

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**  
**PRACTICA #1**

**Nombre:** Sistema mínimo basado en microcontrolador 16F84A.

**Objetivo:** Implementar el circuito mínimo basado en microcontrolador 16F84A para el desarrollo de varias aplicaciones didácticas. Además deberá leer un dato de 5 bits por el puerto A y enviarlo a través del puerto B.

**Material:**

- 1 microcontrolador 16F84
- 1 XTAL de 4MHz
- 1 R de 10K $\Omega$
- 10 R de 470 $\Omega$  o 330 $\Omega$
- 2 C de 22pF
- 1 C de 100 pF
- 1 push button
- 8 Led's
- 1 dip switch de 8 líneas
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores
- 1 software MPLAB

**PROCEDIMIENTO**

- 1) Armar el siguiente circuito eléctrico basado en microcontrolador 16F84A, **además conectar un dip switch en el puerto A con sus respectivas resistencias y conectar 8 LED's en el puerto B.**
- 2) Programar el microcontrolador utilizando el MPLAB o el software adecuado para el programador.
- 3) Probar el circuito introduciendo diferentes datos binarios.

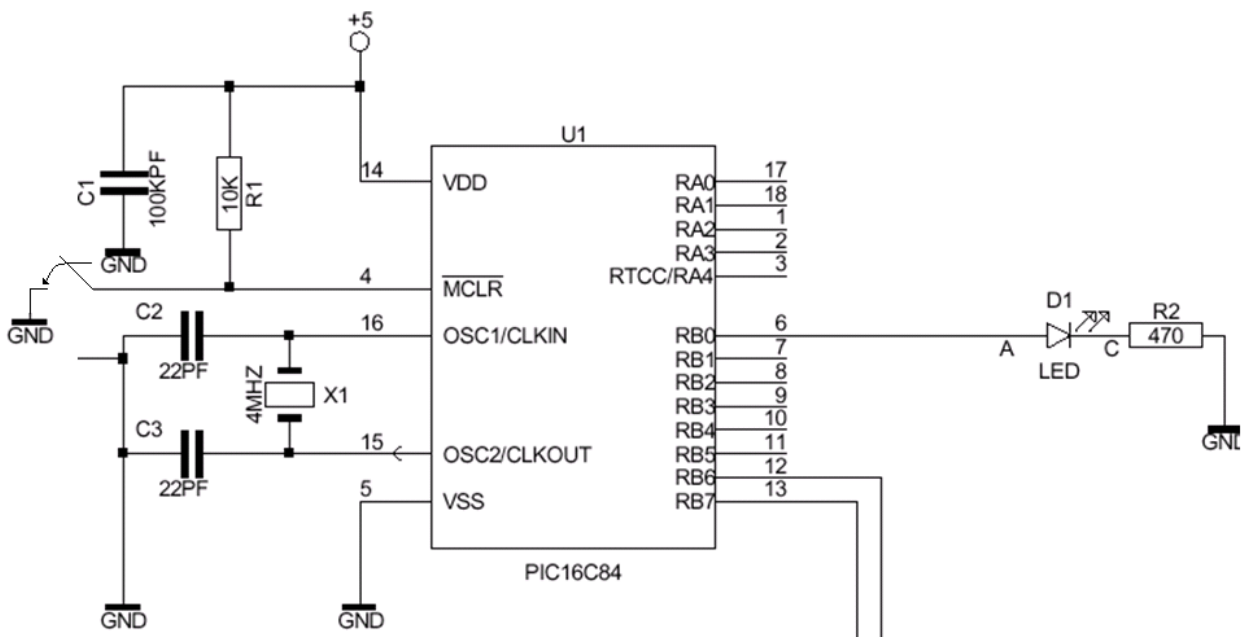


Figura 1) Diagrama eléctrico del sistema mínimo basado en microcontrolador 16F84A.

Elaboró: M.C. Everardo Inzunza González

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**  
**PRACTICA #2**

**Nombre:** Operaciones aritméticas con el microcontrolador PIC 16F84A.

**Objetivo:** Desarrollar dos programas para el microcontrolador PIC 16F84A, de tal forma que realice las siguientes operaciones aritméticas:

$$PB = (5PA - 4) / 2$$
$$PB = \sqrt{PA}$$

Donde, PA = Port A, PB = Port B del microcontrolador PIC 16F84A.

**Material:**

- 1 Microcontrolador 16F84A
- 1 XTAL de 4MHz
- 1 R de 10KΩ
- 10 R de 470Ω o 330Ω
- 2 C de 22pF
- 1 C de 100 pF
- 1 Push button
- 8 Led's
- 1 Dip switch de 8 líneas
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores PIC's
- 1 Software MPLAB v7.40 o superior

**Diagrama a bloques**

**Paso # 1**

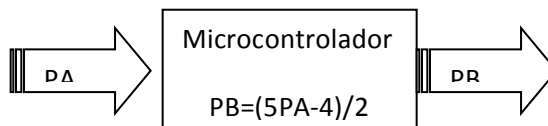


Figura 1) Diagrama a bloques del sistema mínimo para realizar la operación aritmética  $PB = (5PA - 4) / 2$ .

**PASO #2**

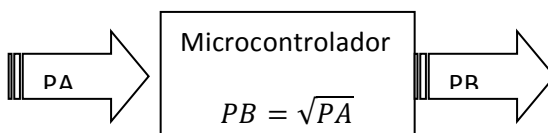


Figura 2 ) Diagrama a bloques del sistema mínimo para realizar la operación aritmética  $PB = \sqrt{PA}$ .

**Nota:** En las entradas de datos del microcontrolador PIC, deberá conectar un dip switch y en la salida LED's para visualizar en código binario el resultado de las operaciones.

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**  
**PRACTICA #3**

**Nombre:** Comparación de registros con el microcontrolador 16F84A.

**Objetivo:** Desarrollar un programa para el microcontrolador PIC 16F84A, para que esté leyendo los datos de entrada a través del puerto A y realice las siguientes tareas:

- a) Cuando el dato en el puerto A sea mayor que 27, únicamente deberá activar en ALTO la salida RB0.
- b) Cuando el dato en el puerto A, sea igual 27, únicamente deberá activar en ALTO la salida RB1.
- c) Cuando el dato en el puerto A, sea menor que 27, únicamente deberá activar en ALTO la salida RB2.

