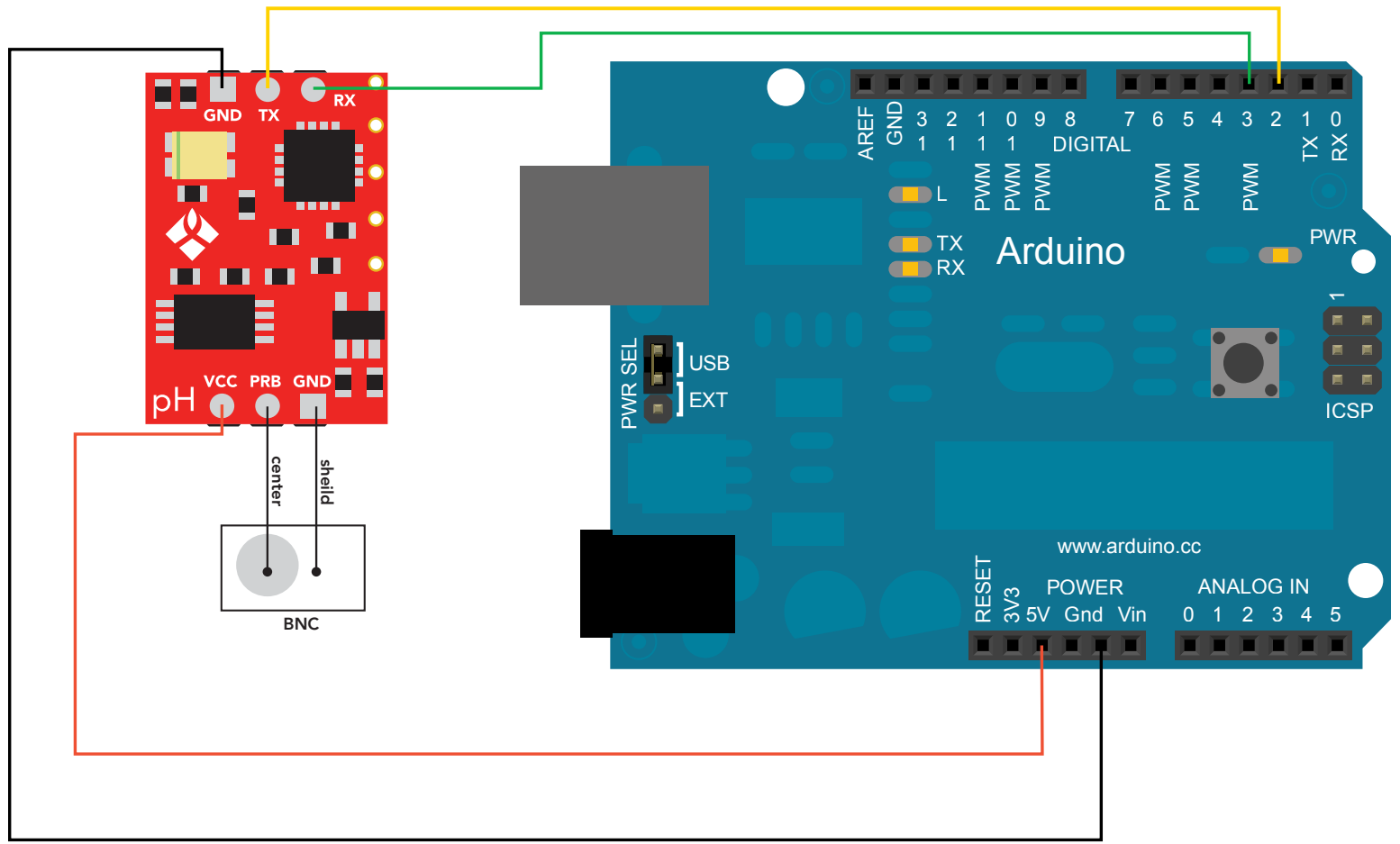




# EZ pH

//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 pH circuit.  
 //set the var Arduino\_only to equal 1 to watch the Arduino take over control of the pH circuit.



```
#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 ph_data[20]; //we make a 20 byte character array to hold incoming data from the pH.
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 arduino_only=0; //if you would like to operate the pH Circuit with the Arduino only and not use the
//serial monitor to send it commands set this to 1. The data will still come out on the
//serial monitor, so you can see it working.

byte startup=0; //used to make sure the Arduino takes over control of the pH Circuit properly.
float ph=0; //used to hold a floating point number that is the pH.
byte string_received=0; //used to identify when we have received a string from the pH circuit.

void setup(){
  Serial.begin(38400); //enable the hardware serial port
  myserial.begin(38400); //enable the hardware serial port
}

void serialEvent(){
  if(arduino_only!=1){
    received_from_computer=Serial.readBytesUntil(13,computerdata,20);
    computerdata[received_from_computer]=0;
    myserial.print(computerdata);
    myserial.print('\r');
  }
}

void loop(){
  if(myserial.available() > 0){
    received_from_sensor=myserial.readBytesUntil(13,ph_data,20);
    ph_data[received_from_sensor]=0;
    string_received=1;
    Serial.println(ph_data);
  }
  if(arduino_only==1){Arduino_Control();}

}

void Arduino_Control(){
  if(startup==0){
    myserial.print("e\r"); //if the Arduino just booted up, we need to set some things up first.
    delay(50); //take the pH Circuit out of continues mode.
    myserial.print("e\r"); //on start up sometimes the first command is missed.
    delay(50); //so, let's send it twice.
    startup=1; //a short delay after the pH Circuit was taken out of continues mode is used to make sure we don't
    //over load it with commands.
    //startup is completed, let's not do this again during normal operation.
  }

  delay(800); //we will take a reading ever 800ms. You can make this much longer or shorter if you like.
  myserial.print("R\r"); //send it the command to take a single reading.
  if(string_received==1){ //did we get data back from the pH Circuit?
    ph=atof(ph_data); //many people ask us "how do I convert a sting into a float?" This is how...
    if(ph>=7.5){Serial.println("high\r");} //This is the proof that it has been converted into a float.
    if(ph<7.5){Serial.println("low\r");} //This is the proof that it has been converted into a float.
    string_received=0; //reset the string received flag.
  }
}

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

void s_cal(){ //calibrate to a pH of 7
  myserial.print("s\r"); //send the "s" command to calibrate to a pH of 7.00

}

void f_cal(){ //calibrate to a pH of 4
  myserial.print("f\r"); //send the "f" command to calibrate to a pH of 4.00

}

void t_cal(){ //calibrate to a pH of 10
  myserial.print("t\r"); //send the "t" command to calibrate to a pH of 10.00

}

void phFactoryDefault(){ //factory defaults the pH 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 phSetLEDs(byte enabled) //turn the LEDs on or off
{
  if(enabled) //if enabled is > 0
  myserial.print("L1\r"); //the LED's will turn ON
  else //if enabled is 0
  myserial.print("L0\r"); //the LED's will turn OFF
}
```

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