

Errata Corrige

DP

Prova la teoria

Il TMR0 come contatore

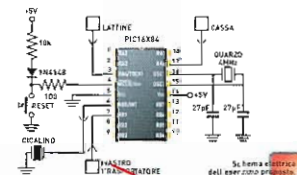
Fra i diversi temporizzatori, il TMR0 può lavorare in due modi diversi: come temporizzatore (per contare il tempo), o come contatore (per contare gli impulsi che arrivano dall'esterno). In un programma precedente abbiamo simulato il funzionamento di un allarme utilizzando le temporizzazioni. Tuttavia non possiamo ancora utilizzare le istruzioni temporizzatori necessarie per lavorare con il TMR0 direttamente, abbiamo utilizzato le istruzioni delay, che però, in quanto congegno a velocità costante, permettono di utilizzare il TMR0 in modalità di contatore di impulsi, anche questa volta utilizzando le istruzioni delay.

Definizione del problema

Si tratta di simulare il funzionamento di una libreria di contatore. Un nostro trasportatore lascia passare le lattine solo a compiere una corsa da 12 unità, poi si ferma e si aziona un cicalino per un secondo, al fine di avvertire l'operatore perché sostituisca la cassa con una nuova. La cassa attiverà un sensore al loro passaggio, quindi non è obbligatorio che l'intervallo tra le casse sia regolare.

Schema elettronico

La possibile implementazione nel passaggio delle casse rende



Schema elettronico dell'esercizio proposto.

È necessario l'utilizzo di un sensore, altrimenti potremmo suggerire un tempo di riempimento. Abbiamo già visto gli inconvenienti di contare gli elementi "a mano", ragion per cui sfrutteremo le opportunità che ci



Programma dell'esercizio proposto.

PROGETTO DI PROGRAMMI

DP 020

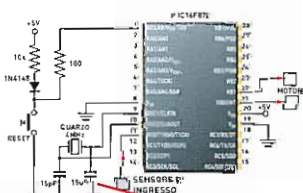
Prova la teoria

Il TMR1 come contatore

Così come il TMR0, anche il TMR1 può lavorare in modo contatore utilizzando gli impulsi applicati ai uno dei piedini del PIC. Questo tipo di funzionamento è conosciuto come "contatore asincrono", possibile non si conosce quando si verificano gli impulsi, in quanto non sono generati a intervalli costanti. In modo sincrono ma dipendono da fattori esterni, ad esempio sensori, e dall'utente stesso che sta utilizzando il sistema. Di seguito il caso che tratteremo nel prossimo capitolo: il modo di funzionamento è quello della figura per il TMR1, non è un modo contatore, e lo privilegiamo in un capitolo successivo.

Differenze con il contatore del TMR0

Oltre al fatto che il TMR0 è un temporizzatore di 8 bit mentre il TMR1 è un temporizzatore da 16,



Schema elettronico dell'esercizio proposto.

quando il TMR1 lavora in modo contatore e il TMR0 come temporizzatore asincrono, fra loro esiste una differenza: mentre il primo può essere configurato in modo che l'incremento sia abilitato con il fronte di salita,

Definizione del problema

È necessario compilare la capienza di un locale in epoche la porta rimanga aperta sino a che non si raggiunge un numero limite di persone, e si chiude quando questo limite viene raggiunto. Il problema non è un caso del tutto reale, dato che normalmente si è qualcuno distanzia il locale, si spingerebbe diminuire il conteggio e questo non è possibile con i temporizzatori

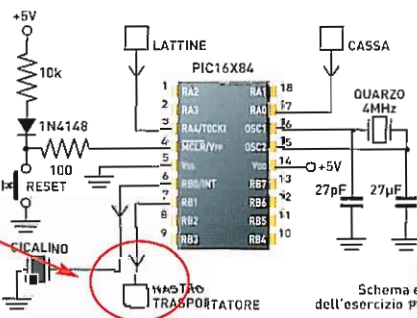
TMR0 (8 bit)	CONTATORE (fronte di salita/discesa)	
	TEMPORIZZATORE	
TMR1 (16 bit)	CONTATORE (fronte di salita)	SINCRONO
	TEMPORIZZATORE	ASINCRONO
TMR2 (16 bit)	TEMPORIZZATORE	

