



## mBot Matriz de LEDs



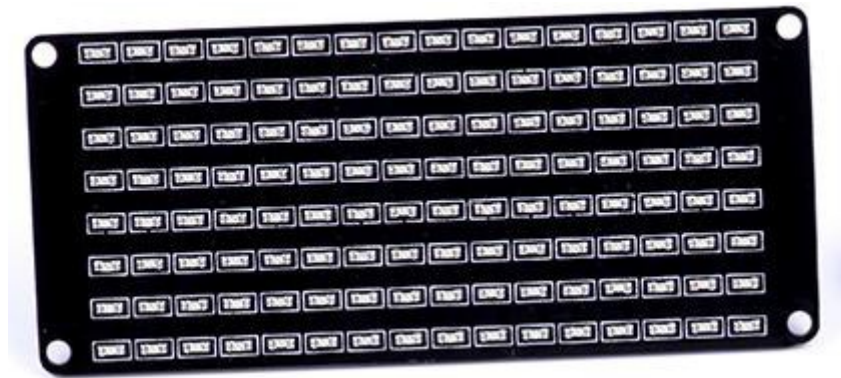
## Instrucciones para el módulo matriz Me LED 8×16

### 1. Introducción del módulo

**Voltaje de funcionamiento:** 5V DC

**Interfaz de comunicación:** interfaz digital doble (etiqueta azul). La matriz LED se puede conectar la placa controladora mCore en los puertos 1, 2, 3, y 4.

