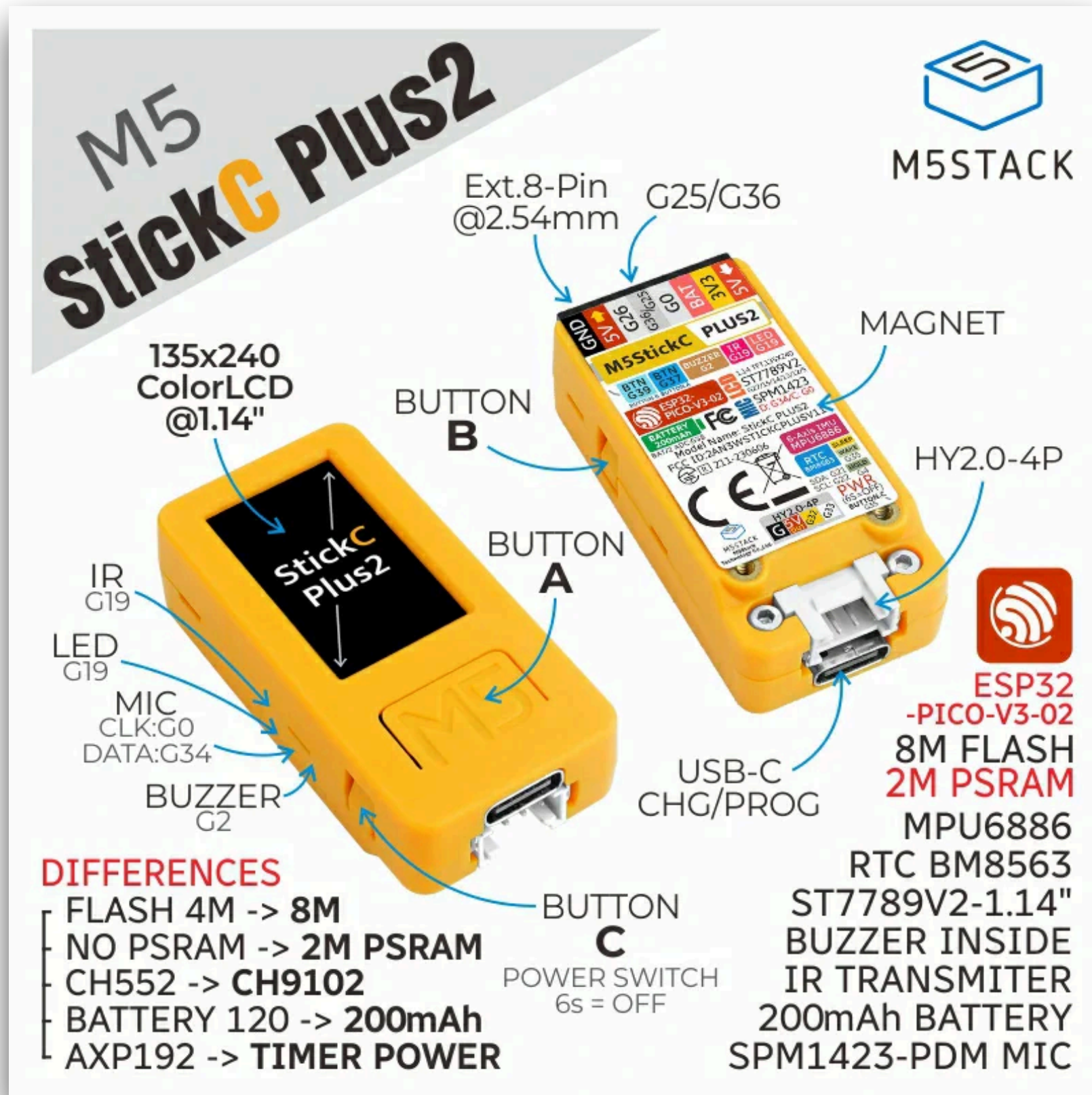


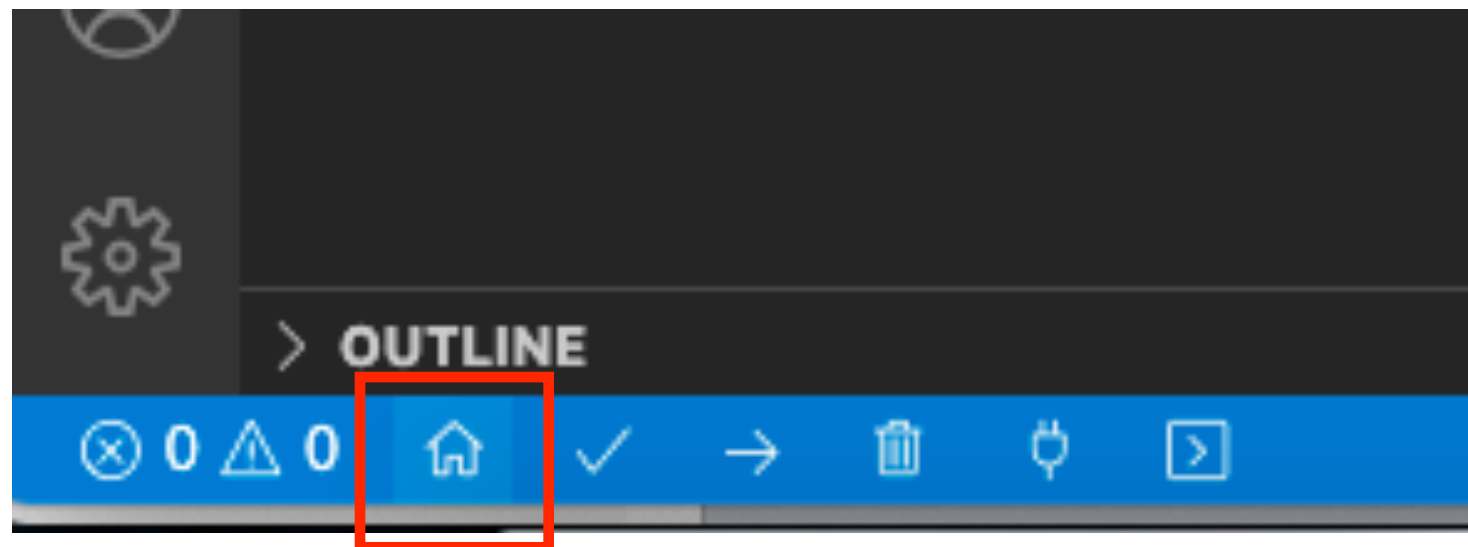
# M5StickC Plus2



# M5StickC Gebruiken

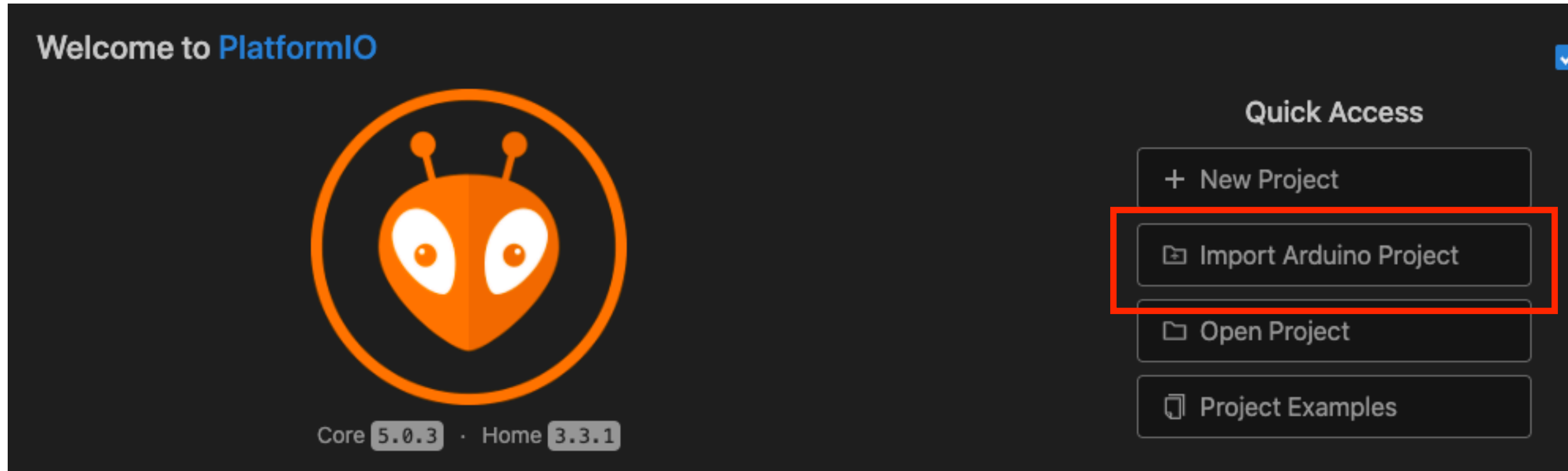
1. Download m5-project.zip van Canvas en pak de zip uit
2. (Als je Linux gebruikt) voer in de terminal uit:  
`sudo apt install python3-venv`  
en daarna `sudo apt-get install python3-setuptools`
3. Als je VirtualBox draait): Controleer dat VirtualBox toegang heeft tot je USB poort: <https://www.debugpoint.com/enable-usb-virtualbox/>
4. Ga naar <https://platformio.org/install/ide?install=vscode> en volg de installatie-instructies  
(zie <https://itsfoss.com/install-deb-files-ubuntu> voor instructies over hoe je op Ubuntu de VSCode .deb file installeert)

