T.Y. Diploma : Sem. VI
[EJ/ET/EN/EX/IS/IC]
Embedded Systems

Time : 3 Hrs. Prelim Question Paper Solution **Marks : 100**

Q.1(a) Attempt any THREE questions. **[12]**

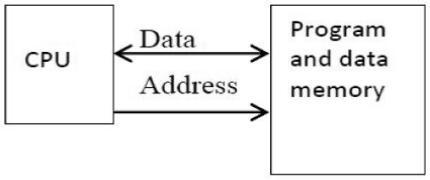
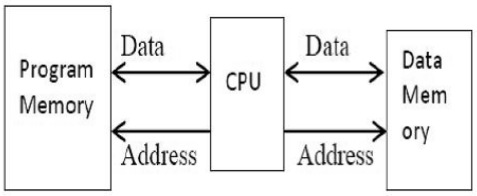
Q.1(a) (i) Compare assembly language and embedded C language. (Any Four) **[4]**

Ans.: [Any Four - 1 mark each]

	Parameter	Assembly	C
1.	Ease of writing Program	Complex	Easy
2.	Time for coding	More	Less
3.	Time for execution	Less	More
4.	Debugging	Tedious	Simple
5.	Updating	Tedious	Simple
6.	Hex file size	Small	More
7.	Memory required	Less	More
8.	Compatibility	Changes with MC	Portable

Q.1(a) (ii) Compare Von Neumann and Hardware architecture. (Any Four) **[4]**

Ans.: [Relevant 4 points – 4 marks]

	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.

Q.1(a) (iii) List the software development tools in an embedded system and state the function of compiler and debugger. **[4]**

Ans.: Software development tools: [List - 2 marks, explanation compiler & debugger – 1 mark each]

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

The function of compiler

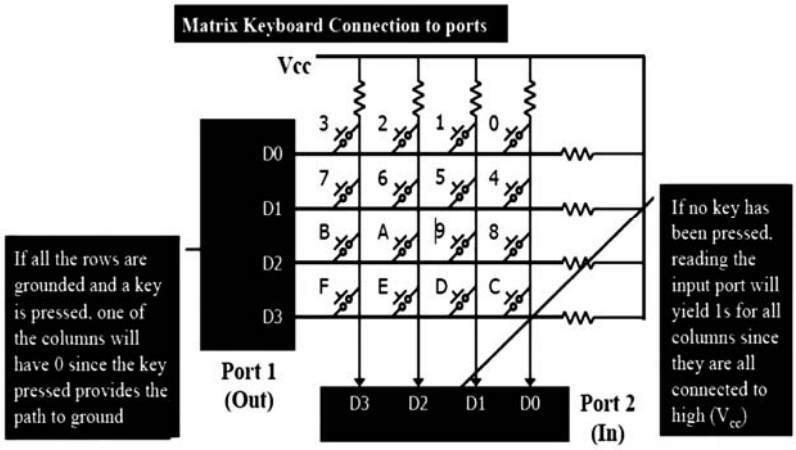
1) **Compiler:** It is a computer program that transforms the source code written in a programming or source language into another computer language i.e. target language i.e. binary code known as object code.

The function of debugger.

2) **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).

Q.1(a) (iv) Draw interfacing diagram of 4 × 4 matrix keyboard with 89C51 micro controller (No program). **[4]**

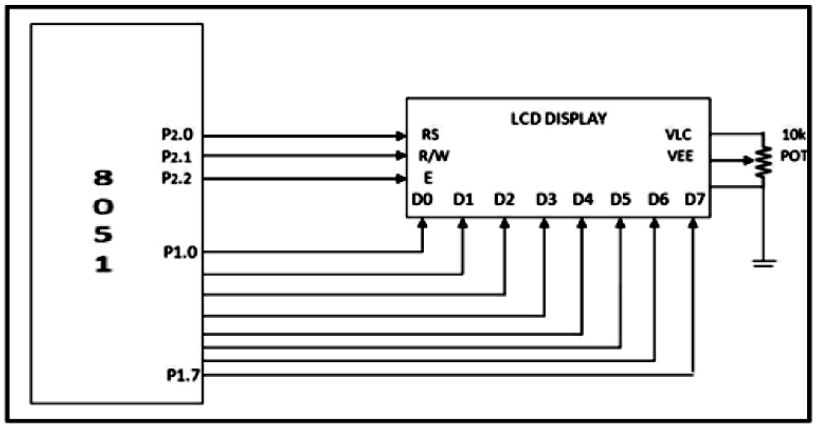
Ans.: [Correct labelled diagram – 4 marks]



Q.1(a) (v) Draw the interfacing diagram of 16 × 2 LCD display with 89C51 and state the function of **[4]**

(1) RS (2) EN (3) R/W

Ans.: [Interfacing diagram - 1 mark, function of each pin - 1 mark]



Function:

RS: RS is used to make the selection between data and command register.
RS = 0, command register is selected
RS = 1 data register is selected.

RW: R/W gives you the choice between writing and reading.
R/W = 1, reading is enabled.
R/W = 0 then writing is enabled.

EN: Enable pins are used by the LCD to latch information presented to its data pins. When data is supplied to data pins, a high to low pulse must be applied to this pin in-order for the LCD to latch in the data present at the data pins.

