WINTER– 17 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 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.

1a) Attempt any three of the foll. 12 marks

i) State and describe any 4 design matrices of embedded system.

Ans. (any four matrices - 1 mark each)

1)Processor power

- Selection of the processor is based on the amount of processing power to get the job done and the register width required.
- 8 bit, 16 bit, 32 bit and 64 bit microcontrollers are provided.
- Processing power is different for different microcontrollers.
- High clock, speed and addressing capable microcontrollers are available.
- Very powerful DSPS are available for real time analysis of audio and video signals.

2)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.

3)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 the embedded system.
- It is categorized as follows : Embedded OS, real time OS and mobile OS. These operating systems occupy less area in memory than desktop.
- For real time applications, we should use real time OS.
- We can develop our own OS kernel.
- We can use open source OS like Linux. This OS is free and can be customized.

4)Reliability

- Embedded system often reside in machines that are expected to run continuously for years without errors and in some cases recover by themselves, if an error occurs.

- So, the software is usually developed and tested more carefully than that for personal computers and unreliable moving parts such as disk drives, switches or buttons are avoided.

5)Unit cost

- The monetary cost of manufacturing each copy of the system, excluding NRE cost.

6)NRE cost

- The monetary cost of designing the system. Once the system is designed, any number of units can be manufactured without incurring any additional design cost (hence the term "non-recurring").

7)Size

- The physical space required by the system, often measured in bytes for software, and gates or transistors for hardware.

8)Performance

- The execution time or throughput of the system.

9)Power

- The amount of power consumed by the system, which determines the lifetime of a battery, or the cooling requirements 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. Software is typically considered very flexible.

11)Time to market

- The amount of time required to design and manufacture the system to the point the system can be sold to customers.

12)Time to prototype

- The amount of time to build a working version of the system, which may be bigger or more expensive than the final system implementation, but can be used to verify the system's usefulness and correctness and to refine the system's functionality.

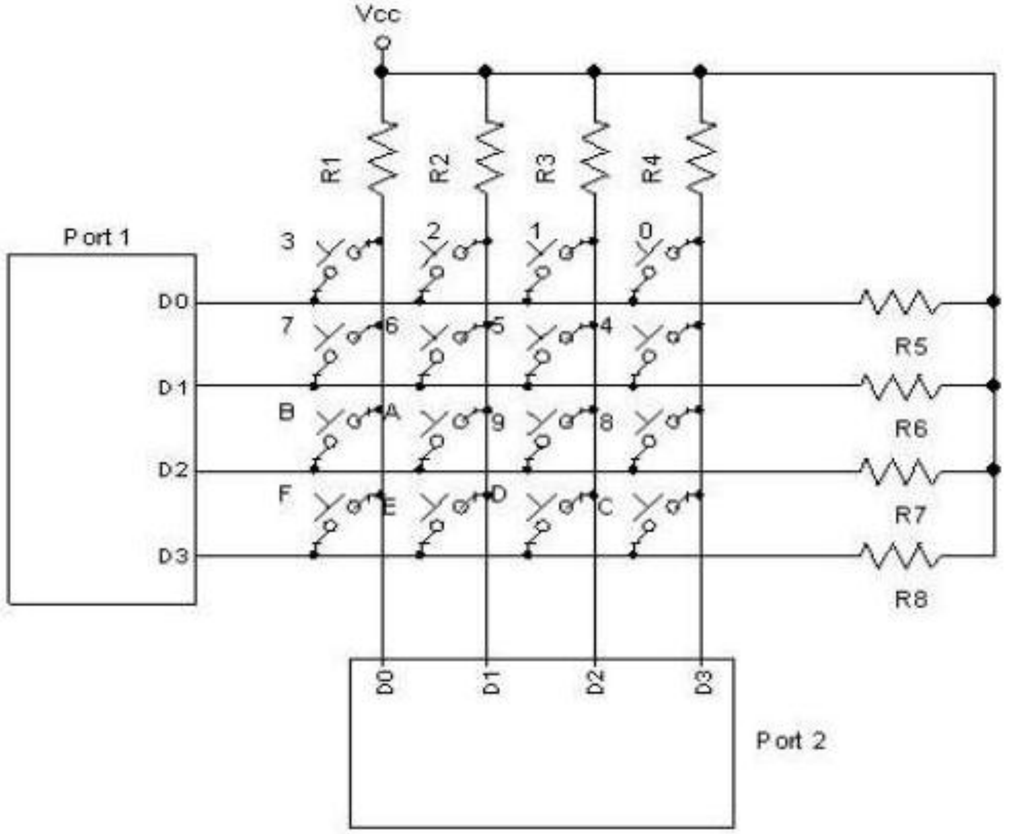
13)Correctness

- Our confidence that we have implemented the system's functionality correctly. We can check the functionality throughout the process of designing the system and we can insert test circuitry to check that manufacturing was correct.

14) Safety

- The probability that the system will not cause harm.

ii) Draw interfacing diagram of 4*4 matrix keyboard with 89c51 µc.
Ans:- (Proper relevant diagram- 4 mks)



iii) Describe the function of following software development tools for 89c51 µc.
1) Compiler
2) Linker
3) Debugger
4) Cross compiler

Ans: (Each function- 1mks)

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.

Linker/Locator:

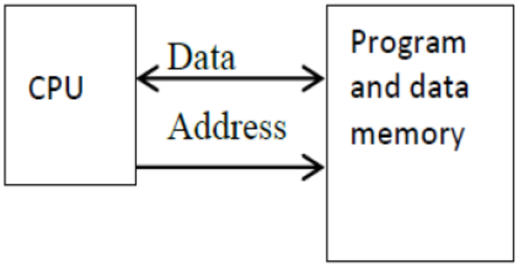
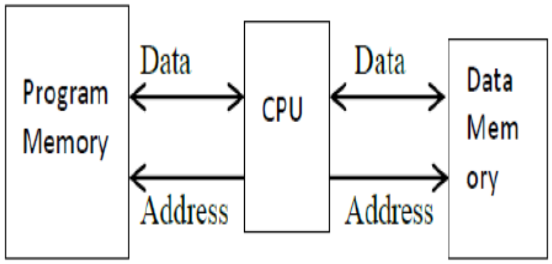
It is used for relocation process.
It is done during compilation also it can be done at run time by a relocating loader.
It is a program that takes one or more objects generated by compiler and combines them into a single executable program.

A **Debugger** allows you to download your code to the emulator's memory and then 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).

