MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

Winter – 15 EXAMINATIONS

Subject Code: **17658** **Model Answer**

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the 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.

1. a) Attempt any three of the following. 12

i) List ports of 89C51 microcontroller and list alternative functions of port-3 pins.

Ans:-Ports of 89C51 are- (2 mks)

a) Port 0
b) Port 1
c) Port 2
d) Port 3

Port 3 can be used as input or output. The alternate functions of port 3 are given in the table below

Alternate functions of Port 3 (2 mks)

P3 BIT	FUNCTION	PIN
P3.0	RXD	10
P3.1	TXD	11
P3.2	$\overline{\text{INT0}}$	12
P3.3	$\overline{\text{INT1}}$	13
P3.4	TO	14
P3.5	TI	15
P3.6	$\overline{\text{WR}}$	16
P3.7	$\overline{\text{RD}}$	17

ii) List any four different hardware units in embedded system. Write function of any two of them.

Ans:- (Any 4 components -2 mks, functions of any two-1 mks each)

Hardware components:

- Key pad
- Mic
- Processor
- Blue-tooth
- LCD Screen
- Speaker
- Head Phone
- Power supply
- Reset Circuit
- Oscillator circuit
- Driver circuit
- Data memory
- Program memory

Processor:

The processor is the heart of embedded system. The selection of processor is based on the following consideration

1. Instruction set
2. Maximum bits of operation on single arithmetic and logical operation
3. Speed
4. Algorithms processing and capability
5. Types of processor(microprocessor, microcontroller, digital signal processor, application specific processor, general purpose processor)

Power source:

Internal power supply is must. Es require from power up to power down to start time task. Also it can run continuously that is stay "On' system consumes total power hence efficient real time programming by using proper 'wait' and 'stop' instruction or disable some unit which are not in use can save or limit power consumption.

Clock / oscillator Circuits

The clock ckt is used for CPU, system timers, and CPU machine cycles clock controls the time for executing an instruction. Clock oscillator may be internal or external .It should be highly stable

iii) Draw and explain CAN bus protocol.

Ans:- (Diagram – 1mks, explanation-3 mks)

- CAN (Controller Area Network) is a serial bus system used to communicate between several embedded 8-bit and 16-bit microcontrollers.
- It was originally designed for use in the automotive industry but is used today in many other systems (e.g. home appliances and industrial machines).
- Highest Baud Rate is 1Mbit.
- CAN uses a message oriented transmission protocol.
- There are no defined addresses, just defined messages.

SOF – Start of Frame

Identifier – Tells the content of message and priority

RTR – Remote Transmission Request

IDE – Identifier extension (distinguishes between CAN standard,11 bit identifier, and CAN extended, 29 bit identifier.)

DLC – Data Length Code

Data – holds up to 8 bytes of data

CRC – "Cyclic Redundant Check" sum

ACK – Acknowledge

EOF – End of Frame

IFS – Intermission Frame Space. Minimum number of bits separating consecutive messages.

iv) Draw the pin diagram of 14 pin LCD display. State any function of each pin.

Ans:- (Pin diagram -1 mks, pin description-3 mks)

1)VCC,VSS and VEE

While VCC and VSS provide +5V and ground respectively ,VEE is used for controlling LCD contrast

2)D0-D7

The 8 – bit data pins , D0-D7 are used to send information to the LCD or read the contents of the LCD's internal registers

Explanation of RS, R/W and EN

(i) **RS :-** RS is used to make the selection between data and command register.

RS=0, command register is selected

RS=1 data register is selected.

(ii) **RW:-** R/W gives you the choice between writing and reading.

R/W=1, reading is enabled.

R/W=0 then writing is enabled.

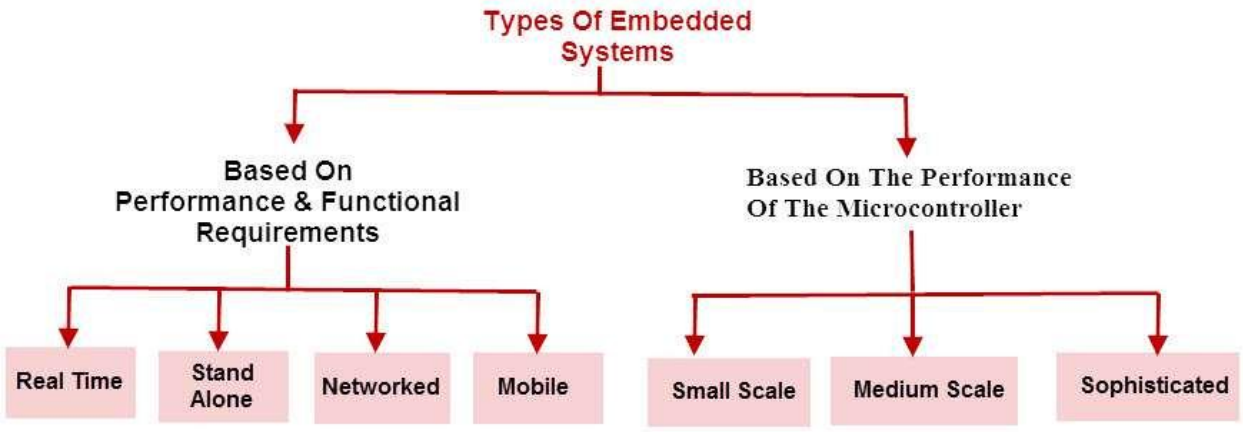
(iii) **EN :-** Enable pins is used by the LCD to latch information presented to its data pins.

When data is supplied to data pins, a high to low pulse must be applied to this pin in-order for the LCD to latch in the data present at the data pins.

b) Attempt any one of the following. **6**

i) Explain the classification of an embedded system.

Ans:- (Classification- 2mks, explanation-4 mks)



1. Stand Alone Embedded Systems

Stand-alone embedded systems do not require a host system like a computer, it works by itself. It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives or displays the connected devices. Examples for the stand alone embedded systems are mp3 players, digital cameras, video game consoles, microwave ovens and temperature measurement systems.

2. Real Time Embedded Systems

A real time embedded system is defined as, a system which gives a required o/p in a particular time. These types of embedded systems follow the time deadlines for completion of a task. Real time embedded systems are classified into two types such as soft and hard real time systems.

3. Networked Embedded Systems

These types of embedded systems are related to a network to access the resources. The connected network can be LAN, WAN or the internet. The connection can be any wired or wireless. This type of embedded system is the fastest growing area in embedded system applications. The embedded web server is a type of system wherein all embedded devices are connected to a web server and accessed and controlled by a web browser. Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP

4. Mobile Embedded Systems

Mobile embedded systems are used in portable embedded devices like cell phones, mobiles, digital cameras, mp3 players and personal digital assistants, etc. The basic limitation of these devices is the other resources and limitation of memory.

5. Small Scale Embedded Systems

These types of embedded systems are designed with a single 8 or 16-bit microcontroller that may even be activated by a battery. For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).

6. **Medium Scale Embedded Systems**

These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs. These types of embedded systems have both hardware and software complexities. For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, and JAVA, Visual C++, and RTOS, debugger, source code engineering tool, simulator and IDE.

