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

5. Generate Square Wave

5.1: Aim:
5.2: Code:
5.3 :Hex Code:
5.4: Simulated Output:
5.5: Conclusion:

6. Hardware Delay Function

6.1: Aim:
6.2: Code:
6.3: Hex Code:
6.4 :Simulated Output:
6.5: Conclusion:

7. Software Delay Function

7.1: Aim:
7.2: Code:
7.3: Hex Code:
7.4: Simulated Output:
7.5 :Conclusion:

8. Auto reload mode

8.1: Aim:
8.2: Code:
8.3: Hex Code:
8.4 :Simulated Output:
8.5 :Conclusion:

9. Interrupt using timer

9.1: Aim:
9.2: Code:
9.3 :Hex Code:
9.4: Simulated Output:
9.5: Conclusion:

10. Pattern Using PYTHON

10.1: Aim:
10.2 :Code:
10.3:Hex Code:
10.4:Simulated Output:
10.5: Conclusion:

1.Blinking All LED in Port P1

1.1 : AIM

Blinking of LED in Port P1 with some delay.

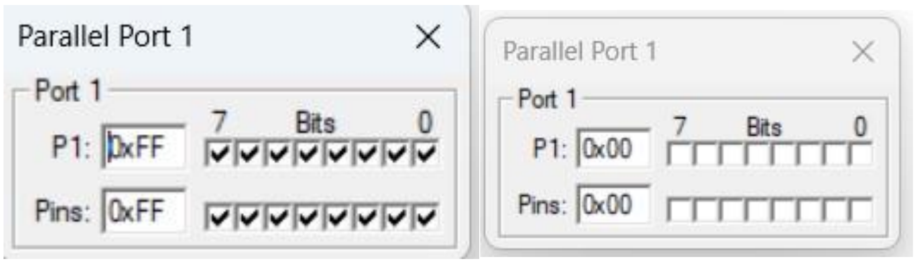
1.2 : CODE

```
ORG00 START:
MOV P1, #0XFF
CALL AGAIN
MOV A, P1
        CPL A
MOV P1, A
CALL AGAIN
        SJMP START  AGAIN:MOV
R2, #10
AGAIN1:MOV R3, #200
AGAIN2:MOV R4, #200
        DJNZ R4,$
        DJNZ R3, AGAIN2
        DJNZ R2, AGAIN1
RET
END
```

1.3 : HEX CODE

```
:100000007590FF110EE590F4F590110E80F27A0ACA
:0B0010007BC87CC8DCFEDBFADAF622BD
:00000001FF
```

1.4 : SIMULATED OUTPUT


Parallel Port 1
Port 1
P1: 0xFF 7 Bits 0 ✓✓✓✓✓✓✓✓
Pins: 0xFF ✓✓✓✓✓✓✓✓

Parallel Port 1
Port 1
P1: 0x00 7 Bits 0 ☐☐☐☐☐☐☐☐
Pins: 0x00 ☐☐☐☐☐☐☐☐

1.5 : CONCLUSION.

We have done that All the LEDs connected to port P1 is blinking with some delay.

2. Sequence Generation using LED

2.1: AIM

Sequence generation using LED, moving from right to left.

