First:_____ Last:_____ Jonathan W. Valvano

Closed book, 2 pieces of paper crib notes, 8.5 by 11 in, 2-sided. No computer, no calculator.

(5) Question 1. Four voltage samples are collected at 1 kHz. Decimal fixed-point resolution is 1mV.

```
int32_t a[4]; // four samples with units of mV
```

The goal is to calculate DFT for the k=1 term, A_1.

Part a) Derive two equations, one for the real part and another for imaginary part. Do not leave it in summation form, write both equations as functions of `a[0]` `a[1]` `a[2]` and `a[3]`.

Part b) What is the equivalent frequency of the k=1 term?

(5) Question 2. Design a voltmeter that uses a 0 to 2V 12-bit ADC. The voltage will be represented in 8 bit binary fixed-point with resolution of 1/64 volts. The given function **ADC_In()** returns a 12-bit sample.

```
uint32_t ADC_In();
```

Write a C function that calls **ADC_In()**, converts the sample into fixed point, and returns the fixed-point number. Clearly specify the input and output parameters of your C function.

(5) Question 3. *T* is the measured temperature, *Tstar* is the desired temperature, and *U* is the actuator output. The MSP430 is a 16-bit microcontroller without multiply or divide instructions. This controller code works, but it runs too slowly on the MSP430.

```
int16_t U,T,Tstar; // do not change these definitions
    U = 0.75*(X-Xstar);
```

Rewrite the C code for this controller without floating point, without divide, and without multiply.	What type of controller is this?

(5) Question 4. Match each definition with the best debugging term. Put one letter in each box

Debugging that has no effect on the system being debugged…… [ ]

Fixing the input parameters so the code can be run over and over, yielding the same outputs.. [ ]

Store data into arrays and look at the arrays later…… [ ]

Debugging that can measure CPU utilization for an ISR…… [ ]

Method to measure system current as a function of time…… [ ]

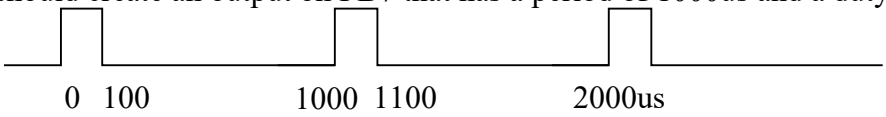
A) Atomic	E) Acceleration	I) Probability mass function
B) Nonintrusive	F) Triple toggle and scope	J) Toggle LED once
C) Intrusive	G) Stabilization	K) Black box
D) Current sense amplifier	H) Dump	L) Voltmeter

(5) Question 5. A buck-boost regular converts an 8-V battery to a 3-V supply. The regulator frequency is 2.4 MHz. The bus clock is 80 MHz. All circuits in the system run off +3V. The temperature is 25 °C. The regulator's efficiency is 75%. You measure 1A current between the battery and the regulator. How much current is flowing through the system? Show your work.

(10) Question 6. You will implement **PWM** on PB7 using periodic SysTick interrupts. There are two global variables that are initialized, but their values can be changed by other code while your system is running. You may assume the values of **High** and **Low** range from 100 to 2000.

```
uint32_t High=100; // number of usec PB7 should be high
uint32_t Low=900;  // number of usec PB7 should be low
```

The above values should create an output on PB7 that has a period of 1000us and a duty cycle of 10%.



PB7 is initialized as an output, writing 0x80 to **PB7** sets it high, and writing 0 clears it low.

```
#define PB7 (*((volatile uint32_t *)0x40005200))
```

The bus clock is 80MHz. The following code initializes SysTick and cannot be changed.

