

**BBM 432 - Gömülü Sistemler**  
**Bahar 2016-2017**  
**Arasınay**  
**5/4/2017**  
**Süre: 90 Dakika**

---

**Name:** \_\_\_\_\_

The exam consists of 5 questions.

Grade Table (for instructor use only)

Question	Points	Score
1	10	
2	10	
3	15	
4	30	
5	20	
Total:	85	

---

1. (a) (5 points) What is the meaning of the volatile keyword in C? Explain when we may get errors when we define a variable without the volatile keyword.
- (b) (5 points) Explain what is the atomic part of a code in an embedded system.

Consider the pseudocode on the right. I2C\_ISR is an interrupt routine. I2C\_ISR reads value from an input device (I2C\_BUF) and assigns it to a global variable (Command). What is the possible shared data bug here? Suggest a solution.

```

unsigned char Command;

void main()
{
    while(1)
    {
        if (Command == 1)
        {
            /* Send data 1 */
        }
        else if (Command == 2)
        {
            /* Send data 2 */
        }
        else
        {
            /* Send data 3 */
        }
    }
}

void I2C_ISR(void)
{
    Command = I2C_BUF;
}

```

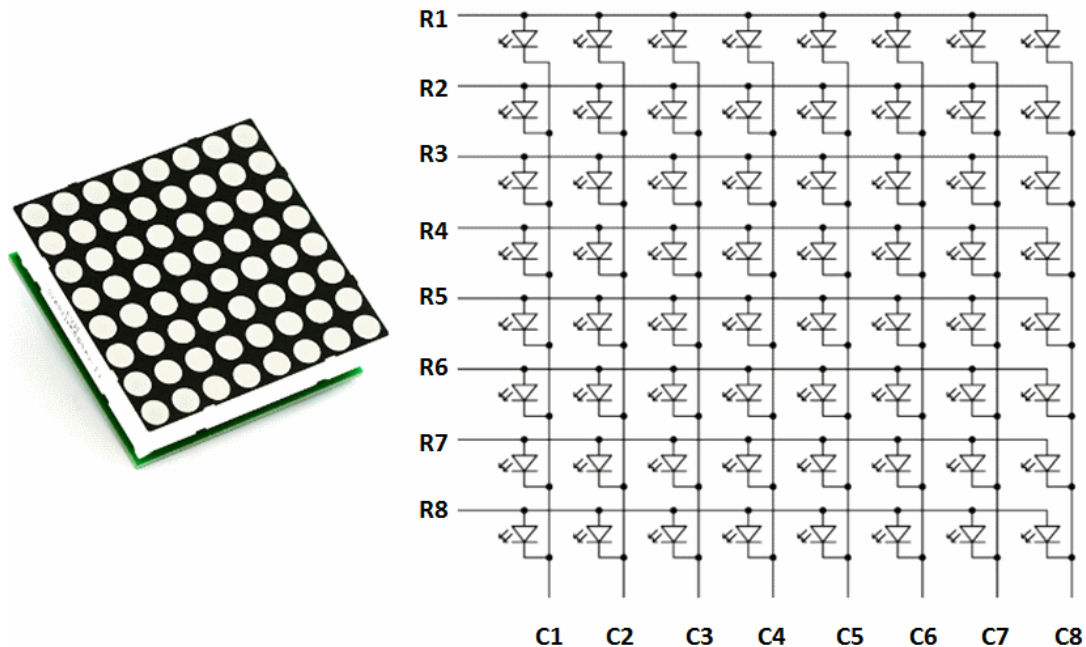
3. (15 points) You are asked to debug the following code. It is supposed to toggle the LED connected to PF2 but it does not work. In its correct operation, the LED should be on for 2 ms and off for 2 ms and so on. There are mistakes in three lines, find and rewrite them correctly.

```

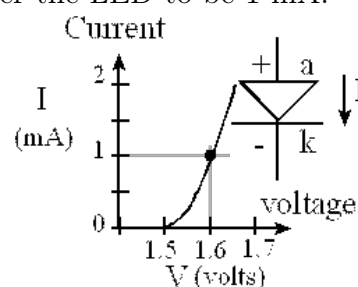
1 void SysTick_Init(unsigned long period){
2     NVIC_ST_CTRL_R = 0;
3     NVIC_ST_RELOAD_R = period-1;
4     NVIC_ST_CURRENT_R = 0;
5     NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x40000000;
6     NVIC_ST_CTRL_R = 0x05;
7 }
8 void SysTick_Handler(void){
9     GPIO_PORTF_DATA_R ^= 0x04;
10 }
11 int main(void){ // running at 16 MHz
12     SYSTCL_RCGC2_R |= 0x00000030;
13     GPIO_PORTF_DIR_R |= 0x04;
14     GPIO_PORTF_AFSEL_R &= ~0x04;
15     GPIO_PORTF_DEN_R |= 0x04;
16     GPIO_PORTF_PCTL_R = (GPIO_PORTF_PCTL_R&0xFFFFF0FF)+0x00000000;
17     GPIO_PORTF_AMSEL_R = 0;
18     SysTick_Init(16000);
19     EnableInterrupts();
20     while(1){
21         WaitForInterrupt();
22     }
23 }

```

4. Below is the circuit diagram of a 8x8 LED matrix display. You will connect row pins (R1-R8) to the Port A and column pins (C1-C8) to the Port B of the TM4C microcontroller. Remember that LEDs are directional and do not emit light when the voltage across them is reversed.



