Subject code: 12269
Q 1 a) Attempt any THREE of the following:

Q 1 a i) Describe Timer modes of 8051.

Ans: Timer 0 and Timer 1 can both be used as either Counters or Timers. There are 4 different operating modes of timers, which can be set using a special function register TMOD.

Mode 0, 1 and 2 are same for both the timers but mode 3 is different. The format of TMOD register is as shown below.

<table>
<thead>
<tr>
<th>TMOD.7</th>
<th>TMOD.6</th>
<th>TMOD.5</th>
<th>TMOD.4</th>
<th>TMOD.3</th>
<th>TMOD.2</th>
<th>TMOD.1</th>
<th>TMOD.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>GATE</td>
<td>C/T</td>
<td>M1</td>
<td>M0</td>
<td>GATE</td>
<td>C/T</td>
<td>M1</td>
<td>M0</td>
</tr>
<tr>
<td>TIMER 1</td>
<td></td>
<td></td>
<td></td>
<td>TIMER 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To select the mode C/T and M0, M1 is to be programmed as below

<table>
<thead>
<tr>
<th>C / T</th>
<th>Timer or Counter Selector</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/T= 0 timer operation (input from internal system clock).</td>
<td></td>
</tr>
<tr>
<td>C/T= 1 Counter operation (input from “Tx” input pin)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1 M0</th>
<th>Mode 0 : 8-bit timer /counter THx with TLx as 5 bit pre scaler.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>Mode 1 : 16 bit Timer /Counter THx, TLx are cascaded; there is no pre scaler</td>
</tr>
<tr>
<td>0 1</td>
<td>Mode 2: 8 – bit auto reload timer/counter THx holds value which is to be reloaded into TLx each time it overflows</td>
</tr>
<tr>
<td>1 0</td>
<td>Mode 3: Timer 0 - TL0 is an 8bit timer /counter controlled by standard Timer 0 control bits. TH0 is an 8 – bit timer controlled by timer 1 control bits</td>
</tr>
</tbody>
</table>

---------2 Marks for above description
Mode 0: When programmed to operate in Mode 0, the timer is 13 bit wide, this mode is same for both Timer0 and Timer1. When the count rolls over from all 1s to all 0s, it set the flag TF1 or TF0 according to the timer used. The 13 bit register consists of all 8 bits of TH1 and lower 5 bits of TL1. Upper 3 bits of TL1 are not used. Maximum count applicable is 1FFFH.

Mode 1: Mode 1 is same as mode 0 except that the timers are 16 bit wide and settings are similar for both Timer 0 and Timer 1. Maximum count applicable is FFFFH.

Mode 2: Mode 2 operation is same for both timer 1 and timer 0. Timer register is configured as 8 bit counter TL1 and TH1 contains the reload value for TL1. When TL1 overflows i.e. crosses #FFH the flag bit TF1 is set and at the same time TL1 is loaded with value in TH1. This is also called auto reloading of timer register.

Mode 3: This mode is also called as split timer mode. In this mode TH0 and TL0 are used as two separate timers and timer 1 is used just to hold count.

Q 1 a ii) What are different types of Advanced Serial high speed Bus.

Ans: Different types of buses are

1) I²c (Inter Integrated circuit)
2) CAN (Controller Area Network)
3) Fire Wire (IEEE 1394 Bus standard)
4) USB (Universal Serial Bus)

Q 1 a iii) With suitable flow chart explain the steps involved in Embedded Software development cycle.

Ans: EMBEDDED SOFTWARE DEVELOPMENT CYCLE: Development cycle of embedded systems is different from that of conventional software development cycle. This is because embedded systems are custom designed. Embedded system development cycle is as shown below.
Q 1 a iv) Explain Interprocess Communication.

Ans: - Interprocess communication

In computing, Inter-process communication (IPC) is a set of methods for the exchange of data among multiple threads in one or more processes. Processes may be running on one or more computers connected by a network. IPC methods are divided into methods for message passing, synchronization, shared memory, and remote procedure calls (RPC). Interprocess communication makes use of global variable many times.

