



# Salida Comunicación Serial

La **conexión USB** utilizada por Galileo para la carga de Sketch puede ser utilizada de igual forma para **intercambiar información** con el computadora.

Esta salida permite mostrar información en un formato amigable con los seres humanos!



La comunicación serial es una herramienta invaluable para la depuración de aplicaciones, es tan sencillo su uso que basta 1 línea de código para habilitar este periférico.



Los datos transmitidos por el puerto serial pueden ser sometidos a un formato de transmisión. Mas información en **Arduino Language Reference**.

```
Serial.print(32, HEX);
```

# Serial.begin() Serial.print() Serial.println()

---

## Serial.begin()

Rutina para abrir el puerto serial de Galileo. Junto a la apertura del puerto se debe de definir la velocidad de transmisión de datos.

### Sintaxis:

```
Serial.begin(SPEED)
```

**SPEED:** Velocidad en bits por segundo.

## Serial.print()

Esta rutina da inicio a la transmisión de caracteres a través del puerto serial.

### Sintaxis:

```
Serial.print(VALOR)
```

**VALOR:** cualquier tipo de dato, desde cadenas hasta caracteres.

## Serial.println()

Exhibe el mismo comportamiento que Serial.print(), pero a diferencia de este, Serial.println() agregar una salto de línea después de la cadena transmitida.

# Código blinkSerial.ino

```
int led = 13;
```

```
void setup() {  
  pinMode(led, OUTPUT);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  Serial.print("ALTO\n");  
  digitalWrite(led, HIGH);  
  delay(1000);  
  Serial.println("BAJO");  
  digitalWrite(led, LOW);  
  delay(1000);  
}
```

Inicializa el puerto serial con una velocidad de transmisión de 9600 bits por segundo.



**IMPORTANTE:** El receptor de la comunicación serial debe estar configurado a la misma velocidad de transmisión, de lo contrario la transmisión de información no se llevara a cabo de manera satisfactoria.

El uso de Serial.println() nos ahorra la necesidad de incluir el carácter de retorno de línea ("\n") en cada cadena enviada.



# Entrada Comunicación Serial

Como se vio anteriormente, la tarjeta Arduino cuenta con un puerto de comunicación serial, con el cual se puede intercambiar información entre otros dispositivos, permitiendo comunicarnos con la tarjeta de manera mas amigable.



Este puerto es Full-Duplex, por lo que puede enviar y recibir información al mismo tiempo, comportándose entonces como una terminal de Entrada/Salida.

# Serial.begin() Serial.available() Serial.read()

---

## Serial.begin()

Inicializa el puerto serial de Galileo y define el «**baud rate**» (Velocidad de transmisión de datos).

### Sintaxis:

```
Serial.begin(speed)
```

speed -> Baud Rate

## Serial.available()

Esta función devuelve el número de bytes disponibles para ser leídos en el puerto serial..

### Sintaxis:

```
Serial.available()
```

## Serial.read()

Lee los datos que entran en el puerto serial.

Regresa -1 si no hay datos disponibles.

### Sintaxis:

```
Serial.read()
```

# Código SerialInput.ino

```
int ledPin = 13;
int inByte = 0;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
}

void loop() {
  if (Serial.available() > 0) {
    inByte = Serial.read();
    if (inByte == 'a') {
      if(digitalRead(ledPin)){
        digitalWrite(ledPin, LOW);
      }
      else {
        digitalWrite(ledPin, HIGH);
      }
    }
  }
}
```

Variable que recibirá el byte de entrada en el puerto serial

Inicializa el puerto serial con una velocidad de transmisión de 9600 bits por segundo.

La terminal 13 se configura como salida.

Se comprueba que existan datos disponibles para su lectura en el puerto serial

Lectura del dato entrante en el puerto serial

Encendido y apagado del LED.



**IMPORTANTE:** El transmisor de la comunicación serial debe estar configurado a la misma velocidad del receptor, de lo contrario la transmisión de información no se llevara a cabo de manera satisfactoria.