STM32 定时器与 PWM

Lecture 6

第一部分
PWM 调光与呼吸灯

第二部分
时钟源 主模式 从模式

课程顺序：先完成任务，再学习协同

第一部分 独立完成 PWM 调光

1. PSC / CNT / ARR / CCR 与频率计算
2. 接线、CubeMX、启动 PWM
3. 三档亮度 → 呼吸灯 → 调试验收

第二部分 从官方结构图读懂主从

1. 时钟源与 TRGI 输入路径
2. MMS / TRGO → ITR / TS → SMS
3. 主从级联、外部计数与控制实验

LED 只亮一半，必须输出 1.65 V 吗？

直接使用GPIO

GPIO 输出高电平 → 点亮
GPIO 输出低电平 → 熄灭

改“电压高度”？

怎么做到

仍然只输出高 / 低电平
却让 LED 呈现不同亮度

还是改“点亮时间”？

软件在等，还是硬件在数？

HAL_Delay()

软件延时

CPU 在程序中主动等待
计时存在抖动，适合实验演示，不适合精确计时。
会阻塞CPU。

TIMx

硬件定时器

外设根据时钟自主计数
CPU 可以处理其他工作
配置后可持续生成 PWM

TIM简介

- TIM（Timer）定时器
- 定时器可以对输入的时钟进行计数，并在计数值达到设定值时触发中断
- 16位计数器、预分频器、自动重装寄存器的时基单元，在72MHz计数时钟下可以实现最大59.65s的定时
- 不仅具备基本的定时中断功能，而且还包含内外时钟源选择、输入捕获、输出比较、编码器接口、主从触发模式等多种功能
- 根据复杂度和应用场景分为了高级定时器、通用定时器、基本定时器三种类型

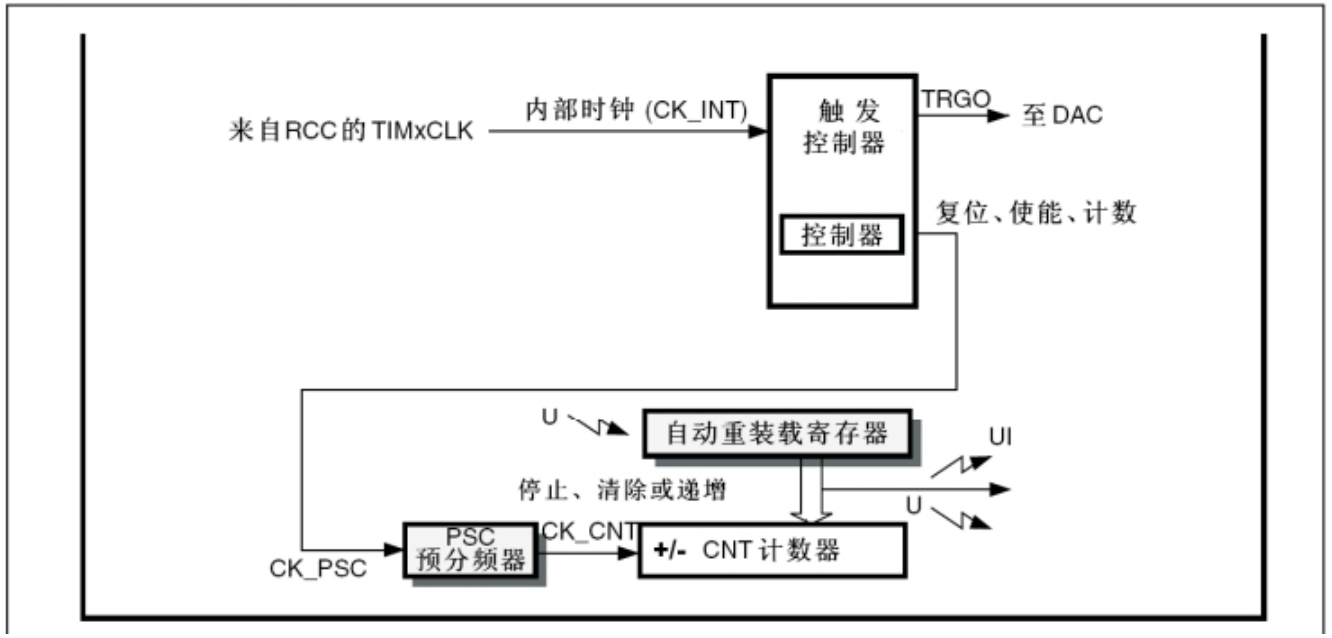
定时器类型

类型	编号	总线	功能
高级定时器	TIM1、TIM8	APB2	拥有通用定时器全部功能，并额外具有重复计数器、死区生成、互补输出、刹车输入等功能
通用定时器	TIM2、TIM3、TIM4、TIM5	APB1	拥有基本定时器全部功能，并额外具有内外时钟源选择、输入捕获、输出比较、编码器接口、主从触发模式等功能
基本定时器	TIM6、TIM7	APB1	拥有定时中断、主模式触发DAC的功能

- STM32F103C8T6定时器资源：TIM1、TIM2、TIM3、TIM4

基本定时器

图144　基本定时器框图



先掌握四个寄存器

PSC 预分频器

输入时钟按 PSC + 1 分频
本课：PSC = 71

CNT 计数器

每来一个计数节拍，向上加 1
本课：0 → 1 → … → 999

ARR 自动重装载寄存器

决定一个周期的计数上限
本课：ARR = 999

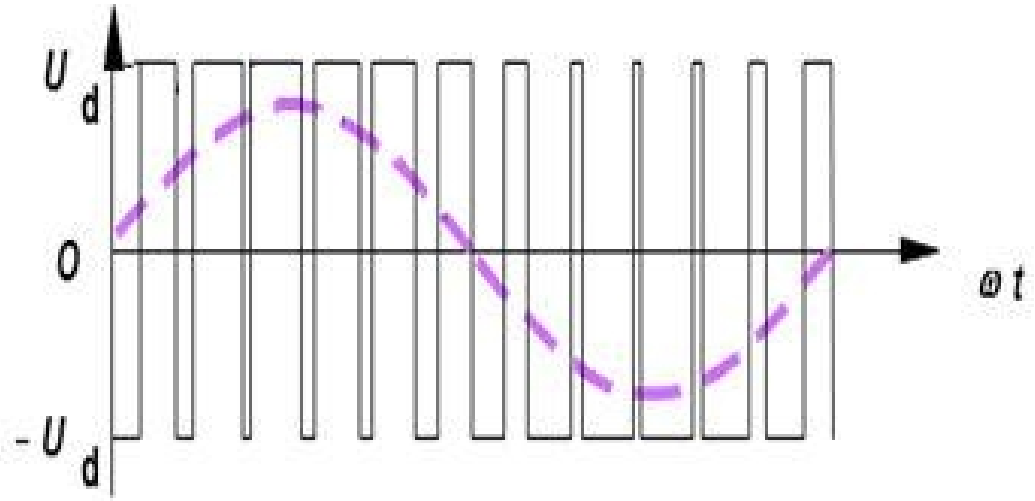
CCR1 输出比较寄存器

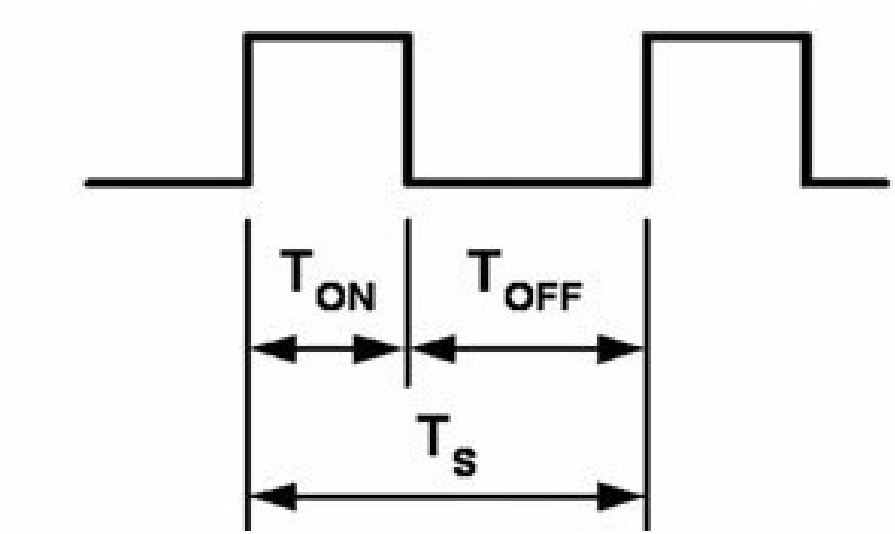
与 CNT 比较，决定高电平长度
例：CCR1 = 250

PWM简介

- PWM（Pulse Width Modulation）脉冲宽度调制
- 在具有惯性的系统中，可以通过对一系列脉冲的宽度进行调制，来等效地获得所需要的模拟参量，常应用于电机控速等领域
- PWM参数：

  频率 = 1 / T_S　　占空比 = T_{ON} / T_S　　分辨率 = 占空比变化步距





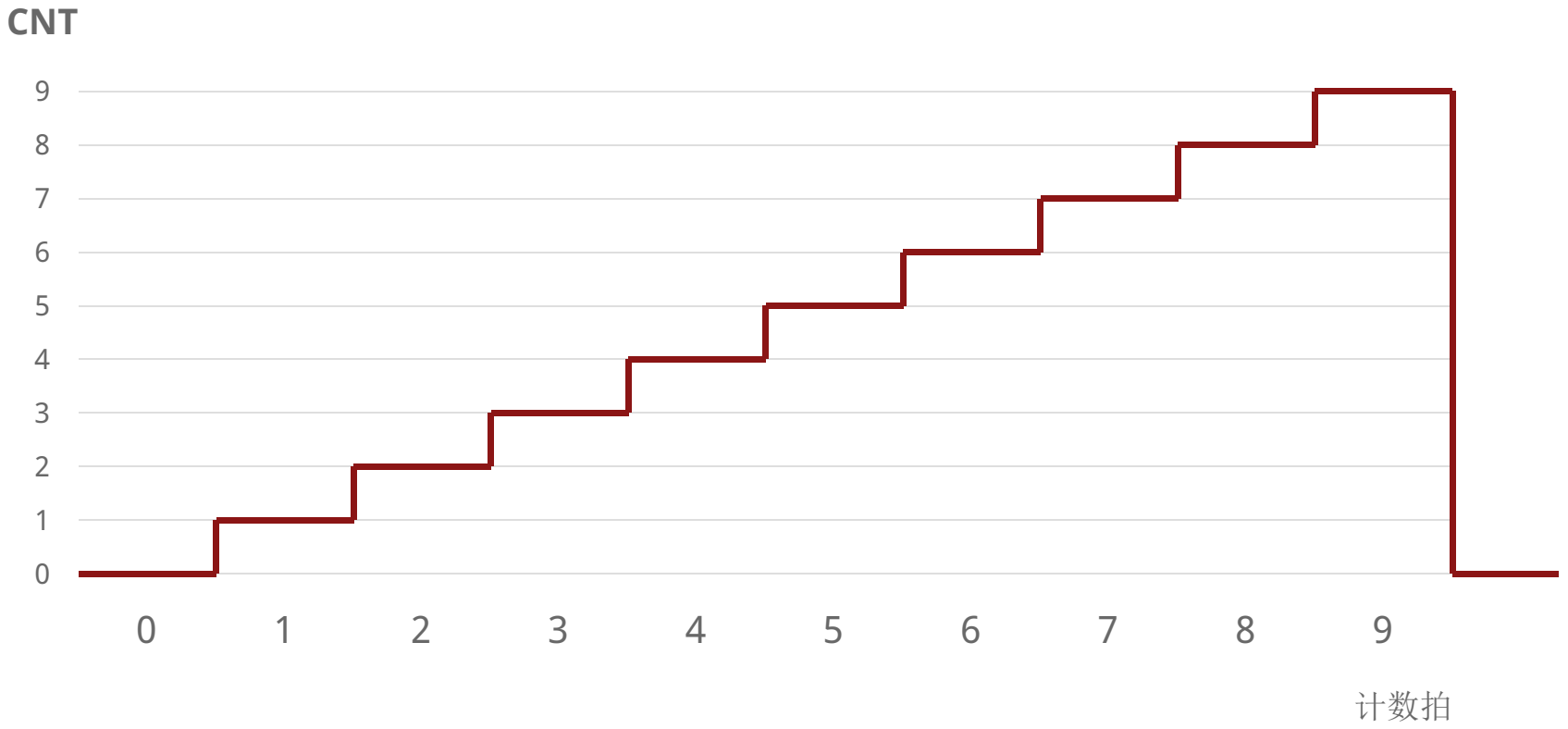
已知：**ARR = 9，向上计数**



问题 1：9 后面的下一拍是什么？

问题 2：一个完整周期是 9 拍，还是 10 拍？

从 0 数到 9，包含的是 10 个状态

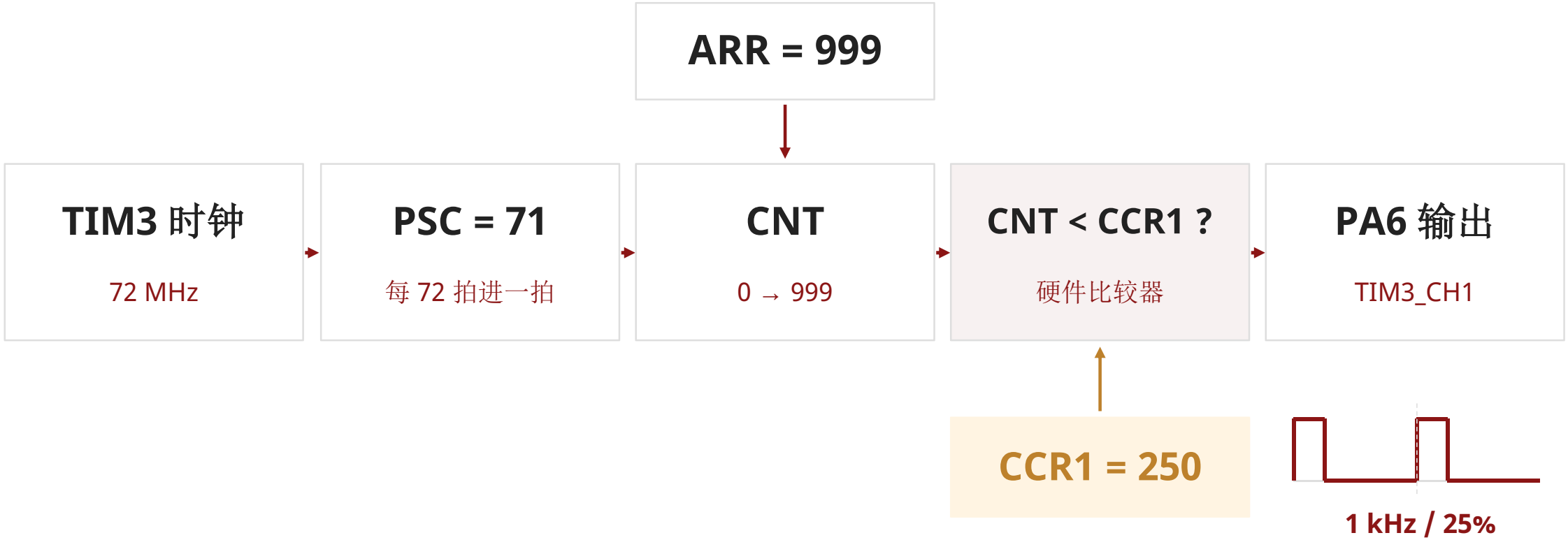


10 拍

不是 9 拍

0 也占一拍

硬件路径

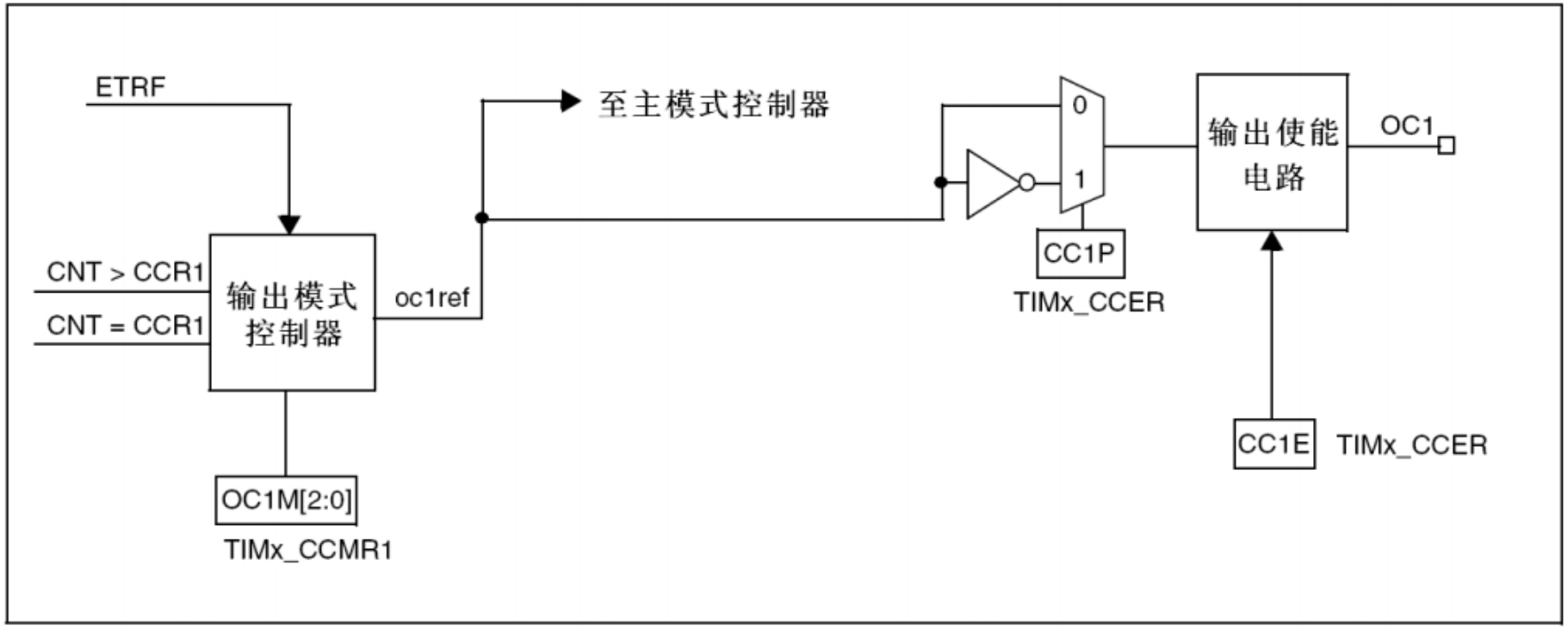

ARR = 999
TIM3 时钟
72 MHz
PSC = 71
每 72 拍进一拍
CNT
0 → 999
CNT < CCR1 ?
硬件比较器
PA6 输出
TIM3_CH1
CCR1 = 250
1 kHz / 25%

