SUMMER – 2019 EXAMINATION
Model Answer

Subject Name: Microcontroller and Embedded System **Subject Code: 22434**

Important Instructions to Examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance. Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgment on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

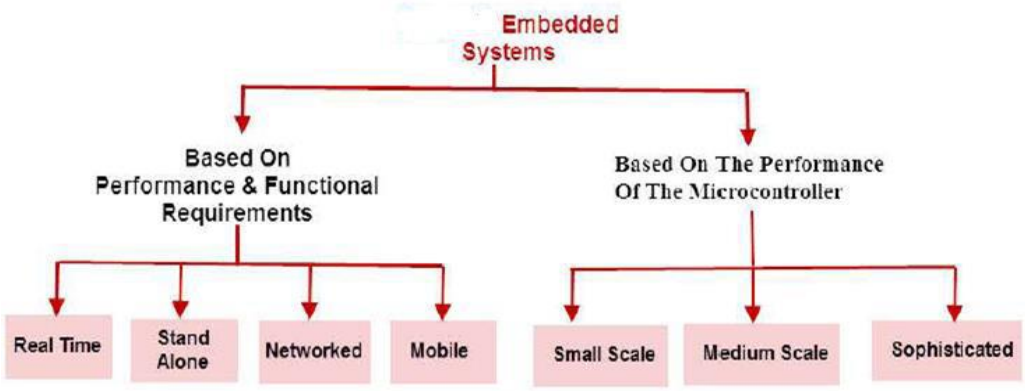
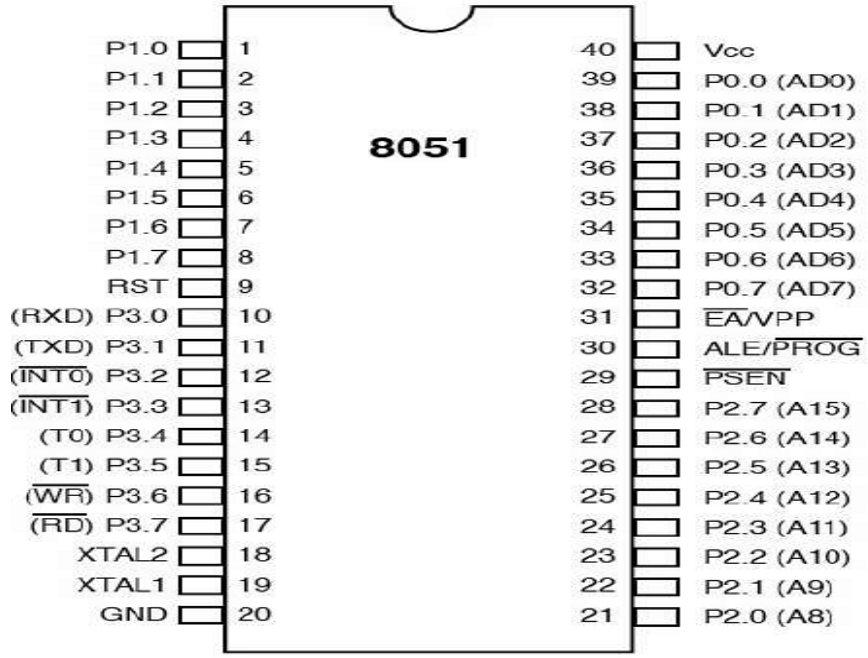
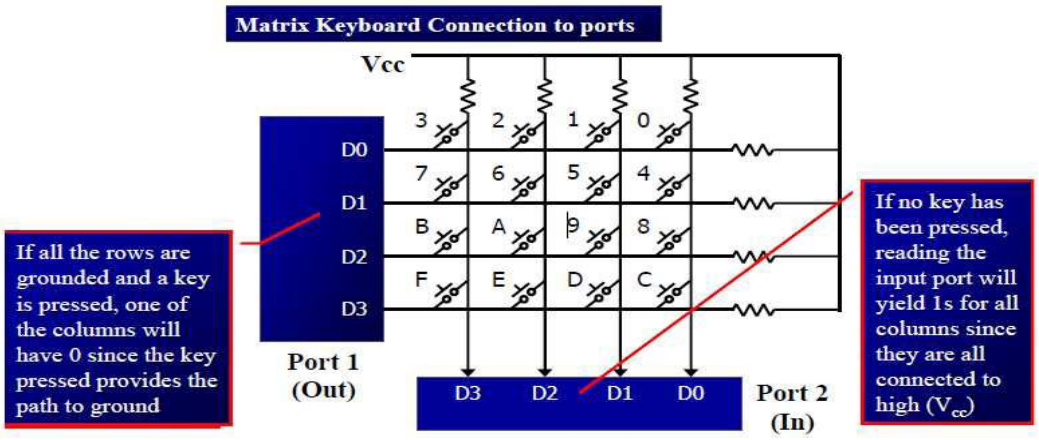
<table>
<tr><th>Q. No.</th><th>Sub Q. N.</th><th>Answer</th><th>Marking Scheme</th></tr>
<tr><td>1.</td><td></td><td>Attempt any FIVE of the following:</td><td>10 M</td></tr>
<tr><td></td><td>a)</td><td>Differentiate between microprocessor and microcontroller based on the following parameters:
i) Memory
ii)Ports
Ans:
<table>
<tr><th>Parameter</th><th>Microprocessor</th><th>Microcontroller</th></tr>
<tr><td>Memory</td><td>Do not have inbuilt RAM or ROM.</td><td>Inbuilt RAM/or ROM</td></tr>
<tr><td>Ports</td><td>I/O ports are not available requires extra devices like 8155or 8255</td><td>I/O ports are available</td></tr>
</table>
Table: Difference between microprocessor and microcontroller</td><td>02M</td></tr>
<tr><td></td><td>b)</td><td>Write the classification of embedded system.
Ans:
 

