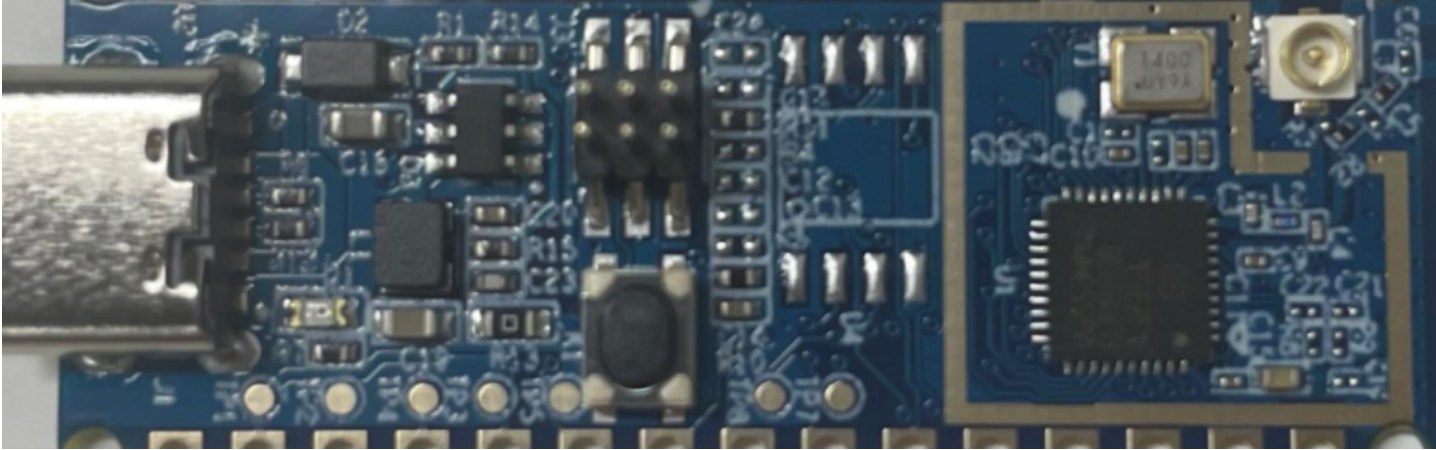


GD32VW553开发环境

萤火工场推出基于兆易创新（GigaDevice）GD32VW553是一款 RISC-V 内核的 32位 MCU 的开源硬件,小巧而且是最新的RISC-V架构。现在的国产芯片大量的开始发力自主可控的芯片架构。GD作为国内的老厂家相信大家也非常信任，愿意尝试这款GD32VW553。



GD32VW55x MCU 参数表

存储规格

参数	数值
FLASH	4096 KB
SRAM	320 KB

时钟系统

类型	数量
General Timer (16-bit)	2
General Timer (32-bit)	2
Advanced Timer (16-bit)	1
Basic Timer (16-bit)	1
SysTick (64-bit)	1
Watchdog	2
RTC	1

