

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



MANUAL DE PRÁCTICAS

UNIDAD DE APRENDIZAJE: 11686 MICROCONTROLADORES

PROFESOR:

DR. EVERARDO INZUNZA GONZÁLEZ

Ensenada, Baja California, Febrero del 2016

FORMATO DEL REPORTE DE PRÁCTICAS DEL LABORATORIO

PORTADA

Nombre de la universidad

Nombre de la Facultad

Carrera

Materia

Alumno

Nombre y número de Práctica

Nombre del maestro

Lugar y fecha

CONTENIDO

Objetivo

Lista de material y equipo

Antecedentes relacionados al tema de la práctica

Desarrollo experimental

 Diagrama a bloques

 Descripción detallada del procedimiento

 Fotografías de los prototipos y experimentos realizados

Análisis y discusión de resultados

 Cálculos

 Tablas de mediciones

 Gráficas de resultados

Código fuente de los programas realizados

Simulaciones (PROTEUS, MPLAB, etc.)

Screen o pantallazos de los resultados visualizados en el monitor y del trabajo realizado en la práctica (de los comandos o software utilizado)

Conclusiones individuales

Bibliografía

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 Ω
- 10 R de 470 Ω o 330 Ω
- 2 C de 22pF
- 1 C de 100 pF
- 1 push button
- 8 Led's (Puede ser barra de 8 o 10 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.

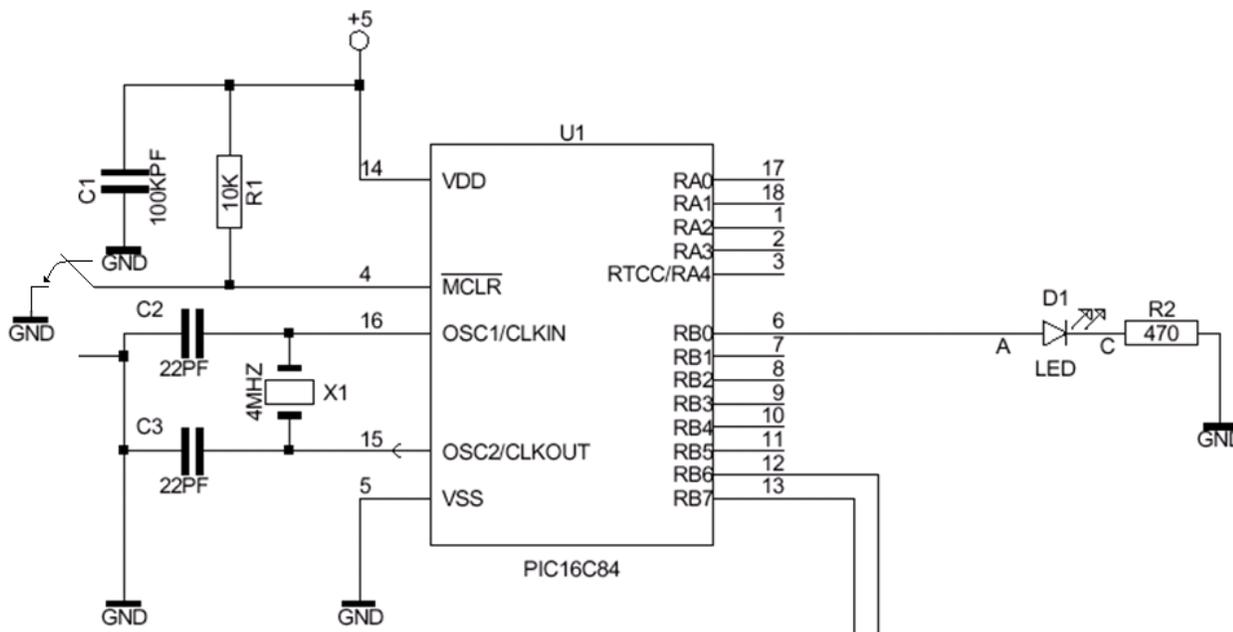


Figura 1) Diagrama eléctrico del sistema mínimo basado en microcontrolador 16F84A.

Elaboró: Dr. Everardo Inzunza González

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA ENSENADA
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:

$$PB=(6*PA-3)/2$$

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

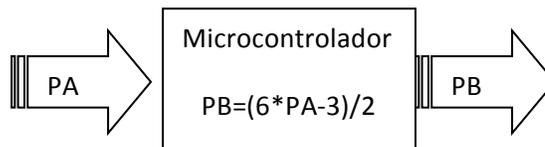


Figura 1) Diagrama a bloques del sistema mínimo para realizar la operación aritmética $PB=(3PA-1)/2$.

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 AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO
MICROCONTROLADORES
PRACTICA #3

Nombre: Contador de pulsaciones detectadas en RA0

Objetivo: Desarrollar un programa para el microcontrolador PIC 16F84A de tal forma que contabilice la cantidad de pulsaciones detectadas en el pin RA0, cuando llegue a 16₁₀, deberá re-iniciar la cuenta en cero. El resultado del conteo lo deberá desplegar en el puerto B y el resultado guardarlo en la memoria RAM.

Material:

- 1 microcontrolador 16F84
- 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

PROCEDIMIENTO

- 1) Armar el siguiente circuito eléctrico basado en microcontrolador 16F84A, además conectar un push button en el pin RA0 con sus respectiva resistencia y conectar los 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.

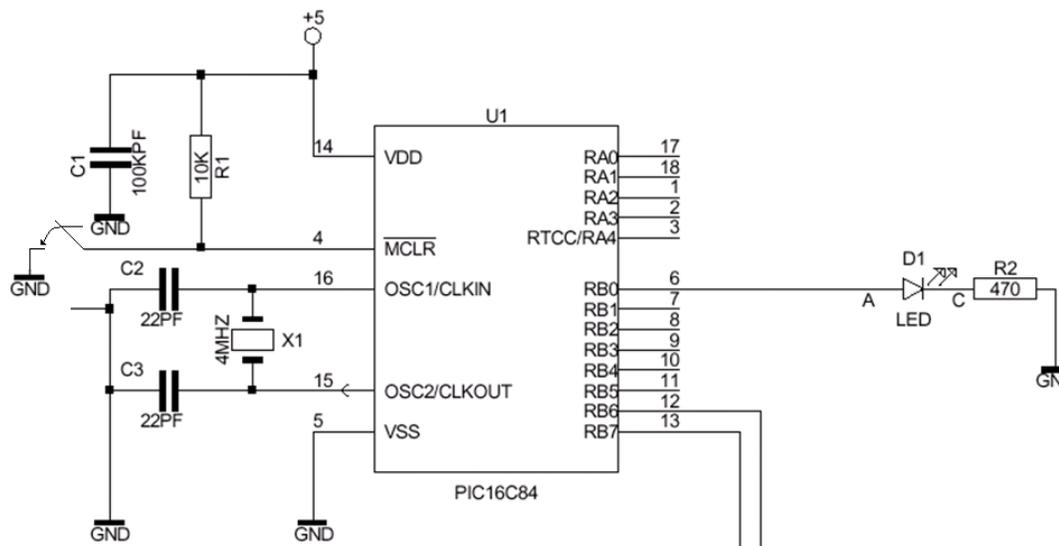


Figura 1) Diagrama eléctrico del sistema mínimo basado en microcontrolador 16F84A.

