

STEMSEL Intermediate Project 4: Program an AI to be able to talk to you or others

(runlinc version)

Problem

How can we use microchips to turn on objects by asking and AI to do it? What if we're too hot and want to cool down?

Background

By learning STEMSEL, you can learn to program microchips to tell it what to do. Microchips can monitor devices and warn people like when your laptop battery is low, and you need to plug the cable in. However, they can also control people's behaviour more directly, such as electronic road signs and traffic lights. But it is important that the microchips send the right message, otherwise there can be some problem like turning on the green lights at the same time.

By using runlinc we can create an AI to do the things we ask it to do or even have conversations with it. By using AI's we can help streamline what?

Ideas

Look at the STEMSEL controller board. Can you see any inputs, i.e. something that we can touch or change to tell the microchip something? What about an output, i.e. something the microchip can change to tell us something? What kind of inputs and outputs are normally on an alarm system? What outputs can we use for our AI can use?

Plan

So we want the runlinc AI to be able to talk to us and others. First we are going to need away to talk to our AI. We'll also need to give our AI information so it can talk and give responses. To do these we'll have to create an input through our webpage, by doing this it will allow our AI to talk to us.

OUTPUTS

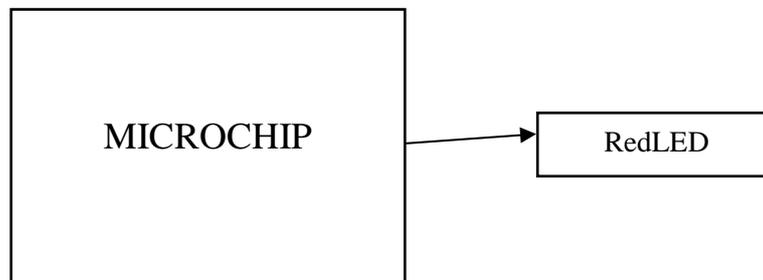


Figure 1 Block diagram of Microchip Outputs

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

Note: refer to runlinc Wi-Fi setup guide document to connect to runlinc

In our circuit design we will be using the RedLED. We happen to have this in our kits, so it can be used on our circuit design, as per the plan.

On the runlinc webpage remember to type in the Names for the equipment we are using

For port C4, we name it RedLED and set it as DIGITAL_OUT

runlinc V1.1 Copyright and International Patent Pending. All rights reserved

File

Board

Board IP:

STEMSEL

PORT	CONFIGURATION	NAME	STATUS
A3	DISABLED	<input type="text"/>	
B4	DISABLED	<input type="text"/>	
B6	DISABLED	<input type="text"/>	
C0	DISABLED	<input type="text"/>	
C1	DISABLED	<input type="text"/>	
C2	DISABLED	<input type="text"/>	
C3	DISABLED	<input type="text"/>	
C4	DIGITAL_OUT	<input type="text" value="RedLED"/>	<input type="button" value="OFF"/>
C5	DISABLED	<input type="text"/>	
C6	DISABLED	<input type="text"/>	
C7	DISABLED	<input type="text"/>	

Figure 2 I/O Configurations and Connection

Part B: Build the Circuit

Use the runline I/O to connect the hardware. Remember that black wires connect to the negative port (-), red wires to the positive port (+) and white wires connect to the pin designated in the circuit design.

