Paolo Rognoni (Paolino)

LO HAI MAI REALIZZATO CON UN PIC? - IL CONTA MARCE

8 December 2009

Presentazione

Vorrei iniziare, qui su su **ElectroYou** (tempo permettendo), un **tutorial** dal nome '"**Lo hai mai realizzato con un PIC?**"'. Il mio intento è quello di mostrare come, con l'elettronica embedded a microcontrollore (e in particolare con i PIC Micro di Microchip Technology), si possa rispondere a molti quesiti e produrre soluzioni a problemi altrimenti proposte con altri metodi. Il primo esempio trae spunto da una richiesta che un utente del forum ha effettuato in **questo post**: realizzare un **contamarce per KART**.

Il contamarce

L'obiettivo di questo progetto è quello di realizzare un contatore Up/Down, con alcune specifiche ben chiare. Posto che **il progetto ha fini prettamente didattici**, dovendo realizzare un contatore per tenere traccia della marcia inserita si può pensare che:

- la marcia minima sia la prima;

- la marcia massima sia la settima;

- la visualizzazione avvenga su un solo display a sette segmenti;

- l'alimentazione provenga da una piccola batteria.

Con queste premesse, si può pensare di impiegare un PIC piuttosto "piccolo", come il PIC12F675 (per il datasheet si vada il paragrafo dei riferimenti) che fornisca ad un driver per display a sette segmenti, come ad esempio CD4511, una codifica per la rappresentazione del numero da visualizzare. È ragionevole pensare che un kart non disponga di un cambio di velocità con più di sette marce (non sono un esperto di kart, ma questo mi sembra un assunto abbastanza ragionevole) e che quindi i numeri da rappresentare vanno da 1 a 7.

Osservando la tabella della verità del CD4511 si nota come dei quattro bit di controllo ne bastino solo 3 per rappresentare i numeri da 1 a 7.

I tre segnali in ingresso che vengono connessi al PIC son pertanto A, B, C mentre D viene collegato direttamente a massa.

Truth Table

Inputs							Outputs							
LE	$\overline{BI}$	$\overline{LT}$	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	B
X	0	1	X	X	X	X	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	0	1	0	0	0	0	0	0	0	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X				*				*

X = Don't Care

*Depends upon the BCD code applied during the 0 to 1 transition of LE.

