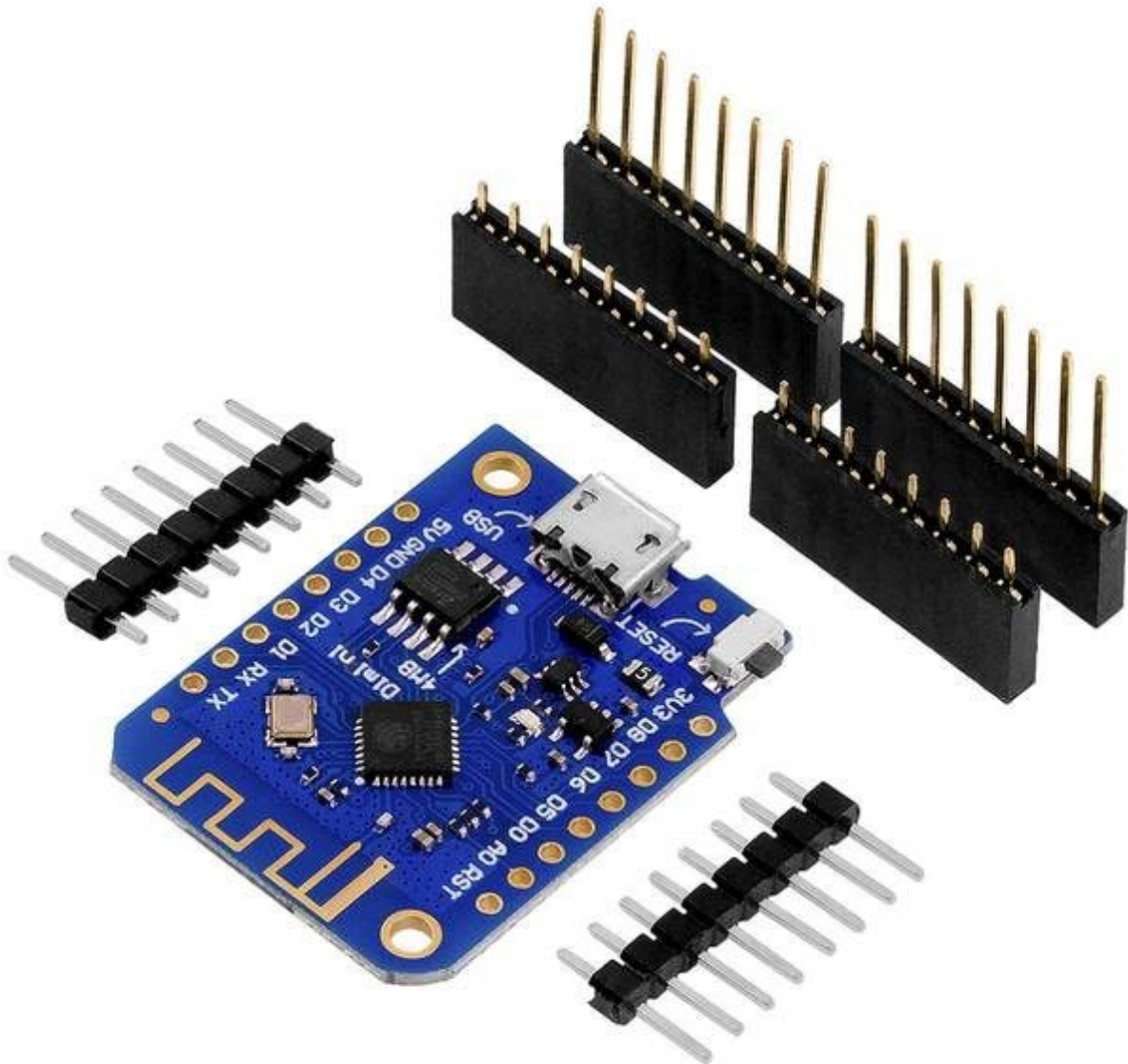


Az-Delivery

Bienvenido.

Gracias por adquirir nuestro *AZ-Delivery D1 Mini V3*. En las siguientes páginas le explicaremos cómo utilizar y configurar este práctico dispositivo.

¡Que te diviertas!





