Indian Institute of Information Technology
Bhagalpur - 813210

Microprocessor & Interfacings Lab Report

Submitted by:

SNEH RANJAN	2001047	ECE
DHRUV SINGH RATHORE	2001003	CSE
SANI KUMAR	2001125	ECE

Department of Electronics & Communication Engineering

IIIT, BHAGALPUR, BIHAR 813210, INDIA

Jul-Dec 2022

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Bhagalpur-83210, Bihar, INDIA

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that ***Mr. SNEH RANJAN***(***2001047***), ***Mr. DHRUV SINGH RATHORE*** (***2001003***), ***Mr. SANI KUMAR***(***2001125***) has satisfactorily completed the course in ***Microprocessor & interfacings (EC304)*** during the academic year 2020-2024.

Date: 22/11/2022

Place: IIIT, Bhagalpur

Dr. Suraj
Assistant Professor, ECE
IIIT Bhagalpur 813210

Contents

1. Toggling all LEDs of Port 1

1.1 Aim

Toggle all LEDs connected to port P1 with some delay.

1.2 Code

//Toggle all LEDs connected to port p1 with some delay

```
#include<reg51.h>
void delay (unsigned int i);
void main()
{
    while(1)
    {
        P1=0xFF;
        delay(10000);
        P1=0x00;
        delay(10000);
    }
}
void delay (unsigned int i)
{
    unsigned k=0;
    for(k=0;k<i;k++);
}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to toggle all LEDs connected to port 1           %
%                                                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

1.3 Hex Code

:03000000020835BE

:0C083500787FE4F6D8FD75810902080008

:10080000759055E4F508F5090509E5097002050834

:10081000B410F5E508B427F06390FFE4F508F50996

:10082000C3E5099410E508942750D50509E509703A

:05083000EF050880EB5C

:00000001FF

1.4 Simulated Output

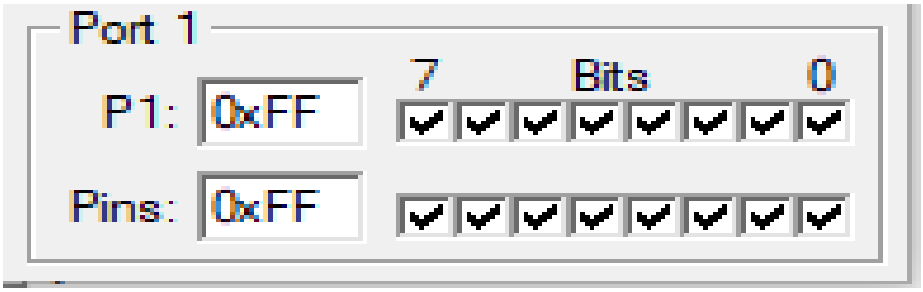


Figure 1.a: Simulated output at Port 1 with odd pin being high

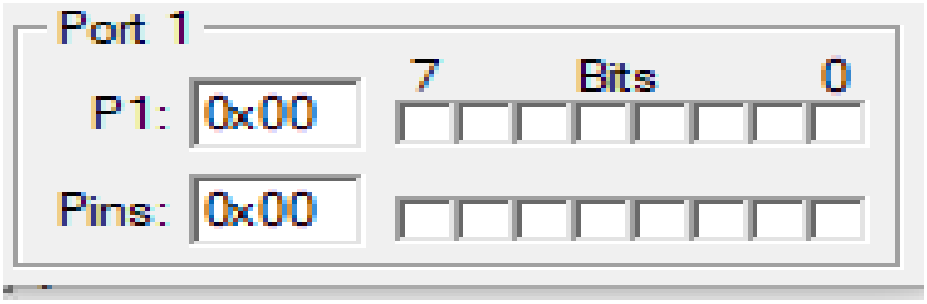


Figure 1.b: Simulated output at Port 1 with even pins being high

1.5 Conclusion

Hence, All the LEDs connected to port 1 has been blinked using a delay function which provides a delay.

2. Toggling LEDs Alternatively

2.1 Aim

Toggle all LEDs alternatively on port P1 with some delay.

2.2 Code

//Toggle all LEDs alternatively on port p1 with some delay

```
#include <reg51.h>
void delay (unsigned int i);
void main()
{
	while(1)
	{
		P1=0x55;
		for(i=0;i<10000;i++);
		P1=~P1;
		for(i=0;i<10000;i++);
	}
}
void delay (unsigned int i)
{
unsigned int k;
for(k=0;k<I;k++);
}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to toggle alternate LEDs connected to port 1
%
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

2.3 Hex Code

:03000000020835BE

:0C083500787FE4F6D8FD75810902080008

:10080000759055E4F508F5090509E5097002050834

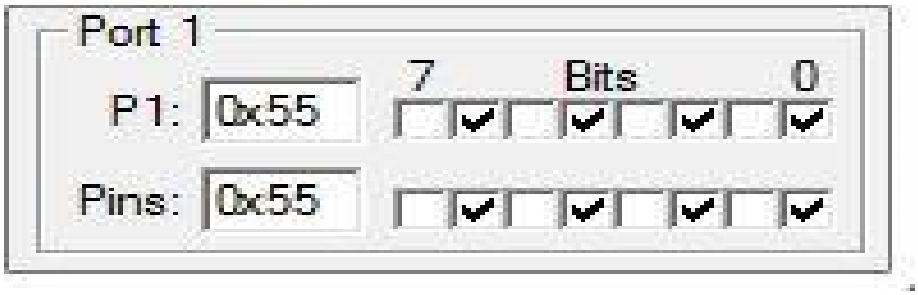
:10081000B410F5E508B427F06390FFE4F508F50996

:10082000C3E5099410E508942750D50509E509703A

:05083000EF050880EB5C

:00000001FF

2.4 Simulated Output



2.5 Conclusion

Hence, alternate LEDs connected to Port 1 has been blinked using delay function which produces a certain delay.

3. Left Shifting of Bits

3.1 Aim

Shift the blinking of LED that connected to port P1 from Right to left Keeping previous LED off.

3.2 Code

// Shift the blinking of LED that connected to port P1 from Right to left Keeping previous LED off.

```
#include <reg51.h>
unsigned int i;
unsigned int j;
void main()
{
	while(1)

	{
		P1=0x01;
		for(j=0;j<8;j++)
		{
		for(i=0;i<10000;i++);
		P1=P1<<1;
		for(i=0;i<10000;i++);
	}
	}
}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to shift blinking of LED connected to port 1
%
%                                                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

3.3 Hex Code

:03000000020848AB

:0C084800787FE4F6D8FD75810B020800F3

:10080000759001E4F50AF50BE4F508F5090509E52D

:1008100009700205 08B410F5E508B427F0E5902545

:10082000E0F590E4F508F5090509E5097002050809

:10083000B410F5E508B427F0050BE50B7002050AC6

:080840006408450A70C280B88B

:00000001FF

3.4 Simulated Output

3.5 Conclusion

Hence, LEDs connected to Port 1 has been blinked from Right to left keeping Previous LED off, with some delay so that we can observe shifting properly.

4. Shift Bits to Right

4.1 Aim

Shift the blinking of LED that connected to port P1 from Right to left Keeping previous LED off.

4.2 Code

// Shift the blinking of LED that connected to port P1 from Right to left Keeping previous LED off.

```
#include <reg51.h>
unsigned int i;
unsigned int j;
void main()
{
	while(1)

	{
		P1=0x80;
		for(j=0;j<8;j++)
		{
		for(i=0;i<10000;i++);
		P1=P1>>1;
		for(i=0;i<10000;i++);
	}
	}
}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to shift blinking of LEDs connected to port 1
%
%                                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

4.3 Hex Code

:03000000020848AB

:0C084800787FE4F6D8FD75810B020800F3

:10080000759080E4F50AF50BE4F508F5090509E5AE

:100810000970020508B410F5E508B427F0E590C3A7

:1008200013F590E4F508F5090509E50970020508D6

:10083000B410F5E508B427F0050BE50B7002050AC6

:080840006408450A70C280B88B

:00000001FF

4.4 Simulated Output

4.5 Conclusion

Hence, LEDs connected to Port 1 has been blinked from Left to R keeping Previous LED off, with some delay so that we can observe shifting properly.

5.Shifting of LEDs Left Keeping Previous 1

5.1 Aim

Shift the blinking of LED that connected to port P1 from Right to left Keeping previous LED on.

5.2 Code

// Shift the blinking of LED that connected to port P1 from Right to left Keeping previous LED on.

```
#include <reg51.h>
unsigned int i;
unsigned int j;
void main()
{
	while(1)

	{
		P1=0x01;
		for(j=0;j<8;j++)
		{
		for(i=0;i<60000;i++);
		P1=P1<<1;
		for(i=0;i<60000;i++);
			P1=P1+0x01;
	}
	}

}

/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to toggle all LEDs connected to port 1          %
%                                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

5.3 Hex Code

:0300000002087A79

:0C087A00787FE4F6D8FD75810B020800C1

:10080000759001E4F50AF50BE4F508F509AE08AFBB

:1008100009E4FCFD7B607AEAF9F8D3120864400A27

:100820000509E50970E7050880E3E59025E0F59006

:10083000E4F508F509AE08AF09E4FCFD7B607AEA4F

:10084000F9F8D3120864400A0509E50970E70508BC

:1008500080E30590050BE50B7002050A6408450A64

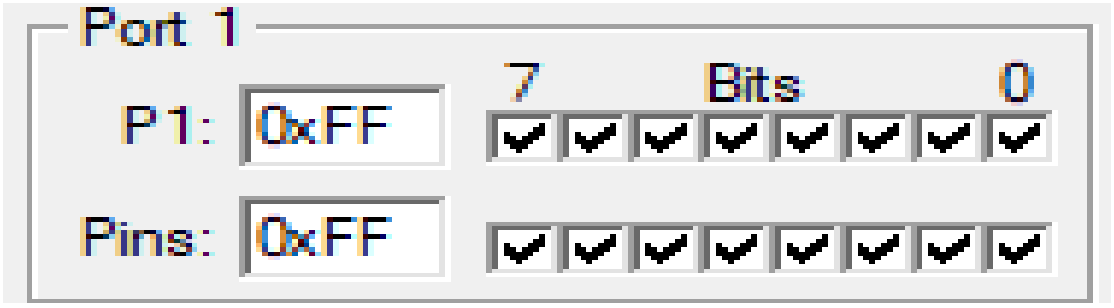
:0408600070A6809C62

:10086400EB9FF5F0EA9E42F0E99D42F0EC6480C80B

:0608740064809845F022AB

:00000001FF

5.4 Simulated Output



5.5 Conclusion

Hence, LEDs connected to Port 1 has been blinked from Right to Left keeping Previous LED on, with some delay so that we can observe shifting properly.

6.Shifting of LEDs Left Keeping Previous 1

6.1 Aim

Shift the blinking of LED that connected to port P1 from Left to Right Keeping previous LED on.

6.2 Code

// Shift the blinking of LED that connected to port P1 from Right to left Keeping previous LED on.

```
#include <reg51.h>
unsigned int i;
unsigned int j;
void main()
{
    while(1)

    {
        P1=0x80;
        for(j=0;j<8;j++)
        {

        for(i=0;i<60000;i++);
        P1=P1>>1;
        for(i=0;i<60000;i++);
            P1=P1+0x80;
    }
    }

}


/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to toggle all LEDs connected to port 1          %
%                                                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

6.3 Hex Code

:0300000002087E75

:0C087E00787FE4F6D8FD75810B020800BD

:10080000759080E4F50AF50BE4F508F509AE08AF3C

:1008100009E4FCFD7B607AEAF9F8D3120868400A23

:100820000509E50970E7050880E3E590C313F59035

:10083000E4F508F509AE08AF09E4FCFD7B607AEA4F

:10084000F9F8D3120868400A0509E50970E70508B8

:1008500080E374802590F590050BE50B7002050A86

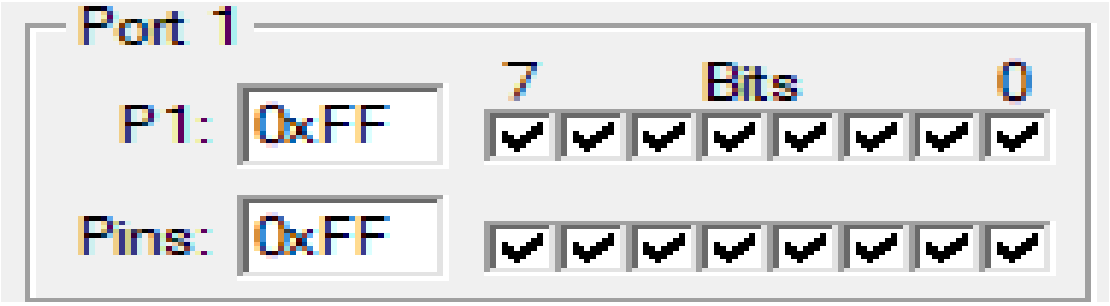
:080860006408450A70A28098AB

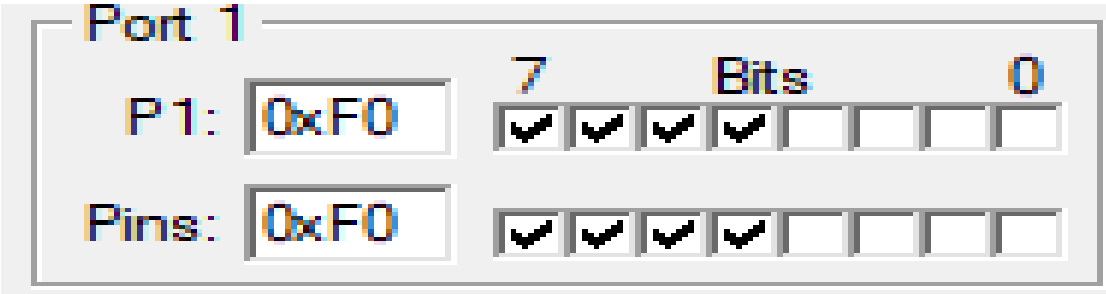
:10086800EB9FF5F0EA9E42F0E99D42F0EC6480C807

:0608780064809845F022A7

:00000001FF

6.4 Simulated Output





6.5 Conclusion

Hence, LEDs connected to Port 1 has been blinked from Left to Right keeping Previous LED on, with some delay so that we can observe shifting properly.

7.Implementing Various Patterns

7.1 Aim

Implement the pattern from 1 to 5, using switch keys connected to port 2.

