EECS 473 Midterm Exam

Fall 2015

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.	/9
3.	/11
4.	/13
5.	/13
6.	/8
7.	/8
7.	/28
Total	**/100**

NOTES:

1. Closed book and Closed notes
2. There are **12** pages total for the exam as well as a handout. The last 2 pages of the exam can be removed and used as reference for the last problem.
3. Calculators are allowed, but no PDAs, Portables, Cell phones, etc. Using a calculator to store notes is not allowed nor is a calculator with any type of wireless capability.
4. You have about 120 minutes for the exam.
5. **Be sure to show work and explain what you've done when asked to do so. That may be very significant in the grading of this exam.**

1) Circle the best answer. **[10 points, -2 per wrong or blank answer, minimum 0]**

a. PCB manufactures use a number of different terms to describe the distances. The most common are "thou", "mm" and "mil". A thou ***is 0.001 inches / 0.001 meters / 0.001 cm / 0.001 feet***. A mil ***is 0.001 inches / 0.001 meters / 0.001 cm / 0.001 feet***.

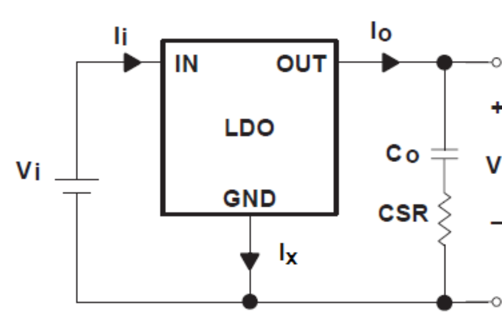
b. Busybox is a utility commonly found on embedded Linux platforms. It is generally used because:
- ***It enables a better form of deferred interrupts on Linux.***
- ***It provides a relatively small executable that implements a wide variety of standard programs.***
- ***It uses soft links to provide a guarantee of minimum response time on interrupts.***
- ***Even though it was originally a multi-channel drum machine program, it is now commonly used for simple audio applications that involve warning beeps and the like.***

c. Priority inversion is where:
- ***A semaphore or other mutex is used to lock a resource and that lock is not released when it should be.***
- ***A semaphore or other mutex is used with "1" being free and "0" being not-free, rather than the other way around.***
- ***Priority inheritance sets the high-priority task to the lowest possible priority.***
- ***A high priority task is indirectly preempted by a medium priority task which effectively inverts the relative priorities of the two tasks.***

d. Which of the following is a technique we might use on a real-time systems that we would almost never use in a desktop computer? The technique and its reason for use must both be correct.
- ***Turning off the data cache to enable more consistent run times.***
- ***Running code with optimization turned off to enable memory-mapped I/O.***
- ***Running code with optimization turned off to enable more consistent run times.***
- ***Declaring variables "volatile" to improve processing speed.***

e. On a PCB, traces on two different layers of a board are connected by ***vias / traces / solder mask / artwork / patterns***.

f. In the diagram on the right, CSR and C_O are the output load and Vi is the input voltage. "I_X" could best be described as: ***output current / drive current / ecstatic current / ideal current / strong current / quiescent current***.



2) Say you have the following groups of tasks. For each group find the CPU utilization and identify which groups are RM and which are EDF schedulable. Indicate if you needed to do the critical instant analysis. *<u>If needed</u>*, ***<u>clearly</u>*** show that analysis. The following equation may prove useful. **[9 points]**

The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.

3x(2^(1/3) - 1) = 77.98%

Group	T1 Execution Time	T1 Period	T2 Execution Time	T2 Period	T3 Execution Time	T3 Period	% Utilization (Total)
A	2	7	2	5	1	4	**93.57**
B	2	7	2	13	1	3	**77.29**
C	3	7	2	5	3	15	**102.85**

Group	EDF Schedulable?	RM Schedulable?	Did you need to examine the critical instance?
A	**Y**	**N**	**Y**
B	**Y**	**Y**	**N**
C	**N**	**N**	**N**

1: OOOXOOO…miss
2: OXXOOXXOOOXXOOOXOXOOOXXOOXXO
3: XOOOXOOOXOOOXOOOXOOOXOOOXOOO

3) RM and EDF **[11 points]**

a. Why might one prefer to use RM scheduling over EDF? Why might one prefer EDF over RMS? **[4]**

RM might be preferred over EDF because
1. RM applies static priority, which is more stable.
2. RM is easier to program and
3. has lower overhead.

