STM32考试习题及答案

**一、填空题**

1. 当 STM32 的 I/O 端口配置为输入时， 被禁止， 被激活。根据输入配置（上拉，下拉或浮动）的不同，该引脚的 被连接。出现在I/O脚上的数据在每个APB2时钟被采样到输入数据寄存器，对 的访问可得到I/O状态。
2. STM32的所有端口都有外部中断能力。当使用 时，相应的引脚必须配置成 。
3. STM32具有单独的位设置或位清除能力。这是通过 和 寄存器来实现的。
4. ST公司还提供了完善的通用I/O接口库函数，其位于 stm32f10x\_gpio.c ，对应的头文件为 stm32f10x\_gpio.h 。
5. 为了优化不同引脚封装的外设数目，可以把一些 重新映射到其他引脚上。这时，复用功能不再映射到 上。在程序上，是通过设置 复用重映射和调试I/O配置寄存器(AFIO\_MAPR) 来实现引脚的重新映射。
6. ST公司的STM32系列芯片采用了 Cortex-M3内核，其分为两个系列。 STM32F101系列为标准型，运行频率为 36MHz ；STM32F103系列为标准型，运行频率为 72MHz 。
7. STM32全系列芯片都具有引脚到引脚 的特点，并且相同封装的内部资源均相同，这就给用户升级带来很大方便。
8. STM32提供了一种最简单的程序下载方法，即 ，只需要串口连接到PC机上，便可以进行程序下载。
9. STM32 芯片内部集成的 位 ADC 是一种逐次逼近型模拟数字转换器， 具有 个通道，可测量 个外部和 个内部信号源。
10. 在 STM32中，只有在 的转换结束时才产生 请求，并将转换的数据从 寄存器传输到用户指定的目的地址。
11. 在有两个 ADC 的 STM32 器件中，可以使用 模式。在 模式里，根据 寄存器中 DUALMOD[2:0] 位所选的模式，转换的启动可以是ADC1 主和 ADC2 从的交替触发或同时触发。
12. ADC 的校准模式通过设置 ADC\_CR2 寄存器的 CAL 位来启动。
13. 在 STM32 中， ADC\_CR2 寄存器的 ALIGN 位选择转换后数据储存的对齐方式。
14. 在 STM32 内部还提供了 温度传感器 ，可以用来测量器件周围的温度。温度传感器在内部和 ADC\_IN16 输入通道相连接，此通道把传感器输出的电压转换成数字值。内部参考电压 VREFINT 和 ADC\_IN17 相连接。
15. STM32 的 嵌套向量中断控制器(NVIC) 管理着包括Cortex-M3核异常等中断，其和ARM处理器核的接口紧密相连，可以实现 低延迟 的中断处理，并有效地处理 晚到 中断。
16. STM32的外部中断/事件控制器（EXIT）由 19 个产生事件/中断要求的边沿检测器组成。每个输入线可以独立地配置 输入类型（脉冲或挂起）和对应的触发事件（上升沿或下降沿或者双边沿都触发） 。每个输入线都可以被独立的屏蔽。 挂起寄存器 保持着状态线的中断要求。
17. STM32 的 EXTI 线 16 连接到 PVD 输出 。
18. STM32 的 EXTI 线 17 连接到 RTC 闹钟事件 。
19. STM32 的 EXTI 线 18 连接到 USB 唤醒事件 。
20. STM32 的 USART 为通用同步异步收发器，其可以与使用工业标准 NRZ 异步串行数据格式的外部设备之间进行全双工数据交换。
21. STM32 的 USART 可以利用 分数波特率 发生器提供宽范围的波特率选择。
22. 智能卡是一个 单线半双工 通信协议， STM32的智能卡功能可以通过设置 USART\_CR3 寄存器的 SCEN 位来选择。
23. STM32提供了CAN 总线结构，这是一种 基本扩展 CAN(Basic Extended CAN) ，也就是 bxCAN 。
24. 系统计时器（SysTick ）提供了1 个 24 位、降序、零约束、写清除 的计数器， 具有灵活的控制机制。
25. STM32 的通用定时器TIM ，是一个通过 可编程预分频器 驱动的 16 位自动装载计数器构成。
26. STM32通用定时器TIM的16位计数器可以采用三种方式工作，分别为 向上计数 模式、 向下计数 模式和 中央对齐 模式。
27. ST公司还提供了完善的TIM接口库函数，其位于 stm32f10x\_tim.c ，对应的头文件为 stm32f10x\_tim.h 。
28. 除了通用定时器外，STM32还提供了一个高级控制定时器 TIM1 。 TIM1由一个 16 位的自动装载计数器组成，它由一个 可编程预分频器 驱动。
