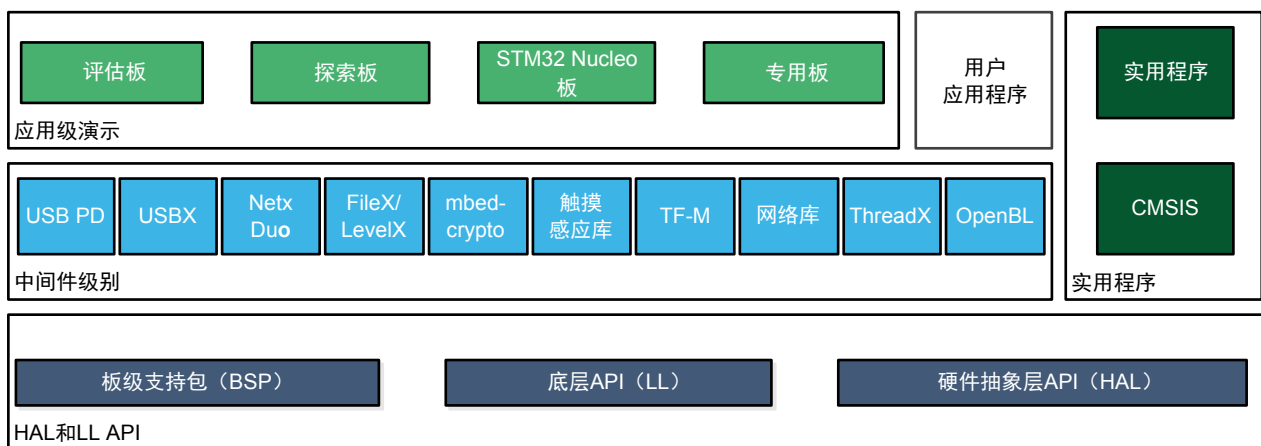


## STM32U5 系列的 STM32Cube MCU 包示例

### 引言

STM32CubeU5 MCU 包带一组丰富的运行于意法半导体板件之上的示例。示例按板件进行管理，提供预先配置的项目给主要支持的工具链（请参考）。

图 1. STM32CubeU5 固件组件



## 1 参考文档

以下各项构成了本应用笔记中出现的示例的参考集合：

- 用于 STM32U5 系列的 32 位微控制器（基于具有 Arm® TrustZone® 的 Arm® Cortex®-M 处理器）最新的 [STM32CubeU5](#) MCU 包
- [面向 STM32U5 系列的 STM32CubeU5 入门 \(UM2883\)](#)
- [STM32U5 HAL 与底层驱动程序说明 \(UM2911\)](#)
- [STM32CubeU5 TFM 应用程序入门 \(UM2851\)](#)
- [Arm® TrustZone® STM32 系列微控制器上的安全启动和安全固件更新解决方案概述 \(AN5447\)](#)

提示

Arm 和 TrustZone 是 Arm Limited（或其子公司）在美国和或其他地区的注册商标。



## 2 STM32CubeU5 示例

示例根据其适用的 STM32Cube 级别进行分类。各个类别的命名如下：

- **示例**  
 这些示例只使用 HAL 和 BSP 驱动程序（不使用中间件）。其目的是演示产品或外设的特性和使用方式。这些示例按外设进行管理（每个外设（如 TIM）一个文件夹）。从给定外设的基础应用（如使用定时器生成 PWM）到多个外设的集成（如使用 DAC 从 TIM6 和 DMA 同步生成信号），各类情况的复杂程度不一。板件资源的使用量严格地控制在最低水平。
- **Examples\_LL**  
 这些示例只使用 LL 驱动程序（不使用 HAL 驱动程序和中间件组件）。它们提供了外设特性和配置流程的典型用例的最佳实现。示例按外设进行管理（每个外设（如 TIM）一个文件夹），主要部署在 Nucleo 板上。
- **Examples\_MIX**  
 这些示例只使用 HAL、BSP 和 LL 驱动程序（不使用中间件组件）。它们旨在演示如何在同一应用中使用 HAL 和 LL API，以便结合两种 API 的优势：
  - HAL 提供面向高级功能的 API，具有高度可移植性，这是通过对最终用户隐藏产品/IP 复杂性来实现的。
  - LL 提供寄存器层面的底层 API，获得更好的优化效果。
 示例按外设进行管理（每个外设（如 TIM）一个文件夹），只部署在 Nucleo 板上。
- **应用**  
 应用演示产品性能和可用中间件栈的使用方式。应用按中间件（每个中间件（如 Azure® RTOS ThreadX）一个文件夹）或需要高级固件模块（如 LPBAM）的产品特性进行管理。此外，还支持使用多个中间件栈的应用的集成。
- **演示**  
 演示旨在集成和运行最大数量的外设和中间件栈，以便展示产品特性和性能。
- **模板项目**  
 提供的模板项目帮助用户在给定板件上使用 HAL 和 BSP 驱动程序快速构建固件应用。
- **Template\_LL 项目**  
 提供的模板 LL 项目帮助用户在给定板件上使用 LL 驱动程序快速构建固件应用。

示例位于 STM32Cube\_FW\_U5\_VX.Y.Z\Projects\目录下。

这些示例具有相同结构，默认产品的配置禁用 Arm® TrustZone®

- \*Inc 文件夹，包含所有头文件
- 包含源代码的\*Src 文件夹
- \*EWARM、\*MDK-ARM 和\*STM32CubeIDE 文件夹包含每个工具链的预配置项目
- \*README.md 和\*readme.html 文件，描述了示例行为和运行示例所需的环境

启用了 Arm® TrustZone®的示例带有后缀“\_TrustZone”（TFM 应用除外）并具有相同结构：

- \*Secure\Inc 文件夹，包含所有安全项目头文件
- \*Secure\Src 和\*Secure\_nsclib\文件夹，包含所有安全项目源代码
- \*NonSecure\Inc 文件夹，包含所有非安全项目头文件
- \*NonSecure\Src 文件夹，包含所有非安全项目源代码
- \*EWARM、\*MDK-ARM 和\*STM32CubeIDE 文件夹包含每个工具链的预配置项目
- \*README.md 和\*readme.html 文件，描述了示例行为和运行示例所需的环境

为了运行示例，执行如下操作：

1. 使用首选工具链打开示例。
2. 重新编译所有文件，并将二进制文件加载到目标内存中。
3. 按照\*README.md 和\*readme.html 文件中的指示运行示例。

