC di bordo.

Il primo genera una frequenza di 12 MHz mentre il secondo genera una frequenza di 32768 Hz.

La scelta di disporre di due circuiterie di clock distinte serve a ridurre i costi nella maggioranza delle applicazioni di medio alta velocità e di poter aumentare notevolmente le prestazioni nelle applicazioni che lo richiedono.

Dal punto di vista delle prestazioni si ricorda che **CAN AVR** ha installato un microprocessore di tipo RISC, in grado di eseguire, mediamente, circa una istruzione ogni ciclo di clock.

Pertanto, tenendo conto del quarzo montato, la velocità di esecuzione può arrivare a quasi 12 MIPS.

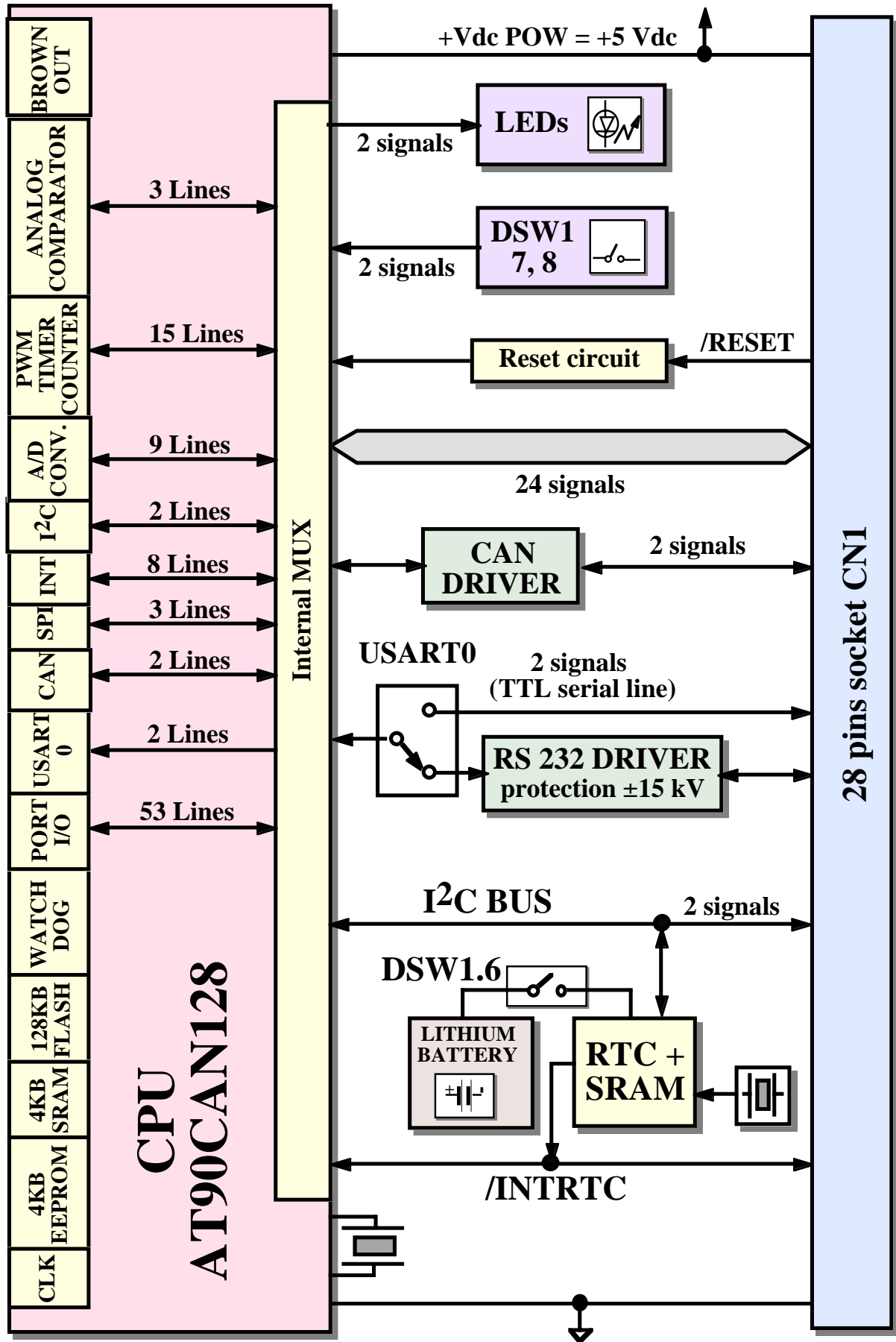


FIGURA 2: SCHEMA A BLOCCHI

## DISPOSITIVI DI MEMORIA

La scheda è dotata di un massimo di 136,25KByte di memoria variamente suddivisi con un massimo di **128K**Bytes FLASH EPROM, **4K**Bytes di SRAM interna, **4K**Bytes di EEPROM e 240 Bytes di SRAM nel modulo RTC, questi ultimi raggiungibili solo via I<sup>2</sup>C BUS.

La scelta d'uso delle memorie da utilizzare può avvenire in relazione all'applicazione da risolvere e quindi in relazione alle esigenze dell'utente.

Grazie alla EEPROM di bordo (ed alla SRAM dell'RTC, quando la batteria viene inserita) c'è inoltre 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.

Qualora la quantità di memoria per dati risulti insufficiente (ad esempio per sistemi di data login) si possono sempre collegare dei dispositivi esterni di memoria nelle tecnologie SRAM, EEPROM e FLASH tramite le comode ed efficienti interfacce SPI ed I<sup>2</sup>C BUS della scheda.

Il mappaggio e la gestione delle risorse di memoria avviene direttamente all'interno del microcontrollore come descritto nella documentazione del componente o nell'APPENDICE A di questo manuale.

## LINEA I<sup>2</sup>C BUS

Il pin out standard **grifo® Mini Modulo** del connettore a 28 vie riserva due segnali, il 6 ed il 7, all'interfaccia I<sup>2</sup>C BUS. Questi segnali sono dotati di un pull-up da 4,7 kΩ che si trova a bordo del Mini Modulo.

Nel **CAN AVR** viene usata l'interfaccia hardware della CPU utilizzabile mediante i suoi registri interni. Questa può funzionare sia come master che come slave, in ricezione e trasmissione.

Grazie a questa interfaccia possono essere collegati dispositivi dotati dello stesso standard di comunicazione in modo da espandere localmente le potenzialità del Mini Modulo.

Una ricca serie di esempi software prevede la gestione delle più comuni e diffuse interfacce I<sup>2</sup>C BUS come A/D e D/A converter, memorie, sensori di temperatura, l'RTC di bordo, ecc.

Notevoli tra le periferiche connettabili in I<sup>2</sup>C BUS sono i pannelli operatore **grifo®** della serie **QTP**, in grado di gestire display sia grafici che alfanumerici e tastiere di varie dimensioni, a seconda del modello.

Le schede di supporto ai Mini Moduli prodotte dalla **grifo®** (come la **CAN GMT**) prevedono, tra le altre cose, anche un connettore dedicato all'I<sup>2</sup>C BUS, in modo da facilitare la massima le connessioni con il campo.

Si ricorda che il dispositivo RTC con SRAM tamponata occupa in maniera permanente lo slave address **A0**, pertanto le applicazioni utente non possono utilizzare questo slave address.

## LINEA SPI

Il **Mini Modulo grifo® CAN AVR** dispone di una linea seriale sincrona SPI hardware incorporata nel microcontrollore.

Sul connettore CN1 i segnali MISO, MOSI ed SCK dell'interfaccia SPI sono disponibili rispettivamente sui pins 8, 9 e 12. Poiché i segnali MISO e MOSI sono multiplexati con la linea CAN su CN1, bisogna agire sul dip switch DSW2 per scegliere quale interfaccia usare (vedere il paragrafo DIP SWITCH). Tutti i parametri di gestione si possono regolare mediante la programmazione dei registri interni. Per ulteriori informazioni si vedano i data sheet del componente o si consulti l'appendice A di questo manuale.



## LINEA CAN

Il Mini Modulo CAN AVR è dotato di una potente interfaccia CAN hardware, in grado di raggiungere il massimo bit rate di 1 Mbit/sec. I segnali CAN L e CAN H si trovano rispettivamente ai pin 8 e 9 di CN1.

L'interfaccia è pienamente compatibile con le specifiche CAN 2.0 parti A e B.

## DIP SWITCH

Il Mini Modulo CAN AVR è dotato di due dip switch di bordo, uno da otto vie ed uno da quattro vie, il cui scopo è l'impostazione di vari parametri elettrici del Mini Modulo stesso e della modalità di funzionamento della scheda.

Infatti il dip switch DSW1 permette di stabilire se l'interfaccia seriale **USART0** debba essere bufferata in **RS 232** o in **TTL**, permettendo in questo caso di bufferarle esternamente, ad esempio, in **RS 422**, **RS 485** o **current loop**, usando opportuni drivers.

Il dip switch a 4 vie DSW2 invece permette di selezionare se sui pin 8 e 9 di CN1 devono essere presenti i segnali dell'interfaccia CAN o dell'interfaccia SPI.

Inoltre DSW1 permette di collegare o meno la batteria di back up del modulo RTC + SRAM

Infine, lo stato di alcuni switch può essere acquisito tramite lettura di appositi segnali del microcontrollore (PA2 e PA3).

Per ulteriori informazioni si veda anche il paragrafo CONFIGURAZIONE MINI MODULO.



FIGURA 3: FOTO CAN AVR

## A/D CONVERTER

Il Mini Modulo **CAN AVR** è dotato un A/D converter da 5 canali, multiplexati sulle linee del port PF, ognuno dei quali ha risoluzione di 10 bit.

La tensione di riferimento può essere prelevata esternamente da un pin oppure internamente, da un apposito generatore. Il range dei segnali collegabili va da 0 a 5 Vdc, e devono essere sorgenti a bassa impedenza.

La fine della conversione può essere usata come trigger per un interrupt, se questo è abilitato.

La gestione della periferica avviene tramite l'opportuna manipolazione degli appositi registri interni del micro. Per ulteriori informazioni si vedano i data sheet nell'appendice A di questo manuale o si consultino i commenti nei programmi di esempio ad alto livello.

## COMPARATORE ANALOGICO

Il microcontrollore Atmel AT90CAN128 incorpora un comparatore analogico che può selezionare come input sia una tensione di riferimento interna che diversi pin (pin 10 e 11 di CN1, ma anche ingressi dell'A/D converter) mediante un commutatore interno.

Per ulteriori informazioni, si possono consultare le varie application notes disponibili sul sito Atmel.

## TIMER COUNTER E PWM

Il microcontrollore mette a disposizione quattro Timer/Counter (due ad otto bit, due a sedici bit), in grado di contare gli impulsi di clock (con prescaler programmabile), transizioni di livello su opportuni pin e generare interrupt.

Inoltre possono essere usati in modalità PWM, per generare dei segnali di frequenza e duty cycle definibile via software con risoluzione otto o sedici bit.

Le applicazioni tipiche di tali segnali sono il controllo della velocità dei motori, infatti molti azionamenti dispongono di azionamenti compatibili. Oppure la generazione di segnali analogici, tramite facilmente ottenibili aggiungendo un semplice circuito integratore.

## WATCH DOG

Il microcontrollore Atmel AT90CAN128 incorpora un watch dog hardware programmabile in grado di resettare la CPU se è abilitato ed il programma applicativo non lo retriggera entro il tempo stabilito. Il tempo di intervento può essere programmato da circa 16 millisecondi a 2 secondi.

## REAL TIME CLOCK

Il Mini Modulo dispone di un Real Time Clock (in I<sup>2</sup>C BUS allo slave address **A0**), in grado di gestire ore, minuti, secondi, anno, mese, giorno della settimana e giorno. Dispone inoltre di una memoria SRAM di 240 Byte.

Il componente è alimentato dalla circuiteria di back up in modo da garantire la validità dei dati in ogni condizione operativa ed è completamente gestito via software.

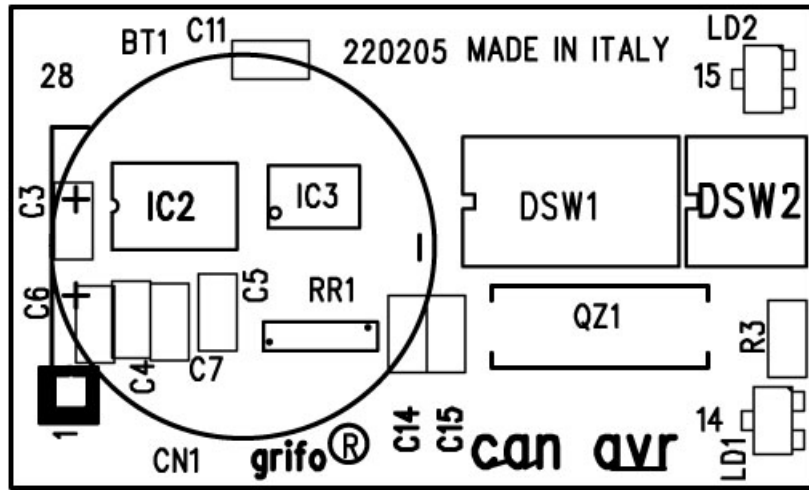


FIGURA 4: PIANTA COMPONENTI (LATO COMPONENTI)

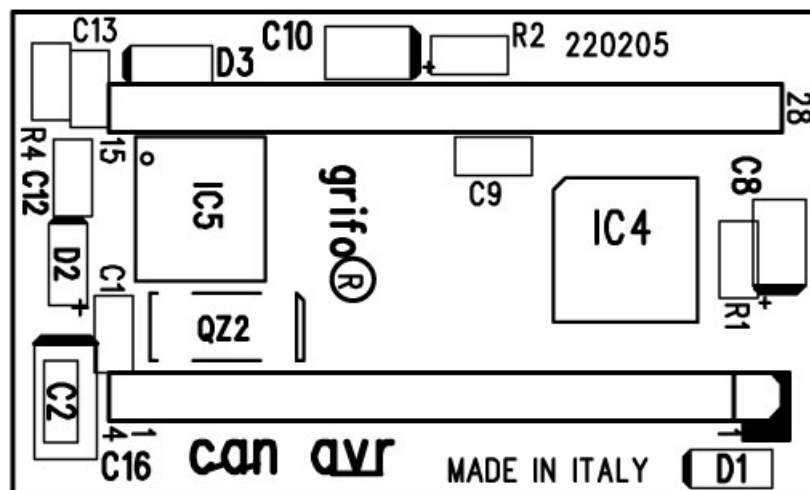


FIGURA 5: PIANTA COMPONENTI (LATO SALDATURE)

## SPECIFICHE TECNICHE

**CARATTERISTICHE GENERALI**

<b>Risorse di bordo:</b>	24 linee di I/O digitale 2 ingressi analogici su comparatori 5 canali di A/D converter 1 sezione Watch Dog 4 Timer Counter multifunzione (PWM, ecc.) 1 interfaccia SPI 1 interfaccia I <sup>2</sup> C BUS 1 interfaccia CAN 1 sezione di Real Time Clock 37 sorgenti di interrupt 1 linea seriale RS 232 o TTL 1 Dip Switch a 8 vie, 1 Dip Switch a 4 vie 2 LEDs di stato
<b>Memorie:</b>	128 KBytes FLASH EPROM per codice 4 KBytes SRAM interna per dati utente 4 KBytes EEPROM interna per dati utente 240 Bytes SRAM esterna (su I <sup>2</sup> C BUS) per dati utente
<b>CPU di bordo:</b>	Atmel AT90CAN128
<b>Frequenza clock:</b>	12 MHz
<b>Frequenza massima contatori:</b>	frequenza clock I/O
<b>Tempo di power on:</b>	da 37 msec a 93 msec impostabile con i bit di configurazione
<b>Tempo intervento watch dog:</b>	programmabile da circa 16 msec fino a 2 sec.
<b>Risoluzione A/D converter:</b>	10 bits
<b>Tempo di conversione A/D:</b>	minimo 6 µsec (con 10 bit di precisione).

