



## runlinc Intermediate Project 5: Vending Machine (STEMSEL version)

---

### Contents

Introduction .....	1
Part A: Design the Circuit on runlinc.....	2
Part B: Program the Circuit .....	3
Extensions .....	8
Summary .....	8

### Introduction

#### Problem

How does a vending machine work? How can we use a website to simulate a vending machine? How can we convey to a customer what they are purchasing, and what is in their shopping cart?

#### Background

Vending machines are commonly seen at schools, bus stations and parks etc. Products in the vending machine can be purchased by simply selecting the product, then inserting the correct amount of money. Then the machine will calculate how much change should be returned, and in the meantime, the product will be ejected to the customer for collection. In this project, we will use a website to select the products and complete the order. We will use the speech synthesiser on the browser to tell the user what they are purchasing and how much it costs when they ask for a summary.

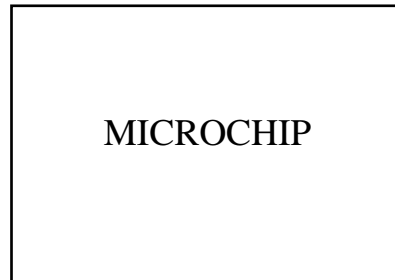
#### Ideas

How can we record what items the user wants to purchase? How can the vending machine know when the order is complete? How can the vending machine respond to the user?

## Plan

For the Vending machine, we can use buttons that when clicked, will add an item to a virtual basket. Once the order has been completed, another button can be used to add up and display a summary of how much money is due.

## OUTPUTS



*Figure 1 Block diagram of Microchip Outputs*

No outputs are used

## 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

In our circuit design, we will be using no outputs.

## Part B: Program the Circuit

Use the blocks on the right side of the runlinc webpage to program the functions of the vending machine. Use the HTML to add content, CSS to add style to your taste and JavaScript to program the microchip. For this case, CSS, HTML and JavaScript are needed to program runlinc to act as out vending machine. Type the following code.

In the CSS block, we are going to want to move to make sure the button is in the right spot for what we want:

```
button{
    margin: 0.5rem;
}
```

The screenshot shows the runlinc V1.1 web interface. At the top left, there is a 'File' section with 'Load File' and 'Save' buttons, and a 'Board' section with 'Send' and 'Get' buttons. Below these are 'Run Code' and 'Stop Code' buttons, and a 'Board IP' field showing 'http://192.168.1.60'. A 'STEMSEL' dropdown menu is visible. The main area contains a table with columns 'PORT', 'CONFIGURATION', 'NAME', and 'STATUS'. The 'CONFIGURATION' column has 'DISABLED' dropdowns for ports A3 through C7. The 'NAME' column has empty input fields. To the right of the table are three code blocks: 'CSS' containing the code 'button{ margin: 0.5rem; }', 'HTML' (empty), and 'JavaScript Loop' (empty). Below the JavaScript Loop are 'Select Macro', 'Select Device', and 'Add Macro' buttons. At the bottom left, it says 'Network Status: Active'.

Figure 2 CSS code block

Now that the CSS is set up for the button, we can now create the buttons in the HTML block:

```
<button onclick="addToBasket(this.innerHTML.trim(), 4)">
    Peanut M&Ms
</button>
```

This code will allow you to create a button with the name “Peanut M&Ms”, it also declares that when the button is clicked an item will be added to the basket.

Let’s make the next button:

```
<button onclick="addToBasket(this.innerHTML.trim(), 5)">
  Unsweetened Orange Juice
</button><br>
```

Now we can create the Summarise button:

```
<button onclick="summarise()">Summarise Order</button>
```

Now we can create the output to show the orders details:

```
<div id="order-details"></div>
```

runlinc v1.1 Copyright and International Patent Pending. All rights reserved.

File: Load File, Save; Board: Send, Get; Board IP: http://192.168.1.60; Run Code, Stop Code

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

Network Status: Active

HTML Code Block:

```
<button onclick="addToBasket( this.innerHTML.trim(), 4 )" >
  Peanut M&Ms
</button>
<button onclick="addToBasket( this.innerHTML.trim(), 5 )" >
  Unsweetened Orange Juice
</button><br/>
<button onclick="summarise()">Summarise Order</button>

<div id="order-details"></div>
```

Figure 3 HTML Code block

Now that we have set up the buttons in HTML, we can now program our vending machine.

