



北京交通大学
BEIJING JIAOTONG UNIVERSITY



嵌入式系统开发

赵翔

电子信息工程学院

电工电子国家级实验教学示范中心





Keil 集成开发环境

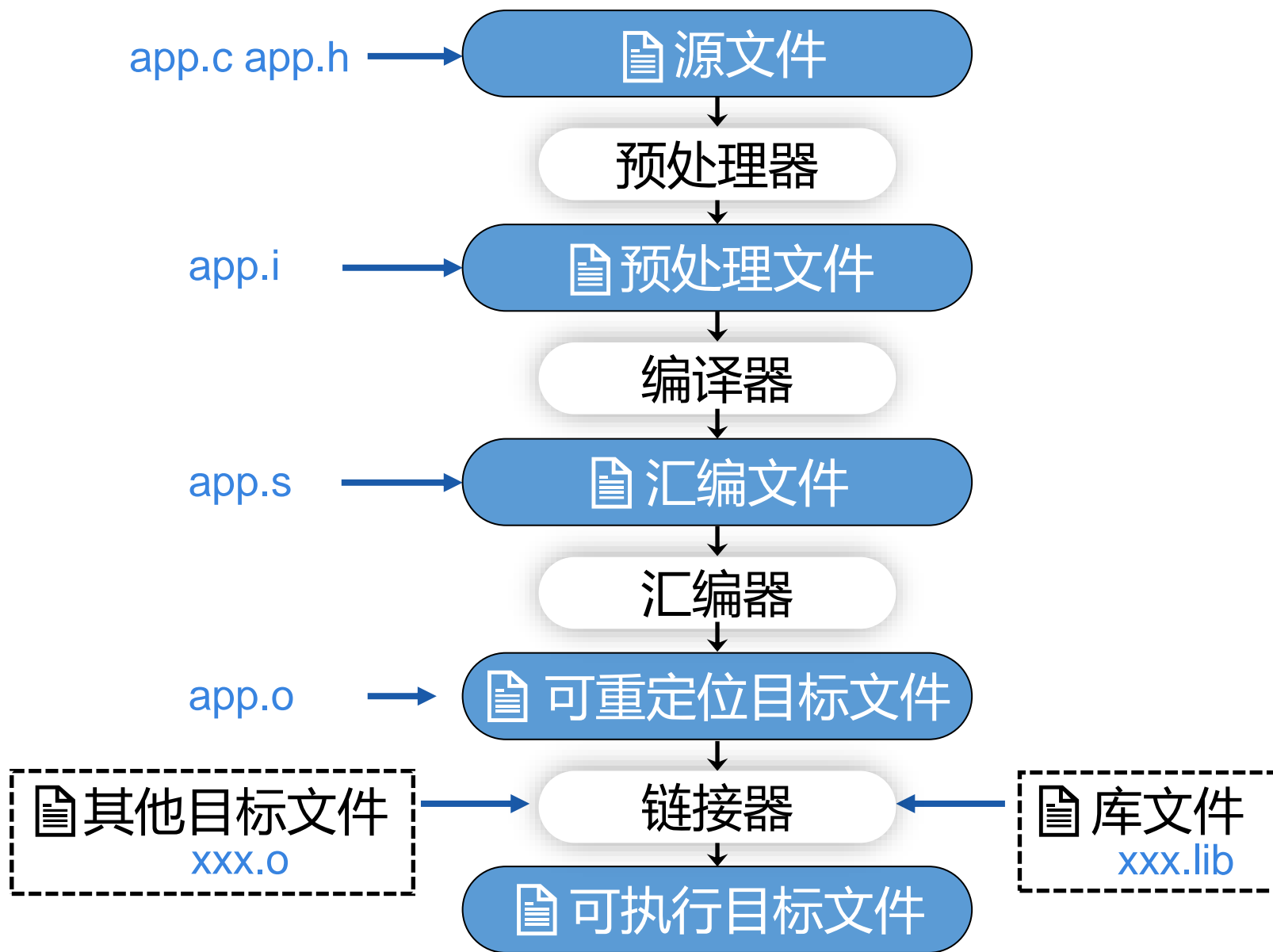




集成开发环境



嵌入式软件生成过程



预处理器

编译器

汇编器

链接器

C/C++运行库

代码编辑器

调试器

项目管理器

软件开发包管理工具

集成开发环境



Product Information

[Product Overview](#)[Supported Microcontrollers](#)[Shows and Seminars](#)

Technical Support

[Support Knowledgebase](#)[Product Manuals](#)[Application Notes](#)[Discussion Forum](#)

Software Downloads

[Product Downloads](#)[File Downloads](#)

Other Information

[Books](#)[Links](#)

Contact Information

[Corporate](#)[Sales Channels](#)[Distributors](#)

Embedded Development Tools

[Downloads](#)[Request a Quote](#)

ULINKplus

Debug adapter for
power optimization
and test automation

arm DESIGNSTART



Creating a custom
chip has never
been easier...

arm KEIL MDK

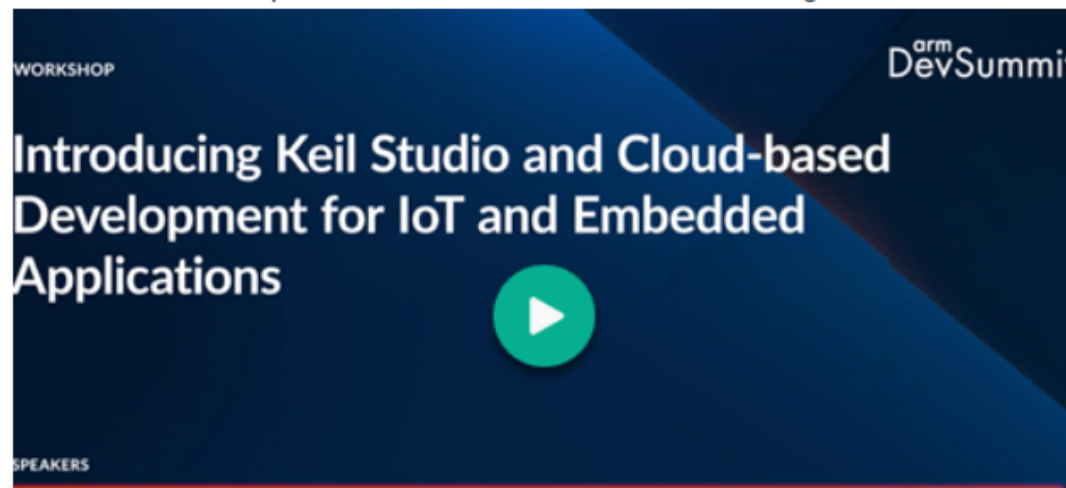
Best tools for
STM32F0/G0/L0
available for free!



