

# Experiencia 2b:

## Microcontroladores (PIC)



### 1 Introducción

La presente experiencia tiene por objetivo introducir el manejo de microcontroladores a los alumnos del seminario de diseño -Energías Renovables y uso Eficiente-.

En esta experiencia trabajaremos con un microcontrolador en especial, el PIC16F873A. Su objetivo será aprender las técnicas básicas de programación de estos dispositivos a través de la realización de un semáforo (tal como en la experiencia de LabView), pero con la ventaja de que éste será independiente de un computador.

Estos elementos son bastante versátiles y existen muchos modelos diferentes, adecuándose cada uno de ellos a distintos requerimientos, ya sea el número de funciones a controlar, la cantidad de información a manejar, etc.

En general, los microcontroladores cuentan con las siguientes 5 partes principales:

- Unidad central de procesamiento (CPU) → Lugar donde se ejecutan las instrucciones del programa, varían desde 8 a 64 bits.
- Interfaz de entrada y salida → Corresponden al medio por el cual se recibe y entrega la información, en este caso los "pines".
- RAM → Unidad que permite el almacenamiento (escritura y lectura) temporal de datos, fundamental para el proceso de cálculo.
- ROM → Memoria de solo lectura, lugar de almacenamiento del programa.
- Reloj → Circuito que produce una señal de referencia temporal. Comúnmente incluye un Cristal, elemento que al ser excitado produce una señal repetida a una frecuencia fija.

Luego de esta escueta descripción, tome el PIC y mírelo. Podrá notar que cuenta con muchos "pines". Más adelante veremos que cada uno cumple una cierta función: unos para obtener información, otros para emitirla y otros para alimentación y funciones propias del PIC. Los "pines" varían en número y función dependiendo del modelo de PIC utilizado, por lo que es necesario recurrir (a través del buscador favorito en internet) al datasheet del modelo correspondiente, para así saber cuál es la función de cada una. Más adelante tendrán oportunidad de identificar las distintas opciones de su microcontrolador, para así utilizarlas en la creación del semáforo.

## 2 A trabajar

Para realizar esta experiencia habrá que separar el trabajo en dos partes, la programación y el montaje de circuitos. Para lo primero utilizaremos el programa PICC, que nos permitirá a través del lenguaje C (similar a JAVA) dar instrucciones al microcontrolador.

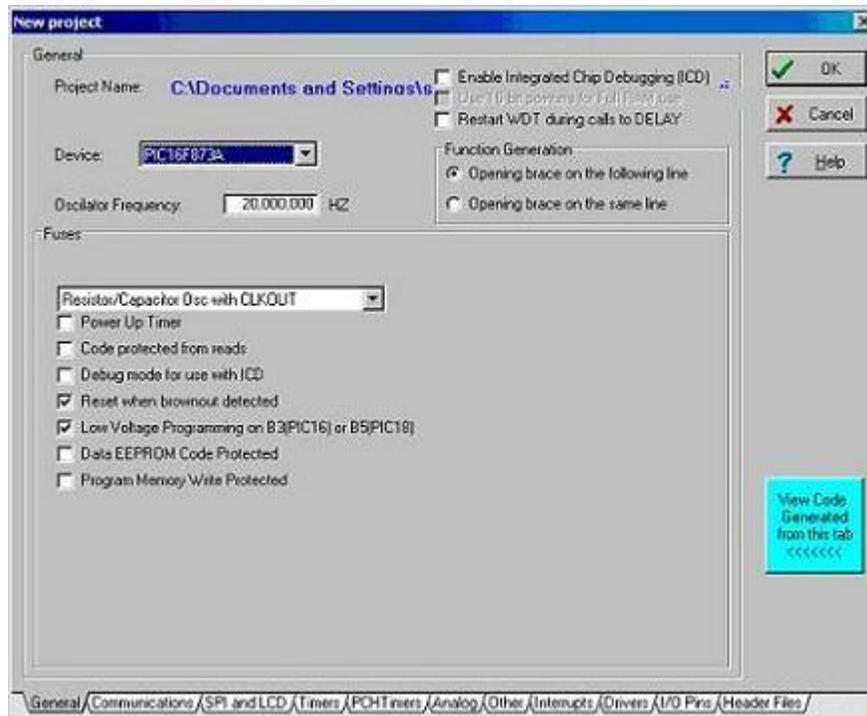
(Ojo: para poder compilar en PICC debe estar en una cuenta administrador, pida a su auxiliar que ingrese a tal cuenta).

