EECS 373 Final Exam

Winter 2017

Name: _____ unique name: _____

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

_____

Scores:

Problem	Points
1	/10
2	/17
3	/15
4	/10
5	/15
6	/17
7	/16
Total	**/100**

NOTES:

1. Closed book/notes.
2. There are 13 pages including this one.
3. Do not remove any pages from the test.
4. Calculators permitted. Accessing course-related data or special-purpose routines entered into calculator memory before the exam is forbidden.
5. You have 120 minutes for the exam.
6. Show work for partial credit.
7. There are two blank pages at the end for overflow work. You must write "**see end pages**" by the relevant problem for us to check them.

1. (10 pts.) Design a shaft encoder capable of detecting direction and distance of rotation with a resolution of at least 22.5 degrees, and also capable of determining absolute orientation with an error of at most 90 degrees. Subject to these requirements, minimize the number of light sensors and emitters; having more increases failure rate and maintenance time.

   No need to specify every detail. However, you must provide the following information.

   a. Absolute orientation
      i. Number of tracks

      ii. Code sequence cycle

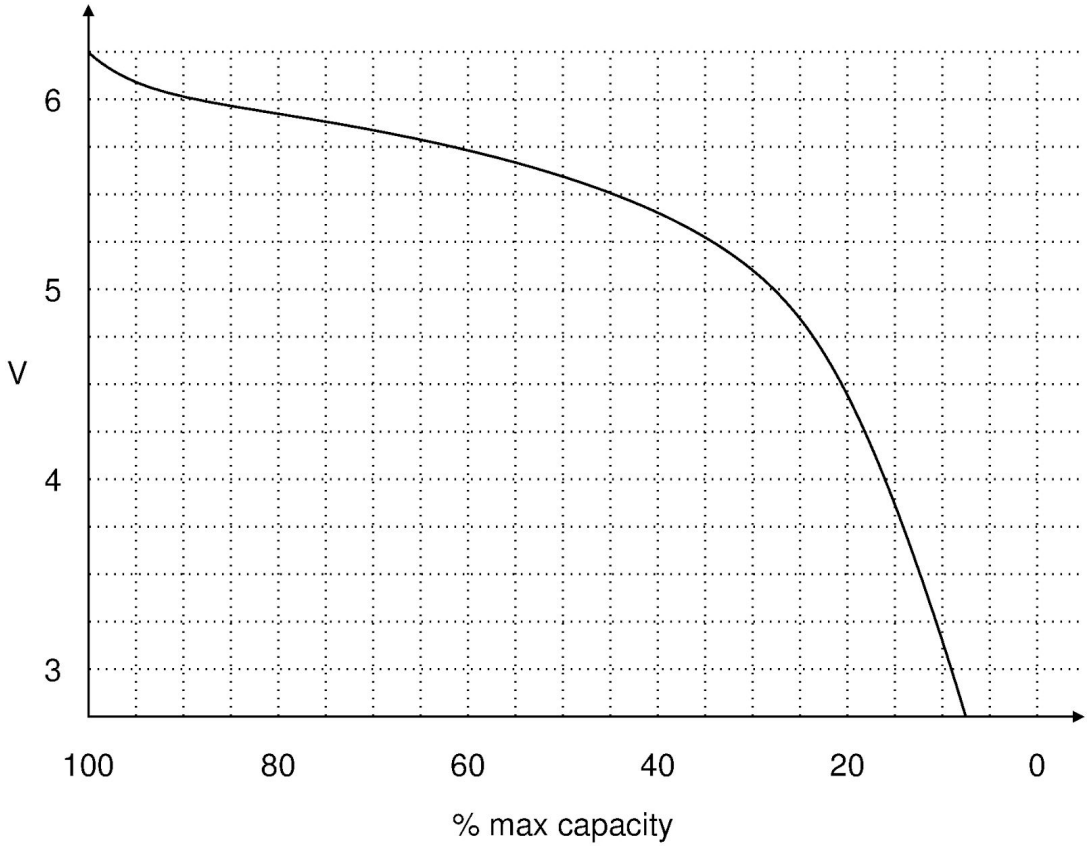
      iii. Number of times cycle repeats

   b. Relative rotation
      i. Number of tracks

      ii. Code sequence cycle

      iii. Number of times cycle repeats

2. (17 pts) Estimate the battery lifespan for the following embedded system, which is powered from a 2A-h battery pack having the following discharge curve.



