



# runlinc Project 9 I1: Set Fire Risk Limit (E32 Version)

---

## Contents

Introduction .....	1
Part A: Design the Circuit on runlinc.....	3
Part B: Build the Circuit .....	4
Part C: Program the Circuit .....	7
Expected Result .....	9
Summary .....	10

## Introduction

### Problem

- How can we measure something, such as fire risk?
- How can we change the fire risk setting without reprogramming?

### Background

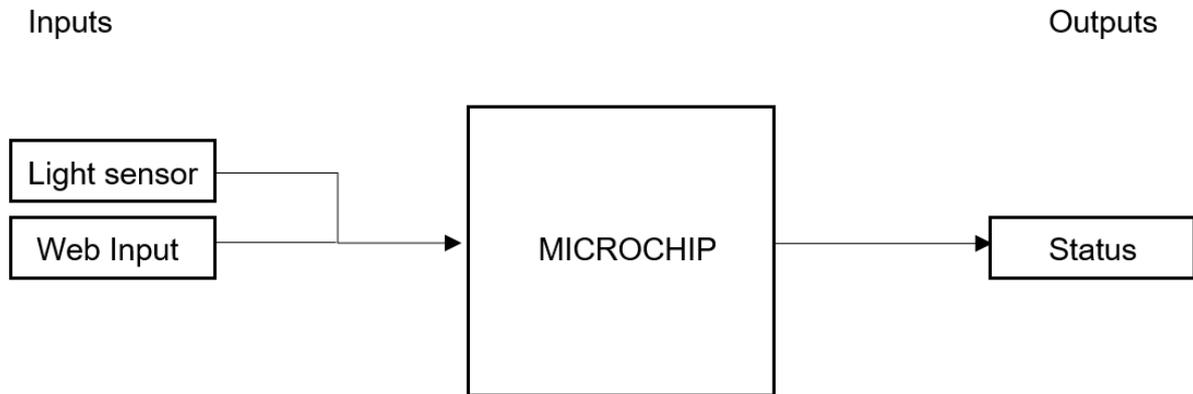
You may have heard or known that in Australia bushfires are common during the summer period. What causes such fire is the dry environment and the heat produced by the sun rays (and also other factors such as lightning and human activities). In the summer period, days are longer and the sun rays are shine with more intensity. The increased intensity warms up any surfaces that are in contact with the sun rays as the surfaces absorb the energy. In a non-dry environment, water will absorb the excess heat from the sunlight and evaporate, thereby preventing a build-up of energy that causes a surface to start burning. However, in Australia, there is not enough rain to keep up with evaporation during the summer, which leads to an increased fire risk. This leads up to a situation where bushfires easily occur without notice. Therefore, it can be said, where the light intensity of a region is low, it is less likely for a fire to erupt, whereas the opposite can be said for an area with high light intensity.

### Ideas

How can we measure the light intensity (level)? How can the microchip and runlinc tell the user if the level is high or low? Is a visual warning on a webpage enough, or would an additional method be advantageous? What kind of thresholds might be good? Is there any other information that might be good to display on our Webpage?

## Plan

We have Photoresistor Light Sensor (PLS) in our kits which we can use to check the light intensity of a possible fire, and a web input to change the light intensity risk.



**Figure 1:** Block diagram of Microchip outputs

## runlinc Background

runlinc is a web page inside a Wi-Fi chip. The programming is done inside the browsers compare to programming inside a chip. 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: Design the Circuit on runlinc

**Note:** Refer to runlinc Wi-Fi Setup Guide document to connect to runlinc

Use the left side of the runlinc web page to construct an input/output (I/O).

For port D33 name it Light\_Sensor and set it as ANALOG\_IN.

