

# Assignment 3- Vending\_machine

## V1.0

Generated by Doxygen 1.8.17



<b>1 Bug List</b>	<b>1</b>
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 File Documentation</b>	<b>5</b>
3.1 CMakeLists.txt File Reference	5
3.1.1 Function Documentation	5
3.1.1.1 cmake_minimum_required()	5
3.2 main.c File Reference	5
3.2.1 Function Documentation	6
3.2.1.1 but1press_cbfunction()	6
3.2.1.2 but2press_cbfunction()	6
3.2.1.3 but3press_cbfunction()	7
3.2.1.4 but4press_cbfunction()	7
3.2.1.5 but5press_cbfunction()	8
3.2.1.6 but6press_cbfunction()	8
3.2.1.7 but7press_cbfunction()	9
3.2.1.8 but8press_cbfunction()	9
3.2.1.9 CONF_BUTT()	9
3.2.1.10 main()	10
3.2.2 Variable Documentation	10
3.2.2.1 Check_credit_Flag	10
3.2.2.2 Down_Flag	10
3.2.2.3 Flag_100_Cent	10
3.2.2.4 Flag_10_Cent	11
3.2.2.5 Flag_20_Cent	11
3.2.2.6 Flag_50_Cent	11
3.2.2.7 gpio0_dev	11
3.2.2.8 NS	11
3.2.2.9 product	11
3.2.2.10 Return_Flag	11
3.2.2.11 Up_Flag	11
3.3 vending_machine.h File Reference	12
3.3.1 Detailed Description	12
3.3.2 Function Documentation	12
3.3.2.1 but1press_cbfunction()	12
3.3.2.2 but2press_cbfunction()	13
3.3.2.3 but3press_cbfunction()	13
3.3.2.4 but4press_cbfunction()	14
3.3.2.5 but5press_cbfunction()	14
3.3.2.6 but6press_cbfunction()	15
3.3.2.7 but7press_cbfunction()	15

3.3.2.8 <code>but8press_cbfunction()</code> . . . . .	15
---	----

<b>Index</b>	<b>17</b>
--------------	-----------

## Chapter 1

# Bug List

File [vending\\_machine.h](#)

No known bugs.



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.c</a> . . . . .	5
<a href="#">vending_machine.h</a>	
The vending machine accepts a subset of coins and allows the user to browse available products, buy one product and return the credit . . . . .	12





## Chapter 3

# File Documentation

### 3.1 CMakeLists.txt File Reference

#### Functions

- [cmake\\_minimum\\_required](#) (VERSION 3.20.0) find\_package(Zephyr REQUIRED HINTS \$ENV

#### 3.1.1 Function Documentation

##### 3.1.1.1 cmake\_minimum\_required()

```
cmake_minimum_required (  
    VERSION 3.20.  0 )
```

### 3.2 main.c File Reference

#### Functions

- void [but1press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 1 is pressed, activate the 10 cents flag.*
- void [but2press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 2 is pressed, activate the 20 cents flag.*
- void [but3press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 3 is pressed, activate the 50 cents flag.*
- void [but4press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 4 is pressed, activate the 1 eur flag.*
- void [but5press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 5 is pressed, activate the Up flag.*
- void [but6press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 6 is pressed, activate the Down flag.*
- void [but7press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 7 is pressed, activate the check\_credit flag.*
- void [but8press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 8 is pressed, activate the return flag.*
- void [main](#) (void)  
*Brief decription of [main\(\)](#).*
- void [CONF\\_BUTTON](#) ()

## Variables

- volatile int `Flag_10_Cent` = 0
- volatile int `Flag_20_Cent` = 0
- volatile int `Flag_50_Cent` = 0
- volatile int `Flag_100_Cent` = 0
- volatile int `Up_Flag` = 0
- volatile int `Down_Flag` = 0
- volatile int `Check_credit_Flag` = 0
- volatile int `Return_Flag` = 0
- volatile int `NS` = 0
- volatile int `product` = 150
- const struct device \* `gpio0_dev`

## 3.2.1 Function Documentation

### 3.2.1.1 `but1press_cbfunction()`

```
void but1press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 1 is pressed, activate the 10 cents flag.

```
void but1press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Flag_10_Cent = 1;
}
```

#### Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

#### Returns

No returns

### 3.2.1.2 `but2press_cbfunction()`

```
void but2press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 2 is pressed, activate the 20 cents flag.

```
void but2press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Flag_20_Cent = 1;
}
```

## Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

## Returns

No returns

**3.2.1.3 but3press\_cbfunction()**

```
void but3press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 3 is pressed, activate the 50 cents flag.