In the JavaScript block type the following:

```
function speak( text ){
  const speech = new SpeechSynthesisUtterance( text );
  window.speechSynthesis.speak( speech );
}
//Above will give our vending machine a voice.

var total = 0;
var basket = {};
function summarise(){
  speak(
    'Your items include the following: ' + (
      Object.entries( basket ).map(
        ([name, count]) => `${count} ${name}, `
      ).join("")
    )
  );
  speak( `The total price is ${total} dollars` );
}
function addToBasket( itemName, price ){
  if( basket[ itemName ]){
    basket[ itemName ]++;
  }else{
    basket[ itemName ] = 1;
  }
  total += price;
  updateDetails();
}
function updateDetails(){
  document.querySelector( '#order-details' ).innerHTML = (
    `total = $$${total}<br/>` +
    `items = ${JSON.stringify(basket)}`
  );
}
```

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

**File**

**Board**

Board IP: http://192.168.1.77

**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	DISABLED	<input type="text"/>	
C5	DISABLED	<input type="text"/>	
C6	DISABLED	<input type="text"/>	
C7	DISABLED	<input type="text"/>	
GPIO0	DISABLED	<input type="text"/>	

Network Status: Active

**JavaScript**

```

function speak( text ){
  const speech = new SpeechSynthesisUtterance( text );
  window.speechSynthesis.speak( speech );
}
//Above will give our vending machine a voice.

var total = 0;
var basket = {};
function summarise(){
  speak(
    'Your items include the following: ' + (
      Object.entries( basket ).map(
        ([name, count]) => `${count} ${name}, `
      ).join('')
    )
  );
  speak( `The total price is ${total} dollars` );
}
function addToBasket( itemName, price ){
  if( basket[ itemName ]){
    basket[ itemName ]++;
  }else{
    basket[ itemName ] = 1;
  }
  total += price;
  updateDetails();
}
function updateDetails(){
  document.querySelector( '#order-details' ).innerHTML = (
    `total = ${total}<br/>` +
    `items = ${JSON.stringify(basket)}`
  );
}
        
```

**JavaScript Loop**

Figure 4 JavaScript block

## Final Code:

For **CSS** the code is:

```

button{
  margin: 0.5rem;
}
        
```

For **HTML** the code is:

```

<button onclick="addToBasket( this.innerHTML.trim(), 4 )">
  Peanut M&Ms
</button>
<button onclick="addToBasket( this.innerHTML.trim(), 5 )">
  Unsweetened Orange Juice
</button><br/>
<button onclick="summarise()">Summarise Order</button>

<div id="order-details"></div>
        
```

For **JavaScript** the code is:

```
function speak( text ){
  const speech = new SpeechSynthesisUtterance( text );
  window.speechSynthesis.speak( speech );
}

var total = 0;
var basket = {};
function summarise(){
  speak(
    'Your items include the following: ' + (
      Object.entries( basket ).map(
        ([name, count]) => `${count} ${name}, `
      ).join("")
    )
  );
  speak( `The total price is ${total} dollars` );
}

function addToBasket( itemName, price ){
  if( basket[ itemName ] ){
    basket[ itemName ]++;
  }else{
    basket[ itemName ] = 1;
  }
  total += price;
  updateDetails();
}

function updateDetails(){
  document.querySelector( '#order-details' ).innerHTML = (
    `total = $$${total}<br/>` +
    `items = ${JSON.stringify(basket)}`
  );
}
```

**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	DISABLED ▼	<input type="text"/>	
C5	DISABLED ▼	<input type="text"/>	
C6	DISABLED ▼	<input type="text"/>	
C7	DISABLED ▼	<input type="text"/>	
GPIO0	DISABLED ▼	<input type="text"/>	

Network Status: Active

**CSS**

```

button{
margin: 0.5rem;
}

```

**HTML**

```

<button onclick="addToBasket( this.innerHTML.trim(), 4 )">
Peanut M&Ms
</button>
<button onclick="addToBasket( this.innerHTML.trim(), 5 )">
Unsweetened Orange Juice
</button><br/>
<button onclick="summarise()">Summarise Order</button>

<div id="order-details"></div>

```

**JavaScript**

Select Macro ▼
Select Device ▼
Add Macro

```

function speak( text ){
const speech = new SpeechSynthesisUtterance( text );
window.speechSynthesis.speak( speech );
}
//Above will give our vending machine a voice.

var total = 0;
var basket = {};
function summarise(){
speak(
'Your items include the following: ' + (
Object.entries( basket ).map(
([name, count]) => `${count} ${name}, `
).join('')
)
);
speak( `The total price is ${total} dollars` );
}
function addToBasket( itemName, price ){
if( basket[ itemName ]){
basket[ itemName ]++;
}else{
basket[ itemName ] = 1;
}
total += price;
updateDetails();
}
function updateDetails(){
document.querySelector( '#order-details' ).innerHTML = (
`total = ${total}<br/>` +
`items = ${JSON.stringify(basket)}`
);
}

```

Figure 5 runlinc webpage image

## Extensions

Now that the main code is done, we can look at the ways to expand our vending machine to make it better.

1. With the code done for the items, we can look at creating more items that might be found inside of a vending machine.
2. Use HTML to make a vending machine look, with the buttons lined up
3. Add some colours to the machine
4. Is there a way to add in cents?

## Summary

In this project, we have learnt how to program and create a virtual vending machine that will remember what you have selected, adding up the prices for a total.