Índice

Resumen Sistemas Empotrados 1

Mejor configuración de un timer B para una ticks personalizados

El msp430 cuenta con registros de módulo de 16 bits para módulo, hasta un valor de $2^{16} - 1 = 65.535$.

A su vez cuenta con dos prescalers, el básico que puede hacer prescaler de **1, 2, 4 y 8**. Y una extensión del prescaler que permite un **postprescalado de 1, 2, 3, 4, 5, 6, 7 y 8**.

Con estas combinaciones podemos conseguir el valor del tick desesado.

1. Hay que realizar una tabla empezando por las combinaciones de prescaler más grandes que se puedan formar ya que, cuanto mayor sea, menos consumo producirá.
2. Para cada prescaler seleccionado, se debe calcular el tiempo de tick y el valor que contendría con ese tiempo de tick.
3. Se calcula el tiempo resultante y ese sería el tiempo entre interrupciones.

PS	T_{CNT}	MOD	T
Valor total prescaler	Tiempo de tick conseguido	Valor del registro de módulo	Tiempo resultante entre interrupciones.

Ecuación en diferencias a partir de función de transferencia

Se parte de la función de transferencia del controlador digital, dada por:

$$\frac{U(z)}{E(z)} = \frac{55.55(1 - 0.9901z^{-1})}{1 - z^{-1}} \tag{1}$$

Multiplicando ambos lados por el denominador (1 - z^{-1}) para despejar (U(z)):

$$U(z)(1 - z^{-1}) = 55.55(1 - 0.9901z^{-1})E(z) \tag{2}$$

Desarrollando ambos lados:

$$U(z) - U(z)z^{-1} = 55.55E(z) - 55.55 \cdot 0.9901E(z)z^{-1} \tag{3}$$

Aplicando la transformada inversa (Z^{-1}), se obtiene la ecuación en diferencias en el dominio del tiempo discreto:

$$u[k] - u[k-1] = 55.55\, e[k] - 54.947\, e[k-1] \tag{4}$$

Finalmente, despejando (u[k]):

$$u[k] = u[k-1] + 55.55\, e[k] - 54.947\, e[k-1] \tag{5}$$

Esta es la ecuación en diferencias que describe el comportamiento del controlador en el dominio temporal discreto, con un periodo de muestreo de 100 ms.

Representación en coma fija de parámetros con palabra de 2 bytes

Se dispone de una CPU que trabaja en coma fija con palabras de 2 bytes (16 bits). Se desea representar los números reales `2.4567`, `-1.3654` y `0.456` utilizando una misma representación **sin sesgo** y con **ponderación de bit unitaria**, de manera que se **maximice la precisión**.

1. Elección del formato de representación

El formato seleccionado es el de coma fija **sin sesgo**, utilizando el esquema `Qm.n`, donde:

- `m`: número de bits para la parte entera (sin contar el bit de signo),
- `n`: número de bits para la parte fraccional,
- `m + n + 1 = 16` (1 bit reservado para el signo).

Los valores a representar están en el rango:

```
min = -1.3654
max =  2.4567
```

Se requiere como mínimo representar valores desde `-2` hasta al menos `2.5`, por lo que se necesitan 3 bits para la parte entera (incluyendo el signo). Esto implica:

```
I = 3  -->  F = 13
```

Por tanto, se selecciona el formato **Q2.13**:

- 1 bit de signo
- 2 bits para la parte entera
- 13 bits para la parte fraccional

2. Precisión de la representación

La resolución mínima que se puede representar es:

```
delta = 2^-13 = 0.00012207
```

3. Representación y error de cada número

Para 2.4567

```
Código entero       = round(2.4567 / 2^-13) = round(20125.28) = 20125
Valor representado  = 20125 * 2^-13 = 2.4566650
Error absoluto      = |2.4566650 - 2.4567| = 0.00003496
```

Para -1.3654

```
Código entero       = round(-1.3654 / 2^-13) = -round(11185.35) = -11185
Valor representado  = -11185 * 2^-13 = -1.365356
Error absoluto      = |-1.365356 + 1.3654| = ...
```

Para 0.456

```
Código entero       = round(0.456 / 2^-13) = round(3735.552) = 3736
Valor representado  = 3736 * 2^-13 = 0.4560546
Error absoluto      = |0.4560546 - 0.456| = 0.0000546874
```

Se concluye que el formato `Q2.13` permite representar todos los valores deseados con una precisión de hasta `2^-13 = 0.00012207`, lo cual garantiza errores absolutos menores a dicha resolución.