WINTER– 18 EXAMINATION

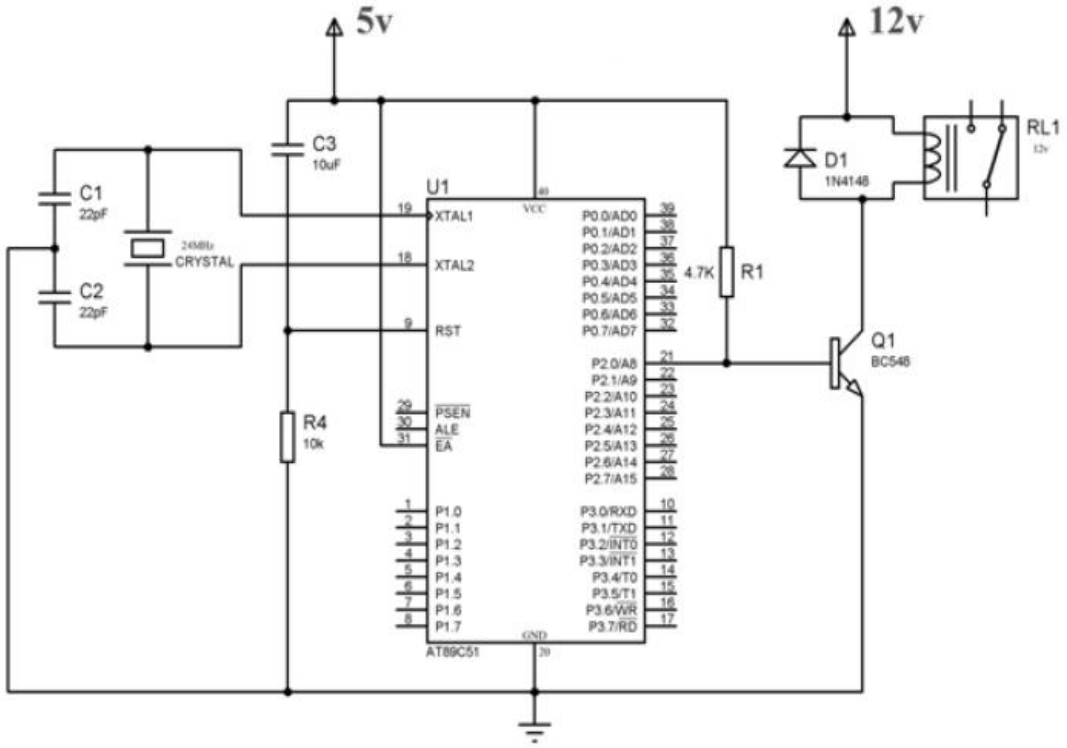
Subject Name: Embedded Systems **Model Answer** **Subject Code:** 17658

Important Instructions to examiners:

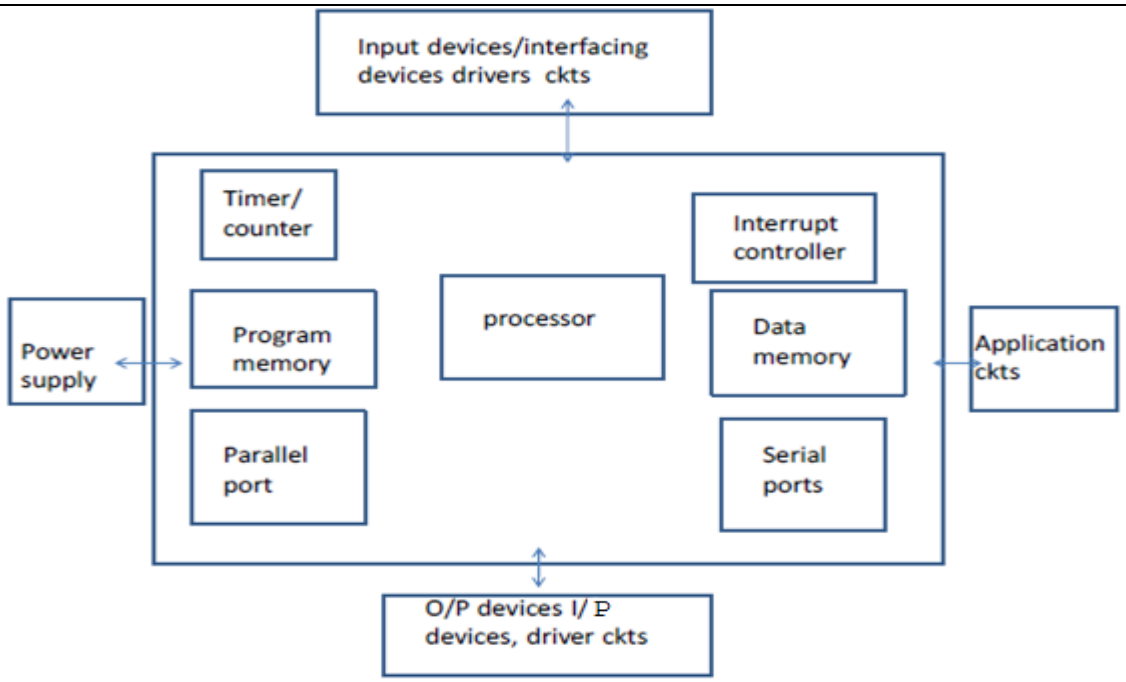
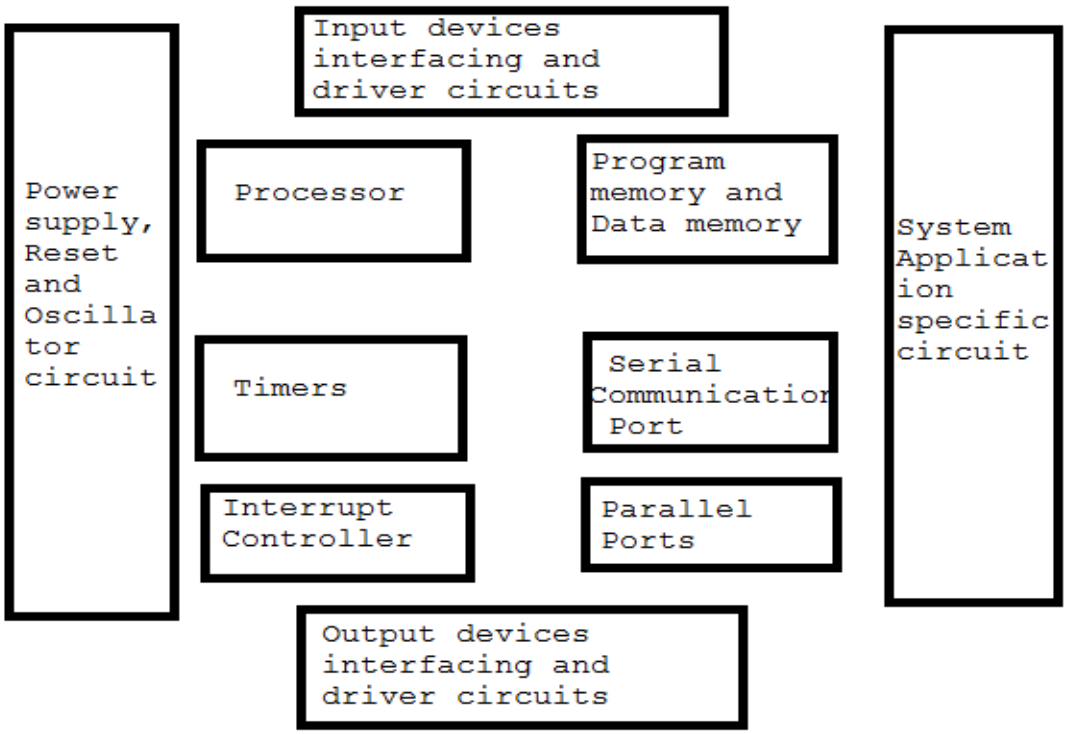
1) The answers should be examined by key words and not as word-to-word as given in themodel answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given moreImportance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in thefigure. The figures drawn by candidate and model answer may vary. The examiner may give credit for anyequivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constantvalues 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 judgement 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.

