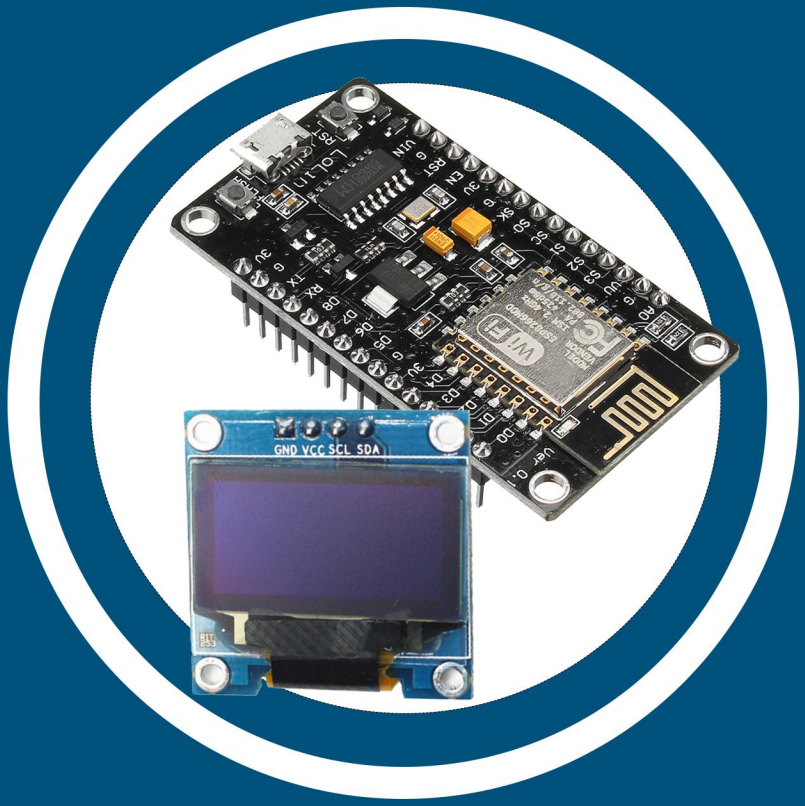
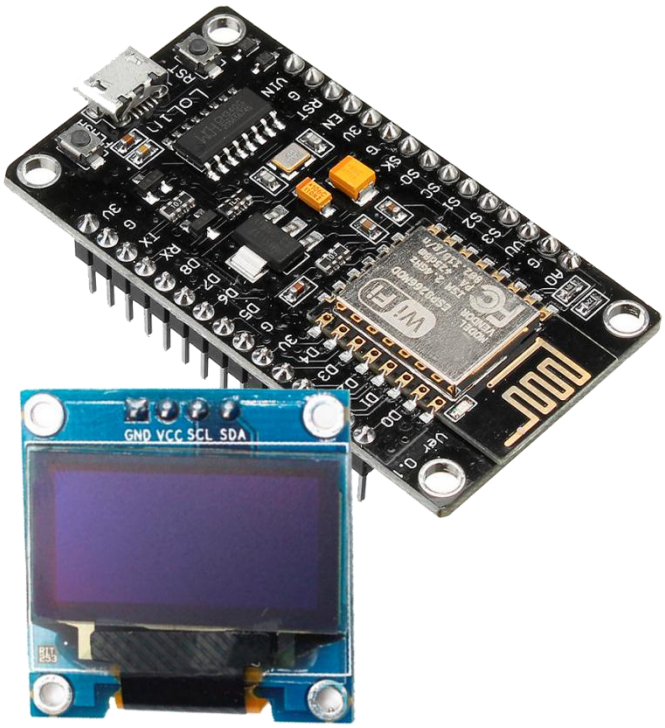


NodeMCU

+

OLED



NodeMCU+OLED

Escrito por

Franco Pozzetti

Profesor

Misael Cudek

Materia

Integración Tecnológica

Instituto Da Vinci – Octubre 2020

INDICE

INTRODUCCIÓN

Este trabajo fue desarrollado para la materia “Integración Tecnológica” dentro de la carrera de Analista de Sistemas junto a la coordinación del profesor Misael Cudek utilizando la placa de desarrollo NodeMCU junto al componente OLED para introducirme en el mundo de IoT y de las placas de desarrollo, entender su funcionamiento y lograr actividades interesantes.

IDEA

Nuestro profesor nos mencionó en la primera clase algunos ejercicios hechos con previos alumnos utilizando las placas de desarrollo Arduino. Nos comentaba que en las clases presenciales podía prestar a algunos las placas pero debido a la crisis sanitaria del año 2020 esto no era posible. Nos enseñó una alternativa que resulta ser más barato y nos introdujo a utilizar la placa NodeMCU.

Luego de la clase, mi interés creció en utilizar esta tecnología y realizar algunas pruebas para verificar su funcionamiento. Investigando un poco, encontré algunos proyectos que utilizan dicha placa junto a algún sensor o componente. Uno de estos componentes es una pantalla oled, el cual me interesó utilizar y probar no solo para este proyecto sino también para utilizarlo en el futuro.

En este trabajo, voy a documentar todo lo que aprendí y describir los ejercicios que pude encontrar y realizar.

Internet Of Things (IOT)

Un término muy importante cuando hablamos de placas de desarrollo como Arduino o NodeMCU es la Internet de las cosas. Básicamente IOT se define como la **capacidad de interconectar objetos físicos comunes y corrientes con la Internet**. A partir de este concepto salen las ideas como la domótica que son los sistemas capaces de automatizar las funciones de un hogar como el manejo de las luces, temperatura, seguridad, etc.

Utilizando un NodeMCU, podemos conectarlo junto a un sensor o componente que luego el usuario puede administrarlo a través de alguna aplicación o interfaz. Cada vez que el usuario quiera cambiar el estado del componente, la aplicación o interfaz envía un mensaje al Node a través de la red, como puede ser el wifi del hogar. La placa recibe el mensaje y lo interpreta, generando el cambio en el componente.

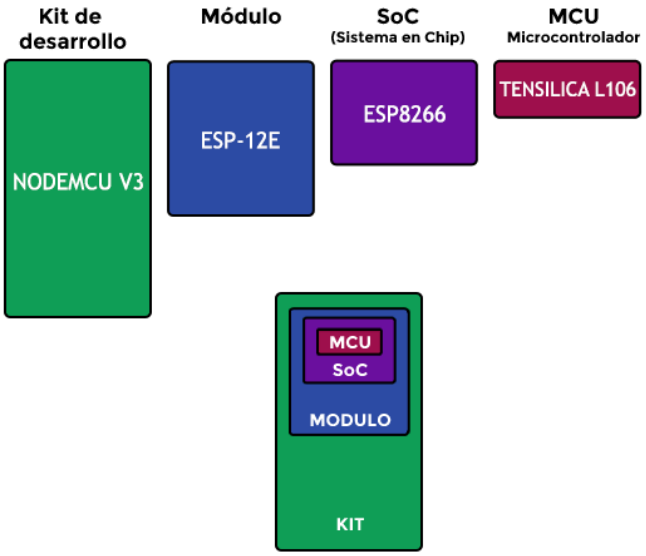
NodeMCU

El NodeMCU es una placa de desarrollo que permite programar un microcontrolador para realizar proyectos de IoT o robótica. Esta placa está conformada por 3 partes principales:

1) Microcontrolador: Se encarga de ejecutar todas las órdenes cargadas en la memoria.

2) SoC (System on a Chip): Es un chip que tiene todo integrado (mcu,wifi,ram,etc.) permitiendo que funcione de forma autónoma. No contiene memoria flash.

3) Modulo: Contiene la memoria flash y una antena para reforzar la señal de conexión wifi. La memoria flash es donde se almacena nuestro código. SoC se comunica con esa memoria, obtiene las órdenes y el microcontrolador las ejecuta.



