Jonathan W. Valvano First:_____ Last:_____

October 10, 2019, 3:30-4:45pm. This is a closed book exam, with one 8.5 by 11-inch crib sheet. You have 75 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting***.

(5) Question 1. Consider the IoT system from Lab 4 that communicates between the TM4C123 and the Blynk app on the phone. For this system, explain in one sentence how **sockets** are used within the ESP8266.

(5) Question 2. Consider the IoT system from Lab 4 that communicates between the TM4C123 and the Blynk app on the phone. For this system, explain in one sentence how **virtual pins** are used.

(5) Question 3. Consider an ideal capacitor, with capacitance *C*. Give the **differential equation** that relates capacitor voltage to capacitor current.

(5) Question 4. You are asked to modify an embedded system to lower the noise on the 3.3V power line. Describe how you would **measure power line noise** to determine if your changes were successful in reducing noise. Be as explicit as possible, including both test equipment and mathematical relations.

(5) Question 5. Consider a producer consumer system that uses a 16-element FIFO to pass data from the producer to the consumer. The producer is software, creating data, putting the data into the FIFO. The consumer is an interrupt-driven **output** device, which gets data from the FIFO and **outputs** it. We have added debugging instruments to the system to determine its status. After every time data is put into the FIFO, we measure the number of elements in the FIFO, and we create a probability mass function (pmf). Roughly sketch the **pmf graphs** illustrating the two cases of I/O bound and CPU bound. Label the axes.

I/O bound	CPU bound

(5) Question 6. Consider the interaction between this ISR and this main program.

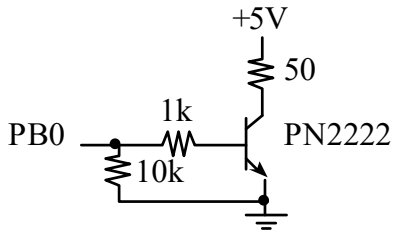
```
void SysTickHander(void){
  static uint32_t counter;
  counter++;
// other stuff
}
```

```
void main(void){
  static uint32_t counter;
  Init();
  while(1){
    counter++;
// other stuff
  }
}
```

Do these read modify write accesses create a **critical section**? Answer yes or no. If no, justify your answer. If yes, describe how you would remove the bug.

(15) Question 7. Consider this interface between the microcontroller pin PB0 and a 50-ohm speaker:
You may assume

$V_{BE(sat)}$ = 0.8 V,
$I_{C(max)}$ = 500 mA,
$V_{CE(sat)}$ = 0.5 V,
$h_{FE(max)}$ = 100



You will answer the same three questions for each of three scenarios:
Scenario 1) Assume the PB0 is uninitialized with DEN=0

In which mode is the BJT? Circle one of these four possibilities: ……

- Cutoff (off),
- Forward active (linear),
- Saturated (fully on), or
- Reversed active

What is the current out of the TM4C123 PB0 pin?

What is the current across the 50-ohm speaker?

Scenario 2) Assume the PB0 is initialized as an output with DEN=1, DIR=1, DR8R=1, but DATA=0.

In which mode is the BJT? Circle one of these four possibilities: ……

- Cutoff (off),
- Forward active (linear),
- Saturated (fully on), or
- Reversed active

What is the current out of the TM4C123 PB0 pin?

What is the current across the 50-ohm speaker?

Scenario 3) Assume the PB0 is initialized as an output with DEN=1, DIR=1, DR8R=1, and DATA=1.

In which mode is the BJT? Circle one of these four possibilities: ……

- Cutoff (off),
- Forward active (linear),
- Saturated (fully on), or
- Reversed active

What is the current out of the TM4C123 PB0 pin?

What is the current across the 50-ohm speaker?

(20) Question 8. You are given two tasks: **`Task20()`** should be executed every 20 ms and **`Task25()`** should be executed every 25 ms. The maximum time to execute either task is 1ms. Minimize the jitter on executing **`Task20()`**. This means **`Task20()`** is never delayed by the running any other software. Assume the PLL is active, making the bus clock 12.5 ns (80 MHz). You may assume there are other interrupts, but **`Task20`** is most important.

