Paolo Rognoni (Paolino)

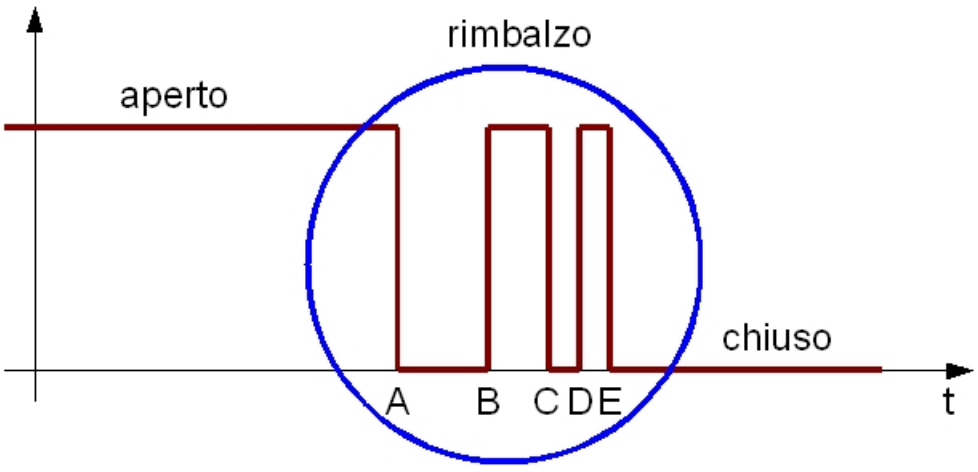
LO HAI MAI REALIZZATO CON UN PIC?
UNA TECNICA ANTIRIMBALZO

3 January 2010

Molti progetti realizzati con microcontrollore sfruttano sistemi di I/O per l'interazione tra l'Uomo e l'elettronica. Senz'altro il sistema di immissione dati in un microcontrollore è il pulsante; l'idea che il contatto presenti solo due stati, aperto o chiuso, è ottima per la gestione di segnali binari (0, 1). Ma attenzione alle insidie che si nascondono dietro ad un aspetto che, se trascurato, porta a malfunzionamenti e a comportamenti inattesi del circuito: **il rimbalzo del contatto**.

Cos'è il rimbalzo

Questo fenomeno è tipico dei contatti meccanici i quali portandosi dallo stato di aperto a quello di chiuso non lo fanno in maniera istantanea. Ciò che accade a livello macroscopico è il passaggio netto da aperto a chiuso, ma analizzando in dettaglio il comportamento del contatto si nota che il passaggio dallo **stato stabile aperto** a quello **stabile chiuso** è caratterizzato da stati intermedi di chiuso e aperto. Il contatto rimbalza nel vero senso della parola cioè prima di portarsi in modo certo alla posizione di contatto chiuso, si apre e si chiude in successione alcune volte.



Rimbalzo.jpg

La durata del rimbalzo dipende da diversi fattori:

- il tipo di contatto;

- la presenza di una molla di richiamo;
- la forza con cui viene azionato;
- la presenza di archi elettrici;
- ecc.

Per le applicazioni a microcontrollore si può considerare concluso un rimbalzo quando il contatto assume valore stabile entro un periodo di tempo compreso tra 30 e 50 ms.

Gli stati intermedi, che si manifestano in successione durante la chiusura del contatto, rischiano di essere acquisiti dal microcontrollore fornendo un'informazione non corretta al firmware che gestisce l'applicazione; per questo motivo il rimbalzo va eliminato. Per evitare di aggiungere componenti esterni al microcontrollore (fatto salvo un eventuale resistore di pull-up o pull-down, qualora non si possano utilizzare quelli interni al micro), il metodo che viene illustrato si basa esclusivamente sull'utilizzo di tecniche firmware. Benché si faccia esplicito riferimento ai **PICMicro**, questa metodologia può essere utilizzata anche con altri microcontrollori, andando ad attivare opportunamente i meccanismi hardware interni.