### 提示

如需了解关于 MCU 包开发和确认使用的软件/硬件环境的更多信息，请参考固件包版本说明中的“开发工具链和编译器”和“支持的器件和评估板”章节。不保证所提供示例在其他环境中也能正确运行，例如在使用不同编译器或板件版本时。

可以对示例进行定制，以便在任何兼容的硬件上运行：如果具有相同硬件功能（LED、LCD、按钮及其他），只需更新板件的 BSP 驱动程序即可。BSP 基于模块化架构，可通过实现低级例程轻松移植到任何硬件。

表 1. STM32CubeU5 固件示例 包含随 STM32CubeU5 MCU 包提供的示例的列表。


该表格中的标签  表示项目是使用 STM32CubeMX，即 STM32Cube 初始化代码生成器创建的。这些项目可使用此工具打开，以便修改项目本身。其他项目为手动创建，用于演示产品特性。该表格中的标签 TrustZone 表示项目是为启用了 Arm® TrustZone® 的器件创建的。读取项目 \*\README.md 和 \*\readme.html

表 1. STM32CubeU5 固件示例

STM32CubeMX 生成的示例标有MX。

级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-H0T02A
模板	-	TrustZoneDisabled	此项目提供了基于 STM32Cube HAL API 的参考模板，在没有能使安全特性（TZEN = 0）的情况下，可用于构建任何固件应用。	X	X	X
		TrustZoneEnabled	此项目提供了基于 STM32Cube HAL API 的参考模板，在激活了 TrustZone®安全特性（选项位 TZEN = 1）的情况下，可用于构建任何固件应用。	X	X	X
	模板总数：6			2	2	2
Templates_LL	-	TrustZoneDisabled	此项目提供了使用 LL API 的参考模板，可用于构建任何固件应用。	X	X	X
	templates_ll 的总数：3			1	1	1
示例	-	BSP	如何使用板件的不同 BSP 驱动程序。	X	-	X
	ADC	ADC_AnalogWatchdog	如何使用具有 ADC 模拟看门狗的 ADC 外设来监控通道并检测相应转换数据是否超出窗口阈值。	-	MX	-
		ADC_DMA_Transfer	如何配置和使用 ADC2 转换外部模拟输入，并使用 DMA 传输通过 HAL API 获取结果。	-	MX	-
		ADC_DifferentialMode	该示例描述如何配置和使用 ADC1 在差分模式下转换外部模拟输入（VINN 和 VINP 上外部电压之间的差异）。	-	MX	-
		ADC_SingleConversion_TriggerSW_IT	如何在每次软件启动时使用 ADC 转换单个通道。使用中断编程模型执行转换。	-	MX	-
	COMP	COMP_Interrupt	如何使用比较器外设在中断模式下比较施加在 GPIO 引脚上的电压电平和内部电压基准（VREFINT）。	-	MX	-
	CORDIC	CORDIC_Sin_DMA	如何在 DMA 模式下使用 CORDIC 外设计算正弦波数组。	-	MX	-
	CORTEX	CORTEXM_ModePrivilege	如何修改 Thread 模式特权访问和栈。在复位时或从异常返回时进入 Thread 模式。	-	MX	-
		CORTEXM_SysTick	如何使用具有 1 ms 时基的默认 SysTick 配置切换 LED。	-	MX	-

级别	模块名称	项目名称	说明	STM32U575I-LEV	NUCLEO-U575I-Q	B-U585I-H0T02A
示例	CRC	CRC_Bytes_Stream_7bit_CRC	如何使用 HAL API 配置 CRC。CRC（循环冗余校验）计算单元计算 8 位数据（字节）缓冲区的 7 位 CRC 码。	-	MX	-
		CRC_Example	如何使用 HAL API 配置 CRC。CRC（循环冗余校验）计算单元使用固定的生成多项式（0x4C11DB7）计算给定 32 位数据字缓冲区的 CRC 码。	-	MX	-
		CRC_UserDefinedPolynomial	如何使用 HAL API 配置 CRC。CRC（循环冗余校验）计算单元基于用户定义的生成多项式计算给定 32 位数据字缓冲区的 8 位 CRC 码。	-	MX	-
	CRYPT	CRYPT_AES_GCM	如何使用 CRYPT 外设，在伽罗瓦计数器模式（GCM）下使用 AES 加密和解密数据。	-	-	MX
		CRYPT_SAES_ECB_CBC	如何在安全特性禁用（TZEN = 0）的情况下，通过安全 AES 协处理器（SAES）外设使用 AES ECB 和 CBC 算法加密和解密数据。	-	-	MX
		CRYPT_SAES_SharedKey	如何使用安全 AES 协处理器（SAES）外设与 AES 外设共享应用密钥。	-	-	MX
		CRYPT_SAES_WrapKey	如何通过安全 AES 协处理器（SAES）外设使用硬件密钥 DHUK 封装应用密钥，然后使用此密钥在轮询模式下执行加密。	-	-	MX
	DAC	DAC_SignalsGeneration_DMA	如何通过 DMA 控制器使 DAC 外设生成正弦信号。	-	MX	-
		DAC_SimpleConversion	如何使用 DAC 外设执行简单转换。	-	MX	-
	DCACHE	DCACHE_Maintenance	如何在两个主设备（CPU 和 DMA）访问的共享存储器缓冲区上执行数据缓存维护。	MX	-	-
	DCMI	DCMI_ContinuousCap_EmbeddedSynchMode	该示例描述了如何配置 DCMI 外设的连续模式和内嵌码同步模式。在安全特性禁用（TZEN = 0）的情况下，帧捕获的挂起和恢复基于 STM32Cube HAL API。	MX	-	-
	DLYB	DLYB_OSPI_NOR_FastTuning	如何使用具有快速调谐功能的延迟模块（DLYB）。	MX	-	-
		DLYB_OSPI_PSRAM_ExhaustiveTuning	如何使用具有全面调谐功能的延迟模块（DLYB）。	-	-	MX
	DMA	DMA_DataHandling	如何使用 DMA 控制器，通过 HAL API 执行传输自源的数据与传输到目标的数据之间的数据处理。	-	MX	-
		DMA_FLASHToRAM	如何使用 DMA，通过 HAL API 将 Flash 存储器中的字数据缓冲传输到嵌入式 SRAM。	-	MX	-



