



runlinc Beginners Project 2: Smart Lights (E32W Version)

Contents

Introduction.....	1
Part A: Design the Circuit on runlinc	3
Part B: Build the Circuit.....	4
Part C: Program the Circuit	7
Summary.....	8

Introduction

Problem

We use lots of streetlights to light the roads at night. However, all of those lights use a lot of electricity which is entirely wasted if they turn on when it is not dark. We want to use our STEMSEL boards to make a smart streetlight that is only on when it is dark. We also want to program it using runlinc technology.

Background

Streetlights should only come on when it is dark, either at night or during very cloudy days. Having the streetlights only come on when they are needed saves electricity. This is clearly a good thing since fossil fuels like gas and coal that we burn generate electricity creates greenhouse gasses which contribute to global warming. This project allows you to create a smart streetlight.

Ideas

So how can we turn on the streetlight only when it is needed? We could use a timer to turn the light on at say 6:00 in the evening and off again at 6:00 in the morning, but the problem with this is that the days are not always the same length. Because of this, sometimes the streetlight might be on when it is still light, or still off when it is already dark. The best solution would be if the streetlight could see if it was dark or light and turn itself on or off automatically. Can we do this using our kits?

Plan

We have light-dependent resistors (LDR) in our kits which we can use to check if it is bright or dark. We can simulate the lightbulb in a streetlight using the smaller LED in our kits. First, we will use the LDR to check the light level, then the microchip can turn on the lightbulb when the sensor reading is HIGH and turn it off when the sensor reading is LOW.

