

第六章 通用定时器 GPTM

定时器是计算机应用中必不可少的外设部件。定时器，顾名思义，就是用来做定时的，使用它可以产生一些周期性的事件，如定时中断/延时等。所谓通用定时器，是相对于一些高级定时器而言的，通用定时器功能相对简单，一般只提供定时/计数等基本功能，而某些高级定时器可能还提供可编程计数器阵列等特殊功能。

1、定时器简介

定时器具有定时和计数功能，它们的功能一般是复用的，所以定时器一般也写作定时/计数器，计数功能一般是记录某个引脚上输入的外部脉冲的个数，而定时功能则是和 SysTick 定时器类似，由系统时钟提供时钟源，触发计数器递减或递增计数，当计数值从最大值减到 0 或由 0 增到设定值时产生中断，并重新开始计数。实际上 GPTM 的功能远比 SysTick 复杂。

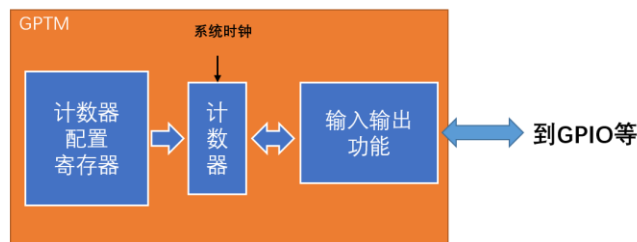


图 1 定时器基本结构示意图

图 1 为定时器的基本结构示意图，图中的计数器里存放着一个数，这个数可以在系统时钟信号的触发下增加或减少；计数器的配置寄存器是通过一系列寄存器来配置计数器的模式等；输入输出功能结合 GPIO 来完成一些操作，例如：产生周期性事件（定时输出某种电平）、捕获外部输入引脚事件的发生时刻（捕获上升沿或下降沿）。

对于定时器，每隔 1 个系统时钟周期计数值加 1 或者减 1，假如一个系统时钟周期为 1ms（系统时钟周期取决于系统时钟），当计数值从 0 加到 1000 或者从 1000 减到 0，我们就认为经过了 1s，这就是定时器的原理。

2、定时器的工作模式

2.1 定时器模块（GPTM）模块特点

TM4C123GH6PM 共包括 6 个 16/32 位 GPTM 模块和 6 个 32/64 位的宽 GPTM 模块，其中每个 16/32 位 GPTM 模块都包含 2 个 16 位的定时器即定时器 A（TimerA）和定时器 B

（TimerB），它们可以独立作为16位的定时器使用，也可以合并为1个32位的定时器使用。同样32/64位的宽GPTM模块包含2个32位的定时器，它们除了可以独立使用外，也可以合并为1个64位的定时器使用。

当16/32位GPTM模块独立地运用作2个16位定时器时，还可以使用8位预分频器将定时器扩展为24位；而当32/64位GPTM模块用作32位定时器时，还可以使用16位预分频器将定时器扩展为48位。

TM4C123GH6PM的GPTM模块还具有以下特点：

- （1）每个GPTM模块有2个CCP引脚。CCP引脚即Capture/Compare/PWM引脚，具有捕获/比较/PWM输出等功能；
- （2）每个模块都有5种运行模式，分别为：单次/周期性定时模式，RTC实时模式，输入边沿计数模式，输入边沿计时模式，PWM模式；
- （3）可以递增（向上）计数，也可以递减（向下）计数；
- （4）中断可以产生ADC及uDMA的触发信号；
- （5）可以同步不同的GPTM模块，使之同时开始计数；
- （6）可以选择链式触发模式。

每个GPTM 模块的主要组成部分是两个自由运行的向上/向下计数器，它们可工作在5种不同的模式：单次/周期定时模式、RTC模式、输入边沿计数模式、输入边沿计时模式和PWM模式，各个模式的工作原理和配置不同，功能也不同。

各种模式具体介绍如下。

2.2 定时器工作模式

2.2.1 单次/周期定时器模式

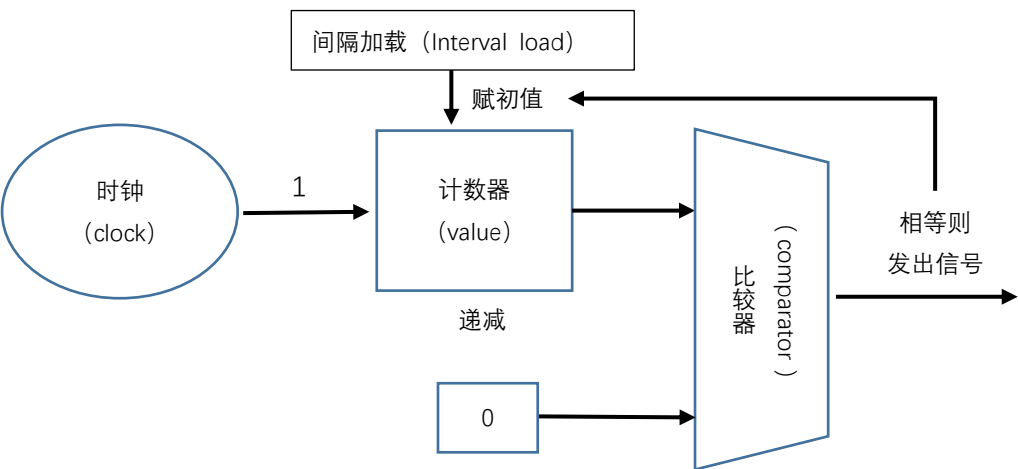


图2 定时器基本模型

图2为定时器的基本模型。当GPTM定时器启用时，定时器在时钟作用下开始从0 向上

计数或者是从预装载值开始向下计数。不论是递增（向上）计数，即从0增加到预装载值Interval load，还是递减（向下）计数，即从预装载值Interval load减至0，当计数完成后，计数器都将回到初始值0（递增模式时）或是预装载值Interval load（递减模式时），并产生一个超时中断。

当定时器向下计数并达到超时事件（计数到0），定时器会为下一个循环重新装载值，这个值从GPTMTnILR寄存器（不使用预分频器扩展时）或者GPTMTnILR寄存器和预分频寄存器GPTMTnPR（使用预分频器扩展时）组合得到。当定时器向上计数并达到超时事件（达到预装载值），定时器重新装载0。如果配置的是单次模式，定时器就会停止计数。如果配置成周期模式，定时器会进入下个周期继续开始计数。

除此之外，在计数过程中，当计数器中的数值与匹配值（存放于GPTMTnMACHER）相同时，也同样可以产生一个匹配中断。具体产生的是哪种中断可以通过寄存器GPTMRIS中的TnMIE位或TnTORIS位查询。

图3分别表示的是递增（向上）计数和递减（向下）计数时，计数值的变化规律。



图3 周期定时模式计数值变化规律

图3中寄存器GPTMTnILR中存放的是预装载值，寄存器GPTMTnMACHER中存放的就是预设的匹配值，图3是在没使用预分频器时的情况。如果使用预分频器，则预设的匹配值为预分频器GPTMTnPR和预分频匹配寄存器GPTMTnPMR组合的值。