Q.1(b) Attempt any ONE questions. [6]

Q.1(b) (i) Describe any 6 design metrics of an embedded system. List any 3 applications of an embedded system. [6]

Ans.: Design metrics of an embedded system [Any Six - ½ marks each]

- The embedded system designer must of course construct an implementation that fulfills desired functionality, but a difficult challenge is to construct an implementation that fulfills desired functionality, but a difficult challenge is to construct an implementation that simultaneously optimizes numerous design metrics. For out purposes, an implementation consists of a software processor with an accompanying program, a connection of digital gates, or some combination thereof. A design metric is a measurable feature of a system's implementation. Common relevant metrics include:
- **Unit cost** : the monetary cost of manufacturing each copy of the system, excluding NRE cost.
- **NRE cost (Non-Recurring Engineering 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").
- **Size** : the physical space required by the system, often measured in bytes for software, and gates or transistors for hardware.
- **Performance** : the execution time or throughput of the system.
- **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.
- **Flexibility:** the ability to change the functionality of the system without incurring heavy NRE cost. Software is typically considered very flexible.
- **Time-to-market:** The amount of time required to design and manufacture the system to the point the system can be sold to customers.
- **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.
- **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.
- **Safety** : the probability that the system will not cause harm.

Applications [Any Three - 1 marks each]

- **Consumer electronics** : cell phones, pagers, digital cameras, videocassette recorders, portable video games, calculators, and personal digital assistants;
- **Home appliances** : microwave ovens, answering machines, thermostat, home security, washing machines, and lighting systems;
- **Office automation:** fax machines, copiers, printers, and scanners;
- **Business equipment** : cash registers, curbside check-in, alarm system, card readers, product scanners, and automated teller machines;
- **Automobiles** : transmission control, cruise control, fuel injection, anti-lock brakes, and active suspension.

Q.1(b) (ii) State different scheduling algorithms of RTOS and describe Round Robin scheduling algorithm. [6]

Ans: [Different methods - 3 marks, description of round robbin algorithm - 3 marks]

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.

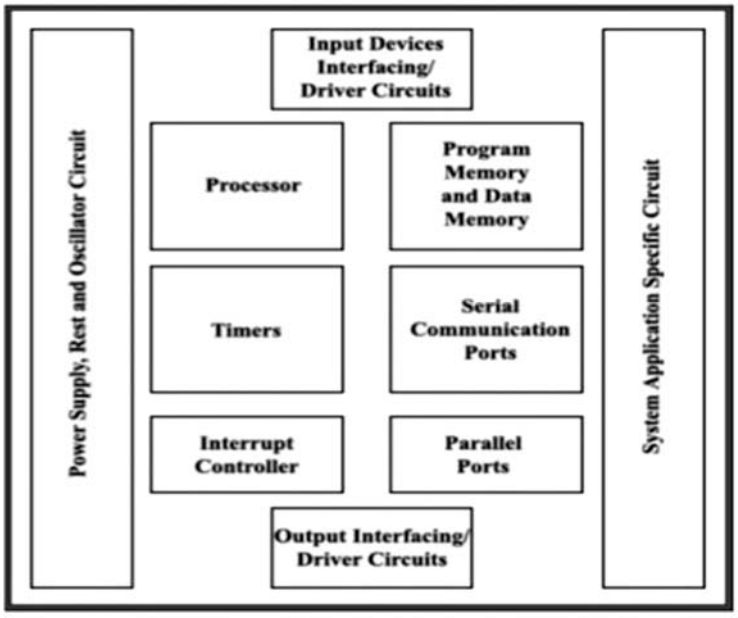
The kernel gives control to the next task if

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 multimeter, microwave oven has this algorithm .

Q.1(b) (iii) Draw and explain different hardware units of an embedded system. [6]

Ans.: [Draw - 2 marks, Explanation - ½ mark each]



(i) Embedded processor

It is the heart of the embedded system. It has two essential units control unit and execution unit. Control unit fetches instructions from memory and execution unit includes ALU and circuits to perform execution of the instructions for a program control taks.

(ii) Power supply, reset and oscillator circuit

- Most of the systems have their own power supply. Some embedded systems do not have their own power supply. These embedded systems are powered by external power supply e.g. USB based embedded system, network interface card, Graphics Accelerator etc. are powered by PC power suppply.

- Reset means that processor begins processing of instructions from starting address set by default in program counter on power up.
- The clock circuit controls execution time of instructions, CPU machine cycles.

(iii) Timers

Timer circuit is suitably configured as system clock or RTC (Real time clock). To schedule various tasks and for real time programming an RTC (Real Time Clock), or system clock is needed.

(iv) Program & data memory

In embedded system, secondary memory like disk is avoided. Most of the embedded processors have internal memory such as ROM, RAM, flash/EEPROM, EPROM/PROM for storing program and data.

(v) Interrupt controller

It is an interrupt handling mechanism which must exist in embedded system to handle interrupts from various processes and for handling multiple interrupts simultaneously pending for service.

(vi) I/O ports

I/O ports are used to interface external devices like sensors, key buttons, transducers, LEDs, LCD actuators, alarms, motors, values, printer etc. There are two types of ports, parallel and serial port. The parallel ports are used in short distance communication while serial ports are used in long distance communication.

(vii) Input & output device interfacing/driver circuits

Some I/O devices like motors, actuators, valves, sensors are not compatible with the processor. Hence the I/O interface circuits are designed to drive such input and output devices interfaced to the embedded processor.

