



# Capítulo 9

## Posibles mejoras Líneas de estudio futuras

---

- 9.1.- Uso de más elementos internos del PIC
  - 9.1.1.- Uso del comparador interno del PIC
  - 9.1.2.- Uso de la tensión de referencia
- 9.2.- Mejor montaje
- 9.3.- Cálculo del tiempo residual
- 9.4.- Medición de la temperatura
- 9.5.- Filtrado de muestras

Aunque ya en este primer intento se tiene un buen resultado con el prototipo, existen distintas formas de intentar mejorar el diseño y la realización del anemómetro, para futuras versiones. El objetivo es mejorar alguna faceta en la medición, emplear menos número de componentes o dotar al anemómetro de alguna función distinta vía software.

A continuación se exponen estos posibles cambios en el diseño que en futuras versiones pueden añadirse para mejorar o añadir alguna tarea del dispositivo.

## 9.1.- Uso de más elementos internos del PIC

Una de las cosas que se podría intentar es realizar algunas funciones más usando los módulos del PIC. Como se describió en el apartado correspondiente al microcontrolador usado, el PIC18F458 cuenta con un gran número de módulos internos, con distinta funcionalidad. Entre ellos existen algunos que podrían resultar interesantes para este caso. Se expondrán las razones por las que pueden ser interesantes y porqué no se han usado en este primer proyecto:

### 9.1.1.- Uso del comparador interno del PIC

El microcontrolador contiene un módulo de comparación con dos comparadores analógicos. Las entradas de estos comparadores pueden ser tomadas en sendos pines de entrada o también pueden ser generadas en el módulo de tensión de referencia para el comparador. De esta forma una de ellas podría ser la salida de la parte de amplificación mientras que la otra, tensión constante de referencia, puede ser generada por dicho módulo de tensiones de referencia. El ahorro que se podría introducir en cuanto a número de dispositivos en placa es la eliminación de un nuevo componente discreto que realice la labor de comparación. Sería este propio módulo el que realizara esta labor y el que generara la interrupción que pusiera fin a la cuenta de tiempo.

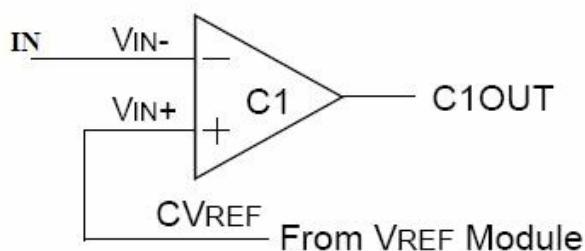


Ilustración 85: Comparador y tensión de referencia

Este comparador interno del PIC no se ha usado porque fue probado en un primer momento con resultados no muy satisfactorios, y por tanto se decidieron probar otros dispositivos. Una posible causa de estos resultados es que en estos primeros momentos se tenía una salida de la parte de amplificación no muy concreta ni muy definida, por lo que muy probablemente en las condiciones actuales si se produzca un resultado mucho mejor.

### 9.1.2.- Uso de la tensión de referencia

El pin número 2 del PIC es el pin *RA0/AN0/CVREF* y puede ser usado para tener una tensión de referencia generada por el PIC internamente. Esta tensión es la misma que puede ser usada para el módulo de comparación anterior, y a lo que llevaría su uso es al ahorro de elementos que generen ese valor de voltaje.

## **9.2.- Mejor montaje**

Cabe esperar un buen resultado si se realiza una placa profesional con todo el equipo de componentes que se han probado y escogido. La placa que se está utilizando ya presenta un buen resultado a pesar de todos los problemas y cambios que se hayan realizado en ella tras su montaje inicial, de ahí que la realización de una placa final más profesional puede ofrecer un resultado muy fiable.

### 9.3.- Calculo del tiempo residual

Para dar una estimación de este tiempo residual se podría actuar de la forma siguiente. Para ello sería necesario que las dos distancias entre sensores en los dos sentidos fueran diferentes, aunque rondarán ambas los 20cm. En este caso se intentaría averiguar por diferencias de tiempo:

$$\overline{T}_{01} = \frac{T_{01} + T_{10}}{2} = \frac{t_{01} + t_{10}}{2} + \Phi \rightarrow \text{Correspondiente al recorrido } d_{01}$$

$$\overline{T}_{23} = \frac{T_{23} + T_{32}}{2} = \frac{t_{23} + t_{32}}{2} + \Phi \rightarrow \text{Correspondiente al recorrido } d_{23}$$