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO
MICROCONTROLADORES
PRACTICA #4

Nombre: ALARMA DIGITAL

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

Requerimientos generales del sistema

El sistema tendrá como entrada 4 sensores del tipo ON/OFF y una entrada para ACTIVAR/DESACTIVAR (RA4) la alarma. Estas 4 entradas serán a través del puerto A (RA0:RA3) 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

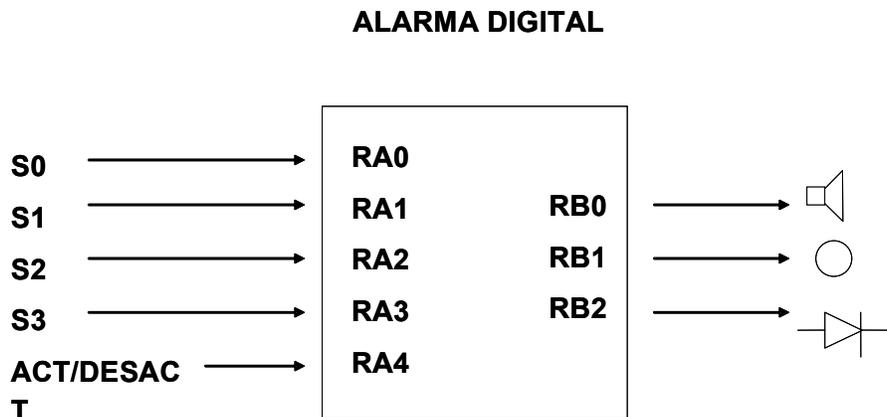


Figura 1) Diagrama a bloques de la alarma digital.

Elaboró: Dr. Everardo Inzunza González

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

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

Objetivo: Desarrolle un programa para que encienda 6 motores de CD en forma secuencial con un retardo entre secuencias de un 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 (Transistor de potencia TIP 41A). El tiempo inter-secuencia es de 1000 mS.

Material:

- 1 Microcontrolador 16F84
- 1 XTAL de 4MHz
- 1 R de 10K Ω
- 10 R de 470 Ω o 330 Ω
- 2 C de 22pF
- 1 C de 100 pF
- 2 push button
- 6 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
- 6 motores de CD de 5V o superior
- 6 transistores de potencia TIP 41A

PROCEDIMIENTO:

Diseñe y construya el circuito basado en microcontrolador PIC 16F84A, interconecte los botones de arranque y paro secuencial, 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 #6

Nombre: Contador de pulsaciones de 0-9999 empleando un microcontrolador PIC 16F84A y un LCD.

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 ASCII y lo envíe por el Puerto B a un display de cristal líquido, cuando llegue a 1000 reinicie la cuenta en cero.

Requerimientos: En el primer renglón de la LCD deberá aparecer el mensaje: **CONTADOR-UABC**. En el segundo renglón deberá aparecer: **conteo =XXXX**, donde XXXX es el valor numérico de la variable contador, la cual deberá comenzar en 0.

Material:

- 1 Microcontrolador 16F84
- 1 XTAL de 4MHz
- 1 R de 10K Ω
- 10 R de 470 Ω o 330 Ω
- 2 C de 22pF
- 1 C de 100 pF
- 1 push button
- 1 LCD 16X2
- 1 Fuente de voltaje 5 Volts
- 1 Programador de microcontroladores
- 1 software MPLAB

Diagrama a bloques

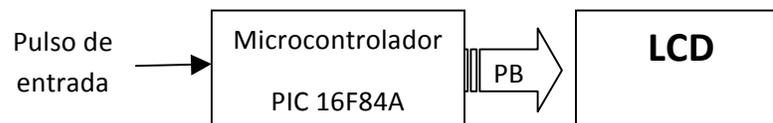


Figura 1) Diagrama a bloques del sistema mínimo para contabilizar pulsaciones de 0-9999.

PROCEDEMIENTO

- 1.- Interconectar el sistema mínimo basado en microcontrolador 16F84A.
- 2.- Interconectar directamente al puerto B un LCD en modo de comunicación de 4 Bits.
- 3.- Utilice el bit RA0 para activar cada conteo..
- 3.- Escriba el programa en ASM para que detecte los pulsos en RA0, una vez detectados incremente a un contador y posteriormente convierta el número binario (o de base 10) a código ASCII y lo envíe al LCD.

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

Nombre: CONTROL DIGITAL ON/OFF DE TEMPERATURA EMPLEANDO UN MICROCONTROLADOR PIC 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 líquido (LCD) 16 X 2.

Descripción:

Desarrolle un programa en lenguaje C, para que el microcontrolador PIC esté midiendo periódicamente la temperatura ambiente empleando el sensor LM 35 o equivalente. Cuando la temperatura esté por debajo del umbral inferior (16°C), el microcontrolador deberá accionar un calentador de 127Vca e indicar en el LCD que está activado el Calentador y además desplegar el valor de la temperatura. Cuando la temperatura esté por arriba del umbral superior (30°C), el microcontrolador deberá accionar un ventilador de 127Vca e indicar esta acción en el LCD.

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

- a) Implemente la subrutina en lenguaje C de despliegue para el LCD.
- b) Interconecte el sensor y el acondicionamiento de señal al microcontrolador.

- c) Interconecte el LCD.
- d) Realice las simulaciones correspondientes en MPLAB y PROTEUS.
- e) Realice las pruebas experimentales y haga una calibración del instrumento (termómetro) para que la medición sea mas 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 de deberá medir la temperatura desde 0°C hasta 150°. Deberá desplegar en un display de cristal liquido (LCD) la siguiente información: UABC-INGENIERIA, La temp= xxx °C.

Material:

1

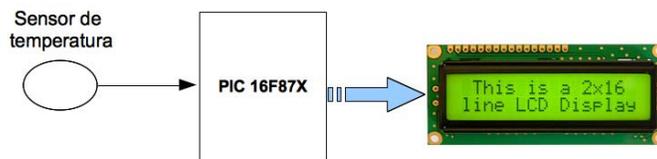


Figura 2. Esquema a bloques del termómetro digital basado en microcontrolador pic 16F87X.

