6416/6432 Dot Matrix LED Display Information Board User's Guide

Table of contents

NOTES:

Product Version : Ver 1.0

Document Version : Ver 1.0

Chapter1.Brief Introduction

1-1. Overview

LED dot-matrix info board is an economical solution for advertising or displaying applications that require various displaying effects. It can be used in shops, restaurants, exhibition saloons, airports and railway stations. This series of dot-matrix LED info board features gentle color and high resolution and offers mono/bicolor selections.

1-2. Gallery

Sure Electronics offers 5 different LED dot-matrix info boards and users may refer to the following table for detailed specifications:

Product Number	Style	Diameter of each LED(mm)	Size	Color supported	Photo
DE-DP029	6416 bicolor	5	4.8inch*19.1inch	Red and green	Figure 1（1） Not illuminated Figure 1（2） Yellow snow
DE-DP030	6432 bicolor	5	9.6inch*19.1inch	Red and green	Figure 2（1） Not illuminated Figure 2（2） Yellow snow
DE-DP031	6416 monocolor	5	4.8inch*19.1inch	Red	Figure 3（1） Not illuminated Figure 3（2） Red koala
DE-DP032	6432 monocolor	5	9.6inch*19.1inch	Red	Figure 4（1） Not illuminated Figure 4（2） Red snow

DE-DP033	6432 bicolor	3	5inch*10inch	Red and green	Figure 5（1） Not illuminated Figure 5（2） Yellow koala