5. Klik op Home:



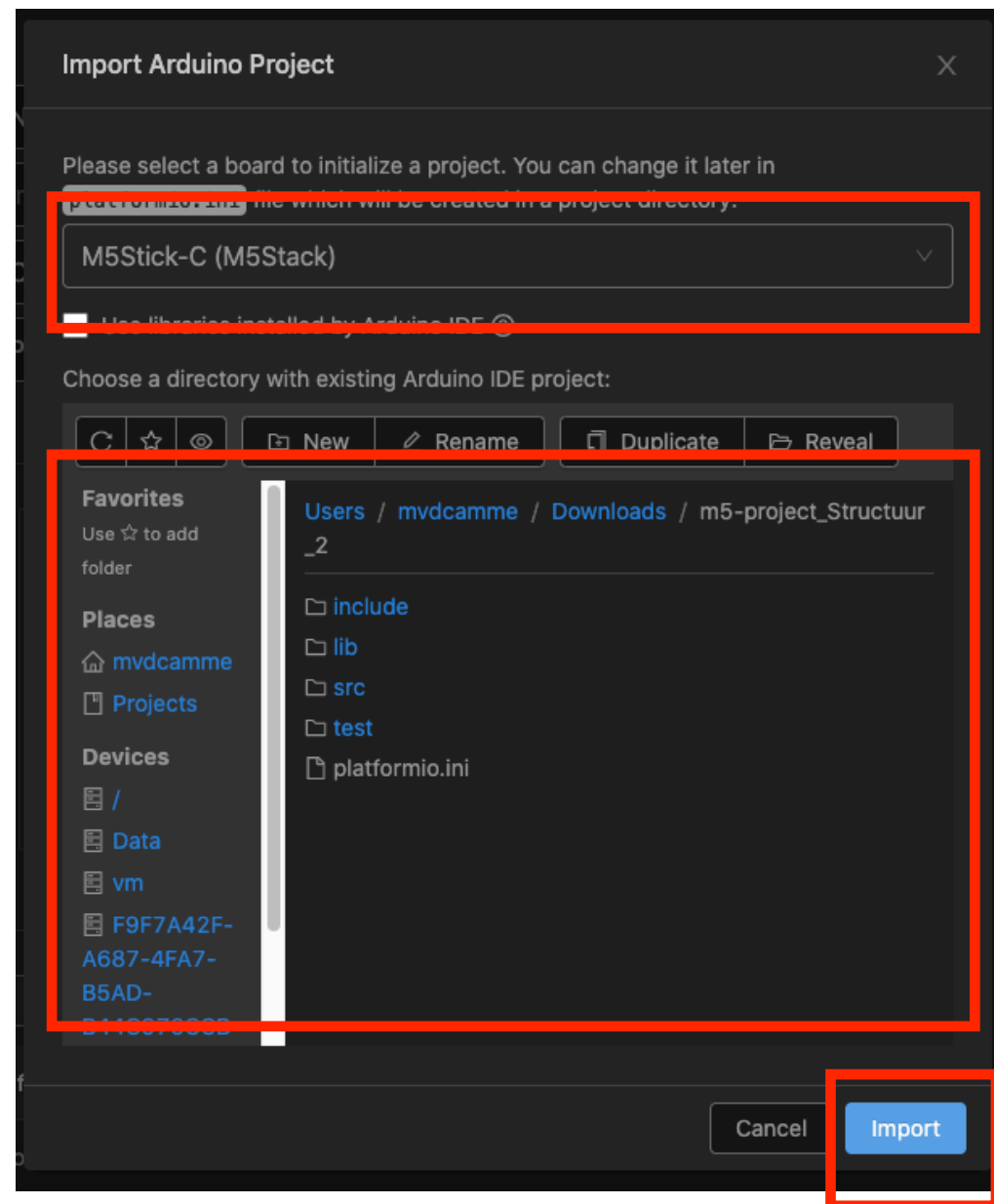
# M5StickC Gebruiken

## 5. Klik op Import Arduino Project



# M5StickC Gebruiken

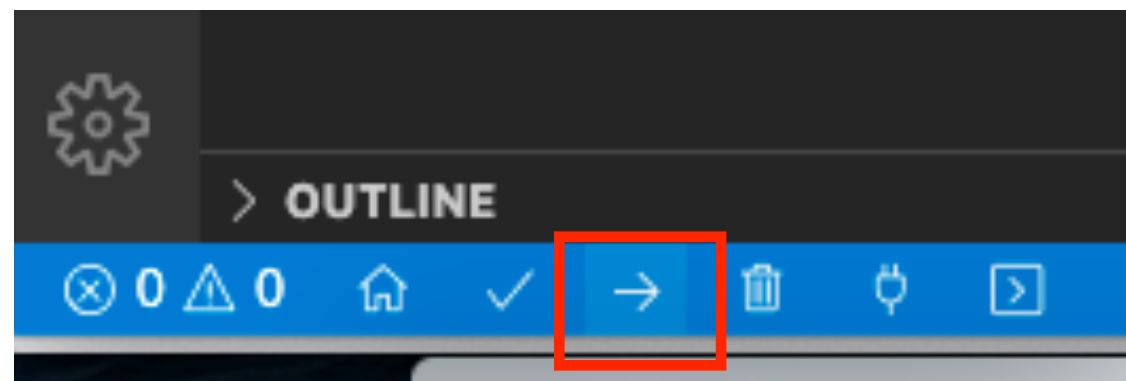
6. Importeer project:



Typ en selecteer "M5Stick-C ("M5Stack")"

Open het project

7. Voer code uit (controller ingeplugd en opgeladen):



# Mogelijke Foutberichten

1. (Op Linux): Als tijdens de installatie Python3 niet kan gevonden worden:  
voer `sudo apt-get install curl python3-virtualenv python3-venv` uit  
Volg daarna de Super-Quick stap voor Linux op:  
<https://docs.platformio.org/en/latest/core/installation/methods/installer-script.html#super-quick-macos-linux>
2. (Als je VirtualBox draait): Controleer dat VirtualBox toegang heeft tot je USB poort: <https://www.debugpoint.com/enable-usb-virtualbox/>
3. Als je de waarschuwing "Please install `99-platformio-udev.rules`" krijgt:  
volg de link <https://docs.platformio.org/en/latest/core/installation/udev-rules.html>  
en voer de stappen voor de Terminal uit. Herstart daarna je computer