级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-H0T02A
示例	DMA	DMA_LinkedList	如何使用 DMA 执行传输列表。传输列表采用链表结构。每当当前传输结束时，DMA 自动重载下一个传输的参数并开始传输（无 CPU 干预）。	-	MX	-
		DMA_RepeatedBlock	如何配置和使用 DMA HAL API，以便执行重复块事务。	MX	-	-
		DMA_Trigger	如何配置和使用 DMA HAL API，以便执行 DMA 触发事务。	MX	-	-
	DMA2D	DMA2D_BlendingWithAlphaInversion	在安全特性禁用（TZEN = 0）的情况下，如何基于 STM32Cube HAL API 为存储器至存储器模式下的 DMA2D 外设配置混合传输模式和 $\alpha$ 反转模式。	MX	-	-
	FDCAN	FDCAN_Loopback	如何配置 FDCAN，以便以环回模式操作。	MX	-	-
	FLASH	FLASH_ChangeOptionBytes	如何配置和使用 FLASH HAL API，以便修改 STM32U5 器件选项字节。	-	X	-
		FLASH_EraseProgram	如何配置和使用 FLASH HAL API，以便进行内部 Flash 存储器的擦除和编程。	-	MX	-
		FLASH_EraseProgram_TrustZone	如何在 TrustZone®安全特性激活（选项位 TZEN = 1）的情况下配置和使用 FLASH HAL API，以便进行内部 Flash 存储器的擦除和编程。	-	X	-
	FMAC	FMAC_IIR_PollingToDMA	如何使用 FMAC 外设执行从轮询模式到 DMA 模式的 IIR 滤波。	-	MX	-
	FMC	FMC_SRAM	如何在安全特性禁用（TZEN = 0）的情况下配置 FMC 控制器，以便基于 STM32Cube HAL API 访问 SRAM。	MX	-	-
		FMC_SRAM_ReadWrite_DMA	如何在安全特性禁用（TZEN = 0）的情况下配置 FMC 控制器和 DMA，以便基于 STM32Cube HAL API 访问 SRAM。	MX	-	-
	GPIO	GPIO_EXTI	如何配置外部中断线。	-	MX	MX
		GPIO_IOToggle	如何通过 HAL API 配置和使用 GPIO。	MX	MX	MX
		GPIO_IOToggle_TrustZone	如何在 TrustZone®安全特性激活（选项位 TZEN = 1）的情况下使用 HAL GPIO API，以便切换安全和非安全 I/O。	-	MX	MX
	GTZC	GTZC_MPCWM_IllegalAccess_TrustZone	如何在 TrustZone®安全特性激活（选项位 TZEN = 1）的情况下使用 GTZC MPCWM 和 TZIC 构建任何示例。	MX	-	-

级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-H0T02A
示例	GTZC	GTZC_TZSC_MPCBB_TrustZone	如何在 TrustZone®安全特性激活（选项位 TZEN = 1）的情况下使用 HAL GTZC MPCBB，以便构建具有 SecureFault 检测功能的任何示例。	-	MX	-
	HAL	HAL_TimeBase_RTC_ALARM	如何自定义HAL使用RTC 报警作为主要时基来源，代替SysTick。	-	MX	-
		HAL_TimeBase_RTC_WKUP	如何自定义HAL使用RTC 唤醒作为主要时基来源，代替SysTick。	-	MX	-
		HAL_TimeBase_TIM	如何自定义HAL使用通用定时器作为主要时基来源，代替SysTick。	-	MX	-
	HASH	HASH_HMAC_SHA1MD5	如何使用 HASH 外设，以便通过 HMAC SHA-1 和 HMAC MD5 算法生成哈希数据。	-	MX	-
	I2C	I2C_TwoBoards_AdvComIT	如何使用中断处理一个主设备和一个从设备之间的多重 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_TwoBoards_ComDMA	如何以 DMA 模式处理两个板件之间的 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_TwoBoards_ComDMA_Autonomous_Master	如何以 DMA 模式通过 GPDMA1 通道 3 触发信号自主处理两个板件之间的 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_TwoBoards_ComDMA_Autonomous_Slave	如何以 DMA 模式通过 GPDMA1 通道 3 触发信号自主处理两个板件之间的 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_TwoBoards_ComDMA_LowPower	如何以 DMA 模式处理两个板件之间低功耗模式下的 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_TwoBoards_ComIT	如何使用中断处理两个板件之间的 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_TwoBoards_ComPolling	如何以轮询模式处理两个板件之间的 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_TwoBoards_RestartAdvComIT	如何以中断模式并使用重新开始条件执行两个板件之间的多重 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_TwoBoards_RestartComIT	如何以中断模式并使用重新开始条件处理两个板件之间的单一 I2C 数据缓冲发送/接收。	-	MX	-
		I2C_WakeUpFromStop	如何在器件处于停机模式时使用中断处理两个板件之间的 I2C 数据缓冲发送/接收。	-	MX	-



级别	模块名称	项目名称	说明	STM32U575I-LEV	NUCLEO-U575I-Q	B-U585I-H0T02A
示例	ICACHE	ICACHE_Memory_Remap	如何执行通过 ICACHE HAL 驱动程序配置的重映射区中的代码。	MX	-	-
	IWDG	IWDG_Reset	如何处理 IWDG 重载计数器，以及模拟软件故障，即在预设时间量后生成 MCU IWDG 复位。	-	MX	-
		IWDG_WindowMode	如何定期更新 IWDG 重载计数器，以及模拟软件故障，即在预设时间量后生成 MCU IWDG 复位。	-	MX	-
	LPTIM	LPTIM_IC_LSE	如何通过 HAL LPTIM API，在低功耗模式下使用 LPTIM 外设测量外部信号的频率（使用 LSE 作为计数器时钟）。	-	MX	-
		LPTIM_PWM_LSE	如何通过 HAL LPTIM API，在低功耗模式下配置和使用 LPTIM 生成 PWM（使用 LSE 作为计数器时钟）。	-	MX	-
	MDF	ADF_AudioRecorder	如何使用 MDF HAL API（ADF 实例）执行单路音频录音。	MX	-	-
		ADF_AudioSoundDetector	如何通过 MDF HAL API（ADF 实例）来使用音频声音活动检测功能。	-	-	MX
	OCTOSPI	OSPI_HyperRAM_MemoryMapped	如何在存储器映射模式下使用 OCTOSPI HyperRAM 存储器。	MX	-	-
		OSPI_HyperRAM_ReadWrite_IT	如何在间接模式下使用 OCTOSPI HyperRAM 存储器。	MX	-	-
		OSPI_NOR_AutoPolling_DTR	如何在自动轮询模式下使用 OCTOSPI NOR Flash 存储器。	-	-	MX
		OSPI_NOR_MemoryMapped	如何在存储器映射模式下使用 OCTOSPI NOR Flash 存储器。	MX	-	-
		OSPI_NOR_ReadWrite_DMA_DTR	如何在 DMA 模式下使用 OCTOSPI NOR Flash 存储器。	-	-	MX
		OSPI_PSRAM_ExecutelnPlace	如何在代码加载后执行 OCTOSPI 存储器中的代码。	-	-	MX
		OSPI_PSRAM_MemoryMapped	如何在存储器映射模式下使用 OCTOSPI PSRAM 存储器。	-	-	MX
	OPAMP	OPAMP_Follower	如何在与 DAC 和 COMP 互连的情况下配置 OPAMP 外设的跟随模式。	-	MX	-