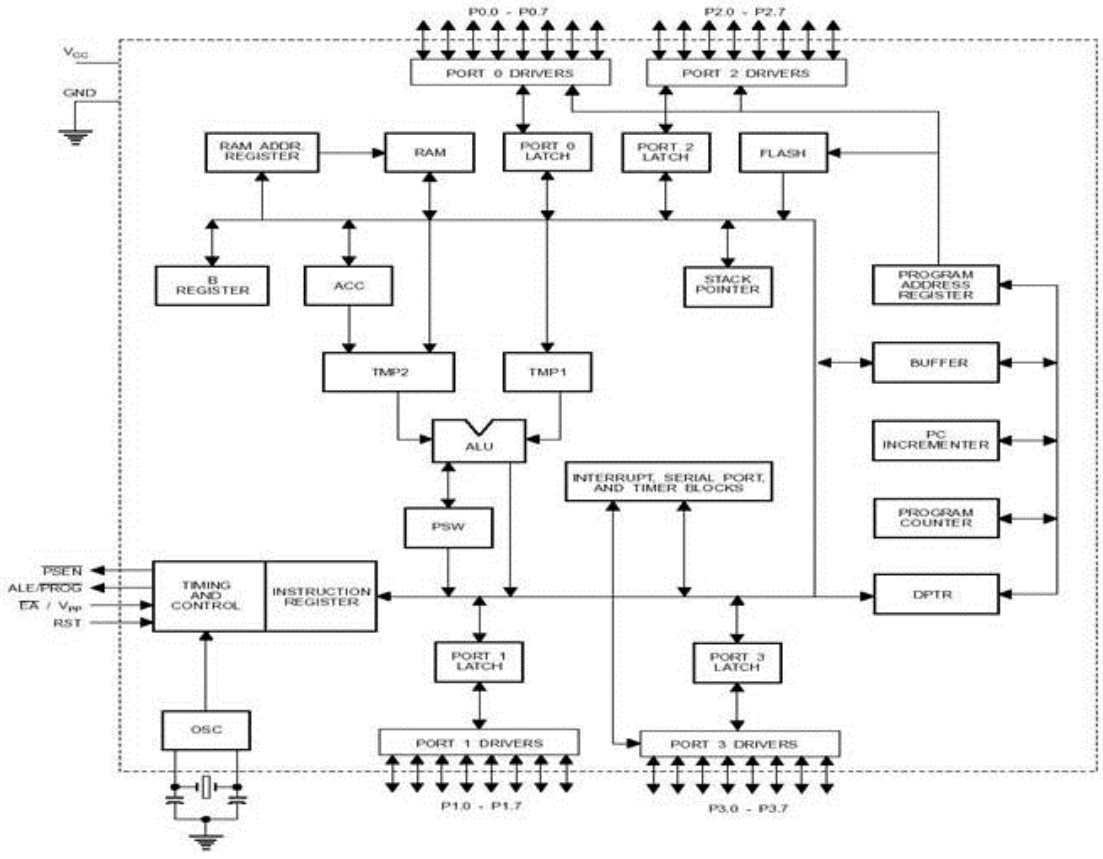
(viii) System Application specific circuits

These are the circuits that can control specific target circuits. They consist of ADC, DAC, relays, sensors etc.

Q.2 Attempt any FOUR questions. [16]

Q.2(a) Draw the architecture of 89C51. [4]

Ans.: Block Diagram [4 marks]



Q.2(b) Define the semaphore and deadlock. **[4]**

Ans.: Semaphore : [2 marks]

- RTOS kernels provide a semaphore to ensure the integrity of data during sharing the common data.
- Resource count and a wait queue is associated with semaphore where the resource count shows availability of resource and the wait queue manages the tasks waiting for resources from the semaphore.
- A semaphore functions as a manager which decides whether a task has the access to the 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.
- The 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.

Deadlock : [2 marks]

- A deadlock is a situation in which two or more tasks or processes waiting for each other to release a resource, or more than two processes are waiting for resources in a circular chain and neither request can be satisfied.
- Deadlock is a common problem in multiprocessing and multitasking where numbers of tasks or processes share a specific type of mutually exclusive resource.
- Now a day, the computers or embedded systems designed for the time-sharing or real time are equipped with a hardware lock which provides exclusive access to processes, forcing serialized access.
- There is no general solution to avoid deadlocks.

Q.2(c) State any two features of IDE and ICE. **[4]**

Ans.: Features of IDE : [Two features of each- 2 marks]

(i) It supports for defining a processor family and its version.
(ii) Supports a user definable assembler to support a new version.
(iii) Provides multiuser environment.
(iv) Supports conditional and unconditional break points.
(v) Provides debugger.

Features of ICE

(i) It is hardware device to debug the software of embedded system.
(ii) Provides a window into the embedded system.
(iii) It emulates the CPU of the embedded systems computer.

Q.2(d) Write 89C51 'C' language program to toggle all bits of port P2 continuously with 500 ms delay. **[4]**

Ans.: [Correct program - 4 marks]

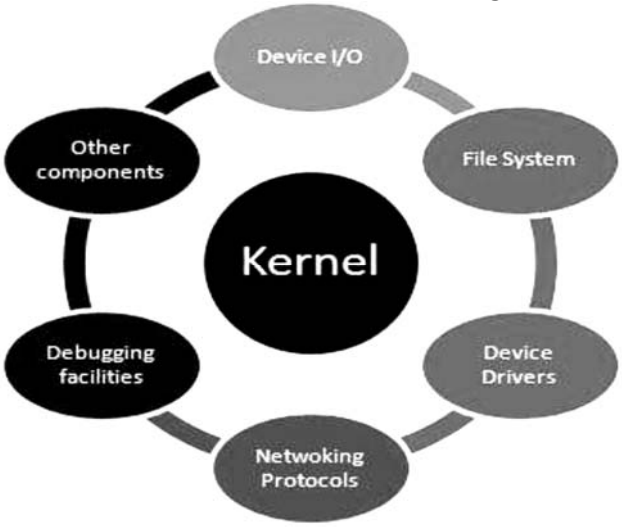
```
#include <reg51.h>
void delay (unsigned int);
void main (void)
{
while(1) //repeat loop
{
P2=0xff; //toggle all bits of port2
delay (500); //add delay
P2=0x00; //toggle all bits of port2
delay (500); //add delay
}
}
void delay (unsigned int i)
```

```
{
Unsigned int x, y;
for(x=0; x< i; x++)
for (y=0; y<1275; y++);
}
```

