runlinc Project 2 B2: Smart Lights (E32W Version)

Contents

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 and climate change. 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.

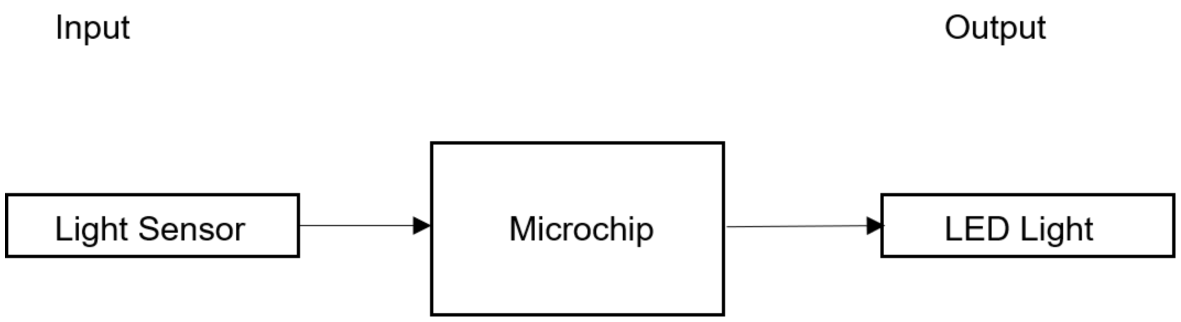

Figure 1: Block diagram of Microchip outputs

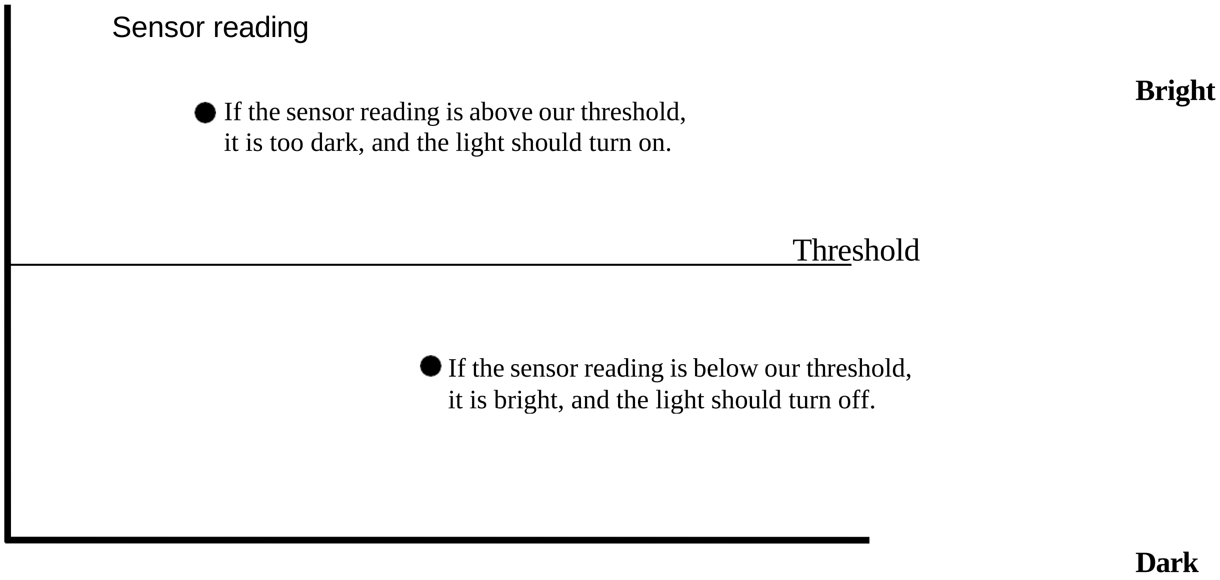

Figure 2: Light Level for a light sensor

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 D5 name it LED_Light and set it as DIGITAL_OUT.
For port D19 set it as DIGITAL_OUT (used as negative pin of LED – no name needed).
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.