级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-IOT02A
示例	OTFDEC	OTFDEC_Data_Decrypt	如何使用 OTFDEC 外设解密 Octo-SPI 外部 Flash 存储器中的数据（使用 CRYIP 外设加密）。B-U585I-IOT02 LED 用于如下状态监控：当校验的数据正确时，绿色 LED 点亮。	-	-	MX
	PKA	PKA_ECCDoubleBaseLadder	如何使用 PKA 运行 ECC 双基阶梯运算。该示例旨在运行于意法半导体 B-U585I-IOT02A 板内置的 STM32U585AIxQ 器件上。	-	-	MX
		PKA_ECCProjective2Affine	如何使用 PKA 运行 ECC 射影转仿射运算。该示例旨在运行于意法半导体 B-U585I-IOT02A 板内置的 STM32U585xx 器件上。	-	-	MX
		PKA_ECDSA_Sign	如何基于椭圆曲线数字签名算法（ECDSA）计算签名消息。	-	-	MX
		PKA_ModExpProtected_IT	如何使用 PKA 运行受保护模幂运算。该示例旨在运行于意法半导体 B-U585I-IOT02A 板内置的 STM32U585xx 器件上。	-	-	MX
	PWR	PWR_LPMode_RTC	如何进入不同的可用低功耗模式，并使用 RTC 唤醒定时器生成的中断从这些模式唤醒。	-	MX	-
		PWR_ModesSelection	如何使用 HAL 驱动程序配置系统，以便测量不同低功耗模式下的电流消耗。	-	MX	-
		PWR_SLEEP	如何进入睡眠模式，并使用中断从该模式唤醒。	-	MX	MX
		PWR_STANDBY	如何进入待机模式，并使用外部复位或 WKUP 引脚从该模式唤醒。	-	MX	MX
	RAMCFG	RAMCFG_ECC_Error_Generation	如何配置和使用 RAMCFG HAL API，以便通过 RAMCFG 外设管理 ECC 错误。	-	MX	-
		RAMCFG_WriteProtection	如何配置和使用 RAMCFG HAL API，以便配置 RAMCFG SRAM 写保护页。	MX	-	-
	RCC	RCC_ClockConfig	如何使用 RCC HAL API 配置系统时钟（SYSCLK）并在运行模式下修改时钟设置。	MX	MX	MX
		RCC_LSEConfig	如何使用 RCC HAL API 在运行时间启用/禁用低速外部（LSE）RC 振荡器（约 32 KHz）。	-	MX	-
		RCC_LSIConfig	如何使用 RCC HAL API 在运行时间启用/禁用低速内部（LSI）RC 振荡器（约 32 KHz）。	-	MX	-
	RNG	RNG_MultiRNG	如何使用 HAL API 配置 RNG。该示例使用 RNG 生成 32 位随机数。	-	MX	-



级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575I-Q	B-U585I-H0T02A
示例	RNG	RNG_MultiRNG_IT	如何使用 HAL API 配置 RNG。该示例使用 RNG 中断生成 32 位随机数。	-	MX	-
	RTC	RTC_ActiveTamper	如何配置主动篡改检测用来备份寄存器擦除。	-	MX	-
		RTC_Alarm	如何使用 RTC HAL API 配置和生成 RTC 报警。	-	MX	-
		RTC_Calendar	如何使用 RTC HAL API 配置日历。	MX	-	-
		RTC_LSI	如何使用 LSI 时钟源自动校准获取精确 RTC 时钟。	-	MX	-
		RTC_LowPower_STANDBY_WUT	如何通过 RTC 唤醒定时器（WUT）定期进入和从待机模式唤醒。	-	MX	-
		RTC_Tamper	如何配置篡改检测用于备份寄存器擦除。	-	-	MX
		RTC_TimeStamp	如何配置 RTC HAL API 以便演示时间戳特性。	-	MX	-
		RTC_TrustZone	如何在 TrustZone 安全特性激活（选项位 TZEN = 1）的情况下配置 TrustZone®感知 RTC 外设。RTC 的有些特性可能是安全的，而其他特性可能是非安全的。	-	MX	-
	SAI	SAI_AudioPlay	如何使用 DMA 循环模式，通过 SAI 播放音频文件。	MX	-	-
	SD	SD_ReadWrite_DMALinkedList	该示例基于链表特性，以 SDMMC 内部 DMA 模式对 SD 卡执行一些写入和读取传输。	MX	-	-
	智能卡	SMARTCARD_ComDMA	该示例展示了如何在 DMA 模式下与智能卡通信。	MX	-	-
	SMBUS	SMBUS_TwoBoards_ComIT_Autonomous_Master	如何以自主模式处理两个板件之间的 SMBUS 数据缓冲发送/接收。	-	MX	-
		SMBUS_TwoBoards_ComIT_Autonomous_Slave	如何以自主模式处理两个板件之间的 SMBUS 数据缓冲发送/接收。	-	MX	-
		SMBUS_TwoBoards_ComIT_Master	如何以中断模式处理两个板件之间的 SMBUS 数据缓冲发送/接收。	-	MX	-

