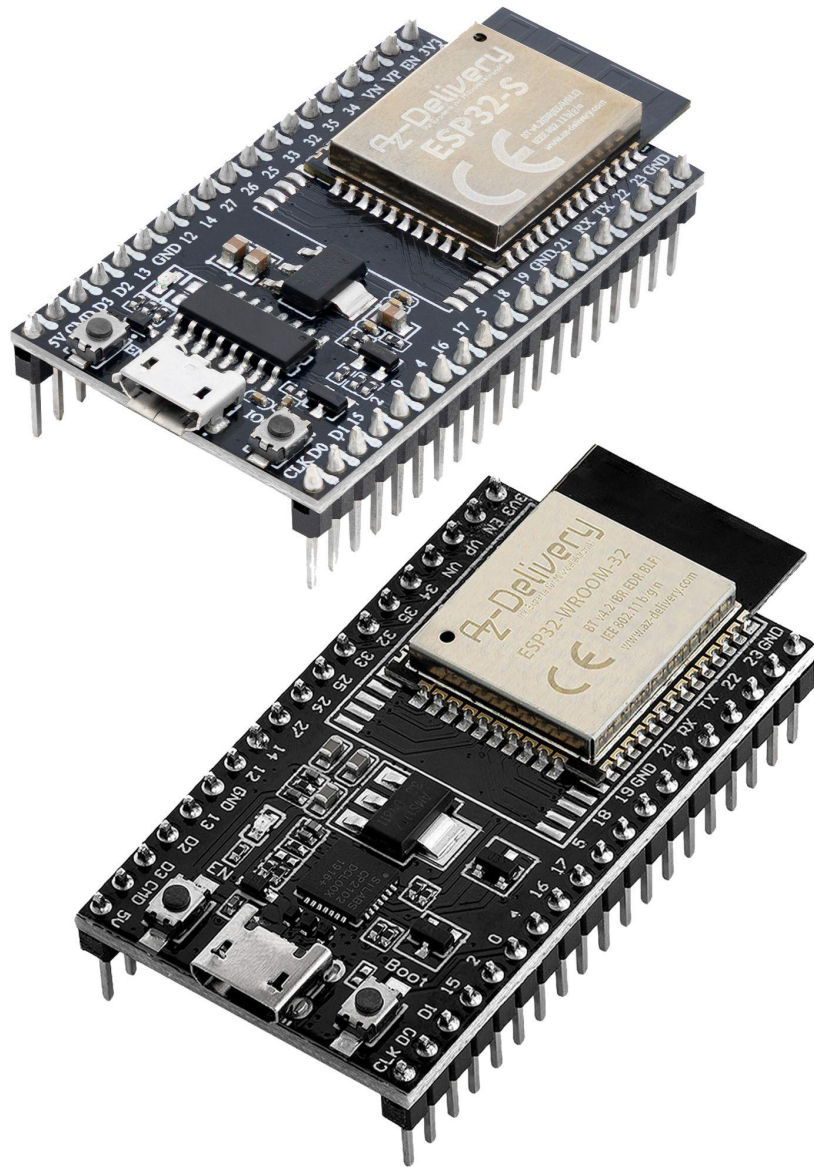


ESP32 Dev Kit C V4

e-book

ESP32 Dev Kit C V4

con CP2102 o CH340



Índice de contenidos

Descripción	3
Especificaciones	4
ESP32 Dev Kit C V4	5
Pinout	6
Descripción de los pines	7
Clavijas del sensor táctil capacitivo	8
Pines del convertidor analógico-digital	8
Pines del convertidor de digital a analógico	8
Reloj en tiempo real Pines GPIO	9
Reloj en tiempo real Pines GPIO	9
Los pines de la interfaz I2C	9
Pines de la interfaz SPI	10
Pernos de flejado	10
Pins HIGH at Boot	10
Activación (EN)	11
Comunicación de USB a serie	11
Comunicación Wi-Fi	11
Comunicación Bluetooth	12
Instalación del IDE de Arduino	13
Configuración adicional	17
Ejemplo de cableado del ESP32 Dev Kit C V4	20
Ejemplos de bocetos	21

Descripción

El ESP32 Dev Kit C V4 es una placa de desarrollo creada alrededor del chip ESP32 WROOM 32, que contiene un regulador de voltaje y un circuito programador USB para el chip ESP32, y algunas otras muchas características.

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

ESP32 Dev Kit C V4 viene con un firmware preinstalado que permite trabajar con el lenguaje interpretado, enviando comandos a través del puerto serie (chip CP2102/CH340). Las placas ESP32 son una de las plataformas más utilizadas para proyectos de Internet de las Cosas (IoT).

La placa ESP32 Dev Kit C V4 está especialmente diseñada para trabajar en breadboards. Tiene un regulador de voltaje que le permite alimentarse directamente del puerto USB. Los pines de entrada/salida funcionan a 3,3V. El chip CP2102(CH340) es el responsable de la comunicación USB a serie.