Q. No.	Sub Q. N.	Answers	Marking Scheme
1	**a)**	**Attempt any THREE of the following:**	**12- Total Marks**
	(i)	**What is multicore processor? Describe in brief.**	**4M**
	Ans:	**Multicore Processor:** • It is an integrated circuit in which two or more processors core have been packaged for enhanced performance, reduced power consumption and more efficient simultaneous multitasking. • In multi core technology architecture, a single physical processor contains the core logic of two or more processors and these processors are packaged into a single integrated circuit. • The multicore technology is mainly used for parallel computing which increases computer speed and efficiency. • Multiple processes are used in mobile devices, desktops, workstations and servers. • Multicore processors will give the benefits to all software especially for multithreaded program.	**Description -4M**

	(ii)	**State any four features of IDE.**	**4M**
	Ans:	**Features of IDE-** 1) IDE normally consists of source code editor, a compiler, build automation tools and debugger. 2) It supports for defining a processor family and its version. 3) Supports a user definable assembler to support a new version or type of processor. 4) Provides multiuser environment. 5) Supports conditional and unconditional break points. 6) Provides debugger.	**Any four features -4M** **(1M each)**
	(iii)	**State four features of Zigbee.**	**4M**
	Ans:	**Zigbee features (any4):-** • IEEE Standard 802.15.4 • Frequency (GHz) 0.868, 0.915, 2.4 • Maximum bit rate (Mbps) 0.250 or 250 Kbps, low bit rate • Typical data throughput (Mbps) 0.2 • Maximum (Outdoor) Range (Meters) 10-100 • Relative Power Consumption Very low • Battery Life Months to years • Network Size 64,000+	**Any four features -4M,** **(Each feature-1M)**
	(iv)	**Draw the labeled interfacing diagram to interface relay with 89C51 microcontroller.**	**4M**
	Ans:		**Correct labeled diagram -4M**



	b)	**Attempt any ONE of the following:**	**06- Total Marks**
	(i)	**Draw the block diagram of an embedded system and describe the hardware units of an embedded system.**	**6M**

	Ans:	 **OR** 	**Block diagram -2M** **Description of hardware units- 4M**

Explanation:
The embedded system consist of different components embedded into it as follows:

1) Embedded Processor
2) Power supply , reset and Oscillator circuits
3) System timers
4) Serial communication port
5) Parallel ports
6) Interrupt controller
7) Output and Input Interfacing and driver circuits
8) System application specific circuits
9) Program and Data 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)

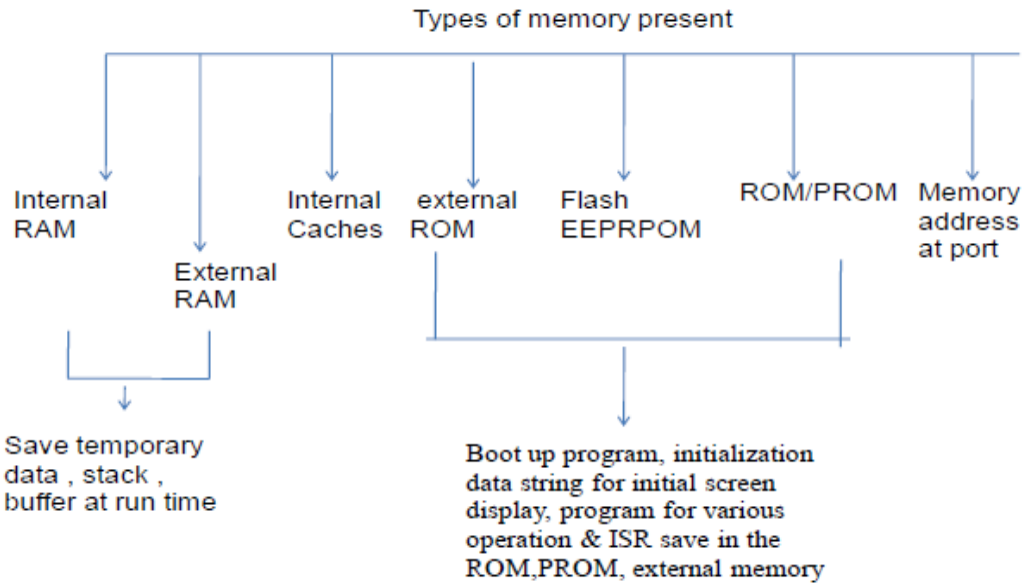
Power supply, Reset, Oscillator circuit and system timers:
Power source:
Internal power supply is must. Es require from power up to power down to start time task. Also it can run continuously that is stay "On' system consumes total power hence efficient real time programming by using proper 'wait' and '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.
Resets 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 reset and runs the program in the following way
1. System program that execute from beginning
2. System boot up program
3. System initialization program

Serial and Parallel communication ports:
Serial communication port and parallel communication ports are used to interface serial and parallel devices with the system and communicate between processor and devices.

Interrupt controller:
It is used to receive interrupt from various sources and resolve the priority and provides the service to that interrupts.

Input and output interfacing and driver circuits:
Characteristics of input or output devices may be different from the processor like voltage and current requirement to drive that specific device, hence driver circuits are needed to drive input or output devices.

Program and Data memory:
The most microcontroller have inbuilt separate memory to store data and program. Following are types of memories used in embedded system.

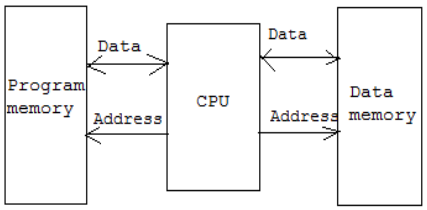
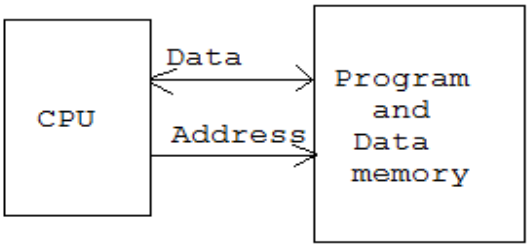


	(ii)	**State the methods of task synchronization. Describe semaphore with suitable example.**	**6M**
	Ans:	**The methods of task synchronization are:** • Semaphore • Message queue. • Mutual exclusion.	**State or List of methods -2M**

		• Dead lock.
		• Mailboxes.
		• Message Queues.
		
		Semaphores:
		• It is a system of sending message by using flags. Multiple concurrent threads of execution with an application must be able to synchronize their execution & co-ordinate mutually exclusive access to shared resources.
		• To fulfill this requirement RTOS kernel provides a semaphore object that one or more threads of execution can acquire or release for the purpose of synchronization or mutual exclusion. Semaphore is like a key that allows a test to carry out some operation or to access a resource. When task acquires the semaphore, it receives an access to the resource.
		• The number of times the semaphore can be acquired by the task is determined by the resource count of a semaphore.
		• A resource count is decremented by one, when a task acquires the semaphore and its resource count is incremented by one when a task releases the semaphore.
		• A kernel supports many different types of semaphores :
		•
		Binary: Binary semaphores are used for both mutual exclusion and synchronization purposes. A binary semaphore is used to control sharing a single resource between tasks..
		
		Counting: it is a semaphore that increments when an IPC is given by a task. It decrements when a waiting task unblocks and starts running.
		
		Mutex or Mutually Exclusion semaphore: In this a mutex key is used to lock the shared resource, if it is acquired by the task, so that any other task cannot acquire until it is released.
		
		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;
			Description-3M
			
			Example-1M

		}	
Q. No.	Sub Q. N.	Answers	Marking Scheme
2		**Attempt any FOUR of the following:**	**16- Total Marks**
	a)	**Compare Harvard and Von Neuman architecture with a suitable diagram.**	**4M**

Ans:

Sr. No.	Harvard architecture	Von-Neumann architecture
1		
2	Harvard architecture uses physically separate memories for their instructions and data.	Von –Neumann architecture uses single memory for their instructions and data.
3	Requires separate and dedicated buses for instructions and data.	It requires single bus for instruction and data
4	Design is complicated.	Its design is simpler.
5	Instructions and data can be fetched simultaneously which increases the operation speed.	Instructions and data have to be fetched in sequential order which limits the operation speed.

