runlinc Beginners Project 1:
Alarm System (STEMSEL Version)

Problem

- How can we use microchips to protect our valuables?
- How could a chip know if there is a thief and what can it do when it detects one?
- How can runlinc communicate with the microchip to implement these commands?

Background

Have you heard of a microchip before? Do you know what microchip means? Micro means very small, and a chip is a small piece, like a potato chip or a wood chip. A microchip is actually a circuit that has been fitted into a tiny piece of silicon, and people can use that tiny silicon circuit to form a tiny computer which can store a list of instructions to make the microchip do many different things. Microchips are all around, in everything from computers to cars, credit cards to toys. One thing that microchips are very useful for is alarm systems, since unlike a person the microchip can't get bored of watching the same thing for hours on end.

Alarm Systems are used to protect a wide variety of valuables, from a car to household belongings. Alarms use sensors which can be a simple switch or more complex heat, light and motion sensors. A key component of an alarm system is a microchip to gather sensor information and decide when to activate the alarm.

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 inputs and outputs can we use on our alarm system?

runlinc Background

Runlinc is a web page inside a Wi-Fi chip. The programming is done in the browser and sent to the chip over Wi-Fi. 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.

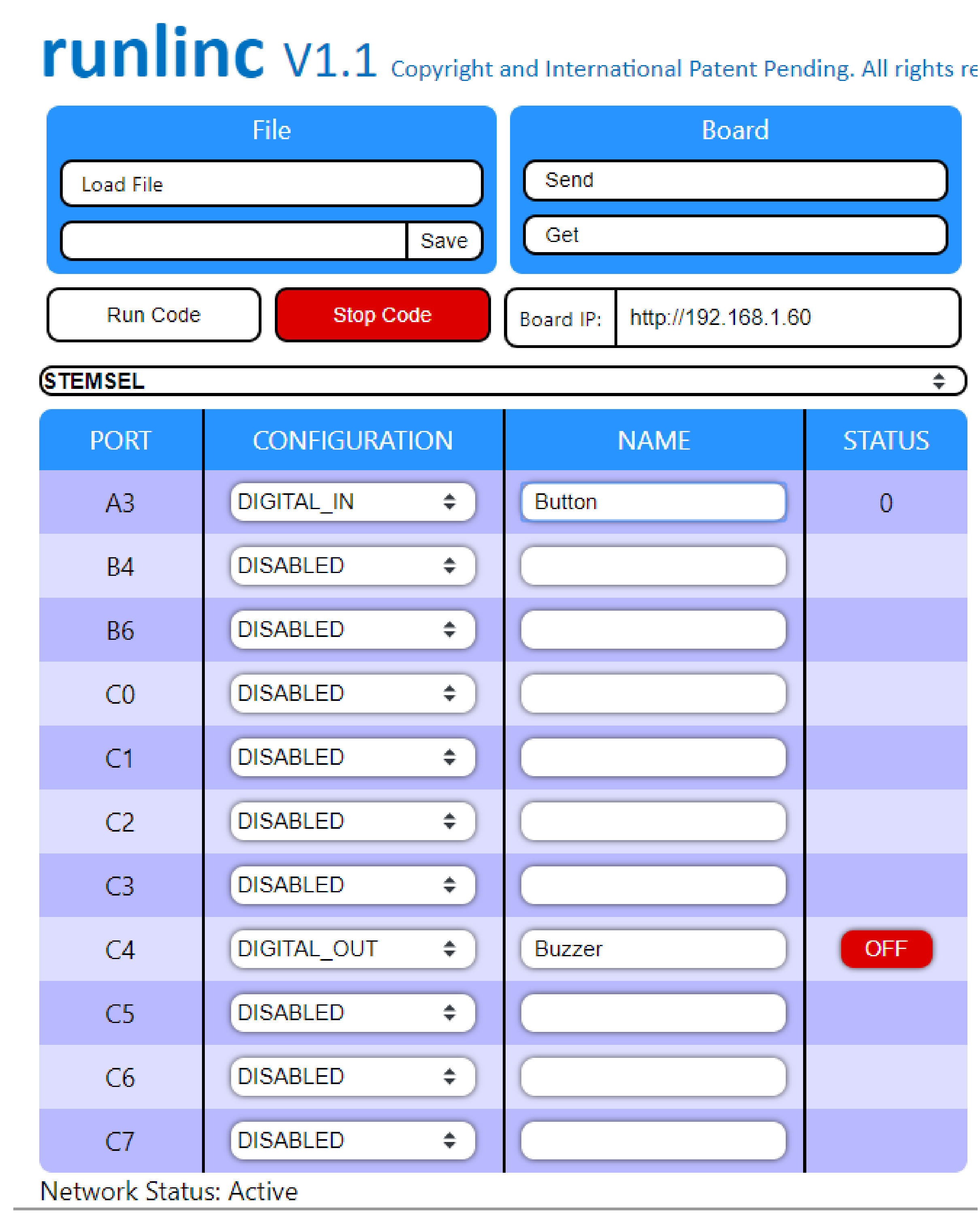
Part A: Design the Circuit on runlinc

Note: Refer to runlinc Wi-Fi Setup Guide document to connect to runlinc

Using the left side of the runlinc web page to construct an input/output (I/O). The ports C4, C5, C3 and C6 are output ports, whereas the ports B6, B5, C7 and B7 are input ports.

For the port A3, name it as Button and set it as DIGITAL_IN.

A buzzer will be added to C4, which can be named as Buzzer and set to DIGITAL_OUT.



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

Board IP: http://192.168.1.60

STEMSEL

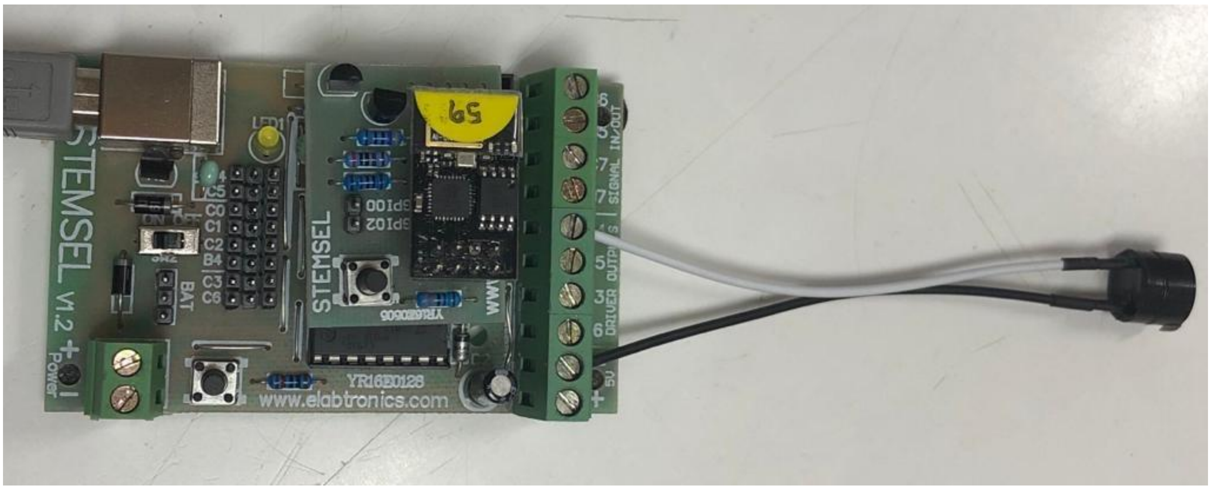
PORT	CONFIGURATION	NAME	STATUS
A3	DIGITAL_IN	Button	0
B4	DISABLED		
B6	DISABLED		
C0	DISABLED		
C1	DISABLED		
C2	DISABLED		
C3	DISABLED		
C4	DIGITAL_OUT	Buzzer	OFF
C5	DISABLED		
C6	DISABLED		
C7	DISABLED		

Network Status: Active

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.

In our designed circuit, it requires to use a buzzer on the microcontroller, which we already have in our STEMSEL kits. Thus, we will add the buzzer to our circuit as planned.



