

Comparazione di tempi

Chiamiamo così la seconda forma di lavoro dei moduli CCP, che funziona in maniera simile al modo captur, però in senso inverso. Ogni volta che il valore dei registri del modulo CCP corrispondente coincide con il valore del TMR1, si genera un evento sui pin RC2 (per il modulo CCP1) o RC1 (per il modulo CCP2). Vediamo come utilizzarlo.

Schema di funzionamento

I registri che intervengono sono gli stessi del modo captur, con il risultato che le differenti forme di funzionamento dei moduli CCP sono fra di loro esclusive.

Il pin RC2 (o RC1) dovrà essere configurato come uscita, poiché su di esso si riceverà il segnale di quando i registri CCPRxH - CCPRxL coincidono con TMR1H - TMR1L. Nella

CCPXM3-0	MODULO LAVORO DEL MODULO IN COMPARAZIONE
0000	Modulo CCPx scollegato 1000 Modo Comparazione che attiva il pin RCy/CCPx al coincidere dei valori
1001	Modo Comparazione che disattiva (0) il pin RCy/CCPx al coincidere dei valori
1010	Modo Comparazione che genera un interrupt software (non influenza RCy/CCPx)
1011	Modo comparazione in cui si produce un'attivazione speciale diversa per ogni modulo.

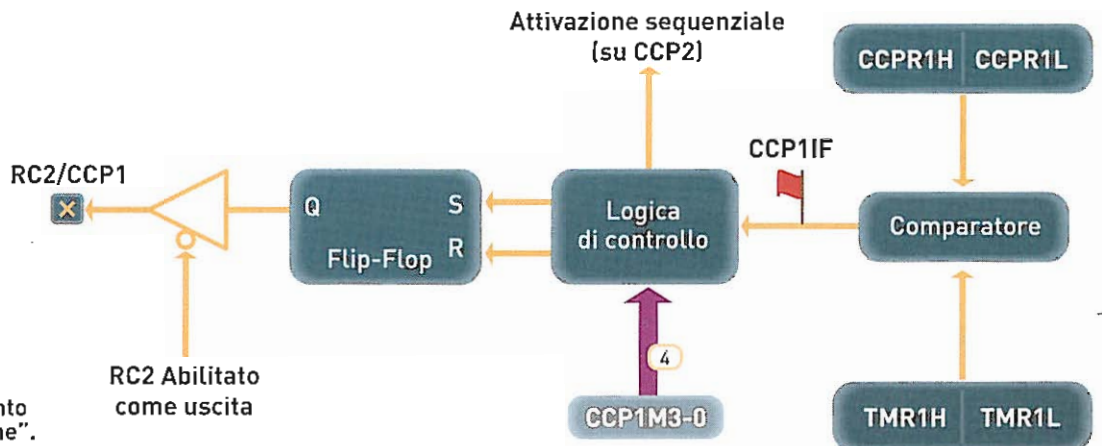
Modo lavoro del modulo in comparazione.

figura in basso è rappresentato lo schema del modulo CCP1.

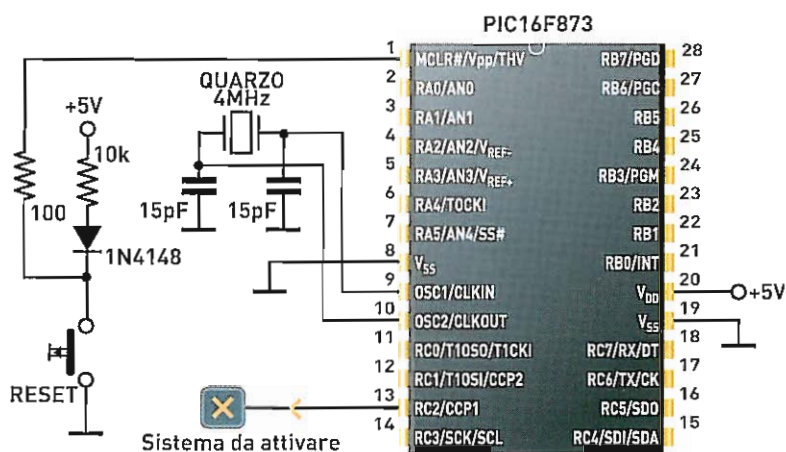
Gli eventi che possono succedere quando si verifica un'eguaglianza sono riportati nella tabella della figura in alto e dipendono dalla configurazione dei bit CCP1M3-0 o CCP2M3-0, che appartengono ai registri CCP1CON e CCP2CON, come già succedeva in modo captur.

