EECS 473 Final Exam

Fall 2014

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.	/10
3.	/11
4.	/9
5.	/15
6.	/4
7.	/8
Design	/33
Total	**/100**

NOTES:

1. Closed book and Closed notes
2. For purposes of this exam, assume the speed of light is exactly 3×10^8 m/s.
3. There are **13** pages total for the exam as well as 2 handouts.
4. Calculators are allowed, but no PDAs, Portables, Cell phones, etc. Using a calculator to store notes is not allowed.
5. You have about 120 minutes for the exam.
6. **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. What does it mean when we say the GPL license is "viral"?
        - **<u>Any use of GPL'ed source code in your program means that your code must also be licensed under the GPL.</u>**
        - **<u>Any use of the binary (complier, editor, OS, etc.) when building software requires that that software also be licensed under the GPL.</u>**
        - **<u>Any use of the GPL'ed source code by a company requires that they license all of their software under the GPL.</u>**

    b. If you are drawing "3C" from a single lithium polymer battery the *longest* you could expect the battery to last is **<u>3 hours / 1 hour / 20 minutes / 10 minutes / 5 minutes</u>**.

    c. Given a certain bandwidth, signal power and noise power, Shannon's limit tells us
        - **<u>The maximum number of symbols we can reliably send per second.</u>**
        - **<u>The maximum rate of information we can reliably send per second.</u>**
        - **<u>The maximum distance at which we can expect to be able to reliably receive data.</u>**
        - **<u>The maximum SNR that we can have and still reliably receive data.</u>**

    d. What is the primary disadvantage of fixed point when compared to floating point?
        - **<u>It takes more power to perform a fixed point operation than for floating point.</u>**
        - **<u>It doesn't do as good of a job representing very large and very small numbers.</u>**
        - **<u>For a given number of bits used, it can represent many fewer distinct values.</u>**
        - **<u>Floating point addition is much faster than fixed point addition.</u>**

    e. A common reason to prefer RMS over EDF scheduling is that
        - **<u>EDF scheduling cannot schedule tasks groups that RMS can.</u>**
        - **<u>EDF scheduling requires changing task priorities while RMS does not.</u>**
        - **<u>EDF scheduling *can* result in live-lock as the tasks jump in-and-out of execution much more rapidly than under RMS.</u>**
        - **<u>EDF scheduling *can* result in dead-lock if two tasks have identical deadlines, while RMS will not.</u>**

    f. The three things we can modulate for communication are
        - **<u>phase, frequency, and amplitude.</u>**
        - **<u>phase, I/Q values, and amplitude.</u>**
        - **<u>I/Q values, amplitude and power.</u>**
        - **<u>frequency, power, and amplitude.</u>**

2. Answer the following questions about power converters.
   [10 points, -2 per blank or wrong answer, minimum 0]

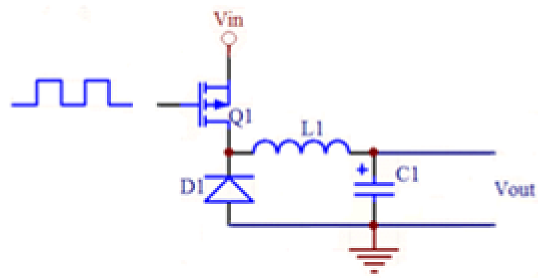


Figure 1: A voltage converter

   a. Figure #1 is a(n)
      LDO / buck / boost / buck-boost converter.

   b. What is the purpose of the diode in figure 1?
      - ***It prevents the capacitor from getting too high of a voltage.***
      - ***It gives the inductor someplace to draw current from when the transistor is off.***
      - ***It stores energy to provide power when the transistor is off.***
      - ***It controls the transistor's switching rate.***

   c. A(n) ***LDO / buck / boost / buck-boost*** converter which converts 10V to 8V can never be more than 80% efficient.