MDK Microcontroller Development Kit

Keil MDK is the complete software development environment for a range of Arm Cortex-M based microcontroller devices. MDK includes the [µVision IDE](#) and [debugger](#), [Arm C/C++ compiler](#), and essential [middleware](#) components. It supports all silicon vendors with [more than 9,500 devices](#) and is easy to learn and use.

Watch the workshop from [Arm DevSummit 2021](#) introducing Keil Studio:



News

- [Arm introduces MDK-Community edition](#)
- [Keil MDK supports Arm Virtual Hardware](#)
- [Keil Studio workshop available](#)

Updates

- [MDK-ARM V5.37](#)
- [Notify Me!](#)
Receive e-mail when new updates are available.

Events

Arm Cortex-M

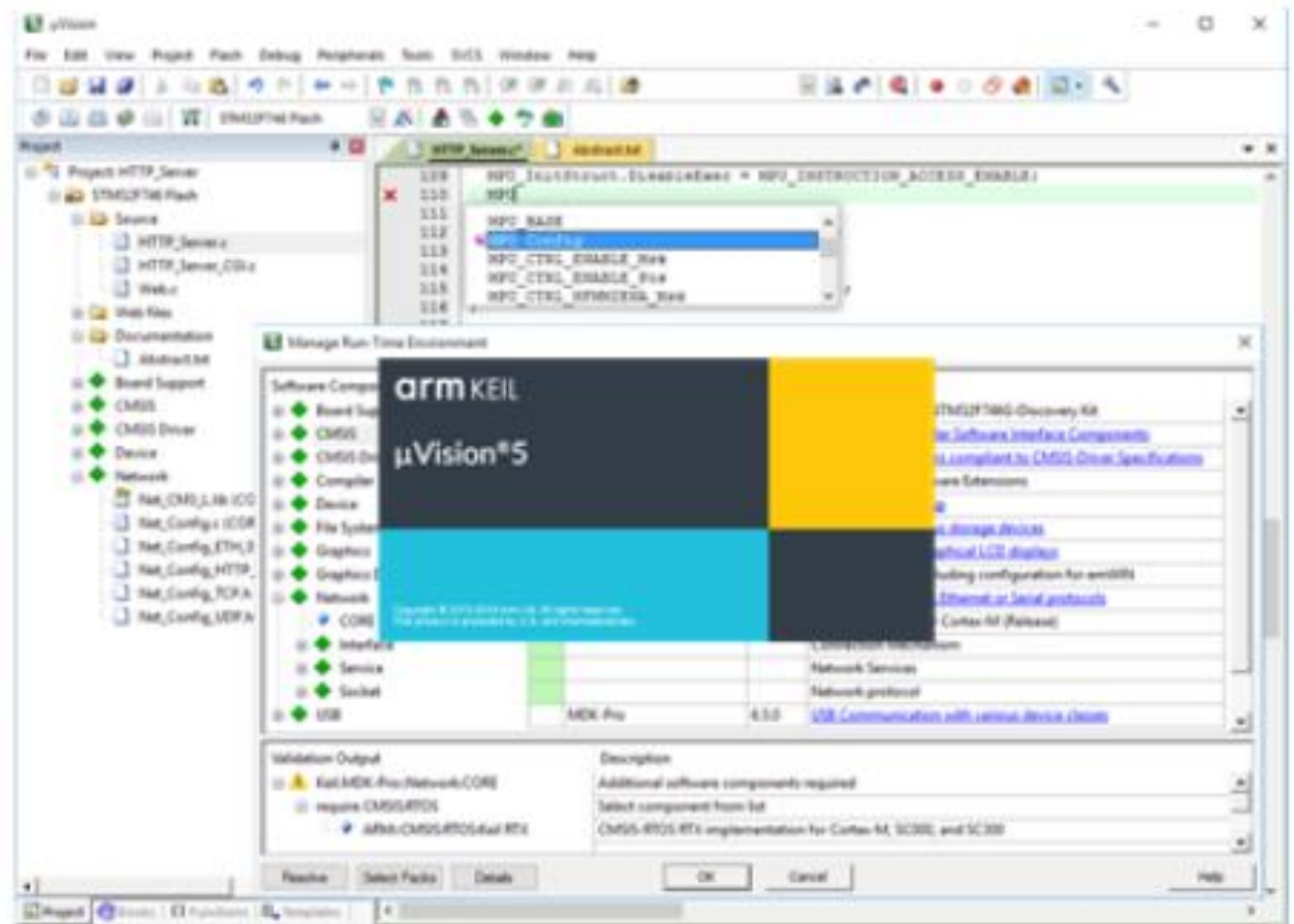
C166

8051

251

ULINK debug adapters

Evaluation boards



MDK-Arm

Version 5.37 (May 2022)

Development environment for Cortex and Arm devices.



C51

Version 9.60a (May 2019)

Development tools for all 8051 devices.



C251

Version 5.60 (May 2018)

Development tools for all 80251 devices.