4. Als je de error `Could not open /dev/ttyUSB0, the port does not exist`, of de fout `PermissionError: [Errno 13] Permission denied: '/dev/ttyUSB0'` krijgt, voer dan `sudo usermod -a -G dialout <jouw_username>` uit.  
Herstart daarna je computer
5. Als je een `OSError: [Errno 5] Input/Output Error` krijgt:  
plug de controller uit en terug in

# M5StickC Code

```
#include <M5Unified.h>
```

```
// Code die eenmalig wordt uitgevoerd
```

```
void setup() {  
    M5.begin();  
    M5.Imu.init();  
    M5.Lcd.fillScreen(M5.Lcd.color565(0, 0, 0));  
}
```

```
// Code die voortdurend wordt uitgevoerd
```

```
void loop() {  
    M5.update();  
    M5.Lcd.printf("Hello World\n");  
}
```

# M5StickC Code

```
#include <M5Unified.h>
```

```
// Code die eenmalig wordt uitgevoerd
```

```
void setup() {
```

```
    M5.begin();
```

```
    M5.Imu.init();
```

```
    M5.Lcd.fillScreen(M5.Lcd.color565(0, 0, 0));
```

```
}
```

```
// Code die voortdurend wordt uitgevoerd
```

```
void loop() {
```