7.2 Code

// Implement the pattern from 1 to 5, using switch keys connected to port 2.

```
#include <reg51.h>
unsigned int i;
unsigned int j;
void delay(unsigned int t);
sbit d=P2^0;  //proj1
sbit r=P2^1;  //proj2
sbit c=P2^2;  //proj3
sbit l=P2^3;  //proj4
sbit u=P2^4;  //proj5
void main()
{
      while(1)
      {
            if(d==1)
            {
            P1=0xFF;
            delay(10000);
            P1=0x00;
            delay(10000);
            }
      if(r==1)
      {
            P1=0x55;
            delay(10000);
            P1=~P1;
            delay(10000);
      }
  if(c==1)
      {
            P1=0x01;
            for(j=0;j<8;j++)
            {
             delay(10000);
             P1=P1<<1;
             delay(10000);
            }
      }
      if(l==1)
      {
            P1=0x80;
```

```
            for(j=0;j<8;j++)
            {
            for(i=0;i<10000;i++);
            P1=P1>>1;
            for(i=0;i<10000;i++);
        }


         }

        if(u==1)
            {
            P1=0x80;
            for(j=0;j<8;j++)
            {
             delay(10000);
             P1=P1>>1;
             delay(10000);
             P1=P1+0x80;
          }
         }

}
}
void delay(unsigned int t)
{
        for(i=0;i<t;i++);
}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to toggle all LEDs connected to port 1           %
%                                                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

7.3 Hex Code

:030000000208E013

:0C08E000787FE4F6D8FD75810B0208005B

:1008000030A00A7590FF1208C3E41208C130A10C91

:1008100075905512 08C36390FF1208C330A2237568

:100820009001E4F50AF50B1208C3E59025E01208E3

:10083000C1050BE50B7002050AC39408E50A940094

:1008400040E530A346759080E4F50AF50BE4F50821

:10085000F5090509E50970020508B410F5E508B4C5

:1008600027F0E590C313F590E4F508F5090509E5CF

:10087000970020508B410F5E508B427F0050BE58A

:100880000B7002050A6408450A70C220A40302081E

:1008900000759080E4F50AF50BC3E50B9408E50AB2

:1008A000940040030208001208C3E590C313120825

:1008B000C174802590F590050BE50B70DC050A806E

:0108C000D85F

:0608C100F5907F107E2778

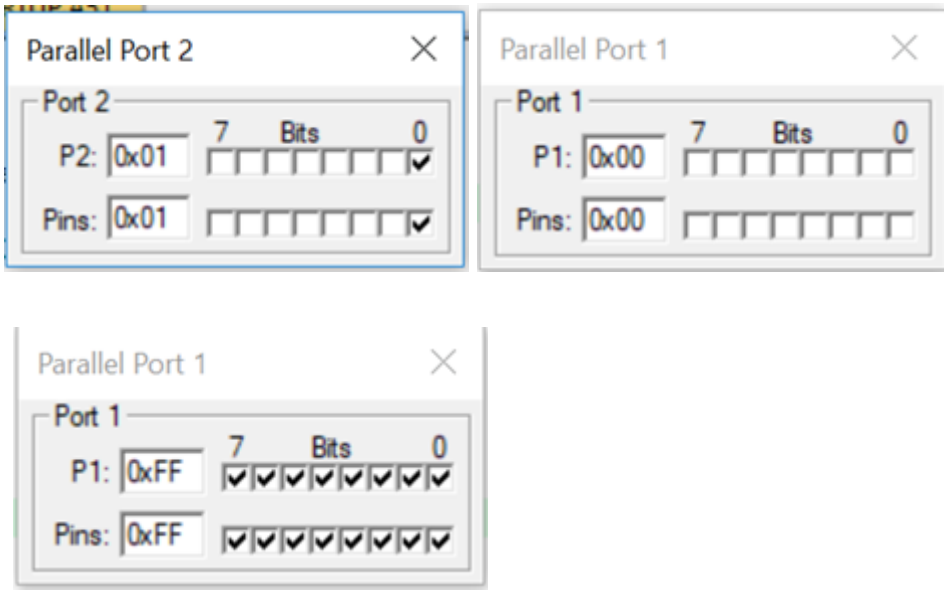
:1008C700E4F508F509C3E5099FE5089E500A0509FF

:0908D700E50970F1050880ED222D

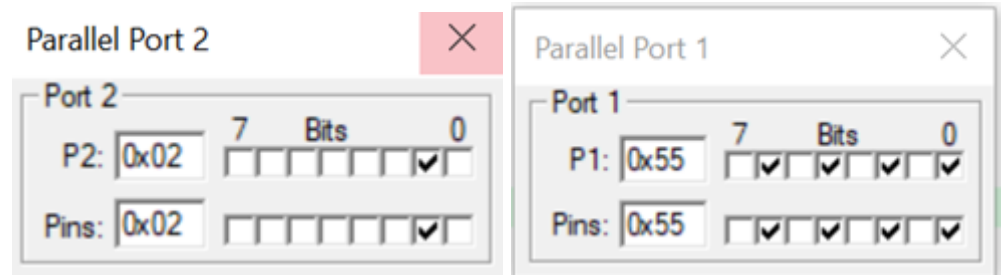
:00000001FF

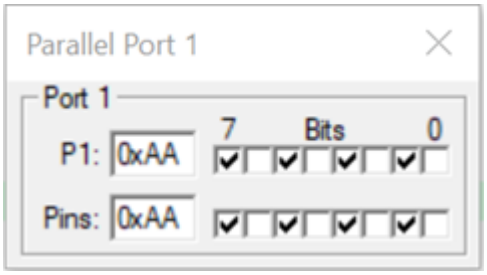
7.4 Simulated Output

Q1.

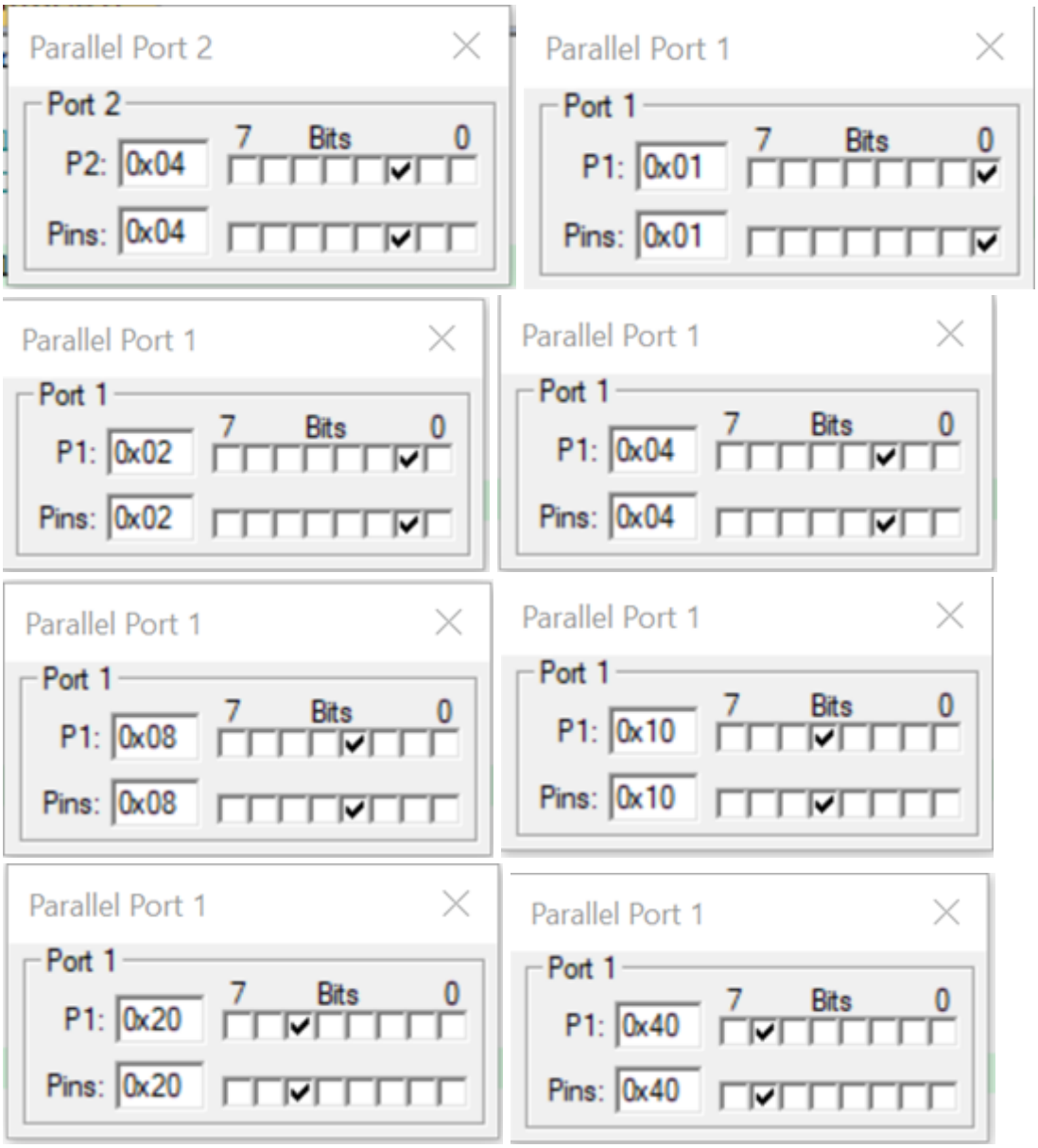


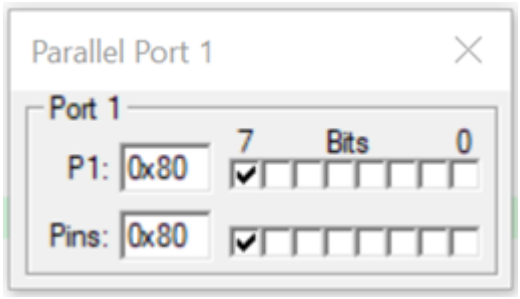
Q2.


Parallel Port 2
Port 2
P2: 0x02
Pins: 0x02
7 Bits 0
Parallel Port 1
Port 1
P1: 0x55
Pins: 0x55
7 Bits 0

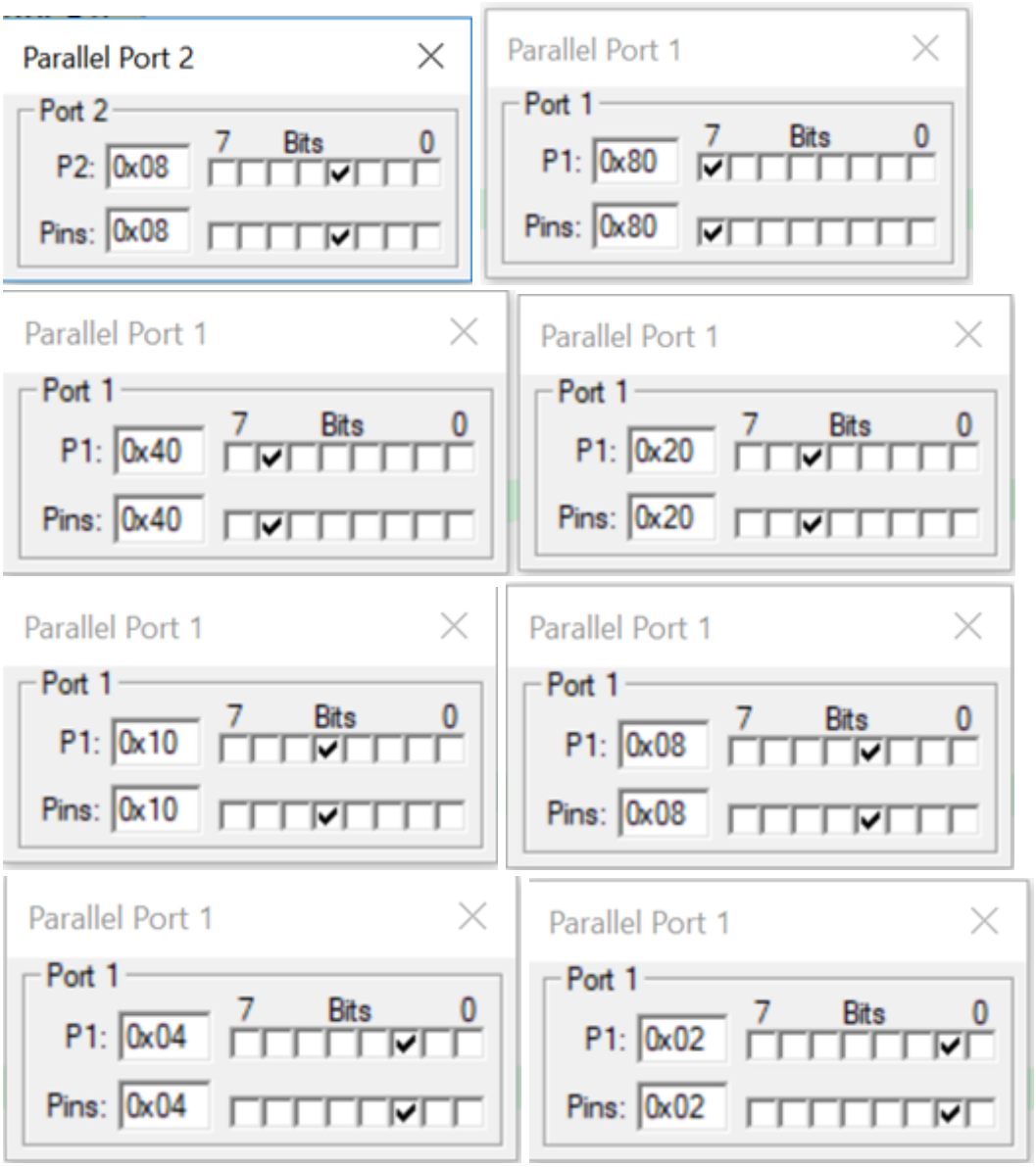

Parallel Port 1
Port 1
P1: 0xAA
Pins: 0xAA
7 Bits 0

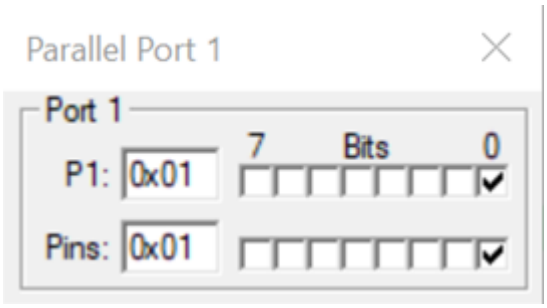
Q3.


Parallel Port 2
Port 2
P2: 0x04
Pins: 0x04
7 Bits 0
Parallel Port 1
Port 1
P1: 0x01
Pins: 0x01
7 Bits 0
Parallel Port 1
Port 1
P1: 0x02
Pins: 0x02
7 Bits 0
Parallel Port 1
Port 1
P1: 0x04
Pins: 0x04
7 Bits 0
Parallel Port 1
Port 1
P1: 0x08
Pins: 0x08
7 Bits 0
Parallel Port 1
Port 1
P1: 0x10
Pins: 0x10
7 Bits 0
Parallel Port 1
Port 1
P1: 0x20
Pins: 0x20
7 Bits 0
Parallel Port 1
Port 1
P1: 0x40
Pins: 0x40
7 Bits 0

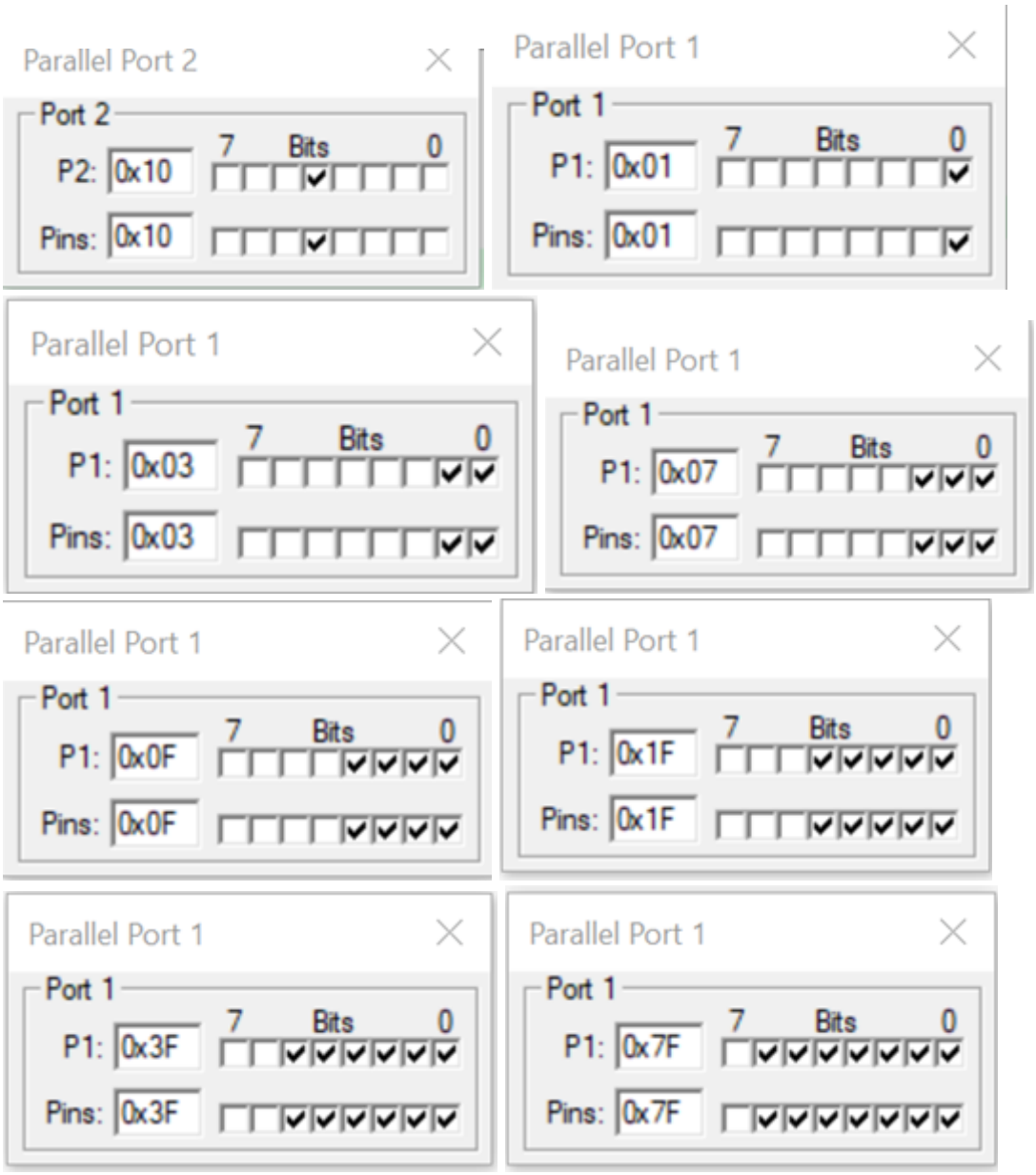

Parallel Port 1
Port 1
P1: 0x80
7 Bits 0
Pins: 0x80

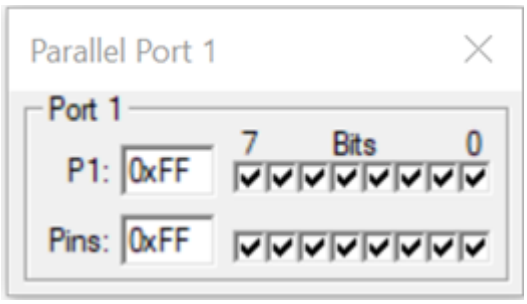
Q4.


Parallel Port 2
Port 2
P2: 0x08
7 Bits 0
Pins: 0x08
Parallel Port 1
Port 1
P1: 0x80
7 Bits 0
Pins: 0x80
Parallel Port 1
Port 1
P1: 0x40
7 Bits 0
Pins: 0x40
Parallel Port 1
Port 1
P1: 0x20
7 Bits 0
Pins: 0x20
Parallel Port 1
Port 1
P1: 0x10
7 Bits 0
Pins: 0x10
Parallel Port 1
Port 1
P1: 0x08
7 Bits 0
Pins: 0x08
Parallel Port 1
Port 1
P1: 0x04
7 Bits 0
Pins: 0x04
Parallel Port 1
Port 1
P1: 0x02
7 Bits 0
Pins: 0x02


Parallel Port 1
Port 1
P1: 0x01
7 Bits 0
Pins: 0x01

Q5.


Parallel Port 2
Port 2
P2: 0x10
7 Bits 0
Pins: 0x10
Parallel Port 1
Port 1
P1: 0x01
7 Bits 0
Pins: 0x01
Parallel Port 1
Port 1
P1: 0x03
7 Bits 0
Pins: 0x03
Parallel Port 1
Port 1
P1: 0x07
7 Bits 0
Pins: 0x07
Parallel Port 1
Port 1
P1: 0x0F
7 Bits 0
Pins: 0x0F
Parallel Port 1
Port 1
P1: 0x1F
7 Bits 0
Pins: 0x1F
Parallel Port 1
Port 1
P1: 0x3F
7 Bits 0
Pins: 0x3F
Parallel Port 1
Port 1
P1: 0x7F
7 Bits 0
Pins: 0x7F



7.5 Conclusion

Hence, all the previous questions have been implemented in this code, its output has been observed on peripherals of software setup via different switches.

8.Hardware implementation

8.1 Aim

Implement the pattern from Q1 to Q5 using switch keys connected to port P2 in real time hardware.

8.2 Code

// implement the pattern from 1 to 5 on hardware