29. TIM1的 溢出 /下溢时更新事件(UEV) 只能在重复向下计数达到0 的时候产生。这对于能产生PWM 信号非常有用。
30. TIM1具备 16 位可编程预分频器，时钟频率的分频系数为 1～ 65535 之间的任意数值。
31. ST 公司还提供了完善的TIM1接口库函数， 其位于 stm32f10x\_tim1.c ， 对应的头文件为 stm32f10x\_tim1.h 。
32. STM32的DMA 控制器有 7 个通道，每个通道专门用来管理来自于一个或多个外设对存储器访问的请求。还有一个 仲裁器 来协调各个DMA 请求的优先权。
33. 在DMA处理时，一个事件发生后，外设发送一个请求信号到 DMA控制器 。DMA 控制器根据通道的 优先权 处理请求。
34. DMA控制器的每个通道都可以在有固定地址的 外设寄存器和存储器地址 之间执行DMA传输。DMA传输的数据量是可编程的，可以通过 DMA\_CCRx 寄存器中的 PSIZE 和 MSIZE 位编程。
35. ST公司还提供了完善的DMA接口库函数，其位于 stm32f10x\_dma.c ，对应的头文件为 stm32f10x\_dma.h 。
36. 在 STM32 中，从外设 (TIMx 、ADC 、SPIx、I2Cx和 USARTx) 产生的 7 个请求，通过逻辑 与 输入到 DMA 控制器，这样同时 只能有一 个请求有效。
37. 在STM32中，闪存存储器有 主存储块 和 信息块 组成。
38. STM32的Flash闪存的指令和数据访问是通过 AHB 总线完成的。预取模块是用于通过 ICode 总线读取指令的。仲裁是作用在闪存接口，并且 DCode 总线上的数据访问优先。
39. STM32的Flash闪存编程一次可以写入 16 位。STM32的Flash闪存擦除操作可以按 页面 擦除或 完全 擦除， 完全擦除 不影响信息块。写操作(编程或擦除)结束时可以触发中断。仅当闪存控制器接口时钟开启时，此中断可以用来从 WFI 模式退出。
40. ST公司还提供了完善的Flash闪存接口库函数，其位于 stm32f10x\_flash.c ，对应的头文件为 stm32f10x\_flash.h 。
41. 选项字节寄存器结构中，RDP为 读出选项字节 ，USER为 用户选项字节 ，Data0为 数据0 选项字节，Data1为 数据1 选项字节，WRP0为 写保护0 的选项字节，WRP1为 写保护1 的选项字节，WRP2为 写保护2 的选项字节，WRP3为 写保护3 的选项字节。
42. STM32系列ARM Cortex-M3芯片支持三种复位形式，分别为 系统 复位、 电源 复位和 备份区域 复位。
43. STM32还提供了用户可通过多个预分频器，可用来进一步配置 AHB 、高速 APB(APB2) 和低速 APB(APB1) 域的频率。
44. 用户可用通过 32.768k Hz外部振荡器，为系统提供更为精确的主时钟。在时钟控制寄存器 RCC\_CR 中的 HSERDY 位用来指示高速外部振荡器是否稳定。
45. ST公司还提供了完善的RCC接口库函数，其位于 stm32f10x\_rcc.c ，对应的头文件为 stm32f10x\_rcc.h 。
46. ST公司还提供了完善的RTC接口库函数，其位于 stm32f10x\_rtc.c ，对应的头文件为 stm32f10x\_rtc.h 。
47. 当STM32复位后， HSI振荡器 将被选为系统时钟。当时钟源被直接或通过PLL 间接作为系统时钟时，它将不能被 停止 。只有当 目标时钟源 准备就绪了(经过启动稳定阶段的延迟或PLL 稳定)，才可以从一个时钟源切换到另一个时钟源。在被选择时钟源没有就绪时，系统时钟的切换 不会发生 。
48. 在STM32中，备份寄存器是 16 位的寄存器，共 10 个，可以用来存储 20 个字节的用户应用程序数据。
49. 备份寄存器位于 备份域 里，当 主电源VDD 被切断，他们仍然由 VBAT 维持供电。当系统在待机模式下被唤醒，或系统复位或电源复位时，他们 也不会 被复位。
50. STM32的备份寄存器还可以用来实现 RTC 校准功能。为方便测量，32.768kHz的RTC 时钟可以输出到 侵入检测 引脚上。通过设置RTC 校验寄存器(BKP\_RTCCR)的 CCO 位来开启这一功能。
51. 当STM32的 ANTI\_TAMP 引脚上的信号发生跳变时，会产生一个侵入检测事件，这将使所有数据备份寄存器 复位 。
52. ST公司还提供了完善的备份寄存器接口库函数，其位于 stm32f10x\_bkp.c ，对应的头文件为 stm32f10x\_bkp.h 。

