



Paolo Rognoni (Paolino)

LO HAI MAI REALIZZATO CON UN PIC? RTCC - IL PIC TIENE LA DATA E L'ORA

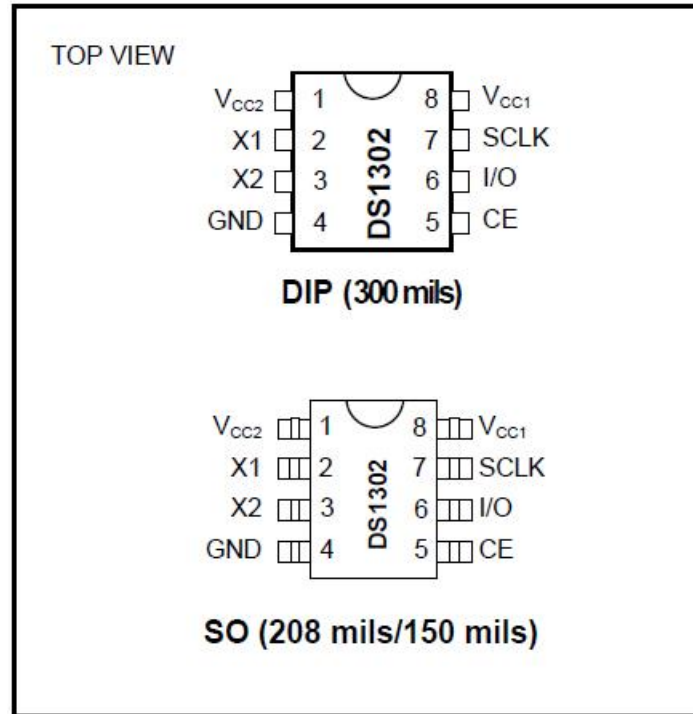
5 November 2011

Nelle applicazioni a microcontrollore uno dei progetti cui ci si imbatte spesso, soprattutto per chi è all'inizio (ma non solo), è la realizzazione di un **orologio digitale** dotato di **calendario**. Il conteggio del tempo non è di per sé un affare complicato, avendo a che fare con sistemi dotati di oscillatori! Il primo approccio è quindi quello di usare il clock di sistema, dividerlo con vari prescaler e/o postscaler ed assegnare il conteggio ad un registro conta secondi: 60 secondi per un minuto, 60 minuti formano l'ora, 24 ore per un giorno, 365 (o 366) giorni per un anno, eccetera... Se poi ci mettiamo il conteggio dei mesi con la composizione della data (comprensiva dei giorni della settimana) il progetto "semplice" di un orologio digitale diventa invece abbastanza complicato. Con tutti i distinguo del caso.

Per le applicazioni in cui l'ora è sì un aspetto importante (ad esempio una centralina di irrigazione) ma la sua gestione non deve richiedere troppe risorse sia in termini di pin di I/O sia di memoria occupata, l'impiego di chip dedicati è la soluzione migliore. Il dispositivo illustrato, benché non sia di ultimissima generazione, è ottimo nelle applicazioni con microcontrollori privi della porta SPI oppure quando la porta SPI è già destinata ad altri scopi e non può essere distolta da compiti differenti.

Real Time Clock Calendar

Uno tra i più famosi ed utilizzati chip **RTCC (Real Time Clock Calendar)** è il **Dallas DS1302**. È un chip a 8 pin, con interfaccia seriale sincrona, del tutto simile a SPI ma con specifiche differenti.

