



runlinc Intermediate Project 3: Wind Power (E32W Version)

Contents

Introduction.....	1
Part A: Design the Circuit on runlinc.....	5
Part B: Build the Circuit.....	6
Part C: Program the Circuit.....	11
Bonus	13
Summary	14
Appendix – Expected Results	15

Introduction

Problem

We want to create an electricity generator that does not produce any greenhouse gasses.

Background

We all use electricity every day for all sorts of different purposes, but did you ever stop to think about what electricity is made of, and where it comes from? You can find out in the following video: <https://youtu.be/ru032Mfsfig> .

The word electricity derives from the word *electron*. If you have already learned about atoms, you may have heard of electrons before. Atoms are the tiny particles that make up the world around us, and they are made up of three even smaller particles. The positive *protons* and the neutral *neutrons* are clumped in the middle of the atom and make up the nucleus. The negative electrons orbit around the nucleus. Figure 1¹ shows a well-known simplified model of an atom; however, advances in physics, in the quantum field, reveals a more detailed model that helps us understand more about an atom's behaviour.

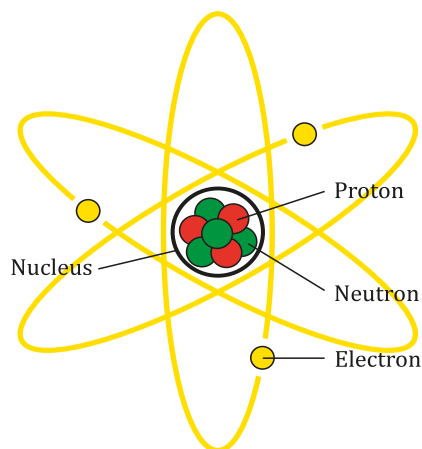


Figure 1 A simplified model of an atom.

¹ https://commons.wikimedia.org/wiki/File:Atom_Diagram.svg

In certain types of materials, the electrons are only loosely attached to their atoms, and if given a little nudge can move from one atom to another. These materials are called conductors, and it is those loosely bound electrons that allow them to conduct electric current. The moving electrons are what we call electricity, and voltage is how we measure the strength of the push. You can see how moving electrons work in figure 2².

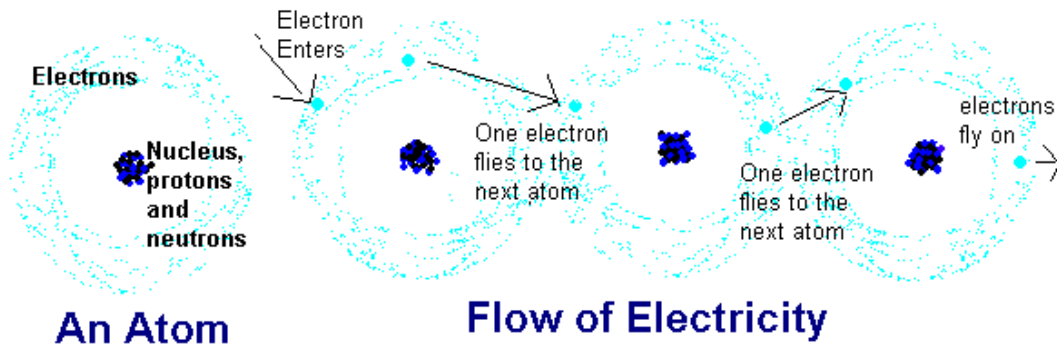


Figure 2 A simplified view on flow of electricity (moving electrons).

So, how can we give these electrons the push they need to become electricity? While there are several ways, one of the most used methods is to pass a magnet close to a metal wire.

The magnetic field generated by the magnet can give the electrons the energy they need to flow from one atom to the next, but only while the magnet is passing by. In this way, the kinetic (movement) energy of the magnet is turned into electrical energy. To continually pass the magnet past the wire, it can be attached to a shaft which is then rotated at high speed. Can you think of something in your kit that rotates at high speed?



Figure 3 DC Motor with Fan blades.

The DC motor in your kit, check figure 3, uses this same principle typically in reverse to turn electrical energy into movement energy and spin around. For this project, we want to use the fan on the motor to make electricity. Large fan-like objects called wind turbines are used in some areas as an environmentally friendly source of electricity. Since wind turbines only use the wind, they do not burn fossil fuels and therefore do not produce any greenhouse gases, making them better for the environment than other types of electricity generators. You can find out about wind turbines here: https://youtu.be/qSWm_nprfqE .

² <http://www.qrg.northwestern.edu/projects/vss/docs/power/2-whats-electron-flow.html>

Ideas

How should we connect the fan to the E32W controller board? Will the electricity be an input or an output? How can we display how much electricity is being produced?

Plan

We will use the fan to generate the voltage (by blowing air with your mouth into the fan) and hence electricity. However, we will need to use this as an input to the microchip. We will display the exact voltage the motor can produce on the browser. But when we aren't looking at the browser, we will not be able to view the precise voltage, but we can use several LEDs to display several voltage levels.

