Jonathan W. Valvano   First:_____ Last:_____
April 21, 2017, 10:00-10:50am. Open book, open notes, calculator (no laptops, phones, devices with screens larger than a TI-89 calculator, devices with wireless communication). You have 50 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting***.

(12) Question 1. For each type of voltage regulator, choose the best description of that device. Let V_{in} I_{in} be the input voltage and current. Let V_{out} I_{out} be the output voltage and current. In each case you may assume the output, V_{out}, is constant. Put one letter, A – H, into each box.

(4) Part a) LM2937 linear regulator

(4) Part b) Buck regulator

(4) Part c) Boost regulator

A) The device creates an output current, I_{out}, that is constant.
B) The device creates an output voltage that is very low noise, and a maximum I_{out} of less than 1 mA.
C) The input current is approximately equal to the output current $I_{in} \approx I_{out}$.
D) The device converts DC to AC, uses a transformer to increase the voltage, then converts AC to DC, so the V_{out} is a constant.
E) The device uses a switching network, a diode, and an inductor to increase the voltage ($V_{in} > V_{out}$). It is very power efficient, $V_{in}*I_{in} \approx V_{out}*I_{out}$.
F) The device converts DC to AC, uses a transformer to decrease the voltage, then converts AC to DC, so the V_{out} is a constant.
G) The device uses a switching network, a diode, and an inductor to decrease the voltage ($V_{in} > V_{out}$). It is very power efficient, $V_{in}*I_{in} \approx V_{out}*I_{out}$.
H) The device creates an output voltage that is a linear function of the input voltage ($V_{out} = mV_{in}$).

(16) Question 2. Let *I* be an *m*-bit unsigned integer, and *J* be an *n*-bit unsigned integer, where $n \leq m$. For each operation, specify the number of bits needed to store the integer *K*, which occurs as a result of the integer operation.

(4) Part a) Addition: *K=I+J*

(4) Part b) Multiplication: *K=I*J*

(4) Part c) Division: *K=I*/1000

(4) Part d) Right shift: *K=I>>4*

(12) Question 3. This interface uses SPI. Assume we are using Freescale mode with the TM4C123 as master. The following is a timing diagram of the external device, which can be used to input 10 bits into the microcontroller.
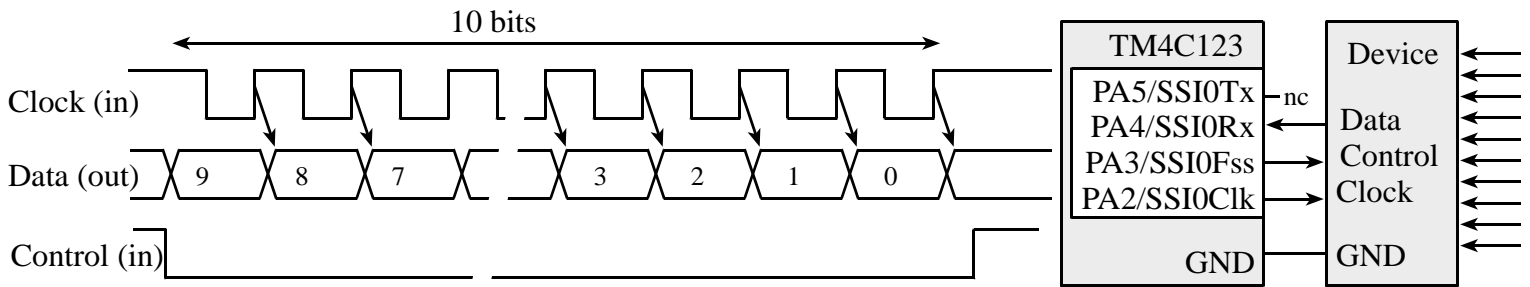

(4) Part a) What value should the software write to **DSS** during initialization?

(4) Part b) What value should the software write to **SPO** during initialization?

**(4) Part c)** What value should the software write to **SPH** during initialization?

(10) Question 4. The following spectrum was made at the input of an 11-bit, 0 to 5V ADC. The microcontroller samples this signal at 1 kHz.

