

Az-Delivery

Bienvenido.

Gracias por adquirir nuestro *AZ-Delivery D1 R32*. En las siguientes páginas le explicaremos cómo utilizar y configurar este práctico dispositivo.

¡Que te diviertas!





Índice

Introducción.....	3
Especificaciones.....	4
D1 R32.....	5
Pinout.....	6
Descripción de las clavijas.....	7
Pines del sensor táctil capacitivo.....	8
Clavijas del conversor analógico-digital.....	9
Clavijas del convertidor de digital a analógico.....	9
Reloj en tiempo real Pines GPIO.....	10
Clavijas PWM (modulación por ancho de pulsos).....	11
Los pines de la interfaz I2C.....	11
Pines de interfaz SPI.....	12
Pasadores de fleje.....	12
Alfileres HIGH en Boot.....	13
Comunicación USB a serie.....	14
Comunicación WiFi.....	15
Comunicación Bluetooth.....	16
Otras prestaciones.....	18
Cómo configurar Arduino IDE.....	19
Configuración adicional.....	23
Ejemplos de bocetos.....	26



Introducción

La D1 R32 es una placa de desarrollo (Arduino) Uno creada alrededor del chip ESP32 WROOM-32, que contiene un regulador de voltaje, un circuito programador USB para el chip ESP32, y muchas otras características.

Para el desarrollo de aplicaciones se puede elegir entre Arduino IDE o ESP-IDF (plataforma nativa). La mayoría de los usuarios eligen Arduino IDE por su sencillez y compatibilidad. La comunidad de usuarios de Arduino es muy activa y soporta plataformas como ESP32.

La D1 R32 viene con un firmware preinstalado que permite trabajar con el lenguaje interpretado, enviando comandos a través del puerto serie (chip CH340). Las placas ESP32 son una de las plataformas más utilizadas para proyectos de Internet de las Cosas (IoT).

La placa D1 R32 está diseñada para trabajar con diferentes periféricos y tiene compatibilidad con algunos shields diseñados particularmente para esta placa. Dispone de un regulador de voltaje que le permite alimentarse directamente del puerto USB o de la toma para un jack DC. Los pines de entrada/salida funcionan a 3,3V. El chip CH340 se encarga de la comunicación USB a serie.

Especificaciones

Tensión de alimentación (microUSB)	5VDC
Tensión de entrada CC	7-12V
Tensión de entrada/salida	3.3V
Corriente de funcionamiento necesaria	mín. 250 mA
SoC	ESP32 WROOM-32
Gama de frecuencias de reloj	240 MHz
RAM	512kB
Memoria flash externa	4MB
Clavijas digitales	20
Clavijas analógicas	6
Interfaces de comunicación	SPI, I2C, I2S, IR, UART, PWM
Protocolos Wi-Fi	802.11 b/g/n/i (802.11n hasta 150 Mbps)
Frecuencia Wi-Fi	2,4 GHz - 2,5 GHz
Antena inalámbrica	PCB
Chip USB a serie	CH340
Dimensiones	70x55x13mm(2.7x2.1x0.5in)

