Experiment No: 01

Experiment Name: Design and develop a reprogrammable embedded computer using 8051 microcontroller.

Objective: The main objective of this experiment is to develop a reprogrammable embedded computer using 8051 microcontroller.

Theory:

Embedded computer: An embedded computer function as part of a ~~computer~~ complete device rather than being standalone computer typically performs highly specific function.

Microcontroller: A microcontroller is a small computer on a single metal oxide semiconductor (MOS). VLSI integrated circuit (IC) chip. It contains one or more CPU's along with memory and programmable input/output peripherals.

8051 microcontroller designed in 1981. It has 40 pins, 4 kb ROM and 128 bytes of RAM storage also has

2 16 bit timers. It has crystal frequency 12 Hz.

It has 8 bit data bus, 16 bit address bus and control register.

Apparatus:

i) 8051 microcontroller

ii) LED

iii) A connected ground

iv) Connecting wires

C program:

```
#include <reg51.H>
void Delay (unsigned int time);
void main (void)
{
  P2 = 0xAA;
  Delay (1000);
  P2 = 0x55;
  Delay (1000);
}
void Delay (unsigned int time)
{
  unsigned int i,j;
```

```
for (i=0 ; i<time ; i++)
{
  for (j=0; j<23; j++)
  {
  }
  }
}
```

Circuit Design:

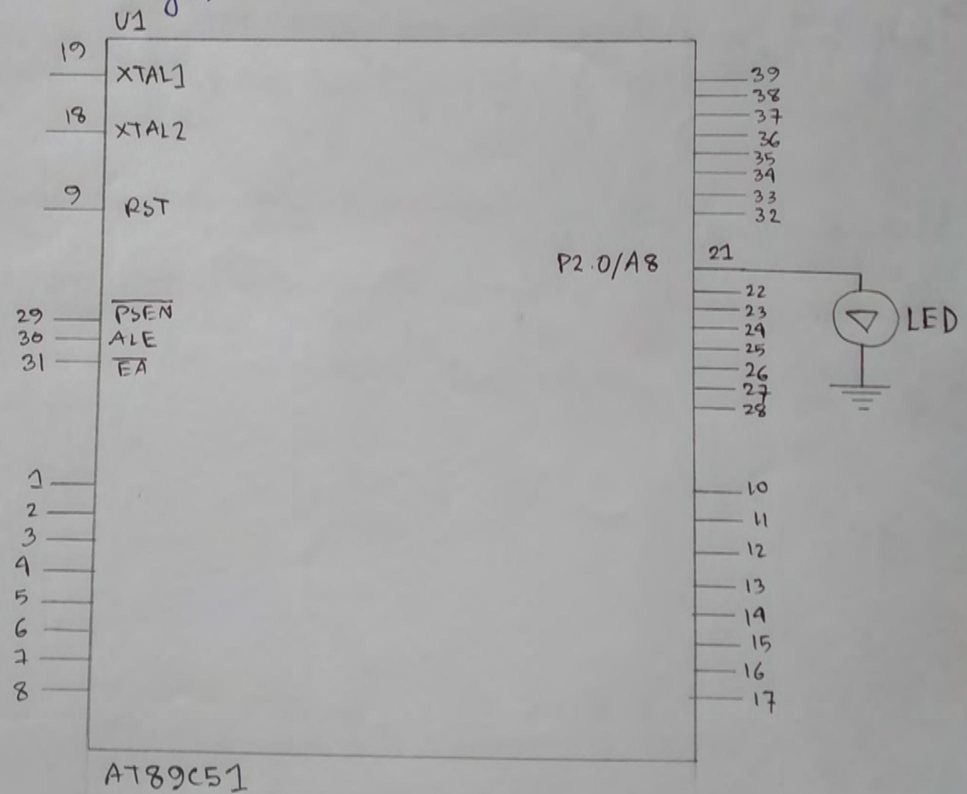


Fig: 8051 microcontroller with LED circuit design

Procedure:

1. Initially by taking AT89C51 microcontroller and LED for testing connected ~~w~~ the LED ~~wint~~ with pin 21.

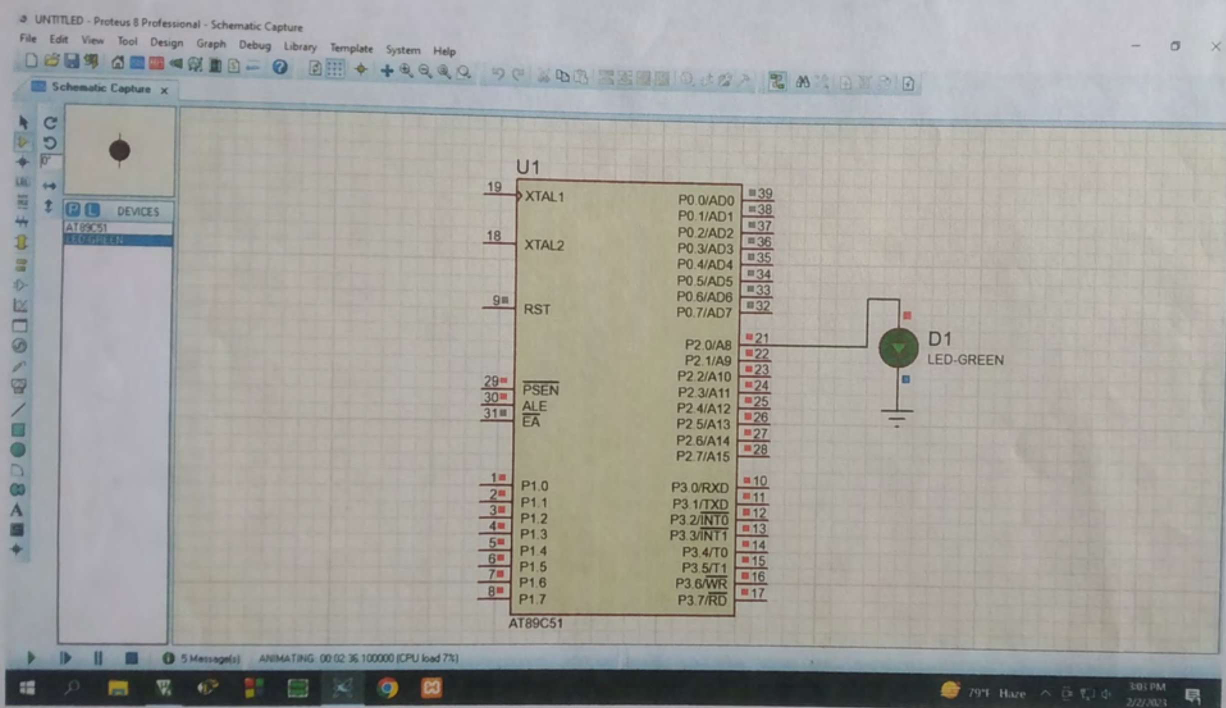
2. Connect the LED with the ground.

3. Click right on the AT89C51, browse the file created by using keil, set the hz as 11.0592.

4. Now, run the simulation and that is all for this experiment shows in fig 1.2.

Discussion: By this experiment, we can switch on/off on an LED with microcontroller. It may be used in real life in future for blinking of an LED.

Output:


UNTITLED - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
AT89C51
LED-GREEN
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
D1
LED-GREEN
5 Message(s)
ANIMATING: 00:02:36.100000 (CPU load 7%)
79°F Haze

Experiment No: 02

Experiment Name: Configure time control registers of 8051 and develop a program to generate delay.

Objective: The main objective of this experiment is to configure registers and develop program to generate delay.

Theory: An embedded computer function as part of a complete device rather than being a standalone computer. Typically, performs highly specific function. TCON is an 8 bit register used for generating interrupts internal or external.

ITO/II1 → used for timer interrupts.

IE0/IE1 → used for external interrupts.

TR0/TR1 → Timer 0/1 run control flag when it is 1, it means timer is running.

TF0 /TF1 → Timer 0/1 overflow fig. When it is I, then it is overflow. 8051 microcontroller designed in 1981 It has 40 pins, 4kb ROM and 128 bytes of RAM storage.

Apparatus:

i) AT89C51

ii) LED

iii) Ground

iv) Connecting wires

Source code:

```
#include <reg51.h>
void Delay (unsigned int time);
void main (void)
 {
  while (1)
  {
   P2 = P1;
   Delay (1000);
   }
 }
void Delay (unsigned int time)
 {
   unsigned int i,j;
   for (i=0 ; i<time ; i++)
   {
     for (j=0 ; j<23 ; j++)
     {
     }
   }
 }
```

Circuit Design:

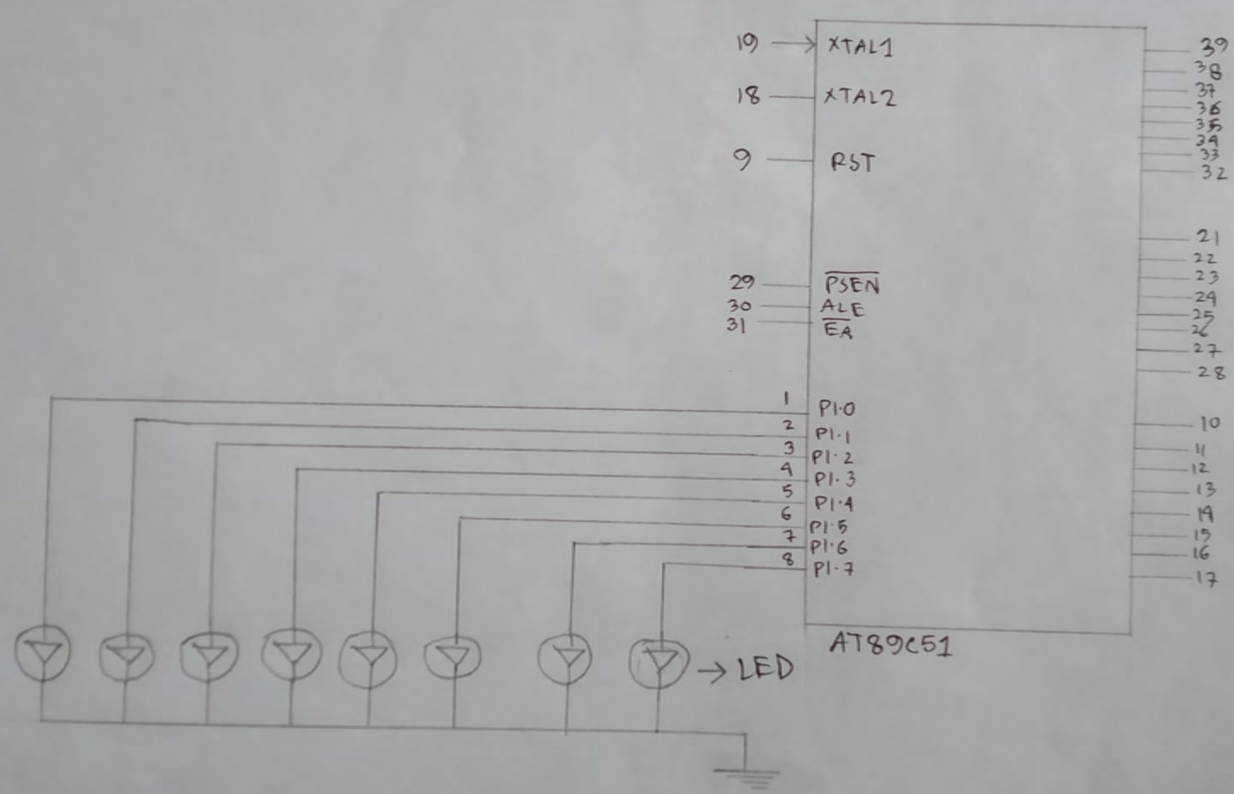


Fig: Time control reg with LED to determine delay

Procedure:

1. We will take AT89C51 microcontroller, LED and a ground.

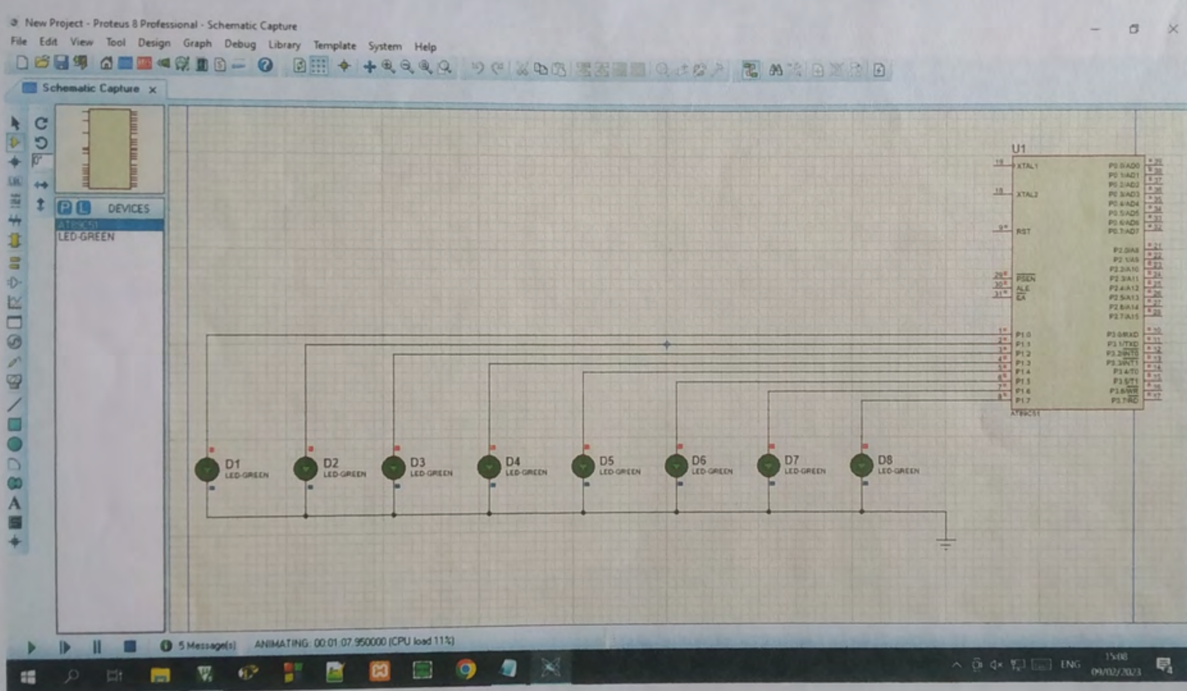
2. We will connect first LED with the pin 1.

3. We will connect second LED with pin 2 and sequentially connect the other LED.

4. All the LED's are connected to the ground.

Discussion: By this experiment, we will be able to make delay of the control register of 8051. This can be used in real life in future to turn off/on the LED's when there's a lot of LED.

Output:


New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
LED-GREEN
U1
XTAL1
XTAL2
RST
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
D1
D2
D3
D4
D5
D6
D7
D8
5 Message(s)

Experiment No: 03

Experiment Name: Use of general purpose I/O port of two controllers for data transfer.

Objectives: The main objective of this experiment is to configure two microcontroller for data transfer using general purpose I/O ports.

Theory: Dedicated GPI/O are always available. It has a frequency of a maximum 500MHz. It supports multiple standards. It can be configured either input or output.

8051 microcontroller designed in 1981. It has 40 pins, 4kb ROM and 128 bytes RAM storage. Also has 2,16 bit timers. It has crystal frequency 12 Hz. It has 8 bit data ~~bas~~ bus and 16 bit address bus and control signals.

Apparatus:

i) Two AT89C51 controllers

ii) LED (any color)

iii) Ground

iv) Connecting wires.

Source code:

Controller 1

```
#include <reg51.h>
void Delay (unsigned int time);
void main (void)
{
  while (1) {
    P2 = P1;
    Delay (1000);
   }
 }
void Delay (unsigned int time)
{
 unsigned int i,j;
 for (i=0 ; i< time ; i++) {
  for (j=0 ; j< 23 ; j++) {
  }
 }
}
```

Controller 2

```
# include <reg51.h>
void Delay (unsigned int time);
void main (void)
{
 P2 = 0xAA ;
 Delay (1000);
 P2 = 0x55 ;
Delay (1000);
}
void Delay (unsigned int time)
{
unsigned int i,j;
 for (i=0 ; i< time ; i++){
  for (j=0 ;j<23 ; j++){
   }
  }
}
```

Circuit Design:

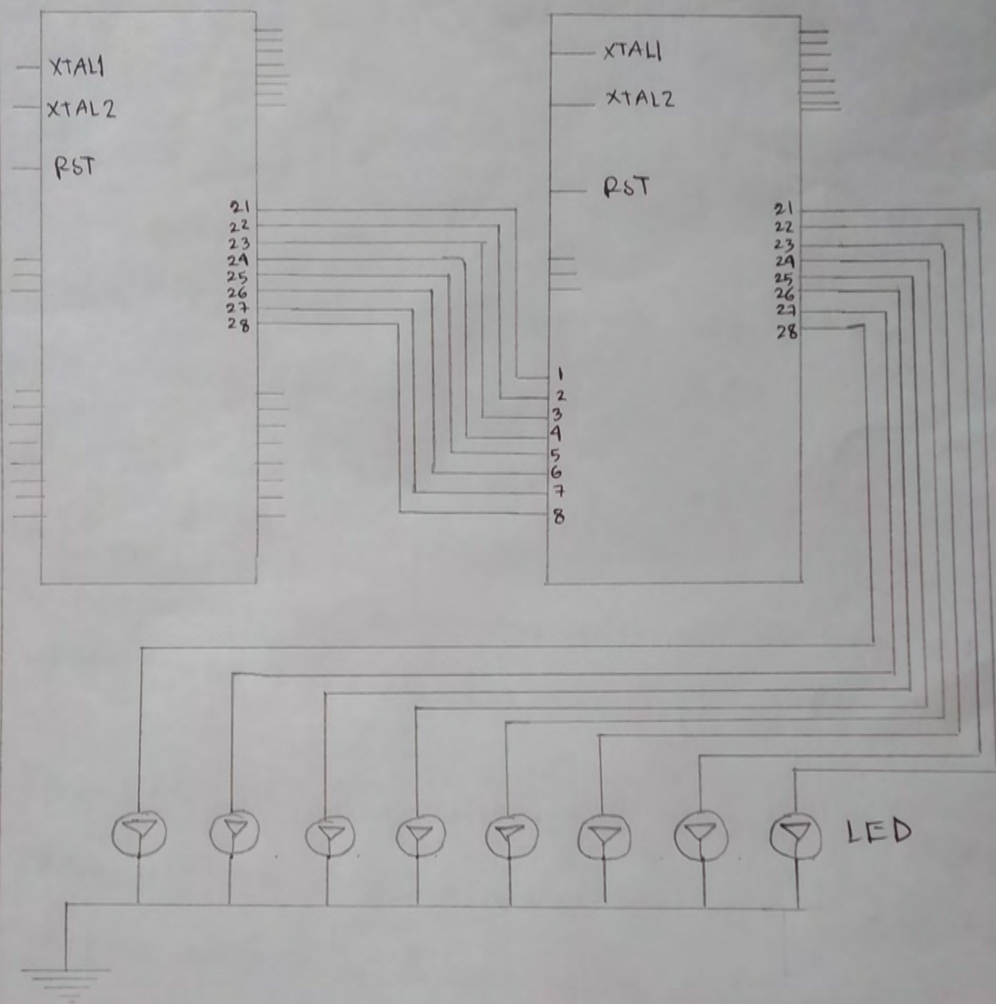


Fig: Transfering data between two controller

Procedure:

1. We will take two AT89C51 controller and eight LED's and a ground.

2. We will connect first microcontrollers pin no 21 and second controllers pin 1.

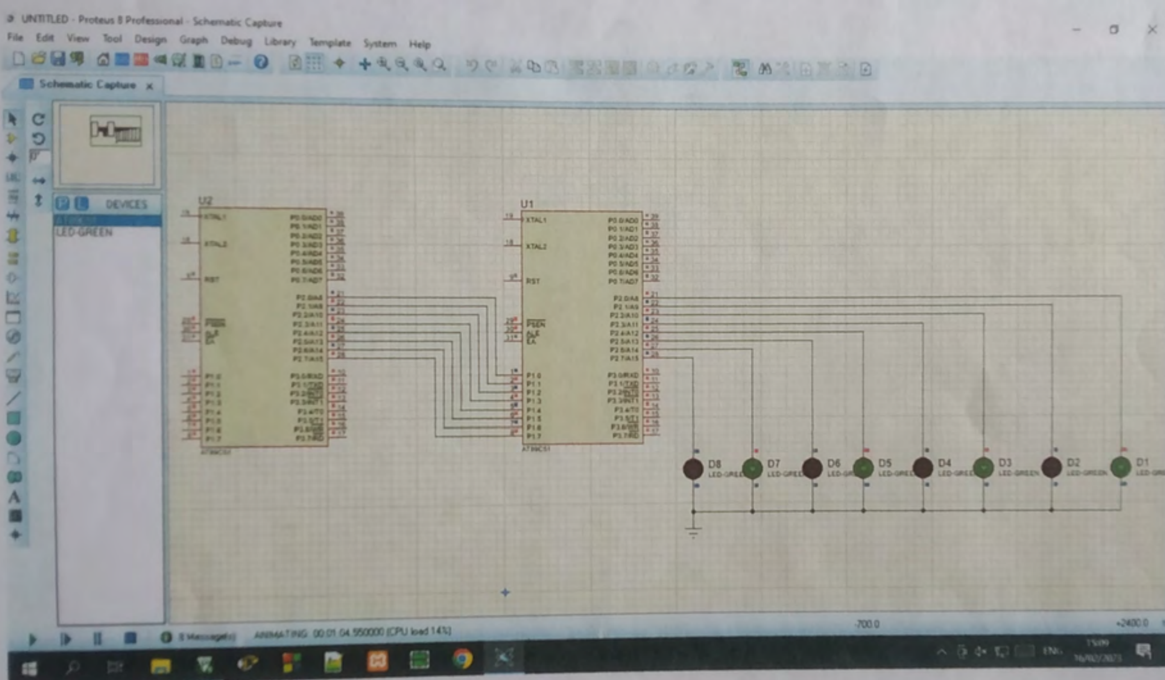
3. Then pin 21 of first controller and 2nd pin of 2nd controller and so on.

4. We will connect LED with pin 21 to pin 28, total 8LED.

5. We will connect a ground with all the LED at bottom and that's all for this experiment.

Discussion: This is a pattern of blinking of some LED's. It looks amazing when they blink serially. It may be helpfull in future while designed in lighting of any place or further more.

Output:


UNTITLED - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
LED-GREEN
U2
U1
D8
D7
D6
D5
D4
D3
D2
D1
ANIMATING: 00:01:04.950000 (CPU load 14%)

Experiment No: 04

Experiment Name: Simulate Binary Counter (8 bits) on LEDs.

Objective: The main objective of this experiment is to simulate binary counter on LEDs.

Theory: A binary counter is a hardware circuit that is made out of a series of flip-flops. The output of one flip-flop is sent to the input of the next flip-flop in the series. A binary counter can be either asynchronous or synchronous, depending on how the flip flop are connected together.

8051 microcontroller designed in 1981. It has 40 pins, 4kb ROM and 128 bytes of RAM storage. Also has 2 16 bit timers. It has crystal frequency 12 Hz. It has 8 bit data bus, 16 bit address bus and control ~~register~~ signals.

Apparatus:

i) 8051 microcontroller

ii) LED (any color)

iii) Ground

iv) Connecting wires

Source code:

```
#include <reg51.h>
void delay (int time);
void main ()
{
P1=00000000;
while (1)
{
P1++;
delay (100);
}
}
void delay (int time)
{
int i,j;
for (i=0; i<=time; i++)
{
for (j=0; j<=23; j++)
{
}
}
}
```

Circuit design:

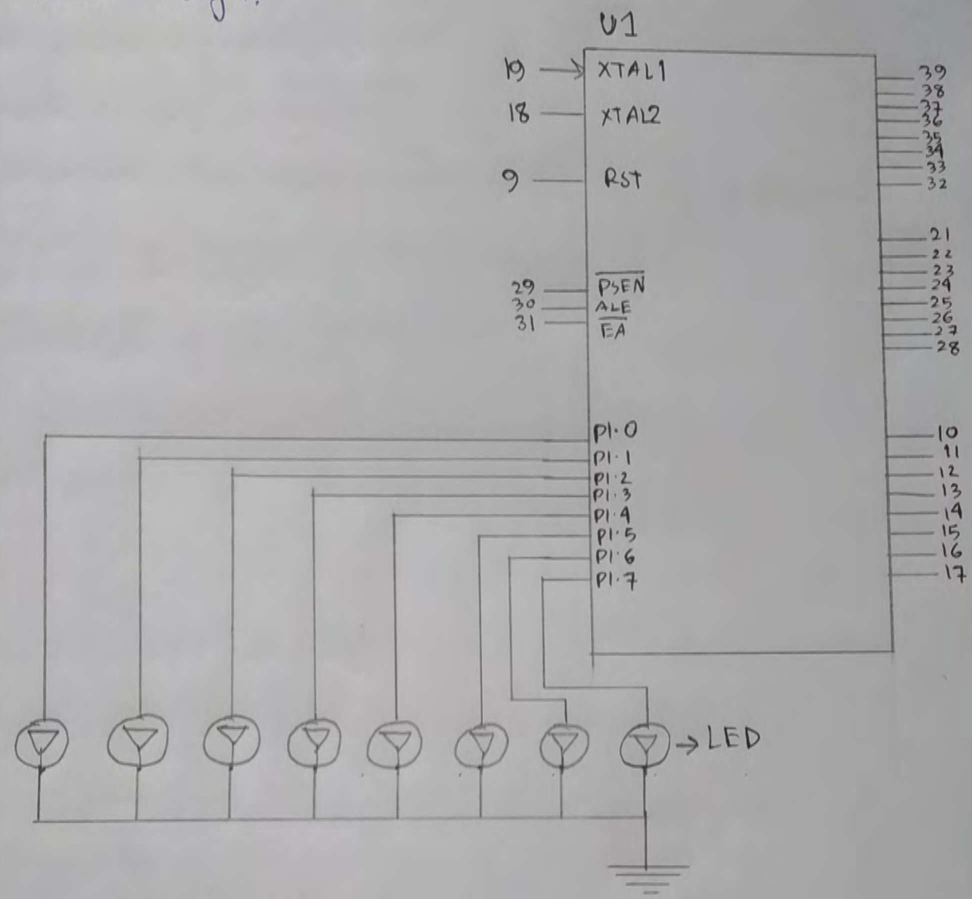


Fig: Binary counter simulator on LED

Procedure:

1. Firstly we will take AT89C51 controller, some LED and a ground.

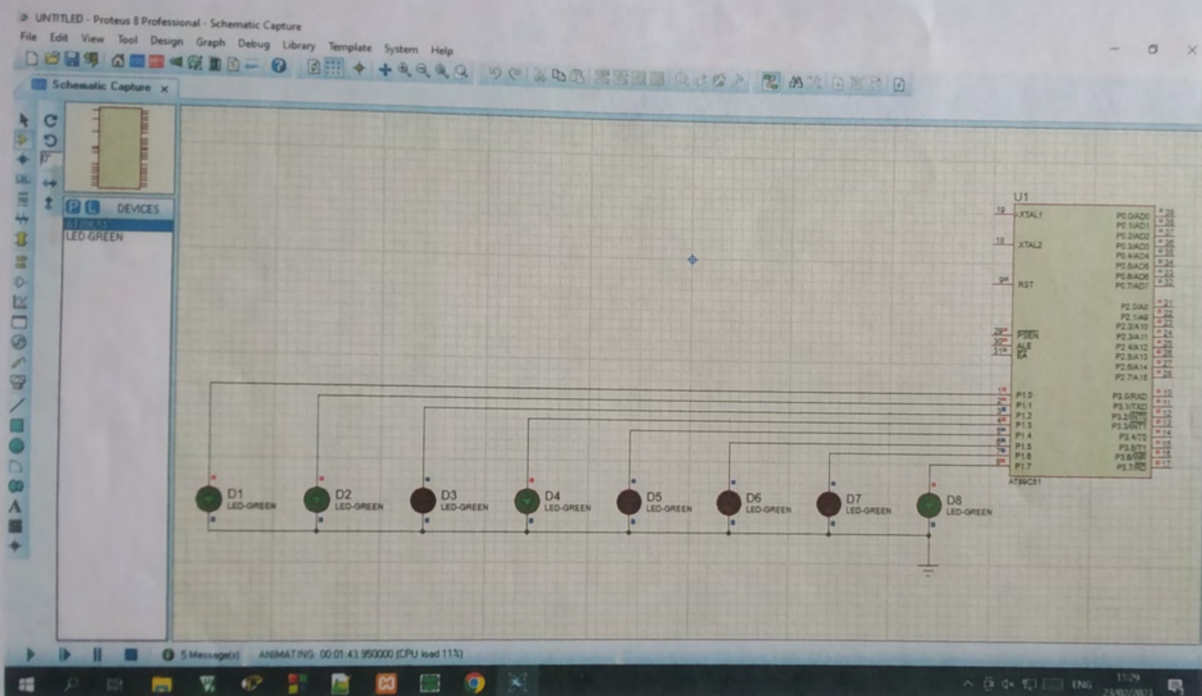
2. We will connect pin 1 with first LED.

