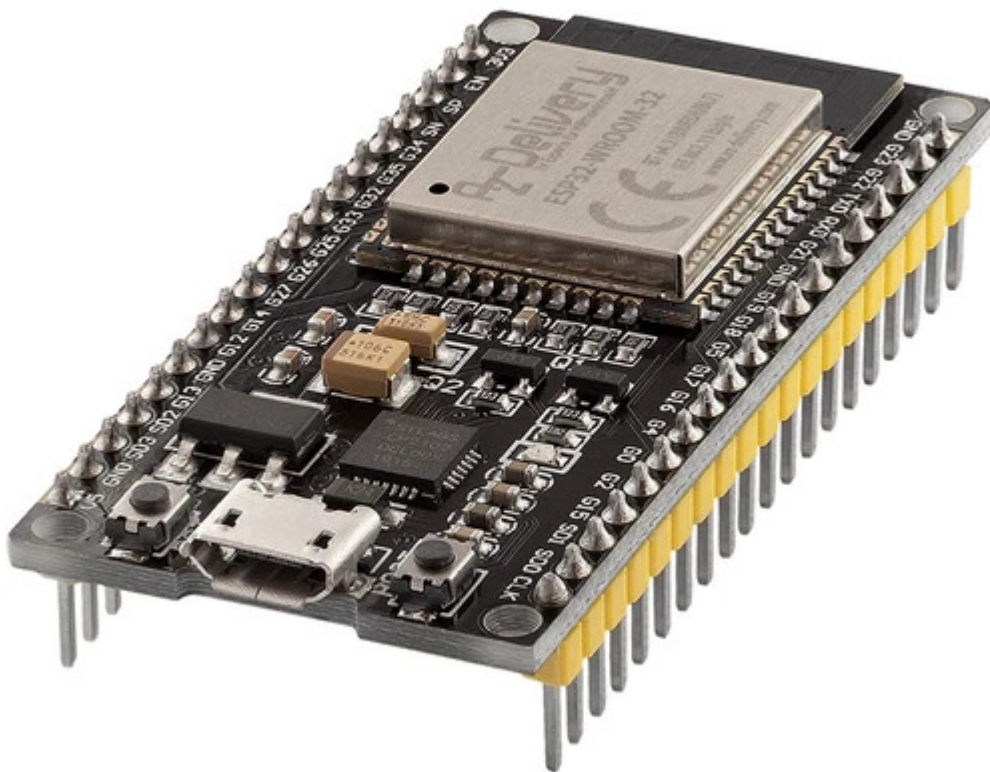


AZ-Delivery

Benvenuto!

Grazie per aver acquistato il nostro *ESP-32 Dev Kit C V2 AZ-Delivery*. Nelle pagine seguenti, ti illustreremo come utilizzare e configurare questo pratico dispositivo.

Buon divertimento!



Indice dei Contenuti

Introduzione.....	3
Specifiche.....	4
ESP32 Dev Kit C V2.....	5
Piedinatura.....	6
Descrizione pin.....	7
Pin Sensore Touch Capacitivo.....	8
Pin convertitore da Analogico a Digitale.....	9
Pin del Convertitore Digitale-Analogico.....	9
Pin Real Time Clock GPIO.....	10
Pin PWM (Pulse Width Modulation).....	11
Pin Interfaccia I2C.....	11
Pin interfaccia SPI.....	12
Pin Strapping.....	12
Pin HIGH all'Avvio.....	13
Abilita (EN).....	13
Comunicazione da USB a Seriale.....	14
Comunicazione WiFi.....	15
Comunicazione Bluetooth.....	16
Altre caratteristiche.....	18
Come configurare l'Arduino IDE.....	19
Configurazione aggiuntiva.....	23
Esempio di cablaggio ESP32 Dev Kit C V2.....	27
Esempi di sketch.....	28



Introduzione

L'ESP32 Dev Kit C V2 è una scheda di sviluppo creata intorno al chip ESP32 WROOM 32, contenente un regolatore di tensione e un circuito programmatore USB per il chip ESP32, e alcune altre caratteristiche.

Per lo sviluppo di applicazioni si può scegliere tra Arduino IDE o ESP-IDF (piattaforma nativa). La maggior parte degli utenti sceglie l'Arduino IDE per la sua semplicità e compatibilità. La comunità di utenti scheda microcontrollore è molto attiva e supporta piattaforme come ESP32.

ESP32 Dev Kit C V2 viene fornito con un firmware preinstallato che permette di lavorare con il linguaggio interpretato, inviando comandi attraverso la porta seriale (chip CP2102). Le schede ESP32 sono una delle piattaforme più utilizzate per i progetti Internet of Things (IoT).

La scheda ESP32 Dev Kit C V2 è appositamente progettata per lavorare su breadboard. Ha un regolatore di tensione che gli permette di alimentarsi direttamente dalla porta USB. I pin di input/output lavorano a 3.3V. Il chip CP2102 è responsabile della comunicazione da USB a seriale.

