# IE3-CO Lab Session #2: Preparation sheet

|  |  |
| --- | --- |
| Surname, First Name | Group: |
| Krishnasamy Punitha | Group 2 |

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 | 0xFF | 0x00 | 0xFF | 0x89 | 0x89 | 0x00 | 0x89 | 0x89 | 0xFF | 0x00 | 0x08 | 0x08 | 0x08 | 0x00 | 0xFF | 0x81 | 0x81 | 0x00 | 0xFF | 0x81 | 0x81 | 0xFF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 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).

|  |
| --- |
| checkTurningPoint(void)  {  // Keeping in mind the above mentioned assumptions  while (GPIO\_PORTD\_DATA\_R == 0)  printf("R to L");  while (GPIO\_PORTD\_DATA\_R != 0)  printf("L to R");  } |

## 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:

|  |
| --- |
| timerConfig(void)  {  SYSCTL\_RCGC1\_R |= (1<<16);  TIMER0\_CTL\_R &= ~0x0001; // disable TIMER 0  TIMER0\_CFG\_R = 0x04; // 2 x 16 bit mode  TIMER0\_TAMR\_R = 0x22; // periodic and match enable  TIMER0\_TAPR\_R = /\* prescale value for 10 ms \*/;  } |

## 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.

|  |
| --- |
| timerWait(unsigned short usec)  {  while(1)  {  timerConfig();  TIMER0\_TAILR\_R= /\* load value \*/;  TIMER0\_CTL\_R |= 0x0001; // enable timer 0  for(int i=0;i<usec;i++)  {  ; // empty for loop for delay  }  TIMER0\_ICR\_R |= (1<<4); // set to 1, clear interupt flag  }  } |