当定时器独立工作时，可以使用预分频器将计数器的位数由 16/32 位扩展为 24 位/48 位。下面说明预分频器和匹配预分频器的作用。

当 TimerA 和 TimerB 独立工作时，它们作为 16 位或 32 位的定时器使用。而预分频器可以将 16/32 位定时器扩展为 24/48 位的定时器，此时预设值由间隔加载寄存器和预分频器的值共同组成，匹配值由匹配寄存器和预分频匹配寄存器的值共同组成。预分频器只在 TimerA 和 TimerB 独立工作时有有效。

2.2.2 实时时钟定时器模式

实时时钟定时器RTC模式将产生与实际时钟一致的标准时间信号。

在实时时钟模式下，定时器A和定时器B被合并为一个定时器使用（即32位或64位）且其计数器被配置为向上（递增）计数，计数器中的初始值被设定为0x1。

RTC模式需要一个32.768kHz的外部时钟信号，通过CCP0引脚输入作为时钟源。该时钟信号首先被分频到1Hz，然后输入给计数器进行计数，那么计数器每加1，表示1s的时

间，这样就可以达到计算标准时间的功能。

同样，当GPTMCTL中的TnEN（定时器启用）置位时，计数就会正式开始。当计数值达到匹配值或是计数值达到最大值时，也会产生匹配中断和超时中断。

2.2.3 输入边沿计数模式

输入边沿计数模式用于统计输入信号的边沿信息，可以通过它统计由CCP引脚输入的信号的边沿数目，并且当输入信号的边沿数达到预设值时，计数器还会产生一个中断信号（捕获中断）。

检测上升沿时，输入信号必须在上升沿之后保持至少两个系统时钟周期的高电平，检测下降沿时，输入信号必须在下降沿之后保持至少两个系统时钟周期的低电平。根据这个标准，边沿检测的最大输入频率是系统频率的1/4。

在输入边沿计数模式时，可以选择的边沿有上升沿，下降沿或双边沿，通过配置寄存器GPTMCTL的TnEVENT位来选择。同样，当GPTMCTL中的TnEN置位（定时器启用）时，计数就会开始。在递增计数模式下，计数器从0一直计数到等于匹配值时，产生一个捕获匹配中断信号，然后计数归0，继续计数。

图4中描述了边沿计数递增（向上）计数时的定时器模型。

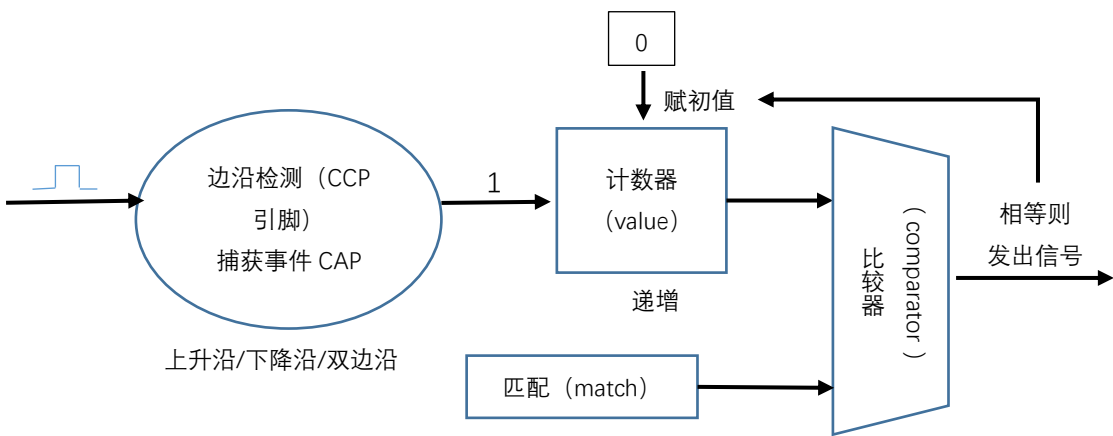


图4 输入边沿计数模式模型

而在递减计数模式下，计数器从预设值开始计数，直到与匹配值相等时，产生一个捕获匹配中断。之后由于GPTM自动将GPTMCTL寄存器的TnEN清零，因此计数器将停止计数，这一点与递增计数模式不同。

图5中表示了递增（向上）和递减（向下）2种方式时计数值的变化规律。图中的计数值是在没使用预分频器时的情况，如果使用预分频器，则要算上预分频器的值。

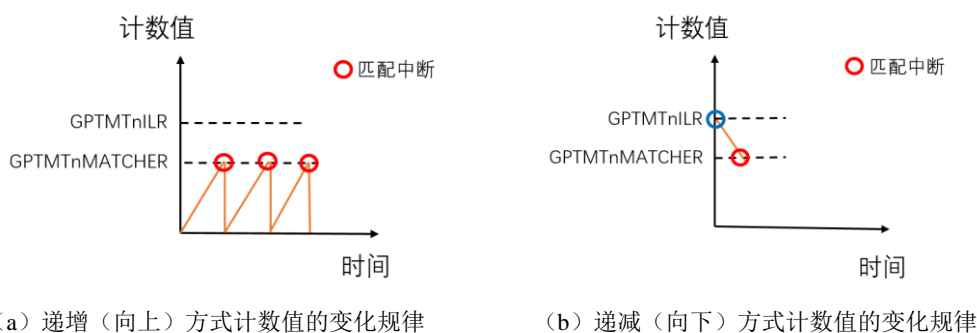


图5 边沿计数模式时计数值的变化规律

当发生捕获匹配中断时，GPTMTnR和GPTMTnPS记录了此事件发生时计数器的值和预分频器的值（事件发生时的值）。而GPTMTnV寄存器和GPTMTnPV寄存器会保存定时器当前运行的计数值和预分频器的值（运行时的实时值）。

图6描述了输入边沿计数模式工作原理。这个例子中，采用递减计数模式，上升沿和下降沿都被计入，定时器开始值设置为GPTMTnILR=0x000A，而匹配值设置为GPTMTnMATCHR=0x0006，那么就会计数四个边沿事件。注意最后两个边沿没有被计数，因为在当前计数值和GPTMTnMATCHR 中值匹配后，定时器会自动清除TnEN 位，这样计数器会忽略之后的边沿事件。