通信接口

接口类型	数量
USART	1
UART	2
I2C	2
SPI	1
QSPI	1
Wi-Fi	1
BLE5.2	1

模拟功能

参数	数值
ADC Units	1
ADC Channels	9

官方环境

可以看到板子虽然小，功能还是挺丰富的。而且拥有WiFi和蓝牙可玩性相当高。但是作为新芯片没有资料怎么版？好在官方给我们提供了文档来进行开发。

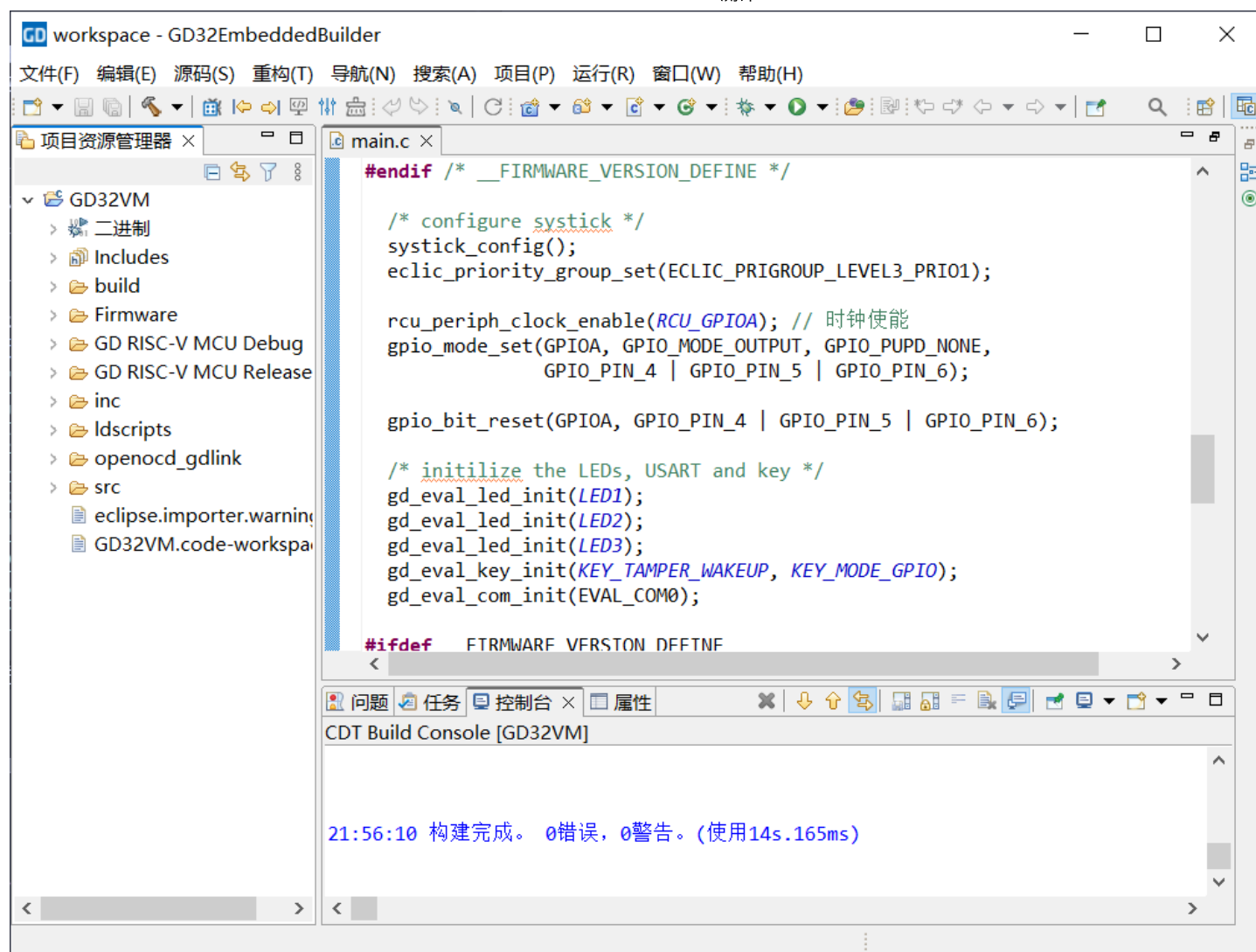
01_GPIO_Running_LED	2025/8/16 15:41	文件夹
02_GPIO_Key_Polling_mode	2025/8/16 15:41	文件夹
03_EXTI_Key_Interrupt_mode	2025/8/16 15:41	文件夹
04_USART_Printf	2025/8/16 15:41	文件夹
05_USART_Echo_Interrupt_mode	2025/8/16 15:41	文件夹
06_USART_DMA	2025/8/16 15:41	文件夹
07_ADC_conversion_triggered_by_ti...	2025/8/16 15:41	文件夹
08_I2C_EEPROM	2025/8/16 15:41	文件夹
09_SPI_LCD	2025/8/16 15:41	文件夹
10_TRNG_Get_Random	2025/8/16 15:41	文件夹
11_CAU	2025/8/16 15:41	文件夹
12_HAU	2025/8/16 15:41	文件夹
13_PKCAU_Modular_Addition_Interr...	2025/8/16 15:41	文件夹
14_RCU_Clock_Out	2025/8/16 15:41	文件夹
15_PMU_sleep_wakeup	2025/8/16 15:41	文件夹
16_RTC_Calendar	2025/8/16 15:41	文件夹
17_IFRP	2025/8/16 15:41	文件夹
18_TIMER_Breath_LED	2025/8/16 15:41	文件夹
19_QSPI_Flash	2025/8/16 15:41	文件夹