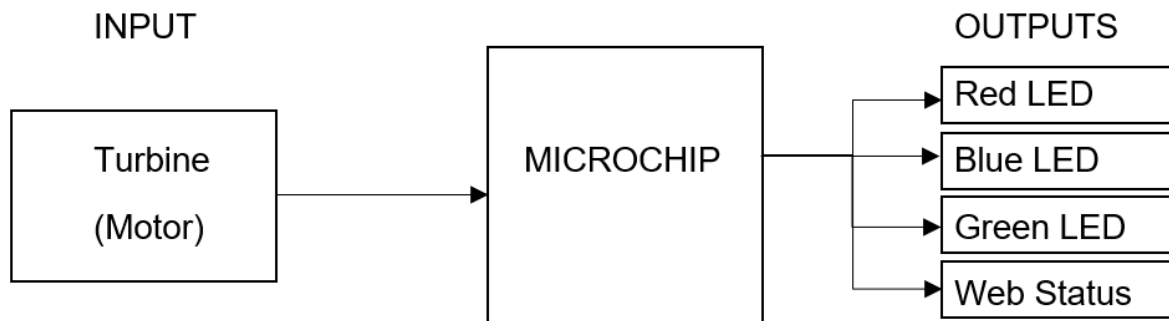


Figure 4 I/O Plan for this project.

So, if the voltage is above the first threshold one LED will be on, two LEDs will be on if it's above the second threshold, and all three LEDs will be on if the voltage rises above the third threshold.

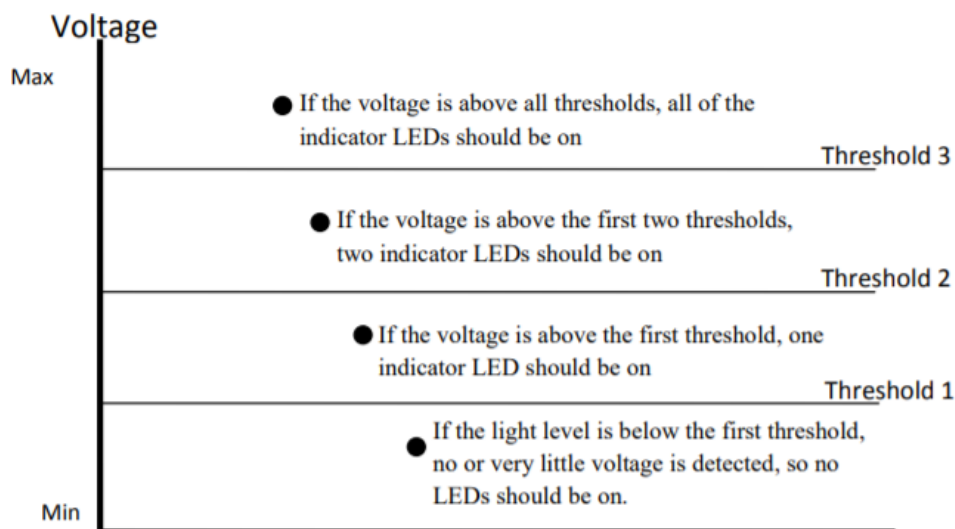


Figure 5 Threshold implementation graph.

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 D18 name it Green and set it as DIGITAL_OUT.

For port D19 name it Red and set it as DIGITAL_OUT.

For port D21 name it Blue and set it as DIGITAL_OUT.

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

TX2	DISABLED		
D18	DIGITAL_OUT	Green	OFF
D19	DIGITAL_OUT	Red	OFF
D21	DIGITAL_OUT	Blue	OFF
D22	DISABLED		
D23	DISABLED		
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	DISABLED		
D33	ANALOG_IN	Turbine	0
D34	DISABLED		

Figure 6: I/O configuration connections.

Part B: Build the Circuit

Use the 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).