Specifications

Tensión de alimentación (USB)	5V DC
Tensión de entrada/salida	3.3 V DC
Corriente de trabajo requerida	min. 500 mA
SoC	ESP32-WROOM 32
CPU	Xtensa® single-dual-core 32-bit LX6
Gama de frecuencias de reloj	80 MHz / 240 MHz
RAM	512 kB
Memoria flash externa	4 MB
I/O pins	34
Canales ADC	18
Resolución ADC	12-bit
Canales DAC	2
Resolución del DAC	8-bit
Interfaces de comunicación	SPI, I2C, I2S, CAN, UART, PWM, SDIO, GPIO, ADC, DAC
Protocolos Wi-Fi	802.11 b/g/n (802.11n up to 150 Mbps)
Frecuencia Wi-Fi	2.4 GHz – 2.5 GHz
Bluetooth	V4.2 - BLE y Bluetooth clásico
Antena inalámbrica	PCB
Dimensiones	56 × 28 × 13 mm (2.2 × 1.1 × 0.5 in)

ESP32 Dev Kit C V4

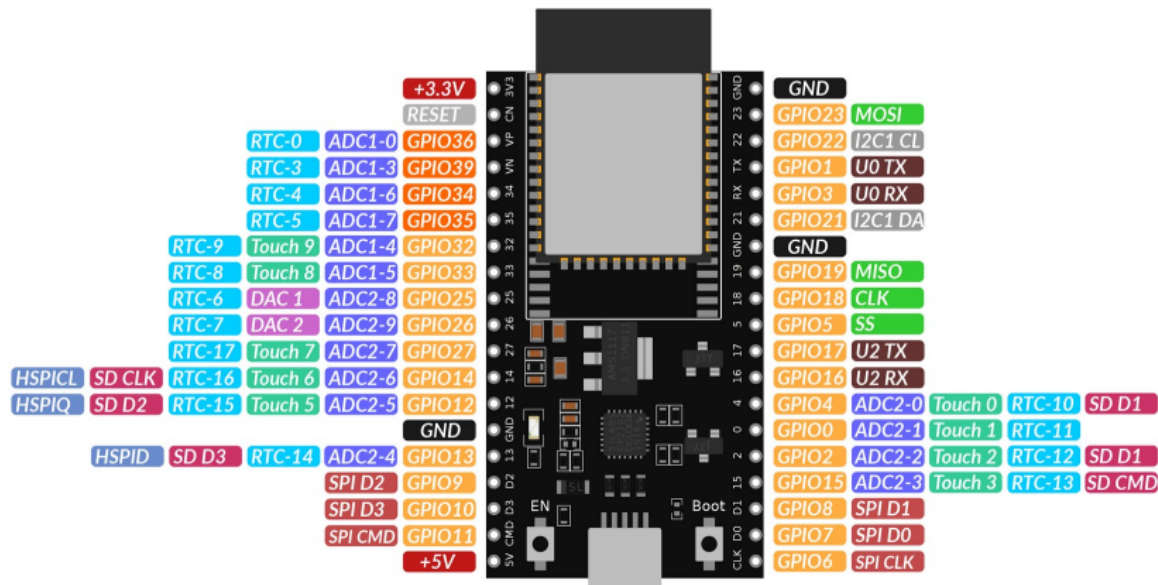
La serie de chips Wi-Fi ESP32 WROOM 32 es producida por Espressif Systems. El ESP32 WROOM-32 es un módulo Wi-Fi asequible adecuado para proyectos de bricolaje en el campo del Internet de las cosas (IoT). Este módulo viene con muchos GPIOs y soporte para una variedad de protocolos como SPI, I2C, I2S, UART, y más. La mejor parte 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 los 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 de recepción de bajo ruido, filtros y módulos de gestión de energía en un pequeño paquete. Ofrece Wi-Fi de 2,4 GHz (802.11 b/g/n, que admite velocidades de hasta 150 MB/s), comunicación inalámbrica BLE y Bluetooth® clásica, 34 pines de E/S, interfaces I2C e I2S, ADC (conversión analógica a digital), DAC (conversión digital a analógica), interfaz SPI, UART en pines dedicados y PWM (modulación de ancho de pulso).

El núcleo del procesador, denominado LX6 por Espressif, se basa en el controlador del procesador LX6 de doble núcleo Xtensa® de 32 bits y funciona a una frecuencia de entre 80 y 240MHz. Tiene una ROM de arranque de 448kB, 520kB de SRAM en el chip y 4MB de memoria flash externa a la que se puede acceder a través de la interfaz SPI.

Pinout

El ESP32 Dev Kit C V4 tiene 38 pines. El pinout se muestra en la siguiente imagen:



- Orange Digital In/Out ports (all support PWM)
- Dark orange Digital Input ports
- Blue Analog Input 12 bits, 0 to 3.3V
- Purple Analog Output 8 bits, 0 - 3.3V
- Green Capacitive Touch Sensor ports
- Light blue I/O -pins from RTC ultra low power processor, usable in deep sleep mode
- Pink SD card interface
- Red SPI bus for Flash-memory, do not use

The following pins show the default assignment. All these signals can be changed to any In/Out port. This applies also to UART0 and UART1, which cannot be accessed in the default assignment.

- Grey I2C bus (Wire)
- Green VSPI bus
- Brown Serial interfaces
- Blue HSPI bus

NOTE: La corriente máxima absoluta consumida por un GPIO es de 10mA.

Descripción de los pines

Al igual que una placa Arduino normal, el ESP32 Dev Kit C V4 tiene pines digitales de entrada/salida (pines GPIO - General Purpose Input/Output pins). Estas entradas/salidas digitales funcionan a 3,3V.

