

第四章 STM32CubeMX软件的使用

主讲人：漆强

电子科技大学

ytqiqiang@163.com

本章内容



HAL库的基本特点



STM32CubeMX软件的使用步骤



利用MDK软件编写应用程序

教学目标



了解HAL库的基本特点



熟练掌握STM32CubeMX软件的使用流程



熟练掌握MDK软件的程序编写和程序下载方法



电子科技大学
University of Electronic Science and Technology of China

4.1 HAL库的基本特点

MCU固件包

HAL库属于MCU
固件包的一部分



Windows

macOS®



Initialization code

Embedded software for STM32

Examples and demos

Middleware components

Hardware abstraction layer

1. 例程
2. 中间件
3. HAL库

F0

G0

F1

F3

G4

F2

F4

MP1*

F7

H7

L0

L1

L4

L4+

WB



STM32微控制器各个产品系列都有独立的MCU固件包

HAL库路径

查找HAL库存放路径的方法：

- ① 启动CubeMX软件
- ② Help -> Updater Settings



MCU固件包文件结构

STM32Cube_FW_F4_1.24.2

Documents

Drivers

Middleware

Projects

Utilities

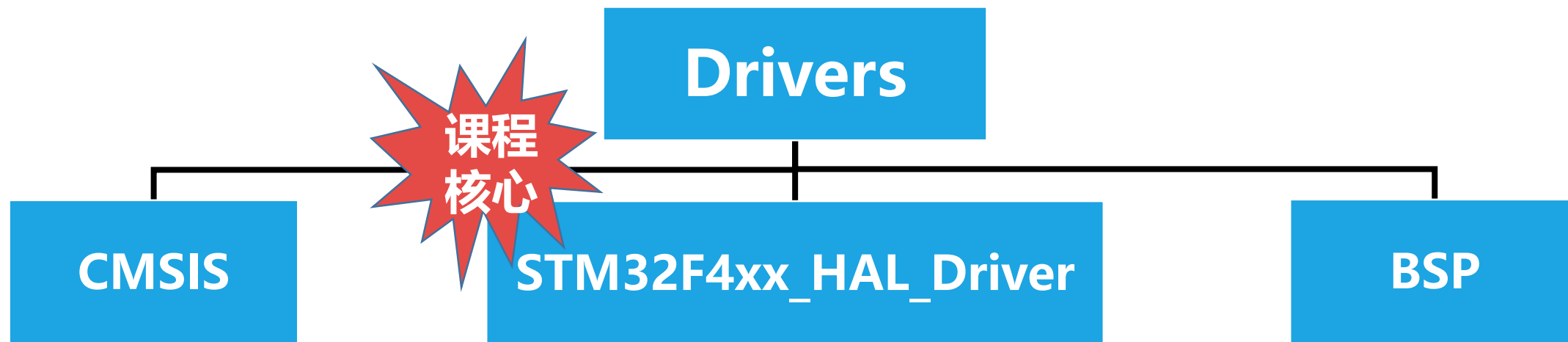


关于MCU固件包和HAL
库使用的官方文档

各类支撑文件，
如字体文件和图
形应用例程中使
用的图片文件等



Drivers文件夹

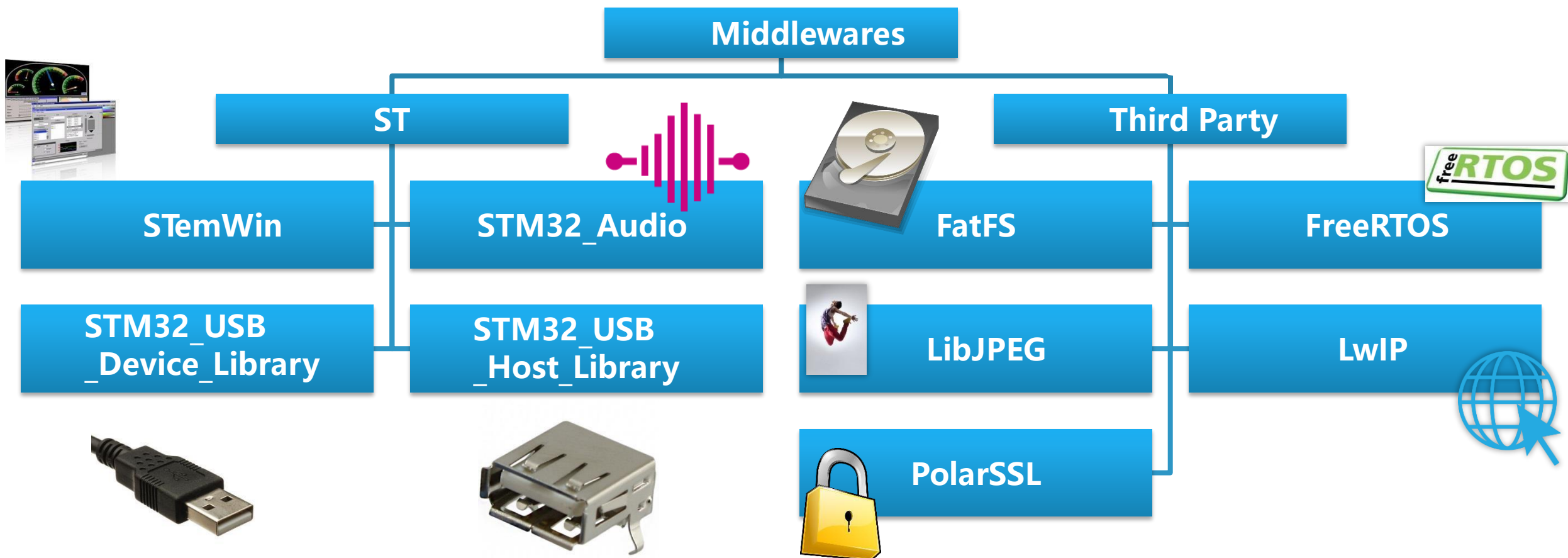


由ARM公司提供的Cortex微控制器软件接口标准, 包括Cortex内核寄存器定义、启动文件等

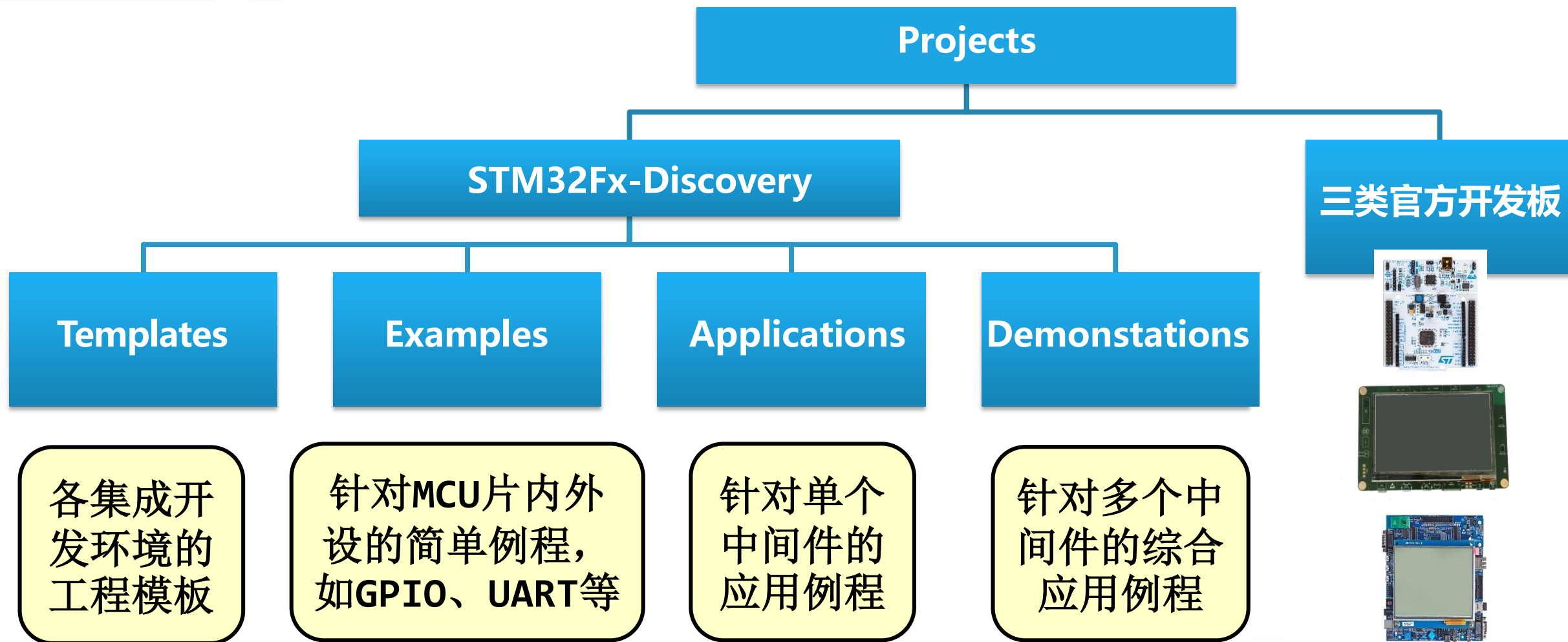
STM32微控制器片内外设的HAL库驱动文件
stm32f4xx_hal_ppp.h
stm32f4xx_hal_ppp.c