EDF might be preferred because
1. some case not schedulable under RM might be schedulable with EDF

b. For rate monotonic scheduling (RMS), it is the case that if the CPU utilization of a given set of tasks is below a certain bound and the tasks meet certain requirements, we know that the tasks are schedulable. Now consider a processor (and tasks) where *no preemption can occur*. Does such a bound exist when using the same scheduling algorithm?

If there is such a bound, provide a bound (it need not be a tight bound, just one where if the CPU utilization is lower than that it, it will always be schedulable). If there is no such bound simply state that. Provide an informal proof (basically an obviously true argument) for your answer. A correct answer with no explanation will get no points. **[7 points]**

No such boundary exists. RM without preemption is effectively a FIFO scheme. We learnt it from the assignment that no matter how low the CPU utilization is, we can always come up with a pair of tasks that are not schedulable with FIFO scheme.

4) Short answer **[13 points]**

a. What is a "deferred interrupt"? Why do we use them? Be clear and specific. **[4]**

Deferred interrupt is to have ISR scheduling another task to handle the interrupt instead of performing all the workload in ISR.

Reasons:
1. It is used to make the interrupt handler more responsive.
2. The start time of a task matters but the work load is not.

b. Why might a lower-power processor use more energy than a high-power processor? **[4]**

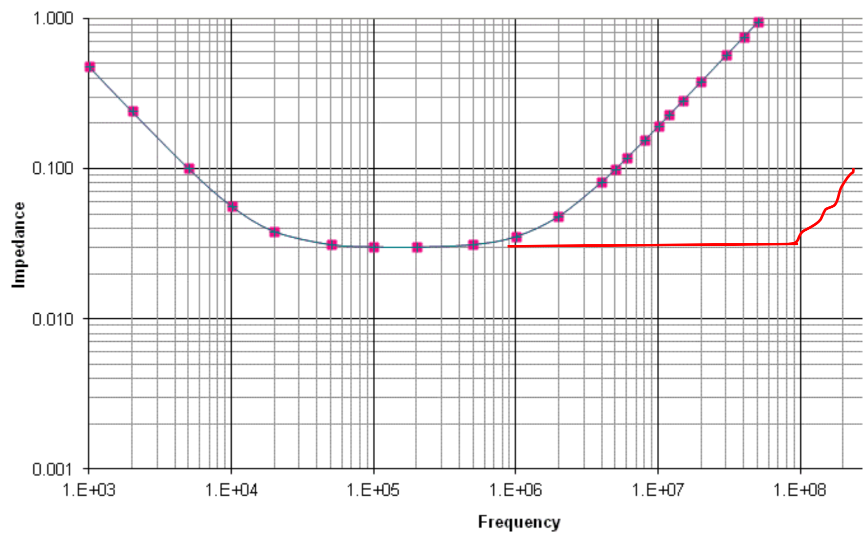
Lower-power processor takes longer to accomplish the same task.

c. Say we have a 5V 2000mAh battery. We are using it through an *ideal* LDO which outputs 3V to a processor that uses 0.06W. Nothing other than the processor is driven by the LDO or battery. How long would you expect the battery to last? You must clearly show your work to get credit. **[5]**

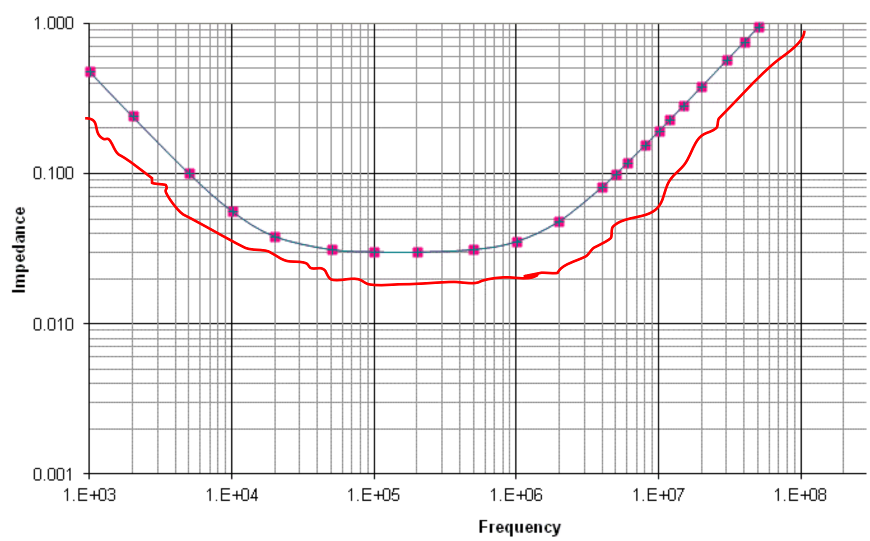
Current: 0.06W / 3V = 20mA
Since the current is much smaller than 1C, we assume that 2000mAh is available.
Therefore: 2000mAh / 20mA = 100 hour