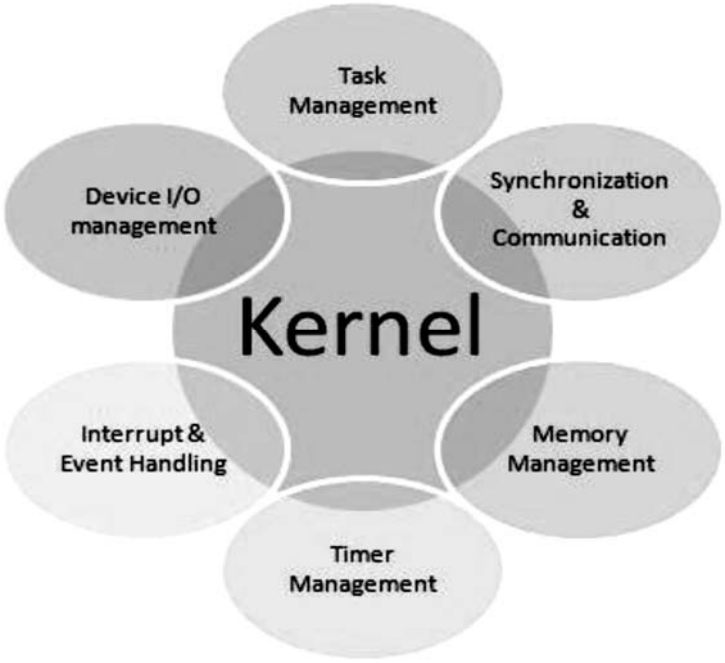
Q.2(e) Draw and describe architecture of RTOS. **[4]**

Ans.: [Diagram- 2 marks, Description- 2 marks]

- 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:

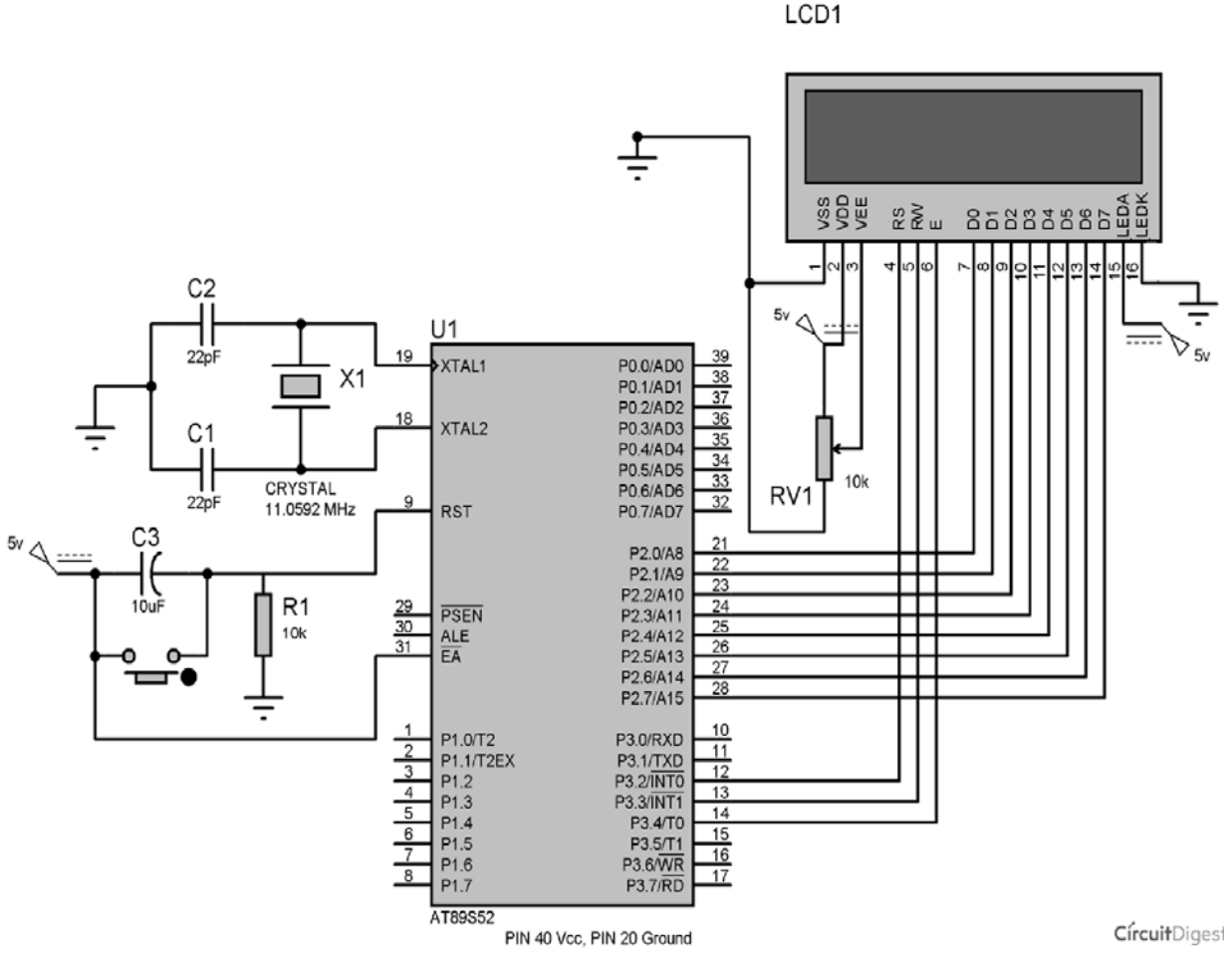


- An operating system generally consists of two ports: kernel space and user space.
- RTOS kernel acts as an observation layer between the hardware and the applications.
- Kernel is the smallest and central component of an operating system. Its services include managing memory and devices and to provide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



