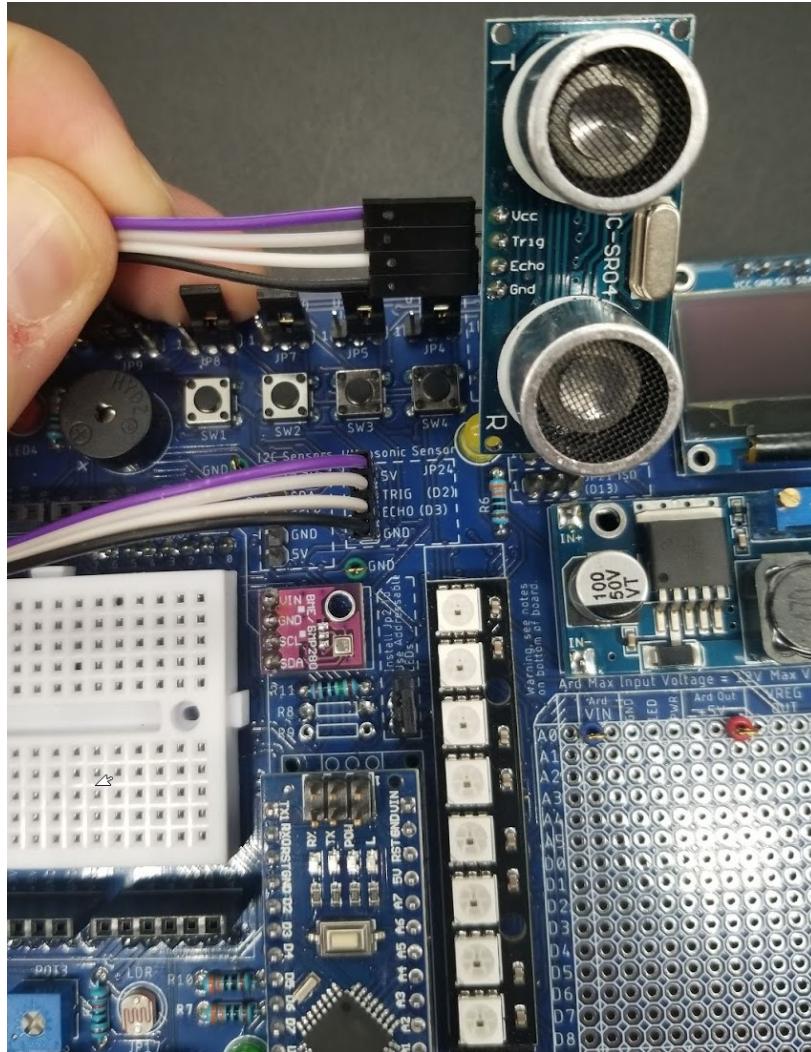


DEMO PROJECTS  
FOR THE PIONEER AND EXPLORER EDITIONS



DOCUMENT REVISTION HISTORY	REV NUMBER
Initial Doc Creation (2/27/2020)	A.0.1
Corrected Water Leak detector details for Explorer (4/20/2020)	A.0.2

## CONTENTS

FOREWARD .....	6
Premium Sketches .....	7
Demo Code – How They work .....	9
Section 1- Code description .....	9
Section 2- General Setup .....	11
Section 3- User Variables .....	12
Section 4- Setup .....	12
Section 5- LOOP .....	13
Selecting Starter Projects.....	15
Larson Scanner .....	16
Larson Scanner   Hardware Used On Pioneer And Explorer .....	17
Larson Scanner  How it works.....	18
Annoy-A-Buddy.....	20
Annoy-A-Buddy   Hardware Used On Pioneer And Explorer .....	21
Annoy-A-Buddy   How it works.....	22
Night Light .....	23
Night Light   Hardware Used On Pioneer And Explorer .....	24
Night Light  How it works.....	25
Distance Detector and collison warning System .....	26
Distance Detector And Collision Warning System   Hardware Used On Pioneer And Explorer .....	27
Distance detector and collision warning  How it works.....	28
RGB-Controller with Recall Function! .....	29
RGB controller with recall funciton   Hardware Used On Pioneer And Explorer .....	30
RGB controller with recall funciton  How it works.....	31
Selecting PREMIUM Projects .....	34
Installing Premium Projects .....	35
Color Guage .....	36
Color Gauge   Hardware Used On Pioneer And Explorer .....	37
Color Guage How it works .....	37
The Perfect Kitchen timer.....	39
The perfect kitchen timer   Hardware Used On Pioneer And Explorer.....	40

The perfect kitchen timer  How it works .....	41
Water leaK Detector .....	43
Water Leak Detector   Hardware Used On Pioneer And Explorer .....	44
Water Leak Detector  How it works.....	45
You're The Tone Man .....	47
You're The Tone Man   Hardware Used On Pioneer And Explorer .....	48
You're The Tone Man  How it works.....	49
Light Theramin.....	50
Light Theramin  How it works.....	51
Light Theramin  How it works.....	52
Closing Thoughts .....	53
Disclaimer .....	<b>Error! Bookmark not defined.</b>
No Liability .....	<b>Error! Bookmark not defined.</b>

# IMPORTANT!

BEFORE YOU BEGIN, REVIEW THE TERMS AND CONDITIONS WHEN USING  
PRODUCTS FROM DR.DUINO.

If that link doesn't work copy and paste this one into your browser:

<https://bit.ly/3e8FOHC>

## FOREWARD

Many kits come with “sample” code, which is, well... let’s face it, less than helpful.

It’s written in such a way that you need a PHD in bio-rocketry surgery engineering (how’s that for a degree? I just made that up) or just plain ole doesn’t work.

Well... that’s exactly why I created demo projects which not only WORK, but work incredibly well with your hardware.

This manual focuses exclusively on the 10 demo projects which came with your Dr.Duino Pioneer or Dr.Duino Explorer.

Depending on which version you purchased, your kit either came with 5 Starter projects or 5 starter projects and 5 premium projects.

## STARTER SKETCHES

---

**Larson Scanner**, remember KITT from knight rider or the Cylons from Battlestar Galactica? Well, you will turn your kit into your own iconic versions of their cycling lights!

**Annoy-A-Buddy**, tuck away your Pioneer or Explorer in a coworker’s cube, sit back and get ready to laugh. Watch them go crazy looking for where an annoying chirping sound is coming from. BUT, it has a nice annoying trick up it’s sleeve. You’ll see ☺

**Night Lite**, thanks to the built in LDR sensor, you’ll be able to create your very own night lite perfect for those late-night trips to the loo or keeping your little one’s scary monsters away at night!

**RGB Controller**, there’s nothing more mesmerizing than being able to control the color and intensity of LED’s especially when you can mix RED, GREEN & BLUE together. You’ll be able to create up to 16 million color variations!

**Collision Warning System**, thanks to the magic of sonar, you’ll turn your pioneer or explorer into an obstacle warning machine which can warn you as you get too close to the object.

---

## PREMIUM SKETCHES

**Water Leak Detector**, ever worry that your boiler, sink or washer machine will start to leak but you won't know till its waaaaaaay too late? Well, worry no more! I'll show you how to detect water and sound an alarm!

**You're The Tone Man**, create an awesome 88 tone capable electronic keyboard with ease! You'll be making some sweet, sweet sounds in no time.

**The Perfect Kitchen Timer**, wouldn't it be awesome if you could customize a timer for anything you need? But not an ordinary timer, something which ads a bit of lighting effects too! Well, look no further, you'll have the code to do it!

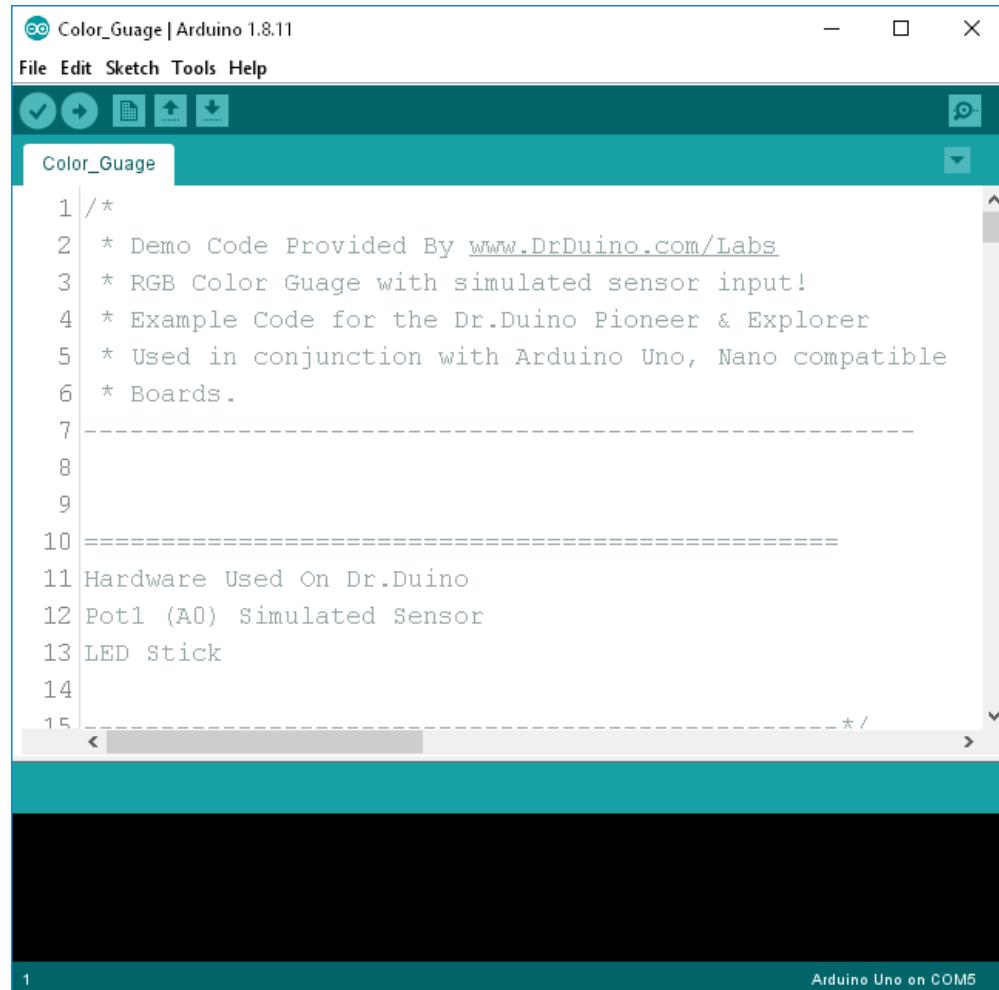
**Color Gauge**, tons of projects need a way to notify the user how much time is left, or how full something is like an oil tank in a visual way. Well, this will do exactly that using high brightness LEDS! You'll be able to see the status of your gauge from clear across the room!

**Light Theramin**, a unique twist on an oldie. Shine a little light on to your kit and enjoy the sweet, sweet sound of converting light into beeps and boops! This one is a ton of fun to play with!

If you didn't purchase the Premium version of the kit but would still like the premium sketches they are available for purchase here:

[www.drduino.com/PremiumSketches](http://www.drduino.com/PremiumSketches)

# DEMO CODE | HOW IT'S SETUP



The screenshot shows the Arduino IDE interface with the title bar "Color\_Guage | Arduino 1.8.11". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for back, forward, upload, and download. The main window displays the code for the "Color\_Guage" sketch. The code is as follows:

```
1 /*  
2 * Demo Code Provided By www.DrDuino.com/Labs  
3 * RGB Color Guage with simulated sensor input!  
4 * Example Code for the Dr.Duino Pioneer & Explorer  
5 * Used in conjunction with Arduino Uno, Nano compatible  
6 * Boards.  
7 -----  
8  
9  
10 ======  
11 Hardware Used On Dr.Duino  
12 Pot1 (A0) Simulated Sensor  
13 LED Stick  
14  
15 -----*/
```

The status bar at the bottom indicates "1" and "Arduino Uno on COM5".

## DEMO CODE – HOW THEY WORK

Every code sample provided is commented to the hilt, so you will know exactly what it's doing and how to change it.

BUT! Before we begin, I'd like to show you how the code is formatted.

I wrote it specifically so that each code contains the same sections.

This way when you move from sketch to sketch, you know exactly where things are.

Let's take a look at a premium sketch named, "You're The Tone Man" and describe its various sections.

### SECTION 1- CODE DESCRIPTION



```
00 Youre_The_Tone_Man | Arduino 1.8.12 (Windows Store 1.8.33.0)
File Edit Sketch Tools Help

1 //*****
2 Dr.Duino Pioneer Shield Sketch
3 Written by: Steven Calinski
4 www.DrDuino.com
5
6 This fancy little sketch turns your Pioneer or Explorer shields into a
7 beeping and booping machine which can play up to 4 differnt notes.
8
9 In the user vaiable section of the code, you can assign different note frequencies
10 then when the user presses any of the four buttons, it will make that sound from
11 the piezo buzzer!
12
13 This one is a lot of fun to play with!
14 -----Note-----
15 Arduino is a trademark of Arduino AG. Dr.Duino is not affiliated or endorsed by Arduino AG
16 *****/
17 /* Hardware used on Pioneer or Explorer
18 * SW1-SW4- Button press
19 * Piezo Buzzer
20 */

```

This first section is actually broken up into two different sections.