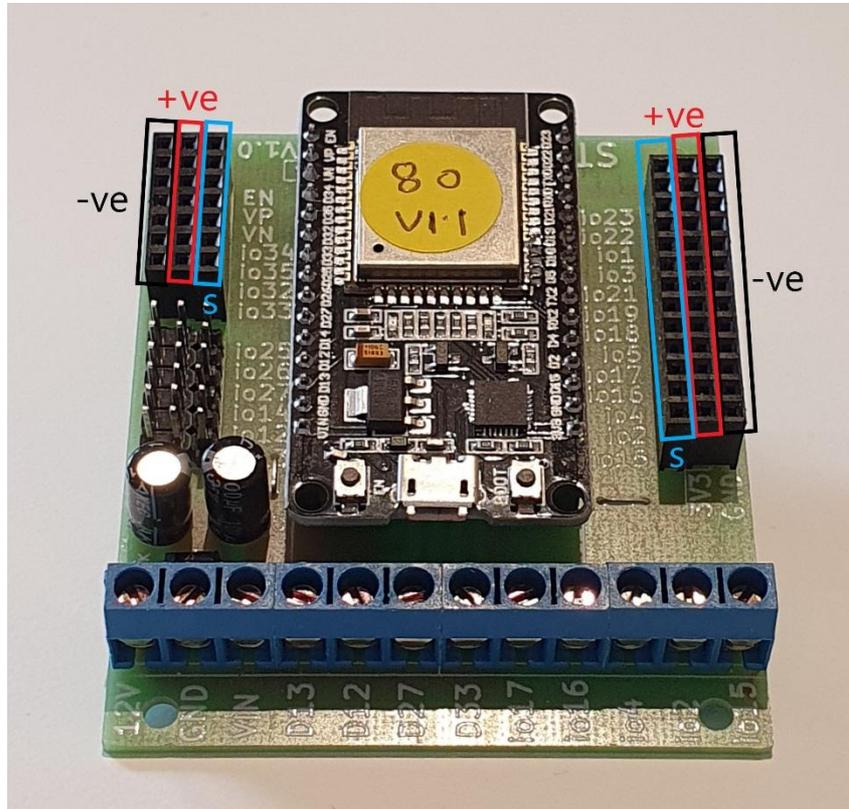
In our circuit design, we will be using the light sensor. We happen to have this in our kits, so these can be used on our circuit design, as per the plan.

D23	DISABLED		
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	DISABLED		
D33	ANALOG_IN	Light_Sensor	103
D34	DISABLED		

**Figure 2:** I/O configurations connections

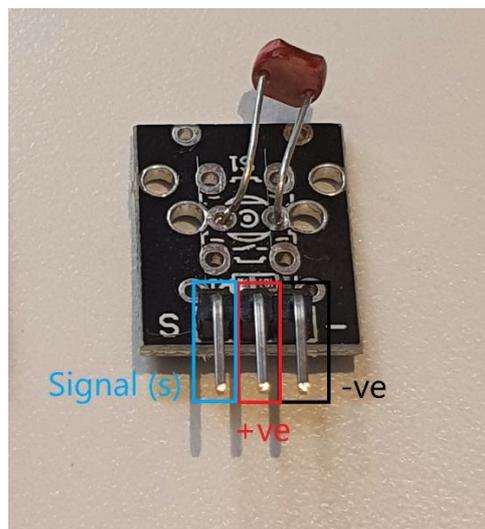
## Part B: Build the Circuit

Use the STEMSEL E32 board to connect the hardware. For this project we are using both the left and right I/O ports, with **negative port (-ve)** on the outer side, **positive port (+ve)** on the middle and **signal port (s)** on the inner side (as shown below).