**Nota:** Observe que solamente deberá estar una salida activa en ALTO, dependiendo del resultado de la comparación.

**Material:**

- 1 Microcontrolador 16F84A
- 1 XTAL de 4MHz
- 1 R de 10KΩ
- 10 R de 470Ω o 330Ω
- 2 C de 22pF
- 1 C de 100 pF
- 1 push button
- 8 Led's
- 1 Dip switch de 8 líneas
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores
- 1 Software MPLAB

**Diagrama a bloques**

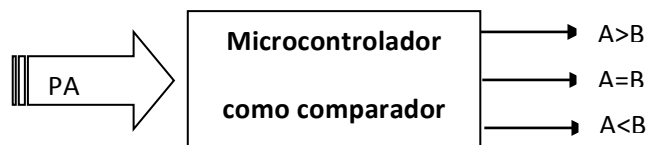


Figura 1) Diagrama a bloques del sistema mínimo para realizar comparaciones.

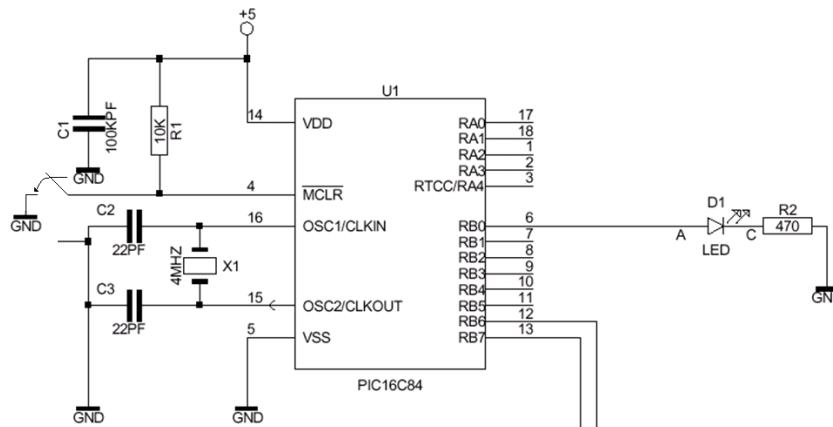


Figura 2) Diagrama eléctrico del sistema mínimo basado en microcontrolador 16F84A.

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**

**PRACTICA No. 4**

**CONTROL ON/OFF DE TEMPERATURA BASADO EN MICROCONTROLADOR**

**Objetivo:**

Diseñar y construir un circuito digital basado en microcontrolador PIC 16F84A para controlar la temperatura cuando este fuera del rango 18-35°C.

**FUNCIONAMIENTO DEL CIRCUITO.** Cuando la temperatura sea mayor a 35°C se encienda un ventilador 12 Vcd o 127Vca para enfriar el sistema. Cuando la temperatura este por debajo de 18°C se deberá activar un sistema de calentamiento. Cuando la temperatura este dentro del rango normal (18-35°C), el microcontrolador deberá encender un LED verde conectado en cualquiera de los pines.

**Material:**

- Convertidor Analógico a Digital ADC0804
- PIC 16F84
- Oscilador LM555
- Amplificador operacional LM741
- Amplificador operacional (comparador de precisión)
- Sensor de temperatura LM35
- Ventilador
- Relays, u optoacopladores
- TRIAC's
- 8 Diodos emisores de luz (LEDS)
- 1 resistor variable de 1 Kohms
- 1 resistor de 16 Kohms, 2 Kohms, 75 ohms, 10 Kohms
- 2 resistores de 500 Kohms
- 8 resistores de 330 ohms
- 2 capacitores 0.1  $\mu$ F, 1  $\mu$ F
- 1 capacitor de 150 pF
- Protoboard
- Fuente de poder de  $\pm$ 5 volts
- Alambres de diferentes longitudes

**Procedimiento:**

Se sugiere integrar esta práctica por etapas, es decir, primeramente configurar el sensor y verificar que mida bien la temperatura, posteriormente configurar el ADC y diseñar el circuito de acondicionamiento de señal. Hacer pruebas de conversión de A/D. Identificar bien el valor binario correspondiente a 18°C y 35°C. Relizar el programa en ensamblador para hacer las comparaciones y la toma de decisiones.

**Diagrama a bloques**

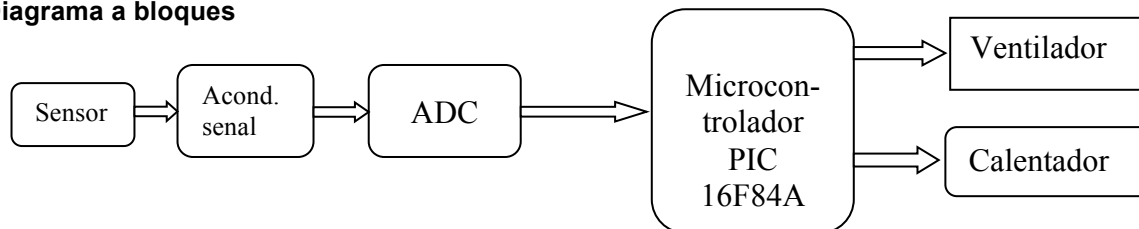


Figura 1) Diagrama a bloques del controlador de temperatura basado en microcontrolador PIC 16F84A.

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**  
**PRACTICA #5**

**Nombre:** Contador de pulsaciones de 0-999.

**Objetivo:** Desarrollar un programa para el microcontrolador 16F84A de tal forma que contabilice el número de pulsaciones detectadas en RA0, que el resultado (código binario) lo convierta a código de 7 segmentos y lo envíe por el Puerto B en forma multicanalizada por división de tiempo, cuando llegue a 1000 reinicie la cuenta en cero.