**CARATTERISTICHE FISICHE**

<b>Dimensioni (L x A x P):</b>	42 x 25 x 15 mm
<b>Peso:</b>	12 g
<b>Connettori:</b>	zoccolo maschio da 28 piedini, passo 100 mils, largo 600 mils
<b>Range di temperatura:</b>	da 0 a 50 gradi Centigradi
<b>Umidità relativa:</b>	20% fino a 90% (senza condensa)

**CARATTERISTICHE ELETTRICHE**

<b>Tensione di alimentazione:</b>	+Vdc POW = +5 Vdc
<b>Consumo di corrente +Vdc POW:</b>	
minimo:	12 mA
normale:	31 mA
massimo:	40 mA
<b>Tensione batteria di back up:</b>	3,0 V
<b>Consumo di corrente back up:</b>	3,9 $\mu$ A
<b>Impedenza generatori segnali analogici:</b>	<10 K $\Omega$
<b>Protezione RS 232:</b>	$\pm$ 15 kV
<b>Resistenza di pull-up su I<sup>2</sup>C BUS:</b>	4,7 k $\Omega$
<b>Soglie del brown out:</b>	programmabile da 2,5 a 4,1 Vdc, con isteresi

## 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 dei connettori, dei LEDs, dei dip switch, ecc. presenti sulla **CAN AVR**.

### SEGNALAZIONI VISIVE

La scheda **CAN AVR** è dotata delle segnalazioni visive descritte nella seguente tabella:

LED	COLORE	FUNZIONE
LD1	Verde	Viene pilotato dalla linea PA.0 , AD0 del mini modulo e può essere usato come LED di attività, gestito via software.
LD2	Rosso	Viene pilotato dalla linea PA.1 , AD1 del mini modulo e può essere usato come LED di attività, gestito via software.

**FIGURA 6: TABELLA DELLE SEGNALAZIONI VISIVE**

La funzione principale di questo LED 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 12, mentre per ulteriori informazioni sui LEDs si faccia riferimento al paragrafo LED DI ATTIVITA'.

### CONNESSIONI

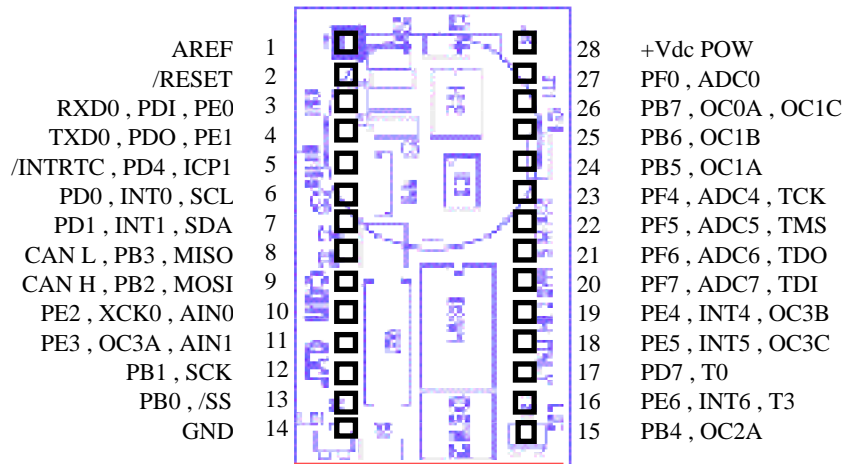
Il modulo **CAN AVR** è provvisto di 1 connettore con cui vengono effettuati tutti i collegamenti con il campo e con le altre schede del sistema di controllo da realizzare. Di seguito viene riportato il suo pin out ed il significato dei segnali collegati; per una facile individuazione di tale connettore, si faccia riferimento alla figura 12, 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 e presentano alcuni dei collegamenti più frequentemente richiesti.

#### **CN1 - CONNETTORE CON SEGNALI DEL MINI MODULO**

Il connettore CN1 é uno zoccolo maschio da 28 piedini con passo 100 mils e larghezza 600 mils. Su questo connettore sono presenti tutti i segnali d'interfacciamento del mini modulo come l'alimentazione, le linee di I/O, le linee di comunicazione seriale sincrona ed asincrona, i segnali delle periferiche hardware di bordo, le linee di selezione del modo operativo, ecc.

Alcuni piedini di questo connettore hanno una duplice o triplice funzione infatti, via software, alcune sezioni interne della CPU possono essere multiplexate con i segnali di I/O e per completezza la seguente figura li riporta tutti. I segnali presenti su CN1 sono quindi di diversa natura, come descritto nel successivo paragrafo INTERFACCIAMENTO CONNETTORI CON IL CAMPO e seguono il pin out standardizzato dei Mini Moduli **grifo**<sup>®</sup>.

Al fine di evitare problemi di conteggio e numerazione la figura 7 descrive i segnali direttamente sulla vista dall'alto della **CAN AVR**, inoltre la serigrafia riporta la numerazione sui 4 angoli della scheda sia sul lato superiore che inferiore



**FIGURA 7: CN1 - ZOCCOLO CON SEGNALI DEL MINI MODULO**

Legenda:

- RXD0** = I - Linea di ricezione seriale USARTn; può essere in RS 232 o TTL
- TXD0** = O - Linea di trasmissione seriale USART0; può essere in RS 232 o TTL
- CAN L** = I/O - Linea differenziale L di ricetrasmisione dell'interfaccia CAN
- CAN H** = I/O - Linea differenziale H di ricetrasmisione dell'interfaccia CAN
- PDI** = I - Linea di uscita dati per la programmazione ISP
- PDO** = O - Linea di ingresso dati per la programmazione ISP
- PAx,PBx,....,PFx** = I/O - Segnale x del Port n di I/O digitale della CPU
- SCL** = O - Linea clock dell'interfaccia I<sup>2</sup>C BUS
- SDA** = I/O - Linea dati dell'interfaccia I<sup>2</sup>C BUS
- Tn** = I - Segnale di conteggio per Timer n della CPU
- INTn** = I - Linea d'interrupt n della CPU
- ADCn** = I - Ingresso analogico n della sezione A/D converter
- AINn** = I - Ingresso n-esimo della sezione di comparatore analogico
- AREF** = I - Ingresso della tensione di riferimento dell'A/D converter
- OCn(A/B/C)** = O - Uscite della n-esima sezione di confronto
- TDO,TDI** = I/O - Segnali dati dell'interfaccia JTAG
- TCK,TMS** = I - Segnali di clock e di controllo dell'interfaccia JTAG
- XCKn** = I - Segnale di clock esterno per USARTn (in funzionamento sincrono)
- /INTRTC** = I - Segnale di interrupt periodico proveniente dall'RTC PCF 8583
- /RESET** = I - Segnale di reset
- MOSI** = I/O - Linea di uscita dati dell'interfaccia SPI
- MISO** = I/O - Linea di ingresso dati dell'interfaccia SPI
- /SS** = I - Linea di selezione unità slave dell'interfaccia SPI
- SCK** = I/O - Linea di clock dell'interfaccia SPI o della programmazione ISP
- +Vdc POW** = I - Linea di alimentazione
- GND** = - Linea di massa



## CONFIGURAZIONE MINI MODULO

A bordo del Mini Modulo **CAN AVR** sono presenti un dip switch ad 8 vie ed un dip switch a 4 vie, con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento dello stesso. Nelle successive figure ne è riportato l'elenco, l'ubicazione e la loro funzione nelle varie modalità di connessione.

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

Per individuare la posizione degli elementi di configurazione si faccia riferimento alla figura 12.

Per ulteriori informazioni sulla configurazione delle linee di comunicazione seriele, si faccia riferimento al paragrafo "SELEZIONE COMUNICAZIONE SERIALE".

SWITCH	POSIZIONE	UTILIZZO	DEF.
DSW2.1	ON	Collega l'uscita CAN L del driver CAN al segnale CANL , PB3 , MISO di CN1. Usato in abbinamento a DSW2.3.	*
	OFF	Non collega l'uscita CAN L del driver CAN al segnale CANL , PB3 , MISO di CN1. Usato in abbinamento a DSW2.3.	
DSW2.2	ON	Collega l'uscita CAN H del driver CAN al segnale CANH , PB2 , MOSI di CN1. Usato in abbinamento a DSW2.4.	*
	OFF	Non collega l'uscita CAN H del driver CAN al segnale CANH , PB2 , MOSI di CN1. Usato in abbinamento a DSW2.4	
DSW2.3	ON	Collega il segnale CANL , PB3 , MISO di CN1 al pin 13 del microcontrollore (segnaleMISO , PB3). Usato in abbinamento a DSW2.1.	*
	OFF	Non collega il segnale CANL , PB3 , MISO di CN1 al pin 13 del microcontrollore (segnaleMISO , PB3). Usato in abbinamento a DSW2.1.	
DSW2.4	ON	Collega il segnale CANH , PB2 , MOSI di CN1 al pin 12 del microcontrollore (segnale MOSI , PB2). Usato in abbinamento a DSW2.2	*
	OFF	Non collega il segnale CANH , PB2 , MOSI di CN1 al pin 12 del microcontrollore (segnale MOSI , PB2). Usato in abbinamento a DSW2.2	

**FIGURA 8: TABELLA DIP SWITCH DSW2 A 4 VIE**



SWITCH	POSIZIONE	UTILIZZO	DEF.
DSW1.1	ON	Collega segnale RXD0 , PDI , PE.0 di ricezione seriale del microcontrollore al driver RS232. Usato in abbinamento a DSW1.3,5.	*
	OFF	Non collega segnale di ricezione seriale del microcontrollore al driver RS232 consentendo il collegamento diretto al segnale RXD0 , PDI , PE.0 su CN1. Usato in abbinamento a DSW1.3,5.	
DSW1.2	ON	Collega segnale TXD0 , PDO , PE.1 su CN1 al driver seriale RS 232. Usato in abbinamento a DSW1.4.	*
	OFF	Non collega segnale TXD0 , PDO , PE.1 su CN1 al driver seriale RS 232 consentendo il collegamento diretto al microcontrollore. Usato in abbinamento a DSW1.4.	
DSW1.3	ON	Collega segnale RXD0 , PDI , PE.0 su CN1 al driver seriale RS 232. Usato in abbinamento a DSW1.1,5.	*
	OFF	Non collega segnale RXD0 , PDI , PE.0 su CN1 al driver seriale RS 232 consentendo il collegamento diretto al microcontrollore. Usato in abbinamento a DSW1.1,5.	
DSW1.4	ON	Collega segnale TXD0 , PDO , PE.1 su CN1 direttamente al microcontrollore, eliminando il driver seriale RS 232. Usato in abbinamento a DSW1.2.	
	OFF	Non collega segnale TXD0 , PDO , PE.1 su CN1 al microcontrollore, consentendo l'uso del driver seriale RS 232. Usato in abbinamento a DSW1.2.	*
DSW1.5	ON	Collega segnale RXD0 , PDI , PE.0 su CN1 direttamente al microcontrollore, eliminando il driver seriale RS 232. Usato in abbinamento a DSW1.1,3.	
	OFF	Non collega segnale RXD0 , PDI , PE.0 su CN1 al microcontrollore, consentendo l'uso del driver seriale RS 232. Usato in abbinamento a DSW1.1,3.	*
DSW1.6	ON	Collega la batteria di bordo all'RTC PCF 8583, consentendo di mantenere data, ora e contenuto della SRAM anche in assenza dell'alimentazione.	
	OFF	Non collega la batteria di bordo all'RTC PCF 8583, quindi in assenza dell'alimentazione si perdono data, ora e contenuto della SRAM.	*
DSW1.7	ON	Collega il segnale PA.2 , AD2 al segnale di GND. Questo switch può essere usato come ingresso utente.	
	OFF	Non collega il segnale PA.2 , AD2 al segnale di GND, che rimane collegato ai +Vdc POW tramite pull-up. Questo switch può essere usato come ingresso utente.	*
DSW1.8	ON	Collega il segnale PA.3 , AD3 al segnale di GND. Questo switch può essere usato come ingresso utente.	
	OFF	Non collega il segnale PA.3 , AD3 al segnale di GND, che rimane collegato ai +Vdc POW tramite pull-up. Questo switch può essere usato come ingresso utente.	*

FIGURA 9: TABELLA DIP SWITCH DSW1 AD 8 VIE

## SELEZIONE COMUNICAZIONE SERIALE

La linea di comunicazione seriale della scheda **CAN AVR** può essere bufferata in RS 232 o TTL. Dal punto di vista software sulla linea può essere definito il protocollo fisico di comunicazione tramite la programmazione di alcuni registri interni del microprocessore.

L'interfaccia seriale (denominate USART0) possiede un gruppo indipendente di registri per la configurazione, e può funzionare in maniera totalmente indipendente rispetto alle altre periferiche. La selezione del protocollo elettrico avviene via hardware e richiede un'opportuna configurazione del dip switch di bordo, come descritto nelle precedenti tabelle; l'utente può autonomamente passare da una configurazione all'altra seguendo le informazioni riportate di seguito:

### - LINEA SERIALE USART0 SETTATA IN RS 232 (configurazione default)

DSW1.1	=	ON
DSW1.2	=	ON
DSW1.3	=	ON
DSW1.4	=	OFF
DSW1.5	=	OFF

### - LINEA SERIALE USART0 SETTATA IN TTL

DSW1.1	=	OFF
DSW1.2	=	OFF
DSW1.3	=	OFF
DSW1.4	=	ON
DSW1.5	=	ON

Le figure 10 ed 11 illustrano come collegare un generico sistema esterno alla linea seriale della **CAN AVR** rispettivamente in configurazione ES 232 e TTL.

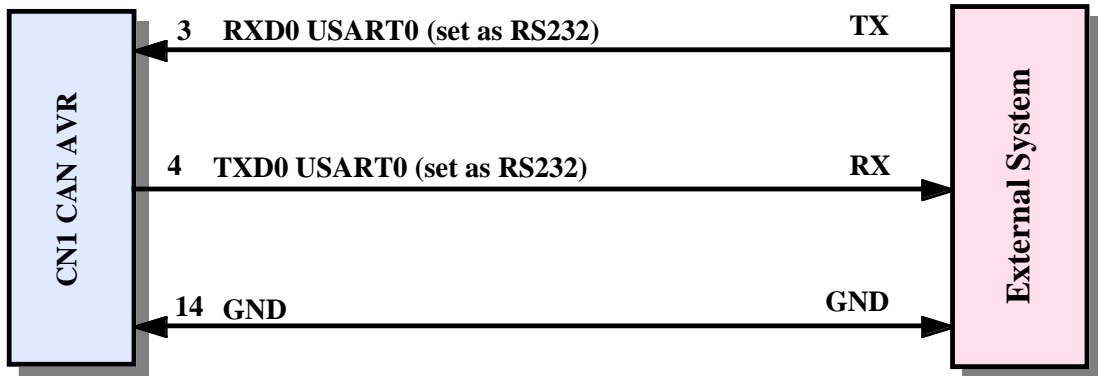


FIGURA 10: ESEMPIO COLLEGAMENTO SERIALE RS 232

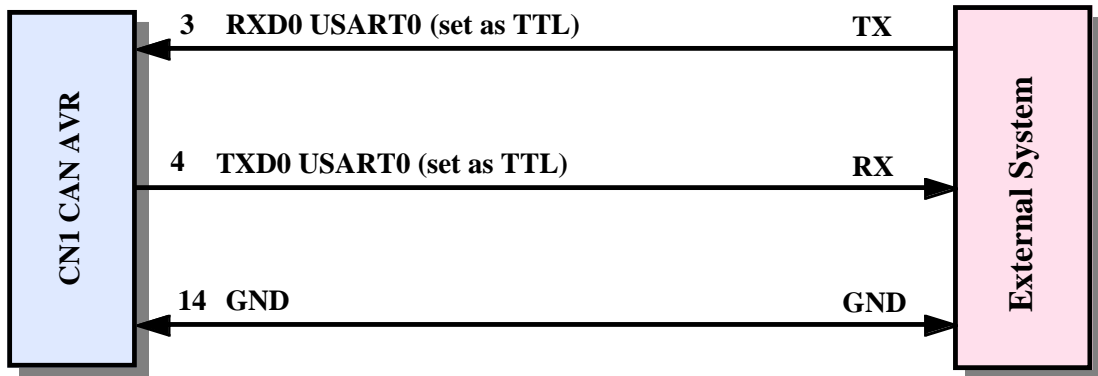


FIGURA 11: ESEMPIO COLLEGAMENTO SERIALE TTL

## INTERFACCIAMENTO CONNETTORI CON IL CAMPO

Al fine di evitare eventuali problemi di collegamento della scheda con tutta l'elettronica del campo a cui **CAN AVR** si deve interfacciare, si devono seguire le informazioni riportate nei vari paragrafi.