Ejemplo de algunas simulaciones:

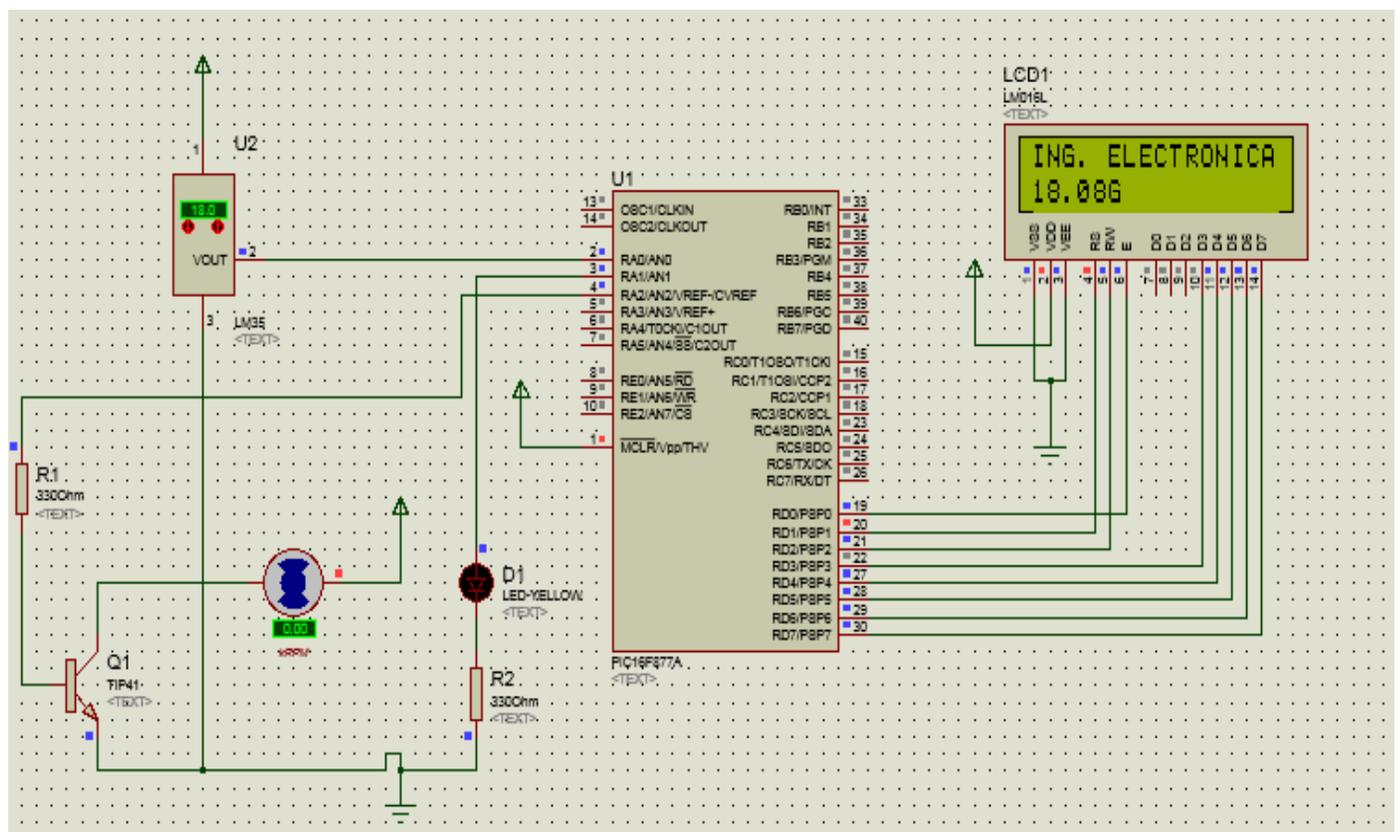


Figura 3. Ejemplo de simulación del circuito eléctrico detallado del termómetro digital basado en microcontrolador pic 16F87X. Nota: En este caso se esta empleando en modo de 4 bits al LCD.

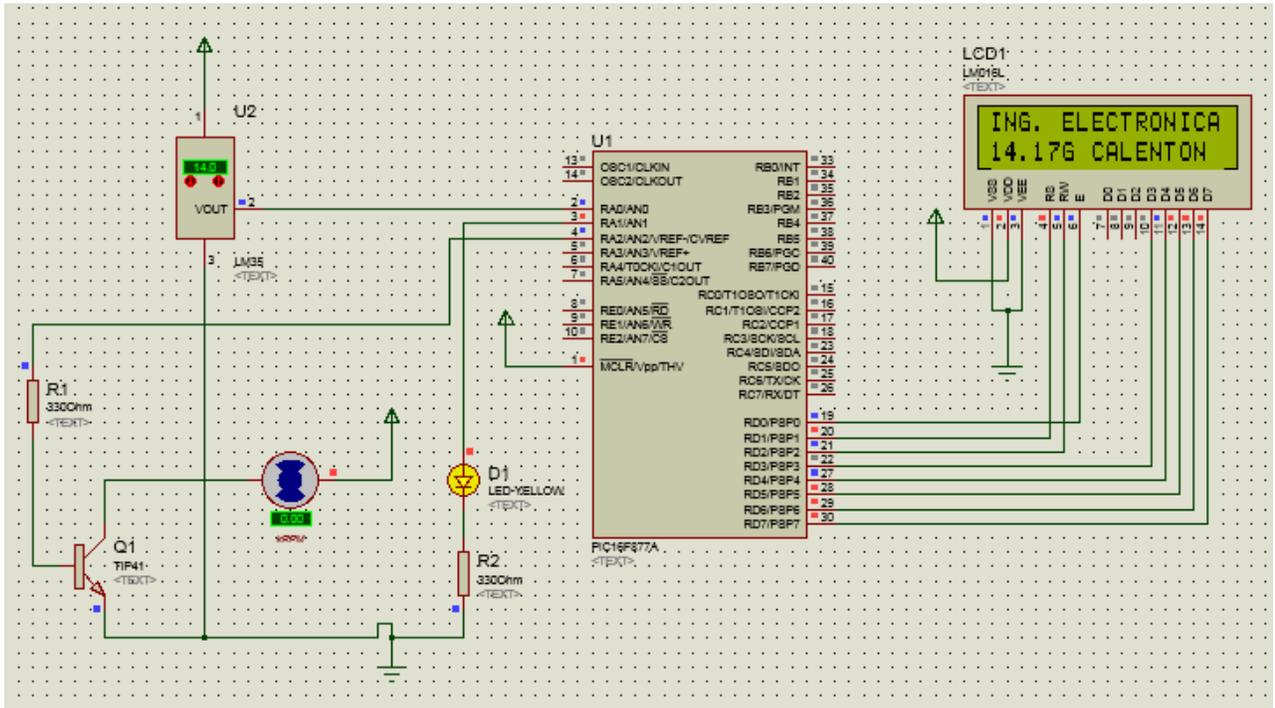


Figura 4. Ejemplo de simulación indicando que está activado el Calentón eléctrico.

