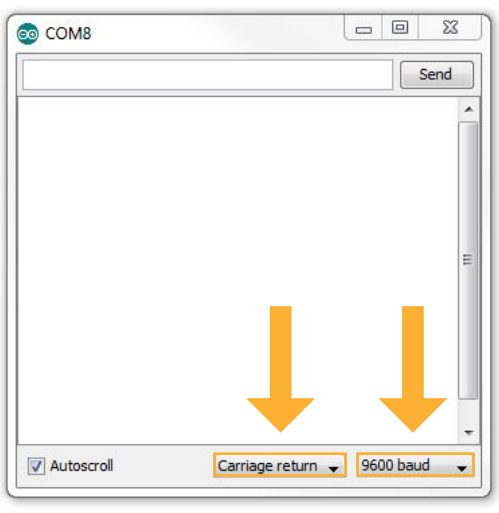
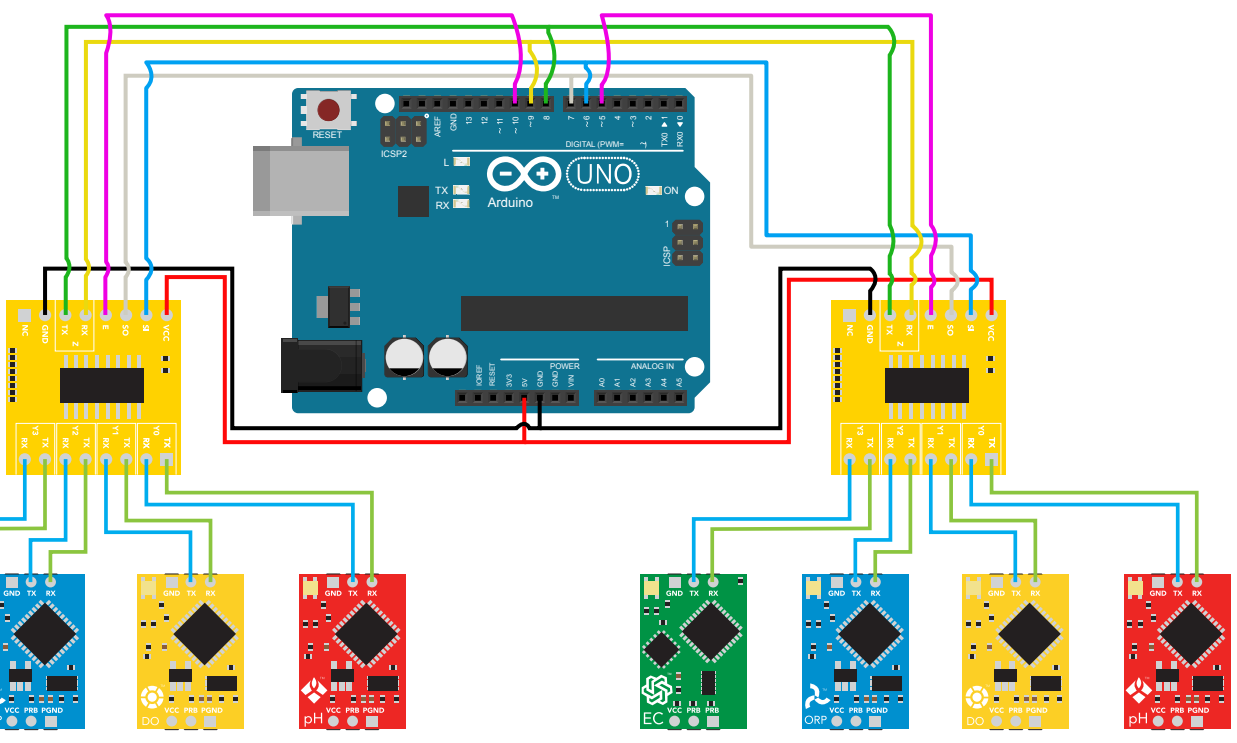
Serial Port Expander sample code




```
//This sample code was written on an Arduino UNO.
//It will allow you to control up to 8 Atlas Scientific devices through 1 soft serial RX/TX line.
//To open a channel (marked on the board as Y0 to Y7) send the number of the channel, a colon and the command ending with
// a carriage return.
```




```
//0:r<CR>
//1:i<CR>
//4:c<CR>
//7:r<CR>

//To open a channel and not send a command just send the channel number followed by a colon.

//1:<CR>
//6:<CR>

//This code uses the Altsoft softserial library. The library file can be downloaded here:
//http://www.pjrc.com/teensy/td_libs_AltSoftSerial.html
//This softserial library Automatically sets TX as pin 9 and RX as pin 8.
