
STM32F4DISCOVERY 外设固件示例

前言

本应用笔记介绍了为 STM32F4DISCOVERY 套件提供的一些外设固件示例。

这些示例可直接使用，可帮助用户快速了解 STM32F4xx 外设和 STM32F4DISCOVERY 板硬件。每个示例都配备了一些预配置项目，其中涵盖 EWARM、MDK-ARM、TrueSTUDIO 和 TASKING 工具。

可以从 www.st.com/stm32f4-discovery 下载固件应用程序软件包，其中包含了这些示例。

建议用户首先阅读 STM32F4DISCOVERY 套件的软件和固件环境入门 (UM1467) 以熟悉 STM32F4DISCOVERY 套件。

目录

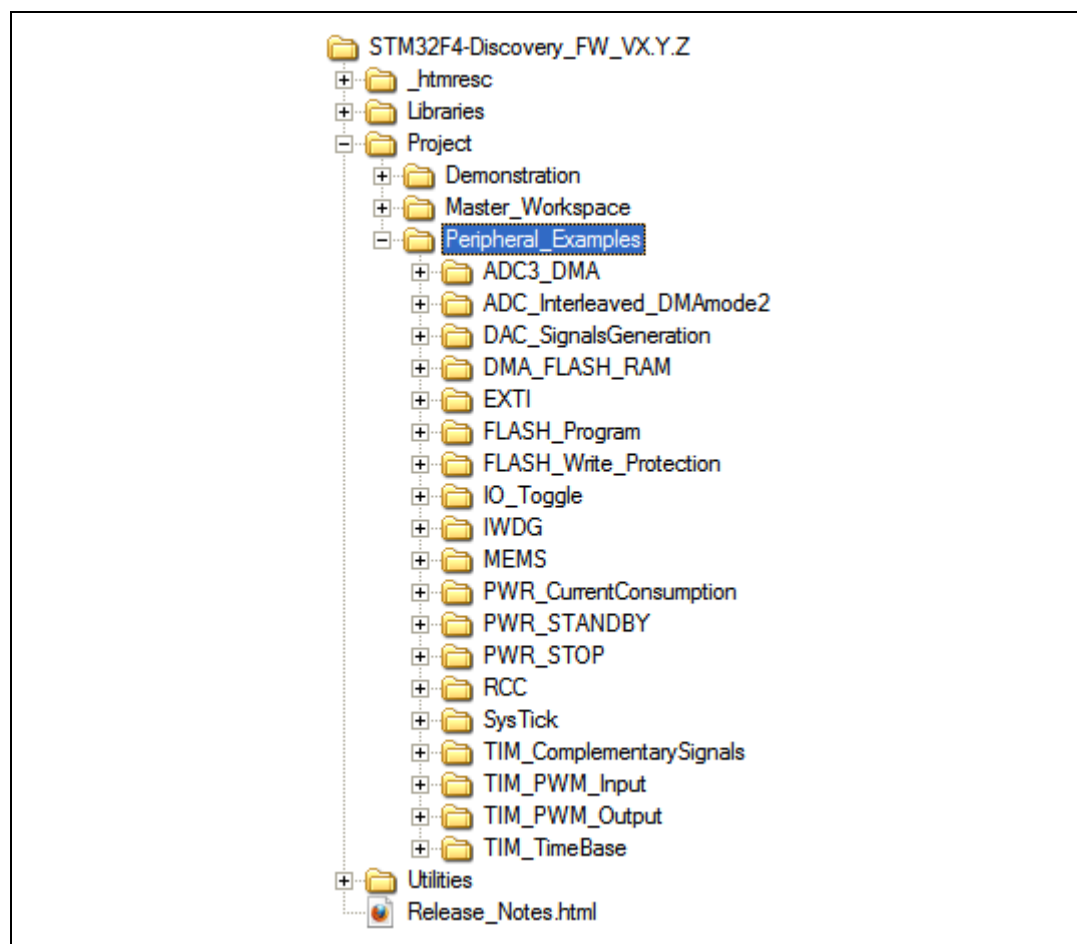
1	外设固件示例结构概述	4
2	外设固件示例说明	5
2.1	GPIO 切换示例	5
2.1.1	目的	5
2.1.2	说明	5
2.2	EXTI 示例	5
2.2.1	目的	5
2.2.2	说明	5
2.3	SysTick 示例	6
2.3.1	目的	6
2.3.2	说明	6
2.4	待机模式示例	6
2.4.1	目的	6
2.4.2	说明	6
2.5	停止模式示例	7
2.5.1	目的	7
2.5.2	说明	7
2.6	电流消耗示例	7
2.6.1	目的	7
2.6.2	说明	8
2.7	Flash 程序示例	9
2.7.1	目的	9
2.7.2	说明	9
2.8	Flash 写保护示例	9
2.8.1	目的	9
2.8.2	说明	9
2.9	IWDG（独立看门狗）示例	10
2.9.1	目的	10
2.9.2	说明	10
2.10	ADC 交替 DMA 模式 2 示例	10
2.10.1	目的	10
2.10.2	说明	10