Any four points- 4M which includes 1M for diagram

	b)	**State any four 'C' data types with their value range.**	**4M**

Ans:

Data Type	Size in Bits	Data Range/Usage
Unsigned char	8-bit	0 to 255
Signed char	8-bit	-128 to + 127
Unsigned int	16-bit	0-65535
signed int	16-bit	-32768 to + 32767
sbit	1-bit	SFR bit-addressable only
bit	1-bit	RAM bit-addressable only
sfr	8-bit	RAM addresses 80 –FFH only

Any four data types- 4M, (Each- 1M)

c) What are the features of IEEE 802.11 wireless LAN protocol? **4M**

Ans:

IEEE 802.11

- Wi-Fi follows the IEEE 802.11 standard.
- Supports asynchronous communication.
- Provides 1 or 2 Mbps transmission in the 2.4 GHz band.
- Transmission rates of 1 and 2 Mbps.
- Provides time bound delivery service.
- Provides Multicast as well as broadcast services.
- Continuity of service within extended areas via a distribution system, such as Ethernet.
- Supports Network management services.
- Provides registration and authentication service

Any 4 features - 4M

d) Write a program in 'C' language for generating triangular waveform using DAC 0808. **4M**

Ans: ***NOTE: Program may change. Student can also use the other logic. Please check the logic and understanding of students. any amount of delay program can also be used.***

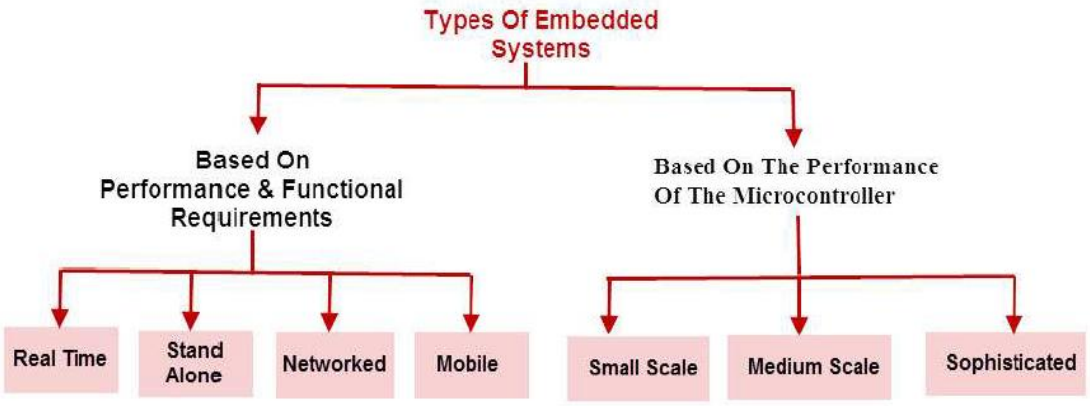
Program:

```
#include<reg51.h>
void main(void)
{
```

Correct program -4M

		unsigned char d;
		while(1)
		{
		for(d=0; d<=255; d++)
		{
		P1 = d;
		}
		for(d=255; d>=0; d--)
		{
		P1 = d;
		}
		}
		}	

	e)	**State any four key specifications of RTOS.**	**4M**
	Ans:	**Key Specifications of RTOS:** 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 fulfill 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 cellphone, ECG machine, etc. 5. **Scalability:** RTOS can be used in a wide variety of embedded. They must be able to scale-up or scale-down to suit the application.	**Any four specifications-4M,** **(Each specification-1M)**
	f)	**Give classification of embedded systems. Explain any one.**	**4M**

	Ans:	 **Explanation :(any two of the following) 2M** **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 where 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 where 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	**Classification- 2M and** **Any one type detail explanation-2M**

		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.	

Q. No.	Sub Q. N.	Answers	Marking Scheme
3		**Attempt any FOUR of the following:**	**16- Total Marks**
	a)	**Write 89C 51 program in 'C' language to toggle bits of port P1 continuously.**	**4M**

	Ans:	***NOTE: Program may change. Student can also use the other logic. Please check the logic and understanding of students. any other amount of delay program can also be used.***
		Program:
		#include < reg 51.h>
		void delay (unsigned int);
		void main (void)
		{
		while(1) //repeat loop
		{
		P1=0xff; //toggle all bits of port1
		delay (400); //add delay
		P1=0x00; //toggle all bits of port1
		delay (400); //add delay
		}
		}
		void delay (unsigned int)
		{
		Unsigned int x, y;
			(**Correct program 4M)**

```
 for(x=0; x<i ; x++)
for(y =0 ; y <1275 ; y++);
}
```

b) Compare CAN and I2C protocols with respect to: **4M**

(i) Data transfer rate
(ii) Number of fields
(iii) Addressing bit and
(iv) Application

Ans: **(1M each point)**

Sr No	Parameters	CAN	I2C
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) What is interrupt and event handling in RTOS? Explain in brief. **4M**

Ans: **4M for correct explanation**

An interrupt is a hardware mechanism used to inform the CPU that an asynchronous event has occurred.

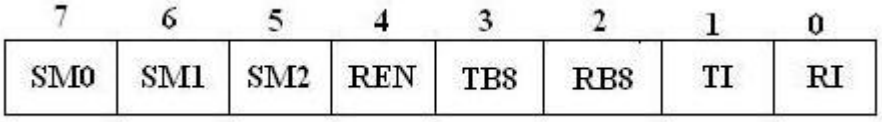
A fundamental challenge in RTOS design is supporting interrupts and thereby allowing asynchronous access to internal RTOS data structures. The interrupt and event handling mechanism of an RTOS provides the following functions:

- Defining interrupt handler
- Creation and deletion of ISR
- Referencing the state of an ISR
- Enabling and disabling of an interrupt
- Changing and referencing of an interrupt mask and help to ensure:
    - Data integrity by restricting interrupts from occurring when modifying a data structure
    - Minimum interrupt latencies due to disabling of interrupts when RTOS is performing critical operations
    - Fastest possible interrupt responses that marked the preemptive performance of an RTOS
    - Shortest possible interrupt completion time with minimum overhead

	d)	**Give advantages and disadvantages of embedded systems.**	**4M**
	Ans:	**Advantages of an embedded systems (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. **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	**(2 M :advantages ,** **2M: disadvantages)**
	e)	**Write 89C51 'C' program to rotate stepper motor by 90^0 in clockwise direction. Motor has step angle of 1.8^0.**	**4M**

	Ans:	***NOTE: Program may change. Student can also use the other logic. Please check the logic and understanding of students.***
		Step angle=1.8 degree
		For Four step sequence , 1.8 X 4 = 7.2 degrees
		Count for four step sequence = 90/7.2=12.5
		Take count as 12 , as 12 X 7.2 = 86.4 degrees , give two more steps for further 3.6 degrees
		
		#include <reg 51.h>
		void delay( );
		void main(void)
		{
		int x;
		for (x=0; x<12; x++)
		{
		P0=0x99;
		delay(10);
		P0=0xcc;
		delay(10);
		P0=0x66;
		delay(10);
		P0=0x33;
		delay(10);
		}
		P0=0x99;
		delay(10);
		P0=0xcc;
		delay(10);
		}
		void delay(unsigned int i)
		{
		unsigned int x,y;
		for(x=0;x<=i;x++)
		for(y=0;y<=1275;y++);
		}
			(correct logic 4 marks)

Q. No.	Sub Q. N.	Answers	Marking Scheme
4	**a)**	**Attempt any THREE of the following:**	**12- Total Marks**

	(i)	**Draw the format of SCON. Describe the function of each bit.**	**4M**
	Ans:	 SM0 SCON.7 : Serial port mode specifier SM1 SCON.6 : Serial port mode specifier. SM2 SCON.5 : Used for multiprocessor communication (Makeit0.) REN SCON.4 : Set/cleared by software to enable/ disable reception. TB8 SCON.3 : 9th bit to be transmitted in mode 2 and 3. RB8 SCON.2 : 9th received bit in mode 2 and 3. TI SCON.1 : Transmit interrupt flag. Set by hardware at the beginning of the stop Bit in mode1. Must be cleared by software. RI SCON.0 : Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software	**(2M : format , 2M:Bit explanation)**
	(ii)	**Compare serial and parallel communication(any four points)**	**4M**
	Ans:		**(1M : each point)**

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 less
6	Less reliable	More reliable

