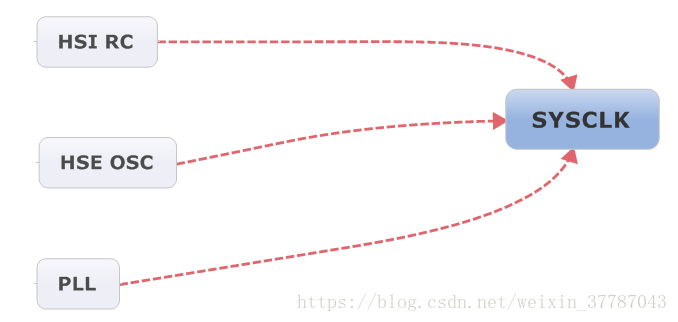
<https://blog.csdn.net/weixin_37787043/article/details/81283262>

1、AHB系统总线分为APB1（36MHz）和APB2（72MHz），其中2>1，意思是APB2接高速设备；

2、Stm32f10x.h相当于reg52.h（里面有基本的位操作定义），另一个为stm32f10x\_conf.h专门控制外围器件的配置，也就是开关头文件的作用；

3、 HSE Osc（High Speed External Oscillator）高速外部晶振，一般为8MHz，HSI RC（High Speed InternalRC）高速内部RC，8MHz；

4、 LSE Osc（Low Speed External Oscillator）低速外部晶振，一般为32.768KHz，LSI RC（Low Speed InternalRC）低速内部晶振，大概为40KHz左右，提供看门狗时钟和自动唤醒单元时钟源；

5、 SYSCLK时钟源有三个来源：HSI RC、HSE OSC、PLL；  


6、 MCO[2：0]可以提供4源不同的时钟同步信号；

7、 GPIO口貌似有两个反向串联的二极管用作钳位二极管；

8、 总线矩阵采用轮换算法对系统总线和DMA进行仲裁

9、 ICode总线，DCode总线、系统总线、DMA总线、总线矩阵、AHB/APB桥

10、在使用一个外设之前，必须设置寄存器RCC\_AHBENR来打开该外设的时钟

11、数据字节以小端存储形式保存在存储器中

12、 内存映射区分为8个大块，每个块为512MB

13、 FLASH的一页为1K(小容量和中容量)，大容量是2K。

14、 系统存储区（SystemMemory）为ST公司出厂配置锁死，用户无法编辑，用于对FLASH区域进行重新编程。所以我们烧写程序务必选择BOOT1 = 0，这样通过内嵌的自举程序对FLASH进行烧写，比如中断向量表和代码

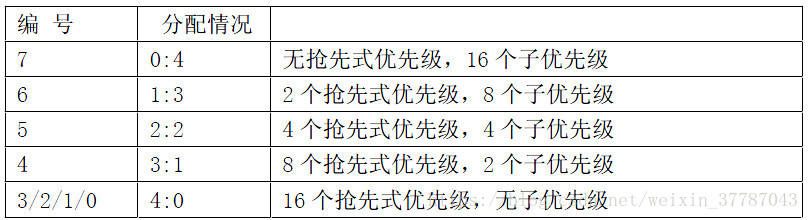
15、 STM32核心电压为1.8V

16、 STM32复位有三种：系统复位、上电复位、备份区域复位。其中系统复位除了RCC\_CSR中的复位标志和BKP中的数值不复位之外，其他的所有寄存器全部复位。触发方式例如外部复位、看门狗复位、软件复位等；电源复位由于外部电源的上电/掉电复位或者待机模式返回。复位除了BKP中的寄存器值不动，其他全部复位；备份区域复位的触发源为软件复位或者VDD和VBAT全部掉电时。

17、 单片机复位后所有I/O口均为浮空输入状态

18、 68个可屏蔽中断通道，16个可编程优先级，16个内核中断，一共68+16=84个中断。103系列只有60个中断，107系列才有68个中断

19、 系统启动从0x00000004开始，0x000 0000保留

20、 （NestedVectored Interrupt Controller）NVIC嵌套向量中断控制器，分为两种：抢先式优先级（可嵌套）和中断优先级（副优先级，不能嵌套）。两种优先级由4位二进制位决定。分配下来有十六种情况：  


21、0号抢先优先级的中断，可以打断任何中断抢先优先级为非0号的中断；1号抢先优先级的中断，可以打断任何中断抢先优先级为2、3、4号的中断；……；构成中断嵌套。如果两个中断的抢先优先级相同，谁先出现，就先响应谁，不构成嵌套。如果一起出现（或挂在那里等待），就看它们2个谁的子优先级高了，如果子优先级也相同，就看它们的中断向量位置了。原来中断向量的位置是最后的决定因素！！！！

22、 上电初始化后AIRC初始化为0，为16个抢先式优先级，但是由于所有的外部通道中断优先级控制字PRI\_n为0，所以抢先式优先级相同，此时就不能嵌套了

23、 NVI中有ISER[2]（Interrupt Set-Enable Registers），ICER[2]（Interrupt Clear-Enable Registers），ISPR[2]（Interrupt Set-Pending Registers），ICPR[2]（Interrupt Clear-Pending Registers），IABR[2]（Active Bit Registers），IPR[15]（InterruptPriority Registers）定义。其中ISER和ICER分别为中断使能和中断失能寄存器，都是写1来使能/失能中断的。为什么写1？为什么不采用一个寄存器而用两个寄存器来表示中断使能/失能状态？由于硬件，写0比较复杂，并且可能造成其他位的状态改变，所以用1来表示打开或者关断是比较合理的

24、 **中断标志位需要手动清除**；

25、 配置外围器件的一般步骤：1、打开端口时钟。2、定义初始化结构体并初始化。3、调用

26、串口的奇偶校验：如果是奇偶校验，那么USART\_InitStructure.USART\_WordLength= USART\_WordLength\_9b;这个数据的长度必须设定为9位！

27、ADC的规则组可以自定义转换通道顺序和转换的通道个数。在实际应用中，有时候希望有一些特别的通道具有很高的优先权，需要在规则组进行转换的时候强制打断，进行另一个通道的转换，这样一组通道，叫做注入组。

28、定时器的输出比较模式：Timing（冻结，什么都不做，普通定时）,Active（OCxREF输出高电平有效）,Inactive（OCxREF输出低电平）,Toggle（比较成功后翻转电平）。

29、STM32的定时器从0开始计数，满足一些条件，给出标志位（比如匹配成功、时间更新、溢出等）然后从0开始计数。这一点和51不同。

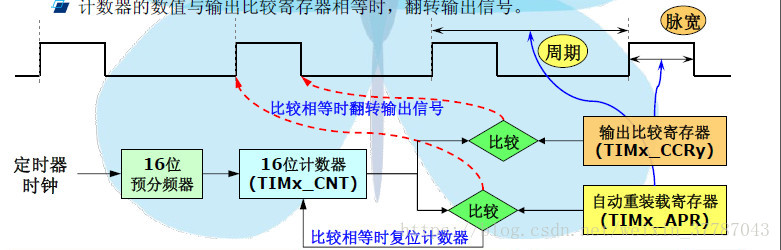
30、OCx=OCxREF+极性

31、自动装载寄存器和影子寄存器：前者相当于51当中的溢出设定数值。而影子寄存器顾名思义是影子，就是寄存器的另一分copy。实际起作用的是影子寄存器，而程序员操纵的则是自动装载寄存器。如果APPE位使能，表明自动装载寄存器的值在下一次更新事件发生后才写入新值。否则，写入自动装载寄存器的值会被立即更新到影子寄存器。