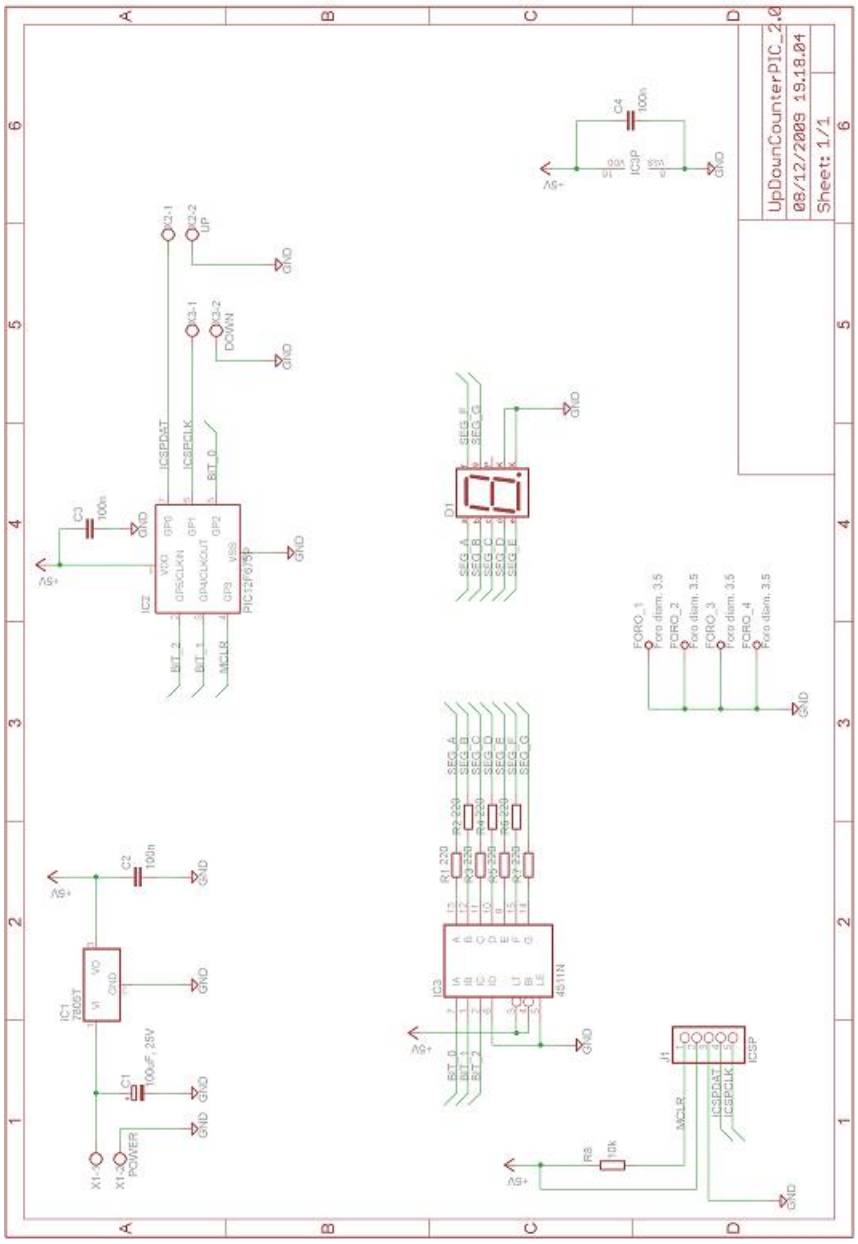
CD4511_TruthTable.JPG

Per evitare di appensantire il circuito, si possono sfruttare alcune caratteristiche del PIC e in particolare:

- impiego dell'oscillatore interno a 4MHz;
- utilizzo delle resistenze di pull-up interne.

Lo schema elettrico

Nella figura seguente è proposto lo schema elettrico (si veda il paragrafo "Progetto" per poter scaricare il file PDF). Il progetto si compone di uno stadio di alimentazione a 5V stabilizzati che fornisce tensione a tutto il circuito. Il PIC ha in ingresso i pulsanti UP e DOWN e fornisce al driver 4511 la codifica della numerazione da 1 a 7 mediante bit.



SchemaElettrico_GearCounter.JPG

I componenti utilizzati sono i seguenti:

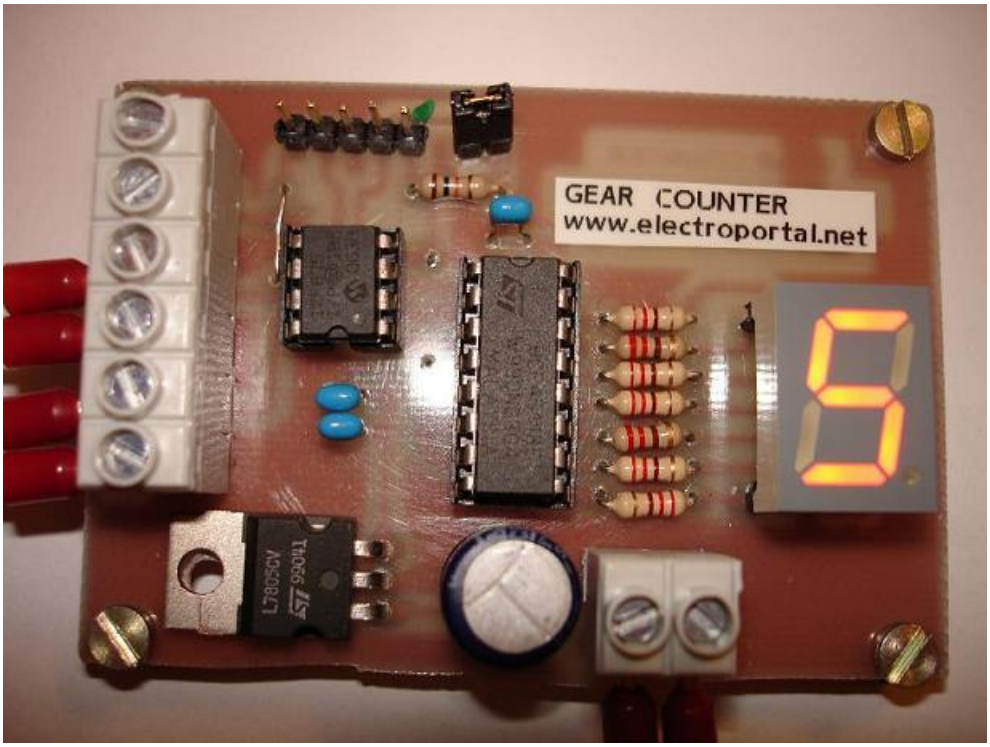
- C1 100uF, 25V
- C2 100n
- C3 100n
- C4 100n
- D1 Display 7 segmenti a catodo comune
- IC1 7805
- IC2 PIC12F675P

