WINTER – 19EXAMINATION

Subject Name: Embedded System **Model Answer** Subject Code: 22532

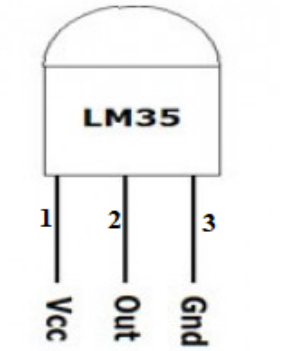
Important Instructions to examiners:

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 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.	Answer	Marking Scheme
Q.1		**Attempt any FIVE of the following:**	**10-Total Marks**
	a)	**List out four types of embedded systems.**	**2M**
	Ans:	1. Small Scale Embedded Systems 2. Medium Scale Embedded Systems 3. Sophisticated Embedded Systems 4. Stand Alone Embedded Systems 5. Real Time Embedded Systems 6. Networked Embedded Systems 7. Mobile Embedded Systems	**(any 4: ½ mark each)**
	b)	**State four advantages of embedded system.**	**2M**
	Ans:	1) Design and Efficiency 2) Cost 3) Accessibility 4) Maintenance 5) Redundancies	**(any 4: ½ Mark each)**
	c)	**State the use of MAX 232 in communication.**	**2M**
	Ans:	MAX 232 is line driver that converts from RS232 voltage levels to TTL voltage levels & vice versa in serial communication	**Correct answer 2 M**
	d)	**Illustrate any two logical operators used in C with their examples.**	**2M**
	Ans:	**Sr no:** / **Operator** / **Bitwise logical operator** / **Example** 1. / NOT / ~ / Y= ~A 2. / AND / & / Y= A&B	**(any 2 : 1mark each)**

			3.	OR	|	Y= A|B
			4.	EX-OR	^	Y=A ^B
	e)	**State two examples of RTOS.**				**2M**
	Ans:	1. LynxOS. 2. OSE. 3. QNX. 4. RTLinux. 5. VxWorks. 6. Windows CE.				**(any 2: 1 mark each)**
	f)	**Develop a 'C' program to transfer the data from port P0 to port P1.**				**2M**

	Ans:	#include<reg51.h>
		void main (void)
		{
		unsigned char X;
		P0=0XFF; // P0 as input port
		P1=0X00;//P1 as output port
		while(1) //do forever
		{
		X = P0;//read port0
		P1 = X;// output data to port1
		}
						(correct program : 2 marks) Any other correct program logic should be given marks

	g)	**Sketch pin-out diagram of LM35 and label its pin.**				**2M**
	Ans:					**2M**

Q.2		**Attempt any THREE of the following:**	**12-Total Marks**
	a)	**Compare features of PIC and AVR microcontrollers (any four)**	**4M**
	Ans:		**(any 4 features: 1 mark each)**

Parameters	PIC	AVR
Bus width	8/16/32-bit	8/32-bit
Communication Protocols	UART, USART, LIN, CAN, Ethernet, SPI, I2C	UART, USART, SPI, I2C, (special purpose AVR support CAN, USB, Ethernet)
Speed	4 Clock/instruction cycle, operating Frequency upto 20MHz.	1 clock / instruction cycle, operating frequency up to

		25MHz.
Memory	SRAM, FLASH,EEPROM	Flash, SRAM, EEPROM
ISA	Some feature of RISC	RISC
Memory Architecture	Harvard architecture	Modified
Power Consumption	Low	Low
Families	PIC16,PIC17, PIC18, PIC24, PIC32	Tiny, Atmega, Xmega, special purpose AVR
Manufacturer	Microchip Average	Atmel
Popular Microcontrollers	PIC18fXX8, PIC16f88X, PIC32MXX	Atmega8, 16, 32, Arduino Community

b) **Write a C language program to operate port 0 and port 2 as output port and port 1 and port 3 as input port.** **4M**