La tensión 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 entrada solamente. Estos pines no tienen resistencias internas de pull-up o pull-down. No pueden ser utilizados como salidas, así que utiliza estos pines sólo como entradas: GPIO 34, GPIO 35, GPIO 36, GPIO 39. Hay una flash SPI integrada en el chip ESP-WROOM-32. Los pines GPIO6 a GPIO 11 están expuestos en ciertas placas de desarrollo ESP32.

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

GPIO 6 (SCK/CLK), GPIO 7 (SD0/SD0), GPIO 8 (SD1/SD1), GPIO 9 (SHD/SD2), GPIO 10 (SWP/SD3), GPIO 11 (CSC/CMD).

Capacitive Touch sensor pins

El ESP32 tiene 10 sensores táctiles capacitivos internos. Los pines táctiles capacitivos también se pueden utilizar para despertar el ESP32 del sueño profundo. Estos sensores táctiles internos están conectados a estos GPIOs:

T0 (GPIO 4), T1 (GPIO 0), T2 (GPIO 2), T3 (GPIO 15), T4 (GPIO 13), T5 (GPIO 12), T6 (GPIO 14), T7 (GPIO 27), T8 (GPIO 33), T9 (GPIO 32).

Pines del convertidor analógico-digital

El ESP32 tiene 18x12 bits ADC (Analog to Digital converter) canales de entrada (mientras que el ESP8266 sólo tiene 1x 10 bits ADC). Estos son los GPIOs que se pueden utilizar como ADC y los canales respectivos:

**ADC1_CH0 (GPIO 36), ADC1_CH1 (GPIO 37), ADC1_CH2 (GPIO 38),
ADC1_CH3 (GPIO 39), ADC1_CH4 (GPIO 32), ADC1_CH5 (GPIO 33),
ADC1_CH6 (GPIO 34), ADC1_CH7 (GPIO 35), ADC2_CH0 (GPIO 4),
ADC2_CH1 (GPIO 0), ADC2_CH2 (GPIO 2), ADC2_CH3 (GPIO 15),
ADC2_CH4 (GPIO 13), ADC2_CH5 (GPIO 12), ADC2_CH6 (GPIO 14),
ADC2_CH7 (GPIO 27), ADC2_CH8 (GPIO 25), ADC2_CH9 (GPIO 26)**

Pines del convertidor de digital a analógico

Hay 2 canales DAC (convertidor de digital a analógico) de 8 bits en el ESP32 para convertir las señales digitales en salidas de señal de tensión analógica. 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 GPIOs enrutados al subsistema de bajo consumo RTC se pueden utilizar cuando el ESP32 está en reposo profundo. Estos GPIOs RTC pueden utilizarse para despertar al ESP32 del sueño profundo cuando el coprocesador de ultra bajo consumo (ULP) está en funcionamiento. Los siguientes GPIOs se pueden utilizar como fuente de activación externa:

**RTC_GPIO0 (GPIO36), RTC_GPIO3 (GPIO39), RTC_GPIO4 (GPIO34),
RTC_GPIO5 (GPIO35), RTC_GPIO6 (GPIO25), RTC_GPIO7 (GPIO26),
RTC_GPIO8 (GPIO33), RTC_GPIO9 (GPIO32), RTC_GPIO10 (GPIO4),
RTC_GPIO11 (GPIO0), RTC_GPIO12 (GPIO2), RTC_GPIO13 (GPIO15),
RTC_GPIO14 (GPIO13), RTC_GPIO15 (GPIO12), RTC_GPIO16 (GPIO14),
RTC_GPIO17 (GPIO27).**

Reloj en tiempo real Pines GPIO

El controlador PWM (modulación de ancho de pulso) del LED ESP32 tiene 16 canales independientes que pueden ser configurados para generar señales PWM con diferentes propiedades. Todos los pines que pueden actuar como salidas pueden ser utilizados como pines PWM (los GPIOs 34 a 39 no pueden generar PWM). Para configurar una señal PWM, es necesario definir estos parámetros en el código: Frecuencia de la señal, ciclo de trabajo, canal PWM, GPIO por donde se quiere dar salida a la señal.

Los pines de la interfaz I2C

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

GPIO 21 (SDA), GPIO 22 (SCL).

Pines de la interfaz SPI

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

SPI	MOSI	MISO	CLK	CS
VSPI	GPIO 23	GPIO 19	GPIO 18	GPIO 5
HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15

Pernos de flejado

Los siguientes pines se utilizan para poner el ESP32 en modo bootloader o flasheo: **GPIO 0, GPIO 2, GPIO 4, GPIO 5 (debe ser HIGH durante el arranque), GPIO 12 (debe ser LOW durante el arranque), GPIO 15 (debe ser HIGH 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 algunos periféricos están conectados a los pines de arranque y el IDE no puede cargar el código o flashear el ESP32, puede ser porque esos periféricos están impidiendo que el ESP32 entre en el modo correcto. Después de reiniciar, flashear o arrancar, esos pines funcionan como se espera. Más y más extensas explicaciones no están en el alcance de este libro electrónico, así que por favor, consulte la hoja de datos.

