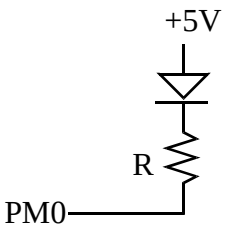
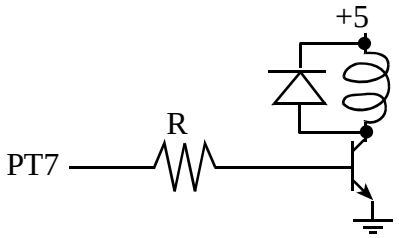
Jonathan W. Valvano March 7, 2007, 1:00pm-1:50pm.

Question 1. A B or C	**B) minimally intrusive**	**Question 8.** A B C or D	**A) EEPROM**
Question 2. A B or C	**A) nonintrusive**	**Question 9.** A B C or D	**B) global RAM**
Question 3. A B or C	**C) highly intrusive**	**Question 10.** A B C or D	**C) Reg D** or **D) stack RAM**
Question 4. A B or C	**B) minimally intrusive**	**Question 11.** A B C or D	**B) global RAM**
Question 5. A B or C	**C) highly intrusive**	**Question 12.** A B C or D	**D) stack RAM**
Question 6. A - H	**F) `long`**	**Question 13.** A - I	**G) `TCNT` equals `TC7`**
Question 7. Yes/no (why)	**no**	**It is atomic with**	**interrupts disabled.**

(5) Question 14. The 1mA LED current means you can connect it directly to the 9S12C32.
R=(5-2-0.8V)/1mA =2.2V/1mA = 2200 Ω





(10) Question 15. You need an interface that can sink 50 mA, like the 2N2222. Since it is a +5V stepper, we will connect one side of the stepper coil to +5V. Since the stepper coil needs 50 mA, and $\mathbf{h_{fe}}$ is 100, the base current needs to be at least 0.5mA. When the 9S12C32 outputs a high, the voltage on PT7 will be at least 4.2V ($\mathbf{V_{OH}}$). The 2N2222 will be active if $\mathbf{V_{be}}$ is over 0.6V. To select R for this interface, we set (4.2V-0.6V)/R > 0.5mA. R < (4.2V-0.6V)/0.5mA = 7200 Ω.

(35) Question 16.

```
StateType fsm[2]={
{  300,{  2,  4,  6,  8},{ S1, S0, S1, S0}},
{  500,{  0,  5, 10, 15},{ S1, S1, S0, S1}}
};
void InitFSM(void){
  TSCR1 = 0x80;  // enable TCNT
  TSCR2 = 2;     // divide by 4, creating 1 MHz TCNT
  TIOS |= 0x04;  // channel 2 is output compare
  TIE |= 0x04;   // arm output compare 2
  DDRM = 0x3C;   // PM5,4,3,2 outputs    PM1,0 inputs
  Pt = S0;       // initial state
  TC2 = TC2+300; // wait for S0
  asm cli        // enable
}
void interrupt 10 OC2han(void){unsigned char input;
  TFLG1 = 0x04;             // acknowledge
  input = PTM&0x03;         // PM1 and PM0 are inputs
  PTM = 4*(Pt->Out[input]); // output depends on input and state
  Pt = Pt->Next[input]);    // next  depends on input and state
  TC2 = TC2+Pt->Time;
}
```