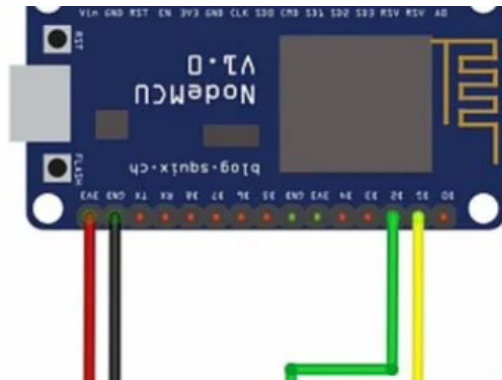


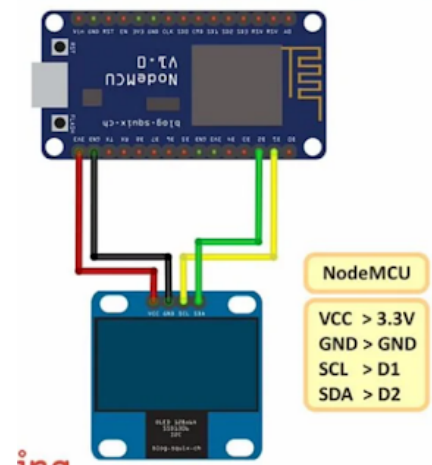
NodeMCU: 10_ESPNOW: inciso, cómo dibujar en pantalla OLED 128x64

agrportfolioeducativo.blogspot.com/2020/04/nodemcu-10espnow-inciso-como-dibujar-en.html



Utilizamos la pequeña pantalla OLED 128x64 para poder mostrar información del dispositivo NodeMCU. Podremos dibujar líneas, figuras y logotipos.

Conexionado



Programa para averiguar la dirección I2C

Lo primero que tenemos que hacer es averiguar la dirección del dispositivo para poder comunicarnos con él a través del I2C. Lo conseguiremos con el siguiente programa:

```
#include <Wire.h>
```

```
void setup()
```

```
{
```

```
Wire.begin();
```

```
Serial.begin(9600);
```

```
while (!Serial); // Espera al monitor serie
```

```
Serial.println("\nEscaner I2C");
```

```
}
```

```
void loop()
```

```
{
```

```
byte error, address;
```

```
int nDevices;

Serial.println("Escaneando...");

nDevices = 0;
for(address = 1; address < 127; address++ )
{
  // El escaner i2c utiliza el valor devuelto por la instrucción
  // Wire.endTransmission para reconocer la dirección a la que
  // está conectado cada dispositivo.
  Wire.beginTransmission(address);
  error = Wire.endTransmission();

  if (error == 0)
  {
    Serial.print("Dispositivo I2C encontrado en la direccion 0x");
    if (address<16)
    Serial.print("0");
    Serial.print(address,HEX);
    Serial.println(" !");

    nDevices++;
  }
  else if (error==4)
  {
    Serial.print("Error desconocido en la direccion 0x");
    if (address<16)
    Serial.print("0");
    Serial.println(address,HEX);
  }
}

if (nDevices == 0)
  Serial.println("Ningun dispositivo I2C encontrado\n");
else
  Serial.println("Hecho\n");

delay(5000); // espera 5 segundos para el siguiente escaneo
}
```

[view raw NODEMCU averiguar I2C OLED.ino](#) hosted with ♥ by [GitHub](#)