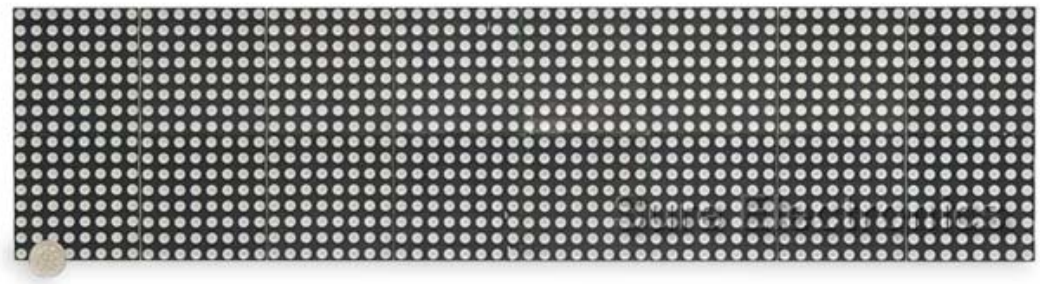


Figure 1（1）

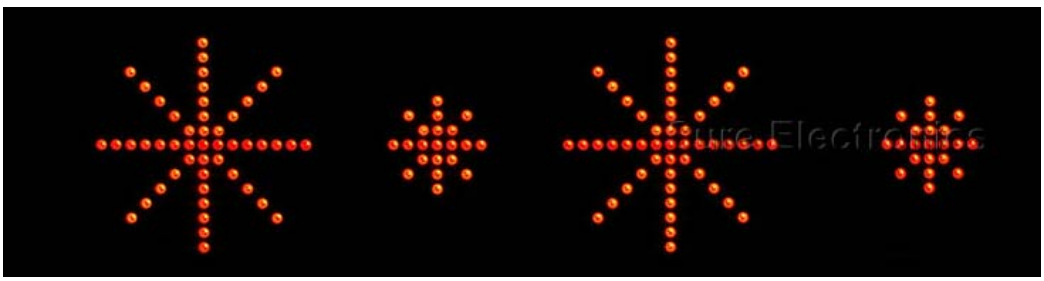


Figure 1（2）

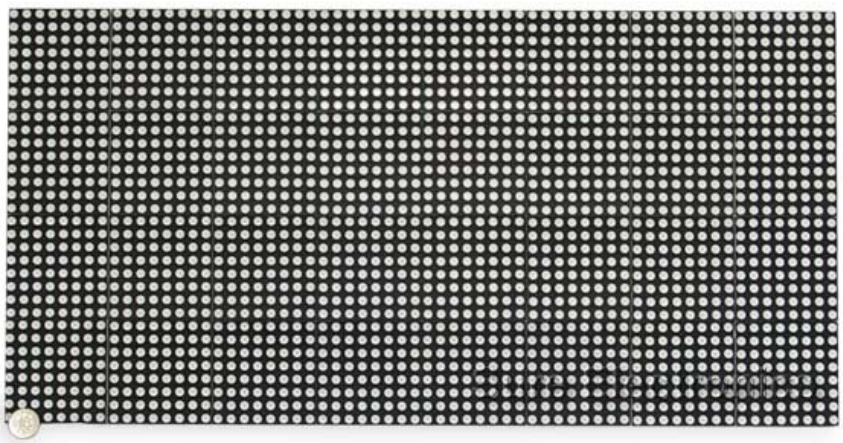


Figure 2（1）

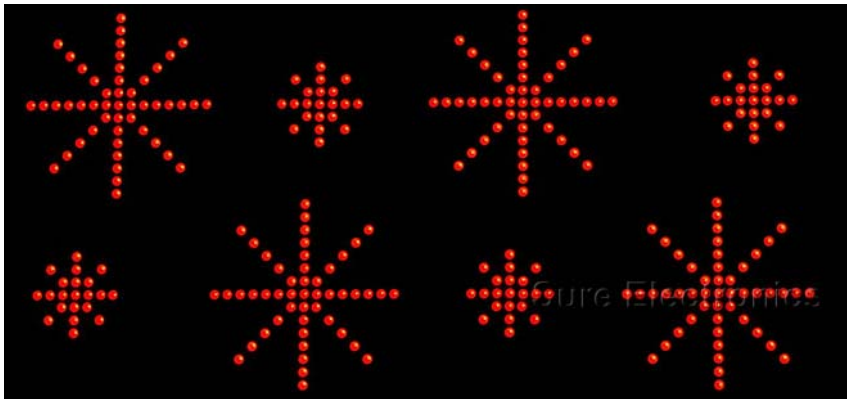