级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-H0T02A
示例	SMBUS	SMBUS_TwoBoards_ComIT_Slave	如何以中断模式处理两个板件之间的 SMBUS 数据缓冲发送/接收。	-	MX	-
	SPI	SPI_FullDuplex_ComDMA_Autonomous_Master	如何通过 SPI 以 DMA 模式自主处理两个板件之间的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComDMA_Autonomous_Slave	如何通过 SPI 以 DMA 模式自主处理两个板件之间的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComDMA_LowPower_Master	如何通过 SPI 以 DMA 模式处理两个板件之间低功耗模式下的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComDMA_LowPower_Slave	如何通过 SPI 以 DMA 模式处理两个板件之间低功耗模式下的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComDMA_Master	如何通过 SPI 以 DMA 模式处理两个板件之间的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComDMA_Slave	如何通过 SPI 以 DMA 模式处理两个板件之间的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComIT_Master	如何通过 SPI 以中断模式处理两个板件之间的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComIT_Slave	如何通过 SPI 以中断模式处理两个板件之间的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComPolling_Master	如何通过 SPI 以轮询模式处理两个板件之间的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComPolling_Slave	如何通过 SPI 以轮询模式处理两个板件之间的数据缓冲发送/接收。	-	MX	-
	TIM	TIM_InputCapture	如何使用 TIM 外设测量外部信号频率。	-	MX	-
		TIM_OCActive	如何在输出比较活动模式下配置 TIM 外设（当计数器匹配捕获/比较寄存器时，对应的输出引脚置为主动状态）。	-	MX	-
		TIM_OCInactive	如何在输出比较非活动模式下根据每个通道的对应中断请求配置 TIM 外设。	-	MX	-
		TIM_OCToggle	如何配置 TIM 外设以四种不同频率生成四个不同信号。	-	MX	-

级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575I-Q	B-U585I-H0T02A
示例	TIM	TIM_PWMInput	如何使用 TIM 外设测量外部信号的频率和占空比。	-	MX	-
		TIM_PWMOutput	该样例展示了如何在 PWM（脉宽调制）模式下配置 TIM 外设。	-	MX	-
	TSC	TSC_BasicAcquisition	如何使用 TSC 以轮询模式执行一个通道的连续采集。	MX	-	-
		TSC_BasicAcquisition_Interrupt	如何使用 TSC 以中断模式执行一个通道的连续采集。	MX	-	-
	UART	UART_Printf	如何将 C 库 printf 函数重路由至 UART。	MX	-	-
		UART_ReceptionToIdle_CircularDMA	如何使用 HAL UART API 以 DMA 循环模式接收 IDLE 事件。	-	MX	-
		UART_TwoBoards_ComDMA	如何在两块板之间以 DMA 模式执行 UART 发送（发送/接收）。	-	MX	-
		UART_TwoBoards_ComDMAlinkedlist	如何在两块板之间使用链表以 DMA 模式执行 UART 发送（发送/接收）。	-	MX	-
		UART_TwoBoards_ComIT	如何在两块板之间以中断模式执行 UART 发送（发送/接收）。	-	MX	-
		UART_TwoBoards_ComPolling	如何在两块板之间以轮询模式执行 UART 发送（发送/接收）。	-	MX	-
	USART	USART_SlaveMode	该示例描述了两块板之间的 USART-SPI 通信（发送/接收），其中 USART 被配置为从设备。	-	MX	-
		USART_SlaveMode_DMA	该示例描述了两块板之间使用 DMA 的 USART-SPI 通信（发送/接收），其中 USART 被配置为从设备。	-	MX	-
	WWDG	WWDG_Example	如何配置 HAL API，以便在预定义的时间段结束时定期更新 WWDG 计数器并模拟生成 MCU WWDG 复位的软件故障。	-	MX	-
	示例总数：136			26	87	23
Examples_LL	ADC	ADC_AnalogWatchdog_Init	如何使用具有 ADC 模拟看门狗的 ADC 外设来监控通道并检测相应转换数据是否超出窗口阈值。	-	MX	-

级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-H0T02A
Examples_LL	ADC	ADC_Oversampling_Init	如何使用具有过采样功能的 ADC 外设。	-	MX	-
		ADC_SingleConversion_TriggerSW_IT_Init	如何在每次软件启动时使用 ADC 转换单个通道。使用中断编程模型执行转换。	-	MX	-
		ADC_SingleConversion_TriggerSW_Init	如何在每次软件启动时使用 ADC 转换单个通道。使用轮询编程模型执行转换。	-	MX	-
	CRC	CRC_CalculateAndCheck	如何配置 CRC 计算单元，以便基于固定的发生器多项式（默认值为 0x4C11DB7）计算给定数据缓冲区的 CRC 码。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
		CRC_UserDefinedPolynomial	如何配置和使用 CRC 计算单元，以便基于用户定义的生成多项式来计算给定数据缓冲区的 8 位 CRC 码。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
	DMA	DMA_LinkedList	如何使用 DMA 执行传输列表。传输列表采用链表结构。每当当前传输结束时，DMA 自动重载下一个传输的参数并开始传输（无 CPU 干预）。	-	X	-
	EXTI	EXTI_ToggleLedOnIT_Init	该示例展示了如何配置 EXTI，以及用户按钮被按下时如何使用 GPIO 点亮/熄灭板上的用户 LED。该示例基于 STM32U5xx LL API。使用 LL 初始化功能执行外设初始化，以便演示 LL 初始化的使用方法。	-	MX	-
	GPIO	GPIO_InfiniteLedToggling_Init	如何配置和使用 GPIO，以便每隔 250 ms 点亮/熄灭板载用户 LED。	-	MX	-
	I2C	I2C_OneBoard_Communication_IT_Init	I2C 主设备接收来自 I2C 从设备的一个数据字节时，如何处理接收过程。两个设备均以中断模式工作。使用 LL 初始化功能初始化外设，以便演示 LL 初始化的使用方法。	-	MX	-
	IWDG	IWDG_RefreshUntilUserEvent_Init	如何配置 IWDG 外设，以确保定期更新计数器并在按下 PUSH1 按钮时生成 MCU IWDG 复位。使用 LL 单一服务功能初始化外设，以便优化性能和大小。	-	MX	-
	PWR	PWR_EnterStandbyMode	如何进入待机模式，并使用外部复位或唤醒引脚从该模式唤醒。	-	MX	-
		PWR_EnterStopMode	如何进入停机 0 模式。	-	MX	-
	RCC	RCC_OutputSystemClockOnMCO	如何配置 MCO 引脚（PA8），以便输出系统时钟。	-	MX	-
		RCC_UseHSI_PLLasSystemClock	如何在运行时间修改 PLL 参数。	-	MX	-
	RNG	RNG_GenerateRandomNumbers	如何配置 RNG，以便生成 32 位随机数。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-



