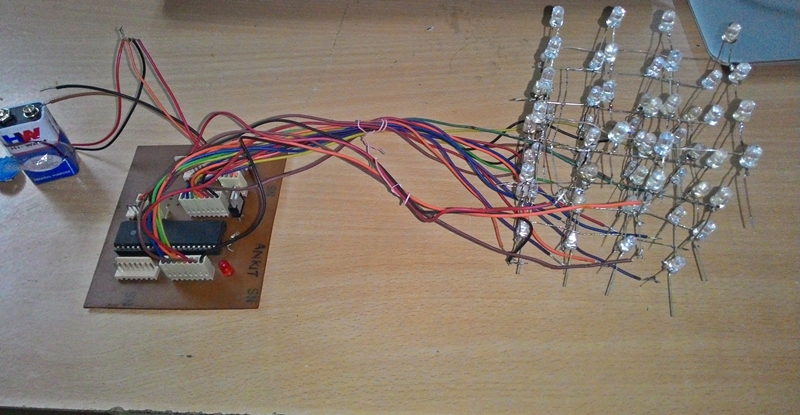
**3D LED CUBE (4x4x4)**

The basic aim, to create a 3D led cube which is capable of showing pre-defined animations.

A Typical 3D LED Display is a collection of LEDs, somehow connected and arranged in a 3D pattern and controlled so that the LEDs can be made to turn on and off in a controlled manner, thus creating interesting and pleasant patterns of light.

Probably the most common type of 3D LED display is the LED cube. The 3D matrix is constructed either by soldering the legs of the LEDs directly together or by constructing a matrix of wires which form a frame that the LEDs are connected to. There are also many different ways to control a 3D LED display which vary in complexity, the simplest is probably a single micro-controller, with more complicated methods using special LED Driver ICs or even direct PC control.



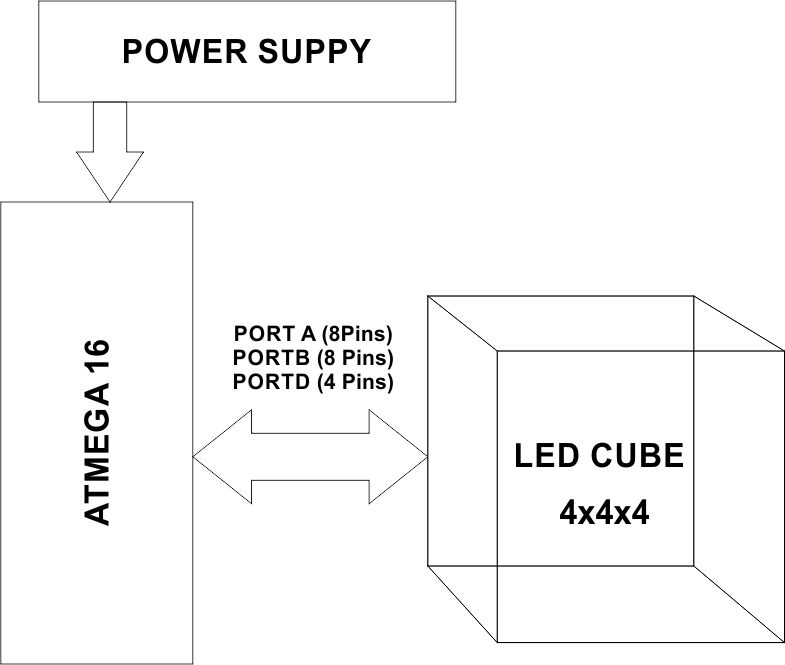
**Working:**

The LED cube has 64 LEDs (4x4x4). Each LED is dedicate an IO port in a microcontroller. LED cube rely on optical phenomenon called Persistence of Vision. If you flash a LED really fast it will remain in your retina for some time even after LED is switched off. By flashing each layer of cube very fast after one another it gives the feeling of 3d.