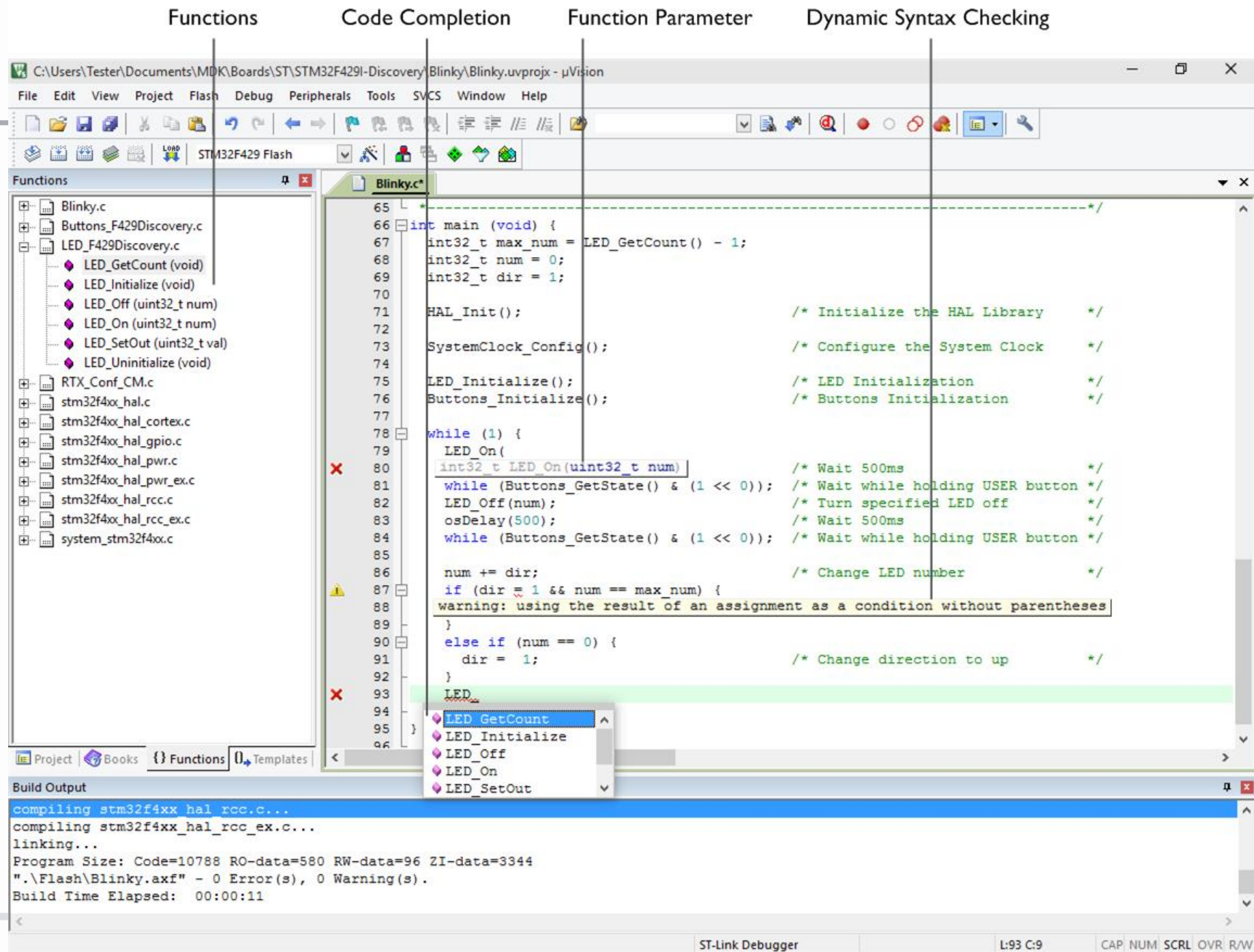
C166

Version 7.57 (May 2018)

Development tools for C166, XC166, & XC2000 MCUs.



- 菜单栏
- 工具栏
- 工作空间窗口
- 编辑窗口
- 信息窗口
- 状态栏





- 调试信息
- 反汇编窗口
- 内存窗口
- 寄存器窗口

Event Statistics Execution Timing Event Recorder

μVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

EventStart

Event Statistics

Source

- Event Start/Stop Group A - enabled
- Event Start/Stop Group B - enabled
 - Slot=0
 - Min t: Start: v1=732 v2=0
 - Max t: Start: v1=840 v2=0
- Event Start/Stop Group C - enabled
 - Slot=0
 - Min t: Start: v1=533 v2=0
 - Max t: Start: v1=917 v2=0
- Event Start/Stop Group D - enabled
 - Slot=0
 - Min t: Start: "EventRecord.c" (113)
 - Max t: Start: "EventRecord.c" (113)

EventRecord.c

RTX_Config.c

Event Recorder

Enable Recorder: ☒ Mark:

Time (sec)	Component	Event Property
50.99881209	RTX Thread	ThreadDelay
50.99881730	RTX Thread	ThreadBlocked
50.99882258	RTX Thread	ThreadSwitched
51.00878403	RTX Thread	ThreadDelayCompleted
51.00878792	RTX Thread	ThreadUnblocked
51.00879415	RTX Thread	ThreadPreempted
51.00879880	RTX Thread	ThreadSwitched
51.00880378	Prime Num...	Calculated prime number
51.00880801	RTX Thread	ThreadDelay
51.00881322	RTX Thread	ThreadBlocked
51.00881850	RTX Thread	ThreadSwitched
51.01878403	RTX Thread	ThreadDelayCompleted
51.01878792	RTX Thread	ThreadUnblocked
51.01879415	RTX Thread	ThreadPreempted

Performance Analyzer

Reset Show: Modules

Module/Function	Calls	Time(Sec)	Time(%)
EventStat_Example		222.780 s	100%
C:/Keil_v5/ARM/PACK/A...		206.983 s	93%
SVC_Handler	737	206.980 s	93%
PendSV_Handler	0	0us	0%
SysTick_Handler	8030	2.767 ms	0%
RTE/CMSIS/RTX_Config.c		15.698 s	7%
osRtxIdleThread	1	6.252 s	3%
osRtxErrorNotify	1715	9.446 s	4%
EventRecord.c		51.100 ms	0%
CalcSinTable	1	752.330 us	0%
CalcCosTable	1	880.000 us	0%
Prime	1	6.208 ms	0%
app_main	1	0.400 us	0%

