



Biblioteca Universitaria

La consulta de este documento, que se lleva a cabo mediante claves de identificación y responsabilidad personal, es posible exclusivamente para fines de estudio personal o investigación. No se autoriza a reproducir su texto más que en forma de breves citas entrecomilladas, indicando el nombre del autor y la fuente. Por tanto, no se permite descargar, copiar, transformar ni grabar su contenido.

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Carrera

**SISTEMA EMBEBIDO PARA MEDIDA DE TEMPERATURAS
EN MÁQUINAS SÍNCRONAS DE IMANES PERMANENTES**
(Embedded system for temperature measurement in permanent magnet
synchronous machine)

**Para acceder al Título de
INGENIERO TÉCNICO DE TELECOMUNICACIONES**

Autor: Ismail Laarossi
Septiembre 2011

Con este proyecto, pongo fin a tres años de trabajo, esfuerzo y dedicación. Por eso, me gustaría dar las gracias a todas y cada una de las personas que han compartido conmigo este tiempo y en especial:

A Mario Mañana Canteli por confiar en mí y darme la oportunidad de realizar este proyecto.

A mis padres, mis hermanos y mis sobrinos, por todo el cariño y el apoyo que me han mostrado siempre.

A todos los profesores, por todo lo aprendido durante estos tres años.

A todos mis compañeros de clase porque han hecho que me quede un gran recuerdo de estos años.

A todos mis amigos, en especial a Lucia y Youssef, porque gracias a ellos las horas de estudio han sido más fáciles.

ÍNDICE

1.	INTRODUCCION.....	1
2.	MEDIDA DE TEMPERATURA.....	3
2.1	INTRODUCCIÓN.....	3
2.2	BIMETALES.....	3
2.3	DETECTORES DE TEMPERATURA RESISTIVOS: RTD.....	4
2.3.1	Ejemplos comerciales.....	7
2.4	DISPOSITIVOS RESISTIVOS SEMICONDUCTORES: TERMISTORES.....	8
2.4.1	Ejemplos comerciales.....	11
2.5	SENSORES DE TEMPERATURA TERMOELÉCTRICOS: TERMOPARES.....	12
2.5.1	Ejemplos comerciales.....	13
2.6	SENSORES DE TEMPERATURA INTEGRADOS.....	14
2.6.1	Ejemplos comerciales.....	15
2.7	TABLA COMPARATIVA DE LOS SENSORES VISTOS.....	15
3.	MICROCONTROLADOR PIC18F46K20.....	16
3.1	INTRODUCCIÓN.....	16
3.2	CARACTERÍSTICAS DE LOS MICROCONTROLADORES.....	16
3.3	ELECCIÓN DEL MICROCONTROLADOR.....	17

3.4 μ C PIC18F46K20.....	18
3.4.1 Organización de la memoria.....	19
3.4.2 Puertos de entrada/salida.....	22
3.4.2.A Puerto A.....	23
3.4.2.B Puerto B.....	23
3.4.2.C Puerto C.....	23
3.4.2.D Puerto D.....	24
3.4.2.E Puerto E.....	24
3.4.2.F Corriente máxima.....	24
3.4.3 Temporizadores.....	24
3.4.3.A Temporizador0.....	25
3.4.3.B Temporizador1.....	26
3.4.3.C Temporizador2.....	27
3.4.3.D Temporizador3.....	28
3.4.4 Configuraciones del oscilador.....	29
3.4.5 Reset.....	32
3.4.6 Módulos Captura/Comparación/PWM.....	33
3.4.6.A Modo de captura.....	33
3.4.6.B Modo de comparación.....	34
3.4.6.C Modo de modulación de anchura de pulsos PWM.....	36
3.4.7 Módulo Convertidor Analógico a Digital (A/D).....	37
3.4.8 Módulo comparador.....	39
3.4.9 Módulo Puerto Serie Síncrono Principal (MSSP).....	41
3.4.9.A Modo SPI.....	41
3.4.9.B Modo I ² C.....	42
3.4.10 Transmisor/Receptor Asíncrono/Síncrono Universal (USART).....	42

3.4.10.A Modo Asíncrono del USART	43
3.4.10.B Modo Maestro Síncrono del USART.....	43
3.4.10.C Modo Esclavo Síncrono del USART.....	43
3.5 HERRAMIENTAS DE DESARROLLO.....	44
4. DESARROLLO.....	46
4.1 HARDWARE DEL SISTEMA.....	46
4.1 SOFTWARE DEL SISTEMA.....	53
5. RESULTADOS.....	67
6. CONCLUSIONES.....	74
7. BIBLIOGRAFÍA.....	76

1. Introducción

Muchas son las aplicaciones que necesitan medir y controlar la temperatura, una magnitud que atribuye de una forma o de otra en las propiedades físicas y/o químicas de todos los materiales, como por ejemplo su estado (sólido, líquido, gaseoso, plasma), su volumen, la solubilidad, la presión de vapor, su color o la conductividad eléctrica.

A lo largo de la historia se han desarrollado muchos instrumentos para la medida de esta variable, según el uso a que se destinan y su modo de utilización, aprovechando el efecto producido por la temperatura sobre un cuerpo.

Aquí, en este proyecto de fin de carrera, se ha diseñado un sistema más, que cumpla unas especificaciones especiales, que se resumen en:

- Menor tamaño
- Medir un rango amplio de temperatura
- Consumir la mínima energía posible

Estas restricciones han sido origen fundamentalmente, del uso del diseño, ya que este sistema fue pensado para integrarse en el rotor de maquinas síncronas de imanes permanentes para poder medir y registrar las variaciones de temperatura, tanto en la superficie de los imanes de dichas maquinas, así como la temperatura ambiente en la que se encuentran.

Y como se puede intuir, el tamaño de la placa, donde van a ir todos los componentes utilizados en el diseño, tiene que ocupar el mínimo espacio posible permitiendo así, una fácil integración en el rotor.

Las temperaturas a las que se puede llegar con estos imanes, en el caso de que las maquinas están funcionando, son muy altas (rondan los 200°C). En el caso en el que el rotor esté parado y en situaciones de mucho frio hay que cubrir también las

temperaturas negativas (-20°C aproximadamente). Eso es lo que explica el amplio rango de medida que se pide al sistema.

El último objetivo de este diseño ha sido el ahorro de energía, factor muy importante y que está presente en todos los diseños. En este proyecto, este factor ha sido muy decisivo a la hora de elegir los componentes a usar, ya que se ha tenido muy en cuenta el consumo de cada uno y la posibilidad de poner el dispositivo en modo reposo, lo cual ha limitado mucho las posibilidades de elección.

Finalmente cabe añadir, que aunque este sistema ha sido diseñado y pensado para un caso concreto, tiene la gran ventaja de poder ser utilizado en diversos sistemas, que necesiten alguna de las características anteriormente mencionadas; lo cual permite que este diseño sea propuesto con una visión global y no simplemente particular.

2. Medida de Temperatura

2.1 INTRODUCCIÓN

Un sensor de temperatura es un dispositivo capaz de detectar variaciones de temperatura y transformarlas en magnitudes que facilitan la medida, generalmente tensión, intensidades o impedancias. En la actualidad existen numerosos tipos de sensores de temperatura que pueden ser usados para diferentes aplicaciones. A continuación se indican algunos tipos de los sensores usuales junto a algunas de sus características más relevantes.

2.2 BIMETALES

Los bimetales son sensores de temperatura que generan una señal de salida no eléctrica transducible, y están formados por dos láminas de metal muy unidas y con distinto coeficiente de dilatación térmica. Lo que hace que, cuando se someten a un cambio de temperatura, una lámina se dilata más arrastrando la otra, haciendo que el bimetel se deforme según un arco circular uniforme [1].

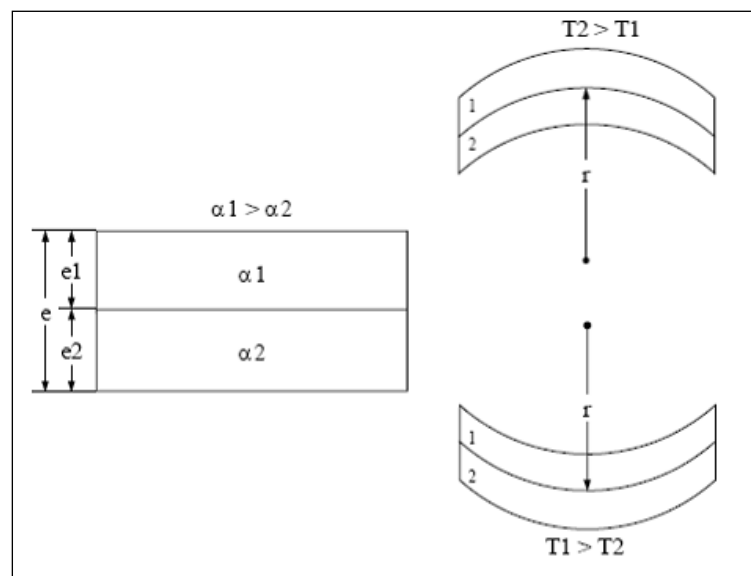


Figura 2.1: Bimetel

El radio de curvatura de un bimetel al pasar de temperatura T_1 a otra T_2 , viene dado por:

$$r = \frac{e[3(1+m)^2 + (1+mn)\left(m^2 + \frac{1}{mn}\right)]}{6(\alpha_1 - \alpha_2)(T_2 - T_1)(1+m)^2} \quad (2.1)$$

Donde:

- α_1, α_2 son coeficientes de dilatación lineal
- e es el espesor total de la pieza
- $n = \frac{E_1}{E_2}$, relación en módulos de elasticidad
- $m = \frac{e_2}{e_1}$, relación entre espesores

Habitualmente se emplean materiales con módulos de elasticidad y espesores similares, por lo tanto la expresión anterior se reduce a:

$$r = \frac{2e}{6(\alpha_A - \alpha_B)(T_2 - T_1)} \quad (2.2)$$

Algunos de los campos de aplicación más importantes para el bimetálico son:

- Control de temperatura (termostatos de ambiente, planchas, etc.).
- Protección de sobrecarga eléctrica (motores eléctricos, transformadores, etc.).
- Compensación de temperatura (para carburación de automotores, etc.).
- Medida de temperatura (termómetros, etc.).

Se suministra en forma de cintas laminadas en frío arrolladas y en espesores a partir de 0,5 mm. También en forma de cintas rectas cortadas a longitudes máximas de 4m, con un ancho de 1,5 - 150 mm y de 0,08 - 3,5 mm de espesor. También piezas sueltas terminadas estampadas, cortadas, dobladas o en espiral.

2.3 DETECTORES DE TEMPERATURA RESISTIVOS: RTD

El fundamento de los RTD (Resistance Temperature Detector), está basado en la variación de la resistencia de un conductor con la temperatura. En un conductor las vibraciones de los átomos, alrededor de sus posiciones, aumentan según lo hace la temperatura, reduciendo la velocidad media de los electrones. Y como el número de

electrones disponibles para la conducción no cambia apreciablemente con la temperatura, la resistencia del material aumenta. La dependencia con la temperatura se puede expresar como:

$$R_T = R_0(1 + \alpha T + \alpha_2 T^2 + \alpha_3 T^3 + \dots + \alpha_n T^n); T \text{ en } ^\circ\text{C} \quad (2.3)$$

Donde R_T y R_0 son las resistencias a la temperatura T y 0°C respectivamente y $\alpha, \alpha_2, \dots, \alpha_n$ son constantes del material. El símbolo de este dispositivo es el de la figura siguiente:

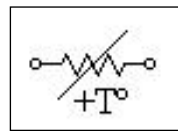


Figura 2.2: Símbolo del RTD

Los RTD son principalmente de dos tipos:

- Cilíndricos: que se utilizan con vainas de protección.
- Planos: para la medida de temperatura en superficies planas.

Las ventajas principales de estos dispositivos son su sensibilidad (diez veces más grande que la de los termopares), la alta repetibilidad y exactitud en el caso del platino, y el bajo coste en el caso del cobre y del níquel, que son los tres metales ampliamente usados. Para estos tres metales, ya para tener una alta precisión, se requieren generalmente de 2 a 3 coeficientes α_i . Muchas veces sólo se utiliza la constante α y así lograr una buena linealidad sobre rangos de temperatura amplios. La ecuación anterior se reduce a [1]:

$$R_T = R_0(1 + \alpha T) \quad (2.4)$$

En la tabla 2.1 se representan los parámetros más importantes del platino, cobre y del níquel y en la figura 2.3 las curvas de variación de la resistencia relativa en función de la temperatura. Se puede observar que el platino es el que mejor prestaciones ofrece.

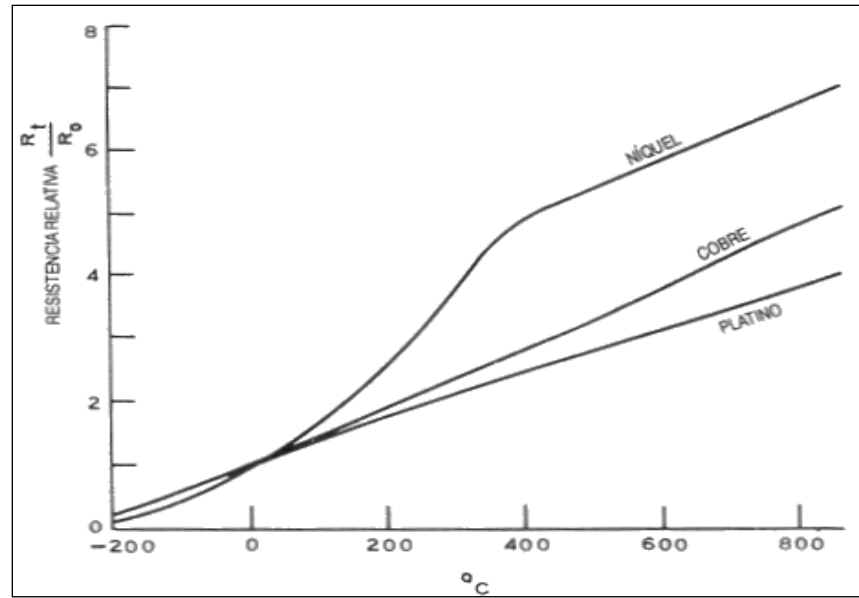


Figura 2.3: Curvas de variación de resistencia relativa frente a temperatura de diversos detectores de temperatura resistivos

Tabla 2.1: Parámetros de diversos detectores de temperatura resistivos

Parámetro	Platino(Pt)	Cobre (Cu)	Níquel (NI)
Resistividad ($\mu\Omega\text{xcm}$)	10.6	1.673	6.844
α ($\Omega/\Omega/^{\circ}\text{C}$)	0.00385	0.0043	0.00681
R_0 , Ω a 0°C	25, 50, 100, 200, 500,...	10 (20°C)	50,100,120
Margen ($^{\circ}\text{C}$)	-200 a $+850^{\circ}\text{C}$	-200 a $+260^{\circ}\text{C}$	-80 a $+320^{\circ}\text{C}$

Sin embargo hay varias limitaciones para medir la temperatura con estos dispositivos, como las que se citan a continuación:

- No se puede medir temperaturas próximas ni superiores a la de fusión del conductor.
- Hay que evitar deformaciones mecánicas que provocan variaciones en la resistencia del conductor.
- Es necesario evitar autocalentamientos que se producen por el circuito de medida.
- Tiempo de respuesta alto.
- Hay que linializar su característica (puente de Wheatson o circuito integrado e.j 1B41 “analog Devices”).

2.3.1 Ejemplos comerciales

➤ *DRG 010632A : SENSOR, PT100, TEMPERATURA, 6X50MM*

Temperature Sensing Range:-50 °C a +200 °C; Resistencia: 100OHM; Lead Length: 2m; Material:Acero inoxidable; Intervalos de medición:-50 °C a +200 °C; Temperatura de operacion Max:200°C; diámetro de la sonda :6mm; longitud de la sonda :50mm;

PRECIO: 44,51 €



➤ *XF-314-FAR: SONDA, PT100, 150X3MM*

Rango de temperatura:-50 °C a +250 °C; Resistencia: 100OHM; Categoría de homologación: BS 1904 (1984) Clase B, DIN 43760; Diámetro externo :3mm; Intervalos de medición:-50 °C a +250 °C; temperatura de operación Max:250°C; longitud de la sonda:15mm;

PRECIO: 40,60 €



➤ *HRTS-5760-B-U-0-12:*

Rango de temperatura:-200 a 400 °C ; Resistencia: 1000ohms, coeficiente de temperatura: 0.00375Ohm/Ohm/°C

PRECIO: 20,00 € [2]



2.4 DISPOSITIVOS RESISTIVOS SEMICONDUCTORES: TERMISTORES

Los termistores son sensores, basados en la variación de la resistencia de un semiconductor con la temperatura, debido a la variación de la concentración de los portadores. Se utilizan con mucha frecuencia en aplicaciones que requieren mucha precisión, porque son los más sensibles (tienen un coeficiente de temperatura muy elevado), permitiendo detectar pequeñas variaciones de la temperatura que no se pueden detectar con los RTD o los termopares [1].

El término termistor proviene de la contracción del término en inglés Thermally Sensitive Resistor. Se diferencia entre dos tipos de termistores:

- NTC (Negative Temperature Coefficient), con coeficiente de temperatura negativo, al aumentar la temperatura, aumenta también el número de portadores reduciéndose la resistencia.
- PTC (Positive Temperature Coefficient), el semiconductor está fuertemente dopado, lo que hace que éste adquiere propiedades metálicas, presentando coeficiente de temperatura positivo.

En la figura que sigue se representan los símbolos de los dos tipos:



Figura 2.4: Símbolos de los NTC y PTC

Para un NTC la resistencia varía con la temperatura siguiendo la expresión:

$$R_T = R(T_0)e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)} = R(T_0)a_0e^{B/T} \quad \text{con} \quad a_0 = \frac{1}{e^{B/T_0}} \quad (2.5)$$

Donde:

- T_0 es la temperatura de referencia en K típicamente $T_0=298K$.
- B es un parámetro denominado temperatura característica del material.
- $R(T_0)$ es la resistencia a la temperatura de referencia.

Para usar el NTC en la medida de temperatura, hay que tener en cuenta las intensidades que atraviesan estos dispositivos. Porque si las corrientes son pequeñas, el consumo de potencia (RI^2) va a ser pequeño para registrar aumentos apreciables de temperatura, o lo que es igual, resistencia óhmica nominal (a la temperatura ambiente). Ahora si se aumenta la tensión aplicada al termistor, se llegará a un valor de intensidad en que la potencia consumida provocará aumentos de temperatura suficientemente grandes como para que la resistencia del termistor NTC disminuya apreciablemente, incrementándose la intensidad hasta que se establezca el equilibrio térmico. Y esto provocara confusión en la medida, debido al autocalentamiento del termistor.

Para un PTC la relación entre la resistencia y la temperatura es difícil expresarla mediante una fórmula. Por lo tanto los cálculos con PTCs se basan en métodos gráficos. Y según la composición y el dopado hay dos tipos de comportamiento. Los de tipo cerámico presentan un cambio brusco de resistencia cuando se alcanza la temperatura de Curie¹. Y los basados en silicio dopado presentan una variación más suave con la temperatura.

La característica estática corriente/voltaje del PTC muestra los límites de corriente que puede soportar. Se observa que hasta un determinado valor de voltaje, la característica I/V sigue la ley de Ohm, pero al seguir aumentando la tensión el PTC alcanza la temperatura de conmutación, aumentando su resistencia, lo que hace que la intensidad que circula por él disminuye (Figura 2.5).

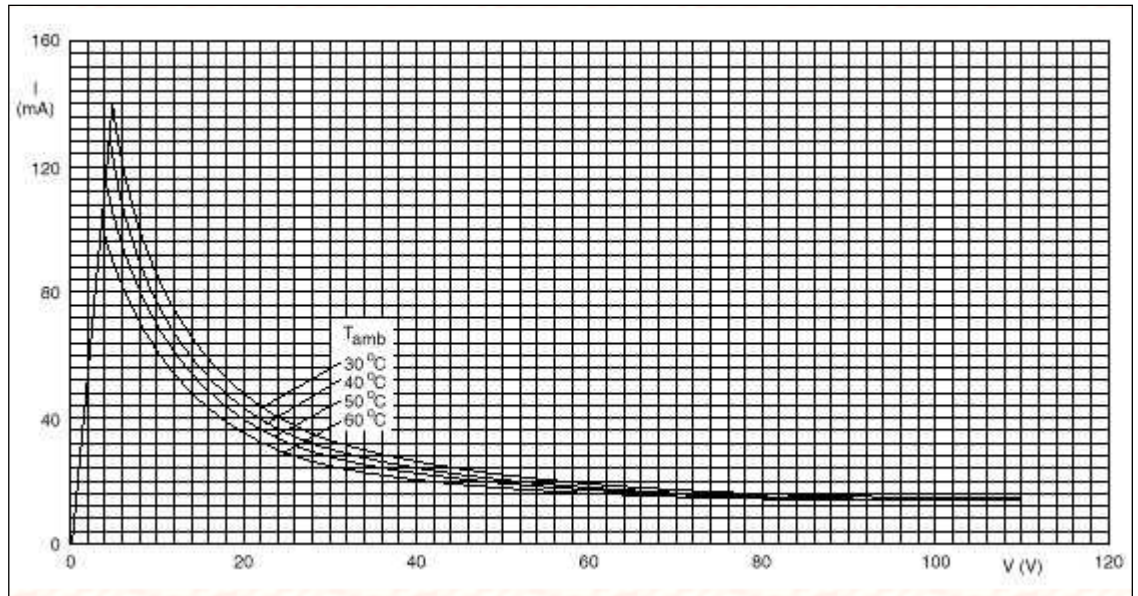


Figura 2.5: Característica I-V de un termistor PTC

Puede por lo tanto verse que el termistor PTC tiene unas excelentes propiedades de protección, limitando la corriente de carga si el voltaje de suministro, la temperatura o la corriente exceden de un valor crítico.