Figure 2（2）

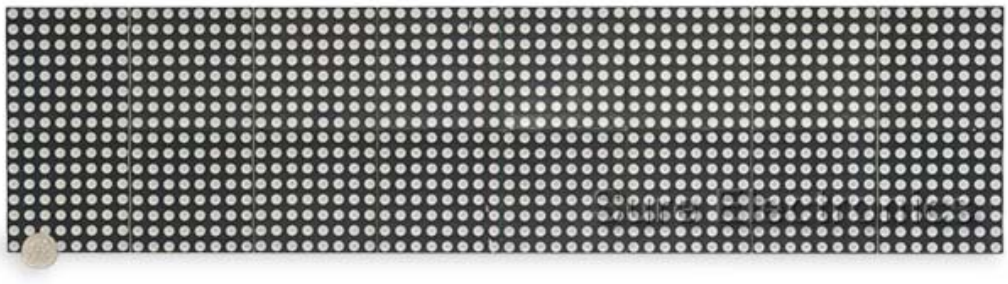


Figure 3（1）

Figure 3（2）

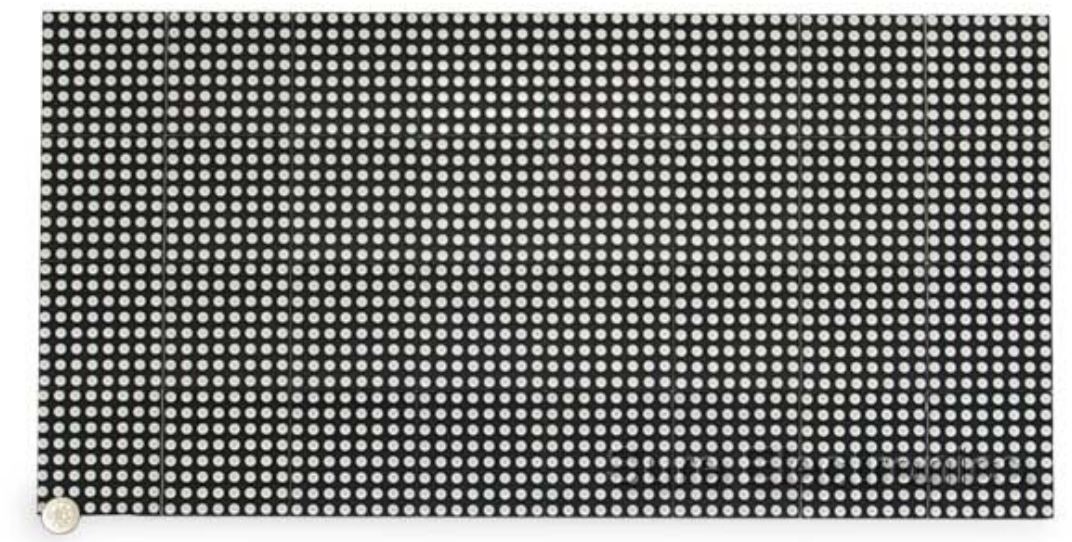


Figure 4（1）

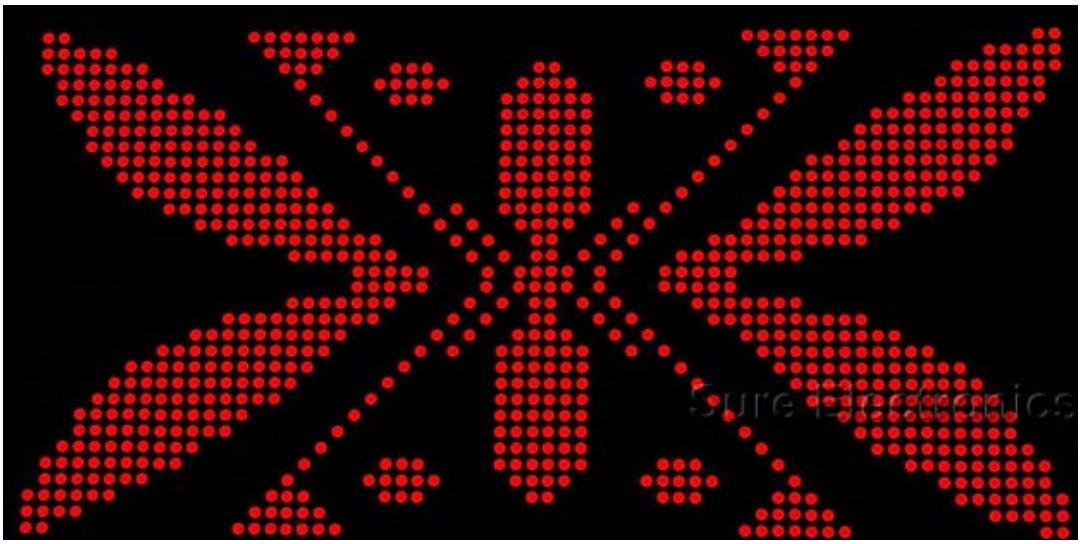


Figure 4（2）

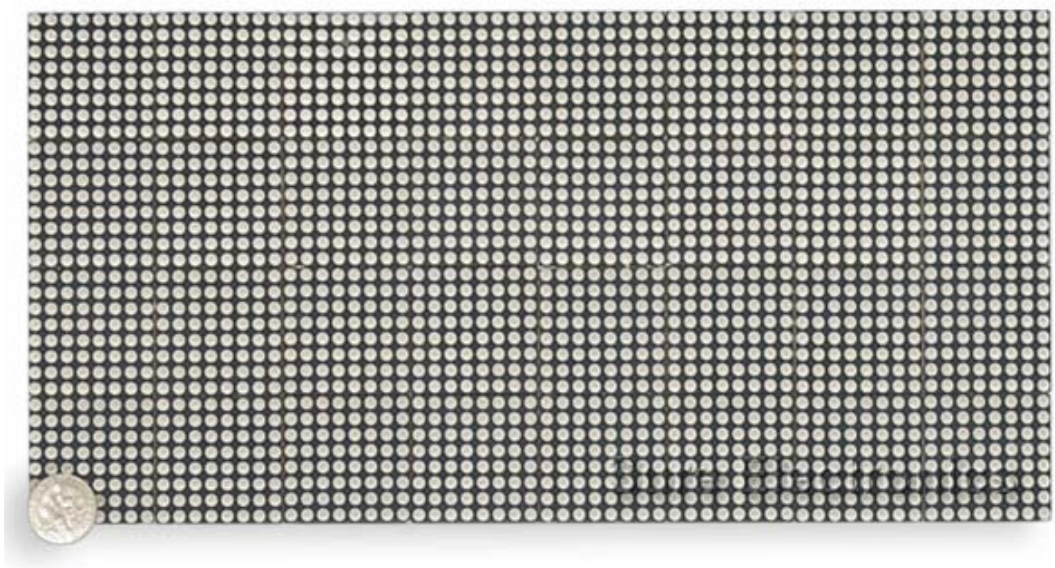