Índice

Introducción.....	3
Especificaciones	4
Características	5
El pinout	6
Pines de E/S digitales	7
PWM.....	8
Entrada analógica	8
Puertos serie	8
I2C.....	9
SPI.....	9
Módulo D1 Mini V3 - Software	10
Pines de E/S digitales.....	10
Clavija de entrada analógica	11
Comunicación en serie	12
Compartir tiempo de CPU con la parte de RF	13
Cómo configurar Arduino IDE	14
Configuración adicional.....	18
Ejemplos de bocetos	21
Ejemplo de sketch de parpadeo	21
Ejemplo de esquema PWM por software	22
Ejemplo de sketch de comunicación serie	23



Introducción

El módulo ESP8266 es un sistema en un chip (SoC). Se compone de un microcontrolador de 32 bits Tensilica L106 y un transceptor Wi-Fi. Tiene 11 pines de entrada/salida de propósito general, o para abreviar pines GPIO, y una entrada analógica. Esto significa que se puede programar como cualquier otro microcontrolador. Lo mejor del ESP8266 es que tiene comunicación Wi-Fi, lo que significa que se puede conectar a Wi-Fi, o a Internet, alojar un servidor web con páginas web reales, conectarse a un smartphone, etc. Soporta protocolos de red como Wi-Fi, TCP, UDP, HTTP, DNS, etc.

El módulo D1 Mini V3 es una placa de desarrollo basada en el chip ESP8266. Tiene 11 pines de entrada/salida digital y un pin de entrada analógica. Todos los pines de E/S digitales tienen capacidades de interrupción, PWM, I2C y 1-wire en software. El rango de tensión de entrada analógica está entre 0V y 3,3V. El módulo utiliza el puerto microUSB y el chip CH340C con un circuito programador para programar el ESP8266. Además, el puerto microUSB proporciona una fuente de alimentación para el módulo.

Existen diferentes formas de programar el módulo D1 Mini: utilizando Arduino IDE, el SDK oficial de ESP para programación en C, el intérprete Lua, el firmware MicroPython, etc.

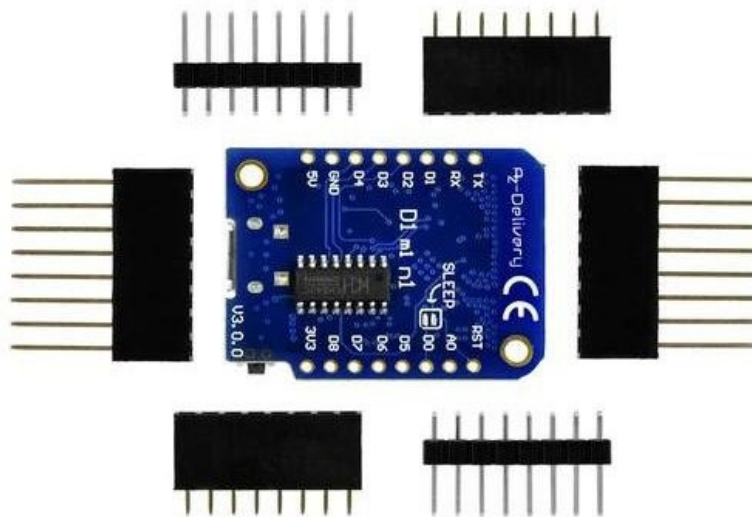
Especificaciones

Tensión de funcionamiento	3.3V
Chip principal	ESP8266-12F
Velocidad de reloj	80 MHz / 160 MHz
Flash	4MB
Pines de E/S digitales	11
Pines de entrada analógica	1
Rango de tensión de entrada analógica	de 0 V a 3,3 V
Puerto USB	microUSB
Chip USB	CH340C
Antena	Cerámica (a bordo), Conector externo
Dimensiones	25x35x6 mm [0,98x1,4x0,24 pulg].

El módulo D1 Mini Pro tiene un LED integrado. El LED está conectado internamente al puerto GPIO2.

Las cabezas de las clavijas

El módulo D1 Mini V3 viene sin soldar con un par de conectores macho de ocho patillas, un par de conectores hembra de ocho patillas y un par de conectores hembra de ocho patillas con patas extra largas (los llamados conectores apilables).