-------- 2 Mark for above description
Interprocess communication methods

1. Signal
2. Semaphore
3. Queue, Pipe and mailboxes
4. Socket
5. Remote procedure call for distributed process

------------- 2 Mark for mention and brief explanation of any two of above methods

Q 1 b) Attempt any ONE of the following:

Q 1 b (i) State any four features and advantages of embedded system.

Ans: Features of Embedded System

1-Embedded systems exhibit real-time performance which is the highlighting feature among all microprocessor based systems.

2-Embedded systems are not always standalone devices. Many embedded systems consist of small, computerized parts within a larger device that serves a more general purpose.

3- Size & Weight: Microcontrollers are designed to deliver maximum performance for minimum size and weight.

4-Reduce cost improved performance and high efficiency

5- Compact size, increased functionality.

------------- 3 Mark for any three relevant points given in above description

Advantages of embedded system:-

1. Real time operation: where reactive computations occur in response to external events.
2. Small size and low weight.
3. Low power
4. Withstanding of harsh environment condition like heat, vibration, shock, lightning, RF interface etc.
5. Safe critical operations.

------------- 3 Mark for any three relevant points

Q 1 b ii) State scheduling algorithm of RTOS and describe concept of Round Robin Scheduling with example.
Different Scheduling algorithms of RTOS:

- First in first out
- Round-robin algorithm
- Round-robin with priority
- Shortest job first
- Non-preemptive multitasking
- Preemptive multitasking

--------- 3 Mark for any three correct names

Round-robin Scheduling 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. 7.4, if their tasks 1, 2 and 3 are waiting in the queue, the CPU first executes task1 then task2 then task3 and then task1.

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>Running</td>
<td>Running</td>
<td>Running</td>
<td>Running</td>
<td>Running</td>
</tr>
</tbody>
</table>

--------- 3 Mark for relevent description

Q 2 Attempt any FOUR of the following:

Q 2 a) State two external hardware interrupt I 8051 microcontroller. At what port pins are they located? State the two bits TCON.0 and TCON.2 play in execution of the external interrupts.

Ans: - Two external hardware interrupts of 8051 microcontroller are INT0 and INT1 they are also called EX1 and EX2.

--------- 1 Mark for correct names
INT0 is located at Pin no.12 that is P3.2

INT1 is located at Pin no.13 that is P3.3

--- 1 Mark for above two points

<table>
<thead>
<tr>
<th>TCON.7</th>
<th>TCON.6</th>
<th>TCON.5</th>
<th>TCON.4</th>
<th>TCON.3</th>
<th>TCON.2</th>
<th>TCON.1</th>
<th>TCON.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF1</td>
<td>TR1</td>
<td>TF0</td>
<td>TR0</td>
<td>IE1</td>
<td>IT1</td>
<td>IE0</td>
<td>IT0</td>
</tr>
</tbody>
</table>

IT1: TCON.2: Interrupt 1 type control bit. Set/ Cleared by software to specify falling edge/ low-level triggered external interrupt.

IE0: TCON.1: External Interrupt 0 edge flag. Set by CPU when external interrupt (H-to L transition) edge is detected cleared by CPU when interrupt is processed.

--- 2 Mark for above description

Q 2 b) Describe the format of SCON

Ans: - Serial Port Control (SCON) Register

<table>
<thead>
<tr>
<th>Bit name</th>
<th>Value after reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM0 -</td>
<td>0</td>
</tr>
<tr>
<td>SM1 -</td>
<td>0</td>
</tr>
<tr>
<td>SM2 -</td>
<td>0</td>
</tr>
<tr>
<td>REN -</td>
<td>0</td>
</tr>
<tr>
<td>TB8 -</td>
<td>0</td>
</tr>
<tr>
<td>RB8 -</td>
<td>0</td>
</tr>
<tr>
<td>TI -</td>
<td>0</td>
</tr>
<tr>
<td>RI -</td>
<td>0</td>
</tr>
</tbody>
</table>