2.11	ADC DMA 示例	11
2.11.1	目的	11
2.11.2	说明	11
2.12	MEMS 加速计 (LIS302DL) 示例	11
2.12.1	目的	11
2.12.2	说明	11
2.13	RCC（复位和时钟控制）示例	12
2.13.1	目的	12
2.13.2	说明	12
2.14	DMA Flash RAM 示例	12
2.14.1	目的	12
2.14.2	说明	12
2.15	DAC 信号生成示例	13
2.15.1	目的	13
2.15.2	说明	13
2.16	TIM 互补信号示例	13
2.16.1	目的	13
2.16.2	说明	13
2.17	TIM 时基示例	14
2.17.1	目的	14
2.17.2	说明	14
2.18	TIM PWM 输入示例	15
2.18.1	目的	15
2.18.2	说明	15
2.19	TIM PWM 输出示例	15
2.19.1	目的	15
2.19.2	说明	15
3	版本历史	17

1 外设固件示例结构概述

在 STM32F4DISCOVERY 固件应用程序软件包中提供了一些外围固件示例，这些示例位于 \Project 文件夹下，如 [图 1](#) 中所示。

图 1. 硬件环境



1. VX.Y.Z 表示软件包版本，例如 V1.0.0。

要运行示例，请使用首选工具打开项目，然后编译，再加载和运行项目。

某些示例可能需要额外硬件，如示波器。有关必需硬件的更多详细信息，请参见每个示例中提供的自述文件。

2 外设固件示例说明

2.1 GPIO 切换示例

2.1.1 目的

本示例说明如何使用 GPIO 端口位置位/复位寄存器 (BSRRL 和 BSRRH) 实现 I/O 切换。

2.1.2 说明

在本示例中：

- 使能 GPIOD 时钟。
- 配置 GPIOD 引脚 12、13、14 和 15。
- 在 while 循环中，通过将端口位置位/复位寄存器 (BSRRL) 中的相应位置 1，使 GPIOD 输出数据寄存器 (ODR) 中的 ODR12、ODR13、ODR14 和 ODR15 位置 1。然后，通过将端口位复位寄存器 (BSRRH) 中的相应位置 1，使 GPIOD 输出数据寄存器 (ODR) 中的 ODR12、ODR13、ODR14 和 ODR15 位复位。
- GPIOD ODR12、ODR13、ODR14 和 ODR15 位的置位与复位操作之间插入了一段延时。

执行程序时，LD3、LD4、LD5 和 LD6 这四个 LED 轮流闪亮。点亮与熄灭状态之间的持续时间即相当于插入的延时。

2.2 EXTI 示例

2.2.1 目的

本示例说明如何配置外部中断线。

2.2.2 说明

在本示例中：

- PA0 引脚配置为输入悬空。
- PA0 用作外部中断源线 0 (EXTI0)。
- EXTI 线 0 配置为在 PA0 引脚上每检测到一个上升沿时生成一个中断。每次按 User（用户）按钮时都会触发中断。
- 在 NVIC（嵌套向量中断控制器）中，配置了 EXTI 线 0 中断优先级并且使能了中断。

