第6章 NVIC

6.1 NVIC简介
6.2 常用库函数
6.3 应用实例

电子与通信工程系

华东理工大学信息学院

6.1 NVIC简介

6.1 NVIC简介

NVIC：嵌套向量中断控制器（Nested Vectored Interrupt Controller）

1)、CM4内核支持256个中断，其中包含了16个内核中断和240个外部中断，并且具有256级的可编程中断设置。

2)、 STM32F4并没有使用CM4内核的全部中断定义，而是只用了它的一部分。 -STM32F42xx/STM32F43xx则总共有101个中断

3)、 STM32F42xx/STM32F43xx的101个中断里面，包括10个内核中断和91个可屏蔽中断，具有16级可编程的中断优先级，而我们常用的就是这91个可屏蔽中断。

6.1 NVIC简介

6.1.1 中断类型

1、10个异常

最多可以有16个

《STM32F4xx中文参考手册》P237 表46

优先级	优先级类型	名称	说明	地址
-	-	-	保留（实际存的是 MSP 地址）	0x0000 0000
-3	固定	Reset	复位	0x0000 0004
-2	固定	NMI	不可屏蔽中断。 RCC 时钟安全系统	0x0000 0008
			(CSS) 连接到 NMI 向量	
-1	固定	HardFault	所有类型的错误	0x0000 000C
0	可编程	MemManage	存储器管理	0x0000 0010
1	可编程	BusFault	预取指失败，存储器访问失败	0x0000 0014
2	可编程	UsageFault	未定义的指令或非法状态	0x0000 0018
-	-	-	保留	0x0000 001C-0x0000 002B
3	可编程	SVCall	通过 SWI 指令调用的系统服务	0x0000 002C
4	可编程	Debug Monitor	调试监控器	0x0000 0030
-	-	-	保留	0x0000 0034
5	可编程	PendSV	可挂起的系统服务	0x0000 0038
6	可编程	SysTick	系统嘀嗒定时器	0x0000 003C

6.1 NVIC简介

6.1.1 中断类型

2、外设可屏蔽中断

0	7	settable	WWDG	Window Watchdog interrupt	0x0000 0040
1	8	settable	PVD	PVD through EXTI line detection interrupt	0x0000 0044
2	9	settable	TAMP_STAMP	Tamper and TimeStamp interrupts through the EXTI line	0x0000 0048
3	10	settable	RTC_WKUP	RTC Wakeup interrupt through the EXTI line	0x0000 004C
4	11	settable	FLASH	Flash global interrupt	0x0000 0050
5	12	settable	RCC	RCC global interrupt	0x0000 0054
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000 0058

87	94	settable	SAI1	SAI1 global interrupt	0x0000 019C
88	95	settable	LCD-TFT	LTDC global interrupt	0x0000 01A0
89	96	settable	LCD-TFT	LTDC global Error interrupt	0x0000 01A4
90	97	settable	DMA2D	DMA2D global interrupt	0x0000 01A8

6.1 NVIC简介

6.1.2 NVIC中断管理方法

1、中断优先级分组

对STM32中断进行分组为0~4。同时，对每个中断设置一个抢占优先级和一个响应优先级值。

分组配置是在寄存器SCB_AIRCR(Application interrupt and reset control register:应用中断和复位控制寄存器)中配置：

组	AIRCR[10：8]	IP bit[7：4]分配情况	分配结果
0	111	0:4	0位抢占优先级，4位响应优先级
1	110	1:3	1位抢占优先级，3位响应优先级
2	101	2:2	2位抢占优先级，2位响应优先级
3	100	3:1	3位抢占优先级，1位响应优先级
4	011	4:0	4位抢占优先级，0位响应优先级

中断优先级寄存器：

7	6	5	4	3	2	1	0

6.1 NVIC简介

6.1.2 NVIC中断管理方法

2、中断优先级管理方法

1）抢占优先级较高的中断是可以打断正在进行的抢占优先级较低的中断；

2）抢占优先级相同的中断，响应优先级高的不可以打断响应优先级低的中断；

3）抢占优先级相同的中断，当两个中断同时发生的情况下，哪个响应优先级高，哪个先执行；

4）如果两个中断的抢占优先级和响应优先级都是一样的话，则看哪个中断先发生就先执行。

5）如果两个中断的抢占优先级和响应优先级都是一样的话，且同时请求，则根据中断表中的排位顺序决定。

抢占式优先级>响应优先级>中断表中的排位顺序

6.1 NVIC简介

6.1.2 NVIC中断管理方法

3、举例

假定设置中断优先级组为2，然后设置:
--中断3(RTC中断)的抢占优先级为2，响应优先级为1。
--中断6（外部中断0）的抢占优先级为3，响应优先级为0。
--中断7（外部中断1）的抢占优先级为2，响应优先级为0。

那么这3个中断的优先级顺序为：中断7>中断3>中断6。

6.1 NVIC简介

6.1.2 NVIC中断管理方法

4、特别说明

一般情况下，系统代码执行过程中，只设置一次中断优先级分组，比如:分组2，设置好分组之后一般不会再改变分组。

随意改变分组会导致中断管理混乱，程序出现意想不到的执行结果。

6.2 常用库函数

6.2 常用库函数

misc.h
misc.c

1、中断优先级分组：

void NVIC_PriorityGroupConfig(uint32_t NVIC_PriorityGroup)

涉及到的寄存器：SCB_AIRCR

```
void NVIC_PriorityGroupConfig(uint32_t NVIC_PriorityGroup)
{
  assert_param(IS_NVIC_PRIORITY_GROUP(NVIC_PriorityGroup));
  SCB->AIRCR = AIRCR_VECTKEY_MASK | NVIC_PriorityGroup;
}
```

