

## NIELIT, Gorakhpur

**Course Name: A-level (1<sup>st</sup> Sem.)**

**Subject:IoT**

**Topic: USART In Arduino**

**Date: 17.04.2020**

### **Introduction**

USART (Universal Serial Asynchronous Receiver Transmitter) is a serial communication protocol used to transmit/receive data serially at a specific baud rate.

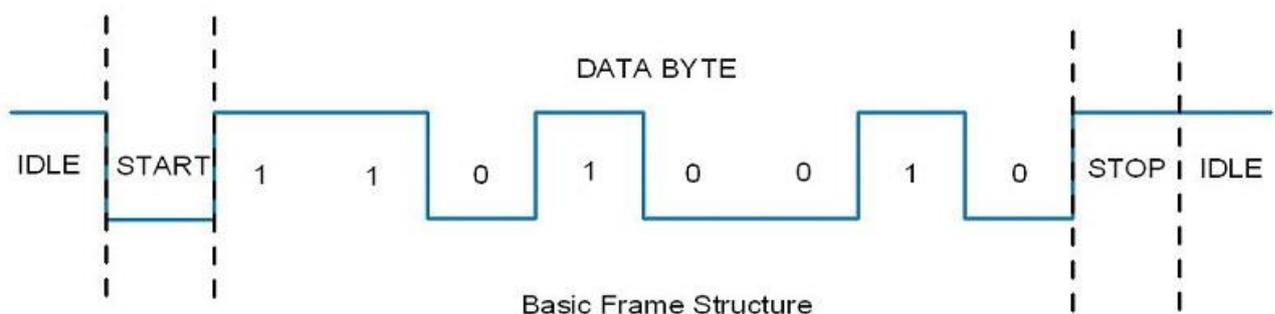
Several devices such as GPS, GSM, RFID, sensors, etc need to communicate with the microcontroller for transmitting or receiving information. To communicate with the microcontroller, several communication protocols are used such as RS232, SPI, I2C, CAN etc. Basically, a protocol is a set of rules agreed by both, the sender and the receiver, on -

- How the data is packed?
- How many bits constitute a character?
- When the data begins and ends?

With the help of USART, we can send / receive data to a computer or other devices. USART is also used in interfacing microcontroller with various modules like Wi-Fi (ESP8266), Bluetooth, GPS, GSM, etc.

Let us start with the USART based serial communication.

**Asynchronous Communication:** Asynchronous operation means that a process operates independently of other processes. Asynchronous communication means each character (data byte) is placed in between the start and stop bits. The start bit is always 0 (low) and the stop bit is always 1 (high).

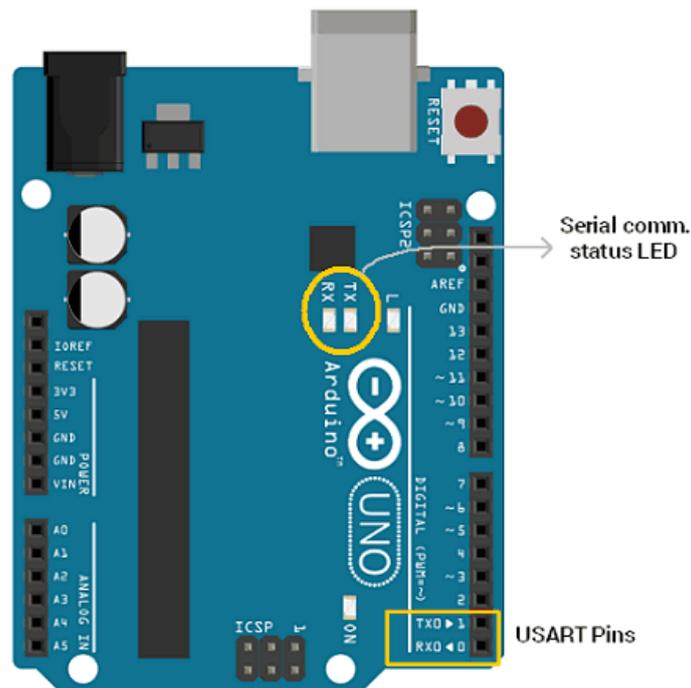


**Bit Rate & Baud Rate:** The rate of data transfer in serial data communication is stated in bps (bits per second). Another widely used terminology for bps is baud rate; means, number of changes in signal per second. Here the signal is in bits, therefore bit rate = baud rate.

## Arduino USART

Arduino Uno (ATmega328) has an in-built USART which is useful for serial communication. It can be used to communicate data between PC or various serial devices like GSM, GPS, Bluetooth, etc.

Arduino communicate with serial devices over digital pins 0 (RXD) and 1 (TXD). Whereas it communicates with PC/laptop over USB. USART pins of Arduino Uno is given in following image.