执行程序时，若用户按下 User（用户）按钮（EXTI0 中断程序），将对连接到 PD12 的 LD4 LED 进行切换。

2.3 SysTick 示例

2.3.1 目的

本示例说明如何配置系统节拍定时器及如何使用该定时器生成 1 ms 的时基。

2.3.2 说明

在本示例中：

- 系统节拍定时器初始化。
- 系统节拍定时器中断在 NVIC 中使能。
- 系统节拍定时器/计数器以自由运行模式启动以生成周期中断。
- 每隔 1 ms 触发一次系统节拍定时器中断。
- 根据系统节拍定时器计数结束事件，实现延迟函数。

LD3、LD4、LD5 和 LD6 这四个 LED 通过延迟函数定义的时序进行切换。

2.4 待机模式示例

2.4.1 目的

本示例说明如何将系统置于待机模式，以及如何使用外部复位、RTC 闹钟 A 或 WKUP 引脚将系统从该模式中唤醒。

2.4.2 说明

在本示例中：

- 系统节拍定时器初始化。
- 系统节拍定时器中断在 NVIC 中使能。
- 系统节拍定时器/计数器以自由运行模式启动以生成周期中断。系统节拍定时器中断每 250 ms 触发一次。LD4 将切换，指示 MCU 处于待机模式还是运行模式。
- EXTI 线 1 配置为在 PA1 引脚上每检测到一个上升/下降沿时生成一个中断。每次 PA1 电平发生更改（GND 或 VDD）时，都将生成外部中断。
 - 在 EXTI 行上检测到下降或上升沿时，将生成一个中断。在 EXTI 处理器程序中，RTC 配置为在 5 秒内生成闹钟事件，之后系统将进入待机模式，使 LD4 停止切换。
 - WKUP 引脚（User（用户）按钮）出现上升沿时或执行外部复位时都会将系统从待机模式唤醒。如果 5 秒内既没有在 WKUP 引脚（User（用户）按钮）上出现上升沿，也没有生成外部复位，则 RTC 闹钟 A 会将系统唤醒。

从待机模式中唤醒后，将按照复位后的方式重新执行程序，RTC 配置（时钟源和预分频器）得到保留，LD4 再次切换。因此，无需配置 RTC。

LD4 和 LD3 这两个 LED 按如下方式监视系统状态：

- LD3 亮：RTC 配置失败（系统将进入无限循环）
- LD4 切换：系统处于运行模式
- LD4 灭：系统处于待机模式

这些步骤以无限循环方式重复。

2.5 停止模式示例

2.5.1 目的

本示例说明如何将系统置于停机模式，以及如何使用与 EXTI 线 22 或 EXTI 线 0 相连的 RTC 唤醒定时器事件将系统从该模式唤醒。

2.5.2 说明

在本示例中：

- EXTI 线 0 配置为出现上升沿时生成中断。
- EXTI 线 22 在内部与 RTC 唤醒事件相连，并且配置为每 4 s 于上升沿生成一个中断。
- SysTick 编程为每 250 ms 生成一个中断。在 SysTick 中断处理器中，将切换 LD3，用以指示 MCU 是处于停机模式还是运行模式。

系统进入停机模式，等待每 4 秒生成一个 RTC 唤醒事件或用户按下 User（用户）按钮。

- 如果通过 RTC 唤醒事件 (EXTI_Line22) 从停机模式唤醒系统，则将切换 LD4。
- 如果通过按下 User（用户）按钮 (EXTI_Line0) 从停机模式唤醒系统，则将切换 LD6。

LED 用于监视系统状态：

- LD3 切换：系统处于运行模式。
- LD4 切换：使用 RTC 唤醒中断从停机模式唤醒系统。
- LD6 切换：使用 EXTI 线 0（User（用户）按钮）从停机模式唤醒系统。

2.6 电流消耗示例

2.6.1 目的

本示例说明如何配置 STM32F4xx 系统以测量不同低功耗模式的电流消耗。低功耗模式为：

- 睡眠模式
- 带有 RTC 的停机模式
- 不带 RTC 和 BKPSRAM 的待机模式
- 带有 RTC 的待机模式
- 带有 RTC 和 BKPSRAM 的待机模式

