

Jonathan W. Valvano First Name: _____ Last Name: _____
 December 9, 1999, 9 am to 12 noon

This is an open book, open notes exam. You may put answers on the backs of the pages, but please don't turn in any extra sheets.

(15) Question 1. In this problem you will design a frequency meter using input capture, output compare, and interrupt synchronization. The digital signal is connected to PT1. You will also use the timer channel 5. The range of frequencies is 0 to 1000 Hz, and the desired frequency resolution is 0.0625 Hz. You do not need to check for overflow, i.e., you may assume the signal will be less than 1000 Hz.

(5) Part a) To store frequency data in memory, you will need a fixed-point number system. What fixed-point format is best? In particular, how would you represent the frequency 123.375 Hz in memory?

(10) Part b) The following is the frequency measurement program from the end of Chapter 6. Modify this program so that the frequency measurement resolution is 0.0625 Hz. Add comments to explain *each change* you make.

```

unsigned int Freq; /* Frequency with units of 0.1 Hz */
unsigned char Done; /* Set each measurement, every 10 s */
unsigned int FourHundred; // Used to create 10sec interval
unsigned int Count; // Number of rising edges
#pragma interrupt_handler TIC1handler()
void TIC1handler(void) {
    Count++; // number of rising edges
    TFLG1=0x02; } // ack, clear C1F
#define Rate 50000 // 25 ms
#pragma interrupt_handler TOC5handler()
void TOC5handler(void) {
    TFLG1= 0x20; // Acknowledge
    TC5 = TC5+Rate; // every 25 ms
    if (++FourHundred==400) {
        Freq = Count; // 0.1 Hz units
        FourHundred=0;
        Done = 0xff;
        Count = 0; } } // Setup for next
void ritual(void) {
asm(" sei "); // make atomic
TIOS| = 0x20; // enable OC5
TSCR = 0x80; // enable
TMSK2 = 0x32; // 500 ns clock
TMSK1 = 0x22; // Arm OC5 and IC1
TCTL4 = (TCTL4&0xF3)|0x04; /* C1F set on rising edges */
TC5=TCNT+Rate; // First in 25 ms
Count = 0; // Set up for first
Done=0; // Set on subsequent meas
FourHundred=0;
TFLG1 = 0x22; // Clear C5F, C1F
asm(" cli "); }

```

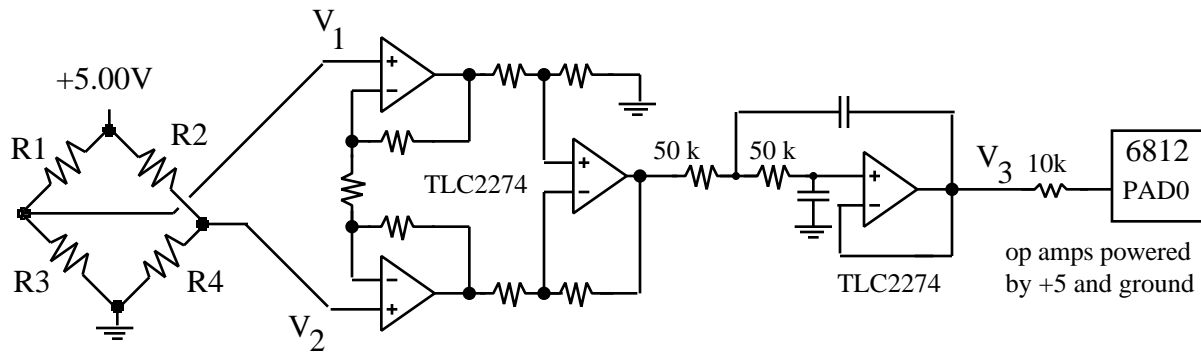
(40) Question 2. In this problem, you will design a blood pressure DAS. The range of pressure is 0 to 300 mmHg. Because you are using the 8-bit ADC and the system is linear, the measurement precision will also be 8 bits. The frequencies of interest are 0 to 250 Hz. The pressure is measured with 4 resistive strain gages placed in a bridge. The sensitivity of each gage is $1/30 \text{ } \Omega/\text{mmHg}$. The change in transducer resistance, R , is linearly related to pressure.

(5) Part a) Complete the following table. ADR0H is the digital result from the 6812's 8-bit ADC. The system will use the full scale ADC range. Refer to the circuit in part c).

P	R	R1	R2	R3	R4	V1	V2	V1-V2	V3	ADR0H
0	0	1000	1000	1000	1000					
150	5	995	1005	1005	995					
300	10	990	1010	1010	990					
mmHg						V	V	V	V	

(5) Part b) What sampling rate would you choose? Why?

(10) Part c) Specify resistor and capacitor values to complete the analog interface. Include a LPF at 250 Hz. Some of the resistors values ($R1, R2, R3, R4, 50 \text{ k } \Omega, 50 \text{ k } \Omega,$ and $10 \text{ k } \Omega$) are fixed, and you can not change their values.



(5) Part d) What is the maximum allowable noise of the analog circuit, referred to its input (at V1, V2)?

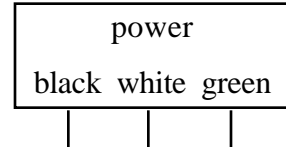
(5) Part e) Show the ritual that initializes the system (data structures, port configuration, and interrupt arm). The main program will not directly participate in the measurement, but be free to execute other functions. You will not write the main program. Add and explain any variables you need.

(10) Part f) Show the interrupt service routine that samples the ADC, calculates pressure in mm Hg, (let $x(n)$ be the current pressure measurement), and implements the following averaging digital LPF. This calculation is performed for each sample, not calculated once a second. I.e., for each sample calculate the average of the previous 1-second data. The constant N will simply be the sampling rate, f_s . The correct answer has **no backward jumps** in the interrupt service routine. Deal with the **overflow** potential in both the calculation of pressure and in the digital filter.

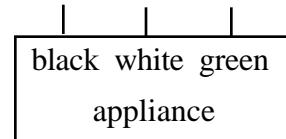
$$y(n) = \frac{1}{N} \sum_{i=0}^{N-1} x(n-i)$$

(15) Question 3.

(10) Part a) Design an interface that allows the microcomputer software to turn on/off a standard 120 VAC 60 Hz appliance. The maximum appliance power is about 25 W, requiring a current of about 0.25 A. Your interface must be isolated, i.e., the 6812 ground is not connected to the appliance ground through your interface. You may use any interfacing chip or device mentioned in the book, but you may not use any device that is not in the book. You must use actual specific devices (with manufacturer and part number) rather than generic devices. Show the design steps and all connections to each chip. Label all chip numbers (but not pin numbers). The 120 VAC power line has three wires black (hot), white (return) and green (ground). You may assume the standard appliance, which you are controlling, also has black, white, and green wires. Specify all power supply connections, resistor/capacitor/diode types/values.



Computer
Output
Port

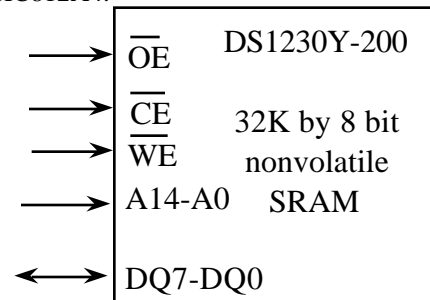


(5) Part b) What type of microcomputer-based closed-loop controller uses this type of interface?

(30) Question 4. One of the options when you need more program space (the 4096 bytes of EEPROM is not enough) is to add a battery-backed RAM. In this problem, you will interface a DS1230Y-200 32K by 8-bit nonvolatile SRAM to a MC68HC812A4 running at 8 MHz. Your microcomputer will boot up in special expanded narrow mode. Assume a 10 ns gate delay through any digital logic gate.

(5) Part a) At what addresses should you place the DS1230Y-200? Why?

(5) Part b) Show the digital interface between the DS1230Y-200 and the MC68HC812A4.



(5) Part c) As far as the read cycle is concerned, how many cycle-stretches are required? Show your work.

(5) Part d) As far as the write cycle is concerned, how many cycle-stretches are required? Show your work.

(10) Part e) Show the combined *read cycle timing diagram*. Show E, R/W, A15-A0, CE, WE, OE, RDA, and RDR. All signals are outputs except Read Data Required. Use arrows to signify causal relations. Show the timing delays as arrows with numbers in nanoseconds. Calculate the actual RDA and RDR intervals.