例：NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);

6.2 常用库函数

2、中断优先级设置

void NVIC_Init(NVIC_InitTypeDef NVIC_InitStruct);*

```
typedef struct
{
  __IO uint32_t ISER[8];
       uint32_t RESERVED0[24];
  __IO uint32_t ICER[8];
       uint32_t RSERVED1[24];
  __IO uint32_t ISPR[8];
       uint32_t RESERVED2[24];
  __IO uint32_t ICPR[8];
       uint32_t RESERVED3[24];
  __IO uint32_t IABR[8];
       uint32_t RESERVED4[56];
  __IO uint8_t  IP[240];
       uint32_t RESERVED5[644];
  __O  uint32_t STIR;
}  NVIC_Type;
```

涉及到的寄存器：

__IO uint8_t IP[240]; //中断优先级控制的寄存器组
__IO uint32_t ISER[8]; //中断使能寄存器组
__IO uint32_t ICER[8]; //中断失能寄存器组
__IO uint32_t ISPR[8]; //中断挂起寄存器组
__IO uint32_t ICPR[8]; //中断解挂寄存器组
__IO uint32_t IABR[8]; //中断激活标志位寄存器组

6.2 常用库函数

2、中断优先级设置

void NVIC_Init(NVIC_InitTypeDef NVIC_InitStruct);*

```
typedef struct
{
  uint8_t NVIC_IRQChannel; //设置中断通道
  uint8_t NVIC_IRQChannelPreemptionPriority;//设置抢占优先级
  uint8_t NVIC_IRQChannelSubPriority; //设置响应优先级
  FunctionalState NVIC_IRQChannelCmd; //使能/禁止
} NVIC_InitTypeDef;
```

```
NVIC_InitTypeDef   NVIC_InitStructure;
NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;// 37 串口1中断
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority=1;// 抢占优先级为1
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 2;//响应优先级为2
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;//IRQ通道使能
NVIC_Init(&NVIC_InitStructure);    //根据上面指定的参数初始化NVIC寄存器
```

6.3 应用实例

6.3 应用实例

中断优先级设置步骤

① 系统运行后先设置中断优先级分组。调用函数：

```
void NVIC_PriorityGroupConfig(uint32_t
  NVIC_PriorityGroup);
```

整个系统执行过程中，只设置一次中断分组。

②针对每个中断，设置对应的抢占优先级和响应优先级：

```
void NVIC_Init(NVIC_InitTypeDef* NVIC_InitStruct);
```

③ 如果需要挂起/解挂，查看中断当前激活状态，分别调用相关函数即可。用的不多！

通常片上外设中断管理只需要第1步和第2步！

6.3 应用实例

编写NVIC中断初始化程序实现如下功能：

1、设置中断优先级组为1组

2、设置外部中断1的抢占优先级为0，响应优先级为2

3、设置定时器1的溢出更新中断的抢占优先级为1，响应优先级为4

4、设置USART1的抢占优先级为1，响应优先级为5

并说明当同时出现以上3个中断请求时，中断服务程序执行的顺序？

6.3 应用实例

```
void NVIC_Configuration(void)
{
  NVIC_InitTypeDef   NVIC_InitStructure;//声明NVIC初始化临时变量
  /* 将NVIC中断优先级组设置为1组*/ ①
 NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);//1位抢占优先级可以是：0和1
                                                //3位响应优先级可以是：0-7

 /* 1设置中断源 为EXTI1_IRQn */ ②-1
  NVIC_InitStructure.NVIC_IRQChannel = EXTI1_IRQn;
 /* 设置抢先优占级*/
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
 /*设置相应优先级*/
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 2;
 /* 使能这一中断 */
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
 /* 完成以上中断的初始化功能*/
 NVIC_Init(&NVIC_InitStructure);
```

```
199    EXTI1_IRQn                  = 7,
```

中断号：定义在stm32f4xx.h

注意：当有多个片上中断需要初始化时，需要分别调用NVIC_Init(&NVIC_InitStructure)进行单独初始化，相互独立不影响。！！！

6.3 应用实例

```
/* 2设置中断源 为TIM1_UP_TIM10_IRQn */②-2
 NVIC_InitStructure.NVIC_IRQChannel = TIM1_UP_TIM10_IRQn;
/* 设置抢占优先级*/
 NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
/*设置相应优先级*/
 NVIC_InitStructure.NVIC_IRQChannelSubPriority = 4;
/* 使能这一中断 */
 NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
/* 完成以上中断的初始化功能*/
 NVIC_Init(&NVIC_InitStructure);

/* 3设置中断源 为USART1_IRQn*/          ②-3
 NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
/* 设置抢占优先级*/
 NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
/*设置相应优先级*/
 NVIC_InitStructure.NVIC_IRQChannelSubPriority = 5;
/* 使能这一中断 */
 NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
/* 完成以上中断的初始化功能*/
 NVIC_Init(&NVIC_InitStructure);
}//初始化函数结束
```

6.3 应用实例

1、设置中断优先级组为1组
2、设置外部中断1的抢占优先级为0，响应优先级为2
3、设置定时器1的溢出更新中断的抢占优先级为1，响应优先级为4
4、设置USART1的抢占优先级为1，响应优先级为5
并说明当同时出现以上3个中断请求时，中断服务程序执行的顺序？

三个中断的优先级次序：

EXTI1_IRQn-> TIM1_UP_TIM10_IRQn-> USART1_IRQn

程序中的EXTI1_IRQn、TIM1_UP_TIM10_IRQn以及USART1_IRQn用于表示相应中断的中断号。

都定义在头文件“**stm32f4xx.h**”中。

其他的片上外设中断源的设置都可以按照以上的方法进行。