VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)

School Of Information Technology and Engineering

October, 2015

EMBEDDED SYSTEMS LAB RECORD

(ITE 306)

of

B.TECH

in

Information Technology

by

Shivam Sharma(13BIT0032)

FACULTY- PROF. ASHA JERLIN

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

```
#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;
```

}

}

OUTPUT: HHHHHH…..

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

```
#include<stdio.h>

#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');


}
```

OUTPUT: MESSAGEMESSAGEMESSAGE….

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;

}

}
```

INPUT: P
OUTPUT: P3: 0X50

4)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;
```

```
}
```

OUTPUT: P0: 0X01
P1: 0X03
P2: 0X01

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

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

int main()
{
P0=0x02;
P1=0x03;
P2_0=P0_0 & P1_3;



}
```

OUTPUT: P0: 0X02
P1: 0X03
P2: 0XFE

6)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;



}
```

OUTPUT: P1:0X01

P2:0XFF

P3:0X01

7)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);
}
}
```

OUTPUT: P1:0X06…….0X00

8)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);
}
}
```

OUTPUT: P2:0X00…….0X09

9)C Program to send hex data to port 0

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

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



}
```

10)C Program to send ASCII value of the characters

```
#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++)
{}
}
main()

    {
    Int i=0;
```

```
P0='A';
For(i=0;i<26;i++)
{
P0=P0+1;
}
}
```

OUTPUT: P0:0X59

Pins:0x40

11)C Program to toggle LEd's at 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()
{
while(1)
{
P1=0x01;
delay(20000);
P1=0x03;   }


}
```

OUTPUT: P1:0X01

P1:0X03

12)C program to toggle to alternate bits 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<1275;j++)
{}
}
int main()
{
while(1)
{
P1=0xaa;
delay(600);
P1=0x55;   }


}
```

OUTPUT: P1:0XAA

P1:0X55

13)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_0=0;   }


}
```

OUTPUT: P1:0XFE

P1:0XFF

14)C program to toggle MSB 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_7=1;
delay(20000);
P1_7=0;   }


}
```

OUTPUT: P1:0X7F

P1:0XFF

14)C program to implement Traffic Control Signal.

```
#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();
}
int main()
{
    P0=0x00;
while(1)
{
red();
```

```
yellow();
green();
yellow();

}
}
```

OUTPUT: P0:0X01
P0:0X10
P0:0X80

15)C program to implement WaterLevel Detector.

```
#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);
}
}
```

OUTPUT: P0:0X00……..0XFF
P2:0X01

(Cycle Sheet 1)

Q.1 ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION

ADDITION:

```
MOV A,#05H
MOV B,#06H
ADD A,B
END
```

OUTPUT: a 0x0B

SUBTRACTION:

```
MOV A,#07H
MOV B,#05H
SUBB A,B
END
```

OUTPUT: a 0x02

MULTIPLICATION:

```
MOV A,#02H
MOV B,#05H
MUL AB
END
```

OUTPUT: a 0x0A

DIVISION:

```
MOV A,#0AH
MOV B,#05H
DIV AB
END
```

OUTPUT: a 0x02

Q.2 COMPLEMENT OF A NUMBER

```
MOV R1,#07H
MOV A,R1
CPL A
INC A
END
```

OUTPUT: a 0xF9

Q.3 MAXIMUM AMONG 10 NUMBERS

```
MOV DPTR,#1000H
MOV R1,#0AH
MOV B,#00H
AGAIN: MOVX A,@DPTR
        CJNE A,B,LABEL1
      SJMP LABEL2
LABEL1: JC LABEL2
        MOV B,A
LABEL2: INC DPTR
        DJNZ R1,AGAIN
        END
```

INPUT: X:0x001000: 06 01 05 09 08 02 00 03 04 07

OUTPUT: a 0x07
b 0x09

Q.4 16 BIT ADDITION

