STEMSEL Intermediate Project 2: Radition Counter (runlinc Version)

Problem

When we are in an environment with radiation, we want to make sure the level of the radiation is in the safe range. What can we do to determine the radiation level?

Background

When we think about radiation, we often think of things like nuclear weapons and accidents like those at the Chernobyl or Fukushima nuclear reactors. However, low levels of radiation are all around us every day, from the sun to building materials and even in our food! Brazil nuts and bananas are two foods quite well known for being slightly radioactive. Don't worry about eating too much of such foods though; you could not eat enough of them to cause any health problems (You should always try and eat a good amount of fruit anyways!).

Radiation is also very useful in many applications; many smoke detectors use a tiny amount of radioactive Americium to detect the smoke. Radiation is often used in medicine as well, including in the treatment of some cancers and viewing the inside of the body using X-rays (radiography), radio-opaque dyes, and nuclear medicine procedures. Radioactive substances can also be used in scientific research. Radiography is also used in many industries for quality control or to analyse the internal working of machinery. We also use radioactive elements to power nuclear power plants, submarines and satellites. The Curiosity Mars rover is not only powered by radioactive elements, but also uses them for analysing rock and soil samples it finds on the red planet.

However, radiation can be very dangerous for living things. The effects on the human body of receiving too much radiation vary, depending on which areas of the body were exposed and how much radiation was absorbed. Radiation damages and kills the cells that make up our bodies, and may therefore cause burns (sunburn is caused by UV radiation), loss of white blood cells, haemorrhaging (bleeding), fatigue and weakness, and even cancer. If radiation damages vital organs like the brain, heart, lungs, digestive tract etc., then the effects will be more severe than if only the surface of your skin or your hand was affected.

Despite these dangers people must sometimes enter dangerous radioactive zones. For example, people cleaning up the areas around the Fukushima nuclear reactor in Japan. Or when working in places where radioactive compounds are used or stored such as a research centre or even in outer space. They often carry measuring devices like Geiger counters to help them measure how much radiation they have been exposed to. It is important to consider the level of radiation they have been exposed to, as well as the length of time they were exposed. If there was only a low radiation level, you could stay in the area for a longer time without any negative effects than if the radiation level was higher. For today's project, we will implement a counter to keep track of the total amount of radiation we have been exposed to. We will be using a safer type of radiation though, simply measuring the amount of light.

Note: Yes, (visible) light is a type of radiation. Much like radio waves, microwaves, infrared and x-rays are all radiation and are part of the electromagnetic spectrum. The danger in radiation generally comes with how short the wavelength for the type is. Types with longer 2 wavelengths such as radio waves are safer than gamma waves or x-rays, which have significantly shorter wavelengths and have more energy per packet of light.

Ideas

How can we measure the radiation (light) level? How can the microchip and runlinc tell the user if the level is high or low? How can we make our device count up faster the more radiation is detected? Is a visual warning on a webpage enough, or would 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 can use a LDR as our light sensor, and display the level on the using a webpage.

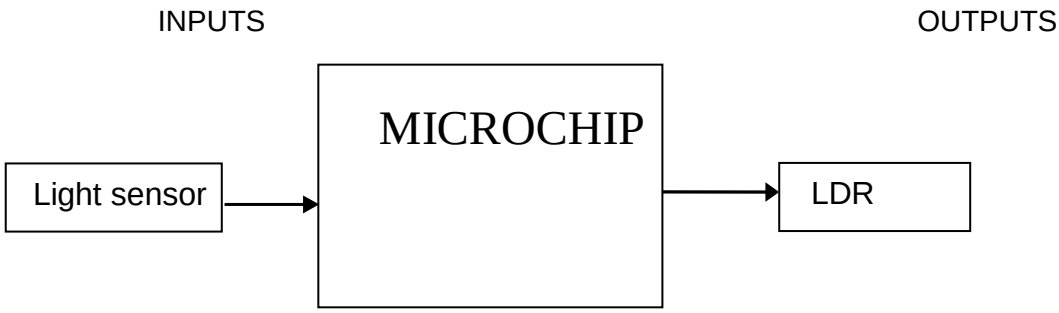

Figure 1 Block diagram of Microchip outputs

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

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

For port C7 name it RAD_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.