- SM0 - Serial port mode bit 0 is used for serial port mode selection.
- SM1 - Serial port mode bit 1.
- SM2 - Serial port mode 2 bit, also known as multiprocessor communication enable bit. When set, it enables multiprocessor communication in mode 2 and 3, and eventually mode 1. It should be cleared in mode 0.
- REN - Reception Enable bit enables serial reception when set. When cleared, serial reception is disabled.
- TB8 - Transmitter bit 8. Since all registers are 8-bit wide, this bit solves the problem of transmitting the 9th bit in modes 2 and 3. It is set to transmit logic 1 in the 9th bit.
- RB8 - Receiver bit 8 or the 9th bit received in modes 2 and 3. Cleared by hardware if 9th bit received is logic 0. Set by hardware if 9th bit received is logic 1.
- TI - Transmit Interrupt flag is automatically set at the moment the last bit of one byte is sent. It's a signal to the processor that the line is available for a new byte transmit. It must be cleared from within the software.
- RI - Receive Interrupt flag is automatically set upon one byte receive. It signals that byte is received and should be read quickly prior to being replaced by a new data. This bit is also cleared from within the software.
Q 2 c) State the functions of:

(i) **Compiler**: It is a software program which translates high level language program into machine language. In compiling process the code used by the user is called the source code and the output from compiler is called object code.

(ii) **Debugger**: Debugging is the process of finding an error or a source bug in the code, which is initially not taken into account. With the help of debugger software we can discover the bugs after the testing and simulating the actions of the codes.

(iii) **Simulator**: Simulator is a software program which simulates the hardware performance of the software program written without actually employing a specific device or microcontroller. It allows a developer to run a program designed for one type of machine (Target machine) on another machine (development machine). It simulates the running conditions of the target machine on the development machine. With the help of simulator we can step through the code while the program is running and also change the parts of code for getting different solutions for the problem. Disadvantage is simulation is they do not support real interrupts or devices. Thus real time results are not observed.

(iv) **Emulator**: Emulator is also called as In- circuit Emulator(ICE). It is the Hardware which replaces the microprocessor and microcontroller of interest in plug-in socket like a simulator, Emulator also facilitates you to control the program execution and monitoring at different steps of program. Emulators are expensive.
Q 2 d) Draw interfacing diagram of LCD display and explain the functions of RS, EN, R/W.

Ans.: RS (register select)

- There are two very important registers inside LCD. The RS pin is used for their selection as follows.
- If RS=0, the instruction command code is selected, allowing the user to send the command such as clear display, cursor at home, etc.
- If RS=1, the data register is selected, allowing the user to send data to be displayed on the LCD.

R/W (read/write)

- R/W input allows the user to write information to the LCD or read information from it.
- R/W = 1 when reading; R/W = 0 when writing.

EN (enable)

- The enable pin is used by the LCD to latch information presented to its data pin.
- When data is supplied to its data pin, a high-to-low pulse must be applied to this in order for the LCD to latch in the data presented at the data pins.
- This pulse must be a minimum of 450 ns wide.

-------- 3 Mark for correct pin description and 1 mark for circuit diagram
Q 2 e) State the methods of Task synchronization and describe any one in detail.

Ans :-

Method of task synchronizations are

- semaphores
- message queues,
- Mutual Exclusion
- Dead Lock
- Mailboxes
- Message Queues
- Tasks, along with task-management services, allow developers to design applications for concurrency to meet multiple time constraints and to address various design problems inherent to real-time embedded applications.

---- 1 Mark for any four methods mentioned

Sample Description of one method given below

**Deadlock**

- Deadlock is the situation in which multiple concurrent threads of execution in a system are blocked permanently because of resource requirements 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 resource 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.

--- 3 Mark for correct description of any one method with suitable example
Q3. Attempt any Four of the following:

Q 3 a) Explain alternate pin functions of port 3.