La principal ventaja del NodeMCU sobre otras placas como Arduino es su precio barato y accesible junto a la ventaja de tener un módulo wifi. De esta forma, resulta ser una buena opción para introducirse al mundo del desarrollo de placas de hardware y de IoT.

ITEMS NECESARIOS

Para realizar los ejercicios enseñados en este trabajo se necesita lo siguiente:

Hardware:

- NodeMCU
- Cables puente o jumper wires con pines hembra – hembra
- Display OLED 0.96 pulgadas de 128x64 pixeles
- Cable USB a micro USB

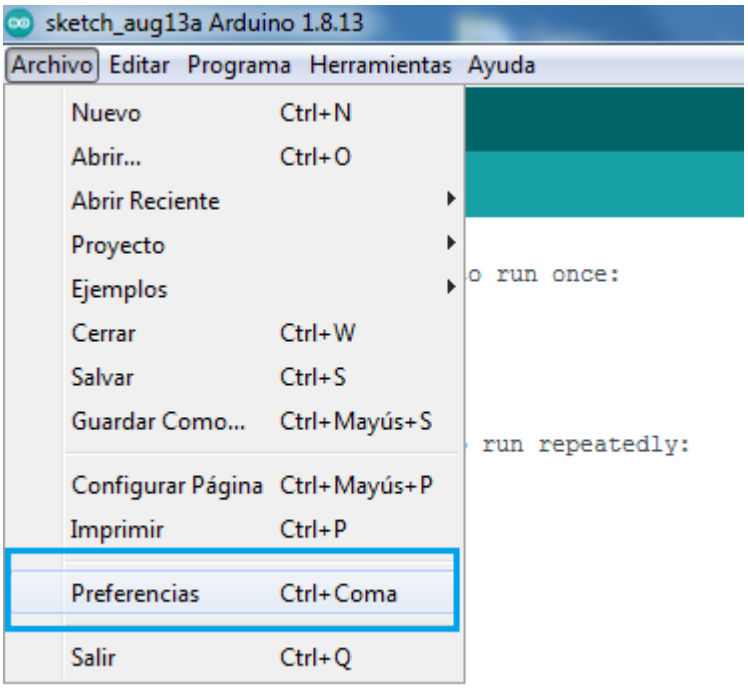
Software:

- Arduino IDE (https://www.arduino.cc/en/Main/Software)

Arduino IDE

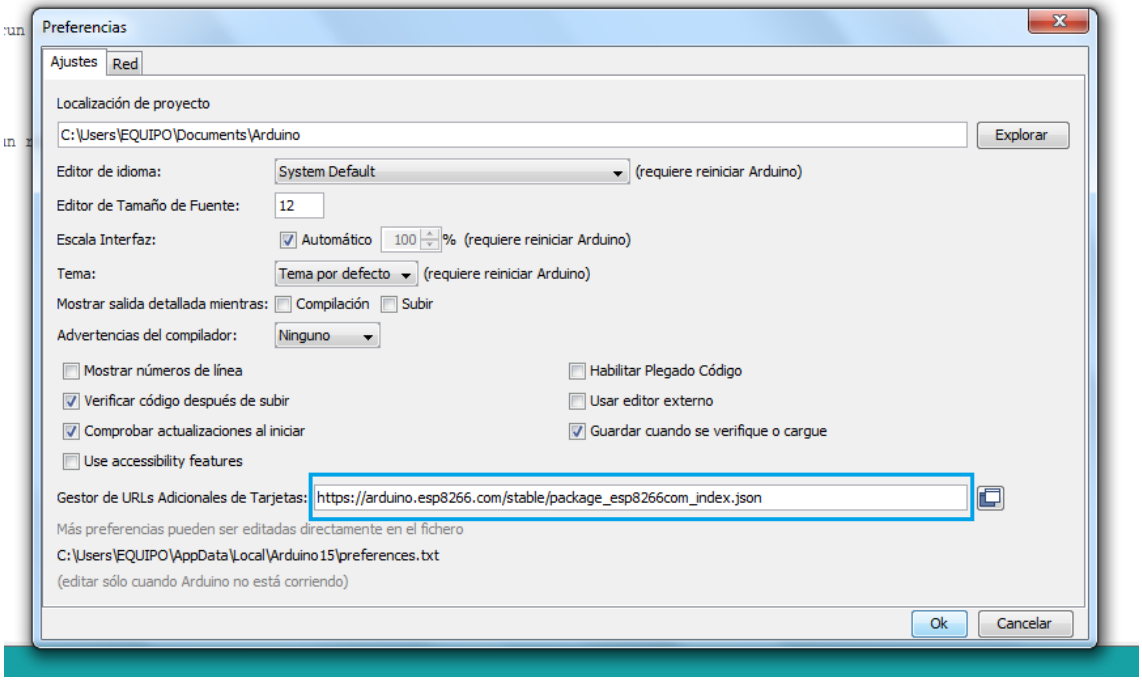
Arduino IDE es el programa que utilizaremos para escribir nuestro código, compilar el programa y subirlo a nuestro Node. Pero antes de realizar nuestra primera prueba hace falta destacar de que en principio el IDE no soporta placas con SoC esp8266. Para solucionar esto tenemos que configurar manualmente utilizando los pasos que se describirán a continuación.

1) Abrir el programa.
2) Seleccionar Archivo/Preferencias (File/Preferences en inglés).