Figure 5（1）

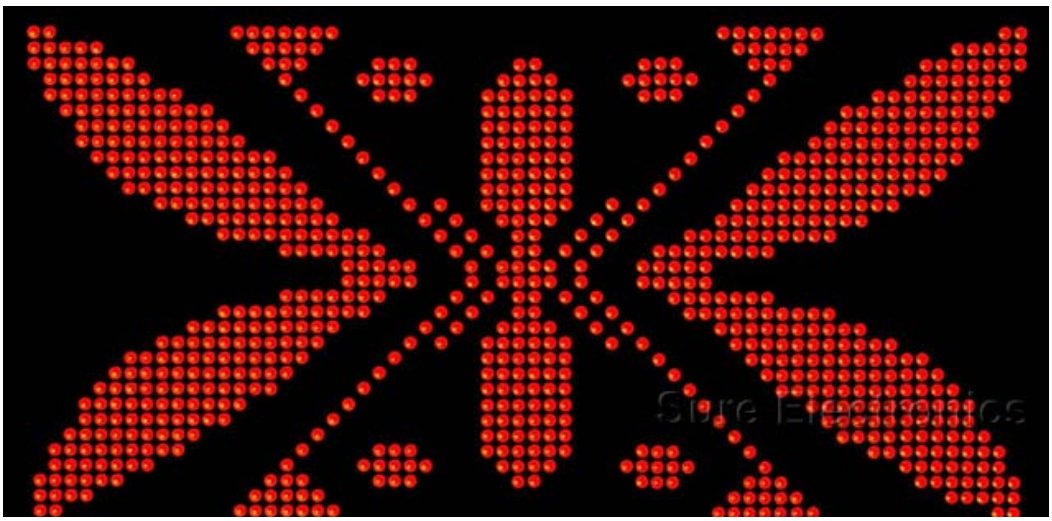


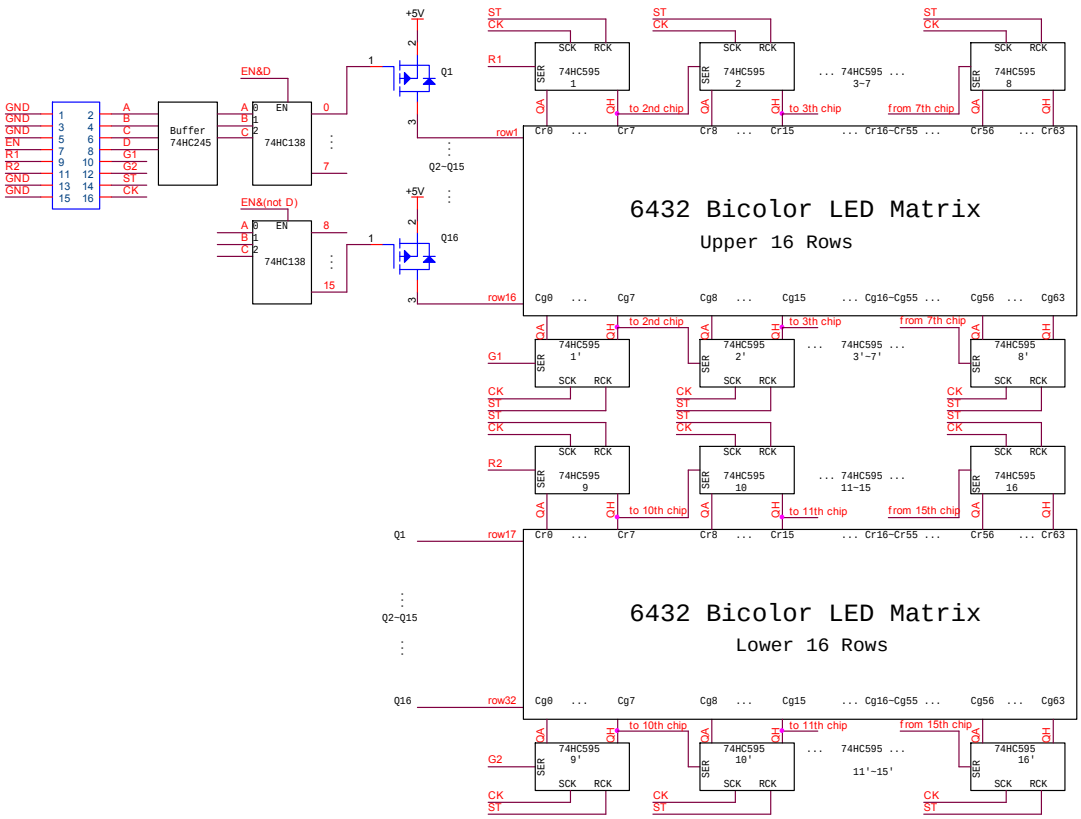
Figure 5（2）

Note：When a bicolor LED is illuminated in both red and green, LED emits yellow light.

