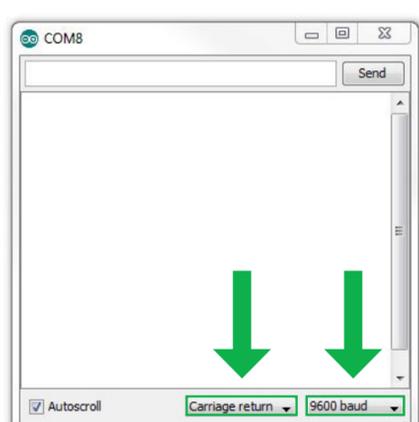
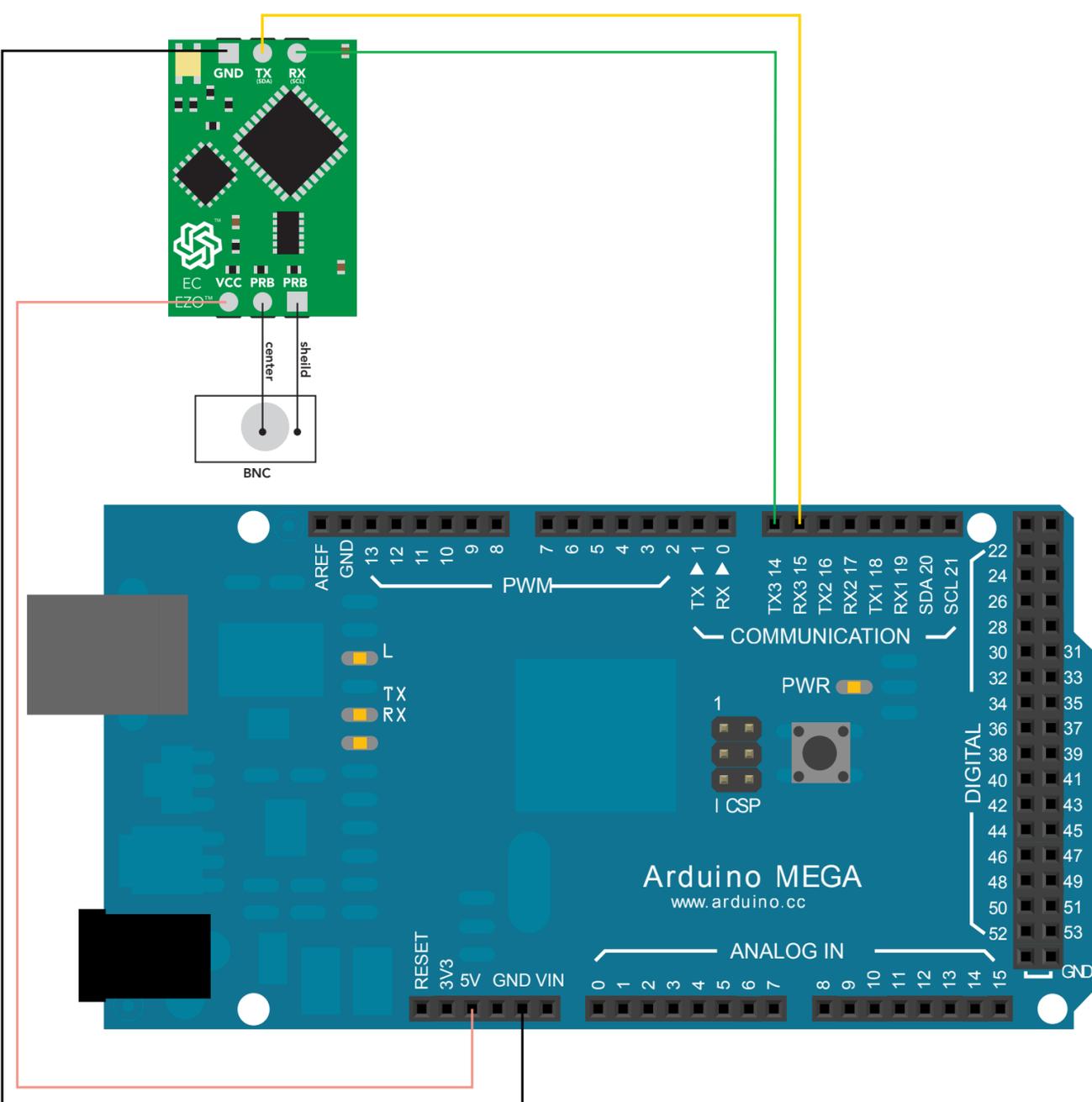




Arduino Mega Conductivity Sample Code



```
//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 EC circuit.
//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() {

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

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

```
//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 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_EC_data(void) {

  char sensorstring_array[30];
  char *EC;
  char *TDS;
  char *SAL;
  char *GRAV;
  float f_ec;

  sensorstring.toCharArray(sensorstring_array, 30);
  EC = strtok(sensorstring_array, ",");
  TDS = strtok(NULL, ",");
  SAL = strtok(NULL, ",");
  GRAV = strtok(NULL, ",");

  Serial.print("EC:");
  Serial.println(EC);

  Serial.print("TDS:");
  Serial.println(TDS);

  Serial.print("SAL:");
  Serial.println(SAL);

  Serial.print("GRAV:");
  Serial.println(GRAV);
  Serial.println();

  //f_ec= atof(EC);
}
```

```
//this function will pars the string

//we make a char array
//char pointer used in string parsing
//used to hold a floating point number that is the EC

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

//we now print each value we parsed separately
//this is the EC value

//we now print each value we parsed separately
//this is the TDS value

//we now print each value we parsed separately
//this is the salinity value

//we now print each value we parsed separately
//this is the specific gravity
//this just makes the output easier to read

//uncomment this line to convert the char to a float
```

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