Lesson 14 LED Matrix Display

14.1 Overview

This lesson focuses on the use of a 16*8 LED matrix display. It details the necessary components, working principles, wiring methods, demonstration procedures, and code implementation. Through this lesson, users can learn how to control the LED matrix to display specific patterns, enhancing their understanding of electronic components and programming applications.

14.2 Required Components

Components	Quantity	Picture
Adeept Robot Control Board	1	
Type-C USB Cable	1	
LED matrix module	1	

14.3 Principle Introduction

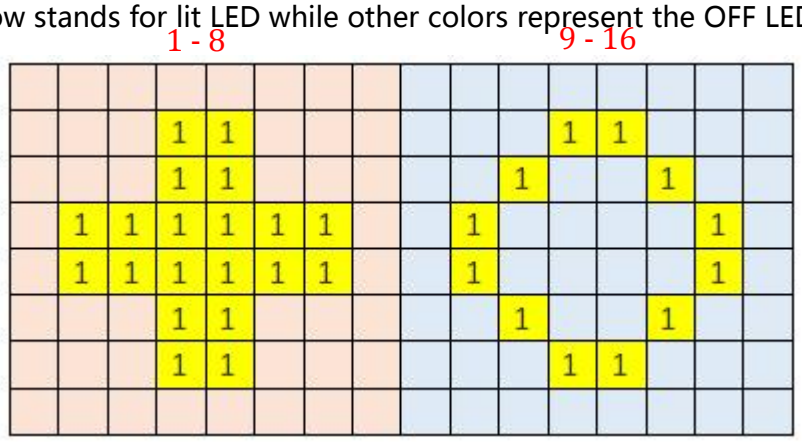
A LED matrix is a rectangular display module that consists of a uniform grid of LEDs. The following is an 8X8 monochrome LED matrix containing 64 LEDs (8 rows by 8 columns).

In order to facilitate the operation and reduce the number of ports required to drive this component, the positive poles of the LEDs in each row and negative poles of the LEDs in each column are respectively connected together inside the LED matrix module, which is called a common anode. There is another arrangement type. Negative poles of the LEDs in each row and the positive poles of the LEDs in each column are respectively connected together, which is called a common cathode. The default address of LED matrix is 0x70.

The principle of 8*16 LED matrix:

a byte has 8 bits, each bit is 0 or 1. When a bit is 0, turn off LED and when a bit is 0, turn on LED. Thereby, one byte can control the LED in a columns of dot matrix, so 16 bytes can control 16 columns of led lights, that is, 8*16 dot matrix.

We divide the LED matrix into two sides and display “+” on the left and “o” on the right. As shown below, yellow stands for lit LED while other colors represent the OFF LED.

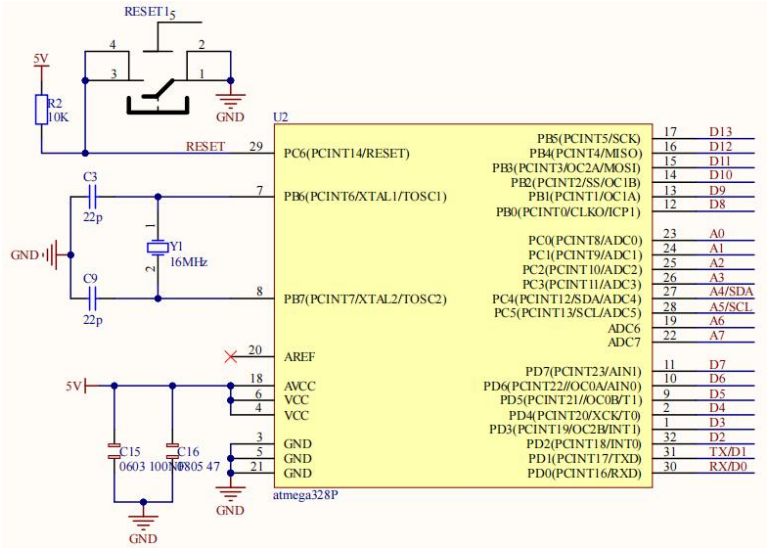


Below, the table on the left corresponds to the "+" above, and the table on the right corresponds to the "o" above.