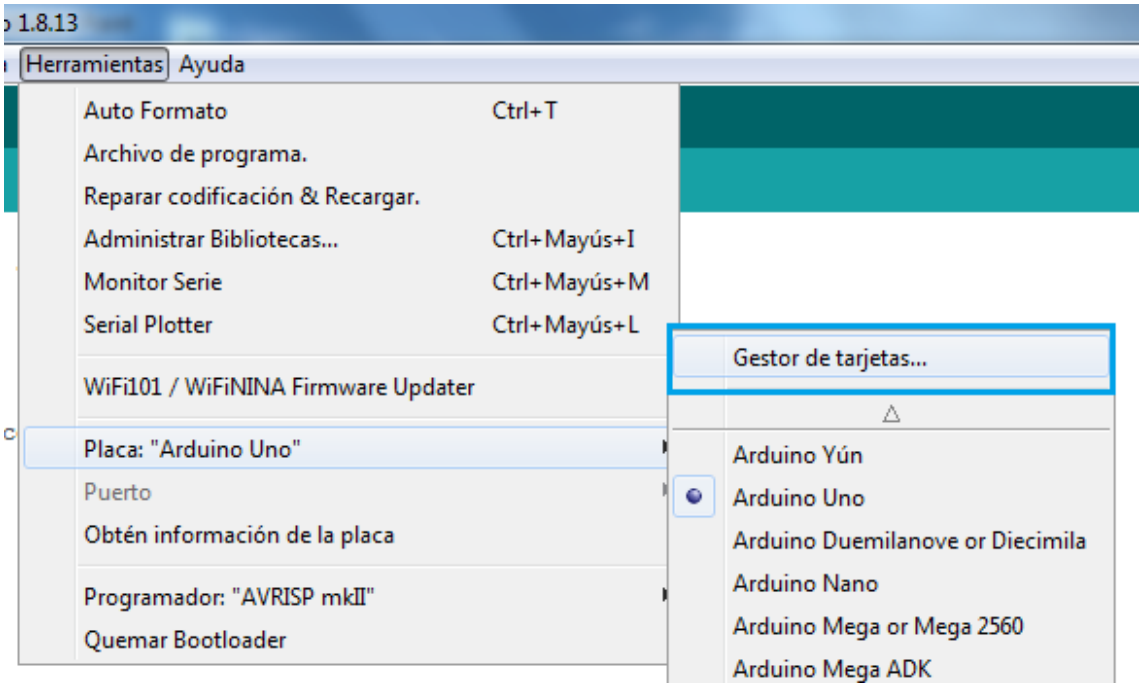


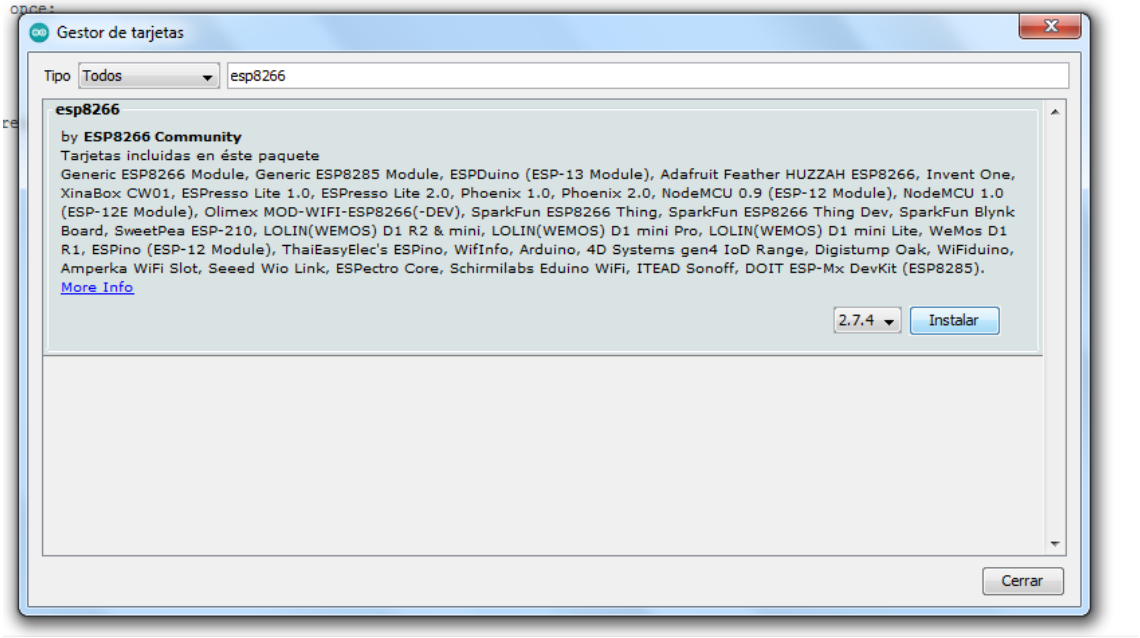
3) Dentro de la sección “Gestor de URLs adicionales de tarjetas” (Additional Boards Manager URL en inglés) tenemos que insertar el siguiente link: https://arduino.esp8266.com/stable/package_esp8266com_index.json

   Esto nos sirve para tener disponible la instalación del esp8266 en el gestor de tarjetas. Una vez ingresado el link, presionar ok.

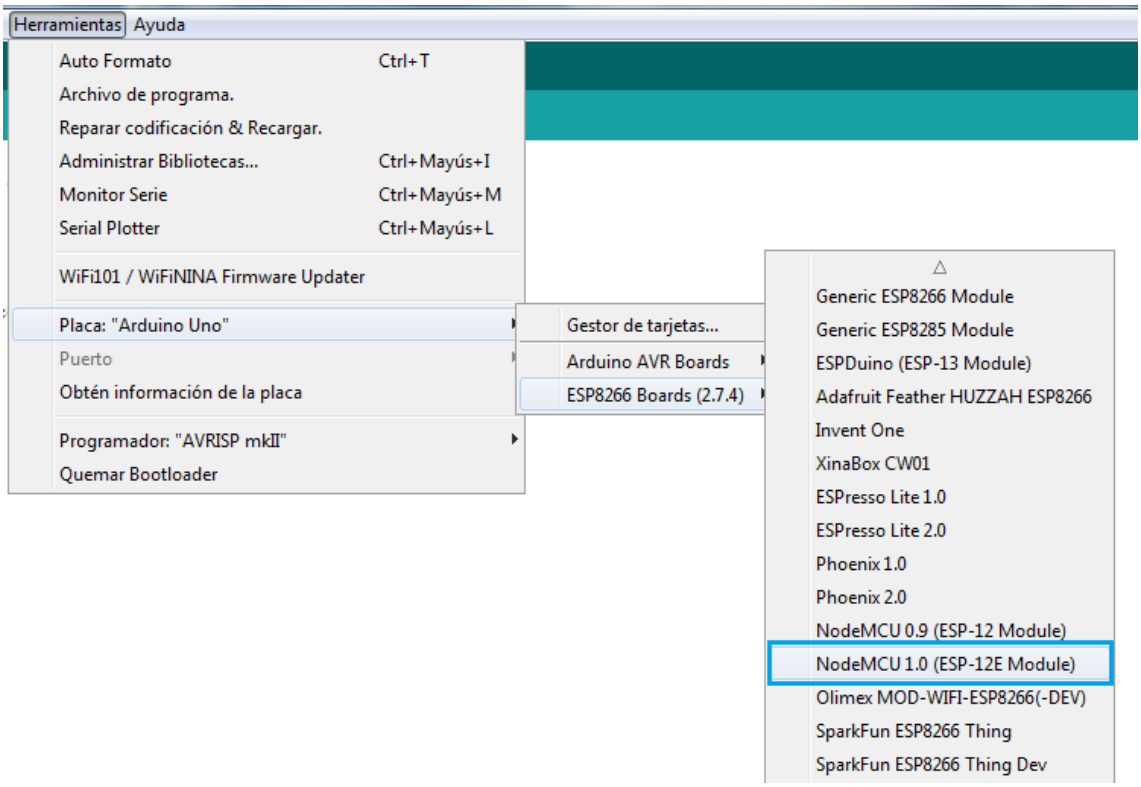


4) Seleccionar Herramientas/Placa/Gestor de tarjetas (Tools/Board/Boards Manager en inglés). Dentro de la nueva ventana, escribir “esp8266” y aparecerá una opción que no era posible que aparezca si no hubiéramos pasado por el paso 3. Presionar Instalar y, luego de terminar con la instalación, cerrar la ventana.

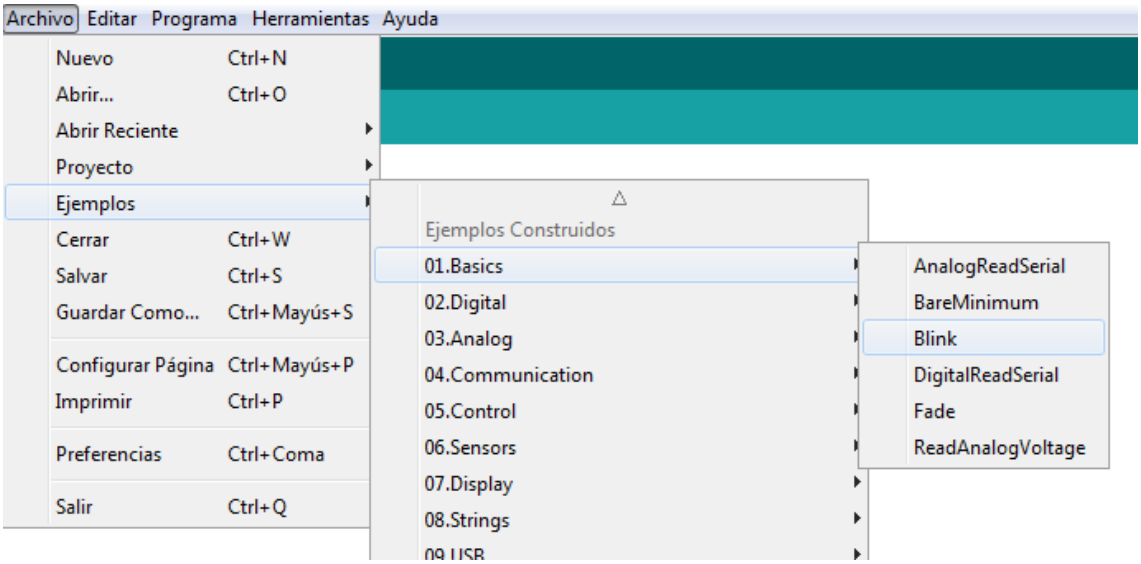




5) Volver a seleccionar Herramientas/Placa y ahora aparece la nueva opción “ESP8266 Boards”. Seleccionar dicha opción y buscaremos y seleccionaremos del listado la opción “NodeMCU 1.0 (ESP-12E Module)”.



