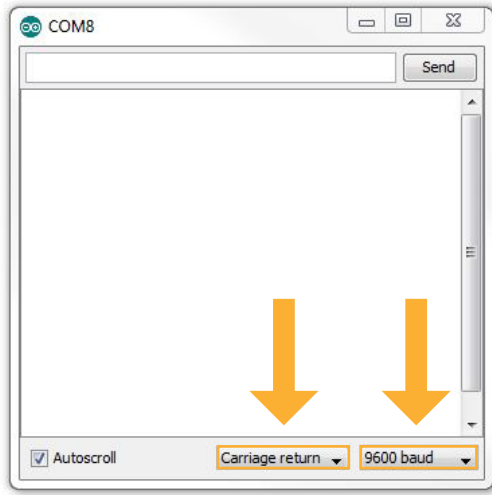
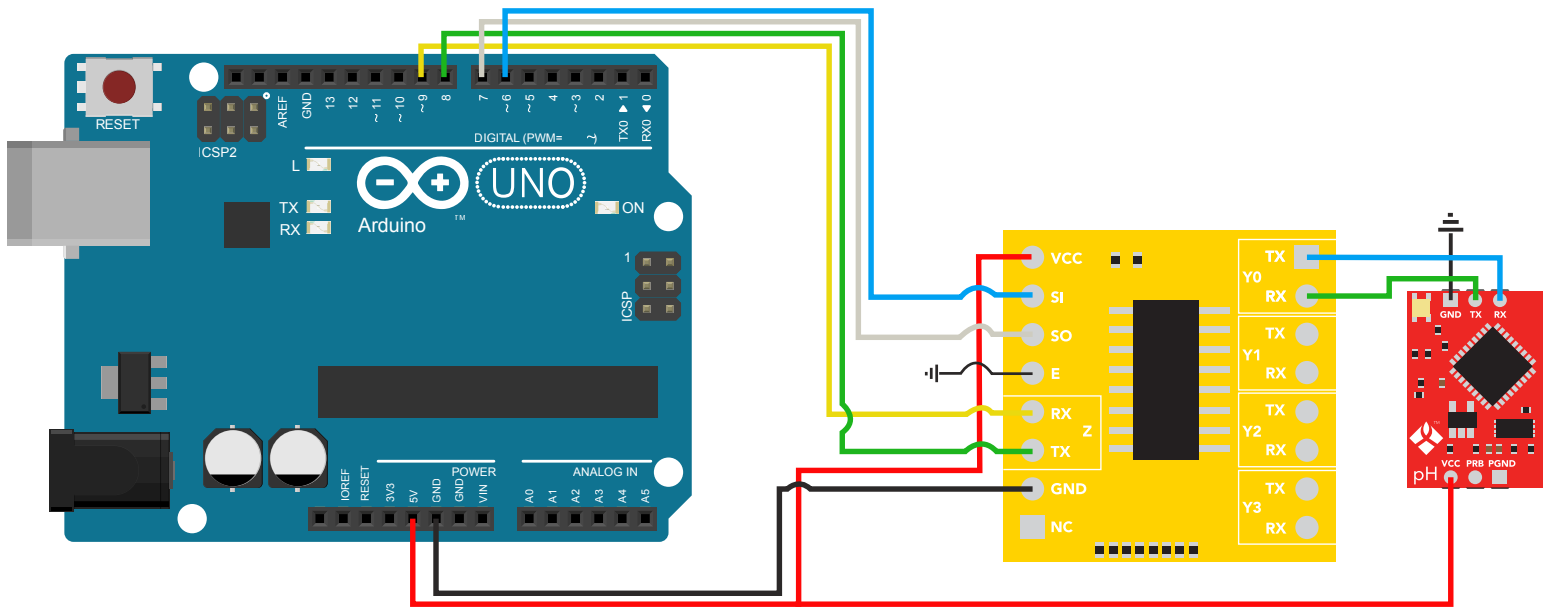


Serial Port Expander sample code



//This sample code was written on an Arduino UNO.
 //It will allow you to control up to 4 Atlas Scientific devices through 1 soft serial RX/TX line.
 //To open a channel (marked on the board as Y0 to Y3) send the number of the channel, a colon and the command ending with
 //a carriage return.



```
//0:r<CR>
//1:i<CR>
//2:c<CR>
//3: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>
AltSoftSerial altSerial;

int s0 = 7;
int s1 = 6;

char computerdata[20];
char sensordata[30];
byte computer_bytes_received=0;
byte sensor_bytes_received=0;

char *channel;
char *cmd;

void setup() {
  pinMode(s1, OUTPUT);
  pinMode(s0, OUTPUT);
  Serial.begin(9600);
  altSerial.begin(9600);
}

void serialEvent(){
  computer_bytes_received=Serial.readBytesUntil(13,computerdata,20);
  computerdata[computer_bytes_received]=0;
}

void loop(){
  if(computer_bytes_received!=0){
    channel= strtok(computerdata, ":");
    cmd= strtok(NULL, ":");
    open_channel();
    altSerial.print(cmd);
    altSerial.print("\r");
    computer_bytes_received=0;
  }

  if(altSerial.available() > 0){
    sensor_bytes_received=altSerial.readBytesUntil(13,sensordata,30);
    sensordata[sensor_bytes_received]=0;
    Serial.println(sensordata);
  }
}

void open_channel(){
  switch (*channel) {
    case '0':
      digitalWrite(s0, LOW);
      digitalWrite(s1, LOW);
      break;
    case '1':
      digitalWrite(s0, HIGH);
      digitalWrite(s1, LOW);
      break;
    case '2':
      digitalWrite(s0, LOW);
      digitalWrite(s1, HIGH);
      break;
    case '3':
      digitalWrite(s0, HIGH);
      digitalWrite(s1, HIGH);
      break;
  }
}

//Include the software serial library
//Name the software serial library altSerial (this cannot be omitted)

//Arduino pin 7 to control pin S0
//Arduino pin 6 to control pin S1

//A 20 byte character array to hold incoming data from a pc/mac/other
//A 30 byte character array to hold incoming data from the sensors
//We need to know how many characters bytes have been received
//We need to know how many characters bytes have been received

//Char pointer used in string parsing
//Char pointer used in string parsing

//Set the digital pin as output.
//Set the digital pin as output.
//Set the hardware serial port to 9600
//Set the soft serial port to 9600

//This interrupt will trigger when the data coming from
//the serial monitor(pc/mac/other) is received
//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

//If computer_bytes_received does not equal zero
//Let's parse the string at each colon
//Let's parse the string at each colon
//Call the function "open_channel" to open the correct data path
//Send the command from the computer to the 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 data has been transmitted from an Atlas Scientific device
//we read the data sent from the Atlas Scientific device until
//we see a <CR>. We also count how many character 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
//let's transmit the data received from the Atlas Scientific
//device to the serial monitor
```

[Click here to download the *.ino file](#)