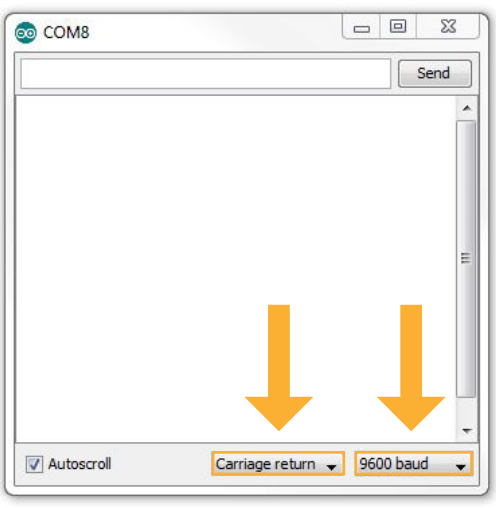
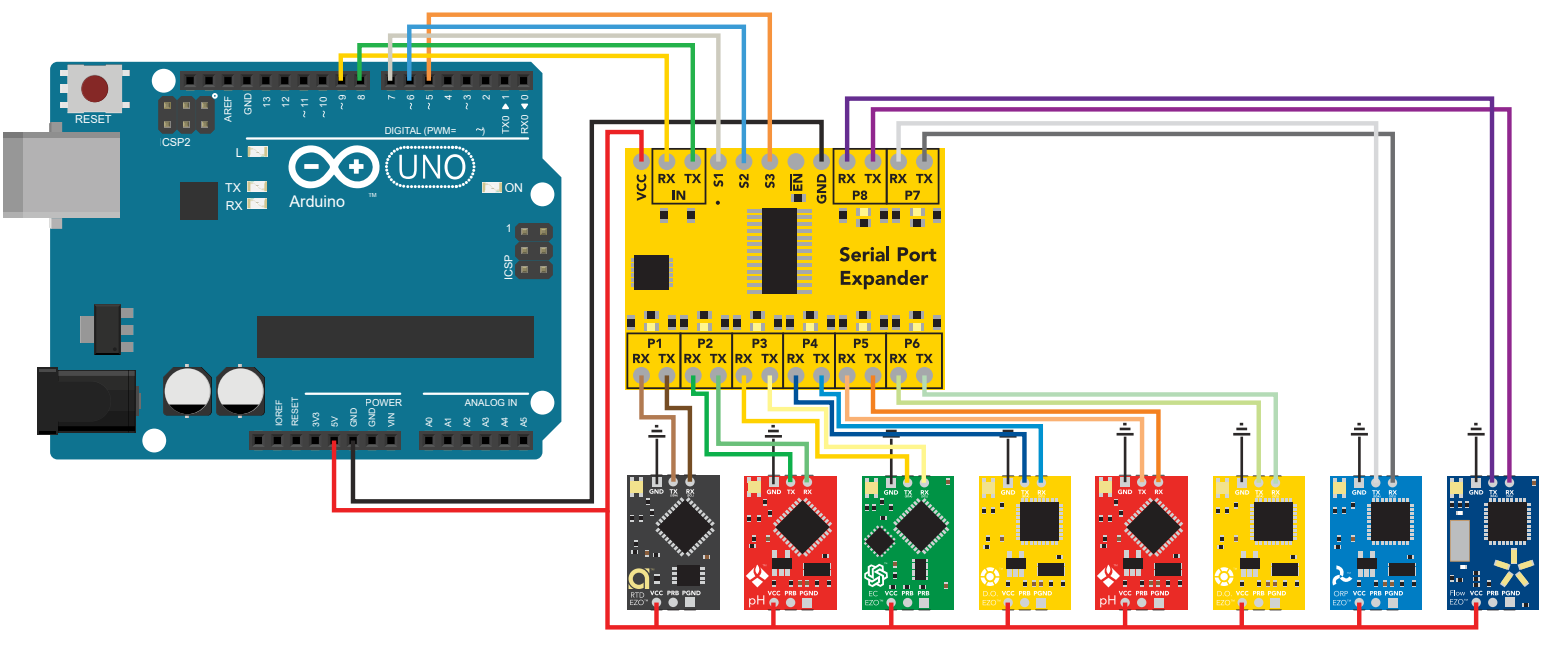
8:1 Serial Port Expander sample code

Revised 11/03/16




```
//This code was written in the Arduino 1.6.9 IDE
//An Arduino UNO was used to test this code.
//This code was written to be easy to understand. Code efficiency was not considered.
//Modify this code as you see fit.
//This code 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 P1 to P8) send the number of the channel followed by a colon and the command
//(if any) that you want to send. End the string with a carriage return.
//1:r<CR>
//2:i<CR>
//3:c<CR>
//4:r<CR>

//To open a channel and not send a command just send the channel number followed by a colon.
//1:<CR>
//3:<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)

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

int s1 = 5;                                 //Arduino pin 5 to control pin S1
int s2 = 6;                                 //Arduino pin 6 to control pin S2
int s3 = 7;                                 //Arduino pin 7 to control pin S3
int port = 0;                               //what port to open

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
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(s2, OUTPUT);                      //Set the digital pin as output
  pinMode(s3, 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
                                                                        //the serial monitor(pc/mac/other) is received
  computer_bytes_received = Serial.readBytesUntil(13, computerdata, 20); //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
  computerdata[computer_bytes_received] = 0;                            //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 data path


    if (cmd != 0) {                             //If a command has been sent
      altSerial.print(cmd);                     //Send the command from the computer to the Atlas Scientific device using the softserial port
      altSerial.print("\r");                    //After we send the command we send a carriage return <CR>
    }
    computer_bytes_received = 0;                //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
                                                                            //we see a <CR>. We also count how many character have
                                                                            //been received
    sensordata[sensor_bytes_received] = 0;                                  //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
    Serial.println(sensordata);                                             //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

  port = atoi(channel);                     //Convert the ASCII char value of the port to be opened into an int
  if (port < 1 || port > 8)port = 1;        //If the value of the port is within range (1-8) then open that port. If it's not in range set it port 1
  port -= 1;                                //So, this device knows its ports as 0-1 but we have them labeled 1-8 by subtracting one from
                                            //the port to be opened we correct for this.


  digitalWrite(s1, bitRead(port, 0));       //Here we have two commands combined into one
  digitalWrite(s2, bitRead(port, 1));       //The digitalWrite command sets a pin to 1/0 (high or low)
  digitalWrite(s3, bitRead(port, 2));       //The bitRead command tells us what the bit value is for a specific bit location of a number
  delay(2);                                 //this is needed to make sure the channel switching event has completed
  return;                                   //go back


}
```

Click here to download the *.ino file

Copyright © Atlas Scientific LLC All Rights Reserved