3. Then the second LED with the pin no 2 and so on.

4. A ground will be connected with all the LEDs and that's all for this experiment.

Discussion: This experiment shows us a different color matching pattern which can be used in real life while designing home, office or anything like this.

Output:


UNTITLED - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
LED-GREEN
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
AT89C51
D1
D2
D3
D4
D5
D6
D7
D8
LED-GREEN
ANIMATING

Experiment No: 05

Experiment Name: To interface 8 LED at I/O ports and create different patterns.

Objective: The main objective of this experiment is to make different patterns and also with 8 LEDs at I/O ports.

Theory: An I/O port is a socket on a computer that a cable is plugged into. The port connets the CPU to a peripheral device via a hardware interface or to the network via a network interface. I/O port is an address to transfer data.

8051 microcontroller designed in 1981. It has 40 pins, 4kb ROM and 128 bytes of RAM storage. Also has 2 16 bit timers. It has crystal frequency. 12 Hz. It has 8 bit data bus, 16 bit address bus and control signals.

Apparatus:

i) 8051 microcontroller

ii) LED (any color)

iii) A ground

iv) Connecting wires

Source code:

```
# include <reg51.h>
void delay ();
void main ()
{
 while (1)
 {
 P1 = 0xAA ;
 delay ();
 P1 = 0x55 ;
 delay ();
 }
}
void delay()
{
unsigned int i,j ;
for (i=0 ; i<23 ; i++)
{
 for (j=0 ; j<1000 ; j++)
 {
 }
 }
}
```

Circuit design:

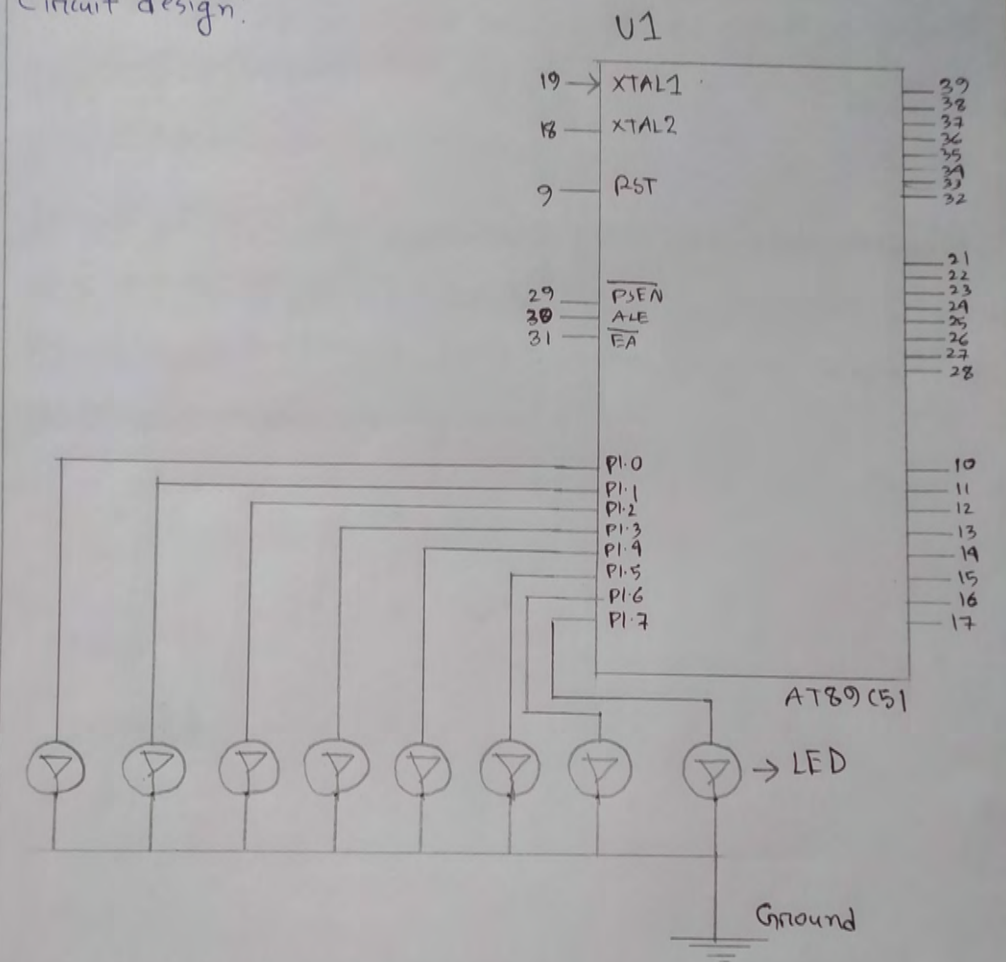


Fig: Different patterns of light with I/o ports.

Procedure:

1. We will take AT89C51 controller, some LED and a ground.

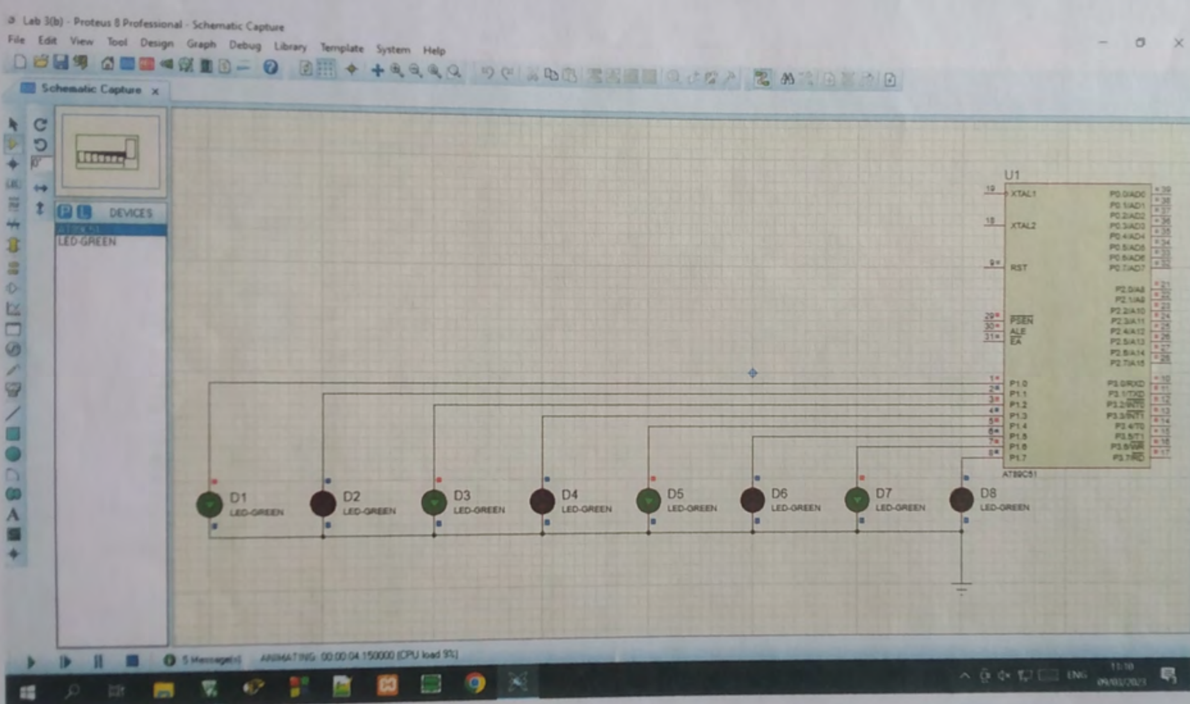
2. We will connect first LED with pin no 1.

3. second pin will connect to the second LED and so on.

4. All the LED will the be connected to the ground, then we will run this to see different patterns

Discussion: By this experiment we can also decorate our home, office or like this place with LEDs. It can be used in different purpose for bigger business or anything that can bring profit.

Output:


Lab 3(b) - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
LED-GREEN
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
AT89C51
D1
D2
D3
D4
D5
D6
D7
D8
LED-GREEN

Experiment NO: 06

Experiment Name: Serial I/O: Configure 8051 serial port for Asynchronous serial communication.

Objective: The objective of this experiment is to configure 8051 serial port for asynchronous serial communication.

Theory: Asynchronous serial communication is a form of serial communication in which the communicating endpoints intertafaces are not continuously synchronized by a common clock signals. Instead of common synchronization signal the data steam contains synchronization ~~signal~~ information in form of start and stop signals, before and after each transmission respectively. The start signals prepare the receiver for arrival of data and the stop signal resets its state to enable triggering of a new sequence.

