Software Serial Port

Software serial port feature allows Arduino to transmit and receive serial data using any two digital pins except pin 0 (RX) and pin 1 (TX) of Arduino Uno board.

If you want to connect Arduino Uno with a Bluetooth and Wi-Fi module then we can connect one with a hardware serial port and other through a software serial port.

In order to create software serial port, SoftwareSerial.h header file is included in the Arduino code using the following statement:

```
#include <SoftwareSerial.h>
```

Next step in software serial port programming is to create an object of type SoftwareSerial and this can be done using the following statement:

```
SoftwareSerial mySerial (10,11);
```

Here pin 10 signifies that it will be used for serial data reception and pin 11 will be used for serial data transmission

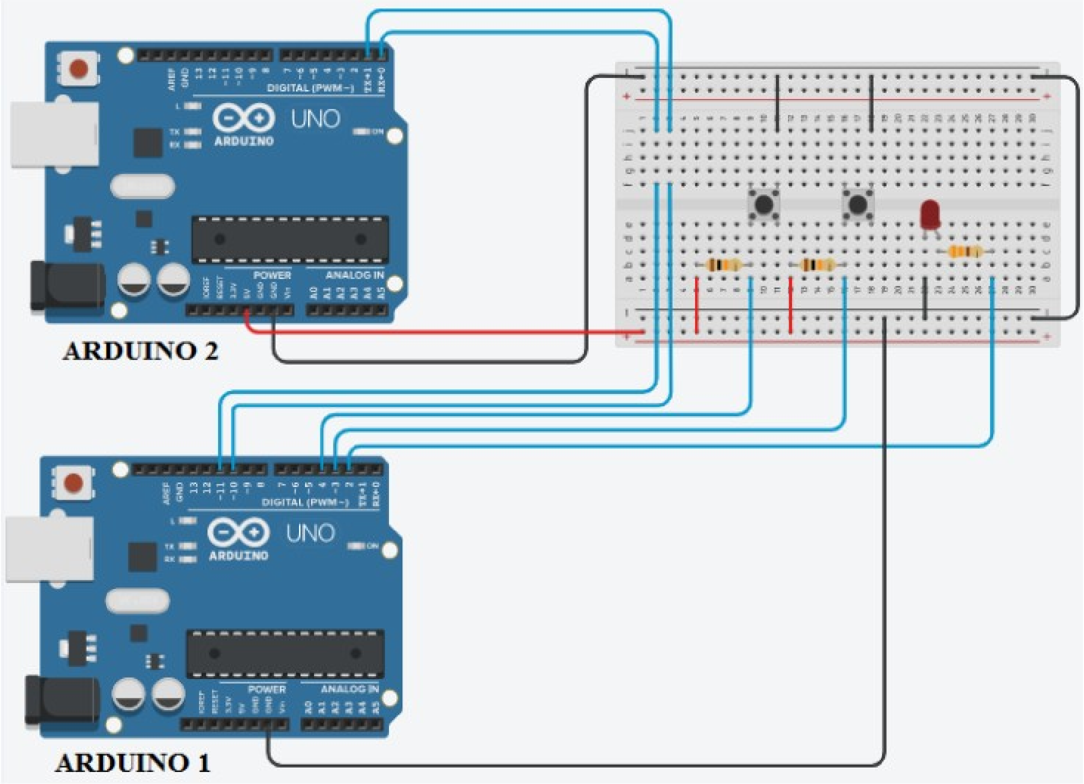
Setting Baud Rate

- In order to initialize the software serial port at a particular baud rate value, the following statement can be used:

  ```
  mySerial.begin (9600);
  ```

  This statement will initialize the software serial port at a baud rate value of 9600 bps.
- You can use all the functions that are valid for hardware serial data port except one small change. You must use the name of the serial object that you have created for the software serial port that you have created.

Programming Example



Serial data communication
Between two Arduino Uno
boards

Arduino 1:
LED cnnected to pin 2
Push button 1 connected to pin 3
Push button 2 connected to pin 4

Code For Arduino 1

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10,11);
// software serial port 10-rx, 11-tx
char temp; // variable to store serial data byte received
void setup()
{
mySerial.begin(9600);
//initialize software serial port at 9600 bps
pinMode(3,INPUT); // pin 3 configured as input
pinMode(4,INPUT); // pin 4 configured as input
pinMode(2,OUTPUT); // pin 2 configured as output
digitalWrite(2,LOW); // initial state of pin 2 is LOW
digitalWrite(3,HIGH); // make initial state of pin 3 HIGH
digitalWrite(4,HIGH); // make initial state of pin 4 HIGH
}
```

```
void loop()
{
if(digitalRead(3)==LOW) // check for push button press
{
mySerial.print('A'); // send 'A' on software serial port
while(digitalRead(3)==LOW);
// wait for push button to be released
}
}
if(digitalRead(4)==LOW)
{
mySerial.print('B'); // send 'B' on software serial port
while(digitalRead(4)==LOW);
// wait for push button to be released
}
```

Prog Contd.

```
if(mySerial.available()>0)
// check for returned serial data
{
temp = mySerial.read();
// read data byte and store in temp
if(temp=='A') // check if temp is 'A'
{
digitalWrite(2,HIGH);
// make state of pin 2 HIGH
}
if(temp=='B') // check if temp is 'B'
{
digitalWrite(2,LOW);
// make state of pin 2 LOW
}
}
```

Code For Arduino 2

```
char x;
// variable defined for storing the serial data byte
// data stored in variable x will be sent back to Arduino 1 serially
void setup()
{
Serial.begin(9600); // initialize serial communication at 9600 bps
}
```

```
void loop()
{
while(!Serial.available());
// wait for serial data to be received
if(Serial.available()>0) // if serial data is received
{
x = Serial.read();
// read the serial data byte and store in x
Serial.print(x);
// send data byte stored in variable x serially
}
}
```