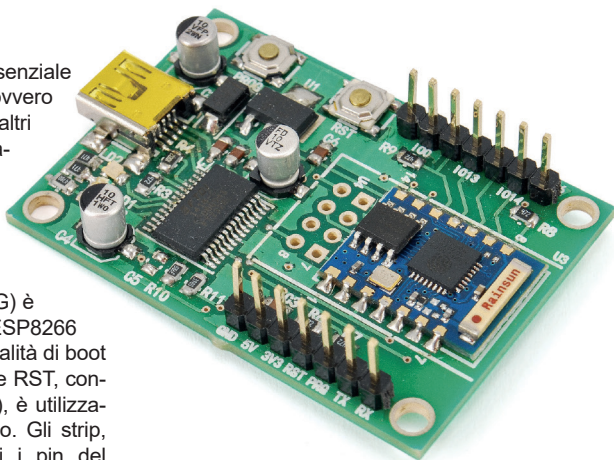


# Demoboard ESP03

(cod. FT1191M)

La demoboard ESP03 contiene l'essenziale per creare semplici applicazioni, ovvero il convertitore USB/seriale e pochi altri componenti necessari al funzionamento. Tra questi troviamo uno stabilizzatore di tensione, alcuni connettori strip e dei pulsanti che semplificano la gestione del modulo. Dall'analisi dello schema è possibile notare che un pulsante (PROG) è connesso al pin GPIO0 del chip ESP8266 ed è utilizzato per attivare la funzionalità di boot mode del modulo, mentre il pulsante RST, connesso al pin CH\_PD (Power Down), è utilizzato come reset hardware del modulo. Gli strip, semplicemente rendono disponibili i pin del chip ESP8266, facilitando, mediante dei cavetti jumper, la connessione a periferiche esterne. La scheda supporta l'utilizzo dei moduli ESP01 (Fig. 1) e ESP03 (Fig. 2); ricordiamo che, mentre il primo utilizza un connettore strip e quindi può essere facilmente sostituito, il secondo va saldato direttamente sullo stampato, il che ne complica l'eventuale sostituzione. Dall'analisi della configurazione interna del chip ESP8266 è possibile notare come esso disponga di tutte le funzionalità tipiche di un microcontrollore con architettura a 32 bit, oltre al modulo di comunicazione UART dispone dell'interfaccia SPI ad alta velocità e dell'interfaccia I<sup>2</sup>C. Tutti



i pin sono configurabili come ingresso o uscita ed alcuni anche come ingressi analogici mentre le uscite analogiche possono essere ottenute tramite il segnale PWM disponibili in alcuni pin.

### Configurare il modulo

Vediamo ora come configurare il modulo per il funzionamento in stand-alone.

La prima soluzione è il progetto chiamato NodeMCU il cui sito di riferimento è [http://nodemcu.com/index\\_cn.html](http://nodemcu.com/index_cn.html).

I sorgenti del progetto e tutti i file attinenti sono disponibili presso il sito:

<https://github.com/nodemcu/nodemcu-firmware>

e <https://github.com/nodemcu/nodemcu-flasher>.

Per prima cosa è necessario installare il firmware in grado di supportare il funzionamento in

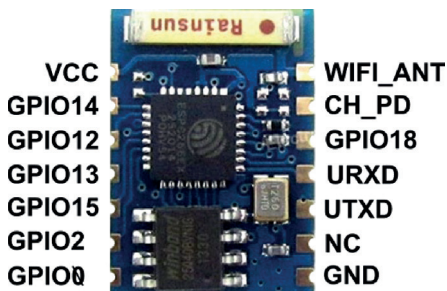


Fig. 1 - Piedinatura del modulo ESP03.