Más adelante encontrará la sección relacionada con el montaje del circuito.

### 2.1 PICC

Para poder trabajar, necesitaremos de la ayuda del programa PICC. Los pasos para llegar a el son los siguientes.

- Abrir PICC (C:\Archivos de programa\PICC\pcw),
- Iniciar un nuevo proyecto (Project/New/PIC Wizard),
- Guarde su proyecto ordenadamente (cree una carpeta, pues en un proyecto se usan varios archivos),
- Luego de hacer todo lo anterior aparecerá la siguiente ventana en que debe definir las características de su microcontrolador:



1. Detalles importantes en la pestaña **General**:
  - "Device" define el tipo de PIC.
  - "Oscillator Frequency" depende del cristal (oscilador) utilizado (en este caso de 20MHz).

↳ En el menú dentro de "fuses" se debe realizar otro cambio que UD debe descubrir, relacionado con el oscilador (si no lo descubre pregunte a su auxiliar!).

2. Detalles importantes en la pestaña **Analog**:
  - En esta pestaña el único cambio a realizar consiste en definir "Analog Pins" en → A0 A1 A2 A3 A5 E0 E1 E2 Ref=Vdd

- Con todo lo anterior listo, finalmente presiona



### 2.1.1 Programación

Al finalizar los pasos de la parte anterior aparecerá una ventana con un bloque de instrucciones. La primera es un "void main()", instrucción que indica el bloque de ejecución principal. Luego, dentro de este bloque, aparecerá lo siguiente:

```
setup_adc_ports(ALL_ANALOG);  
setup_adc(ADC_CLOCK_INTERNAL);  
setup_spi(FALSE);  
setup_counters(RTCC_INTERNAL,RTCC_DIV_1);  
setup_timer_1(T1_DISABLED);  
setup_timer_2(T2_DISABLED,0,1);  
setup_comparator(NC_NC_NC_NC);  
setup_vref(FALSE);
```

Estas instrucciones están asociadas a lo que se ajustó en un principio al iniciar el nuevo proyecto. Entre la última de estas instrucciones y el corchete de finalización del main podrá definir los comandos para que su PIC haga lo que debe. Podrá definir variables, realizar ciclos y muchas de las cosas que aprendió en JAVA.

En las próximas líneas se enumerarán algunas de las instrucciones más importantes para realizar los programas en esta experiencia.

1. **output\_low(PIN\_BX); & output\_high(PIN\_BX);** → Estas instrucciones permiten establecer el estado 0 o 1 del pin BX (X puede variar entre 0 y 5), es decir, que el pin esté a 0 o 5 Volts. Esta función se utiliza para activar o desactivar algún elemento asociado al pin BX. Más adelante usted deberá descubrir cuál de las dos instrucciones anteriores sirve para encender y cuál para apagar las luces de su semáforo.

2. **while(1){...}** → Respecto a esta instrucción no hay mucho que decir. El 1 es análogo a TRUE, es decir, este ciclo hará lo que usted defina indefinidamente.
3. **delay\_ms(X);** → Para poder producir un retraso en el hilo de ejecución de su programa, utilice este comando. Si necesita que algo demore una cantidad de tiempo hasta que suceda, reemplace X por esa cantidad de tiempo en MILISEGUNDOS.
4. **long nombre\_variable;** → Instrucción para definir variables numéricas de tipo long.
5. **set\_adc\_channel(X); & read\_adc();** → Estos comandos se relacionan para poder obtener información. La primera define el pin (AX) que estará leyendo y la segunda recupera la información. read\_adc() devuelve un long que variará entre 0 y 255 (16 bits) directamente proporcional al voltaje (entre 0 y 5V) al que esté sometido el pin AX.

Hay que resaltar que cada instrucción hay que finalizarla con ;.

### 2.1.2 TOPMAX

Luego de realizar su programa, habiéndolo compilado y estando todo OK, estará en todo el derecho de preguntarse ¿cómo pasará mi programa del PC al PIC? Bueno, eso se soluciona con el TOPMAX, un elemento que permite justamente eso, introducir programas en el PIC. Los pasos a seguir son los siguientes:

- Encuentre el computador que tiene asociado el TOPMAX, si existe la posibilidad de trasladarlo al computador en que usted está trabajando, mejor aún.
- Coloque el PIC en la posición que corresponde en el topmax (deje que lo ayuden la primera vez, para no cometer errores que podrían inducir la muerte del PIC).
- Encienda el topmax (en la parte posterior está el switch).
- Busque el programa asociado al topmax en el computador (programas/EETools/Max Loader)
- Al tener abierto el programa, lo primero que hay que hacer es elegir el PIC con el que estamos trabajando. Busque el botón SELECT.
- Luego debe cargar el programa compilado en PICC. Para ello busque el botón LOAD.
- Cuando se abra el menú de programas, busque su programa y antes de abrirlo, elija en el menú de tipo de archivo la opción ALL HEX FILE(\*.hex,...). Ahora sí, cargue su programa.