Ans:

```
#include<reg51.h>
void main (void )
{
unsigned char X,Y
P0=0X00; // P0 as output port
P2=0X00;// P2 as output port
P1=0XFF;//P1 as input port
P3=0XFF;// P3 as input port
while(1)
{
X= P1;
Y=P3;
P0=X;
P2=Y;
}//end of while
}//end of main
```

(correct program : 4 marks) Any other correct program logic should be given marks as there are many other ways of writing same program

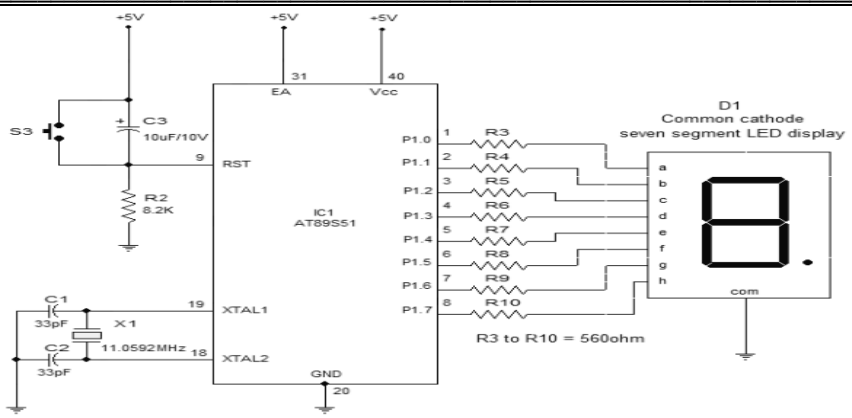
c) **Compare synchronous and asynchronous communication.(any four points)** **4M**

Ans:

SR NO.	Synchronous communication	Asynchronous communication
1.	Single clock is used for both transmitter and receiver	Two different clocks are used for both transmitter and receiver
2.	Data bits are transmitted with sync character	No sync character is required

(any 4 points : 1mark each)

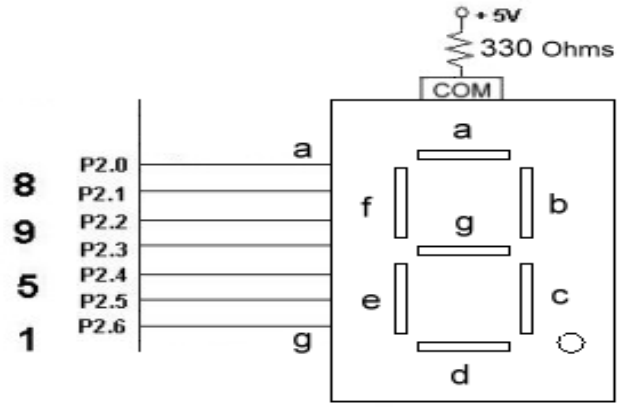
			3.	Start and stop bits are not used	Start and stop bits are used	
			4.	Used for data transfer rate >= 20 Kbps	Used for data transfer rate <= 20 Kbps	
			5.	Used for transferring block of data at a time	Used to transfer one character at a time	
			6.	Character is received at a constant rate	Character is received at a any rate	
			7.	Less reliable	More reliable	
	d)	**Explain the need to consider following factors in design matrix of embedded system:** **(i) Processor** **(ii) Memory** **(iii) Power** **(iv) Non- recurring engineering cost.**				**4M**
	Ans:	1. **Processor:** 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 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. **Power :** It is the amount of power consumed by the system which may determine lifetime of battery , or cooling requirements of the IC , since more power means more heat 4. **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.				**(1 mark each)**
Q.3		**Attempt any THREE of the following:**				**12-Total Marks**
	a)	**Sketch circuit diagram showing interfacing of one 7-segment display to 89C51. Write a 'C' program to display 'F' and 'Fi' alternately.**				**4M**
	Ans:	**Note** : • Since Fi cannot be displayed in single digit seven segment display, program is written to display F and E alternately. • Both common anode and common cathode interfacing is given here. Marks to be given if student write any one • Any other program with correct logic may be given marks **For common cathode display**:				



