

**INSTITUTO POLITÉCNICO NACIONAL  
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS  
AVANZADAS**

## **Práctica No. 8. Control de puertos.**

**Unidad Temática:** V Manejo de Archivos y Puertos

**Lugar de realización:** Laboratorio de Cómputo

**Duración:** 6 hrs.

### **Objetivo**

Desarrollar un programa para el envío de información a través de los puertos.

### **Resultados Esperados**

- Identificar los componentes para el envío y recepción de elementos en C.
- Realizar dispositivos digitales que se conectarán a un programa en C con ayuda de su profesor.
- Desarrollar un programa que reciba información a través del puerto serial.
- Uso de variables provenientes de elementos externos para generar programas más completos.
- Comprender las bases para modificar parámetros digitales y analógicos como luces, motores, pantallas, entre otros.



### **Pre-reporte.**

Para el buen desarrollo de la práctica el alumno entregará un trabajo previo que incluya:

1. Definición y partes de puerto serial.
2. Investigación e instalación de Arduino manipulando el ejemplo blink.

**Nota:** El pre-reporte cuenta como calificación de la práctica, en caso de no entregar el pre-reporte se le restará 25% a dicha calificación.

*"La primera necesidad es comunicarse"  
(Teresa de Calcuta)*

## Material y Equipo.

- Arduino UNO
- IDE Dev-Cpp o compatible.
- Computadora
- Resistencias
- Led rojo
- Sensor de temperatura TMP36

## Introducción.

### Puerto Serie

Un puerto serie, puerto serial o puerto de comunicación COM (también llamados RS-232) es una interfaz de comunicaciones de datos digitales donde la información es transmitida bit a bit enviando **un solo bit** a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente. Se denomina “serial” porque el puerto serie “serializa” los datos. Esto quiere decir que toma un byte de datos y transmite los 8 bits del byte de uno en uno.

El símbolo del puerto es:

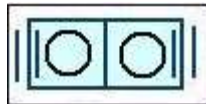


Figura 8.1. Símbolo de puerto

El puerto serie puede ser un conector de 25 o de 9 pines.



Figura 8.2. Conector de puerto serie

En el caso del de 9 pines:

| Nombre | Descripción   | Sentido | Pin |
|--------|---|---------|-----|
| TXD    | Transmit Data   | Salida  | 3   |
| RXD    | Receive Data  | Entrada | 2   |
| RST    | Request to send (Reconoce al modem que UART está lista para intercambiar datos) | Salida  | 7   |
| CTS    | Clear to Send. (cuando el modem está listo para el intercambio)                 | Entrada | 8   |
| DTR    | Data Terminal Ready   | Salida  | 4   |
| DSR    | Data Ready State (UART establece el vínculo)                                    | Entrada | 6   |
| DCD    | Data Carrier Detected. Se activa en caso de un acarreo                          | Entrada | 1   |
| RI     | Ring Indicator. Se activa cuando el modem detecta una señal ring del PSTN       | Entrada | 9   |
| GND    |   | Masa    | 5   |

### Puerto serial en C

Para programar se usa la librería es **dos.h** y se debe:

- Se debe de definir el puerto pasado en la dirección del COM correspondiente  

```
#define PORT1 0x3F8
```
- Se configura el puerto tomando en cuenta: bit de paridad, bit de paro y baudios

#### Configuración de puerto

```

outportb(PORT1 + 0 , 0x06); /*tasa de baudios Low Byte*/
outportb(PORT1 + 1 , 0x00); /*tasa de baudios High Byte*/
outportb(PORT1 + 3 , 0x03); /*8 Bits, NoParity, 1StopBit*/
outportb(PORT1 + 2 , 0xC7); /*Registro de Control FIFO */
outportb(PORT1 + 4 , 0x0B); /*Enciende DTR, RTS, y OUT2*/

```

- Para recibir se usa `inport` y se puede manipular como un caracter  

```
c = inportb(PORT1 + 5);
```
- Para enviar se usa `outport` y envía carácter por carácter de forma serlizada  

```
outportb(PORT1, ch);
```

Nota: Podría ser necesario instalar los drivers en caso de que no los tenga configurada la computadora

## Desarrollo.

### Programa

Construir un programa que permita censar la temperatura del medio ambiente, si la temperatura rebasa los 200°F encender la alerta audible. Siga los siguientes pasos:

1. Construir el circuito según el siguiente diagrama. Recuerde que el TMP36 es de bajo voltaje y permite ser usado entre los 2.7V y los 5.5V, lo cual es ideal para el Arduino.