**Apariencia:**

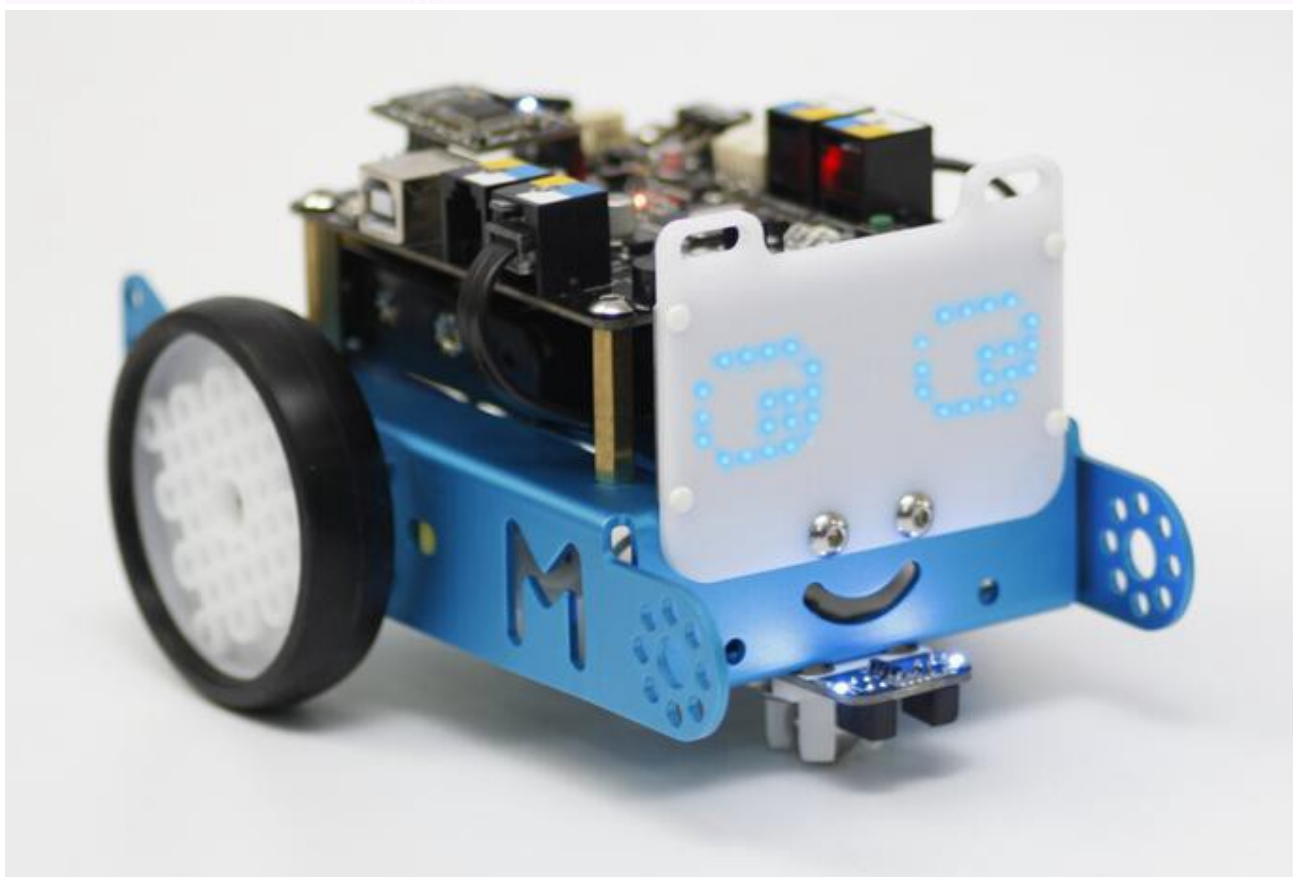
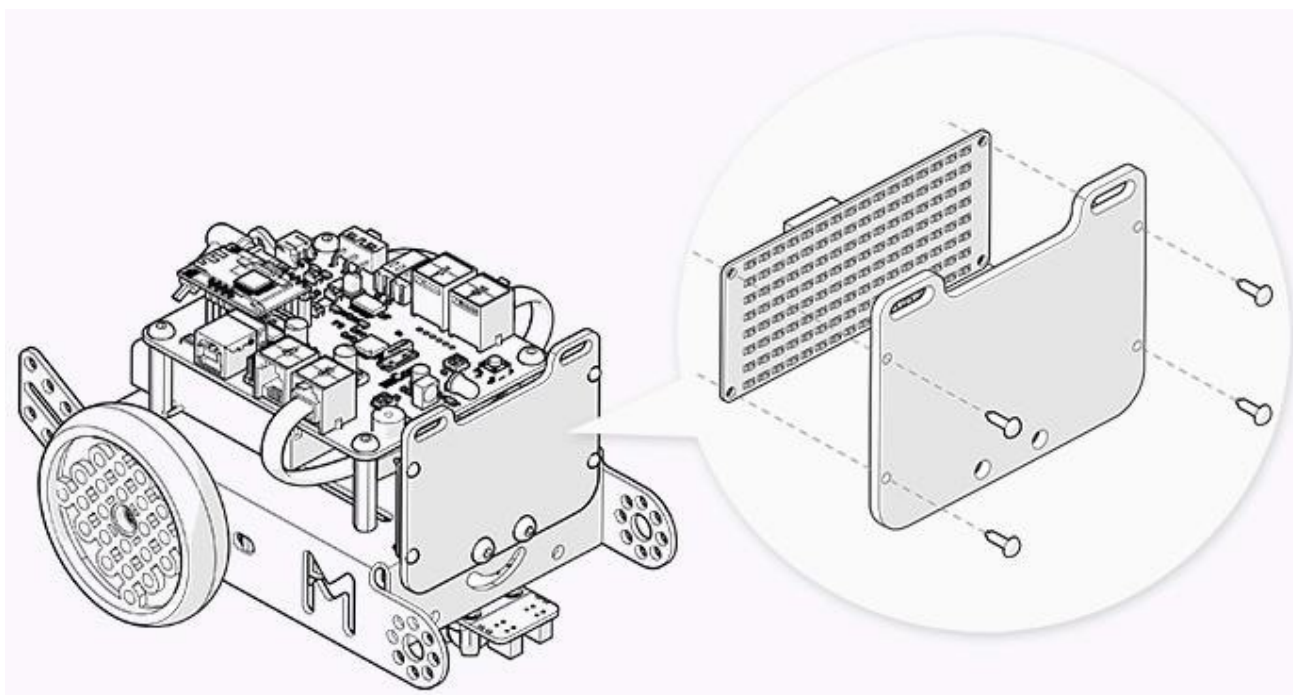