Property

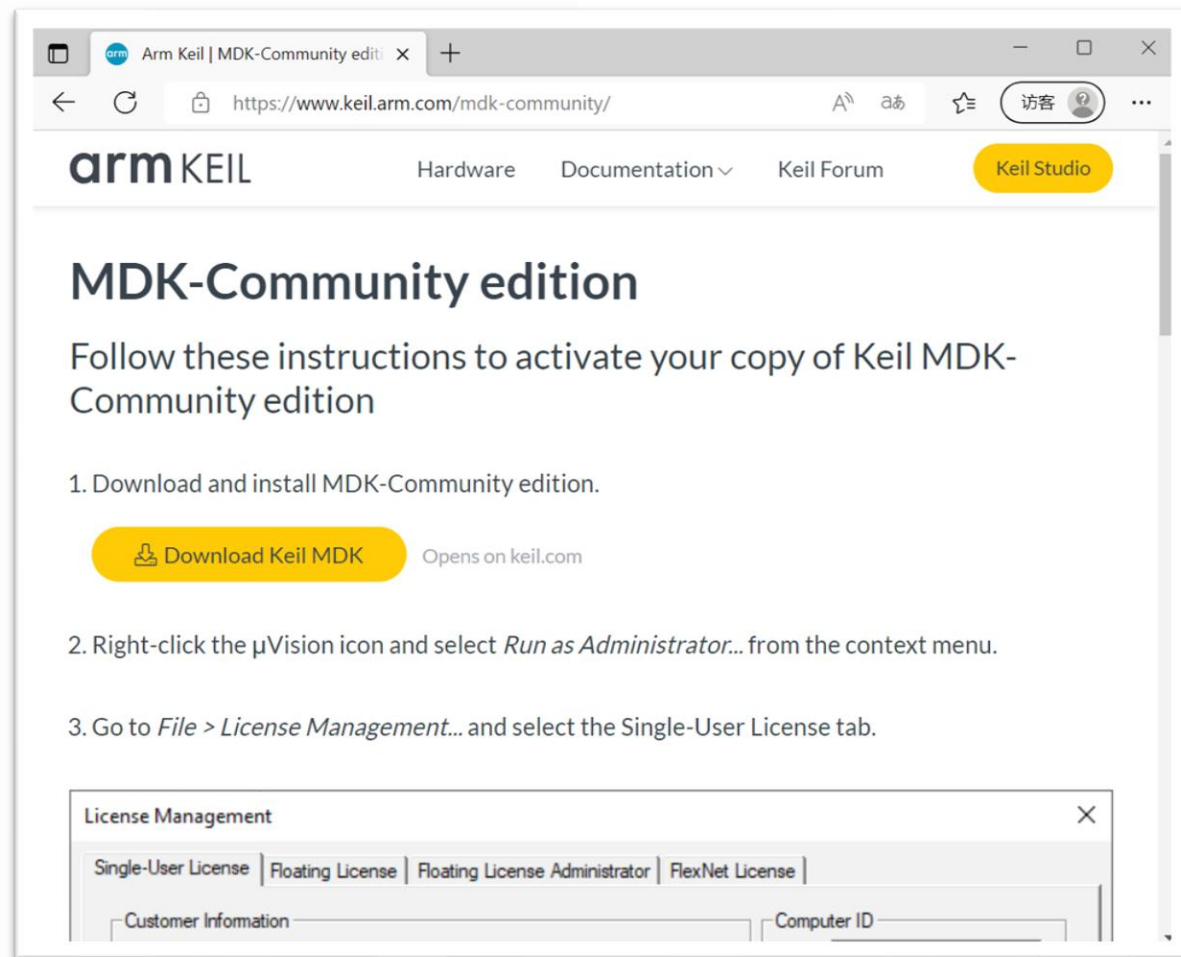
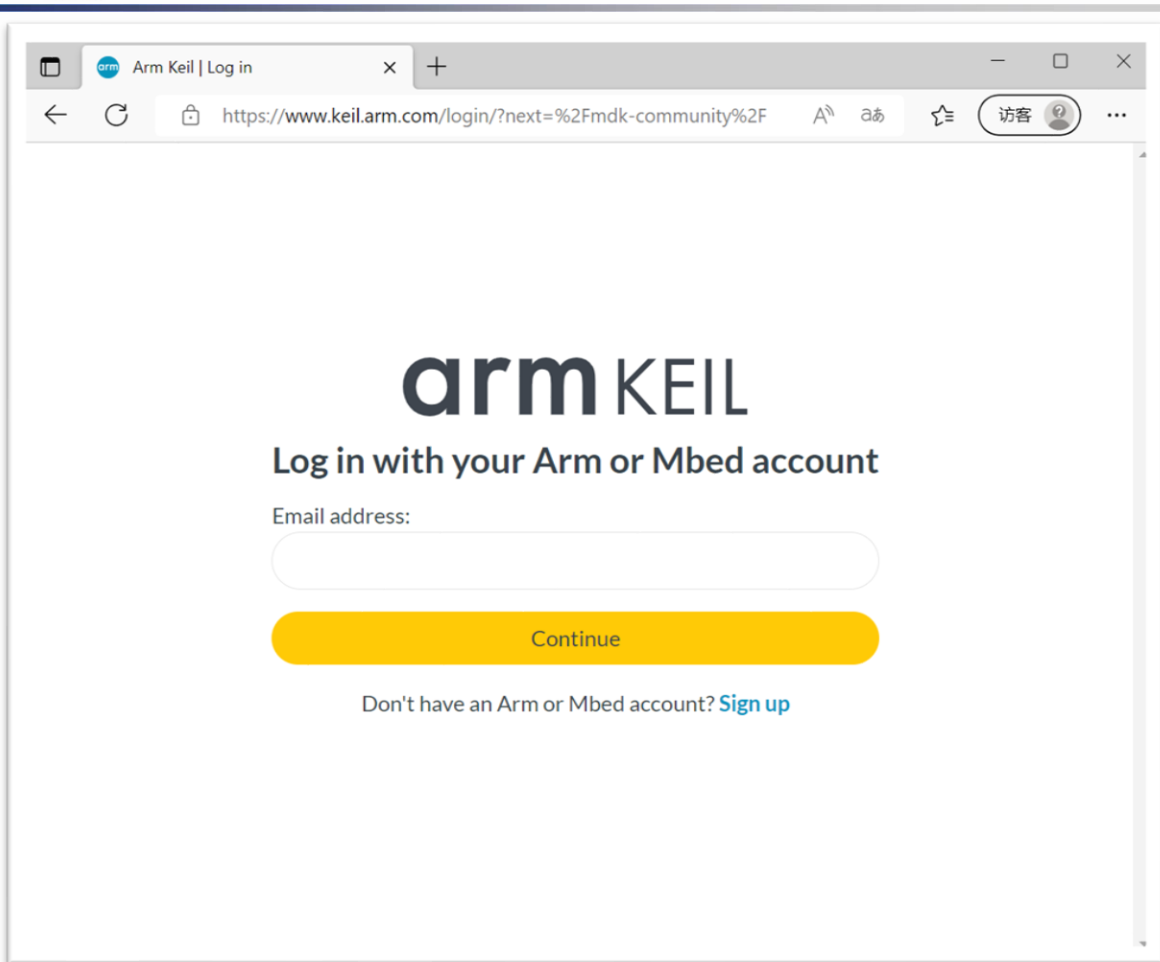
Value

- Default Thread Stack Size: 2048
- ISR FIFO Queue: Size: 16, Used: 0
- Object specific Memory allocation
- Threads
 - id: 0x200005F0, osRtxIdleThread
 - State: osThreadRunning
 - Priority: osPriorityIdle
 - Attributes: osThreadDetached
 - Stack: Used: 0% [0], Max: 25% [64]
 - Flags: 0x00000000
 - id: 0x20000634, osRtxTimerThread
 - State: osThreadBlocked
 - Priority: osPriority...
 - id: 0x200003D0, app_main
 - State: osThreadBlocked
 - Priority: osPriority...