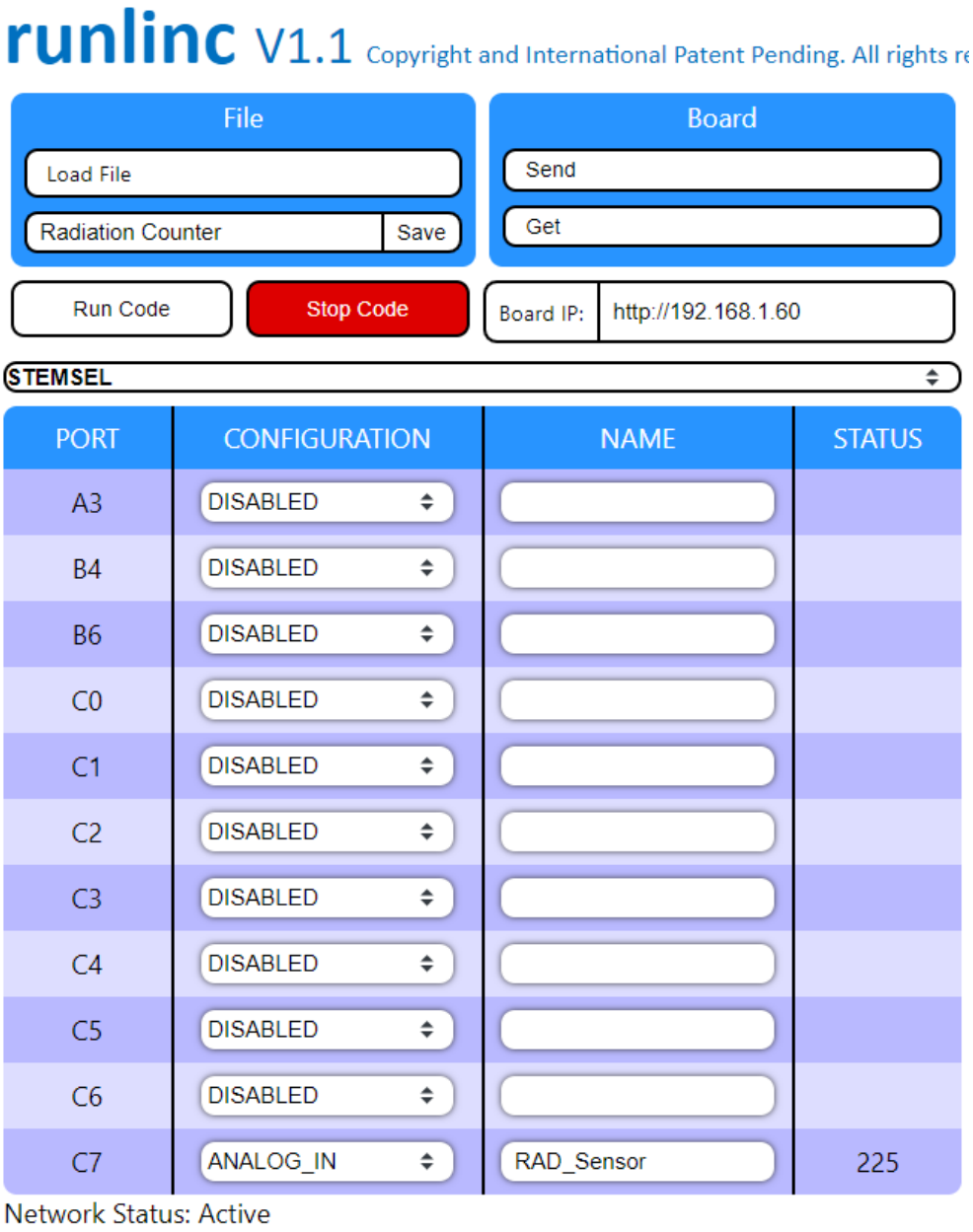

Figure 2 I/O configurations connections

Part B: Build the Circuit

Use the runlinc I/O to connect the hardware. Remember that turning the screws clockwise will close the clamps and turning the screws anticlockwise will open them. All black wires should go in the negative (-) terminal, red wires go in the positive (+) terminal, and white wires go in the terminal we designated in the runlinc web page port.