**Figure 3:** Negative, Positive and Signal port on the E32 board

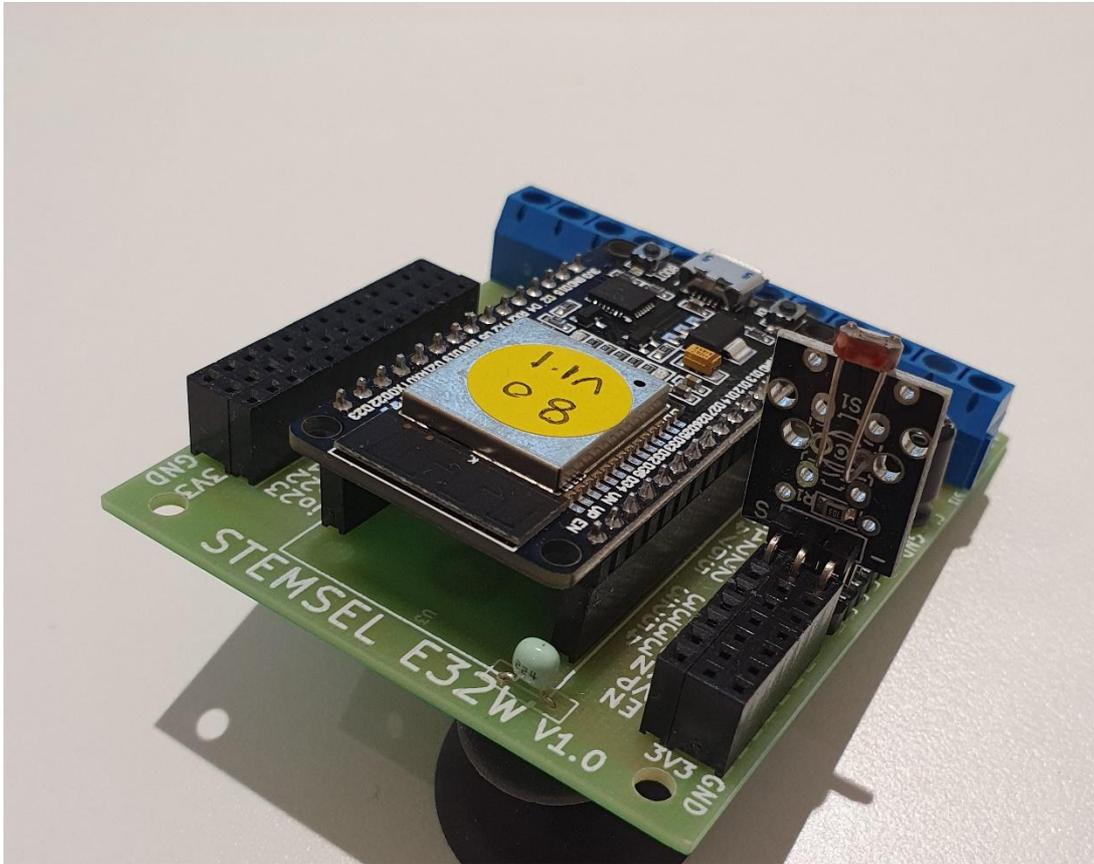
There is one I/O part we are using for this project, a Photoresistor Light Sensor (PLS) module (KY-018), their respective pins are shown in the figure below.



**Figure 4:** I/O part with negative, positive and signal pins indicated

**Wiring instructions**

- a.) Plug in the Photoresistor Light Sensor (PLS) to io33 on the E32 board.
- b.) Make sure the (-ve) pin are on the GND (outer) side of the I/O ports.