Eliminare il rimbalzo

L'obbiettivo è quindi quello di misurare uno **stato stabile entro un periodo di tempo prefissato**. Per dare esempio di come si può fare, stabiliamo:

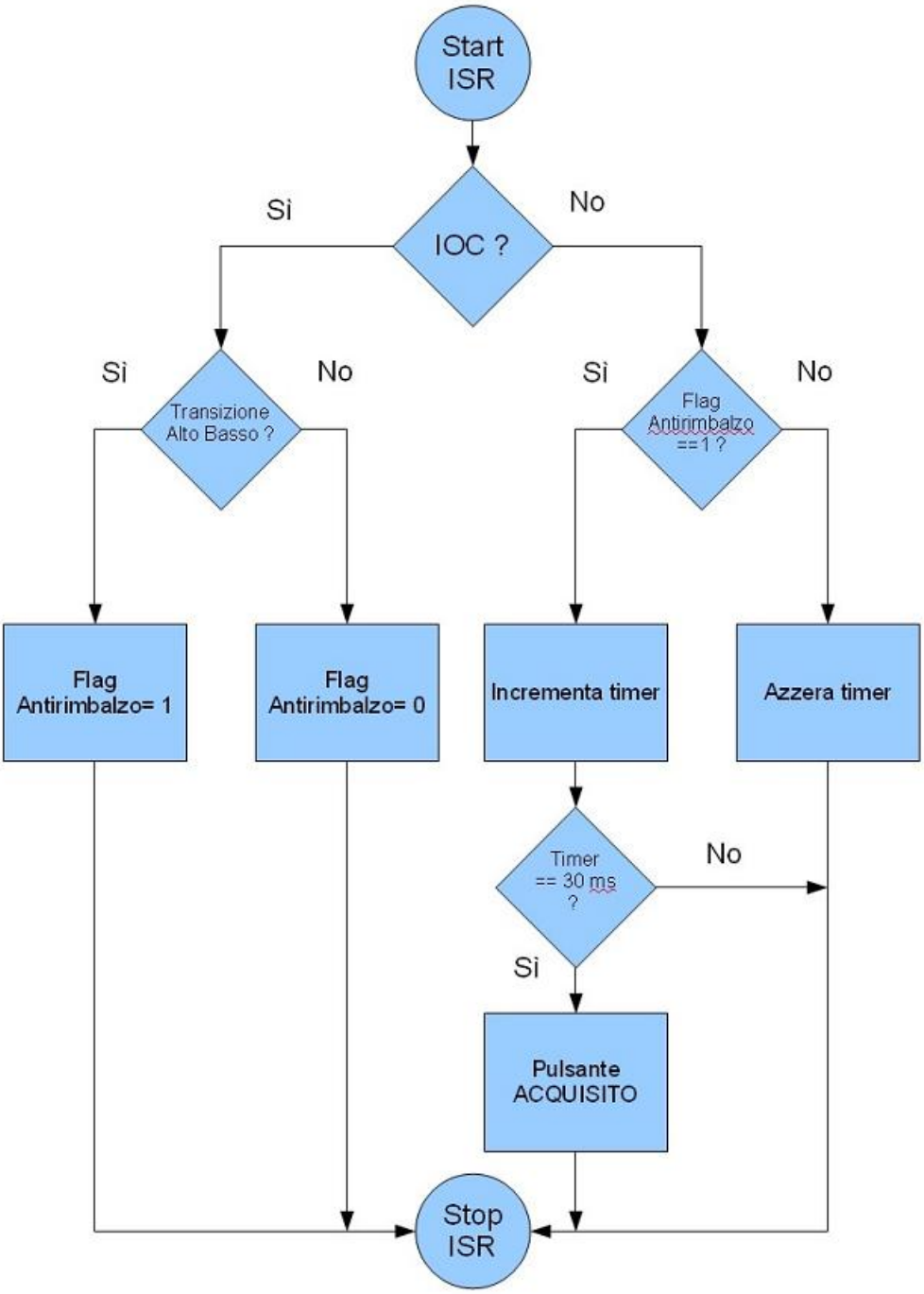
- tipo contatto: normalmente aperto

- stato iniziale: aperto

- stato finale: chiuso

- periodo di antirimbalzo: 30ms

Il comportamento del PIC è il seguente: non appena il pulsante cambia di stato, portandosi da aperto a chiuso, viene avviato un timer; se il valore dello stato non cambia entro i 30 ms stabiliti, il comando impartito dal pulsante è considerato valido. Se invece avviene un rimbalzo come prima cosa il timer viene resettato ed il PIC resta in attesa di una nuova commutazione da aperto a chiuso. Così facendo, i rimbalzi che si manifestano nell'intervallo **A-E** (vedi figura precedente) vengono completamente ignorati. Gli stati stabili aperto e chiuso vengono presi in considerazione senza che il rimbalzo possa ingannare il PIC. In figura è riportato il flow chart che gestisce l'antirimbalzo, con la descrizione a blocchi della routine di interrupt.



DebounceFlowChart.jpg

Un esempio

La routine di interrupt che gestisce l'antirimbalzo, viene proposta in linguaggio C per MikroC PRO, ma è facilmente esportabile in assembly. In questo caso, il sorgente

è stato preso dall'esempio MerryChristmas che utilizza un PIC12F675/PIC16F683 per la riproduzione di alcuni brani audio. Qui nel seguito è riportato un estratto del codice sorgente con riferimento alla routine di interrupt.

```
// Routine di interrupt
void interrupt (void)
{
     // Interrupt on change
     if (INTCON.GPIF)
     {
        INTCON.GPIF = 0;
        // Acquisizione degli input
        chInput = GPIO & 0x20;

        // Se INPUT è premuto, si abilita il flag associato
        if (INPUT == PREMUTO)
        {
           flDebounceInput = 1;
        } else {
           flDebounceInput = 0;
        }
     }

     // Overflow TMR0