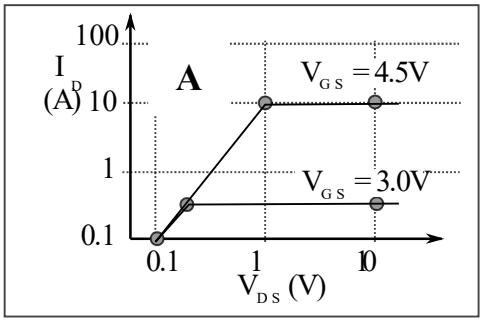
```
void SysTick_Init(void){
  NVIC_ST_CTRL_R = 0;
  NVIC_ST_RELOAD_R = 8000; // 100us = 8000*12.5ns
  NVIC_ST_CURRENT_R = 0;
  NVIC_SYS_PRI3_R |= 0xE0000000; // priority 7
  NVIC_ST_CTRL_R = 0x07;
  PB7 = 0x80;
  EnableInterrupts();
}
```

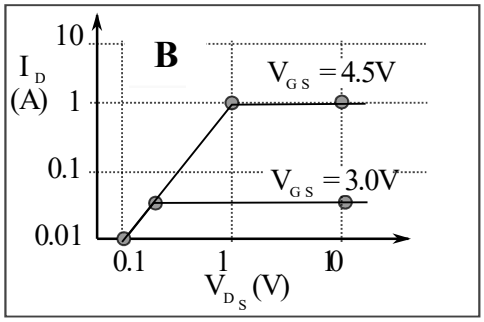
Show the SysTick ISR code

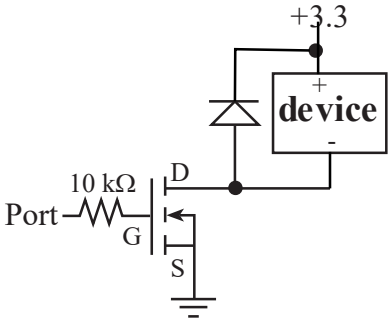
```
void SysTick_Handler(void){
```

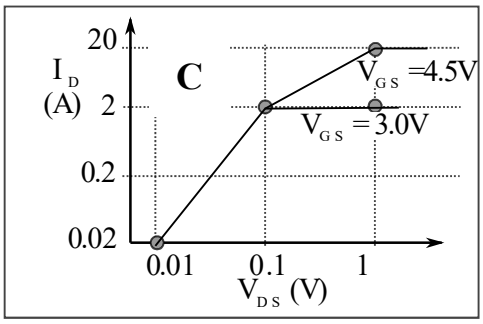
(5) Question 7. We wish to interface a *device* using the same circuit as the speaker in Lab 3 (replacing speaker with *device*). The *device* requires a maximum current of 1 A. We use a 3.3V supply that can source 1A. V_{OH} of the microcontroller is 3.0V. We want V_{DS} to be strictly less than 0.1V. The I_{OL} and I_{OH} of the microcontroller are 8mA. Below are I_D versus V_{DS} curves for four different MOSFETs. Which MOSFET would you choose? Justify your choice by 1) choosing which V_{GS} to consider (3.0 or 4.5V) and 2) **labeling the operating point on the curve** for your choice.

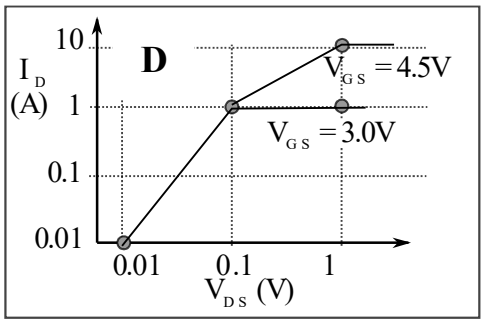
Select A, B, C, or D



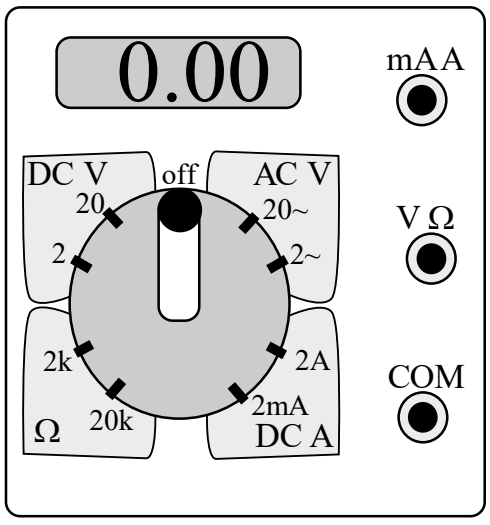


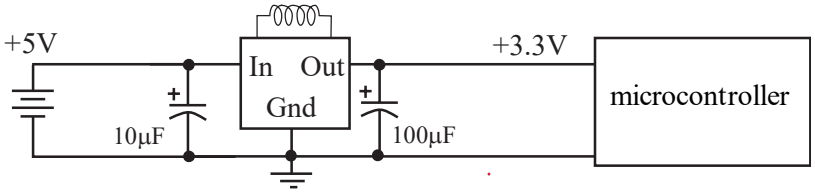






(5) Question 8. Show how to measure microcontroller current with a multimeter. The numbers on the meter specify maximum value for that mode. A) Circle one of the numbers on the dial specifying the correct mode to use. B) Cut wire(s) with an **X** if needed. C) Add wires between the circuit and the multimeter to perform the measurement.





(10) Question 9. Measure the **frequency** on PB7 input using edge-triggered and SysTick periodic interrupts. The edge-triggered ISR is invoked on each rising edge of PB7. The edge-triggered interrupt occurs on every rising edge of PB7. The SysTick interrupt occurs every 1 ms. The desired frequency resolution is 1 kHz. ISRs have equal priorities.

```
uint32_t Count;
uint32_t Freq;   // in kHz
#define PB7 (*((volatile uint32_t *)0x40005200))
```

```
void GPIOPortB_Handler(void){
  GPIO_PORTB_ICR_R = 0x80;




}
```

```
void SysTick_Handler(void){

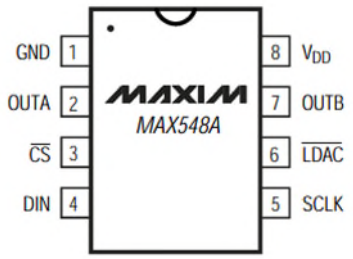


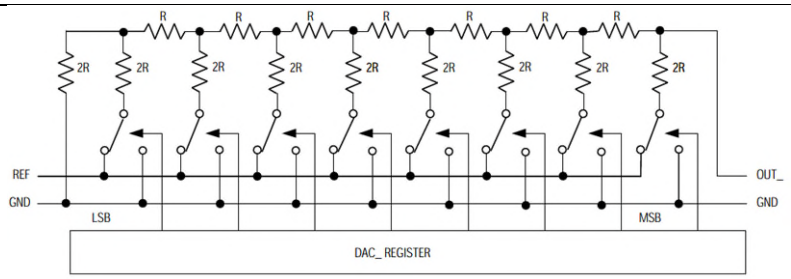

}
```

(5) Question 10. Design an analog circuit to interface the transducer to the ADC. The range of V_{in} is -0.1 to +0.1V. The ADC range is $0<V_{out}<2.0$V. Show your work and label all resistor values. You do not have to show pin numbers. You do not need to add a low pass filter.



(5) Question 11. Write a timing equation for **data required interval** for the MAX548A DAC at 5.5V.