Alfileres HIGH en Boot

Algunos GPIOs cambian su estado a HIGH o emiten señales PWM al arrancar o reiniciar. Esto significa que si se conectan salidas a estos GPIOs se pueden obtener resultados inesperados cuando el ESP32 se reinicia o arranca. GPIO 1,

ESP32 Dev Kit C V4

GPIO 3, GPIO 5, GPIO 6 a GPIO 11 (conectados a la memoria flash SPI integrada del ESP32 - no se recomienda su uso), GPIO 14, GPIO 15.

Activación (EN)

Enable (EN) es el pin de habilitación del regulador de 3,3V. Tiene un estado de pull up, y necesita ser conectado a tierra para desactivar el regulador de 3,3V. Esto significa que este pin puede ser conectado a un botón para reiniciar su ESP32, por ejemplo.

Comunicación de USB a serie

El ESP32 Dev Kit C V4 tiene un puerto de conexión microUSB. Está hecho alrededor del chip CP2102 fabricado por Silicon Laboratories (alternativamente el chip CH340 fabricado por WCH) que permite la comunicación serial USB a UART. El chip tiene la característica de puerto COM virtual (VCP) que aparece como un puerto COM en las aplicaciones de PC. La interfaz USB UART implementa todas las señales RS-232, incluyendo las señales de control y handshaking, por lo que no es necesario modificar el firmware del sistema existente. Para poder utilizar el ESP32 es necesario instalar el controlador.

Comunicación Wi-Fi

ESP32 Dev Kit C V4 tiene integrada la interfaz de comunicación Wi-Fi y puede funcionar en tres modos diferentes: Estación Wi-Fi, punto de acceso Wi-Fi y ambos al mismo tiempo. Soporta las siguientes características:

- Velocidades de datos 802.11b y 802.11g
- 802.11n MCS0-7 en los anchos de banda de 20MHz y 40MHz
- 802.11n MCS32

ESP32 Dev Kit C V4

- 802.11n Intervalo de guarda de 0,4µS
- Velocidad de datos de hasta 150 Mbps
- Recepción STBC 2×1
- Hasta 20 dBm de potencia de transmisión
- Potencia de transmisión ajustable
- Diversidad y selección de antenas (hardware gestionado por software)

Comunicación Bluetooth

El ESP32 Dev Kit C V4 tiene una radio Bluetooth integrada y soporta las siguientes características:

- Potencias de salida de transmisión de clase 1, clase 2 y clase 3 y rango de control dinámico de más de 30 dB
- Modulación $\pi/4$ DQPSK y 8 DPSK
- Alto rendimiento en la 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 toda velocidad, la mezcla de voz y datos y el funcionamiento completo de la piconet
- Lógica para la corrección de errores hacia adelante, control de errores de cabecera, correlación de códigos de acceso, CRC, demodulación, generación de flujos de bits de encriptación, blanqueo y conformación de impulsos de transmisión
- ACL, SCO, eSCO and AFH
- CODEC de audio digital A-law, μ -law y CVSD en interfaz PCM
- CODEC de audio SBC
- Gestión de la 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 HCI SDIO / SPI


ESP32 Dev Kit C V4

- Interfaz I2C
- Interfaz de audio PCM / I2S

Instalación del IDE de Arduino

Puedes descargar el entorno de desarrollo gratuito Arduino IDE desde el siguiente enlace: <https://www.arduino.cc/en/Main/Software>

Downloads



Arduino IDE 2.0.0

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

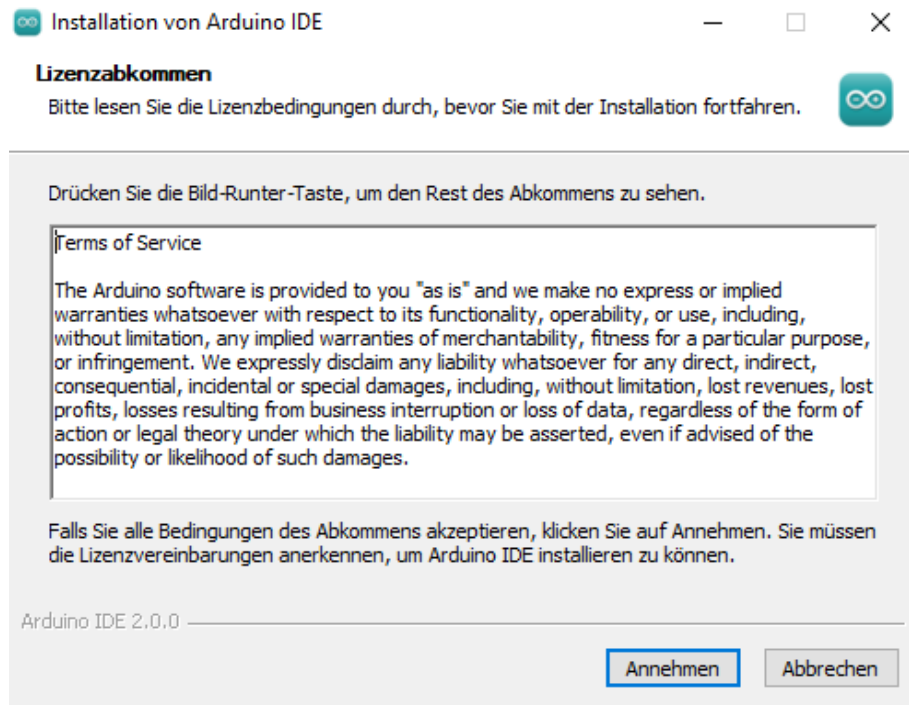
Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS 10.14: "Mojave" or newer, 64 bits