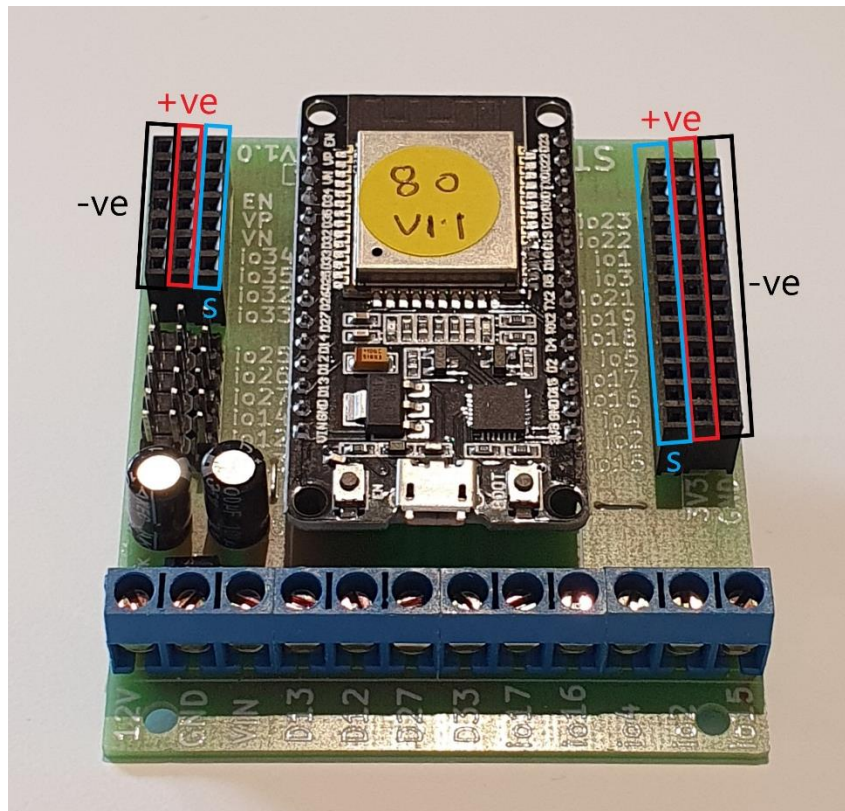


Figure 7: Negative, Positive and Signal port on the E32 board.

There are two I/O part we are using for this project, a DC motor with fan blade, and a 4-pin RGB LED module (KY-016), their respective pins are shown in the figure below.

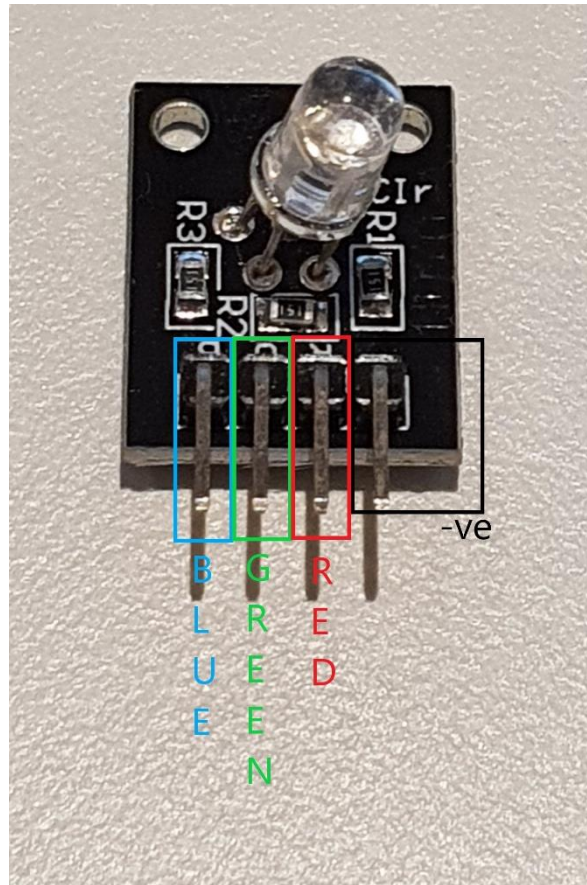


Figure 8: 4-Pin RGB LED with respective pins indicated.

