

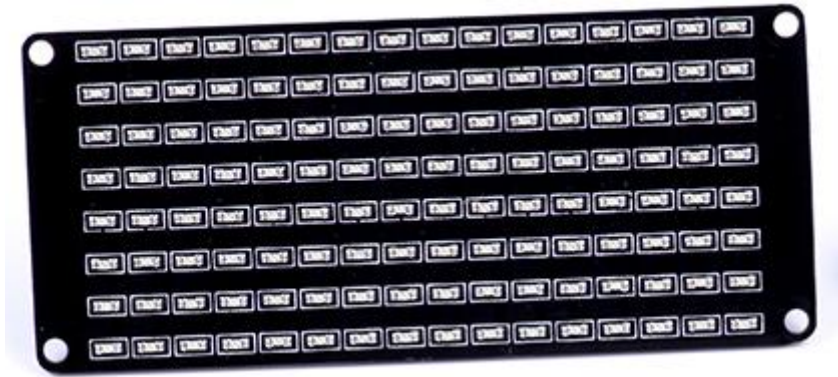
Instructions for Me LED Matrix 8×16 Module

1. Module introduction

Operating voltage: 5V DC

Communication interface: Double digital interface (blue label). The LED Matrix can be connected to interface 1, 2, 3, and 4 of mCore and connected to interface 3, 4, 5, and 6 of Orion. Baseboard can be connected with interface 3, 4, 5, 6, 7, and 8.

Appearance:

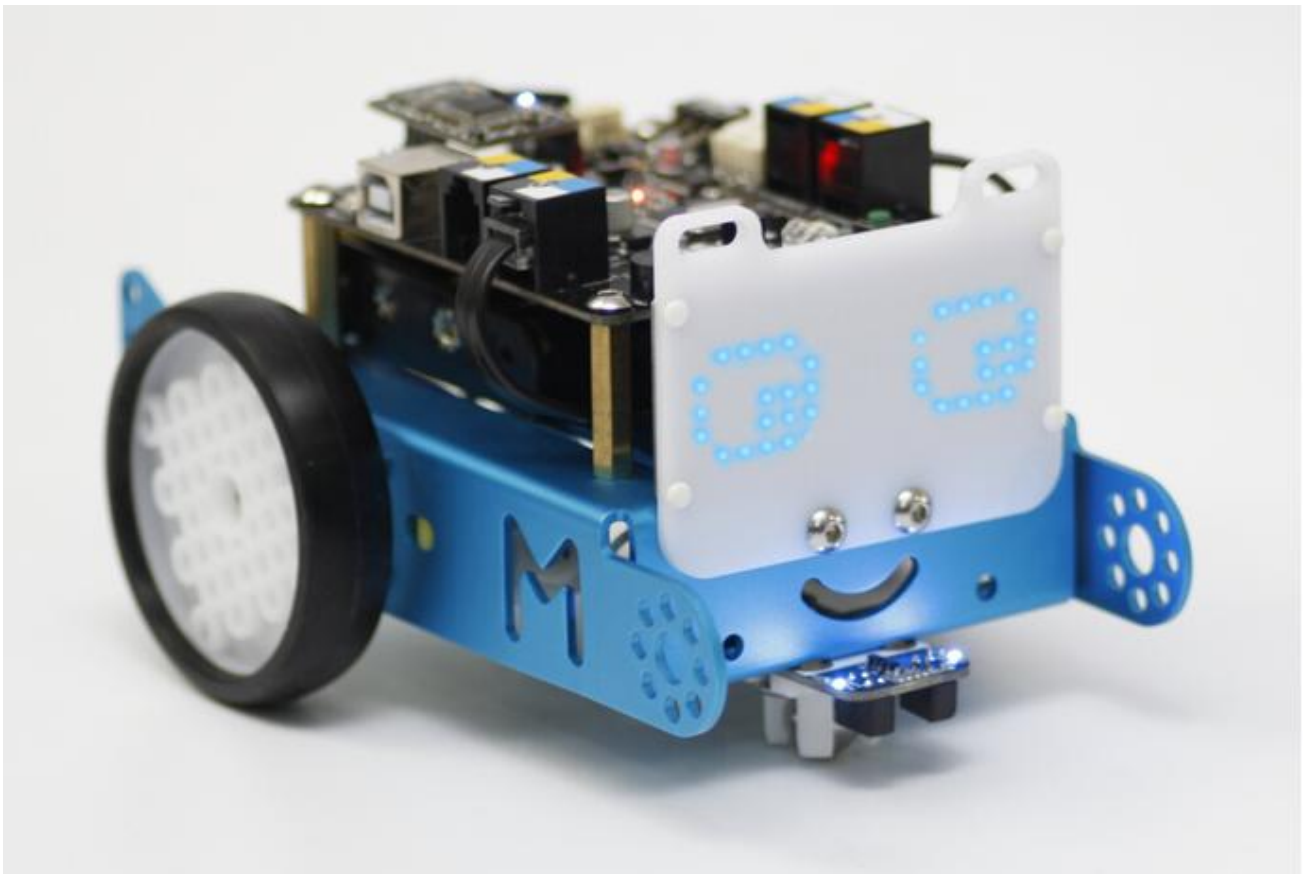
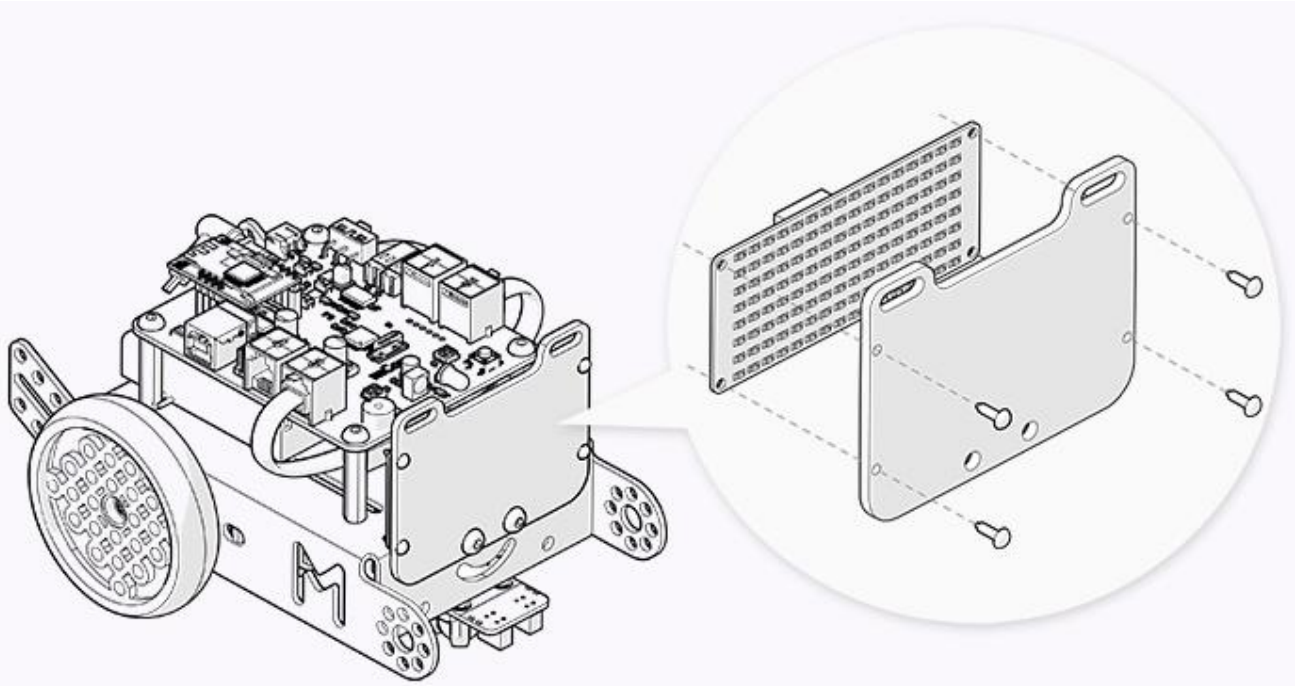


