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.

Q1.a) Attempt any three of the following:- **12M**

i) State four differences between RISC and CISC architectures.

Ans:-

(Any four points – 01M each)

RISC	CISC
1.Reduced instruction set computer	1.Complex instruction set computer.
2.Instruction set is simple and limited.	2.Instruction set is very large.
3.Instruction set is not flexible, hence the program is long.	3.Instruction set is flexible, hence the program is short
4.Provides few addressing modes normally register.	4.Provides many addressing modes
5.Large memory is required	5.Less memory required
6.Provides large number of registers	6.No. of registers are less.
7.Processor architecture & control unit is simple	7.Processor architecture and control unit is complicated.
8.Instruction are shorter, hence execution speed is fast.	8.Instructions are lengthy hence execution speed is slow.
9.External memory is accessed rarely	9.External memory is accessed frequently
10.Each instruction required few bus cycle.	10.Each instruction requires many bus cycle.
e. g. ARM, ATMEL, AVR, MIPS, PIC, POWER PC, SUNSPARC, i960, etc.	Interx86, Motorola, 68000 series.

ii) List any four software development tools used in an embedded system and state the function of each .

Ans:

(List :02M and Function of any four tools: ½ M each)

Software development tools:

- Compiler
- Cross assembler
- Cross compiler
- Locators
- Loaders
- Simulators
- Debugger
- Integrated development environment (IDE)

Explanation :

Compiler:

It is a computer program that transforms the source code written in a programming or source language into another computer language i.e. target language i.e. binary code known as object code.

Cross assembler:

It is useful to convert object codes for microcontrollers or processor to other codes for another microcontrollers or processor and vice versa.

Cross compiler:

It is used to create executable code other than one on which the compiler is run. They are used to generate executable for embedded systems or multiple platforms.

Linker/Locator:

- It is used for relocation process.
- It is done during compilation also it can be done at run time by a relocating loader.
- It is a program that takes one or more objects generated by compiler and combines them into a single executable program.

Simulators:

- A simulator is the s/w that simulates an h/w unit like emulator, peripheral, network and I/O devices on a PC.
- It defines a processor or processing device as well as various versions for the target system.
- Monitors the detailed information of as source code part with labels and symbols during the execution for each single step.
- Provides the detailed information of the status of memory RAM and simulated ports, simulated peripheral devices of the defined target system.

Integrated Development Environment (IDE):-

An IDE is a software application that provides comprehensive facilities to computer programmers for software development. An IDE consists of:

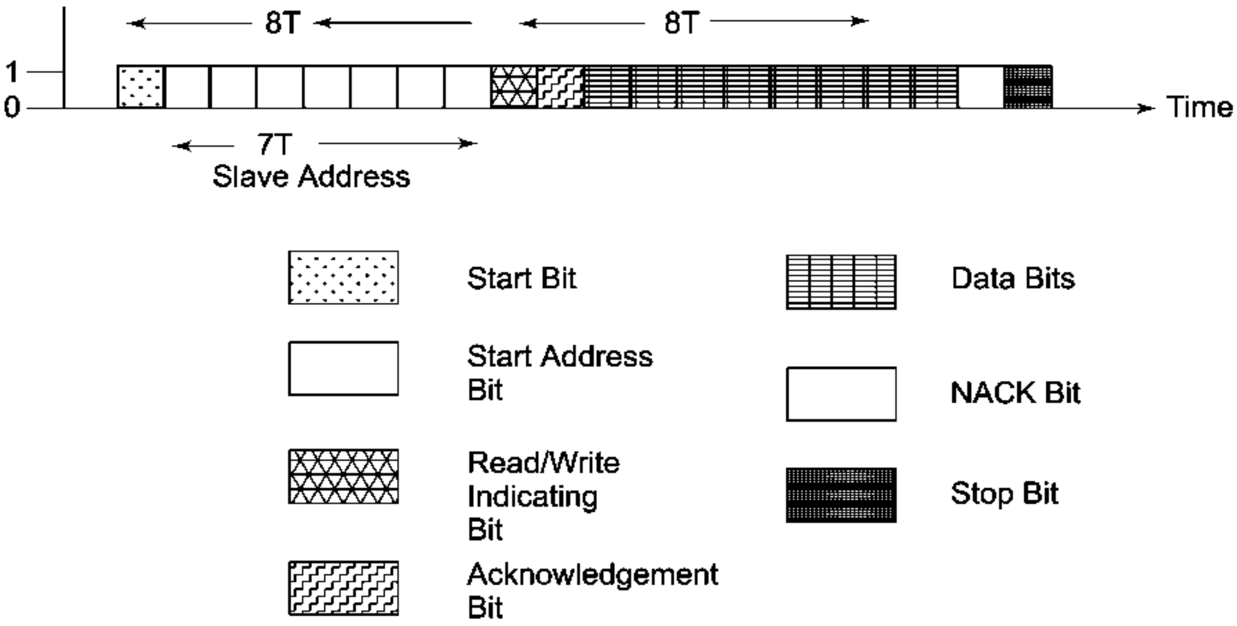
- A source code editor.
- A compiler and or interpreter.
- Build automation tools.
- A debugger.

IDE is dedicated to a specific programming language, allowing a feature set that most closely matches the programming paradigms [model] of the language. IDE’s typically present a single program in which all development is done. This program typically provides many features for authoring, modifying, compiling, deploying, and debugging software.

iii) Draw the format of I2C and explain each field in brief.

Ans:-

(Format- 02M,Explanation-02M)



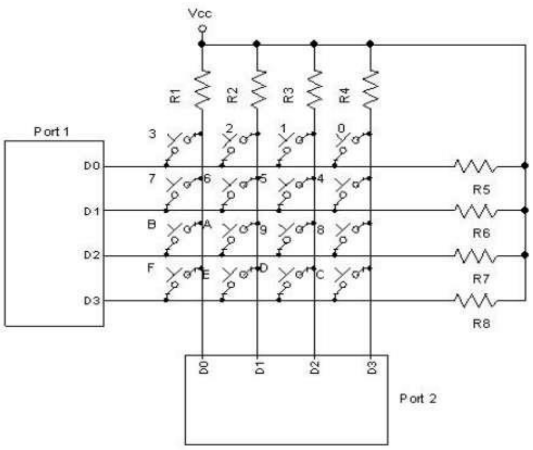
- First field of 1 bit – START bit = one
- Second field of 7 bits – address field. It defines the slave address, which is being sent the data frame(of many bytes) by the master
- Third field of 1 control bit – defines whether a read or write cycle is in progress
- Fourth field of 1 control bit – defines whether is the present data is an acknowledgment (from slave)
- Fifth field of 8 bits – I^{2}C device data byte (transmitted by either master or slave)
- Sixth field of 1 bit – bit NACK (negative acknowledgment) from the receiver (when master is receiving). If active the acknowledgment is expected from the slave
- Seventh field of 1 bit – stop bit send by master

iv) Draw the interfacing diagram of 4 X 4 matrix keypad with 89C51microcontroller .

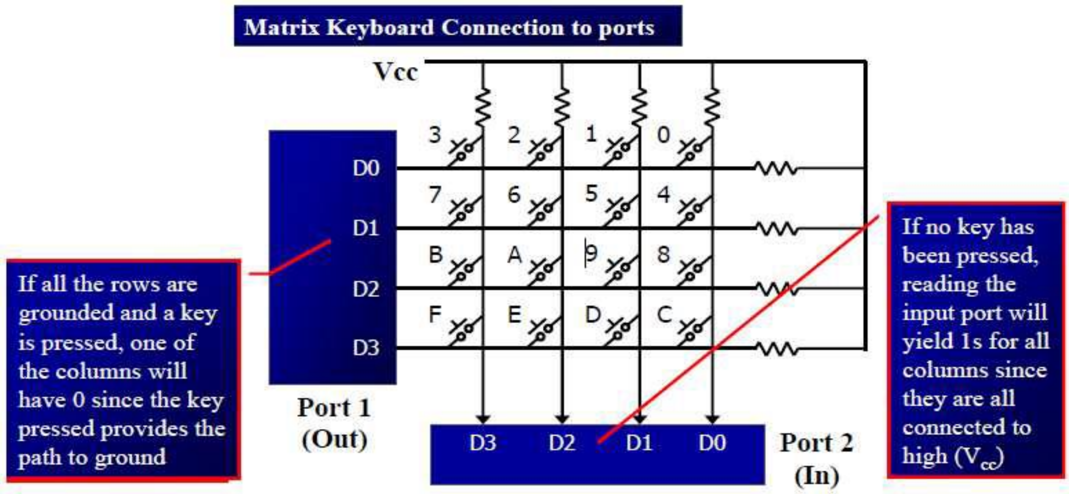
Ans:-

(Diagram 03M, labeling 01M)