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.

iv) Compare Von – Neumann and Harvard Architecture.
Ans:- (Relevant 4 points – 4 mks)

Sr.No	Von Neumann architecture	Harvard architecture
1		 Program Memory Data Address CPU Data Address Data Mem ory
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.

b) Attempt any One of the following: 6 marks

i. **List any six data types in embedded 'C' with their size in bits and data range.**

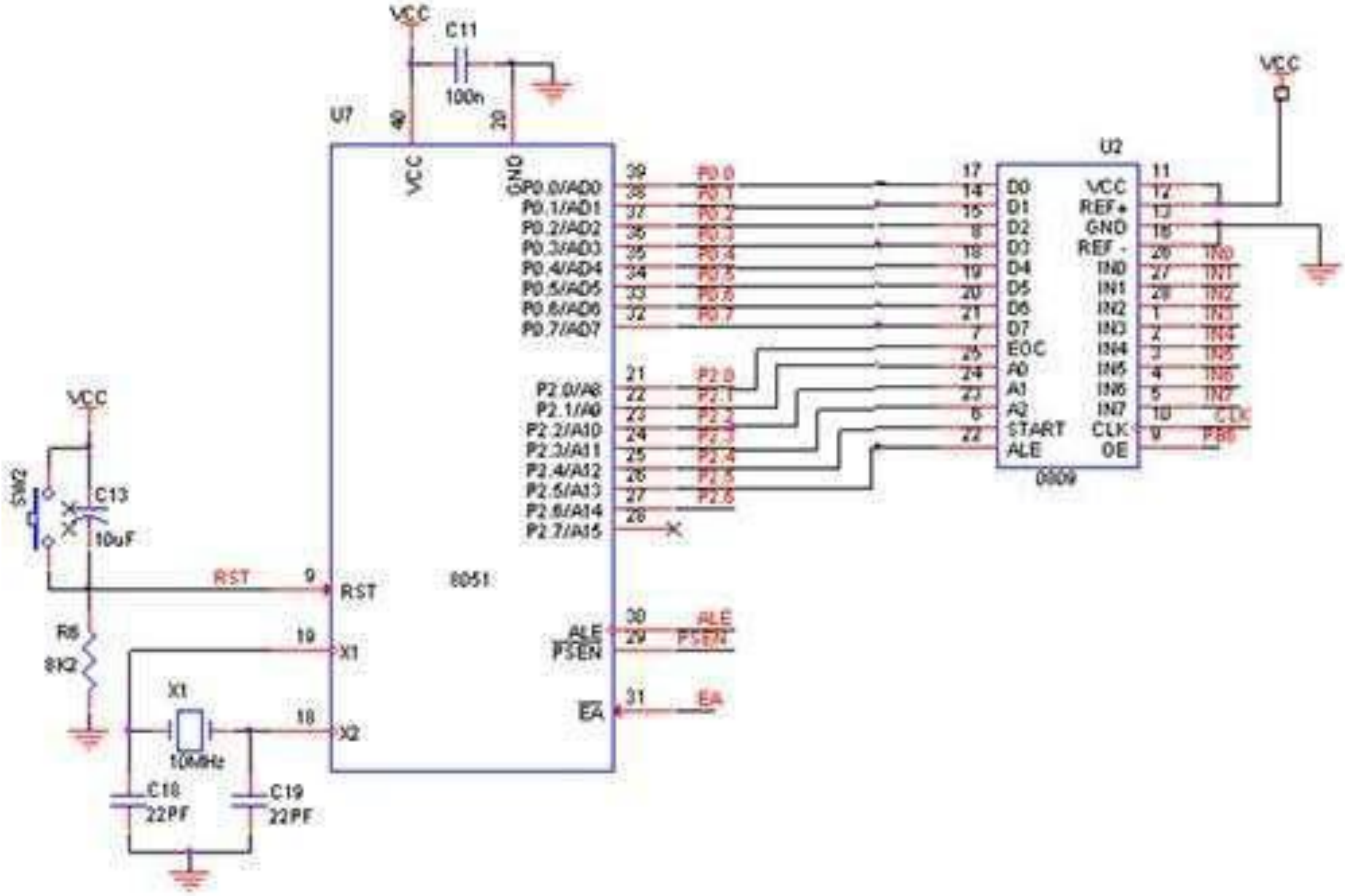
Ans:- (Six Data Type- 1 mk each)

Data types	**Size in bits**	**Data Range/usage**
Unsigned char	**8**	**0 to 255**
Signed char	**8**	**-128 to +127**
Unsigned int	**16**	**0 to 65535**
Signed int	**16**	**-32768 to +32767**
Sbit	**1**	**SFR bit addressable only**
Bit	**1**	**RAM bit addressable only**
sfr	**8**	**RAM addresses 80-FF H only**

ii. **Draw interfacing diagram of ADC with 89c51 µc and explain function of following pins of ADC.**

1) **SOC,**
2) **EOC**
3) **OE**

Ans:- (Diagram- 3 mks, each definition- 1 mks)



1) **SOC- Start of conversion of analog data to digital**
2) **EOC-End of conversion of analog data to digital**
3) **OE-output enable to enable the ADC IC**

iii) State different scheduling algorithms of RTOS and describe Round Robin

scheduling algorithm.

Ans:- (Different methods- 3 mks, description of round robbin algorithm- 3 mks)

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

Round robin algorithm

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

The kernel gives control to the next task if

a) the current task has completed within the time slice

b) the current task has no work to do

c) the current task has completed its allocated time slice

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

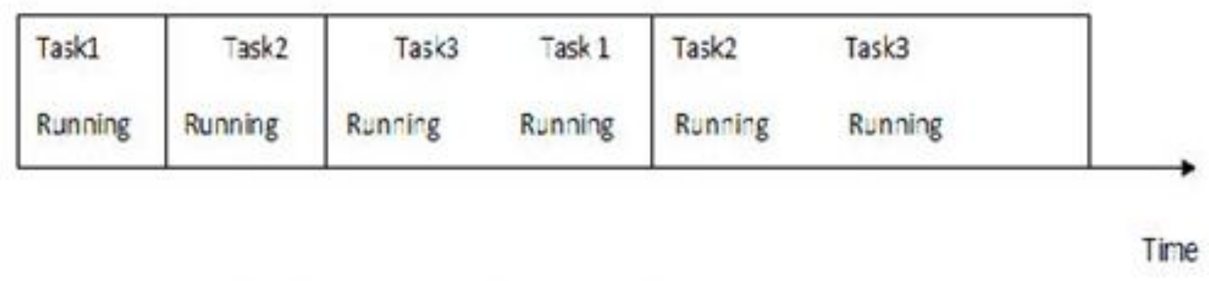


Fig. Round robin scheduling algorithm

2 Attempt any Four of the following : 16 Marks

a. Compare synchronous and asynchronous type of serial communication.

Ans:- (Any four relevant points- 4 mks)

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
7	Error checking is not possible	Error checking is possible with parity bit.

b) State any two features of IDE and ICE.
Ans:- (Two features of each- 2 mks)

Features of IDE-

1)It supports for defining a processor family and its version.\

2) Supports a user definable assembler to support a new version

3) Provides multiuser environment

4) Supports conditional and unconditional break points.