DISPLAY	DP	G	F	E	D	C	B	A	HEX VALUE
F	0	1	1	1	0	0	0	1	0x71
E	0	1	1	1	1	0	0	1	0x79

```
#include<reg51.h>

#define Led P1

Void Delay();

Void main()
 {
       Led = 0x00;
          while(1)
           {
             Led = 0x71;
              Delay();
              Led = 0x79;
              Delay();
            }
       }
Void Delay()
       {
          Unsigned int x,y;
           for(y=0;y<10;y++)
           for(x=0;x<1275;x++);
}
```

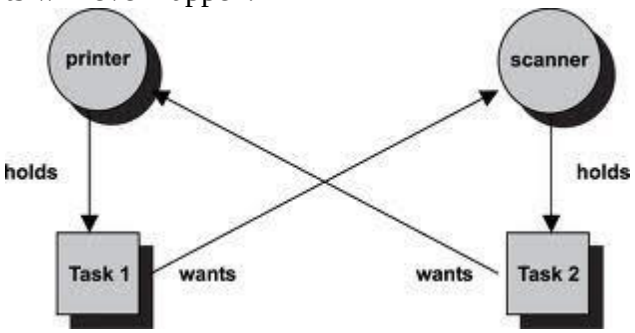
For Common anode display

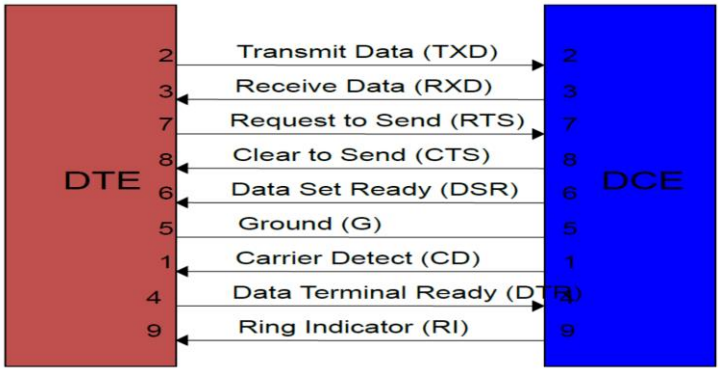


DISPLAY	DP	G	F	E	D	C	B	A	HEX VALUE
F	1	0	0	0	1	1	1	0	0x8E
E	1	0	0	0	0	1	1	0	0x86

```
#include<reg51.h>

#define Led P2

Void Delay();

Void main()

{

 Led = 0x00;
 while(1)
 {
       Led = 0x8E;
         Delay();
        Led = 0x86;
          Delay();
   }
    }
Void Delay()
    {
  Unsigned int x,y;
          for(y=0;y<10;y++)       for(x=0;x<1275;x++);
}
```

b)	**Explain the term 'Deadlock'. State reason of occurance.**	**4M**
Ans:	A deadlock is a situation where in two or more competing actions are each waiting for the other to finish, and thus neither ever does. • Assume thread/process T1 has exclusive access to resource R1. • Thread/ process T2 has exclusive access to resource R2.	**2M for Deadlock**