Un termistor PTC se difiere de otro NTC en varios aspectos:

- El coeficiente de temperatura de un PTC es positivo sólo sobre un rango de temperaturas. Lo que hace que el uso de un PTC se restringe al rango en el que el coeficiente es positivo.
- El valor absoluto del coeficiente de temperatura de un PTC, en muchos casos, es mucho más alto que el de un termistor NTC.
- En los PTCs, a la temperatura a la que el coeficiente de temperatura del termistor comienza a tener valor aprovechable, se le llama temperatura de conmutación (T_{switch}), que es la temperatura más alta de las dos en las que el valor de la resistencia del PTC es el doble de la resistencia nominal.

En la figura 2.6 se muestra una comparación de las curvas características resistencia/temperatura de los termistores PTC y NTC.

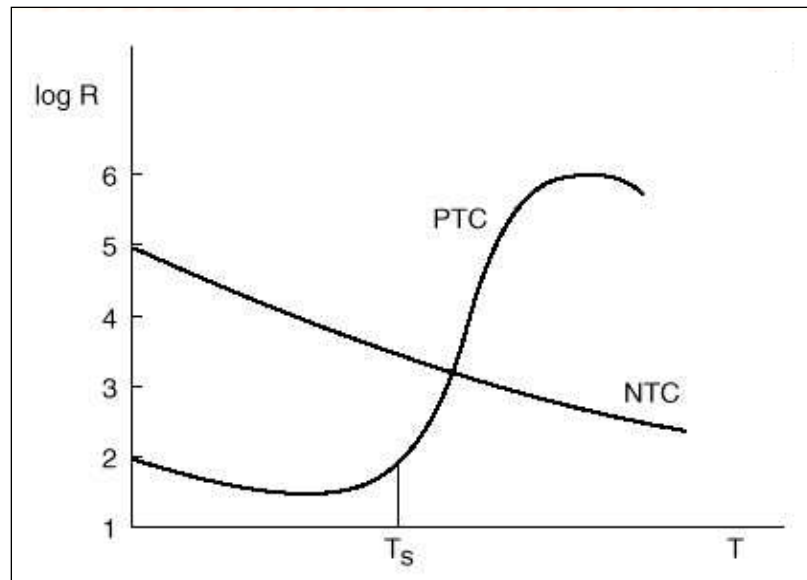


Figura 2.6: Comparación del comportamiento de los termistores PTC y NTC.

El mayor inconveniente de los termistores es la no linealidad que presentan, debido a la dependencia altamente exponencial con la temperatura. Para resolver este problema, se recurre a las técnicas de linealización para mejorar el comportamiento de no linealidad del termistor, como por ejemplo utilizar el método del divisor de tensión colocando una resistencia fija en serie con el NTC.

2.4.1 Ejemplos comerciales:

- B57560G502F: termistor NTC, coeficiente de temperatura $xx\%/^{\circ}\text{C}$, constante de tiempo 15s,

Tolerancia: $\pm 1\%$, Potencia nominal máxima 50mW. Rango de temperatura: -40 a 300 $^{\circ}\text{C}$

PRECIO: 4,92 €



- Termistor conex. axiales DO-35 NTC,30K:Termistor PTC, coeficiente de temperatura $3.8\%/^{\circ}\text{C}$, constante de tiempo 7s, Tolerancia: $\pm 10\%$, Rango de temperatura: -55 a 250 $^{\circ}\text{C}$

PRECIO: 1,576 €



- Termistor conex. axiales DO-35 NTC,20K: Termistor NTC, coeficiente de temperatura 3.8%/°C, constante de tiempo 7 s, Tolerancia: ±10%, Potencia nominal máxima 50mW. Rango de temperatura: -55 a 250°C
PRECIO: 1,576 € [2]



2.5 SENSORES DE TEMPERATURA TERMOELÉCTRICOS: TERMOPARES

Los termopares son dispositivos basados en los fenómenos termoeléctricos (efecto seebeck), que se producen cuando existe una diferencia de temperatura entre partes distintas de un sistema conductor o semiconductor.

Para medir la temperatura utilizando el efecto seebeck, dos conductores de diferente material se unen en el punto en donde se quiere medir la temperatura (unión de medida) y los extremos libres de ambos conductores se mantienen a la misma temperatura, que debe ser conocida (unión de referencia o unión fría).

Si se somete el termopar a una temperatura T_j , con el efecto seebeck, aparece una tensión que es función de la temperatura de la unión de medida y de la composición de los metales:

$$V_{AB} = \alpha_{AB}(T_j - T_{ref}) + b(T_j^2 - T_{ref}^2) \quad (2.6)$$

Donde α_{AB} y b son constantes termoeléctricas que dependen de los materiales que forman las uniones, T_j es la temperatura de la unión de medida y T_{ref} es la temperatura de la unión de referencia.

Los termopares se suministran calibrados dentro de un amplio rango de temperaturas. Los valores correspondientes a la tensión obtenida con determinados termopares se tabulan en función de la temperatura de la unión de medida cuando la unión de referencia se mantiene a 0°C.

Para aplicaciones de precisión, sobre todo en sistemas de medida automático se pueden emplear polinomios que aproximan las tablas con una exactitud dependiente de su orden:

$$T = a_0 + a_1 v + a_2 v^2 + \dots + a_n v^n \quad (2.7)$$

Donde v es la tensión del termopar en voltios, T es la temperatura en °C y a_n son coeficientes que dependen de los materiales que componen el termopar.

Los bajos niveles de salida que proporcionan los termopares es la principal desventaja de estos dispositivos. Este problema se soluciona con circuitos acondicionadores (ej:1B51 el AD594/AD595). [1]

2.5.1 Ejemplos comerciales:

- Z2-K-2M (IEC) : N.º of Strands x Strand Size:1 x 0.2mm; longitud del cable :2m; Material :Nickel Chromium / Nickel Aluminium; Tipo de termopar:k; rango de temperatura: -50 °C a +250 °C; Diseño del conductor:1/0,2 mm; Composición del material: Nickel-chromium; PRECIO: 5,36 €



- Z2-K-1 X 5 : SENSOR, T/PAR, K, Z2, PAQUETE 5 UDS
N.º of Strands x Strand Size:1 x 0.2mm; Longitud del cable:1m; Material:Nickel Chromium / Nickel Aluminium; Tipo de termopar:k; Rango de temperatura:-75 °C a +250 °C; Diseño del conductor:1/0,2 mm; PRECIO: 24,44 € [2]



2.6 SENSORES DE TEMPERATURA INTEGRADOS

Para todos los tipos de sensores estudiados anteriormente se necesita añadir circuitería adicional para producir una tensión o una intensidad de salida razonablemente grandes. Por eso, generalmente, se dispone de estos circuitos electrónicos en un solo circuito integrado.

Estos transductores de temperatura están basados en la dependencia con la temperatura de la tensión de una unión PN polarizada directamente. Los modelos comerciales presentan una característica entrada-salida lineal.

El coeficiente de temperatura de una unión PN de un diodo viene dado por:

$$TC(V_D) = \frac{V_D - V_{Go}}{T} - 3 \frac{k}{q} \quad (2.8)$$

Donde V_D es la tensión entre los terminales del diodo y V_{Go} es la tensión de salto de banda.

Para un transistor, y en la mayoría de los casos, se utilizan transistores emparejados de modo que $I_{S1} = I_{S2}$, el coeficiente de temperatura es constante y viene dado por la expresión:

$$\alpha = \frac{k}{q} \ln 2 = 59.78 \frac{\mu v}{K} \quad (2.9)$$

Los sensores basados en este principio se llaman PTAT (transductores proporcionales a la temperatura absoluta), y dependiendo de la salida del PTAT, se habla de dos tipos:

- VPTAT: sensor que produce una tensión a la salida en función de la temperatura (ej. LM335 y LM34).
- IPTAT: sensor que produce una intensidad a la salida en función de la temperatura (ej. LM334 y AD590 y AD592).

2.6.1 Ejemplos comerciales:

- LM95172EWG: es un sensor de temperatura integrado con salida digital con un rango de temperatura de -40 a 200 °C. Puede ser programado para operar con 13 o 16 bits.

Precio: 117.56 € [2]

2.7 TABLA COMPARATIVA DE LOS SENSORES VISTOS:

	Termopar	RTD	Termistor	Integrado
Ventajas	*Autogenerador *Imple *Robusto *Barato *Gran variedad *Amplio rango de temperatura	*El más estable *El más exacto *más lineal que el termopar	*Salida grande *Rápido	*El más lineal *el de mayor salida *barato
Desventajas	*No lineal *Salida baja *Requiere referencia *el menos estable * el menos sensible	*Caro *requiere fuente de corriente *Pequeño DR *Baja resistencia *autocalentamiento	*No lineal *Rango limitado de temperaturas *Frágil *requiere fuente de corriente *Autocalentamiento	*T < 200° *Requiere fuente de alimentación *lento *Auto calentamiento *Configuraciones Limitadas [1]

3. Microcontrolador PIC18F46K20

1.1 INTRODUCCIÓN

Los microcontroladores (μ C o MCU por microcontroller unit) consisten en sencillos pero completos computadores digitales integrados, de alta escala de integración. Cuentan con un microprocesador o unidad de procesamiento central (CPU), una memoria para almacenar el código del programa, una memoria para almacenar datos, y otra para el almacenamiento de datos no volátiles, puertos digitales de propósito general, puertos de comunicación y conversores. A diferencia de los microprocesadores, que no son más que circuitos integrados que contienen una unidad central de proceso (CPU), los microcontroladores son unidades autosuficientes y más económicas [3].

Por las características mencionadas y su alta flexibilidad (ya que cambiar las características de control en muchas ocasiones supone únicamente modificar un programa), son muchos los sistemas que incorporan microcontroladores como cerebro para el control de máquinas, componentes de sistemas complejos como aplicaciones industriales de automatización y robótica, domótica, equipos médicos, sistemas de telecomunicación, e incluso dispositivos de la vida diaria como automóviles, hornos de microondas, cámaras, televisores, etc.

3.2 CARACTERÍSTICAS DE LOS MICROCONTROLADORES

Un microcontrolador dispone normalmente de los siguientes componentes:

- Unidad de Procesamiento Central (CPU): Típicamente de 8 bits pero también las hay de 16,32, y hasta 64 bits, la mayoría con arquitectura Harvard, la CPU está conectada a dos memorias por intermedio de dos buses separados, una contiene solamente las instrucciones del programa y la otra solo almacena los datos (Figura 1). Ambos buses son totalmente independientes y pueden ser de distinto tamaño.

- Memoria de Datos: Es una memoria RAM (Random Access Memory) que típicamente puede ser de 1, 2, 4, 8, 16, 32 kilobytes
- Memoria de Programa: Es una memoria ROM (Read-Only Memory), EPROM (Electrically Programmable ROM), EEPROM (Electrically Erasable/Programmable ROM)
- Flash que almacena el código del programa y que típicamente puede ser de 1 kilobyte a varios megabytes.
- Generador del Reloj: Usualmente un cristal de cuarzo de frecuencias que genera una señal oscilatoria de entre 1 a 64 MHz, o también resonadores o circuitos RC.
- Interfaz de Entrada/Salida: Puertos paralelos, seriales (UARTs, Universal Asynchronous
- Receiver/Transmitter, I²C (Inter-Integrated Circuit), Interfaces de Periféricos Seriales (SPIs, Serial Peripheral Interfaces), Red de Area de Controladores (CAN, Controller Area Network), USB (Universal Serial Bus).
- Otras opciones
 - Conversores Análogo-Digitales (A/D, analog-to-digital) para convertir un nivel de voltaje en un cierto pin a un valor digital manipulable por el programa del microcontrolador.
 - Moduladores por Ancho de Pulso (PWM, Pulse-Width Modulation) para generar ondas cuadradas de frecuencia fija pero con ancho de pulso modificable.
 - Temporizadores
 - Comparador analógico
 - Convertidor Digital/analógico
 - Etc [3]

3.3 ELECCIÓN DEL MICROCONTROLADOR

Existe una gran variedad de microcontroladores y a la hora de seleccionar uno a emplear en una aplicación dada hay que tener en cuenta muchos factores, como la documentación, herramientas de desarrollo disponibles, la cantidad de fabricantes que lo producen el coste, y por supuesto de las características del μC , como:

- Procesamiento de datos: Nos puede interesar un dispositivo que sea rápido para, por ejemplo, realizar cálculos críticos en tiempo limitado. O la precisión de los datos a manejar, si no es suficiente con microcontrolador de 8 bits, es necesario acudir a uno de 16,32 o incluso de 64 bits.
- Entrada Salida: Después de hacer un análisis al sistema para identificar la cantidad y el tipo de las señales a controlar, se determinan las necesidades de Entrada/salida.
- Consumo: En algunas aplicaciones como las que se alimentan con batería, este factor es muy crítico a la hora de seleccionar el μC .
- Memoria: Es necesario tener muy claro el tipo y la cantidad de memoria, que la aplicación requiere.
- Ancho de palabra: En este punto hay que seleccionar el μC con el menor ancho de palabra que satisfaga los requisitos de la aplicación y dejar los μC s más potentes a aplicaciones que requieren sus altas prestaciones, y así reducir notablemente los costes.
- Diseño de la placa: La selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos.

En este proyecto se ha tenido en cuenta estos factores y se ha elegido el PIC18F46K20, por su sencillez de manejo (tiene un juego de instrucción reducido), su precio (su coste es comparativamente inferior al de sus competidores), por su velocidad de funcionamiento y por la cantidad de información y las herramientas de desarrollo que proporciona el fabricante (muchas aplicaciones se pueden bajar libremente desde la página web del fabricante: <http://www.microchip.com>).

En el siguiente apartado se va a analizar las características más representativas de este μC .

3.4 μC PIC18F46K20

El μC PIC18F46K20 de Microchip Technology Inc. Es un μC con tecnología nanowatts, que sigue el modelo Harvard, permitiendo a la CPU acceder

simultáneamente a la memoria de instrucciones y la de datos, además se aplica la técnica de segmentación (“pipe-line”) en la ejecución del programa y así la CPU realiza, al mismo tiempo, la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar una instrucción en un ciclo de instrucción (equivale a cuatro ciclos de reloj), excepto las instrucciones de salto que necesitan dos ciclos, al no conocer la dirección de salto hasta resolver la condición de bifurcación.

Las instrucciones en este μC son ortogonales (Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino) y son de la misma longitud (16 bits). El procesador es de juego de instrucciones reducido con casi 60 instrucciones máquina.

3.4.1 Organización de la memoria

La memoria de datos de este microcontrolador es de tipo SRAM (3963 Bytes), donde residen los Registros Específicos (SFR) que se utilizan por la CPU y por los módulos periféricos para controlar el funcionamiento deseado del dispositivo y los Registros de Propósito General (GPR) que están disponibles para su uso por todas las instrucciones. La RAM del PIC18F46K20 se halla dividida en 16 bancos de 256 bytes cada uno (FIGURA-x). El μC contiene también una memoria EEPROM (1024 bytes), memoria de sólo lectura programable y borrable eléctricamente donde, opcionalmente, se pueden almacenar datos que no se pierden al desconectar la alimentación [4].

Tabla 3.1: resumen de las características del μC .

Device	Program Memory		Data Memory		I/O ⁽¹⁾	10-bit A/D (ch) ⁽²⁾	CCP/ ECCP (PWM)	MSSP		EUSART	Comp.	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I ² C™			
PIC18F23K20	8K	4096	512	256	25	11	1/1	Y	Y	1	2	1/3
PIC18F24K20	16K	8192	768	256	25	11	1/1	Y	Y	1	2	1/3
PIC18F25K20	32K	16384	1536	256	25	11	1/1	Y	Y	1	2	1/3
PIC18F26K20	64k	32768	3936	1024	25	11	1/1	Y	Y	1	2	1/3
PIC18F43K20	8K	4096	512	256	36	14	1/1	Y	Y	1	2	1/3
PIC18F44K20	16K	8192	768	256	36	14	1/1	Y	Y	1	2	1/3
PIC18F45K20	32K	16384	1536	256	36	14	1/1	Y	Y	1	2	1/3
PIC18F46K20	64k	32768	3936	1024	36	14	1/1	Y	Y	1	2	1/3

Note 1: One pin is input only.

2: Channel count includes internal fixed voltage reference channel.

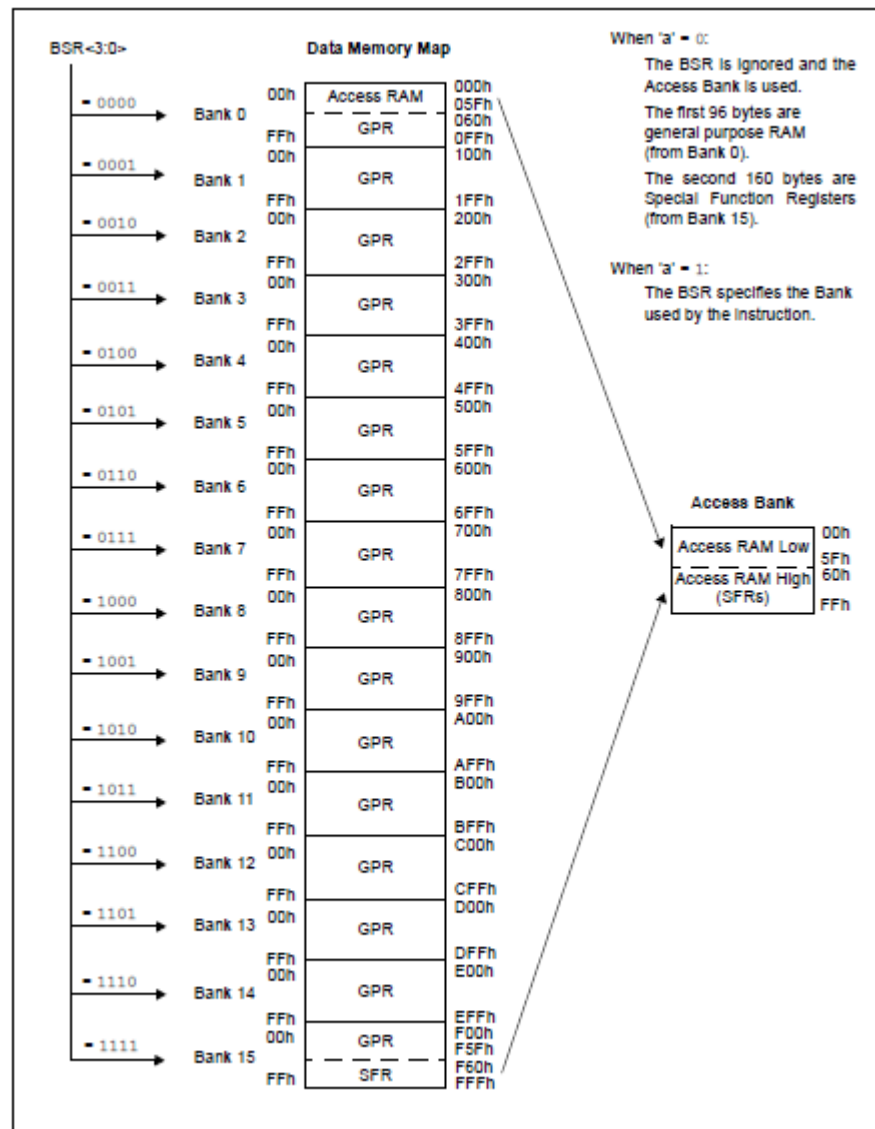


Figura 3.1: Mapa de la memoria SRAM del PIC18F26K20/46K20

La de programa es de tipo Flash (con las mismas características que la EEPROM pero con menor consumo de energía y mayor capacidad de almacenamiento) y es de 64 KB de tamaño con lo cual puede almacenar hasta 32768 instrucciones (Tabla 3.1).

El contador de programa (Program Counter o PC), también llamado Puntero de instrucciones (Instruction Pointer), especifica la dirección de la instrucción para ejecutar. El PC es de 21 bits y se encuentra en tres diferentes registros de 8 bits. El byte menos significativo, llamado PCL es de lectura y escritura, mientras que los bytes <15-08> (llamado PCH) y <20-16> (llamado UCP) no son accesibles

directamente y se actualizan a través de los registros PCLATH y PCLATU respectivamente. El contador de programa es incrementado automáticamente en cada ciclo de instrucción de tal manera que las instrucciones son leídas en secuencia desde la memoria. Ciertas instrucciones, tales como las bifurcaciones y las llamadas y retornos de subrutinas, interrumpen la secuencia al colocar un nuevo valor en el contador de programa [4 y 5].

