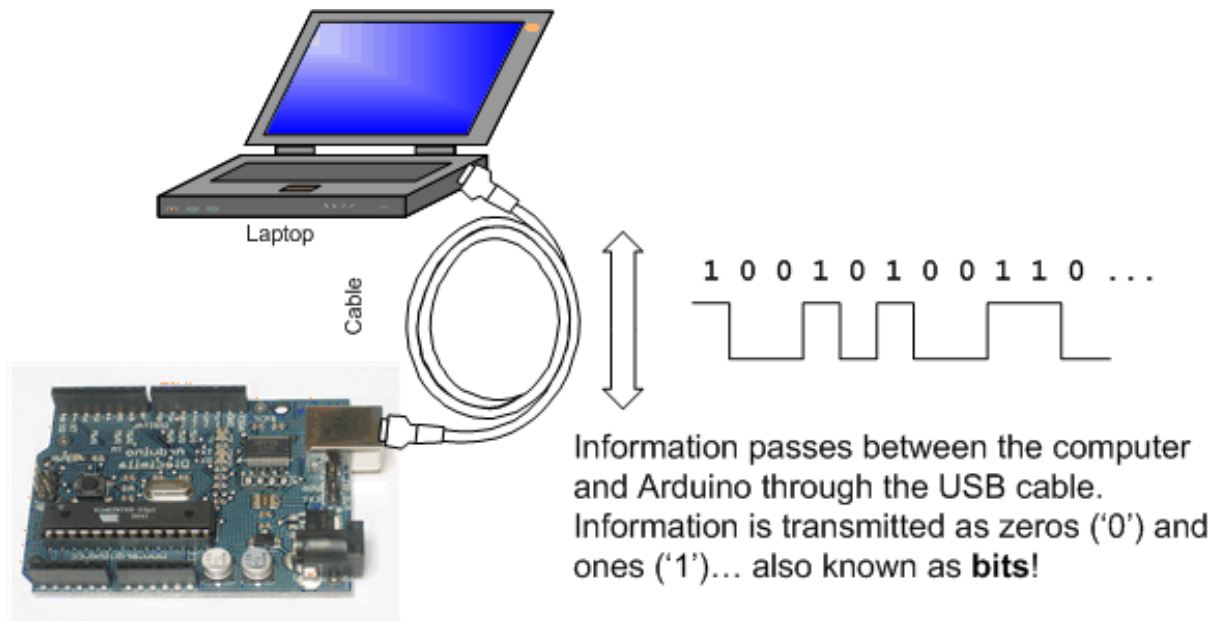


## Introduction



- **Compiling** turns your program into binary data (ones and zeros)
- **Uploading** sends the bits through USB cable to the Arduino
- The two LEDs near the USB connector blink when data is transmitted
  - **RX** blinks when the Arduino is receiving data
  - **TX** blinks when the Arduino is transmitting data

## USART Communication

### a) `Serial.begin(baud_rate)`

- `baud_rate` : The baud rate that will be used for serial communication. Can be 4800, 9600, 14400, 19200, etc.
- This function is used to define the baud rate that will be used for serial communication. For communicating with specific devices, the device baud rate needs to be used.
- Example `Serial.begin(9600)` defines 9600 baud rate for communication.

### b) `Serial.available()`

- This function is used to get the number of bytes available for reading from the serial port. It gives the number of bytes of data that has arrived and is stored in the serial receive buffer.
- Example `if(Serial.available())`  
If data available at serial port, take action.

### c) `Serial.print(value)`

- `value` : character, string, number to be printed.

- This function is used to print data to a serial port in a form that is human readable (character, strings, numbers).
- Example `Serial.print("Hi 1234")`  
Prints Hi 1234 on the serial monitor.

**d) `Serial.println(value)`**

- value : character, string, number to be printed.
- This function is used to print data to a serial port in a form that is human readable (character, strings, numbers) followed by a carriage return (`\r`) and a new line character (`\n`).

**e) `Serial.read()`**

- This function returns a character that was received on the Rx pin of Arduino.
- Example `char read_byte`  
`read_byte = Serial.read()`  
Byte of data read is stored in `read_byte`.

**f) `Serial.write(value)`, `Serial.write(string)`, `Serial.write(buff, length)`**

- value : value to be sent as a single byte.
- string : string to be sent as a series of bytes.
- buff : array of data to be sent as bytes.
- length : number of bytes to be sent.
- This function writes data in binary form to the serial port. Data is sent in form of bytes or series of bytes.
- Example `Serial.write(100)`  
`Serial.write("Hello")`

**Exercise:**

Write a program to print data received through serial communication on pressing a switch to the serial monitor of Arduino