		• If T1 needs exclusive access to R2 and T2 needs exclusive access to R1, • Neither thread can continue. • They are deadlocked. The simplest example is that of two tasks: 1 and 2. Each task requires two mutexes: A and B. If Task 1 takes mutex A and waits for mutex B while Task 2 takes mutex B and waits for mutex A, then each task is waiting for the other to release the mutex.: These tasks may run without problems for a long time, but eventually one task may be preempted in between the wait calls, and the other task will run. In this case, Task 1 needs mutex B to be released by Task 2, while Task 2 needs mutex A to be released by Task 1. Neither of these events will ever happen.  Causes of Deadlock Mutual exclusion: only one process at a time can use a resource Hold and wait: A process holding at least one resource is waiting to acquire additional resources held by other resources No preemption: A resource can be released only voluntarily by the process holding it after that process has completed the task Circular wait: A set of processes- P1 to Pn …P1 waiting for the resource held by P2, P2 waiting for resource held by P3 etc.	**Explanation, 2M for Reasons**
	c)	**Explain the process of handshaking in RS232 standard based communication.**	**4M**
	Ans:	RS232 monitoring hardware establishes a connection between data terminal equipment (DTE) and data communication equipment (DCE). In order to link these devices, an RS232 D9 pinout is essential, as this pinout will allow you to connect two devices successfully. An RS232 pinout 9 pin cable features nine pins: 1. **Data Carrier Detect** – After a data terminal is detected, a signal is sent to the data set that is going to be transmitted to the terminal. 2. **Received Data** – The data set receives the initial signal via the receive data line (RxD). 3. **Transmitted Data** – The data terminal gets a signal from the data set, a confirmation that there is a connection between the data terminal and the data set. 4. **Data Terminal Ready** – A positive voltage is applied to the data terminal ready (DTR) line, a sign that the data terminal is prepared for the transmission of data. 5. **Signal Ground** – A return for all the signals on a single interface, the signal ground (SG) offers a return path for serial communications. Without SG, serial data cannot be transmitted between devices. 6. **Data Set Ready** – A positive voltage is applied to the data set ready (DSR) line, which ensures the serial communications between a data terminal and a data set can be completed. 7. **Request to Send** – A positive voltage indicates the request to send (RTS) can be performed, which means the data set is able to send information to the data terminal without interference. 8. **Clear to Send** – After a connection has been established between a data terminal and a	

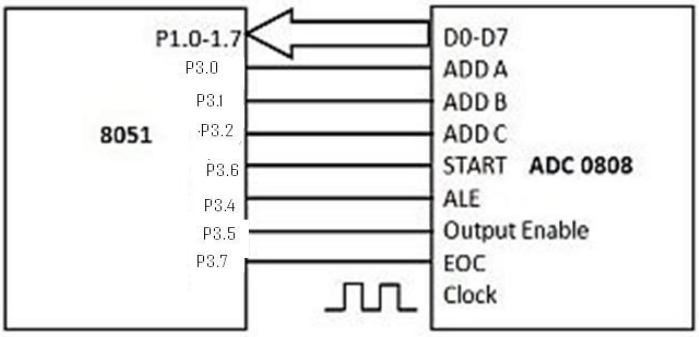
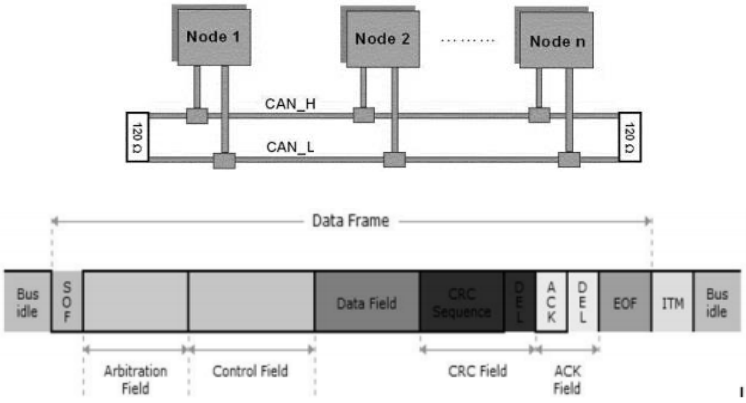
		distant modem, a clear to send (CS) signal ensures the data terminal recognizes that communications can be performed. 9. **Ring Indicator** – The ring indicator (RI) signal will be activated if a modem that operates as a data set detects low frequency. When this occurs, the data terminal is alerted, but the RI will not stop the flow of serial data between devices.  	
	d)	**Write a 'C' language program to mask the upper four bits of the data given in port 0 and write the answer in port 1.**	**4M**