基于HAL库开发的官方开发板的板级支持包, 提供指示灯、按键等外围电路的驱动程序

Middleware文件夹





Projects文件夹



三种库比较

标准库、HAL库和LL库的对比

ST免费提供		可移植性	存储空间占用	执行效率	易用性	完备性	硬件覆盖范围
 标准外设库		++	++	++	+	++	+++
 STM32Cube	HAL	+++	+	+	++	+++	+++++
	LL	+	+++	+++	+	++	++

UM1718: STM32CubeMX软件使用参考手册

UM1725: STM32Cube 硬件抽象层驱动使用参考手册



电子科技大学
University of Electronic Science and Technology of China

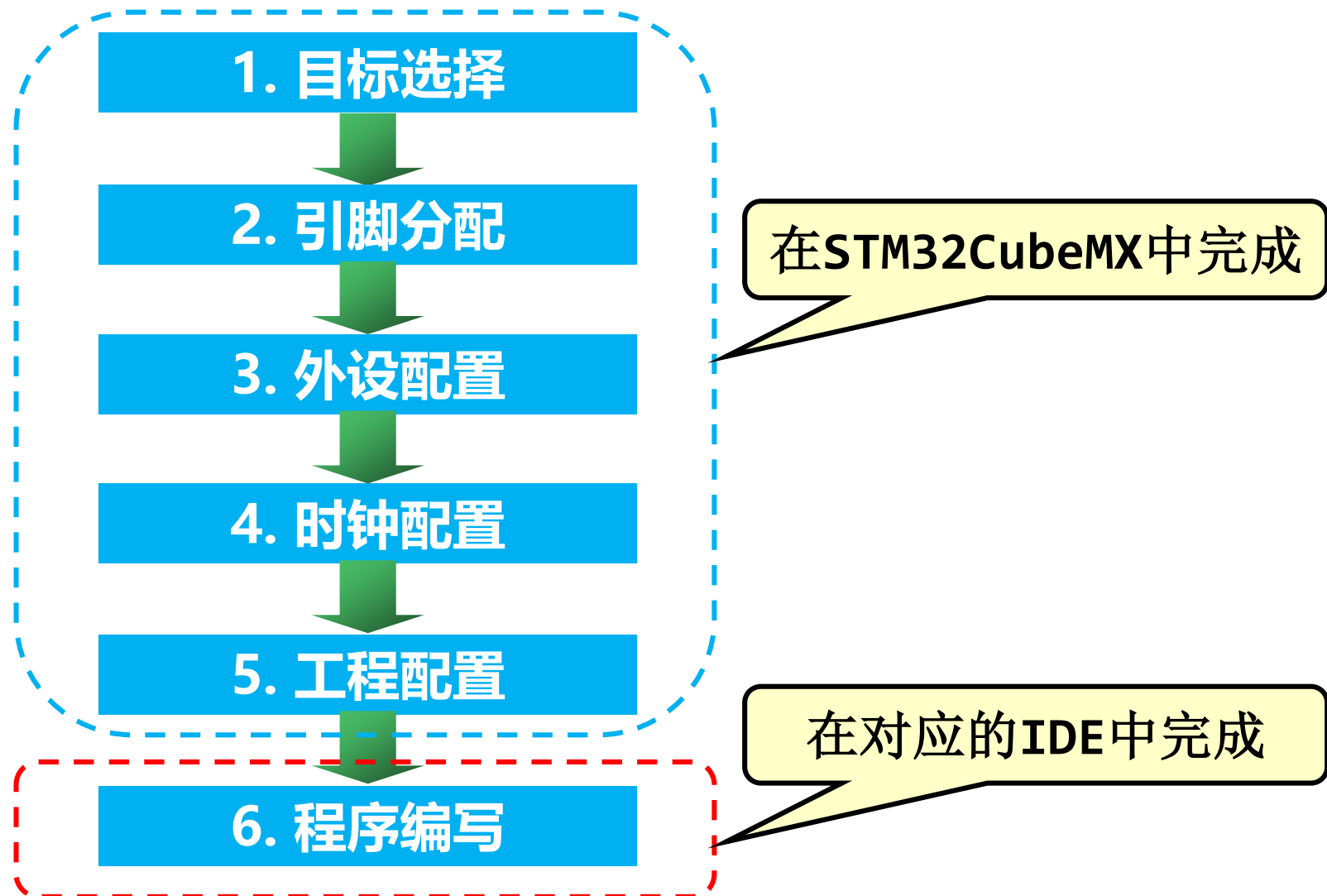
4.2 STM32CubeMX使用步骤

基本特点



- 1 集成MCU Finder功能，便于用户进行芯片选型
- 2 图形化方式配置时钟、片内外设，自动生成初始化代码
- 3 自动生成工程框架，支持MDK、EWARM等多种开发环境
- 4 提高开发效率，聚焦业务层的设计

使用步骤



演示例程：驱动指示灯

1 例程目标

掌握STM32CubeMX软件使用步骤

2 例程内容

控制Nucleo开发板上的用户指示灯LD2每隔1s闪烁

建议

建议用户在桌面上新建一个名为CubeMX的文件夹，用于存放本课程的例程代码。

对于STM32微控制器的外设学习，可以再建立子文件夹：按照外设名建立，如GPIO、EXTI、TIMER、UART等。

注意：CubeMX建立的工程必须存放在英文路径下。



电子科技大学
University of Electronic Science and Technology of China

1 目标选择

1

File

Window

Help

菜单栏

启动界面

软件管理

已存在工程

新建工程

Existing Projects

2

Recent Opened Projects

UART_DMA.ioc

Last modified date : 14/10/2019 23:5

UART_IDLE_INT.ioc

Last modified date : 14/10/2019 11:22:36

COM_Protocol.ioc

Last modified date : 13/10/2019 23:40:44

UART_INT.ioc

Last modified date : 11/10/2019 10:3

UART_Polling.ioc

Last modified date : 13/10/2019 17:58:59

Other Projects

基于芯片

基于开发板

交叉选型

New Project

3

I need to :

Start My project from MCU

ACCESS TO MCU SELECTOR

Start My project from STBoard

ACCESS TO BOARD SELECTOR

Start My project from Cross Sel...

ACCESS TO CROSS SELECTOR

选择基于芯片方式新建工程

Manage software installations

4

Check for STM32CubeMX and embedded soft...

CHECK FOR UPDATES

Install or remove embedded software packages

INSTALL / REMOVE

安装/卸载
固件包检查软
件更新

标签页切换

1

芯片/开发板选择界面

MCU/MPU Filters

Part Number Search



2

Core

Series

筛选器

Check/Uncheck All

☐ STM32F0☐ STM32F1☐ STM32F2☐ STM32F3☐ STM32F4☐ STM32F7☐ STM32G0☐ STM32G4☐ STM32H7☐ STM32L0☐ STM32L1☐ STM32L4

Features

Block Diagram

Docs & Resources



Datasheet



Buy



Start Project

3

芯片文档

MCUs/MPUs List: 1517 items

+ Display similar

4

芯片/开发板列表框

*	Part No	Reference	Marketing ...	Unit Price for 1 ...	Board	Package	Flash	RAM	IO	Freq.	GFX Sc...
☆	STM32F030C6	STM32F...	Active	0.542		LQFP48	32 kByt...	4 kBytes	39	48 MHz	0.0
☆	STM32F030C8	STM32F...	Active	0.657		LQFP48	64 kByt...	8 kBytes	39	48 MHz	0.0
☆	STM32F030CC	STM32F...	Active	1.0		LQFP48	256 kB...	32 kByt...	37	48 MHz	0.0
☆	STM32F030F4	STM32F...	Active	0.385		TSSOP...	16 kByt...	4 kBytes	15	48 MHz	0.0
☆	STM32F030K6	STM32F...	Active	0.471		LQFP32	32 kByt...	4 kBytes	25	48 MHz	0.0
☆	STM32F030R8	STM32F...	Active	0.685	NUCLEO-F030R8 32F0308DISCOV...	LQFP64	64 kByt...	8 kBytes	55	48 MHz	0.0
☆	STM32F030RC	STM32F...	Active	1.1		LQFP64	256 kB...	32 kByt...	51	48 MHz	0.0
☆	STM32F031C4	STM32F...	Active	0.97		LQFP48	16 kByt...	4 kBytes	39	48 MHz	0.0
☆	STM32F031C6	STM32F...	Active	1.013		LQFP48	32 kByt...	4 kBytes	39	48 MHz	0.0

目标选择

MCU/MPU Filters

★ 🔍 📁 ↺

Part Number Search

1 🔍 STM32F411RE ▼

输入芯片型号



MCUs/MPUs List: 1 item

2 Display similar items

双击

*	Part No	Reference	Marketing ...	Unit Price for 1 ...	Board	Package	Flash	RAM	IO	Freq.	GFX Sc...
★	STM32F411RE	STM32F...	Active	3.155	NUCLEO-F411RE	LQFP64	512 kB...	128 kB...	50	100 ...	0.0