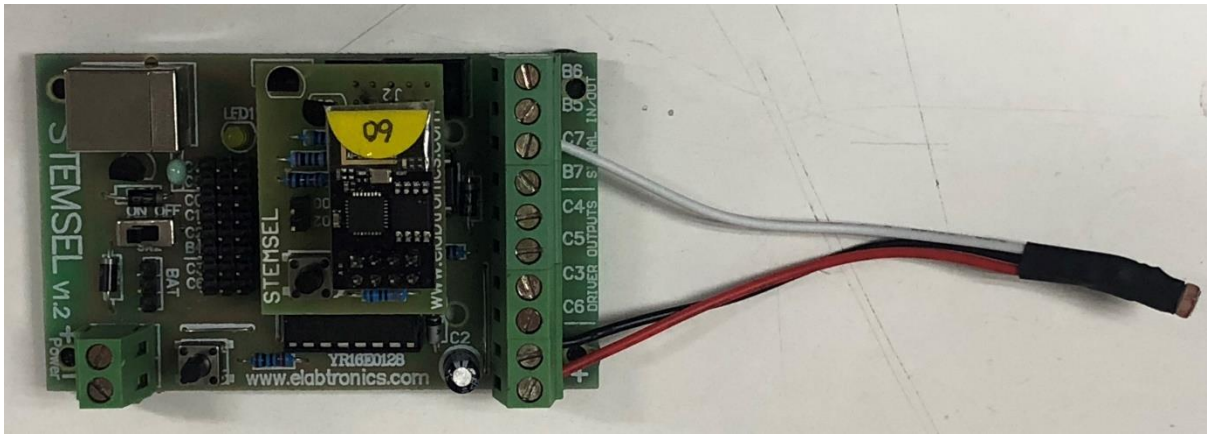

Figure 3 Circuit connection on microcontroller

a.) For Lightsensor connect the white wire to C7, black wire to negative port and red wire to positive port.

Part D: Program the Circuit

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

For **HTML** box type the following:

```
<h2>runlinc Radiation
Counter</h2>
Radioactivity:
<font id=RadNow></font> Curie
```

This will set up our webpage to receive the Information from the RAD_Sensor.

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

```
sense_num = analogIn(RAD_Sensor);
sense_num = sense_num / 51;
Curie = sense_num.toFixed(2);
document.getElementById('RadNow').innerHTML = Curie;
```

Here, a variable is initialized with it containing the input light level from the light sensor.

Then we are sending it to the HTML webpage so it can be displayed.

At this stage save and run the code and have a look at what you get. If it doesn't work go through your code and check for any errors.

Now we can expand on what has been done to now display the radiation level. To do this we will need to add to the HTML script and the JavaScript Loop as well as the JavaScript.

For **HTML** box type the following under what has already been done:

```
<br>
<font face="verdana" id=TotRads></font> Rads
```

For the **JavaScript** box type the following:

```
let sense_num = analogIn(RAD_Sensor);
let total_rads = 0;
```

For the **JavaScript Loop** box type the following under what has already been done:

```
total_rads = total_rads + sense_num;
await mSec( 500 );
document.getElementById('TotRads').innerHTML = total_rads.toFixed(2);
```

Now Save and run the code and see what's new. If it's not working double check your code for any errors, baring any errors refresh the page and load in you program to try again.

Now we're going to create the alerts, to do this we're going to be adding to the HTML box and the JavaScript Loop box.

For **HTML** box type the following under what has already been done:

```
<br>
<font id=Message></font>
```

For the **JavaScript Loop** box type the following under what has already been done:

```
if(sense_num < 2)
document.getElementById('Message').innerHTML = "Too LOW";
else
document.getElementById('Message').innerHTML = "Too HIGH";
```

Once again save the code and run it. If it doesn't work check your code for any errors, if no errors refresh the page and reload your code then try and run it again.