作为新鲜的芯片GD也是使用了自己最新的软硬件生态,不仅是不同于过去传统的新标准库,还有GD自己的IDE，类似现在STM的CubeIDE，不过我们的GD32VW553没有可视化的引脚配置，好在引脚也不多?。

应用软件 (共8个)				
文档名	版本	英文文档	中文文档	发布时间
GD32 All-In-One Programmer	4.2.7.27420		无	2024-12-13
Introduction: 基于芯片内部Bootloader的编程软件, 通讯接口支持UART和USB, 支持对MCU的Flash和OB进行编程/擦除/读取等基本操作。				
GD32 Embedded Builder	1.4.7.26843		无	2024-11-13
Introduction: GD32MCU的集成开发环境, 支持图形化配置、代码生成、工程编译和调试。				
GD32_ISP_Console(Linux)	4.0.5.26748		无	2024-11-12
Introduction: Linux系统下的ISP命令行工具, 用于串口和USB烧录GD MCU Flash数据。				
GigaDevice GD-Link Programmer	4.10.3.26725		无	2024-11-11
Introduction: GD-Link编程调试工具PC端软件。				
GD32VW55x Wi-Fi&BLE SDK	1.0.2		无	2024-07-19
Introduction: GD32VW553系列Wi-Fi & BLE SDK				
GD32VW55x Firmware Library	1.2.0		无	2024-01-18
Introduction: GD32VW55x标准固件库。包括程序、数据结构和宏定义, 覆盖所有集成外设的特征, 并包括了全部相关驱动和示例程序。				
GD32VW55x AddOn	1.1.0		无	2024-01-18
Introduction: 包含一个文件, 具体说明如下: 1. IAR_GD32VW55x_ADDON_1.1.0.exe IAR环境补丁, 支持IAR EW for RISC-V 3.10.1以上版本。 2. GD32VW55x - 1.00.emPackage SES环境补丁, 支持SEGGER Embedded Studio for RISC-V 7.32a及以上版本。				
SEGGER Embedded Studio IDE	7.32a			2023-12-11
Introduction: GD32V系列RISC-V MCU的用户可下载免费商用的SEGGER Embedded Studio多平台集成开发环境(IDE), 请访问SEGGER官网 wiki.segger.com/GD32V 页面下载软件并注册激活。				

CSDN @今日待办

下载好里面的资料就可以直接开始使用了.没错!!! 这个IDE居然是即开即用, 比罐头都方便?. 因为是基于Eclipse二次开发的, IDE本身是英文的, 但是我们可以通过[下载插件来进行汉化](#)。选择好芯片创建好工程,就可以正常的进行编码了, 用它的模板编译一下。



非常完美？

基于对代码的分析，接下来为大家介绍这个由 GD32EmbeddedBuilder 自动生成的软件框架：

GD32EmbeddedBuilder 软件框架介绍

1. 框架概述

这是一个基于**GD32VW55x 系列 RISC-V MCU**的嵌入式软件开发框架，由 GD32EmbeddedBuilder 工具自动生成。该框架采用分层架构设计，提供了完整的硬件抽象层和标准外设库。