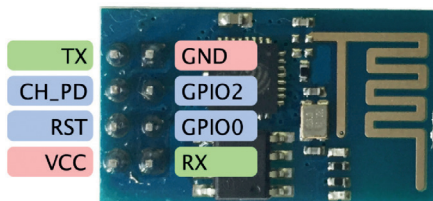
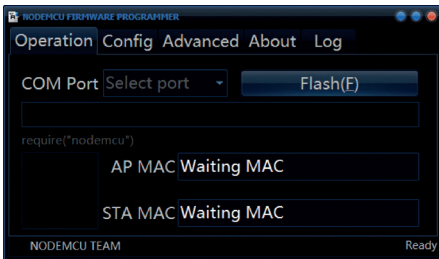
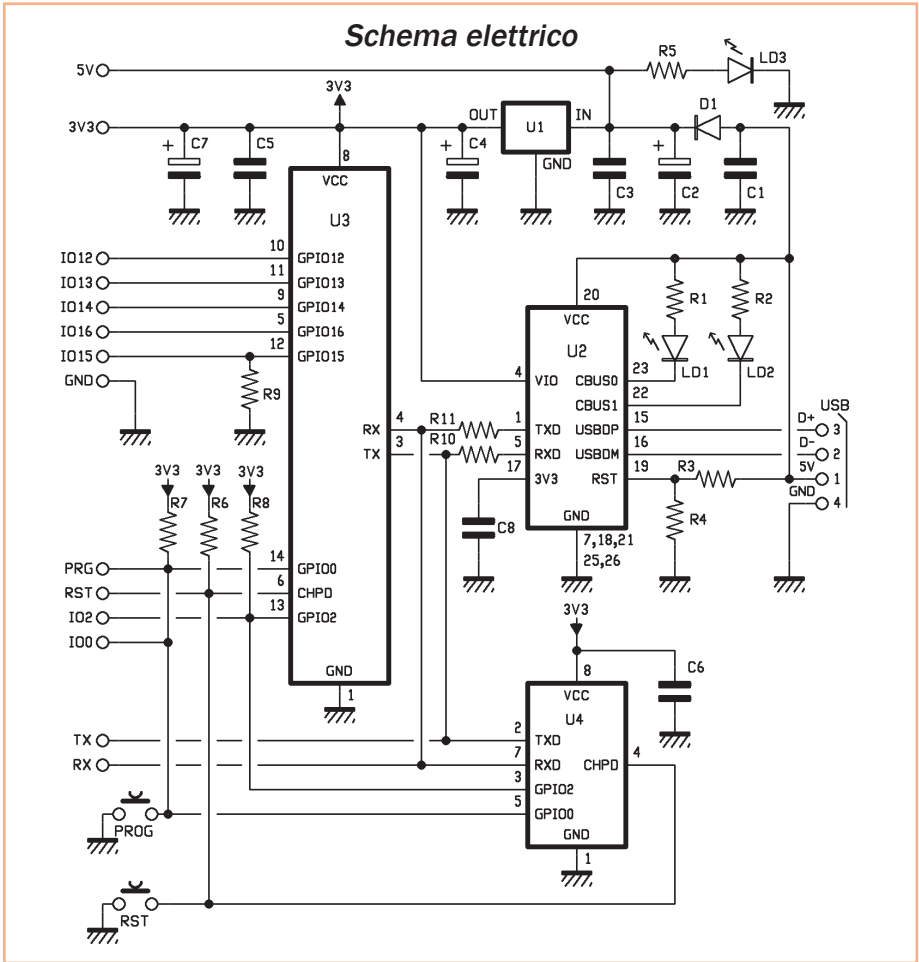


Fig. 2 - Piedinatura del modulo ESP01.



**Fig. 3** - Il software usato per caricare il firmware.

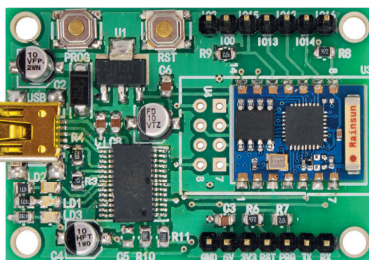
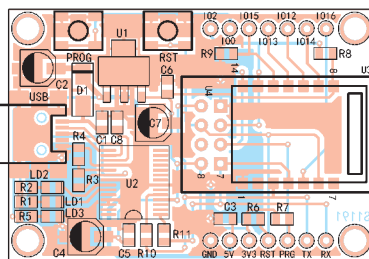


