

## 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 32 MIPS ( $F_{cy} = 32 \text{ MHz}$ ).
2. Si vuole acquisire la misura dei due segnali di uscita Xout e Yout di un accelerometro biassiale ADXL203 della Analog Devices. Il sensore è alimentato a 3,3 V e la sua caratteristica di uscita è raziometrica (i.e. dipende dalla tensione di alimentazione), pertanto può essere definita come segue (per Xout e Yout):
  - Range di accelerazione misurabile: +/- 1,7 g
  - Zero g output (offset): 1,65 V
  - Sensitività di output: 0,66 V/g
3. Le due uscite Xout e Yout del sensore sono collegate rispettivamente agli input analogici AN0 ed AN3. Si desidera effettuare il campionamento simultaneo dei due input (pertanto con risoluzione della conversione necessariamente a 10 bit) ed in modo che il tempo di campionamento sia fissato a 5 ms.  
**NOTA BENE:** Si suppone di utilizzare un dsPIC33F non dotato di modulo DMA, nel quale è consigliabile impostare il convertitore in modo che generi un interrupt ogni due cicli di conversione (v. campo SMPI in AD1CON2). Si noti inoltre che con questa modalità i risultati della conversione dei segnali assegnati ai sample/hold CH0 e CH1 sono memorizzati rispettivamente nei registri ADC1BUF0 e ADC1BUF1.
4. Si deve effettuare la messa in scala della misura delle due accelerazioni (esprese in g) 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 il Timer3 a 16 bit con un opportuno valore di prescaler oppure la concatenazione Timer2/Timer3 a 32 bit 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 le accelerazioni in g, 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.
- Si noti che per il campionamento simultaneo degli input AN0 e AN3 esistono due (e solo due) configurazioni possibili per i sample/hold CH0 e CH1. Si noti inoltre che il modulo prevede una configurazione "alternata" Sample A / Sample B di tali canali che può essere scambiata automaticamente. Tale funzionalità non è richiesta e la configurazione dei canali CH0 (tramite registro AD1CHS0) e CH1 (tramite registro AD1CHS123) può essere impostata per il solo Sample A.

## RISPOSTA:

Valori di configurazione degli SFR:

```
////ADC CONFIG
//Pins RA0 (AN0) and RB1 (AN3) Tristate as INPUTs
TRISAbits.TRISA0 = 1;
TRISBbits.TRISB1 = 1;
// Config AN0 and AN3 as analog pins
AD1PCFGLbits.PCFG0 = 0;
AD1PCFGLbits.PCFG3 = 0;

// Initialize MUXA Input Selection for CH0..
AD1CHS0bits.CH0SA = 0; // Select AN0 for CH0 +ve input
AD1CHS0bits.CH0NA = 0; // Select VREF- for CH0 -ve input
// .. and CH1
AD1CHS123bits.CH123SA = 1; // Select AN3-4-5 for CH1-2-3 +ve inputs
// (AN4-5 and CH2-3 are actually not used)
AD1CHS123bits.CH123NA = 0; // Select VREF- for CH0 -ve input

/* OTHER VALID POSSIBILITY FOR MUXA CONFIGURATION:
AD1CHS0bits.CH0SA = 3;
..
AD1CH123bits.CH123SA = 0; // AN0-1-2 on CH1-2-3 (AN1-2/CH2-3 unused)
*/
// 10 Bit ADC
AD1CON1bits.AD12B = 0;
// Integer format
AD1CON1bits.FORM = 0;
// ADC trigger (Sample Source Select bits)
AD1CON1bits.SSRC = 2; //Timer 3
// SIMULTANEOUS SAMPLING
AD1CON1bits.SIMSAM = 1;
// ENABLE Auto-sample
AD1CON1bits.ASAM = 1;

// VOLTAGE REFERENCES: Avdd (3,3V) / Avss (0v)
AD1CON2bits.VCFG = 0;
// CONVERT CH0 AND CH1
AD1CON2bits.CHPS = 1;
// Interrupt every 2 conversions
AD1CON2bits.SMPI = 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
T3CONbits.TCKPS = 1; // Prescaler 1:8
TMR3 = 0;
// 5 ms period @ 32 MIPS / 8 (prescaler): 20.000
PR3 = 20000;
// 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 AccX_Q15, AccY_Q15;

void __attribute__((interrupt,no_auto_psv)) _ADC1Interrupt(void)
{
// READ ADC CONVERSION RESULTS AND SCALE ACCELERATIONS

// Kadc = 2^10 / 3,3 [V] = 1024 / 3,3 [V] = 310,3
//
// ADC Value = Vout * Kadc
//
// Vout of the sensor is AccX/Y * 0,66 [V/g] + 1,65 [V]
//
// Ksens = 0,66 [V/g]
//
// AccX/Y = (ADC Value - 1,65 [V]*Kadc) * [ 1 / (Kadc*Ksens) ]
//
// Offset = 1,65 [V] * Kadc = 512 (half scale!)
//
// [ 1 / ( Kadc*Ksens) ] = 0,0048828215 ==> Q15 (i.e. * 2^15) ==> 160

AccX_Q15 = (int32_t) (ADC1BUF0 - 512) * 160;
AccY_Q15 = (int32_t) (ADC1BUF1 - 512) * 160;

// NOTE: WITH THE OTHER VALID MUXA CONFIGURATION:
// AccX_Q15 is related to ADC1BUF1 and AccY_Q15 to ADC1BUF0!!

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

} // END ADC1Interrupt ISR

```