Ans:-

<table>
<thead>
<tr>
<th>Port Pins</th>
<th>Alternate Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3.0</td>
<td>RXD (Serial Input Port)</td>
</tr>
<tr>
<td>P3.1</td>
<td>TXD (Serial Output Port)</td>
</tr>
<tr>
<td>P3.2</td>
<td>INT 0 (External Interrupt 0)</td>
</tr>
<tr>
<td>P3.3</td>
<td>INT 1 (External Interrupt 1)</td>
</tr>
<tr>
<td>P3.4</td>
<td>T0 (Timer/Counter 0 External input)</td>
</tr>
<tr>
<td>P3.5</td>
<td>T1 (Timer/Counter 1 External input)</td>
</tr>
<tr>
<td>P3.6</td>
<td>WR (External Data Memory Write strobe)</td>
</tr>
<tr>
<td>P3.7</td>
<td>RD (External Data Memory Read strobe)</td>
</tr>
</tbody>
</table>

--- 4 Mark for correct format and description

Q 3 b) Draw and explain architecture of ARM 7 Processor.

Ans: Architecture of ARM7

Reference book Embedded System by Raj Kamal
ARM – Advance Risk Machine

ARM 7 microcontrollers is designed by ARM corporation which is a widely used complex system.

ARM 7 is a combination of both RISK and CISC facility.

It is having RISK core which compiles the CISC instructions into RISK format and then incomplete then RISK core of microcontroller.

It is having same addressing modes such as index, auto index.

On chip memory of 16 kb

Data and Cache of each 16 kb

Physical capacity of C-programmed is 4 GB

Internal Bus width of 32 bits

Due to less power requirement, it is widely used in Mobile Corporation where power is biggest criteria.

Q 3 c) Explain the software used in processor specific assembly language and high level language

Ans: software used in processor specific assembly language

**Editor:** It is a software module used for writing assembly, mnemonics or c codes using keyboard of the host computer system (Development Machine). It provides the facilities of entry, addition, deletion, merging, storing of files etc and creates a source file.

**Assembler:** Assembly language programs are not directly executable, they must be translated into machine language. Assembler is a software program which carries out this translation of assembly mnemonics into binary opcodes and generates executable binary file. It also creates a list file which consists of addresses, source code (assembly language mnemonics), and hex codes (object codes). The list file is printable. The source hex file generated by assembler is essential to program (or burn) the microcontroller device.

**Dissembler:** It is a software program which translates the object codes into the mnemonics form of assembly language. With this we can understand the previously made object code.

software used in processor specific high level language

**Compiler:** It is a software program which translates high level language program into machine language. While translating program instructions written in language such as C, the translation makes use of three components which are
Subject code: 12269
1. Preprocessor
2. Compiler
3. Linker

In compiling process the code used by the user is called the source code and the output from compiler is called object code.

Source code → Compiler → object code

**Linker and loader**: Linker creates an absolute object module. It links the needed object code files and library code files. (The recently modified development machines are very powerful and sometimes the use of a linker is no longer absolutely necessary). It is done before the loader relocates the addresses and puts the codes at the physical address in the memory and the program runs. The difference between loader and locator is loader puts the codes at physical memory addresses in host machine and locator puts it on a target machine with the help of a device programmer.

**Cross compiler**: It is a compiler that creates binary executable files for the target system processor. In short, it runs on one type of computer and produces machine code for a different type of computer (or processor on target machine)

**Library and Library Manager**: It is a specially ordered and formatted program collection of object codes. This collection can be used during development of different source files by the linker. Library manager helps in creating new libraries and also make the effective use of the existing library files.

**Simulator**: Simulator is a software program which simulates the hardware performance of the software program written without actually employing a specific device or microcontroller. It allows a developer to run a program designed for one type of machine (Target machine) on another machine (development machine). It simulates the running conditions of the target machine on the development machine.

**Debugger**: Debugging is the process of finding an error or a source bug in the code, which is initially not taken into account. With the help of debugger software. We can discover the bugs after the testing and simulating the actions of the codes the testing and simulating the actions of the codes.

--- 4 Mark for correct description of any four points
Q 3 d) Write an assembly language program to generate a 4 step sequence for a 4-phase stepper motor.