D5	DIGITAL_OUT	LED_Light	OFF
D12	DISABLED		
D13	DISABLED		
D14	DISABLED		
D15	DISABLED		
RX2	DISABLED		
TX2	DISABLED		
D18	DISABLED		
D19	DIGITAL_OUT		OFF
D21	DISABLED		
D22	DISABLED		
D23	DISABLED		
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	DISABLED		
D33	ANALOG_IN	Lightsensor	234

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).

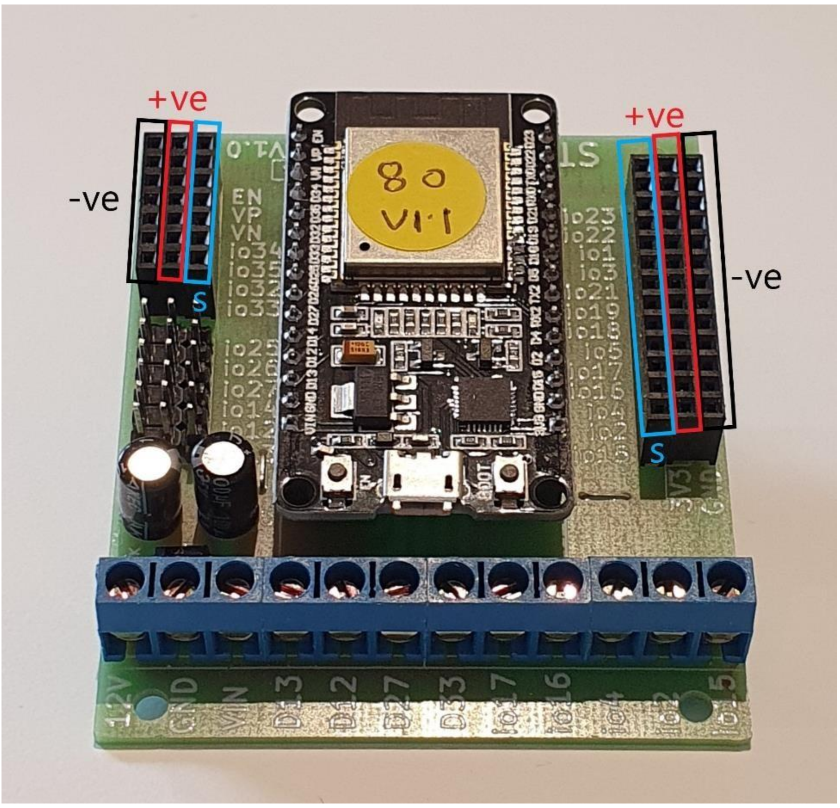

Figure 4: Negative, Positive and Signal port on the E32W board

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. In some kits, due to different manufacturers, the green pin and the red pin for the LED can be inverted. But no worries, we will test it after we build the circuit.

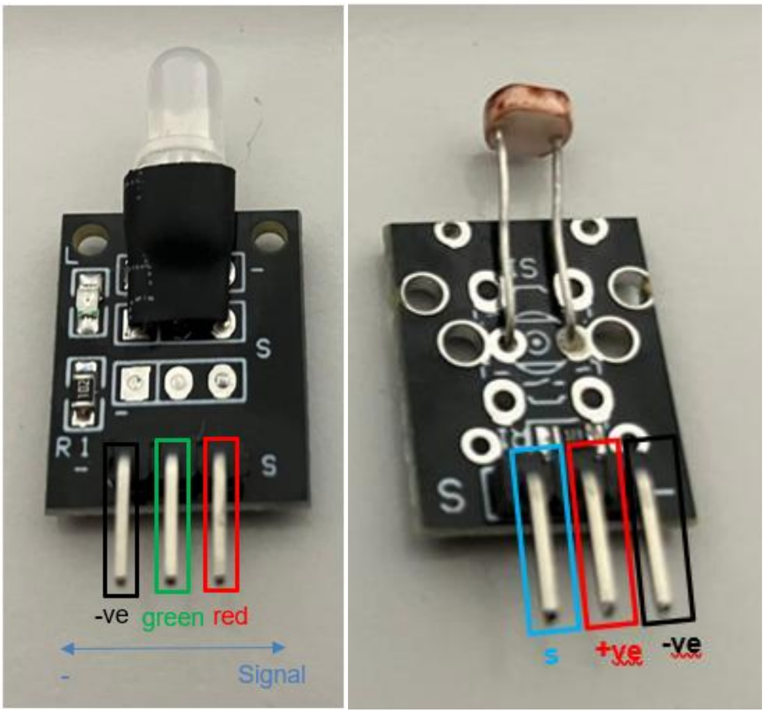

Figure 5: I/O parts with negative, positive and signal pins indicated

Wiring instructions

a.) Plug in the LED to signal ports of io5, io18 and io19 on the E32W board with the “S” pin on the light goes into io5 (we did not configure port D18 on the webpage for this connection because for this 2 color LED light, the middle pin which is connected to io18 is for lighting up the green color, and we don’t need it for this project).

