Multiplexing Based Displaying of Data in a 3D Matrix of LEDs

By

Ayush Goyal

Enrolment No.: 16121007

Electrical Engineering Department

Indian Institute of Technology, Roorkee

January, 2018

Objective

Under the project, it is aimed to make a working model for Multiplexing Based Displaying of Data in a 3D Matrix of LEDs. An 8x8x8 3D matrix of LEDs is planned to be used for the purpose.

Introduction to Multiplexing

In telecommunications and computer networks, multiplexing is a method by which multiple analog or digital signals are combined into one signal over a shared medium. The aim is to share an expensive resource. For example, in telecommunications, several telephone calls may be carried using one wire. Multiplexing originated in telegraphy in the 1870s, and is now widely applied in communications. In telephony, George Owen Squier is credited with the development of telephone carrier multiplexing in 1910.

The multiplexed signal is transmitted over a communication channel such as a cable. The multiplexing divides the capacity of the communication channel into several logical channels, one for each message signal or data stream to be transferred. A reverse process, known as de-multiplexing, extracts the original channels on the receiver end.

A device that performs the multiplexing is called a multiplexer (MUX), and a device that performs the reverse process is called a de-multiplexer (DEMUX or DMX).

Inverse multiplexing (IMUX) has the opposite aim as multiplexing, namely to break one data stream into several streams, transfer them simultaneously over several communication channels, and recreate the original data stream.

Materials used

For making the project, the following materials have been used

Sl. No.	Material	Quantity
1	Transistors : MMBT904	72 Nos.
2	Resistors: 100Ohm 1206SMD	136 Nos.
3	Resistors: 1000Ohm 1206SMD	144 Nos.
4	Capacitor: 0.1uF	9 Nos.
5	Integrated Circuit-Shift Register: 74HC95D	9 Nos.
6	MOSFETs: IRF9Z34N	8 Nos.
7	LEDs	512 Nos.

8	Copper clad boards	2 Nos.
9	Tin wire (for structural support)	3m
10	Solder wire	80g
11	Flux	50ml

Methodology

The making of the Circuit

First of all circuit layout for the two set of PCBs(printed circuit boards) were made from the schematic diagrams. One of the set would act as a base for the cube and will also contain the anode power supply circuit and would work in conjunction with the second set of boards that would control the 64 cathodes and therefor enable multiplexing.