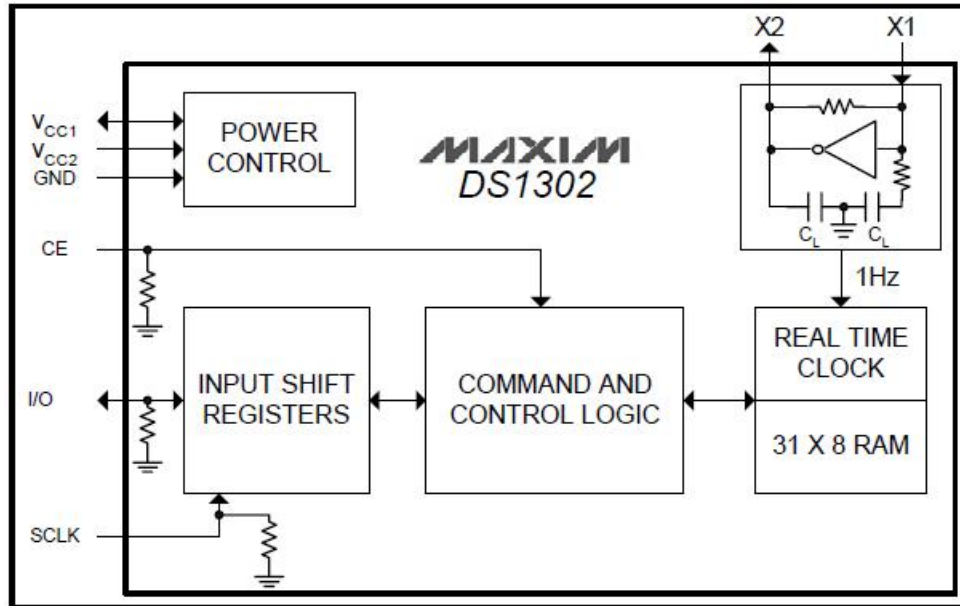


Pinout DS1302

Qui di seguito la descrizione dei pin:

Pin Denominazione	Funzione svolta
1 V _{cc2}	Alimentazione principale
2 X1	Ai pin X1 ed X2 si collega un quarzo da 32768 Hz (vedi datahseet)
3 X2	Ai pin X1 ed X2 si collega un quarzo da 32768 Hz (vedi datahseet)
4 GND	Connessione di massa
5 CE	Chip Enable. Questo segnale va tenuto alto durante le operazioni di lettura o scrittura. Questo pin ha un pulldown interno di 40 kOhm (valore tipico)
6 I/O	Pin di I/O digitale; su questa linea transitato in dati dal micro al modulo RTCC e viceversa
7 SCLK	Serial CLOCK. Pin usato per la generazione del segnale di clock usato per sincronizzare le operazioni di I/O bidirezionali. Questo pin ha un pulldown interno di 40 kOhm (valore tipico)
8 V _{cc1}	Alimentazione secondaria, utilizzata per alimentare DS1302 quando l'alimentazione principale non è disponibile. A questo pin va collegata una batteria tampone.

Il dispositivo ben si adatta per quelle applicazioni in cui è richiesta l'alimentazione a batteria oppure, nel caso di applicazioni alimentate da rete elettrica, quando non si vuole perdere l'ora in caso di black-out. Il datasheet mostra come è costituito all'interno il modulo RTCC:

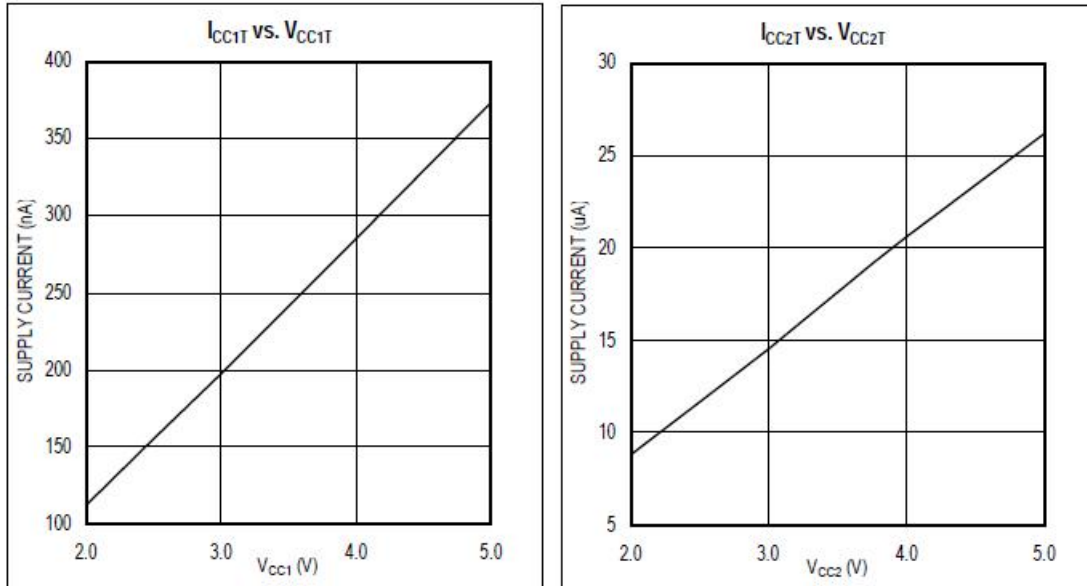


Blocchi logici interni a DS1302

Nei diagrammi che seguono è mostrato l'assorbimento in corrente dai pin V_{cc1} e V_{cc2} ; con alimentazione da batteria (V_{cc1}), il consumo di DS1302, nel pieno dello svolgimento delle sue funzioni di mantenimento aggiornato il calendario, è di qualche centinaia di nA, valori certamente accettabili per la maggior parte delle applicazioni.

TYPICAL OPERATING CHARACTERISTICS

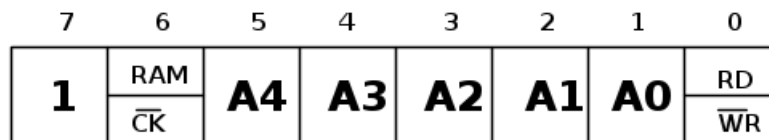
($V_{CC} = 3.3V$, $T_A = +25^\circ C$, unless otherwise noted.)



Correnti assorbite da DS1302

Ma come si comanda?

Il dispositivo è in grado di ricevere comandi dal microcontrollore e di fornire i dati richiesti. La formattazione del pacchetto è a byte ed è strutturato come indicato in figura.



Byte di indirizzamento e comando

Il significato dei bit è il seguente:

Bit	Significato
------------	--------------------

Va posto a 1 per le operazioni di lettura ed a 0 per quelle di scrittura. Il byte di comando 0 è sempre un'operazione di lettura, quindi questo bit va posto a 1 per inviare i comandi.

- 1 Bit 0 dell'indirizzo del registro interno che si vuole leggere e/o scrivere.
- 2 Bit 1 dell'indirizzo del registro interno che si vuole leggere e/o scrivere.
- 3 Bit 2 dell'indirizzo del registro interno che si vuole leggere e/o scrivere.

- 4 Bit 3 dell'indirizzo del registro interno che si vuole leggere e/o scrivere.
- 5 Bit 4 dell'indirizzo del registro interno che si vuole leggere e/o scrivere.
- 6 Va posto a 1 per i dati di calendario ed orologio, mentre va posto a 1 per accedere alla RAM interna.
- 7 Deve **sempre assumere valore 1**, altrimenti ogni operazioni di scrittura su DS1302 viene disabilitata.

I comandi vanno scritti e letti secondo questo digramma di timing:

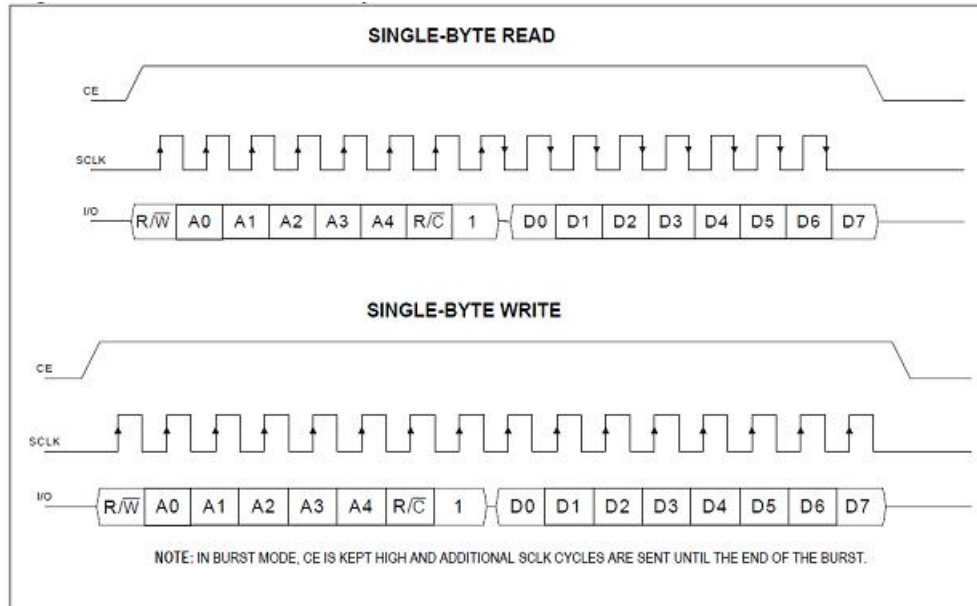


Diagramma di timing

Si noti come la sequenza è sempre quella di due byte: il comando precede il dato. Nel caso di lettura di un dato, questa operazione avviene sul fronte di discesa del segnale di clock, mentre la scrittura è attiva sul fronte di salita.

I comandi che il dispositivo rende disponibili sono raggruppati nella seguente tabella:

RTC										
READ	WRITE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	RANGE
81h	80h	CH		10 Seconds		Seconds				00-59
83h	82h			10 Minutes		Minutes				00-59
85h	84h	12/24	0	10 AM/PM	Hour	Hour				1-12/0-23
87h	86h	0	0	10 Date		Date				1-31
89h	88h	0	0	0	10 Month	Month				1-12
8Bh	8Ah	0	0	0	0	0	Day			1-7
8Dh	8Ch	10 Year				Year				00-99
8Fh	8Eh	WP	0	0	0	0	0	0	0	—
91h	90h	TCS	TCS	TCS	TCS	DS	DS	RS	RS	—

Tabella dei comandi disponibili

Si noti che i byte che contengono i dati presentano dei valori estremi, pertanto le ore sono comprese tra 00 e 23 (nel caso di gestione dell'orario su 24 ore), i minuti ed i secondi stanno tra 00 e 59, e così via.

Un po' di codice

Tra i compilatori C di terze parti che Microchip annovera come ufficiali e che sono disponibili con l'installazione di MPLAB, è presente anche il compilatore PCB di CCS (vedi riferimenti in bibliografia), in versione limitata. Tra i vari file forniti con il compilatore ce ne è uno che interfaccia DS1302. Prendendo spunto da quel file, ho rimodellato il codice C secondo le mie esigenze, andando a modificare alcune funzioni proprie del compilatore PCB ma che non andavano bene nella generalità dei casi. I file, (disponibili nella sezione download di questo articolo) racchiudono le funzioni di gestione del dispositivo e i comandi da fornire per leggere e scrivere i dati. Un esempio è riportato nel seguito:

```
/*  
  
DS1302 RTC interface  
  
*/  
  
/* GLOBAL VARIABLES */  
extern unsigned char RTCDS1302_hr,RTCDS1302_min,RTCDS1302_sec;  
extern unsigned char RTCDS1302_day,RTCDS1302_month,RTCDS1302_year,RTCDS1302_dow;  
  
/* WRITE AND READ COMMANDS */  
#define W_HR          0x84  
#define W_MIN         0x82  
#define W_SEC         0x80  
#define W_DAY         0x86  
#define W_MONTH       0x88  
#define W_YEAR        0x8C  
#define W_DOW         0x8A  
#define W_TRICKLE     0x90  
#define W_CONTROL     0x8E  
  
#define R_HR          0x85  
#define R_MIN         0x83  
#define R_SEC         0x81  
#define R_DAY         0x87
```

```
#define R_MONTH      0x89
#define R_YEAR       0x8D
#define R_DOW        0x8B
#define R_TRICKLE    0x91
#define R_CONTROL    0x8F

/* FUNCTION PROTOTYPES */

extern void write_ds1302_byte (unsigned char );
extern void ds1302_io_definition (void);
extern void write_ds1302 (unsigned char , unsigned char );
extern unsigned char read_ds1302 (unsigned char );
extern void rtc_init (unsigned char);
extern unsigned char get_bcd (unsigned char );
extern unsigned char rm_bcd (unsigned char );

extern void rtc_set_datetime (unsigned char , unsigned char , unsigned char , unsigned
extern void rtc_set_time(unsigned char hr, unsigned char min);
extern void rtc_get_date (void);
extern void rtc_get_time (void);
extern void rtc_write_nvr (unsigned char , unsigned char );
extern unsigned char rtc_read_nvr (unsigned char );
```