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 Arduino 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 y un buzzer para que escuchen las personas invidentes. 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 ARDUINO UNO o NANO
- 7 Optoacopladores MOC 3011
- 7 TRIAC's 2N6344
- 3 focos verdes, 2 focos ámbar y 3 focos rojos de 127 Vca @ 60Watts.
- 1 R de 10K Ω
- 7 R de 470 Ω o 330 Ω
- 2 R 220 Ω
- 3 push button
- 1 Fuente de voltaje 5 Volts
- 1 software ARDUINO IDE

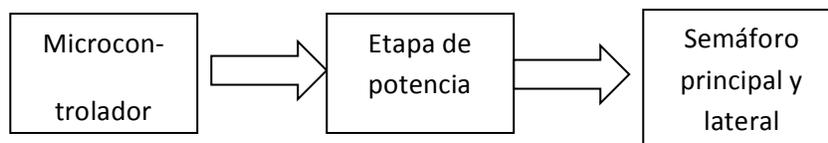


Figura 1) Diagrama a bloques del semáforo digital.

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

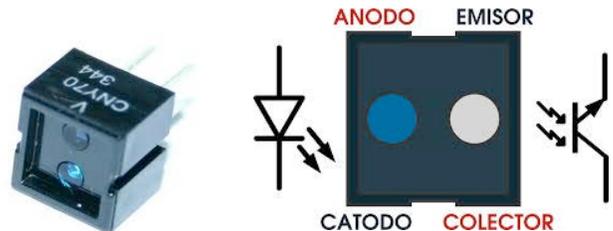
Nombre: Robot seguidor de línea basado en Arduino

Objetivo: Diseñar e implementar un robot seguidor de línea blanca basado en microcontrolador Arduino.

Material:

- 1 Tarjeta Arduino
- 1 Software arduino
- 3 Sensores CNY70
- 2 Servomotores de CD ó Moto-reductores
- 1 Chasis para robot seguidor o de carrito
- 1 Rueda loca
- 22 Llantas o ruedas de carrito
- 1 Fuente de voltaje
- 1 Batería de 9 Volts
- 2 potenciómetros
- Amplificadores operacionales, capacitores y resistencias
- CD4016B (Inversor con histéresis)
- LM 293D (opcional)
- 2 TIP 41C

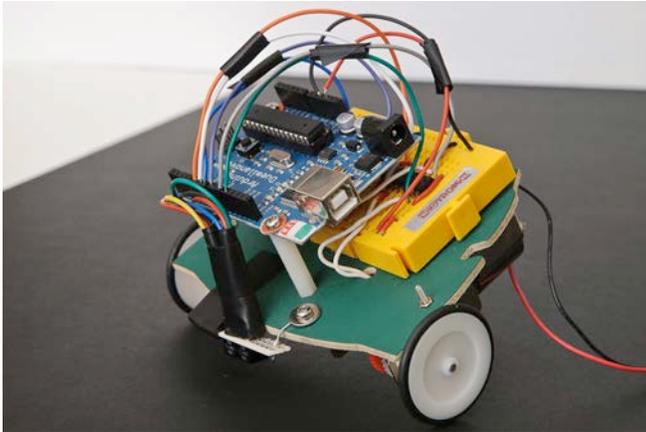
Sensor CNY70



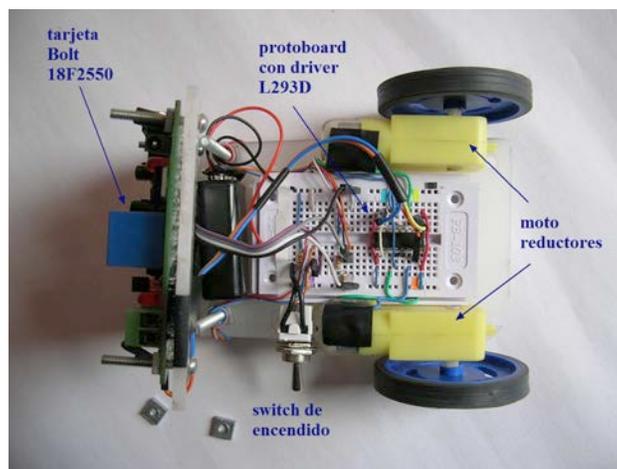
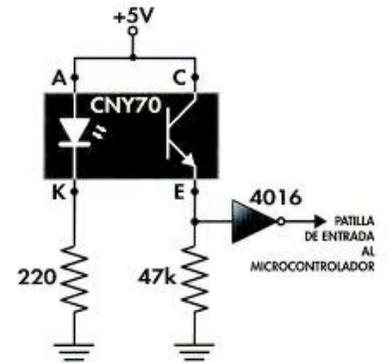
PROCEDIMIENTO EXPERIMENTAL

- 1) Diseñar el circuito de acondicionamiento de la señal de los sensores CNY70 y la etapa de comparación con OP-AMPS o empleando inversores lógicos con hysteresis (CD4016B)
- 2) Diseñar la etapa de potencia para el control de los motores de CD (Con transistors de potencia drivers como el LM293D, etc.
- 3) Diseñar y construir el chasis del robot considerando dos llantas que funcionen de forma autónoma con cada uno de los motores y que tenga una rueda loca (o con dirección).
- 4) Desarrollar el programa para Arduino con la lógica para que se siga una línea blanca con forma de elipse y del grosor de un tape (cinta aislar blanca) con fondo negro.
- 5) Probar el programa y circuito.

Ejemplo de seguidor de líneas



Circuito típico



Elaboró: Dr. Everardo Inzunza González

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA ENSENADA
MICROCONTROLADORES
PRACTICA #10

Nombre: Medición de temperatura e iluminación ambiental con microcontrolador Arduino.

Objetivo: Desarrollar un programa para el microcontrolador Arduino para medir la temperatura ambiental, e iluminación ambiental.

Requerimientos:

Las variables medidas deberán ser desplegadas en una LCD 16X2 y enviadas por puerto USB a una computadora.