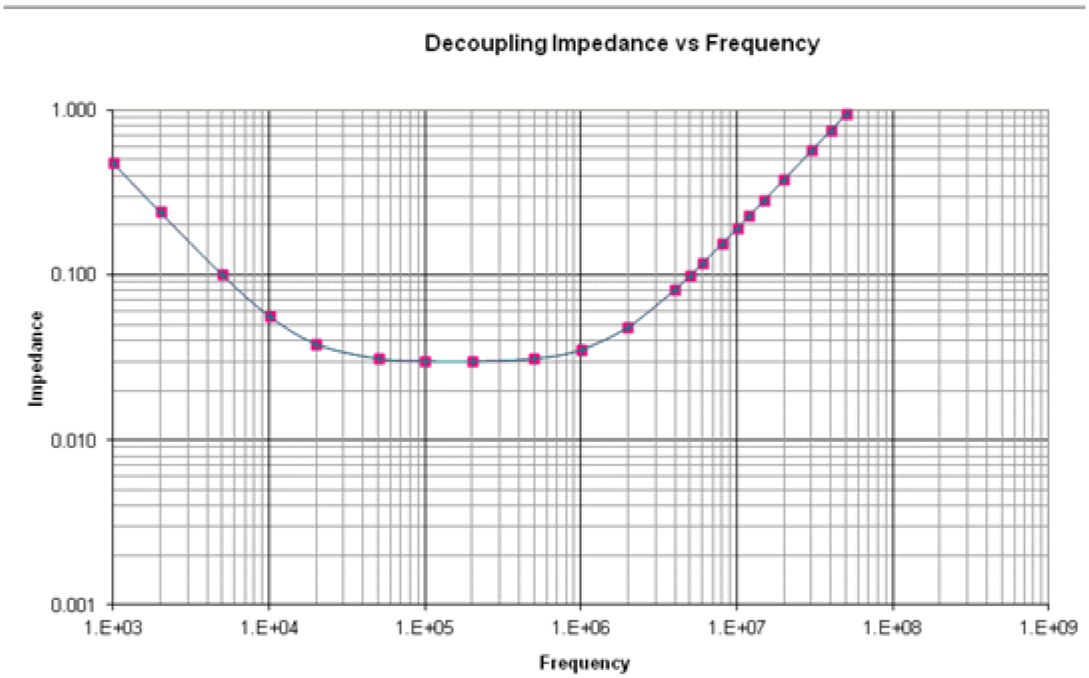
   d. An otherwise ideal buck converter which has a quiescent current of 5mA and an efficiency of

      90% will pull _____________mA from its 12V input if it is generating 100mA at 5 volts. (Round to the nearest mA).

   e. Sometimes, when we want a particular voltage, we use a buck converter followed by an LDO, but almost never use an LDO followed by buck converter. What is the most common reason for using a buck converter followed by an LDO? Assume we only need 1 voltage level for the entire board.
      - ***The buck converter generates an approximate voltage, while the LDO can provide much finer-grain control.***
      - ***The buck converter can drop small voltages efficiently, while the LDO can drop large voltages efficiently.***
      - ***The buck converter can drop large amounts of voltage efficiently, while the LDO generates a cleaner output.***
      - ***The buck converter can drop large voltages efficiently, while the LDO can drop small voltages efficiently.***

   f. The "L" in an LDO converter indicates that
      - **the output voltage will be considerably lower than the input voltage.**
      - **the output voltage can be nearly as high as the input voltage.**
      - **you only need a low value output capacitor, often less than a µF.**
      - **the output has very low noise.**

3. Short answer **[11 points]**
   a. The following 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. Modify the curve to show what it would look like if the *ESR* were instead 0. **[5]**



   b. Consider using binary FSK as a modulation scheme. The two frequencies used are "A" 3.4000GHz and "B" 3.4001GHz. If the receiver was traveling away from the sender, it's possible that when the sender sends one of those two, the receiver would see the other due to Doppler effects. Assume there are no reflections.
      - Could A appear to be B or B appear to be A? **[1]**
      - How fast would the receiver have to be traveling for this to occur? Show your work. **[5]**

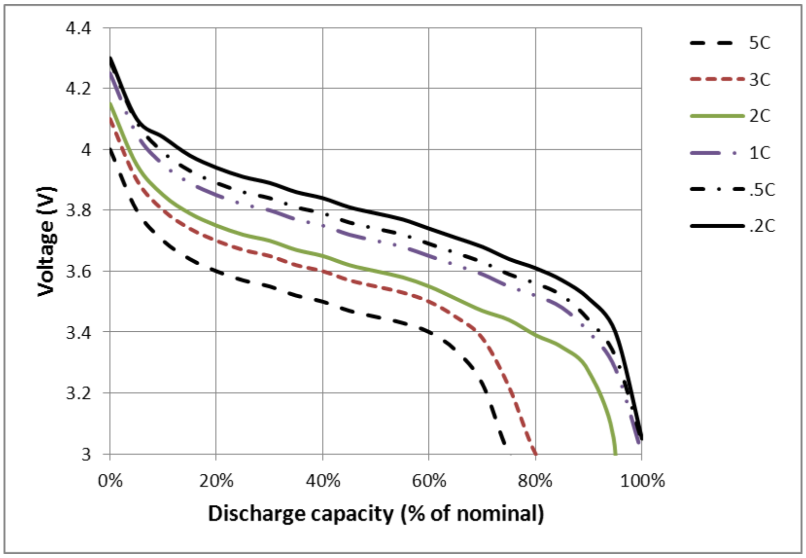
4. Say you have the following groups of tasks. For each group find the CPU utilization and identify which groups are RM schedulable. Indicate if you needed to do the critical instant analysis. *If needed*, ***clearly*** show that analysis. The following equation may prove useful.
[9 points]