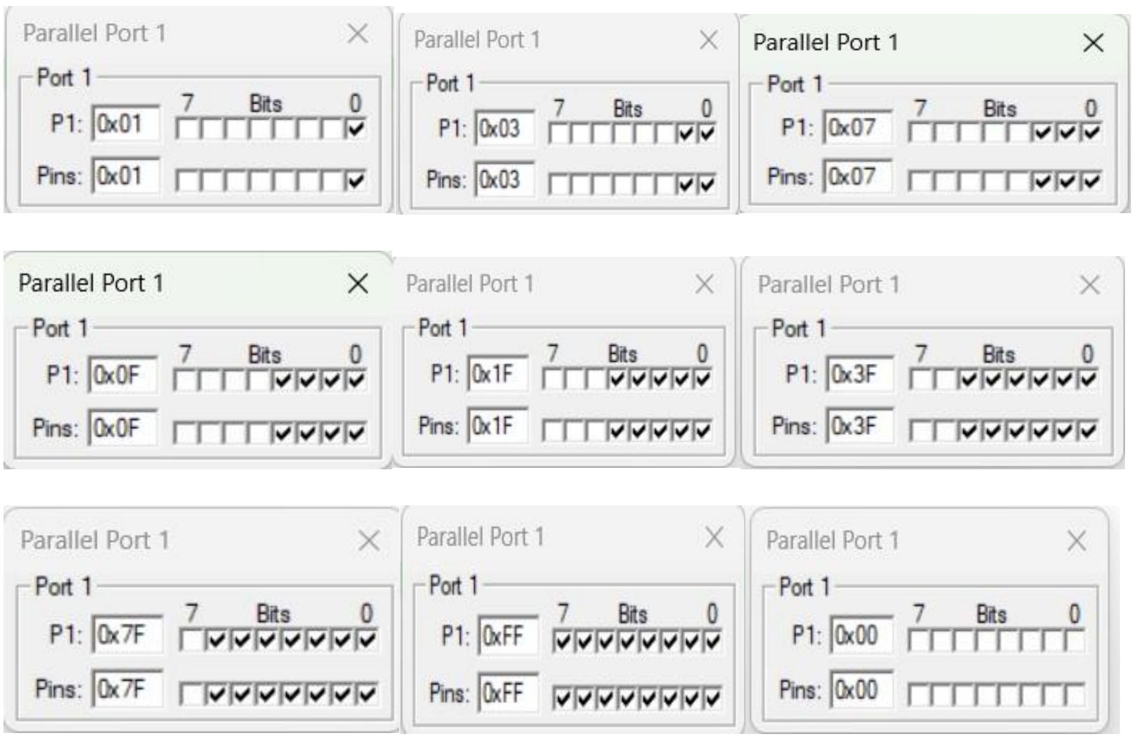
2.2: CODE

```
#include<reg51.h> #include<stdio.h>
#include<math.h>
int a; int
b; inti;
int j;
void main(){ while(1){
      P1 =0x01; for(i=0;i<7;i++){
         a = P1;          b
= P1<<1;          P1
=a|b;
for(j =0;j<30000;j++){
if(P1 ==0xFF)
               P1 =0x00;
}
}
}
      }
```

2.3 :HEX CODE

```
:0300000002084DA6
:0C084D00787FE4F6D8FD75810F020800EA
:10080000759001E4F50CF50DAF907508008F09AFF8
:1008100090EF25E0F50BE433F50AE509450BF5907B
:10082000E4F50EF50FAF90EFF47002F590050FE5CB
:100830000F7002050EB430EDE50EB475E8050DE558
:0D0840000D7002050C6407450C70BD80B3FF
:00000001FF
```

2.4 :SIMULATED OUTPUT:



```
Parallel Port 1          Parallel Port 1          Parallel Port 1
Port 1                   Port 1                   Port 1
P1:   0x01               P1:   0x03               P1:   0x07
Pins: 0x01               Pins: 0x03               Pins: 0x07

Parallel Port 1          Parallel Port 1          Parallel Port 1
Port 1                   Port 1                   Port 1
P1:   0x0F               P1:   0x1F               P1:   0x3F
Pins: 0x0F               Pins: 0x1F               Pins: 0x3F

Parallel Port 1          Parallel Port 1          Parallel Port 1
Port 1                   Port 1                   Port 1
P1:   0x7F               P1:   0xFF               P1:   0x00
Pins: 0x7F               Pins: 0xFF               Pins: 0x00
```

2.5: CONCLUSION

Sequence generated using some delay with LED, which shift from right to left.

3.Toggling of LED Connected to Port P1

3.1: AIM

Toggle LED of port 1 by using software delay.