32、RCC\_PCLK1Config(RCC\_HCLK\_Div4);PCLK1的4分频给定时器基准时钟

33、定时器配置：RCC、NVIC、GPIO（OC输出或者PWM）、TIMx

34、通用定时器可以输出4路不同的PWM，高级定时器可以输出4路不同的PWM外，还可以输出3路互补的PWM信号（驱动三相电机），一共有7路。这样算出来STM32可以产生30路PWM=7\*2+4\*4

35、

36、高级定时器时钟源挂在了APB2上，而通用定时器挂在APB1上。AHB（72mhz）→APB1分频器（默认2）→APB1时钟信号（36mhz）→倍频器（\*2倍）→通用定时器时钟信号（72mhz）。如果APB1没有分频，那么通用定时器的时钟信号频率就直接等于APB1的时钟频率，没有上述的倍频器\*2过程。TIM\_SetAutoreload()用来改变PWM的频率,TIM\_SetCompare1()用来改变占空比

37、有刷电机一般启动力矩大一些，无刷电机启动力矩小，运行起来力矩大。有刷电机采用电刷机械电流换向，而无刷电机则通过霍尔传感器测出转子的电流来判断电机的运动位置和方向，返回给控制回路。

38、死区是必须要有的，因为这涉及到电路的短路问题。晶闸管在换向的时候需要死区时间来彻底关断线路

39、刹车功能用来在控制回路出现问题时，硬件自动给予外部电机进行紧急刹车制动，反应在PWM上持续给出一个固定的占空比？（三相驱动也是？）

40、PWM输出最好采用PWM模式，其他的比较输出模式相位会慢慢改变，不精准；

41、对FLASH的读写需要先解锁后加锁。FLASH写0容易，写1难。

42、下载程序有两种方式，一种为ICP（在线编程），适用于JTAG或SWD协议下的烧写程序。另一种成为IAP（在应用编程），适用于很多接口（USB，串口，CAN）下载到存储器中，IAP允许在程序运行时重新烧写FLASH

43、FLASH分为主存储器（这里放置用户的程序代码）和信息块（启动代码），除此之外，还有一部分叫做系统存储器，这一块用户不可操作，为ST公司出产后固化，为系统的上电自举程序

44、FLASH在写的时候，一定不能读，如果有读操作，那么将会锁住总线

45、对FLASH操作时，必须打开HIS；

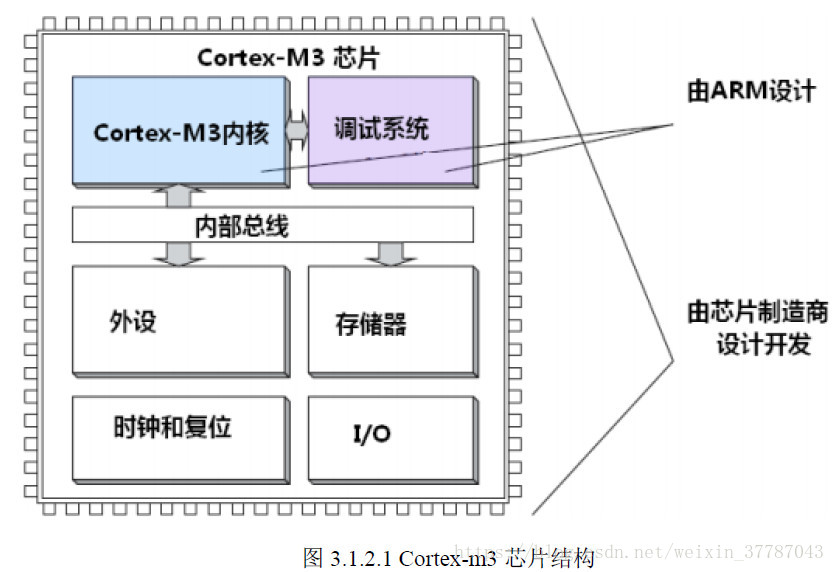
46、STM32有两种看门狗（IWDG独立看门狗《独立时钟》，WWDG窗口看门狗《由APB1分频而来》）

47、SPI的的最高频率为36MHz(fpclk/2)

48、 TIM1和TIM8高级定时器在输出PWM时，需要配置一下主输出功能（CtrlPWMOutputs）才能输出PWM。其他的通用定时器不需要这样配置。但是TIM6和TIM7没有PWM输出功能。

49、\*\*Code为程序代码部分  
RO-data 表示程序定义的常量(如：const temp等);  
RW-data 表示已初始化的全局变量  
ZI-data 表示未初始化的全局变量,以及初始化为0的变量\*\*  
Code, RO-data,RW-data…………..flash  
RW-data, ZIdata……………….RAM  
初始化时RW-data从flash拷贝到RAM

50、STM32F103ZET6有144个引脚（Z为144），其中，可用IO口为112个（7X16=112，ABCDEFG口）；

51、ARM公司只生产内核标准，不生产芯片。ST、TI这样的公司从ARM公司那里购买内核，然后外加自己的总线结构、外设、存储器、时钟和复位、I/O后就组成了自己的芯片。  


52、CMSIS标准用于在向上的用户层和下面的硬件层交换信息。这个架构当然可以自己定义，但是这样的话就会没有标准。所以强制使用CMSISI标准来设计芯片。通俗点的讲就是系统初始化的函数名称CMSIS定义为SystemInit()，GPIO\_ResetBits()等

53、端口复用和端口重映射是两个概念：前者在使能其对应的端口和对应的功能时钟即可。后者需要打开AFIO时钟，然后进行端口的重映射GPIO\_PinRemapConfig()

54、下载程序只能使用串口1，在硬件设计时一定要注意！

55、J-TAG调试频率一般设定为2MHz，而SWD调试频率可以设定为10MHz

56、SysTick的中断实现可以有两种方式：**循环等待和中断法**。推荐用循环等待，中断法可能会出问题而且占用资源。

57、部分I/O引脚是5V兼容的。单个I/O的最大驱动电流和灌入电流均为25mA。整个芯片的电流为150mA

58、KEIL支持位段操作，可以利用C语言中的位段知识定义位段结构体，然后对单独的寄存器进行单独的位操作。

59、关于内部上下拉电阻的设置：如果外部的按键另一头接地，那么需要设置成上拉电阻。（理由是当没有按下按键时，由于上拉，输入为高电平；按下时，由于外部接地，输入为低电平。）同理，如果外部的按键另一头接高电平，那么需要设置成下拉电阻。

60、串口中断TXE和TC的区别：其实很明显，**一个是发送寄存器空标志，一个是发送完成标志**。因为串口在发送的时候首先需要把发送寄存器中的数据移位到移位寄存器（影子寄存器）后再串行发送出去。所以当发送寄存器DR空时说明现在可能正在往外面发送数据，数据可能还没有发送完。但是发送完成不一样，他是在移位寄存器将本次数据全部移位完成后设置的标志位（也就是发送完了停止位）。这么看来：TXE允许程序有更充裕的时间填写TDR寄存器，保证发送的数据流不间断。TC可以让程序知道发送结束的确切时间，有利于程序控制外部数据流的时序。

61、窗口看门狗顾名思义有一个窗口，这个窗口的横坐标为时间，意思是在指定的时间范围内刷新寄存器，否则单片机复位。窗口的上限由人来设定W[6:0]，下线定死为0x40Twwdg=(4096×2^WDGTB×(T[5:0]+1)) /Fpclk1；Twwdg为超时时间ms，Fpclk1为APB1时钟KHz，

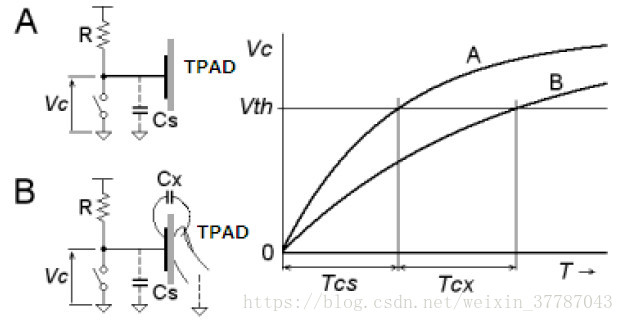
62、TIMx通用定时器有4个独立通道，分别可以用来作为：输入捕获、比较输出、PWM生成、单脉冲模式输出。