要选择待测量的低功耗模式，请取消注释 stm32f4xx_lp_modes.h 文件中的相应行。

- 注：
- 1 通过移除跳线 JP1（标记为 IDD）并连接一个电流表，便可以在 STM32F4DISCOVERY 板上测量 STM32F4xx 功耗。
 - 2 在 STM32F4DISCOVERY 板上，由于 BOOT0 引脚连接了一个 R31 电阻，因此功耗将增大 500uA 左右。移除该电阻即可得到正确的功耗值。

2.6.2 说明

复位后，程序将等待用户按下与 PA.00 相连的 User（用户）按钮，从而进入所选的低功耗模式。

如果低功耗模式配置中未使用 RTC，可重新按下 User（用户）按钮退出低功耗模式。

使用 RTC 时，将由 RTC 自动生成从低功耗模式的唤醒事件（4 秒后）。

不同低功耗模式配置为：

睡眠模式

- 系统按 PLL (168 MHz) 运行
- Flash 3 等待状态
- 从内部 Flash 运行代码
- 禁止所有外设
- 使用 EXTI 线（User（用户）按钮 PA.00）唤醒

停机模式

- 由 LSI 提供 RTC 时钟
- 调节器处于 LP 模式
- HSI、HSE 关闭 且（如果未用作 RTC 时钟源）LSI 关闭
- 无 IWDG
- 处于深度掉电模式的 Flash
- 使用由 LSI 提供时钟的 RTC 自动唤醒（约 20 s 后）

待机模式

- 备份 SRAM 和 RTC 关闭
- IWDG 和 LSI 关闭
- 使用唤醒引脚 (PA.00) 唤醒

带有 RTC（由 LSI 提供时钟信号）的待机模式

- 由 LSI 提供 RTC 时钟
- IWDG 和 LSI 均关闭（如果未用作 RTC 时钟源）
- 备份 SRAM 关闭
- 使用由 LSI 提供时钟的 RTC 自动唤醒（约 20 s 后）

带有 RTC（由 LSI 提供时钟信号）和 BKPSRAM 的待机模式

- 由 LSI 提供 RTC 时钟
- 备份 SRAM 开启
- IWDG 关闭
- 使用由 LSI 提供时钟的 RTC 自动唤醒（约 20 s 后）

2.7 Flash 程序示例

2.7.1 目的

本示例说明如何对 STM32F4xx 内部 Flash 进行编程。

2.7.2 说明

在本示例中：

- 在复位后，将锁定 Flash 程序/擦除控制器。FLASH_Unlock 函数用于将其解锁。
- 在对所需地址进行编程之前，将使用 Flash 擦除扇区功能来执行擦除操作。擦除过程首先计算要使用的扇区数量。将通过调用 FLASH_EraseSector 函数来逐一擦除这些扇区。
- 将使用 FLASH_ProgramWord 函数来执行编程操作。随后将检查写入的数据并且编程操作的结果将存储在 MemoryProgramStatus 变量中。

2.8 Flash 写保护示例

2.8.1 目的

本示例说明如何使能和禁止 STM32F4xx Flash 的写保护功能。

2.8.2 说明

在本示例中：

- 复位时通过一直按下 User（用户）按钮，程序将检查 FLASH_WRP_SECTORS（在 main.c 中定义）的写保护状态。
 - 如果 FLASH_WRP_SECTORS 受到写保护，则禁止写保护。如果正确执行了禁止操作，LD6 将点亮。否则 LD5 点亮。
 - 如果 FLASH_WRP_SECTORS 未受到写保护，则使能写保护。如果正确执行了禁止操作，LD4 将点亮。否则 LD5 点亮。
- 如果复位后没有按下 User（用户）按钮，程序点亮 LD3。