With this setup we will need only 16(anode) + 4(layers) IO ports to control each led.

LED has two legs one positive and one negative .The positive end is connected to the pillars of LED cube which acts as anode. The negative one is connected to the layer. Hence, to switch on a particular LED we have to give current to the corresponding pillar and ground the layer.

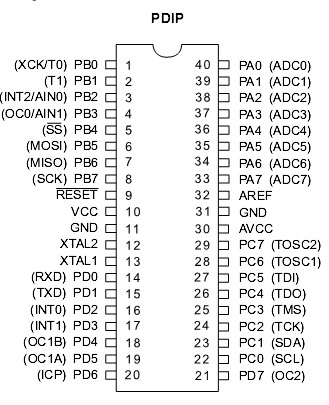
**Block Diagram:**

****

**Power Supply:**  We use 5V regulated power supply using 7805 voltage regulator and 9V battery, for our project.

**Micro Controller:**

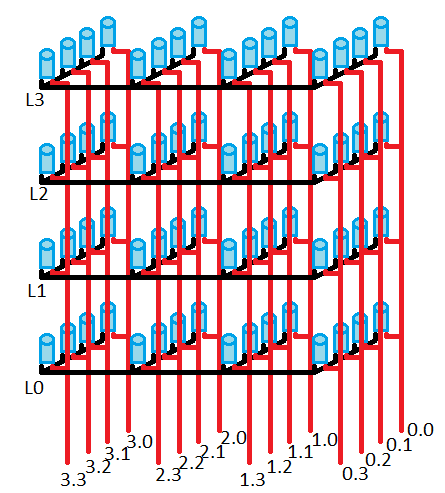
It is the heart of the project. It is used to control LED Cube by generating predefined animations. The input/ output ports of the microcontroller are used for this purpose. We Use Atmega 16 Micro controller in our Project.

