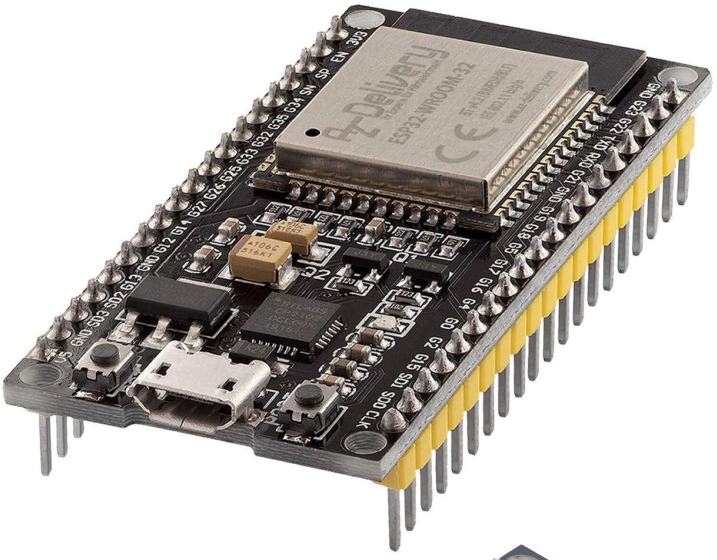
e-book

ESP32 Dev Kit C V2

with CP2102 or CH340



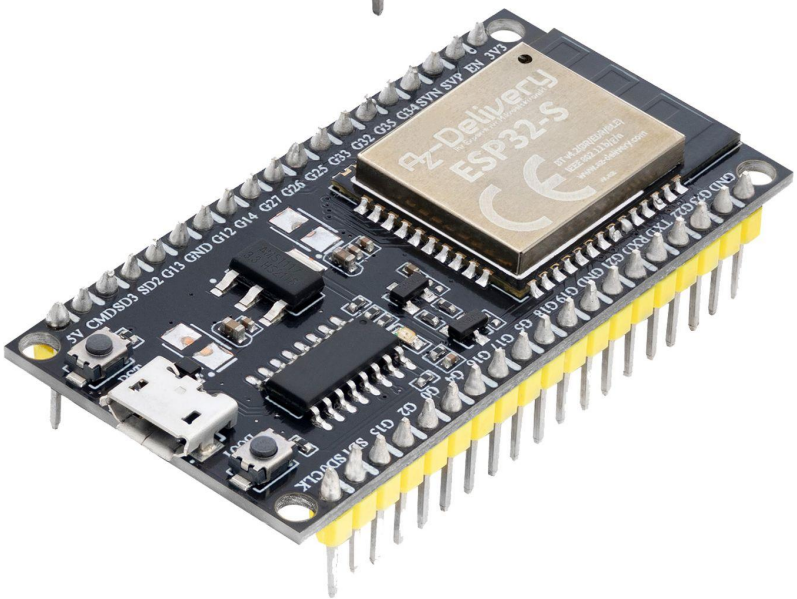


Table of Contents

Description

The ESP32 Dev Kit C V2 is a development board created around ESP32 WROOM 32 chip, containing voltage regulator and USB programmer circuit for ESP32 chip, and a few many other features.

For application development, there is a choice between Arduino IDE or ESP-IDF (Native platform). Most users choose the Arduino IDE because of its simplicity and compatibility. The Arduino user community is very active and supports platforms such as ESP32.

ESP32 Dev Kit C V2 comes with a pre-installed firmware which allows working with the interpreted language, sending commands through the serial port (CP2102/CH340 chip). The ESP32 boards are one of the most used platforms for Internet of Things (IoT) projects.

The ESP32 Dev Kit C V2 board is specially designed to work on breadboards. It has a voltage regulator that allows it to feed directly from the USB port. The input/output pins work at 3.3V. The CP2102(CH340) chip is responsible for USB to serial communication.

Specifications

Power supply voltage (USB)	5V DC
Input/Output voltage	3.3 V DC
Operating current required	min. 500 mA
SoC	ESP32-WROOM 32
CPU	Xtensa® single-dual-core 32-bit LX6
Clock frequency range	80 MHz / 240 MHz
RAM	512 kB
External flash memory	4 MB
I/O pins	34
ADC channels	18
ADC resolution	12-bit
DAC channels	2
DAC resolution	8-bit
Communication interfaces	SPI, I2C, I2S, CAN, UART, PWM, SDIO, GPIO, ADC, DAC
Wi-Fi protocols	802.11 b/g/n (802.11n up to 150 Mbps)
Wi-Fi frequency	2.4 GHz – 2.5 GHz
Bluetooth	V4.2 – BLE and Classic Bluetooth
Wireless antenna	PCB
Dimensions	56 × 28 × 13 mm (2.2 × 1.1 × 0.5 in)