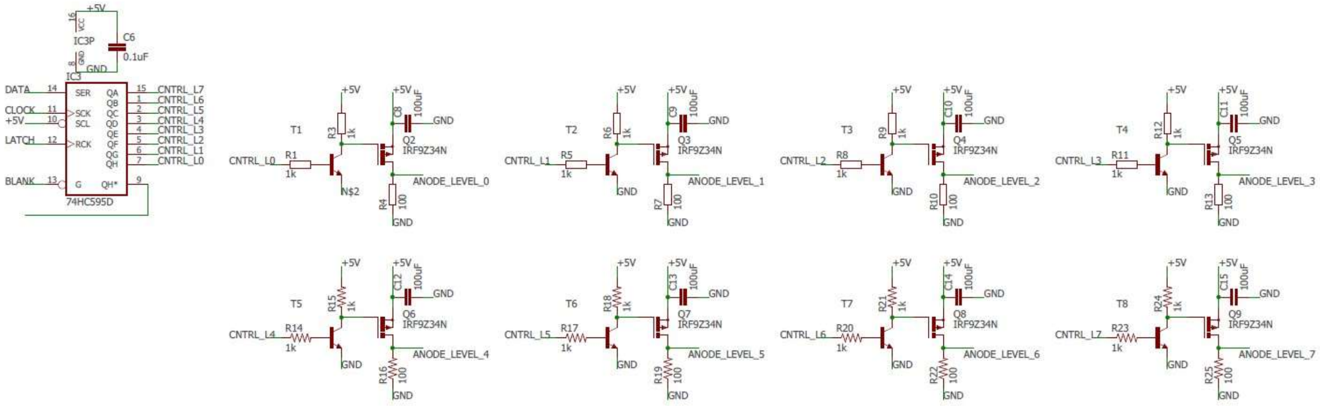
+5V
16 VCC
IC3P
C6
0.1uF
8 GND
GND
IC3
DATA 14 SER
CLOCK 11 SCK
+5V 10 SCL
LATCH 12 RCK
BLANK 13 G
QA 15 CNTRL_L7
QB 1 CNTRL_L6
QC 2 CNTRL_L5
QD 3 CNTRL_L4
QE 4 CNTRL_L3
QF 5 CNTRL_L2
QG 6 CNTRL_L1
QH 7 CNTRL_L0
QH* 9
74HC595D
T1 R1 1k R3 1k CNTRL_L0 N$2 C8 100uF GND Q2 IRF9Z34N ANODE_LEVEL_0 R4 100 GND
T2 R5 1k R6 1k CNTRL_L1 GND C9 100uF Q3 IRF9Z34N ANODE_LEVEL_1 R7 100
T3 R8 1k R9 1k CNTRL_L2 GND C10 100uF Q4 IRF9Z34N ANODE_LEVEL_2 R10 100
T4 R11 1k R12 1k CNTRL_L3 GND C11 100uF Q5 IRF9Z34N ANODE_LEVEL_3 R13 100
T5 R14 1k R15 1k CNTRL_L4 GND C12 100uF Q6 IRF9Z34N ANODE_LEVEL_4 R16 100
T6 R17 1k R18 1k CNTRL_L5 GND C13 100uF Q7 IRF9Z34N ANODE_LEVEL_5 R19 100
T7 R20 1k R21 1k CNTRL_L6 GND C14 100uF Q8 IRF9Z34N ANODE_LEVEL_6 R22 100
T8 R23 1k R24 1k CNTRL_L7 GND C15 100uF Q9 IRF9Z34N ANODE_LEVEL_7 R25 100
+5V
GND

Figure 1: Anode control part of the schematics

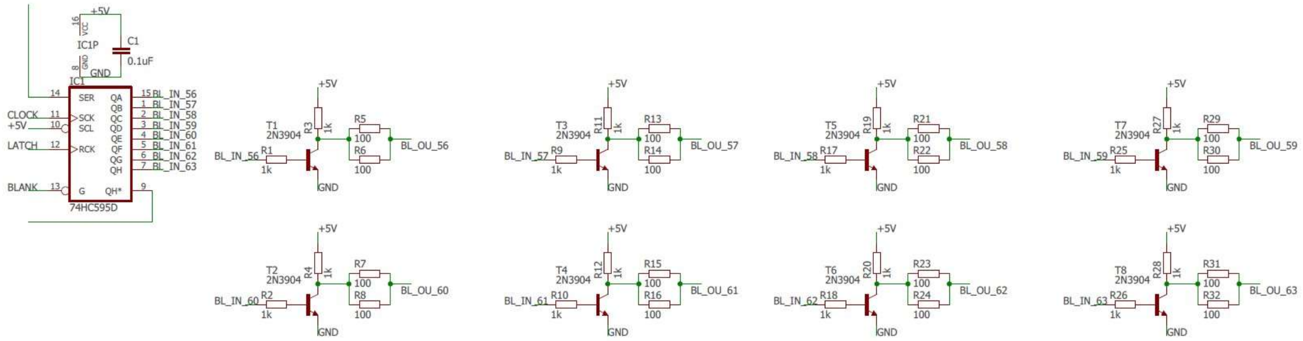
+5V
16 VCC
IC1P
C1
0.1uF
8 GND
GND
IC1
14 SER
CLOCK 11 SCK
+5V 10 SCL
LATCH 12 RCK
BLANK 13 G
QA 15 BL_IN_56
QB 1 BL_IN_57
QC 2 BL_IN_58
QD 3 BL_IN_59
QE 4 BL_IN_60
QF 5 BL_IN_61
QG 6 BL_IN_62
QH 7 BL_IN_63
QH* 9
74HC595D
T1 2N3904 R3 1k BL_IN_56 R1 1k R5 100 R6 100 BL_OU_56 GND
T3 2N3904 R11 1k BL_IN_57 R9 1k R13 100 R14 100 BL_OU_57 GND
T5 2N3904 R19 1k BL_IN_58 R17 1k R21 100 R22 100 BL_OU_58 GND
T7 2N3904 R27 1k BL_IN_59 R25 1k R29 100 R30 100 BL_OU_59 GND
T2 2N3904 R4 1k BL_IN_60 R2 1k R7 100 R8 100 BL_OU_60 GND
T4 2N3904 R12 1k BL_IN_61 R10 1k R15 100 R16 100 BL_OU_61 GND
T6 2N3904 R20 1k BL_IN_62 R18 1k R23 100 R24 100 BL_OU_62 GND
T8 2N3904 R28 1k BL_IN_63 R26 1k R31 100 R32 100 BL_OU_63 GND
+5V

