

## Serial.h library

Arduino UNO has a UART on D0(RX) and D1(TX) with a 64 byte buffer used by USB. RX is the data received by the hardware USB port and TX is the data to send. When the USB is disconnected they can be used by other devices, e.g. a Bluetooth module.

```
#define BAUD 9600
#define CONFIG SERIAL_8N1

int v;
int z;
int b;

Serial.begin(BAUD, CONFIG); // enable TX/RX, set BAUD rate and configuration
v = Serial.available(); // no. of bytes (chars) available in buffer (max 64)
v = Serial.read(); read a byte(char), returns -1 if no data in buffer
v = Serial.peek(); next char in buffer, not removed, -1 if none
Serial.flush(); // wait for TX of outgoing data
z = Serial.print(VALUE, FORMAT); // format not required
z = Serial.println(VALUE, FORMAT); // adds CRLF
    // VALUE is any data type (byte, char, int long, float, string)
    // FORMAT is BIN, OCT, DEC for ints, N decimal places for floats
    // flash memory strings can be VALUE with F("Hello World")
    // "\n" = CRLF, "\t" in strings = TAB
b = Serial.write(VAL|STR|BUF, LEN); // return no. bytes/string sent
    // write a VAL byte|char, STR string (of bytes), or BUFfer of LENGth
Serial.end(); // disables TX RX for serial comms
```

## SoftwareSerial.h library

SoftwareSerial.h allows serial comms on other pins, thus no conflict with USB pins 1 & 2. It can also create more than one port.

```
#include <SoftwareSerial.h>

// TX is data sent out from the Arduino, RX is data received in,
// for example connect ATX to BRX and ARX to BTX (B = Bluetooth adapter)
#define RXPIN 2
#define TXPIN 3
#define BAUD 9600

boolean b;
char c;
byte b;
```

```

int z;

SoftwareSerial mySerial(RXPIN, TXPIN);
  // create new SoftwareSerial object.
mySerial.begin(BAUD); // begin TX/RX on defined pins at BAUD rate
v = mySerial.available(); // no. of bytes (chars) available in buffer (max 64)
mySerial.listen(); // enables listening on port
b = mySerial.isListening(); // true if port is listening
b = mySerial.overflow(); // true if port has overflowed 64 byte buffer
c = mySerial.read(); // read a char, returns -1 if no data in buffer
c = mySerial.peek(); // next char in buffer, not removed, -1 if none
z = mySerial.print(...); // same as print in Serial above
z = mySerial.println(...); // same as println in Serial above
b = mySerial.write(...); // same as write in Serial above

```

## Example

USB is hardware port to Arduino, Bluetooth module is connected to pins 2 & 3, program reflects Bluetooth data to USB and USB data to Bluetooth. Cross connect Arduino pins RX and TX to Bluetooth pins TX & RX.

```

// library allows pins other than 0 (USB RX) & 1 (USB TX) to be used
#include <SoftwareSerial.h>

// RXPIN (data received by USB, to be sent on BT TX) and
// TXPIN (data received by RX BT, to be sent on USB)
// at BAUD rate
#define RXPIN 2
#define TXPIN 3
#define BAUD 9600

SoftwareSerial mySS(RX, TX); // create serial object

void setup()
{
  Serial.begin(BAUD); // start USB serial connection
  mySS.begin(BAUD); // start mySS serial communication
}

void loop()
{
  int x, y;

  if(mySS.available()) // data in mySS?
  {
    x = mySS.read(); // get byte from mySS - Bluetooth
    Serial.write(x); // write to USB
  }

  if(Serial.available()) // data in USB?
  {
    y = Serial.read(); // get byte from USB
    mySS.write(y); // write to Bluetooth
  }
}

```