Embedded C programming

TASK 4

NAME : A G SRIKANTH

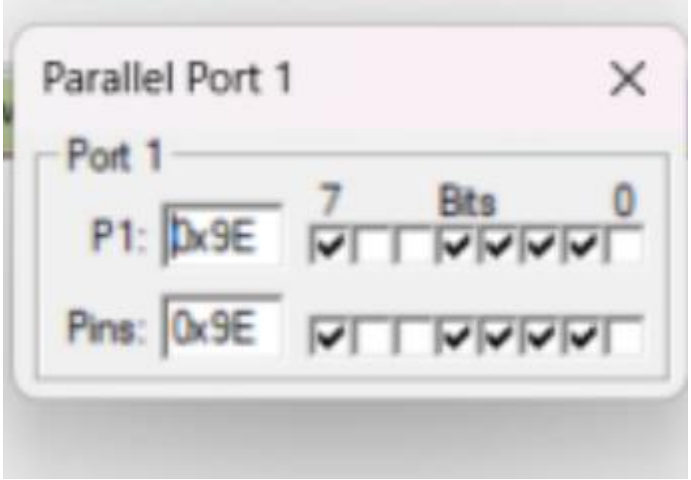
REG NO : 21BML0162

SLOT : L09+L10

FACULTY : DR VIDHYAPTHI C M

1. Using for loop to pass values into port:

```
#include<reg51.h>
//A G Srikanth
//21BML0162
void main(void)
{
unsigned char z;
P1=0x00;
for(z=0;z<=255;z++){
P1=z;
}
}
```

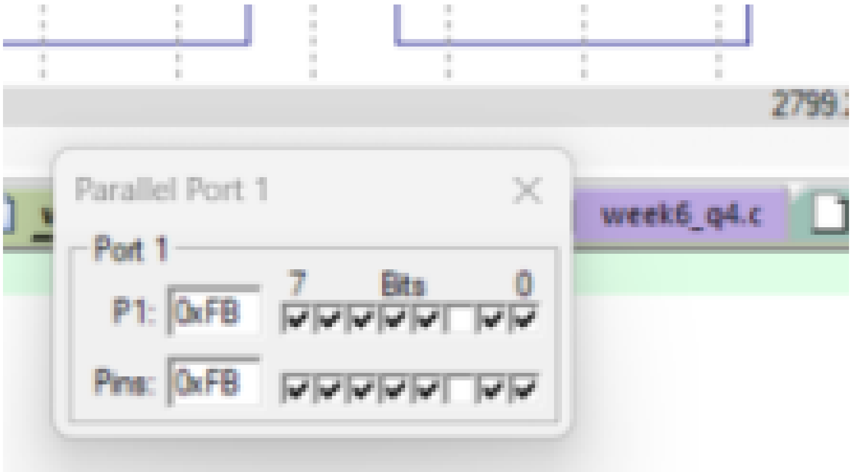


P1: 0x9E ☑☐☐☑☑☑☑☐

Pins: 0x9E ☑☐☐☑☑☑☑☐

2. Toggling a bit by generating delay using for loop

```
#include<reg51.h>

//A G Srikanth

//21BML0162

sbit mybit= P1^2;

void main(void)

{

unsigned int z;

mybit=0;

while(1)

{

mybit=1;

for(z=0;z<=10000;z++);

mybit=0;

for(z=0;z<=10000;z++);

}

}
```

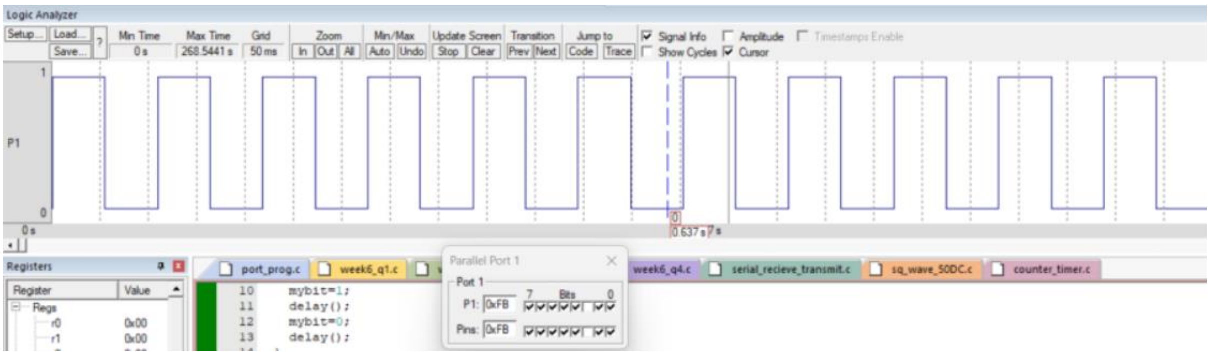


P1: 0xFB ☑ ☑ ☑ ☑ ☑ □ ☑ ☑

Pins: 0xFB ☑ ☑ ☑ ☑ ☑ □ ☑ ☑

3. Toggling a bit using a delay function

```
#include<reg51.h>

//A G Srikanth

//21BML0162

void delay(void);

sbit mybit= P1^2;

void main(void){

mybit=0;

while(1)

{

mybit=1;

delay();

mybit=0;

delay();

}

}

void delay(void){

unsigned int z;

for(z=0;z<=10000;z++);

}
```



Min Time: 0 s
Max Time: 268.5441 s
Grid: 50 ms
☑ Signal Info
□ Amplitude
□ Timestamps Enable
□ Show Cycles
☑ Cursor

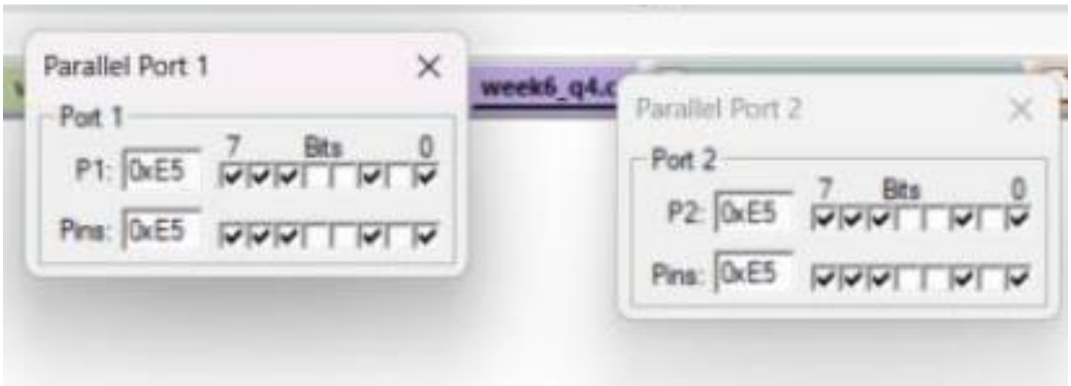
Register	Value
Regs	
r0	0x00
r1	0x00

```
    mybit=1;
    delay();
    mybit=0;
    delay();
```

P1: 0xFB ☑☑☑☑☑□☑☑
Pins: 0xFB ☑☑☑☑☑□☑☑

4. Transferring data from one port to another

```
#include<reg51.h>

//A G Srikanth

//21BML0162

void main(void)

{

unsigned char z;

P1=0xff;

P2=0x00;

while(1)

{

z=P1;

P2=z;

}

}
```



Parallel Port 1

Port 1

7 Bits 0

P1: 0xE5 ☑☑☑□□☑□☑

Pins: 0xE5 ☑☑☑□□☑□☑

Parallel Port 2

Port 2

7 Bits 0

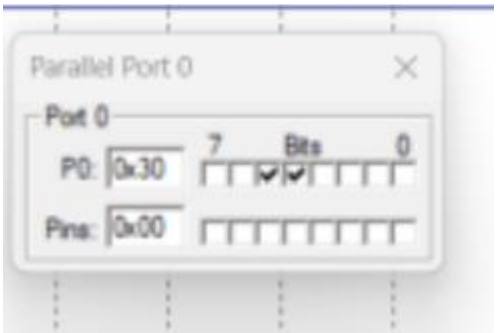
P2: 0xE5 ☑☑☑□□☑□☑

Pins: 0xE5 ☑☑☑□□☑□☑

5. Read pin P1.0 and P1.1 and send the respective ASCII data to different ports (P0, P2, P3)

```
#include<reg51.h>

//A G Srikanth

//21BML0162

void main(void)

{

unsigned char z;

P1=0xff;

P2=0x00;

P0=0x00;

P3=0x00;

while(1){

z= P1;

z&= 0x03;

switch(z){

case(0):P0='0';

break;

case(1):P2='1';

break;

case(2):P3='2';

break;

default:

P0='3';

P2='3';

P3='3';

}

}

}
```

