Jonathan W. Valvano First:__________________ Last:______________ EID:__________

This is the closed book section. Calculator is allowed (no laptops, phones, devices with wireless communication). You must put your answers in the boxes. When you are done, you turn in the closed-book part and can start the open book part.

(5) Question 1. For each description, choose the best voltage regulator that matches the description. Let (V_{in}, I_{in}) be the input (voltage, current). Let (V_{out}, I_{out}) be the output (voltage, current). In each case, you may assume the output, V_{out}, is constant. Put one letter, A – D, into each box. You may choose an answer more than once or not at all.

A) Linear regulator like the LM2937, LP2950, or 78M05
B) Buck regulator
C) Boost regulator
D) None of the above

(1) Part a) A □ creates an output current, I_{out}, that is constant.

(1) Part b) The input current of a □ is approximately equal to the output current $I_{in} \approx I_{out}$.

(1) Part c) A □ converts DC to AC, uses a transformer to increase the voltage, then converts AC to DC, so the V_{out} is a constant.

(1) Part d) A □ uses a switching network, a diode, and an inductor to increase the voltage ($V_{in} < V_{out}$). It is very power efficient, $V_{in}*I_{in} \approx V_{out}*I_{out}$.

(1) Part e) A □ separates the grounds, so that the ground from the input voltage is not connected to the ground of the output voltage.

(5) Question 2. For each sentence, choose the best device that matches. Put one letter, A – E, into each box. You may choose an answer more than once or not at all.

A) Resistor
B) Ceramic capacitor
C) Tantalum capacitor
D) Inductor
E) None of the above

(1) Part a) A □ is polarized, meaning it only works with current in one direction.

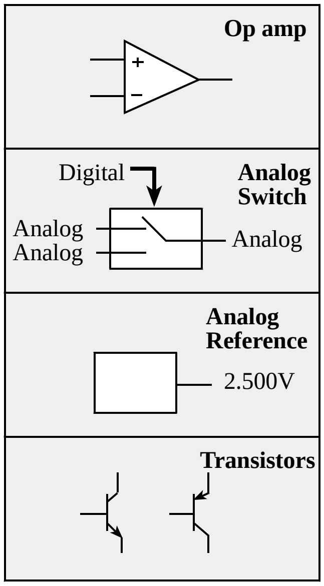
(1) Part b) A □ is used with the crystal to create a high-precision clock.

(1) Part c) We add a □ to the motor interface circuit to remove back EMF.

(1) Part d) We use a □ in a solid state relay interface circuit to set the desired current.

(1) Part e) Power dissipated in a □ is a linear function of the slope of the current through the device versus the voltage across it.

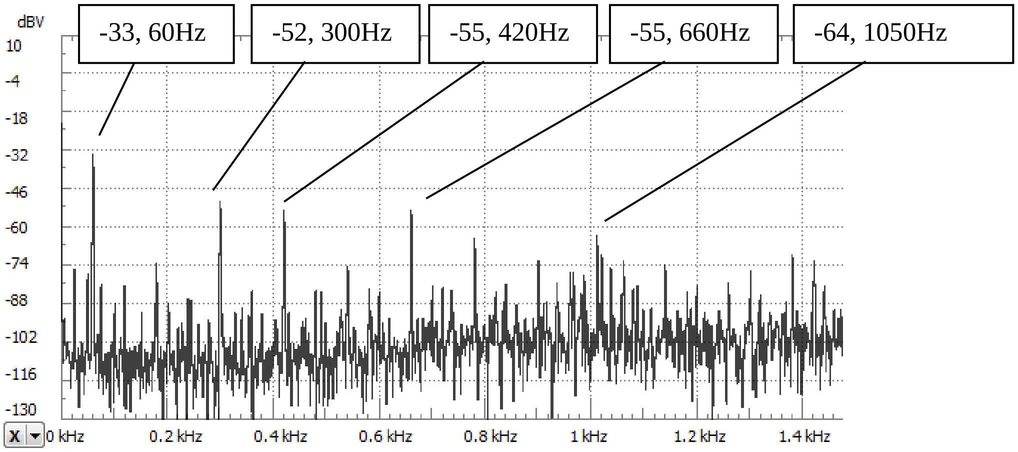
(10) Question 3. Show the circuit for a **3-bit resistor-string DAC.** You may use resistors and capacitors without showing their values. You may select any of the components from the menu on the right. There are three digital inputs and one analog output. You can make the output range whatever you wish. Show enough detail so the basic theory of operation is apparent. Not all components may be needed.