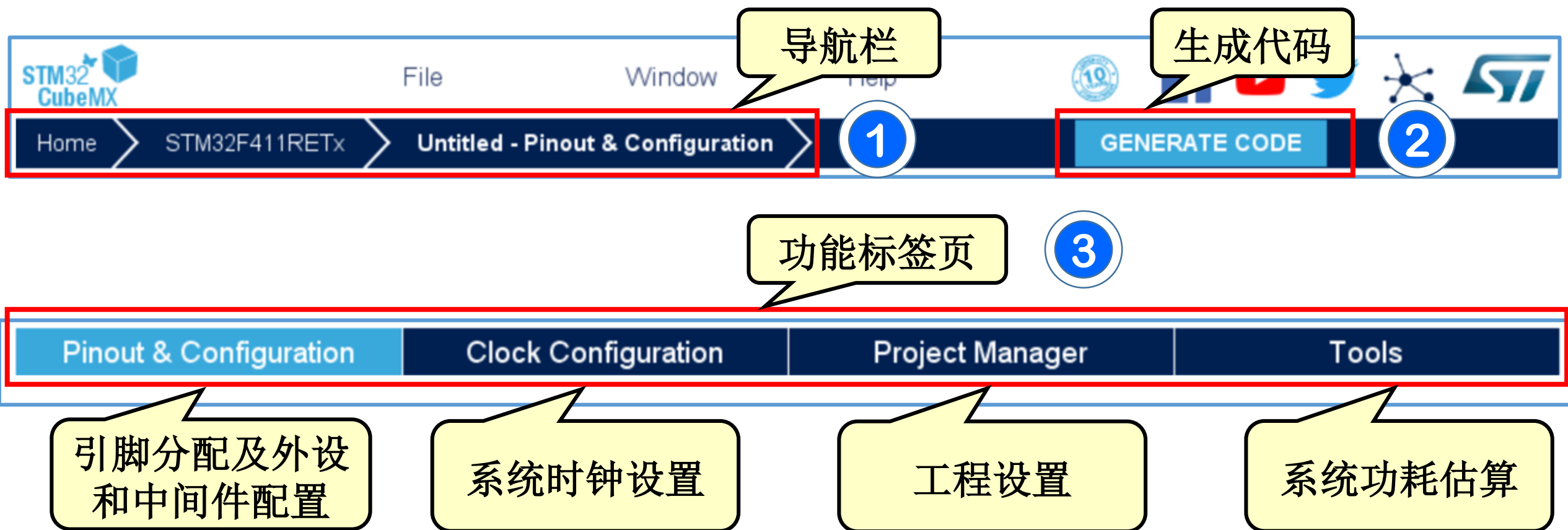


电子科技大学
University of Electronic Science and Technology of China

2 引脚分配

配置主界面

配置主界面完成CubeMX软件的所有配置操作





Categories

A->Z

System Core >

Analog

1

Timers >

Connectivity >

Multimedia >

Computing >

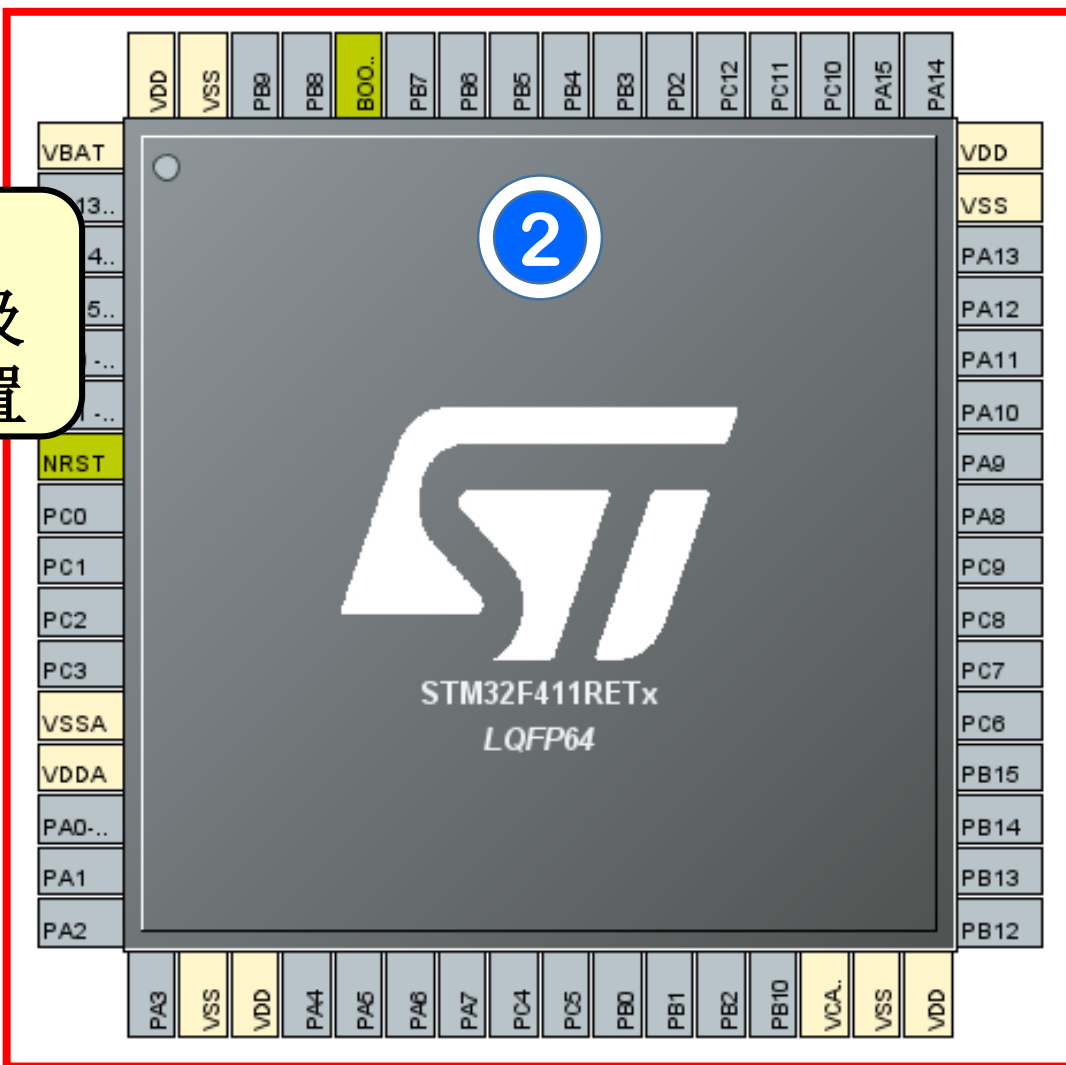
Middleware >

类别栏
用于外设及
中间件配置

放大/缩小/
旋转引脚图

Pinout view

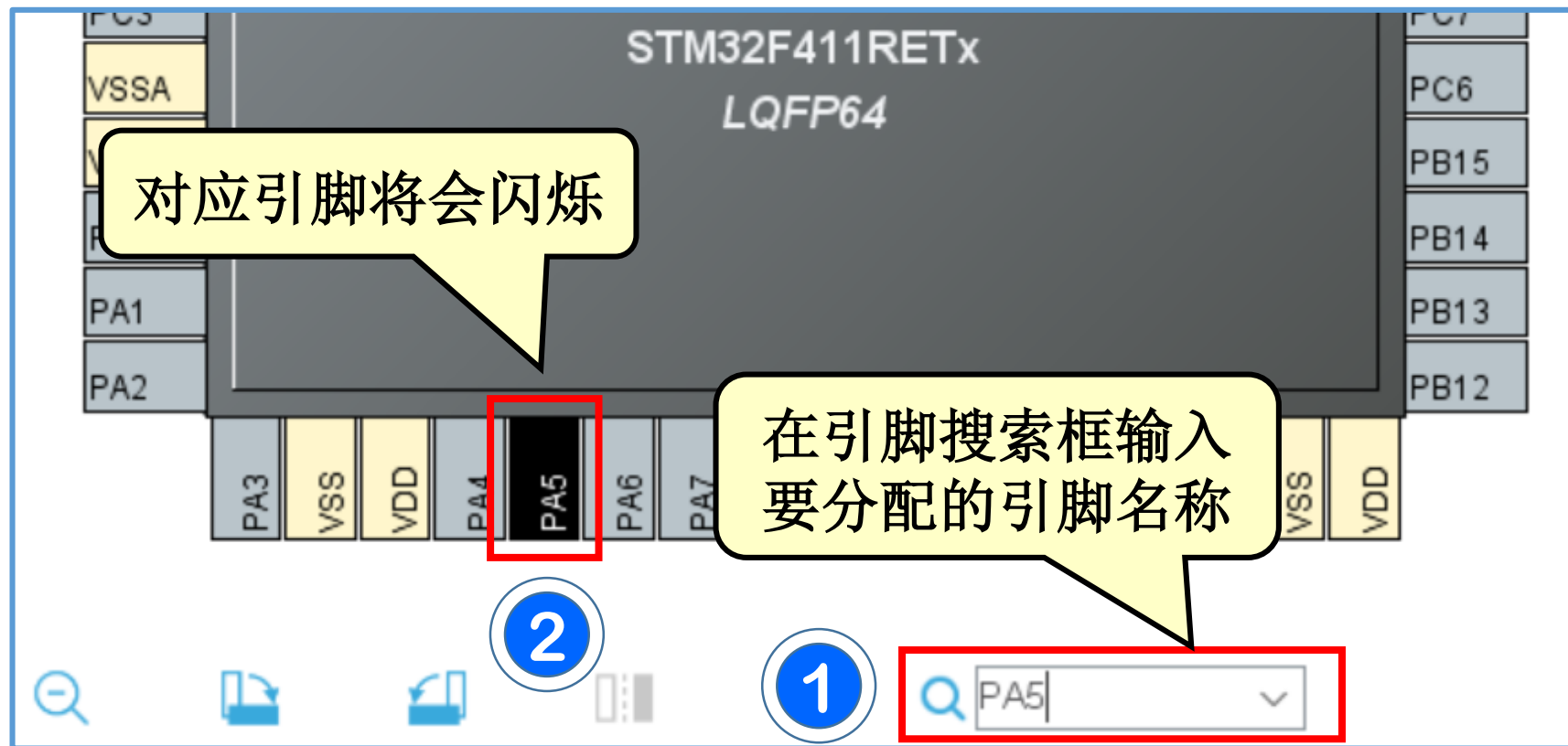
System view



芯片引脚图
用于引脚分配



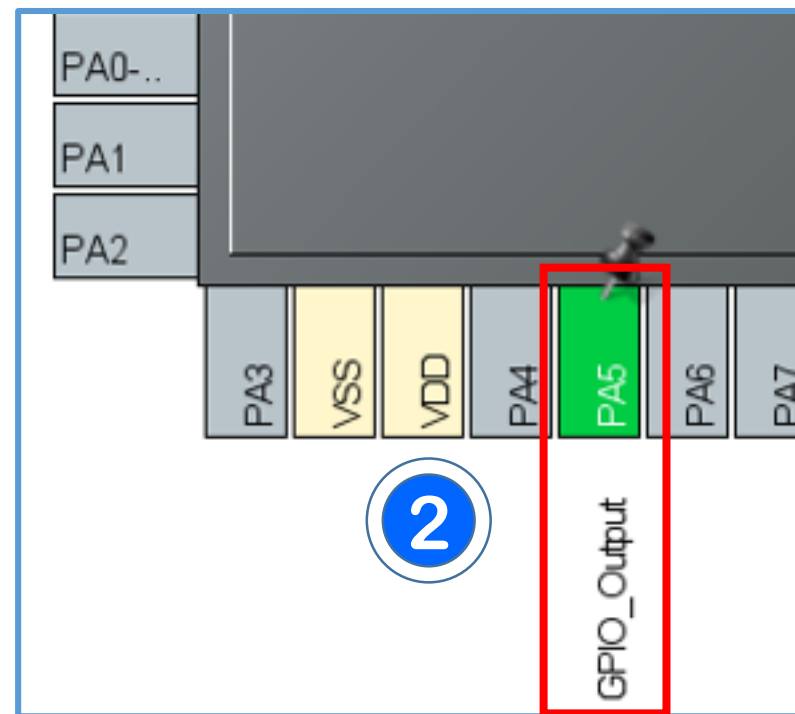
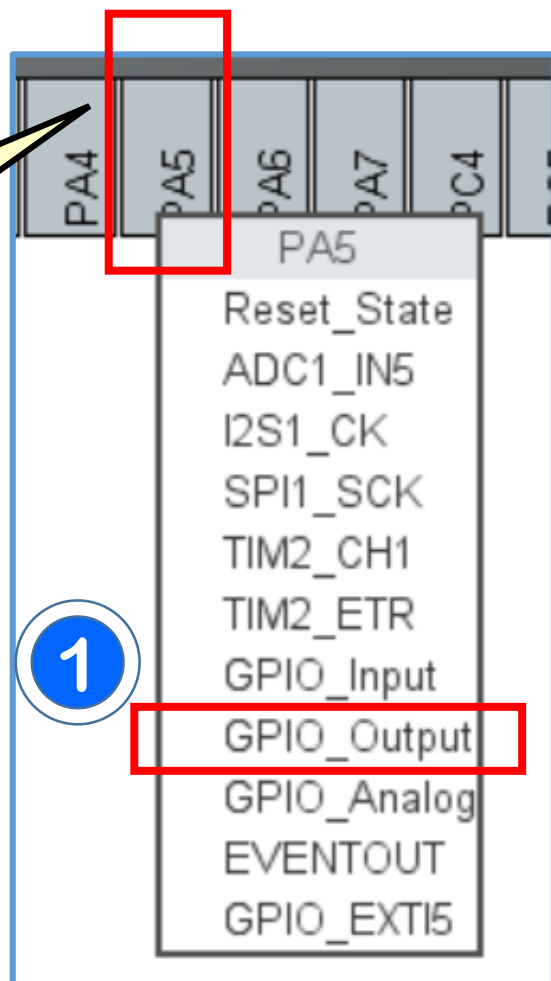
查找引脚



选择引脚功能

Nucleo开发板上利用PA5引脚作为通用I/O控制指示灯LD2

单击PA5





电子科技大学
University of Electronic Science and Technology of China

3 外设配置

Pinout & Configuration

Categories

A->Z

System Core



Analog



Timers



Connectivity



Multimedia



Computing



Middleware



- ① **System Core**: 用于配置GPIO、时钟源、中断系统以及系统相关的外设
- ② **Analog**: 用于配置模/数和数/模转换外设
- ③ **Timers**: 用于配置定时器和实时时钟外设
- ④ **Connectivity**: 用于配置I2C、SPI和UART等连接外设
- ⑤ **Multimedia**: 用于配置I2S等音频数据传输外设
- ⑥ **Computing**: 用于配置CRC校验外设