	(iii)	**Describe any four design metrics of an embedded system.**	**4M**
	Ans:	1. **Processor power:** Selection of processor depends upon amount of processing power and the register width required. Powerful 8bit, 16 bit , 32 bit & 64bit processors are available . The clock speed and memory addressing capability is also measure of processor power. Powerful DSPs are available for real time analysis of audio and	**(1M**

		video signals 2. **Memory:** Designer has to make an estimate for memory requirement and must make provision for expansion. There are different types of memories in a system ,like RAM, ROM , EEPROM etc. Flash memories are used in embedded system , hence operating systems can be ported in target hardware system. 3. **Reliability:** It is the ability that a system will not fail. The ability of system to work properly in all situations without human intervention is called reliability. 4. **Safety:** It is the probability that system will not cause harm. 5. **NRE cost (Non-Recurring Engineering cost):** It is the one-time monetary cost of designing the system. Once system is designed any number of units can be manufactured without incurring any additional design cost. 6. **Unit cost:** The monetary cost of manufacturing each copy of the system, excluding NRE cost 7. **Size:** It is measured in terms of the physical space required by the system , memory required in bytes for software and number of gates, transistors in hardware 8. **Performance:** instruction execution time in the system measures performance. Smaller execution time means higher performance 9. **Power :** The amount of power consumed by the system which may determine lifetime of battery , or cooling requirement s of the IC , since more power means more heat 10. **Flexibility:** The ability to change the functionality of the system without incurring heavy NRE cost. Most of the system flexibility is provided by system software 11. **Time-to-prototype:** The time needed to build a working version of the system, which may be bigger or more expensive than final system implementation , but can be used to verify system's usefulness and correctness and to refine systems functionality 12. **Time-to-market:** The time required to develop a system to the point that it can be released and sold to customers. The revenue generated in embedded system depends upon this time. 13. **Maintainability:** The ability to modify the system after its initial release, especially by designers who did not originally design the system 14. **Correctness:** The implemented system should work functionally correct. To achieve this functionality of the system is checked throughout the designing process . 15. **Operating system:** In desktop the selection of O.S. is limited. In embedded system , a variety of operating systems are Available which can be ported into embedded system. Operating systems like Embedded OS, real Time OS (RTOS) ,and mobile OS , occupy less area in memory than desktop OS. For real time applications RTOS should be used. Also many open source operating systems are available.	**each)**
	(iv)	**Compare RTOS with desktop operating system(any four points)**	**4M**

Ans:

Sr.no.	Desktop O.S.	RTOS
1.	Does not have deterministic time response	It has deterministic time response
2.	Generalized kernel	Real time kernel
3.	There is no task deadline	There is task deadline in RTOS
4.	Memory required depends on version	Memory required(footprint) is less
5.	Applications are compiled separately from O.S.	Applications are compiled and linked together with RTOS
6.	It is used in general desktop computer	It is used in embedded systems
7.	It is less reliable	It is more reliable
8.	e.g. Windows, Linux	e.g. RTLinux, VXWorks, Micro C / OSII

(1M each point)

b) Attempt any ONE of the following: **06- Total Marks**

(i) Write 'C' program to generate frequency of 2.5 KHz on P2.7 . Use timer 1 in mode 2 to generate the delay.(f_{osc} = 12 mHz) **6M**

Note: fosc should be in MHz, it seems to be typing mistake in question as mHz. So if students have done the calculation with mHz, marks must be given for correct logic.

Calculations:

Crystal value = 12 MHz.

Frequency = 2.5khz

Therefore Time period T = 1 /2.5KHZ = 0.4 ms

Therefore Required time delay = 0.4ms / 2 = 0.2 ms =200usec.

Required time delay = (12 / Fosc) x number of increments (N)

200us = (12 / 12MHZ) x number of increments (N)

200us = 1 usec. x N

N = 200

Using TIMER 1 in MODE 2,

COUNT = 2^8 - N

COUNT = 256 – 200 = 56= 38H

Therefore TH1 = 0X 38

Calculations --- 2Marks, Program --- 4 Marks

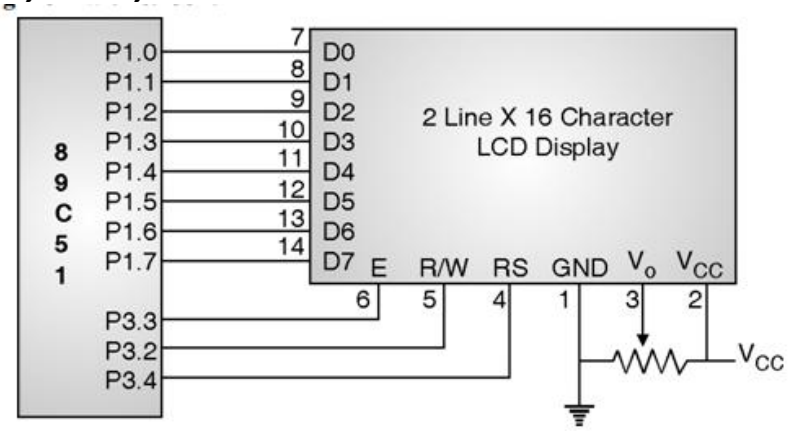
```
#include <reg51. h>
sbit SQR=P2^7;
```

```
void main(void)
{
TMOD=0x20;//timer1 mode2
TH1=0X38  ; // Load count
TR1=1; // start timer1
while (1)       // repeat forever
{
while (TF1= = 0) ; // wait for Timer to overflow
SQR = ~ SQR ;  // toggle bit P2^7
TF1=0; // clear  TF1
} // end of while
} // end of main
```

(ii) **Draw the labeled diagram of LCD interface with 89C 51. Write a program in 'C' language to display "WELCOME" on LCD.** **6M**

Ans:

Note: Student may use any other method



(interfacing diagram : 02M, program : 04M)

PROGRAM:

```
#include <reg 51.h>
 Sbit rs=P3^4;
Sbit rw=P3^2;
sbit en=P3^3;
 void delay(unsigned int);
 void lcdcmd(unsigned char);
void lcddta(unsigned char);
 void main( void )
 {
 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('W');
delay(10);
lcddta('E');
 delay(10);
lcddta('L');
 delay(10);
lcddta('C');
 delay(10);
lcddta('O');
 delay(10);
lcddta('M');
 delay(10);
lcddta('E’);
 delay(10);
while(1) ;
 }
 }
 void lcddta(unsigned char dta)
 {
rs=1;
 rw=0;
P1=dta;
 en=1;
delay(2);
 en=0;
 }
void lcdcmd(unsigned char cmd)
{
rs=0;
 rw=0;
 P1=cmd;
en=1;
delay(2);
```

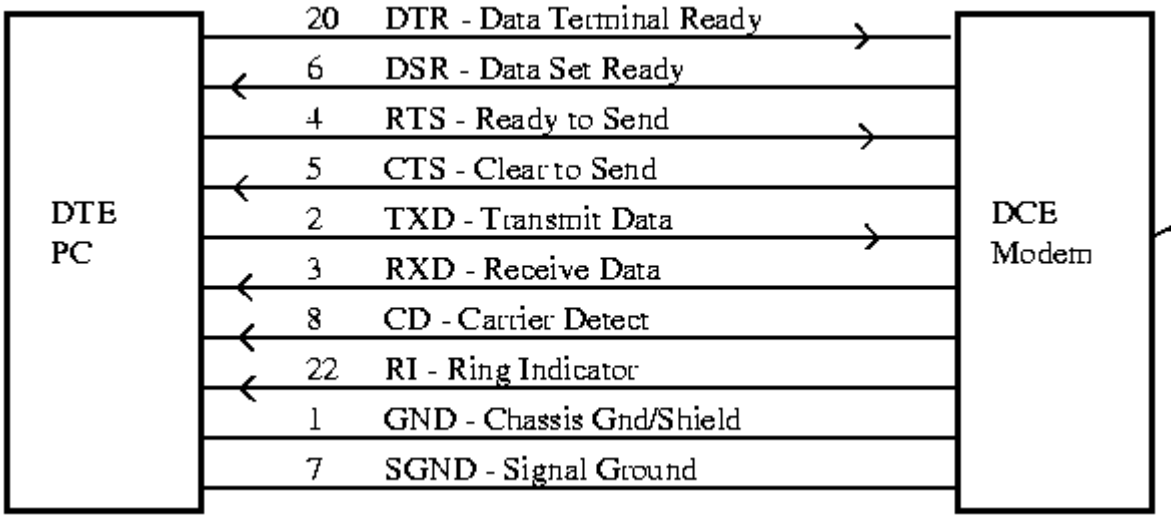
```
 en=0;
 }
void delay(unsigned int t)
 {
 Unsigned in tx,y;
for(x=0;x<=t;x++)
 for(y=0;y<=1275;y++);
 }
```