Figure 2: Cathode control part of the schematics

The complete schematic diagram is given below in figure 3.

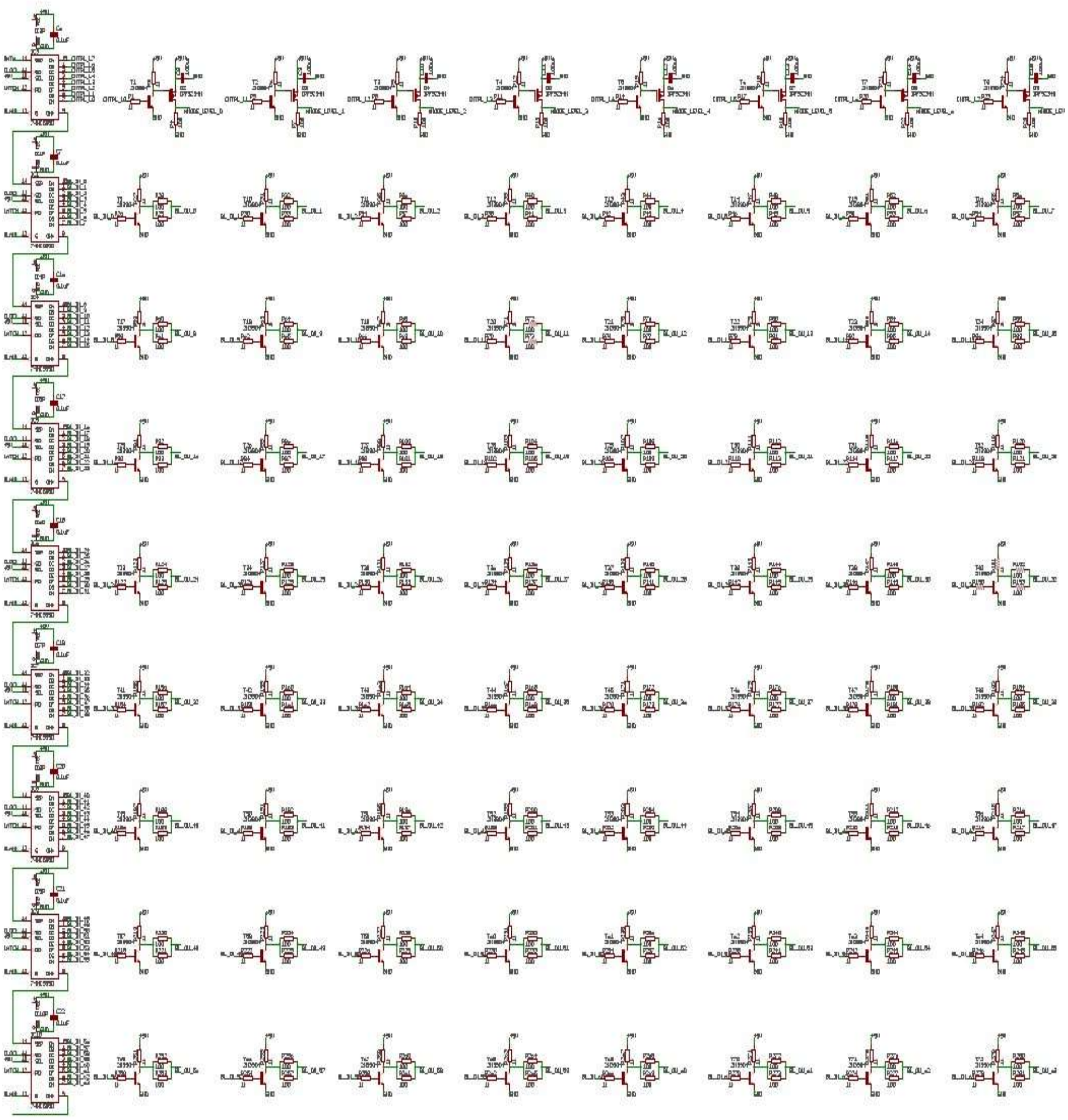

Figure 3: Complete schematics

The layouts for these two PCBs made from the schematic diagram are given below at figure 4 and 5

.