PROGETTO DI PROGRAMMI

DP 022

teoria

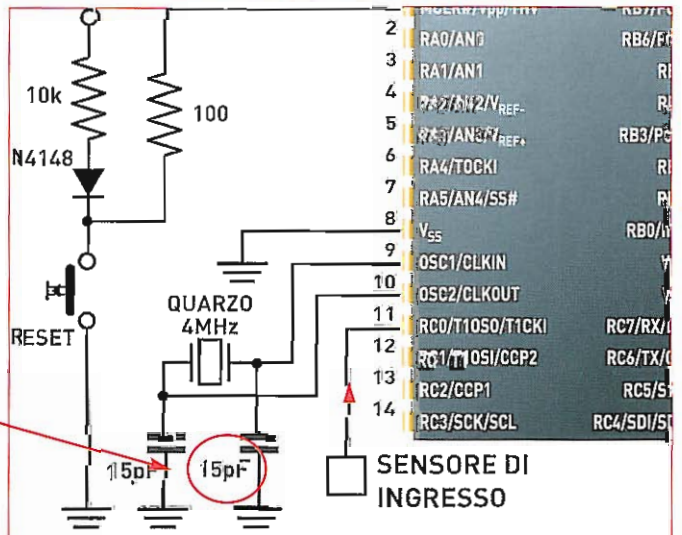
Il contatore



Schema elettronico dell'esercizio proposto.

necessario l'utilizzo di un sensore, altrimenti potremmo sopporre un tempo di riempimento. Abbiamo già visto gli inconvenienti di contare gli elementi "a mano", ragion per cui sfrutteremo le opportunità che ci

fornisse il contatore automatico del TMR0. Il suo unico svantaggio è che il piedino d'ingresso degli impulsi deve essere obbligatoriamente RA4, per cui se è necessario realizzare più di un conteggio simultaneo, solamente



Schema elettronico dell'esercizio proposto.

quando il TMR0 lavora in modo

con il fronte di dis



JP

Prova la teoria

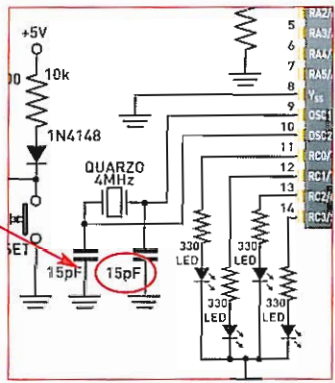
IL TMR1 come temporizzatore

Differenza fra i temporizzatori

Scheda elettronica

Definizione del programma

PROGETTO DI PROGRAMMA: DP 0211



Prova la teoria

Organigramma

Realizzazione del programma

PROGETTO DI PROGRAMMA: DP 0211

frequenza, 8 nel nostro esempio. Trattandosi di un temporizzatore a 16 bit il valore massimo che si potrà contare sarà di 216, che non è molto dato che un ciclo di un PIC a 4 MHz è di **1 μs**. Arrivando al valore massimo il timer tornerà a 0 e continuerà a contare, quindi se noi vogliamo avere valori maggiori dovremo incrementare un'altra variabile ad ogni "giro". Dopo aver messo in marcia il

CP

Basic per PIC

Invio dei dati

Ricezione dei dati

COME PROGRAMMARE: CP 032

Invio dei dati

L'istruzione per l'invio dei dati è BUSOUT. L'unico parametro di cui ha bisogno, posto fra parentesi, può essere un numero, una variabile o un'espressione. Il primo che si invia corrisponde all'indirizzo dello "slave" si vuole realizzare la comunicazione, e il bit me

Assembler per PIC

Operazioni multiple e complesse

Prima delle operazioni in MPLAB

COME PROGRAMMARE: CP 042

operazione e di somma, questo bit tornerà a 1 quando c'è riporto, e si tornerà a 0 se il risultato è contenuto in un registro da 8 bit o, in altre parole, se non c'è riporto. Se invece l'operazione è di sottrazione, questo bit funziona al rovescio: infatti se si vuole sottrarre $A < B$ e $A > B$, il bit sarà un "avanzo", e il bit s'imposta a 1, e, se al contrario, $A < B$, il bit andrà a 0 perché ci sarà una "anzanza". Il bit DC funziona come C però per i numeri BCD.

operazioni multiple

Assembler per PIC

Istruzioni di ingresso e uscita dei dati

Dichiarazione di variabili

COME PROGRAMMARE: CP 051

Basic per PIC

Istruzioni di ingresso e uscita dei dati

Basic per PIC

Creazione e gestione di tabelle

Definizione di una tabella di dati

COME PROGRAMMARE: CP 061

Assembler per PIC

Creazione e gestione di tabelle

LP

Basic per PIC

Istruzioni di cancellazione

Cambio di posizione dei bit di un registro

COME PROGRAMMARE: LP 011

Basic per PIC

Istruzioni generali

Per le istruzioni Assembler ve ne sono tre che non sono state classificate in alcun gruppo perché la loro funzione non ha similitudini con nessun'altra. Vediamo come si comportano con alcuni esempi.