Summary

We are surrounded by radiation every day and although it is useful, radiation can cause serious and ongoing health problems. We need to be careful of how much radiation we are exposed to, so people working in areas with high radiation need to monitor their exposure with devices similar to the one we made in this project.

```
<h2>runlinc Radiation Counter</h2>
Radioactivity:
<font id=RadNow></font> Curie
<br>
<font face="verdana" id=TotRads></font> Rads
<br>
<font id=Message></font>
```

Figure 4 HTML Code

```
let sense_num = analogIn(RAD_Sensor);
let total_rads = 0;
```

Figure 5 JavaScript Code

```
sense_num = analogIn(RAD_Sensor);
sense_num = sense_num / 51;
Curie = sense_num.toFixed(2);
document.getElementById('RadNow').innerHTML = Curie;

total_rads = total_rads + sense_num;
await mSec( 500 );
document.getElementById('TotRads').innerHTML =
total_rads.toFixed(2);

if(sense_num < 2)
document.getElementById('Message').innerHTML = "Too LOW";
else
document.getElementById('Message').innerHTML = "Too HIGH";
```

Figure 6 JavaScript Loop Code

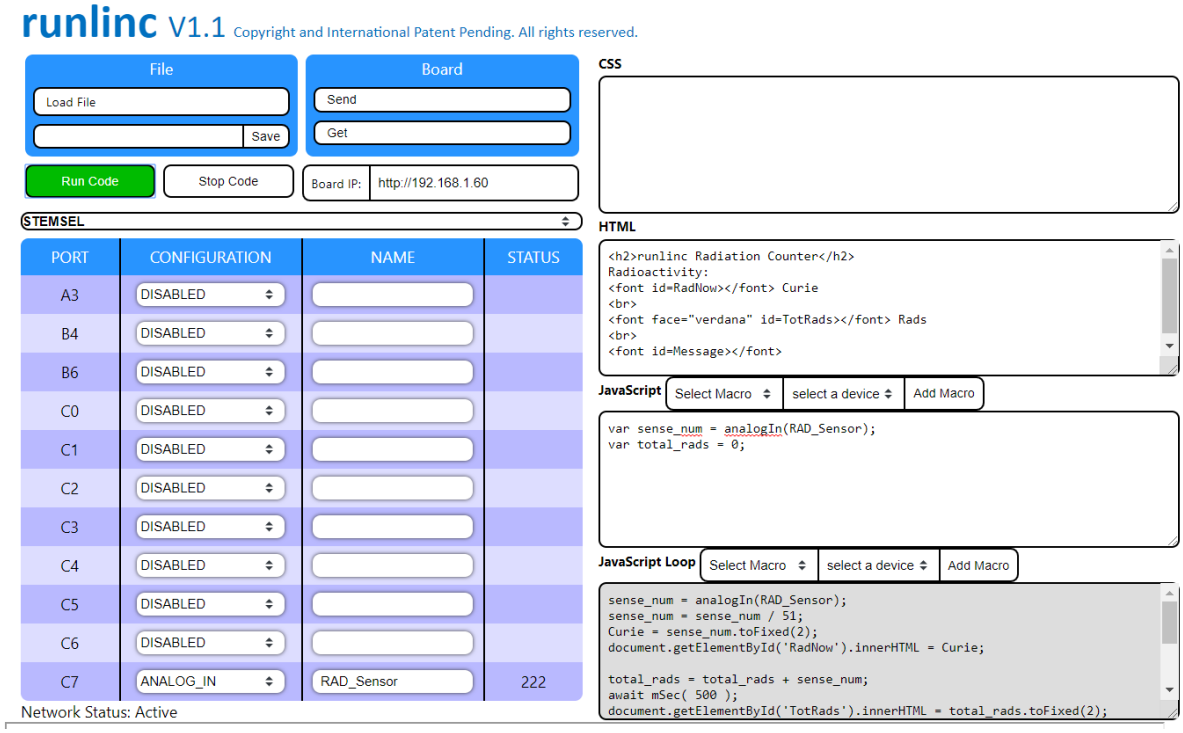

Figure 7 Runlinc Image