Command Performance Analyzer

Call Stack + Locals Memory 1 RTX RTOS

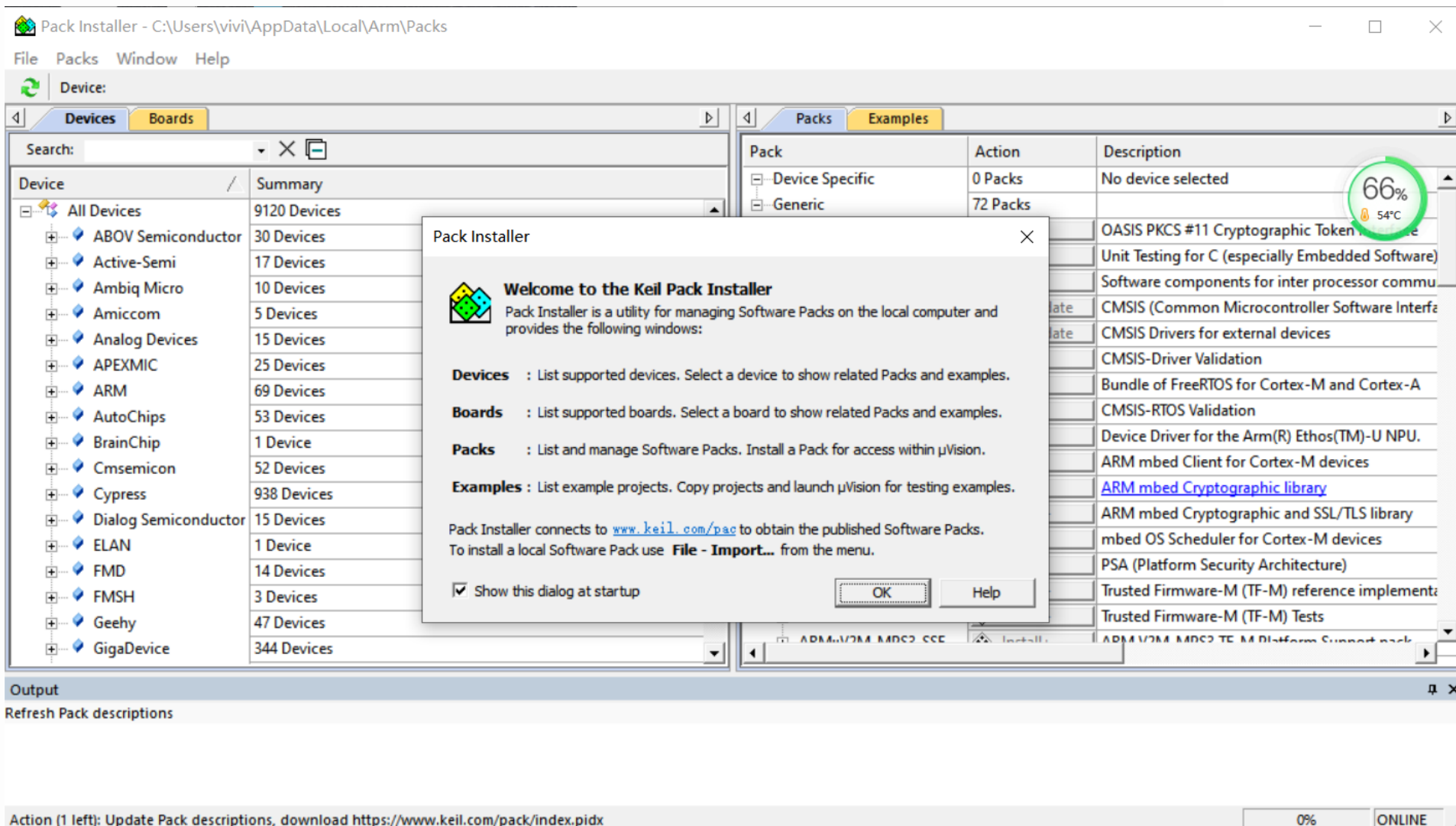
ULINK Pro Cortex Debug



<https://www.keil.arm.com/mdk-community/>



Pack





嵌入式软件生成主要步骤

创建项目

给项目添加文件

- 加入已有的源文件、编译过的目标代码
- 建立新文件并加入项目

设置编译和连接参数

- 选择处理器（内核/芯片）
- 选择运行库配置和选项
- 设置编译器预处理命令、优化参数、头文件路径等

设置调试/下载方式

- 选择调试器驱动
- 设定下载、调试运行环境





北京交通大学
BEIJING JIAOTONG UNIVERSITY



STM32驱动函数库





实现PB0输出高电平:

- `GPIOB->BSRR = 0x0001;`
- `HAL_GPIO_WritePin(GPIOB,GPIO_PIN_0, 1);`

寄存器 VS HAL库



```
/* IO operation functions *****/
GPIO_PinState HAL_GPIO_ReadPin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin);
void HAL_GPIO_WritePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState);
void HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin);
HAL_StatusTypeDef HAL_GPIO_LockPin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin);
void HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin);
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin);

/* I/O operation functions *****/
HAL_StatusTypeDef HAL_ADC_Start(ADC_HandleTypeDef* hadc);
HAL_StatusTypeDef HAL_ADC_Stop(ADC_HandleTypeDef* hadc);
HAL_StatusTypeDef HAL_ADC_PollForConversion(ADC_HandleTypeDef* hadc, uint32_t Timeout);

HAL_StatusTypeDef HAL_ADC_PollForEvent(ADC_HandleTypeDef* hadc, uint32_t EventType, uint32_t Timeout);

HAL_StatusTypeDef HAL_ADC_Start_IT(ADC_HandleTypeDef* hadc);
HAL_StatusTypeDef HAL_ADC_Stop_IT(ADC_HandleTypeDef* hadc);

void HAL_ADC_IRQHandler(ADC_HandleTypeDef* hadc);

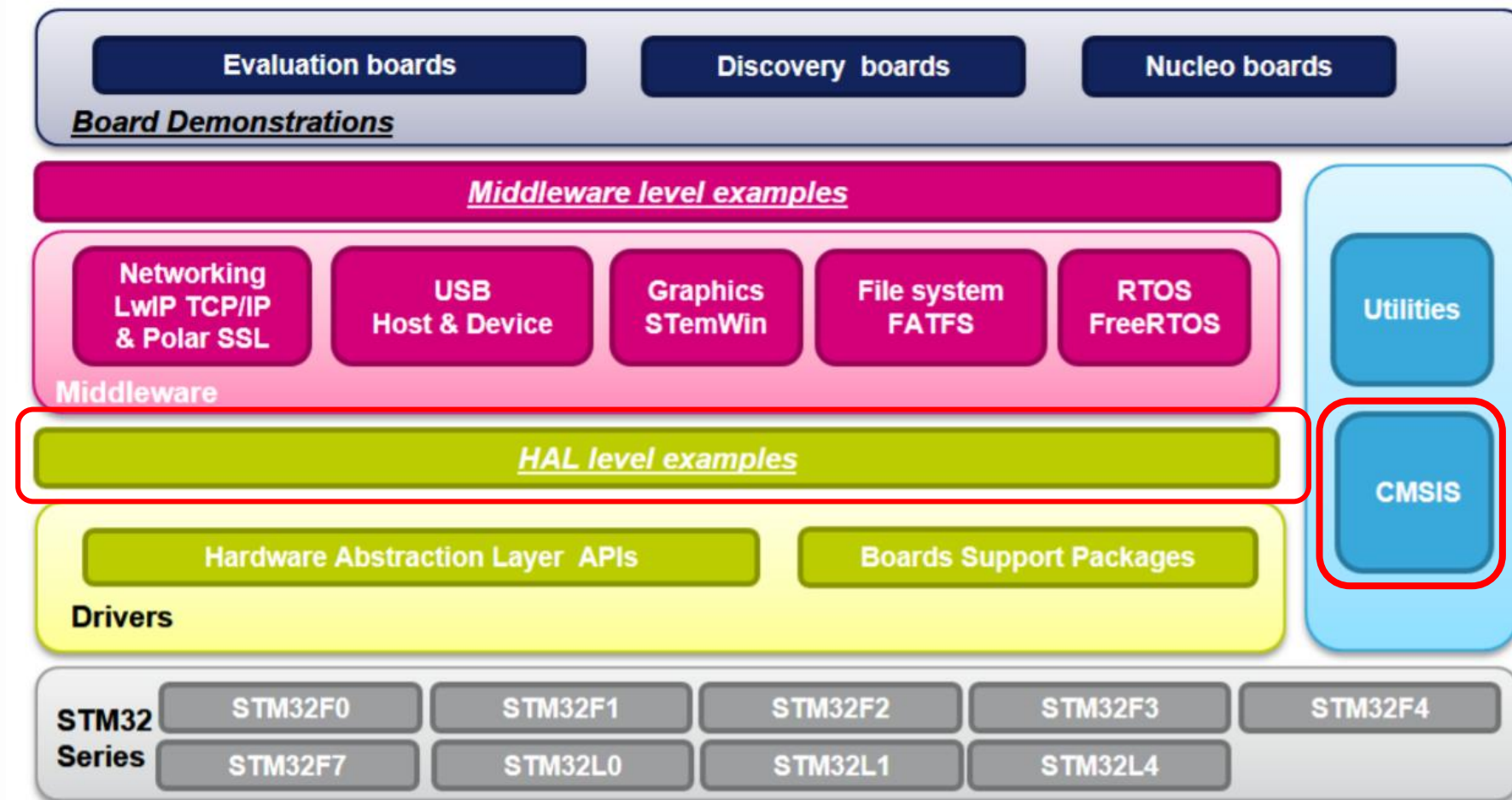
HAL_StatusTypeDef HAL_ADC_Start_DMA(ADC_HandleTypeDef* hadc, uint32_t* pData, uint32_t Length);
HAL_StatusTypeDef HAL_ADC_Stop_DMA(ADC_HandleTypeDef* hadc);

uint32_t HAL_ADC_GetValue(ADC_HandleTypeDef* hadc);

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc);
void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef* hadc);
void HAL_ADC_LevelOutOfWindowCallback(ADC_HandleTypeDef* hadc);
void HAL_ADC_ErrorCallback(ADC_HandleTypeDef *hadc);
```



驱动函数库





STM32的HAL驱动函数库

› STM32Cube_FW_F7_V1.15.0 › Drivers

名称 ^

- BSP
- CMSIS
- STM32F7xx_HAL_Driver

- Drivers
 - BSP
 - CMSIS
 - STM32F7xx_HAL_Drivers
- Middleware
 - FatFs
 - FreeRTOS
 - LwIP
 - STemWin
 - USBH_Library



STM32的HAL驱动函数库

HAL overview HAL file components

File	Description
stm32f4xx_hal_ppp.c/.h	peripheral driver with portable APIs
stm32f4xx_hal_ppp_ex.c/.h	extended peripheral features APIs
stm32f4xx_hal.c	contains HAL common APIs (HAL_Init, HAL_DeInit, HAL_Delay,...)
stm32f4xx_hal.h	HAL header file, it should be included in user code
stm32f4xx_hal_conf.h	config file for HAL, should be customized by user to select the peripherals to be included
stm32f4xx_hal_def.h	contains HAL common typedefs and macros
stm32f4xx_ll_ppp.c	implements low level functions in case of some complex peripherals, they are called from stm32f4xx_hal_ppp.c