- IC3 CD4511N
- J1 Strip 5 poli
- R1 220
- R2 220
- R3 220
- R4 220
- R5 220
- R6 220
- R7 220
- R8 10k
- X1 Connettore a vite passo 5.08
- X2 Connettore a vite passo 5.08
- X3 Connettore a vite passo 5.08

Dallo schema, sviluppato con EAGLE, è stato realizzato con un circuito stampato. Nelle figure che seguono sono mostrati tanto il rendering effettuato al computer quanto le foto del prototipo (che presenta qualche piccola differenza realizzativa rispetto alla versione finale). La semplicità del progetto non necessita in modo specifico di un circuito stampato. Per chi volesse, il connettore J1 per la programmazione in-circuit ICSP, può essere omesso qualora si utilizzi un programmatore diverso da ICD2 o ICD3.

GearCounter.JPG

L'alimentazione va fornita al morsetto X1, mentre ai morsetti X2 e X3 vanno connessi rispettivamente i pulsanti (di tipo N.A.) per l'incremento (UP) ed il decremento (DOWN) del conteggio.



GearCounterPrototipo.JPG

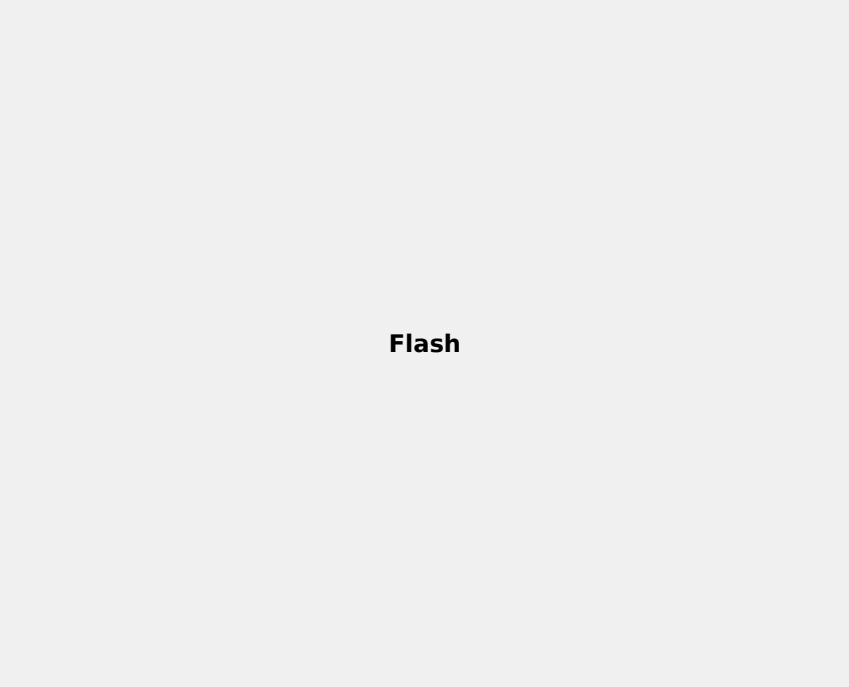
Firmware

Il progetto si compone anche del firmware, scritto in **linguaggio, C con MikroC PRO** (si veda il paragrafo dei riferimenti). Il codice gestisce l'interrupt tanto sull'evento Interrupt-On-Change quanto sull'overflow del TIMER0. Il primo viene impiegato per acquisire i segnali di INPUT mentre il secondo gestisce il fenomeno del rimbalzo del contatto garantendo che il sistema sia immune da incertezze. La routine di configurazione si preoccupa di impostare tanto il tristate in modo opportuno, cioè aderente a quanto stabilito con lo schema elettrico, quanto l'abilitazione dei pull-up interni, la risposta all'interrupt IOC e su overflow di TRM0. Nel main() il codice non gestisce più la pressione del pulsante bensì il segnale che viene generato dalle routine di interrupt qualora il pulsante premuto abbia concluso il proprio rimbalzo.