输出比较简介

- OC（Output Compare）输出比较
- 输出比较可以通过比较CNT与CCR寄存器值的关系，来对输出电平进行置1、置0或翻转的操作，用于输出一定频率和占空比的PWM波形
- 每个高级定时器和通用定时器都拥有4个输出比较通道
- 高级定时器的前3个通道额外拥有死区生成和互补输出的功能

输出比较通道(通用)

图125　捕获/比较通道的输出部分(通道1)



输出比较模式

模式	描述
冻结	CNT=CCR时，REF保持为原状态
匹配时置有效电平	CNT=CCR时，REF置有效电平
匹配时置无效电平	CNT=CCR时，REF置无效电平
匹配时电平翻转	CNT=CCR时，REF电平翻转
强制为无效电平	CNT与CCR无效，REF强制为无效电平
强制为有效电平	CNT与CCR无效，REF强制为有效电平
PWM模式1	向上计数：CNT<CCR时，REF置有效电平，CNT≥CCR时，REF置无效电平 向下计数：CNT>CCR时，REF置无效电平，CNT≤CCR时，REF置有效电平
PWM模式2	向上计数：CNT<CCR时，REF置无效电平，CNT≥CCR时，REF置有效电平 向下计数：CNT>CCR时，REF置有效电平，CNT≤CCR时，REF置无效电平

比较器的规则：CNT < CCR1 时输出高

ARR = 9，CCR1 = 3

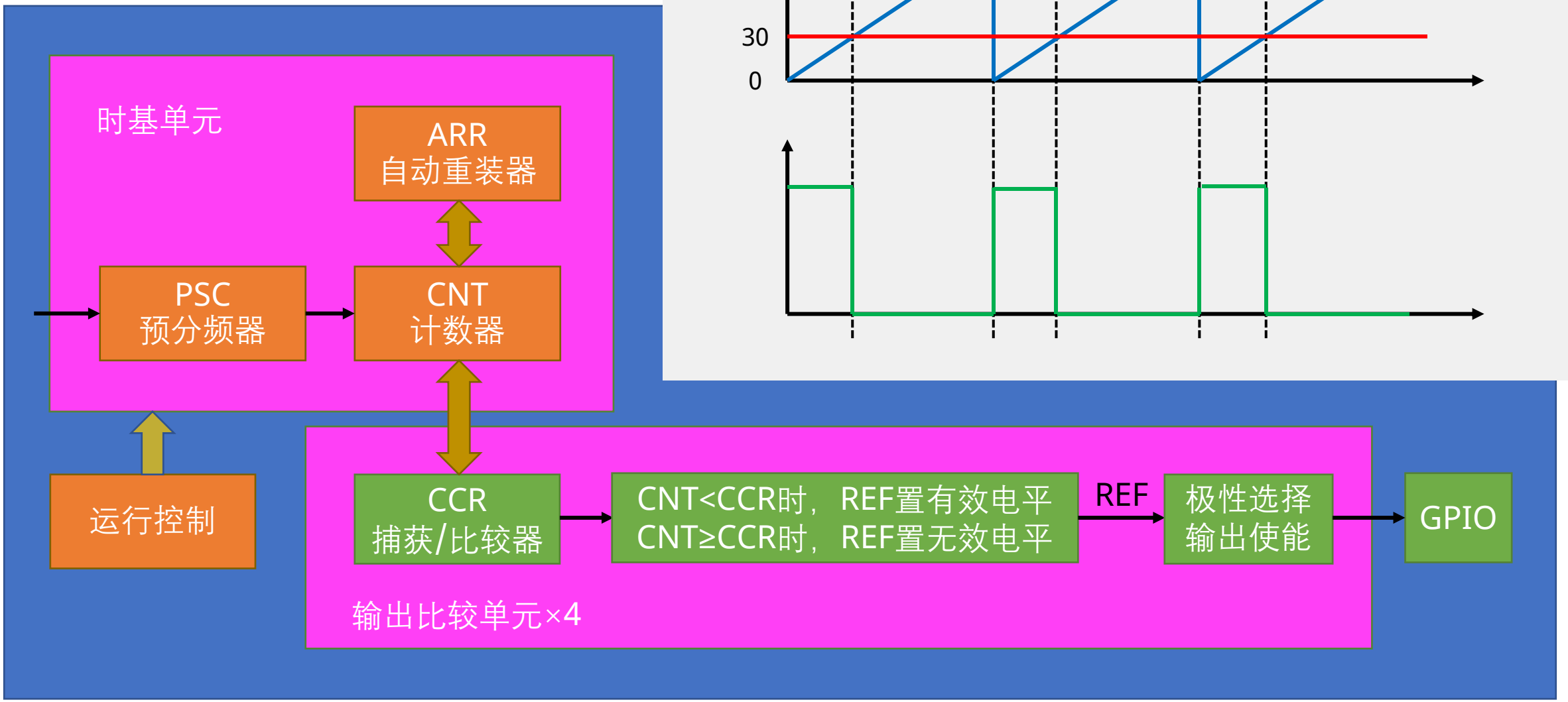
CNT	0	1	2	3	4	5	6	7	8	9
输出	高	高	高	低	低	低	低	低	低	低

高电平占 3 拍

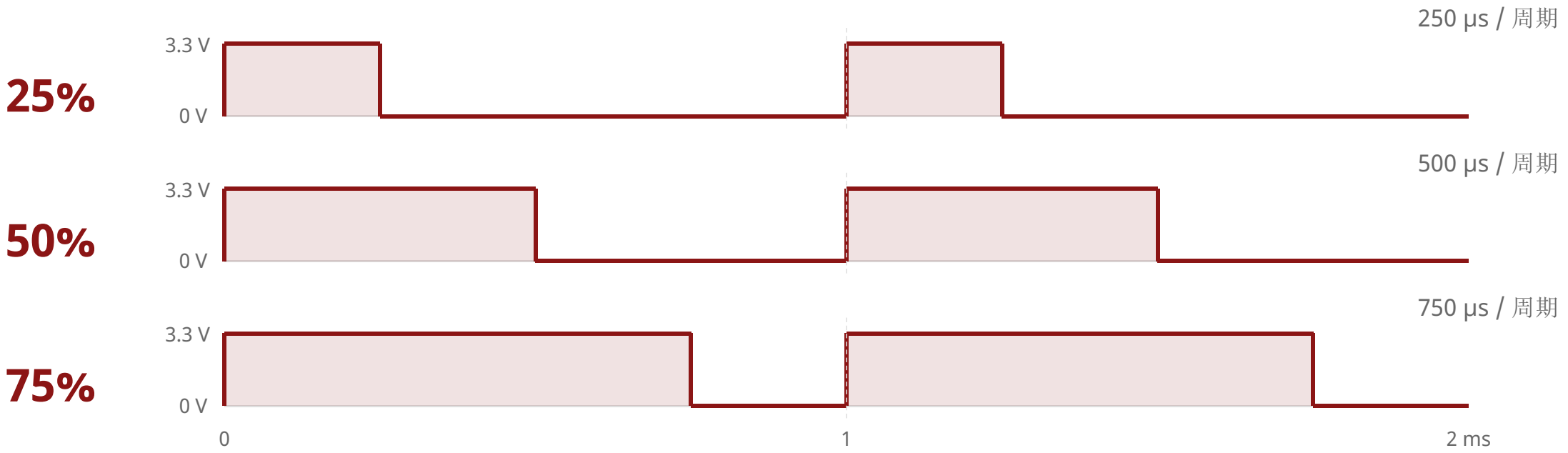
完整周期有 10 拍

D = 3 / 10 = 30%　　比较的是 CNT < CCR1，而不是 CNT ≤ CCR1。

PWM基本结构


99
30
0
时基单元
ARR
自动重装器
PSC
预分频器
CNT
计数器
运行控制
CCR
捕获/比较器
CNT<CCR时，REF置有效电平
CNT≥CCR时，REF置无效电平
REF
极性选择
输出使能
GPIO
输出比较单元×4

PWM 改变的是脉宽，不是高电平电压



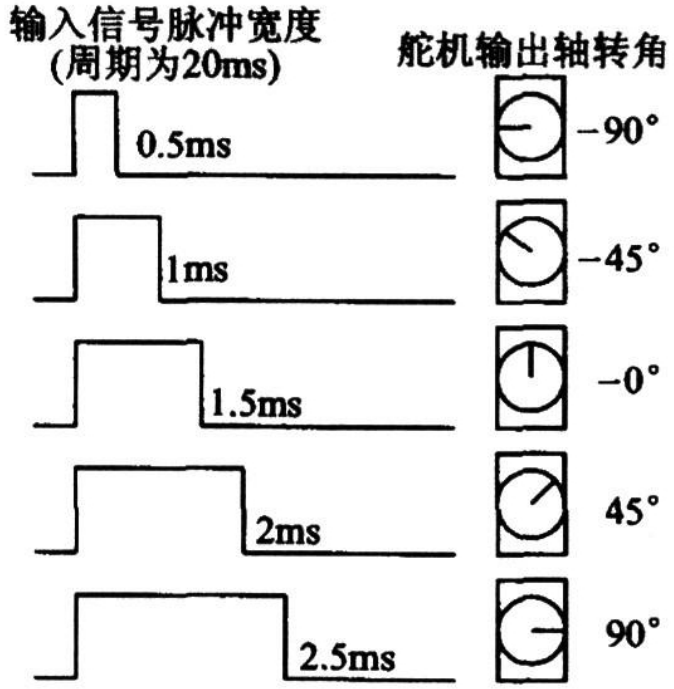
$$D = t_{HIGH} / T = CCR1 / (ARR + 1)$$

舵机简介

- 舵机是一种根据输入PWM信号占空比来控制输出角度的装置
- 输入PWM信号要求：周期为20ms，高电平宽度为0.5ms~2.5ms







频率计算

fCNT = fTIM / (PSC + 1)

fPWM = fTIM / [(PSC + 1) × (ARR + 1)]

1 MHz

72 MHz ÷ (71 + 1)

每一个计数拍 = 1 μs

1 kHz

1 MHz ÷ (999 + 1)

1000 拍 = 1 ms / 周期

自己设计：2 kHz、25% 占空比

01 已知 TIM3CLK = 72 MHz，PSC = 71。

02 目标：fPWM = 2 kHz，占空比 = 25%。

03 每轮拍数 = ____；ARR = ____；CCR1 = ____。

2 kHz 的参数：ARR = 499，CCR1 = 125

01

先求计数频率

72 MHz ÷ 72 = 1 MHz

02

再求一轮拍数

1 MHz ÷ 2 kHz = 500 拍

03

最后填寄存器

ARR = 500 − 1 = 499；CCR1 = 500 × 25% = 125

125 / (499 + 1) = 25%；周期为 0.5 ms。

先预测，再测量：高电平各有多长？

固定：**TIM3CLK = 72 MHz，PSC = 71，ARR = 999**

CCR1 = 250	CCR1 = 500	CCR1 = 750
高电平：____ μs	高电平：____ μs	高电平：____ μs

追加问题：三组波形的周期变了吗？请说出理由。

高电平变长，周期仍是 1000 μs

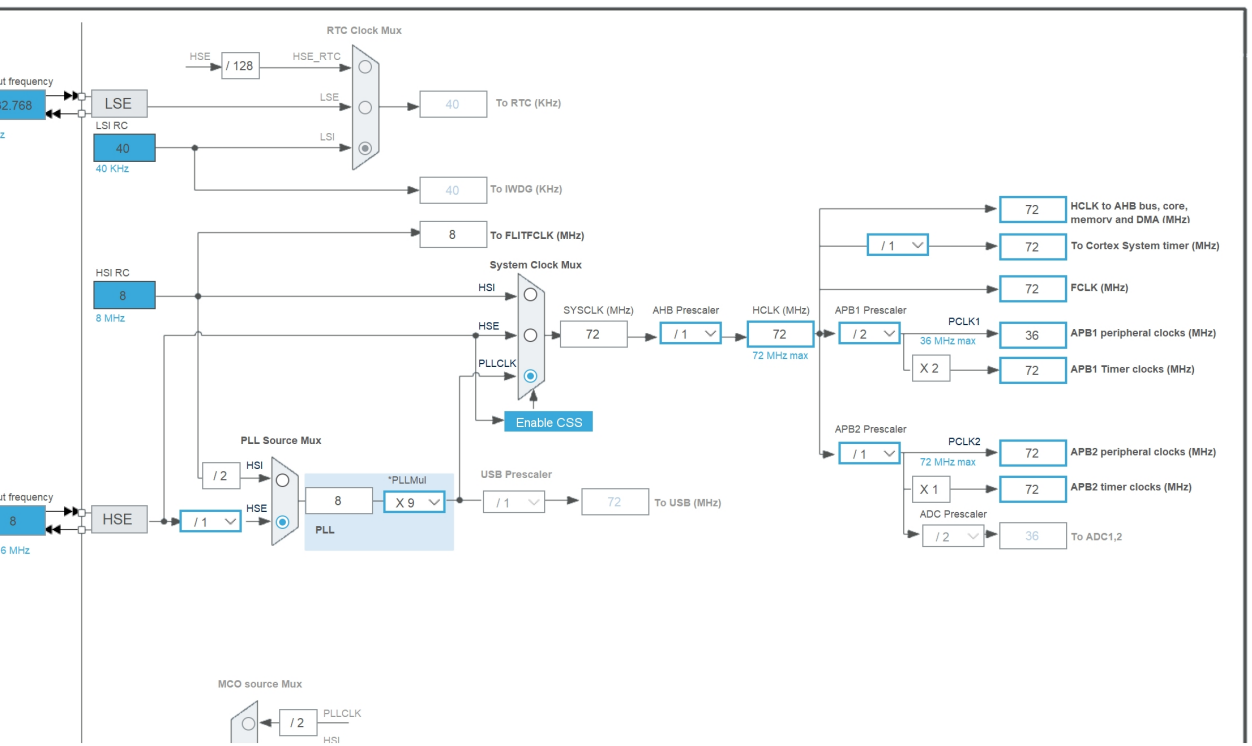
250 μs

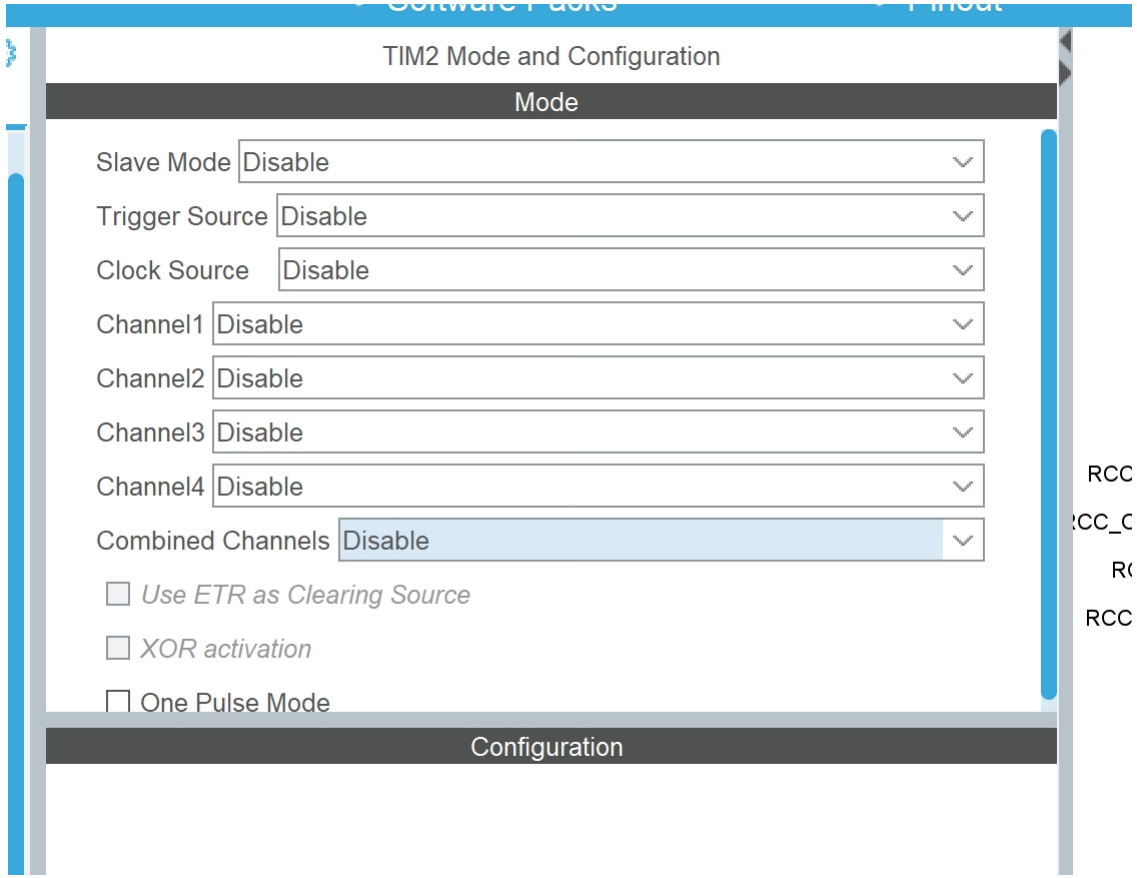
500 μs

750 μs

CCR1	0	100	500	900	1000
占空比	0%	10%	50%	90%	100%
LED 观感	熄灭	较暗	中等	较亮	常亮

CubeMX / CubeIDE：先时钟，再通道





一组参数，让 PA6 准备好输出 1 kHz

参数位置 / 字段	取值	对应含义
Prescaler (PSC)	71	72 分频 → 1 MHz
Counter Mode	Up	CNT 向上计数
Counter Period (ARR)	999	一轮 1000 拍
Pulse	0	初始 CCR1 = 0
OC Polarity	High	高电平有效

生成后再检查：**PA6 的 GPIO 复用推挽配置，以及 TIM3_CH1 的引脚映射。**

代码执行过程

生成初始化

MX_GPIO_Init(); MX_TIM3_Init();

↓

USER CODE BEGIN 2

启动 **PWM：HAL_TIM_PWM_Start(...)**

↓

现有 **while (1) / USER CODE BEGIN 3**

放三档调光代码，或替换为呼吸灯代码

显式启动 PWM

```
1 /* main.c: USER CODE BEGIN 2 */
2 if (HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1) != HAL_OK)
3 {
4     Error_Handler();
5 }
6 /* USER CODE END 2 */
```

启动之后

Pulse = 0
LED 应熄灭
定时器仍在计数

位置：完成初始化之后，进入 **while (1)** 之前。

只改 CCR，观察三档亮度

```
1 __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, 100);
2 HAL_Delay(1000);
3
4 __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, 500);
5 HAL_Delay(1000);
6
7 __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, 900);
8 HAL_Delay(1000);
```