Case 0

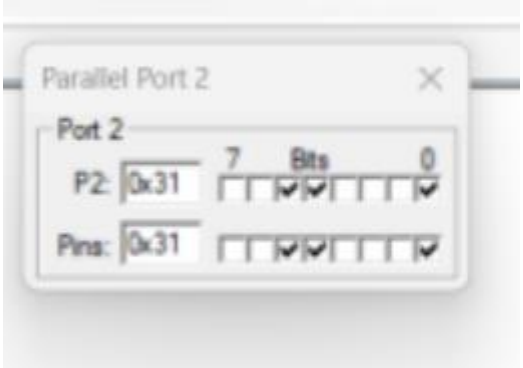

Parallel Port 0

Port 0

P0: 0x30 — Bits 7 ☐☐☑☑☐☐☐☐ 0

Pins: 0x00 — ☐☐☐☐☐☐☐☐

Case 1

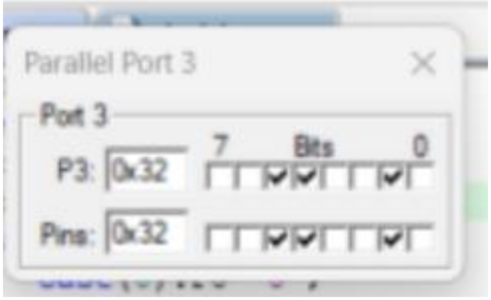

Parallel Port 2

Port 2

P2: 0x31 — Bits 7 ☐☐☑☑☐☐☐☑ 0

Pins: 0x31 — ☐☐☑☑☐☐☐☑

Case 2

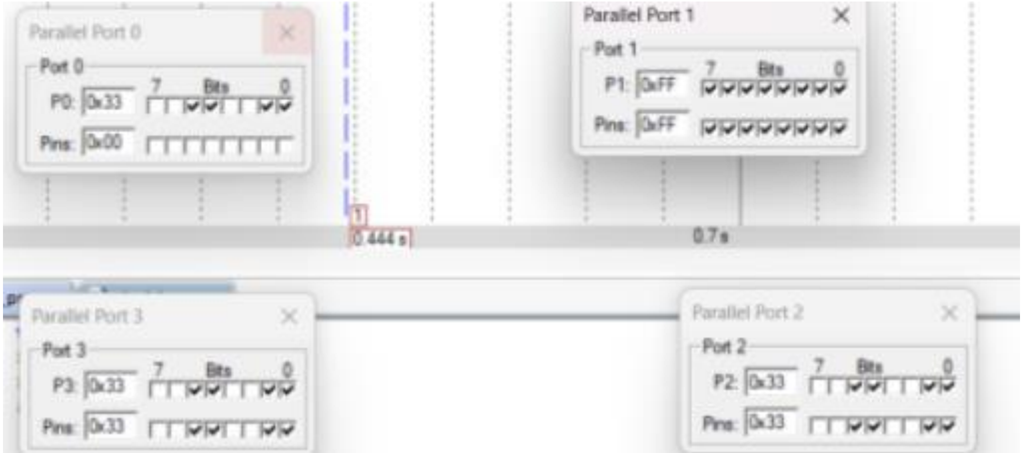

Parallel Port 3

Port 3

P3: 0x32 — Bits 7 ☐☐☑☑☐☐☑☐ 0

Pins: 0x32 — ☐☐☑☑☐☐☑☐

Default case



Parallel Port 0
Port 0
P0: 0x33
Pins: 0x00

Parallel Port 1
Port 1
P1: 0xFF
Pins: 0xFF

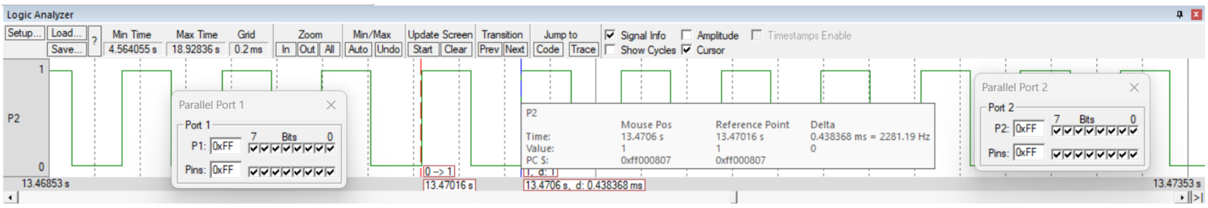
Parallel Port 3
Port 3
P3: 0x33
Pins: 0x33

Parallel Port 2
Port 2
P2: 0x33
Pins: 0x33

6. **Write an 8051 C program to monitor a switch connected at P1.2. If the switch is in 'HIGH' state generate a square waveform of 2.5 KHz with 50% duty cycle using Timer 0 in MODE 1 at P2.6, Else 1.5 KHz with 50 % Duty Cycle using Timer 1 in MODE 1 at P2.6. Assume XTAL=11.0592 MHz.**

```
#include <reg51.h>
//A G Srikanth
//21BML0162
sbit mybit= P1^2;
sbit mybit1= P2^6;
void delay_timer0(void);
void delay_timer1(void);
void main(void)
{
        mybit=1;
        mybit1=0;
while(1)
{
        if(mybit==1)
        {
                mybit1=1;
                delay_timer0();
                mybit1=0;
```

```
                    delay_timer0();
            }
            else
            {
                    mybit1=1;
                    delay_timer1();
                    mybit1=0;
                    delay_timer1();
             }
            }
    }

    void delay_timer0(void)
    {
            TMOD=0x01;
            TL0=0x48;
            TH0=0xFF;
            TR0=1;
            while(TF0==0);
            TF0=0;
            TR0=0;
    }
    void delay_timer1(void)
    {
            TMOD=0x10;
            TL1=0xD2;
            TH1=0XFF;
            TR1=1;

            while(TF1==0);

            TF1=0;

            TR1=1;

    }
```



Min Time 4.564055 s
Max Time 18.92836 s
Grid 0.2 ms

☑ Signal Info ☐ Amplitude ☐ Timestamps Enable
☐ Show Cycles ☑ Cursor

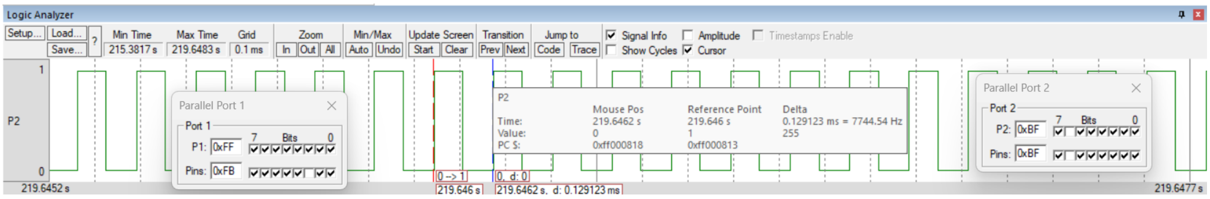
Parallel Port 1
Port 1
P1: 0xFF Bits 7–0: ☑☑☑☑☑☑☑☑
Pins: 0xFF Bits 7–0: ☑☑☑☑☑☑☑☑

P2

	Mouse Pos	Reference Point	Delta
Time:	13.4706 s	13.47016 s	0.438368 ms = 2281.19 Hz
Value:	1	1	0
PC $:	0xff000807	0xff000807	

13.47016 s
13.4706 s, d: 0.438368 ms

Parallel Port 2
Port 2
P2: 0xFF Bits 7–0: ☑☑☑☑☑☑☑☑
Pins: 0xFF Bits 7–0: ☑☑☑☑☑☑☑☑



Min Time 215.3817 s
Max Time 219.6483 s
Grid 0.1 ms

☑ Signal Info ☐ Amplitude ☐ Timestamps Enable
☐ Show Cycles ☑ Cursor

Parallel Port 1
Port 1
P1: 0xFF Bits 7–0: ☑☑☑☑☑☑☑☑
Pins: 0xFB Bits 7–0: ☑☑☑☑☑☐☑☑

P2

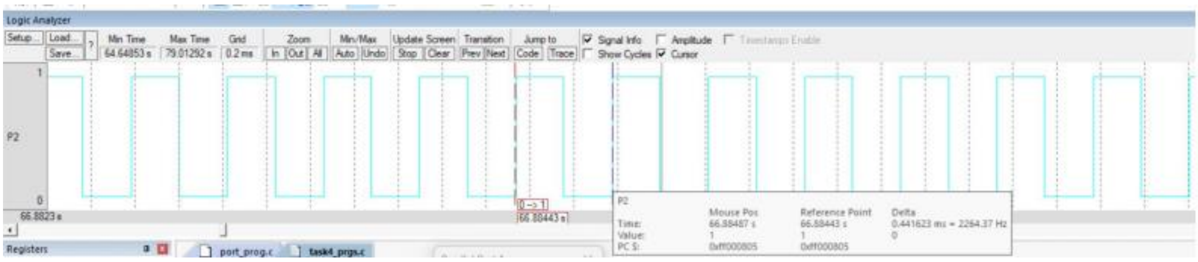
	Mouse Pos	Reference Point	Delta
Time:	219.6462 s	219.646 s	0.129123 ms = 7744.54 Hz
Value:	0	1	255
PC $:	0xff000818	0xff000813	