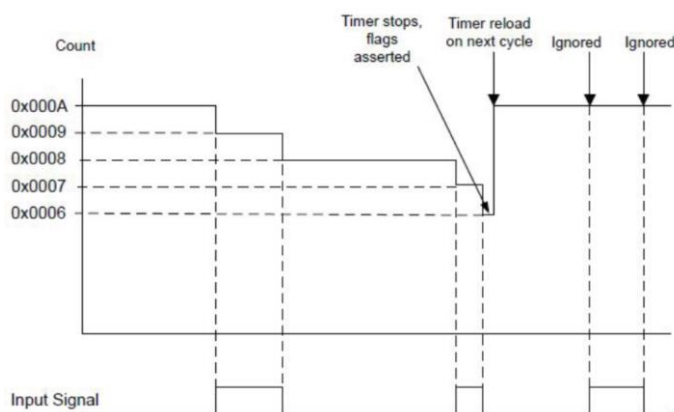


图6 输入边沿向下计数模式

2.2.4 输入边沿计时模式

输入边沿计时模式用于捕捉输入信号的边沿时间，当输入信号有一个待捕捉的边沿时（上升沿或下降沿），计数器会产生一个中断信号。

同样边沿检测的最大输入频率是系统频率的1/4。

在输入边沿计时模式下，可以选择的边沿同样有上升沿，下降沿或双边沿，通过配置寄存器GPTMCTL的TnEVENT位来选择。当定时器器独立工作时，可以使用预分频器将计数器的位数由16/32位扩展为24位/48位。

GPTMCTL中的TnEN被置位，计时开始。当有捕获事件（捕获到边沿信号）发生时，

计数器的数值会被存储在寄存器GPTMTnR和 GPTMTnPS中，并且可以被微处理器读取，之后产生一个捕获事件中断用于处理捕获事件。处理完成后计数继续进行，直到GPTMCTL中的TnEN位被置0。而GPTMTnV寄存器和GPTMTnPV寄存器会保存当前独立运行的计数值和预分频器的值。当计数从0达到了计数的初值（递增计数模式，由装载寄存器设置）或是由初值达到0（递减计数模式）时，计数器重新装载初始值继续计数。

下面举一个例子进行说明：采用递减计数模式，只捕获上升沿，计数器预设值为GPTMTnILR=0Xffff，每当捕获一个上升沿时，计数器中的数值被写入GPTMTnR和GPTMTnPS寄存器（当使用预分频器时才使用此寄存器）中，时序图如图7所示。

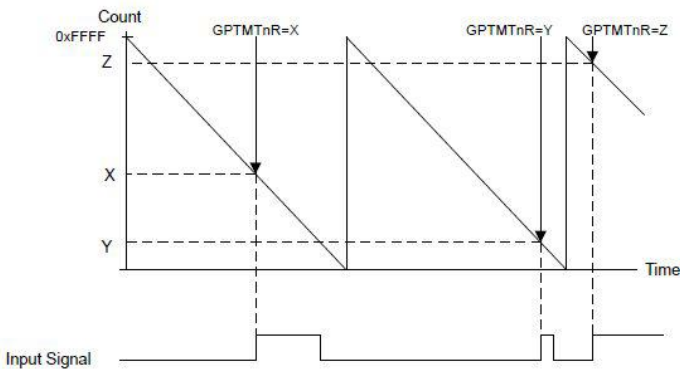


图7 输入边沿计时模式实例

2.2.5 PWM 模式

GPTM支持简单的PWM模式，可以产生PWM波形。

在PWM模式中，计数器被配置为递减计数模式。计数器的位数为16位或32位，也可以使用预分频器扩展计数器的位数为24位或48位。

当GPTMCTL中的TnEN被置位，计数器开始递减计数，直到计数值为0，然后重新装载寄存器初值。

图8为PWM模式时的模型，PWM的输出信号在计数器重新装载初值时会发生一次翻转，当计数值达到匹配值时又会翻转回来，这样就产生了PWM信号。另外，可以通过设置寄存器GPTMCTL中的TnPWML位使输出的PWM信号进行翻转。还可以设置PWM在输出上升沿、下降沿或双边沿时产生中断。

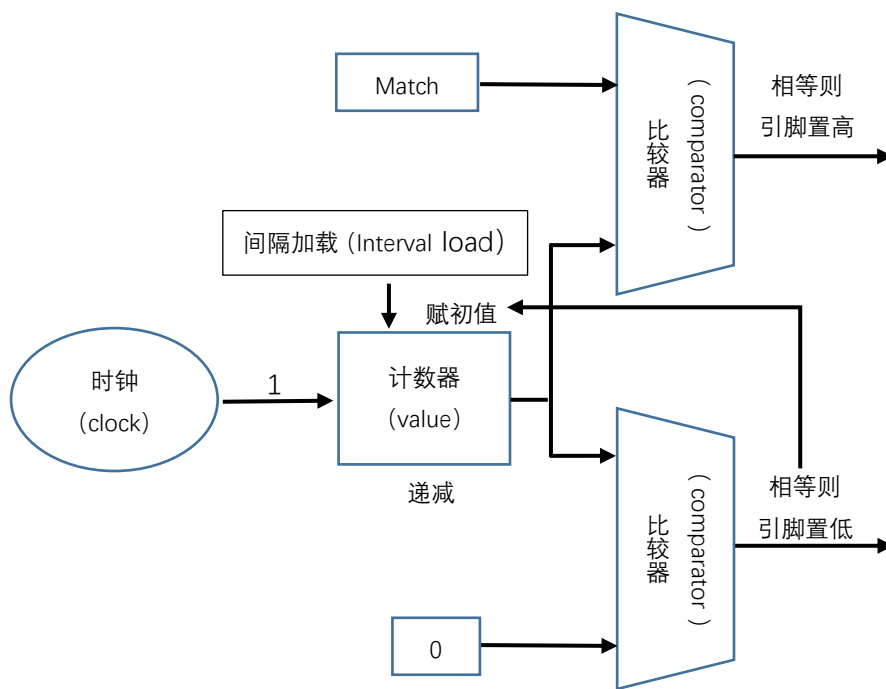


图8 PWM模式定时器的模型

下面举一个例子：预设值为0xC350，即寄存器GPTMTnILR=0xC350（十进制为50000），匹配值为0x411A（十进制为16666），即寄存器GPTMTnMATCHR=0x411A。生成PWM的时序图如图5所示。

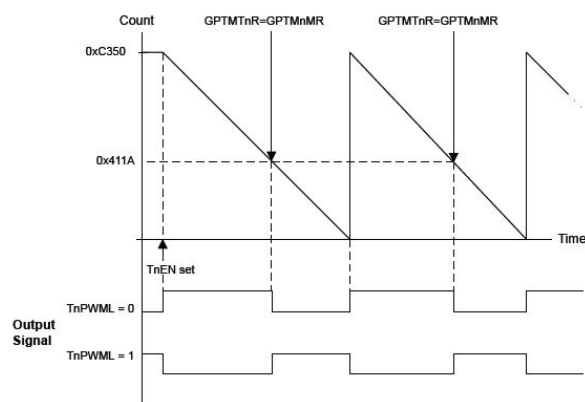


图9 PWM模式实例

图9显示了如何产生一个1ms周期，且占空比（高电平占整个周期的比例）为66%的PWM波形，假定系统时钟为50MHz，并且TnPWML=0（如果TnPWML=1的时候占空比为33%，输出波形与TnPWML=0时正好相反）。