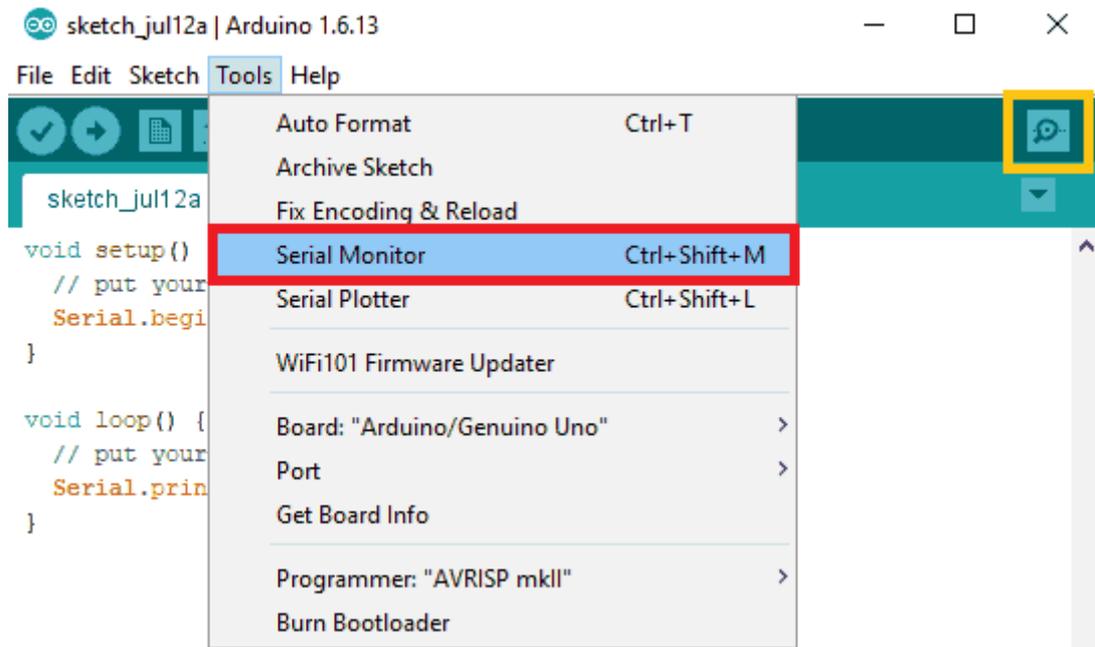
**Arduino USART Pins**

- In above image, two led indicators i.e. TX and RX are also highlighted which are status LED used to indicate serial transmission and reception.
- When PC transmits data serially to Arduino Uno over USB, RX led glow to indicate data is received. When Arduino Uno transmits data to PC over USB, TX led glow to indicate data is transmitted.
- LED status indication acts completely opposite when communicating data between serial module/devices. In case of Arduino Uno, simultaneous serial communication over USB and RX-TX pins are not allowed.
- Serial communication over RX/TX pin uses TTL logic levels. Serial communication occurs between Arduino and computer/serial devices at a same baud rate like 4800, 9600, 115200, 34800, etc.

## Arduino Serial Monitor

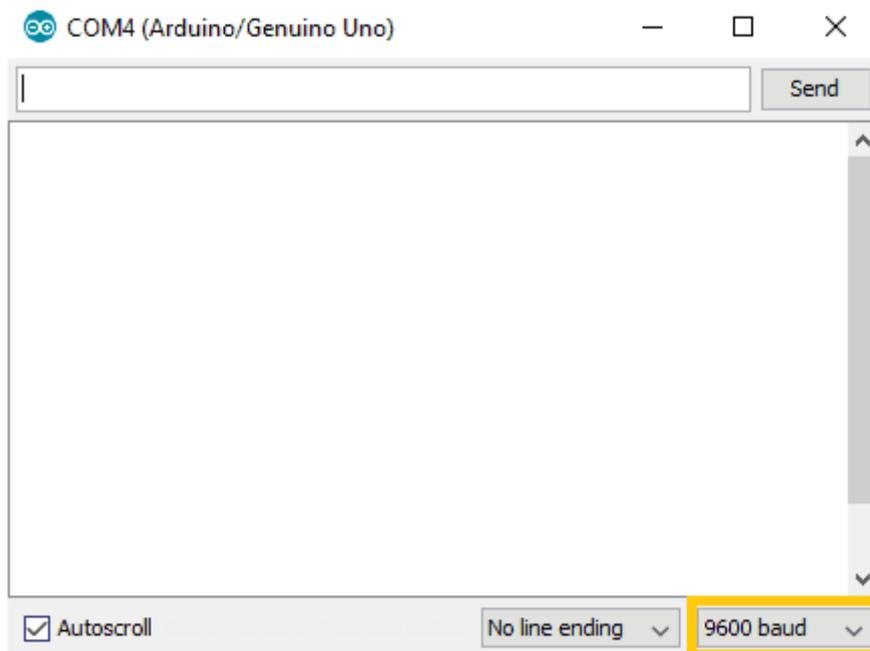
Arduino IDE has integrated serial monitor which can be used to watch serial data. After uploading program to the Arduino, open Serial Monitor to watch Serial data. You can find Serial monitor as shown below,

Go to **Tools --> Serial Monitor**



One can also access serial monitor by clicking on icon present on top-right corner of Arduino IDE shown in above image.