5) Provides debugger

Features of ICE

1. It is hardware device to debug the software of embedded system.
2. Provides a window into the embedded system.
3. It emulates the CPU of the embedded systems computer.

c) State any two advantages and two applications of embedded system.
Ans:- (Two advantages-2mks, two applications-2 mrks)

Embedded systems play a vital role in our day to day life, starting from home to the computer industry, where most of the people find their job for a livelihood.

1. **Consumer electronics :**Camcoders, cameras, mp3 players, DVD players etc.
2. **Household applications :** Television, washing machine, fridge, microwave oven etc.
3. **Home automation and security systems :** Air conditioners, sprinklers, intruder detection alarms, fire alarms etc.
4. **Automotive industry :** Antilock breaking system, engine control, ignition systems, navigation system etc.
5. **Telecom :** Cellular telephones, handset multimedia applications.
6. **Computer peripherals :** Printers, scanners, fax machine etc.
7. **Computer networking system :** Network routers, switches, hubs, firewalls etc.
8. **Healthcare :** Different kind of scanners, ECG machines etc.

9. **Measurement and instrumentation :** Digital multi meters, digital CROs, logic analysers, PLS systems etc.
10. **Banking and Retail :** Automatic teller machines (ATM) and currency counters.
11. **Card Readers :** Barcode, smart card readers etc.

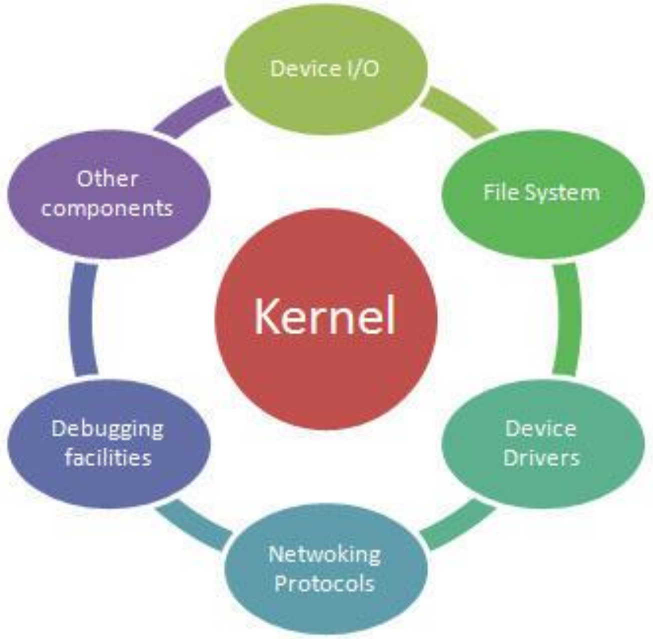
Advantages- (any two)

1. **Design and Efficiency :** The central processing core in embedded systems is generally less complicated making it easier to maintain.
2. **Cost :** The streamline makeup of most embedded system allows their parts to be smaller less expensive to produce.
3. **Accessibility :** Embedded systems are difficult to service as they are embedded inside the machine, so they have to be developed carefully.
4. **Maintenance :** They are easier to maintain because the supplied power is embedded in the system and does not require remote maintenance.
5. **Redundancies :** Embedded systems do not involve the redundant programming and maintenance involved in system models.

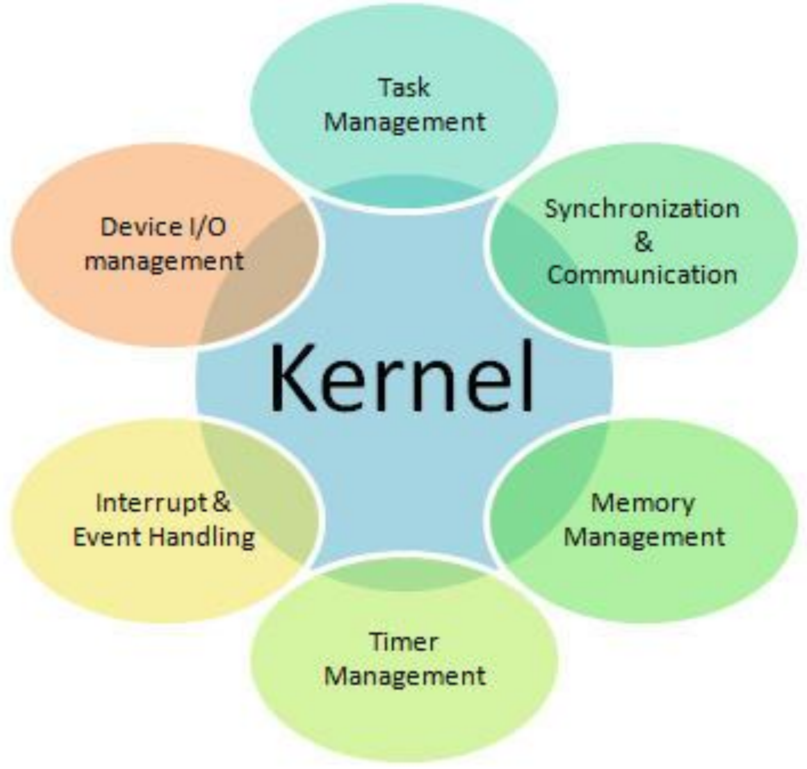
d) Draw and describe Architecture of RTOS.

Ans:- (Diagram- 2 mks, description- 2 mks)

- For simple applications, RTOS is usually a kernel but as complexity increases, various modules like networking protocol, debugging facilities, device I/Os are included in addition to the kernel.
- The general architecture of RTOS is shown in the figure:

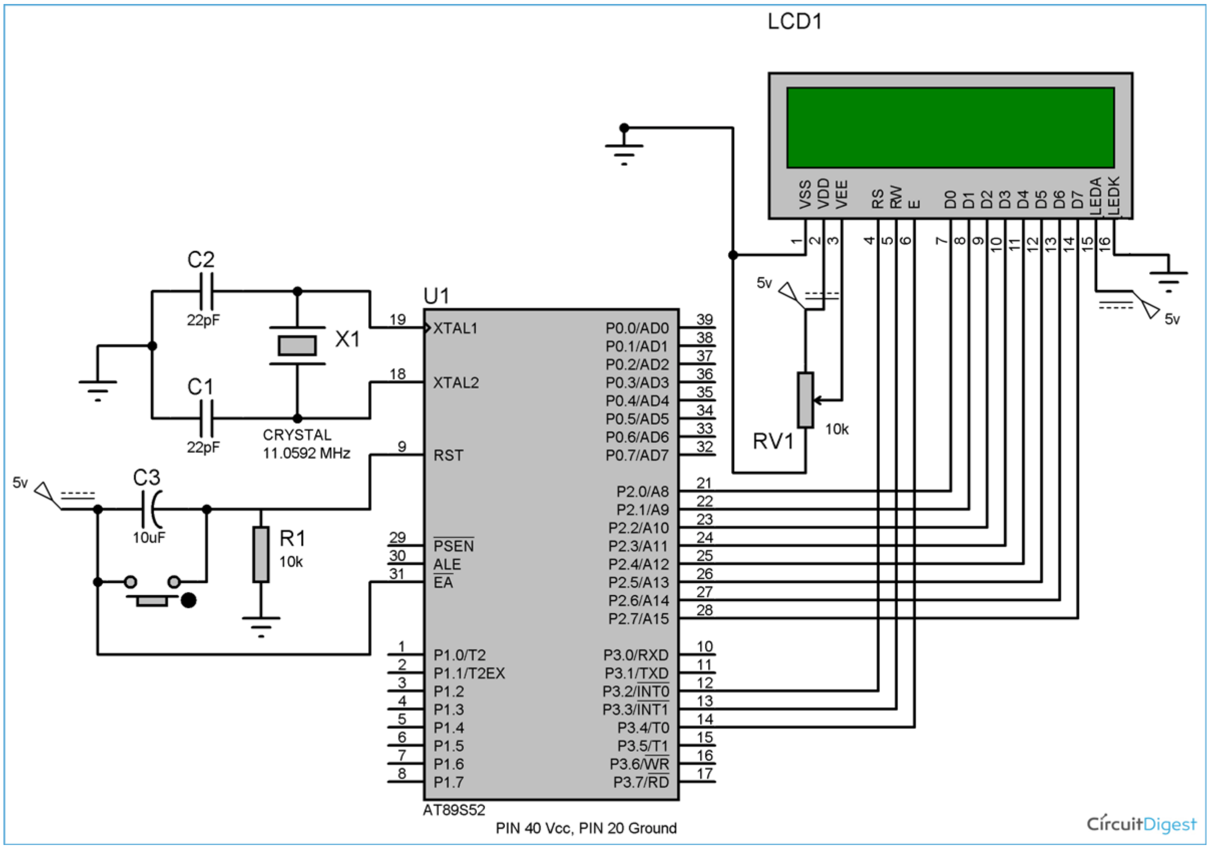


- An operating system generally consists of two ports: kernel space and user space.
- RTOS kernel acts as an observation layer between the hardware and the applications.
- Kernel is the smallest and central component of an operating system. Its services include managing memory and devices and to provide an interface for software applications to use the resources.
- Six types of common services provided by the kernel are shown below in the figure:



e) Draw interfacing diagram of LCD with 89c51 µc.

Ans:- (Proper relevant diagram- 4 mks)



3.Attempt any Two of the following : 16 Marks

a) Draw interfacing diagram of DAC with 89c51 µc and Write 'c' language program to generate triangular waveform using DAC.

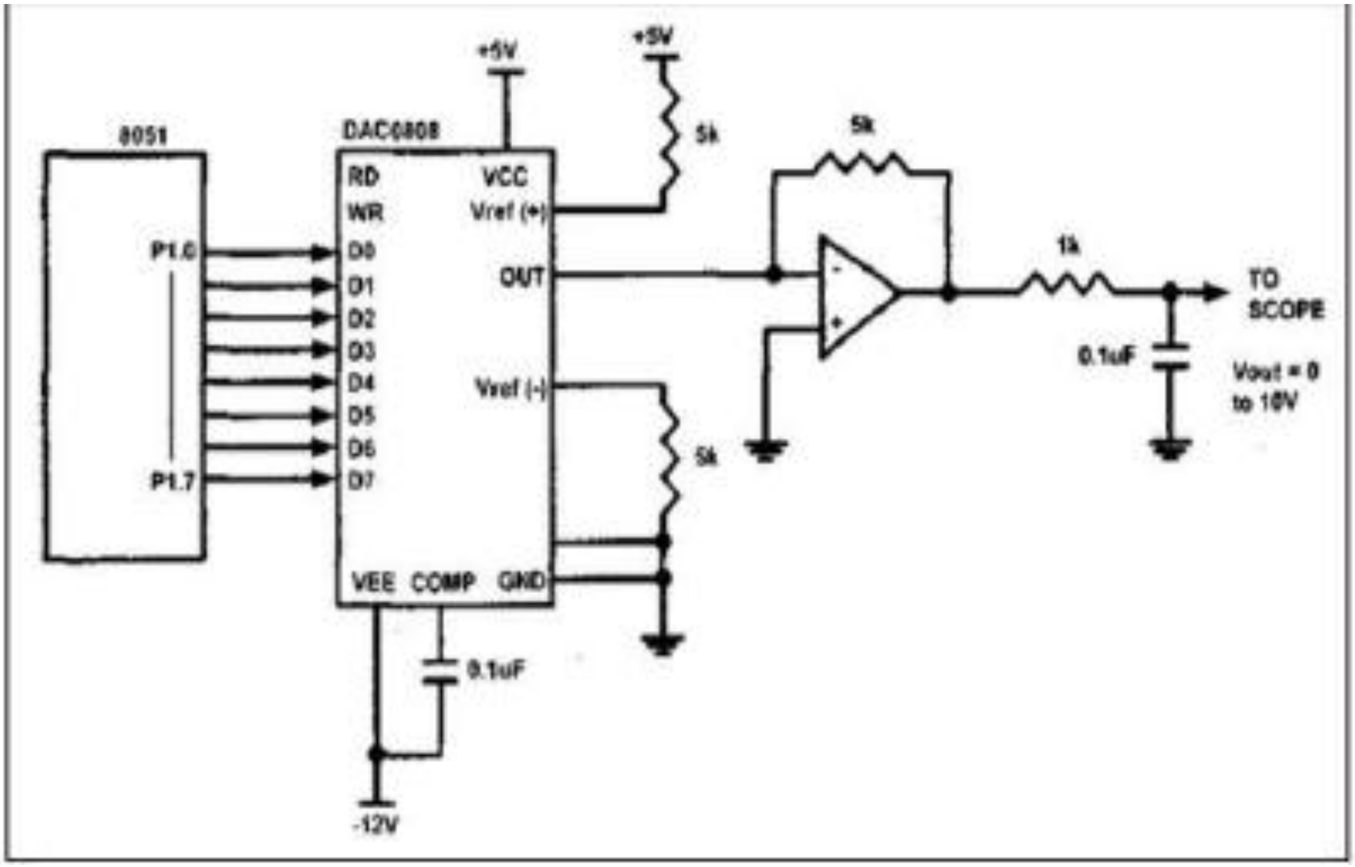
Ans:- (Diagram- 4 mks, program – 4 mks)

Program:

Note : Any ports pin can be defined for handshaking signals.

```
#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;
}
}
}
```



b) Write 'c' language program to generate Square wave of frequency 5 KHz on pin 3.5 of 89c51 µc. Use timer 1, mode 1 to generate delay. Assume XTAL=11.0592 MHz .