Frontal

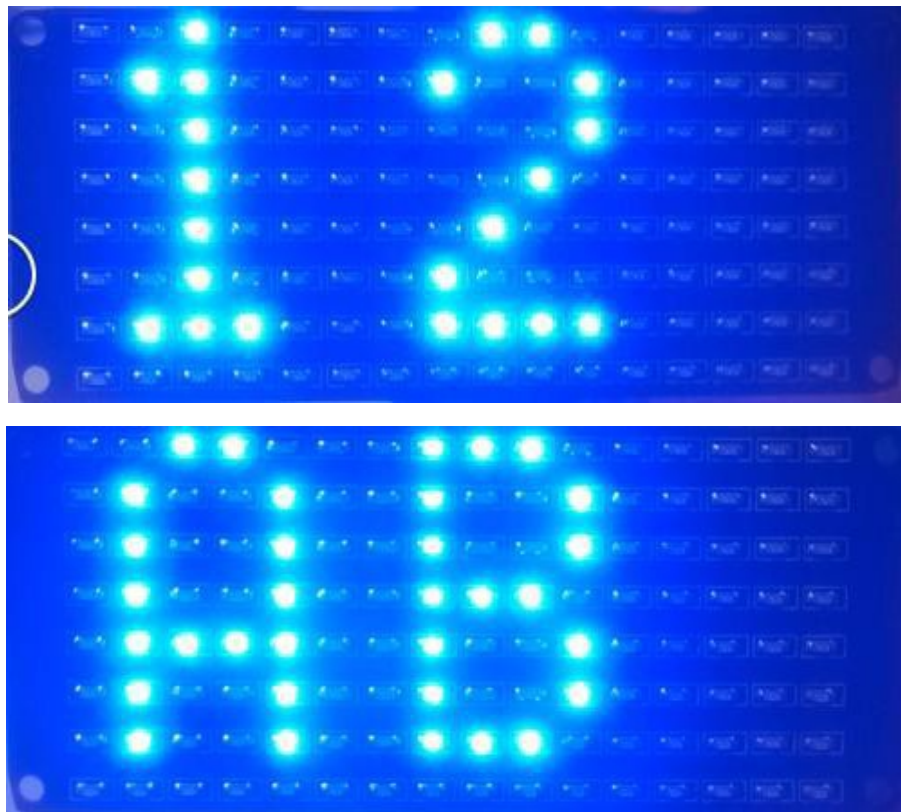


Posterior

## Montaje:



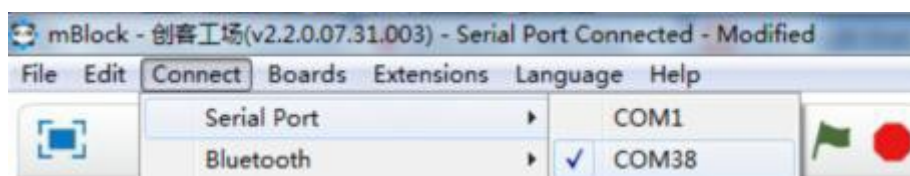
**Aplicación:** El módulo es una pantalla de matriz de 8X16 LED compuesto por 8 lámparas LED azules verticales y 16 horizontales. Durante el encendido o apagado de las lámparas LED, podrán mostrarse ciertas figuras básicas, letras y patrones simples. Véase el siguiente diagrama:



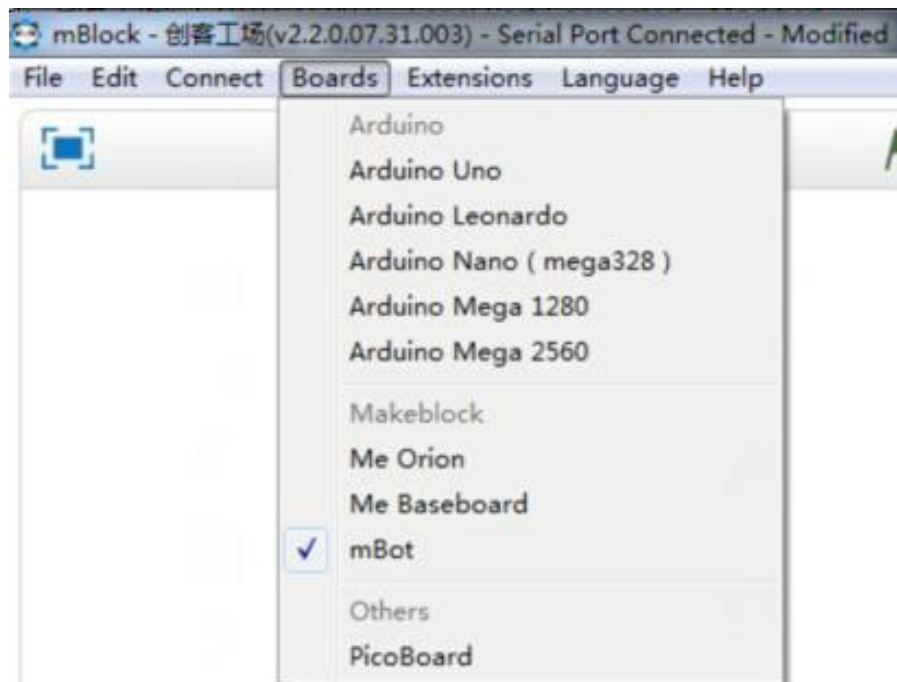
## 2. Instrucciones

### 2.1 mBlock programación

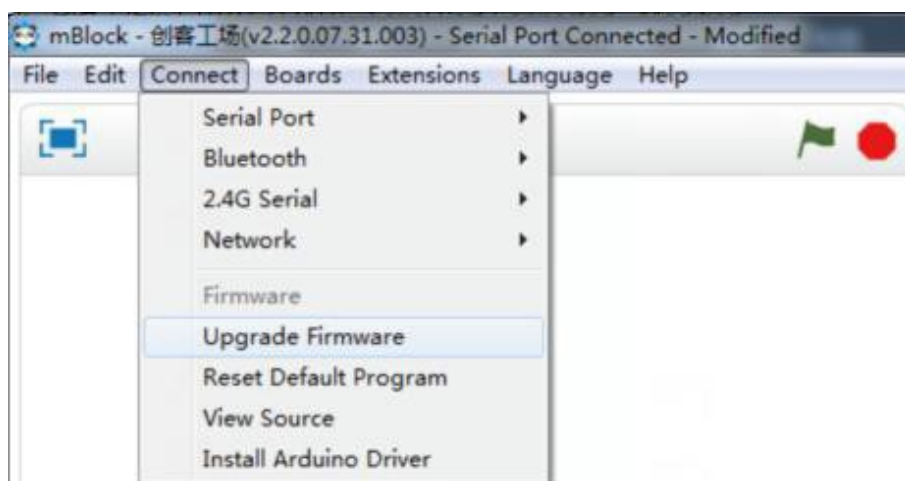
- **Paso 1** Después de conectar, mediante cable USB y encender el Mbot, tendrá que elegir el puerto de serie en primer lugar.