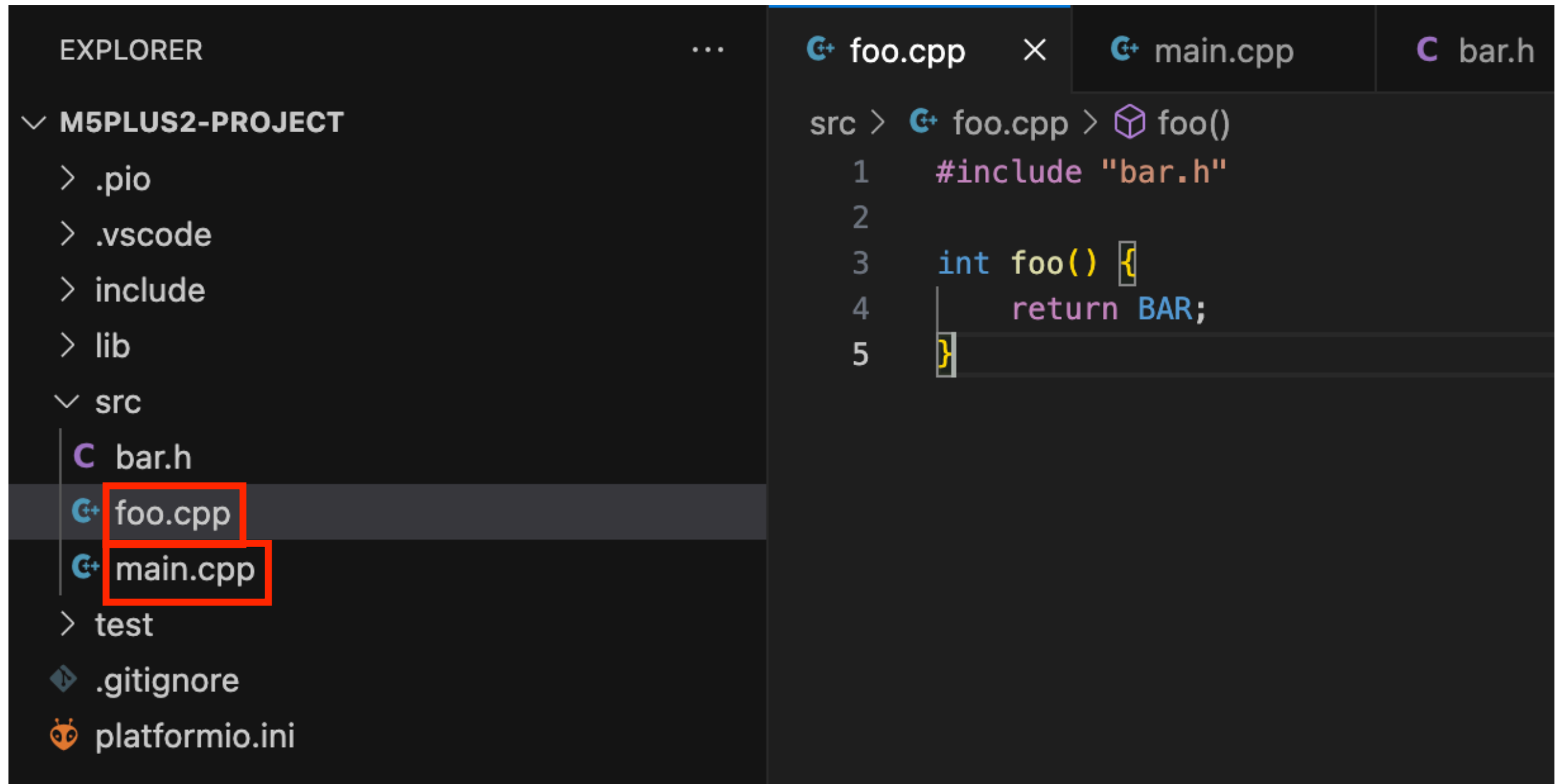
```
    M5.update();
```

```
    M5.Lcd.printf("Hello World\n");
```

```
}
```