Apparatus:

I) 8051 microcontroller

II) virtual terminal

III) Connected wires

Source code:

```
#include<reg51.h>
void send (char x);
void main (void)
{
 TMOD = 0x20;
 TH1 = 0xFD;
 SCON = 0x50;
 TR1 = 1;
 send('T');
 send ('A');
 send('M');
 send ('I');
 send ('M');
 send ('A');
 while(1);
}
void send (char x)
{
 SBUF = x;
 while(TI==0);
 TI=0;
}
```

Circuit design:

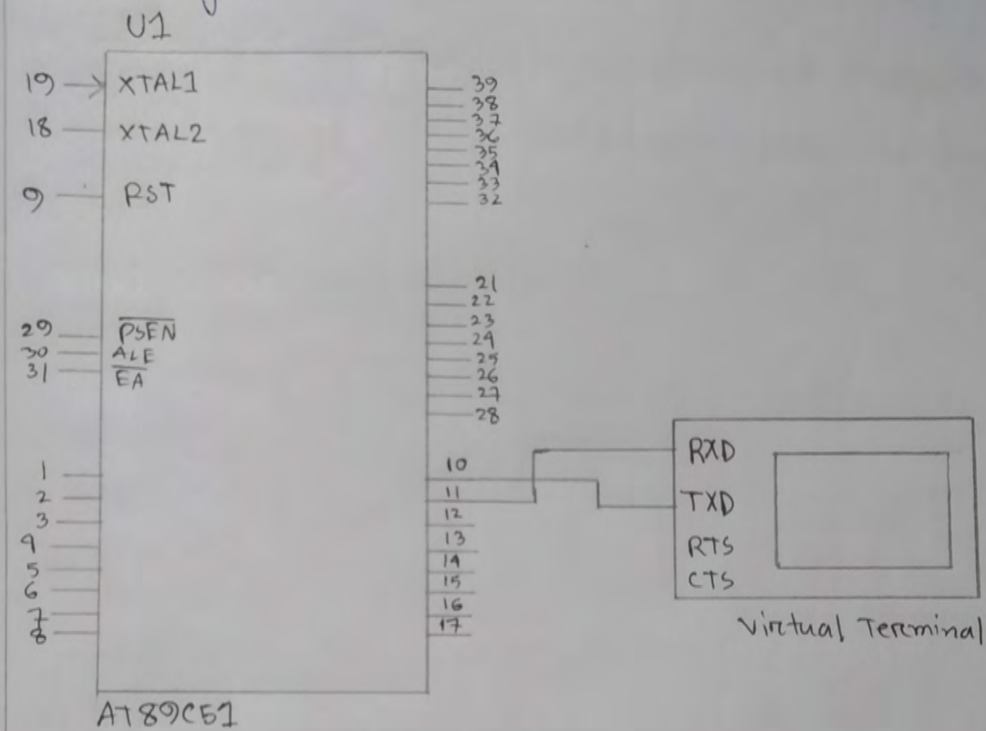


Fig: Asynchronous serial communication

Procedure:

1. We will take AT89C51 microcontroller and a virtual terminal.

2. Then we will connect 'RXD' port with pin no 11 and 'TXD' port with pin no 10.

3. Then we will show the output of the experiment.

Discussion: This asynchronous serial transmission can be used show number of sequence of characters. Different types of strings are shown here in virtual terminal.

Experiment No: 07

Experiment Name: To demonstrate seven segments LED display and generate counting from 0 to 99.

Objective: The main objective of this experiment is to demonstrate seven segments LED display and generates counting from 0 to 99.

Theory: A seven segment display is a form of electronic display device for displaying decimal numerals that is an alternative to more complex dot matrix analysis. This displays are widely used in digital clocks, electronic meters, basic calculator etc.

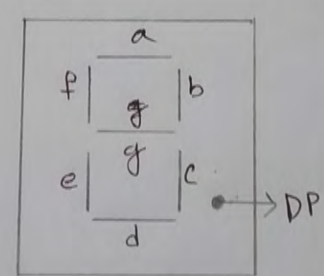


Fig: 7 segment display

Apparatus:

i) AT89C51 microcontroller
ii) Power
iii) Ground
iv) 7 segment display
v) Connecting wires.

Source code:

```
#include <reg51.h>
void delay (unsigned int ms)
{
unsigned int i,j;
for (i=0 ; i<ms ;i++)
{
for (j=0 ; j<1275 ;j++)
{
}
}
}
void main (void)
{
Char number [] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D,
                  0x7D, 0x07, 0x7F, 0x6F};
int i,j;
P2 = 0x00 ;
P3 = 0x00 ;
while (1)
{
for (i=0 ; i=9 ;i++)
{
P2 = number [i];
for (j=0 ; j<=9 ; j++)
{
P3 = number [j];
delay (50) ;
}
}
}
}
```

Circuit design:

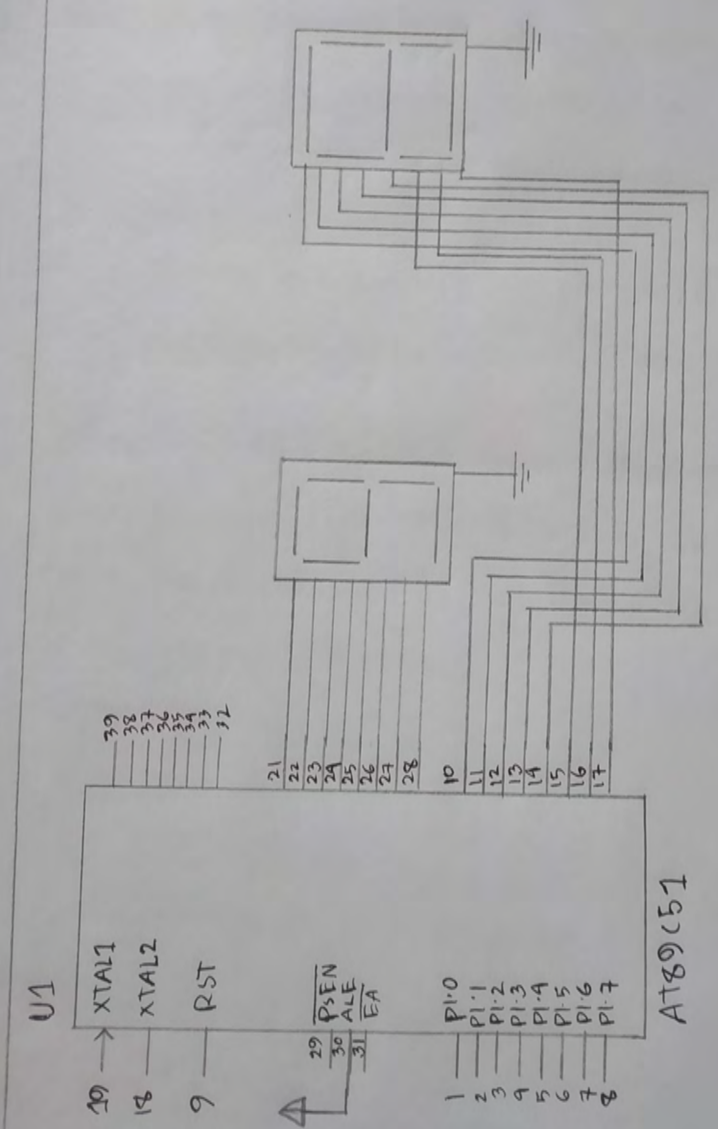


Fig: Seven segment display with AT89C51 microcontroller

Procedure:

1. we will take AT89C51 microcontroller, 7 segment display, ground and power.

2. First seven segment display pins will connected from pin 21 to 27 and last pin will connect to ground.

3. second seven segment display's pins will connect with microcontroller's pin 10 to ~~17~~ 16. Last pin will connect to ground.

4. A power will connect to pin 29 ($\overline{PSEN}$). That's all for this experiment, then we will show the output.

Discussion: This seven segment display used like everywhere. Stopwatch process work through this process. Clock, machines, meters etc are all depends on this seven segment display.

Experiment No: 08

Experiment Name: Interface 8051 with D/A converter and generate square wave on oscilloscope.

Objective: The main objective of this experiment is to convert with D/A converter and generate square wave on Oscilloscope.

Theory: Digital/Analog converter is a device that converts digital code into analog signals mostly for audio. D/A converters (DACs) convert digital audio samples into the analog waveforms sent to audio amplifier and speakers. It also convert digital TV broadcast to analog TV convert box.