- (a) (5 points) To make this matrix work, how should you set the direction of the pins of Port A and B? Explain your reasoning and write the code that configures the direction of those pins.
- (b) (5 points) You need to use resistors in connecting the LED matrix. Why? Explain your reasoning. Indicate on the diagram where you connect the resistors.
- (c) (5 points) The LEDs have the following voltage-current relationship. Assume that the high output voltage of the pins is  $V_{OH} = 3.3$  V and the low output voltage of the pins is  $V_{OL} = 0$  V, what is the optimum value of the external resistor for the current over the LED to be 1 mA.



(a) LED curve

- (d) (5 points) Write the code block which lights up the pixel at the intersection of R3 and C3. You do not need to write the initialization routines.
- (e) (5 points) Write the code which only lights up the two pixels at the intersection of R3-C3 and R5-C5 at the same time. Is it possible? Explain your reasoning.
- (f) (5 points) Humans perceive an image as still if you display its parts frequently enough. Write the code which displays the character O on this display.

- 
5. (a) (5 points) Assume a UART link is running at 19200 baud is used to transmit the ASCII encoded text “Done”. If the link uses 8 bit characters with 1 start bit, 1 stop bit and no parity, how long will it take to transmit the whole word?
- (b) (5 points) What is the difference between asynchronous and synchronous serial communications?
- (c) (5 points) What is the role of RXFE flag in the UART function of the TM4C microcontroller.
- (d) (5 points) If the RXFE flag erroneously sticks at one, how is the operation of the UART receiver affected?

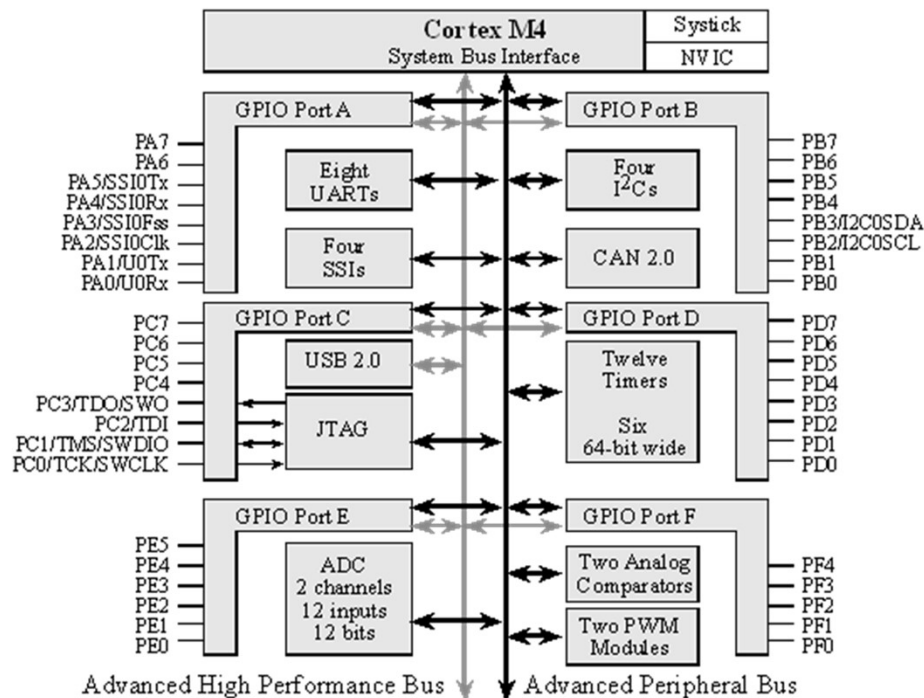
Address	31-24	23-17	16	15-3	2	1	0	Name
\$E000E010	0	0	COUNT	0	CLK_SRC	INTEN	ENABLE	NVIC_ST_CTRL_R
\$E000E014	0	24-bit RELOAD value						NVIC_ST_RELOAD_R
\$E000E018	0	24-bit CURRENT value of SysTick counter						NVIC_ST_CURRENT_R

Address	31-29	28-24	23-21	20-8	7-5	4-0	Name
\$E000ED20	TICK	0	PENDSV	0	DEBUG	0	NVIC_SYS_PRI3_R

### SysTick registers.

Table shows the SysTick registers used to create a periodic interrupt. SysTick has a 24-bit counter that decrements at the bus clock frequency. Let  $f_{BUS}$  be the frequency of the bus clock, and let  $n$  be the value of the RELOAD register. The frequency of the periodic interrupt will be  $f_{BUS}/(n+1)$ .

