(https://www.edx.org)

UTAustinX: UT.6.01x Embedded Systems - Shape the World

KarenWest (/dashboard) ▼

Courseware (/courses/UTAustinX/UT.6.01x/1T2014/courseware) **Course Info (/courses/UTAustinX/UT.6.01x/1T2014/info)**

Discussion (/courses/UTAustinX/UT.6.01x/1T2014/discussion/forum) **Progress (/courses/UTAustinX/UT.6.01x/1T2014/progress)**

Questions (/courses/UTAustinX/UT.6.01x/1T2014/a3da417940af4ec49a9c02b3eae3460b/)

Syllabus (/courses/UTAustinX/UT.6.01x/1T2014/a827a8b3cc204927b6efaa49580170d1/)

Embedded Systems Community (/courses/UTAustinX/UT.6.01x/1T2014/e3df91316c544d3e8e21944fde3ed46c/)

Help

Vector address	Number	IRQ	ISR name in **Startup.s**	NVIC	Priority bits
0x00000038	14	-2	**`PendSV_Handler`**	**`NVIC_SYS_PRI3_R`**	23 – 21
0x0000003C	15	-1	**`SysTick_Handler`**	**`NVIC_SYS_PRI3_R`**	31 – 29
0x00000040	16	0	**`GPIOPortA_Handler`**	**`NVIC_PRI0_R`**	7 – 5
0x00000044	17	1	**`GPIOPortB_Handler`**	**`NVIC_PRI0_R`**	15 – 13
0x00000048	18	2	**`GPIOPortC_Handler`**	**`NVIC_PRI0_R`**	23 – 21
0x0000004C	19	3	**`GPIOPortD_Handler`**	**`NVIC_PRI0_R`**	31 – 29
0x00000050	20	4	**`GPIOPortE_Handler`**	**`NVIC_PRI1_R`**	7 – 5
0x00000054	21	5	**`UART0_Handler`**	**`NVIC_PRI1_R`**	15 – 13
0x00000058	22	6	**`UART1_Handler`**	**`NVIC_PRI1_R`**	23 – 21
0x0000005C	23	7	**`SSI0_Handler`**	**`NVIC_PRI1_R`**	31 – 29
0x00000060	24	8	**`I2C0_Handler`**	**`NVIC_PRI2_R`**	7 – 5
0x00000064	25	9	**`PWM0Fault_Handler`**	**`NVIC_PRI2_R`**	15 – 13
0x00000068	26	10	**`PWM0_Handler`**	**`NVIC_PRI2_R`**	23 – 21

0x0000006C	27	11	**PWM1_Handler**	**NVIC_PRI2_R**	31 – 29
0x00000070	28	12	**PWM2_Handler**	**NVIC_PRI3_R**	7 – 5
0x00000074	29	13	**Quadrature0_Handler**	**NVIC_PRI3_R**	15 – 13
0x00000078	30	14	**ADC0_Handler**	**NVIC_PRI3_R**	23 – 21
0x0000007C	31	15	**ADC1_Handler**	**NVIC_PRI3_R**	31 – 29
0x00000080	32	16	**ADC2_Handler**	**NVIC_PRI4_R**	7 – 5
0x00000084	33	17	**ADC3_Handler**	**NVIC_PRI4_R**	15 – 13
0x00000088	34	18	**WDT_Handler**	**NVIC_PRI4_R**	23 – 21
0x0000008C	35	19	**Timer0A_Handler**	**NVIC_PRI4_R**	31 – 29
0x00000090	36	20	**Timer0B_Handler**	**NVIC_PRI5_R**	7 – 5
0x00000094	37	21	**Timer1A_Handler**	**NVIC_PRI5_R**	15 – 13
0x00000098	38	22	**Timer1B_Handler**	**NVIC_PRI5_R**	23 – 21
0x0000009C	39	23	**Timer2A_Handler**	**NVIC_PRI5_R**	31 – 29
0x000000A0	40	24	**Timer2B_Handler**	**NVIC_PRI6_R**	7 – 5
0x000000A4	41	25	**Comp0_Handler**	**NVIC_PRI6_R**	15 – 13
0x000000A8	42	26	**Comp1_Handler**	**NVIC_PRI6_R**	23 – 21
0x000000AC	43	27	**Comp2_Handler**	**NVIC_PRI6_R**	31 – 29
0x000000B0	44	28	**SysCtl_Handler**	**NVIC_PRI7_R**	7 – 5
0x000000B4	45	29	**FlashCtl_Handler**	**NVIC_PRI7_R**	15 – 13