ESP32 Dev Kit C V2

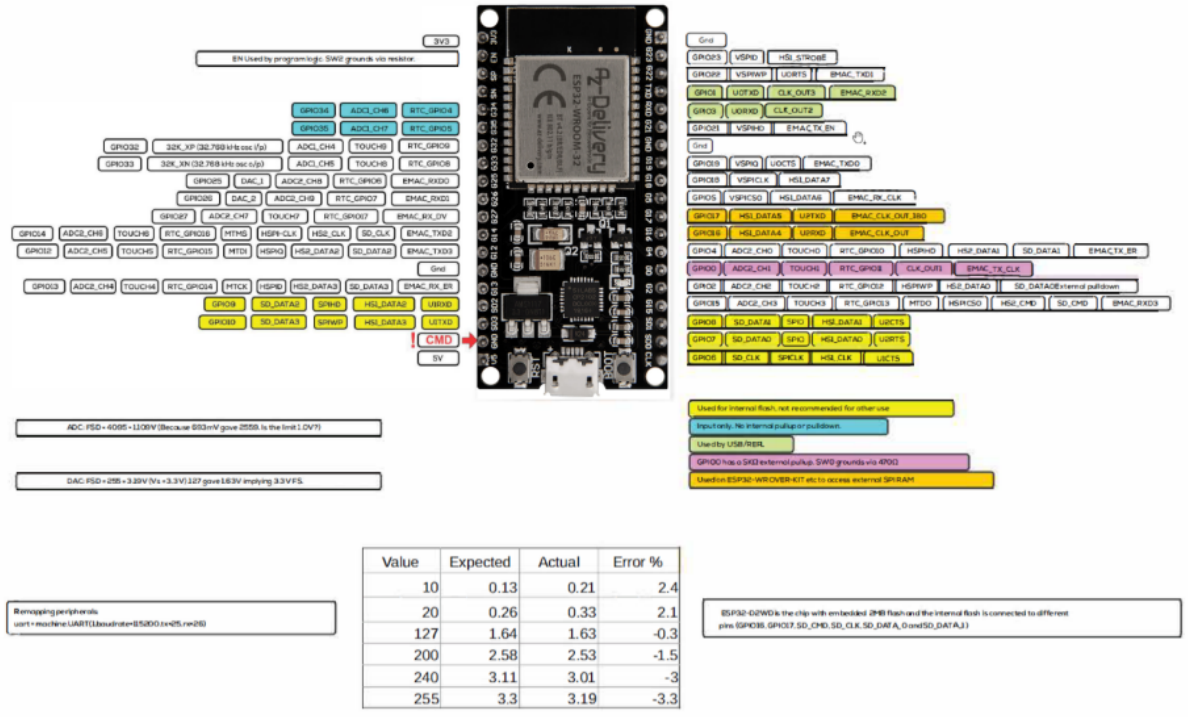
The ESP32 WROOM 32 series of Wi-Fi chips is produced by Espressif Systems. ESP32 WROOM-32 is an affordable Wi-Fi module suited for DIY projects in the Internet of Things (IoT) field. This module comes with many GPIOs and support for a variety of protocols like SPI, I2C, I2S, UART, and more. The best part is that it comes with wireless networking included, which makes it different to other microcontrollers like the Arduino. This means that it can easily control and monitor devices remotely via Wi-Fi and Bluetooth® at an affordable price.

ESP32 WROOM 32 is a system-on-chip (SoC) integrating a 32-bit Tensilica microcontroller, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules into a small package. It provides 2.4GHz Wi-Fi (802.11 b/g/n, supporting speeds up to 150MB/s), BLE and classic Bluetooth® wireless communication, 34 I/O pins, I2C and I2S interfaces, ADC (analog to digital conversion), DAC (digital to analog conversion), SPI interface, UART on dedicated pins, and PWM (Pulse Width Modulation).

The processor core, called LX6 by Espressif, is based on Xtensa® dual core 32-bit LX6 processor controller and runs at a frequency range between 80-240MHz. It has a 448kB boot ROM, 520kB of on-chip SRAM, and 4MB of external flash memory which can be accessed through SPI interface.

Pinout

The ESP32 Dev Kit C V2 has 38 pins. The pinout is shown on the following image:



Value	Expected	Actual	Error %
10	0.13	0.21	2.4
20	0.26	0.33	2.1
127	1.64	1.63	-0.3
200	2.58	2.53	-1.5
240	3.11	3.01	-3
255	3.3	3.19	-3.3

NOTE: The absolute maximum current drawn per one GPIO is 10mA.

Pins description

Just like a normal Arduino board, the ESP32 Dev Kit C V2 has digital input/output pins (GPIO pins - General Purpose Input/Output pins). These digital input/outputs operate at 3.3V.

5V voltage must not be connected to any ESP32 chip pins!

The pins are not 5V tolerant, applying more than 3.3V on any pin will destroy the chip.

The GPIO pins 34 to 39 are GPIs – input only pins. These pins do not have internal pull-ups or pull-down resistors. They cannot be used as outputs, so use these pins only as inputs: GPIO 34, GPIO 35, GPIO 36, GPIO 39 There is an integrated SPI flash on the ESP-WROOM-32 chip. The pins GPIO6 to GPIO 11 are exposed in certain ESP32 development boards.

These pins are connected to the integrated SPI flash on the chip and are not recommended for other uses.

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

Capacitive Touch sensor pins

The ESP32 has 10 internal capacitive touch sensors. The capacitive touch pins can also be used to wake up the ESP32 from deep sleep. These internal touch sensors are connected to these 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).