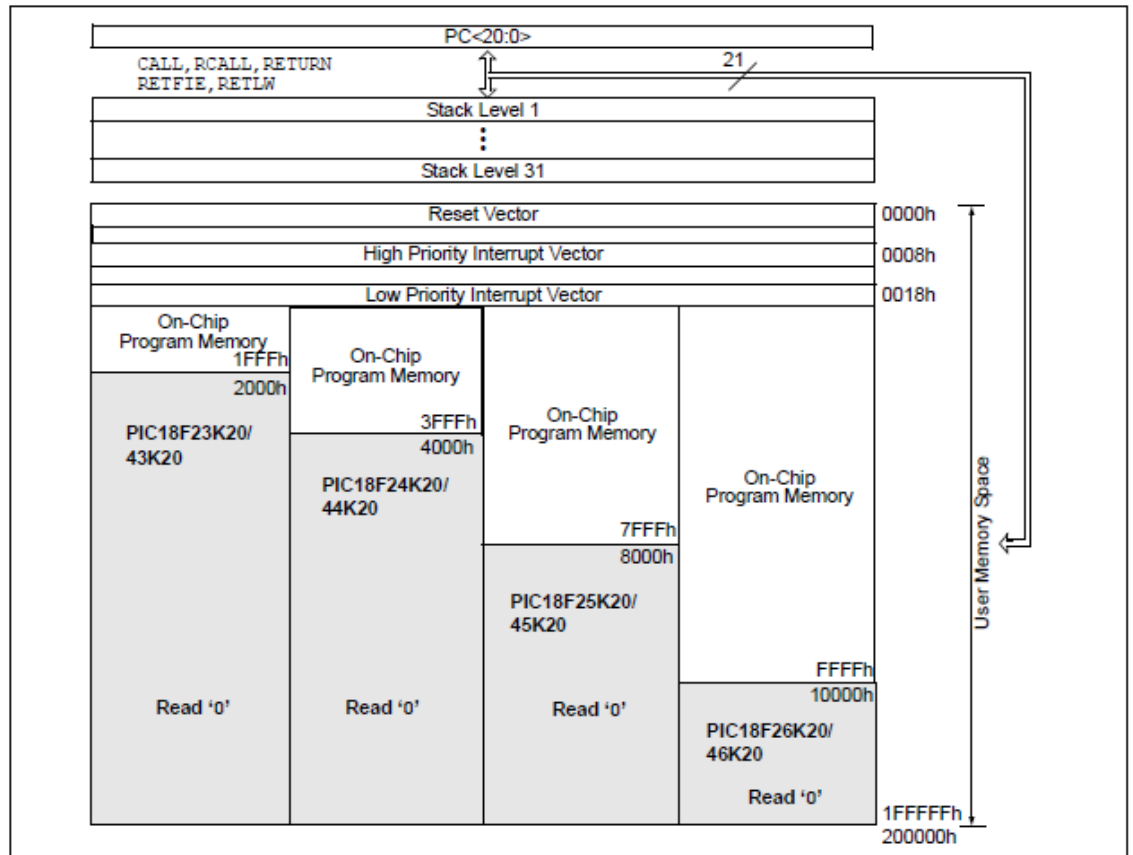


Figura 3.2: Mapa de memoria de programa del PIC18F2XK20/4XK20

Cuando ocurre un Reset, el contador de programa (PC) apunta a la dirección 0000h, y el micro se inicia nuevamente. Por esta razón, en la primera dirección del programa se debe escribir todo lo relacionado con la iniciación del mismo.

Ahora, si ocurre una interrupción el contador de programa (PC) apunta a la dirección 0008h o 0018h, dependiendo de la prioridad de la interrupción, entonces ahí se escribe la programación necesaria para atender dicha interrupción.

La pila o Stack consta de 31 posiciones (o niveles), es una estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO (Last In First Out, último en entrar, primero en salir) que permite almacenar y recuperar el valor del PC cuando se producen llamadas (CALL) o interrupciones.

3.4.2 Puertos de entrada/salida

Como se puede ver en la figura 3.3 es un μC con 44 pines, 36 de ellos son pines de entrada salida, que se dividen en cinco puertos (RA0-RA7, RB0-RB7, RC0-RC7, RD0-RD7, RE0-RE3).

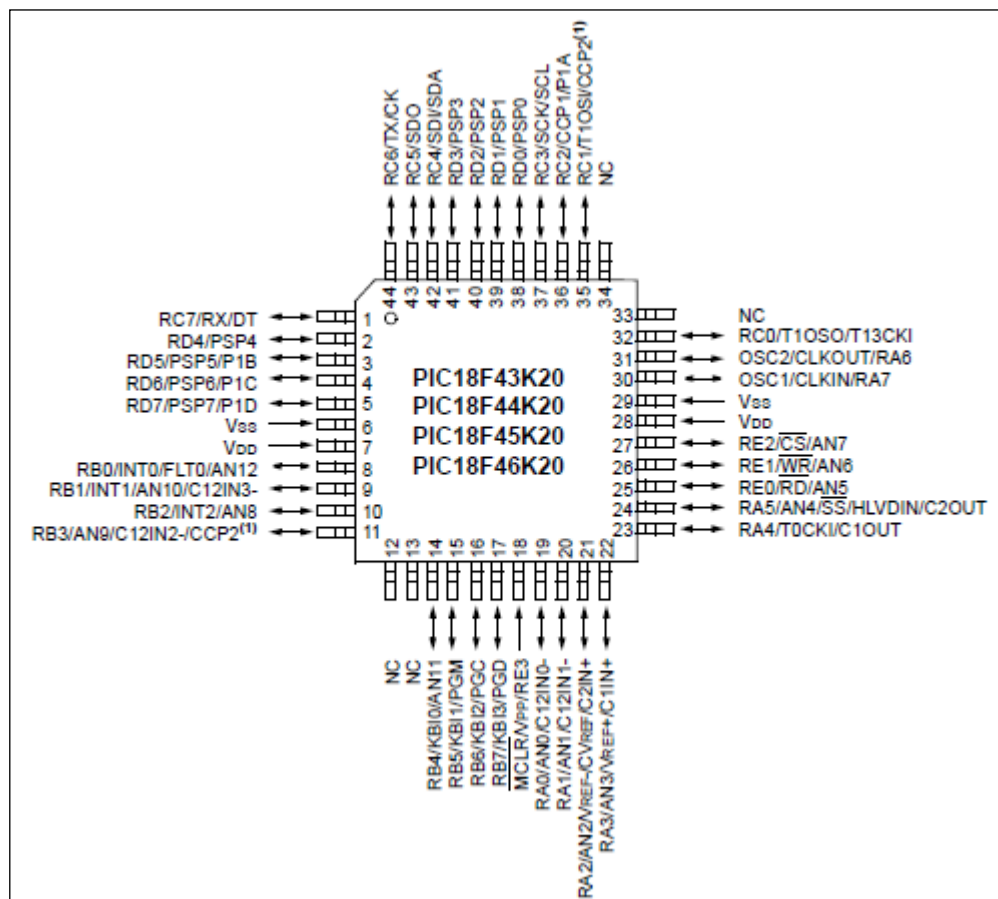


Figura 3.3: Pines del μC PIC18F46K20.

Algunos pines de los puertos de I/O se multiplexan con otra función de las características periféricas en el dispositivo. Cada puerto tiene tres registros para su operación:

- Registro TRIS

- Registro PORT
- Registro LAT

El registro TRIS es el registro de dirección de datos. Activando un bit del registro TRIS se convierte el pin del puerto en una entrada. Desactivando un bit del registro TRIS se convierte el pin del puerto en una salida. El registro LAT es útil para leer, modificar o escribir el valor que los pines I/O conducen. PORT es un puerto bidireccional (entre 3 y 8 bits) [4 y 5].

3.4.2.A Puerto A

El puerto A esta compuesto por 8 pines (RA0-RA7), donde cada pin esta multiplexado con otras funciones. Por ejemplo el pin RA4 puede ser configurado como un pin de entrada/salida, como entrada del temporizador/contador o como la salida 1 del comparador para convertirse en el pin de RA4/T0CKI/C1OUT. Cuando este pin se programa como entrada digital, funciona como un disparador de Schmitt (Schmitt trigger, ST), esto quiere decir que puede reconocer señales un poco distorsionadas y llevarlas a niveles lógicos (cero y cinco voltios). Cuando se usa como salida digital se comporta como colector abierto y no puede manejar cargas como fuente, sólo en el modo sumidero.

3.4.2.B Puerto B

Es un puerto de 8 pines, cuatro de ellos (RB7:RB4) se pueden configurar individualmente como entradas de interrupciones por cambio de estado, se comparan con el valor de la última lectura del Puerto B, en caso de que no haya coincidencia se genera la interrupción de cambio del Puerto B con el bit indicador, RBIF (registro INTCON).

3.4.2.C Puerto C

También es de 8 pines todos multiplexados, donde, por ejemplo, los pines RC4-RC6 pueden configurarse como serial porte además de ser entrada/salida digital convencional.

3.4.2.D Puerto D

Todos los pines (8 líneas) del puerto D se implementan con buffers de entrada de disparador de Schmitt. Cada pin se puede configurar individualmente como entrada o salida. Se puede configurar como un puerto 8 bits del microprocesador (puerto auxiliar paralelo) activando el bit PSPMODE (TRISE <4>) En este modo, los buffers de entrada son TTL.

3.4.2.E Puerto E

El puerto E, a diferencia de todos los puertos anteriores, tiene solo 4 bits. Tres de estos bits (RE0/RD/AN5, RE1/WR/AN6 y RE2/CS/ AN7) se pueden configurar individualmente como entradas o salidas. El ultimo es un pin (MCLR/VPP/RE3) de entrada solamente, está controlado por el bit de configuración MCLRE y no tiene el bit correspondiente en los registros TRIS y LAT.

3.4.2.F Corriente máxima

La máxima corriente que puede suministrar una línea programada como salida o entradas es de 25 miliamperios, pero si se utiliza todas las líneas de un puerto programadas como salida, no deberá exceder de 200mA para todo el puerto. Si las programamos como entradas (Sumidero), la corriente máxima que puede manejar una sola línea es de 25 mA. Para el caso de programar todas las como entrada, la máxima es de 185 mA. En caso de querer utilizar periféricos que manejen mayor cantidad de corriente de la especificada, habrá que aplicar un circuito acoplador como por ejemplo los buffers, transistores que se encarguen de controlar la corriente, etc.

3.4.3 Temporizadores

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

El PIC18F46K20 tiene 4 temporizador, a continuación se describen las características más importantes de cada uno de ellos.

3.4.3.A Temporizador0.

EL Timer0 puede operar como temporizador o contador (8 o 16 bits), según lo que indica el bit T0CS, si este está a 0 el TIMER0 trabaja como temporizador, al contrario, si está a 1 trabaja como contador.

Se genera Interrupción por desbordamiento de FFh a 00h en modo 8-bit y FFFFh a 0000h en el modo 16-bit. También se da la posibilidad de seleccionar la fuente, externa o interna, y el flanco del reloj.

El registro T0CON controla todos los aspectos del funcionamiento del módulo, incluyendo la selección de la preescala, ésta puede variar de 1:2 hasta 1:256, según lo que indican los bits T0CON<2-0>. Un diagrama de bloques simplificado del módulo Timer0 en modo de 8 y 16bits se muestra en la Figura 3.4 [4].

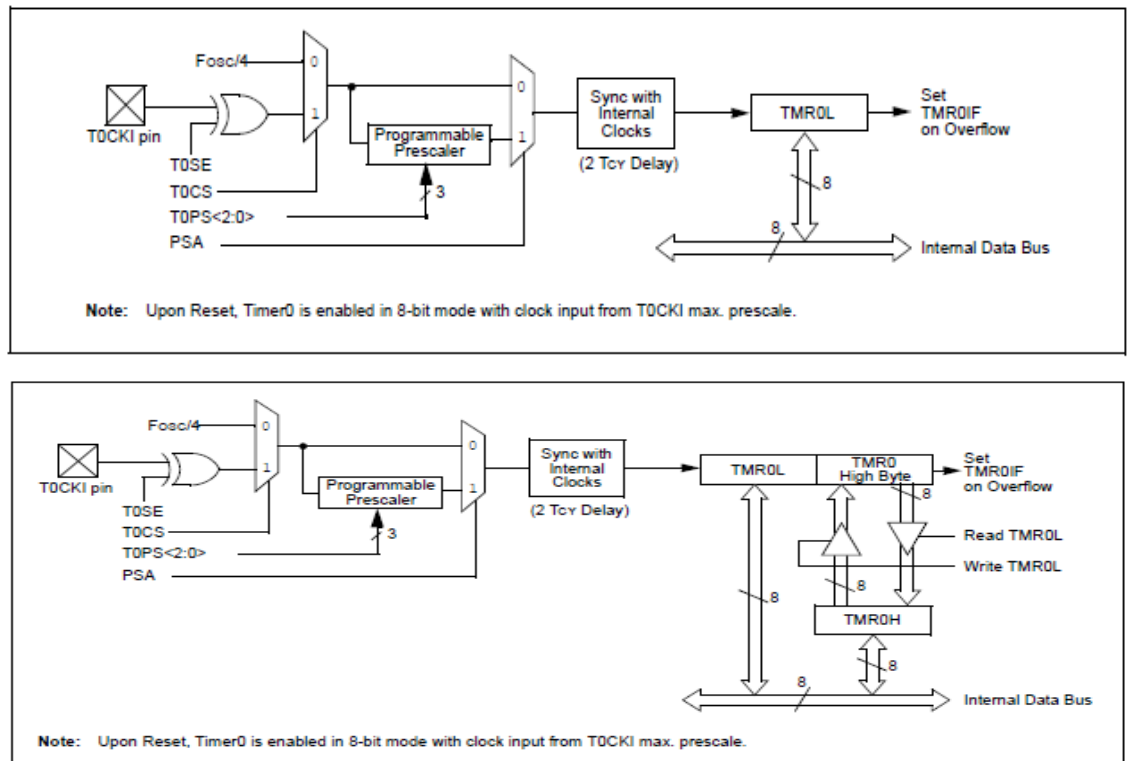


Figura 3.4: Diagrama de bloques del TIMER0 (Modo 8y 16 bits)

En el caso de seleccionar el modo contador y reloj externo el TIMER0 se incrementará, en cada flanco de subida o baja del pin RA4/T0CKI.

Un contador de 8 bits está disponible como prescalador para el módulo Timer0. EL valor de éste es fijado por el PSA y los bits T0PS <2:0> que determinan la asignación de la preescala y la relación de preescala. El bit PSA desactivado asignará el prescalador al módulo Timer0 [4].

3.4.3.B Temporizador1

Tiene las mismas características que el Timer0, pero con una sola diferencia, y es que solo puede ser temporizador o contador síncrono o asíncrono, de 16 bits (dos registros de 8 bits; TMR1H y TMR1L).

El modo de funcionamiento de este modulo se controla por los bits del registro T1CON, por ejemplo el bit T1OSCEN activa el oscilador del Timer1, el bit de control TMR1ON (T1CON<0>) puede activar o desactivar el Timer, el bit TMR1CS (T1CON<1>) selecciona el reloj. Cuando TMR1CS = 0, Timer1 incrementa cada ciclo de la instrucción. Cuando TMR1CS = 1, Timer1 incrementa en cada flanco de subida de la entrada de reloj externa o del oscilador del Timer1, si está activado. Cuando se activa el oscilador del Timer1 (T1OSCEN habilitado), los pines RC1/T1OSI y RC0/T1OSO/T1CKI se convierten en entradas. Timer1 también tiene una entrada interna de Reset (puede ser generado por el módulo CCP)

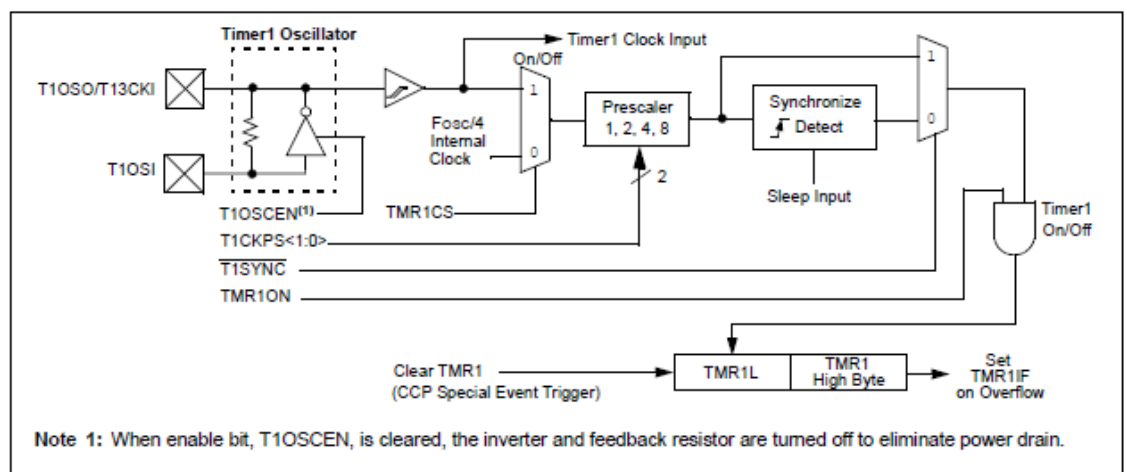


Figura 3.5: Diagrama de bloques del Timer1

El par de registros (TMR1H:TMR1L) se incrementan desde 0000h a FFFFh. Cuando se llega al desbordamiento se genera una interrupción de TMR1, si está

permitida, que está asociado al bit indicador TMR1IF (PIR1<0 >). Esta interrupción se puede habilitar o deshabilitar, activando/desactivando el bit TMR1IE (PIE1<0 >).

El Timer1 incorpora un oscilador de baja potencia entre los pines T1OSI y T1OSO, se activa mediante el bit T1OSCEN (T1CON<3 >) [4].

3.4.3.C Temporizador2

El Timer2 tiene las siguientes características:

- Temporizador de 8 bits (registro TMR2)
- Registro de período de 8 bits (PR2)
- Se puede leer y escribir
- Preescalador programable por software (1:1, 1:4, 1:16)
- Postescalador programable por software (1:1 a 1:16)
- Interrupción en la coincidencia de los registros TMR2 y PR2
- Uso opcional por el módulo de SSP de la salida TMR2 para generar cambio del reloj
- Se puede apagar con poner a cero el bit de control TMR2CON del registro T2CON, para reducir el consumo de energía.

Un diagrama de bloques simplificado del módulo se muestra en la Figura 14-1.

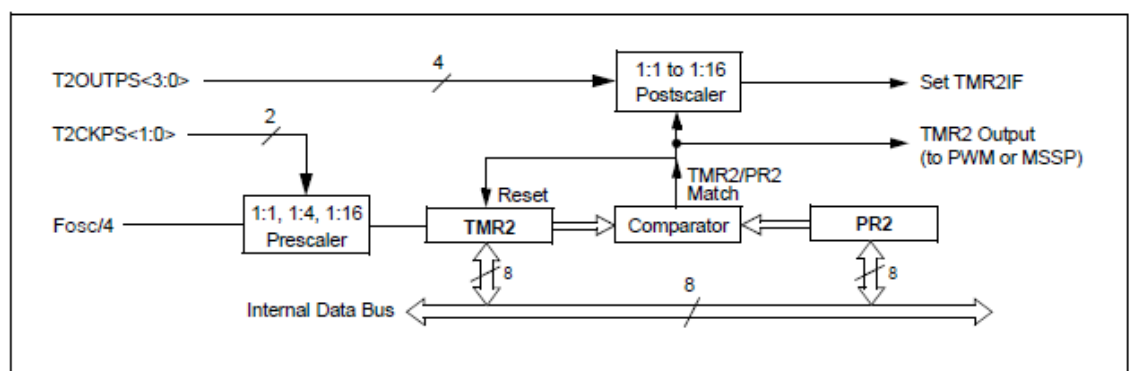


Figura 3.6: Diagrama de bloques del TIMER2

En funcionamiento normal el TMR2 es incrementado desde 00h en cada ciclo de reloj (Fosc/4), teniendo la opción de preescala de 1:1, 1:4, 1:16, seleccionado el valor correspondiente en los bits T2CKPS1:T2CKPS0 del registro de control

T2CON. En cada ciclo de reloj se compara el registro temporizador TMR2 con el registro de periodo PR2. En caso de coincidencia se genera una interrupción. Se puede utilizar el Timer2, también, como base de tiempo para el modo de PWM del módulo CCP.

3.4.3.D Temporizador3

Es un temporizador con las mismas características que el Timer1, ya que comparten las mismas características:

- Temporizador/contador de 16-bit (dos registros de 8 bits; TMR3H y TMR3L)
- Los dos registros se pueden leer y escribir
- Selección de reloj externo e interno
- Interrupción por overflow de FFFFh a 0000h
- Reset del módulo CCP

Un diagrama de bloques de este Timer se muestra en la siguiente figura:

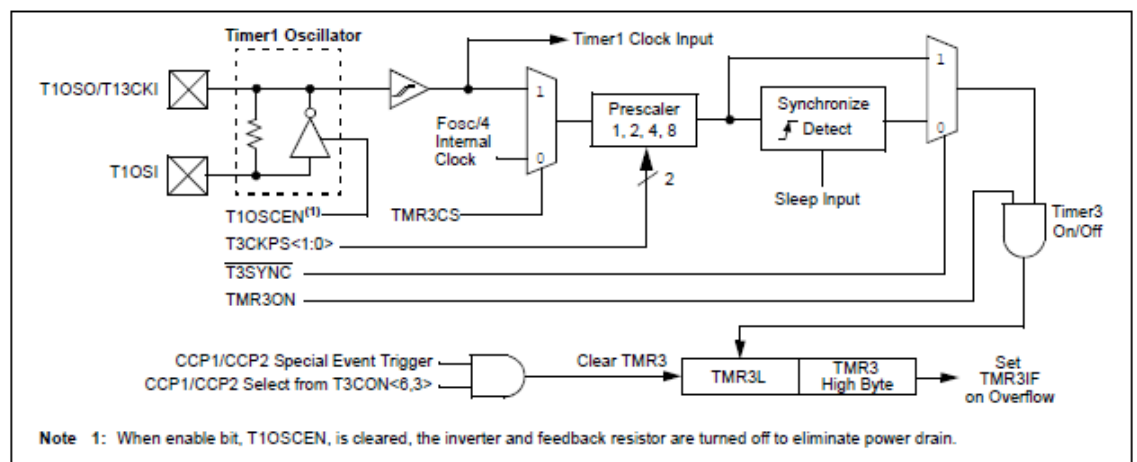


Figura 3.7: Diagrama de bloques del Timer3

Este módulo puede operar como temporizador o contador síncrono y asíncrono. El modo de funcionamiento se selecciona con el bit de control TMR3CS del registro T3CON (registro de control de este módulo). Cuando éste está a cero Timer3 se incrementa en cada ciclo de instrucción y cuando está a uno el Timer3 se incrementa en cada flanco de subida de la entrada de reloj externa o del oscilador del Timer1, si

está activado. Cuando se activa el oscilador del Timer1 (T1OSCEN habilitado), los pines RC1/T1OSI y RC0/T1OSO/T1CKI se convierten en entradas. Timer3 también tiene una entrada interna de Reset (puede ser generado por el módulo CCP)

El par de registros TMR3 (TMR3H:TMR3L) se incrementan desde 0000h a FFFFh, generando por desbordamiento una interrupción de TMR3, si ésta está habilitada. Esta interrupción está asociado al bit indicador TMR3IF (PIR2<1 >), y se puede habilitar/deshabilitar, activando/desactivando el bit TMR3IE (PIE2<1 >).