**Material:**

- 1 Microcontrolador 16F84
- 1 XTAL de 4MHz
- 1 R de 10K $\Omega$
- 10 R de 470 $\Omega$  o 330 $\Omega$
- 2 C de 22pF
- 1 C de 100 pF
- 2 push button
- 3 Display's de 7 segmentos (ánodo común o cátodo común)
- 3 Transistores npn o pnp
- 1 dip switch de 8 líneas
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores
- 1 software MPLAB

**Diagrama a bloques**

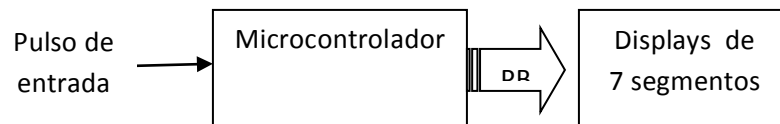


Figura 1) Diagrama a bloques del sistema mínimo para contabilizar pulsaciones de 0-999.

**PROCEDIMIENTO**

- 1.- Interconectar el sistema mínimo basado en microcontrolador 16F84A.
- 2.- Interconectar directamente al puerto B TRES displays de 7 segmentos, de tal forma que en el pin RB0 esté conectada la terminal **a** del display, y en el pin RB6 esté conectada la terminal **g** de los 3 displays.
- 3.- Utilice los bits RA1, RA2 y RA3 para activar cada transistor de selección de display.
- 3.- Escriba el programa para que detecte los pulsos en RA0, una vez detectados incremente a un contador y posteriormente convierta el número binario del contador a código de 7 segmentos.
- 4.- El código 7 segmentos deberá ser enviado por el puerto B en forma multicanalizada por división de tiempo, es decir por un periodo de tiempo se despliegan las unidades, en otro periodo de tiempo se despliegan las decenas y en otro periodo de tiempo las centenas. (Este retardo puede ser del orden de 5 a 15 mS).

Elaboró: M.C. Everardo Inzunza González

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**  
**PRACTICA #6**

**Nombre: ALARMA DIGITAL**

**Objetivo:** Diseñar y construir una alarma digital utilizando el microcontrolador 16F84A.

**Requerimientos generales del sistema**

El sistema tendrá como entrada 8 sensores del tipo ON/OFF y una entrada para ACTIVAR/DESACTIVAR la alarma. Estas 8 entradas serán a través del puerto A (RA0:RA3) y puerto B (RB3:RB6), pueden ser activadas en ALTO/BAJO, esto depende del tipo de sensor. Cuando la alarma esté activada y cualquiera de los sensores se active, la alarma deberá encender los bits 0, 1 y 2 del puerto B, los cuales encenderán una bocina (buzzer) (de 12 Vcd o 24Vcd o 17vca), una luz de emergencia (12vcd o 127 vca) y un indicador led. Cuando la alarma este desactivada, deberá ignorar todas las señales provenientes de los sensores.

**Material:**

- 1 Microcontrolador 16F84A
- 1 Luz de emergencia (torreta)
- 1 Bocina de alarma (Buzzer) (12V)
- 8 sensores para alarma.
- Dispositivos de potencia (Relays, optoacopladores, TRIAC's, transistores de potencia)
- 1 XTAL de 4MHz
- 1 R de 10KΩ
- 1 LED
- 2 C de 22pF
- 1 C de 100 pF
- 2 push button
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores
- 1 software MPLAB

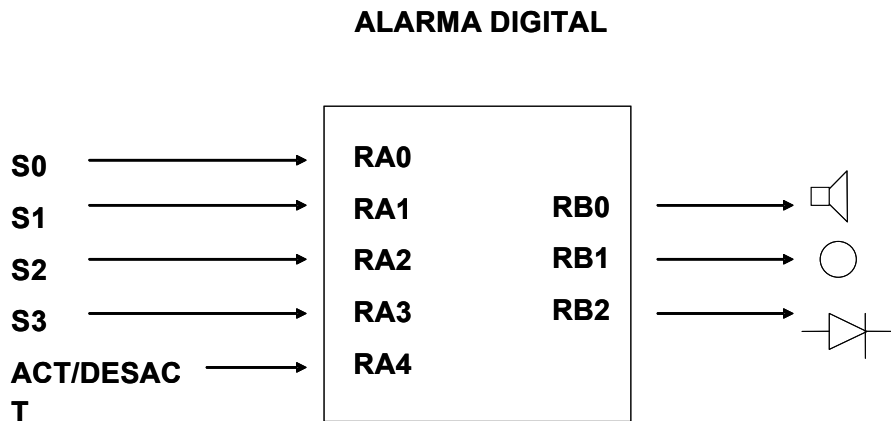


Figura 1) Diagrama a bloques de la alarma digital.

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**  
**PRACTICA #7**

**Nombre:** Generación de retardos y secuencias con el microcontrolador PIC 16F84A.

**Objetivo:** Desarrolle un programa para que encienda 8 motores de CD en forma secuencial con un retardo entre secuencias de medio segundo.

**Requerimientos:**

El sistema deberá tener un botón de **arranque** para que inicie el encendido secuencial y un botón de **paro** secuencial en orden inverso a la secuencia de arranque. Además considere el uso de un botón de **paro de Emergencia** para apagar inmediatamente (al mismo tiempo) todos los motores. Considere que el botón de arranque secuencial está conectado en el pin RA0 y el de Paro secuencial está en el pin RA1. El paro de emergencia se encuentra conectado al pin RA2. Los motores deberán conectarse en el puerto B por medio de una etapa de potencia. El tiempo inter-secuencia es de 500 mS.

**Material:**

- 1 Microcontrolador 16F84
- 1 XTAL de 4MHz
- 1 R de 10K $\Omega$
- 10 R de 470 $\Omega$  o 330 $\Omega$
- 2 C de 22pF
- 1 C de 100 pF
- 2 push button
- 8 Led's
- 1 dip switch de 8 líneas
- 1 Fuente de voltaje (potencia) con capacidad para alimentar a los 8 motores de CD
- 1 Programador de microcontroladores
- 1 software MPLAB
- 8 motores de CD de 12V o superior
- 8 relays u 8 transistores de potencia

**PROCEDIMIENTO:**

Diseñe y construya el circuito basado en microcontrolador PIC 16F84A, interconecte los botones de arranque y paro, diseñe la etapa de potencia para interconectar cada motor de CD al microcontrolador, elabore el programa en lenguaje ensamblador para generar los retardos y la lógica de operación de arranque/paro secuencial y paro de emergencia, simule el prototipo en MPLAB y PROTEUS, implemente físicamente el circuito, realice pruebas experimentales.

