Posted on **January 30, 2013**

I am testing nested interrupt recently on my STM32 dev board in order to ensure the relationships between nested interrupts are clear for me. I've read lots of sample codes online but never tried on my own since I only used systick and timer based encoder interrupt last year.

before set up the priority for interrupts, we need to determine the NVIC priority group first.

NVIC refers to nested vector interrupt controller, is a controller built in cortex arm M3 M4 processors, therefore this feature can also be found at some other brand's arm M3 M4 processors other than stm32.

There are 2 different kinds of priorities: **preemption priorities and sub priorities**

Usually whoever has higher preemption priority can be executed first, this is always happening with nested interrupts. when 2 interrupts have same preemption priority, then the one who has higher sub priority will be execute first. If both interrupts both have same preemption priority and sub priority, the one comes first will be execute first(first come first serve).

Cortex Arm M3 M4 processors use 8 bits to store the priorities for preemption and sub priorities. In the header file core_CM3.h or core_CM4.h, it uses only 4 bits to store those values. If we use all 4 bits to store preemption priority, then there will be no bit left for sub priority, and vice versa, there will be no bit available for preemption priority when all 4bits are used for sub priority. This process is call priority grouping. There are 5 different groups we can set.

group0 0 bits for preemption, 4 bits for sub priority

group1 1 bits for preemption, 3 bits for sub priority

group2 2 bits for preemption, 2 bits for sub priority

group3 3 bits for preemption, 1 bits for sub priority

group4 4 bits for preemption, 0 bits for sub priority

thus, we can call function NVIC_PriorityGroupConfig to setup priority grouping now.

I choose group 4 since I only want different preemption priority in my micromouse between interrupts, so I call function as follow:

```
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_4);    //4 bits for preemp priority 0 bit for sub priority
```

NVIC_PriorityGroupConfig(NVIC_PriorityGroup_4);   //4 bits for preemp priority 0 bit for sub priority

If you don't set the priority grouping, the default grouping in ST library is group 2, which is 2 bits for preemption priority and 2 bits for sub priority

now we can go set the priorities for each peripheral interrupts.

For instance, I want to set the priority for TIM4 as 1 (lower number has higher priority), I call the code as follow:

```
NVIC_InitTypeDef NVIC_InitStructure;

NVIC_InitStructure.NVIC_IRQChannel = TIM4_IRQn; //TIM4 IRQ Channel
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;//Preemption Priority
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0; //Sub Priority
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
```

If you want to setup the priorities for other device, just replace 'TIM4_IRQn' to the one you want.

However, if you want to setup the priorities for those peripherals whose values for NVIC vector table are smaller than 0, the method we used above will be no long valid. The correct way is to call the function as follow(assume I want systick has highest priority):
NVIC_SetPriority(SysTick_IRQn, 0);//set systick interrupt priority, 0 is the highest for all

you might be confused here since this function didn't specify the preemption and sub priories in parameter. In fact, they are both included. Since I mentioned stm32 uses 4 bits to store preemption and sub priorities, the high bits are for preemption and low bits are for sub priority. If the parameter for priority is 0 for the function above(that's also what I had in the function), 0 is 0x00 in hex and 0000 in binary, since the default priority grouping for stm32 is group 2, which is 2 bits for preemption priority and 2 bits for sub priority, we get 00 for preemption and 00 for sub. If we change the input from 0 to 7 for the function, the binary form for 7 is 0111, so higher 2 bits is 01 which is 1 in decimal gives you preemption priority as 1, and the lower 2 bits is 11 which is 3 gives you sub priority as 3.

If your priority grouping is same as mine which is group 4, then there is no sub priority, the input in parameter in function NVIC_SetPriority only refers to your preemption priority in this case.

PS: never set your preemption and sub priorities out of range, it might work, but something unexpected may happen.

This entry was posted in **STM32** by **Green**. Bookmark the **permalink [http://micromouseusa.com/?p=279]** .

23 THOUGHTS ON “HOW TO CONFIG INTERRUPTS PRIORITIES FOR STM32”

bahram rouzdar
on **January 2, 2014 at 5:11 am** said:

Thanks a lot for this useful information.i have a question:Have the STM32F10x micro controllers a Global Interrupt bit to Enable or Disable all interrupts?
Thanks
form azerbaijan.