Programa que dibuja logo, líneas , círculos, texto

El siguiente programa tiene las órdenes necesarias para dibujar los elementos descritos. Muchas de ellas están comentadas, siendo solo visibles las primeras en pantalla. Para ver cómo funcionan, descomentar y comentar cada parte.

```
/*
```

```
ESP8266 - Ejemplo de uso de pantalla OLED 128x64 monocroma por Dani No www.esploradores.com
```

```
Este ejemplo utiliza el núcleo de la librería gráfica de Adafruit GFX y
```

```
la librería SSD1306 específica para pantallas monocromas 128x64 y 128x32.
```

```
Estas librerías deben estar instaladas para su funcionamiento.
```

```
Para instalarlas se debe seleccionar:
```

```
-Programa -> Incluir librería -> Gestionar Librerías -> Adafruit GFX Library
```

```
-Programa -> Incluir librería -> Gestionar Librerías -> Adafruit SSD1306
```

```
Cuando se instala la librería SSD1306 está configurada por defecto para pantallas 128x32.
```

```
Al utilizarla para una pantalla 128x64 es preciso reconfigurarla.
```

```
Para hacerlo debe abrir el fichero Adafruit_SSD1306.h con un editor y
```

```
descomentar la línea 69 (// #define SSD1306_128_64 ---> #define SSD1306_128_64) y
```

```
comentar la línea 70 (#define SSD1306_128_32 ---> //define SSD1306_128_32).
```

```
Así mismo se debe verificar en la línea 51 que la dirección de I2C de la pantalla es correcta
```

```
(#define SSD1306_I2C_ADDRESS 0x3C // 128x64 0x3D (default) or 0x3C (if SA0 is grounded))
```

```
Por último, también se debe verificar que en la línea 108 de este sketch
```

```
la dirección I2C de la pantalla también es correcta.
```

```
Modificado por Aurelio Gallardo - día 16-Abril-2020
```

```
*/
```

```
/*
```

```
* https://www.diarioelectronicohoy.com/blog/pantalla-oled-0-96 (descripción de las órdenes)
```

```
*
```

```
*/
```

```
#include <SPI.h>
```

```
#include <Wire.h>
```

```
#include <Adafruit_GFX.h> //Núcleo de la librería gráfica.
```

```
#include <Adafruit_SSD1306.h> //Librería para pantallas OLED monocromas de 128x64 y 128x32
```

```
#include <Fonts/FreeMonoBoldOblique12pt7b.h>
```

```
#include <Fonts/FreeSerifBold9pt7b.h> // Biblioteca de fuentes: https://learn.adafruit.com/adafruit-gfx-graphics-library/using-fonts
```

```
//
```

```
// IMPORTANTE: definición de pantalla, y RESET que funciona, sin tocar librería adafruit_SSD1306
```

```
//
```

```

#define SCREEN_WIDTH 128 // OLED display width, in pixels

#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

#define OLED_RESET 0 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

//
=====

// #define OLED_RESET 0

// Adafruit_SSD1306 display(OLED_RESET);

#define NUMFLAKES 10

#define XPOS 0

#define YPOS 1

#define DELTAY 2

int x = 0;

int y = 0;

const unsigned char PROGMEM logo [] = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xF0, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFC, 0x00, 0x00, 0x07, 0xF0, 0x00, 0x03, 0xF8, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFC, 0x00, 0x00, 0x07, 0xC0, 0x00, 0x00, 0xF8, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFE, 0x00, 0x00, 0x07, 0x80, 0x00, 0x00, 0x78, 0x00, 0x00,
0x00, 0x00, 0x00, 0x7E, 0x00, 0xFF, 0xFE, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00,
0x00, 0x00, 0x00, 0xFF, 0x01, 0xFF, 0xFE, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x00, 0x00, 0x03, 0xFF, 0xC0, 0xFF, 0xFE, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x00, 0x00, 0x03, 0xFF, 0xF0, 0xFF, 0xFF, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x00, 0x00, 0x03, 0xFF, 0xF0, 0x7F, 0xFF, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x03, 0xFC, 0x07, 0xFF, 0xF8, 0x3F, 0xFF, 0x80, 0x00, 0x06, 0x03, 0x00, 0x30, 0x18, 0x00, 0x00,
0x03, 0xFE, 0x07, 0xFF, 0xF8, 0x1F, 0xFF, 0x80, 0x00, 0x06, 0x1F, 0xE1, 0xFE, 0x18, 0x00, 0x00,
0x0F, 0xFF, 0x07, 0xFF, 0xF8, 0x07, 0xFF, 0x80, 0x00, 0x06, 0x30, 0x33, 0x03, 0x18, 0x00, 0x00,
0x0F, 0xFF, 0xC7, 0xFF, 0xFC, 0xE3, 0xFF, 0xC0, 0x00, 0x06, 0x20, 0x12, 0x01, 0x18, 0x00, 0x00,
0x1F, 0xFF, 0xC3, 0xFF, 0xFC, 0xE0, 0x7F, 0xC0, 0x00, 0x06, 0x60, 0x0C, 0x01, 0x98, 0x00, 0x00,
0x1F, 0xFF, 0xC1, 0xFF, 0xFC, 0xFC, 0x1F, 0xC0, 0x00, 0x06, 0x63, 0x0C, 0x31, 0x98, 0x00, 0x00,
0x1F, 0xFF, 0xE0, 0xFF, 0xFE, 0x7F, 0x03, 0xE0, 0x00, 0x06, 0x60, 0x0C, 0x01, 0x98, 0x00, 0x00,

