1. keilv5
2. Project
3. New uVision project
4. File Name
5. Search (AT89C51)
6. OK
7. NO
8. Target1 (Right click)
9. Option for Target1
10. Xtal(MHz): 11.0592
11. Output->Tick(Create Hex file)
12. Ok
13. Target1 (+)->Source Group1(right)->Add New Iteam
14. C file
15. Name:- Pract1
16. Add
17. Write code
18. Target1(Right click)->Build Target
19. Then open proteus
20. New project
21. Name :- pehele se likha huva
22. Next…….Finish
23. Then click P
24. AT89C51
25. Ok -->P-->LED
26. After Diagram🡪Click on AT89C51 diagram🡪clock frequency🡪11.0592MHz
27. Program File🡪Select File Name(code vala)🡪ok🡪 Run

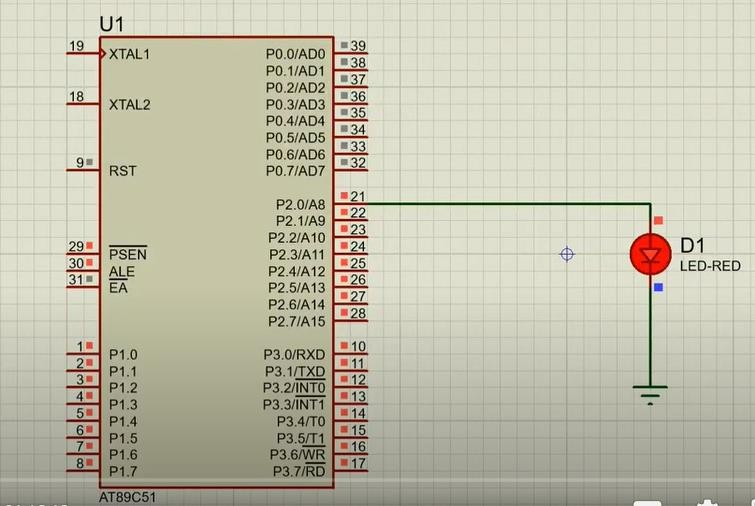
Pratical1:

Design and develop a reprogrammable embedded computer using 8051 microcontrollers and to show the following aspects.

1. Programming
2. Execution
3. Debugging Code:

|  |
| --- |
| #include<reg52.H> |
| sbit LED = P2^0; |
| void Delay(void); |
| void main(void) |
| { |
| while(1) |
| { |
| LED = 0; |
| Delay(); |
| LED = 1; |
| Delay(); |
| } |
| } |
| void Delay(void) |
| { |
| int j; |
| int i; |
| for(i = 0; i < 10; i++) |
| { |
| for(j = 0; j < 10000; j++) |
| { |
| } |
| } |
| } |

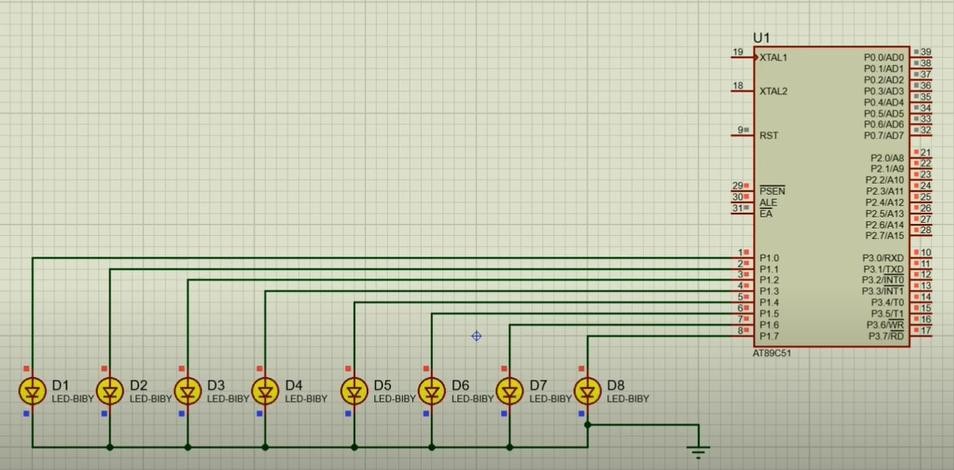
Output:



|  |  |
| --- | --- |
| **2. A** | Configure timer control registers of 8051 and develop a program to generate given time delay. |