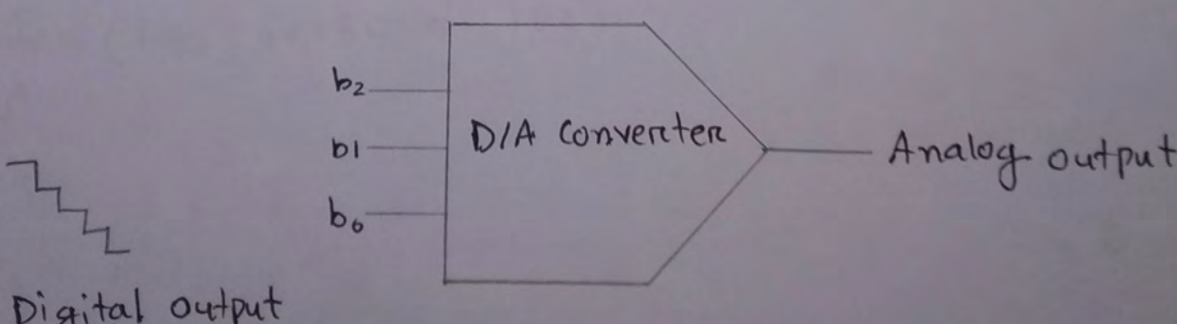


Fig: D/A converter

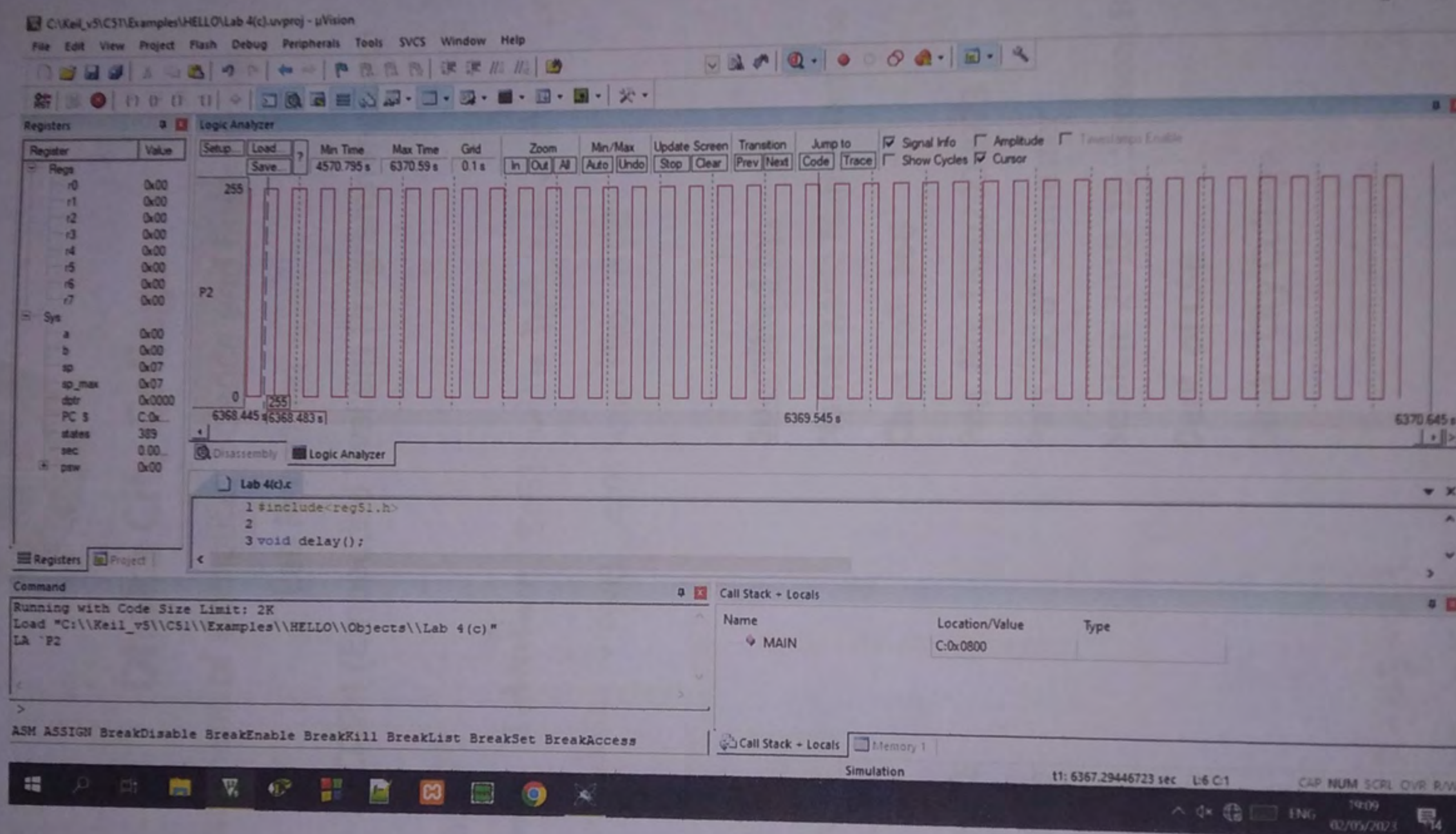
Source code:

```
# include <reg 51.h>
void delay();
void main ()
{
P2 = 0x00 ;
while (1)
{
P2 = 0xFF ;
delay ();
P2 = 0x00;
delay();
}
}
void delay()
{
int i;
for(i=0; i<=5000 ; i++)
{
}

}
```

Discussion: For this experiment we don't need any circuit design. By single software we can do that. Different sizes of square wave are mady by D/A converter.

Output:


C:\Keil_v5\C51\Examples\HELLO\Lab 4(c).uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Logic Analyzer
Min Time 4570.795 s
Max Time 6370.59 s
Grid 0.1 s
P2
6368.445 s
6369.545 s
6370.645 s
Lab 4(c).c
1 #include<reg51.h>
2
3 void delay();
Command
Running with Code Size Limit: 2K
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\Lab 4(c)"
LA `P2
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Call Stack + Locals
Name
MAIN
Location/Value
C:0x0800
Type
Simulation
t1: 6367.29446723 sec

Experiment No: 09

Experiment Name: Interface 8051 with D/A converter and generate Triangular waves on oscilloscope.

Objective: The main objective of this experiment is to interface 8051 with D/A converter and generate Triangular waves on oscilloscope.

Theory: Digital/Analog converter is a device that converts digital code into analog signal mostly for audio. D/A converter covert digital audio samples into the analog waveforms sent to audio amplifier and speakers. It also convert digital TV broadcast to analog TV convert box.

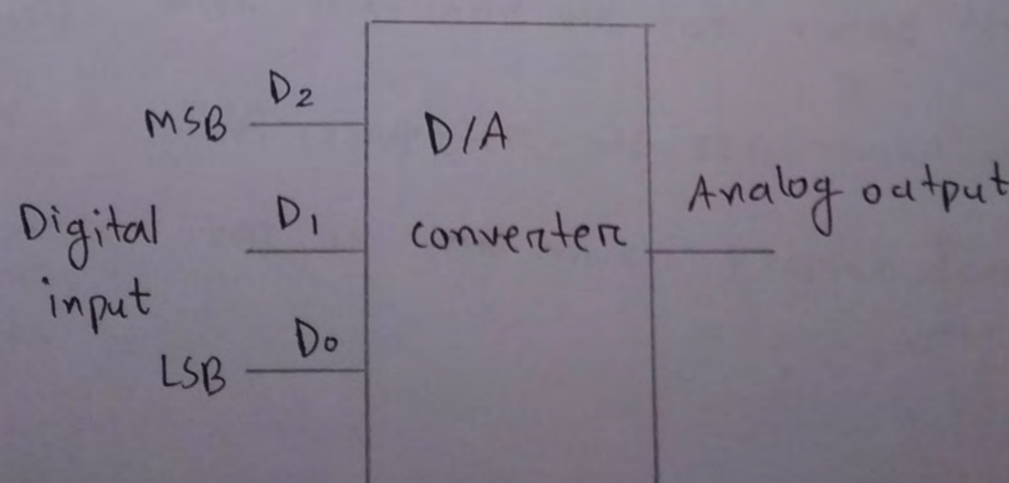


Fig: D/A converter.

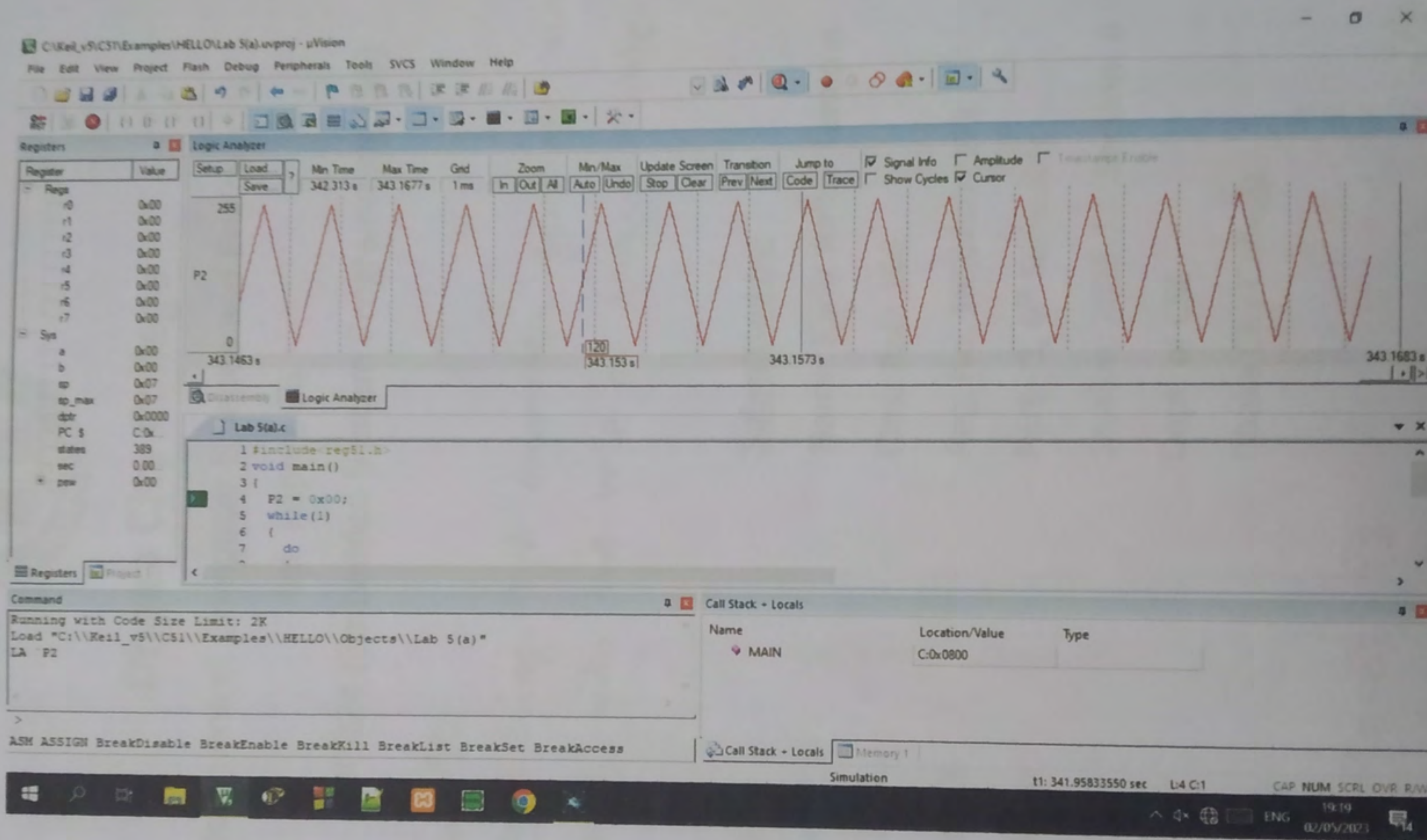
Source code:

```
#include<reg51.h>
void main ()
{
P2 = 0x00;
while (1)
{
do
{
P2 +- 0x05;
}
while(P2<0xFF);
do
{
P2- = 0x05;
}
while (P2>0x00);
}
}
```

Discussion: With this code we can make triangular wave. By D/A converter we can make any shape of wave that is very much helpful for other works.

A2

Output:


C:\Keil_v5\C51\Examples\HELLO\Lab 5(a).uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Logic Analyzer
Min Time 342.313 s
Max Time 343.1677 s
Grid 1 ms
P2
255
0
343.1463 s
343.153 s
343.1573 s
343.1683 s
Lab 5(a).c
1 #include<reg51.h>
2 void main()
3 {
4 P2 = 0x00;
5 while(1)
6 {
7 do
Command
Running with Code Size Limit: 2K
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\Lab 5(a)"
LA P2
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Call Stack + Locals
Name
MAIN
Location/Value
C:0x0800
Type
Simulation
t1: 341.95833550 sec
L4 C1

Experiment No: 10

Experiment Name: Using D/A converter generate sine wave on oscilloscope with the help of lookup table.