b.) Plug in the Light Sensor to io33 on the E32W board.

c.) Make sure light sensor’s (-ve) pins are on the GND (outer) side of the I/O ports.

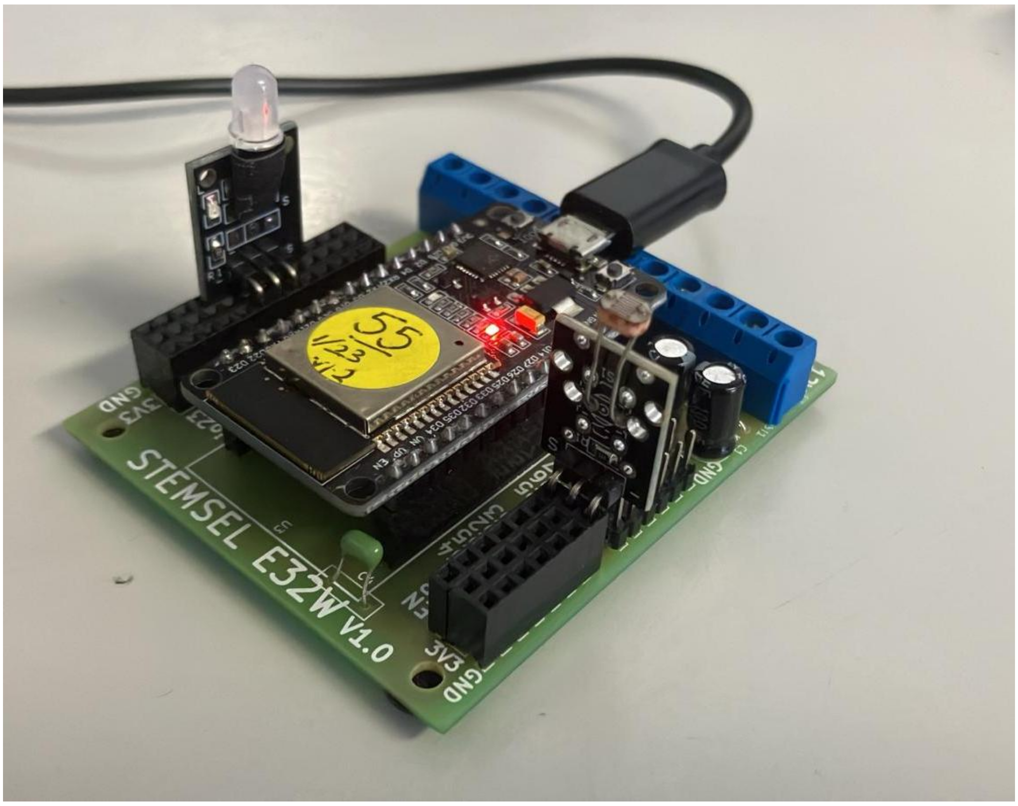

Figure 6: Circuit board connection with I/O parts (side view)

