

IoT Device

Example 2.01

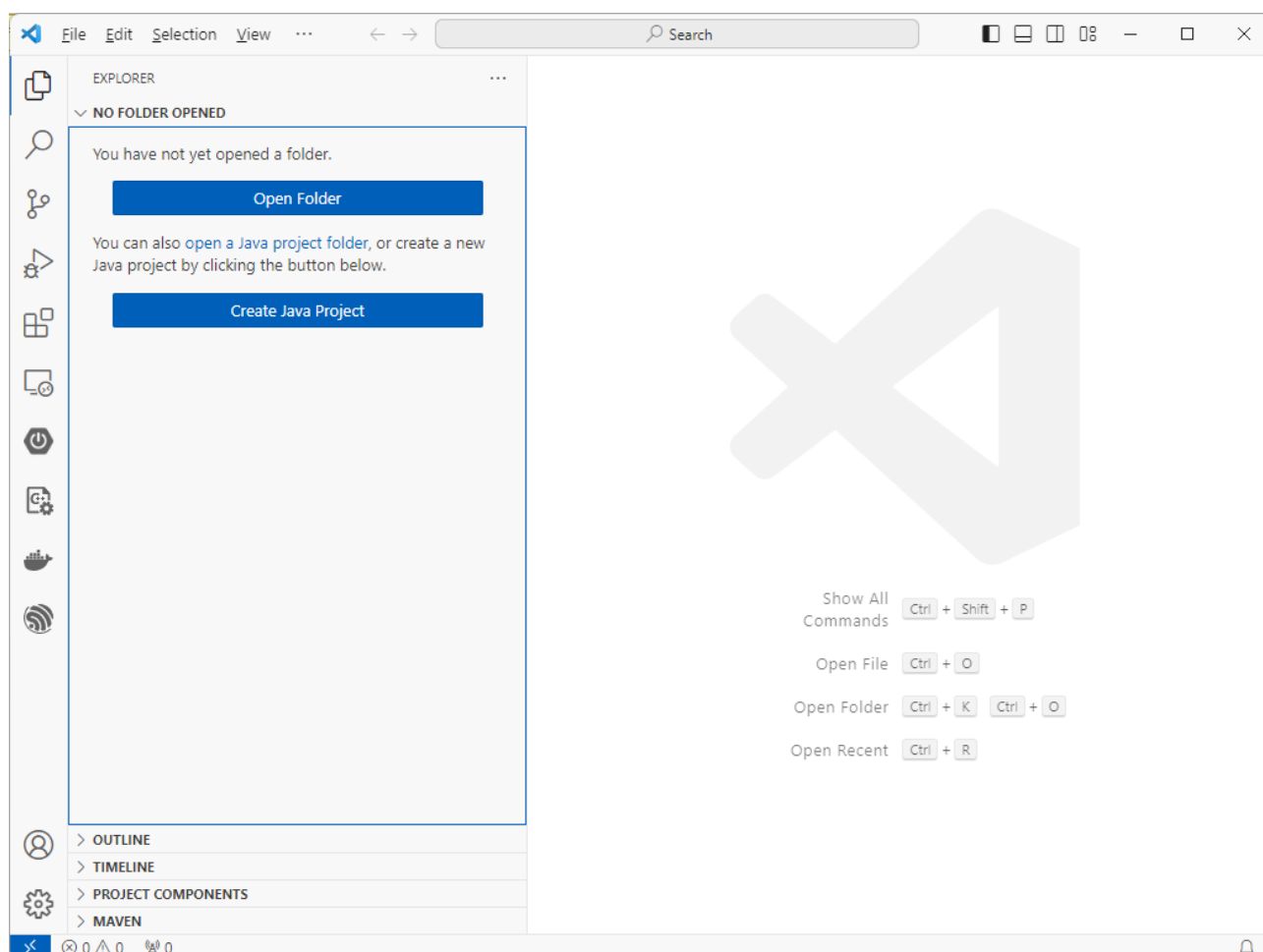
Mục tiêu: Thiết lập môi trường lập trình ESP-IDF (Espressif IoT Development Framework)

Yêu cầu:

- ✓ Cài đặt Visual Studio Code
- ✓ Cài đặt và cấu hình Espressif IDF Extensions


Hướng dẫn:

Bước 1: Download tại <https://code.visualstudio.com>, sau đó tiến hành cài đặt Visual Studio Code



Bước 2: Cài đặt và cấu hình Espressif IDF

- ✓ Để cài đặt Extension Espressif IDF cho Visual Studio Code bạn hãy thực hiện Mở Extensions (Ctrl+Shift+X), tìm kiếm **Espressif IDF**




Espressif IDF

v1.6.5

Espressif Systems espressif.com | 515,985 | ★★★★★ (97)


Develop and debug applications for Espressif ESP32, ESP32-S2 chips with ESP-IDF


[Disable](#)
[Uninstall](#)


This extension is enabled globally.