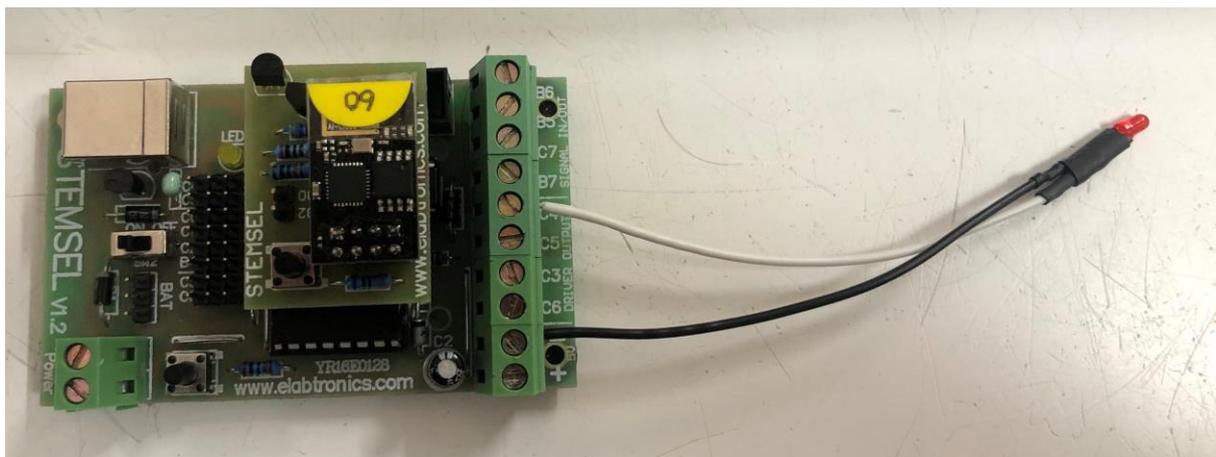


Figure 3 Circuit connection with microship

- a.) Connect the white wire of the RedLED to C4
- b.) Connect the black wire to the negative port (-)

Part C: Program the Circuit

Use the blocks on the right side of the runlinc webpage to program the functions of the traffic light. Use the HTML to add contents, CSS to add style in your favour and Javascript to program the microchip. For this case we will be using HTML and JavaScript to program our AI chat bot. Type the following code.

After naming the port C4. We are going to Program the circuit. First, we need HTML block to add in the graphics for what we are going to do.

First this we'll need to do is give our webpage a title, let's call it My runlinc AI Chat Bot. To do this we'll need to use the Heading tag <h3>.

```
<h3>My runlinc AI Chat Bot</h3>
```

now that we have our title we need to create the text box and a text line saying what it's for. To do this we will use the paragraph tag <p>.

```
<p>Chat to runlinc AI BOT. Type your text...</p>
```

Now that we have set up the text sections we need to add an input text field for a user to type in their conversations.

```
<input type="text" id="myText" value="type here...">
```

Now we need a button to actually send what has been typed in to the AI. So using the button tag:

```
<button onclick="myFunction()">send</button>
```

Make sure to put in a space between the different sections using the
 tag at least 3 times to give a good gap.

Now that we have created our output field we need to create our output field so the AI can respond. First we should make sure that it's clear that it's the AI talking, to do this we'll type in:

```
"runlinc AI says..."
```

Make sure to exclude the " from what you just typed.

Now put in another small break using the
 tag.

Now we can put in the output for the AI:

```
<p id="replyText"></p>
```

PORT	CONFIGURATION	NAME	STATUS
A3	DISABLED		
B4	DISABLED		
B6	DISABLED		
C0	DISABLED		
C1	DISABLED		
C2	DISABLED		
C3	DISABLED		
C4	DIGITAL_OUT	RedLED	OFF
C5	DISABLED		
C6	DISABLED		
C7	DISABLED		

```

HTML
<h3>My runlinc AI Chat Bot</h3>
<p>Chat to runlinc AI BOT. Type your text...</p>
<input type="text" id="myText" value="type here...">
<button onclick="myFunction()">send</button>
<br> <br> <br>
runlinc AI says...
<br>
<p id="replyText"></p>
    
```

Figure 4 HTML code

Now that we have set up all the inputs and outputs we can now look at creating our AI. To do this we'll need to use the JavaScript block.

To start with we will create a function called myFunction and declare a variable called x:

```
function myFunction() {
    var x = document.getElementById("myText").value;
```

Now that we have set up the variable we can now give our AI its responses. To do this it will need to receive what has been typed, then compare it with what it is expecting which will allow it to give a response.

```
var n = x.includes("Hello");
if ( n == true ) {
    x = "Hello to you too!";
    turnOff( RedLED );
}
```

Now we have set up the AI to respond to somebody saying Hello to it the AI will respond. Now we can give some more functions. Let's say that the AI is in danger and that it will then light up the RedLED when it is.

```
var n = x.includes("danger");
if ( n == true ) {
```

```

x = "Oh now Im scared";
turnOn( RedLED );
}

```

Now we should tell it to relax meaning it's no longer in danger.

```

var n = x.includes("relax");
if ( n == true ) {
  x = "OK Thanks. That was close.";
  turnOff( RedLED );
}

```

Now that we have created a basic AI we need it the JavaScript to communicate with the HTML to do this we need to use the innerHTML method:

```

document.getElementById("replyText").innerHTML = x;
}

```

runlinc V1.1 Copyright and International Patent Pending. All rights reserved.