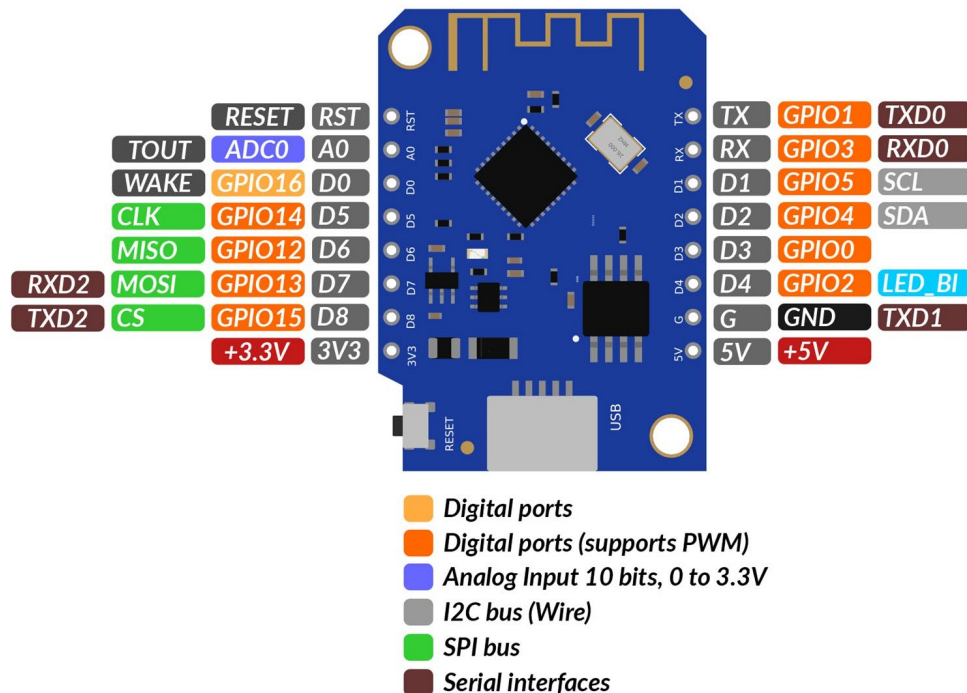


Características

- 802.11 b/g/n
- MCU de 32 bits de bajo consumo integrada
- ADC de 10 bits integrado
- Pila de protocolos TCP/IP integrada
- Conmutador TR integrado, balun, LNA, amplificador de potencia y red de adaptación
- PLL integrado, reguladores y unidades de gestión de potencia
- Admite diversidad de antenas
- Wi-Fi 2,4 GHz, compatible con WPA/WPA2
- Admite los modos de funcionamiento STA/AP/STA+AP
- Admite la función Smart Link para dispositivos Android e iOS
- SDIO2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- Agregación de A-MPDU y A-MSDU e intervalo de guarda de 0,4s
- Consumo en reposo profundo <10uA, corriente de fuga en apagado < 5uA
- Despertar y transmitir paquetes en < 2ms
- Consumo en modo de espera < 1,0 mW (DTIM3)
-+20dBm de potencia de salida en modo 802.11b
- Temperatura de funcionamiento: -40°C ~ 125°C

El pinout

El módulo D1 Mini V3 tiene dos filas de ocho pines (dieciséis pines en total). La distribución de pines se muestra en la siguiente imagen:



Nota: El módulo D1 Mini V3 tiene un pin de salida de tensión de 5V para alimentar los dispositivos electrónicos externos conectados a él, ¡pero no es tolerante a 5V! El pin de 5V es el pin de salida de energía, que emite energía desde el puerto microUSB.

Nota: El módulo D1 Mini V3 también tiene un pin de salida de tensión de 3,3 V para alimentar dispositivos electrónicos externos. Esta tensión está regulada a 3,3 V por el regulador de tensión de 3,3 V integrado.



Pines digitales de E/S

Como cualquier otra placa Atmega328p, el módulo D1 Mini V3 tiene pines de entrada/salida digital o GPIO - General Purpose Input/Output pines. Como su nombre indica, se puede utilizar como entrada digital para leer un voltaje digital, o como salida digital para dar salida a 0V o 3.3V.

El módulo D1 Mini V3 tiene un microcontrolador que funciona en un rango de tensión de 0V-3,3V.

La corriente máxima que se puede extraer de un solo pin GPIO es de 12 mA.

NOTA: Los pines del módulo D1 Mini V3 no toleran 5V, ¡aplicar más de 3,3V en cualquier pin podría dañar el chip!

GPIO1 y GPIO3 se utilizan como TX y RX para el puerto serie de hardware (UART), por lo que en la mayoría de los casos, estos dos no se puede utilizar como una normal de E / S, mientras que el envío / recepción de datos en serie.



PWM (modulación por ancho de pulsos)

A diferencia de la mayoría de los chips Atmel (Atmega328p), el módulo D1 Mini V3 no soporta PWM por hardware. Sin embargo, soporta PWM por software en todos los pines digitales. El rango PWM por defecto es de 10bits a 1kHz.

Entrada analógica

El módulo D1 Mini V3 tiene una única entrada analógica, con un rango de tensión de entrada de 0V-3,3V. Si se aplican más de 3,3V, el chip podría dañarse. El convertidor analógico-digital (ADC) tiene una resolución de 10bits.

Puertos serie

El módulo D1 Mini V3 dispone de dos UARTS hardware (puertos serie): UART0 en los pines 1 y 3 (TX0 y RX0 respectivamente), y UART1 en los pines 2 y 8 (TX1 y RX1 respectivamente). Sin embargo, GPIO8 se utiliza para conectar el chip flash. Esto significa que UART1 sólo puede transmitir datos. En la mayoría de los casos un solo puerto UART es más que suficiente.