Diagram:



OR


Matrix Keyboard Connection to ports
Vcc
3 2 1 0
7 6 5 4
B A 9 8
F E D C
D0
D1
D2
D3
Port 1 (Out)
D3 D2 D1 D0
Port 2 (In)
If all the rows are grounded and a key is pressed, one of the columns will have 0 since the key pressed provides the path to ground
If no key has been pressed, reading the input port will yield 1s for all columns since they are all connected to high (Vcc)

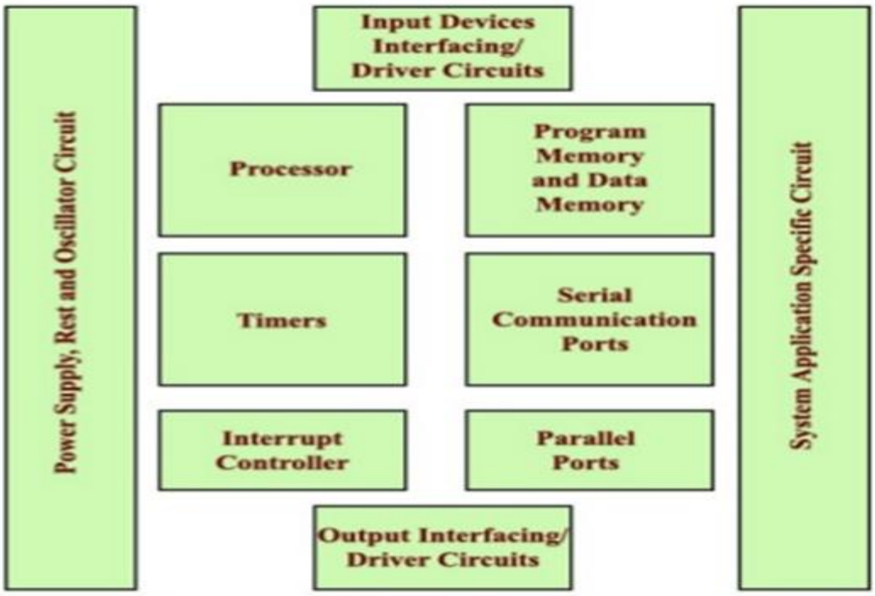
Q1.b) Attempt any one of the following :- 06M

i) **Draw block diagram of embedded system and describe any four hardware units of embedded system.**

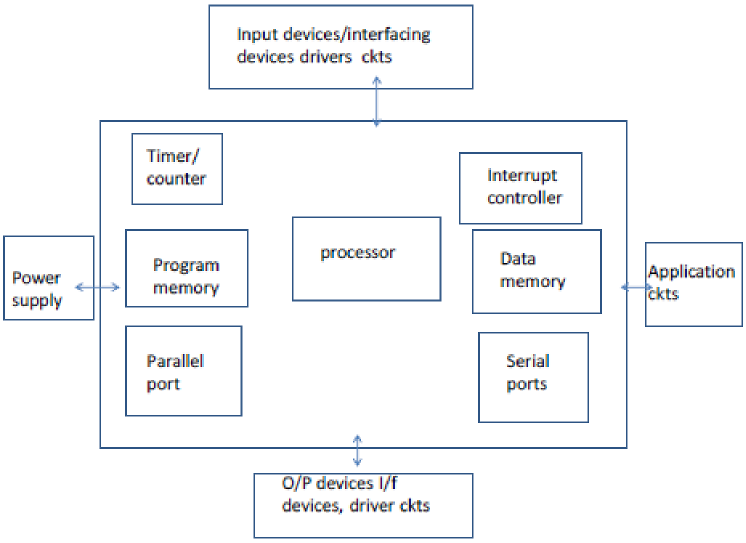
Ans:-

(Block diagram of embedded system-02M, description of any four hardware units- 01M each.)

Block diagram:


Power Supply, Rest and Oscillator Circuit
Input Devices Interfacing/ Driver Circuits
Processor
Program Memory and Data Memory
Timers
Serial Communication Ports
Interrupt Controller
Parallel Ports
Output Interfacing/ Driver Circuits
System Application Specific Circuit

OR



Processor:

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

- Instruction set
- Maximum bits of operation on single arithmetic and logical operation
- Speed
- Algorithms processing and capability
- 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

Real time clock(RTC):

It require to maintain scheduling various tasks and for real time programming RTC also use for driving timers, counters needs in the system.

Reset Ckts and power on reset:

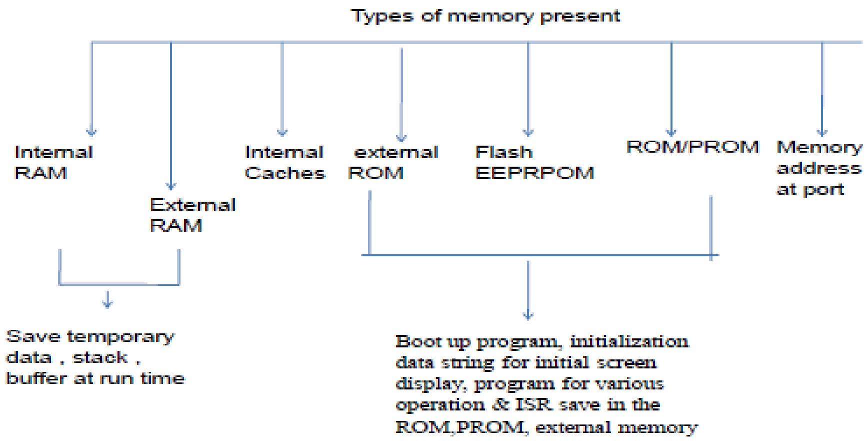
Reset process starts executing various instruction from the starting address. The address is set by the processor in the program counter. The reset step resent and runs the program in the following way.

- System program that execute from beginning
- System boot up program
- System initialization program

Peripherals:

The peripheral devices are provided on the embedded system boards for an easy integration. Typical devices include serial port, parallel port, network port, keyboard and mouse ports, a memory unit port and monitor port. Some specialized embedded systems also have other ports such as CAN-bus

Memories:


Types of memory present
Internal RAM
External RAM
Internal Caches
external ROM
Flash EEPRPOM
ROM/PROM
Memory address at port
Save temporary data , stack , buffer at run time
Boot up program, initialization data string for initial screen display, program for various operation & ISR save in the ROM,PROM, external memory

ii) **List scheduling algorithms of RTOS. Describe concept of pre-emptive multitasking scheduling algorithm of RTOS with suitable example.**

Ans:-

(List-02M,description of multitasking scheduling algorithm- 02M)

1. First in first out
2. Round-robin algorithm
3. Shortest job first
4. Non Pre-emptive multitasking
5. Pre-emptive multitasking

Pre-emptive multitasking :

- In pre-emptive multitasking the highest priority task always given the CPU time .
- If lower priority task is presently running and higher priority task is in ready to run state then the running task is pre-empted and the higher priority task is executed .
- The Kernel restore the state of current task and pass control of the CPU to the higher priority task.
- Again after some time the high priority task may releases the CPU and the low priority task executed.
- It is used in commercial embedded systems.