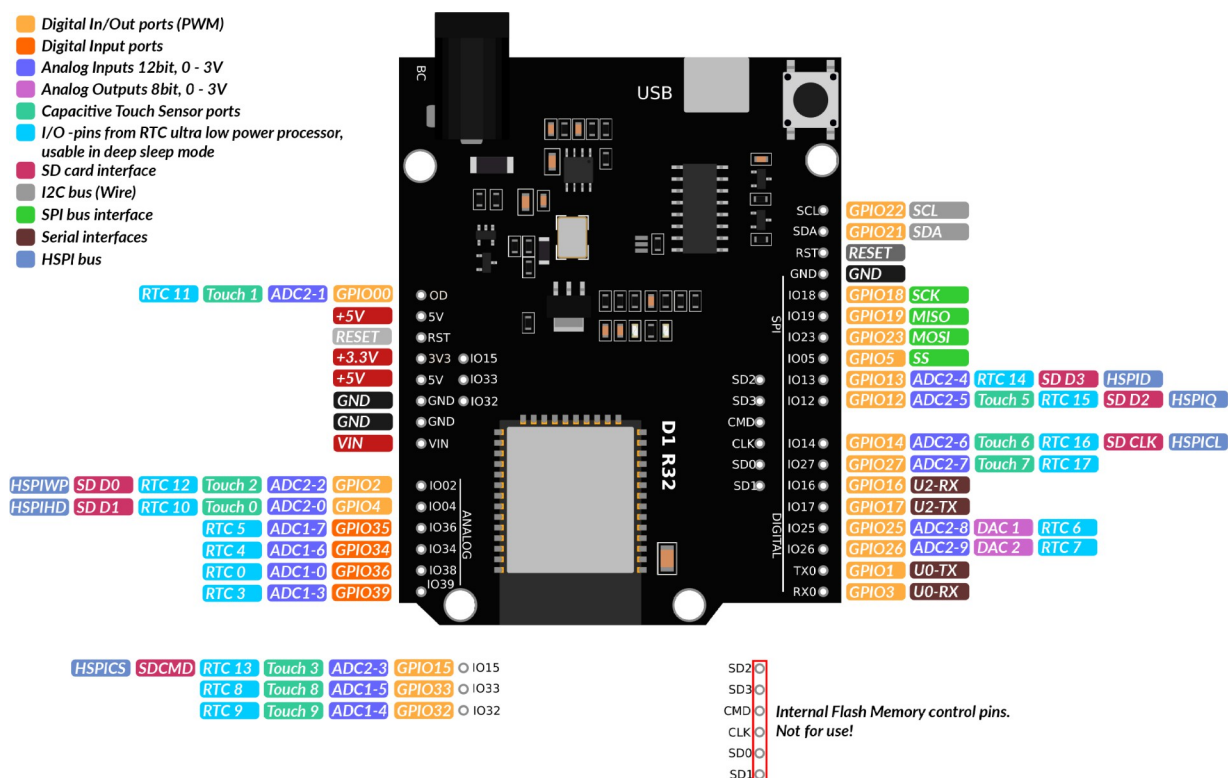
D1 R32

La serie ESP32 de chips Wi-Fi es producida por Espressif Systems. ESP32 WROOM-32 es un módulo Wi-Fi asequible adecuado para proyectos de bricolaje en el campo de Internet de las Cosas (IoT). Este módulo viene con muchos GPIOs y soporte para una variedad de protocolos como SPI, I2C, I2S, UART, y más. Notable es que viene con red inalámbrica incluida, lo que lo hace diferente a otros microcontroladores como el Arduino. Esto significa que puede controlar y supervisar fácilmente dispositivos de forma remota a través de Wi-Fi y Bluetooth® a un precio asequible.

ESP32 WROOM-32 es un sistema en chip (SoC) que integra un microcontrolador Tensilica de 32 bits, interfaces periféricas digitales estándar, conmutadores de antena, balun de RF, amplificador de potencia, amplificador receptor de bajo ruido, filtros y módulos de gestión de energía en un pequeño paquete. Proporciona Wi-Fi de 2,4 GHz (802.11 b/g/n, que soporta una velocidad de hasta 150 MB/s), BLE y comunicación inalámbrica Bluetooth® clásica, los pines generales proporcionan conectividad a los pines GPIO que soportan PWM (modulación por ancho de pulsos), pines GPI (sólo entrada), sensores táctiles capacitivos, interfaces I2C e I2S, ADC (conversión analógica a digital), DAC (conversión digital a analógica), interfaz SPI o UART en pines dedicados.

Pinout

El D1 R32 tiene 22 pines. El pinout se muestra en la siguiente imagen:



Para una descripción detallada de los pines y las capacidades de E/S, consulte la hoja de datos que puede encontrar en el siguiente [enlace](#).



Descripción de las clavijas

Al igual que una placa Arduino normal, el D1 R32 tiene pines digitales de entrada/salida (pines GPIO - General Purpose Input/Output pins). Estas entradas/salidas digitales funcionan a 3.3V.

¡El voltaje de 5V no debe conectarse a ningún pin del chip ESP32!

Los pines no son tolerantes a 5V, aplicar más de 3.3V en cualquier pin destruirá el chip.

Los pines GPIO 34 a 39 son GPIOs - pines de sólo entrada. Estos pines no tienen resistencias internas de pull-up o pull-down. No se pueden utilizar como salidas, por lo que estos pines sólo se pueden utilizar como entradas: GPIO34, GPIO35, GPIO36, GPIO39

Hay una flash SPI integrada en el chip ESP-WROOM-32. Los pines GPIO6 a GPIO11 están expuestos en ciertas placas de desarrollo ESP32. Estos pines: GPIO6(SCK/CLK), GPIO7(SDO/SD0), GPIO8(SDI/SD1), GPIO9(SHD/SD2), GPIO10(SWP/SD3), GPIO11(CSC/CMD) están conectados

a la flash SPI integrada en el chip y no se recomiendan para otros usos.