Q.2(f) Draw interfacing diagram of LCD with microcontroller 89C51. **[4]**

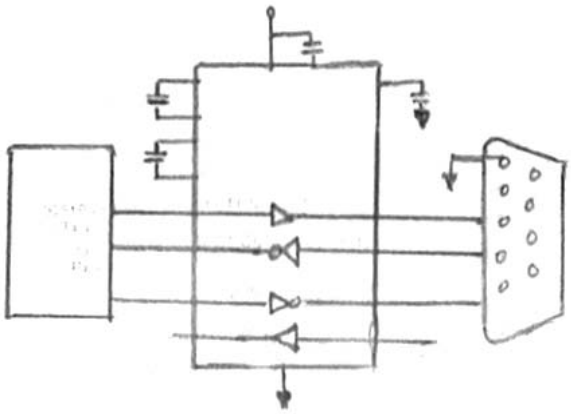
Ans.: [Proper relevant diagram- 4 marks]



Q.3 Attempt any FOUR questions. **[16]**

Q.3(a) Draw and describe the RS 232 interface with 8051 using Max232C. **[4]**

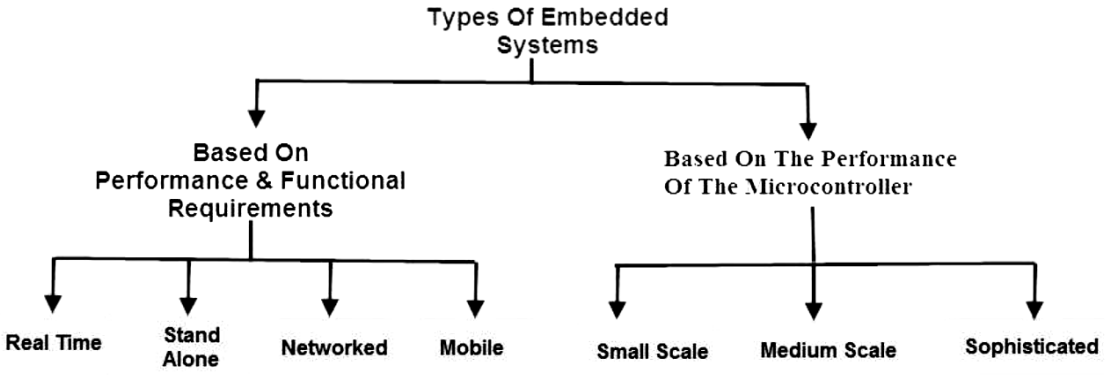
Ans.: [Diagram- 2 marks, Description- 2 marks]



The above figure shows the interface between microcontroller and RS 232. It uses max 232 line driver/voltage converters for RS 232 signal to TTL voltage level acceptable to 8051. Max 232 works on single power supply i.e. +5V. It has two sets of line driver. It uses 4 capacitors ranging from 1 microfarad to 220 microfarad.

Q.3(b) State classification of Embedded system and describe any one types with example. **[4]**

Ans.: [Classification- 3 marks, description with example any one - 1 mark]



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.

Q.3(c) Write 89C51 'C' Program to transfer the message "MSBTE" serially at 4800 baud rate continuously. Use 8 bit data and 1 stop bit. **[4]**

Ans.: [4 marks]

```
#include <stdio.h>
#include <reg51.h>
void sertext (unsigned char) ;
void main( )
 {
   TMOD = 0x20 ;
   SCON = 0x50 ;
   TH1=0xFA ;
   TR1 = 1 ;
sertext ('M') ;
sertext ('S') ;
sertext ('B') ;
sertext ('T') ;
sertext ('E') ;
   while (1)
    {
    }
 }

void sertext (unsigned char x)
  {
    SBUF = x ;
    while (TI = = 0) ;
     {
       TI = 0 ;
     }
  }
```

Q.3(d) Write C language program to rotate stepper motor by 90 degree clockwise. Assume step angle is 1.8 degree and 4 step sequence. **[4]**

Ans.: Step angle=1.8 degree [4 marks]

Total steps required=90/1.8=50 Four step sequence so

Count=50/4=12.5=approximately 13

```
#include<regx51.h>
  void delay();
  void main()
  {
  int x;
  for (x=0;x,13;x++)
  {
  P0=0x33;
  delay(10);
  P0=0x66;
  delay(10);
  P0=0xcc
  delay(10);
  P0=0x99;
  delay(10);
  }
  }
  void delay(unsigned int t)
```

```
{
unsigned int x,y;
for(x=0;x<=t;x++)
for(y=0;y<=1275;y++);
}
```

Q.3(e) Differentiate between CAN with I2C protocols with respective to [4]
(1) Data transfer rate (2) Number of fields
(3) Addressing bit (4) Application

Ans.: [1 mark each]

	Parameter	CAN	I2C
1.	Data Transfer Rate	2Mbps	100 kbps, 400kbps
2.	Number of field	8	7
3.	Address bit	11 (or 12)	7
4.	Application	Automobile	Interconnection of IC, Serial bus

Q.3(f) List the Parallel Communication Protocol and describe any one. [4]

Ans.: (i) Industry Standard Architecture (ISA) [List - 1 mark each, Description (any one) - 1 mark]
(ii) Peripheral Component Interface (PCI)
(iii) PCI – X

- **PCI:** PCI stands for Peripheral Component Interconnect/Interface bus.
- It is popular for higher bandwidth processor independent which can function as peripheral bus.
- It is introduced by Intel in 90's.
- It is 32 bit local bus and extended up to 64 bit by processor it requires.
- It has high speed I/O subsystem performance.
- The PCI is designed to meet economically the i/o requirement of modern system.
- It supports ten i/o devices and provides 3 types of synchronous parallel interface.
- It has two versions:
  (i) 32 bit (33 MHz) (ii) 64 bit (66 MHz)
- The data transfer rate for synchronous is 132 mbps and for asynchronous it is 528 mbps.
- The PCI driver can access hardware automatically or by programmer can assign address.
- The automatic detection and assignment of addresses of various devices simplifies the addition and removal of the system peripheral.
- PCI is designed to support variety of microprocessor best configuration including single and multi processing system.