北京交通大学
BEIJING JIAOTONG UNIVERSITY

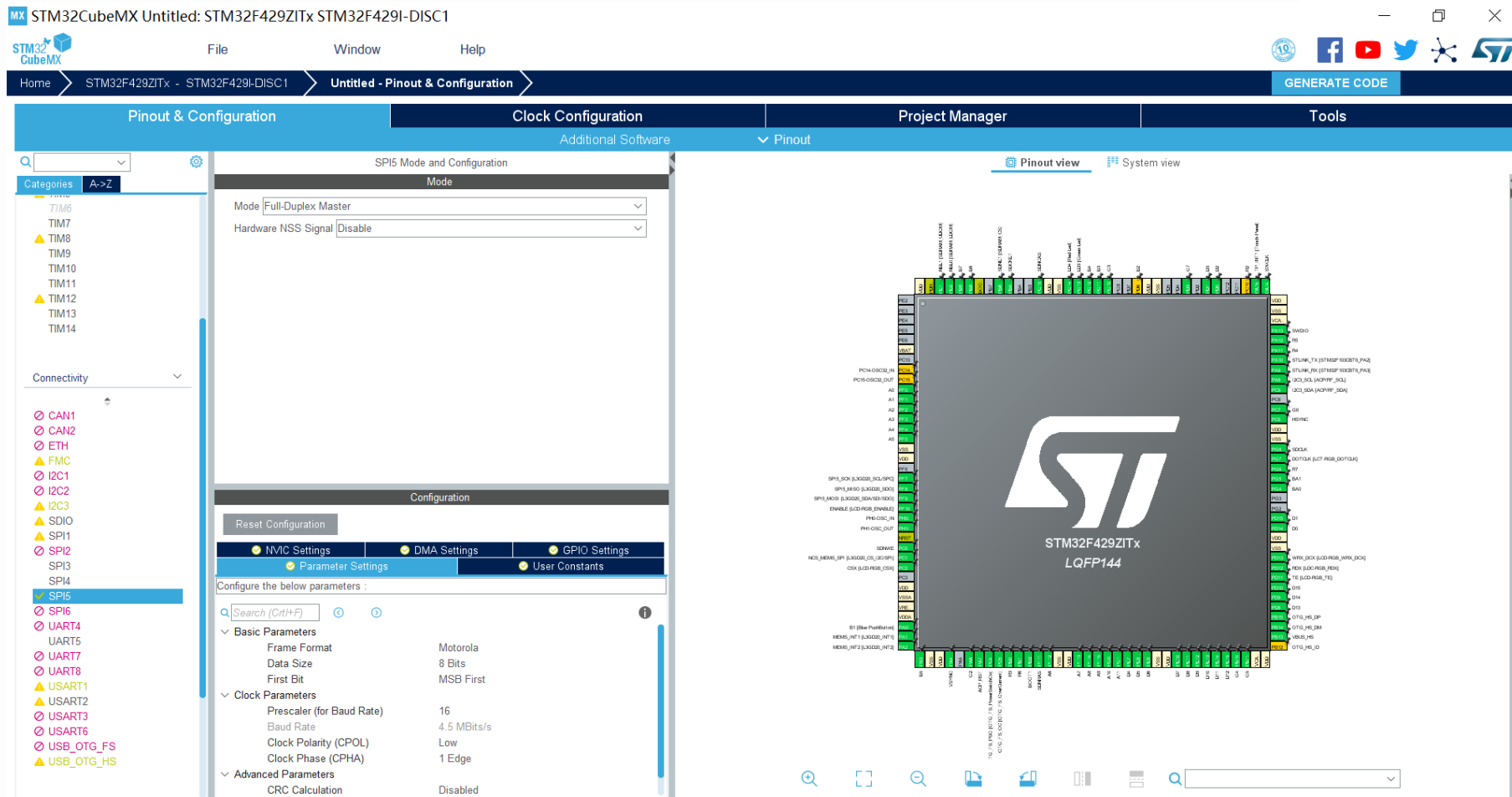


STM32CubeMX





ST - CubeMX





```
void HAL_GPIO_Init(GPIO_TypeDef *GPIOx, GPIO_InitTypeDef *GPIO_Init)
{
    uint32_t position = 0x00;
    uint32_t legposition = 0x00;
    uint32_t lincurrent = 0x00;
    uint32_t temp = 0x00;

    /* Check the parameters */
    assert_param(IS_GPIO_ALL_INSTANCE(GPIOx));
    assert_param(IS_GPIO_PIN(GPIO_Init->Pin));
    assert_param(IS_GPIO_MODE(GPIO_Init->Mode));
    assert_param(IS_GPIO_PULL(GPIO_Init->Pull));

    /* Configure the port pins */
    for(position = 0; position < GPIO_NUMBER; position++)
    {
        /* Get the IO position */
        legposition = ((uint32_t)0x01) << position;
        /* Get the current IO position */
        lincurrent = (uint32_t)(GPIO_Init->Pin) & legposition;

        if(lincurrent == legposition)
        {
            /*----- GPIO Mode Configuration -----*/
            /* In case of Alternate Function mode selection */
            if((GPIO_Init->Mode == GPIO_MODE_AF_PP) || (GPIO_Init->Mode == GPIO_MODE_AF_OD))
            {
                /* Check the pin */
                assert_param(IS_GPIO_AF(GPIO_Init->AF));

                /* Configure Alternate Function mode */
                temp = GPIOx->AF[position >> 3];
                temp &= ~(uint32_t)0x0F << ((uint32_t)position & (uint32_t)0x07) * 4;
                temp |= ((uint32_t)(GPIO_Init->Alternate) << ((uint32_t)position & (uint32_t)0x07) * 4);
                GPIOx->AF[position >> 3] = temp;
            }

            /* Configure IO Direction mode (Input, Output, Alternate or Analog) */
            temp = GPIOx->MODER;
            temp &= ~(GPIO_MODE_MODER0 << (position * 2));
            temp |= (GPIO_Init->Mode & GPIO_MODE) << (position * 2);
            GPIOx->MODER = temp;

            /* In case of Output or Alternate function mode selection */
            if((GPIO_Init->Mode == GPIO_MODE_OTTUT_PP) || (GPIO_Init->Mode == GPIO_MODE_AF_PP) ||
               (GPIO_Init->Mode == GPIO_MODE_OTTUT_OD) || (GPIO_Init->Mode == GPIO_MODE_AF_OD))
            {
                /* Check the Speed parameter */
                assert_param(IS_GPIO_SPEED(GPIO_Init->Speed));
                /* Configure the IO Speed */
                temp = GPIOx->OSPEEDR;
                temp &= ~(GPIO_OSPEEDER_OSPEEDR0 << (position * 2));
                temp |= (GPIO_Init->Speed << (position * 2));
                GPIOx->OSPEEDR = temp;

                /* Configure the IO Output Type */
                temp = GPIOx->OTYPER;
                temp &= ~(GPIO_OTYPER_OT_0 << position);
                temp |= ((GPIO_Init->Mode & GPIO_OTYPER_TYPE) >> 4) << position;
                GPIOx->OTYPER = temp;
            }

            /* Activate the Pull-up or Pull down resistor for the current IO */
            temp = GPIOx->PUPDR;
            temp &= ~(GPIO_PUPDR_PUPDR0 << (position * 2));
            temp |= (GPIO_Init->Pull) << (position * 2);
            GPIOx->PUPDR = temp;

            /*----- EXTI Mode Configuration -----*/
            /* Configure the External Interrupt or event for the current IO */
            if((GPIO_Init->Mode & EXTI_MODE) == EXTI_MODE)
            {
                /* Enable SYSCFG Clock */
                __HAL_RCC_SYSCFG_CLK_ENABLE();

                temp = SYSCFG->EXTICR[position >> 2];
                temp &= ~(((uint32_t)0x0F) << (4 * (position & 0x03)));
                temp |= ((uint32_t)(GPIO_EXT_INDEX(GPIOx)) << (4 * (position & 0x03)));
                SYSCFG->EXTICR[position >> 2] = temp;

                /* Clear EXTI line configuration */
                temp = EXTI->IMR;
                temp &= ~(uint32_t)lincurrent;
                if((GPIO_Init->Mode & GPIO_MODE_IT) == GPIO_MODE_IT)
                {
                    temp |= lincurrent;
                }
                EXTI->IMR = temp;

                temp = EXTI->EMR;
                temp &= ~(uint32_t)lincurrent;
                if((GPIO_Init->Mode & GPIO_MODE_EVT) == GPIO_MODE_EVT)
                {
                    temp |= lincurrent;
                }
                EXTI->EMR = temp;

                /* Clear Rising/Falling edge configuration */
                temp = EXTI->RISR;
                temp &= ~(uint32_t)lincurrent;
                if((GPIO_Init->Mode & RISING_EDGE) == RISING_EDGE)
                {
                    temp |= lincurrent;
                }
                EXTI->RISR = temp;
            }
        }
    }
}
```