0x000000B8	46	30	**GPIOPortF_Handler**	**NVIC_PRI7_R**	23 – 21
0x000000BC	47	31	**GPIOPortG_Handler**	**NVIC_PRI7_R**	31 – 29
0x000000C0	48	32	**GPIOPortH_Handler**	**NVIC_PRI8_R**	7 – 5
0x000000C4	49	33	**UART2_Handler**	**NVIC_PRI8_R**	15 – 13
0x000000C8	50	34	**SSI1_Handler**	**NVIC_PRI8_R**	23 – 21
0x000000CC	51	35	**Timer3A_Handler**	**NVIC_PRI8_R**	31 – 29
0x000000D0	52	36	**Timer3B_Handler**	**NVIC_PRI9_R**	7 – 5
0x000000D4	53	37	**I2C1_Handler**	**NVIC_PRI9_R**	15 – 13
0x000000D8	54	38	**Quadrature1_Handler**	**NVIC_PRI9_R**	23 – 21
0x000000DC	55	39	**CAN0_Handler**	**NVIC_PRI9_R**	31 – 29
0x000000E0	56	40	**CAN1_Handler**	**NVIC_PRI10_R**	7 – 5
0x000000E4	57	41	**CAN2_Handler**	**NVIC_PRI10_R**	15 – 13
0x000000E8	58	42	**Ethernet_Handler**	**NVIC_PRI10_R**	23 – 21
0x000000EC	59	43	**Hibernate_Handler**	**NVIC_PRI10_R**	31 – 29
0x000000F0	60	44	**USB0_Handler**	**NVIC_PRI11_R**	7 – 5
0x000000F4	61	45	**PWM3_Handler**	**NVIC_PRI11_R**	15 – 13
0x000000F8	62	46	**uDMA_Handler**	**NVIC_PRI11_R**	23 – 21
0x000000FC	63	47	**uDMA_Error**	**NVIC_PRI11_R**	31 – 29

Table 12.1. Some of the interrupt vectors for the TM4C. The TM4C123 has over 100 possible interrupt sources.

To activate an interrupt source we need to set its priority and enable that source in the NVIC. This activation is in addition to the arm and enable steps. Table 12.1 lists some of the interrupt sources available on the TM4C family of microcontrollers. Interrupt numbers 0 to 15 contain the faults, software interrupt and SysTick; these interrupts will be handled differently from interrupts 16 and up.

Table 12.2 shows some of the priority registers on the NVIC. Each register contains an 8-bit priority field for four devices. On the TM4C microcontrollers, only the top three bits of the 8-bit field are used. This allows us to specify the interrupt priority level for each device from 0 to 7, with 0 being the highest priority. The interrupt number (number column in Table 12.1) is loaded into the **IPSR** register. The servicing of interrupts does not set the I bit in the **PRIMASK**, so a higher priority interrupt can suspend the execution of a lower priority ISR. If a request of equal or lower priority is generated while an ISR is being executed, that request is postponed until the ISR is completed. In particular, those devices that need prompt service should be given high priority.

Address	31 – 29	23 – 21	15 – 13	7 – 5	Name
0xE000E400	GPIO Port D	GPIO Port C	GPIO Port B	GPIO Port A	NVIC_PRI0_R
0xE000E404	SSI0, Rx Tx	UART1, Rx Tx	UART0, Rx Tx	GPIO Port E	NVIC_PRI1_R
0xE000E408	PWM Gen 1	PWM Gen 0	PWM Fault	I2C0	NVIC_PRI2_R
0xE000E40C	ADC Seq 1	ADC Seq 0	Quad Encoder	PWM Gen 2	NVIC_PRI3_R
0xE000E410	Timer 0A	Watchdog	ADC Seq 3	ADC Seq 2	NVIC_PRI4_R
0xE000E414	Timer 2A	Timer 1B	Timer 1A	Timer 0B	NVIC_PRI5_R
0xE000E418	Comp 2	Comp 1	Comp 0	Timer 2B	NVIC_PRI6_R
0xE000E41C	GPIO Port G	GPIO Port F	Flash Control	System Control	NVIC_PRI7_R
0xE000E420	Timer 3A	SSI1, Rx Tx	UART2, Rx Tx	GPIO Port H	NVIC_PRI8_R
0xE000E424	CAN0	Quad Encoder 1	I2C1	Timer 3B	NVIC_PRI9_R
0xE000E428	Hibernate	Ethernet	CAN2	CAN1	NVIC_PRI10_R
0xE000E42C	uDMA Error	uDMA Soft Tfr	PWM Gen 3	USB0	NVIC_PRI11_R
0xE000ED20	SysTick	PendSV	--	Debug	NVIC_SYS_PRI3_R

