
EL6483 Quiz 2 SPRING 2021 (Open book, notes, etc)

Name: _____

ID: _____

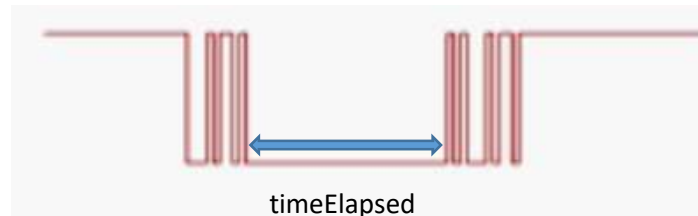
NYU Tandon School of Engineering

1. Suppose you are required to retrieve the temperature from the Analog Devices AD5933 chip, which is used to make very precise impedance measurements. The interface is I2C and the relevant portions of the datasheet for this chip are attached. Also assume you have access to several I2C functions as follows:

- a. int Start_I2C()*
 - i. Initializes all of the I2C hardware pins etc.
- b. int I2C_Write_Byte(uint8_t data)*
 - i. Writes 'data' on the I2C Bus. Returns 1 if slave ACK, 0 is slave NAK
- c. int I2C_Send_Start_Condition(uint8_t address, int isWrite)*
 - i. Send a start condition, followed by 7-bit address, followed by 'isWrite' bit.
 - ii. Returns 1 if successful, 0 otherwise
- d. int I2C_Send_Stop_Condition()*
 - i. returns 1 if successful
- e. int I2C_Request_Read(uint8_t *buffer, int numBytes)*
 - i. reads 'numBytes' bytes off the bus, each followed by an master ACK except for the last byte, which if followed by a master NAK. Returns 1 if it gets all the bytes, 0 otherwise

Using the datasheet provided, write a C code function *float GetTemperature()* required to get the temperature of the AD5933 chip. You can add any additional variables as long as you state what they are for and be sure to state any assumptions made.

2. The requirement for this question is to simply measure the time elapsed while a single button (GPIO) is being pressed using interrupts. The GPIO pin is pulled to a high state when the button is not depressed. In this application, the button is mechanical and experiences significant bouncing. The goal is to measure the contiguous time (in ms) between when the button is pressed and when the button is released, excluding the bouncing. See diagram below.



You have already setup 3 interrupt handlers. One handler is the ISR that handles Timer0's overflow. That timer is set up with a prescaler of 32, a counter TOP value of 250, and is set up to reset to 0 and count up to TOP. The CLK is running at 16MHz. Here is the handler definition:

```
void Timer0_OV_Handler()
```

The other 2 handlers are for the pin that is connected to the button. One handles the RISING transition, and the other handles the FALLING transition. Here are the definitions:

```
void pinRising_Handler()
```

```
void pinFalling_Handler()
```

Write the required code snippets (in each handler) that assigns the variable *timeElapsed* to the specified time in the timing diagram. You may add any additional variables, but **be sure to state any assumptions.**

3. Consider the following set of tasks:

Task	Arrival time	Computation time	Waiting time	Turnaround time
T_1	0	3		
T_2	2	6		
T_3	4	4		
T_4	6	5		
T_5	8	2		

- (a) Fill in the table above and draw a chart showing the arrival time of each task, and the schedule according to a round-robin scheduler with a quantum size of 2, a preemptive SJF, and FCFS scheduler.
- (b) What is the average waiting time for each?

Consider the following set of tasks:

Task	Arrival time	Computation time	Period
T_1	10	10	30
T_2	0	15	40
T_3	0	5	50

- (a) Compute the utilization. Is this guaranteed to be schedulable using RMS? EDF?
- (b) Show the task arrival times and schedule with RMS and EDF as we did in class.