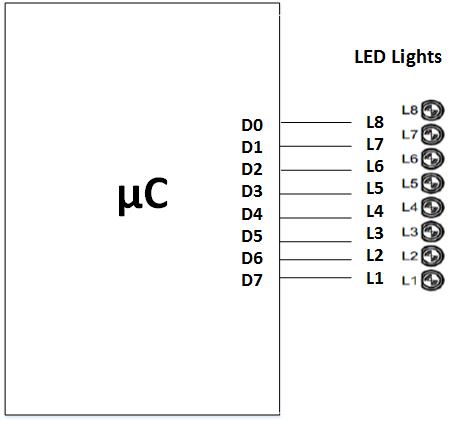
**A. Sending data to a port**

Eight numbers of LEDs are connected to 8 pins of Port D. Write a code to illuminate the LEDs and after one second to make them OFF. Then repeat the process.

For the code, the component connection should be as shown in the figure below.



Make the necessary connections according to the diagram. Connect the AVR kit with PC using the communication cable.

The code is developed in CVAVR. In this software, select ATmega32 as the chip with 16MHz frequency. Then select all pins of PORTD as the output pins and generate the code.

The code should be as follows:

#include <mega32.h>

#include <delay.h>

void main(void)

{

DDRD=0xFF;

PORTD=0xFF;

while (1)

{

// Place your code here

delay\_ms(1000);

PORTD=~PORTD;

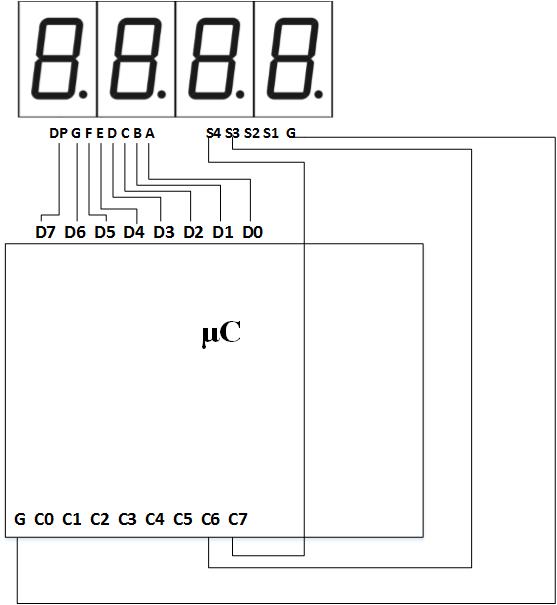
}

}

After writing the code, build the project. Then run Extreme Burner software. Select Atmega32 as the chip, open the hex file, select write all and see the effect.

**B. Multiple 7 segment display**

We have to display two-digit numbers in multi-digit 7 segment display.



Make the necessary connections according to the diagram. Connect the AVR kit with PC using the communication cable.

The code is developed in CVAVR. In this software, select ATmega32 as the chip with 16MHz frequency. Then select all pins of PORTD, sixth and seventh pins of PORTC as the output pins and generate the code.

The code should be as follows:

#include <mega32.h>

#include <delay.h>

#define data\_port PORTD

#define control\_port PORTC

char digit\_cathode[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

char x,y,i=0,j;

void main(void)

{

DDRD=0xFF;

DDRC=0xC0;

data\_port=0;

control\_port=0;

while (1)

{

// Place your code here

x=i%10;

y=i/10;

for(j=0;j<20;j++)

{

control\_port=0x40;

data\_port=digit\_cathode[x];

delay\_ms(20);

control\_port=0x80;

data\_port=digit\_cathode[y];

delay\_ms(20);

}

i++;

if (i>99) i=0;

}

}

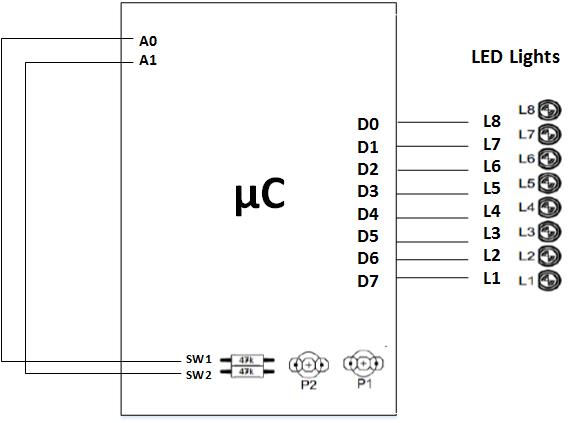
After writing the code, build the project. Then run Extreme Burner software. Select Atmega32 as the chip, open the hex file, select write all and see the effect.

**C. Input of digital data**

Let us say two push button switches P1 & P2 are connected at PortA.0 and PortA.1. And 8 LEDs are connected at Port D.

When switch 1 is pressed, glowing of one of the LEDs will shift right with 0.2 second interval and if switch 2 is pressed glowing of LED will shift left with 0.2 second interval.