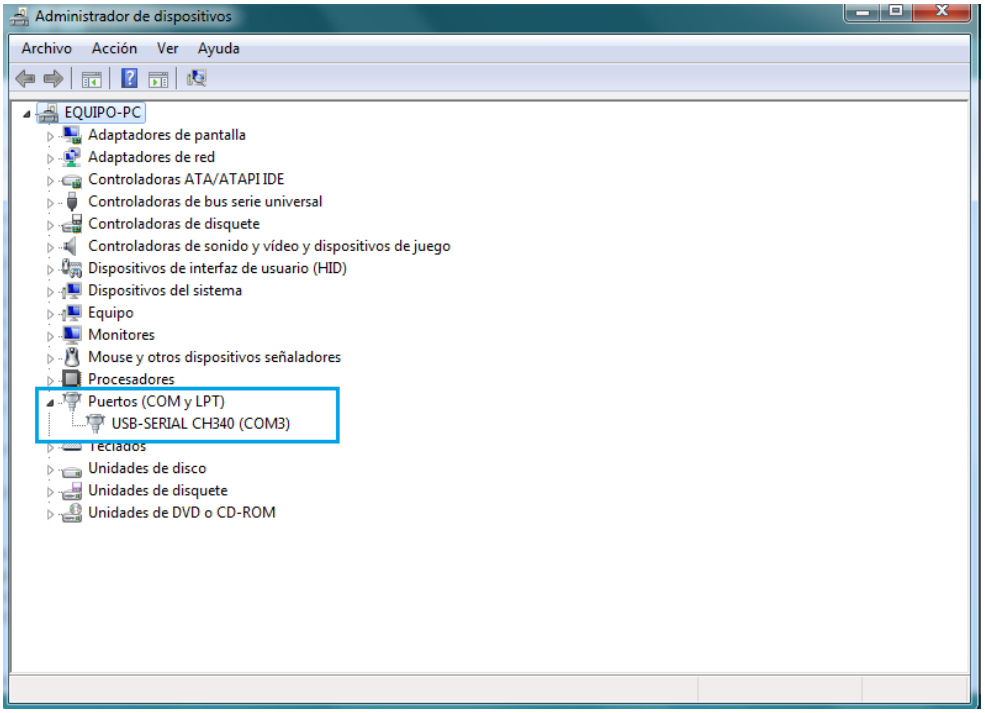
6) Ya tenemos configurado el Arduino IDE para que pueda trabajar con nuestro Node. Para comprobar su funcionamiento vamos a utilizar el ejemplo de parpadeo de luz. Seleccionamos Archivo/Ejemplos/01.Basics/Blink (File/Examples/01.Basics/Blink en inglés). Se abrirá una ventana con el código escrito y nosotros tenemos que conectar el Node a nuestra computadora utilizando un cable USB a micro USB.



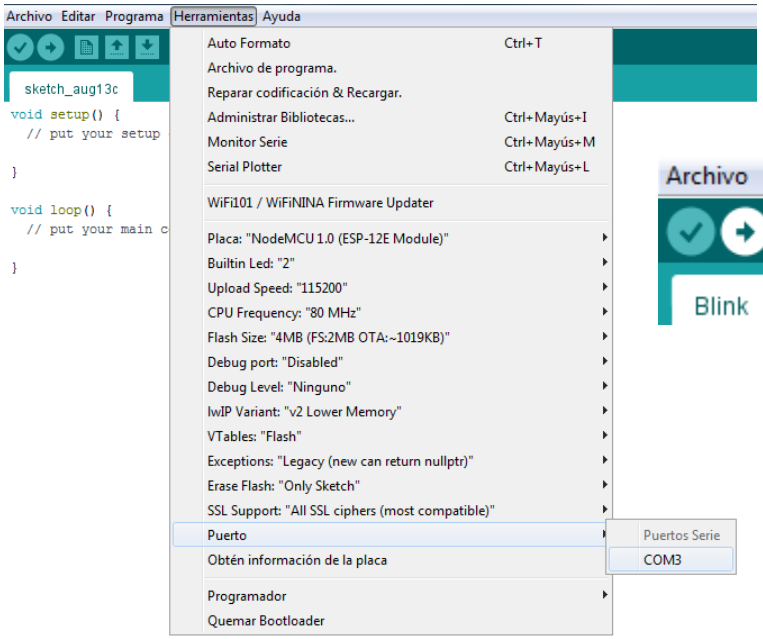
7) Al conectar el Node, la computadora debería instalar los drivers necesarios para que ambos se puedan comunicar. Es muy probable que no se pueda instalar correctamente. En ese caso, tenemos que descargar e instalar manualmente dicho driver. Para eso tenemos que ir a la siguiente página y hacer click en el botón azul con el icono de nube: http://www.wch.cn/download/CH341SER_EXE.html

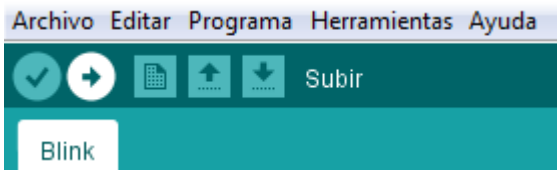
A primera vista parece ser un sitio no seguro pero es completamente lo opuesto.

8) Luego de descargar el instalador y haber pasado por el proceso de instalación, al conectar devuelta el Node a la computadora deberían poder comunicarse correctamente. Para confirmar en que puerto está conectado podemos utilizar el administrador de dispositivos en caso de que tengamos varios dispositivos conectados.



9) Seleccionar el puerto correspondiente dentro del IDE seleccionando Herramientas/Puerto (Tools/Port en inglés) y subir el código del ejemplo blink al Node presionando el botón circular “Subir” o “Upload” en inglés. ¡Configuración exitosa!





OLED

Características

La pantalla OLED que estaremos utilizando tiene un tamaño de 0.96 pulgadas o 24 milímetros, con una resolución de 128x64 pixeles. Es monocromático, por lo que solo utiliza el color blanco para mostrar en pantalla.

Tiene integrado el controlador o driver SSD1306 que convierte los datos recibidos en las señales electrónicas para controlar la pantalla. Este controlador se comunica por bus I2C (Inter-Integrated Circuit).

Ventajas

- No retroiluminación: Como la pantalla tiene su propia luz, no necesita de retroiluminación o backlight. Esto reduce significativamente la energía necesaria para hacer funcionar la OLED.
- Contraste alto: Tiene mejor visibilidad en ambientes luminosos y más brillo en ambientes oscuros.
- Bajo consumo: Sus píxeles consumen energía sólo cuando están encendidos, por lo que la pantalla OLED consume menos energía en comparación con otras pantallas.

I2C

Nuestro oled utiliza el estándar I2C para la comunicación interna de dispositivos electrónicos. Requiere únicamente dos cables para su funcionamiento, uno para la señal de reloj (SCL) y otro para el envío de datos (SDA).