Q. No.	Sub Q. N.	Answers	Marking Scheme
5.		**Attempt any FOUR of the following:**	**16- Total Marks**
	a)	**Write 89C51 'C' program to transfer character 'MSBTE' serially at 9600 baud rate continuously , use 8 bit data and 1 stop bit. Assume crystal frequency of 11.0592 MHz.**	**4M**
	Ans:	***NOTE: Program may change. Student can also use the other logic. Please check the logic and understanding of students.***	**4M for correct program**

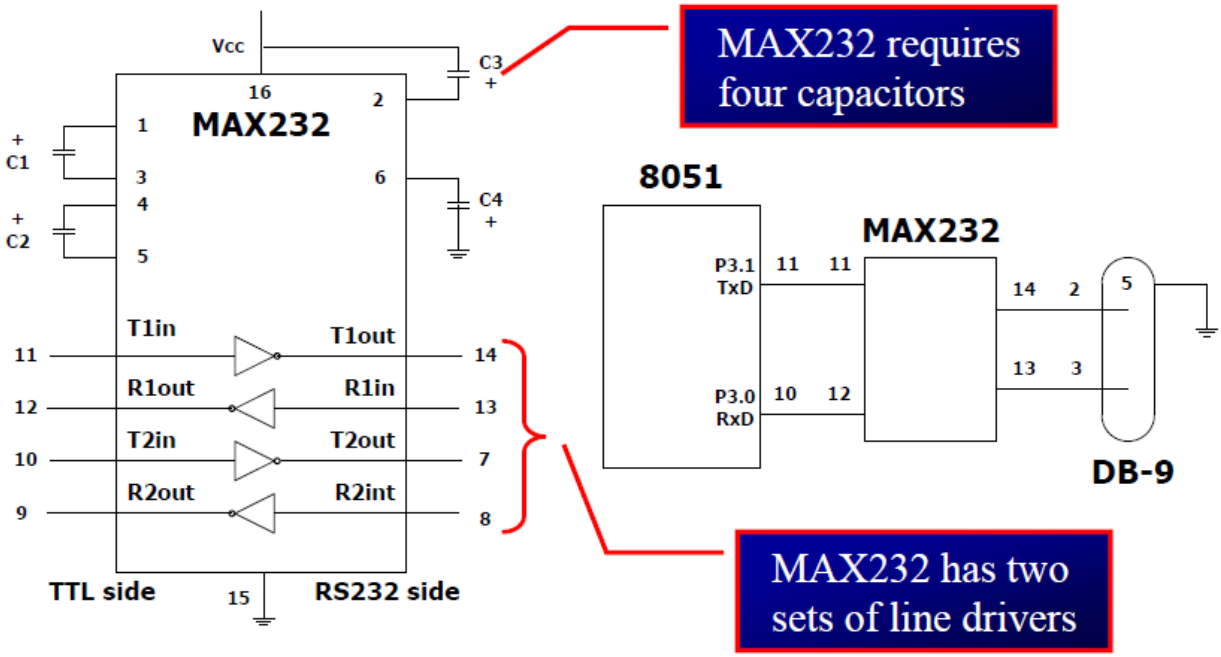
```
#include <reg51.h>
void main(void)
{
unsigned char text[ ] = "MSBTE";          //initialize array
TMOD = 0x20;                              //Initialize timer 1 in mode 2
TH1 = 0xFD;                               //baud rate 9600
SCON = 0x50;                              //start serial communication ( 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
}
}
```

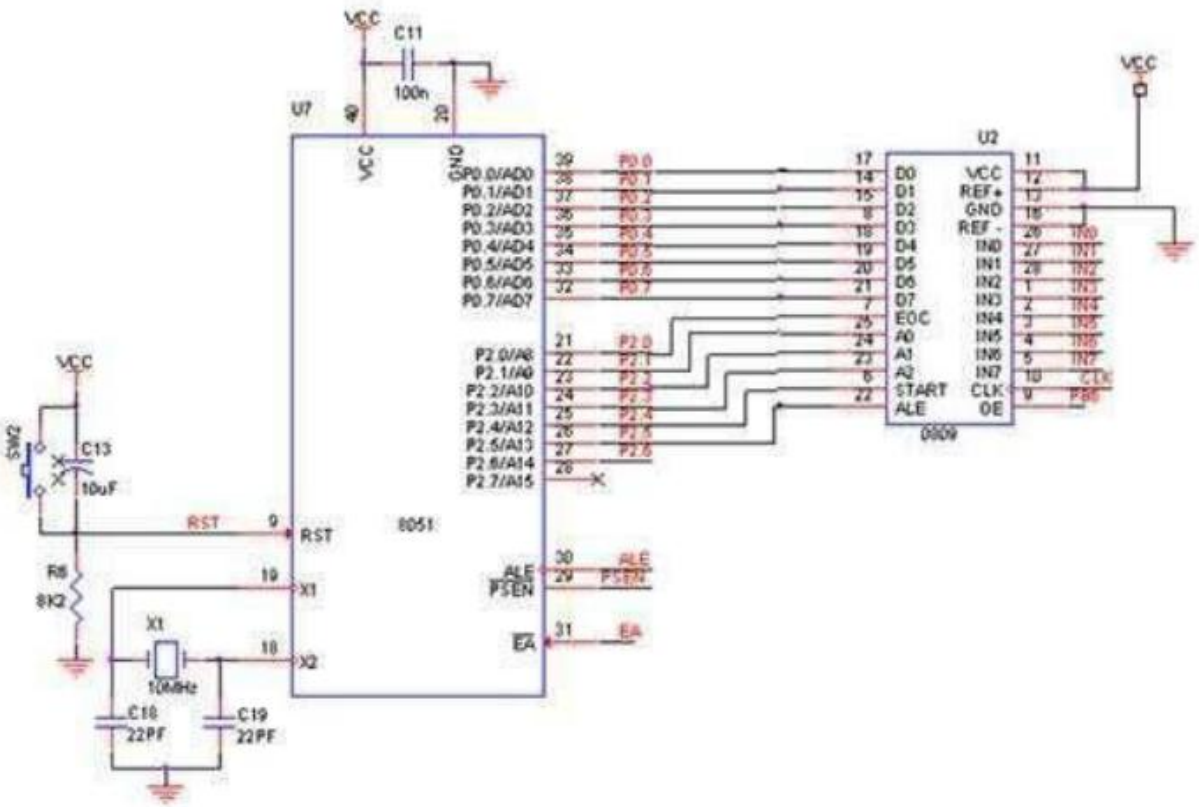
	b)	**Describe the need of RS-232 and MAX-232 with a suitable diagram.**	**4M**

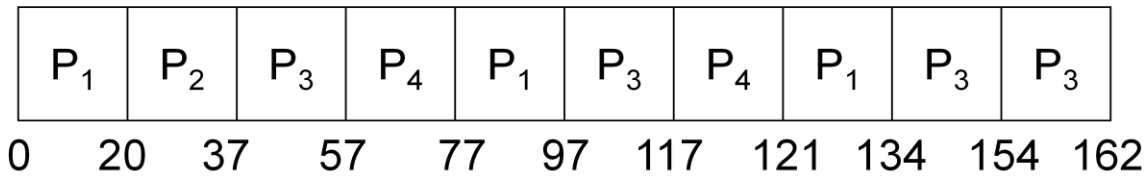
	Ans:	**Need of RS232:** ⦿ RS-232 is a standard by which two serial devices communicate (recommended standard) ⦿ The RS-232 interface is the Electronic Industries Association (EIA) standard for the interchange of serial binary data between two devices. ⦿ It was initially developed by the EIA to standardize the connection of computers with telephone line modems ⦿ The connection must not be longer than 50 feet. (15 m) ⦿ The standard defined voltage levels made it immune to noise disturbances and reduced the error in data exchange. ⦿ The maximum slew rate in RS232 is limited to 30V/µs. Also, a maximum bit rate of 20 Kbps is also defined. These limitations of the standard help in reducing the cross – talk with adjacent signals.  **Need of MAX 232:** • A line driver such as the MAX232 chip is required to convert RS232 voltage levels to TTL levels, and vice versa.	**2M -RS-232** **2M - MAX 232** **(1M for any two needs & 1M for diagram each)**