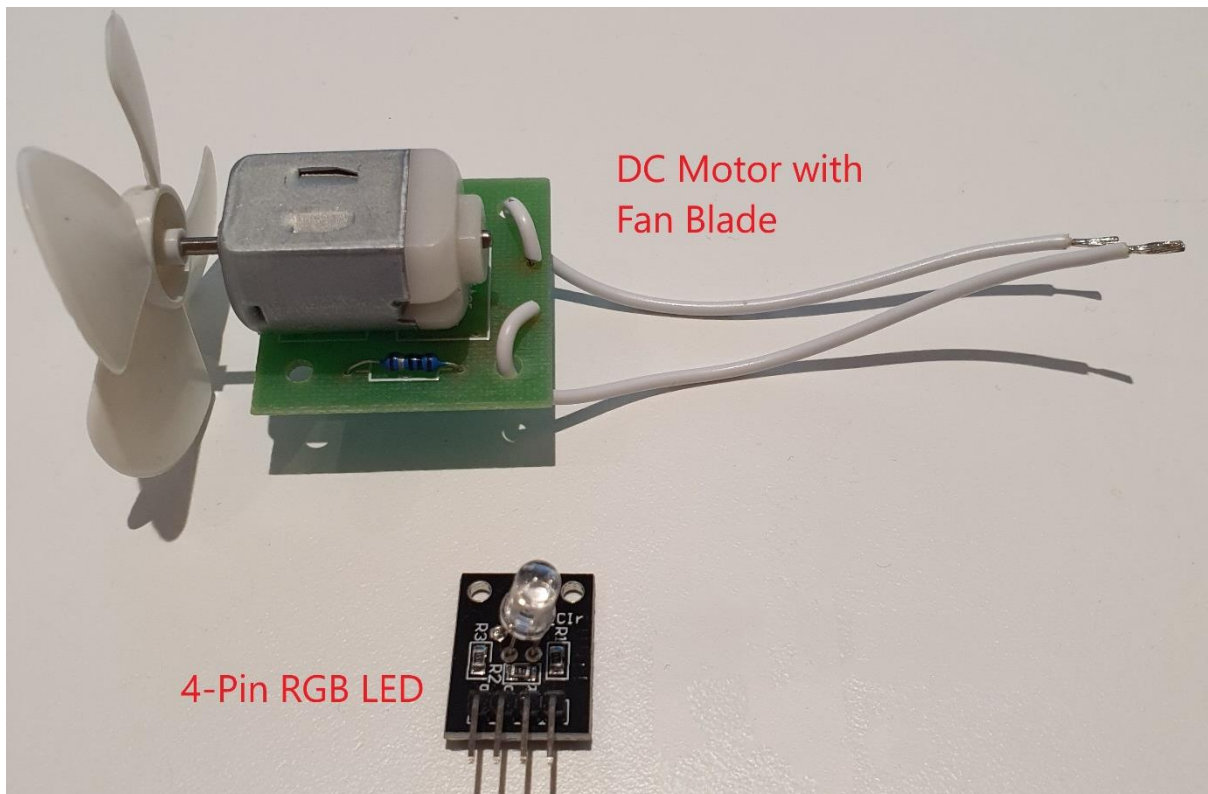


Figure 9: Both I/O parts labelled.

Wiring Instructions

Since we only have 3-pins port slots, we need to plug in all 4 pins on the RGB LED into the signal ports.

- a.) Plug in the negative pin on the 4-pins RGB LED into port io3, the rest should be in horizontal order and plug into ports io21, io19, io18. Please refer to **Figure 11** for more detail. (Note that there should be 3 rows of ports left starting from the GND pin.)
- b.) Connect both wires of the DC motor to the signal(s) port and GND port of io33. If the wires do not fit then you may need to modify them with jumper plug pin ends. (See figure below)

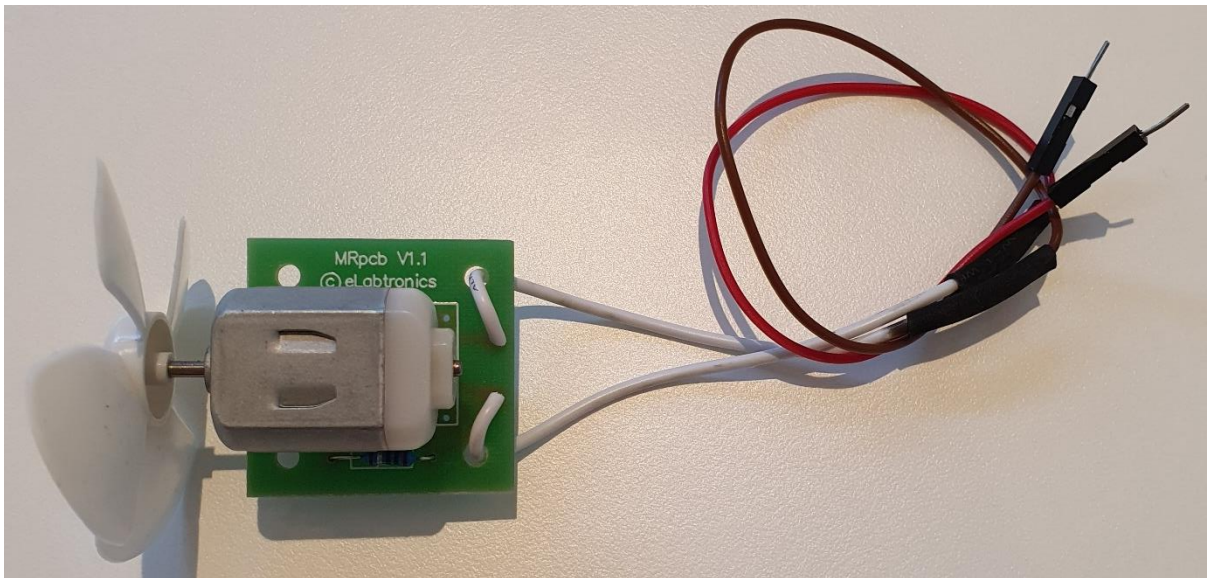


Figure 10: Modified DC motor with jumper plug wires (male ends).