(5) Question 4. What does the Civil Rights Act of 1964 have in common with the IEEE Code of Ethics?

(5) Question 5. The goal of modular design is to maximize the number of software modules while **minimizing the coupling** between modules. One obvious measure of coupling is the amount of data flowing between the modules; this is categorized as bandwidth coupling. Describe two other different types of coupling that should also be minimized.

(5) Question 6. Consider a real-time data acquisition system with an 8-bit ADC sampled at f_S. The following data were sampled at the input of the ADC. The desired signal exists in the 0 to 10 Hz range, and the rest of what you see in this spectrum is noise. Yes, it needs an analog filter. However, if you were to sample this signal exactly like this, what is the slowest sampling rate f_s allowed that will prevent aliasing? Show your work. In particular, calculate the ADC resolution in dB_{FS} and draw it as a horizontal line on this graph.



(10) Question 7. The FSM is initialized in main and the FSM controller runs in the SysTick handler.

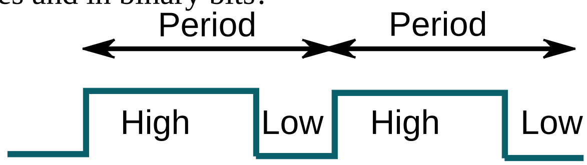
```
int main(void){
  Robot_Init(); // Init F, SysTick
  S = Center;   // initial state
  EnableInterrupts();
  while(1){
    GPIO_PORTF_DATA_R ^= 0x08;
    Body();     // other stuff
  }
}
```

```
uint32_t S;  // index to current state
void SysTick_Handler(void){
uint32_t input;   // state input: 0,1,2,3
  input = (GPIO_PORTF_DATA_R&0x01)+
    ((GPIO_PORTF_DATA_R&0x10)>>3);
  S = fsm[S].Next[input]; // next
  GPIO_PORTF_DATA_R = (fsm[S].Out<<1)+
    (GPIO_PORTF_DATA_R&~0x06);
}
```

(5) Part a) Does this usage create a critical section? Select: critical or not critical

(5) Part b) If critical, fix the bug. If not critical, prove it has no bug.

(5) Problem 8. A PWM system uses an 8 MHz clock to generate a 1 kHz wave. What is the **precision** of the PWM system in alternatives and in binary bits?



Jonathan W. Valvano First:_________________ Last:_____________ EID:__________

Open book, open notes, calculator (no laptops, phones, devices with screens larger than a TI-89 calculator, devices with wireless communication). You must put your answers on these pages. Please don't turn in any extra sheets.