	Ans:	#include<reg51.h>
		Void main()
		{
		unsigned char a ,b;
		P0 = 0xff; //P0 as an input port
		P1 = 0x00; //P1 as an output port
		while(1)
		{
		a = P0;
		b=a & 0x0F; //masking lower 4 bit
		P1=b;
		}
		}
		(For any other logic marks can be given)
			4M

Q.4		**Attempt any THREE of the following :**	**12- M**
	a)	**Write 'C' program to generate delay of 50msec for microcontroller 89C51 with crystal frequency of 11.0592 MHz.**	**4M**

	Ans:	**Use Timer 0, mode 1 (16-bit) to create the delay. Assume XTAL=11.0592 MHz=> T=1.085µs Count=50ms/1.085µs =46083**
		Initial count = 65536-46083 =19453, Count in Hex = 4BFDH
		
		`#include <reg51.h>`
		`void Delay(void);`
		`sbit mybit=P1^5;`
		`void main(void)`
		`{`
		`while (1)`
		`{`
		`mybit=~mybit; //toggle P1.5`
		`Delay();`
		`}}`
		
		`void Delay(void)`
		`{`
		`TMOD=0x01; // Timer 0, mode 1`
		`TL0=0xFD;`
		`TH0=0x4B;`
		`TR0=1;`
		`while (TF0==0);`
		`TR0=0;`
		`TF0=0;`
		`}`
		(For any other logic marks can be given)
			1M for count calculation, 3M for Program

	b)	**List out eight features of USB.**	**4M**
	Ans:	The Universal Serial Bus has the following features: • The computer acts as the host. • Up to 127 devices can connect to the host, either directly or by way of USB hubs. • Individual USB cables can run as long as 5 meters; with hubs, devices can be up to 30 meters (six cables' worth) away from the host. • With USB 2.0,the bus has a maximum data rate of 480 megabits per second (10 times the speed of USB 1.0). • A USB 2.0 cable has two wires for power (+5 volts and ground) and a twisted pair of wires to carry the data. The USB 3.0 standard adds four more wires for data transmission. While USB 2.0 can only send data in one direction at a time (downstream or upstream), USB 3.0 can transmit data in both directions simultaneously. • On the power wires, the computer can supply up to 500 milliamps of power at 5 volts. A USB 3.0 cable can supply up to 900 milliamps of power.	**4 points. 1M each**

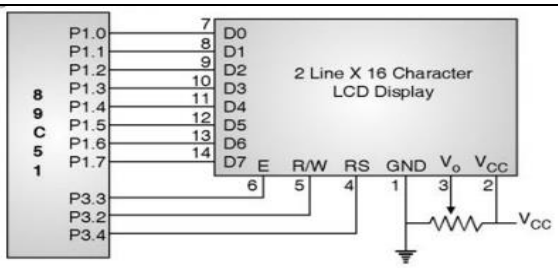
		• Low-power devices (such as mice) can draw their power directly from the bus. High-power devices (such as printers) have their own power supplies and draw minimal power from the bus. Hubs can have their own power supplies to provide power to devices connected to the hub. • USB devices are **hot-swappable**, meaning you can plug them into the bus and unplug them any time. A USB 3.0 cable is compatible with USB 2.0 ports -- you won't get the same data transfer speed as with a USB 3.0 port but data and power will still transfer through the cable. • Many USB devices can be put to sleep by the host computer when the computer enters a power-saving mode.	
	c)	**Draw the interfacing diagram of ADC with 89C51 and state the function of SOC, EOC and OE pins.**	**4M**
	Ans:	**SOC [Start of conversion]:** When High to low signal is appearing to this pin of ADC; ADC then starts conversion. **EOC [End of conversion]:** ADC sends this high EOC signal to Micro-Controller to indicate completion of conversion. **OE [Output Enable]:** When a high signal is applied to this pin, the output latch of ADC get enables and the converted data is then available to Micro-Controller. 	**function s – 1M each** **diagram – 1M**
	d)	**Explain 'CAN' bus protocol and list out its two applications.**	**4M**
	Ans:	Controlled Area Network [CAN]:- Can is mainly used in automotive electronics. CAN bus is a standard bus in distributed network. It has a bi-directional serial line which receives or sends a bit at an instance by operating at maximum rate of 1Mbps. It employs a twisted pair connection to each node. The pair can run to a maximum length of 40m. 	**explanation – 3M, applications – ½ M each**

