

# Puerto Paralelo

# Comunicación en paralelo

- La comunicación en paralelo permite enviar información en un bus de comunicación.
- Ventajas:
  - Se pueden transferir varios bits en forma simultánea
- Desventajas
  - Corta distancia para transmisión

# Puerto paralelo

- El puerto paralelo es una interfaz paralela (como su nombre lo indica)
- Permite conectar dispositivos externos
- Típicamente se utilizaba para conectar impresoras
- Conector DB-25



- Se conectaba a las impresoras por medio de una interfaz Centronics

# Interfaz Centronics

- La interfaz Centronics permite la conexión de una impresora a una computadora a través del puerto paralelo



# Puerto paralelo

- El puerto paralelo se utiliza para conectar otros dispositivos a la computadora
- En algunas ocasiones se utiliza para realizar la comunicación entre dos computadoras
- Cuenta con un bus de 8 bits de datos, además de las señales de control

# Características eléctricas

- Nivel alto: 3.3 V o 5 V
- Nivel bajo: 0 V
- Corriente máxima de salida: 2.6 mA
- Corriente máxima de entrada: 24 mA

# Puertos paralelos

- Dependiendo de la computadora, típicamente se tienen 1, 2 o 3 puertos (generalmente 1 o 2)

<b>Puerto</b>	<b>Interrupción</b>	<b>Dirección inicial</b>	<b>Dirección final</b>
LPT1	IRQ 7	0x378	0x37F
LPT2	IRQ 5	0x278	0x27F
LPT3	IRQ 7	0x3BC	0x3BF

# Puertos paralelos

- La ubicación de los puertos está ubicada en memoria
- Hasta antes de W7, era posible usar el programa “debug” para revisar el contenido de esta memoria, ya que venía incluido con el sistema operativo.

Dirección en memoria	Dirección de puerto
0040:0008h	LPT1
0040:000Ah	LPT2
0040:000Ch	LPT3
0040:000Eh	LPT4

# Puertos paralelos

- Como alternativa, se puede usar el programa de línea de comandos que se incluye en Visual Studio
- En esta herramienta sí aparece el programa “debug”

Dirección en memoria	Dirección de puerto
0040:0008h	LPT1
0040:000Ah	LPT2
0040:000Ch	LPT3
0040:000Eh	LPT4

# Terminales del puerto



# Terminales del puerto

Pin No (D-Type 25)	Pin No (Centronics)	SPP Signal	Direction In/out
1	1	nStrobe	In/Out
2	2	Data 0	Out
3	3	Data 1	Out
4	4	Data 2	Out
5	5	Data 3	Out
6	6	Data 4	Out
7	7	Data 5	Out
8	8	Data 6	Out
9	9	Data 7	Out
10	10	nAck	In
11	11	Busy	In
12	12	Paper-Out / Paper-End	In
13	13	Select	In
14	14	nAuto-Linefeed	In/Out
15	32	nError / nFault	In
16	31	nInitialize	In/Out
17	36	nSelect-Printer / nSelect-In	In/Out
18 - 25	19-30	Ground	Gnd

