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.1

a) Attempt any THREE of the following **12 M**

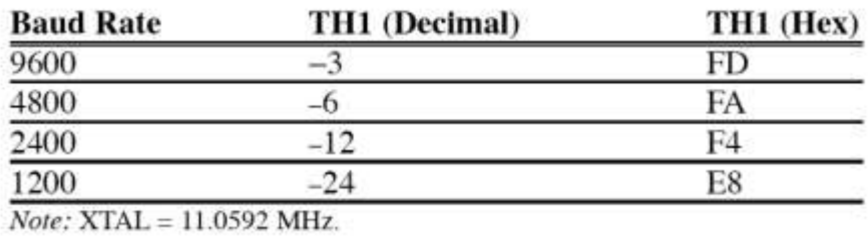
i) List various SFRs needed for serial communication using microcontroller 89C51 . Also list various baud rate for serial communication.

Ans:

(List 1 Marks, Baud rate 3 Marks)

List : SCON , PCON , SBUF **1M**

Standard baud rate : **3M**



Baud Rate	TH1 (Decimal)	TH1 (Hex)
9600	–3	FD
4800	–6	FA
2400	–12	F4
1200	–24	E8

Note: XTAL = 11.0592 MHz.

ii) List various software development tools available in IDE . Explain any one in brief.

Ans: (List: 2Marks, Any one tool Explanation: 2Marks)

Software development tools: **2M**

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

Explanation :(Any one tool)

Compiler: **2M**

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

Cross assembler:

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

Cross compiler:

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

Linker/Locator:

It is used for relocation process.

It is done during compilation also it can be done at run time by a relocating loader.

It is a program that takes one or more objects generated by compiler and combines them into a single executable program.

Simulators:

A simulator is the s/w that simulates an h/w unit like emulator, peripheral, network and I/O devices on a PC

- It defines a processor or processing device as well as various versions for the target system
- Monitors the detailed information of as source code part with labels and symbols during the execution for each single step.
- Provides the detailed information of the status of memory RAM and simulated ports, simulated peripheral devices of the defined target system

Integrated Development Environment (IDE):-

- It supports for defining a processor family and its version
- Support a user definable assembler to support a new version or a type of processor.
- Provides multiuser environment
- Supports conditional and unconditional break points
- Provide Debugger.

iii) List the features of I2C bus.

Ans:

(Any four feature: 1 Mark each)

Features of I2C Bus:

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

iv) Draw labeling diagram to interface 16 X 2 LCD with 89c51 .state the function of pins

1) RS

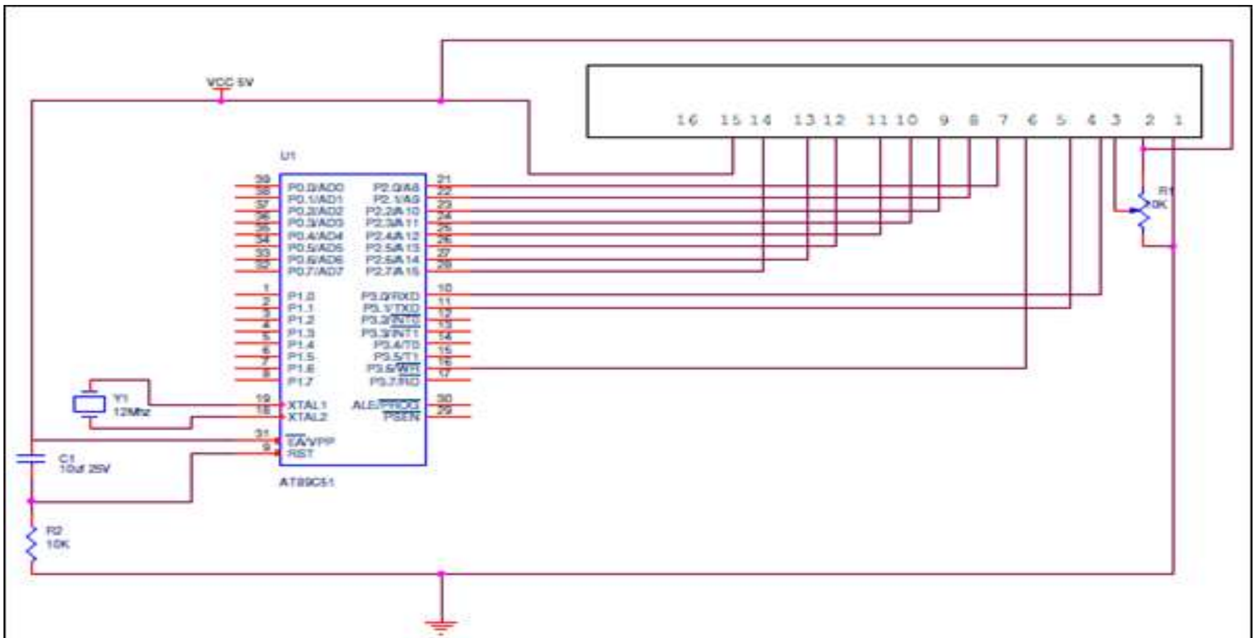
2) R/W

3) EN

Ans:

(Diagram: 2 labeling: 1/2Marks, ½ Mark each function)

16 X 2 LCD with 89c51 Diagram: $2^{1/2}$



Circuit Diagram of Interfacing LCD to AT89C51

Function:

$1^{1/2}$

1) RS: RS is the register select pin used to write display data to the LCD (characters), this pin has to be high when writing the data to the LCD. During the initializing sequence and other commands this pin should low.

2)R/W: Reading and writing data to the LCD for reading the data R/W pin should be high (R/W=1) to write the data to LCD R/W pin should be low (R/W=0)

3)EN: Enable pin is for starting or enabling the module. A high to low pulse of about 450ns pulse is given to this pin.

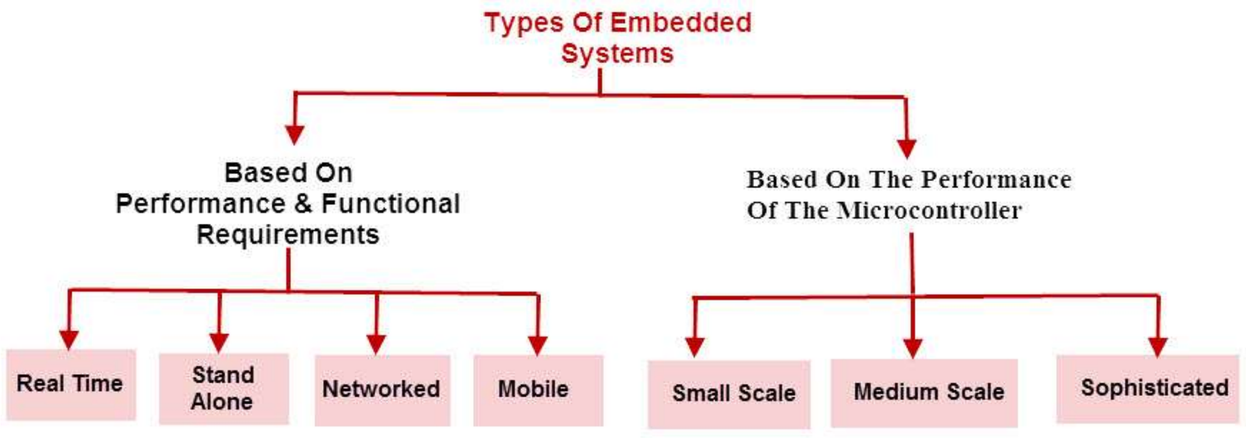
b) Attempt any ONE of the following **6M**

i) State various types of embedded system. Explain any two in brief .state four applications of embedded system.

Ans:

(Types: 2 Marks, Explanation any two: 1 Mark Each, four application: ½ Mark Each)

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. They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

Applications: (any four of the following) **2M**

1. Applications of Embedded Systems:

Embedded systems are used in different applications like automobiles, telecommunications, smart cards, missiles, satellites, computer networking and digital consumer electronics.

2. Embedded Systems in Automobiles and in telecommunications

- Motor and cruise control system
- Body or Engine safety
- Entertainment and multimedia in car
- E-Com and Mobile access
- Robotics in assembly line
- Wireless communication
- Mobile computing and networking

3. Embedded Systems in Smart Cards, Missiles and Satellites

- Security systems
- Telephone and banking
- Defense and aerospace
- Communication

4. Embedded Systems in Peripherals & Computer Networking

- Displays and Monitors
- Networking Systems
- Image Processing
- Network cards and printers

5 Embedded Systems in Consumer Electronics

- Digital Cameras
- Set top Boxes
- High Definition TVs
- DVDs

ii) State various task scheduling algorithm in RTOS. Explain any one in brief .

Ans:

(**State tasks: 2 Marks, Explanation: (4 marks)**

2M

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

Explanation: (any one of the following) **4M**

1. First in first out:

In first in first out scheduling algorithm the task which are ready to run are kept in queue ND the cpu serves the task on first in first served basis. this scheduling algorithm is shown in fig. is very simple to implement but not well suited for most applications because it is difficult to estimate the amount of time a task has to wait for being executed . However this is good algorithm for an embedded system has to perform few small tasks all with small execution time. If there is no time critically and the number of tasks is small, this algorithm can be implemented.

