

# Documentazione Challenge

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Materiale Hardware</b>	<b>3</b>
<b>3</b>	<b>Software Necessario</b>	<b>4</b>
<b>4</b>	<b>Conoscenza della Board</b>	<b>6</b>
4.1	ESP8266 e l'IDE di Arduino . . . . .	7
4.2	Caratteristiche Tecniche(Essenziali) . . . . .	8
4.3	WIFI . . . . .	9
<b>5</b>	<b>Conclusione</b>	<b>9</b>

# 1 Introduzione

Come riportato nell'annuncio ufficiale dell'evento, ecco la documentazione per poter partecipare alla challenge. Nel seguente testo non sarà riportata alcuna informazione riguardo alla competizione, in quanto tutte le informazioni in merito, quali obiettivi e metodi di valutazione, saranno fornite il giorno dell'evento.

Viene data per assodata la conoscenza della programmazione di base e un minimo di confidenza con board Arduino o compatibili. In caso negativo vi rimandiamo al sito ufficiale di [Arduino](#). Tali conoscenze sono necessarie per poter essere competitivi durante la challenge.

I nostri sponsor hanno messo a disposizione delle board ESP8266 12-E che, grazie ad una community di sviluppatori, è completamente programmabile con il codice prodotto dall'IDE di Arduino.

Per evitare che le board vengano danneggiate durante la saldatura, necessaria per collegarle all'adattatore usb/seriale (Profilic PL2303) per poterle programmare, abbiamo provveduto già a saldarle in modo tale da renderle direttamente utilizzabili out-of-kit.

Fatte queste premesse, passiamo alle conoscenze e al materiale (Hardware e Software) necessario per la challenge.

## 2 Materiale Hardware

- 1) PC con una versione del sistema operativo GNU/Linux installata (preferibile Ubuntu 16.04). Tale scelta è stata motivata dal fatto che la missione della nostra associazione è diffondere Linux e in generale il software libero. Inoltre sussiste anche un un motivo tecnico: i convertitori USB/Seriale che abbiamo a disposizione non sono compatibili con versioni successive a Windows 7.
- 2) Saldatore;
- 3) Tronchesine;
- 4) Stagno;
- 5) Multimetro;
- 6) Forbici o taglierino.

### 3 Software Necessario

- 1) [Fritzing](#);
- 2) Scaricare da [qui](#) i componenti e il circuito per Fritzing;
- 3) [Arduino IDE versione 1.6.5-r5](#), attualmente unica compatibile con la board;
- 4) Compilatore per ESP8266 in Arduino, in modo tale che l'IDE possa produrre codice compatibile per l'ESP.

Per effettuare questa operazione basta seguire i seguenti passi :

Per effettuare questa operazione basta seguire i seguenti passi :

- a) Avviare Arduino IDE ed aprire “File” > “Impostazioni” ed inserire [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) in “Additional Boards Manager URLs”:

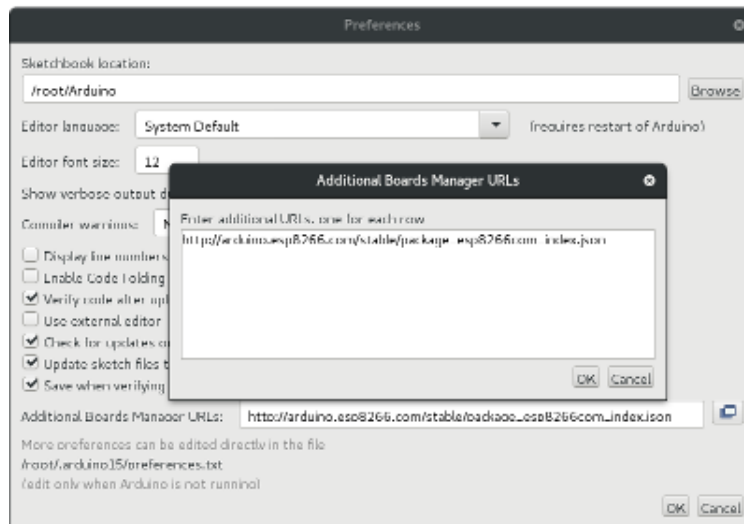


Figura 1: Inserimento Link

b) Aprire “Strumenti” > “Scheda” > “Boards Manager” ed installare “esp8266 by ESP8266 Community”:

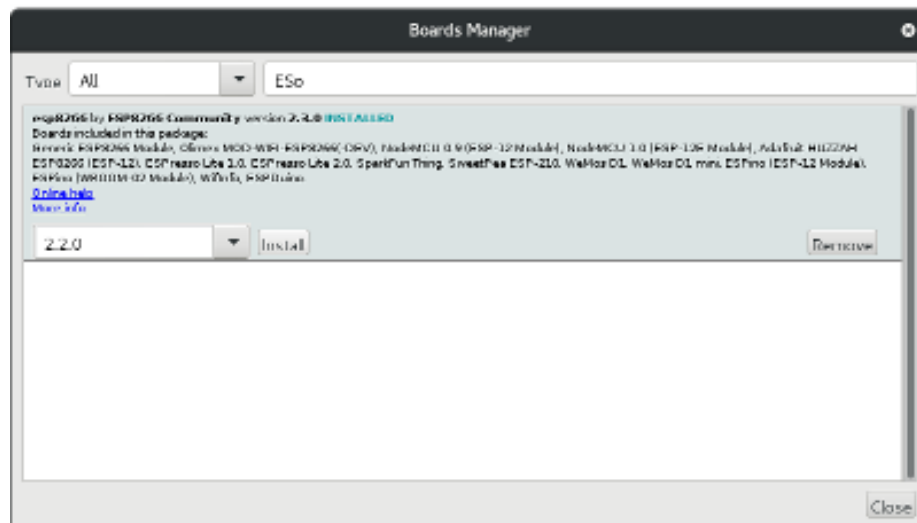


Figura 2: Inserimento Link

## 4 Conoscenza della Board

Prima di procedere con l'illustrazione delle conoscenze necessarie per poter utilizzare la board durante la competizione è importantissimo fare la seguente premessa: “L'ESP8266 è un microcontrollore con antenna WI-FI a basso costo prodotto dall'azienda cinese EspressIf, ma col tempo si è scoperto essere molto più di quanto annunciato al suo rilascio; infatti ora è utilizzabile come una board simile ad un Arduino a bassissimo costo e di dimensioni ridottissime. Sembrerebbero esserci solo vantaggi con questa board ma c'è un piccolo problema: non esiste una documentazione ufficiale, infatti per avere tutte le informazioni necessarie al suo funzionamento, specialmente hardware, si è proceduto di reverse-engineering”. Quindi, purtroppo, non potremo fornirvi informazioni dettagliatissime, ma siamo sicuri saranno per la maggior parte corrette e non porteranno alla distruzione della board in quanto già ampiamente testate dai nostri smanettoni. E con questo, “HAPPY HACKING”.

## 4.1 ESP8266 e l'IDE di Arduino

Come prima cosa per utilizzare l'ESP8266 con l'IDE di Arduino dobbiamo cambiare le opzioni dell'IDE.

- 1) Recarsi in "Tool" > "Board" > "Olimex MOD-WIFI-ESP8266(-DEV)";
- 2) Selezionare in "Upload Speed" > "57600";
- 3) Selezionare la porta USB corretta.

Per quanto riguarda l'utilizzo della seriale bisogna settare un baud rate di "57600"; avremo pertanto la seguente chiamata per inizializzare la seriale: "Serial.begin (57600)". Una volta fatto ciò, bisogna settare il baud rate anche nel monitor seriale come si evince dall'immagine in fig.3.

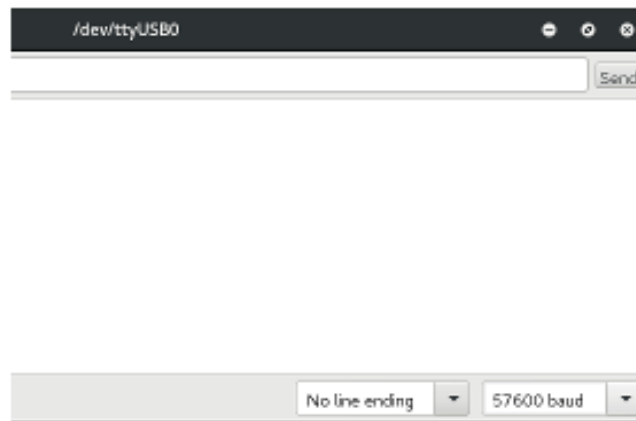


Figura 3: Inserimento Link

Inoltre, per effettuare il flash dello sketch da voi scritto dovete ponticellare i fili blu e nero. Qualora l'upload non dovesse andare a buon fine, rimuovete l'adattatore e ricollegatelo. NOTA: Se non risulta possibile accedere alla seriale avviate Arduino IDE come amministratore (da terminale recarsi nella cartella dove avete estratto l'IDE e digitare "sudo ./arduino").