Ans:- (Calculation- 2 mks, program-6 mks)

Crystal frequency= 11.0592 MHz

I/Pclock=11.0592 X 106 =11.0592MHz

1/12x11.0592Mhz= 921.6Khz

Tin =1.085µsec

For5 kHzsquarewave

Fout = 5KHz

Tout =1/5 X 10^3

Tout =200µsec

Consider half ofit=Tout = 100µ sec

N =Tout / Tin =100/1.085= 92.16

65536-92= 65444(10)=FFA5 H

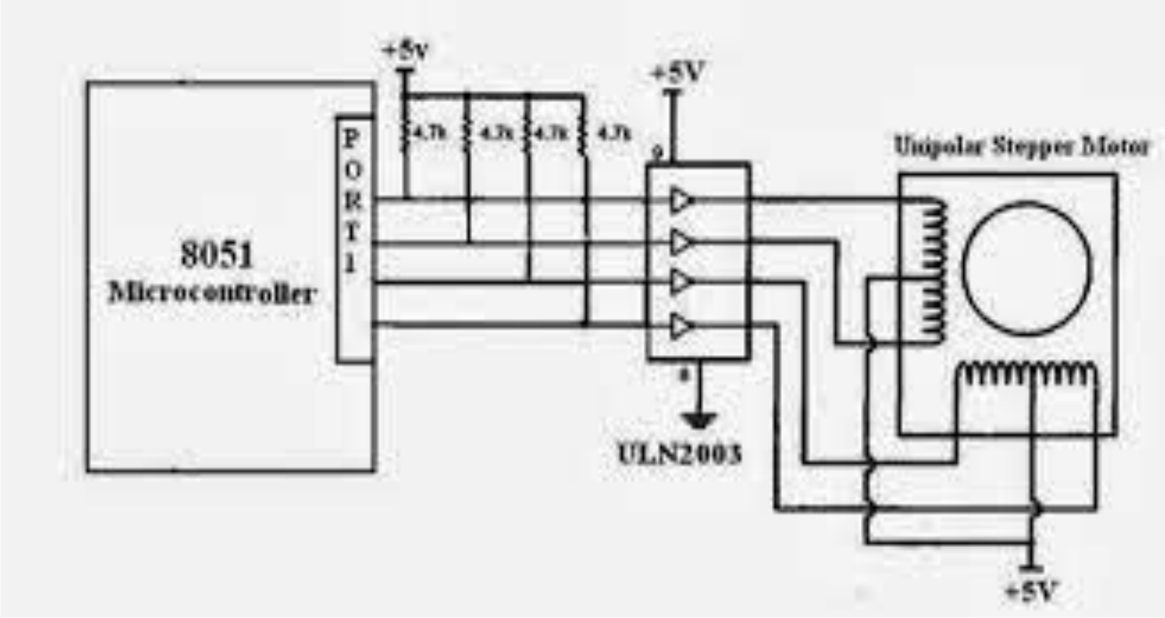
Program

```
#include<reg51.h>

void delay(void);

sbit p=P3^5;

void main (void)

{

while (1)

{

        p=~p;

delay();

}

            }


void delay()
```

```
{
        TMOD=0X10;          //set timer 1 in mode 1 i.e. 16 bit number
        TL0=0XA5H;          //load TL register with LSB of count
        TH0=0XFFH ;         //Load TH register with MSB of count
         TR1  =1 ;            //Start timer 0
While(TF1==0)        //wait until timer rolls over
  TR1=0;               //Stop timer 0
 TF1=0;               //Clear timer flag 0
   }
```

c) Write 'c' language program to rotate stepper motor in clockwise direction continuously. Draw interfacing diagram of stepper motor with 89c51 µc.

Ans:- (Interfacing diagram- 3 mks, program- 3 mks)



Program:

#include<reg51.h>

```
main()
{
void delay(void);
while( 1)
{
P0=0x06;
delay();
P0=0x0C;
delay();
P0=0x09;
delay();
P0=0x03;
delay();

}
}
void delay(void)
{
unsigned char cnt, cnt1;
for(cnt=0; cnt<=254;cnt++)
for(cnt1=0;cnt1<254;cnt1++);
}
```

4. Attempt any three of the following : 12

i. Compare Bluetooth and zigbee wireless communication protocols.

Ans:- (Four relevant points- 4 mks)

Bluetooth features	**Zigbee features**
IEEE Standard 802.15.1	IEEE Standard 802.15.4
Frequency (GHz) 2.4	Frequency (GHz) 0.868, 0.915, 2.4
Maximum raw bit rate (Mbps) 1-3	Maximum raw bit rate (Mbps) 0.250
Typical data throughput (Mbps) 0.7-2.1	Typical data throughput (Mbps) 0.2
Maximum (Outdoor) Range (Meters) 10 (class 2), 100 (class 1)	Maximum (Outdoor) Range (Meters) 10-100

ii. State any 4 features of USB serial communication protocol.

Ans:-(any four relevant features- 4 mks)

1. **Multiple device connection:** Upto 127 different devices can be connected on single USB bus.
2. **Transfer rate:** The initial USB supported 12 MBps transfer rate where USB 2.0 supports higher rate currently 60 MB/sec.

3. **Support for large range of peripherals**: Low bandwidth devices such as keyboard, mouse, joystick, and game -port, FDD

.

4. **Hub architecture:** The devices are not daisy chained. Each device is connected to an USB

hub. The USB hub interacts with PC on one side and peripheral on other side.

5. **Plug ability:** The USB device can be connected without powering off a PC i.e. plug and play feature in BIOS together with the device takes care of detection, handling and device recognition.

6. **Power allocation:** USB controller in the PC detects the presence or absence of the USB devices and does the allocation of power.

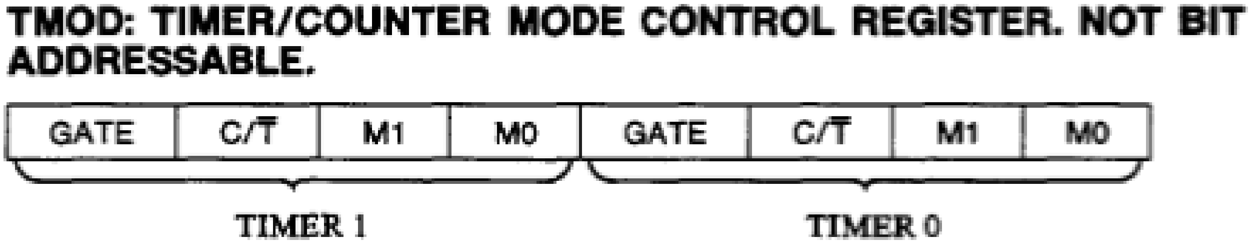
7. **Ease of installation:** There is only one cable. A 4-pin cable carries signals like power signal

(-), signal (+), ground.