Fig: Classification of embedded system</td><td>02M</td></tr>
</table>

	c)	**Draw the pin diagram of 89C51 microcontroller.** **Ans:**  **Fig: Pin diagram of 89C51 microcontroller**	02M
	d)	**Write two characteristics of embedded system.** **Ans:** 1. Embedded systems are application specific & single functioned; application is known a Prior, the programs are executed repeatedly. 2. Efficiency is of paramount importance for embedded system. 3. They are optimized for energy, code size, execution time, weight & dimensions and cost. 4. Embedded systems are typically designed to meet real time constants; a real time system reacts to stimuli from the controlled object/operator within the time interval dictated by the environment. For real time systems. 5. Embedded systems often interact with external world through sensors and actuators and hence are typically reactive systems; a reactive system is continual interaction with the environment and executes at a pace determined by that environment. 6. They generally have minimal or no user interface.	02M
	e)	**Draw interfacing diagram of 4*4 matrix keyboard with 89C51 microcontroller.** **Ans:**  **Fig: 4*4 matrix keyboard with 89C51 microcontroller**	02M

f) Differentiate between serial and parallel communication (any two points) — **02M**

Ans:

Sr. No.	Parallel communication	Serial communication
1	Group of data bits, usually 8 bits are transferred at a time	One bit at a time is transferred serially
2	It requires n- number of transmission lines for n- bit data	It requires one or two lines for data transfer.
3	Speed of data transfer is fast.	Speed of data transfer is slow
4	Preferred for short distance communication	Preferred for long distance communication
5	Installation cost is high	Installation cost is low
6	Less reliable	More reliable

Table: Difference between serial and parallel communication

g) Draw interfacing diagram of relay with 89C51 microcontroller. — **02M**

Ans:

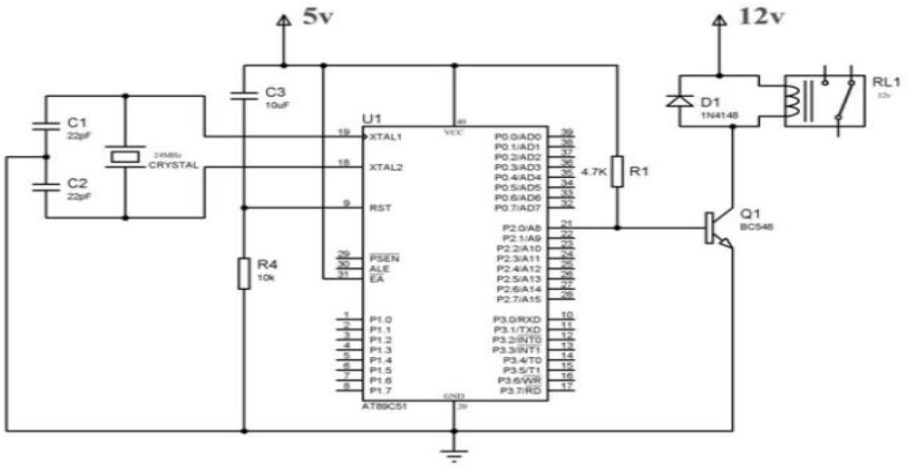


Fig: Interfacing diagram of relay with 89C51 microcontroller

2. Attempt any THREE of the following: — **12 M**

a) List the IDE tools. State the function of Editor and debugger. — **02M**

Ans:

List of IDE tools:

1. Compiler
2. Cross assembler
3. Cross compiler
4. Locators
5. Loaders
6. Simulators
7. Debugger
8. Integrated development environment (IDE)

Function of Editor and debugger: — **01M**

Editor: An editor is a program which helps you to construct your assembly language program in right format so that the assembler will translate it correctly to machine language. So, you can type your program using editor. This form of your program is called as source program and extension of program must be .asm or .src depending on which assembler is used. The DOS based editor such as EDIT, WordStar, and Norton

		Editor etc. can be used to type your program. **Debugger**: is used to find the error and it keeps the control over the system environment and ability to test or follow the execution of the program. Debugger allows the user to load program in to the system memory, executes the program by single stepping and detect logical errors in the program.	**01M**

	b)	**Develop an 89C51 program to read the number 1 from port1, number 2 from port2 and then add them. Store the result and send it to port 3.**
		Ans:
		`#include<reg51.h>`
		`void main( )`
		`{`
		`unsigned char a,b,c;`
		`P1=0xff;`
		`P2=0xff;`
		`P3=0x00;`
		`a=P1;`
		`b=P2;`
		`c=a+b;`
		`P3=c;`
		`}`
		OR
		`#include<reg51.h>`
		`void main( )`
		`{`
		`unsigned char a,b,c;`
		`P1=0xff;`
		`P2=0xff;`
		`P3=0x00;`
		`while(1)`
		`{`
		`a=P1;`
		`b=P2;`
		`c=a+b;`
		`P3=c;`
		`}`
		Table: 89C51 program to read the number 1 from port1, number 2 from port2
			04M

	c)	**Write four features of 89C51 microcontroller.** **Ans:** 1. 4 KB on chip program memory. 2. 128 bytes on chip data memory (RAM). 3. 128 user defined software flags. 4. 8-bit data bus 5. 16-bit address bus 6. 32 general purpose registers each of 8 bits 7. 16 bit timers (usually 2, but may have more, or less). 8. 3 internal and 2 external interrupts. 9. Bit as well as byte addressable RAM area of 16 bytes. 10. Four 8-bit ports, (short models have two 8-bit ports). 11. 16-bit program counter and data pointer. 12. 1 Microsecond instruction cycle with 12 MHz Crystal.	**04M**

	d)	**Develope 89C51 C program to toggle all bits of port P1 continuously.**
		Ans:
		`#include<reg51.h>`
		`Void main( )`
		`{`
		`P1=0x55;`
		`while(1)`
		`{`
		`P1=~P1;`
		`}`
		`}`
		OR
		`#include<reg51.h>`
		`Void main( )`
		`{`
		`while(1)`
		`{`
		`P1=0x55;`
		`P1=0xAA;`
		`}`
		`}`
		Table: 89C51 C program to toggle all bits of port P1 continuously
			04M

3.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**State the logical operators of embedded C and give one example of each.** **Ans:**	**List 02 M example 02 M**

Operator	Symbol	Example
AND	& is called Logical AND operator.	Y= A&B
OR	| is called Logical OR operator.	Y= A|B
NOT	~ is called NOT operator.	Y= ~A
EX-OR	^ is called logical Ex-OR operator.	Y=A ^B
Right shift	>>is called right shift operator	P0= P0>>1
Left shift	<< is called left shift operator	P0= P1<<1