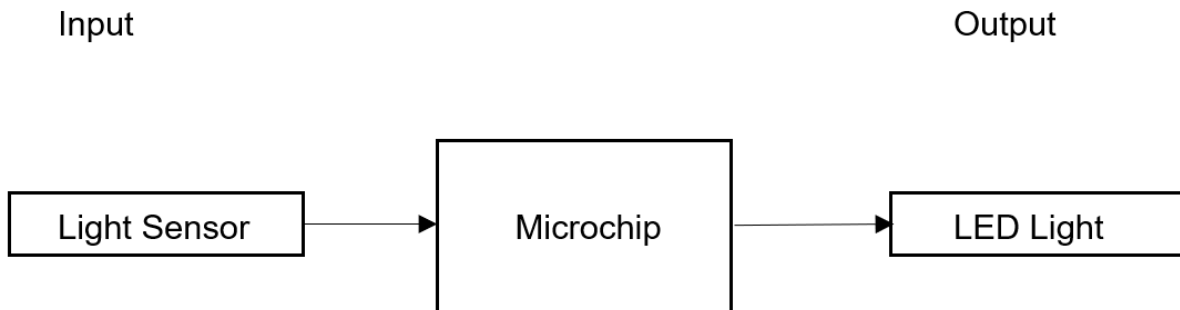


Figure 1: Block diagram of Microchip outputs

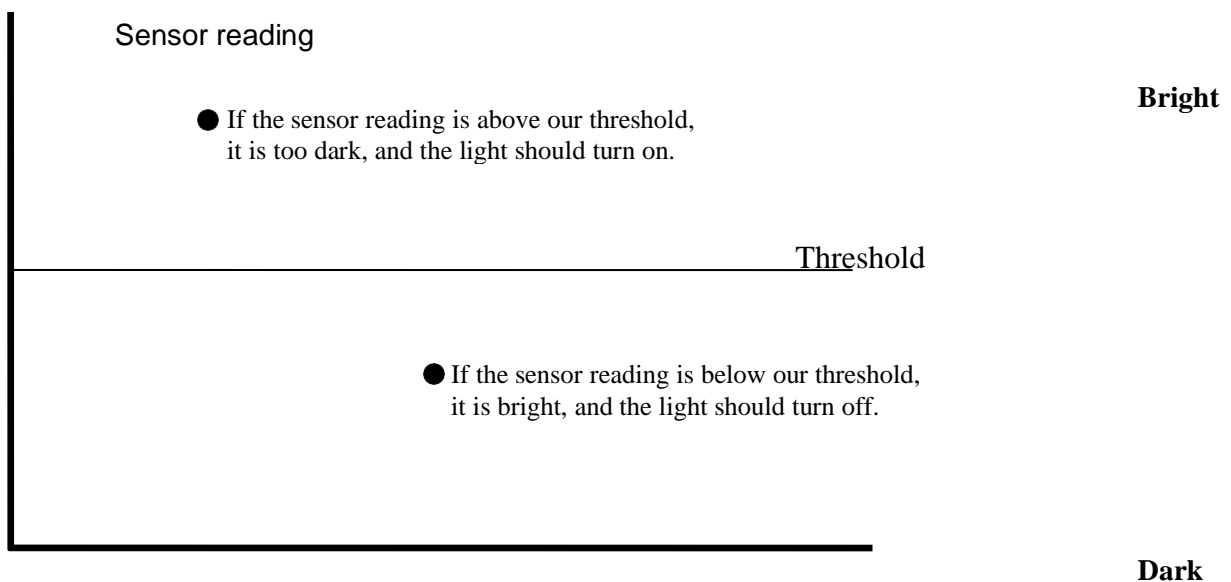


Figure 2: Light Level for a light sensor

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 D23 name it LED_Light and set it as DIGITAL_OUT.

For port D33 name it Lightsensor and set it as ANALOG_IN.

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

D19	DISABLED		
D21	DISABLED		
D22	DISABLED		
D23	DIGITAL_OUT	LED_Light	OFF
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	DISABLED		
D33	ANALOG_IN	Lightsensor	4
D34	DISABLED		
D35	DISABLED		

Figure 3: I/O configurations connections

Part B: Build the Circuit

Use the STEMSEL E32W 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).

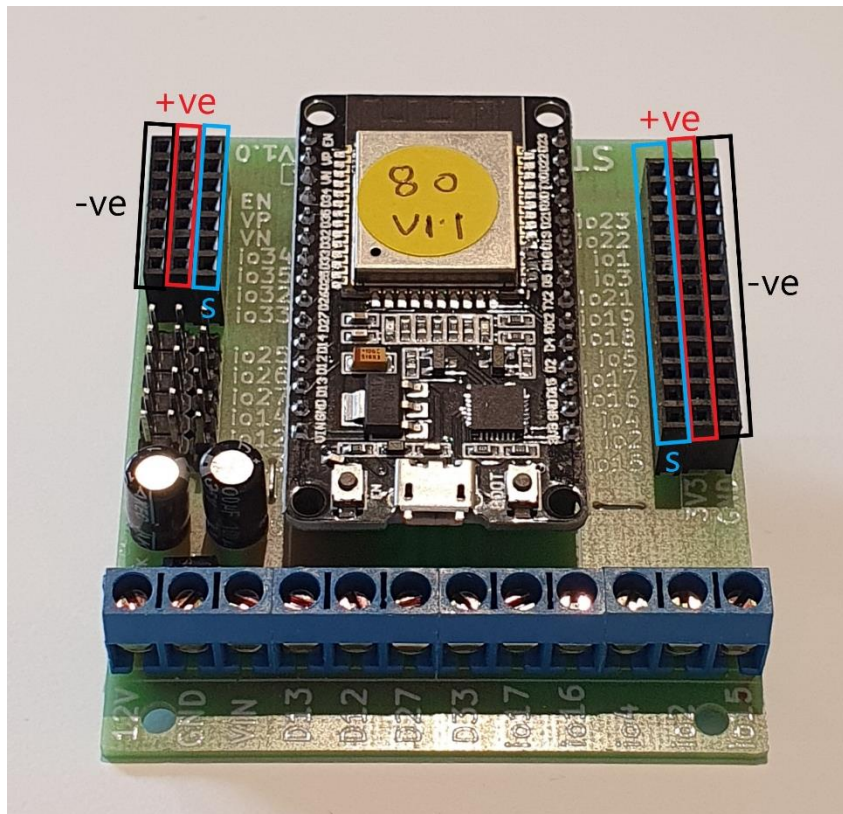


Figure 4: Negative, Positive and Signal port on the E32W board

There are two I/O parts we are using for this project, a 3-pin LED light and a Light Sensor, their respective pins are shown in the figure below.

