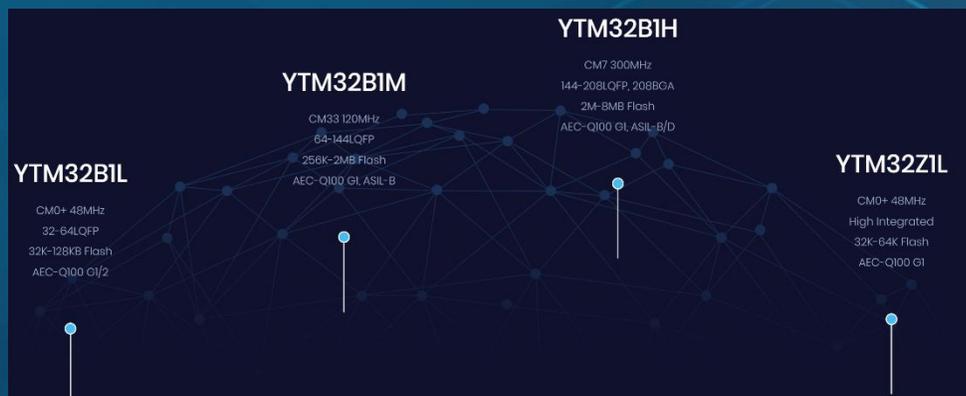
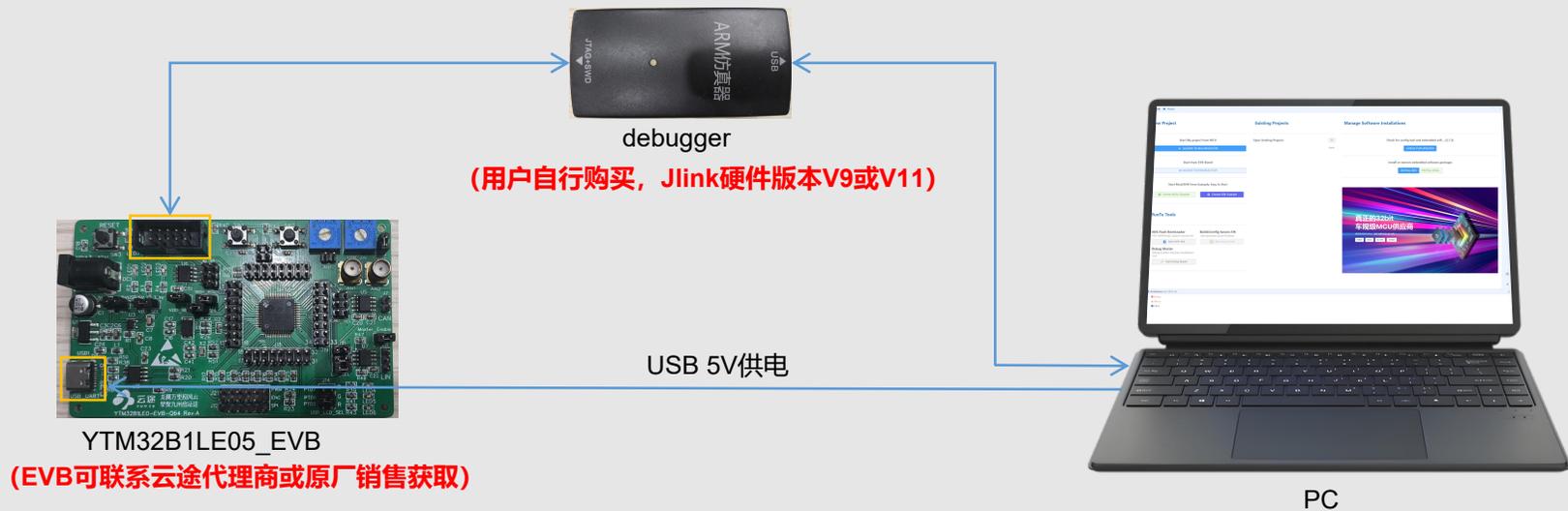


云途MCU开发入门介绍——开发环境搭建



准备

PREPARE



- ✓ 硬件控制板 (YT_EVB or 用户硬件)
- ✓ 调试器 (推荐使用Jlink)
- ✓ 电脑 (集成开发环境、云途配置工具、Jlink驱动软件、开发文档等)
- ✓ 电源 (建议USB-TypeC 5V供电)

目录

CONTENTS

1. 云途用户系统及配置工具(YT Config Tool)概述
2. 集成开发环境搭建
3. Jlink软件安装及patch
4. 运行demo工程
5. 创建用户工程
6. FAQ

01

云途用户系统及配置工具 (YT Config Tool) 概述

云途用户系统及配置工具(YT Config Tool)概述

云途用户系统和配置工具YCT (YT Config Tool) 是客户获取各系列芯片资料，快速入门开发的必需工具，具体功能介绍如下。

1.注册并登录云途用户系统 (<https://account.ytmicro.com>) 后，可获取如下开发资料

开发文档

开发文档

- 选择子组SG
- 选择子组DS
- 查看手册RM
- 选择子组ES
- 应用手册AN
- 硬件设计支持资料

YCT工具使用手册

MCAL用户手册

开发工具

- YCT软件安装包
- 云途IDE安装包

开发工具

YCT软件安装包

云途IDE安装包

YT Config Tool Download Pages

YT Config Tool 2.4.0

YT Config Tool 2.4.0.exe

yt_config_tool_2.4.0_arm54.deb

软件包：SDK、MCAL、SafLib、中间组件等离线包

软件包：SDK、MCAL、SafLib、中间组件等离线包

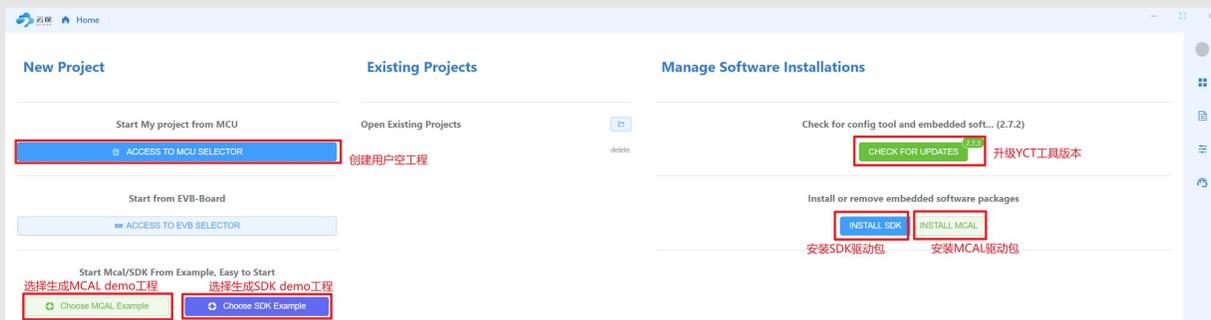
YUNTU SDK Download Pages

Name	Version	Type	Size	ModifyTime	Operations
YTM32B1ME0	1.3.1	SDK	Size:5726108Byte	2023-11-09T09:33:25.000Z	Readme Download
freertos	1.8.5	Middleware	Size:5161658Byte	2023-11-09T09:39:55.000Z	Readme Download
uIRy-pint	1.3.4	Middleware	Size:442878Byte	2023-11-09T09:16:03.000Z	Readme Download

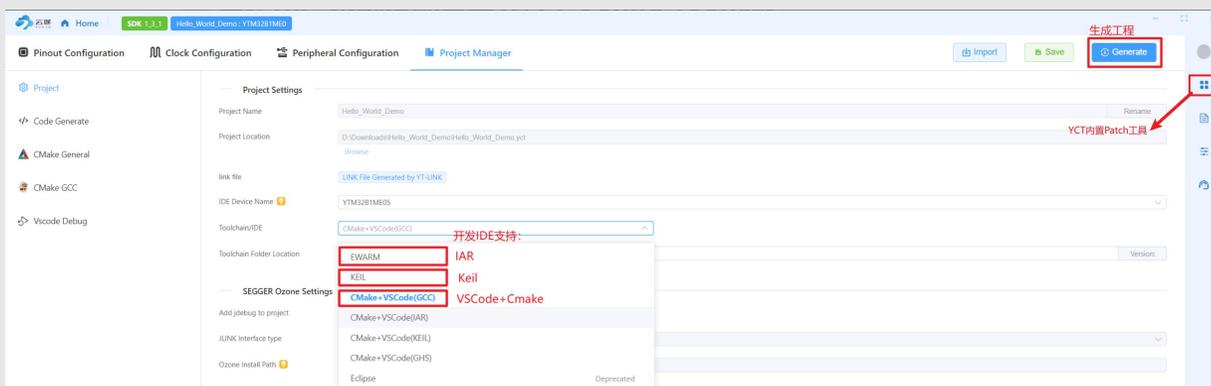
云途用户系统及配置工具(YT Config Tool)概述

2. 云途用户系统 **同一账号** 可用于YCT软件登录

注意：若用户需使用开源的VSCode (GCC+Nijia+Cmake) 工具链，安装YCT时可自动部署环境（详见第二章：VSCode 环境搭建）



- 支持生成SDK、MCAL的demo工程
- 支持创建SDK、MCAL用户空工程并进行底层驱动配置



- 支持生成IAR、Keil、VSCode三种工程
- 支持使用YCT内置的patch工具安装芯片pack包(详见第三章介绍)

02

集成开发环境搭建

- ✓ 概述
 - ✓ 芯片包获取
 - ✓ Keil环境搭建
 - ✓ IAR环境搭建
 - ✓ VSCode (GCC+Nijia+Cmake) 环境搭建
-

集成开发环境搭建 - 概述

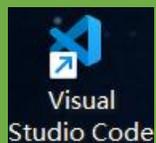
集成开发环境（Integrated Development Environment，简称 IDE）可以为用户提供一个软件开发平台，实现代码编辑、集成编译、调试功能。这里主要介绍以下三种IDE：