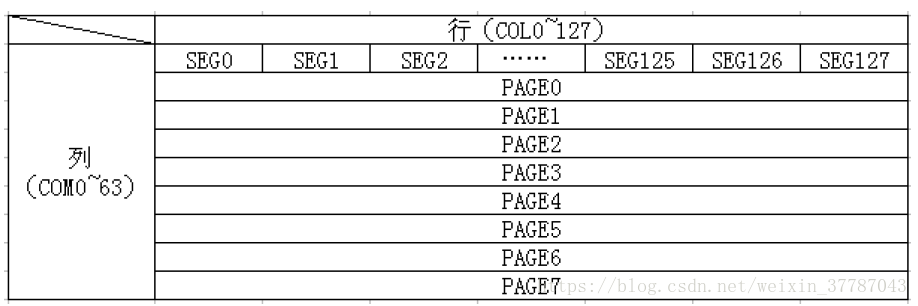
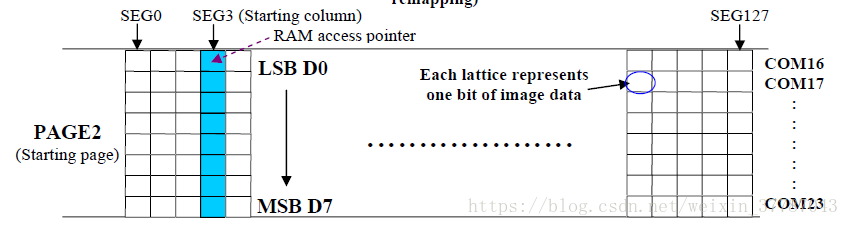
63、定时器的时钟来源有4个：内部时钟（CK\_INT），外部时钟模式1（TIx），外部时钟模式2（ETR），内部触发模式（ITRx，这个用来定时器的同步）

64、定时器中断溢出更新时间：Tout=((arr+1)\*(psc+1))/Tclk，ARR为自动装载寄存器（1~65535）、PSC为分频系数，TCLK为输入时钟频率（Mhz）

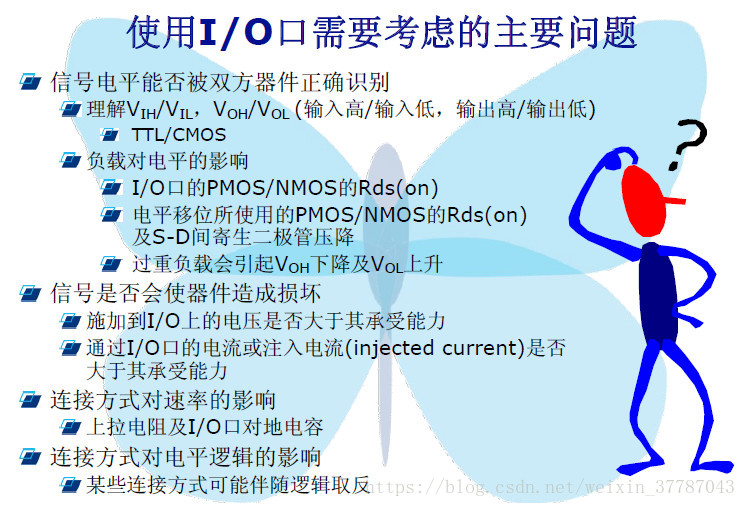
65、PWM1和PWM2模式的区别仅在于相位的180度。前者高电平时，后者低电平。OCxREF极性就可以实现这个功能。

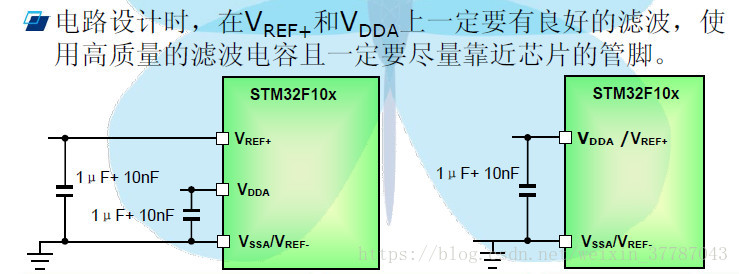
66、定时器输入捕捉有一个滤波器，顾名思义滤波器起到的就是滤波的作用，在捕捉外部信号时，信号可能不稳定，此时需要滤波：当检测到有外部输入时，需要再连续采样N次如果确定为高电平/低电平，则触发响应中断（如果开启了的话）。

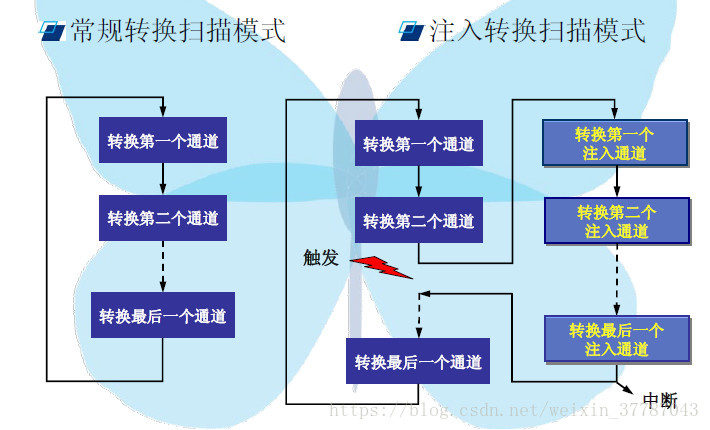
67、电容触摸屏原理：通过充放电的曲线不同来检测是否被按下。 实际的实验过程中，TPAD可以用一块覆铜区域来替代，通过电容的充放电常数来确定是否按下。  


68、OLED，即有机发光二极管（OrganicLight-Emitting Diode），又称为有机电激光显示（Organic Electroluminesence Display，OELD）。下图为OLED的GRAM与屏幕的对应表；  
  
PAGE2单独列出来：  


69、USART可以操纵SPI设备。不过最大频率只有4.5MHz；

70、使用I/O口时应该注意的问题；  


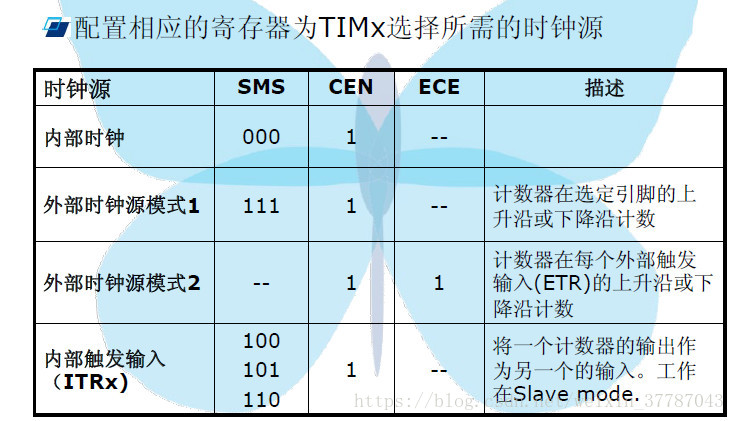
71、ADC的Vref+和Vdda与VSS，Vref-一定要加高质量的滤波电容，且尽量靠近芯片的管脚；  