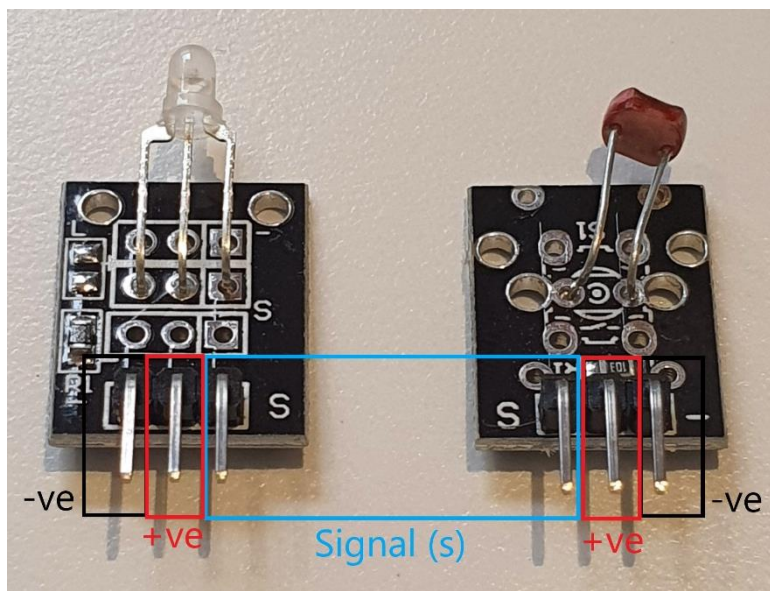


Figure 5: I/O parts with negative, positive and signal pins indicated

Wiring instructions

- a.) Plug in the LED to io23 on the E32W board.
- b.) Plug in the Light Sensor to io33 on the E32W board.
- c.) Make sure all (-ve) pins are on the GND (outer) side of the I/O ports.

