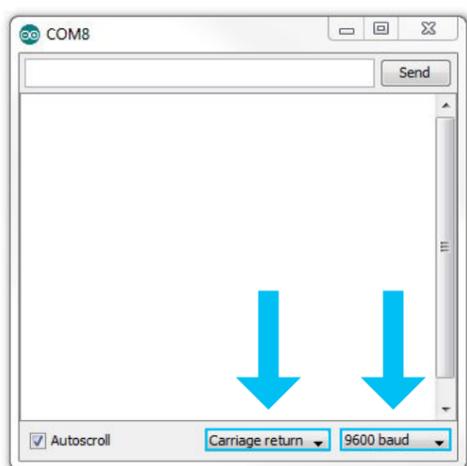


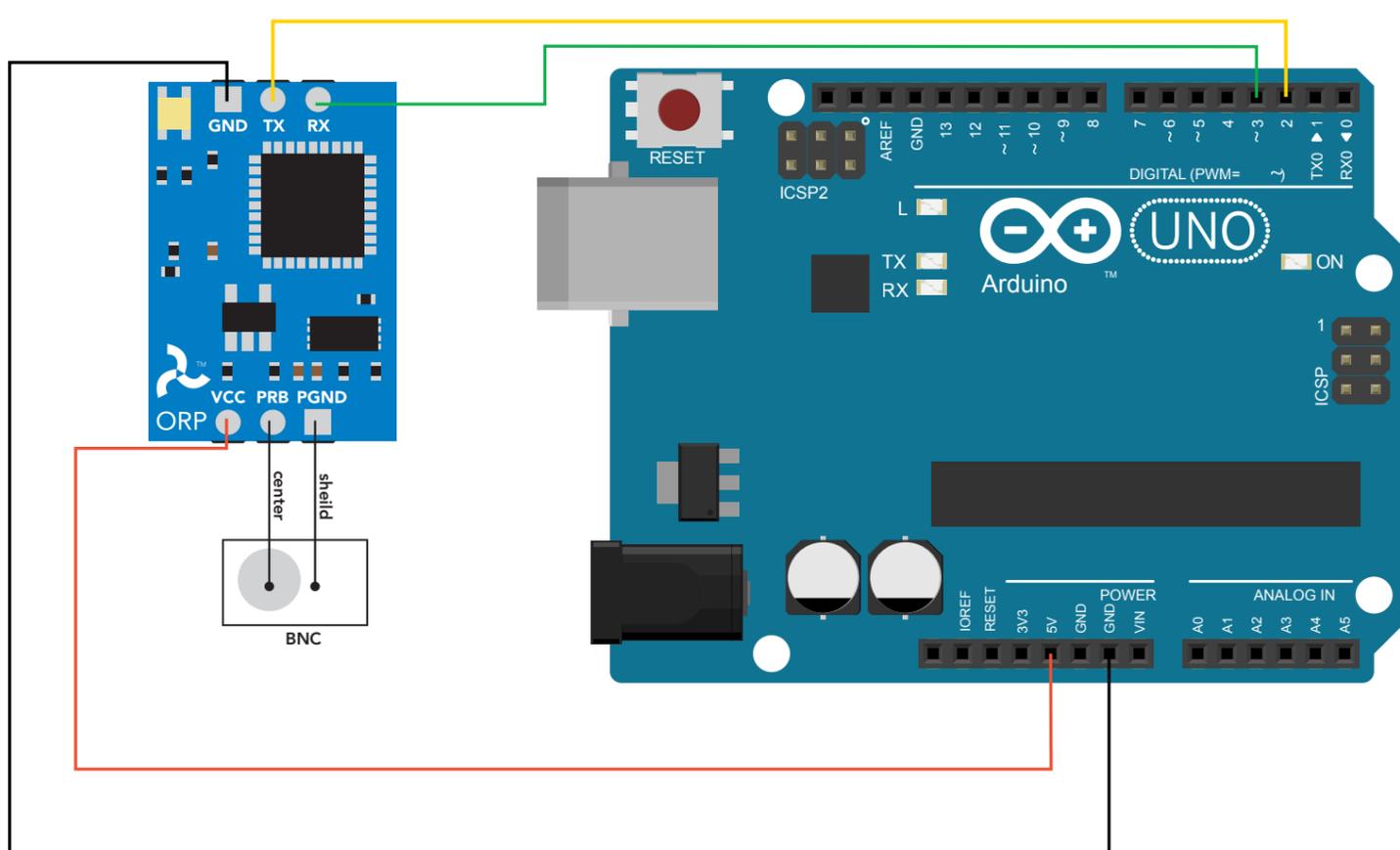


Arduino Uno

ORP 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 ORP circuit.
//This code was written in the Arduino 1.6.5 IDE
//An Arduino UNO was used to test this code.
```



```
#include <SoftwareSerial.h>
#define rx 2
#define tx 3

SoftwareSerial myserial(rx, tx);
```

```
//we have to include the SoftwareSerial library, or else we can't use it
//define what pin rx is going to be
//define what pin tx is going to be

//define how the soft serial port is going to work
```

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

```
//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
//used to hold a floating point number that is the ORP.
```

```
void setup() {
  Serial.begin(9600);
  myserial.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 the software serial port 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 loop() {

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

  if (myserial.available() > 0) {
    char inchar = (char)myserial.read();
    sensorstring += inchar;
    if (inchar == '\r') {
      sensor_string_complete = true;
    }
  }
}
```

```
//here we go...

//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 we see that the Atlas Scientific product has sent a character
//get the char we just received
//add the char to the var called sensorstring
//if the incoming character is a <CR>
//set the flag
```

```
if (sensor_string_complete == true) {
  Serial.println(sensorstring);
  if (isdigit(sensorstring[0])) {
    ORP = sensorstring.toFloat();
    if (ORP >= 500.0) {
      Serial.println("high");
    }
    if (ORP <= 499.9) {
      Serial.println("low");
    }
  }
  sensorstring = "";
  sensor_string_complete = false;
}
```

```
//if a string from the Atlas Scientific product has been received in its entirety
//send that string to the PC's serial monitor
//if the first character in the string is a digit
//convert the string to a floating point number so it can be evaluated by the Arduino
//if the ORP is greater than or equal to 500
//print "high" this is demonstrating that the Arduino is evaluating the ORP as a number
//and not as a string
//if the ORP is less than or equal to 499.9
//print "low" this is demonstrating that the Arduino is evaluating the ORP as a number
//and not as a string

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

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