



Universidad de Chile
Facultad de Cs. Físicas y Matemáticas
Departamento de Ingeniería Eléctrica
Taller de Proyectos-Energías Renovables y su uso eficiente

Experiencia Introdutoria 2

Guía Teórica-Práctica

Profesor: Rodrigo Palma

Autores: Ignacio Polanco
Jannik Haas
Carlos García
Daniel Aparicio

Introducción

La presente guía tiene como objetivo presentarle dos herramientas de gran potencial que podrán ser utilizadas en el desarrollo de su proyecto durante el semestre. Claramente, debido al poco tiempo disponible para realizar la experiencia, no será necesario que entienda el por qué de algunas cosas, sino que, lo importante será que visualice el uso que pueda darles.

Teoría 1: Micro controladores

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los actuadores, que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S (entradas-salidas) sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador que consiste en un sencillo, pero completo computador contenido en el corazón (chip) de un circuito integrado.

Un micro controlador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM / PROM / EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD (Conversores Analógico/Digital), CDA (Conversores Digital/Analógico, etc.)
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que incorporan un microcontrolador para control, disponen de las siguientes ventajas:

1. Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
2. Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos, disminuye el riesgo de averías y se precisan menos ajustes.

3. Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra.
4. Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El micro controlador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado.

Ante lo útil y cómodo de su utilización, el campo actual de su aplicación incluye la mayoría de los procesos que requiere algún tipo de automatización, ya sea desde un juguete hasta el control de alguna aplicación de una nave espacial. Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes.

A continuación se presenta una figura esquemática con los puertos (o pines) del micro controlador PIC 16F873-A, que se utilizará para la experiencia:

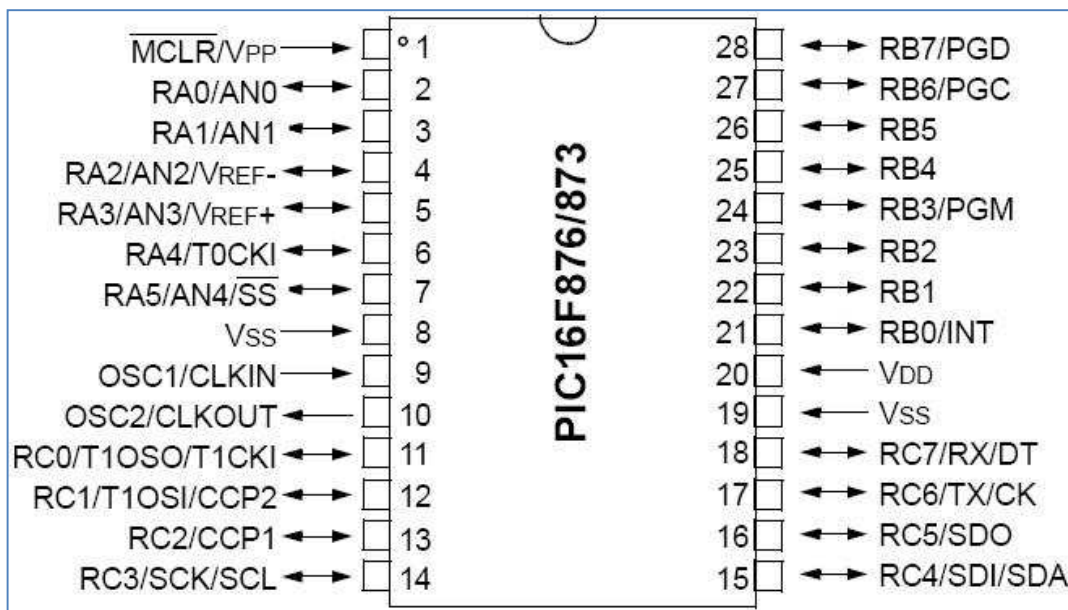


Figura 1: Esquema de pines, PIC16F873A

Básicamente, desde el pin 2 (o RA0) hasta el 7 (o RA5) son pines de I/O (entrada/salida) análoga, es decir, pueden enviar o recibir diferencias de potencial **entre** 0 y 5 [V]. En cambio, del pin RB0 al RB7, son pines de I/O discretas, es decir, pueden enviar o recibir diferencias de voltaje 0 ó 5 [V]