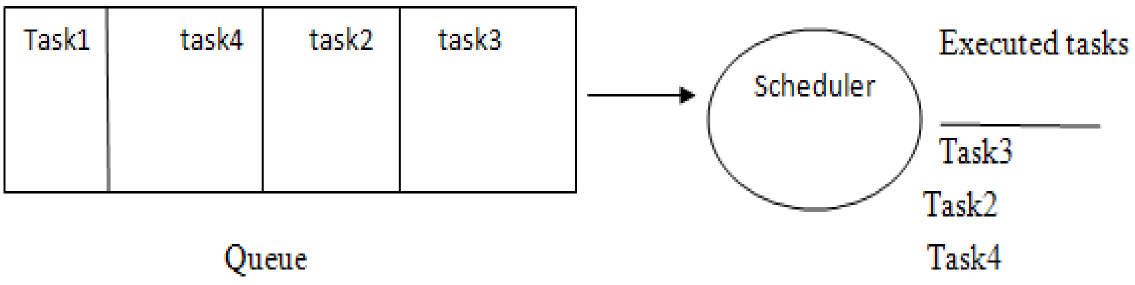


Fig Scheduling algorithm : FIFO

2. 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 millimeter , microwave oven has this algorithm

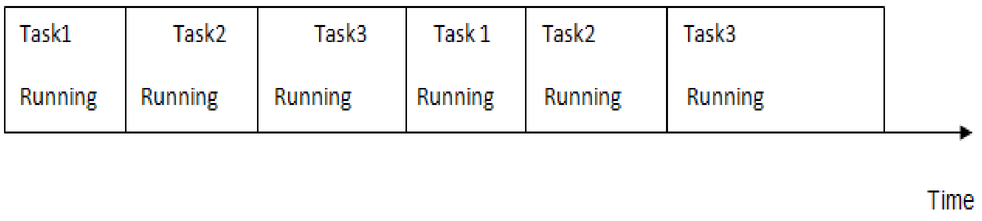


Fig. Round robin scheduling algorithm

3. Round robin with priority :

The round robin algorithm can be slightly modified by assigning priority levels to some or all the tasks . a high priority task can interrupt the cpu so that it can be executed . this scheduling algorithm can meet the desired response time for high priority task. For example in a bar code scanner high priority is assigned to the scanning operation. The cpu can execute this task by suspending the task that displays the item / price value . soft real time system can use this algorithm .

4. Shortest Job first :

In shortest job first task scheduling algorithm that task that will take minimum time to be executed will be given priority. This approach satisfies the maximum number of tasks but some tasks may have to wait forever

If the time for each task can be estimated beforehand the cpu can execute the task which take the least amount of time . effectively this is like a priority assignment , the priority being decided by the amount of time , the higher the execution time, the lesser the priority .the advantage of this algorithm is high number of tasks will be executed but the task with the highest amount of time will have to wait forever

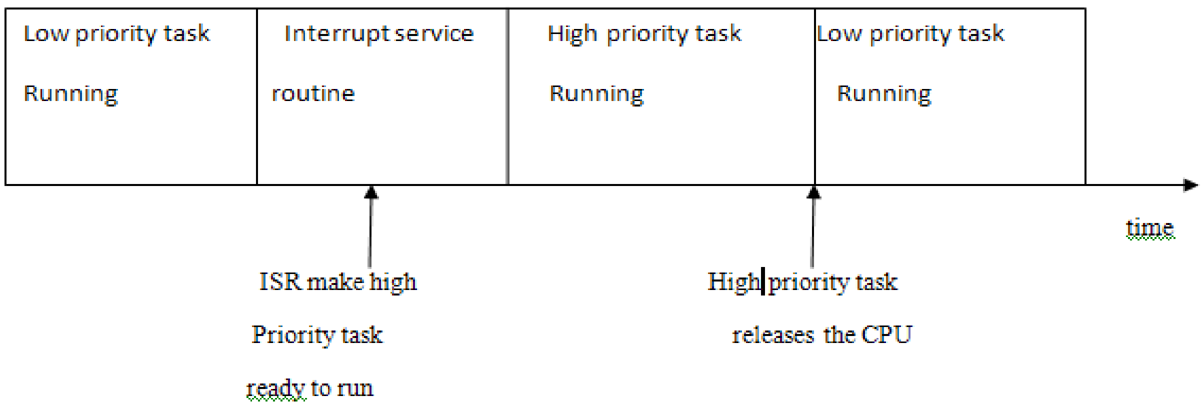
the kernel used in ES can implement priority based multitasking scheduling algorithms of two types

1.Non Preemptive multitasking

2.Preemptive multitasking

5. Non Preemptive multitasking:

Assume that you are making telephone call at public call office . you need to make many call, but you see another person waiting . you may make one call, ask the other person to finish his call and then you can make your next call . This is non preemptive multitasking also known as cooperative multitasking you are cooperating with the others in the queue .In non preemptive multitasking the tasks cooperate with each other to get their share of the CPU time . hence each task has to release the cpu and give control to another task on its own .each task is given a priority, but the priority has to be respected by the other tasks. If interrupts are enabled a high priority task can interrupt the running task and make high priority task ready to run . after the ISr is executed the CPU will continue to execute the low priority task only and when the low priority task releases the cpu the high priority task executed .The main advantage of this is that high priority task may have to wait for long time . also it is not possible to determine the exact response time of each of the tasks this type of multitasking is not suitable for ES real time system


Low priority task Running
Interrupt service routine
High priority task Running
Low priority task Running
time
ISR make high Priority task ready to run
High priority task releases the CPU

6. Preemptive multitasking :

In preemptive multi tasking the highest priority task always given the cpu time . if lower priority task is presently running and higher priority task is in ready to run state the running task is preempted and the higher priority task is executed this mechanism is shown in fig

Consider two task one is low priority task and other is high priority task . to start with low priority task is running as high priority task is in waiting for an external event to occur. After some time , the external event occurs and the high priority task now move to ready to run state through an interrupt the ISR is executed to move the high priority task from waiting state to ready to run state . then another ISR is executed to put the high priority task in the running state. Again after some time the high priority task may releases the cpu and the low priority task executed .if there are number of task in ready to run state the task with higher priority is always the first chance by the task scheduler Execution time of the high priority can be calculate is the advantage . It is used in commercial ES

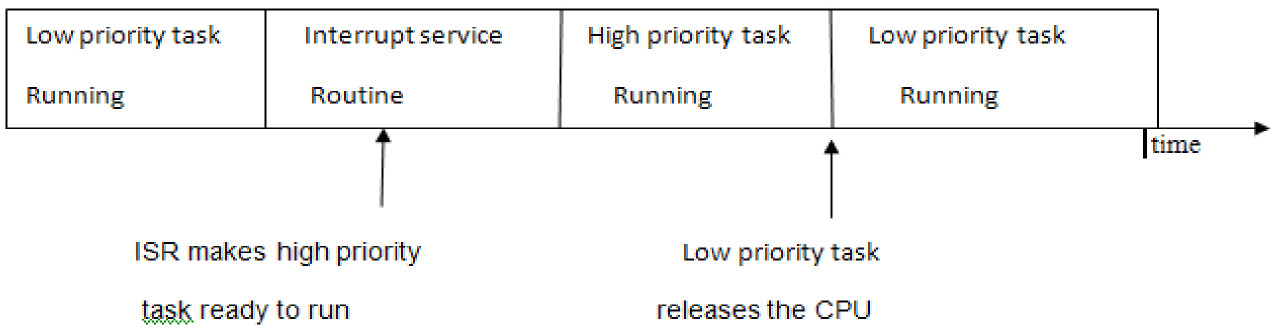

Low priority task Running
Interrupt service Routine
High priority task Running
Low priority task Running
time
ISR makes high priority task ready to run
Low priority task releases the CPU

Fig Preemptive multitasking

Q. 2) Attempt any FOUR of the following. **12M**

a)Compare RISC and CISC architecture

Ans:

(Any 4 point : Each 1 Marks)

Comparison of RISC and CISC architecture :

Aspect	RISC	CISC
Acronym	Reduced Instruction Set Computer	Complex Instruction Set Computer
Instruction set	Reduced	Complex
Compiler design	Easy to design	Hard to design
Hardware/Software	Stresses more on software	Stresses more on hardware
Memory management	Memory-to-memory operations	Register-to-register operations
Code size	High in code size	Less in code size
Clocking	Single clock is used	Multiple clocks are used
Instruction length	Single word instruction	Variable length instruction
Pipelining	Pipelining is the major feature	Doesn't support pipelining

b)Write 'C' program for 89C51 to read data from port p1 and p2 . compare the data and send bigger data on port p3.

Ans: **4M**

Program for 89C51:

```
#include <reg51.h>
void main(void)
{
unsigned char  a, b ;
P1=0xFF; //make P1 input port
P2=0xFF; //make P2 input port

P3=0x00; //make P0 output port
while (1)
{
a=P1; //get a byte from P1
b=p2;//get data from p2
if (a>b)
P3=a;
else
P3=b;
}


}
```

C)Distinguish between CAN and I2C protocols with respect to:

1)Data transfer rate

2) Number of fields

3)Addressing bits

4)Applications

Ans:

(Each point 1 Marks)

CAN and I2C protocols:

	I2C	**CAN**
Data transfer	Synchronous with 3 speeds 100kbps, 400kbps and 3.4mbps	Asynchronous with 250kbps upto 1mbps.
Number of field	07	08 (including 7 bits of frame end and 3 bits of inter frame gap)
Addressing bit	7-bit or 10-bit address	11 bit
application	To interface devises like watch dog, flash &RAM memory, Real time clock, Microcontrollers	elevator controllers, copiers, telescopes, production-line control systems, and medical instruments

d)Draw labeled diagram to interface 4 X 4 matrix keyboard to microcontroller 80C51.