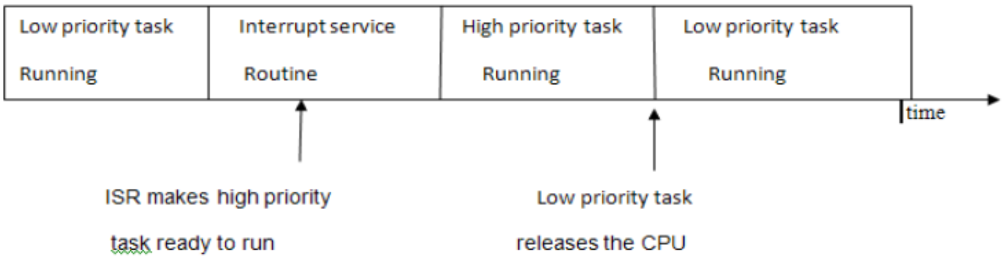
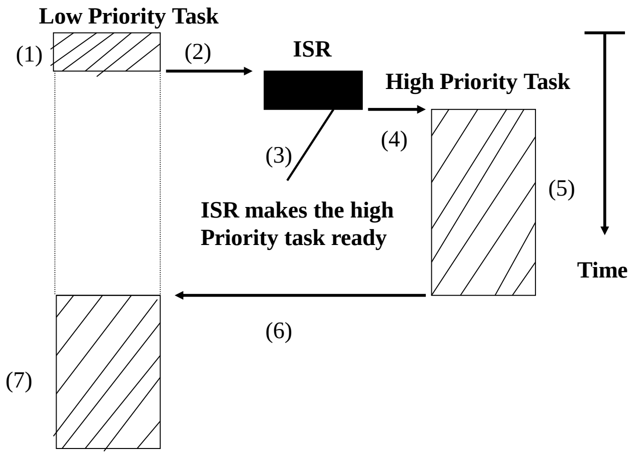

Low priority task Running
Interrupt service Routine
High priority task Running
Low priority task Running
time
ISR makes high priority task ready to run
Low priority task releases the CPU

Fig Preemptive multitasking

OR

PREEMPTIVE KERNEL



Q2. Attempt any four of the following:- 16M

a) Draw the format of TMOD. Describe the function of each bit.

Ans:-

(Format- 02M,description- 02M)

MSB **LSB**

GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
← Timer 1 →				← Timer 0 →			

GATE : Starts and stops Timer / Counter by means of a signal provided to the pin $\overline{\text{INT1}}$ / $\overline{\text{INT0}}$

1 – Timer / counter operates only if the bit INT1 / INT0 is set

0 – Timer / Counter operates regardless of the state of the bit INT 1 /INT0

$C/\overline{T}$: Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1, M0 : These two bits selects the Timer operating modes.

M1	M0	Mode	Description
0	0	0	13-bit timer
0	1	1	16-bit timer
1	0	2	8-bit auto-reload
1	1	3	Split mode

b) **Write 89c51 "C" program to toggle all the pins of port P2 continuously with a 400 millisecond delay**

Ans:

(correct program- 04M)

NOTE: Program may change. Student can also use the other logic.

Please check the logic and understanding of students.

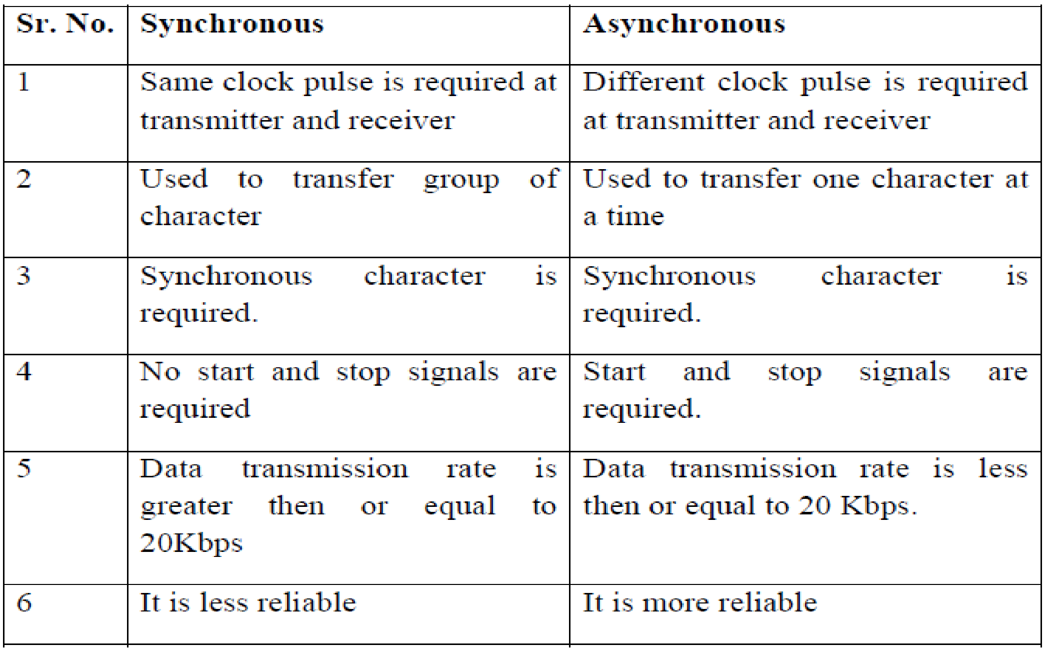
```
#include <reg51.h>
void  delay (unsigned  int);
void main (void)
{
while(1) //repeat loop
{
P2=0xff; //toggle all bits of port0
delay (400); //add delay
P2=0x00; //toggle all bits of port0
delay (400); //add delay
}
}
void  delay (unsigned inti)
{
Unsignedintx,y;
for(x=0;x<i; x++)
for (y=0;y<1275;y++);
}
```

c) **Differentiate between synchronous communication and asynchronous communication.(any four points)**

Ans:-

(Any four points -01M each.)

Note: - Ant valid points can be considered



Sr. No.	Synchronous	Asynchronous
1	Same clock pulse is required at transmitter and receiver	Different clock pulse is required at transmitter and receiver
2	Used to transfer group of character	Used to transfer one character at a time
3	Synchronous character is required.	Synchronous character is required.
4	No start and stop signals are required	Start and stop signals are required.
5	Data transmission rate is greater then or equal to 20Kbps	Data transmission rate is less then or equal to 20 Kbps.
6	It is less reliable	It is more reliable

d) Write C program to read the status of key connected to P1.3.If the key is pressed, turn on the LED connected to P3.5 for 20 msec.

Ans:-

(Correct program 04M)

```
#include <reg51.h>
void delay(unsigned int);
Sbitsw =P1^3;
Sbit  led=P3^5;

void main(void)
{
P1=0xFF; //make P1 input port
led = 0
while (1)
{
if  sw=0
{
led=1;
Delay(20);
led=0;
}
}
}
voiddelay (unsigned inti)
{
Unsignedintx,y;
for(x=0;x<i; x++)
for (y=0;y<1275;y++);
}
```

e) Explain the concept of deadlock with suitable schematic.

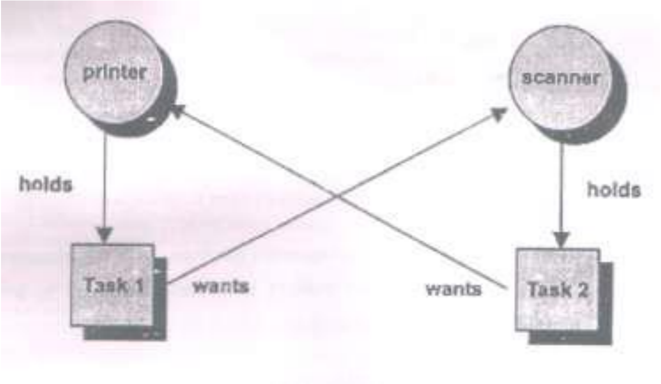
Ans:- **(Explanation 03M, schematic 01M)**

Deadlock:

A deadlock is a situation in which two threads are each unknowingly waiting for resource held by other.

- Assume thread T1 has exclusive access to resource R1.
- Thread T2 has exclusive access to resource R2.
- If T1 needs exclusive access to R2 and T2 needs exclusive access to R1,
- neither thread can continue.
- They are deadlocked.
- 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 the following characteristics of embedded system:
(i) Processor power
(ii) Memory
(iii) Reliability
(iv) Safety

Ans:- **(01M each characteristic)**

i) Processor power:

In order to improve the utility of the embedded system, the power consumption of the entire system has to be minimized. Since the major computations are done by the embedded processor/controller, energy minimization of the processor is also vital for the total power reduction. The amount of power consumed by the system, determines the lifetime of a battery, or the cooling requirements of the IC, since more power means more heat.

ii) Memory:

- Designer has to make an estimate of the memory requirement and must make provision for expansion.
- In a system, there are different types of memories: RAM, ROM, EPROM, PROM, etc.
- Secondary storage devices like HDD can be embedded into the system like mobile.
- Flash memory can be used instead of secondary memory. Hence, we can load NT in embedded system. E.g. Embedded Linux Os can be loaded into wristwatches.

iii) Reliability:

- It is measure of how much % you can rely upon the proper functioning of system.
- Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTP) are used in determining reliability.