Row	Binary	Hexadecimal	Row	Binary	Hexadecimal
1	0000 0000	0x00	9	0000 0000	0x00
2	0001 1000	0x18	10	0001 1000	0x18
3	0001 1000	0x18	11	0010 0100	0x24
4	0111 1110	0x7e	12	0100 0010	0x42
5	0111 1110	0x7e	13	0100 0010	0x42
6	0001 1000	0x18	14	0010 0100	0x24
7	0001 1000	0x18	15	0001 1000	0x18
8	0000 0000	0x00	16	0000 0000	0x00

Matrix Screen	Arduino(X12)
SDA	A4
SCL	A5
GND	GND
VCC	5V

14.4 Wiring Diagram



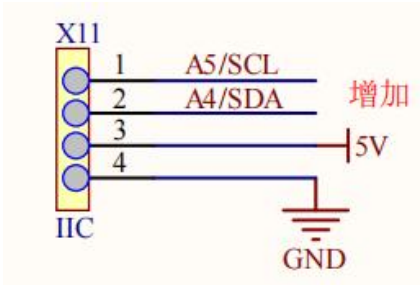
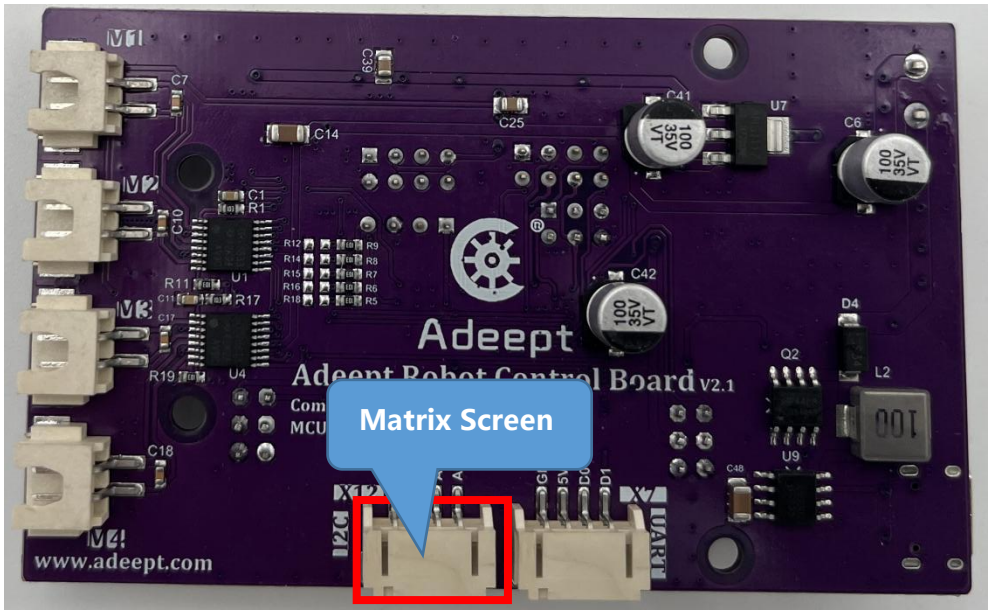
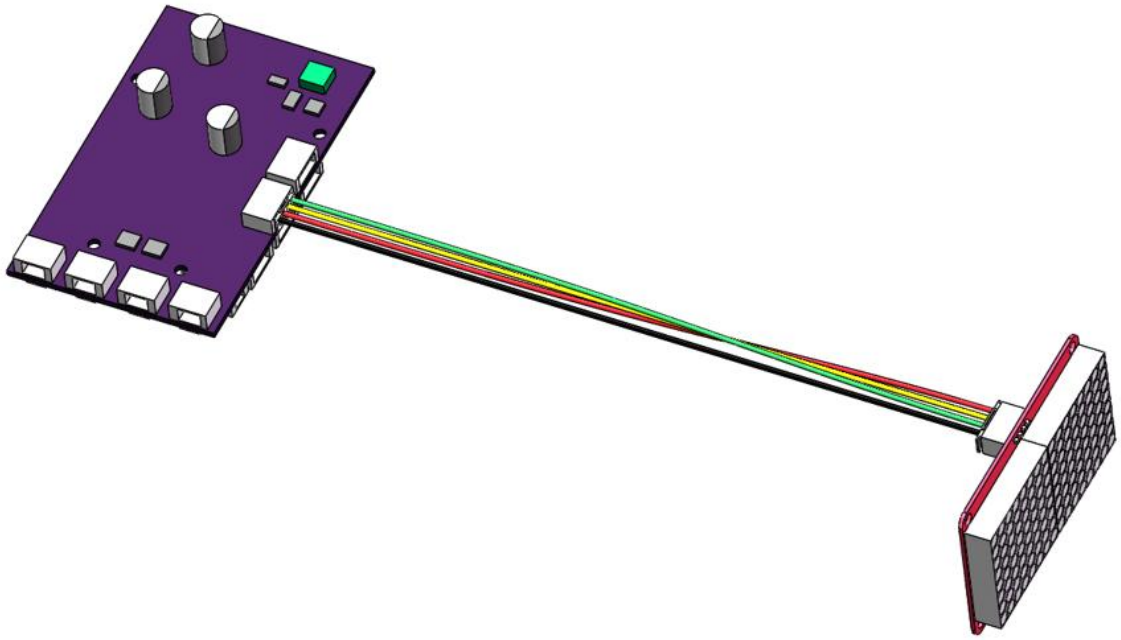