Ans:

(Diagram 3 Marks , labeling 1 Mark)

Diagram: Interfacing keyboard to microcontroller

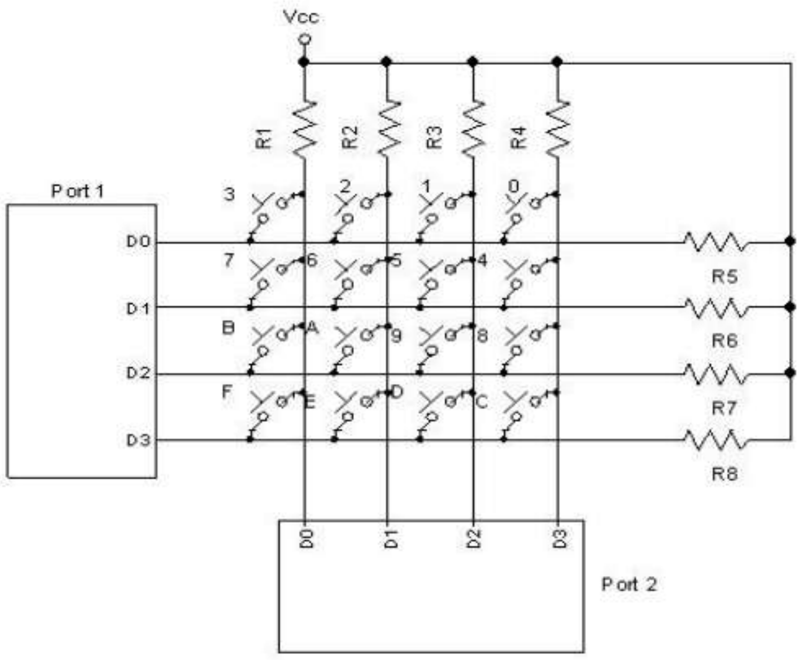
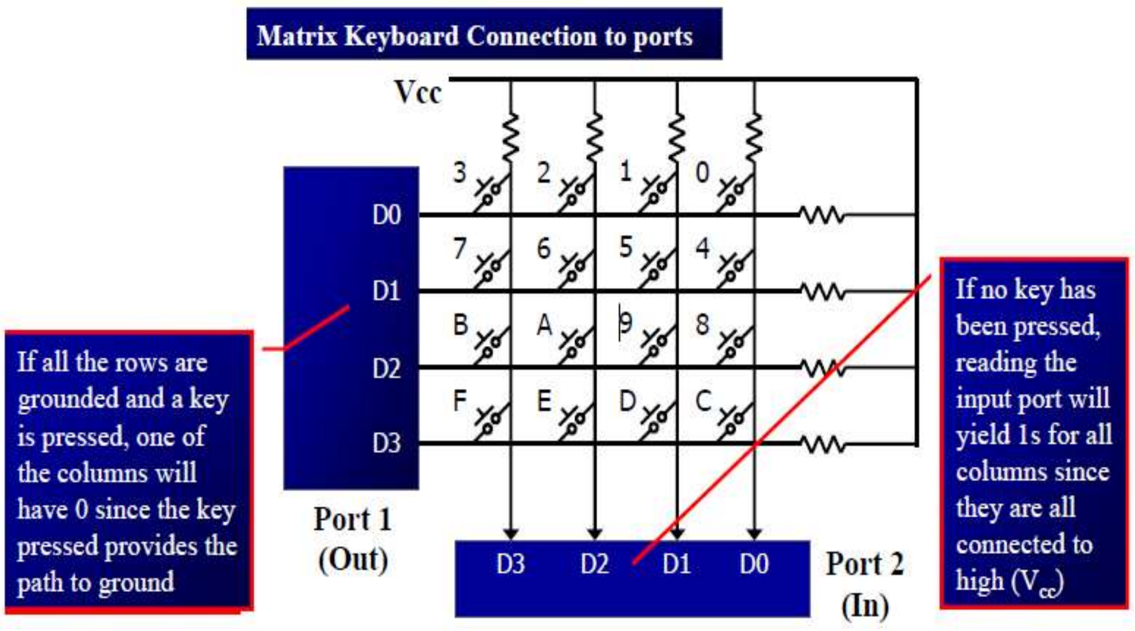


Fig. 1 Interfacing keypad to Microcontroller

Or

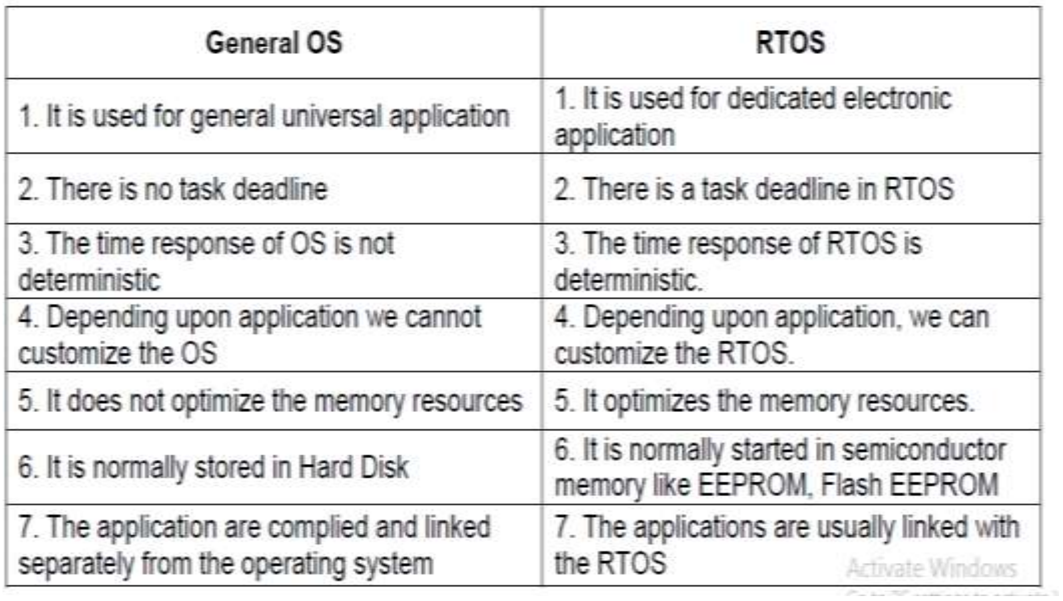

Matrix Keyboard Connection to ports
Vcc
3
2
1
0
7
6
5
4
B
A
9
8
F
E
D
C
D0
D1
D2
D3
Port 1 (Out)
D3
D2
D1
D0
Port 2 (In)
If all the rows are grounded and a key is pressed, one of the columns will have 0 since the key pressed provides the path to ground
If no key has been pressed, reading the input port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

e)Compare general purpose Operating system and RTOS

Ans:

(Any four point : 1 Mark Each)

General OS and RTOS:



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

f)Explain any four characteristics of embedded system.

Ans:

(**Any four :1 Marks Each)**

a) Embedded systems are application specific & single functioned; application is known a priori, the programs are executed repeatedly.
b) Efficiency is of paramount importance for embedded systems. They are optimized for energy, code size, execution time, weight & dimensions, and cost.
c) Embedded systems are typically designed to meet real time constraints; a real time system reacts to stimuli from the controlled object/ operator within the time interval dictated by the environment. For real time systems, right answers arriving too late (or even too early) are wrong.
d) Embedded systems often interact (sense, manipulate & communicate) with external world through sensors and actuators and hence are typically reactive systems; a reactive system is in continual interaction with the environment and executes at a pace determined by that environment.

e) They generally have minimal or no user interface.

Q 3.Attempt any FOUR of the following. 16M

a)Draw the pinout diagram of RS-232(DB9). State the function of all pins.

Ans:

(Diagram 2 Marks, Explanation 2Marks)

Pinout diagram of RS-232(DB9):

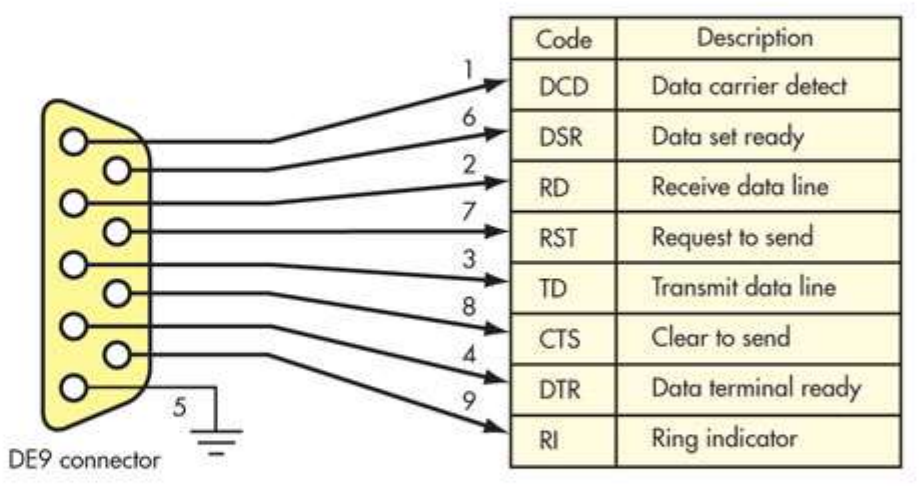


Fig. Pinout diagram of RS-232(DB9):

Function of all pins:

- **Pin 1 - Data carrier detect (DCD):** The DCE tells the DTE it is receiving a valid input signal.
- **Pin 2 - Received data (RD):** This is the actual signal received from the DTE.
- **Pin 3 -Transmit data (TD):** This is the transmitted signal from the DTE.
- **Pin 4 -Data terminal ready (DTR):** This line is from the DTE to the DCE indicating readiness to send or receive data.
- **Pin 5 -Signal ground:** This is the common ground connection for all signals.
- **Pin 6 -Data set ready (DSR):** The DCE tells the DTE it is connected and ready to receive.
- **Pin 7 -Request to send (RTS):** This signal from the DTE tells the DCE it is ready to transmit
- **Pin 8 -Clear to send (CTS):** This line from the DCE tells the DTE it is ready to receive data.
- **Pin 9 -Ring indicator (RI):** This line was used in older modem connections but isn't used anymore.

b)Write 89C51 'C' program to receive data serially from RX pin and send the data on port 1 continuously .Assume baud rate to be 9600 and crystal frequency as 11.0592 MHz.
Ans:

(Calculations: 1 Marks, Program 3 Marks)

Calculations: **4M**

We are assuming crystal value 11.0592 MHz.

o Timer clock Frequency is = XTAL / 12
= 11.0592MHz / 12 = 921.6 KHz
o UART Frequency is = Timer clock Frequency / 32
= 921.6KHz / 32 = 28.8 KHz
o Baud rate = UART Frequency / COUNTER Value

o COUNTER Value = UART Frequency / Baud Rate
= 28.8 KHz / 9600
= 3 As timer in microcontroller is up counter / timer, so counter value
= -3

Program:

```
#include <reg51.h>
Void main (main )
{
unsigned char mybyte;
TMOD=0x20;                    //use timer 1,8 –bit auto-reload
TH1=0xFD;                     //9600 baud rate
SCON=0x50
TR1=1;                        //start timer
while(1)                      //repeat forever
        {
                while(RI==0);   //wait to receive
                mybyte=SBUF;    //save value
                P1=mybyte;      //write value to port
                RI=0;
        }
}
```

OR

```
#include<reg51.h>
void baud_rate_9600(void);
void main(void)
{
        baud_rate_9600();
        P1= 0x00;          //make port 1 as output port
        RI=0;              //make initial rx=0
        while(1)
        {
              If (RI = = 1)        //wait for data received
              {
                    P1= SBUF;  //if data is received transfer it at P1
                    RI=0;      //Make RI i.e RX = 0
              }
        }
}

        void baud_rate_9600 ()
{
        TMOD=0x20;       // START THE TIMER IN AUTO RELOAD MODE
        TH1=0xFD;        // set th1 counter as 0fdh
        TL1=0xFD;
        PCON &= 0x7F;    // SET SMOD = 0
        SCON = 0x50      // SET UART IN MODE 1, ENABLE SRECIVE MODE
        TR1=1;           //Start the timer

}
```

(NOTE: Program may change but Calculations will not change. Student can also use the other logic. Please check the logic and understanding of students.)

c)What is deadlock in an embedded system .state the schemes to avoid deadlock.
Ans:
(Deadlock: 2 **Marks, Schemes :2Marks)**

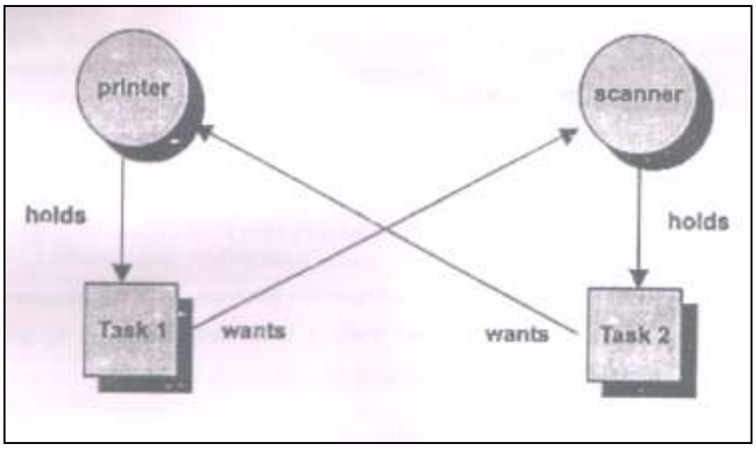
Deadlock:

A deadlock, also called as deadly embrace, is a situation in which two threads are each unknowingly waiting for resource held by other.

- Assume thread T1 has exclusive access to resource R1.
- Thread T2 has exclusive access to resource R2.
- If T1 needs exclusive access to R2 and T2 needs exclusive access to R1,
- Neither thread can continue.
- They are deadlocked.
- The simplest way to avoid a deadlock is for threads to:
    - Acquire all resources before proceeding
    - Acquire the resources in the same order
    - Release the resource in the revere order

- Deadlock is the situation in which multiple concurrent threads of execution in a system are blocked permanently because of resources requirement that can never be satisfied.
- A typical real-time system has multiple types of resources and multiple concurrent threads of execution contending for these resources. Each thread of execution can acquire multiple resources of various types throughout its lifetime.
- Potential for deadlock exist in a system in which the underlying RTOS permits resources sharing among multiple threads of execution.

Following is a deadlock situation between two tasks.



- In this example, task #1 wants the scanner while holding the printer. Task #1 cannot proceed until both the printer and the scanner are in its possession.
- Task #2 wants the printer while holding the scanner. Task #2 cannot continue until it has the printer and the scanner.
- Because neither task #1 nor task#2 is willing to give up what it already has, the two tasks are now deadlocked because neither can continue.

Methods of avoid Deadlock:-

- Mutual exclusion
- Hold & wait or resource holding
- No preemption
- Circular wait

d)Draw the block diagram of embedded system.explain any one subsystem.

Ans:

(Diagram 2 Marks, Explanation 2Marks).

Block diagram of embedded system:

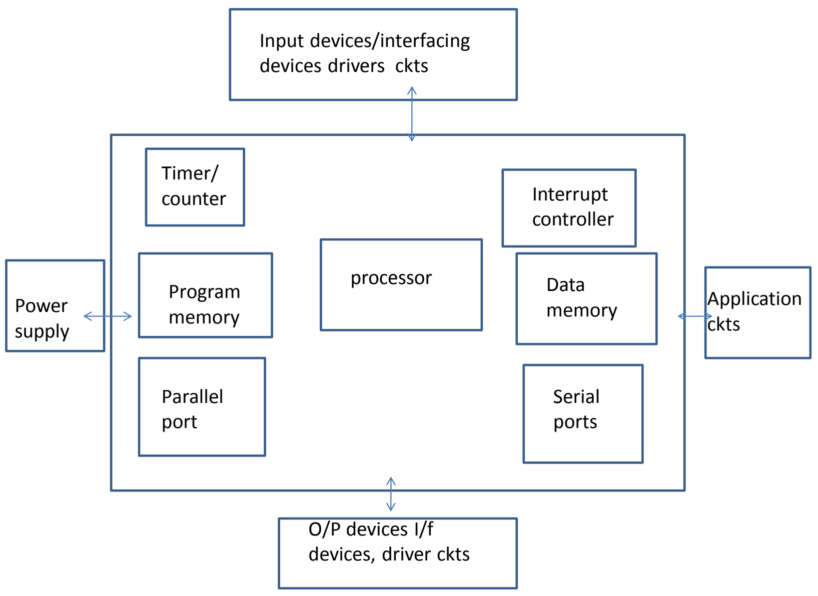


Fig. block diagram of embedded system.

Explanation:

Processor:

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

1. Instruction set
2. Maximum bits of operation on single arithmetic and logical operation
3. Speed
4. Algorithms processing and capability
5. Types of processor(microprocessor, microcontroller, digital signal processor, application specific processor, general purpose processor)

Power source:

Internal power supply is must. Es require from power up to power down to start time task. Also it can run continuously that is stay "On' system consumes total power hence efficient real time programming by using proper 'wait' and 'stop' instruction or disable some unit which are not in use can save or limit power consumption.

Clock / oscillator Circuits

The clock ckt is used for CPU, system timers, and CPU machine cycles clock controls the time for executing an instruction. Clock oscillator may be internal or external .It should be highly stable

Real time clock(RTC):

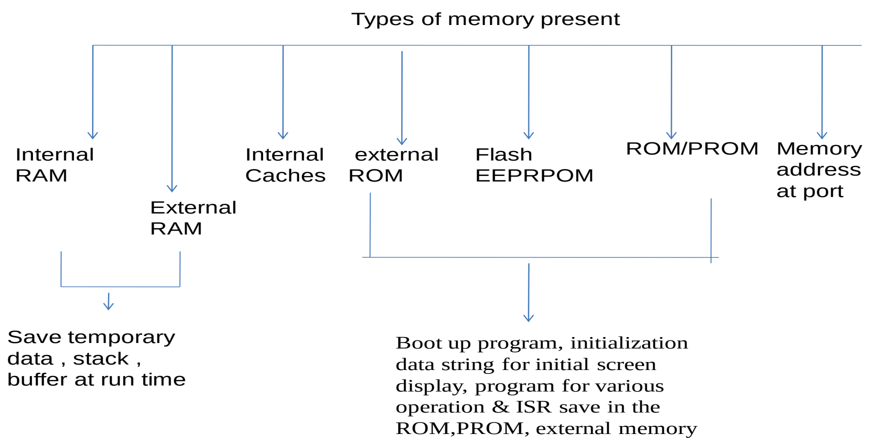
It require to maintain scheduling various tasks and for real time programming RTC also use for driving timers, counters needs in the system.