两个边界：为什么 CCR1 = 1000 可以常亮？

CCR1 = 0

CNT < 0 从不成立

0% 高电平
外接 LED 熄灭

CCR1 = 1000

CNT < 1000 始终成立

100% 高电平
外接 LED 常亮

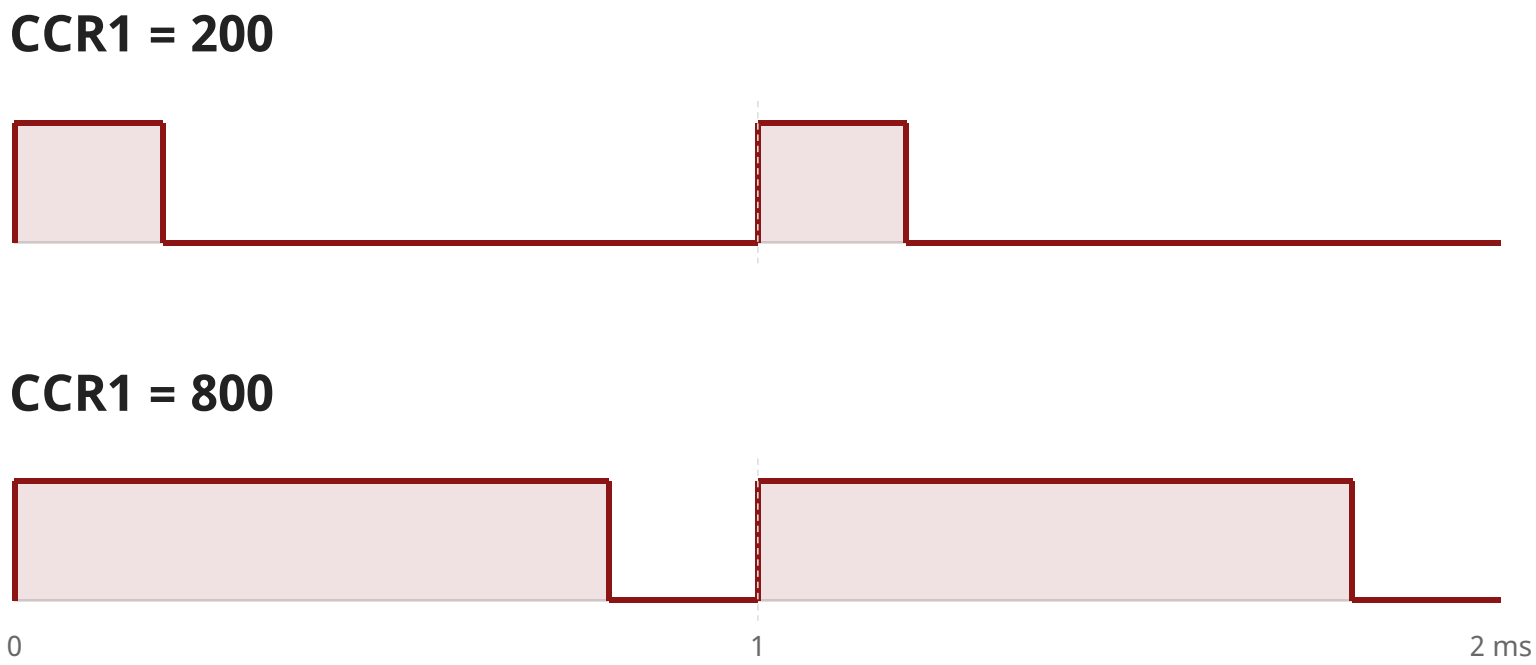
因为 CNT 最大只有 999：100% 对应 ARR + 1，而不是 ARR。

改 CCR：改变脉宽，不改变周期

保持不变

PSC = 71
ARR = 999

fPWM = 1 kHz
T = 1 ms



HAL_Delay(1000) 控制“这一档停多久”，不生成 **PWM** 的每一个脉冲。

呼吸灯代码：先渐亮，再渐暗

```
for (int32_t ccr = 0; ccr <= 1000; ccr += 10)
{
    __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1,
                          (uint32_t)ccr);
    HAL_Delay(8);
}
for (int32_t ccr = 1000; ccr >= 0; ccr -= 10)
{
    __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1,
                          (uint32_t)ccr);
     HAL_Delay(8);
}
```

每次改 10

1 个百分点

每次等 8 ms

控制渐变节奏

两段循环

暗 → 亮 → 暗

位置：原有 while (1) 内；本例 ARR = 999，比较值范围 0～1000。

频率拓展：保持 50%，只改变重复快慢

200 Hz

PSC = 71
ARR = 4999
CCR1 = 2500



T = 5 ms · 高 2.5 ms

1 kHz

PSC = 71
ARR = 999
CCR1 = 500



T = 1 ms · 高 0.5 ms

2 kHz

PSC = 71
ARR = 499
CCR1 = 250



T = 0.5 ms · 高 0.25 ms

随堂测验 1 / 2：参数与波形

01 72 MHz 输入，PSC = 71，CNT 每秒计数多少次？

02 PSC = 71、ARR = 999：PWM 频率是多少？
40% 占空比的 CCR1 应填多少？

03 只把 CCR1 从 200 改成 800，波形什么变、什么不变？

04 ARR = 999 时，为什么 CCR1 = 1000 可以全周期为高？

随堂测验 2 / 2：时钟与控制逻辑

05 PCLK1 = 36 MHz、APB1 分频为 2，
为什么 TIM3CLK 是 72 MHz？

06 PWM 调光能否理解成 GPIO 持续输出 1.65 V？为什么？

07 HAL_Delay(8) 改为 HAL_Delay(20)，
什么变了？PWM 是否降到 50 Hz？

08 LED 改接在 3.3 V 与 PA6 之间、低电平点亮，
亮暗关系可能怎样变化？

可选进阶：用百分比设置亮度

```
1 static void LED_SetDutyPercent(uint8_t percent)
2 {
3     if (percent > 100u) percent = 100u;
4     uint32_t counts = __HAL_TIM_GET_AUTORELOAD(&htim3) + 1u;
5     uint32_t compare = (counts * percent) / 100u;
6     __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, compare);
7 }
8 /* Use: LED_SetDutyPercent(35); */
```

PWM 调光成功后，再进入主从同步

第二部分：时钟源、主模式与从模式

1. 能稳定输出 1 kHz，并观察 10% / 50% / 90% 三档亮度。
2. 呼吸灯能往返渐变；能区分 PWM 周期与亮度调整间隔。
3. 保存成功工程：PWM_LED；另建副本：TIM_SYNC_LAB。

现在的问题变为：谁让 CNT 计数？谁让它复位、暂停或启动？

本专题不再把“调亮度”与“从模式配置”交叉穿插。

从哪里找到定时器的官方结构图？

先找到原图，再用中文标注拆解信号路径

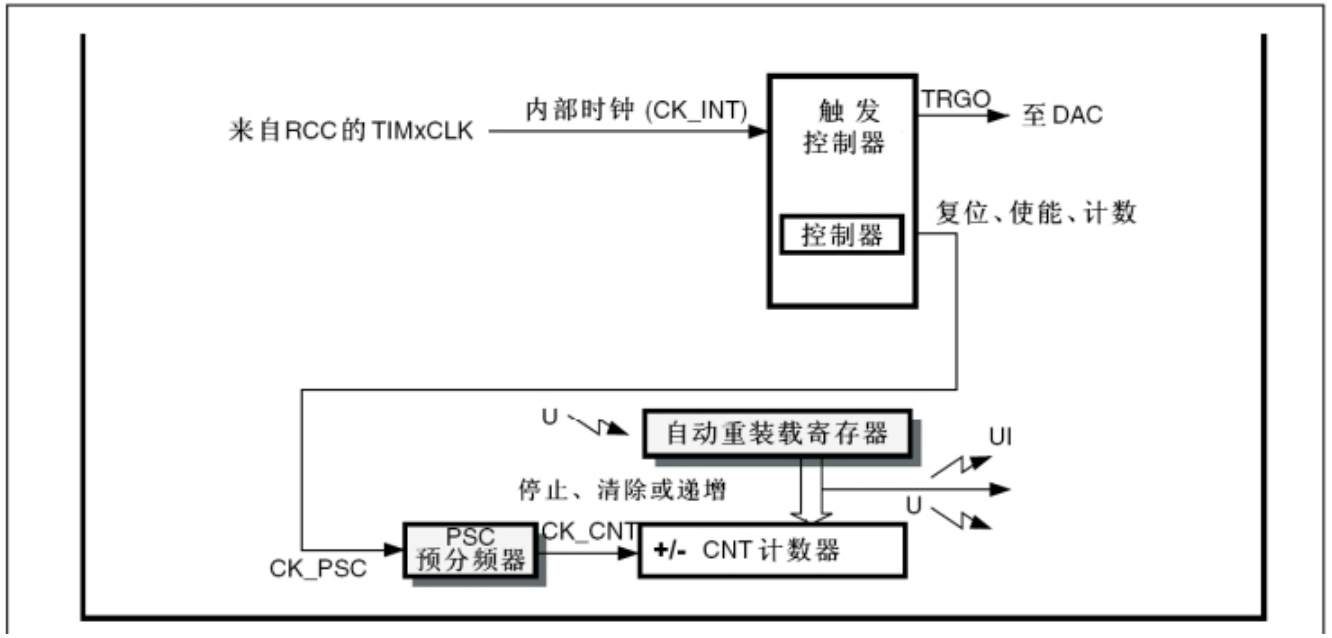
用途	官方文档与定位	课堂读图目标
时钟与从模式入口	AN4776 Rev4 · 图8 · p.19	CK_INT、ETR、TIx → TRGI / PSC
单只 TIM 的 输入与输出	AN4013 Rev14 · 图4 · p.25	TRGI → 控制器 → TRGO
两只 TIM 的级联	RM0008 Rev9 · 图139 · p.350	MMS → TRGO → ITR → TS → SMS