- **Paso 2**, elige el tipo de placa



- **Paso 3, descargar el Firmware**



- **Paso 4, arrastre los bloques de script a la matriz LED**





## Introducción bloques de script

### 1) Caracteres de entrada



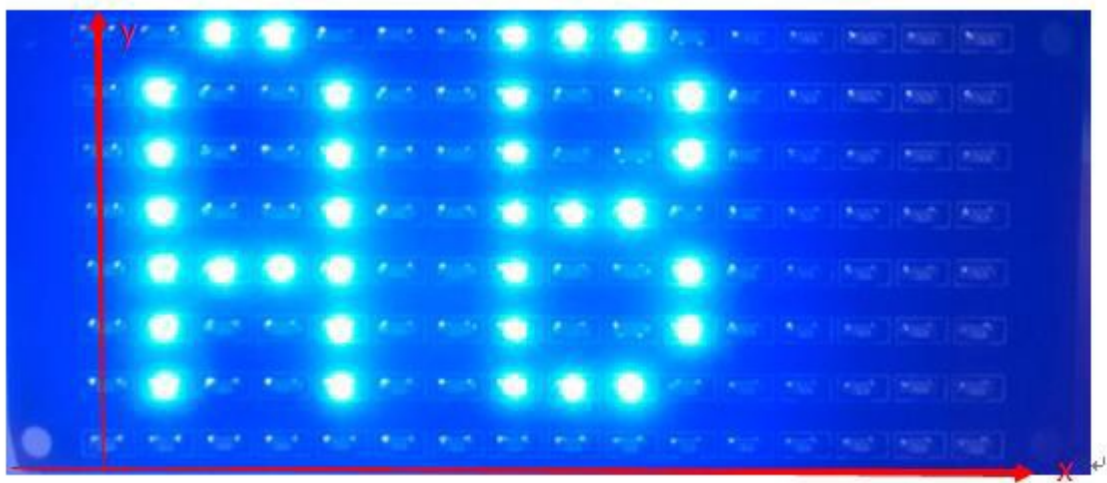
#### Parámetro:

“Port1” – Elige el Puerto correspondiente al que ha conectado la matriz LED

“x:0” – Establece la coordenada x

“y:0” – Establece la coordenada y.

“caracteres: ” – Establece los caracteres a mostrar mostrados como Hello



### 2) Mostrar hora



#### Parámetro:

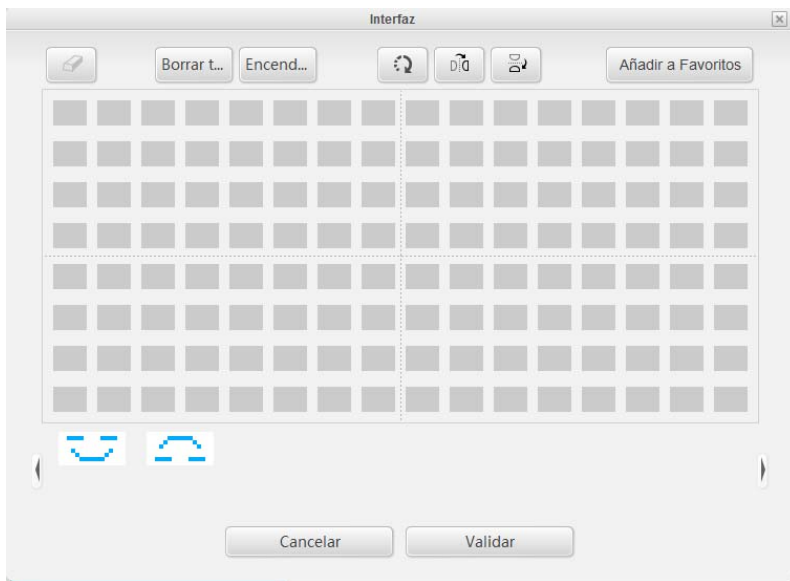
“Port1” – Elige el Puerto correspondiente al que ha conectado la matriz LED.

hora&min – Los caracteres serán mostrados con la hora actual

### 3) Patrones y caracteres definidos por el usuario



Haga clic en el área representada y el panel indicado se mostrará en la pantalla



Seleccione los leds que desee activar y haga clic, se mostrará en la pantalla

#### 4) Mostrar valores y datos de los sensores



Por ejemplo si se quiere que se muestre el valor de un sensor, por ejemplo la distancia



## 2.2 Programación de Arduino IDE

Consejos: Cómo ver el código fuente del módulo matriz de LED

Primero, abra el software mBlock, luego seleccione "Extensions->Manage ->mBot->view source code" (Extensiones, administrar, mbot, ver código Fuente)

