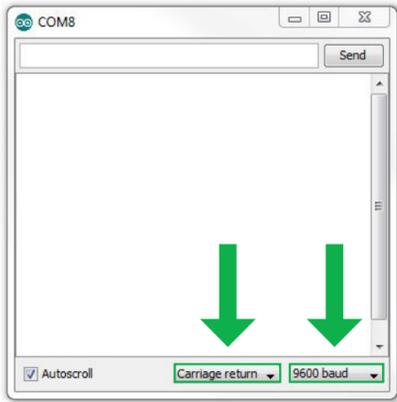


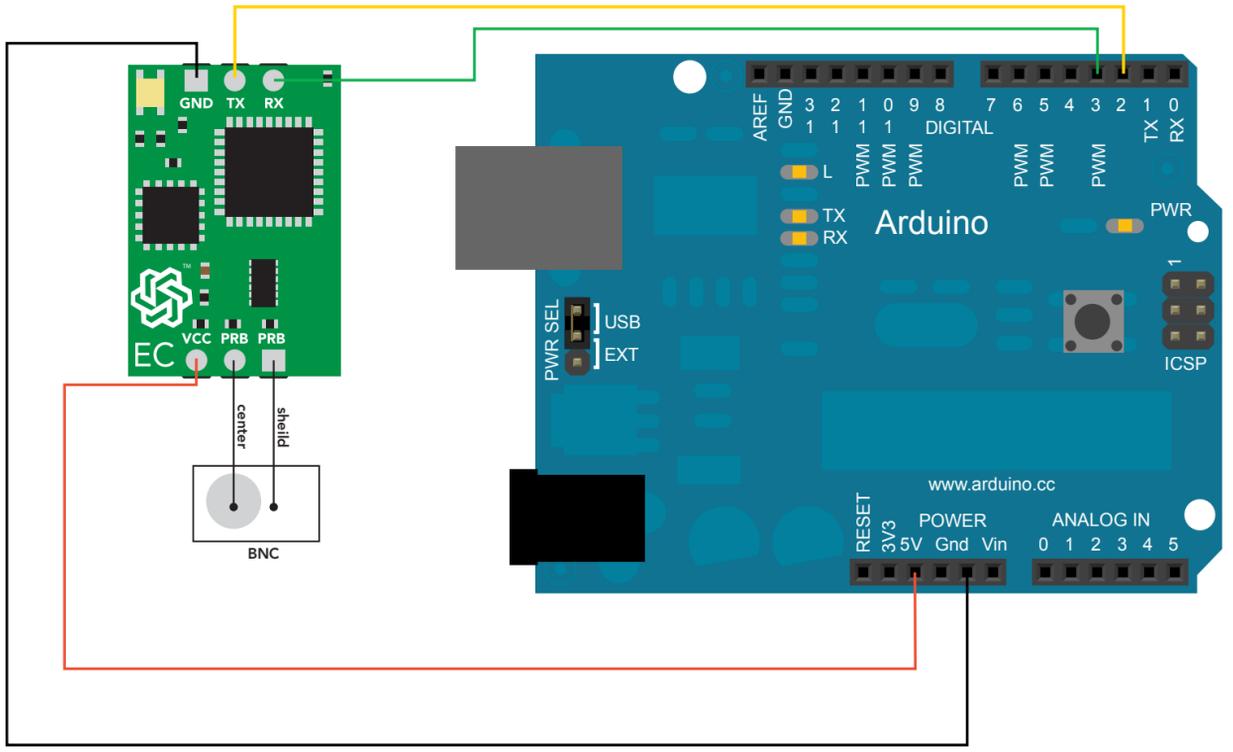


# Arduino E.C. Sample Code



//This code has intentionally has been written to be overly lengthy and includes //unnecessary steps. Many parts of this code can be truncated. 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.

//As of 11/6/14 the default baud rate has changed to 9600. //The old default baud rate was 38400.



```
#include <SoftwareSerial.h>           //we have to include the SoftwareSerial library, or else we can't use it.
#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.

char EC_data[48];                    //we make a 48 byte character array to hold incoming data from the EC.
char computerdata[20];               //we make a 20 byte character array to hold incoming data from a pc/mac/other.
byte received_from_computer=0;      //we need to know how many characters have been received.
byte received_from_sensor=0;        //we need to know how many characters have been received.
byte string_received=0;             //used to identify when we have received a string from the EC circuit.

float EC_float=0;                   //used to hold a floating point number that is the EC.
float TDS_float ;                   //used to hold a floating point number that is the EC.
float SAL_float;                    //used to hold a floating point number that is the Salinity.
float SAL_grav;                     //used to hold a floating point number that is the Specific gravity.

char *EC;                           //char pointer used in string parsing
char *TDS;                           //char pointer used in string parsing
char *SAL;                           //char pointer used in string parsing
char *GRAV;                          //char pointer used in string parsing

void setup(){
  Serial.begin(9600);                //enable the hardware serial port
  myserial.begin(9600);              //enable the hardware serial port
}

void serialEvent(){
  received_from_computer=Serial.readBytesUntil(13,computerdata,20); //this interrupt will trigger when the data coming from
  computerdata[received_from_computer]=0; //the serial monitor(pc/mac/other) is received.
  myserial.print(computerdata); //we read the data sent from the serial monitor
  myserial.print('\r'); //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.
//we transmit the data received from the serial monitor
//(pc/mac/other) through the soft serial port to the EC Circuit.
//all data sent to the EC Circuit must end with a <CR>.

void loop(){

  if(myserial.available() > 0){
    received_from_sensor=myserial.readBytesUntil(13,EC_data,48); //if we see that the EC Circuit has sent a character.
    EC_data[received_from_sensor]=0; //we read the data sent from EC Circuit 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.
    //if ec_data[0] is a digit and not a letter

    if((EC_data[0] >= 48) && (EC_data[0] <=57)){
      pars_data();
    }
    else
      Serial.println(EC_data); //if the data from the EC circuit does not start with a number
    //transmit that data to the serial monitor.
  }

  void pars_data(){

    EC=strtok(EC_data, ","); //let's pars the string at each comma.
    TDS=strtok(NULL, ","); //let's pars the string at each comma.
    SAL=strtok(NULL, ","); //let's pars the string at each comma.
    GRAV=strtok(NULL, ","); //let's pars the string at each comma.

    Serial.print("EC:"); //We now print each value we parsed sepratly.
    Serial.println(EC); //this is the EC value.

    Serial.print("TDS:"); //We now print each value we parsed sepratly.
    Serial.println(TDS); //this is the TDS value.

    Serial.print("SAL:"); //We now print each value we parsed sepratly.
    Serial.println(SAL); //this is the salinity value.

    Serial.print("GRAV:"); //We now print each value we parsed sepratly.
    Serial.println(GRAV); //this is the Specific gravity.
    Serial.println(); //this just makes the output easier to read.
  }

  //here are some functions you might find useful
  //these functions are not enabled
  /*

  void ECFactoryDefault(){ //factory defaults the EC circuit
    myserial.print("X\r"); //send the "X" command to factory reset the device

  void read_info(){ //get device info
    myserial.print("I\r"); //send the "I" command to query the information

  void sleep(){ //send the "sleep" command to put the EC circuit in a low power state
    myserial.print("sleep\r");

  void ECSetLEDs(byte enabled) //turn the LEDs on or off
  {
    if(enabled) //if enabled is > 0
      myserial.print("L,1\r"); //the LED's will turn ON
    else //if enabled is 0
      myserial.print("L,0\r"); //the LED's will turn OFF
  }
  */
```



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