级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-H0T02A
Examples_LL	RNG	RNG_GenerateRandomNumbers_IT	如何配置 RNG，以便使用中断生成 32 位随机数。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
	RTC	RTC_Alarm_Init	如何配置 RTC LL API，以便使用 RTC 外设配置和生成报警。使用 LL 初始化功能初始化外设。	-	MX	-
		RTC_ExitStandbyWithWakeUpTimer_Init	如何通过 RTC 唤醒定时器（WUT）定期进入和从待机模式唤醒。	-	MX	-
		RTC_Tamper_Init	如何使用 RTC LL API 配置 Tamper。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
		RTC_TimeStamp_Init	如何使用 RTC LL API 配置 Timestamp。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
	SPI	SPI_OneBoard_HalfDuplex_IT_Init	如何配置 GPIO 和 SPI 外设，以便以中断模式从 SPI 主设备发送字节到 SPI 从设备。该示例基于 STM32U5xx SPI LL API。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
		SPI_TwoBoards_FullDuplex_IT_Master_Init	使用中断模式通过 SPI 发送和接收数据缓冲。该示例基于 STM32U5xx SPI LL API。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
		SPI_TwoBoards_FullDuplex_IT_Slave_Init	使用中断模式通过 SPI 发送和接收数据缓冲。	-	MX	-
	TIM	TIM_BreakAndDeadtime_Init	如何配置 TIM 外设，以便生成三个中心对齐 PWM 和互补 PWM 信号，插入指定的死区时间值，使用断路特征，并锁定断路和死区时间配置。	-	MX	-
		TIM_InputCapture_Init	如何使用 TIM 外设测量外部信号发生器或另一个定时器实例提供的周期信号频率。	-	MX	-
		TIM_OnePulse_Init	如何配置定时器，从而在输出比较模式下在延迟 t <sub>DELAY</sub> 后生成长度为 t <sub>PULSE</sub> 的正脉冲。	-	MX	-
		TIM_OutputCompare_Init	如何配置 TIM 外设，以便以不同的输出比较模式生成输出波形。该示例基于 STM32U5xx TIM LL API。	-	MX	-
		TIM_PWMOutput_Init	如何使用 TIM 外设生成 PWM 输出信号并更新 PWM 占空比。	-	MX	-
		TIM_TimeBase_Init	如何配置 TIM 外设，以便生成时基。	-	MX	-
	USART	USART_Communication_Rx_IT_Continuous_Init	如何配置 GPIO 和 USART 外设，以便使用中断模式以异步模式连续接收来自超级终端（PC）的字符。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-



级别	模块名称	项目名称	说明	STM32U575I-LEV	NUCLEO-U575I-Q	B-U585I-HOT02A
Examples_LL	USART	USART_Communication_Rx_IT_Continuous_VCP_Init	如何配置 GPIO 和 USART 外设，以便使用中断模式以异步模式连续接收来自超级终端（PC）的字符。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
		USART_Communication_Rx_IT_Init	如何配置 GPIO 和 USART 外设，以便使用中断模式以异步模式接收来自超级终端（PC）的字符。使用 LL 初始化功能执行外设初始化，以便演示 LL 初始化的使用方法。	-	MX	-
		USART_Communication_Rx_IT_VCP_Init	如何配置 GPIO 和 USART 外设，以便使用中断模式以异步模式接收来自超级终端（PC）的字符。使用 LL 初始化功能执行外设初始化，以便演示 LL 初始化的使用方法。	-	MX	-
		USART_Communication_Tx_IT_Init	如何配置 GPIO 和 USART 外设，从而以中断模式向超级终端（PC）异步发送字符。该示例基于 STM32U5xx USART LL API。使用用于优化用途（性能和大小）的 LL 单一服务功能执行外设初始化。	-	MX	-
		USART_Communication_Tx_IT_VCP_Init	如何配置 GPIO 和 USART 外设，从而以中断模式向超级终端（PC）异步发送字符。该示例基于 STM32U5xx USART LL API。使用用于优化用途（性能和大小）的 LL 单一服务功能执行外设初始化。	-	MX	-
		USART_Communication_Tx_Init	如何配置 GPIO 和 USART 外设，从而以轮询模式向超级终端（PC）异步发送字符。如果无法在指定时间内完成传输，则可通过超时（具有超时错误代码）退出流程。该示例基于 STM32U5xx USART LL API。使用用于优化用途（性能和大小）的 LL 单一服务功能执行外设初始化。	-	MX	-
		USART_Communication_Tx_VCP_Init	该示例展示了如何配置 GPIO 和 USART 外设，从而以轮询模式向超级终端（PC）异步发送字符。如果无法在指定时间内完成传输，则可通过超时（具有超时错误代码）退出流程。该示例基于 STM32U5xx USART LL API。使用用于优化用途（性能和大小）的 LL 单一服务功能执行外设初始化。	-	MX	-
	UTILS	UTILS_ConfigureSystemClock	如何通过 UTILS LL API，以 HSI 作为源时钟使用 PLL 配置系统时钟。	-	MX	-
		UTILS_ReadDeviceInfo	该示例读取 UID、设备 ID 和版本 ID，并将其保存到通用信息缓冲区中。	-	MX	-
	WWDG	WWDG_RefreshUntilUserEvent_Init	如何配置 WWDG，以便定期更新计数器并在按下用户按钮时生成 MCU WWDG 复位。使用用于优化用途（性能和大小）的 LL 单一服务功能初始化外设。	-	MX	-
examples_ll 的总数： 41				0	41	0
Examples_MIX	ADC	ADC_SingleConversion_TriggerSW_IT	如何在每次软件启动时使用 ADC 转换单个通道。使用中断编程模型执行转换。	-	MX	-
	CRC	CRC_PolynomialUpdate	如何通过 STM32U5xx CRC HAL 和 LL API 使用 CRC 外设。	-	MX	-
	SPI	SPI_FullDuplex_ComPolling_Master	通过 SPI 以轮询模式处理两个板件之间的数据缓冲发送/接收。	-	MX	-
		SPI_FullDuplex_ComPolling_Slave	通过 SPI 以轮询模式处理两个板件之间的数据缓冲发送/接收。	-	MX	-