Specifiche

Tensione di alimentazione (USB)	5V DC
Tensione Input/Output	3.3V DC
Tensione di esercizio richiesta	min. 500mA
SoC	ESP32-WROOM 32
CPU	Xtensa® single-dual-core 32-bit LX6
Range frequenza di clock	80MHz / 240MHz
RAM	512kB
Memoria flash esterna	4MB
Pin I/O	34
Canali ADC	18
Risoluzione ADC	12-bit
Canali DAC	2
Risoluzione DAC	8-bit
Interfacce di comunicazione	SPI, I2C, I2S, CAN, UART
Protocolli Wi-Fi	802.11 b/g/n (802.11n fino a 150 Mbps)
Frequenza Wi-Fi	2.4 GHz - 2.5 GHz
Bluetooth	V4.2 - BLE e Bluetooth Classico
Antenna wireless	PCB
Dimensioni	56x28x13mm(2.2x1.1x0.5in)

ESP32 Dev Kit C V2

La serie di chip Wi-Fi ESP32 WROOM 32 è prodotta da Espressif Systems. ESP32 WROOM-32 è un modulo Wi-Fi economico adatto a progetti Fai da Te nel campo dell'Internet of Things (IoT). Questo modulo viene fornito con molti GPIO e supporto per una varietà di protocolli come SPI, I2C, I2S, UART, e più. La parte migliore è che viene fornito con rete wireless inclusa, il che lo rende diverso da altri micro controller come scheda microcontrollore. Questo significa che può facilmente controllare e monitorare i dispositivi a distanza via Wi-Fi e Bluetooth® ad un prezzo accessibile.

ESP32 WROOM-32 è un system-on-chip (SoC) che integra un microcontrollore Tensilica a 32 bit, interfacce periferiche digitali standard, interruttori d'antenna, balun RF, amplificatore di potenza, amplificatore di ricezione a basso rumore, filtri e moduli di gestione della potenza in un piccolo pacchetto. Fornisce un Wi-Fi 2.4GHz (802.11 b/g/n, che supporta velocità fino a 150MB/s), BLE e comunicazione wireless classica Bluetooth®, 34 pin I/O, interfacce I2C e I2S, ADC (conversione analogico-digitale), DAC (conversione digitale-analogica), interfaccia SPI, UART su pin dedicati, e PWM (Pulse Width Modulation).

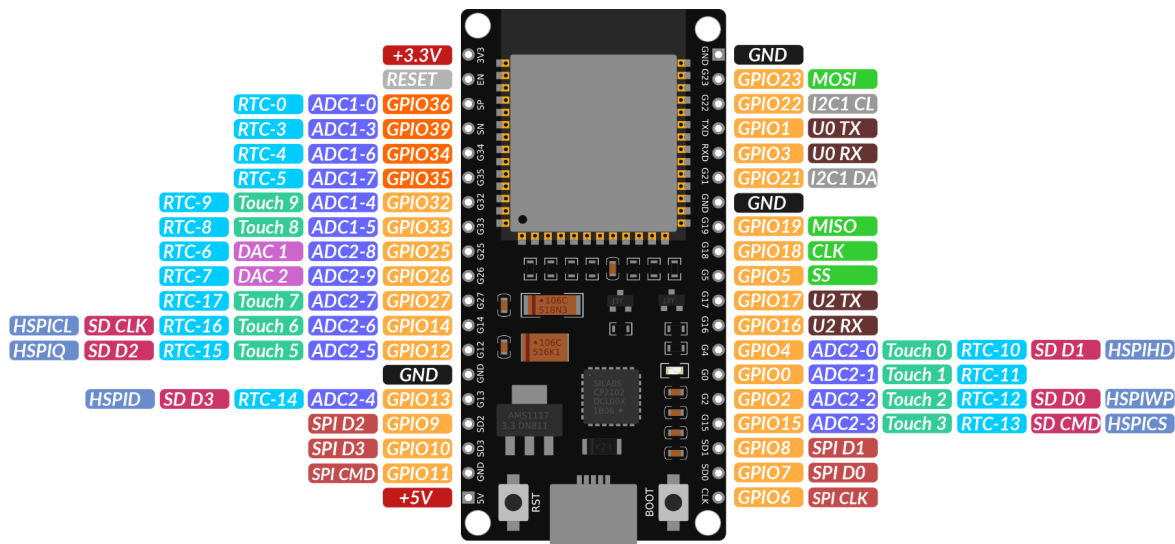
Il nucleo del processore, chiamato LX6 da Espressif, è basato sul controller del processore Xtensa® dual-core 32-bit LX6 e funziona a una gamma di frequenza tra 80-240MHz. Ha una ROM di avvio di 448kB, 520kB di SRAM

Az-Delivery

on-chip, e 4MB di memoria flash esterna a cui si può accedere tramite interfaccia SPI.

Piedinatura

L'ESP32 Dev Kit C V2 ha 38 pin. La piedinatura è mostrata nell'immagine seguente:



- Digital In/Out ports (all support PWM)
- Digital Input ports
- Analog Input 12 bits, 0 to 3.3V
- Analog Output 8 bits, 0 - 3.3V
- Capacitive Touch Sensor ports
- I/O -pins from RTC ultra low power processor, usable in deep sleep mode
- SD card interface
- 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.