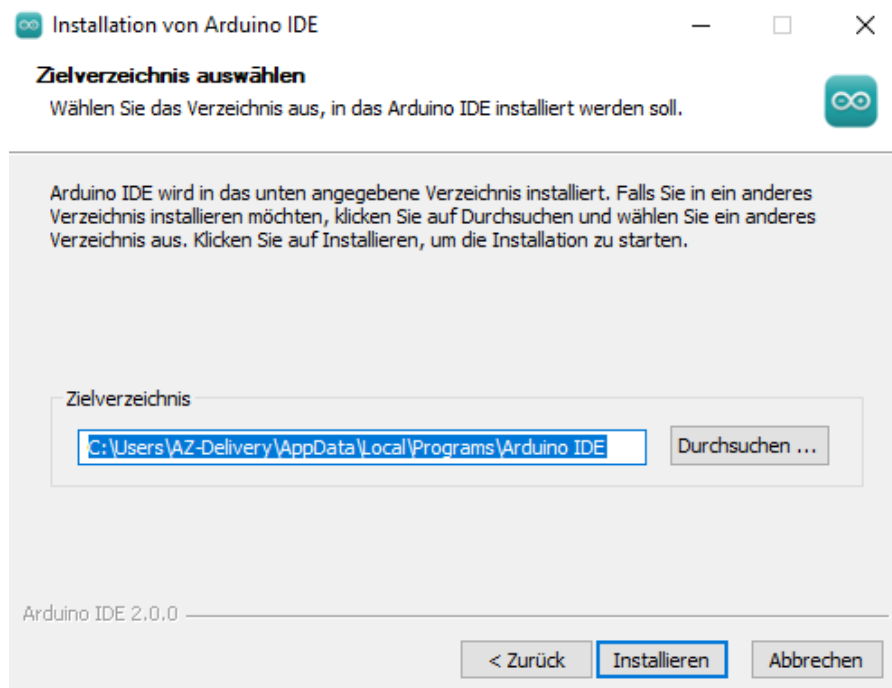
Después de iniciar el archivo de instalación de Arduino IDE

"arduino-ide_2.0.0_Windows_64bit.exe" hay que leer y aceptar las condiciones de la licencia del software.



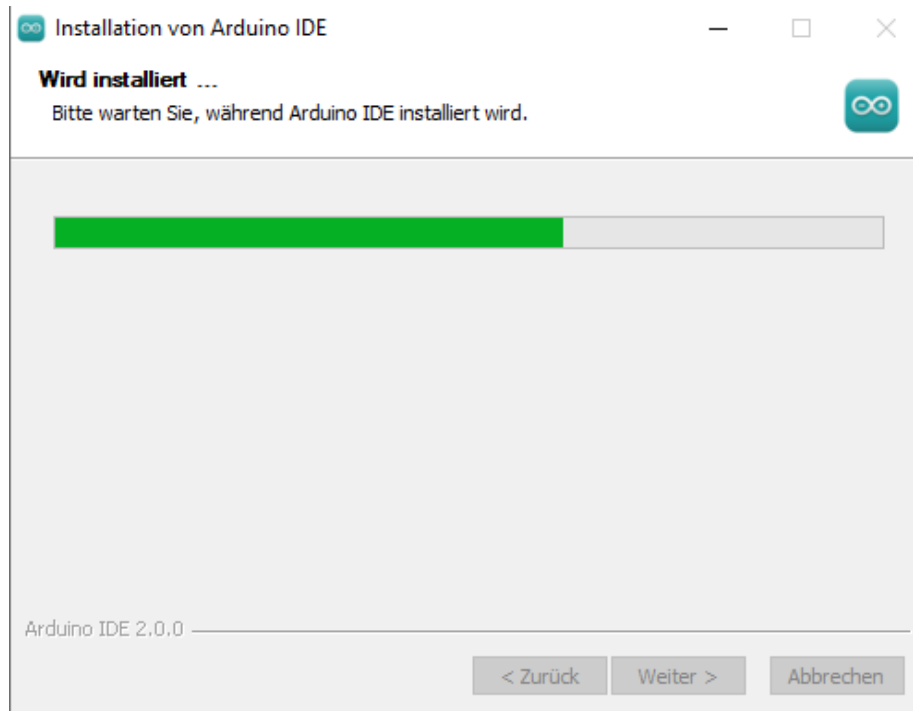
En el siguiente paso, se pueden seleccionar diferentes opciones para la instalación.

Por último, hay que especificar la carpeta de destino. La instalación requiere aproximadamente 500 MB de espacio libre en el disco.