Ans: -

```
ORG 0000h
MOV A,#00110011b ; 4 step sequence for a 4-phase stepper motor.
MOV R0,#200

CLOCKWISE:
MOV P1,A ; Sequence output at Port 1
RR A ; Sequence Rotation in one direction
ACALL DELAY
DJNZ R0, CLOCKWISE
MOV R0,#200

ANTICLOCKWISE:
MOV P1,A ; Sequence Rotation in other direction
RL A
ACALL DELAY
DJNZ R0, ANTICLOCKWISE
MOV R0,#200
SJMP CLOCKWISE
```

```
                        ; The delay routine
     ;------------------------------------------------------------------------;
     ;------------------------------------------------------------------------;
DELAY: MOV R1,#255
LOOP_OUTER: MOV R2,#255
DJNZ R2,$
DJNZ R1, LOOP_OUTER
RET
END
```

----- 4 Mark for any correct Program

“C” Language program for a 4 step sequence for a 4-phase stepper motor.

```
#include<reg51.h>
main()
{
    void delay(void);
```
Subject code: 12269

Model Answer

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 Mark for any correct Program

Q 3 e) Draw the circuit diagram to interface DAC to 8051

Ans:- DAC is interfacing:

Interface DAC (Digital to Analog converter) to 8051.
Q4 a) Attempt any THREE of the following:

Q4 a i) With Suitable figure explain the CAN bus interface with signal.

Ans:- Out of the scope of syllabus but students can be assessed for the CAN bus data frame for 4 mark
Q 4 a ii) Draw the interfacing diagram for seven segment display

--- 4 Mark for correct description of any four points
Q 4 a iii) Generate a square wave with an ON and OFF time of 3 msec on all pins of port 1. Assume crystal frequency of 12 MHz. Write a program in assembly language.
Subject code: 12269

Ans:-

Delay Calculation:-

Crystal Frequency = 12 MHz

Timer clock frequency = crystal frequency X 1/12 = 12 MHz/12 = 1MHz

Timer period = 1/1 MHz = 1 μsec.

The tON and tOFF to be generated is 3 msec.

Count calculation= required delay / Timer Period

= 3 msec / 1 μsec.

=3000

The count to be loaded in timer register

= (65536 – 3000) d

= (62536) d

= (F448) H

TH1= (F4) H

TL1= (48) H

Program to generate square wave.

```assembly
MOV TMOD, #01H;       // Timer 1 in mode 1
AGAIN: CLR TF1;       // Clear timer 1 overflow flag
    CLR A;            // Clear Accumulator
    MOV P1, A;        // Move acc. Data to port 1
    MOV TL1, #48H;    // Move 48H data to TL1
    MOV TH1, #0F4H;   // Move 0F4H Data to TH1
    SETB TR1;         // Set timer 1
    JNB TF1, AGAIN;   // Clear Timer 1

CPL A;                // Complement accumulator
MOV P1, A             // Move acc. Data to port 1
```


Q 4 a iv) Describe Serial mode of 8051.

As seen, serial port mode is selected by selecting the SM0 and SM2 bit combination of SCON SFR as shown below:

<table>
<thead>
<tr>
<th>SM0</th>
<th>SM1</th>
<th>MODE</th>
<th>DESCRIPTION</th>
<th>BAUD RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8-bit Shift Register</td>
<td>1/12 the quartz frequency</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>8-bit UART, variable data rate</td>
<td>Determined by the timer 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>9-bit UART, fixed data rate</td>
<td>1/32 the quartz frequency (1/64 the quartz frequency)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>9-bit UART, variable data rate</td>
<td>Determined by the timer 1</td>
</tr>
</tbody>
</table>

**Mode 1:** It is full duplex mode. Here 1 is transmitted when there is change in data, otherwise zero is transmitted. This is referred as non return to zero (NRZ) operation. In this mode 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and stop bit (1). On receive; the stop bit goes into RB8 in Special Function Register SCON. The baud rate variable.

**Mode 2:** 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (1). On transmit, the 9th data bit (TB 8 in SCON) can be assigned the value of 0 or 1. Or, for example the parity bit (P in the PSW) could be move into TB8. On receive, the 9 data bits goes into RB8 in special Function register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

**Mode 3:** 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). In fact, Mode 3 is same as Mode 2 in all respect except baud rate. The baud rate is in Mode 3 is variable.
Subject code: 12269

Model Answer

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition $R1=1$ and $REN=1$. Reception is initiated in the other modes by the incoming start bit is $REN = 1$.