读图顺序：信号来源 → 控制动作 → 计数与输出

后续中文结构图均为依据官方文档的可编辑重绘；文档名与图号可点击打开原图。

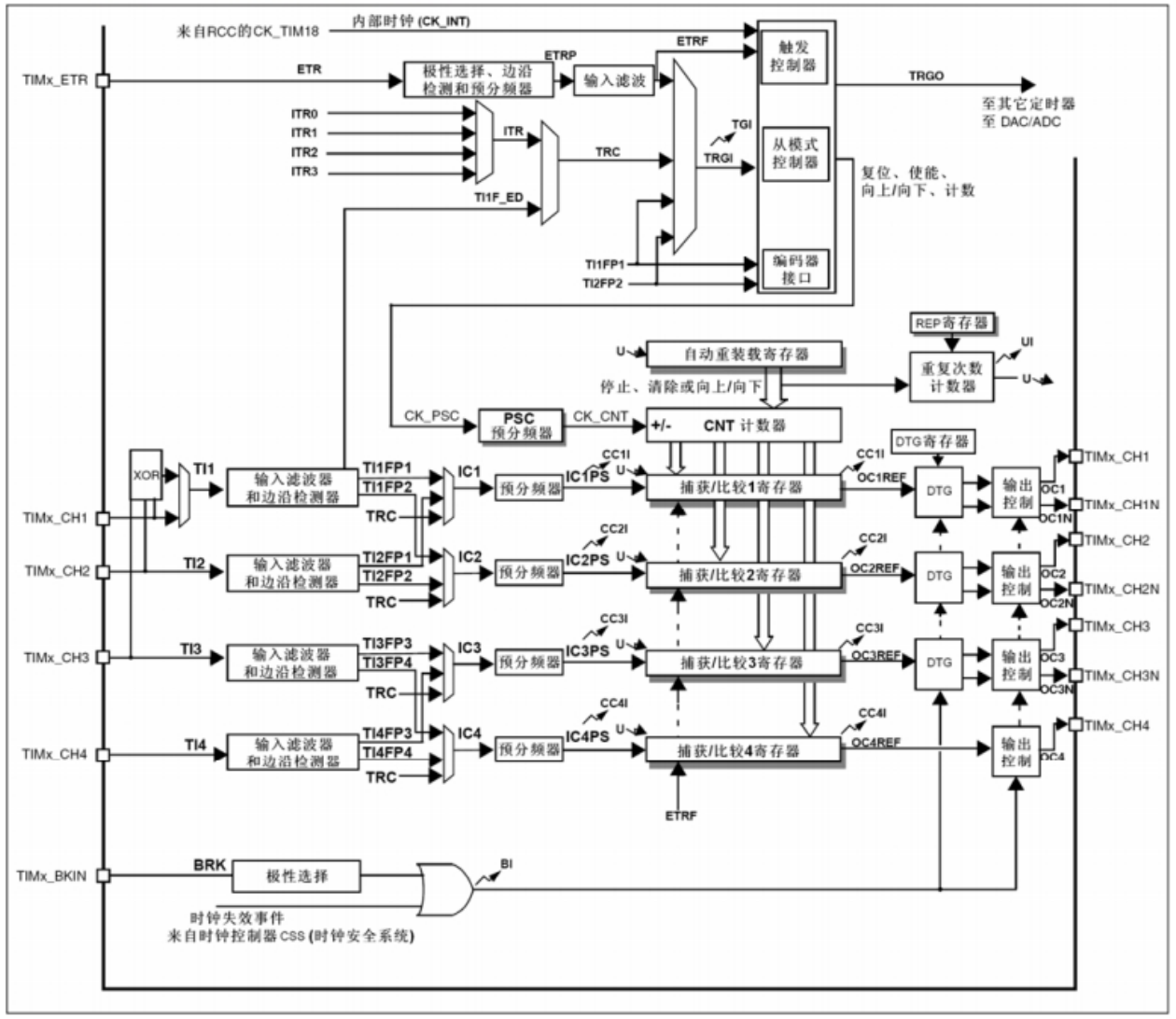
基本定时器

图144　基本定时器框图



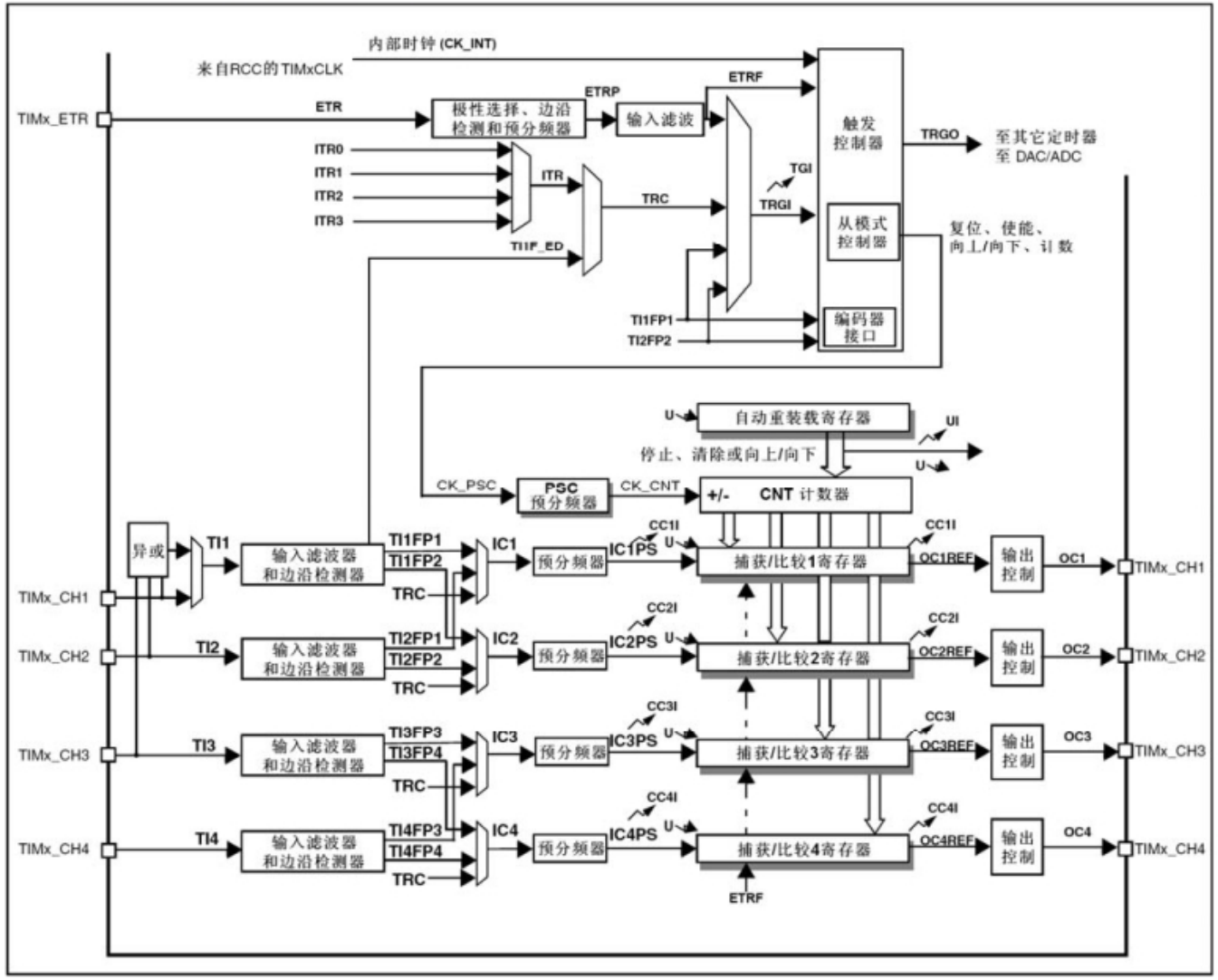
高级定时器

图50　高级控制定时器框图



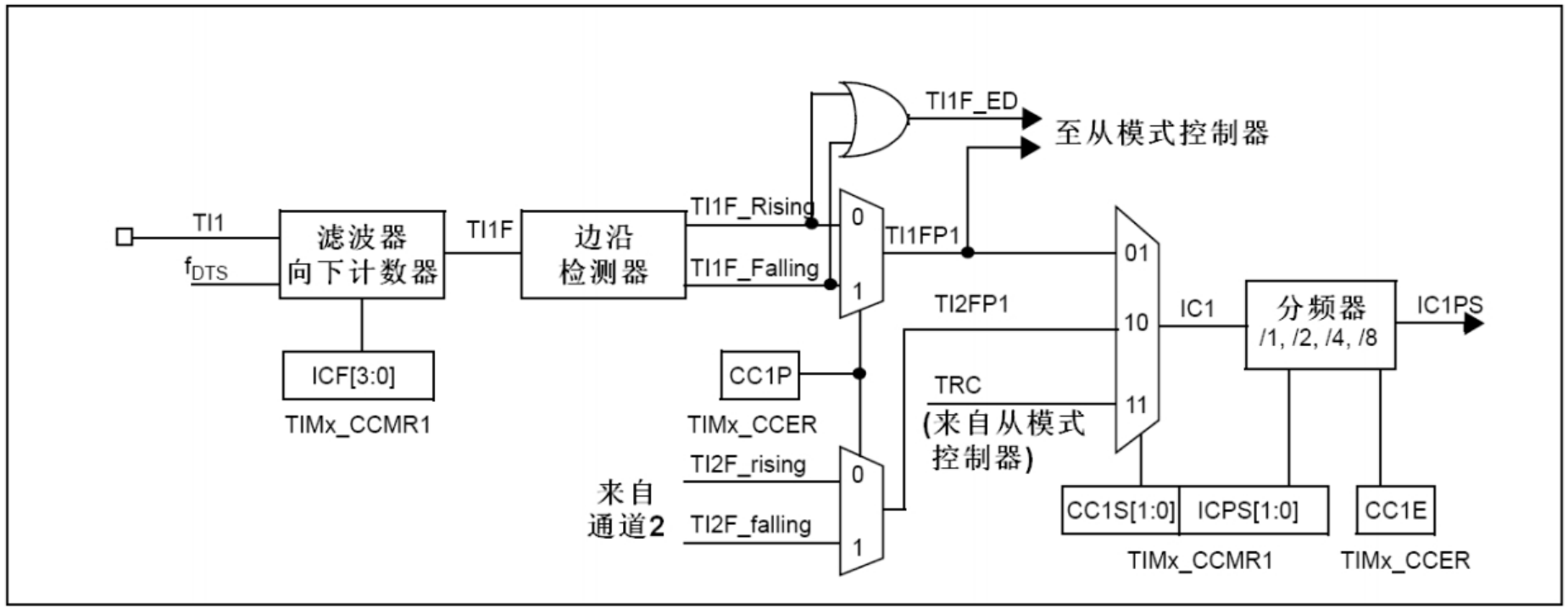
通用定时器

图98　通用定时器框图

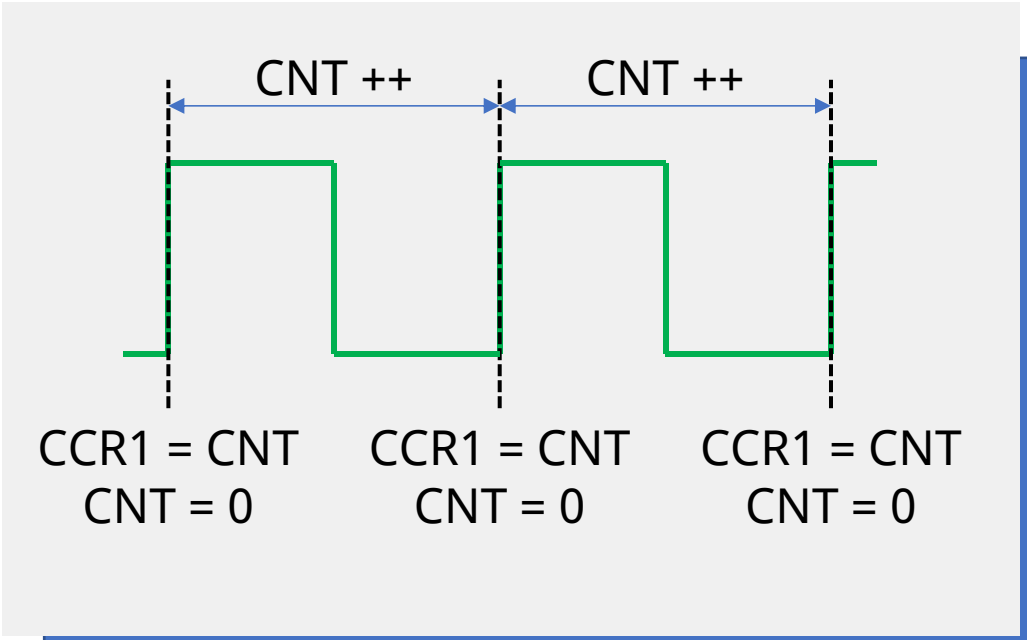


输入捕获通道

图123　捕获/比较通道(如：通道1输入部分)



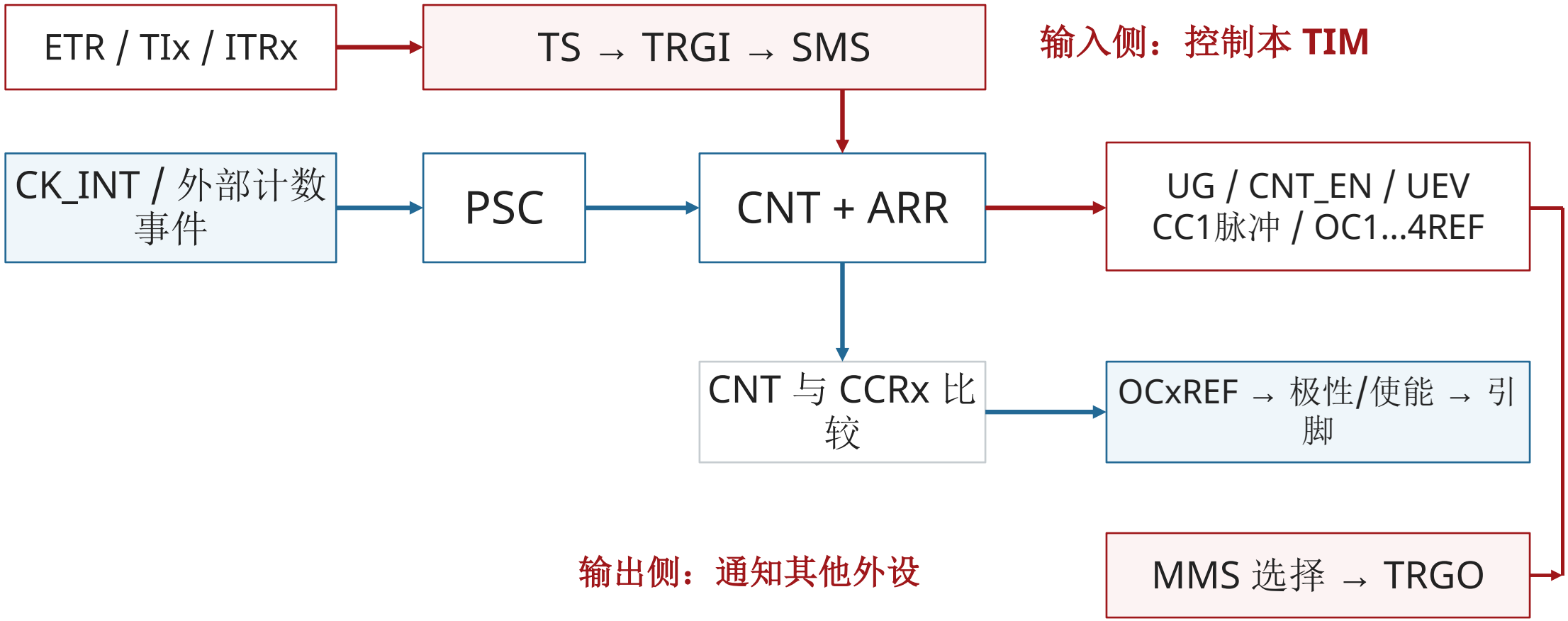
输入捕获基本结构


CNT ++
CNT ++
CCR1 = CNT
CNT = 0
CCR1 = CNT
CNT = 0
CCR1 = CNT
CNT = 0

从模式
Reset
时基单元
ARR
自动重装器
触发源选择
PSC
预分频器
CNT
计数器
GPIO
滤波器
边沿检测
极性选择
TI1FP1
分频器
CCR1
捕获/比较器
输入捕获单元1

一张总图：计数、输入控制、输出同步

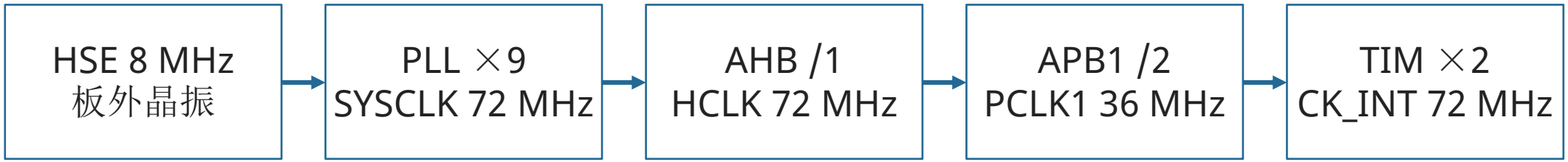
同一个定时器内，这些模块同时存在，并非三选一

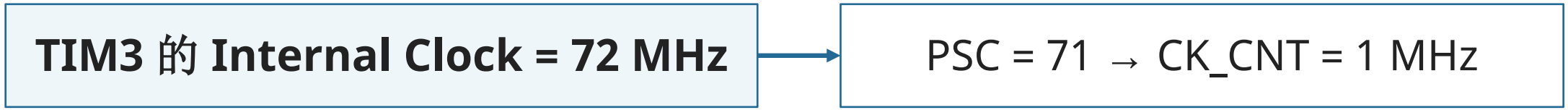


教学重绘 | AN4776 图8；AN4013 图4；STM32F103 RM0008

Internal Clock 不是“必须使用 HSI”

先分清振荡源、系统时钟、定时器内核时钟与计数节拍





HSI / HSE / PLL：决定 RCC 上游来源。
LSI / LSE：本课 F103 的 TIM2 / TIM3 没有直接选择它们的内核时钟入口。

判断频率，看最终 TIMx clock；不要只看 CPU 主频或 PCLK。

依据 STM32F103 时钟树与 RM0008 的 RCC / TIM 章节

Clock Source：谁提供计数事件？

不同入口，最后都要经过 PSC，才推动 CNT

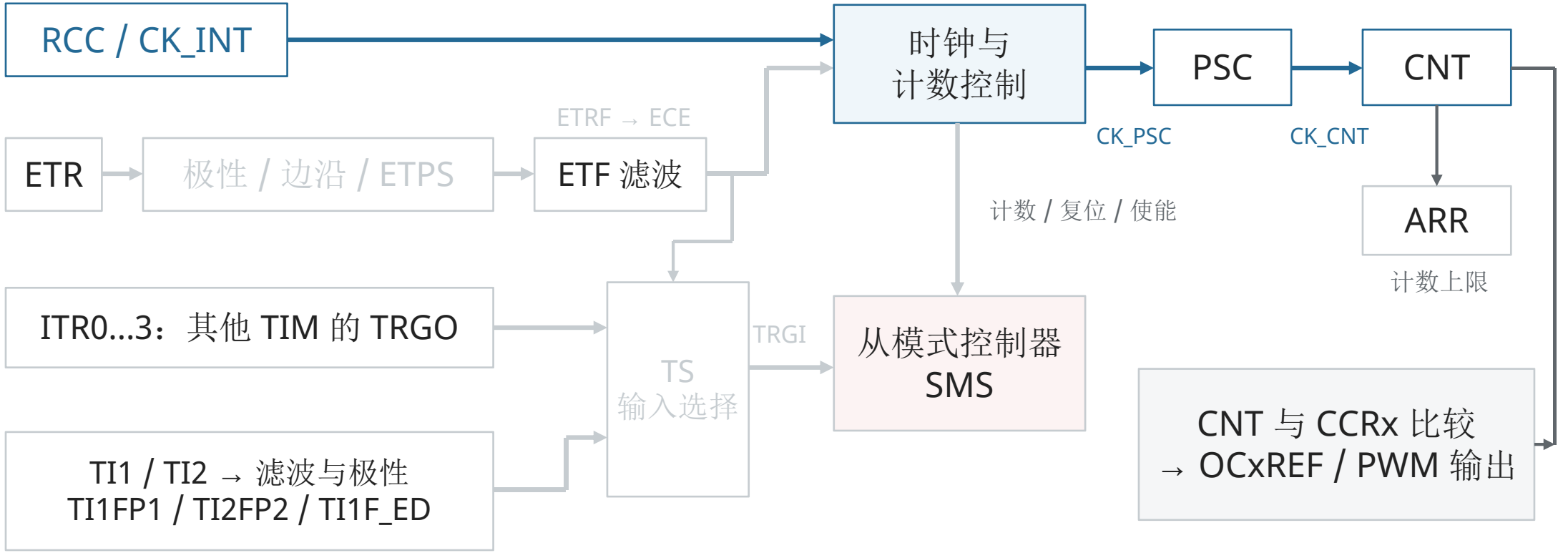
内部时钟	RCC → CK_INT	PSC → CNT
外部时钟模式 1	ITR / TIx / ETRF → TS → TRGI → SMS=111	PSC → CNT
外部时钟模式 2	ETR → 极性 / ETPS / ETF → ETRF → ECE=1	PSC → CNT