**二、选择题**

1. 在APB2上的I/O脚的翻转速度为（ A ）。

A．18MHz B．50MHz

C．36MHz D．72MHz

1. 当输出模式位MODE[1:0]=“10”时，最大输出速度为（ B ）。

A．10MHz B．2MHz

C．50MHz D．72MHz

1. 下列哪个不是RealView MDK开发环境的特点（ D ）。

A．Windows风格 B．兼容的Keil μVision界面

C．全面的ARM处理器支持 D．体积庞大

1. 下列哪种方法可以对STM32进行程序下载（ ABCD ）。（多选）

A．Keil ULink B．J-Link

C. 在应用编程 D．以上都可以

1. 下列哪些不是STM32闪存存储器的特点（ C ）。

A．大容量 B．高速

C．掉电不保存 D．具有选择字节加载器

1. 哪些是STM32的ADC系统的特点（多选）（ ABCD ）。

A．12-位分辨率 B．自校准

C．可编程数据对齐 D．单次和连续转换模式

1. 在ADC的扫描模式中，如果设置了DMA位，在每次EOC后，DMA控制器把规则组通道的转换数据传输到（ A ）中。

A．SRAM B．Flash

C．ADC\_JDRx寄存器 D．ADC\_CR1

1. STM32规则组由多达（ A ）个转换组成。

A．16 B．18

C．4 D．20

1. 在STM32中，（ A ）寄存器的ALIGN位选择转换后数据储存的对齐方式。

A．ADC\_CR2 B．ADC\_JDRx

C. ADC\_CR1 D．ADC\_JSQR

1. ARM Cortex-M3不可以通过（  D  ）唤醒CPU。

A．I/O端口  B．RTC 闹钟

C．USB唤醒事件  D．PLL

1. 通用定时器 TIMx 的特性（ ABCD ）。（多选）

A．具备 16 位向上，向下，向上 /向下自动装载计数器。

B．具备 16 位可编程预分频器。

C．具备 4 个独立通道。

D．可以通过事件产生中断，中断类型丰富，具备 DMA 功能。

1. 通用定时器 TIMx 的特殊工作模式包括（ ABCD ）。（多选）

A ．输入捕获模式 B． PWM输入模式

C. 输出模式 D．单脉冲模式(OPM)

1. STM32的可编程TIM1定时器的时基单元包含（ ABCD ）。（多选）

A．计数器寄存器(TIM1\_CNT)

B．预分频器寄存器 (TIM1\_PSC)

C．自动装载寄存器 (TIM1\_ARR)

D．周期计数寄存器 (TIM1\_RCR)

