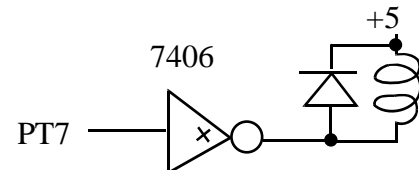
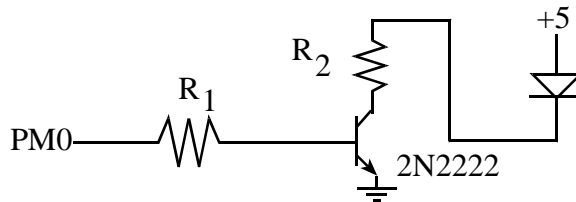


Jonathan W. Valvano

March 7, 2007, 1:00pm-1:50pm.

<b>Question 1.</b> A B or C	<b>A) nonintrusive</b>	<b>Question 8.</b> A B C or D	<b>C) EEPROM</b>
<b>Question 2.</b> A B or C	<b>C) highly intrusive</b>	<b>Question 9.</b> A B C or D	<b>D) global RAM</b>
<b>Question 3.</b> A B or C	<b>B) minimally intrusive</b>	<b>Question 10.</b> A B C or D	<b>A) Reg D or B) stack RAM</b>
<b>Question 4.</b> A B or C	<b>C) highly intrusive</b>	<b>Question 11.</b> A B C or D	<b>D) global RAM</b>
<b>Question 5.</b> A B or C	<b>B) minimally intrusive</b>	<b>Question 12.</b> A B C or D	<b>B) stack RAM</b>
<b>Question 6.</b> A - H	<b>H) long</b>	<b>Question 13.</b> A - I	<b>F) TCNT equals TC7</b>
<b>Question 7.</b> Yes/no (why)	Yes, this is read-modify-write nonatomic access to a global. We need to disable interrupts to make it atomic in the main program	<b>asm sei Count++; asm cli</b>	

(5) **Question 14.** You need an interface that can sink 100 mA, like the 2N2222. Since the LED needs 2V we connect one end to a voltage larger than  $2V + V_{ce}$ . Since the LED needs 100 mA, and  $h_{fe}$  is 100, the base current needs to be at least 1mA. When the 9S12C32 outputs a high, the voltage on PT7 will be at least 4.2V ( $V_{OH}$ ). The 2N2222 will be active if  $V_{be}$  is over 0.6V. To select the LED resistor,  $R_2 = (5 - 2 - 0.3V) / 1mA = 2.7V / 100mA = 27 \Omega$ . To select base resistor  $R_1$  for this interface, we set  $(4.2V - 0.6V) / R_1 > 1mA$ .  $R_1 < (4.2V - 0.6V) / 1mA = 3600 \Omega$ .



(10) **Question 15.** You need an interface that can sink 20 mA, like the 7406. Since it is a +5V stepper, we will connect one side of the stepper coil to +5V. The snubber diode protects the interface from back EMF.

(35) **Question 16.**

```

StateType fsm[2]={
{ 500,{ 1, 3, 5, 7},{ S0, S0, S1, S1}},
{ 300,{ 0, 4, 8, 12},{ S1, S1, S1, S0}}
};
void InitFSM(void){
    TSCR1 = 0x80; // enable TCNT
    TSCR2 = 2; // divide by 4, creating 1 MHz TCNT
    TIOS |= 0x40; // channel 6 is output compare
    TIE |= 0x40; // arm output compare 2
    DDRM = 0x0F; // PM3,2,3,0 outputs PM5,4 inputs
    Pt = S0; // initial state
    TC6 = TC6+500; // wait for S0
    asm cli // enable
}
void interrupt 14 OC6han(void){unsigned char input;
    TFLG1 = 0x40; // acknowledge
    input = (PTM&0x30)>>4; // PM5 and PM4 are inputs
    PTM = Pt->Out[input]; // output depends on input and state
    Pt = Pt->Next[input]; // next depends on input and state
    TC6 = TC6+Pt->Time;
}

```