Hacettepe University
Computer Engineering Department
BBM434: Embedded Systems Laboratory
4th Lab Assignment

Şerafettin Berk SEVGİ
21328414

Cemal ÜNAL
21328538

20.03.2017

OBJECTIVE

The purpose of this assignment is to reimplement the 3rd laboratory experiment with using interrupts to carry out delays. Main idea is same with last lab. experiment — lighting up the LED's with specified order&duration and changing that order by pressing a button.

THEORETICAL BACKGROUND FOR THE LAB

Since the interrupts are used in the experiment, it is good to give some information about interrupts. Interrupt is the automatic transfer of software execution in response to a hardware event called trigger. A hardware event can be internal or can be external. For example, an internal event can be periodic timer, bus fault, memory fault etc. ; on the other hand, an external event can come up with an external device like switches.

PROCEDURE AND RESULTS

Two interrupt signals were used in the experiment. First one is to create delays between LED lightnings. And the second one is to check switch button is pressed or not. After choosing which port will be used in the experiment, necessary initializations were made to use interrupts.
Since PORTE was used, initializations for switch as given below:

The switch is connected to the pin PE0

`GPIO_PORTE_IS_R &= ~0x01;`	Make PE0 is edge-sensitive
`GPIO_PORTE_IBE_R &= ~0x01;`	PE0 is not both edges
`GPIO_PORTE_IEV_R |= 0x01;`	Make PE0 rising edge event
`GPIO_PORTE_ICR_R = 0x01;`	Acknowledge flag0
`GPIO_PORTE_IM_R |= 0x01;`	Arm interrupt on PE0
`NVIC_PRI1_R = (NVIC_PRI1_R&0xFFFFFF00)|0x00000020;`	Give priority 1
`NVIC_EN0_R = 0x00000010;`	Enable interrupt for PortE

These initializations were only for switch interrupts. After that initialization for interrupts must be done for SystickInit.

It can be seen from the following code fragment:

`NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x40000000;`	Give priority 2
`NVIC_ST_CTRL_R = 0x07;`	Enable SysTick with core clock and interrupts

DIR	AFSEL	PMC	IS	IBE	IEV	IME	Port mode
0	0	0000	0	0	0	1	Input, falling edge trigger, interrupt
0	0	0000	0	0	1	1	Input, rising edge trigger, interrupt
0	0	0000	0	1	-	1	Input, both edges trigger, interrupt

Figure.1 *GPIO_PORTE_IS_R, GPIO_PORTE_IBE_R, GPIO_PORTE_IEV_R and GPIO_PORTE_IM_R registers are assigned to variables to get rising edge trigger according to the Figure.1.*

Vector address	Number	IRQ	ISR name in **Startup.s**	NVIC_ _ _ _	Priority bits
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

Figure.2

In case of wondering where the priority register values come from, then Figure.2 can answer this question. Since PORTE was used in the experiment, bits between 7 and 5 are used to determine the priority. Because of priority bits has 3 bits, priority can change between the values 0 and 7. 0 has the highest priority and priority is decreasing as we move towards 7. So 7 has the lowest priority.

In this experiment switch has the priority with value 1 and SysTick has the priority with value 2. So priority of switch is higher than Systick. To shown this calculation:

Let's consider bits between 7-5 of NVIC_PRI1_R. And then to make value of these bits 1, these bits become 001 respectively. If this binary representation is adapted to hexadecimal representation, then 0x00000020 value obtained.

VIDEO OF LABORATORY ASSIGNMENT DEMO

- BBM434 - Lab 4