Resets Ckts and power on reset:

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

1. System program that execute from beginning 2. System boot up program 3. System initialization program

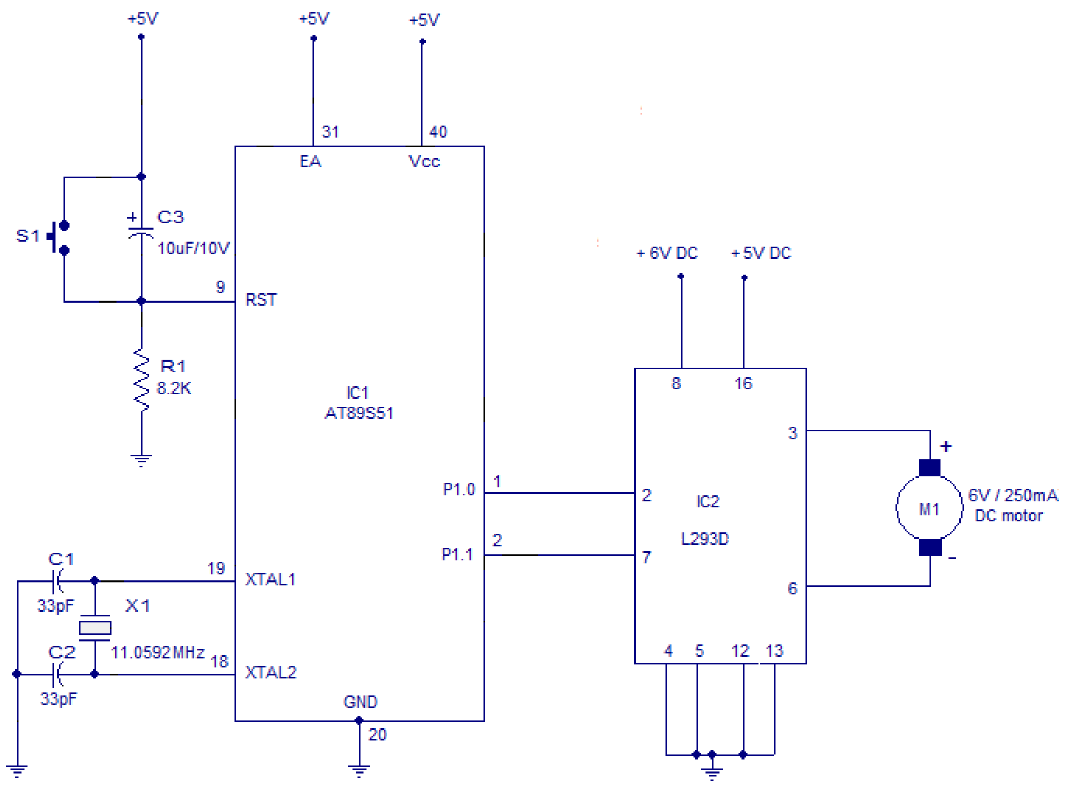
Memories:


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

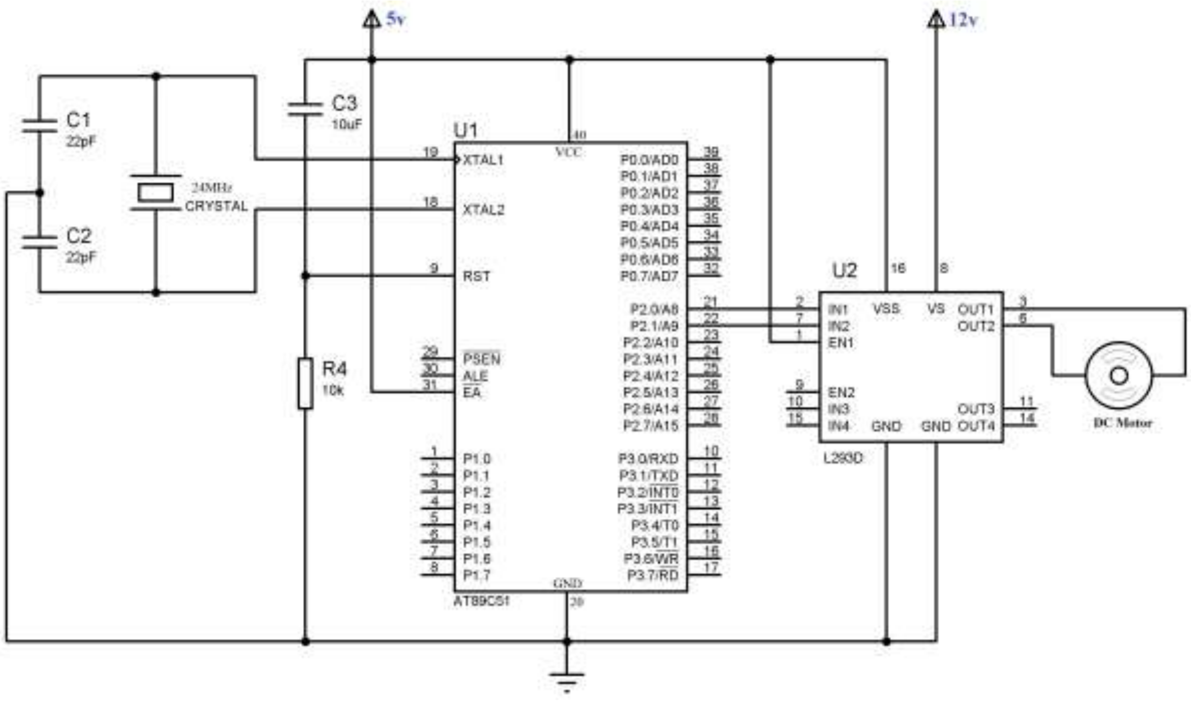
e)Draw labelled diagram to interface DC motor with 89C51. Write 'C' program to rotate the motor continously.

Ans:

(Diagram 2 Marks, Program 2Marks).



OR



Program:

```
#include<reg51.h>
sbit CW_DIRECTION=P1^0;           //P1.0 for Clock Wise direction
sbit CCW_DIRECTION=P1^1;          //P1.1 for Counter Clock Wise direction
void main(void)
{
   CW_DIRECTION =0;               //Motor OFF
   CCW_DIRECTION =0;
   while(1)
   {
          CW_DIRECTION =1;        // Rotate the motor in clockwise direction continously
          CCW_DIRECTION =0;
   }
}
```

(NOTE: Program may change. Student can also use the other logic. Please check the logic and understanding of students.)

Q4. a) Attempt any THREE of the following: 12M

i) Explain DSP in brief. State any two applications.

Ans: [DSP explanation 3 marks and any two applications 1 Mark]

DSP: Digital Signal Processors 3M

DSP are powerful special purpose 8 / 16 / 32 bit microprocessor designed to meet the computational demands and powerful constraints of today's embedded studio, video and communication applications.

- DSP are 2 to 3 times faster than the general purpose microprocessor in signal processing applications. This is because of the architectural difference between the two.
- DSPs implement algorithms in hardware which speeds up the execution, depends primarily on the clock for the processor.
- DSP can be viewed as a microchip designed for performing high sped computational operations for addition, subtraction and division.
- DSP unit having the following key units - Computational Engine, Program Memory, Data memory and I/O units

DSP Applications: 1M

- Audio video signal processing
- Telecommunication
- Multimedia
- Applications where processing of large amount of real time calculations are required.

ii) Compare Synchronous and asynchronous serial communication

Ans: (Each compares point - 1 Mark) **4M**

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.

iii) List advantages and disadvantages of embedded systems.

Ans: [Only list - Advantages 2 marks and Disadvantages 2 Marks]

Advantages: **2M**

1) Design and Efficiency

2) Cost

3) Accessibility

4) Maintenance

5) Redundancies

Disadvantages: **2M**

1) It is designed to do a specific task only, other task can not be executed.
2) It is difficult to upgrade the machine
3) Harder to carry files from one machine to another

iv) Explain inter process communication in brief. State various inter process communication methods.

Ans: (Explanation 2 marks and any 2 Methods- 2 Mark)

Interprocess communication (IPC): **2M**

i. Interprocess communication (IPC) is a set of programming interfaces that allow a programmer to coordinate activities among different program processes that can run concurrently in an operating system.
ii. This allows a program to handle many user requests at the same time.
iii. Since even a single user request may result in multiple processes running in the operating system on the user's behalf, the processes need to communicate with each other.
iv. The IPC interfaces make this possible.
v. Each IPC method has its own advantages and limitations so it is not unusual for a single program to use all of the IPC methods.

IPC methods: **2M**

1. Pipes - Named pipes and un named pipes.
2. Message queue
3. Semaphores
4. Shared memory
5. Sockets

b) Attempt any ONE of the following: 6M

i) Write 89C51 'C' language program to generate square wave program of 10 KHz on pin P2.7 using timer 0. Assume crystal frequency as 12MHz.

Ans:

(Calculations --- 2Marks, Program --- 4 Marks)

Calculations: 2M

Crystal value = 12 MHz.

- Look at the following steps for 10 KHz frequency calculations with 12 MHz.
- The period of the square wave = 1 / 10 KHz
  = 0.1 ms.
  =100 uSec
- The high or low portion of the square wave = Time period / 2
  = 100 us / 2
  = 50 uSec.
- Timer clock Frequency is = XTAL / 12
  = 12 MHz / 12
  = 1 MHz
- Timer clock period is = 1/ Timer Frequency
  = 1 / 1 MHz
  = 1 uSec
- Counter = Delay / timer clock period
  =50 uSec / 1 uSec
  = 50
- Timer Reload value = Maximum Count – Counter
  = 65536 – 50
  = (65486)d
- Timer Reload value in HEX = (65486)d
  = (FFCE) h .
- TL0 = 0xCE h and TH0 = 0xFF h.