The connection diagram is as follows:



Make the necessary connections according to the diagram. Connect the AVR kit with PC using the communication cable.

The code is developed in CVAVR. In this software, select ATmega32 as the chip with 16MHz frequency. Then select all pins of PORTD as the output pins, 0th and 1st pin of PORTA as the input pins and generate the code.

The code should be as follows:

#include <mega32.h>

#include <delay.h>

void main(void)

{

char move=0;

DDRD=0xFF;

PORTD=0x80;

DDRA=0x03;

PORTA=0;

while (1)

{

// Place your code here

if (PINA.0==0) move=1;

if (PINA.1==0) move=2;

if (move==1)

{

PORTD=PORTD>>1;

if (PORTD==0)

PORTD=0x80;

}

if (move==2)

{

PORTD=PORTD<<1;

if (PORTD==0)

PORTD=0x01;

}

delay\_ms(200);

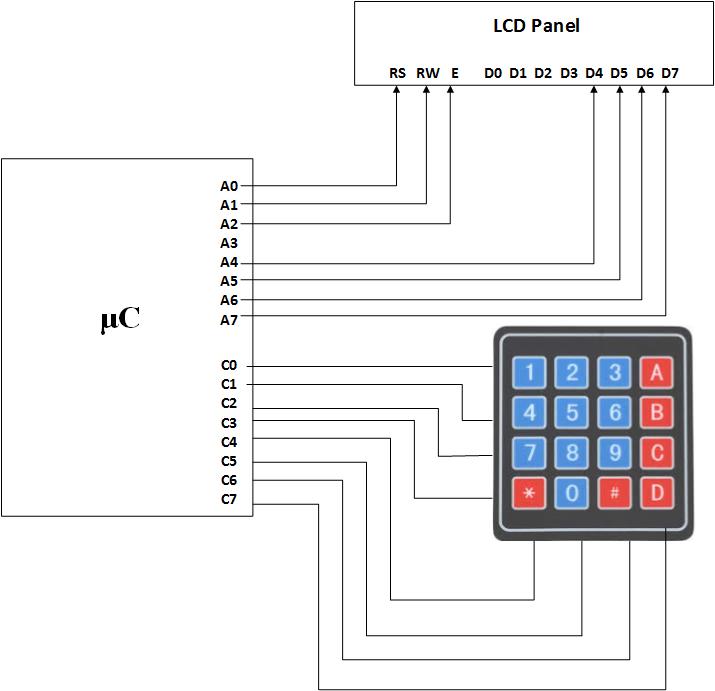
}

}

After writing the code, build the project. Then run Extreme Burner software. Select Atmega32 as the chip, open the hex file, select write all and see the effect.

**E. Detecting a key pressed in a key pad**

Let us say a 4X4 key pad is connected in Port C and a 2-line LCD is connected in Port A. If a key is pressed, it will be detected and displayed on the LCD.

The connection diagram may be that shown in the figure below:

Make the necessary connections according to the diagram. Connect the AVR kit with PC using the communication cable.

The code is developed in CVAVR. In this software, select ATmega32 as the chip with 16MHz frequency. Then select all pins of PORTA as output pins, first four pins of PORTC as the input pins, and last four pins of PORTC as the output pins and generate the code.

The code should be as follows:

#include <mega32.h>

#include <delay.h>

#include <alcd.h>

#define keypad\_ddr DDRC

#define keypad\_port PORTC

#define input\_data PINC

void main(void)

{

keypad\_ddr=0xF0;

lcd\_init(16);

while (1)

{

// Place your code here

keypad\_port=0b11101111;

lcd\_gotoxy(0,0);

if (input\_data.0==0)

lcd\_putsf("1");

if (input\_data.1==0)

lcd\_putsf("4");

if (input\_data.2==0)

lcd\_putsf("7");

if (input\_data.3==0)

lcd\_putsf("\*");

keypad\_port=0b11011111;

lcd\_gotoxy(0,0);

if (input\_data.0==0)

lcd\_putsf("2");

if (input\_data.1==0)

lcd\_putsf("5");

if (input\_data.2==0)

lcd\_putsf("8");

if (input\_data.3==0)

lcd\_putsf("0");

keypad\_port=0b10111111;

lcd\_gotoxy(0,0);

if (input\_data.0==0)

lcd\_putsf("3");

if (input\_data.1==0)

lcd\_putsf("6");

if (input\_data.2==0)

lcd\_putsf("9");

if (input\_data.3==0)

lcd\_putsf("#");

keypad\_port=0b01111111;

lcd\_gotoxy(0,0);

if (input\_data.0==0)

lcd\_putsf("A");

if (input\_data.1==0)

lcd\_putsf("B");

if (input\_data.2==0)

lcd\_putsf("C");

if (input\_data.3==0)

lcd\_putsf("D");

}

}

After writing the code, build the project. Then run Extreme Burner software. Select Atmega32 as the chip, open the hex file, select write all and see the effect.