72、ADC分为规则组和注入组，前者有16个通道，后者有4个通道。并且16个通道公用一个数值寄存器，注入组的4个通道分别有一个数值寄存器。  


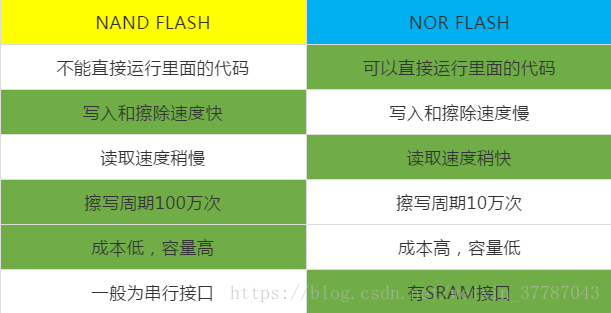
73、采样频率越高，输入阻抗要求越小；

74、Stm32进入中断的最短周期为6个周期；

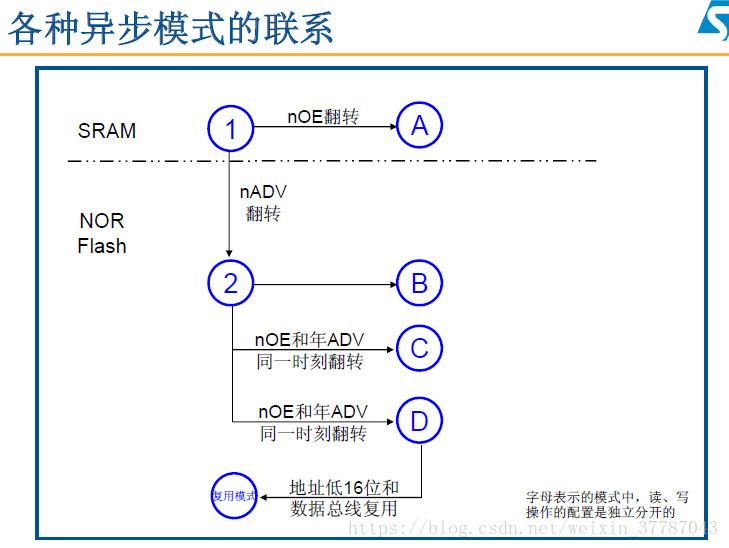
75、降低功耗的基本原则：  
降低系统时钟；  
关闭APB和AHB总线上未使用外设的时钟；

76、配置相应的寄存器为TIMx选择所需的时钟源；  


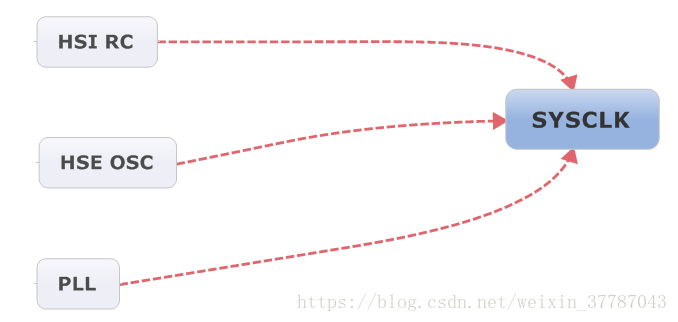
77、FSMC，即灵活的静态存储控制器。能够与同步或异步存储器和16位PC存储器卡接口，STM32的FSMC接口支持包括SRAM、NANDFLASH、NORFLASH和PSRAM等存储器；

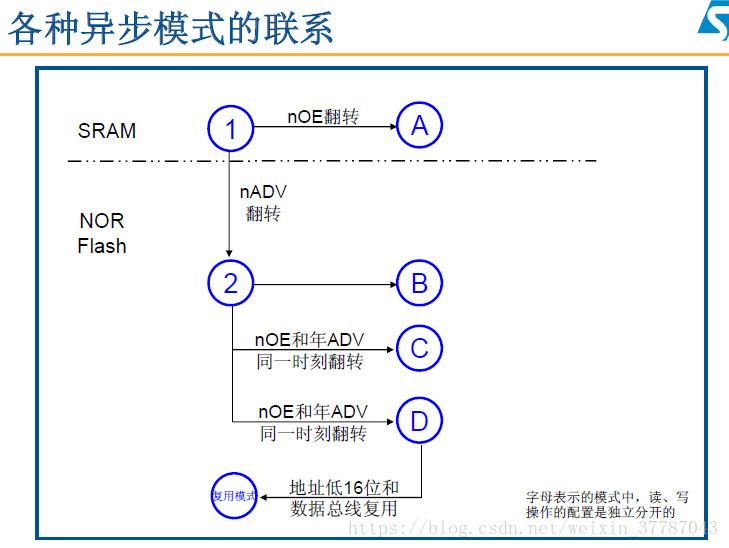
78、平时所说的U盘里的FLASH存储器有两种类型：NANDflash和NOR flash。  


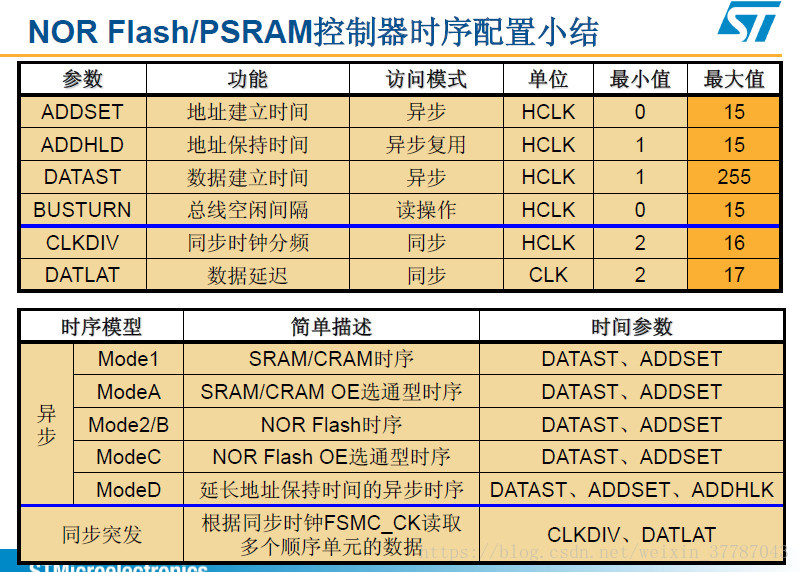
79、TFT在操作时，可以当作外部SRAM来操作，这样的话，如果单片机有FSMC接口，就可以使用NORFLASH的SRAM接口去控制，速度非常快。

80、Stm32的的FSMC有4个256MB的存储块，一共寻址1GB的外部存储器空间；  


81、在STM32内部，FSMC的一端通过内部高速总线AHB连接到内核Cortex－M3，另一端则是面向扩展存储器的外部总线。内核对外部存储器的访问信号发送到AHB总线后，经过FSMC转换为符合外部存储器通信规约的信号，送到外部存储器的相应引脚，实现内核与外部存储器之间的数据交互。

82、FSMC中的DATASET和ADDSET的设置需要参看外部存储器的时序图来确定。一般而言，DATASET指的是数据建立时间，也就是读/写信号开始到读/写信号停止（上升沿存储数据）的持续时间。（一般来说写比读快！）。而ADDSET指的是地址建立时间，指的是片选之后到读/写操作之前的时间，这是针对SRAM来说的，如果操纵的是TFT，不存在地址线，所以此时的ADDSET就是读/写信号结束到RS电平的转换时间。  


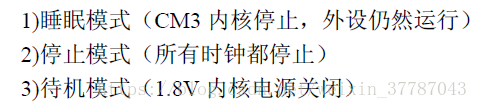
83、各种异步模式的联系；  


84、NOR Flash/PSRAM控制器时序小结；  


85、FSMC的三个配置寄存器：FSMC\_BCRx(片选控制配置)、FSMC\_BTRx(片选时序)、FSMC\_BWTRx(片选写时序)。

86、 RTC时钟配置必须要用到BKP寄存器，BKP寄存器在单片机复位、电源复位、待机唤醒模式下是不会更改值的，他的供电由VDD供电，VDD被切断后自动切换至外部的VBAT供电。