2.9 IWDG（独立看门狗）示例

2.9.1 目的

本示例说明如何定期更新 IWDG 重载计数器，以及如何模拟软件故障，使编程好的时间段到期时生成 MCU IWDG 复位。

2.9.2 说明

在本示例中：

- 独立看门狗超时设置为 250 ms。
- 系统节拍配置为每 250 ms 生成一个中断。
- 在系统节拍中断服务程序中，将重载独立看门狗计数器以防止独立看门狗复位，并将切换 LD3。
- 连接到 PA0 引脚的 EXTI 线 0 配置为在其上升沿生成中断。
- 在 NVIC 中，将使能 EXTI 线 0 对应中断向量且优先级等于 0，并且 SysTick 中断向量优先级设置为 1（EXTI 中断优先于 SysTick 中断）。
- EXTI 线用于模拟固件故障：当触发 EXTI 线事件时（在 STM32F4DISCOVERY 板上按下 User（用户）按钮后），将提供对应的中断。在 ISR 中，LD6 熄灭并且 EXTI 线挂起位未清零。CPU 无限期执行 EXTI 线 ISR 并且永远不会进入系统节拍中断程序，因此不会重载独立看门狗计数器。因此，当独立看门狗计数器达到 00 时，独立看门狗将生成一个复位。

当程序在运行并且生成了独立看门狗复位后，LD4 将在系统恢复操作之后点亮。

2.10 ADC 交替 DMA 模式 2 示例

2.10.1 目的

本示例简要介绍了如何使用 ADC 外设以速度为 7.2 Msps 的 DMA 模式 2 的三重交替模式下转换常规通道。

2.10.2 说明

本示例将生成三个 DMA 请求：

- 传输 ADC2 和 ADC1 数据（ADC2 数据占用高位半字，ADC1 数据占用低位半字）。
- 传输 ADC1 和 ADC3 数据（ADC1 数据占用高位半字，ADC3 数据占用低位半字）。
- 传输 ADC3 和 ADC2 数据（ADC3 数据占用高位半字，ADC2 数据占用低位半字），依此类推。

每发送一个 DMA 请求（两个数据项可用），就会以字的形式传输表示两个 ADC 转换数据项的两个半字。

每当有 2 个数据项可用时，就会生成一个 DMA 请求：

- 第 1 个请求：ADC_CDR[31:0] = (ADC2_DR[15:0] << 16) | ADC1_DR[15:0]（第 1 步）
- 第 2 个请求：ADC_CDR[31:0] = (ADC1_DR[15:0] << 16) | ADC3_DR[15:0]（第 2 步）
- 第 3 个请求：ADC_CDR[31:0] = (ADC3_DR[15:0] << 16) | ADC2_DR[15:0]（第 3 步）
- 第 4 个请求：ADC_CDR[31:0] = (ADC2_DR[15:0] << 16) | ADC1_DR[15:0]（第 1 步）
- 依此类推。

转换由软件触发。

ADC1、ADC2 和 ADC3 配置为转换 ADC 通道 12。这样，ADC 便能达到 7.2 Msps。每五个周期对同一通道转换一次。本示例中，系统时钟频率为 144 MHz，APB2 = 72 MHz 且 ADC 时钟 = APB2/2。

由于 ADCCLK= 36 MHz 且转换速率 = 五个周期，因此转换时间 = 36 MHz/5cyc = 7.2 Msps。

2.11 ADC DMA 示例

注： 将外部信号（介于 0 V 到 3.3 V 之间）连接到要转换的 ADC3 引脚 (PC.02)。

2.11.1 目的

本示例介绍如何使用 ADC3 和 DMA 将连续转换的数据从 ADC3 传输到存储器中。

2.11.2 说明

在本示例中：

- ADC3 配置为连续转换通道 7。
- 每次转换结束后，DMA 都以循环模式将转换的数据从 ADC3 DR 寄存器传输到 ADC3ConvertedValue 变量中。
- 系统时钟频率等于 144 MHz，APB2 时钟频率等于 72 MHz 且 ADC 时钟频率等于 APB2/2。
- 由于 ADC3 时钟频率为 36 MHz 且采样时间设为 3 个周期，转换为 12 位数据的时间为 12 个周期，因此总转换时间为 $(12+3)/36 = 0.41 \text{ us}$ (2.4 Msps)。

2.12 MEMS 加速计 (LIS302DL) 示例

2.12.1 目的

本示例说明如何配置 MEMS 加速计，以检测 X/Y 轴上的加速度及 Z 轴上的单击/双击操作。

2.12.2 说明

启动后，程序将检查 MEMS 加速计状态寄存器，具体操作如下：

- 如果开发板移动，则检测 X/Y 轴上的加速度，并根据运动方向和速度切换相应的 LED。
- 如果在 Z 轴上检测到单击操作，则在 3 S 内切换 LED3 和 LED6。
- 如果在 Z 轴上检测到双击操作，则在 3 S 内切换所有 LED。