OR

PCI-X

- It is an extension of PCI bus and supports 64 bit, 100MHz transfer.
- PCI-X is revised to double the maximum clock speed to improve the data exchange transfer between processor and peripherals.
- The data exchange rate is 1.06 gbps.

Q.4(a) Attempt any THREE questions. [12]

Q.4(a) (i) Find the content of Accumulator after execution of the following code [4]
(1) ACC = 0 × 94 >> 5; (2) ACC = 0 × 5A << 2

Ans.: (1) ACC = 0 × 94 >> 5 [2 marks]
ACC = 0 × 04

(2) ACC = 0 × 5A << 2 [2 marks]
ACC = 0 × 68

Q.4(a) (ii) Write any four features of RTOS. [4]

Ans.: Features of RTOS [Any Four - 1 mark each]

- **Reliability**
  A reliable system is the system which is available all the time. A system that does not fail is a reliable system. RTOS is reliable but it does not guarantee the reliability of an embedded system. The reliability of an ES depends on the hardware, the application code.
- **Predictability**
  RTOS have predictable behavior in which the completion of OS calls occurs within known time frame.
- **Performance**
  The system must perform fast enough to fulfil the timing requirements. The performance of RTOS can be measured on a call-by-call basis. Time stamps are produced when a system call starts and when it completes. This method of analyzing is useful in designing stage but the true performance is measured as whole.
- **Compactness**
  RTOS used in embedded system are extremely constraint as far as resources are concerned. The design requirements limit the system memory which limits the size of the application and the operating system.
- **Scalability**
  RTOS are most used in variety of embedded system they must be able to scale up/down to suit the application.

Q.4(a) (iii) State any four features of USB serial communication protocol. [4]

Ans.: (1) Multiple device connection [Any Four relevant features - 1 mark each]
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.

Q.4(a) (iv) Differentiate RTOS with desktop operating system. (Any four points). [4]

Ans.: [Any Four - 1 mark each]

	RTOS	Desktop OS
1.	Predictable schedule	Unpredictable but efficient schedule
2.	More deterministic	Less deterministic
3.	Ability to scale up and down to meeting application needs (scalable)	Less scalability
4.	Fast context switching	Higher context switch latency

5.	Uses priority preemptive scheduling	Uses round robin way of scheduling
6.	Support for diskless embedded systems by allowing executable to boot and run from RAM or RAM	Has to be booted from disk
7.	Reduced memory requirements	More memory requirements
8.	Faster performance	Slower

Q.4(b) Attempt any ONE questions. **[6]**

Q.4(b) (i) Draw the interfacing of key and LED to 89C51 microcontroller pins P1.0 and P2.0 respectively. Write C language program to read the status of key and display it on LED. **[6]**
(Key open = LED OFF and key closed = LED ON)

Ans.: Interfacing Diagram [3 marks]

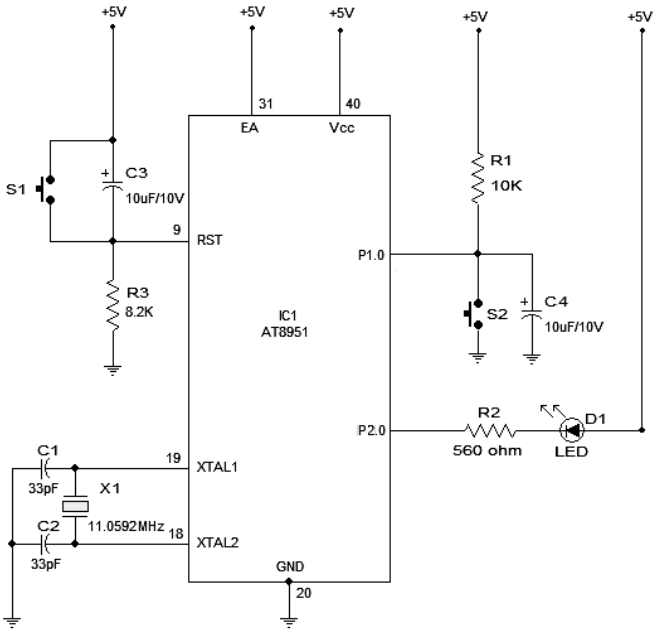


Fig.: Interfacing of LED and Switch with 8051/89c51

***C* Program** [3 marks]

```
#include<reg51.h>              /* special function register declarations */

sbit LED_pin = P2^0;           //Defining LED PIN
sbit switch_pin = P1^0;        //Defining Switch PIN
void Delay(int);               //Function prototype declaration

void main (void)               //Main Function
{
    switch_pin = 1;            //Making Switch PIN input
    LED_pin=1;                 //LED off initially

    while(1)                   //infinite loop
    {
        if(switch_pin == 0 )       //If switch pressed
        {
            LED_pin = 0;       //LED ON by making port pin Low
            Delay(2000);       //Delay for 2sec.
            LED_pin = 1;       //LED OFF by making port pin High
        }                  //End of  if
    }                  //End of while
```

```
}                   //End of Main

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

Q.4(b) (ii) Write C language program to generate a square wave of 2 KHz frequency on P1.1 pin by using timer 0 and mode 1. Assume XTAL frequency is 11.0592 MHz. [6]

Ans.: Crystal frequency= 11.0592 MHz [1 mark]

I/P clock = 11.0592 X 106 = 11.0592MHz

1/12 x 11.0592Mhz = 921.6 Khz

Tin = 1.085μ sec

For 2 kHz square wave [1 mark]

Fout = 2 KHz Tout = 1/2 X 103

Tout = 500μ sec

Consider half of it = Tout = 250μ sec

N = Tout / Tin = 250/1.085= 230

65536-230= (65306p10) = FF1A

Program [4 marks]