Code:

|  |
| --- |
| #include<reg51.H> |
| void Delay(void); |
| void main(void) |
| { |
| while(1) |
| { |
| P1 = 0xFF; // Make all bits of P1 high |
| Delay(); |
| P1 = 0x00; // Make all bits of P1 low |
| Delay(); |
| } |
| } |
| void Delay(void) |
| { |
| int j; |
| int i; |
| for(i = 0; i < 1000; i++) |
| { |
| } |
| for(j = 0; j < 1000; j++) |
| { |
| } |
| } |

Output:

|  |  |
| --- | --- |
| **B** | To demonstrate use of general purpose port i.e. Input/ output port of two controllers for data transfer between them. |

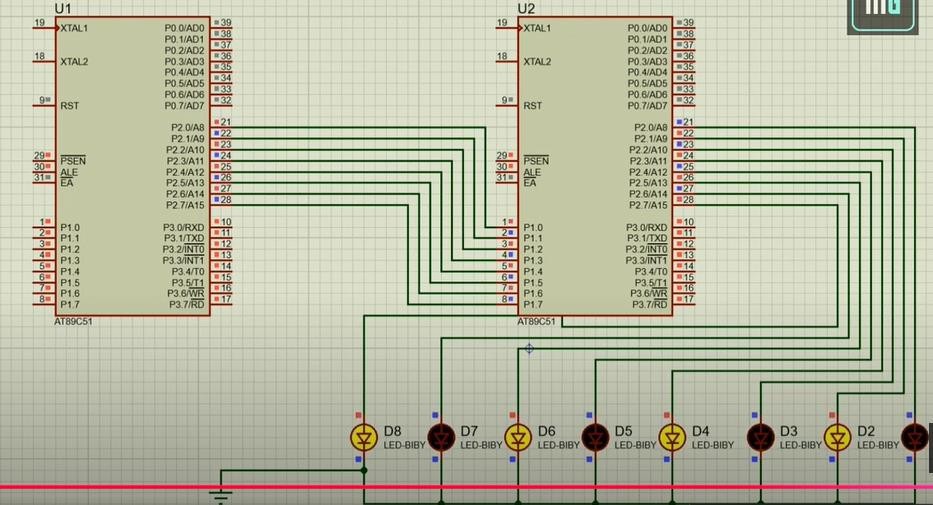
Code control 1:

|  |
| --- |
| #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++){ |
| } |
| } |
| } |

Control2 :

|  |
| --- |
| #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++){ |
| } |

|  |
| --- |
| } |
| } |

Output:

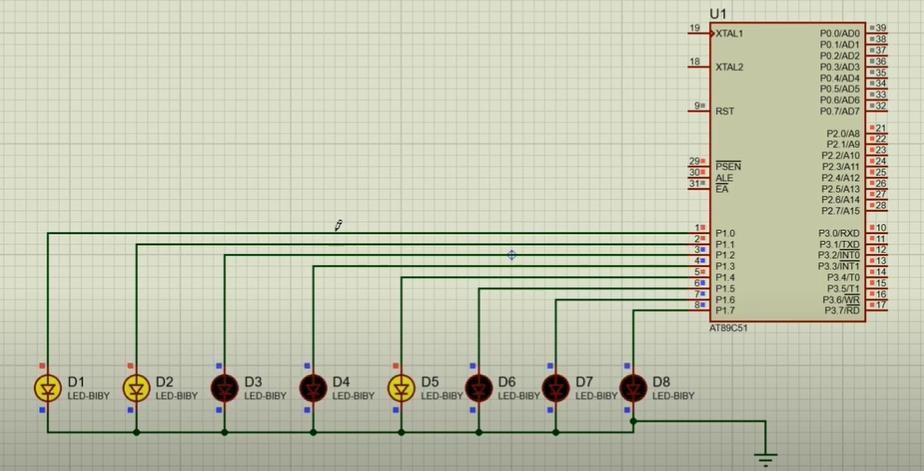
|  |  |
| --- | --- |
| **3. A** | Port I / O: Use one of the four ports of 8051 for O/P interfaced to eight LED’s. Simulate binary counter (8 bit) on LED’s |

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++) |

|  |
| --- |
| { |
| } |
| } |
| } |

Output:

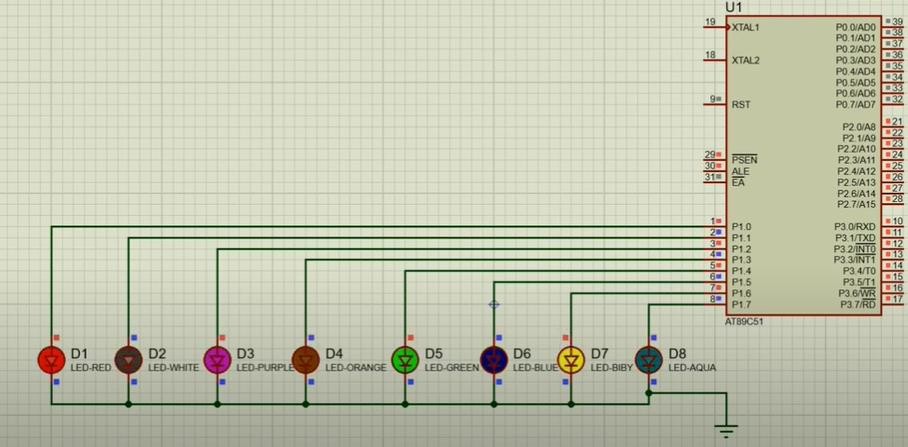


3b} Code:

|  |  |
| --- | --- |
| **B** | To interface 8 LEDs at Input-output port and create different patterns. |

|  |
| --- |
| #include<reg51.h> |
| void delay(); |
| void main() |
| { |
| while(1) |
| { |
| P1 = 0xAA; // Pattern to turn on alternate LEDs -  10101010 |
| delay(); |
| P1 = 0x55; // Reverse the pattern |
| delay(); |
| } |
| } |
| void delay() |
| { |
| unsigned int i, j; |
| for(i = 0; i < 23; i++) |
| { |

|  |
| --- |
| for(j = 0; j < 1000; j++) |
| { |
| } |
| } |
| } |

Output:

3c

|  |  |
| --- | --- |
| **3C** | To demonstrate timer working in timer mode and blink LED without using any loop delay routine. |

Code:

|  |
| --- |
| #include<reg51.h> |
| int i = 0; |
| void timer\_ISR(void)interrupt 1 |
| { |
| i++; |
| if(i == 10) |
| { |
| i=0; |
| P1++; |
| } |
| } |
| void main(void) |
| { |
| TMOD = 0x01; |
| ET0 = 1; |
| TR0 = 1; |
| EA = 1; |
| while(1); |
| }  Output |

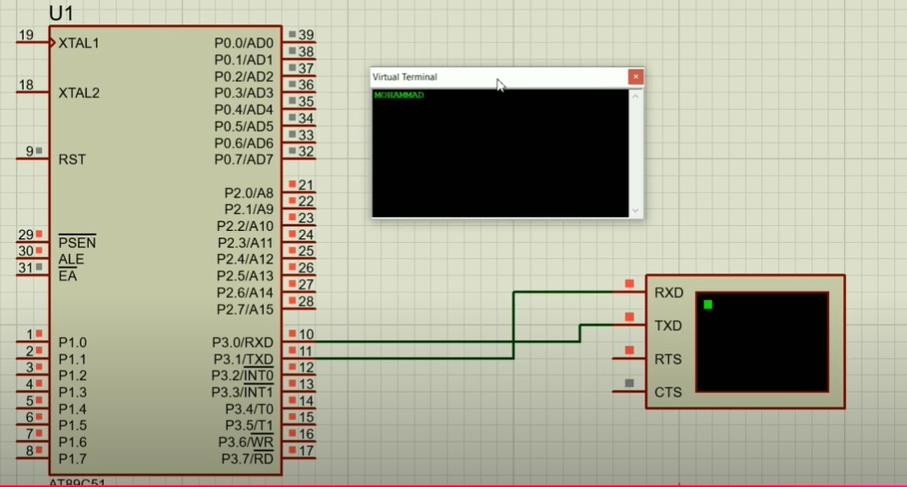
|  |  |
| --- | --- |
| **4. A** | Serial I / O: Configure 8051 serial port for asynchronous serial communication with serial port of PC exchange text messages to PC and display on PC screen. Signify end of message by carriage return. |