LED 是否点亮由 TIM4 捕捉比较通道控制。

2.13 RCC（复位和时钟控制）示例

2.13.1 目的

本示例说明如何执行以下操作：

- 将 HSE（高速时钟）配置为 RCC 时钟
- 使用时钟安全系统 (CSS) 功能生成 NMI 中断
- 在 MCO2 上输出系统时钟

2.13.2 说明

出于调试目的，RCC_GetClocksFreq() 函数用于检索不同片上时钟的当前状态和频率。

您可以使用工具调试器来查看 RCC_ClockFreq 结构内容（其中保存了不同片上时钟的频率）。

本示例还介绍了高速外部时钟 (HSE) 故障检测：当 HSE 时钟消失时（断路或者与外部石英时钟断开连接），将禁止 HSE 和 PLL（但是不会对 PLL 配置进行任何更改），HSI 选为系统时钟源并生成中断 (NMI)。在 NMI ISR 中，HSE 和 HSE 就绪中断使能，一旦 HSE 时钟恢复，立即生成 HSERDY 中断；在 RCC ISR 程序中，将系统时钟重新配置为 HSE 时钟发生故障之前的状态。可以在 MCO2 引脚 (PC9) 上监视系统时钟。

四个 LED 按照延迟函数定义的时序进行切换。

2.14 DMA Flash RAM 示例

2.14.1 目的

本示例介绍如何使用 DMA 通道将字数据缓冲区从 Flash 传输到嵌入式 SRAM 存储器。

2.14.2 说明

DMA2 数据流 0 通道 0 配置为将 Flash 中存储的 32 字数据缓冲区的内容传输到 RAM 中声明的接收缓冲区。

传输的启动由软件触发。将使能 DMA2 数据流 0 通道 0 存储器到存储器传输。还将使能源地址和目标地址的递增。

通过将 DMA2 数据流 0 通道 0 的通道使能位置 1 来启动传输。在传输结束时，将生成传输完成中断，因为已使能该中断。然后将传输完成中断挂起位清零。

DMA 传输完成后，硬件将禁止 DMA 数据流。

主应用程序可以检查数据流使能状态，以此检测传输是否结束，也可以检查剩余传输量，传输结束时该值应等于 0。

还将比较源缓冲区与目标缓冲区，以检查是否所有数据都已正确传输。

可使用 STM32 评估板上的 LED 来监视传输状态：

- 程序启动时 LD4 点亮。
- 配置阶段结束后启动传输时 LD3 点亮。
- 传输完成时（进入传输完成中断程序）LD5 点亮。
- 源缓冲区与目标缓冲区之间的比较结果通过时 LD6 点亮。

修改 main.h 文件中的定义值，即可选择在 DMA 传输示例中使用其它数据流和/或通道。

注： 只有 DMA2 数据流可以执行存储器到存储器的传输。

可通过多种方法检查 DMA 传输是否结束：

- 使用 DMA 传输完成中断。
- 使用 DMA 使能状态（传输完成时，硬件将禁止 DMA 数据流）。
- 使用 DMA 数据流传输计数器值（传输正在执行时计数器值递减，传输结束后计数器值等于 0）。
- 使用 DMA 传输完成标志（轮询模式）。
- 本示例提供了方法 1、方法 2 和方法 3（取消 main.c 文件等待循环中相关代码的注释即可在方法 2 和方法 3 之间进行选择）。

2.15 DAC 信号生成示例

2.15.1 目的

本示例简要介绍了如何使用 DAC 外设通过 DMA 控制器来生成多个信号。

2.15.2 说明

用户按下 User（用户）按钮时，DMA 会将两个所选波形传输到 DAC。

每按一次 User（用户）按钮，便会选择两个信号，并且可在两个 DAC 通道中对此加以监视：

- 阶梯波形（通道 1）和正弦波形（通道 2）。
- 噪声波形（通道 1）和三角波形（通道 2）。

2.16 TIM 互补信号示例

2.16.1 目的

本示例说明如何配置 TIM1 外设以生成三个互补的 TIM1 信号，以分别用于插入定义的死区值，使用断路功能以及锁定所需参数。

2.16.2 说明

TIM1CLK 固定为 SystemCoreClock，TIM1 预分频器等于 0，因此使用的 TIM1 计数器时钟为 SystemCoreClock (168 MHz)。