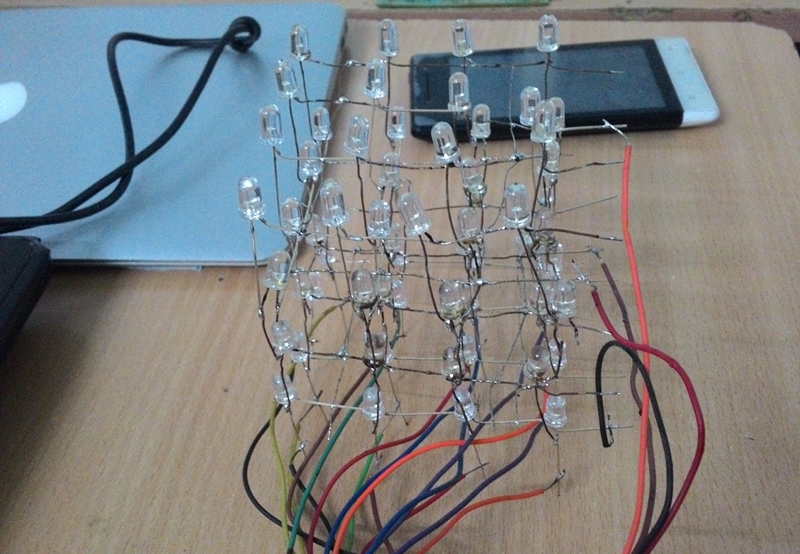
****

****

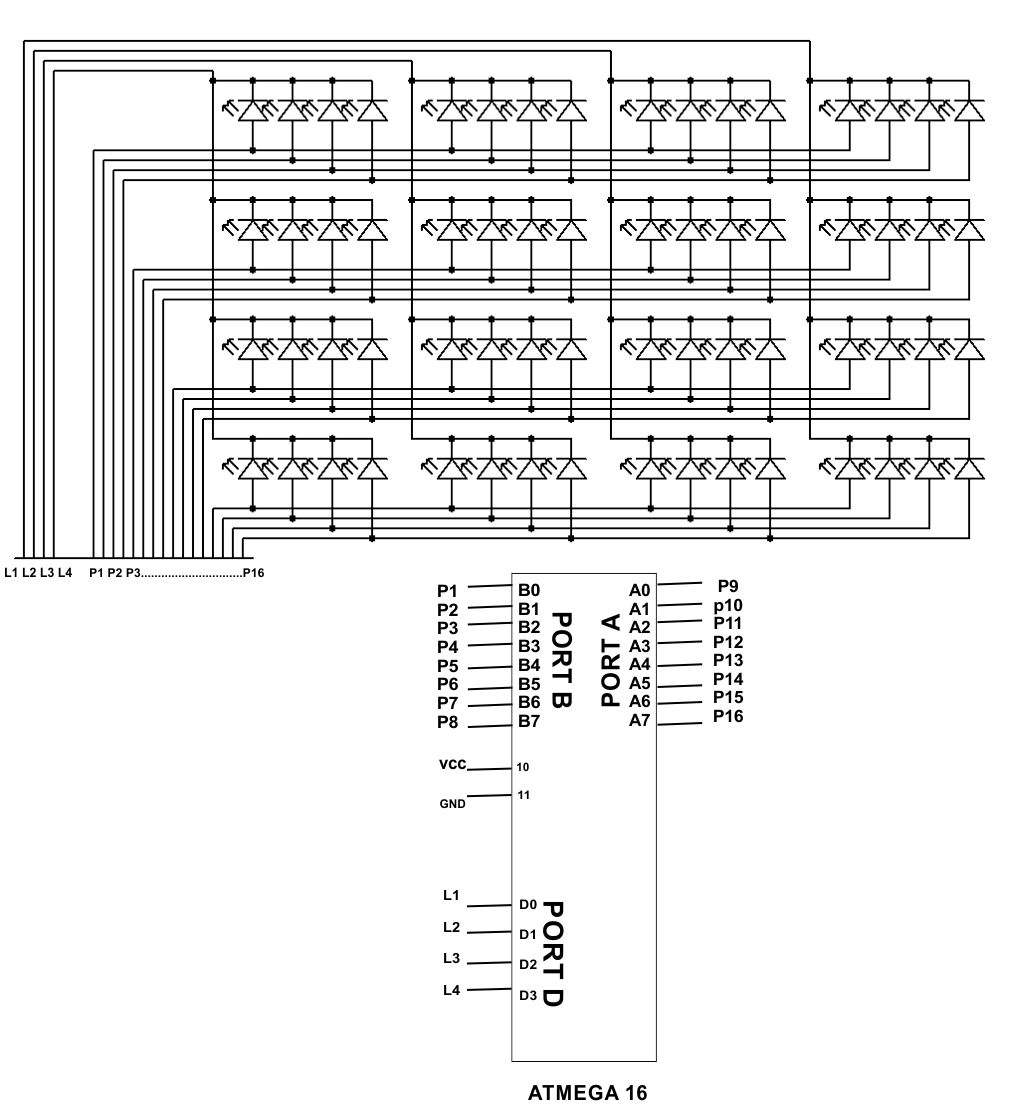
**LED Cube:**

The connection of LED cube is in a following manner. All the anodes of a layer is connected as a pillar of the cube. And cathodes are shorted around the layer. Thus we have 4 pins of cathode in a 4 layers of cube, and 16 Pins of cathode as a pillars of Cube.

****



**Circuit :**

****

**Programming :**

Programming is done under Eclipse IDE in Embedded C. The code is attached with this document.

I have Created 4 patterns of Animations in each ‘for loop’.