- I2C bus (Wire)
- VSPI bus
- Serial interfaces
- HSPI bus

Az-Delivery

Per la descrizione dettagliata della piedinatura e le capacità di I/O si prega di fare riferimento al datasheet che si può trovare al seguente [link](#).

NOTA: La corrente massima assoluta assorbita per un GPIO è 40mA secondo la sezione "Condizioni operative raccomandate" nel datasheet dell'ESP32.

Descrizione pin

Proprio come una normale scheda microcontrollore, l'ESP32 Dev Kit C V2 ha dei pin di ingresso/uscita digitali (pin GPIO - Pin General Purpose Input/Output). Questi input/output digitali funzionano a 3.3V.

La tensione di 5V non deve essere collegata a nessun pin del chip ESP32!

I pin non tollerano 5V, l'applicazione di più di 3,3V su qualsiasi pin distruggerà il chip.

I pin GPIO da 34 a 39 sono GPI - pin solo input. Questi pin non hanno resistenze interne di pull-up o pull-down. Non possono essere usati come uscite, quindi usate questi pin solo come ingressi: GPIO 34, GPIO 35, GPIO 36, GPIO 39

C'è una memoria flash SPI integrata sul chip ESP-WROOM-32. I pin da GPIO6 a GPIO 11 sono esposti in alcune schede di sviluppo ESP32. Questi pin sono collegati alla flash SPI integrata sul chip e non sono raccomandati per altri usi.

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

Pin Sensore Touch Capacitivo

L'ESP32 ha 10 sensori touch capacitivi interni. I pin touch capacitivi possono anche essere usati per svegliare l'ESP32 dal deep sleep. Questi sensori touch interni sono collegati a questi GPIO: 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).

Pin convertitore da Analogico a Digitale

The ESP32 ha dei canali di ingresso 18x12 bit ADC (Analog to Digital converter) (mentre l'ESP8266 solo ha 1x 10 bits ADC). Questi sono i GPIO che possono essere utilizzati come ADC e i rispettivi canali:

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).

Pin del Convertitore Digitale-Analogico

Ci sono 2 canali DAC (Digital to Analog converter) a 8 bit sull'ESP32 per convertire i segnali digitali in uscite di tensione analogica. Questi sono i canali DAC:

DAC1 (GPIO25), DAC2 (GPIO26).

Pin Real Time Clock GPIO

Sull'ESP32 c'è il supporto RTC (Real time clock) GPIO. I GPIO indirizzati al sottosistema RTC a basso consumo possono essere utilizzati quando l'ESP32 è in deep sleep. Questi GPIO RTC possono essere usati per svegliare l'ESP32 dal deep sleep quando il co-processore Ultra Low Power (ULP) è in funzione. I seguenti GPIO possono essere utilizzati come fonte di risveglio esterno: 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).

Pin PWM (Pulse Width Modulation)

Il controller ESP32 LED PWM (Pulse width modulation) ha 16 canali indipendenti che possono essere configurati per generare segnali PWM con diverse proprietà. Tutti i pin che possono agire come uscite possono essere utilizzati come pin PWM (i GPIO da 34 a 39 non possono generare PWM). Per configurare un segnale PWM, è necessario definire questi parametri nel codice: Frequenza del segnale, Ciclo di lavoro, canale PWM, GPIO dove si vuole emettere il segnale.

Pin Interfaccia I2C

L'ESP32 ha due canali I2C e qualsiasi pin può essere impostato come SDA o SCL. Quando si usa l'ESP32 con l'Arduino IDE, i pin I2C di default sono: GPIO 21 (SDA), GPIO 22 (SCL).

Pin interfaccia SPI

Di default, la mappatura dei pin per i pin SPI è:

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

Pin Strapping

I seguenti pin sono utilizzati per mettere l'ESP32 in modalità bootloader o flashing:

GPIO 0, GPIO 2, GPIO 4, GPIO 5 (deve essere HIGH durante l'avvio), GPIO 12 (deve essere LOW durante l'avvio), GPIO 15 (deve essere HIGH durante l'avvio).

La maggior parte delle schede di sviluppo mette i pin nello stato giusto per il flashing o la modalità di avvio. Se alcune periferiche sono collegate ai pin di strapping e l'IDE non sono in grado di caricare il codice o fare il flash dell'ESP32, può essere perché queste periferiche stanno impedendo all'ESP32 di entrare nella modalità giusta. Dopo il reset, il flashing o l'avvio, questi pin funzionano come previsto. C'è una guida di documentazione sulla selezione della modalità di avvio al seguente link. Ulteriori e più ampie spiegazioni non rientrano nello scopo di questo eBook quindi, per favore, fate riferimento al datasheet.

Pin HIGH all'Avvio

Alcuni GPIO cambiano il loro stato a HIGH o emettono segnali PWM all'avvio o al reset. Questo significa che se le uscite sono collegate a questi GPIO si possono ottenere risultati inaspettati quando l'ESP32 si resetta o si avvia.