Front

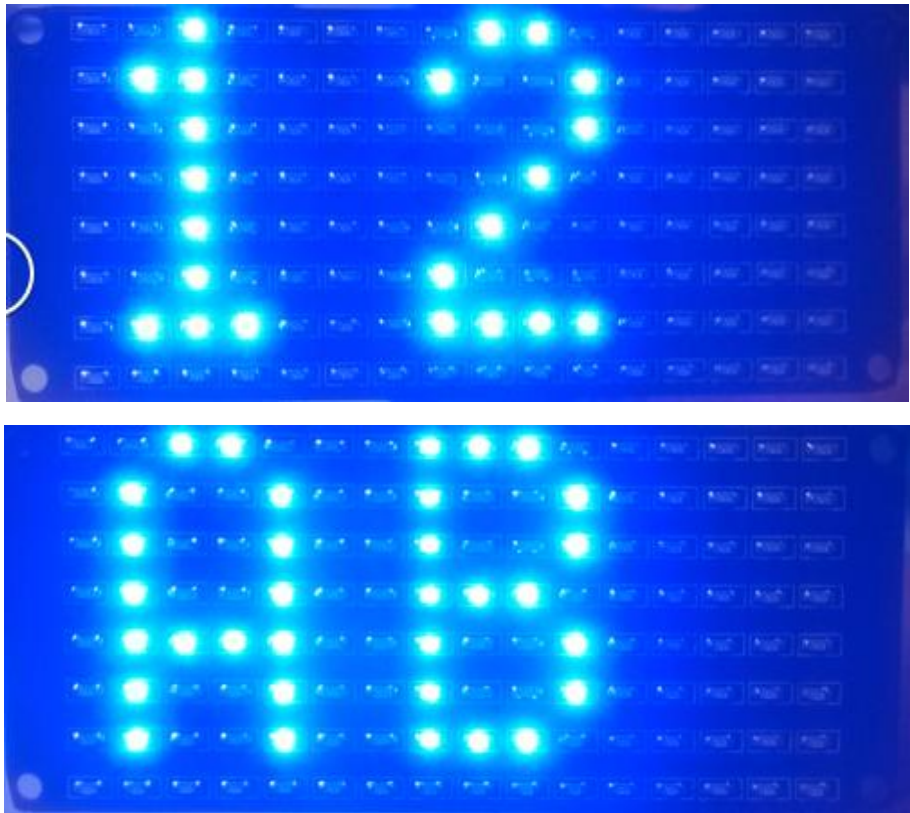


Back

Assembly:



Application: The module is an 8X16 LED matrix screen composed of 16 horizontal and 8 vertical blue LED lamps. Through turning on or off LED lamps, certain basic figures, letters, and simple patterns can be showed. See the following diagram.



2. Instructions

2.1 Arduino IDE Programming

Tips: How to view source code about LED Matrix module?

First, open the software mBlock, then select "Extensions->Manage Extensions ->mBot->view source code"

Library function list

Function name	Description	Property
MeLEDMatrix	Constructor	Public function
clearScreen	Screen clearing function	Public function
setBrightness	Brightness setting function	Public function
setColorIndex	Color index show setting (1: Normally lighting, 0: Color negation);	Public function

drawBitmap	Bitmap (lattice diagram) show function;	Public function
drawStr	Character string show function;	Public function
showClock	Clock show function	Public function
showStr	Character strings show function	Private function

Introduction of functions performance

1. MeLEDMatrix Constructor 1

Function name	MeLEDMatrix
Function object	MeLEDMatrix ();
Performance description	Empty constructor
Input parameters	N/A
Returned value	N/A
Prerequisite	N/A
Called function	N/A

[Return to library function list](#)→

2. MeLEDMatrix Constructor 2

Function name	MeLEDMatrix
Function object	MeLEDMatrix (uint8_t port);
Performance description	Constructor. Initialize module according to port parameters;
Input parameters	Port: Module control port. Value range: PORT_1 ~ PORT_8

Returned value	N/A
Prerequisite	N/A
Called function	writeByte (); // function of writing a single-byte; setBrightness (); // Brightness setting function; clearScreen (); // Screen clearing function;

[Return to library function list→](#)

3. MeLEDMatrix Constructor 3

Function name	MeLEDMatrix
Function object	MeLEDMatrix (uint8_t SCK_Pin,uint8_t DIN_Pin);
Performance description	Constructor. Initialize module according to pin parameters;
Input parameters	SCK_Pin: Module clock line control pin DIN_Pin: Module data line control pin
Returned value	Nil
Prerequisite	Nil
Called function	writeByte (); // function of writing a single-byte; setBrightness (); // Brightness setting function; clearScreen (); // Screen clearing function;

[Return to library function list→](#)

4. clearScreen

Function name	clearScreen
Function object	void clearScreen ();
Performance description	Screen clearing function
Input parameters	N/A

Returned value	N/A
Prerequisite	N/A
Called function	writeBytesToAddress (); // Write certain bytes to specified address;

[Return to library function list→](#)