```
#include <reg51.h>
unsigned int i;
unsigned int j;
void delay(unsigned int t);
sbit d=P2^0;  //proj1
sbit r=P2^1;  //proj2
sbit c=P2^2;  //proj3
sbit l=P2^3;  //proj4
sbit u=P2^4;  //proj5
void main()
{
      while(1)
      {
            if(d==0)
            {
                  P1=0xFF;
            delay(10000);
            P1=0x00;
            delay(10000);
            }
      if(r==0)
      {
```

```
          P1=0x55;
          delay(10000);
          P1=~P1;
          delay(10000);
     }
  if(c==0)
     {
          P1=0x01;
          for(j=0;j<8;j++)
          {
     delay(10000);
           P1=P1<<1;
           delay(10000);
          }
     }
     if(l==0)
     {
     P1=0x80;
          for(j=0;j<8;j++)
          {
          for(i=0;i<10000;i++);
          P1=P1>>1;
          for(i=0;i<10000;i++);
     }


      }

     if(u==0)
          {
          P1=0x80;
          for(j=0;j<8;j++)
          {
           delay(10000);
          P1=P1>>1;
        delay(10000);
               P1=P1+0x80;
        }
      }

}
}
void delay(unsigned int t)
{
     for(i=0;i<t;i++);
}

/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Hex Codes of above C file is below        %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

8.3 Hex Code

:030000000208E013

:0C08E000787FE4F6D8FD75810B0208005B

:1008000030A00A7590FF1208C3E41208C130A10C91

:100810007590551208C36390FF1208C330A2237568

:100820009001E4F50AF50B1208C3E59025E01208E3

:10083000C1050BE50B7002050AC39408E50A940094

:1008400040E530A346759080E4F50AF50BE4F50821

:10085000F5090509E50970020508B410F5E508B4C5

:1008600027F0E590C313F590E4F508F5090509E5CF

:100870000970020508B410F5E508B427F0050BE58A

:100880000B7002050A6408450A70C220A40302081E

:1008900000759080E4F50AF50BC3E50B9408E50AB2

:1008A000940040030208001208C3E590C313120825

:1008B000C174802590F590050BE50B70DC050A806E

:0108C000D85F

:0608C100F5907F107E2778

:1008C700E4F508F509C3E5099FE5089E500A0509FF

:0908D700E50970F1050880ED222D

:00000001FF

8.5 Conclusion:

To implement the required various pattern in real time (Hardware Kit), we burned the hex code to the microcontroller kit through SST software and after pressing the RESET button, we were able to observe the corresponding pattern for its designed key press.

9.Blinking of LEDs using timer delay

9.1 Aim

Blink LEDs connected to port 1 for a delay of 1 sec timer 0 mode.

9.2 Code

```
// Blink LEDs connected to port 1 for a delay of 1 sec timer 0 mode.

#include <reg51.h>
#include <stdio.h>
void delay();
unsigned char i;
void main()
{
	TMOD=0x01;

	while(1)
	{
		for(i=0;i<50;i++)
		{
			delay();
		}
		P2=~P2;
	}
}
	void delay()
	{
		TH0=0x87;
		TL0=0xCE;
		TR0=1;
		while(TF0==0);
		TR0=0;
		TF0=0;
	}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to toggle all LEDs connected to port 1              %
%                                                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

9.3 Hex Code

:0300000002082DC6

:0C082D00787FE4F6D8FD75810802080011

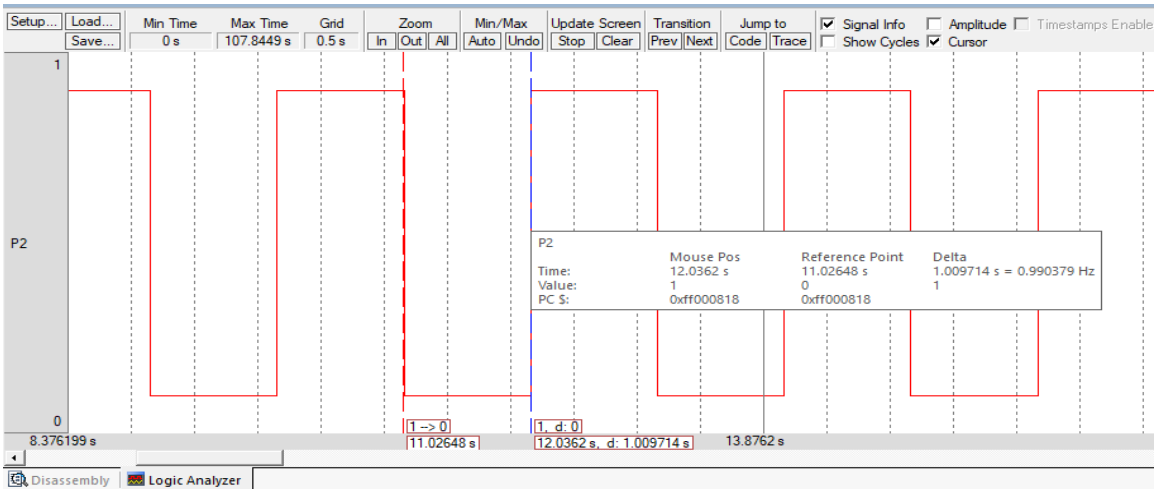
:10080000758901E4F508C3E508943274809480503A

:0D0810000712081D050880EE63A0FF80E6BA

:10081D00758C87758ACED28C308DFDC28CC28D229F

:00000001FF

9.4 Simulated Output



9.5 Conclusion

Hence in this problem we have blinked the LED's at an interval on 1s. This was achieved by operating the timer 0 in mode 1.

10.Pattern implementation using timer

10.1 Aim

Implement the pattern from 1 to 5, using switch key connected to port P2 with timer delay (Use timer in mode 1).

10.2 Code

// Implement the pattern from 1 to 5 using switch key connected to port P2 with timer delay(using timer 2 mode 1)

```
#include<reg51.h>
unsigned int i;
sbit down=P2^0;
sbit right=P2^1;
sbit center=P2^2;
sbit left=P2^3;
sbit up=P2^4;
void delay()
{
          for(i=0;i<50;i++)
          {
                    TH0=0x88;
                    TL0=0x00;
                    TR0=1;
                    while(TF0==0);
                    TF0=0;
                    TR0=0;
          }
}
void main()
{
          TMOD=0x01;
          P2=0x00;
          while(1)
          {
                    if(down==1)
                    {
                              P1=0xAA;
                              delay();
                              P1=0x55;
                              delay();
                    }
                    else if(right==1)
                    {
                              if(P1==0x00||P1==0x80)
                              {
```

```
                    P1=0x01;
                    delay();
          }
          P1=P1<<1;
    delay();
  }
  else if(center==1)
  {
          if(P1==0x01||P1==0x00)
          {
                    P1=0x80;
                    delay();
          }
          P1=P1>>1;
    delay();
  }
  else if(left==1)
  {
          P1=(P1<<1)+1;
          delay();
          if(P1==0xff)
          {
                    P1=0x01;
                    delay();
          }
  }
  else if(up==1)
  {
          P1=(P1>>1)+0x80;
          delay();
          if(P1==0xff)
          {
                    P1=0x80;
                    delay();
          }
     }
}
}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to implement the pattern from 1 to 5     %
% using timer                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

10.3 Hex Code

:10088500E4F508F509758C88E4F58AD28C308DFD80

:10089500C28DC28C0509E5097002050864324 50858

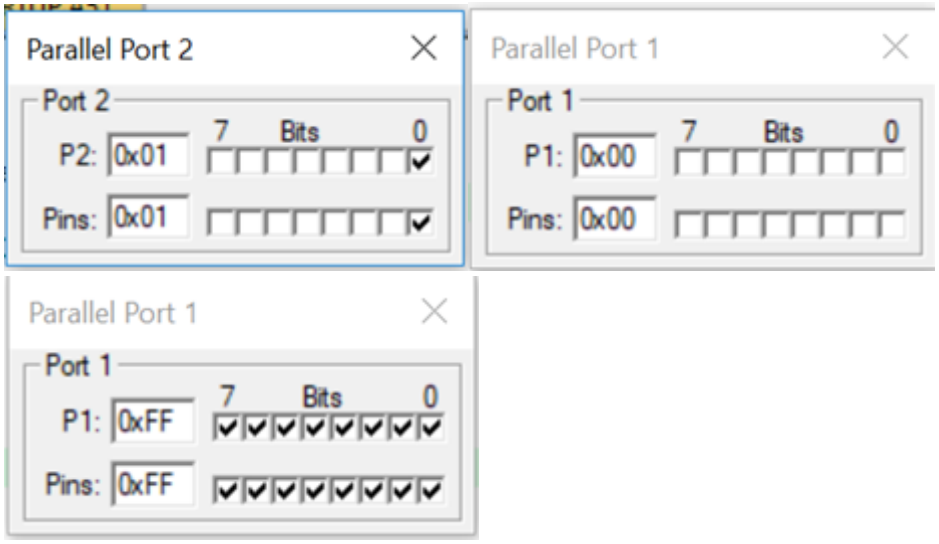
:0308A50070E322DB

:10080000758901E4F5A030A00B7590AA12088575D2

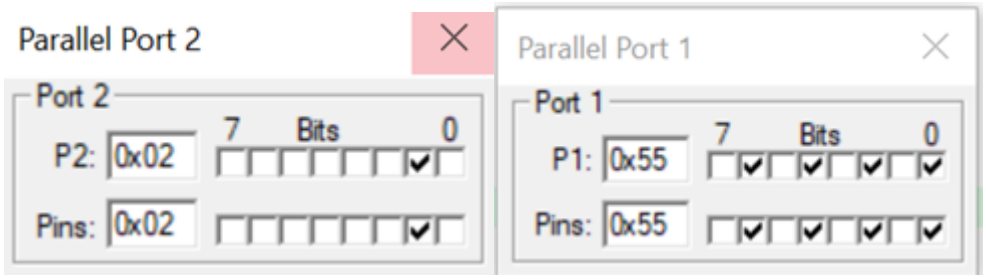
:10081000905580503 0A11AAF90EF6007AF90EF6411
:10082000807006759001120885E59025E0F59080AE
:100830003330A21AAF90EF64016005AF90EF7006FD
:10084000759080120885E590C313F590801630A34B
:1008500018E59025E004F590120885AF90EFF4704C
:10086000A5759001120885809D30A49AE590C31368
:100870002480F590120885AF90EFF4708975908010
:050880001208858081D3
:030000000208A84B
:0C08A800787FE4F6D8FD75810902080095
:00000001FF