“触发信号”只有在 External Clock Mode 1 中，才直接充当计数事件源。

教学重绘 | AN4776 §2.3–2.5，图8

沿官方结构图读：Internal Clock

先回看已经完成的 PWM 路径；其他输入支路暂时不使用

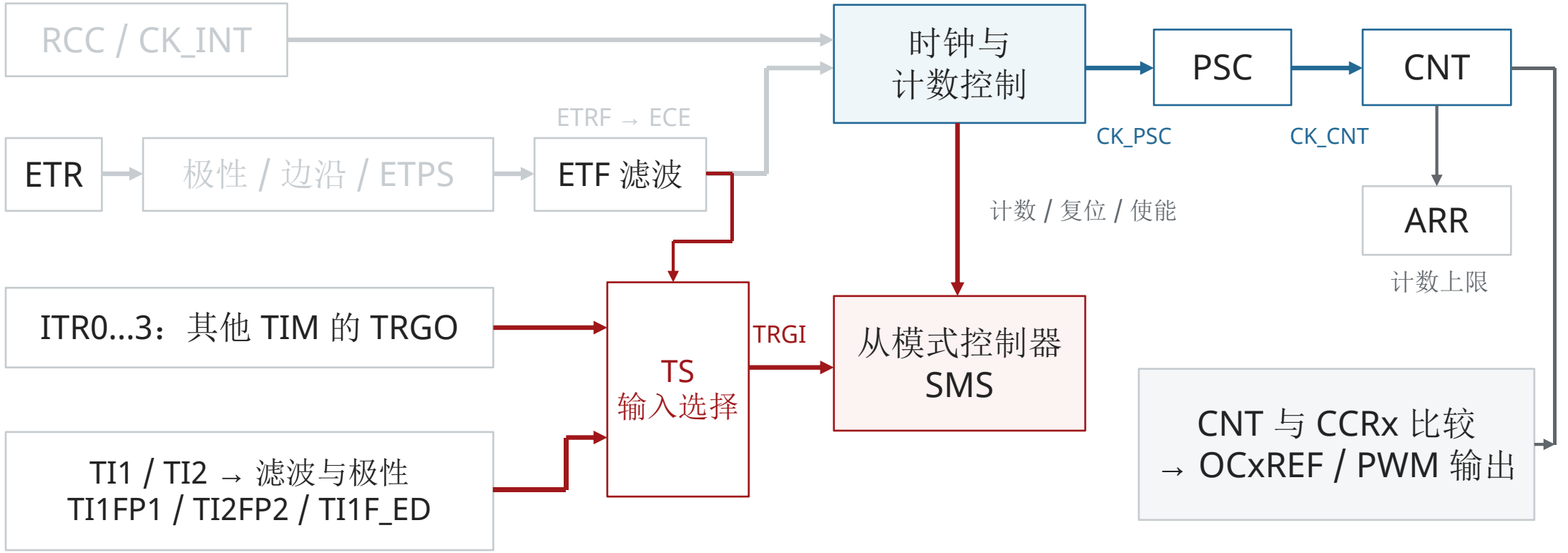


PWM 基线：Internal Clock；Slave Mode = Disable；Trigger Source = Disable。

中文教学重绘（非 ST 原图）｜对应 AN4776 图8 的共同输入/时基部分

沿官方结构图读：外部时钟模式 1

TS 先选 TRGI；SMS=111 再把 TRGI 事件送入计数路径

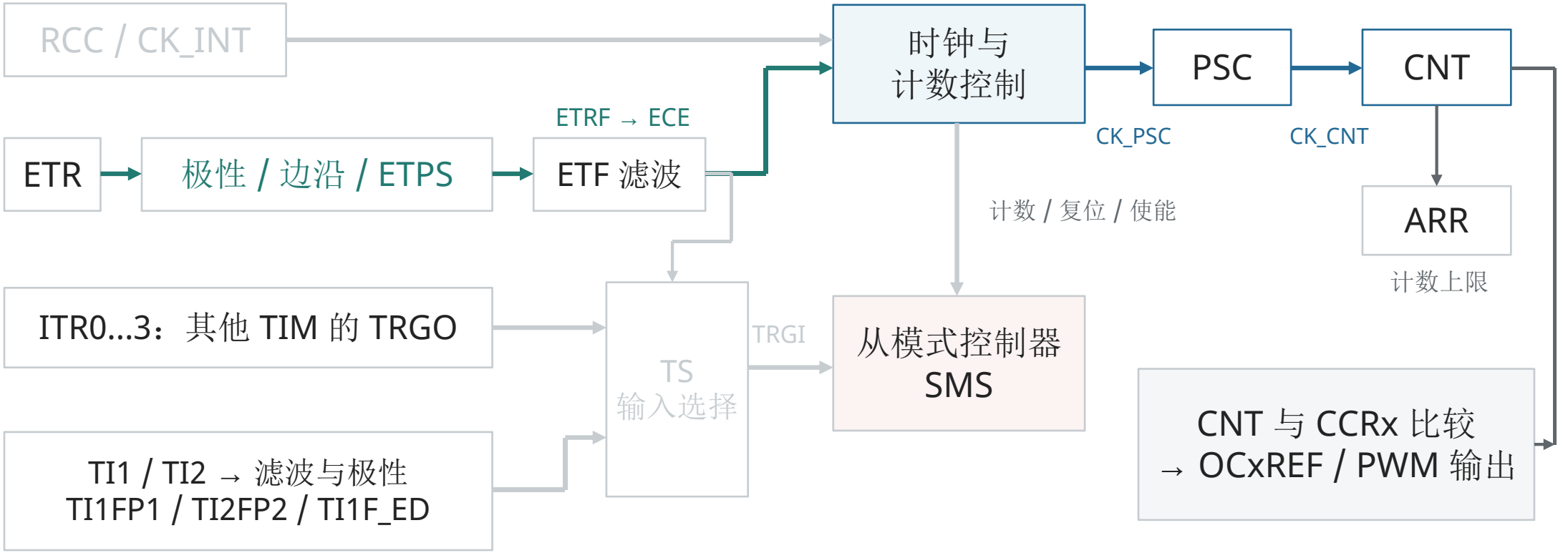


ITRx、TI1FP1、TI2FP2、TI1F_ED 或 ETRF → TRGI → SMS → PSC → CNT。

中文教学重绘（非 ST 原图）｜AN4776 图8、§2.3

沿官方结构图读：外部时钟模式 2

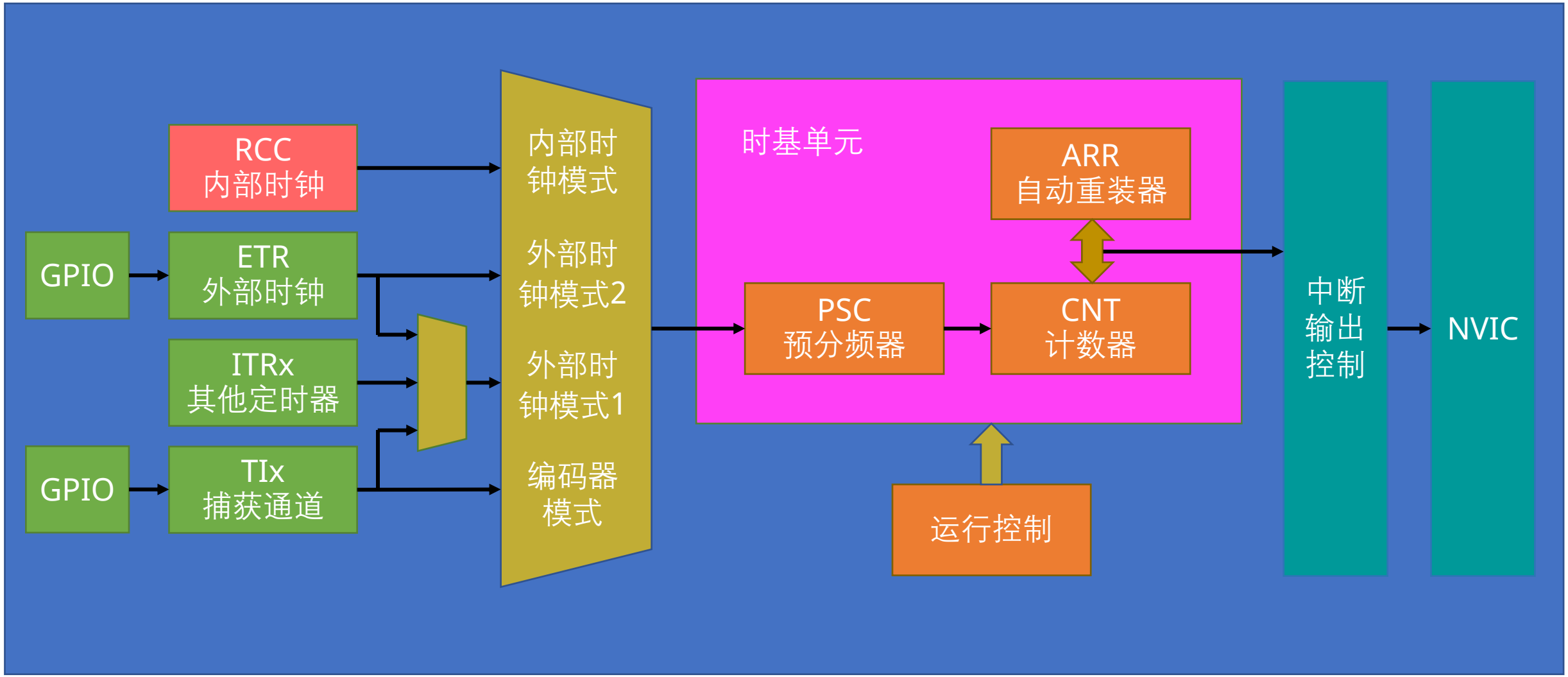
ETRF 走专用计数入口；不必占用 TS 选择的 TRGI 路径



ECE=1 选择 ETRF 作为外部计数来源；TRGI 可以承担另一种控制作用。

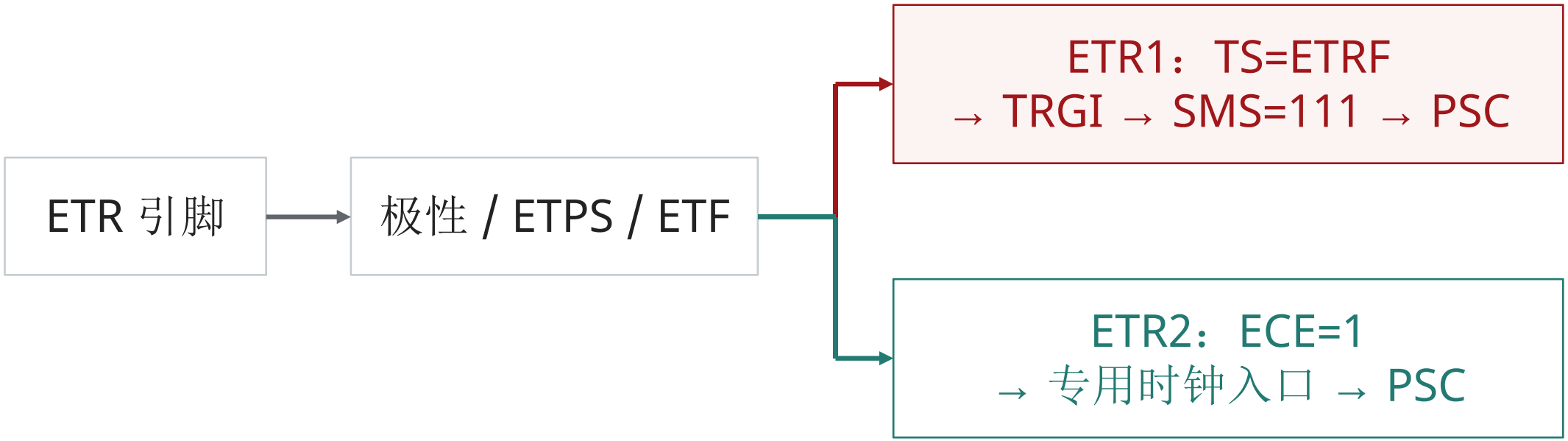
中文教学重绘（非 ST 原图）｜AN4776 图8、§2.4–2.5

定时中断基本结构


RCC
内部时钟
GPIO
ETR
外部时钟
ITRx
其他定时器
GPIO
TIx
捕获通道
内部时钟模式
外部时钟模式2
外部时钟模式1
编码器模式
时基单元
ARR
自动重装器
PSC
预分频器
CNT
计数器
运行控制
中断输出控制
NVIC

ETR1 与 ETR2：同一根 ETR，两种接法

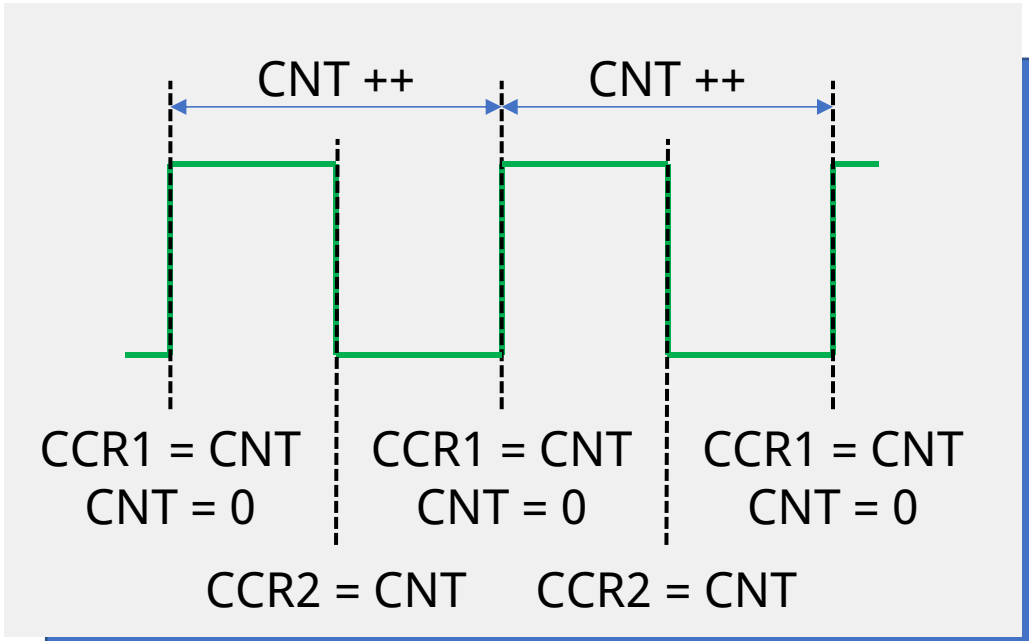
数字 1 / 2 指外部时钟路径，不是两根 ETR 引脚



区别在内部信号怎么走，不在开发板上多了一根引脚。

教学重绘 | AN4776 §2.5；RM0008 的 ECE / TS / SMS 定义

PWMI基本结构

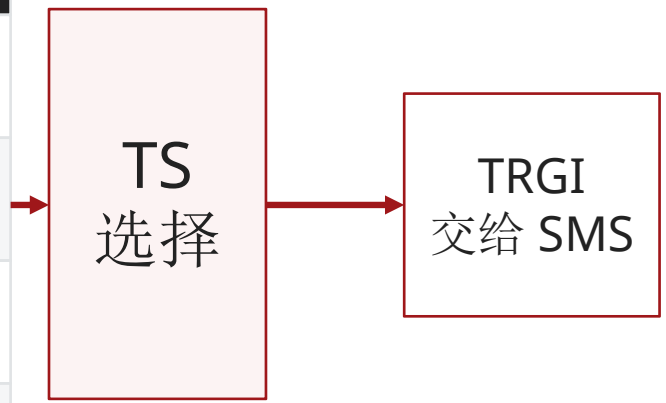

CNT ++
CNT ++
CCR1 = CNT
CNT = 0
CCR1 = CNT
CNT = 0
CCR1 = CNT
CNT = 0
CCR2 = CNT
CCR2 = CNT

从模式
Reset
时基单元
ARR
自动重装器
触发源选择
PSC
预分频器
CNT
计数器
GPIO
滤波器
边沿检测
极性选择
TI1FP1
TI1FP2
分频器
CCR1
捕获/比较器
分频器
CCR2
捕获/比较器

Trigger Source：先选“谁进入 TRGI”

TS 只是输入选择器；它本身不决定计数、复位、门控还是启动

TS 编码	来源	含义
000–011	ITR0…3	其他 TIM 的 TRGO
100	TI1F_ED	TI1 两种边沿产生的脉冲
101	TI1FP1	TI1 滤波 / 极性后的信号
110	TI2FP2	TI2 滤波 / 极性后的信号
111	ETRF	ETR 整形后的信号

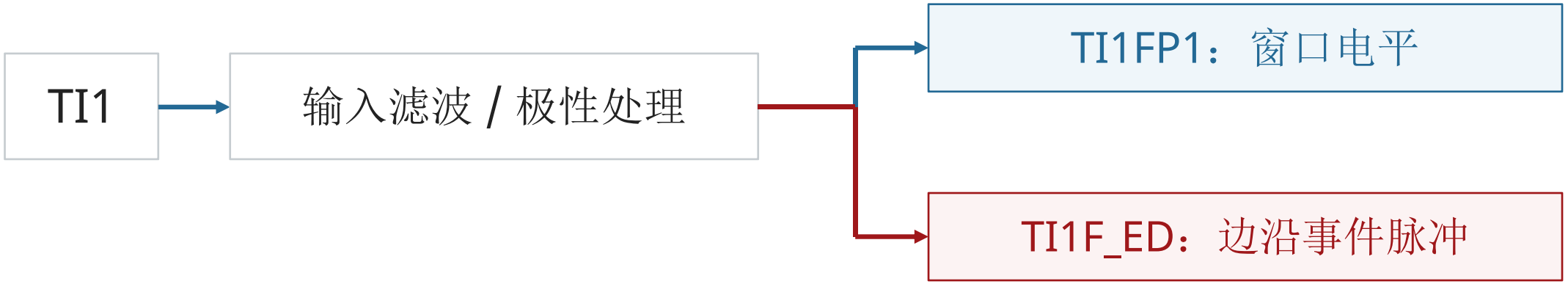


Trigger Source 选输入；Slave Mode 选收到输入后执行的动作。

依据 AN4013 图4 与 RM0008：TIMx_SMCR.TS

TI1FP1 与 TI1F_ED：电平信号和边沿脉冲

Gated 需要一个持续的有效窗口，不是一个很窄的边沿事件





F103 门控模式不要选 TI1F_ED；先说清你要的是“窗口”还是“事件”。

教学语义图 | RM0008：TIMx_SMCR.TS、Gated mode 注意事项

工作模式

表77　计数方向与编码器信号的关系

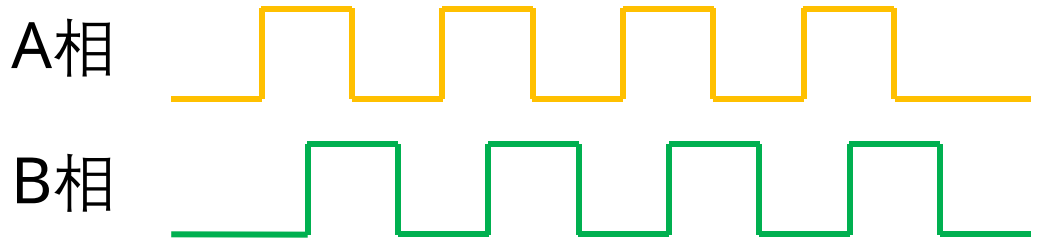
有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

编码器接口简介

- Encoder Interface 编码器接口
- 编码器接口可接收增量（正交）编码器的信号，根据编码器旋转产生的正交信号脉冲，自动控制CNT自增或自减，从而指示编码器的位置、旋转方向和旋转速度
- 每个高级定时器和通用定时器都拥有1个编码器接口
- 两个输入引脚借用了输入捕获的通道1和通道2

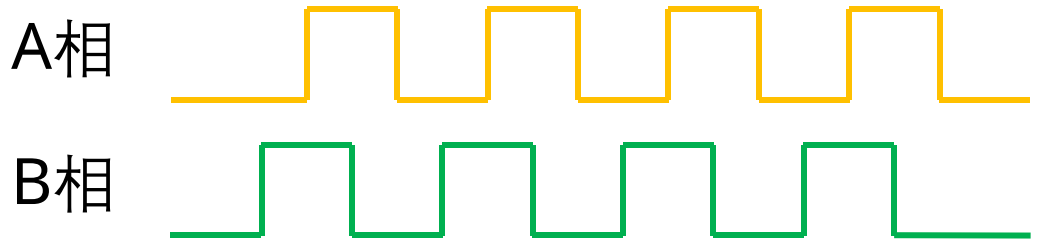
正交编码器

正转



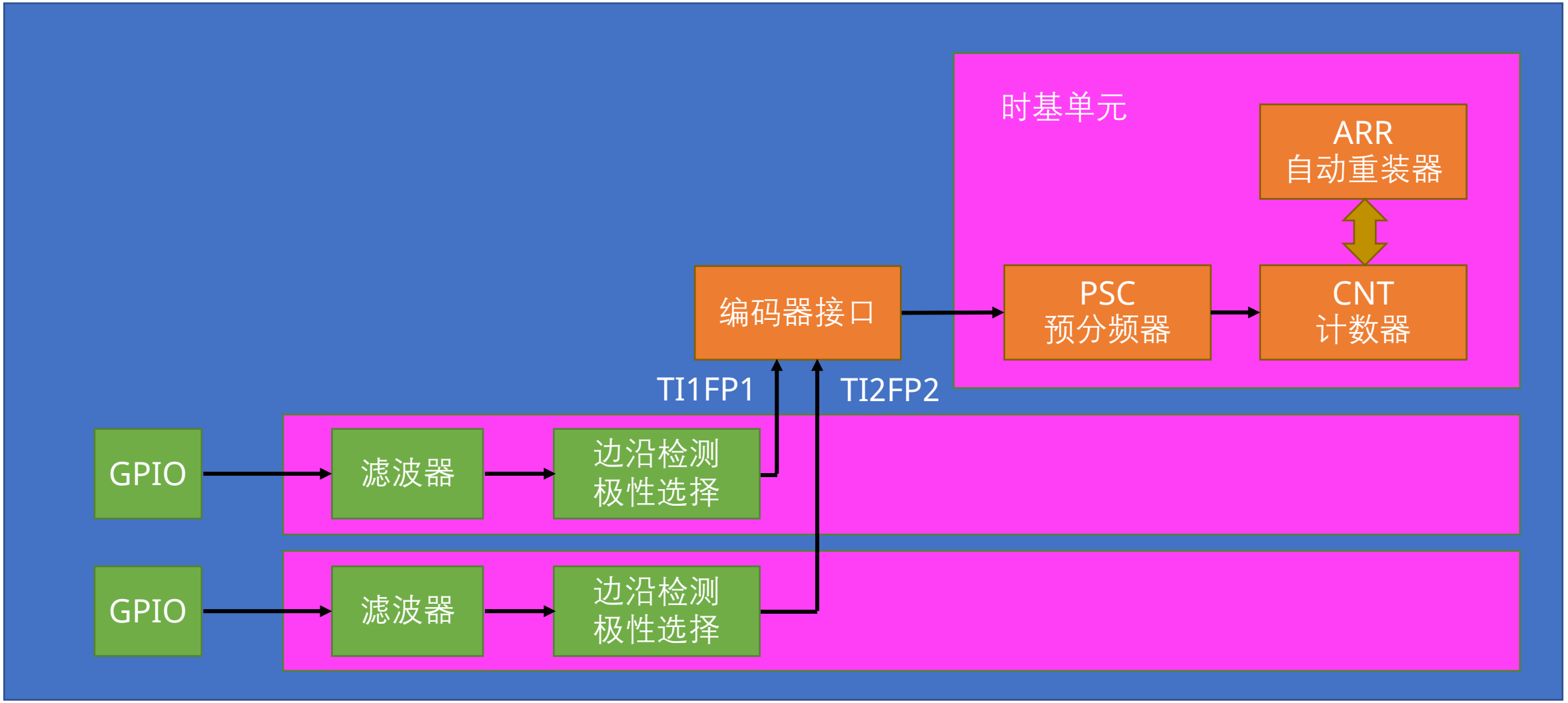
边沿	另一相状态
A相↑	B相低电平
A相↓	B相高电平
B相↑	A相高电平
B相↓	A相低电平