2.3 定时器模式总结

2.3.1 定时器不同工作模式特点

定时器5种工作模式在独立运行和级联运行时各寄存器的大小如表1所示。

表1 定时器5种工作模式特点

模式	定时器使用	计数方向	计数器大小		预分频器大小*		预分频器状态（计数方向）
			16/32 位 GPTM	32/64 位宽 GPTM	16/32 位 GPTM	32/64 位宽 GPTM	
单次	单独	向上/向下	16 位	32 位	8 位	16 位	定时器扩展（向上） 预分频器（向下）
	级联	向上/向下	32 位	64 位	-	-	N/A
周期	单独	向上/向下	16 位	32 位	8 位	16 位	定时器扩展（向上） 预分频器（向下）
	级联	向上/向下	32 位	64 位	-	-	N/A
RTC	级联	向上	32 位	64 位	-	-	N/A
边沿计数	单独	向上/向下	16 位	32 位	8 位	16 位	定时器扩展（向上/向下）
边沿定时	单独	向上/向下	16 位	32 位	8 位	16 位	定时器扩展（向上/向下）
PWM	单独	向下	16 位	32 位	8 位	16 位	定时器扩展

2.6.2 应用举例

5 种工作模式可以有不同的用途：

- （1）单次/周期定时模式：定时器最基本的功能，定时产生中断。
- （2）实时时钟模式：外接一个 32.768kHz 的晶振，分频 2 的 15 次方，成为 1Hz 的时钟，用于电子日历等需要实时时钟的地方。
- （3）输入边沿计数模式：统计一段时间内一个外部信号的周期个数，从而求出该信号的频率。
- （4）输入边沿计时模式：可以得知外部信号相邻几个边沿(上升沿/下降沿均可)到来时，间隔的时间。从而求出占空比或周期等信息。
- （5）PWM 模式：输出 PWM 波。

2.6.3 定时器中断类型

总结一下，定时器能产生的中断有：

- (1)超时中断：计数器增长到最大值/减小到最小值时发生中断。
- (2)匹配中断：计数器和匹配值相同时发生中断。
- (3)捕获事件中断：检测到输入信号的边沿。
- (4)捕获匹配中断：发生捕获事件（检测到信号边沿）时，计数器和匹配值相同时发生中

断。

(5)RTC 中断：发生超时中断或匹配中断。

(6)PWM 中断：当输出引脚产生边沿（可以设置为上升沿/下降沿/双边沿）时发生的中断。

那么，定时器在 5 种不同模式可以产生的中断类型如表 2 所示。

表 2 定时器不同模式可以产生的中断类型

模式	中断类型
单次/周期	超时/匹配中断
RTC	RTC 中断（超时/匹配中断）
边沿计时	捕获事件中断
边沿计数	捕获匹配中断
PWM	PWM 中断

3、定时器 GPTM 模块结构及相关寄存器

3.1 GPTM 模块的结构

GPTM 模块部分结构框图（选取定时器 A 的结构，定时器 B 是一样的）如图 2 所示。

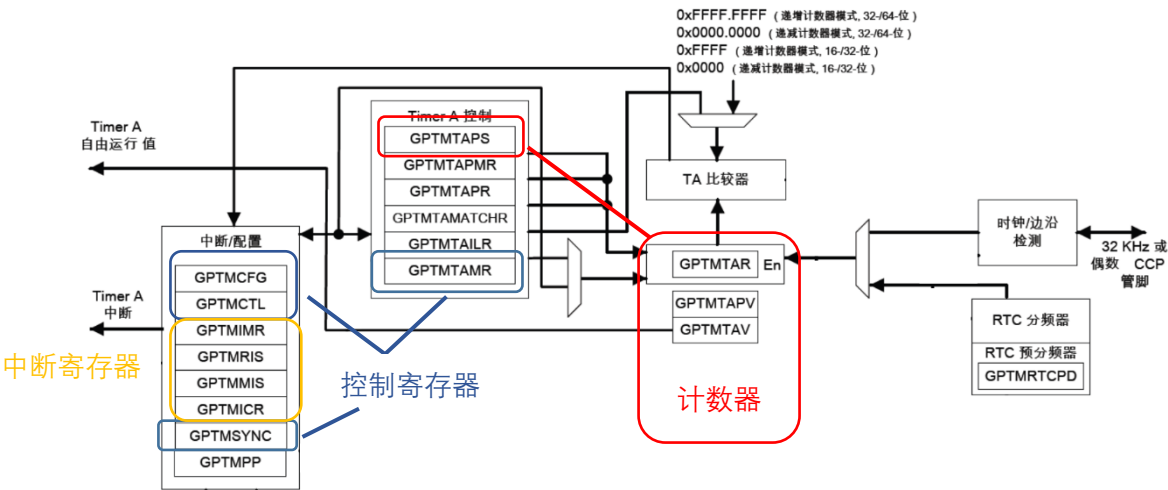


图 10 GPTM 结构框图

GPTM 模块和宽 GPTM 模块的结构相似，因此只介绍 GPTM 模块的结构。每个 GPTM 模块分为三个部分：控制寄存器、计数器/比较器以及输入输出控制。

图 10 中左侧的寄存器主要有两种：中断/配置寄存器以及 Timer A 和 Timer B 的控制寄存器。其中中断/配置寄存器包括多个寄存器，主要用于 GPTM 模块的中断控制以及计数器计数模式的整体配置，如 Timer A 和 Timer B 是否独立运行等。Timer A 和 Timer B 控制寄存器则用于具体运行模式的控制，例如 Timer A 是运行于单次计数模式还是周期性计数

模式。

图 10 中右侧的输入输出控制模块主要为时钟/边沿检测模块，用于 CCP 引脚的输入，或是输出 PWM 波形到 CCP 引脚。若作为输入引脚，CCP 具有捕获（Capture）功能，即可以捕捉 CCP 输入引脚上的边沿。而作为输出引脚，CCP 具有比较（Compare）和 PWM 波形输出两种功能，这两种功能相似，都是将计数器的值与某个值进行比较，然后根据结果输出相应的矩形波。

图 10 中间部分是 GPTM 模块的核心部分，即计数器和比较器的执行单元，其中计数器根据预先设定的运行模式进行递增和递减计数，所得计数值在比较器中与预先设定的特定值进行比较，进而可以产生中断及触发信号。此外，CPU 还可以直接读取计数器的计数值用作其他功能。

3.2 GPTM 寄存器分类

Timer0到Timer5都是16/32 Bit定时器模块，它们的TimerA和TimerB都是16位，组合起来是32位。WideTimer0到WideTimer5是32/64 Bit宽定时器模块。

按照功能结构图可以将每个GPTM模块的寄存器分为4类：控制类寄存器，装载类寄存器，数值类寄存器和中断类寄存器。下面分别介绍这些寄存器的功能，具体的寄存器定义可以查询数据手册。

