CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test - III

Sub:	Microcontroller							Code:	17EE52
Date:	16/11/2019	Duration:	90 mins	Max Marks:	50	Sem:	5	Branch:	EEE
Solution									

1. Explain the principle of stepper motor. Write a program to rotate motor 64^0 in clockwise direction. The motor has step angle of 2^0. Write the 4 step sequence also. The motor has steps per revolution=180, number of rotor teeth=45, movement per 4 step sequence=8^0.

Review Questions

1. Give one application where would you use a relay.
2. Why do we place a driver between the microcontroller and the relay?
3. What is an NC relay?
4. Why are relays that use coils called electromechanical relays?
5. What is the advantage of a solid-state relay over EMR?
6. What is the advantage of an optoisolator over an EM relay?

SECTION 17.2: STEPPER MOTOR INTERFACING

This section begins with an overview of the basic operation of stepper motors. Then we describe how to interface a stepper motor to the 8051. Finally, we use Assembly language programs to demonstrate control of the angle and direction of stepper motor rotation.

Stepper motors

A *stepper motor* is a widely used device that translates electrical pulses into mechanical movement. In applications such as disk drives, dot matrix printers, and robotics, the stepper motor is used for position control. Stepper motors commonly have a permanent magnet *rotor* (also called the *shaft*) surrounded by a *stator* (see Figure 17-7). There are also steppers called variable reluctance *stepper motors* that do not have a PM rotor. The most common stepper motors have four stator windings that are paired with a center-tapped common as shown in Figure 17-8. This type of stepper motor is commonly referred to as a *four-phase* or unipolar stepper motor. The center tap allows a change of current direction in each of two coils when a winding is grounded, thereby resulting in a polarity change of the stator. Notice that while a conventional motor shaft runs freely, the stepper motor shaft moves in a fixed repeatable increment, which allows one to move it to a precise position. This repeatable fixed movement is possible as a result of basic magnetic theory where poles of the same polarity repel and opposite poles attract. The direction of the rotation is dictated by the stator poles. The stator poles are determined by the current sent through the wire coils. As the direction of the current

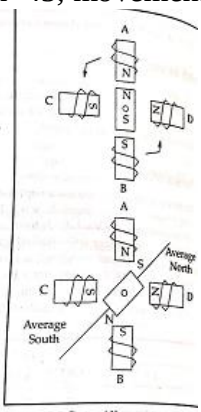


Figure 17-7. Rotor Alignment

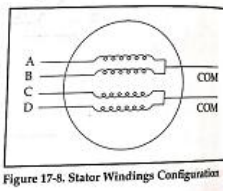


Figure 17-8. Stator Windings Configuration

Write a program to rotate a motor 64° in the clockwise direction. The motor has a step angle of 2°. Use the 4-step sequence in Table 17-3.

Solution:

A motor with a 2° step angle has the following characteristics:
Step angle: 2°
Steps per revolution: 180
No. of rotor teeth: 45
Movement per 4-step sequence: 8°
To move the rotor 64°, we have to send eight consecutive 4-step sequences, i.e., 32 steps.

```
        ORG   0000H
        MOV   A,#66H
        MOV   R0,#32
BACK:   RR    A
        MOV   P1,A
        ACALL DELAY
        DJNZ  R0,BACK
        END
```

2. a. Explain the bit status of SCON special function register.

b. Write a C program for 8051 to transfer the letter ‘A’ serially at 4800 baud continuously using timer 1 in mode 2.

- SCON is an 8-bit register used to program the start bit, stop bit, and data bits of data framing, among other things

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Used for multiprocessor communication
REN	SCON.4	Set/cleared by software to enable/disable reception
TB8	SCON.3	Not widely used
RB8	SCON.2	Not widely used
TI	SCON.1	Transmit interrupt flag. Set by HW at the begin of the stop bit mode 1. And cleared by SW
RI	SCON.0	Receive interrupt flag. Set by HW at the begin of the stop bit mode 1. And cleared by SW

Note: *Make SM2, TB8, and RB8 =0*

Write a program for the 8051 to transfer letter "A" serially at 4800 baud, continuously.

Solution:

```
        MOV  TMOD,#20H  ;timer 1,mode 2(auto reload)
        MOV  TH1,#-6    ;4800 baud rate
        MOV  SCON,#50H  ;8-bit, 1 stop, REN enabled
        SETB TR1        ;start timer 1
AGAIN:  MOV  SBUF,#"A"  ;letter "A" to transfer
HERE:   JNB  TI,HERE    ;wait for the last bit
        CLR  TI         ;clear TI for next char
        SJMP AGAIN      ;keep sending A
```

3. a. Briefly show control word of 8255 and specify mode selection.
   b. Calculate the control word of 8255 for the following cases:
   (i) All the ports A, B and C are output ports(mode 0).
   (ii) PA=in, PB=out, PCL=out and PCH=out.

simple I/O programming

Intel calls mode 0 the *basic input/output mode*. The more commonly used term is *simple I/O*. In this mode, any of por
, B, or C can be programmed as input or output. It must be noted that in this mode, a given port cannot be both a
tput and output port at the same time.