# Trabajo

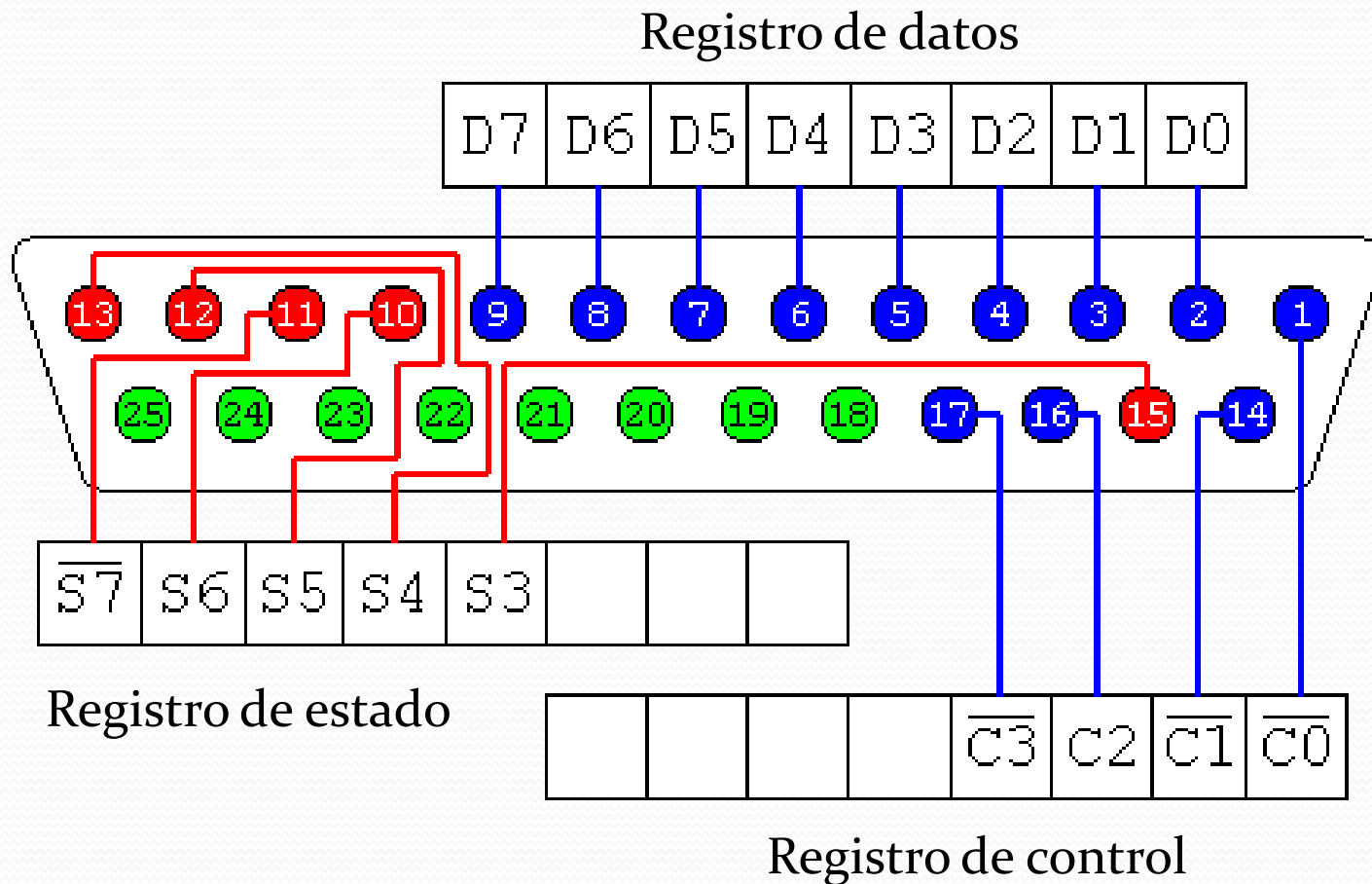
- Consultar para qué se usa cada una de las líneas del puerto paralelo, en la interacción con la impresora.
- Entrega: Próxima sesión (integrado en libreta)

# Registros del puerto paralelo

- Para acceder al puerto paralelo, se utilizan registros ubicados en las direcciones de E/S correspondientes a cada puerto.

Registro	Ubicación	E/S	Ubicación LPT1
Registro de datos	Dir. Base + 0	S	0x378
Registro de estado	Dir. Base + 1	E	0x379
Registro de control	Dir. Base + 2	S	0x37A

# Correspondencia con pines



# Escritura en puerto paralelo

- Tradicionalmente se usan las instrucciones de acceso a puertos proporcionados por algunos compiladores
  - `outp`
  - `_out`
  - `outportb`
- Los sistemas operativos actuales no permiten el acceso directo a los puertos a los programas de aplicación
- ¿Por qué?

# Esquema de protección

# DLL para entrada/salida por puertos

- <http://logix4u.net/>
- Inpout32.dll for Windows 98/2000/NT/XP
- Permite el acceso a los puertos de la computadora
- Trabaja con el nivel de privilegio 0
- El DLL se coloca en la carpeta `Windows\System32`

# Acceso a puertos con C#

- Se debe importar el DLL

```
namespace Puerto1_CSharp
{
    class PortAccess
    {
        [DllImport("inpout32.dll",
EntryPoint = "Out32")]
        public static extern void Output
(int address, int value);
    }
}
```

# Acceso a puertos con C#

- Se usa el DLL

```
namespace Puerto1_CSharp
{
    class Program
    {
        static void Main(string[] args)
        {
            PortAccess.Output(0x378, 0x01);
            Console.WriteLine("Presione ENTER");
            Console.ReadLine();

            PortAccess.Output(0x378, 0x00);
        }
    }
}
```