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**  
**PRACTICA #8**

**Nombre: SEMÁFORO DIGITAL**

**Objetivo:** Desarrollar un sistema basado en un microcontrolador PIC 16F84A para controlar un semáforo de cruce de una calle principal con una avenida.

**Requerimientos generales del sistema**

Se requiere diseñar un semáforo digital para controlar el tráfico vehicular en la intersección de una calle de tráfico muy denso con una calle de tráfico moderado.

La calle principal deberá tener una luz verde encendida durante un tiempo de 30 segundos. Antes de agotarse el tiempo de 30 segundos, la luz verde deberá parpadear 3 veces y posteriormente apagarse. La calle lateral debe tener la luz verde encendida durante 20 segundos, de igual manera antes de agotarse los 20 segundos la luz verde deberá parpadear 3 veces. La luz ámbar de precaución de ambos semáforos tiene que durar 5 segundos y debe ser activada cuando se apaga la luz verde de cada semáforo. La luz roja de ambos semáforos es encendida una vez agotado el tiempo de la luz ámbar de cada semáforo. Considere el uso de un botón de cruce peatonal, cuando este sea activado el sistema deberá esperar que se agote el tiempo de la luz verde (de los carros calle lateral o principal), así como el de la luz ámbar y posteriormente activar la luz verde de cruce peatonal. La luz roja de cruce peatonal se mantiene encendida mientras están activadas las luces verdes y/o ambar del cruce de vehículos de la calle principal o lateral. Considere una entrada para activar al semáforo en modo nocturno, es decir, en este modo la luz ámbar de la calle principal debe estar intermitente, mientras que la luz roja de la avenida también estará estado intermitente.

Favor de construir una base para cada semáforo.

**Material:**

- 1 Microcontrolador 16F84
- 1 Reelevadores u optoacopladores con TRIAC's
- 3 focos verdes, 2 focos ámbar y 3 focos rojos de 127 Vca @ 60Watts.
- 1 XTAL de 4MHz
- 1 R de 10K $\Omega$
- 6 R de 470 $\Omega$  o 330 $\Omega$
- 2 C de 22pF
- 1 C de 100 pF
- 2 push button
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores
- 1 software MPLAB

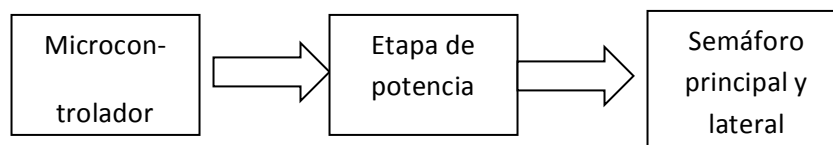


Figura 1) Diagrama a bloques del semáforo digital.

Elaboró: M.C. Everardo Inzunza González



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO  
MICROCONTROLADORES  
PRACTICA #9**

**Nombre: TERMÓMETRO DIGITAL BASADO EN EL CONVERTIDOR DE A/D DEL MICROCONTROLADOR 16F87x.**

**Objetivo:** Configurar e implementar el convertidor analógico a digital que se encuentra internamente en el microcontrolador PIC 16F87X para realizar la medición de temperatura y su correspondiente despliegue de la información en un display de cristal liquido (LCD) 16 X 2.

**Material:**

- 1 Microcontrolador 16F877
- 1 Pantalla LCD 16 X 2
- 1 Sensor de temperatura LM 35
- Amplificadores operacionales
- 1 potenciómetro de precisión de 5 KOhms o 10 KOhms.
- 1 XTAL de 4MHz
- 1 R de 10KΩ
- 10 LED's
- 10 R de 330 Ohms
- 2 C de 22pF
- 1 C de 100 pF
- 1 push button
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores
- 1 software MPLAB

**PROCEDIMIENTO**

**Requerimientos del sistema para la fase #1 de esta práctica (1ra semana).**

El circuito tendrá las siguientes características: entrada por el canal AN0, se utilizará como voltaje de referencia el interno (Vdd), resolución de 10 bits, frecuencia de muestreo  $f_{osc}/32$ , los datos justificados a derecha para que sean enviados a través de los puertos B y C.

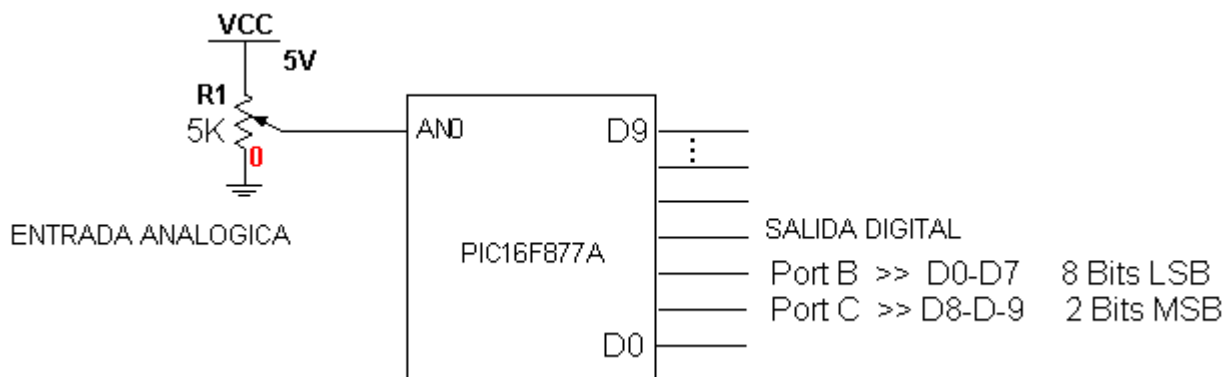


Figura 1) Diagrama a bloques del convertidor analógico a digital basado en el microcontrolador 16F877A.