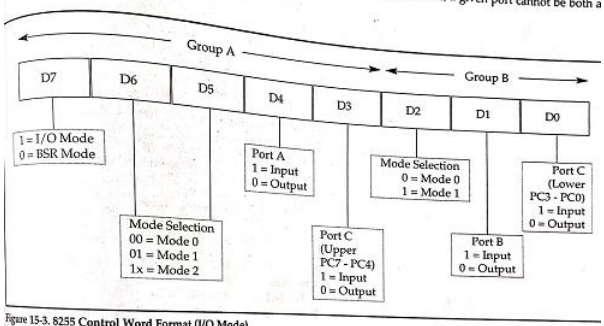


Figure 15-3. 8255 Control Word Format (I/O Mode)
[Reprinted by permission of Intel Corporation, Copyright Intel Corp., 1983]

Find the control word of the 8255 for the following configurations:
(a) All the ports of A, B, and C are output ports (mode 0).
(b) PA = in, PB = out, PCL = out, and PCH = out.

Solution:

From Figure 15-3 we have:
(a) 1000 0000 = 80H (b) 1001 0000 = 90H

4. a. Write the steps required to transfer data serially on 8051.

b. What are Interrupts? Specify vector location of interrupts in 8051.

- In programming the 8051 to transfer character bytes serially
  1. TMOD register is loaded with the value 20H, indicating the use of timer 1 in mode 2 (8-bit auto-reload) to set baud rate
  2. The TH1 is loaded with one of the values to set baud rate for serial data transfer
  3. The SCON register is loaded with the value 50H, indicating serial mode 1, where an 8-bit data is framed with start and stop bits
  4. TR1 is set to 1 to start timer 1
  5. TI is cleared by `CLR TI` instruction
  6. The character byte to be transferred serially is written into SBUF register
  7. The TI flag bit is monitored with the use of instruction `JNB TI,xx` to see if the character has been transferred completely
  8. To transfer the next byte, go to step 5

An *interrupt* is an external or internal event that interrupts the microcontroller to inform it that a device needs its service

Interrupt vector table

Interrupt	ROM Location (hex)	Pin
Reset	0000	9
External HW (INT0)	0003	P3.2 (12)
Timer 0 (TF0)	000B	
External HW (INT1)	0013	P3.3 (13)
Timer 1 (TF1)	001B	
Serial COM (RI and TI)	0023	

5. Write an 8051 C program using interrupts to do the following:

(i) Receive the data serially and send it to P0.

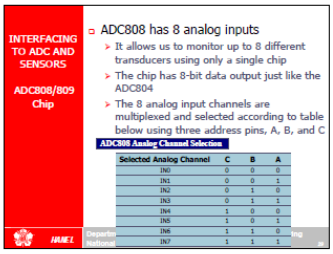
(ii) Read port P1, transmit data serially and give a copy to P2.

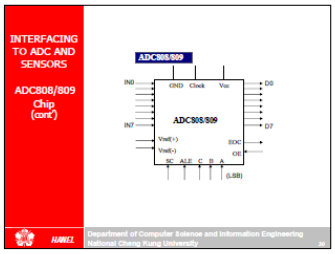
(iii) Generate a square ware of 5 KHz frequency on P0.1.

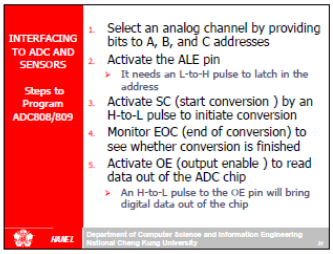
Set the baud rate at 4800.

```
      ORG  0
      LJMP MAIN
      ORG  000BH  ;ISR for timer 0
      CPL  P0.1   ;toggle P0.1
      RETI        ;return from ISR
      ORG  23H    ;
      LJMP SERIAL ;jump to serial interrupt ISR
      ORG  30H
MAIN: MOV  P1,#0FFH ;make P1 an input port
      MOV  TMOD,#22H;timer 1,mode 2(auto reload)
      MOV  TH1,#0F6H;4800 baud rate
      MOV  SCON,#50H;8-bit, 1 stop, ren enabled
      MOV  TH0,#-92 ;for 5kHZ wave
      MOV  IE,10010010B ;enable serial int.
      SETB TR1     ;start timer 1
      SETB TR0     ;start timer 0
BACK: MOV  A,P1    ;read data from port 1
      MOV  SBUF,A  ;give a copy to SBUF
      MOV  P2,A    ;send it to P2
      SJMP BACK    ;stay in loop indefinitely
;-----------------SERIAL PORT ISR
      ORG  100H
SERIAL:JB   TI,TRANS;jump if TI is high
      MOV  A,SBUF  ;otherwise due to receive
      MOV  P0,A    ;send serial data to P0
      CLR  RI      ;clear RI since CPU doesn't
      RETI         ;return from ISR
TRANS: CLR  TI     ;clear TI since CPU doesn't
      RETI         ;return from ISR
      END
```