//C language program to generate square wave over Port Pin P2.7 using timer 0 2M

```
#include <Intel\8052.h>
#include <standard.h>
Void T0M1delay (void);          //Timer 0, Mode 1(16 bit timer)
SBIT  OUTPUT    P2^7;           // Initialize Port pin P2.7 as output
Void main ()
{
        While (1)
        {
                OUTPUT= ~ OUTPUT;  // toggle P2.7
                T0M1delay ();                    // delay of 50 uSec
        }
}

Void T0M1delay ()               // Timer 0, Mode 1(16 bit timer) - delay of 50 uSec
        {
                TMOD        = 0x01;       // Timer 0, Mode 1(16 bit timer)
                TL0   = 0xCE;      //Load TL0 = CEh
                TH0   = 0xFF;      //Load TL0 = FFh
                TR0   = 1;         //Run the timer 0
                while (TF0 = = 0)  // Wait for TF0 to overflow
                TR0   = 0;         //Stop the timer 0
                TF0   = 0;         //Clear TF0
        }
```

(NOTE: Program may change but Calculations will not change.

Student can also use the other logic. Please check the logic and understanding of students.)

ii) Draw the diagram to interface DAC0808 to microcontroller 89C51. Write a 'C' language to generate a saw tooth wave continuously.

Ans:

(Neat and labeled diagram - 3 Marks, Correct program in Assembly or 'C' - 3 Marks)

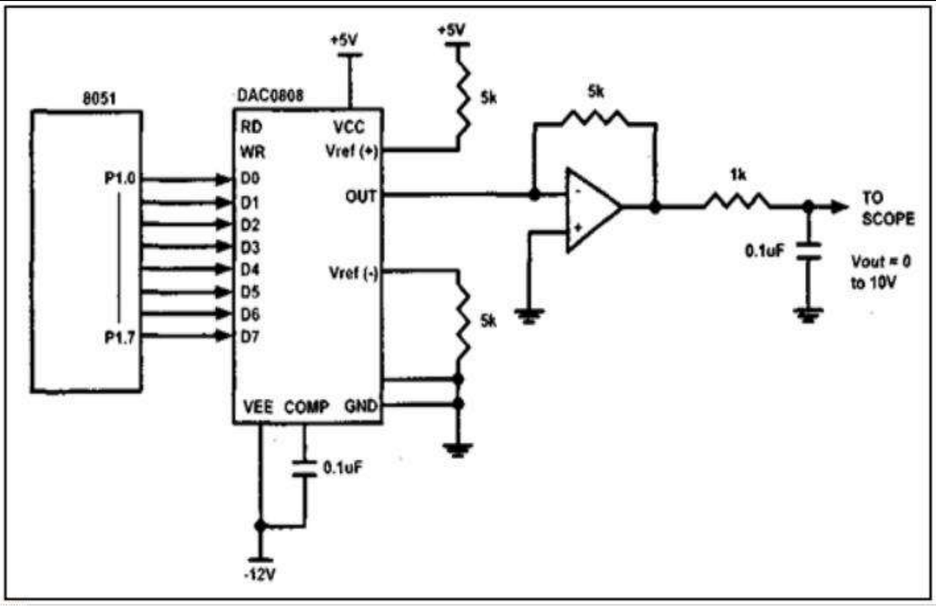


Fig. Circuit Diagram of interfacing DAC0808 with 8051 Microcontroller

//C language program GENERATING SAWTOOTH WAVE USING DAC0808

```
//P1 = DAC D0 TO D7
#include <Intel\8052.h>
#include <standard.h>
unsigned char c;
void main ()
{       while (1)
        {               for (c=0;c<256;c++)
                        {
                                P1 = c;
                        }
        }
}
```

(NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

Q.5 Attempt any FOUR of the following: 16M

a) If the content of ACC = 0 x 02 and P1=0 x F3. State the result after execution of following statements independently:

(i) result = ACC & P1

(ii) result = ACC | P1

(iii) result = ACC ^ P1

(iv) result = ~ P1

Ans: (Each answer 1 Mark)

ACC = 0 x 02 and P1=0 x F3.

(i) result = ACC & P1= 0 x 02 & 0 xF3 = **result = 0 x 02** (Bit by bit AND)

(ii) result = ACC | P1 = 0 x 02 & 0 xF3 = **result = 0 x F3** (Bit by bit OR)

(iii) result = ACC ^ P1 = 0 x 02 & 0 xF3 = **result = 0 x F1** (Bit by bit EX-OR)

(iv) result = ~ P1 = 0 x 02 & 0 xF3 = **result = 0 x 0C** (NOT of P1)

b) State the features of Zigbee. State four applications.

Ans: (Any 4 Features of Zigbee - 2 Marks, Any 4 Applications of Zigbee - 2 Marks)

Features: 2M

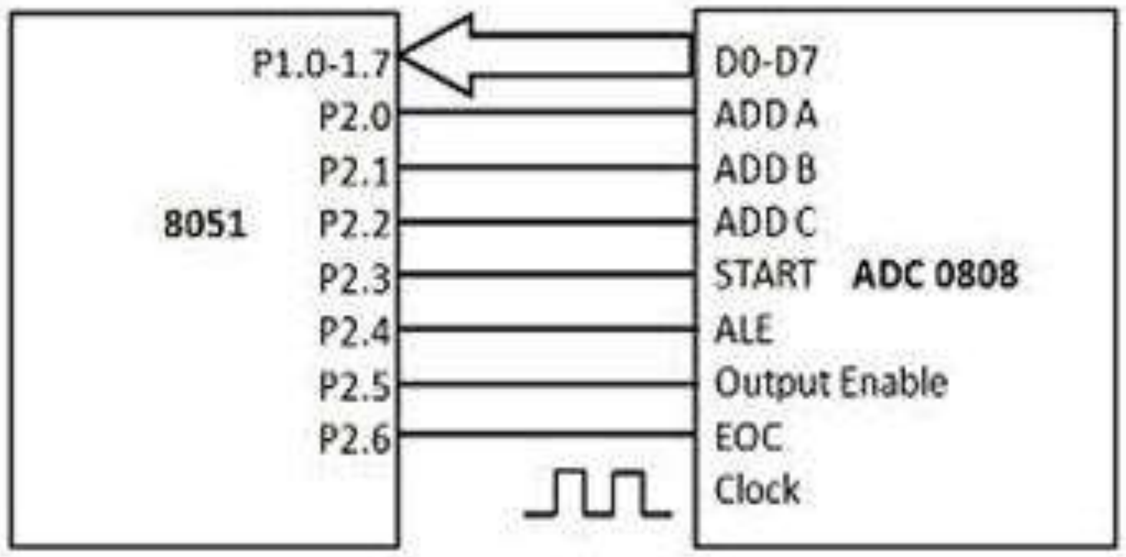
1. Outdoor range up to 300 feet (90 m)
2. Indoor range up to 100 feet (30 m)
3. Data rate up to 250 Kbps
4. 2.4 GHz frequency band (accepted world-wide)
5. Communication: CMOS UART @ ~3.3V
6. Voltage requirements: 2.8 to 3.4 VDC (reduced performance below 3.0V)
7. Digital I/O lines
8. I/O line passing
9. Analog-to-digital conversion
10. 16 channels
11. 128-bit encryption

Application Ideas: **2M**

1. Remote Industrial Control and Monitoring
2. Long range remote control
3. Wireless data acquisition
4. Home Automation
5. Automated Meter reading system
6. Security systems
7. Point to point cable replacement
8. Lighting Control
9. Medical / Patient tracking system

c) Draw labelled diagram to interface ADC 0808 with microcontroller 89C51

Ans: (Neat ADC Interfacing diagram – 4 Marks)