反转



边沿	另一相状态
A相↑	B相高电平
A相↓	B相低电平
B相↑	A相低电平
B相↓	A相高电平

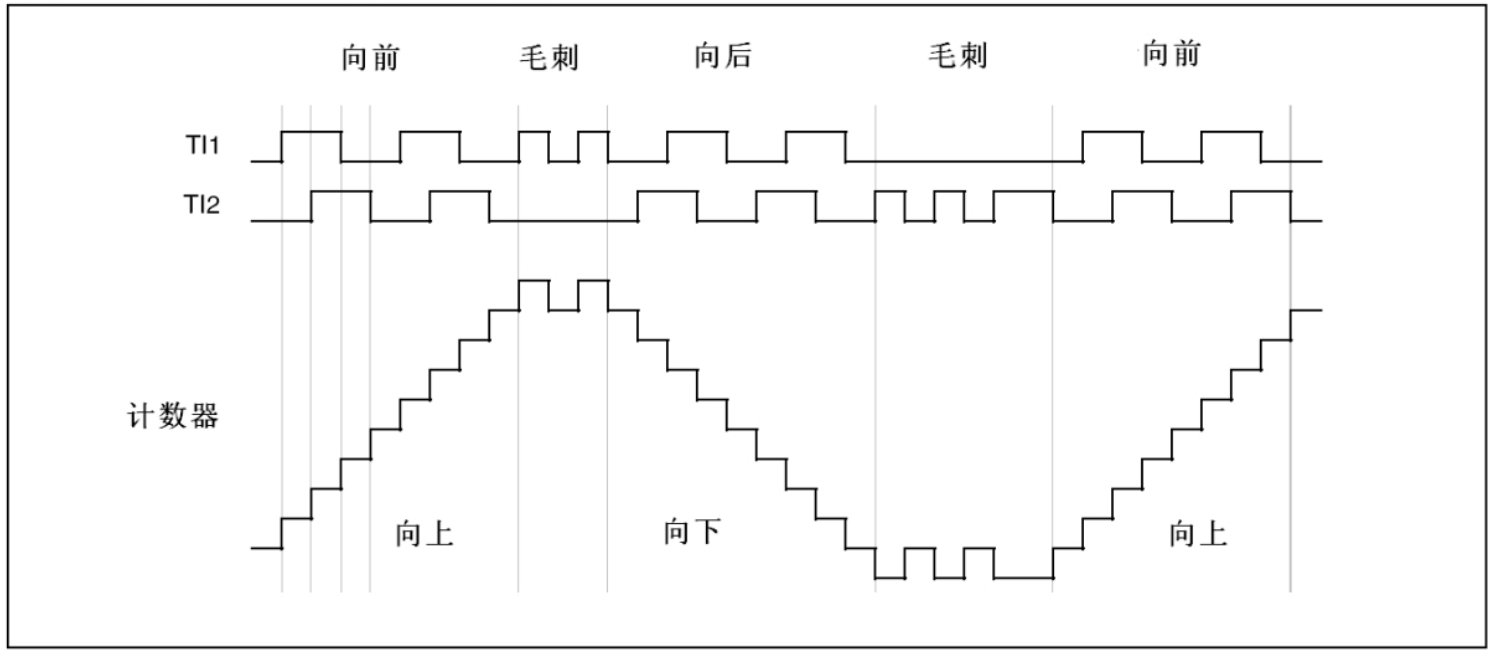
编码器接口基本结构


时基单元
ARR
自动重装器
CNT
计数器
PSC
预分频器
编码器接口
TI1FP1
TI2FP2
GPIO
滤波器
边沿检测
极性选择
GPIO
滤波器
边沿检测
极性选择

实例（均不反相）

有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

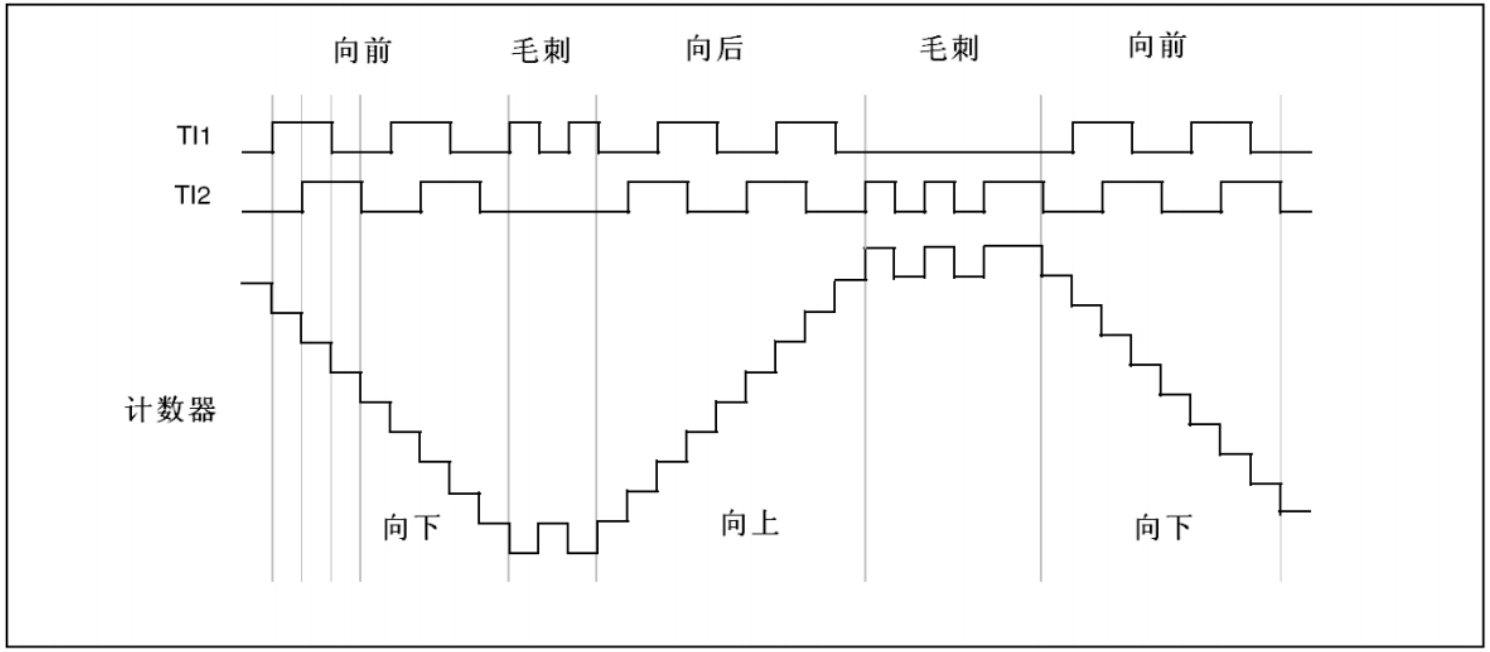
图132　编码器模式下的计数器操作实例



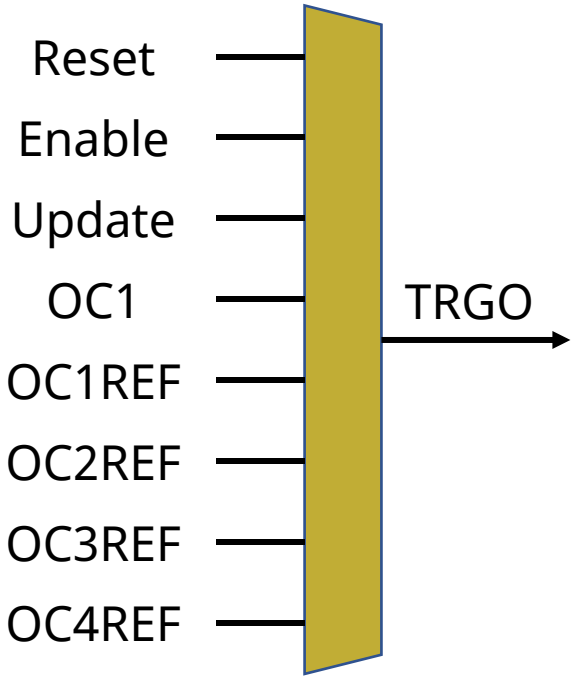
实例（TI1反相）

有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

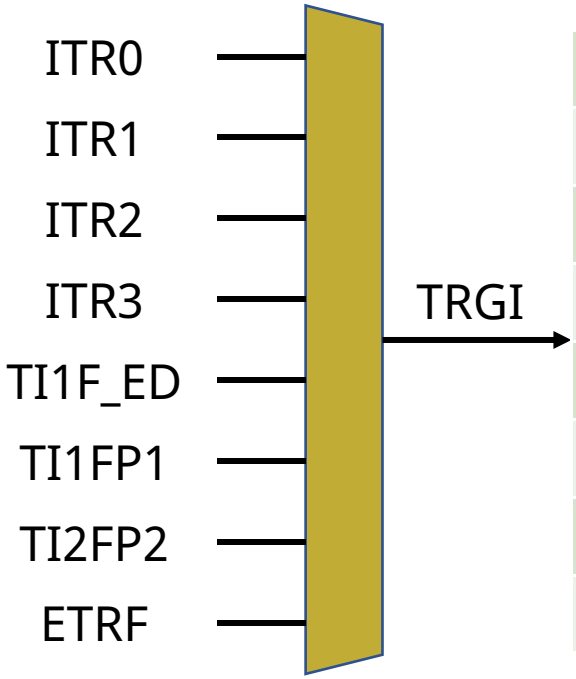
图133　IC1FP1反相的编码器接口模式实例



主从触发模式



主模式

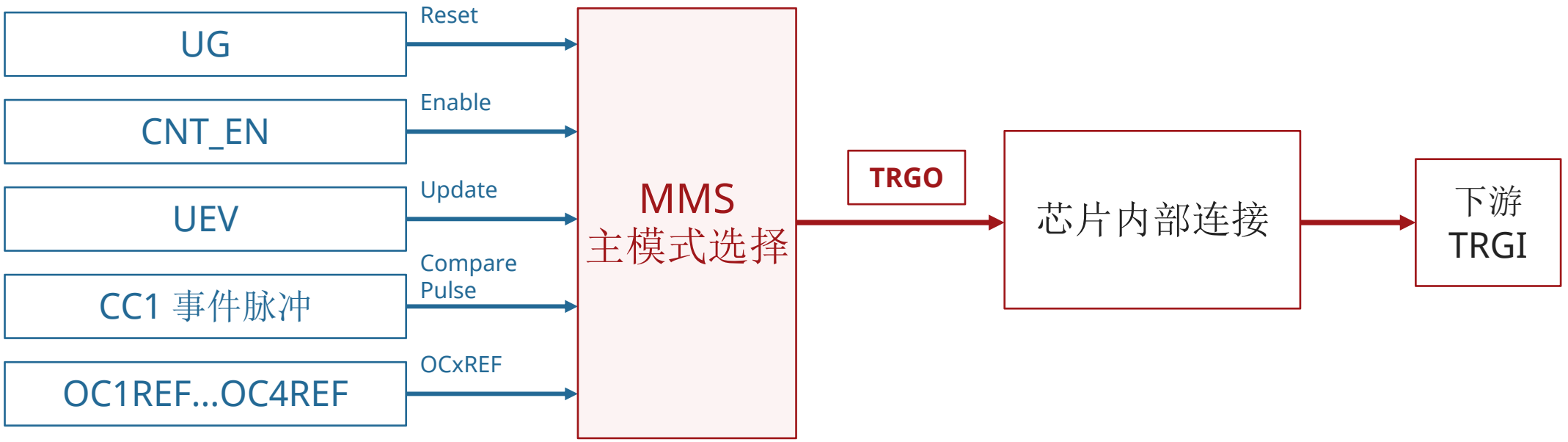


触发源选择

从模式
Closed
Encoder1
Encoder2
Encoder3
Reset
Gated
Trigger
External1

主模式的结构：MMS 选择“发出什么”

MMS → TRGO 是输出侧；它不负责选择本 TIM 的输入时钟

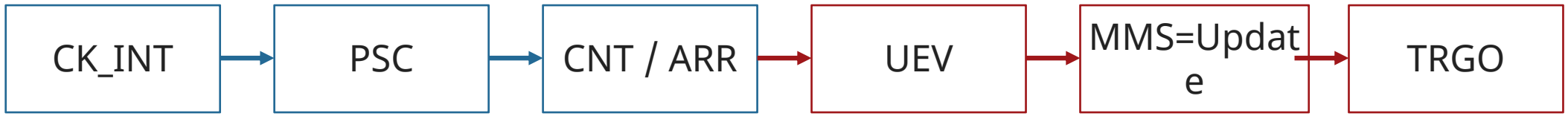


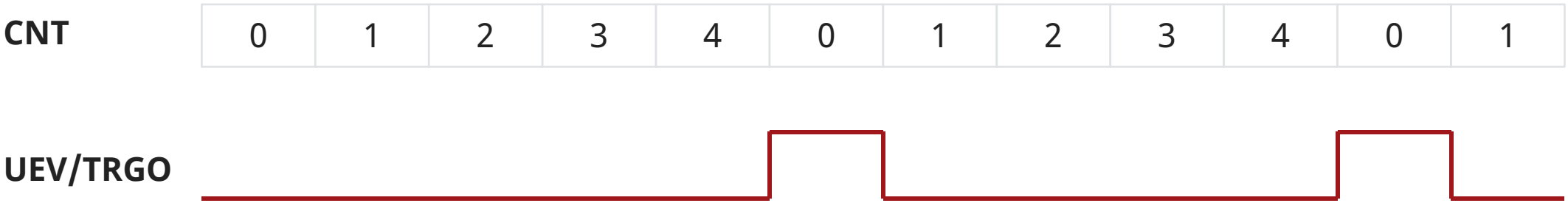
主模式 = 选择 TRGO 的来源；不是“打开 PWM 的总开关”。

教学重绘 | AN4013 §3.2；F103 的实际选项以 RM0008 的 MMS 为准

主模式 Update：把“一个周期”变成一次通知

MMS=010：选择更新事件 UEV 作为 TRGO





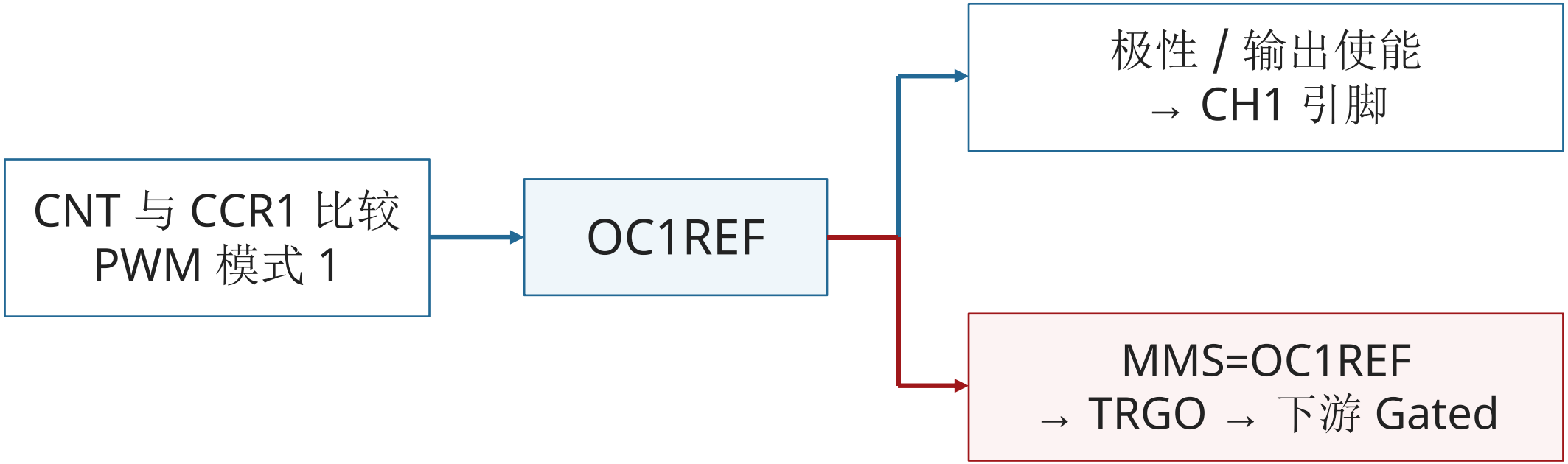
示意：ARR=4；忽略脉冲宽度细节；普通周期运行，不额外写 UG。

主 TIM 每 1 ms 发出一次 UEV，接收方就得到 1 kHz 的触发事件。

依据 AN4013 §3.2、RM0008：MMS=Update

主模式 OCxREF：把 PWM 变成同步窗口

MMS=100…111：选择 OC1REF…OC4REF，而不是读回引脚电平



主 TIM 发出一个高电平窗口；从 TIM 是否在这个窗口内计数，由 SMS 决定。

教学重绘 | AN4013 §3.2；RM0008 输出比较与主模式章节

其他主模式输出：不要只凭菜单名猜

这些选项都在决定 TRGO 输出什么，而不是本 TIM 如何计数

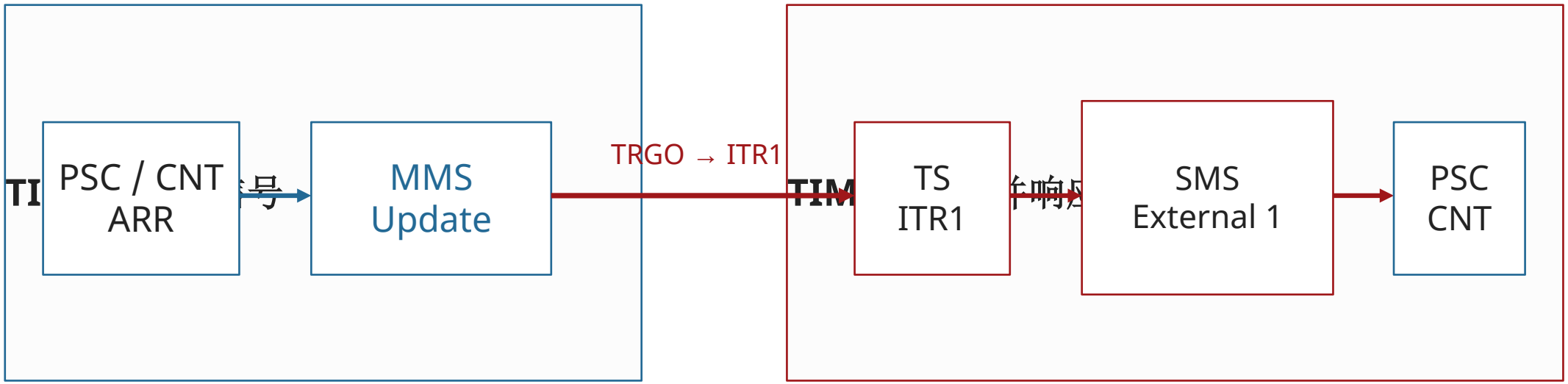
MMS	内部结构路径	输出语义
000 · Reset	TIMx_EGR.UG → MMS → TRGO	软件更新相关的触发信号
001 · Enable	CNT_EN → MMS → TRGO	计数使能状态，可作为窗口
011 · Compare Pulse	CC1 事件 → 脉冲 → MMS → TRGO	每个捕获/比较事件发一个脉冲
100…111 · OCxREF	OC1…4REF → MMS → TRGO	输出比较参考电平