```
MOV DPTR,#0102H
MOV A,#21H
MOV B,#34H
ADDC A,DPL
MOV DPL,A
   MOV A,B
ADDC A,DPH
MOV DPH,A
END
```

OUTPUT: a 0x35

b 0x34

DPTR 0x3523

Q.5 MININUM OF 10 NUMBERS

```
MOV DPTR,#1000H
MOV R1,#0AH
MOV B,#99H
AGAIN:MOVX A,@DPTR
      CJNE A,B,LABEL1
      AJMP LABEL2
LABEL1:JNC LABEL2
       MOV B,A
LABEL2:INC DPTR
       DJNZ R1,AGAIN
       END
```

INPUT: X:0x001000: 06 02 10 09 05 01 04 0b 08 07

OUTPUT: a 0x07

b 0x01

Q.6 FIRST 10 FIBONACCI NUMBERS

```
MOV DPTR,#1000H
MOVX A,@DPTR
MOV R1,#0AH
MOV A,#00H
```

```
MOVX @DPTR,A
MOV B,#01H
INC DPTR
MOV A,B
MOVX @DPTR,A
MOV A,R1
SUBB A,#02H
MOV R1,A
MOV A,#00H
LOOP:MOV R2,B
        ADD A,B
        INC DPTR
        MOVX @DPTR,A
        MOV B,A
        MOV A,R2
        DJNZ R1,LOOP
        END
```

OUTPUT: X:0x001000: 00 01 01 02 03 05 08 0D 15 22

Q.7 FACTORIAL OF A NUMBER

```
MOV DPTR,#1000H
MOVX A,@DPTR
MOV R1,A
MOV A,#01H
LOOP: MOV B,R1
        MUL AB
        MOVX @DPTR,A
```

```
        INC DPTR
        DJNZ R1,LOOP
        END
```

INPUT: X:0x001000: 05

OUTPUT: X:0x001000: 05 14 3C 78 78

a 0X78

Q.8 8 BIT AND 16 BIT BCD ADDITION

PROGRAM(8-BIT):

```
MOV A,#03H
MOV B,#02H
ADD A,B
DA A
END
```

OUTPUT: a 0X11

PROGRAM(16-BIT):

```
MOV DPTR,#0102H
MOV A,#21H
MOV B,#34H
ADDC A,DPL
DA A
MOV DPL,A
```

```
MOV A,B
ADDC A,DPH
DA A
MOV DPH,A
END
```

OUTPUT: a 0x35
b 0x34
dptr 0x3523

Q.9 BUBBLE SORT

```
MOV R0,#09H
AGAIN:MOV DPTR,#2000H
      MOV R1,#09H
BACK:MOV R2,DPL
   MOVX A,@DPTR
   MOV B,A
   INC DPTR
   MOVX A,@DPTR
   CJNE A,B,NEXT
   AJMP SKIP
NEXT:JNC SKIP
   MOV DPL,R2
   MOVX @DPTR,A
   INC DPTR
   MOV A,B
   MOVX @DPTR,A
```

```
SKIP:DJNZ R1,BACK
    DJNZ R0,AGAIN
    END
```

Input: X:0x002000: 20 40 01 02 10 09 08 07 04

Output: X:0x002000: 01 02 04 07 08 09 10 20 40

Q.10 GCD OF 2 NUMBERS

```
MOV R2,#06H
MOV R1,#03H
MOV A,R2
MOV B,R1
L2:MOV R2,B
  DIV AB
  MOV A,B
  JZ NEXT
  MOV A,R2
  SJMP L2
NEXT:MOV A,R2
    END
```

OUTPUT: a 0x03

Q.11 DATA BLOCK TRANSFER FROM CODE MEMORY TO EXTERNAL MEMORY

```
   MOV DPTR,#1000H
```

```
  MOV R0,#40H
  MOV B,#0AH
  AGAIN:MOV A,@R0
        MOVX @DPTR,A
      INC DPTR
      INC R0
      DJNZ B,AGAIN
      END
```

INPUT: D:0x40: 02 01 05 03 06 08 04 09 0B 07

OUTPUT: X:0x1000: 02 01 05 03 06 08 04 09 0B 07