--- 4 Mark for correct relevant description

Q 4 b) Attempt any ONE of the following:

Q 4 b (i) State any three features of device programmer and in circuit emulator.

Ans:

1. Device programmer downloads the binary program into the microcontroller from development processor memory into target board processor memory.
2. It has the ability to read the program from the target machine memory
3. It provides the facility to erase on chip memory of target machine
4. Target machine memory blank check facility
5. It can run and test the target machine program in real time.

--- 3 Mark for any three correct features

1. In circuit Emulator is a hardware that ideally behaves like real microcontroller chip with all functionalities integrated
2. It’s a powerful debugging tool working in real time
3. It is faster than simulator and allows checking the contents of register and memory
4. It enables the programmer to control and monitor the embedded system program

--- 3 Mark for any three correct features

Q4 b ii) Draw the labeled diagram of interfacing of ADC with 8051 and write a program in assembly language.

ADC 0808/0809 INTERFACED WITH 8051 MICROCONTROLLER
Subject code: 12269

**Model Answer**

--- 3 Mark for correct Figure (The answer of students showing clk signal directly should also be given full mark)

Programming ADC 0808/0809 in Assembly:

- **ALE** BIT P2.4 ; assigning ALE, OE, SC, SOC signals to port pins
- **OE** BIT P2.5
- **SC** BIT P2.6
- **EOC** BIT P2.7
- **ADDR_A** BIT P2.0 ; selection of channel
- **ADDR_B** BIT P2.1
- **ADDR_C** BIT P2.2
- **MYDATA** EQU P1
Subject code: 12269

Model Answer

ORG 0H

MOV MYDATA,#0FF ;make p1 input

SETB EOC ;make EOC input

CLR ALE ;clear ALE

CLR SC ;clear SC

CLR OE ;clear OE

BACK:

CLR ADDR_C ;C=0

CLR ADDR_B ;B=0

CLR ADDR_A ;A=0

ACALL DELAY ;make sure addr is stable

SETB ALE ;latch address

ACALL DELAY ;delay for fast DS89C4X0 chip

SETB SC ;start conversion

ACALL DELAY

CLR ALE

CLR SC

HERE:

JB EOC,HERE ;wait until done

HERE1:

JNB EOC,HERE1 ;wait until done

SETB OE ;enable RD

ACALL DELAY ;wait

MOV A, MYDATA ;read data

CLR OE ;clear RD for next time

ACALL CONVERSION ;hex to ASCII

ACALL DATA_DISPLAY ;display data

SJMP BACK

--- 3 Mark for Correct assembly program (Correct program in C can also be given 3 mark)
Q 5 a) Draw interface diagram of LCD interfacing and write a program to send information “WELCOME” to LCD

Ans: Diagram of LCD interfacing

This program displays “WELCOME” on LCD

```
ORG 0000h

LCDDISP: ; programming of command register

MOV A,#3Ch
ACALL COMMAND

MOV A,#0Eh
ACALL COMMAND

MOV A,#06h
ACALL COMMAND

MOV A,#01h
ACALL COMMAND

MOV A,#87h
ACALL COMMAND

MOV A,#'W'
ACALL COMMAND
```

4 mark for correct diagram and labels
Subject code: 12269

ACALL DISPLAY

MOV A,#'E'
ACALL DISPLAY

MOV A,#'L'
ACALL DISPLAY

MOV A,#'C'
ACALL DISPLAY

MOV A,#'O'
ACALL DISPLAY

MOV A,#'M'
ACALL DISPLAY

MOV A,#'E'
ACALL DISPLAY

HERE: SJMP HERE

; LCD strobe subroutine

COMMAND:

ACALL READY

MOV P1,A
CLR P3.3
CLR P3.4
SETB P3.5
CLR P3.5
RET

DISPLAY: ; Routine for display

ACALL READY

MOV P1,A
SETB P3.3
Subject code: 12269

Model Answer

CLR     P3.4
SETB    P3.5
CLR     P3.5
RET

READY: CLR     P3.5
MOV     P1,#0FFh
CLR     P3.3
SETB    P3.4

WAIT:
CLR     P3.5
SETB    P3.5
JB      P1.7,WAIT
CLR     P3.5
RET
END

NOTE: Correct Program in C language can also be assessed for 4 marks

Q 5 b) Write the steps executed by microcontroller on activation of interrupt.