Analog to Digital converter pins

The ESP32 has 18×12 bits ADC (Analog to Digital converter) input channels (while the ESP8266 only has 1x 10 bits ADC). These are the GPIOs that can be used as ADC and respective channels:

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)

Digital to Analog converter pins

There are 2 × 8 bits DAC (Digital to Analog converter) channels on the ESP32 to convert digital signals into analog voltage signal outputs. These are the DAC channels:

DAC1 (GPIO25), DAC2 (GPIO26).

Real Time Clock GPIO pins

There is RTC (Real time clock) GPIO support on the ESP32. The GPIOs routed to the RTC low-power subsystem can be used when the ESP32 is in deep sleep. These RTC GPIOs can be used to wake up the ESP32 from deep sleep when the Ultra Low Power (ULP) co-processor is running. The following GPIOs can be used as an external wake up source:
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).

Real Time Clock GPIO pins

The ESP32 LED PWM (Pulse width modulation) controller has 16 independent channels that can be configured to generate PWM signals with different properties. All pins that can act as outputs can be used as PWM pins (GPIOs 34 to 39 cannot generate PWM). To set a PWM signal, you need to define these parameters in the code: Signal's frequency, Duty cycle, PWM channel, GPIO where you want to output the signal.

The I2C interface pins

The ESP32 has two I2C channels and any pin can be set as SDA or SCL. When using the ESP32 with the Arduino IDE, the default I2C pins are:
GPIO 21 (SDA), GPIO 22 (SCL).

SPI interface pins

By default, the pin mapping for SPI pins is:

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

Strapping Pins

Following pins are used to put the ESP32 into bootloader or flashing mode: **GPIO 0, GPIO 2, GPIO 4, GPIO 5 (must be HIGH during boot), GPIO 12 (must be LOW during boot), GPIO 15 (must be HIGH during boot).**

Most development boards put the pins in the right state for flashing or boot mode. If some peripherals are connected to the strapping pins and the IDE is unable to upload the code or flash the ESP32, it may be because those peripherals are preventing the ESP32 to enter the right mode. After resetting, flashing, or booting, those pins work as expected. Further and more extensive explanations are not in the scope of this eBook so please, refer to the datasheet.

Pins HIGH at Boot

Some GPIOs change their state to HIGH or output PWM signals at boot or reset. This means that if outputs are connected to these GPIOs this may get unexpected results when the ESP32 resets or boots. GPIO 1, GPIO 3, GPIO 5, GPIO 6 to GPIO 11 (connected to the ESP32 integrated SPI flash memory - not recommended for use), GPIO 14, GPIO 15.

Enable (EN)

Enable (EN) is the 3.3V regulator's enable pin. It has a pulled up state, and it needs to be connected to ground to disable the 3.3V regulator. This means that this pin can be connected to a push button to restart your ESP32, for example.

USB to Serial communication

The ESP32 Dev Kit C V2 has a microUSB connection port. It is made around the CP2102 chip made by Silicon Laboratories (alternatively CH340 chip made by WCH) which allows USB to UART serial communication. The chip has the virtual COM port (VCP) feature that appears as a COM port in PC applications. The USB UART interface implements all RS-232 signals, including control and handshaking signals, so existing system firmware does not need to be modified. To be able to use the ESP32 the driver has to be installed.

Wi-Fi Communication

ESP32 Dev Kit C V2 has integrated Wi-Fi communication interface and can operate in three different modes: Wi-Fi station, Wi-Fi access point, and both at the same time. It supports the following features:

- 802.11b and 802.11g data-rates
- 802.11n MCS0-7 in both 20MHz and 40MHz bandwidth
- 802.11n MCS32
- 802.11n 0.4µS guard-interval
- Data-rate up to 150 Mbps
- Receiving STBC 2×1
- Up to 20 dBm transmitting power
- Adjustable transmitting power
- Antenna diversity and selection (software-managed hardware)

Bluetooth Communication

The ESP32 Dev Kit C V2 has an integrated Bluetooth Radio and supports the following features:

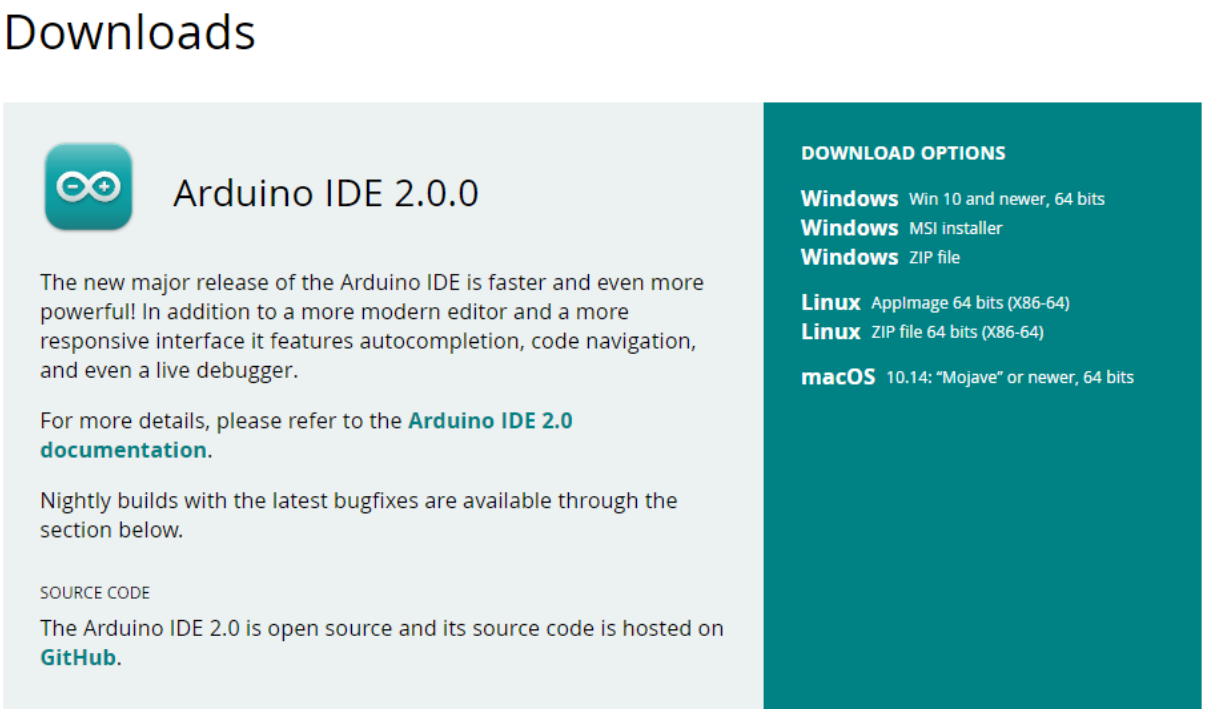
- Class-1, class-2 and class-3 transmit output powers and over 30 dB dynamic control range
- π/4 DQPSK and 8 DPSK modulation
- High performance in NZIF receiver sensitivity with over 98 dB dynamic range
- Class-1 operation without external PA
- Internal SRAM allows full speed data transfer, mixed voice and data, and full piconet operation
- Logic for forward error correction, header error control, access code correlation, CRC, demodulation, encryption bit stream generation, whitening and transmit pulse shaping
- ACL, SCO, eSCO and AFH
- A-law, μ-law and CVSD digital audio CODEC in PCM interface
- SBC audio CODEC
- Power management for low power applications
- SMP with 128-bit AES

Also, the Bluetooth Radio has support for the following communication interface protocols:

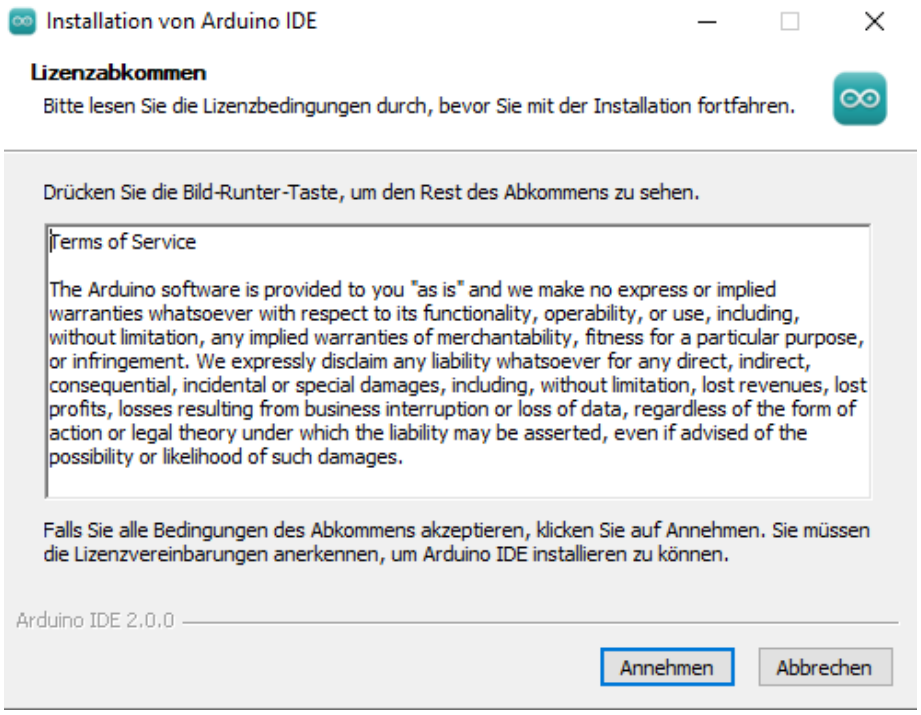
- UART HCI interface, up to 4 Mbps
- SDIO / SPI HCI interface
- I2C interface
- PCM / I2S audio interface

Installation of the Arduino IDE

You can download the free Arduino IDE development environment from the following link: https://www.arduino.cc/en/Main/Software