a.) Connect the white wire of the Buzzer to C4

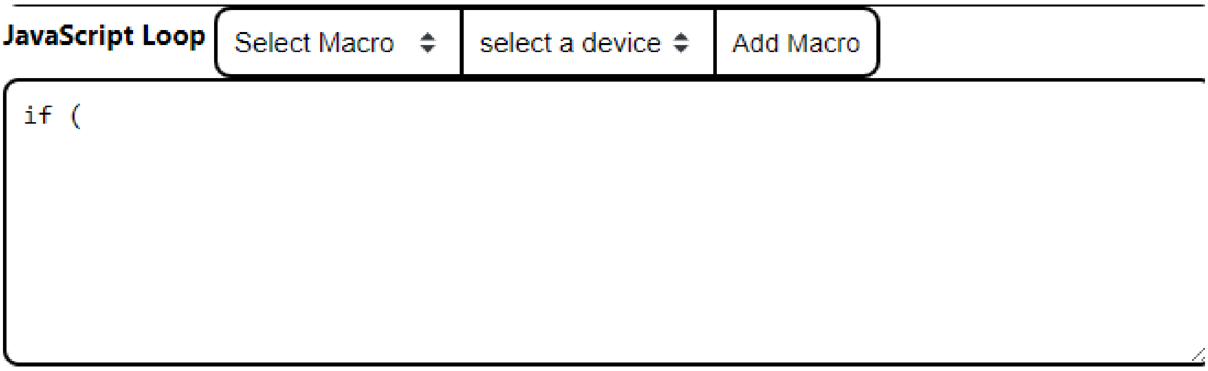
b.) Connect the black wire to the negative port (-)

Part C: Program the Circuit

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

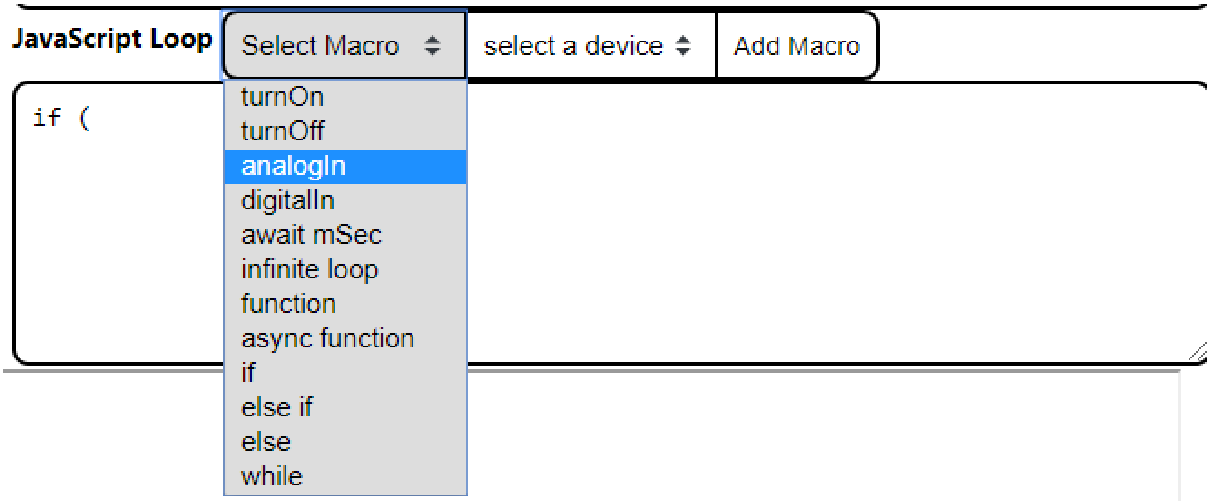
First, we need to declare an if statement. To do this, typing the following code in the first row:

if (



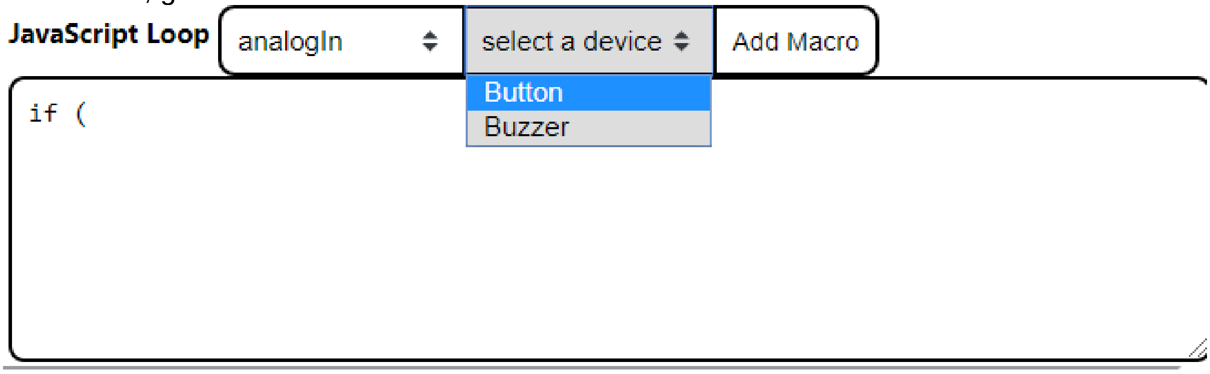
```
if (
```

Next, we need to add a macro for the button. To do this, go to the select macro button and choose analogIn.