Clavijas del sensor táctil capacitivo

El ESP32 tiene sensores táctiles capacitivos internos. Los pines táctiles capacitivos también se pueden utilizar para despertar el ESP32 del sueño profundo.

Los pines en D1 R32 con capacidad táctil capacitiva son:

GPIO4(Touch 0), GPIO00(Touch 1), GPIO2(Touch 2), GPIO12(Touch 5), GPIO14(táctil 6), GPIO27(táctil 7).



Clavijas del conversor analógico-digital

El ESP32 tiene 16x12 bits ADC(Analog to Digital converter) canales de entrada (mientras que el ESP8266 sólo tiene 1x10bits ADC). Estos son los GPIOs que se pueden utilizar como ADC en el D1 R32:

GPIO36(ADC1-0), GPIO39(ADC1-3), GPIO32(ADC1-4), GPIO33(ADC1-5), GPIO35(ADC1-7), GPIO34(ADC1-6), GPIO04(ADC2-0), GPIO00(ADC2-1), GPIO2(ADC2-2), GPIO15(ADC2-3), GPIO13(ADC2-4), GPIO12(ADC2-5), GPIO14(ADC2-6), GPIO27(ADC2-7), GPIO25(ADC2-8), GPIO26(ADC2-9)

Clavijas del convertidor de digital a analógico

Hay 2x8 bits DAC(Digital to Analog converter) canales en el ESP32 para convertir señales digitales en salidas de señal analógica de tensión. Estos son los canales DAC:

DAC1(GPIO25), DAC2(GPIO26).



Reloj en tiempo real Pines GPIO

Hay RTC (reloj en tiempo real) GPIO apoyo en el ESP32. Los GPIO enrutados al subsistema de bajo consumo RTC se pueden utilizar cuando el ESP32 está en reposo profundo. Estos GPIOs RTC se pueden utilizar para despertar el ESP32 del sueño profundo cuando el coprocesador de ultra bajo consumo (ULP) se está ejecutando. Los siguientes GPIOs se pueden utilizar como fuente externa de activación:

GPIO36(RTC-0), GPIO39(RTC-3), GPIO34(RTC-4), GPIO35(RTC-5),
GPIO25(RTC-6), GPIO26(RTC-7), GPIO33(RTC-8), GPIO32(RTC-9),
GPIO04(RTC-10), GPIO00(RTC-11), GPIO2(RTC-12), GPIO15(RTC-13),
GPIO13(RTC-14), GPIO12(RTC-15), GPIO14(RTC-16), GPIO27(RTC-17).



Clavijas PWM (modulación por ancho de pulsos)

El controlador PWM (modulación por ancho de pulsos) del ESP32 tiene 16 canales independientes que se pueden configurar para generar señales PWM con diferentes propiedades. Todos los pines que pueden actuar como salidas pueden ser utilizados como pines PWM (GPIOs 34 a 39 no pueden generar PWM). Para configurar una señal PWM, hay que definir los parámetros en el código: Frecuencia de la señal, Ciclo de trabajo, Canal PWM, GPIO por donde saldrá la señal.

Los pines de la interfaz I2C

El ESP32 tiene dos canales I2C y cualquier pin se puede establecer como SDA o SCL. Cuando se utiliza el ESP32 con el IDE de Arduino, los pines I2C por defecto son: GPIO21(SDA), GPIO22(SCL).

Pines de interfaz SPI

Por defecto, la asignación de pines para los pines SPI es:

SPI	MOSI	MISO	CLK	CS
VSPI	GPIO23	GPIO19	GPIO18	GPIO5
HSPI	GPIO13	GPIO12	GPIO14	GPIO15

Pasadores de fleje

Los siguientes pines se utilizan para poner el ESP32 en modo bootloader o flasheo: GPIO0, GPIO2, GPIO4, GPIO5 (debe estar ALTO durante el arranque), GPIO12 (debe estar BAJO durante el arranque), GPIO15 (debe estar ALTO durante el arranque).