1、控制类寄存器

控制类寄存器有3个，分别为配置寄存器GPTMCFG、控制寄存器GPTMCTL和模式寄存器GPTMTnMR。

GPTM控制类寄存器，有的可以一个寄存器同时控制TimerA和TimerB，例如GPTMCTL控制寄存器，它的[6:0]位控制TimerA，[14:8]位控制TimerB。例如第0位是TAEN，第8位是TBEN，后面都用TnEN来代表这两个位。

有的寄存器分成了A，B两个，例如模式寄存器GPTMTAMR和GPTMTBMR。因为功能是一样的，所以后面全都用类似于GPTMTnMR（n表示A或B）寄存器这样的名字来代表这个寄存器。

GPTMTnMR寄存器用来配置定时器的模式，递增/递减计数和中断类型等，控制寄存器GPTMCTL通常来控制一些细节等。

配置寄存器(GPTMCFG)决定是否要把TimerA和TimerB级联。

表3 控制类寄存器

寄存器名称	功能
配置寄存器(GPTMCFG)	决定是否要把TimerA和TimerB级联
控制寄存器(GPTMCTL)	决定定时器是否使能及每种模式的细节等
模式寄存器(GPTMTnMR)	配置定时器的模式、递增/递减计数、中断类型等。

2、装载类寄存器

装载类寄存器用来存放计数的初始值，分为间隔加载寄存器 GPTMTnILR，匹配寄存器 GPTMTnMATCHR，预分频寄存器 GPTMTnPR，预分频匹配寄存器 GPTMTnPMR。

表 4 装载类寄存器

寄存器名称	功能
间隔加载寄存器 GPTMTnILR	存放计数器的计数初值。
匹配寄存器 GPTMTnMATCHR	存放匹配值，计数器中的值会实时的和它作比较。
预分频寄存器 GPTMTnPR	当 TimerA 和 TimerB 独立使用的时候，可以将定时器的位数进行扩展，与 GPTMTnILR 一起组成一个更大的计数初值。
预分频匹配寄存器 GPTMTnPMR	当 TimerA 和 TimerB 独立使用的时候，可以将定时器的位数进行扩展，它可以设定预分频器的匹配值，常常与 GPTMTnMATCHR 一起使用，组成一个更大的匹配值。

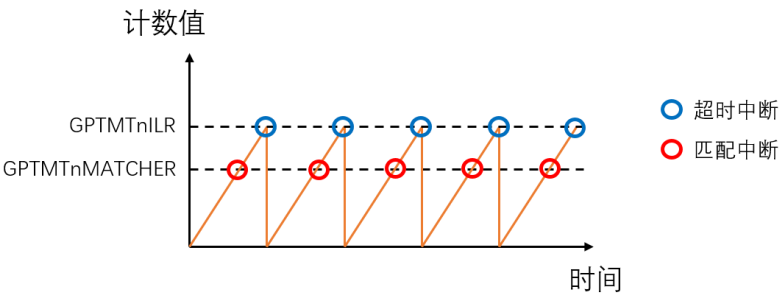


图 11 定时器递增方式下的 2 种计数方式

下面举例说明定时器上述装载类寄存器的作用。定时器有递增和递减 2 种工作方式，以递增方式为例，定时器有 2 种计数方式，一种是计数器的计数值（value）从 0 开始递增，当计数值递增到设定的初值（GPTMTnILR 间隔加载寄存器中设定）时，定时器会产生超时中断；另一种方式是计数值（value）从 0 开始递增，当递增到一个匹配值（还没达到最大值，这个匹配值在匹配寄存器中设定）时，定时器产生匹配中断。计数值的变化规律如图 11 所示。

2、当前数值类寄存器

读取这些值，往往能实时知道定时器当前的计数器的值。

定时器的当前数值类寄存器分为定时器寄存器 GPTMTnR，定时器值寄存器 GPTMTnV，预分频器值寄存器 GPTMTnPV 和预分频快照值寄存器 GPTMTnPS。GPTM 有 5 种工作模式，将在后续介绍。

定时器寄存器 GPTMTnR 和定时器值寄存器 GPTMTnV 有区别，虽然在大部分工作模式（共 5 种工作模式）下，两个寄存器存放的都是当前计数器的值，但定时器寄存器 GPTMTnR 在边沿计时和边沿计数模式下存放的不是当前计数器的值，而是记录了捕获事件发生时边沿产生的时间或边沿的个数。

而定时器值寄存器 GPTMTnV 在任何模式下都是存放的定时器运行时当前计数值，该

寄存器除了读外，还可以写，向 GPTMTnV 写入值时，会在下个时钟周期将当前 GPTMTnV 值写入到定时器寄存器 GPTMTnR 中。

预分频值寄存器 GPTMTnPV 存放的是当前预分频器的值，预分频快照寄存器 GPTMTnPS 存放的是快照（Snapshot）事件发生时预分频器的值。

当有快照（Snapshot）事件发生时，往往 GPTMTnR 和 GPTMTnPS 配合使用，存放快照（Snapshot）事件发生时计数器和预分频器的值，而定时器还会继续工作，GPTMTnV 和 GPTMTnPV 存放的是定时器实时运行时计数器和预分频器的值。

表 5 数值类寄存器

寄存器名称	功能
定时器寄存器 GPTMTnR	一般存放的是当前计数值，边沿计数/计时模式时，存放的是捕获事件发生时的计数值。
预分频快照值寄存器 GPTMTnPS	边沿计数/计数模式时，存放快照事件发生时预分频器的值。
定时器值寄存器 GPTMTnV	永远存放的是定时器运行时当前的计数值
预分频器值寄存器 GPTMTnPV	永远存放的是定时器运行时当前的预分频器的值

3、中断类寄存器

定时器主要有三种中断：

(1)超时中断：计数器和预分频器都增长到设定值/减小到 0 时发生中断。

(2)匹配中断：计数器和预分频器中的值都和各自的匹配寄存器中的值相同时发生中断。

(3)捕获中断：捕获事件发生时产生的中断（后续介绍）。

因此中断类寄存器分为以下几类（跟 GPIO 中断寄存器的分类类似）：分为中断使能、状态和清除寄存器：GPTMIMR（中断屏蔽）,GPTMRIS（原始中断状态）,GPTMMIS（屏蔽中断状态），GPTMICR（中断清除）。

控制类寄存器中有与中断配置相关的寄存器，即模式寄存器 GPTMTnMR，它用来设置定时器的中断事件类型是超时中断或匹配中断还是捕获中断，是否开启 PWM 等。

3.3 GPTM 的引脚分配

表 6 可用的 CCP 和定时器引脚分配

定时器	递增/递减计数器	偶数CCP管脚	奇数CCP管脚
16/32 位 Timer 0	Timer A	T0CCP0	-
	Timer B	-	T0CCP1
16/32 位 Timer 1	Timer A	T1CCP0	-
	Timer B	-	T1CCP1
16/32 位 Timer 2	Timer A	T2CCP0	-
	Timer B	-	T2CCP1

