Generating interrupt in NON-security world

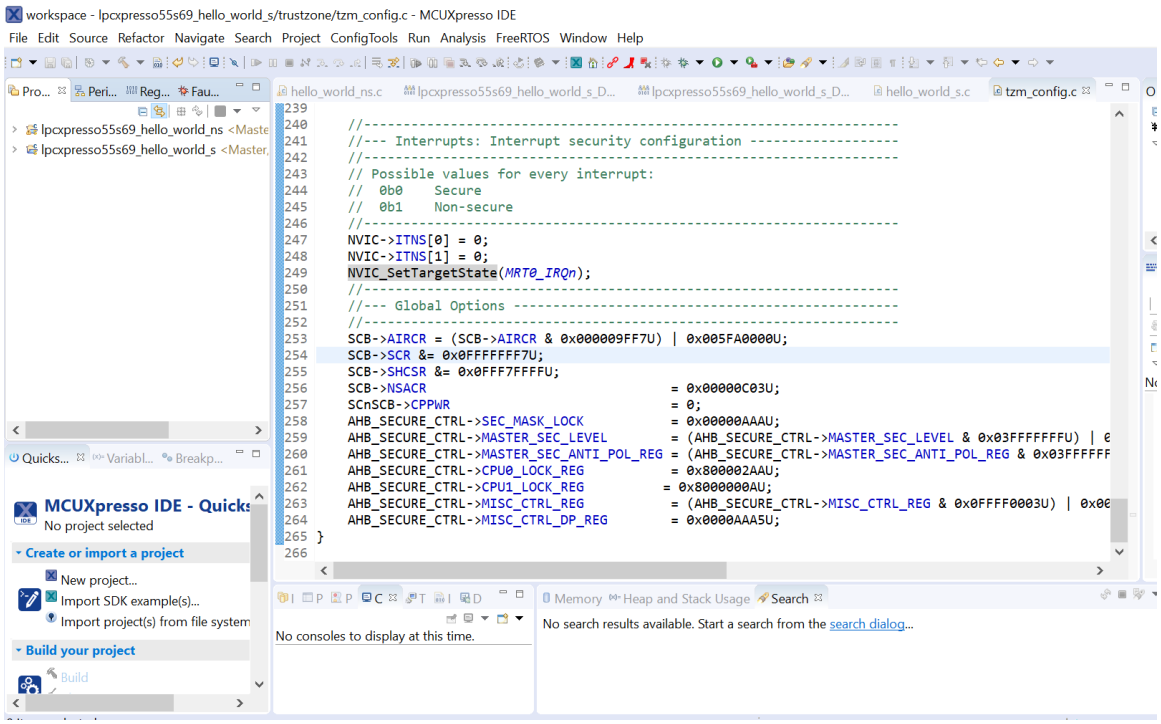
For the CM33 of LPC55S6x family, the trust zone module is integrated, the memory space and peripherals are classified as security and non-security space. In order to generate interrupt in non-security mode, the NVIC module especially the NVIC_ITNSx register must be initialized in security mode so that interrupt module can generate interrupt in non-security mode.

The example demos that MRT0 module generates interrupt in non-security mode, the NVIC module is initialized at security mode, MRT0 is initialized at non-security mode.

The project is based on MCUXpresso IDE ver11.1 tools, LPC55S69-EVK board and SDK_2.x_LPCXpresso55S69 SDK package version 2.7.1.

1) security and non-security mode introduction

The SDK package for LPC55S6x has an example:



The lpcxpresso55s69_hello_world_s is the code that the LPC55S69 runs in the security mode.

The lpcxpresso55s69_hello_world_ns is the code that the lpc55S69 runs in non-security mode.

Users can compile the two project, then download ONLY the lpcxpresso55s69_hello_world_s project, the lpcxpresso55s69_hello_world_ns is loaded to flash automatically by the lpcxpresso55s69_hello_world_s project.

The lpcxpresso55s69_hello_world_s is loaded into PROGRAM_FLASH (rx) : ORIGIN = 0x10000000, so the LPC55S69 runs in security mode after Reset.