8. **Host centric:** The CPU software initiates every transaction on the USB bus. Hence the overhead on the PC increases when there are large number of peripherals involving large number of transactions.

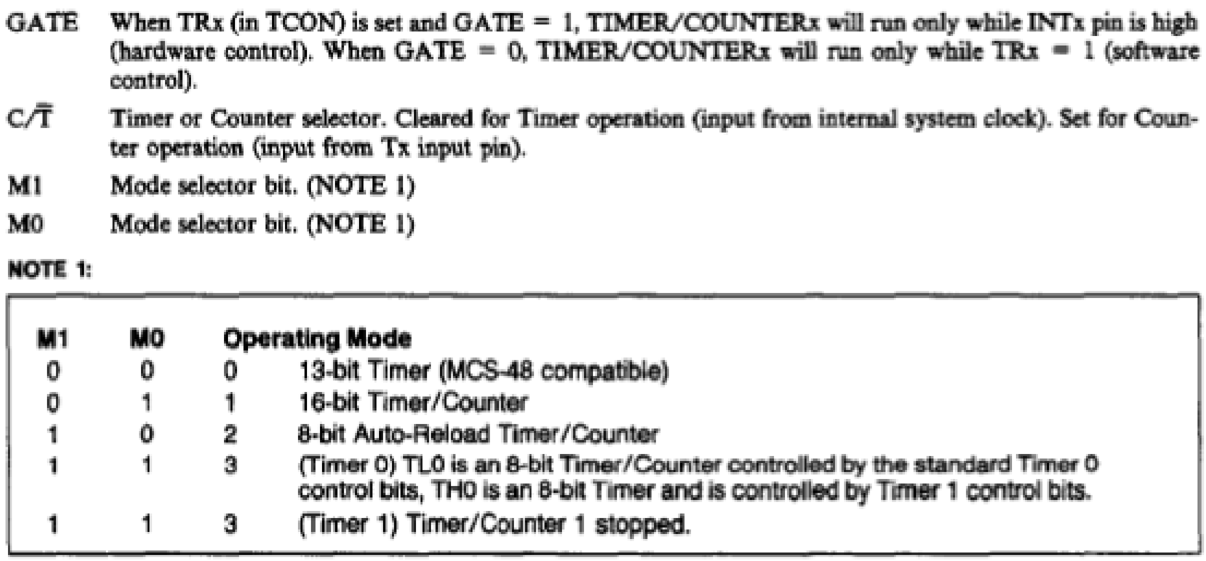
iii. **Draw 8 bit format of TMODSFR and explain how modes of timer can be selected using TMOD.**

Ans:- (format- 2 mks, explanation- 2 mks)



TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
TIMER 1				TIMER 0			



GATE When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while $\overline{\text{INTx}}$ pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).

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

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

NOTE 1:

M1	M0	Operating Mode	
0	0	0	13-bit Timer (MCS-48 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

iv. **Describe the function of following:**

1) Simulator 2) Emulator

Ans:- (Each description- 2 mks)

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

Emulators 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

b) Attempt any One of the following: 6

i) Write ‘c’ language program to toggle bit P1.5 of part 1 continuously after 50ms delay.
Generate delay using for loops.

Ans:- (Relevant program- 6 mks)

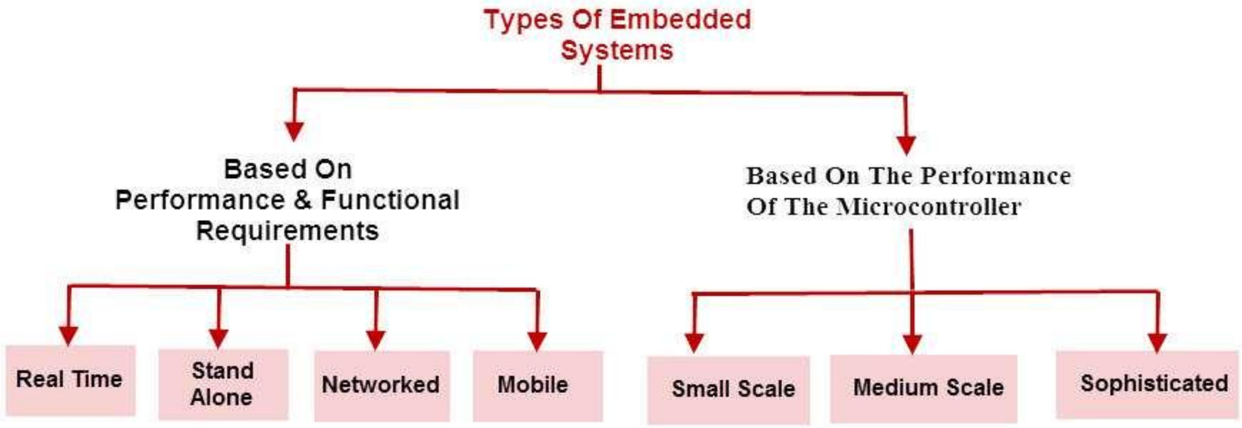
Program:

```
#include<reg51.h>
sbit led=P1^5; //make P1.5
as LED
void delay( unsigned int);
void main ()
{
led=1;
delay(50);
led=0;
while(1);
}
void delay (unsigned int
itime)
{
unsigned int x,y;
for(x=0; x<itime; x++)
for (y=0; y<1275; y++);
}
```

ii) State classification of Embedded system and describe any two types with example .

Ans:- (Classification- 3 mks, description of each -1/2 mks)

i. **Types of embedded system: 2M**


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

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 wherein all embedded devices are connected to a web server and accessed and controlled by a web browser. Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP

4. Mobile Embedded Systems

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

5. Small Scale Embedded Systems

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

6. **Medium Scale Embedded Systems**

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

7. Sophisticated Embedded Systems

These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors.

5. Attempt any Four of the following : 16

a. **Compare RISC and CISC.**

Ans:- (Four points-4 mks)

RISC	CISC
Reduced Instruction Set Computer	Complex Instruction Set Computer
Emphasis on software	Emphasis on hardware
Single clock instruction only	Includes multi clock instruction
Register to register, load and store are independent instructions	Memory to memory load and store incorporated instruction
Low cycles per second, large code sizes	Small code sizes, high cycles per second
Spends more transistors on memory registers	Transistors used for strong complex instructions.

b. **Write ‘c’ language program to check bit P1.2. If it is high send 55H to PO, otherwise send AAH to P2.**

Ans:- (Proper relevant program- 4 mks)

```
#include<regx51.h>

sbit data_bit = p1^2;                        // declaration of single bit of port 1

void main(void)

{

        data_bit = 1;                  // set bit P1.7 as an input

        while(1)

        {

                if(data_bit == 1)        // check P1.7 bit
```

```
            P0 = 0x55;              // if 1 then send 55H to port 0

            else

            P2 = 0xAA;              // if 0 then send AAH to port 2

      }

}
```

c. **Describe flowing wireless communication protocols:**
1) IrDA   2) WiFi

Ans:- (Each relevant discussion- 2 mks)

1) IrDA:

IrDA is a serial half duplex, line of sight based wireless technology for data communications between devices

The remote control of TV ,VCD players etc. works on infrared data communication principal.