5) Using capacitors **[13 points]**

a. To the right is a graph of the effective impedance of a 330µF capacitor with an ESR of 0.03Ω and an ESL of 3nH at a wide range of frequencies. *Carefully* modify the curve to show what it would look like if the ESL were instead 30pH. **[5]**



b. Explain why we often use more than one capacitor of the same type in parallel when trying to maintain power integrity. Use the graph to the right to illustrate your point. Be sure to clearly explain why the effects that you show are helpful to power integrity. **[5]**



Say you have 3 such capacitors in parallel, curve will shift down maintaining the shape touching on 0.01 line

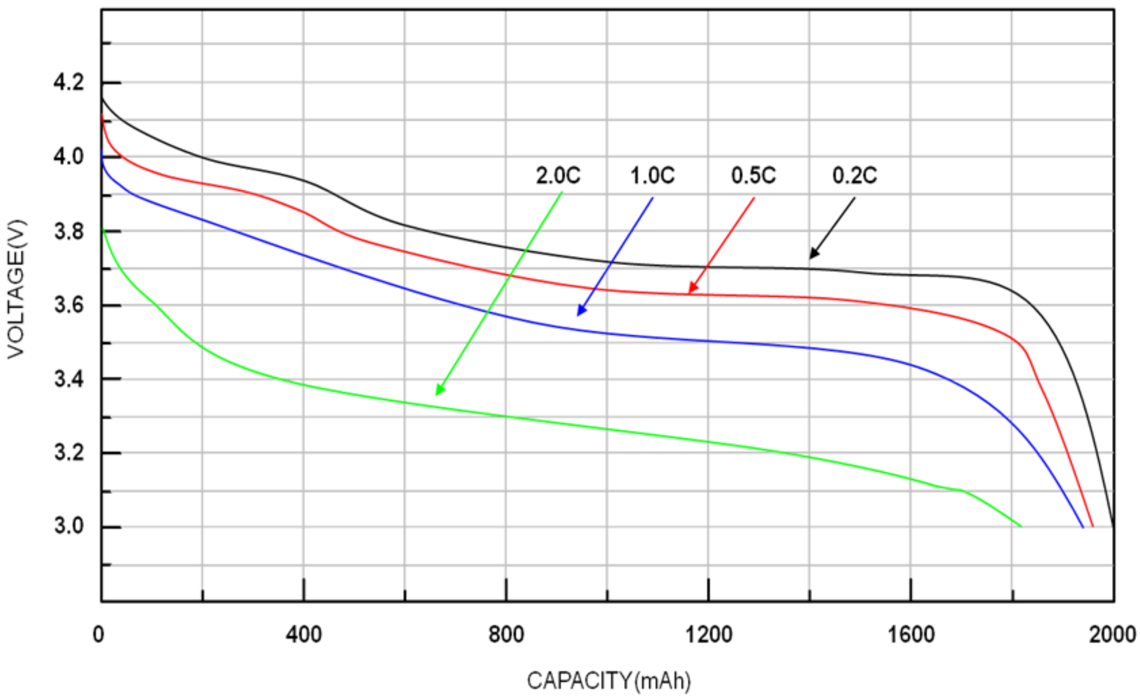
c. With respect to power integrity, why is having a PCB ground and power plane useful? **[3]**

Lower the voltage drop from power to components.

For signal integrity, it acts as a small capacitor to lower the impedance at high frequency.

6) Batteries **[8 points]**

Say you have a 2000mAh battery with the following characteristics:



a. If your embedded system (e.g. an RC car) needs 5-3.3V to function and draws 4A, how long will it be able to run on this battery? Show your work. **[3]**

800mAh/4A=.2hours=12 minutes.

b. If you used two of these batteries in series and used an ideal (i.e. current in=current out and no minimum voltage drop) linear regulator, how long could your 4A system run? Show your work. **[3]**

1800mAh/4A=0.45hours=27 minutes

c. Why might one choose **[2]**

i. An acid-lead battery over a LIPO battery?

Generally cheaper per Ah, and can easily generate large currents.

ii. An alkaline battery over a LIPO battery?