Ans:-

1. Microcontroller finishes the instruction it is executing and saves the address of the next instruction (PC) on the stack
2. Microcontroller also saves the current status of all the interrupts internally (i.e. not on the stack)
3. Microcontroller jumps to a fixed location in memory, called the interrupt vector table that holds the address of the ISR
4. The microcontroller gets the address of the ISR from the interrupt vector table and jumps to it and starts to execute the interrupt service subroutine until it reaches the last instruction of the subroutine.
5. Upon executing the ISR, the microcontroller returns to the place where it was interrupted. First, it gets the program counter (PC) address from the stack by popping the top two bytes of the stack into the PC. Then it starts to execute from that address.

--------- 8 mark for correct steps enlisted above
Subject code: 12269

Q 5 c) Interface 4 X 4 matrix keyboard with 8051 microcontroller ports. Draw and explain interfacing diagram and flowchart.

Interface 4 X 4 matrix keyboards with 8051 microcontroller

ports

--------4 Mark for correct Diagram

Ans: - flowchart 4 X 4 matrix keyboard
Q 6 Attempt any FOUR of the following:

Q6 a) Write the steps to program timer in mode 2 and write a program to generate a square wave of frequency 1 KHz on pin P1.2. Assume crystal frequency of 22 MHz

Ans: Steps to program mode 2:

To generate a time delay using the timer’s mode 2, take the following steps.

1. Load the TMOD value register indicating which timer (Timer 0 or Timer 1) is to be used, and select the timer mode (mode 2)
2. Load the TH registers with the initial count value.
3. Start the timer.
4. Keep monitoring the timer flag (TF) with the “JNB TFx, target “instruction to see whether it is raised. Get out of the loop when TF goes high.
5. Clear the TF flag.
6. Go back to step 4, since mode 2 is auto reload.

--- 2 Mark for Correct steps

MOV TMOD, #20H
MOV TH1, #1BH
SETB TR1
AGAIN: MOV R2, #04
BACK: JNB TF1, BACK
CLR TF0
DJNZ R2, BACK
CPL 1.2
SJMP AGAIN

--- 2 Mark for Correct Program

NOTE: Correct C language program also carries 2 mark
Q 6 b) State any four features of I2C and CAN Bus Protocol.

Ans: - Features of I2C

1. Inter IC control is used to established a mutual network between the ICs in a device circuit.
2. Its high speed serial bus protocol
3. It is single cable connectivity with two wires Serial Data (SDA) and Serial Clock (SCL) that are connected to each device.
4. Data transfer rate is 100Kbps to 400 Kbps.

--- 2 Mark for four features

Features of CAN Bus Protocol

1. Control Area Network is used to create a network between the Embedded System that are located at different places.
2. It is a high speed Serial Bus Protocol that uses twisted pair of wires only for data.
3. It works at data link and physical layer.
4. It has multi-master and multi-feature capacity.
5. Data rate is upto 1Mbps

--- 2 Mark for four features

Q 6 c) Draw and explain PCON

Ans.