Infrared communication technique uses infrared waves of the electromagnetic spectrum for transmitting the data.

Range from contact to at least 1 meter and can be extended to 2 meters.

Bi-directional point-to-point communication

.

Data transmission from 9600 b/s with primary speed/cost steps of 115 kb/s an maximum speed up to 16 Mb/s.

Data packets are protected using a CRC.

2) Wifi

Wi-Fi is a technology for wireless local area networking with devices based on the IEEE 802.11 standards. Wi-Fi is a trademark of the Wi-Fi Alliance, which restricts the use of the term Wi-Fi Certified to products that successfully complete interoperability certification testing.

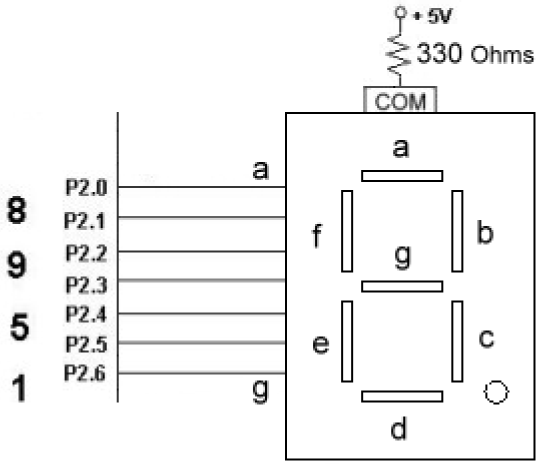
Devices that can use Wi-Fi technology include personal computers, video-game consoles, phones and tablets, digital cameras, smart TVs, digital audio players and modern printers. Wi-Fi compatible devices can connect to the Internet via a WLAN and a wireless access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can be as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points.

Wi-Fi most commonly uses the 2.4 gigahertz (12 cm) UHF and 5.8 gigahertz (5 cm) SHF ISM radio bands. Anyone within range with a wireless modem can attempt to access the network; because of this, Wi-Fi is more vulnerable to attack (called eavesdropping) than wired networks.

d. **Describe the features of I2C serial communication protocol.**
Ans: (Any four feature: 1 Mark each)

1 Independent Master, Slave, and Monitor functions.
2 Supports both Multi-master and Multi-master with Slave functions.
3 Multiple I 2C slave addresses supported in hardware.
4 One slave address can be selectively qualified with a bit mask or an address range in order to respond to multiple I 2C bus addresses.
5 10-bit addressing supported with software assist.
6 I2c operates in 3 speeds 100kbps, 400kbps and 3.4mbps

e. **Describe interfacing diagram of 7 segment LED display with microcontroller 8051.**
Ans: - (proper relevant diagram- 4 mks)