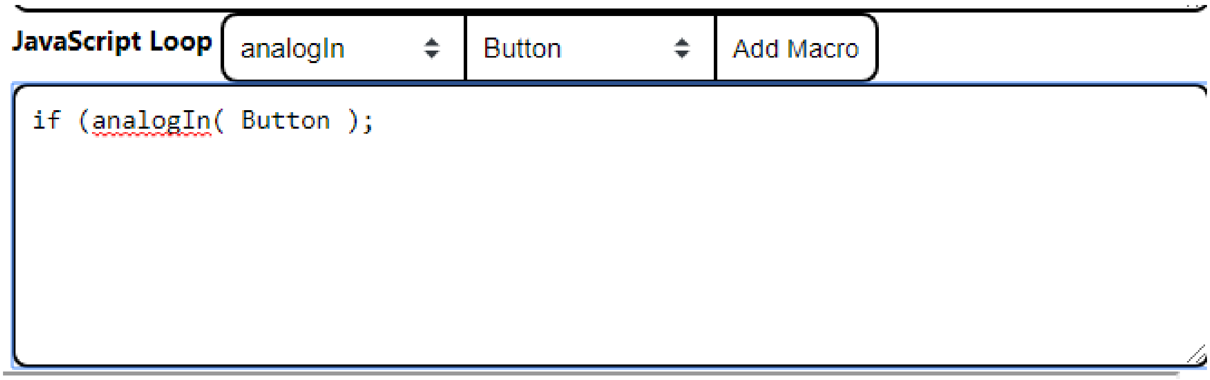
```
if (
```

Afterwards, go to the select device button and choose Button.



```
if (
```

Click Add Macro to add the macro to the code.

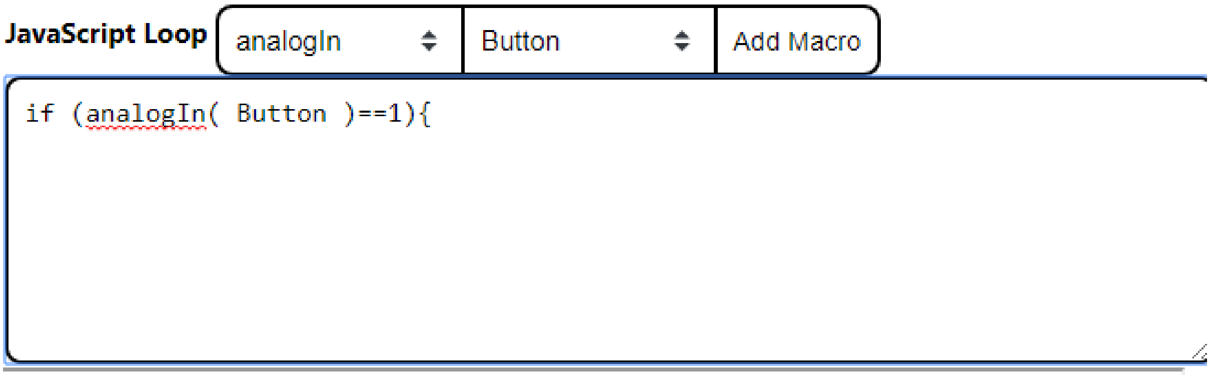


```
if (analogIn( Button );
```

To complete the if statement, delete the semicolon and replace it with the following code snippet: ==1){