- Per i segnali che riguardano la comunicazione seriale con i protocolli RS 232 e CAN, fare riferimento alle specifiche standard degli stessi.
- Tutti i segnali a livello TTL possono essere collegati a linee dello stesso tipo riferite alla massa digitale della scheda. Il livello 0V corrisponde allo stato logico 0, mentre il livello pari alla tensione di alimentazione corrisponde allo stato logico 1. La connessione di tali linee ai dispositivi del campo (fine corsa, encoders, elettrovalvole, relé di potenza, ecc.) deve avvenire tramite apposite interfacce di potenza che preferibilmente devono essere optoisolate in modo da mantenere isolata la logica del Mini Modulo dagli eventuali disturbi dell'elettronica di potenza.
- I segnali d'ingresso dei comparatori analogici devono essere collegati a generatori di segnali analogici a bassa impedenza che rispettino il range di variazione ammesso ovvero da 0 V alla tensione di riferimento selezionata.
- I segnali d'ingresso degli A/D converter devono essere collegati a generatori di segnali analogici a bassa impedenza che rispettino il range di variazione ammesso ovvero da 0 V alla tensione di riferimento selezionata.
- I segnali PWM generati dalle sezioni Timer Counter e OCM, sono a livello TTL e devono essere quindi opportunamente bufferati per essere interfacciati all'azionamento di potenza. Le classiche circuiterie da interporre possono essere dei semplici driver di corrente se é ancora necessario un segnale PWM, oppure un integratore qualora sia necessario un segnale analogico.
- Anche i segnali I<sup>2</sup>C BUS ed SPI sono a livello TTL, come definito dallo stesso standard; per completezza si ricorda solo che dovendo realizzare una rete con numerosi dispositivi e con una discreta lunghezza si deve studiare attentamente il collegamento oppure configurare lo stadio d'uscita, le molteplici modalità operative ed il bit rate programmabili opportunamente in modo da poter comunicare in ogni condizione operativa. Sul Mini Modulo i segnali di SDA ed SCL hanno un pull-up verso +Vdc POW di 4,7 kΩ.

## ALIMENTAZIONE

Il Mini Modulo può essere alimentato mediante una tensione di +5 Vdc, indicata con +Vdc POW. Sulla scheda sono state adottate tutte le scelte circuitali e componentistiche che tendono a ridurre la sensibilità ai disturbi ed i consumi, compresa la possibilità di far lavorare il microcontrollore in quattro diverse modalità a basso assorbimento.

Nella condizione ottimale si arriva ad un consumo minimo (in modalità power down) inferiore ad 1 mA che ad esempio salvaguarda la durata di batterie, nel caso di applicazioni portatili.

Informazioni più dettagliate sono riportate nel capitolo CARATTERISTICHE ELETTRICHE.

## INTERRUPTS

Una caratteristica peculiare della **CAN AVR** è la notevole potenza nella gestione delle interruzioni. Di seguito viene riportata una breve descrizione di quali sono i dispositivi che possono generare interrupts e con quale modalità; per quanto riguarda la gestione di tali interrupts si faccia riferimento ai data sheets del microprocessore oppure all'appendice A di questo manuale.

- Pin 6 di CN1 -> Genera un INT0 sulla CPU.
- Pin 7 di CN1 -> Genera un INT1 sulla CPU.
- Pin 19 di CN1 -> Genera un INT4 sulla CPU.
- Pin 18 di CN1 -> Genera un INT5 sulla CPU.
- Pin 16 di CN1 -> Genera un INT6 sulla CPU.
- Periferiche della CPU -> Generano un interrupt interno. In particolare le possibili sorgenti d'interrupt sono: Timer Counter, OCM, USART0, CAN, comparatore analogico, I<sup>2</sup>C BUS, SPI, EEPROM, A/D converter.

Incorporata nel microcontrollore si trova la logica di gestione degli interrupt che consente di attivare, disattivare, mascherare le 37 sorgenti d'interrupt e che regola l'attivazione contemporanea di più interrupts. In questo modo l'utente ha sempre la possibilità di rispondere in maniera efficace e veloce a qualsiasi evento esterno, stabilendo anche la priorità delle varie sorgenti.

## INTERFACCIA JTAG

Il microcontrollore a bordo del Mini Modulo **CAN AVR** è dotato di interfaccia JTAG.

I segnali di tale interfaccia (TDI, TDO, TMS e TCK) sono rispettivamente disponibili su pin dal 20 al 23 di CN1. Essendo multiplexati con altre funzioni, è necessario programmare opportunamente i registri interni del micro per utilizzarli.

Inoltre l'interfaccia JTAG si può disabilitare programmando un bit di configurazione.

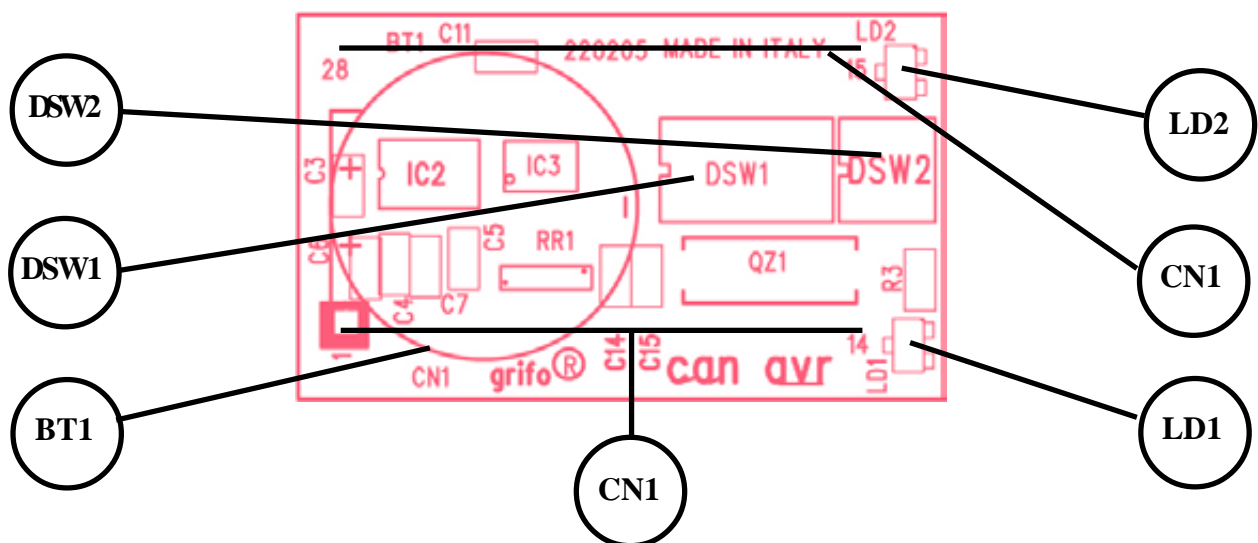


FIGURA 12: DISPOSIZIONE LEDs, DIP SWITCHES, ECC.

## SCHEDE DI SUPPORTO

Il Mini Modulo **CAN AVR** può essere utilizzato come macro componente per alcune schede di supporto sia sviluppate dall'utente che già disponibili nel carteggio **grifo**<sup>®</sup>. Nei successivi paragrafi vengono illustrate le configurazioni delle schede di supporto più interessanti.

### UTILIZZO CON IL MODULO BLOCK GMB HR84

La **GMB HR84** si distingue per essere una scheda che fornisce ai Mini Moduli da 28 pin fino a 8 ingressi optoisolati; 4 uscite a relè, la possibilità di montaggio meccanico su barra ad omega ed il cablaggio tramite comode morsettiere.

La descrizione completa del prodotto è disponibile nel manuale tecnico e nel manuale dell'accoppiata, mentre in questo paragrafo sono riportate le potenzialità offerte.

La **GMB HR84** permette facilmente di:

- alimentare il Mini Modulo tramite l'alimentatore di bordo;
- riportare otto linee dei port di I/O sugli ingressi optoisolati che possono essere indifferentemente connessi come NPN o PNP. Lo stato di tutti gli 8 ingressi viene visualizzato tramite LEDs Verdi. Essendo le linee multiplexate con le periferiche interne è possibile creare rapidamente funzioni evolute come contatori, riconoscimento combinazioni, ecc.;
- riportare quattro linee dei port di I/O su uscite a Relè visualizzate tramite LEDs Rossi.
- avere le linee dell'I<sup>2</sup>C BUS e dell'alimentazione su un connettore distinto;
- collegare la linea di comunicazione seriale tramite un connettore DB9 da 9 vie;
- bufferare, esternamente al Mini Modulo, i segnali della linea seriale 0 TTL in RS 422, RS 485 o Current Loop;
- collegare i segnali PWM su un connettore AMP MOD II da 8 vie;



FIGURA 13: FOTO DI GMB HR84 E CAN AVR

## UTILIZZO CON CAN GMT

Nel carteggio **grifo®** la **CAN GMT** si distingue per essere il modulo prototipale progettato esplicitamente per fare da supporto ai Mini Moduli **CAN xxx** 28 pins.

La **CAN GMT** permette facilmente:

- di alimentare il Mini Modulo tramite l'alimentatore di bordo
- di riportare le linee dei port di I/O e dell'A/D converter su un comodo connettore a scatolino compatibile con lo standard **I/O ABACO®**
- di avere le linee dell'I<sup>2</sup>C Bus e dell'interrupt su un connettore a morsettiera dedicato, per poter espandere il sistema con un qualunque dispositivo I<sup>2</sup>C Bus controllato indifferentemente sia in polling sia in interrupt
- di collegare immediatamente le linee RS 232 - TTL e CAN tramite due comodi connettori a vaschetta
- di impostare e visualizzare lo stato di fino a 6 linee di I/O del microcontrollore tramite pulsanti e LEDs di colori differenti escludibili tramite jumpers
- di generare feedback sonori mediante il buzzer autoscillante a bordo
- di sviluppare rapidamente e confortevolmente qualunque applicazione avvalendosi della vasta area prototipale con ridondanza di ogni segnale

La seguente configurazione consente di usare l'accoppiata **CAN GMT** + **CAN AVR** nella loro versione base, ovvero con linea seriale in RS 232:

### *Configurazione CAN AVR*

DSW1.1	=	ON
DSW1.2	=	ON
DSW1.3	=	ON
DSW1.4	=	OFF
DSW1.5	=	OFF
DSW1.6	=	OFF
DSW1.7	=	OFF
DSW1.8	=	OFF

### *Configurazione CAN GMT*

J1	=	1-2
J2	=	1-2
J3	=	1-2
J4	=	1-2
J5	=	1-2
J6	=	1-2
J7	=	1-2
J8	=	non connesso
J9	=	1-2

Cavo collegamento seriale con P.C. di sviluppo = CCR 9+9 E (ovvero cavo prolunga con vaschetta D9 Femmina e D9 Maschio).



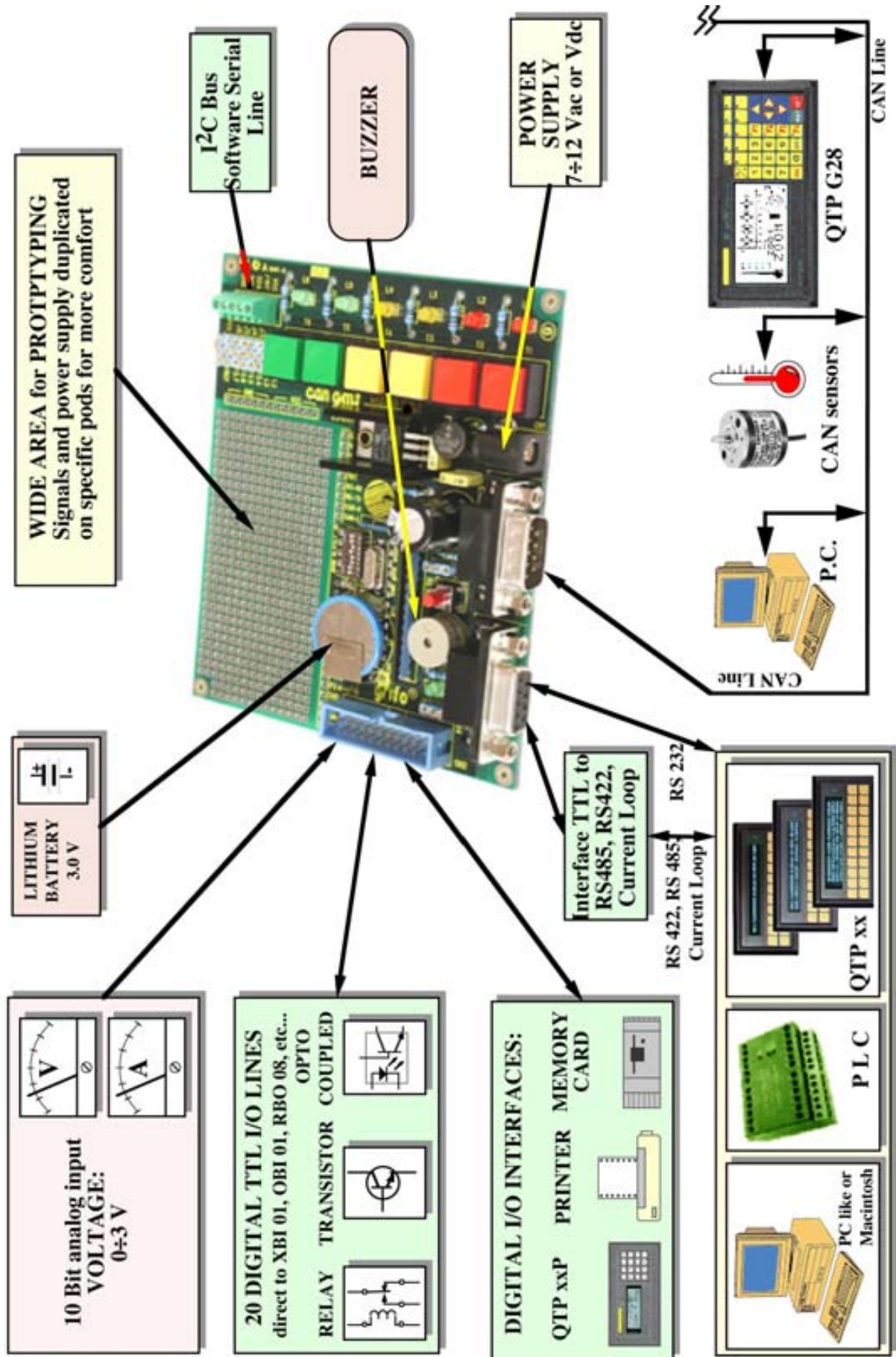


FIGURA 14: ESEMPIO DI COLLEGAMENTO DI CAN GMT' CON CAN AVR

## COME INIZIARE

In questa fase si ipotizza di poter disporre di una **CAN GMT** o di una **GMB HR84** ove montare il Mini Modulo **CAN AVR**.