- ⑦ **Middleware**: 用于配置RTOS和GUI等中间件

配置时钟模式

RCC_OSC_IN	PH0 - ..
RCC_OSC_OUT	PH1 - ..

展开或收起
配置窗口

The screenshot shows the STM32CubeMX interface. On the left, the 'System Core' category is selected, and the 'RCC' component is highlighted. The main panel displays the 'RCC Mode and Configuration' settings. The 'High Speed Clock (HSE)' is set to 'Disable', and the 'Low Speed Clock (LSE)' is also set to 'Disable'. The 'BYPASS Clock Source' is selected, and the 'Crystal/Ceramic Resonator' is listed as an option. A red box highlights the 'BYPASS Clock Source' option. A yellow callout box explains the clock modes: 'Disable: 内部时钟', 'BYPASS: 旁路时钟', and 'Crystal: 晶振/陶瓷振荡器'. A red box highlights the expand/collapse icon in the top right corner of the configuration window.

RCC Mode and Configuration

Mode

High Speed Clock (HSE) Disable

Low Speed Clock (LSE) Disable

☐ Master Clock Output

☐ Master Clock Output

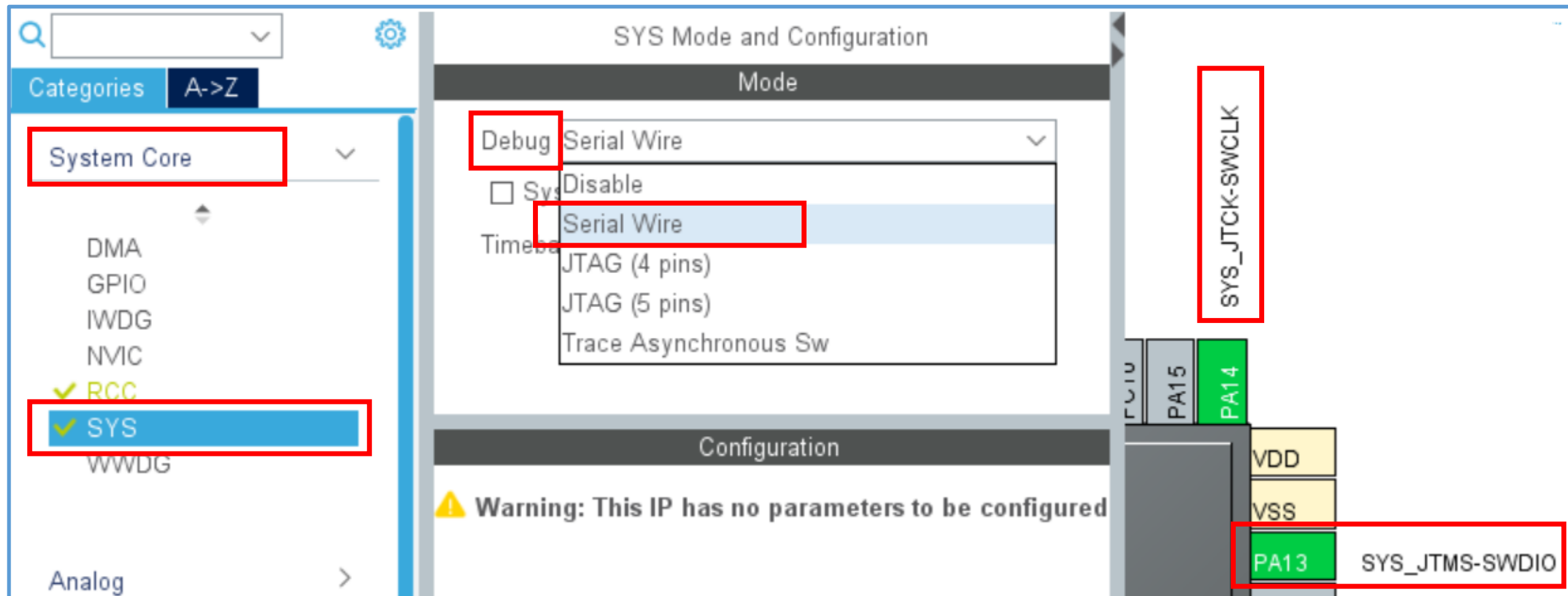
☐ Audio Clock Input (I2S)