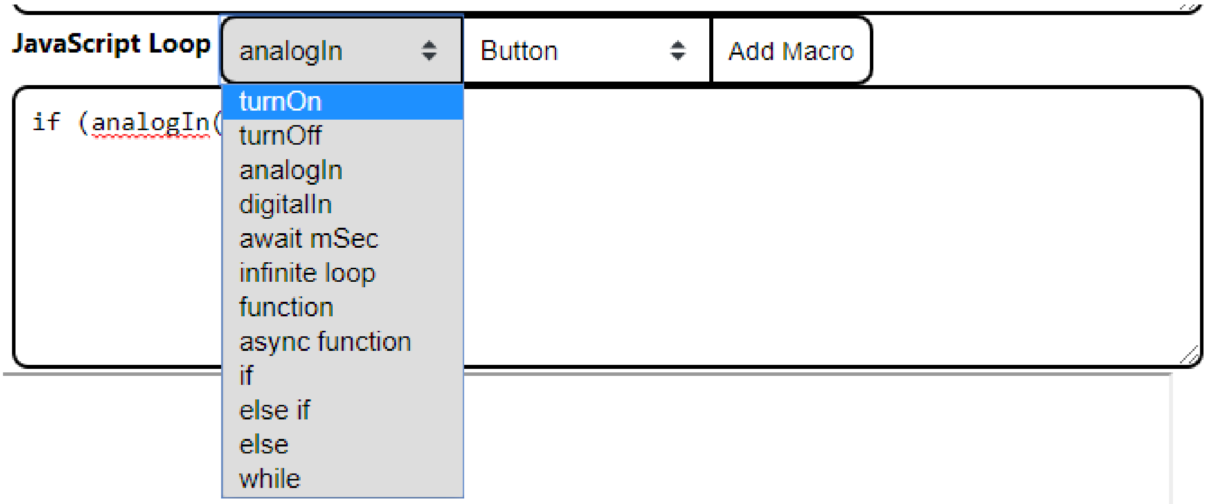
This makes sure that the microchip will determine if the button is pressed or not all the time.

.



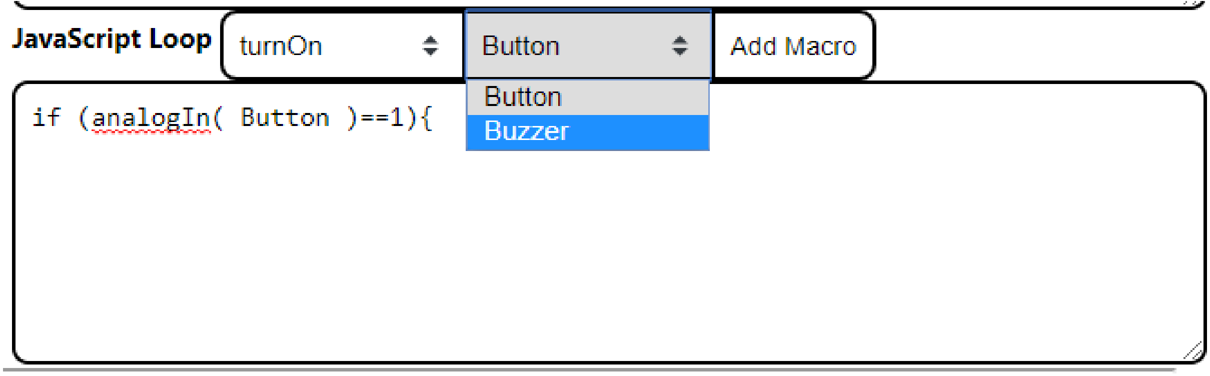
```
if (analogIn( Button )==1){
```

At this stage, we need to add an output which is the buzzer, by using a macro. Go to the select macro button and choose turnOn.