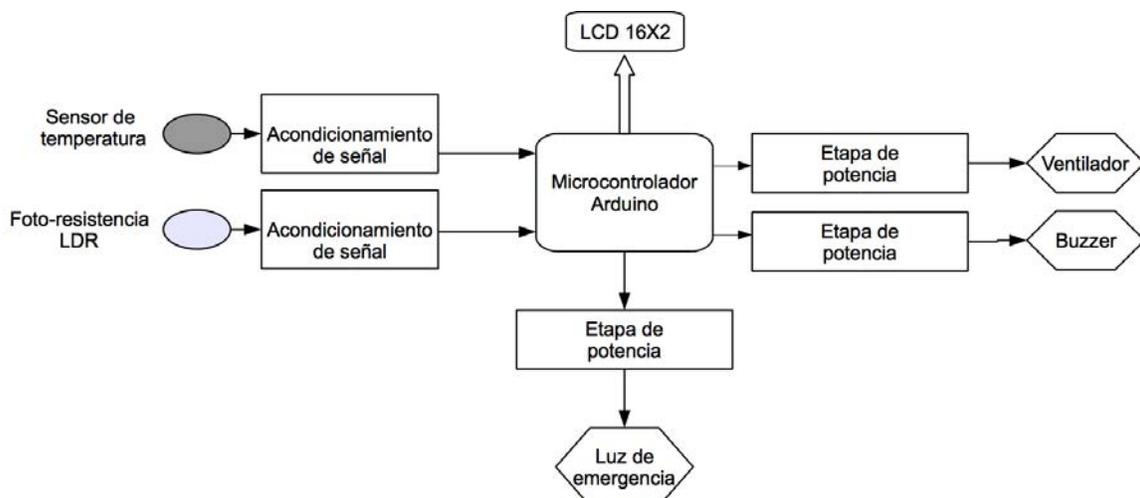
- a) Cuando la temperatura sea mayor que 30°C, el microcontrolador arduino deberá encender un ventilador de 110 Vac y este se deberá apagar hasta que la temperatura sea menor a 29°C.
- b) Cuando el porcentaje de iluminación sea menor a 35%, el microcontrolador deberá activar una luz de emergencia hasta detectar que la iluminación sea mayor al 60%.
- c) Cuando la temperatura sea mayor a 30°C y al mismo tiempo el porcentaje de iluminación menor a 35%, el microcontrolador deberá desplegar un mensaje de alarma en la LCD, activar un buzzer de 5V y una luz de emergencia de 110 Vac.

Nota: Solamente se permite utilizar **sensores analógicos**.

Material:

- 1 Microcontrolador Arduino UNO o similar.
- 1 Sensor de temperatura LM 35 o equivalente.
- 1 Sensor de humedad de suelo YL-83 o equivalente.
- 1 Foto resistencia LDR
- 2 Moc 3011
- 2 TRIAC 2N 6344
- 1 Transistor 2N2222
- 1 Fuente de voltaje 5 Volts
- 1 Computadora personal o Laptop
- 1 software Arduino IDE
- Resistencias varias

Diagrama a bloques



Elaboró: Dr. Everardo Inzunza González

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA ENSENADA
MICROCONTROLADORES
PRACTICA #11

Nombre: Medición de variables ambientales en un invernadero utilizando un microcontrolador Arduino.

Objetivo: Desarrollar un programa para el microcontrolador Arduino para medir la temperatura ambiental, humedad ambiental y humedad de suelo de un invernadero de legumbres .

Requerimientos:

Las variables medidas deberán ser desplegadas en una LCD 16X2 y enviadas por puerto USB a una computadora y que sean desplegadas de forma ordenada (tabla) en la terminal serial de software arduino. Es decir cada columna de la tabla debe corresponder a una de las variables. La frecuencia de muestreo debe ser de 10Hz.

- a) Cuando la temperatura sea mayor que 29°C, el microcontrolador arduino deberá encender un ventilador de 110 Vac y este se deberá apagar hasta que la temperatura sea menor a 28°C y mayor a 20°C. Además activar un buzzer como alerta sonora, desplegar un mensaje en la LCD de Alerta de Temperatura elevada y encender un LED rojo.
- b) Cuando la temperatura sea menor que 20°C, el microcontrolador arduino deberá encender un calentador de 110 Vac (Resistencia eléctrica de potencia) y este se deberá apagar hasta que la temperatura sea menor a 21°C y mayor a 29°C. Además activar un buzzer como alerta sonora, desplegar un mensaje en la LCD de Alerta de Temperatura baja y encender un LED rojo.
- c) Cuando el porcentaje de humedad de suelo sea menor a 30%, el microcontrolador deberá activar una bomba de agua de 110Vac ó 12 VDC para iniciar con el proceso de riego y deberá apagarse la bomba hasta que la humedad sea mayor al 75%. Además activar un buzzer como alerta sonora, desplegar un mensaje de Alerta de Sequia en la LCD y encender un LED rojo.
- d) Mientras que la temperatura esté dentro del rango de 20 °C y 29 °C, la humedad del ambiente esté dentro del rango de 60% a 100% y la humedad de suelo sea mayor al 75%, el microcontrolador deberá encender una luz verde (apagar luz roja), indicando que todo está operando en condiciones normales.
- e) En caso de que alguna de las variables se salga del rango normal (inciso a,b,c), el microcontrolador deberá encender una luz roja (apagar luz verde) e indicar en la LCD cual de las variables se encuentra fuera de rango.

Nota 1: Solamente debe usarse un LED rojo (como indicador de variable fuera de rango) y un LED verde (como indicador de operación normal).

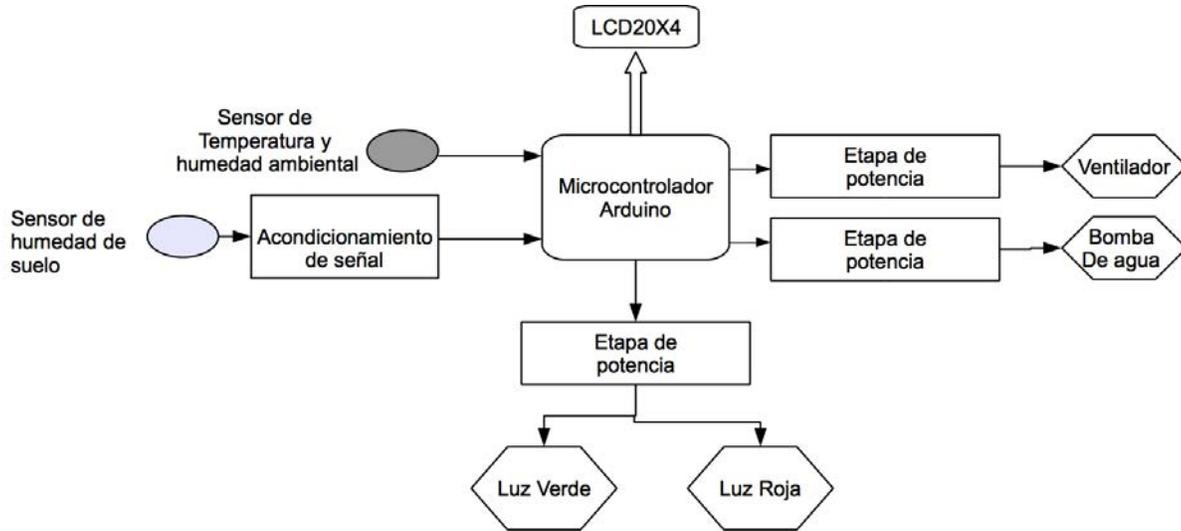
Nota 2. Para la medición de temperatura y humedad ambiental debe usarse el sensor DHT 11 o DHT 22.

NOTA 3. Para la medición de humedad de suelo debe usarse el sensor YL-83.

