



```
Ejemplo_Serial | Arduino 1.8.0
File Edit Sketch Tools Help
Ejemplo_Serial
//#include <SoftwareSerial.h>
//SoftwareSerial XBee(3,2);

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  //XBee.begin(9600);
}

void loop() {
  // XBee.available() {
  // XBee.write(Serial.read());
  //Serial.write(XBee.read());
  Serial.write(Serial1.read());
}
if (Serial.available()) {
  // XBee.write(Serial.read());
  Serial1.write(Serial.read());
}
}
```

Arduino Leonardo on COM7

Comunicación Serial Arduino UNO vs Leonardo

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 of. 1105, Providencia, Santiago, Chile.
+56 2 23339579 | www.olimex.cl | info@olimex.cl

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 Oficina 1105
Providencia, Santiago, Chile

www.olimex.cl
info@olimex.cl

Tel: +56 2 23339579
Fax: +56 2 23350589

® MCI Ltda. 2017

INTRODUCCIÓN

El Leonardo es la primera placa de desarrollo de Arduino que utiliza un microcontrolador con USB incorporado. Usando el ATmega32U4 como su único microcontrolador permite que sea más barato y más simple. También, debido a que el 32U4 está manejando el USB directamente, están disponibles bibliotecas de código que permiten a la placa emular un teclado de computadora, mouse y más usando el protocolo USB HID!

Tiene 20 pines digitales de entrada / salida (de los cuales 7 puede utilizarse para salidas PWM y 12 entradas como analógicas), un oscilador de 16MHz, una conexión micro USB, un conector de alimentación, un cabezal ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB, adaptador AC a DC o batería para empezar.

TABLA COMPARATIVA DE ARDUINO UNO CON LEONARDO

Característica	Leonardo	UNO
Microcontrolador	ATmega32u4	ATmega328
Entrada y salida Digitales	20*	20**
Entradas analógicas	12	6
Salidas PWM	7	6
Comunicación I2C	SDA:D2, SCL:D3	SDA:A4, SCL:A5
Interrupciones Externas	5	2
Memoria de programa (FLASH)	32 KB (4KB para el bootloader)	32KB (0.5KB para el bootloader)
Memoria de datos (SRAM)	2.5KB	2KB
Memoria auxiliar (EEPROM)	1KB	1KB

*El Arduino Leonardo tiene 20 entradas y salidas digitales, de las cuales 7 se pueden configurar como PWM y 12 se pueden configurar como entradas analógicas A0-A5 y A6-A11 que son los pines digitales 4,6,8,9,10 y 12

**El Arduino Uno tiene 20 entradas y salidas digitales de las cuales 6 se pueden configurar como PWM y 5 como entradas analógicas A0-A5.

PUERTO SERIALES

El Arduino Leonardo posee 2 puertos seriales, el primero llamado Serial que es por el cual cargamos el código a nuestra tarjeta y un puerto adicional llamado Serial1 en los pines 0(RX) y 1(TX) que nos sirve para comunicarnos con otro dispositivo como: 3GBee, BLEBee, BlueBee, GPRSBee, entre otros; la diferencia con el Arduino UNO es que éste solo posee un puerto Serial, y se debe hacer uso de la librería SoftwareSerial() en los pines 2(TX) y 3(RX) en caso de querer comunicarnos con otro dispositivo como los anteriormente mencionados y mantener comunicación con el PC. En la siguiente imagen puedes tener una referencia de cómo van conectado las líneas de comunicación en las diferentes tarjetas.