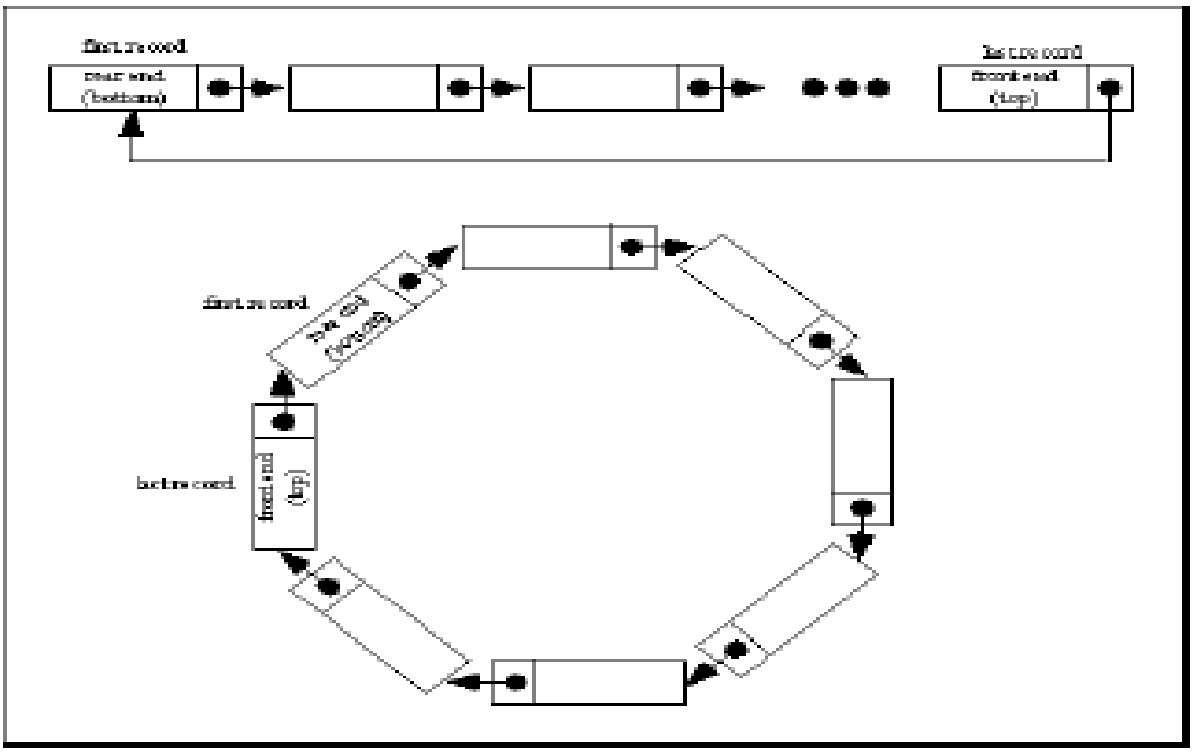
- 8051 has two pins that are used specifically for transferring and receiving data serially.
- These two pins are called TxD and RxD and are part of the port 3 group (P3.0 and P3.1).
- These pins are TTL compatible; therefore, they require a line driver to make them RS232 compatible.
- We need a line driver (voltage converter) to convert the R232's signals to TTL voltage levels that will be acceptable to 8051's TxD and RxD pins.



	c)	**Draw labeled interfacing diagram of ADC 0808 with 89C51 microcontroller.**	**4M**

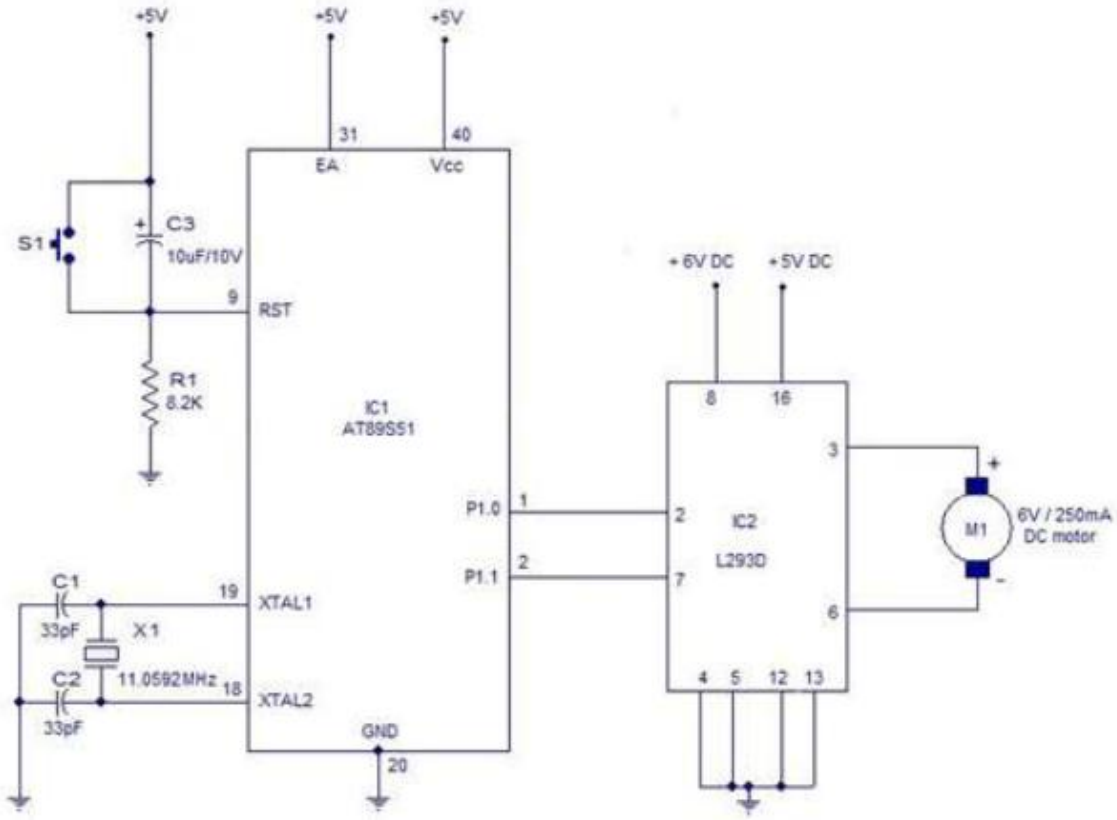
	Ans:		**Correct drawing -4M**
	d)	**Describe round robin scheduling algorithm with suitable example.**	**4M**
	Ans:	• Each process gets a small unit of CPU time (*time quantum*), usually 10-100 milliseconds. After this time has elapsed, the process is preempted and added to the end of the ready queue. • If there are *n* processes in the ready queue (as a FIFO) and the time quantum is *q*, then each process gets 1/*n* of the CPU time in chunks of at most *q* time units at once. No process waits more than (*n*-1)*q* time units. • Performance – *q* large • ⇒ FIFO • Some process will release its execution voluntarily – *q* small • ⇒ *q* must be large with respect to context switch, otherwise overhead	**2M explain** **2M example**