- MTBF gives the frequency of failures in hours/weeks/months.
- MTTR specifies how long the system is allowed to be out of order following a failure.

iv) Safety:

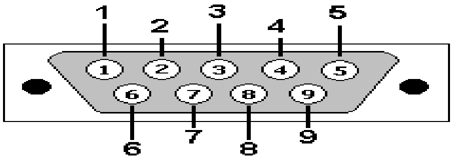
- It deals with possible damages that can happen to the operators, public and the environment due to breakdown of embedded system,
- Or due to the emission of radioactive or hazardous materials from embedded products.
- The probability that the system will not cause harm.
- Safety analysis is a must in product engineering to evaluate the anticipated damages and determine best course of action.

Q3) Attempt any Four of the following 16M

a) Draw pin out of RS-232 (DB-9) connector and the interface diagram of RS232 with 89C51
Ans:-

(RS232 pin diagram 01M, 01M for pin name, interfacing 02M)

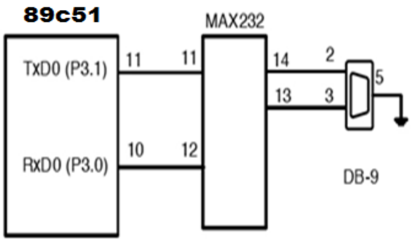
RS232 pin diagram:-



Pin names:-

Pin No.	Name	Dir	Notes/Description
1	DCD	IN	Data Carrier Detect. Raised by DCE when modem synchronized.
2	RD	IN	Receive Data (a.k.aRxD, Rx). Arriving data from DCE.
3	TD	OUT	Transmit Data (a.k.aTxD, Tx). Sending data from DTE.
4	DTR	OUT	Data Terminal Ready. Raised by DTE when powered on. In auto-answer mode raised only when RI arrives from DCE.
5	SGND	-	Ground
6	DSR	IN	Data Set Ready. Raised by DCE to indicate ready.
7	RTS	OUT	Request To Send. Raised by DTE when it wishes to send. Expects CTS from DCE.
8	CTS	IN	Clear To Send. Raised by DCE in response to RTS from DTE.
9	RI	IN	Ring Indicator. Set when incoming ring detected - used for auto-answer application. DTE raised DTR to answer.

Interfacing :-



b) Find the content of port after execution of the following codes:

i) P2 = 0 x74 >>3

ii) P3 = 0 x 04 | 0 x 68;

Ans:-

i) P2 = 0 x74 >>3 **02M**

P2=0X74>>3

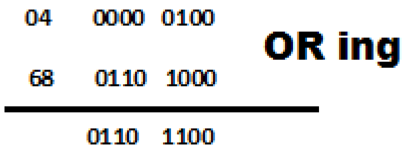
This indicates shift 74 by right 3 times.

74H = 0111 0100

After shifting by 3 times towards right we get

0000 1110 = 0X0E

ii) P3 = 0 x 04 0 x 68; **02M**



```
04    0000  0100
                        OR ing
68    0110  1000
--------------------
      0110  1100
```

Ans P3=0X6C

c) Explain inter-task communication with reference to RTOS

Ans:- **04M**

Inter task communication:-

Inter task communication involves sharing of data among tasks through sharing of memory space, transmission of data, etc.. Some of the mechanisms available for executing Inter Task communications are :

a) Message queue

b) Pipes

c) Remote procedure call

Inter task communications is executed using following mechanisms

- **Message queues -**

A message queue is an object used for inter task communication through which task send or receive messages placed in a shared memory.

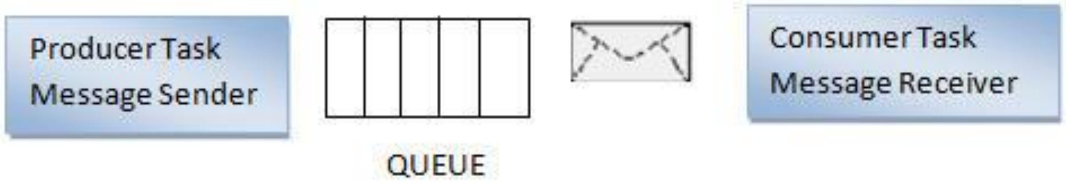
The queue may follow:

1) First In First Out (FIFO)

2) Last in First Out(LIFO) or

3) Priority (PRI) sequence.

Usually, a message queue comprises of an associated queue control block (QCB), name, unique ID, memory buffers, queue length, maximum message length and one or more task waiting lists. A message queue with a length of 1 is commonly known as a mailbox.



- **Pipes –**

A pipe is an object that provide simple communication channel used for unstructured data exchange among tasks. A pipe does not store multiple messages but stream of bytes. Also, data flow from a pipe cannot be prioritized.

- **Remote procedure call (RPC) -**

It permits distributed computing where task can invoke the execution of another task on a remote computer.

d) Define embedded system. List any two advantages and disadvantage if embedded system.

Ans:- (Definition 02M, Advantage ½ M each point, disadvantage ½ M each point)

Definition:-

An Embedded system is a combination of computer hardware and software. As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller. The Embedded system hardware includes elements like user interface, Input/output interfaces, display and memory, etc. Generally, an embedded system comprises power supply, processor, memory, timers, serial communication ports and system application specific circuits.

Advantages (any two):-

1) Design and Efficiency:

The central processing core in embedded system is generally less complicated, making it easier to design. The limited function required of embedded system allows them to design to most efficiently perform their function.

2) Cost:

The streamline make-up of most embedded system allows their parts to be smaller less expensive to produce.

3) Accessibility:

If something goes wrong with certain embedded systems they can be too inaccessible to repair. This problem is addressed in the design stage, so by programming an embedded system. So that it will not affect related system negatively when malfunctioning.

4) Maintenance:

Embedded systems are easier to maintain because the supplied power is embedded in the system and does not required remote maintenance.

5) Redundancies:

Embedded system does not involve the redundant programming

Disadvantages (any two):-

1. Difficult to change configurations and features: -

Once an embedded system is deployed (or finalized), it will be difficult to change its configuration - both its hardware and software. Remote update of software is possible provided the capability is included. Hence, proper requirement analysis is a must before deployment. Hardware configuration change will be much more trickier which may require existing boards be completely replaced. I have seen this happen and it is not pretty.

2. Issue of scalability:-

Because it is difficult to change configuration, an embedded system cannot be easily scaled up as demand/scope changes. Said so, embedded systems can be designed to scale up for example using expansion ports or networking etc. This means it must be decided before hand during design phase for scale up provisions.

3. Limitation of hardware:-

With a limited memory or computing capability in most embedded systems, there is always a limitation (or an upper limit) on our software design(upgrade). Be always aware of "Memory" and "Speed".

4. Applied for a specific purpose:-

By definition, embedded systems are constrained in their objectives. If it is decided to "rehash" an existing embedded system for a completely different purpose, it will normally result in significant change(s) in either or both its hardware or/and software.

e) Write a ‘C’ program to transfer the message “MSBTE” serially at 9600 baud rate continuously

Ans:-

(04M for program with comments)

<u>NOTE</u>: Student may use another method for program any other correct method can be consider

```
#include <reg51.h>
void main(void)
{
unsigned char text[ ] = “MSBTE”;          initialize array
TMOD = 0x20;                              Initialize     timer 1  in mode2
TH1 = 0xFD;                               baud rate 9600
SCON = 0x50;                           start serial communication (  8bit , 1 stop bit  ,  REN )
TR1 = 1;                              start timer 1
for(i=0;i<6;i++)                      Read array and transmit  serially till end
{
SBUF = text[i];
while(TI==0);                         check interrupt
TI = 0;                            clear interrupt
}
}
```

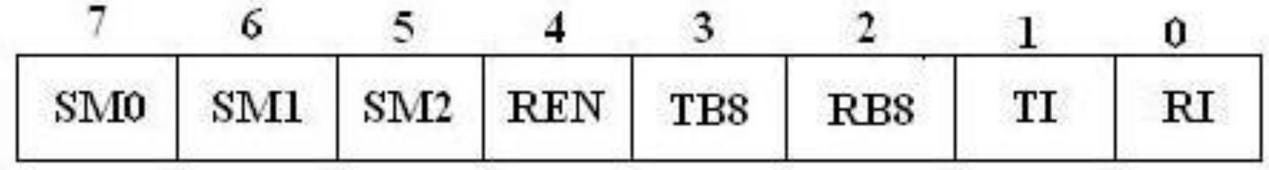
Q4 a) Attempt any three of the following: 12M

i) Draw the format of SCON register and explain all the bits

Ans:-

(Format 02M, explanation 02M)

SCON Register format:



7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Explanation:

SM0, SM1 bits in SCON are used to define the type of the serial communication, baud rate and framing.