219.646 s
219.6462 s, d: 0.129123 ms

Parallel Port 2
Port 2
P2: 0xBF Bits 7–0: ☑☐☑☑☑☑☑☑
Pins: 0xBF Bits 7–0: ☑☐☑☑☑☑☑☑

7. **Write an 8051 C program to monitor a switch connected at P1.2. If the switch is in 'HIGH' state generate a square waveform of 2.5 KHz with 50% duty cycle using Timer 0 in MODE 1 at P2.6, Else 1.5 KHz with 50 % Duty Cycle using Timer 1 in MODE 1 at P2.6. Assume XTAL=11.0592 MHz.**

```
#include<reg51.h>

void delay_1(void);

void delay_2(void);

sbit sq_wave = P2^6;

sbit sw = P1^2;

void main(void){

sw = 1;

while(1){

if(sw==1){

sq_wave = 1;

delay_1();

sq_wave = 0;

delay_1();

}

else{

sq_wave = 1;

delay_2();

sq_wave = 0;

delay_2();

}

}

}

void delay_1(void){

TMOD = 0x01;

TL0 = 0x48;
TH0 = 0xFF;

TR0 = 1;

while(TF0==0);

TF0 = 0;

TR0 = 0;

}

void delay_2(void){

TMOD = 0x10;

TL1 = 0xCC;

TH1 = 0xFE;

TR1 = 1;

while(TF1==0);

TF1 = 0;

TR1 = 0;

}
```



Min Time 64.64853 s

Max Time 79.01292 s

Grid 0.2 ms

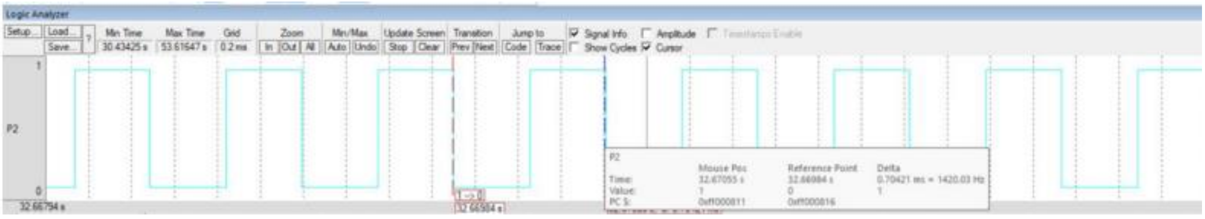
☑ Signal Info ☐ Amplitude

☐ Show Cycles ☑ Cursor

66.8823 s

66.88443 s

P2	Mouse Pos	Reference Point	Delta
Time:	66.88487 s	66.88443 s	0.441623 ms = 2264.37 Hz
Value:	1	1	0
PC $:	0xff000805	0xff000805	



Min Time 30.43425 s

Max Time 53.61647 s

Grid 0.2 ms

☑ Signal Info ☐ Amplitude

☐ Show Cycles ☑ Cursor

32.66794 s

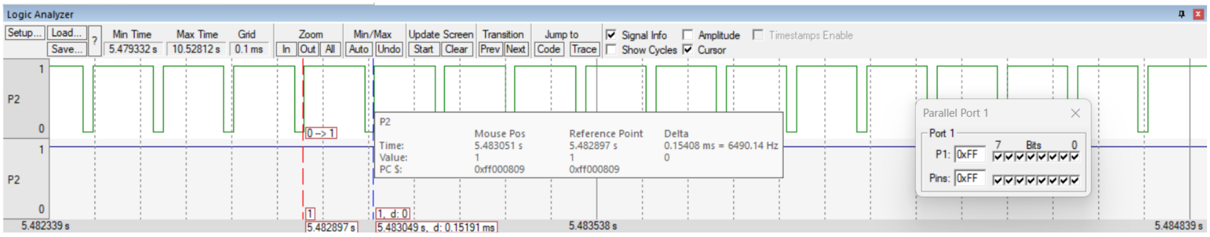
32.66984 s

P2	Mouse Pos	Reference Point	Delta
Time:	32.67055 s	32.66984 s	0.70421 ms = 1420.03 Hz
Value:	1	0	1
PC $:	0xff000811	0xff000816	

8. **Write an 8051 C program to monitor a switch connected at P1.5. If the switch is in ‘HIGH’ state generate a square waveform of 6 KHz with 75% duty cycle using Timer 0 in MODE 1 at P2.6, Else 3 KHz with 25 % Duty Cycle using Timer 1 in MODE 1 at P2.7. Assume XTAL=11.0592 MHz.**

```
#include <reg51.h>

//A G Srikanth

//21BML0162

void delay_timer0(void);

void delay_timer1(void);

sbit mybit=P1^5;

sbit mybit1=P2^6;

sbit mybit2=P2^7;

void main(void)

{

        mybit=1;

        mybit1=0;

        mybit1=0;

        while(1)

        {

        if(mybit==1)

        {

                mybit1=1;

                delay_timer0();

                delay_timer0();

                delay_timer0();

                mybit1=0;

                delay_timer0();

        }

        else{

                mybit2=1;

                delay_timer1();
```

```
                    mybit2=0;
                    delay_timer1();
                    delay_timer1();
                    delay_timer1();
          }
          }
}

void delay_timer0()
{
          TMOD=0x02;
          TL0=0xDA;
          TH0=0xFF;
          TR0=1;
          while(TF0==0);
          TF0=0;
          TR0=0;
}

void delay_timer1()
{
          TMOD=0x10;
          TL1=0xB4;
          TH1=0xFE;
          TR1=1;
          while(TF1==0);
          TF1=0;
          TR1=0;
}
```



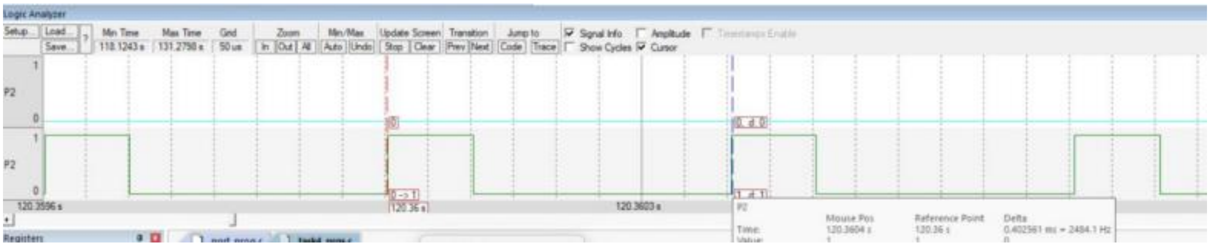
Min Time 5.479332 s
Max Time 10.52812 s
Grid 0.1 ms

☑ Signal Info
□ Amplitude
□ Timestamps Enable
□ Show Cycles
☑ Cursor

P2	Mouse Pos	Reference Point	Delta
Time:	5.483051 s	5.482897 s	0.15408 ms = 6490.14 Hz
Value:	1	1	0
PC $:	0xff000809	0xff000809	

5.482339 s
5.482897 s
5.483049 s, d: 0.15191 ms
5.483538 s
5.484839 s

Port 1
P1: 0xFF ☑☑☑☑☑☑☑☑
Pins: 0xFF ☑☑☑☑☑☑☑☑



Min Time 118.1243 s
Max Time 131.2758 s
Grid 50 us

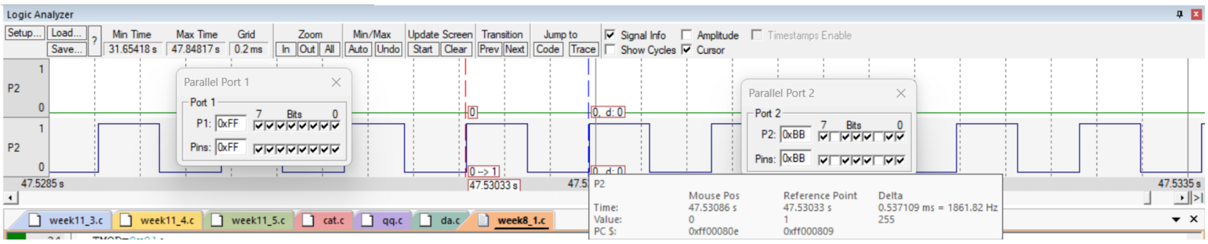
☑ Signal Info
□ Amplitude
□ Timestamps Enable
□ Show Cycles
☑ Cursor

120.3596 s
120.36 s
120.3603 s

P2	Mouse Pos	Reference Point	Delta
Time:	120.3604 s	120.36 s	0.402561 ms = 2484.1 Hz
Value:	1	1	0