La mayoría de las placas de desarrollo ponen los pines en el estado correcto para el flasheo o el modo de arranque. Si hay periféricos conectados a los pines de arranque y el IDE no puede cargar el código o flashear el ESP32, puede deberse a que esos periféricos están impidiendo que el ESP32 entre en el modo correcto. Después de reiniciar, flashear o arrancar, esos pines funcionan como se espera. Hay una guía de documentación sobre la selección del modo de arranque en el siguiente [enlace](#). Más explicaciones y más extensa no están en el ámbito de este eBook así que por favor, consulte la hoja de datos.



Clavijas HIGH en Boot

Algunos GPIOs cambian su estado a HIGH o emiten señales PWM en el arranque o reinicio. Esto significa que si se conectan salidas a estos GPIOs se pueden obtener resultados inesperados cuando el ESP32 se reinicia o arranca.

GPIO1, GPIO3, GPIO5, GPIO6 a GPIO11 (conectados a la memoria flash SPI integrada del ESP32), GPIO14, GPIO15.



Comunicación de USB a serie

El D1 R32 dispone de un puerto de conexión microUSB. Está hecho alrededor del chip CH340 que permite la comunicación serie USB a UART. El chip tiene la característica de puerto COM virtual (VCP) que aparece como puerto COM en aplicaciones de PC. La interfaz UART del CH340 implementa todas las señales RS-232, incluidas las de control y handshaking, por lo que no es necesario modificar el firmware del sistema existente. Para poder utilizar el D1 R32 es necesario instalar el controlador correspondiente en el ordenador/sistema operativo.



Comunicación WiFi

El D1 R32 dispone de interfaz de comunicación Wi-Fi integrada y puede funcionar en tres modos diferentes: Estación Wi-Fi, punto de acceso Wi-Fi y ambos al mismo tiempo. Es compatible con las siguientes funciones:

- Velocidades de datos 802.11b y 802.11g
- 802.11n MCS0-7 con ancho de banda de 20 MHz y 40 MHz
- 802.11n MCS32
- Intervalo de guarda 802.11n 0,4μS
- Velocidad de datos de hasta 150 Mbps
- Recepción STBC 2x1
- Potencia de transmisión de hasta 20 dBm
- Potencia de transmisión regulable
- Diversidad y selección de antenas (hardware gestionado por software)

Comunicación Bluetooth

El D1 R32 tiene una radio Bluetooth integrada y admite las siguientes funciones:

- Potencias de salida de transmisión de clase 1, clase 2 y clase 3 y rango de control dinámico de hasta 20 dB
- Modulación $\pi/4$ DQPSK y 8 DPSK
- Alto rendimiento en sensibilidad del receptor NZIF con más de 98 dB de rango dinámico
- Funcionamiento de clase 1 sin megafonía externa
- La SRAM interna permite la transferencia de datos a máxima velocidad, la combinación de voz y datos y el funcionamiento completo en piconet.
- Lógica para la corrección de errores hacia delante, control de errores de cabecera, correlación de códigos de acceso, CRC, demodulación, generación de flujos de bits de encriptación, blanqueamiento y conformación de impulsos de transmisión.
- ACL, SCO, eSCO y AFH
- Códec de audio digital A-law, μ -law y CVSD en interfaz PCM
- CODEC de audio SBC
- Gestión de energía para aplicaciones de bajo consumo
- SMP con AES de 128 bits



Además, la radio Bluetooth es compatible con los siguientes protocolos de interfaz de comunicación:

- Interfaz UART HCI, hasta 4 Mbps
- Interfaz SDIO/SPI HCI
- Interfaz I2C
- Interfaz de audio PCM/I2S.

Otras características

El chip ESP32-WROOM 32 tiene un sensor de efecto Hall integrado que detecta cambios en el campo magnético de su entorno.