GPIO 1, GPIO 3, GPIO 5, GPIO 6 to GPIO 11 (collegato alla memoria flash SPI integrata dell'ESP32 - non si raccomanda l'uso), GPIO 14, GPIO 15.

Abilita (EN)

Enable (EN) è il pin di abilitazione del regolatore a 3,3V. Ha uno stato pulled up e deve essere collegato a terra per disabilitare il regolatore 3.3V. Ciò significa che questo pin può essere collegato a un pulsante per riavviare il vostro ESP32, per esempio.



Comunicazione da USB a Seriale

L'ESP32 Dev Kit C V2 ha una porta di connessione microUSB. È costruito intorno al chip CP21202 prodotto da Silicon Laboratories che permette la comunicazione seriale da USB a UART.. Il chip ha la funzione di porta COM virtuale (VCP) che appare come porta COM nelle applicazioni PC. L'interfaccia UART CP2102 implementa tutti i segnali RS-232, compresi i segnali di controllo e di handshake, quindi non è necessario modificare il firmware del sistema esistente. Per poter utilizzare l'ESP32 è necessario installare il driver.

Comunicazione WiFi

ESP32 Dev Kit C V2 ha un'interfaccia di comunicazione Wi-Fi integrata e può funzionare in tre diverse modalità: Stazione Wi-Fi, punto di accesso Wi-Fi ed entrambi allo stesso tempo. Supporta le seguenti funzioni:

- 802.11b e data rate 802.11g
- 802.11n MCS0-7 con larghezza di banda sia da 20MHz che da 40MHz
- 802.11n MCS32
- 802.11n intervallo di guardia 0.4 μ S
- Data-rate fino a 150 Mbps
- Ricevente STBC 2x1
- Fino a 20 dBm di potenza di trasmissione
- Potenza di trasmissione regolabile
- Diversità antenna e selezione (hardware gestito da software)

Comunicazione Bluetooth

L'ESP32 Dev Kit C V2 ha una radio Bluetooth integrata e supporta le seguenti caratteristiche:

- Classe-1, classe-2 e classe-3 trasmettono potenze di uscita e oltre 30 dB di gamma di controllo dinamico
- $\pi/4$ DQPSK e modulazione 8 DPSK
- Alte prestazioni nella sensibilità del ricevitore NZIF con oltre 98 dB di gamma dinamica
- Funzionamento in classe 1 senza PA esterno
- La SRAM interna permette il trasferimento di dati a piena velocità, voce e dati misti e il funzionamento completo piconet
- Logica per la correzione degli errori in avanti, controllo degli errori dei termini, correlazione dei codici di accesso, CRC, demodulazione, generazione di flussi di bit di crittografia, sbiancamento e modellamento degli impulsi di trasmissione
- ACL, SCO, eSCO e AFH
- A-law, μ -law e CODEC audio digitale CVSD nell'interfaccia PCM
- CODEC audio SBC
- Gestione della potenza per applicazioni a basso consumo
- SMP con 128-bit AES

Az-Delivery

Inoltre, la Radio Bluetooth ha il supporto per i seguenti protocolli di interfaccia di comunicazione:

- Interfaccia UART HCI, fino a 4 Mbps
- Interfaccia SDIO / SPI HCI
- Interfaccia I2C
- Interfaccia audio PCM / I2S.

Altre caratteristiche

Il chip ESP32-WROOM 32D ha un sensore a effetto Hall integrato che rileva i cambiamenti del campo magnetico nei suoi dintorni.

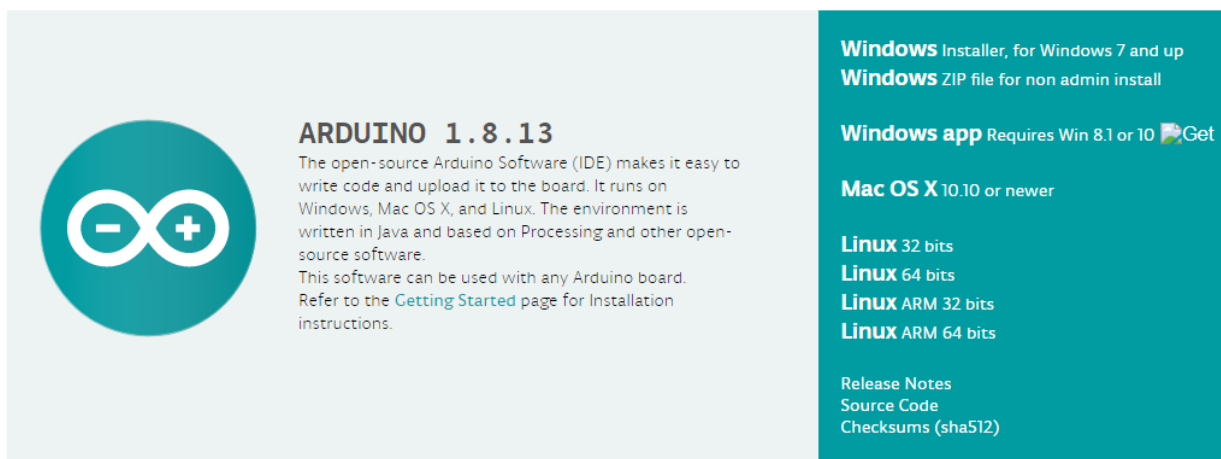
Il sensore di Hall è basato su una resistenza a N portanti. Quando il chip è nel campo magnetico, il sensore Hall sviluppa una piccola tensione sul resistore, che può essere misurata direttamente dal convertitore analogico-digitale (ADC), o amplificata dal preamplificatore analogico a bassissimo rumore e poi misurata dall'ADC.

Il sensore di temperatura genera una tensione che varia con la temperatura. La tensione è convertita internamente tramite un convertitore analogico-digitale in un codice digitale. Il sensore di temperatura ha un range da -40°C a 125°C. Poiché l'offset del sensore di temperatura varia da chip a chip a causa della variazione di processo, insieme al calore generato dal circuito Wi-Fi stesso (che influenza le misurazioni), il sensore di temperatura interno è adatto solo per applicazioni che rilevano variazioni di temperatura invece di temperature assolute e anche per scopi di calibrazione. Tuttavia, se l'utente calibra il sensore di temperatura e usa il dispositivo in un'applicazione minimamente alimentata, i risultati potrebbero essere abbastanza accurati.

Come configurare l'Arduino IDE


Se l'Arduino IDE non è installato, seguire il link e scaricare il file di installazione del sistema operativo scelto. La versione di Arduino IDE usata per questo eBook è la 1.8.13.

Download the Arduino IDE



ARDUINO 1.8.13
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.

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 \(sha512\)](#)

Per gli utenti Windows, fare doppio clic sul file .exe scaricato e seguire le istruzioni nella finestra di installazione..

Az-Delivery

Per gli utenti Linux, scaricare un file con estensione `.tar.xz`, che è necessario estrarre. Quando lo si estrae, andare nella directory estratta, e aprire il terminale in quella directory. È necessario eseguire due `script .sh`, il primo chiamato `arduino-linux-setup.sh` e il secondo chiamato `install.sh`.

Per eseguire lo script, aprire il terminale nella directory in cui è stato salvato lo script ed eseguire il seguente comando:

```
sh arduino-linux-setup.sh user_name
```

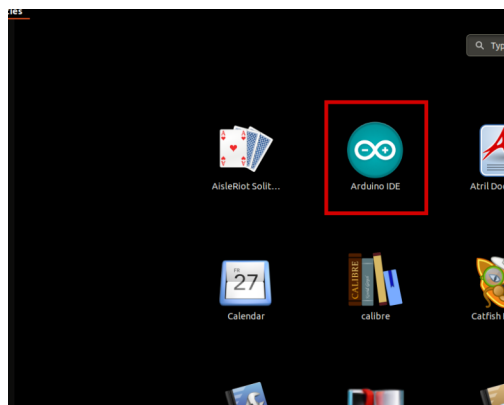
`user_name` - è il nome di un superutente nel sistema operativo Linux..

All'avvio del comando deve essere inserita una password per il superutente.

Aspettate qualche minuto che lo script completi tutto.

Il secondo script, chiamato `install.sh`, deve essere usato dopo l'installazione del primo script. Eseguire il seguente comando nel terminale (directory estratta): **`sh install.sh`**

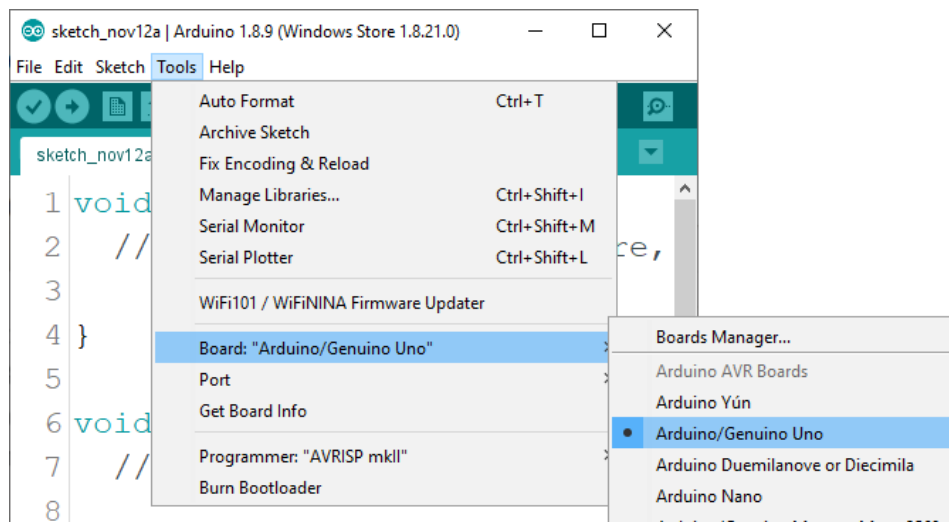
Dopo l'installazione di questi script, andare su *Tutte le App*, dove troverai l'*Arduino IDE* installato.



Az-Delivery

Quasi tutti i sistemi operativi sono dotati di un editor di testo preinstallato (ad esempio *Windows* viene fornito con *Notepad*, *Linux Ubuntu* viene fornito con *Gedit*, *Linux Raspbian* viene fornito con *Leafpad*, ecc.). Tutti questi editor di testo sono perfettamente adatti allo scopo dell'eBook.

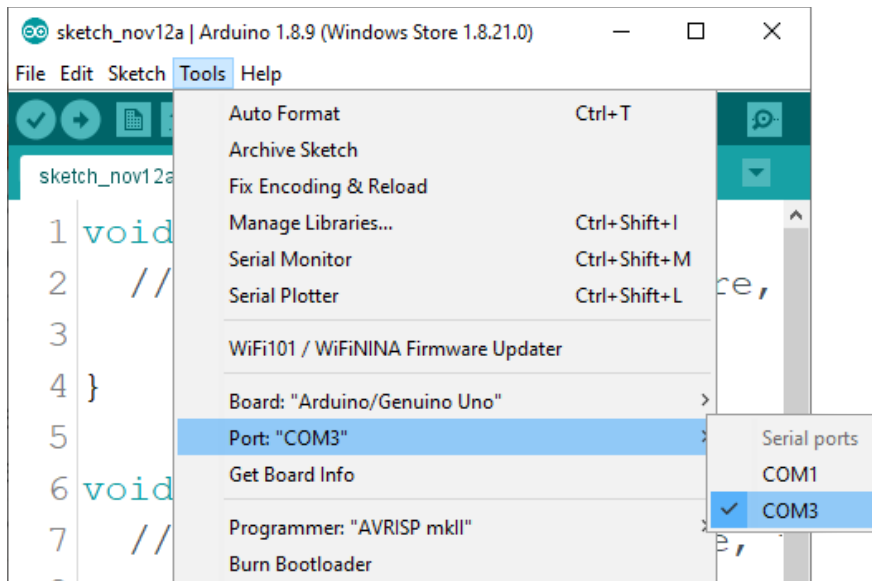
La prossima cosa da fare è controllare se il PC è in grado di rilevare la scheda microcontrollore. Aprite l'Arduino IDE appena installato e andate su: *Strumenti > Scheda > {your board name here}*
{your board name here} dovrebbe essere l'*Arduino/Genuino Uno*, come si può vedere nella seguente immagine:



Deve essere selezionata la porta alla quale è collegata la scheda microcontrollore. Vai su: *Strumenti > Porta > {port name goes here}* e quando la scheda microcontrollore è collegata alla porta USB, il nome della porta è visibile nel menu a tendina dell'immagine precedente.

Az-Delivery

Se si utilizza l'Arduino IDE su Windows, i nomi delle porte sono i seguenti:



Per gli utenti Linux, il nome della porta è `/dev/ttyUSBx` per esempio, dove `x` rappresenta un numero intero compreso tra `0` e `9`.



Configurazione aggiuntiva

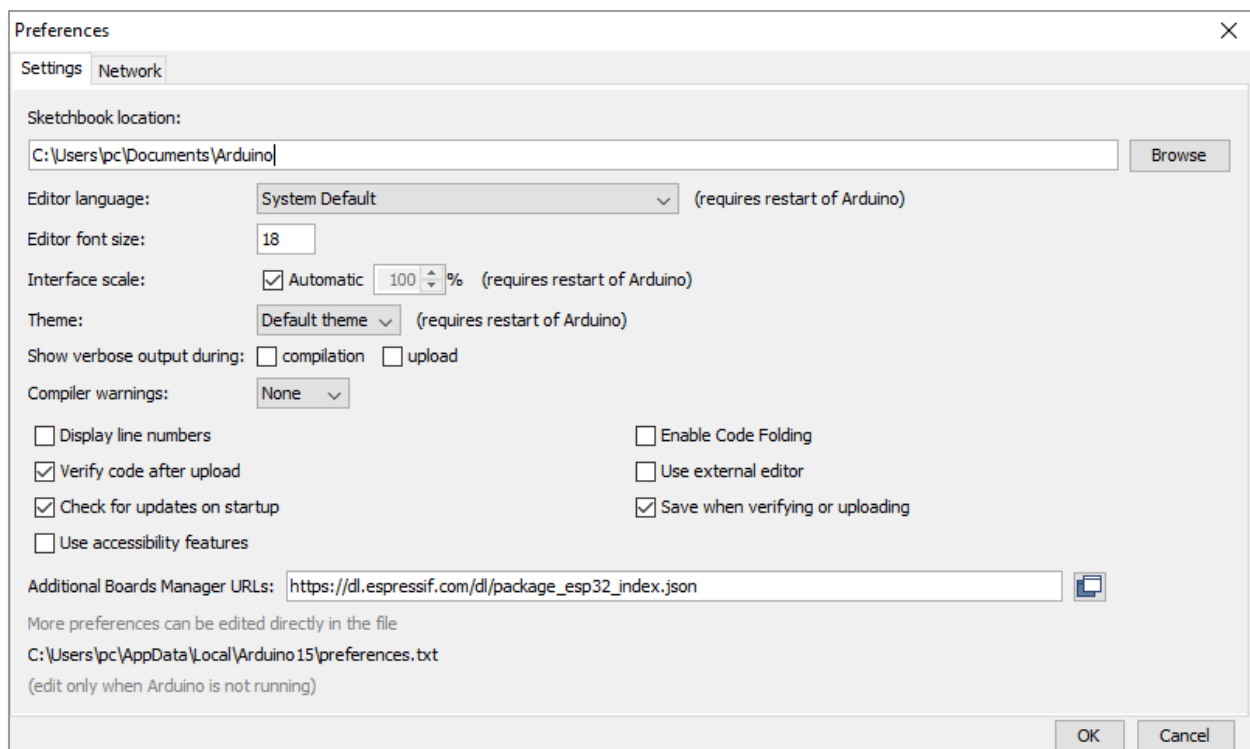
Per usare ESP32 Dev Kit C V2 con Arduino IDE, seguite alcuni semplici passi. Prima di impostare l'Arduino IDE, bisogna installare il driver per la comunicazione USB-seriale. Se il driver non viene installato automaticamente, c'è una pagina di supporto che contiene i driver per Windows/Mac o Linux e si può scegliere a seconda di quale si usa. I driver possono essere scaricati dal seguente [link](#)..

Az-Delivery

Successivamente, per installare il supporto per la piattaforma ESP32, aprire Arduino IDE e andare su: *File > Preferenze*, e trovare il campo URL aggiuntivi.

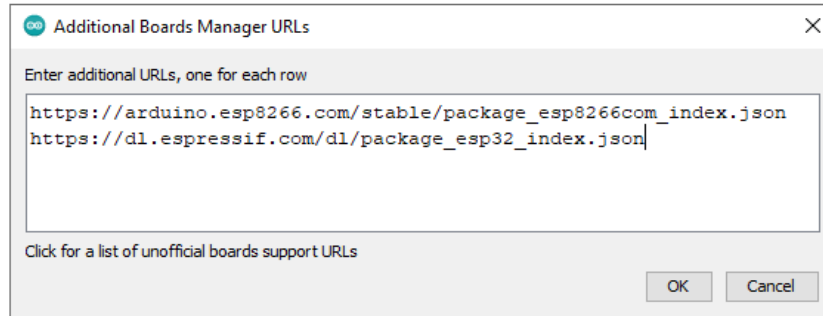
Poi copiare il seguente URL:

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



AZ-Delivery

Incolla questo link nel campo URL aggiuntivi. Se in questo campo ci sono uno o più link, basta aggiungere una virgola dopo l'ultimo link, incollare il nuovo link dopo la virgola e cliccare sul pulsante **OK**.



Aprire nuovamente l'Arduino IDE e andare su:

Strumenti > Scheda > Gestione Schede

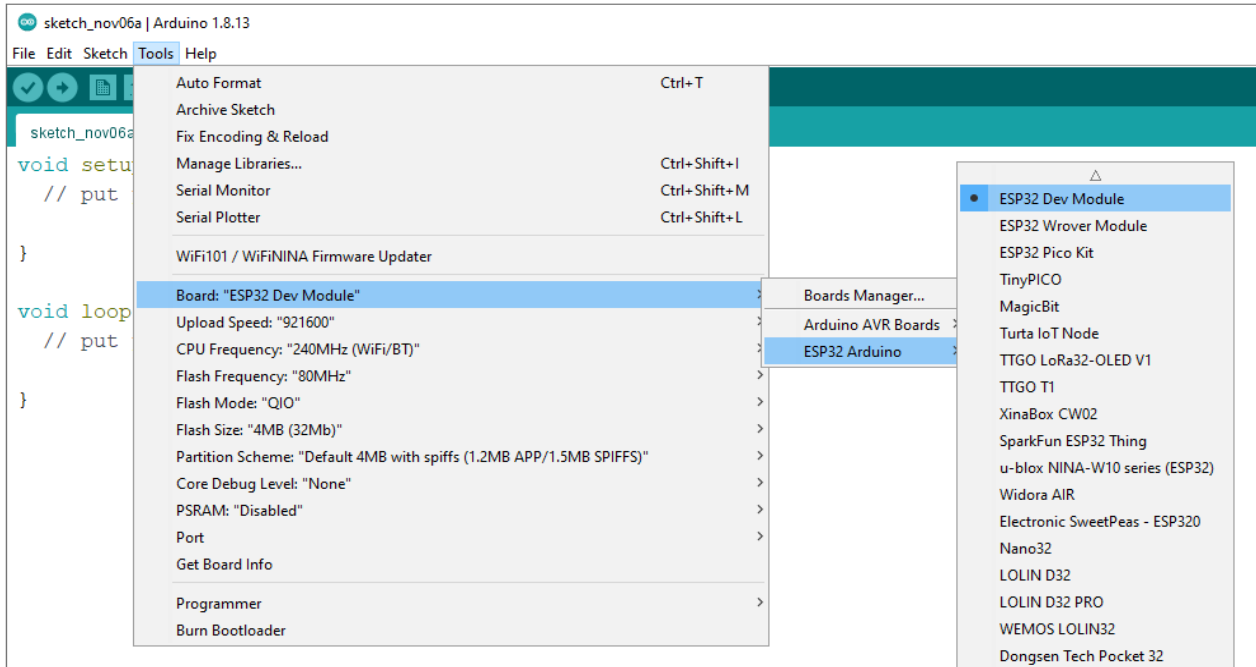
Quando si apre una nuova finestra, digitate *esp32* nella casella di ricerca e installate la scheda chiamata *esp32* prodotta da *Espressif Systems*, come mostrato nella seguente immagine:



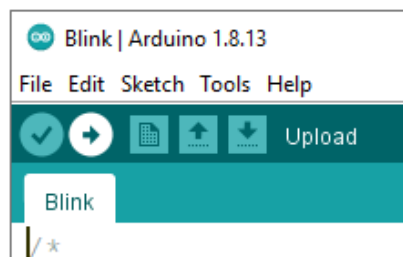
AZ-Delivery

Per selezionare la scheda ESP32, andare su:

Strumenti > Scheda > ESP32 > ESP32 Dev Module



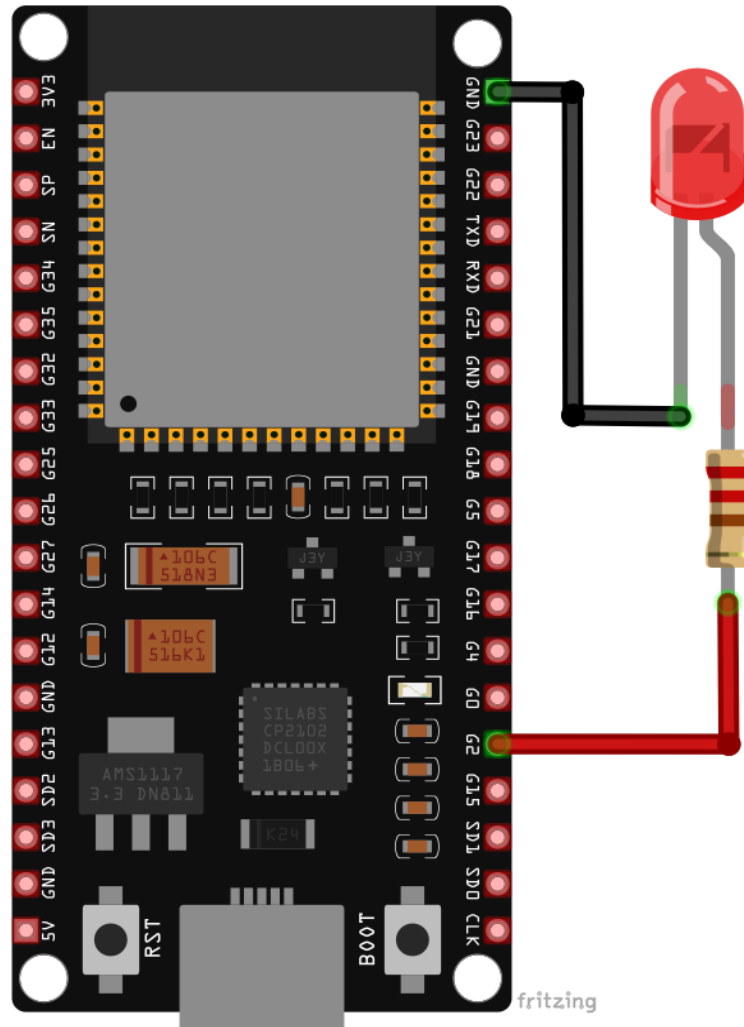
Per caricare il codice dello sketch sulla scheda ESP32, selezionate prima la porta alla quale avete collegato la scheda. Andate a: **Strumenti > Porta > {port name}0**



Se il caricamento non funziona al primo tentativo, può essere utile premere il "Pulsante di Boot" sul modulo durante il caricamento. Si prega di utilizzare un cavo certificato USB 2.0 per la programmazione.

Esempio di cablaggio ESP32 Dev Kit C V2

Collegare l'ESP32 Dev Kit C V2 con un LED e una resistenza come mostrato nel seguente schema di collegamento:



Pin ESP32 Dev Kit C V2	Pin LED	Colore filo
GPIO2 (pin2)	Anodo (+) attraverso la resistenza	Filo rosso
GND	Catodo (-)	Filo nero

Az-Delivery

Esempi di sketch

LED lampeggiante

```
int ledPin = 2;

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

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

AZ-Delivery

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);
}
```

AZ-Delivery

E ora è tempo di imparare e di creare dei Progetti da solo. Lo puoi fare con l'aiuto di molti script di esempio e altri tutorial, che puoi trovare in internet.

Se stai cercando dei microelettronica e accessori di alta qualità , AZ-Delivery Vertriebs GmbH è l'azienda giusta dove potrai trovarli. Ti forniremo numerosi esempi di applicazioni, guide di installazione complete, e-book, librerie e l'assistenza dei nostri esperti tecnici.

<https://az-delivery.de>

Buon divertimento!

Impressum

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