Por adentro, tiene una arquitectura de tipo maestro-esclavo. El dispositivo maestro inicia la comunicación con los esclavos, y puede mandar o recibir datos de los esclavos. Los esclavos no pueden iniciar la comunicación (el maestro tiene que preguntarles), ni hablar entre si directamente. Por lo tanto, la comunicación es half-dúplex y unicast.

Además, el bus I2C es sincrónico. El maestro proporciona una señal de reloj, que mantiene sincronizados a todos los dispositivos del bus. De esta forma, se elimina la necesidad de que cada dispositivo tenga su propio reloj.

Conexión

La pantalla OLED posee 4 pines para conectar 4 cables (2 de energía y 2 de datos).

En cuestión de energía, acepta voltajes de 3.3V o 5V (voltios)

- VCC: fuente de alimentación.

- GND: de tierra o ground.

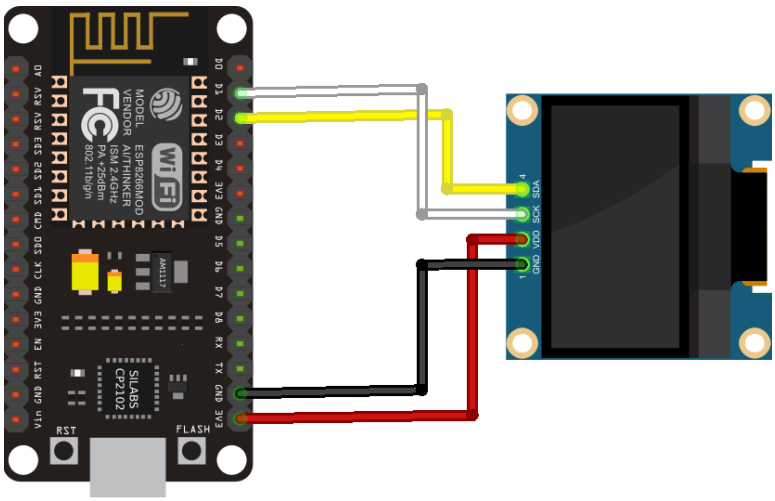
En cuestión de datos, se comunica por el bus I2C por lo cual se necesita dos conexiones

- SCL: señal de reloj.

- SDA: envió de datos.

Para conectar la pantalla a nuestra placa necesitamos utilizar cables puente y conectarlo de la siguiente manera:

OLED	NodeMCU
VCC	**3V (3.3V)**
GND	**G (GND)**
SCL	**D1 (GPIO 5)**
SDA	**D2 (GPIO 4)**



Librerías

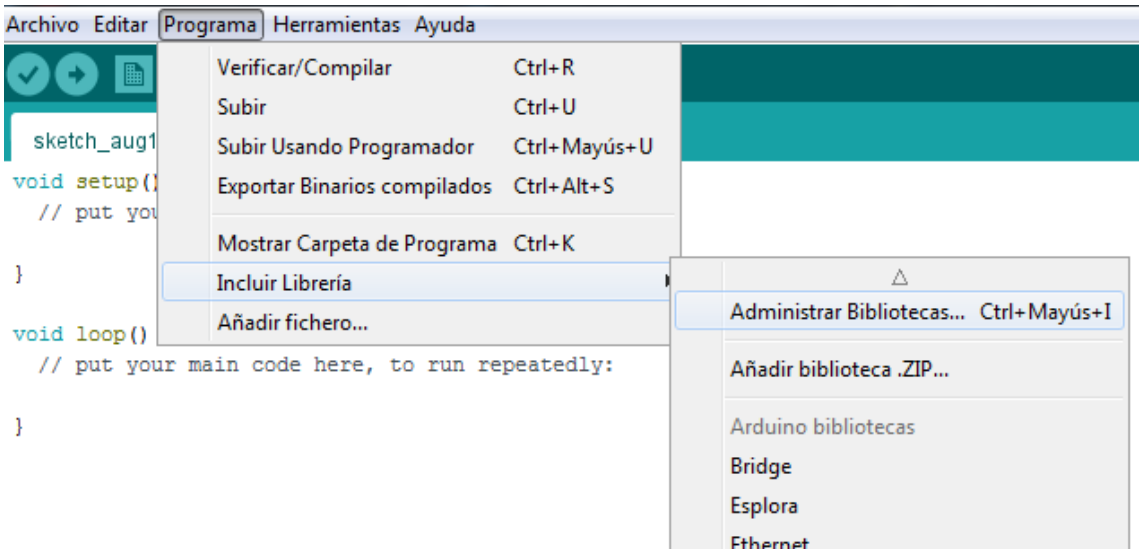
Nuestra pantalla necesita de dos librerías para su correcto funcionamiento:

- Adafruit_SSD1306: se utiliza para inicializar la pantalla y brinda funciones de visualización de bajo nivel.
- Adafruit GFX Library: proporciona funciones gráficas para mostrar texto, dibujar líneas y círculos, etc.

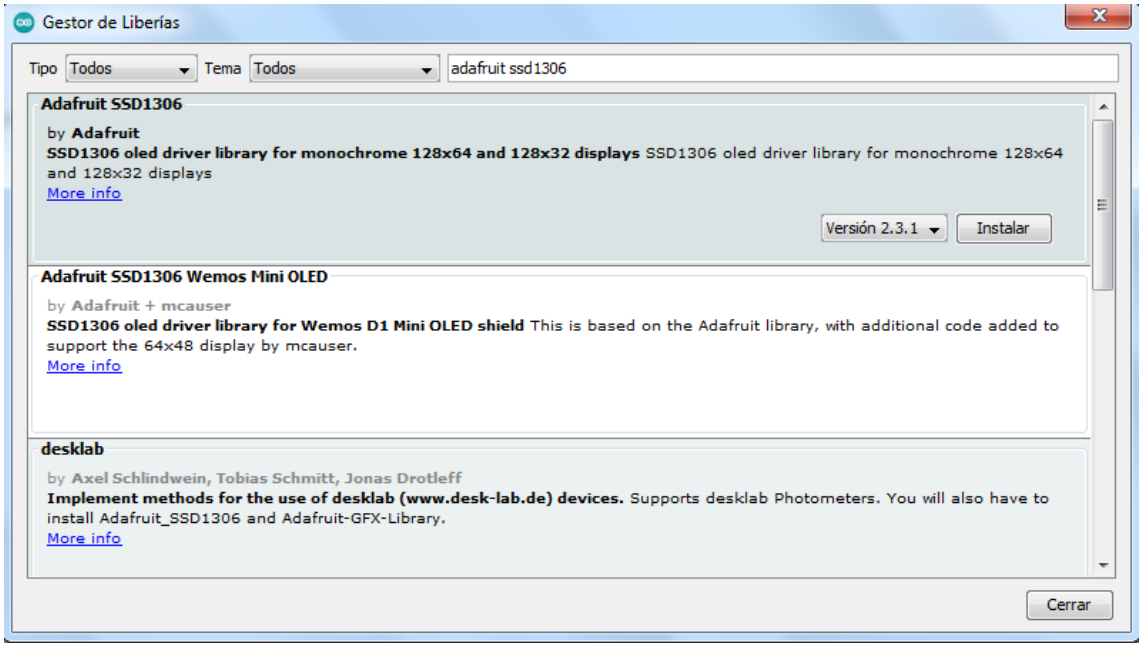
A continuación explicaremos como instalar correctamente dichas librerías y utilizar uno de sus ejemplos para verificar su correcto funcionamiento.