2) NVIC-ITNSx registers introduction

For the CM33 core, there is additional NVIC-ITNSx registers,

Table 4-35 NVIC registers summary

Address	Name	Type	Required privilege	Reset value	Description
0xE000E100-0xE000E13C	NVIC_ISER0-NVIC_ISER15	RW	Privileged	0x00000000	*4.4.2 Interrupt Set Enable Registers* on page 4-305
0xE000E180- 0xE000E1BC	NVIC_ICER0-NVIC_ICER15	RW	Privileged	0x00000000	*4.4.3 Interrupt Clear Enable Registers* on page 4-306
0xE000E200- 0xE000E23C	NVIC_ISPR0-NVIC_ISPR15	RW	Privileged	0x00000000	*4.4.4 Interrupt Set Pending Registers* on page 4-307
0xE000E280- 0xE000E2BC	NVIC_ICPR0-NVIC_ICPR15	RW	Privileged	0x00000000	*4.4.5 Interrupt Clear Pending Registers* on page 4-307
0xE000E300-0xE000E33C	NVIC_IABR0-NVIC_IABR15	RW	Privileged	0x00000000	*4.4.6 Interrupt Active Bit Registers* on page 4-308
0xE000E380-0xE000E3BC	NVIC_ITNS0-NVIC_ITNS15	RW[v]	Privileged	0x00000000	*4.4.7 Interrupt Target Non-secure Registers* on page 4-308.
0xE000E400-0xE000E5DC	NVIC_IPR0-NVIC_IPR119	RW	Privileged	0x00000000	*4.4.8 Interrupt Priority Registers* on page 4-309
0xE000EF00	STIR	WO	Configurable[w]	0x00000000	*4.4.9 Software Trigger Interrupt Register* on page 4-310

Table 4-42 NVIC_ITNSn bit assignments

Bits	Name	Function
[31:0]	ITNS	Interrupt Targets Non-secure bits. For ITNS[m] in NVIC_ITNSn, this field indicates and allows modification of the target Security state for interrupt 32n+m. **0** The interrupt targets Secure state. **1** The interrupt targets Non-secure state.

The CM33 core has two interrupt vector table located at different memory space, one is for security mode, another is for non-security mode. In other words, each interrupt source has two interrupt vector, one is located in security interrupt vector table, another is located at non-security interrupt vector table.

If user wants to generate interrupt in non-security mode, user has to set the interrupt source bit in NVIC_ITNSx register in the secure project. In this way, the security project can control whether the interrupt source is allowed or not allowed in non-security project.

Generally, the code to initialize the NVIC_ITNSx register is located in the tzm_config.c to initialize the trust zone in the security project.

```
NVIC->ITNS[0] = 0;
NVIC->ITNS[1] = 0;
NVIC_SetTargetState(MRT0_IRQn);
```

3) MRT0 and NVIC code

3.1 The code required to be located at the lpcxpresso55s69_hello_world_s project:

The NVIC module initialization must be located at the security side so

that the security side can determine what resource is open to non-security side. User can use NVIC module or NVIC_NS module, either of them is okay, because security can access non-security peripherals.

```
void MRT0_InterruptInit(void)
{

    NVIC->IPR[9]=0x00;
    NVIC->ISER[0]|=1<<9;
    NVIC->ICPR[0]|=1<<9;

}
```

Or

```
void MRT0_NS_InterruptInit(void)
{

    NVIC_NS->IPR[9]=0x00;
    NVIC_NS->ISER[0]|=1<<9;
    NVIC_NS->ICPR[0]|=1<<9;

}
```

The NVIC->ITNSx register must be initialized in security side.

```
    NVIC->ITNS[0] = 0;
    NVIC->ITNS[1] = 0;
    NVIC_SetTargetState(MRT0_IRQn);
```

3.2 The code required to be located at the lpcxpresso55s69_hello_world_ns project:

```
void Init_MRT0_NS(void)
{
    SYSCON_NS->AHBCLKCTRL.AHBCLKCTRL1|=1<<0;
    //set up PIT0
    MRT0_NS->CHANNEL[0].INTVAL=12000000;
    //repeated interrupt mode
    MRT0_NS->CHANNEL[0].CTRL=0x00;
    //enable MRT channel0
    MRT0_NS->CHANNEL[0].CTRL=1<<0;}

void MRT0_IRQHandler(void)
{
    //toggle LED
    //clear flag
    MRT0_NS->CHANNEL[0].STAT=1<<0;
    GPIO_NS->NOT[1]|=1<<7;
}
//LEDG PIO1_7 as GPIO output mode
void Init_PIO1_7_NS(void)
```

```
{
    //enable gated GPIOP1 clock
    SYSCON_NS->AHBCLKCTRL.AHBCLKCTRL0|=1<<15;
    //set the mux

    //set GPIO direction reg
    GPIO_NS->DIR[1]|=1<<7;
    //toggle the PIO1_4
    GPIO_NS->NOT[1]=1<<7;
    __asm("nop");

}
```

4. conclusion

In order to generate interrupt in non-security side, the NVIC module must be initialized in security side, which includes the initialization of the registers: ISER,ICPR,IPR and ITNS of NVIC. The module which generate interrupt can be initialized in the non-security side.