**Figure 5:** Circuit board connection with I/O part (side view)

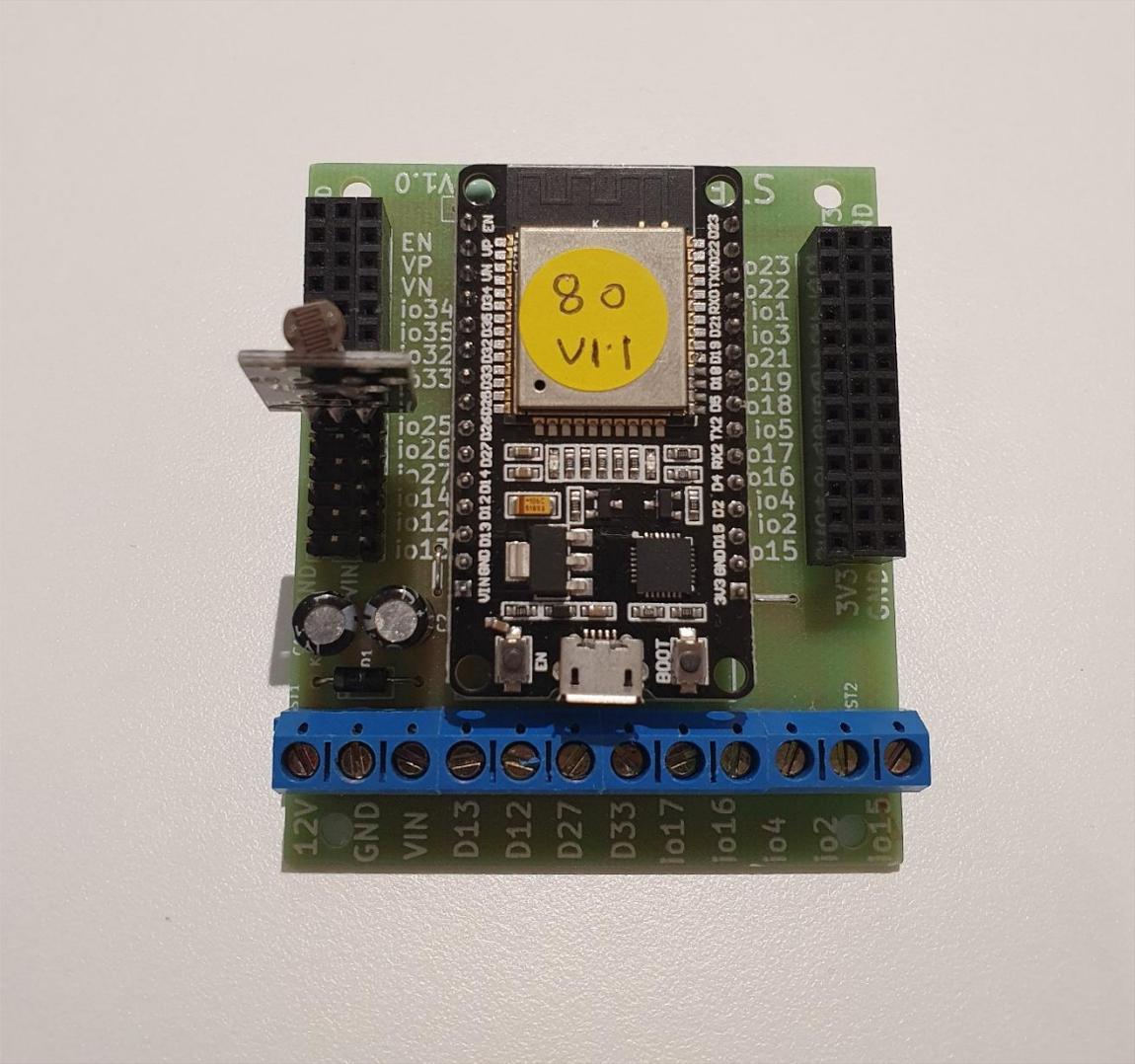


Figure 6: Circuit board connection with I/O part (top view)

## Part C: Program the Circuit

### HTML:

Before we program the logic into E32W board, we will first set up an HTML page.

1. Set up the text of the page and place it in the centre:

```
<div style="text-align:center">
</div>
```

2. Within the tags we will setup headings and body texts with respective unique IDs:

```
<h3>Fire Risk Dashboard powered by runlinc</h3>
<td>Current Light Intensity Now:</td>
<font id="Intensity">Loading...</font> Light Intensity out of 5 Light Intensity
<br>
Analysis: <font id="Message"></font>
<br>
<br>
Set Fire Risk Limit Setting:
<br>
<br>
Current Low Risk Limit is <font id="FRlimit">Loading...</font> Light Intensity.
```

3. Then between "Set Fire Risk Limit Setting:" and "Current Low Risk Limit..." we will add a slider that will help us to easily change the limit of the fire risk:

```
<input type="range" id="limit" min="0" max="5" step="0.1" value="2">
```

The final code should be typed into the HTML block:

```
<div style="text-align:center">
<h3>Fire Risk Dashboard powered by runlinc</h3>
<td>Current Light Intensity Now:</td>
<font id="Intensity">Loading...</font> Light Intensity out of 5 Light Intensity
<br>
Analysis: <font id="Message"></font>
<br>
<br>
Set Fire Risk Limit Setting:
<br>
<input type="range" id="limit" min="0" max="5" step="0.1" value="2">
<br>
Current Low Risk Limit is <font id="FRlimit">Loading...</font> Light Intensity.
</div>
```

This will set up our webpage to receive the information from the Photoresistor Light Sensor (PLS).

### **JavaScript Loop:**

Then we will program the code into E32W board using the Javascript Loop.

