



runlinc Online Project A2: 2 Line ON/OFF Buttons (runlinc Online Version)

Contents

Introduction.....	1
Part A: HTML for web page text and graphics - dashboard.....	2
Part B: JavaScript to program the control, sensing, data logging actions.....	3
Challenge.....	4
Extension.....	5
Summary.....	6

This project uses runlinc online which can be found here:

<http://www.runlinc.com/online/control.html>

Make sure to right-click the link and click 'open in new tab'.



Check that your board IP says <http://runlinc.com> as shown to the right. If it does not try clearing your site data.

This project does not require a runlinc Wi-Fi Inventors' Kit.

Introduction

Problem

We want to be able to remotely control a security light over Wi-Fi.

Background

Two buttons on a web page can be used to control anything anywhere, like lights on your porch so you can see, or scare off intruders. Using runlinc we can program a microchip to remotely turn on a security light.

Ideas

As we are using runlinc online, we can't use any physical lights, what kind of things could we use instead that we can display on a webpage? Should we be able to turn it off?

Plan

We will create on and off buttons on the website to control a lamp. We will use an image of a lamp in place of a physical light, which we would use if we had the runlinc Wi-Fi Inventors' Kit.

runlinc Background

runlinc is a web page inside a Wi-Fi chip. The programming is done in the browser and sent to the chip over Wi-Fi. The runlinc web page inside the Wi-Fi chip will command the microchips to do sensing, control, data logging Internet of Things (IoT). It can predict and command.

Part A: HTML for web page text and graphics - dashboard

In the HTML we need to create a heading for the page, two buttons to turn the lamp on and off and display an image to represent the lamp. The following code is the HTML we will be using, make sure you type it into the HTML box.

```
<h1>My Control Web Page</h1>
<button onclick="lampON();" ><h1>Lamp ON</h1></button>
<button onclick="lampOFF();" ><h1>Lamp OFF</h1></button>
<br>
<br>

```

The first line is our heading, the `<h1>` tag tells the browser that the text should be the largest heading format. The next two lines are our buttons. They have labels which use the `<h1>` tags again to make the text more prominent. The most important part of the buttons are the `onclick` attributes. They determine the function that will be executed when the button is pressed. We will be writing these functions in JavaScript.

After the buttons, we have a space (line breaks, the `
` tags) and then a picture of the lamp. Two pictures are stored on `runlinc.com` which are accessed by putting their link (URL) in the `src` attribute.

Now we need to go back to the CSS box above the HTML and add some CSS so that the buttons are large enough to fit the H1 text. (On some devices the button sizes are fixed unless they are changed in CSS, so the large text doesn't fit). We will set a height of 75 pixels and a width of 175 pixels.

```
button
{ height:
  75px; width:
  175px;
```

Part B: JavaScript to program the control, sensing, data logging actions

Now we need to write the JavaScript which our buttons will execute when they are pressed. We need two functions, one to turn the lamp on, and one to turn it off. This code needs to go in the 'JavaScript' box.

```
function lampON(){
  document.getElementById("img").src="http://runlinc.com/online/picLampON.jpg";
}

function lampOFF() {
  document.getElementById("img").src="http://runlinc.com/online/picLampOFF.jpg";
}
```

These are the two functions that we need. One lampON function to change the image to show a picture with a lamp on, and one lampOFF function to change the picture to 'off'.

The functions use the 'img' ID as a reference to the image in the HTML.

When writing JavaScript functions remember to insert your lines of code between the curly brackets.

runlinc V1.1 Copyright and International Patent Pending. All rights reserved.

File

Load File Save

Run Code Stop Code

Board

Send Get

Board IP:

STEMSEL

PORT	CONFIGURATION	NAME	STATUS
A3	DISABLED	<input type="text"/>	
B4	DISABLED	<input type="text"/>	
B6	DISABLED	<input type="text"/>	
C0	DISABLED	<input type="text"/>	
C1	DISABLED	<input type="text"/>	
C2	DISABLED	<input type="text"/>	
C3	DISABLED	<input type="text"/>	
C4	DISABLED	<input type="text"/>	
C5	DISABLED	<input type="text"/>	
C6	DISABLED	<input type="text"/>	
C7	DISABLED	<input type="text"/>	
GPIO0	DISABLED	<input type="text"/>	

CSS

```
button {
  height: 75px;
  width: 175px;
}
```

HTML

```
<h1>My Control Web Page</h1>
<button onclick="lampON();" ><h1>Lamp ON</h1></button>
<button onclick="lampOFF();" ><h1>Lamp OFF</h1></button>
<br>

```

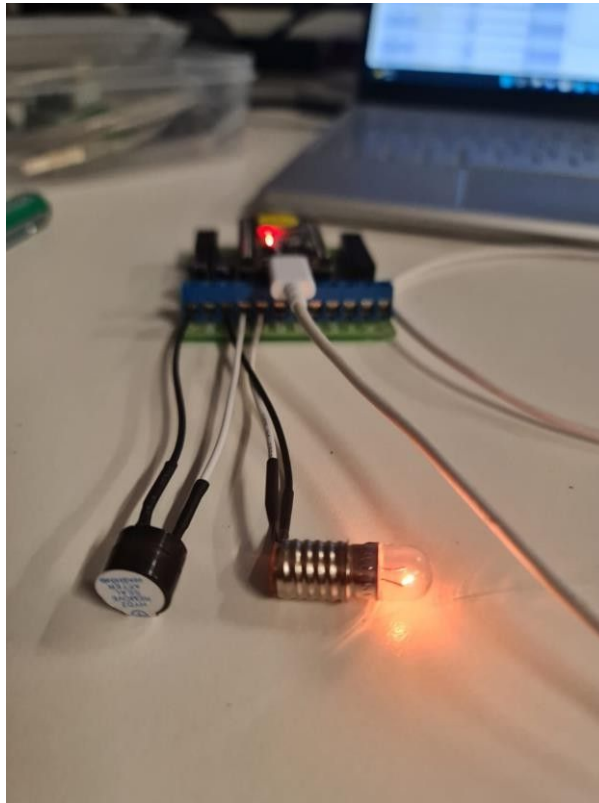
JavaScript

```
function lampON(){
  document.getElementById("img").src="http://runlinc.com/online/picLampON.jpg";
}

function lampOFF() {
  document.getElementById("img").src="http://runlinc.com/online/picLampOFF.jpg";
}
```