- 安装Keil：使用需要licence，用户自行获取
- Keil Patch



- 安装IAR：使用需要licence，用户自行获取
- IAR Patch

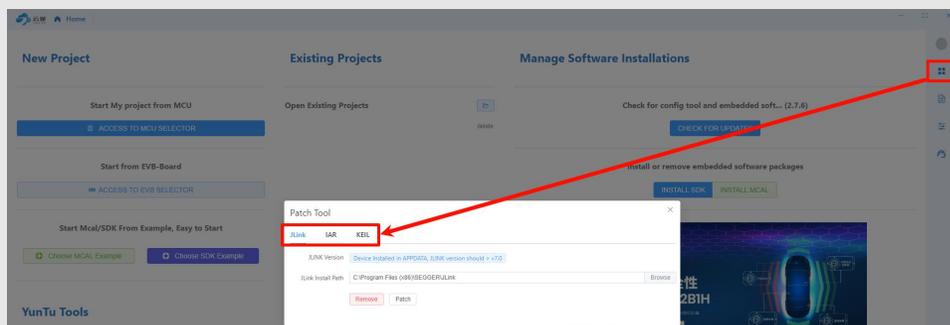


- 开源工具链VSCoDe(GCC+Ninja+Cmake)，无需licence
- 安装VSCoDe：编辑、编译软件
- 安装Ozone：调试软件

集成开发环境搭建 - 芯片包获取

我们提供离两种方式用于Keil、IAR、Jlink工具的patch。

方式1: YCT配置工具联网登录后, 直接通过配置工具打patch (具体操作方式详见后文)



方式2: 获取离线芯片包进行patch (目前最新版本patch包为V3.5, 后续更新后会同步在论坛帖: [Keil、IAR、JLINK如何添加云途芯片 | YunTu ForumYCT](#))

集成开发环境搭建 - Keil

1. 安装Keil (MDK) 软件

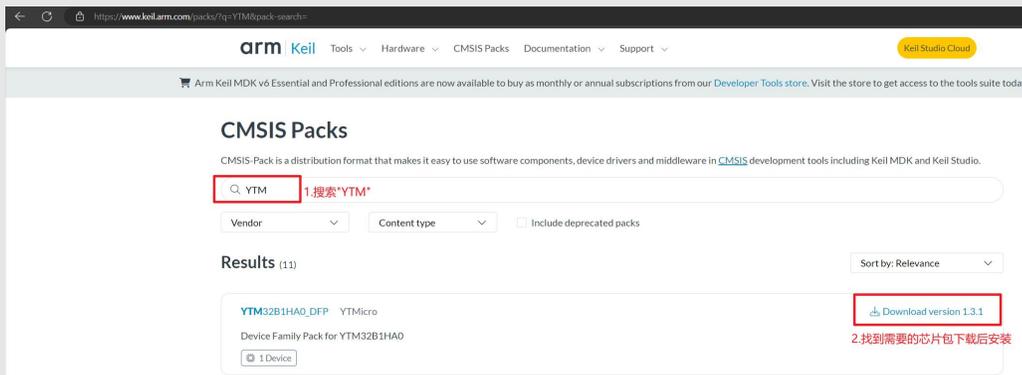
- 软件版本要求：V5.36及之后的版本
- 软件安装包获取：用户自行从keil官网(<https://www.keil.com/update/>)下载安装

2. Keil patch

建议安装所有系列芯片的pack包

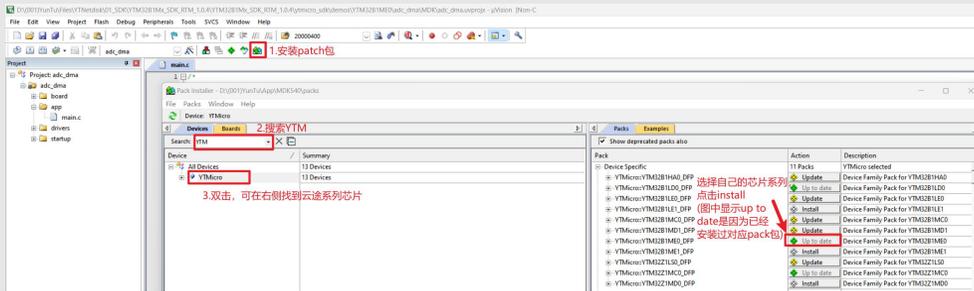
- 可联网用户

(1) Keil官网获取芯片pack包<https://www.keil.arm.com/packs/>

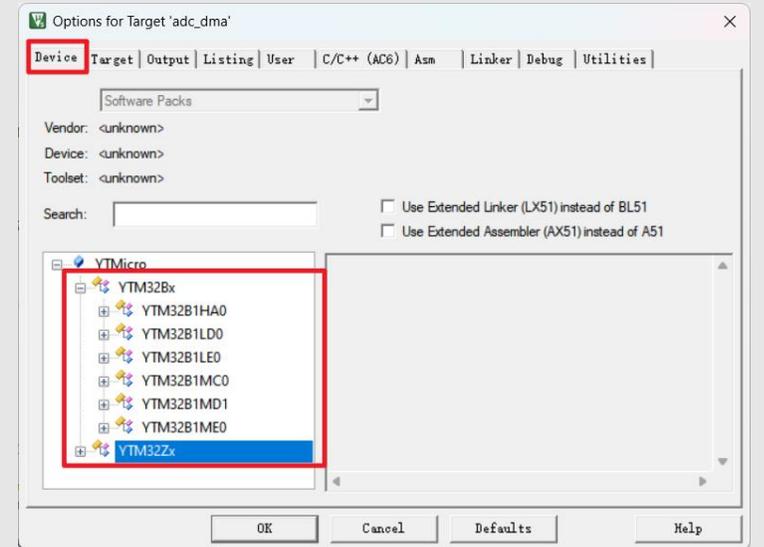
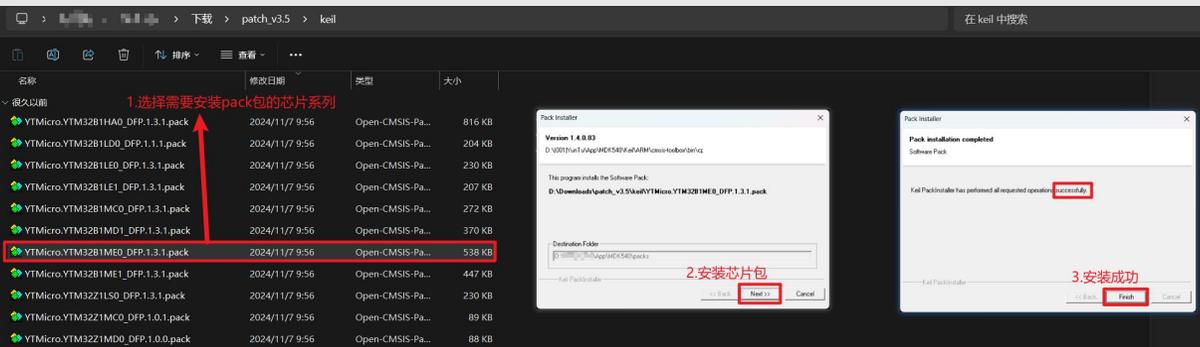


集成开发环境搭建 - Keil

- 可联网用户
- (2) 打开keil软件直接搜索安装芯片pack包



- 无法联网用户
- 下载patch包进行patch(可能不定期更新)
方式如下图



pack安装成功后可在keil软件的decive中找到YT芯片包

集成开发环境搭建 - IAR

1. 安装IAR (EWARM) 软件

- 软件版本要求：V8.40及之后的版本，

注意：IAR 9.50版本之前需要手动patch，IAR 9.50版本(含)之后IAR安装包已包含芯片包，无需手动patch

- 软件安装包获取：用户自行从IAR官网(<https://updates.iar.com/?product=EWARM>)下载安装

2. IAR patch

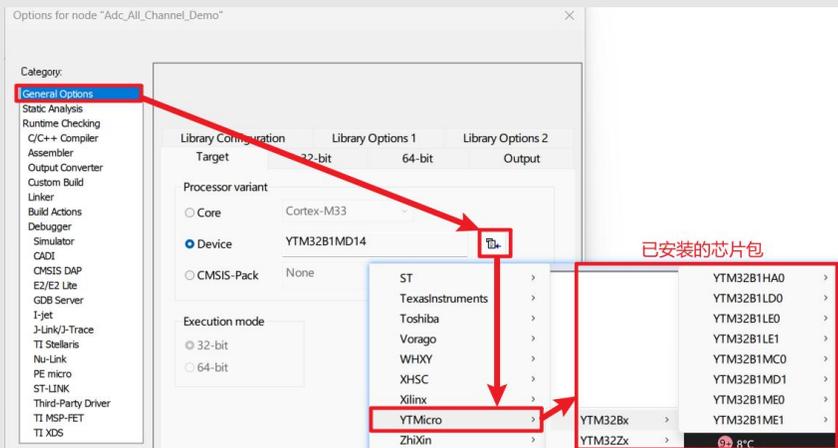
- 方式1：通过YCT软件内置patch tool进行patch

注意：YCT软件必须“以管理员身份打开”