Se puede resetear el Timer3 usando el disparo del modulo CCP, si está configurado en modo comparación para generar un evento por disparo especial (CCP1M3:CCP1M0 = 1011). Y para que se cumpla lo anterior el Timer3 se debe configurar como temporizador o contador síncrono. En el caso de que se configura como contador asíncrono no se puede garantizar el Reset del Timer3 [4].

3.4.4 Configuraciones del oscilador

El oscilador tiene una amplia variedad de fuentes de reloj y una función de selección que le permiten ser utilizado en una amplia gama de aplicaciones, maximizando el rendimiento y minimizando el consumo de energía. La figura 3.8 muestra un diagrama de bloques del oscilador.

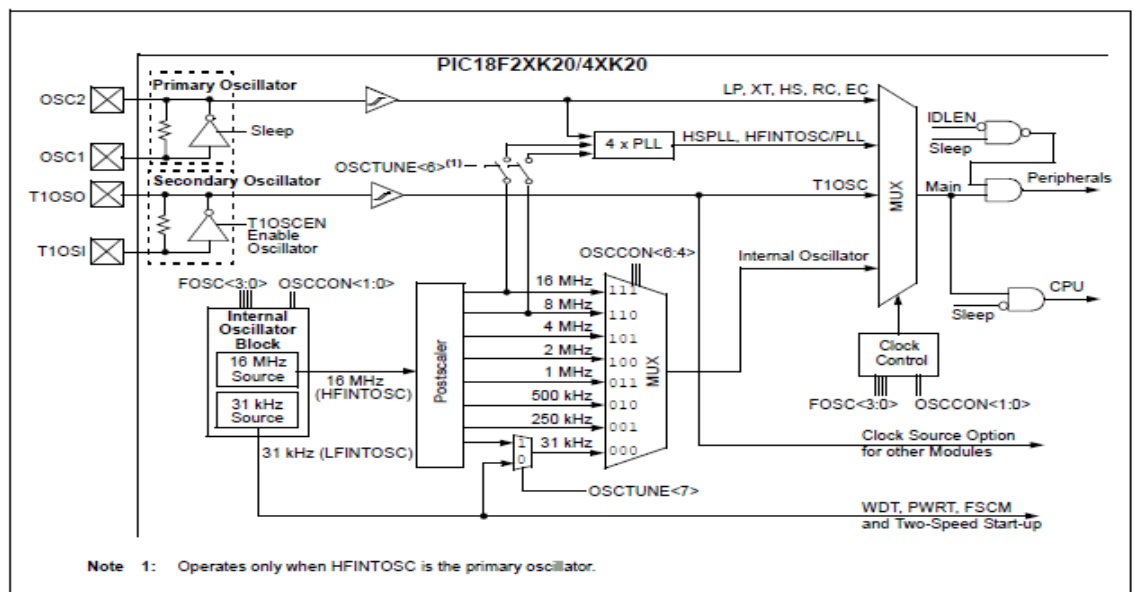


Figura 3.8: Diagrama de bloques del oscilador

EL PIC18F46K20 se puede configurar para operar en uno de los 10 modos. Para elegir uno de ellos se deben programar 4 bits de configuración FOSC<3:0>:

1. LP, cristal de baja potencia
2. XT, cristal/ resonador
3. HS, cristal/ resonador de alta velocidad
4. HSPLL, cristal/ resonador de alta velocidad con PLL activado
5. RC, resistencia/capacidad externa con salida de FOSC/4 en RA6
6. RCIO, resistencia/capacidad externo con pin de entrada/salida (RA6) habilitado
7. INTOSC, Oscilador interno con FOSC / 4 en la salida RA6 y E / S en RA7
8. INTOSCIO, Oscilador interno con los pines RA6 y RA7 configurados como E/S.
9. EC, reloj externo
10. ECIO, reloj externo con pin de entrada/salida habilitado RA6

El control del oscilador se lleva a cabo con el registro OSCON. En los modelos de oscilador LP, XT, HS o HS+PLL, se necesita un cristal o resonador cerámico, que debe ser conectado a los pines OSC1 y OSC2 para establecer la oscilación. En el modo oscilador RC La frecuencia es una función del voltaje de fuente, de la resistencia (REXT) y de los valores del condensador (CEXT) y de la temperatura de funcionamiento. Se debe también considerar la variación debido a la tolerancia de los componentes resistivos y capacitivos externos usados. En este la frecuencia del oscilador dividida por 4 está disponible en el pin OSC2. Esta señal puede ser utilizada para propósitos de prueba o sincronización de otra lógica.

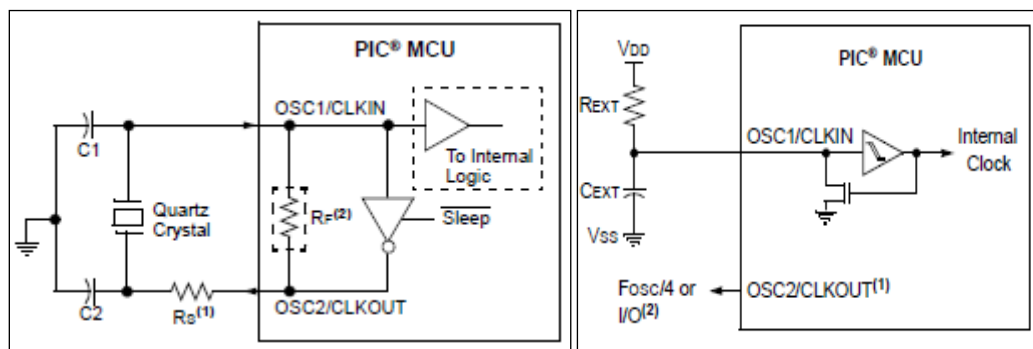


Figura 3.9: Resonador cerámico y Oscilador RC

Los modos del oscilador EC y ECIO requieren una fuente externa de reloj para ser conectada con el pin OSC1. En el modo del oscilador EC, también la frecuencia del oscilador dividida por 4 está disponible en el OSC2.

El modo del oscilador de ECIO funciona como el modo de la EC, excepto que el pin OSC2 se convierte en un pin adicional de los generales de I/O.

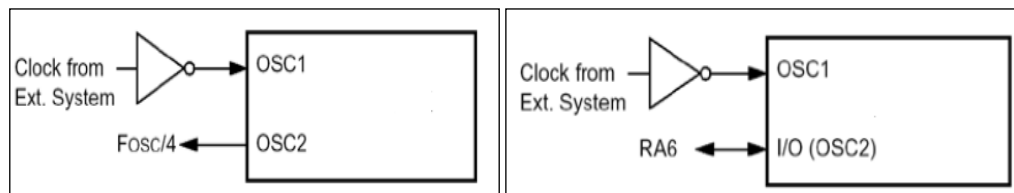


Figura 3.10: Oscilador EC y ECIO

Un circuito Phase Locked Loop (PLL) se proporciona como opción programable para los usuarios que desean multiplicar la frecuencia de la señal entrante del cristal oscilador por 4. Para una frecuencia de reloj de entrada de 10 MHz, la frecuencia de reloj interna será multiplicada a 40 MHz. El PLL puede ser habilitado solamente cuando los bits de la configuración del oscilador se programan para el modo de HS [4].

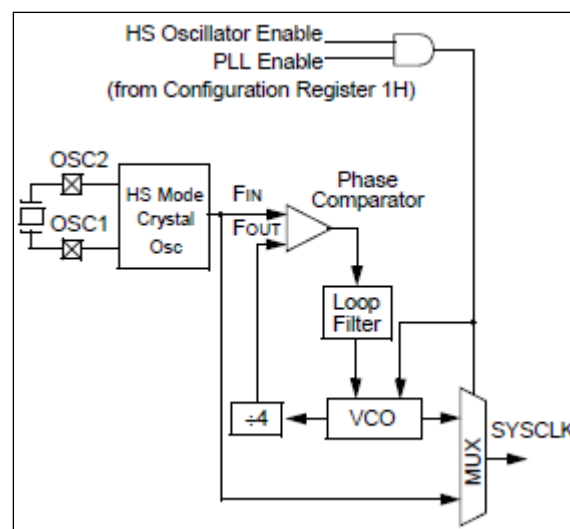


Figura 3.11: Diagrama de bloques del PLL

3.4.5 Reset

El PIC18FXXX diferencia entre en varias clases de Reset:

- a) Power-on Reset (POR)
- b) MCLR Reset durante operación normal
- c) MCLR Reset durante SLEEP
- d) Watchdog Timer (WDT) Reset (durante la ejecución)
- e) Programmable Brown-out Reset (BOR)
- f) RESET Instrucción
- g) Stack Full Reset
- h) Stack Underflow Reset

La mayoría de los registros no se ven afectados por un Reset. Su estado es desconocido para POR y permanece sin cambios para el resto de los Resets. Los otros registros se fuerzan a un estado de Reset en Power-on Reset, MCLR, WDT, Brownout Reset, MCLR Reset durante SLEEP y por la instrucción del Reset.

La mayoría de los registros no se ven afectados por un wake-up de WDT, puesto que esto se ve como la recuperación de la operación normal. Los bits de estado del registro RCON, RI, TO, PD, POR y BOR, se activan o se desactivan de forma diferente en diversas situaciones de Reset. Estos bits se utilizan en software para determinar la naturaleza del Reset [4 y 5].

3.4.6 Módulos Captura/Comparación/PWM

El microcontrolador PIC18F46K20 dispone de dos módulos de captura, comparación y PWM (modulación de anchura de pulsos), denominados CCP1 y CCP2. Cada uno de estos módulos contiene un registro de 16 bits que puede operar como:

- Registro de captura de 16 bits.
- Registro de comparación de 16 bits.
- Registro de control del ciclo de trabajo en la modulación de anchura de pulsos.

Los dos módulos son idénticos en su modo de funcionamiento, con la excepción del modo de disparo especial. Seguidamente se describe el funcionamiento de CCP1, entiendo que CCP2 funciona igual, excepto en el modo de disparo especial.

El registro de Captura/Comparación/PWM de módulo 1(CCP1) es de 16 bits y está compuesto de dos registros de 8 bits: CCPR1L (byte bajo) y CCPR1H (byte alto). El registro de control del CCP1 posee los bits de control de este módulo.

El modo de captura especial es generado cuando el contenido del TIMER1 y CCP1 poseen el mismo valor, provocando que este el reset del Timer1.

3.4.6.A Modo de captura

En este modo CCPR1H:CCPR1L, capturan el valor de que posee TMR1 o de TMR3, Tabla VI, cuando ocurre alguno de los eventos siguientes en el pin RC2/CCP1:

- Cada flanco de bajada
- Cada flanco de subida
- Cada 4 flancos de subida
- Cada 16 flancos de subida

Tabla 3.2: Recursos CCP

CCP/ECCP Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2

El evento es seleccionado por los bits de control CCP1M3:CCP1M0 (CCP1CON<3:0 >). Cuando se hace una captura, se activa el bit indicador de petición de interrupción CCP1IF (PIR1<2 >). Si ocurre otra captura antes de que el valor en el registro CCPR1 se lea, el valor anteriormente capturado es sobrescrito por el nuevo valor capturado.

En modo captura, el pin RC2/CCP1 debe ser configurado como entrada activando el bit TRISC<2>.

Cuando se trabaja en el modo captura, el Timer1/Timer3 debe funcionar en modo temporizador o contador síncrono. En modo contador asíncrono puede que la operación de captura no funcione. El temporizador que se utilizará con cada módulo CCP se selecciona en el registro de T3CON.

Cuando se van a cambiar las condiciones del modo de la captura, es conveniente desactivar el modulo CCP para evitar falsas capturas [4 y 6].

El preescalador tiene 4 modos de funcionamiento especificados por los bits CCP1M3:CCP1M0. Siempre que el módulo CCP esté apagado o no esté en modo captura, el contador del preescalador estará desactivado. Esto significa que cualquier Reset despejará el contador del preescalador. Cambiar de un preescalador de captura a otro puede generar una interrupción [6].

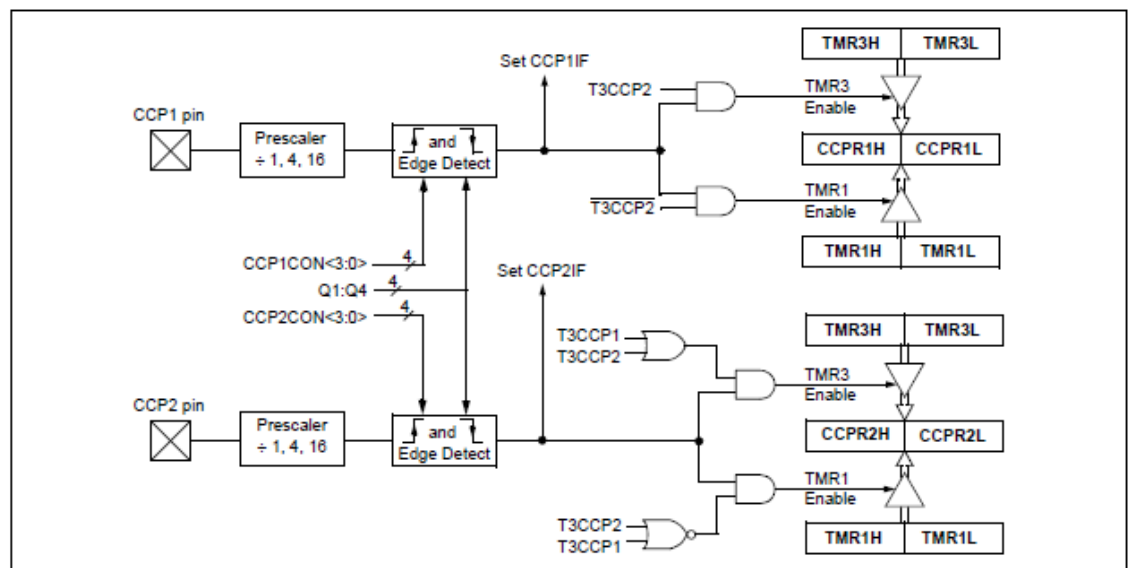


Figura 3.12: Diagrama de bloques del modo captura

3.4.6.B Modo de comparación

En el modo de comparación, los 16 bits del registro CCPR1 (CCPR2), se comparan constantemente con el valor del registro TMR1, o el valor del registro

TMR3. Cuando se produce una igualdad, se puede provocar una de las siguientes acciones, en el pin RC2/CCPX (debe estar configurado como salida):

- Se activa
- Se desactiva
- Cambia a la situación opuesta (Activa a Desactiva o viceversa)
- Permanece invariable

La acción en el pin se basa en el valor de los bits de selección de modo (CCPxM <3:0>). Además cuando se produce la igualdad, entre los registros, el flag de interrupción CCP1IF se pone a uno.

Timer1 y/o Timer3 deben funcionar en modo temporizador o contador síncrono si el módulo de CCP está utilizando la característica de comparación. En modo asíncrono, la operación de comparación puede no funcionar.

En el disparo por evento especial, el módulo CCP1 pone a cero los dos registros de TMR1, y el CCPR1 funciona como un registro de 16 bits con periodo programable, capaz de generar periódicamente interrupciones. Además, el disparador por evento especial CCP2, comenzará una conversión A/D si el módulo A/D está habilitado.

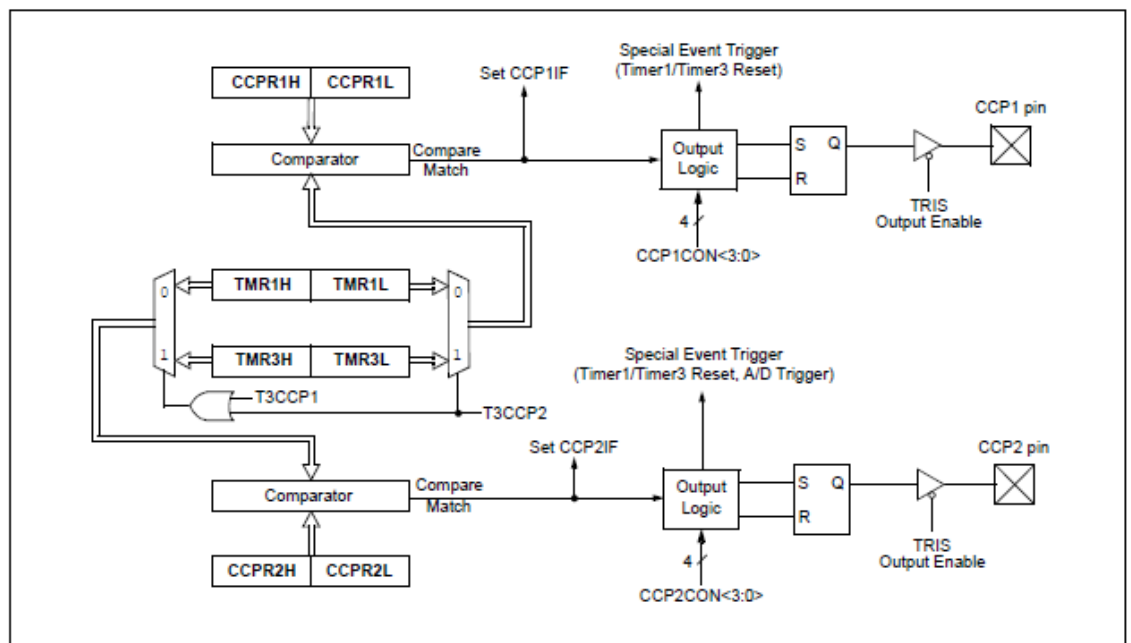


Figura 3.13: Diagrama de bloques del modo de comparación

3.4.6.C Modo de modulación de anchura de pulsos PWM

En modo de modulación de la anchura de pulso (PWM), el pin CCP1 produce pulsos de nivel alto con anchura variable, y una resolución de hasta 10 bits. Puesto que el pin CCP1 se multiplexa con los datos del Puerto C, se debe poner a cero el bit TRISC<2> para hacer que el pin CCP1 se configura como salida.

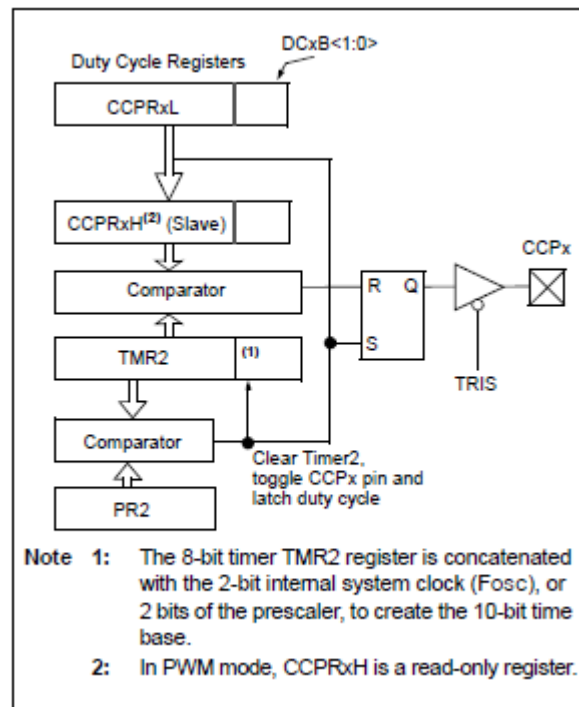


Figura 3.14: diagrama de bloques simplificado PWM

El período de PWM se especifica mediante el registro PR2 de Timer2, y se puede calcular con la fórmula de la ecuación 3.1:

$$PWM\ Period = [PR2 + 1] * 4 * T_{osc} * (TMR2\ prescal\ value) \quad (3.1)$$

Cuando el valor almacenado en el registro PR2 coincide con el de TMR2, pasa lo siguiente:

- Se resetea el TMR2
- El pin CCP1 se habilita (excepto si el PWM duty cycle = 0%, el pin CCP1 no se habilitará)
- El PWM duty cycle se asocia a los registros CCPR1L:CCPR1H

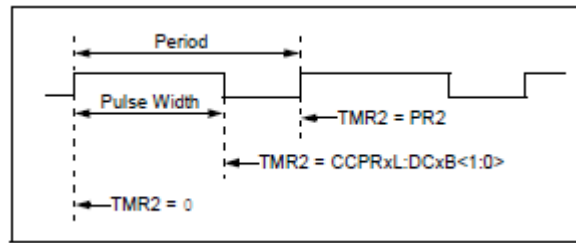


Figura 3.15: Salida PWM

El PWM duty cycle (el tiempo en el que la salida esta a nivel alto) depende del valor cargado en el registro CCPR1L y los bits CCP1CON<5:4 > y su valor viene dado por la fórmula:

$$\text{PWM duty cycle} = (\text{CCPR1L: CCP1CON} < 5:4 >) * \text{TOSC} * (\text{TMR2 prescale value})$$

La resolución determina el número de ciclos de trabajo disponibles para un período determinado [4].

$$\text{Resolution} = \frac{\log [4(\text{PR2} + 1)]}{\log (2)} \text{ bits} \quad (3.2)$$

La configuración del módulo CCP para la operación PWM conlleva los siguientes pasos:

1. Fijar el período del modo PWM escribiendo al registro PR2.
2. Fijar el PWM duty cycle escribiendo en el registro CCPR1L y los bits CCP1CON<5:4 >.
3. Habilitar el pin CCP1 como una salida reseteando el bit TRISC<2 >.
4. Fijar el valor del preescalador del TMR2 y habilitar el Timer2 [4 y 6]

3.4.7 Módulo Convertidor Analógico a Digital (A/D)

La conversión analógica-digital (CAD) consiste en la transformación de señales analógicas en señales digitales (valor binario), con el propósito de facilitar su procesamiento (codificación, compresión, etc.) y hacer la señal resultante más inmune al ruido y otras interferencias a las que son más sensibles las señales analógicas. A continuación describimos las características más importantes del módulo A/D de este μC [3].

Este módulo usa 10 bits para codificar la señal analógica en la entrada de uno de los 12 pines disponibles para esta función. El resultado se obtiene después de aproximaciones sucesivas y es almacenado en los registros (ADRESH y ADRESL). Al final de cada conversión se genera una interrupción, activando el bit indicador de interrupción A/D, ADIF.