5. setBrightness

Function name	setBrightness
Function object	void setBrightness (uint8_tBright);
Performance description	Brightness setting function
Input parameters	<p>Bright: Brightness parameters. Value range:</p> <p style="padding-left: 40px;">Brightness_0 // Darkest (turning off) Brightness_1 Brightness_2 Brightness_3 Brightness_4 Brightness_5 Brightness_6 Brightness_7 Brightness_8 // Brightest</p>
Returned value	N/A
Prerequisite	N/A
Called function	writeByte (); // function of writing a single-byte;

[Return to library function list→](#)

6. setColorIndex

Function name	setColorIndex
Function object	void setColorIndex (bool Color_Number);
Performance description	Color index show setting (1: Normally lighting, 0: Color negation);
Input parameters	Color_Number: Color parameters. 1: Show diagram with lighting lamps; 0: Show with turning off lamps;

	Remarks: <1> Default of color parameter system of LED lattice screen is 1 (showing diagram with lighting lamps); <2> clearScreen(), after screen clearing function is called, the color parameter will be reset as 1;
Returned value	N/A
Prerequisite	N/A
Called function	N/A

[Return to library function list→](#)

7. drawBitmap

Function name	drawBitmap
Function object	void drawBitmap (uint8_t x, uint8_t y, uint8_t Bitmap_Width, uint8_t *Bitmap);
Performance description	Bitmap (lattice diagram) show function. Bitmap height is fixed as 8 (pixel);
Input parameters	x: Value range of top left corner x-coordinate of bitmap (lattice diagram): 0 ~ 15; y: Value range of top left corner y-coordinate of bitmap (lattice diagram): 0 ~ 7; Bitmap_Width: Width (pixel) of bitmap (lattice diagram) : Bitmap: Array name of bitmap lattice code (buffer area);
Returned value	N/A
Prerequisite	N/A
Called function	writeBytesToAddress (); // Write certain bytes to specified address;

[Return to library function list→](#)

8. drawStr

Function name	drawStr
Function object	void drawStr (int16_t X_position, int8_t Y_position, const char *str);
Performance description	Character strings show function. Provisionally, it is only able to show upper and lower case letters, figures, and blank space, and unknown character will be filled with blank space;

	Character strings height is fixed as 8 (pixel);
Input parameters	X_position: Bottom left corner X-coordinate of first character of strings Value range: The smallest value is decided by length of character strings, The biggest value: 16 Y_position: Bottom left corner Y-coordinate of first character of strings Value range: -1 ~ 15; str: Starting address of character strings buffer area (array) to be shown;
Returned value	N/A
Prerequisite	N/A
Called function	showStr (); // Character strings show function, private function.

[Return to library function list→](#)

9. showClock

Function name	showClock
Function object	void showClock (uint8_t hour, uint8_t minute, bool point_flag = PointOn);
Performance description	Clock show function.
Input parameters	hour: Hour; minute: Minute; point_flag: Two points (colon) show label between clock point_flag = PointOn (point_flag = 1) indicates that two points are shown; point_flag = PointOff (point_flag = 0) indicates that two points are not shown;
Returned value	N/A
Prerequisite	N/A
Called function	writeBytesToAddress (); // Write certain bytes to specified address;

[Return to library function list→](#)

Example code:


```

#include<mBot.h>
#include <MeLEDMatrix.h>
#include <MePort.h>
//MeBoard myBoard(MakeblockOrion);
MeBoard myBoard(mBot);
MeLEDMatrix Matrix_1(PORT_4);
char string_data[]="MAKEBLOCK 012345678";
uint8_t Bitmap_Heart[16]=
{
    0x00,0x38,0x44,0x42,0x21,0x21,0x42,0x44,0x38,0x44,0x42,0x21,0x21,0x42,0x44,0x38,
};

int move_times = sizeof(string_data)*6;

void setup()
{
    Matrix_1.setBrightness(Brightness_8);//

}

void loop()
{
    /***** Example 1 *****/
    Matrix_1.setColorIndex(1);//
    Matrix_1.drawBitmap(0, 0, sizeof(Bitmap_Heart), Bitmap_Heart);
    for(uint8_t k=0; k<3; k++)
    {
        for(uint8_t i=0;i<8;i++)
        {
            Matrix_1.setBrightness(i);
            delay(100);
        }

        for(uint8_t i=7;i>0;i--)
        {
            Matrix_1.setBrightness(i);
            delay(100);
        }
    }
    /***** Example 1 *****/

    /***** Example 2 *****/
    Matrix_1.setColorIndex(0);

