stm32cube使用初次感受

1. 初次感受

已经配置了底层的寄存器就是很方便。

1. 实现printf

在stm32cube上实现printf

#ifdef \_\_GNUC\_\_

#define PUTCHAR\_PROTOTYPE int \_\_io\_putchar(int ch)

#else

#define PUTCHAR\_PROTOTYPE int fputc(int ch, FILE \*f)

#endif

PUTCHAR\_PROTOTYPE

{

HAL\_UART\_Transmit(&huart1 , (uint8\_t \*)&ch, 1, 0xFFFF);

return ch;

}

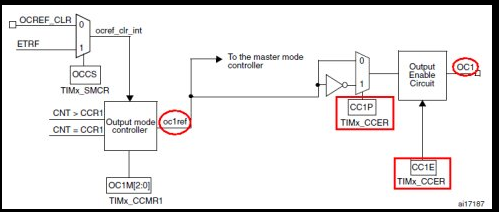
1. Time

Time的OCxref，OCx，CCxP的关系：

ocxREF是输出阶段产生一个中间波形OCxREF（高有效）为参考。输出信号的急性体现在信号连的末端。

TIMx\_CCMR1寄存器OC1M的值，一个输出的类型：

1. 000 FROZEN冻结：输出比较寄存器TIMX\_CCR1中的内容与计数器TIMX\_CNT中的内容之间的比较对输出无影响（用于时基的产生）
2. 001 当匹配时，强行拉高oc1ref电平
3. 010 当匹配时，强行拉低oc1ref电平
4. 011 翻转oc1ref电平 （toggle）切换pwm
5. 100 强制无效电平
6. 101 强制有效电平 (forced)强制
7. 110 PWM模式1,向上计数模式中，只要TIMx\_CNT<TIMX\_CCR1,通道一有效，反之无效。向下计数中。TIMx\_CNT>TIMX\_CCR1,无效电平，反之有效。
8. 110 PWM模式2 。具体和pwm模式相反。



CC1P控制着最后的输出oc1和oc1ref的关系。

4.1ClockDivision

定时设置：ClockDivision ：定义在定时器时钟(CK\_INT)频率与数字滤波器(ETR， TIx)使用的采样频率之间的分频比例  
它是输入捕获中滤波器的采样频率吧

4.1Repetition Counter

TIM\_TimeBaseStructure.TIM\_RepetitionCounter=0;  
是重复计数，就是重复溢出多少次才给你来一个溢出中断,

4.2计数模式center aligned

01：中央对齐模式1 计数器交替地向上和向下计数。输出比较中断标志位，只在计数器向下计数时被设置

10：中央对齐模式2 计数器交替地向上和向下计数。输出比较中断标志位，只在计数器向上计数时被设置

11：中央对齐模式3 计数器交替地向上和向下计数。输出比较中断标志位，只在计数器向下和向上计数时均被设置

* 1. 输入捕获和PWM捕获的特点

输入捕获就一个通道，用来捕获一个脉冲的时间。

PWM捕获是两个通道，一个捕获脉冲时间，一个捕获周期

Stm32cube 中的设置：

Input Filter设置滤波参数：

0000-1111 依次为频率的2，8,16,32。次数2,4,8，。具体还是看一下参考。

HAL\_TIM\_IC\_Start\_IT (&htim1,TIM\_CHANNEL\_1);

* 1. 输出捕获和pwd输出

输出捕获有中断。

在stm32cube 中的配置：类似的

TIM\_OCInitStructure.TIM\_OCPolarity=TIM\_OCPolarity\_High;//设置输出有效极性

TIM\_OCInitStructure.TIM\_OCNPolarity=TIM\_OCNPolarity\_Low;//设置互补输出极性

TIM\_OCInitStructure.TIM\_OCIdleState=TIM\_OCIdleState\_Set;//选择空闲状态下得非工作状态

TIM\_OCInitStructure.TIM\_OCNIdleState=TIM\_OCNIdleState\_Reset;//选择互补空闲状态下得非工作状态