6. Draw the block diagram of DAC 0808 interfaced to 8051 at port P1 and write an 8051 program to generate a sine wave.



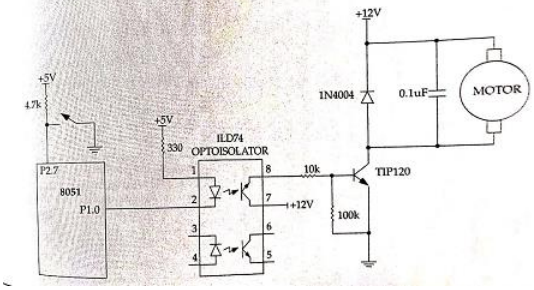




7. How to interface DC motor to 8051 microcontroller using Optoisolator? Write a C program to move DC motor with 25 % duty cycle pulse.

```
#include <reg51.h>
sbit SW = P2^7;
sbit MTR = P1^0;
void MSDelay(unsigned int value);
void main()
  {
  SW = 1;
  MTR = 0;
  while(1)
     {
      if(SW == 1)
        {
          MTR = 1;
          MSDelay(25);
          MTR = 0;
          MSDelay(75);
        }
      else
        {
```

```
        MTR = 1;
        MSDelay(50);
        MTR = 0;
        MSDelay(50);
     }
  }
d MSDelay(unsigned int value)
  unsigned char x, y;
  for(x=0; x<1275; x++
    for(y=0; y<value; y++
```



8. With regard to the interrupt of 8051,
   (i) Give the vector address of the interrupts.
   (ii) Briefly explain the procedure of enabling/disabling the entire interrupt system and enabling/disabling of individual interrupts.
   (iii) Indicate the default priority on reset and procedure to alter this default priority.

IE (Interrupt Enable) Register

D7							D0
EA	--	ET2	ES	ET1	EX1	ET0	EX0

EA (enable all) must be set to 1 in order for rest of the register to take effect

EA	IE.7	Disables all interrupts
--	IE.6	Not implemented, reserved for future use
ET2	IE.5	Enables or disables timer 2 overflow or capture interrupt (8952)
ES	IE.4	Enables or disables the serial port interrupt
ET1	IE.3	Enables or disables timer 1 overflow interrupt
EX1	IE.2	Enables or disables external interrupt 1
ET0	IE.1	Enables or disables timer 0 overflow interrupt
EX0	IE.0	Enables or disables external interrupt 0

Interrupt vector table

Interrupt	ROM Location (hex)	Pin
Reset	0000	9
External HW (INT0)	0003	P3.2 (12)
Timer 0 (TF0)	000B	
External HW (INT1)	0013	P3.3 (13)
Timer 1 (TF1)	001B	
Serial COM (RI and TI)	0023	

- To enable an interrupt, we take the following steps:
  1. Bit D7 of the IE register (EA) must be set to high to allow the rest of register to take effect
  2. The value of EA
     - If EA = 1, interrupts are enabled and will be responded to if their corresponding bits in IE are high
     - If EA = 0, no interrupt will be responded to, even if the associated bit in the IE register is high

9. Write a program using interrupts to get data from P1 and send it to P2 while Timer 1 is turning ON and OFF the LED connected to P0.4 every second.

Delay = 1 s

$\frac{1}{1.085\mu} = 92165$

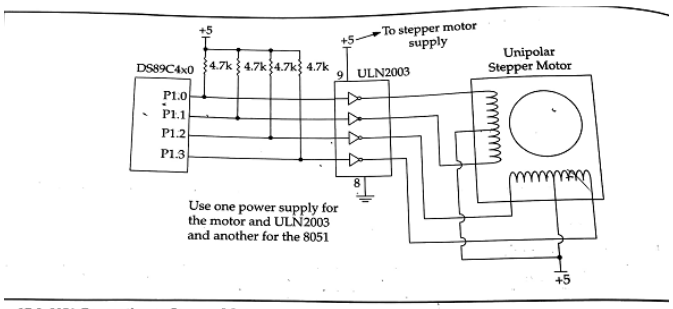
65535 1.4

```
#include<reg51.h>
sfr  P_1 = 0x90;
sfr  P_2 = 0xA0;
sbit P0_4 = P0^4;

void timer1 (void) interrupt 3
{
 P0_4 = 0;
 P0_4 = 1;
}

void main()
{
 A = P_1;
 P_2 = A;
TMOD = 0x10;
TH1 = 00;
TL1 = 00;
IE = 0x88;
}
```

10. Show the interfacing of a stepper motor to 8051 and write 8051 assembly/C program to rotate stepper motor 2 rotations clockwise and one rotation anticlockwise with appropriate delay.



re 17-9. 8051 Connection to Stepper Motor

```
ORG  00H
Mov  A, #66H
Mov  R0, #2     ; 2 rotations clockwise
L1: RL  A
    Acall delay
    DJNZ  R0, L1

    Mov  A, #66H
    Mov  R1, #1     ; 1 rotation anti-clockwise.
L2: RR  A
    Acall delay
    DJNZ  R1, L2

Delay: Mov  R2, #0FFH
L4: Mov  R3, #0FFH
L3: DJNZ  R3, L3
    DJNZ  R2, L4
    Ret

    End
```