Part a) Show the ritual to initialize this system. You will use SysTick interrupts. You may add global variables.

```
void SysTick_Init(void){
  NVIC_ST_RELOAD_R =                  ;
  NVIC_ST_CTRL_R =                    ;
  NVIC_SYS_PRI3_R =
     (NVIC_SYS_PRI3_R&                )|                  ;
  EnableInterrupts();
}
```

Part b) Show the SysTick interrupt service routine. No **for**, **while**, or **do-while** loops are allowed. You do not write **`Task20()`** or **`Task25()`**, just call them from the ISR.

Part c) There should be no jitter on **`Task20`**, but what is the worst case jitter on **`Task25`**.

(5) Question 9. We will store the value -8 cm with the integer -64 and store the value +8 cm with the value +64. Assuming the integer is stored as an 8-bit signed number, what are the minimum, maximum, precision and resolution of this fixed point number system? Give units for each.

Minimum value =	Precision =
Maximum value =	Resolution of the value =

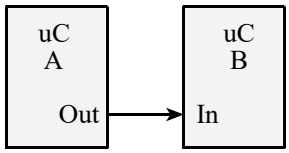
(10) Question 10. These are the parameters of the GPIO pins on ***microcontroller A***:

I_{OL} = 1mA, I_{OH} = 1mA, I_{IL} = 1µA, I_{IH} = 1µA,
V_{OL} = 0.3V, V_{OH} = 2.7V, V_{IL} = 0.5V, V_{IH} = 2.0 V

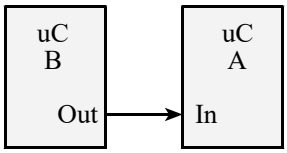
These are the parameters of the GPIO pins on ***microcontroller B***:

I_{OL} = 4mA, I_{OH} = 4mA, I_{IL} = 20µA, I_{IH} = 20µA,
V_{OL} = 0.7V, V_{OH} = 3.2V, V_{IL} = 1.0V, V_{IH} = 2.5 V

Part a) Can you directly connect a GPIO output from microcontroller A to a GPIO input on microcontroller B? If yes, prove it. If no, show at least one parameter/equation not satisfied.



Part b) Can you directly connect a GPIO output from microcontroller B to a GPIO input on microcontroller A? If yes, prove it. If no, show at least one parameter/equation not satisfied.



(5) Question 11. Give the definition and one example of a **firm real-time system**.

(15) Question 12. Make PB1 an input and PB2 an output. PB1 will be an input squarewave of unknown frequency f, and PB2 will be an output squarewave of frequency f/2. Basically, you will toggle PB2 on every rising edge of PB1.

Part a) Show the ritual to initialize this system. Fill in the boxes with a value. It need not be friendly.

```
void PortB_Init(void){ uint32_t volatile delay;
  SYSCTL_RCGCGPIO_R |= 0x02;

  delay = SYSCTL_RCGCGPIO_R;
  GPIO_PORTB_DIR_R  =  [          ] ;
  GPIO_PORTB_DEN_R  =  [          ] ;
  GPIO_PORTB_IS_R   =  [          ] ;
  GPIO_PORTB_IBE_R  =  [          ] ;
  GPIO_PORTB_IEV_R  =  [          ] ;
  GPIO_PORTB_IM_R   =  [          ] ;
  NVIC_PRI0_R = (NVIC_PRI0_R&0xFFFF00FF)|0x00008000; // priority 4
  NVIC_EN0_R =         [          ] ;
  EnableInterrupts();
}
```

Part b) Show the interrupt service routine (Port B handler). No backward jumps are allowed.

```
void GPIOPortB_Handler(void){
```

Parameters for the TM4C123 microcontroller (with 8mA mode selected)
I_{OL} = 8mA, I_{OH} = 8mA, I_{IL} = 2μA, I_{IH} = 2μA,
V_{OL} = 0.4V, V_{OH} = 2.4V, V_{IL} = 1.3V, V_{IH} = 2.0 V