The first (lines 1-13) tell you what the sketch will make your Pioneer or Explorer do.

Lines 17-20 tell you what hardware on the Pioneer or Explorer is used.

Fun fact... Notice the Written by on this one?

This was written by one of my students who I have been training in the ways of the electronic force since he was 14 years old.

He is part of my STEEAM (Science, Technology, Engineering, Entrepreneurship, Art and Mathematics) class I teach.

At the time of this writing, he is 20 years old and going to school for engineering.

Awesome, right!

[Here is one of our early projects we build when they were all around 14 or 15.](#)

If that link doesn't work, try copying and pasting this into your browser.

<https://bit.ly/333GvNh>

## SECTION 2- GENERAL SETUP

This section of the code sets up all of the libraries necessary to run your code, headers files (line 25) and also “instantiates” (line 28) them.

That's just a fancy way of saying on line 28:

“Hey, you, IDE, I'm going to use the library I just told you about (`DrDuino_Pioneer` highlighted in orange) and name it `Pioneer` (in black)”.

That “block” of code is actually, what you installed from the Dr.Duino Labs when you built your kit.

```
22 /*---- WARNING, don't play with the next few lines or bad things will ensue.  
23 Including but not limited to angering of the compiler god-----*/  
24 //-----Calling libraries needed to run-----//  
25 #include <DrDuino_Pioneer.h>  
26  
27 //-----Object Instantiation-----//  
28 DrDuino_Pioneer Pioneer;  
29  
30 //-----Generic Variable Declarations. Please do not change these-----//  
31  
32 // NOTE* (Pun Intended), here are all the notes you can play Use these in your code!  
33  
34 #define NOTE_B0 31  
35 #define NOTE_C1 33  
36 #define NOTE_CS1 35  
37 #define NOTE_D1 37  
38 #define NOTE_DS1 39  
39 #define NOTE_E1 41  
40 #define NOTE_F1 44  
41 #define NOTE_FS1 46
```

Finally, each sketch will use different variables and declarations (lines 30 to 41). These are specific to this code.

With that said, avoid making any changes to anything in this section.

Bad things will ensue if you do!

### SECTION 3- USER VARIABLES

Just about every one of the sketches provided has a section where you are encouraged to change the values.

In the example below, its from lines 124 to 127, but every sketch is different.

However it will always say:

```
//USER MODIFIABLE VARIABLES, PLAYING IS ENCOURAGED, NAAAY REQUIRED!
```

```
124 //-----USER MODIFIABLE VARIABLES, PLAYING IS ENCOURAGED, NAAAY REQUIRED!-----  
125  
126 int duration      = 100; // change this to change how long the note is played for. 100ms is a good start.  
127           // but if you want faster reaction time from the buttons, lower the number.  
128  
129
```

### SECTION 4- SETUP

The Arduino eco system is awesome for so many reasons, one of which is the setup area.

This is code, which is only executed once and only at startup.

Once it leaves this area, it will never return unless the program is re-downloaded, you hit the reset button or cycle power.

In this particular sketch, we didn't need anything but if we did, it would go in between the brackets on line 131 and 134.

```
130 void setup()  
131 {  
132  
133  
134 }
```

## SECTION 5- LOOP

The loop or the “meat and potatoes” as I like to call it is where all the magic happens.

This code executes forever and ever in a loop and is where all of your code is.

```
136 void loop()
137 {
138
139     /* This sketch is really simple, it just listen for button presses,
140      * depending on which switch has been pressed, it will emit the
141      * musical note which is set in the Generic Variable Declarations section.
142      *
143      * If you want to play a different note than what the sketch is currently
144      * setup for, just grab a note from that section, and plug it into
145      * the code below!
146      *
147      * Easy Peasy!
148      *
149      * Tone function only requires 3 arguments
150      * 1) The pin you want to use with your piezo
151      * 2) The frequency of the note.
152      * 3) How long do you want to play it for.
153      *
154      * In it's generic form it looks like this
155      * tone(Pin#, Frequency, duration)
156      */
157
158     if(digitalRead(Pioneer.SW1) == LOW)
159     {
160         tone(Pioneer.PiezoBuzzer,NOTE_C6,duration);
161     }
162     if(digitalRead(Pioneer.SW2) == LOW)
163     {
164         tone(Pioneer.PiezoBuzzer,NOTE_D6,duration);
165     }
166     if(digitalRead(Pioneer.SW3) == LOW)
167     {
168         tone(Pioneer.PiezoBuzzer,NOTE_E6,duration);
169     }
170     if(digitalRead(Pioneer.SW4) == LOW)
171     {
172         tone(Pioneer.PiezoBuzzer,NOTE_F6,duration);
173     }
```

Note\*- The line numbers will change from sketch to sketch.

This is just for demonstration purposes and to orient you while you are reviewing the code.

## STARTER PROJECTS



---

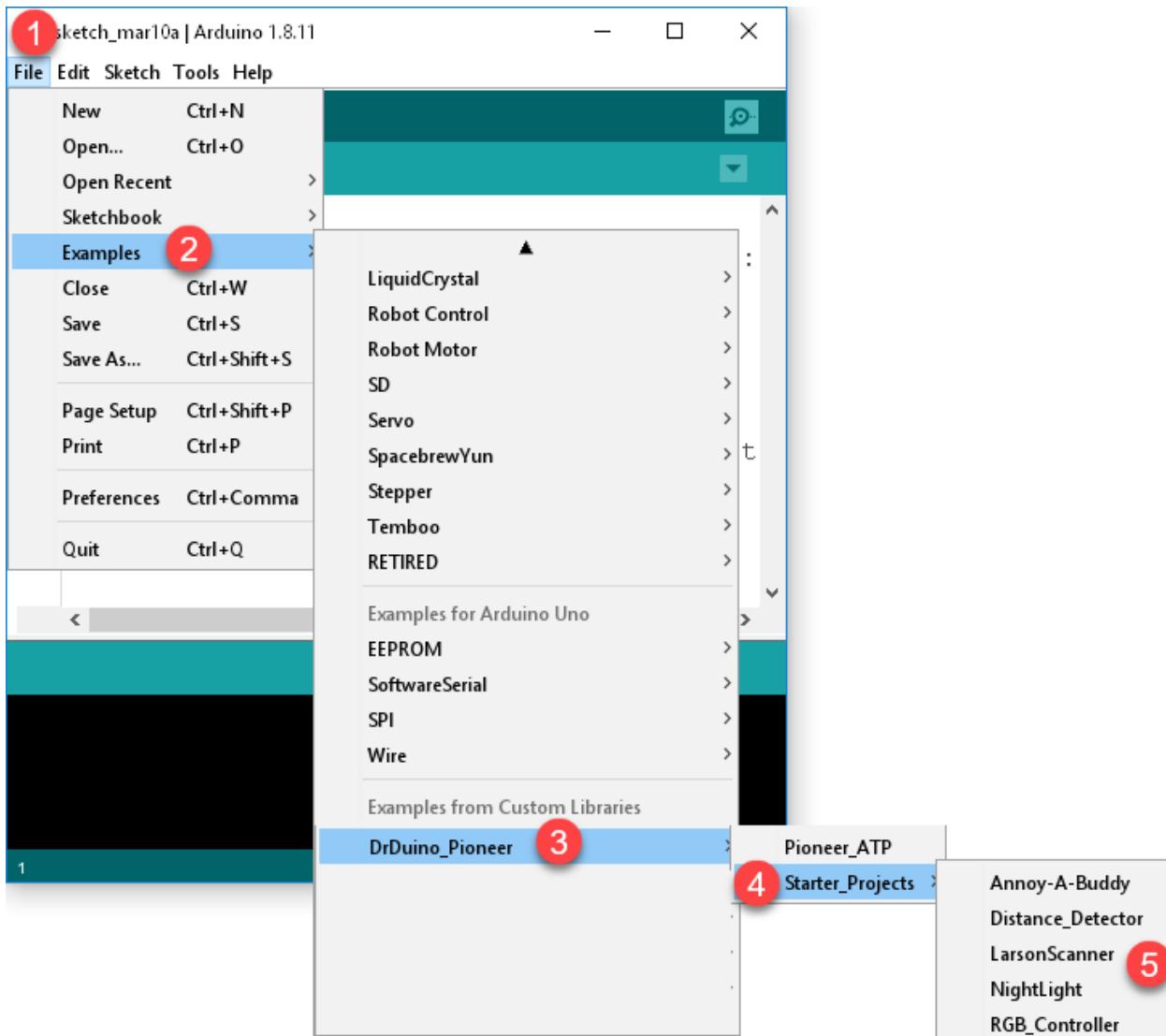
## SELECTING STARTER PROJECTS

When you installed the libraries, you automatically installed the starter projects too!

So all you need to do is navigate to the following location and select whichever project you would like to run first.

Below is the example for the Pioneer, but if you have the Explorer, it will be identical except it will say DrDuino\_Explorer instead of DrDuino\_Pioneer.

All of the projects will work identically on either platform.



## LARSON SCANNER

This is hands down one of my favorite projects, which is named after its inventor Glen A. Larson.

He was a producer in shows like Battle Star Galactica, Buck Rodgers, and my personal favorite KNIGHT RIDER!

Man, oh man, I used to love watching that.

I was fascinated by the tech, especially KITT's strobing lights in the front of the grill.

Well, I made your Pioneer or Explorer into your very own KITT!

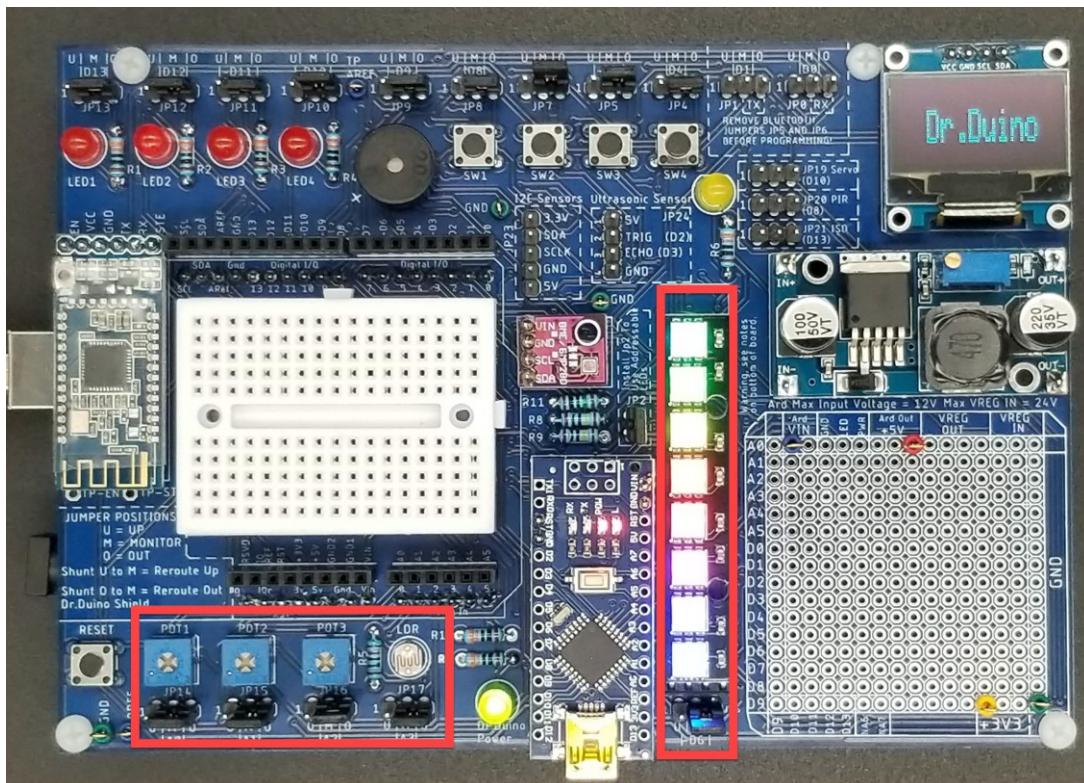
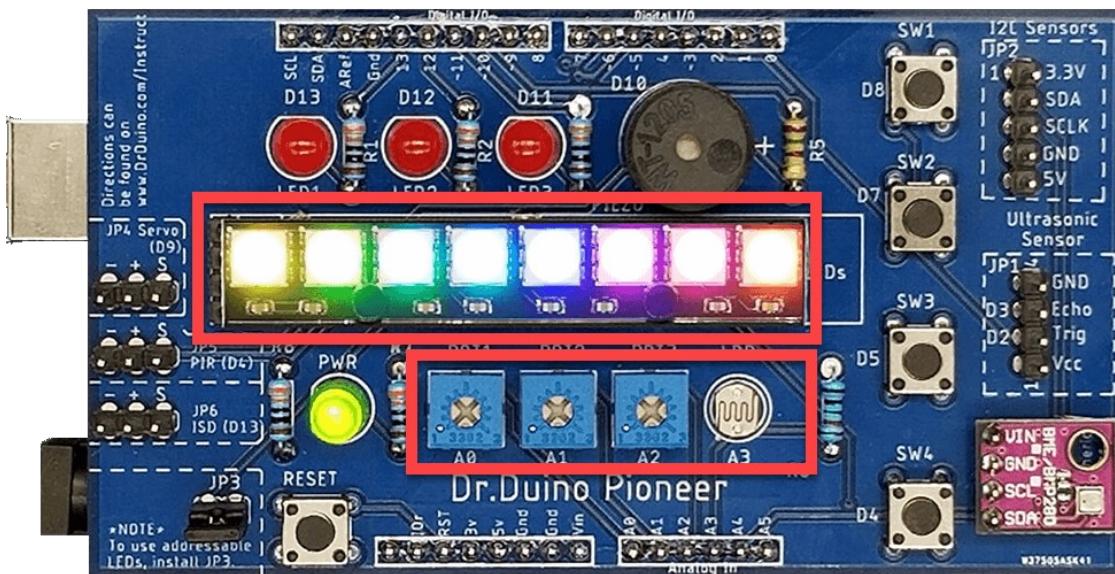
Here is what you do:

- 1) After you download your code, grab yourself a screwdriver.
- 2) Slowly turn the potentiometers Pot 1, 2 and 3 to whatever you want.