Per ulteriori informazioni sull'accoppiata **GMB HR84 + CAN AVR**, consultare lo specifico manuale.

### A) COLLEGAMENTO SERIALE TRA CAN AVR ED IL PC

A1) Per prima cosa dovete realizzare fisicamente il collegamento seriale tra il Mini Modulo **CAN AVR** ed il PC. Per questo bisogna costruire un cavo che effettui il collegamento descritto in figura 15.

Il programma già presente nel momento in cui la scheda viene fornita al cliente è il programma di collaudo, e comunica attraverso la USART0. Si ricorda che il programma di collaudo non è progettato per funzionare al di fuori delle strutture di collaudo della **grifo**<sup>®</sup>, pertanto viene fortemente consigliato di **non collegare altri segnali** all'infuori di quelli della USART0 in questa fase.

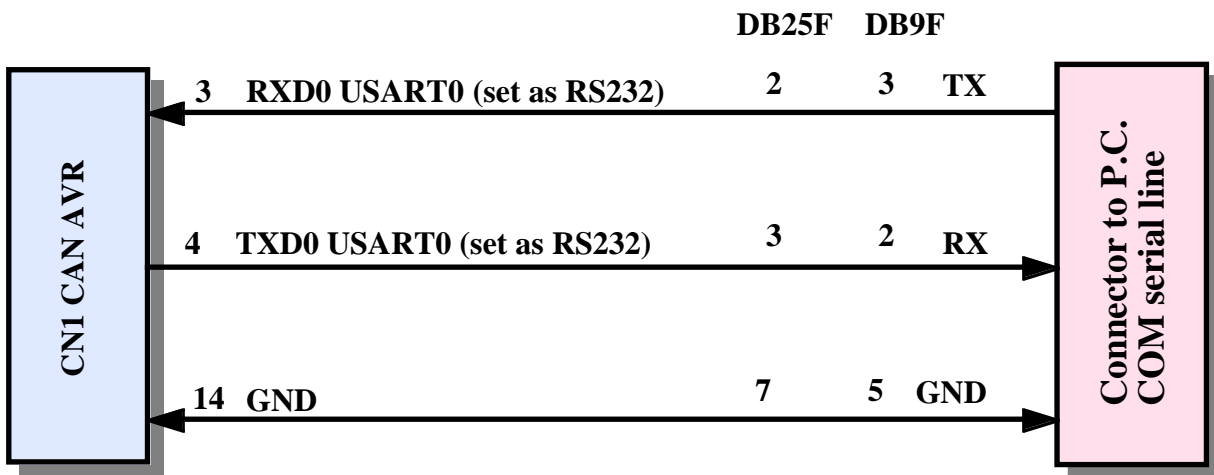


FIGURA 15: COLLEGAMENTO SERIALE RS 232 TRA CAN AVR E PC

A2) Aprire un emulatore di terminale sul PC, configurarlo per usare la porta seriale collegata al **CAN AVR** con 19200 baud, 8 bit di dati, 1 bit di stop, nessuna parità. Se avete il **BASCOM AVR** potete usare l'emulatore terminale incorporato nel compilatore.

A3) Alimentare **CAN GMT** o **GMB HR84**. La schermata del programma di collaudo della **CAN AVR** deve apparire nell'emulatore di terminale. Se ciò non dovesse accadere, ricontrollate il cavo di collegamento e l'alimentazione.



PROGRAMMI PER MINI MODULI E MINI BLOCK													
TIPO DI SCHEDA	GET	ASM	Ladder	Abaco® Link BUS	BASIC CBZ80	BASIC BASCOM 8051	BASIC BASCOM AVR	PIC BASIC	BASIC VARI	MCS® Basic 52	C	PASCAL	TIPO DI CPU / BLOCK
VARI	-	-	-	-	-	●	-	-	-	-	-	-	-
CAN GM0	●	-	-	-	-	●	-	-	-	-	●	-	Atmel T89C51cc03 - 8051 Code
CAN GM1	●	-	-	-	-	●	-	-	-	-	●	-	Atmel T89C51cc01 - 8051 Code
CAN GM2	●	-	-	-	-	●	-	-	-	-	●	-	Atmel T89C51cc02 - 8051 Code
GMM 5115	●	-	-	-	-	●	-	-	-	-	●	-	Atmel T89C5115 - 8051 Code
GMM 876	●	-	-	-	-	-	-	●	-	-	-	-	Microchip PIC16F876A - PIC 14 Code
GMM 932	●	-	-	-	-	●	-	-	-	-	-	-	PHILIPS P89LPC932 - 8051 Code
GMM AC2	●	-	-	-	-	●	-	-	-	-	●	-	Atmel T89C51AC2 - 8051 Code
GMM AM08	●	-	-	-	-	-	●	-	-	-	-	-	Atmel ATmega08 - AVR Code
GMM AM32	●	-	-	-	-	-	●	-	-	-	-	-	Atmel ATmega32 - AVR Code
GMB HR84	●	-	-	-	-	●	●	●	-	-	●	-	Mini Block 8 input opto 4 output relè
GMB HR168	●	-	-	-	-	●	-	-	-	-	●	-	Mini Block 16 input opto 8 output relè




FIGURA 16: TABELLA ESEMPI

## B) RIPROGRAMMAZIONE DELLA FLASH

La programmazione del Mini Modulo avviene utilizzando un Boot Loader preprogrammato nella Flash del Mini Modulo stesso.

Tale programma permette di creare una comunicazione tra il Mini Modulo ed il PC, e di usarla per inviare un file in formato .hex nella memoria codice e/o nella EEPROM del microcontrollore.

Il Boot Loader non permette di programmare i bit di configurazione ed i fusibili di sicurezza del microcontrollore, cosa che comunque è possibile fare con la programmazione ISP (si veda l'appendice B per ulteriori informazioni).

L'occupazione del Boot Loader è pari a 4KBytes, ovvero viene riservato il blocco alla fine della memoria Flash pari ad 2KWord, che quindi non è più disponibile per il programma utente.

Il Boot Loader comunica tramite la porta seriale del PC installando una apposita utility (scaricabile gratuitamente dal nostro sito [www.grifo.it](http://www.grifo.it) o dal nostro CD) chiamata AVRBootloaderGrifo, che permette di selezionare la porta seriale da usare e di selezionare i files con cui programmare le memorie del Mini Modulo.

Vi sono metodi di programmazione che richiedono l'uso di un programmatore ISP. Questi sono descritti nell'appendice B.

La casella combinata "Com Port" permette di selezionare la porta seriale da usare per la comunicazione con in Mini Modulo.

Mettendo lo spunto alla casella "Application Code" il file il cui nome è scritto nella casella di testo accanto verrà memorizzato nella Flash del Mini Modulo. Per scegliere il nome del file premere il pulsante "Browse" adiacente.

Analogamente, la casella "EEPROM Code" controlla la programmazione del contenuto della EEPROM.

Esiste inoltre la possibilità di utilizzare direttamente il AVRBootloaderGrifo da finestra DOS, Prompt dei comandi, linea di comando o come tool esterno in un IDE.

Le opzioni accettate dalla linea di comando sono:

/com1 , /com2 , ... , /com9    Numero della porta seriale usata per la comunicazione.  
Sono supportate le porte dalla COM1 alla COM9

/f <ProgrammaFLASH>    Specifica il nome completo di path del file .HEX da scrivere nella memoria FLASH del microcontrollore

/e <ProgrammaEEPROM>    Specifica il nome completo di path del file .HEX da scrivere nella memoria EEPROM del microcontrollore

Ad esempio:

```
AVRBootloaderGrifo.exe /com2 /f C:\Progetti\ControlloMotori\Main.hex
```

apre una connessione sulla porta seriale COM2 verso il Boot Loader ed invia il file Main.hex che si trova nella cartella C:\Progetti\ControlloMotori\.

Per integrare AVRBootloaderGrifo.exe in un IDE, ad esempio quello del BASCOM AVR (descritto nel punto C), si deve.

- 1) Aprire la finestra del menu Options | Programmer
  - 2) Scegliere nella casella di riepilogo Programmer "External Programmer"
  - 3) Scegliere la linguetta Other
  - 4) Nella casella Program inserire il path di AVRBootloaderGrifo.exe premendo il pulsante Browse
  - 5) Nella casella di testo Parameters inserire i parametri per la linea di comando di AVRBootloaderGrifo. Tenete presente che inserendo la stringa {file} nella suddetta casella di testo, il BASCOM AVR la sostituisce automaticamente con il nome del file .hex appena generato.
- Si veda la documentazione del BASCOM AVR per ulteriori informazioni.



- B1) Localizzare e salvare in una posizione comoda sul disco rigido del PC il file si chiama "prCANAVR.HEX" seguendo il percorso: Italiano | Programmi di Esempio | Programmi Mini Moduli e Mini Block | CAN AVR (vedere la figura 16). Eventualmente, rimuovere l'attributo di sola lettura (read-only) del file salvato.
- B2) Collegare il Mini Modulo alla porta seriale del PC realizzando il collegamento elettrico descritto nel punto A. Chiudere l'emulatore terminale.
- B3) Indicare nella casella combinata "Com Port" la porta seriale collegata al punto precedente.
- B4) Mettere lo spunto alla casella "Application Code" e premere il pulsante "Browse" adiacente, quindi selezionare il file precedentemente salvato al punto B1.
- B5) Togliere lo spunto alla casella "EEPROM code", se presente.
- B6) Premere il pulsante "Synch to Bootloader..." o la combinazione di tasti Alt+S sul PC, quindi resettare il Mini Modulo o togliere e rimettere l'alimentazione. Il file viene scaricato nel Mini Modulo. Se questo non dovesse accadere, e il programma indicasse il messaggio "No response from target bootloader", provare a ripetere l'operazione diminuendo il tempo che intercorre tra la pressione del tasto su PC ed il reset del Mini Modulo. Se il problema dovesse persistere, ricontrollare il cavo ed i collegamenti.
- B7) Alla fine dell'operazione il programma ne riporta l'esito. In caso di problemi ricontrollare il cavo ed i collegamenti.
- B8) Avviare l'emulatore terminale configurato come in A2 e verificare che il programma applicativo appena memorizzato venga eseguito dalla FLASH interna.



FIGURA 17: BOOT LOADER UTILITY

## C) CREAZIONE DEL CODICE ESEGUIBILE DEL PROGRAMMA DEMO

- C1) Installare sul disco rigido del PC l'ambiente di sviluppo scelto per realizzare programma applicativo. Sono disponibili diversi ambienti in modo da soddisfare le richieste di ogni utente, ma qui si ricordano quelli più diffusi come il BASCOM AVR, ICC AVR, ecc., si faccia riferimento al manuale del software per ulteriori informazioni.
- C2) Sul CD **grifo**® oltre al file con il codice eseguibile del demo, descritto al punto B2, sono presenti anche il/i file sorgenti dello stesso. Questi hanno un'estensione che identifica l'ambiente di sviluppo usato (prCANAVR.bas per il BASCOM AVR, prCANAVR.c per ICC AVR) e sono opportunamente organizzati nelle tabelle degli esempi presenti sul CD, insieme agli eventuali file di definizione o di progetto (ad esempio: prCANAVR.prj per ICC AVR). Una volta localizzati questi file devono essere salvati in una posizione comoda sul disco rigido del PC di sviluppo.
- C3) Ricompilare il sorgente usando l'ambiente di sviluppo scelto, in modo da ottenere il file prCANAVR.hex identico a quello presente sul CD **grifo**® e già usato nei punti a partire da B2. Questa operazione si differenzia notevolmente a seconda dell'ambiente di sviluppo utilizzato, pertanto qui di seguito vengono esposti i passi dettagliati:

### C3 Bascom AVR) Ricompilazione con BASCOM AVR.

C3 Bascom AVR a) Una volta entrati nell'IDE del BASCOM, caricare il programma sorgente con il menu File | Open:

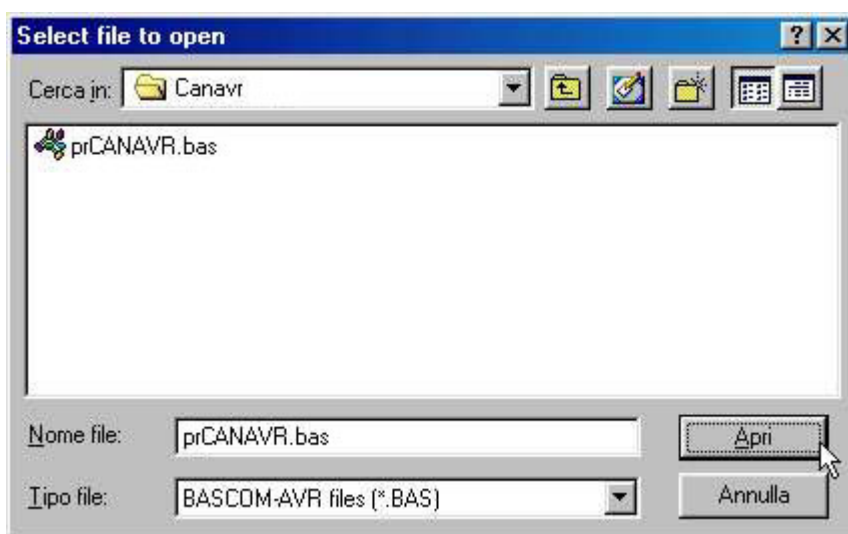
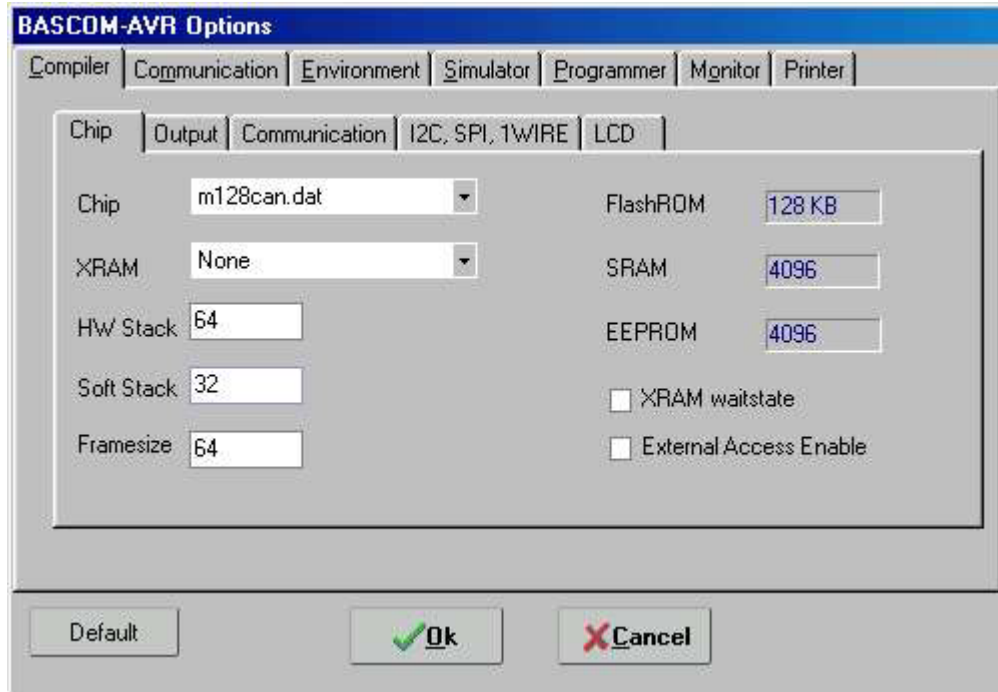