What is the resolution of this ADC in dB_{FS}? Show your work
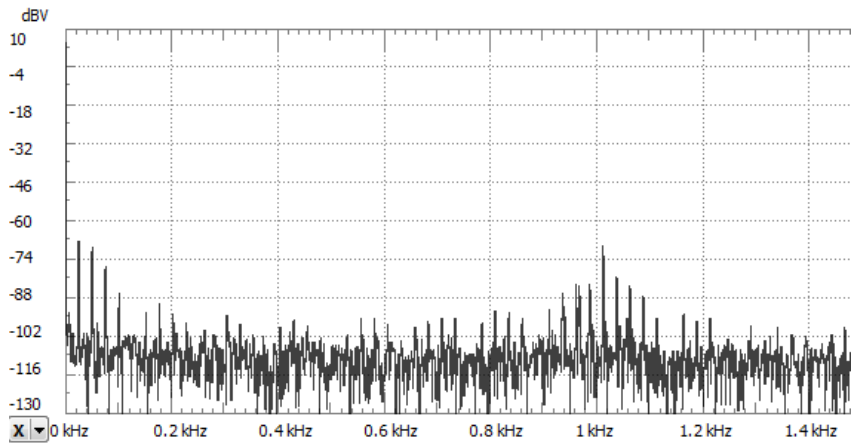
Is there aliasing, yes or no?

(15) Question 5. You will write a C function that approximates the sin function using fixed-point math. No floating point is allowed. Assume the input is an integer x in degrees, from 0 to 90. The output will be $y=1000*\sin(x)$. Basically, you will implement the cubic equation $y = -0.0006x^3 - 0.0213x^2 + 17.921x - 2.12$. The prototype is given and you must write the body of the function. For example, if x is 90, return 1000.
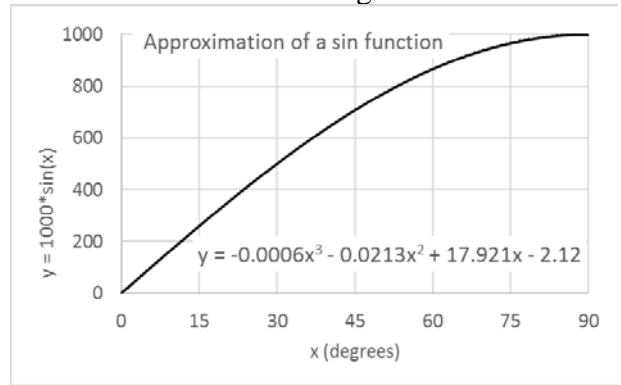


```
int32_t sin(int32_t x){ // assume x is an integer between 0 and 90
```

(10) Question 6. Your job is to interface this transducer to the ADC on the TM4C123. The transducer output is a differential voltage called $V_{in} = (V_+ - V_-)$. V_- is about 1 V, and V_+ varies such that V_{in} varies from 0 to +0.10V. You goal is to make the output voltage vary from 0 to 3 V. The output, V_{out}, is connected to the microcontroller ADC. R1 and R2 are already chosen such that the analog reference is 1.65V. You may use any chips shown in the book or presented in class. Show your work and label all chip numbers and resistor values. You do not have to show pin numbers. You do not need to add a low pass filter.
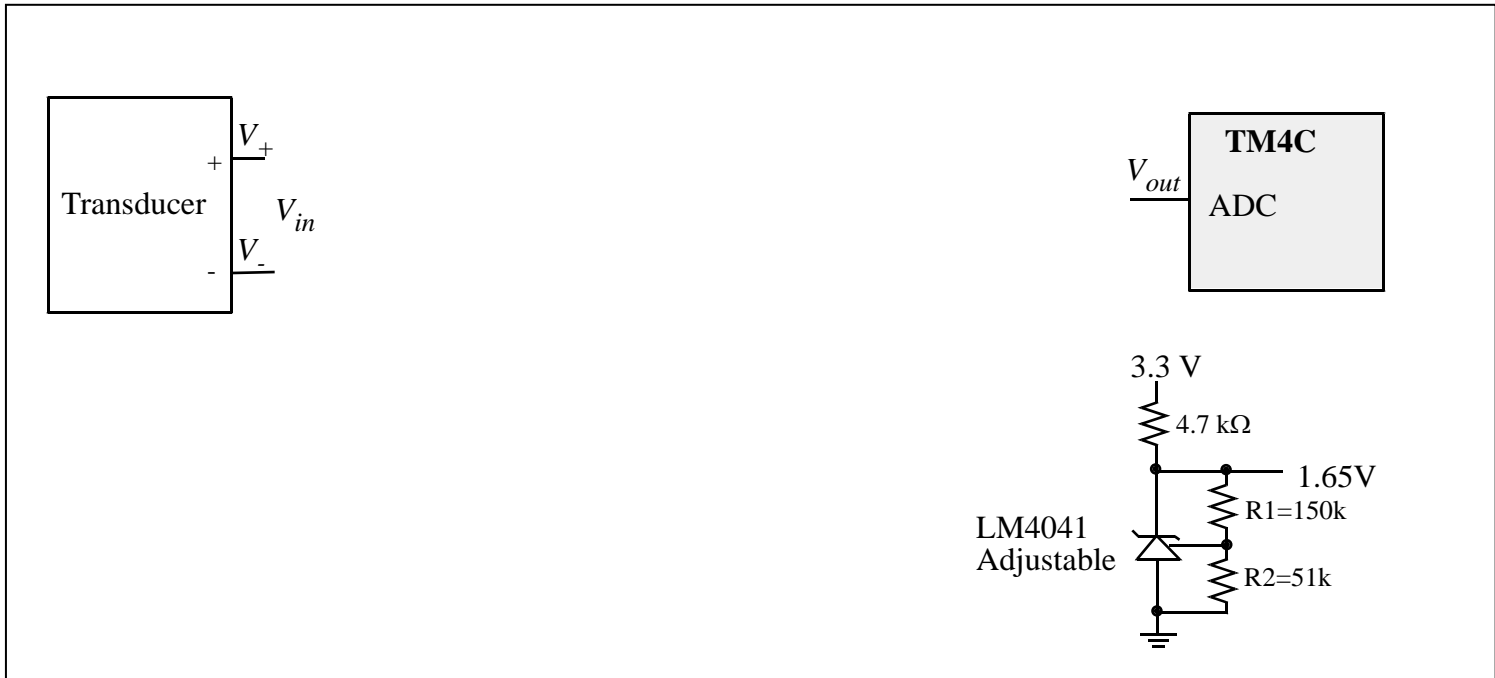
(25) Question 7. Port B bits 2, 1, and 0 are inputs from digital Hall-effect sensors. These three inputs cycle through the values 1,0,4,6,7,3 over and over. The time between input changes varies, but the time is always greater than 1ms. There are three outputs on Port D bits 2, 1, 0. The values of the outputs can be high, low, or hiZ. Your software should use edge-triggered interrupts to implement this input/output relationship.