The embedded system nominally operates at 3.3V. The components of the embedded system have the following characteristics:

CPU

- Operating voltage range: 3.2V-3.4V
- Average current
    - Low-power mode: 3mA
    - Normal mode: 300mA
    - Sprint mode: 600mA

Wireless interface

- Operating voltage range: 3.1V-3.4V
- Average current
    - Low-power mode: 5mA
    - Active mode: 500mA

Other information

- The system uses a Buck converter requiring that its input voltage be 1.2V higher than its output voltage.
- The CPU spends 80% of its time in low-power mode, 15% of its time in normal mode, and 5% of its time in sprint mode. It has a hardware timer that remains operational when the CPU is in low-power mode, and the timer can be used to enact a CPU power management state change.
- The wireless interface needs to enter active mode for 1ms per hour. However, it contains internal logic forcing it to remain in active mode for 10 minutes before returning to low-power mode.

Show your work.

a. CPU average power consumption.

b. Wireless interface average power consumption.

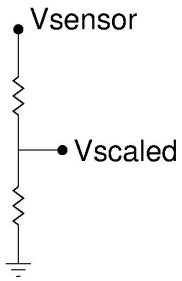
c. System average power consumption.

d. System lifespan starting from full battery charge.

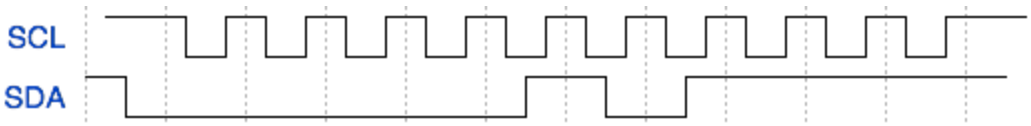
3. (15 pts.) Consider a 32-bit microcontroller with a 64-bit hardware timer. Implement a function to capture the 64-bit number. The function prototype is given below.
    - The address of the higher 32 bits is 0x40004504.
    - The address of the lower 32 bits is 0x40004500.
    - The microcontroller can load at most one word per instruction so multiple reads will be required for multiple words.
    - Think carefully about how you should arrange the order of the reads. In an extreme case, you might read a value with very high error. In this case, you should return 0 to inform the user the read was unsuccessful.

```
uint64_t Capture(void) {




}
```

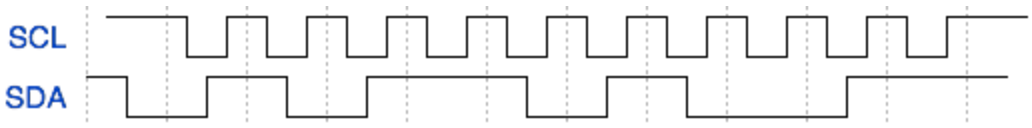
4. (10 pts.) Assume you are working with a sensor that operates between 0V and 5V. Your ADC only works between 0V and 2.5V. Construct a simple voltage divider to divide the sensor voltage by 2 with 10K resistors. When you hook up the circuit you discover that the sensor voltage is divided by 3. Yes, 3. Assuming you only have 10K resistors to work with, fix the divider by adding series resistance or parallel resistance so that will produce a 0 to 2.5V output range. It is fine to neglect capacitive and inductive parasitic effects.



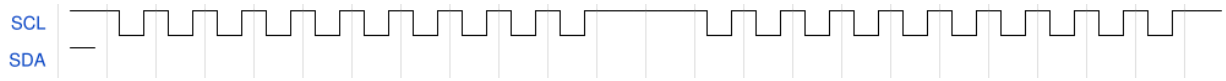
5. (15 pts.) See reference at end of exam.
   a. You have written a library to have the microcontroller interface over I2C with the RGB LED device at address 0x4D and an accelerometer at address 0x05. Reading from the accelerometer at address 0x05 returns three bytes representing the acceleration in the X, Y, and Z axes. However, when you attempt to do a read from the accelerometer's X axis register, you receive no response. The logic analyzer trace is below. What is the problem?