A continuación se verá donde se quiere llegar y un ejemplo con valores obtenidos experimentalmente:

$$\left. \begin{array}{l} \overline{T}_{01} \cong \frac{e_{01}}{c} + \Phi \\ \overline{T}_{23} \cong \frac{e_{23}}{c} + \Phi \end{array} \right\} \Rightarrow \overline{T}_{01} - \overline{T}_{23} = \frac{e_{01} - e_{23}}{c}$$

$$t_{01} = \frac{e_{01}}{c} = e_{01} \cdot \frac{\overline{T}_{01} - \overline{T}_{23}}{e_{01} - e_{23}}$$

$$t_{23} = \frac{e_{23}}{c} = e_{23} \cdot \frac{\overline{T}_{01} - \overline{T}_{23}}{e_{01} - e_{23}}$$

Se ha eliminado con este cálculo la dependencia con el tiempo residual  $\Phi$  y con ello se tiene una estimación del tiempo de propagación entre sensores en ambos casos. La expresión denota que las distancias entre transmisores y receptores en ambas direcciones ( $e_{01}$  y  $e_{23}$ ) no pueden ser exactamente iguales para que se pueda usar este método.

Un ejemplo de su uso se expone a continuación, usando los datos siguientes obtenidos en simulación:

|                          |       |
|--------------------------|-------|
| $\overline{T}_{01}$ (μs) | 646,2 |
| $\overline{T}_{23}$ (μs) | 676   |
| $e_{01}$ (cm)            | 19,4  |
| $e_{23}$ (cm)            | 20,4  |

Para estos datos se tendrían los siguientes resultados:

$$\left. \begin{array}{l} t_{01} = 578,1 \mu s \\ t_{23} = 607,9 \mu s \end{array} \right\} \Rightarrow \Phi = 68,1 \mu s$$

Esos tiempos serían los que se usarían en el cálculo de  $\Delta v$  en cada dirección, quitando de esta forma un error que se produce al usar un valor constante. Así se tendría en cuenta la dependencia con la temperatura.

## 9.4.- Medición de la temperatura

Este puede ser sin duda el objetivo más complicado y más ambicioso. También es a la vez el más útil para ser integrado en el sistema donde se desea. Por esta razón será el primero de los objetivos a estudiar tras la satisfactoria realización del anemómetro.

Dar una estimación a la vez de la velocidad del viento y de la temperatura es complicado porque ambos están interrelacionados. La *velocidad adiabática* del sonido vimos que depende de los factores atmosféricos. Cuando queremos estimar la velocidad del viento se puede abstraer su dependencia de factores como la temperatura o la humedad siempre que el error que esto genere no sea superior a un cierto límite que se considere admisible.

El conjunto de pasos que se daría para calcularla en función de los resultados que tenemos son los siguientes. Se parte de una situación inicial de temperatura conocida  $T_0^a$  y un valor de tiempo medio para una de las dos direcciones (por ejemplo la 01)  $T_{01}^0$  correspondiente a esa temperatura. Para distintos valores de temperatura ese tiempo  $T_{01}$  será distinto, mayor a menor temperatura y viceversa. Midiendo la diferencia de tiempo con respecto a esa situación conocida se puede averiguar la variación de la temperatura  $\Delta T^a$  de la siguiente forma:

$$v = 331,5 + 0,6 \cdot T^a$$

$$t = \frac{e}{v} = \frac{e}{331,5 + 0,6 \cdot T^a}$$

$$\Delta T_{01} = T_{01} - T_{01}^0$$

$$331,5 + 0,6 \cdot T^a = \frac{e}{t} \Rightarrow T^a = \frac{1}{0,6} \cdot \left( \frac{e}{t} - 331,5 \right)$$

Pasando a diferenciales podemos quedarlo en función de datos conocidos para así obtener la relación final con respecto a las diferencias de tiempo respecto a la situación conocida:

$$\frac{\partial T^a}{\partial t} = \frac{-1}{0,6} \cdot \frac{e}{t^2} \Rightarrow \text{Y de ahí a la relación que queremos llegar en función de las diferencias de tiempo}$$

$$\frac{\Delta T^a}{\Delta T_{01}} = \frac{-1}{0,6} \cdot \frac{e}{t^2} \Rightarrow \boxed{\Delta T^a = \frac{-1}{0,6} \cdot \frac{e}{t^2} \cdot \Delta T = \frac{k}{0,6} \cdot \Delta T}$$

Haciendo uso del valor  $k$  calculado anteriormente

## 9.5.- Filtrado de muestras

Se quiere añadir un filtrado a realizar sobre las muestras para suavizar los cambios en la velocidad en lugar de realizar una media sobre los datos que se toman directamente. Con estas rutinas se quieren usar los valores tomados en lugar de realizar el cálculo de los valores medios sobre un número grande de valores.

