School Of Information Technology and Engineering

October, 2015

EMBEDDED SYSTEMS LAB RECORD

(ITE 306)

of

B.TECH

in

Information Technology

by

<ANAND UJWAL>-(13BIT0088)

1)Write an ALP & Embedded C program to transmit a letter "M" continuously at a baud rate of 9600,8 bit data and 1 bit stop and start bit, using Timer 1 in mode 2.

ALP

mov scon, #050h

mov tmod, #020h

mov th1, #0feh

setb tr1

again:

mov sbuf, #'m'

loop: jnb ti, loop

clr ti

sjmp again

end

C program

```c
#include<stdio.h>

#include<regx51.h>

int main()

{

SCON=0x050;

TMOD=0x020;

TH1=0x0fe;

TR1=1;

while(1)

{

SBUF='m';

while(TI==0);

TI=0;

}
```

}

2)Write an ALP & Embedded C program to transmit a word "MESSAGE" continuously at a baud rate of 4800, with an oscillator frequency of 11.0592 Mhz,8 bit data and 1 bit stop and start bit, using Timer 1 in mode 1

ALP

```
mov scon, #050h

mov tmod, #020h

mov th1, #0feh

setb tr1

again:

mov sbuf, #'M'

acall loop

mov sbuf, #'E'

acall loop

mov sbuf, #'S'
```

```
acall loop

mov sbuf, #'S'

acall loop

mov sbuf, #'A'

acall loop


mov sbuf, #'G'

acall loop

mov sbuf, #'E'

acall loop

sjmp again

loop: jnb ti, loop

clr ti

ret

end
```

C program

```c
#include<stdio.h>
```

```c
#include<regx51.h>

void fun(unsigned char x)

{SBUF=x;

while(TI==0);

TI=0;

}

int main()

{

SCON=0x050;

TMOD=0x020;

TH1=0x0fe;

TR1=1;

while(1)

{

 fun('M');

 fun('E');

 fun('S');

 fun('S');

 fun('A');
```

fun('G');

fun('E');

}

3)Write an ALP & Embedded C program   to receive a letter at a baud rate 2400, with an oscillator frequency of 12 MHz, 8 bit data and 1 bit stop and start bit mode , using Timer 1 in mode 0.Simultaneously send the received byte to port3.

ALP

mov scon, #050h

mov tmod, #020h

mov th1, #0feh

setb tr1

again:

mov a,sbuf

mov p3,a

mov sbuf, a

loop: jnb ri, loop

clr ri

sjmp again

end

C program

#include<stdio.h>

#include<regx51.h>

int main()

{unsigned char x;

SCON=0x050;

TMOD=0x020;

TH1=0x0fe;

TR1=1;

while(1)

{

x=SBUF;

P3=x;

SBUF=x;

while(RI==0);

RI=0;

}

}

3)C program to AND 8b data of port 0 and port 1 and to send result to port 2

```
#include<stdio.h>
#include<regx51.h>

int main()
{
P0=0x01;
P1=0x03;
P3=P0 & P1;



}
```

4)C Program to AND P0.0 and P1.3 send result to P2.0

```
#include<stdio.h>
#include<regx51.h>

int main()
{
P0=0x01;
```

```
P1=0x03;
P2_0=P0_0 & P1_3;



}
```

5)C Program to read p1 and send data to p2 and p3 based on condition

```
#include<stdio.h>
#include<regx51.h>

int main()
{
unsigned int x;
P1=0x01;
x=P1;
if(x%2==0)
P2=x;
else
P3=x;



}
```

6)C Program to left shift data at port 1 repetitively

```
#include<stdio.h>
#include<regx51.h>
void delay(unsigned int x)
{    unsigned int i,j;
```

```
for(i=0;i<x;i++)
for(j=0;j<120;j++)
{}
}
int main()
{
unsigned int x;
P1=0x06;
x=P1;
while(1)
{
 P1=P1<<1;
 delay(1000);
}
}
```

7)C program to send 0-9 to port 2

```
#include<stdio.h>
#include<regx51.h>
void delay(unsigned int x)
{     unsigned int i,j;
for(i=0;i<x;i++)
for(j=0;j<120;j++)
{}
}
int main()
{
unsigned int x;
for(x=0;x<=9;x++)
{P2=x;
delay(1000);
```

```
}
}
```