7. Sophisticated Embedded Systems

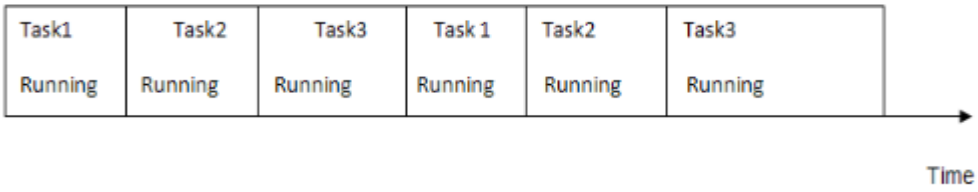
These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors. They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

ii) State the scheduling algorithms of RTOS and describe the concept of round robin scheduling.

Ans:-The various task scheduling algorithms of RTOS are-(any 4- 2 mks)

1. First in first out
2. Round-robin algorithm
3. Round robin with priority:
4. Shortest job first
5. Non Preemptive multitasking
6. Preemptive multitasking

Round Robin scheduling-(4 mks)



In the round robin algorithm, the kernel allocates a certain amount of time for each task waiting in the queue . the time slice allocated to each task is called quantum. As shown in fig . if three tasks 1,2, 3 are waiting in the queue the CPU first executes task1 then task2 then task 3 and the again task1 in round robin algorithm each task waiting in the queue is given a fixed time slice . the kernel gives control to the next task if the current task has completed its work within the time slice or if the current task has completed it allocated time

The kernel gives control to the next task if

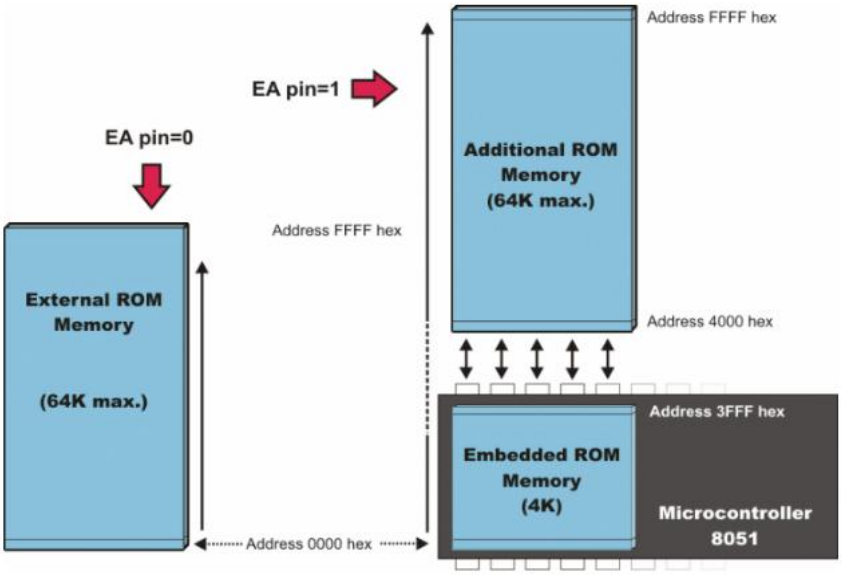
a) the current task has completed within the time slice
b) the current task has no work to do
c) the current task has completed its allocated time slice

this algorithm is very simple to implement but there is no priorities for any task. All tasks are considered of equal importance . if time critical operation are not involved then this algorithm will be sufficient . digital millimeter , microwave oven has this algorithm

2. Attempt any four of the following. 16

a) Draw the internal data memory structure of 89C51 and describe register banks.

Ans:- (Internal Register memory 2 mks, register bank explanation- 2mks)



Explanation : First 32 bytes from address 00H to 1FH are organized as four banks of eight registers each. Four registers bank are numbered as bank 0, bank 1, bank2, bank 3. Each register in a specified bank can be addressed by its name or by its RAM address. Bits RS0 and RS1 in PSW determine which bank of register is currently in use.

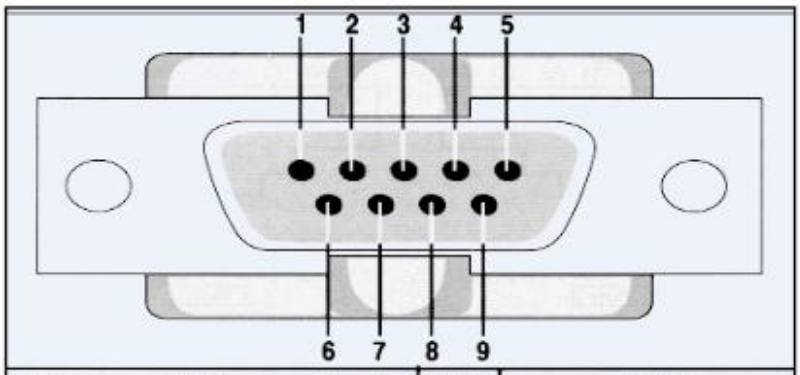
b) Write the steps for programming 8051 microcontroller to receive data serially.

Ans:- (Proper steps- 4mks)

1. TMOD register is loaded with the value 20H, indicating the use of timer 1 in mode (8-bit auto-reload) to set baud rate
2. TH1 is loaded to set baud rate
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. RI is cleared by CLR RI instruction
6. The RI flag bit is monitored with the use of instruction JNB RI,xx to see if an entire character has been received yet
7. When RI is raised, SBUF has the byte, its contents are moved into a safe place
8. To receive the next character, go to step 5

c) Draw the pin-out of RS232C and describe the function of TXD, RXD, DTE and DCE.

Ans:- (Pin diagram-2mks, Pin description- ½ mks each)



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

1) Transmit Data (TD or TXD): the line where the data is transmitted from the computer to the modem.

2) Receive Data (RD or RXD): The line where data is received transmitted from the modem to the computer

3) and 4) DCE an DTE-

The RS-232 cable has two terminal devices namely Data Terminal Equipment (DTE) and Data communication Equipment (DCE).

Both device will sends and receives the signals.

The data terminal equipment is computer terminal and data communication Equipment is modems, or controllers etc.

The DTE **transmits** data on TXD and **receives** data from the DCE on RXD.

d) Draw the interfacing diagram of 4 x 4 matrix keyboard with 89C51 microcontroller.