初始化代码

VS

工作代码

```
void HAL_GPIO_WritePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState)
{
    /* Check the parameters */
    assert_param(IS_GPIO_PIN(GPIO_Pin));
    assert_param(IS_GPIO_PIN_ACTION(PinState));

    if(PinState != GPIO_PIN_RESET)
    {
        GPIOx->BSRR = GPIO_Pin;
    }
    else
    {
        GPIOx->BSRR = (uint32_t)GPIO_Pin << 16;
    }
}
```

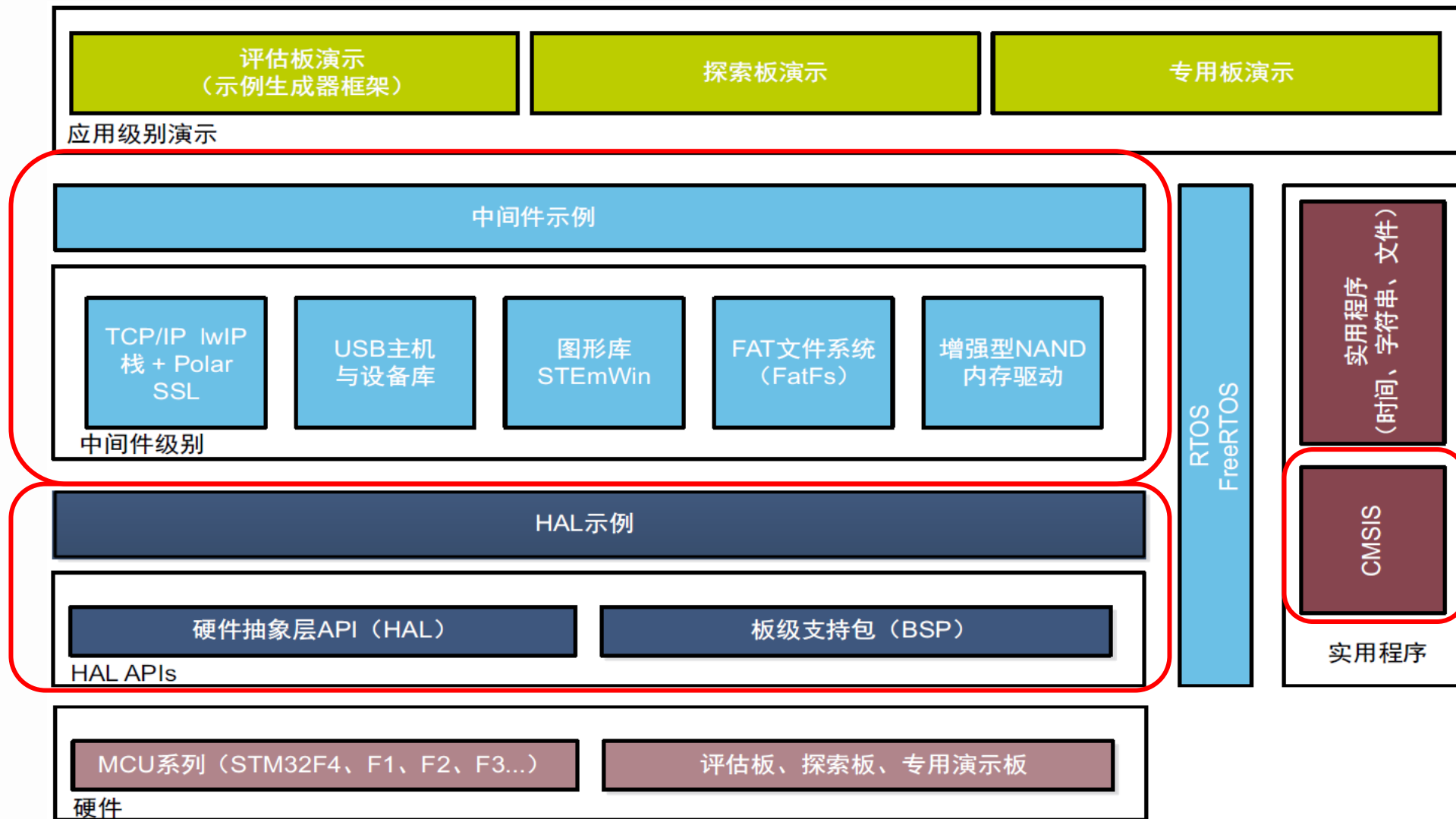




✚ STM32CubeMX是用于STM32微控制器的图形工具。它既可作为独立应用，也可作为Eclipse插件集成到集成开发环境（IDE）中。

✚ STM32CubeMX 有以下主要特性：

- 微控制器选择方便，覆盖整个STM32产品
- 可从一系列意法半导体的开发板中选择板子
- 微控制器配置简单（引脚、时钟树、外设、中间件）
- 将以前保存的配置导入新的MCU项目即可轻松地转换到其他微控制器
- 将当前配置轻松地导出到兼容的MCU
- 生成配置报告
- 为IAR KEIL GCC等开发环境生成嵌入C项目包括生成的初始化代码、HAL驱动程序、中间件协议栈，以及所有相关文件。





```
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */
}
```



+ 04 嵌入式系统软件开发工具

4-5 实验-Keil集成开发环境

4-8 实验-STM32CubeMX图形化开发工具