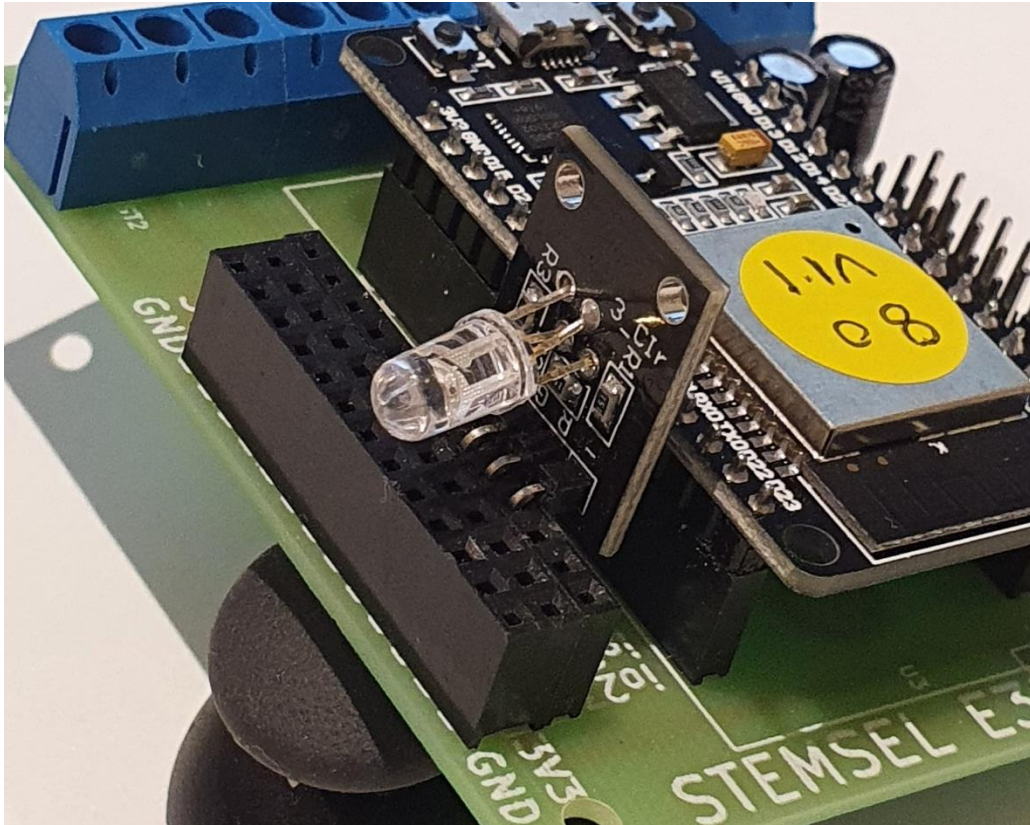


Figure 11: E32W board connection with RGB module (side view).

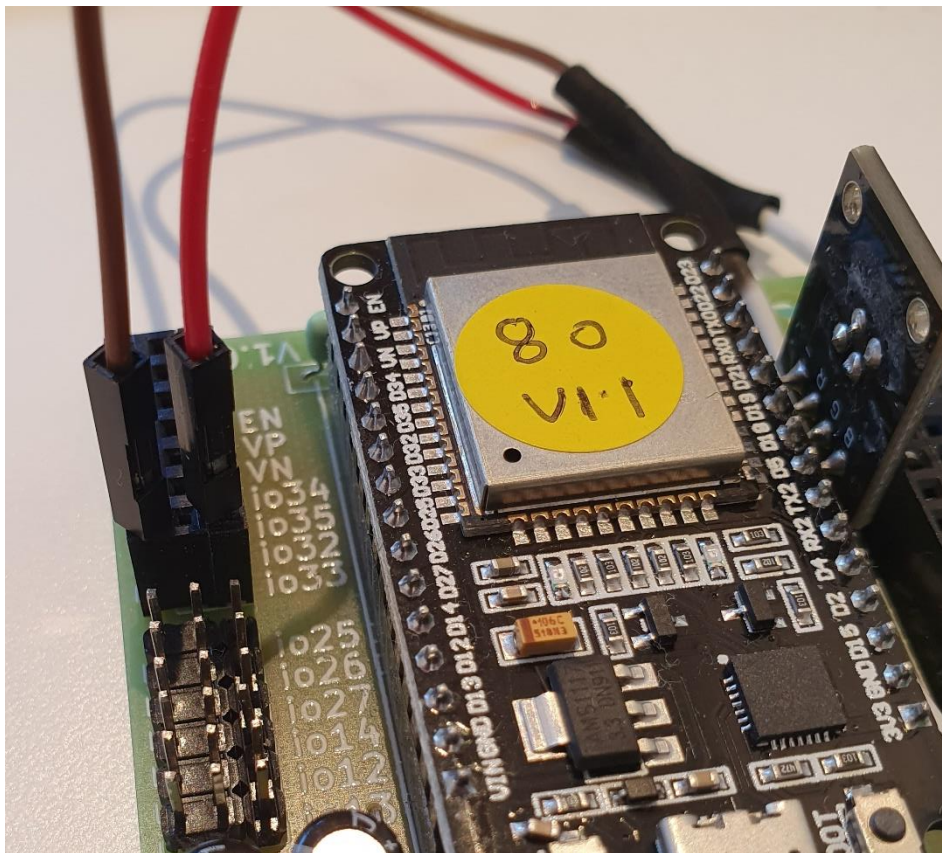


Figure 12: E32W board connection with both DC motor's wires plugged-in at io33 (side view).