集成开发环境搭建 - IAR

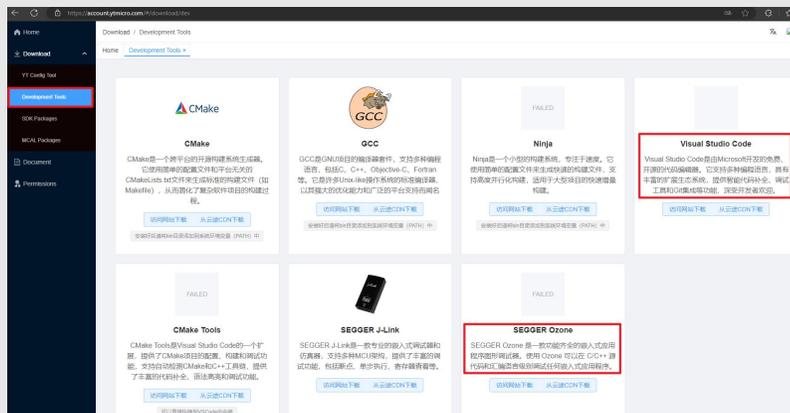
- 方式2: 通过patch包进行patch



patch成功后可在IAR软件的
general Options中找到YT芯片包

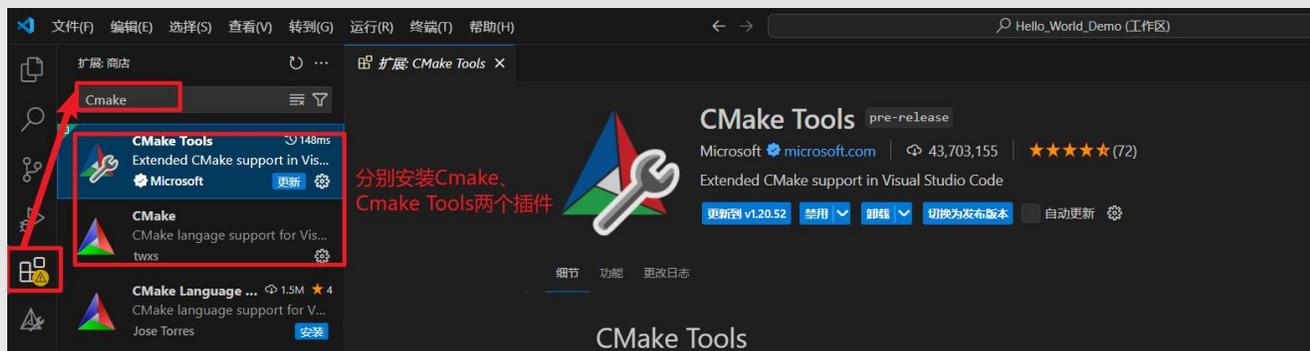
集成开发环境搭建 - VSCode (GCC+Ninja+Cmake)

1. 安装Vscode软件、ozone软件



- VSCode软件用于编辑，编译代码
- ozone软件用于调试代码
- 软件安装包获取
 - (1)云途CDN下载安装
 - (2)官网下载安装

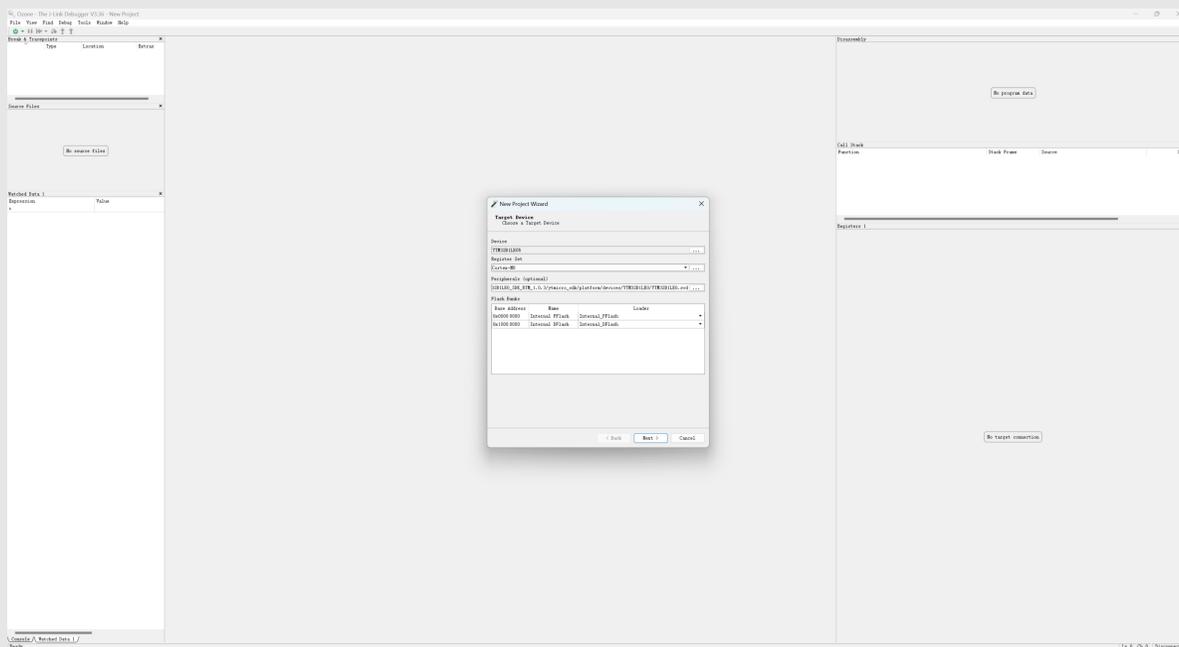
2. Vscode安装后，还需安装Cmake、Cmake Tools插件



集成开发环境搭建 - VSCode (GCC+Ninja+Cmake)

3. ozone 介绍

Ozone是SEGGER旗下的一款免费跨平台支持的**代码调试软件**，支持Segger J-link和J-Trace调试器。可以加载IAR、keil、GCC等工具链编译生成的可执行文件，进行C/C++源码调试和汇编指令调试。



集成开发环境搭建 - VSCode (GCC+Ninja+Cmake)

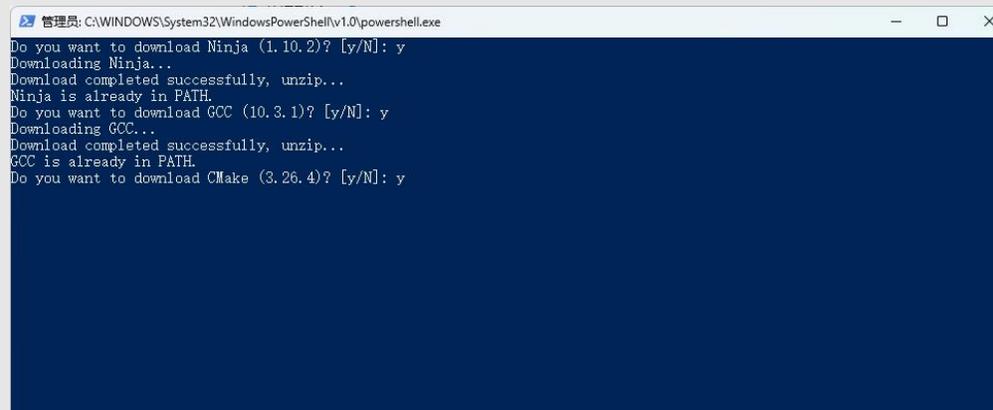
4.Cmake、Ninja、GCC环境部署（使工程可以在VSCode环境下编译）

建议在安装YCT软件时同步完成部署工作：

下图左弹窗出现时选择“是”，弹出右图终端窗口时，依此输入三次“y”自动下载并部署Ninja、GCC、Cmake环境。

注意：

- 该环境部署工作只需一次即可，后续更新YCT软件时无需再次部署
- 无需使用VSCode环境的用户在下图左弹窗选择“否”即可



03

Jlink 软件安装及 Patch

- ✓ Jlink 版本说明
 - ✓ Jlink patch
-

Jlink软件安装及Patch - Jlink版本说明

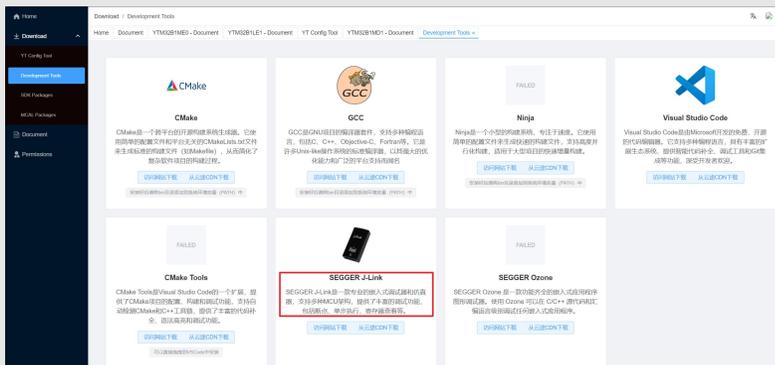
调试器用于连接用户PC和硬件电路板，实现软件调试功能。

无论用户使用哪种IDE，**均需要安装Jlink驱动软件并进行Jlink patch**操作。

1.jlink硬件版本要求V9，V11(推荐使用V9版本)



2.jlink驱动软件版本要求V7.92及以上



软件安装包获取:

(1)云途CDN下载安装<https://account.ytmicro.com/#/download/devhttps://account.ytmicro.com/#/download/dev>

(2)官网下载安装

Jlink软件安装及Patch - Jlink patch

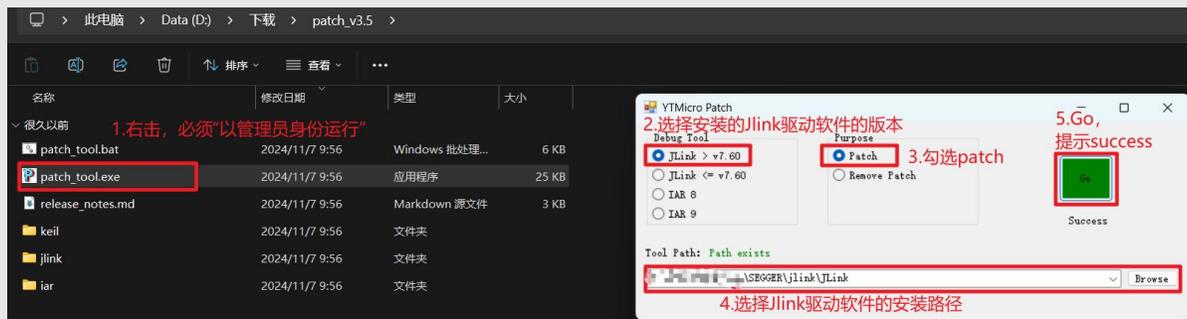
1.jlink驱动程序打patch

- 方式1: 通过YCT软件内置patch tool进行patch

注意: YCT软件必须“以管理员身份打开”