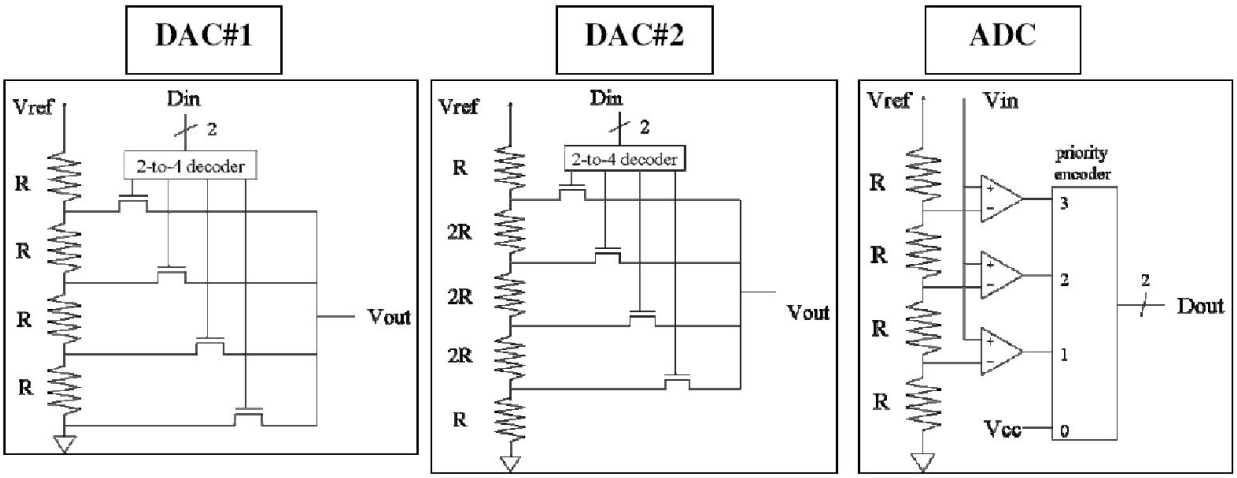


b. Now that you can read from the accelerometer's X axis register, you have moved on to writing to the RGB LED. To write to the RGB LED, you send three bytes: one representing the amount of red desired, one representing the amount of green, and one representing the amount of blue (all from 0 to 255). If only one byte is sent, then only the red component of the RGB LED's color is updated. You attempt a write to the RGB LED. However, the RGB LED does not change color. A copy of the first frame of the logic analyzer reading is below. What is the problem?



c. Please draw the proper timing diagram to write a red component value of 0x8C to the RGB LED.





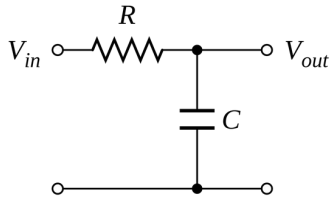
6. (17 pts.) The following questions are related to ADC and DAC conversion. Refer to the figures above. Vref=16V. The ADC and DAC output values have INL error up to ± ¼ the least significant bit. In DACs, the INL is the maximum output error. In ADCs, the INL is the maximum input threshold error. This error could make it possible for multiple outputs to be produced, given a particular input. Report all possible answers, or a range, as appropriate. The voltage comparators return true if the comparison voltage is greater than or equal to the reference voltage.
    a. If 12 V is put on the ADC input and D_{out} is connected to D_{in} of of DAC#1, what are the values that might possibly be present on V_{out} of the DAC?

    b. If “01” is put on the DAC#2 input and Vout is connected to the ADC’s Vin, what are the values that might possibly be present on the ADC’s Dout?

c. One application of ADCs is in the sampling and manipulation of audio signals. In order to properly sample audio signals and avoid aliasing, a filter must be applied before the signal is sampled and converted. Use the fact that humans can process audio signals up to 20kHz to answer the following two questions about the design of an audio sampling ADC. State the minimum sampling rate that the ADC must sample at in order to avoid perceptible information loss.

d. Given a capacitor value of C = 0.1uF, choose an R value to complete a low pass filter with a cutoff of 3dB, i.e., $\sqrt{1/2}$ the passband voltage, at the proper frequency. Use the following formula to do your calculations:

$$|V_{out}| = |V_{in}| \cdot \frac{1}{\sqrt{1 + w^2 \cdot R^2 \cdot C^2}} \quad \text{where} \quad w = 2\pi f$$



7. (16 pts.) See reference at end of exam.