Table 12.2. The LM3S/TM4C NVIC registers. Each register is 32 bits wide. Bits not shown are zero.

There are five enable registers **NVIC_EN0_R** through **NVIC_EN4_R**. The 32 bits in register **NVIC_EN0_R** control the IRQ numbers 0 to 31 (interrupt numbers 16 – 47). In Table 12.1 we see UART0 is IRQ=5. To enable UART0 interrupts we set bit 5 in **NVIC_EN0_R**, see Table 12.3. The 32 bits in **NVIC_EN1_R** control the IRQ numbers 32 to 63 (interrupt numbers 48 –

79). In Table 12.1 we see UART2 is IRQ=33. To enable UART interrupts we set bit 1 (33-32=1) in **NVIC_EN1_R**, see Table 12.3. Not every interrupt source is available on every TM4C microcontroller, so you will need to refer to the data sheet for your microcontroller when designing I/O interfaces. Writing zeros to the **NVIC_EN0_R** through **NVIC_EN4_R** registers has no effect. To disable interrupts we write ones to the corresponding bit in the **NVIC_DIS0_R** through **NVIC_DIS4_R** register.

The SysTick interrupts do not have an enable register, so when configuring interrupts on SysTick we will skip the step of setting bits in one of the **NVIC_ENx_R**registers.

Address	31	30	29-7	6	5	4	3	2	1	0	Name
0xE000E100	G	F	…	UART1	UART0	E	D	C	B	A	NVIC_EN0_R
0xE000E104			…						UART2	H	NVIC_EN1_R

Table 12.3. Some of the TM4C NVIC interrupt enable registers. There are five such registers defining 139 interrupt enable bits.

Each interrupt has an interrupt number and an IRQ number. Let *x* be the interrupt number and let *n* be the IRQ number. The interrupt number is

$x = n+16$

The interrupt number defines the position in the vector table. The vector address is

$4*x$.

For example SysTick is interrupt number 15 and IRQ number -1. Therefore the SysTick vector is at memory location 60, which is ROM location 0x0000003C.

For those interrupts with IRQ numbers greater and or equal to 0, we can find its priority register by dividing *n* by 4. Let *m* = *n*/4 (integer divide). The priority register number will be *m*. We can find the bit field for that IRQ by looking the remainder, *p* = *n*%4, where *p* = 0, 1, 2, or 3. The three bits will be 8**p*+7, 8**p*+6, and 8**p*+5.

p	Bit Field	Interrupt
3	Bits 31:29	Interrupt [4*m*+3]
2	Bits 23:21	Interrupt [4*m*+2]
1	Bits 15:13	Interrupt [4*m*+1]
0	Bits 7:5	Interrupt [4*m*]

For example, Timer3A is interrupt number 51 and IRQ number 35. *n*=35, so *m*=35/4 = 8. *p*=35%4=3. So the Timer3A priority register is 8 (NVIC_PRI8_R), and the three priority bits are be 8*3+7, 8*3+6, and 8*3+5, which are 31, 30, and 29.

For those interrupts with IRQ numbers greater and or equal to 0, we can find its enable register by dividing *n* by 32. Let *a* = *n*/32 (integer divide). The enable register number will be *a*. We can find the bit field for that IRQ by looking the remainder, *b* = *n*%32, where *b* = 0, 1, ... or 31. The one bit will be used to enable the interrupt.

Again, Timer3A is interrupt number 51 and IRQ number 35. *n*=35, so *a*=35/32 = 1. *b*=35%32=3. So the Timer3A enable register is 1 (NVIC_EN1_R), and the enable bit is bit 3.