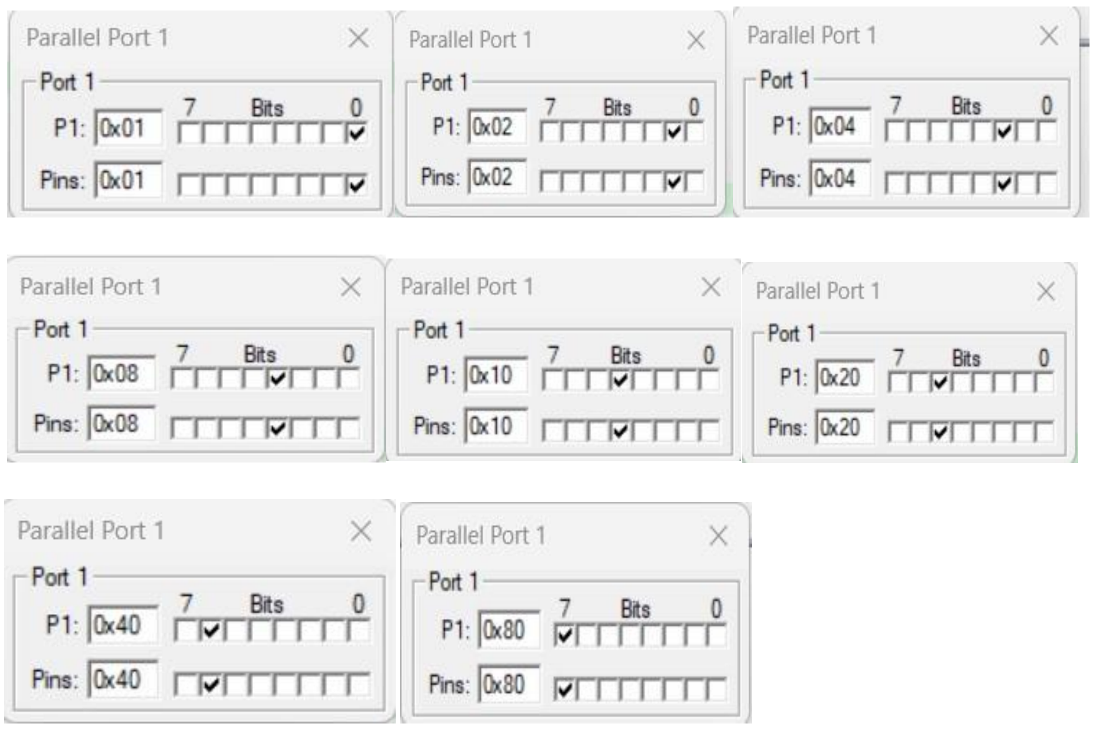
3.2 :CODE

```
#include<reg51.h> int b;
int c; void
main(){
while(1){
       P1 =0x01;
for(b=0;b<8;b++){ for(c =0;
c<25000; c ++);
              P1=P1<<1;
}
}
}
```

3.3: HEX CODE

```
:03000000020833C0
:0C083300787FE4F6D8FD75810B02080008
:10080000759001E4F508F509E4F50AF50B050BE52B
:100810000B7002050AB4A8F5E50AB461F0E590256D
:10082000E0F5900509E509700205086408450870BF
:03083000D780CDA1
:00000001FF
```

3.4 :SIMULATED OUTPUT



```
Parallel Port 1
Port 1
        7    Bits    0
P1:   0x01  ☐☐☐☐☐☐☐✓
Pins: 0x01  ☐☐☐☐☐☐☐✓

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

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

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

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

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

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

Parallel Port 1
Port 1
        7    Bits    0
P1:   0x80  ✓☐☐☐☐☐☐☐
Pins: 0x80  ✓☐☐☐☐☐☐☐
```

:

3.5: CONCLUSION

We have implemented the toggling of LEDof port P1 using software delay from right to left.

4. UART Communication

4.1: AIM

Write a program to generate using UART communication.