```
void but3press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Flag_50_Cent = 1;
}
```

## Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

## Returns

No returns

**3.2.1.4 but4press\_cbfunction()**

```
void but4press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 4 is pressed, activate the 1 eur flag.

```
void but4press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Flag_100_Cent = 1;
}
```

## Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

**Returns**

No returns

**3.2.1.5 but5press\_cbfunction()**

```
void but5press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 5 is pressed, activate the Up flag.

```
*void but5press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Up_Flag = 1;
}
```

**Parameters**

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

**Returns**

No returns

**3.2.1.6 but6press\_cbfunction()**

```
void but6press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 6 is pressed, activate the Down flag.

```
*void but6press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Down_Flag = 1;
}
```

**Parameters**

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

**Returns**

No returns

### 3.2.1.7 but7press\_cbfunction()

```
void but7press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 7 is pressed, activate the check\_credit\_flag.

```
*void but7press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Check_credit_Flag = 1;
}
```

#### Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

#### Returns

No returns

### 3.2.1.8 but8press\_cbfunction()

```
void but8press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 8 is pressed, activate the return flag.

```
*void but8press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Return_Flag = 1;
}
```

#### Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

#### Returns

No returns

### 3.2.1.9 CONF\_BUTT()

```
void CONF_BUTT ( )
```

### 3.2.1.10 main()

```
void main (
    void )
```

Brief decription of [main\(\)](#).

Main has no input arguments. Our system have eight buttons: the button 1 is the 10 cents; the button 2 is the 20 cents; the button 3 is the 50 cents and the button 4 is the 1 eur; the button 5 is the up, the button 6 is the down; the button 7 is the selected product and the button 8 is the return credit.

If we insert a 10 cent coin, we are left with a 10 cent credit; if we add a 20 cent coin, we add 20 cent to the previous credit, that is, we are left with 30 cent and so on.

In the up(5)/down(6) buttons, we can select the desired product; where in the terminal we can see the previous, selected and next product.

If credit is available, it is possible to select and take the desired product. The available credit will be the previously available credit minus the product price.

We can insert the coins and if we don't want to select any product, we can do the return to get the money inserted. When we select a product and we still have credit available, we can return the change.

#### Returns

[main\(\)](#) always returns 0

## 3.2.2 Variable Documentation

### 3.2.2.1 Check\_credit\_Flag

```
volatile int Check_credit_Flag = 0
```

### 3.2.2.2 Down\_Flag

```
volatile int Down_Flag = 0
```

### 3.2.2.3 Flag\_100\_Cent

```
volatile int Flag_100_Cent = 0
```

#### 3.2.2.4 Flag\_10\_Cent

```
volatile int Flag_10_Cent = 0
```

#### 3.2.2.5 Flag\_20\_Cent

```
volatile int Flag_20_Cent = 0
```

#### 3.2.2.6 Flag\_50\_Cent

```
volatile int Flag_50_Cent = 0
```

#### 3.2.2.7 gpio0\_dev

```
const struct device* gpio0_dev
```

#### 3.2.2.8 NS

```
volatile int NS = 0
```

#### 3.2.2.9 product

```
volatile int product = 150
```

#### 3.2.2.10 Return\_Flag

```
volatile int Return_Flag = 0
```

#### 3.2.2.11 Up\_Flag

```
volatile int Up_Flag = 0
```

### 3.3 vending\_machine.h File Reference

The vending machine accepts a subset of coins and allows the user to browse available products, buy one product and return the credit.

#### Functions

- void [but1press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 1 is pressed, activate the 10 cents flag.*
- void [but2press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 2 is pressed, activate the 20 cents flag.*
- void [but3press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 3 is pressed, activate the 50 cents flag.*
- void [but4press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 4 is pressed, activate the 1 eur flag.*
- void [but5press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 5 is pressed, activate the Up flag.*
- void [but6press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 6 is pressed, activate the Down flag.*
- void [but7press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 7 is pressed, activate the check\_credit flag.*
- void [but8press\\_cbfunction](#) (const struct device \*dev, struct gpio\_callback \*cb, uint32\_t pins)  
*If button 8 is pressed, activate the return flag.*

#### 3.3.1 Detailed Description

The vending machine accepts a subset of coins and allows the user to browse available products, buy one product and return the credit.

Contains the functions needed to create a vending machine.

##### Author

Frederico Moreira, Ana Sousa, Pedro Rodrigues

##### Date

17 May 2022

**Bug** No known bugs.

#### 3.3.2 Function Documentation

##### 3.3.2.1 but1press\_cbfunction()

```
void but1press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 1 is pressed, activate the 10 cents flag.

```
void but1press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Flag_10_Cent = 1;
}
```



## Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

## Returns

No returns

**3.3.2.2 but2press\_cbfunction()**

```
void but2press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 2 is pressed, activate the 20 cents flag.

```
void but2press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Flag_20_Cent = 1;
}
```

## Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

## Returns

No returns

**3.3.2.3 but3press\_cbfunction()**

```
void but3press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 3 is pressed, activate the 50 cents flag.