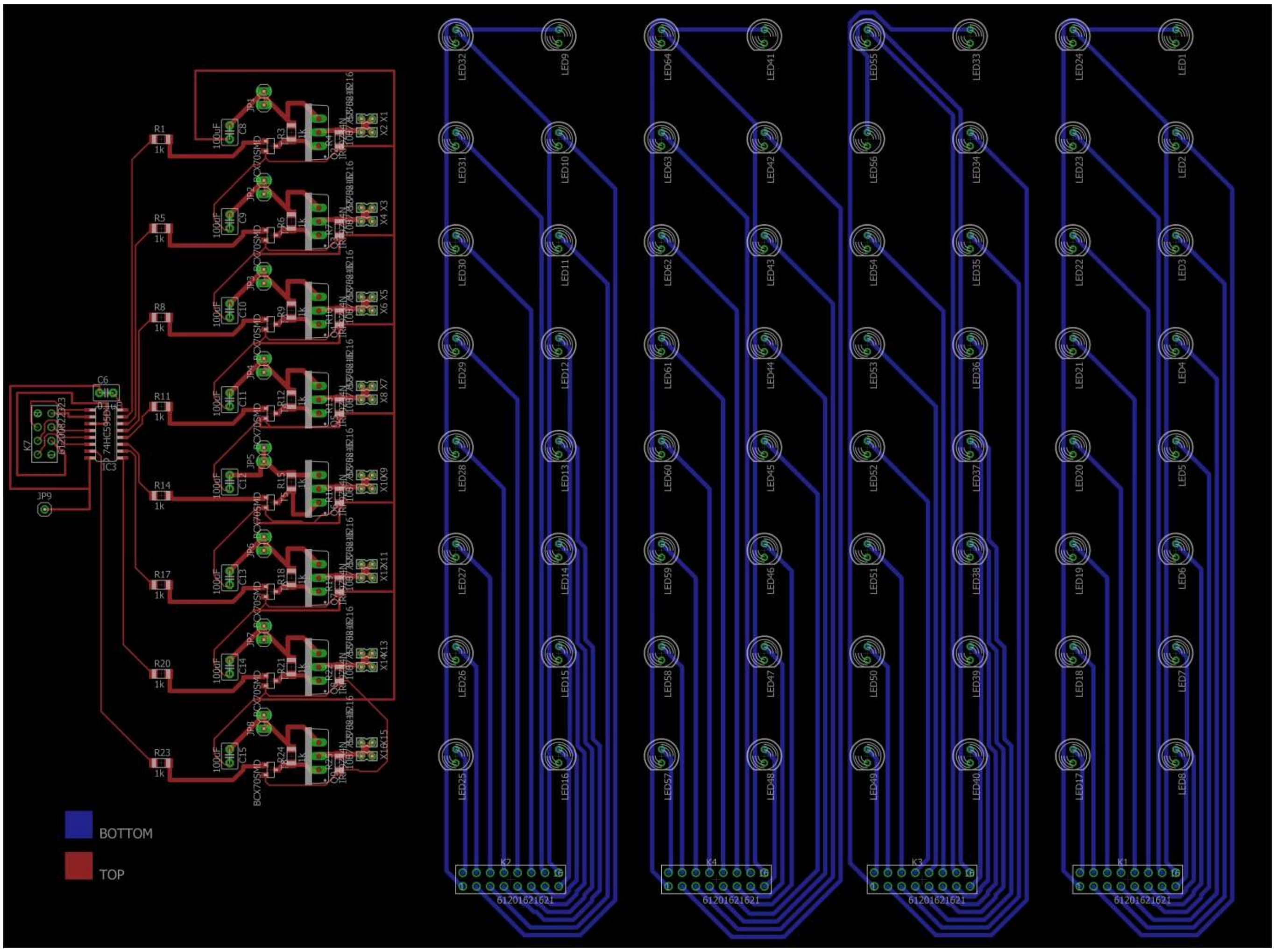

Figure 4: Final layout of the cathode control board in conjunction to the position when the cathodes