Ans:- (Any one relevant proper diagram- 4 mks)

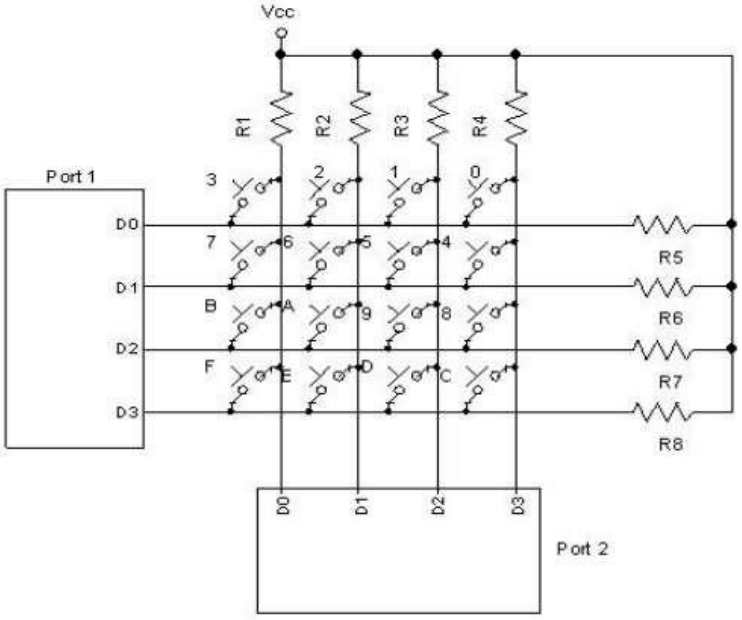
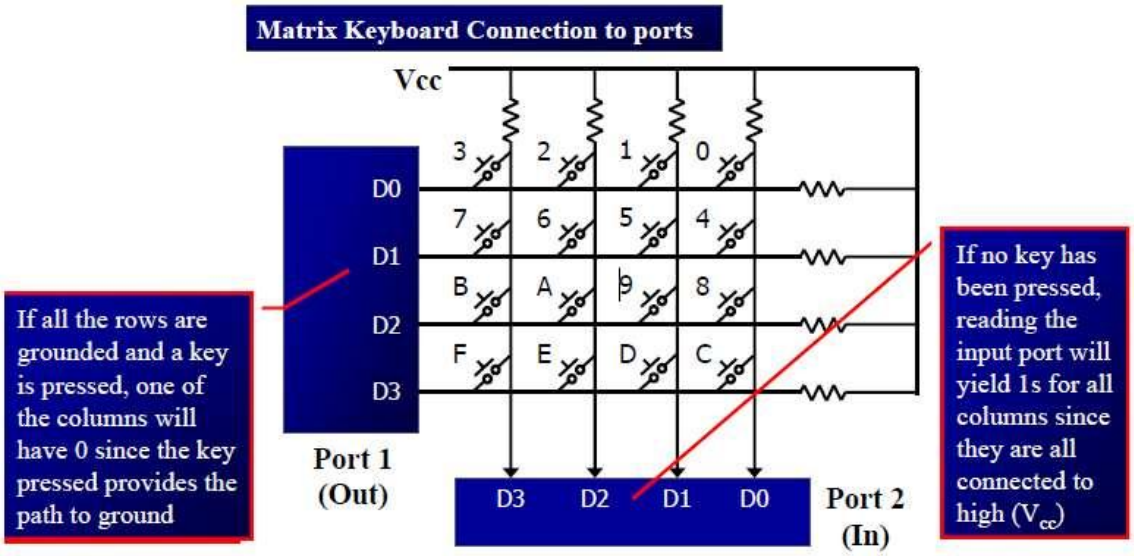


Fig. 1 Interfacing keypad to Microcontroller

OR



e) State the methods of task synchronization and explain any one in detail.

Ans:-(Any 4 Methods- 2 mks, any one explanation-2 mks)

Method of task synchronizations are

- Semaphores
- Message queue,
- Mutual Exclusion
- Dead Lock
- Mailboxes
- Message Queues

Deadlock:

A deadlock, also called as deadly embrace, is a situation in which two threads are each unknowingly waiting for resource held by other.

o Assume thread T1 has exclusive access to resource R1.

o Thread T2 has exclusive access to resource R2.

o If T1 needs exclusive access to R2 and T2 needs exclusive access to R1,

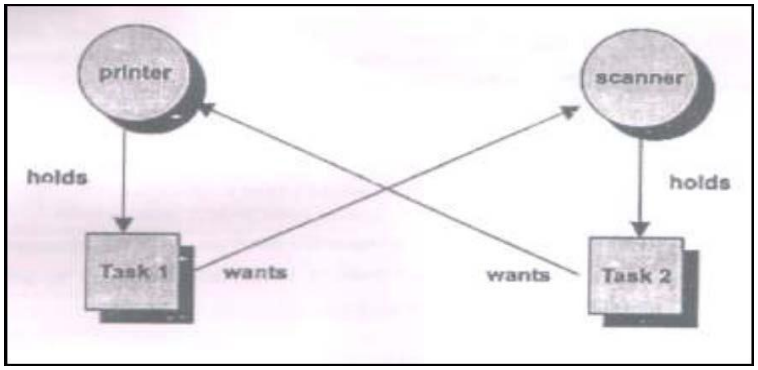
o Neither thread can continue.

o They are deadlocked.

o The simplest way to avoid a deadlock is for threads to:

- Acquire all resources before proceeding
- Acquire the resources in the same order
- Release the resource in the revere order
- Deadlock is the situation in which multiple concurrent threads of execution in a system are blocked permanently because of resources requirement that can never be satisfied.
- A typical real-time system has multiple types of resources and multiple concurrent threads of execution contending for these resources. Each thread of execution can acquire multiple resources of various types throughout its lifetime.
- Potential for deadlock exist in a system in which the underlying RTOS permits resources sharing among multiple threads of execution.

Following is a deadlock situation between two tasks.



In this example, task #1 wants the scanner while holding the printer. Task #1 cannot proceed until both the printer and the scanner are in its possession. Task #2 wants the printer while holding the scanner. Task #2 cannot continue until it has the printer and the scanner. Because neither task #1 nor task#2 is willing to give up what it already has, the two tasks are now deadlocked because neither can continue.

f) Describe any four applications of an embedded system.

Ans:- (Any four description-1 mks each)

1. Applications of Embedded Systems:

Embedded systems are used in different applications like automobiles, telecommunications, smart cards, missiles, satellites, computer networking and digital consumer electronics.

2. Embedded Systems in Automobiles and in telecommunications

- Motor and cruise control system
- Body or Engine safety
- Entertainment and multimedia in car
- E-Com and Mobile access
- Robotics in assembly line
- Wireless communication
- Mobile computing and networking

3. Embedded Systems in Smart Cards, Missiles and Satellites

- Security systems
- Telephone and banking
- Defense and aerospace
- Communication

4. Embedded Systems in Peripherals & Computer Networking

- Displays and Monitors
- Networking Systems
- Image Processing
- Network cards and printers

5 Embedded Systems in Consumer Electronics

- Digital Cameras
- Set top Boxes
- High Definition TVs
- DVDs

3. Attempt any four of the following. **16**

a) Compare between CAN and 12C protocols on following points:

Ans:- (Each point 1mks)

	I2C	CAN
Data transfer	Synchronous with 3 speeds 100kbps, 400kbps and 3.4mbps	Asynchronous with 250kbps upto 1mbps.
Number of field	07	08 (including 7 bits of frame end and 3 bits of inter frame gap)
Addressing bit	7-bit or 10-bit address	11 bit
application	To interface devises like watch dog, flash &RAM memory, Real time clock, Microcontrollers	elevator controllers, copiers, telescopes, production-line control systems, and medical instruments

b) What are different logical operators in C for 89C51? Give one example each (any four).

Ans:-The logical operators are usually performed for decision making and program control transfer. The list of logical operators supported by C are listed below-

Following table shows all the logical operators supported by C language. Assume variable **A** holds 1 and variable **B** holds 0, then –

Operator	Description	Example
NOT	~- Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.	Example Z=~x
AND	&-Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	Example P=q& r
OR	|-Called Logical OR Operator. If any of the two operands is non-zero, then the condition becomes true.	B=a|d
EXOR	^- Called logical exor operator ,o/p is true for odd no. of non zero operands.	C=a^b

Left Shift	<<	U=s<<1
Right Shift	>>	G=h>>4

c) State any four functions of RTOS.
(Any 4 functions- 4 mks)

Ans:- A real-time operating system (RTOS) is a program that schedules execution in a timely manner, manages system resources, and provides a consistent foundation for developing application code. An operating system is considered real-time if it invariably enables its programs to perform tasks within specific time constraints, usually those expected by the user.

Functions of RTOS are

1. Task management
2. Synchronization and communication
3. Memory management\
4. Time management
5. Interrupt and event handler
6. I/O management

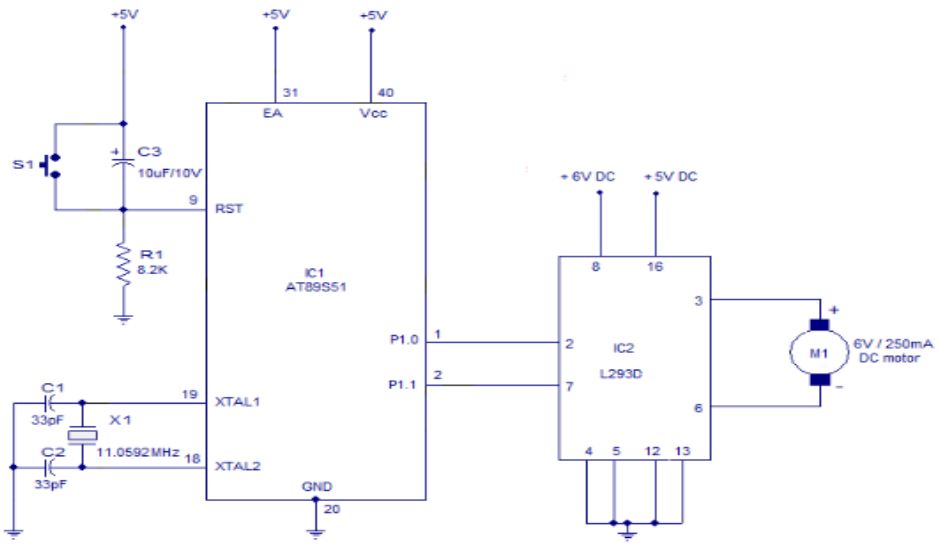
d) Classify an embedded system. Describe any two points.

Ans:- same as Q.1 b) (i)

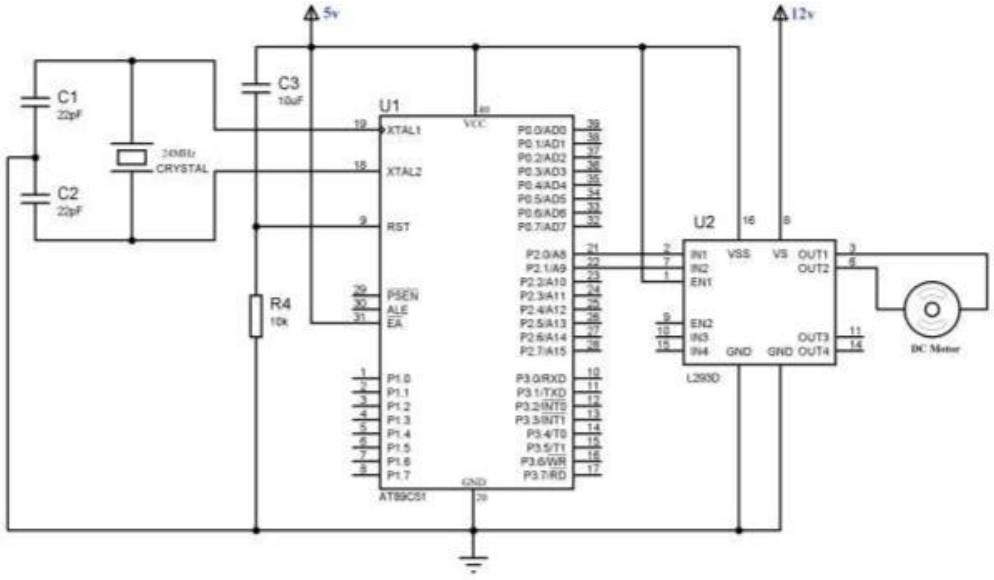
Types or classification of Embedded system- (2 mks), Explain 2 marks

e) Draw labeled interfacing diagram to interface DC motor with 8051 microcontroller.

Ans:-(Any one relevant diagram – 4 mks)



OR



Q4 a) Attempt any three of the following:

(i) List the interrupts of 89c51 microcontroller with their vector locations and order of priority.

Ans: (correct answer 4 marks)

Interrupt Source	Vector address	Interrupt priority
External Interrupt 0 – INT0	0003H	1
Timer 0 Interrupt	000BH	2
External Interrupt 1 – INT1	0013H	3
Timer 1 Interrupt	001BH	4
Serial Interrupt	0023H	5

All the 5 interrupts of 8051 has got different priorities.
Interrupts are serviced according to it's priority order.
From the table above, you can see that INT0 has the highest priority of 1 and Timer 0 comes next with priority value 2.
The order of priority works like this – consider a case where two interrupts are raised at the same time – one from INT0 and another from Timer 1 interrupt. In such a case, processor would serve the interrupt according to it's priority. In our case INT0 is of high priority (priority order 1)and Timer 1 interrupt is of low priority (priority order 4). So processor will execute ISR of INT0 first and then later, after finishing ISR of INT0, processor will begin executing ISR of Timer 1 interrupt. (explanation not compulsory)

(ii) State any four features of Bluetooth Technology.

Ans: (any four features 4 marks)

1. Short range Radio Frequency at 2.4GHz
2. Point to Point or point to multiple points protocol
3. Voice and data transmission

4. Transmit through walls up to 10 meter
5. Supports synchronous as well as asynchronous communication.

(iii) **Describe any four specifications of RTOS.Give any four examples of RTOS**.
Ans: (Four specifications - 2 marks, four examples -2 marks)

The four specifications of RTOS are:

1) Reliability: The RTOS is reliable, because it is available for all time and normally it does not fail to perform any function/operation. The reliability of system also depends on the hardware board support package and application code.

2) Predictability: In RTOS, the user knows within How much time period the RTOS is going to perform the task i.e. The RTOS has predictability. We can predict, determine how much time takes by RTOS.

3) Performance: The performance of RTOS is very fast so that it can fulfil all timing requirement.

4) Compactness: The RTOS provide compactness. It required less memory space for storage and hence can be used for portable application, like cell phone, ECG machine, etc.

Examples of RTOS are

A mobile phone, web server, microwave oven, Automated Industry, chess computer.etc

(iv) Explain the meaning of following terms with reference to embedded system:

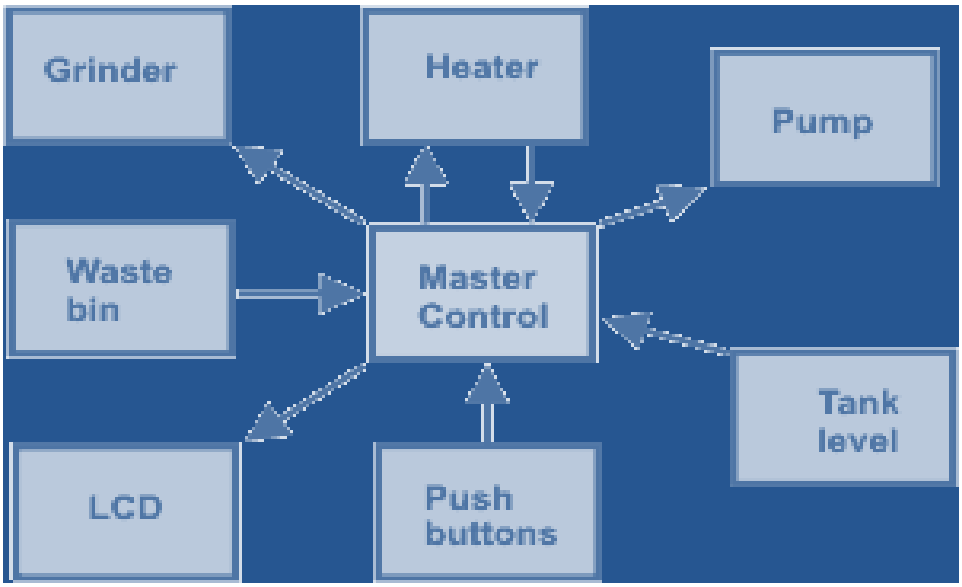
Ans: (Intertask communication explanation-2 marks, Multitasking-2 marks)

1) **Intertask communication:** Intertask 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

- **Message queues:** A message queue is an object used for intertask communication through which task send or receives messages placed in a shared memory. The queue may follow first in first out, last in first out or priority sequence. Usually a message queue comprises of an associated queue control block, name, unique ID, memory buffer, queue length, maximum message length and waiting list. A message queue of length 1 is called a mailbox.
- Pipes: A pipe is an object that provides simple communication channel used for unstructured data exchange among task. A pipe does not store multiple messages but stream of bits. Also data flow from a pipe cannot be prioritized. There are two descriptors respectively at each end of the pipe for reading and writing. Data is written in to the pipes an unstructured byte streamed via one descriptor and read from the pipe in the FIFO order from the other.
- Remote procedure call: It permits distributed computing where task can invoke the execution of the another task on remote computer.

2) **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. The question is, how do you

program it so it will divide its attention between multiple tasks, Round robin and round robin with interrupts are ways to achieves the multitasking Imagine now a program with 25 steps in the main function plus 5 sub-functions (heating water, updating the display, watching 3 push buttons). Suppose some of those sub-functions also contain a number of steps with delays. Very, very quickly the complexity of the program becomes quite impossible to untangle.



Example: Grinding control unit

b) Attempt any one of the following:

i) Write 89C51 'c' program to transfer the message "INDIA" serially at 9600 baud rate continuously. Use 8 bit data and 1 stop bit.

Ans:(correct program -6 marks)

(NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

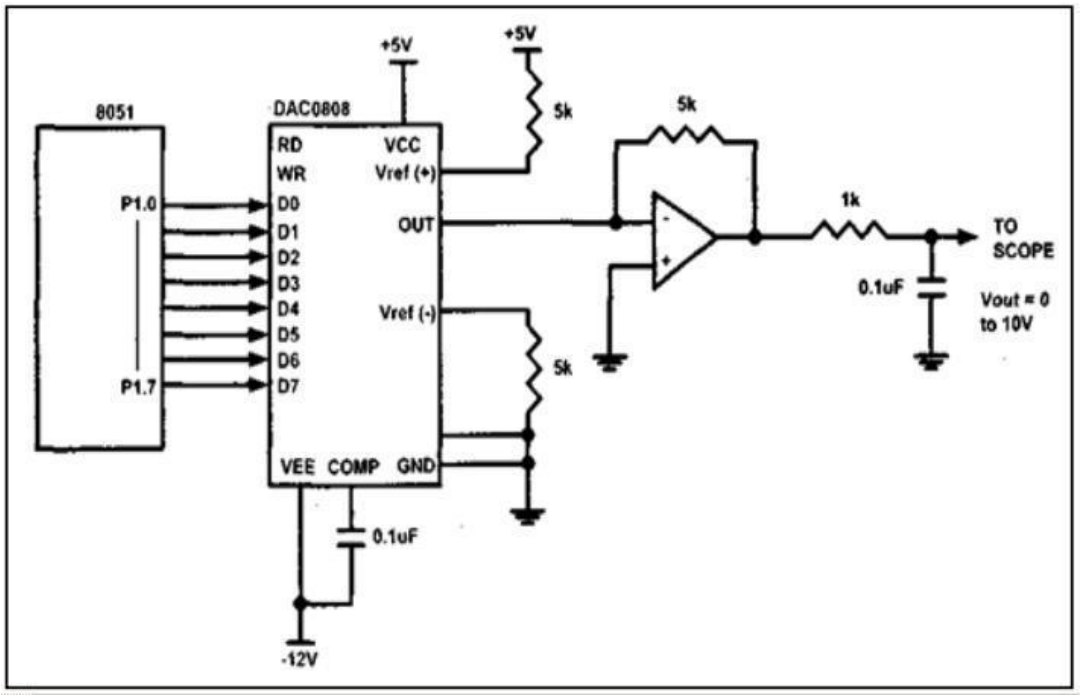
```
#include <reg51.h>
void  SerTx (unsigned char);
void main(void)
{
TMOD=0x20;          //timer 1 in mode 2
TH=0xFD;            //set baud rate to9600
SCON=0x50;          //8 bit data,1 stop bit,receiver enable
TR1=1;              //start timer 1
while (1)           //repeat transmission
{
SerTx ('I');        //send I serially
SerTx ('N')         //send N serially
SerTx('D');         //send D serially
SerTx('I');         //send I seraill y
SerTx ('A');        //send A serially
}
```

```
}
Void SerTx (unsigned char x)
{
SBUF=x;
while(TI==0);          //wait for transmission to complete
TI=0;                  //clear T1 for next transmission
}
```

ii) Draw the interfacing diagram of DAC with 89c51 mic .write a program in 'c' language to generate positive ramp voltage.

Ans: (correct diagram -3 marks, correct program-3 marks)
(NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

Circuit Diagram of interfacing DAC with 8051 Microcontroller.



Program in 'c' language to generate positive ramp voltage.

```
Include<reg51.h>
void  main ()
{
unsigned char x;
while(1)
{
for(x=0x00;x<0x7f;x++)
{
P2=x;
```

```
}
for(x=0x7f;x<0x00;x--)
{
P2=x;
}
}
}
```

(NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

Q 5. Attempt any four of the following.

a) **Describe how assembly language instruction can be included in 89c51 C programme**.
Ans:(explaination-3marks,structure-1mark)

- In C program its possible to write assembly code in the embedded software development.
- Assembly code can be called in the C program or one can write assembly code in line in their C program.
- To do so, we can use pair of inline assembly directives #pragma and #pragma endasm to include the assembly code in C program.
- The ASM directive signals the beginning merge of a block of source tex into the .SRC file generated using the SRC directive.
- The ENDASM directive is used to signal the end of the source block.

Concept structure:

```
#include<reg51.h>

void main(void)

{

    C instructions;

    #pragma asm

        Assembly intructions

    #pragma endasm

    C instructions

}
```

b) Differentiate synchronous and asynchronous communication. (any 4 points)
Ans:(4 points-4 marks)

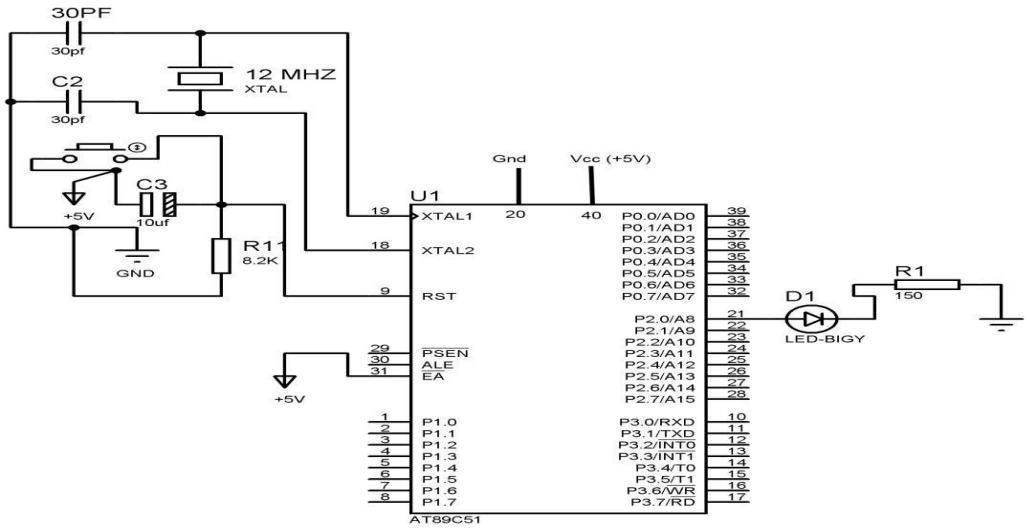
Asynchronous data transfer	Synchronous data transfer
Used to transfer one character at a time	Used to transfer group of character at a time
Data transmission rate<= 20kbps	Data transfer rate>=20kpbs
Transmission begins with start bit followed by data and then stop bit	First synchronous character is transmitted and then data is transmitted
Different clock sources are required for transmitter and receiver	Same clock sources are required or transmitter and receiver
	

c) Draw labelled diagram to interface LED to P2.1 of 89c51. Write programme to turn ON & OFF this LED after some delay.

Ans: (LED should be connected to P2.1)

(Diagram -2 marks,program-2 marks)

(NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)



Program:

```
# include <reg51.h>

sbit  LED=P2^1;

void delay (int)    // delay function

void main (void)
```

```
{
while (1)        // infinite while loop
{
  LED=1;              //led is on
  delay (100);     //delay
 LED= 0 ;           //led is off
delay (100);       //delay


}
}
void delay(int k)
{
int i,j;
for(i=0;i<k;i++);
for(j=0;j<100;j++);
}
```

**(NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)**

d) Explain the concept of starvation and deadlock on RTOS.

Ans: (Deadlock explaination-2 marks,Starvation explaination-2 marks)

Deadlock:

A deadlock, also called as deadly embrace, is a situation in which two threads are each unknowingly waiting for resource held by other.

o Assume thread T1 has exclusive access to resource R1.

o Thread T2 has exclusive access to resource R2.

o If T1 needs exclusive access to R2 and T2 needs exclusive access to R1,

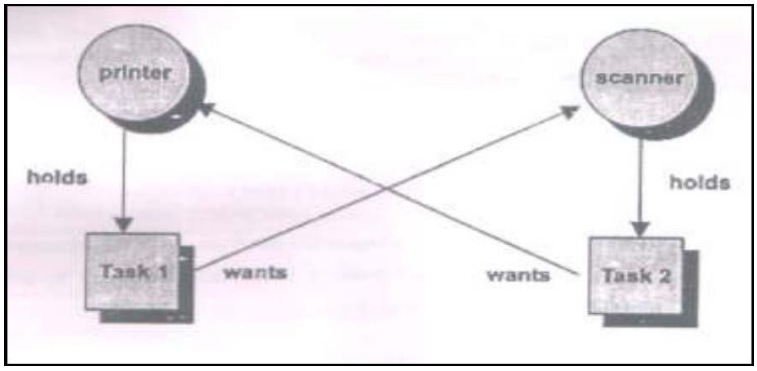
o Neither thread can continue.

o They are deadlocked.
o The simplest way to avoid a deadlock is for threads to:

- Acquire all resources before proceeding
- Acquire the resources in the same order

- Release the resource in the revere order
- Deadlock is the situation in which multiple concurrent threads of execution in a system are blocked permanently because of resources requirement that can never be satisfied.
- A typical real-time system has multiple types of resources and multiple concurrent threads of execution contending for these resources. Each thread of execution can acquire multiple resources of various types throughout its lifetime.
- Potential for deadlock exist in a system in which the underlying RTOS permits resources sharing among multiple threads of execution.

Following is a deadlock situation between two tasks.



In this example, task #1 wants the scanner while holding the printer. Task #1 cannot proceed until both the printer and the scanner are in its possession. Task #2 wants the printer while holding the scanner. Task #2 cannot continue until it has the printer and the scanner. Because neither task #1 nor task#2 is willing to give up what it already has, the two tasks are now deadlocked because neither can continue.

Methods of avoid Deadlock:

- Mutual exclusion
- Hold & wait or resource holding
- No preemption
- Circular wait

Starvation:

- Multiple shared resources have multiple semaphores associated with them.
- The semaphores are all independent of one another. If one task takes semaphore x them another task can take semaphore Y, without blocking.
- But this property can lead to deadlock.
- For example, suppose task 1 calls function to take semaphore X & get it but fore it can call the function to take semaphore Y the RTOS stops the execution of task 1 & runs task 2.
- The task2 calls the function to take semaphore & gets it.
- But when task2 calls the function to take the semaphore X, it is blocked since, the Task1 has already taken the semaphore X.
- The RTOS now switch back to task 1, which now calls the function to take semaphore Y. since task2 has semaphore Y, task 1 is also now blocked.
- There is no escape from this for either tasks, since now both are blocked, waiting for semaphores that the other has.
- This problem due to multi-tasking is called Starvation, where a task is denied necessary resources repeatedly without those resources the task will never be completed.

- Deadlock is a special case of Starvation.

e) **Describe the program downloading to ISP/IAP**.

Ans(ISP-2 marks,IAP-2 marks)

In system Programing (ISP) is a method where a programmable device is programmed after the device is soldered in target hardware.

- ISP reduces the extra handling step required in the manufacturing process to program the devices on an external programmer before placing them on target hardware.
- ISP supports the functions of some programmable logic devices ,and other programmable electronic devices to be programed during installed in a complete system.
- The advantage of ISP is that the manufacturer of embedded devices can integrate programing and testing into a production stage instead of a separate programming stage before assembling a the system.
- Basically, the device supporting ISP have an internal circuitry to generate required programming voltage from the system supply voltage, and communicate with the programmer using a serial protocol.