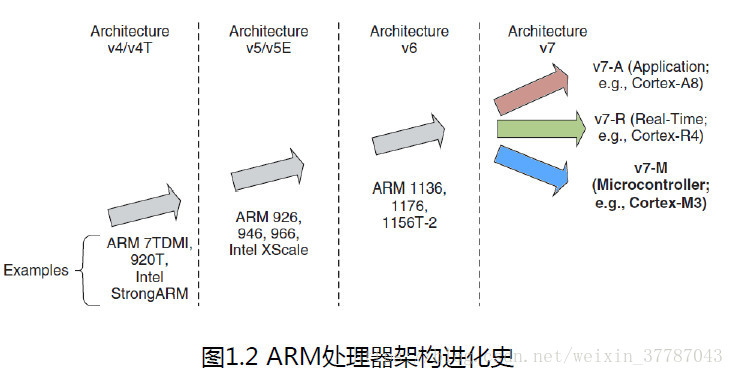
87、 要修改BKP寄存器的值，必须取消其写保护的标志。BKP寄存器在上电时自动写保护；

88、STM32有3种省点模式；  
  
三种省电模式中，耗电量从上到下依次降低，待机模式的电流仅为2uA。

89、 从待机模式中唤醒单片机等效于让单片机复位，但是电源寄存器的值会有一个标志位指示单片机是被唤醒的，不是被复位的。

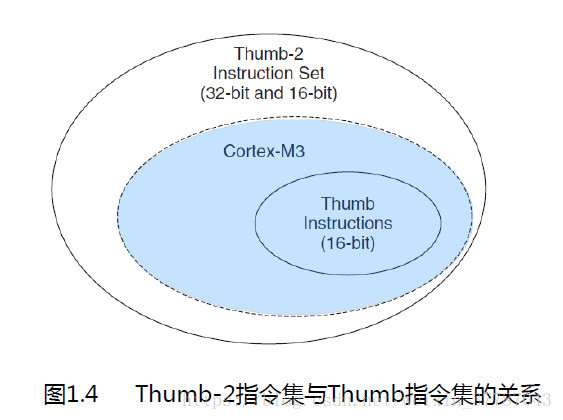
90、 ADC的时钟不要超过14MHz，否则转换精度会下降。最大转换速率为1MHz，即转换周期为1us(14MHz,采样周期为1.5个ADC时钟)

91、 Tcovn=采样时间+12.5个周期。采样时间尽量选长一点，这样精度高一些，但是转换速率下降，这也是有利必有弊。

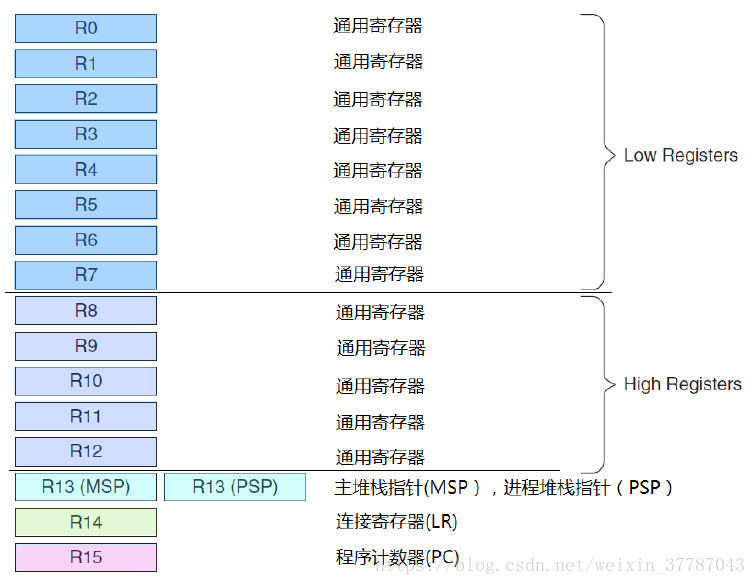
92、ARM处理器架构进化史；  


93、 拿ARM7TDMI来说，T代表Thumb指令集，D是说支持JTAG调试(Debugging)，M意指快速乘法器，I则对应一个嵌入式ICE模块。

94、 MMU作为嵌入式处理器与应用处理器的分水岭标志à具有内存管理单元的嵌入式处理器可以定位为应用处理器。这么说M系列和A系列的处理器的区别在于A系列的处理器具有MMU单元可以进行内存模块的管理。

95、ARM处理器有两种状态：ARM状态和Thumb状态。  


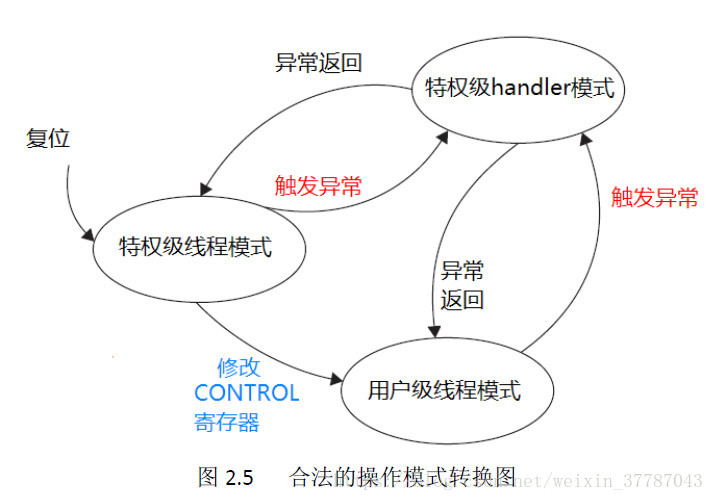
96、 这张图说明了：Thumb2指令集做了一件很伟大的事情：将16位和32位的指令集融为一体，兼容性非常强！（这么说CM3不支持某些32位ARM指令集？？）

97、寄存器  


98、MSP是系统复位后使用的堆栈指针，PSP由用户的代码使用。两个堆栈指针为4字节对齐！！

99、在ARM编程领域中，凡是打断程序运行的事件，统称为异常(exception)。

100、 因为存在LR（链接寄存器），所以可支持1级的子程序调用而不用压栈到内存，大大提高了运行速度。—à这就是说，我们在编程的时候，一级调用是不会耗费太多时间的，除非是二级调用！

101、 处理器有两种操作模式：handler模式和线程模式。  
  
处理器也有两种特权分级：特权级和用户级。 这张图说明了一切：复位进入特权级线程模式，如果有异常，进入特权级的handler模式处理异常或中断例程，然后返回至特权级线程模式。通过修改CONTROL寄存器可以进入用户级线程模式。

102、 两个高级定时器TIM1和TIM8是挂接在APB1总线上

103、 STM32的外部中断是以组来区分的，也就是说PA0,PB0,PC0单片机是无法区分其中哪个触发的中断à均为EXIT0线中断服务例程。所以，外部中断支持16路的中断分辨率。从另一个方面来讲，我们可以设置GPIO\_EXTILineConfig(GPIO\_PortSourceGPIOx, GPIO\_PinSourcex);来开通中断线实现组内的不同中断。