     if (INTCON.T0IF)
     {
        INTCON.T0IF = 0;
        TMR0 = 150;
        // Se il flag di antirimbalzo del tasto Input è abilitato
        if (flDebounceInput)
        {
           // si incrementa il contatore.
           chDebounceInput++;
           // Quando il contatore raggiunge il valore prestabilito
           if (chDebounceInput == 100)
           {
              chPulsInput = ON;
              flDebounceInput = 0;
              chDebounceInput = 0;
           }
        } else {
           chDebounceInput = 0;
           chPulsInput = OFF;
```

```
        }
    }
}
```

Il tutto funziona se le impostazioni di IOC e TMR0 sono state attivate. Ecco un estratto della routine di inizializzazione:

```
// Abiiltazione weak pull-up su GP5
     OPTION_REG.NOT_GPPU = 0;
     WPU = 0x20;

// Interrupt on change su GP5
     IOC = 0X20;

// TIMER 0: Prescaler 1:64
     OPTION_REG.PS0 = 1;
     OPTION_REG.PS1 = 0;
     OPTION_REG.PS2 = 1;
     OPTION_REG.T0CS = 0;
     TMR0 = 150;

// Inizializza le variabili: flag antirimbalzo pulsanti
     flDebounceInput = 0;

// Inizializza le variabili: contatori di antirimbalzo
     chDebounceInput = 0;
     chPulsInput = OFF;

// Attivazione degli interrupt
     INTCON.GPIE = 1;
     INTCON.INTE = 0;
     INTCON.T0IE = 1;
     INTCON.PEIE = 1;
     INTCON.CMIE = 0;
     INTCON.GIE  = 1;
```

Riferimenti

Come detto l'esempio si basa sul circuito elettrico del progetto MerryChristmas i cui file sono disponibile al link **http://www.electroportal.net/users/files/MerryChristmas.zip**

Datasheet PIC12F675: http://ww1.microchip.com/downloads/en/DeviceDoc/41190F.pdf

Data sheet PIC12F683: http://ww1.microchip.com/downloads/en/DeviceDoc/41211D_.pdf

MikroC PRO: http://www.mikroe.com/en/compilers/mikroc/pro/pic/

Pillole di microcontrollori PIC: http://www.inwaredizioni.it/pic2/

Estratto da "http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Paolino:antirimbalzo"