Programmazione di Digital Signal Controller (dsPIC33F):

Si descriva la configurazione degli SFR e la programmazione della Interrupt Service Routine di un dsPIC33FJ128MC802 per l'applicazione con le seguenti caratteristiche tecniche:

1. L'oscillatore del dsPIC33F è stato configurato (a priori) per avere frequenza di istruzione 20 MIPS (Fcy = 20 MHz).

2. Si prevede di attivare un elettromagnete tramite un convertitore di potenza comandato dal pin PWM2H1 del modulo Motor Control PWM2 (McPWM2). La frequenza della modulazione PWM deve essere di 2 KHz ed il timer del modulo McPWM2 deve essere impostato in Up/Down Counting Mode:

14-2: PWM Period Calculation in Up/Down Counting Modes (PTMOD = 10 or 11)

$$PxTPER = \frac{FCY}{FPWM \times (PxTMR\ Prescaler) \times 2} - 1$$

3. Per il controllo dell'elettromagnete si deve acquisire la misura del segnale di uscita di un sensore di campo magnetico ad effetto Hall A1318 di Allegro Microsystems, alimentato a 3,3 V e la cui caratteristica di uscita è definita come segue:
   - Quiescent Voltage Output (tensione di uscita in assenza di campo magentico): 1,65 V
   - Sensitività di output: : 2,5 mV/Gauss

4. Il sensore di campo magnetico è collegato all'input analogico AN3, da convertire con risoluzione 10 bit ed in modo che il campionamento sia sincronizzato con l'istante centrale del periodo della modulazione PWM
   NOTA BENE: vedere SSRC, Sample Source Clock Select, per ADC e P2SECMP, Special Event Compare SFR per McPWM2 e ricordare che il timer va configurato in Up/Down Counting Mode.

5. Si deve effettuare la messa in scala della misura di campo magnetico (espressa in Gauss) utilizzando solamente aritmetica intera (Fixed-Point).

6. 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:

- Determinare l'operazione di messa in scala dalla caratteristica ingresso/uscita del sensore, al fine di ottenere come risultato il campo in Gauss 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 RB1 (AN3) Tristate as INPUT
    TRISBbits.TRISB1 = 1;
    // Config analog pins all digital..
    AD1PCFGL = 0xFFFF;
    //with one exception (AN3)
    AD1PCFGLbits.PCFG3 = 0;

    // Initialize MUXA Input Selection
    AD1CHS0bits.CH0SA = 3; // Select AN3 for CH0 +ve input
    AD1CHS0bits.CH0NA = 0; // Select VREF- for CH0 -ve input

    // Set AVdd/AVss ADVREF+/ADREF-
    AD1CON2bits.VCFG = 0;

    // 10 Bit ADC
    AD1CON1bits.AD12B = 0;

    // ADC trigger (Sample Source Select bits)
    AD1CON1bits.SSRC = 5; //McPWM2

    // 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;

//// McPWM2 config
    // Timebase OFF (turned on later), no pre/post-scaler, free-running
    P2TCON = 0x0000;
    P2TPER = 4999; // Period register for 2 KHz with Fcy = 20 MHz

    P2TCONbits.PTMOD = 2 // Continuous Up/Down Mode

    PWM2CON1bits.PMOD1 = 1; // Independent PWM pairs (PWM2H1/L1)
    PWM2CON1bits.PEN1H = 1; // PWM2H1 as PWM output

    P2SECMPbits.SEVTDIR = 0; // Special event on up-counting
    P2SECMPbits.SEVTCMP = 4999; // Special event in the middle of PWM
                                //period with an UP/DOWN Counting Mode!!
    P2DC1 = 0; // Zero-duty on PWM2L1

    P2TCONbits.PTEN = 1; // enable the PWM generator
```

```
#include <stdint.h>

int32_t MagF_Q15;

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

// Kadc = 2^10 / 3,3 [V] = 1024 / 3,3 [V] = 310,3
//
// ADC Value = Vout * Kadc
//
// Vout of the sensor is MagField * 0,0025 [V/Gauss] + 1,65 [V]
//
// Ksens = 0,0025 [V/Gauss]
//
// MagField = (ADC Value – 1,65 [V]*Kadc) * [ 1 / (Kadc*Ksens) ]
//
// Offset = 1,65 [V] * Kadc = 512 (half scale!)
//
// [ 1 / ( Kadc*Ksens) ] = 1,289075089 ==> Q15 (i.e. * 2^15) ==> 42240

MagF_Q15 = (int32_t) (ADC1BUF0 – 512) * 42240;

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

} // END ADC1Interrupt ISR
```