

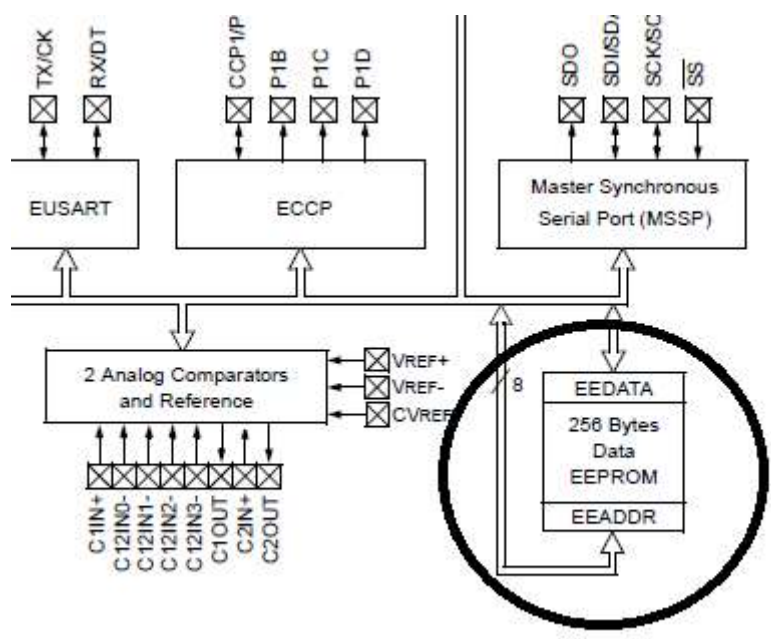
# EEPROM nel microcontrollore

## PIC16F887

In questa dispensa vi spiego quanto è facile salvare dati utilizzando la **EEPROM presente nel microcontrollore PIC16F887**. Nella maggior parte dei progetti in tempo reale potremmo dover salvare alcuni dati che non devono essere cancellati anche quando l'alimentazione viene spenta. Potrebbe essere il caso in cui stiamo sviluppando una cassaforte e dobbiamo fare in modo che l'acquirente possa memorizzare un proprio codice d'accesso.

Come vi ho già spiegato, EEPROM sta per "Memoria di sola lettura cancellabile e programmabile elettronicamente". Come suggerisce il nome, è una memoria presente all'interno del microcontrollore PIC in cui possiamo scrivere / leggere i dati facendolo attraverso righe di programma. La quantità di spazio di archiviazione disponibile in EEPROM varia su ciascun microcontrollore; i dettagli saranno forniti nei Datasheet.

Nel nostro caso per PIC16F887 lo spazio disponibile è di 256 byte, come indicato nel suo Datasheet riportato sotto.



Ora vediamo come possiamo usare questi 256 byte per leggere / scrivere i dati usando una semplice configurazione sperimentale.

I registri che ci interessano sono:

### REGISTER 10-1: EEDAT: EEPROM DATA REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0
bit 7							bit 0

Questo registro serve, in caso di scrittura, a per caricare il dato che si vuole memorizzare mentre in caso di lettura in questo registro viene memorizzato dal micro il valore della memoria letto.

**REGISTER 10-2: EEADR: EEPROM ADDRESS REGISTER**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
bit 7							bit 0

Il registro EEADR permette di indicare l'indirizzo della locazione di memoria in cui si vuole scrivere o si vuole leggere.

**REGISTER 10-5: EECON1: EEPROM CONTROL REGISTER**

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD
bit 7							bit 0

Quest'ultimo è il registro per la gestione della memoria. Il bit EEPGD (Program/Data EEPROM Select bit) se resettato permette di scegliere se si vuole accedere alla memoria EEPROM Data (ossia la memoria di cui ci stiamo occupando) mentre se è settato permette di accedere alla EEPROM Program (cioè la memoria in cui è scritto il programma), infatti nel Datasheet troviamo altri due registri EEDATH e EEADRH che permetterebbero di completare il valore (14 bit) e l'indirizzo (13 bit) per leggere/ scrivere la memoria programma. Nel nostro caso ci assicureremo di tenere il bit EEPGD sempre a 0.

Per l'operazione di lettura si procede secondo quanto disposto nella funzione EEPROM\_RD riportata sotto:

```
char EEPROM_rd (char indirizzo)
{
    EEADR = indirizzo;           // copio l'indirizzo nel registro EEADR
    EECON1bits.EEPGD = 0;       // puntamento alla memoria eeprom dati
    EECON1bits.RD = 1;          // effettuo la lettura

    return (EEDAT);              // restituisco il dato letto
}
```

Come potete vedere è una funzione molto semplice, basta passare il parametro riguardante l'indirizzo in cui si vuole leggere (da 0 a 255) e tale indirizzo viene caricato nel registro EEADR. Successivamente ci si assicura, attraverso il bit EEPGD, di voler utilizzare la EEPROM Data e per effettuare la lettura basta settare il bit RD di EECON1 ed il dato sarà caricato dall'hardware sul registro EEDAT.

L'operazione di scrittura è un po' più delicata!!!!. Vediamo....

```
void EEPROM_wr (char indirizzo, char dato)
{
    EEADR = indirizzo;
    EEDAT = dato;
    EECON1bits.EEPGD = 0;       // puntamento alla memoria eeprom dati
    EECON1bits.WREN = 1;        // abilito la scrittura sulla memoria EEPROM
}
```

```

INTCONbits.GIE = 0;           // disabilito tutti gli interrupt

EECON2 = 0x55;               // sequenza richiesta
EECON2 = 0xAA;

EECON1bits.WR = 1;           // avvia scrittura

while (EECON1bits.WR);        // aspetto la fine della scrittura

INTCONbits.GIE = 1;           // abilito tutti gli interrupt

EECON1bits.WREN = 0;          // disabilito la scrittura sulla memoria EEPROM
}

```

Innanzitutto bisogna passare alla funzione i seguenti parametri: l'indirizzo della locazione di memoria in cui si intende scrivere (da 0 a 255) ed ovviamente il valore da scrivere (da 0 a 255) che verranno rispettivamente caricati nei registri EEADR e EEDAT. Come è stato spiegato già prima, il bit EEPGD va messo a zero.

Il processo di scrittura vero e proprio si compone dei seguenti passi:

1. Settare il bit WREN (Write Enable) per comunicare al micro l'inizio della procedura di scrittura.
2. Portare a 0 il GIE perché durante la procedura di scrittura non possono essere ammesse interruzioni che potrebbero compromettere il processo stesso.
3. Caricare nel registro EECN2 la sequenza di valori 0x55 e successivamente 0xAA. Considerate queste due istruzioni come una chiave hardware richiesta dal micro.
4. Quando poi si setta il bit WR avviene la scrittura vera e propria della memoria. Terminata la scrittura, questo bit viene riportato a 0 dall'hardware del micro.
5. Ora si possono riabilitare gli eventuali interrupt ponendo GIE ad 1.
6. Il processo di scrittura termina riportando il bit WREN a 0.

A titolo di esempio nel MAIN si usa la funzione di scrittura per scrivere valori da 0 ad 8 nella memoria:

```

for (ind=0; ind<=8; ind++)    // scrivo 8 valori di 'val' in 9 locazioni EEPROM diverse
{
    EEPROM_wr (ind, val);
    val++;
}

```

e poi gli stessi vengono letti e caricati su PORTD.

```

for (ind=0; ind<=8; ind++)    // leggo le 9 locazioni scritte precedentemente
{
    dato = EEPROM_rd(ind);

    PORTD = dato;              // e le carico su PORTD

    __delay_ms(500);
}

```