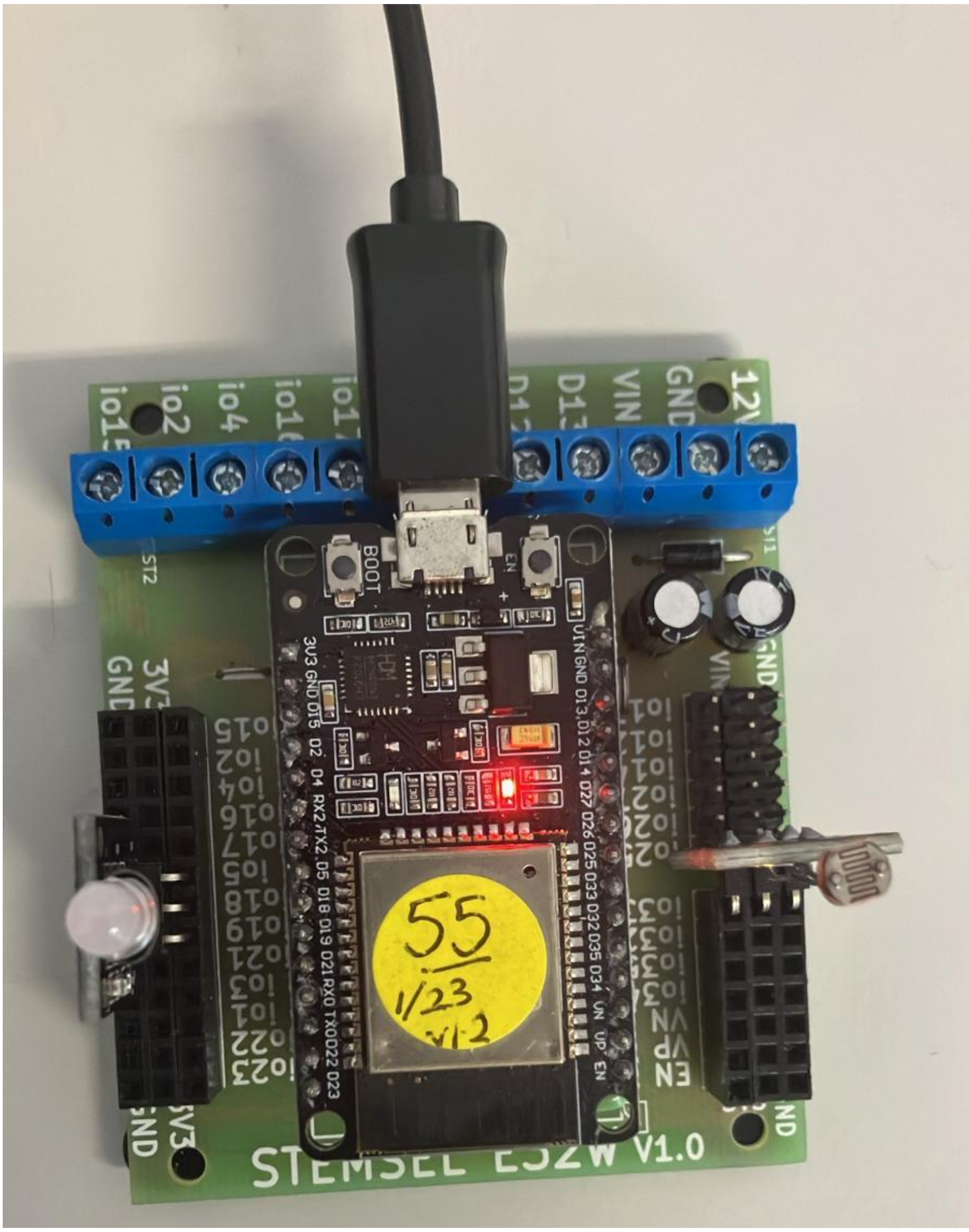

Figure 7: Circuit board connection with I/O parts (top view)

Part C: Program the Circuit

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

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.**

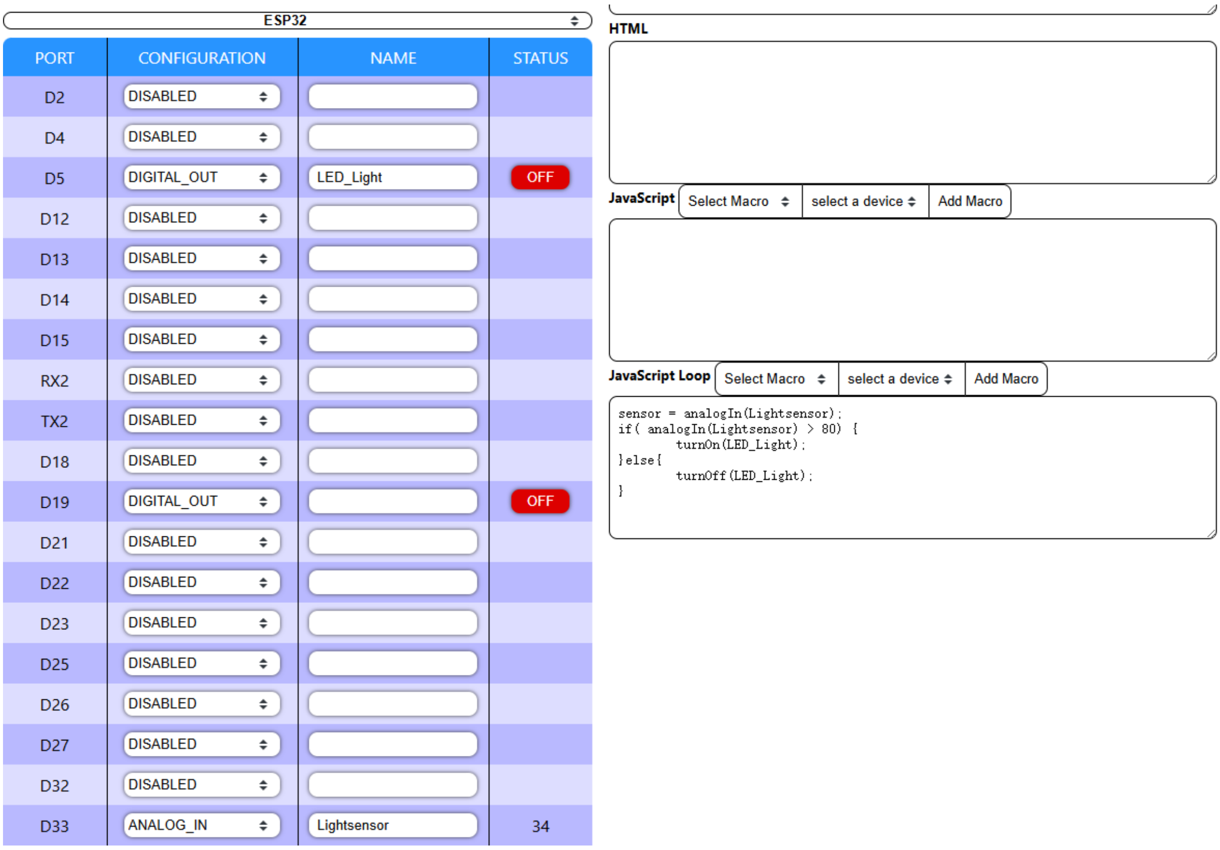

Figure 9: runlinc webpage screenshot

If you cover the surface of the light sensor’s probe part, the “Lightsensor” port should get a higher reading, and the darker the probe “feels”, the higher the reading will be.

If the reading is over 80, the statement(command) in the braces after the “if” statement in the program will be executed, because the execution (trigger) of the code in the braces after the “if” statement is set to “a light sensor reading over 80” in the parentheses after “if”.

The command is to give the port named as “LED_Light” voltage. And that will turn the LED on.

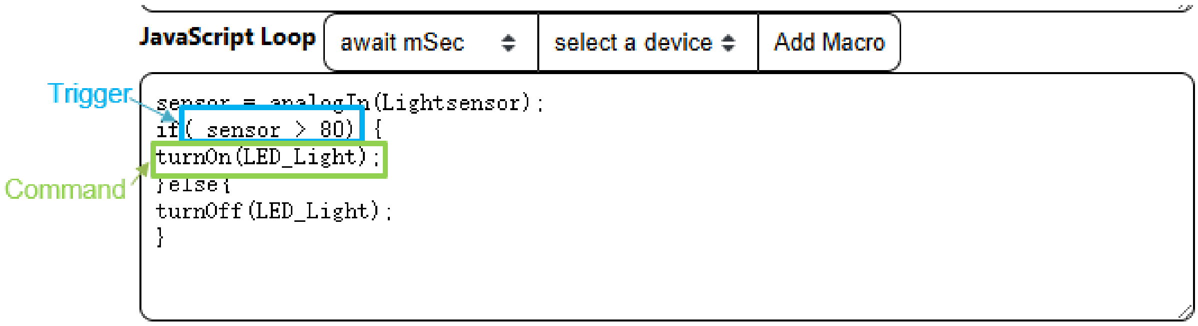

Figure 8: runlinc webpage JavaScript Loop

If you make it “feels” brighter by exposing the surface to light, and make the reading not higher than 80, the statement(command) in the braces after “else” will be executed, because the execution of the code in the braces after this “else” is set to “anything but not an over 80 reading from the light sensor”. The statement(command) will turn the LED off.

Part D: The Working Project

If you successfully built the circuit and the program correctly, it should act as what this part describe:

Run the program by pressing "Run Code" button on the webpage;

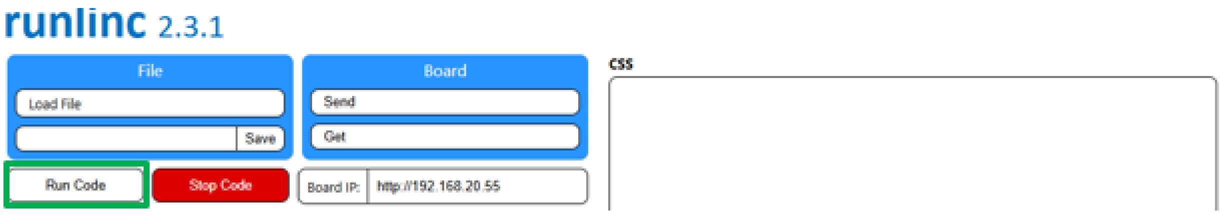

Figure 9: runlinc webpage "Run Code"

When the project is running, if you cover the surface of the light sensor's probe part, the LED will be turned on.

Note: If you see a light blinking green, change the name and setting from port D5 to port D18:

Set the port D18 on the control web page as "DIGITAL_OUT"
Copy the name "LED_Light" on port D5
Paste it to port D18
Delete the port name on D5
Set D5 as "DISABLED"

and you will correct it.