级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-IOT02A
Examples_MIX	TIM	TIM_PWMInput	如何使用 TIM 外设测量外部信号频率和占空比。	-	<b>MX</b>	-
	examples_mix 的总数: 5			0	5	0
应用	-	OpenBootloader	该应用会利用 OpenBootloader 中间件来演示如何开发、使用 IAP 应用。	-	-	<b>X</b>
		SBSFU	SBSFU 提供可信根解决方案，该解决方案包含安全启动和安全固件升级功能。在执行应用之前使用，并提供了安全服务的示例（GPIO 切换），安全服务与非安全应用相互隔离但可以在运行时间被非安全应用使用。	-	-	<b>X</b>
		TFM	TFM 提供可信根解决方案，该解决方案包含安全启动和安全固件升级功能。在执行应用之前使用，并提供 TFM 安全服务，安全服务与非安全应用相互隔离但可以在运行时间被非安全应用使用。	-	-	<b>X</b>
	BLE	BLE_AT_Client	该示例演示了 B-U585I-IOT02A 板的 STM32WB5M 模块上的 BLE 连接。	-	-	<b>X</b>
	FileX	Fx_Dual_Instance	该应用提供了两个存储介质的工作示例，这两个存储介质由 STM32U575I-EV 板上运行的 Azure® RTOS FileX/Azure® RTOS LevelX 的两个独立实例进行管理。	<b>MX</b>	-	-
		Fx_MultiAccess	该应用提供了在 STM32U575I-EV 板上使用 Azure® RTOS FileX 栈的示例。其中演示了 FileX 并发文件访问功能。设计该应用的目的是在 SD 卡设备上执行文件操作，代码提供了处理 SD 卡 I/O 操作所需的软件代码。	<b>MX</b>	-	-
		Fx_NoR_Write_Read_File	该应用提供了在 B-U585I-IOT02A 板上使用 Azure® RTOS FileX 和 Azure® RTOS LevelX 栈的示例。其中演示了如何在 NOR Flash 存储器上使用 FileX 和 LevelX 创建 FAT 文件系统。设计该应用的目的是在 MX25LM51245G NOR Flash 设备上执行文件操作，代码提供了正常完成任务所需的软件代码。	-	-	<b>MX</b>
		Fx_SRAM_File_Edit_Standalone	该应用提供了以独立模式（无 Azure® RTOS ThreadX）运行时在 NUCLEO-U575ZI-Q 板上使用 Azure® RTOS FileX 栈的示例。其中演示了如何使用 FileX API 在内部 SRAM 上创建 Fat 文件系统。	-	<b>X</b>	-
		Fx_uSD_File_Edit	该应用提供了在 STM32U575I-EV 板上使用 Azure® RTOS FileX 栈的示例，其中展示了如何开发基础 SD 卡文件操作应用。	<b>MX</b>	-	-
	LPBAM	LPBAM_ADC_InternalExternalChannelSwitch	如何处理内部和外部通道配置之间的 ADC 开关，然后在低功耗模式下利用 DMA 链表特性转换数据。该应用使用 LPBAM 实用程序。	-	<b>X</b>	-
		LPBAM_COMP_InputSwitch	如何在低功耗模式下使用 DMA 链表特性处理 COMP 开关输入并读取比较值。该应用使用 LPBAM 实用程序。	-	<b>X</b>	-
		LPBAM_DAC_OPAMP_ContinuousConversion	如何在低功耗模式下使用 DMA 链表特性处理 DAC 连续转换和 OPAMP 开关配置。该应用使用 LPBAM 实用程序。	-	<b>X</b>	-
		LPBAM_DMA_MultiQueueExecution	如何在低功耗模式下使用 DMA 链表特性处理多队列执行。该应用使用 LPBAM 实用程序。	-	<b>X</b>	-

级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-H0T02A
应用	LPBAM	LPBAM_I2C_SequentialTransfer	如何在低功耗模式下使用 DMA 链表特性处理两个板件之间具有数据重载的 I2C 顺序发送/接收。该应用使用 LPBAM 实用程序。	-	X	-
		LPBAM_LPGPIO_IOToggle	如何在低功耗模式下通过 DMA 链表特性并使用 LPBAM 实用程序每秒切换一次 LPGPIO 引脚。	-	X	-
		LPBAM_LPTIM_PWMGeneration	如何在低功耗模式下使用 DMA 链表特性生成 LPTIM PWM。该应用使用 LPBAM 实用程序。	-	X	-
		LPBAM_LPUART_TransmitReceive	如何在低功耗模式下使用 DMA 链表特性处理两个板件之间的 LPUART 发送/接收。该应用使用 LPBAM 实用程序。	-	X	-
		LPBAM_SPI_FullDuplex_Simplex	如何在低功耗模式下使用 DMA 链表特性处理两个板件之间的两次连续 SPI 通信（全双工然后单工）。该应用使用 LPBAM 实用程序。	-	X	-
	NetXDuo	Nx_MQTT_Client	该应用提供了使用 Azure® RTOS NetX/NetXDuo 栈的示例。	-	-	MX
		Nx_SNTP_Client	该应用提供了使用 Azure® RTOS NetX/NetXDuo 栈的示例。	-	-	MX
		Nx_TCP_Echo_Client	该应用提供了使用 Azure® RTOS NetX/NetXDuo 栈的示例。	-	-	MX
		Nx_TCP_Echo_Server	该应用提供了使用 Azure® RTOS NetX/NetXDuo 栈的示例。	-	-	MX
		Nx_UDP_Echo_Client	该应用提供了使用 Azure® RTOS NetX/NetXDuo 栈的示例。	-	-	MX
		Nx_UDP_Echo_Server	该应用提供了使用 Azure® RTOS NetX/NetXDuo 栈的示例。	-	-	MX
		Nx_WebServer	该应用提供了使用 Azure® RTOS NetX/NetXDuo 栈的示例。	-	-	MX
	ThreadX	Tx_CMSIS_Wrapper	该应用提供了 Azure® RTOS ThreadX 的 CMSIS RTOS 适配层的示例。其中展示了如何使用 CMSIS RTOS 2 API 开发应用。	-	-	X
		Tx_FreeRTOS_Wrapper	该应用提供了使用 Azure® RTOS ThreadX 栈的示例。其中展示了如何使用 ThreadX 的 FreeRTOS 适配层开发应用。	-	X	-
		Tx_LowPower	该应用提供了使用 Azure® RTOS ThreadX 栈的示例。其中展示了如何在低功耗模式下使用 ThreadX 开发应用。	-	MX	-