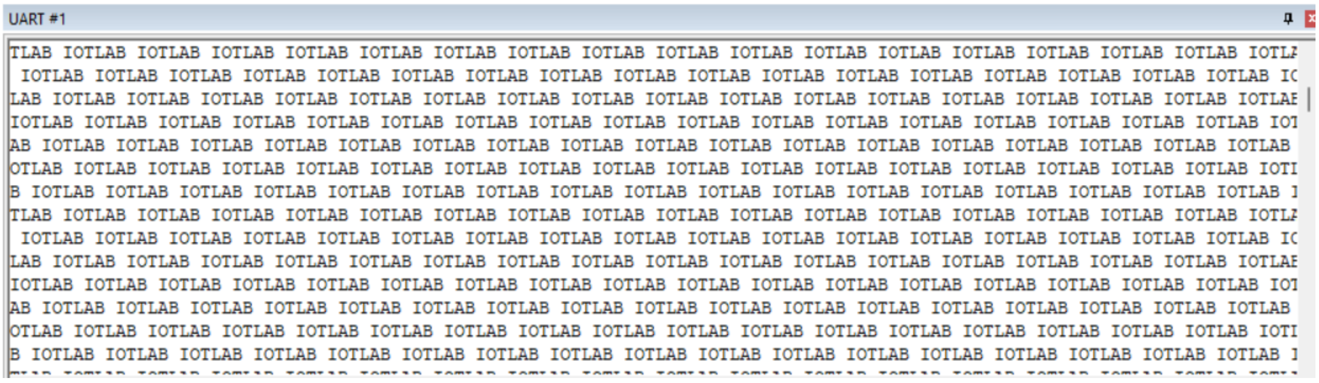
4.2 : CODE

```
ORG 0000H;

MOV TMOD,#20H;
MOV SCON,#50H;
MOV TH1,#0FDH;

SETB TR1;
REPEAT:
MOV SBUF,#'I';
      RZ11:JNB TI,RZ1;
      CLR TI;
MOV SBUF,#'0';
      RZ2:JNB TI,RZ2;
      CLR TI;
MOV SBUF,#'T';
      RZ3:JNB TI,RZ3;
      CLR TI;
MOV SBUF,#'L';
      RZ4:JNB TI,RZ4;
      CLR TI;
MOV SBUF,#'A';
      RZ5:JNB TI,RZ5;
      CLR TI;
MOV SBUF,#'B';
      RZ6:JNB TI,RZ6;
      CLR TI;
MOV SBUF,#' ';
      RZ7:JNB TI,RZ7;
      CLR TI;

      SJMP REPEAT;
END
```

4.3 :HEX CODE

```
:03000000020800F3
:0C080000787FE4F6D8FD75810702000047
:10000000758920759850758DFDD28E7599493099F6
:10001000FDC29975994F3099FDC2997599543099DF
:10002000FDC29975994C3099FDC2997599413099E5
:10003000FDC2997599423099FDC299759920309900
:05004000FDC29980C61D
:00000001FF
```

4.4: SIMULATED OUTPUT


UART #1

```
TLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLA
 IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IO
LAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB
IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOT
AB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB
OTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTL
B IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB I
TLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLA
 IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IO
LAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB
IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOT
AB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB
OTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTL
B IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB IOTLAB I
```

4.5: CONCLUSION

We transfer UART Serial Communication by loading data into SBUF with the baud rate of 9600 Hz and sent the my name 'RAJAT ' at this baud rate.

5.Square wave Generation

5.1: AIM

Square wave generation from given data: i/p frequency=1khz, xtal frequency=11.0592Mhz,port P1.7.

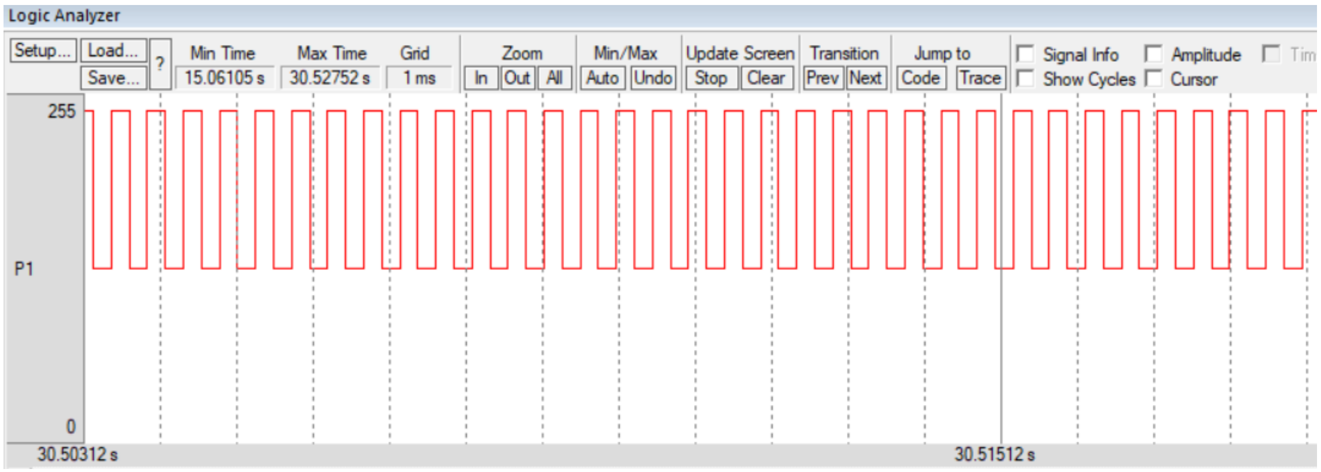
5.2: CODE

```
ORG 0000H MOV
TMOD,#01H;
LOOP:
MOV TH0,#0FEH;
MOV TL0,#33H;
SETB TR0;
   CHECK:JNB TF0,CHECK;
   CPL P1.7;
   CLR TR0;
   CLR TF0;
   SJMP LOOP;
END
```