- ✓ Để cấu hình Extension Espressif IDF cho Visual Studio Code bạn hãy thực hiện Mở Command Palette (Ctrl+Shift+P), chọn **ESP-IDF: Configure ESP-IDF extension**

> ESP-IDF: Configure ESP-IDF extension


ESP-IDF Explorer: Focus on ESP-IDF Docs search results View similar commands 

ESP-IDF: Configure ESP-IDF extension 

ESP-IDF Explorer: Focus on IDF App Tracer View

ESP-IDF: Add Arduino ESP32 as ESP-IDF Component

ESP-IDF: Clear ESP-IDF Search results



ESPRESSIF


ESP-IDF Extension for Visual Studio Code

Welcome.

are installed before choosing the setup mode.

Choose a setup mode.

Select where to save these settings:

Global 

EXPRESS


Fastest option. Choose ESP-IDF, ESP-IDF Tools directory and python executable to create ESP-IDF.

ADVANCED

Configurable option. Choose ESP-IDF, ESP-IDF Tools directory and python executable to create ESP-IDF.
Can choose ESP-IDF Tools download or manually input each existing ESP-IDF tool path.

USE EXISTING SETUP

Select existing ESP-IDF setup saved in the extension or find ESP-IDF in your system.

 **ESPRESSIF**

ESP-IDF Extension for Visual Studio Code

Select download server:

Github ▼

☐ Show all ESP-IDF tags


Select ESP-IDF version:

v5.1.2 (release version) ▼

Enter ESP-IDF container directory


C:\Users\NAT\esp

\esp-idf



Enter ESP-IDF Tools directory (IDF_TOOLS_PATH)

C:\Users\NAT\espressif



Install

Example 2.02

Mục tiêu: Tạo và quản lý ứng dụng chạy trên vi điều khiển **ESP32**:

Yêu cầu: Ứng dụng thực hiện các chức năng sau:

- ✓ In ra Terminal nội dung **Hello World**

Hướng dẫn:

Bước 1: Sử dụng Visual Studio Code, mở Command Palette (Ctrl+Shift+P) và nhập **IDF: New Project**) để bắt đầu tạo project với các thông tin như sau:

- ✓ Project Name: **example202**
- ✓ Enter Project directory: **D:\java-projects**
- ✓ Choose ESP-IDF Board: **ESP32-C3 chip(via ESP USB Bridge)**
- ✓ Choose serial port: **COMx**
- ✓ Choose Template: **template-app**

New Project

Project Name
example201

Enter Project directory
D:\IoT-Projects \example201

Choose ESP-IDF Board
ESP32-C3 chip (via ESP USB Bridge)

Choose serial port
COM1

Add your ESP-IDF Component directory

Choose Template

New Project

ESP-IDF

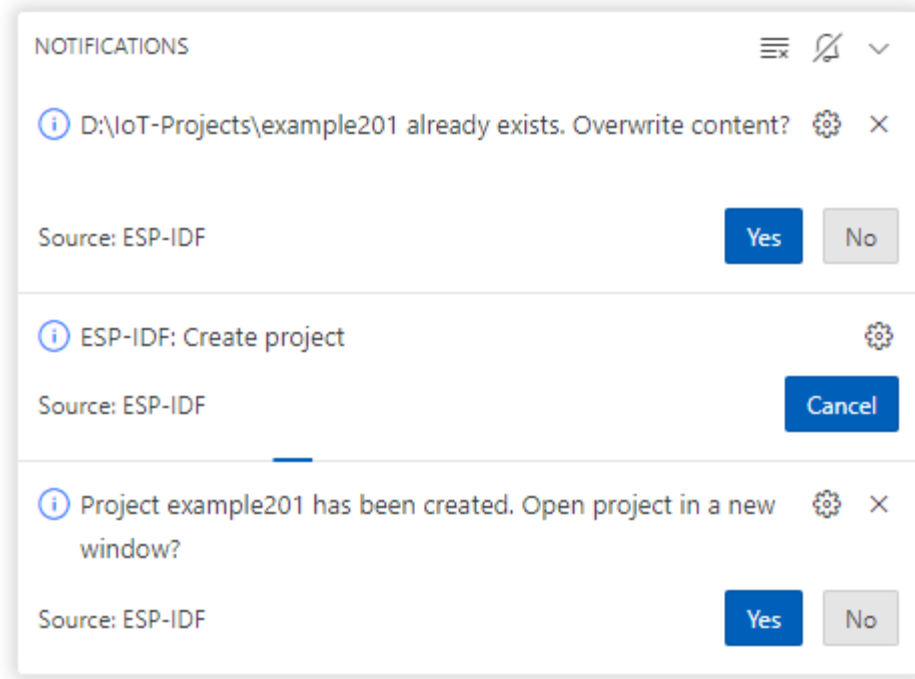
Create project using template hello_world