runlinc Intermediate Project 3: Wind Power (E32W Version)

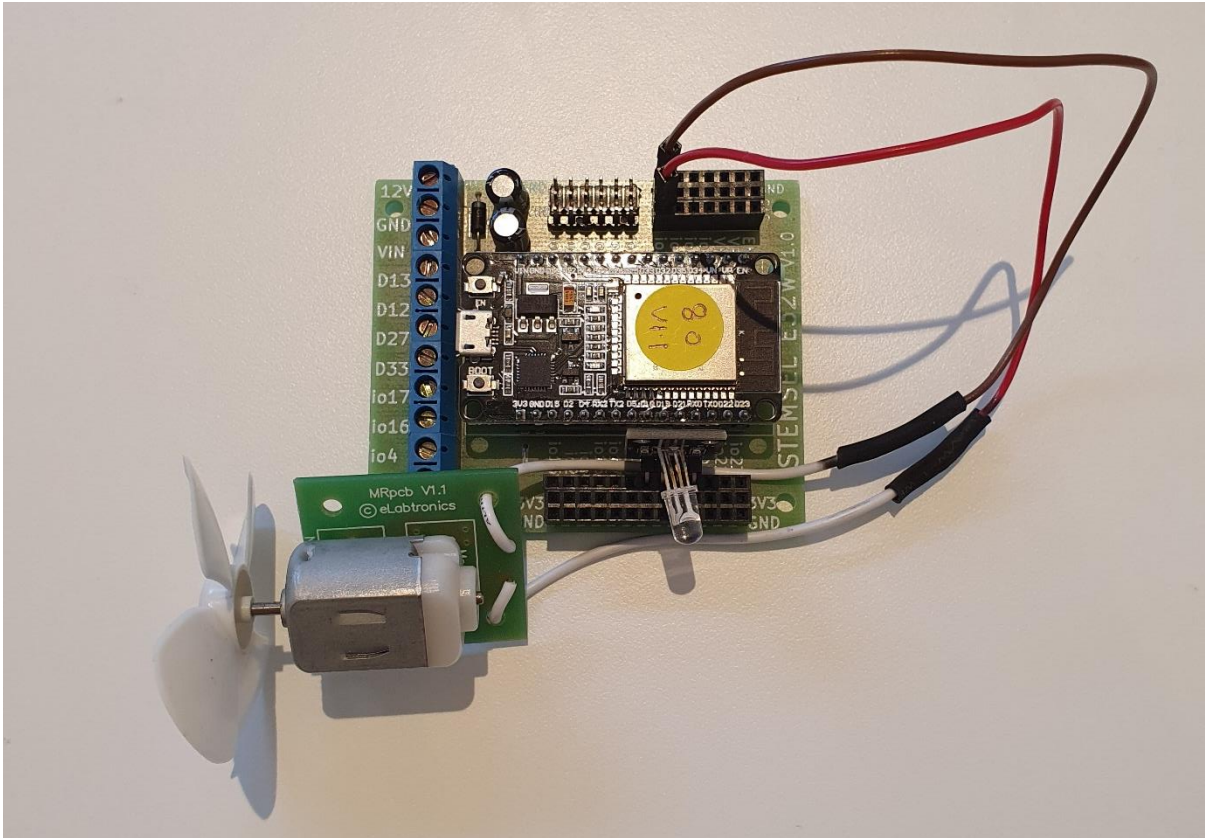


Figure 13: E32W board connection with both I/O parts connected (top view).

Part C: Program the Circuit

HTML:

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

1. Set up the text of the page to be centred:

```
<div style="text-align:center">  
<h1>Wind Power dashboard by runlinc</h1>  
</div>
```

2. Within the <div></div> tags we will setup headings and body texts with respective unique IDs and we'll place a circle which will indicate the same light colour as the LEDs:

```
<p>  
Instantaneous Power: <font id="currPower"></font> Watt  
<br>  
Turbine Status: <font id="status">Retrieving Status</font>  
<br>  
<svg height="25" width="25"><circle id="statusIndicator" r="10" cx="12.5"  
cy="12.5" stroke="black" stroke-width="3" fill="black" />  
</svg>  
<br>  
<br>  
<hr>  
<br>  
Power Generated: <font id="accPower"></font> Watt  
</p>
```

3. Then, after </svg> and the first
 we will add input elements that will help us to easily change the limit of the wind power to meet the threshold of that day:

```
Max Idle Limit: <input id="cap_0" step="0.01" type="number" min="0" max="5"  
value="0.05"><br>  
Near Capacity Limit: <input id="cap_1" step="0.01" type="number" min="0"  
max="5" value="0.3"><br>  
Max Capacity Limit: <input id="cap_2" step="0.01" type="number" min="0"  
max="5" value="0.4"><br>
```

JavaScript Block:

We will initialise three variables that will be used in JavaScript loop.

```
var capacity = [0.05, 0.3, 0.4];  
var totalPower = 0;  
var currentPower = 0;
```

JavaScript Loop:

Then we will program the code into the E32W board using the JavaScript loop.