Table: Logical operators of embedded C and give one example of each

	b)	**Draw interfacing diagram of DAC 0808 with 89C51 microcontroller and write C program to generate triangular wave.** **Ans:**	**03M**

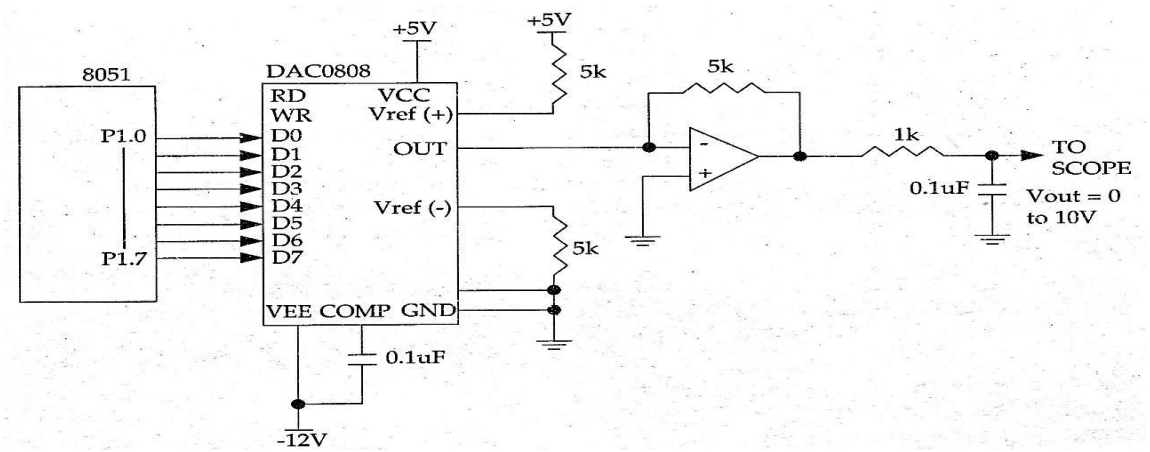


Fig: Interfacing diagram of DAC 0808 with 89C51 microcontroller

```
#include<reg51.h>
unsigned char d;
{
while(1)
{
for(d=0; d<255; d++)
{
P1 = d;
}
for(d=255; d>0; d--)
{
P1 = d;
}
}
}
```

Table: C program to generate triangular wave. **01M**

c) Describe the function of following pins of 89C51 microcontroller

i) PSEN
ii) EA
iii) ALE
iv) RST

Ans:

PSEN: PSEN stands for "program store enable." In an 8031-based system in which an external ROM holds the program code, this pin is connected to the /OE pin of the ROM. In other words, to access external ROM containing program code, the 8031/51 uses the PSEN signal. When the EA pin is connected to GND, the 8031/51 fetches opcode from external ROM by using PSEN. In systems based on the 8751/89C51/ DS5000 where EA is connected to VCC, these chips do not activate the PSEN pin. This indicates that the on- chip ROM contains program code.

EA: EA stands for External access pin and it is active low. When it is held high, executes instruction from the internal program memory till address 0FFFH, beyond this address the instructions are fetched from external program memory. If this pin is low, all the instructions are fetched from the external memory. During normal operation, this pin should not be floated. (Should be connected to ground).

ALE: ALE stands for address latch enable. It is an output pin and is active high for Latching the low byte of address during accesses to external memory. The ALE pin is Used for demultiplexing the address and data by connecting to the G pin of the 74LS373 chip.

RST: RST stands for reset. The RST pin of 8051 is made high for two machine cycles, while the oscillator is running. A power on reset circuit is used. A pull down resistor of 8.2K form the RST pin to Vss and a capacitor of 10uf from the reset circuit. These component values are sufficient to provide a delay, so as to make the RST line high for 24 oscillations. To support the manual reset function, if desired so, a switch may be added across the 10uf capacitor.

04M

d) Describe SPI protocol with diagram.