BYPASS Clock Source

Crystal/Ceramic Resonator

Disable: 内部时钟
BYPASS: 旁路时钟
Crystal: 晶振/陶瓷振荡器

配置调试接口



配置GPIO

STM32CubeMX GPIO Configuration Interface

Categories: A->Z

System Core

- DMA
- GPIO**
- IWDG
- NVIC
- ✓ RCC
- ✓ SYS
- WWDG

GPIO Mode and Configuration

Configuration

☐ Group By Peripherals

☒ GPIO | ☒ RCC | ☒ SYS

Search Signals

☐ Show only Modified Pins

Pin...	Signal ...	GPIO ...	GPIO ...	GPIO ...	Maxim...	User L...	Modified
PA5	n/a	Low	Output...	No pull...	Low		<input type="checkbox"/>

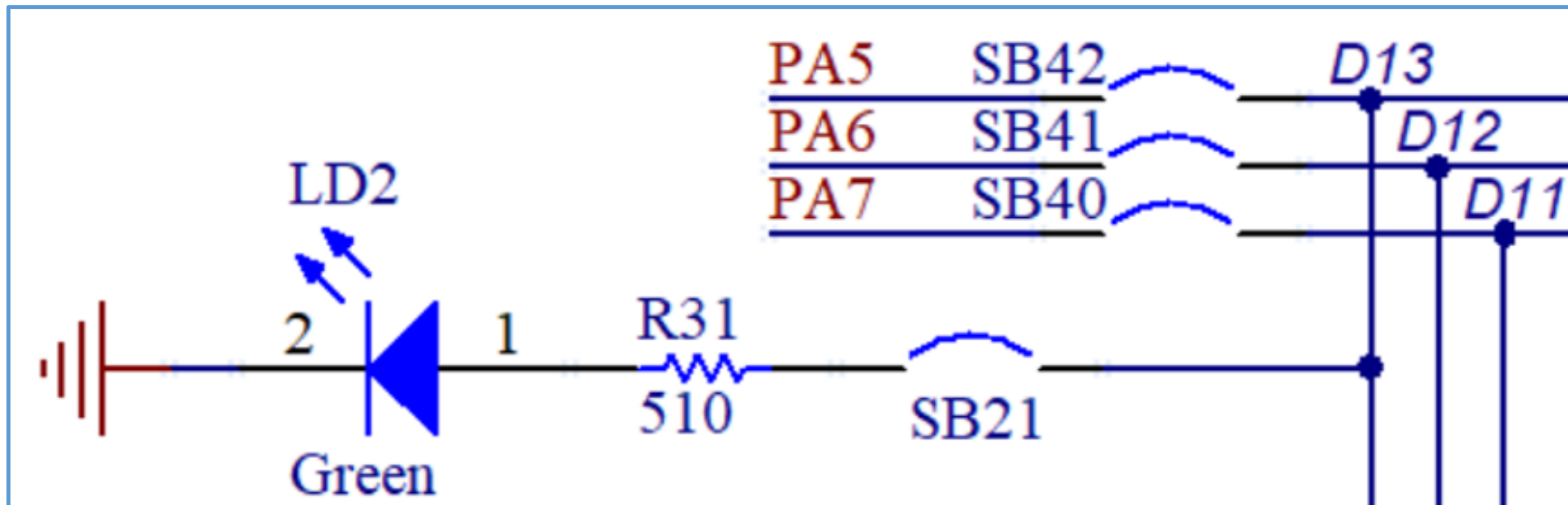
单击

指示灯电路

高电平驱动方式

PA5输出高电平，指示灯LD2开启

PA5输出低电平，指示灯LD2关闭



配置GPIO

配置GPIO参数

PA5 Configuration :