SM0	SM1	Mode	Function
0	0	0	Shift register, baud rate = f/12, **Synchronous** serial communication mode
0	1	1	8-bit UART; baud = variable , asynchronous serial communication mode
1	0	2	9-bit UART; baud = f/32 or f/64, asynchronous serial communication mode
1	1	3	9-bit UART; baud = variable, asynchronous serial communication mode

SM2 –

Bit is used for multiprocessor communication. Set or cleared by the program to enable multiprocessor communications in mode 2 and 3. When set to 1 an interrupt is generated if bit 9 of the received data is a 1; no interrupt is generated if bit 9 is 0;

REN –

Receiver enable bit. To accept reception of data this bit must be 1;

TB8 –

Transmitted bit 8.

RB8 –

Received bit 8.

TI –

Transmit interrupt flag. This will be enabled when all bits in the transmitted buffer is shifted out.

RI –

Receive interrupt flag. This will be enabled when a character is received in the receiver buffer.

ii) List four features of each of the following:

1) Bluetooth

2) Zigbee

Ans:- **(Bluetooth features ½ M each, Zigbee features ½ M each)**

1) Bluetooth features (any 4):-

- IEEE Standard 802.15.1
- Frequency (GHz) 2.4
- Maximum raw bit rate (Mbps) 1-3
- Typical data throughput (Mbps) 0.7-2.1
- Maximum (Outdoor) Range (Meters) 10 (class 2), 100 (class 1)
- Relative Power Consumption Medium
- Example Battery Life in Days
- Network Size 7
- Cost Low

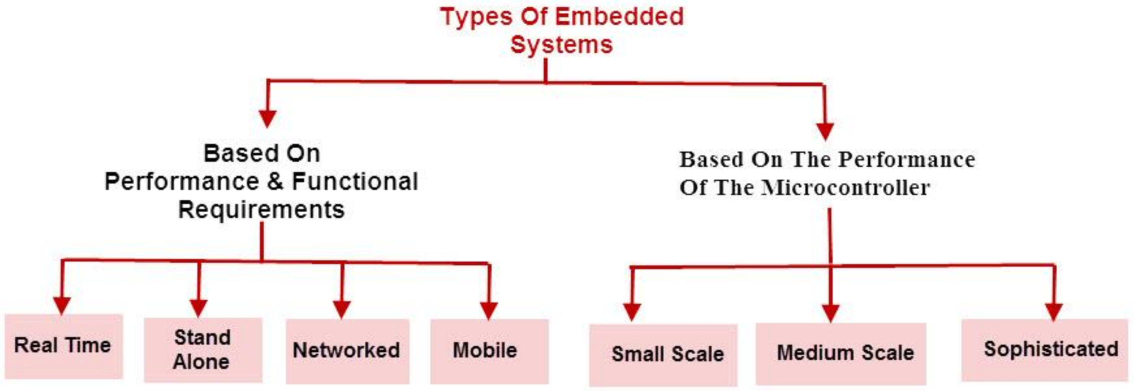
2) Zigbee features (any4):-

- IEEE Standard 802.15.4
- Frequency (GHz) 0.868, 0.915, 2.4
- Maximum raw bit rate (Mbps) 0.250
- Typical data throughput (Mbps) 0.2
- Maximum (Outdoor) Range (Meters) 10-100
- Relative Power Consumption Very low
- Example Battery Life Months to years
- Network Size 64,000+

iii) Classify an embedded system. Describe any two types.

Ans:- **(Classification 02M, descriptions any two 01M each)**

Classification of Embedded Systems


Types Of Embedded Systems
Based On Performance & Functional Requirements
Based On The Performance Of The Microcontroller
Real Time
Stand Alone
Networked
Mobile
Small Scale
Medium Scale
Sophisticated

.

Descriptions:- (Any two)

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 and 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++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.

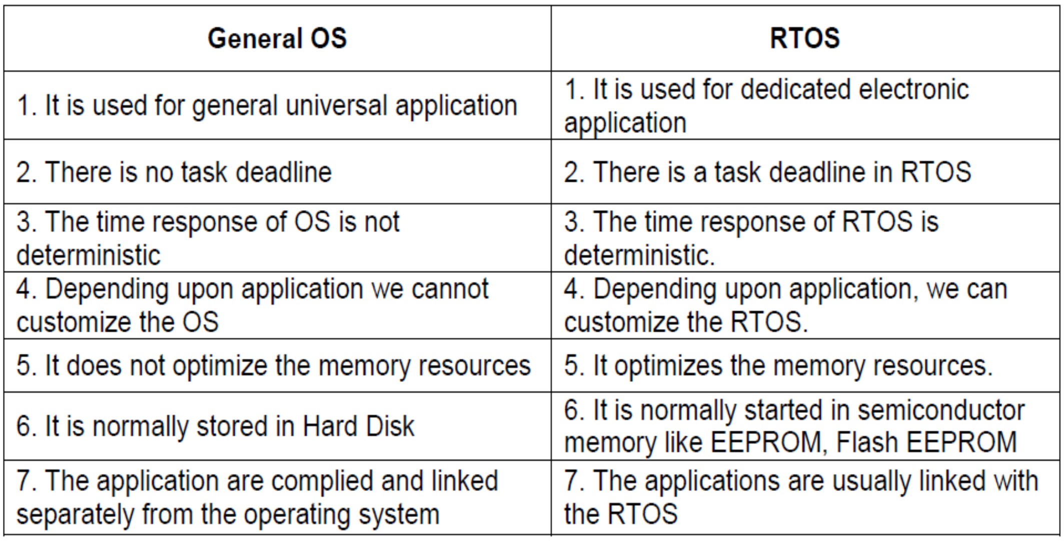
7) Sophisticated Embedded Systems

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

iv) Differentiate RTOS with desktop OS (any four points)

Ans:- **(Any 4 correct points 01M each)**

.



General OS	RTOS
1. It is used for general universal application	1. It is used for dedicated electronic application
2. There is no task deadline	2. There is a task deadline in RTOS
3. The time response of OS is not deterministic	3. The time response of RTOS is deterministic.
4. Depending upon application we cannot customize the OS	4. Depending upon application, we can customize the RTOS.
5. It does not optimize the memory resources	5. It optimizes the memory resources.
6. It is normally stored in Hard Disk	6. It is normally started in semiconductor memory like EEPROM, Flash EEPROM
7. The application are complied and linked separately from the operating system	7. The applications are usually linked with the RTOS

Q4 b) Attempt any one of the following: - 06M

i) Write a 'C' program to generate a square wave of 100Hz on P1.3. Also draw the output Observed on P1.3

Ans :-(Calculation 02M, program 03M, neat square waveform as output clearly marking Ton and Toff 01 M)

Assume Crystal value = 12 MHz (Crystal value can be assumed as 11.0592MHz also)

- Look at the following steps for 100 Hz frequency calculations with 12 MHz
- The period of the square wave = 1 / 100Hz
    - = 0.01 s.
    - =10mSec
- The high or low portion of the square wave = Time period / 2
    - = 10ms / 2
    - = 5mSec.
- Timer clock Frequency is = XTAL / 12
    - = 12 MHz / 12
    - = 1 MHz
- Timer clock period is = 1/ Timer Frequency
    - = 1 / 1 MHz
    - = 1 µSec
- Counter = Delay / timer clock period
    - =5mSec / 1 µSec
    - = 5000
    - =50X100
- Timer Reload value = Maximum Count – Counter

- = 65536 – 5000
- = (60536)d

- Timer Reload value in HEX = (60536)d
  - = (EC78) h.
- TL0 = 0x78   and TH0 = 0xEC.

