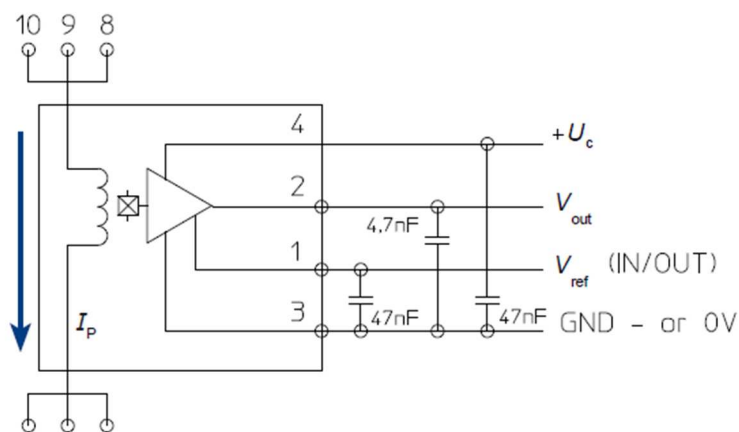


## Programmazione di Digital Signal Controller (dsPIC33F):

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. Acquisizione del segnale analogico AN3 con conversione in digitale a 10 bit
2. Il tempo di campionamento del segnale deve essere configurato nel modo più preciso possibile e pari a 4 ms.
3. L'oscillatore del dsPIC33F è configurato per avere frequenza di istruzione 25 MIPS ( $F_{cy} = 25 \text{ MHz}$ ).
4. L'alimentazione del dsPIC33F è a 3,3 V.
5. Il dispositivo collegato al segnale AN3 è un sensore di corrente ad effetto Hall (LEM HSLR 10-P) caratterizzato dal seguente schema logico:



e dalla seguente caratteristica ingresso/uscita:

$$V_{out} = I_p \times 0,046 + V_{ref}$$

Nella quale  $I_p$  è la corrente da misurare (in Ampere) e  $V_{ref}$  è fissata a 1,65 V.

6. Si deve effettuare la messa in scala della misura di corrente utilizzando solamente aritmetica intera (Fixed-Point).

### **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 la corrente in A rappresentata 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

// 10 Bit ADC
AD1CON1bits.AD12B = 0;

// 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;
// 4 ms period @ 25 MIPS / 8 (prescaler): 12.500
PR3 = 12500;
// 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 Current_Q15;

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

// Kadc = 1024 / 3,3 [V]
// ADC Value = Vout * Kadc
// Ksens = 0,046
// Current = ( ADC Value - 1,65[V] * Kadc) * [ 1 / ( Kadc * Ksens) ]
// [ 1 / ( Kadc * Ksens) ] = 0,0700577 ==> Q15 (i.e. * 2^15) ==> 2295
Current_Q15 = ((int32_t) ADC1BUF0 - 512 ) * 2295;

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

}

```