(4) Question 1. To make it allocated in permanent RAM, C) static
(4) Question 2. Because of all the calculations, use B) Binary fixed-point
(4) Question 3. To increase $V_{OH}$ C) No, add a resistor to $+5V$ (it will effect $I_{OL}$)
(4) Question 4. ADC range is 2V, Precision is 2V/0.001V, which is 2000 alternatives. Use an 11-bit ADC, which has a precision of 2048 alternatives.
(4) Question 5. Need NPN to sink current, need resistor to set LED current (C)
(4) Question 6. Slowing down the 9S12 will B) make the batteries last longer
(4) Question 7. Because the Fifo is empty, the software is not putting much data into it
   A) The system is CPU bound
(4) Question 8. Because the Fifo is empty, the input rate is slow
   C) The system is I/O bound
(4) Question 9. Functional debugging means we are interested in the input/output data. minimally intrusive means the debugging instrument needs to execute fast as compared to the time between interrupt calls. The dump is the fastest way to collect data
   D) asm sei
   \[ \text{if}(n<100) \{ \text{BufX}[n]=x; \text{BufY}[n]=y; \text{n++;} \} \]
   asm cli
(4) Question 10. Start with coworkers (you probably measured something wrong), then go to your boss, then go to the safety officer at the company. As a last resort, contact federal safety commissions.
(5) Question 11. Clock is normally high (CPOL=1), 9S12 needs to latch on the falling edge (other edge from the input device) (CPHA=0). See book figure 7.40
(10) Question 12. Use $+12V$ supply because it is a 12V motor. Use a NPN, because it is simpler to interface a current sink to the 9S12 (they share common ground, but do share a common supply). Need $12V/10\Omega = 1.2A$, so use a TIP120, which has a maximum ICE of 3A. The hFE=1000 of the TIP120 means the I1H of the 9S12 will only be 1.2mA. The resistor can be added to limit the base current to a maximum of 1.2mA. The snubber diode (1N914) prevents back EMF.

(15) Question 13. Address decoder is A15, output positive logic, synchronized to E

---

Part b) The delays on first start of the interval are maximums, and at the end we use minimums. **Read Data Available** = $[250+15+15+60,500+5+5+10] = [340,520]$
(15) Question 14. A data acquisition system to measure angle.
Part a) Good CMRR and differential means use instrumentation amp. Need gain = 5 (no
Rf is needed, leave pins 1,8 open)

Part b) It is unsigned, and the integer will be 16 bits. Since 360/1024 is 0.35, I would
choose 0.1 ° for the resolution
Part c) Scale 0 to 1023 into 0 to 3600.
unsigned short Convert(unsigned short rawData){
    unsigned long answer;
    answer = 3600*rawData;
    return answer/1023;
}
unsigned short Convert(unsigned short rawData){
    return 3600*rawData /1023;  // some compilers will do this correctly
}
unsigned short Convert(unsigned short rawData){
    return 225*rawData /64;  // some compilers will do this correctly
}

(15) Question 15. TDRE will be 1 at the time of each IC interrupt
unsigned char Count; // 0,1,2,3,4 used to find every fifth edge
void main(void){
    TIOS &= ~0x01;  // PT0 input capture
    DDRT &= ~0x01;  // PT0 is input
    TSCR1 = 0x80;   // enable TCNT
    TCTL4 = (TCTL4&0xFC)|0x01; // rising
    TFLG1 = 0x01;   // Clear C0F
    TIE |= 0x01;    // Arm IC0
    SCIBD=13;       // br=MCLK/(16*baudRate)
    SCICR1 = 0;     // 1 start, 8 data, 1 stop, no parity
    SCICR2 = 0x0C;  // TE, enable transmit, RE, enable receive(optional)
    asm cli
    while(1){};
}

interrupt 8 void TC0handler(void){
    unsigned char dummy;
    TFLG1 = 0x01; // ack by clearing C0F
    Count--;
    if(Count == 0){
        Count = 5; // fifth rising edge
        dummy = SCISR1; // read status with TDRE set
        SCIDRL = 0x56; // output 'V'
    }
}