Green
on **January 2, 2014 at 5:21 am** said:

i don’t rememberthere is anything to serve such a purpose. i just wrote my own macro to disable my interrupts all toghther.

Patrick Hisni Brataas
on **January 20, 2014 at 4:42 pm** said:

You can use __enable_irq() and __disable_irq() on STM32.

hellboy
on **September 16, 2014 at 5:56 pm** said:

thanks. I have just cleared my doubts about this.

saber mirzaei
on **January 7, 2015 at 11:50 am** said:

hi.. thank's for very useful information..

Robert Thomas
on **October 19, 2015 at 11:13 pm** said:

If you don't set the priority grouping, the default grouping in ST library is group 2, which is 2 bits for preemption priority and 2 bits for sub priority.

Where did you find the above default value. I have looked all over and think I noticed something in the misc.c file at line 123, but I'm not sure. I have notice several programmers leave it out and the program works fine. I just what to see it with my own eyes.

Thanks

Green
on **October 20, 2015 at 3:32 am** said:

Try to read default value from system before you make any change.
One core.cm4.h for Arm M4 MCU you will see this:
/** \brief Get Priority Grouping

The function reads the priority grouping field from the NVIC Interrupt Controller.

\return Priority grouping field (SCB->AIRCR [10:8] PRIGROUP field).
*/
__STATIC_INLINE uint32_t NVIC_GetPriorityGrouping(void)
{
return ((uint32_t)((SCB->AIRCR & SCB_AIRCR_PRIGROUP_Msk) >>
SCB_AIRCR_PRIGROUP_Pos));
}
The Arm M3 MCU library has a NVIC file and NVIC is integrated with some other files in M4.

Robert Thomas
on **October 20, 2015 at 8:33 am** said:

Okay, I found this in core_cm4.h. but I can't find the default value for 2 and 2. I think what I am reading is, if there is a conflict then the lowest will be taken, which would be 0. This sight is the only one that has given me anything close to why we don't have to include the NVIC_PriorityGroupConfig() function. This site says it defaults to NVIC_PriorityGroup_2. You help to find this will greatly help my OCD so I can go on. LOL

The following section of code is from core_cm4.h lines 1412 to 1429.

The function sets the priority grouping field using the required unlock sequence.
The parameter PriorityGroup is assigned to the field SCB->AIRCR [10:8] PRIGROUP field.
Only values from 0..7 are used.
In case of a conflict between priority grouping and available
priority bits (__NVIC_PRIO_BITS), the smallest possible priority group is set.

\param [in] PriorityGroup Priority grouping field.
*/
__STATIC_INLINE void NVIC_SetPriorityGrouping(uint32_t PriorityGroup)
{
uint32_t reg_value;
uint32_t PriorityGroupTmp = (PriorityGroup & (uint32_t)0x07); /* only values 0..7 are used */

reg_value = SCB->AIRCR; /* read old register configuration */
reg_value &= ~(SCB_AIRCR_VECTKEY_Msk | SCB_AIRCR_PRIGROUP_Msk); /* clear bits to change */
reg_value = (reg_value | ((uint32_t)0x5FA << SCB_AIRCR_VECTKEY_Pos) |
(PriorityGroupTmp <AIRCR = reg_value;
}

If you could point me into the direction of the default value, I would be very grateful or show me why it doesn't have to be included.

Thank you

Robert Thomas
on **October 20, 2015 at 1:25 pm** said:

FYI: I have been searching the STM32F4 libraries for the default NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2). The only place I can find it is in a usart.c or .h file. I can not fine a default other than than one.

Green
on **October 23, 2015 at 8:02 pm** said:

seems the ST library after 2014 had the default NVIC group set to 4, which leaves no priorities for sub-priorities. So I guess this is more like a library behaviors.

Robert Thomas
on **October 24, 2015 at 8:30 pm** said:

See library core_cm3.h roughly line 392.

The line of code appears to put 7 (0b111) at SCB_AIRCR_PRIGROUP.