$$\sum_{i=1}^{n} U \leq n(2^{1/n}-1)$$

Group	T1 Execution Time	T1 Period	T2 Execution Time	T2 Period	T3 Execution Time	T3 Period	% Utilization (Total)
A	10	20	6	18	1	7	
B	10	20	5	11	1	22	
C	10	20	3	10			

Group	RM Schedulable?	Did you need to examine the critical instance?
A		
B		
C		

5. Batteries and more **[15 points]**
Say you have a family of batteries with the following discharge profile (. You can assume that any draw rate lower than 0.2C follows the 0.2C curve and any draw rate greater than 5C to follow the 5C curve.



a. If your embedded system needs at least 3.6V to function and draws 40mA, about how long will it be able to run on this battery if its nominal capacity is 200mAh? **[3]**

b. You are considering designing a system in one of two ways—either a low-power processor that runs continuously or a high-power processor that runs only some of the time. The lower-power processor uses 1mA, while the high-powered processor uses 10mA but only needs to run for 4 seconds every minute. Both processors require 3.3V and are using a LDO with a minimum drop of 0.1V. **[6]**

   i. What is the minimum battery capacity you would need for this family of batteries to run each of these systems for 1 week? Show your work.

      1. Low-power? ____________

      2. High-power? ____________

c. As part b, but you only want to run for 20 minutes. Again, show your work. **[6]**

   1. Low-power? ____________

   2. High-power? ____________

6. The XBee chips a number of groups used in class use a 2MHz wide frequency band at around 2.4 GHz. They generally can achieve a 250,000 bit/sec data rate. Assuming you have background noise of 0.001µW and signal strength of 0.1mW at the receiver, what is the maximum reliable data rate you could theoretically achieve with these devices? Show your work. **[4 points]**

$$C = B\log_2\left(1 + \frac{S}{N}\right)$$

- **C** is the channel capacity in bits per second;
- **B** is the bandwidth of the channel in Hertz
- **S** is the total received signal power measured in Watts or Volts^2
- **N** is the total noise, measured in Watts or Volts^2

7. You have data being sent from point A to point Z using on-off keying (OOK). A and Z are 10m apart and data is being sent at a rate of 5 million symbols per second on a carrier frequency of 2.5GHz. Due to a reflection, the signal also travels along one additional path that is 14.5m long. The reflected signal and the direct signal both arrive with a power of 1µW. **[8 points]**
   a. Is the reflected signal primarily causing inter-symbol interference or intra-symbol interference? Justify your answer. **[3]**

   b. Assuming the direct signal and the reflected signal are both "on", what is the power perceived by the receiver? **[5]**

Design problem—Accelerometer controlled camera.

We strongly suggest you read the entire question before doing any one part.

You are an intern at a company which builds a battery-powered camera system targeted toward skydivers. Your task is to implement a few features that use a 3-axis accelerometer (ADXL345) over **I^2C**. Other teams have developed an API and functions that will be useful for you to implement your features. Documentation for these features and APIs are found below. The system uses the BCM 2835, which is the same processor as the Raspberry Pi. The processor will be running FreeRTOS in your application.

For this question you will write the code that will run your intern demo project that will use a Raspberry Pi and an LED to demonstrate your features. Hopefully, your code will make it into production there will be other code running on the controller. Therefore, you are to run any non-ISR code at priority "1" and your ISRs should not exceed 10ms of execution time.

Your project has two basic features:

- When the accelerometer detects a double tap, your software should toggle the camera status (on or off). Additionally, you are to light an LED if the camera is on.
- When the accelerometer detects freefall, your software is to take a GPS reading and record that value. Thankfully, all of the GPS software has been written by someone else—you just need to tell it to take the GPS reading.

More formally, you are to implement the following features:

Tap-To-Record (T-T-R)

What:

When a double tap is detected by the ADXL345 in the z-axis, you are to toggle the camera activity state (on/off). You should set LED0 high while the camera is active. Functions to turn on and off the camera are provided below.

T-T-R specification:

The ADXL345 has a highly configurable double-tap detector. Your group tells you that the following values work well.