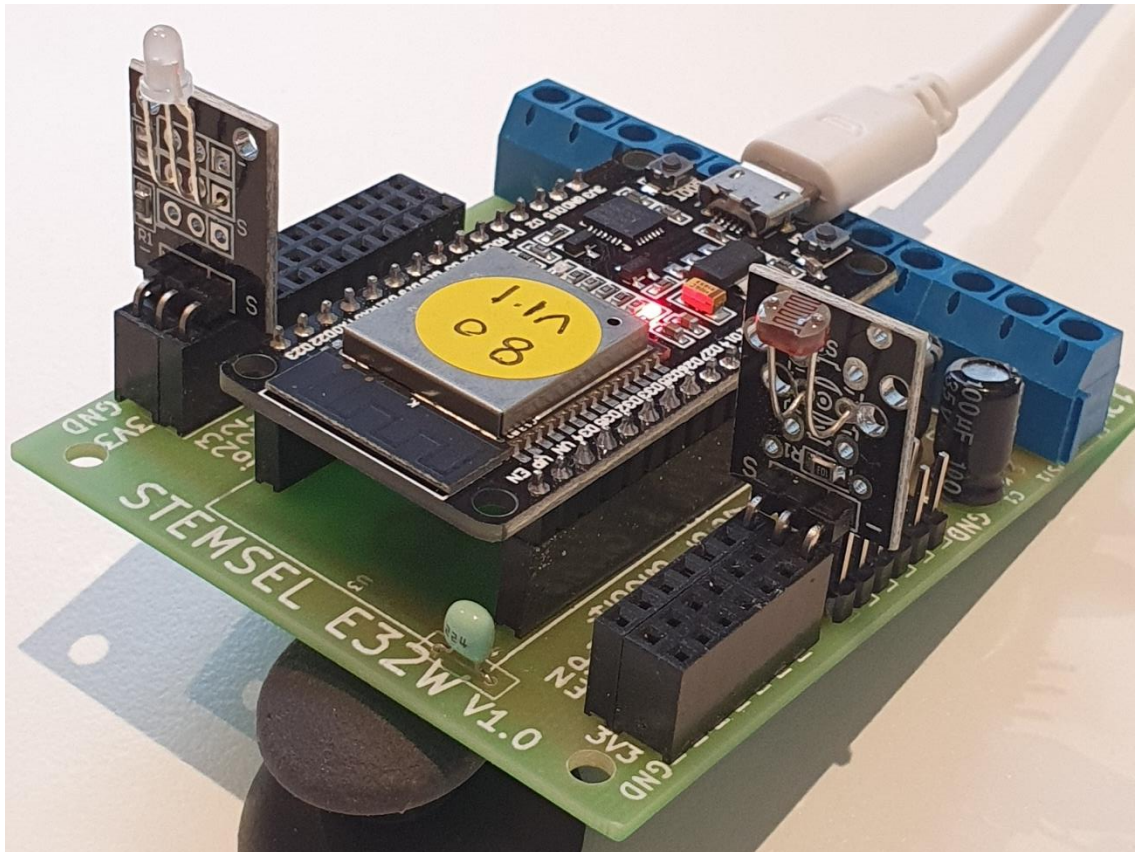


Figure 6: Circuit board connection with I/O parts (side view)