1. 高级定时器TIM1的特性（ ABCD ）。（多选）

A．具备16位上，下，上/下自动装载计数器

B．具备16位可编程预分频器。

C．可以在指定数目的计数器周期之后更新定时器寄存器。

D．可以通过事件产生中断，中断类型丰富，具备DMA功能。

1. 定时器TIM1的特殊工作模式包括（ ABCD ）。（多选）

A．输入捕获模式 B．PWM 输入模式

C. 编码器接口模式 D．单脉冲模式(OPM)

1. STM32提供了三种不同的时钟源，其都可被用来驱动系统时钟SYSCLK，这三种时钟源分别为（ ABC ）。

A．HSI振荡器时钟 B．HSE振荡器时钟

C．PLL时钟 D．HLI振荡时钟

1. 在STM32中，当（ AB ）发生时，将产生电源复位。（多选）

A．从待机模式中返回 B．上电/掉电复位（POR/PDR复位）

C．NRST管脚上的低电平 D．PLL

1. 以下哪个时钟信号可被选作MCO 时钟（ ABCD ）。（多选）

A．SYSCLK B．HSI

C．HSE D．以2分频的PLL 时钟

1. STM32的Flash闪存编程一次可以写入（ A ）位。

A．16 B．8

C．32 D．4

1. STM32主存储块的页大小为（ A ） 字节。

A．1K B．3K

C．2K D．4K

1. 用户选择字节的大小为（ A ）。

A．512字节 B．2K

C．1K D．128K

1. STM32嵌套向量中断控制器(NVIC) 具有（  A  ） 个可编程的优先等级。

A．16  B．43

C．72  D．36

1. STM32的外部中断/事件控制器（EXTI）支持（ C   ）个中断/事件请求。

A．16  B．43

C．19  D．36

1. STM32的USART根据（ A ）寄存器M位的状态，来选择发送8位或者9位的数据字。

A．USART\_CR1 B．USART\_CR2

C．USART\_BRR D．USART\_CR3

1. STM32的bxCAN的主要工作模式为（ ABD ）。

A．初始化模式 B．正常模式

C．环回模式 D．睡眠模式

1. 在程序中，可以将CAN\_BTR寄存器的（ AB ）位同时置1，来进入环回静默模式。（多选）

A．LBKM B．SILM

C. BTR D．以上都不是

1. 在STM32中，备份寄存器是（ A ）的寄存器。

A．16 位 B．32 位

C．8 位 D．4 位

1. 为了允许访问备份寄存器和RTC，电源控制寄存器(PWR\_CR)的DBP 位必须置为（ A ）。

A．1 B．2

C．0 D．3

1. 下列哪个不是备份寄存器（ C ）。

A．BKP\_DR1 B．BKP\_DR3

C．BKP\_RTCCR D．BKP\_DR5

1. DMA控制器可编程的数据传输数目最大为（ A ）。

A．65536 B．65535

C．1024 D．4096

1. 每个DMA通道具有（ A ）个事件标志。

A．3 B．4

C．5 D．6

1. DMA控制器中，独立的源和目标数据区的传输宽度为（ ABCD ）（多选）。

A．字节 B．半字

C．全字 D．以上都可以

1. STM32中，1 个DMA请求占用至少（ B ）个周期的CPU 访问系统总线时间。

A．1 B．2

C．3 D．4

**三、简答题**

1. **简述ARM RealView开发环境的特点。**

答：ARM RealView MDK开发平台具有如下主要特点：

采用Keil μVision3的开发环境和界面，给单片机用户的升级带来极大的方便。

具有Windows风格的可视化操作界面，界面友好，使用极为方便；

支持汇编语言、C51语言以及混合编程等多种方式的单片机设计；

集成了非常全面的ARM处理器支持，能够完成ARM7、ARM9以及ARM Cortex-M3等处理器的程序设计和仿真；

集成了丰富的库函数，以及完善的编译连接工具；

提供了并口、串口、A/D、D/A、定时器/计数器以及中断等资源的硬件仿真能力，能够帮助用户模拟实际硬件的执行效果；

可以与多款外部仿真器联合使用，提供了强大的在线仿真调试能力；

内嵌RTX-51 Tiny和RTX-51 FULL内核，提供了简单而强大的实时多任务操作系统支持；

在一个开发界面中支持多个项目的程序设计；

支持多级代码优化，最大限度地帮助用户精简代码体积；

由于Keil μVision具有最为广泛的用户群，因此相应的代码资源非常丰富，读者可以轻松地找到各类编程资源以加速学习和开发过程。

1. **简述基于ARM Cortex-M3的STM32芯片特点。**

答：ST公司的STM32系列芯片采用了ARM Cortex-M3内核，其分为两个系列。STM32F101系列为标准型，运行频率为36MHz；STM32F103系列为标准型，运行频率为72MHz。STM32全系列芯片都具有引脚到引脚一一对应的特点，并且相同封装的内部资源均相同，这就给用户升级带来很大方便。