Just don't over turn it too much left or right. Once you can't turn it anymore, don't force it.

- 3) Sit back and watch in amazement as the light flows back and forth in various colors.

## LARSON SCANNER | HARDWARE USED ON PIONEER AND EXPLORER



This project is using the Pioneers built in potentiometers and mapping each one to a color.

Pot 1=Red

Pot 2 = Green

Pot 3 = Blue

The sketch is then reading the pots and mixing the values together.

This in turn, translates to a single color.

Well, one of actually over 16 million possible color combinations.

YUP, 16 million!

Why?

That's because each LED can be set with in a range of 0 to 255 bits.

Where zero is off and 255 is max brightness.

When you turn each one of the pots, the microcontroller on the Arduino Uno is constantly reading it.

This value is sent out to the LED stick, which sets the brightness for that individual color.

Since the LED's are so close to one another, inside that tiny little package, the colors end up mixing.

In essence, you are selecting one value out of 255 potential values for each red, green and blue led.

This results in 255 possible combinations of red, 255 possible combinations of Green and 255 possible combinations of blue.

With a little fancy math  $255 \times 255 \times 255 = 16,581,375$  possible combinations.

Which, when mixed together, is actually a single color out of over 16 million color combinations!

Pretty amazing, huh?

I also added a little fun to it.

You can change the code so that by shinning a light on the LDR, that will actually change the speed at which the lights go back and forth!

Just change the line of code from

**UselightSensor = true;**

**To**

**UselightSensor = false;**

Note\* - be careful not to delete the semicolon after the word false, if you do, you will be visited by the angry compiler god.

By changing the word from true to false, that tells the code, don't use the LDR anymore.

Use a fixed time instead. It is currently set to about 50 ms in between each LED.

Feel free to change that value too!

The Larson scanner is awesome for Halloween costumes or just a cool desk decoration hanging out moving back and forth!

That's it, Happy Larson Scanning!

## ANNOY-A-BUDDY

Those who know me know I love a good prank. Well... here is one, which I think you will enjoy.

This sketch will emit an annoying BEEEEEEEEEPPPPP randomly for a period of up to 2 minutes.

The Uno will randomly pick a time anywhere from 0 to 120 seconds and make a BEEEEEEEP.

Sometimes you'll get quick rapid bursts of it, other times very long in between the beeps.

Now, here is where it gets diabolical.

Let's say said Victim finds our little annoy-a-buddy.

Well... it has one more trick up its sleeve.

Remember that light sensor we installed, here is where that comes in.

As soon as it sees that it is not in a dark space anymore, it will not make a peep.

Not a single peep, so they will keep looking for the source. BAHAHAHAHA!

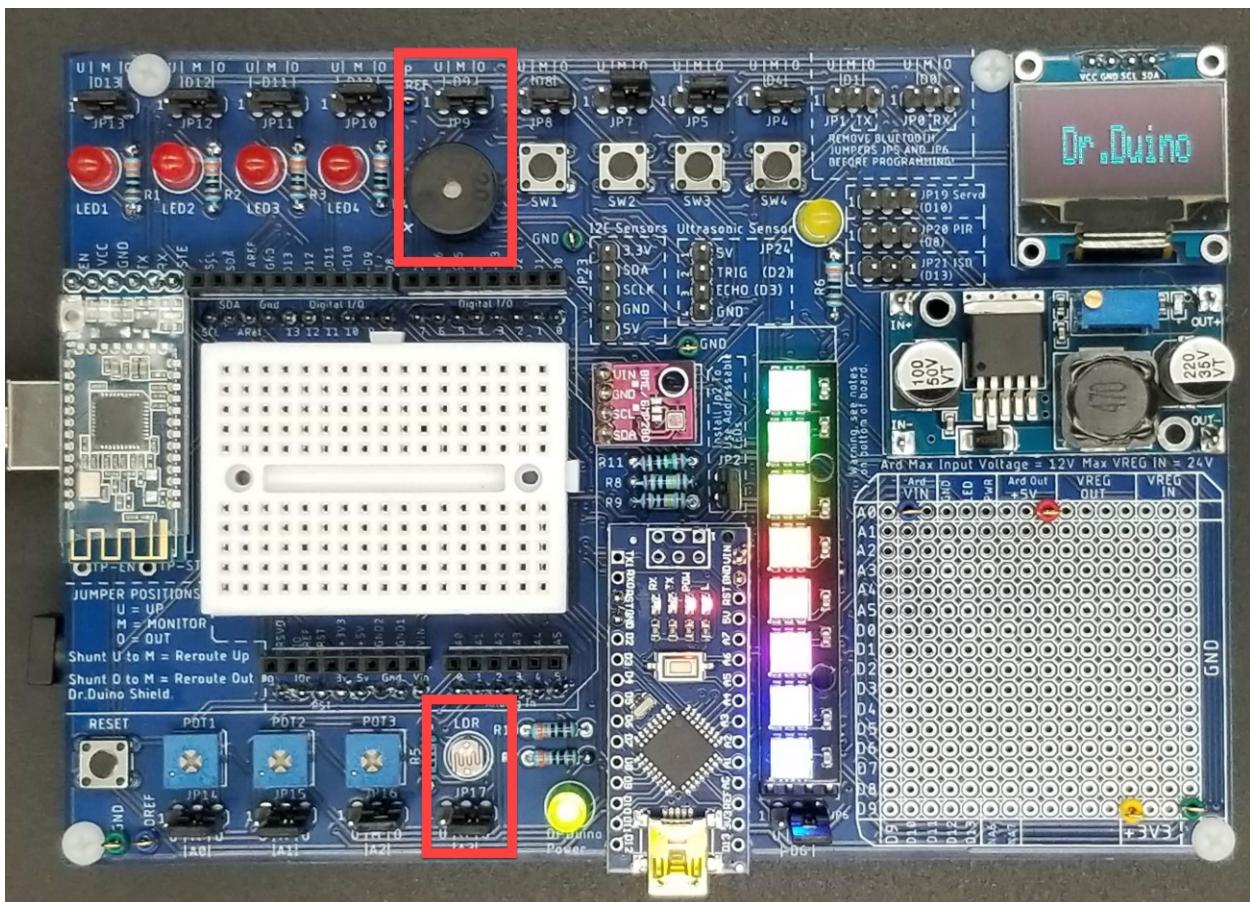
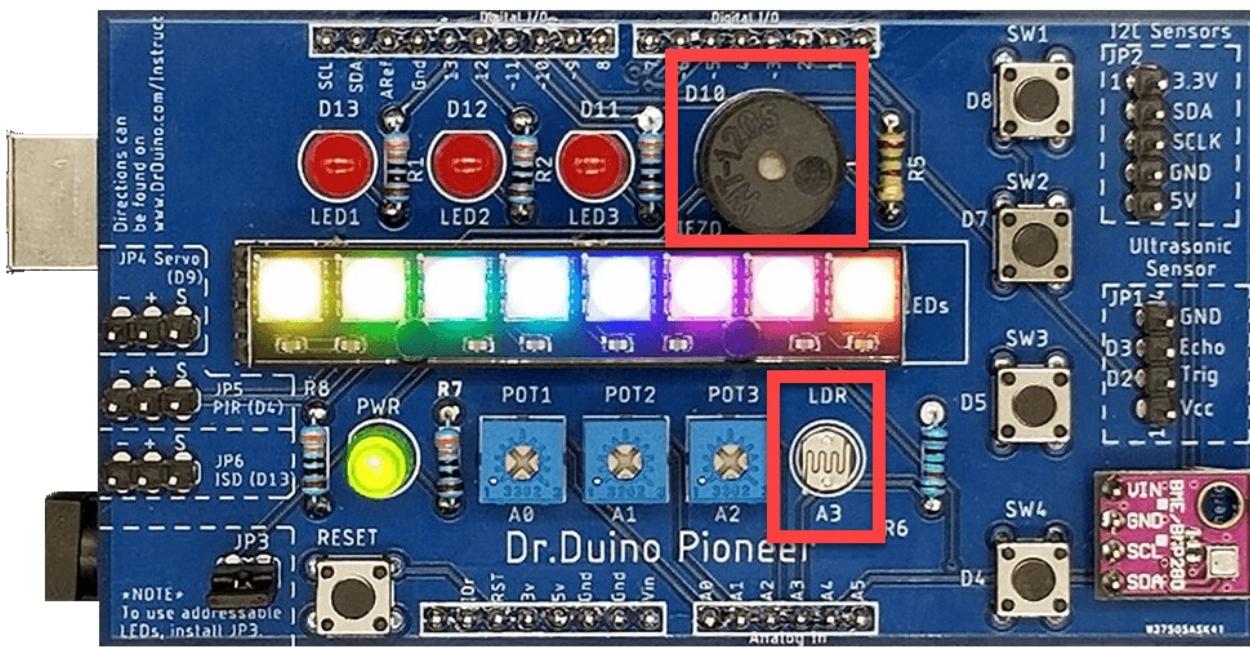
At that time, you will have the pleasure of hiding the box again.

Oh so satisfying 😊

With that said, Happy Annoying !!

---

## ANNOY-A-BUDDY | HARDWARE USED ON PIONEER AND EXPLORER



---

## ANNOY-A-BUDDY | HOW IT WORKS

After you download the Annoy-A-Buddy sketch into your Pioneer or Explorer the fun begins.

- 1) To test it, cover the LDR sensor and wait up to 2 minutes to get a sample of the annoyance you will soon unleash. Mwaahhaahhaaa!
- 2) NOW, find your unsuspecting victim which you want to annoy (teehee).
- 3) Place the entire unit in a spot where they spend a lot of time. But just make sure it's in a dark spot.
- 4) WALK AWAY and get ready to start laughing. It's programmed to give you a 30 second head start before it starts to beep.

A few things to note with this one, you will want to put the entire unit into a plastic container; we do not want anything shorting out.

You will also need to provide your own power, if you are near a computer anyway, then that's not an issue.

But if you're not, then you'll need to power this from a battery.

The VIN pin is perfect for a 9V battery and will annoy someone for a really long time!

## NIGHT LIGHT

This sketch uses the on board LED stick, three potentiometers and the built in LDR (light sensor).