**Fig. 4** - Impostazioni di programmazione.

## [piano di MONTAGGIO]

### Elenco Componenti:

- R1, R2, R5: 470 ohm (0805)
- R3: 4,7 kohm (0805)
- R4: 10 kohm (0805)
- R6÷R9: 4,7 kohm (0805)
- R10, R11: 33 ohm (0805)
- C1: 100 nF ceramico (0805)
- C2: 22  $\mu$ F 6,3 VL elettrolitico ( $\varnothing$ 4mm)
- C3, C5, C6, C8: 100 nF ceramico (0805)
- C4, C7: 22  $\mu$ F 6,3 VL elettrolitico ( $\varnothing$ 4mm)
- C7: 22  $\mu$ F 6,3 VL elettrolitico ( $\varnothing$ 4mm)
- C8: 100 nF ceramico (0805)
- U1: TC1262-3.3VDB
- U2: FT232RL
- U3: ESP03
- U3: ESP01
- RST: Microswitch > Farnell: 2320079
- PROG: Microswitch > Farnell: 2320079
- D1: GF1M
- LD1: LED giallo (0805)
- LD2: LED rosso (0805)
- LD3: LED verde (0805)



Varie:

- Connettore mini USB
- Circuito stampato S1191

stand-alone e per fare questo utilizziamo un apposito software di nome *ESP8266Flasher.exe* disponibile per sistemi Windows a 32 e 64 bit (Fig. 3).

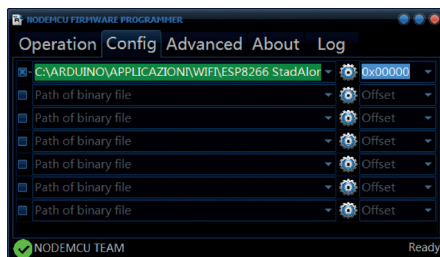
Assicuratevi che il modulo sia connesso al PC, avviate il software e impostate nel menu "ADVANCED" un Baudrate di 9600 (ma anche Baudrate più elevati sono consentiti), una dimensione della Flash di 512 kByte, una velocità

di 40MHz e selezionate come SPI il modo QIO (Fig. 4).

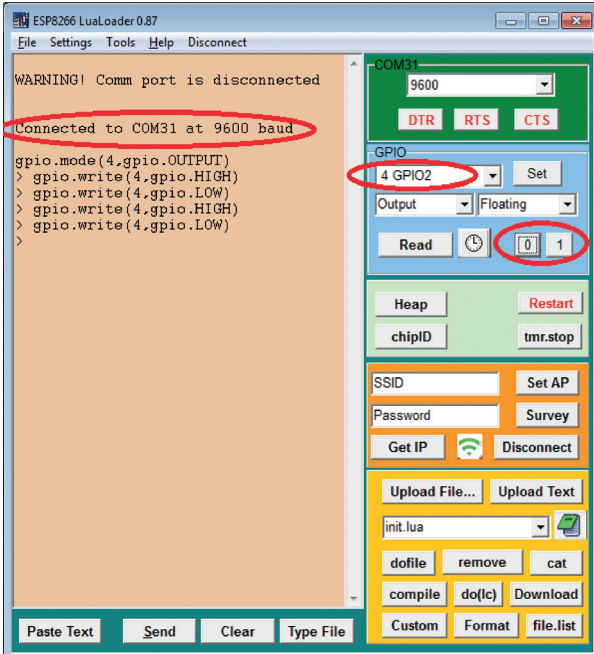
Il firmware da noi usato è contenuto nel file *nodemcu\_integer\_0.9.5\_20150318.bin* quindi apritelo dall'apposito menu (Fig. 5). Nella demoboard, premete e tenete premuto il pulsante PRG, quindi agite per un attimo sul pulsante RST per avviare il modulo in modalità di boot mode, quindi cliccate sul pulsante Flash del software per avviare la scrittura del firmware; dalla scheda LOG del software potrete osservare le operazioni eseguite.

Appena avviato il download potete rilasciare il pulsante PROG. Se tutto è andato a buon fine comparirà una piccola icona verde in basso a sinistra.