La tensión analógica de referencia puede seleccionarse por software como la tensión de alimentación positiva y negativa (VDD y VSS), o la tensión en los pines RA3/AN3/ VREF+ y RA2/AN2/VREF-. El convertidor A/D tiene la característica de poder funcionar mientras que el dispositivo está en modo SLEEP. Para eso, el reloj de conversión A/D se debe derivar del oscilador interno RC. El diagrama de bloques del módulo A/D se muestra en la figura 42 [4].

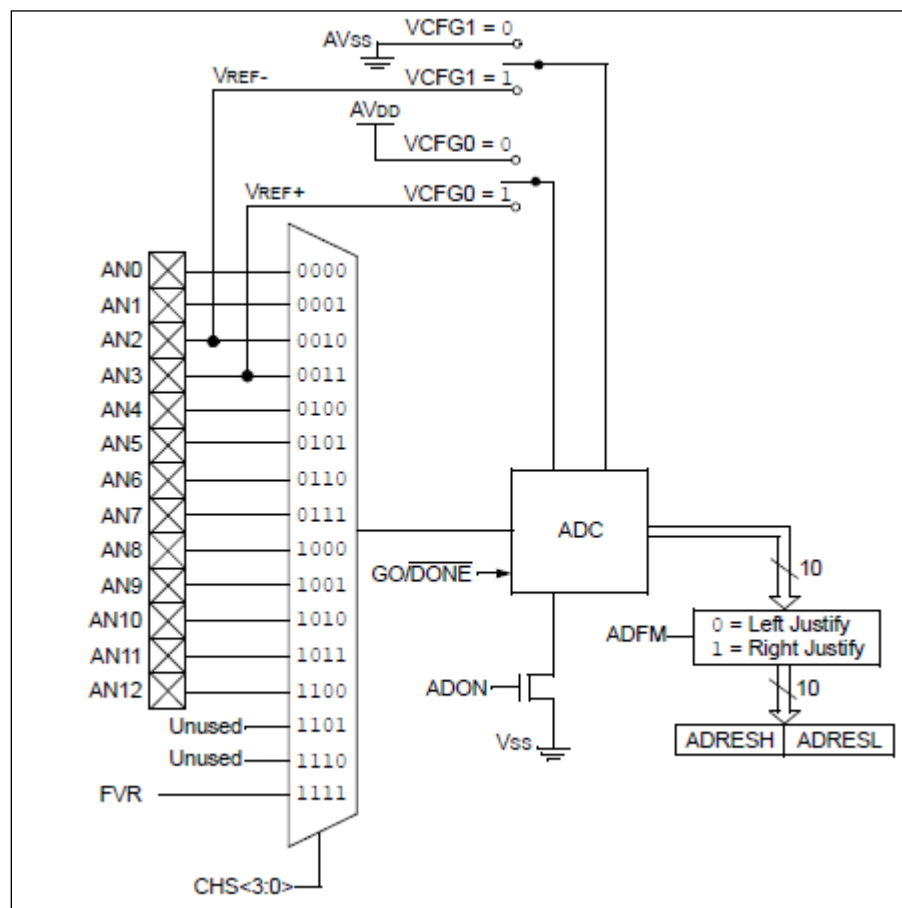


Figura 3.16: Diagrama de bloques ADC

Para el control de este módulo se usan los registros ADCON0, ADCON1 y ADCON2. Después de que el módulo A/D se haya configurado según lo deseado, el

canal seleccionado debe ser adquirido antes de que comience la conversión. Los canales de la entrada analógica deben tener sus bits correspondientes del registro TRIS seleccionados como entrada.

Los pasos necesarios para la conversión se comentan más adelante (desarrollo del proyecto).

La conversión A/D puede comenzar por el disparador por evento especial del módulo CCP2. Esto requiere que los bits CCP2M3:CCP2M0 (CCP2CON<3:0 >) estén programados como 1011 y que el módulo A/D esté habilitado (bit ADON activado). Cuando ocurre el disparo, el bit GO/ DONE se activa, comenzando la conversión A/D, y el Timer1 (o Timer3) se resetea a cero. Si el módulo A/D no se habilita (ADON está desactivado), el disparador de evento especial será ignorado por el módulo A/D, pero reseteará el Timer1 (o Timer3).

3.4.8 Módulo comparador

Dispone de dos comparadores analógicos C1 y C2 cuyas entradas pueden ser seleccionadas entre los pines RA0 a RA3 y cuyas salidas pueden utilizarse para lectura digital interna o bien conectarse a los pines de salida: RA4 y RA5. También podría ser posible emplear un nivel de tensión configurable y generado por otro bloque, el denominado Módulo de Referencia de Tensión.

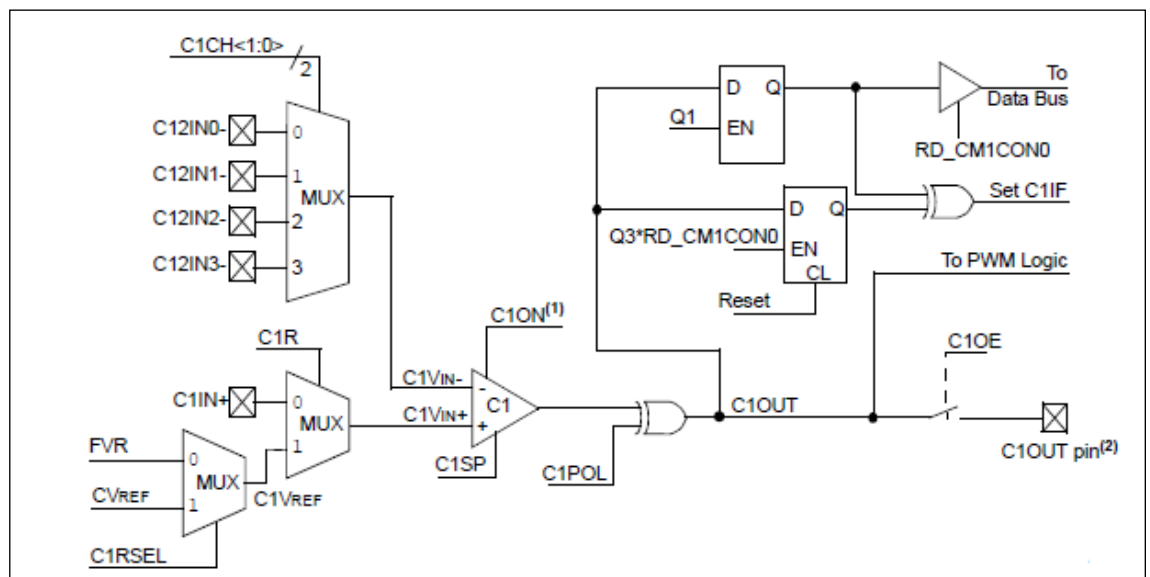


Figura 3.17: Diagrama de bloques simplificada del comparador C1

Los dos comparadores C1 y C2 se controlan a través de los registros CM1CON0 y CM2CON0 respectivamente. Además, el comparador C2 tiene un segundo registro de control, CM2CON1, para controlar la interacción con el Timer1 y la lectura simultánea de ambas salidas del comparador.

El multiplexado para la selección de las entradas a comparar y las salidas a generar depende del registro de control de comparación CM1CON0 (CM2CON0). Existen 8 configuraciones posibles de entradas y salidas de comparación. En ese mismo registro se pueden leer los estados en la salida de cada uno de los comparadores C1 y C2 y si se desea invertir o no la lógica del resultado de la comparación que aparece a la salida.

La referencia de tensión para la comparación puede ser:

- Externa: se conectará a cualquiera de los pines de entrada externos del comparador y esa tensión deberá estar comprendida entre VDD y VSS.
- Interna: se puede emplear una referencia de tensión generada a nivel interno por parte de un módulo interno específico presente dentro del microcontrolador.

El módulo de comparación puede generar una interrupción por activación del flag C1IF (C2IF) presente en el registro PIR2. El flag C1IF (C2IF) se pondrá a 1 si se produce un cambio en la salida de uno de los dos comparadores (C1OUT ó C2OUT) desde la última lectura que se realizó del registro CM1CON0 (CM2CON0).

Si un comparador se encuentra activo y se sitúa al microcontrolador en modo de bajo consumo (SLEEP), el comparador permanecerá activo y las interrupciones si están activadas mediante sus máscaras serán funcionales y podrán sacar al micro de su modo de bajo consumo (lo despiertan) si cambia la salida de alguno de los comparadores desde la última lectura [4].

3.4.9 Módulo Puerto Serie Síncrono Principal (MSSP)

El Puerto Serie Síncrono Principal es una interfaz de comunicaciones de datos digitales, útil para comunicarse con otros dispositivos periféricos o microcontroladores. Donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.

El módulo de MSSP puede funcionar en uno de dos modos siguientes:

Interfaz Periférico Serie (SPI)

Circuito Inter-Integrado (I2C)

1. modo principal completo
2. modo auxiliar

El módulo de MSSP tiene un registro de estado (SSPSTAT) y dos de control (SSPCON1 y SSPCON2). El uso de estos registros y sus bits de configuración son bastante diferentes dependiendo de si el módulo de MSSP está funcionado en modo de SPI o de I2C.

3.4.9.A MODO SPI

El modo SPI permite transmitir y recibir 8 bits de datos síncronamente. Usando los siguientes pines:

- Salida de datos serie (SDO) - RC5/SDO
- Entrada de datos serie (SDI) - RC4/SDI/SDA
- Reloj serie (SCK) - RC3/SCK/SCL/LVDIN
- Modo esclavo (SS) - RA5/SS/AN4

El SPI usa un registro de desplazamiento (SSPSR) y un buffer (SSPBUF) para desplazar y guardar los datos. El SSPSR desplaza los datos (MSb primero), para que luego se almacenen en el SSPBUF y una vez que los 8 bits de datos se hayan recibido, el bit de detección de buffer completo, BF (SSPSTAT<0 >), y el bit indicador de interrupción, SSPIF, se activen. Este almacenamiento doble de los datos recibidos permite que comience la recepción del byte siguiente antes de leer los

datos que acaban de ser recibidos. Cualquier escritura en el registro SSPBUF durante la transmisión/recepción de datos será ignorada, y el bit que detecta la colisión en la escritura, WCOL (SSPCON1<7 >), será activado.

3.4.9.B MODO I²C

El módulo MSSP en modo I²C implementa todas las funciones del maestro y del esclavo, y proporciona interrupciones en los bits de start y de stop en hardware para determinar si el bus está libre. El módulo MSSP implementa las especificaciones estándares, así como direccionamientos 7 bits y 10 bits. Dos pines se utilizan para la transferencia de datos:

- Reloj serie (SCL) - RC3/SCK/SCL
- Datos serie (SDA) - RC4/SDI/SDA

SSPCON1, SSPCON2 y SSPSTAT son los registros de control y de estado en modo de I2C. SSPSR es el registro de desplazamiento, usado para desplazar datos hacia adentro o hacia fuera. SSPBUF es el Buffer, en donde se guardan. El registro de SSPADD almacena la dirección del dispositivo auxiliar o esclavo cuando el SSP se configura en modo auxiliar I2C. Cuando el SSP se configura en el modo principal o maestro, los siete bits más bajos de SSPADD actúan como valor de la frecuencia de transmisión, en baudios. En las operaciones de recepción, SSPSR y SSPBUF juntos, crean un receptor de almacenamiento doble. Cuando SSPSR recibe un byte completo, se transfiere a SSPBUF y se activa la interrupción de SSPIF.

3.4.10 Transmisor/Receptor Asíncrono/Síncrono Universal (USART)

El USART (universal synchronous asynchronous receiver transmitter) es uno de los dos puertos series de los que dispone los PIC16F46K20. Puede funcionar de forma síncrona (half duplex) o asíncrona (full duplex) maestro o esclavo.

El funcionamiento del módulo EUSART es controlado a través de tres registros, TXSTA, RCSTA y BAUDCON.

3.4.10.A Modo Asíncrono del USART

En este modo, el USART utiliza el estándar NRZ, no retorno a cero (un bit de inicio, ocho o nueve bits de datos y un bit de stop). El formato de datos más común es 8-bits. Un generador de frecuencia de 8 bits se puede utilizar para derivar frecuencias estándar de la frecuencia del oscilador. El transmisor y el receptor del USART son funcionalmente independientes, pero utilizan la misma frecuencia y formato de datos. La paridad no es soportada por el hardware, pero se puede implementar por software (almacenado como el noveno bit de datos). El modo asíncrono se selecciona desactivando el bit SYNC (TXSTA<4 >) y se para durante el estado de reposo (SLEEP) [4].

El modulo asíncrono del USART consiste en los siguientes elementos importantes:

- Generador de frecuencia en baudios
- Circuito de muestreado
- Transmisor asíncrono
- Receptor asíncrono

3.4.10.B Modo Maestro Síncrono del USART

En modo maestro síncrono, los datos se transmiten de una manera unidireccional (es decir, la transmisión y la recepción no ocurren en el mismo tiempo). Al transmitir datos, se inhibe la recepción y viceversa. El modo síncrono se habilita activando el bit SYNC (TXSTA<4 >). Además, se activa el bit SPEN (RCSTA<7 >) para configurar los pines I/O, RC6/TX/CK y RC7/RX/DT a las líneas CK (reloj) y DT (datos), respectivamente. El modo principal se habilita activando el bit CSRC (TXSTA<7 >).

3.4.10.C Modo Esclavo Síncrono del USART

El modo auxiliar síncrono se diferencia del modo maestro en el hecho de que el reloj de desplazamiento viene provisto externamente por el pin RC6/TX/CK (en vez de ser provisto internamente como en el modo maestro). Esto permite que el dispositivo transfiera o reciba datos durante el modo reposo. El modo esclavo se habilita activando el bit CSRC (TXSTA<7>).

Para poner en marcha una transmisión esclava síncrona es necesario:

- Habilitar el puerto serie esclavo síncrono activando los bits SYNC y SPEN, y desactivando el bit CSRC.
- Desactivar los bits CREN y SREN.
- Habilitar el bit TXIE, si se usan interrupciones.

3.5 HERRAMIENTAS DE DESARROLLO

Para el desarrollo de este proyecto se ha usado la herramienta MPLAB X IDE, proporcionada por Microchip y que se puede descargar desde su página. Es un programa que corre bajo Windows y como tal, presenta las clásicas barras de programa, de menú, de herramientas de estado, etc. El ambiente MPLAB posee editor de texto, compilador y simulador. Para comenzar un programa desde cero para luego grabarlo al μ C los pasos a seguir son [7]:

1. Crear un Proyecto nuevo
2. Elegir el microcontrolador a utilizar desde SELECT DEVICE
3. Seleccionar el programador
4. Seleccionar el compilador
5. Elegir el nombre y la ubicación del proyecto
6. Crear un nuevo archivo con extensión dependiente del compilador y nombre cualquiera

Una vez realizado esto, se está en condiciones de empezar a escribir el programa para luego compilarlo y grabarlo en el PIC.

El compilador usado es el MPLAB C18 que es un lenguaje de programación que sigue la norma ANSI C, salvo en particularidades de los microcontroladores. Proporciona Librerías para comunicaciones SPI, I2C, UART, USART, generación PWM, cadena de caracteres y funciones matemáticas de coma flotante. Maneja números reales de 32 bits (float y double).

Algunas reglas que hay que seguir a la hora de programar en C18 son:

1. Incluir las librerías que hay que utilizar en el proyecto incluida la del PIC a usar, la cual tiene las direcciones de los registros del mismo.
2. Configuración de los fuses del microcontrolador.
3. Definir las variables globales.
4. Y finalmente, escribir el código del programa

Además de los calificadores ANSI (const, volatile) el MPLAB C18 introduce los siguientes nuevos tipos: far, near, rom y ram. Las variables tipo RAM residen en la memoria de datos y dependiendo del calificador far/near, se supondrá que están en cualquier banco o restringidas a la posiciones más bajas de la RAM, que permite el acceso en modo banco. Las variables tipo ROM residen en memoria de programa. Si se utiliza el calificativo far la variable puede estar en cualquier posición de ROM y si es un puntero puede contener direcciones por encima de los 64KB. Al contrario si es near la variable se sitúa en posiciones menores a 64KB o si es un puntero puede acceder sólo hasta posiciones por debajo de 64KB [8].

Finalmente para grabar el código se ha usado el depurador/grabador MPLAB ICD3, es un depurador en circuito que facilita la depuración y puesta a punto del programa de aplicación, haciendo que el dispositivo PIC pueda ejecutar dicho programa paso a paso, con puntos de ruptura, tiempo real, etc. Por otro lado el ICD3 incluye el hardware y firmware (puede ser actualizado fácilmente desde el sitio web de Microchip) necesario para realizar las tareas propias de grabación.

El MPLAB ICD 3 es muy similar al MPLAB ICD 2, pero la supera en algunas prestaciones y velocidad. Cuenta con interfaz USB 2.0 que le permite la conexión con el PC. Algunas características relevantes de este depurador/programador son [9]:

- Depuración en tiempo real
- Interfaz sólida
- Interconectividad estándar
- Alta velocidad
- Emulación a bajo voltaje
- Fácil mantenimiento
- Bajo coste

4. Desarrollo

En este capítulo se va a explicar el funcionamiento del sistema, describiendo tanto la parte Hardware, así como la parte software del mismo.

4.1 HARDWARE DEL SISTEMA

Como núcleo del sistema se ha usado el microcontrolador PIC18F46K20 por todas las consideraciones mencionadas en capítulos anteriores, adquiriendo y procesando los datos provenientes de los dos sensores de temperatura: EL analógico basado en un termopar tipo K y el acondicionador de señal (AD595) y el digital basado en un sensor de temperatura digital (ADT7301). La conexión de los diversos dispositivos es la mostrada a continuación.

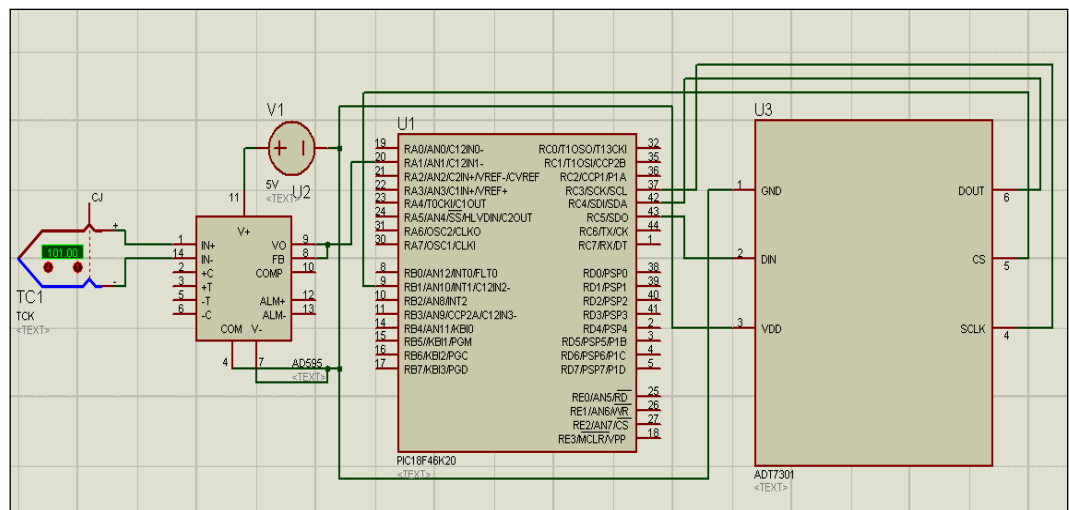


Figura 4.1: Esquema de conexión del sistema

El rango de temperatura a medir y el tamaño de la placa han sido dos factores muy decisivos a la hora de elegir los sensores de temperatura. Puesto que uno de los requisitos del proyecto es medir temperaturas de -20 hasta 200°C, y esto no se puede conseguir con todos los sensores disponibles, sobre todo con los integrados. Y por otro lado, al ser un sistema embebido, no se ha usado ni los RTD ni los termistores por el autocalentamiento y por su necesidad de una fuente de corriente. Lo cual, ha limitado las posibilidades de elección a los termopares, por ser los únicos autogeneradores (ya que no es necesario alimentarlos).

El termopar usado es de tipo K (Cromo (Ni-Cr) / Aluminio (aleación de Ni - Al)), y ha sido elegido por su disponibilidad (está disponible en una variedad de sondas), su bajo coste, su rango de temperatura (de -200°C a $+1.372^{\circ}\text{C}$) y la buena resistencia a la oxidación.

La salida de los termopares es muy baja por eso se ha recurrido al AD595 de Analog Devices, que es un acondicionador de señal que además de amplificar la señal proporcionada por el termopar, compensa la tensión generada por la unión fría de éste (Figura 4.2) [1 y 10].

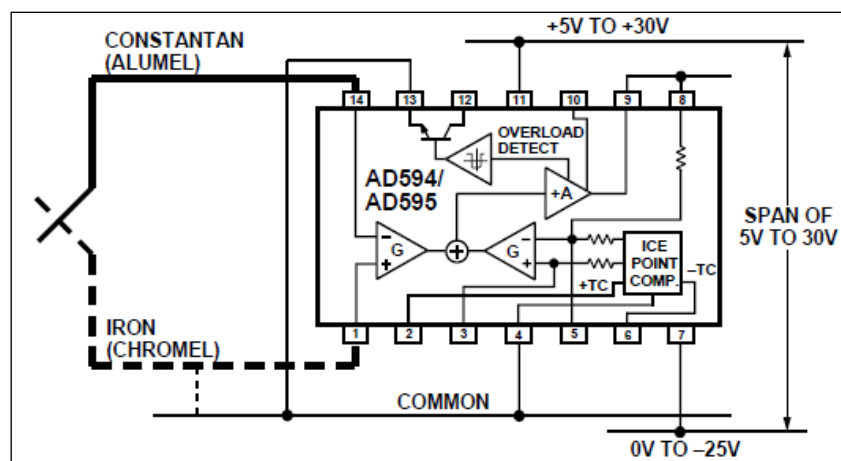


Figura 4.2: Diagrama de bloques del AD594/AD595

Este dispositivo tiene una sensibilidad de $10\text{mV}/^{\circ}\text{C}$ y se obtiene en los terminales 8 y 9 (conectados entre sí). El termopar se conecta a los terminales 1 y 14 (respetando la polaridad del termopar). Se puede alimentar desde 5 hasta 30V, si se utiliza una fuente de alimentación de +5v con la configuración de la figura 4.2, se pueden medir temperatura hasta los 300°C . Y si se quiere un rango más amplio, se puede aumentar la alimentación. Por ejemplo, si la fuente de alimentación es de 15V el rango de medida aumentará hasta los 1250°C con un termopar tipo K. Aunque se recomienda el uso de la menor tensión de alimentación que satisface las necesidades, para minimizar los errores debidos al autocalentamiento. Incluye también una alarma que se activa cuando se detecta una rotura o una desconexión del termopar.