Objective: The objective of this experiment is to use D/A converter and generates sine wave on oscilloscope with the help of lookup table.

Theory: D/A (Digital to Analog) converter converts digital signals into analog signals. It uses everywhere. Television formula use this technology.

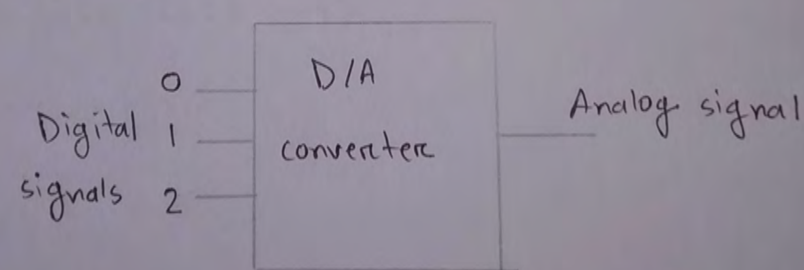


Fig: D/A converter

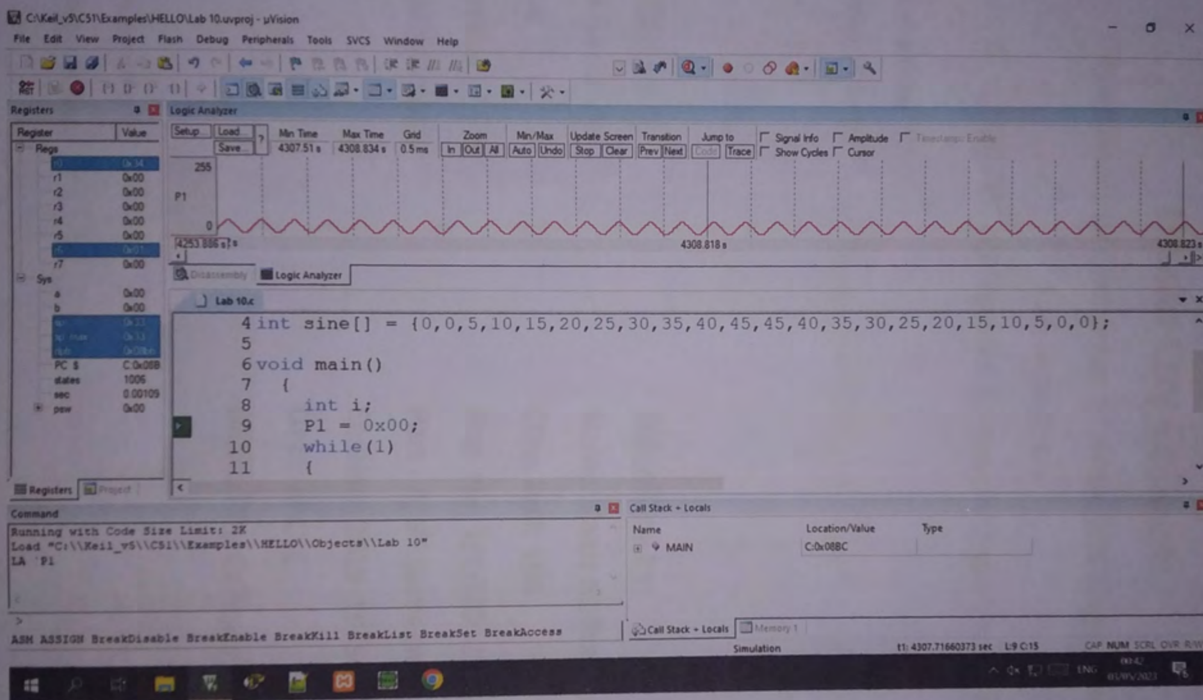
A sine wave is a geometric waveform that oscillates periodically and defined by the function $y = \sin x$.

Source code:

```
#include <reg51.h>
#include < stdio.h>
int sine [ ] = {0,0,5,10,15,20,25,30,35,40,45,45,40,
                35,30,25,20,15,10,5,0,0};

void main ()
{
int i;
 P1 = 0x00 ;
 while (1)
 {
for (i=0 ;i<22 ; i++)
{
P1 = sine [i];
 }
 }
}
```

Discussion: With D/A converter we can generate triangular, square and even sine wave. sine wave works on 'y' axis. That's why sine function is y= sin(x). This kinds of experiment needed to design different waves making.


C:\Keil_v5\C51\Examples\HELLO\Lab 10.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Logic Analyzer
Min Time 4307.51 s
Max Time 4308.834 s
Grid 0.5 ms
Zoom
Min/Max
Update Screen
Transition
Jump to
Signal Info
Amplitude
Show Cycles
Cursor
255
P1
0
4308.818 s
Logic Analyzer
Lab 10.c
4 int sine[] = {0,0,5,10,15,20,25,30,35,40,45,45,40,35,30,25,20,15,10,5,0,0};
5
6 void main()
7 {
8 int i;
9 P1 = 0x00;
10 while(1)
11 {
Command
Running with Code Size Limit: 2K
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\Lab 10"
LA `P1
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Call Stack + Locals
Name
Location/Value
Type
MAIN
C:0x08BC
Simulation

Experiment No: 11

Experiment Name: Interface stepper motor with 8051.

Objective: The objective of this experiment is to interface stepper motor with 8051.

Theory: A stepper motor also known as step motor or stopping motor is a brushless DC electric motor that divides a full rotation into a number of equal steps. It's position can be commanded to move ~~on~~ and hold at one of these steps without any position sensor for feedback, as long as the motor is correctly sized to the application in respect to torque and speed.

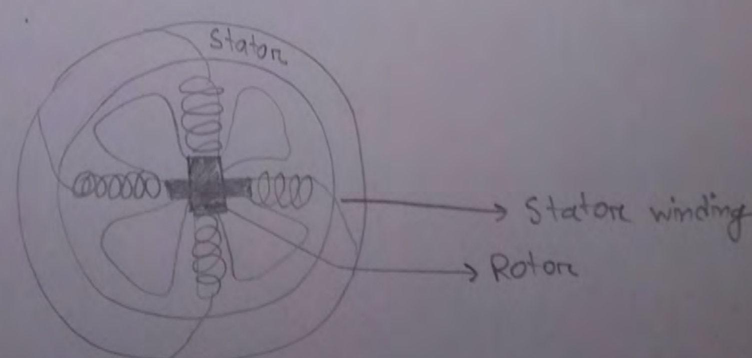


Fig: stepper motor

Source code:

```
#include <reg51.h>
void delay()
{
int i,j;
for (i=0; i<=100; i++)
{
for (j=0; j<100; j++)
{
}
}
}
void main()
{
while(1)
{
P2 = 0x09;
delay();
P2 = 0x03;
delay();
P2 = 0x06;
delay();
P2 = 0x0C;
delay();
}
}
```

Procedure:

1) At first we will take AT89C51 microcontroller and a stepper motor.

2) There we will connect motor's left ~~with~~ side's upper and lower pin with microcontroller's pin no: 21 and pin no: 24.

3) Then we will connect motor's left side's upper and lower pin with the controller's pin no 21 and pin no 22.

4) The middle pin of motor of both sides will connect to the ground.

By this process we can see the output.

Circuit design:

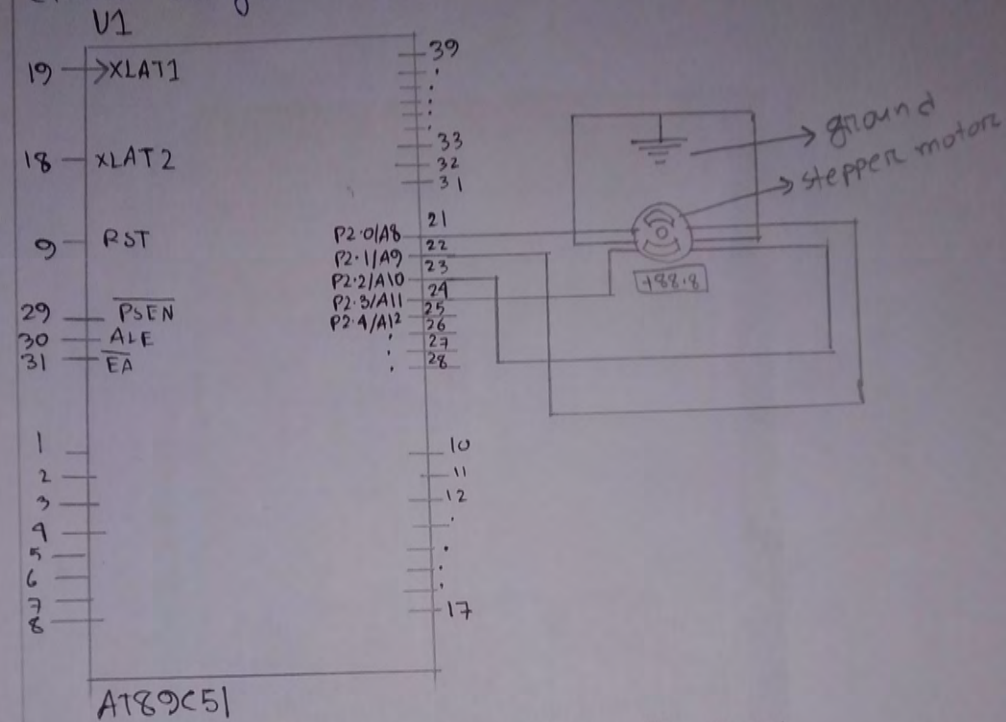
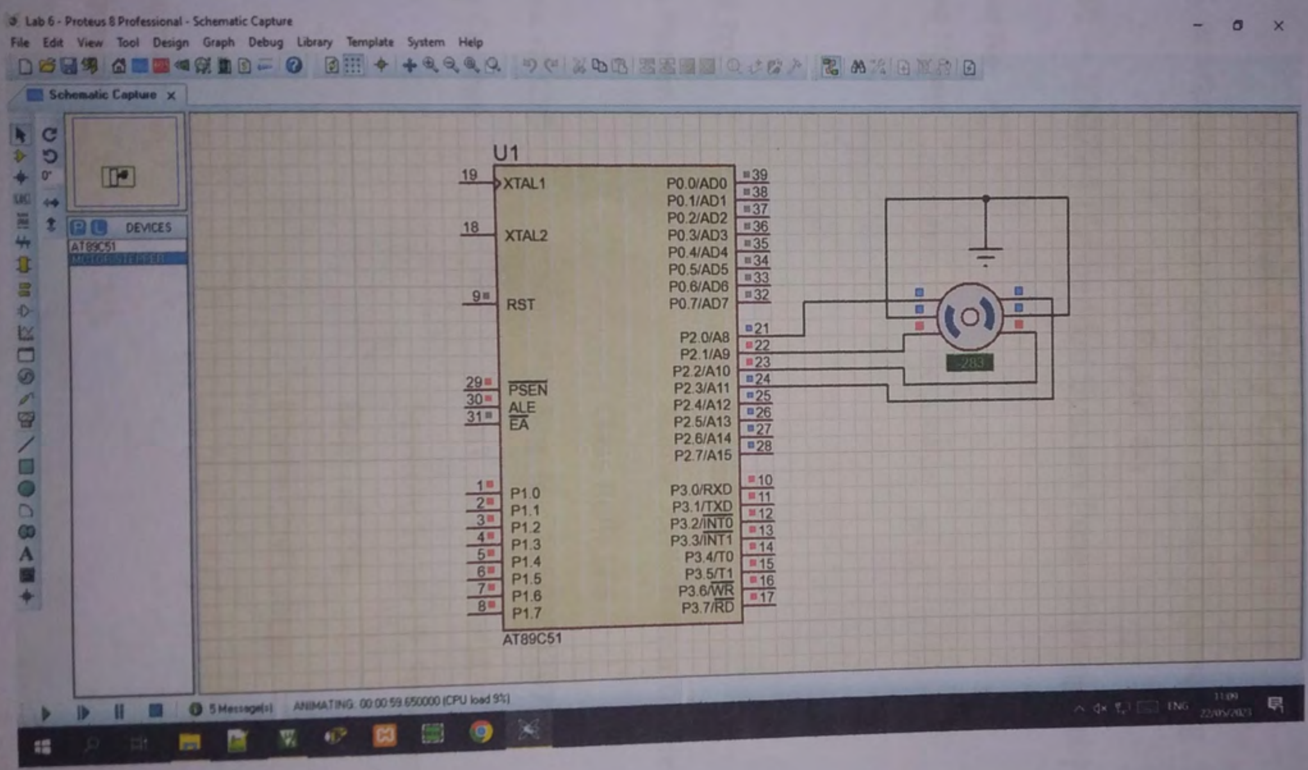


Fig: stepper motor with 8051

Discussion: stepper motor nowadays uses many sectors. We can control it's rotation, angle and other things. By using this motor we can convert electric power to machine power and can do many important works.


Lab 6 - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
AT89C51
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
ANIMATING: 00:00:59.650000 (CPU load 9%)

8

5

51

Experiment No: 12

Experiment Name: Generate Traffic signals.

Objective: The main objective of this experiment is to generate traffic signals.

Theory: Traffic signals, light or spotlight - known as robots in South Africa. It consist normally of three signals. The regular traffic light colors are red, yellow, and green arranged vertically or horizontally in that order. It was first introduced December 1868 on Parliament square in London to reduce need for police officers to control.

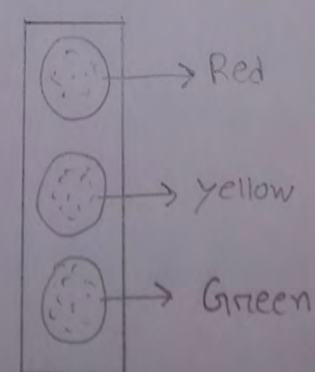


Fig: Traffic Light / signals.

Source code:

```
# include <reg51.h>
sbit red = P2^0 ;
sbit yellow = P2^1 ;
sbit green = P2^2 ;
void delay (int time) ;
void main()
{
red = yellow = green =0 ;
while (1).
{
red =1 ;
delay (1000) ;
red =0 ;
yellow =1;
delay (200) ;
yellow =0;
green =1;
delay (1000) ;
green =0 ;
yellow =1 ;
delay (200) ;
yellow = 0 ;
}
```

```
void delay (int time)
{
int i,j;
for(i=0;i<time;i++)
{
for(j=0;j<1000;j++)
{
}
}
}
```

Circuit design:

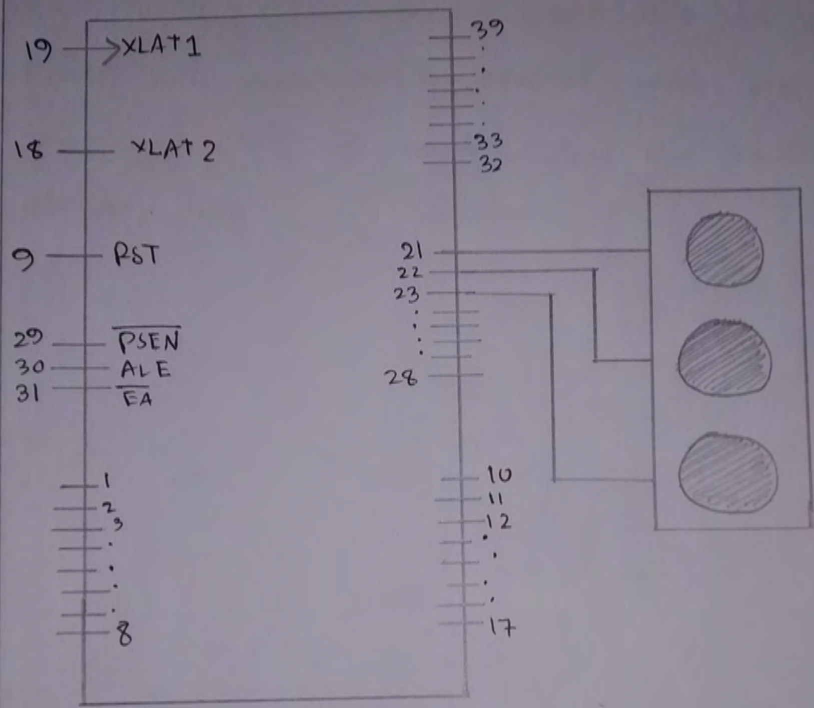


Fig: Traffic Signals

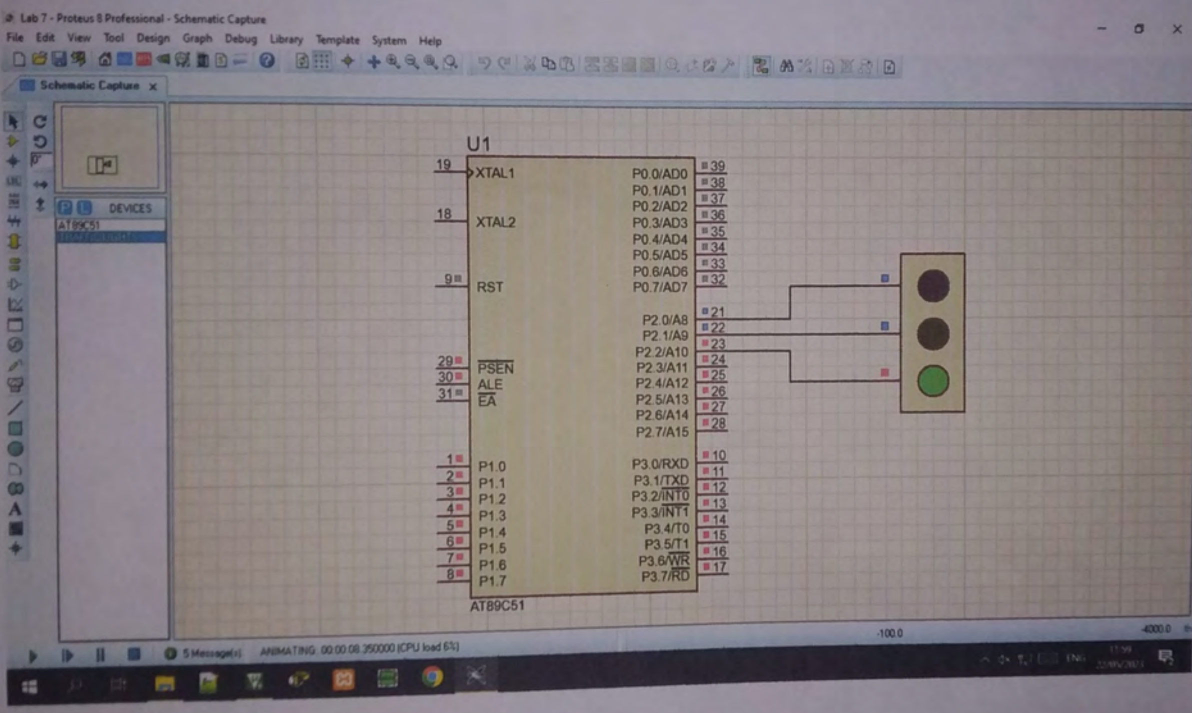
Procedure:

1) we will take At89C51 controller and traffic light.

2) Pin 21, 22 and 23 will connect to as follows: first pin, second pin and third pin of traffic light.

That's all for this experiment.

Discussion: Traffic signals are widely used nowadays. Everywhere we can see this in road. Police can now easily control with the help of this light. We all must follow the traffic rules at any cost.


Lab 7 - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
AT89C51
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
ANIMATING

56