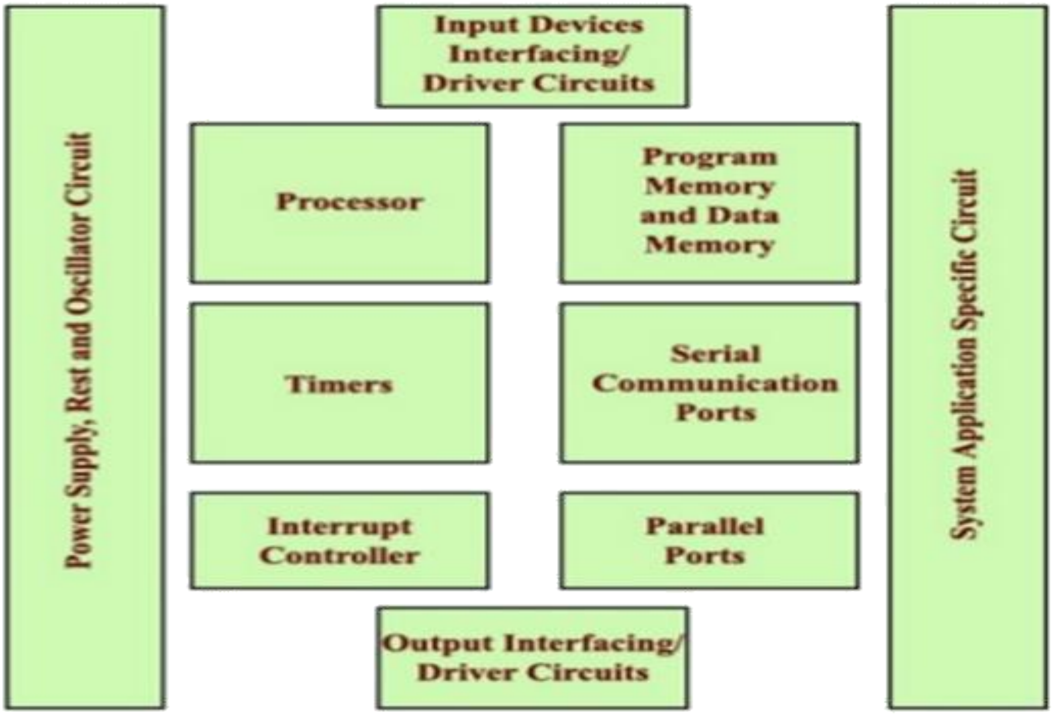


Other pin connections of 8051 like cystal, reset ,power pins as it is

6. Attempt any Four of the following : 16

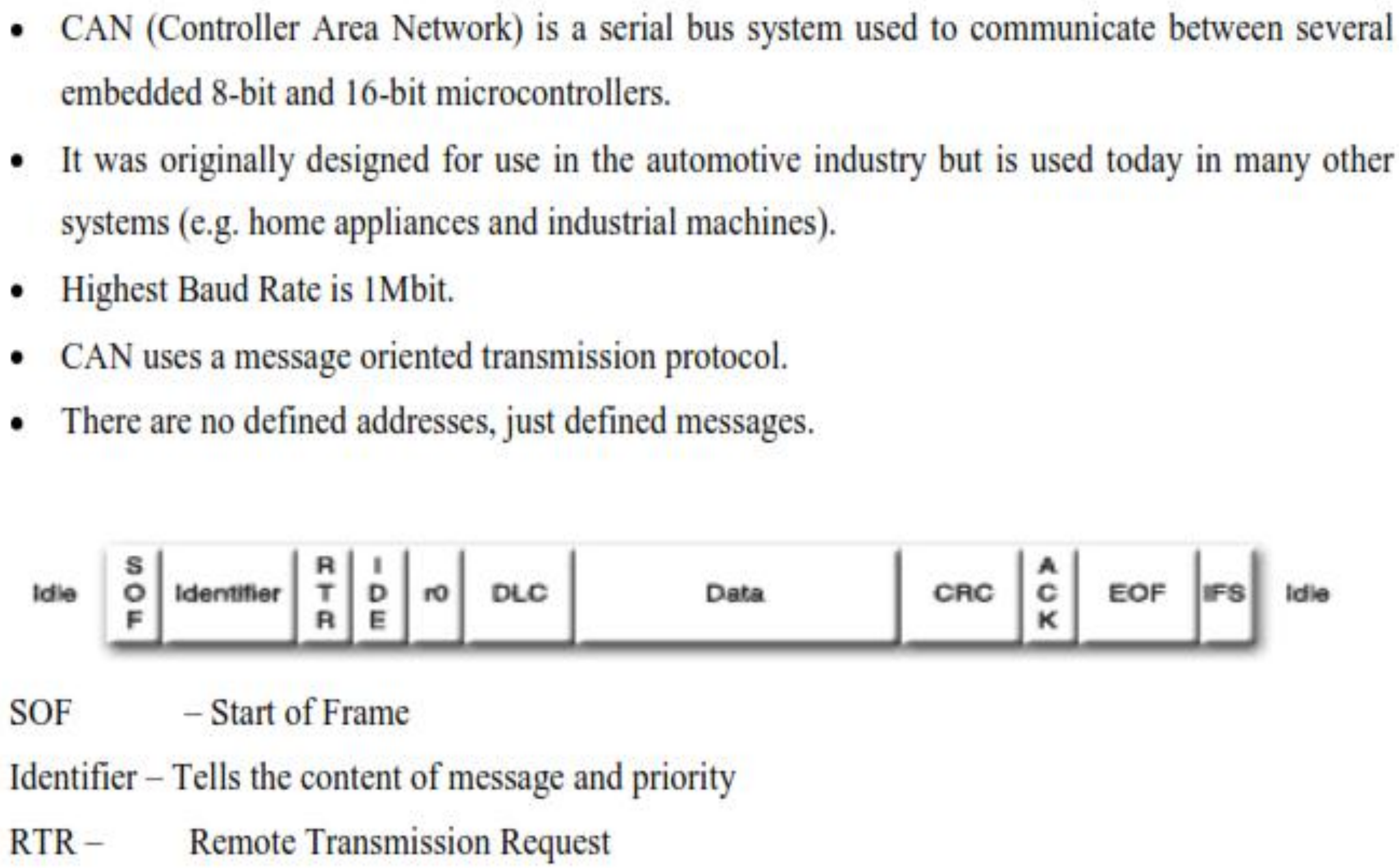
a. Draw block diagram of Embedded system.

Ans:- (Block diagram- 4 mks)


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

b) Describe the function of CAN bus protocol.

Ans:- (Relevant and proper description- 4 mks)



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

Idle	SOF	Identifier	RTR	IDE	r0	DLC	Data	CRC	ACK	EOF	IFS	Idle

SOF – Start of Frame

Identifier – Tells the content of message and priority

RTR – Remote Transmission Request

c) State any four specifications of RTOS.
Ans:- (any four specifications- 4 mks)

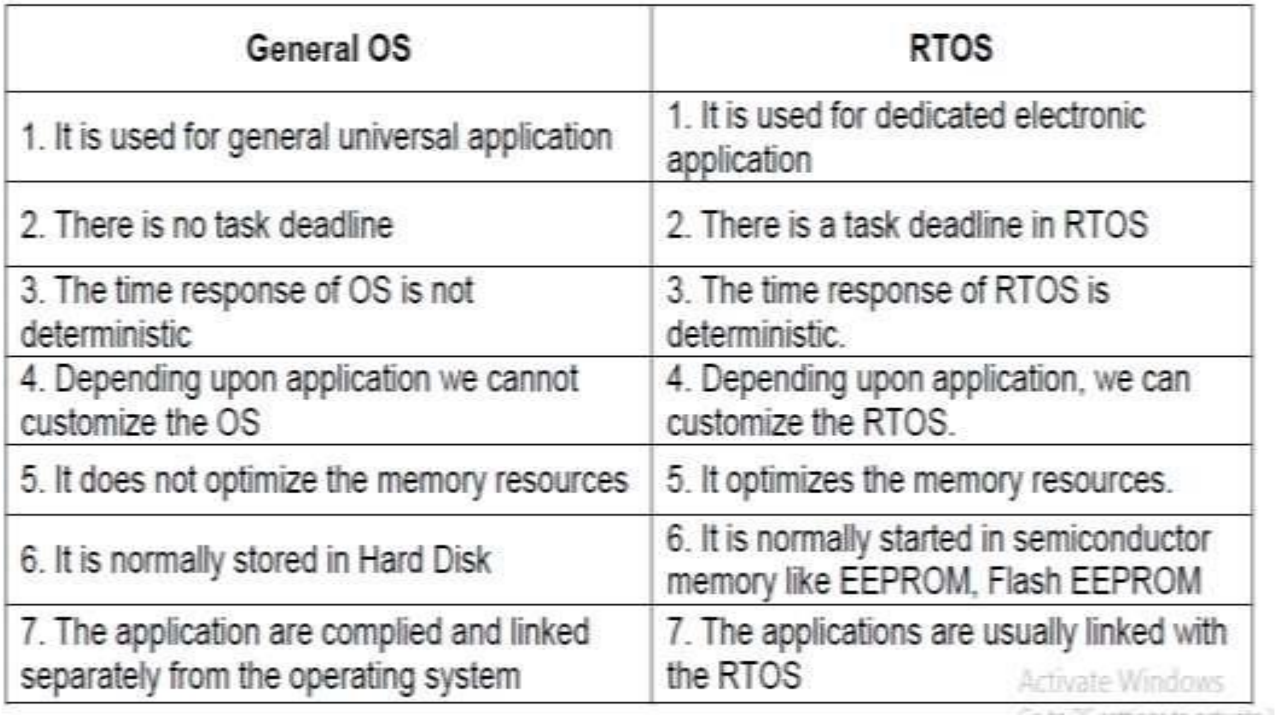
Specifications of RTOS:

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 cell phone, 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

d) Differentiate between general purpose operating system (GPOS) and real time operating system (RTOS).

Ans:- (Relevant four points- 4 mks)



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

e) Describe hard and soft real time operating system with example.

Ans:- (Each description- 2 mks)

A hard real-time system (also known as an immediate real-time system) is hardware or software that must operate within the confines of a stringent deadline. The application may be considered to have failed if it does not complete its function within the allotted time span.

The chief design goal is not high throughput, but rather a guarantee of a soft or hard performance category. An RTOS that can usually or generally meet a deadline is a soft real-time OS, but if it can meet a deadline deterministically it is a hard real-time OS

Hard time real operating system example – defence systems like missiles

Soft time real operating system example- audio and video systems