//C language program to generate square wave over Port Pin P1.3 using timer0

```
        #include <Intel\8052.h>
        #include <standard.h>
        Void T0M1delay (void);              //Timer 0, Mode 1(16 bit timer)
        SBIT   OUTPUT       P1^3;           // Initialize Port pin P1.3 as output
        Void main ()
        {
                While (1)
                {

                        OUTPUT= ~ OUTPUT;       // toggle P1.3
                        T0M1delay ();           // delay of 5mSec
                }

        }
    void T0M1delay ()               // Timer 0, Mode 1(16 bit timer) - delay of 5mSec
        {
                TMOD= 0x01;         // Timer 0, Mode 1(16 bit timer)
                TL0    = 0x78;      //Load TL0 = 78h
                TH0    = 0xEC;      //Load TH0 = ECh
                TR0    = 1;         //Run the timer 0
                while (TF0 = = 0);  // Wait for TF0 to overflow
                TR0    = 0;         //Stop the timer 0
                TF0    = 0;         //Clear TF0
        }
```

OR

Alternate Method

Assume crystal frequency as 11.0592 MHz
Timer input CLK frequency = 11.0592 MHz / 12

Timer input CLK period = 12/11.0592 = 1.085 microsec
The period of the square wave = 1 / 100Hz

- = 0.01 s.
- =10mSec

The high or low portion of the square wave = Time period / 2

- = 10ms / 2

- = 5mSec.

For Mode 1

(65535-count+1) X1.085 microsec = 5msec

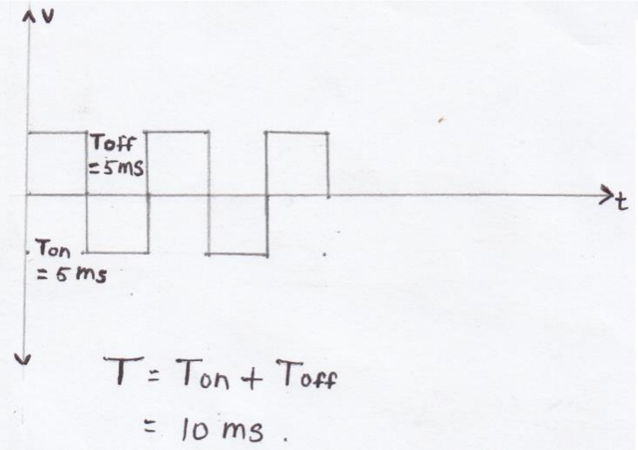
65536-count = 5msec / 1.085 microsec = 4608.29

count = 65536-4608.29 = 60927.71 = EE00 H

ie TL0=00 H and TH0 = EEH

Program:

```
        #include <reg51..h>
        void T0M1delay (void);                    //Timer 0, Mode 1(16 bit timer)
        sbitoutbit = P1^3;                // Initialize Port pin P1.3 as output
        void main (void)
        {
                while (1)
                {
                        outbit = ~ outbit;                  // toggle P1.3
                        T0M1delay ();                        // delay of 5mSec
                }
        }

    void T0M1delay (void)                 // Timer 0, Mode 1(16 bit timer) - delay of 5mSec
        {
                TMOD= 0x01;           // Timer 0, Mode 1(16 bit timer)
                TL0     = 0x00;           //Load TL0 = CEh
                TH0     = 0xEE;           //Load TL0 = FFh
                TR0     = 1;              //Run the timer 0
                while (TF0 = = 0);        // Wait for TF0 to overflow
                TR0     = 0;              //Stop the timer 0
                TF0     = 0;              //Clear TF0
        }
```



T = Ton + Toff
= 10 ms .

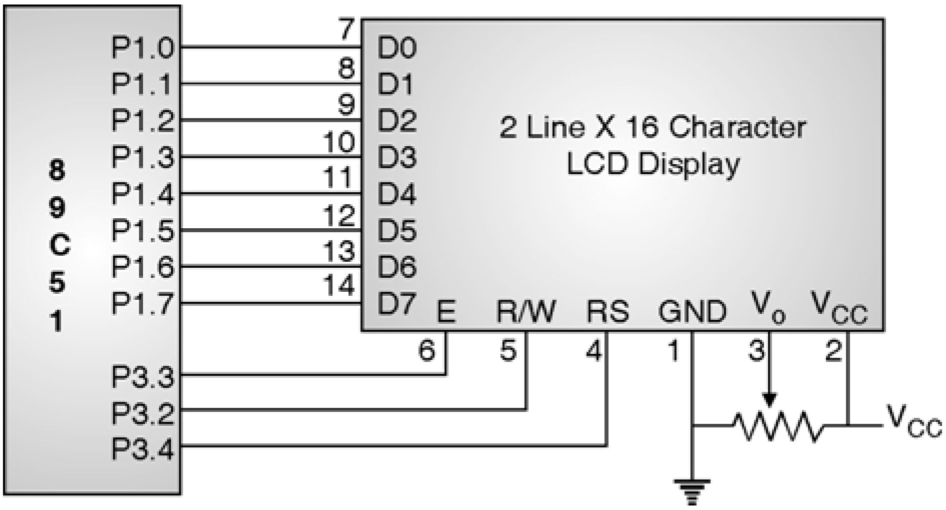
ii) Draw the circuit diagram to interface LCD with 89C51. Write 'C' program to send letters 'M' , 'D', and 'E' to the LCD display

Ans:-

(interfacing 02M, program : 04)

Note:* *Student may use any other method

Interfacing LCD with 89C51:-



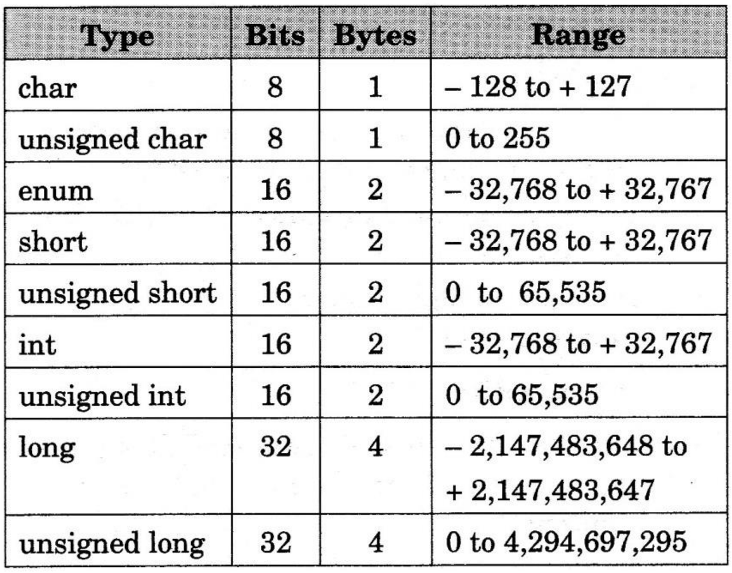
```
#include<reg51.h>
sbitrs=P3^4;
sbitrw=P3^2;
sbit en=P3^3
void delay(unsigned int);
voidlcdcmd(unsigned char);
voidlcddta(unsigned char);
void main()
{
rs=0;
rw=0;
en=0;

lcdcmd(0x38);
delay(10);
lcdcmd(0x0e);
delay(10);
lcdcmd(0x06);
delay(10);
lcdcmd(0x01);
delay(10);
lcdcmd(0x80);
delay(10);
lcddta('M');
delay(10);
lcddta('D');
delay(10);
lcddta('E');
delay(10);
while(1)
{
}
}
voidlcddta(unsigned char dta)
{
rs=1;
rw=0;
P1=dta;
en=1;
delay(2);
en=0;
}
voidlcdcmd(unsigned char cmd)
{
rs=0;
rw=0;
P1=cmd;
en=1;
delay(2);
en=0;
}
void delay(unsigned int t)
{
Unsignedintx,y;
for(x=0;x<=t;x++)
for(y=0;y<=1275;y++);
}
```

Q5) Attempt any four of the following: 16M

a) **State any four data types used in embedded system, with their value range**.

