# IE3-CO Lab Session #2: Preparation sheet

|  |  |
| --- | --- |
| Surname, First Name | Group: |
| Ali Emami | 3 |

This preparation sheet must be filled and uploaded to EMIL by every student before the lab session. Cooperation within the lab team (of 2-3 students) is possible (same solution). However, no cooperation across lab teams.

## Question 1:

Give the 8 x n-matrix and the corresponding hexadecimal value which you want to display on the LED pendulum.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LED7 | ◼ |  |  |  |  |  | ◼ |  |  | ◼ |  |  | ◼ |  | ◼ |  |  |  |  |  | ◼ |  |  |  |  |  |  |  |  |
| LED6 | ◼ | ◼ |  |  |  | ◼ | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ | ◼ |  |  |  |  | ◼ |  |  |  |  |  |  |  |  |
| LED5 | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  |  |  | ◼ |  |  |  |  |  |  |  |  |
| LED4 | ◼ |  |  | ◼ |  |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  |  | ◼ |  |  | ◼ |  |  |  |  |  |  |  |  |
| LED3 | ◼ |  |  |  |  |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  |  |  | ◼ |  | ◼ |  |  |  |  |  |  |  |  |
| LED2 | ◼ |  |  |  |  |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  | ◼ |  |  |  |  | ◼ | ◼ |  |  |  |  |  |  |  |  |
| LED1 | ◼ |  |  |  |  |  | ◼ |  |  | ◼ |  |  | ◼ |  | ◼ |  |  |  |  |  | ◼ |  |  |  |  |  |  |  |  |
| LED0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| HEX CODE | 0xFE | 0x40 | 0x20 | 0x10 | 0x20 | 0x40 | 0xFE | 0x00 | 0x7C | 0x82 | 0x7C | 0x00 | 0xFE | 0x00 | 0xFE | 0x40 | 0x20 | 0x10 | 0x08 | 0x04 | 0xFE |  |  |  |  |  |  |  |  |  |

## Question 2:

In order to realize a stable pattern on the LED pendulum, the character string has to be aligned (synchronized) to the turning points indicated by an edge of the -signal.

Give C code that detects the left and the right turning point of the pendulum. Assume that the -signal is connected to PD(0) and that the PORT D has been correctly configured (clock activated, port enabled and directions set).

|  |
| --- |
| void Detect\_Pendulum\_turning\_point(void)  {  while(1)  {  while (E!=1)  {    }    printf("It is on the Left Margin\n");  while (E=1)  {  }  printf("It is on the right Margin\n");    }  } |

## Question 3:

Develop a function **timerConfig(void)** that configures TIMER0A as a 16 bit periodic timer that can represent times up to 10 ms (i.e. does not overflow within 10ms). Give the C-Code:

|  |
| --- |
| void timerConfig(void)  {  int waitcycle=0;    SYSCTL\_RCGC1\_R |= (1<<16);  waitcycle++; //Wait  TIMER0\_CTL\_R &= ~0x0001; //Timer 0 disabled  TIMER0\_CFG\_R = 0x04; // 2 x 16-bit mode  TIMER0\_TAMR\_R = 0x22; // periodic mode + match enable  TIMER0\_TAPR\_R = 30-1; // prescaler PR= ceil(16M/2^16\*0.12)-1  } |

## Question 4:

Develop a function **timerWait(unsigned short usec)** that realizes a delay by Timer 0A and uses the configuration of timerConfig(). The function sets the interval load value, enables the timer, waits for the time-out of the timer, clears the interrupt flags and disables the timer. usec is the value in µs after after which the timer reaches time-out.

|  |
| --- |
| void timerWait(unsigned short usec)  {  TIMER0\_TAILR\_R= (usec\*16)/29; // load value  TIMER0\_CTL\_R |= 0x0001; // Enabling the timer  while((TIMER0\_RIS\_R & (1<<0))==0); //wait until the flag set  TIMER0\_ICR\_R |= (1<<0); //clear the interrupt flag  TIMER0\_CTL\_R &= ~0x0001; //Disable the Timer  **}** |