8)C Program to send hex data to port 0

```c
#include<stdio.h>
#include<regx51.h>

int main()
{
P0=0x01;
P1=0x03;
P2_0=P0_0 & P1_3;



}
```

9)C Program to send ASCII value of the characters

```c
#include<stdio.h>
#include<regx51.h>
void delay(unsigned int x)
{     unsigned int i,j;
for(i=0;i<x;i++)
for(j=0;j<120;j++)
{}
}
int main()
{
unsigned char x='a';
```

```
{P2=x;

}
}
```

10)C Program to toggle LEd's at port 1

```c
#include<stdio.h>
#include<regx51.h>
void delay(const unsigned int x)
{     unsigned int i,j;
for(i=0;i<x;i++)
for(j=0;j<120;j++)
{}
}
int main()
{
while(1)
{
P0=0x01;
delay(20000);
P0=0x03;   }

}
```

11)C program to toggle to alternate bits of port 1

```c
#include<stdio.h>
#include<regx51.h>
void delay(const unsigned int x)
{     unsigned int i,j;
for(i=0;i<x;i++)
```

```
for(j=0;j<1275;j++)
{}
}
int main()
{
while(1)
{
P0=0xaa;
delay(600);
P0=0x55;   }



}
```

12)C Program to toggle LSB bit of Port 1

```
#include<stdio.h>
#include<regx51.h>
void delay(const unsigned int x)
{     unsigned int i,j;
for(i=0;i<x;i++)
for(j=0;j<120;j++)
{}
}
int main()
{
P1=0x00;
while(1)
{
P1_0=1;
delay(20000);
P1=0x00;   }
```

}

13)C program to toggle MSB bit of port 1

```c
#include<stdio.h>
#include<regx51.h>
void delay(const unsigned int x)
{     unsigned int i,j;
for(i=0;i<x;i++)
for(j=0;j<120;j++)
{}
}
int main()
{
P1=0x00;
while(1)
{
P1_7=1;
delay(20000);
P1=0x00;   }


}
```

14)C program to implement Traffic Control Signal.

```c
#include<regx51.h>
void delay()
{
unsigned int i,j;
for(i=0;i<1000;i++)
for(j=0;j<10000;j++)
{
}
}
void red()
{
P0_0=1;
P0_4=0;
P0_7=0;
delay();
}
void yellow()
{
P0_0=0;
P0_4=1;
P0_7=0;
delay();
}
void green()
{
P0_0=0;
P0_4=0;
P0_7=1;
delay();
}
```

```c
int main()
{
    P0=0x00;
while(1)
{
red();
yellow();
green();
yellow();

}
}
```
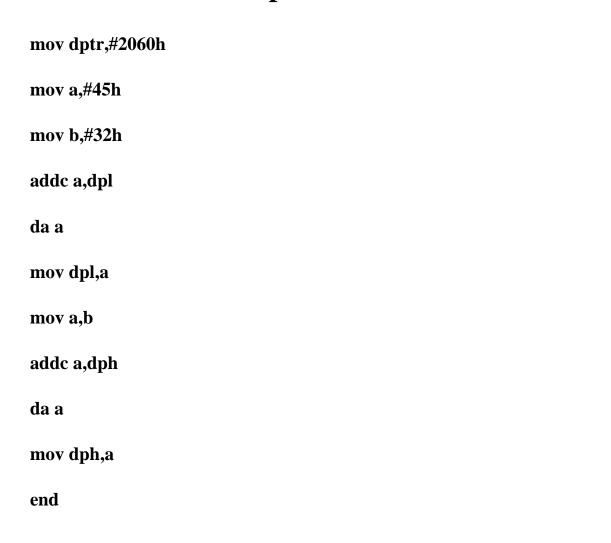
15)C program to implement WaterLevel Detector.

```c
#include<regx51.h>
#include<Math.h>
void delay()
{
unsigned int i,j;
for(i=0;i<1000;i++)
for(j=0;j<10000;j++)
{
}
}
int main()
{
unsigned int i;
P1=0x00;
P2=0x00;
for(i=0;i<9;i++)
{
```

```
P1=pow(2,i)-1;
delay();


}
if(P1_7==1)
{
P2_0=1;
while(1);
}
}
```

(Cycle Sheet 1)