Thresh_tap: 3125mg; DUR: 30ms; Latent: 50ms; Window: 140ms;

In addition, they tell you that you should not set the suppress bit in Register 0x2A.

Freefall Detect (FD)

What:

Whenever freefall is detected by the ADXL345, you are to call a function which will cause the GPS to measure your position and record it.

FD specification:

The ADXL345 has a highly configurable freefall detector. Your group tells you that the following values work well.

Thresh_FF: 500 mg; Time_FF: 120 ms

Additional notes

The RTOS tick is 500µs.
Ignore the accelerometer axis offset.
Assume BCM 2835 only has one I^2C peripheral connected to it.
You may only use one interrupt pin and you are the only one using GPIO interrupts
You are to ground the SDO/ALT ADDRESS pin.

API Functions (Unless otherwise noted all functions return in less than 1ms)

`bool camera_is_active(void)`
Returns 1 if recording 0 if not.

`void activate_camera(void)`
Turns the camera on.

`void deactivate_camera(void)`
Turns the camera off

`void i2c_transaction(uint8_t daddr, uint8_t raddr, char* data)`
Performs either a read or write based on address.
daddr: The 8-bit read/write device address
raddr: The address of the register to be read or written
data: a pointer to a local buffer

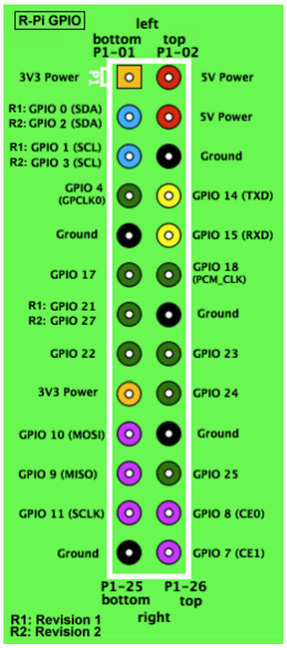
void capture_and_log_GPS_pos(void)
This function may block for up to 50 ms.

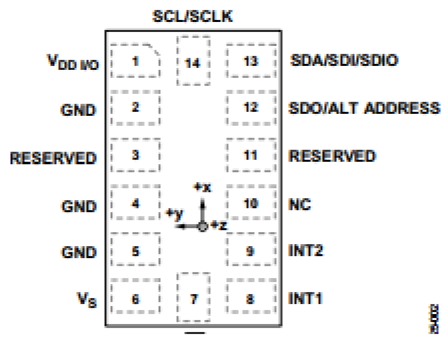
void set_gpio(unit8_t pin_num, unit8_t pin_val)
Pin_val is active high.

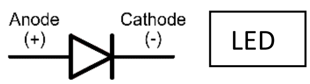
FreeRTOS functions are defined in a separate handout.

1. Answer the following questions. **[8 points]**
   a. What are the I²C read and write addresses of the accelerometer? **[2]**

   b. Add the wires and other devices needed to hook-up accelerometer and LEDs. **[6]**







2. Thankfully, *<u>you</u>* have a more junior intern working with you. She is to write a function which does most of the configuration of the ADXL345. You need to tell her what values to set things to. She will handle all of the needed configuration other than INT_ENABLE and INT_MAP. To what values should she initialize the following registers? **[7 points]**

Name	**Value (in HEX)**
THRESH_TAP	
DUR	
Latent	
Window	
THRESH_FF	
TIME_FF	
TAP_AXES	

3. Fill in the template below. Please use the example code. Feel free to write additional functions as needed/desired. **[18 points]**

```
#include <FreeRTOS.h>
#include <task.h>
#include <semphr.h>

#include "api.h"  // These are the API functions defined above.

#define ON  1
#define OFF 0
#define STACK_SIZE 128

// Please fill in pin numbers
#define LED0_PIN
#define INT_PIN
```

```
void gpio_ISR(int num, void*parm){
      ClearGpioInterrupt( INT_PIN );












}
```

```c
int main (void) {
	disable_interrupts();

	// This function configures the BCM_2835 interrupt
	// vector, registers gpio_ISR (defined above) as the gpio
	// interrupt handler and enables it.  It is written for
	// you.

	configure_BCM2835_interrupts();

	// This function configures the LED defined by the
	// LED0_pin macro as an output and sets its initial state
	// to low. It is written for you.


	configure_LEDs();

	// This is the function the intern you are working with wrote.
	// It will configure things as you specfifed in the table.
	// Recall You still need to configure INT_ENABLE and INT_MAP.

	configure_ADXL385_no_interrupts();














	vTaskStartScheduler();
	enable_interrupts();









	return 0;
}
```