### Biblioteca de funciones

Nombre de la función	Descripción	Propiedad
<a href="#">MeLEDMatrix</a>	Constructor	Función pública
<a href="#">clearScreen</a>	Función de borrar pantalla	Función pública
<a href="#">setBrightness</a>	Función de ajuste de brillo	Función pública
<a href="#">setColorIndex</a>	Ajuste de índice de color (1: Iluminación normal, 0: sin color);	Función pública

<a href="#">drawBitmap</a>	Mostrar función de mapa de bits (diagram cuadriculado)	Función pública
<a href="#">drawStr</a>	Mostrar función de cadena de caracteres	Función Pública
<a href="#">showClock</a>	Mostrar función de reloj	Función pública
<a href="#">showStr</a>	Mostrar función de cadena de caracteres	Función privada

## Introducción de funciones de desarrollo

### 1. MeLEDMatrix Constructor 1

Nombre de función	MeLEDMatrix
Objeto de función	MeLEDMatrix ();
Descripción de desarrollo	constructor vacío
Parámetros de entrada	N/A
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	N/A

[Volver a la biblioteca de funciones→](#)

### 2. MeLEDMatrix Constructor 2

Nombre de función	MeLEDMatrix
Objeto de función	MeLEDMatrix (uint8_t port);
Descripción de desarrollo	Constructor. Inicializar el módulo de acuerdo con los parámetros del puerto;
Parámetros de entrada	Puerto: Módulo control de puerto. Rango de valores: PORT_1 ~ PORT_8



Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeByte ( ); // Función de escritura de un byte; setBrightness ( ); // Función de ajuste de brillo; clearScreen ( ); // Función de borrado de pantalla;

[Volver a la biblioteca de funciones→](#)

### 3. MeLEDMatrix Constructor 3

Nombre de función	MeLEDMatrix
Objeto de función	MeLEDMatrix (uint8_t SCK_Pin,uint8_t DIN_Pin);
Descripción de desarrollo	Constructor. Inicializar módulo de acuerdo con los parámetros de pin;
Parámetros de entrada	SCK_Pin: Pin de módulo de control de línea de reloj  DIN_Pin: Pin de módulo de control de línea de datos
Valor devuelto	Nil
Prerequisito	Nil
Función de llamada	writeByte ( ); // función de escritura de un byte; setBrightness ( ); // Función de ajuste de brillo; clearScreen ( ); // Función de borrado de pantalla;

[Volver a la biblioteca de funciones→](#)

### 4. clearScreen (limpiar pantalla)

Nombre de función	clearScreen
Objeto función	void clearScreen ();
Descripción de desarrollo	Función de limpieza de pantalla
Parámetros de entrada	N/A

Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeBytesToAddress ( ); // Escribe algunos bytes en la dirección especificada;

[Volver a la biblioteca de funciones→](#)

#### 5. setBrightness (ajustar brillo)

Nombre de función	setBrightness
Objeto de	void setBrightness (uint8_tBright);
Descripción de desarrollo	Función de ajuste de brillo
Parámetros de entrada	<p>Bright: Parámetros de brillo. Rango de valores:</p> <p style="text-align: center;"> Brightness_0 // El más oscuro (apagar)  Brightness_1  Brightness_2  Brightness_3  Brightness_4  Brightness_5  Brightness_6  Brightness_7  Brightness_8 // El más brillante </p>
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeByte ( ); // función de escritura de un solo byte;

[Volver a la biblioteca de funciones→](#)

#### 6. setColorIndex (ajustar color)

Nombre de función	setColorIndex
Objeto de función	void setColorIndex (bool Color_Number);
Descripción de desarrollo	Color index show setting (1: Normally lighting, 0: Color negation);
Parámetros de entrada	Color_Number: Parámetros de color. 1: Mostrar diagrama con lámparas de iluminación; 0: Mostrar con las lámparas desactivadas;