A questo punto è possibile interagire con il



**Fig. 5** - Apertura del file .bin contenente il firmware.

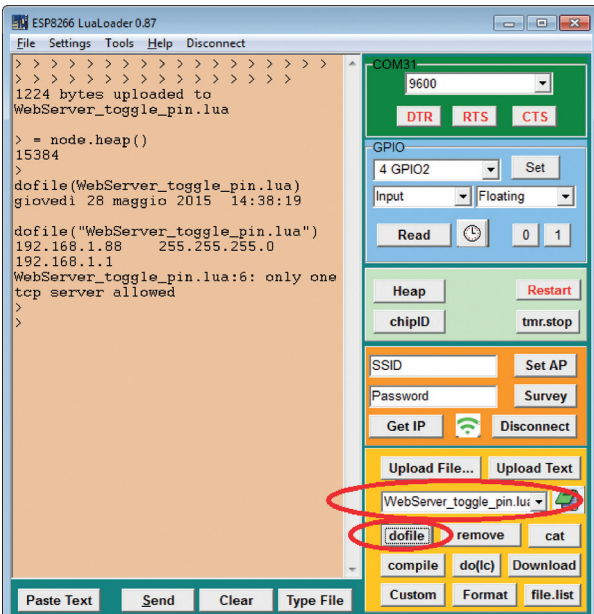


**Fig. 6 - LuaLoader: gestione GPIO.**

modulo tramite l'utilizzo di opportuni ambienti di programmazione, il più semplice ed immediato dei quali si chiama *ESP8266LuaLoader* ed è disponibile in Internet a questo indirizzo: <http://benlo.com/esp8266/>; il file da avviare si chiama *LuaLoader.exe* e si trova su <http://benlo.com/esp8266/index.html#LuaLoader>.

Non appena avviato questo IDE e stabilita la comunicazione con il modulo (dal menu *Settings*), sarete in grado di lavorare con esso in modo semplice ed immediato. Ora connettete un LED con in serie una resistenza da 220 ohm tra il pin GPIO2 e GND e provate a gestirla: utilizzate i comandi del riquadro azzurro per impostare il pin come uscita (pulsante *Set*) ed accendere e spegnere il LED con i pulsanti *0* e *1*.

Usate i comandi del riquadro arancione per impostare la rete WiFi; il pulsante *Survey* permette di conoscere gli AP (Access Point WiFi) disponibili: inserendo il nome della rete e la password è possibile accedervi. Il pulsante *GET IP* mostrerà l'indirizzo IP assegnato dal router al modulo (**Fig. 6**). Questi primi esempi permettono di testare il funzionamento del modulo e di tutto l'hardware ma ancora il modulo non è del tutto autonomo. Infatti al riavvio il modulo è impostato per avviare lo sketch



**Fig. 7 - Gestione file in LuaLoader.**



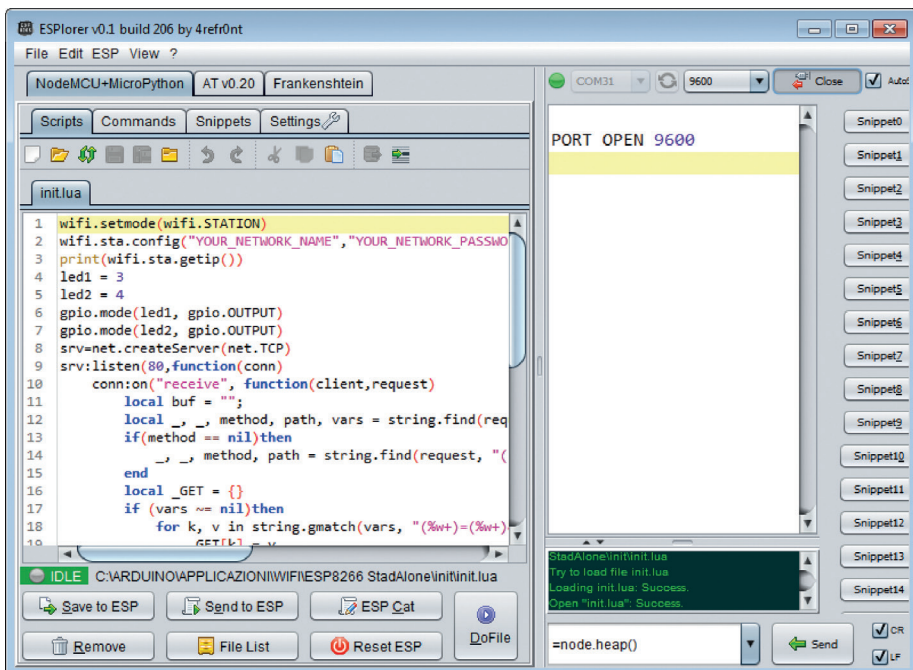
**Fig. 8** - WebServer dell'esempio *WebServer\_toggle\_pin.lua*.

di nome *init.lua* che conterrà tutte le istruzioni necessarie a svolgere il compito previsto. Vediamo come fare quindi a programmare il modulo in maniera permanente.