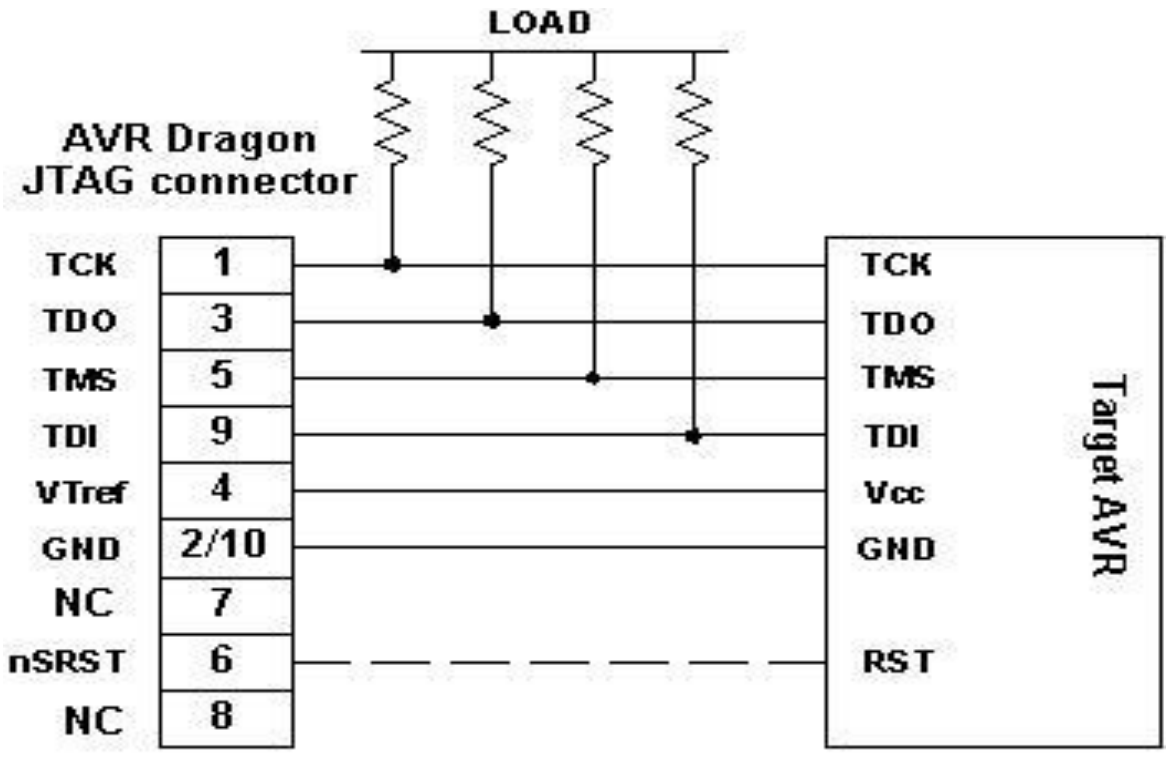
		is too high Example: Process Burst time P1 53 P2 17 P3 68 P4 24  • The Gantt chart is: • Typically, higher average turnaround than SJF, but better response	

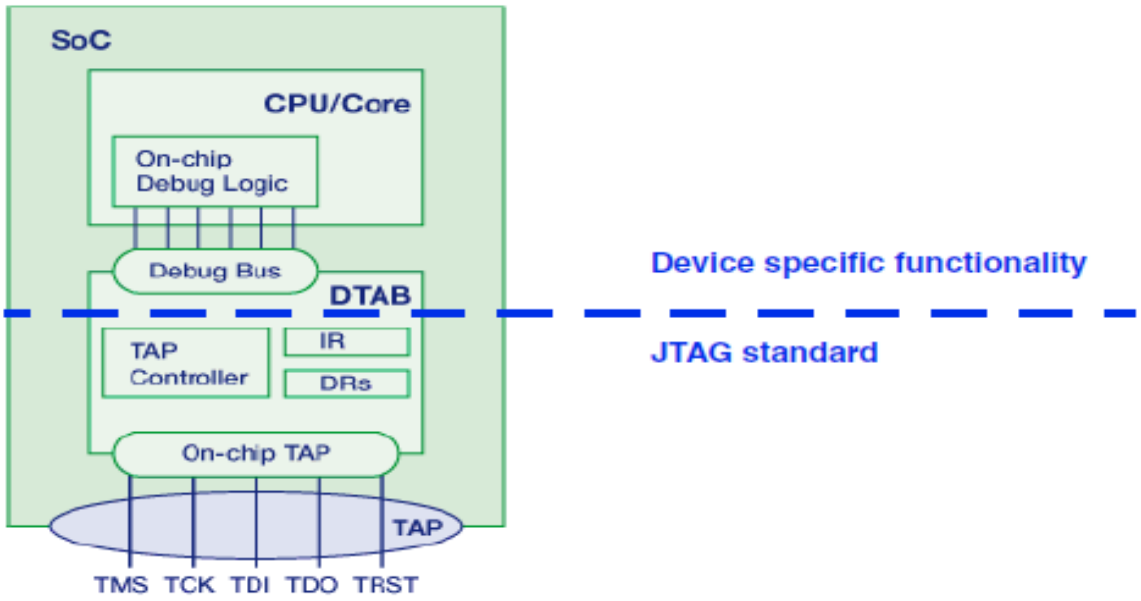


	e)	**List the software development tools used in an embedded system and state the functions of any two.**	**4M**
	Ans:	**Software development tools:** • Compiler • Cross assembler • Cross compiler • Locators • Loaders • Simulators • Debugger	**List 2M** **1M each for any 2**

		• Integrated development environment (IDE). The function of 1) **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. 2) **Cross assembler:** It is useful to convert object codes for microcontrollers or processor to other codes for another microcontrollers or processor and vice versa. 3) **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. 4) **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. 5) **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. 6) **Debugger: -** Allows you to download your code to the emulator's memory and the control all of the functions of the emulator from a PC. Common debugging features include the capability to examine and modify the microcontroller's on chip registers, data- and program-memory; pausing or stopping program executing at defined program locations by setting breakpoints; single-stepping (execute one instruction at a time) through the code and looking at a history of executed code (trace). 7) **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.	

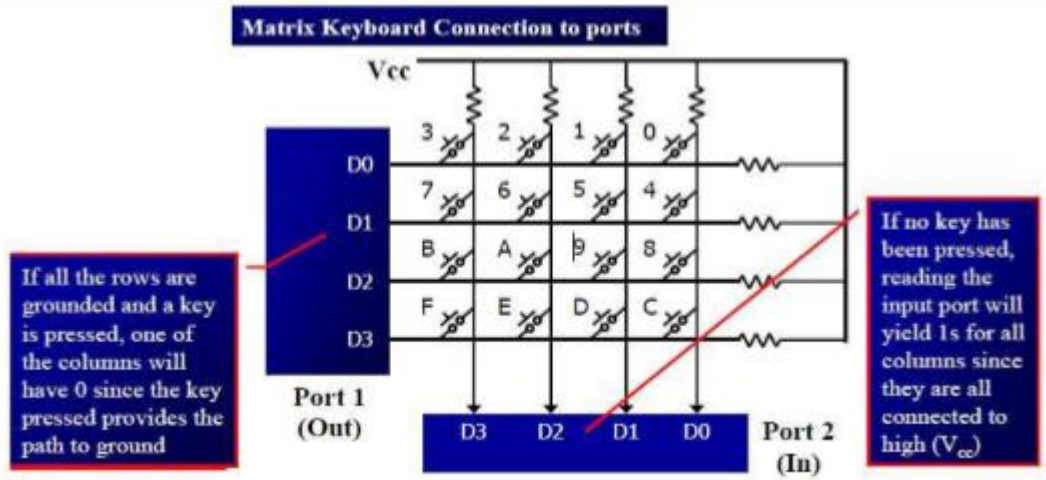
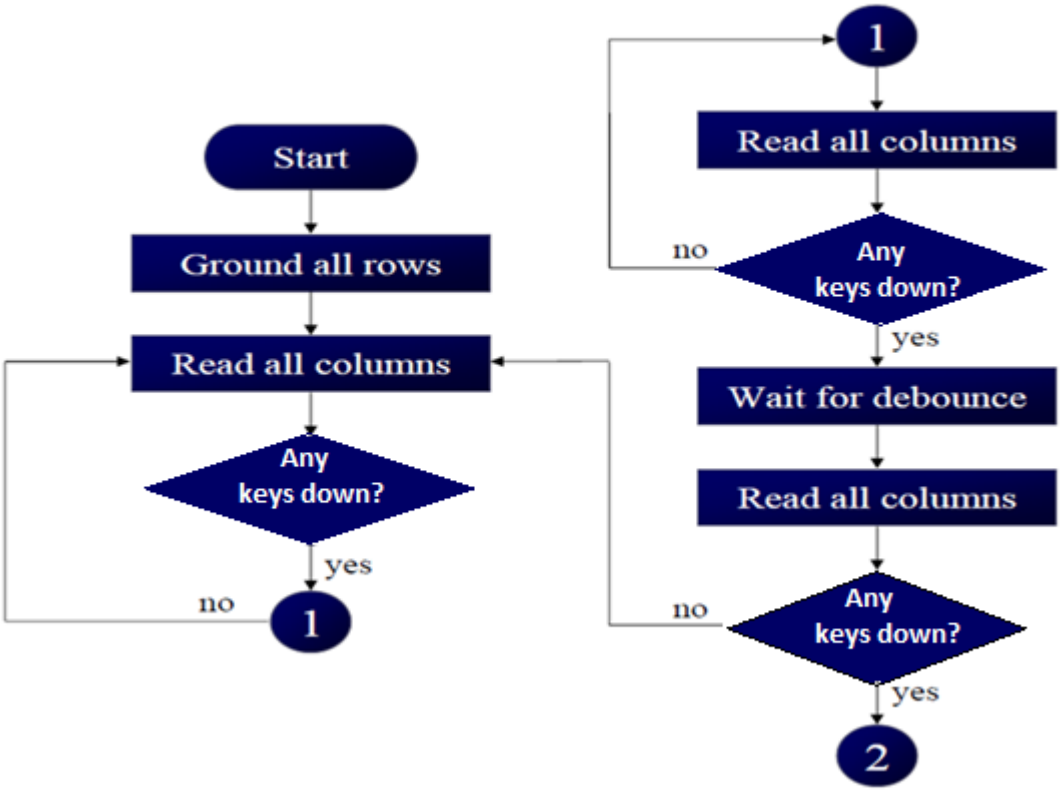
	f)	**Draw and explain the interfacing of DC motor with 89C51 microcontroller.**	**4M**
	Ans:	 **Explanation:** • The maximum current which can be sunk from 89C51 is upto 15mA at 5V. • But DC motor needs more current depending on its type. Hence DC motor cannot be connected directly to the microcontroller. • L293D motor driver stops the back emf produced by the motor from affecting the microcontroller. • Port pins P1.0 and P1.1 are connected to the input pins 2 and 7 of L293D motor driver. • The output pins of the L293D ,pin 3 and 6 are connected to the DC motor. • Direction of the rotation of the motor can be changed by inverting the output terminals	**2M Interfacing** **2M explanation**