CH Polarity ：

设置的有效值

CH Idle State

选择空闲状态下得非工作状态

Pwm mode ：

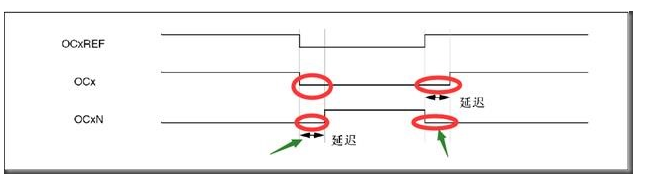
110：PWM模式1－ 在向上计数时，一旦TIMx\_CNT<TIMx\_CCR1时通道1为有效电平，否则为无效电平；在向下计数时，一旦TIMx\_CNT>TIMx\_CCR1时通道1为无效电平(OC1REF=0)，否则为有效电平(OC1REF=1)。  
111：PWM模式2－ 在向上计数时，一旦TIMx\_CNT<TIMx\_CCR1时通道1为无效电平，否则为有效电平；在向下计数时，一旦TIMx\_CNT>TIMx\_CCR1时通道1为有效电平，否则为无效电平。

* 1. 主从模式

主模式：定时器使能只受驱动时钟控制或者输出控制信号(TRGO)。

从模式：

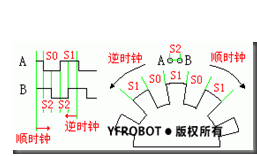
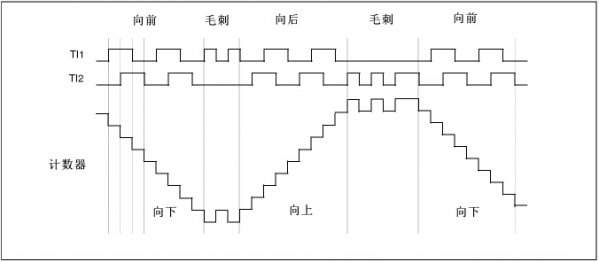
1. 复位模式---在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果IMx\_CR1寄存器的URS 位为低，还产生一个更新事件UEV。
2. 门模式-----计数器的使能依赖于选中的输入端的电平
3. 触发模式---计数器的使能依赖于选中的输入端上的事件
4. 外部时钟模式+触发模式----外部时钟模式2可以与另一种从模式(外部时钟模式1和编码器模式除外)一起使用。这时，ETR信号被用作外部时钟的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。不建议使用TIMx\_SMCR寄存器的TS位选择ETR作为TRGI。
   1. 互补输出和死亡输入



主要是延时一下了

* 1. 使用刹车功能:PWM急停

刹车信号(BREAK[MOE])触发时，输出OCx与OCxN在不同的CCxE,CCxP,OISx,CCxNE, CCxNP, OISxN位控制下的输出组合

* 1. 编码器接口模式
     1. 原理：如果两个信号相位相差90度，则这两个信号称为正交。由于两个信号相差90度。如图：A,B两点对应两个光敏接受管，A,B两点间距为 S2 ,码盘的光栅间距分别为S0和S1。S0+S1的距离是S2的四倍。这样保证了A，B波形相位相差90度
     2. 

当T1,T2脉冲是连续产生的时候计数器加一或减一一次，而当某个接口产生了毛刺或抖动，则计数器计数不变，也就是说该接口能够容许抖动。



当某次我们使得编码器正转产生中断后，立即反转则又该怎么办呢。需要加上方向，参考文档

* + 1. 参考 <http://bbs.eeworld.com.cn/forum.php?mod=viewthread&tid=456178>
  1. 霍尔传感器接口

参考：<http://blog.sina.com.cn/s/blog_7d9d562b0102vncs.html>

霍尔输入连接到TIMx定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器TIMx的PWM配置

* 1. HAL中的开定时器函数

普通计数器：HAL\_TIM\_Base\_Start\_IT

开PWM的值HAL\_TIM\_PWM\_Start(&htim1,TIM\_CHANNEL\_4);