Cheaper per Ah , safer, sits on the shelf longer without losing charge. Most useful for rarely used things (most flashl

7) Embedded Linux—Answer the following questions. **[8 points]**

a. Consider the following as returned by "ls –l" **/dev/**

```
crw------- 1 root root 4,  3 Sep 22 22:31 tty3
crw--w---- 1 root tty  4, 30 Sep 22 22:29 tty30
```

Explain the following things:

- What does the "rw" after the leading c indicate? **[1]**

  readable writeable to root (the owner)

- In tty3 what does the "4, 3" indicate? **[2]**

  Major and minor number

- In tty30 what does the "root tty" in tty30 indicate? **[2]**

  Owner and group So root is the owner and tty is the group

b. Explain how major numbers and minor numbers are used by devices. **[3]**

Major number - device category
Minor number - which specific device

The major number is used to select which module's code to run. The minor number is passed as an argument to that code and allows you to identify details of the specific device.

In the example above, both devices have a major number of 4. That means the same code will be called when interacting with either device. But the minor number allows that code to figure out which tty device is being addressed (tty3 or tty30).

8) Alarm System to prevent driver drowsiness by monitoring air quality. **[28 points]**

Billy is an intern at a car company. He needs your help in designing a prototype alarm which will detect if there is too much CO_2 in the air of a car[1]. The alarm system shall be able to measure the concentration of CO2 and trigger a warning mechanism if the value is too high.

To demonstrate the idea, Billy has decided to use a CO_2 sensor (MG811), connected to an Arduino Uno, which lights an LED whenever the CO_2 level is too high. Your job is to design a circuit diagram (devices shown below) and to write a sketch to accomplish this. ***Be sure to read the entire question briefly look over the various documents available to you before you start.***

a. How much voltage does the CO_2 sensor require? **[1]**

6V

b. What is the range of CO2 concentration that this sensor can measure? What are the voltages values associated with those two extremes? **[2]**

350 — 10,000 ppm: 325 — 265mV

c. If you want the input to your Arduino to never exceed 4V, what integer should you multiply the input voltage to get the widest possible output range? **[1]**