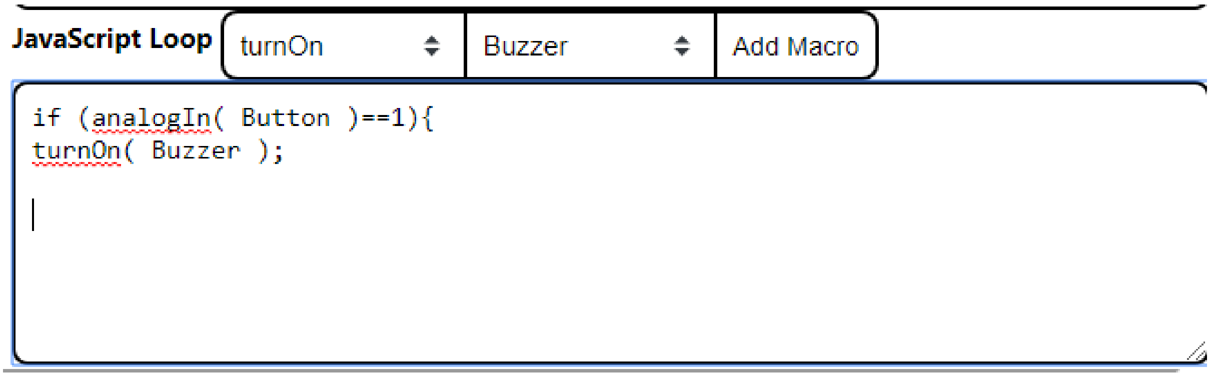
```
if (analogIn(
```

Now, select device and choose Buzzer.



```
if (analogIn( Button )==1){
```

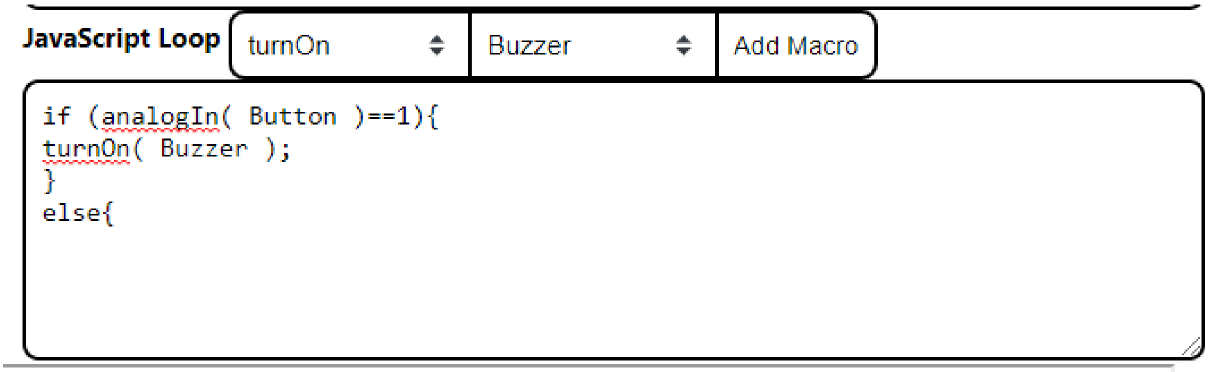
Now add the Macro.



```
if (analogIn( Button )==1){
turnOn( Buzzer );
```

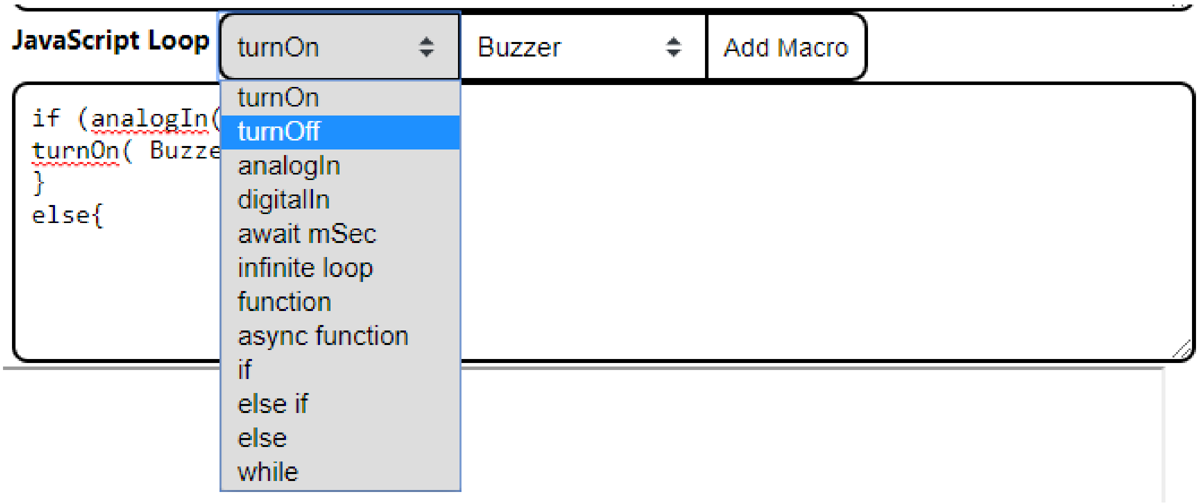
Type in the following code snippet in next line:

}
else{



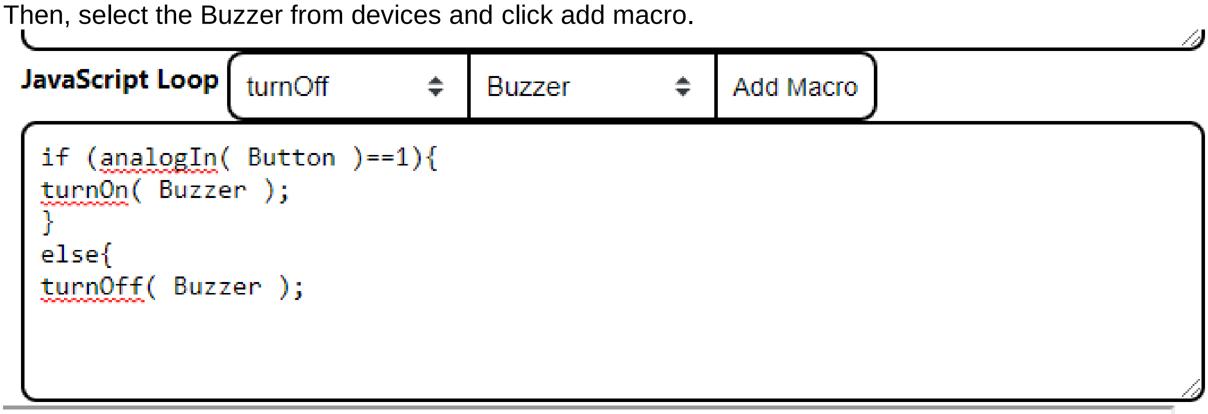
```
if (analogIn( Button )==1){
turnOn( Buzzer );
}
else{
```

As we don't want the buzzer going off constantly we need to add a turnOff macro to turn off the buzzer. To do this select the turnOff macro from the select macro button.



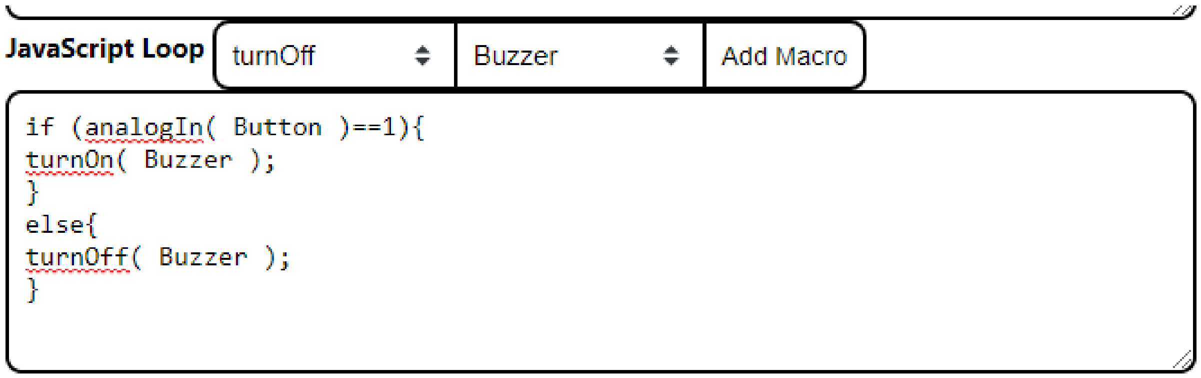
```
if (analogIn(
turnOn( Buzze
}
else{
```

Then, select the Buzzer from devices and click add macro.



```
if (analogIn( Button )==1){
turnOn( Buzzer );
}
else{
turnOff( Buzzer );
```

At the very end of the code, the program needs a closing bracket "}".



```
if (analogIn( Button )==1){
turnOn( Buzzer );
}
else{
turnOff( Buzzer );
}
```

For **JavaScriptLoop**
the whole script should like below:

```
if( digitalIn( Button )==1){
  turnOn( Buzzer );
}
else {
  turnOff( Buzzer );
}
```

Summary

Microchips are basically small and simple computers that we can use to do many different things. We use microchips every day in many devices, such as alarm systems. In this project, we built an alarm system that used a pushbutton from the runlinc webpage as input and would activate a buzzer to let us know that the alarm had been triggered. This project should have given you a good background about how to program the microchip on the STEMSEL controller board via runlinc.