Chapter2.Hardware Description

2-1.Schematic

The product family of dot-matrix LED display info board shares the same features. The schematic below shows what a typical 6432 bicolor LED would be. Chip 74HC138 serves as a row selector which controls MOSFET in providing positive polar for LED while chip 74HC595 is a shift register for providing negative polar for each LED. In addition, a simple method to distinguish the type of LED info board is to check the number of 74HC595 chip at the reverse side of each 8*8 dot-matrix, one suggests monocolor LED display and two suggests bicolor LED display.

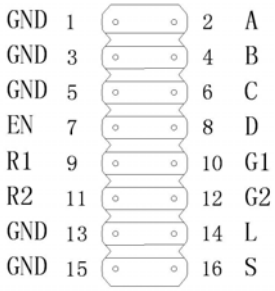


2-2. Main Features

- Programmable.
- Power supply voltage: 5V.
- Maximum current：5.26A. the first red LED rows of both halves of 6432 dot-matrix display are lit up in a dynamic scanning way. Data transfer of a single row is 92.8μs with EN pin switched on.
- Average current value: 1.64A. 16 pieces red LED of the first rows of both halves of 6432 display are lit up in a dynamic scanning way. Data transfer of a single row is 92.8μs with EN pin switched on.
- 6432/6416 dot matrix on each board.
- Gentle, comfortable and optional color: Green or Red.
- Each board contains 32 or 16 pieces 0808 LED dot-matrix modules.
- Serial MCU interface----R1/R2/G1/G2, CK,ST
- More LED boards can be connected in series for extended applications

Chapter3. Application Notes

3-1.Pin definitions



Port Definitions

Port Number	Port	Function Description
1 ,3 ,5 ,13 ,15	GND	Ground Reference
7	EN	Receives any signal that could enable the decoding function of 74HC138
9	R1	Data input for 74HC595 shift registers (active low). Data of 16 rows on upper half of 6432 dot-matrix info board are provided by 8 pieces 74HC595 shift register. LED emits red color.
11	R2	Provide data input for 16 rows on another half of 6432 dot-matrix board. LED emits red color (active low).This pin is useless in 6416 board application.
2 ,4 ,6 ,8	A, B, C, D	These four pins receive and feed signals (active high or low) to a buffer, in which the signals are strengthened. The signals are then output

		to two 74HC138 decoders
10 ,12	G1, G2	Their definitions are the same as those of R1 and R2, except that the emit color of LED is green. They provide data to rows of 6432 dot-matrix board via other 16 pieces 74HC595 shift registers.
16	L	Clock input for 74HC595 shift register.
14	S	Via this pin, data of 74HC595 Shift Register is transferred to output latches.

Note:1. Data from 74HC595 shift registers for this board is active low.

2. If you want to light up a LED, select 0 (active low), otherwise 1 (active high).
3. The first port sits across from port A. This may help users locate each port.
4. The LEDs of dot-matrix board (Dia: 3mm) are effective when EN=0 while it's 1 for 5mm LED matrix board
5. Where no silkscreen can be found around a pin, this pin is GND.
6. The function definition of pins that are marked with “ST” is the same as that of pins marked with “L”. Similarly, pins that are marked with “CK” and”S” have the same function definition.