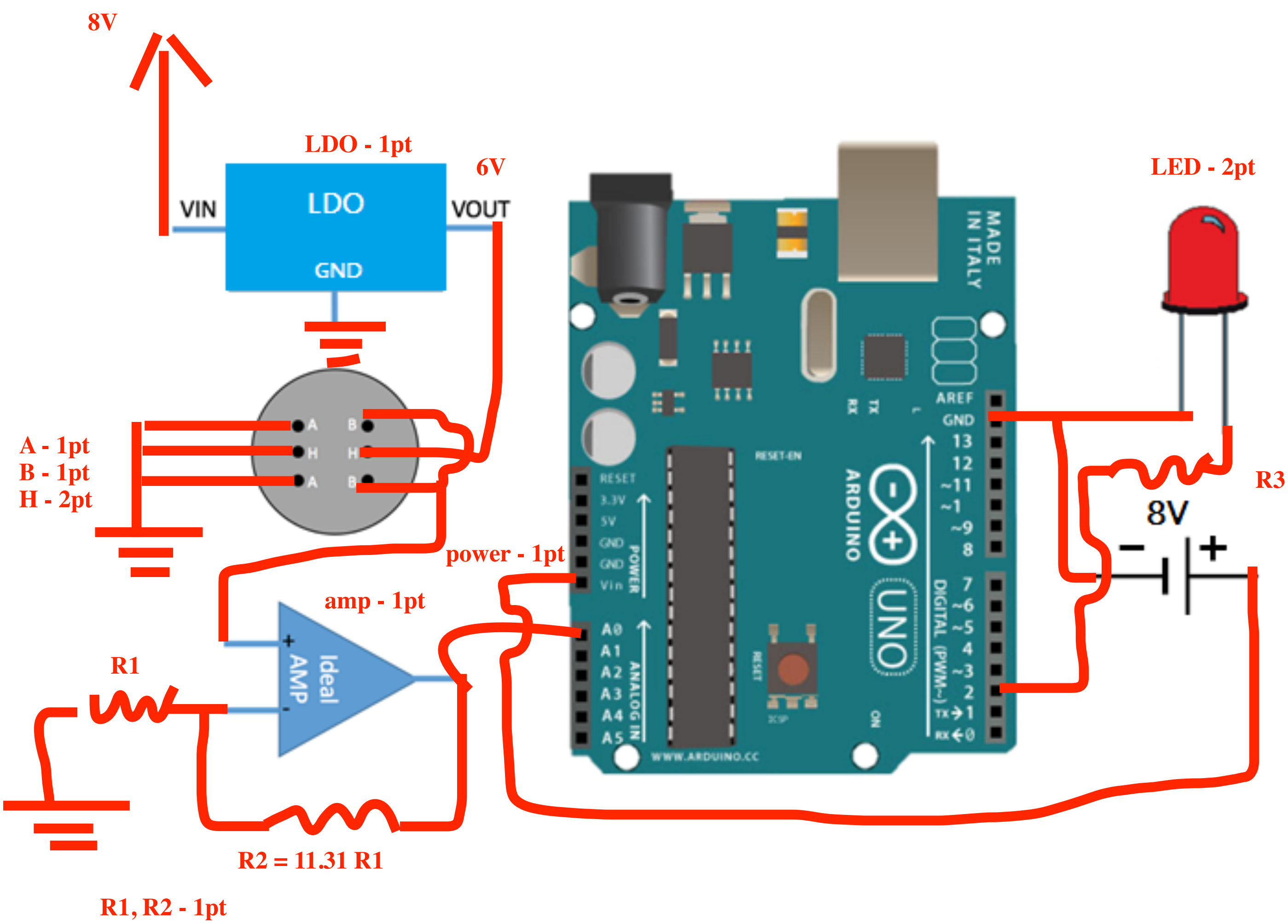
floor(4V / 325mV) = 12

[1] Hyundai engineers claim elevated carbon dioxide levels created by occupant respiration inside the vehicle cabin can cause drowsiness and slow reaction times. Academic papers support this.

d. Below are a number of components: an LDO, the CO_2 sensor, an ideal opamp, an Arduino Uno, an LED and an 8V power supply. No devices have power unless you supply it. What voltage level would you want to get from the LDO? **[2]**

6V

e. Indicate, by drawing wires, how you will connect the six components, and draw any other components needed. **[9]**

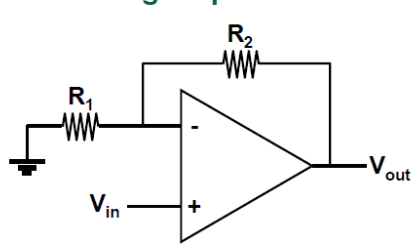


UNO:

Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA

Ideal opamp

- **Non-inverting amplifier**



$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_{in}$$

f. Write an Arduino sketch that lights the LED if CO_2 concentration reaches a level of 3000ppm. You should sample the sensor about once every 10 seconds and print the concentration through serial port. **[13]**
Hint: you might want some helper functions and macros.

```
#define MG_PIN 0
#define LED_PIN 2

#define CURVE_START_VOLTAGE 0.324 // 400 ppm
#define CURVE_START_LOG_PPM 2.602 // log(400)
#define CURVE_END_VOLTAGE 0.265 // 10000 ppm
#define CURVE_SLOPE (-0.0422) // (CURVE_END_VOLTAGE - CURVE_START_VOLTAGE) / (log(10000) - log(400))
#define DC_GAIN 12.35 // 4 volts / CURVE_START_VOLTAGE, result of question 3

#define MG_WARMUP_TIME 120000 // optional warm up time for the sensor
#define MEASURE_PERIOD 10000

// get sensor voltage output (true voltage without dc gain)
float MGGetVoltage();
// fit voltage onto the curve and calculate the concentration
int MGCalcConcentration(float voltage);

int co2_concentration = 0;
float voltage = 0;

void setup() {
	Serial.begin(9600);
	pinMode(LED_PIN, OUTPUT);
	digitalWrite(LED_PIN, LOW);
	Serial.println("Warm up the sensor...");
	delay(MG_WARMUP_TIME);
}

void loop() {
	voltage = MGGetVoltage();
	co2_concentration = MGCalcConcentration(voltage);
	if (co2_concentration > 3000) {
		digitalWrite(LED_PIN, HIGH);
	}
	Serial.print("ppm: ");
	Serial.println(co2_concentration);
	delay(MEASURE_PERIOD);
}

float MGGetVoltage() {
	return analogRead(MG_PIN) * 5 / 1024;
}

int MGCalcConcentration(float volt) {
	float sensorVolt = volt/DC_GAIN;
	if (sensorVolt >= CURVE_START_VOLTAGE)
		return -1;
	else
		return pow(10, (sensorVolt - CURVE_START_VOLTAGE)/CURVE_SLOPE + CURVE_START_LOG_PPM);
}
```