First, we clear the ENABLE bit to turn off SysTick during initialization. Second, we set the RELOAD register. Third, we write any value to NVIC\_ST\_CURRENT\_R to clear the counter. Lastly, we write the desired mode to the control register, NVIC\_ST\_CTRL\_R. We must set CLK\_SRC=1, because CLK\_SRC=0 external clock mode is not implemented on the LM3S/TM4C family. We set INTEN to enable interrupts. We establish the priority of the SysTick interrupts using the TICK field in the NVIC\_SYS\_PRI3\_R register. We need to set the ENABLE bit so the counter will run. When the CURRENT value counts down from 1 to 0, the COUNT flag is set. On the next clock, the CURRENT is loaded with the RELOAD value. In this way, the SysTick counter (CURRENT) is continuously decrementing. If the RELOAD value is  $n$ , then the SysTick counter operates at modulo  $n+1$  (... $n$ ,  $n-1$ ,  $n-2$  ... 1, 0,  $n$ ,  $n-1$ , ...). In other words, it rolls over every  $n+1$  counts.



Address	7	6	5	4	3	2	1	0	Name
\$400F.E108			GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	SYSCTL_RCGC2_R
\$4000.43FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTA_DATA_R
\$4000.4400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTA_DIR_R
\$4000.4420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTA_AFSEL_R
\$4000.4510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTA_PUR_R
\$4000.451C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTA_DEN_R
\$4000.4524	1	1	1	1	1	1	1	1	GPIO_PORTA_CR_R
\$4000.4528	0	0	0	0	0	0	0	0	GPIO_PORTA_AMSEL_R
\$4000.53FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTB_DATA_R
\$4000.5400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTB_DIR_R
\$4000.5420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTB_AFSEL_R
\$4000.5510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTB_PUR_R
\$4000.551C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTB_DEN_R
\$4000.5524	1	1	1	1	1	1	1	1	GPIO_PORTB_CR_R
\$4000.5528	0	0	AMSEL	AMSEL	0	0	0	0	GPIO_PORTB_AMSEL_R
\$4000.63FC	DATA	DATA	DATA	DATA	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DATA_R
\$4000.6400	DIR	DIR	DIR	DIR	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DIR_R
\$4000.6420	SEL	SEL	SEL	SEL	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_AFSEL_R
\$4000.6510	PUE	PUE	PUE	PUE	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_PUR_R
\$4000.651C	DEN	DEN	DEN	DEN	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DEN_R
\$4000.6524	1	1	1	1	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_CR_R
\$4000.6528	AMSEL	AMSEL	AMSEL	AMSEL	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_AMSEL_R
\$4000.73FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTD_DATA_R
\$4000.7400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTD_DIR_R
\$4000.7420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTD_AFSEL_R
\$4000.7510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTD_PUR_R
\$4000.751C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTD_DEN_R
\$4000.7524	CR	1	1	1	1	1	1	1	GPIO_PORTD_CR_R
\$4000.7528	0	0	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	GPIO_PORTD_AMSEL_R
\$4002.43FC			DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTE_DATA_R
\$4002.4400			DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTE_DIR_R
\$4002.4420			SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTE_AFSEL_R
\$4002.4510			PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTE_PUR_R
\$4002.451C			DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTE_DEN_R
\$4002.4524			1	1	1	1	1	1	GPIO_PORTE_CR_R
\$4002.4528			AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	GPIO_PORTE_AMSEL_R
\$4002.53FC				DATA	DATA	DATA	DATA	DATA	GPIO_PORTF_DATA_R
\$4002.5400				DIR	DIR	DIR	DIR	DIR	GPIO_PORTF_DIR_R
\$4002.5420				SEL	SEL	SEL	SEL	SEL	GPIO_PORTF_AFSEL_R
\$4002.5510				PUE	PUE	PUE	PUE	PUE	GPIO_PORTF_PUR_R
\$4002.551C				DEN	DEN	DEN	DEN	DEN	GPIO_PORTF_DEN_R
\$4002.5524				1	1	1	1	CR	GPIO_PORTF_CR_R
\$4002.5528				0	0	0	0	0	GPIO_PORTF_AMSEL_R

	31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0	
\$4000.452C	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTA_PCTL_R
\$4000.552C	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTB_PCTL_R
\$4000.652C	PMC7	PMC6	PMC5	PMC4	0x1	0x1	0x1	0x1	GPIO_PORTC_PCTL_R
\$4000.752C	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTD_PCTL_R
\$4002.452C			PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTE_PCTL_R
\$4002.552C				PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTF_PCTL_R
\$4000.6520	LOCK (write 0x4C4F434B to unlock, other locks) (reads 1 if locked, 0 if unlocked)								GPIO_PORTC_LOCK_R
\$4000.7520	LOCK (write 0x4C4F434B to unlock, other locks) (reads 1 if locked, 0 if unlocked)								GPIO_PORTD_LOCK_R
\$4002.5520	LOCK (write 0x4C4F434B to unlock, other locks) (reads 1 if locked, 0 if unlocked)								GPIO_PORTF_LOCK_R

Table 2.1. Some TM4C123/LM4F120 parallel ports. Each register is 32 bits wide. For PMCx bits, see Table 2.2. JTAG means do not use these pins and do not change any of these bits.