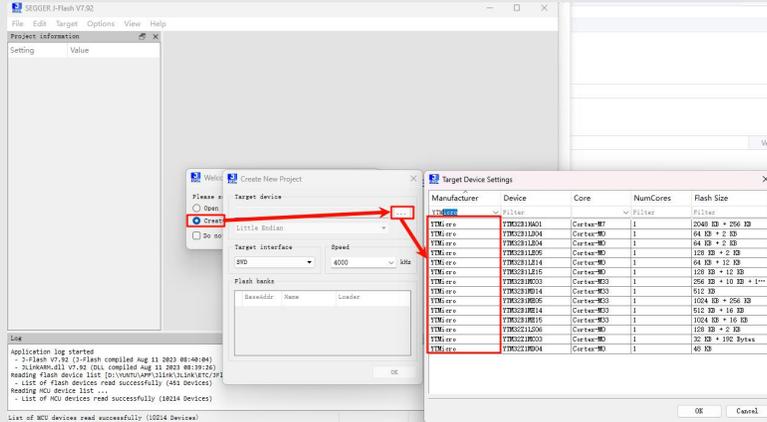


- 方式2: 通过patch包进行patch



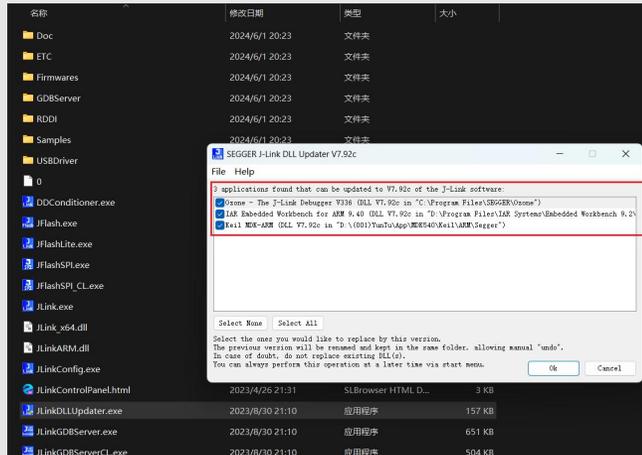
Jlink软件安装及Patch - Jlink patch

2.jlink打patch成功标志



patch成功后可在j-flash软件的device中找到YT芯片包

3.运行JLinkDLLUpdater可执行程序，下图弹窗表示Jlink已与对应的IDE关联起来(该项为可选项，patch成功后会自动运行JLinkDLLUpdater，若未自动运行，需手动运行)



运行JLinkDLLUpdater后，可以看到jlink V7.92已成功与keil、IAR、ozone相关联
注意：对于用户而言，用到哪种IDE确认关联该IDE即可，无需关联所有

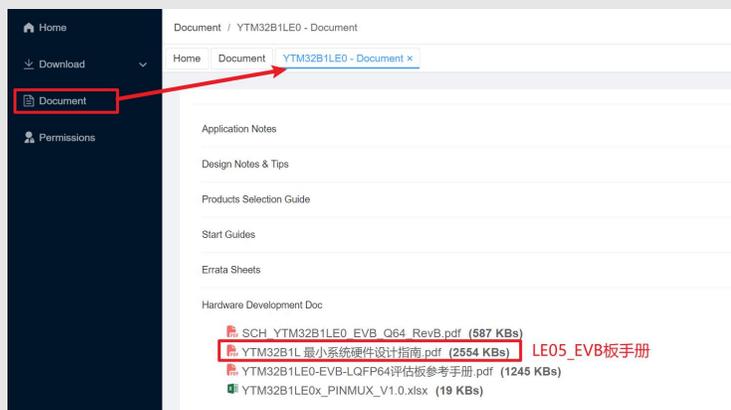
04

运行demo工程

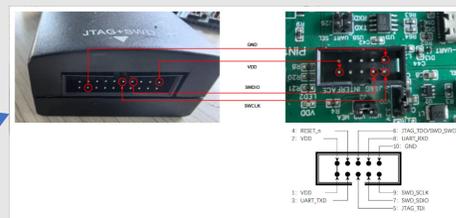
- ✓LE05_EVB硬件接线
 - ✓YCT生成demo工程
 - ✓Keil运行demo工程
 - ✓IAR运行demo工程
 - ✓VSCode工具链运行demo工程
-

运行demo工程 - LE05_EVB硬件接线

以YTM32B1LE05 EVB为例，介绍硬件连接，主要是**电源**和**调试器接线**，其它外设详细描述见评估板参考手册文档。



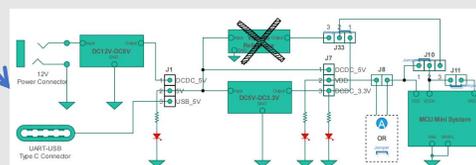
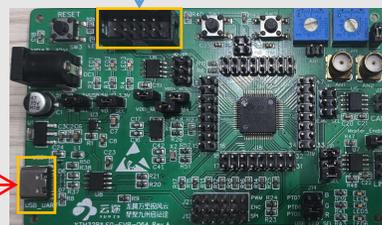
云途用户系统：LE05_EVB评估板手册



接VCC、GND、SWD_IO、SWD_CLK共4根线即可



DC 5V



支持5V/12V供电，建议使用USB线5V供电

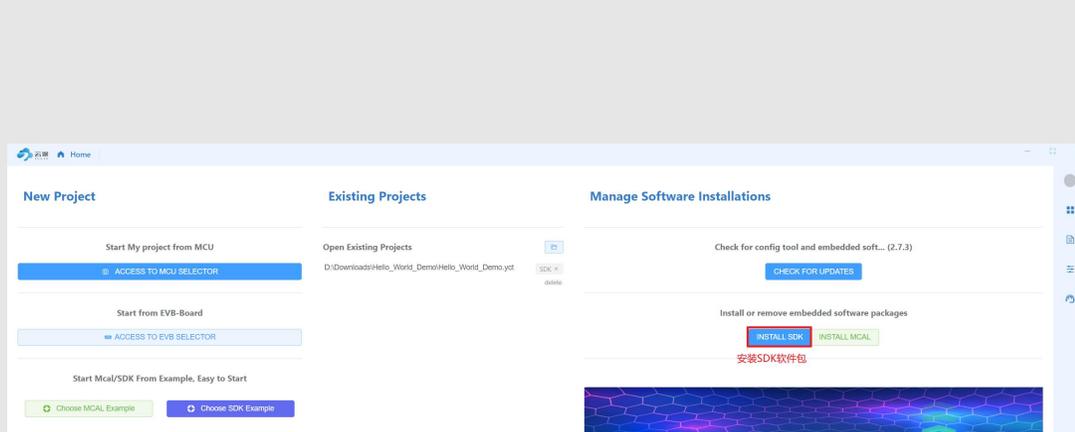
以USB 5V供电为例，

- J1跳右侧：选择USB 5V供电
- J2左右均可，选择MCU供电电压
- J3、J4、J5均默认短接

运行demo工程 - YCT生成demo工程

以SDK demo软件为例(MCAL软件操作流程相似), 介绍LE05如何使用YCT生成demo工程。

1. 安装SDK软件包



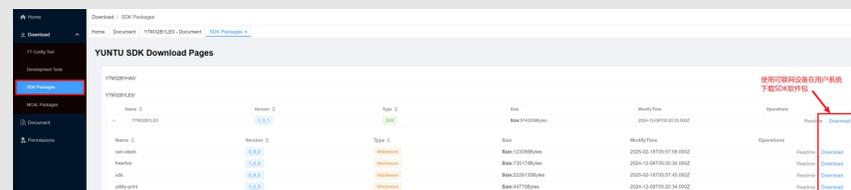
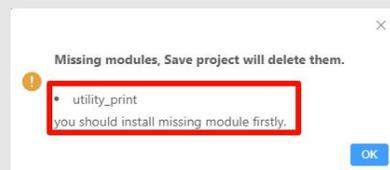
在线安装

离线安装

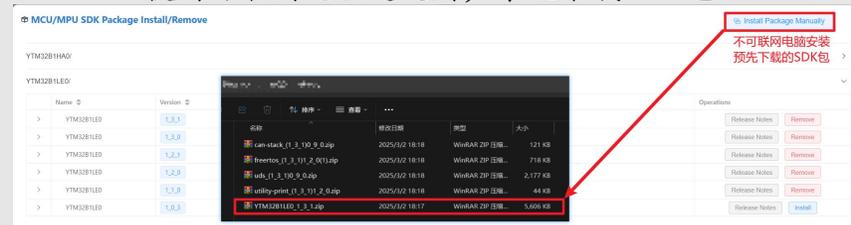


注意:
YCT配置工具中提供的demo软件均带有串口打印代码, 下载SDK包时必须下载中间件utility-print (下图左), 否则生成工程时会报错 (下图右)

Name	Version	Type	Size	ModifyTime	Operations
YTM32B1LE0	1.3.1	SDK	Size:5740359Bytes	2024-12-09T05:20:55.000Z	Release Notes Remove
middleware available for this release 1.3.1					
Name	Version	Type	Size	ModifyTime	Operations
YTM32B1LE0_1.3.1_HOTFIX	001	SDK_HOTFIX	Size:5359812Bytes	2025-06-10T06:28:13.000Z	Release Notes Remove
can-stack	0.3.0	Middleware	Size:120369Bytes	2025-02-18T05:57:58.000Z	Release Notes Remove
freertos	1.2.0	Middleware	Size:731748Bytes	2024-12-09T05:20:54.000Z	Release Notes Remove
uks	0.3.0	Middleware	Size:2291358Bytes	2025-02-18T05:57:45.000Z	Release Notes Remove
utility-print	1.3.0	Middleware	Size:447705Bytes	2024-12-09T05:20:54.000Z	Release Notes Remove



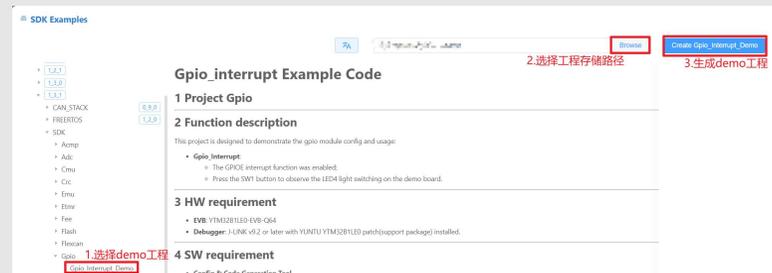
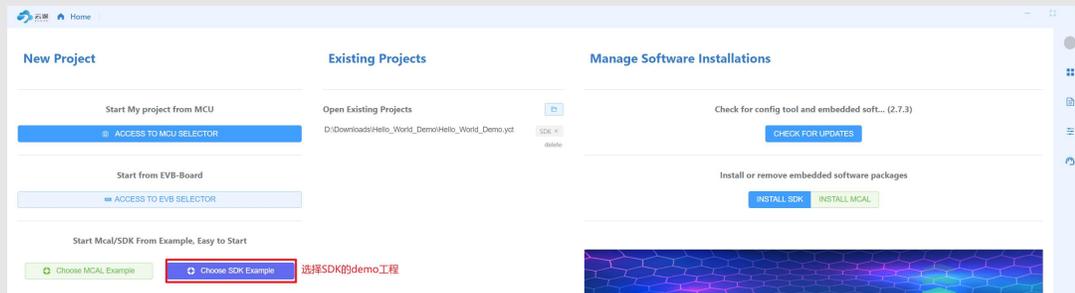
使用可联网电脑通过用户系统下载SDK包