```
#include<reg51.h>
void delay(void);
sbit p=P1^5;
void main (void)
{
while (1)
{
    p=~p;
    delay();
}
}
void delay()
{
TMOD=0X01;      //set timer 0 in mode 1 i.e. 16 bit number
TL0=0X1AH;      //load TL register with LSB of count
TH0=0XFFH ;     //Load TH register with MSB of count
TR0=1           //start timer 0
While(TF0==0)   //wait until timer rolls over
TR0=0;          //Stop timer 0
TF0=0;          //Clear timer flag 0
}
```

Q.5 Attempt any FOUR questions. **[16]**

Q.5(a) How the assembly language instruction is used in C language Program. **[4]**

Ans.: Assembly language instruction in c : [Explanation with example - 4 mark]

```
#include<reg51.h>
void main()
{
    int a = 3, b = 3, c;

    asm {
        mov ax,a
        mov bx,a
        add ax,bx
        mov c,ax
    }
}
```

It is used by using asm in { } brackets Or writing asm before each instruction as shown below :

```
#include<reg51.h>
void main()
{
    int a = 3, b = 3, c;

    asm mov ax,a
    asm mov bx,a
    asm add ax,bx
    asm mov c,ax
}
```

Q.5(b) Describe hard and soft real time operating system with example. **[4]**

Ans.: Hard and soft real time operating system [Each description - 2 marks]

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

Q.5(c) Describe following wireless communication protocols: **[4]**

(i) IrDA **(ii) WiFi**

Ans.: (i) IrDA : [Each relevant discussion - 2 marks]

- 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.

(ii) 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.

Q.5(d) Write 'C' language program to check bit P1.2. If it is high send 55 H to PO, otherwise send AAH to P2. **[4]**

Ans.: [Proper relevant program - 4 marks]

```
#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
    }
}
```

Q.5(e) Write 'C' language program to mask the lower 4 bits of port P_0 and upper 4 bits of port P_2 using logical operator. **[4]**

Ans.: [2 marks for each correct syntax]

