

STEMSEL Intermediate Project 4: Vending Machine

(runlinc version)

Problem

How does a vending machine work? How can we use a POT to adjust the amount of money that a customer wants to insert? How can we use runlinc to display more than one source of information at once?

Background

Vending machines are commonly seen at schools, bus stations and parks etc. Products in the vending machine can be purchased by simply 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 for the customer to collect. In this project, a potentiometer will be used to adjust the amount of money inserted, a display will show the amount of money inserted each time and the microchip will keep a count of how much money has been input so far. Due to the complexity of this project change will not be returned instead put towards the purchase of additional items.

Ideas

How can we calculate the remaining amount of money? When should the webpage display the amount of money? How can the vending machine know if the amount of money inserted reaches the required amount? How does the vending machine indicate this?

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

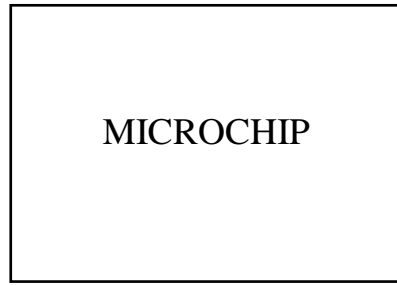


Figure 1 Block diagram of Microchip Outputs

No outputs are used

Runlinc Background

Runlinc is a web page inside a Wi-Fi chip. The programming is done 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 no outputs.

Part B: 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 CSS, HTML and JavaScript is 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 make sure the button is in the right spot for what we want:

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

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"/>	

Network Status: Active

CSS

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

HTML

JavaScript

JavaScript Loop

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 it will add an item 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>
```

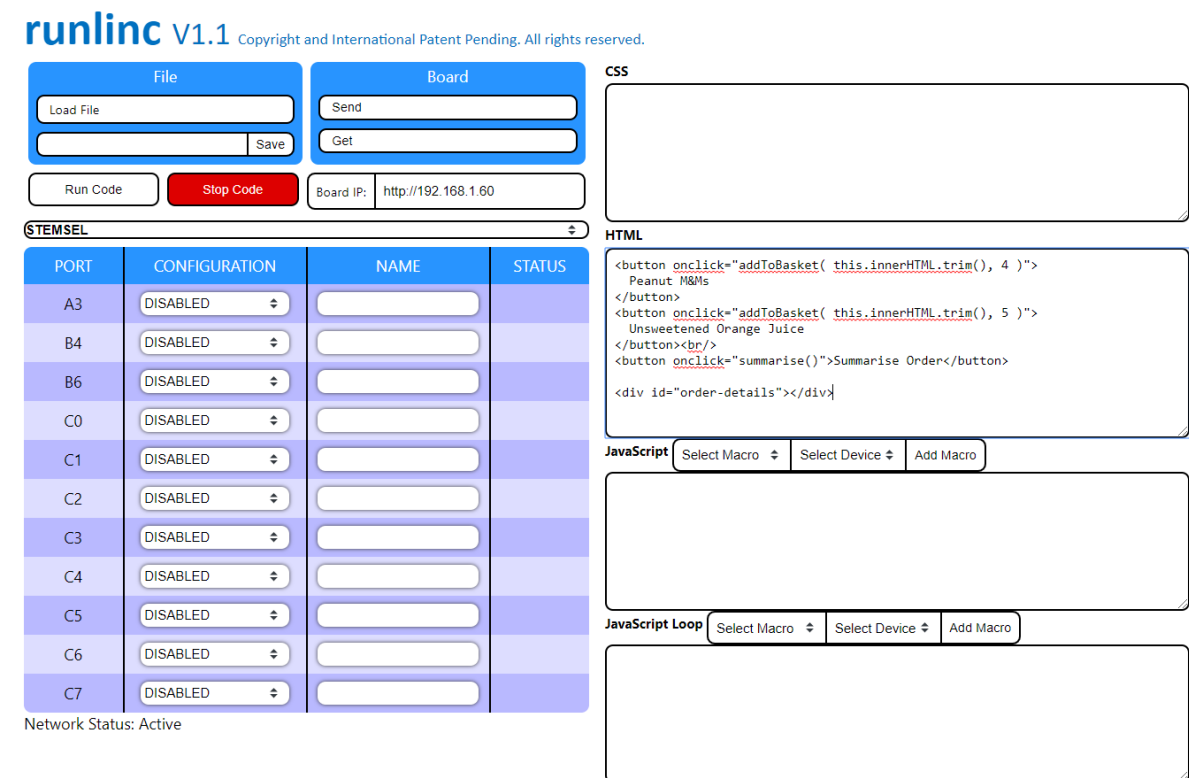


Figure 3 HTML Code block

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} items, `
            ).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)}`
    );
}

```

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		

```

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} items, `
      ).join('')
    )
  );
  speak( `The total price is ${total} dollars` );
}

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

Figure 4 JavaScript block

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} items, `
      ).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)}`
);
}
```

The screenshot displays the runlinc V1.1 web interface. At the top, there are two main sections: 'File' and 'Board'. The 'File' section includes a 'Load File' input field, a 'Save' button, and a 'Run Code' button. The 'Board' section includes a 'Send' input field, a 'Get' input field, and a 'Stop Code' button. Below these is a 'Board IP:' field with the value 'http://192.168.1.60'. A 'STEMSEL' dropdown menu is positioned above a table with the following data:

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		

Below the table, the network status is shown as 'Active'. To the right of the table are three code editors: 'CSS', 'HTML', and 'JavaScript'. The CSS editor contains a simple button style. The HTML editor contains buttons for adding items to a basket and summarizing the order. The JavaScript editor contains logic for speaking the contents of the basket and calculating the total price.

Figure 5 runlinc webpage image

Extensions

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

1. With the code done for 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 colour to the machine
4. Is there away 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 created, add up the prices for a total.