```

```

Matrix_1.drawBitmap(0, 0, sizeof(Bitmap_Heart), Bitmap_Heart);
for(uint8_t k=0; k<3; k++)
{
    for(uint8_t i=0;i<8;i++)
    {
        Matrix_1.setBrightness(i);
        delay(100);
    }

    for(uint8_t i=7;i>0;i--)
    {
        Matrix_1.setBrightness(i);
        delay(100);
    }
}

```

***** Example 2*****/

/***** Example 3*****

```

for(int16_t i=0; i<move_times; i++)
{
    if(i >move_times)i=0;
    Matrix_1.drawStr(15-i,7,string_data);

    delay(100);
}

```

***** Example 3*****/

/***** Example 4 *****

```

for(uint8_t k=0; k<5; k++)
{
    Matrix_1.showClock(12, 34, PointOn);
    delay(500);
    Matrix_1.showClock(12, 34, PointOff);
    delay(500);
}

```

***** Example 3*****/

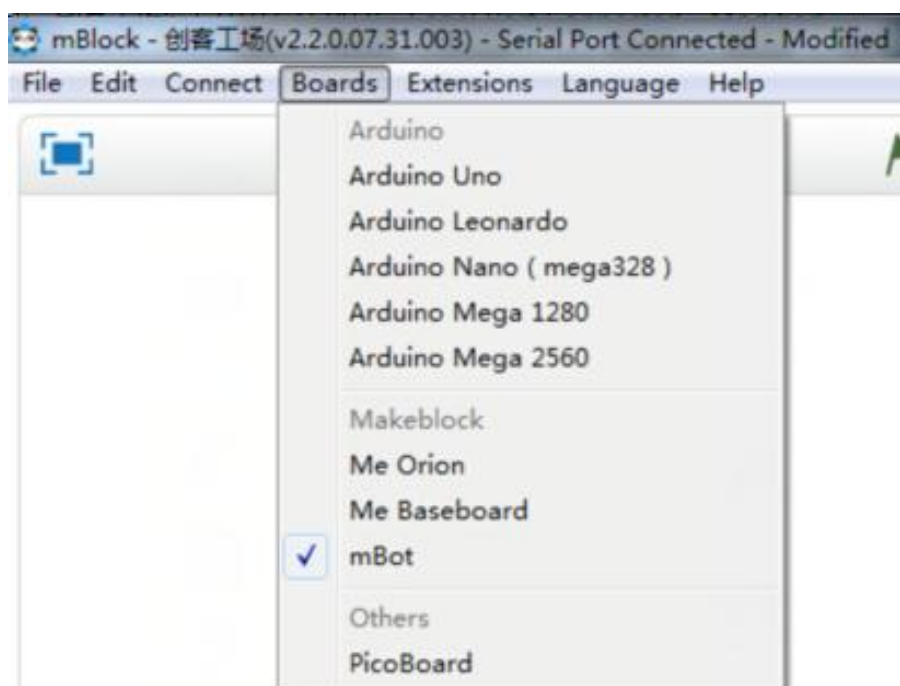
}

2.2 mBlock programming

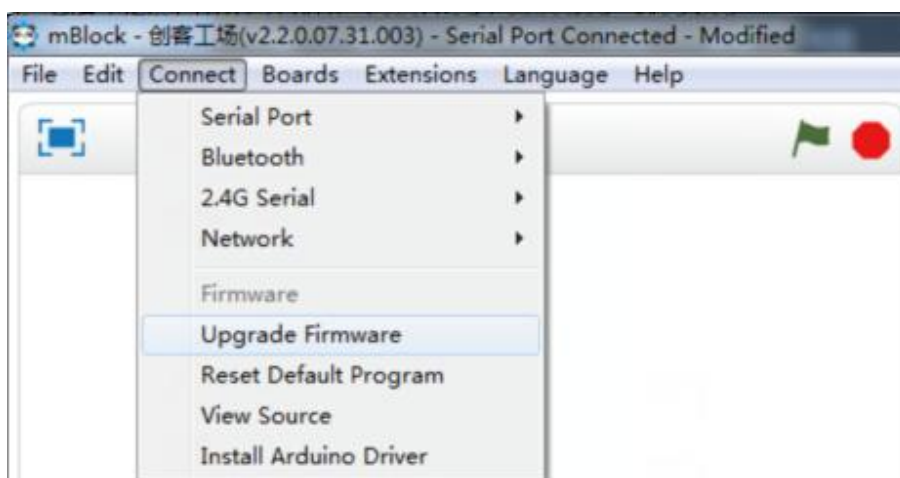
- **Step 1** , After you connect with USB cable and turn on the mBot, you need to choose the serial port first.