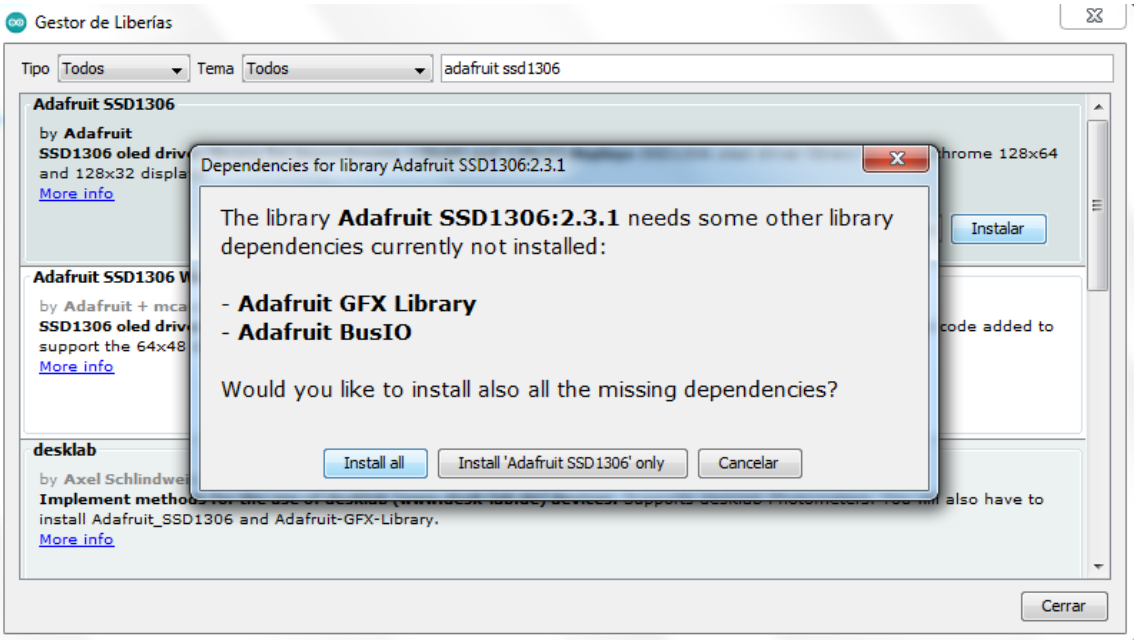
INSTALACION DE LIBRERIAS

1) Abrir Arduino IDE y seleccionar Programa/Incluir Librería/Administrar Bibliotecas (Sketch/Include Library/Manage Libraries en inglés).

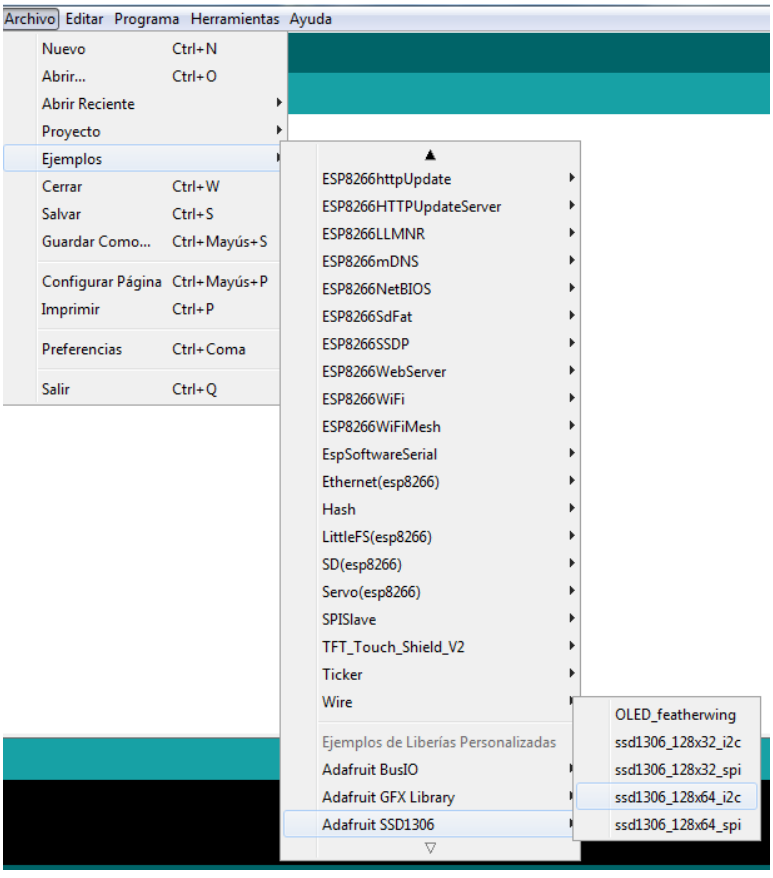


2) Dentro del gestor de librerías, escribir en el buscador: adafruit ssd1306 e instalar la primera opción (verificar que el autor sea Adafruit). Al instalar les va a preguntar si desea instalar dos dependencias de la librería. Uno de ellos es la segunda librería que necesitamos así que hacemos click en la opción “Install all”.





3) Ahora que tenemos las librerías, vamos a subir uno de los ejemplos al NodeMCU. Para ello tenemos que ir a Archivo/Ejemplos/Adafruit SSD1306/ssd1306_128x64_i2c (File/Examples/Adafruit SSD1306/ssd1306_128x64_i2c en inglés). Una vez abierto el ejemplo, conectar el Node, seleccionar el puerto en Herramientas/Puerto (Tools/Port en inglés) y subir el código presionando el botón circular “Subir” o “Upload” en inglés.



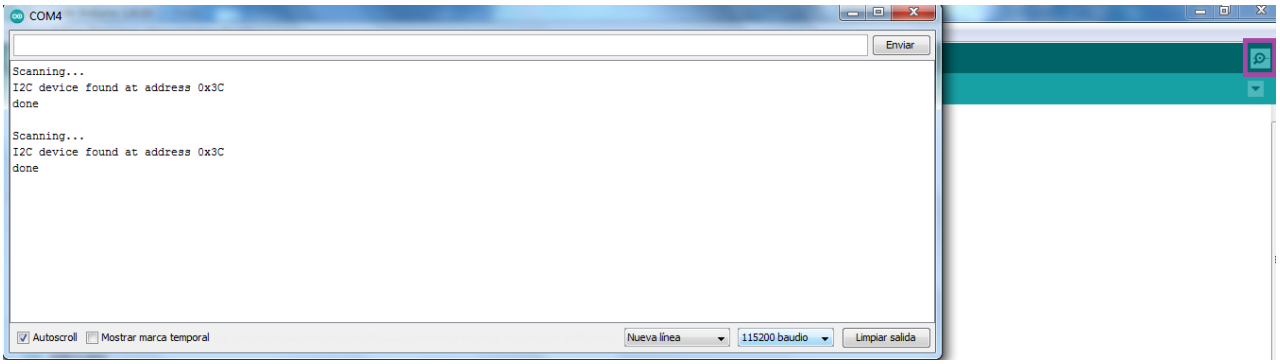
4) Posiblemente puede que luego de subir el código no aparezca nada por la pantalla. Esto se puede resolver fácilmente. Para solucionarlo se deben realizar dos cambios al código. Primero, donde se define la variable OLED_RESET cambiar el valor 4 por -1. Segundo, donde comienza el método de setup se llama a una función llamado display.begin(). El segundo parámetro está definido como “0x3D”. Nosotros debemos cambiarlo por “0x3C” (sin comillas). Luego de estos cambios, volver a subir el código y verificar si funciona.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#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     4 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define NUMFLAKES     10 // Number of snowflakes in the animation example
```

```
void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3D)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  // Show initial display buffer contents on the screen --
  // the library initializes this with an Adafruit splash screen.
  display.display();
  delay(2000); // Pause for 2 seconds