In-system programming the following components are needed:

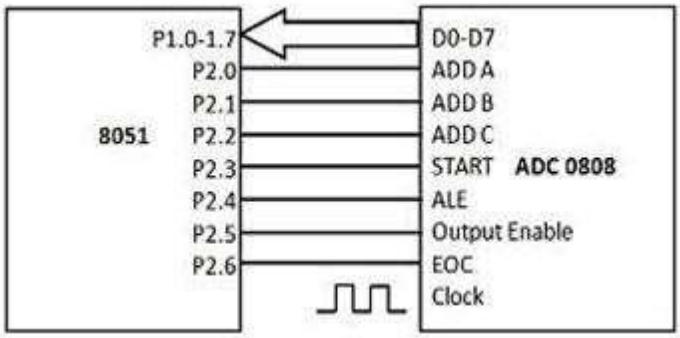
1. A software tool running on the desktop PC which is able to control the programming interface through standard port.
2. A programming adapter which is used to connect the programming interface to any standard port available at the desktop PC, like USB, RS-232, printer port etc.
3. A programming interface such as SPI i.e. serial programming interface.

IAP: In- Application programming

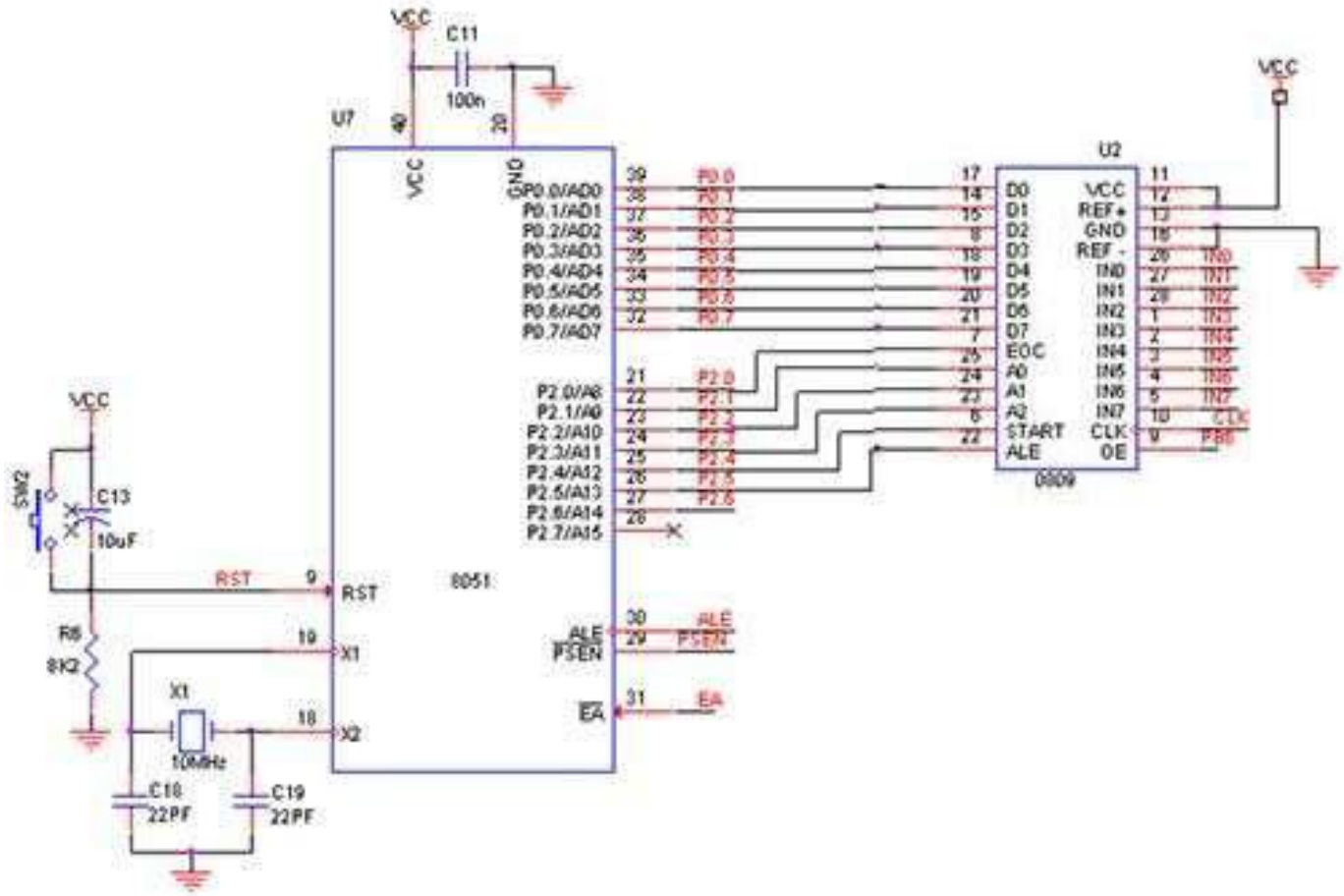
- In- Application programming is a method where a programmable device is programmed after the device is soldered in target hardware while the application code is running
- Implementation of application with IAP is possible , because embedded system can be reprogrammed remotely without a service personal to be actually present.
- Basically, IAP is used with external flash memory which is a separate component mounted on the circuit board.
- But in some case addition programme memory is available from which the microcontroller can execute code.

f) **Draw the interfacing diagram of ADC with 8051.**

Ans: (correct diagram-4 marks)



Or



Q **6. Attempt any four of the following.**

a) Compare between assembly language programme with an embedded C with reference to following points. I) Execution time ii) time for coding iii) hex file size iv) debugging
Ans: (each correct difference-1 mark)

PARAMETER	ALP	Embedded C
Execution time	Faster(Less execution time required)	Slower(More execution time required)
Time for coding	More time is required for coding	Less time required for coding and code is more efficient
Hex file size	less	More
Debugging	Not so easy	Easy

b) Draw and explain USB protocol.
Ans:(correct explaination-3marks,diagram-1 mark)

Universal Serial Bus (USB) is a set of interface specifications for high speed wired communication between electronics systems peripherals and devices with or without PC/computer.

USB offers users simple connectivity. It eliminates the mix of different connectors for different devices like printers, keyboards, mice, and other peripherals. That means USB-bus allows many peripherals to be connected using a single standardized interface socket. Another main advantage is that, in USB environment, DIP-switches are not necessary for setting peripheral addresses and IRQs. It supports all kinds of data, from slow mouse inputs to digitized audio and compressed video.

