Quiz 1, Spring 2022 Solutions

(10) Question 1.

a) Once

D) Connect to access point

B) Call DNS

b) every 1 second

A) Create a socket (allocates a data structure from the operating system)

C) Connect to server using socket

H) Send a TCP message

E) Receive TCP message

F) Close Socket (returns socket to operating system)

Alternate possibility (it does not work this way, but this sequence occurs for UDP)

a) Once

D) Connect to access point

B) Call DNS

A) Create a socket (allocates a data structure from the operating system)

C) Connect to server using socket

b) every 1 second

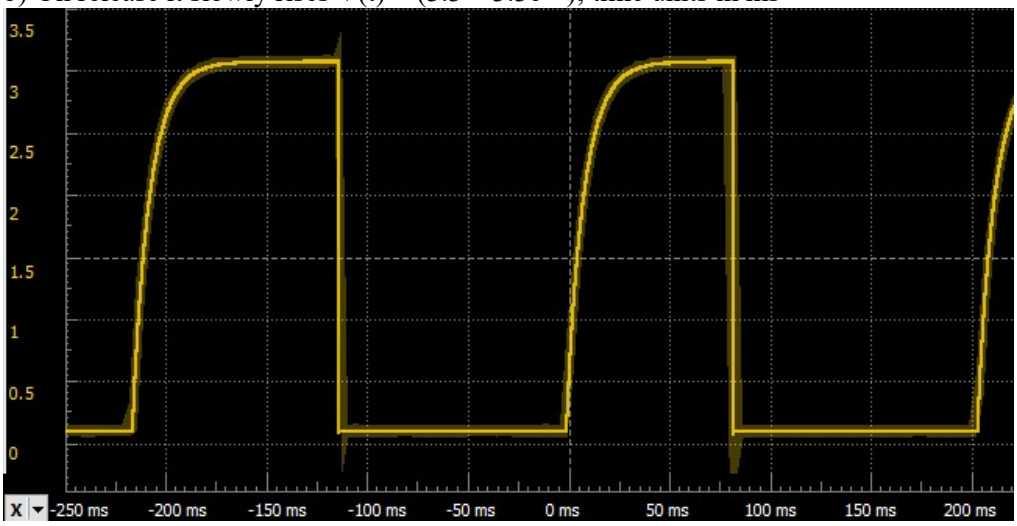
H) Send a TCP message

E) Receive TCP message

(10) Question 2. Time constant τ = 1E5 * 1E-7 = 1E-2 = 10ms

a) On touch it instantly falls from 3.3V to 0V (it will spark across the switch); time units in ns. If you build this, place a 100 ohm resistance is series with the switch so 3.3V to 0 has a time constant of τ = 1E2 * 1E-7 = 1E-5 = 10us

b) On release it slowly rises $V(t) = (3.3 - 3.3e^{-t/\tau})$; time units in ms



(15) Question 3. Disconnect the jumper across H24/H25 and place the DVM in current mode across the two pins. This is exactly what you were asked to do in multiple labs.

(10) Question 4. Set the priority of Timer0A equal to the priority of Timer1A. Set the priority of Timer2A lower than the other two (larger priority value)

It is not a good idea, but you could disable interrupts at the start of Timer0A and 1A and reenable at end.

(10) Question 5. $dB_{FS} = 20 \log_{10}(V/3.3)$. Two factors affect SNR:

1) 12-bit DAC to create an analog sine wave.
2) the size of the table containing one period of the wave is 256 elements

The 12-bit DAC should have produced $20 \log_{10}(4096) = 72.2$ dB
But the system is restricted to the table size $20 \log_{10}(256) = 48.2$ dB

(10) Question 6. Loudness is a function of the power delivered to the 8-ohm speaker. $P = V^2/R$.
Determine the voltage across the speaker
Left $V_{left} = (V_{CC} - V_{DS})$
Right $V_{right} = (V_{BUS} - V_{DS})$
Increase $(V_{BUS} - V_{DS})^2 / (V_{CC} - V_{DS})^2$
I didn't ask for it, but $(5 - 0.5)^2 / (3.3 - 0.5)^2 = 2.58$!

(35) Question 7. Same idea as the EdgeInterruptDebounce_4C123 project showed in lecture 10.
Part a) Show the ritual to initialize this system. You may add global variables. Do not worry about priority. Assume a 16 MHz bus clock on the TM4C123.

```
void Init(void){
  SYSCTL_RCGCGPIO_R |= 0x01;
  NVIC_ST_RELOAD_R = 3999; // (250us/62.5ns = 16MHz/4kHz)-1
  NVIC_ST_CTRL_R = 0x05;   // initially off
  NVIC_ST_CURRENT_R = 0;
  GPIO_PORTA_DIR_R &= ~0x80;
  GPIO_PORTA_DIR_R |= 0x40;
  GPIO_PORTA_DEN_R |= 0xC0;
  GPIO_PORTA_IS_R &= ~0x80;
  GPIO_PORTA_IBE_R |= 0x80; // both edges
  GPIO_PORTA_IEV_R =0;      // this does not matter
  GPIO_PORTA_ICR_R = 0x80;
  GPIO_PORTA_IM_R = 0x80; // arm PA7
  NVIC_EN0_R = 1;
  EnableInterrupts();
}
```

Part c) Show the **GPIOPortA_Handler** interrupt service routine.

```
void GPIOPortA_Handler(void){
  GPIO_PORTA_ICR_R = 0x80;     // ack
  GPIO_PORTA_DATA_R &= ~0x40;  // goes low on every interrupt
  NVIC_ST_RELOAD_R = 3999; // optional (not really needed)
  NVIC_ST_CURRENT_R = 0;   // restart, clear COUNT flag
  NVIC_ST_CTRL_R = 0x07;   // arm SysTick
}
```

Part b) Show the SysTick interrupt service routine. No **for**, **while**, or **do-while** loops are allowed.

```
void SysTick_Handler(void){
  GPIO_PORTA_DATA_R |= 0x40;  // goes high 250us later
  NVIC_ST_CTRL_R = 0x05;   // disarm SysTick
}
```

Later in the course we will learn how to use input capture to measure period. We could measure period of PA7 for the previous wave (**Period**) in bus cycles and change the 3999 constant to (**Period**/4)-1