of the bottom layer of the led cube

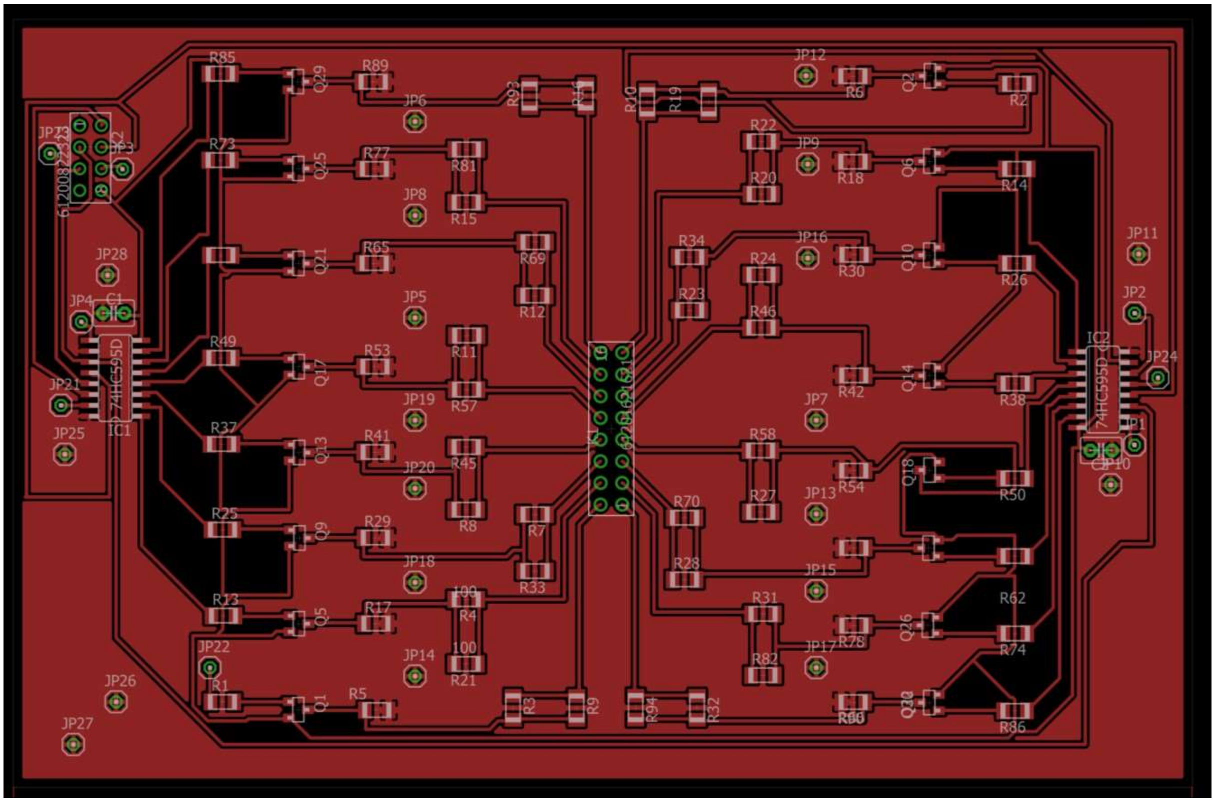

Figure 5: Cathode control board (4 of these are required)

