

CALIBRACION POR EL METODO DE DOS PUNTOS

Ing. Rafael Oliva

1. Introducción

El procesamiento de los datos obtenidos por un sistema con convertidor Analógico/Digital (A/D) es muy dependiente de la aplicación, pero inevitablemente requiere alguna forma de calibración manual o automática. En general, la calibración por el *método de dos puntos* proporciona una manera de minimizar los errores promedio de lectura del sistema. Supondremos aquí una aplicación típica con un convertidor A/D de n bits, y que la computadora puede hacer uso de números de punto flotante. Además, supondremos una estructura de conversión de datos [Ref.1] como la dada en la **Figura 1**, y la expresión U.I. será utilizada para magnitudes físicas en Unidades de Ingeniería (ej.: Temperatura en °C, presión en hPa, etc.). El circuito "front end" o de entrada incluye un sensor con una cierta relación lineal de conversión G_T (V/U.I.), un amplificador con una ganancia A_V , un sumador de tensión de bias V_B , un filtro, un multiplexor ideal y el conversor A/D de n bits y una tensión de referencia FSV.

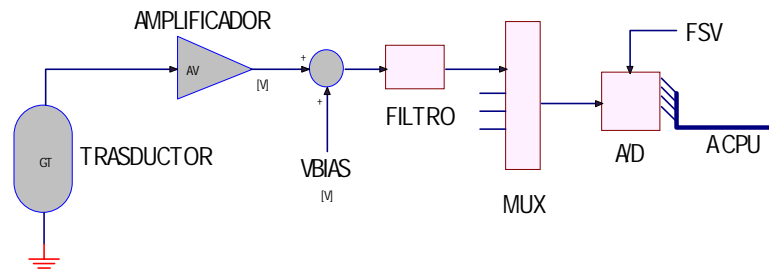


Figura 1 - Esquema de Conversión A/D típica

La ecuación general para la conversión del valor numérico (cuenta) ADC_{counts} a la salida del conversor A/D en Unidades de Ingeniería [Ref.1] es:

$$U.I. = \left(\frac{ADC_{counts} * FSV}{2^n - 1} - V_{bias} \right) * \frac{1}{G_T * A_V} \quad (eq.0)$$

2. Implementación del Método

A partir de la (eq.0), se pueden agrupar convenientemente los valores “por defecto” en dos coeficientes C_{off} y G_{conv} , y definir dos nuevos coeficientes de corrección: a) la corrección en cuentas C_{cal} , con un valor inicial 0, y b) la corrección en ganancia G_{cal} , con un valor 1.0 inicial, según se muestra en la ecuación siguiente:

$$U.I. = (ADC_{counts} + C_{offset} + C_{cal}) * G_{conv} * G_{cal} \quad (eq.1)$$

El método de los dos puntos con sensores de característica lineal involucra comparar dos mediciones de precisión conocida en puntos distintos utilizando la (eq.1), y despejar de las ecuaciones los valores de G_{cal} , C_{cal} . Supongamos que tenemos dos mediciones $U.I._1$, $U.I._2$ y los correspondientes valores de cuenta proporcionados por el A/D, C_1 y C_2 . Entonces, se cumple:

$$U.I._1 = (C_1 + C_{offset} + C_{cal}) * G_{conv} * G_{cal} \quad (eq.2)$$

$$U.I._2 = (C_2 + C_{offset} + C_{cal}) * G_{conv} * G_{cal} \quad (eq.3)$$

Restando ambas ecuaciones, se cancela el valor de C_{cal} , y es posible despejar sucesivamente los valores de G_{cal} y C_{cal} como sigue:

$$G_{cal} = \frac{1}{G_{conv}} \left[\frac{U.I._1 - U.I._2}{C_1 - C_2} \right] \quad (eq.4)$$

$$C_{cal} = \frac{U.I._1}{G_{conv} * G_{cal}} - C_1 - C_{offset} \quad (eq.5)$$

La implementación de esta rutina se muestra en forma de diagrama de flujo en las figuras del Anexo I. Básicamente, lo que se hace es solicitarle al usuario es que realice dos mediciones sucesivas, y que ingrese los valores conocidos $U.I._1$ y $U.I._2$, tomados con un instrumento de precisión conocida. En cada medición, el programa registra los valores correspondientes de C_1 y C_2 . Si la secuencia es correcta, se hacen cálculos equivalentes a (eq.4,5).

