

Programmazione di Digital Signal Controller (dsPIC33F):

Si descriva la configurazione degli SFR e la programmazione della Interrupt Service Routine di un dsPIC33F per l'applicazione con le seguenti caratteristiche tecniche:

1. L'oscillatore del dsPIC33F è stato configurato (a priori) per avere frequenza di istruzione 30 MIPS ($F_{cy} = 30 \text{ MHz}$).
2. Si vuole acquisire la misura di una cella di carico serie FC22 della Measurement Specialties mostrata in figura:



avente le seguenti caratteristiche:

- Tensione di alimentazione: 3,3 V
 - Carico misurabile: da 0 a 100 libbre (1 lbs = 0,453592 Kg)
 - Uscita analogica con range tra 10% e 90% (rispettivamente corrispondenti a 0 e 100 libbre) della tensione di alimentazione.
3. L'uscita del sensore è collegata al segnale analogico AN1, del quale è richiesta la conversione in digitale a 12 bit con tempo di campionamento di 10 ms.
 4. Si deve effettuare la messa in scala della misura dalla cella di carico utilizzando solamente aritmetica intera (Fixed-Point).
 5. L'alimentazione del dsPIC33F è a 3,3 V e non sono previste alternative all'uso di tale tensione come V_{REFH} per il convertitore A/D (e V_{REFL} a 0 V).

Suggerimenti:

- Utilizzare un timer a 16 bit (e.g. Timer3 o Timer5) con un opportuno valore di prescaler oppure una concatenazione di timer a 32 bit (e.g. Timer2+Timer3 o Timer4+Timer5) come Sample Clock Source Select per l'ADC.
- Determinare l'operazione di messa in scala dalla caratteristica ingresso/uscita del sensore, al fine di ottenere come risultato il carico in Kg (N.B. non in libbre!) sulla cella, rappresentato in formato Fixed-Point Q15 (i.e. valori reali moltiplicati per 2^{15} e arrotondati all'intero), ma con una variabile intera a 32 bit con segno.

RISPOSTA:

Valori di configurazione degli SFR:

```
////ADC CONFIG
//Pin RA1 (AN1) Tristate as INPUT
TRISAbits.TRISA1 = 1;
// Config analog pins
// all digital..
AD1PCFGL = 0xFFFF;
//with one exception (AN1)
AD1PCFGLbits.PCFG1 = 0;

// Initialize MUXA Input Selection
AD1CHS0bits.CH0SA = 1; // Select AN3 for CH0 +ve input
AD1CHS0bits.CH0NA = 0; // Select VREF- for CH0 -ve input

// 12 Bit ADC
AD1CON1bits.AD12B = 1;

// ADC trigger (Sample Source Select bits)
AD1CON1bits.SSRC = 2; //Timer 3

// ENABLE Auto-sample
AD1CON1bits.ASAM = 1;

// Power ON converter
AD1CON1bits.ADON = 1;

// ADC Interrupt Enabled (flag is set when ADC is done!)
IEC0bits.AD1IE = 1;

//// TIMER3 config (16 bit, with prescaler 1:8)
T2CONbits.T32 = 0; // 16 bit mode (NOTE: 16/32 bit mode is in T2CON!)
T3CONbits.TON = 0; // ALL other details are in T3CON
T3CONbits.TCS = 0; // Internal clock source
T3CONbits.TGATE = 0; // NO gate control
T2CONbits.TCKPS = 1; // Prescaler 1:8
TMR3 = 0;
// 10 ms period @ 30 MIPS / 8 (prescaler): 37.500
PR3 = 37500;
// RESET FLAG, BUT DO NOT ENABLE INTERRUPT (using only ADC interrupt)
IFS0bits.T3IF = 0;
IEC0bits.T3IE = 0;
T3CONbits.TON = 1; //START Timer
```

```

#include <stdint.h>

int32_t Load_Q15;

void __attribute__((interrupt,no_auto_psv)) _ADC1Interrupt(void)
{
// READ ADC CONVERSION RESULT AND SCALE LOAD CELL MEASURE

// Kadc = 2^12 / 3,3 [V] = 4096 / 3,3 [V] = 1241,21
//
// ADC Value = Vout * Kadc
//
// Vout of the sensor is 3,3 [V] * 10% at 0 kg
//                    and 3,3 [V] * 90% at 45,3592 kg
// Ksens = (3,3 * 90% - 3,3 * 10%) [V] / 45,3592 [kg] = 0,0582
//
// Load = (ADC Value - 3,3 [V]*10% * Kadc) * [ 1 / ( Kadc * Ksens) ]
// OR EQUIVALENTLY
// Load = ADC Value / ( Kadc * Ksens) - 3,3 [V]*10% / Ksens
// NOTE: this second choice is better, since 3,3 [V]*10% * Kadc = 409,6
//       that must be necessarily ROUNDED to 410, introducing a large error!
//
// [ 1 / ( Kadc * Ksens) ] = 0,013843 ==> Q15 (i.e. * 2^15) ==> 453

Load_Q15 = (int32_t) ADC1BUF0 * 453 - 185798;

// NOTE: 185798 is the 10% offset in the same Q15 representation of the
// load: 3,3 [V] * 10% / Ksens * 2^15 = 185798

// DON'T FORGET TO RESET THE INTERRUPT FLAG BEFORE EXIT!!!
IFS0bits.AD1IF = 0;

} // END ADC1Interrupt ISR

```