```

0x1F, 0xFF, 0xE0, 0x7F, 0xFE, 0x7F, 0xC0, 0xE0, 0x00, 0x06, 0x20, 0x12, 0x01, 0x18, 0x00, 0x00,
0x0F, 0xFF, 0xE0, 0x1F, 0xFE, 0x7F, 0xF8, 0x40, 0x00, 0x06, 0x30, 0x33, 0x03, 0x18, 0x00, 0x00,
0x0F, 0xFF, 0xF7, 0x8F, 0xFF, 0x3F, 0xFC, 0x00, 0x00, 0x06, 0x1C, 0xE1, 0xCE, 0x18, 0x00, 0x00,
0x0F, 0xFF, 0xF7, 0xC0, 0xFF, 0x3F, 0xFF, 0x80, 0x00, 0x06, 0x07, 0x80, 0x78, 0x18, 0x00, 0x00,
0x07, 0xFF, 0xF7, 0xF8, 0x1F, 0x3F, 0xFF, 0xC0, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x03, 0xFF, 0xFB, 0xFE, 0x0F, 0x1F, 0xFF, 0xE0, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x00, 0xFF, 0xFB, 0xFF, 0x03, 0x9F, 0xFF, 0xF0, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x08, 0xFF, 0xFB, 0xFF, 0xE0, 0x1F, 0xFF, 0xF8, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x0E, 0x0F, 0xFD, 0xFF, 0xF8, 0x0F, 0xFF, 0xF8, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
0x0F, 0x03, 0xFD, 0xFF, 0xFF, 0x0F, 0xFF, 0xFC, 0x00, 0x07, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00,
0x0F, 0xF0, 0xFD, 0xFF, 0xFF, 0x0F, 0xFF, 0xFC, 0x00, 0x07, 0x80, 0x00, 0x00, 0x78, 0x00, 0x00,
0x07, 0xF8, 0x1C, 0xFF, 0xFF, 0x8F, 0xFF, 0xFC, 0x00, 0x07, 0xC0, 0x00, 0x00, 0xF8, 0x00, 0x00,
0x07, 0xFF, 0x04, 0xFF, 0xFF, 0xC7, 0xFF, 0xF8, 0x00, 0x07, 0xF8, 0x00, 0x07, 0xF8, 0x00, 0x00,
0x07, 0xFF, 0xE0, 0xFF, 0xFF, 0xE7, 0xFF, 0xF0, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00,
0x03, 0xFF, 0xE0, 0xFF, 0xFF, 0xE7, 0xFF, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00,
0x03, 0xFF, 0xFE, 0x7F, 0xFF, 0xF3, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0xFF, 0xFE, 0x7F, 0xFF, 0xF3, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xFF, 0xFE, 0x7F, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xFF, 0xFF, 0x3F, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xFF, 0xFF, 0x3F, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xFF, 0xFF, 0x3F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x06, 0x00, 0x00, 0x00, 0x00,
0x00, 0xFF, 0xFF, 0x9F, 0xC0, 0x00, 0x07, 0x00, 0x00, 0x03, 0xC0, 0x0F, 0x00, 0x00, 0x00, 0x00,
0x00, 0xFF, 0xFF, 0x9C, 0x00, 0x00, 0x1F, 0x80, 0x00, 0x03, 0xC0, 0x0F, 0x00, 0x00, 0x00, 0x00,
0x00, 0xFF, 0xFF, 0x80, 0x00, 0x00, 0x3F, 0x80, 0x00, 0x01, 0x80, 0x06, 0x00, 0x00, 0x00, 0x00,
0x00, 0x7F, 0xFF, 0x00, 0x00, 0x00, 0x63, 0x80, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x7F, 0xF8, 0x00, 0x00, 0x00, 0xE2, 0x00, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x7F, 0x80, 0x00, 0x00, 0x00, 0xE0, 0x0F, 0x8F, 0xE7, 0x0E, 0x1C, 0x71, 0xC7, 0x39, 0xC0,
0x00, 0x3C, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x1F, 0xBF, 0xFF, 0x0E, 0x7D, 0xF3, 0x9F, 0xFF, 0xC0,
0x00, 0x20, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x3B, 0xBF, 0x7F, 0x7F, 0xFD, 0xF3, 0x9F, 0xFF, 0xC0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x3F, 0x8E, 0x06, 0x1C, 0x18, 0x63, 0x87, 0xBD, 0xC0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7E, 0x7F, 0x1C, 0x0E, 0x1C, 0x38, 0xE3, 0x0E, 0x73, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3E, 0x7C, 0x1C, 0x0E, 0x1C, 0x38, 0xE7, 0x0E, 0x73, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x78, 0x1C, 0x0E, 0x18, 0x38, 0xE7, 0x0E, 0x73, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0x71, 0x9C, 0x0C, 0xF9, 0xB3, 0xEF, 0x6E, 0x73, 0xB0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x0E, 0x73, 0xB8, 0x1F, 0xFF, 0xFF, 0xFF, 0xFC, 0xE3, 0xE0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x1C, 0x1C, 0x7F, 0x38, 0x1F, 0x3E, 0x7D, 0xFF, 0xDC, 0xE7, 0xC0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x3C, 0x7E, 0x38, 0x1E, 0x3C, 0x79, 0xE7, 0x1C, 0xE3, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};