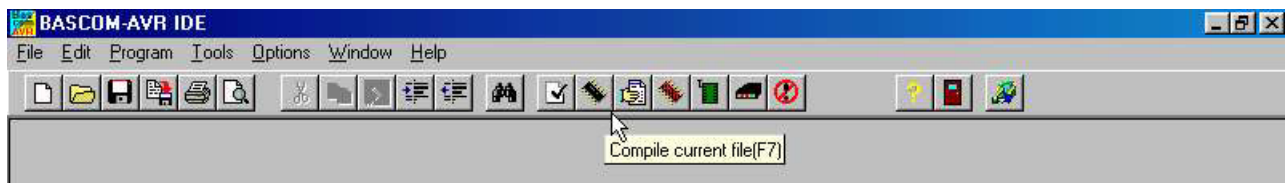
FIGURA 18: CARICAMENTO SORGENTE CON BASCOM AVR

C3 Bascom AVR b) Dal menu Options | Compiler | Chip impostare i valori 64 per HW Stack, 32 per Soft Stack, 64 per Framesize e premere OK. Tali valori sono da considerarsi dei minimi e, se necessario, vanno aumentati:



**FIGURA 19: CONFIGURAZIONE COMPILATORE BASCOM AVR**

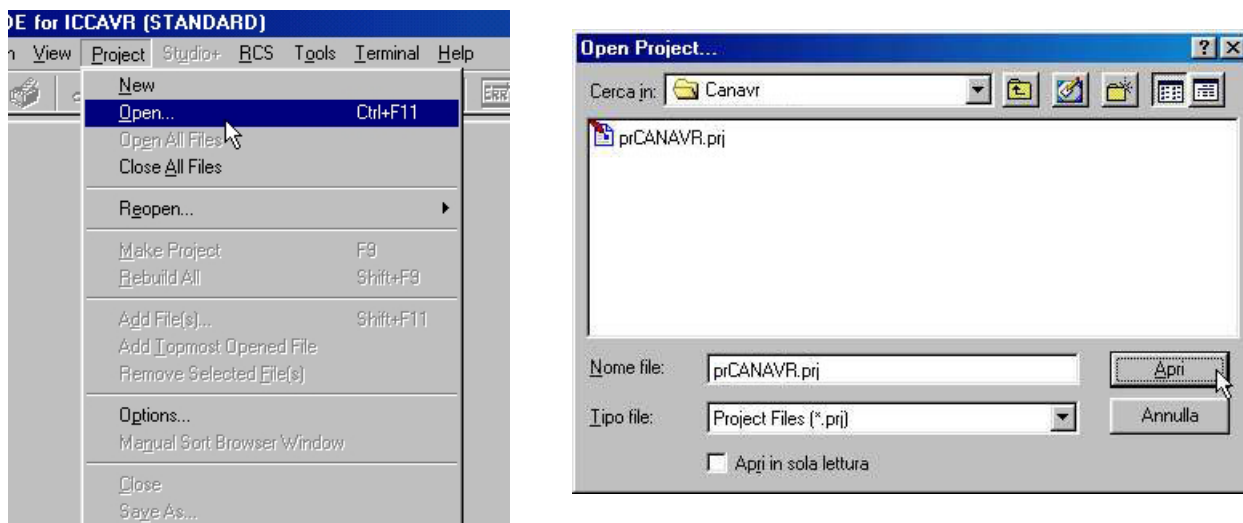
C3 Bascom AVR c) Compilare il sorgente premendo il pulsante con il disegno del circuito integrato:



**FIGURA 20: COMPILAZIONE CON BASCOM AVR**

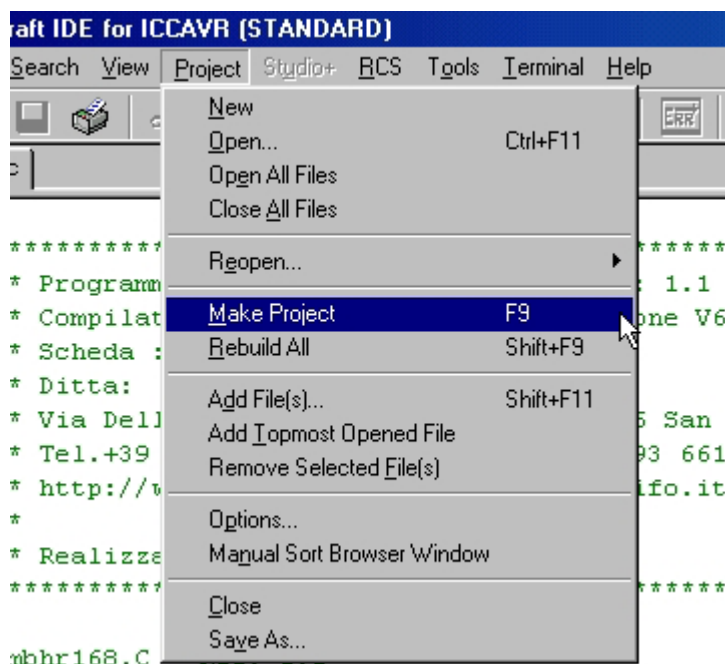
## C3 ICC AVR) Ricompilazione con ICC AVR.

C3 ICC AVR a) Una volta aperto l'editor standard, caricare il file di progetto usando il menu Project | Open...:



**FIGURA 21: CARICAMENTO FILE DI PROGETTO CON ICC AVR**

C3 ICC AVR b) Compilare usando il menu Project | Make Project:



**FIGURA 22: COMPILAZIONE CON ICC AVR**



- C4) Effettuare il salvataggio del file ottenuto nella FLASH del Mini Modulo, ripetendo i punti della sezione B.

## D) *PREPARAZIONE DEFINITIVA DELL'APPLICATIVO*

- D1) Chiudere il programma di comunicazione con il Boot Loader.

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

In caso di esecuzione corretta di tutte le fasi sopra descritte l'utente ha realizzato e salvato il suo primo programma applicativo coincidente con il demo del Mini Modulo **CAN AVR**.

A questo punto è possibile modificare il sorgente del/dei programmi demo in modo da soddisfare le richieste dell'applicazione da realizzare e provarla con i passi sopra elencati (parti B e C) in modo ciclico, fino a quando il programma applicativo realizzato è perfettamente funzionante.

Raggiunto questo obiettivo, si può eliminare il PC di sviluppo.

Ricordarsi di riconfigurare la seriale 0 del Mini Modulo **CAN AVR** in RS 232, se richiesto.



## 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 il Mini Modulo può sfruttare tutte le risorse software per il microprocessore montato e tutti i pacchetti ideati per la famiglia AVR, sia ad alto che a basso livello. Tutti i pacchetti di sviluppo software forniti dalla **grifo**® sono sempre accompagnati da esempi che illustrano come gestire le sezioni della scheda e da una completa documentazione.

### **BASCOM AVR**

Cross compilatore a basso costo per files sorgenti scritti in BASIC, disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione un editor, il compilatore ed un simulatore molto potente per il debugger del sorgente. Comprende molti modelli di memoria, svariati tipi di dati, istruzioni dedicate alle risorse hardware e librerie aggiuntive per simulare od interfacciare dispositivi intelligenti esterni (tastiere PC, display alfanumerici e grafici, lettori di badge, ecc.). L'IDE é in grado di eseguire anche programmi esterni per la programmazione ISP.

Viene fornito su CD con la relativa manualistica tecnica e gli esempi d'uso.

### **ICC AVR**

Cross compilatore per files sorgenti scritti in ANSI C, disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione un editor, il compilatore un ottimizzatore ed un linker. Comprende molti modelli di memoria, svariati tipi di dati, le classiche librerie ANSI con i relativi sorgenti, gestione completa del floating point, un application builder, ecc. Il compilatore si integra con il pacchetto AVR Studio con cui si può effettuare il debugger a livello sorgente dell'applicativo utente in C e con programmi esterni per la programmazione ISP.

Viene fornito su CD con la relativa manualistica tecnica e gli esempi d'uso.

### **DDS MICRO C AVR**

E' un comodo pacchetto software, a basso costo, che tramite un completo IDE permette di utilizzare un editor, un compilatore "C" (integer), un assembler, un linker ed un ottimizzatore. Sono inclusi i sorgenti delle librerie, una serie di utility ed una ricca documentazione su dischetto da 3"1/2 nel formato MS-DOS.

### **AVR Studio**

E' un ambiente di sviluppo per la famiglia AVR di microprocessori che consente di controllare l'esecuzione dei programmi applicativi tramite appositi in circuit emulator esterni oppure in simulazione su PC. AVR Studio consente il debug a livello sorgente assembly e/o C di codici oggetto generati da pacchetti esterni (assemblatori, compilatori, ecc.) e si presenta con una serie di finestre che riportano le principali informazioni necessarie come i registri, la memoria, le periferiche, le variabili in modo da fornire all'utente il pieno controllo di ogni elemento. Include inoltre un "costruttore di applicativo" che facilita la generazione del codice che inizializza le varie periferiche hardware (UART, SPI, Port, ADC, ecc) partendo da un'interfaccia grafica.

Notevole inoltre è la possibilità di pilotare l'interfaccia JTAG "JTAG ICE" prodotta dalla ATMEL. Una interfaccia JTAG permette di entrare nel cuore del microcontrollore per esaminarne lo stato durante l'esecuzione stessa direttamente nella struttura dell'applicativo.

L'utente ha la possibilità di inserire dei breakpoint sia hardware che software, e quando l'esecuzione è ferma può esaminare lo stato dei registri interni e della memoria.

L'interfaccia JTAG permette anche di programmare le memorie del microcontrollore.

Usando l'interfaccia JTAG la difficoltà di molti problemi di debugging viene drasticamente ridotta, aumentando così la facilità di eliminazione, e diminuendo di conseguenza il tempo necessario per giungere all'applicativo finale totalmente debuggato.



FIGURA 23: VISTA DALL'ALTO E DAL BASSO DELLA CAN AVR

## DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO

Di seguito viene riportata una descrizione dettagliata della gestione software delle periferiche di bordo. Qualora la documentazione riportata fosse insufficiente fare riferimento direttamente alla documentazione tecnica della casa costruttrice del componente. In questo paragrafo inoltre non vengono descritte le sezioni che fanno parte del microprocessore; per quanto riguarda la programmazione di quest'ultime si faccia riferimento all'appendice A di questo manuale. Nei paragrafi successivi si usano le indicazioni **0÷7** e **.0÷7** per fare riferimento ai bits della combinazione utilizzata nelle operazioni di I/O.

### LED DI ATTIVITA'

La **CAN AVR** consente la gestione software di due LEDs di attività o stato, LD1 e LD2, tramite due linee di I/O del microprocessore con la seguente corrispondenza:

<b>PA0 = 0</b>	->	<b>LD1 attivo</b>
<b>PA0 = 1</b>	->	<b>LD1 disattivo</b>
<b>PA1 = 0</b>	->	<b>LD2 attivo</b>
<b>PA1 = 1</b>	->	<b>LD2 disattivo</b>

Si ricorda che entrambi i segnali PA0 e PA1 non sono disponibili su CN1.

I segnali PA0 e PA1 sono mantenuti alti in fase di reset o power on, di conseguenza in seguito ad una di queste fasi i LEDs sono disattivi.

### DIP SWITCH

La **CAN AVR** consente la lettura via software dello stato di due dip switches, DSW1.7 e 8, tramite due linee di I/O del microprocessore con la seguente corrispondenza:

<b>DSW1.7 chiuso</b>	->	<b>PA2 è 0</b>
<b>DSW1.7 aperto</b>	->	<b>PA2 è 1</b>
<b>DSW1.8 chiuso</b>	->	<b>PA3 è 0</b>
<b>DSW1.8 aperto</b>	->	<b>PA3 è 1</b>

Si ricorda che entrambi i segnali PA2 e PA3 non sono disponibili su CN1.

## SRAM TAMPONATA + RTC SERIALE

Il Mini Modulo **CAN AVR** dispone 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 alimentato dalla circuiteria di back up in modo da garantire la validità dei dati in ogni condizione operativa ed é completamente gestito via software. La sezione di RTC può inoltre generare interrupt in corrispondenza di intervalli di tempo programmabili sempre via software, in modo da poter periodicamente distogliere la CPU dalle normali operazioni oppure periodicamente risvegliarla dagli stati di basso consumo.

Per quanto riguarda la gestione specifica del modulo di SRAM+RTC seriale, 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. In dettaglio si deve realizzare una comunicazione sincrona con il protocollo standard I<sup>2</sup>C BUS, tramite alcune linee di I/O della CPU:

PD1 , INT1 , SCL (input/output) -> linea DATA (SDA)  
PD0 , INT0 , SCL (input/output) -> linea CLOCK (SCL)

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

Pertanto il suddetto modulo occupa in permanenza lo slave address esadecimale **A0**, che non è più disponibile per l'utente.

Infine, il suddetto modulo è in grado di generare un segnale periodico programmabile, collegato ad un pull-up di 4,7 kΩ, e disponibile sul pin 11 del connettore CN1 del Mini Modulo, nonché collegato al segnale PD4 del microcontrollore.

## PERIFERICHE DELLA CPU

La descrizione dei registri e del relativo significato di tutte le periferiche interne del microprocessore (COMPARATORI, A/D CONVERTERS, TMR CNT, ICU, USART0, CAN, I<sup>2</sup>C BUS, SPI, CCU, ecc) é disponibile nell'apposito manuale tecnico e manuale d'uso della casa costruttrice.

Per ulteriori informazioni si vedano la BIBLIOGRAFIA e l'appendice A di questo manuale.