El diagrama de flujo corresponde a una rutina programada en lenguaje C, a la que el programa le “pasa” un puntero indicado como HPtr. Ese puntero permite acceder a una estructura o arreglo de datos para el canal correspondiente, que luego se almacena en memoria no-volátil. A efectos de comprender mejor el diagrama de flujo, se transcribe el template (plantilla) de dicha estructura y los elementos que la componen.

```

struct calib {
  UBYTE   ch; /* Numero del Canal uso interno */
  char    Name[CAL_NAMELEN]; /* Nombre del Canal */
  char    Label [CAL_LABELLEN]; /* Etiqueta del canal, modifiable. */
  char    SensorTyp[CAL_SENSORTYPLEN]; /* Tipo Sensor, N° serie, Fabric. */
  char    ADCRange[CAL_ADCRANGELEN]; /* Cadena indica rango A/D */
  UBYTE   ADCRng; /* Parametro rango del MAX197, AlOng */
  char    Units[CAL_UNITSLEN]; /* Cadena, unidades de U.I... */
  BOOLEAN CalY_N; /* Calibrado or no... */
  char    CalDate[CAL_DATELEN]; /* ... y cuando fue. */
  BOOLEAN EnabledY_N; /* Puede ser deshabilitado... */
  char    EURange[CAL_EURANGELEN]; /* Rango en U.I. ... */
  FP      C_Def; /* Parametro default Coefficient */
  FP      G_Def; /* Parametro default Gconv */
  FP      C_Cal; /* Calibr. Cuenta Ccal Labrosse */
  FP      G_Cal; /* Calibr. Ganancia Gcal */
};

```

Por facilidad de implementación, se observará en el diagrama de flujo que el programa no sigue exactamente la secuencia de (eq.4,5), sino que los coeficientes se calculan realizando primero el cálculo de C_{cal} , que resulta:

a) una operación sencilla si $U.I._1 = 0$ (es común tomar el primer valor en 0.0)

$$C_{cal} = -(C_1 + C_{offset}) \quad (\text{eq.6})$$

b) una división de las ecuaciones (eq.10,11) si $U.I._1 \neq 0$. Esto permite despejar:

$$\frac{U.I._1}{U.I._2} = \frac{(C_1 + C_{offset} + C_{cal}) * G_{conv} * G_{cal}}{(C_2 + C_{offset} + C_{cal}) * G_{conv} * G_{cal}} \quad (\text{eq.7})$$

$$C_{cal} \left(\frac{U.I._1}{U.I._2} \right) - C_{cal} = C_1 + C_{offset} - \left(\frac{U.I._1}{U.I._2} \right) (C_2 + C_{offset}) \quad (\text{eq.8})$$

o sea que puede obtenerse:

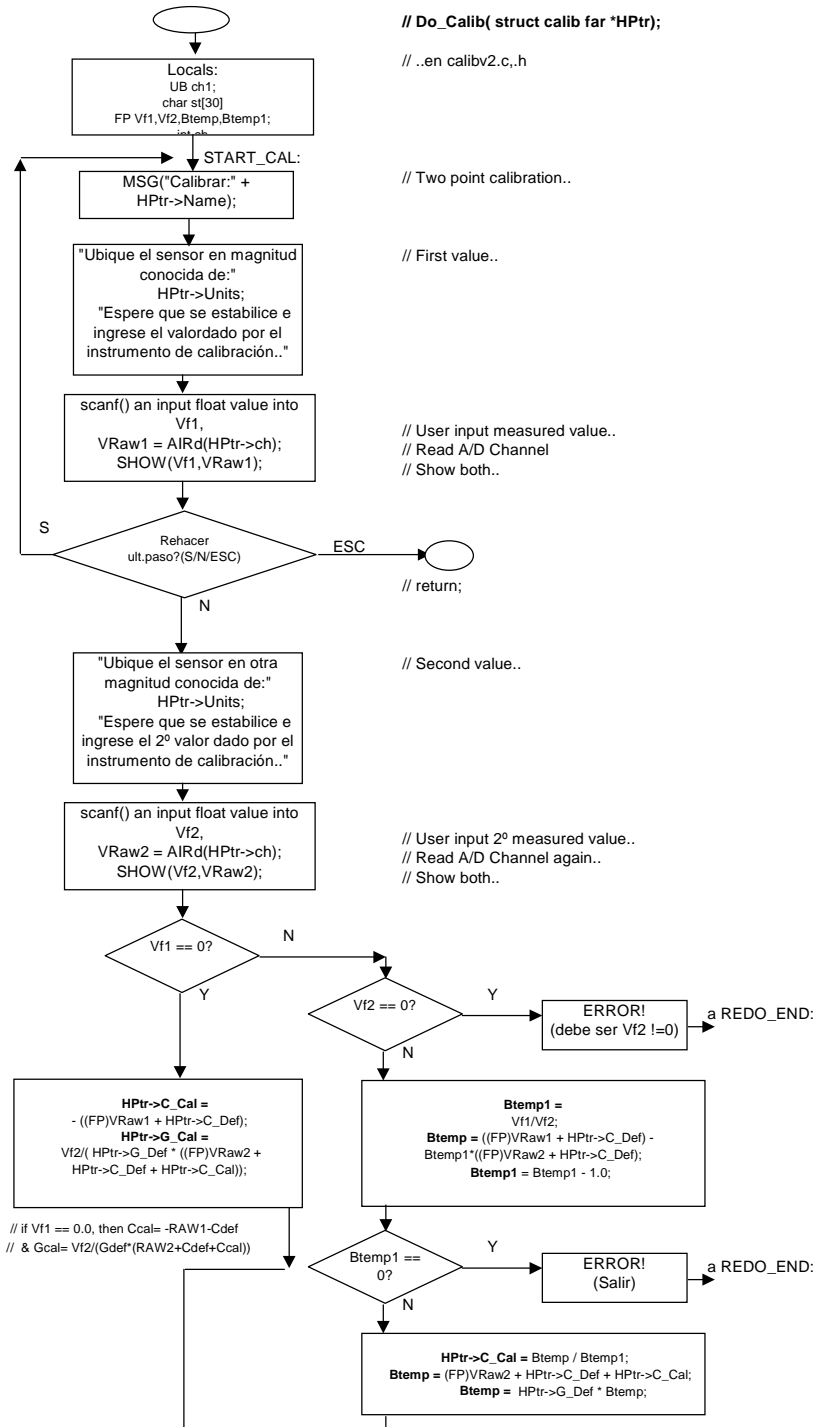
$$C_{cal} = \frac{C_1 + C_{offset} - \left(\frac{U.I._1}{U.I._2} \right) (C_2 + C_{offset})}{\left(\frac{U.I._1}{U.I._2} - 1 \right)} \quad (\text{eq.9})$$

Finalmente, para ambos casos a) y b), se sigue que:

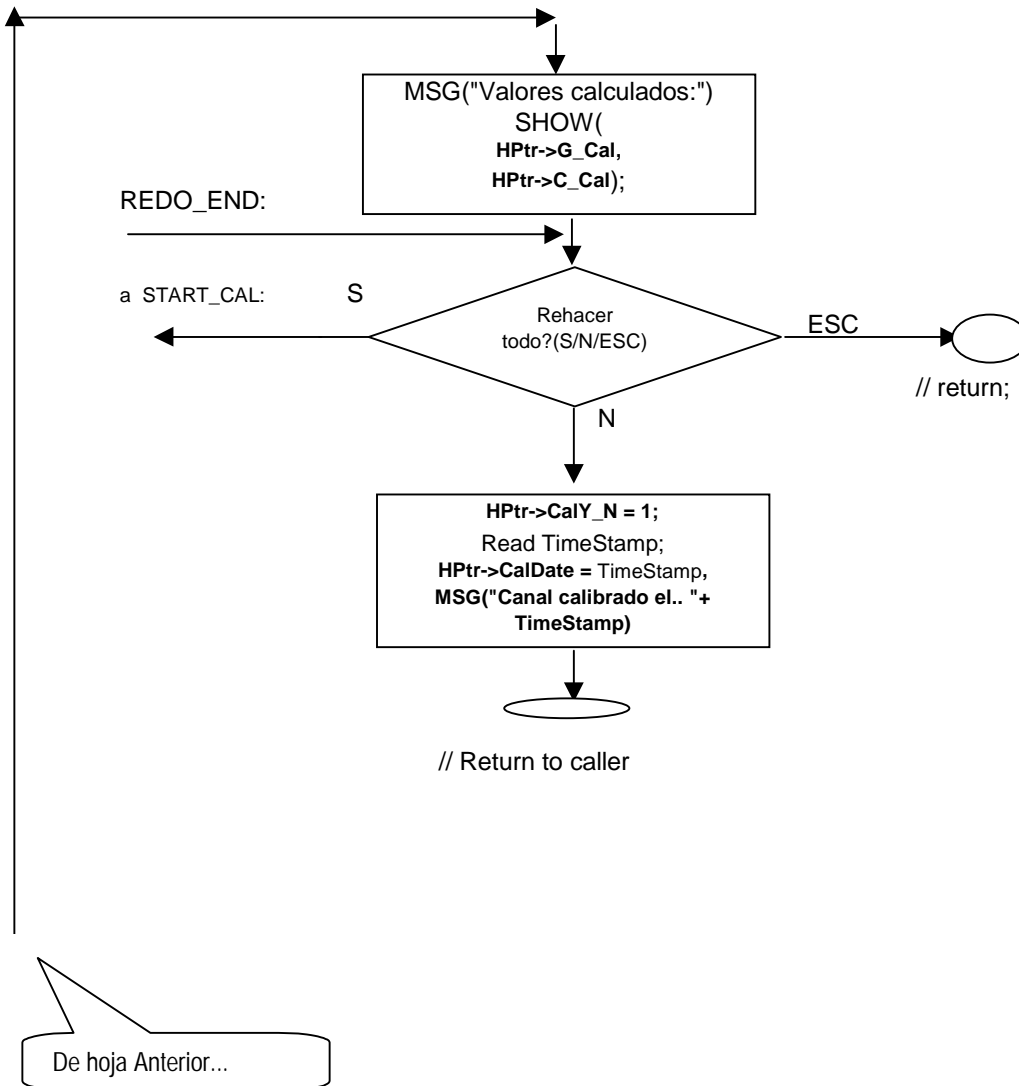
$$G_{cal} = \frac{1}{G_{conv}} \left[\frac{U.I._2}{C_{offset} + C_2 + C_{cal}} \right] \quad (\text{eq.10})$$



ANEXO I DIAGRAMA EN BLOQUES DE RUTINA DE CALIBRACIÓN - METODO DE DOS PUNTOS



Continúa en Hoja Siguiente



3. Referencias

[Ref.1] *Embedded System Building Blocks, 2nd.Ed.*, Jean Labrosse, Ch. 10. R&D Books 2000, ISBN 0-87930-604-1