1. **简述STM32最小硬件开发系统的组成及其各部分的作用。**

答：读者可以参阅正文中的第3节，其中包括如下几个部分：

主芯片：采用STM32系列，为整个开发系统的核心，本书所有的程序都运行在其中。

晶体振荡部分：提供了硬件时序以及实时时钟使用。

供电部分：采用AM1117为芯片提供稳定的3.3V电压。

复位部分：采用一个按键开关来实现。

1. **简述不同复用功能的重映射。**

答：为了优化不同引脚封装的外设数目，可以把一些复用功能重新映射到其他引脚上。这时，复用功能不再映射到它们原始分配的引脚上。在程序上，是通过设置复用重映射和调试I/O配置寄存器(AFIO\_MAPR)来实现引脚的重新映射。各个复用功能的重映射可以参阅正文的介绍，由于内容比较多，正文介绍非常详细，这里省略。

1. **简述STM32的GPIO的一些主要特点（至少5个）。**

答：主要特点如下：

通用I/O，可以作为输出、输入等功能。

单独的位设置或位清除。

外部中断/唤醒线。

复用功能(AF)和重映射。

GPIO锁定机制。

1. **简述STM32的ADC系统的功能特性。**

答：STM32的ADC系统的主要功能特性包括如下几个方面：ADC开关控制、ADC时钟、ADC通道选择、ADC的转换模式、中断、模拟看门狗、ADC的扫描模式、ADC的注入通道管理、间断模式、ADC的校准模式、ADC的数据对齐、可编程的通道采样时间、外部触发转换、DMA请求、双ADC模式和温度传感器。

1. **简述STM32的双ADC工作模式。**

答：在有两个ADC的STM32器件中，可以使用双ADC模式。在双ADC模式里，根据ADC\_CR1寄存器中DUALMOD[2:0]位所选的模式，转换的启动可以是ADC1主和ADC2从的交替触发或同时触发。双ADC工作模式主要包括如下几种：同时注入模式、同时规则模式、快速交替模式、慢速交替模式、交替触发模式和独立模式。

1. **简述嵌套向量中断控制器（NVIC）的主要特性。**

答：STM32的嵌套向量中断控制器(NVIC) 管理着包括Cortex-M3核异常等中断，其和ARM处理器核的接口紧密相连，可以实现低延迟的中断处理，并有效地处理晚到的中断。STM32嵌套向量中断控制器(NVIC)的主要特性如下：

具有43 个可屏蔽中断通道（不包含16 个Cortex-M3 的中断线）。

具有16 个可编程的优先等级。

可实现低延迟的异常和中断处理。

具有电源管理控制。

系统控制寄存器的实现。

1. **简述STM32的USART的功能特点。**

答：STM32的USART为通用同步异步收发器，其可以与使用工业标准NRZ异步串行数据格式的外部设备之间进行全双工数据交换。USART还可以利用分数波特率发生器提供宽范围的波特率选择。

STM32的USART支持同步单向通信和半双工单线通信。同时，其也支持LIN(局部互连网)，智能卡协议和IrDA(红外数据)SIR ENDEC规范，以及调制解调器(CTS/RTS)操作。STM32还具备多处理器通信能力。另外，通过多缓冲器配置的DMA方式，还可以实现高速数据通信。

1. **简述STM32TIM的计数器模式。**

答：STM32通用定时器TIM的16位计数器可以采用三种方式工作，分别为向上计数模式、向下计数模式和中央对齐模式(向上/向下计数)。

1. **简述STM32的高级控制定时器TIM1的结构。**

答：STM32提供了一个高级控制定时器(TIM1)。TIM1由一个16位的自动装载计数器组成，它由一个可编程预分频器驱动。TIM1适合多种用途，包含测量输入信号的脉冲宽度，或者产生输出波形。使用定时器预分频器和RCC时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