(10) Question 9. This system (Program 6.5) has no bugs, it measures pulse width of the input signal connected to both **PB6** and **PB7**, placing the 24-bit measurement in **PW** and setting the **Done** flag. You are asked to change the software to use **PF0** and **PF1** (the input is now connected to both **PF0** and **PF1)**. Cross out parts of the code you wish to delete and insert necessary additions.

```
uint32_t PW;     // 24 bits, 12.5 ns units
int Done;        // set each falling
void PWMeasure2_Init(void){
  SYSCTL_RCGCTIMER_R |= 0x01;
  SYSCTL_RCGCGPIO_R |= 0x02;
  Done = 0;
  GPIO_PORTB_DIR_R &= ~0xC0;
  GPIO_PORTB_DEN_R |= 0xC0;
  GPIO_PORTB_AFSEL_R |= 0xC0;
  GPIO_PORTB_PCTL_R = (GPIO_PORTB_PCTL_R&0x00FFFFFF)+0x77000000;
  TIMER0_CTL_R &= ~0x00000003;
  TIMER0_CFG_R = 0x00000004;
  TIMER0_TAMR_R = 0x00000007;
  TIMER0_CTL_R = (TIMER0_CTL_R&(~0x0C))+0x04;
  TIMER0_TAILR_R = 0x0000FFFF;
  TIMER0_TAPR_R = 0xFF;
  TIMER0_IMR_R |= TIMER_IMR_CAEIM;
  TIMER0_ICR_R = TIMER_ICR_CAECINT;
  TIMER0_TBMR_R = 0x00000007;
  TIMER0_CTL_R = (TIMER0_CTL_R&(~0x0C00))+0x00;
  TIMER0_TBILR_R = 0x0000FFFF;
  TIMER0_TBPR_R = 0xFF;
  TIMER0_IMR_R &= ~0x700;
  TIMER0_CTL_R |= 0x00000003;
  NVIC_PRI4_R = (NVIC_PRI4_R&0x00FFFFFF)|0x40000000;
  NVIC_EN0_R = 1<<19;
  EnableInterrupts();}
void Timer0A_Handler(void){
  TIMER0_ICR_R = 0x00000004;
  PW = (TIMER0_TBR_R-TIMER0_TAR_R)&0x00FFFFFF;
  Done = 1;}
```

(10) Problem 10. You will implement a PID controller that runs as a SysTick background thread at 1 kHz. There is a state estimator function that returns the current speed as an integer in rps (ranges from 0 to 1000 rps), and an actuator function that sets the power to the motor as an integer in mW (ranges from 0 to 1000 mW). These two functions are given (prototypes below), and you do not need to write them.

```
int16_t CurrentSpeed(void);   // current speed in rps
void SetPower(int16_t power); // apply power to the motor in mW
```

There is a shared global variable containing the desired speed in rps (ranges from 100 to 500 rps).

```
int16_t Xstar=100;            // desired speed in rps (may change)
```

Theoretically, we define the setpoint as *X** in rps, and the state estimation as *X'* in rps. Calculate error

$E(t)=X^*(t)- X'(t)$

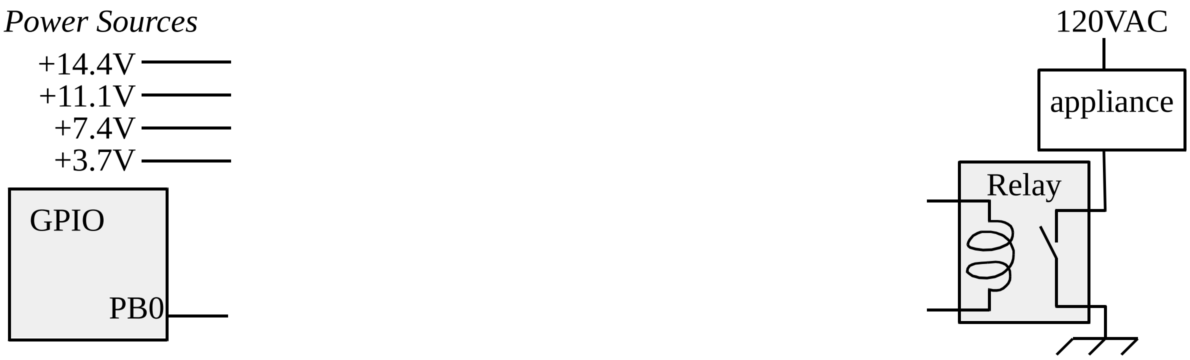
Theoretically, we define the actuator as *U* in mW, and the goal of the PID controller is to implement this control equation every 1 ms.