表6列出了部分通用定时器模块的外部CCP（捕获Capture/比较Compare/PWM）引脚，一般偶数管脚分配给TimerA，奇数引脚分配给TimerB。

表7描述了部分GPTM模块可以使用的引脚（全部引脚可以查看英文数据手册706页表11-2）。按照GPIO的引脚配置方法，应将GPIO引脚配置为复用的定时器功能，即GPIO备用功能选择（GPIOAFSEL）寄存器里的相应位应被置位为1，然后配置相应的端口控制寄存器（GPIOPCTL）的位域（引脚后面括号中表示的是应配置的PCMx位域值）。例如：要将PB6配置为T0CCP0引脚，也就是将PB6配置为定时器0的TimerA的CCP引脚，除了将GPIO模块B的GPIOAFSEL寄存器（0：7位分别设置0-7引脚是否为复用功能）的第6位置1之外，还应将该模块的端口控制器GPIOPCTL的PMC6域设置为0x7。

表 7 通用定时器引脚

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 ^a	描述
T0CCP0	1 28	PB6 (7) PF0 (7)	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 0。
T0CCP1	4 29	PB7 (7) PF1 (7)	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 1。
T1CCP0	30 58	PF2 (7) PB4 (7)	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 0。
T1CCP1	31 57	PF3 (7) PB5 (7)	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 1。
T2CCP0	5 45	PF4 (7) PB0 (7)	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 0。
T2CCP1	46	PB1 (7)	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 1。
T3CCP0	47	PB2 (7)	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 0。
T3CCP1	48	PB3 (7)	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 1。

4 定时器的初始化配置

要使用GPTM 模块，需要对它进行初始化和配置。有些配置是相同的，称为通用配置，通用配置结束后，再根据不同的模式作不同的配置。本节还是寄存器配置，相关库函数请查阅pdf文件《TM4C库函数指南》和TIVAWARE中driverlib\timer.c中的函数。

4.1 通用配置

- 1.使能GPTM的时钟信号：配置寄存器RCGCTIMER或RCGWTIMER；实际上调用的是

SysCtlPeripheralEnable（）函数，使能定时器模块；

2.找到该GPTM模块中TnCCP0和TnCCP1（或是WTnCCP0和WTnCCP1）所对应的GPIO引脚，使能这些引脚所属的GPIO模块的时钟信号：配置寄存器RCGCGPIO；实际上调用的也是SysCtlPeripheralEnable（）函数，使能GPIO模块；

3.由于引脚有复用，所以需要将TnCCP0和TnCCP1（或是WTnCCP0和WTnCCP1）所对应的GPIO引脚配置成GPTM功能：配置寄存器GPIOAFSEL；实际上调用的是GPIODirModeSet（）函数，将引脚设置为复用外设功能，如果要设置为定时器功能，可直接调用GPIOPinTypeTimer（）函数；

4.将GPTM模块的信号TnCCP0和TnCCP1（或是WTnCCP0和WTnCCP1）配置到具体的GPIO引脚上去：配置寄存器GPIOPCTL。实际调用的是GPIOPinConfigure（）函数，其参数为GPIO_P??_??，这个函数的参数可以在driverlib/pin_map.h中查到。

除了要作通用配置外，每种定时器根据不同模式还要作具体的初始化配置，以下为配置实例。

4.2 单次/周期定时器模式

GPTM 通过以下顺序配置成单次或者周期模式：

1. 确保定时器处于禁用状态，即确认GPTMCTL中TnEN 位清零。
2. 选择合并计数模式：写0x0000.0000 到GPTM 配置寄存器GPTMCFG。
3. 选择是单次计数模式还是周期性计数模式。

配置GPTM 定时器n 模式寄存器GPTMTnMR 的TnMR 位：

- a. 写0x1 启用单次模式
- b. 写0x2 启用周期模式

4. 配置GPTMTnMR中的TnWOT和TnCDIR 位，这些位确定是否开启触发等待功能（后文介绍），向上还是向下计数。

5. 设置预设值，将预设值加载到GPTM 定时器n时间间隔加载寄存器GPTMTnILR。

6. 使能需要的中断：设置GPTM中断屏蔽寄存器GPTMIMR 的相应中断位。

7. 使能GPTM模块，开始计数：设置GPTMCTL 寄存器中TnEN 位来启用定时器并开始计数。

8. 轮询GPTMRIS 寄存器或者等待中断的产生。这两种情况下，写1到GPTMICR 寄存器的相应位置都会将状态标志清除。

4.3 RTC 模式

要使用RTC 模式，得先有一个32.768KHz的外部时钟信号从偶数CCP引脚输入，然后，按照以下步骤启用RTC功能：

1. 确保在任何更改前定时器是禁用的，即确认GPTMCTL中TnEN 位清零。
2. 如果定时器在此之前是工作在其他不同的模式，在重新配置前需要清除GPTMTnMR寄存器中的任何残留位。
3. 选择RTC计数模式：写0x0000.0001 值到GPTM 配置寄存器GPTMCFG。
4. 写匹配值到GPTM 匹配寄存器GPTMTnMATCHR。
5. 按照需要设置/清除GPTMCTL中的RTCEN位。
6. 如果需要使用中断，则使能中断位：设置GPTMIMR 中的RTCIM 位。
7. 设置GPTMCTL的TnEN 位启用定时器并开始计数。

4.4 输入边沿计数模式

定时器通过以下步骤配置成输入边沿计数模式：

1. 确保定时器在做任何更改是禁用的，即确认GPTMCTL中TnEN 位清零。
2. 选择独立计数模式;写0x0000.0004值到GPTMCFG寄存器。
3. 选择输入边沿计数模式：在GPTMTnMR 寄存器中，写0x0到TnCMR，写0x3到TnMR。
4. 选择要捕获的边沿：写GPTMCTL的TnEVENT 可以设置定时器需要捕获的事件类型。
5. 如果使用预分频器，则将预分频值写入GPTMTnPR 寄存器中。
6. 设置预设值，将定时器的预设值装载入GPTMTnILR 寄存器。
7. 设置预分频器的匹配值：将预分频器的匹配值装载入GPTMTnPMR 寄存器
8. 设置计数匹配值，将计数匹配值装载入GPTMTnMATCHR 寄存器。注意当执行向下计数时，GPTMTnPR 和GPTMTnILR 中的数值必须大于GPTMTnPMR 和GPTMTnMATCHR中的数值。
9. 如果使用中断，需要设置GPTMIMR的CnMIM 位。
10. 设置GPTMCTL的TnEN 位启用定时器并开始等待边沿事件。
11. 轮询GPTMRIS的CnMRIS位或者等待中断。这两种情况下，写1到GPTMICR的CnMCINT位都会将状态标志清除。