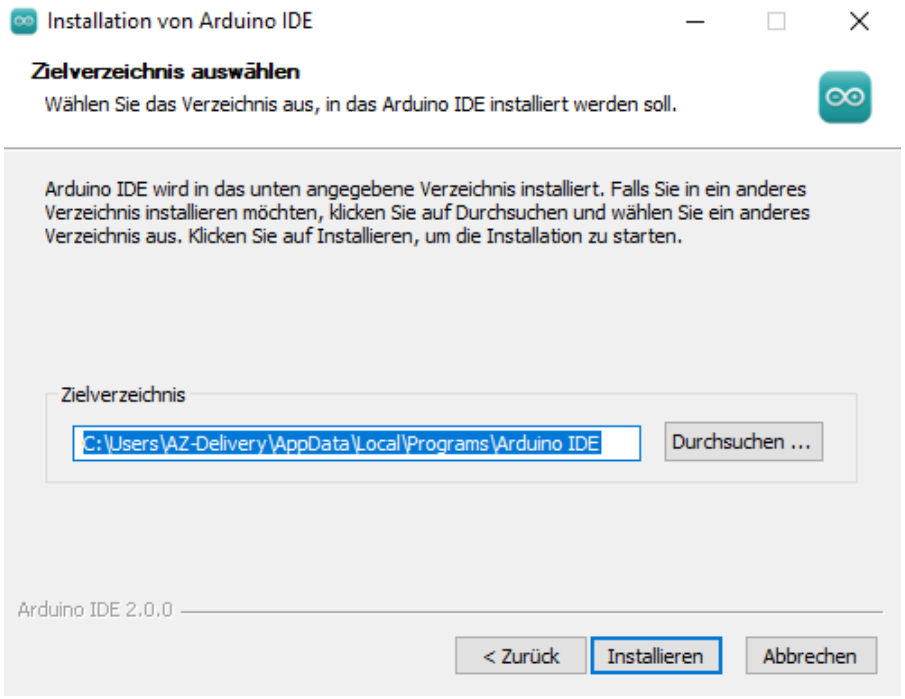


After starting the Arduino IDE installation file "`arduino-ide_2.0.0_Windows_64bit.exe`" the licence conditions of the software must be read and accepted.

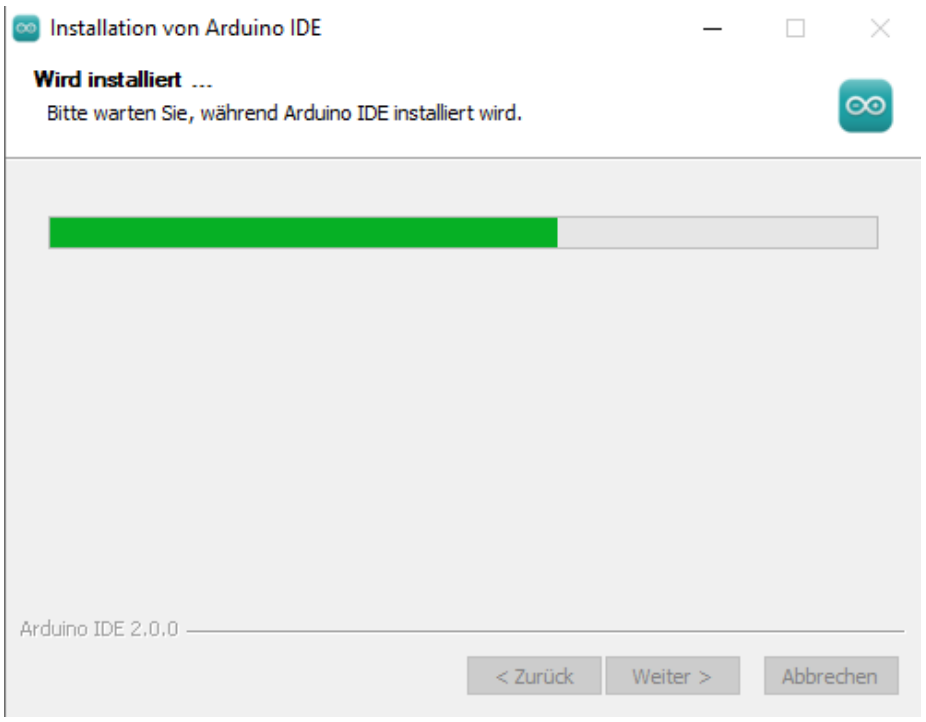


In the next step, different options can be selected for installation.

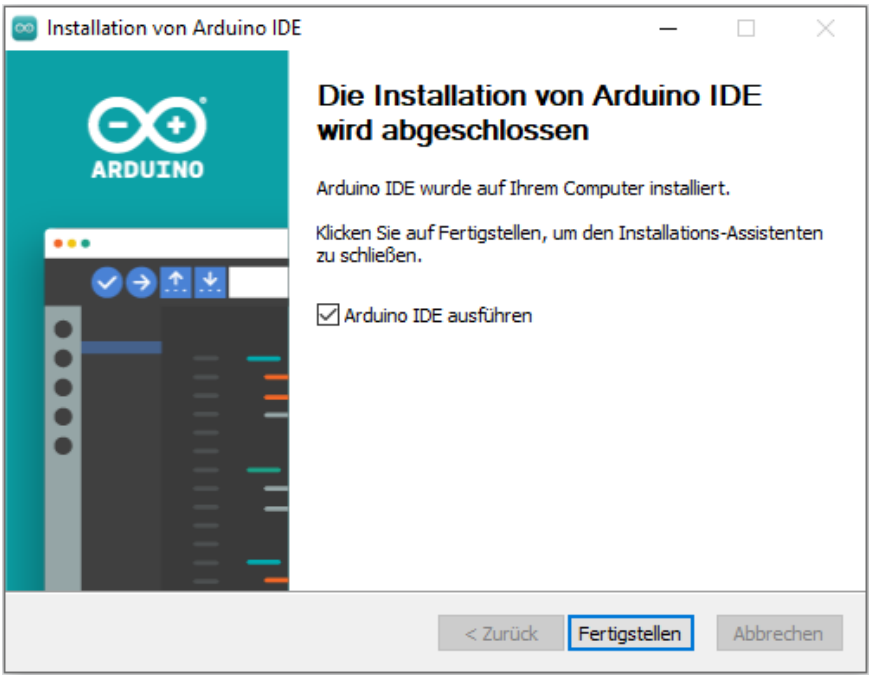
Finally, the destination folder must be specified. The installation requires approx. 500MB of free disk space.