GPIO output level

Low

初始电平

GPIO mode

Output Push Pull

引脚模式

GPIO Pull-up/Pull-down

No pull-up and no pull-down

上/下拉电阻

Maximum output speed

Low

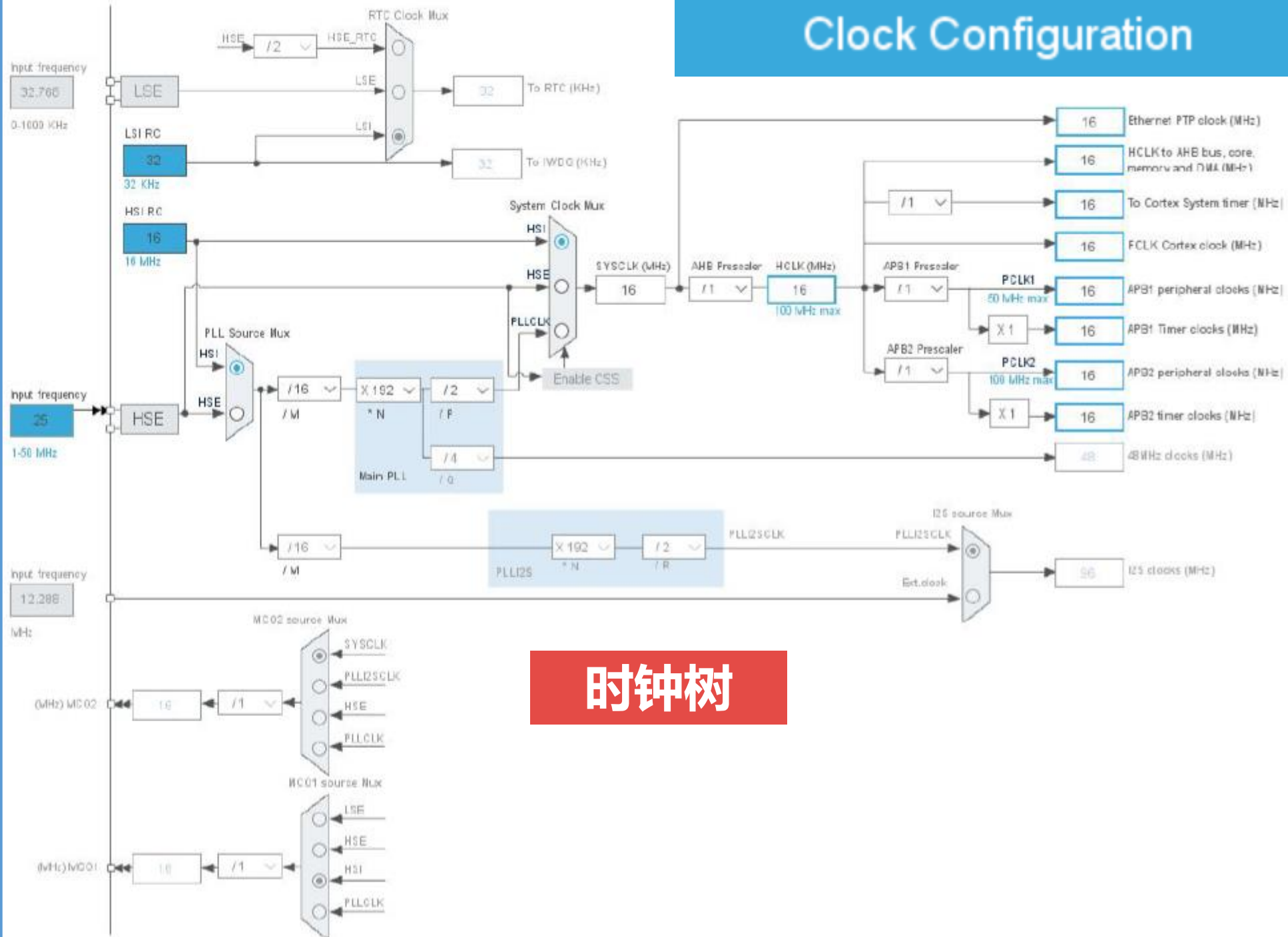
引脚速度

User Label

引脚名称

4 时钟配置

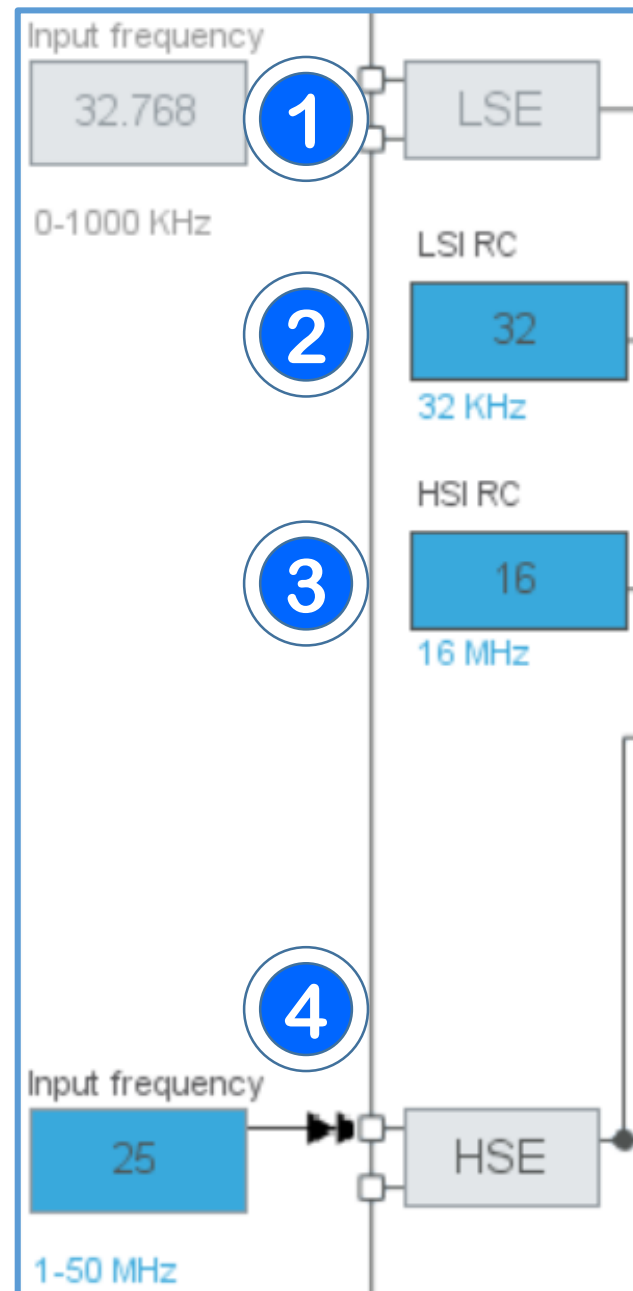
Clock Configuration



时钟树

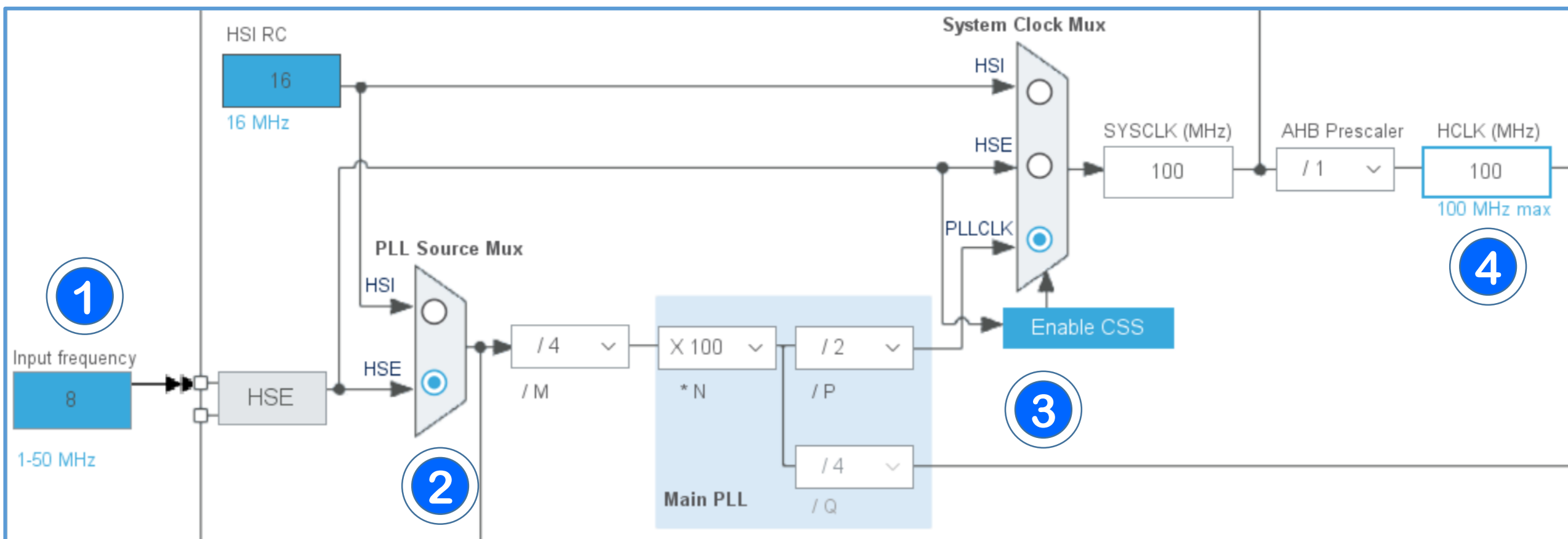
时钟源模块

- ① 外部低速时钟LSE: 32.768KHz
- ② 内部低速时钟LSI : 32KHz
- ③ 内部高速时钟HSI : 16MHz
- ④ 外部高速时钟HSE:
 - 震荡模式: 4MHz ~ 26MHz晶振
 - 旁路模式: 1MHz ~ 50MHz信号



时钟配置步骤

- ① 修改时钟源频率 : 25 → 8
- ② 选择锁相环输入时钟 : HSE
- ③ 选择系统时钟源 : PLLCLK
- ④ 设置HCLK时钟频率 : 100, 并回车





电子科技大学
University of Electronic Science and Technology of China

5 工程配置

配置模块

Project Manager

Project

工程管理

Code Generator

代码生成

Advanced Settings

高级设置

工程管理

Project Settings

Project Name

Demo

工程名称

1

Project Location

C:\Users\qiqiang\Desktop\CubeMX

工程路径

2

Application Structure

Basic

完成工程名称和工程路径设置后，将自动在CubeMX文件夹下新建一个名为Demo的文件夹

Toolchain Folder Location

C:\Users\qiqiang\Desktop\CubeMX\Demo\

Toolchain / IDE

MDK-ARM

Min Version

V5

选择IDE

3

工程管理

Linker Settings

Minimum Heap Size

Minimum Stack Size

Mcu and Firmware Package

Mcu Reference

Firmware Package Name and Version

☒ Use latest available version

☒ Use Default Firmware Location

[Browse](#)

代码生成设置

STM32Cube MCU packages and embedded software packs

- ☒ Copy all used libraries into the project folder
- ☐ Copy only the necessary library files
- ☐ Add necessary library files as reference in the toolchain project configu..

- 复制所有库
- 复制必要库
- 引用参考

Generated files

- ☐ Generate peripheral initialization as a pair of '.c/.h' files per peripheral
- ☐ Backup previously generated files when re-generating
- ☒ Keep User Code when re-generating
- ☒ Delete previously generated files when not re-generated

片内外设初始化
采用独立文件

HAL Settings

- ☐ Set all free pins as analog (to optimize the power consumption)
- ☐ Enable Full Assert

- 设置模拟功能
- 使能断言语句

Template Settings

Select a template to generate customized code

Settings...

模板设置

高级设置

Driver Selector

Search (Ctrl+F) ⏪ ⏩ A↓

RCC

GPIO

HAL

HAL

LL

Generated Function Calls

Rank	Function Name	IP Instance Name	<input type="checkbox"/> Not Generate Function Call	<input type="checkbox"/> Visibility (Static)
1	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	SystemClock_Config	RCC	<input type="checkbox"/>	<input type="checkbox"/>

选择HAL库
或者LL库

由CubeMX生成的
外设初始化函数

是否使用CubeMX所
生成外设初始化函数

生成工程

生成基于MDK-ARM集成开发环境的工程

GENERATE CODE



工程未打开选择
“Open Project”

工程已打开选择
“Close”

MX Code Generation



The Code is successfully generated under C:/Users/qiqiang/Desktop/CubeMX/Demo

Open Folder

Open Project

Close



电子科技大学
University of Electronic Science and Technology of China

4.3 在MDK中编写应用程序

C:\Users\qiqiang\Desktop\CubeMX\Demo\MDK-ARM\Demo.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

1

菜单栏

2

工具栏

Project

- Application/MDK-ARM
- Application/User
 - main.c
 - stm32f4xx_it.c
 - stm32f4xx_hal_msp.c
- Drivers/STM32F4xx_HAL_Drv
- Drivers/CMSIS
 - system_stm32f4xx.c
 - CMSIS

3

工程窗口

main.c main.h stm32f4xx_hal.c

```
100
101  /* USER CODE BEGIN 3 */
102  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5); // 翻
103  HAL_Delay(1000); // 延
104
105  /* USER CODE END 3 */
106 }
107
```