```

```
#include <AltSoftSerial.h>              //Include the software serial library
AltSoftSerial altSerial;                //Name the software serial library altSerial (this cannot be omitted)

int s0 = 7;                             //Arduino pin 7 to control pin S0
int s1 = 6;                             //Arduino pin 6 to control pin S1
int enable_1= 5;                        //Arduino pin 5 to  control pin E on board 1
int enable_2=10;                        //Arduino pin 10 to control pin E on board 2

char computerdata[20];                  //A 20 byte character array to hold incoming data from a pc/mac/other
char sensordata[30];                    //A 30 byte character array to hold incoming data from the sensors
byte computer_bytes_received=0;         //We need to know how many characters bytes have been received
byte sensor_bytes_received=0;           //We need to know how many characters bytes have been received

char *channel;                          //Char pointer used in string parsing
char *cmd;                              //Char pointer used in string parsing


void setup() {
  pinMode(s1, OUTPUT);                  //Set the digital pin as output.
  pinMode(s0, OUTPUT);                  //Set the digital pin as output.
  pinMode(enable_1, OUTPUT);            //Set the digital pin as output.
  pinMode(enable_2, OUTPUT);            //Set the digital pin as output.
  Serial.begin(9600);                   //Set the hardware serial port to 9600
  altSerial.begin(9600);                //Set the soft serial port to 9600
 }
```

```
void serialEvent(){                                                 //This interrupt will trigger when the data coming from
        computer_bytes_received=Serial.readBytesUntil(13,computerdata,20);  //the serial monitor(pc/mac/other) is received
        computerdata[computer_bytes_received]=0;                    //We read the data sent from the serial monitor
        }                                                           //(pc/mac/other) until we see a <CR>. We also count
                                                                    //how many characters have been received
                                                                    //We add a 0 to the spot in the array just after the last
                                                                    //character we received.. This will stop us from
                                                                    //transmitting incorrect data that may have been left
                                                                    //in the buffer
```

```
void loop(){

    if(computer_bytes_received!=0){                                 //If computer_bytes_received does not equal zero
      channel=strtok(computerdata, ":");                            //Let's parse the string at each colon
      cmd=strtok(NULL, ":");                                        //Let's parse the string at each colon
      open_channel();                                               //Call the function "open_channel" to open the correct
      altSerial.print(cmd);                                         //data path
      altSerial.print("\r");                                        //Send the command from the computer to the
      computer_bytes_received=0;                                    //Atlas Scientific device using the softserial port
      }                                                             //After we send the command we send a carriage return <CR>
                                                                    //Reset the var computer_bytes_received to equal 0


if(altSerial.available() > 0){                                      //If data has been transmitted from an Atlas Scientific device
     sensor_bytes_received=altSerial.readBytesUntil(13,sensordata,30);  //we read the data sent from the Atlas Scientific device until
     sensordata[sensor_bytes_received]=0;                           //we see a <CR>. We also count how many character have
     Serial.println(sensordata);                                    //been received
   }                                                                //we add a 0 to the spot in the array just after the last
}                                                                   //character we received. This will stop us from transmitting
                                                                    //incorrect data that may have been left in the buffer
                                                                    //let's transmit the data received from the Atlas Scientific
                                                                    //device to the serial monitor
```

```
void open_channel(){                    //This function controls what UART port is opened.

   switch (*channel) {                  //Looking to see what channel to open

    case '0':                           //If channel==0 then we open channel 0
       digitalWrite(enable_1,LOW);      //Setting enable_1 to low activates carrier board #1
       digitalWrite(enable_2,HIGH);     //Setting enable_2 to high deactivates carrier board #2
       digitalWrite(s0, LOW);           //S0 and S1 control what channel opens
       digitalWrite(s1, LOW);           //S0 and S1 control what channel opens
     break;                             //Exit switch case

      case '1':
       digitalWrite(enable_1,LOW);
       digitalWrite(enable_2,HIGH);
       digitalWrite(s0, HIGH);
       digitalWrite(s1, LOW);
     break;

     case '2':
       digitalWrite(enable_1,LOW);
       digitalWrite(enable_2,HIGH);
       digitalWrite(s0, LOW);
       digitalWrite(s1, HIGH);
     break;

     case '3':
       digitalWrite(enable_1,LOW);
       digitalWrite(enable_2,HIGH);
       digitalWrite(s0, HIGH);
       digitalWrite(s1, HIGH);
     break;

      case '4':
       digitalWrite(enable_1,HIGH);
       digitalWrite(enable_2,LOW);
       digitalWrite(s0, LOW);
       digitalWrite(s1, LOW);
     break;

      case '5':
       digitalWrite(enable_1,HIGH);
       digitalWrite(enable_2,LOW);
       digitalWrite(s0, HIGH);
       digitalWrite(s1, LOW);
     break;

      case '6':
       digitalWrite(enable_1,HIGH);
       digitalWrite(enable_2,LOW);
       digitalWrite(s0, LOW);
       digitalWrite(s1, HIGH);
     break;

      case '7':
       digitalWrite(enable_1,HIGH);
       digitalWrite(enable_2,LOW);
       digitalWrite(s0, HIGH);
       digitalWrite(s1, HIGH);
     break;
    }
}
```

Click here to download the *.ino file