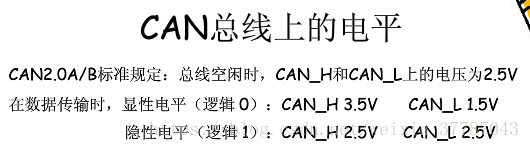
104、 DAC有两个寄存器，一个是DHR（Data HoldingRegister）数据保持寄存器，一个DOR（Data Output Register）数据输出寄存器。真正起作用的是DOR寄存器，该寄存器把值给数模转换发生单元输出以VREF+为参考电压的电压值。如果是硬件触发转换，系统将在1个ABP时钟周期后把值给DOR，如果是软件触发转换，时间为3个APB时钟周期。然后，均等待Tsetting时间（Typical为3us,Max为4us）后真正输出电压值。

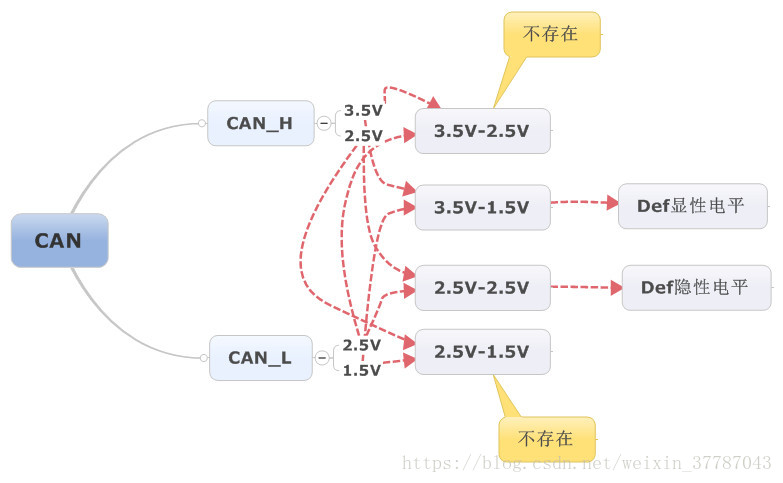
105、 DAC分8位模式和12位模式，其中后者可以选择左右对齐

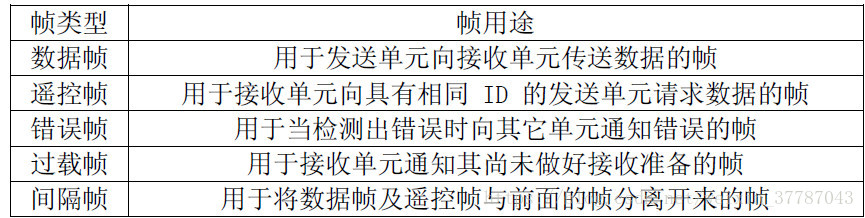
106、 DMA仲裁器分为软件和硬件两种。软件部分分为4个等级，分别是很高优先级、高优先级、中等、低。硬件部分由通道的大小来决定优先级，越低优先级越高。

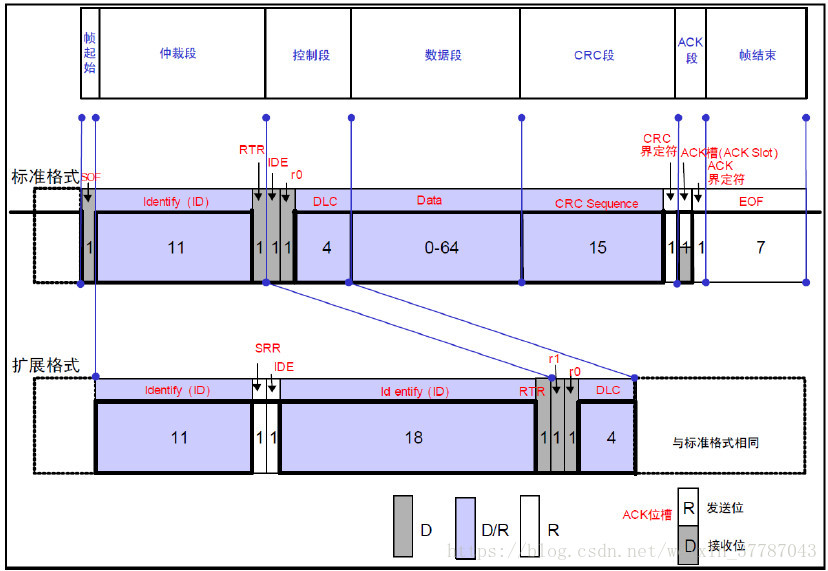
107、 DMA有一个实时的传输数据量寄存器叫做DMA\_CNDTR，最大值为65535，存放的是当前传输所要传输的数据量。当数据量变为0时，表明传输完成。

108、 CAN总线（ControllerArea Network）。CAN控制器根据两根线上的电位差来判断总线电平，总线电平又分为显性电平和隐性电平，二者必居其一。  
109、 CAN总线具有6个特点：1：多主控制（挂接在总线上的所有设备均可以成为主设备，并且设备ID是用来决定设备的优先级，没有设备地址概念），2：系统若软性（没有设备地址概念），3、通讯速度较快，通讯距离较远（1Mbps下40M，5kbps下10KM），4、具有错误检测、错误通知（通知其他设备）和错误恢复功能（强制结束发送，重复发送接收错误的信息。），5、故障封闭，当总线上的设备发生连续故障错误时，CAN控制器会把改控制器踢出总线。6、连接节点多。理论上可以无限制加载，但是受到时间延迟和电气负载的限制，实际数目是有限制的。降低传输速度可以适当增加可挂接负载个数。

110、 CAN协议有两个标准，ISO11898(针对125kbps~1Mbps的高速速率)和ISO11519-2(125kbps以下的低速速率)  


111、CAN总线电平特性；  


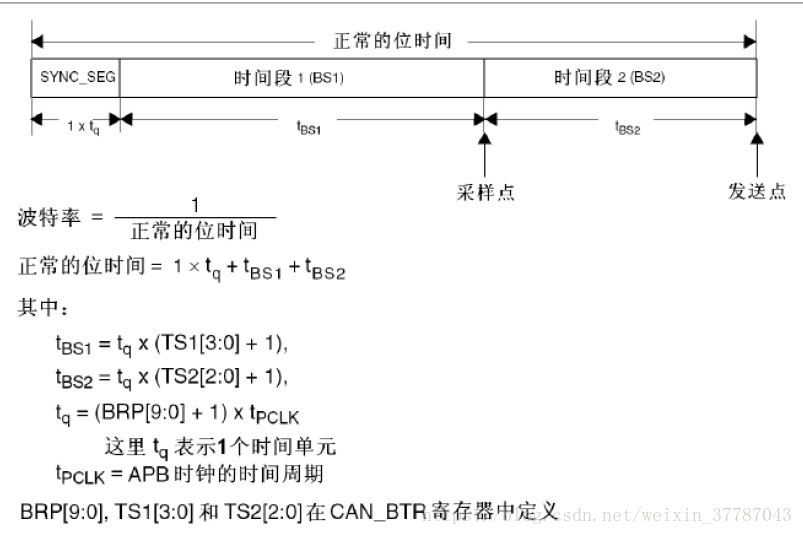
112、 CAN协议的有5种类型的帧：数据帧、遥控帧、错误帧、过载帧、帧间隔。其中前两种帧有标准格式（11位ID）和扩展格式（29位ID）。  


113、CAN 数据帧构成：  
（1） 帧起始。 表示数据开的段帧起始。  
（2） 仲裁段。 表示该帧优先级的仲裁段。  
（3） 控制段。 表示数据的字节及保留位段。  
（4） 数据段。 数据的内容，一帧可发送0~8个字节的数据。  
（5） CRC段。 检查帧的传输错误段。  
（6） ACK段。 表示确认正常接收的段。  
（7） 帧结束。 表示数据的段帧结束。  


114、 Stm32f103系列只有一个CAN控制器，有3个发送邮箱和3级深度的2个FIFO，14个过滤组器。

115、 STM32的每个过滤组可以配置为1个32位过滤器和2个16位过滤器。除此之外，还可以配置为屏蔽位模式（ID+屏蔽）和标识符列表（ID和屏蔽寄存器均用来做ID寄存器）模式。