MMS 的 Reset 是“输出源选择”；SMS 的 Reset 才是“收到触发后复位”。

依据 AN4013 §3.2；RM0008：TIMx_CR2.MMS

两只定时器如何连起来？

沿官方 Master/Slave timer example 读：输出 → 连线 → 输入 → 动作



发什么？	从哪接？	接到以后做什么？
TIM2：MMS=Update	TIM3：TS=ITR1	TIM3：SMS=External Clock Mode 1

依据 RM0008 Rev9 图139（p.350）重绘；实例路由按 TIM3 接收表确认

ITR 编号必须站在“接收方”看

F103C8 的常用内部连接：不存在的 TIM 不能拿来接线

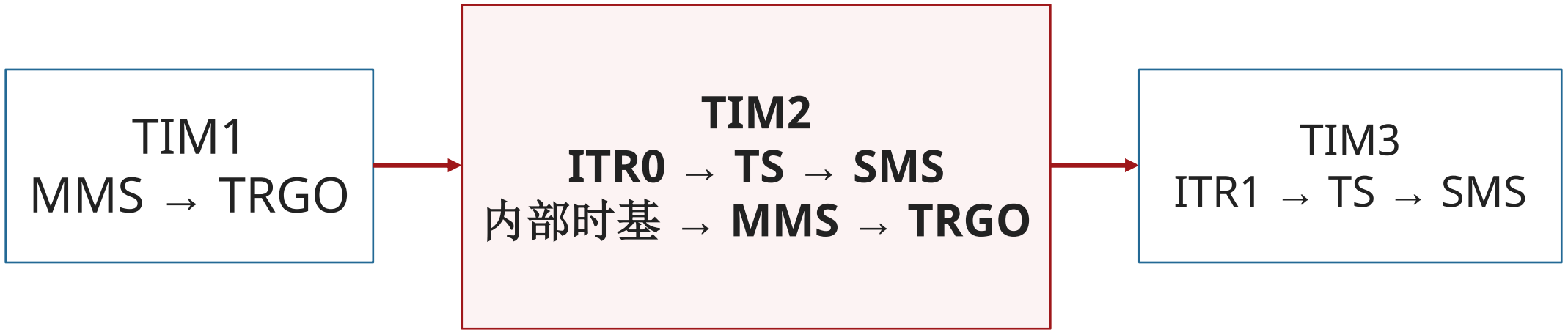
接收方	ITR0	ITR1	ITR2	ITR3
TIM2	TIM1	—（TIM8）	TIM3	TIM4
TIM3	TIM1	TIM2	—（TIM5）	TIM4
TIM4	TIM1	TIM2	TIM3	—（TIM8）



依据 RM0008 Rev9 表76（p.359）；“—”表示 F103C8 没有该源定时器

一只 TIM 可以既是从，又是主

“输入怎么响应”与“输出发什么”是两个独立方向

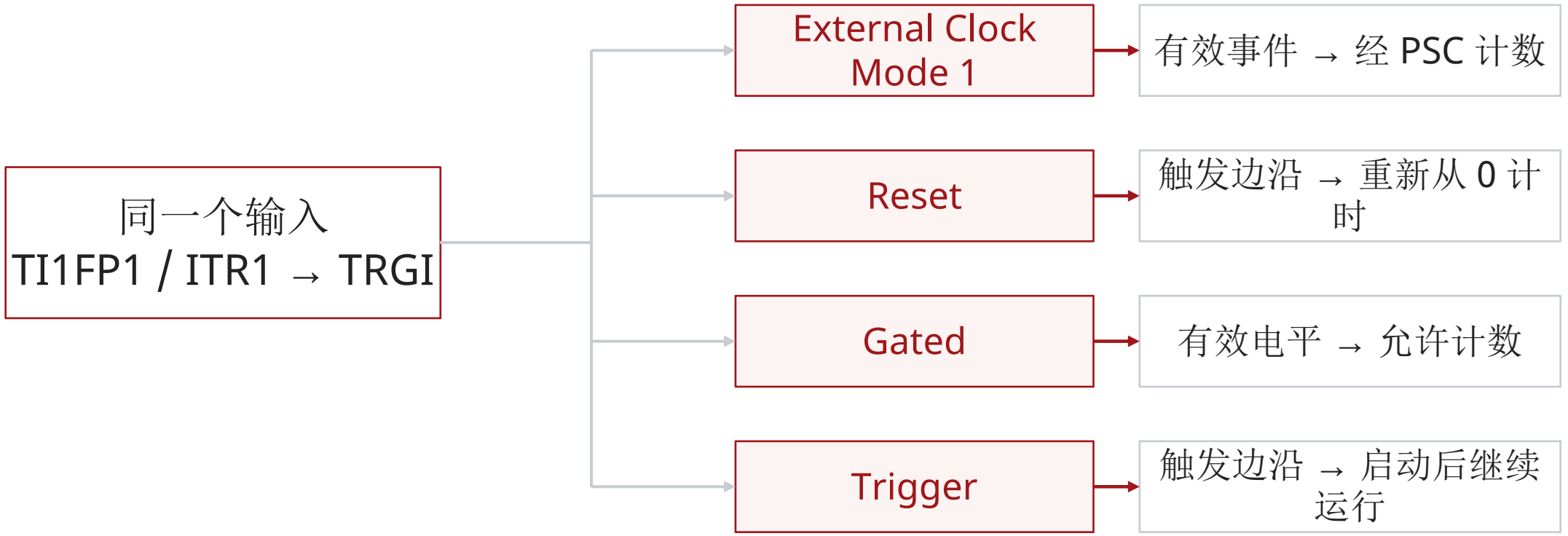


MMS	TS / SMS	MSM
选择输出信号	选择输入 / 输入动作	同步协调，不是“打开主模式”

教学重绘 | AN4013 §3.1–3.3；RM0008：MSM 定义

从模式：同一个 TRGI，可以做四种事

先固定输入来源，再改变 SMS，看计数器的反应

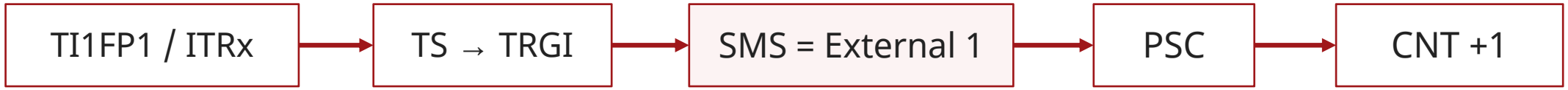


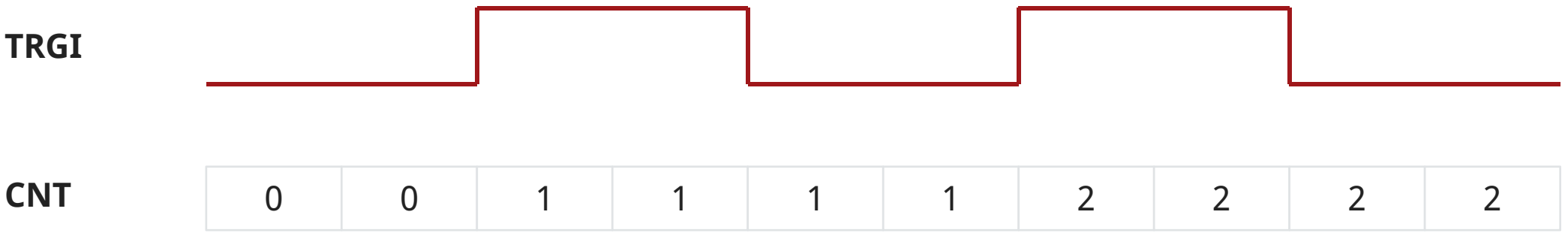
Disable 只是禁用从模式控制；Internal Clock 仍可让定时器正常计数。

依据 AN4013 §3.3；RM0008：TIMx_SMCR.SMS

External Clock Mode 1：输入事件就是计数拍

SMS=111；这里固定 PSC=0，ARR 足够大，不发生溢出





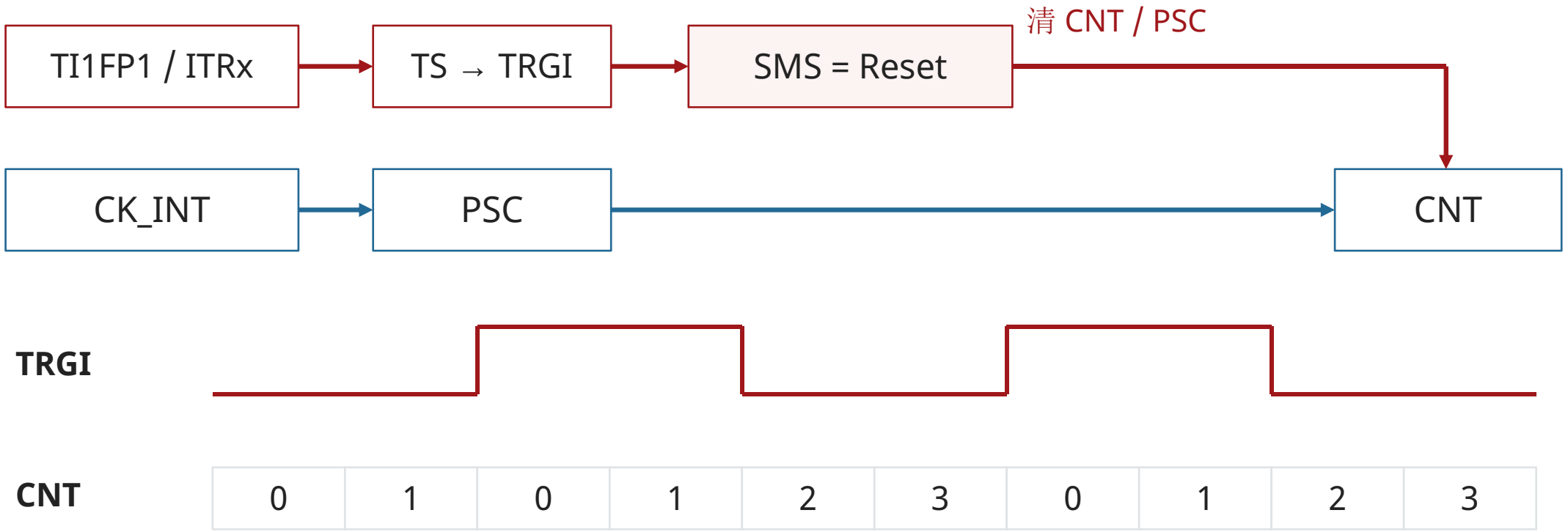
高电平保持多久并不重要；只统计配置所选的有效事件。

“输入高时不停计数”是门控；“每个有效事件计一次”才是外部时钟模式 1。

教学结构/时序示意｜AN4776 §2.3；未表示输入同步延时

Reset Mode：一来触发，就重新计时

SMS=100；本例的计数时钟仍为 Internal Clock

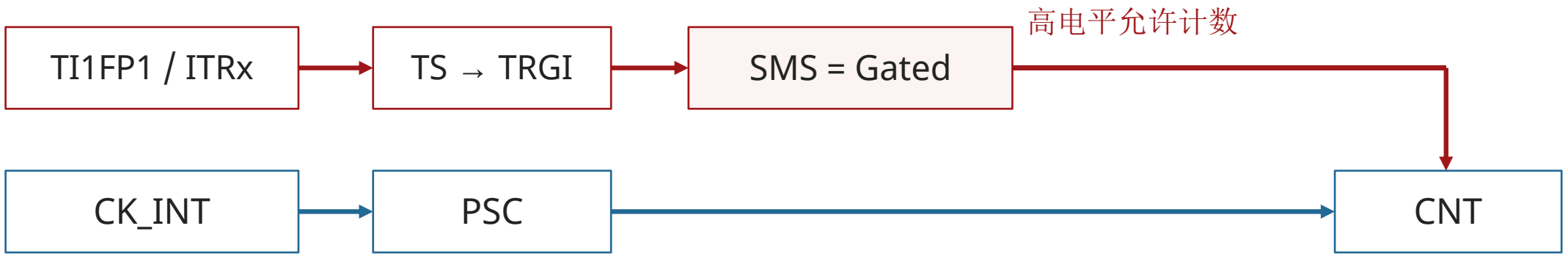


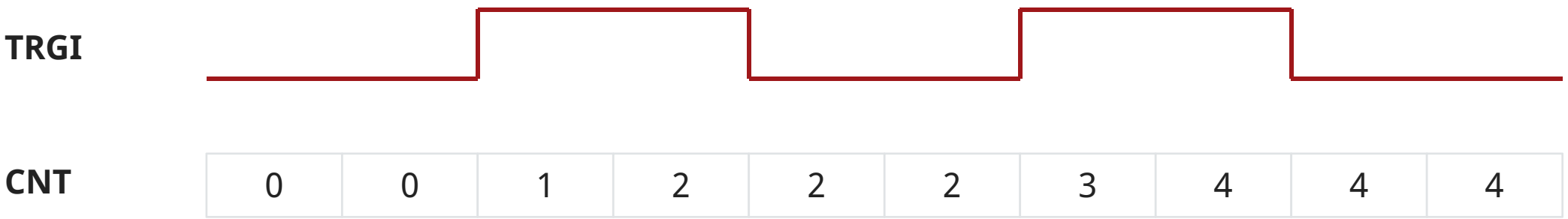
触发负责“清零重新计时”；两次触发之间的快慢仍由 **CK_INT / PSC** 决定。

教学结构/时序示意 | AN4013 §3.3；RM0008 Rev9 图135

Gated Mode：允许时计数，不允许时保持

SMS=101；本例 CEN=1，TRGI 高电平为有效窗口



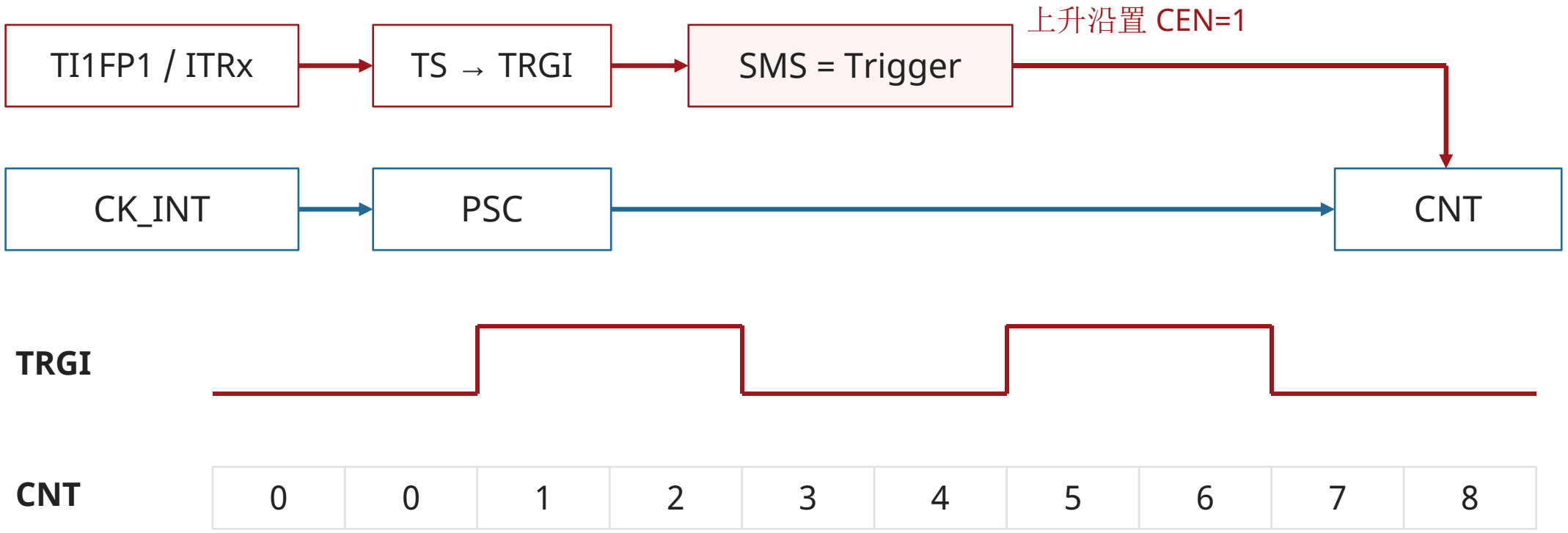


门关：保持原值；门开：继续计数。门控暂停不等于强制 **PWM** 输出低电平。

教学结构/时序示意 | AN4013 §3.3；RM0008 Rev9 图136

Trigger Mode：触发启动，不是反复重启

SMS=110；初始 CEN=0，且本例未开启 One Pulse Mode



它回答“何时开始”；不是 Reset，也不是“TRGI 低就停”的 Gated。

依据 AN4013 §3.3：F1 HAL_TIM_Base_Start 的 Trigger 模式分支

同一个输入，四种 CNT 行为

忽略同步延时与溢出；每一列代表一个示意时间格



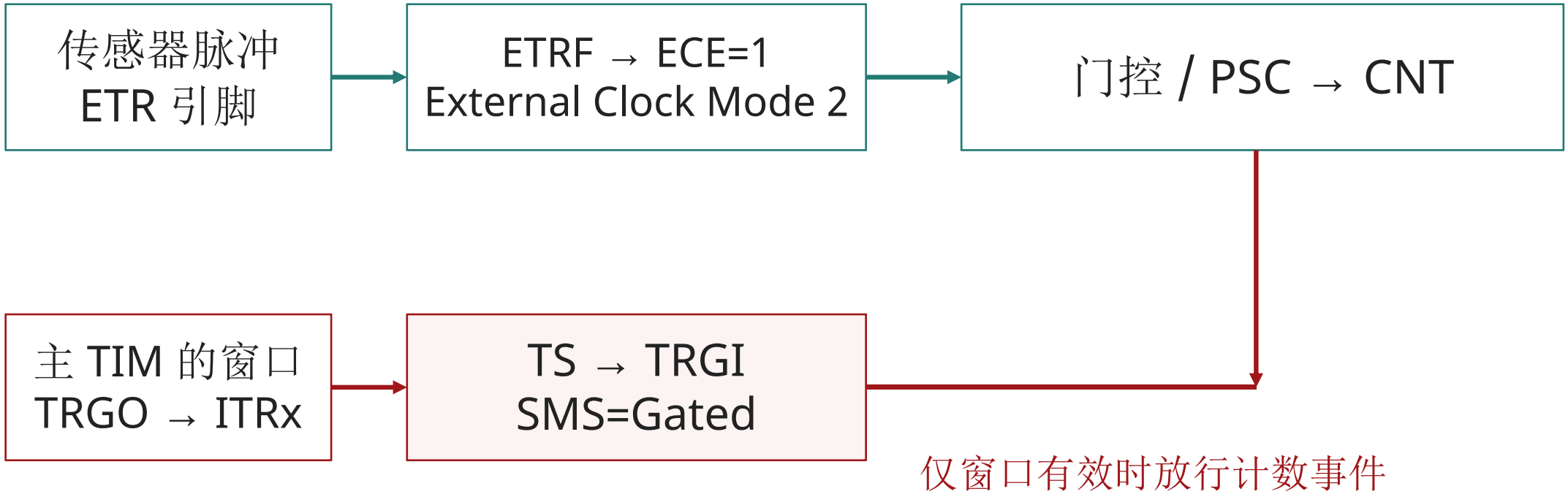
External 1	0	0	1	1	1	1	2	2	2	2
Reset	0	1	0	1	2	3	0	1	2	3
Gated	0	0	1	2	2	2	3	4	4	4
Trigger	0	0	1	2	3	4	5	6	7	8