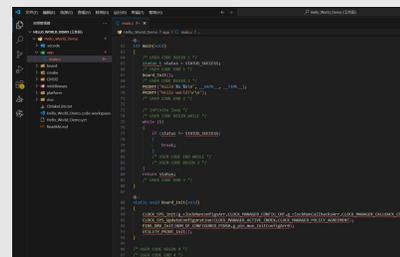
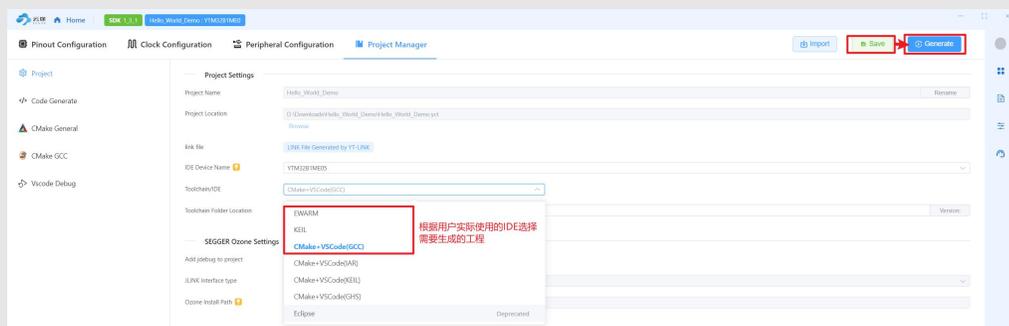
将预先下载的SDK包安装到离线电脑

运行demo工程 - YCT生成demo工程

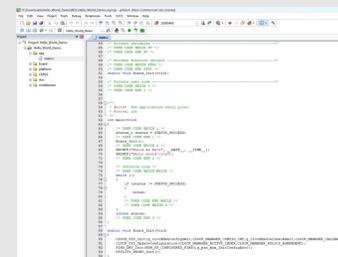
2. 选择SDK demo工程，此处以“Gpio_Interrupt_Demo”工程为例



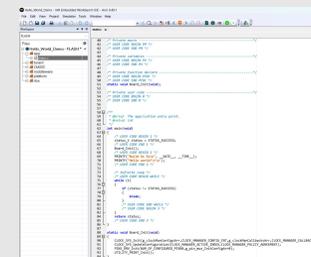
3. 选定IDE后生成demo工程
(点击生成后会根据选择的IDE自动打开keil或IAR或VSCode工程)



VSCode工程



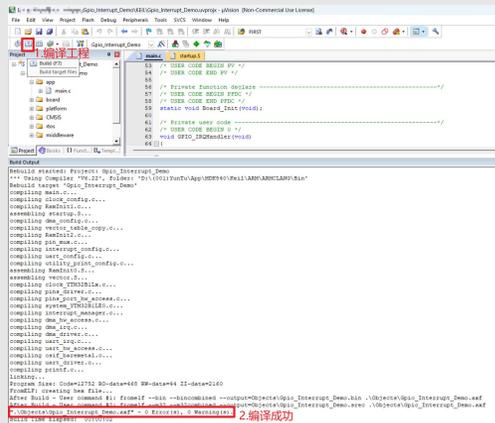
keil工程



IAR工程

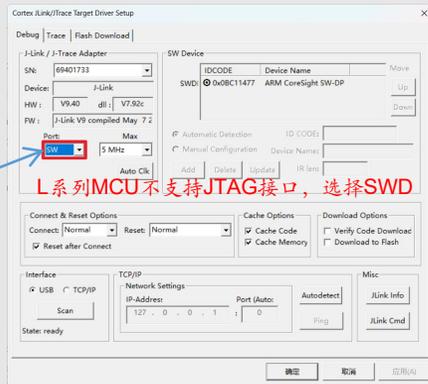
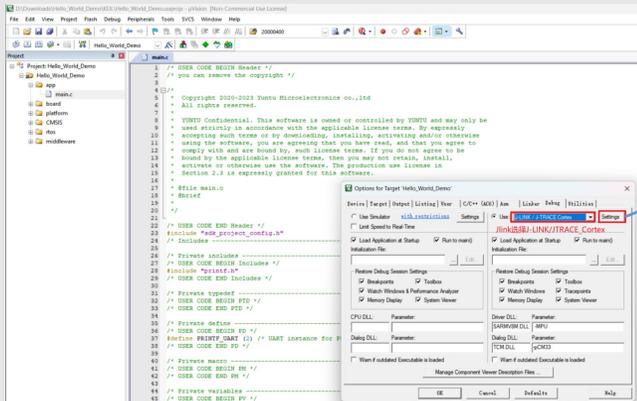
运行demo工程 - Keil运行demo工程

1. 工程编译



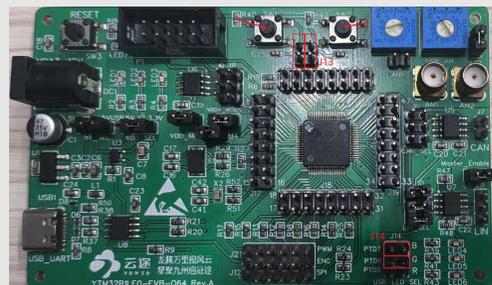
2. Debugger选择

初次生成的keil工程中debugger有一个默认选项，用户需根据实际的debugger来选择，例如jlink选择J-LINK / J-TRACE Cortex



3. 烧录代码并运行

注意：下图中标注的J13两个跳帽、J14两个跳帽需要短接。



- 按下SW1，LED4（蓝灯）亮灭状态反转
- 按下SW2，LED5（绿灯）亮灭状态反转

调试软件，按下SW1时，触发GPIO中断，程序运行到下图断点位置

```
63 void GPIO_IRQHandler(void)
64 {
65     if (PINS_DRV_GetPortIntFlag(SW1_PORT) & (1 << SW1_PORT_INDEX))
66     {
67         /* SW1 is pressed, toggle LED 4 */
68         PINS_DRV_TogglePins(LED1_PORT, (1 << LED1_PORT_INDEX));
69
70         /* Clear interrupt flag. */
71         PINS_DRV_ClearPinIntFlagCmd(SW1_PORT, SW1_PORT_INDEX);
72     }
73     if (PINS_DRV_GetPortIntFlag(SW2_PORT) & (1 << SW2_PORT_INDEX))
74     {
75         /* SW2 is pressed, toggle LED2 */
76         PINS_DRV_TogglePins(LED2_PORT, (1 << LED2_PORT_INDEX));
77         /* Clear interrupt flag. */
78         PINS_DRV_ClearPinIntFlagCmd(SW2_PORT, SW2_PORT_INDEX);
79     }
80 }
```

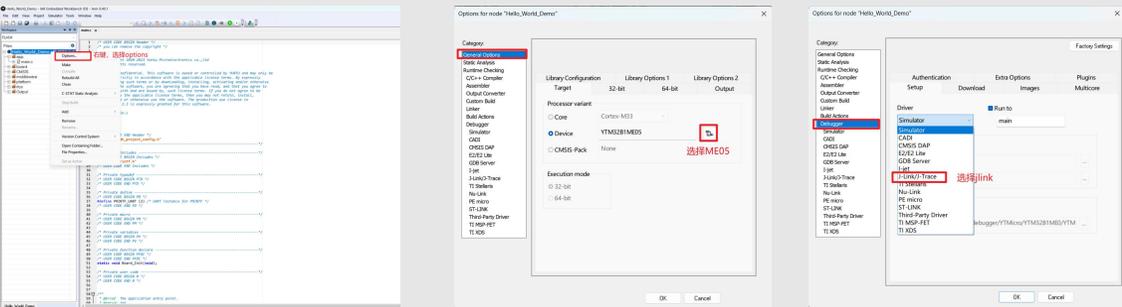
运行demo工程 - IAR运行demo工程

1.工程编译



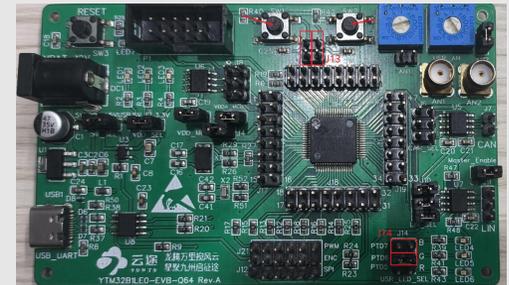
2. Debugger选择

初次生成的keil工程中debugger默认是Simulator，用户需根据实际的debugger来选择，例如jlink选择J-LINK / J-TRACE Cortex