- Nuevamente en la interfaz, busque el botón OPTIONS. Debe fijarse que todas las opciones que estén ENABLE, queden DISABLE. OK.
- Finalmente presione AUTO para cargar el programa.
- Cuando haya cargado, PRIMERO APAGUE el topmax y luego quite el PIC.

## 2.2 El Circuito

En esta parte confeccionarán el circuito donde irá ubicado el PIC.

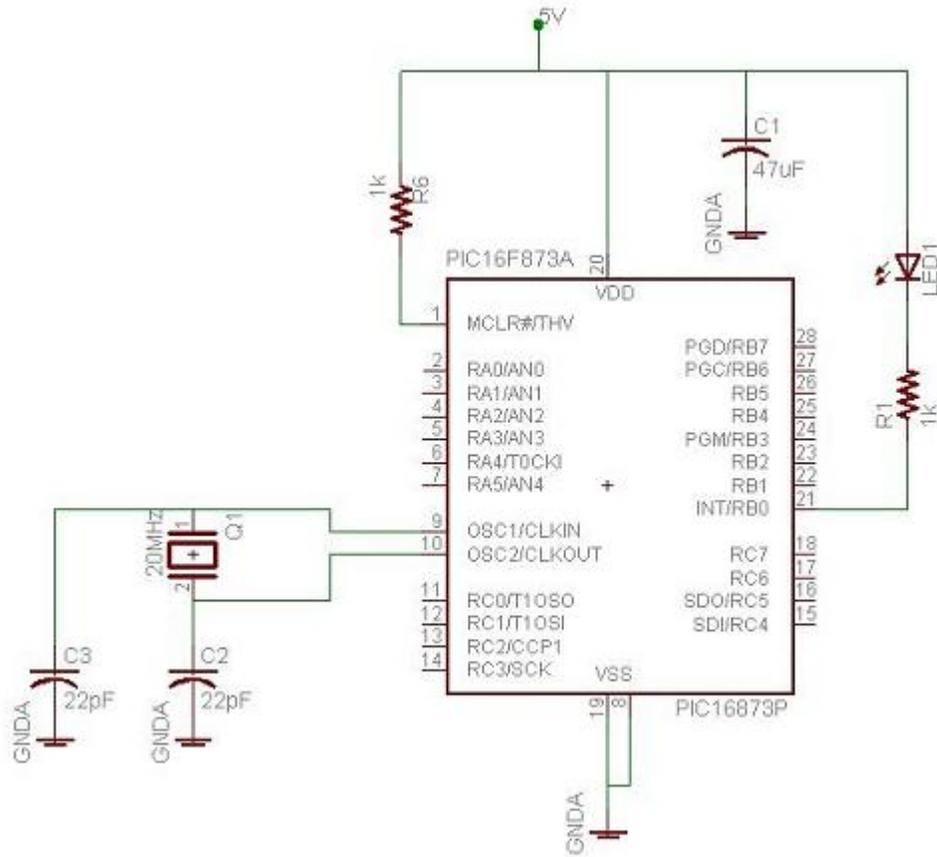
### 2.2.1 Materiales

Para esta experiencia deberá contar con los siguientes materiales (proporcionados por su auxiliar de turno). Preocúpese de que estén todos y en buenas condiciones.

- 1x PIC16F873A
- 4x Resistencia  $1[k\Omega]$
- 1x Resistencia  $1.2[k\Omega]$
- 3x Led (rojo, verde y amarillo)
- 1x Condensador Electrolítico  $47[\mu F]$
- 2x Condensador  $22[pF]$
- 1x Potenciómetro de  $5[k\Omega]$
- 1x Oscilador 20MHz
- 1x Protoboard
- 1x Fuente

## 2.2.2 Circuito Básico

Para empezar, debe montar el circuito mostrado en la siguiente figura.



Como salta a la vista, el rectángulo con los pines numerados corresponde al PIC. Existe un único LED conectado al pin número 21, que para efectos de programación corresponde al PIN B0. En los pines 9 y 10 está conectado el cristal junto con un par de condensadores. Estos cumplen la función de reloj, la que se explicó al comienzo.