UART0 también tiene control de flujo por hardware en los pines 15 y 13 (RTS0 y CTS0 respectivamente). Estos dos pines también se pueden utilizar como alternativa para los pines TX0 y RX0.



La interfaz I2C

El módulo D1 Mini V3 no tiene un hardware I2C (TWI - Two Wire Interface), sino que está implementado en software. Esto significa que se pueden utilizar dos pines digitales cualesquiera como pines I2C. Por defecto, la librería I2C utiliza GPIO4 como SDA y GPIO5 como SCL. La velocidad máxima del reloj I2C es de aproximadamente 450kHz.

La interfaz SPI

El módulo D1 Mini V3 tiene una conexión SPI disponible para el usuario, denominada HSPI. Puede utilizarse tanto en modo esclavo como maestro (¡por software!).

Utiliza:

- GPIO14 como reloj - CLK,
- GPIO12 como MISO,
- GPIO13 como MOSI y
- GPIO15 como Selección Esclavo - SS.



E/S digitales pins

Al igual que con una placa Atmega328p normal, la función del pin se puede establecer utilizando la siguiente línea de código:

```
pinMode(pin, mode)
```

donde *pin* es el nombre del pin GPIO, y *mode* puede ser *INPUT* (que es el predeterminado), o *OUTPUT*, o *INPUT_PULLUP* para habilitar las resistencias pull-up incorporadas para los pines *GPIO0-15*. Para habilitar la resistencia pull-down para GPIO16 utilice *INPUT_PULLDOWN_16*.

Para poner un pin de salida en *ALTO* (3,3V) o *BAJO* (0V), utilice la siguiente línea de código:

```
digitalWrite(pin, valor)
```

donde *pin* es el nombre del pin GPIO, y *valor* 1 o 0 (o *HIGH* y *BAJA*).

Para leer una entrada, utilice la siguiente línea de código:

```
digitalRead(pin)
```

Az-Delivery

Para activar PWM en un pin determinado, utiliza la siguiente línea de código:

```
analogWrite(pin, valor)
```

donde *pin* es el nombre del pin GPIO, y *value* es un número entre 0 y 1023.

El rango de la salida PWM se puede cambiar utilizando la siguiente línea de código: `analogWriteRange(new_range)`

La frecuencia del PWM se puede cambiar utilizando la siguiente línea de código:

```
analogWriteFreq(nueva_frecuencia)
```

donde *nueva_frecuencia* debe estar entre 100Hz y 1000Hz.

Entrada analógica pin

Al igual que en cualquier placa Atmega328p, la función `analogRead(A0)` se puede utilizar para obtener el voltaje analógico en la entrada analógica (0 = 0V, 1023 = 1.0V).

El módulo D1 Mini V3 también puede utilizar el ADC para medir la tensión de alimentación (VCC). Para ello, incluye `ADC_MODE(ADC_VCC)` en la parte superior del boceto, y utiliza `ESP.getVcc()` para obtener realmente la tensión.

NOTA: Si se utiliza una patilla de entrada analógica para leer la tensión de alimentación, ¡no se puede conectar nada más a la patilla de entrada analógica!



Comunicación serie

Para usar UART0 (TX=GPIO1, RX=GPIO3), usa el objeto Serial, igual que en una placa Atmega328p: `Serial.begin(velocidad_baudios)`

Para utilizar los pines alternativos (TX=GPIO15, RX=GPIO13), utilice la siguiente línea de código: `Serial.swap()`
después de `Serial.begin()`

Para utilizar UART1 (TX=GPIO2), utilice el objeto
Serial1.

NOTA: Todas las funciones de flujo de Atmega328p, como *read()*, *write()*, *print()*, *println()*, etc. también están soportadas.



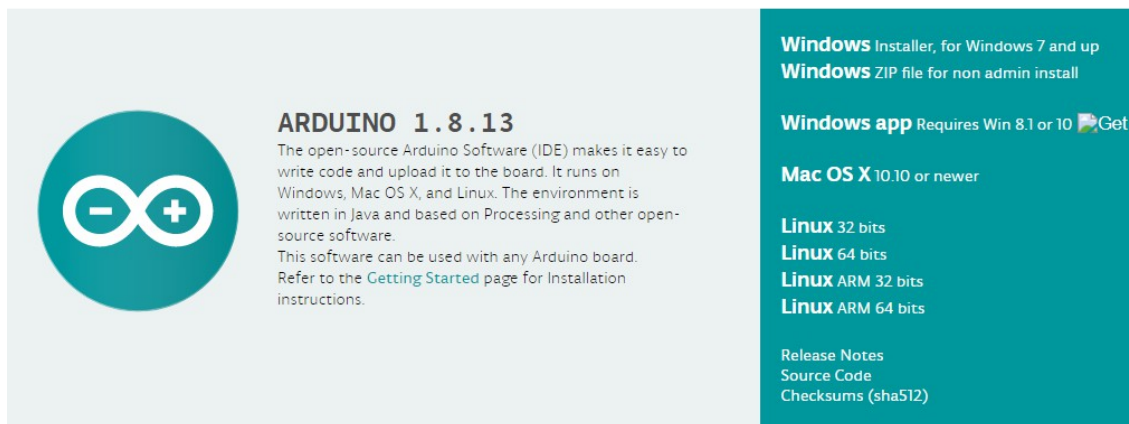
Compartiendo tiempo de CPU con la parte RF