PARTE 1

a) Circuito de alimentación del microprocesador

Materiales

- 1 PIC 16F873A
- 1 Resistencia de 1[KOhm]
- 1 Condensador Electrolítico de 47[uF]
- 2 Condensadores Cerámicos de 22[pF]
- 1 Oscilador de 20 [MHz]
- 1 Fuente de Voltaje

Para que el PIC funcione es necesario que disponga de un circuito polarizador, es decir, un circuito que le entregue requerimientos energéticos básicos. El esquema del circuito del PIC con el que trabajará es el siguiente:

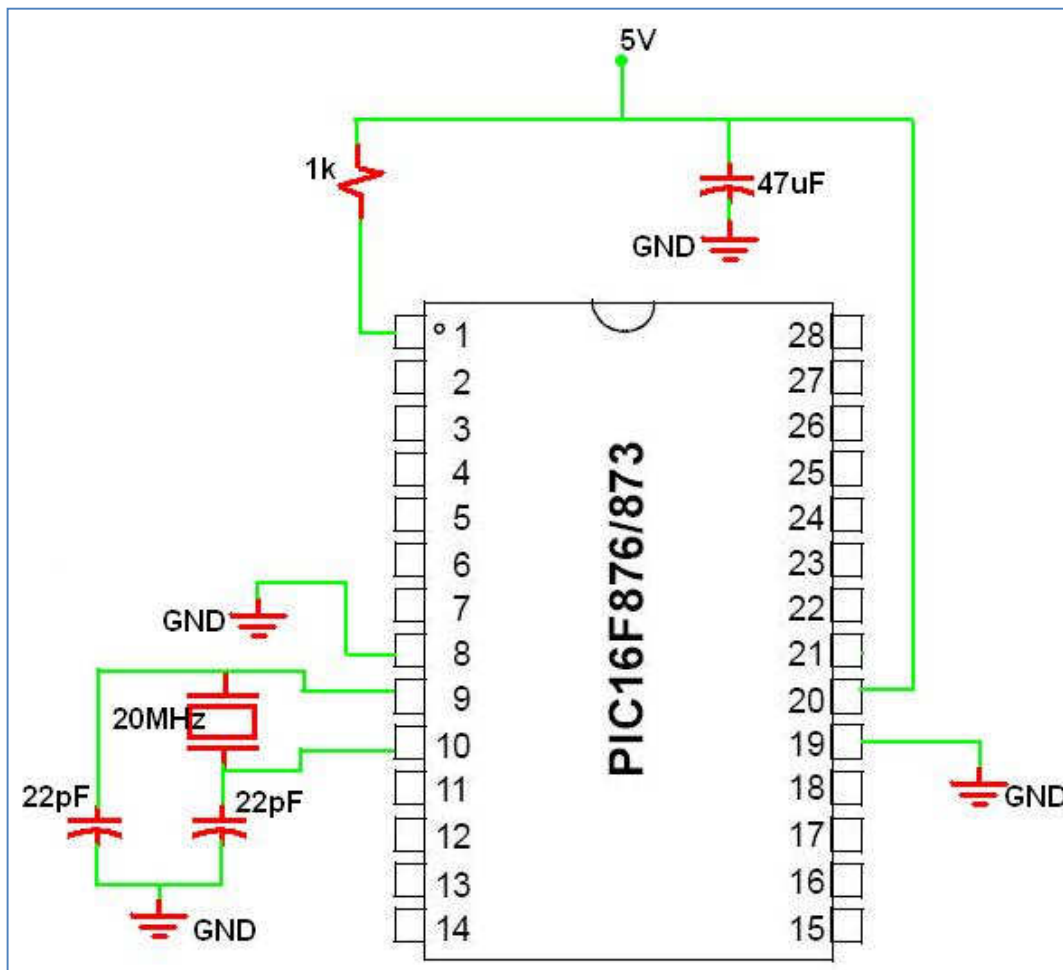


Figura 2: Circuito de Polarización

ARMELO!

b) Prueba del circuito de alimentación

Una vez que esté seguro de que su circuito está correctamente armado, agregue los elementos que se muestran y encienda la fuente (**antes de conectarla a la protoboard, ajústela a 5V**). El microcontrolador tiene puesto un programa de prueba*. Este consiste en que al alimentar el pin A0 (consulte el diagrama de pines) con +5 [V] en el pin B0 se generará una diferencia de potencial de +5[V] y medio segundo después será de 0[V], lo que encenderá y apagará un LED, conectado a tierra, según la figura siguiente:

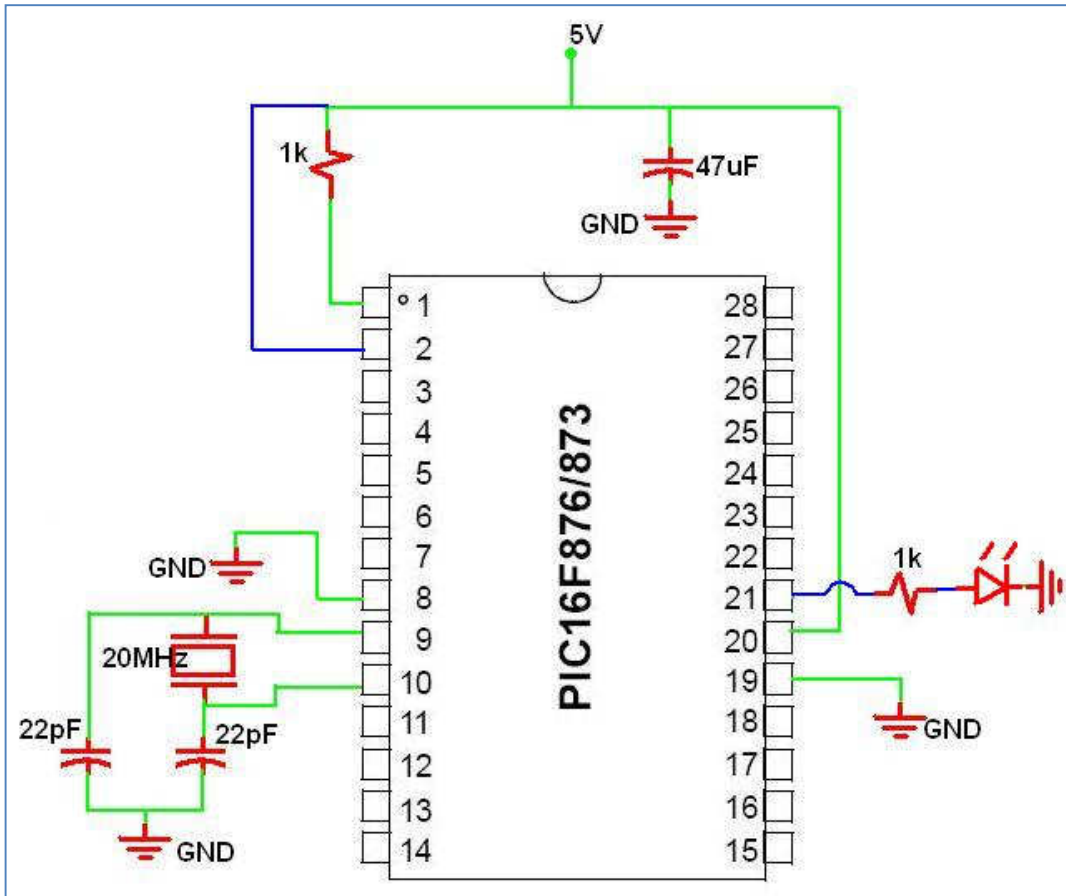


Figura 3: Circuito de Prueba

Si todo funciona correctamente, avísele a su ayudante, para evaluarlo. Luego, continúe con la actividad 2.

*El programa de prueba consiste de las siguientes líneas:

```
// cargar librerías
#include "C:\..."
#use delay(clock=2000000)

//declaracion de las variables
long V=0;

void main() {
    // configuración del proyecto
    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);

    //definición puerto A_X de lectura
    set_adc_channel(0);

    while(1){
        // lectura de información en A0
        V=read_adc();

        //Ver si A0 está sometido a 5v, si es cierto se hace parpadear al LED en B0
        if(V>250){
            output_high(PIN_B0);
            delay_ms(500);
            output_low(PIN_B0);
            delay_ms(500);
        }
    }
}
```

PARTE 2: Programación del PIC

Ahora que está seguro que su circuito funciona, el PIC está listo para recibir otras órdenes, por lo tanto, procederá a reprogramar el micro controlador. Para esto, abra el programa PICC, que es un compilador de C, en el computador asignado, luego vaya a **PROYECT, NEW, PIC WIZARD**, seleccione o cree una carpeta de su agrado en el escritorio y póngale un nombre a su trabajo. En la ventana emergente configure las siguientes opciones:

- Seleccione el modelo de PIC que utilizará en **DEVICE**.
- En la zona de **FUSES**, deselectione todas las casillas.
- Donde dice **Resistor/Capacitor Osc with CLKOUT**, seleccione **High speed Osc**.
- En la pestaña **I/O** verifique que los pines del PIC que se van a utilizar efectivamente estén configurados como entradas o salidas, según correspondan (al menos 3 digitales BX como output y 1 análogo AX como input).
- En la pestaña **Analog** define al menos un pin AX como análogo (por ejemplo "A0 A1 A2 A3 A4").
- Acepte los cambios con **OK**.