## 4.2 Caratteristiche Tecniche(Essenziali)

La board ha una tensione di alimentazione che non supera i 3.3 V; inoltre, altra info HW importantissima, è che l'ADC di cui è fornito può convertire tensioni da 0 a 1 V. Come ogni board di sviluppo, anche l'ESP ha una serie di pin GPIO che permettono di interconnetterla e farla interagire con l'ambiente circostante, in fig.4 ritroviamo la piedinatura della board. Bisogna però premettere che nella programmazione della board mediante l'IDE di Arduino non ritroveremo la spessa piedinatura; infatti viene effettuato una sorta di mapping uno-a-uno coi pin che l'IDE di Arduino, che ci permette di programmarla molto semplicemente; tale mapping lo ritroviamo nella fig.5.

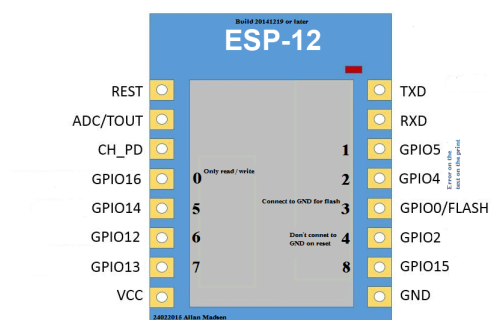


Figura 4: Pin OUT standard

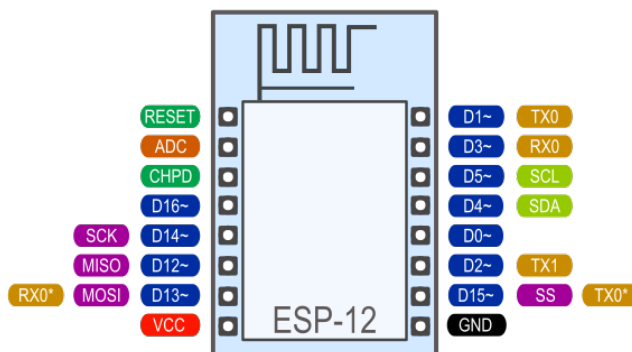


Figura 5: Pin OUT Arduino IDE

Dalla fig.2 notiamo che, per utilizzare il pin GPIO15 nell'ide di Arduino, basterà eseguire le funzioni di I/O sul pin 15, come ad esempio `DigitalWrite(15,HIGH)`.

Inoltre notiamo che abbiamo a disposizione un solo ADC, il quale - per chi non lo sapesse - è un dispositivo HW che ci permette di convertire segnali da



analogico a digitale, come ad esempio un valore di tensione in un numero, che poi possiamo utilizzare nel nostro controllore per effettuare delle elaborazioni. Teniamo ancora a precisare che al massimo l'ADC può effettuare letture con valori di tensioni fino ad 1V.

### 4.3 WIFI

Come detto precedentemente, ESP è nato come antenna WI-FI, pertanto il team che ha permesso di utilizzare ESP con l'IDE di Arduino ha scritto anche una serie di funzioni che permettono di utilizzare la parte WI-FI. Per utilizzare tali funzionalità, basta utilizzare la libreria <ESP8266WiFi.h>, che ritroviamo già presente nella nostra IDE. Inoltre, sono state fornite una serie di altre librerie per realizzare client o server HTTP e molto altro. Vi consigliamo vivamente di farvi un giro nella sezione esempi dell'IDE, dopo aver scelto di utilizzare la board ESP come citato precedentemente.

## 5 Conclusione

Vogliamo comunque precisare che quelle sopra citate non sono informazioni complete e soddisfacenti, agli occhi dei più competenti in materia, ma non vogliamo semplificare troppo la challenge, fornendo completamente tutto il materiale per quanto riguarda l'utilizzo della board, poiché effettuare ricerche per apprendere, da documentazioni ufficiali e non, fa anche esso parte della gara. Di seguito lasciamo alcuni link che riteniamo utili ai fini dell'apprendimento.

- 1) <https://www.arduino.cc/en/Reference/HomePage>
- 2) <https://github.com/esp8266/Arduino/blob/master/doc/reference.md>
- 3) <https://github.com/esp8266/Arduino/blob/master/doc/libraries.md#wifiesp8266wifi-library>

E con questo non ci resta che augurarvi buona fortuna e dirvi che la challenge è già iniziata.

Per ulteriori dubbi e informazioni potete contattarci tramite l'e-mail dell'associazione "info@nalug.net" o alla nostra pagina Facebook. Il team NALUG cercherà di rispondervi al più presto.