Una cosa a tener en cuenta al escribir programas para el módulo D1 Mini V3 (ESP8266) es que el boceto tiene que compartir recursos (tiempo de CPU y memoria) con las pilas Wi-Fi y TCP (el software que se ejecuta en segundo plano y gestiona todas las conexiones Wi-Fi e IP). Si el código tarda demasiado en ejecutarse, y no deja que las pilas TCP hagan lo suyo, el programa podría bloquearse, o los datos podrían perderse. Es mejor mantener el tiempo de ejecución del bucle por debajo de un par de cientos de milisegundos. Cada vez que se repite el bucle principal, un sketch cede al Wi-Fi y al TCP para manejar todas las peticiones WiFi y TCP. Si el bucle tarda más que esto, el tiempo de CPU tiene que ser dado explícitamente a las pilas Wi-Fi/TCP, usando incluyendo `delay(0)` o `yield()`. Si esto no se hace, la comunicación de red no funcionará como se espera, y si es superior a 3 segundos, el soft WDT (Watchdog Timer) reinicia el ESP. Si el WDT suave está desactivado, después de un poco más de 8 segundos, el WDT hardware reinicia el chip. Sin embargo, desde la perspectiva del microcontrolador, 3 segundos es un tiempo muy largo (240 millones de ciclos de reloj), así que a menos que se haga algún cálculo numérico extremadamente pesado, o se envíen cadenas extremadamente largas por serie, el sketch no se verá afectado por esto. Sólo ten en cuenta añadir el `yield()` dentro de los bucles `for` o `while` que podrían tardar más de, digamos 100ms.

Cómo configurar Arduino IDE

Si el IDE de Arduino no está instalado, siga el [enlace](#) y descargue el archivo de instalación para el sistema operativo de su elección. **La versión de Arduino IDE utilizada para este eBook es la 1.8.13.**

Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a teal circle with a white infinity symbol containing a minus and a plus sign. To its right, the text reads: **ARDUINO 1.8.13**. Below this, it states: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." On the right side of the page, there is a teal sidebar with white text listing download options: "Windows Installer, for Windows 7 and up", "Windows ZIP file for non admin install", "Windows app Requires Win 8.1 or 10", "Mac OS X 10.10 or newer", "Linux 32 bits", "Linux 64 bits", "Linux ARM 32 bits", "Linux ARM 64 bits", "Release Notes", "Source Code", and "Checksums (sha512)".

Para los usuarios de *Windows*, haga doble clic en el archivo `.exe` descargado y siga las instrucciones de la ventana de instalación.

Az-Delivery

Para los usuarios de *Linux*, descarga un archivo con la extensión *.tar.xz*, que hay que extraer. Cuando esté extraído, ve al directorio extraído y abre el terminal en ese directorio. Hay que ejecutar dos scripts *.sh*, el primero llamado *arduino-linux-setup.sh* y el segundo llamado *install.sh*.

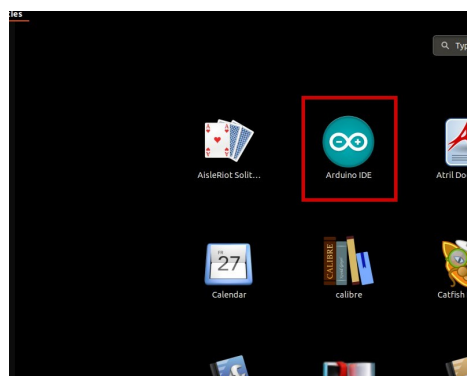
Para ejecutar el primer script en el terminal, abra el terminal en el directorio extraído y ejecute el siguiente comando:

```
sh arduino-linux-setup.sh nombre_usuario
```

nombre_usuario - es el nombre de un superusuario en el sistema operativo Linux. Se debe introducir una contraseña para el superusuario cuando se inicie el comando. Espera unos minutos a que el script lo complete todo.