Ans:

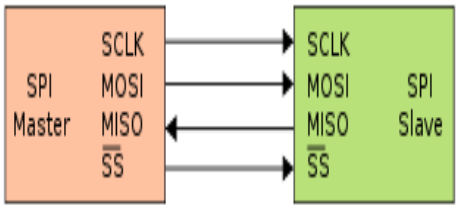
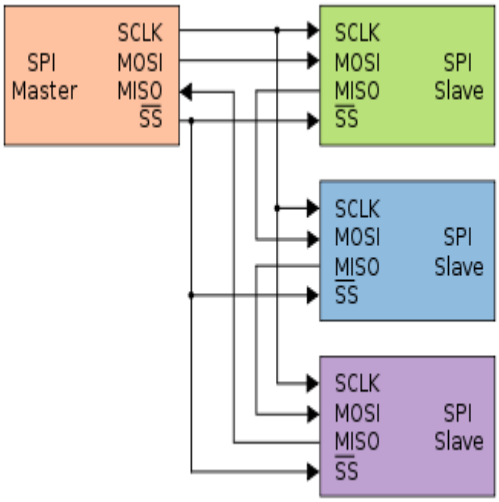
Single Master Single Slave	Single Master Multiple Slave
	

Table: Single Master Single Slave and Single Master Multiple Slave

SPI Protocol:

The Serial Peripheral Interface (SPI) is a synchronous serial communication interface specification used for short-distance communication, primarily

02M

in embedded systems. SPI devices communicate in full duplex mode using a master-slave architecture with a single master. The master device originates the frame for reading and writing. Multiple slave-devices are supported through selection with individual slave select (SS) (sometimes called chip select (CS)) lines. SPI is one master and multi slave communication.

The SPI bus specifies four logic signals:

1. SCLK: Serial Clock (output from master).
2. MOSI: Master Output Slave Input, or Master Out Slave In (data output from master).
3. MISO: Master Input Slave Output, or Master In Slave Out (data output from slave).
4. SS: Slave Select (often active low, output from master).

02M

4. **Attempt any THREE of the following:** **12 M**

a) **List two advantages and two disadvantages of embedded system.**

Ans:

Advantages of embedded system:

1. Cost is low.
2. Small in size.
3. Highly reliable.
4. Operation is fast.
5. Easy for mass production.
6. Less interconnection.
7. Improves product quality.

02M

Disadvantages of embedded system:

1. Hard for maintenance as it is use and throw device.
2. No technological improvement.
3. Hard to back up of embedded files.
4. Less power supply durability if it is battery operated.

02M

b) **Differentiate between assembly language program and embedded C with reference to.**

i) Execution time
ii) time for coding
iii) Hex file size
iv) Debugging

parameters	Assembly language	Embedded C
i) Execution time	Faster(less execution time required)	Slower(More execution time required)
ii) time for coding	slow	Less time required for coding and code is more efficient
iii) Hex file size	less	large
iv) Debugging	Not so easy	easy

Table: Difference between assembly language program and embedded C

04M

c) **Differentiate Between 8051 and 8052 microcontroller (any four points)**

Ans:

			Parameters / **8051** / **8052** Clocks per instruction cycle (fewer is better) / 12 / 12 Timer / 2 / 3 UARTs/Serial port / 1 / 1 Internal DATA RAM bytes / 128 / 256 Maximum PIO Port pins / 32 / 32 No. of interrupts / Fixed / Fixed **Table: Difference Between 8051 and 8052 microcontroller**	**04M**
	d)	**Find the content of accumulator after execution of following code** **i) ACC=0x56>>3** **ii) ACC=0x3F<<4** **Ans:** i) ACC=0x56>>3 ACC=0x0A ii) ACC=0x3F<<4 ACC=0xF0	**02M** **02M**
	e)	**Describe the need of multitasking and inter task communication in RTOS.** **Ans:** **Need of multitasking:** Embedded system are generally specific but need to perform many task for same application let us consider example of grinding control machine A simple microcontroller program can only do one thing at a time. However, because it can do things very fast (millions of operations per second), it can be made to switch between tasks so fast that it gives an illusion of doing several things concurrently. **Need of inter task communication (IPC):** Inter task communication is a set of programming interfaces that allow a programmer to coordinate activities among different program processes that can run concurrently in an operating system. This allows a program to handle many user requests at the same time. Since even a single user request may result in multiple processes running in the operating system on the user's behalf, the processes need to communicate with each other. The Inter task communication interfaces make this possible. Each Inter task communication method has its own advantages and limitations so it is not unusual for a single program to use all of the IPC methods. Inter task communication involves sharing of data among task through sharing of memory space, transmission of data etc. It is executed using following mechanism 1. Message queues 2. Pipe 3. Remote procedure calls	**02M** **02M**
5.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Draw interfacing diagram of ADC 0808/9 with 89C51 microcontroller and describe.** **Ans:** 1. Select an analog channel by providing bits to A,B,C addresses 2. Active the ALE pin 3. Active SC (start conversion) by an L to H pulse to initiate conversion. 4. Monitor EOC to see whether conversion is finished. H to L output indicates that the data is converted and is ready to be picked up. If we do not use EOC, we can read the converted digital data after a brief time delay. 5. Active OE (output enable) to read data out of ADC chip. An L to H pulse to the OE pin will bring digital data out of the chip. Also notice that the the same as the RD pin in	**02M**