```
void but3press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Flag_50_Cent = 1;
}
```

## Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

**Returns**

No returns

**3.3.2.4 but4press\_cbfunction()**

```
void but4press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 4 is pressed, activate the 1 eur flag.

```
void but4press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Flag_100_Cent = 1;
}
```

**Parameters**

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

**Returns**

No returns

**3.3.2.5 but5press\_cbfunction()**

```
void but5press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 5 is pressed, activate the Up flag.

```
*void but5press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Up_Flag = 1;
}
```

**Parameters**

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

**Returns**

No returns

### 3.3.2.6 but6press\_cbfunction()

```
void but6press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 6 is pressed, activate the Down flag.

```
*void but6press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Down_Flag = 1;
}
```

#### Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

#### Returns

No returns

### 3.3.2.7 but7press\_cbfunction()

```
void but7press_cbfunction (
    const struct device * dev,
    struct gpio_callback * cb,
    uint32_t pins )
```

If button 7 is pressed, activate the check\_credit\_flag.

```
*void but7press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    Check_credit_Flag = 1;
}
```

#### Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

#### Returns

No returns

### 3.3.2.8 but8press\_cbfunction()

```
void but8press_cbfunction (
    const struct device * dev,
```

```
    struct gpio_callback * cb,  
    uint32_t pins )
```

If button 8 is pressed, activate the return flag.

```
*void but8press_cbfunction(const struct device *dev, struct gpio_callback *cb, uint32_t pins)  
{  
    Return_Flag = 1;  
}
```

#### Parameters

<i>arg3</i>	const struct device *dev, struct gpio_callback *cb, uint32_t pins.
-------------	--

#### Returns

No returns

# Index

- but1press\_cbfunction
  - main.c, [6](#)
  - vending\_machine.h, [12](#)
- but2press\_cbfunction
  - main.c, [6](#)
  - vending\_machine.h, [13](#)
- but3press\_cbfunction
  - main.c, [7](#)
  - vending\_machine.h, [13](#)
- but4press\_cbfunction
  - main.c, [7](#)
  - vending\_machine.h, [14](#)
- but5press\_cbfunction
  - main.c, [8](#)
  - vending\_machine.h, [14](#)
- but6press\_cbfunction
  - main.c, [8](#)
  - vending\_machine.h, [14](#)
- but7press\_cbfunction
  - main.c, [8](#)
  - vending\_machine.h, [15](#)
- but8press\_cbfunction
  - main.c, [9](#)
  - vending\_machine.h, [15](#)
- Check\_credit\_Flag
  - main.c, [10](#)
- cmake\_minimum\_required
  - CMakeLists.txt, [5](#)
- CMakeLists.txt, [5](#)
  - cmake\_minimum\_required, [5](#)
- CONF\_BUTT
  - main.c, [9](#)
- Down\_Flag
  - main.c, [10](#)
- Flag\_100\_Cent
  - main.c, [10](#)
- Flag\_10\_Cent
  - main.c, [10](#)
- Flag\_20\_Cent
  - main.c, [11](#)
- Flag\_50\_Cent
  - main.c, [11](#)
- gpio0\_dev
  - main.c, [11](#)
- main
  - main.c, [9](#)

- main.c, [5](#)
  - but1press\_cbfunction, [6](#)
  - but2press\_cbfunction, [6](#)
  - but3press\_cbfunction, [7](#)
  - but4press\_cbfunction, [7](#)
  - but5press\_cbfunction, [8](#)
  - but6press\_cbfunction, [8](#)
  - but7press\_cbfunction, [8](#)
  - but8press\_cbfunction, [9](#)
  - Check\_credit\_Flag, [10](#)
  - CONF\_BUTT, [9](#)
  - Down\_Flag, [10](#)
  - Flag\_100\_Cent, [10](#)
  - Flag\_10\_Cent, [10](#)
  - Flag\_20\_Cent, [11](#)
  - Flag\_50\_Cent, [11](#)
  - gpio0\_dev, [11](#)
  - main, [9](#)
  - NS, [11](#)
  - product, [11](#)
  - Return\_Flag, [11](#)
  - Up\_Flag, [11](#)
- NS
  - main.c, [11](#)
- product
  - main.c, [11](#)
- Return\_Flag
  - main.c, [11](#)
- Up\_Flag
  - main.c, [11](#)
- vending\_machine.h, [12](#)
  - but1press\_cbfunction, [12](#)
  - but2press\_cbfunction, [13](#)
  - but3press\_cbfunction, [13](#)
  - but4press\_cbfunction, [14](#)
  - but5press\_cbfunction, [14](#)
  - but6press\_cbfunction, [14](#)
  - but7press\_cbfunction, [15](#)
  - but8press\_cbfunction, [15](#)