5.3: HEX CODE:

```
:10000000758901758CFE758A33D28C308DFDB2975F
:06001000C28CC28D80EDE0
:00000001FF
```

5.4: SIMULATED OUTPUT:



5.5 : CONCLUSION:

We have generated the square wave of the given data.

---

6. Hardware Delay Function.

6.1: AIM

Creation of Hardware delay using Timer Mode 1.

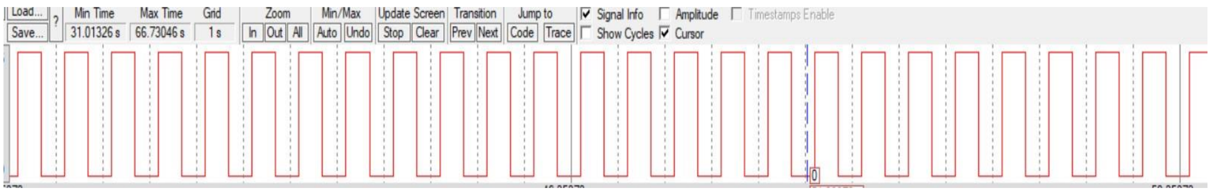
6.2 : CODE

```
#include <reg51.h>

void delay(){
TH0 =0xD8;
  TL0 =0xFF;
TR0 =1;
while(TF0 ==0);
TF0 =0;
  TR0 =0;
} inti;
void main(){
  TMOD =0x01;
while(1){     P1
=~P1;

for(i=0;i<100;i++){
delay();
} }
}
```

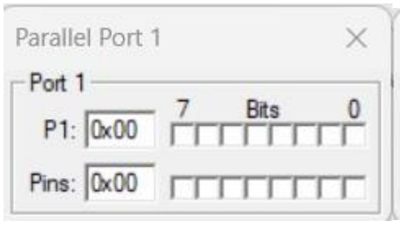
6.3: HEX CODE

```
:03000000020833C0
:0C083300787FE4F6D8FD7581090208000A
:10082300758CD8758AFFD28C308DFDC28DC28C2217
:100800007589016390FFE4F508F5091208230509CD
:10081000E50970020508C39410E508648094A740B8
:03082000EA80E08B :00000001FF
```

6.4 : SIMULATED OUTPUT



Min Time 31.01326 s
Max Time 66.73046 s
Grid 1 s
☑ Signal Info ☐ Amplitude ☐ Timestamps Enable
☐ Show Cycles ☑ Cursor

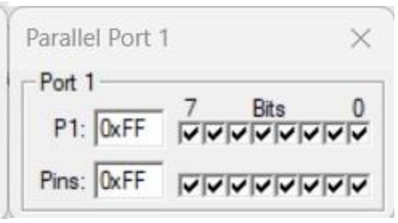


Parallel Port 1

Port 1

P1: 0x00 — Bits 7 … 0: ☐☐☐☐☐☐☐☐

Pins: 0x00 — ☐☐☐☐☐☐☐☐



Parallel Port 1

Port 1

P1: 0xFF — Bits 7 … 0: ✓✓✓✓✓✓✓✓

Pins: 0xFF — ✓✓✓✓✓✓✓✓

6.5: CONCLUSION

We have implemented hardware delay using timer mode 1.

