



runlinc Project 6 B3: Fridge Alarm (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

Fridges only work when the door is closed. We need to create an alarm that will tell us if the fridge door has been left open. This can be done by using runlinc.

Background

Fridges have become essential in our daily lives. By keeping our food cold, we can keep our food fresh for a longer period. Previously it was necessary either buy perishable groceries every day or to preserve them in a time-consuming way. The invention of the cooling box was useful, but it still required large blocks of ice to be bought periodically to keep the food cold as there was no electricity. Now we can buy food and keep it fresh in the fridge for days. Refrigerators also keep pests like mice and cockroaches away from our food which prevents disease. However, if the fridge door is left open, the fridge can't keep working, and the pests could contaminate our food! This project constructs an alarm that sounds when the fridge door has been left open. It uses the STEMSEL controller board, Wi-Fi runlinc chip, a Light Dependent Resistor (LDR), a yellow LED and web buzzer tone generator. When the fridge door is opened, the LDR senses the light in the fridge and activates a web buzzer tone generator, and a LED as a "door open" alarm.

Ideas

How can the microchip tell if the fridge door is open? What are the differences between when the door is open and when it is closed? Do we want the alarm to activate as soon as the door is opened, or should we have a delay? How can the alarm let us know that the door is open?

Plan

Two things might change when the door is open: the light level and the temperature. We have light-dependent resistors (LDR) in our kits which we can use to check if the door is open, the microchip can turn on the lightbulb when the sensor reading is LOW and turn it off when the sensor reading is HIGH. When the alarm turns on, we will activate a buzzer sound to let the user know the fridge door is still open.

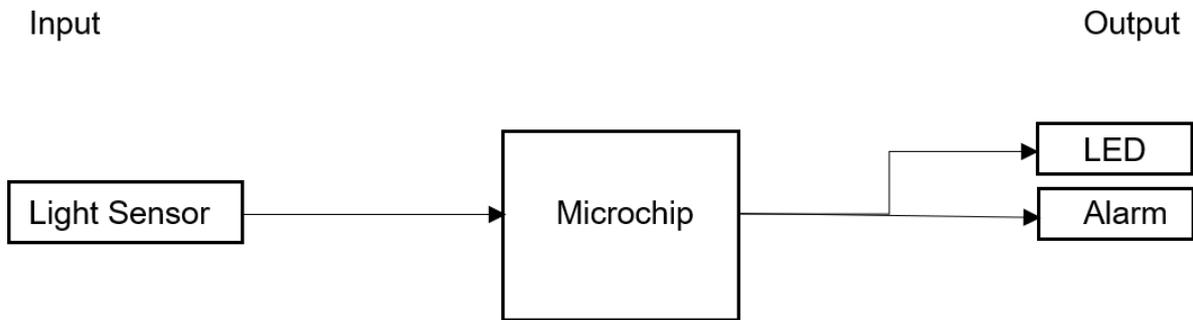


Figure 1: Block diagram of Microchip outputs

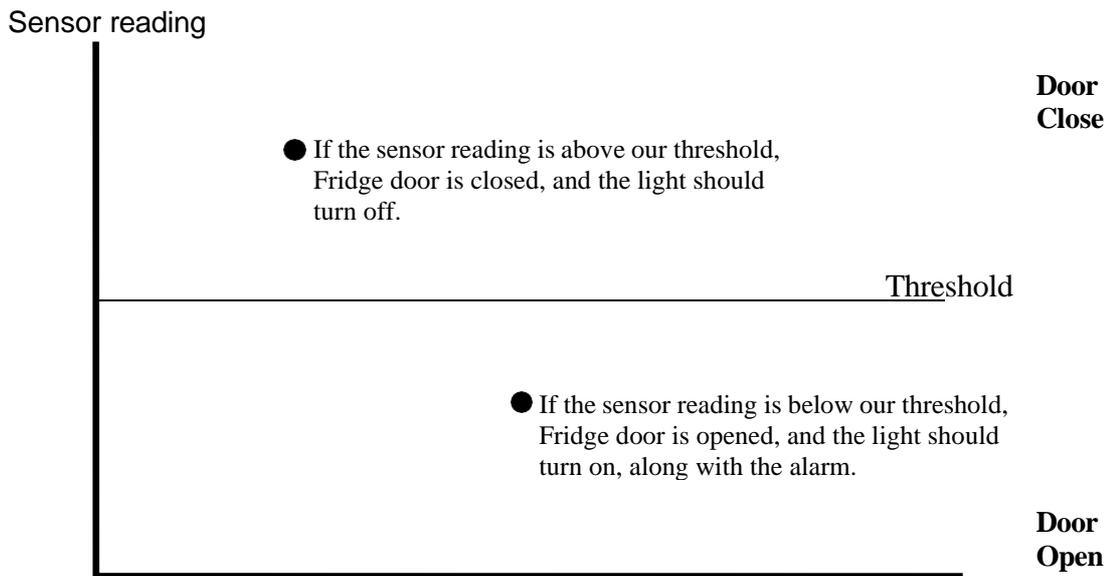


Figure 2: Light Level for a light sensor

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. Buzzer won't be used because it can't communicate. Instead, a tone generator from the web browser is used.

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 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).

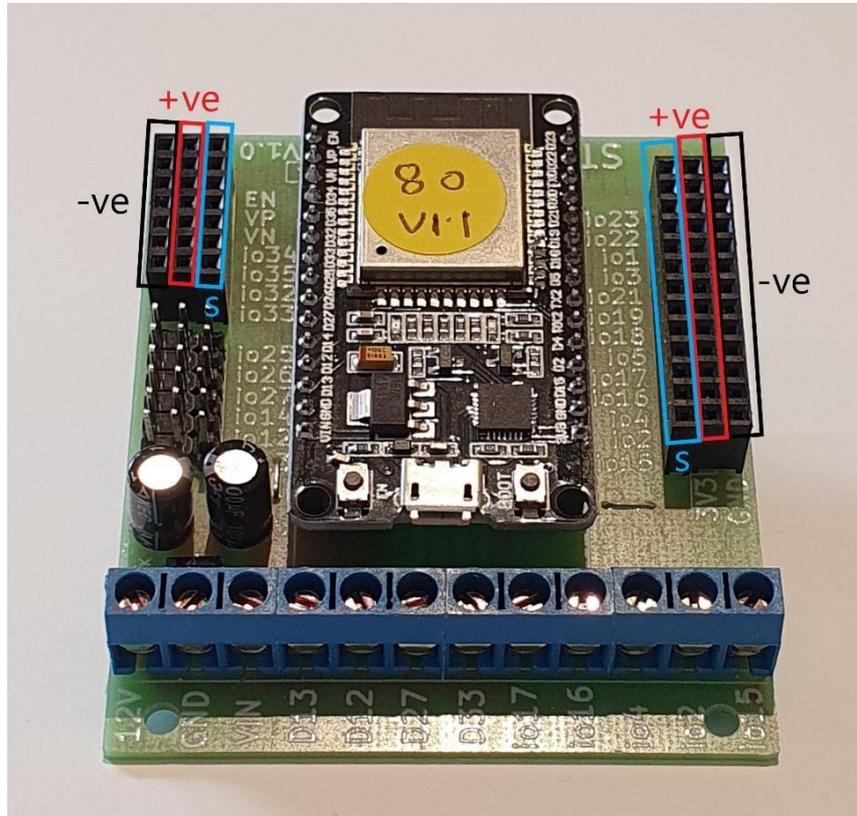


Figure 4: Negative, Positive and Signal port on the E32 board

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.

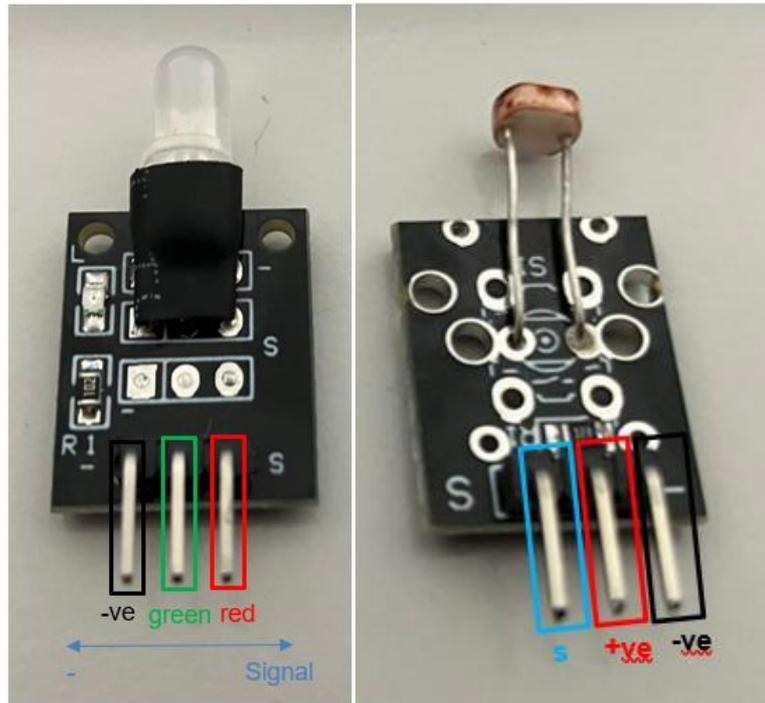


Figure 5: I/O parts with negative, positive and signal pins indicated

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

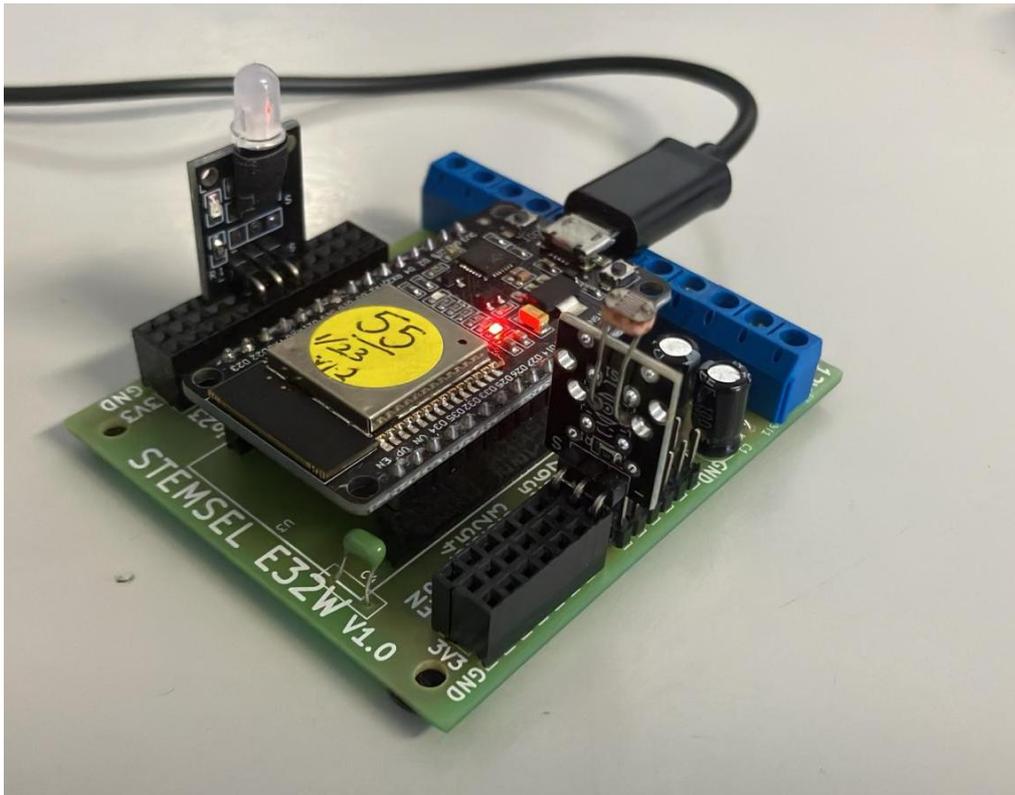


Figure 6: Circuit board connection with I/O parts (side view)

Part C: Program the Circuit

Use the blocks on the right side of the runlinc webpage to program the functions of the fridge alarm. Use the HTML to add contents, CSS to add style in your favour and JavaScript to program the microchip. For this case, JavaScript Loop is needed to program it to act as an alarm. Type the following code.

For **JavaScript Loop** box:

```
Result = analogIn(Lightsensor);
if(Result > 80 ){
  turnOff (LED_Light);
}else{
  let audioCtx = new AudioContext();
  let oscillator = audioCtx.createOscillator();
  oscillator.frequency.value = 4440;
  oscillator.connect(audioCtx.destination);
  oscillator.start();
  oscillator.stop(audioCtx.currentTime + 3);
  turnOn (LED_Light);
}
```

If the light sensor input is below 80, the fridge alarm will turn on.

If light sensor input is above 80, the fridge alarm will turn off.

We can then modify the fridge alarm program so that the alarm is activated after the fridge door has been opened for more than five seconds. The time delay of five seconds should only occur when the door is opened and not any other time in the program. We will edit the **}else{** section to:

```
Result = analogIn(Lightsensor);

if(Result > 80 ){
  turnOff ( LED_Light );
}else{
  await mSec( 5000 );
  Result = analogIn(Lightsensor);
  if(Result <= 80){
    let audioCtx = new AudioContext();
    let oscillator = audioCtx.createOscillator();
    oscillator.frequency.value = 4440;
```

runlinc Project 6 B3: Fridge Alarm (E32W Version)

```
oscillator.connect(audioCtx.destination);
oscillator.start();
oscillator.stop(audioCtx.currentTime + 3);
turnOn ( LED_Light );
}
}
```

The new else loop includes a time delay of 5000 milliseconds, then check the light sensor's reading is above or below the threshold. If the reading is below 80, the LED Light will turn on. However, if the reading is above 80, the LED Light will turn off.

After click the **Run Code** button, the light sensor reading is already below 80, therefore after 5 seconds the LED light should turn on, and a buzzer sound should be heard over the speaker. If the light sensor is covered, then reading for the light sensor should go up, and the LED will be turned off.

Note: If you see a light blinking with the colour you don't like, 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.

Remember our Light Sensor have a higher reading when its surrounding is darker, and a lower reading when its surrounding is brighter.

runlinc Project 6 B3: Fridge Alarm (E32W Version)

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	51

JavaScript Select Macro select a device Add Macro

JavaScript Loop turnOff select a device Add Macro

```

Result = analogIn(Lightsensor);
if(Result > 80 ){ turnOff ( LED_Light );
}else{
await mSec( 5000 );
Result = analogIn(Lightsensor); if(Result <= 80){
let audioCtx = new AudioContext();
let oscillator = audioCtx.createOscillator(); oscillator.frequency.value = 4440;
oscillator.connect(audioCtx.destination); oscillator.start();
oscillator.stop(audioCtx.currentTime + 3); turnOn ( LED_Light );
}
}

```

Figure 8: runlinc webpage screenshot (final)

Summary

The electric fridge is a highly useful device, but it must be used properly, otherwise we waste electricity and run the risk of illness from spoiled food. Our fridge alarm will warn us if the fridge is left open too long and also if the temperature gets too high in the fridge.