MDK-ARM软件主界面

4

代码编辑窗口

Build Output

Build target 'Demo'
"Demo\Demo.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:00

5

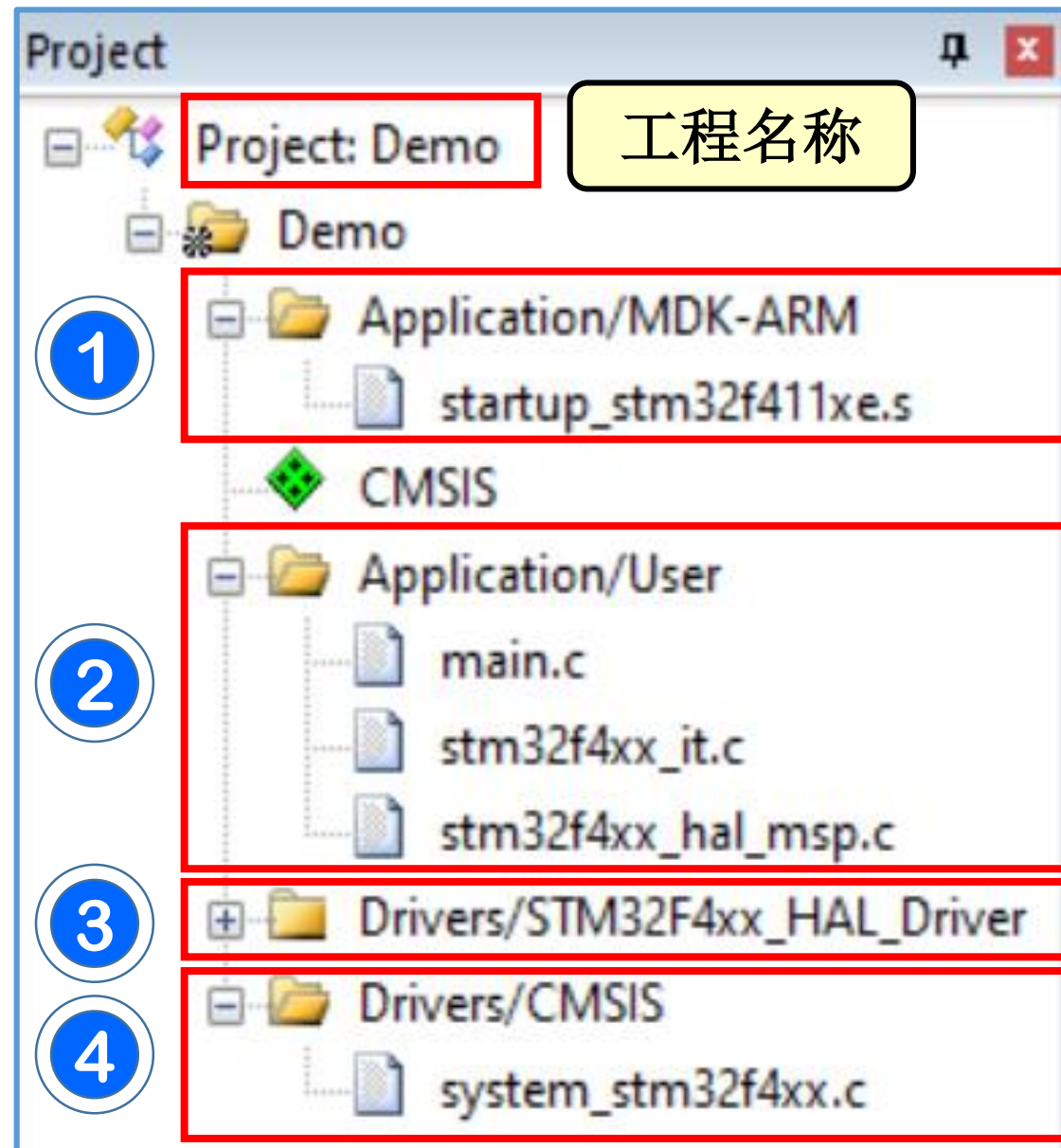
信息输出窗口

ST-Link Debugger

L:23 C:1

工程框架

- ① MDK-ARM组：启动代码文件
- ② User组：用户编程文件
- ③ HAL_Driver组：HAL库驱动文件
- ④ CMSIS组：系统初始化文件



main.c

```
1. #include "main.h"           // 包含头文件
2. void SystemClock_Config(void); // 时钟初始化函数声明
3. void MX_GPIO_Init(void);     // GPIO引脚初始化函数声明
4. int main(void)
5. {
6.     HAL_Init();              // HAL库初始化
7.     SystemClock_Config();    // 时钟初始化
8.     MX_GPIO_Init();          // GPIO引脚初始化
9.     while (1)                // while无限循环
10.    {
11.    }
12. }
```

→ 添加控制引脚电平状态变化以及延时1s的代码

删除注释语句及内部函数

接口函数：HAL_TogglePin

函数原型	void HAL_GPIO_TogglePin (GPIO_TypeDef * GPIOx, uint16_t GPIO_Pin)	
功能描述	翻转引脚电平状态	
入口参数1	GPIOx: 引脚端口号, 范围是 GPIOA ~ GPIOK	
入口参数2	GPIO_Pin: 引脚号, 范围是 GPIO_PIN0 ~ GPIO_PIN15	
返回值	无	指示灯控制引脚PA5对应的端口号为GPIOA, 引脚号为GPIO_PIN5 函数调用形式: HAL_GPIO_TogglePin(GPIOA,GPIO_PIN5)
注意事项	无	

接口函数：HAL_Delay

函数原型	void HAL_Delay (uint32_t Delay)
功能描述	延时函数，提供以ms单位的延时
入口参数	Delay：需要的延时时间，以ms为单位
返回值	无
注意事项	延时功能利用系统节拍定时器SysTick实现，将占用该定时器资源

延时1S的函数调用形式：HAL_De1ay(1000)

添加用户代码

注意

用户代码必须位于
USER CODE BEGIN 和
USER CODE END 之间

用户添加的代码

```
1. /* Infinite loop */
2. /* USER CODE BEGIN WHILE */
3. while (1)
4. {
5.     /* USER CODE END WHILE */
6.     /* USER CODE BEGIN 3 */
7.     // 翻转PA5引脚电平，控制指示灯LD2的开启和关闭
8.     HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
9.     HAL_Delay(1000); // 延时1s
10.}
11. /* USER CODE END 3 */
```

程序编译

程序编译

1

Build Output

linking...

Program Size: Code=5672 RO-data=472 RW-data=1184

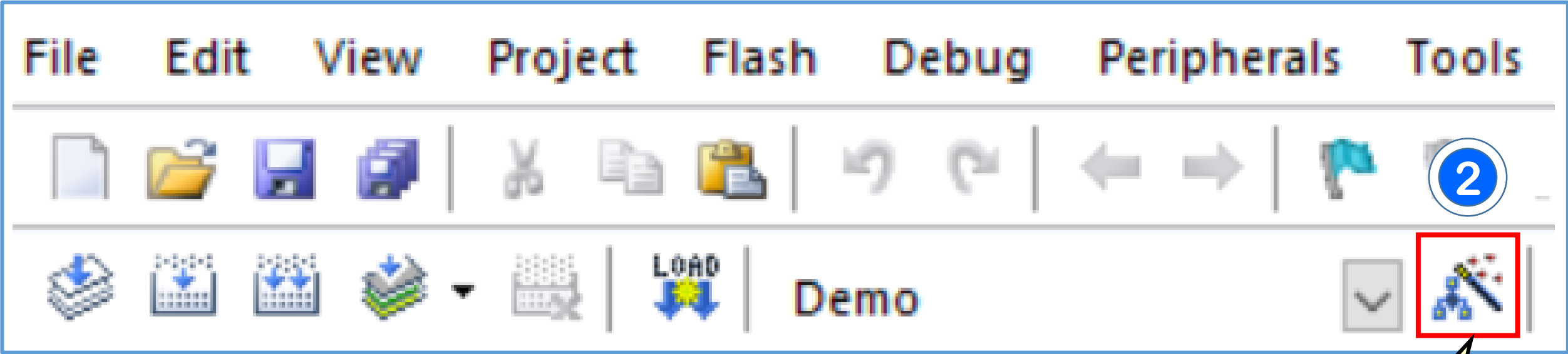
FromELF: creating hex file...

"Demo\Demo.axf" - 0 Error(s), 0 Warning(s).