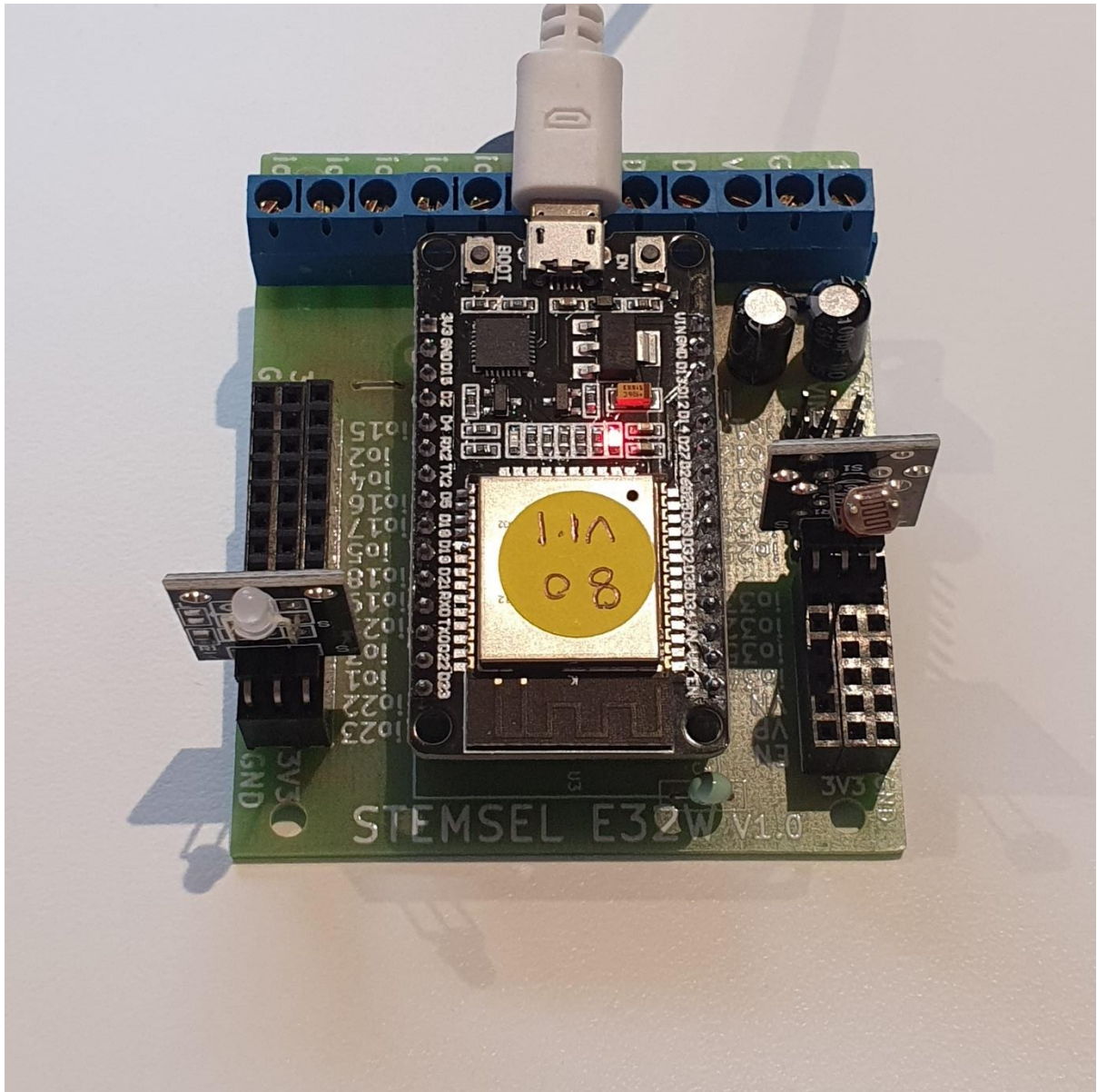


Figure 7: Circuit board connection with I/O parts (top view)

Part C: Program the Circuit

Now we can start to program the functions of the smart streetlight. To do this, JavaScript and JavaScript Loop is needed.

For **JavaScript** box type the following:

```
sensor = analogIn(Lightsensor);
```

A variable is initialised with it containing the input reading from the light sensor.

For **JavaScript Loop** type the following code:

```
sensor = analogIn(Lightsensor);
if( analogIn(Lightsensor) > 80) {
    turnOn(LED_Light);
}else{
    turnOff(LED_Light);
}
```

Here, a variable is initialised with it containing the threshold value for the light sensor.

Then an 'if statement loop' is added to the light sensor for it to decide when to turn on the LED Light. If the reading is above 80, the LED Light will turn on. However, if the reading is below 80, the LED Light will turn off. **Remember our Light Sensor have a higher reading when its surrounding is darker, and a lower reading when its surrounding is brighter.**

The screenshot shows the runlinc webpage interface. On the left is a pin configuration table with columns for pin number, mode, device, and status. On the right are two JavaScript code blocks for programming the circuit.

D12	DISABLED		
D13	DISABLED		
D14	DISABLED		
D15	DISABLED		
RX2	DISABLED		
TX2	DISABLED		
D18	DISABLED		
D19	DISABLED		
D21	DISABLED		
D22	DISABLED		
D23	DIGITAL_OUT	LED_Light	OFF
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	DISABLED		
D33	ANALOG_IN	Lightsensor	2

JavaScript Select Macro select a device Add Macro

```
sensor = analogIn(Lightsensor);
```

JavaScript Loop Select Macro select a device Add Macro

```
sensor = analogIn(Lightsensor);
if( analogIn(Lightsensor) > 80) {
    turnOn(LED_Light);
}else{
    turnOff(LED_Light);
}
```

Figure 8: runlinc webpage screenshot

Summary

It is important to light the streets at night so that people can see where they are going, but it is also important to save electricity. By ensuring our streetlight only comes on when it is dark, both considerations are addressed. During this project, you should have learned how to use analogue inputs to use a light sensor, and a compare icon to do different things depending on that input.