

SERIAL TRANSMISSION & RECEPTION

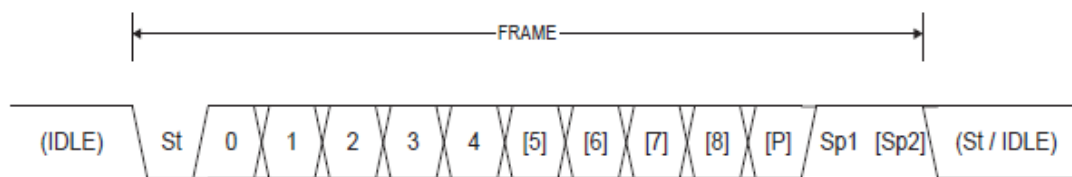
-M.Amutha Bharathi

INTRODUCTION:

Serial communication is one of the simplest ways of communication between a microcontroller and PC or vice versa. It requires only single wire for transmission of a data and another for receiving. This communication can be used for controlling the robot from a PC manually or the computer program controls it.

SERIAL PROTOCOL:

Its nothing but a sequence of data bits encapsulated between start and stop bits.



- St Start bit, always low.
- (n) Data bits (0 to 8).
- P Parity bit. Can be odd or even.
- Sp Stop bit, always high.

Example: A number ,say 5 will be transmitted as

<start bit> 0000101 <stop bit>

PORT PARAMETERS:

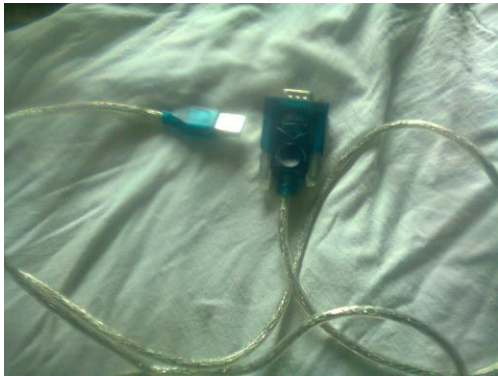
1. **Baud Rate** – No. of bits transmitted per second
Commonly used baud rate values- 1200,2400,4800,9600,19200,etc
2. **Size of the data**-8 bit or 9 bit
3. **Stop Bits**- No of stop bits you want to use.(normally 1 is sufficient)
4. **Parity bit**-Odd or Even Parity (Not necessary in short distance communication)

SERIAL PORT IN PC/LAPTOP:

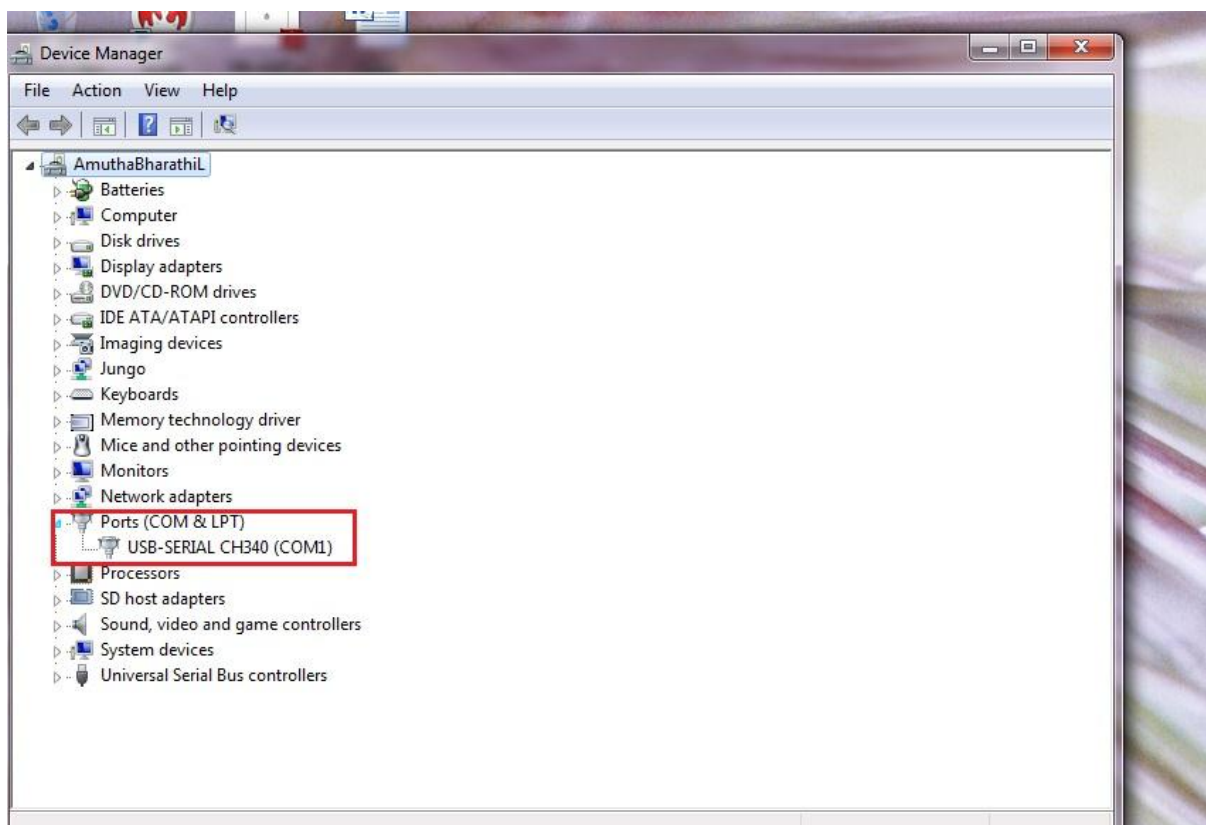
Modern PC/Laptop doesnt come with a serial port ☹️ So what do we do now?!