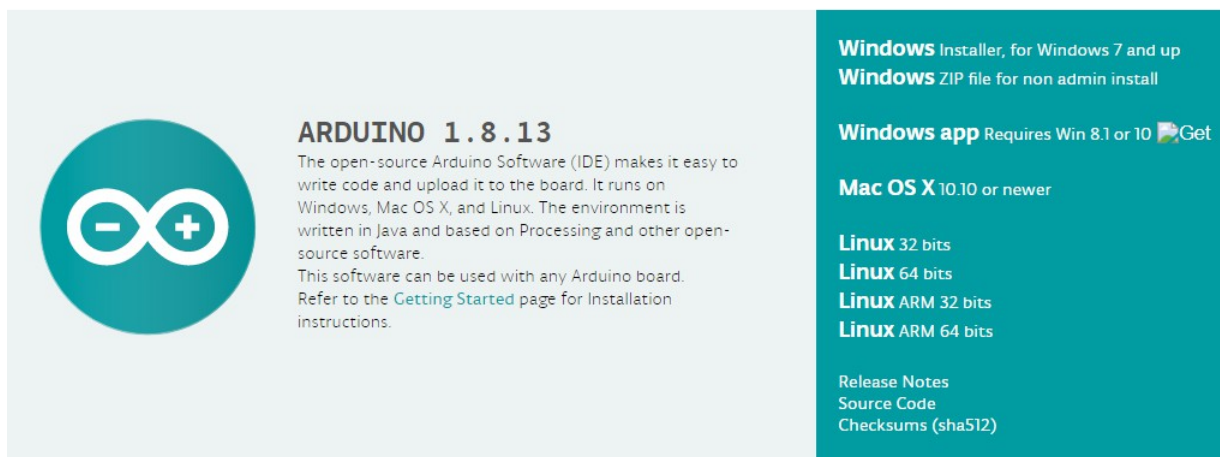
El sensor Hall se basa en una resistencia de N portadoras. Cuando el chip se encuentra en el campo magnético, el sensor Hall desarrolla una pequeña tensión en la resistencia, que puede ser medida directamente por el convertidor analógico-digital (ADC), o amplificada por el preamplificador analógico de ruido ultrabajo y luego medida por el ADC.

El sensor de temperatura genera una tensión que varía con la temperatura. La tensión se convierte internamente mediante un convertidor analógico-digital en un código digital. El sensor de temperatura tiene un rango de -40°C a 125°C. Dado que el desplazamiento del sensor de temperatura varía de un chip a otro debido a la variación del proceso, junto con el calor generado por el propio circuito Wi-Fi (que afecta a las mediciones), el sensor de temperatura interno sólo es adecuado para aplicaciones que detectan cambios de temperatura en lugar de temperaturas absolutas y también con fines de calibración. Sin embargo, si el usuario calibra el sensor de temperatura y utiliza el dispositivo en una aplicación con una alimentación mínima, los resultados podrían ser lo suficientemente precisos.

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 large teal circle containing the Arduino logo (an infinity symbol with a minus sign on the left and a plus sign on the right). To the right of the logo, the text reads: **ARDUINO 1.8.13**. Below this, it says: "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 the following links and 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](#)
- [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*.