Figure as below：

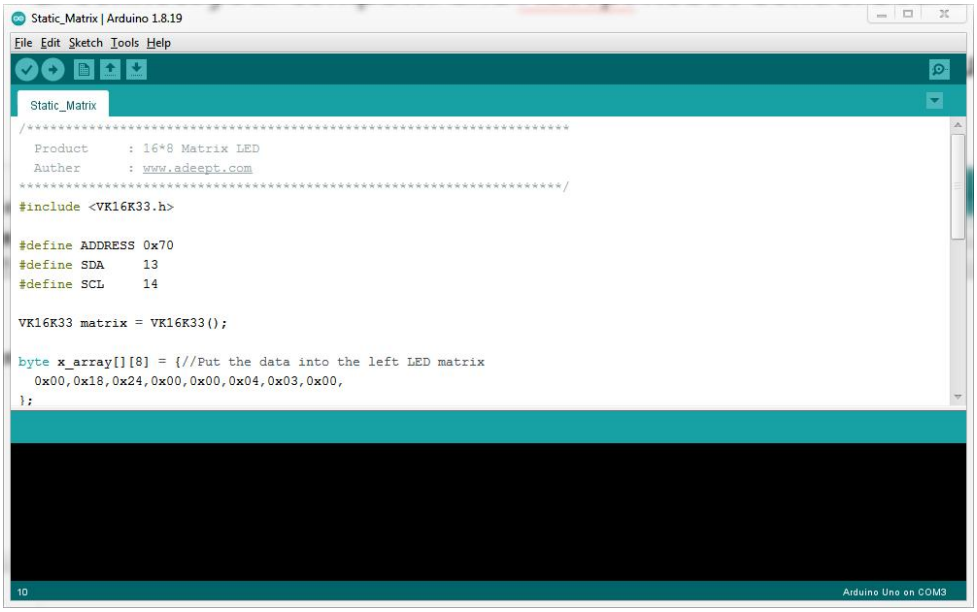




14.5 Demonstration

1. Connect your computer and Adeept Robot Control Board (Arduino Board) with a USB cable.

2. Open "**10_Matrix/Static_Matrix**" folder in "/Code" ,double-click "Static_Matrix.ino" .

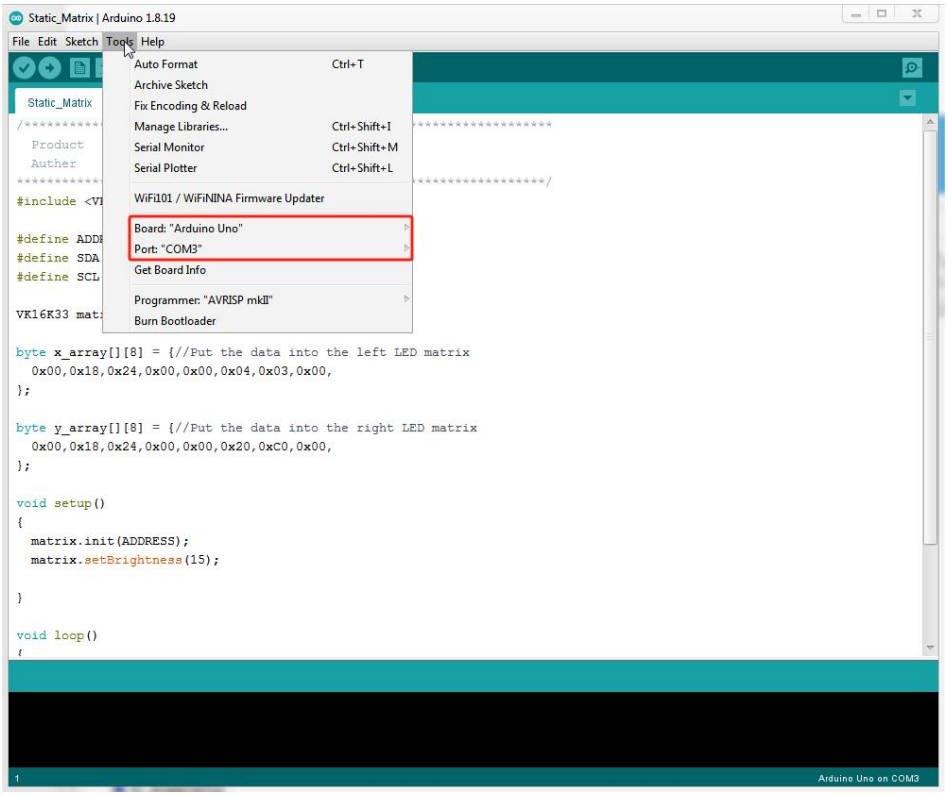