After making the final layout of the PCBs they were transferred to the copper clad boards using screen printing technique. After drying of the ink they were put in an etching bath containing a solution of 1:1 HCl (37% v/v) and H_2O_2 (6.7% v/v). [Note: Mineral acid etching is very fast and produces a lot of fumes so should be don't in a fume hood or in the open] **After** etching, the boards were washed and dried. Thereafter, the holes were drilled on the boards and then they were cleaned by light rubbing with steel wool.

After cleaning the components were placed and soldered.

The making of the 3D LED matrix

Anodes of all the LEDs were bent at a 90° angle to fit the stencil made by drilling holes in a piece of wood. The stencil contained 8 holes so as to hold 8 LEDs, and thereby allowing to make linear array of 8 LEDs by soldering the bent anode on one LED to the anode of the next LED. 64 such linear arrays were made, each having 8 LEDs. Now 8 such arrays were arranged in a stencil made with the help of paper pins. These 8 arrays were solder by connecting their cathodes using tin wire in a vertical sense so as to make a slice, NOT A LAYER because if the linear arrays are soldered in horizontal sense, they would have 64 cathodes that would have to be connected to the next layer where as in these slices there are only eight connections that need to be made. Eight such slices of 8x8 LEDs were made and there after they were stacked upon each other using spacers that can be removed and soldered the anodes. A tin wire was used to provide structural rigidity. This way, a cube of 8x8x8 LEDs was formed to get a 3D LED matrix.

After this, the LED matrix was placed and soldered on a PCB that was fabricated with lay out diagram in figure 4. The picture of the completed LED matrix after fixing on the PCB is given in figure 6 below.

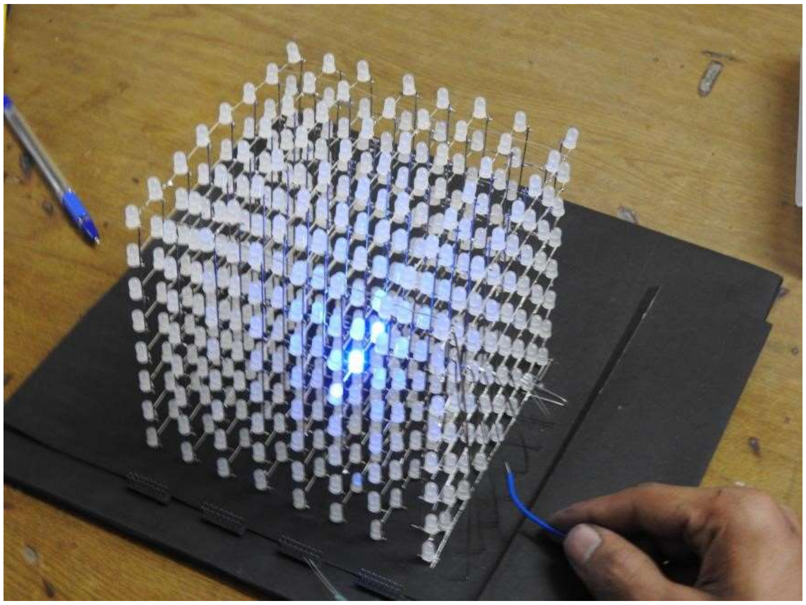

Figure 6: Completed LED matrix

The second set of PCBs connect to the PCB that holds the LED matrix via four 2x8 connectors that are place at the bottom and at the middle of the respective boards.