PCON : Power Control Register (Not Bit Addressable)

<table>
<thead>
<tr>
<th>SMOD</th>
<th>PCON.7</th>
<th>PCON.6</th>
<th>PCON.5</th>
<th>PCON.4</th>
<th>GF1</th>
<th>GF0</th>
<th>PD</th>
<th>IDL</th>
</tr>
</thead>
</table>
| SMOD | PCON.7 | Double baud rate bit. If SMOD = 1, the baud rate is doubled when the serial part is used in mode 1, 2 and 3.
| -    | PCON.6 | Not implemented, reserved for future use.
| -    | PCON.5 | Not implemented, reserved for future use.
| -    | PCON.4 | Not implemented, reserved for future use.
| GF1  | PCON.3 | General purpose bit.
| GF0  | PCON.2 | General purpose bit.
| PD   | PCON.1 | Power Down bit. If set, the oscillator is stopped. A reset or an interrupt (83C154 and 83C154D only) can cancel this mode (Note 1).
| IDL  | PCON.0 | IDLE bit. If set the activity CPU is stopped. A reset or an interrupt can cancel this mode (See Note 1).

--- 4 Mark

Q 6 d) With Suitable example explains the concept of device drivers.

Ans.:- The device driver is software for controlling, sending and receiving a byte or stream of bytes from or to a device. In case of physical devices, a driver uses the hardware status flag and control
register bits that are in set and reset states. In case of virtual devices also, a driver uses the status and control words and the bits exist in set and reset status.

Driver controls three functions:

1. Initializing that is activated by placing appropriate bits at the control register or word.
2. Calling an ISR or interrupt or on setting a status flag in the status register and drive or run the ISR. These ISR is also called Interrupt Handler Routine.
3. Resetting the status flag after interrupt service.

A driver may be designed for asynchronous operation (multiple use by tasks one after another) or synchronous operations (concurrent use by the task). This is because the device may get activated when an interrupt arises and the device driver routine, services that.

Using OS function, a device driver coding can be made to hide the underlying hardware. An API (Application Programming Interface) then defines the hardware separately. This makes the driver usable when device hardware changes in the system.

A device driver accesses a parallel port or a serial port, keyboard, mouse, disc, network, display, file, pipe and socket at specific addresses. And OS may also provide device drivers codes for the system port addresses and for the acces mechanism (read, save and write) for the device hardware.

--- 4 Mark

Q 6 e) Describes the need of RTOS in an Embedded System and state any two specifications of RTOS.

Ans. Necessity of RTOS for embedded system can be decided based on following points:-

1. Scheduling, Synchronization: RTOS provides effective scheduling and synchronization techniques which enables deterministic behavior.
2. Fast Execution: it provides running the user threads in Kernel space so that they execute fast. RTOS also provides faster memory allocation.
3. Task Priority: Pre-emption, priority in heritance takes care of task priorities.
4. Short-interrupt latency and deadlines: Interrupt latency = hardware delay to get interrupt signal to the processor + time to complete the current instruction + time executing system code in preparation for transferring execution the device interrupt handler.

--- 2 Mark for Necessity

Specification:

1. Predictability: RTOS offers a predictability response to system input within the given time frame (deadline)
2. Reliability: RTOS is available all the times and it does not fail. RTOS offers reliable performance provided the embedded hardware, BSP (Board Support Package) and application code do not fail.
3. Compactness: RTOS are designed very critically with the appropriate consideration of resources and memory. It offers small, portable, compact, low cost and versatile programming flexibility and application.
**4. Performance:** it offers the most satisfactory performance on the basis of time, resource management, ruggedness, development flexibility and compactness of criteria.

**5. Scalability:** in order to suit and fulfill the application demand, the RTOS is capable of adding or deleting modular component of functionality with this feature, only the necessary function becomes a part of the application and memory is also reduce.

---- 2 Mark for 2 points

**Q 6 f) Describe the concept of multitasking.**

**Ans.**

1. Tasks are concurrent and independent threads of execution.
2. Multitasking is the ability of the operating system to handle multiple tasks within set deadlines.
3. A real-time kernel might have multiple tasks that it has to schedule to run. One such multitasking scenario is illustrated in following diagram.
4. In this scenario, the kernel multitasks in such a way that many threads of execution appear to be running concurrently; however, the kernel is actually interleaving executions sequentially, based on a preset scheduling algorithm.
5. The scheduler must ensure that the appropriate task runs at the right time.
There are two types of multitasking:

- Pre-emptive multitasking
- Non pre-emptive multitasking

---4 Mark for above description