	Observaciones: <1> El parámetro de color del Sistema de LED por defecto en pantalla de cuadrícula es 1 (mostrando el diagrama con las lámparas de iluminación) <2> clearScreen( ), después de llamar a la función de borrado de pantalla, el parámetro de color se reseteará al 1;
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	N/A

[Volver a la biblioteca de funciones→](#)

## 7. drawBitmap (dibujar mapa de bits)

Nombre de función	drawBitmap
Objeto de función	void drawBitmap (uint8_t x, uint8_t y, uint8_t Bitmap_Width, uint8_t *Bitmap);
Descripción de desarrollo	Mostrar la función de mapa de bits (diagrama cuadrícula. La altura del mapa de bits se fija en 8 (píxeles);
Parámetros de entrada	x: Rango de valores de la <b>esquina superior izquierda</b> de la coordenada x del mapa de bits (diagrama de red): 0 ~ 15; y: Rango de valores de la <b>esquina superior izquierda</b> de la coordenada y del mapa de bits(diagrama de red): 0 ~ 7; Bitmap_Width: Ancho (píxeles) del mapa de bits (diagrama de red) : Bitmap: nombre de la matriz del código de mapa de bits de red (buffer area);
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeBytesToAddress ( );      Escribe algunos bytes en la dirección especificada;

[Volver a la biblioteca de funciones→](#)

## 8. drawStr

Nombre de función	drawStr
Objeto de función	void drawStr (int16_t X_position, int8_t Y_position, const char *str);
Descripción de desarrollo	Función de muestra de las cadenas de caracteres. Provisionalmente, sólo es capaz de mostrar las letras mayúsculas y minúsculas, cifras y espacios en blanco, y el carácter desconocido se dejará en blanco;

	La altura de la cadena de caracteres se fijará en 8 (píxeles);
Parámetros de entrada	<p>X_position: Esquina <b>inferior izquierda</b> x-coordina la primera cadena de caracteres. Rango de valores: El valor más pequeño se decide por la longitud de las cadenas de caracteres, Valor más grande: 16</p> <p>Y_position: Bottom <b>left corner</b> Y-coordinate of first character of strings Value range: -1 ~ 15;</p> <p>str: La dirección inicial de cadenas de caracteres en buffer área(matriz) que se muestra;</p>
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	showStr ();            // función de muestra de cadena de caracteres, función privada

[Volver a la biblioteca de funciones→](#)

#### 9. showClock (mostrar reloj)

Nombre de función	showClock
Objeto de función	void showClock (uint8_t hour, uint8_t minute, bool point_flag = PointOn);
Descripción de desarrollo	Función de muestra de reloj
Parámetros de entrada	<p>hour: Hora;</p> <p>minute: Minuto;</p> <p>point_flag: Dos puntos muestran la etiqueta entre el reloj (colon) point_flag = PointOn (point_flag = 1) indica qué dos puntos son mostrados; point_flag = PointOff (point_flag = 0) indica qué dos puntos no son mostrados</p>
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeBytesToAddress ( );        // Escribe algunos bytes a la dirección especificada;

[Volver a la biblioteca de funciones→](#)

Código de ejemplo:

```
#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()
{
    /***** Ejemplo1 *****/
    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);
        }
    }
    /***** Ejemplo 1 *****/

    /***** Ejemplo 2 *****/
    Matrix_1.setColorIndex(0);
    Matrix_1.drawBitmap(0, 0, sizeof(Bitmap_Heart),
```

```
itmap_Heart); for(uint8_t k=0; k<3; k++)
```

-1 ~ 15;

```
{
    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);
    }
}
```

\*\*\*\*\* Ejemplo 2\*\*\*\*\*/

/\*\*\*\*\* Ejemplo

```
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);
}
```

\*\*\*\*\* Ejemplo  
3\*\*\*\*\*/

/\*\*\*\*\* Ejemplo 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);
}
```

\*\*\*\*\* Ejemplo  
3\*\*\*\*\*/

```
}
```