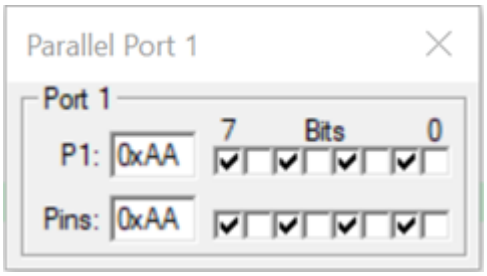
10.4 Simulated Output

Q1

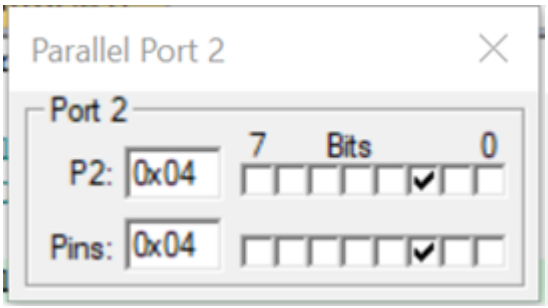


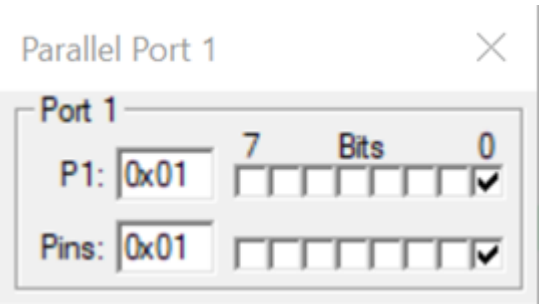
Q2

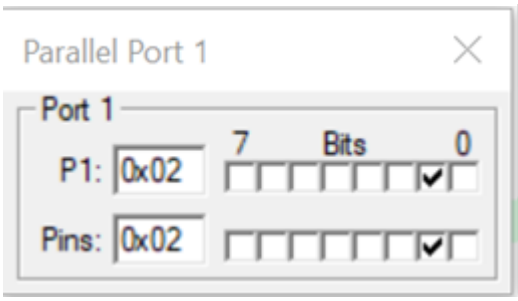


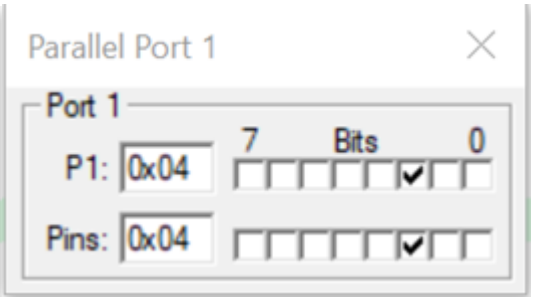

Parallel Port 1
Port 1
P1: 0xAA
7 Bits 0
Pins: 0xAA

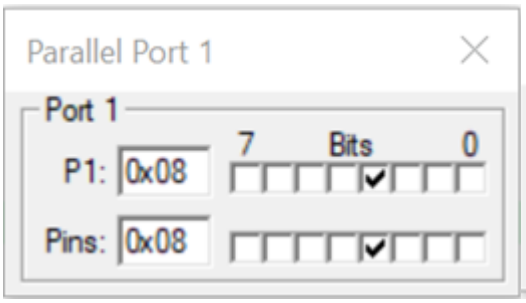
Q3

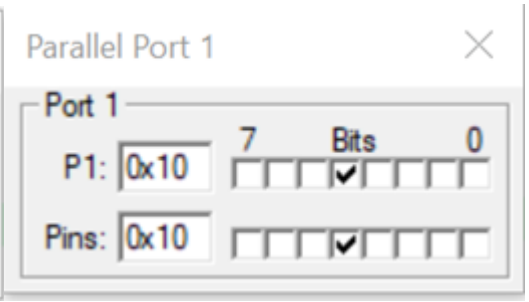

Parallel Port 2
Port 2
P2: 0x04
7 Bits 0
Pins: 0x04

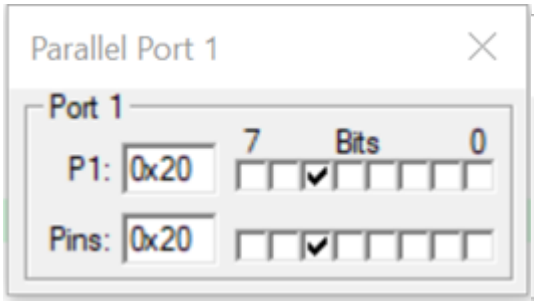

Parallel Port 1
Port 1
P1: 0x01
7 Bits 0
Pins: 0x01

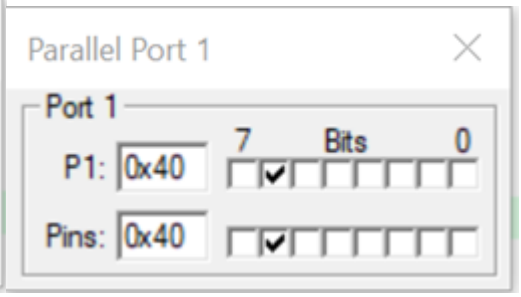

Parallel Port 1
Port 1
P1: 0x02
7 Bits 0
Pins: 0x02

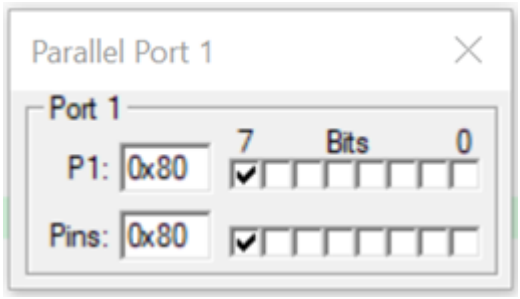

Parallel Port 1
Port 1
P1: 0x04
7 Bits 0
Pins: 0x04


Parallel Port 1
Port 1
P1: 0x08
7 Bits 0
Pins: 0x08

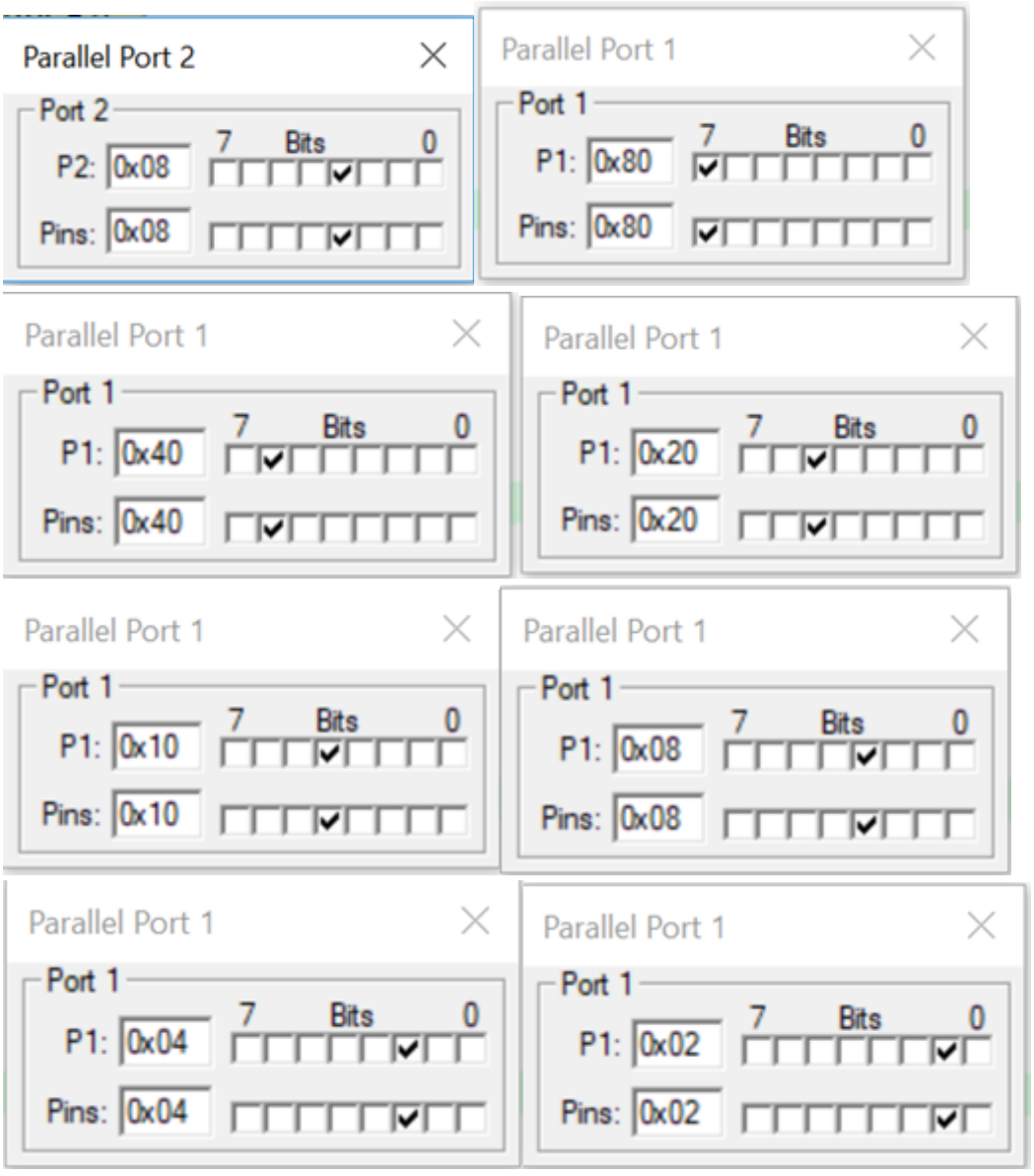

Parallel Port 1
Port 1
P1: 0x10
7 Bits 0
Pins: 0x10

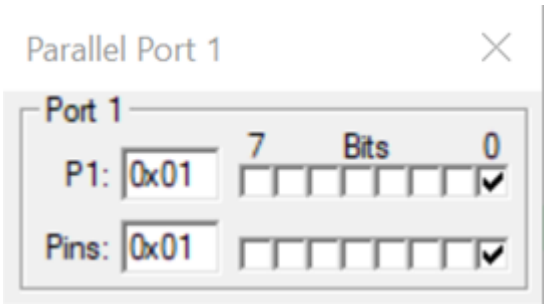

Parallel Port 1
Port 1
P1: 0x20
7 Bits 0
Pins: 0x20


Parallel Port 1
Port 1
P1: 0x40
7 Bits 0
Pins: 0x40

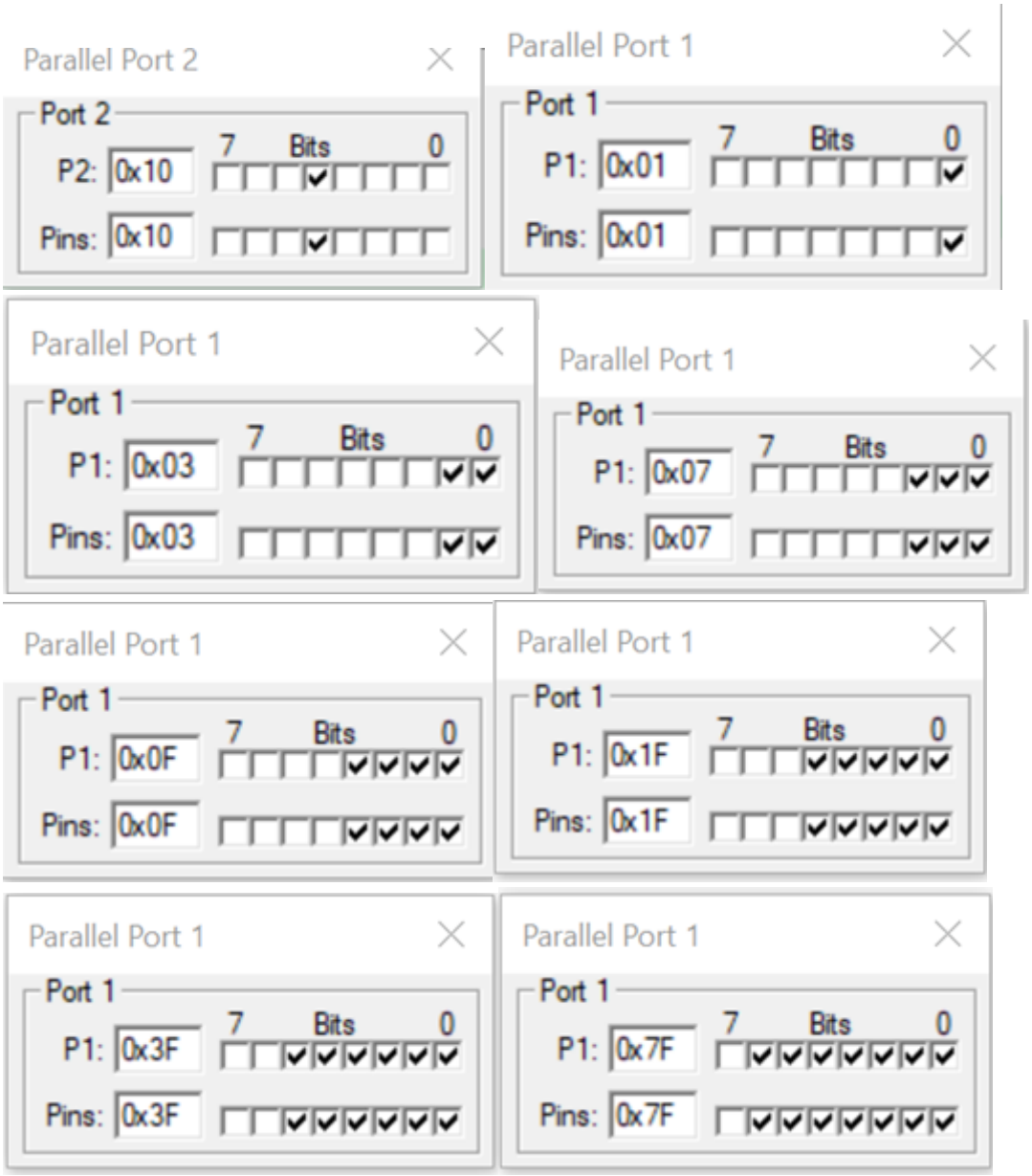

Parallel Port 1
Port 1
P1: 0x80
7 Bits 0
Pins: 0x80

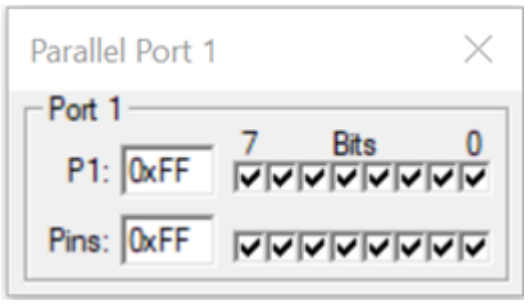
Q4


Parallel Port 2
Port 2
P2: 0x08
7 Bits 0
Pins: 0x08
Parallel Port 1
Port 1
P1: 0x80
7 Bits 0
Pins: 0x80
Parallel Port 1
Port 1
P1: 0x40
7 Bits 0
Pins: 0x40
Parallel Port 1
Port 1
P1: 0x20
7 Bits 0
Pins: 0x20
Parallel Port 1
Port 1
P1: 0x10
7 Bits 0
Pins: 0x10
Parallel Port 1
Port 1
P1: 0x08
7 Bits 0
Pins: 0x08
Parallel Port 1
Port 1
P1: 0x04
7 Bits 0
Pins: 0x04
Parallel Port 1
Port 1
P1: 0x02
7 Bits 0
Pins: 0x02


Parallel Port 1
Port 1
P1: 0x01
7 Bits 0
Pins: 0x01

Q5


Parallel Port 2
Port 2
P2: 0x10
7 Bits 0
Pins: 0x10
Parallel Port 1
Port 1
P1: 0x01
7 Bits 0
Pins: 0x01
Parallel Port 1
Port 1
P1: 0x03
7 Bits 0
Pins: 0x03
Parallel Port 1
Port 1
P1: 0x07
7 Bits 0
Pins: 0x07
Parallel Port 1
Port 1
P1: 0x0F
7 Bits 0
Pins: 0x0F
Parallel Port 1
Port 1
P1: 0x1F
7 Bits 0
Pins: 0x1F
Parallel Port 1
Port 1
P1: 0x3F
7 Bits 0
Pins: 0x3F
Parallel Port 1
Port 1
P1: 0x7F
7 Bits 0
Pins: 0x7F



10.5 Conclusion:

We made different functions for each pattern so that whenever this key is pressed, its corresponding function gets called and executes the specific task to get the desired pattern for that key press. We used timer 0 in mode 1 to generate time delay of 1 second between two statuses of the LED. And to do this task continuously, we used while loop with condition which is always true.

11.Generation of PWM signals

11.1 Aim

Generate following PWM signal as mentioned below using timer 1 mode 1 (polling method)