La sensibilidad de este circuito es de $40,44\text{ }\mu\text{V}/^{\circ}\text{C}$, por lo que la ganancia es de 247,3 (resulta de dividir $10\text{ mV}/^{\circ}\text{C}$ por $40,44\text{ }\mu\text{V}/^{\circ}\text{C}$). El circuito elimina

internamente el offset, sumando a la tensión del termopar un voltaje de de $11\ \mu\text{V}$. Así, la función de transferencia de este dispositivo queda de la forma [10]:

$$\text{Salida del AD595} = (\text{Tensión termopar tipo K} + 11\mu\text{V}) * 247,3 \quad (4.1)$$

En este diseño se ha alimentado el dispositivo con 5V, y su salida se ha conectado a la entrada 1 del convertidor analógico digital del microcontrolador que corresponde al pin (RA1/AN1/C12IN1-), para que su valor analógico sea convertido a un valor digital de 10 bits, siguiendo el modelo de aproximaciones sucesivas.

Y como se muestra en el esquema de la figura 4.3, el diseño consta también de un sensor de temperatura integrado, el ADT7301, para medir la temperatura ambiente.

El dispositivo opera en el rango de -40 hasta 150°C , y se puede alimentar con 2.7 hasta 5.25V. Tiene una salida digital de 13 bits, más un bit de signo, con una resolución de 0.03125°C y está disponible en dos encapsulados el SOT-23 y MSOP. La figura 4.3 muestra el diagrama de bloques de este dispositivo [11]:

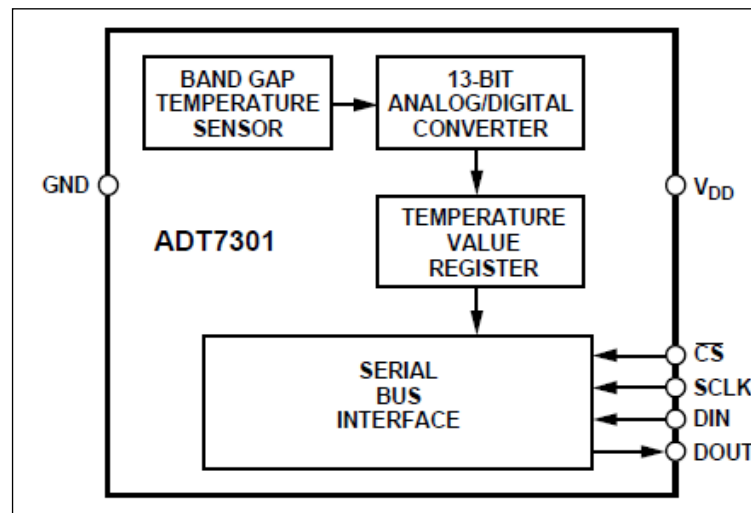


Figura 4.3: Diagrama de bloques del ADT7301

Su modo de funcionamiento es el siguiente: un sensor de temperatura recoge la temperatura ambiente, generando un voltaje a su salida, éste pasa por el convertidor analógico digital de 13 bits para convertirlo en valor numérico que se

almacena en un registro llamado valor de temperatura, para que luego se transmite a través bit a bit a través del bus serial.

La descripción de los pines se recoge en la tabla siguiente:

Tabla 4.1: Descripción de los pines del ADT7301

Pin	descripción
GND	Tierra
DIN	Entrada serial de datos
VDD	Tensión de alimentación, 2.7 a 5.25V
SCLK	Entrada de reloj del puerto serie
/CS	Entrada de selección del chip
DOUT	Salida serial de datos
NC	No conectar

Como se puede ver en la figura 4.3, la interfaz serial del ADT7301 consta de cuatro pines: CS, SCLK, DIN, y DOUT, y puede operar en el modo de 3 hilos con DIN conectado a tierra, en cuyo caso la interfaz tiene la capacidad de sólo lectura, con los datos se leen desde el registro de datos a través de la línea de DOUT. El pin CS se usa para crear una ventana de comunicación, activando o desactivando el chip, lo que mejora la sincronización entre el ADT7301 y el dispositivo maestro.

En el caso de una operación de lectura, se accede a los datos del registro (valor de la temperatura) mientras que en una operación de escritura, los datos se escriben en el registro de control.

En este diseño el AD7301, se ha configurado para trabajar en el modo de 3 hilos, anulando la entrada DIN. Y para el su correcto funcionamiento, se ha tenido en cuenta a la hora de programar el microcontrolador, los tiempos que indica el fabricante en sus hojas de características (Figura 4.4):

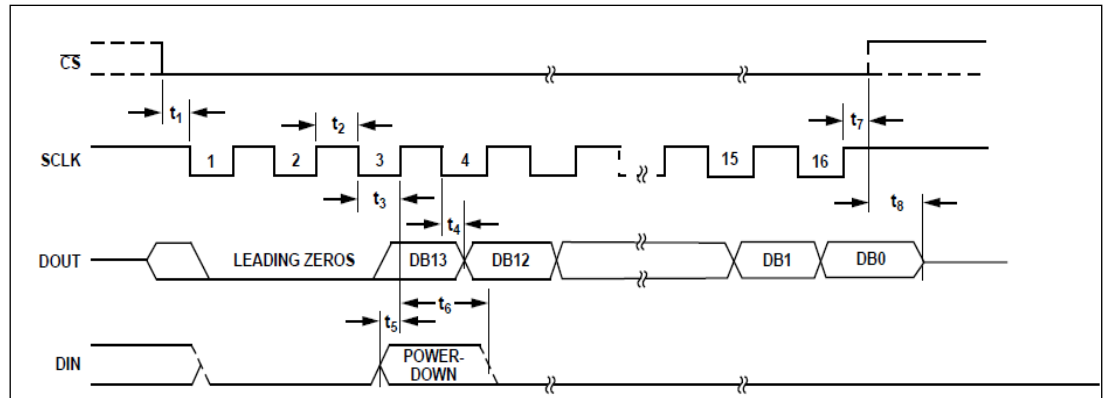


Figura 4.4: Diagrama de tiempo de la interfaz serial

La figura de abajo muestra una descripción de esos tiempos:

Parameter	Limit	Unit	Comments
t_1	5	ns min	$\overline{CS}$ to SCLK setup time
t_2	25	ns min	SCLK high pulse width
t_3	25	ns min	SCLK low pulse width
t_4	35	ns max	Data access time after SCLK falling edge
t_5	20	ns min	Data setup time prior to SCLK rising edge
t_6	5	ns min	Data hold time after SCLK rising edge
t_7	5	ns min	$\overline{CS}$ to SCLK hold time
t_8	40	ns max	$\overline{CS}$ to DOUT high Impedance

Figura 4.5: Descripción de los tiempos de la figura en la anterior

Pasando 16 ciclos de reloj, los datos están disponibles en el registro valor de la temperatura, que es un registro de 14 bits de sólo lectura, y es el encargado de almacenar la lectura de la temperatura generada por el ADC en formato de complemento a dos (13 bits más un bit de signo). Donde el bit más significativo (DB13) es el bit de signo [11].

El Puerto Serie Síncrono Principal (MSSP) del μC , se ha configurado como SPI para realizar la comunicación serie con el dispositivo. La conexión de los pines es la de la figura que sigue (se usa el pin RB1 como el pin selección del chip):

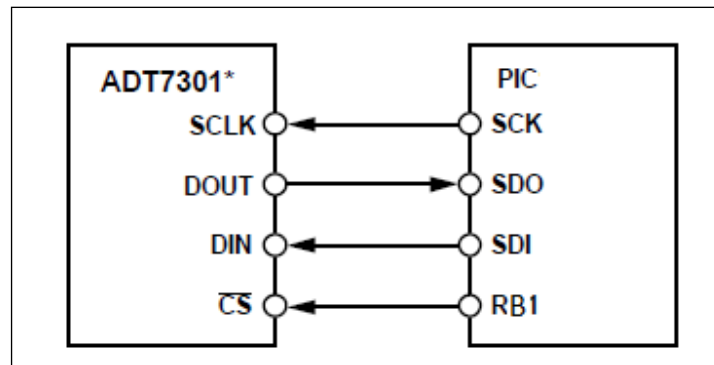


Figura 4.6: Conexión del AD7301 con el PIC18F46K20
(no se han dibujado todos los pines por simplificación)

Para alimentar el microcontrolador se ha usado una pila de 3V, ya que éste se puede trabajar con un voltaje que puede variar desde 1.8V hasta 3.6V. La pila es de litio del fabricante Panasonic y se caracteriza por la alta densidad energética, baja descarga de corriente, amplio rango de temperatura, alta fiabilidad y la larga vida útil. También dispone de un terminal para montaje en PCB.



Figura 4.7: Pila de litio

Mientras que tanto el AD595 así como el ADT7301 necesitan una fuente de alimentación de 5v. Por eso se ha recorrido al convertidor DC/DC (MPDTY102S) para convertir los 3V generados por la pila a los 5V requeridos por los dos dispositivos. El diagrama de bloques del convertidor es el que se muestra a continuación.

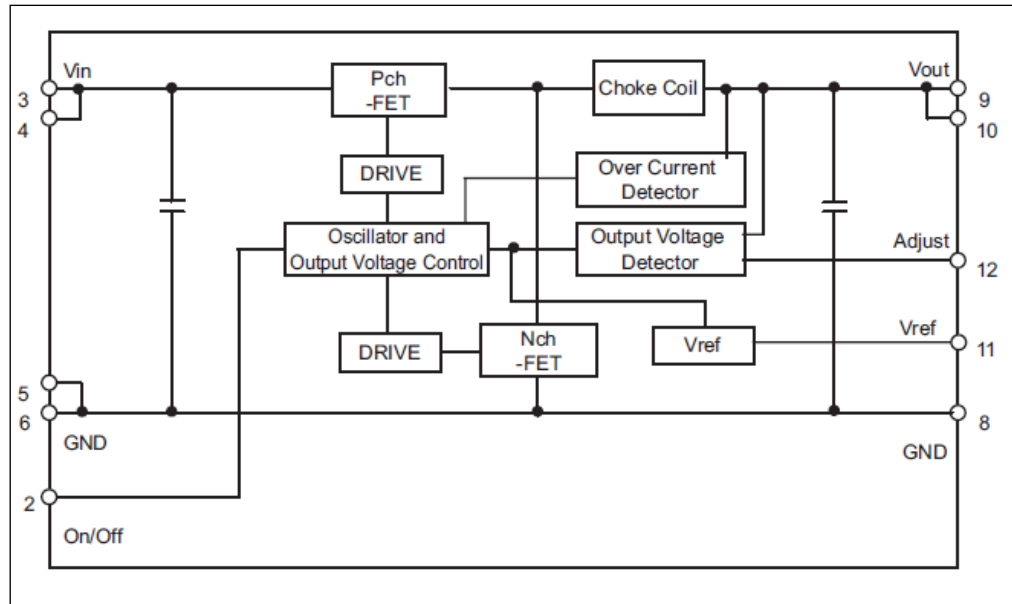


Figura 4.8: Diagrama de bloques del convertidor MPDLY102S

El voltaje de salida de este dispositivo es ajustable, de forma que puede variar desde 0.8 hasta 5.0V. Para eso hay que colocar una resistencia entre los pines 11 y 12 o 12 y 8, dependiendo si se quiere aumentar o disminuir el voltaje de salida (figura 4.9). Para calcular el valor de dicha resistencia se usa una de las formulas siguientes [13]:

$$R1 = \frac{159,25 - 31.7 * V_{out}}{V_{out} - 2.5} \quad (4.2)$$

$$R2 = \frac{19.7 * V_{out} - 11.8}{2.5 - V_{out}} \quad (4.3)$$

Y como en este caso interesa aumentar el voltaje de salida, la expresión usada es la primera, dando un valor de resistencia de 330Ω.

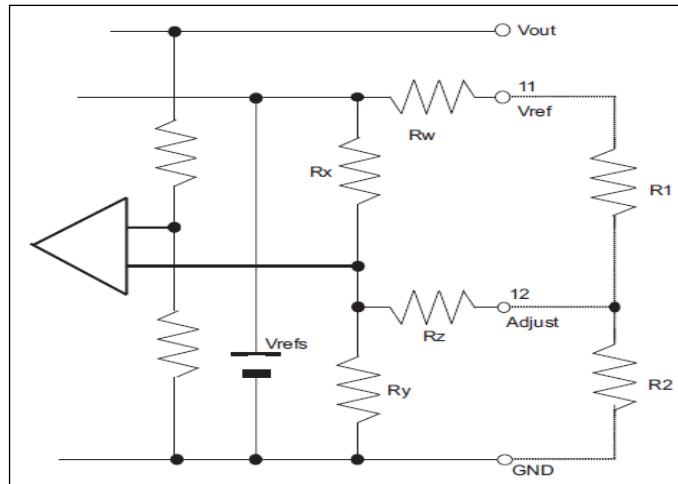


Figura 4.9: Ajuste del voltaje de salida

4.2 SOFTWARE DEL SISTEMA

Inicialmente se tiene que considerar la manera de funcionamiento de los dos sensores, que como anteriormente se mencionó, uno tiene salida analógica de forma que entrega 10mV por cada grado centígrado, y el otro con salida digital de 13 bit (14 bits con el signo) que se transmiten de manera serial.

Como la salida del sensor compuesto por el termopar y el acondicionador de señal el AD595 es analógica, ésta se convertirá en una cantidad digital para facilitar su manejo. Para esto se empleará el módulo convertidor analógico digital del microcontrolador descrito anteriormente.

Los pasos necesarios para llevar a cabo la conversión son [4]:

1. Configurar el pin RA1/AN1/C12IN1 como una entrada analógica en ANSEL.
2. Establecer las referencias de tensión en ADCON1.
3. Seleccionar la fuente de reloj, y el tiempo de adquisición en ADCON2.
4. Seleccionar el canal y activar el ADC en ADCON0.
5. Iniciar la conversión en ADCON0.

Para usar un pin como una entrada analógica, no debe ser usado por otras funciones periféricas multiplexadas en el mismo, y poner un 1 en el bit

correspondiente a este pin tanto en el registro TRIS (para indicar que es una entrada) así como en el registro ANSEL o ANSELH (para indicar que es una entrada analógica).

Los bits VCFGx del registro ADCON1 (Figura 4.10) sirven para seleccionar el voltaje de referencia del ADC, que puede ser VDD y VSS, alguna referencia externa proveniente de los pines AN2 y AN3 o alguna combinación de las dos. Y como en este diseño se ha usado las tensiones de referencia VDD y VSS el registro queda de la siguiente forma:

ADCON1 = 0b00000000

U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	—	VCFG1	VCFG0	—	—	—	—
bit 7						bit 0	

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6	Unimplemented: Read as '0'
bit 5	VCFG1: Negative Voltage Reference select bit 1 = Negative voltage reference supplied externally through VREF- pin. 0 = Negative voltage reference supplied internally by Vss.
bit 4	VCFG0: Positive Voltage Reference select bit 1 = Positive voltage reference supplied externally through VREF+ pin. 0 = Positive voltage reference supplied internally by VDD.
bit 3-0	Unimplemented: Read as '0'

Figura 4.10: Registro de control del ADC ADCON1

El convertidor no capta la señal directamente del pin AN1, sino que primero espera a que dicho nivel de tensión se carga en la capacidad de muestreo (CHOLD), para luego iniciar la conversión desde allí. El tiempo que dura el condensador conectado a un canal del ADC desde el instante en el que se pone a 1 el bit GO/DONE hasta que comience la conversión se le denomina tiempo de adquisición. Éste varia, principalmente de acuerdo con la impedancia del circuito conectado al canal del ADC y de la impedancia del interruptor de muestro. La figura 4.11 muestra el modelo de la entrada analógica y la variación de la impedancia del interruptor de muestro con la tensión VDD [4 y 12].

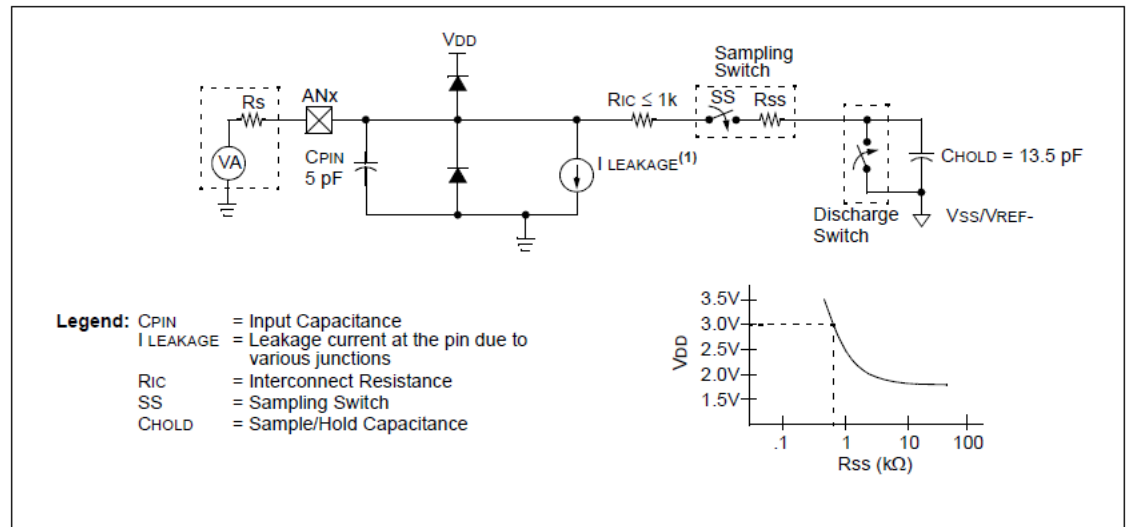


Figura 4.11: Modelo de la entrada analógica

En la documentación proporcionada por Microchip, vienen las formulas para calcular el tiempo de adquisición. Para está aplicación, se ha establecido el valor más alto de este parámetro (20TAD), poniendo el valor 111 en los bits ACQT<2:0> del registro ADCON2.

El reloj de ADC se debe establecer lo más corto posible, pero con la condición de ser mayor que el tiempo que se tarda en ejecutar la conversión de cada bit (TAD). En las hojas de características del PIC1846K20, el fabricante establece el tiempo de 1.4 μ s como tiempo mínimo de TAD para este dispositivo. Y como el oscilador está configurado para trabajar con una frecuencia de $F_{osc} = 1\text{MHz}$, se ha seleccionado en los bits ADCS2:ADCS0 del registro ADCON2, una frecuencia igual que la mitad de F_{osc} , dando un reloj de 500KHz, o lo que es un periodo de 2 μ s, que es mayor que el mínimo pedido. LA figura que sigue muestra la descripción de los bits del registro ADCON2.

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **ADFM:** A/D Conversion Result Format Select bit
1 = Right justified
0 = Left justified

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT<2:0>:** A/D Acquisition time select bits. Acquisition time is the duration that the A/D charge holding capacitor remains connected to A/D channel from the instant the GO/DONE bit is set until conversions begins.
000 = 0⁽¹⁾
001 = 2 T_{AD}
010 = 4 T_{AD}
011 = 6 T_{AD}
100 = 8 T_{AD}
101 = 12 T_{AD}
110 = 16 T_{AD}
111 = 20 T_{AD}

bit 2-0 **ADCS<2:0>:** A/D Conversion Clock Select bits
000 = Fosc/2
001 = Fosc/8
010 = Fosc/32
011 = Frc⁽¹⁾ (clock derived from a dedicated internal oscillator = 600 kHz nominal)
100 = Fosc/4
101 = Fosc/16
110 = Fosc/64
111 = Frc⁽¹⁾ (clock derived from a dedicated internal oscillator = 600 kHz nominal)

Note 1: When the A/D clock source is selected as FRC then the start of conversion is delayed by one instruction cycle after the GO/DONE bit is set to allow the SLEEP instruction to be executed.

Figura 4.12: Registro de control del ADC ADCON2

Cabe destacar que el bit ADFM del registro ADCON2 se ha puesto a 0, lo que significa que el resultado se justifica a la derecha (los 8 bits más significativos en el registro ADRESH y los 2 restantes en los bits más significativos de ADRESL (Figura 4.13).

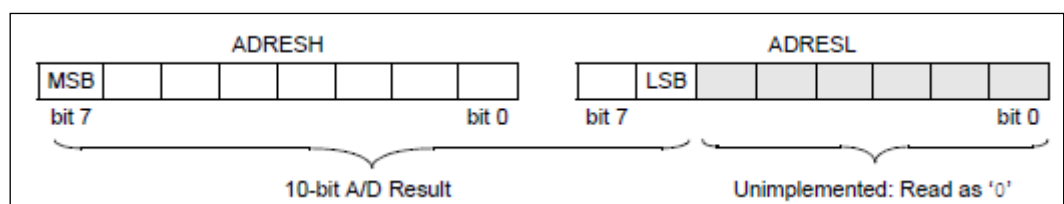


Figura 4.13: Formato del resultado

Con todo lo anterior, el registro ADCON2 queda cargado con el valor: 0b00111000.

La salida del sensor está conectada a la entrada 5 del ADC, por lo que hay que seleccionar el canal 5 en el registro ADCON0. Y finalmente, y para acabar de configurar este registro, se pone a 1 el bit ADON para habilitar el ADC y se deja a 0 el bit GO/DONE hasta la hora de iniciar la conversión. Por lo que el registro ADCON0 (Figura 4.14) se ha establecido con el valor 0b00000101.