## Requerimientos del sistema para la fase #2 de esta práctica (2da semana).

- Continúe con el circuito de la práctica del ADC con microcontrolador (fase #1), modifique el código fuente, para que el microcontrolador realice la conversión de código binario (8 bits MSB) a código ASCII.
- Posteriormente implemente la subrutina de despliegue para el LCD.
- Interconecte el sensor y el acondicionamiento de señal al microcontrolador.
- Interconecte el LCD.
- Realice las simulaciones correspondientes en MPLAB y PROTEUS.
- Realice las pruebas experimentales y haga una calibración del instrumento (termómetro) para que la medición sea más exacta. Para hacer esta calibración, se requiere hacer la comparación de la temperatura medida con algún termómetro digital comercial o de mercurio.

### Características generales del termómetro:

El termómetro deberá medir la temperatura desde 0°C hasta 150°. Deberá desplegar en un display de cristal líquido (LCD) la siguiente información: UABC-INGENIERIA, La temp= xxx °C.

### Material:

1

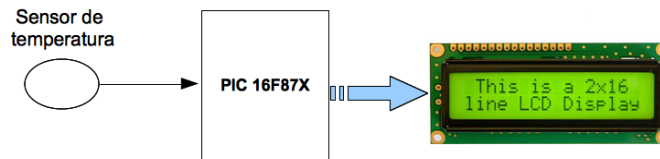


Figura 2. Esquema a bloques del termómetro digital basado en microcontrolador pic 16F87X.

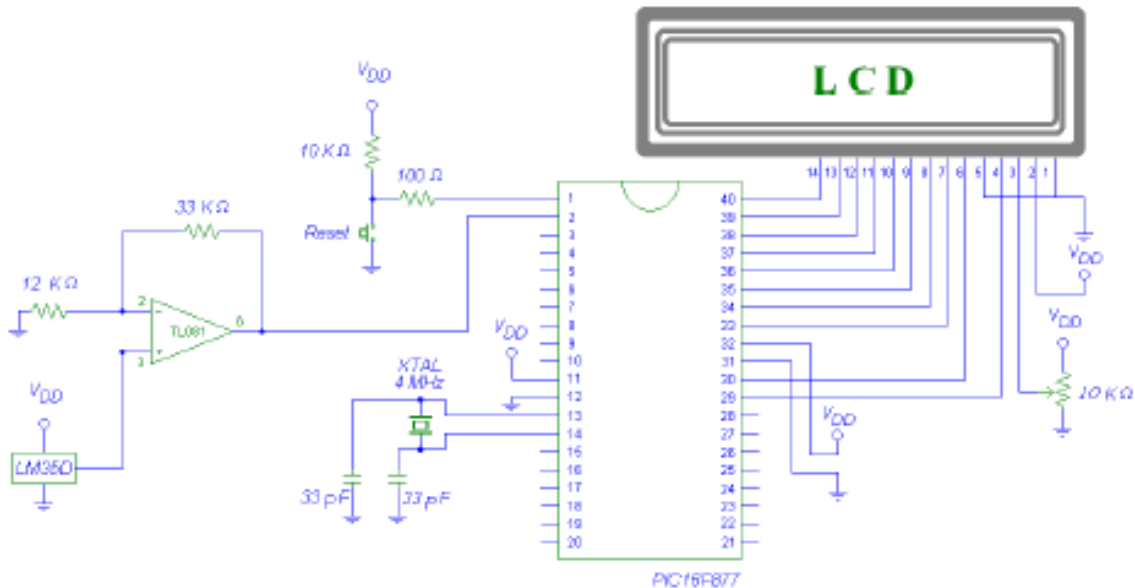


Figura 3. Circuito eléctrico detallado del termómetro digital basado en microcontrolador pic 16F87X. Nota: En este caso se está empleando en modo de 8 bits al LCD.

## PROTOTIPO FINAL

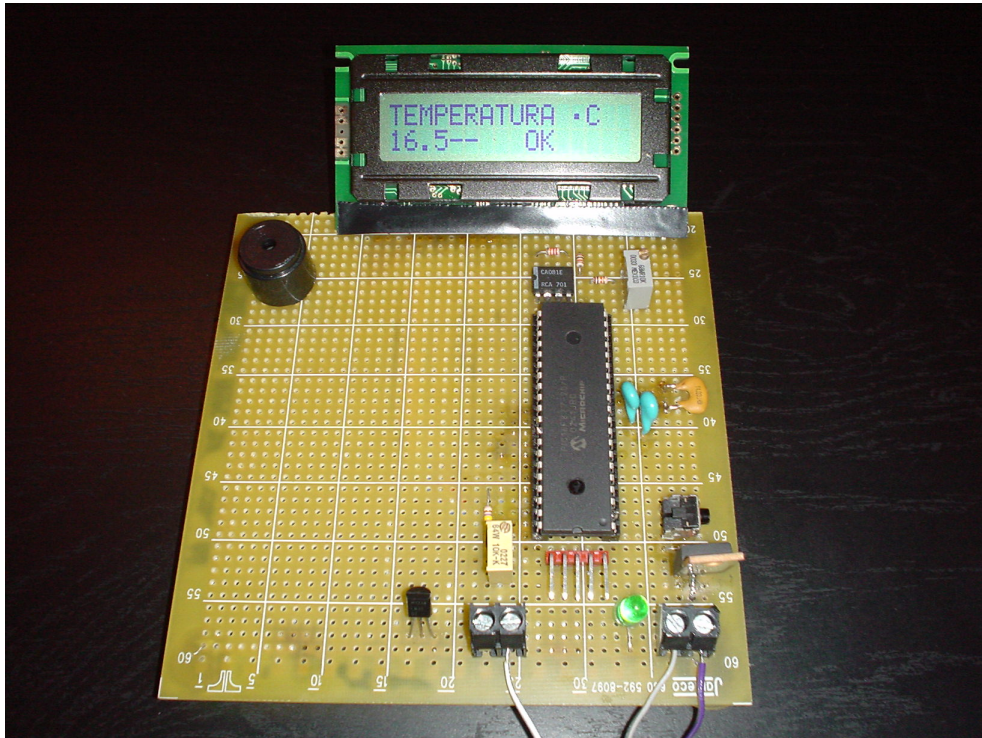


Figura 4. Ejemplo de un prototipo final de termómetro digital basado en microcontrolador. Observe que en la practica a realizar, es distinto el mensaje de texto.

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**  
**MICROCONTROLADORES**  
**PRACTICA #10**

**Nombre:** CONTROL DE UNA BANDA TRANSPORTADORA

**Objetivo:** Desarrollar un programa para el microcontrolador 16F84A de tal forma que controle una Banda Transportadora de recipientes, para que sean llenados automáticamente con agua. El sistema deberá tener un botón de arranque, para que encienda al motor y la banda comience a girar, cuando se detecte que llegue el primer recipiente, la banda deberá detenerse y se encenderá una bomba hidráulica para que comience a llenar con agua el recipiente. Una vez llenado este recipiente, se apaga la bomba y se enciende el motor para que gire la banda y acerque el siguiente recipiente a ser llenado. El sistema deberá llevar el conteo de cuantos recipientes se han llenado en una jornada de trabajo, el despliegue consiste en un LCD. En caso de algún accidente, considerar un botón de paro de emergencia.

**Material:**

- 1 Microcontrolador 16F84
- 1 XTAL de 4MHz
- 1 R de 10K $\Omega$
- 2 C de 22pF
- 1 C de 100 pF
- 2 push button
- 3 Display's de 7 segmentos (ánodo común o cátodo común) o un LCD 2X16
- 1 motor de CD
- 1 Bomba Hidráulica (puede ser de pecera ó wiper's)
- 2 relays
- 2 transistores de potencia
- 1 Fuente de voltaje para el motor y la bomba
- 1 Láser, LED infrarrojo y un foto detector.
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores
- 1 software MPLAB

**Diagrama a bloques**

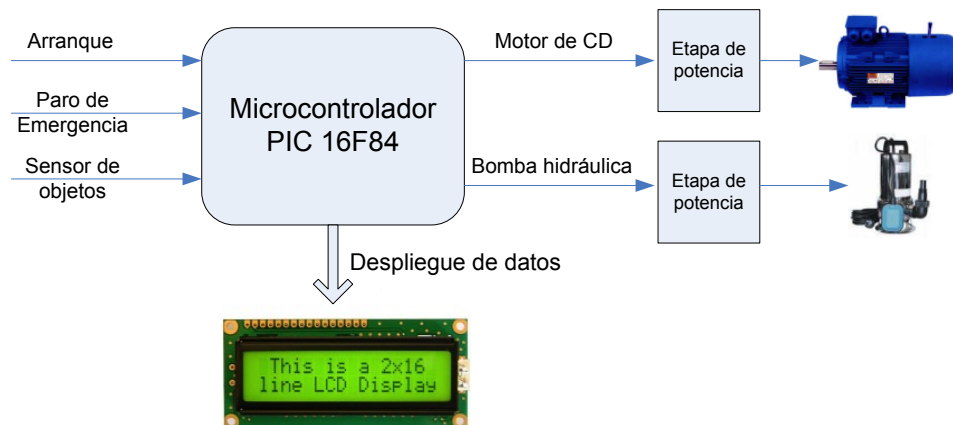


Figura 1) Diagrama a bloques del sistema automático de llenado de recipientes.

Elaboró: M.C. Everardo Inzunza González

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO  
MICROCONTROLADORES  
PRACTICA #11**

**Nombre: COMUNICACIÓN SERIAL CON EL MICROCONTROLADOR**

**Objetivo:** Diseñe y construya un termómetro digital basado en microcontrolador de la familia 16F87 con comunicación RS-232 hacia una computadora personal.

**Requerimientos generales del sistema**

El termómetro de deberá medir la temperatura desde 0°C hasta 150°. Deberá desplegar en un display de cristal líquido (LCD) la siguiente información: UABC-INGENIERIA, La temp= xxx °C. El dato binario deberá ser enviado a una PC por medio del puerto USART del microcontrolador hacia una PC. La PC deberá desplegar el contenido del dato en código binario, sistema decimal y en grafica de barra de la temperatura.

**Material:**

- 1 Microcontrolador 16F877
- 1 IC Max232.
- 1 XTAL de 4MHz
- 1 R de 10KΩ
- 10 LED's
- 10 R de 330 Ohms
- 2 C de 22pF
- 1 C de 100 pF
- 1 push button
- 1 Cable serial null modem con conector DB-9 (RS-232)
- 1 Fuente de voltaje 5 Volts
- 1 Sensor de temperatura
- 1 op-amp
- 1 LCD de 2X16.
- 1 Programador de microcontroladores
- 1 software MPLAB

**INTRODUCCIÓN:**

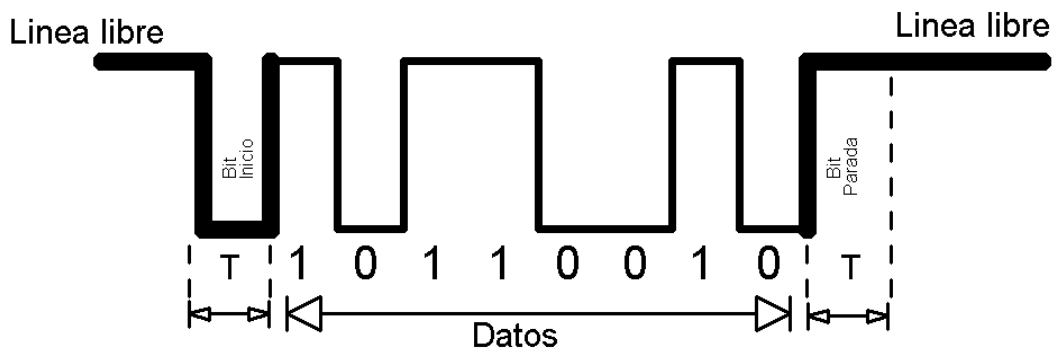
Cuando se ha logrado implementar cualquier protocolo de comunicación entre una PC y un microcontrolador se puede decir que una gran cantidad de aplicaciones están a la puerta, entre las cuales se pueden nombrar monitoreo, control digital, impresión de datos en papel, impresión en pantalla de datos inmediatos, etc. Los protocolos más comúnmente implementados en microcontroladores y PC son: RS232, IP, Paralelo, etc.

El microcontrolador PIC16F877A tiene incluido dentro de su estructura interna un modulo USART (Transmisor receptor universal sincrono asíncrono) que es la abreviación de “universal synchronous asynchronous receiver transmitter” en inglés también es conocido como SCI (“serial communications interface”) y es uno de los mas comúnmente usados para la implementación de puertos seriales, la configuración asíncrona full dúplex es una de las mas populares de la USART pues es usada para interfaces con el puerto serie de las computadoras personales o PC usando el protocolo RS-232.

La función principal del modulo USART es enviar y recibir datos de forma serial, esta operación puede ser dividida en dos categorías transmisión síncrona y transmisión asíncrona. La operación síncrona usa una línea de datos y una de reloj mientras que la operación asíncrona solo usa la línea de datos. Esta es la principal diferencia entre estos dos modos de operación.

**Modo Asíncrono:**

La comunicación asíncrona es la que generalmente se utiliza para la comunicación entre un microcontrolador y una PC. Para la comunicación serie con la PC se utiliza el estándar “no retorno a cero” (NRZ) en la forma conocida como 8-N-1 que quiere decir 8 bits de datos, sin paridad y un bit de parada tal como se muestra en la figura 1.



*Figura 1. Cronograma del protocolo RS-232.*

Línea libre se define como un estado lógico alto o uno. El inicio de la transmisión de datos (Bit de inicio) se define como un estado lógico bajo o cero. Los bits de datos son enviados enseguida del Bit

de inicio, enviando en primer lugar el Bit menos significativo. Después de los Bits de datos se envía el Bit de parada que es un estado lógico alto o uno. El periodo T depende de la velocidad de transmisión y es ajustado de acuerdo a las necesidades de transmisión. Para la velocidad de 9600 baudios, T es de 104 uS. En la figura 2 se muestra un conector RS-232 y en la tabla 1 muestra los nombres de cada pin.

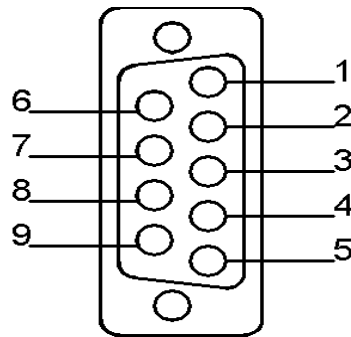


Figura 2. Conector DB9 estándar.

Tabla 3. Nombres de las patas del conector DB9.

Pin	Nombre	Descripción
1	CD	Acarreo detectado
2	RXD	Recepción de datos
3	TXD	Transmisión de datos
4	DTR	Terminal de datos lista
5	GND	Tierra
6	DSR	Grupo de datos listo
7	RTS	Petición de envío
8	CTS	Libre para enviar
9	RI	Indicador de Ring

Los pasos para la configuración del modulo USART como un puerto asíncrono serie, con la configuración no retorno a cero en la forma 8-N-1 son los siguientes:

**Para la transmisión de datos**

1. Inicializar el Registro generador de Baud Rate (SPBRG) con el valor apropiado si se piensa usar una velocidad alta el Bit BGRH debe ser puesto a uno.
2. Se habilita el puerto serie asíncrono limpiado o poniendo a cero el Bit (SYNC ) y se pone a uno el Bit SPEN (Serial port enable)
3. Cargar el dato en el registro TXREG (después de esto se inicializa la transmisión)
4. Habilitar la transmisión poniendo a uno el Bit TXE

**Para la recepción de datos**

1. Inicializar el Registro generador de Baud Rate (SPBRG) con el valor apropiado si se piensa usar una velocidad alta el bit BGRH debe ser puesto a uno.
2. Se habilita el puerto asíncrono serie limpiado o poniendo a cero el Bit (SYNC) y se pone a uno el bit SPEN (Serial port enable)
3. Habilitar la transmisión poniendo a uno el Bit CREN
4. Leer el registro RCREG para conocer los datos recibidos

Para conocer el valor que se requiere introducir en el registro SPBRG para una determinada velocidad de transferencia asíncrona se usan la siguiente fórmula:

$$SPBRG = \frac{frec\_clock\_entrada}{64 * Velocidad\_deseada} - 1$$

El valor obtenido por esta fórmula se redondea al entero inferior más cercano y se introduce en el registro SPBRG para fijar la velocidad de transferencia deseada. En la tabla 4 se muestran algunos valores para SPBRG para diferentes velocidades y un cristal externo de 4 MHz



Tabla 2. Valores para fijar diferentes velocidades de transferencia.

Baud Rate (K)	Kbaud	%Error	Valor de SPBRG (Decimal)
0.3	0.300	0	207
1.2	1.202	0.17	51
2.4	2.404	0.17	25
9.6	8.929	6.99	6
19.2	20.833	8.51	2
28.8	31.250	8.51	1
33.6	-	-	-
57.6	62.500	8.51	0

La interfase física que se utiliza comúnmente para conectar un microcontrolador con la PC se muestra en la figura 6. El circuito MAX232 realiza la tarea de generar de acuerdo a los valores lógicos entregados por el microcontrolador los niveles necesarios de voltaje para la comunicación RS-232.

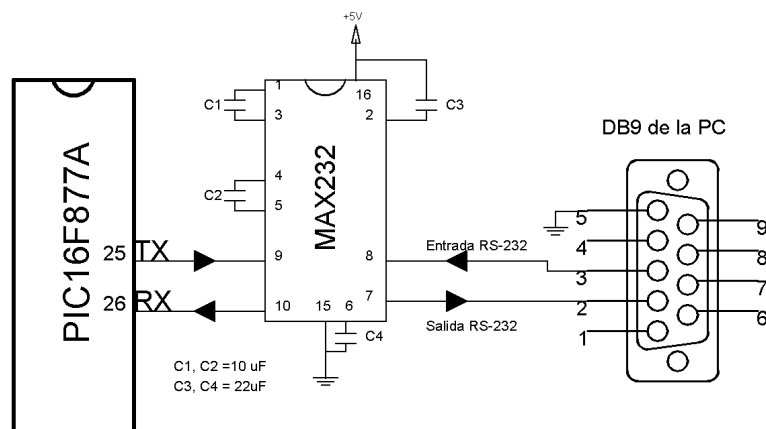


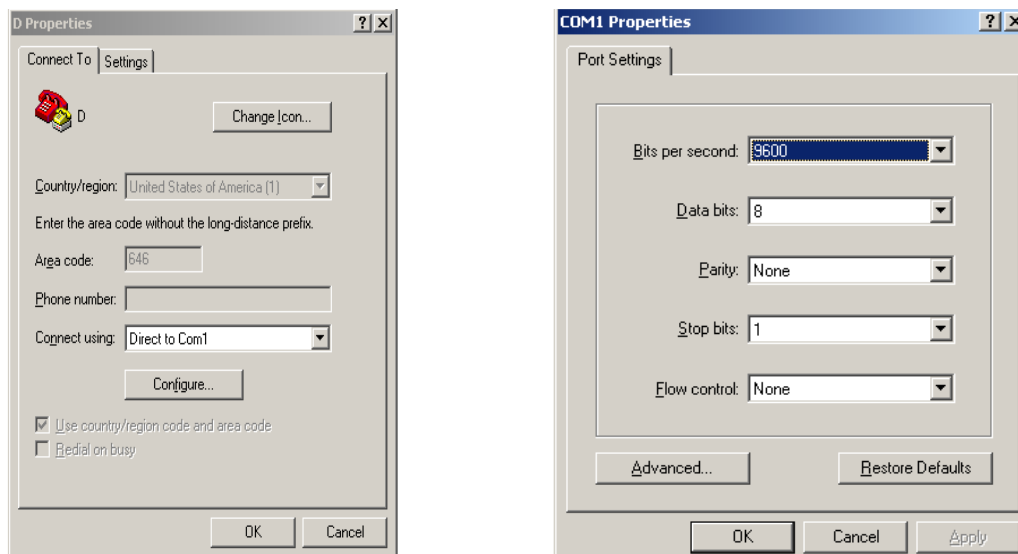
Figura 3. Conexión física para la comunicación RS-232.

## **DESARROLLO DE LA COMUNICACIÓN RS-232:**

Para implementar la comunicación serie entre una PC y un microcontrolador en la etapa experimental se suele utilizar el programa HyperTerminal figura 4.

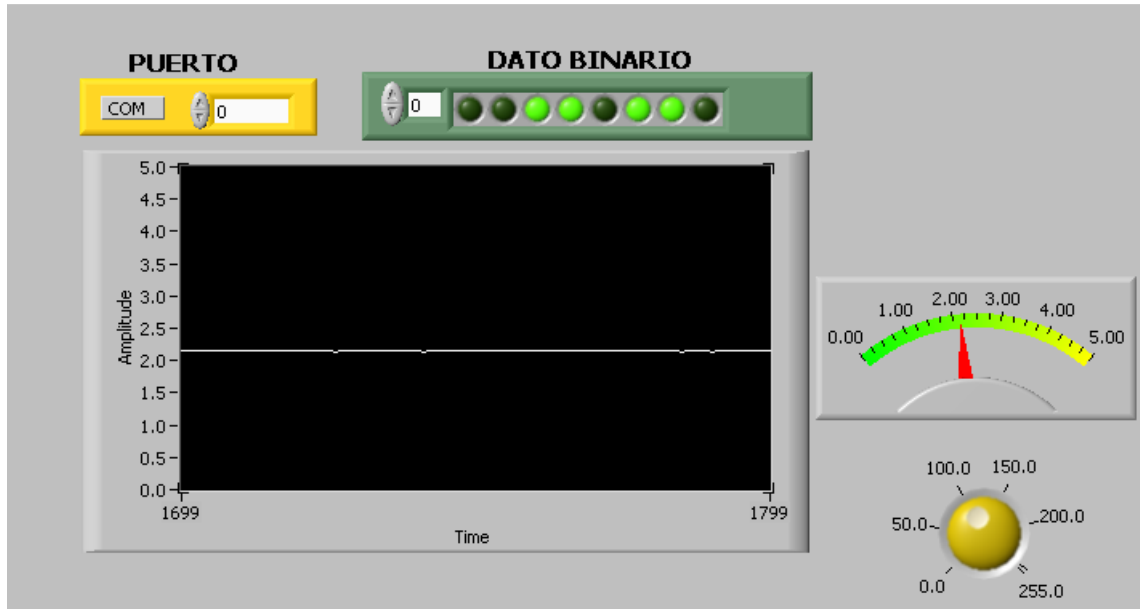


*Figura 4. Pantalla de inicio del Programa HyperTerminal.*



*Figura 5. Configuración y elección del puerto disponible a utilizar.*

Una vez que se ha establecido la comunicación entre HyperTerminal y el microcontrolador se puede pasar a programar la aplicación que se desee en el paquete que se desee MATLAB, LABVIEW, VISUAL BASIC, etc. En la presente práctica se optó por LABVIEW, del cual se presenta pantalla principal en la figura 6.



### PROCEDIMIENTO

- a) Continúe con el circuito de la práctica del ADC con microcontrolador.
- b) Implemente la conversión de código binario a código ASCII con el ensamblador y la subrutina de despliegue para el LCD.
- c) Interconecte el sensor y el acondicionamiento de señal al microcontrolador.
- d) Interconecte el LCD.
- e) Escriba la subrutina de código para transmitir los datos por el puerto USART del microcontrolador.
- f) Interconecte el circuito MAXIM 232 al microcontrolador para convertir los niveles de voltaje TTL a RS-232.
- g) Realice el programa de la PC para la captura y graficación de datos en LABVIEW.
- h) Realice las pruebas y haga una calibración del instrumento para que la medición sea más exacta.

**REFERENCIAS:**

- [1] Anónimo, "*PICmicro Mid-Range MCU Family Reference Manual*", Microchip rev/DS33023A, December 1997.
- [2]. Anónimo, "*PIC16F87X Data sheet*" Microchip rev/30292C, 2001.
- [3]. Anónimo, "*MAX232 Data sheet*", Texas Instruments, 2003.