(10) **Question 1.** We use an instrumentation amp, because of the differential nature and the input impedance. We add a 2.5V to the ground signal of the amp to scale the signals near 2.5V. Gain of 100 is calculated as $\frac{100k}{10k}(1+2\times\frac{90k}{20k})$.

---

(5) **Question 2.** Design a two pole analog low-pass filter with a cutoff frequency of 1150 Hz.

**initial starting point**

- Using **LPF.x1s**
  - $fc$ (Hz) 1
  - $R$ (kohm) 10
  - $C1$ (µF) 141
  - $C2$ (µF) 70.7

**first design step is to select the cutoff frequency**

- $fc$ (Hz) 1125 fill this in
- $RA$ (kohm) 10 same as initial $R$
- $C1A$ (µF) 0.02 is $\frac{141.4}{2\times\pi\times fc}$
- $C2A$ (µF) 0.01 is $\frac{70.7}{2\times\pi\times fc}$ or $0.5\times C1A$

---

(5) **Question 3.** First, determine the measurement resolution, which is 10 cm/sec divided by the ADC precision of 4096. This system can resolve velocity differences of 0.00244 cm/sec. Binary fixed-point is selected because multiple calculations will be required. A **16-bit unsigned binary fixed-point** number system with a resolution of $\frac{1}{512}$ cm/sec will be able to represent all measured values. The range is 0 to 127 cm/sec. Possible 16-bit unsigned binary fixed-point solutions are

- Resolution = $\frac{1}{512}$ cm/sec
  - Range = 0 to 127 cm/sec
  - 5 cm/sec stored as 2560

---

(5) **Question 4.** Frame size is $29+32+36=97$ bits. Bandwidth is 4 bytes/frame times 100,000 bits/sec divided by 97 bits/frame = **4123 BYTES/SEC**.

---

(5) **Question 5.** The short messages can arrive every $75\times5\mu$s (375 µs). There are five receive buffers on the 9S12C32 CAN interface. After one buffer is full, the RXF flag is set. $4\times375$ µs (1.5ms) later it is possible that all five buffers will be full. Latency requirement is 1.5ms. This value assumes the CAN interrupt will receive all pending messages, emptying the FIFO.
(10) Question 6. See Table 9.17 in View22.

(5) Part a) The propagation delay from clock to output change through the 74FCT374 has a minimum of 2 ns and a maximum of 6.5 ns. Assuming 9 ns 74FCT139 gate delay max and 1.5 ns delay minimum, E1 falls at [127,131.5]+[1.5,9] ns. RDR is (735, 750). Thus, the access time must be less than 735-140.5 = 594.5 ns.

(5) Part b) The fall of E2 is 750 ns. WDA is (132, 752). So the setup time must be less than 750-132 = 618 ns.

(20) Question 7. The first step is to convert the sin waves into square waves. A Rail-to-Rail op amp with positive feedback will give a clean 0-to-5 V squarewave with the same phase as the original sin wave. Positive feedback creates hysteresis, making it more immune to noise. The diode protects the op amp input from negative voltages. A HPF could have been used to center both waves at 0 V. The AND gate creates a 100 Hz square-wave such that the duty cycle is linear with phase. The 9S12C32 input capture measures pulse width of the wave.

Part a) Show the hardware analog circuit

Part b) 0 to 90 deg is converted to Phase 0 to 10000 (0.009 deg resolution).

unsigned short Phase; // units of 360deg*250ns/10ms = 0.009 deg
unsigned short Rising; // TCNT at rising

void Init(void) {
    asm sei         // make atomic
    TIOS &=~0x80;   // clear bit 7
    DDRT &=~0x80;   // PT7 is input capture
    TSCR1 =0x80;    // enable
    TSCR2 =0x00;    // 250 ns clock
    TCTL3|=0xC0;    // Both edges IC7
    TIE |= 0x80;    // arm IC7
    TFLG1 = 0x80;   // clear C7F
    Rising = TCNT; // first measurement may be wrong
    asm cli
}

Part c) Show the interrupt service routine(s) that measure phase.

void interrupt 15 TC7handler(void){
    if(PPT&0x80){ // PT7=1 if rising
        Rising = TC7; // Setup for next
    } else{
        Phase = TC7-Rising; // measurement resolution 0.009 deg
    }
    TFLG1 = 0x80; // ack, clear C7F
(10) Question 8. From the problem description
\[ \Delta U = 2500 \text{ (duty-units)} \]
From the graph
\[ L = 1 \text{ second} \]
\[ R = \frac{10000-5000}{2-1} = 5000 \text{ (period-units/sec)} \]
It is recommended to run PI controllers with \( \Delta t = 0.1L = 0.1 \text{sec}. \)
\[ K_P = 0.9 \frac{\Delta U}{(L*R)} = 0.9 \times \frac{2500}{(1\times5000)} = 0.45 \]
\[ K_I = \frac{K_P}{(3.33L)} = 0.135 \]
\[ E(n) = \text{Desired - Period} \]
\[ P(n) = K_p \times E(n) = 0.45 \times E(n) = \left(\frac{9}\times E(n)\right)/20 \]
\[ I(n) = I(n-1) + K_I \times E(n) \times \Delta t = I(n-1) + 0.135 \times E(n) = I(n-1) + \left(\frac{27\times E(n)}{200}\right) \]
\[ \text{Duty} = U(n) = P(n) + I(n) \]
Putting together, and simplifying
\[ E = \text{Desired} - \text{Period}; \]
\[ P = \left(\frac{9\times E}{20}\right); \]
\[ I = I + \left(\frac{27\times E}{200}\right); \]
\[ \text{Duty} = P + I; \]

(20) Question 9. A detailed solution will not be given because this problem will become a lab assignment. No major changes are required to the existing Lab17 code. The basic idea is to implement `fork()` as a software interrupt call. This will push all the registers on the stack. The SWI handler will
  - Allocate a block to be used for a new TCB
  - Copy all the data from the running TCB to the new TCB
  - Generate a new ID and set the ID field of the new TCB
  - Adjust the SaveSP field to point to the new stack of the new TCB
  - Link the new TCB into the circular linked list
  - Set the value on the New TCB so RegD returns 0
  - Set the value on the running TCB so RegD returns the ID of the new thread