After selecting Serial monitor, you will get serial monitor window as shown below,

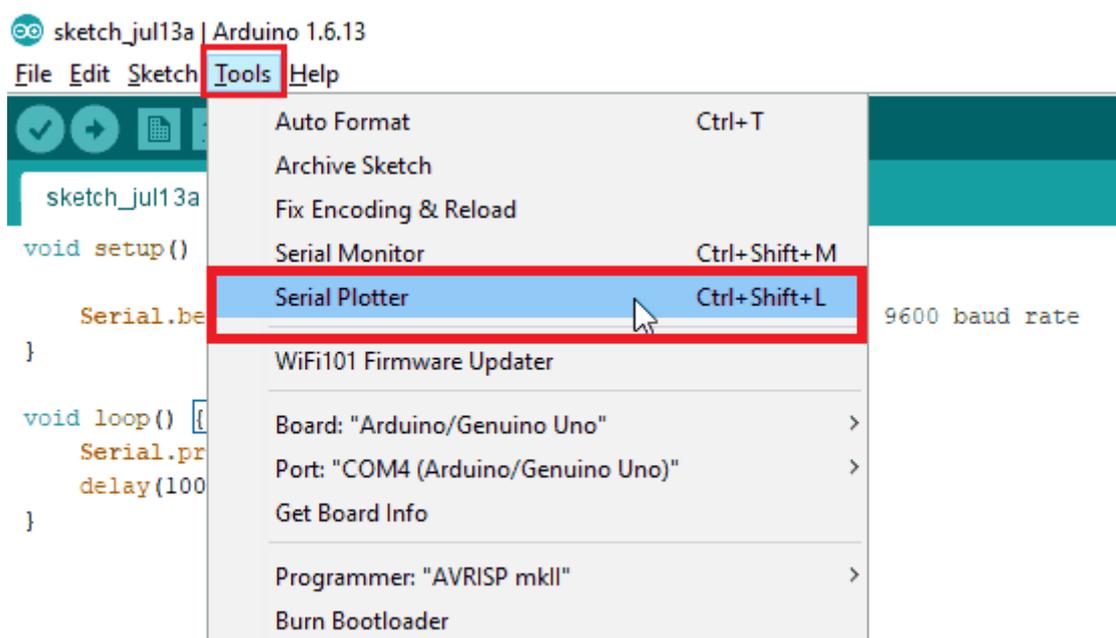


One can change baud rate by selecting one of value highlighted in above image. If baud rate of two devices mismatched, then we will get nothing or garbage value.

## Arduino Serial Plotter

Arduino Serial Plotter is a very cool feature present in Arduino ide. It visualizes the data receiving on computer serial port in the form of waveforms. It is very useful in visualizing data, debugging code and visualizing our variables as waveforms.

One can find Serial Plotter in Arduino Ide given below,



## Functions Used for Serial Communication

- **Serial.begin (baud rate)**

This function is used to initiate the serial communication with baud rate.

**e.g.** Serial.begin(9600)

- **Serial.print (data, format type (optional))**

It sends ASCII character on serial port so human can read it easily. This function converts the data to ascii character and then send (print) it.

**e.g.** Serial.print (123);

```
Serial.print ("123");  
Serial.print (12, HEX)
```

- **Serial.println (data, format type (optional))**

It sends ASCII character on serial port followed by carriage return.

- **Serial.write()**

This function writes binary data to the serial port. It returns number of byte written.

**e.g.** Serial.write (65); // it writes actual byte to serial port

- **Serial.available ()**

This function returns the number of bytes available to read.

- **Serial.read()**

This function is used to read data serially.

**e.g.** received\_data = Serial.read()

- **Serial.readString()**

This function is used to read the received string.

**e.g.** String received\_data = Serial.readString()

## Transmit Data Serially to Computer

Let's write a program to transmit data serially to the computer using Arduino at 9600 Baud rate.

### Sketch for transmitting data to the computer

```
void setup() {  
  Serial.begin (9600); //initiate serial communication with 9600 baud rate  
}  
void loop() {  
  Serial.println("hello"); //send data to computer  
  delay(1000);  
}
```