RC2 (RC1) può passare al livello alto o basso a seconda dell'opzione, o non cambiare di valore e produrre solo un interrupt software. Questo interrupt si può produrre anche nei primi due casi.

L'opzione di "attivazione speciale" è l'unica differenza fra i moduli CCP1 e CCP2 in quanto a modo lavoro. Con questa scelta entrambi i moduli, quando si produce l'uguaglianza fra i registri CCPRx e il TMR1, producono un interrupt e tornano a impostare a 0 il TMR1, permettendo al processo di ripetersi e producendo in questo modo interrupt in forma periodica. Il modulo CCP2 inoltre, inizierà una conversione del convertitore A/D, sarà quindi possibile far eseguire delle conversioni automaticamente e in modo periodico. Anche se non si abilitano gli interrupt, i bit CCP1F



Schema di funzionamento del "modo comparazione".



Schema elettronico dell'esercizio proposto.

e CCP2IF segneranno le uguaglianze rilevate dal comparatore passando a valore 1.

Definizione del problema

La forma di funzionamento dei moduli CCP nel modo comparazione, si può confondere con il funzionamento normale dei temporizzatori, poiché entrambi controllano il termine di un determinato periodo. La differenza fondamentale è che i moduli CCP comunicano con l'esterno cambiando automaticamente il valore di una linea o producendo eventi in modo periodico.

Noi li utilizzeremo per attivare automaticamente un secondo sistema collegato al nostro impostando il pin corrispondente al livello 1.

Schema elettronico

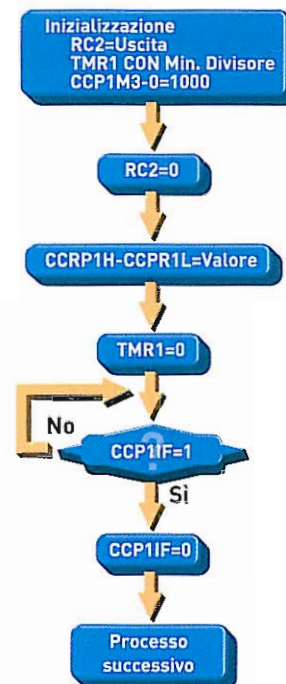
Non è necessario alcun elemento di ingresso e come

uscita utilizzeremo la linea del modulo CCP, che sarà la RC2(CCP1) o la RC1(CCP2). Nell'esempio abbiamo scelto la prima.

Organigramma e programma commentato

L'organigramma ci mostra i passi necessari che rispondono al nostro enunciato, RC2 dovrà essere configurato come uscita.

Il TMR1 è stato definito con il minimo divisore, anche se questo dipenderà dal valore che si vuole ottenere. In questo caso bisogna tenere presente che l'attivazione di RC2 è automatica e quindi dovremo servirci del valore del TMR1 cioè non potremo fare in modo che il pin si attivi dopo X volte che si è verificata un'eguaglianza. Con il valore "1000" selezioneremo il modo lavoro del modulo che ci interessa. Siccome vogliamo che RC2 cambi di livello, per prima cosa lo inizieremo a 0.



Organigramma dell'esercizio proposto.

In CCPR1H-CCPR1L dovremo impostare il valore di uguaglianza con il TMR1.

Una delle possibilità potrebbe essere far introdurre dall'utente, mediante interruttori, questo valore. Imposteremo il TMR1 a 0 per avere un punto noto di inizio, ed entreremo in un ciclo sino a quando non si attiverà il flag. In realtà questo non è necessario se il programma continua, e non si prevede di tornare a utilizzare il modulo in questo programma, però tale impostazione è stata fatta perché non è specificato quale sia il "processo successivo". Se questo processo fosse il termine del programma, prima di eseguirlo dovremo raggiungere il nostro obiettivo, e questo non si consegue sino a quando il valore del flag non passa a 1.