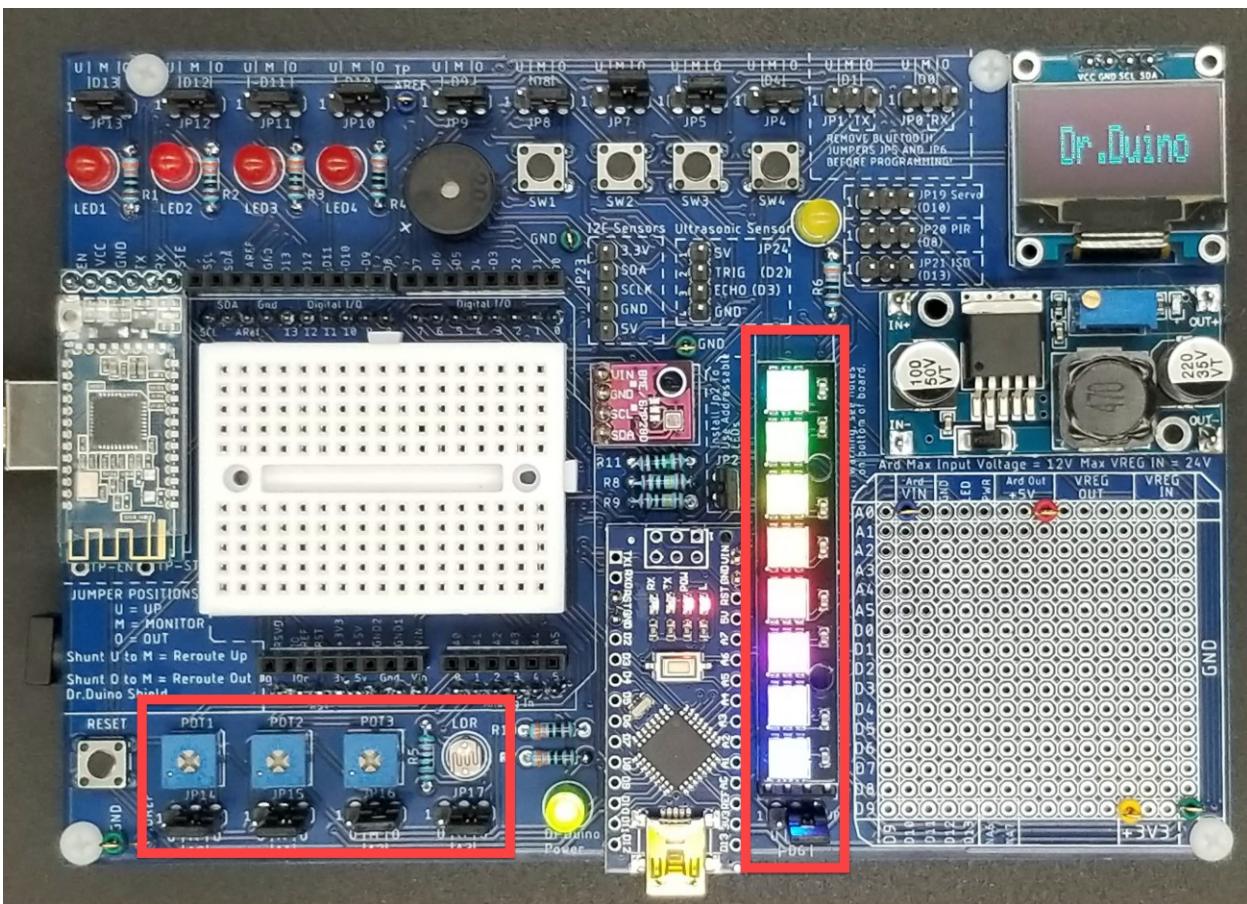
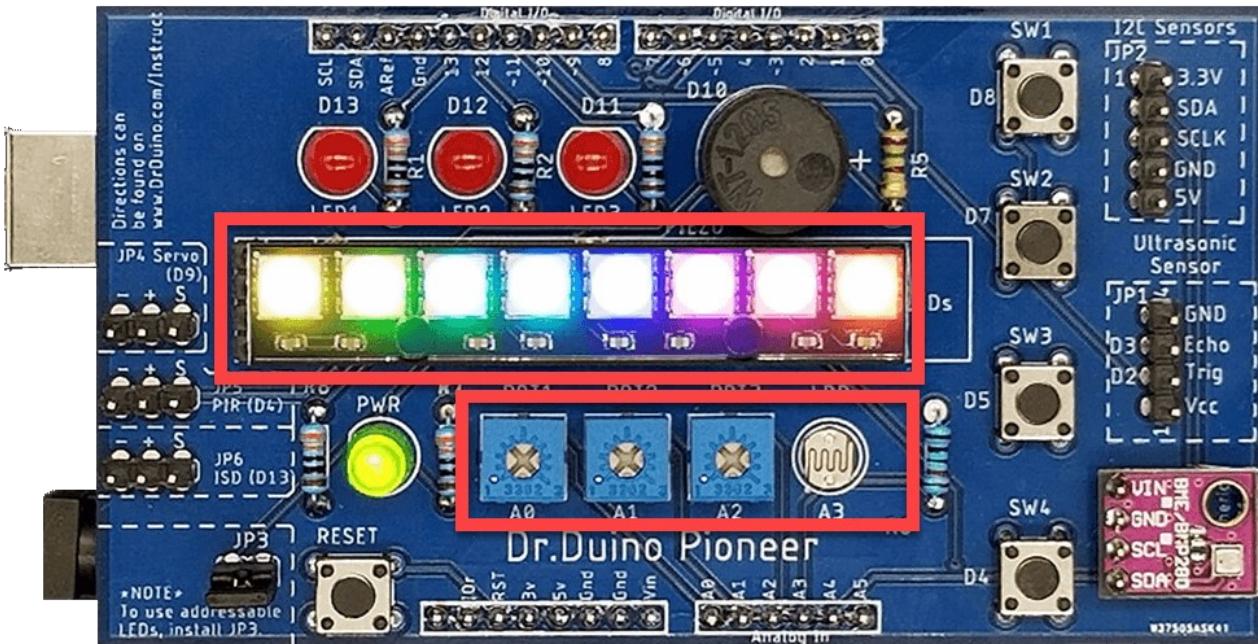
I love this little simple sketch because it easily turns your Pioneer or Explorer into midnight run, loo lighting machine!

Or... if you have little ones, a monster combatting light keeper.

Something fun would be to have your little one choose the color they want at night, which can help get them excited for bedtime!

Let's dig into how it works.

## NIGHT LIGHT | HARDWARE USED ON PIONEER AND EXPLORER



---

## NIGHT LIGHT | HOW IT WORKS

This works similarly to the Larson scanner except we use the LDR to determine if the room it's in has the lights on or not.

If it's dark, it turns on the LED stick to the color you set via the pots.

If it's not, it automatically turns off.

The value in which the LDR determines if it is light or dark enough is adjustable in the code.

You may have to play with that value a bit to calibrate it to your particular room.

Have a look in the comments of the code for how to do that it's pretty easy.

## DISTANCE DETECTOR AND COLLISON WARNING SYSTEM

This sketch uses the ultrasonic sensor and your Pioneer or explorer kit.

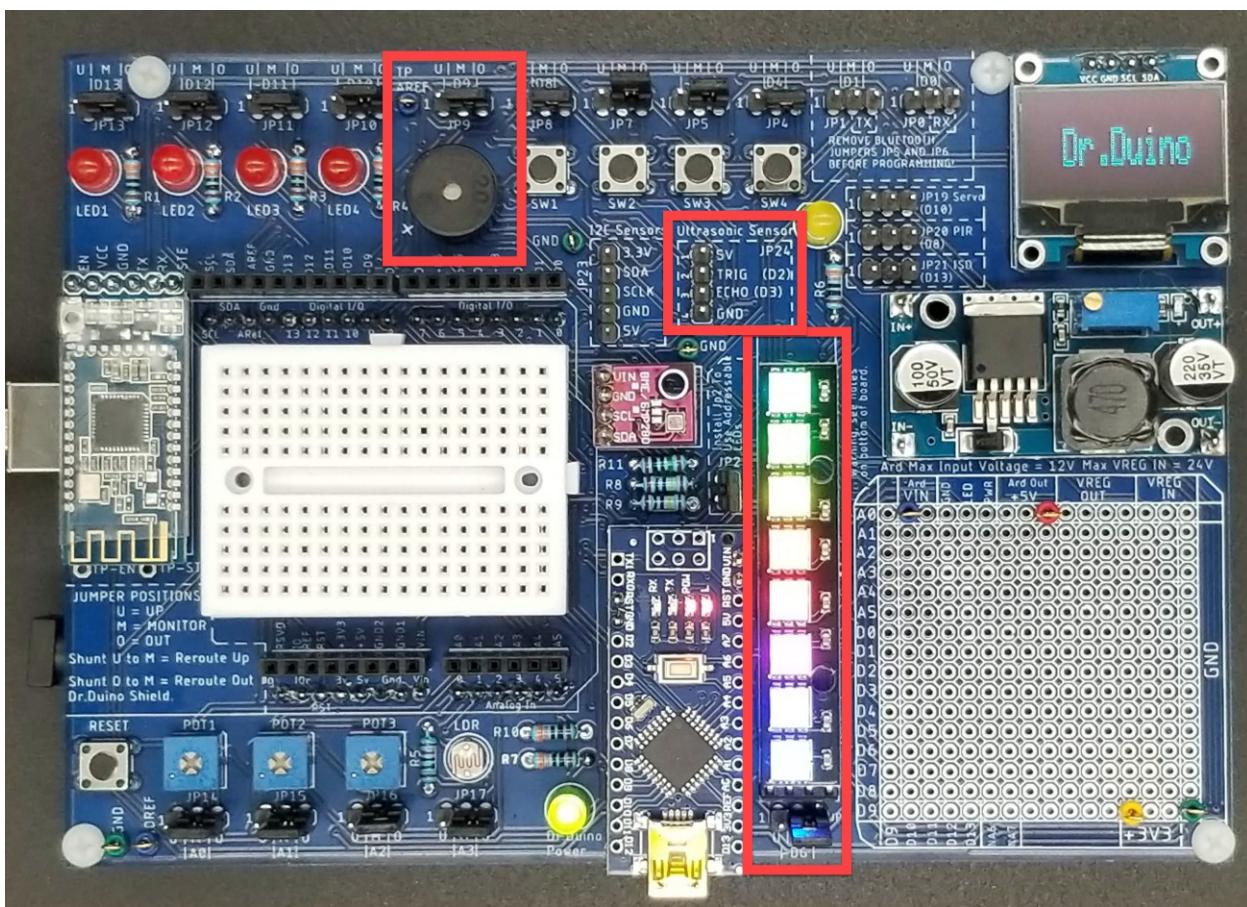
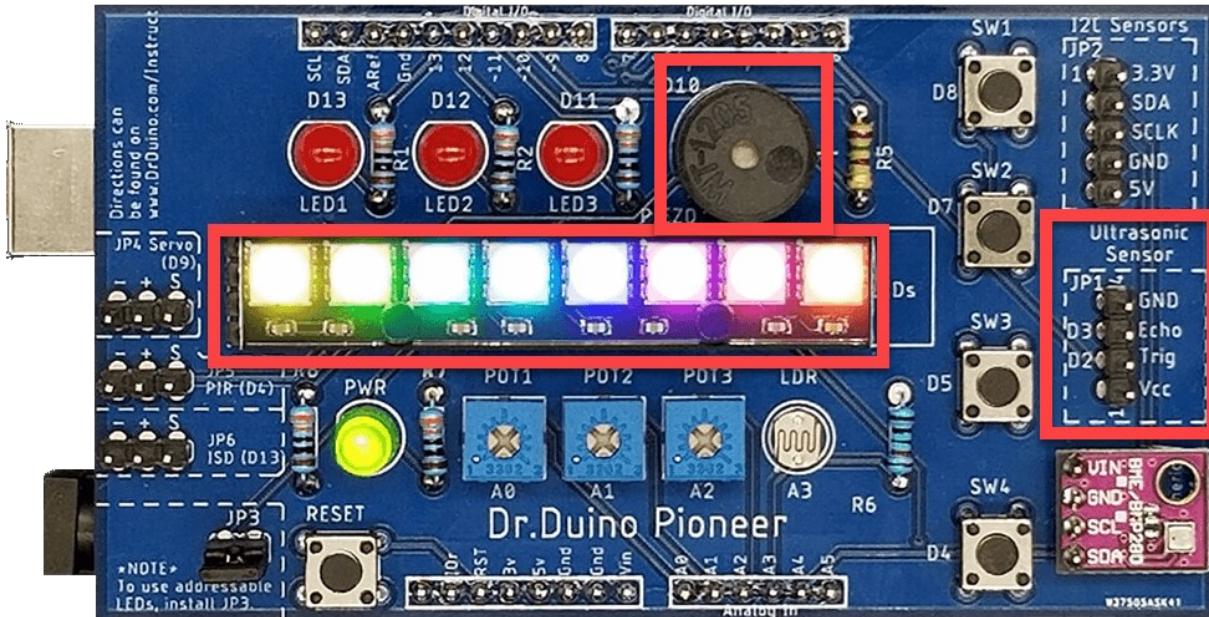
**Refer to the assembly manual for how to connect the ultrasonic sensor to your kit.**

This works on the same exact principle which modern cars use when backing up to determine how close/far they are from something.

As the object you are measuring moves closer to the ultrasonic sensor, it begins to warn you via various colors!

Let's dig in.

DISTANCE DETECTOR AND COLLISION WARNING SYSTEM | HARDWARE USED ON PIONEER AND EXPLORER



---

## DISTANCE DETECTOR AND COLLISION WARNING | HOW IT WORKS

- 1) Attach the Ultrasonic Sensor to the Ultrasonic sensor port on your hardware.
  - a. This is detailed in the assembly manual.
- 2) After downloading the code, point the ultrasonic sensor at whatever object you want to measure your distance from.
- 3) Next turn on serial monitor in the Arduino IDE.
- 4) If the area is all clear, the LED stick will light up solid green. It can detect up to around 13 feet away.
- 5) As the object moves closer, the color of the stick will change to yellow, then solid red, then blinking red and finally, if you are less than 10" away from hitting the object solid white!
  - a. As you back away from the object, the same happens just in reverse.

The distances which trigger the different colors lights is completely configurable in the code too.

Note! If you find that the LED stick is not lighting up at all, its because POT 1 on the Explorer or Pioneer is set to 0. It's using that to determine how bright or dark to make the lights. If this is the case, just pump up the volume!!

## RGB-CONTROLLER WITH RECALL FUNCTION!

This sketch uses the on board LED stick, 4 switches, 3 pots and 3 red LED's to accomplish a pretty slick version of a standard RGB controller.