3-2. Timing Diagrams

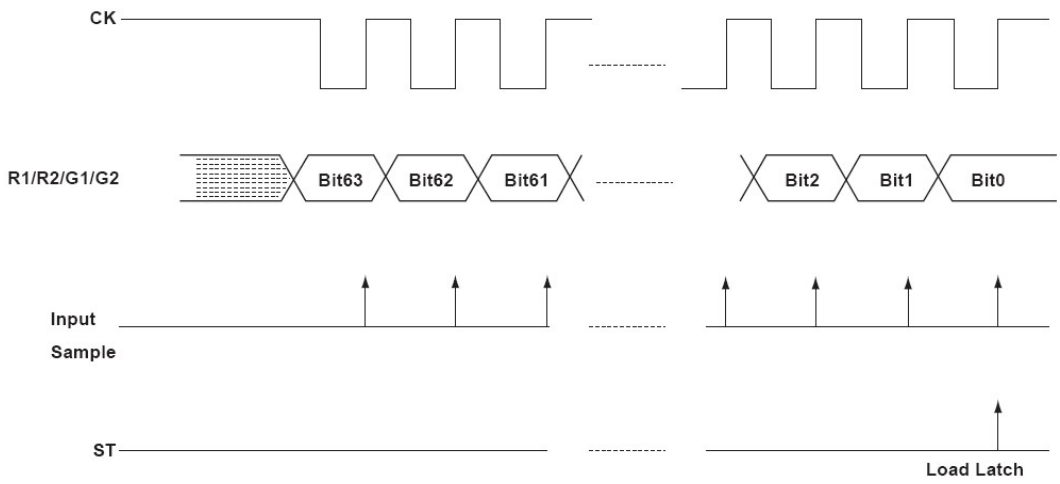


Figure 3

The board displaying frequency is 70 frames per second and screen refresh process should be completed in every frame.

The scanning time of one row should be no less than 0.78125ms.

The latch port will implement a low level to high level conversion (0 to1) to output data for display after sending the 64 bits data.

3-3. Notes:

1. Secure the power supply and GND connection of the LED display with the DEMO board. Otherwise, the LED display cannot be illuminated.

2. Secure the connection of IDC (flat cable) with both double-row socket on board and interface at the reverse side of LED display. Firstly, the IDC shall be connected with the interface at the reverse side of LED display marked with “in”, if you cannot find any interface marked with “in”, you shall use the one that sits closest to chip 74HC245. Pin 1 shall correspond to port 1 of the interface.

3 .Please be careful with the pins at the reverse side of LED display, they may pierce you finger.

4. The brightness that some of the LEDs of dot-matrix display (10 pcs for 6432 and 5 pcs for 6416) give off are comparatively weak after illumination, which is normal.

5. In order to prevent the LED info board from burning, be VERY CAREFULLY with the positive and negative polarity of the 5V power supply. At the same time, keep the reverse side of LED info board insulated from the work bench to avoid short circuit.