116、 CAN接收到有效报文被放置在3级邮箱深度的FIFO中，FIFO完全由硬件来管理。

117、 CAN总线的波特率  


118、 触摸屏一般分为电阻式触摸屏和电容式触摸屏。前者检测触摸的位置原理是利用触摸屏控制器中的A/D转换器经过两次A/D读值后得出X和Y的坐标值。注意：这个X和Y的值是相对于触摸屏的，而非LCD屏。所以在这里需要注意两个概念：触摸屏和LCD屏。这是两个不同的概念，也是两个不同的物理结构，其中电阻触摸屏是由上下两个导电层中间夹着一层非常薄的透明隔层；而LCD就是指显示屏。

119、 电阻触摸屏有X和Y、X和Y的比例因子、坐标轴方向、偏移量。LCD也有自己的这些参数。两者完全不相干，所以在定位的时候需要进行坐标转换。公式：  
这里写图片描述  
通过对屏幕的四个点进行校准，得到四元一次方程，求解即可。

120、 NEC协议的数据帧格式：同步码头、地址码、地址反码、控制码、控制反码。同步码由一个9ms的低电平和一个4.5ms的高电平组成，地址码、地址反码、控制码、控制反码均是8位数据格式。按照低位在前，高位在后的顺序发送。

121、 NEC协议在发送的时候，会有560us的38KHz的载波信号，而在接收的时候这部分载波信号被认定为低电平，而剩余的（2.25ms-650us）的逻辑“1”和（1.12ms-650us）的逻辑“0”时间则被认定为高电平。

122、 在单位时间内的位移被定义为速度，速度有线速度和角速度之分，分别对应两种传感器测量这两种不同的速度：线速度传感器（加速度计）、角速度传感器（陀螺仪）。前者多应用在静态或者低慢速运动中的姿态求解，后者多应用在动态运动中姿态求解。

123、 根据标准约定，零加速度（或零 G 准位）通常定义为相当于最大输出值（12 位输出为 4096，10 位输出为 1024 等）一半的输出。对于提供 12 位输出的加速度计，零 G 准位将等于 2048。输出大于 2048 表示正加速度。输出小于 2048 表示负加速度。加速度的数量通常用单位 g (1g = 9.8m/s2 = 重力加速度）表示。通过确定测量的输出与零 G 准位之间的差值，然后除以加速度计的灵敏度（用计数/g 或 LSB/g表示）来计算加速度。对于提供 12 位数字输出的 2g 加速度计，灵敏度为 819 计数/g 或 819 LSB/g。加速度等于：a = (Aout - 2048)/(819 计数/g)，单位为 g。

124、 加速度计测得的加速度的方向和设备设定的坐标系是相反的，因为原理表明在测量力的时候采用的是非惯性系参考系，而我们高中时代研究的坐标系是惯性系参考系，前者在物体进行运动产生加速度时，假想一个与速度方向相反的力作用在物体上，这个力就是惯性力；后者我们说不存在惯性力，只说存在惯性，因为在惯性坐标系中，我们研究的是物体，而非坐标系（即假定坐标系相对地球静止），当我们把坐标系也考虑在内时，当坐标系运动，就产生了惯性力f，这种力作用会假想作用在物体上，只是与运动方向相反。

125、 由上可知，加速度计的本质是测量力而非加速度。

126、 NRF24L01工作在2.4GHz的频段，由于频段频率较高，所以传输速率较快，为2Mbps

127、 STM32的闪存模块由：主存储器、信息块和闪存存储器接口寄存器3个部分构成。主存储器用来存放代码和const常量；信息块由两个部分组成：启动程序代码、用户选择字节。其中启动程序代码为ST公司自带的启动程序，用于串口下载。最后的闪存存储器接口寄存器用于控制整个对闪存区域的操作。

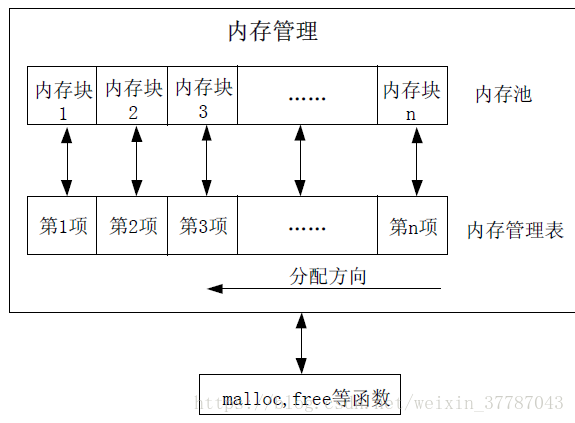
128、 CPU的运行速度比FLASH的操作速度快的多，一般FLASH的最快访问速度≤24Mhz。如果CPU的速度超过这个频率，那么在读取FLASH的时候必须加入等待时间（FLASH\_ACR设置）

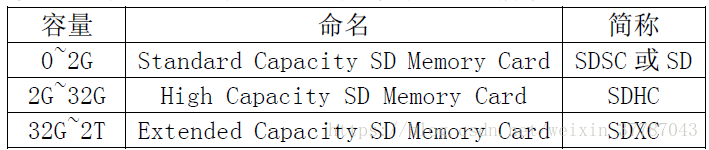
129、 FLASH编程时，写入必须为半字（16位）。并且在写入的时候必须保证所写区域的数据必须为0xFFFF。

130、 STM32的FSMC有HADDR[27:0]，其中[27:26]用来选择BANK区域的4个不同块。剩下的[25:0]则用来连接外部存储区域的地址线FSMC\_A[25:0]。如果数据宽度是8bit，此时的HADDR[25:0]和FSMC\_A[25:0]是完全对应的。如果数据宽度是16bit，此时的HADDR[25:1]和FSMC\_A[24:0]是对应起来的。需要注意：无论数据宽度是多少，外部的FSMC\_A[0]和A[0]总是对应的。

131、 关于LB和UB的信号控制是由硬件自动控制的，当AHB的数据宽度小于外部存储器的数据宽度时，此时LB和UB的控制信号自动产生（比如字节读取/写入16bit的外部存储器）

132、 **attribute** (函数属性、变量属性、类型属性等)。如果在使用SRAM时，可以采用u32 sram\_array[xx] **attribute** ((at(0x68000000))代表将外部SRAM的空间全部给了sram\_array这个变量，他具有在at0x68000000这个地址的属性。往里面写值就直接在SRAM里面写值。

133、内存管理；  
  
内存管理有一种方式叫做分块式内存管理。  
注意表中的分配方向，从顶到底。每一项对应一个内存块。里面的数值代表了内存池的状态：如果为0，表示该内存没有被分配；如果非0，那么数值的大小就表示了该块内存被连续占用的内存数。比如说数值为20，意思是包括该项在内的内存块被连续占用了20块分给了指针。