对照时一次只改 SMS；否则很难分清差异来自模式还是来自时钟。

依据官方各模式语义制作的教学示意；不是原图复刻或实测波形

组合示例：ETR2 负责“数”，TRGI 负责“门”

计数来源与运行控制可以是两个不同信号



不要把“在哪里数脉冲”与“什么时候允许数”合并成一个问题。

教学重绘丨RM0008：External clock mode 2 与其他从模式组合

PSC、ETPS、CKD、输入滤波不是一回事

先沿结构图找它在哪一级，再判断它改变什么

对象	所在路径	用途
ETPS	ETR → ETRP	仅对 ETR 事件作 /1、/2、/4、/8
PSC	CK_PSC → CK_CNT	选中的计数事件统一除以 PSC+1
CKD	CK_INT → tDTS	影响滤波 / 死区等采样时基，不是 PWM 再分频
ETF / ICxF	ETR 或 TIx 输入整形	抑制不满足采样条件的脉冲，并带来延迟

外部信号也要同步和滤波；不能认为 GPIO 上每个极窄脉冲都一定被计到。

依据 AN4776 §1.4.1、§2.2；RM0008：PSC / CKD / ETPS / ETF

把 CubeMX 菜单放回结构图中

配置不是背选项，而是指出这个选项控制哪一段路径

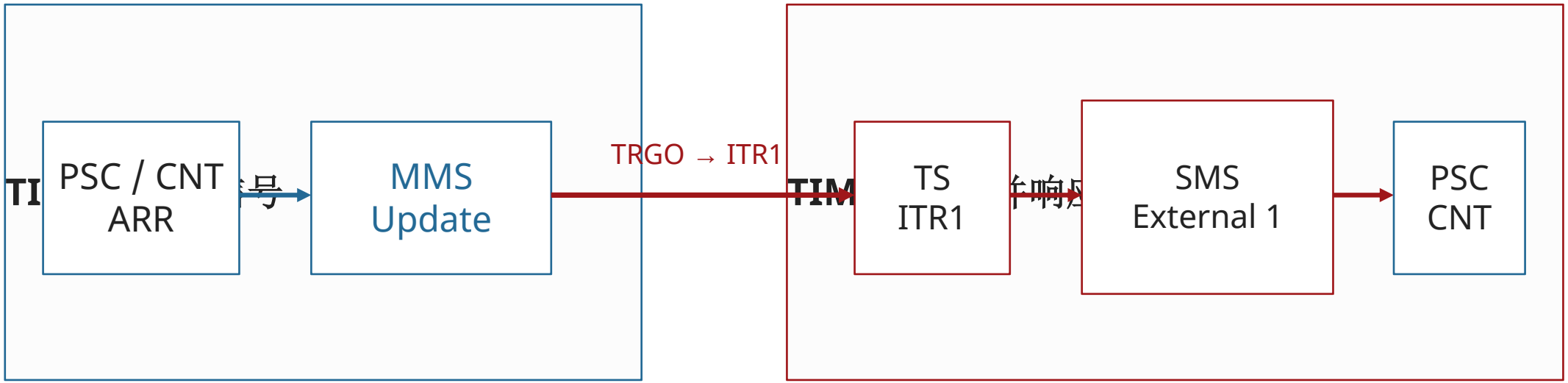
CubeMX 配置	硬件字段	在结构图上控制什么
Trigger Source	SMCR.TS	哪个来源接到 TRGI
Slave Mode	SMCR.SMS	TRGI 到来后如何影响计数
Clock Source = ETR2	SMCR.ECE	ETRF 是否进入专用时钟路径
Master Output Trigger	CR2.MMS	哪个内部事件 / 电平送到 TRGO
Master/Slave Mode	SMCR.MSM	主从级联的同步协调

先画清楚信号路径，再去找对应菜单；不要按截图位置机械照点。

依据 F103 RM0008 与 STM32F1 HAL 配置结构

实验 C：片内级联，1 kHz 事件变成 100 Hz PWM

不用额外连线：TIM2 发 Update，TIM3 接 ITR1 并输出 PA6



TIM2 主定时器	TIM3 从定时器	预期输出
Internal Clock；PSC=71； ARR=999 MMS=Update	TS=ITR1；SMS=External 1 PSC=0；ARR=9；CH1 PWM1；CCR1=5	PA6：100 Hz / 50% 每10次 TRGO 完成一轮

实验设计依据 RM0008 的 timer synchronization 与 TIM3 ITR1 映射

实验 C：先准备接收方，再启动发送方

所有 MX_..._Init() 完成后执行；不需要靠 CPU 中断转发触发

```
1 __HAL_TIM_SET_COUNTER(&htim2, 0);
2 __HAL_TIM_SET_COUNTER(&htim3, 0);
3 HAL_StatusTypeDef status;
4 status = HAL_TIM_PWM_Start(
5     &htim3, TIM_CHANNEL_1);
6 if (status != HAL_OK) Error_Handler();
7 status = HAL_TIM_Base_Start(&htim2);
8 if (status != HAL_OK) Error_Handler();
```

验收
T = 10 ms
tHIGH = 5 ms

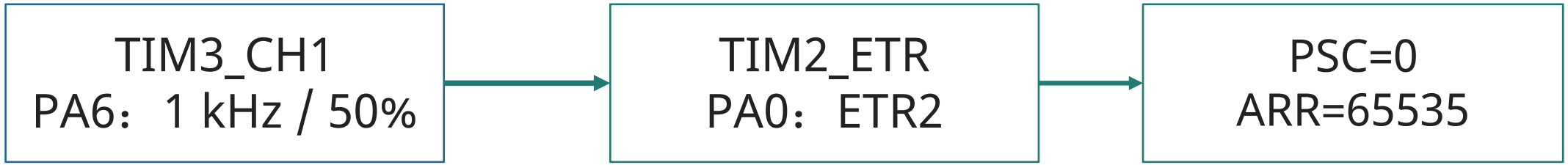
只把 TIM3 的 ARR 改成 19、CCR1 改成 10，再预测并验证输出。

拓展答案：50 Hz / 50%。变化来自下游周期，不是主 TIM 的 1 kHz 改了。

HAL 片段；非独立工程；参数计算与官方主从级联路径一致

实验 D：把 PWM 接回 ETR，统计真实边沿

改用独立配置：TIM3 输出 1 kHz；TIM2 使用 ETR2



杜邦线 + 共地

1. 先运行接收计数器，再启动 TIM3 PWM；检查所选 ETR 极性。
2. 用程序固定采样间隔，计算计数增量；1 s 内预期约 1000 次。
3. 只把 ETPS 改为 /2：在相同采样间隔内，预期计数约减半。

实验设计依据 F103 ETR 默认映射与 AN4776 外部时钟模式2

实验 E：同一输入，只改 Reset / Gated / Trigger

TIM3 产生 10 Hz / 50% 波形；TIM2 以 10 kHz 内部节拍运行

PA6：10 Hz / 50% → PA0 / TI1FP1 → TIM2 TRGI

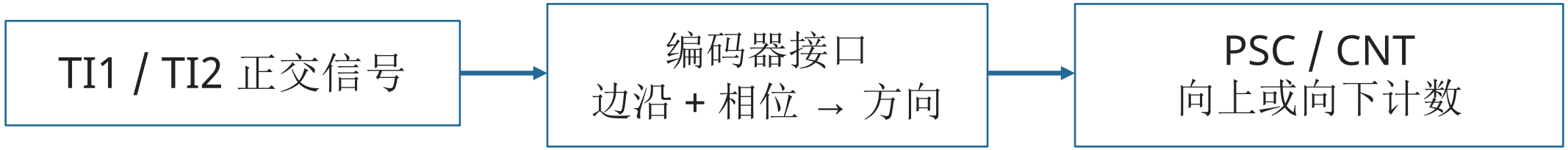
只改 SMS	保持相同的时钟与输入	应该观察到
Reset	TIM2：PSC=7199；ARR=65535	每100 ms重启；计数约0...1000
Gated	TIM2：CEN=1；TRGI高有效	高50 ms约加500；低50 ms保持
Trigger	TIM2：起始 CEN=0；OPM=0	第一次有效边沿后持续计数

TRGI 来源和计数时钟不变，才能把现象差异归因到 Slave Mode。

实验参数按 F103 时基公式设计；模式行为依据 AN4013 §3.3

编码器与辅助选项：放回各自的模块

不要把页面上的每个复选框都当成“另一种时钟源”

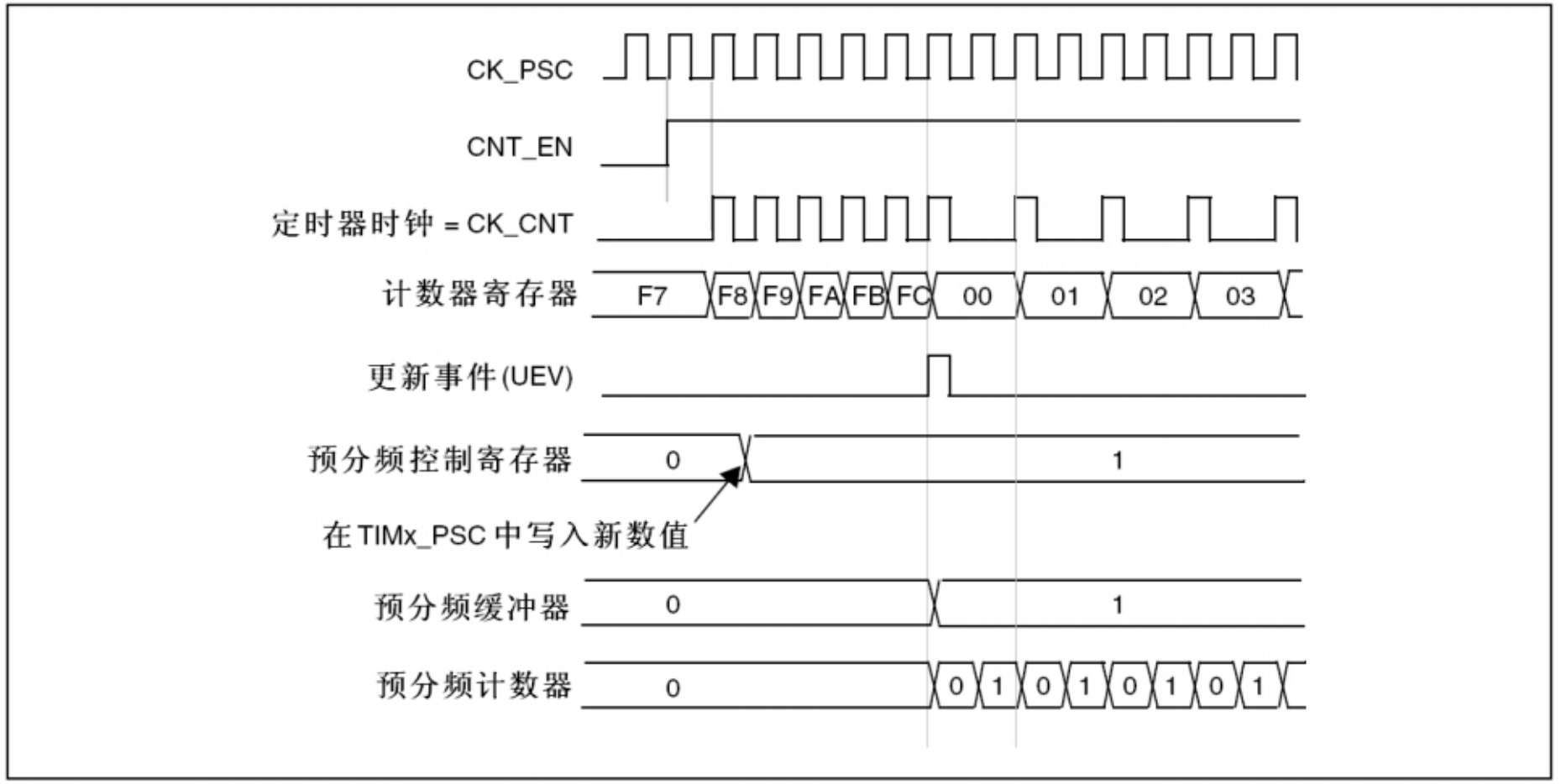


选项	作用位置	不要误解成
XOR activation	输入组合逻辑	新的独立晶振 / 时钟源
Use ETR as Clearing Source	OCxREF 清除控制	ETR2 外部计数的同义词
One Pulse Mode	更新时停止计数的控制	Trigger 模式每个触发都自动重启

依据 F103 RM0008：Encoder interface、OCREF clear、One-pulse mode

预分频器时序

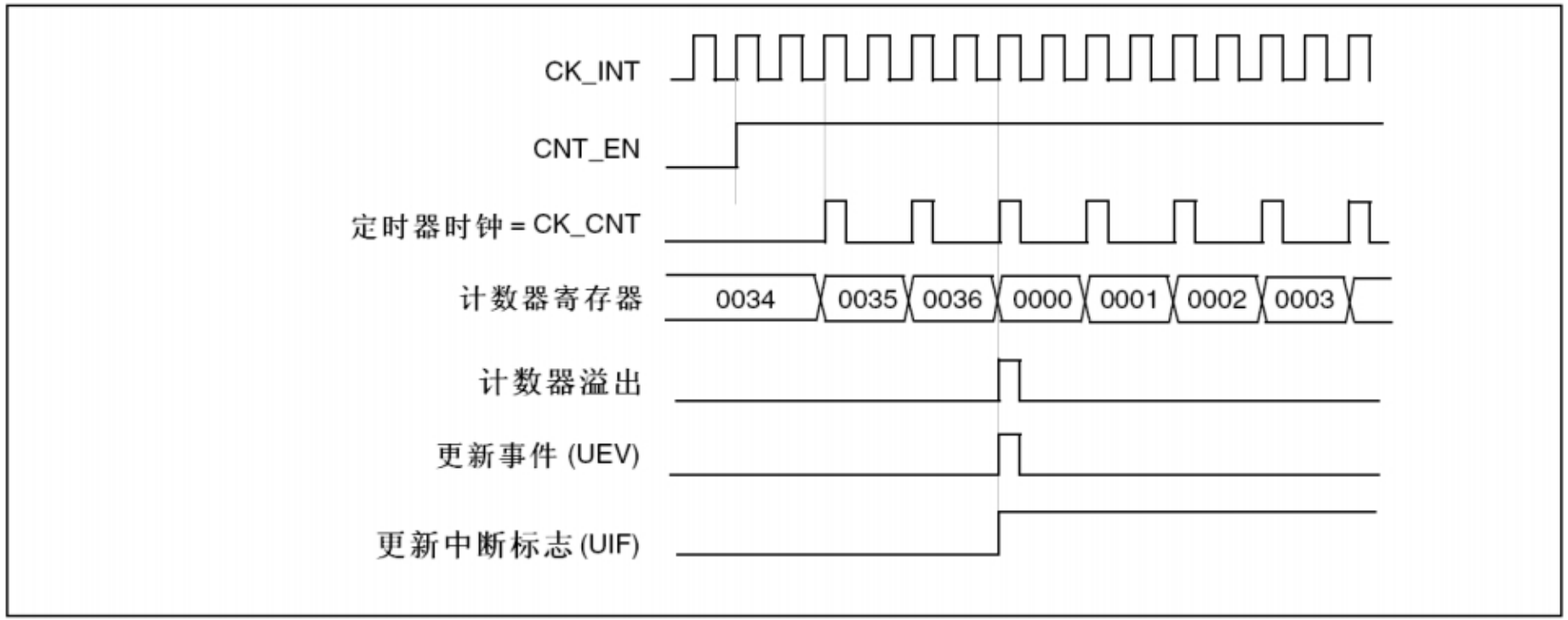
图99　当预分频器的参数从1变到2时，计数器的时序图



- 计数器计数频率： CK_CNT = CK_PSC / (PSC + 1)

计数器时序

图102　计数器时序图，内部时钟分频因子为2



- 计数器溢出频率：CK_CNT_OV = CK_CNT / (ARR + 1)
  = CK_PSC / (PSC + 1) / (ARR + 1)

配置前，先排除这五种混淆

1. “Internal Clock 就是 HSI” → 它是 RCC 提供的 CK_INT。
2. “ITR1 就是 TIM1” → 路由由接收定时器的连接表定义。
3. “有 PWM 就必须开启主模式” → PWM 输出不依赖主从同步。
4. “Trigger 每次清零” → 清零是 Reset；Trigger 负责启动。
5. “Gated 停止就是灯灭” → 它保持计数，未必关闭输出。

概念核对依据 AN4013 §3 与 F103 RM0008

配置练习：按结构图给出路径，不只报菜单名

先独立回答，再解释每个字段作用在哪一级

1. TIM2 每1 ms的更新事件，要让 TIM3 每10次完成一个周期，怎样配置？
2. 外部 ETR 脉冲需要只在另一 TIM 给出的窗口内计数，用哪两条路径？
3. 希望输入每次上升沿都让 CNT 从0重新计时，该用哪种从模式？
4. 希望第一下触发启动，以后输入变低也继续计数，该用哪种从模式？
5. 主定时器应发“Update脉冲”还是“OCxREF电平”来产生门控窗口？
6. 只是用 PA6 做呼吸灯，是否必须配置 MMS、ITR 和 Slave Mode？

参考答案：从源头一直追到 CNT / TRGO

题号	结构路径与关键配置
1	TIM2 MMS=Update → TIM3 ITR1 / TS → SMS=External1；TIM3 PSC=0、ARR=9
2	ETRF → ECE=1 提供计数；另一 TRGO → ITR / TS → SMS=Gated 控制窗口
3	TI1FP1 等有效输入 → TRGI → SMS=Reset → 复位 CNT / PSC
4	TRGI → SMS=Trigger；起始 CEN=0，未启用 One Pulse Mode
5	通常用 OCxREF 等持续窗口电平；Update 是离散事件脉冲

MMS 管输出，ITR 管连线，TS 管输入选择，SMS 管响应动作。

官方原图索引与使用边界

AN4776 Rev4	图8，p.19：外部时钟1 / 2与输入路径
AN4013 Rev14	图4，p.25：TRGI、触发控制器与TRGO
RM0008 Rev9（历史版）	图99 p.321；图139 p.350；表76 p.359
RM0008 官网文档	实际 F103 功能、寄存器、连线与限制以此核对