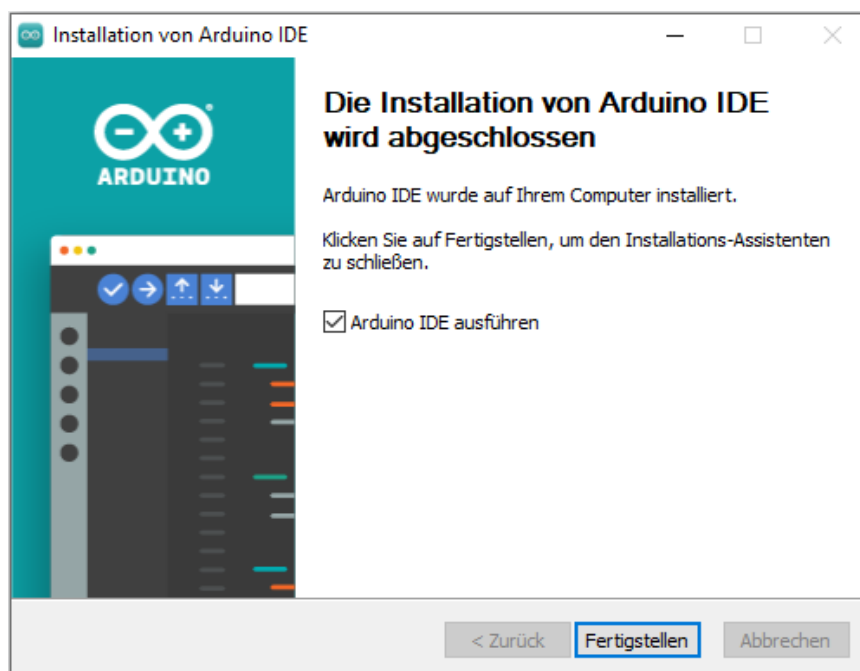


Haga clic en "Instalar" para iniciar la instalación.

ESP32 Dev Kit C V4



Después de la instalación exitosa, el programa de instalación puede ser terminado a través del botón "Finish".



Configuración adicional

Para utilizar el ESP32 Dev Kit C V4 con el IDE de Arduino, siga unos sencillos pasos. Antes de configurar el IDE de Arduino, hay que instalar el driver para la comunicación USB a Serie. Si el controlador no se instala automáticamente, puede descargar e instalar el controlador aquí:

For CP2102:

<https://www.silabs.com/interface/usb-bridges/classic/device.cp2102>

For CH340:

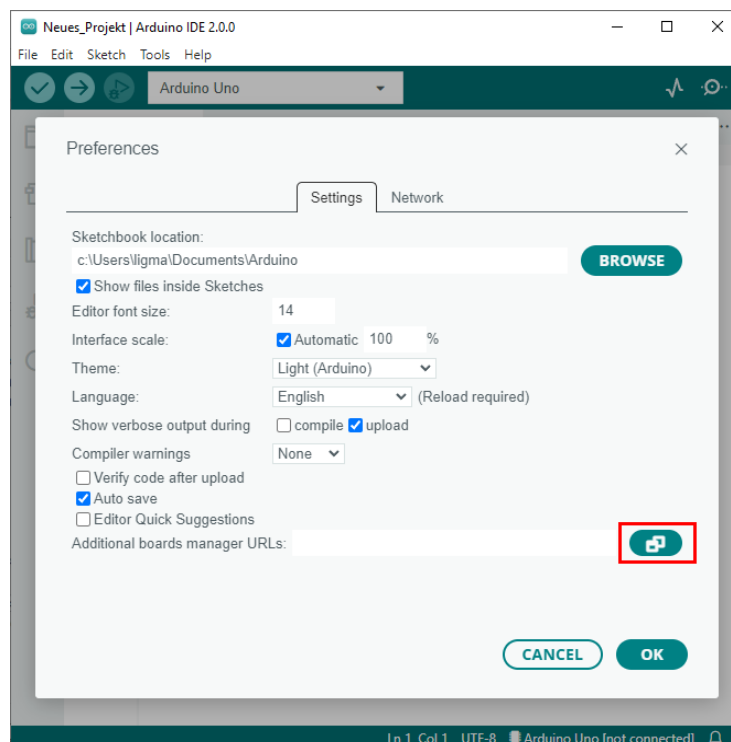
http://www.wch.cn/download/CH341SER_ZIP.html

A continuación, para instalar el soporte para la plataforma ESP32, abra el IDE de Arduino 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

