Data Serialization in C

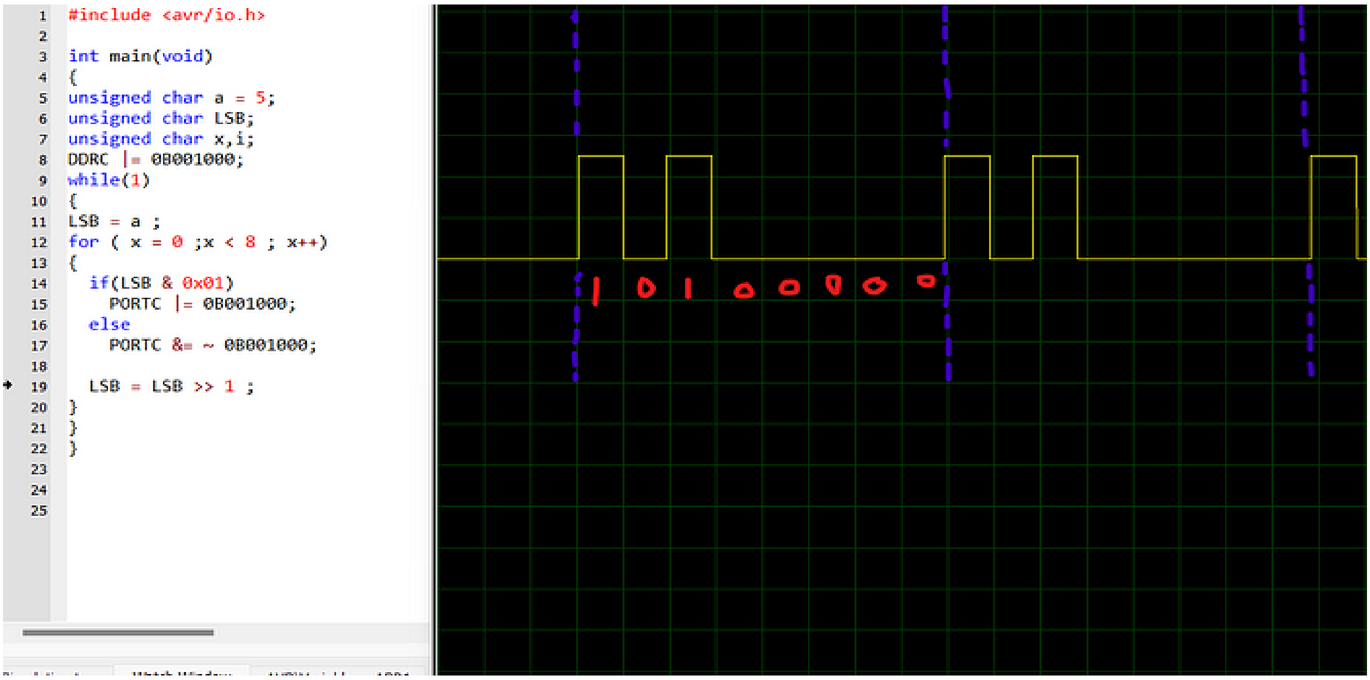
Serializing data is way of sending a byte of data one bit at a time through a single pin of microcontroller.

There are two ways to transfer a byte of data serially :

1. Using the serial port. In using serial port, the programmer has very limited control over the sequence of the data transfer

2. The second method of serializing data is to transfer data one bit at a time and control the sequence of data and spaces between them. In many devices such as LCD, EEPROM, ADC ; serial data communication is common. Although there are standards of serial communication such as I2C, SPI, CAN but not all devices supports such standards.

For this reason, lets see the **data serialization** using the C language, How it is done in C and respective simulation.



```
1  #include <avr/io.h>
2
3  int main(void)
4  {
5  unsigned char a = 5;
6  unsigned char LSB;
7  unsigned char x,i;
8  DDRC |= 0B001000;
9  while(1)
10 {
11 LSB = a ;
12 for ( x = 0 ;x < 8 ; x++)
13 {
14   if(LSB & 0x01)
15     PORTC |= 0B001000;
16   else
17     PORTC &= ~ 0B001000;
18
19   LSB = LSB >> 1 ;
20 }
21 }
22 }
23
24
25
```

Explanation of code : Above code send out the value of 5 serially one bit a time via PORTC on

pin no 3 on ATMega328p, Here LSB goes first.

Line no 1 includes the library “ #include<avr/io.h>”, it enables users to include Input/Output functions and also serves the access to the internal registers of AVR microcontroller.

Line no 3 includes the main of the program in which logic of the code resides. It serves as the entry point to the code.

Line no. 5 declared the data which we want to send serially , here in this case it is 5.

It is stored in a 1 byte long data type char and to have full range of this data type i.e. 0- 255, we declared it as a unsigned char a = 5;

Another unsigned char variable is declared on line no 6 and line no 7 i.e. LSB and x respectively.

We will see its use further,

After that DDRC |= 0B001000; is used to set pin no 3 on PORTC as output by setting bit 3 on data direction register pin DDRC.

In while loop of our code we write our main logic. on line no 11, we assigned the value of *a* to LSB variable.

For loop on the next line is used to iterate x upto 8 times so that all 8 bits from variable *a* can be seen on PORTC.3 pin serially.

Inside the for loop, we have if condition which gets true only if the last bit on variable is 1 and then it sets 1 on PORTC.3 pin otherwise it sets 0 on PORTC.3

Initially on 1st iteration we have LSB = 00000101 and it gets & with 00000001 and result is true that is why the PORTC.3 pin is set and 1 appear across it. As you can see on simulation high bit is appears first. After that LSB is left shifted and LSB becomes 00000010.

Now on 2nd iteration, LSB & with 00000001 and which results in false that is why else part executed and PORTC.3 pin set to low. Again LSB is shifted and LSB becomes 00000001.

Now on 3rd iteration, LSB & with 00000001 and which results in true that is why if part executed and PORTC.3 pin set to high. Again LSB is shifted and LSB becomes 00000000.

Now on 4th to 8th iteration, else part gets executed which sets PORTC.3 pin to zero.

If we measure the voltage across the PORTC.3 pin throughout this for loop iterations, we get voltage waveform as shown in above figure which represents the 8 bit representation of number 5 from LSB i.e. 10100000.