$$U(t) = K_p E(t) + \int_0^t K_i E(\tau) d\tau + K_d \frac{dE(t)}{dt}$$

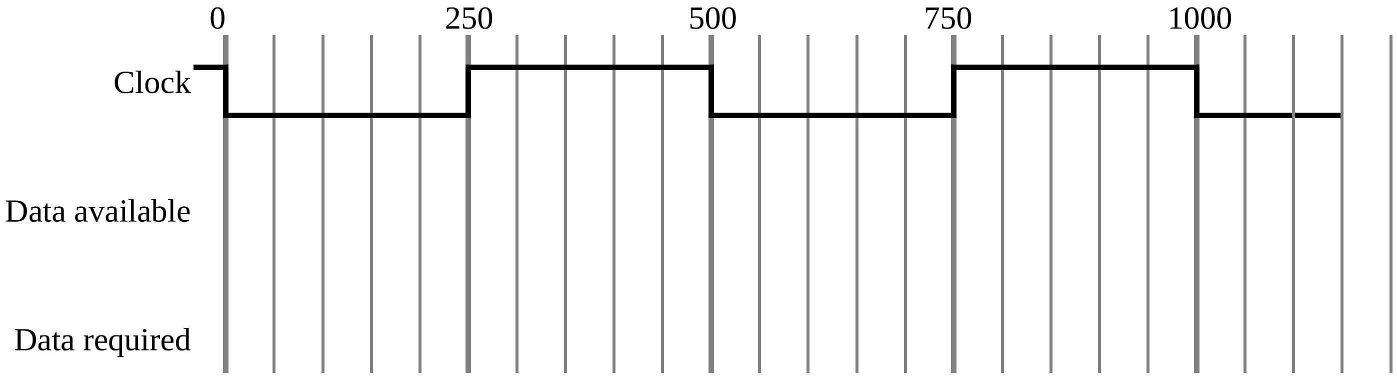
where K_p = 0.1 mW/rps, K_i = 0.25 mW/rps/ms, and K_d = 0.01 mW-ms/rps. You do not need to implement antireset windup. You may assume SysTick is configured for 1 kHz interrupts, and do not need to show I/O initialization or SysTick initialization, just show the SysTick ISR that implements the PID controller. Please do not use floating point. The essence of this question is to implement integration and differentiation in software operated on sampled digital data.

```
void SysTick_Handler(void){ // Executed every 1ms
```

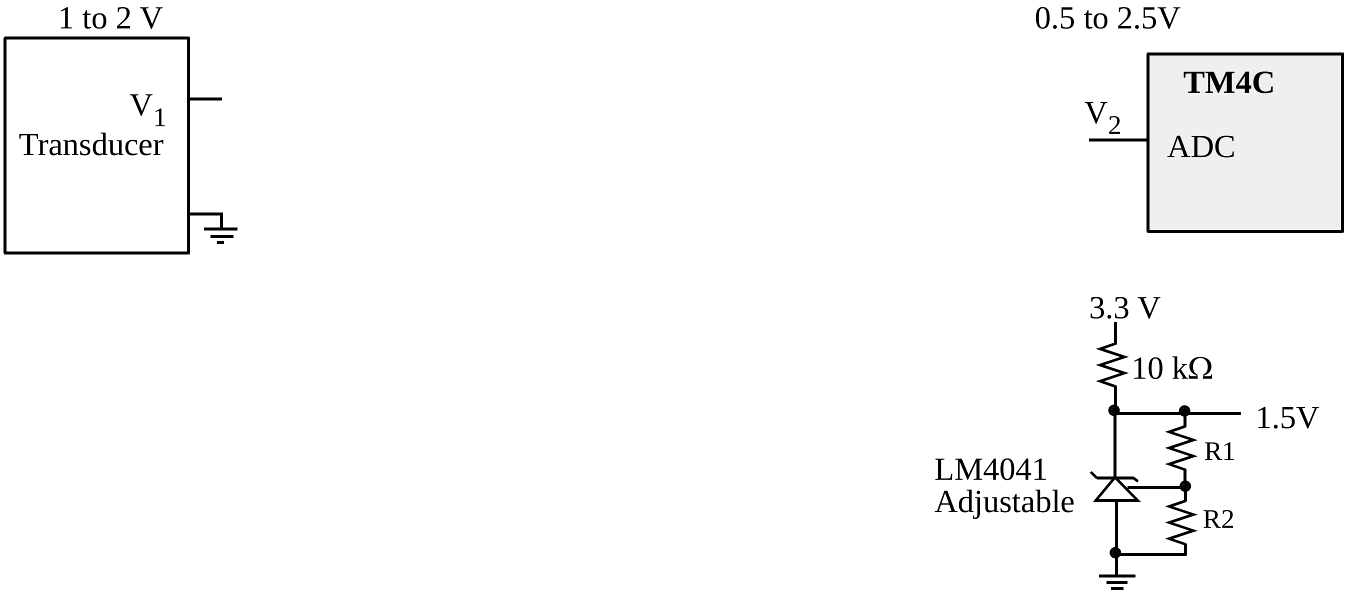
(10) Question 11. The Li-Ion battery cell voltage is 3.7V. Using multiple cells, we can create a power source at integer multiples of 3.7V. Interface this electromagnetic relay to the microcontroller. To activate, the relay needs anywhere from 6 to 8 V at 600 mA. Include protection against back EMF. Label part numbers for all interface components and resistor values. If you use NPN or PNP transistors, assume the V_{CE} is 0.5V and the V_{BE} is 1V. You must select which Li-Ion battery to use.