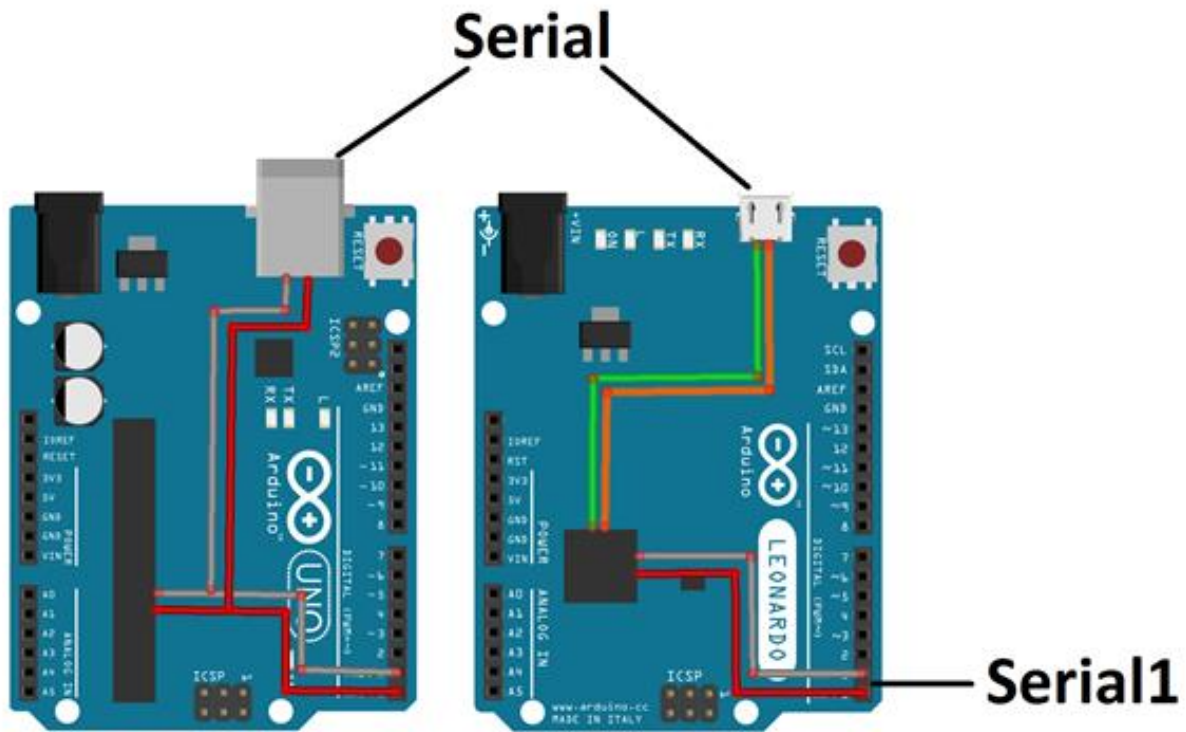


Figura 1. Comunicación Serial

Como se observa las líneas de comunicación del Arduino UNO van al USB y al mismo tiempo a los pines 0 y 1 por lo cual no podemos conectarnos a otros dispositivos si tenemos el Arduino conectado al PC, en comparación con la imagen del Arduino Leonardo vemos que son líneas diferentes lo cual es un beneficio y de esta manera se tiene más pines disponibles ya que no debes usar los pines 2 y 3 para comunicarte con otro dispositivo.

En el siguiente ejemplo se puede observar como pasar un código de Arduino UNO para el Arduino Leonardo o viceversa. Se comentarán las líneas que no se van a utilizar:

Arduino UNO	Arduino LEONARDO
<pre>#include <SoftwareSerial.h> SoftwareSerial XBee(3,2); void setup() { Serial.begin(9600); //Serial1.begin(9600); XBee.begin(9600); } void loop() { if (XBee.available() > 0) { //if (Serial1.available() > 0) { Serial.print(XBee.read()); //Serial.print(Serial1.read()); } if (Serial.available()) { XBee.print(Serial.read()); //Serial1.print(Serial.read()); } }</pre>	<pre>///#include <SoftwareSerial.h> ///SoftwareSerial XBee(3,2); void setup() { Serial.begin(9600); Serial1.begin(9600); //XBee.begin(9600); } void loop() { //if (XBee.available() > 0) { if (Serial1.available() > 0) { //Serial.print(XBee.read()); Serial.print(Serial1.read()); } if (Serial.available()) { // XBee.print(Serial.read()); Serial1.print(Serial.read()); } }</pre>

Tabla 1. Modificación del código.

Estas tarjetas usan la comunicación Serial para transferir información con el computador y la Serial1 para controlar los módulos que se encuentren en el socket XBee compatible.

A continuación presentamos las funciones que normalmente se usan de la puerta Serial1:

Serial1.begin(9600); // Inicialización de la puerta Serial1 que va al socket XBee compatible

Serial1.available(); // Entrega la cantidad de datos que están en el puerta Serial1

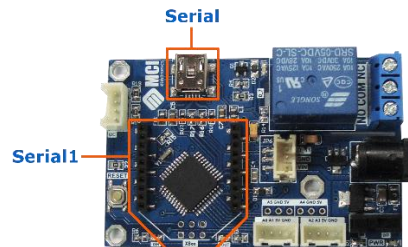
Serial1.read(); // Lee los datos que le envía el módulo que está en el socket

Serial1.print("HOLA"); // Envía la palabra HOLA al módulo que se encuentra en el socket

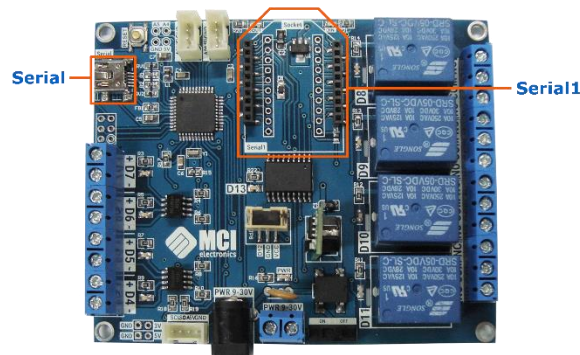
TARJETAS ARDUINOS LEONARDO COMPATIBLES

A continuación se presentan las tarjetas compatibles con Arduino Leonardo, es decir que ocupan el mismo microcontrolador Atmega32u4.

- **TIC (Arduino Leonardo compatible) SKU:[MCI01922](#)**
 - ✓ Conector USB mini B o micro USB (dependiendo de la versión)
 - ✓ Resistencias pull-up disponibles para comunicación I2C (pines digitales D2 y D3)
 - ✓ Socket compatible con formato XBee, con regulador de voltaje de 3.3V @ 500mA
 - ✓ Conectores Grove con pines análogos disponibles, A0-A1 y A2-A3
 - ✓ Relé 220VAC o 30VDC, con conectores Terminal Block de 3 pines 5mm
 - ✓ Switch ON/OFF



- **XBee IO PRO V2 R2 (Arduino Leonardo compatible) SKU:[MCI01567](#)**
 - ✓ Socket XBee (compatible con XBee, BlueBee, 3GBee y GPRSbee)
 - ✓ 4 entradas optoacopladas
 - ✓ 4 salidas con relés
 - ✓ Entradas análogas (Grove)
 - ✓ Power Jack 3.5mm
 - ✓ Switch On/Off
 - ✓ Conector JST para módulos GPRSbee y 3GBee
 - ✓ Voltaje de entrada de 9VDC a 36VDC
 - ✓ Comunicación I2C (Grove)
 - ✓ Resistencias pull-up disponibles para comunicación I2C (pines digitales D2 y D3)



Revisión	Fecha	Editado por	Descripción/Cambios
1.0	27 de Febrero de 2017	Alexandra Vasquez	Versión inicial del documento