



runlinc Intermediate Project 5: Vending Machine (E32W Version)

Contents

Introduction.....	1
Part A: Design the Circuit on runlinc	3
Part B: Program the Circuit	4
Expected Result	6
Extension	7
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.

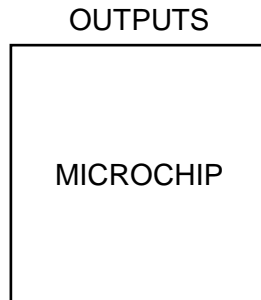


Figure 1: Block diagram of Microchip outputs (no physical outputs are used)

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

Since we have no outputs, therefore we do not need to design our circuit.

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.

CSS:

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;
}
```

HTML:

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>
```

Now we need to set variable:

runlinc Intermediate Project 5: Vending Machine (E32W Version)

```
<script>
var total = 0;
var basket = {};
</script>
```

Full HTML code:

```
<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>
<script>
var total = 0;
var basket = {};
</script>
```

JavaScript:

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 );
}

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)}`
  );
}
```

Expected Result

Peanut M&Ms

Unsweetened Orange Juice

Summarise Order

Figure 2: Expect webpage screenshot

runlinc Intermediate Project 5: Vending Machine (E32W Version)

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

File

Board

Board IP:

ESP32

PORT	CONFIGURATION	NAME	STATUS
D2	DISABLED	<input type="text"/>	
D4	DISABLED	<input type="text"/>	
D5	DISABLED	<input type="text"/>	
D12	DISABLED	<input type="text"/>	
D13	DISABLED	<input type="text"/>	
D14	DISABLED	<input type="text"/>	
D15	DISABLED	<input type="text"/>	
RX2	DISABLED	<input type="text"/>	
TX2	DISABLED	<input type="text"/>	
D18	DISABLED	<input type="text"/>	
D19	DISABLED	<input type="text"/>	
D21	DISABLED	<input type="text"/>	
D22	DISABLED	<input type="text"/>	
D23	DISABLED	<input type="text"/>	
D25	DISABLED	<input type="text"/>	
D26	DISABLED	<input type="text"/>	
D27	DISABLED	<input type="text"/>	
D32	DISABLED	<input type="text"/>	

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

```
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)}`
);
}
```

JavaScript Loop

Figure 3: Expect runlinc result screenshot

Extension

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. This program works similar to a real-life vending machine, and can also be applied to online shopping, where the user select a number of items, and the total cost are displayed in shopping cart.