File

Load File

Save

Run Code Stop Code

Board

Send

Get

Board IP: http://192.168.1.60

CSS

HTML

JavaScript

```

function myFunction() {
  var x = document.getElementById("myText").value;

  var n = x.includes("Hello");
  if ( n == true ) {
    x = "Hello to you too!";
    turnOff( RedLED );
  }

  var n = x.includes("danger");
  if ( n == true ) {
    x = "Oh now Im scared";
    turnOn( RedLED );
  }

  var n = x.includes("relax");
  if ( n == true ) {
    x = "OK Thanks. That was close.";
    turnOff( RedLED );
  }

  document.getElementById("replyText").innerHTML = x;
}

```

JavaScript Loop

STEMSEL

PORT	CONFIGURATION	NAME	STATUS
A3	DISABLED	<input type="text"/>	
B4	DISABLED	<input type="text"/>	
B6	DISABLED	<input type="text"/>	
C0	DISABLED	<input type="text"/>	
C1	DISABLED	<input type="text"/>	
C2	DISABLED	<input type="text"/>	
C3	DISABLED	<input type="text"/>	
C4	DIGITAL_OUT	RedLED	OFF
C5	DISABLED	<input type="text"/>	
C6	DISABLED	<input type="text"/>	
C7	DISABLED	<input type="text"/>	

Network Status: Active

Figure 5 JavaScript code

For **HTML** the code is:

```
<h3>My runlinc AI Chat Bot</h3>

<p>Chat to runlinc AI BOT. Type your text...</p>

<input type="text" id="myText" value="type here...">

<button onclick="myFunction()">send</button>

<br> <br> <br>

runlinc AI says...

<br>

<p id="replyText"></p>
```

For **JavaScript** the code is:

```
function myFunction() {
    var x = document.getElementById("myText").value;

    var n = x.includes("Hello");
    if ( n == true ) {
        x = "Hello to you too!";
        turnOff( RedLED );
    }

    var n = x.includes("danger");
    if ( n == true ) {
        x = "Oh now Im scared";
        turnOn( RedLED );
    }
}
```

```
var n = x.includes("relax");  
if ( n == true ) {  
  x = "OK Thanks. That was close.";  
  turnOff( RedLED );  
}  
  
document.getElementById("replyText").innerHTML = x;  
}
```

Extensions

Now that we have created the core code we can now look at expanding it to do more than just talk.

The First challenge is to make it have a full conversation with you or somebody else, even turning on and off other devices like a fan. You can achieve this by copying and pasting the code from one of the var declarations and changing the expected input and outputs to whatever you want.

The Second Challenge is to get the AI to talk. Here a segment of code that could help:

```
speech = new SpeechSynthesisUtterance("Here is a square.")  
window.speechSynthesis.speak(speech);
```

The Third challenge is to get it to draw shapes when it's asked to. To give you a hand here is a small section of code that might just help (goes in the JavaScript Block:

```
if (x.includes("square")) {  
  speech = new SpeechSynthesisUtterance("Here is a square.")  
  window.speechSynthesis.speak(speech);  
  x = "Here is a square.";  
  var canvas = document.getElementById('canvas');  
  var ctx = canvas.getContext('2d');  
  ctx.strokeStyle = 'black'
```

```

ctx.beginPath();
ctx.strokeRect(520,0,100,100);
ctx.stroke();
}

```

And this line of code should go into the HTML block:

```
<canvas id="canvas" height=1000 width=1000></canvas>
```

Summary

People can use programming to tell an AI what to do. However, sometimes those AI's can be programmed to have conversations with people, so it is important to program them correctly. In this project, we learned that we can an AI to talk to people and turn on and off devices.

runlinc V1.1 Copyright and International Patent Pending. All rights reserved.

PORT	CONFIGURATION	NAME	STATUS
A3	DISABLED		
B4	DISABLED		
B6	DISABLED		
C0	DISABLED		
C1	DISABLED		
C2	DISABLED		
C3	DISABLED		
C4	DIGITAL_OUT	RedLED	OFF
C5	DISABLED		
C6	DISABLED		
C7	DISABLED		

```

function myFunction() {
  var x = document.getElementById("myText").value;

  var n = x.includes("Hello");
  if ( n == true ) {
    x = "Hello to you too!";
    turnOff( RedLED );
  }

  var n = x.includes("danger");
  if ( n == true ) {
    x = "Oh now I'm scared";
    turnOn( RedLED );
  }

  var n = x.includes("relax");
  if ( n == true ) {
    x = "OK Thanks. That was close.";
    turnOff( RedLED );
  }

  document.getElementById("replyText").innerHTML = x;
}

```

Figure 6 runlinc webpage screenshot