Jonathan W. Valvano First:_____ Last:_____

October 9, 2015, 10:00am-10:50am. This is a closed book exam, with one 8.5 by 11 inch crib sheet. You have 50 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting***.

(10) Question 1. Consider the use of a **probability mass function** to describe the noise processes involved in ADC sampling. Assume there is enough ADC noise such that repeated sampling of the same input voltage doesn't always yield the same digital result. Draw a realistic probability mass function assuming the input is a fixed constant of 1.65 V and now hardware averaging. The sampling rate is 1000 Hz. The ADC is 12 bits and has a range of 0 to 3.3V. Label both axes including units.

(10) Question 2. Consider the following two ISRs that are running at the same priority level. You may assume there are no other accesses to Port A. Assume PA4 and PA5 are outputs connected to a scope.

```
void ISR1(void){
  GPIO_PORTA_DATA_R ^= 0x10;
// other stuff
}
```

```
void ISR2(void){
  GPIO_PORTA_DATA_R ^= 0x20;
// other stuff
}
```

Do these read-modify-write accesses to Port A create a critical section? If yes, modify the code to correct the error. If no, justify why these accesses are not critical.

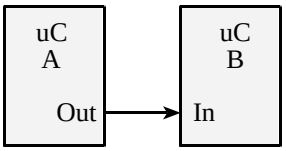
(10) Question 3. These are the parameters of the GPIO pins on ***microcontroller A***:

I_{OL} = 1mA, I_{OH} = 1mA, I_{IL} = 1μA, I_{IH} = 1μA,
V_{OL} = 0.3V, V_{OH} = 2.5V, V_{IL} = 0.5V, V_{IH} = 2.0 V

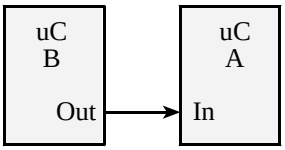
These are the parameters of the GPIO pins on ***microcontroller B***:

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} = 3.0 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.



(40) Question 4. The goal is to increment **`Counter`** every time SW1 is pressed and decrement the same **`Counter`** every time SW2 is pressed. Assume the switch bounces, but the bounce time is less than 10 ms. Also assume each switch is pressed for at least 25 ms when touched, and the release time is also greater than 25 ms. The **`Counter`** is a shared global. Use SysTick periodic interrupts for this solution, where once the initialization is called the main program is free to run other unrelated operations. You must debounce the switches using SysTick interrupts. You cannot use any of the edge-triggered features of Port A. Assume the bus clock is 16 MHz.

`int32_t Counter; // difference between touches on SW1 and touches on SW2`

Part a) Interface SW1 to PA7 and SW2 to PA6. Implement the hardware such that a switch touch results in a zero input. Draw the circuit including resistors and capacitors as needed. Minimize cost.

Part b) Show the ritual to initialize this system. Initialize both Port A GPIO, and SysTick interrupts. Define additional variables as needed. You need not set the priority register, but you should clear the I bit. Initialize **`Counter`** to zero. Do not include a main program.

Part c) Show the SysTick interrupt service routine. No backward jumps are allowed.

```
void SysTick_Handler(void){
```

(10) Question 5. Assume that PA2 is a squarewave output at 1000 Hz. Interface a 60-ohm speaker using +3.3V power so the speaker generates a loud 1000 Hz tone. Label all parts. For the necessary resistor(s) show the equations needed to determine the value(s).

(5) Question 6. What is the purpose of the DNS?

(5) Question 7. What is the difference between UDP and TCP communication? More specifically when should we use UDP and when should we use TCP?