Write an ALP to perform 16 bit BCD addition

mov dptr,#2060h

mov a,#45h

mov b,#32h

addc a,dpl

da a

mov dpl,a

mov a,b

addc a,dph

da a

mov dph,a

end

Write an ALP to perform BUBBLE SORT
(ASCENDING ORDER)

```
MOV DPTR,#1000H

MOV R6,#04H

OUTER:

MOV DPL,#00H

MOV R7,#04H

INNER:

MOV R0,DPL

MOVX A,@DPTR

MOV B,A

INC DPTR

MOV R1,DPL

MOVX A,@DPTR

CJNE A,B,LB

LB:

JNC LABEL

MOV DPL,R0

MOVX @DPTR,A

MOV DPL,R1

MOV A,B
```

MOVX @DPTR,A

LABEL :DJNZ R7,INNER

DJNZ R6,OUTER   END

Write an ALP to perform complement of a number

MOV A,#08H

CPL A

INC A

END

Write an ALP to perform GCD operation by
EULERS Algorithm

MOV A,#18H

MOV B,#10H

LABEL:

MOV R0,B

MOV R7,A

DIV AB

MOV A,B

CJNE A,#00H,LOOP

MOV A,R0

JMP EN

LOOP:

MOV A,R0

JMP LABEL

EN:

END

Write an ALP to find factorial of a number

MOV R0,#07H

MOV A,#01H

AGAIN:

MOV B,R0

MUL AB

DJNZ R0,AGAIN

END

Write an ALP to find FIBONACCI SEQUENCE

MOV DPTR,#1000H

MOV R0,#0AH

MOV A,#00H

MOV R1,A

MOVX @DPTR,A

INC DPTR

MOV A,#01H

MOVX @DPTR,A

INC DPTR

LOOP:

MOV R2,A

ADDC A,R1

MOVX @DPTR,A

INC DPTR

MOV B,R2

MOV R1,B

DJNZ R0,LOOP

END

Write an ALP to perform INSERTION SORT
for 10 numbers in ascending order

```
        MOV DPTR,#1000H

        MOV R5,#01H

        MOV R6,#09H

OUTER:

        MOV DPL,R5

        MOVX A,@DPTR

        MOV B,A

        MOV A,R5

        MOV R7,A

INNER:

        DEC R7

        MOV DPL,R7

        MOVX A,@DPTR

        CJNE A,B,LB

LB:

        JC LBA

        INC R7

        MOV DPL,R7

        MOVX @DPTR,A
```

```
    MOV A,B

    DEC R7

    MOV DPL,R7

    MOVX @DPTR,A

    INC R7

    DJNZ R7,INNER
LBA:

    INC R5

        DJNZ R6,OUTER

    END
```

Write an ALP to find max of 10 nos

```
MOV DPTR,#1000H

MOV R0,#0AH

MOV B,#00H

LOOP:

    MOVX A,@DPTR

        CJNE A,B,LB

        LB:

        JC LBA

        MOV B,A

LBA:

    INC DPTR

        DJNZ R0,LOOP

        END
```

Write an ALP to find MINIMUM OF 10 NOS

MOV DPTR,#1000H

MOV R0,#0AH

MOV B,#9FH

LOOP:

   MOVX A,@DPTR

      CJNE A,B,LB

      LB:

      JNC LBA

      MOV B,A

LBA:

   INC DPTR

      DJNZ R0,LOOP

      END

Write an ALP to move a data block from one memory to another

MOV DPTR,#2000H

MOV R0,#40H

MOV B,#0AH

AGAIN:

    MOV A,@R0

        MOVX @DPTR,A

        INC DPTR

        INC R0

        DJNZ B,AGAIN

        END

Write an ALP to perform selection sort for 10 values

MOV DPTR,#1000H

MOV R5,#00H

MOV R6,#09H

OUTER:

MOV DPL,R5

MOV A,R6

MOV R7,A

MOVX A,@DPTR

MOV B,A

MOV R3,DPL

INNER:

INC DPTR

MOVX A,@DPTR

CJNE A,B,LB

LB:

JNC LBA

MOV R3,DPL

MOV B,A

LBA:

DJNZ R7,INNER

```
        MOV DPL,R5

        MOVX A,@DPTR

        MOV R0,A

        MOV A,B

        MOVX @DPTR,A

        MOV DPL,R3

        MOV A,R0

        MOVX @DPTR,A

        INC R5

        DJNZ R6,OUTER

        END
```

Write an ALP to perform sum of first 15 natural nos

MOV R0,#0fH

MOV A,R0

INC R0

MOV B,R0

MUL AB

MOV B,#02H

DIV AB

END