El segundo script, llamado *install.sh*, debe utilizarse después de la instalación del primer script. Ejecute el siguiente comando en el terminal (directorio extraído): **sh install.sh**

Después de la instalación de estos scripts, vaya a *Todas las aplicaciones*, donde el *IDE Arduino* está instalado.



Az-Delivery

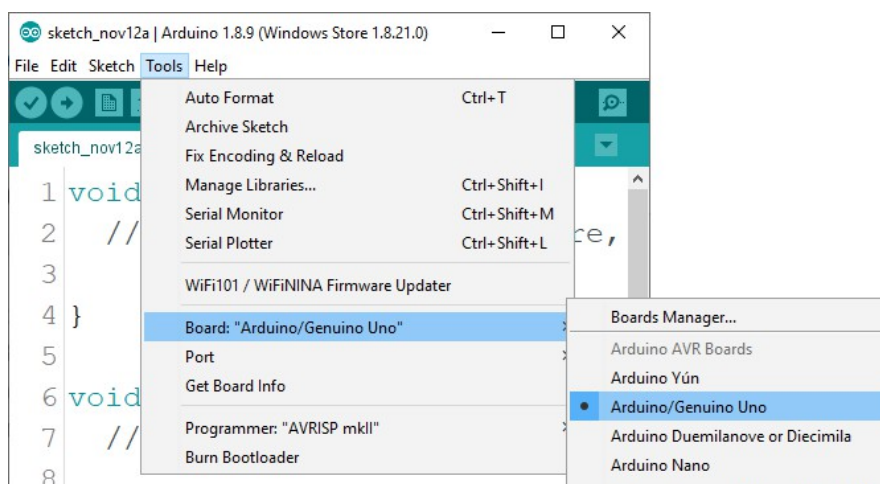
Casi todos los sistemas operativos vienen con un editor de texto preinstalado (por ejemplo, *Windows* viene con *Notepad*, *Linux Ubuntu* viene con *Gedit*, *Linux Raspbian* viene con *Leafpad*, etc.). Todos estos editores de texto son perfectamente adecuados para el propósito del eBook.

Lo siguiente es comprobar si tu PC puede detectar una placa Atmega328p.

Abra recién instalado Arduino IDE, y vaya a:

Herramientas > Tablero > {nombre de su tablero aquí}

{el nombre de tu placa aquí} debería ser el *Arduino/Genuino Uno*, como se puede ver en la siguiente imagen:



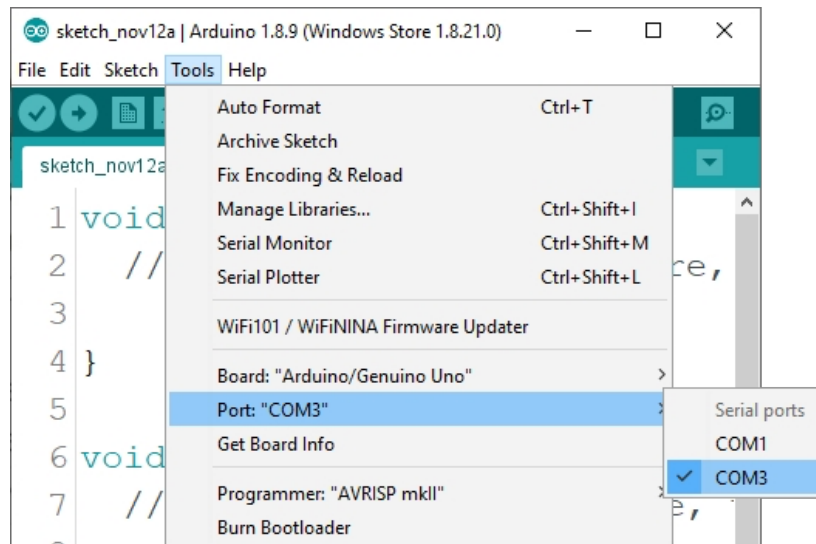
Hay que seleccionar el puerto al que está conectada la placa Atmega328p. Ir

a: *Herramientas > Puerto > {el nombre del puerto va aquí}*

y cuando la placa Atmega328p está conectada al puerto USB, se puede ver el nombre del puerto en el menú desplegable de la imagen anterior.

Az-Delivery

Si se utiliza el IDE Arduino en Windows, los nombres de los puertos son los siguientes:



Para los usuarios de *Linux*, por ejemplo, el nombre del puerto es */dev/ttyUSBx*, donde *x* representa un número entero entre 0 y 9.