## 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 del Mini Modulo **CAN AVR**.

Documentazione tecnica MAXIM: *True RS 232 Transceivers*

Manuale PHILIPS: *I2C-bus compatible ICs*

Documentazione tecnica ATMEL: *Data Sheet Atmel AT90CAN128*

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



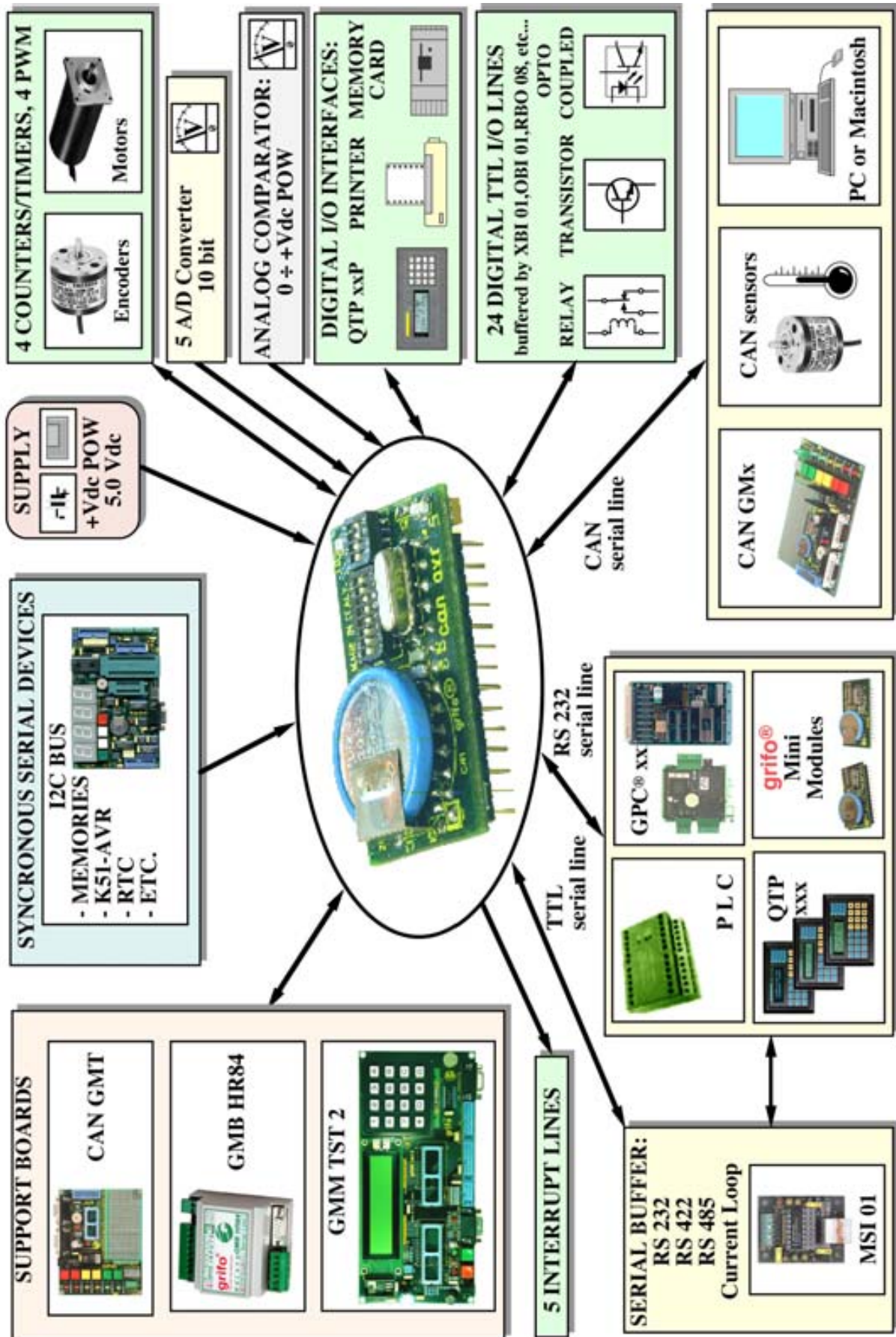


FIGURA 24: SCHEMA DELLE POSSIBILI CONNESSIONI





## APPENDICE A: DESCRIZIONE COMPONENTI DI BORDO

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

### AT90CAN128

Link: Home | Servizio Documentazione Tecnica | ATMEL | Data-Sheet AT90CAN128

URL: <http://www.grifo.it/PRESS/DOC/Atmel/AT90CAN128.pdf>

#### Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Non volatile Program and Data Memories
  - 128K Bytes of In-System Reprogrammable Flash
    - Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    - Selectable Boot Size: 1K Bytes, 2K Bytes, 4K Bytes or 8K Bytes
    - In-System Programming by On-Chip Boot Program (CAN, UART)
    - True Read-While-Write Operation
  - 4K Bytes EEPROM (Endurance: 100,000 Write/Erase Cycles)
  - 4K Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Programming Flash (Hardware ISP), EEPROM, Lock & Fuse Bits
  - Extensive On-chip Debug Support
- CAN Controller 2.0A & 2.0B
  - 15 Full Message Objects with Separate Identifier Tags and Masks
  - Transmit, Receive, Automatic Reply and Frame Buffer Receive Modes
  - 1Mbits/s Maximum Transfer Rate at 8 MHz
  - Time stamping, TTC & Listening Mode (Spying or Autobaud)
- Peripheral Features
  - Programmable Watchdog Timer with On-chip Oscillator
  - 8-bit Synchronous Timer/Counter-0
    - 10-bit Prescaler
    - External Event Counter
    - Output Compare or 8-bit PWM Output
  - 8-bit Asynchronous Timer/Counter-2
    - 10-bit Prescaler
    - External Event Counter
    - Output Compare or 8-Bit PWM Output
    - 32KHz Oscillator for RTC Operation
  - Dual 16-bit Synchronous Timer/Counters-1 & 3
    - 10-bit Prescaler
    - Input Capture with Noise Canceler
    - External Event Counter
    - 3-Output Compare or 16-Bit PWM Output
    - Output Compare Modulation
  - 8-channel, 10-bit SAR ADC
    - 8 Single-ended channels
    - 7 Differential Channels
    - 2 Differential Channels With Programmable Gain at 1x, 10x, or 200x
  - On-chip Analog Comparator
  - Byte-oriented Two-wire Serial Interface
  - Dual Programmable Serial USART
  - Master/Slave SPI Serial Interface
    - Programming Flash (Hardware ISP)
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - 8 External Interrupt Sources
  - 5 Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down & Standby
  - Software Selectable Clock Frequency
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines
  - 64-lead TQFP and 64-lead QFN
- Operating Voltages
  - 2.7 - 5.5V
- Operating temperature
  - Industrial (-40°C to +85°C)
- Maximum Frequency
  - 8 MHz at 2.7V - Industrial range
  - 16 MHz at 4.5V - Industrial range



8-bit **AVR®**  
Microcontroller  
with  
128K Bytes of  
ISP Flash  
and  
CAN Controller

**AT90CAN128**

Rev. 4250F-CAN-04/05



1



**PCF 8583**

Link: Home | Servizio Documentazione Tecnica | I<sup>2</sup>C BUS | Data-Sheet PCF8583  
 URL: <http://www.grifo.it/PRESS/DOC/PHILIPS/PCF8583.pdf>

Philips Semiconductors

Philips Semiconductors

Product specification

Product specification

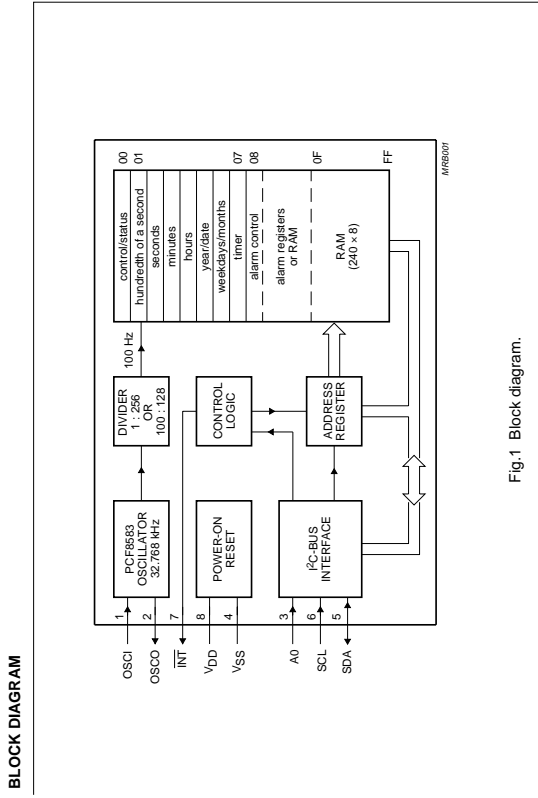
**Block/calendar with 240 × 8-bit RAM**
**Block/calendar with 240 × 8-bit RAM**


Fig.1 Block diagram.

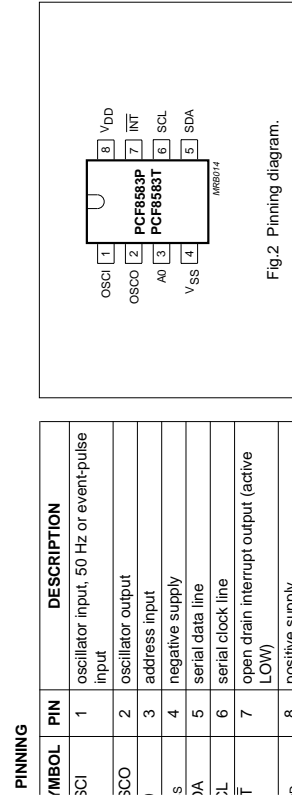


Fig.2 Pinning diagram.

**FEATURES**

- I<sup>2</sup>C-bus interface operating supply voltage: 2.5 V to 6 V
- clock operating supply voltage (0 to +70 °C): 0 V to 6.0 V
- 40 × 8-bit low-voltage RAM
- data retention voltage: 1.0 V to 6 V
- operating current (at f<sub>SCL</sub> = 0 Hz): max. 50 μA
- clock function with four-year calendar
- universal timer with alarm and overflow indication 4 or 12 hour format
- 2.768 kHz or 50 Hz time base
- serial input/output bus (I<sup>2</sup>C)
- automatic word address incrementing
- programmable alarm, timer and interrupt function slave address:

- READ: A1 or A3
- WRITE: A0 or A2.

**QUICK REFERENCE DATA**

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
V <sub>DD</sub>	supply voltage operating mode	I <sup>2</sup> C-bus active	2.5	—	6.0	V
V <sub>DD</sub>	supply voltage inactive mode	I <sup>2</sup> C-bus inactive	1.0	—	6.0	V
I <sub>DD</sub>	supply current operating mode	f <sub>SCL</sub> = 100 kHz	—	—	200	μA
I <sub>DD</sub>	supply current clock mode	f <sub>SCL</sub> = 0 Hz; V <sub>DD</sub> = 5 V	—	10	50	μA
I <sub>DD</sub>	operating ambient temperature range	f <sub>SCL</sub> = 0 Hz; V <sub>DD</sub> = 1 V	—	2	10	μA
T <sub>STG</sub>	storage temperature range		-40	—	+85	°C
T <sub>OP</sub>	operating temperature range		-65	—	+150	°C

**ORDERING INFORMATION**

TYPE NUMBER	PACKAGE NAME	DESCRIPTION	VERSION
PCF8583P	DIP8	plastic dual in-line package; 8 leads (300 mil)	SOT197-1
PCF8583T	SO8	plastic small outline package; 8 leads; body width 7.5 mm	SOT1176-1

## APPENDICE B: PROGRAMMAZIONE ISP CON GMM TST 2

In questa appendice vengono date le istruzioni specifiche per poter programmare il Mini Modulo **CAN AVR** a bordo di una **GMM TST 2** usando un programmatore ISP esterno.

Si tratta di un metodo di programmazione della FLASH e della EEPROM del Mini Modulo alternativo a quello presentato nella sezione "B" del capitolo "Come Iniziare" del manuale.

Le altre voci del suddetto capitolo rimangono comunque valide.

La programmazione ISP permette di modificare anche i bit di configurazione ed i fusibili di sicurezza, oltre a premettere di programmare completamente sia la FLASH che la EEPROM.

L'utilizzo del Boot Loader non permette di programmare completamente la FLASH, in quanto la sezione in cui risiede il Boot Loader stesso è protetta.

Inoltre i bit di configurazione, che tra l'altro abilitano il Boot Loader stesso, ed i fusibili di sicurezza sono al di fuori della portata del Boot Loader e devono necessariamente essere programmati in modalità ISP.

Se si ritiene più conveniente o comunque sufficiente usare un Boot Loader su linea seriale RS 232, si faccia riferimento alla suddetta sezione.

Per informazioni su come collegare una porta seriale del PC alla **GMM TST 2** (ed anche al Mini Modulo da solo) per provare il programma, si faccia riferimento alla sezione "A" del capitolo "Come Iniziare" del manuale.

Per informazioni su come sviluppare e debuggare il firmware, si faccia riferimento alla sezione "C" del capitolo "Come Iniziare" del manuale.

I programmatori utilizzabili sono:

- **grifo® MP-AVR 51+** ; **grifo® UEP 49**
- Equinox EPSILON5
- ATMEL AVR ISP

### A) OPERAZIONI COMUNI PER TUTTI I PROGRAMMATORI

A1) La programmazione del Mini Modulo a bordo della **GMM TST 2** avviene sfruttando un apposito connettore della scheda ed usando eventualmente un adattatore di interfacciamento tra il programmatore ed il connettore stesso.

Per i due programmatori **grifo®** che si possono interfacciare con **GMM TST 2**, ovvero **MP-AVR 51+** e **UEP 49**, lo schema di collegamento delle interfacce viene mostrato rispettivamente nelle figure B-1 e B-2. Sul lato sinistro delle figure si trova l'elenco delle vie del connettore CN7 di **GMM TST 2** che vanno collegati ai pin del connettore del programmatore indicati con la freccia.

Per i programmatori AVR ISP ed Equinox EPSILON5 non c'è bisogno di alcuna interfaccia specifica.

A2) L'interfacciamento del Mini Modulo **CAN AVR** e uno degli zoccoli a 40 vie della **GMM TST 2** avviene tramite un apposito zoccolo di programmazione ISP.

Per effettuare la programmazione ISP della **GMM AM128** installata sullo zoccolo Z1 o Z2 di una **GMM TST 2**, bisogna porre lo zoccolo di programmazione ISP tra il Mini Modulo e la **GMM TST 2** stessa.

Lo zoccolo deve collegare i pin 14 e 15 della **GMM TST 2** rispettivamente ai pin 3 e 4 del Mini Modulo, ed isolare i pin 14 e 15 della **GMM TST 2** dal Mini Modulo, come indicato nella figura B-4. Si tenga presente che la numerazione di figura B-4 si riferisce alla zoccolo adattatore di 28 pin, mentre la **GMM TST 2** ha solo zoccoli a 40 pin.

A3) La configurazione dei jumper della **GMM TST 2** e dei Dip Switch del Mini Modulo è quella riportata a pagina 24 del manuale e che viene qui ripetuta per comodità:

*Configurazione CAN AVR*

DSW1.1 = OFF  
 DSW1.2 = OFF  
 DSW1.3 = OFF  
 DSW1.4 = ON  
 DSW1.5 = ON  
 DSW1.6 = OFF  
 DSW1.7 = OFF  
 DSW1.8 = OFF

*Configurazione GMM TST 2*

J1 = 2-3  
 J2 = 2-3  
 J3 = non connesso  
 J4 = non connesso  
 J5 = non connesso  
 J6 = non connesso  
 J7 = non connesso

DSW2.1 = OFF  
 DSW2.2 = OFF  
 DSW2.3 = OFF  
 DSW2.4 = OFF

A4) Localizzare e salvare in una posizione comoda sul disco rigido del PC il file si chiama "prCANAVR.HEX" seguendo il percorso: Italiano | Programmi di Esempio | Programmi Mini Moduli e Mini Block | CAN AVR (fare riferimento alla figura 16 del manuale).

Nei punti successivi viene descritta la configurazione dettagliata per ogni singolo programmatore.

