NewSoftSerial

As you've seen, serial is an incredibly useful way to transfer information between computers. In order to transmit and receive serial data, you need, logically enough, a serial port. This is provided by a bit of hardware called a UART (Universal Asynchronous Receiver / Transmitter) that lives on the processor chip. The UART does the hard work of generating and decoding the signals used to send data across the RX and TX lines, so all you (as a programmer) need to do, is feed it characters that you want sent, and ask it for any characters it's received while you were off doing something else. Easy!

The smaller Arduinos based on the ATmega328 (the Uno, Duemilanove, Pro and Pro Mini) have one UART, giving you one serial port. The Arduino Mega, based on the ATmega2560, has four (!) UARTs, giving you four serial ports.

Typically you'll connect one device to each serial port. On all Arduinos, the lowest numbered (or sole) serial port is connected to the main computer through the USB cable, but you could also attach it to anything else you want, such as a GPS receiver, XBee, etc.

This starts becoming a problem if you want to connect your Arduino to more than one serial device. What if you want to connect your (sole) serial port to a GPS unit, but also look at debugging information back on your PC? Or forward GPS data through a radio link, or display it on a serial LCD? This isn't a big problem on the Arduino Mega, as it has four serial ports. But on the smaller Arduinos, which have only one serial port, it can be a significant restriction.

But it turns out that what the UART does with hardware, can also be done in software. Several libraries are available that emulate hardware serial ports in software, allowing you to create additional "soft" serial ports if needed.

Arduino 0022 comes with a library called SoftwareSerial (http://arduino.cc/en/Reference/SoftwareSerial) that provides this functionality. However this library has significant limitations (9600-baud only, won't receive when it's not active, etc.). Mikal Hart has written a greatly-improved library called NewSoftSerial, which solves most of these problems. For now you'll need to download and install the library yourself, but it is expected that NewSoftSerial will replace SoftwareSerial in the official Arduino IDE at some point in the future.

XBee Wireless, New SoftSerial Library

SparkFun Electronics Summer Semester

To install the NewSoftSerial library:

1. If it doesn't already exist, create a "libraries" folder in your Arduino sketches folder.
2. Download the latest library from http://arduiniana.org/libraries/newsoftserial/
3. Open the zip file
4. Drag the NewSoftSerial folder into the "libraries" folder you just made.
5. Start or restart the Arduino IDE. NewSoftSerial should now be listed in the "Sketch / Import library" menu.

You use NewSoftSerial very much like the hardware serial port. The only extra steps are to #include the library and set up the new port at the beginning of your sketch:

```
#include <NewSoftSerial.h>

const int RXpin = 2;
const int TXpin = 3;
NewSoftSerial myPort(RXpin, TXpin);
```

You can of course name the port anything you want (it's often useful to name the port after its function, such as "GPS" or "display"). For the rest of your program, you use NewSoftSerial exactly like a normal serial port:

```
void setup()
{
 myPort.begin(9600);
 myPort.println("hello, world!");
}

void loop()
{
 char c;

 // check if there are received characters
 if (myPort.available() > 0)
 {
      c = myPort.read();
      // do something with c here
 }
}
```

XBee Wireless, New SoftSerial Library

SparkFun Electronics Summer Semester

There's some fine print you should be aware of. Generating (and receiving) serial data in software is extremely processor-intensive, since it needs to send (and receive) every bit in every character at *exactly* the right time, thousands of times per second. The problem becomes worse at high serial speeds, and the more soft serial ports you create. Once the processor can't keep up with all those bits, you'll start losing data. Here are some tips to help you get the most out of soft serial ports:

- When choosing which ports to use, save the hardware port for the fastest / heaviest connection you have. Use soft serial ports for low speed / lightly used connections if possible. (However, if you're using the USB serial link back to your PC, you'll have to use the hardware port for that).
- Keep your soft serial rates as low as possible, and between 9600 and 57600 baud. Other baud rates (lower or higher) may not work reliably.
- Soft serial ports may interfere with other interrupt-driven libraries, such as Servo. Google for advice in these cases.
- If you create more than one soft serial port, only one of them (the one you last used) will be able to receive data at a time. This may not be a huge problem, as you can often structure your program to access them sequentially as required.
- The NewSoftSerial library is most useful on the smaller Arduinos that only have one serial port. However, it can be of use on the Mega as well. Check the NewSoftSerial webpage (http://arduiniana.org/libraries/newsoftserial/) for the latest news as to compatibility with the Mega. (As of this writing, version 10C does not fully support the Mega, however beta version 11 does on certain pins).