7	6	5	4	3	2	1	0	Name
DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTB_DATA_R
DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTB_DIR_R
IS	IS	IS	IS	IS	IS	IS	IS	GPIO_PORTB_IS_R
IBE	IBE	IBE	IBE	IBE	IBE	IBE	IBE	GPIO_PORTB_IBE_R
IEV	IEV	IEV	IEV	IEV	IEV	IEV	IEV	GPIO_PORTB_IEV_R
IME	IME	IME	IME	IME	IME	IME	IME	GPIO_PORTB_IM_R
RIS	RIS	RIS	RIS	RIS	RIS	RIS	RIS	GPIO_PORTB_RIS_R
MIS	MIS	MIS	MIS	MIS	MIS	MIS	MIS	GPIO_PORTB_MIS_R
ICR	ICR	ICR	ICR	ICR	ICR	ICR	ICR	GPIO_PORTB_ICR_R
SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTB_AFSEL_R
DRV2	DRV2	DRV2	DRV2	DRV2	DRV2	DRV2	DRV2	GPIO_PORTB_DR2R_R
DRV4	DRV4	DRV4	DRV4	DRV4	DRV4	DRV4	DRV4	GPIO_PORTB_DR4R_R
DRV8	DRV8	DRV8	DRV8	DRV8	DRV8	DRV8	DRV8	GPIO_PORTB_DR8R_R
ODE	ODE	ODE	ODE	ODE	ODE	ODE	ODE	GPIO_PORTB_ODR_R
PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTB_PUR_R
PDE	PDE	PDE	PDE	PDE	PDE	PDE	PDE	GPIO_PORTB_PDR_R
SLR	SLR	SLR	SLR	SLR	SLR	SLR	SLR	GPIO_PORTB_SLR_R
DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTB_DEN_R
CR	CR	CR	CR	CR	CR	CR	CR	GPIO_PORTB_CR_R
AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	GPIO_PORTB_AMSEL_R

IS=0 means edge, IS=1 means level
IBE=1 means both, IBE=0 means one
If IBE=0, IEV=1 means rising, IEV=0 means falling

Address	31 – 29	23 – 21	15 – 13	7 – 5	Name
0xE000E400	GPIO Port D	GPIO Port C	GPIO Port B	GPIO Port A	NVIC_PRI0_R
0xE000E404	SSI0, Rx Tx	UART1, Rx Tx	UART0, Rx Tx	GPIO Port E	NVIC_PRI1_R
0xE000E408	PWM Gen 1	PWM Gen 0	PWM Fault	I2C0	NVIC_PRI2_R
0xE000E40C	ADC Seq 1	ADC Seq 0	Quad Encoder	PWM Gen 2	NVIC_PRI3_R
0xE000E410	Timer 0A	Watchdog	ADC Seq 3	ADC Seq 2	NVIC_PRI4_R
0xE000E414	Timer 2A	Timer 1B	Timer 1A	Timer 0B	NVIC_PRI5_R
0xE000E418	Comp 2	Comp 1	Comp 0	Timer 2B	NVIC_PRI6_R
0xE000E41C	GPIO Port G	GPIO Port F	Flash Control	System Control	NVIC_PRI7_R
0xE000ED20	SysTick	PendSV	--	Debug	NVIC_SYS_PRI3_R

Address		30		19		6	5	4	3	2	1	0	Name
0xE000E100		F		Timer0A		UART1	UART0	E	D	C	B	A	NVIC_EN0_R
0xE000E104											UART2		NVIC_EN1_R

Address	31-24	23-17	16	15-3	2	1	0	Name
$E000E010	0	0	COUNT	0	CLK_SRC	INTEN	ENABLE	NVIC_ST_CTRL_R
$E000E014	0	24-bit RELOAD value						NVIC_ST_RELOAD_R
$E000E018	0	24-bit CURRENT value of SysTick counter						NVIC_ST_CURRENT_R