Configuración adicional de

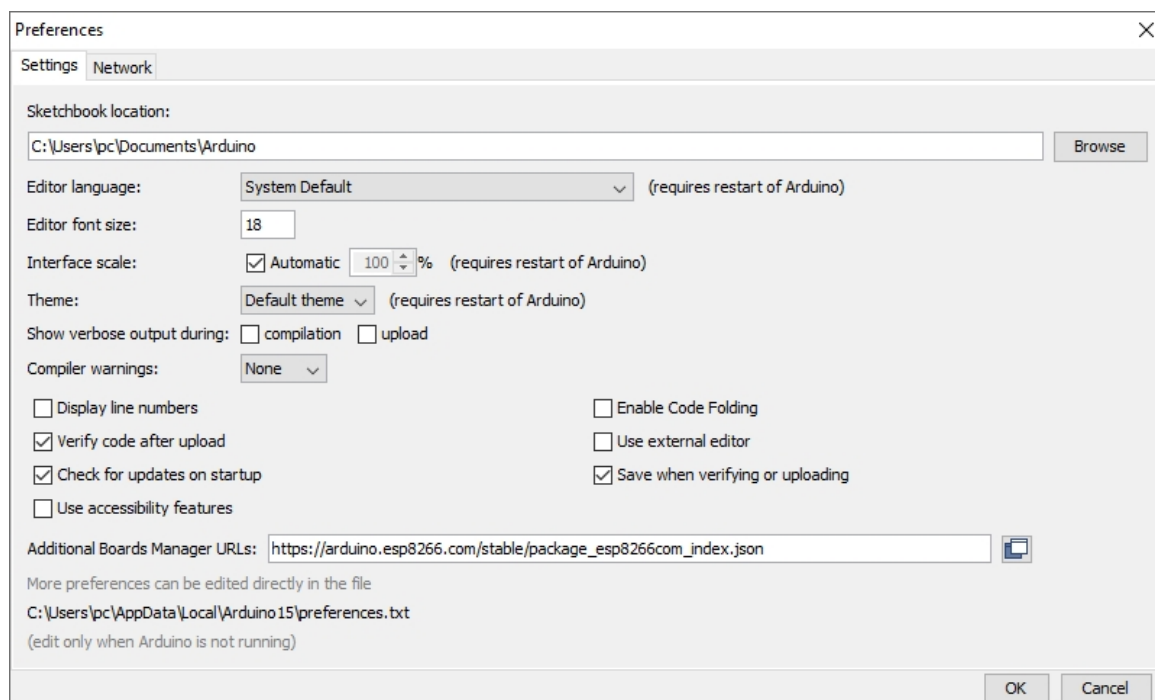
Para utilizar D1 Mini V3 con Arduino IDE, siga unos sencillos pasos. Antes de configurar el IDE Arduino, hay que instalar el driver para la comunicación USB a Serie. Si el driver no se instala automáticamente, hay una página de soporte que contiene los drivers para Windows/Mac o Linux y se puede elegir en función del que se utilice. Los drivers se pueden descargar desde el siguiente [enlace](#).

A continuación, para instalar el soporte para la plataforma ESP8266, abra Arduino IDE y vaya a:

Archivo > Preferencias, y busque el campo URLs adicionales.

A continuación, copie la siguiente URL:

https://arduino.esp8266.com/stable/package_esp8266com_index



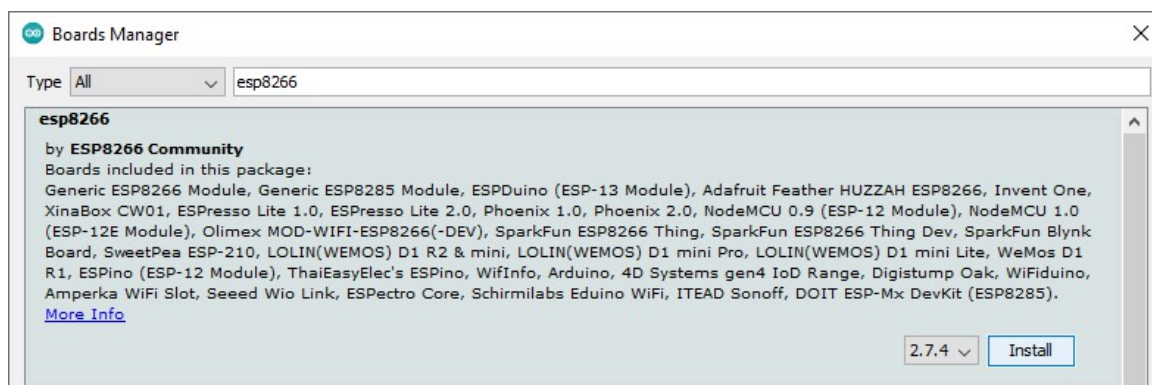
Pegue este enlace en el campo URLs adicionales. Si hay uno o más enlaces dentro de este campo, añada una coma después del último enlace, pegue el nuevo enlace después de la coma y pulse el botón OK.