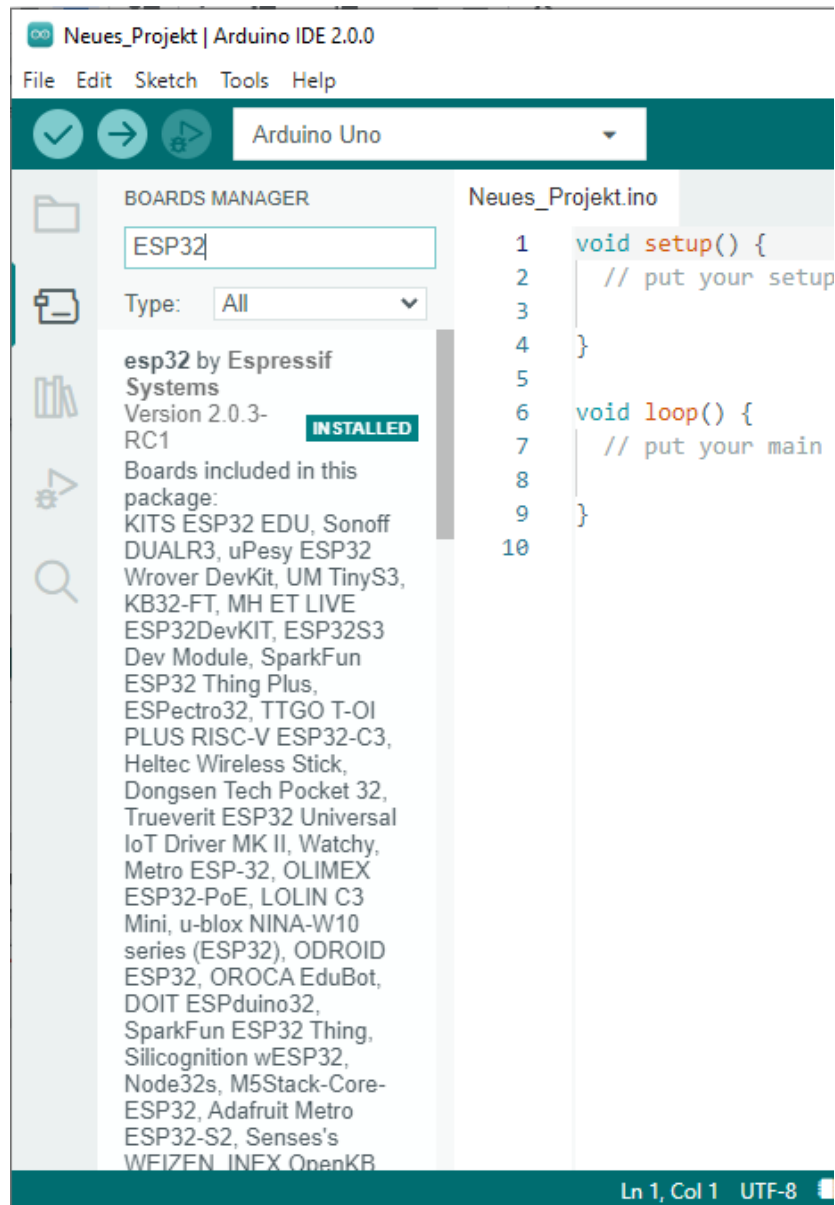


ESP32 Dev Kit C V4

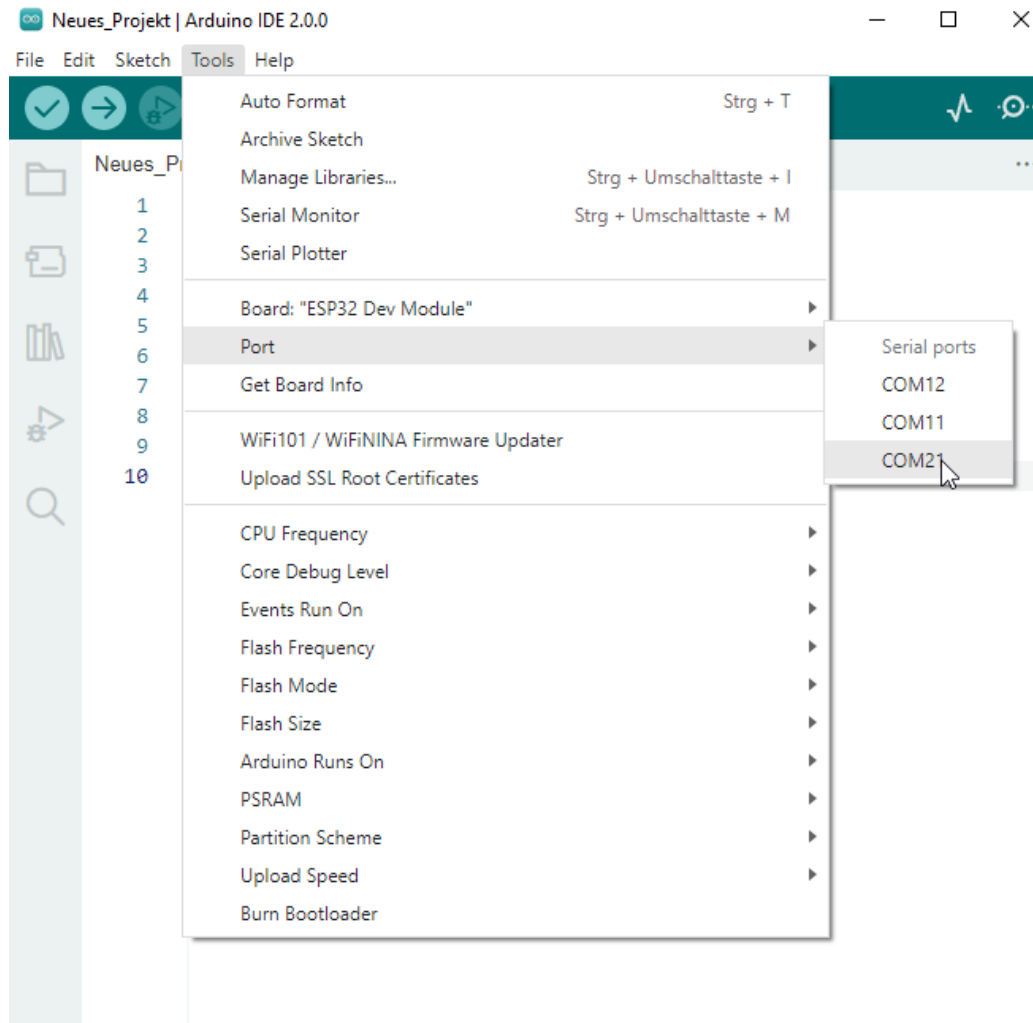
Pegue este enlace en el campo URLs adicionales. Si hay uno o más enlaces dentro de este campo, simplemente añada una coma después del último enlace, pegue un nuevo enlace después de la coma y haga clic en el botón Aceptar.