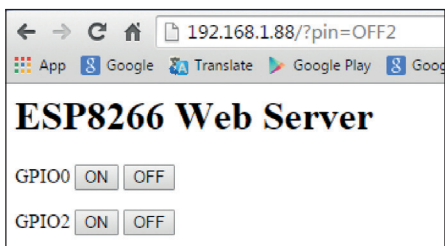
Partiamo con un esempio: aprite con un normale editor di testo, come il Blocco Note, il file *WebServer\_toggle\_pin.lua* che trovate nella cartella esempio del programma LuaLoader, modificate le voci SSID e PASS della riga

*wifi.sta.config("SSID","PASS")* con i dati della vostra rete, rispettivamente il nome e la password di accesso e salvare. Nel riquadro arancione selezionate il file che volete avviare (nel nostro esempio *WebServer\_toggle\_pin.lua*) e cliccate sul comando *dofile* per far eseguire le istruzioni contenute (Fig. 7). Aprite il browser ed inserite l'indirizzo IP che vedete sulla schermata, avrete così la possibilità di accendere e spegnere da remoto il LED connesso a GPIO2 (Fig. 8).

Come già detto, il modulo al riavvio è impostato per avviare lo sketch di nome *init.lua* che conterrà tutte le istruzioni necessarie al compito da svolgere, pertanto quando avete deciso il file da usare, dovrete rinominarlo con il nome *init.lua* e caricarlo sul modulo: ad ogni avvio verrà avviato questo sketch ed eseguite le istruzioni in esso contenute. Non è più necessario che il modulo sia connesso al PC, pertanto è sufficiente alimentare la scheda con una tensione di 5V tramite la porta USB.



**Fig. 9** - ESPlorer: apertura file di esempio.



**Fig. 10** - WebServer dell'esempio  
*WebServer\_toggle\_pins.lua*.

È possibile utilizzare un software di gestione alternativo: si tratta dell'applicazione chiamata *ESPExplorer* scaricabile all'indirizzo <https://github.com/4refr0nt/ESPExplorer/>.

Anche questo, come gli altri programmi, non necessita di installazione ma richiede che sul PC sia installato l'ambiente java; il file da avviare si chiama *ESPExplorer.jar*.

Prendete il file di esempio di nome *WebServer\_toggle\_pins.lua* e rinominatelo in *init.lua*, quindi apritelo dal menu *File>Open from disk* e caricatelo sul modulo con il comando *Save to ESP* (Fig. 9), poi riavviate il modulo con il comando *Reset ESP*, oppure lanciate il file con il comando *DoFile*; sulla schermata comparirà l'indirizzo IP assegnato dal router al modulo: inserite questo indirizzo nel browser per verificare il funzionamento del webserver (Fig. 10).

Analizziamo il listato per meglio comprenderne il funzionamento. Le prime righe sono utilizzate per la connessione alla rete WiFi (in questo caso al router di casa) e, come abbiamo visto, dovranno essere modificate aggiungendo il nome della rete e la password di accesso; lo sketch provvederà alla stampa dell'indirizzo IP assegnato dal router:

```
wifi.setmode(wifi.STATION)
wifi.sta.config("SSID","PASSWORD")
print(wifi.sta.getip())
```

Le successive righe servono ad impostare i pin GPIO come uscite e per avviare il server mettendolo in ascolto sulla porta 80, predefinita per le richieste html:

```
led1 = 3
led2 = 4
gpio.mode(led1, gpio.OUTPUT)
```



**Fig. 11** - WebClient Time dal server Google.

```
gpio.mode(led2, gpio.OUTPUT)
srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
```