Verifique que se hayan generado las siguientes líneas de código:

```
#include "C:\Documents and Settings\Administrador\Escritorio\pic\pic.h"
#use delay(clock=20000000)

void main()
{
    setup_adc_ports(ALL_ANALOG);
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_spi(FALSE);
    setup_timer_0(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);
}
```

Figura 4: Código Inicial Parte2

Agregue las líneas que falten.

El programa PICC es un compilador basado en lenguaje C, similar a JAVA, por lo que no será difícil entenderlo. Con este programa podrá escribir, revisar y generar un archivo con la programación necesaria para que el PIC haga lo que desee. A continuación se explica se explican algunas líneas básicas de código:

1. **#use delay**: Agrega la librería para manipular los tiempos de ejecución del PIC.
2. **setup_....**: Representa la configuración y asignación de puertos, tiempos y referencias del PIC.
3. **delay_ms(X)**: Genera un retardo en el hilo de ejecución del programa de X [ms].
4. **set_adc_channel(X)**: Define el pin (AX) que estará leyendo
5. **Read_adc()**: Recupera la información que se está leyendo. Devuelve un long que variará entre 0 y 255 (16 bits) directamente proporcional al voltaje (entre 0 y 5 [V]) al que esté sometido el pin AX (definido anteriormente).

6. **Output_high(Pin_BX)**: genera una diferencia de potencial igual a 5V respecto a tierra (deja pasar corriente)
7. **Output_low(Pin_BX)**: genera una diferencia de potencial igual a 0V respecto a tierra (no deja pasar corriente)

Otras líneas útiles (para otras situaciones) son:

8. **Getchar()**: Es una función que espera un carácter por el pin C7(serial).
9. **Output_bit(PIN_Bx,y)**: Genera un potencial de 5[V] ó 0[V] en el puerto discreto designado (por ejemplo si desea tener 0[V] en el puerto B0, entonces x=0 e y=0).
10. **#include <string.h>**: Agrega la librería para operar con Strings.
11. **#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)**: Agrega la librería para la comunicación serial del PIC con otro equipo, en particular, PIN_C7 es el que recibe una transmisión serial.

a) Circuito de control de voltaje

Materiales

- Circuito de alimentación del PIC
- 2 Fuentes de Voltaje
- 3 LED
- 3 resistencias de 1[KOhm]

A continuación se programará el PIC, para que, cuando el pin de entrada (definido por usted) recibe un voltaje entre 0 y 3V se encienda un LED verde, cuando se encuentra entre 3 y 4 uno amarillo y uno rojo cuando el voltaje está entre 4 y 5. Lo anterior tiene muchas aplicaciones, por ejemplo para ver si se puede conectar un LED sin quemarlo, para controlar el estado de carga de una batería, para supervisar visualmente la intensidad de un transductor de señal (los transductores de señal son dispositivos electrónicos que *traducen* alguna magnitud física en voltaje, por ejemplo ruido, caudal, intensidad de luz, radiación, temperatura...). De esta forma es fácil darse cuenta lo que está ocurriendo dentro del circuito. Para muchas otras aplicaciones, los LEDs usados en esta experiencia, claramente pueden representar el accionamiento de otros dispositivos, tales como la energización de un motor, bomba, válvula, contador, etc.

Ayuda para lo anterior:

- Siga las instrucciones mostradas anteriormente para la creación de un nuevo proyecto
- Acuérdesse del número de pin de entrada definido
- Acuérdesse de los números de pines de salida definidos
- Programe en el *main* las acciones condicionales
- Adecúe el circuito anterior para este programa

Una vez seguro de su código, compílelo haciendo clic en el botón COMPILE. Si todo está bien, guarde su archivo en la carpeta.