a) P2.1=1, Generate PWM signal of frequency 1kHz and duty cycle = 80% on P1.1

b) P2.2=1, Generate PWM signal of frequency 2kHz and duty cycle = 20 % on P1.2

c) P2.3=1 , Generate PWM signal of frequency 500Hz and duty cycle = 40% on P1.3

11.2 Code

//Toggle all LEDs alternatively on port p1 with some delay

```
# include <stdio.h>
# include <reg51.h>
sbit sw1=P2^1;
sbit sw2=P2^2;
sbit sw3=P2^3;
sbit  pin1=P1^1;
sbit  pin2=P1^2;
sbit  pin3=P1^3;
void delay(unsigned char R );
void main()
{ while(1)
                {
  TMOD=0x10;
                pin1=0;
  pin2=0;
                pin3=0;
                if(sw1==1)
                {
                                delay(2);
                                pin1=1;
                                delay(8);
                                pin1=0;
                }
                if(sw2==1)
                {
                  delay(4);
                                pin2=1;
                                delay(1);
```

```
                              pin2=0;
               }
               if(sw3==1)
               {
                              delay(12);
                              pin3=1;
                              delay(8);
                              pin3=0;
               }
}
}
void delay(unsigned int R )
{
while(R!=0)
{
               TH1=0xFE;
               TL1=0xE0;
               TR1=1;
               while(TF1==0);
               TR1=0;
               TF1=0;
  R--;
}
}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to to generate PWM  waves.          %
%                                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

11.3 Hex Code

:030000000208599A

:0C085900787FE4F6D8FD7581070208 00E6

:10080000758910C291C292C29330A10E7F02120864

:100810003ED2917F0812083EC29130A20E7F041290

:10082000083ED2927F0112083EC29230A3D27F0CC2

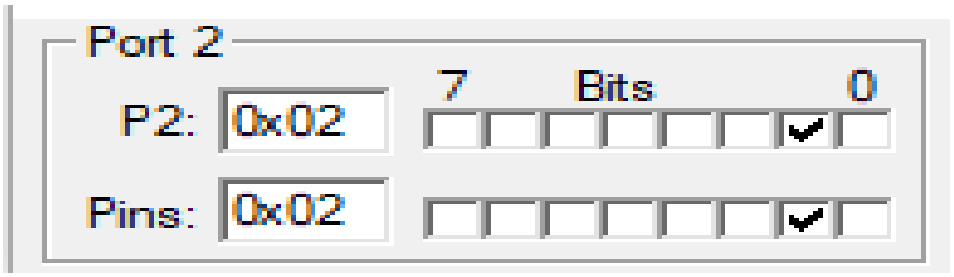
:0E08300012083ED2937F0812083EC29380C287

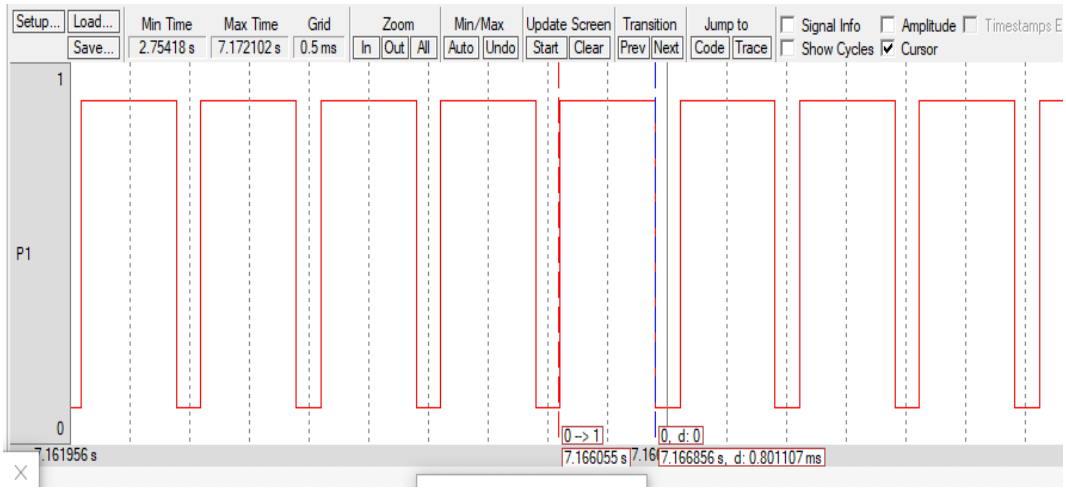
:10083E00EF4E6016758DFE758BE0D28E308FFDC239

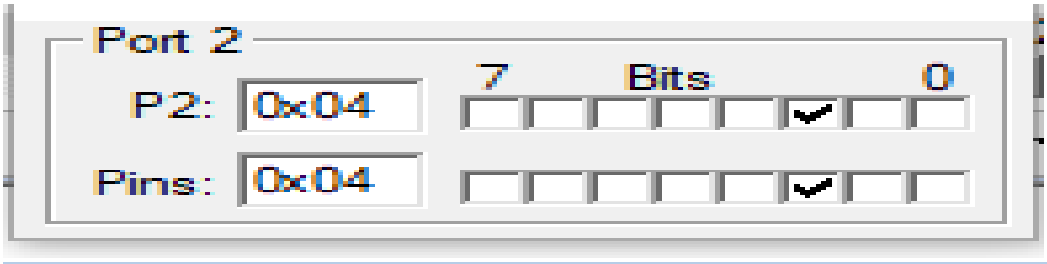
:0B084E008EC28FEF1F70E91E80E622B3

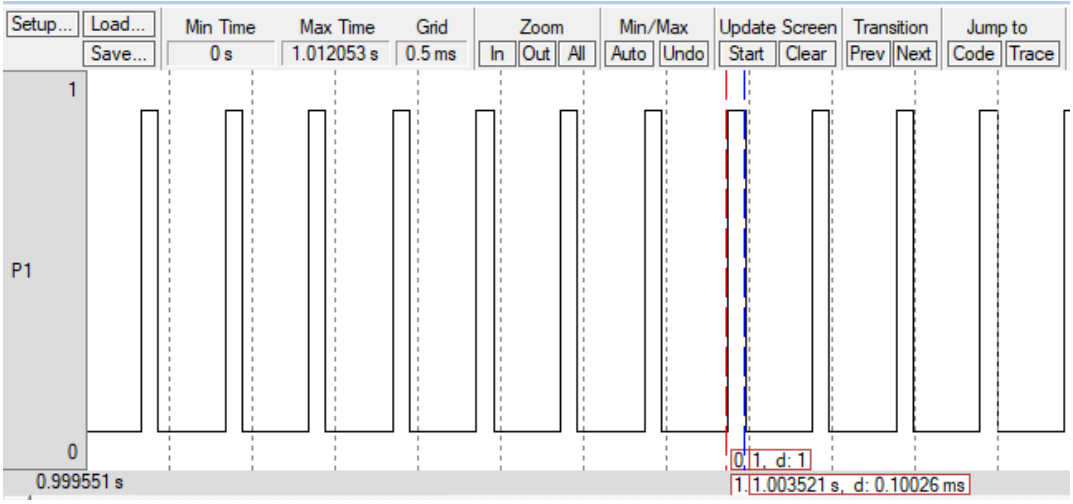
:00000001FF

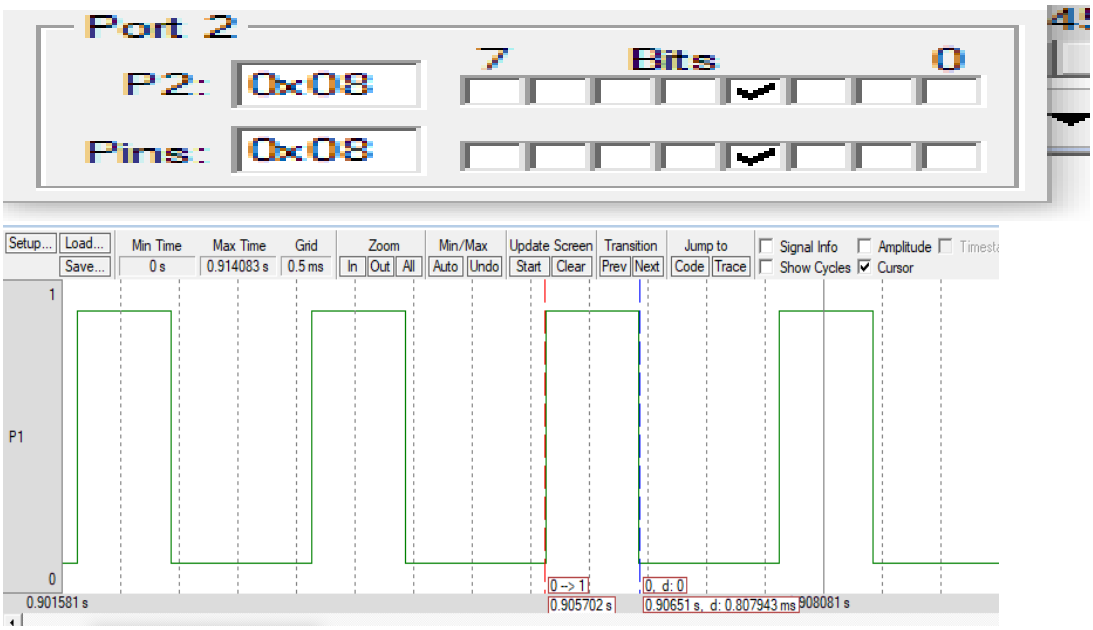
11.4 Simulated Output


Port 2
P2: 0x02
Pins: 0x02
7 Bits 0


Setup... Load... Save...
Min Time 2.75418 s
Max Time 7.172102 s
Grid 0.5 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Start Clear
Transition Prev Next
Jump to Code Trace
Signal Info
Amplitude
Timestamps E
Show Cycles
Cursor
1
P1
0
0 -> 1
0, d: 0
7.161956 s
7.166055 s
7.166856 s, d: 0.801107 ms


Port 2
P2: 0x04
Pins: 0x04
7 Bits 0


Setup... Load... Save...
Min Time 0 s
Max Time 1.012053 s
Grid 0.5 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Start Clear
Transition Prev Next
Jump to Code Trace
1
P1
0
0, 1, d: 1
0.999551 s
1, 1.003521 s, d: 0.10026 ms



11.5 Conclusion

Hence, we have generated PWM signal of different duty cycle on port 1 based on switching condition of port 2 bits, results have been shown in above figures for different case.

12.DELAY OF 1msec

12.1 Aim

Generate Delay of 1ms using timer 0 interrupt in mode 1.

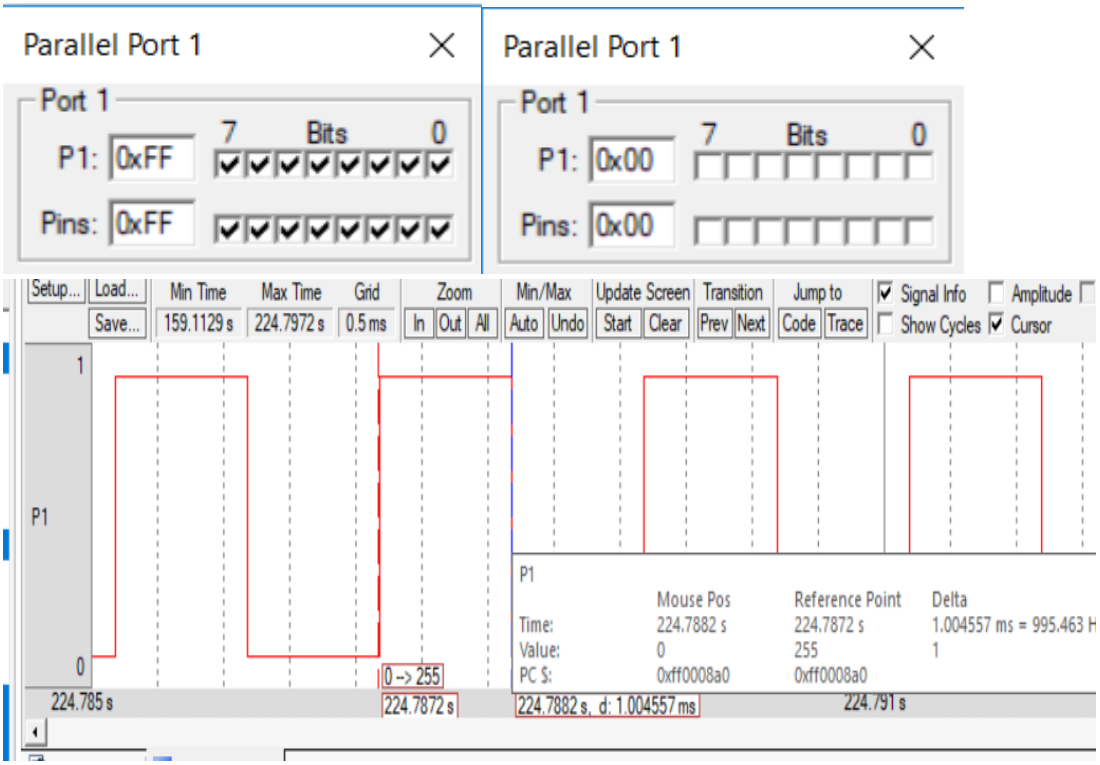
12.2 Code

```
//Generate Delay of 1ms using timer 0 interrupt in mode 1.
//Generate Delay of 1ms using timer 0 interrupt in mode 1.
// for the observation of delay square wave signal is generated at port P1.1

#include <stdio.h>
#include <reg51.h>
#define MSB 0xF4
#define LSB 0x00
bit flag1=0;
void main()
{
		TMOD=0x01;
		EA=1;
		ET0=1;
		TH0=MSB;
		TL0=LSB;
		TR0=1;
		while(1)
		{
			if(flag1==1)
			{
				flag1=0;
				P1=~P1;

			}

		}

}
void delay(void) interrupt 1
{

		flag1=1;
		TH0=MSB;
		TL0=LSB;
		TR0=1;

}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to generate delay of 1ms.          %
%                                                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

12.3 Hex Code

```
:03000000020800F3
:0C080000787FE4F6D8FD758120020847DF
:0208B000C10085
:10088C00758901D2AFD2A9758CF4E4F58AD28C307B
:09089C0000FDC2006390FF80F62C
:03000B000208A543
:0B08A500D200758CF4758A00D28C32F2
:10080C0002088CE493A3F8E493A34003F68001F26E
:10081C0008DFF48029E493A3F85407240CC8C333ED
:10082C00C4540F4420C8834004F456800146F6DFBC
:10083C00E4800B01020408102040809008B0E47E94
:10084C00019360BCA3FF543F30E509541FFEE493B1
:10085C00A360010ECF54C025E060A840B8E493A378
:10086C00FAE493A3F8E493A3C8C582C8CAC583CAA3
:10087C00F0A3C8C582C8CAC583CADFE9DEE780BE5B
:0108B2000045
:00000001FF
```

12.4 Simulated Output



12.5 Conclusion

Hence, the delay of 1 msec has been generated using timer 0 interrupt, observation of delay generated has been shown in analyser by generating square wave signal at port 1.

13.DELAY OF 1sec

13.1 Aim

Generate Delay of 1s using timer 1 interrupt in mode 1.

13.2 Code

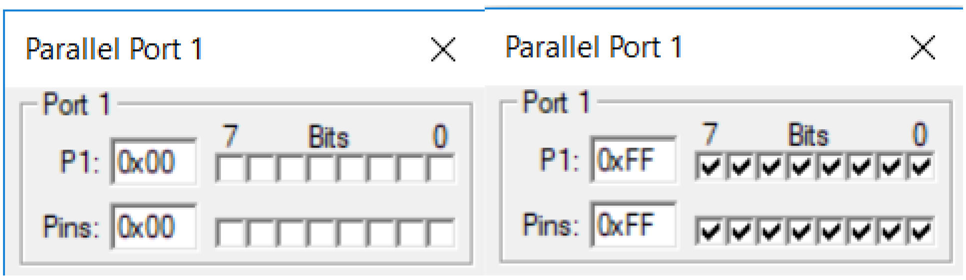
```
//Generate Delay of 1s using timer 1 interrupt in mode 1.
// for the observation of delay square wave signal is generated at port P1.1
//20ms*50=1 sec

#include <stdio.h>
#include <reg51.h>
#define MSB 0x10
#define LSB 0x00
bit flag1=0;
unsigned char R=50;
void main()
{
                TMOD=0x10;
                EA=1;
                ET1=1;
                TH1=MSB;
                TL1=LSB;
                TR1=1;
                while(1)
                {
                                if(flag1==1)
                                {
                                                flag1=0;
                                                P1=~P1;

                                }

                }

}
void delay(void) interrupt 3
{
                if(R==0)
                {
```

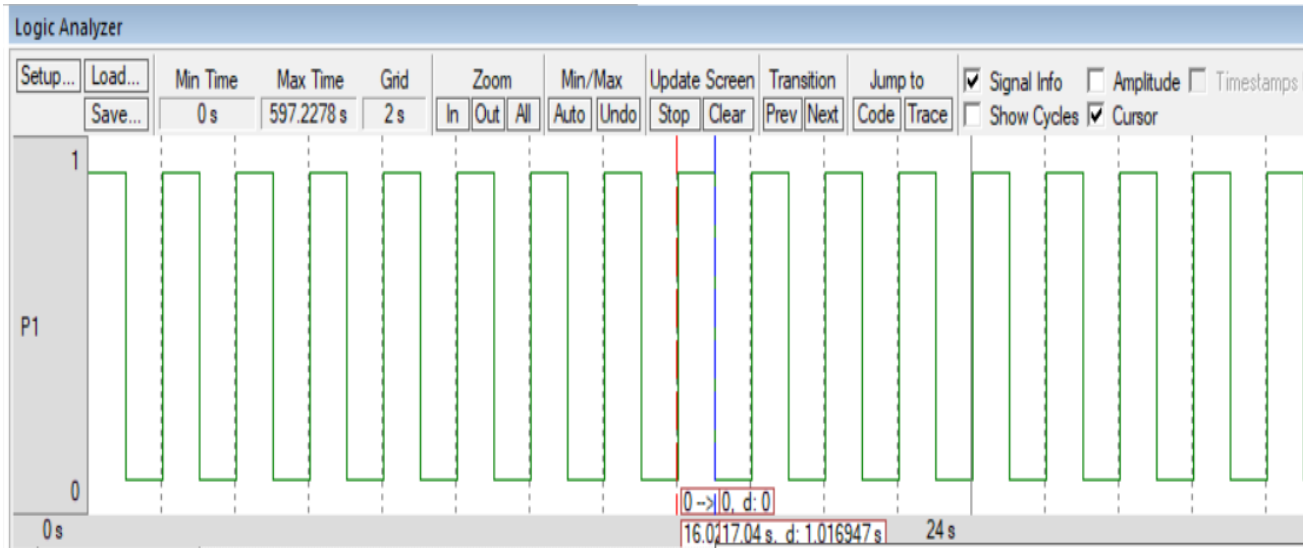
```
                    flag1=1;
                    R=50;
                    }
                    TH1=MSB;
                    TL1=LSB;
                    TR1=1;
                    R--;
}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to generate Delay of 1s using timer 1 interrupt in
mode 1.          %
%                                                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

13.3 Hex Code

:03000000020800F3
:0C080000787FE4F6D8FD758120020847DF
:0508BD00C1000108323A
:10088C00758910D2AFD2AB758D10E4F58BD28E304A
:09089C0000FDC2006390FF80F62C
:03001B000208A533
:1008A500C0E0E5087005D200750832758D10758BAE
:0808B50000D28E1508D0E032DC
:10080C0002088CE493A3F8E493A34003F68001F26E
:10081C0008DFF48029E493A3F85407240CC8C333ED
:10082C00C4540F4420C8834004F456800146F6DFBC
:10083C00E4800B0102040810204080900 8BDE47E87
:10084C00019360BCA3FF543F30E509541FFEE493B1
:10085C00A360010ECF54C025E060A840B8E493A378
:10086C00FAE493A3F8E493A3C8C582C8CAC583CAA3
:10087C00F0A3C8C582C8CAC583CADFE9DEE780BE5B
:0108C2000035
:00000001FF

13.4 Simulated Output





13.5 Conclusion

Hence, the delay of 1 sec has been generated using timer 1 interrupt, observation of delay generated has been shown in analyser by generating square wave signal at port 1.

14. DELAY OF 3sec

14.1 Aim

Generate Delay of 3s using timer 0 interrupt in mode 1.

14.2 Code

```
///Generate Delay of 1s using timer 1 interrupt in mode 1.
// for the observation of delay square wave signal is generated at port P1.1
//20ms*150=3sec