Abre de nuevo el IDE de Arduino y ve a Herramientas > Placas > Gestor de Placas

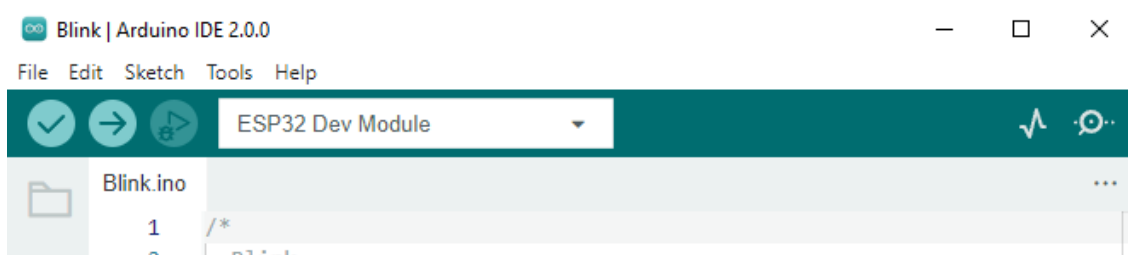
Cuando se abra la nueva ventana, escribe esp32 en el cuadro de búsqueda e instala la placa llamada esp32 fabricada por Espressif Systems, como se muestra en la siguiente imagen:



Para seleccionar la placa ESP32, vaya a: Herramientas > Placa > ESP32 Arduino > ESP32 Dev Module



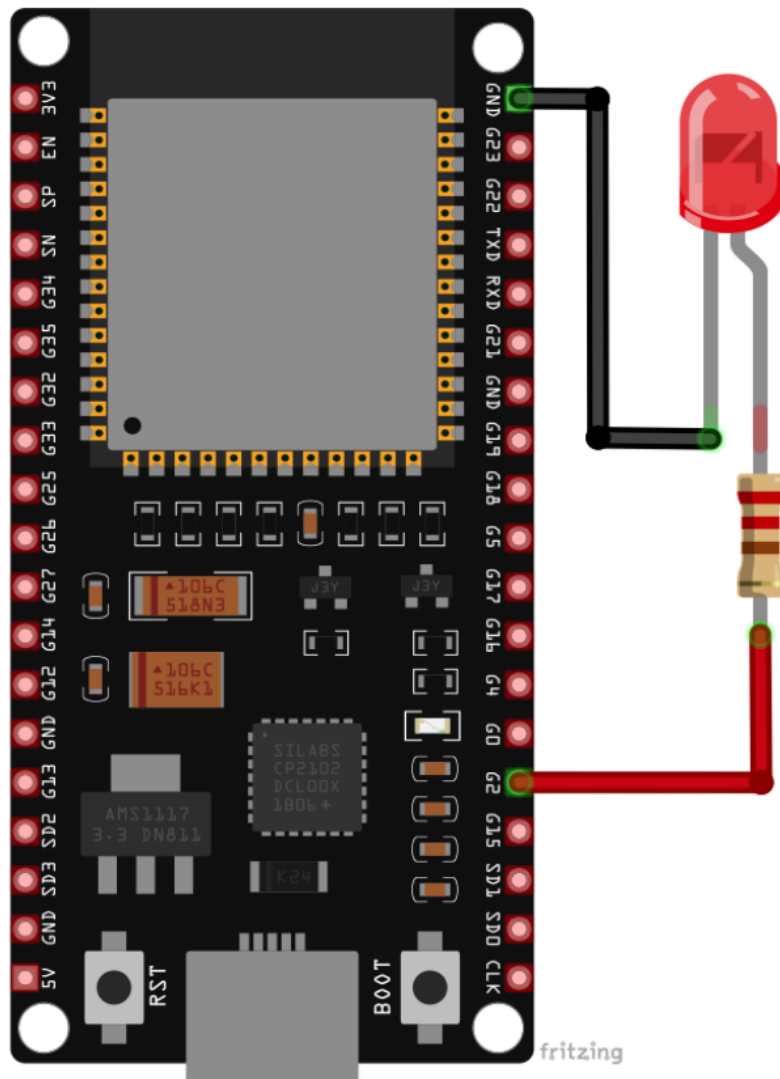
Para subir el código del sketch a la placa ESP32, primero selecciona el puerto en el que has conectado la placa. Ve a: Herramientas > Puerto > {nombre del puerto}



ESP32 Dev Kit C V4

Ejemplo de cableado del ESP32 Dev Kit C V4

Conecte el ESP32 Dev Kit C V4 con un LED y una resistencia como se muestra en el siguiente diagrama de conexión:



ESP32 Dev Kit C V4 pin	LED pin	Color del cable
GPIO2 (pin2)	Ánodo (+) a través de la resistencia	Cable rojo
GND	Cátodo (-)	Cable negro

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 pulsos

```
#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);
}
```

ESP32 Dev Kit C V4

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 para conseguirlos. Dispondrá de numerosos ejemplos de aplicación, guías de instalación completas, libros electrónicos, bibliotecas y asistencia de nuestros expertos técnicos.

<https://az-delivery.de>

¡Diviértete!

Impressum

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