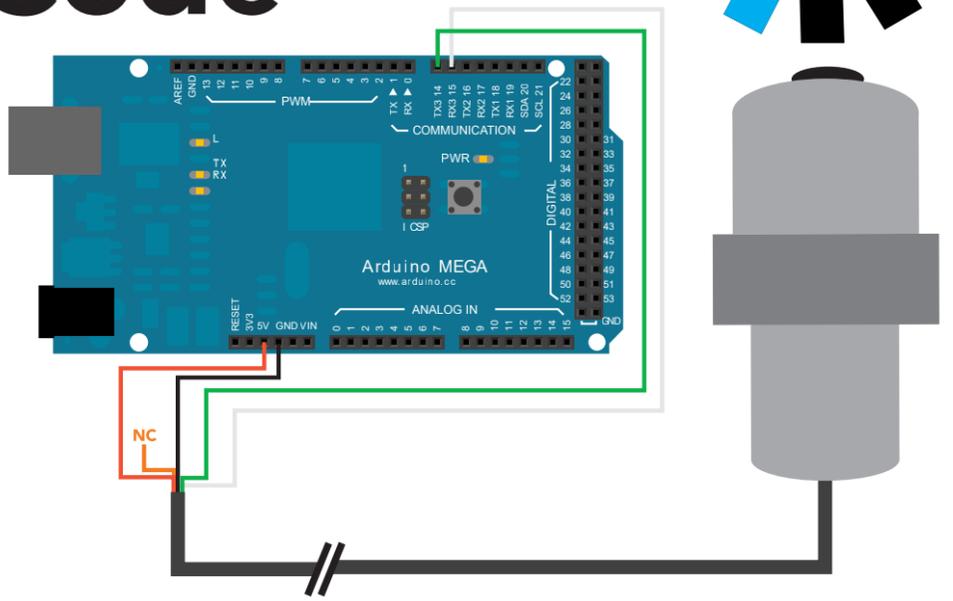
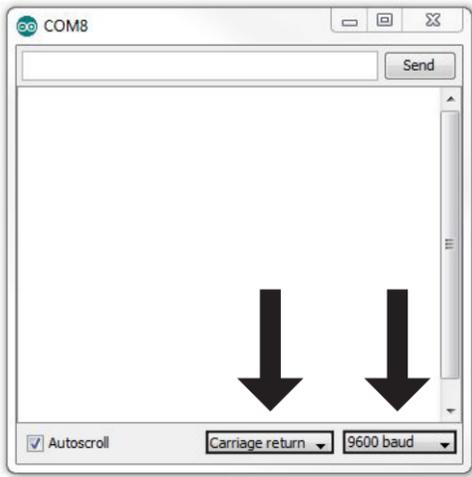


# Arduino Mega Sample Code



//This code was written to be easy to understand.  
//Code efficiency was not considered.  
//Modify this code as you see fit.  
//This code will output data to the Arduino serial monitor.  
//Type commands into the Arduino serial monitor to control the EZO-RGB.  
//This code was written in the Arduino 1.6.5 IDE  
//An Arduino MEGA was used to test this code.

```
String inputstring = "";
String sensorstring = "";
boolean input_string_complete = false;
boolean sensor_string_complete = false;
```

```
//a string to hold incoming data from the PC
//a string to hold the data from the Atlas Scientific product
//have we received all the data from the PC
//have we received all the data from the Atlas Scientific product
```

```
void setup() {
  Serial.begin(9600);
  Serial3.begin(9600);
  inputstring.reserve(10);
  sensorstring.reserve(30);
}
```

```
//set up the hardware
//set baud rate for the hardware serial port_0 to 9600
//set baud rate for software serial port_3 to 9600
//set aside some bytes for receiving data from the PC
//set aside some bytes for receiving data from Atlas Scientific product
```

```
void serialEvent() {
  inputstring = Serial.readStringUntil(13);
  input_string_complete = true;
}
```

```
//if the hardware serial port_0 receives a char
//read the string until we see a <CR>
//set the flag used to tell if we have received a completed string
//from the PC
```

```
void serialEvent3() {
  sensorstring = Serial3.readStringUntil(13);
  sensor_string_complete = true;
}
```

```
//if the hardware serial port_3 receives a char
//read the string until we see a <CR>
//set the flag used to tell if we have received a completed string
//from the PC
```

```
void loop() {
```

```
//here we go...
```

```
  if (input_string_complete == true) {
    Serial3.print(inputstring);
    Serial3.print('\r');
    inputstring = "";
    input_string_complete = false;
  }
```

```
//if a string from the PC has been received in its entirety
//send that string to the Atlas Scientific product
//add a <CR> to the end of the string
//clear the string
//reset the flag used to tell if we have received a completed string
//from the PC
```

```
  if (sensor_string_complete == true) {
    if (isdigit(sensorstring[0]) == false) {
      Serial.println(sensorstring);
    }
    else
    {
      print_RGB_data();
    }
    sensorstring = "";
    sensor_string_complete = false;
  }
}
```

```
//if a string from the Atlas Scientific product has been received in its entirety
//if the first character in the string is a digit
//send that string to the PC's serial monitor
```

```
//if the first character in the string is NOT a digit
```

```
//then call this function
```

```
//clear the string
//reset the flag used to tell if we have received a completed string from
//the Atlas Scientific product
```

```
void print_RGB_data(void) {
```

```
//this function will pars the string
```

```
  char sensorstring_array[30];
  char *red;
  char *grn;
  char *blu;
  int int_red;
  int int_grn;
  int int_blu;
```

```
//we make a char array
//char pointer used in string parsing
//char pointer used in string parsing
//char pointer used in string parsing
//used to hold an int that is the color red
//used to hold an int that is the color green
//used to hold an int that is the color blue
```

```
  sensorstring.toCharArray(sensorstring_array, 30);
  red = strtok(sensorstring_array, ",");
  grn = strtok(NULL, ",");
  blu = strtok(NULL, ",");
```

```
//convert the string to a char array
//let's pars the array at each comma
//let's pars the array at each comma
//let's pars the array at each comma
```

```
  Serial.print("RED:");
  Serial.println(red);
```

```
//we now print each value we parsed separately
//this is the red value
```

```
  Serial.print("GREEN:");
  Serial.println(grn);
```

```
//we now print each value we parsed separately
//this is the green value
```

```
  Serial.print("BLUE:");
  Serial.println(blu);
```

```
//we now print each value we parsed separately
//this is the blue value
```

```
// int_red= atoi(red);
// int_grn= atoi(grn);
// int_blu= atoi(blu);
}
```

```
//uncomment this line to convert the char to an int
//uncomment this line to convert the char to an int
//uncomment this line to convert the char to an int
```

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