



# 4 Port UART sample code

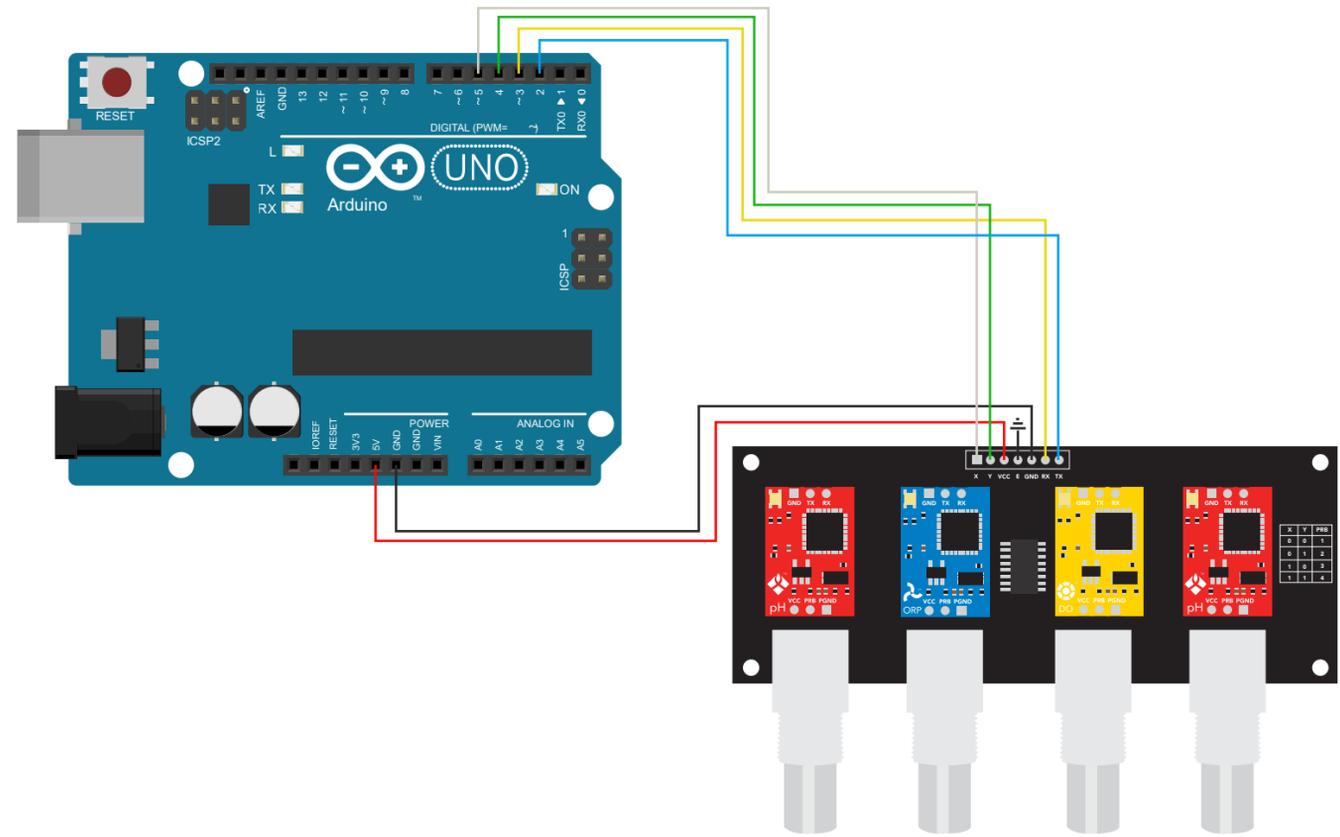
Revised 7/10/15

//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 PRB 1 to PRB 4) send the number of the channel,  
//a colon and the command ending 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>
```



```
#include <SoftwareSerial.h>           //Include the software serial library
#define rx 2                          //define what pin rx is going to be
#define tx 3                          //define what pin tx is going to be

SoftwareSerial myserial(rx, tx);      //define how the soft serial port is going to work.

int Pin_x = 5;                       //Arduino pin 5 to control pin X
int Pin_y = 4;                       //Arduino pin 4 to control pin Y

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(Pin_x, OUTPUT);            //Set the digital pin as output.
  pinMode(Pin_y, OUTPUT);            //Set the digital pin as output.
  Serial.begin(9600);                //Set the hardware serial port to 9600
  myserial.begin(9600);              //Set the hardware serial port to 9600
}

void serialEvent(){
  computer_bytes_received=Serial.readBytesUntil(13,computerdata,20); //This interrupt will trigger when the data coming from
  computerdata[computer_bytes_received]=0; //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

void loop(){

  if(computer_bytes_received!=0){    //If computer_bytes_received does not equal zero
  channel= strtok(computerdata, ":"); //Let's pars the string at each colon
  cmd= strtok(NULL, ":");           //Let's pars the string at each colon
  open_channel();                  //Call the function "open_channel" to open the correct data path
  if(cmd!=0){                       //if no command has been sent, send nothing
  myserial.print(cmd);              //Send the command from the computer to the Atlas Scientific device using the softserial port
  myserial.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(myserial.available() > 0){
  sensor_bytes_received=myserial.readBytesUntil(13,sensordata,30); //If data has been transmitted from an Atlas Scientific device
  sensordata[sensor_bytes_received]=0; //we read the data sent from the Atlas Scientific device until
  Serial.println(sensordata); //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

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

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

  case '1':                        //If *channel==1 then we open channel 1
  digitalWrite(Pin_x, LOW);        //Pin_x and pin_y control what channel opens
  digitalWrite(Pin_y, LOW);        //Pin_x and pin_y control what channel opens
  break;                            //Exit switch case

  case '2':
  digitalWrite(Pin_x, LOW);
  digitalWrite(Pin_y, HIGH);
  break;

  case '3':
  digitalWrite(Pin_x, HIGH);
  digitalWrite(Pin_y, LOW);
  break;

  case '4':
  digitalWrite(Pin_x, HIGH);
  digitalWrite(Pin_y, HIGH);
  break;
  }
}
```

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