Para cargar su programa en el PIC, será necesario utilizar un dispositivo llamado TOPMAX (Pídaselo a su ayudante, si no sabe cuál es). Los pasos a seguir son los siguientes:

1. Encuentre el computador que tiene asociado el TOPMAX, si existe la posibilidad de trasladarlo al computador en que usted está trabajando, mejor aún.
2. 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).
3. Encienda el TOPMAX (en la parte posterior está el switch).
4. Busque el programa asociado al TOPMAX en el computador (programas/EETools/MaxLoader).
5. Al tener abierto el programa, lo primero que hay que hacer es elegir el PIC con el que está trabajando. Busque el botón SELECT.
6. Luego debe cargar el programa compilado en PICC. Para ello busque el botón LOAD.
7. 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,...).
8. Ahora sí, cargue su programa.

9. Nuevamente en la interfaz, busque el botón OPTIONS. Debe fijarse que todas las opciones que estén ENABLE, queden DISABLE. OK.
10. Finalmente presione AUTO para cargar el programa.
11. Cuando haya cargado, PRIMERO APAGUE el TOPMAX y luego quite el PIC.
12. Finalmente ajuste el circuito anterior al programa hecho.

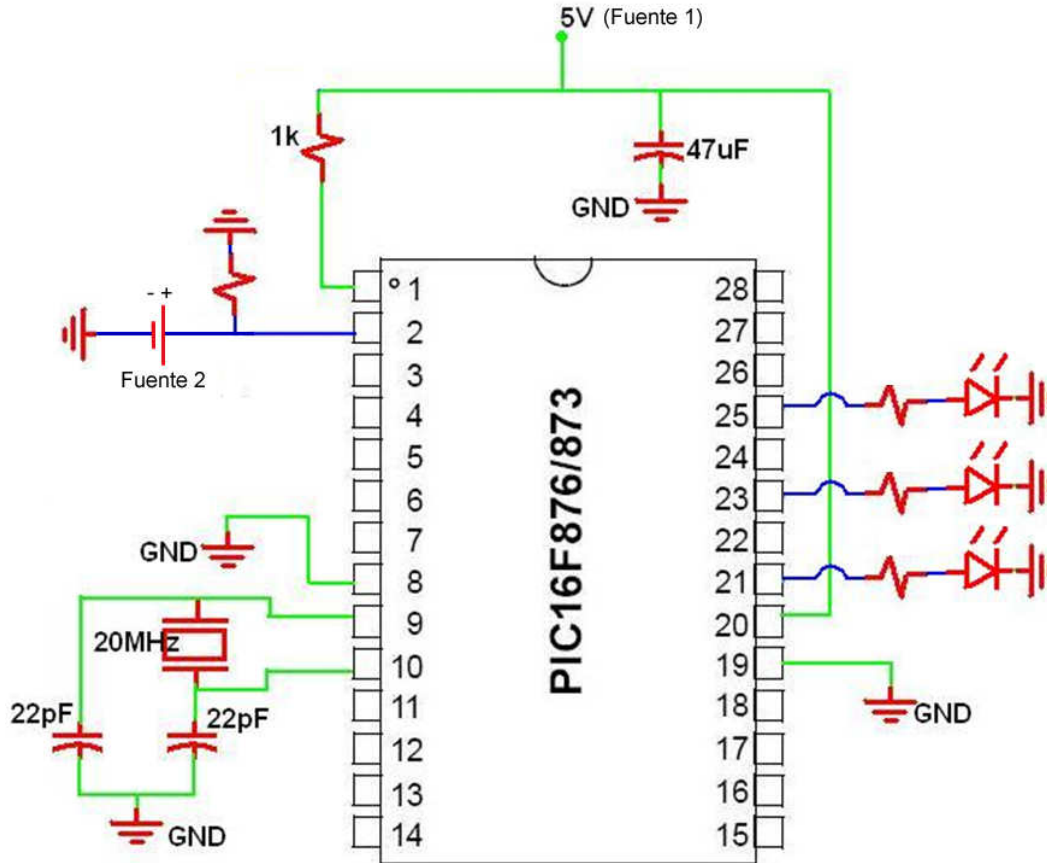


Figura 5: Ejemplo de circuito de control de voltaje

- En la figura se muestra un posible circuito para el control de voltaje. Notar que los pines de salida usados fueron los digitales B0, B2 y B4. El pin de entrada es el análogo A0.