#include <stdio.h>
#include <reg51.h>
#define MSB 0x10
#define LSB 0x00
bit flag1=0;
unsigned char R=150;
void main()
{
                TMOD=0x01;
                EA=1;
                ET0=1;
                TH0=MSB;
```

```
		TL0=LSB;
		TR0=1;
		while(1)
		{
				if(flag1==1)
				{
						flag1=0;
						P1=~P1;

				}

		}

}
void delay(void) interrupt 1
{
		if(R==0)
		{
		flag1=1;
		R=150;
		}
		TH0=MSB;
		TL0=LSB;
		TR0=1;
		R--;
}
/*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function file is created to generate delay of 3 sec.         %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*/
```

14.3 Hex Code

:03000000020800F3

:0C080000787FE4F6D8FD758120020847DF

:0508BD00C100010896D6

:10088C00758901D2AFD2A9758C10E4F58AD28C305F

:09089C0000FDC2006390FF80F62C

:03000B000208A543

:1008A500C0E0E5087005D200750896758C10758A4C

:0808B50000D28C1508D0E032DE

:10080C0002088CE493A3F8E493A34003F68001F26E

:10081C0008DFF48029E493A3F85407240CC8C333ED

:10082C00C4540F4420C8834004F456800146F6DFBC

:10083C00E4800B0102040810204080900 8BDE47E87

:10084C00019360BCA3FF543F30E509541FFEE493B1

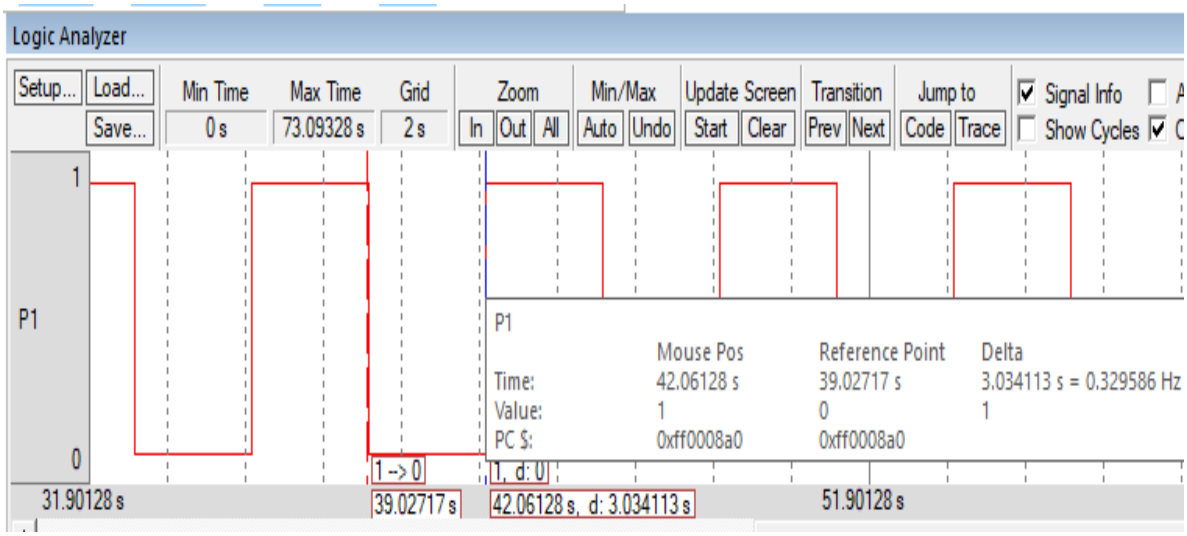
:10085C00A360010ECF54C025E060A840B8E493A378

:10086C00FAE493A3F8E493A3C8C582C8CAC583CAA3

:10087C00F0A3C8C582C8CAC583CADFE9DEE780BE5B

:0108C2000035
:00000001FF

14.4 Simulated Output



14.6 Conclusion

Hence, the delay of 3 sec has been generated using timer 0 interrupt, observation of delay generated has been shown in analyser by generating square wave signal at port 1.

15 LCD Interfacing

15.1 Aim

Interface the LCD with 8051 micro controller and display your Name and Roll number in first and second line respectively.

15.2 Code

```
#include<reg51.h>
voidlcd_init(void);
voidlcd_cmd(unsigned char command);
voidlcd_data(unsigned char disp_data);
void delay(unsigned int t);
unsigned int i;
sbit RS=P2^7;
sbit RW=P2^6;
sbit e=P2^5;
void main()
{
Unsigned char a[6]="SNEH SANI DHRUV";
Unsigned char b[8]="2001047 2001125 2001003";
while(1)
{
lcd_init();
lcd_cmd(0x80); force cursor to blink on first line
for(i=0;i<7;i++)
{
lcd_data(a[i]);
delay(500);
lcd_cmd(0x06);
}
lcd_cmd(0xc0); force cursor to blink on second  line
for(i=0;i<8;i++)
{
lcd_data(b[i]);
delay(500);
lcd_cmd(0x06);  //cursor shift
}
}
}
Void lcd_cmd(unsigned  char command)
{
    P0=command;
```

```
    RS=0;
    RW=0;
    e=1;
    delay(10);
    e=0;
}
Void lcd_data(unsigned char disp_data)
{
    P0=disp_data;
    RS=1;
    RW=0;
    e=1;
    delay(10);
    e=0;
}
Void lcd_init()
{
lcd_cmd(0x38);   //turn on
    delay(10);
lcd_cmd(0xF0) or 0x0F; //clear display
    delay(10);
lcd_cmd(0x01);  //cursor glow
    delay(10);
}
void delay(unsigned int t)
{
Unsigned intj,k ;
for(k=0;k<t;k++)
{
for(j=0;j<100;j++);
}
}
```

15.3 Hex Code

```
:0F09C90048454D414E5448553136454530363O3E
:1008F60078087C007D007BFF7A0979C97E007F0736
:100906001208D0780F7C007D007BFF7A0979D07EB3
:10091600007F081208D01209AD7F801209E4E4F5C1
:1009260017F51874082518F8E6FF1209EE7FF47E0D
:100936000112098D7F061209E40518E518700205F3
:1009460017C39407E517940040D97FC01209E4E461
:10095600F517F518C3E5189408E517940050B77411
:100966000F2518F8E6FF1209EE7FF47E0112098DB5
:0F0976007F061209E40518E51870D9051780D51A
:0A09E4008F80C2A7120985C2A52268
:0A09EE008F80D2A7120985C2A5224E
:1009AD007F381209E41209897F0F1209E4120989AF
```

```
:0C09BD007F011209E47F0A7E0002098D10
:08098500C2A6D2A57F0A7E0084
:10098D00E4FDFCC3ED9FEC9E5015E4FBFA0BBB00A0
:0F099D00010AEB64644A70F50DBD00010C80E4A3
:0109AC002228
:030000000209D81A
:0C09D800787FE4F6D8FD7581180208F65F
:10080000E709F608DFFA8046E709F208DFFA803EDA
:1008100088828C83E709F0A3DFFA8032E309F608C7
:10082000DFFA8078E309F208DFFA807088828C832F
:10083000E309F0A3DFFA806489828A83E0A3F608E3
:10084000DFFA805889828A83E0A3F208DFFA804CBD
:1008500080D280FA80C680D4806980F28033801094
:1008600080A680EA809A80A880DA80E280CA8033FD
:1008700089828A83ECFAE493A3C8C582C8CCC58375
:10088000CCF0A3C8C582C8CCC583CCDFE9DEE78045
:100890000D89828A83E493A3F608DFF9ECFAA9F0C4
:1008A000EDFB2289828A83ECFAE0A3C8C582C8CC1A
:1008B000C583CCF0A3C8C582C8CCC583CCDFEADE33
:1008C000E880DB89828A83E493A3F208DFF980CC95
:1008D00088F0EF60010E4E60C388F0ED2402B4048E
:1008E0000050B9F582EB2402B4040050AF23234535
:0608F00082239008507302
:00000001FF
```

15.6 Conclusion

In this experiment we have learnt how to interface an LCD using 8051 controller. We use init function to initialize LCD, cmd function to pass a command to LCD, data function to send data to be displayed to LCD. RS, RW & e are control signals of LCD.

16. Use timer 0 in Auto reload mode (use timer in Interrupt mode)

16.1 Aim :

Generate a PWM of 1kHz and 70 % duty cycle at P1, Using timer 0 in Auto reload mode (use timer in Interrupt mode)

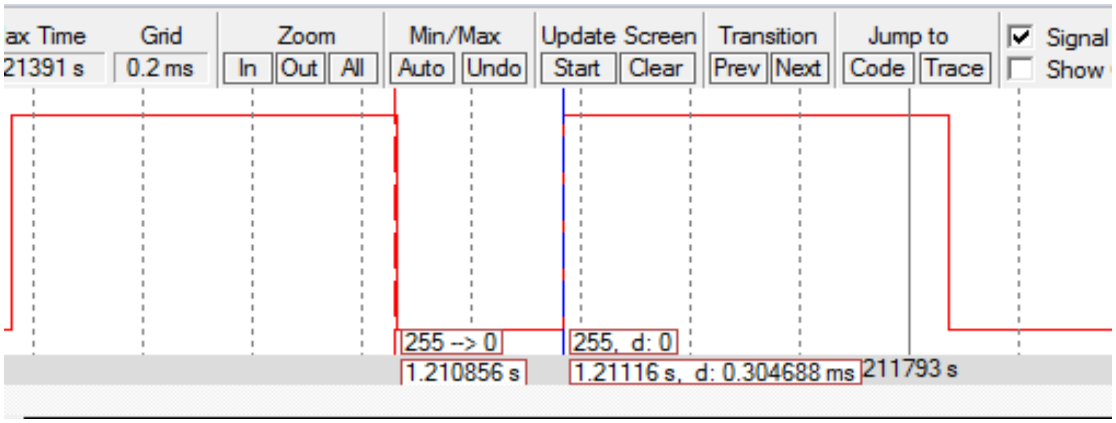
16.2 Code :

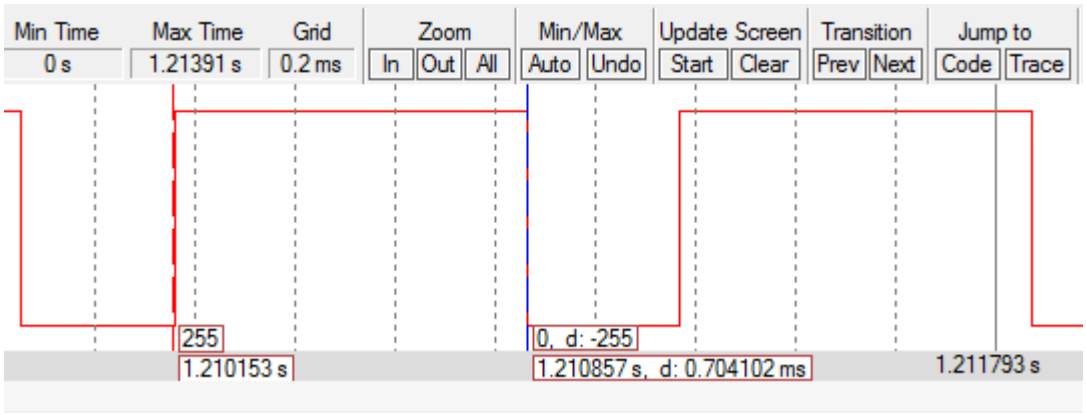
```
#include <reg51.h>
void delay(void);
unsigned int x;
void main()
{
  TMOD=0x02;
    IE=0x82;
    TH0=0xE1;
    TR0=1;
    while(1)
    {
        if(x==70)
        {
        P1=0x00;
        }
        if(x==100)
        {
        P1=0xFF;
        x=0;
      }
    }
}
void isr(void)interrupt 1
{
    x++;
}
```

***16.3 Hex Code* :**

```
:03000000020833C0
:0C083300787FE4F6D8FD7581090208000A
:1008000075890275A882758CE1D28CE5096446452C
:10081000087002F590E5096464450870EE7590FF74
:06082000F508F50980E572
:03000B00020826C2
:0D082600C0E00509E50970020508D0E032C8
:00000001FF
```

16.*4* **Simulated Output :**





16.6 Conclusion :

We used timer 0 in auto reload mode (interrupt mode) to generate the PWM of 70 percentage duty cycle

17. **Blink the LED when an external signal received**

17.1 Aim :

Blink the led connected to port1.4 when an eternal signal received by 8051 controller.

17.2 Code :

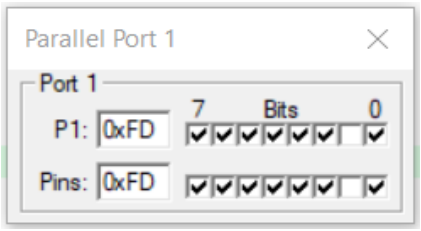
```
#include <reg51.h>
sbit a=P1^1;
sbit b=P1^4;
unsigned char flag=0;
long int i,j;
void main()
{
    EA=1;
    EX0=1;

    IT0=1;
    while(1)
    {
        for(i=1;i<100;i++)
        {
            for(j=1;j<100;j++)
            {
            a=~a;
            }
        }
        if (flag==1)
        {
            flag=0;
            b=~b;
        }
    }
}

void ex_int(void) interrupt 0
{
    flag=1;
}
```

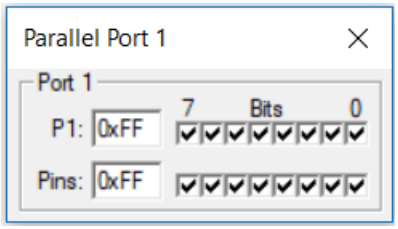
17.3 Hex Code :

```
:03000000020800F3
:0C080000787FE4F6D8FD758110020847EF
:03092C00010800BF
:10088C00D2AFD2A8D288E4750C01F50BF50AF509A4
:10089C00E47F64FEFDFCAB0CAA0BA90AA809C312E9
:1008AC0009125056E4751001F50FF50EF50DE47FA5
```

```
:1008BC0064FEFDFCAB10AA0FA90EA80DC312091201
:1008CC00501DB291E4FAF9F8E5102401F510EA355F
:1008DC000FF50FE9350EF50EE8350DF50D80CFE46B
:1008EC00FAF9F8E50C2401F50CEA350BF50BE935B2
:1008FC000AF50AE83509F5098096E5086401708661
:06090C00F508B2948080A2
:03000300020928C7
:04092800750801321B
:10080C0002088CE493A3F8E493A34003F68001F26E
:10081C0008DFF48029E493A3F85407240CC8C333ED
:10082C00C4540F4420C8834004F456800146F6DFBC
:10083C00E4800B01020408102040809092CE47E17
:10084C00019360BCA3FF543F30E509541FFEE493B1
:10085C00A360010ECF54C025E060A840B8E493A378
:10086C00FAE493A3F8E493A3C8C582C8CAC583CAA3
:10087C00F0A3C8C582C8CAC583CADFE9DEE780BE5B
:01092F0000C7
:10091200EB9FF5F0EA9E42F0E99D42F0EC6480C85C
:0609220064809845F022FC
:00000001FF
```

17.4 Simulated Output :





***17.5 Conclusion* :**

We toggled the LED whenever negative edge of external interrupt is coming. For that we set the flag in external interrupt ISR and took the toggling action in while loop.

18 Measuring DC Voltage

18.1 Aim :

Generate a 8051 Code for measuring DC voltage, also display the measured voltage Using 16*2 LCD (use ADC0808).

18.2 Code :

```
#include <reg51.h>
#include <intrins.h>
#define display_port P0
sbit ALE=P3^1;
sbit EOC=P3^2;
sbit OE=P3^3;
sbit SOC=P3^4;
sbit CS_A=P3^5;
sbit CS_B=P3^6;
sbit CS_C=P3^7;
unsigned int a,b,c,d,e,f,g,h,i,j,k,V[5];
void ADC_INI(void);
void ADC_READ(void);
void calculate(unsigned int u);
sbit RS=P2^7;
sbit RW=P2^6;
sbit EN=P2^5;
void DISP_LCD(void);
void lcd_ini(void);
void lcd_cmd(unsigned char command);
void lcd_data(unsigned char dispdata);
void delay (unsigned int t);
void main()
{
    ADC_INI();
    while(1)
    {
        ADC_READ();
       DISP_LCD();
    }
}
void ADC_INI()
{
    P1=0xFF;
    ALE=0;
    SOC=0;
    OE=0;
    EOC=1; //because we are making EOC as input pin
```