目标是生成 17.57 kHz 的 PWM 信号：

$$\text{TIM1_Period} = (\text{SystemCoreClock} / 17570) - 1$$

三个占空比的计算方式如下：

- 通道 1 占空比设置为 50%，因此通道 1N 设置为 50%。
- 通道 2 占空比设置为 25%，因此通道 2N 设置为 75%。
- 通道 3 占空比设置为 12.5%，因此通道 3N 设置为 87.5%。

定时器脉冲的计算方式如下：

$$\text{ChannelxPulse} = \text{DutyCycle} * (\text{TIM1_Period} - 1) / 100$$

将在两个不同互补信号之间插入一个等于 $11/\text{SystemCoreClock}$ 的死区，并且选择锁定电平 1。
 出现高电平时使用断路极性。
 可以使用示波器来显示 TIM1 波形。

2.17 TIM 时基示例

2.17.1 目的

本示例说明如何在输出比较定时模式中配置 TIM 外设（包含每个通道的对应中断请求），以生成四个不同时基。

2.17.2 说明

为了使 TIM3 计数器时钟等于 500 kHz，需要将 TIM3CLK 频率设置为 $\text{SystemCoreClock} / 2$ (Hz)：

- 预分频器的计算公式如下：

$$\text{预分频器} = (\text{TIM3CLK} / \text{TIM3 计数器时钟}) - 1$$
- 将 SystemCoreClock 设置为 168 MHz
- TIM3 CC1 寄存器值等于 54618

$$\text{CC1 更新速率} = \text{TIM3 计数器时钟} / \text{CCR1_Val} = 9.154 \text{ Hz}, \text{ 因此 TIM3 通道 1 每 } 109.2 \text{ ms 生成一个中断}$$
- TIM3 CC2 寄存器等于 27309

$$\text{CC2 更新速率} = \text{TIM3 计数器时钟} / \text{CCR2_Val} = 18.31 \text{ Hz}, \text{ 因此 TIM3 通道 2 每 } 54.6 \text{ ms 生成一个中断}$$
- TIM3 CC3 寄存器等于 13654

$$\text{CC3 更新速率} = \text{TIM3 计数器时钟} / \text{CCR3_Val} = 36.62 \text{ Hz}, \text{ 因此 TIM3 通道 3 每 } 27.3 \text{ ms 生成一个中断}$$
- TIM3 CC4 寄存器等于 6826

$$\text{CC4 更新速率} = \text{TIM3 计数器时钟} / \text{CCR4_Val} = 73.25 \text{ Hz}, \text{ 因此 TIM3 通道 4 每 } 13.65 \text{ ms 生成一个中断。}$$

当计数器值达到输出比较寄存器值时，将生成输出比较中断，并且处理器程序中的四个引脚（PD.12、PD.13、PD.14 和 PD.15）分别按以下频率切换：

- PD.12: 4.57 Hz (CC1)
- PD.13: 9.15 Hz (CC2)
- PD.14: 18.31 Hz (CC3)
- PD.15: 36.62 Hz (CC4)

2.18 TIM PWM 输入示例

2.18.1 目的

本示例说明如何使用 TIM 外设来测量外部信号的频率和占空比。

2.18.2 说明

- TIMxCLK 频率设置为 SystemCoreClock/4 (Hz)。预分频器为 0，因此计数器时钟等于 SystemCoreClock/2 (Hz)。
- 对于 STM32F4xx 器件的版本 A，SystemCoreClock 设置为 168 MHz。
- TIM4 配置为 PWM 输入模式：外部信号连接到用作输入引脚的 TIM4 通道 2。