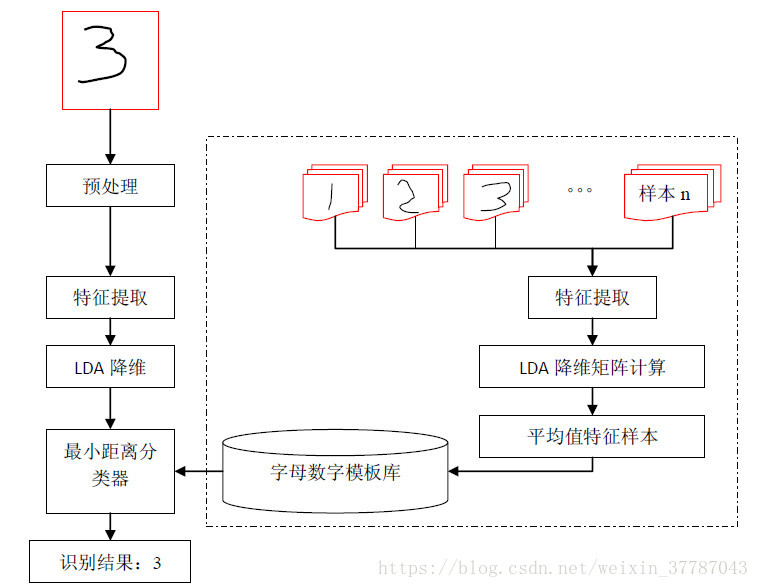
134、SD卡的分类：  
  
一般的SD卡支持两种传输模式：SD卡模式（SDIO）、SPI模式。显然前面一种是专用模式，所以速度比较快。

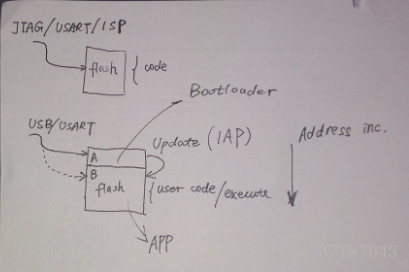
135、 常用的汉字内码系统有GB2313、GB13000、GBK、BIG5(繁体)。其中GB2313只有几千个汉字，而GBK则有2万多汉字。

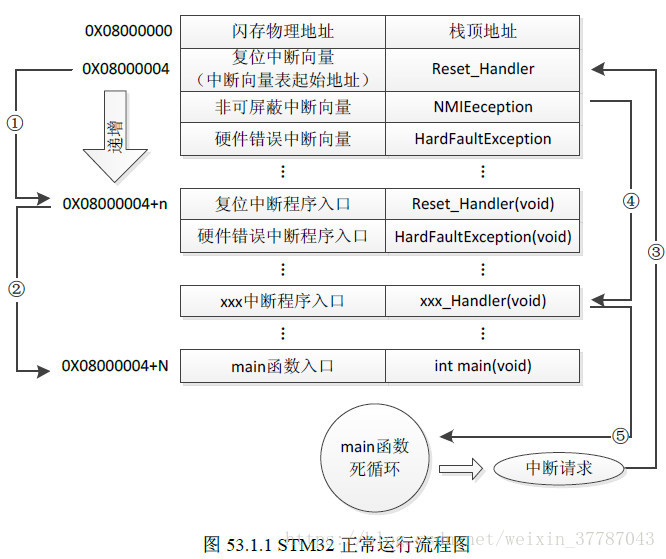
136、 要显示汉字，采用的方式如果用点阵的形式是不可取的，因为这无法查找汉字。采用的方式就是内码系统。GBK标准中，一个汉字对应2个字节：前者称为区(0x81~0xFE)后者为(0x40~0x7E)和(0x80~0xFE)。前者有126个区，后者有190，那么可以显示的汉字数量有126\*190=23940个。根据这两个值用来查找字库，字库中存放的还是每个汉字的点阵数据。这个字库非常大，如果是16\*16的字体，那么一个字体就需要32个字节，如此说来需要23940\*32=748K的空间，可见非常大，所以需要外部的Flash来存储这个字库。

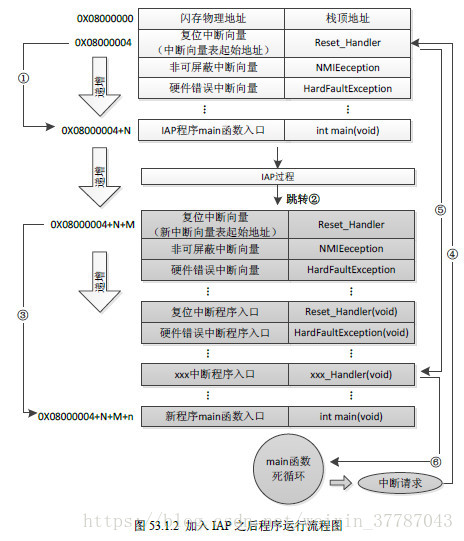
137、 由于汉字内码系统不具有国际通用性，但是Unicode几乎把所有的语言都放置进来，这样在单片机中操作汉字时，就需要将GBK和Unicode转化。尤其是在FATFS中，创建中文文件名和读取中文文件信息时需要将Unicode换转为GBK后再进行修改操作，再反转换成Unicode保存修改。这么说，两者的存在是由于标准的不统一，并且Unicode中只有6064个汉字，而GBK显然是一种汉字扩展。

138、 BMP图片编码的顺序是从左到右，从下到上。

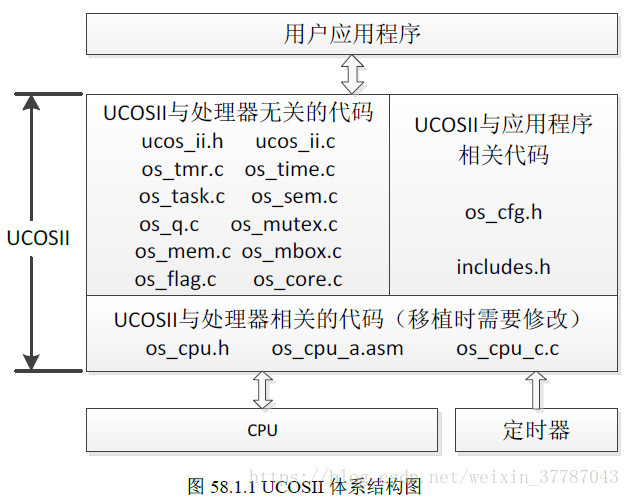
139、 VS1053是一款高性能的数字音频解码芯片，从SD卡中将mp3等音乐音频文件通过SPI送给VS1053后，由其进行音频解码，输出音乐给耳机。耳机驱动可以采用TDA1308芯片，这款芯片为AB类耳机驱动芯片。  


140、 IAP(In Application Programming)在应用编程是为了后期开发更新程序方便而提出的概念；  
  
在普通编程中，flash中的code是通过JTAG和ISP等工具下载到单片机中。而在IAP编程中，flash被分区为A和B两个区域，A区域只允许用USB/USART等方式下载，此区域作为更新B区域的代码用。B区域则是用户的code区域，真正的代码在这里被执行，放置的就是app。

141、STM32正常运行流程图；  
  
上图表示STM32正常运行的流程图，可以看到上电复位后系统从0x80000004处开始运行程序，这里放置的是复位中断向量，然后跳转至复位中断程序入口后再跳转至main函数运行用户的程序。

142、加入IAP之后程序运行流程图；  
  
上图表示加入IAP后的STM32程序运行流程图。可以看到上电复位后跳到IAP程序的main函数处运行IAP过程（这个过程就是把下面灰底色块的程序代码烧进B区域à代码更新）。后面的过程和STM32正常运行一样，如果出现中断请求，还是跳转到A区域中的中断向量表中，然后再跳转到B区域的中断服务入口。

143、 USB有四根线，VCC、GND、D+、D-。在USB主机上，D+和D-均通过一个15K的电阻接地，这样两条线均为低电平。在USB设备中，对于高速设备会在D+通过一个1.5K的电阻接到VCC，而低俗设备会在D-通过一个1.5K的电阻接到VCC。这样主机就可以通过D+和D-的高电平的到来来检测是否有设备接入，并且识别高低速设备。

144、 UCOSII是一种实时操作系统，具有执行效率高、占有空间小（最小内核2KB）、实施性能优良、扩展性强和移植性强等优点。  
  
UCOS具有多任务并发工作的特点（注意，任何时候只有一个任务能够占用CPU。并发只是任务轮流占用CPU而不是同时工作）。最大支持255个任务并发工作。