After tons of research, this is the only place I see a default. What you say is a second witness to what I see is right. The article above said 2 was default. However, I don't think he went as deep as I did to find this. I think perhaps he saw and used the usart.c library that has a usart1 and usart2 if statement where it has 2 defaulted. But truthfully the usart.c library is more of an initialization function than a library. So I don't know where 2 came from. Hopefully the author will explain. Are you the author Mr. Green? There is no disrespect intended. Just a bad case of OCD.

```
#define SCB_AIRCR_PRIGROUP_Pos 8
/*!< SCB AIRCR: PRIGROUP Position */

#define SCB_AIRCR_PRIGROUP_Msk (7UL <<
SCB_AIRCR_PRIGROUP_Pos) /*!< SCB AIRCR: PRIGROUP Mask */
```

Thanks!!

Kyu Chong
on **December 1, 2015 at 2:02 pm** said:

Below is from core_cm4.h (stm32f4-discovery)
NVIC_SetPriority (SysTick_IRQn, (1<<__NVIC_PRIO_BITS) – 1);
I checked the value for 1<<__NVIC_PRIO_BITS) – 1 and it is 15.

Does this make sense?

Your article states "NVIC_SetPriority(SysTick_IRQn, 0);//set systick interrupt priority, 0 is the highest for all".

Green
on **December 1, 2015 at 3:04 pm** said:

-1 is not really allowed for user. it's used for some internal handlers.

Green
on **December 4, 2017 at 9:53 pm** said:

the default value is 15.
I made it as 0

jake99
on **December 4, 2017 at 5:41 am** said:

Hello,
With NVIC_InitTypeDef NVIC_InitStructure; I get unknown type name error is there any packages or #include I need to add?
Thanks

Green
on **December 4, 2017 at 9:52 pm** said:

probably the basic STM32 library needs to be included. Otherwise, you won't be able to type def any other peripherals either.

Dev
on **February 27, 2018 at 10:31 am** said:

Can we give different priorities to channels of a same timer in stm32f3 with group 4 priority configuration.

Green
on **February 27, 2018 at 1:47 pm** said:

No, because the concept doesn't apply.

Ewing Kang
on **January 27, 2019 at 2:29 am** said:

Hello, thanks for the great explaination that solved my biggest bug for the last few days.
However, I might have been a little confused, just want to be sure. So, if not explicitly changed, the entire STM32 program execution will share the same grouping setting, i.e. NVIC_PriorityGroupConfig() is a "globally effective" configuration, right?
I'm having this issue is because in the Example2 of this webpage (http://www.aimagin.com/learn/index.php?title=STM32_Interrupt_Service_Routine_Priority), it seems to me that you can actually set different IRQ with different grouping method? And one of the very poor quality code I've received actually calls NVIC_PriorityGroupConfig() each time it want to setup an IRQ, with different group setting (0~4).

Green
on **January 27, 2019 at 4:04 am** said:

I don't think it is really true. I believe it was just coincident that one interrupt still have higher overall priority than the other.
I've seen this before.
Besides, there is no point to frequent change priorities after the initial config. Who knows what is going to happen during the transition of messing with priorities, right?

Ewing Kang
on **January 27, 2019 at 6:24 am** said:

Thanks for the quick reply!
I totally agree with you. I've found the problem in this code since it's constantly changing the grouping setting during its init config for different EXTI. However, the groupConfig actually configures a single register [AIRC]. So it seems to me the grouping setting is a global thing.

If this is true, the link I posted above is giving out INCORRECT information! That is quite scarry.

p.s. Additional info: (https://community.arm.com/iot/embedded/b/embedded-blog/posts/cutting-through-the-confusion-with-arm-cortex-m-interrupt-priorities)

Jewel James
on **August 30, 2019 at 2:27 am** said:

Thanks! Great post.

Mete
on **November 29, 2019 at 1:01 am** said:

Hi Green,
If I want to use two Timer interrupts simultaneously (STM32F767) for ex TIM2_IRQHandler(){cnt1++ …} and TIM5_IRQHandler(){cnt2 ++…} as counters I have to set ;

NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);

NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;

but only cnt1 counts, cnt2 does not!
could you help?

thanks for advance,

Mete