Click on "Install" to start the installation.



After successful installation, the installation programme can be terminated via the "**Finish**" button.

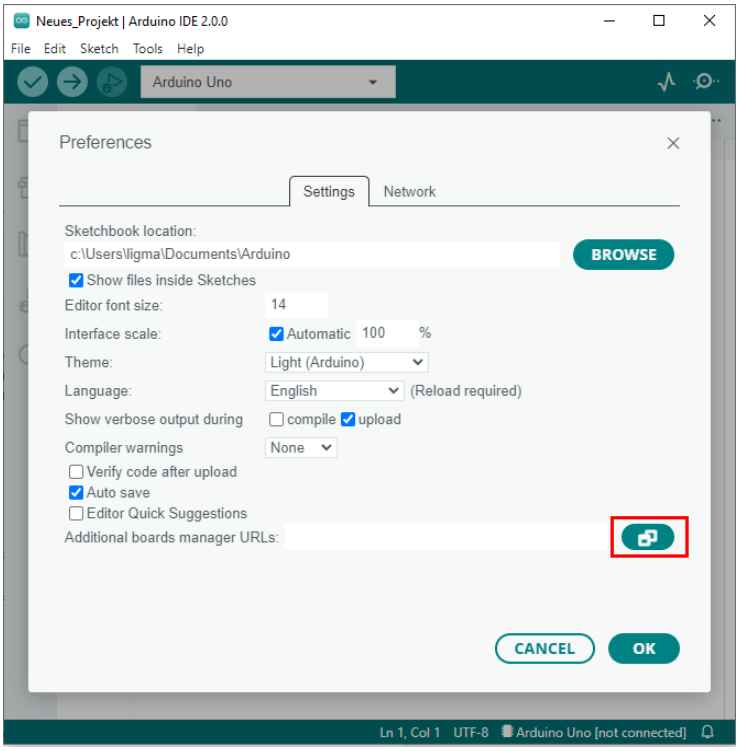


Additional setup

In order to use ESP32 Dev Kit C V2 with Arduino IDE, follow a few easy steps. Before setting the Arduino IDE, the driver for the USB to Serial communication has to be installed. If the driver is not installed automatically, you can download and install the driver here:
For CP2102:
`https://www.silabs.com/interface/usb-bridges/classic/device.cp2102`

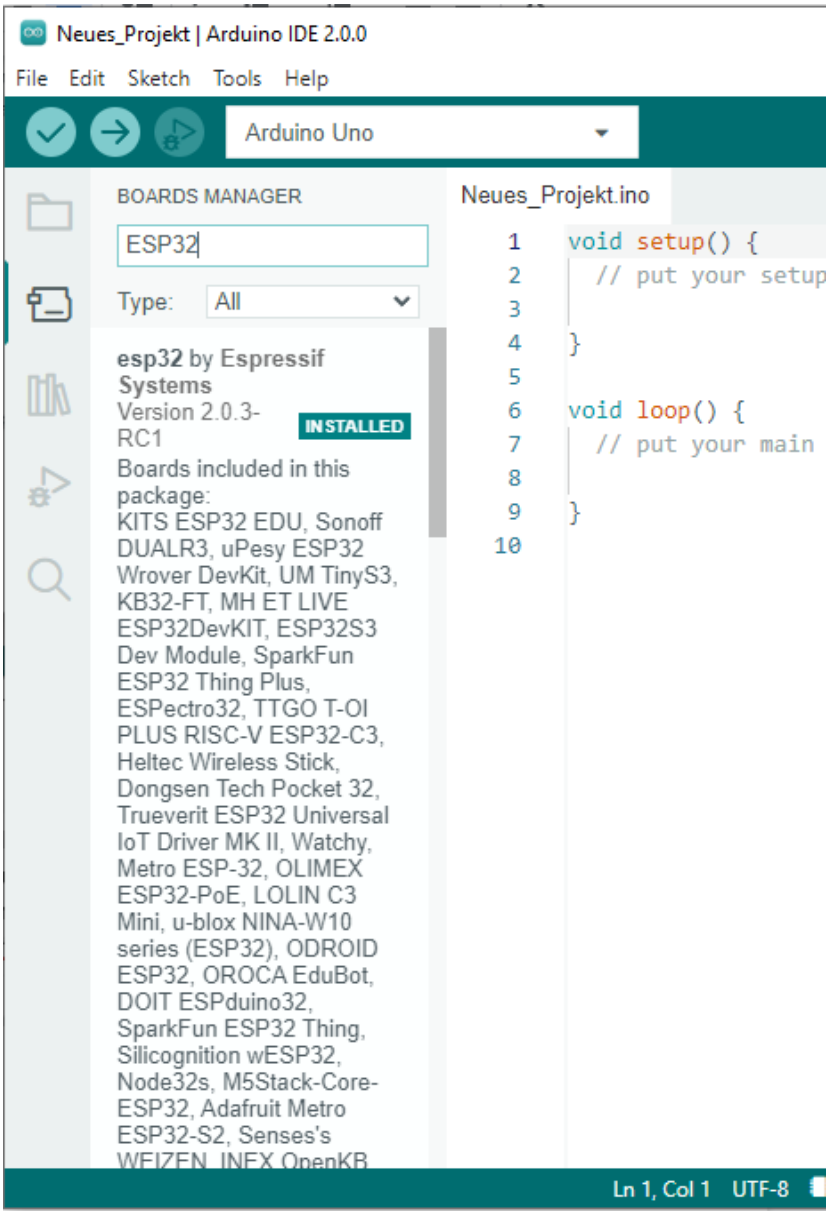
For CH340:
`http://www.wch.cn/download/CH341SER_ZIP.html`