OR

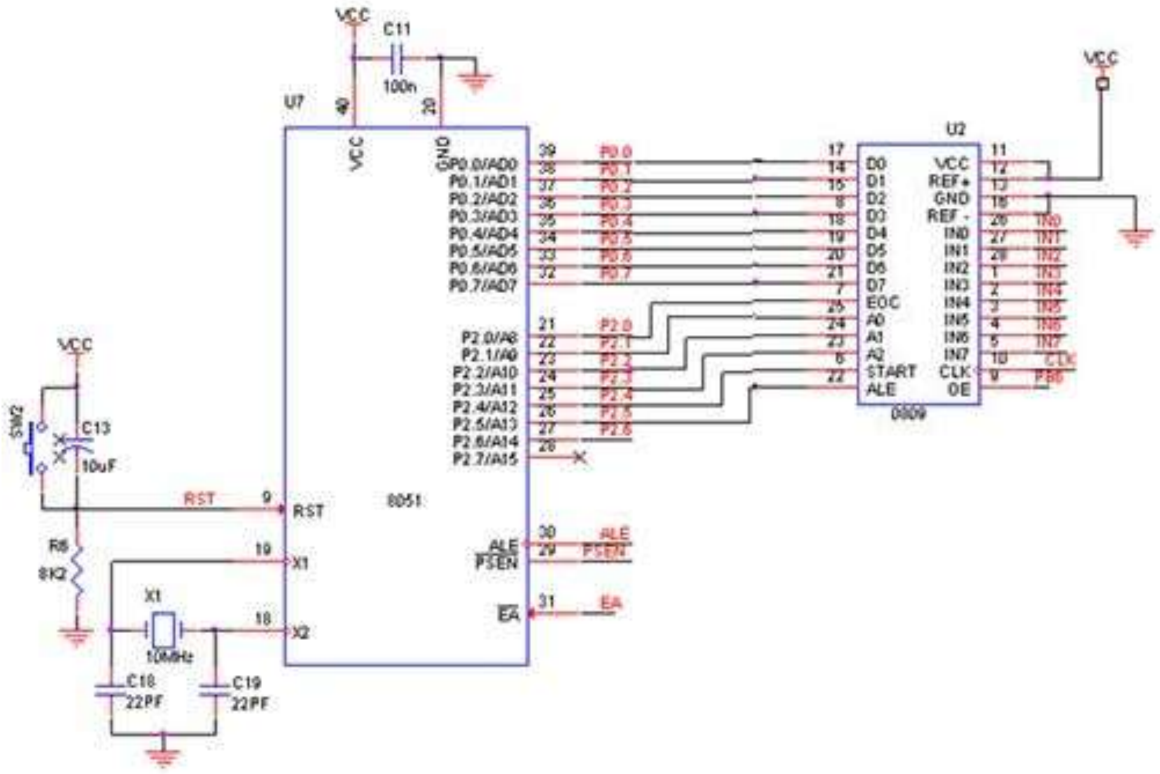


Fig. ADC Interfacing diagram

d) State and explain any four key specifications of RTOS

Ans: [4 key specifications – 4 marks]

- **Key Specifications of RTOS:** **4M**
  1. **Reliability:** The RTOS is reliable, because it is available for all time and normally it does not fail to perform any function/operation. The reliability of system also depends on the hardware board support package and application code.
  2. **Predictability:** In RTOS, the user knows within How much time period the RTOS is going to perform the task i.e. The RTOS has predictability. We can predict, determine how much time takes by RTOS.
  3. **Performance:** The performance of RTOS is very fast so that it can fulfill all timing requirement.

4. **Compactness:** The RTOS provide compactness. It required less memory space for storage and hence can be used for portable application, like cellphone, ECG machine, etc.
5. **Scalability:** RTOS can be used in a wide variety of embedded. They must be able to scale-up or scale-down to suit the application.

e) Compare assembly language program and embedded 'C' programming. (Any four points)

Ans: [Any 4 points – 4 marks]

Sr. No.	**Point**	**Assembly Language**	**Embedded C**
1	Time for Coding	More time required for coding	Less time is required for coding
2	Ease for Coding	Coding in assembly language is difficult	Coding in Embedded C is Easy
3	Processor Dependency	Assembly Language is Processor / Controller dependent	Embedded C language is Processor / Controller independent
4	Type of Language	Assembly Language is LOW Level language	Embedded C language is HIGH Level language
5	Memory Required for Program Coding	Program memory required is Less.	Program memory required is More.

f) Write 'C' program to rotate the stepper motor by two complete rotations and then stop. Assume step angle as 1.8^0.

Ans:

(1 Marks for count for two rotations, 3 Marks for programming)

Step-angle of stepper – motor = 1.8^0

Two rotations = 360^0 X 2 = 720^0

Total no's of step required = $720^0/1.8^0$ = (400)

We are sending 4 pulses in one loop and then checking counter. So counter is 400 / 4 = 100

```
;   A   B   C   D
;   1   0   0   1   =09   A & D COIL
;   1   1   0   0   =0C   A & B COIL
;   0   1   1   0   =06   B & C COIL
;   0   0   1   1   =03   C & D COIL
```