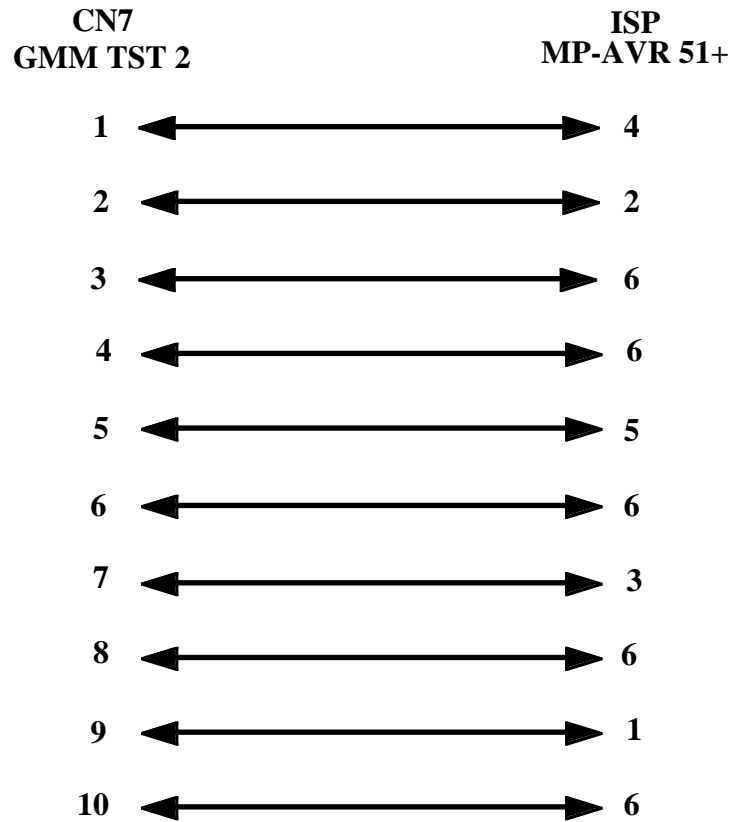


FIGURA B-1: SCHEMA DELL'INTERFACCIA TRA GMM TST 2 E MP-AVR 51+

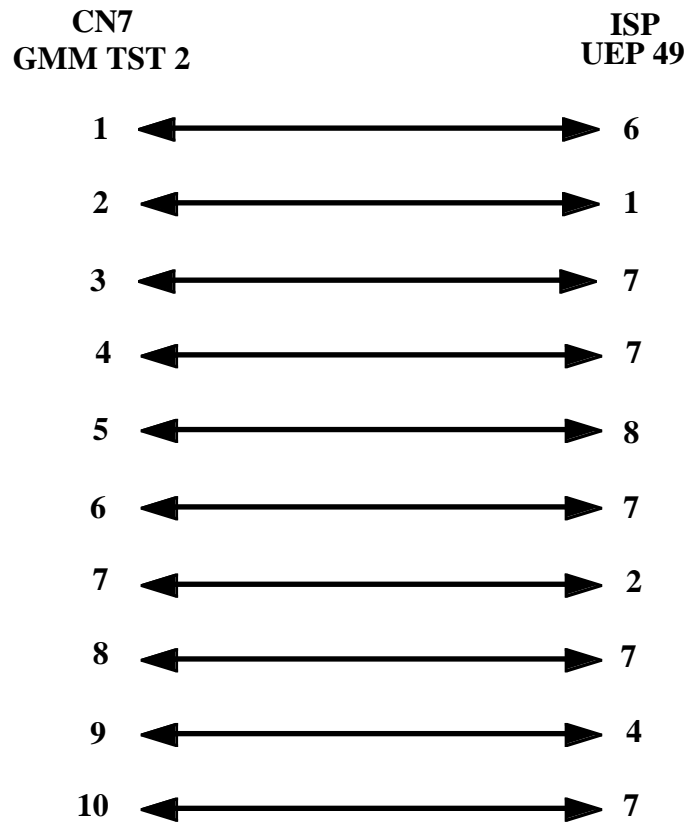


FIGURA B-2: SCHEMA DELL'INTERFACCIA TRA GMM TST 2 E UEP 49

**B) PROGRAMMAZIONE CON GRIFO® MP-AVR 51+ ; GRIFO® UEP 49**

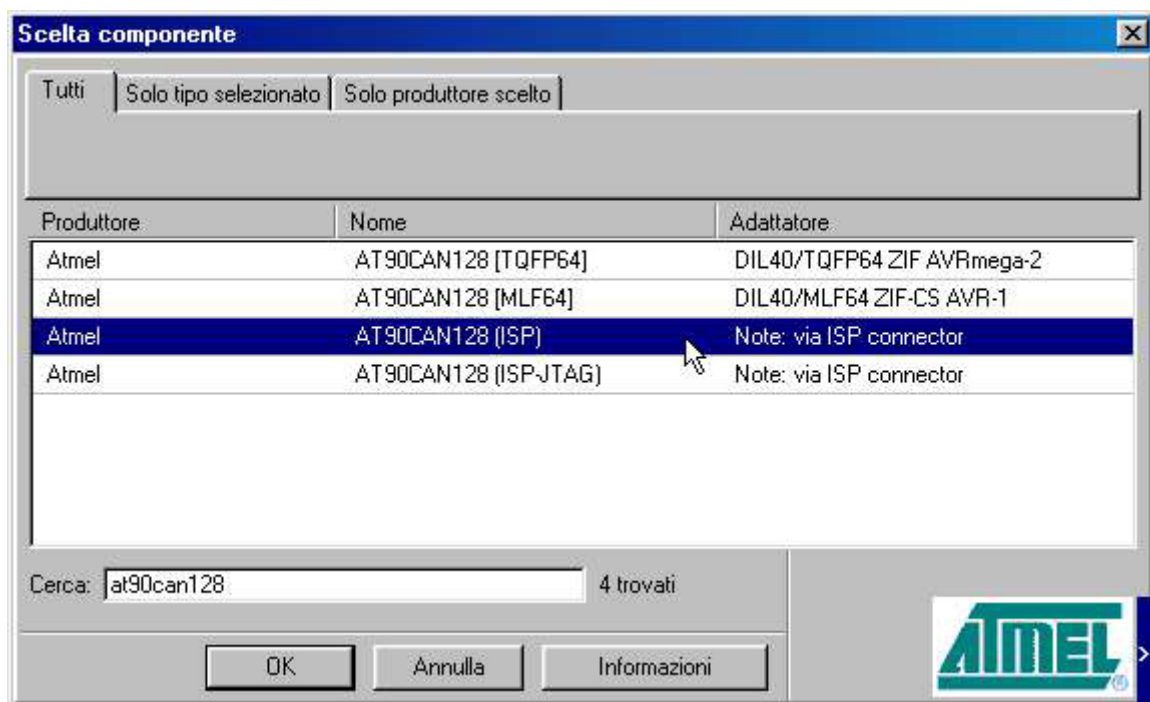
B1) Per installare, collegare al PC ed utilizzare il programmatore, si prega di fare riferimento al manuale fornito con il programmatore stesso. Nella figura di seguito viene mostrata un esempio di schermata iniziale del programma di controllo dei programmatori **grifo®**. Questo programma si chiama PG4UW e verrà indicato con tale nome nel seguito.

B2) Collegare il programmatore alla **GMM TST 2** usando l'interfaccia descritta al punto A2.

B3) Inserire il Mini Modulo nello zoccolo Z1 o Z2 di **GMM TST 2** usando l'interfaccia e le istruzioni descritte al punti A2 ed A3, poi alimentare la **GMM TST 2** come descritto nel manuale della **GMM TST 2** stessa.

B4) Nel PG4UW, selezionare come componente da programmare ATmega128 in modalità ISP, come indicato in figura B-3.

B5) Caricare il file precedentemente salvato al punto A4) usando il pulsante "Carica".



**FIGURA B-3: SELEZIONE DEL COMPONENTE CON PG4UW**

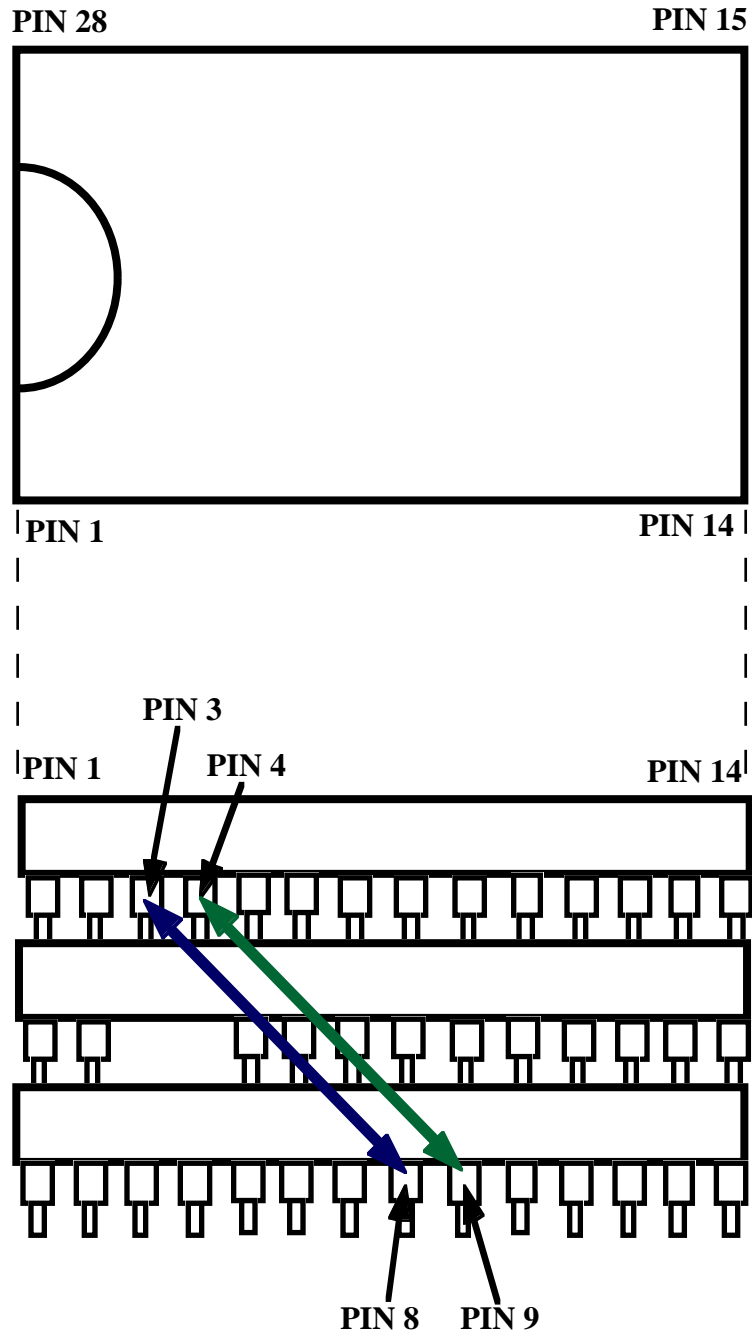


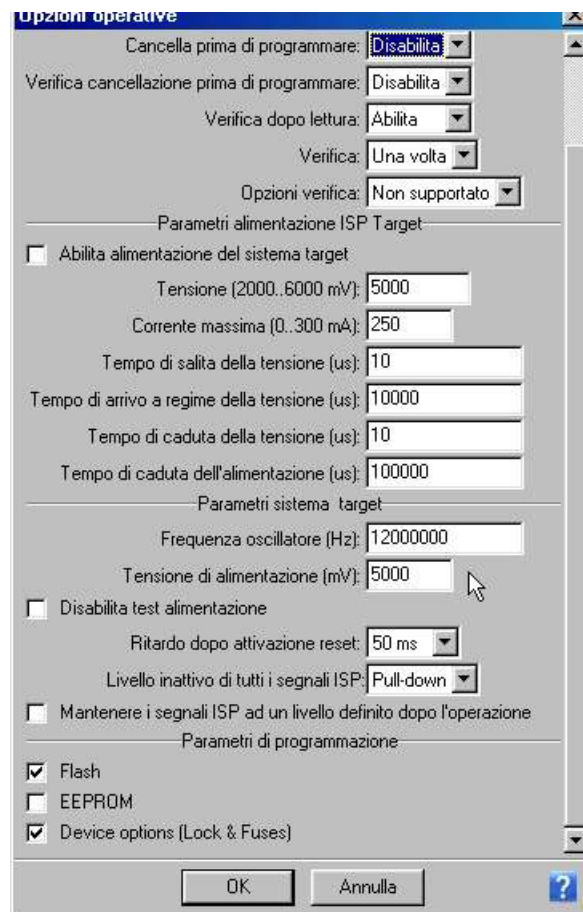
FIGURA B-4: ZOCCOLO DI PROGRAMMAZIONE ISP GMM AM128 PER GMM TST 2

B6) Configurare il componente usando il sottomenu Opzioni componente | View/Edit options and security. Il componente deve essere configurato come indicato in figura B-5, ovvero per fare funzionare il demo in prCANAVR.HEX il quarzo ed il tempo di avvio devono essere configurati come nella figura. Le altre impostazioni sono irrilevanti. Ovviamente, l'utente può effettuare la configurazione che preferisce.

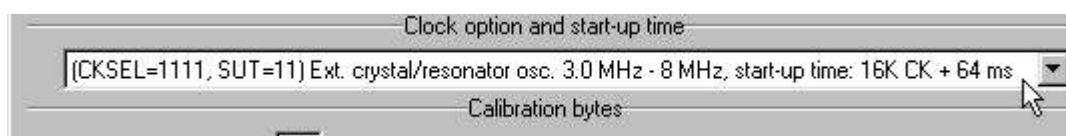
B7) Configurare il programmatore usando il sottomenu Opzioni componente | Opzioni operative. Il programmatore deve essere configurato come indicato in figura B-6.

B8) Avviare la programmazione, premendo il pulsante "Prog.".

B9) Durante la programmazione la barra progressiva avanza fino al completamento. In caso di errori, questi vengono segnalati immediatamente.



**FIGURA B-5: CONFIGURAZIONE DEL COMPONENTE CON PG4UW**



**FIGURA B-6: CONFIGURAZIONE DEL PROGRAMMATORE PG4UW**



## C) PROGRAMMAZIONE CON EQUINOX EPSILON5

C1) La programmazione del Mini Modulo a bordo della **GMM TST 2** avviene sfruttando un apposito connettore della scheda. Il programmatore Equinox EPSILON5 deve innanzitutto essere configurato fisicamente per collegarsi alla **GMM TST 2**. Per effettuare tale configurazione è necessario:

- Collegare il cavo flat da 10 vie al connettore "J7-ATMEL10" dell'EPSILON5
- Chiudere il jumper J9 dell'EPSILON5

per ulteriori informazioni si prega di fare riferimento al manuale dell'EPSILON5.

C2) Dopo avere configurato e richiuso l'EPSILON5, è necessario installare il suo programma di controllo, chiamato EQTools. Per ulteriori informazioni su come installare il programma e collegare l'EPSILON5 al PC, si faccia riferimento al manuale utente del suddetto.

C3) Collegare il programmatore al connettore CN7 della **GMM TST 2**.

C4) Inserire il Mini Modulo nello zoccolo Z1 o Z2 di **GMM TST 2** usando l'interfaccia e le istruzioni descritte ai punti A2 ed A3, poi alimentare la **GMM TST 2** come descritto nel manuale della **GMM TST 2** stessa.

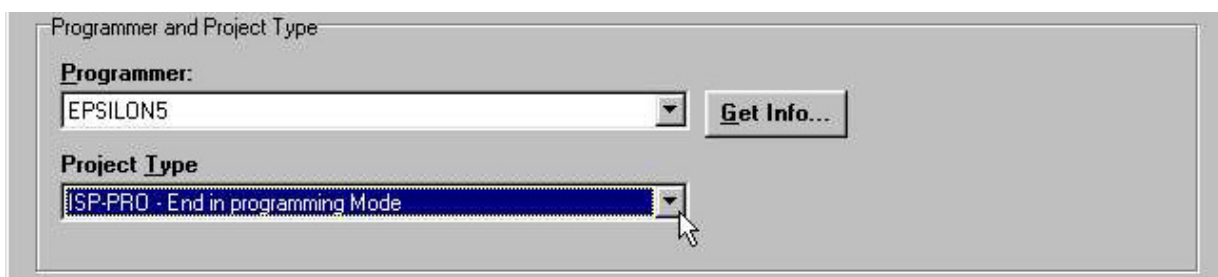
C5) Per usare l'Equinox EPSILON5 collegato al PC (anziché stand-alone) bisogna aprire un file di progetto (estensione EDS). Questo si può fare creandone uno nuovo dalla schermata iniziale che presenta EQTools all'avvio, usando gli appositi menu e pulsanti o caricando un progetto esistente. Per ulteriori informazioni sulla gestione sui file di progetto, si prega di fare riferimento al manuale di EQTools.