开输出比较HAL\_TIM\_OC\_Start(&htim1,TIM\_CHANNEL\_1);

Node1：输出比较，周期的改变在HAL中要自己改变。

void TIM\_SetTIM3Compare4(u32 compare)

{

TIM3->CCR4=compare;

}

参考 http://blog.sina.com.cn/s/blog\_a6bd181f0102wd1h.html

* 1. 参考资料

<http://www.cnblogs.com/OneFri/p/5924918.html>

<http://blog.sina.cn/dpool/blog/s/blog_3ba262a10101esd1.html?type=-1>

1. 串口问题点

对于应用层的代码，已有的代码架构有点问题。比如串口。

HAL\_UART\_Receive\_IT(&huart1,re\_buff,3);用来打开串口中断，只要当收到3个才会触发回调。好在回调函数可以重新定义（\_\_weak）.

6.1 综述

FMC能够与同步或异步存储器，SDRAM存储器和16位PC存储器卡连接。STM32F429的FMC接口支持包括SRAM，SDRAM,NAND FLASH,NOR FLASH和PSRAM等存储器。这些设备共用地址数据总线等信号，他们具有不同CS以区分不同的设备。

简单介绍一些相关FLASH和RAM。

* NOR的特点是芯片内执行（XIP, eXecute In Place），这样应用程序可以直接在[flash](https://baike.baidu.com/item/flash) 闪存内运行
* PSRAM中文名称是假静态随机存储器。它体积更为轻巧，售价更具竞争力。PSRAM就是伪SRAM，内部的内存颗粒跟SDRAM的颗粒相似，但外部的接口跟SRAM相似，不需要SDRAM那样复杂的控制器和刷新机制，PSRAM的接口跟SRAM的接口是一样的。



这些设备共用地址数据总线等信号，他们具有不同CS以区分不同的设备。

STM32F429的FMC将外部存储器划分为6个固定大小为256M字节的存储区域。

不同区域支持不同的存储器。

例如区域一由28条寻址：



6.2

6个块支持的设备类型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 内部控制器 | 存储块 | 管理的地址管理 | 支持的设备类型 | 配置存储器 |
| NOR FLASH 控制器 | BANK1 | SRAM/ROM  NOR FLASH  PSRAM | SRAM/ROM  NOR FLASH  PSRAM | FMC\_BCR1/2/3/4  FMC\_BTR1/2/3/4  FMC\_BWTR1/2/3/4 |
| NAND FLASH/PC CARD 控制器 | BANK2  BANK3  BANK4 | NAND FLASH  PC CARD | NAND FLASH  PC CARD | FMC\_PCR2/3/4  FMC\_SR2/3/4  FMC\_PMEM2/3/4  FMC\_PATT2/3/4  FMC\_PI04  FMC\_ECCR2/3 |
| SDRAM 控制器 | BANK5  BANK6 | SDRAM | SDRAM | FMC\_SDCR1/2  FMC\_SDTR1/2  FMC\_SDCMR  FMC\_SDCMR  FMC\_SDSR |

* 1. SDRAM
     1. 综述

SDRAM 是 Synchronous Dynamic Random Access Memory 。即同步动态随机存储器，相较于SRAM(静态存储器)，有容量大和价格便宜的特点。(本次使用的是W9825G6KH)

* + 1. SDRAM 信号线

|  |  |
| --- | --- |
| 信号线 | 说明 |
| CLK | 时钟信号，在该时钟的上升沿采集输入信号 |
| CKE | 时钟使能，禁止时钟时，SDRAM会进入自动刷新模式 |
| CS# | 片选信号，低电平有效 |
| RAS# | 行地址选通信号，低电平时，表示行地址 |
| CAS# | 列地址选通信号，低电平时，表示列地址 |
| WE# | 写使能信号，低电平有效 |
| A0-A12 | 地址线（行/列） |
| BS0，BS1 | BANK地址线(这次有四个空间) |
| DQ0-15 | 数据线 |
| LDQM,UDQM | 数据掩码，表示DQ的有效部分 |