other ADC chips.

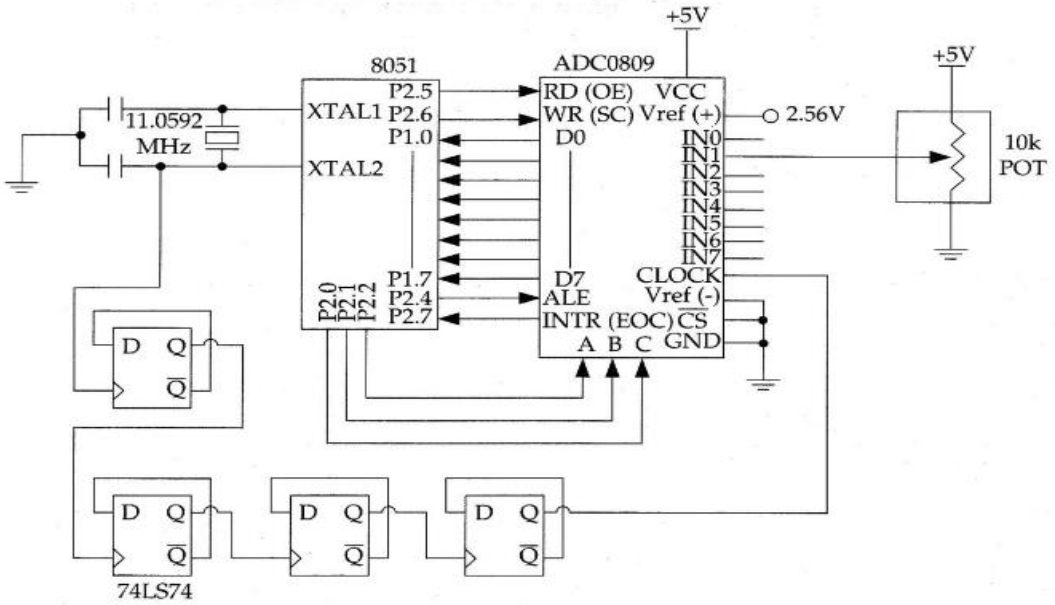


Fig: ADC 0808/0809 with 89C51 microcontroller

04M

b) State three features of I2C and USB serial communication Protocols.

Ans:

Features of I2C:

1.Independent Master, Slave and monitor functions
2. Supports both Multi-master and Multi master with slave functions.
3. Multiple I2C slave addresses supported in hardware.
4. One slave address can be selectively qualified with a bit mask or an address range in order to respond to multiple I2C bus address.
5. 10-bit addressing supported with software assist.
6. I2C operates in 3 speeds 100kbps, 400kbps and 3.4 mbps.