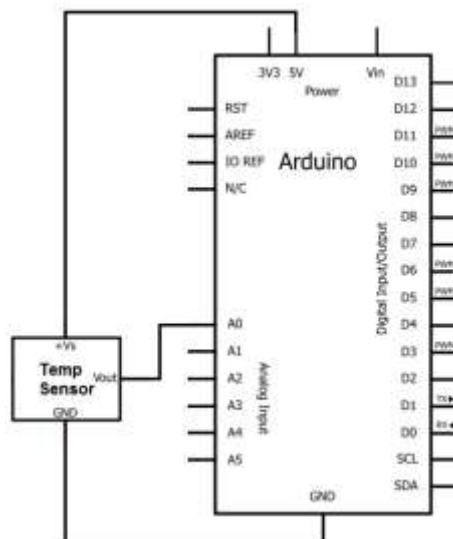


Figura 8.3. Diagrama del circuito

2. Codifique el siguiente programa en Arduino, que lea el sensor y envíe ese dato al programa en lenguaje C a través del puerto serial.

```
int sensorPin = 0; //pin analógico
void setup() //inicializa
{
    Serial.begin(9600); //Comienza conexión serial
}
void loop() //Ejecución continua
{
    int reading = analogRead(sensorPin);
    //convertir a voltaje
    float voltage = reading * 5.0;
    voltage /= 1024.0;
    float temperatureC = (voltage - 0.5) * 100 ;
    Serial.print(temperatureC); //envía por serial
    Serial.println(" degrees C");
    delay(1000); //pausa de un seg
}
```

3. Pruebe con el monitor serial de Arduino, (imprima pantalla para su reporte)
4. Escriba el programa en lenguaje C que permita:
  - a. Recibir los datos desde serial.
  - b. Hacer la conversión a °F
  - c. Escribir el código que permita evaluar si la temperatura sensada excede los 200°F; en caso de ser así emitir una alerta audible
5. Grafique la temperatura sensada

### Problema de Aplicación

Haciendo uso de sensores y programas en lenguaje C resuelva el siguiente problema:

**Cortina automática.** Se requiere evaluar el nivel de luminosidad en el exterior para determinar la apertura o no de una cortina controlada por un motor a pasos. En caso de cierre o apertura de la cortina imprimir en pantalla la hora, la acción (apertura o cierre) y el nivel de luz censado.

**Propuesta de solución.** Utilice una fotocelda para medir la luz en el exterior (esta parte requiere un código similar al mostrado en el programa anterior), determine los valores que darán pie a las acciones de apertura y cierre. Configure el motor a pasos para que gire en ambos sentidos según sea el caso.

### Proyectos (opcionales).

**Caja registradora.** Hacer un programa en C que permita el cobro de productos a través del código de barras, obtener la cuenta y al cobrar abrir la gaveta correspondiente con un motor a pasos.

**Invernadero.** Monitorear la temperatura, humedad y luminosidad de un invernadero, en caso de que la temperatura suba activar un ventilador, si la humedad baja activar regadera y si la luminosidad del lugar es baja aumentarla progresivamente.

### Ponderación de la Práctica.

| Sección     | Elemento a Evaluar                      |
|-------------|---|
| Pre-reporte | Antecedentes de puerto serial y Arduino |
| Ejercicio 1 | Construcción del Circuito               |
| Ejercicio 2 | Codificación del puerto                 |
| Ejercicio 3 | Uso del monitor serial de Arduino       |
| Ejercicio 4 | Uso de funciones y puerto serial        |

### Bibliografía.

Deitel P.J.y Deitel H. M., Como Programar C++, Ed. Prentice Hall, 6<sup>a</sup> Impresión, México, 2009, ISBN: 970-26-1273-X, Págs: 1-1050.

Deitel P.J. y Deitel H. M., Como programar en C#, Ed. Prentice Hall, 2<sup>a</sup> Impresión, México, 2007,ISBN: 9702610567, Págs: 1-1080.

Guardati, Silvia, Estructura de Datos Orientado a Objetos con C ++, Ed.Prentice, México 2007, ISBN: 9702607922, Págs: 1–183.

**NOTA:** Presentar el reporte en un documento (PDF o DOC), el código fuente y la impresión de la pantalla de ejecución.

**"Puedes apoyarte de tus compañeros y profesor para aclarar tus dudas"**