3-4. Codes and notes for testing the LED display

/*This program is used to ensure the dot matrix display's (6432(monocolor and bicolor), 6416(monocolor and bicolor)) function and if it can display characters correctly. So the testing process contains two sections. First, the upper half of the screen displays "If Glitch Press &Hold" Second; an operation cycle requires you to press the key for 13 times. */

```
#include<pic.h>

__CONFIG(WDTDIS & HS & UNPROTECT);
/*This will disable the watchdog timer, specify an HS crystal 25MHz
and leave the code space unprotected*/

#define Line_SELA       RC0//RC0 is connected to A pin of double-row socket
#define Line_SELB       RC1//RC1 is connected to B pin of double-row socket
#define Line_SELC       RC2//RC2 is connected to C pin of double-row socket
#define Line_SELD       RC3//RC3 is connected to D pin of double-row socket

#define Line_EN    RB1//RB1 is connected to EN pin of double-row socket
#define CLK        RB3//RB3 is connected to CLK pin of double-row socket
#define LAT        RB2//RB2 is connected to ST pin of double-row socket

#define DAT_R1          RC4//RC4 is connected to R1 pin of double-row socket
#define DAT_R2          RC5//RC5 is connected to R2 pin of double-row socket
#define DAT_G1          RC6//RC6 is connected to G1 pin of double-row socket
#define DAT_G2          RC7//RC7 is connected to G2 pin of double-row socket

#define KEY             RB0 //RB0 is connected to key

/*This group of data is specifically used to display "If Glitch Press & Hold"
And "Sure Electronics"*/
```

```
const unsigned char CoName[224]={
                                    0x0e,0x03,0x38,0x06,0x21,0x00,0x01,0x00,
                                    0x84,0x04,0x44,0x04,0x20,0x00,0x01,0x00,
                                    0x84,0x00,0x04,0x84,0x71,0x38,0x0d,0x00,
                                    0xc4,0x01,0x74,0x04,0x21,0x04,0x13,0x00,
                                    0x84,0x00,0x44,0x04,0x21,0x04,0x11,0x00,
                                    0x84,0x00,0x44,0x04,0x21,0x45,0x11,0x00,
                                    0x8e,0x00,0x78,0x8e,0xc3,0x38,0x11,0x00,

                                    0x0f,0x00,0x00,0x80,0x11,0x01,0x06,0x04,
                                    0x11,0x00,0x00,0x40,0x12,0x01,0x04,0x04,
                                    0x51,0xe3,0x38,0x4e,0x11,0x39,0x84,0x05,
                                    0xcf,0x14,0x05,0x81,0xf0,0x45,0x44,0x06,
                                    0x41,0xf0,0x39,0x4e,0x15,0x45,0x44,0x04,
                                    0x41,0x10,0x40,0x50,0x12,0x45,0x44,0x04,
                                    0x41,0xe0,0x3c,0x8f,0x15,0x39,0x8e,0x07,

                                    0x0e,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
                                    0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
                                    0x41,0xd2,0x38,0x00,0x00,0x00,0x00,0x00,
                                    0x4e,0x32,0x45,0x00,0x00,0x00,0x00,0x00,
                                    0x50,0x12,0x7c,0x00,0x00,0x00,0x00,0x00,
                                    0x51,0x12,0x04,0x00,0x00,0x00,0x00,0x00,
                                    0x8e,0x15,0x38,0x00,0x00,0x00,0x00,0x00,

                                    0x9f,0x01,0x00,0x00,0x00,0x00,0x04,0x00,
                                    0x01,0x01,0x00,0x02,0x00,0x00,0x00,0x00,
                                    0x01,0xe1,0x38,0x47,0xe3,0x34,0xc6,0xf1,
                                    0x1f,0x11,0x05,0xc2,0x14,0x4d,0x24,0x08,
                                    0x01,0xf1,0x05,0x42,0x10,0x45,0x24,0x70,
                                    0x01,0x11,0x44,0x52,0x10,0x45,0x24,0x82,
                                    0x9f,0xe3,0x38,0x4c,0xe0,0x44,0xce,0x79
                                };

unsigned char cnt,flag,guide,cnt_flag;//flag =0 suggests diameter of LED is 5mm
                                    //flag =1 suggests diameter of LED is 3mm
                                    //guide =0 suggests waiting to be confirm
                                    //guide =1 suggests testing process
                                    //Variable “cnt_flag” ranges from 0 to 111,
        assistant flag of element of data array
```

```
//These variables are used to store key states
unsigned char keydata,key_last,key_now;
void delay_100ms(void)
{
    unsigned char i,j;
    for(j=0;j<208;j++)
        for(i=0;i<250;i++);
}
void delay_sometime()
{
    unsigned char   j;
    for(j = 0; j < 100; j++);
}

void init(void)
{
    TRISA=0b0000;
    TRISB=0b00001;              //RB0 input ,RB1,RB2,RB3 output
    TRISC=0b00000000;           //RC0,RC1,RC2,RC3,RC4,RC5,RC6,RC7output


    key_last=KEY;
    key_now=KEY;

    keydata=12;                 //Dieplay Notice
    guide =0;                   //Wait to be confirmed
    flag=0;
    cnt_flag=0;
}

void judgekey(void)
{
    key_now=KEY;
    if((key_last==1)&(key_now==0))
    {
        delay_sometime(); //Delay for according to buffeting
        if(key_now==0)
        {
            keydata++;
            keydata=keydata%13;
```

```
            }
        }
        key_last=key_now;
}

//send a line of data which contains 64 bits.
//send one bit with clk down to up
//every 64 bits should be ended with latch down to up

void sendone(unsigned char line)
{
        unsigned char k,value,t;
        //Variable “k” counts times of bit sent in a line
        //Variable “value” tells value to be sent
        //Variable “t” is available only when sending character of data array
        if (!flag)
                Line_EN=0;//Turn off DE-DP029~DE-DP032
        else
                Line_EN=1;//Turn off DE-DP033
        value=0x00;
        if((keydata==0)||(keydata==3))
                value=~value;
        for(k=0;k<64;k++)//Send a line of data which contains 64 bits.
        {
                switch(keydata)
                {