C6) Se si crea un nuovo progetto, bisogna assicurarsi che siano eseguite le impostazioni mostrate nelle figure in posizione B-7. Se si apre un progetto esistente, bisogna assicurarsi che siano presenti. Le impostazioni servono a garantire che:

- Il progetto venga eseguito correttamente da un EPSILON5
- Il componente target sia un AT90CAN128
- Il file che viene programmato sul target sia prCANAVR.HEX
- La configurazione del componente target sia corretta

per ulteriori informazioni sulla configurazione di un progetto, si faccia riferimento alla documentazione di EQTools.

- C7) Una volta completata la configurazione del progetto, se lo si sta creando dal nulla, bisogna indicare di usarlo in modalità "Test EDS". Per attivare l'operazione di scrittura nella memoria Flash del Mini Modulo, bisogna tornare al menu della Flash, mettere lo spunto alla casella "Edit Menu" e premere il pulsante "Write".
- C8) La finestra "Write Block to Flash" fornisce un riassunto di alcune delle impostazioni correnti, per verificarle. Se risultano corrette, premendo OK inizia il procedimento fisico di scrittura nella memoria.
- C9) Lo stato della programmazione viene indicato dalla barra progressiva, al termine una finestra indica l'esito finale dell'operazione.
- C10) Per scrivere i bit di configurazione del componente, bisogna tornare al menu apposito ("Fuses") e premere il pulsante "Write" nel riquadro "Target Fuses".



Fuses	PC Fuse State	Target Fuse State	Description
CKSEL0	1 - UNPROGRAMMED	?	Clock Select 0
CKSEL1	1 - UNPROGRAMMED	?	Clock Select 1
CKSEL2	1 - UNPROGRAMMED	?	Clock Select 2
CKSEL3	1 - UNPROGRAMMED	?	Clock Select 3
SUT0	1 - UNPROGRAMMED	?	Oscillator - Select Start Up Time 0
SUT1	1 - UNPROGRAMMED	?	Oscillator - Select Start Up Time 1
BODEN	1 - UNPROGRAMMED	?	Brown-Out Detector - ENABLE

FIGURA B-7: CONFIGURAZIONE PROGETTO CON EQTOOLS

## D) *PROGRAMMAZIONE CON ATMEL AVR ISP*

- D1) Il programma di controllo dell'**AVR ISP** è **AVR STUDIO**, versione 4 o superiori. L'ultima versione è scaricabile dal sito Atmel [www.atmel.com](http://www.atmel.com). Scaricatelo e installatelo seguendo le istruzioni a video.
- D2) Configurare **AVR ISP** per usare il flat da 10 vie e collegarlo a CN7 di **GMM TST 2**, collegare **AVR ISP** alla porta seriale del PC (vedere istruzioni di **AVR ISP**), configurare la **GMM TST 2** per programmare con **AVR ISP**, seguendo le istruzioni dei punti A2 ed A3, e alimentarla (vedere manuale **GMM TST 2**).
- D3) Lanciare **AVR STUDIO**. Il programma di controllo di **AVR ISP** si attiva premendo il pulsante con il chip AVR come icona.
- D4) Selezionare come CPU AT90CAN128, come indicato in figura B-8.
- D5) Caricare il file precedentemente salvato al punto A4 usando il pulsante "Carica".
- D6) Configurare la CPU come indicato nelle immagini di figura B-9.
- D7) Configurare il programmatore per effettuare la verifica dell'ID, cancellare il dispositivo e riprogrammare con verifica la FLASH e i bit di configurazione, come in figura B-10.
- D8) Eseguire la programmazione premendo il tasto "Start" come indicato nella figura B10.

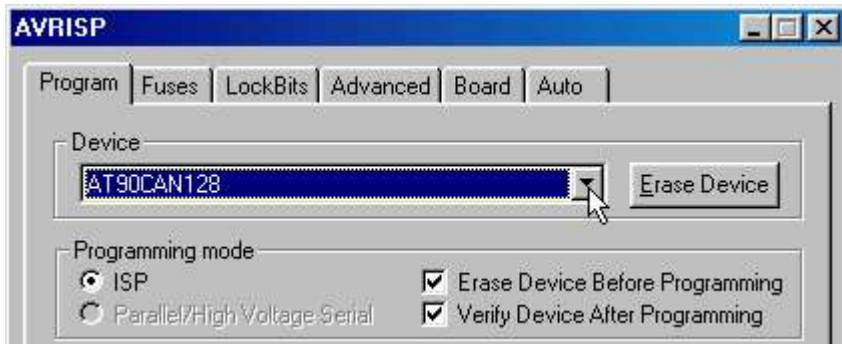


FIGURA B-8: SELEZIONE CPU CON AVR STUDIO

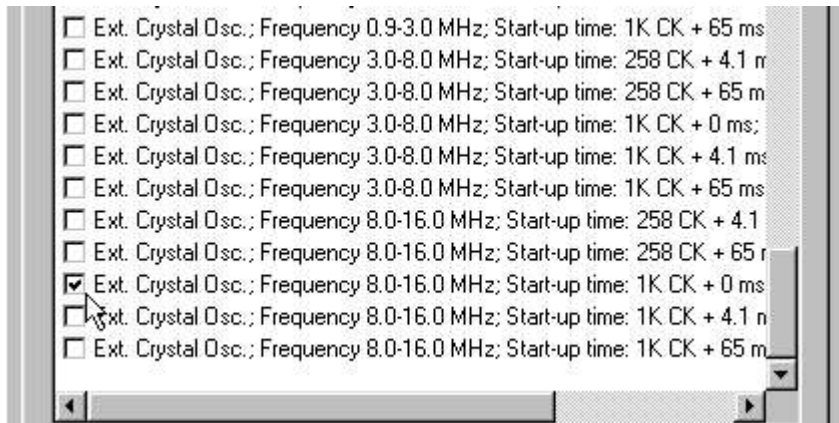


FIGURA B-9: CONFIGURAZIONE CPU CON AVR STUDIO

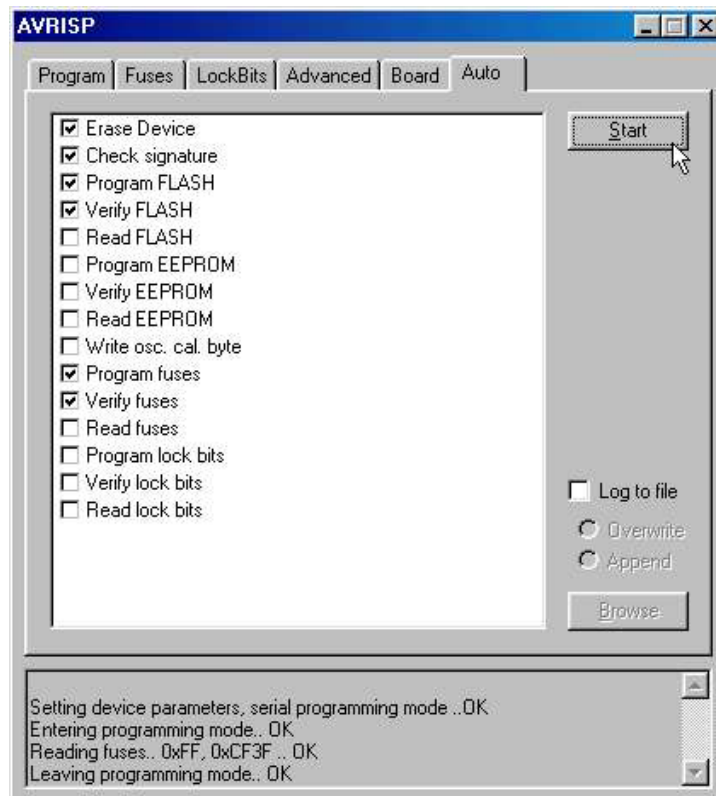


FIGURA B-10: CONFIGURAZIONE AVR ISP

APPENDICE C: SCHEMA ELETTRICO CAN GMT

In questa appendice è disponibile lo schema elettrico della scheda di supporto **CAN GMT** che illustra alcune modalità di connessione dei segnali dei Mini Moduli. Informazioni più dettagliate su questa scheda sono disponibili nel relativo manuale tecnico e l'utente le può usare liberamente ad esempio per realizzare una propria scheda che usa la **CAN AVR** come macro componente.

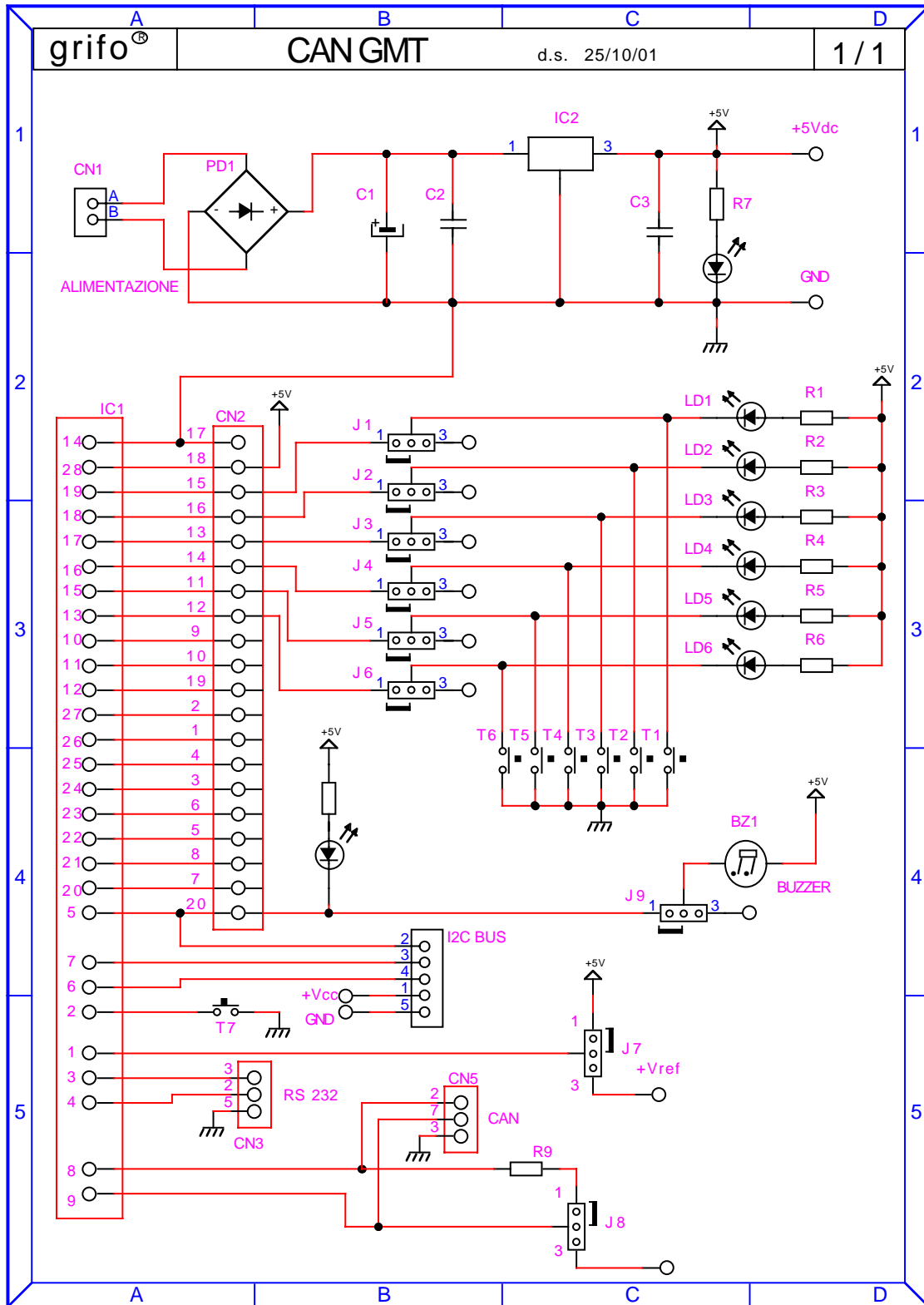


FIGURA C-1: SCHEMA ELETTRICO CAN GMT



## APPENDICE D: INDICE ANALITICO

**SIMBOLI**

+VDC POW 20  
/INTRTC 15  
/RESET 15

**A**

A/D CONVERTER 5, 9, 10, 12, 15, 20  
ALIMENTAZIONE 13, 15, 20  
AVR ISP B-1  
AREA PROTOTIPALE 24  
AVR STUDIO 34, B-9  
AVRBOOTLOADERGRIFO 28

**B**

BACK UP 13  
BASCOM AVR 30, 34  
BIBLIOGRAFIA 38  
BOOT LOADER 5, 28  
BROWN OUT 13

**C**

CAN 9, 15, 16, 21, 24, 37  
CAN GMT 24  
CLOCK 6, 12, 15  
CN1 14  
COLLEGAMENTO SERIALE 26  
COME INIZIARE 26  
COMPARATORE ANALOGICO 10, 12, 15, 20  
COMUNICAZIONE SERIALE 18  
CONFIGURAZIONE MINI MODULO 16  
CONNESSIONI 14  
CONSUMO DI CORRENTE 13  
CPU 12, 37  
CURRENT LOOP 6, 9

**D**

DESCRIZIONE SOFTWARE 34  
DIMENSIONI 13  
DIP SWITCH 9, 12, 16, 18, 36, B-2  
DSW1 16, 17, 18, 36, B-2  
DSW2 16, B-2

**E**

EEPROM 5, 8, 12, 28  
EPROM 8, 12  
EPSILON5 B-7  
ERRORE MASSIMO A/D 12

**F**

FLASH 5, 8, 12, 28  
FREQUENZA MASSIMA CONTATORI 12

**G**

GMB HR84 22  
GMM TST 2 B-1

**I**

I/O DIGITALE 5, 6, 12, 15  
I<sup>2</sup>C BUS 5, 8, 10, 12, 15, 20, 24, 37  
ICC AVR 30, 32, 34  
IMPEDENZA GENERATORI SEGNALI ANALOGICI 13  
INTERRUPT 5, 12, 15, 21  
ISP 5, 15, 28, B-1

**J**

JTAG 5, 15, 21, 34

**L**

LD1 14, 36  
LD2 14, 36  
LEDS 12, 14, 36

**M**

MEMORIE 8, 12  
MISO 8, 15  
MOSI 8, 15  
MPAVR-51+ B-1, B-4  
MSI01 6

**P**

PESO 13  
POWER ON 12  
PROGRAMMAZIONE 28  
PROTEZIONE RS 232 13  
PWM 10, 12, 20



**R**

RANGE DI TEMPERATURA 13  
REAL TIME CLOCK 5, 10, 12, 37  
RESET 15  
RESISTENZA DI PULL-UP SU I2C BUS 13  
RISOLUZIONE A/D CONVERTER 12  
RISORSE 12  
RS 232 5, 6, 12, 15, 16, 18, 20, 24  
RS 422 6, 9  
RS 485 6, 9  
RTC 6, 8, 15, 37

**S**

SCK 8  
SCL 20, 37  
SDA 20, 37  
SERIALE 6, 18, 26  
SOGLIE DEL BROWN OUT 13  
SPI 5, 8, 12, 15, 20  
SRAM 5, 8, 10, 12, 37

**T**

TCK 15, 21  
TDI 15, 21  
TDO 15, 21  
TEMPERATURA 13  
TEMPO DI CONVERSIONE A/D 12  
TEMPO DI POWER ON 12  
TEMPO INTERVENTO WATCH DOG 12  
TIMER COUNTER 5, 10, 12, 15  
TMS 15, 21  
TTL 6, 15, 16, 18, 20, 24

**U**

UEP49 B-1, B-4  
UMIDITÀ RELATIVA 13  
USART0 9, 15, 18

**V**

VERSIONE SCHEDA 3

**W**

WATCH DOG 5, 12