Abra Arduino IDE de nuevo y vaya a:

Herramientas > Tablero > Administrador de tableros

Cuando se abra la nueva ventana, escriba *esp8266* en el cuadro de búsqueda e instale la biblioteca creada por *ESP8266 Community*, como se muestra en la siguiente imagen:



Para seleccionar el tablero D1, vaya a:

Herramientas > Placas > Placas ESP8266 > Módulo ESP8266 genérico

Para subir el código del sketch a la D1 Mini V3, primero selecciona el puerto al que está conectada la placa. Vaya a: *Herramientas > Puerto > {nombre del puerto}*

Boceto ejemplos

Blink sketch ejemplo

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
void loop() {  
    digitalWrite(LED_BUILTIN, LOW); // LOW activo enciende el LED  
    delay(1000);  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(1000);  
}
```

Para el módulo D1 Mini V3, *LED_BUILTIN* es igual a un número 2, lo que significa que el LED de a bordo está conectado al pin GPIO2. Para encender el LED hay que poner el pin GPIO2 en estado BAJO, y para apagarlo hay que poner el pin GPIO2 en estado ALTO, alternando así con un LED externo conectado a GPIO2.



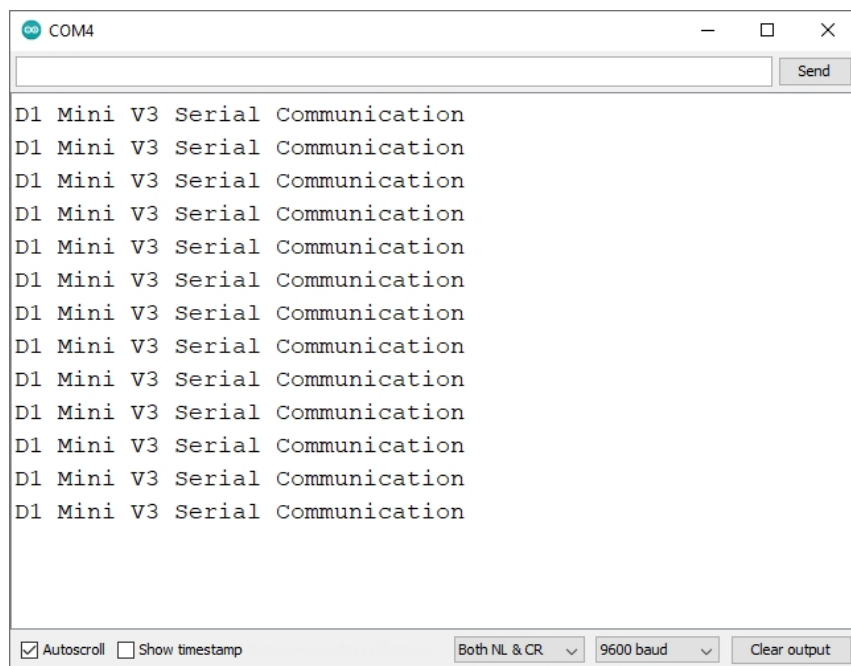
Software PWM sketch ejemplo

```
void setup() {  
}  
  
void loop() {  
    // aumentar el brillo de los LED  
    for (int dutyCycle = 0; dutyCycle < 1023; dutyCycle++) {  
        // cambiar el brillo del LED con PWM  
        analogWrite(LED_BUILTIN, dutyCycle);  
        delay(2);  
    }  
  
    // disminuye el brillo del LED  
    for (int dutyCycle = 1023; dutyCycle > 0; dutyCycle--) {  
        // cambiar el brillo del LED con PWM  
        analogWrite(LED_BUILTIN, dutyCycle);  
        delay(2);  
    }  
}
```

Sketch de comunicación serie ejemplo

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Comunicación serie D1 Mini V3"); delay(1000);  
}
```

Sube el sketch al módulo D1 Mini V3 y abre el Serial Monitor (*Herramientas* > *Serial Monitor*). El resultado debería ser como el de la siguiente imagen:





Ahora es el momento de aprender y hacer tus propios proyectos. Puedes hacerlo con la ayuda de muchos scripts de ejemplo y otros tutoriales, que puedes encontrar en Internet.

Si busca microelectrónica y accesorios de alta calidad, AZ-Delivery Vertriebs GmbH es la empresa adecuada. Dispondrá de numerosos ejemplos de aplicación, guías de instalación completas, libros electrónicos, bibliotecas y asistencia de nuestros expertos técnicos.

<https://az-delivery.de>

¡Diviértete!

Impresionante

<https://az-delivery.de/pages/about-us>