// C language program Stepper motor interfacing

```
#include <Intel\8052.h>
#include <standard.h>
/*
COIL A = P0.0
COIL B = P0.1
COIL C = P0.2
COIL D = P0.3
*/
void delay_ms(unsigned int t);

void main ()
{
        P0 = 0xFF;                  //MOTOR OFF
        For (x=0; x<=100; x=++)
        {
                P0 = 0x09;          //COIL A & D ON
                delay_ms (100);
                P0 = 0x0C;          //COIL A & B ON
                delay_ms (100);
                P0 = 0x06;          //COIL B & C ON
                delay_ms (100);
                P0 = 0x03;          //COIL C & D ON
                delay_ms (100);
        }
        P0 = 0xFF;                  //MOTOR OFF after two rotations
        While (1)
        {                           //infinite loop to terminate the program
        }
}

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

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

Please check the logic and understanding of students.)

Q.6 Attempt any FOUR of the following. 16 M

a)Explain JTAG in brief.

Ans:

Joint Test Action Group(JTAG) 4M

JTAG is the common name for the IEEE 1149.1 standard test access port and boundary scan architecture .

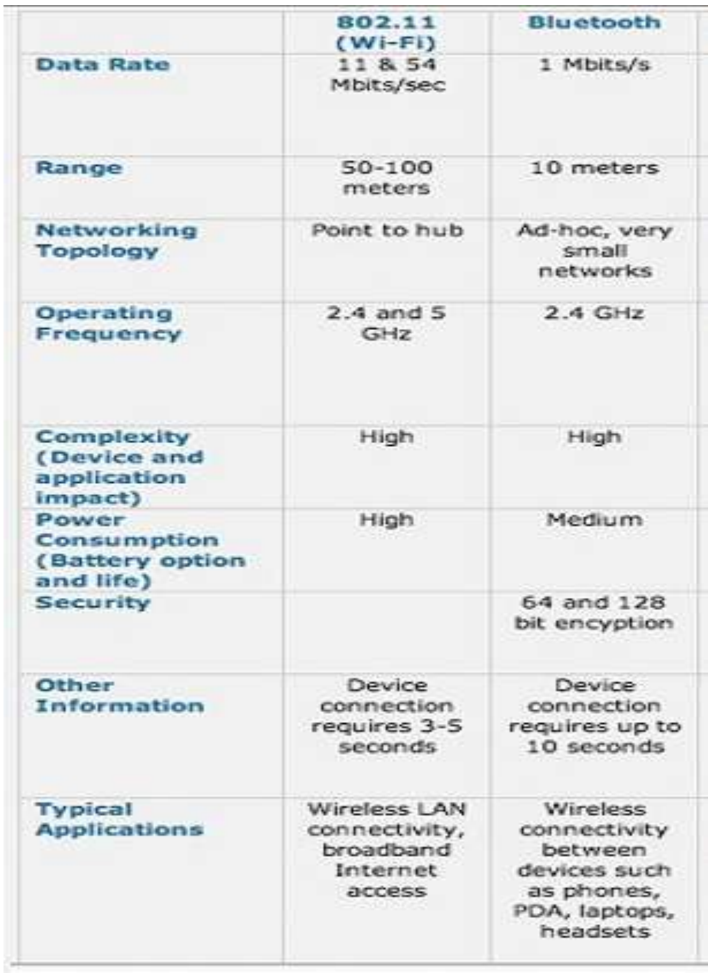
It was initially devised by electronic engineers for testing printed circuit boards (PCB) using boundary scan.

Today it is widely used for IC debug ports. Embedded systems development relies on debuggers communicating with chips with JTAG to perform operations like single stepping & break pointing.

An ICE (In Circuit Emulator)uses JTAG as the transport mechanism to access on-chip debug modules inside the target CPU. This modules let software developers debug the software of an embedded system directly at the machine instruction level when needed .

b)Compare Wi-Fi(IEEE 802.11)with Bluetooth. 4M

Ans:



	802.11 (Wi-Fi)	Bluetooth
Data Rate	11 & 54 Mbits/sec	1 Mbits/s
Range	50-100 meters	10 meters
Networking Topology	Point to hub	Ad-hoc, very small networks
Operating Frequency	2.4 and 5 GHz	2.4 GHz
Complexity (Device and application impact)	High	High
Power Consumption (Battery option and life)	High	Medium
Security		64 and 128 bit encyption
Other Information	Device connection requires 3-5 seconds	Device connection requires up to 10 seconds
Typical Applications	Wireless LAN connectivity, broadband Internet access	Wireless connectivity between devices such as phones, PDA, laptops, headsets

C)Draw the labeled diagram to interface a switch to pin P0.0 and a relay to pin P2.0 of 89C51.

Ans: **4M**

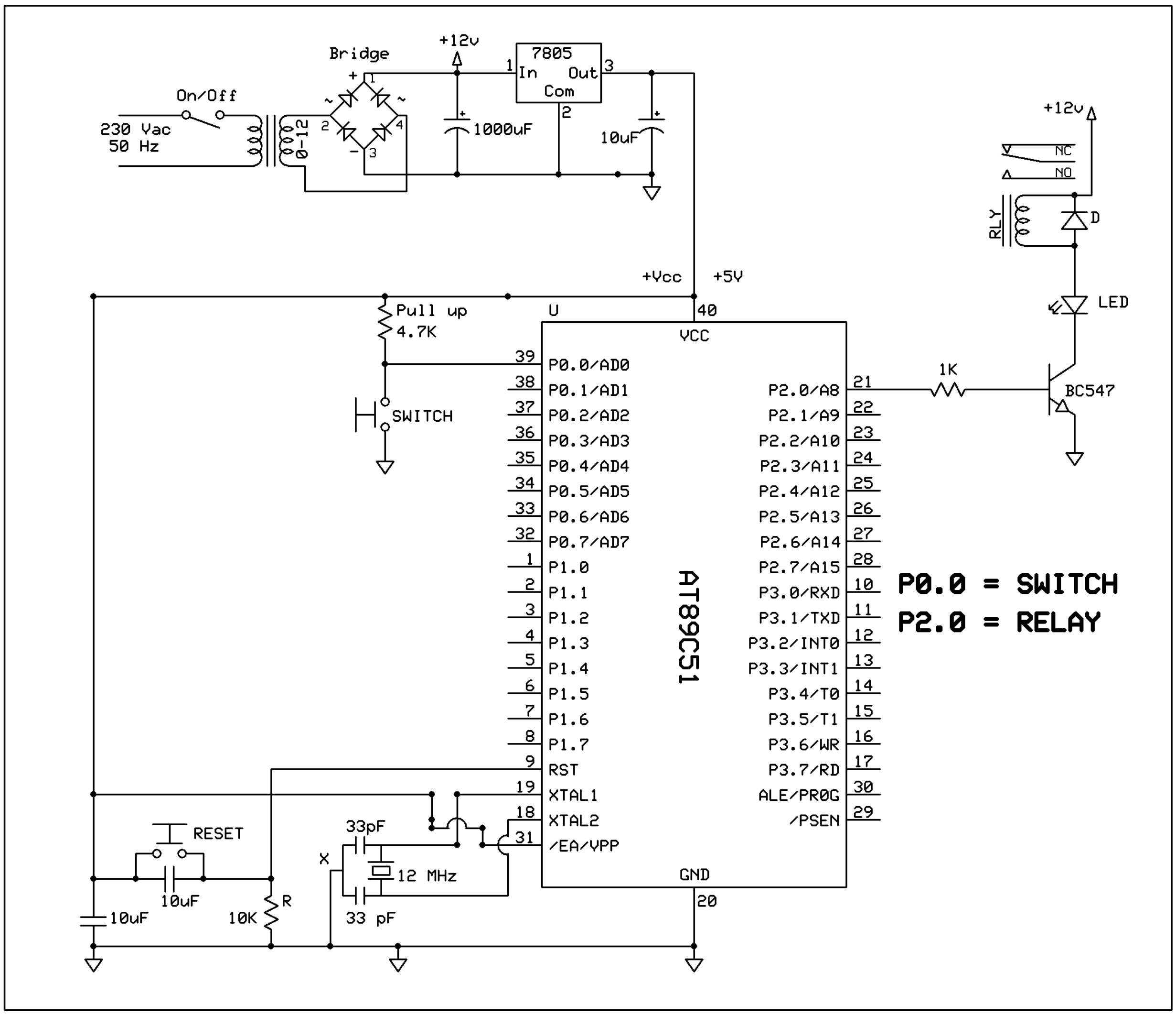


Fig.DS89C4 x0 Connection to relay.

d)Draw the diagram to interface LED. To pin P1.7 of 89C51 .write 'C' program to blink the LED.

Ans:

(Diagram 2 Marks, program 2Marks)

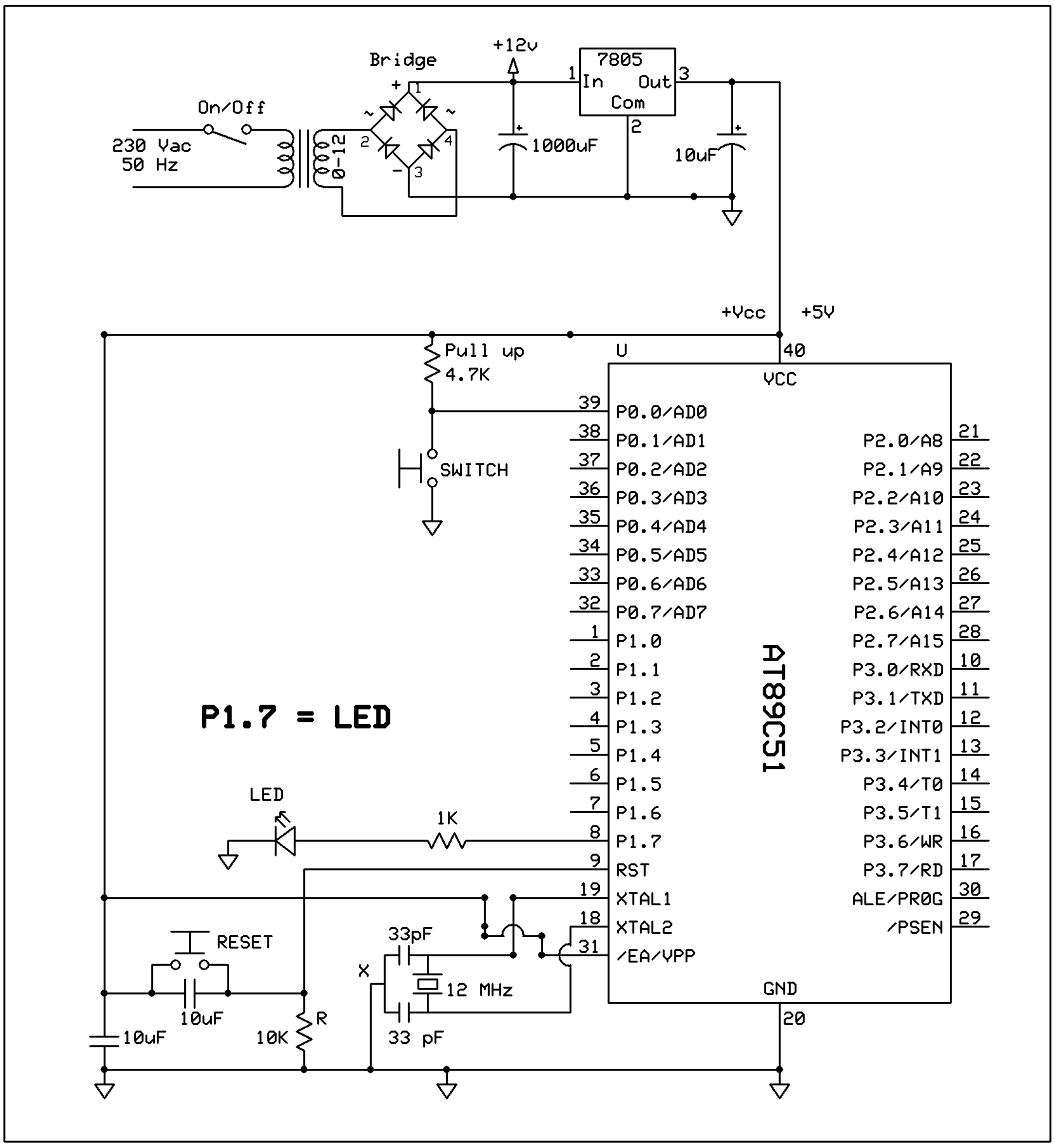


Fig. Interfacing LED. To pin P1.7 of 89C51

Program:

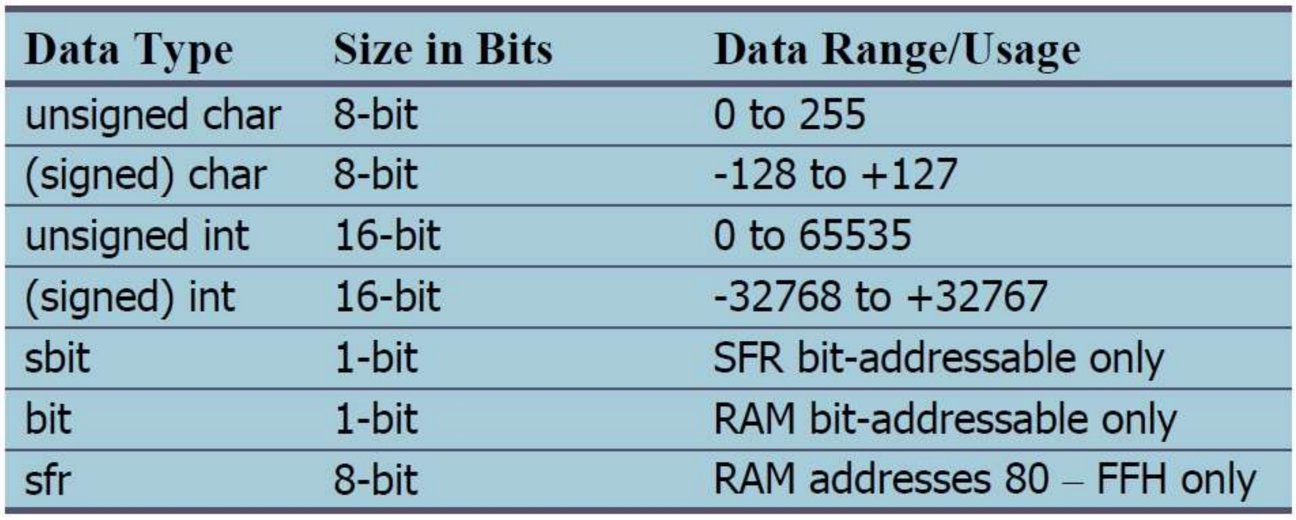
```
//LED BLINK USING C
#include<reg51.h>
sbit LED=P1^7;          //P1.7 as output LEDLED
void delay1();
void main(void)
{
        LED=0;
        while(1)
        {
                LED=0;
                delay1();
                LED=1;
                delay1();
        }
}

void delay1(void)
{
                TMOD=0x10;
                TH1=0x00;
                TL1=0x00;
                TR1=1;
                while(TF1==0);
                TR1=0;
                TF1=0;
}
```

(NOTE: Program may change. Student can also use the other logic. Please check the logic and understanding of students.)

e)List various data types in embedded C with their data range.

Ans:



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