要测量频率和占空比，需使用 TIM4 CC2 中断请求，因此在 TIM4_IRQHandler 程序中计算外部信号的频率和占空比。

Frequency 变量包含外部信号频率：

- TIM4 计数器时钟 = SystemCoreClock / 2。
- Frequency = TIM4 计数器时钟 / TIM4_CCR2（以 Hz 为单位）。
- DutyCycle 变量包含外部信号占空比：
DutyCycle = (TIM4_CCR1*100)/(TIM4_CCR2)（百分比形式）。

要测量的最小频率值是 1280 Hz（TIM4 计数器时钟 / CCR 最大值）。

2.19 TIM PWM 输出示例

2.19.1 目的

本示例说明如何将 TIM 外配置为 PWM（脉冲宽度调制）模式。

2.19.2 说明

TIM3CLK 频率设置为 SystemCoreClock / 2 (Hz)。要使 TIM3 计数器时钟等于 28 MHz，需按以下公式计算预分频器：

- 预分频器 = (TIM3CLK / TIM3 计数器时钟) - 1
- 对于 STM32F4xx 器件的版本 A，SystemCoreClock 设置为 168 MHz。
- TIM3 以 42 kHz 的频率运行：
$$\text{TIM3 频率} = \text{TIM3 计数器时钟} / (\text{ARR} + 1) = 28 \text{ MHz} / 666 = 42 \text{ kHz}$$
- TIM3 CCR1 寄存器值等于 333，因此 TIM3 通道 1 生成一个频率等于 30 kHz 且占空比等于 50% 的 PWM 信号：
$$\text{TIM3 通道 1 占空比} = (\text{TIM3_CCR1} / \text{TIM3_ARR} + 1) * 100 = 50\%$$
- TIM3 CCR2 寄存器值等于 249，因此 TIM3 通道 2 生成一个频率等于 30 kHz 且占空比等于 37.5% 的 PWM 信号：
$$\text{TIM3 通道 2 占空比} = (\text{TIM3_CCR2} / \text{TIM3_ARR} + 1) * 100 = 37.5\%$$

- TIM3 CCR3 寄存器值等于 166，因此 TIM3 通道 3 生成一个频率等于 30 kHz 且占空比等于 25% 的 PWM 信号：

$$\text{TIM3 通道 3 占空比} = (\text{TIM3_CCR3} / \text{TIM3_ARR} + 1) * 100 = 25\%$$

- TIM3 CCR4 寄存器值等于 83，因此 TIM3 通道 4 生成一个频率等于 30 kHz 且占空比等于 12.5% 的 PWM 信号：

$$\text{TIM3 通道 4 占空比} = (\text{TIM3_CCR4} / \text{TIM3_ARR} + 1) * 100 = 12.5\%$$

可以使用示波器来显示 PWM 波形。

3 版本历史

表 1. 文档版本历史

日期	版本	变更
2011 年 09 月 16 日	1	初始版本。
2011 年 09 月 23 日	2	将 STM32F-Discovery 更改为 STM32F4DISCOVERY 更改了 第 4 页上的图 1: 硬件环境 在 第 6 页上的第 2.4 节: 待机模式示例中增加了数据 更新了 第 7 页上的第 2.5 节: 停止模式示例 (EXTI 线 22 或 EXTI 线 0) 在 第 10 页上的第 2.10 节: ADC 交替 DMA 模式 2 示例中增加了数据 更新了 第 11 页上的第 2.11 节: ADC DMA 示例 更新了 第 12 页上的第 2.14 节: DMA Flash RAM 示例

请仔细阅读：

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本文档中信息的提供仅与ST产品有关。意法半导体公司及其子公司（“ST”）保留随时对本文档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有ST产品均根据ST的销售条款出售。

买方自行负责对本文所述ST产品和服务的选择和使用，ST概不承担与选择或使用本文所述ST产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为ST授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在ST的销售条款中另有说明，否则，ST对ST产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且/或（D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此而导致的任何风险，意法半导体的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML或JAN正式认证产品适用于航天应用。

经销的ST产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致ST针对本文所述ST产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大ST的任何责任。

ST和ST徽标是ST在各个国家或地区的商标或注册商标。

本文档中的信息取代之前提供的所有信息。

ST徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

© 2014 STMicroelectronics 保留所有权利

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 - 菲律宾 - 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

www.st.com