Field and its Length	Description of each field in CAN frame
1st field of 12 bits	It is called arbitration field. It contains the packet 11-bit destination address and the RTR [Remote Transmission Request]. When this bit is at 1 this indicates the packet is for the destination address. If this packet for request for a data from a device defined by identifier. The device is at destination address specified in the field.
2nd field of 6 bits	It is called a control field. The 1st bit id the identifier extension. The 2nd bit is always 1, and the last 4 bits are code for data length.
3rd field of 0-64 bits	Its length depends on data length code in the control field.
4th field of 16-bits { 3rd if data field has no bit present}	It is the CRC word. The receiver node uses it to detect errors during transmission.
5th field of 2 bits	1st field is the ACK slot. The sender sends it as 1 and RX sends back 0 in this slot when the receiver detects an error in the reception. Sender after sensing 0 in the ACK slot transmits the data frame. The 2nd bit is the ACK delimiter bit. It signals the end of the ACK field. If the transmitting node does not receive and ACK of data frame within a specified time slot it should retransmit.
6th field of 7-bits	It is for end of the frame specification and has seven 0's.

Applications: Copiers, Telescopes, Medical instruments, Elevator controllers, Automobile industry**.**

e) Sketch interfacing diagram to interface LCD display with 89C51. — **4M**

Ans: — **4M**

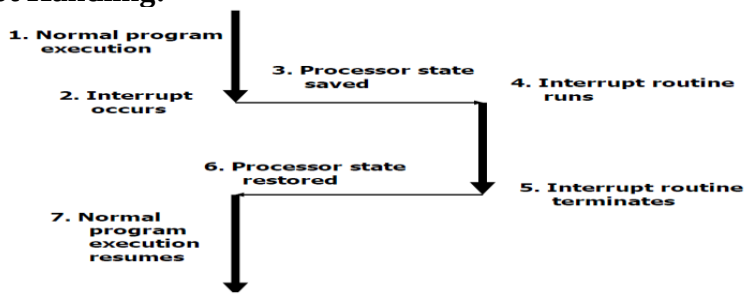


Q.5 — **Attempt any TWO of the following:** — **12 Total Marks**

(a) Explain resource allocation and interrupt handling function of RTOS. — **6M**

Ans:

i) Interrupt Handling:

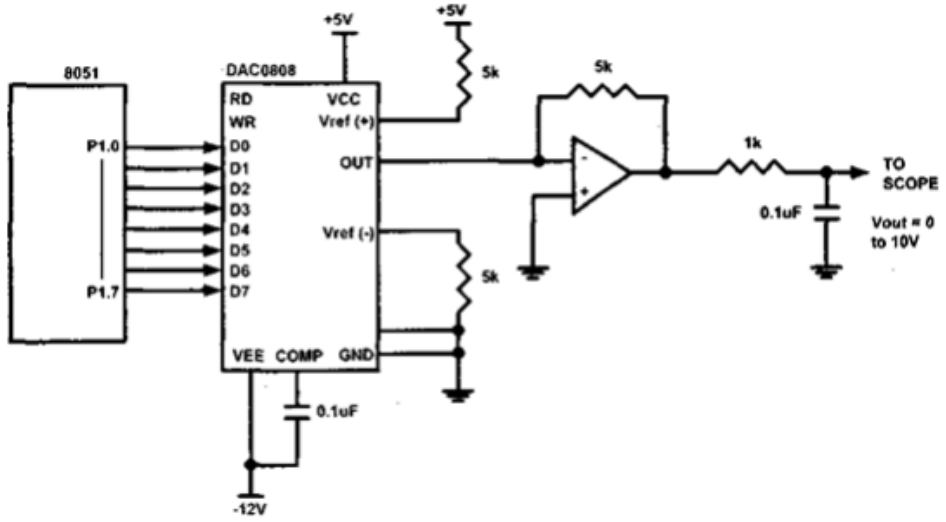


When an interrupt occurs, interrupt Service Routines (ISR) is run.
Most interrupt routines in RTOS Copy peripheral data into a buffer , Indicate to other code that data has arrived and Acknowledge the interrupt (tell hardware)
RTOS normally disable the interrupts while handling critical section and enable after the critical section has been executed. Interrupt latency is a factor to look for, when selecting a

3M- Interrupt Handling,

3M- Resource Allocation

		RTOS. Interrupt latency = Maximum amount of time interrupts are disabled+ time to start execution of first instruction of ISR. It is desirable that RTOS should have minimum interrupt latency **ii) Resource allocation:** Sharing of resources by competing tasks as per their execution schedules is a function RTOS. This means that tasks should have the required resources allocated to them whenever they are needed. The Operating System allocates resources when a task need them. When the task terminates, the resources are de-allocated, and allocated to other tasks that need them. Resources can be allocated in Round Robin method or Priority based. Some resources are non-Pre emptible eg. Mutex. In Round Robin, tasks are scheduled in FIFO manner. Fixed Time quantum is given to the tasks after which it is pre-empted. Priority Scheduling, resources are allocated to processes according to priorities.	
	(b)	**Write a 'C' language program for 89C51 to generate triangular waveform.**	**6M**

	Ans:	
		
		Program:
		#include < reg51.h>
		unsigned char d;
		void main(void)
		{
		while(1)
		{
		for(d=0; d<255; d++)
		{
		P1 = d;
		}
		for(d=255; d>0; d--)
		{
		P1 = d;
		}
		}
		}
			2M-Diagram,
			4M-Program

		(For any other relevant logic marks can be given)	
	(c)	**Write a 'C' language program for serial communication to transfer letter 'M' serially at 9600 baud continuously.**	**6M**

	Ans:	#include <reg51.h>
		void main(void)
		{
		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
		while(1)
		{
		SBUF='M'; // place value in buffer
		while(TI==0);
		TI=0; // clear TI
		}
		}
			4M Program, 2 M comments

Q.6		**Attempt any TWO of the following:**	**12Total Marks**
	(a)	**List out characteristics of RTOS and explain any four characteristics.**	**6M**
	Ans:	**Characteristics of RTOS:** 1. Reliability 2. Consistency 3. Predictability 4. Performance 5. Scalability 6. Compactness • Reliability: A reliable system is one that is available (continues to provide service) and does not fail. Embedded systems and hence RTOS used in such systems must be reliable. • Consistency: A key characteristic of an RTOS is the level of its consistency concerning the amount of time it takes to accept and complete an application's task; the variability is 'jitter'. A 'hard' real-time operating system has less jitter than a 'soft' real-time operating system. • Predictability: The RTOS used in this case needs to be predictable to a certain degree. The term deterministic describes RTOSes with predictable behavior, in which the completion of operating system calls occurs within known timeframes. • Performance: This requirement dictates that an embedded system must perform fast enough to fulfill its timing requirements. • Scalability: Because RTOSes can be used in a wide variety of embedded systems, they must be able to scale up or down to meet application-specific requirements. • Compactness: In embedded systems, where hardware real estate is limited due to size and costs, the RTOS clearly must be small and efficient. In these cases, the RTOS memory footprint can be an important factor.	**2M list,** **1M each characteristic explanation**
	(b)	**Compare:** **(i) RISC with CISC processor**	**6M**

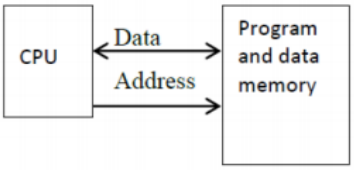
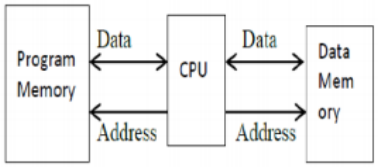
(ii) Harward with Von Neuman architecture.