The second set of PCBs is a shift register bank that takes input from an Arduino via four pins namely DATA, CLOCK, LATCH and BLANK. The Arduino can be programed as required. A sample animation (without initialisation) code is given below:

```
**************CODE START**************
void rainVersionTwo(){
  int x[64], y[64], z[64], addr, leds=64, bright=1, ledcolor, colowheel;
int xx[64], yy[64], zz[64], xold[64], yold[64], zold[64], slowdown;
```

```
for(addr=0; addr<64; addr++){     x[addr]=random(8);
y[addr]=random(8);     z[addr]=random(8);     xx[addr]=random(16);
yy[addr]=random(16);
    zz[addr]=random(16);
  }
start=millis();
  while(millis()-start<20000){

if(ledcolor<200){        for(addr=0;
addr<leds; addr++){
    LED(zold[addr], xold[addr], yold[addr], 0, 0, 0);   if(z[addr]>=7)
  LED(z[addr], x[addr], y[addr], 0, 5, 15);   if(z[addr]==6)
  LED(z[addr],      x[addr],      y[addr],      0,      1,      9);
if(z[addr]==5)
  LED(z[addr],      x[addr],      y[addr],      0,      0,      10);
if(z[addr]==4)
  LED(z[addr],      x[addr],      y[addr],      1,      0,      11);
if(z[addr]==3)
  LED(z[addr],      x[addr],      y[addr],      3,      0,      12);
if(z[addr]==2)
  LED(z[addr],      x[addr],      y[addr],      10,      0,      15);
if(z[addr]==1)
  LED(z[addr],      x[addr],      y[addr],      10,      0,      10);
if(z[addr]<=0)
  LED(z[addr], x[addr], y[addr], 10, 0, 1);
}}

  if(ledcolor>=200&&ledcolor<300){
for(addr=0; addr<leds; addr++){
    LED(zold[addr], xold[addr], yold[addr], 0, 0, 0);   if(z[addr]>=7)
  LED(z[addr], x[addr], y[addr], 15, 15, 0);   if(z[addr]==6)
  LED(z[addr],      x[addr],      y[addr],      10,      10,      0);
if(z[addr]==5)
  LED(z[addr],      x[addr],      y[addr],      15,      5,      0);
if(z[addr]==4)
  LED(z[addr],      x[addr],      y[addr],      15,      2,      0);
if(z[addr]==3)
  LED(z[addr],      x[addr],      y[addr],      15,      1,      0);
if(z[addr]==2)
  LED(z[addr],      x[addr],      y[addr],      15,      0,      0);
if(z[addr]==1)
  LED(z[addr],      x[addr],      y[addr],      12,      0,      0);
if(z[addr]<=0)
LED(z[addr], x[addr], y[addr], 10, 0, 0);
}}

  if(ledcolor>=300&&ledcolor<400){

}
   if(ledcolor>=500&&ledcolor<600){

}

 ledcolor++; if(ledcolor>=300)
ledcolor=0;

  for(addr=0; addr<leds; addr++){
xold[addr]=x[addr];
yold[addr]=y[addr];
  zold[addr]=z[addr];
  }

 delay(15);

  for(addr=0; addr<leds; addr++){
z[addr] = z[addr]-1;
if(z[addr]<random(-100,0)){
x[addr]=random(8);
y[addr]=random(8);    int
select=random(3);
if(select==0){    xx[addr]=0;
zz[addr]=random(16);
  yy[addr]=random(16);
  }    if(select==1){
xx[addr]=random(16);
zz[addr]=0;
yy[addr]=random(16);
  }    if(select==2){
xx[addr]=random(16);
zz[addr]=random(16);
  yy[addr]=0;
  }
  z[addr]=7;
  }
 }
 }
}
```

***************CODE END*************** After this, the LED matrix is ready to take input from the Arduino and display data.

Result and Discussion

The 3D LED matrix combined with the PCBs is capable to displaying data that can be divided in a matrix of 8x8 (like a dot matrix display) with the addition of the element of depth. This project is completely saleable on both higher and lower sides. This also means it can incorporate more colours, i.e., it can be made from an RGB matrix as well, allowing the display of different colours. The number of colours that can be displayed would only be limited by the code.

Further applications can be holographic display allowing users to view an image from different angles.