Material:

- 1 Microcontrolador Arduino UNO o similar.
- 1 Sensor de temperatura y humedad DHT 11 ó DHT 22.
- 1 Sensor de humedad de suelo YL-83 o equivalente.
- 2 MOC 3011
- 2 TRIAC 2N 6344 equivalente
- 1 Transistor 2N2222
- 2 TIP 41C
- 1 Fuente de voltaje 5 Volts
- 1 Computadora personal o Laptop
- 1 software Arduino IDE
- Resistencias varias

Diagrama a bloques



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

PRACTICA #12

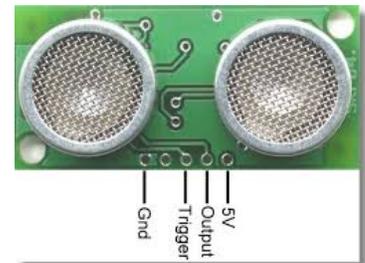
Nombre: Medidor de distancias con ultrasonido

Objetivo #1 : Desarrollar un programa para el microcontrolador Arduino para medir la distancia entre el instrumento y un objeto utilizando el sensor ultrasónico SRF04 en conjunto con el microcontrolador ATmega328P en la plataforma ARDUINO y una pantalla LCD de 20x4.

Requerimientos: El instrumento debe medir la distancia que hay entre el sensor y el objeto, deberá mostrar el resultado en una LCD, y con las siguientes unidades: Cm, metros, Pulgadas, Pies. Además cuando recién se inicialice el instrumento, deberá desplegar un mensaje de bienvenida, los datos de los participantes y finalmente el mensaje “Medidor de distancias ultrasónico”. Utilice despliegue de mensajes en movimiento tipo banner para una mejor apreciación de la información.

Material:

- 1 Tarjeta Arduino Uno R3, Nano ó similar
- 1 Software arduino IDE
- 1 Sensor ultrasónico SRF04
- 1 LCD 20 X 4 o 16X2
- 1 Librería del sensor ultrasónico (descargar de la web)



PROCEDIMIENTO EXPERIMENTAL:

- 1) Interconecte la LCD 16X2 o 20X4 y el sensor ultrasónico a la tarjeta Arduino, siguiendo las especificaciones de los fabricantes.
- 2) Desarrollar un programa en Arduino que cumpla con la función de medidor de distancias (auxíliese de las librerías del sensor ultrasónico).
- 3) Una vez desarrollado todo, realice las pruebas correspondientes para verificar el funcionamiento del circuito y programa.
- 4) Realice los ajustes necesarios hasta que funcione correctamente.

En la figura 1 se muestra un diagrama de tiempos típico de medición de distancias por ultrasonido.

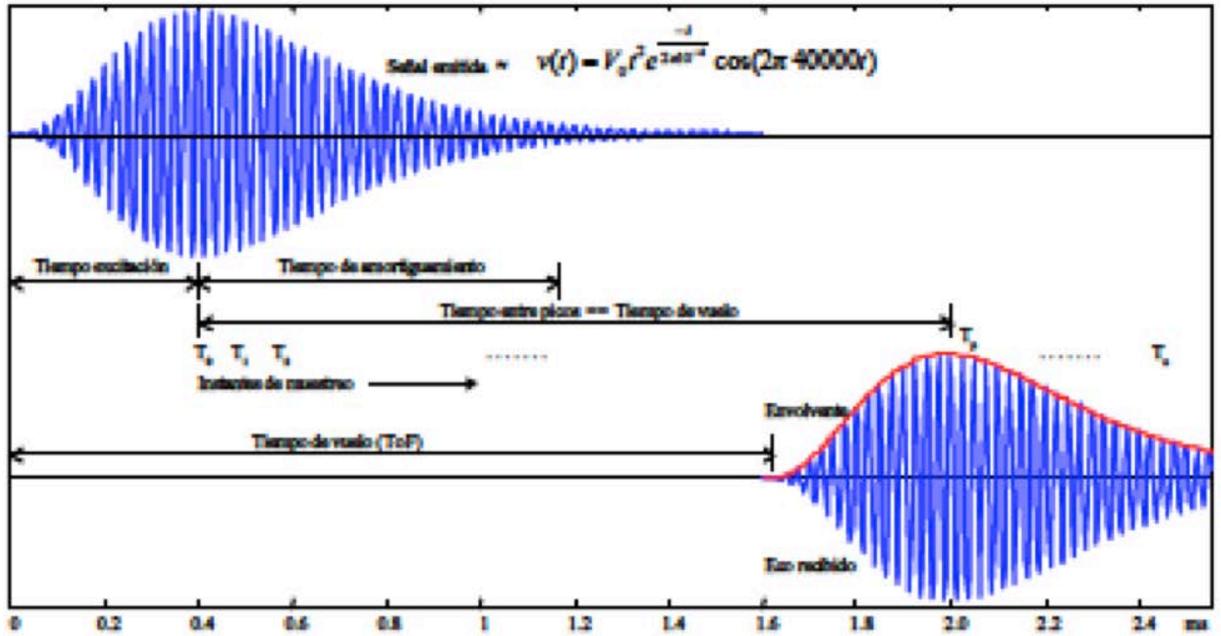


Figura 1. Diagrama de tiempos típico de la medición de distancias por ultrasonido [1].

La figura 2 muestra el esquema a bloques del medidor de distancias por ultrasonido que se propone en este trabajo.

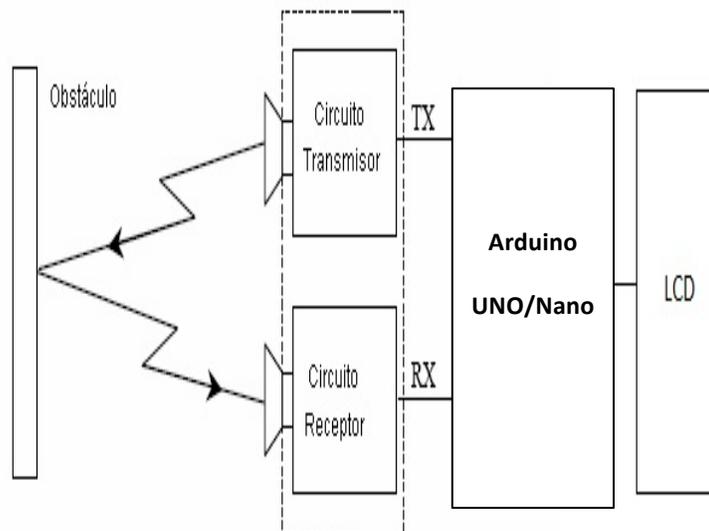


Figura 2. Esquema a bloques del medidor de distancias ultrasónico.