9. **Write an 8051 C program to monitor a switch connected at P1.6. If the switch is in ‘HIGH’ state generate a square waveform of 2 KHz with 50% duty cycle using Timer 0 in MODE 1 at P2.6, Else 4 KHz with 50 % Duty Cycle using Timer 0 in MODE 2 at P2.2. Assume XTAL=11.0592 MHz.**

```
#include <reg51.h>

//A G Srikanth

//21BML0162

void delay_timer0(void);

void delay_timer1(void);

sbit mybit=P1^6;

sbit mybit1=P2^6;

sbit mybit2=P2^2;

void main(void)

{

        mybit=1;

        mybit1=0;

        mybit1=0;

        while(1)

        {

        if(mybit==1)

        {

                mybit1=1;

                delay_timer0();

                mybit1=0;

                delay_timer0();

        }

        else{

                mybit2=1;

                delay_timer1();

                mybit2=0;
                delay_timer1();
        }
        }
}

void delay_timer0()
{
        TMOD=0x01;
        TL0=0x1A;
        TH0=0xFF;
        TR0=1;
        while(TF0==0);
        TF0=0;
        TR0=0;
}

void delay_timer1()
{
        TMOD=0x20;
        TH1=0x8D;
        TR1=1;
        while(TF1==0);
        TF1=0;
        TR1=0;
}
```



Logic Analyzer

Min Time: 31.65418 s　Max Time: 47.84817 s　Grid: 0.2 ms

☑ Signal Info　☐ Amplitude　☐ Timestamps Enable

☐ Show Cycles　☑ Cursor

Parallel Port 1 — Port 1

P1: 0xFF　☑☑☑☑☑☑☑☑

Pins: 0xFF　☑☑☑☑☑☑☑☑

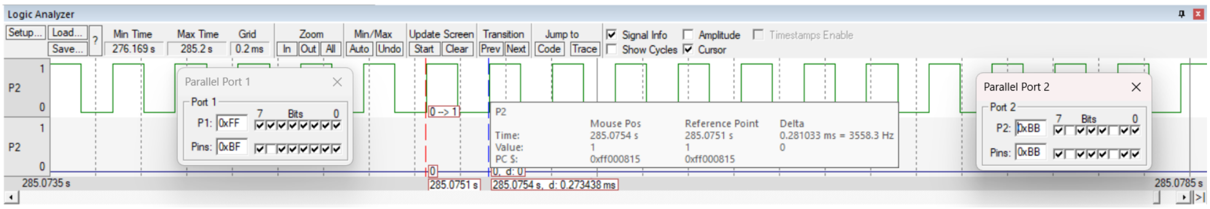
Parallel Port 2 — Port 2

P2: 0xBB　☑☐☑☑☑☐☑☑

Pins: 0xBB　☑☐☑☑☑☐☑☑

P2

	Mouse Pos	Reference Point	Delta
Time:	47.53086 s	47.53033 s	0.537109 ms = 1861.82 Hz
Value:	0	1	255
PC $:	0xff00080e	0xff000809	



Min Time: 276.169 s

Max Time: 285.2 s

Grid: 0.2 ms

☑ Signal Info ☐ Amplitude ☐ Timestamps Enable

☐ Show Cycles ☑ Cursor

Parallel Port 1

Port 1

		7			Bits				0
P1:	0xFF	☑	☑	☑	☑	☑	☑	☑	☑
Pins:	0xBF	☑	☐	☑	☑	☑	☑	☑	☑

P2

	Mouse Pos	Reference Point	Delta
Time:	285.0754 s	285.0751 s	0.281033 ms = 3558.3 Hz
Value:	1	1	0
PC $:	0xff000815	0xff000815	

Parallel Port 2

Port 2

		7			Bits				0
P2:	0xBB	☑	☐	☑	☑	☑	☐	☑	☑
Pins:	0xBB	☑	☐	☑	☑	☑	☐	☑	☑

285.0735 s

285.0751 s

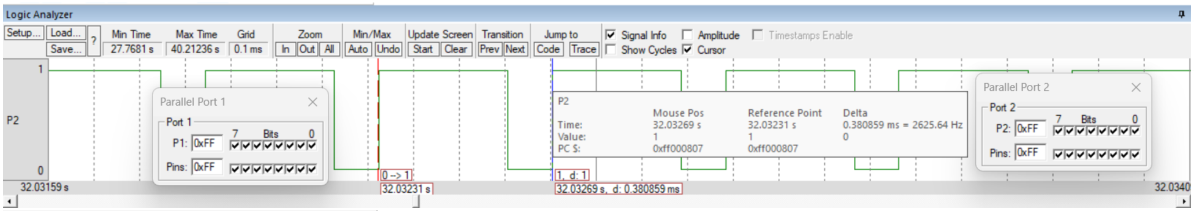
285.0754 s, d: 0.273438 ms

285.0785 s

10. **Write an 8051 C program to monitor a switch connected at P1.1. If the switch is in ‘HIGH’ state generate a square waveform of 3 KHz with 75% duty cycle using Timer 1 in MODE 2 at P2.6, Else 8 KHz with 25 % Duty Cycle using Timer 1 in MODE 1 at P2.6. Assume XTAL=11.0592 MHz.**

```
#include <reg51.h>
//A G Srikanth
//21BML0162
void delay_timer0(void);
void delay_timer1(void);
sbit mybit=P1^1;
sbit mybit1=P2^6;
void main(void)
{
        mybit=1;
        mybit1=0;
        while(1)
        {
        if(mybit==1)
        {
                mybit1=1;
                delay_timer0();
                delay_timer0();
                delay_timer0();
                mybit1=0;
                delay_timer0();
        }
        else{
                mybit1=1;
                delay_timer1();
```

```
                        mybit1=0;
                        delay_timer1();
                        delay_timer1();
                        delay_timer1();
            }
            }
}

void delay_timer0()
{
            TMOD=0x20;
            TH1=0xB3;
            TR1=1;
            while(TF1==0);
            TF1=0;
            TR1=0;
}

void delay_timer1()
{
            TMOD=0x01;
            TL0=0xE4;
            TH0=0xFF;
            TR0=1;
            while(TF0==0);
            TF0=0;
            TR0=0;
}
```



Min Time 27.7681 s

Max Time 40.21236 s

Grid 0.1 ms

☑ Signal Info ☐ Amplitude ☐ Timestamps Enable

☐ Show Cycles ☑ Cursor

Parallel Port 1

Port 1 — 7 Bits 0

P1: 0xFF ☑☑☑☑☑☑☑☑

Pins: 0xFF ☑☑☑☑☑☑☑☑

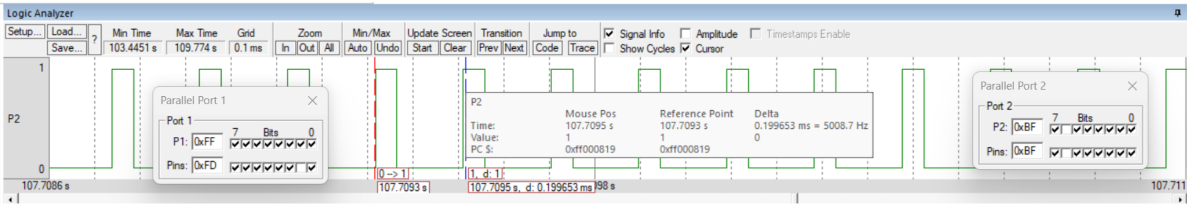
```
P2
          Mouse Pos     Reference Point   Delta
Time:     32.03269 s    32.03231 s        0.380859 ms = 2625.64 Hz
Value:    1             1                 0
PC $:     0xff000807    0xff000807
```

Parallel Port 2

Port 2 — 7 Bits 0

P2: 0xFF ☑☑☑☑☑☑☑☑

Pins: 0xFF ☑☑☑☑☑☑☑☑



Min Time 103.4451 s

Max Time 109.774 s

Grid 0.1 ms

☑ Signal Info ☐ Amplitude ☐ Timestamps Enable

☐ Show Cycles ☑ Cursor

Parallel Port 1

Port 1 — 7 Bits 0

P1: 0xFF ☑☑☑☑☑☑☑☑

Pins: 0xFD ☑☑☑☑☑☑☐☑

```
P2
          Mouse Pos     Reference Point   Delta
Time:     107.7095 s    107.7093 s        0.199653 ms = 5008.7 Hz
Value:    1             1                 0
PC $:     0xff000819    0xff000819
```

Parallel Port 2

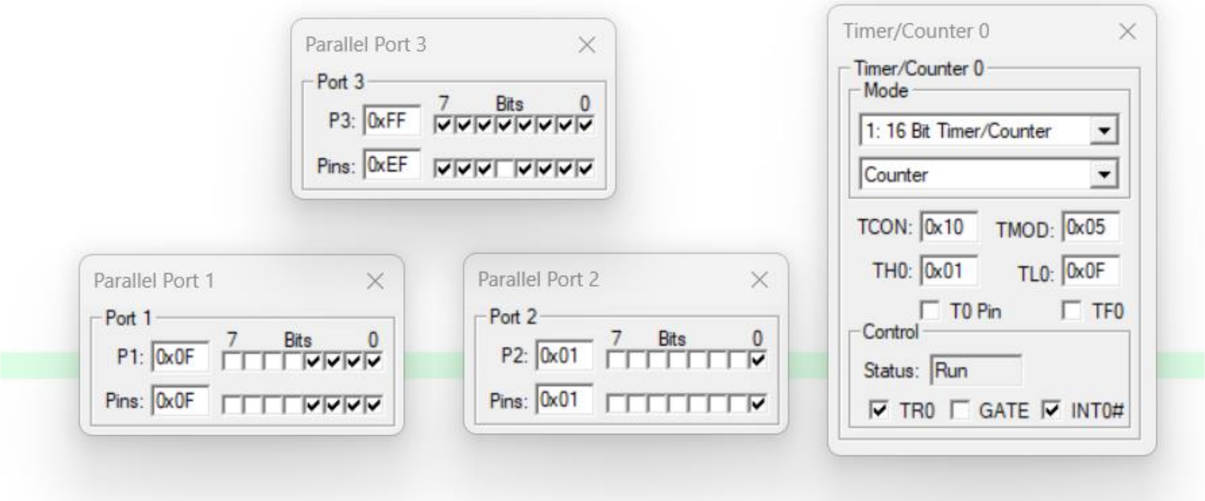
Port 2 — 7 Bits 0

P2: 0xBF ☑☐☑☑☑☑☑☑

Pins: 0xBF ☑☐☑☑☑☑☑☑

11. Assume that a 1-Hz external clock is being fed into pin T0. Write a C program for counter 0 in mode 1 (16-bit) to count the pulses and display the TH0 and TL0 registers on P2 and P1, respectively.

```
#include<reg51.h>

//A G Srikanth

//21BML0162

sbit cnt0=P3^4;

void main(void)

{

    P2=0x00;

    P1=0x00;

    cnt0=1;

    TMOD=0x05;

    while(1)

    {

        TR0=1;

        TH0=0;

        TL0=0;

        while(TF0==0)

        {

            P2=TH0;

            P1=TL0;

        }

        TF0=0;

    }

}
```



Parallel Port 3

Port 3

P3: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑

Pins: 0xEF ☑☑☑□☑☑☑☑

Parallel Port 1

Port 1

P1: 0x0F 7 Bits 0 □□□□☑☑☑☑

Pins: 0x0F □□□□☑☑☑☑

Parallel Port 2

Port 2

P2: 0x01 7 Bits 0 □□□□□□□☑

Pins: 0x01 □□□□□□□☑

Timer/Counter 0

Timer/Counter 0

Mode

1: 16 Bit Timer/Counter

Counter

TCON: 0x10 TMOD: 0x05

TH0: 0x01 TL0: 0x0F

□ T0 Pin □ TF0

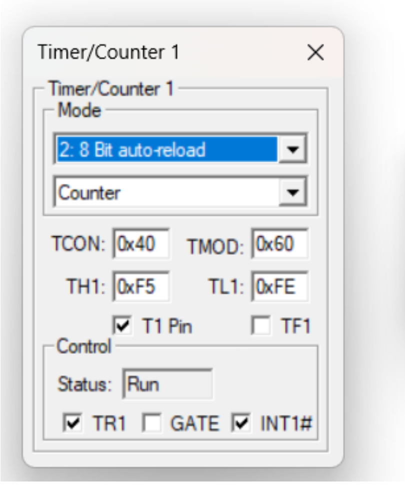
Control

Status: Run

☑ TR0 □ GATE ☑ INT0#

12. **Assume that a 1-Hz external clock is being fed into pin T1. Write a C program for counter 1 in mode 2 in such a way for every 10 count , the controller has to toggle(ON/OFF) an LED connected at P2.1.**

```
#include<reg51.h>

//A G Srikanth

//21BML0162

sbit cnt1=P3^5;

sbit pin=P2^1;

void main(void)

{

        pin=0;

        cnt1=1;

        TMOD=0x60;

        TL1=0xF5;

        TH1=0xF5;

        TR1=1;

        while(1)

        {

                while(TF1==0);

                pin=1;

                TF1=0;

                while(TF1==0);

                pin=0;

                TF1=0;

        }

}
```



Timer/Counter 1

Timer/Counter 1

Mode

2: 8 Bit auto-reload

Counter

TCON: 0x40 TMOD: 0x60

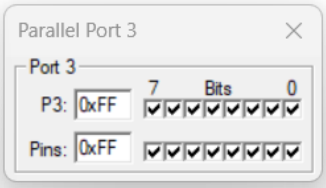
TH1: 0xF5 TL1: 0xFE

☑ T1 Pin ☐ TF1

Control

Status: Run

☑ TR1 ☐ GATE ☑ INT1#



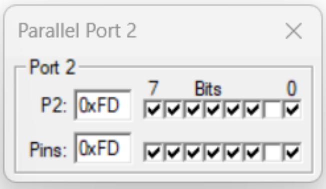
Parallel Port 3

Port 3

7 Bits 0

P3: 0xFF ☑☑☑☑☑☑☑☑

Pins: 0xFF ☑☑☑☑☑☑☑☑



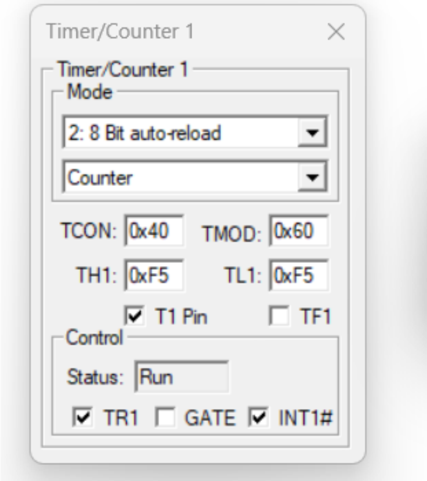
Parallel Port 2

Port 2

7 Bits 0

P2: 0xFD ☑☑☑☑☑☑☐☑

Pins: 0xFD ☑☑☑☑☑☑☐☑



Timer/Counter 1

Timer/Counter 1

Mode

2: 8 Bit auto-reload

Counter

TCON: 0x40 TMOD: 0x60

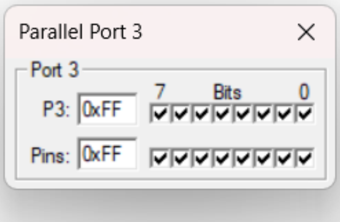
TH1: 0xF5 TL1: 0xF5

☑ T1 Pin ☐ TF1

Control

Status: Run

☑ TR1 ☐ GATE ☑ INT1#



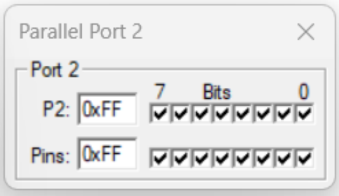
Parallel Port 3

Port 3

7 Bits 0

P3: 0xFF ☑☑☑☑☑☑☑☑

Pins: 0xFF ☑☑☑☑☑☑☑☑



Parallel Port 2

Port 2

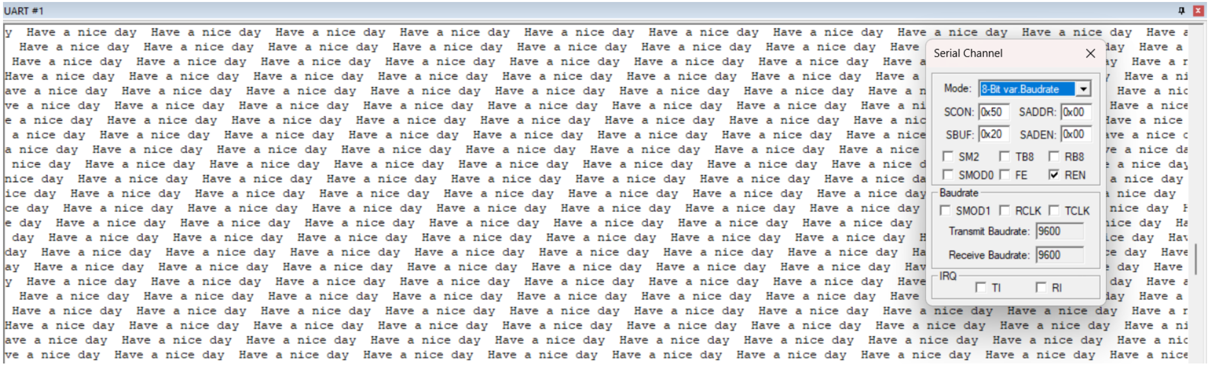
7 Bits 0

P2: 0xFF ☑☑☑☑☑☑☑☑

Pins: 0xFF ☑☑☑☑☑☑☑☑

13. **Write an 8051 C program to serially transmit the message “HAVE A NICE DAY!” with a baud rate of 9600. Assume XTAL=11.0592 MHz**

```
#include<reg51.h>

//A G Srikanth

//21BML0162

void main(void)

{

TMOD=0x20;

TH1=-3;

        SCON=0x50;

        TR1=1;

        while(1)

        {

                unsigned char i;

                unsigned char arr[]=" Have a nice day ";

                i=0;

        while(arr[i]!='\0')

        {

                SBUF=arr[i];

                while(TI==0);

                TI=0;

                i++;

        }

        }

}
```



UART #1

```
y  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a
  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a
 Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a n
Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a ni
ave a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nic
ve a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice
e a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice
 a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice d
a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice da
 nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day
nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day
ice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day
ce day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  H
e day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Ha
 day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Hav
day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have
ay  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have
y  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a
  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a
 Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a n
Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a ni
ave a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nic
ve a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice day  Have a nice
```

Serial Channel

Mode: 8-Bit var.Baudrate

SCON: 0x50 SADDR: 0x00

SBUF: 0x20 SADEN: 0x00

□ SM2 □ TB8 □ RB8

□ SMOD0 □ FE ☑ REN

Baudrate

□ SMOD1 □ RCLK □ TCLK

Transmit Baudrate: 9600

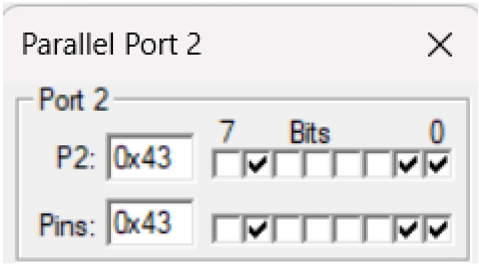
Receive Baudrate: 9600

IRQ

□ TI □ RI

14. Write an 8051 C program to serially receive the message and forward it to P2. Assume XTAL=11.0592 MHz.

```
#include<reg51.h>

//A G Srikanth

//21BML0162

void main(void)

{

        P2=0x00;

        TMOD=0x20;

        TH1=-3;

        SCON=0x50;

        TR1=1;

        while(1)

        {

                P2=SBUF;

                RI=0;

        }

}
```



Parallel Port 2

Port 2

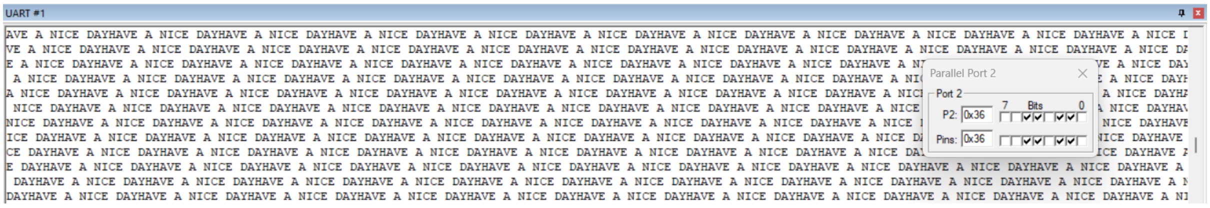
P2: 0x43 □☑□□□□☑☑

Pins: 0x43 □☑□□□□☑☑

15. **Write an 8051 C program to perform the following simultaneously.**
    a) **To serially transmit the message “HAVE A WONDERFUL DAY!” with a baud rate of 19200.**
    b) **To serially receive the message and forward it to P2.**

```
#include<reg51.h>

//A G Srikanth

//21BML0162

void main(void)

{

        unsigned char arr[]= "HAVE A NICE DAY";

        unsigned char i;

        P2=0x00;

        TMOD=0x20;

        PCON=0x80;

        TH1=-3;

        SCON=0x50;

        TR1=1;

        i=0;

        while(1)

        {

                while(arr[i]!='\0')

                {

                SBUF=arr[i];

                while(TI==0);

                TI=0;

                i++;

        }

        while(RI==1)

        {

                P2=SBUF;

                RI=0;

        }
```

```
            i=0;
    }
    }
```



UART #1

```
AVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE D
VE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DA
E A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NI                 VE A NICE DAY
 A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NIC                 E A NICE DAYH
A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE                  A NICE DAYHA
 NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE                  A NICE DAYHAV
NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE I                 NICE DAYHAVE
ICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DA                 ICE DAYHAVE
CE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DA                 ICE DAYHAVE A
E DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A
 DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A N
DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NICE DAYHAVE A NI
```

Parallel Port 2

Port 2

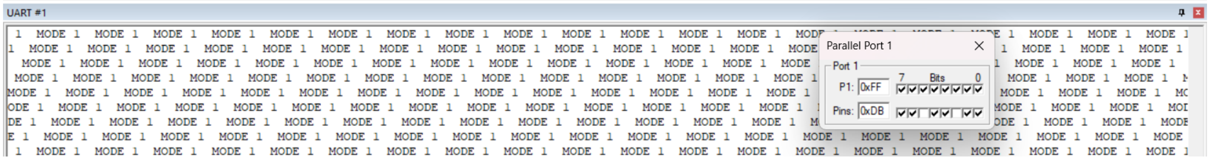
		7			Bits				0
P2:	0x36	□	□	☑	☑	□	☑	☑	□
Pins:	0x36	□	□	☑	☑	□	☑	☑	□

16. Assume a Switch0 (SW0) and Switch1 (SW1) are connected at P1.2 and P1.5. Read the status of the Switch and transmit the messages serially as given below. Assume XTAL=11.0592 MHz.

Switch0 (SW0)	Switch1 (SW1)	Message to be transmitted	Baud Rate
0	0	SERIAL MODE ONE	1200
0	1	SERIAL MODE TWO	4800
1	0	SERIAL MODE THREE	9600
1	1	SERIAL MODE FOUR	19200

```
#include<reg51.h>

//A G Srikanth

//21BML0162

sbit sw0=P1^2;

sbit sw1=P1^5;

void main(void)

{

unsigned char i;

        unsigned char arr[]=" MODE 1 ";

        unsigned char arr1[]=" MODE 2 ";

        unsigned char arr2[]=" MODE 3 ";

        unsigned char arr3[]=" MODE 4 ";

TMOD=0x20;

SCON=0x50;

        TR1=1;

        while(1)

        {

        if(sw0==0&&sw1==0)

        {

        TH1=-24;
        i=0;
        while(arr[i]!='\0')
{
        SBUF=arr[i];
        while(TI==0);
        TI=0;
        i++;
}
}
else if(sw0==0&&sw1==1)
{
TH1=-6;

        i=0;
        while(arr1[i]!='\0')
{
        SBUF=arr1[i];
        while(TI==0);
        TI=0;
        i++;
}
}
else if(sw0==1&&sw1==0)
{
TH1=-3;

        i=0;
```

```
                while(arr2[i]!='\0')
        {
                SBUF=arr2[i];
                while(TI==0);
                TI=0;
                i++;
        }
        }
        else if(sw0==1&&sw1==1)
        {
        TH1=-3;
                PCON=0x80;

                i=0;
                while(arr3[i]!='\0')
        {
                SBUF=arr3[i];
                while(TI==0);
                TI=0;
                i++;
        }
        }
}
        }
```



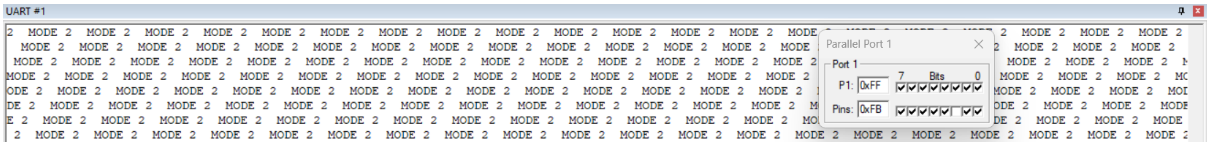
```
 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE                          E 1  MODE 1  MODE 1  MODE 1
1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE                          1  MODE 1  MODE 1  MODE 1
  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE                          1  MODE 1  MODE 1  MODE 1
 MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1                         MODE 1  MODE 1  MODE 1  M
MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1                         MODE 1  MODE 1  MODE 1  MC
ODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  1                      MODE 1  MODE 1  MODE 1  MOI
DE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  M                       ODE 1  MODE 1  MODE 1  MODE
E 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE
 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1  MODE 1
```

Parallel Port 1 ✕

Port 1

P1: 0xFF   7 Bits 0   ☑☑☑☑☑☑☑☑

Pins: 0xDB   ☑☑☐☑☑☐☑☑



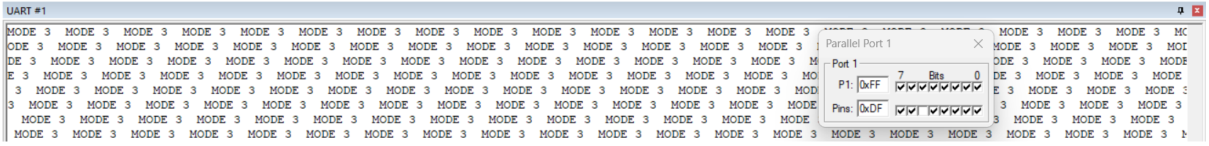
UART #1

```
2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE                      2  MODE 2  MODE 2  MODE 2
  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE                     2  MODE 2  MODE 2  MODE 2
 MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2                    MODE 2  MODE 2  MODE 2  M
MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2                    MODE 2  MODE 2  MODE 2  MC
ODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  I                  MODE 2  MODE 2  MODE 2  MOI
DE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  M                   ODE 2  MODE 2  MODE 2  MODE
E 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MO                    DE 2  MODE 2  MODE 2  MODE
 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2  MODE 2
```

Parallel Port 1 ×

Port 1

		7		Bits					0
P1:	0xFF	☑	☑	☑	☑	☑	☑	☑	☑
Pins:	0xFB	☑	☑	☑	☑	☑	□	☑	☑



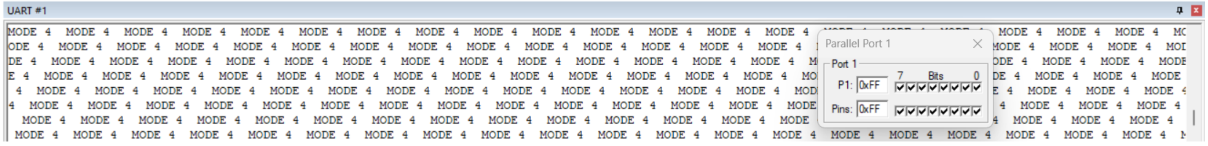
UART #1

```
MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3                    MODE 3  MODE 3  MODE 3  MC
ODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  I                  MODE 3  MODE 3  MODE 3  MOI
DE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  M                   ODE 3  MODE 3  MODE 3  MODE
E 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MO                    DE 3  MODE 3  MODE 3  MODE
 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MOD                    E 3  MODE 3  MODE 3  MODE 3
3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE                     3  MODE 3  MODE 3  MODE 3
  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE                     3  MODE 3  MODE 3  MODE 3
 MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  MODE 3  M
```

Parallel Port 1 ×

Port 1

		7		Bits					0
P1:	0xFF	☑	☑	☑	☑	☑	☑	☑	☑
Pins:	0xDF	☑	☑	□	☑	☑	☑	☑	☑



UART #1

```
MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4                    MODE 4  MODE 4  MODE 4  MC
ODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  I                  MODE 4  MODE 4  MODE 4  MOI
DE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  M                   ODE 4  MODE 4  MODE 4  MODE
E 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MO                    DE 4  MODE 4  MODE 4  MODE
 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MOD                    E 4  MODE 4  MODE 4  MODE 4
4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE                     4  MODE 4  MODE 4  MODE 4
  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE                     4  MODE 4  MODE 4  MODE 4
 MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  MODE 4  M
```

Parallel Port 1 ×

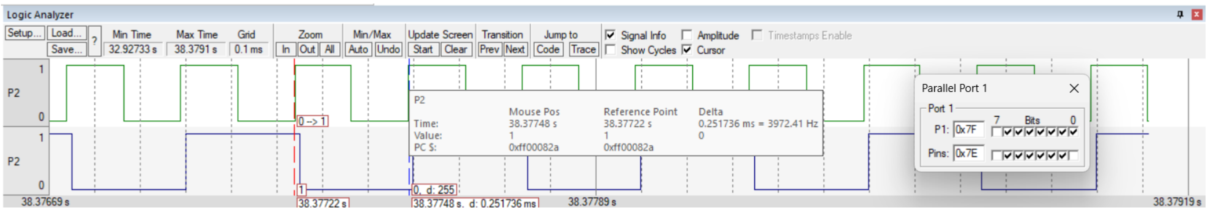
Port 1

		7		Bits					0
P1:	0xFF	☑	☑	☑	☑	☑	☑	☑	☑
Pins:	0xFF	☑	☑	☑	☑	☑	☑	☑	☑

17. **Develop an 8051 embedded C program to perform the following tasks.**
    a) **Transfer data from P1.0 to P1.7**
    b) **Generate a square wave of 4KHz at P2.2 using timer 0.**
    c) **Generate a square wave of 2KHz at P2.7 using timer 1.**
    Assume XTAL =11.0592 MHz

```
#include<reg51.h>

//A G Srikanth

//21BML0162

sbit pin1=P1^0;

sbit pin2=P1^7;

sbit s1=P2^2;

sbit s2=P2^7;

void main(void)

{

	pin2=0;

	s1=0;

	s2=0;

	pin1=1;

	TMOD=0x22;

	TH1=0x1A;

	TH0=0x8D;

	TR0=1;

	TR1=1;

	IE=0x8A;

	while(1)

	{

		pin2=pin1;

	}

}
```

```
void timer0() interrupt 1

{

        s1=~s1;

}


void timer1() interrupt 3

{

        s2=~s2;

}
```



Min Time 32.92733 s
Max Time 38.3791 s
Grid 0.1 ms

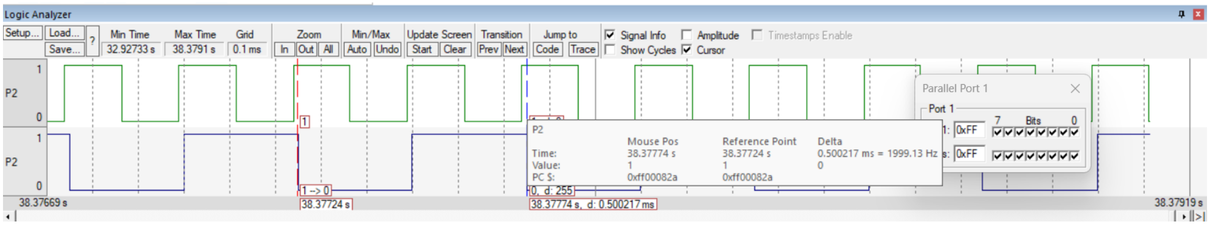
☑ Signal Info ☐ Amplitude ☐ Timestamps Enable
☐ Show Cycles ☑ Cursor

P2

	Mouse Pos	Reference Point	Delta
Time:	38.37748 s	38.37722 s	0.251736 ms = 3972.41 Hz
Value:	1	1	0
PC $:	0xff00082a	0xff00082a	

P1: 0x7F ☐☑☑☑☑☑☑☑
Pins: 0x7E ☐☑☑☑☑☑☑☐

38.37669 s | 38.37722 s | 38.37748 s, d: 0.251736 ms | 38.37789 s | 38.37919 s



Min Time 32.92733 s
Max Time 38.3791 s
Grid 0.1 ms

☑ Signal Info ☐ Amplitude ☐ Timestamps Enable
☐ Show Cycles ☑ Cursor

P2

	Mouse Pos	Reference Point	Delta
Time:	38.37774 s	38.37724 s	0.500217 ms = 1999.13 Hz
Value:	1	1	0
PC $:	0xff00082a	0xff00082a	

1: 0xFF ☑☑☑☑☑☑☑☑
s: 0xFF ☑☑☑☑☑☑☑☑

38.37669 s | 38.37724 s | 38.37774 s, d: 0.500217 ms | 38.37919 s

18. Develop an 8051 embedded C program to perform the following tasks.

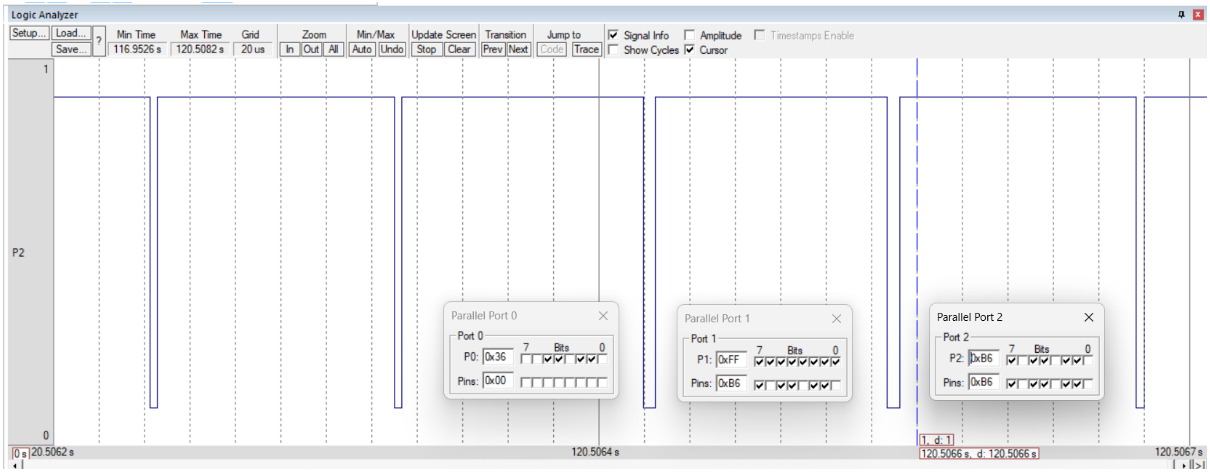
d) Transfer data from P1 to P2

e) Generate a square wave of 4.6 KHz at P2.2 using timer 0.

f) Forward the serially received data to P0.
Assume XTAL =11.0592 MHz

```
#include<reg51.h>

//A G Srikanth

//21BML0162

sbit s1=P2^2;

void main(void)

{

        s1=0;

        P1=0xFF;

        P2=0x00;

        P0=0x00;

        TMOD=0x22;

        TH1=-3;

        TH0=0x9C;

        TR0=1;

        TR1=1;

        SCON=0x50;

        IE=0x92;

        while(1)

        {

                P2=P1;

        }

}


void timer0() interrupt 1
```

```
{
            s1=~s1;
}


void timer1() interrupt 4
{
            P0=SBUF;
            RI=0;
}
```