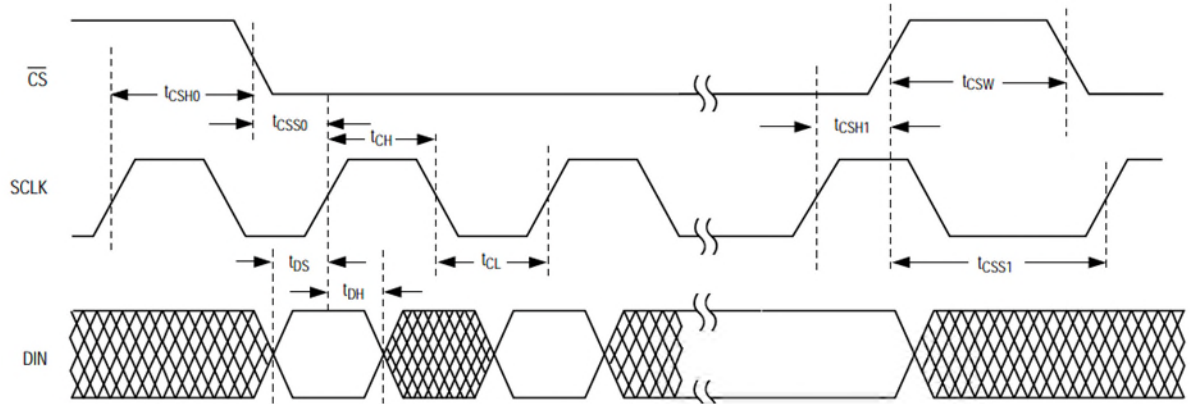




TIMING CHARACTERISTICS

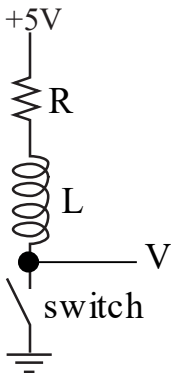
(V_{DD} = +2.5V to +5.5V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Digital inputs switching from 0V to V_{DD}.) (Figure 3) (Note 4)

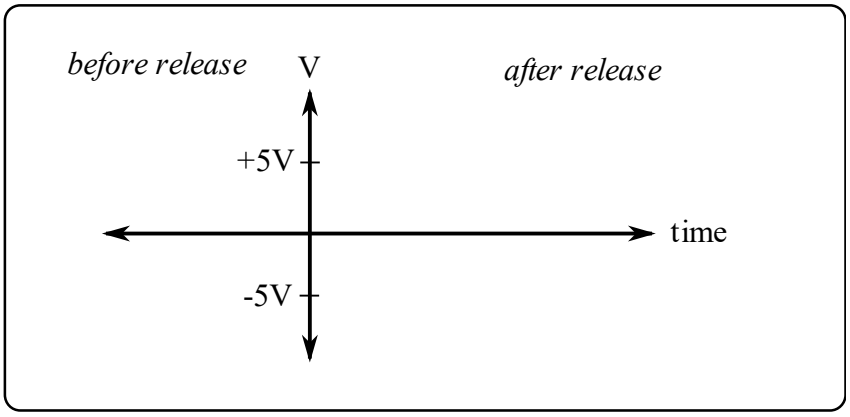
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCLK Pulse Width High	t_{CH}		40			ns
SCLK Pulse Width Low	t_{CL}		40			ns
DIN to SCLK High Setup	t_{DS}		30			ns
DIN to SCLK High Hold	t_{DH}	V_{DD} = 2.5V	0			ns
		V_{DD} = 5.5V	10			
$\overline{CS}$ Low to SCLK High Setup	t_{CSS0}		30			ns
$\overline{CS}$ High to SCLK High Setup	t_{CSS1}		30			ns
SCLK High to CS Low Hold	t_{CSH0}		10			ns
Delay, SCLK High to $\overline{CS}$ High	t_{CSH1}	V_{DD} = 2.5V	10			ns
		V_{DD} = 5.5V	20			
$\overline{CS}$ Pulse Width High	t_{CSW}		40			ns
SCLK Period	t_{CP}		80			ns



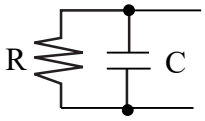
(5) Question 12. What architecture type is the MAX548A DAC?

(5) Question 13. Assume the switch has been pressed for a long time. Make a rough sketch of voltage V as a function of time when the switch is released.





(5) Question 14. What is the impedance magnitude, |**Z**|, as a function of **R**, **C**, and **f**?



|**Z**| =

(5) Question 15. We measure the resistance of a DC motor to be 10 Ω. When powered with a 10V supply, we expected the current to be 10V/10 Ω = 1A. However, under steady state conditions, the measured current is 2 A. Explain how this could happen. Show equations that when invoked lead to the current being 2 A.