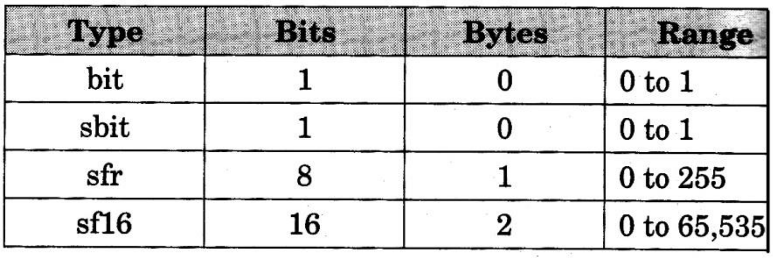
Ans:- **(01M each with their correct range)**

Note:*-*Any four data types with their correct ranges



Type	Bits	Bytes	Range
char	8	1	– 128 to + 127
unsigned char	8	1	0 to 255
enum	16	2	– 32,768 to + 32,767
short	16	2	– 32,768 to + 32,767
unsigned short	16	2	0 to 65,535
int	16	2	– 32,768 to + 32,767
unsigned int	16	2	0 to 65,535
long	32	4	– 2,147,483,648 to + 2,147,483,647
unsigned long	32	4	0 to 4,294,697,295

The Keil or SPJ C compiler provides several new data types to support a microcontroller and embedded systems applications:



Type	Bits	Bytes	Range
bit	1	0	0 to 1
sbit	1	0	0 to 1
sfr	8	1	0 to 255
sf16	16	2	0 to 65,535

b) List the serial and wireless communication protocols. And describe 802.11

Ans:- **(List -02M and describe 802.11 – 02M)**

Serial Communication protocols:

1. Inter-Integrated Circuit bus [I^2C]
2. Controlled Area Network [CAN]
3. Universal Serial Bus [USB]
4. Serial Peripheral Interface [SPI]
5. Synchronous Serial Protocol [SSP]

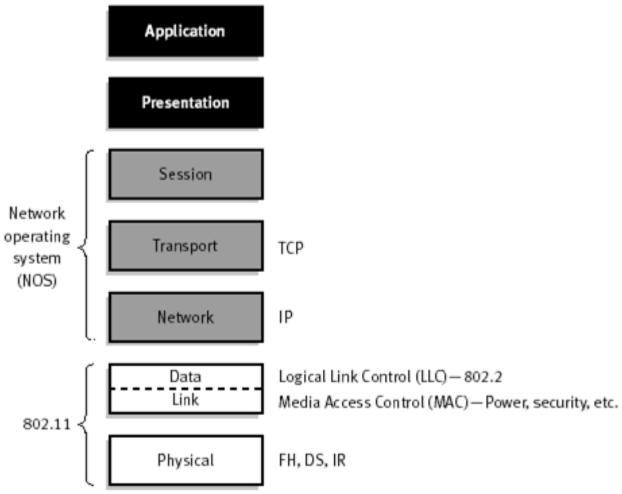
Wireless Communication Protocols:

1. Infrared [IrDA]
2. Bluetooth
3. Zigbee
4. Wi-Fi

802.11 Wi-Fi:

- Wi-Fi is a local area wireless technology that allows an electronic device to exchange data or connect to the internet using 2.4 GHz UHF and 5 GHz SHF radio waves.
- The Wi-Fi Alliance defines Wi-Fi as any "wireless local area network (WLAN) products that are based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards".

- However, since most modern WLANs are based on these standards, the term "Wi-Fi" is used in general English as a synonym for "WLAN". Only Wi-Fi products that complete Wi-Fi Alliance interoperability certification testing successfully may use the "Wi-Fi CERTIFIED" trademark
- Many devices can use Wi-Fi, e.g., personal computers, video-game consoles, smartphones, digital cameras, tablet computers and digital audio players. These can connect to a network resource such as the Internet via a wireless network access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can comprise an area as small as a single room with walls that block radio waves, or as large as many square kilometers achieved by using multiple overlapping access points.
- The Modulation techniques used are either OFDM,DSSS or FHSS. 802.11n supports a data transfer rate of up to 600 Mbps, the data transfer rate of 802.11b is 11Mbps


Application
Presentation
Session
Transport
Network
Data Link
Physical
Network operating system (NOS)
TCP
IP
Logical Link Control (LLC)—802.2
Media Access Control (MAC)—Power, security, etc.
802.11
FH, DS, IR

Advantages:

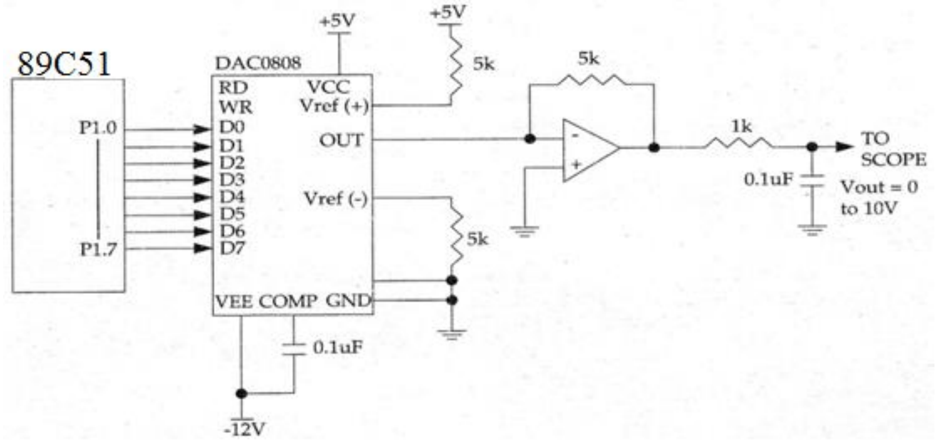
1. Convenient due to wireless nature.
2. Mobility.
3. Productivity as an employees' work can be accomplished from any convenient location.
4. Easy Deployment.
5. Expandability: No additional wiring required for more number of clients.

Disadvantages:

1. Security: Some of the more commonly utilized encryption methods are less secure.
2. Range insufficient for larger structure.
3. Less Reliability as wireless networking signals are subject to a wide variety of interference.
4. Low speed.

c) Draw the interfacing diagram of DAC 0808 with 89C51 microcontroller.

Ans:- **(correct diagram – 04M)**



d) State the methods of task synchronization. Describe semaphore with suitable example.

Ans: **(list methods -01M; Semaphore – 01M; example- 02M)**

The methods of task synchronization are:

- Semaphore
- Message queue.
- Mutual exclusion.
- Dead lock.
- Mailboxes.
- Message Queues.

Semaphore:

A semaphore is a single variable that can be incremented or decremented between zero and some specified maximum value. The value of the semaphore can communicate state information. A mail box flag is an example of a semaphore. The flag can be raised to indicate a latter is waiting in the mailbox. A semaphore is a means of protecting a resource/data shared between threads. It is a token based mechanism for controlling when a thread can have access to the resource/data. Usually a semaphore handle will be able to be received from the system by name/id.

Semaphores are used for two purposes

1) Process Synchronization

2) Critical Section problem / Mutual Exclusion

Example of using semaphores for Synchronization:

Assume two concurrent process P1 and P2 having statements S1 and S2. We want in any case S1 should execute first. this can be achieved easily by initialize Sem=0;

```
In process P1
{
  // Execute whatever you want to do
  // before executing P2
  S1;
signal(Sem);
```

```
    }
    in process P2
    {
    wait(Sem);
       S2;
}
```

e) Differentiate between assembly language program with embedded C w.r.t

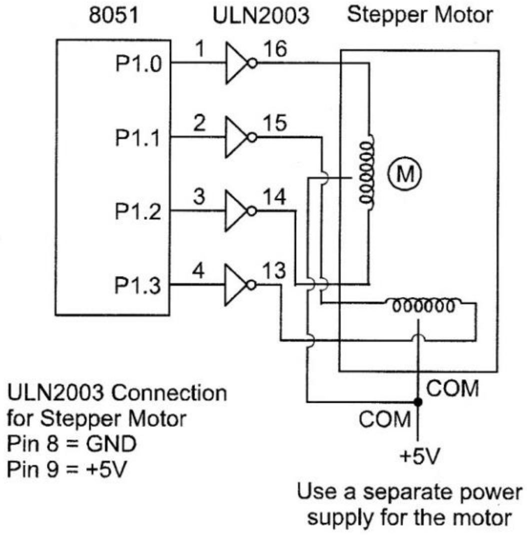
- **Execution time.**
- **Time for coding.**
- **Hex file size.**
- **Debugging.**