Simple! Emulate the serial port. A usb to Serial Converter cable will emulate a virtual COM port(serial port) in your PC. After you plugged in your usb to serial adapter to your laptop's usb port, You can install the drivers and check its availability in Device Manager.

A simple usb to serial converter looks like:



When plugged into your PC,you will find the device listed in Device Manager .

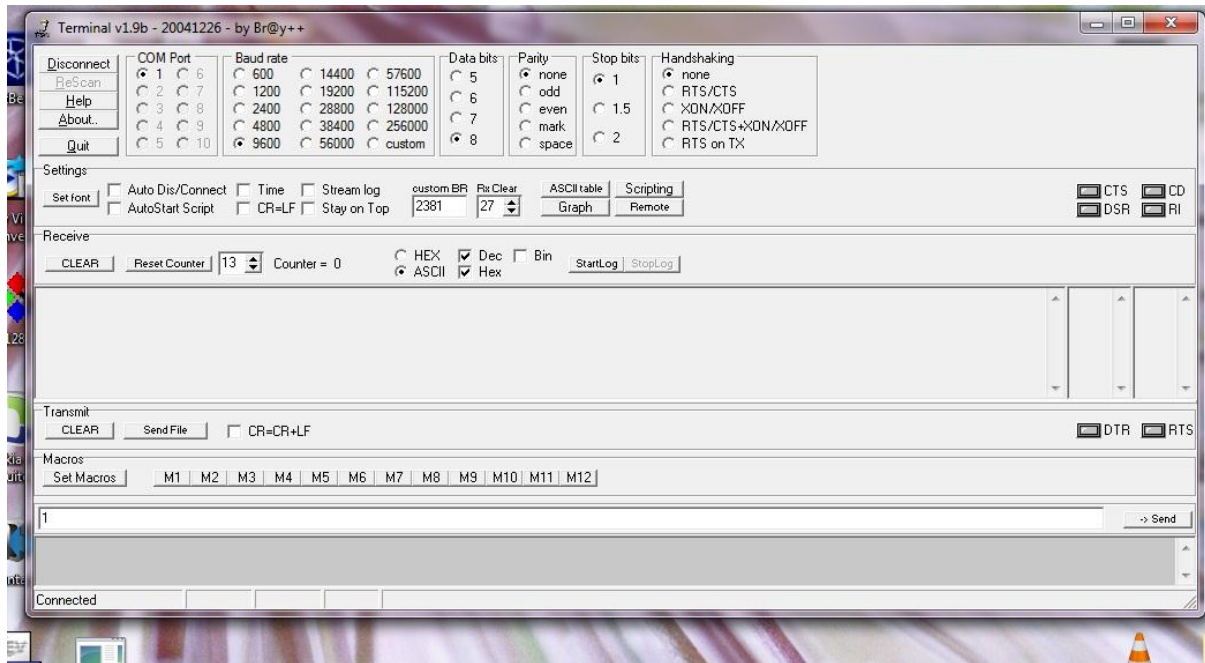


If you find the COM numbers like COM31 ,COM20, change them to a lower value by clicking on the device and going to "Advanced Settings". It is safe to set the values between COM1 and COM8.

TERMINAL:

Windows XP users must have been aware of the HyperTerminal software. Bad that it was not included in recent versions of windows.

Use Bray's terminal software for serial port operations like sending and receiving data. Google for "Bray's Terminal" to get one.



SERIAL PORT PROGRAMMING IN WINDOWS:

Serial port programming is easier in VisualC++ or DevC++ with the help of "windows.h".