JavaScript Loop

Figure 1: final CSS, HTML and JavaScript



Challenge

Let's change our web page to use a circle to represent the lamp. We will use HTML to create the circle, and JavaScript to change the colour (light grey when the lamp is off, and yellow when it is on).

Replace the image from your HTML code with the following three lines, which will create a graphic of a circle:

```
<svg>  
<circle id="lamp" cx="175" cy="75" r="50" fill="lightgrey" stroke="black"  
stroke-width="5px">  
</svg>
```

This circle will represent our lamp. We have given the circle an x position (cx) of 175, a y position (cy) of 75, a radius of 50, a light grey fill so it appears off, and a black border (stroke) with a width of 5 pixels.

Change the JavaScript code so, instead of changing the src attribute to a different image URL, we change the fill attribute, which will change the fill colour of the circle.

```
function lampON()  
{ document.getElementById("lamp").style.fill = "yellow";  
}
```

You need to write the lampOFF function yourself. It needs to change the fill back to what it was before the lamp was turned on, so it looks like it's turned off. (Make sure the ID in you JavaScript matches the ID on the HTML circle).

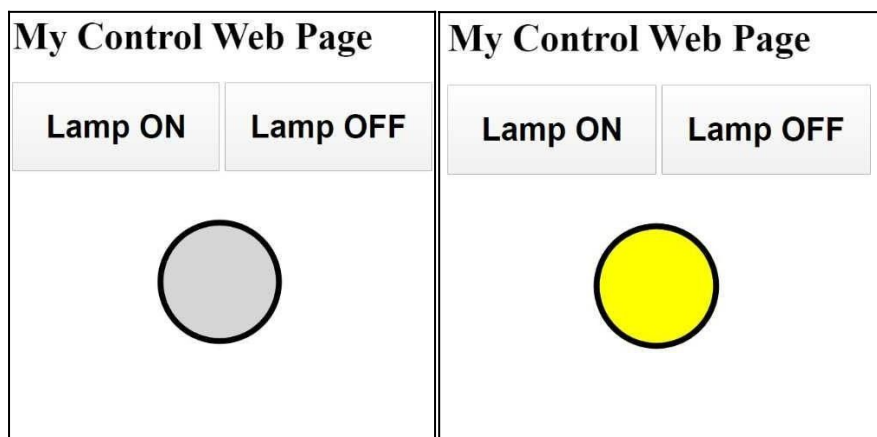


Figure 5: challenge webpage
(Lamp OFF: left, Lamp ON: right)

Extension

We now have two buttons changing the fill of a circle to represent turning a lamp on and off, but it would be better if we only needed one button.

The first thing we need to do is delete one of the buttons from the HTML. The remaining button can be renamed to indicate that it turns the lamp on and off, and the onclick attribute will need to be changed to execute a new function that we will write.

If your button label appears as two lines on the web page you will need to make the button a little taller. Increase the button's height to 100 pixels in the CSS if you need.

Lastly in the HTML, we need to re-position the circle so it appears directly under the button (it will be off to the right if you haven't changed the position yet).

We don't need any of the previous JavaScript, so it can all be deleted.

Before we write our function we need a variable which stores whether the lamp is on or off.

```
var on = false;
```

Now we need to write a function for the button which turns the lamp on if it's off, and off if it's on. Set up the function and put the following code inside of it.

Hint – look back over the challenge section if you can't remember how to write functions.

Make sure you change the button's onclick attribute to your function's name.

```
if(on == true){  
  
} else {  
  
}  
on = !on
```

First, we have an if/else statement. This statement will check if 'on' is true, and if it is it will run the code between the first set of curly brackets. If it finds that 'on' is not true it will run the code between the curly brackets after the 'else'. The last line will reverse the 'on' variable (e.g. if it is 'false' it will become 'true').

We need two lines of code in the if/else statement. One to change the fill so the lamp is on, and one to change the fill so the lamp is off.

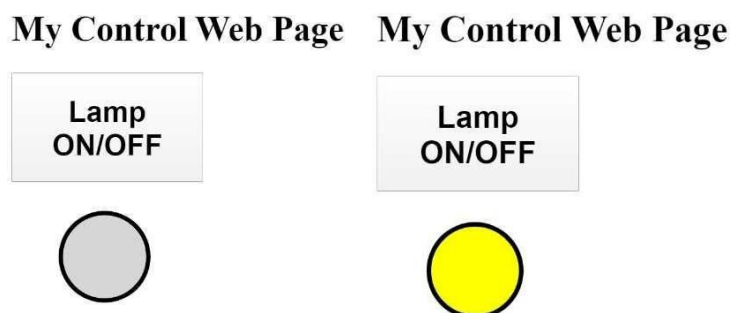


Figure 5: extension webpage
(Lamp OFF: left, Lamp ON: right)

Summary

People can use programming to tell microchips what to do. We can use them to turn on and off equipment, send signals to each other, all using buttons in HTML.