---

7. Software Delay Function

7.1 : AIM

Creation of delay using user defined software delay function.

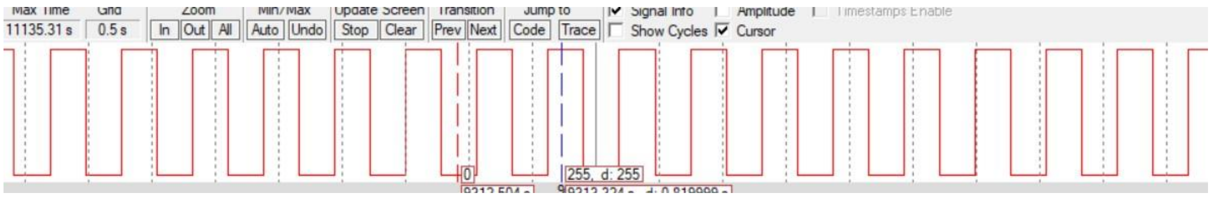
7.2 : CODE

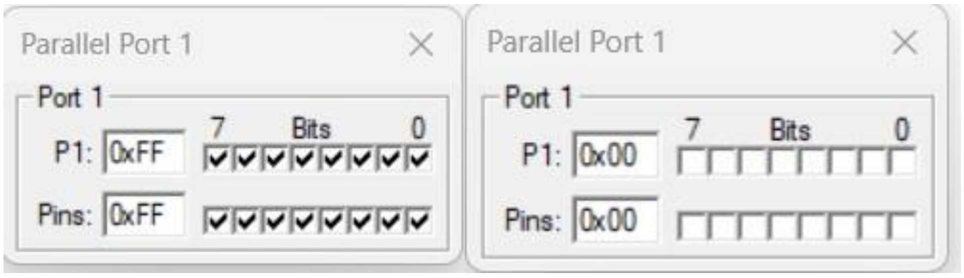
```
#include<reg51.h>

void delay(int d);
void main(){
while(1){
        P1 =0x00; delay(10);
        P1 =0xFF;
delay(10);
} } int a; void
delay(int d){
for(a =0;a<1100*d; a++);
}
```

7.3 : HEX CODE

```
:030000000208559E
:0C085500787FE4F6D8FD75810902082EBA
:10082E00E4F5907F0AFE1208007590FF7F0A7E00A5
:05083E0012080080EB30
:10080000AB07AA06E4F508F5097C047D4CAF03AEF
E
:1008100002120843C3E5099FEE6480F8E50864808E
:0E08200098500A0509E50970E0050880DC2201
:10084300EF8DF0A4A8F0CF8CF0A428CE8DF0A42E
C9
:02085300FE2283 :00000001FF
```

7.4: SIMULATED OUTPUT




Parallel Port 1

Port 1

P1: 0xFF

Pins: 0xFF

Parallel Port 1

Port 1

P1: 0x00

Pins: 0x00

7.5: CONCLUSION

We have generated a specific delay using user defined software delay function.

8. Auto reload mode

8.1: AIM

To generate auto reload mode, using timer T1 and mode 2

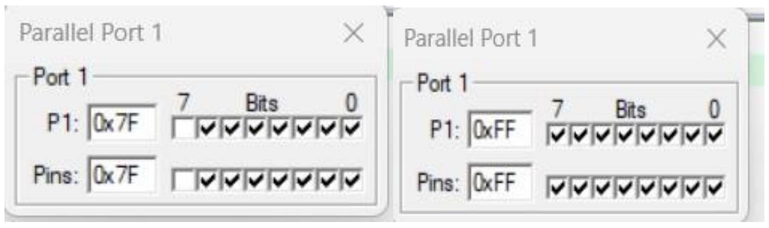
8.2 : CODE

```
ORG0000H
MOV TMOD,#20H
MOV TH1,#8CH
   START:SETB TR1
   HERE:JNB TF1,HERE
   CPL P1.7
   CLR TR1
   CLR TF1
   SJMP START
END
```

8.3 : HEX CODE

```
:10000000758920758D8CD28E308FFDB297C28EC2C
D
:030010008F80F3EB
:00000001FF
```

8.4 : SIMULATED OUTPUT



```
Parallel Port 1
Port 1
        7    Bits    0
P1:   0x7F   ☐✓✓✓✓✓✓✓
Pins: 0x7F   ☐✓✓✓✓✓✓✓

Parallel Port 1
Port 1
        7    Bits    0
P1:   0xFF   ✓✓✓✓✓✓✓✓
Pins: 0xFF   ✓✓✓✓✓✓✓✓
```

8.5 : CONCLUSION

We have generated auto reload mode using timer t1 and mod 2.