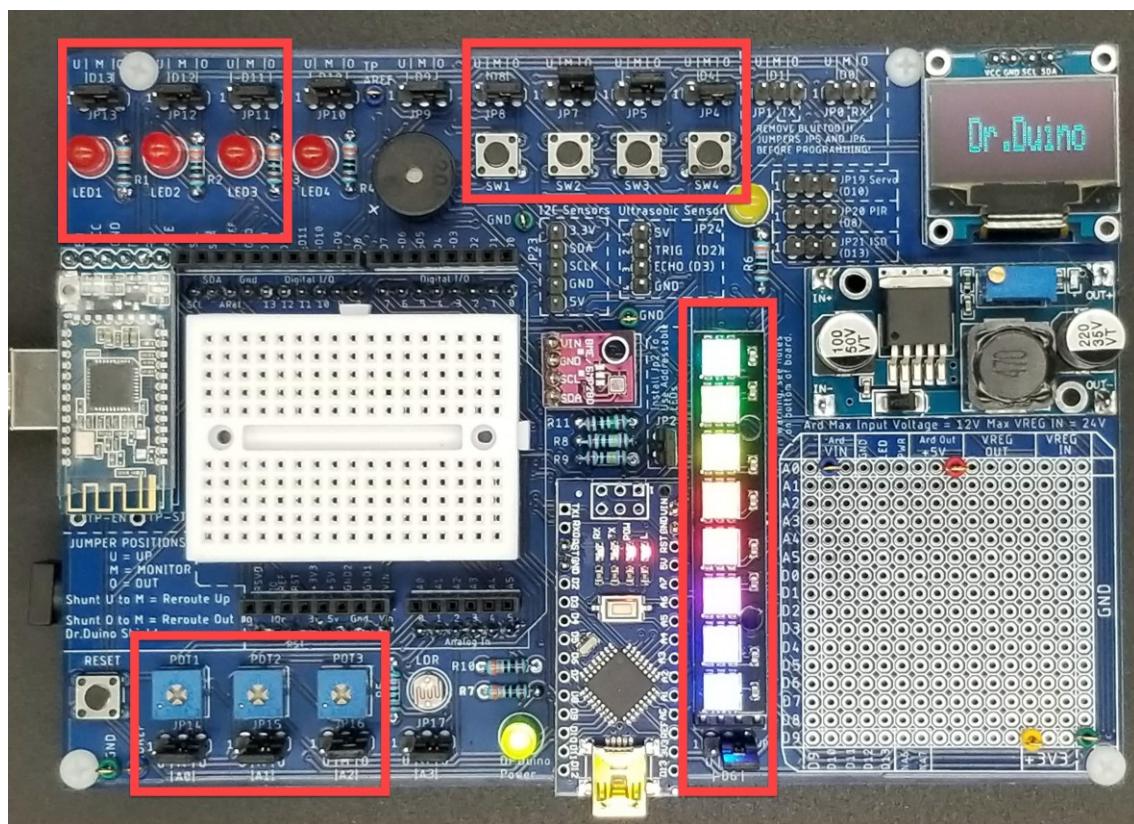
BUT!

Instead of just allowing you to adjust the colors via the pot you can actually store the colors in a type of nonvolatile memory called EEPROM (Electrically Erasable Programmable Read Only Memory).

We use the EEPROM to store any color of your choice, even when power is off!

How cool is that?

RGB CONTROLLER WITH RECALL FUNCITON | HARDWARE USED ON PIONEER AND EXPLORER



After uploading the sketch for the first time, it will look to see if there are any stored colors currently in EEPROM.

If there are, it loads the first one (more on this later), if not, it defaults to red.

If you press SW1, 2, or 3 you will see that the LED stick will turn off.

This is because we haven't stored any colors in EEPROM yet.

So, with that, lets store some colors!

- 1) First press SW4, you will see the red LED1 light up. This tells you it's in programming mode.

At the same time, all three potentiometers are being read and the values are being pumped out to the LED stick.

POT 1 is Red, POT 2 is Green and POT 3 is Blue.

- 2) Start turning each pot one by one until you get to a color you desire.
- 3) Now press SW1, this automatically stores the color you just created into EEPROM and maps it to SW1.

Do the same thing for SW2 and SW3.

LED 1 will still light up, but when you press SW2, LED 2 will also light up for a moment indicating that you stored it to SW2.

With SW3, LED 3 will flash.

- 4) Now just press SW1, or 2 or 3 and you will see that the colors that you created are automatically recalled and pushed out to the LED stick.

Remember when I said that its also stored in EEPROM?

Well, here is how you test it.

- 1) Unplug your computer from your kit, wait a few seconds and then plug it back in.

Upon startup, it goes and looks if anything was saved, if it was, then it pumps out that color to the LED stick instead of the default red.

- 2) You can now click away at SW1, 2, or 3 which will recall from EEPROM the various colors.

I did one more cool thing here...

Let us say you want to clear out the colors you stored in EEPROM.

Easy peasy... just press and long hold SW4.

You'll know when it's done clearing the memory when all the red led's blink.

Then you can set up all of your colors again!

The applications here are pretty infinite, let's say you wanted to put some lighting in your car, will you could do this same technique just with a longer LED strip.

You could also add more switches so that you could map more colors.

It's really limitless!

## PREMIUM PROJECTS



---

#### SELECTING PREMIUM PROJECTS

If you purchased the premium package, you'll have access to 5 additional Dr.Duino™ projects.

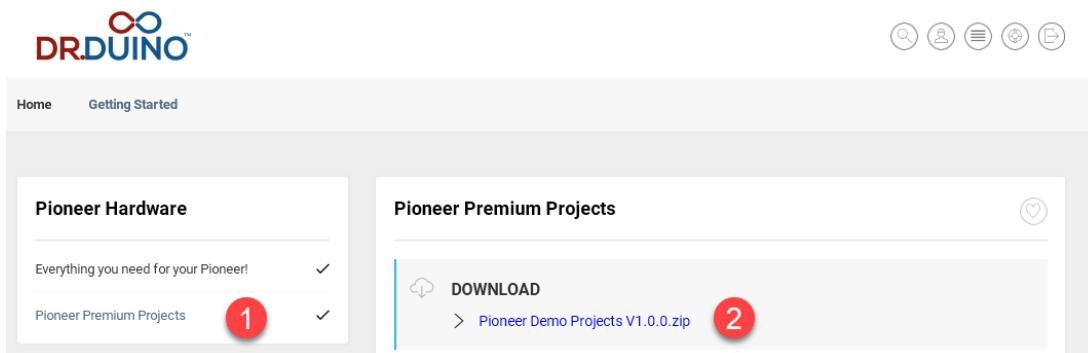
You can gain access to them in your Dr.Duino's labs account.

## [www.DrDuino.com/Labs](http://www.DrDuino.com/Labs)

- 1) After you sign in, if you purchased the premium package, navigate to Getting Started and then select whichever hardware platform you purchased.



- 2) Then navigate to the premium projects section and download the zip file.



If you did not purchase the premium projects and would like to, you can upgrade your order here.

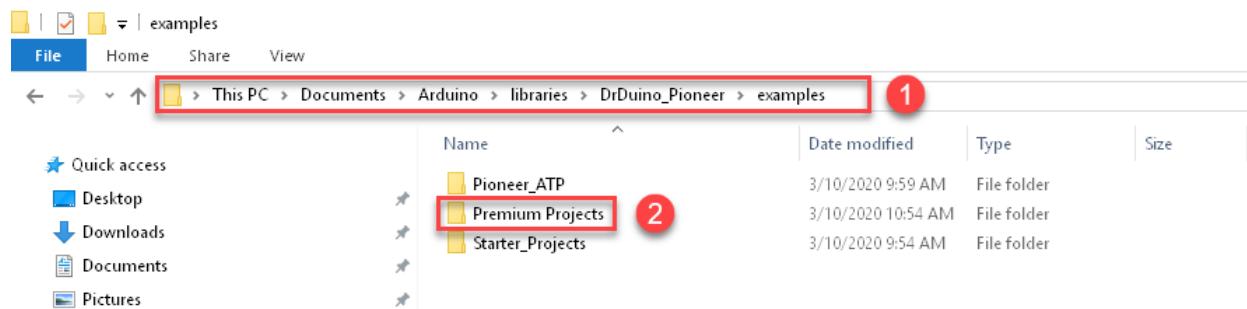
## [www.drduino.com/PremiumSketches](http://www.drduino.com/PremiumSketches)

## INSTALLING PREMIUM PROJECTS

- 1) Extract the entire contents of the zip file into the following location.

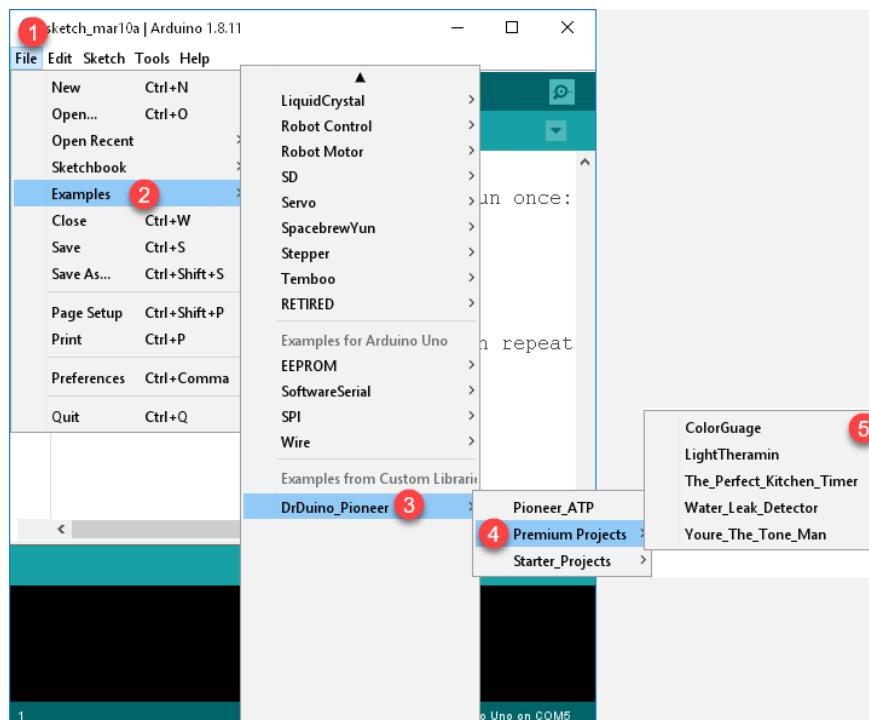
Please note that the location is critical.

The following example is for the Pioneer, follow the same process for the Explorer except you will place it in the DrDuino\_Explorer folder.



- 2) Restart your IDE if you currently have it open.

- 3) Navigate to your shiny new Premium Projects and open whichever one you prefer!