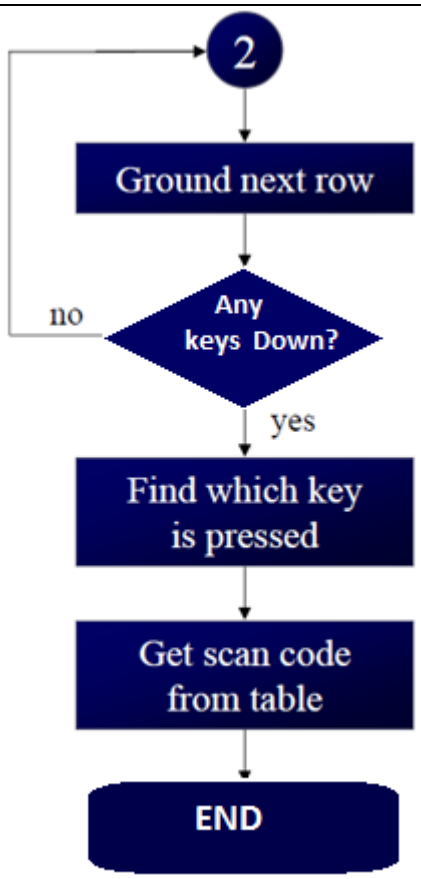
Q. No.	Sub Q. N.	Answers	Marking Scheme
6.		**Attempt any FOUR of the following:**	**16- Total Marks**
	a)	**What is function of JTAG port? Explain in brief.**	**4M**
	Ans:	• The functionality usually offered by JTAG is Debug Access and Boundary Scan: • Debug Access is used by debugger tools to access the internals of a chip making its resources and functionality available and modifiable, e.g. registers, memories and the system state. • Boundary Scan is used by hardware test tools to test the physical connection of a device, e.g. on a PC. 	**4 M Explanation**



- The DTAB (Debug and Test Access Block) is implemented on the target chip as a "passive" device that never sends data without request.
- The DTAB mainly consists of the following:
    - The TAP (Test Access Port) with its physical connections (signals) to the external world.
    - The TAP Controller (a 16-state state machine).
    - One IR (Instruction Register)
    - and several DRs (Data Registers).
    - The Debug Bus for communication with the on-chip debug logic.
- The TAP defines the interface between the DTAB and the debug tool.
- The IEEE standard defines the following TAP signals, used for the serial communication and driving the TAP controller
- TDI : Test Data In : serial data from debugger to target
- TDO : Test Data Out : serial data from target to debugger
- TCK : Test Clock
- TMS: Test Mode: Select controls the TAP controller state transitions

		• TRST: Test Reset optional: resets the TAP controller	
	b)	**State any four features of Bluetooth technology.**	**4M**
	Ans:	Features: • Short range Radio Frequency at 2.4 GHz • Point-to-point or point-to-multiple points • Voice and Data • Transmit through walls up to 10m • Supports both synchronous and asynchronous services • Bluetooth 1.x supports data rate up to 1Mbps. • Bluetooth 2.0 enhanced maximum data rate of 3Mbps over 100m. • Bluetooth devices use a protocol called (FHSS) Frequency-Hopping Spread Spectrum .	**1M each**
	c)	**Draw the interfacing diagram of 4 x 4 matrix keyboard with 89C51 microcontroller. Draw the flowchart for detection and identification of key activation.**	**4M**

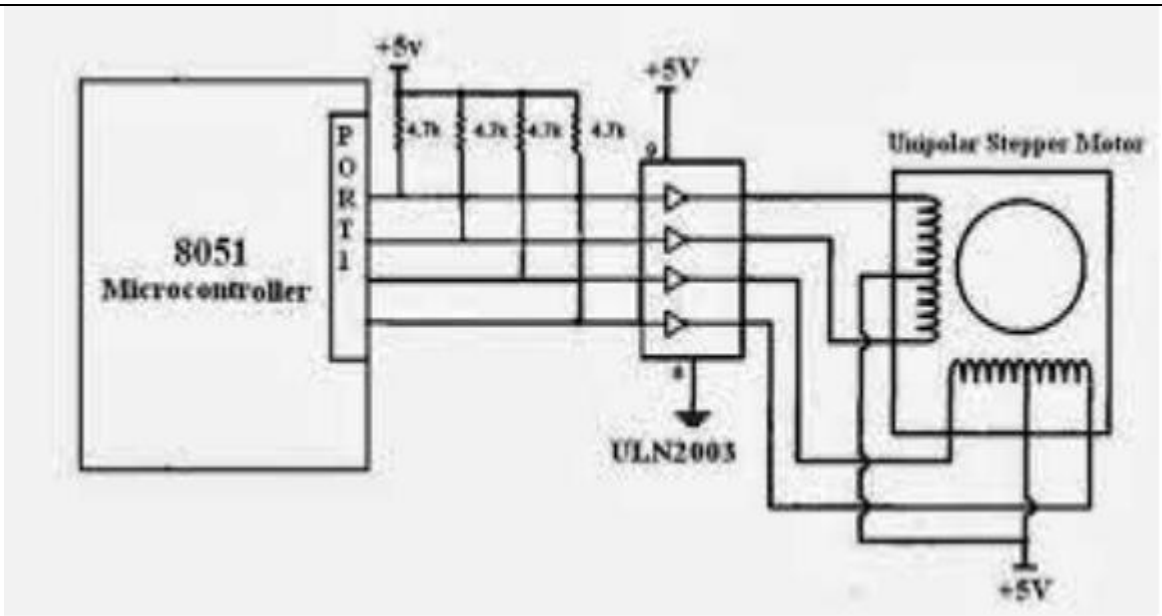
	Ans:	 	2M Diagram 2M Flowchart



d)	**Write program for 89C51 microcontroller in 'C' language to mask the lower four bits of P0 and upper four bits of P1. Combine both ports and output result on port 2.**	**4M**

Ans:	#include<reg51.h>
	Void main()
	{
	Unsigned char a,b;
	a=P0&0xF0; //masking lower 4 bits
	b=P1&0x0F; // masking upper 4 bits
	P2=a+b;
	While(1);
	}
	(For any other logic marks can be given)
		Correct program -4M

e)	**Draw interfacing diagram of stepper motor with 89C51 microcontroller.**	**4M**

	Ans:		**Correct drawing 4M**