**#include**<avr/io.h>

**#include**<util/delay.h>

**void** **main**()

{

DDRA=0xff;

DDRB=0xff;

DDRD=0xff;

**int** i,j,k,l,m;

**while**(1)

{

**for**(i=0;i<6;i++)

{

PORTD=~1;

PORTA=0b00010001;

PORTB=0b00010001;

**\_delay\_ms**(200);

PORTD=~2;

PORTA=0b00010001;

PORTB=0b00010001;

**\_delay\_ms**(200);

PORTD=~4;

PORTA=0b00010001;

PORTB=0b00010001;

**\_delay\_ms**(200);

PORTD=~8;

PORTA=0b00010001;

PORTB=0b00010001;

**\_delay\_ms**(200);

PORTD=~4;

PORTA=0b00100010;

PORTB=0b00100010;

**\_delay\_ms**(200);

PORTD=~2;

PORTA=0b01000100;

PORTB=0b01000100;

**\_delay\_ms**(200);

PORTD=~1;

PORTA=0b10001000;

PORTB=0b10001000;

**\_delay\_ms**(200);

PORTD=~2;

PORTA=0b10001000;

PORTB=0b10001000;

**\_delay\_ms**(200);

PORTD=~4;

PORTA=0b10001000;

PORTB=0b10001000;

**\_delay\_ms**(200);

PORTD=~8;

PORTA=0b10001000;

PORTB=0b10001000;

**\_delay\_ms**(200);

PORTD=~4;

PORTA=0b01000100;

PORTB=0b01000100;

**\_delay\_ms**(200);

PORTD=~2;

PORTA=0b00100010;

PORTB=0b00100010;

**\_delay\_ms**(200);

}

**for**(j=0;j<6;j++)

{

PORTD=~8;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_ms**(200);

PORTD=~4;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_ms**(200);

PORTD=~2;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_ms**(200);

PORTD=~1;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_ms**(200);

}

**for**(k=0;k<6;k++)

{

PORTD=~1;

PORTA=~0b10011111;

PORTB=~0b11111001;

**\_delay\_ms**(200);

PORTD=~2;

PORTA=~0b10011111;

PORTB=~0b11111001;

**\_delay\_ms**(200);

PORTD=~4;

PORTA=~0b10011111;

PORTB=~0b11111001;

**\_delay\_ms**(200);

PORTD=~8;

PORTA=~0b10011111;

PORTB=~0b11111001;

**\_delay\_ms**(200);

PORTD=~8;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_ms**(200);

PORTD=~4;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_ms**(200);

PORTD=~2;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_ms**(200);

PORTD=~1;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_ms**(200);

}

**for**(l=0;l<6;l++)

{

PORTD=~1;

PORTA=1;**\_delay\_ms**(100);

PORTD=~2;

PORTA=0b00100000;**\_delay\_ms**(100);PORTA=0;

PORTD=~4;

PORTB=0b00000100;**\_delay\_ms**(100);

PORTD=~8;

PORTB=0b10000000;**\_delay\_ms**(100);

PORTD=~4;

PORTB=0b00001000;**\_delay\_ms**(100);PORTB=0;

PORTD=~2;

PORTA=0b10000000;**\_delay\_ms**(100);

PORTD=~1;

PORTA=0b00001000;**\_delay\_ms**(100);

PORTD=~2;

PORTA=0b01000000;**\_delay\_ms**(100);PORTA=0;

PORTD=~4;

PORTB=0b00000010;**\_delay\_ms**(100);

PORTD=~8;

PORTB=0b00010000;**\_delay\_ms**(100);

PORTD=~4;

PORTB=0b00000001;**\_delay\_ms**(100);PORTB=0;

PORTD=~2;

PORTA=0b00010000;**\_delay\_ms**(100);

}

**for**(m=0;m<10000;m++)

{

PORTD=~1;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_us**(100);

PORTD=~2;

PORTA=0b00001001;

PORTB=0b10010000;

**\_delay\_us**(100);

PORTD=~4;

PORTA=0b00001001;

PORTB=0b10010000;

**\_delay\_us**(100);

PORTD=~8;

PORTA=0b10011111;

PORTB=0b11111001;

**\_delay\_us**(100);

}

}

}