Logic Analyzer

Min Time 116.9526 s

Max Time 120.5082 s

Grid 20 us

☑ Signal Info
□ Amplitude
□ Timestamps Enable
□ Show Cycles
☑ Cursor

Parallel Port 0

Port 0

P0: 0x36 □ □ ☑ ☑ □ ☑ ☑ □

Pins: 0x00 □ □ □ □ □ □ □ □

Parallel Port 1

Port 1

P1: 0xFF ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑

Pins: 0xB6 ☑ □ ☑ ☑ □ ☑ ☑ □

Parallel Port 2

Port 2

P2: 0xB6 ☑ □ ☑ ☑ □ ☑ ☑ □

Pins: 0xB6 ☑ □ ☑ ☑ □ ☑ ☑ □

1, d: 1

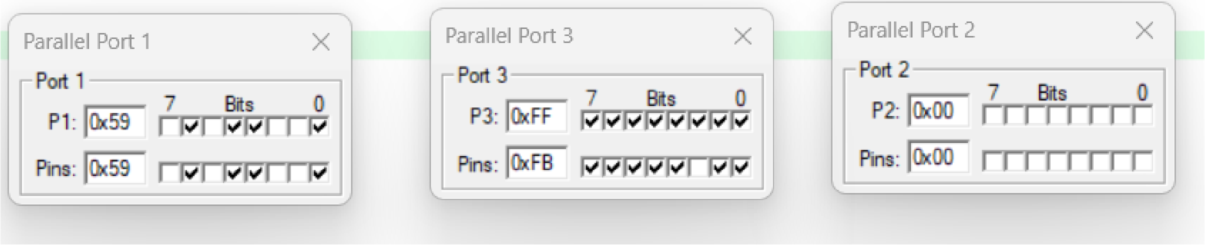
120.5066 s, d: 120.5066 s

19. Develop an 8051 embedded C program to perform the following tasks.

a) Increment the value of P1 if there is an active low signal on INT0.

b) Increment the value of P2 if there is an active low signal on INT1.

```
#include<reg51.h>
//A G Srikanth
//21BML0162
void main()
{
        P1=0x00;
        P2=0x00;
        IE=0x85;
        while(1)
        {
        }
}
void int0() interrupt 0
{
        P1++;
}


void int1() interrupt 2
{
        P2++;
}
```



Parallel Port 1

Port 1

P1: 0x59 (bits 7–0: □ ☑ □ ☑ ☑ □ □ ☑)

Pins: 0x59 (bits 7–0: □ ☑ □ ☑ ☑ □ □ ☑)

Parallel Port 3

Port 3

P3: 0xFF (bits 7–0: ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑)

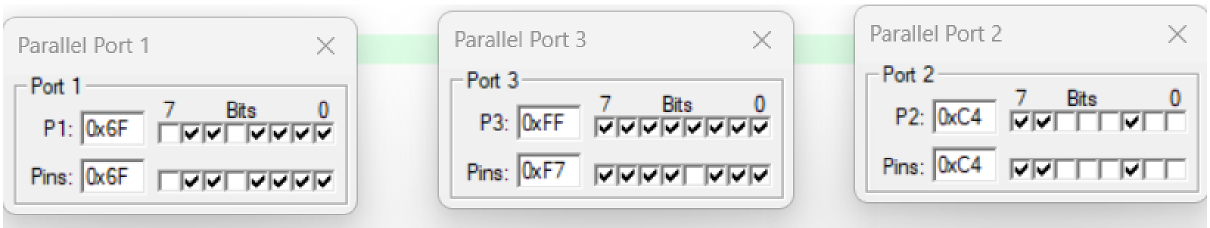
Pins: 0xFB (bits 7–0: ☑ ☑ ☑ ☑ ☑ □ ☑ ☑)

Parallel Port 2

Port 2

P2: 0x00 (bits 7–0: □ □ □ □ □ □ □ □)

Pins: 0x00 (bits 7–0: □ □ □ □ □ □ □ □)



Parallel Port 1

Port 1

7 Bits 0

P1: 0x6F □☑☑□☑☑☑☑

Pins: 0x6F □☑☑□☑☑☑☑

Parallel Port 3

Port 3

7 Bits 0

P3: 0xFF ☑☑☑☑☑☑☑☑

Pins: 0xF7 ☑☑☑☑□☑☑☑

Parallel Port 2

Port 2

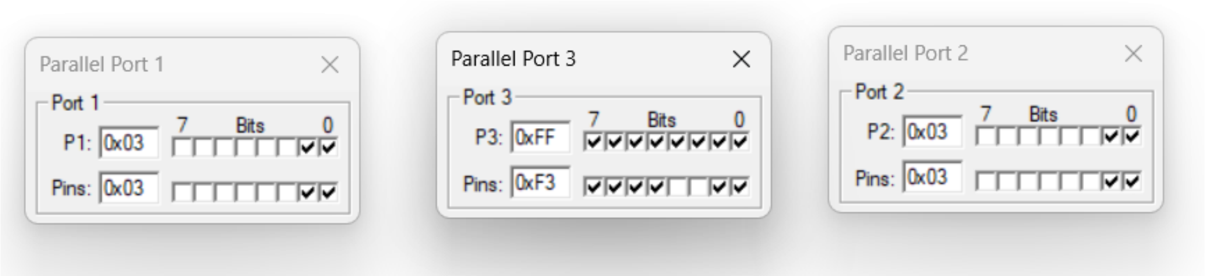
7 Bits 0

P2: 0xC4 ☑☑□□□☑□□

Pins: 0xC4 ☑☑□□□☑□□

20. **Develop an 8051 embedded C program to perform the following tasks.**
    a) **Increment the value of P1 if there is a negative edge on INT0.**
    b) **Increment the value of P2 if there is a negative edge on INT1.**

```
#include<reg51.h>

//A G Srikanth

//21BML0162

void main()

{

        P1=0x00;

        P2=0x00;

        TCON=0x05;

        IE=0x85;

        while(1)

        {

        }

}

void int0() interrupt 0

{

        P1++;

}


void int1() interrupt 2

{

        P2++;

}
```



Parallel Port 1

Port 1

P1: 0x03 — 7 Bits 0: □□□□□□☑☑

Pins: 0x03 — □□□□□□☑☑

Parallel Port 3

Port 3

P3: 0xFF — 7 Bits 0: ☑☑☑☑☑☑☑☑

Pins: 0xF3 — ☑☑☑☑□□☑☑

Parallel Port 2

Port 2

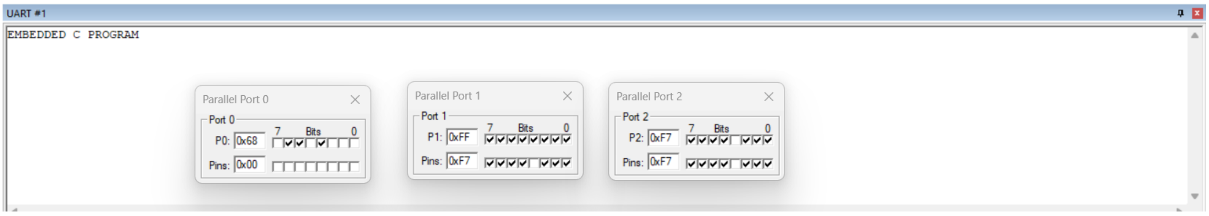
P2: 0x03 — 7 Bits 0: □□□□□□☑☑

Pins: 0x03 — □□□□□□☑☑

21. **Develop an 8051 embedded C program to perform the following tasks.**
    a. **Transfer the data from P1 to P2.**
    b. **Serially if any data received it must be sent to P0.**
    c. **Transmit the message "EMBEDDED C PROGRAM" at the baud rate of 19200.**

    Task c should be done only once. Task a and b should be done continuously.

```
#include<reg51.h>

//A G Srikanth

//21BML0162

void serial_int();

void serial_tx();

void main()

{

unsigned char i=0;

unsigned char arr[] = "EMBEDDED C PROGRAM";

P1=0xff;

P2=0x00;

P0=0x00;

TMOD=0x20;

SCON=0x50;

PCON|=0x80;

TH1=-3;

TR1=1;

EA=1;

ES=1;

while(arr[i]!='\0'){

SBUF=arr[i];

serial_tx();

i++;

}
while(1)

{

P2=P1;

}

}

void serial_int() interrupt 4

{

P0=SBUF;

 RI=0;

}

void serial_tx(){

while(TI==0);

TI=0;

}
```



```
EMBEDDED C PROGRAM
```

P0: 0x68 □☑☑□☑□□□

Pins: 0x00 □□□□□□□□

P1: 0xFF ☑☑☑☑☑☑☑☑

Pins: 0xF7 ☑☑☑☑□☑☑☑

P2: 0xF7 ☑☑☑☑□☑☑☑

Pins: 0xF7 ☑☑☑☑□☑☑☑