Quando arriva una richiesta dal client (il browser) essa verrà elaborata dalla funzione:

```
conn:on("receive",
function(client,request)
```

che provvederà a verificare se la richiesta è compatibile con lo standard html quindi provvederà a costruire la stringa di risposta, ovvero la pagina che sarà inviata al client per la sua visualizzazione. Dalla richiesta del client viene anche estrapolata la stringa relativa al comando (valore ON o OFF) usato per attivare o disattivare i LED; il codice corrispondente è quello di seguito:

```
if(_GET.pin == "ON1")then
  gpio.write(led1, gpio.HIGH);
elseif(_GET.pin == "OFF1")then
  gpio.write(led1, gpio.LOW);
elseif(_GET.pin == "ON2")then
  gpio.write(led2, gpio.HIGH);
elseif(_GET.pin == "OFF2")then
  gpio.write(led2, gpio.LOW);
end
```

Un altro interessante sketch che consente di utilizzare il modulo come client web per richiedere data e ora ai server di Google, il file si chiama *googleTime.lua* (Fig. 11).

In questo caso lo sketch inizia creando la connessione al server di Google con la riga:

```
conn=net.createConnection(net.TCP, 0)
conn:connect(80,'google.com')
```

## Listato 1

```
conn:on("connection",function(conn, payload)
  conn:send("HEAD / HTTP/1.1\r\n"..
    "Host: google.com\r\n"..
    "Accept: */*\r\n"..
    "User-Agent: Mozilla/4.0 (compatible; esp8266 Lua;)"..
    "\r\n\r\n")
end)
```

## Listato 2

```
conn:on("receive", function(conn, payload)
  print('\nRetrieved in `..((tmr.now()-t)/1000)..` milliseconds.')
  print('Google says it is `..string.sub(payload,string.find(payload,"Date: ")
+6,string.find(payload,"Date: ") +35)
  conn:close()
end)
```

Non appena è avvenuta la connessione (*conn:on*) viene inviata la richiesta HTTP verso il server (**Listato 1**). La funzione successiva si occupa di elaborare la stringa di ritorno ed estrapolare i dati richiesti, per poi visualizzarli a video (**Listato 2**). Come finezza viene anche riportato il tempo necessario a completare la richiesta.

### Lavorare con l'IDE Arduino

Pur non essendovi grosse difficoltà, la programmazione non è proprio della massima semplicità. Per prima cosa occorre installare l'ultima versione dell'IDE di Arduino (al momento 1.6.5) o superiore; nelle release recenti l'IDE, oltre ad essere multi-piattaforma è diventato anche multi