  // Clear the buffer
  display.clearDisplay();
```

5) Si el problema persiste, es probable que:
    - El puerto este mal seleccionado.
    - Los cables entre el Node y el OLED estén mal conectados.
    - La dirección I2C sea diferente.

    Con la tercera opción se hace referencia a la variable que cambiamos previamente dentro de la función display.begin(). Junto a este proyecto, se encuentra un código llamado “i2c scanner” que permite reconocer cual es el valor correcto. Al subir el código a nuestra placa, necesitamos abrir el monitor serie que se accede haciendo click en el icono de lupa en el extremo derecho de la interfaz del IDE. Dentro, se mostrará la dirección correcta. Si es la primera vez que utiliza el monitor serie, configurar la velocidad de acuerdo al Node haciendo click en el segundo botón y eligiendo la opción 115200 baudio.



6) Si el problema sigue persistiendo, lamentablemente no tengo otra solución que pueda brindar. Si el ejemplo funcionó luego del paso 4 entonces ya está listo para utilizar el OLED.

Cuerpo básico del código


```
// Display Text
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0,28);
display.println("Hello world!");
display.display();
delay(2000);
```

Para mostrar cada mensaje se suele seguir una serie de líneas que van repitiéndose a lo largo de nuestro código. Estas líneas son las siguientes:

display.clearDisplay();

Limpia el buffer de la pantalla (apaga todos los pixeles).

display.setTextSize(num);

Configura el tamaño de letra utilizando un número como parámetro (puede ser 1 hasta 10).

display.setTextColor(WHITE);

Configura el color del texto. Al ser monocromático tenemos solo 2 opciones: blanco o negro y blanco (negro y blanco hace que el texto sea de color negro y el fondo del texto de color blanco).

display.setCursor(x,y);

Específica en qué posición debe comenzar el texto. El primer parámetro es la posición horizontal y el segundo parámetro la posición vertical. A medida que el valor X aumenta, se va desplazando para la derecha y a medida que el valor Y aumenta, se va desplazando para abajo.

display.println(mensaje)

Se escribe el mensaje que se va a mostrar por pantalla (también se puede utilizar print()).

display.display()

Muestra el mensaje en la pantalla.

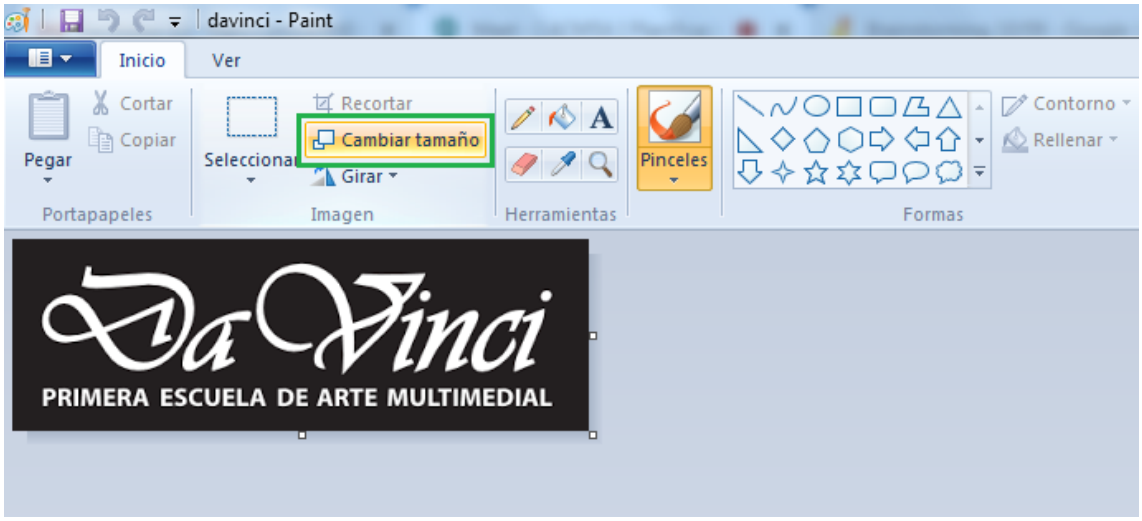
delay(num)

Número en milisegundos de la duración del mensaje en pantalla.

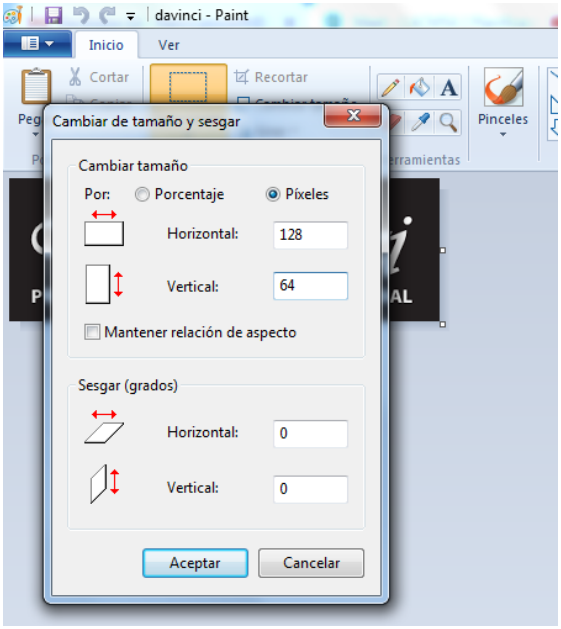
IMAGENES

Las librerías del oled, además de poder mostrar mensajes de texto, también permite dibujar formas (como líneas, cuadrados, rectángulos, etc.) y utilizar imágenes. A continuación detallaremos los pasos previos para insertar una imagen a nuestro código.

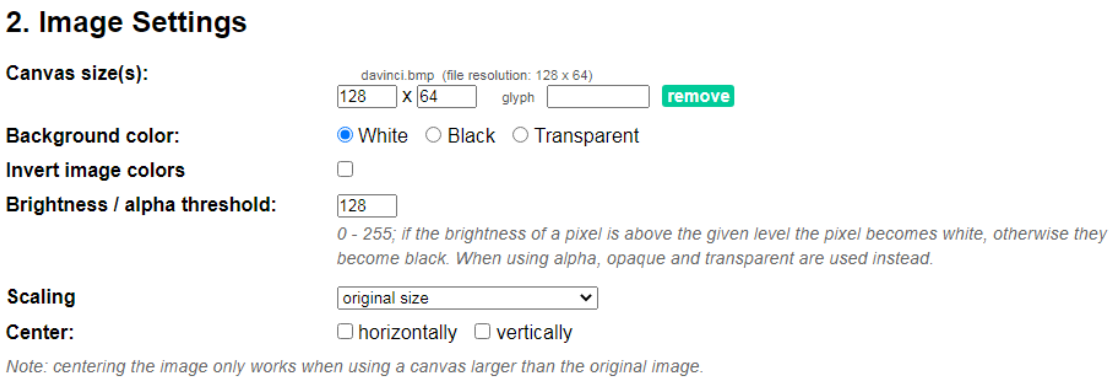
1) Encontrar una imagen que queramos utilizar y abrir el Paint u otro editor de imágenes. Dentro de Paint, abrir la imagen elegida y seleccionar la opción “Cambiar Tamaño”.



2) Dentro de esta nueva ventana, seleccionar “Píxeles” en la sección “Cambiar Tamaño”, desactivar la opción “Mantener relación de aspecto” y configurar el valor Horizontal “128” y el valor Vertical “64”. Luego aceptamos y guardamos la imagen con formato “.bmp” (Ejemplo: davinci.bmp).



3) Abrir el navegador de Internet y dirigirse al siguiente sitio: http://javl.github.io/image2cpp/
El sitio es una herramienta práctica para convertir la imagen en un array de bytes. En otras palabras, es la forma de insertar nuestra imagen al código. Dentro del sitio, seleccionaremos nuestra imagen con formato ".bmp" en la sección "Select image". Luego, únicamente prestar atención a las opciones "Background color" e "Invert image colors". En el primero definimos el color del fondo que tendrá la imagen y en el segundo podemos observar si queremos invertir los colores o no. En mi caso, seleccionaré fondo negro y sin invertir colores.



4) En la cuarta y última sección, cambiar la opción "plain bytes" por "Arduino Code". Aparecerá una opción extra llamada "Identifier" que determina el nombre que va a tener el bloque de código que representa la imagen. En mi caso, escribo davinci. En la última opción "Draw mode" mantener la opción original (Horizontal – 1 bit per pixel) y finalmente seleccionar el botón "Generate code". Copiar todo el texto generado y pegarlo en nuestro código.

4. Output

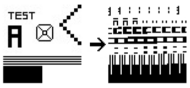
Code output format — Arduino code

Adds some extra Arduino code around the output for easy copy-paste into this example. If multiple images are loaded, generates a byte array for each and appends a counter to the identifier.

Identifier: davinci

Draw mode: Horizontal - 1 bit per pixel

If your image looks all messed up on your display, like the image below, try using a different mode.



Generate code