→ Cuando esté listo con el montaje su misión será la siguiente:

Cree un programa que permita prender y apagar el Led en intervalos de 3 segundos (osea éste debe permanecer prendido durante 3 segundos y luego apagado por otros 3), usando las instrucciones explicadas en la parte de programación (HINT: solo necesita las instrucciones 1, 2 y 3 para realizar lo que se le pide). Cuando esté seguro de que tiene un buen programa, compílelo y cárguelo en el PIC.

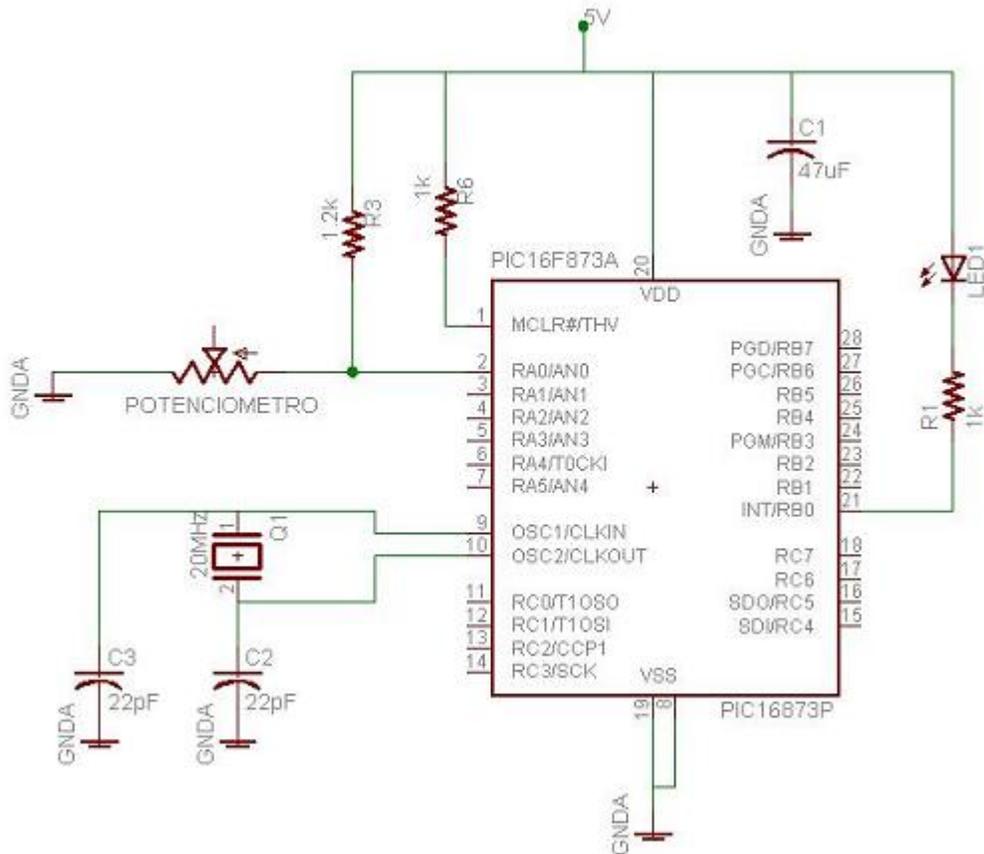
¿Funciona?

Si la respuesta es Sí, mis felicitaciones y pase al siguiente circuito.

Si es No, debe ser capaz de detectar el problema, ya sea de programación o de montaje del circuito y solucionarlo. Si aún no lo logra, recurra a su auxiliar.

### 2.2.3 Circuito Básico 2.0

Este nuevo circuito se diferenciará del anterior en que contará con un sistema de regulación del intervalo de tiempo de encendido y apagado del LED. Para eso deberá montar el siguiente circuito (no es necesario desarmar el anterior!).



Los cambios que se le realizaron al primer circuito permiten tener un parámetro variable (potenciómetro) para así poder variar la frecuencia de encendido del LED. (Hint: acá necesitará las instrucciones 4 y 5 de la parte de programación).

### 2.2.4 Circuito Final

Lo que se busca en este circuito es unir todo lo aprendido. Ud debe conectar tres LEDs (semáforo) y controlar el tiempo que permanecen encendidos (es igual para todos, pero variable). Deberá agregar 2 LEDs al circuito y modificar su programa adecuadamente para lograrlo.