Supported Targets ESP32 ESP32-C2 ESP32-C3 ESP32-C6 ESP32-H2 ESP32-S2 ESP32-S3

Search Template By Name

get-started
blink
hello_world
sample_project
bluetooth

Hello World Example
Starts a FreeRTOS task to print "Hello World".
(See the README.md file in the upper level 'examples' directory for more information about examples.)



Bước 2: Sửa file **hello_world_main.c** như sau:

hello_world_main.c

```
#include <stdio.h>
#include <inttypes.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_chip_info.h"
#include "esp_flash.h"

void app_main(void)
```

```
{  
    printf("Hello world!\n");  
    printf("Restarting now.\n");  
    fflush(stdout);  
    esp_restart();  
}
```

Bước 3: Build project và nạp chương trình vào vi điều khiển



Example 2.03

Mục tiêu: Tạo và quản lý ứng dụng sử dụng FreeRTOS chạy trên vi điều khiển **ESP32**:

Yêu cầu: Ứng dụng thực hiện các chức năng sau:

- ✓ In ra Terminal nội dung Hello World

Hướng dẫn:

Bước 1: Sử dụng Visual Studio Code, mở Command Palette (Ctrl+Shift+P) và nhập **IDF: New Project**) để bắt đầu tạo project với các thông tin như sau:

- ✓ Project Name: **example203**
- ✓ Enter Project directory: **D:\java-projects**
- ✓ Choose ESP-IDF Board: **ESP32-C3 chip(via ESP USB Bridge)**
- ✓ Choose serial port: COMx
- ✓ Choose Template: **template-app**

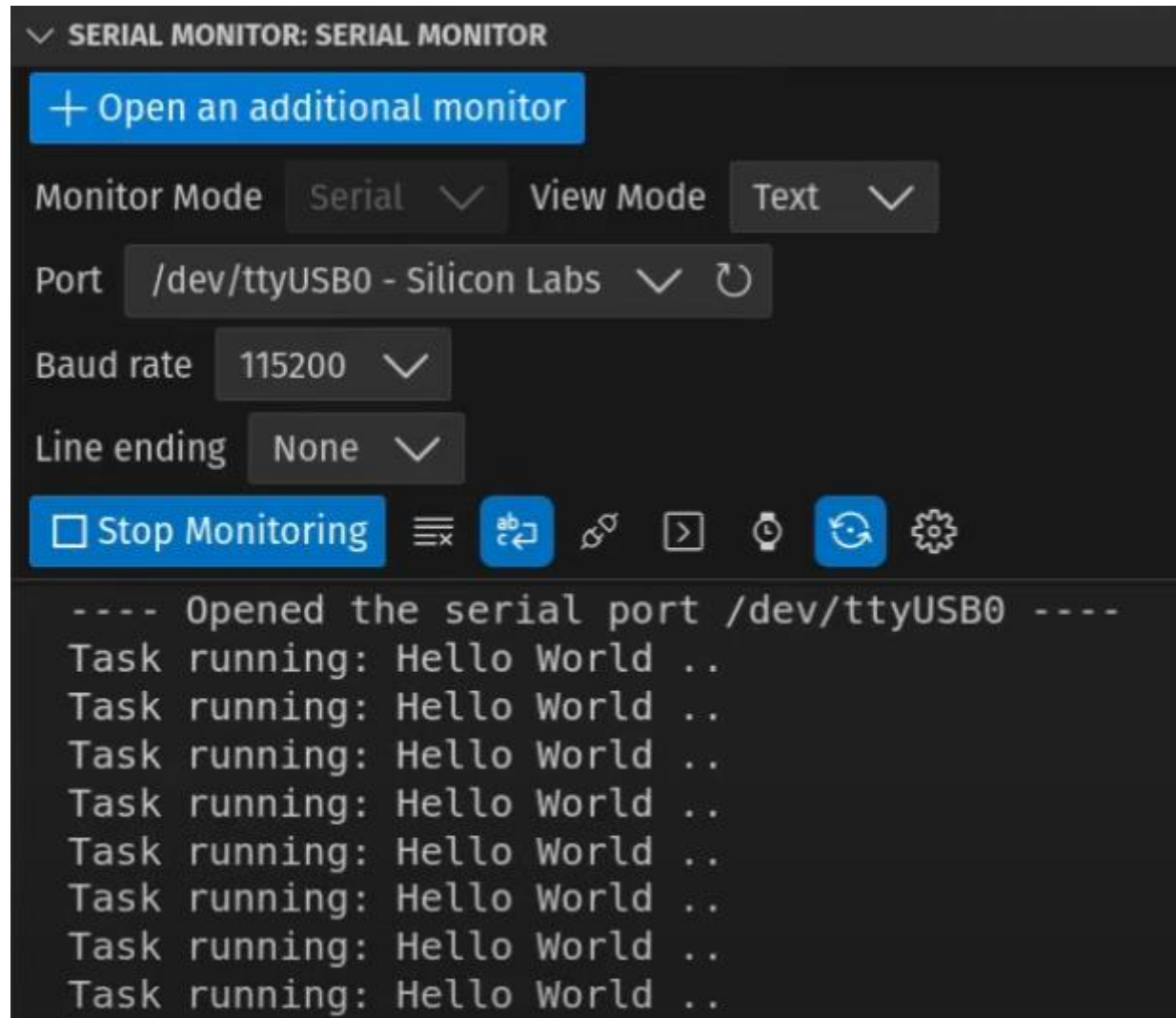
Bước 2: Sửa file `hello_world_main.c` như sau:

hello_world_main.c

```
#include <stdio.h>
#include <inttypes.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_chip_info.h"
#include "esp_flash.h"
TaskHandle_t HelloWorldTaskHandle = NULL;
void HelloWorld_Task(void *arg)
{
    while (1)
    {
        printf("Task running: Hello World ..\n");
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}
void app_main(void)
{
    xTaskCreate(HelloWorld_Task, "HelloWorld", 4096, NULL, 10, &HelloWorldTaskHandle);
    //xTaskCreatePinnedToCore(HelloWorld_Task, "HelloWorld", 4096, NULL, 10, &HelloWorldTaskHandle, 1); // Run on Core 1
}
```

Bước 3: Build project và nạp chương trình vào vi điều khiển





Example 2.04

Mục tiêu: Tạo và quản lý ứng dụng sử dụng FreeRTOS chạy trên vi điều khiển **ESP32**:

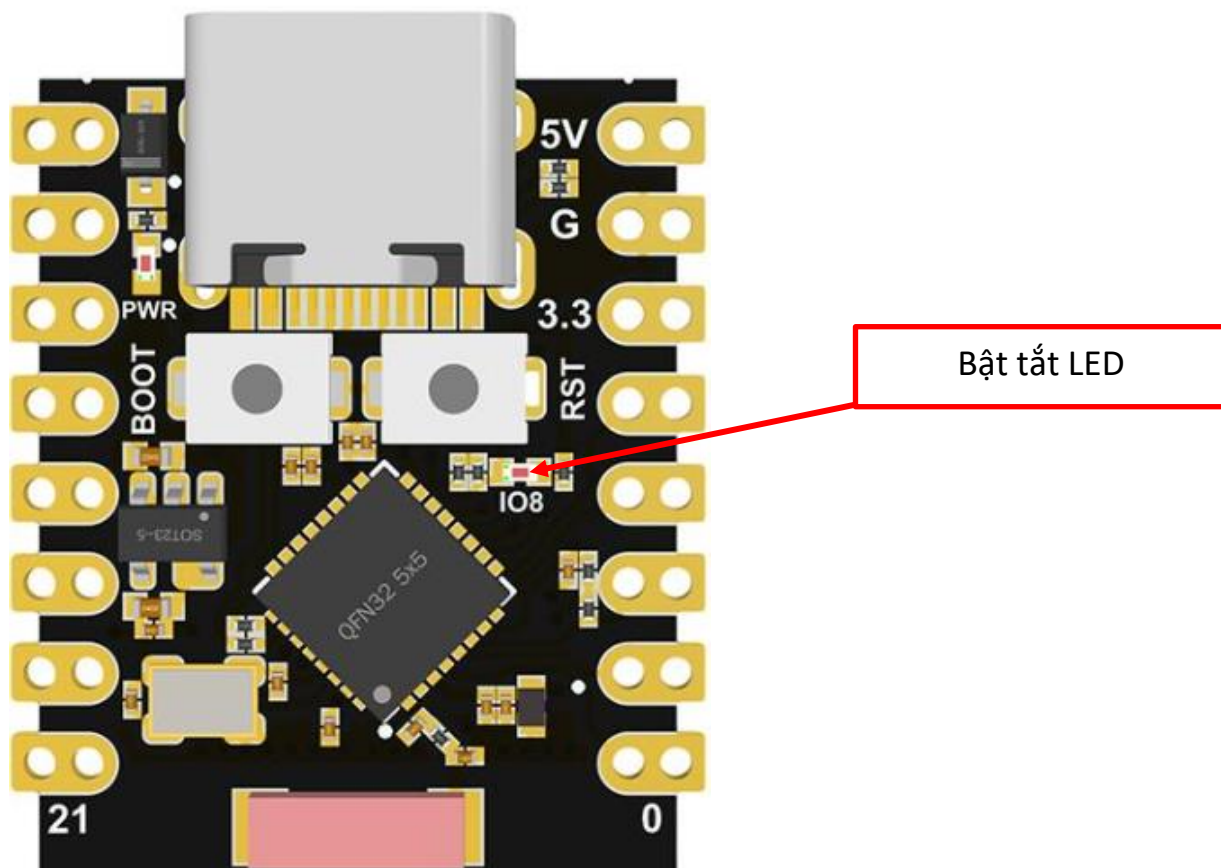
Yêu cầu: Ứng dụng thực hiện các chức năng sau:

- ✓ Thực hiện bật/tắt LED với chu kỳ 1000ms

Hướng dẫn:

Bước 1: Sử dụng Visual Studio Code, mở Command Palette (Ctrl+Shift+P) và nhập **IDF: New Project**) để bắt đầu tạo project với các thông tin như sau:

- ✓ Project Name: **example204**
- ✓ Enter Project directory: **D:\java-projects**
- ✓ Choose ESP-IDF Board: **ESP32-C3 chip(via ESP USB Bridge)**
- ✓ Choose serial port: COMx
- ✓ Choose Template: **template-app**



Bước 2: Sửa file `blink_main.c` như sau:

blink_main.c

```
#include <stdio.h>
#include <inttypes.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_chip_info.h"
#include "esp_flash.h"

#include "driver/gpio.h"

#define BLINK_GPIO GPIO_NUM_32

TaskHandle_t BlinkyTaskHandle = NULL;

void Blinky_Task(void *arg)
{
    esp_rom_gpio_pad_select_gpio(BLINK_GPIO);
    gpio_set_direction(BLINK_GPIO, GPIO_MODE_OUTPUT);

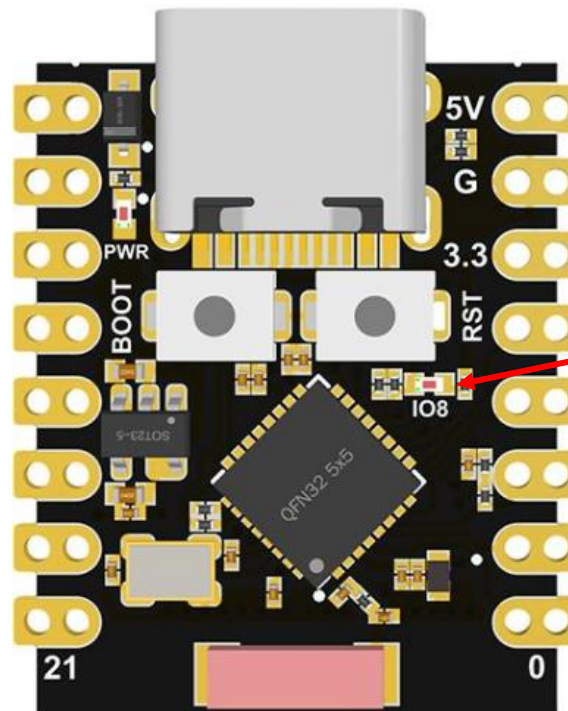
    while (1)
    {
        gpio_set_level(BLINK_GPIO, 1);
        vTaskDelay(1000 / portTICK_PERIOD_MS);
        gpio_set_level(BLINK_GPIO, 0);
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}

void app_main(void)
{

```

```
xTaskCreatePinnedToCore(Blinky_Task, "Blinky", 4096, NULL, 10, &BlinkyTaskHandle, 0); // Core 0  
}
```

Bước 3: Build project và nạp chương trình vào vi điều khiển, sau đó quan sát trạng thái của bóng đèn LED trên board