当在边沿计数模式向下计数时，定时器在检测到边沿事件数时就会停止。要重新启用定时器，确保将TnEN 位清零并且重复4-9步。

4.5 输入边沿计时模式

定时器按照以下步骤配置成输入边沿定时模式：

1. 确保定时器在做任何更改之前是禁用的，即确认GPTMCTL中TnEN 位清零。
2. 选择独立计数模式：写0x0000.0004 到GPTMCFG寄存器。

3. 选择输入边沿计时模式：在GPTMTnMR 寄存器中，写0x1到TnCMR，写0x3到TnMR。
 4. 选择要捕获的边沿：写GPTMCTL 的TnEVENT 可以设置定时器需要捕获的事件类型。
 5. 如果使用预分频器，将预分频值写入GPTMTnPR 中。
 6. 设置预设值Load Value，定时器的预设值写入GPTMTnILR中。
 7. 如果需要使用中断，那么需要设置GPTMIMR 的CnEIM 位。
 8. 设置GPTMCTL的TnEN 位启用定时器并开始计数。
 9. 轮询GPTMRIS 寄存器的CnERIS 位或者等待中断。这两种情况下，写1 到GPTMICR的CnECINT 位都会将状态标志清除。
- 事件发生的时间可以通过读取GPTMTnR 寄存器得到。
- 在输入边沿定时模式中，定时器在检测到边沿事件后会继续运行。

4.6 PWM 模式

定时器按照以下步骤配置成PWM模式：

1. 确保定时器在做任何更改之前是禁用的，即确认GPTMCTL 中TnEN 位清零。
 2. 选择独立计数模式：写0x0000.0004 值到GPTMCFG 寄存器。
 3. 选择PWM模式：在GPTMTnMR 寄存器中，写0x1到TnAMS，写0x0到TnCMR，写0x2到TnMR。
 4. 设置PWM 信号的输出状态：设置GPTMCTL寄存器的TnPWML位可以配置PWM信号的输出状态。
 5. 如果使用预分频器，将预分频值写入GPTMTnPR 中。
 6. 如果使用PWM中断，则使能中断位：首先通过寄存器GPTMCTL中的TnEVENT位来配置中断产生的条件，然后设置GPTMTnMR 的PWMIE位来使能中断。注意当PWM 输出反转情况下，边沿检测产生中断的条件也会反转。
 7. 设置预设值，将预设开始值写入GPTMTnILR中。
 8. 将匹配值写入GPTMTnMATCHR 中。
 9. 设置GPTMCTL的TnEN 位启用定时器并开始产生PWM 输出信号。
- 在PWM 模式下，PWM 信号产生之后定时器会持续运行。PWM周期可以通过写GPTMTnILR寄存器在任何时间进行更改，这一更改会在定时器的下一个时钟周期生效。
- 注意这五种模式中，PWM 模式始终只能递减计数，而 RTC 模式始终只能递增计数。

5 定时器其他功能

5.1 触发等待功能

触发等待功能允许定时器模块采用菊花链方式连接定时器，一旦配置，就能够使用单个定时器链式触发多个定时器。设置GPTMTnMR中的TnWOT 位可以启用触发等待模式。当设置了TnWOT 位，定时器N+1并不会马上开始计数，它必须等到菊花链中在它之前的定时器N达到它的超时事件。菊花链方式时GPTM1遵循GPTM0 的配置，GPTM2 遵循GPTM1 的配置，以此类推。如果定时器A被配置成32位或者64 位定时器，它会在下一个模块中触发TimerA。如果定时器A被配置成16位或者32位定时器，它会在同一模块中触发TimerB，然后TimerB再触发下一个模块的TimerA。这个功能在单次，周期和PWM模式下都是有效的。



图12 级联模式时菊花链触发顺序

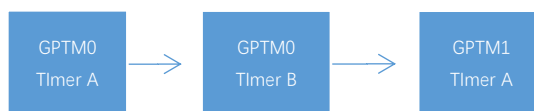


图13 独立运行模式时菊花链触发顺序

5.2 GPTM 同步

GPTM同步寄存器（GPTMSYNC）可以用来同步选中的定时器，让它们同时开始计数。同步定时器不会产生中断。注意：所有的定时器必须使用相同的时钟源，以保证该功能正确工作。

5.3 触发 uDMA 和 ADC 功能

每个定时器可以给ADC发送触发信号。通过设置寄存器GPTMCTL中的TnOTE可以设置定时器产生中断时，是否同时触发ADC功能。

当uDMA功能开启时，定时器产生中断时还能触发uDMA传输，具体操作参考第8章微

型直接存储器访问。

6 例程

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/tm4c123gh6pm.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/sysctl.h"
#include "driverlib/interrupt.h"
#include "driverlib/gpio.h"
#include "driverlib/timer.h"

void Timer0IntHandler(void);

int main(void)
{
    uint32_t ui32Period;

    SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL
_OSC_MAIN);

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_2);

    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);//使能定时器0
    TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC);//设置定时器0为周期定时器模
式

    ui32Period = SysCtlClockGet();
    TimerLoadSet(TIMER0_BASE, TIMER_A, ui32Period - 1);//设置定时器0A的初值

    // IntEnable(INT_TIMER0A);//使能定时器0A中断
    TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);//中断类型为超时中断
    TimerEnable(TIMER0_BASE, TIMER_A);//使能定时器A中断
    TimerIntRegister(TIMER0_BASE, TIMER_A, Timer0IntHandler);
    while(1)
    {
    }
}

void Timer0IntHandler(void)
```

```

{
    // Clear the timer interrupt
    TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);