## COLOR GUAGE

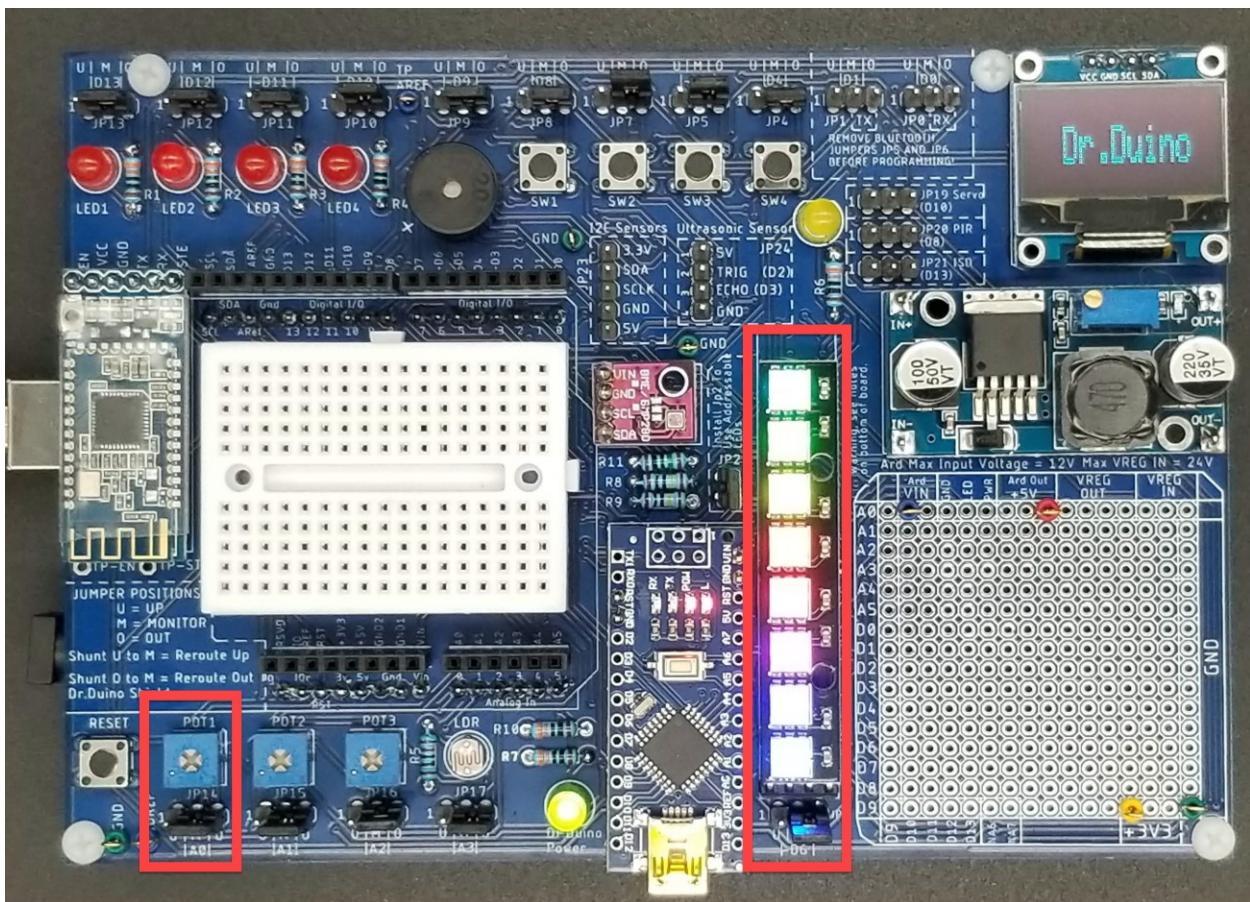
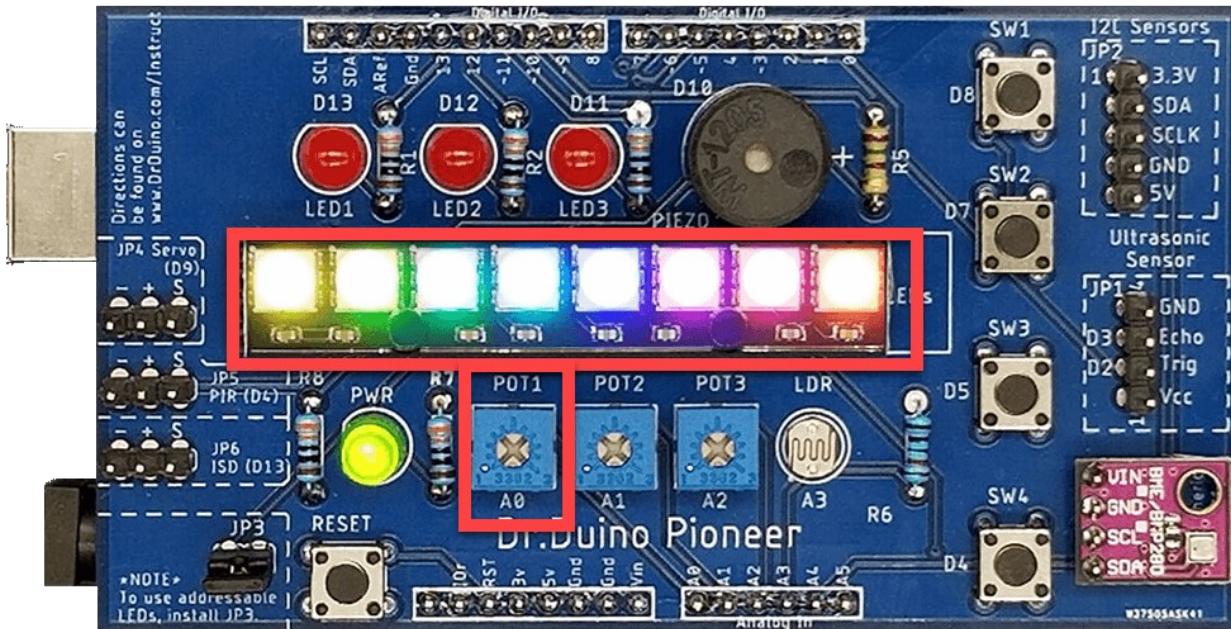
So much of what we want to do with our Arduino projects requires notifying a user that something is or has happened.

Imagine if you wanted to measure how much oil you have left in your oil tank without that silly gauge they currently give you.

Something, which you could clearly see from across the room with a color gradient that instinctively lets you know how much fuel is left.

Well, that's what this will do.

## COLOR GAUGE | HARDWARE USED ON PIONEER AND EXPLORER



## COLOR GUAGE | HOW IT WORKS

After you download the sketch, the microcontroller begins reading the value from Pot 1.

This will simulate how full or empty your oil tank is.

Turning it all the way left is empty, all the way right is full.

You'll notice that as you turn it to the left, the number of lights will decrease while turning it to the right will increase the number of lights.

The lights follow a color scheme as you increase or decrease the pot.

When you decrease it, the green begins to turn off, then the yellow, then the orange and finally the red turns off.

When you increase it, the reverse happens.

This color gradient is handy, especially in this example.

Imagine being across the room and only seeing the red and orange lights being on.

You would know "Utoh... running out of oil, better order some soon".

Here is the color scheme assigned to each LED.

Note LED 1 is located closest to the 4 pin header you attached to the LED stick.

LED Number	Color
1, 2	RED
3,4	Orange
5,6	Yellow
7,8	Green

In the code, you will be able to change the colors easily.

This allows you to make any color combination you like.

In the next project, we will give another example of using this gauge!

## THE PERFECT KITCHEN TIMER

Timers, timers, everywhere, but none do what I want them to, show me how much time is left visually!!!

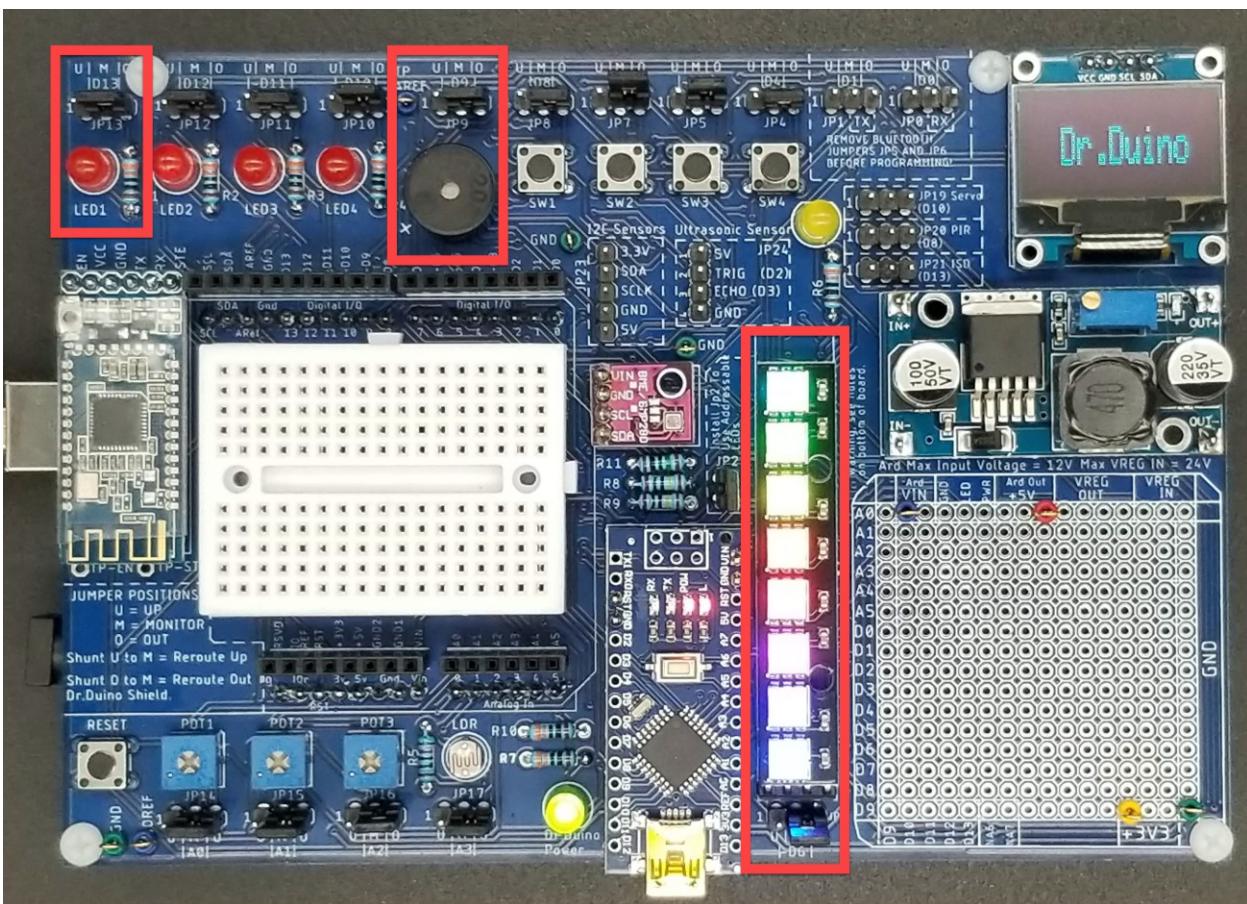
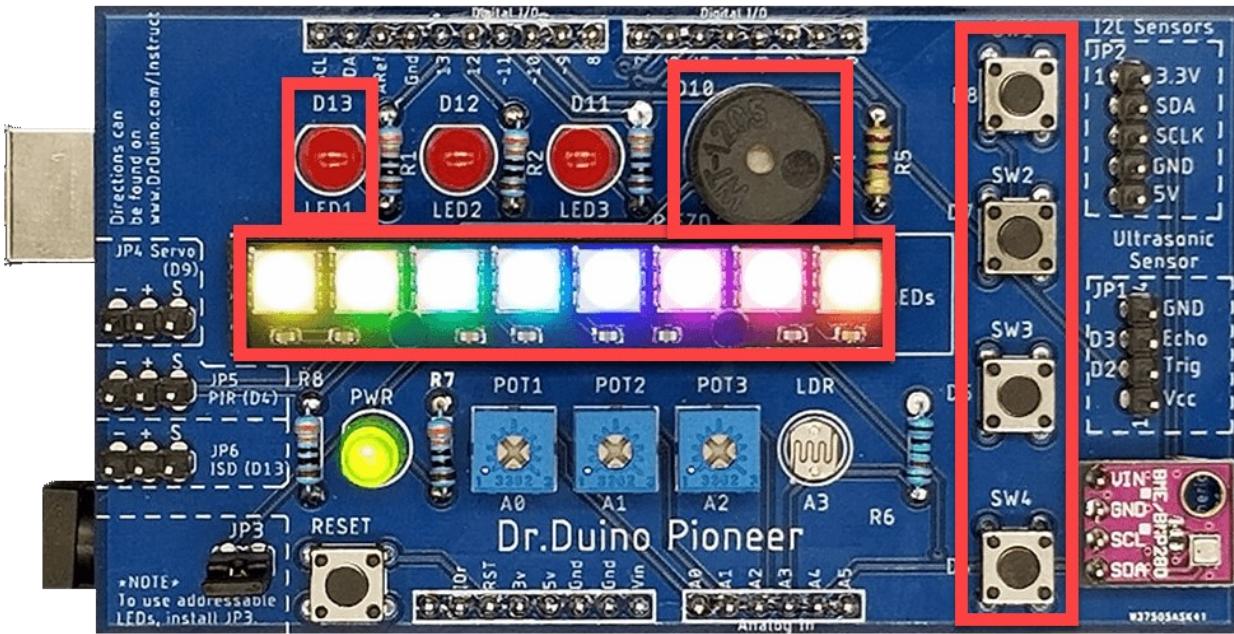
If you haven't noticed by now, I love blinky flashy lights!!!

That's why I thought a custom timer would be a perfect project.

This timer would allow you to see how much time is left without reading some small digits on a display.

Let's dig in!

THE PERFECT KITCHEN TIMER | HARDWARE USED ON PIONEER AND EXPLORER



---

## THE PERFECT KITCHEN TIMER | HOW IT WORKS

After you download the sketch, the microcontroller begins scanning for button pushes.

Here are what the switches do!

Switch Number	Function
<b>SW1</b>	Each button press, adds 1 second or minutes to the timer. You select in code which you prefer. Its defaulted to minutes.
<b>SW2</b>	START! This kicks off the timer.
<b>SW3</b>	PAUSE! This pauses the countdown no matter where you are in the count.
<b>SW4</b>	RESET! Resets the entire timer allowing you to re-enter how much time you want to set your timer for.

After you set the desired time, just press SW2 and sit back!

LED1 will begin to toggle every second on and off.

I call this the heartbeat LED, I use it often in a ton of projects.

The entire LED stick will also light up but in a specific color order.

LED Number	Color
1, 2	RED
3,4	Orange
5,6	Yellow
7,8	Green

This makes it super simple to see how much time is left before the timer expires.

As the time runs out, the LEDs will begin to shut themselves off starting from the Green, then yellow, then the orange and finally the red.

Once time is out, you'll also get a nice audible notification!

Ok, now it's time (pardon the pun) for you to start timing stuff!

Go on now, boil some eggs, let someone know exactly how much time they have left in the shower, or when it's time to wake up.