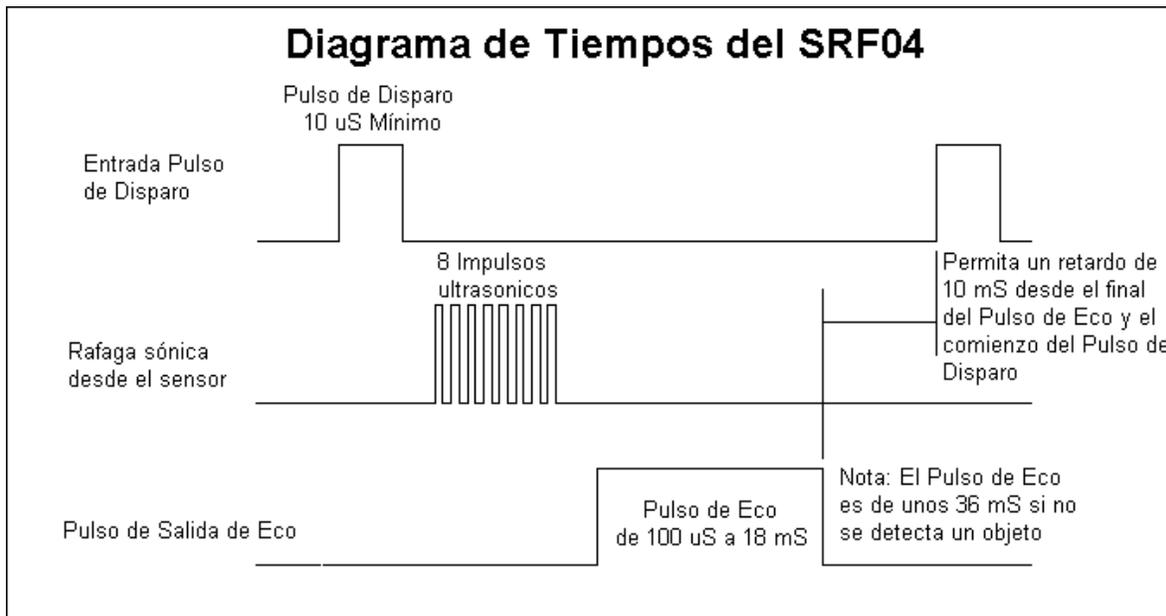


Figura 3. Diagrama de tiempos del sensor ultrasnico SRF04.

Cual es la distancia mnima que puede medir/detectar el sensor ultrasnico?

Terica (segn el fabricante p datasheet)? _____

Experimental (encontrada en el laboratorio)? _____

Cual es la distancia Mxima que puede medir/detectar el sensor ultrasnico?

Terica (segn el fabricante p datasheet)? _____

Experimental (encontrada en el laboratorio)? _____

Investigue al menos 10 aplicaciones de los sensores ultrasnicos y su principio de funcionamiento.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO
MICROCONTROLADORES
PRACTICA #13

Nombre: Comunicación serial con Microcontrolador Arduino y PC.

Objetivo: Desarrollar un programa para el microcontrolador Arduino del tal forma que se comunique de forma serial con una PC y envíe las mediciones de temperatura ambiental, humedad ambiental y temperatura de un estanque.

Requerimientos:

Las variables medidas deberán ser desplegadas en una LCD 20X4 y enviadas por puerto USB a una computadora, de tal forma que sean desplegadas en pantalla y guardadas en un archivo ASCII (TXT) con formato CSV (Comma Separated Value).

- a) Los datos de temperatura del DHT 11/22 y DS 18B20 deberán ser desplegados en la misma figura pero con diferente color las curvas.
- b) Los datos de humedad deberán ser desplegados en una figura diferente.
- c) Todos los datos deberán ser guardados en el mismo archivo ASCII (TXT) con formato CSV.
- d) Para el graficado y almacenamiento de los datos puede usar Matlab™, LabView™, Python, Visual Basic o cualquier otro lenguaje de programación que permita el graficado de la información.

Nota 1: Solamente se permite utilizar **sensores digitales**.

Material:

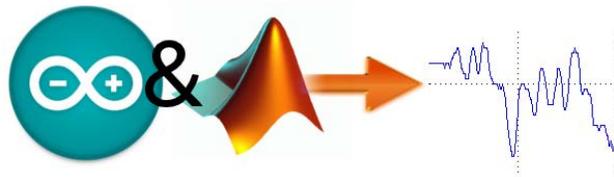
1 Microcontrolador Arduino UNO o similar.
1 Sensor de temperatura y humedad DHT 11 ó DHT 22.
1 Sensor de temperatura DS18B20
1 Computadora personal o Laptop
1 Software (Matlab™, LabView™, Python, Visual Basic u otro)
1 software Arduino IDE
Resistencias varias

PROCEDIMIENTO

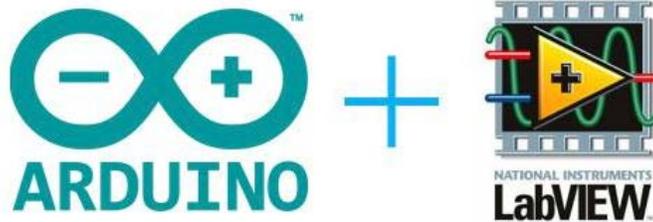
1. Desarrolle el programa Arduino para leer las mediciones de los sensores digitales.
2. Pruebe que las mediciones sean desplegadas en la LCD y enviadas por USB
3. Verifique que los datos sean recibidos en el monitor serie de Arduino.
4. Instale el software Matlab o LabView.
5. Desarrolle el programa en Matlab o Labview para recibir los datos y graficarlos.
6. Realice pruebas para verificar que el programa en Matlab recibe los datos provenientes de Arduino.
7. Modifique el programa en Matlab para que guarde los datos en archivo ASCII con formato CSV.
8. Realice diferentes pruebas experimentales para validar el funcionamiento de todo el sistema.
9. Elabore su reporte de práctica.

Nota 2: No se recomienda tener abiertas simultáneamente el monitor serial de Arduino y el programa en Matlab intentando establecer la comunicación por el puerto COM.

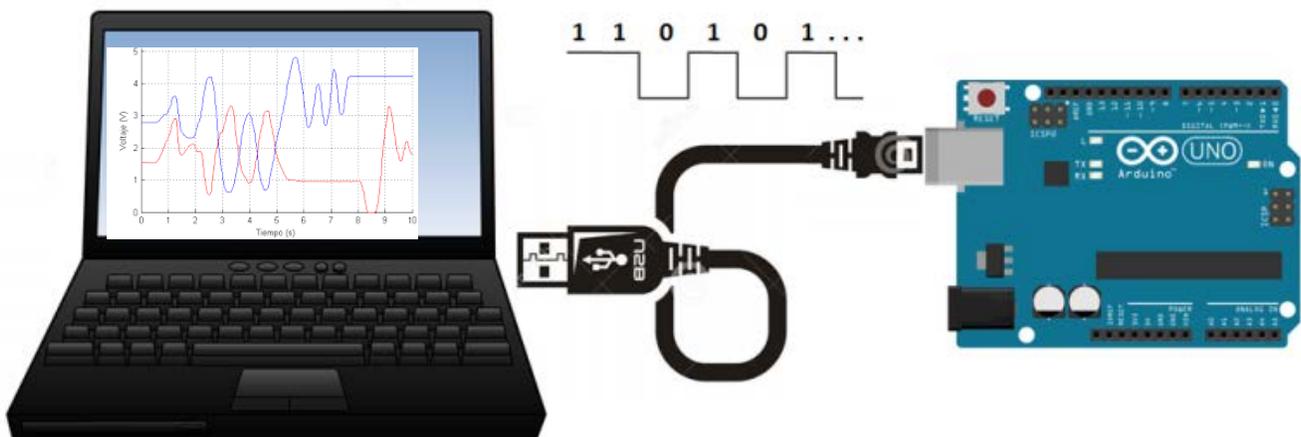
Logotipos de Arduino, Matlab y LabView.



