



4th. April, 2019

Course Code:
CSE345/347/MCT-Elective

Time : 1:00 Hr

Embedded System Design; Mid Term

The Exam Consists of 2 Questions in 4 Pages

Total Marks: 25% Marks

1/4

Student Name:

Course Code:

Question (1):

The ADXL345 is a MEMS 3-axes accelerometer from Analog Devices Co. It puts out three analog voltages which are linear functions of accelerations along X, Y and Z axes. The device powers on 3V putting out 1.5V for 0g (0 m/s²) and have a slope of 174mV/g. The sensor saturates at $\pm 10g$ (for $g = 9.8 \text{ m/s}^2$).

- 1- Select a suitable sequencer for the Accelerometer values readings and specify the steps/channels to use for each sensor in the sequencer. (only write the sequencer number and which channels assigned to which step in the sequencer)

- 2- Write the suitable initialization function ADC_init(void) – using Tivaware APIs, to configure the ADC as follows:

`void ADC_init(void){`

- a. //Enable the clock for the Port to which the sensors are connected.

- b. //Enable the clock for the ADC module.

- c. //Disable the sequencer used.

- d. //Configure the sequencer for Processor trigger.

- e. //Configure the steps of the sequencer.

- f. //Enable the sequencer.

`}`

Important Labels:

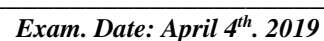
SYSCTL_PERIPH_GPIOx
 x: indicates the Port (A,B,C..)
SYSCTL_PERIPH_ADC0
SYSCTL_PERIPH_ADC1
ADC_REF_EXT_3V
ADC0_BASE
ADC1_BASE
TRIGGER_PROCESSOR
ADC_CTL_CHx
 x: indicates the Channel no.
 (0,1,2..)

Functions names:

`ADCAProcessorTrigger(.....)`
`ADCReferenceSet(.....)`
`ADCSequenceConfigure(.....)`
`ADCSequenceDataGet(.....)`
`ADCSequenceDisable(.....)`
`ADCSequenceEnable(.....)`
`ADCSequenceStepConfigure(..)`

Time : 1:00 Hr

2/4

[illegible]



4th. April, 2019

Course Code:
CSE345/347/MCT-Elective

Time : 1:00 Hr

Embedded System Design; Mid Term

The Exam Consists of 2 Questions in 4 Pages

Total Marks: 25% Marks

3/4

Question (2):

For the following Cortex M4 FreeRTOS code snippet, assume slice time is 1ms. For the same hardware connected in Question (1). The sender task will Trigger the conversion and send the data to the receiver task. Don't use Global variables array update in order to avoid loss of data.

```

74 int main(void)
75 {
76     ConfigureUART();
77     ADC_init();
78     xQueue = xQueueCreate( 6, sizeof( int32_t ) );
79     if( xQueue != NULL )
80     {
81         xTaskCreate( vSenderTask, "Sender2", 256, NULL, 1, NULL );
82         xTaskCreate( vReceiverTask, "Receiver", 256, NULL, 2, NULL );
83         vTaskStartScheduler();
84     }
85     else
86     {
87         /* The queue could not be created. */
88     }
89     for( ;; );
90 }
91 /*-----*/

```

For the sender and receiver tasks handle accelerometer data between sender and receiver tasks with periodicity of 10ms between each 2 updates. *The receiver task prints the readings by UARTprintf, indicating **which axis is being printed**.*

Your Answer:

```

static void vSenderTask( void *pvParameters )
{

```

```

    for( ;; )
    {

```

```


    }

```

```

}

```

AIN SHAMS UNIVERSITY FACULTY OF ENGINEERING Mechatronics Engineering Department/CHEP		
4th. April, 2019	Course Code: CSE345/347/MCT-Elective	Time : 1:00 Hr
Embedded System Design; Mid Term		
The Exam Consists of 2 Questions in 4 Pages	Total Marks: 25% Marks	4/4

```

static void vReceiverTask( void *pvParameters )
{

    for( ;; )
    {

    }

}

```