1. We will first retrieve the limit from the input boxes. It will be used to determine the threshold of each stage of the wind turbine.

```
turnOn( Blue );  
for(var i = 0; i<2; i++){  
  capacity[i] = document.getElementById("cap_" + i).value;  
}
```

2. We will now retrieve the values of the motor and process it such that it is within 0 to 5 range by dividing the value by 51 and is read to 2 decimals.

```
currentPower = analogIn( Turbine );  
currentPower = currentPower / 51;  
showValue = currentPower.toFixed(2);  
document.getElementById("currPower").innerHTML = showValue;
```

3. Next, for the first threshold is when the turbine is barely running. Compare the current power to the first limit which is the limit to indicate it is running. If it is more than the limit, indicate it is running and light up green. Or else, it is less than the limit indicate that it is not running and the circle will be black.

```
if (currentPower > capacity[0]){  
  turnOn( Green );  
  turnOff( Blue );  
  document.getElementById("status").innerHTML = "Turbine spinning at acceptable  
  capacity.";  
  document.getElementById("statusIndicator").style.fill = "green";  
}else{  
  turnOn( Blue );  
  turnOff( Green );  
  document.getElementById("status").innerHTML = "Turbine seems to be not  
  spinning.";  
  document.getElementById("statusIndicator").style.fill = "black";  
}
```

- The second threshold is when the turbine is running near capacity, we will compare the current power to the second limit. If the threshold is met, light up blue on both the circle and the LED and indicate that it is running at the status text. If the threshold isn't met, turn off the blue LED.

```
if (currentPower > capacity[1]){
  turnOff( Blue );
  document.getElementById("status").innerHTML = "Turbine spinning at near
  capacity.";
  document.getElementById("statusIndicator").style.fill = "blue";
} else {
  turnOn( Blue );
}
```

- The third threshold is when the turbine is running at max capacity, we will compare the current power to the third limit. Similar to the last step, light up red and indicate the threshold if the threshold is met or turn off the red LED if it is not met.

```
if (currentPower > capacity[2]){
  turnOn( Red );
  turnOff( Blue );
  document.getElementById("status").innerHTML = "Turbine spinning at max
  capacity.";
  document.getElementById("statusIndicator").style.fill = "red";
} else {
  turnOff( Red );
  turnOn( Blue );
}
```

- And lastly, we will accumulate the power that has been measured. We will show the value to 2 decimal places. We will also add a delay to avoid very fast measurements.

```
totalPower = totalPower + currentPower;
document.getElementById("accPower").innerHTML = totalPower.toFixed(2);
await mSec( 50 );
```

Bonus

What happens if you swap both wires at io33 and then blow air at the fan? Do any of the LEDs turn on? No matter how fast the fan turns, none of the LEDs will turn on. This is because when the electricity flows the other way, the microchip can't detect it (negative current). We found that the faster the fan turn, the more electricity is generated. What other factors affect the amount of electricity made by wind turbines? To find out, follow

this link: <http://environment.nationalgeographic.com.au/environment/global-warming/wind-powerinteractive/>

Summary

By using a magnetic field to interact with the conductive wire, the fan can turn one form of energy into another, either movement energy into electricity, or electrical energy into movement. During this project, we found that the harder we blew on the fan, the faster it spins around, and the more electricity was made, shown by all three LEDs turning on. Wind turbines work in the same way to create electricity without producing greenhouse gases.

Appendix – Expected Results

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

File

Board

Board IP:

ESP32

PORT	CONFIGURATION	NAME	STATUS
D2	DISABLED		
D4	DISABLED		
D5	DISABLED		
D12	DISABLED		
D13	DISABLED		
D14	DISABLED		
D15	DISABLED		
RX2	DISABLED		
TX2	DISABLED		
D18	DIGITAL_OUT	Green	OFF
D19	DIGITAL_OUT	Red	OFF
D21	DIGITAL_OUT	Blue	OFF
D22	DISABLED		
D23	DISABLED		
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	DISABLED		
D33	ANALOG_IN	Turbine	0
D34	DISABLED		
D35	DISABLED		
VP	DISABLED		
RNG1	DISABLED		
RNG2	DISABLED		
VN	DISABLED		