0xE000E100	SETENA0	R/W	0	Enable for external interrupt #0–31 bit[0] for interrupt #0 (exception #16) bit[1] for interrupt #1 (exception #17) ... bit[31] for interrupt #31 (exception #47) Write 1 to set bit to 1; write 0 has no effect Read value indicates the current status

a. We would like to enable external interrupt #73. The address of the set enable base register is 0xE000E100 as shown in the figure above. What is the address of the register we should modify?

b. Which bit should we modify in this register?

c. Complete the following code that enables the interrupt for the input argument.

```
void enable_interrupt(int i) {
        volatile unsigned int * SETEN = (volatile unsigned int*) 0xE000E100;

        *(SETEN + _____) =
                _____;

}
```

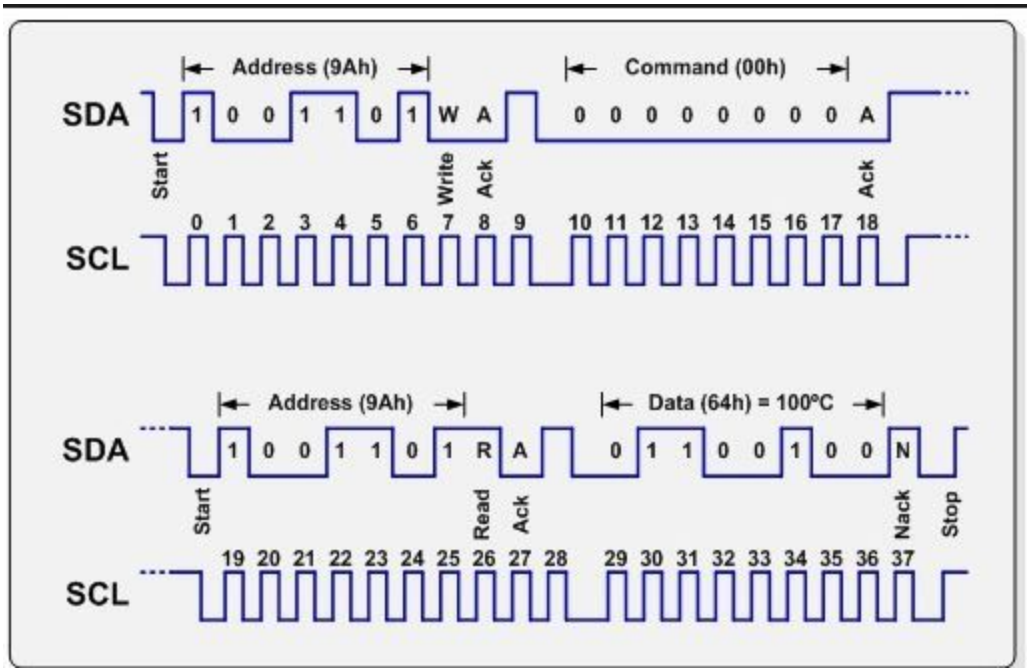
d. In lab we saw that problems can occur when a higher priority interrupt fires while the processor is already handling a low priority interrupt. Using at most once sentence, each, list two ways to prevent this from happening below.

Page for overflow work.

Page for overflow work.

References

I2C Transaction Example



Interrupt Priority Reference

PRIGROUP[2:0]	Binary point position	Pre-emption field	Subpriority field	Number of pre-emption priorities	Number of subpriorities
b000	bxxxxxxx.y	[7:1]	[0]	128	2
b001	bxxxxxx.yy	[7:2]	[1:0]	64	4
b010	bxxxxx.yyy	[7:3]	[2:0]	32	8
b011	bxxxx.yyyy	[7:4]	[3:0]	16	16
b100	bxxx.yyyyy	[7:5]	[4:0]	8	32
b101	bxx.yyyyyy	[7:6]	[5:0]	4	64
b110	bx.yyyyyyy	[7]	[6:0]	2	128
b111	b.yyyyyyyy	None	[7:0]	0	256