The choice is yours, ENJOY!

## WATER LEAK DETECTOR

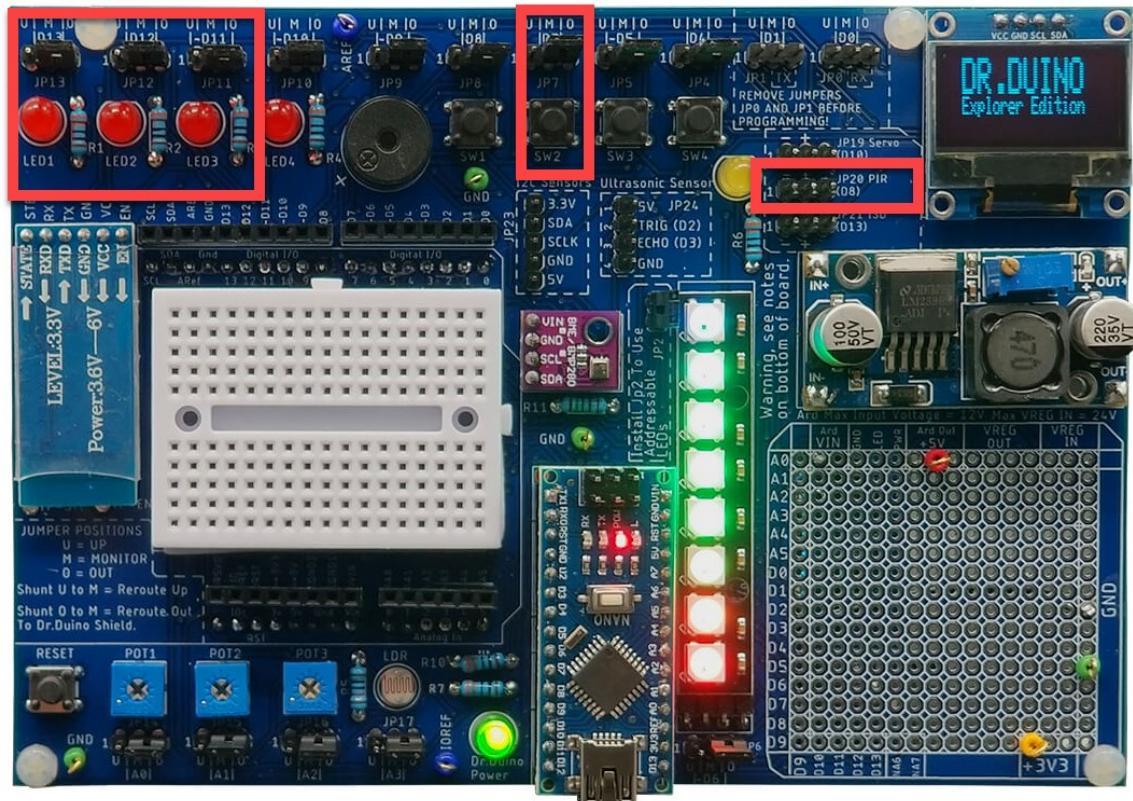
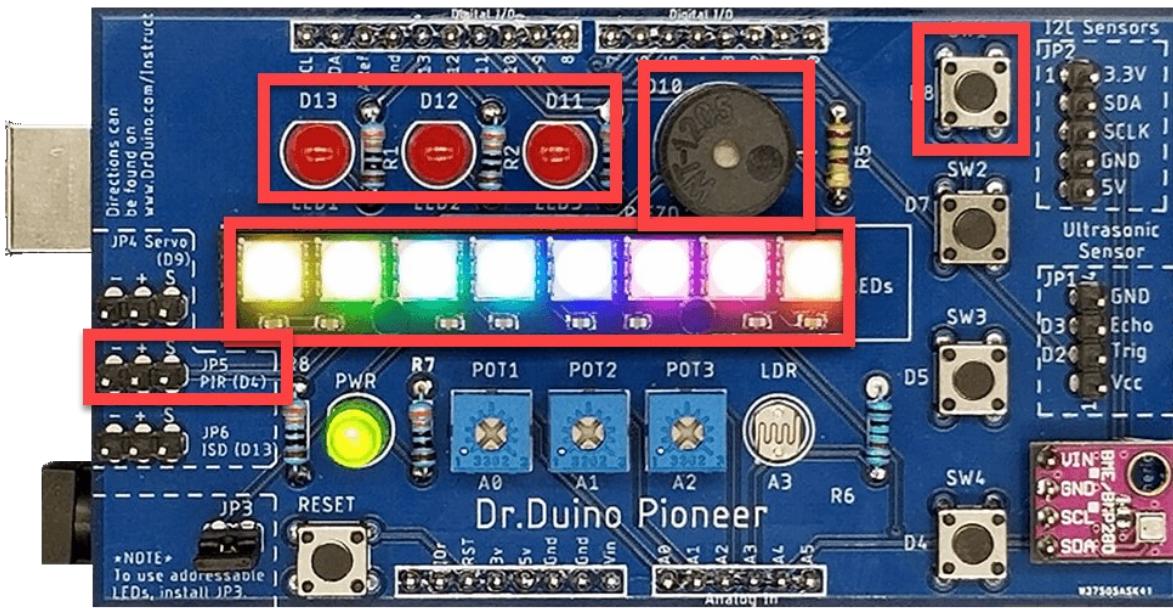
Ever worry that your boiler, sink or washer machine will start to leak but you won't know till its waaaaaaay too late?

I know I have, which made me think... hmmmm... can I turn the Pioneer or Explorer shields into a

- 1) Water sniffing,
- 2) "Hey you, something is wrong"
- 3) AND... I won't stop until you acknowledge that I detected water, MACHINE!

In a nut shell... that's exactly what this sketch will do!

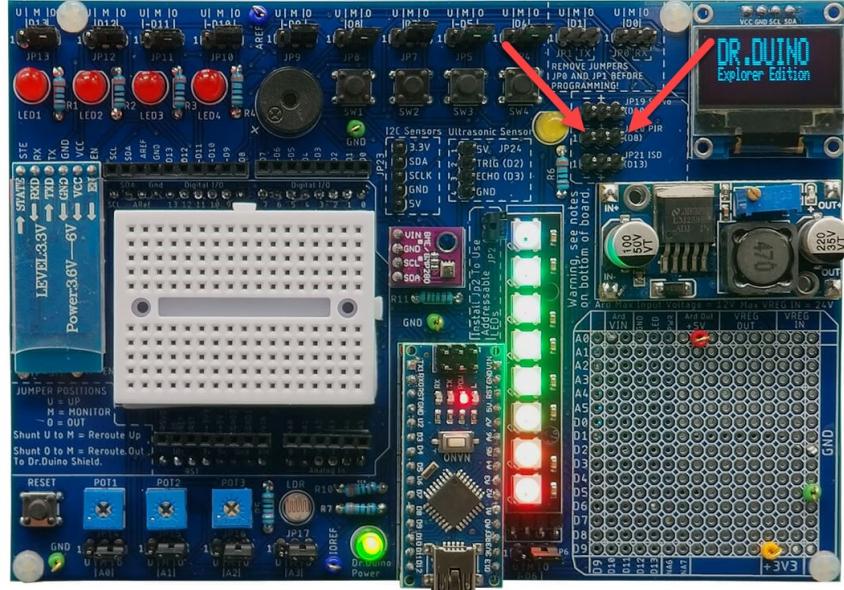
WATER LEAK DETECTOR | HARDWARE USED ON PIONEER AND EXPLORER



---

## WATER LEAK DETECTOR | HOW IT WORKS

1. Grab yourself a bowl and add a little bit of water to it. Be careful not to spill water on your kit. Bad things will happen, including but not limited to damaging your unit.
2. Then add 2 wires to the following locations on your Pioneer or Explorer. The DuPont wires that came with your kit are perfect for this.



3. Now, dip both wires into the water and watch what happens!
4. As soon as both wires hit the water you will hear beeping, and all of the discrete red LEDs will blink just to make sure its extra visible.
5. It will not stop doing this until you actually acknowledge it. You do this by pressing SW1 on the Pioneer or SW2 on the Explorer.
6. The system is now reset and you can dip the wires again!

The reason why we are able to detect water is actually pretty simple.

The signal side of the wire you just attached is connected to one of the microcontrollers input pins.

We then turn on the internal pull-up resistor for that pin.

This makes it so that this pin is always a digital 1 or high.

The ground or (-) signal is connected to the boards ground.

When you dip both wires into the water, you are effectively creating a path to ground.

This is actually a logic 0 or low.

Essentially, you just created a switch that switches on and off when it touches water!

How awesome is that!!

## YOU'RE THE TONE MAN

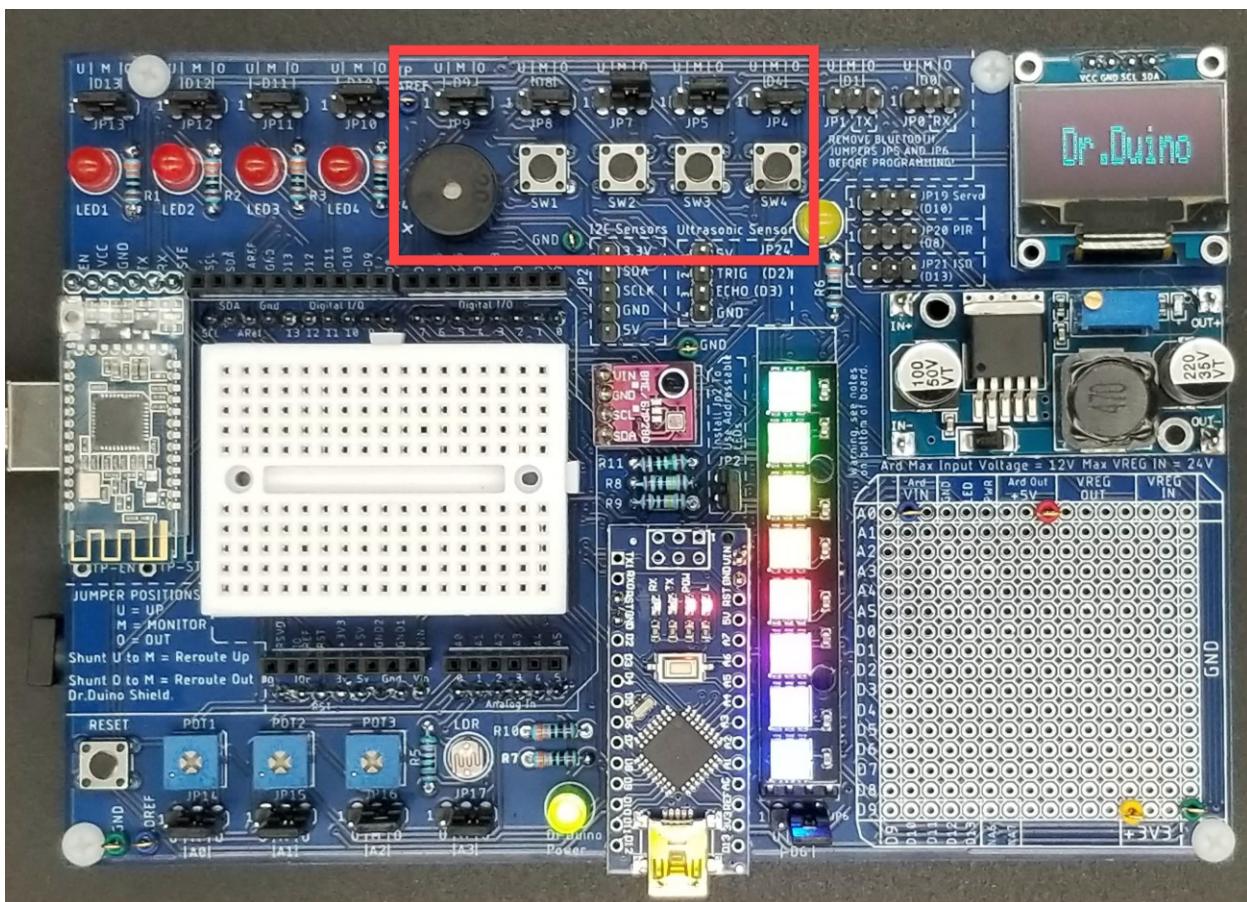
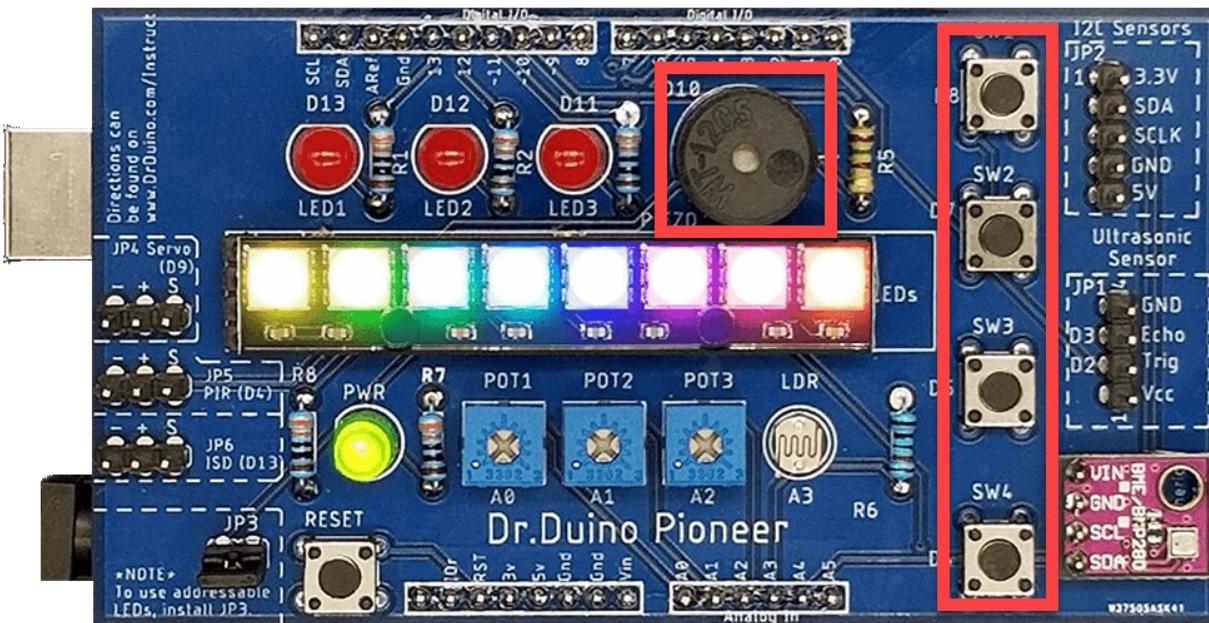
Ok, the name of this sketch is a bit of a play on words.