Build Time Elapsed: 00:00:32

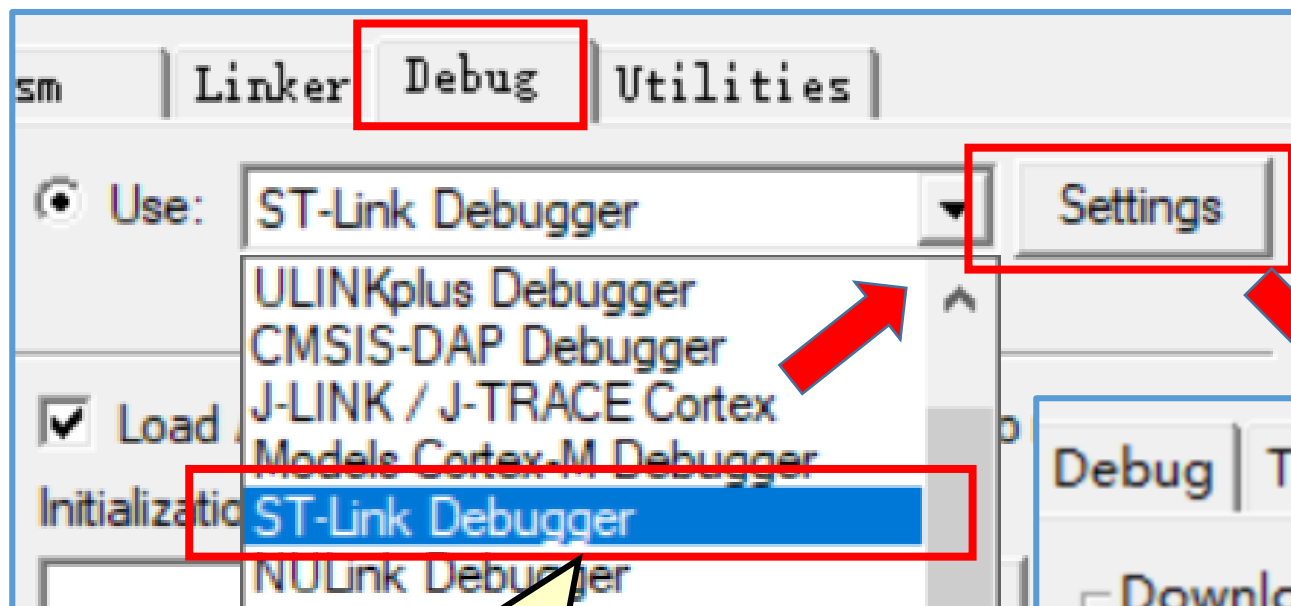
没有错误和警告

工程设置

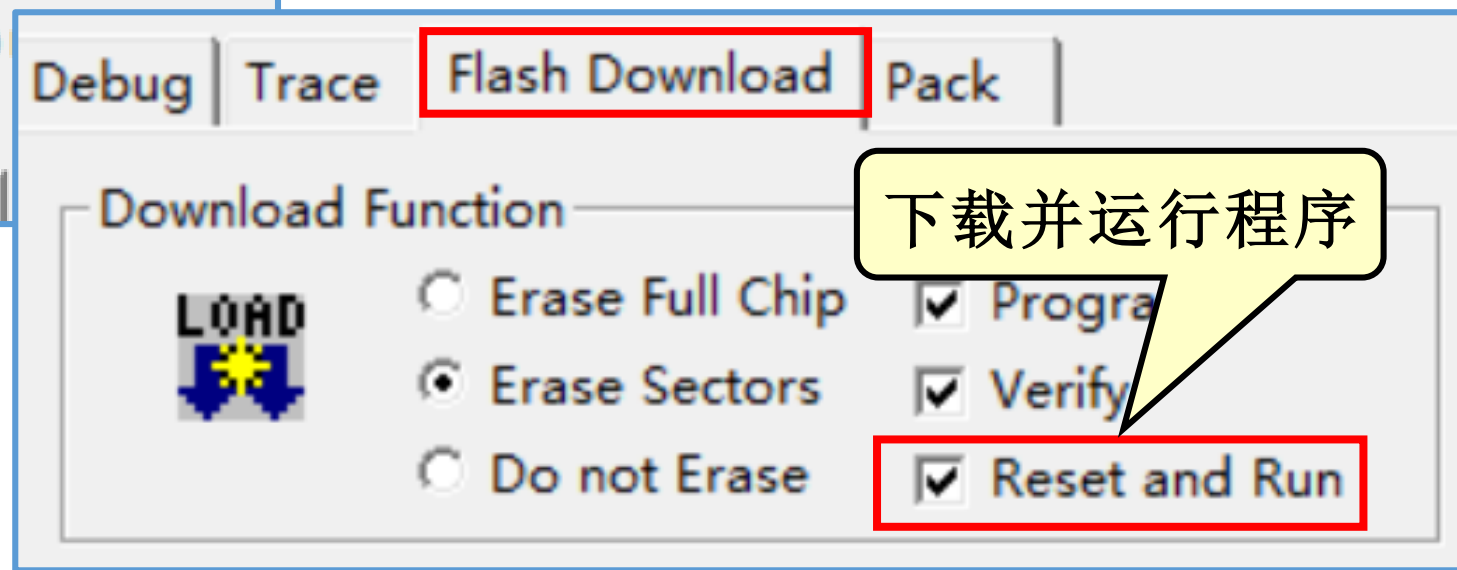


工程设置

仿真器设置



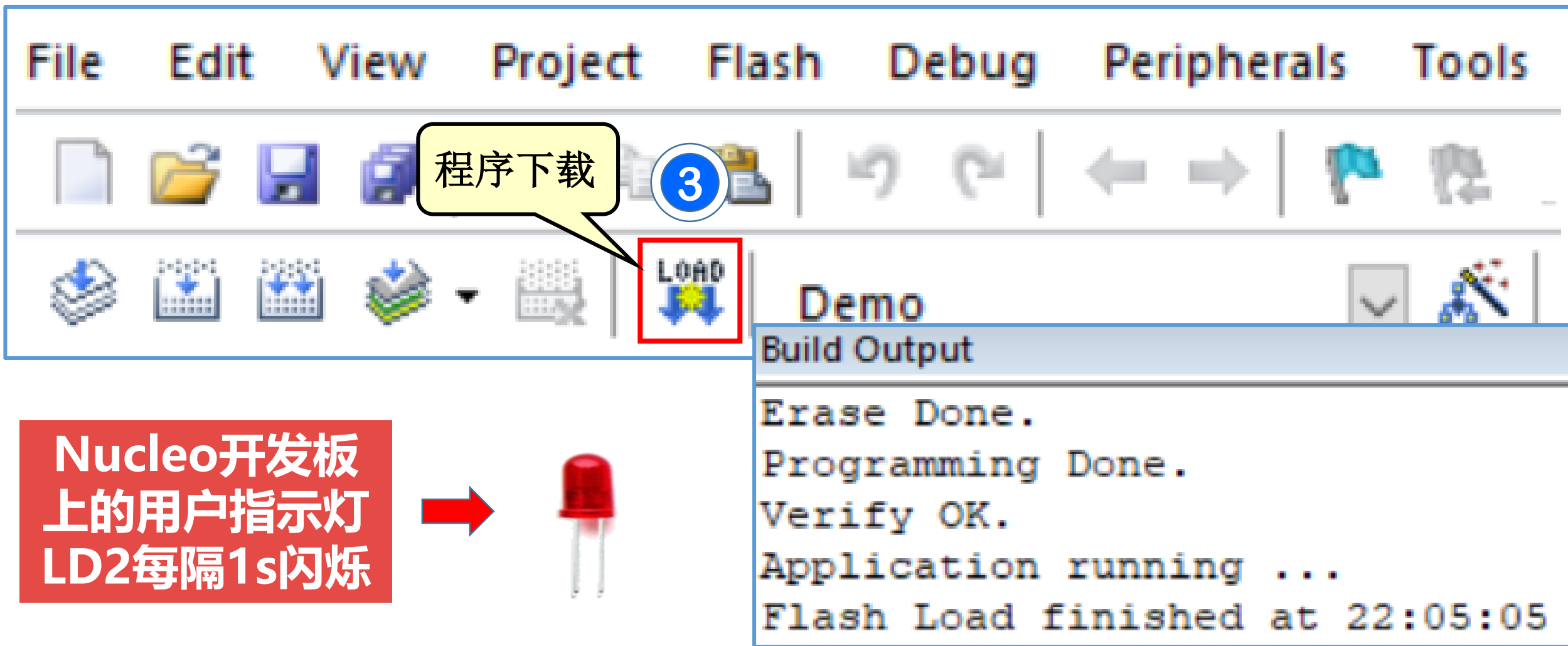
选择ST-Link仿真器



下载并运行程序

程序下载

如果程序下载后没有执行，可以按下复位按键，确保程序执行



The screenshot displays the MDK-ARM IDE interface. The menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, and Tools. The toolbar contains various icons for file operations and debugging. A yellow callout bubble with the text '程序下载' (Program Download) points to the 'LOAD' button, which is highlighted with a red rectangle and a blue circle with the number '3'. Below the toolbar, the 'Build Output' window shows the following text:

```
Demo  
Build Output  
Erase Done.  
Programming Done.  
Verify OK.  
Application running ...  
Flash Load finished at 22:05:05
```

In the bottom left corner, a red box contains the text 'Nucleo开发板上的用户指示灯LD2每隔1s闪烁' (The user indicator LED LD2 on the Nucleo development board flashes every 1s). A red arrow points from this text to a red LED component.

本章结束

本章结束

