



## A TUTORIAL ON USART AND SERIAL COMMUNICATION

Communicating your computer and microcontroller consists of two parts, firstly, the part where the code is to be written and burnt in the microcontroller and the secondly, the part where the code is to be written and included in your project in your computer.

## USART

USART stands for Universal Synchronous Asynchronous Receiver Transmitter, it is a standard and simple protocol for serial data transfer.

## MAKING CONNECTIONS:

The following connection is to be made for serial communication, which is connecting the development board with the computer by a USB to serial port which looks like the following.



The serial part of the port is connected to the development board and the USB part is connected to the computer.

The connection is shown below.



### CODING:

Codes are required to be written and burnt in the microcontroller.

Create a new project and add the following files to your project:

1.usart.h

2.usart.c

These files are provided for download on our website, in the ZIP file named “serial communication”

Adding these files to your project is very important, without this USART won't work. These files are important in the functioning of serial communication.

After adding those files to your project, write down your code, a sample code is given here which does the following job:

If '1' is received, it switches all LED on

If '0' is received it switches off all LED

If 'a' is received it sends back "Hello World" string to the computer

If 'b' is received it sends back "Good bye" to the computer.

The code is given below:

Some important commands used are:

**USARTInit(UBRRVALUE);** [for initializing usart. The number to be entered in parenthesis is the UBRRVALUE. which is 103 for 9600 bits per second, 68 for 14400, and 51 for 19200 Bits/second]

**UDataAvailable();** [To get number of data bytes waiting in queue to be called]

**UWriteData(char data);** [To send data to PC]

**UWriteString(char \*str);** [To send a string to PC]

**UReadData(char data);** [To read Bytes from a queue call]

### **THE SAMPLE PROGRAM:**

```
#include <avr/io.h>
#include "USART.h"
void main()
{
    char cmd;    //Byte received from USART
    //Initialise USART with 19200Kbps
    USARTInit(51);
    //Initialize LED port as output
    DDRC|=0b00001111;
    while(1)
    {
```

```

//wait till we have any data
while(UDataAvailable()==0);
//Read the data
cmd=UReadData();
//Process the command
switch(cmd)
{
    case '1':
        //Switch on all LEDs
        PORTC=0b00000000;
        UWriteString(" <All LEDs on>");
        break;
    case '0':
        //Switch off all LEDs
        PORTC=0b00001111;
        UWriteString(" <All LEDs off>");
        break;
    case 'a':
    case 'A':
        //Write back to serial port
        UWriteString(" <Hello world>");
        break;
    case 'b':
    case 'B':
        //Write back to serial port
        UWriteString(" <Good bye>");
}
}
}

```

After Writing the above code compile the project and burn it in your microcontroller. Precompiled hex files ready to burn will be in the “precompiled-hex-file” folder. Burn this to your microcontroller.

## The part to be included in your project in the computer.

The following files are to be included in your project in your computer along with the main program body written in the project.

The files are :

1. tserial.h
2. tserial.cpp
3. bot\_control.h

These files are to be copied into the project folder and then added in the project. These are very necessary for the functioning of serial communication because they contain all the settings needed for serial communication to work.

These files are available on our site for download along with usart.h and usart.c in the zip file “serial communication”.

Now a sample program is written below. This is the main code of your project. It is a sample program for sending one character through serial communication.

```
#include <stdio.h>
#include <iostream>
#include <conio.h>
#include <stdlib.h>
#include "tserial.h"
#include "bot_control.h"
```

```
serial comm;
```

```
int main() {
```

```
char data;
```

```
printf("enter character to be sent");
scanf("%c",&data);

comm.startDevice("COM2", 9600)

/* "COM 2" refers to the com port in which the USB to SERIAL
port is attached. It is shown by right clicking on my
computer, then going to properties and then device manager.

9600 is the baudrate in bits per second. */

comm.sendData(data);

comm.stopDevice();
    return 0;
}
```

To understand these functions sendData(), startDevice(), stopDevice() go through the file bot\_control.h thoroughly.

The above procedure is to be followed completely in order to send any information (a byte in this case) from your computer to your microcontroller by serial communication.