Quan sát đèn Led

Example 2.05

Mục tiêu: Tạo và quản lý GIPO. Thực hiện bật tắt đèn Led trên board ESP32-C3

Mục tiêu: Tạo và quản lý ứng dụng sử dụng FreeRTOS chạy trên vi điều khiển **ESP32**:

Yêu cầu: Ứng dụng thực hiện các chức năng sau:

- ✓ Thực hiện bật/tắt LED với chu kỳ 1000ms

Hướng dẫn:

Bước 1: Sử dụng Visual Studio Code, mở Command Palette (Ctrl+Shift+P) và nhập **IDF: New Project**) để bắt đầu tạo project với các thông tin như sau:

- ✓ Project Name: **example204**
- ✓ Enter Project directory: **D:\java-projects**
- ✓ Choose ESP-IDF Board: **ESP32-C3 chip(via ESP USB Bridge)**
- ✓ Choose serial port: COMx
- ✓ Choose Template: **template-app**

Hướng dẫn:

Bước 1: Sử dụng Visual Studio Code tạo Project với thông tin như sau:

- ✓ Project Name: **Lab02**
- ✓ Choose ESP-IDF Board: **ESP32-C3 chip (via ESP USB Bridge)**
- ✓ Enter Project directory: **D:\IoT-projects**

Bước 2: Sửa main.c file như sau:

main.c

```
#include <stdio.h>

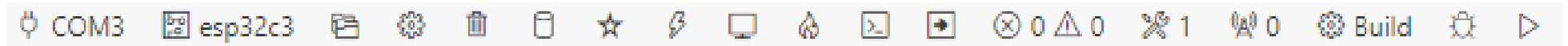
#include "driver\gpio.h"
#include "freeRTOS\freeRTOS.h"
#include "freeRTOS\task.h"

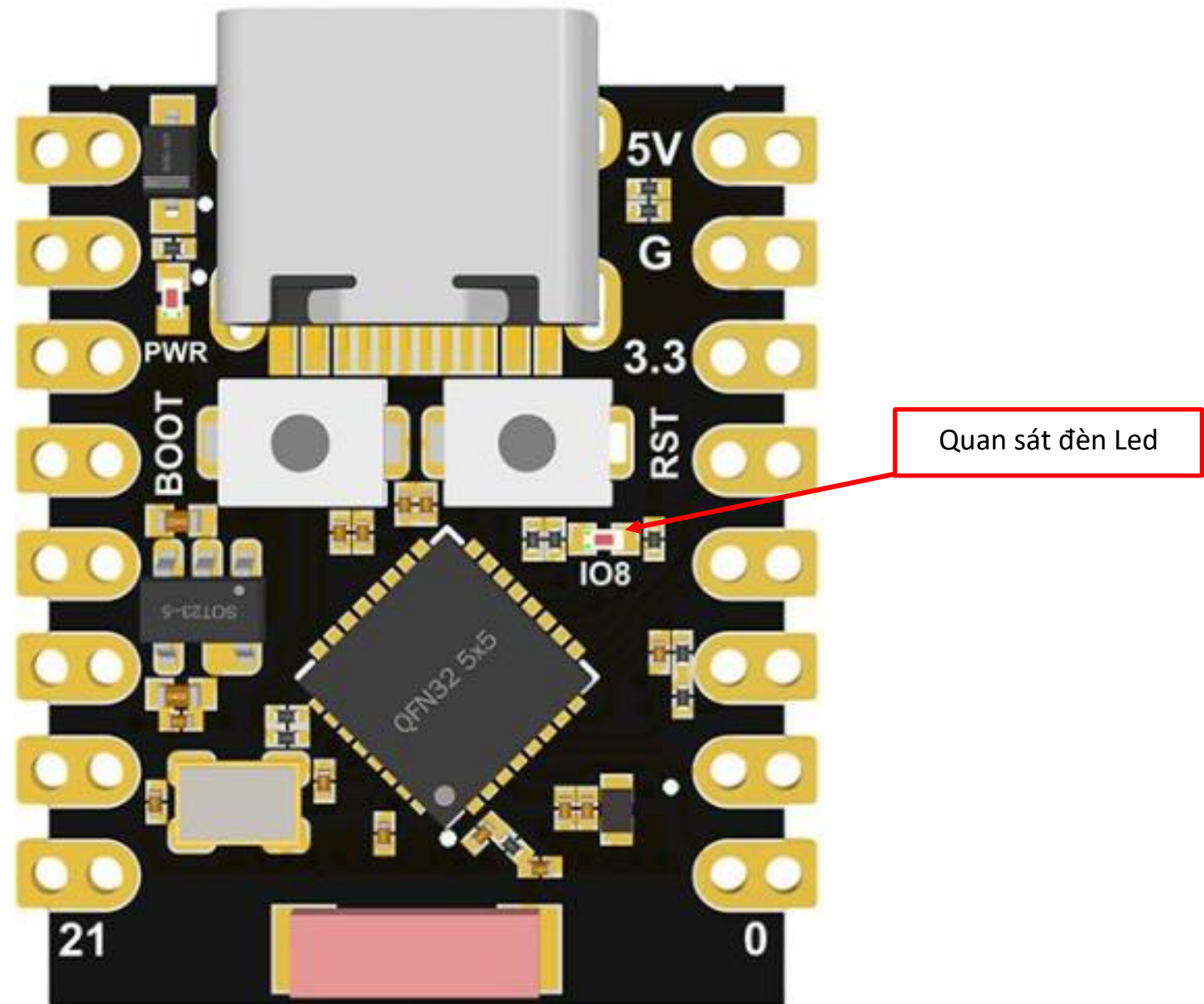
#define LED GPIO_NUM_8

void app_main(void)
{
    gpio_set_direction(LED, GPIO_MODE_DEF_OUTPUT);
    while(1)
    {
        gpio_set_level(LED, 1);
        vTaskDelay(100);
        gpio_set_level(LED, 0);
        vTaskDelay(100);
    }
}
```