```
    delay(1);
    return;
}
void ADC_READ()
{
    CS_A=0;
    CS_B=0;
    CS_C=0;
    ALE=1;
    delay(1);
    SOC=1;
    delay(1);
    ALE=0;
    delay(1);
    SOC=0;
    delay(1);
    while(EOC==1);
    delay(1);
    OE=1;
    delay(1);
    a=P1;
    delay(1);
    OE=0;
    calculate(a);
    return;
}
void calculate(unsigned int u)
{
    h=2500/255;
    b=u*h;
    V[0]=b/1000+0x30;
    c=b%1000;
    V[2]=c/100+0x30;
    d=e%100;
    V[3]=e/10+0x30;
    f=e%10;
    V[4]=f+0x30;
    DISP_LCD();
    return;
}
void DISP_LCD()
{
    unsigned char z[7]="VOLTAGE";
    lcd_ini();
    lcd_cmd(0x80);
    for(i=0;i<7;i++)
    {
```

```
        lcd_data(z[i]);
      delay(500);
    }
    lcd_cmd(0xC0);
    for(i=0;i<5;i++)
  {
        if(i==1)
        {
            lcd_data('.');
        }
        else
        {
          lcd_data(V[i]);
          delay(500);
        }
    }
  return;
}
void lcd_cmd(unsigned char command)
{
    P0=command;
    RS=0;
    RW=0;
    EN=1;
    delay(10);
    EN=0;
}
void lcd_data(unsigned char disp_data)
{
    P0=disp_data;
    RS=1;
    RW=0;
    EN=1;
    delay(10);
    EN=0;
}
void lcd_ini(void)
{
    lcd_cmd(0x38);
    lcd_cmd(0x0F);
    lcd_cmd(0x01);
}
void delay(unsigned int t)
{
    for(j=0;j<t;j++)
    {
    for(k=0;k<1000;k++)
```

```
            { }
        }
}
```

18.3 Hex Code :

```
:03000000020B11DF
:0C0B1100787FE4F6D8FD75812E020B1DE4
:070B2800564F4C54414745B4
:0B0B1D00120ACC120ACC120A6280F807
:100ACC007590FFC2B1C2B4C2B3D2B27F017E000234
:020ADC000A9E70
:100A6200C2B5C2B6C2B7D2B1120A9AD2B4120A9AA7
:100A7200C2B1120A9AC2B4120A9A20B2FD120A9A9A
:100A8200D2B3120A9AAF907512008F13120A9AC249
:080A9200B3AF13AE1202095DBF
:10095D00752000752109AC20AD211208F68E148F7B
:10096D00157C037DE8120908EF2430F509E43EF506
:10097D0008AE14AF157C037DE81209088C168D178F
:10098D00AE16AF177C007D64120908EF2430F50D0B
:10099D00E43EF50CAE1AAF1B7C007D641209088C89
:1009AD00188D19AE1AAF1B7C007D0A120908EF24B1
:1009BD0030F50FE43EF50EAE1AAF1B7C007D0A122A
:1009CD0009088C1C8D1DE51D2430F511E4351CF531
:0409DD00100209E11A
:1009E10078287C007D007BFF7A0B79287E007F07C9
:1009F1001208D0120B027F80120ADEE4F522F523E1
:100A010074282523F8E6FF120AF07FF47E01120A0A
:100A11009E0523E52370020522C39407E522940075
:100A210040DE7FC0120ADEE4F522F523E5236401EE
:100A3100452270077F2E120AF08014E52325E02459
:100A410008F808E6FF120AF07FF47E01120A9E05FB
:100A510023E52370020522C39405E522940040CCCE
:010A61002272
:100ADE008F80C2A7C2A6D2A57F0A7E00120A9EC22E
:020AEE00A5223F
:100AF0008F80D2A7C2A6D2A57F0A7E00120A9EC20C
:020B0000A5222C
:0F0B02007F38120ADE7F0F120ADE7F01020ADE41
:040A9A007F017E005A
:100A9E00E4F524F525C3E5259FE5249E501FE4F5D6
:100AAE0026F5270527E52770020526B4E8F5E52685
:0E0ABE00B403F00525E52570DC052480D82260
:10080000E709F608DFFA8046E709F208DFFA803EDA
:1008100088828C83E709F0A3DFFA8032E309F608C7
:10082000DFFA8078E309F208DFFA807088828C832F
:10083000E309F0A3DFFA806489828A83E0A3F608E3
```

```
:10084000DFFA805889828A83E0A3F208DFFA804CBD
:1008500080D280FA80C680D4806980F28033801094
:1008600080A680EA809A80A880DA80E280CA8033FD
:1008700089828A83ECFAE493A3C8C582C8CCC58375
:10088000CCF0A3C8C582C8CCC583CCDFE9DEE78045
:100890000D89828A83E493A3F608DFF9ECFAA9F0C4
:1008A000EDFB2289828A83ECFAE0A3C8C582C8CC1A
:1008B000C583CCF0A3C8C582C8CCC583CCDFEADE33
:1008C000E880DB89828A83E493A3F208DFF980CC95
:1008D00088F0EF60010E4E60C388F0ED2402B4048E
:1008E0000050B9F582EB2402B4040050AF23234535
:0608F00082239008507302
:1008F600EF8DF0A4A8F0CF8CF0A428CE8DF0A42E16
:02090600FE22CF
:10090800BC000BBE0029EF8DF084FFADF022E4CCD3
:10091800F875F008EF2FFFEE33FEEC33FCEE9DEC9C
:10092800984005FCEE9DFE0FD5F0E9E4CEFD22EDE2
:10093800F8F5F0EE8420D21CFEADF075F008EF2F2C
:10094800FFED33FD4007985006D5F0F222C398FD1D
:050958000FD5F0EA22BA
:00000001FF
```

18.5 Conclusion :

We measured the level of DC voltage and displayed it on LCD. Through potentiometer arrangement, we varied the level of DC voltage. Then we gave this variable DC voltage to ADC and manipulated the output of ADC to display.

19. Display 0 to 9 Using 7 segment display

***19.1 Aim* :**

Write a program to display 0 to 9 Using 7 segment display.

19.2 Code :

```
#include <reg51.h>
sbit x=P2^7;
sbit y=P2^6;
sbit z=P2^5;
void delay(void);
unsigned int i,j,seg=0;
void main()
{
    unsigned char
disp[]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
    while(1)
    {
        k:
        if(seg==0)
        {
            x=0;
            y=0;
            z=0;
            seg++;

        }
        else if (seg==1)
        {
            x=0;
            y=0;
            z=1;
            seg++;

        }
       else if (seg==2)
        {
            x=0;
            y=1;
            z=0;
            seg++;

        }
      else if(seg==3)
        {
            x=0;
```

```
                y=1;
                z=1;
                seg=0;
            }
          for(i=0;i<10;i++)
            {
                P0=disp[i];
                for(j=0;j<1000;j++)
                {delay();}
            }

    }
}
void delay(void)
{
    TMOD=0x01;
    TL0=0xFB;
    TH0=0xF3;
    TR0=1;
    while(TF0==0);
    TF0=0;
    TR0=0;
}
```

19.3 Hex Code :

```
:0300000002098F63
:0C098F00787FE4F6D8FD7581170209D6C8
:0A0A2E003F065B4F666D7D077F6F8A
:040A380002120000A6
:1008F60078087C007D007BFF7A0A792E7E007F0ACD
:100906001208D0E51345127006C2A7C2A6C2A5E515
:1009160013640145127006C2A7C2A6D2A5E51364E8
:100926000245127006C2A7D2A6C2A5E51364034506
:10093600127006C2A7D2A6D2A5E5136404451270AA
:1009460006F512F51380BCE4F514F51574082515A3
:10095600F8E6F580E4F516F517120A1B0517E517F4
:1009660070020516C394E8E516940340EC0515E5F8
:1009760015700205 14C3940AE514940040CE0513BD
:09098600E51370020512020909D3
:100A1B00758901758AFB758CF3D28C308DFDC28D77
:030A2B00C28C2258
:10099B000208F6E493A3F8E493A34003F68001F274
:1009AB0008DFF48029E493A3F85407240CC8C3335D
:1009BB00C4540F4420C8834004F456800146F6DF2C
:1009CB00E4800B01020408102040809 00A38E47E7A
:1009DB00019360BCA3FF543F30E509541FFEE49321
:1009EB00A360010ECF54C025E060A840B8E493A3E8
```

```
:1009FB00FAE493A3F8E493A3C8C582C8CAC583CA13
:100A0B00F0A3C8C582C8CAC583CADFE9DEE780BECA
:010A3C0000B9
:10080000E709F608DFFA8046E709F208DFFA803EDA
:1008100088828C83E709F0A3DFFA8032E309F608C7
:10082000DFFA8078E309F208DFFA807088828C832F
:10083000E309F0A3DFFA806489828A83E0A3F608E3
:10084000DFFA805889828A83E0A3F208DFFA804CBD
:1008500080D280FA80C680D4806980F28033801094
:1008600080A680EA809A80A880DA80E280CA8033FD
:1008700089828A83ECFAE493A3C8C582C8CCC58375
:10088000CCF0A3C8C582C8CCC583CCDFE9DEE78045
:100890000D89828A83E493A3F608DFF9ECFAA9F0C4
:1008A000EDFB2289828A83ECFAE0A3C8C582C8CC1A
:1008B000C583CCF0A3C8C582C8CCC583CCDFEADE33
:1008C000E880DB89828A83E493A3F208DFF980CC95
:1008D00088F0EF60010E4E60C388F0ED2402B4048E
:1008E0000050B9F582EB2402B4040050AF23234535
:0608F00082239008507302
:00000001FF
```

19.4 Conclusion :

We used four 7 segment LED displays to continuously display series of numbers from 0 to 9. We kept shifting this series from first segment to last segment.

20. Real time digital clock

20.1 Aim :

Write a program for real time digital clock using segment digital display

20.2 Code :

```
#include <reg51.h>
unsigned int i,j,sec1,sec2,min1,min2,seg=0;
unsigned char disp=0;
sbit x=P2^7;
sbit y=P2^6;
sbit z=P2^5;
void delay(void);
unsigned char
num[]={0x3F,0x06,0x5B,0x4F,0x66,0X6D,0x7D,0x07,0x7F,0x6F};
void main()
{while(1)
    {
        min2=min1=sec2=sec1=0;
          for(min2=0;min2<6;min2++)
          {
            for(min1=0;min1<10;min1++)
            {
              for(sec2=0;sec2<6;sec2++)
              {
                for(sec1=0;sec1<10;sec1++)
                {
                                    for(i=0;i<1000;i++)
                          {delay();}
                                    }
              }
            }
          }
    }
}
void delay(void)
{
 TMOD=0X01;
 TH0=0XF3;
 TL0=0XFB;
 TR0=1;
 while(TF0==0)
 TR0=0;
 TF0=0;
 disp++;
 disp=disp%4;
```

```
 switch(disp)
 {
  case 0:
  P0=num[sec1];
    x=0;
    y=1;
    z=1;
  break;
  case 1:
  P0= num[sec2];
    x=0;
  y=1;
    z=0;
  break;
  case 2:
  P0= num[min1];
    x=0;
    y=0;
    z=1;
  break;
  case 3:
  P0=num[min2];
    x=0;
    y=0;
    z=0;
  break;
 }
}
```

20.3 Hex Code :

```
:03000000020800F3
:0C080000787FE4F6D8FD758120020847DF
:1009740002090000108000A0B3F065B4F666D7D0B
:03098400077F6F7B
:10088C00E4F519F51AF51BF51CF51DF51EF51FF50C
:10089C0020F51FF520E4F51DF51EE4F51BF51CE411
:1008AC00F519F51AE4F515F5161209100516E516E5
:1008BC0070020515C394E8E515940340EC051AE5A0
:1008CC001A70020519C3940AE519940040D6051C48
:1008DC00E51C7002051BC39406E51B940040C00583
:1008EC001EE51E7002051DC3940AE51D940040AA66
:1008FC000520E5207002051FC39406E51F940040F7
:04090C009402088CBD
:10091000758901758CF3758AFBD28C208D04C28C8D
:1009200080F9C28D0508530803E5081460191460A6
:10093000251460312403703B740B251AF8E6F5800A
```

```
:10094000C2A7D2A6D2A522740B251CF8E6F580C258
:10095000A7D2A6C2A522740B251EF8E6F580C2A771
:10096000C2A6D2A522740B2520F8E6F580C2A7C244
:04097000A6C2A52254
:10080C0002088CE493A3F8E493A34003F68001F26E
:10081C0008DFF48029E493A3F85407240CC8C333ED
:10082C00C4540F4420C8834004F456800146F6DFBC
:10083C00E4800B0102040810204080900974E47ECF
:10084C00019360BCA3FF543F30E509541FFEE493B1
:10085C00A360010ECF54C025E060A840B8E493A378
:10086C00FAE493A3F8E493A3C8C582C8CAC583CAA3
:10087C00F0A3C8C582C8CAC583CADFE9DEE780BE5B
:01098700006F
:00000001FF
```

20.4 Conclusion :

We made digital clock by using four 7 segment LED displays. We displayed Seconds on the last 2 displays and Minutes on the first 2 displays.

21. Generating Waveform Using DAC

21.1 Aim :

Write a program to monitor the state of switch connected to port 2 and generate following waveform Using DAC**.**

When P2.1=1 generate ramp waveform
When P2.2=1 generate triangle waveform
When P2.3=1 generate sinusoidal waveform

21.2 Code :

```
#include <reg51.h>
sbit DAC_WR=P3^0;
unsigned int i;
sbit a=P2^1;
sbit b=P2^2;
sbit c=P2^3;
void main()
{
    unsigned char
DAC1_value[]={0,30,60,90,120,150,180,210,240,255};
```

```
    unsigned char
DAC2_value[]={0,30,60,90,120,150,180,210,240,255,240,210,180,150
,120,90,60,30,0};
    unsigned char
DAC3_value[]={0,30,60,90,120,150,180,210,240,255};
    P1=0xFF;
    if(a==0)
    {
     while(1)
     {
        DAC_WR=1;
        for(i=0;i<10;i++)
        {P1=DAC1_value[i];}
     }
  }
    if(b==0)
    {
     while(1)
     {
        DAC_WR=1;
        for(i=0;i<19;i++)
        {P1=DAC2_value[i];}
     }
  }
    if(c==0)
    {
     while(1)
     {
        DAC_WR=1;
        for(i=0;i<19;i++)
        {P1=DAC3_value[i];}
     }
  }
}
```

21.3 Hex Code :

```
:030000000209C032
:0C09C000787FE4F6D8FD7581300208F65F
:10099900001E3C5A7896B4D2F0FF001E3C5A789655
:1009A900B4D2F0FFF0D2B496785A3C1E00001E3C37
:0709B9005A7896B4D2F0FF5A
:1008F60078087C007D007BFF7A0979997E007F0A63
:100906001208D078127C007D007BFF7A0979A37EDD
:10091600007F131208D078257C007D007BFF7A09C2
:1009260079B67E007F0A1208D07590FF20A11FD2EB
:10093600B0E4F52FF53074082530F8E6F59005306B
:10094600E5307002052F640A452F70EA80E120A287
```

```
:100956001FD2B0E4F52FF53074122530F8E6F59085
:100966000530E5307002052F6413452F70EA80E1EB
:1009760020A31FD2B0E4F52FF53074252530F8E614
:10098600F5900530E5307002052F6413452F70EAA7
:0209960080E1FE
:01099800223C
:10080000E709F608DFFA8046E709F208DFFA803EDA
:1008100088828C83E709F0A3DFFA8032E309F608C7
:10082000DFFA8078E309F208DFFA807088828C832F
:10083000E309F0A3DFFA806489828A83E0A3F608E3
:10084000DFFA805889828A83E0A3F208DFFA804CBD
:1008500080D280FA80C680D4806980F28033801094
:1008600080A680EA809A80A880DA80E280CA8033FD
:1008700089828A83ECFAE493A3C8C582C8CCC58375
:10088000CCF0A3C8C582C8CCC583CCDFE9DEE78045
:100890000D89828A83E493A3F608DFF9ECFAA9F0C4
:1008A000EDFB2289828A83ECFAE0A3C8C582C8CC1A
:1008B000C583CCF0A3C8C582C8CCC583CCDFEADE33
:1008C000E880DB89828A83E493A3F208DFF980CC95
:1008D00088F0EF60010E4E60C388F0ED2402B4048E
:1008E0000050B9F582EB2402B4040050AF23234535
:0608F00082239008507302
:00000001FF
```

21.5 Conclusion :

We generated Ramp Wave, Triangular Wave and Sinusoidal Wave using DAC for different conditions of Switch.

22. Measure unknown signal frequency

22.1 Aim :

Write a program to measure unknown signal frequency Using timer 2 in capture mode.

22.2 Code :