```
#include <stdio.h>
#include <reg51.h>
void main( )
{
    P0 = P0 &0xF0;
    P2=P2&0x0F;
    While(1);
}
```

Q.5(f) Define embedded system. List any two advantages and disadvantages of embedded system. **[4]**

Ans.: **Definition :** [2 marks]

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

Advantages (any two): [1 mark]

- **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.
- **Cost :** The streamline make-up of most embedded system allows their parts to be smaller less expensive to produce.
- **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.
- **Maintenance :** Embedded systems are easier to maintain because the supplied power is embedded in the system and does not required remote maintenance.
- **Redundancies :** Embedded system does not involve the redundant programming.

Disadvantages (any two) : [1 mark]

- **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.
- **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.
- **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".
- **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.

Q.6 Attempt any FOUR questions. **[16]**

Q.6(a) List Date types used in 'C' with their values. **[4]**

Ans.: [Any Four - 1 mark each]

	Data Type	Size in bits	Data Range/Usage
1.	Unsigned Char	8	0 to 255
2.	Signed Char	8	–128 to +127
3.	Unsigned int	16	0 to 65536
4.	Signed int	16	–32768 to +32767
5.	Float	32	+1.175494E–18 to 3.402823E+38
6.	bit	1	0 to 1 bit addressable RAM
7.	sbit	1	SFR bit addressable
8.	sfr	8	RAM addresses 80 to FF only
9.	sfr16	16	0 to 65536 (16 bit SFR only)

Q.6(b) Describe parallel protocols PCI, PCI-X. **[4]**

Ans.: PCI : [2 marks for PCI and 2 marks for PCI-X]

- PCI stands for Peripheral Component Interconnect/Interface bus.
- It is popular for higher bandwidth processor independent which can function as peripheral bus. It is introduced by Intel in 90's.
- It is 32 bit local bus and extended up to 64 bit by processor it requires.

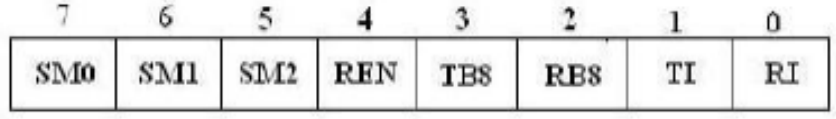
- It has high speed I/O subsystem performance.
- The PCI is designed to meet economically the i/o requirement of modern system.
- It supports ten I/O devices and provides 3 types of synchronous parallel interface.
- It has two versions : 32 bit (33 MHz), 64 bit (66 MHz)
- The data transfer rate for synchronous is 132 mbps and for asynchronous it is 528 mbps.
- The PCI driver can access hardware automatically or by programmer can assign address.
- The automatic detection and assignment of addresses of various devices simplifies the addition and removal of the system peripheral.
- PCI is designed to support variety of microprocessor best configuration including single and multi-processing system.

PCI -X :

- It is an extension of PCI bus and supports 64 bit, 100MHz transfer.
- PCI – X is revised to double the maximum clock speed to improve the data exchange transfer between processor and peripherals.
- The data exchange rate is 1.06 gbps.

Q.6(c) Draw the format of SCON register and explain all the bits. [4]

Ans.: SCON Register format: [Sketch - 1 mark, Explanation - 3 marks]



Function

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

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

SM2 : bit is used for multiprocessor communication. Set or cleared by the program to enable multiprocessor communications in mode 2 and 3. When set to 1 an interrupt is generated if bit 9 of the received data is a 1; no interrupt is generated if bit 9 is 0.

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

TB8 : Transmitted bit 8.

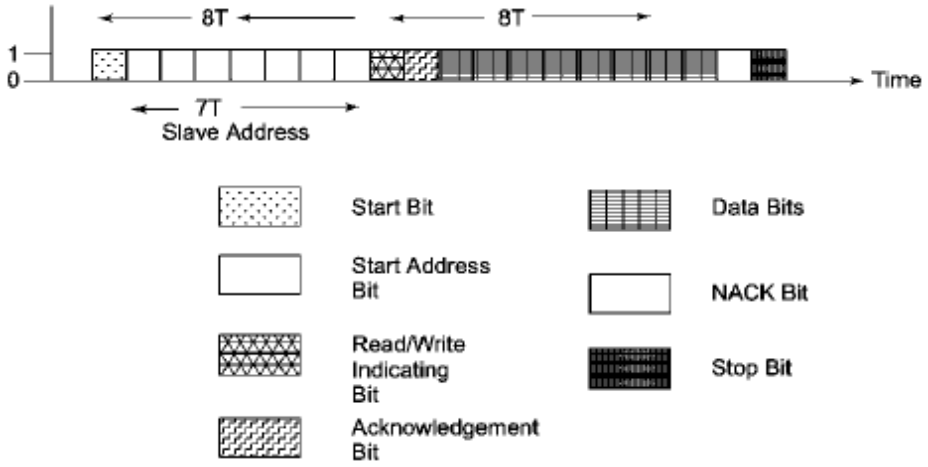
RB8 : Received bit 8.

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

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

Q.6(d) Draw the frame format of I^2C and explain each field in brief. [4]

Ans.: [Sketch - 2 marks, Explanation - 2 marks]



- First field of 1 bit – START bit = one.
- Second field of 7 bits – address field. It defines the slave address, which is being sent the data frame(of many bytes) by the master.
- Third field of 1 control bit – defines whether a read or write cycle is in progress.
- Fourth field of 1 control bit – defines whether is the present data is an acknowledgment (from slave).
- Fifth field of 8 bits – I2C device data byte (transmitted by either master or slave).
- Sixth field of 1 bit – bit NACK (negative acknowledgment) from the receiver (when master is receiving). If active the acknowledgment is expected from the slave.
- Seventh field of 1 bit – stop bit send by master.

Q.6(e) Explain inter–task communication with reference to RTOS. [4]

Ans.: Inter task communication : [Explanation - 4 marks]

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

(i) Message queue (ii) Pipes (iii) Remote procedure call

Inter task communications is executed using following mechanisms.

Message queues :

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

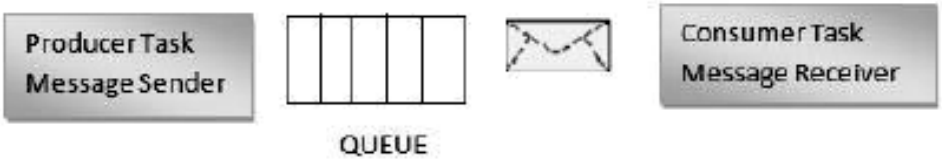
The queue may follow:

(i) First In First Out (FIFO)

(ii) Last In First Out (LIFO) or

(iii) Priority (PRI) sequence.

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



Pipes

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

Remote procedure call (RPC)

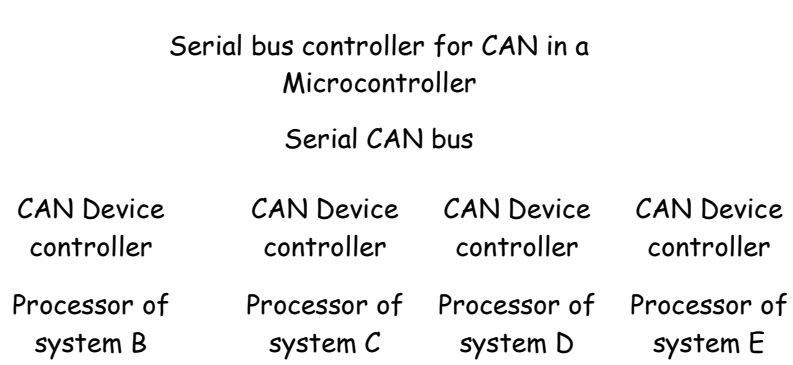
It permits distributed computing where task can invoke the execution of another task on a remote computer.

Q.6(e) Describe the CAN bus protocol with neat diagram. [4]

Ans.: CAN is a serial bus for interconnecting a central network [Description - 3 marks]

- It is used in automobiles where number of devices and controllers are located and distributed in it e.g.: breaks, engine, power, lamp, temperature controller, AC, gate, dash board, display, cruise control, etc.
- It is also used in medical and industrial plant.
- CAN bus is a standard bus in distributed network.
- CAN is implemented with the help of microcontroller.
- It is bidirectional serial line which is operated at 1 mbps.
- It employs a twisted pair connection of 120Ω line impedance at each controller port.
- Length of pair is 40 meter.
- CAN serial line is pulled to logic 1 (+4.5V to 12.3V) through pull up register.
- In ideal state it's in logic 1 also called as recessive state.
- Each node has buffer gate between i/p and CAN serial line.
- A node gets i/p at any instant from the line after sensing that instant line.
- It is pulled down to logic 0 called as dominant state.
- Each node has a current driver between o/p pin and serial line node sends the data bit as a data frame and data frame start through logic 1.
- CAN has a field for bus arbitration.
- Control bits for address and data length, data bits, CRC checkers, acknowledgement and ending bits.
- A CAN bus line available between CAN controller and host node.
- The bus arbitration methods are CSMA (Carrier Sense Multiple Access) and AMP (Arbitration on Message Priority).

[Diagram - 1 mark]



❑ ❑ ❑ ❑ ❑