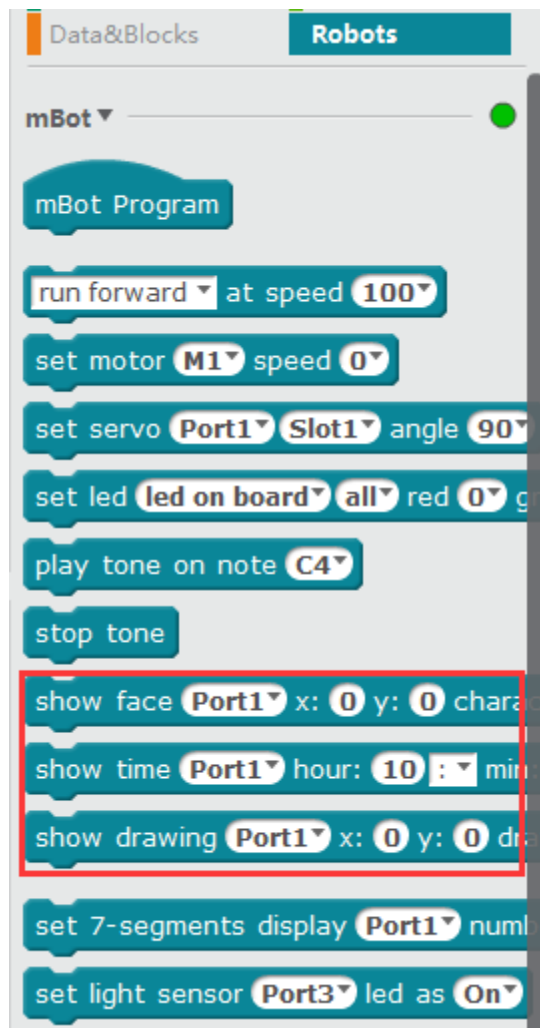
- **Step 2**, Choose the board type



- **Step 3**, Upgrade Firmware



- **Step 4**, Drag script blocks for LED Matrix



Script block introduction:

- 1) Input characters



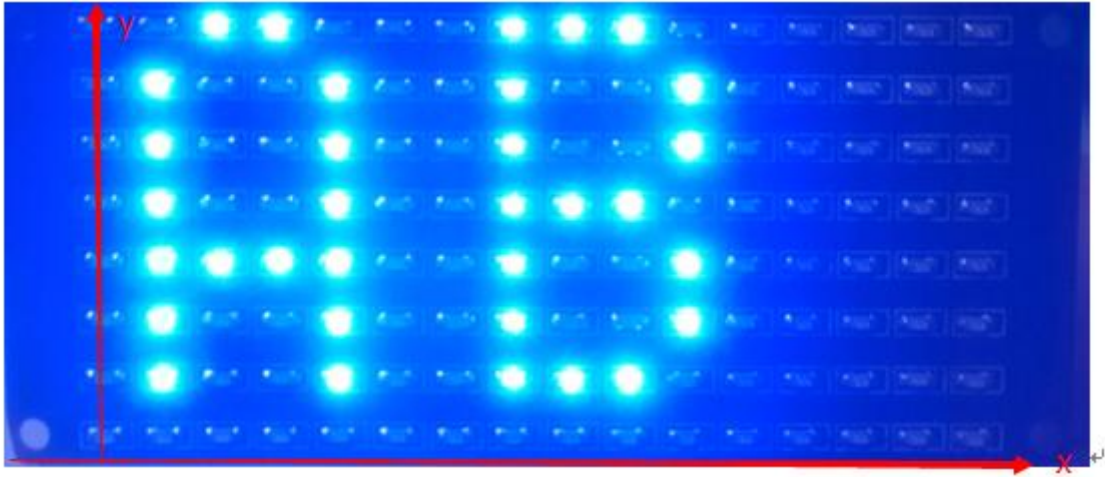
Parameter:

“Port1” - Choose the corresponding port you have connected the LED Matrix to.

“x:0” - Set the x coordinate as 0.

“y:0” - Set the y coordinate as 0.

“characters:Hello” - Set the showing characters as Hello.



2) Input time



Parameter:

“Port1” - Choose the corresponding port you have connected the LED Matrix to.
About hour&min - Set the showing characters as current time.

3) User-defined characters and patterns



Click plotting area and the following plotting panel will be popped-up.



Select LED lamps that you want to turn on and click will be popped-up. n; -1 ~ 15;
-t v