级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-H0T02A
应用	ThreadX	Tx_MPU	该应用提供了使用 Azure® RTOS ThreadX 栈的示例。其中展示了如何使用 ThreadX 模块特性开发应用。	-	X	-
		Tx_SecureLEDToggle_TrustZone	该应用提供了使用 Azure® RTOS ThreadX 栈的示例。其中展示了如何在 TrustZone®特性启用（TZEN = 1）的情况下使用 ThreadX 开发应用。	-	MX	-
		Tx_Thread_Creation	该应用提供了使用 Azure® RTOS ThreadX 栈的示例。其中展示了如何使用 ThreadX 线程管理 API 开发应用。	-	MX	-
		Tx_Thread_MsgQueue	该应用提供了使用 Azure® RTOS ThreadX 栈的示例。其中展示了如何使用 ThreadX 消息队列 API 开发应用。	-	MX	-
		Tx_Thread_Sync	该应用提供了使用 Azure® RTOS ThreadX 栈的示例。其中展示了如何使用 ThreadX 同步 API 开发应用。	-	MX	-
	USBPD	USBPD_SNK_UX_Device_HID_CDC_ACM	该应用是 STM32U585xx 板上使用 Azure® RTOS USBX 栈的 USB-PD Type-C™ 用电设备兼 USB 设备。	-	-	X
		USBPD_SRC_UX_Host_MSC	该应用是使用 Azure® RTOS USBX 栈的 USB-PD Type-C™ 供电设备兼 USB 主机。其中展示了如何开发 USB 主机应用中的 USB-PD Type-C™ 供电设备，该 USB 主机应用基于批量存储（MSC），能够枚举并与可移动 USB Flash 盘通信。	X	-	-
	USBX	Ux_Device_CDC_ACM	该应用提供了在 NUCLEO-U575ZI-Q 板上使用 Azure® RTOS USBX 栈的示例，其中展示了如何开发基于 CDC_ACM 类 USB 通信的应用。	-	MX	-
		Ux_Device_CDC_ECM	该应用提供了在 STM32U575I-EV 板上使用 Azure® RTOS CDC_ECM 栈的示例。其中展示了如何通过 USB 接口运行基于 Web HTTP 服务器的应用栈。该应用通过使用 CDC_ECM 类的 USB 接口，使用 Web HTTP 服务器加载 SD 卡中存储的文件和网页。代码提供了构建合格的 Web HTTP 服务器所需的所有特性。主入口函数 tx_application_define() 在内核启动过程中被 ThreadX 调用。在该阶段，USBX 通过 USBx 类（CDC_ECM）初始化网络层。此外，初始化 FileX 和 NetXDuo 系统，创建并配置 NX_IP 实例和 Web HTTP 服务器。然后，应用创建两个主线程，usb_x_app_thread_entry (Prio: 10; PreemptionPrio: 10) 用于初始化 USB OTG HAL PCD 驱动程序和启动设备。	MX	-	-
		Ux_Device_DFU	该应用提供了在 STM32U585xx 板上使用 Azure® RTOS USBX 栈的示例。其中展示了如何开发基于 USB 设备固件升级（DFU）的应用。	-	-	MX
		Ux_Device_HID	该应用提供了在 STM32U585xx 板上使用 Azure® RTOS USBX 栈的示例。其中展示了如何开发基于 USB 设备人机接口（HID）鼠标的应用。	-	-	MX
		Ux_Device_HID_CDC_ACM	该应用提供了在 STM32U585xx 板上使用 Azure® RTOS USBX 栈的示例。其中展示了如何开发基于复合 USB 设备通信类 HID 和 CDC_ACM 的应用。	-	-	MX
		Ux_Device_MSC	该应用提供了在 STM32U575I-EV 板上使用 Azure® RTOS USBX 栈的示例。其中展示了如何开发基于 USB 设备批量存储类的应用。	MX	-	-



级别	模块名称	项目名称	说明	STM32U575I-EV	NUCLEO-U575ZI-Q	B-U585I-IOT02A
应用	USBX	Ux_Host_CDC_ACM	该应用提供了使用 Azure® RTOS USBX 栈的示例。	MX	-	-
		Ux_Host_DualClass	该应用提供了使用 Azure® RTOS USBX 栈的示例。	MX	-	-
		Ux_Host_HID	该应用提供了使用 Azure® RTOS USBX 栈的示例。	MX	-	-
		Ux_Host_MSC	该应用提供了使用 Azure® RTOS USBX 栈的示例。其中展示了如何开发能够枚举并与可移动 USB Flash 盘通信的 USB 主机批量存储（MSC）。	MX	-	-
	WiFi	WiFiBasics	该应用演示了 B-U585I-IOT02A 板的 MXCHIP EMW3080 模块上的 Wi-Fi 连接。	-	-	MX
	应用总数：46			10	18	18
演示	-	演示	STM32Cube 演示平台作为基于 STM32Cube 的固件包，提供全套基于模块化架构的软件组件。所有模块都可以在独立应用中单独重复使用。所有这些模块均由 STM32Cube 演示内核进行管理，演示内核允许动态添加新模块和访问共有资源（存储、存储器管理和实时操作系统）。STM32Cube 演示平台围绕基础 GUI 界面进行构建，以 STM32Cube HAL、BSP 和若干中间件组件为基础。	X	-	-
		IOT_HTTP_WebServer	STM32Cube 演示平台是基于 STM32Cube 的固件包，以 STM32Cube HAL、BSP 和中间件组件为基础。其中展示了如何使用 MXCHIP Wi-Fi 模块执行 Web 服务器演示。	-	-	X
	演示总数：2			1	0	1
项目总数：239				40	154	45



## 版本历史

表 2. 文档版本历史

日期	版本	变更
2021 年 9 月 9 日	1	初始版本

## 目录

1	参考文档.....	2
2	STM32CubeU5 示例 .....	3
	版本历史 .....	21
	目录 .....	22
	表一览 .....	23

## 表一览

表 1.	STM32CubeU5 固件示例 .....	5
表 2.	文档版本历史 .....	21

**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2021 STMicroelectronics – All rights reserved