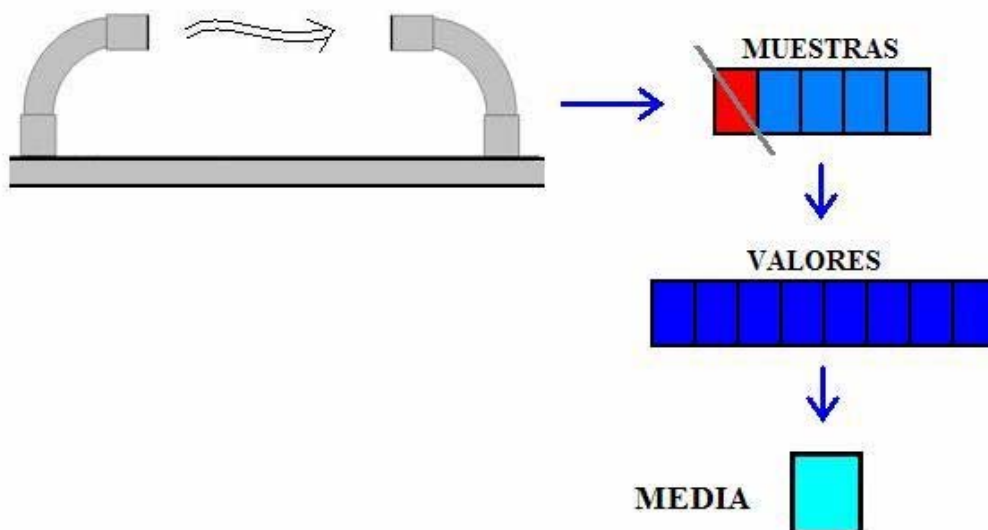


Ilustración 86: Cálculo de la media para un sentido

El funcionamiento consiste en almacenar las muestras en un vector para cada sentido. Una vez que se han tomado NMUESTRAS en cada sentido se procede a calcular la media correspondiente a cada sentido con las siguientes llamadas a función:

```
media01=actualiza_media(muestras01, valores01, &anterior01);
media10=actualiza_media(muestras10, valores10, &anterior10);
media23=actualiza_media(muestras23, valores23, &anterior23);
media32=actualiza_media(muestras32, valores32, &anterior32);
```

Se llama a la función **actualiza\_media** para que calcule el valor medio de cada sentido. Esta función usa a su vez la función **desplaza** para ir colocando los valores nuevos en un vector siempre que la diferencia con la muestra anterior esté por debajo de la cantidad fijada por la constante **UMBRAL**. Así se realizará el filtrado que se quiere sobre las muestras, y con los valores medios de cada sentido se puede seguir como se hacía antes.

Sería bueno en este caso que algunas variables sean declaradas como locales a la función **main**. De esta forma se consigue un ahorro en la cantidad de memoria ocupada en el PIC por el programa final.

```

*****
*      Función actualiza_media      *
*****

int16 actualiza_media(int16 *muestras, int16 *valores, int16 *anterior)
{
    int i=0;
    int v=0;
    int16 media=0;
    int16 diferencia=0;

    for(i=1;i<NMUESTRAS;i++)          // Empieza en la 1 para descartar
    {                                  // la primera

        diferencia=abs((signed int16)(muestras[i]-valores[0]));
        if(diferencia>=UMBRAL)
        {
            if((abs((signed int16)(muestras[i]-anterior)))<UMBRAL)
            {
                desplaza(valores,muestras[i]);
            }
        }
        else
        {
            desplaza(valores,muestras[i]);
        }
        anterior=muestras[i];
    }

    for(v=0;v<NVALORES;v++) media+=valores[v];

    return media/NVALORES;
}

```

```

*****
*      Función desplaza      *
*****

void desplaza(int16 *valores, int16 muestra)
{
    int v=0;

    for(v=NVALORES-2; v==0; v--) valores[v+1]=valores[v];
    valores[0]=muestra;
}

```