

list2.txt

入力:C言語でプログラムしてください。Pico-SDKを使ってください。

GPT: 了解しました。以下はラズベリーパイ Pico 用の C 言語によるPico-SDKを使用したプログラムの例です。

#### 1. Pico-SDKのセットアップ:

まず、Pico-SDKのセットアップが完了していることを確認してください。セットアップがまだの場合、公式の Getting started with Raspberry Pi Pico ガイド <https://datasheets.raspberrypi.com/pico/getting-started-with-pico.pdf> に従ってセットアップしてください。

#### 2. プログラム:

'CMakeLists.txt':

```
cmake_minimum_required(VERSION 3.12)
```

```
include(pico_sdk_import.cmake)
```

```
project(motor_relay_project)
```

```
pico_sdk_init()
```

```
add_executable(motor_relay motor_relay.c)
```

```
pico_enable_stdio_usb(motor_relay 1)
```

```
pico_enable_stdio_uart(motor_relay 1)
```

```
pico_add_extra_outputs(motor_relay)
```

```
target_link_libraries(motor_relay pico_stdlib)
```

'motor\_relay.c':

```
#include "pico/stdlib.h"
```

```
const uint BUTTON_PIN = 15;
```

```
const uint RELAY_PIN = 14;
```

```
int main() {  
    stdio_init_all();
```

```
    gpio_init(BUTTON_PIN);  
    gpio_set_dir(BUTTON_PIN, GPIO_IN);  
    gpio_pull_up(BUTTON_PIN);
```

```
    gpio_init(RELAY_PIN);  
    gpio_set_dir(RELAY_PIN, GPIO_OUT);
```

```
    while (true) {  
        if (gpio_get(BUTTON_PIN) == 0) { // Button is pressed  
            gpio_put(RELAY_PIN, 1); // Turn on relay  
        } else {  
            gpio_put(RELAY_PIN, 0); // Turn off relay  
        }  
        sleep_ms(100); // Debouncing and reduce CPU usage
```

list2.txt

```
}  
    return 0;  
}
```

このプログラムは、ボタンが押されたときにリレーをONにし、ボタンが放されたときにリレーをOFFにする動作を行います。

### 3. ビルドとアップロード:

ターミナルで以下のコマンドを使用して、コードをビルドし、ラズベリーパイ Pico にアップロードします。

```
mkdir build  
cd build  
cmake ..  
make
```

生成された '.uf2' ファイルをラズベリーパイ Pico にドラッグ&ドロップすることで、コードをアップロードできます。

注意: Picoにプログラムをアップロードする前に、必要な接続と電源供給を確認してください。