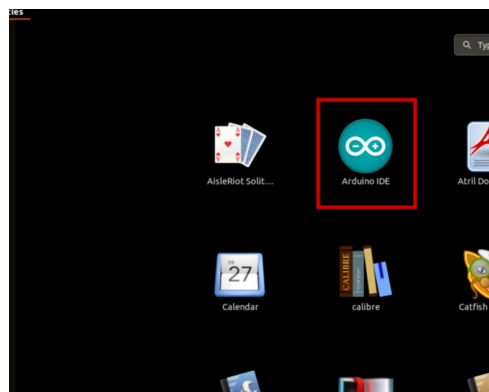
Para ejecutar el primer script en el terminal, abra el terminal en el directorio extraído y ejecute el siguiente comando:

```
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 está instalado el *IDE Arduino*.

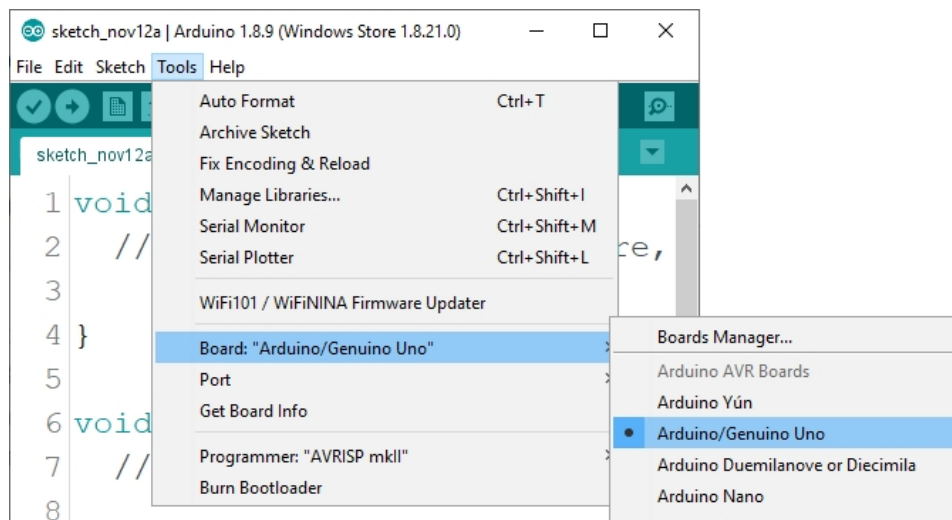


Az-Delivery

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 libro electrónico.

Lo siguiente es comprobar si tu PC puede detectar una placa Arduino. Abra Arduino IDE recién instalado, y vaya a:

Herramientas > Tablero > {nombre de su tablero aquí}
{su nombre de tablero aquí}



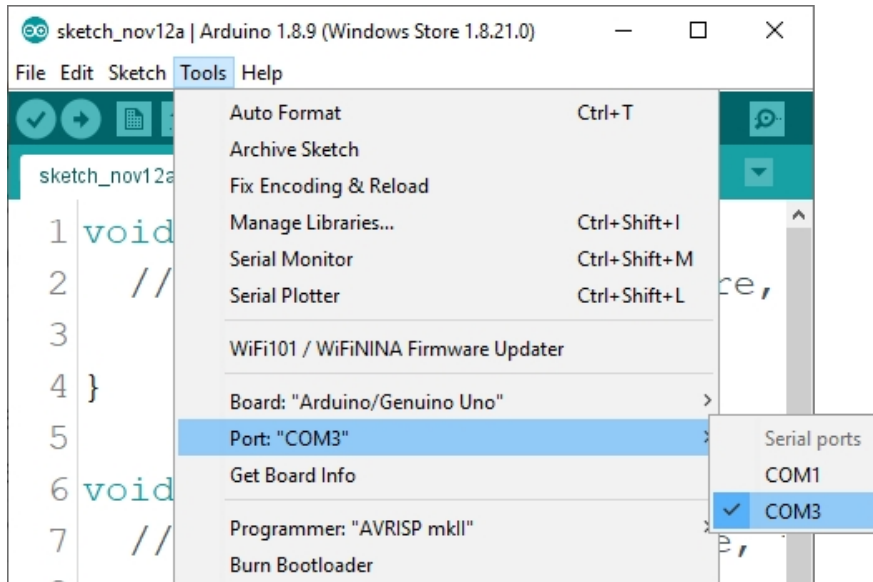
Hay que seleccionar el puerto al que está conectada la placa Arduino. Ir a:

Herramientas > Puerto > {el nombre del puerto va aquí}

y cuando la placa Arduino 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

Para utilizar el D1 R32 con el IDE de Arduino, 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 pueden elegir en función del que se utilice. Los drivers se pueden descargar desde el siguiente [enlace](#).

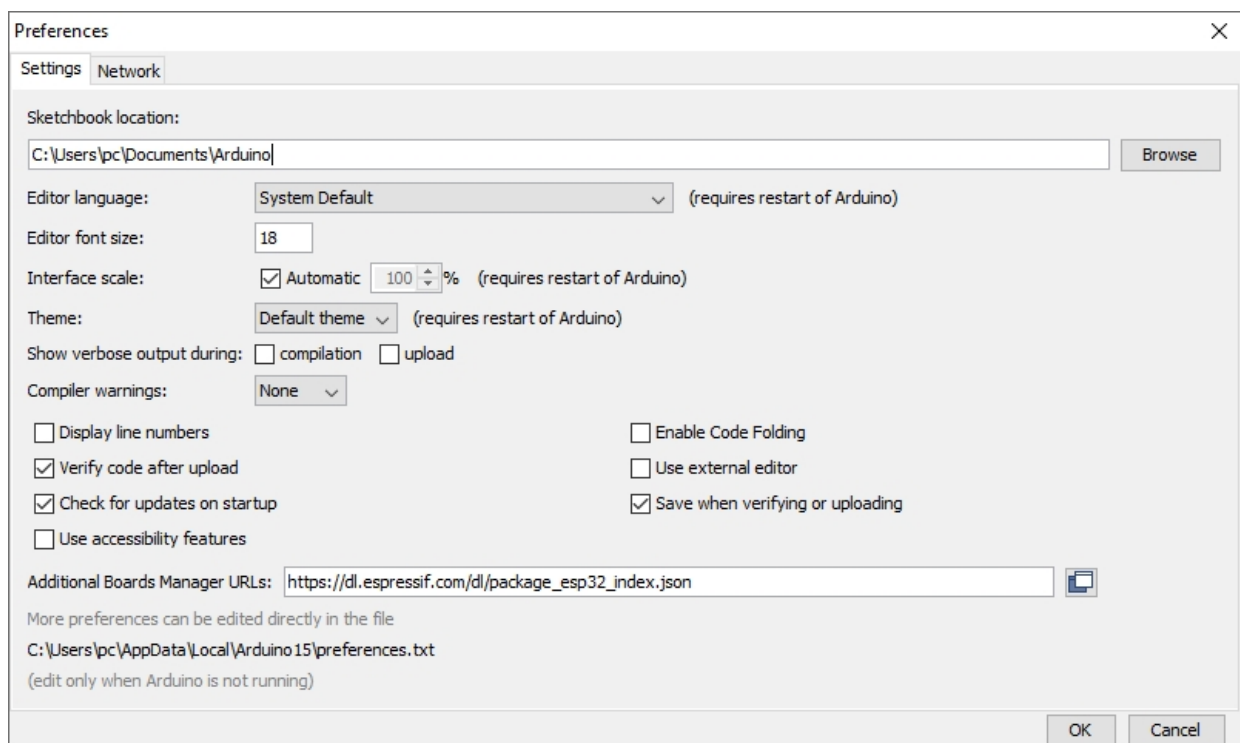
Az-Delivery

A continuación, para instalar el soporte para la plataforma ESP32, abra Arduino IDE y vaya a:

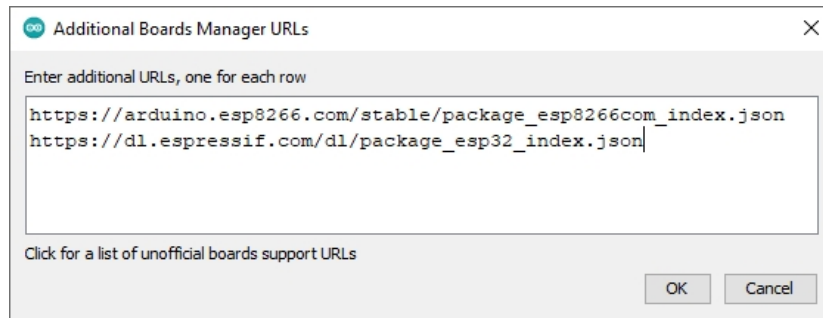
Archivo > Preferencias, y busque el campo URLs adicionales.

A continuación, copie la siguiente URL:

https://dl.espressif.com/dl/package_esp32_index.json



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**.



Cierra el IDE Arduino y ábrelo de nuevo, luego ve a:

Herramientas > Tablero > Administrador de tableros

Cuando se abra la nueva ventana, escriba **esp32** en el cuadro de búsqueda e instale la placa llamada **esp32** fabricada por *Espressif Systems*, como se muestra en la siguiente imagen:



Para seleccionar la placa ESP32, vaya a:

Herramientas > Placa > ESP32 Arduino > Módulo de desarrollo ESP32

Para subir el código del sketch a la placa ESP32, primero selecciona el puerto al que está conectada la placa. Vaya a: *Herramientas > Puerto > {nombre del puerto}*

Ejemplos de bocetos

LED parpadeante

```
int ledPin = 2;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```



PWM - Modulación por ancho de impulsos

```
#define LEDC_CHANNEL_0 0
#define LEDC_TIMER_13_BIT 13
#define LEDC_BASE_FREQ 5000
#define LED_PIN 2

int brightness = 0;
int fadeAmount = 5;

void ledcAnalogWrite(uint8_t channel, uint32_t value, uint32_t valueMax = 255)
{
    uint32_t duty = (8191 / valueMax) * min(value, valueMax);
    ledcWrite(channel, duty);
}

void setup() {
    ledcSetup(LEDC_CHANNEL_0, LEDC_BASE_FREQ, LEDC_TIMER_13_BIT);
    ledcAttachPin(LED_PIN, LEDC_CHANNEL_0);
}

void loop() {
    ledcAnalogWrite(LEDC_CHANNEL_0, brightness);
    brightness = brightness + fadeAmount;
    if (brightness <= 0 || brightness >= 255) {
        fadeAmount = -fadeAmount;
    }
    delay(30);
}
```



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 está buscando productos de alta calidad para Arduino y Raspberry Pi, 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>