CSS

HTML

```
<div style="text-align: center;">
<h1>Wind Power dashboard by runlinc</h1>
<p>
Instantaneous Power: <font id="currPower"></font> Watt
<br>
Turbine Status: <font id="status">Retrieving Status</font>
<br>
<svg height="25" width="25"><circle id="statusIndicator" r="10" cx="12.5"
cy="12.5" stroke="black" stroke-width="3" fill="black" />
</svg>
<br>
Max Idle Limit: <input id="cap_0" step="0.01" type="number" min="0" max="5"
value="0.05"><br>
Near Capacity Limit: <input id="cap_1" step="0.01" type="number" min="0"
max="5" value="0.3"><br>
Max Capacity Limit: <input id="cap_2" step="0.01" type="number" min="0"
max="5" value="0.4"><br>
<br>
<hr>
<br>
Power Generated: <font id="accPower"></font> Watt
</p>
</div>
```

JavaScript

Select Macro | select a device | Add Macro

```
var capacity = [0.05, 0.3, 0.4];
var totalPower = 0;
var currentPower = 0;
```

JavaScript Loop

Select Macro | select a device | Add Macro

```
for(var i = 0; i<2; i++){
capacity[i] = document.getElementById("cap_" + i).value;
}
currentPower = analogIn( Turbine );
currentPower = currentPower / 51;
showValue = currentPower.toFixed(2);
document.getElementById("currPower").innerHTML = showValue;
if (currentPower > capacity[0]){
turnOn( Green );
document.getElementById("status").innerHTML = "Turbine spinning at acceptable
capacity.";
document.getElementById("statusIndicator").style.fill = "green";
}else{
turnOff( Green );
document.getElementById("status").innerHTML = "Turbine seems to be not
spinning.";
document.getElementById("statusIndicator").style.fill = "black";
}
if (currentPower > capacity[1]){
turnOn( Blue );
document.getElementById("status").innerHTML = "Turbine spinning at near
capacity.";
document.getElementById("statusIndicator").style.fill = "blue";
} else {
turnOff( Blue );
}
if (currentPower > capacity[2]){
turnOn( Red );
document.getElementById("status").innerHTML = "Turbine spinning at max
capacity.";
document.getElementById("statusIndicator").style.fill = "red";
} else {
turnOff( Red );
}
totalPower = totalPower + currentPower;
document.getElementById("accPower").innerHTML = totalPower.toFixed(2);
await mSec( 50 );
```

Figure 14 Expected runlinc result.

Wind Power dashboard by runlinc

Instantaneous Power: Watt
Turbine Status: Retrieving Status



Max Idle Limit:
Near Capacity Limit:
Max Capacity Limit:

Power Generated: Watt

Figure 15 Expected HTML Result.

JavaScript Block:

```
var capacity = [0.05, 0.3, 0.4];  
var totalPower = 0;  
var currentPower = 0;
```

HTML block:

```
<div style="text-align: center;">  
<h1>Wind Power dashboard by runlinc</h1>  
<p>  
Instantaneous Power: <font id="currPower"></font> Watt  
<br>  
Turbine Status: <font id="status">Retrieving Status</font>  
<br>  
<svg height="25" width="25"><circle id="statusIndicator" r="10" cx="12.5" cy="12.5" stroke="black"  
stroke-width="3" fill="black" />  
</svg>  
<br>  
Max Idle Limit: <input id="cap_0" step="0.01" type="number" min="0" max="5" value="0.05"><br>  
Near Capacity Limit: <input id="cap_1" step="0.01" type="number" min="0" max="5"  
value="0.3"><br>  
Max Capacity Limit: <input id="cap_2" step="0.01" type="number" min="0" max="5"  
value="0.4"><br>  
<br>  
<hr>  
<br>  
Power Generated: <font id="accPower"></font> Watt  
</p>  
</div>
```


JavaScript Loop Block:

```
turnOn( Blue );
for(var i = 0; i<2; i++){
capacity[i] = document.getElementById("cap_" + i).value;
}
currentPower = analogIn( Turbine );
currentPower = currentPower / 51;
showValue = currentPower.toFixed(2);
document.getElementById("currPower").innerHTML = showValue;
if (currentPower > capacity[0]){
turnOn( Green );
turnOff( Blue );
document.getElementById("status").innerHTML = "Turbine spinning at acceptable capacity.";
document.getElementById("statusIndicator").style.fill = "green";
}else{
turnOn( Blue );
turnOff( Green );
document.getElementById("status").innerHTML = "Turbine seems to be not spinning.";
document.getElementById("statusIndicator").style.fill = "black";
}
if (currentPower > capacity[1]){
turnOff( Blue );
document.getElementById("status").innerHTML = "Turbine spinning at near capacity.";
document.getElementById("statusIndicator").style.fill = "blue";
} else {
turnOn( Blue );
}
if (currentPower > capacity[2]){
turnOn( Red );
turnOff( Blue );
document.getElementById("status").innerHTML = "Turbine spinning at max capacity.";
document.getElementById("statusIndicator").style.fill = "red";
} else {
turnOff( Red );
turnOn( Blue );
}
totalPower = totalPower + currentPower;
document.getElementById("accPower").innerHTML = totalPower.toFixed(2);
await mSec( 50 );
```