    // Read the current state of the GPIO pin and
    // write back the opposite state
    if(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_2))
    {
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0);
    }
    else
    {
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 1<<2);
    }
}

```

该例程中，使用的是Timer0的TimerA，且设置它为周期定时模式，每隔1s产生一次超时中断，中断产生时，读取PF2引脚的状态并取反。

因此该例程的功能是：蓝色LED灯每隔1s循环亮灭。该例程的功能跟前面学过的systick的功能几乎是一样的。

7、看门狗定时器

看门狗定时器是为了防止程序进入死循环而使微处理器死机的定时装置。

看门狗定时器的功能是用来在系统产生故障时（如由于软件编写错误而导致系统锁死、或是外部设备失去响应），重新获取系统的控制权。

在启动了看门狗定时器后，看门狗就开始自动计时，正常的程序会定时管理看门狗的定时装置（也称为“喂狗”），如果超过一定时间不去管看门狗，就意味着程序很有可能进入死循环，即死机，则看门狗计数器就会溢出从而引起中断，进一步造成系统复位，从而解决微处理器死机的问题。

TM4C123GH6PM 微控制器的看门狗定时器模块具有如下功能：

- 32-位减法计数器与可编程装载寄存器。
- 具有独立时钟的看门狗模块。
- 可编程的中断产生机制。
- 可通过使能/禁止逻辑控制复位操作。

看门狗定时器可配置为在第一次超时发生时产生一次中断，并在第二次超时发生时产生复位信号。当看门狗定时器被配置后，用户可通过锁定寄存器以防止看门狗定时器配置被意外更改。

7.1 看门狗模块结构图

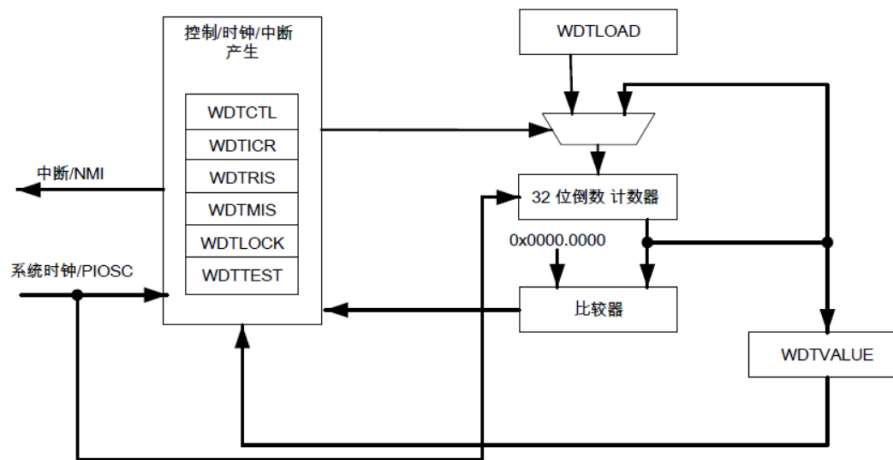


图14 看门狗定时器模块框图

每个WDT（看门狗定时器）模块主要包括两个部分：即控制部分、计数器和比较器部分。WDT结构框图如上图14所示。

控制部分为左侧的控制/时钟/中断信号部分，主要用于计数器的配置、中断信号的控制、时钟信号的控制等。

计数器和比较器部分中，计数器是一个32位的递减计数器，比较器用于计数值与0的比较，确定是否超时。WDT处理过程中使用的主要寄存器包括WDTLOAD和WDTVALUE，前者是计数器的预设值，后者是计数器当前的计数值，微处理器通过读取寄存器WDTVALUE了解计数器当前的计数值。

需要注意的是，TM4C123GH6PM所包含的两个WDT模块有些不同，WDT0模块的时钟源始终是系统时钟，而WDT1模块使用的时钟源始终是PISOC，即内部16MHz时钟，误差为正负1%，因此在使用WDT1模块时需要一个同步装置。除了时钟不同外，两个WDT模块的其余部分都是相同的。

WDT1处在一个独立的时钟域中，因此，必须在恰当的时机才能对其寄存器进行访问。只有在WDTCTL寄存器的WRC位被置1时，才能对WDT1的寄存器进行写入操作；而对读取操作则无限制。注意：由于WDT0由系统时钟驱动，因而无此限制。

7.2 工作流程

启动WDT后，其工作流程如下：

- 1、计数器从初始值开始向下计数。
- 2、当计数器的计数达到0时，表示计数器超时，这是第一次超时。这时计数器会产生一个中断信号，并且重新从初始值开始计数。
- 3、若复位信号被使能，即寄存器WDTCTL中的RSEN位被置1，在中断位没被清除之前，如果计数器又超时，则会产生一个复位信号使得微控制器复位。
- 4、如果中断位在超时前被清除，则计数器重新从初始值开始计数。

需要注意的是如果初始值在计数的过程中被重写了，则WDT的计数器就重新载入新的初始值，开始计数。但是这—个载入过程并不会清除已经存在的中断位。

7.3 WDT 中断控制

WDT计数器超时可以被配置为中断事件，该中断事件可以被配置为可屏蔽的，也可以被配置为不可屏蔽的，通过配置寄存器WDTCTL的INTTYPE 位来选择，通过配置寄存器WDTICR位可清除中断。中断状态可以从WDTRIS及WDTMIS这两个寄存器中读出，其中，前者是原始中断状态，后者是屏蔽后的中断状态。

7.4 初始化及配置

启用看门狗定时器前，必须对看门狗定时器连接的外围设备的时钟门寄存器（RCGCWD）的Rn位进行配置，看门狗定时器的配置使用以下顺序：

1. 使能WDT模块时钟信号配置寄存器RCGC0；
2. 将所需的定时器初始值写入寄存器WDTLOAD 中；
3. 如果使用的是WDT1模块，则需要等待寄存器WDTCTL的WRC位被置位；
4. 如果需要配置看门狗定时器能够触发系统RESET信号，则需配置WDTCTL寄存器的RESEN位为1；
5. 如果使用的是WDT1，则需要等待WDTCTL的WRC位被置位；
6. 使能WDT模块，配置WDTCTL 寄存器的INTEN 位，使能看门狗定时器，并使能中断、锁定控制寄存器。

如果要对看门狗定时器进行锁定，则只需在WDTLOCK 寄存器中写入任意值即可；解锁则需向WDTLOCK 寄存器写入0x1ACC.E551。

8、预习习题

- 1、什么是向上/向下计数？以周期定时器模式为例来说明二者的区别。
- 2、边沿计数模式和边沿计时模式的区别是什么？
- 3、什么是PWM波形？
- 4、练习德研开发板的 DEMO 程序中的 Timer 文件夹下的 counter、period-interrupt、PWM 程序，及 WDT 文件夹下的 watchdog-timer 程序，学习 GPTM 在不同模式下的编程方法。

9、课堂练习

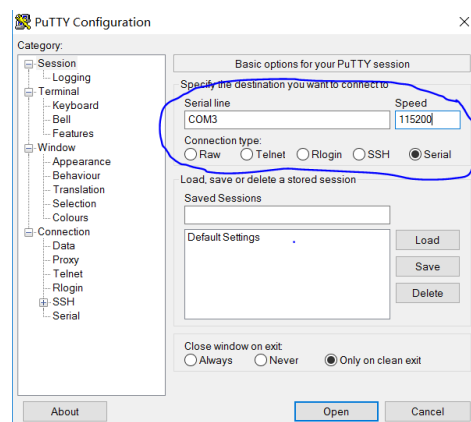
- 使用德研开发板，写一个利用通用定时器中断实现定时查询按键的程序，实现K3键控制D3灯循环亮灭。

10、课后作业

- 1、利用两个定时器编写程序，一个定时器用来产生50KHz正弦波信号，将此方波信号接到另一个定时器的输入端，通过边沿计时模式来测量该信号的周期，并将其显示在液晶上。
- 2、设计综合任务中与GPTM模块和WDT模块相关的任务，即综合任务中与定时器和看门狗有关的部分，列出题目，要求题目应具体。

附：

串口调试助手 putty 的设置方法如下：



选择 Serial，串口号为驱动安装后在设备管理器中出现的虚拟串口号，如 COM3，波特率设置为 115200，然后点击“open”。当程序中使用 UARTprintf 函数时使用。