(5) Question 12. Consider a simplex synchronous serial interface passing data from slave to master. The master clock is 50% duty cycle 2 MHz *Clock*. The master shifts data in on the rising edge of the *Clock*. The **master hold** time is 50 ns and the **setup time** is 100 ns. The slave shifts data out on the falling edge of the *Clock*. The maximum **propagation delay** from *Clock* to slave data output is 200 ns. *Complete the timing diagram to scale showing data available and data required timing, proving that this interface does not work.* Show the transfer of one bit (not the entire frame).



(15) Question 13. Interface this transducer to the ADC. The information is encoded as V_1, and it is relative to ground. The transducer output ranges from 1 to 2V, in other words, $1 \le V_1 \le 2$. The goal of the circuit is to convert the input into an analog signal at the ADC that ranges as $0.5 \le V_2 \le 2.5$V. The signals of interest are 0 to 1 kHz. The software sampling rate will be 5 kHz. There is a large unwanted noise signal at 22 kHz. Please include an antialiasing low pass filter (f_c approximately equal to 2.5 kHz). Show all resistors, capacitors, and chip numbers, choosing standard resistor and capacitor values. The available power supply voltage is 3.3V. Assume R1 and R2 are already chosen to achieve a reference of 1.5V.



Standard values for 5% resistors range from 10 Ω to 22 MΩ. We can multiply a number in Table 1 by powers of 10 to select a standard value 5% resistor. For example, if we need a 25 kΩ 5% resistor, the closest number is 24*1000, or 24 kΩ. Table 2 shows standard capacitor values.

10	11	12	13	15	16	18	20	22	24	27	30
33	36	39	43	47	51	56	62	68	75	82	91

Table 1. Standard resistor values for 5% tolerance

10pF	100pF	1000pF	0.010µF	0.10µF	1.0µF	10µF
12pF	120pF	1200pF	0.012µF	0.12µF	1.2µF	
15pF	150pF	1500pF	0.015µF	0.15µF	1.5µF	
18pF	180pF	1800pF	0.018µF	0.18µF	1.8µF	
22pF	220pF	2200pF	0.022µF	0.22µF	2.2µF	22µF
27pF	270pF	2700pF	0.027µF	0.27µF	2.7µF	
33pF	330pF	3300pF	0.033µF	0.33µF	3.3µF	33µF
39pF	390pF	3900pF	0.039µF	0.39µF	3.9µF	
47pF	470pF	4700pF	0.047µF	0.47µF	4.7µF	47uF
56pF	560pF	5600pF	0.056µF	0.56µF	5.6µF	
68pF	680pF	6800pF	0.068µF	0.68µF	6.8µF	
82pF	820pF	8200pF	0.082µF	0.82µF	8.2µF	

Table 2. Standard capacitor values for 10% tolerance