3.烧录代码并运行

注意：下图中标注的J13两个跳帽、J14两个跳帽需要短接。



- 按下SW1，LED4（蓝灯）亮灭状态反转
- 按下SW2，LED5（绿灯）亮灭状态反转

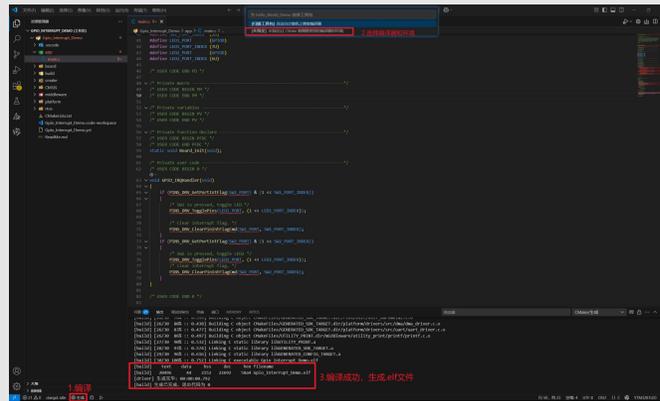
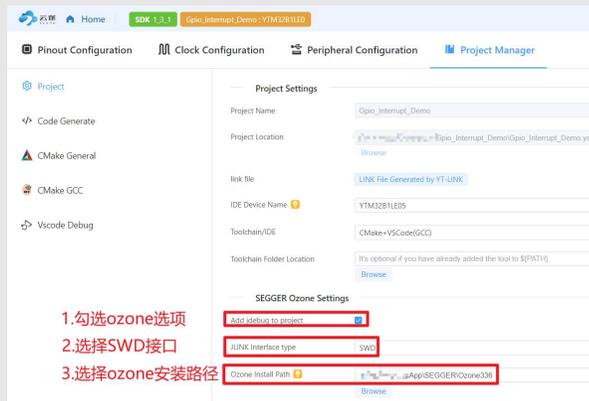
调试软件，按下SW1时，触发GPIO中断，程序运行到下图断点位置

```
63 void GPIO_IRQHandler(void)
64 {
65     if (PINS_DRV_GetPortIntFlag(SW1_PORT) & (1 << SW1_PORT_INDEX))
66     {
67         /* SW1 is pressed, toggle LED4 */
68         PINS_DRV_TogglePins(LED1_PORT, (1 << LED1_PORT_INDEX));
69
70         /* Clear interrupt flag. */
71         PINS_DRV_ClearPinIntFlagCmd(SW1_PORT, SW1_PORT_INDEX);
72     }
73     if (PINS_DRV_GetPortIntFlag(SW2_PORT) & (1 << SW2_PORT_INDEX))
74     {
75         /* SW2 is pressed, toggle LED2 */
76         PINS_DRV_TogglePins(LED2_PORT, (1 << LED2_PORT_INDEX));
77         /* Clear interrupt flag. */
78         PINS_DRV_ClearPinIntFlagCmd(SW2_PORT, SW2_PORT_INDEX);
79     }
80 }
```

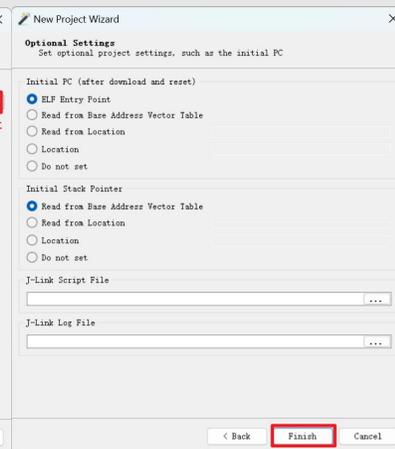
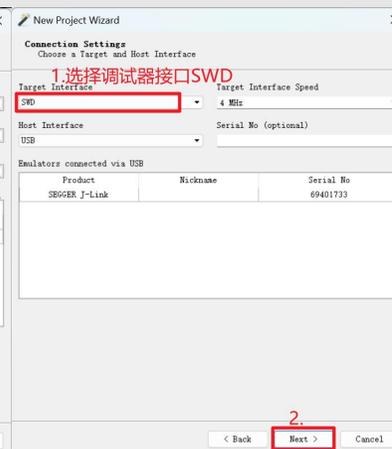
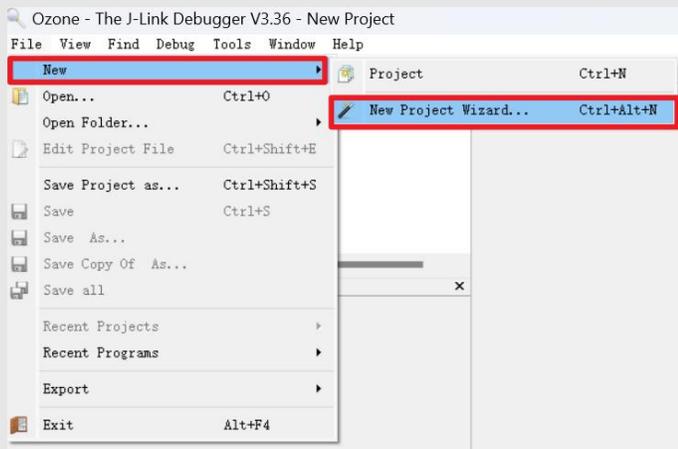
运行demo工程 - VSCode运行demo工程

1. 工程编译并导入ozone

- 方式1: 适用于VSCode+GCC工具链用户, 可执行文件自动导入ozone工程

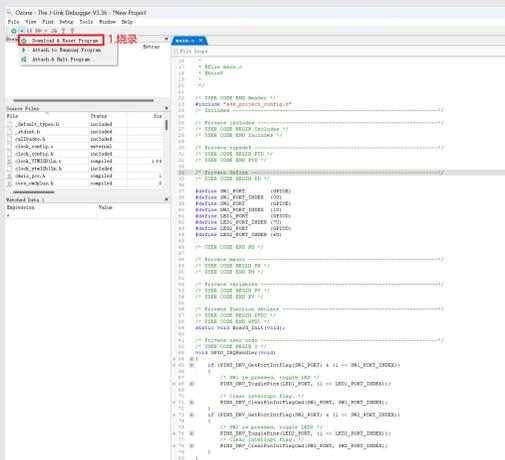


- 方式2: 适用于Keil、IAR用户, 手动创建ozone工程, 导入可执行文件

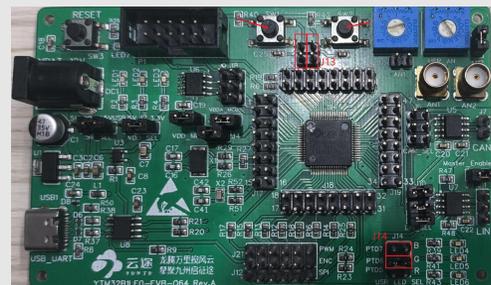


运行demo工程 - VSCode运行demo工程

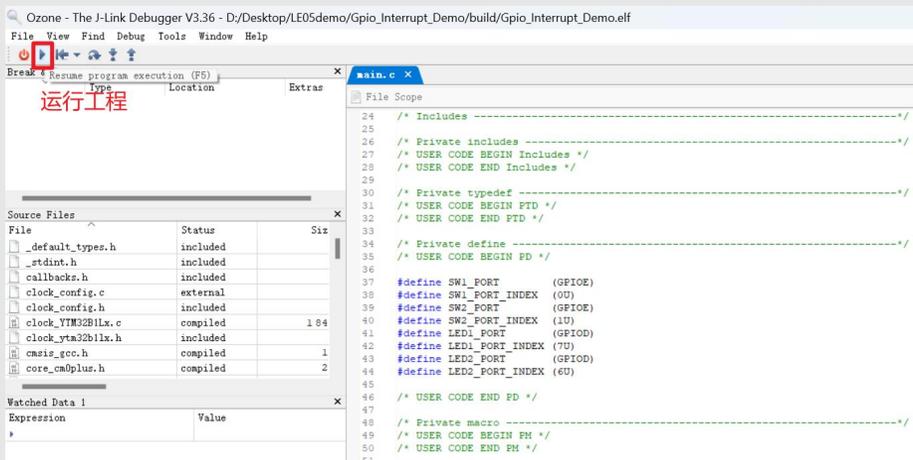
2.烧录代码并运行



注意：下图中标注的J13两个跳帽、J14两个跳帽需要短接。



- 按下SW1, LED4 (蓝灯) 亮灭状态反转
- 按下SW2, LED5 (绿灯) 亮灭状态反转



调试软件, 按下SW1时, 触发GPIO中断, 程序运行到下图断点位置

```
63 void GPIO_IRQHandler(void)
64 {
65     if (PINS_DRV_GetPortIntFlag(SW1_PORT) & (1 << SW1_PORT_INDEX))
66     {
67         /* SW1 is pressed, toggle LED */
68         PINS_DRV_TogglePins(LED1_PORT, (1 << LED1_PORT_INDEX));
69     }
70     /* Clear interrupt flag. */
71     PINS_DRV_ClearPinIntFlagCmd(SW1_PORT, SW1_PORT_INDEX);
72 }
73 if (PINS_DRV_GetPortIntFlag(SW2_PORT) & (1 << SW2_PORT_INDEX))
74 {
75     /* SW2 is pressed, toggle LED2 */
76     PINS_DRV_TogglePins(LED2_PORT, (1 << LED2_PORT_INDEX));
77     /* Clear interrupt flag. */
78     PINS_DRV_ClearPinIntFlagCmd(SW2_PORT, SW2_PORT_INDEX);
79 }
80 }
```

05

创建用户工程

- ✓ 创建VSCode工程并运行
 - ✓ 工程中添加用户代码
-

创建用户工程——创建VSCode工程并运行

以YTM32B1LE05为例，创建一个LED每500ms闪烁一次的SDK用户工程，并基于LE05 EVB板测试运行。

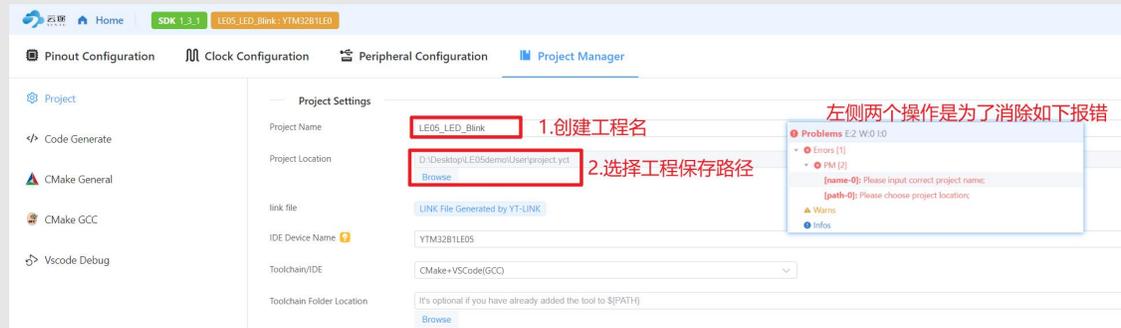
1. 登录YCT，确保已安装SDK软件包，按照下图步骤创建LE05 SDK空工程。

The screenshot shows the YCT MCU/MPU Selector interface. A red box highlights the 'ACCESS TO MCU SELECTOR' button. A red arrow points to the 'YTM32B1LE05' chip in the 'Family' dropdown menu, labeled '1. 选择芯片系列'. The main table lists various chip models with their specifications. A red box highlights the 'YTM32B1LE05' row, labeled '2. 根据用户硬件选择封装, 这里选择64pin封装'. The 'Part Number' field is set to 'YTM32B1LE05H0MLET'. The 'SDK Version' dropdown is set to '1.3.3', labeled '3. 选择最新版本SDK'. A red box highlights the 'Choose' button at the bottom right, labeled '4. Choose'.