3. Select development board and serial port.

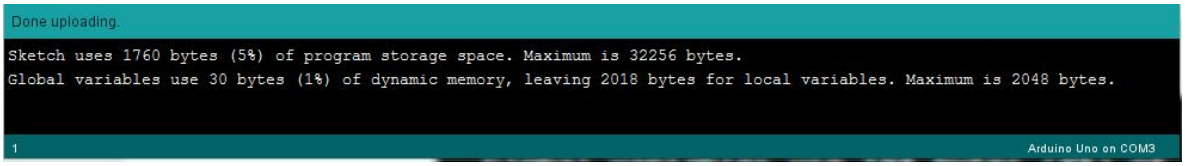
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

Port: Tools --->Port--->COMx

Note: The port number will be different in different computers.

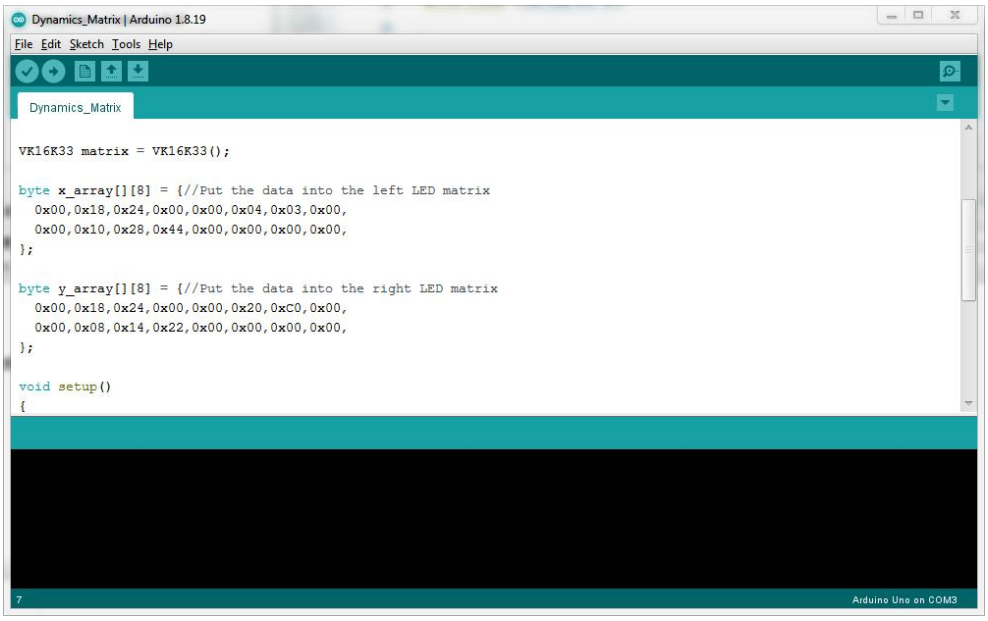


4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.