Cambio di posizione dei bit di un registro

MNEMONICO: swapf

Esercizio con ultrasuoni (8)

LABORATORIO PRATICO: LP 017

Faremo un programma di controllo per gli ultrasuoni. Il codice sorgente è mostrato nell'immagine. Sul pin riceveremo un "0" logico ogni volta che un movimento davanti al robot, attivando i motori posteriori di Pattin nel senso di avanzamento del robot, il jumper della scheda dei sensori è nella stessa posizione dell'esercizio precedente con gli ultrasuoni, per fare in modo che la capsula di sia sempre attiva.

Errata Corrige

CM

I famolosi PIC16F87X La memoria EEPROM

memoria FLASH in quanto, dato che può arrivare ad avere una capacità di 8192 indirizzi, [8K] richiede un indirizzamento superiore agli 8 bit: quindi viene usato EEDRH per contenere la parte alta dell'indirizzo stesso. La stessa cosa succede con il contenuto dell'indirizzo, il quale essendo da 14 bit ha bisogno del registro EEDATAH per contenere i bit che non entrano in EEDATA. Ci sono i PIC16F87X che arrivano sino a 256 byte di EEPROM e l'indirizzo più alto è FF Hex, che utilizza

di bit, **EEPGD** —
Struttura interna del di controllo EECON1
scritture indesiderate verificarsi come con rumore elettrico inteso utilizzato il bit WREN

I famolosi PIC16F87X Moduli CCP in modo comparazione

segnate su di un pin del microcontroller dopo un certo periodo di tempo, controllato dal TMR1. Per fare questo si carica un valore da 16 bit sulla copia di registri CCPRH:CCPRL1. In seguito si pone in marcia il TMR1, il cui valore si incrementa ad ogni ciclo di istruzione, e si compara con ciò che è contenuto in CCPRH:CCPRL1. Quando entrambi i valori coincidono si genera un segnale sul piedino RC2/CCP1, che deve essere configurato come uscita. Nel disegno della figura è rappresentata schematicamente la funzione del modulo CCP1 come comparatore. I differenti segnali che si possono generare tramite il piedino RC2/CCP1

CCP1CON
CCP1M3 CCP1M2 CCP1M1
Esempio di programmazione
Programmare la parte più alta del registro CCP1 in modo comparazione

I famolosi PIC16F87X Moduli CCP in modo Modulazione di Ampiezza degli Impulsi (PWM)

dei TMR2, quando questi due valori coincidono, il TMR2 passa a 00 Hex. Il flip-flop si resetta e passa la sua uscita Q a zero, quando coincidono gli ingressi dell'altro comparatore che confronta CCPRH1 e TMR2. Cambiando i valori che si caricano in CCPRL1 (che successivamente si sposteranno su CPR1H per la comparazione) e in PR2, si varia l'intervallo di tempo in cui le uscite del flip-flop, e quindi il piedino RC2/CCP1, resteranno a 1 o a 0. Per lavorare con una precisione di

determina il periodo in PR2.
2°. Caricare il valore che corrisponde all'ampiezza dell'impulso (duty cycle) in CCPRH1:CCP1X:CCP1Y.
3°. Configurare il piedino RC2/CCP1 come uscita.
4°. Assegnare il valore del predivisor e attivare il TMR2 scrivendo il T2CON.
5°. Configurare il modulo CCP1 in modo PWM tramite il registro CCP1CON.

Altri modelli di PIC

uscite dei dispositivi interni del microcontroller. Vi sono alcuni piedini che supportano sino a quattro funzioni selezionabili tramite programma.
PIC12C671
PIC12C672
PIC12CE673
PIC18F010
PIC18F020

MMMA BASE
OPERAZIONI CON BIT
CF f, b: Cancella bit b su f
SF f, b: Imposta a 1 bit b su f
TFSC f, b: Sottrae b e salta se è 0
TFSS f, b: Sottrae b e salta se è 1
OPERAZIONI CON IMMEDIATI E DI CONTROLLO
LEEP : Passa al modo riposo
LRWDT : Cancella WDT
ETLW k : Ritorno e carica K su W
OPTION : Carica su OPTION W
RISF f : Configurazione linee I/O porta
ALL a : Chiama a subroutine a