Next, to install support for the ESP32 platform, open Arduino IDE and go to:
File > Preferences, and find Additional URLs field.
Then copy the following URL:
https://dl.espressif.com/dl/package_esp32_index.json

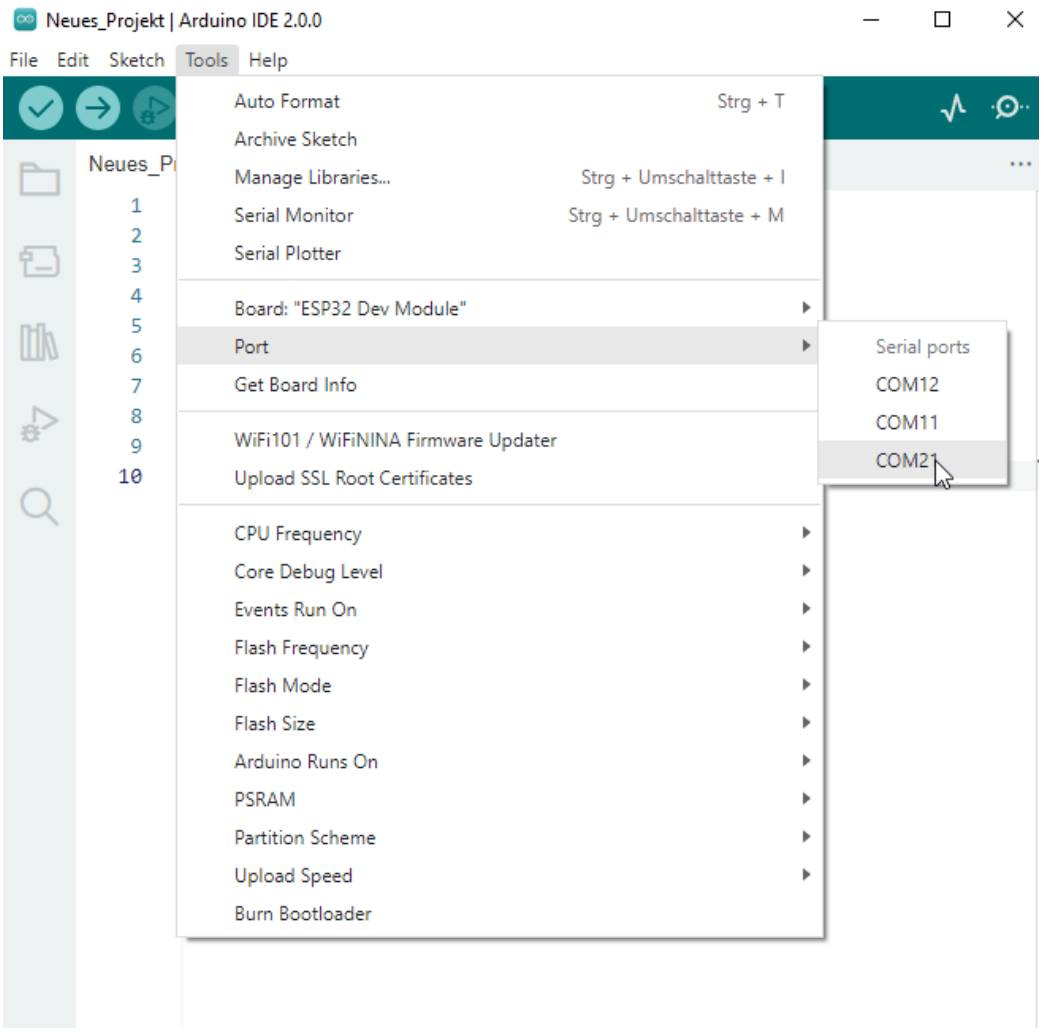


Paste this link in the Additional URLs field. If one or more links are inside this field, just add one comma after the last link, paste a new link after comma and click the OK button.