```
#include "p89v51rd2.h"
sbit rs=P2^7;
sbit rw=P2^6;
sbit e=P2^5;
sbit busy=P0^7;
void delay(unsigned int time);
void lcd_init(void);
void lcd_cmd(unsigned char command);
void lcd_data(unsigned char display_data);
void display(unsigned int);
unsigned int r=0, q=0, i=0;
unsigned int b;
unsigned char a[4]=0;
    unsigned int  xdata count1=0, xdata count2=0;
    unsigned int  freq=0;
int action_flag=0;
void ready(void);
void main()
{
    EA=1;
    ET2=1;
    T2MOD=0X00;
    T2EX=1;
     RCLK=0;
    TCLK=0;
    EXEN2=1;
    T2=0;
    CP_RL2=1;
    RCAP2H=0X00;
    RCAP2L=0X00;
    TH2=0X00;
    TL2=0X00;
    TR2=1;
    lcd_init();
            lcd_cmd(0x80);
    while(1)
    {
        if(action_flag==1)
        {
            action_flag=0;
```

```
            TR2=0;
            count1=a[0];
            count1=count1<<8;
            count1=count1+a[1];
            count2=a[2];
            count2=count2<<8;
            count2=count2+a[3];
            if(r==0)
            {
            freq=count2-count1;
            }
            else
            {
                freq=(r*(65536))-count1+(count2);
            }
            freq= 3076923/freq;

            display(freq);
        }
  }

}

void timer2_isr(void) interrupt 5
    {
        if(TF2==1 && action_flag==0 )
        {
            TF2=0;
            r++;
        }
        if(EXF2==1)
        {
            EXF2=0;
            a[i]=RCAP2H;
            i++;
            a[i]=RCAP2L;
            i++;
            if(i>3)
            {
                i=0;
                TR2=0;
                action_flag=1;
            }
        }
    }
    void lcd_cmd(unsigned char command)
{
```

```
    //ready();
 P0= command;
    rs=0;
    rw=0;
    e=1;
    delay(1);
    e=0;
    return;
}
void lcd_data(unsigned char display_data)
{
    //ready();
    P0 = display_data;
    rs=1;
    rw=0;
    e=1;
    delay(1);
    e=0;
return;
}
void lcd_init(void)
{
     lcd_cmd(0x38);
    delay(10);
    lcd_cmd(0x0F);
    delay(10);
//  lcd_cmd(0x01);
//   delay(10);
//    lcd_cmd(0x81);
//  //  delay(10);
//  lcd_cmd(0x3C);
//   delay(10);
//   lcd_cmd(0x0E);
    delay(10);
    lcd_cmd(0x01);
 delay(10);
//  lcd_cmd(0x06);
// delay(10);
    return;
}
void delay(unsigned int time)
{
    unsigned int k,j ;
    for(k=0;k<time;k++)
{
    for(j=0;j<1000;j++)
    {
```

```
    }
}
}
void display(unsigned int f)
{
//      b=f+'o';
//  lcd_data(b);

    b=f/100000+'0';
    lcd_data(b);
    b=(f/10000)%10+'0';
    lcd_data(b);
    b=(f/1000)%10+'0';
    lcd_data(b);
    b=(f/100)%10+'0';
    lcd_data(b);
    b=(f/10)%10+'0';
    lcd_data(b);
    b=f%10+'0';
    lcd_data(b);
    return;
}
```

22.3 Hex Code :

```
:03000000020A4AA7
:0C0A4A00787FE4F6D8FD758119020A914E
:100C41000216000002140000021200000040C000051
:100C5100000042000000004200020000020A000001
:040C61000208000085
:10097200D2AFD2ADE4F5C9D291C2CDC2CCD2CBC2F4
:1009820090D2C8F5CBF5CAF5CDF5CCD2CA120BC9B7
:100992007F80120C66E5096401450870F8F508F5D8
:1009A20009C2CA900000F0A3E50CF0E0F8E4F0E818
:1009B200900000F0A3E0250DF0900000E03400F07C
:1009C200900002E4F0A3E50EF0E0F8E4F0E8900015
:1009D20002F0A3E0250FF0900002E03400F0E517EA
:1009E20045167019900000E0FEA3E0FFC3900003DB
:1009F200E09FF50B900002E09EF50A802CAE16AF48
:100A020017E4FCFDFBFA7901F8120855900000E0AA
:100A1200FCA3E0FDC3EF9DFFEE9CFE900003E02FE0
:100A2200F50B900002E03EF50AAE0AAF0BAB07AA47
:100A320006E4F9F87F3B7EF37D2EFC120B858E0ACD
:080A42008F0B120B3102099722
:03002B00020AD6F0
:100AD600C0E0C0D075D000C00030CF10E509450891
:100AE600700AC2CF0517E5177002051630CE35C25B
```

:100AF600CE740C2513F8A6CB0513E5137002051268
:100B0600240CF8A6CA0513E51370020512D3940344
:100B1600E5129400400E751200751300C2CA7508DE
:0B0B260000750901D000D0D0D0E032F3
:0A0C66008F80C2A7120BF4C2A52272
:100C1C00AE18AF197C001208007C007D0A12080087
:0B0C2C00ED2430F511E43CF510AF1191
:0A0C37008F80D2A7120BF4C2A52291
:100BC9007F38120C667F0A7E00120BFC7F0F120C15
:100BD900667F0A7E00120BFC7F0A7E00120BFC7FE7
:0B0BE90001120C667F0A7E00020BFC6C
:080BF400C2A6D2A57F017E001C
:100BFC00E4FDFCC3ED9FEC9E5015E4FBFA0BBB002F
:0F0C0C00010ABA03F8BBE8F50DBD00010C80E446
:010C1B0022B6
:100B31008E188F19E4FCFD7BA07A867901F8120BDF
:100B410085EF2430F511E43E120C33AE18AF197C59
:100B5100277D10120C22AE18AF197C037DE8120C10
:100B6100227D64120C1C7D0A120C1CAE18AF197C7C
:100B7100007D0A120800ED2430F511E43CF510AFB8
:040B810011020C371A
:100A5600020972E493A3F8E493A34003F68001F23B
:100A660008DFF48029E493A3F85407240CC8C333A1
:100A7600C4540F4420C8834004F456800146F6DF70
:100A8600E4800B01020408102040800900C41E47EB3
:100A9600019360BCA3FF543F30E509541FFEE49365
:100AA600A360010ECF54C025E060A840B8E493A32C
:100AB600FAE493A3F8E493A3C8C582C8CAC583CA57
:100AC600F0A3C8C582C8CAC583CADFE9DEE780BE0F
:010C6500008E
:10080000BC000BBE0029EF8DF084FFADF022E4CCDC
:10081000F875F008EF2FFFEE33FEEC33FCEE9DECA5
:10082000984005FCEE9DFE0FD5F0E9E4CEFD22EDEB
:10083000F8F5F0EE8420D21CFEADF075F008EF2F35
:10084000FFED33FD4007985006D5F0F222C398FD26
:050850000FD5F0EA22C3
:10085500E88FF0A4CC8BF0A42CFCE98EF0A42CFC42
:100865008AF0EDA42CFCEA8EF0A4CDA8F08BF0A4C0
:100875002DCC3825F0FDE98FF0A42CCD35F0FCEB1F
:100885008EF0A4FEA9F0EB8FF0A4CFC5F02ECD39E4
:0F089500FEE43CFCEAA42DCE35F0FDE43CFC2251
:100B8500C2D5E830E70FB2D5E4C39BFBE49AFAE49B
:100B950099F9E498F8EC30E717B2D5120BBB1208B7
:100BA500E0E4C39BFBE49AFAE499F9E498F880033E
:100BB5001208E030D50DE4C39FFFE49EFEE49DFDE1
:040BC500E49CFC228E
:1008A40075F008758200EF2FFFEE33FECD33CDCC0B
:1008B40033CCC58233C5829BED9AEC99E58298408E

```
:1008C4000CF582EE9BFEED9AFDEC99FC0FD5F0D66B
:1008D400E4CEFBE4CDFAE4CCF9A88222B800C1B995
:1008E4000059BA002DEC8BF084CFCECDFCE5F0CBD3
:1008F400F97818EF2FFFEE33FEED33FDEC33FCEB0C
:1009040033FB10D703994004EB99FB0FD8E5E4F9C6
:10091400FA227818EF2FFFEE33FEED33FDEC33FCB3
:10092400C933C910D7059BE99A4007EC9BFCE99AA7
:10093400F90FD8E0E4C9FAE4CCFB2275F010EF2FEC
:10094400FFEE33FEED33FDCC33CCC833C810D707EC
:100954009BEC9AE899400AED9BFDEC9AFCE899F827
:0E0964000FD5F0DAE4CDFBE4CCFAE4C8F922BA
:00000001FF
```

22.6 Conclusion :

We generated PWM of 1 kHz frequency 20 percentage duty cycle using Timer T2 in 16 bit Auto Reload Mode in Interrupt Mode.

23. Timer 2 based PWM generation

23.1 Aim :

Write a program to generate clock of 1 KHZ using timer T2. (Timer 2 in clock out mode).

23.2 Code :

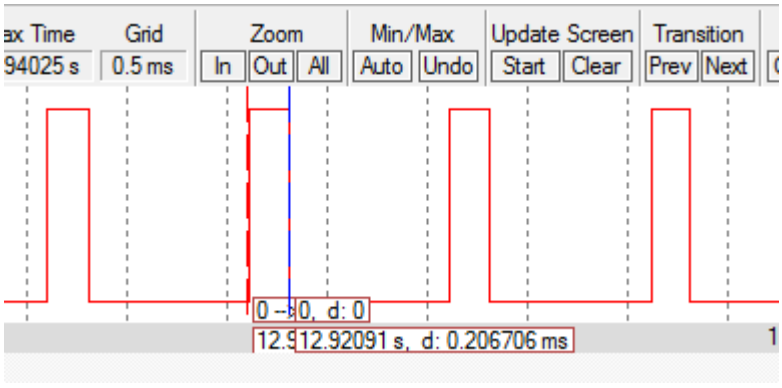
```
#include "p89v51rd2.h"
unsigned int x;
void main()
{
    EA=1;
    ET2=1;
    T2MOD=0X00;
    T2CON=0X00;
  TH2=0XFE;
  TL2=0XCC;
    RCAP2H=0XFE;
    RCAP2L=0XCC;
  TR2=1;
    while(1)
    {
        if(x==2)
        {
```

```
                P1=0x00;
            }
            if(x==10)
            {
                P1=0xFF;
                x=0;
      }
      }
}
void t2_isr(void)interrupt 5
{
 TF2=0;
 x++;
}
```

23.3 Hex Code :

:03000000020841B2
:0C084100787FE4F6D8FD758109020800FC
:10080000D2AFD2ADE4F5C9F5C875CDFE75CCCC75C7
:10081000CBFE75CACCD2CAE509640245087002F560
:1008200090E509640A450870EE7590FFF508F50932
:0208300080E561
:03002B0002083296
:0F083200C0E0C2CF0509E50970020508D0E03229
:00000001FF

23.4 Simulated Output :



23.5 Conclusion :

Generated Clock of 1 kHz frequency using Timer T2 in Clock out mode.

24. generate clock of 1kHz Using timer 2 in clock out mode

24.1 Aim :

Write a program to generate clock of 1kHz Using timer 2 in clock out mode.

24.2 Code :

```
#include "p89v51rd2.h"
unsigned int x;
void main()
{
    T2MOD=0X02;
    T2CON=0X00;
  TH2=0XDB;
  TL2=0XFF;
    RCAP2H=0XDB;
    RCAP2L=0XFF;
  TR2=1;
    while(1);
}
```

24.3 Hex Code :

```
:03000000020816DD
:0C081600787FE4F6D8FD75810902080027
:1008000075C902E4F5C875CDDB75CCFF75CBDB751A
:06081000CAFFD2CA80FEFF
:00000001FF
```

24.4 Conclusion :

We measured the frequency of unknown signal using timer T2 in capture mode. We captured the two counts for one-time period of the waveform and calculated the frequency and displayed it on LCD.

25 UART communication

25.1 Aim :

Write a program for UART communication.

25.2 Code :

```
#include <reg51.h>
int i;
unsigned char A[5]="BHAGALPUR";

void main()
{
        TMOD=0X20;
        TH1=0XFD;
        TCON=0X50;
        TR1=1;
        while(1)
        {
                for (i=0; i<=4; i++)
                {
                        SBUF=A[i];
                        while(TI==0);
                        TI=0;
                }
        }
}
```

25.3 Hex Code

```
:03000000020800F3
:0C080000787FE4F6D8FD75810E020847F1
:0708B900050853555241549C
:10088C00758920758DFD758850D28EE4F50DF50EA9
:10089C007408250EF8E6F5993099FDC299050EE518
:0D08AC000E7002050D6406450D70E580DE3E
:10080C0002088CE493A3F8E493A34003F68001F26E
:10081C0008DFF48029E493A3F85407240CC8C333ED
:10082C00C4540F4420C8834004F456800146F6DFBC
:10083C00E4800B0102040810204080900 8B9E47E8B
:10084C00019360BCA3FF543F30E509541FFEE493B1
:10085C00A360010ECF54C025E060A840B8E493A378
:10086C00FAE493A3F8E493A3C8C582C8CAC583CAA3
:10087C00F0A3C8C582C8CAC583CADFE9DEE780BE5B
:0108C0000037
:00000001FF
```

25.5 Conclusion **:**

We did UART Serial Communication by loading data into SBUF. We used the baud rate of 9600 Hz and sent the data ‘V’ at this baud rate.

26 Generate a PWM of following duty cycle suing PCA timer

26.1 Aim :

Write a program to generate a PWM of following duty cycle suing PCA timer.
When P2.1=1, Generate a PWM of duty cycle of 40%.
When P2.1=1, Generate a PWM of duty cycle of 40%.

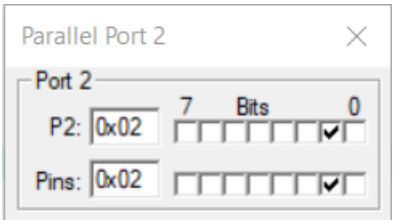
26.2 Code :

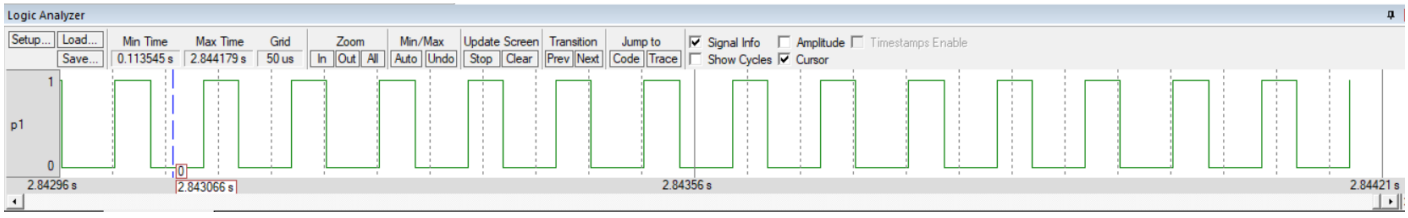
```
#include "header.h"
sbit sw=P2^1;
void main()
{
    P2=0x00;
    CMOD=0x02;
    CCAPM0=0x42;
    CR=1;


    while(1)
    {
        if(sw==0)
        {
            CCAP0L=0X40;
            CCAP0H=0X40;
        }
        if(sw==1)
        {
            CCAP0L=0X99;
            CCAP0H=0X99;
        }
    }
}
```

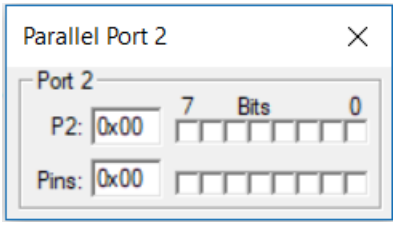
26.3 Hex Code :

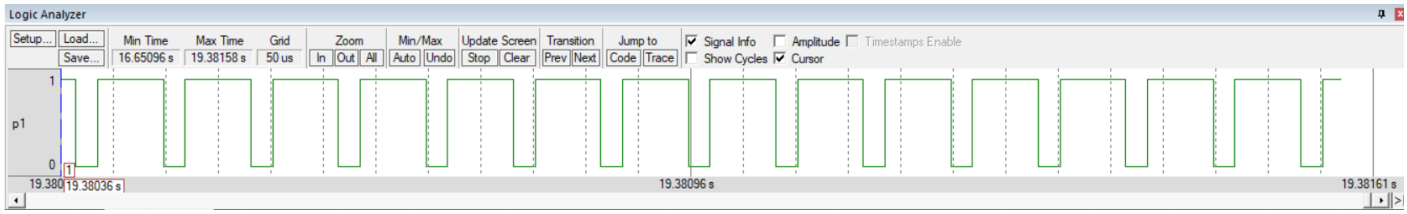
:10080000E4F5A0F5D975DA42D2DE20A10675EA40FA
:0E08100075FA4030A1F475EA9975FA9980ECFA
:0300000002081ED5
:0C081E00787FE4F6D8FD75810702080021
:00000001FF

26.4 Simulated Output :









26.6 Conclusion :

We generated PWM of different duty cycle using PCA timer for different states of switch. We programmed module 0 of PCA timer in 8 bit PWM mode.