Ans:

i) RISC with CISC processor — **1M each**

SR. NO.	RISC	CISC
1	Reduced instructions take 1 cycle	Complex instructions require multiple cycles
2	Only Load and Store instructions can reference memory	Many instructions can reference memory
3	Uses pipelining to execute instructions	Instructions are executed one at a time
4	Many general registers	Few general registers
5	Emphasis on software	Emphasis on hardware

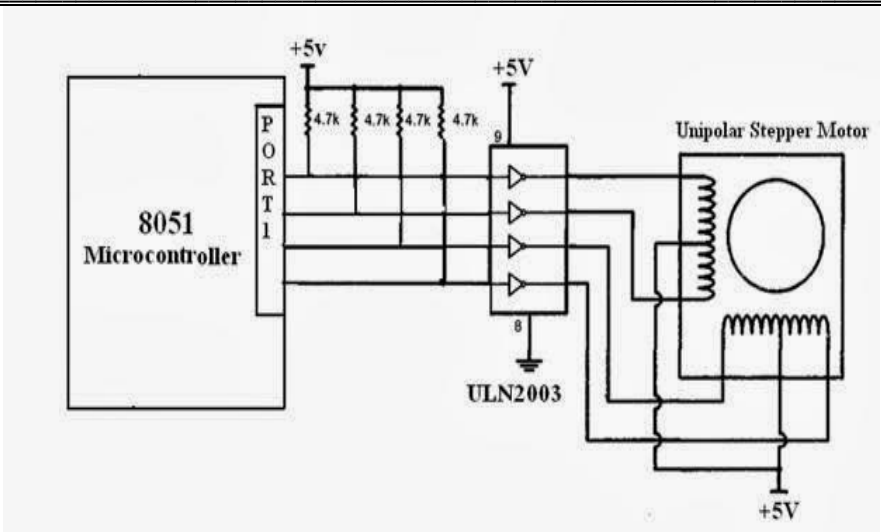
ii) Harward with Von Neuman architecture.

Sr.No	Von Neumann architecture	Harvard architecture
1		
2	The Van Neumann architecture uses single memory for their instructions and data.	The Harvard architecture uses physically separate memories for their instructions and data.
3	Requires single bus for instructions and data	Requires separate & dedicated buses for memories for instructions and data.
4	Its design is simpler	Its design is complicated
5	Instructions and data have to be fetched in sequential order limiting the operation bandwidth.	Instructions and data can be fetched simultaneously as there is separate buses for instruction and data which increasing operation bandwidth.
6	Program segments & memory blocks for data & stacks have separate sets of addresses.	Vectors & pointers, variables program segments & memory blocks for data & stacks have different addresses in the program.

(Any 3 points for RISC & CISC and Harvard & Von- Neumann)

(c) Explain with sketch interfacing of stepper motor with 89C51. Write 'C' language program to rotate the motor clockwise. — **6M**

Ans:



3M- Diagram

Program:

3M- program

```
# include <reg51.h>
void delay (unsigned int);
void main (void)
{
while(1)
{
P1=0x99;
delay (100);
P1=0xCC;
delay(100);
P1=0x66;
delay(100);
P1=0x33;
delay(100);
}
}
void delay(unsigned int k)
{
unsigned int x,y;
for (x=0;x<k;x++)
for (y=0;y<1275;y++);
}
```