Arduino meets LabVIEW



Esquema de comunicación serial Arduino con PC



Elaboró: Dr. Everardo Inzunza González

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA ENSENADA
MICROCONTROLADORES
PRACTICA #14**

Nombre: Desarrollo de un Datalogger de variables ambientales con memoria EEPROM interna de Microcontrolador.

Objetivo: Desarrollar un programa para llevar el registro de variables ambientales en memoria EEPROM interna al microcontrolador.

Requerimientos:

Las **variables a medir son:** Temperatura utilizando el sensor DS18B20, Humedad relativa del ambiente y temperatura ambiental con el sensor DHT 11 o DHT 22.

El microcontrolador **deberá tener un botón para el iniciar las mediciones.**

El sistema deberá **guardar en la EEPROM el promedio de las mediciones de cada variable** cada 30 segundos.

Es indispensable que esté promediando las mediciones dentro del lapso de los 30 segundos.

Cuando el **espacio utilizado de la EEPROM esté cercano a la capacidad máxima** de almacenamiento (512 Bytes) se deberá encender un LED Rojo.

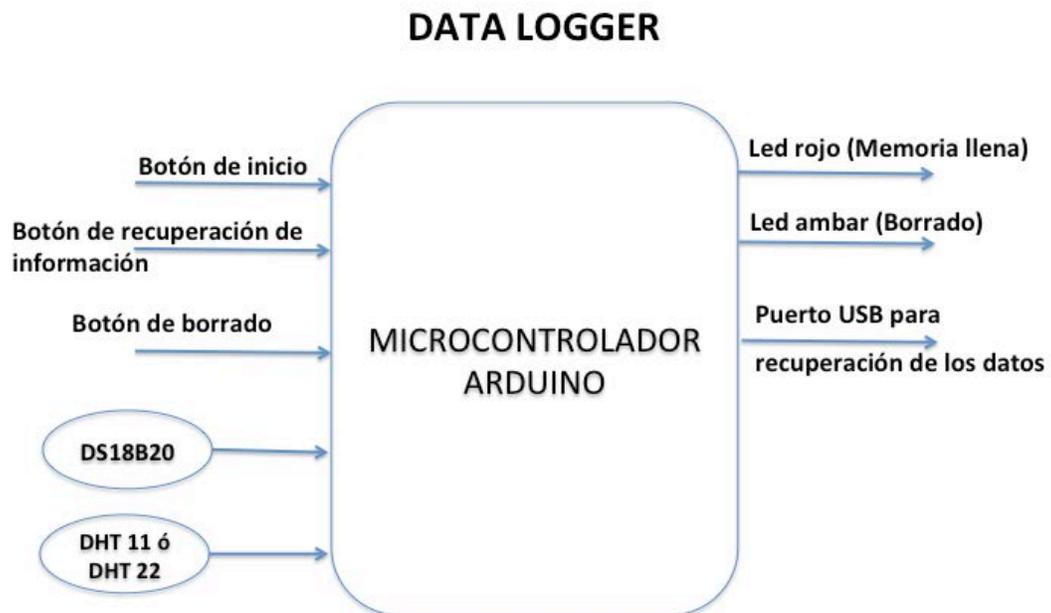
El microcontrolador deberá tener un **botón de recuperación de la información**, de tal forma de que cuando esté activado, todos los datos almacenados en la EEPROM **deberán ser enviados por puerto serial a una PC.** El envío de los datos deberá ser en formato CSV (Comma Separated Value).

Una vez recuperados todos los datos en la PC, el microcontrolador deberá tener un **botón para activar el borrado de toda la información de la EEPROM.** En este caso se recomienda que el Arduino pregunte si está seguro de borrar toda la información?, en caso afirmativo el usuario deberá **enviar una letra 'S'** desde la PC vía USB (serial) confirmando que se desea borrar la información, en caso contrario deberá enviar la letra **'N'** **para cancelar el borrado de la información.** Una vez borrada la información deberá parpadear tres veces un led color ambar (cada 500 mS) como señal de que se borró la información.

Material:

- 1 Microcontrolador Arduino UNO o similar.
- 1 Sensor de temperatura y humedad DHT 11 ó DHT 22.
- 1 Sensor de temperatura DS18B20.
- 1 Computadora personal o Laptop
- 1 software Arduino IDE
- 1 dip switch de 4 pines u 8 pines.
- 2 Push button

Diagrama a bloques



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA ENSENADA
MICROCONTROLADORES
PRACTICA #15

Nombre: Uso, manejo y configuración de teclado matricial con microcontrolador.

Objetivo: Desarrollar un programa para el control de acceso mediante **chapa eléctrica con solenoide**, de tal forma que el Arduino lea una clave de 6 dígitos introducida con teclado matricial.

Requerimientos:

- Se requiere el desarrollo de un programa para control de acceso mediante chapa eléctrica con solenoide, de tal forma que el Arduino lea una clave de 6 dígitos introducida con teclado matricial, cuando esta sea correcta deberá activar un solenoide durante 5 segundos, para que permita el acceso de una persona, en caso contrario y después de 3 intentos fallidos, deberá activar un buzzer como señal de alarma (durante 3 minutos, desplegar un mensaje de clave incorrecta en la LCD y por motivos de seguridad mantener bloqueado el acceso durante 3 minutos. Después de terminado este tiempo, el sistema deberá solicitar de nuevo la clave de acceso para ingresar.
- Al inicio de la ejecución del programa, en la LCD deberá aparecer un mensaje solicitando la clave de acceso, cuando el usuario teclee la clave y por seguridad (para evitar plagio de claves), deberán de aparecer solamente asteriscos (*).

Nota: En este ejemplo **no** se requiere la comunicación con la PC, es indispensable usar LCD.

Material:

- 1 Microcontrolador Arduino UNO o similar.
- 1 Teclado matricial 4X4.
- 1 LCD 16 X 2 ó LCD 20X4 Serial
- 1 Computadora personal o Laptop
- 1 software Arduino IDE
- 1 dip switch de 4 pines u 8 pines.
- 2 Push button

Resistencias varias