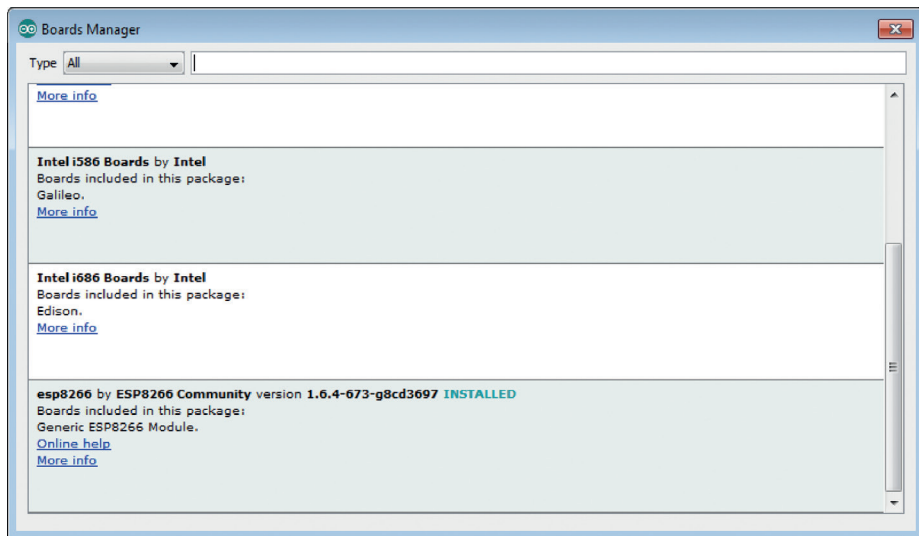
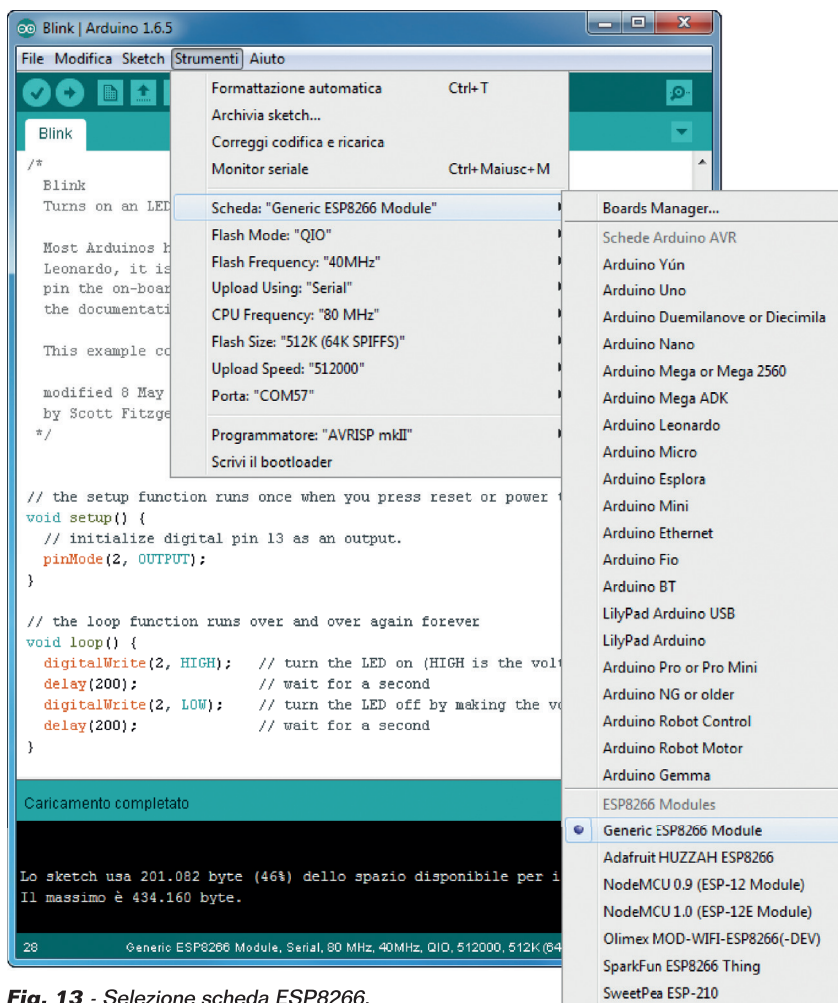


Fig. 12 - Installazione del pacchetto per ESP8266.

board, infatti viene usato per programmare diversi tipi di schede semplicemente installando dei pacchetti aggiuntivi con la procedura descritta qui di seguito. Aprite l'IDE e selezionate il menu *File>Impostazioni*, quindi nel campo *Additional board manager URLs* della finestra che vi appare inserite l'indirizzo [http://arduino.esp8266.com/versions/2.3.0/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/versions/2.3.0/package_esp8266com_index.json). Ora, nella finestra accessibile con il comando di menu *Strumenti>Scheda>Board manager*, cercate *esp8266* by *ESP8266 Com-*

*munity* (**Fig.12**) e fate click su installa. Il supporto per il chip ESP8266 è adesso disponibile.