Come funziona

Una volta alimentato, il software esegue un test, contando da 0 a 7 in modo rapido; questo permette di verificare che tutti i numeri vengono correttamente

rappresentati; successivamente il firmware visualizza il numero minimo (1, in questo caso) e resta in attesa della pressione di uno dei pulsanti. La pressione di UP comporta l'aumento dl conteggio della marcia, mentre DOWN è tale da eseguire un decremento; il firmware controlla che il conteggio non sia mai inferiore al numero minimo e maggiore al massimo delle marce disponibili. Ora, se è realistico pensare che la marcia minima sia la prima (e quindi venga visualizzato il numero 1) possiamo pensare a cambi di velocità con 4, 5, 6 o 7 marce. Il filmato che segue dovrebbe fugare ogni dubbio:



Tips 'n Tricks

Si faccia attenzione ad un aspetto molto importante. Benché PIC12F675 permetta l'adozione dell'oscillatore interno al posto di quello esterno, questa selezione esclude in modo categorico l'impiego del MASTER CLEAR interno! Questo aspetto è citato nel datasheet, pertanto è necessario porre una resistenza di pull-up esterna per MCLR ed impostare il compilatore in modo opportuno.

bit 5 **MCLRE**: GP3/MCLR pin function select(5)
1 = GP3/MCLR pin function is MCLR
0 = GP3/MCLR pin function is digital I/O, MCLR internally tied to VDD

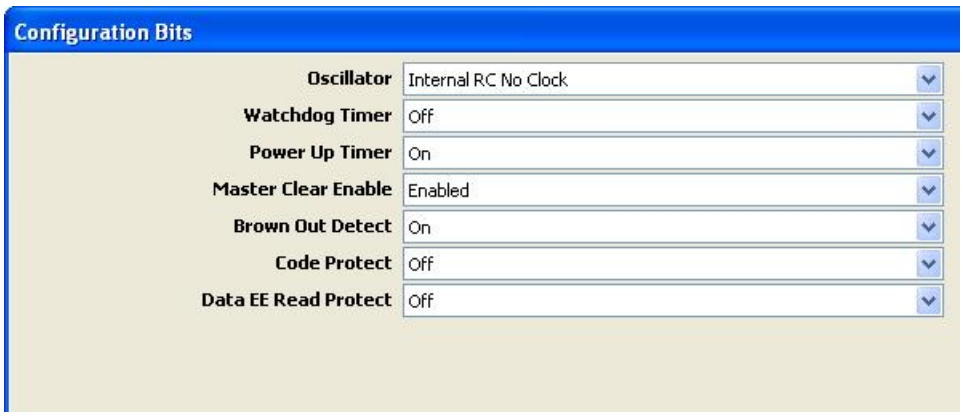
bit 4 **PWRTE**: Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 2-0 **FOSC2:FOSC0**: Oscillator Selection bits
111 = RC oscillator: CLKOUT function on GP4/OSC2/CLKOUT pin, RC on GP5/OSC1/CLKIN
110 = RC oscillator: I/O function on GP4/OSC2/CLKOUT pin, RC on GP5/OSC1/CLKIN
101 = INTOSC oscillator: CLKOUT function on GP4/OSC2/CLKOUT pin, I/O function on GP5/OSC1/CLKIN
100 = INTOSC oscillator: I/O function on GP4/OSC2/CLKOUT pin, I/O function on GP5/OSC1/CLKIN
011 = EC: I/O function on GP4/OSC2/CLKOUT pin, CLKIN on GP5/OSC1/CLKIN
010 = HS oscillator: High speed crystal/resonator on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN
001 = XT oscillator: Crystal/resonator on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN
000 = LP oscillator: Low power crystal on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN

Note 1: The Bandgap Calibration bits are factory programmed and must be read and saved prior to erasing the device as specified in the PIC12F629/675 Programming Specification. These bits are reflected in an export of the configuration word. Microchip Development Tools maintain all calibration bits to factory settings.
2: The entire data EEPROM will be erased when the code protection is turned off.
3: The entire program memory will be erased, including OSCCAL value, when the code protection is turned off.
4: Enabling Brown-out Detect does not automatically enable Power-up Timer.
5: When MCLR is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

DatasheetConfBIts.JPG



ConfBits.JPG

Per concludere

Questo non è che un esempio di come sia possibile realizzare un semplice contatore UP/DOWN con un microcontrollore PIC. Il progetto, con taglio prettamente didattico, lungi dal voler essere un'applicazione di tipo automotive. Per poter applicare questo contatore ad un kart è necessario prendere altre precauzioni ed attenzioni che esulano però dai motivi che stanno alla base di questo esercizio.

Progetto

Il progetto completo scritto per MikroC PRO ed il file PDF dello schema elettrico, sono disponibili al seguente link: http://www.electroportal.net/users/files/UpDownCounter.zip

Il progetto si **compila anche con la versione free** di MikroC PRO.

Riferimenti

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

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

Datasheet CD4511: http://www.datasheetcatalog.org/datasheets/150/109579_DS.pdf

CAD EAGLE: http://www.cadsoftusa.com

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