Ans:- **(correct differentiate 01M each)**

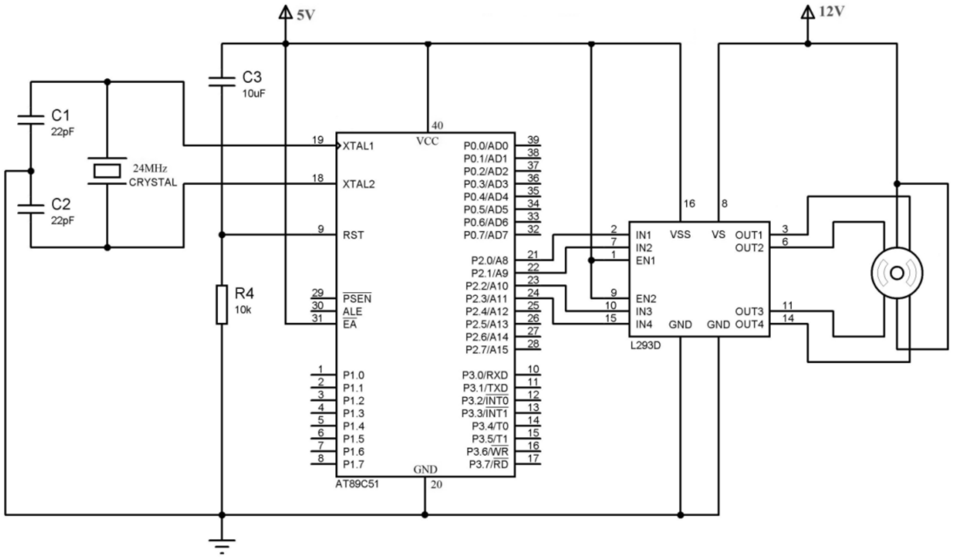
Sr no	**Parameters**	**Assembly Language program**	**Embedded C**
1.	Execution Time	Faster [Less Execution time required]	Slower [More execution time required]
2.	Time for coding	More time is required for coding.	Less time required for coding and code is more efficient.
3.	Hex file size	Less	More.
4.	Debugging	Not so easy	Easy

f) Draw the labelledinterfacing diagram for stepper motor with 89C51microcontroller.

Ans:- **(Any one diagram – 04M)**



OR:



Q6) Attempt any four of the following : 16M

a) Define the terms:

- **In Circuit Emulator:**
- **Integrated Development environment.**

Ans: **[02M each]**

In-Circuit Emulator:

- In-circuit emulator: An in-circuit emulator (ICE) is a hardware device used to debug the software of an embedded system. It was historically in the form of bond-out processor which has many internal signals brought out for the purpose of debugging. These signals provided information about the state of the processor.
- An in-circuit emulator provides a window into the embedded system. The programmer uses the emulator to load programs into the embedded system, run them, step through them slowly, and view and change data used by the system's software.
- More recently the term also covers JTAG based hardware debuggers which provide equivalent access using on-chip debugging hardware with standard production chips.
- ICE's attach a terminal or PC to the embedded system. The terminal or PC provides an interactive user interface for the programmer to investigate and control the embedded system.
- In usage, an ICE provides the programmer with execution breakpoints, memory display and monitoring, and input/output control

Integrated Development Environment [IDE]:

An IDE is a software application that provides comprehensive facilities to computer programmers for software development. An IDE consists of:

- A source code editor.
- A compiler and or interpreter.
- Build automation tools.
- A debugger.

IDE is dedicated to a specific programming language, allowing a feature set that most closely matches the programming paradigms [model] of the language. IDE's typically present a single program in which all development is done. This program typically provides many features for authoring, modifying, compiling, deploying, and debugging software. It even support defining a processor family and its version. It provides windows on screen to show status of program, internal registers, peripherals, RAM and ports.

b) Differentiate between CAN and I^{2}C protocols w.r.t:

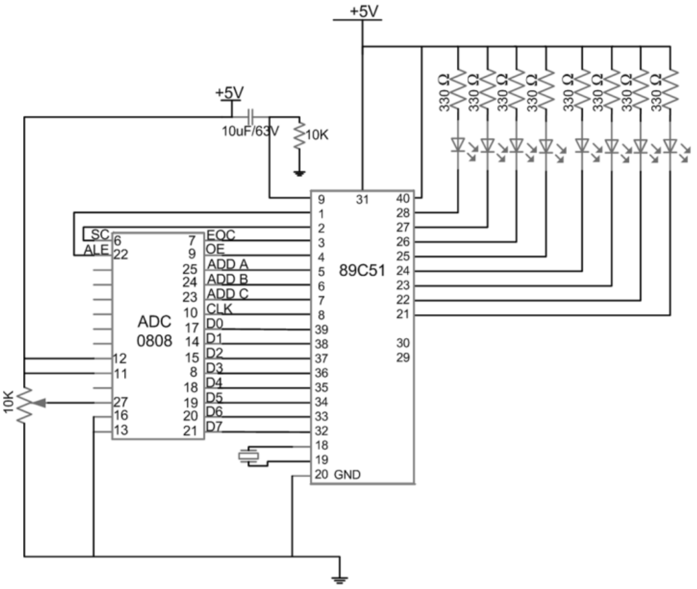
- **Data transfer rate.**
- **No of fields.**
- **Addressing bit**
- **Application**

Ans:- **(Correct differentiation- 01M each point)**

Sr No	Parameters	CAN	I^{2}C
1.	Data transfer rate	Asynchronous with 250 Kbps up-to 1Mbps	Synchronous with 3speeds 100Kbps, 400 Kbps and 3.4Mbps
2.	Number of fields	08 [including 7 bits of frame end and 3 bits of inter frame gap].	07
3.	Addressing bit	11 bit	7-bit 0r 10 bit address
4.	Applications	Copiers, Telescopes, Medical instruments, Elevator controllers, Automobile industry	To interface devices like watch dog, Flash and RAM memory, Real time clock , Microcontrollers

c) Draw the labeled interfacing diagram of ADC 0808 with 89C51microcontroller show the handshake signals clearly.

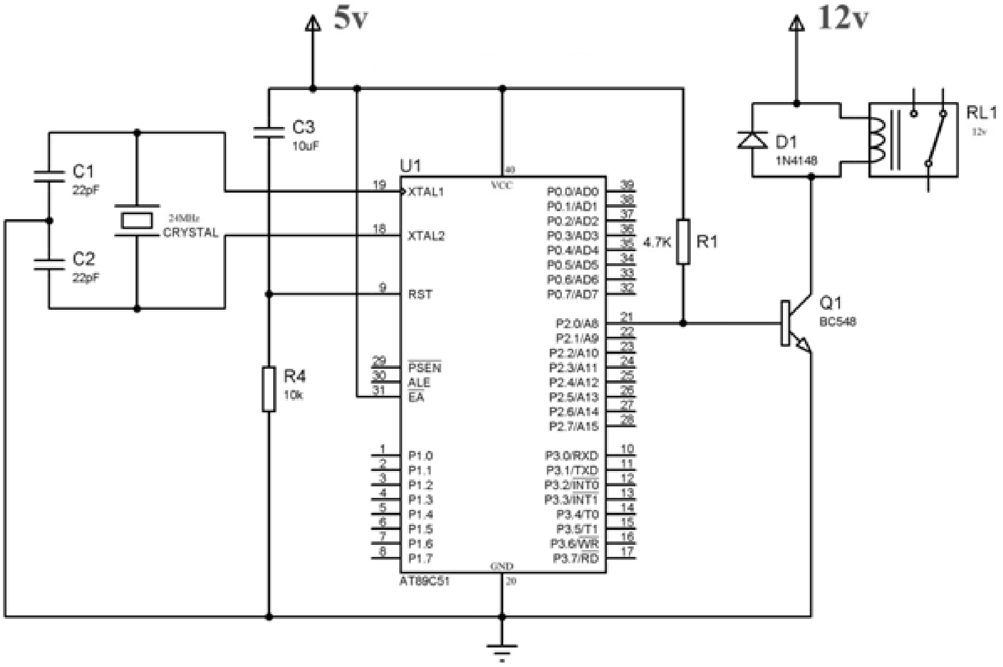
Ans:- **(any one relevant diagram with handshake signals – 04M)**



d) Draw the interfacing diagram of relay with 89C51.

Ans:-

(Correct Diagram – 04M)



e) Write Logical operators in C for AND, OR, Ex-OR and NOT for 89C51 and state one example of each.

Ans:- **(01M each with correct example)**

Sr no:	**Operator**	**Description**	**Example**
1.	NOT	~ is called 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.	Y= ~A
2.	AND	& is called Logical AND operator. If both operands are non-zero then the condition becomes true.	Y= A&B
3.	OR	| is called Logical OR operator. If any one of the two operands are non-zero, then the condition becomes true.	Y= A|B
4.	EX-OR	^ is called logical Ex-OR operator. Output is true for odd number of non-zero operands.	Y=A ^B