A commented simple Code for Serial port operations written in DevC++ is given below.

```
/******
```

```
// PROGRAM : Serial Port Interfacing using windows calls
```

```
//COMPILER: DevC++ 4.9.9.2
```

```
/******
```

```
#include<windows.h>
```

```
#include<string.h>
```

```
#include<conio.h>
```

```
#include<iostream>
```

```

using namespace std;

// Initialize the serial port variable and parameters
HANDLE hPort = CreateFile("COM1",
GENERIC_WRITE|GENERIC_READ,0,NULL,OPEN_EXISTING,FILE_ATTRIBUTE_NORMAL,NULL);

DCB dcb; // Create a DCB struct variable

/*****Function for Sending Data*****/

bool writebyte(char* data)
{
    DWORD byteswritten;
    if (!GetCommState(hPort,&dcb))
    {
        cout<<"\nSerial port cant b opened\n";
        return false;
    }

    dcb.BaudRate = CBR_9600; //9600 Baud
    dcb.ByteSize = 8;      //8 data bits
    dcb.Parity = NOPARITY; //no parity
    dcb.StopBits = ONESTOPBIT; //1 stop

    if (!SetCommState(hPort,&dcb)) //If Com port cannot be configured accordingly
        return false;

    bool retVal = WriteFile(hPort,data,1,&byteswritten,NULL); //Write the data to be sent to Serial port
    return retVal; // return true if the data is written
}

```

```

int ReadByte()
{
int Val;
BYTE Byte;
DWORD dwBytesTransferred;
DWORD dwCommModemStatus;
if (!GetCommState(hPort,&dcb))
return 0;
SetCommMask (hPort, EV_RXCHAR | EV_ERR); //receive character event
WaitCommEvent (hPort, &dwCommModemStatus, 0); //wait for character
if (dwCommModemStatus & EV_RXCHAR)
ReadFile (hPort, &Byte, 1, &dwBytesTransferred, 0);
Val = Byte;
return Val;
}

```

```

int main()
{
cout<<"\n\n  /***** SERIAL PORT PROGRAM *****/\n\n";
cout<<"This is a loopback test..The Tx & Rx pins of serial port are shorted here...\n\n";
char data=2; // Enter any data to b sent
if(writebyte(&data)) // if the function returns non-zero value
cout<<"    Data Sent..  "<<(int)data<<"\n";

cout<<"    Data Received.. "<<ReadByte();

CloseHandle(hPort); // Close the Serial port handler

```

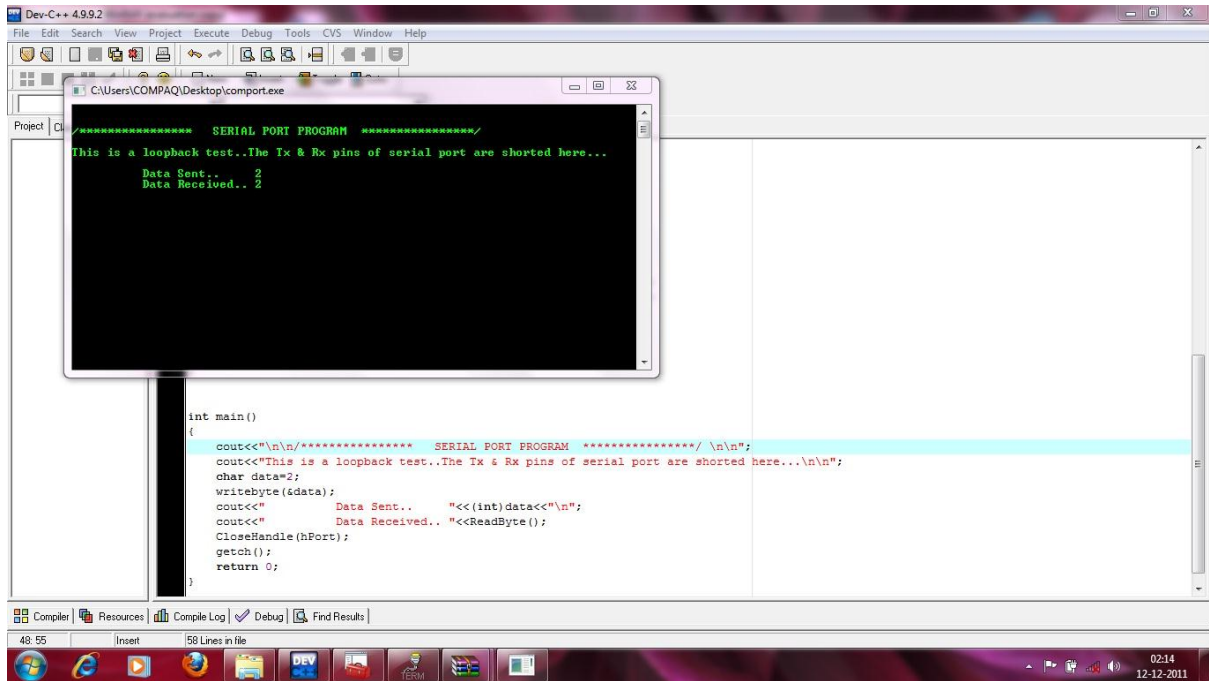
```

getch();

return 0;

}

```



The transmit and receive pins(2&3) are shorted together with a jumper for this test. So, whatever data sent will be looped back immediately.



Now interfacing from PC to microcontroller will be simpler and of course Level-Shifters have to be used. And making the communication wireless is yet another easier topic to search in web! ☺

Any queries?

Feel free to contact us @ dinerdash@pragyan.org or bharathinitt@gmail.com