There is a song called "Scatman", google it you'll see what I mean.

Any who... getting back to the sketch.

This simple piece code will turn your Pioneer or Explorer into a beeping and booping machine.

YOU'RE THE TONE MAN | HARDWARE USED ON PIONEER AND EXPLORER



---

## YOU'RE THE TONE MAN | HOW IT WORKS

After you upload the code to your board, just start smashing SW1, 2, 3 and or 4.

Each one is assigned to specific notes but you can change that easily in the code.

You can assign any of up to 88 different notes to each of the 4 buttons on your kit.

Every time you press any of the four buttons, it will start making that tone.

Keep it depressed, and it will continue making that tone.

Press two or more buttons at the same time and well... it will try to play both tones but will be a bit muddled.

However, still fun!

If you want to assign different notes to each of the four buttons, just look in the code, I show you exactly how to do it.

Happy Music Making!

## LIGHT THERAMIN

If you have never seen or heard of a Theremin before, you're not alone.

It is an odd but also very cool instrument.

It looks like an old-time radio with a long antenna sticking out of it.

One of the major manufacturers of it is a company called MOOG.

Here is what one looks like:

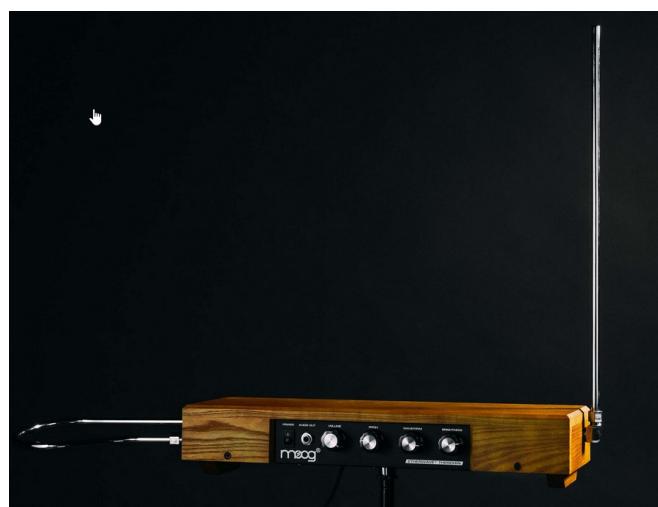


Photo credit- [www.moogmusic.com](http://www.moogmusic.com)

You play this device by moving both hands close to or far away from each of the antenna.

One is responsible for pitch and the other for volume.

It makes a really unique whining sound.

Here is a link to a video of one if you're curious about what one sounds like.

<https://youtu.be/PjnaciNT-wQ>

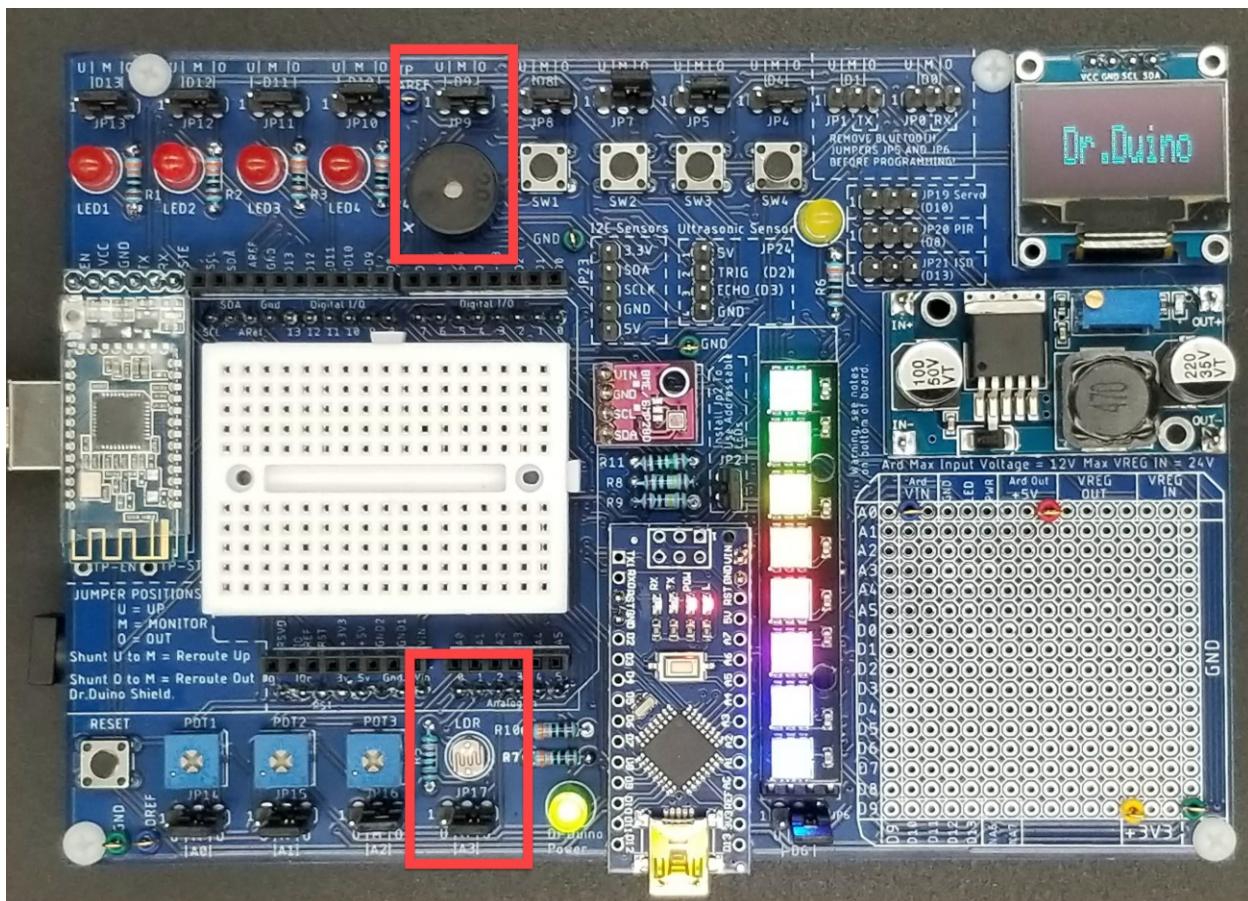
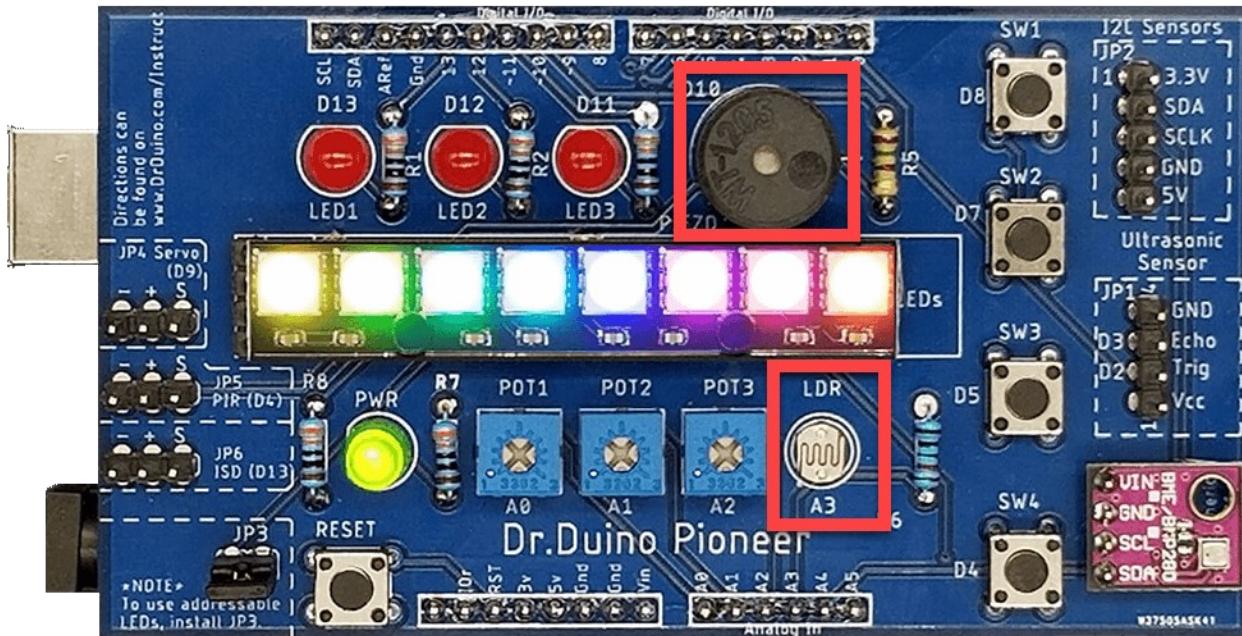
The way it operates is actually very simple.

It's measuring the electromagnetic field around each of the antenna which changes when your hand moves to and from it.

It then translates this into an audible sound.

We can do the same thing with your Pioneer or Explorer, except we will use light instead of an antenna!

## LIGHT THERAMIN | HOW IT WORKS



1. Download the sketch to your board.
2. After its done downloading, you will immediately begin to hear a kind of squealing.

But don't fret, that's because it's expecting a brighter light.

3. Grab yourself a flashlight. Something bright like your cell phone and move it up and down from the LDR. Have a listen to what happens now!

This one is a lot of fun to play with!

The code is setup so that you can modify it too with different frequencies.

Play around and see what sounds you can create!

## CLOSING THOUGHTS

Well... that is it; those are your Pioneer and Explorer demo sketches.

I will be regularly updating them and/or making new ones.

So make sure to check back inside of Dr.Duino's labs often.

I hope that you really enjoyed and learned a ton from these 10 demo projects.

I really wanted to create something more than simply blinking an led which most kits provide you with.

If you want more, I do offer the first of many expansion kits, which are more in depth and do waaay cooler stuff then these first 10 demos.

If you did purchase it, the directions on how to use it is inside Dr.Duino's Labs.

If you did not, but are interested here is a direct link.

[www.DrDuino.com/Expansion](http://www.DrDuino.com/Expansion)

### QUICK REMINDER...

If you are looking for a stellar Arduino hands on programming course, you have access to a really amazing one.

It's from Programming Electronics Academy or PEA for short.

There is a Free and Paid version of the course. This is hands down the best way to learn and comes with tons of videos!

Since you're a Dr.Duino customer, you get 20% off the paid course too.

**Just use coupon code DRDUINO at checkout.**

[Here is a direct link to the course for you to sign up.](#)

<https://electronics.ontraport.com/t?orid=124&opid=39>

## **SUPER IMPORTANT!**

To sign up just enter your email on his site, just don't use a comcast email. For some reason, comcast blocks his emails from getting through.

Use something like a gmail or yahoo or something similar.

After you've gone through the video courses, you'll have the skill set you need to make your pioneer do whatever you want.

Of course, over time, I'm going to be adding new cool features to it as well.

So do keep an eye out on Dr.Duino's labs.

### **FINAL THOUGHTS**

If you have any feedback both positive and negative please shoot me an email at [info@DrDuino.com](mailto:info@DrDuino.com).

This email address is only for feedback so to make sure I see it please type in the subject line:

Demo Code Feedback (Name of the Sketch)

Note- If you do not I will not see the email.

With that, I leave you to your ones and zeros!

Talk Soon,

Guido Bonelli

Founder of Dr.Duino™