2. 核心架构特点

硬件平台

- **MCU:** GD32VW55x 系列（RISC-V 架构）
- **内存配置:**
 - Flash: 4MB (0x08000000)
 - RAM: 288KB (0x20000000)

- 架构: RISC-V 32 位处理器

分层设计

应用层 (Application)

↓

硬件抽象层 (HAL)

↓

标准外设库 (Standard Peripheral Library)

↓

硬件层 (Hardware)

3. 主要组件分析

标准外设库 (Firmware/GD32VW55x_standard_peripheral/)

- **GPIO**: 通用输入输出控制
- **USART**: 串口通信
- **Timer**: 定时器功能
- **ADC**: 模数转换
- **SPI/I2C**: 串行通信接口
- **DMA**: 直接内存访问
- **加密模块**: CAU (Cryptographic Acceleration Unit)
- **哈希模块**: HAU (Hash Acceleration Unit)

RISC-V 核心支持 (Firmware/RISCV/)

- **启动代码**: start.S, entry.S
- **中断处理**: handlers.c
- **系统初始化**: init.c
- **标准库支持**: stubs/ 目录提供系统调用接口

评估板支持 (src/gd32vw553h_eval.c)

- **LED 控制**: 3 个 LED 灯 (LED1, LED2, LED3)
- **按键输入**: TAMPER_WAKEUP 按键
- **串口通信**: USART0 配置
- **GPIO 配置**: 引脚定义和时钟配置

4. 开发工具链

编译环境

- 编译器: RISC-V GCC 工具链
- 构建系统: Makefile + [sources.mk](#)
- 调试支持: OpenOCD + GDLINK

项目结构

```
GD32VM/
├─ src/           # 用户源代码
├─ inc/           # 头文件
├─ Firmware/     # 标准外设库
├─ RISCv/         # RISC-V核心支持
├─ ldscripts/     # 链接脚本
└─ GD RISC-V MCU Debug/ # 调试版本输出
```

5. 示例应用功能

从 main.c 可以看出，这是一个**LED 闪烁 + 串口打印 + 按键检测**的示例程序：

- **LED 控制**: LED2 和 LED3 每 500ms 闪烁一次
- **串口输出**: 打印系统时钟频率信息
- **按键检测**: 检测 TAMPER_WAKEUP 按键，按下时切换 LED1 状态
- **系统定时**: 使用 SysTick 提供精确的时间基准

6. 框架优势

1. **标准化**: 遵循 ARM Cortex-M 类似的 API 设计模式
2. **完整性**: 提供完整的外设驱动和硬件抽象
3. **易用性**: 自动生成的代码结构清晰，易于理解和修改
4. **兼容性**: 支持多种 GD32VW55x 系列芯片
5. **开发效率**: 减少底层硬件配置的重复工作