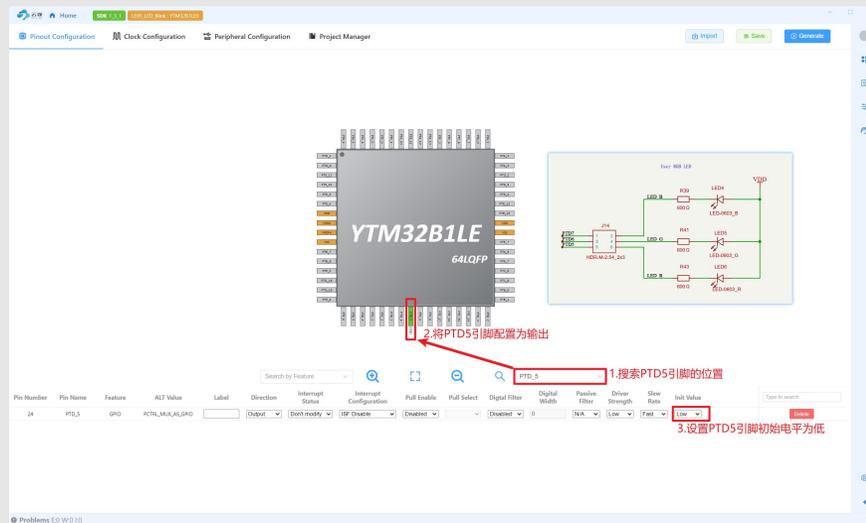
Family	Type	Part Number	Package	Flash	RAM	I/O	GPIO	ADC Channels	CAN	Frequency	High Voltage ID	Lin ID
YTM32B1LE05	SDK	YTM32B1LE05H0MLET	LQFP	128KB	19KB	64	58	16	1	48MHz		
YTM32B1LE05	SDK	YTM32B1LE05H0MLET	LQFP	128KB	16KB	32	28	12	1	48MHz		
YTM32B1LE05	SDK	YTM32B1LE05H0MLET	LQFP	128KB	16KB	32	28	9	1	48MHz		
YTM32B1LE05	SDK	YTM32B1LE05H0MLET	LQFP	128KB	16KB	32	28	9	1	48MHz		
YTM32B1LE05	SDK	YTM32B1LE05H0MLET	LQFP	64KB	8KB	48	43	13	0	48MHz		
YTM32B1LE05	SDK	YTM32B1LE05H0MLET	LQFP	64KB	8KB	32	28	12	0	48MHz		
YTM32B1LE05	SDK	YTM32B1LE05H0MLET	LQFP	64KB	8KB	32	28	9	0	48MHz		