(10) Question 8. Let *N1 N2 N3* be the values of three binary fixed-point numbers each with a resolution of 1/16 Assume **`I1`**, **`I2`**, and **`I3`** are the corresponding integer parts. Write the body of the function that implements fixed-point divide, *N3 = N2/N1*. Minimize dropout, but don't worry about overflow.

```
uint32_t Divide(uint32_t I1, uint32_t I2){ uint32_t I3;




  return  I3;
}
```

Parameter	PN2222 (I_C=150mA) PN2907 (I_C=150mA)	2N2222 (I_C=500mA) 2N2907 (I_C=500mA)	TIP120 (I_C=3A) TIP125 (I_C=3A)
$\mathbf{h_{fe}}$	100	40	1000
$\mathbf{V_{BEsat}}$	0.6	2	2.5 V
$\mathbf{V_{CE}}$ at saturation	0.3	1	2 V

Parameters for the TM4C123 microcontroller (no 12mA mode will be used)

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_PORTA_DATA_R
DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTA_DIR_R
IS	IS	IS	IS	IS	IS	IS	IS	GPIO_PORTA_IS_R
IBE	IBE	IBE	IBE	IBE	IBE	IBE	IBE	GPIO_PORTA_IBE_R
IEV	IEV	IEV	IEV	IEV	IEV	IEV	IEV	GPIO_PORTA_IEV_R
IME	IME	IME	IME	IME	IME	IME	IME	GPIO_PORTA_IM_R
RIS	RIS	RIS	RIS	RIS	RIS	RIS	RIS	GPIO_PORTA_RIS_R
MIS	MIS	MIS	MIS	MIS	MIS	MIS	MIS	GPIO_PORTA_MIS_R
ICR	ICR	ICR	ICR	ICR	ICR	ICR	ICR	GPIO_PORTA_ICR_R
SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTA_AFSEL_R
DRV2	DRV2	DRV2	DRV2	DRV2	DRV2	DRV2	DRV2	GPIO_PORTA_DR2R_R
DRV4	DRV4	DRV4	DRV4	DRV4	DRV4	DRV4	DRV4	GPIO_PORTA_DR4R_R
DRV8	DRV8	DRV8	DRV8	DRV8	DRV8	DRV8	DRV8	GPIO_PORTA_DR8R_R
ODE	ODE	ODE	ODE	ODE	ODE	ODE	ODE	GPIO_PORTA_ODR_R
PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTA_PUR_R
PDE	PDE	PDE	PDE	PDE	PDE	PDE	PDE	GPIO_PORTA_PDR_R
SLR	SLR	SLR	SLR	SLR	SLR	SLR	SLR	GPIO_PORTA_SLR_R
DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTA_DEN_R
CR	CR	CR	CR	CR	CR	CR	CR	GPIO_PORTA_CR_R
AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	GPIO_PORTA_AMSEL_R

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
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

	31–3	2–0	Name
$4003.0000		**GPTMCFG**	TIMER0_CFG_R

	31–4	3	2	1-0	
$4003.0004		**TAAMS**	**TACMR**	**TAMR**	TIMER0_TAMR_R

	14	13	11-10	8	6	5	3-2	0	
$4003.000C	TBPWML	TBOTE	TBEVENT	TBEN	TAPWML	TAOTE	**TAEVENT**	**TAEN**	TIMER0_CTL_R

	31-11	10	9	8	7-4	2	1	0	
$4003.0018		CBEIM	CBMIM	TBTOIM		**CAEIM**	CAMIM	TATOIM	TIMER0_IMR_R

	31-11	10	9	8	7-4	2	1	0	
$4003.001C		CBERIS	CBMRIS	TBTORIS		**CAERIS**	CAMRIS	TATORIS	TIMER0_RIS_R

	31-11	10	9	8	7-4	2	1	0	
$4003.0020		CBEMIS	CBMMIS	TBTOMIS		CAEMIS	CAMMIS	TATOMIS	TIMER0_MIS_R

	31-11	10	9	8	7-4	2	1	0	
$4003.0020		CBECINT	CBMCINT	TBTOCINT		**CAECINT**	CAMCINT	TATOCINT	TIMER0_ICR_R

	31–16	15–0	
$4003.0028	TAILRH	**TAILRL**	TIMER0_TAILR_R

	31–16	15–0	
$4003.0030	TAMRH	TAMRL	_TAMATCHR_R

	31–8	7-0	
$4003.0038		TAPSR	TIMER0_TAPR_R

	31–8	7-0	
$4003.0040		TAPSMR	TIMER0_TAPMR_R

	31–16	15-0	
$4003.0048	TARH	**TARL**	TIMER0_TAR_R