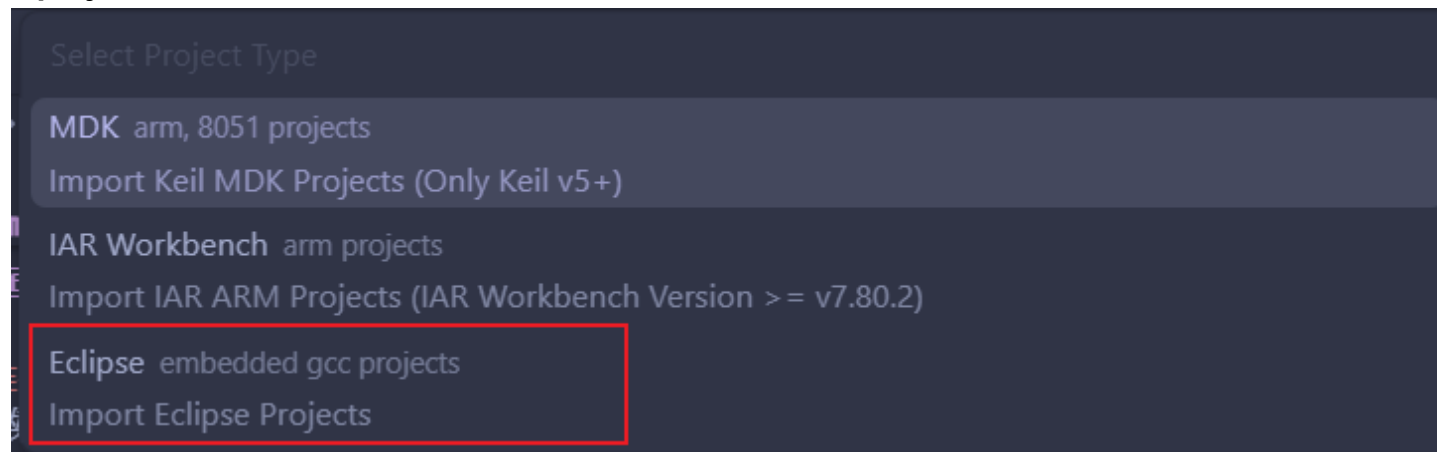
7. 适用场景

- 物联网设备开发
- 工业控制应用
- 消费电子产品
- 学习和教学项目
- 原型快速开发

这个框架为开发者提供了一个完整的嵌入式开发环境，大大简化了 RISC-V MCU 的开发流程，特别适合需要快速原型开发和产品化的项目。

通过EIDE插件在VsCode下开发

如果不喜欢这种界面,也可以通过VsCode的EIDE插件来进行项目的导入,按照提示导入项目中的 **.cproject** 文件，我们可以自动创建一个VsCode项目。



但是会在 **eclipse.importer.warning.txt** 这个文件中会记录一些不适配的参数。直接编译会报错,原因就在这个文件中。比如图中的这个报错就是我们的链接文件有问题。只要参考一下

```
riscv-none-embed-gcc.exe: error: GD32VW55X.lds: No such file or directory  
riscv-none-embed-gcc.exe: error: GD32VW55X.lds: No such file or directory
```

Incompatible Args:

/:

globalArgs:

- <undefined> = undefined
- <undefined> = com.gigadevice.mbs.option.toolSettings.debugging.level.max
- <undefined> = -g3
- <undefined> = ./Tools/nuclei_riscv_newlibc_prebuilt_win32_2022.04/gcc/bin
- <undefined> = ./Tools/Build Tools/2.10-20180103-1919/bin
- <undefined> = riscv-nuclei-elf-
- <undefined> = riscv-nuclei-elf-objcopy
- <undefined> = riscv-nuclei-elf-objdump
- <undefined> = riscv-nuclei-elf-size
- <undefined> = MCU RISC-V GCC
- <undefined> = riscv-nuclei-elf-gcc
- <undefined> = GD32VW553HMQ6
- <undefined> = -march=rv32imac -mabi=ilp32 -mcmmodel=medlow -msmall-data-limit=8

cIncDirs: []

cMacros: []

cCompilerArgs: []

sIncDirs: []

sMacros: []

assemblerArgs:

- <undefined> = undefined

linkerArgs:

- <undefined> = undefined

linkerLibArgs: []

其中提示我们没有进行链接器的设置

```
= -march=rv32imac -mabi=ilp32 -mcmmodel=medlow -msmall-data-limit=8 -mdiv
```

根据报错进行配置

enable

▼

输出调试信息

指令集
(arch)

rv32imac

调用约定
(abi)

ilp32

代码模型
(code-model)

medlow

编译器附加选项（全局）

```
-O0 -fmessage-length=0 -fsigned-char -ffunction-sections -fdata-sections -march=rv32imac -mabi=ilp32 -mcmodel=medlow -msmall-data-limit=8 -mdiv
```

再进行链接脚本的路径配置



就可以通过EIDE编译文件了