Se avete incontrato dei problemi in questa procedura potete visionare il sito Internet di riferimento all'indirizzo <https://github.com/esp8266/arduino>; se tutto è andato bene, invece, riavviate l'IDE e selezionate *Strumenti>Scheda>Generic ESP8266 Module* che ora è presente; assicuratevi che le impostazioni siano le stesse mostrate in **Fig. 13** (*Generic ESP8266 Module*, Serial, 80 MHz, 40 MHz, QIO, 512000, 512K) e aprite



**Fig. 13** - Selezione scheda ESP8266.

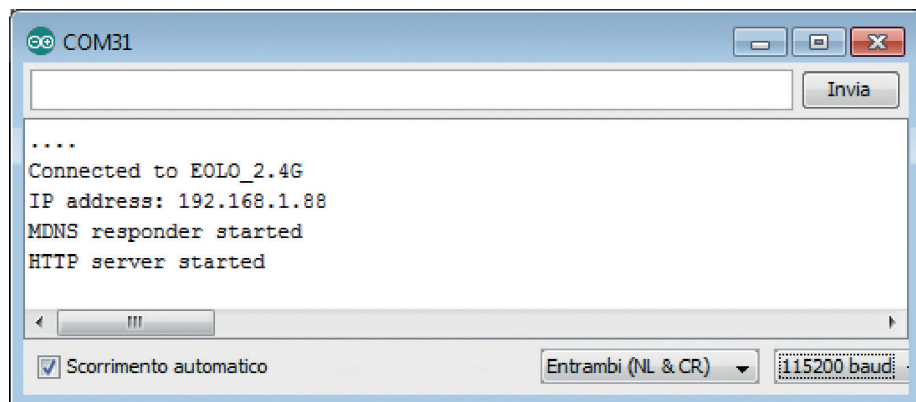


l'esempio blink relativo al lampeggio di un LED (si apre dal menu *File>Esempi>01.Basic*), modificate il pin utilizzato (13) con il nostro GPIO2 (2) al quale avrete connesso un LED con una resistenza in serie. A questo punto dovete trasferire il programma nel modulo ESP, che quindi dobbiamo portare in boot mode con la solita procedura: mantenete premuto il pulsante PROG (GPIO0) e premete per un attimo RST (CH\_PD). Cliccate sul pulsante *Carica* (quello con la freccia) come fareste per caricare uno sketch su Arduino, ed attendete il completamento delle operazioni; non appena avviata la procedura di download potete rilasciare il pulsante PROG. Come avrete intuito non abbiamo usato la funzione di autoreset che viene svolta manualmente quando portiamo il modulo in boot mode. Se avete problemi durante il caricamento potete ridurre la velocità da 512000 a 115200 Baud. A caricamento terminato, premete il pulsante di RST per riavviare il modulo e far partire lo

sketch; il LED inizierà a lampeggiare. Se tutto ha funzionato potete subito sperimentare altre applicazioni, la più classica delle quali è il web server con lo sketch HelloServer, disponibile tra gli esempi del modulo ESP8266. Per caricare ed avviare lo sketch seguite la procedura appena descritta, con una variante: prima di resettare il modulo aprite Serial Monitor ed impostate la velocità di comunicazione a 115200 Baud; ciò vi permetterà di vedere l'indirizzo IP assegnato dal vostro router al modulo. Aprite il browser ed inserite l'indirizzo appena visualizzato per vedere il web server in funzione.

È possibile scaricare dalla scheda del prodotto on-line (cod. FT1191M) un ulteriore esempio di programmazione.

L'articolo completo del progetto è stato pubblicato su: *Elettronica In n. 199*



**Fig. 14** - Indicazione della ESP8266 board con LED su GPIO2 lampeggiante.





**A tutti i residenti nell'Unione Europea****Importanti informazioni ambientali relative a questo prodotto**

Questo simbolo riportato sul prodotto o sull'imballaggio, indica che è vietato smaltire il prodotto nell'ambiente al termine del suo ciclo vitale in quanto può essere nocivo per l'ambiente stesso. Non smaltire il prodotto (o le pile, se utilizzate) come rifiuto urbano indifferenziato; dovrebbe essere smaltito da un'impresa specializzata nel riciclaggio.

Per informazioni più dettagliate circa il riciclaggio di questo prodotto, contattare l'ufficio comunale, il servizio locale di smaltimento rifiuti oppure il negozio presso il quale è stato effettuato l'acquisto.

Distribuito da:

**FUTURA GROUP SRL**

Via Adige, 11 - 21013 Gallarate (VA) Tel. 0331-799775 Fax. 0331-792287

web site: [www.futurashop.it](http://www.futurashop.it)

supporto tecnico: [www.futurashop.it/Assistenza-Tecnica](http://www.futurashop.it/Assistenza-Tecnica)