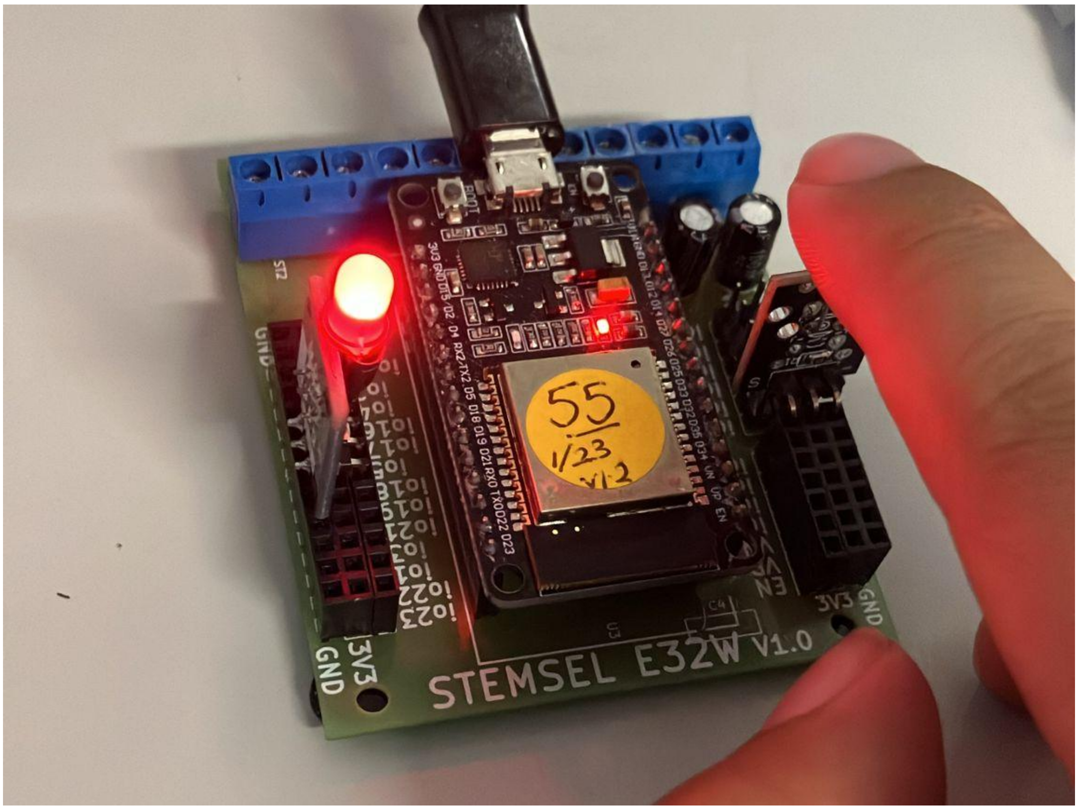

Figure 10: Turning on the LED by covering the sensor surface

If you make it "feels" brighter by exposing the surface to light, the LED will be turned off.

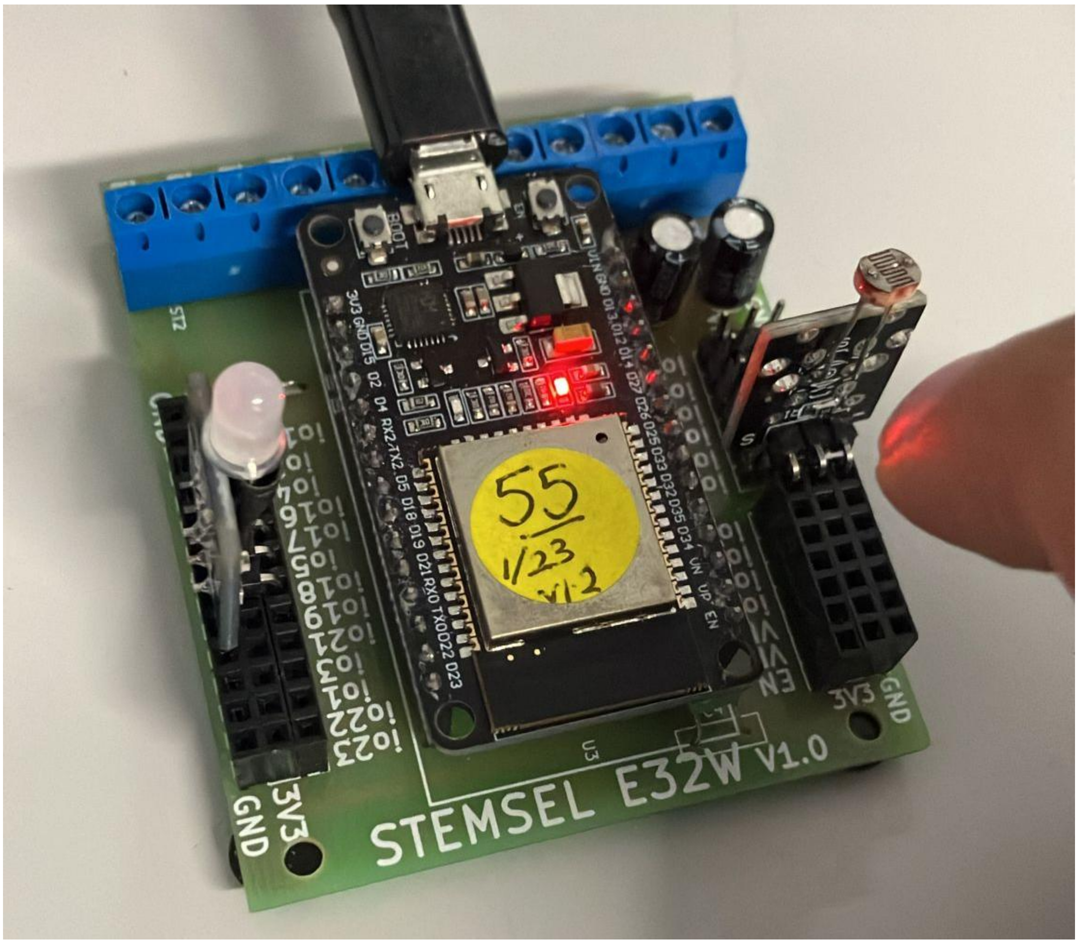

Figure 11: Turning off the LED by exposing the sensor surface to light

Then, we've made a smart light which will turn on when the environment is dark (when the sensor "feels" dark).

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.