9. Interrupt using timer

9.1 : AIM

Interrupt using timer and same time display at port P1.

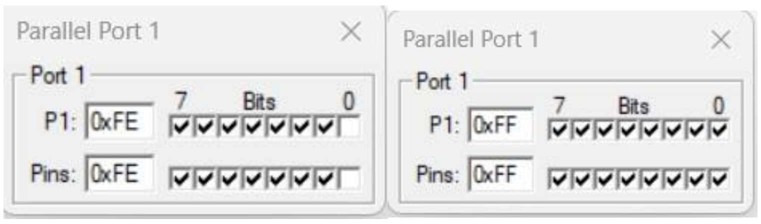
9.2 : CODE

```
ORG0000H;
   SJMP MAIN;
ORG000BH;
      CPL P1.0;
      RETI;
ORG0030H;
   MAIN:
MOV TMOD,#02H;
MOV TH0,#0B7H;
MOV IE,#82H;
SETB TR0;
BACK:
MOV P0,# $
         SJMP BACK
END
```

9.3 : HEX CODE

```
:02000000802E50
:03000B00B290327E
:10003000758902758CB775A882D28C75803B80FB60
:00000001FF
```

9.4 : SIMULATED OUTPUT


```
Parallel Port 1
Port 1
                7    Bits     0
P1:   0xFE      ✓ ✓ ✓ ✓ ✓ ✓ ✓ ☐
Pins: 0xFE      ✓ ✓ ✓ ✓ ✓ ✓ ✓ ☐

Parallel Port 1
Port 1
                7    Bits     0
P1:   0xFF      ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
Pins: 0xFF      ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
```

9.5 : CONCLUSION

We have generated interrupt at port p0 and display at the same time.

10. PATTERN Using PYTHON

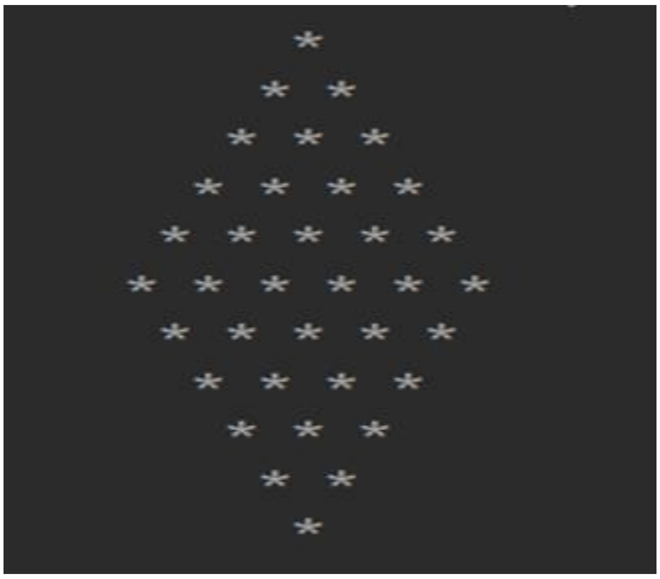
10.1 : AIM

To print Diamond-shaped pattern of stars

10.2 : CODE

```
rows=5 k =2* rows -2
foriinrange(0, rows): for
j inrange(0, k):
print(end=" ")     k = k -
1 for j inrange(0,i+1):
print("* ", end="")
print("")

k = rows -2

foriinrange(rows,-1,-1): for j
inrange(k,0,-1):
print(end=" ")     k = k
+1 for j inrange(0,i+1):
print("* ", end="")
print("")
```

10.3 : SIMULATED OUTPUT



```
     *
    * *
   * * *
  * * * *
 * * * * *
* * * * * *
 * * * * *
  * * * *
   * * *
    * *
     *
```

10.4 : CONCLUSION

We have created Diamond-shaped pattern of stars using python.