03M

Features of USB:

1. **Multiple device connection:** Up to 127 different devices can be connected on single USB bus.
2. **Transfer rate:** The initial USB supported 12 MBps transfer rate where USB 2.0 supports higher rate currently 60 MB/sec.
3. **Support for large range of peripherals**: Low bandwidth devices such as keyboard, mouse, joystick, and game -port, FDD.
4. **Hub architecture:** The devices are not daisy chained. Each device is connected to an USB hub. The USB hub interacts with PC on one side and peripheral on other side.
5. **Plug ability:** The USB device can be connected without powering off a PC i.e. plug and play feature in BIOS together with the device takes care of detection, handling and device recognition.
6. **Power allocation:** USB controller in the PC detects the presence or absence of the USB devices and does the allocation of power.
7. **Ease of installation:** There is only one cable. A 4-pin cable carries signals like power signal (-), signal (+), ground.
8. **Host centric:** The CPU software initiates every transaction on the USB bus. Hence The overhead on the PC increases when there is large number of peripherals involving large number of transaction.

03M

c) Differentiate between RTOS and OS.

Ans:

Sr. No.	Desktop OS	RTOS
1.	Applications are compiled separately from the OS.	Applications are compiled and linked together with the RTOS.
2.	As you turn on your desktop, only OS starts.	At boot up time, application usually gets controlled first and then it starts the RTOS.
3.	It is a less reliable system	It is a more reliable system
4.	It is not able to customize dependency on applications.	It is able to customize dependency on applications.
5.	It does not have deterministic response.	It has deterministic response.
6.	Memory required depends on the version.	Memory required (footprint) is very less.
7.	It protects itself very carefully from applications.	It does not protect itself as carefully from applications.
8.	e.g. Windows, Linux.	e.g. RT Linux, Vx Works.

Table: RTOS and OS

06M (Any 6 points)

6. Attempt any TWO of the following: 12 M

a) Draw interfacing diagram of stepper motor with 89C51 microcontroller, and write a C program to rotate motor in anticlockwise direction. Motor has step angle of 3.6 degree.

Ans:

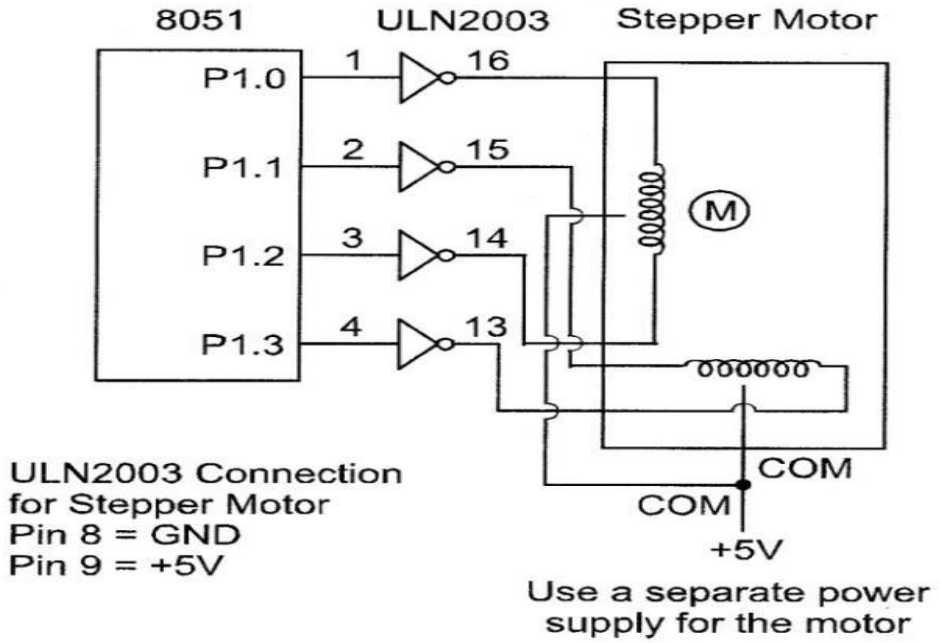


Fig: Interfacing diagram of stepper motor with 89C51 microcontroller

03M

Calculation:

Step angle=3.6°Assumerotationangle=360°

For full step No. of step=n=4 **01M**

Therefore count=angle of rotation / (no. of steps*step angle)

= 360°/(4*3.6°)