在SDRAM内部寻址的方法：先确定BANK号和行地址，然后指定列地址，就可以找到目标地址。

* + 1. 寻址过程：

RAS信号为低电平，选通行地址，地址线A0-A12所表示的地址会被传输并锁存到行地址译码器。同时BANK地址也会被锁存，选中对应的BANK。

CAS信号为低电平，选通列地址，地址线A0-A12所表示的地址，会被传入并所存在列地址。一次传输。

* + 1. 数据传输

在完成寻址以后，数据线DQ0-DQ15 上面的数据会通过数据控制逻辑写入（读出）存储整列。

在实际使用的时候，我们会以8,16,24,32位来读写数据。并不是每条数据线都会使用到。这个使用就需要用到数据掩码（DQM）线来控制。每一个数据掩码线，对应8个位的数据，低电平对数据有效，高电平对数据无效。

6.3．4 控制命令

用驱动线来表示各种命令。下面简单介绍

6.3.4.1 NO-Operation：

空操作命令为接下来的命令发送做准备。

6.3.4.2 Action：

激活命令，该命令必须在读写操作之前被发送，用于设置所需要的Bank和行地址。

6.3.4.3 Read/Write

在发送完激活命令后，再发送列地址就可以完成对SDRAM的寻址，并进行读写操作了，读/写命令和列地址的发送，就是一次传输完成的。

6.3.4.4 Precharge

预充电指令，用于关闭Bank 中所打开的行地址。由于SDRAM的寻址具有独占性。Bank关闭现有行，准备打开新行的操作就叫做预充电（Precharge）。

可以设置自动预充电，来提高读写速度。

6.3.4.5 Refresh

刷新命令对所有行进行刷新操作。一个有两种刷新命令，自动刷新和自我刷新。在发送Refresh命令时，如果CKE有效，则使用自动刷新模式，否则为自我刷新模式。

* 自我刷新：一般为64ms。具体是9个周期。
* 自动刷新：低功耗模式下的。把CKE为低电平。只有当CKE为高时，芯片才接受命令。

6.3.4.6 Mode Register Set

对SDRAM进行初始化的时候，都需要用到。控制单元的相关参数由模式存储器提供。

特例 W9825G6KH的模式：

Burst length：即突发长度（简称BL）。通过A0-A2设置，是指在同一行中相邻的存储单元连续进行数据传输。连续传输所涉及到存储单元（列）的数量就是突发长度。只能在初始化时候确认，一个为1,2,4,8，全页。如果传输的内容小于设定的长度，就需要发送Burst Stop。

Addressing Mode 即突发访问的地址模式。通过A3设置，可以设置为Sequential(顺序)或Interleave（交错），一般选顺序模式。

CAS Latency ，在读命令发送完成后，需要等待几个时钟周期，DQ上的数据才有效。就叫做CL延迟。一个为2/3周期。

写模式：就是写。

6.3.5类似的操作过程

6.3.5.1 初始化：

* 上电发送NOP命令。最少200us。
* 发送预充电命令
* 发送自动刷新命令
* 设置模式寄存器
* 完成

6.3.5.2 写操作

* 发送激活命令
* 发送写命令，拉高A10线，使能自动预充电。
* 使能自动预充电，需要一定时间。
* 执行预充电
* 完成一次数据写入

6.3.5.3 读操作

* 发送激活命令
* 发送写命令
* 使能自动预充电
* 执行预充电
* 完成一次数据写入

6.4 FMC-SDRAM接口使用

一些参数如tRCD tRAS tRP等时间参数，见SDRAM数据手册。在配置FMC会用到。

FMC-SDRAM有两个存储区域。

6.4.1 简单的寄存器

6.4.1.1控制寄存器（FMC\_SDCRx）x=1或2.

主要配置为：

列地址位数（NC）