Bước 3: Thực hiện build và flash chương trình lên board ESP32-C3

- ✓ Thực hiện mở Command Palette (Ctrl+Shift+P) sau đó chọn **ESP-IDF: Build your project** và **ESP-IDF: Flash (UART) your project**





Example 2.06

Mục tiêu: Tạo và quản lý ứng dụng sử dụng FreeRTOS chạy trên vi điều khiển **ESP32**:

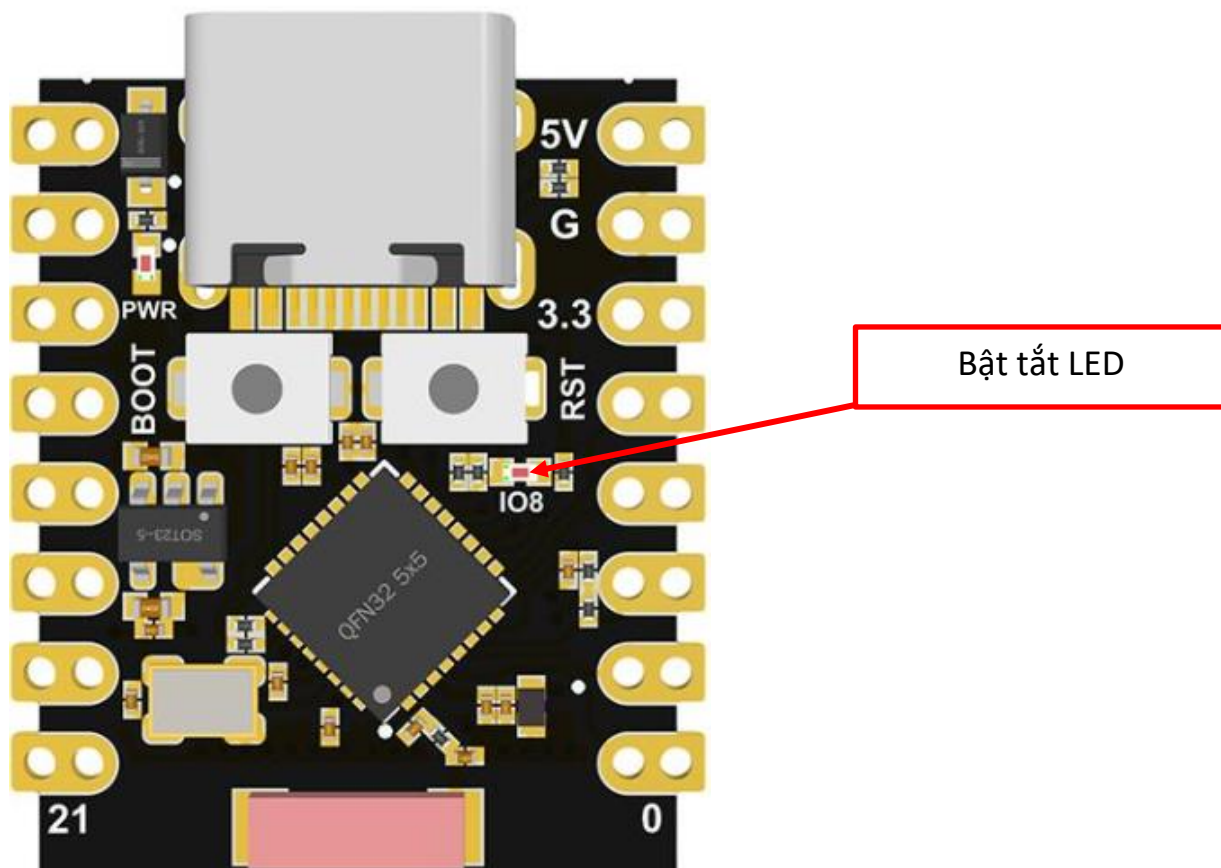
Yêu cầu: Ứng dụng thực hiện các chức năng sau:

- ✓ Thực hiện bật/tắt LED với chu kỳ 1000ms
- ✓ In ra Terminal nội dung Hello World ..

Hướng dẫn:

Bước 1: Sử dụng Visual Studio Code, mở Command Palette (Ctrl+Shift+P) và nhập **IDF: New Project**) để bắt đầu tạo project với các thông tin như sau:

- ✓ Project Name: **example205**
- ✓ Enter Project directory: **D:\java-projects**
- ✓ Choose ESP-IDF Board: **ESP32-C3 chip(via ESP USB Bridge)**
- ✓ Choose serial port: COMx



Bước 2: Sửa file **blink_main.c** như sau:

blink_main.c

```
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "driver/gpio.h"
#include "esp_log.h"
#include "sdkconfig.h"

#define BLINK_GPIO GPIO_NUM_32

TaskHandle_t HelloWorldTaskHandle = NULL;
TaskHandle_t BlinkyTaskHandle = NULL;

void HelloWorld_Task(void *arg)
{
    while (1)
    {
        printf("Task running: Hello World ..\n");
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}

void Blinky_Task(void *arg)
{
    esp_rom_gpio_pad_select_gpio(BLINK_GPIO);
    gpio_set_direction(BLINK_GPIO, GPIO_MODE_OUTPUT);

    int count_second = 0;

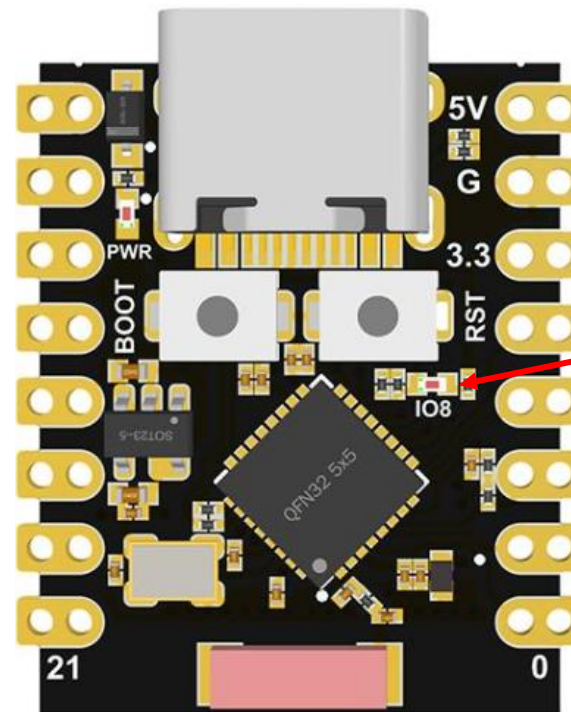
    while (1)
    {
```

```
        count_second += 1;

        switch (count_second)
        {
        case 10:
            vTaskSuspend(HelloWorldTaskHandle);
            printf("HelloWorld task suspended .. \n");
            break;
        case 14:
            vTaskResume(HelloWorldTaskHandle);
            printf("HelloWorld task resumed .. \n");
            break;
        case 20:
            vTaskDelete(HelloWorldTaskHandle);
            printf("HelloWorld task deleted .. \n");
            break;
        default:
            break;
        }
        gpio_set_level(BLINK_GPIO, 1);
        vTaskDelay(1000 / portTICK_PERIOD_MS);
        gpio_set_level(BLINK_GPIO, 0);
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}

void app_main(void)
{
    xTaskCreatePinnedToCore(Blinky_Task, "Blinky", 4096, NULL, 10, &BlinkyTaskHandle, 0); // Core 0
    xTaskCreatePinnedToCore(HelloWorld_Task, "HelloWorld", 4096, NULL, 10, &HelloWorldTaskHandle, 1); // Core 1
}
```

Bước 3: Build project và nạp chương trình vào vi điều khiển, sau đó quan sát trạng thái của bóng đèn LED trên board



Quan sát đèn Led