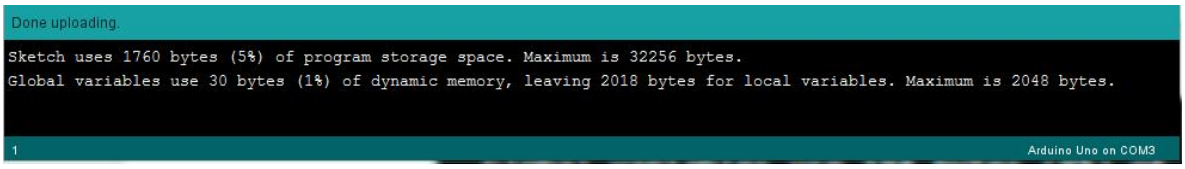


5. After successfully running the program, you will see the pattern defined by the 'x_array' array data on the left side of the 16 * 8 matrix LED display, and the pattern defined by the 'y_array' array data on the right side.

6. Open "**10_Matrix/Dynamics_Matrix**" folder in "/Code", double-click "Dynamics_Matrix.ino".



7. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



8. After running the program, the 16 * 8 matrix LED display screen shows the pattern defined by 'x_array' on the left and the pattern defined by 'y_array' on the right. The patterns are displayed in sequence, with each display lasting 500 ms.

14.6 Code

Complete code refer to Static_Matrix.ino

```
01 /************************************************************************
02   Product     : 16*8 Matrix LED
03   Auther      : www.adeept.com
04 ************************************************************************/
05 #include <VK16K33.h>
06
07 #define ADDRESS 0x70
08 #define SDA     13
09 #define SCL     14
10
11 VK16K33 matrix = VK16K33();
12
13 byte x_array[][8] = {//Put the data into the left LED matrix
14   0x00,0x18,0x24,0x00,0x00,0x04,0x03,0x00,
```

```
};

byte y_array[][8] = {//Put the data into the right LED matrix
  0x00,0x18,0x24,0x00,0x00,0x20,0xC0,0x00,
};

void setup()
{
  matrix.init(ADDRESS);
  matrix.setBrightness(15);

}

void loop()
{
  int count = sizeof(x_array) / sizeof(x_array[0]);
  for (int i = 0; i < count; i++) {
    matrix.showStaticArray(x_array[i], y_array[i]);
    delay(500);
  }
}
```

Complete code refer to Dynamics_Matrix.ino

```
/************************************************************************
  Product     : 16*8 Matrix LED
  Auther      : www.adeept.com
************************************************************************/
#include <VK16K33.h>

#define ADDRESS 0x70
#define SDA     13
#define SCL     14

VK16K33 matrix = VK16K33();

byte x_array[][8] = {//Put the data into the left LED matrix
  0x00,0x18,0x24,0x00,0x00,0x04,0x03,0x00,
  0x00,0x10,0x28,0x44,0x00,0x00,0x00,0x00,
};

byte y_array[][8] = {//Put the data into the right LED matrix
  0x00,0x18,0x24,0x00,0x00,0x20,0xC0,0x00,
  0x00,0x08,0x14,0x22,0x00,0x00,0x00,0x00,
};

void setup()
{
  matrix.init(ADDRESS);
  matrix.setBrightness(15);

}

void loop()
{
  int count = sizeof(x_array) / sizeof(x_array[0]);
  for (int i = 0; i < count; i++) {
```

```
    matrix.showStaticArray(x_array[i], y_array[i]);
    delay(1000);
  }
}
```

Code explanation

Static_Matrix.ino

Initialization Stage:

In the setup () function, first initialize I2C communication and set the pins for the data and clock lines; Then initialize the LED matrix display screen and input the previously defined I2C address; Finally, set the brightness of the display screen to the maximum value of 15.

Loop Control Process:

The current pattern data is written to the LED matrix display screen.

Dynamics_Matrix.ino

Initialization Stage:

In the setup () function, first initialize I2C communication and set the pins for the data and clock lines; Then initialize the LED matrix display screen and input the previously defined I2C address; Finally, set the brightness of the display screen to the maximum value of 15.

Loop Control Process:

Write the current pattern data into the LED matrix display screen; Finally, use the delay (1000) function for a brief delay.