```
// 'davinci', 128x64px
const unsigned char davinci [] PROGMEM = {
	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, 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, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
	0x00, 0x00, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
	0x00, 0x00, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

5) Dentro de nuestro código, luego de inicializar la pantalla, necesitamos utilizar la función display.drawBitmap() para mostrar la imagen. La función tiene los siguientes parámetros: los primeros dos parámetros son X e Y para posicionar el cursor (comúnmente se definen ambos en 0). Luego se encuentra nuestra imagen (nombre del array), la resolución de nuestra imagen (la misma que la pantalla: 128x64) y el color de la imagen en pantalla (Blanco).

```
display.clearDisplay();
display.setCursor(0, 0);
display.drawBitmap(0,0,davinci,128,64,WHITE);
display.display();
delay(5000);
```

6) ¡Eso es todo! También puede ir más lejos y mostrar gifs o iconos animados a partir de múltiples imágenes pero puede haber dificultades debido a la limitada capacidad de memoria disponible.

EJERCICIOS

Slider de mensajes a través del navegador

Autor original: Sergio A. Yañez M. – Colombia, 2020

Dirección: https://www.aquilesvaesa.com/2020/03/pantalla-remota-wifi-oled-y-terminal.html

Descripción: El ejercicio consiste en la creación de un servidor web que puede ser ingresado a través de nuestro navegador si es que estamos conectados a la misma red que nuestro Node. El sitio web que nos crea el ejercicio es como el serial monitor del Arduino IDE. Escribimos los mensajes que queramos y se envían al Node para que la pantalla muestre el mensaje en forma de scroll de derecha a izquierda.

Librerías:

- Manejo de pantalla OLED: Adafruit SSD1306 y Adafruit GFX Library
- Comunicación asíncrona: ESPAsyncTCP y ESPAsyncWebServer
- Terminal donde se ingresan los mensajes: WebSerial

Reloj con representación gráfica

Autor original: Shubham Santosh– India, 2020

Dirección: https://www.hackster.io/shubhamsantosh99/internet-oled-clock-a1b8bd

Descripción: El ejercicio consiste en un reloj que sincroniza la fecha y hora a través de Internet utilizando servidores NTP. NTP (Network Time Protocol) es un protocolo estándar de Internet (IP) que se utiliza para sincronizar el reloj de la computadora conectada a la red. Además de obtener los datos, utiliza funciones de trigonometría para dibujar un reloj dentro de la pantalla que se va actualizando cada segundo.

Librerías:

- Manejo de pantalla OLED: Adafruit SSD1306 y Adafruit GFX Library
- Conexión wifi: ESP8266WiFi y WiFiUdp
- NTP: NTPClient

Muestra de valor oficial y blue del dólar

Autor original: Franco Pozzetti – Argentina, 2020

Descripción: El ejercicio consiste en mostrar el valor actual del dólar, tanto oficial como blue. Utilice el ejemplo de BasicHTTPSClient como base para mi código. Se realiza una petición a la siguiente dirección: https://www.dolarsi.com/api/api.php?type=dolar y obtengo los valores que necesito utilizando un deserializador.

Librerías:

- Manejo de pantalla OLED: Adafruit SSD1306 y Adafruit GFX Library
- Conexión wifi: ESP8266WiFi y ESP8266WiFiMulti
- Cliente y deserializador: ESP8266HTTPClient y ArduinoJson

PROBLEMAS

Lamentablemente, durante el transcurso de este proyecto, me encontré con dos problemas.

Mi primer gran problema fue la incapacidad de poder conectar el NodeMCU a mi router de WiFi. Para solucionar intente lo siguiente:

Asegurarme que el código sea correcto, que los datos (nombre y contraseña) de mi red sean correctos, configurar manualmente la conexión WiFi como modo station (cliente que se conecta a un servidor), configurar manualmente una dirección IP estática, comprobar que la dirección no esté siendo utilizada, comprobar que la dirección se encuentre dentro de un rango finito de direcciones y realizar echos a la dirección.

Finalmente, para solucionar parcialmente el problema, utilice mi celular como Access Point (servidor que espera clientes) y especifique a mi placa que se conecte a la dirección WiFi de mi celular. El Node logra conectarse exitosamente pero me encontré con otro gran problema: la incapacidad de hacer peticiones (siendo más específico, peticiones GET).

Las peticiones son formas de obtener información de algún lugar del Internet. Al principio pensaba que mi código estaba mal pero termine comprobando que, incluso con un código bien escrito, no podía obtener la información que necesitaba. Para comprobar utilice los ejemplos que brinda la librería ESP8266HTTPClient: BasicHttpClient y BasicHttpsClient. En ambos ejemplos me devolvía como código HTTP el valor “-1” que representa un error en la comunicación.

El profesor del proyecto me ayudó mucho dándome distintas formas para poder solucionar estos problemas pero no hubo forma de que funcionaran.

Todo iba para mal hasta que un día, filmando los ejercicios para presentar, me encontré con la sorpresa de que el Node podía realizar las peticiones. Quede sorprendido e inmediatamente decidí realizar mi ejercicio de dólar. Por suerte, el trabajo quedó más completo.

DESENLACE

Originalmente, este proyecto no iba a tener un final feliz. Al lidiar con todos los problemas que tuve, pensaba que no iba a lograr entregar un trabajo completo. Pero por suerte, el proyecto dio un giro positivo y pase de tener un solo ejercicio funcional a tener los tres ejercicios funcionando correctamente. A pesar de todos los bloqueos que tuve en el desarrollo de este proyecto, termine aprendiendo muchas cosas interesantes.

BIBLIOGRAFÍA

Información de NodeMCU

https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/

Información de OLED

https://diyfactory007.blogspot.com/2018/07/oled-i2c-display-arduinonodemcu-tutorial.html

https://startingelectronics.org/tutorials/arduino/modules/OLED-128x64-I2C-display/

https://randomnerdtutorials.com/esp8266-0-96-inch-oled-display-with-arduino-ide/

https://lastminuteengineers.com/oled-display-esp8266-tutorial/

https://www.luisllamas.es/conectar-arduino-a-una-pantalla-oled-de-0-96/

Bus I2C

https://www.luisllamas.es/arduino-i2c/

Librería GFX

https://cdn-learn.adafruit.com/downloads/pdf/adafruit-gfx-graphics-library.pdf

NTP

https://lastminuteengineers.com/esp8266-ntp-server-date-time-tutorial/

https://www.pool.ntp.org/zone/ar

Dólar y Deserializador

https://www.dolarsi.com/

https://arduinojson.org/v6/assistant/