**C++**: kan **methoden** gebruiken

# M5StickC Code



**C++:** VSCode verwacht dat source code bestanden eindigen op “.cpp” i.p.v. “.c”



# Online Documentatie

**NIET** toegelaten om code te kopiëren

<https://docs.m5stack.com/en/arduino/m5unified/helloworld#m5unified%20api>

<https://github.com/m5stack/m5-docs/tree/master/docs/en/api>  
(Wat verouderd)

<https://github.com/m5stack/m5-docs/blob/master/docs/en/api/lcd.md>  
(Displayfuncties)

<https://github.com/m5stack/M5StickC/tree/master/examples>

<https://www.arduino.cc/en/Reference/EEPROM> ("file IO" equivalent)

# Enkele Handige Functies

Voor volledige beschrijving van alle functies:  
zie online documentatie

## Displayfuncties:

```
M5.Lcd.drawPixel(x, y, color); // Teken 1 pixel
M5.Lcd.fillRect(x, y, width, height, color); // Teken rechthoek
M5.Lcd.fillScreen(color); // Kleur volledige scherm
uint16_t color = M5.Lcd.color565(red, green, blue); //Maak nieuwe kleur
(of gebruikt bestaande BLUE, RED, GREEN, BLACK etc. macro's)
M5.Lcd.setRotation(1); // Verander rotatie van display
                        // (1 = 90 graden draaien)
```

Tekst printen:

[illegible]

# Enkele Handige Functies

Initialisatie:

```
M5.begin(); // Initialiseer controller  
M5.Imu.init(); // Initialiseer accelerometer  
EEPROM.begin(memory_size); // Initialiseer memory_size aan bytes  
                             // in het persistente flashgeheugen
```

Systeem:

```
millis(); // Aantal milliseconden verstreken sinds opstarten controller  
rand();   // Bereken (pseudo-)random getal  
M5.update(); // Update sensoren -> uitvoeren in elke iteratie van loop()  
delay(milliseconden); // Pauzeer voor zoveel milliseconden  
M5.Lcd.height(); // Hoogte (in pixels) van display  
M5.Lcd.width(); // Breedte (in pixels) van display
```

# Enkele Handige Functies

Knop ingedrukt?

<code>M5.BtnA.isPressed()</code>	}	Is de knop op dit moment (nog altijd) ingedrukt?
<code>M5.BtnB.isPressed()</code>		
<code>M5.BtnA.wasPressed()</code>	}	Is de knop ingedrukt geweest sinds de laatste keer dat deze methode werd opgeroepen?
<code>M5.BtnB.wasPressed()</code>		

Als je denkt dat `isPressed` of `wasPressed` foutieve waarden teruggeeft, voer dan eerst `M5.update()` uit

# Enkele Handige Functies

Waarde persistent opslaan in flashgeheugen:

```
EEPROM.writeByte(address, byte); // Schrijf (later) byte weg naar adres  
EEPROM.writeInt(address, integer); // Schrijf (later) int weg naar adres  
EEPROM.commit(); // Voer schrijfoperaties effectief uit
```

Waarde inladen uit flashgeheugen:

```
EEPROM.readByte(address); // Lees byte op adres in flash uit  
EEPROM.readInt(address); // Lees int op adres in flash uit
```

Accelerometer:

```
float acc_x = 0, acc_y = 0, acc_z = 0;  
M5.Imu.getAccelData(&acc_x, &acc_y, &acc_z);  
// Lees acceleratie in X-, Y-, en Z-richtingen uit  
// (Genormaliseerd naar -1.0 tot 1.0)
```

# Printen naar Serial Monitor

Voeg

`monitor_speed = 115200`

toe aan `platformio.ini`

```
M5StickC_Default > platformio.ini
1  ; PlatformIO Project Configuration File
2  ;
3  ; Build options: build flags, source filter
4  ; Upload options: custom upload port, speed and extra flags
5  ; Library options: dependencies, extra library storages
6  ; Advanced options: extra scripting
7  ;
8  ; Please visit documentation for the other options and examples
9  ; https://docs.platformio.org/page/projectconf.html
10
11 [env:m5stick-c]
12 platform = espressif32
13 board = m5stick-c
14 framework = arduino
15 monitor_speed = 115200
16 |
```

```
void setup() {
    ...
    Serial.begin(115200);
    Serial.flush();
}
Serial.printf("%i\n", 42);
```

Open  
Serial Monitor