= 25

Program:

```
#include<reg51.h>
void delay (unsigned int)
void main( )
{
 unsigned char a ;
    while(1)
{
       For(a=0;a<25;a++)
   {
       P1=0x08;
      delay(50);
      P1=0x04;
      delay(50);
       P1=0x02;
       delay(50);
        P1=0x01;
        delay(50);
              }
      }
}
   void delay (unsigned int x)
{
 unsigned int  y,z;
for(y=0;y<x;y++)
for (z=0;z<1275;z++);
}
```

02M

Table: C program to rotate motor in anticlockwise direction

b) **Develop 89C51 C program to toggle all the bits of P0, P1 with 250ms delay. Use time 0, mode 1 to generate the delay the crystal frequency is 11.095 MHZ. Calculate the value of the count which is to be loaded in timer register.**

Ans:

```
#include<reg51.h>
void delay(unsigned char);
void main()
{
          TMOD=0x01;
          while(1)
          {
                    P0=0x55;
                    P1=0x55;
                    delay(5);
                    P0=0xAA;
                    P1=0xAA;
```

```
                delay(5);
        }
}

void delay(unsigned char d)
{
        unsigned char e;
        for(e=0; e<d;e++)
        {
           TH0=0x4B;
           TL0=0xFD;
           TR0=1;
        while(TF0==0);
           TR0=0;
           TF0=0;
        }
}
```

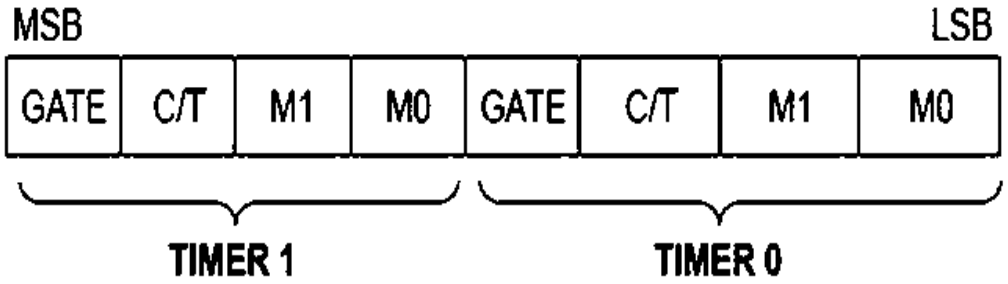
04M

Required delay = 250ms = 50ms x 5
T_{pulse} = 50ms
Count Calculation :
F_{timer} = Fosc/12 = 11.0592MHz/12 = 921.66KHz
T_{timer} = 1/F_{timer} = 1/921.66KHz = 1.085us
T_{pulse} =50ms
Count = T_{pulse}/T_{timer} = 50ms/1.085us = 46083
Count to be loaded in timer = 65536-46083 = 19453 = 4BFDH.

02M

c) Draw the format of TMOD. Describe it. Find the value of TMOD to operate as timer in Mode 1, Timer 1.

Ans:



03M

Fig: Format of TMOD

Gate - Gating Control (how to start-stop timer)
Every timer has a mean of starting and stopping.
GATE=0: Internal control-
The start and stop of the timer are controlled by way of **software.**
Set/clear the TR for start/stop timer.
SETB TR0
CLR TR0
GATE=1: External control –
The hardware way of starting and stopping the timer by **software** and **an external source**. Timer/counter is enabled only while the INT pin is high and the TR control pin is set (TR).
C/T - Counter/Timer Operation

02M

<table>
<tr><td></td><td></td><td>0 = Timer operation (clock is system clock/12)
1 = Counter operation (clock is T0 or T1 pin)
M1:M0 - Mode control
0 0 Mode 0 13-bit timer mode 8-bit THx + 5-bit TLx (x= 0 or 1)
0 1 Mode 1 16-bit timer mode 8-bit THx + 8-bit TLx
1 0 Mode 2 8-bit auto reload 8-bit auto reload timer/counter;
1 1 Mode 3 Split timer mode
Value in TMOD to Operate Timer 1 in mode 1 is 10H
<table>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>GATE</td><td>C/T</td><td>M1</td><td>M0</td><td>GATE</td><td>C/T</td><td>M1</td><td>M0</td></tr>
</table>
TIMER 1 TIMER 0</td><td>01M</td></tr>
</table>