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7		bit 0					
Legend:							
R = Readable bit		W = Writable bit		U = Unimplemented bit, read as '0'			
-n = Value at POR		'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	
bit 7-6	Unimplemented: Read as '0'						
bit 5-2	CHS<3:0>: Analog Channel Select bits						
	0000 = AN0						
	0001 = AN1						
	0010 = AN2						
	0011 = AN3						
	0100 = AN4						
	0101 = AN5 ⁽¹⁾						
	0110 = AN6 ⁽¹⁾						
	0111 = AN7 ⁽¹⁾						
	1000 = AN8						
	1001 = AN9						
	1010 = AN10						
	1011 = AN11						
	1100 = AN12						
	1101 = Reserved						
	1110 = Reserved						
	1111 = FVR (1.2 Volt Fixed Voltage Reference) ⁽²⁾						
bit 1	GO/DONE: A/D Conversion Status bit						
	1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.						
	This bit is automatically cleared by hardware when the A/D conversion has completed.						
	0 = A/D conversion completed/not in progress						
bit 0	ADON: ADC Enable bit						
	1 = ADC is enabled						
	0 = ADC is disabled and consumes no operating current						
Note 1: These channels are not implemented on PIC18F2XK20 devices.							
Note 2: Allow greater than 15 μ s acquisition time when measuring the Fixed Voltage Reference.							

Figura 4.14: Registro de control del ADC ADCON0

Para iniciar la conversión ADC, se pone a 1 el bit GO/DONE del ADCON0. Cuando la conversión se realiza el hardware pone a cero este bit. Por lo que el GO / DONE puede ser encuestados (while (ADCON0bits.GO_DONE == 1)), esperando a

que la conversión se complete. Una vez finalizada la conversión y GO/DONE = 0, el resultado de la conversión ADC puede ser leído desde ADRESH y ADRESL.

Todo lo mencionado anteriormente sobre la configuración del convertidor analógico digital anterior, se ha llevado a cabo mediante dos funciones:

- **ADC_Init():** Como su propio nombre dice, inicializa el módulo ADC, estableciendo el pin AN1 como entrada analógica, para capturar la salida del AD595, luego configura los registros ADCON0, ADCON1 y ADCON2, cargando los valores 0x05, 0x00 y 0x24 respectivamente.
- **ADC_Convert():** Esta función es la encargada de devolver el resultados de la conversión. Pone el bit GO/DONE a 1 para que comience la conversión, espera hasta que este bit se pone a 0, indicando que la finalización de la conversión y luego devuelve el resultado una variable de tipo entero.

Por otra parte, el ADT7301 usa la comunicación serie para comunicarse con otros dispositivos periféricos o microcontroladores. Para eso, y para establecer dicha comunicación, se ha configurado el módulo MSSP para trabajar en el modo SPI. A continuación se detalla el protocolo de funcionamiento seguido para conseguir la comunicación en este modo.

El μ C desempeña el papel del maestro, y inicializa el ciclo de comunicación cuando selecciona el ADT7301 poniendo en bajo el pin RB1. El maestro genera los pulsos de reloj requeridos para intercambio de información en el pin RC3/SCK/SCL/LVDIN. Cuando pasan 16 ciclos de reloj el sensor ya dispone de los datos que serán enviados bit a bit al μ C. Éste los va almacenando en el registro de desplazamiento SSPSR, para que luego los envía al registro SSPBUF. Una vez finalizada la recepción de un byte, se activa el bit BF (del registro SSPSTAT) y el SSPIF (del registro PIR1). Después de cada recepción de un paquete el μ C sincroniza el chip colocando en alto la línea del pin RB1 (chip select /CS.)

Lo anterior conlleva la configuración de todos los registros implicados en este modo, a continuación se describe cada uno de ellos:

1. El registro de control SSPCON1, a través de este registro se puede:

- Detectar si hay una colisión de información mediante el bit WCOL.
- Detectar si hay Overflow: se activa el bit SSPOV=1 cuando se completa la recepción de un nuevo dato antes de leer el dato anterior, perdiendo el dato previamente recibido. Este error sólo es posible si el dispositivo es ESCLAVO.
- Habilitar el módulo MSSP poniendo el bit SSPEN a 1. Si se desea “resetear” el módulo MSSP, se debe poner a 0 este bit y luego volver a ponerlo a 1.
- Seleccionar la polaridad de reloj (bit CKP).
- Programar la velocidad de transferencia en los bits SSPM<3:0>, y se puede elegir entre los valores:
 - $F_{osc}/4$
 - $F_{osc}/16$
 - $F_{osc}/64$
 - Frecuencia de Salida TMR2/2

La figura siguiente muestra, con más detalle, los bits de este registro.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

Legend:							
R = Readable bit		W = Writable bit		U = Unimplemented bit, read as '0'			
-n = Value at POR		'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	

bit 7	WCOL: Write Collision Detect bit (Transmit mode only) 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared by software) 0 = No collision
bit 6	SSPOV: Receive Overflow Indicator bit ⁽¹⁾ <u>SPI Slave mode:</u> 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared by software). 0 = No overflow
bit 5	SSPEN: Synchronous Serial Port Enable bit ⁽²⁾ 1 = Enables serial port and configures SCK, SDO, SDI and $\overline{SS}$ as serial port pins. When enabled, the SDA and SCL pins must be configured as inputs. 0 = Disables serial port and configures these pins as I/O port pins
bit 4	CKP: Clock Polarity Select bit 1 = Idle state for clock is a high level 0 = Idle state for clock is a low level
bit 3-0	SSPM<3:0>: Synchronous Serial Port Mode Select bits ⁽³⁾ 0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin 0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled 0011 = SPI Master mode, clock = TMR2 output/2 0010 = SPI Master mode, clock = Fosc/64 0001 = SPI Master mode, clock = Fosc/16 0000 = SPI Master mode, clock = Fosc/4

Figura 4.15: Registro de control 1 del MSSP (modo SPI)

2. El registro de estado SSPSTAT (Figura 4.16): en este modo de funcionamiento, solo tres bits son funcionales, el resto se usan para el modo I²C:

- El bit SMP indica el momento de muestreo. Si éste está a 0 el muestreo se hace a mitad de bit, y si está a 1 se hace al final del tiempo del bit.
- El bit CKE es usado para configurar los flancos activos.
- Y finalmente el bit BF: Se pone a 1 cuando se completa la recepción de un dato se pone a 0 por hardware cuando se lee el registro SSPBUF. Normalmente se emplea únicamente en modo de recepción [4 y 6].

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0
bit 7	SMP: Sample bit <u>SPI Master mode:</u> 1 = Input data sampled at end of data output time 0 = Input data sampled at middle of data output time <u>SPI Slave mode:</u> SMP must be cleared when SPI is used in Slave mode.						
bit 6	CKE: SPI Clock Select bit⁽¹⁾ 1 = Output data changes on clock transition from active to idle 0 = Output data changes on clock transition from idle to active						
bit 0	BF: Buffer Full Status bit (Receive mode only) 1 = Receive complete, SSPBUF is full 0 = Receive not complete, SSPBUF is empty						

Figura 4.16: Registro de estado del MSSP (modo SPI)

Para el manejo y el control de este módulo, se han desarrollado dos funciones, que se describen a continuación.

- **Inicializa_SPI():** configura el MSSP como SPI (Interfaz Periférico Serie): primero carga los valor 0x00 y 0x30 en los registros SSPTAT y SSPCON respectivamente, y luego deshabilita el ADT7301 con el RB1, poniéndolo a 1.
- **SPI_Read():** esta es la función que se ha utilizado para devolver el byte recibido desde el ADT7301. Su modo de funcionamiento es lo siguiente, pone el bit de bandera BF del registro SSPSTAT a cero, carga el valor 0x00 en el buffer SSPBUF y luego se mite en un bucle y no sale de él hasta que se le indica, a través del bit BF (cuando éste se pone a a1), que la operación de recepción haya finalizado. En este momento devuelve el byte recibido en SSPBUF.

En este proyecto, se ha tenido muy en cuenta el tema de ahorro de energía. Ya que se ha programado el microcontrolador para que entra en estado de reposo, cuando se queda sin realizar ninguna operación.

Los microcontroladores disponen de una instrucción especial (SLEEP), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se “congelan” sus circuitos asociados. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

El PIC se "despierta" (sale del modo SLEEP) al producirse algún suceso determinado [4 y 6]:

- reset externo
- reset del WDT
- ciertos tipos de interrupción

Como no se puede ni acceder al dispositivo ni permitir que se resetea cada cierto con el WDT (Watchdog), se ha escogido la última opción para despertar el microcontrolador. Par eso, se ha empleado la interrupción que genera el temporizador TIMER1, cuando éste está trabajando en modo contador y con el reloj incorporado en este módulo.

La figura siguiente muestra el diagrama de bloques de este temporizador:

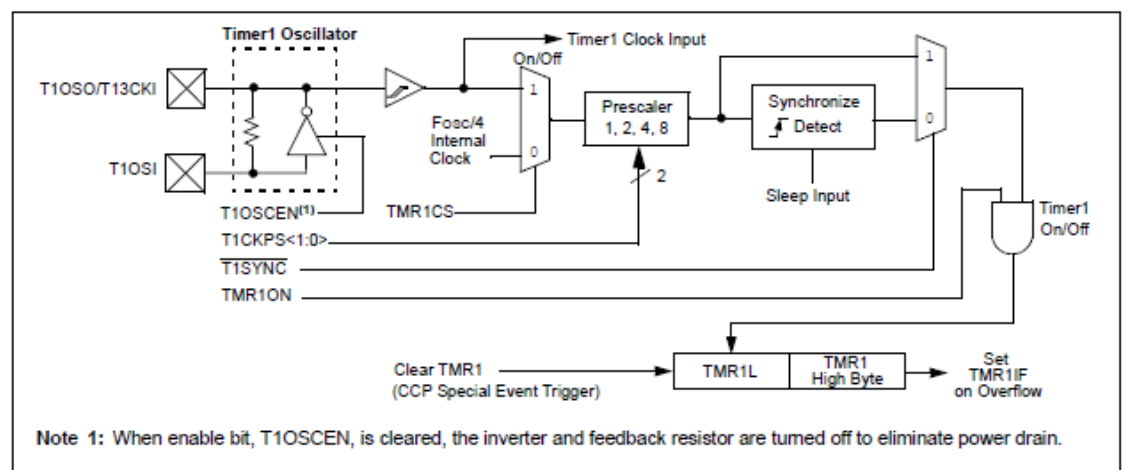


Figura 4.17: Diagrama de bloques del TIMER1

Como se puede observar en este diagrama, este módulo dispone de un oscilador independiente del oscilador del μC , que le permite trabajar en el modo de reposo.

Este módulo lleva asociado un registro para su configuración y control (T1CON), que habrá que programarle de modo adecuado para conseguir la interrupción que saca el PIC del modo SLEEP. La figura siguiente muestra los bits de configuración de dicho registro.

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0
bit 7	RD16: 16-bit Read/Write Mode Enable bit 1 = Enables register read/write of Timer1 in one 16-bit operation 0 = Enables register read/write of Timer1 in two 8-bit operations						
bit 6	T1RUN: Timer1 System Clock Status bit 1 = Main system clock is derived from Timer1 oscillator 0 = Main system clock is derived from another source						
bit 5-4	T1CKPS<1:0>: Timer1 Input Clock Prescale Select bits 11 = 1:8 Prescale value 10 = 1:4 Prescale value 01 = 1:2 Prescale value 00 = 1:1 Prescale value						
bit 3	T1OSCEN: Timer1 Oscillator Enable bit 1 = Timer1 oscillator is enabled 0 = Timer1 oscillator is shut off The oscillator inverter and feedback resistor are turned off to eliminate power drain.						
bit 2	T1SYNC: Timer1 External Clock Input Synchronization Select bit <u>When TMR1CS = 1:</u> 1 = Do not synchronize external clock input 0 = Synchronize external clock input <u>When TMR1CS = 0:</u> This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.						
bit 1	TMR1CS: Timer1 Clock Source Select bit 1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge) 0 = Internal clock (Fosc/4)						
bit 0	TMR1ON: Timer1 On bit 1 = Enables Timer1 0 = Stops Timer1						

Figura 4.18: Registro de control T1CON

El TIMER1 tiene que trabajar con el reloj generado por su oscilador (bit TMR1CS=1), pero con la condición de que éste no se sincroniza con el reloj principal del sistema (T1SYNC=1), porque en caso contrario la sincronización dependerá del reloj del sistema y por consiguiente no se puede trabajar con este módulo en el modo de espera. Luego se habilita el oscilador con el bit T1OSCEN (poniéndolo a 1).

Al igual que el TMR0 el Timer1 puede operar en dos modos: como temporizador y como contador. El modo de funcionamiento está determinado por el tipo de reloj seleccionado, si es interno el TIMER1 trabaja como temporizador y si es externo, en este caso proveniente del oscilador interno del módulo, trabaja como contador. En este último modo y si el TIMER1 está habilitado (TMR1ON=1), el valor de los registros TMR1H y TMR1L se incrementa desde 0000h a FFFFh y una vez que llega a su máximo valor empieza otra vez desde 0 avisándonos de ello por medio de la bandera TMR1F, si está habilitada.

Al producirse esta interrupción, hace que el microcontrolador se despierta y reanuda su trabajo.

El tiempo que tarda el TIMER1 en desbordarse se puede calcular con la expresión [4]:

$$\text{Desbordamiento} = \frac{1}{Fr_{osc_interno}} * Preescalar * 65536 \quad (4.4)$$

La figura siguiente muestra el código C de la función encargada de inicializar el TIMER1:

```
void TIMER1_Init(void)
{
    /*Inicializacion del Timer1*/
    PIE1bits.TMR1IE=1;    /*habilitar la interrupcion*/
    PIR1bits.TMR1IF = 0;  /*poner la bander a 0*/
    T1CON = 0b01111110;   /*Contador 16-bits, preescalar 1:8*/
    /*habilitar el OSC interno, No sincronizar con reloj SYS*/
    TMR1H = 0;            /*Byte más del contador MSB*/
    TMR1L = 0;            /*Byte menos del contador LSB*/
    T1CONbits.TMR1ON = 1; /*Enable Timer1*/
}
```

Figura 4.19: Código C de la función TIMER1_Init

Finalmente, y para terminar la parte de programación, describimos a continuación la función “main”, encargada de la gestión y la organización de todos los módulos, apoyándose en las funciones antes descritas.

El diagrama de flujo de esta función viene reflejado en la siguiente figura.

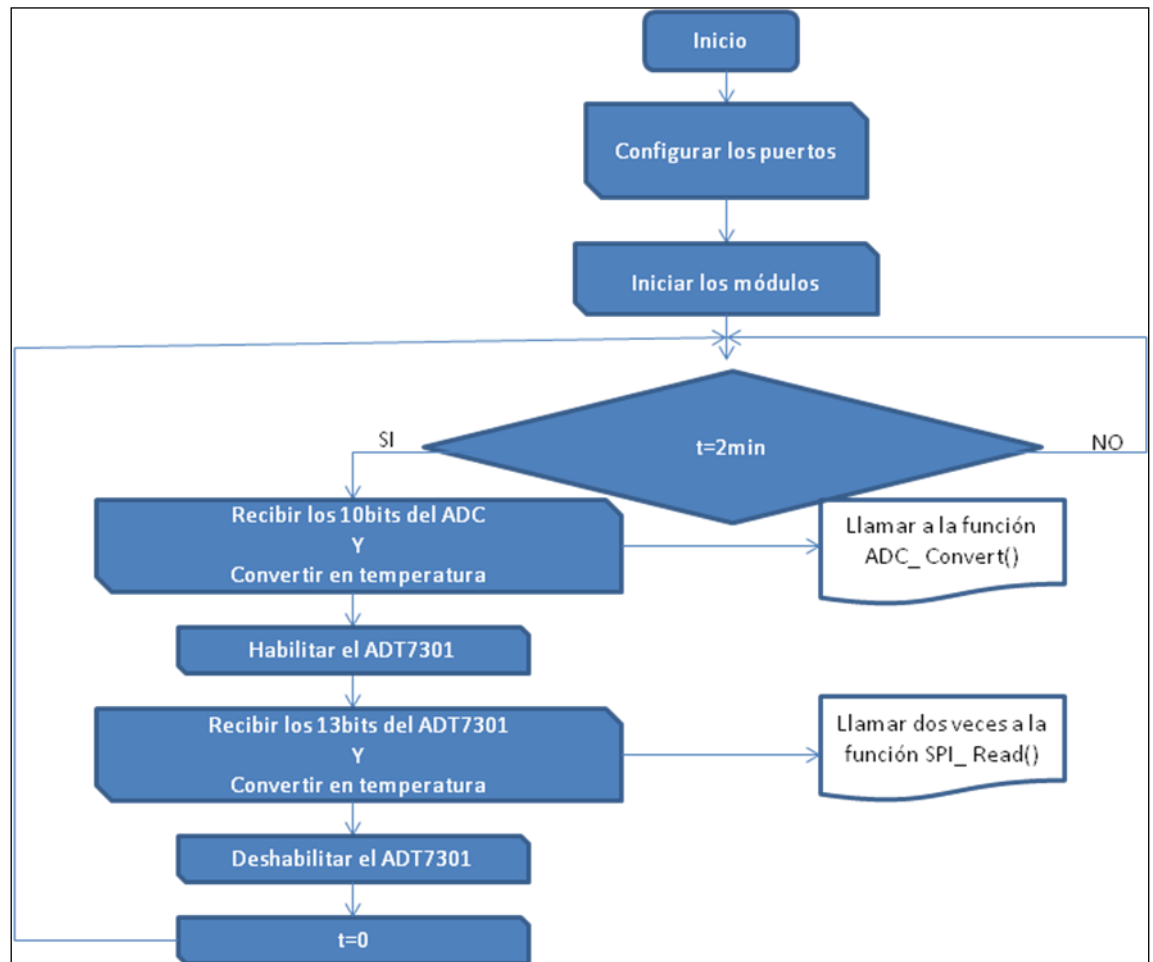


Figura 4.20: diagrama de flujo de la función main

En la función, y como en todas las funciones desarrolladas en C, se declaran las variables al principio. Luego se configuran los pines usados, cargando un 0 o un 1 en el registro TRIS correspondiente, según si el pin es de entrada o de salida.

El paso que viene después es inicializar los módulos ADC, MSSP (para trabajar en modo SPI) y el TIMER1, haciendo llamas a las funciones `ADC_Init()`, `inicializa_SPI()` y `TIMER1_Init()`, respectivamente.

Como ya se ha comentado antes, y para ahorrar energía, el μC se ha programado para hacer una medida cada dos minutos. Durante este tiempo entra en

el modo de espera, esperando a que el TIMER1 lo saque, a través de interrupción que genera éste al desbordarse.

A la hora de reanuda el microcontrolador su trabajo, empieza a recibir los datos proporcionados por los dos sensores, ejecutando las funciones ADC_Convert() y SPI_Read().

La función ADC_Convert() devuelve el valor de la conversión del ADC en entero. Este entero presenta el valor devuelto por el acondicionador de señal AD595. Para recuperar el valor de temperatura a partir de este entero se usa las fórmulas:

$$V_{in} = \frac{\text{valor ADC}}{1023} * V_{ref} (V) \quad (4.5)$$

Y

$$T = 100 * V_{in} (^{\circ}C) \quad (4.6)$$

Siendo V_{in} es la tensión de entrada del ADC y T es la temperatura medida con el termopar.

Por otra parte la función SPI_Read() devuelve solo un byte, por eso hay que hacer dos llamadas a esta función para reconstruir el dato enviado por el ADT7301.

En este caso también el valor enviado por el ADT7301 no corresponde al valor de la temperatura que ha medido, sino que es el valor resultante de hacer la conversión del voltaje generado por su sensor. El valor de temperatura que corresponde al dato enviado se puede calcular con una de las ecuaciones siguientes [11]:

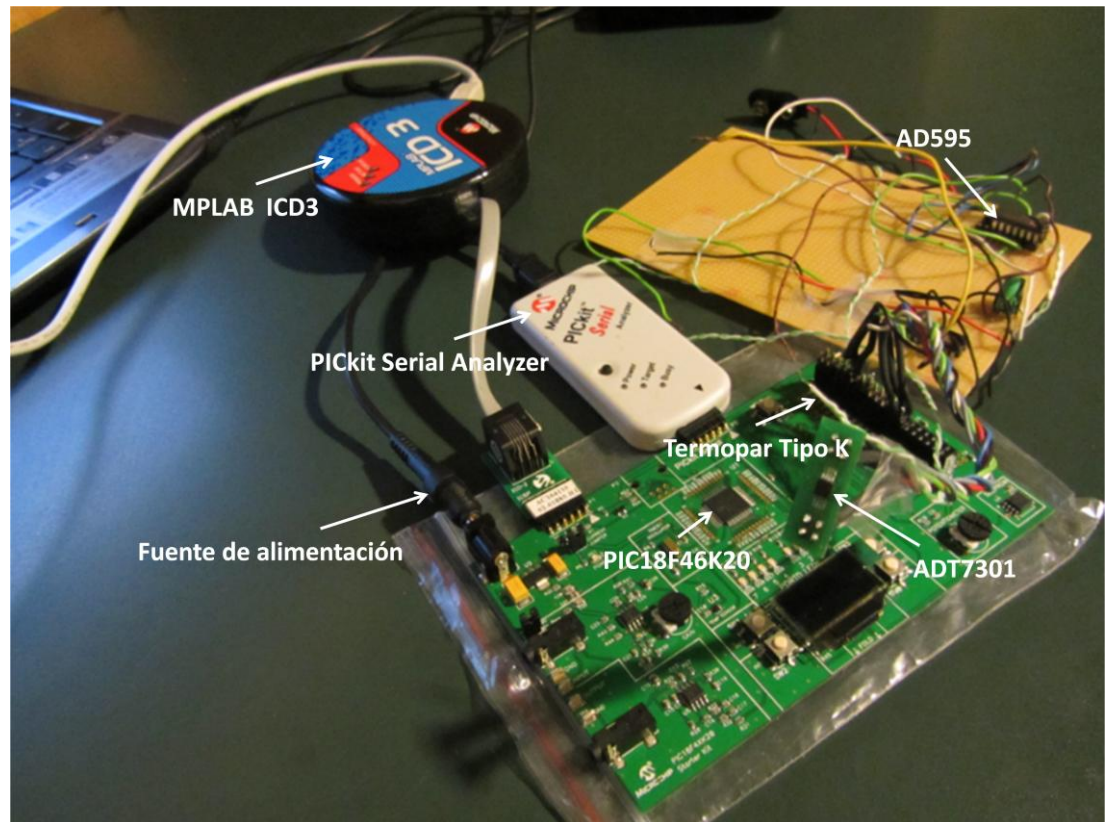
$$\text{Para temperaturas positivas:} \quad T_{positiva} = \frac{\text{valor ADC}}{32} \quad (4.7)$$

Y

$$\text{Para las negativas:} \quad T_{negativa} = \frac{(\text{valor ADC} - 16384)}{32} \quad (4.8)$$

5. Resultados

Para verificar el diseño se ha montado el circuito y se ha comprobado su funcionamiento en la tarjeta de desarrollo, tal y como se muestra en la figura siguiente:



El PICkit serie actualmente soporta los siguientes protocolos [14]:

- I2C™ - Master
- I2C™ - Slave
- SPI - Master
- USART – Asíncrona
- USART – síncrona
- LIN
- MICROWIRE

El dispositivo dispone de, por una lado, un puerto USB que le permite conectarse con el ordenador, y por otro de una interfaz serie, como la que se muestra a continuación, para comunicarse con el microcontrolador.

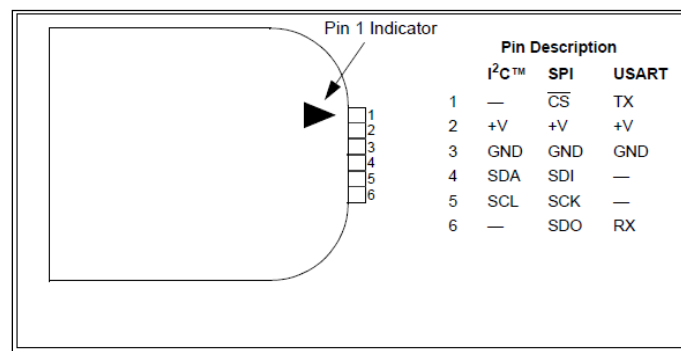


Figura 5.2: Pines de la interfaz serie del PICkit

Como se puede ver en la figura los pines cambian su función, dependiendo del protocolo empleado. En este diseño se ha empleado el protocolo USART (universal synchronous asynchronous receiver transmitter) asíncrono, y así se destina el pin 1 a la transmisión (TX) y el 6 a la recepción (RX), y los pines 2 y 3 a +V y GND respectivamente.

En este protocolo los datos que se envían tienen normalmente el tamaño byte y para sincronizarse los dos equipos (transmisor y receptor) se añade un bit de Start (0) y un bit de Stop (1) para indicar el principio y fin de la transmisión de un byte. Se puede añadir también un noveno bit de datos a los 8 bits del dato, para establecer, por ejemplo, la paridad (Figura 5.3) [14].

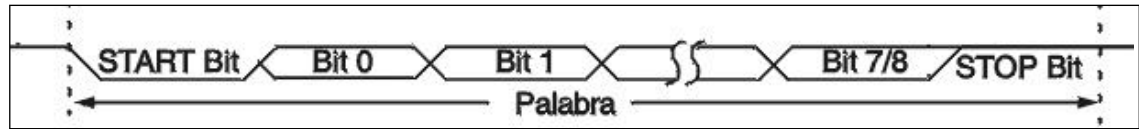


Figura 5.3: transmisión de una palabra

Para establecer este protocolo de comunicación había que configurar el módulo USART del microcontrolador, para trabajar en modo asíncrono, y como únicamente se quiere recibir del PIC y no mandarle datos, éste solo tiene que jugar el papel de transmisor. Para eso se ha tenido que programar primero el generador de baudios. La figura siguiente muestra el diagrama de bloques de este generador así como el diagrama global del USART [4 y 12].

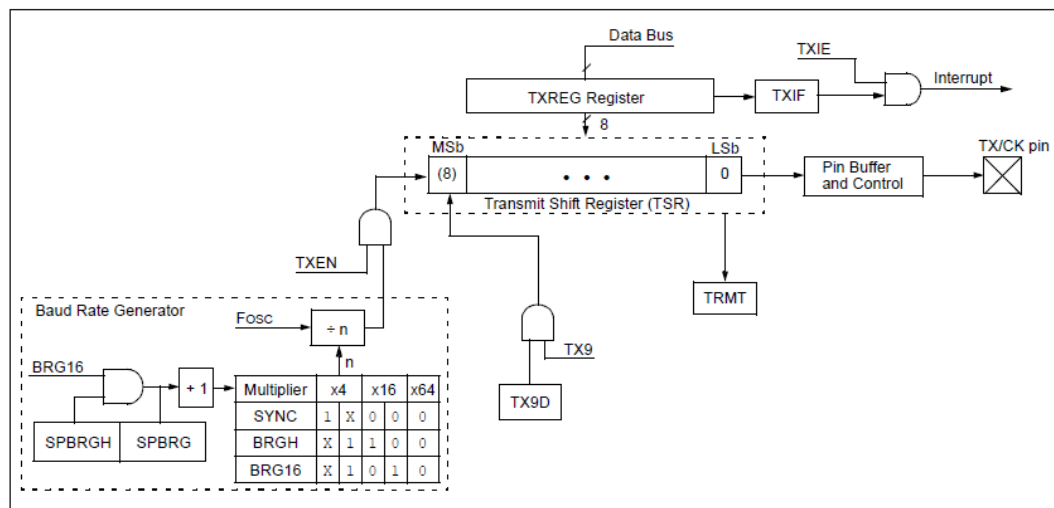


Figura 5.4: Diagrama de bloques del USART

El generador de baudios (BRG) es un temporizador de 8-bits o 16 bits que se dedica a generar la frecuencia requerida. El BRG está controlado por el par de registros SPBRGH: SPBRG, ya que es el que determina el valor de la velocidad en baudios. En el modo asíncrono, el multiplicador del período de la velocidad en baudios está determinada tanto por el bit BRGH del registro TXSTA, así como el bit BRG16 del registro BAUDCON.

Para reducir el error cometido en establecer la velocidad en baudios se recomienda usar el modo de alta velocidad, poniendo el bit BRGH o el BRG16 a nivel alto. La fórmula para calcular dicha velocidad viene dada por la expresión [4]:

$$Velocidad = \frac{Fosc}{4(n+1)} \quad (5.1)$$

Siendo Fosc es la frecuencia del oscilador n es el valor cargado en el par de registros SPBRGH y SPBRG. Por ejemplo, y como se ha hecho en este proyecto, para conseguir una velocidad de 9600 baudios con una frecuencia de 1MHz el valor que hay cargar en los registros antes mencionados es aproximadamente 25. El cálculo del error cometido seria el que sigue:

$$Error (\%) = \frac{V \text{ calculada} - V \text{ requerida}}{V \text{ requerida}} * 100 \quad (5.2)$$

V calculada es la velocidad resultante de cargar 25 en SPBRGH y SPBRG, esta velocidad se puede calcular con la expresión anterior como:

$$V_{calculada} = \frac{1000000}{4(25+1)} = 9615 \quad (5.3)$$

Y por tanto, y reemplazando en la expresión del error, resulta que:

$$Error = \frac{9615 - 9600}{9600} * 100 = 0.16 \% \quad (5.4)$$

Una vez quedan cargados estos registros por los valores que les corresponde. Se pasa a configurar el resto de registros implicados en este módulo. A continuación se muestra el código C de la función que se encarga de programar el USART.

```
void BAUD_config(void)
{
    /*BAUD=9600*/
    SPBRGH =0x00;
    SPBRG  =0x19;           /*[25.04]=25*/
    TXSTAbits.SYNC=0;       /*USART ASync*/
    TXSTAbits.TXEN=1;       /*habilitar transmision*/
    RCSTAbits.SPEN=1;       /*habilitar el puerto serial*/
    TXSTAbits.BRGH=1;       /*High speed*/
    BAUDCONbits.BRG16=1;    /*usar 16-bits*/
}
```

Figura 5.5: Código C de la función BAUD_config()

Como se puede ver en la figura 5.5, se ha establecido una velocidad de 9600, por eso los registros SPBRGH y SPBRG quedan cargados con el valor 0x19 (ver ejemplo anterior). A continuación se indica al microcontrolador que tiene que trabajar en el modo USART asíncrono, poniendo el bit SYNC del registro TXTA a 0 y se habilita tanto la transmisión, poniendo el bit TXEN a 1, así como el puerto serial de este módulo (bit SPEN a 1). Luego se ponen a 1 los bits BRGH y BRG16 del registros TXTA y BAUDCON.

Algunas características del modo USART asíncrono funcionando como transmisor son:

- La transmisión se habilitar mediante el bit TXEN, TXSTA<5>.
- Para transmitir un dato hay que escribirlo en este registro en el registro TXREG.
- Después de haber escrito el TXREG, el dato pasa al registro de desplazamiento TSR, este registro no se carga hasta que el bit de STOP del dato anterior no se ha transmitido.
- Al quedar vacío el TXREG se activa el bit de interrupción TXIF (PIR1 <4>), si está habilitado por el bit TXIE (PIE <4>). (TXIF no se desactiva por software, se desactiva sólo cuando se cargan nuevos datos).
- Hay otro bit el TRMT, TXSTA <1> que muestra el estado del TSR, no produce ninguna interrupción. (Cuando activa TRMT está vacío).
- Para enviar un dato con 9 bits hay habilitar el bit TX9, (TXSTA <6>) y poner el bit que se quiere enviar en TX9D (TXSTA<0>).

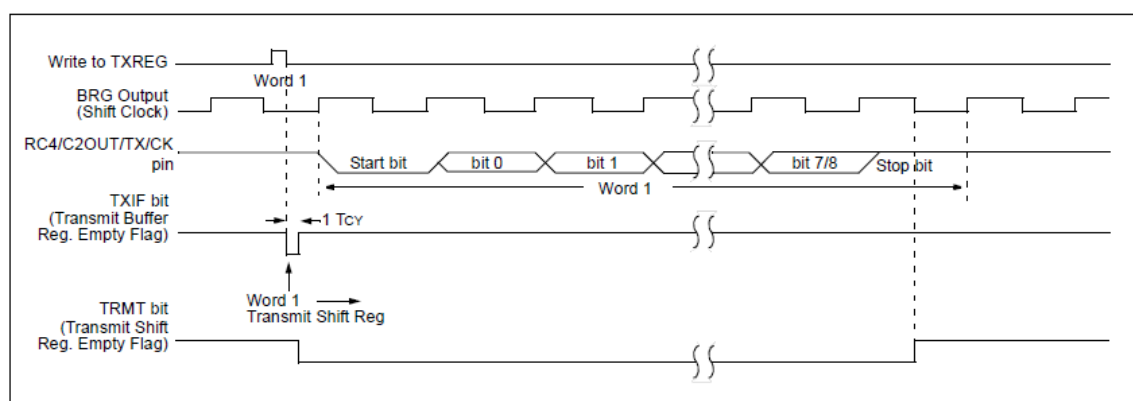


Figura 5.6: Transmisión asíncrona

Nota: se ha configurado el pin RC6 como pin de salida y el RC7 como entrada, para poder conectarlos con los pines RX y TX, del analizador serial PICkit, respectivamente.

A la hora de recibir un dato del microcontrolador, el programa proporcionado por Microchip, el que gestiona el analizador, lo interpreta y lo muestra en la pantalla, dando la opción de ver este dato en el formato hexadecimal o su codificación en el estándar ASCII.

Después de explicar el funcionamiento y la configuración de todos los dispositivos, tanto los dos sensores como el PIC. Se procede ahora a mostrar un resultado obtenido experimentalmente. Donde se ha medido la temperatura ambiente con los dos sensores, con el objeto de comprobar el buen funcionamiento del sistema diseñado. La figura siguiente muestra los resultados obtenidos.

En la figura 5.7, aparecen dos filas, la primera es la que muestra los resultados del sensor compuesto por el termopar tipo K y su correspondiente circuito acondicionador de señal (AD595). En la segunda aparece el resultado de varias medidas realizadas por el sensor ADT7301. Se hace notar que los resultados por ambos sensores se defieren ligeramente debido a, primero el ruido generado por los cables de conexión, y segundo por los errores que introducen los dispositivos en la medida. Y también puede ser porque el termopar está preparado para medir temperaturas superficiales, mientras que el sensor digital es para la medida de temperatura ambiente.

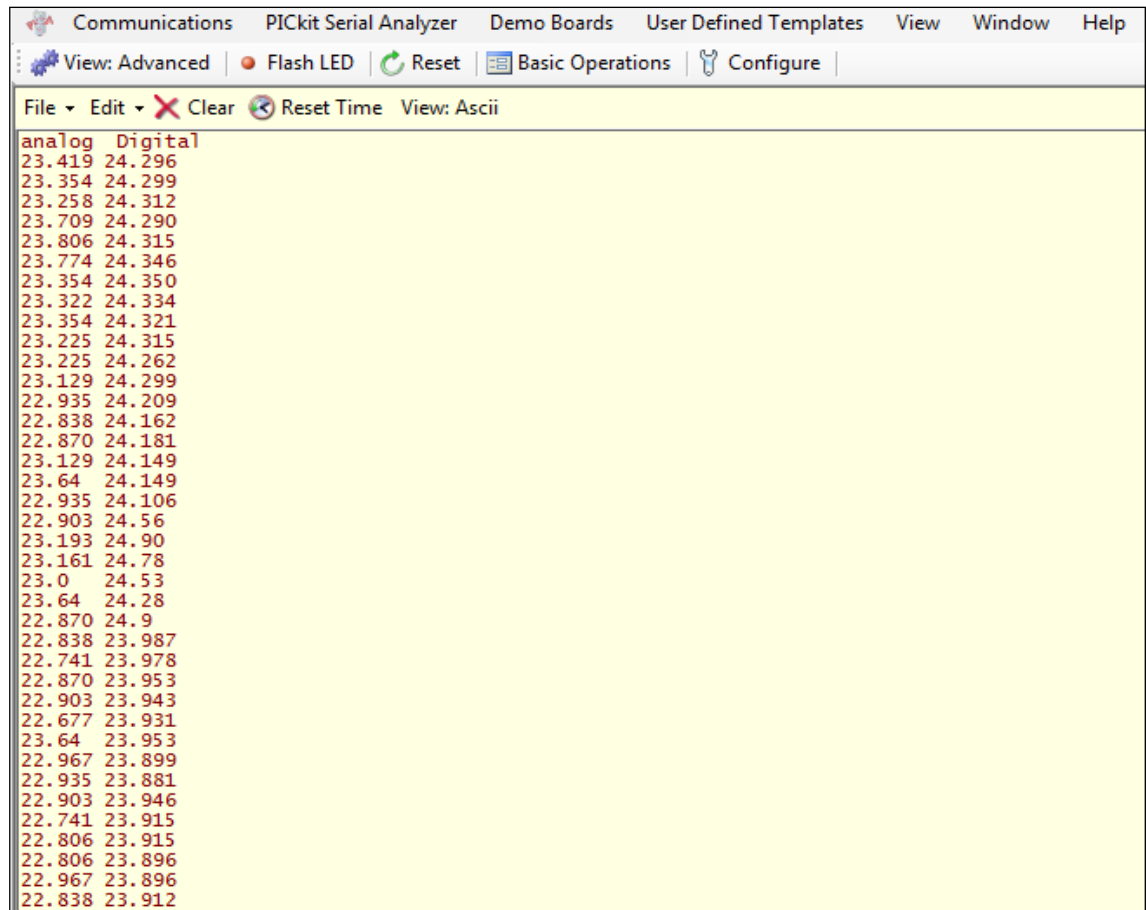


Figura 5.7: Medida experimental

En la fase de comprobación, se ha sometido el circuito a varios experimentos. Podemos, destacar, por ejemplo, la medida de temperatura en el horno, comparando el valor que indica éste con el obtenido con el sistema. También, y para comprobar sobre todo, el funcionamiento del circuito en temperaturas negativas, se ha medido la temperatura en el congelador de la nevera, variando el valor de dicha temperatura y comprobando en cada momento que los resultados obtenidos con el sistema y los que indica el sensor de la nevera coinciden.

Después de todos esos experimentos y al analizar los resultados obtenidos por el sistema, se puede decir que este circuito ha cumplido los objetivos, por los cual ha sido diseñado.

6. Conclusiones

A la vista de los objetivos previstos para este proyecto y los resultados obtenidos, se puede decir que se ha llegado a diseñar un sistema que cumple con las ideas previstas y satisfacer las restricciones fundamentales que han dado origen a este proyecto. Puesto que el sistema ha logrado, por un lado, poder medir un amplio rango de temperatura, y por otro conseguir un tamaño pequeño, lo que le permite una integración fácil en los sistemas donde se le necesita.

También y como se ha comentado anteriormente, en este sistema se intentado consumir la mínima energía posible, empezando primero por escoger los componentes con menor consumo y segundo no dejar que los dispositivos trabajen continuamente, ya que solo se permite el funcionamiento cada dos minutos lo que supone un ahorro muy importante de la energía consumida. Y esto le permite la integración en cualquier sistema, en la cual el factor energía se considera importante. El diseño propuesto puede ser mejorado, convirtiéndolo en un sistema que puede auto-alimentarse, es decir generar el voltaje que se requiere para su alimentación, aprovechando la energía generada por el rotor del motor del sistema en el que esté integrado. Así no hay que preocuparse por cambiar la pila cada vez que ésta se descarga.

Una posible propuesta alternativa puede ser la integración de una bobina, que capture los campos magnéticos, generados por los imanes del rotor y por consiguiente conseguir un voltaje a la salida que pueda regularse mediante un regulador de tensión y de esta manera, se podría sustituir la pila de 3 voltios usada en este proyecto para alimentar el sistema por este circuito.

Otro aspecto que se puede mejorar es la comunicación del sistema con el PC que recoge los datos. Añadiendo, por ejemplo, un transmisor de radio frecuencia a la placa que va enviando los datos obtenidos en tiempo real a un receptor, y éste a su vez al ordenador. Y así, evitar el uso de todo tipo de comunicación por cable con el sistema. Esto, será una ventaja más que se suma al diseño que aumentará la lista de los sistemas que pueden aprovechar el producto.

Para realizar esta función solo hay que elegir con cuidado los módulos a emplear, para que cumplan los objetivos del proyecto, porque se puede aprovechar las funciones desarrolladas anteriormente para establecer el protocolo de comunicación tanto con el transmisor, así como el receptor. Ya que, hoy en día, el mercado ofrece módulos que pueden encargarse de la operación de codificación ya de decodificaciones (normalmente van emparejados). Y por tanto hay que preocuparse solo de la parte del envío y recepción de los datos, no de cómo los codifican/descodifican los dispositivos.

Finalmente, y a nivel personal, este proyecto me ha permitido, dentro de muchas cosas, conocer de muy cerca uno de los dispositivos electrónicos con más presencia en la vida diaria, el microcontrolador. Descubriendo todos los tipos existentes, las herramientas ofrecidas por las diversas empresas que trabajan en este dominio, los lenguajes donde se puede programar y algunas de sus aplicaciones.

También me ha dado la oportunidad de aplicar en el mundo real los conocimientos aprendidos durante la carrera como, por ejemplo, los protocolos de comunicaciones. Ha servido así mismo de repasar y fijar algunos conceptos de varios campos de la electrónica, destacando, por ejemplo, los sensores, tipos y características de memorias, fuentes de alimentación, etc.

7. Bibliografía

- [1] M. Granda Miguel, E. Mediavilla Bolado; *Instrumentación electrónica: Transductores y acondicionadores de señal*, Edit. Universidad de Cantabria, p.495, 2010.
- [2] <http://es.rs-online.com/web/> [última consulta: 18 de julio de 2011].
- [3] J.M. Angulo Usategui; *Microcontroladores "PIC": diseño práctico de aplicaciones. Primera parte: el PIC12F508 y PIC16F84A*, Edit. McGraw-Hill, p.523, 2007.
- [4] <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en026331>
Hoja de características PIC18F46K20 [última consulta: 20 de septiembre de 2011].
- [5] V. González Pelayo; *Diseño e implementación de un medidor de corriente con sensor de efecto Hall basado en un microcontrolador PIC18F452*, Proyecto Fin de Carrera. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Universidad de Cantabria. Santander, 2004.
- [6] <http://www.microchip.com/forums/> [última consulta: 20 de septiembre de 2011].
- [7] http://ww1.microchip.com/downloads/mplab/X_Beta/index.html MPLAB X IDE BETA5 [última consulta: 15 de julio de 2011].
- [8] <http://ww1.microchip.com/downloads/en/devicedoc/51295e.pdf> C18 [última consulta: 20 de septiembre de 2011].
- [9] http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en537580&redirects=icd3 MPLAB ICD3 [última consulta: 30 de julio de 2011].

- [10] www.sparkfun.com/datasheets/IC/AD595.pdf Hoja de características AD595 [última consulta: 7 de septiembre de 2011].
- [11] www.alldatasheet.com/datasheet-pdf/pdf/.../ADT7301ARMZ.html Hoja de características ADT7301 [última consulta: 13 de septiembre de 2011].
- [12] I. Cid Capelo; *Sensor de temperatura y conversor digital en FPGA utilizando VHDL*, Proyecto Fin de Carrera. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Universidad de Cantabria. Santander, 2002.
- [13] www.murata.com/products/power/selection_guide/.../MPDTY102S.pdf Hoja de características MPDTY102S [última consulta: 22 de septiembre de 2011].
- [14] <http://ww1.microchip.com/downloads/en/DeviceDoc/51647C.pdf> *PICkit serial analyzer* [última consulta: 22 de septiembre de 2011].