(5) Question 16. Answer true or false

TCP is usually less reliable than UDP. ……

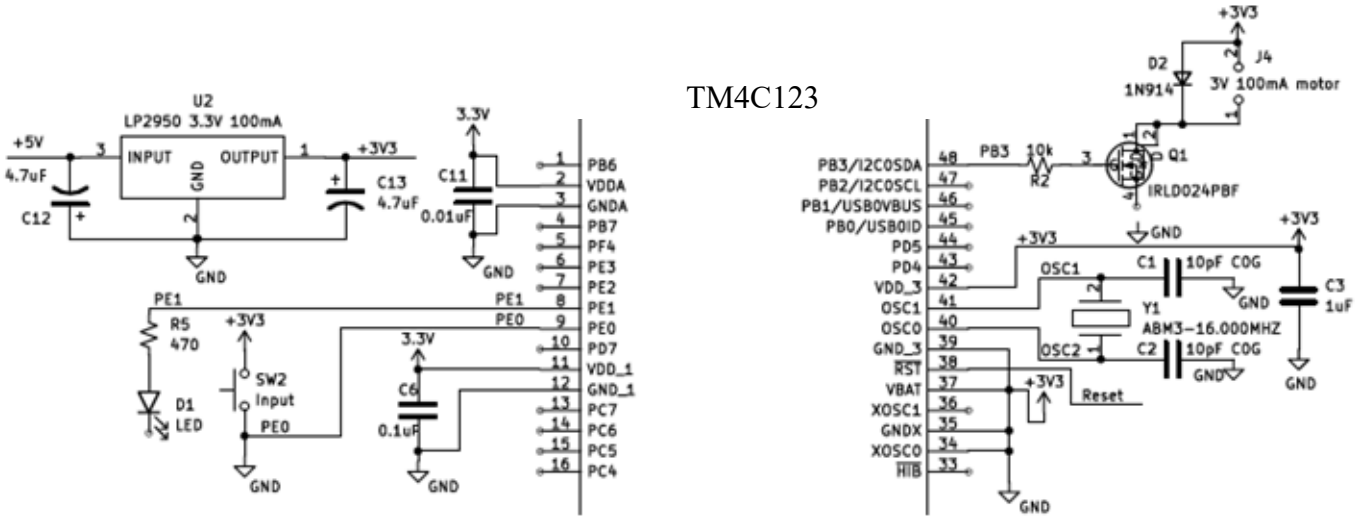
TM4C123 LaunchPad can send wifi transmissions. ……

With MQTT, there is direct communication from the ESP8266 to the web app. ……

The MQTT broker uses DNS to find the IP address of the client. ……

ESP8266 and the web app must connect to the same **port** of MQTT broker. ……

(5) Question 17. There are nine major errors in this portion of the schematic. Major means the circuit will not work. An LED is interfaced to PE1 output, a switch is interfaced to PE0 input, and PB3 output is connected to the gate of the MOSFET. J4 is connected to a 100mA DC motor. There are no errors with the R or C values and no errors on the crystal clock. Other than the debugger pins and additional TM4C123 power/ground pins (not shown) there are no other components on the schematic.



Find five major errors. For each error, circle the mistake on the schematic and draw a line to a box and briefly describe the error in the box. Ignore the 32 TM4C123 pins not shown.

(5) Question 18. The main program calls **put**, and the ISR calls **get**. There are no bugs other than a critical section. Modify the code to remove the critical section. For this system, any nonzero value is data, and 0 means empty.

```
int put(char data){

  if(Size == 32){

    return 0; // full
  }

  FIFO[PutI] = data;      // save

  PutI = (PutI+1)&0x1F;  // next

  Size++;

  return 1;
}
```

```
char get(void){ char data;

  if(Size == 0){

    return 0; // empty
  }

  data = FIFO[GetI];   // retrieve

  GetI = (GetI+1)&0x1F;   // next

  Size--;

  return data;
}
```