高级控制定时器TIM1和通用控制定时器TIMx是完全独立的，它们不共享任何资源，因此可以同步操作。

1. **简述STM32复位的类型。**

答：STM32系列ARM Cortex-M3芯片支持三种复位形式，分别为系统复位、电源复位和备份区域复位。

1. **简述STM32时钟的类型。**

答：STM32提供了三种不同的时钟源，其都可被用来驱动系统时钟SYSCLK，这三种时钟源分别为：

HSI振荡器时钟

HSE振荡器时钟

PLL时钟

这三种时钟源还可以有以下2种二级时钟源：

32kHz低速内部RC，可以用于驱动独立看门狗和RTC。其中，RTC用于从停机/待机模式下自动唤醒系统。

32.768kHz低速外部晶振也可用来驱动RTC(RTCCLK)。

任一个时钟源都可被独立地启动或关闭，这样可以通过关闭不使用的时钟源来优化整个系统的功耗。

1. **简述STM32实时时钟RTC的配置步骤。**

答：在程序中，配置RTC寄存器步骤如下：

（1）查询RTC\_CR 寄存器中的RTOFF位，直到RTOFF的值变为“1”，表示前一次写操作结束。

（2）置CNF值为1，进入配置模式。

（3）对一个或多个RTC 寄存器进行写操作。

（4）清除CNF 标志位，退出配置模式。

（5）查询RTOFF，直至RTOFF 位变为“1” 以确认写操作已经完成。

1. **简述DMA控制器的基本功能。**

答：STM32的DMA 控制器有7个通道，每个通道专门用来管理来自于一个或多个外设对存储器访问的请求。还有一个仲裁器来协调各个DMA 请求的优先权。DMA 控制器和Cortex-M3核共享系统数据线执行直接存储器数据传输。因此，1 个DMA请求占用至少2 个周期的CPU 访问系统总线时间。为了保证Cortex-M3 核的代码执行的最小带宽，DMA 控制器总是在2 个连续的DMA 请求间释放系统时钟至少1 个周期。

**四、编程题**

1. **编写一个初始化定时器的程序。**

答： 由于还没有讲到定时器相关的知识，所以这里旨在让读者给出定时器对GPIO 端口的设置要求，程序示例如下：

GPIO\_InitTypeDef GPIO\_InitStructure;

/\* GPIOC Configuration: Pin6, 7, 8 and 9 in Output \*/

GPIO\_InitStructure.GPIO\_Pin=GPIO\_Pin\_6|GPIO\_Pin\_7 | GPIO\_Pin\_8 | GPIO\_Pin\_9;

GPIO\_InitStructure.GPIO\_Mode=GPIO\_Mode\_Out\_PP; GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOC, &GPIO\_InitStructure);

1. **给出PWM模式下配置TIM外设的程序代码。**

答：

/\* Time Base configuration \*/

TIM\_TimeBaseStructure.TIM\_Prescaler = 0;

TIM\_TimeBaseStructure.TIM\_CounterMode = TIM\_CounterMode\_Up;

TIM\_TimeBaseStructure.TIM\_Period = 4095;

TIM\_TimeBaseStructure.TIM\_ClockDivision = 0;

TIM\_TimeBaseStructure.TIM\_RepetitionCounter = 0;

TIM\_TimeBaseInit(TIM1, &TIM\_TimeBaseStructure);

/\* Channel 1, 2,3 and 4 Configuration in PWM mode \*/

TIM\_OCInitStructure.TIM\_OCMode = TIM\_OCMode\_PWM2;

TIM\_OCInitStructure.TIM\_OutputState = TIM\_OutputState\_Enable;

TIM\_OCInitStructure.TIM\_OutputNState = TIM\_OutputNState\_Enable;

TIM\_OCInitStructure.TIM\_Pulse = CCR1\_Val;

TIM\_OCInitStructure.TIM\_OCPolarity = TIM\_OCPolarity\_Low;

TIM\_OCInitStructure.TIM\_OCNPolarity = TIM\_OCNPolarity\_High;

TIM\_OCInitStructure.TIM\_OCIdleState = TIM\_OCIdleState\_Set;

TIM\_OCInitStructure.TIM\_OCNIdleState = TIM\_OCIdleState\_Reset;

TIM\_OC1Init(TIM1, &TIM\_OCInitStructure);

TIM\_OCInitStructure.TIM\_Pulse = CCR2\_Val;

TIM\_OC2Init(TIM1, &TIM\_OCInitStructure);

TIM\_OCInitStructure.TIM\_Pulse = CCR3\_Val;

TIM\_OC3Init(TIM1, &TIM\_OCInitStructure);

TIM\_OCInitStructure.TIM\_Pulse = CCR4\_Val;

TIM\_OC4Init(TIM1, &TIM\_OCInitStructure);

/\* TIM1 counter enable \*/

TIM\_Cmd(TIM1, ENABLE);

/\* TIM1 Main Output Enable \*/

TIM\_CtrlPWMOutputs(TIM1, ENABLE);