创建用户工程——创建VSCode工程并运行

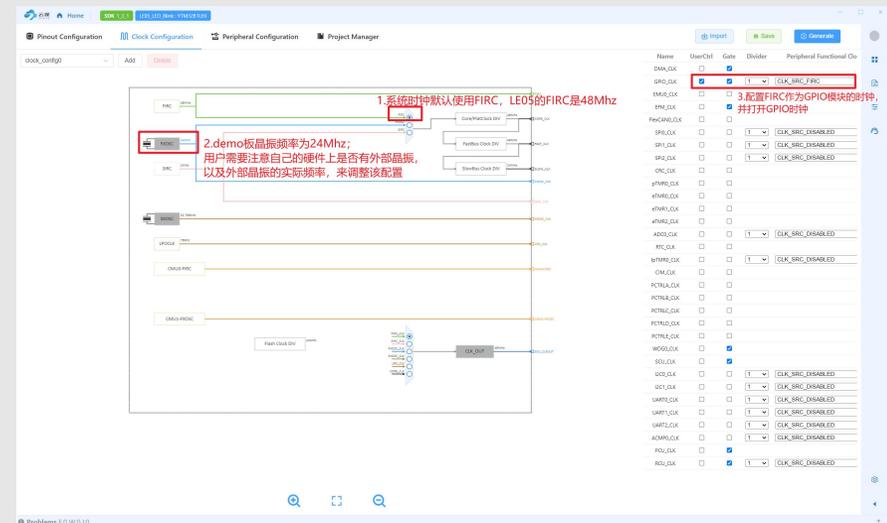
2. 创建工程名及存储路径，消除报错信息。



3. 配置引脚，配置PTD5为输出引脚。

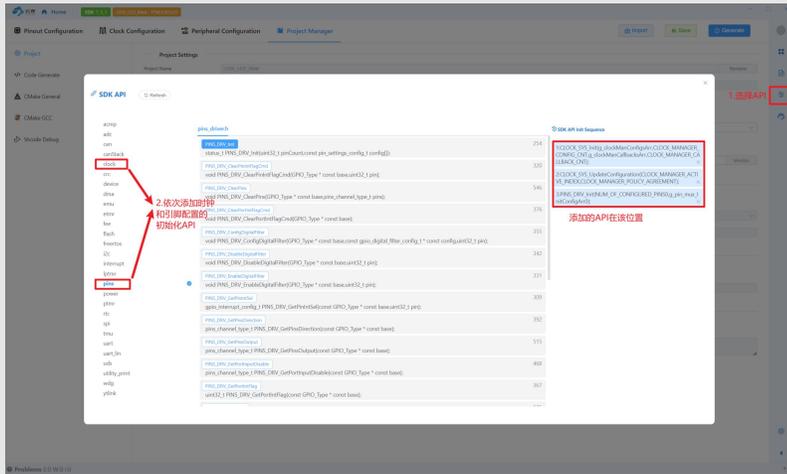


4. 配置系统时钟及外设时钟。

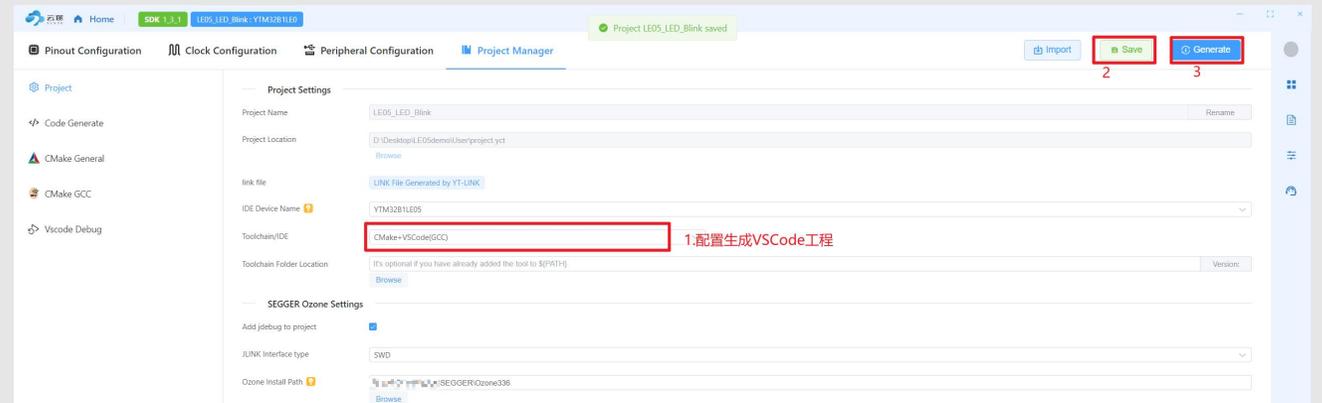


创建用户工程——创建VSCode工程并运行

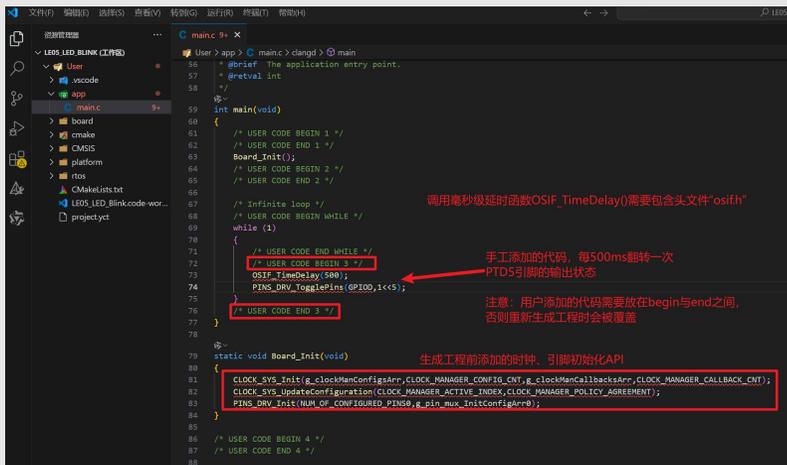
5. 添加时钟和引脚的初始化API。



6. 配置生成VSCode工程。



7. 编辑代码并编译工程。



8. 运行工程。

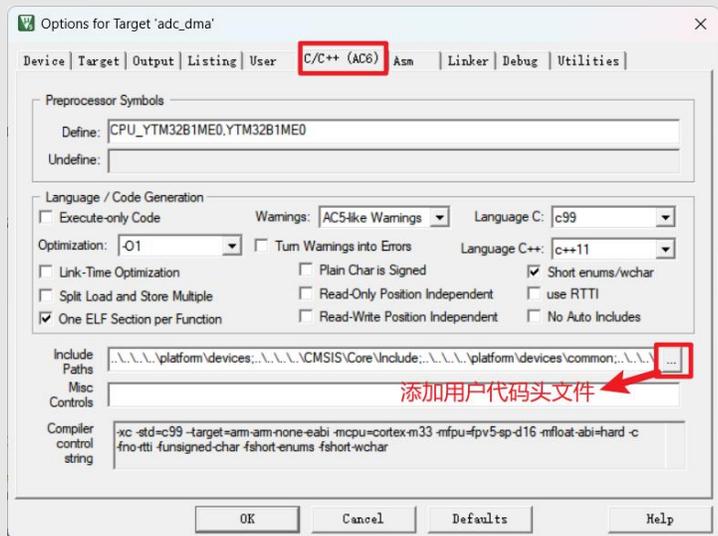
注意：下图中标注的J14一个跳帽需要短接。



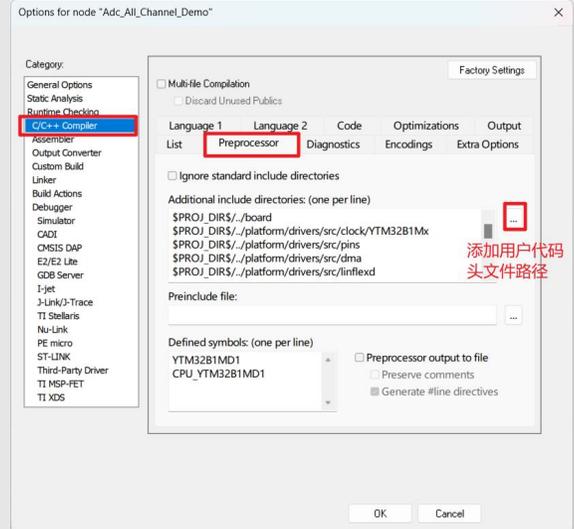
运行工程后，LED6（红灯）每500ms反转一次亮灭状态

创建用户工程——工程中添加用户代码

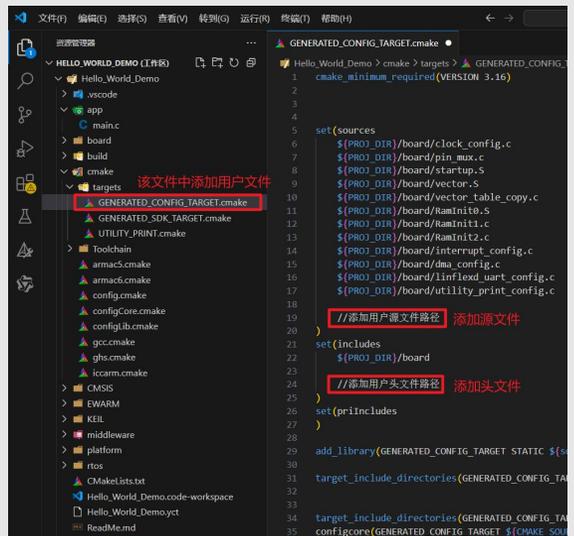
1. Keil工程添加用户文件



2. IAR工程添加用户文件



3. VSCode工程添加用户文件



06

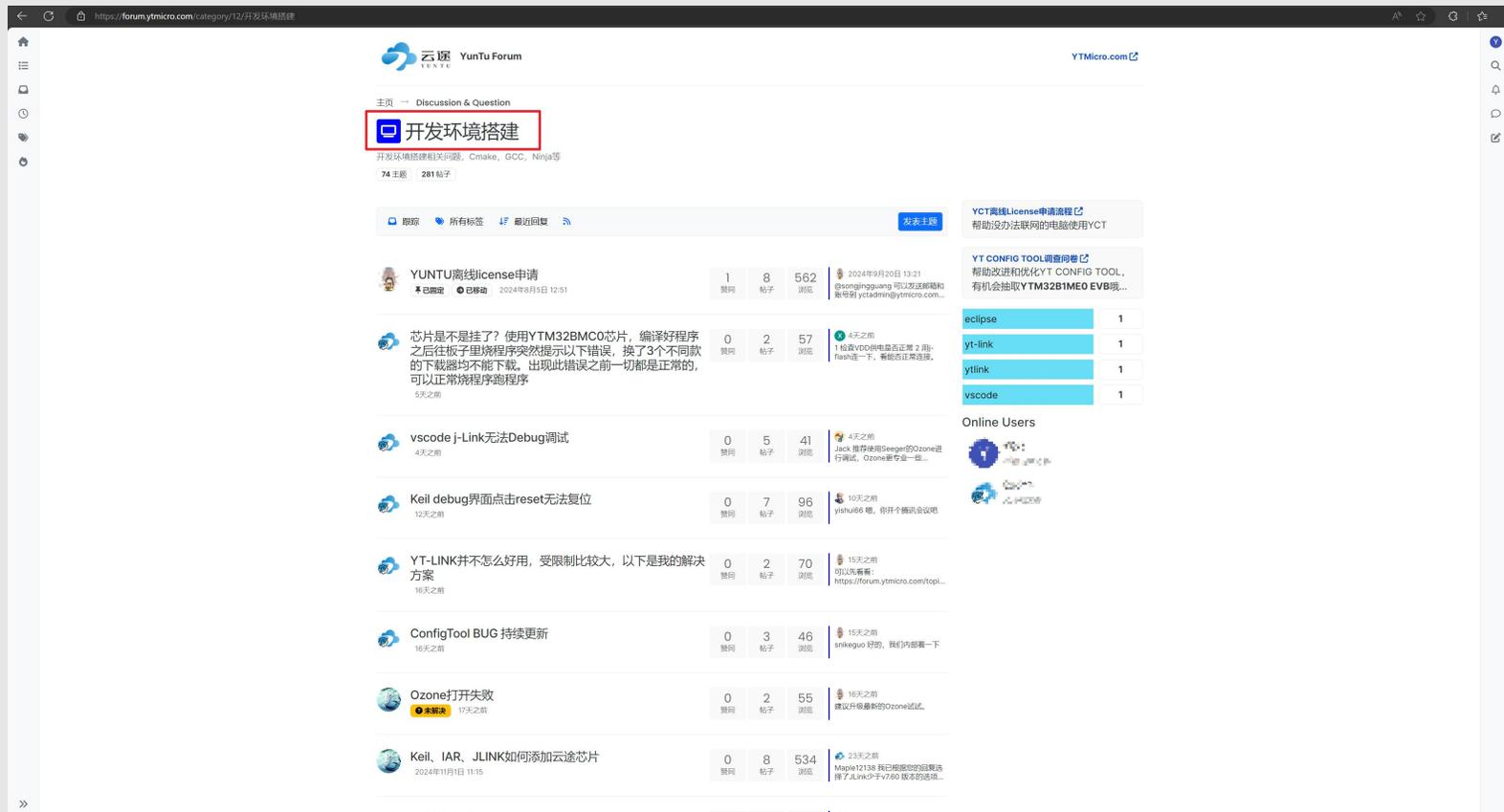
FAQ



FAQ

本文档会整理常见的问题，如遇本文档没有覆盖到的奇怪问题，欢迎在云途论坛“开发环境”板块进行讨论、寻找答案。

<https://forum.ytmicro.com/category/12/%E5%BC%80%E5%8F%91%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA>



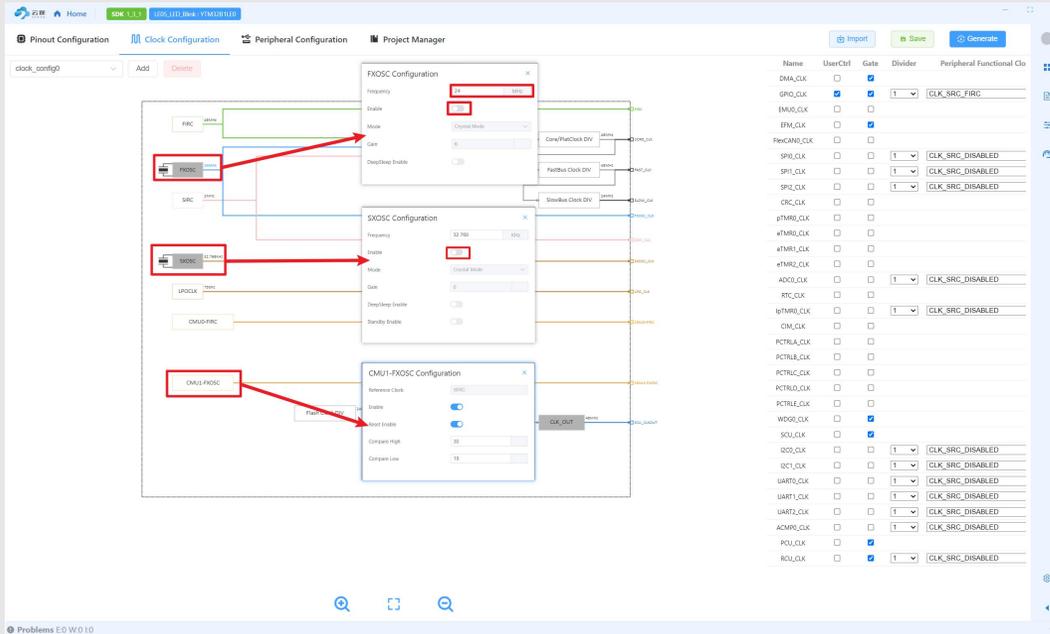
FAQ

Q: 生成工程后发现时钟初始化过不去，什么原因？

A: (1) 硬件电路无外部晶振FXOSC或者SXOSC，软件上就不要配置使能。

(2) 配置工具外部晶振默认是24M（与demo板一致），用户硬件有变化时需要调整。

(3) YCT中时钟监控（CMU）默认是开启的，例如监控外部晶振的上下限为30、18，是基于24M做了上下20%的偏移，晶振频率为其它值的用户需要调整。



FAQ

Q: Jlink连不上MCU，或者可以连接，但调试时状态异常，什么原因？

A: (1) 连不上：确认调试器正常前提下，排查是否将调试口SWDIO、SWDCLK配置为了其它外设功能。

(2) 调试异常：排查是否将调试口RESET引脚配置为了其它外设功能。

谢谢

THANK YOU