void setup() {
display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Inicializa pantalla con en la dirección 0x3D para la conexión I2C.
}

void loop() {

display.fillScreen(0); //Limpiamos la pantalla

// PUNTO Corriendo --> variación con círculos
display.drawPixel(x,y,1); //Dibujo un pixel en pantalla
display.drawCircle(x,y,10,1); // O bien un círculo
display.fillCircle(x,y,10,1); // relleno

// Caracteres
// display.setFont(); //Fuente por defecto -si no la hemos cambiado no es necesario seleccionarla
// display.setFont(&FreeSansBoldOblique12pt7b); // Especificada antes al cargar el programa
// display.drawChar(x,y,'a',1,0,4); //Tamaños de 1 a 8. Los tamaños típicos son 1, 2 o 4
// drawChar (uint16_t x, uint16_t y, char c, uint16_t color, uint16_t bg, uint8_t tamaño)

// Un texto en pantalla
// display.setFont(&FreeMonoBoldOblique12pt7b); // Si no se quiere cambiar la fuente por defecto, ya sabes...
/*
display.setFont(&FreeSerifBold9pt7b);
display.setTextSize(2); // Tamaño
display.setTextColor(1,0); // Color (1,0) ó 0,1 invertido
display.setCursor(x,y); // Posiciones x e y
display.println("Hola, soy Aurelio"); */

// Logo
// display.drawBitmap(0,0,logo,128,64,1); // drawBitmap (int16_t x, int16_t y, uint8_t * mapa de bits, int16_t w, int16_t h,
// uint16_t color)
// https://www.brainy-bits.com/create-arduino-array-from-pictures/
x=x+10;

if (x>=128) {y=y+5; x=0;} // Si se acaba en x, entonces vuelve a la y=0
if (y>=64) {y=0;} // Si se acaba la y, empieza
// display.drawPixel(0,0,1); //Esquina superior izquierda
display.display(); //Refrescamos la pantalla para visualizarlos

```

```
delay(50);
```

```
}
```

[view raw NODEMECU mi ejemplo OLED.ino](#) hosted with ♥ by [GitHub](#)

Programas para codificar imágenes (monocromas)

Usaremos dos programas:

1. **GIMP**. Creamos una nueva imagen de tamaño 128x64 (igual que la pantalla) y, con niveles, intento que salga en blanco y negro. Exporto como BMP
2. Utilizo el programa **LCDAssistant**. Este programa consigue codificar la imagen. En la imagen tenemos los parámetros (horizontal y 8 B)
3. Por fin, hacemos la exportación en el LCDAssistant y abrimos con editor de texto. Guardamos los códigos en el lugar correcto del programa en el IDE de ARDUINO

Los enlaces a los programas LCDAssistant y LCD-Image-Converter son:

1. <https://drive.google.com/drive/folders/1AGauP8sLKEI95Shf9WhaRomC3HW-rkzp?usp=sharing>
2. <https://lcd-assistant.software.informer.com/1.0/>
3. <https://www.riuson.com/lcd-image-converter>

En esta página te explica cómo hacerlo también con LCD-Image-Converter <https://randomnerdtutorials.com/esp8266-o-96-inch-oled-display-with-arduino-ide/>