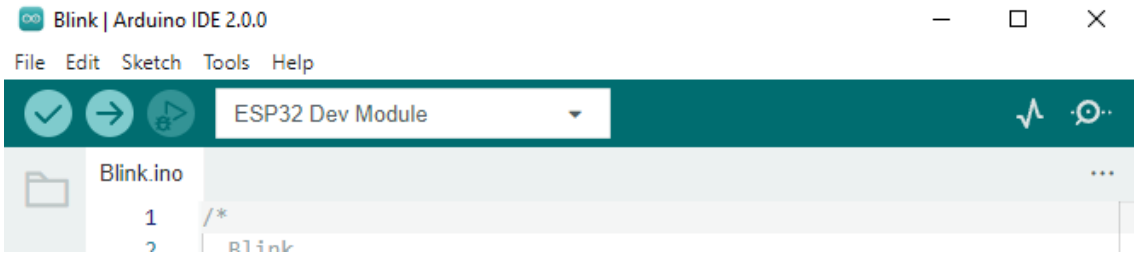
Open Arduino IDE again and go to: Tools > Board > Boards Manager
When new window opens, type esp32 in the search box and install the board called esp32 made by Espressif Systems, as shown on the following image:



To select ESP32 board, go to: Tools > Board > ESP32 Arduino > ESP32 Dev Module

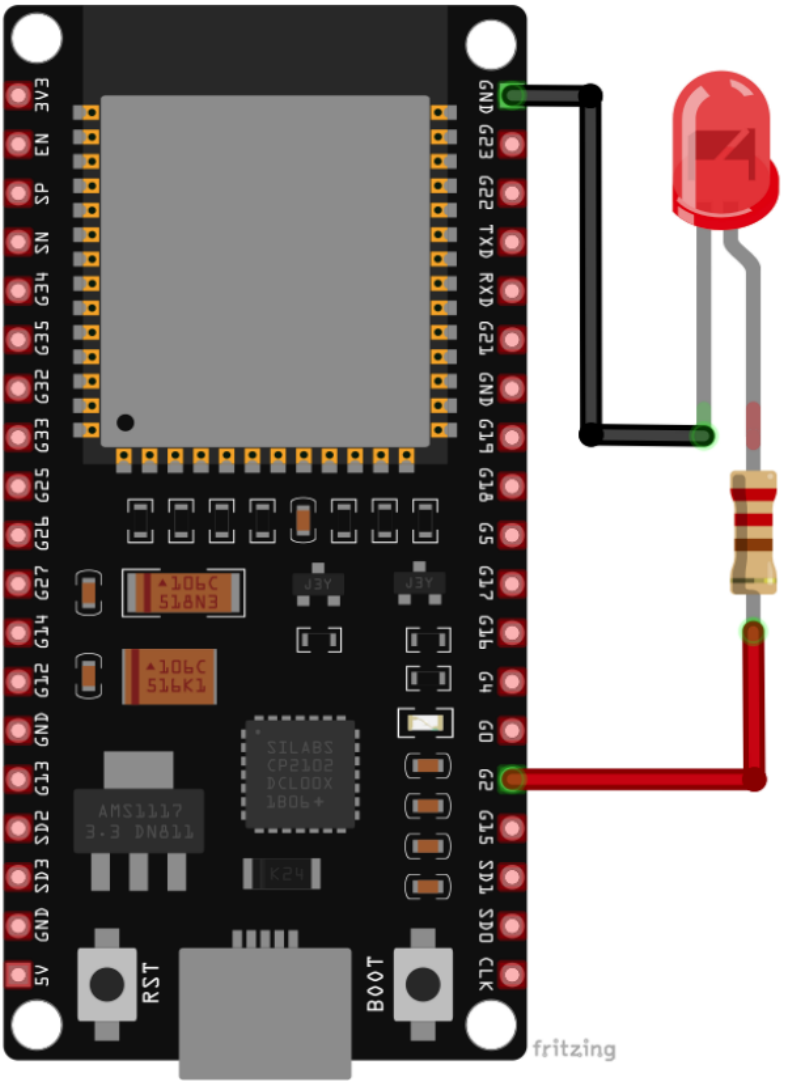


To upload the sketch code to the ESP32 board, first select port on which you connected the board. Go to: Tools > Port > {port name}0



ESP32 Dev Kit C V2 wiring example

Connect the ESP32 Dev Kit C V2 with an LED and resistor as shown on the following connection diagram:



ESP32 Dev Kit C V2 pin	LED pin	Wire color
GPIO2 (pin2)	Anode (+) through resistor	**Red wire**
GND	Cathode (-)	**Black wire**

Sketch examples

Blinking LED

```
int ledPin = 2;

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

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

PWM – Pulse Width Modulation

```
#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);
}
```

Now it is the time to learn and make your own projects. You can do that with the help of many example scripts and other tutorials, which can be found on the Internet.

If you are looking for the high quality products for Arduino and Raspberry Pi, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

https://az-delivery.de

Have Fun!

Impressum

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