Step	PortB	PB2	PB1	PB0	PD2	PD1	PD0
1	1	0	0	1	1	hiZ	0
2	0	0	0	0	1	0	hiZ
3	4	1	0	0	hiZ	0	1
4	6	1	1	0	0	hiZ	1
5	7	1	1	1	0	1	hiZ
6	3	0	1	1	hiZ	1	0
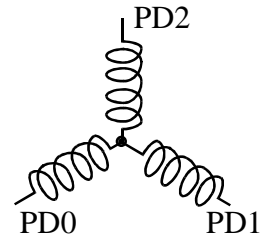
We can re-sort this function table to visualize the input/output pattern (0* means doesn't matter)

Port B Input	Port D Direction	Port D Output
0	6	4
1	5	4
2	0*	0*
3	3	2
4	3	1
5	0*	0*
6	5	1
7	6	2

```
const uint32_t Dir[8]={6,5,0,3,3,0,5,6};
const uint32_t Dat[8]={4,4,0,2,1,0,1,2};
```

(15) Part a) Show the initialization code that configures Ports B and D. It should arm and enable the appropriate edge-triggered interrupts. Do not worry about interrupt priority. You may add private global variables. Assume you have access to all of Ports B and D, so you do not need to be friendly. No backward jumps are allowed in the ISR.

```
void BushlessDCmotor_Init(void){
  SYSCTL_RCGCGPIO_R   [            ]

  [            ]          // initialization of variables

  GPIO_PORTD_DEN_R = 7;
  GPIO_PORTD_DIR_R = 6;
  GPIO_PORTD_DATA_R = 4;

  GPIO_PORTB_DIR_R    [            ]

  GPIO_PORTB_DEN_R    [            ]

  GPIO_PORTB_IS_R     [            ]

  GPIO_PORTB_IBE_R    [            ]
```

GPIO_PORTB_IEV_R

GPIO_PORTB_IM_R

NVIC_PRI0_R = NVIC_PRI0_R&0xFFFF00FF; // priority 0

NVIC_EN0_R

EnableInterrupts(); }

(10) Part b) Show the edge-triggered interrupt service routine. No backward jumps are allowed in the ISR. You do not need to show the main program. Use the two arrays, **Dat[]** and **Dir[]**, to update Port D.
void GPIOPortB_Handler(void){