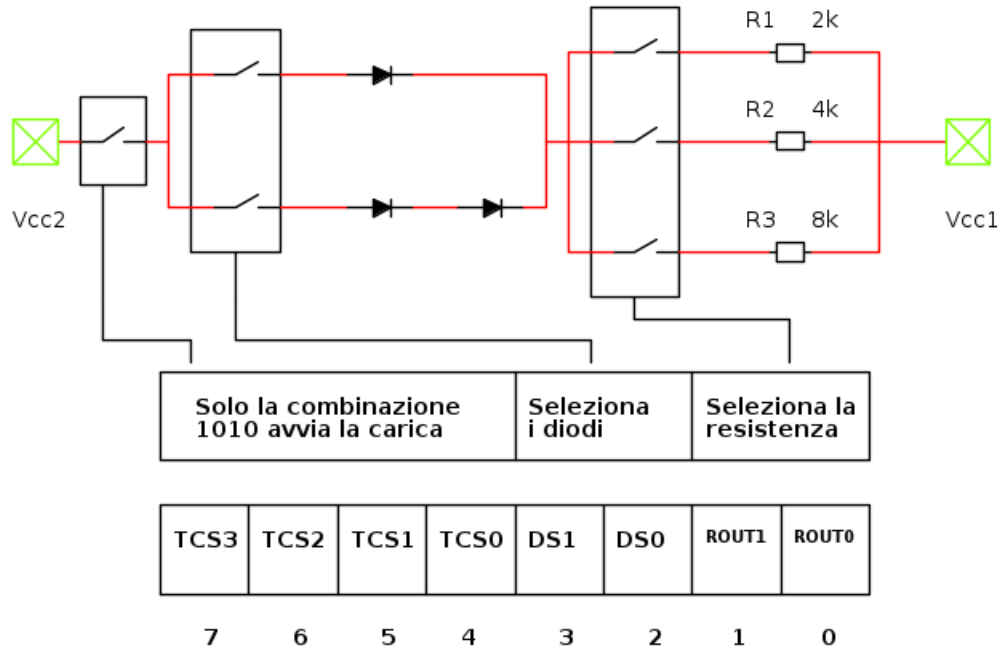
Un esempio dell'uso delle funzioni è il seguente:

```
void main(void)
{
    InitPic (); // Inizializza il micro
    rtc_init (0xA7); // Trickle charger: 1 diodo, R = 8k
    rtc_set_datetime(05,11,11,07,18,15); // 05/11/2011, Sabato, ore 18.15
    while (1)
    {
        rtc_get_date();
        rtc_get_time();
        // ....
    }
}
```

Trickle charger

Il DS1302 può anche caricare una piccola batteria, quella di back-up, od un [supercap](#). In questo modo è possibile tenere alimentata la batteria che, in caso di black-out, mantiene l'orologio in funzione. Per far ciò, è necessario trasferire un byte specifico,

byte che contiene le informazioni circa la corrente di carica. Tale corrente è stabilita da una selezione di diodi e di resistenze, interposti tra la tensione Vcc2 e la Vcc1, secondo il seguente schema:



L'equazione che stabilisce la massima corrente che DS1302 è in grado di fornire alla batteria vale: $I_{max} = \frac{V_{cc2} - V_{diodi}}{R}$

Con una tensione di alimentazione di 5V, selezionando un diodo e la resistenza da $4k\Omega$, supponendo che i diodi abbiano una caduta di tensione di circa 0.7V ciascuno, il valore di corrente massima per la carica della batteria vale:

$$I_{max} \simeq \frac{5 - 0.7}{4k\Omega} \simeq 1.075mA$$

La scelta del numero di diodi e il valore della resistenza avviene mediante la composizione del "byte di carica", secondo questa tabella:

TCS3	TCS2	TCS1	TCS0	DS1	DS0	ROUT1	ROUT0	Funzione
X	X	X	X	X	X	0	0	Trickle charger disabilitato
X	X	X	X	0	0	X	X	Trickle charger disabilitato
X	X	X	X	1	1	X	X	Trickle charger disabilitato
1	0	1	0	0	1	0	1	1 diodo, R = $2k\Omega$
1	0	1	0	0	1	1	0	1 diodo, R = $4k\Omega$
1	0	1	0	0	1	1	1	1 diodo, R = $8k\Omega$

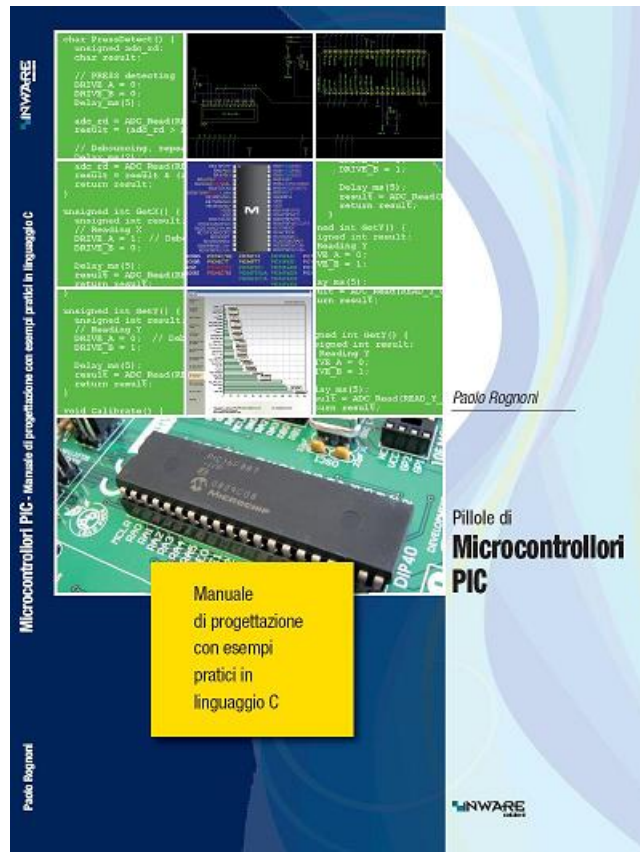
1	0	1	0	1	0	0	1	2 diodi, R = 2k Ω
1	0	1	0	1	0	1	0	2 diodi, R = 4k Ω
1	0	1	0	1	0	1	1	2 diodi, R = 8k Ω
0	1	0	1	1	1	0	0	Condizione iniziale

Download

I due file sorgenti DS1302.c e DS1302.h sono scaricabili a [questo indirizzo](#).

Bibliografia

- **Datasheet DS1302:** <http://pdfserv.maxim-ic.com/en/ds/DS1302.pdf>
- **Ambiente di sviluppo MPLAB:** <http://www.microchip.com/mplab>
- **Compilatore C18 per PICMicro:** <http://www.microchip.com/C18>
- **Compilatore CCS:** <http://www.ccsinfo.com>
- **Sito web [PicExperience](#)**
- **Collana "LO HAI MAI REALIZZATO CON UN PIC?":**
 - [Il contamarce](#)
 - [Una sorpresa musicale per Babbo Natale](#)
 - [Una tecnica antirimbalzo](#)
 - [Il dado elettronico](#)
 - [Un approccio ai timer dei PICMicro](#)
 - [I PIC e i segnali analogici: la conversione A/D](#)
 - [Generare segnali PWM con i PICMicro](#)
 - [La lampada SIBILLA!](#)
 - [Una tecnica di misura della tensione di batteria](#)
- **[Pillole di microcontrollori PIC:](#)**



Pillole di Microcontrollori PIC

Estratto da ["http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Paolino:lo-hai-mai-realizzato-con-un-pic-rtcc-il-pic-tiene-l-ora"](http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Paolino:lo-hai-mai-realizzato-con-un-pic-rtcc-il-pic-tiene-l-ora)