                        case 0://0 illuminates the upper left 4*4 area of each 8*8 dot matrix
                        case 1://1 illuminates the upper right 4*4 area of each 8*8 dot matrix
                        case 2://2 illuminates the lower right 4*4 area of each 8*8 dot matrix
                        case 3:{//3 illuminates the lower left 4*4 area of each 8*8 dot matrix
                                        if(k%4==0)
                                                value=~value;

                                        DAT_R1=value;
                                        DAT_G1=value;
                                        DAT_R2=value;
                                        DAT_G2=value;
                                } break;
//4~7 for checking red LED, 4 illuminates odd number rows
//and 5 illuminates even number rows
```

```
                case 4:{
                        DAT_R1=value;
                        DAT_G1=1;
                        DAT_R2=value;
                        DAT_G2=1;
                        value=~value;
                    }break;
                 case 5:{
                        value=~value;
                        DAT_R1=value;
                        DAT_G1=1;
                        DAT_R2=value;
                        DAT_G2=1;
                    }break;
//6 illuminates the adjacent two lines of LED starting from line 1
//and 2 and those followed at two lines intervals.
//7 illuminates the adjacent two lines of LED starting from line 3
//and 4 and those followed at two lines intervals.
                case 6:
                case 7:{
                        DAT_R1=value;
                        DAT_G1=1;
                        DAT_R2=value;
                        DAT_G2=1;
                }break;
//8~11 for checking green LED, 8 illuminates odd number rows
//and 9 illuminates even number rows
                case 8:{
                        DAT_G1=value;
                        DAT_R1=1;
                        DAT_R2=1;
                        DAT_G2=value;
                        value=~value;
                    }break;
                case 9:{
                        value=~value;
                        DAT_G1=value;
                        DAT_R1=1;
                        DAT_R2=1;
                        DAT_G2=value;
```

```
                    }break;
//10 illuminates the adjacent two lines of LED starting from line 1
//and 2 and those followed at two lines intervals.
            case 10:
//11 illuminates the adjacent two lines of LED starting from line 3
//and 4 and those followed at two lines intervals.
            case 11:{
                    DAT_R1=1;
                    DAT_G1=value;
                    DAT_R2=1;
                    DAT_G2=value;
                    }break;
            case 12:{
                    if(line%8==7)
                        t=1;
                    else
                        {
                            if(k%8==0)
                            {
                                //fetch a character from data array

    value=*(CoName+(guide<<7)-(guide<<4)+cnt_flag);
                                //each LED is illuminated with 0
                                value=~value;
                                cnt_flag++;
                            }
                            t=value & 0x01;
                        }
                    DAT_R1=t;
                    DAT_G1=t;
                    DAT_R2=1;
                    DAT_G2=1;

                    value=value>>1;

                    }break;

        }

        CLK=0;//send one bit with clk down to up
```

```
            CLK=1;

        }
        if(!flag)
            Line_EN=1;//Turn on DE-DP029~DE-DP032
        else
            Line_EN=0;//Turn on DE-DP033

        LAT=0;//Every 64 bits should be ended with latch down to up
        LAT=1;

        if(cnt_flag==112)
        cnt_flag=0;

        delay_sometime();//Delay some time for display

        if(!flag)
            Line_EN=0;//Turn off DE-DP029~DE-DP032
        else
            Line_EN=1;//Turn off DE-DP033

}

/*This program is used for dynamic scanning display */
void rundisplay()
{
        unsigned char t,temp;
        for(t=0;t<16;t++)
        {
            temp=t;
            Line_SELA=temp & 0x01;              //Select line
            temp=temp>>1;
            if((keydata==6)||(keydata==10))
                Line_SELB=0;
            else if((keydata==7)||(keydata==11))
                Line_SELB=1;
            else
                Line_SELB=temp & 0x01;
            temp=temp>>1;
            if(keydata<2)
```

```
                Line_SELC=0;
            else if((keydata==2)||(keydata==3))
                Line_SELC=1;
            else
                Line_SELC=temp & 0x01;
            temp=temp>>1;
            Line_SELD=temp & 0x01;

            sendone(t);
        }
}
/*When time of pressing the key is less than 1 second,
it may be seemed as a short time , no less
than 1 second as a long time */
void LongOrShort(void)
{
    key_now=KEY;
    if((key_last==1)&(key_now==0))
    {
        NOP();//Delay for confirming the pressing action
        while(!key_now)
        {
            delay_100ms();
            cnt++;
            key_now=KEY;
        }

        if(cnt < 10)
            flag = 0;
        else
            flag = 1;
        guide = 1;     //Go to testing process
    }
    key_last=key_now;
}
void main(void)
{
    init();
    while(1)
    {
```

```
            if(guide==0)
                LongOrShort();
            else
                judgekey();
                rundisplay();
        }
}
```

Chapter4. Contact Us

Sure Electronics Co., Ltd.
5F, Zone A,
Qinhuai Technology Innovation Center
105-2 DaMing Rd (Zip Code: 210022)
Nanjing
PRC
Tel: +86-25-66606340 (English Service) GMT1am-10am
Fax: +86-25-66606346
Website: www.sure-electronics.net
www.sureelectronics.net
www.sure-electronics.com