**Ejemplo\_005\_Timer in**

Objetivo: Reloj basado en el Timer2, usando interrupciones

En el ejemplo\_002, el conteo del Timer2 se revisaba leyendo constantemente la bandera TMR2IF, una manera de evitar esta lectura es generar una interrupción.

La interrupción permitirá que cada vez que la bandera se active(TMR2IF), se ejecute una función en la cual podemos agregar un código deseado. Para este caso se genera una interrupción por tiempo, configurando el timer2 a 1m**s.**

El MCU tiene distintos tipos de interrupción, por puerto, tiempo, muestreo, errores de voltaje, recepción de datos etc. Cada una de estas tiene cierto nivel de prioridad, algunos modificables , esto es útil cuando se tiene diferentes fuentes de interrupción.

**Registros asociados a las interrupciones**

• RCON, Contiene los bits que indican las fuentes del ultimo reset o salida de modo sleep. También contiene el bit de habilitación para niveles de interrupción.

• INTCON,INTCON2,INTCON3: Contienen varios bits de habilitación, prioridad y banderas

• PIR1, PIR2, PIR3, Contienen los bits de banderas individuales para las interrupciones por periféricos.

• PIE1, PIE2, PIE3, Contiene los bits de habilitación para las interrupciones por periféricos.

• IPR1, IPR2, IPR3, Contiene los bits de nivel de prioridad para las interrupciones por periféricos.

**CONFIGURACION DE INTERRUPCIONES**

Procedimiento para activar las interrupciones,

-Configurar el modulo respectivo para este ejemplo el Timer2

-Activar interrupción asociada al módulo (INTCONX, PEIE/GIEL,PIE)

-Definir Prioridad de la interrupción del modulo(INTCONX, IPR)

-Activar interrupciones globales (INTCON:GIE)

La función donde son dirigidas las interrupción se nombra mediante la directiva interrupt

void interrupt NombredelaFuncion()

En ella se debe limpiar las banderas y poner el código respectivo, además se debe comparar banderas para saber las fuentes de interrupción,

if(PIR1&TimerIFG)

Para mayor informacion ver la sección 10.0 INTERRUPTS