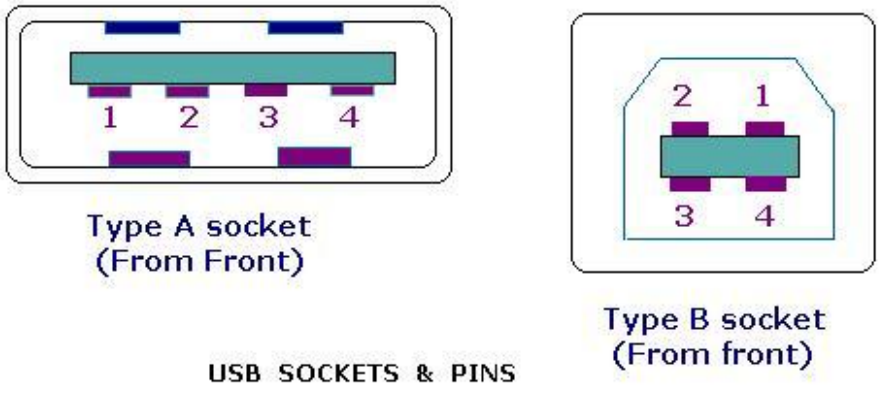
USB also allows hot swapping. The "hot-swapping" means that the devices can be plugged and unplugged without rebooting the computer or turning off the device. That means, when plugged in, everything configures automatically. So the user needs not worry about terminations, terms such as IRQs and port addresses, or rebooting the computer. Once the user is finished, they can simply unplug the cable out, the host will detect its absence and automatically unload the driver. This makes the USB a plug-and-play interface between a computer and add-on devices.

USB sends data in serial mode i.e. the parallel data is serialized before sends and de-serialized after receiving.

The benefits of USB are low cost, expandability, auto-configuration, hot-plugging and outstanding performance. It also provides power to the bus, enabling many peripherals to operate without the added need for an AC power adapter.

Connecting a USB device to a computer is very simple -- you find the USB connector on the back of your machine and plug the USB connector into it. If it is a new device, the operating system auto-detects it and asks for the driver disk. If the device has already been installed, the computer activates it and starts talking to it.

The USB standard specifies two kinds of cables and connectors. The USB cable will usually have an "A" connector on one end and a "B" on the other. That means the USB devices will have an "A" connection on it. If not, then the device has a socket on it that accepts a USB "B" connector.



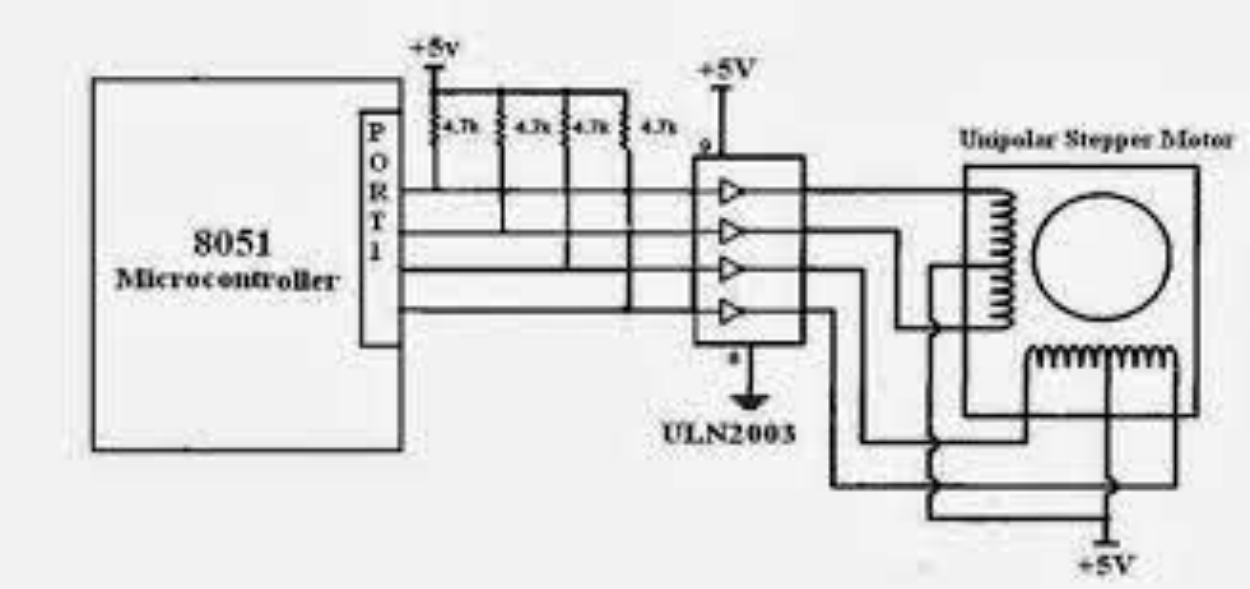
USB SOCKETS & PINS

USB pin connections

Pin No	Signal	Color of the cable
1	+5V power	Red
2	- Data	White / Yellow
3	+Data	Green / Blue
4	Ground	Black/Brown

c) **Draw the interfacing diagram of stepper motor with 8051 microcontroller**.

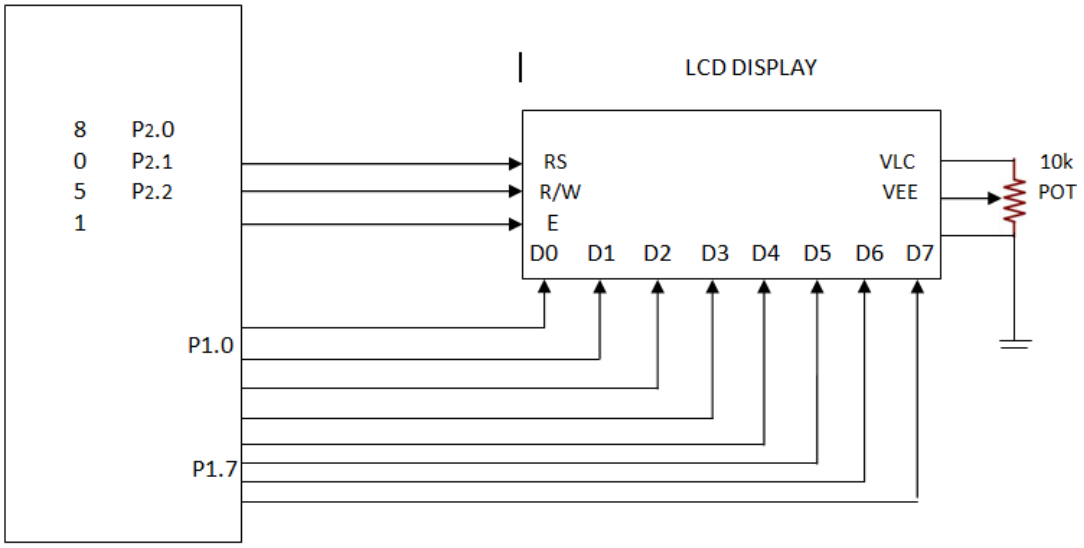
Ans;(correct diagram-4 marks)



Any other suitable diagram also can be drawn

d)Draw the interfacing diagram of LCD display with 8051 microcontroller.

Ans:(correct diagram-4 marks)



e)Write 89c51 "C" program to toggle bits of port P0 continuously with a 200 millisecond delay

Ans:(correct program-4marks)

**(NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)**

```
#include <reg51.h>
void  Add_delay (unsigned  int);
void main (void)
{
while(1)                  //repeat loop
{
P0=0xff;                  //toggle all bits of port0
Add_delay (200);          //add delay
P0=0x00;                  //toggle all bits of port0
Add_delay (200);          //add delay
}
}
void Add_delay (unsigned  int delay)
{
unsigned int x,y;
for(x=0;x<delay;x++)
for (y=0;y<275;y++);
}
```

**(NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of student .**