Code:

|  |
| --- |
| #include<reg51.h> |
| void send(char x); |
| void main(void) |
| { |
| TMOD = 0x20; |
| TH1 = 0xFD; |
| SCON = 0x50; |
| TR1 = 1; |
| send('M'); |
| send('O'); |
| send('H'); |
| send('A'); |
| send('M'); |
| send('M'); |
| send('A'); |
| send('D'); |
| while(1); |
| } |
|  |
| void send(char x) |
| { |
| SBUF = x; |
| while(TI == 0); |

|  |
| --- |
| TI = 0; |
| } |

Output: FOR BLACK SCREEN , ON LEFT SIDE FIND VIRTUAL INSTRUMENT MODE IN THAT VIRTUAL TERMINAL

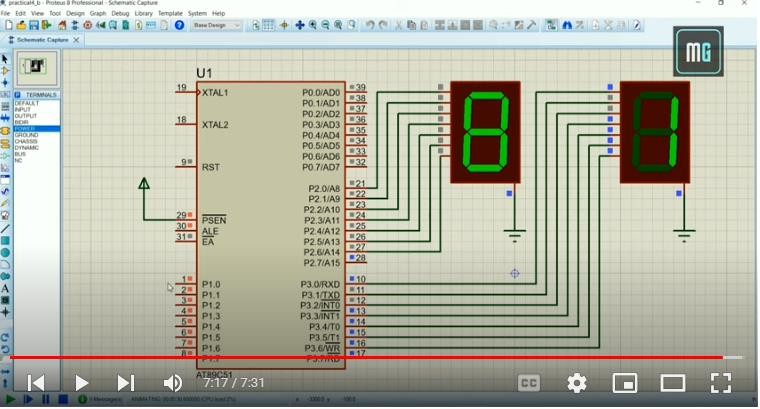


|  |  |
| --- | --- |
| **4B** | To demonstrate interfacing of seven-segment LED display and generate counting from 0 to 99 with fixed time delay. |

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); |
| } |
| } |
| } |
| } |

Output: FOR SCREEN, IN P TYPE 7 SEG-COM-CAT-GRN

|  |  |
| --- | --- |
| **4C** | Interface 8051 with D/A converter and generate square wave of given frequency on oscilloscope. |

Code:

|  |
| --- |
| #include<reg51.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++) |
| { |
| } |
| } |

Output: step: code-run-debug-start/stop debug session-analysis window-setup-new insert(RIGHT side)-type P2-select P2-close- run(LEFT side 2nd option) then for square box zoom out till you get square box

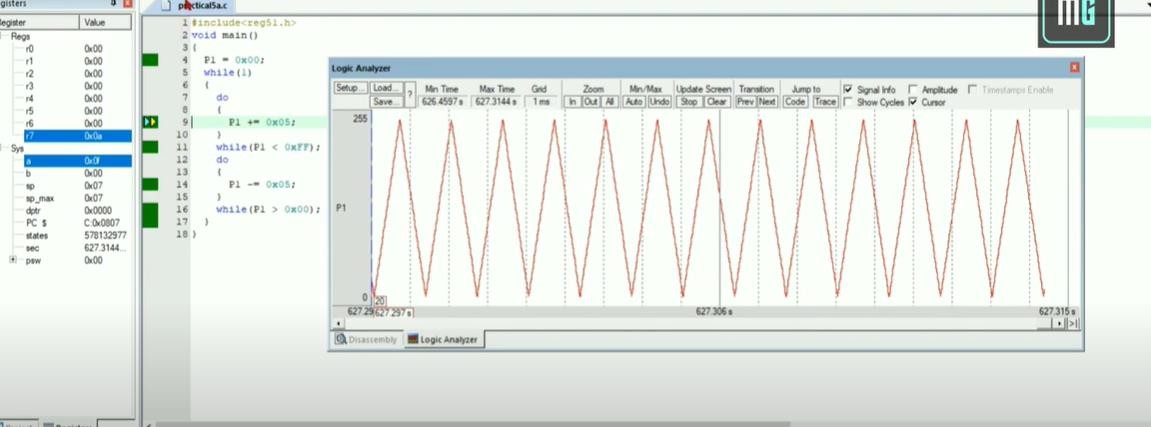
|  |  |
| --- | --- |
| **5. A** | Interface 8051 with D/A converter and generate triangular wave of given frequency on oscilloscope. |

Code:

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

|  |
| --- |
| } |
| while(P2 < 0xFF); |
| do |
| { |
| P2 -= 0x05; |
| } |
| while(P2 > 0x00); |
| } |
| } |

Output:

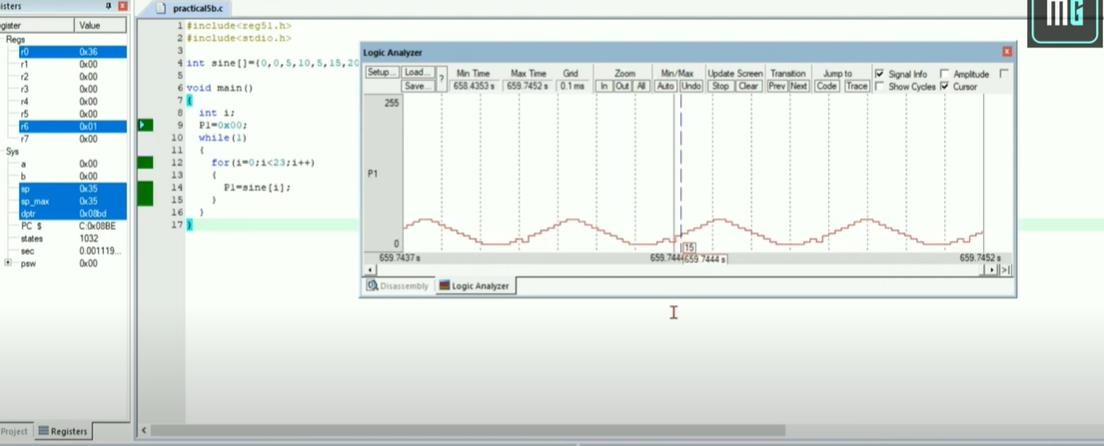


|  |  |
| --- | --- |
| **5B]** | Using D/A converter generate sine wave on oscilloscope with the help of lookup table stored in data area of 8051. |

Code:

|  |
| --- |
| #include<reg51.h> |
| #include<stdio.h> |
| int sine[]={0,0,5,10,5,15,20,25,30,35,40,45,45,40,35,30,25,20,15,10,5,0,0}; |
| void main() |
| { |
| int i; |
| P1=0x00; |
| while(1) |
| { |

|  |
| --- |
| for(i=0;i<23;i++) |
| { |
| P1=sine[i]; |
| } |
| } |
| } |

Output:

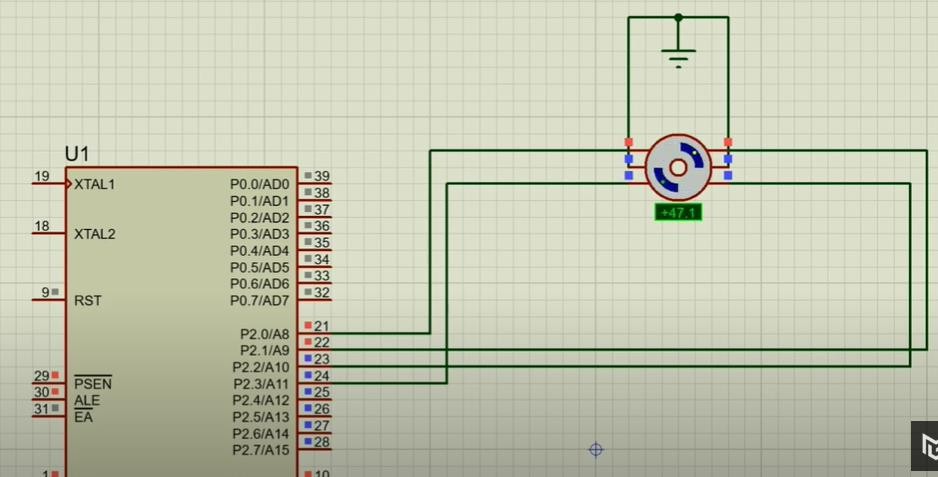
|  |  |
| --- | --- |
| **6.** | Interface stepper motor with 8051 and write a program to move the motor through a given angle in clock wise or counter clock wise direction. |

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(); |
| } |
| } |

Output: MOTOR STEPPER



|  |  |
| --- | --- |
| **7.** | Generate traffic signal. |

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++) |
| { |
| } |
| } |
| } |

Output:

