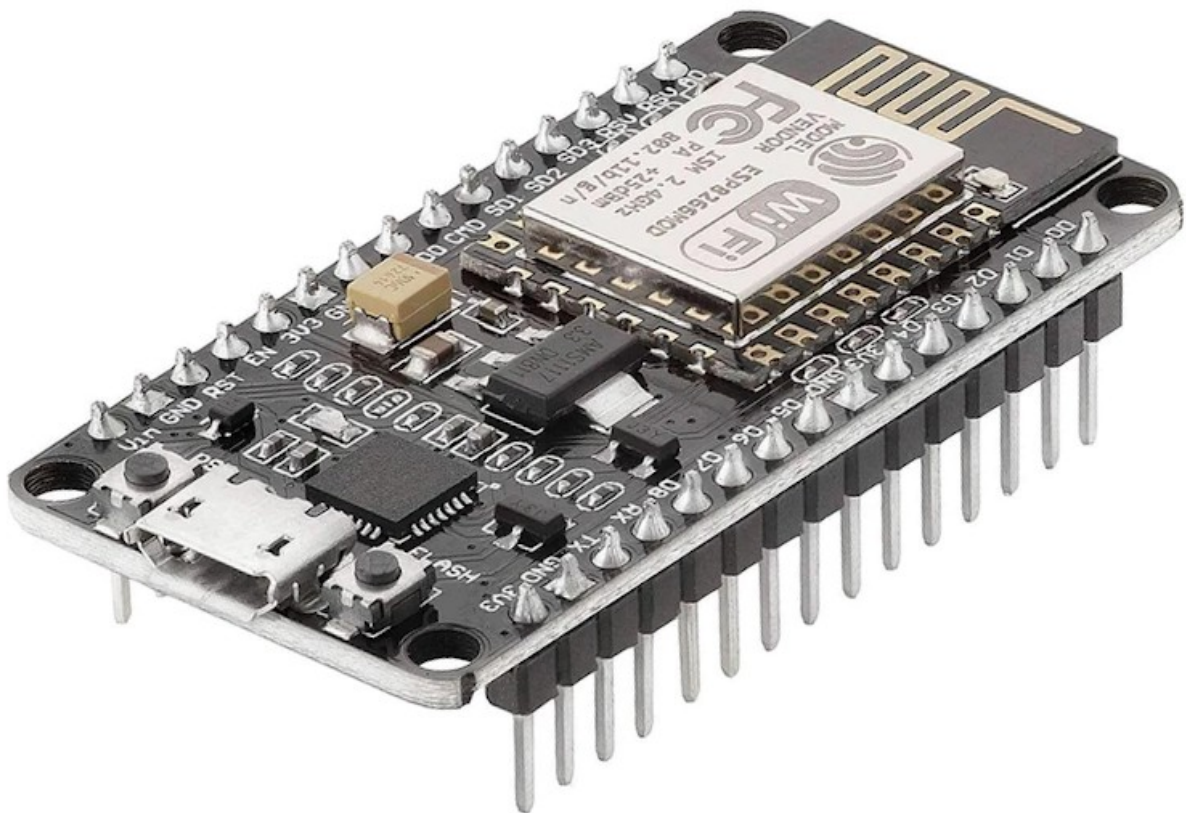


AZ-Delivery

Benvenuto!

Grazie per aver acquistato il nostro *NodeMCU LUA Amica V2 AZ-Delivery*.
Nelle pagine seguenti, ti illustreremo come utilizzare e configurare questo pratico dispositivo.

Buon divertimento!





NodeMCU LUA Amica V2 è una scheda di sviluppo creata attorno al chip *ESP8266*, contenente il regolatore di tensione e il circuito programmatore USB per il chip *ESP8266* e alcune altre funzionalità.

Per lo sviluppo di applicazioni è possibile scegliere tra l'IDE Arduino e la lingua LUA. La community di utenti di Arduino è molto attiva e supporta piattaforme come ESP8266.

NodeMCU viene fornito con un firmware preinstallato che ci consente di lavorare con il linguaggio interpretato LUA, inviando comandi attraverso la porta seriale (*chip CP2102*). La scheda NodeMCU è una delle piattaforme più utilizzate per i progetti Internet of Things (IoT). NodeMCU LUA Amica V2 è pienamente compatibile con Arduino IDE.

La scheda NodeMCU è appositamente progettata per funzionare su breadboard. Ha un regolatore di tensione a piastre che gli consente di alimentare direttamente dalla porta USB. I pin di input/output funzionano a 3.3V. Il chip *CP2102* è responsabile della comunicazione da USB a seriale.



Specifiche:

- » Tensione di alimentazione (USB): 5V DC
- » Tensione di ingresso/uscita: 3.3V DC
- » SoC: ESP8266 (Modulo ESP-12)
- » CPU: Tensilica Xtensa LX3 (32 bit)
- » Frequenza di Clock: 80MHz / 160MHz
- » RAM di avvio: 32kB
- » Data RAM: 96kB
- » Memoria flash esterna: 4MB
- » Pin digitali GPIO: 17 (può essere configurata come PWM at 3.3V)
- » Pin ADC analogico: 1 (MA il range di tensione è: 0 - 1V)
- » UART: 2
- » Chip seriale USB: CP2102
- » Antenna PCB
- » 802.11 b/g/n
- » Wi-Fi Direct (P2P), soft-AP
- » Stack Protocollo TCP/IP Integrato
- » Potenza di uscita di +19.5 dBm in modalità 802.11b
- » Corrente di dispersione inferiore a 10uA
- » Si accende e trasmette i pacchetti in <2ms
- » Consumo energetico in standby <1.0mW (DTIM3)

Az-Delivery



La serie ESP8266 di chip Wi-Fi è prodotta da Espressif Systems, una società di semiconduttori di Shanghai (Cina). ESP8266 è un modulo Wi-Fi conveniente 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, UART e altri. La parte migliore è che viene fornito con la rete wireless inclusa, il che lo distingue da altri microcontrollori come Arduino. Ciò significa che puoi facilmente controllare e monitorare i dispositivi in remoto tramite Wi-Fi a un prezzo accessibile.

ESP8266 è un system-on-chip (SoC) che integra un microcontrollore Tensilica a 32 bit, interfacce periferiche digitali standard, interruttori antenna, balun RF, amplificatore di potenza, amplificatore di ricezione a basso rumore, filtri e moduli di gestione dell'alimentazione in un piccolo pacchetto. Fornisce Wi-Fi a 2,4 GHz (802.11 b/g/n, supporta WPA e WPA2), 17 pin GPIO, interfaccia I2C (IIC), conversione da analogico a digitale (10 bit, su un pin), interfaccia SPI, UART (su pin dedicati, oltre a UART di sola trasmissione può essere abilitato su GPIO2) e PWM (Pulse Width Modulation) nel software su ogni pin GPIO.

Az-Delivery

Il core del processore, chiamato "*L106*" da Espressif, è basato sul core del controller Tensilica Diamond Standard 106Micro a 32 bit e funziona a 80MHz. Ha una ROM di avvio da 64 kB, una RAM delle istruzioni da 32 kB e una RAM dei dati utente di 80 kB. È possibile accedere alla memoria flash esterna tramite l'interfaccia SPI.

Di conseguenza, i fornitori hanno creato molti moduli pcb compatti basati sul chip ESP8266 (come NodeMCU LUA Amica V2). Alcuni di questi moduli hanno identificatori specifici, come "*ESP-01*" tramite "*ESP-14*". I moduli basati su ESP8266 hanno dimostrato di essere una piattaforma multiuso a basso costo, collegabile in rete e globale per facilitare lo sviluppo dell'endpoint IoT.



NodeMCU

NodeMCU è un firmware open source e un kit di sviluppo che ti aiuta a prototipare il tuo prodotto IoT in poche righe di script Lua. NodeMCU è una piattaforma IoT open source. Include il firmware che gira su ESP8266 e l'hardware che si basa sul modulo ESP-12. Il termine "NodeMCU" si riferisce al firmware anziché ai kit di sviluppo. Il firmware utilizza il linguaggio di scripting Lua. Si basa sul progetto eLua e si basa sull'SDK Non-OS Espressif per ESP8266.

Caratteristiche del NodeMCU:

- Kit di Sviluppo – basato su ESP8266
- Hardware IO simile ad Arduino
- API di rete in stile Nodejs: API basata su eventi per applicazioni di rete
- WI-FI a basso costo



Differenza tra ESP8266 (NodeMCU) e Arduino UNO

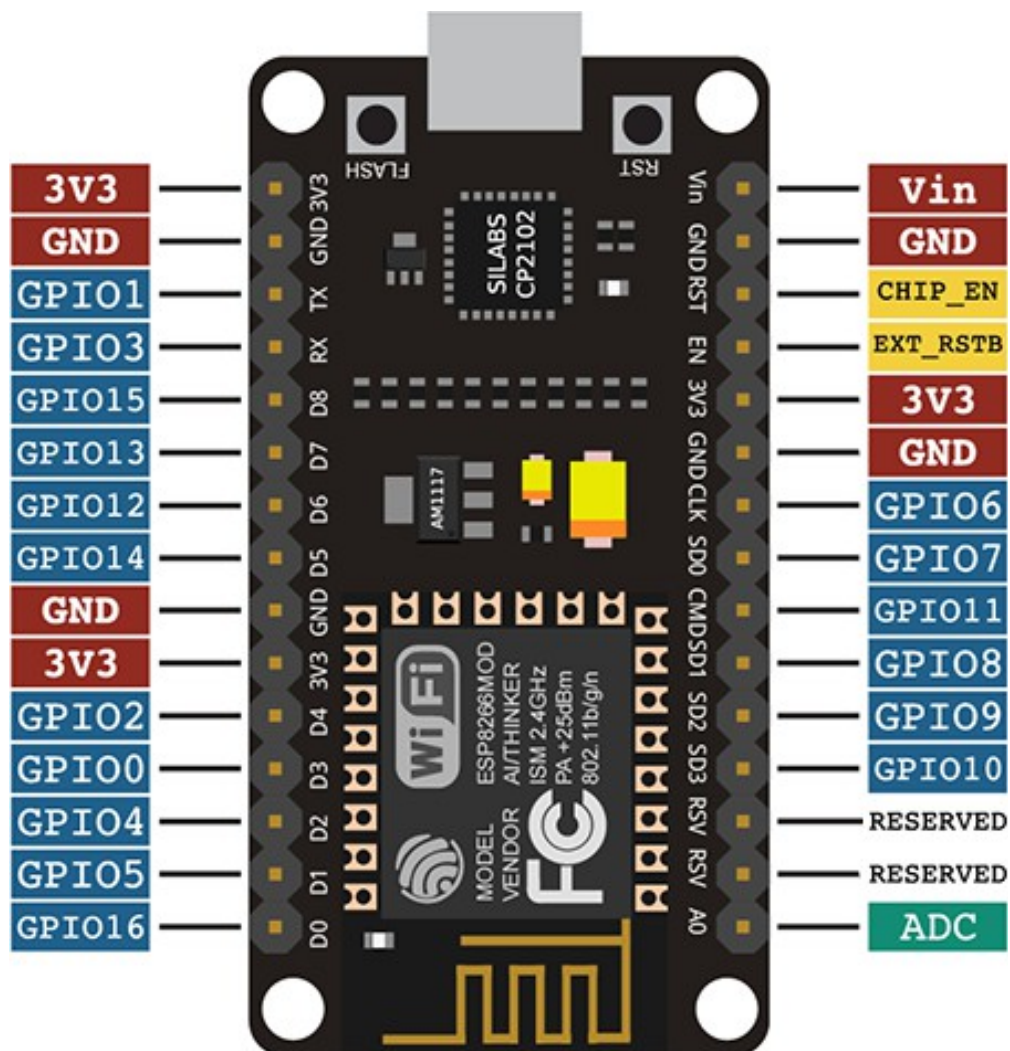
| Specifiche | Uno | ESP8266 |
|-----------------------|------------|--------------------------|
| RAM: | 4kB | 80kB |
| Memoria FLASH: | 32kB | 4MB |
| Velocità: | 16MHz | 80MHz |
| GPIOs (utilizzabile): | 14 | 11 |
| Livello tensione I/O: | 5V | 3.3V |
| ADC (risoluzione): | 6 (10-bit) | 1 (10-Bit) |
| Interfaccia seriale: | 1 | 1 |
| Interfaccia I2C: | 1 | 1 |
| interfaccia SPI: | 1 | Usata dal chip flash |
| PWM, risoluzione: | 6, 8 bit | Tutti i pin GPIO, 10 bit |
| WiFi | No | Si 2MBps |



Differenza tra ESP8266 e ESP32

| Specification | ESP8266 | ESP32 |
|-------------------------------|--------------------|---------------------|
| MCU: | Xtensa Single-core | Xtensa Dual-Core |
| | 32-bit L106 | 32-bit LX6 |
| 802.11 b/g/n Wi-Fi: | HT20 | HT40 |
| Bluetooth: | No | Bluetooth 4.2 e BLE |
| Frequenza tipica: | 80MHz | 160MHz |
| SRAM: | No | Yes |
| Flash: | No | Yes |
| Pin GPIO: | 17 | 36 |
| HW/SW PWM: | Nessuno/ 8 canali | Nessuno/ 16 canali |
| SPI/I2C/I2S/UART: | 2/1/2/2 | 4/2/2/2 |
| ADC: | 10-bit | 12-bit |
| CAN: | No | Si |
| Interfaccia MAC Ethernet: | No | Si |
| Sensore touch: | No | Si |
| Sensore temperatura: | No | Si |
| Sensore effetto hall: | No | Si |
| Temperatura di funzionamento: | -40°C a 125°C | -40°C a 125°C |

Piedinatura NodeMCU LUA Amica V2





Descrizione Pin GPIO

Proprio come una normale scheda Arduino, l'ESP8266 ha pin di ingresso / uscita digitali (pin GPIO - pin di ingresso / uscita per uso generico). Questi ingressi / uscite digitali funzionano a 3.3V.

La tensione a 5 V non deve essere collegata a nessun pin chip ESP8266!

I pin non tollerano 5 V, l'applicazione di più di 3,6 V su qualsiasi pin distruggerà il chip.

La massima corrente che può essere estratta da un singolo pin GPIO è 12mA.

L'ESP8266 ha 17 pin GPIO, tuttavia è possibile utilizzarne solo 11, poiché 6 pin (GPIO 6 - 11) vengono utilizzati per collegare il chip di memoria flash. Questo è il piccolo IC a 8 pin proprio accanto all'ESP8266. Se si tenta di utilizzare uno di questi pin, è possibile che si verifichi un arresto anomalo del programma.

GPIO 1 e 3 sono utilizzati come TX e RX della porta seriale hardware (UART), quindi nella maggior parte dei casi non è possibile utilizzarli come I / O normali durante l'invio / la ricezione di dati seriali.



Modalità di avvio

Pochi pin I/O hanno una funzione speciale durante l'avvio, selezionano una delle tre modalità di avvio:

| GPIO15 | GPIO0 | GPIO2 | Modalità |
|--------|-------|-------|---------------------------------------|
| 0V | 0V | 3.3V | Bootloader UART |
| 0V | 3.3V | 3.3V | Sketch Boot (SPI flash) |
| 3.3V | X | X | Modalità SDIO (non usata per Arduino) |

NOTA: La resistenza pull down esterna di $1k\Omega$ è richiesta su GPIO0, la resistenza pull up esterna su GPIO2 non è richiesta, il pull up interno su questo pin è abilitato all'avvio.

- Il GPIO15 è sempre tirato LOW, quindi non è possibile utilizzare la resistenza di pull-up interna su questo pin. Tienilo a mente quando usi GPIO15 come input per leggere un interruttore o collegarlo a un dispositivo con un'uscita open-collector (o open-drain), come I2C.
- GPIO0 è tirato su HIGH durante il normale funzionamento.
- GPIO2 non può essere LOW all'avvio, quindi non è possibile collegare un interruttore ad esso.

Resistenze interne pull up/down

GPIO 0-15 hanno tutti una resistenza pull up integrata, proprio come in un Arduino. GPIO16 ha un resistore pull down incorporato.



Pin I/O digitali

È possibile impostare la funzione di un pin utilizzando:

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

dove *pin* è il numero GPIO,

e *mode* può essere sia *INPUT*, di default, *OUTPUT*, o *INPUT_PULLUP* per abilitare le resistenze pull up integrate per i GPIO 0 - 15. Per abilitare le resistenze pull-down per GPIO16, usare *INPUT_PULLDOWN_16*.

Per configurare un pin di uscita HIGH (3.3V) o LOW (0V), usare:

```
digitalWrite(pin, value)
```

dove *pin* è il pin digitale,

e *value* o 1 o 0 (o HIGH e LOW).

Per leggere un input, utilizzare `digitalRead(pin)`.



PWM – Pulse-Width Modulation

ESP8266 supporta il software PWM su tutti i pin digitali. La risoluzione PWM predefinita è 10 bit a 1 kHz, ma può essere modificata. Per abilitare PWM su un determinato pin, utilizzare:

```
analogWrite(pin, value)
```

dove *pin* è il pin digitale,

e *value* un numero tra 0 e 1023.

Puoi cambiare il range (profondità bit) dell'output PWM usando `analogWriteRange(range)`.

La frequenza può essere cambiata usando

```
analogWriteFreq(frequency)
```

La frequenza deve essere tra 100Hz e 1000Hz.



Input analogico

L'ESP8266 ha un singolo pin di ingresso analogico, con un intervallo di tensione di ingresso da 0,0 V a 1,0 V. Se si fornisce una tensione di 3,3 V, ad esempio, si danneggerà il chip. Il NodeMCU ha un partitore di tensione resistivo a bordo, per ottenere un intervallo più facile di 0 - 3,3V. L'ADC (analog to digital converter) ha una risoluzione di 10 bit. L'ESP può anche utilizzare l'ADC per misurare la tensione di alimentazione (VCC). Per fare ciò, includere

```
ADC_MODE(ADC_VCC)
```

all'inizio del tuo sketch, e usare:

```
ESP.getVcc()
```

per ottenere effettivamente la tensione. Se lo si utilizza per leggere la tensione di alimentazione, non è possibile collegare nient'altro al pin analogico.



Comunicazione Seriale

L' ESP8266 ha due hardware UARTS (Porte seriali):

- UART0 su pins 1 e 3 (TX0 e RX0 risp.), e
- UART1 su pins 2 e 8 (TX1 e RX1 risp.), tuttavia, GPIO8 viene utilizzato per collegare il chip flash. Ciò significa che UART1 può solo trasmettere dati.

Inoltre UART0 ha il controllo del flusso hardware sui pin 15 e 13 (rispettivamente RTS0 e CTS0). Questi due pin possono essere utilizzati anche come pin TX0 e RX0 alternativi.

Per usare UART0 (TX = GPIO1, RX = GPIO3), puoi usare l'oggetto Seriale, proprio come su un Arduino: `Serial.begin(baud)`

Per abilitare i pin alternativi (TX = GPIO15, RX = GPIO13), usare: `Serial.swap()` dopo `Serial.begin()`.

Per abilitare UART1 (TX = GPIO2), usare l'oggetto `Serial1`.



Comunicazione WiFi

L'ESP può operare in tre diverse modalità: Wi-Fi station, punto di accesso Wi-Fi ed entrambi contemporaneamente.

Altre caratteristiche

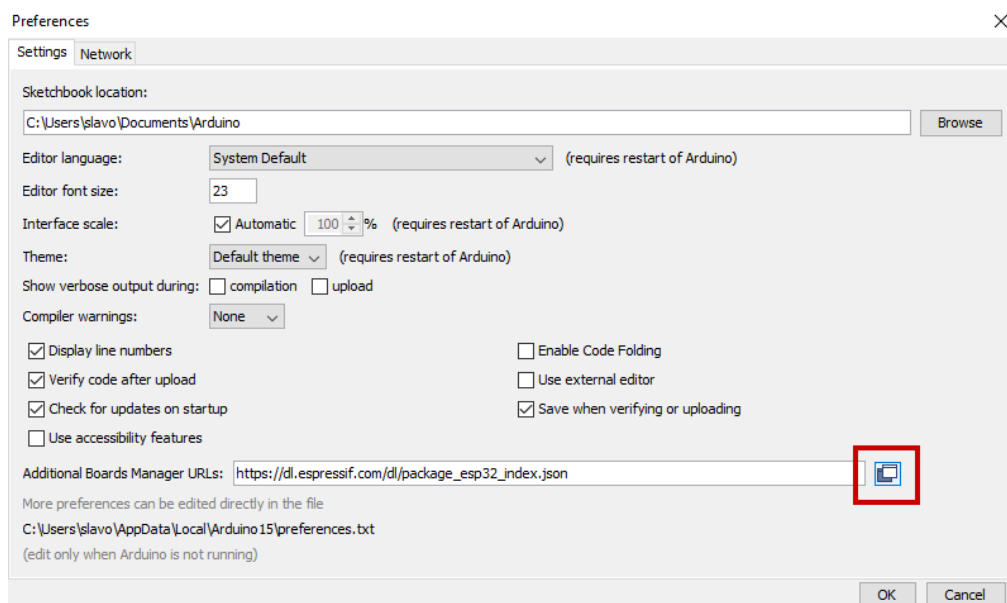
Una delle caratteristiche della NodeMCU LUA Amica V2 è la sua capacità di funzionare come Access Point o hotspot per il tuo progetto Wi Fi. Inoltre, è possibile eseguire un server Web su NodeMCU. Un'altra funzione consente di caricare il codice su NodeMCU LUA Amica V2 via Internet. Si chiama OTA – over-the-air programming e questo è un processo che consente ai dispositivi di aggiornare il proprio firmware o software in modalità wireless senza alcun accesso fisico (tramite Wi-Fi, Bluetooth, GPRS o 4G / 3G). Queste funzioni non sono trattate in questo eBook.

Come usare l'ESP8266/NodeMCU con Arduino IDE

Per utilizzare NodeMCU con Arduino IDE, seguire alcuni semplici passaggi. Il primo è installare il core ESP8266. Per installarlo, apri l'Arduino IDE e vai a:

File > Preferenze,

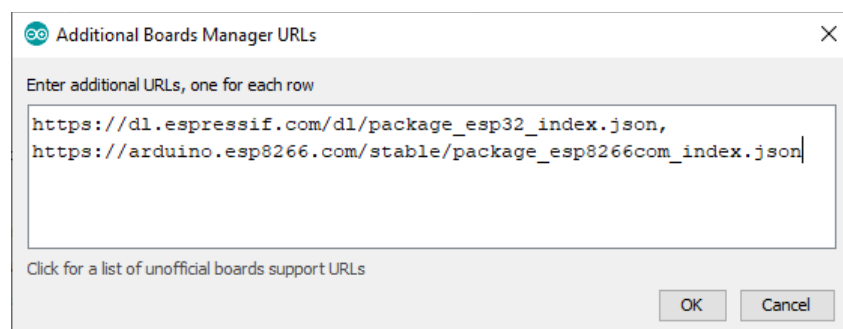
e trova i campi URL aggiuntivi.



Poi copia il seguente URL:

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

Incolla questo link nel campo URL aggiuntivi. Se hai già uno o più link all'interno di questo campo, aggiungi solo una virgola dopo l'ultimo collegamento, incolla il nuovo collegamento dopo la virgola e fai clic sul pulsante OK. Quindi chiudi l'IDE di Arduino.

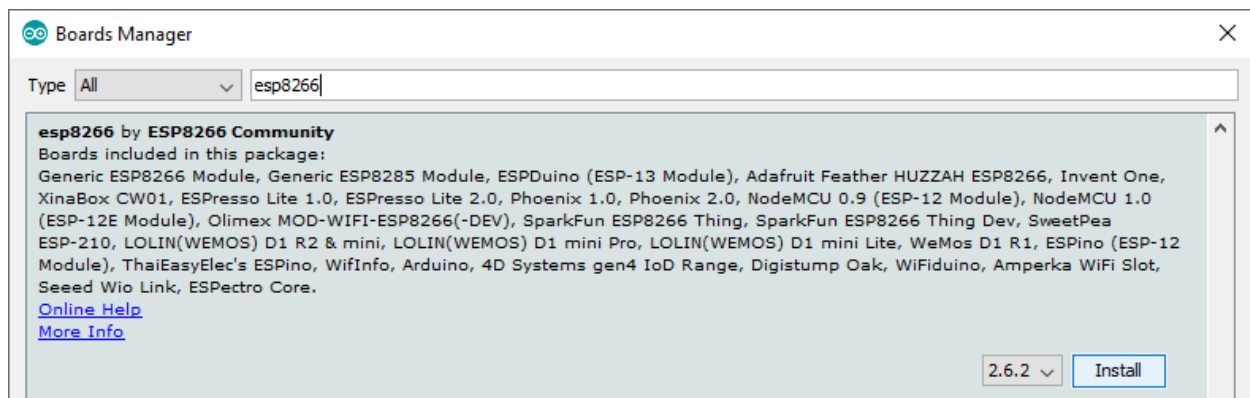


Az-Delivery

Apri di nuovo Arduino IDE e vai a:

Strumenti > Scheda > Gestione Schede

Si aprirà una nuova finestra, digitare “esp8266” nella casella di ricerca e installare la scheda chiamata “esp8266” made by “ESP8266 Community”, come mostrato nell'immagine qui sotto:



Ora hai installato l'ESP8266 core.

Per selezionare la scheda NodeMCU LUA Amica V2, vai su:

Strumenti > Scheda > NodeMCU 1.0 (Modulo ESP - 12E)

Per caricare il codice dello sketch sulla scheda NodeMCU, selezionare innanzitutto la porta su cui è stata collegata la scheda. Vai su:

Strumenti > Porta > {port name}



Lampeggiamento di due LED a bordo della NodeMCU LUA Amica V2

A bordo della NodeMCU LUA Amica V2 sono presenti due LED . Un LED è collegato al pin GPIO 2 e si trova sulla scheda ESP8266. L'altro LED è collegato al pin GPIO 16 e si trova sulla scheda NodeMCU. Se usi lo sketch esempio *Blink* di default che viene fornito con l'Arduino IDE, la macro `LED_BUILTIN` rappresenta i LED collegati al pin 2 GPIO. Per far lampeggiare questi due LED, il seguente codice è l'esempio di sketch:

```
#define led_built_in_ESP    2;
#define led_built_in_Node  16;
void setup() {
    pinMode(led_built_in_ESP, OUTPUT);
    pinMode(led_built_in_Node, OUTPUT);
}
void loop() {
    digitalWrite(led_built_in_ESP, HIGH);
    digitalWrite(led_built_in_Node, LOW);
    delay(1000);
    digitalWrite(led_built_in_ESP, LOW);
    digitalWrite(led_built_in_Node, HIGH);
    delay(1000);
}
```



PWM – Pulse Width Modulation

Useremo la PWM per far sbiadire un LED collegato al pin 2 GPIO. Codice di sketch:

```
#define LED 2
uint16_t brightness = 0; // how bright the LED is
uint8_t fadeAmount = 5; // how many points to fade the LED by
void setup() {
    pinMode(LED, OUTPUT);
}
void loop() {
    analogWrite(LED, 0);
    delay(2000);
    analogWrite(LED, 512);
    delay(2000);
    analogWrite(LED, 1023);
    delay(2000);
    while(1) {
        analogWrite(LED, brightness);
        brightness = brightness + fadeAmount;
        if(brightness <= 0 || brightness >= 1023) {
            fadeAmount = -fadeAmount;
        }
        delay(15);
    }
}
```

Ce l'hai fatta, ora puoi usare il tuo modulo per i tuoi progetti.



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 prodotti di alta qualità per il tuo Arduino e Raspberry Pi, 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>