1. We will first need to have the board measure the light intensity from the light sensor, followed by converting the light intensity into a number within the range of 0 to 5.

```
sense_num = analogIn( Light_Sensor );  
sense_num = sense_num / 51;  
sense_num = sense_num.toFixed(1);
```

2. We will then need to retrieve the intensity that we have set on the HTML page using the HTML slider feature.

```
document.getElementById("Intensity").innerHTML = sense_num;  
Limit = document.getElementById("limit").value;  
document.getElementById("FRlimit").innerHTML = Limit;
```

3. Using the limit from step 3 we can then compare whether the risk is low or high.

```
if(sense_num > Limit)  
  document.getElementById("Message").innerHTML = "Fire Risk is LOW";  
else  
  document.getElementById("Message").innerHTML = "Fire Risk is HIGH";
```

The final code for JavaScript Loop block:

```
sense_num = analogIn( Light_Sensor );  
sense_num = sense_num / 51;  
sense_num = sense_num.toFixed(1);  
document.getElementById("Intensity").innerHTML = sense_num;  
Limit = document.getElementById("limit").value;  
document.getElementById("FRlimit").innerHTML = Limit;  
if(sense_num > Limit)  
  document.getElementById("Message").innerHTML = "Fire Risk is LOW";  
else  
  document.getElementById("Message").innerHTML = "Fire Risk is HIGH";
```

The IF ELSE loop determines where the fire risk is HIGH or LOW, **remember our Photoresistor Light Sensor have a higher reading when its surrounding is darker, and a lower reading when its surrounding is brighter.**

## Expected Result

### Fire Risk Dashboard powered by runlinc

Current Light Intensity Now: 2.2 Light Intensity out of 5 Light Intensity  
 Analysis: Fire Risk is LOW

Set Fire Risk Limit Setting:



Current Low Risk Limit is 2 Light Intensity.

Figure 7: Expect webpage screenshot

**runlinc V1.2** Copyright and International Patent Pending. All rights reserved.

File
Board

Run Code
Stop Code

Board IP:

ESP32

PORT	CONFIGURATION	NAME	STATUS
D2	DISABLED	<input type="text"/>	
D4	DISABLED	<input type="text"/>	
D5	DISABLED	<input type="text"/>	
D12	DISABLED	<input type="text"/>	
D13	DISABLED	<input type="text"/>	
D14	DISABLED	<input type="text"/>	
D15	DISABLED	<input type="text"/>	
RX2	DISABLED	<input type="text"/>	
TX2	DISABLED	<input type="text"/>	
D18	DISABLED	<input type="text"/>	
D19	DISABLED	<input type="text"/>	
D21	DISABLED	<input type="text"/>	
D22	DISABLED	<input type="text"/>	
D23	DISABLED	<input type="text"/>	
D25	DISABLED	<input type="text"/>	
D26	DISABLED	<input type="text"/>	
D27	DISABLED	<input type="text"/>	
D32	DISABLED	<input type="text"/>	
D33	ANALOG_IN	Light_Sensor	147
D34	DISABLED	<input type="text"/>	

**CSS**

**HTML**

```

<div style="text-align:center">
<h3>Fire Risk Dashboard powered by runlinc</h3>
<td>Current Light Intensity Now:</td>
<font id="Intensity">Loading...</font> Light Intensity out of 5 Light
Intensity
<br>
Analysis: <font id="Message"></font>
<br>
Set Fire Risk Limit Setting:
<br>
<input type="range" id="limit" min="0" max="5" step="0.1" value="2">
<br>
Current Low Risk Limit is <font id="FRlimit">Loading...</font> Light
Intensity.
</div>
                
```

**JavaScript** Select Macro select a device Add Macro

**JavaScript Loop** Select Macro select a device Add Macro

```

sense_num = analogIn( Light_Sensor );
sense_num = sense_num / 51;
sense_num = sense_num.toFixed(1);
document.getElementById("Intensity").innerHTML = sense_num;
Limit = document.getElementById("limit").value;
document.getElementById("FRlimit").innerHTML = Limit;
if(sense_num > Limit)
document.getElementById("Message").innerHTML = "Fire Risk is LOW";
else
document.getElementById("Message").innerHTML = "Fire Risk is HIGH";
                
```

Figure 8: Expect runlinc result screenshot

## Summary

What we did today is a form of application that can help us measure the fire risk around us and this case is a way to find out whether the light level is causing a fire risk. But this is not so simple as we just did. It should be noted that we are surrounded by many things that are flammable although they are in a damp environment. The world is a very unpredictable place. By learning how we can measure the level of risk, we can prioritise and control as many parameters as we can by learning what can be done to lower the risks that will be detrimental to our very being.