

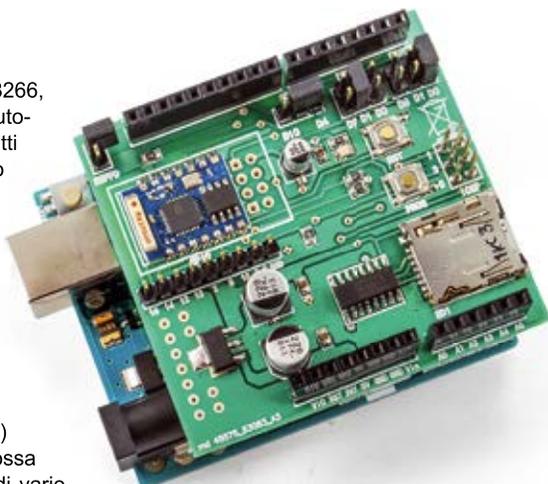
WiFi Shield con ESP8266

(cod. FT1192M)

Lo shield è basato sull'integrato ESP8266, affinché possa funzionare in modo autonomo all'interno di una rete Wi-Fi. Infatti l'ESP8266 contiene non solo lo stadio RF, ma anche un microcontrollore pienamente programmabile ed un registro cui fanno capo degli I/O, oltre a bus di comunicazione come l'SPI e l'I²C. Questo significa che il componente può da solo implementare la comunicazione in WiFi; inoltre, la possibilità di far eseguire al chip un programma, permette la gestione di alcuni pin (GPIO, bus vari) al fine di realizzare un sistema che possa interfacciarsi con periferiche esterne di vario genere o gestire ingressi e uscite. All'interno del chip ESP8266 troviamo un microcontrollore a 32 bit completo di ROM, RAM e SRAM per programma e dati; il chip dispone di linee di I/O digitali ed ingressi analogici, oltre che di porte di comunicazione come I²C, SPI e UART. Le porte di comunicazione possono funzionare da slave in modo da interfacciare memorie esterne al modulo ed ampliarne il funzionamento. Le linee digitali disponibili sono sedici, configurabili come ingressi o uscite, tutte con resistenza di pull-up e con la possibilità di generare un *interrupt*. Sono disponibili anche alcune linee di ingresso e di uscita analogiche basate su un convertitore sigma-delta a PWM. Il chip è stato progettato per funzionare ad una tensione di 3,3V con un massimo ammissibile di 3,6V, gli ingressi digitali dispongono di un apposito circuito (overvoltage protection) per proteggere l'integrato da tensioni in ingresso superiori a 6V.

Supporta tutti i protocolli di cifratura più moderni: WEP (RC4), CCMP (CBC-MAC, counting mode), TKIP (MIC, RC4) o WAPI (SMS4), WEP (RC4), CRC, WPA, WPA2 e WPS.

Nella **Tabella 1** viene riportato il funzionamen-



to dei singoli pin. Le applicazioni cui il componente è destinato sono home automation, reti mesh, wireless industriale, baby monitor, telecamere di rete wireless, reti di sensori, elettronica indossabile, dispositivi di localizzazione wireless location-aware devices, security ID tag ecc.

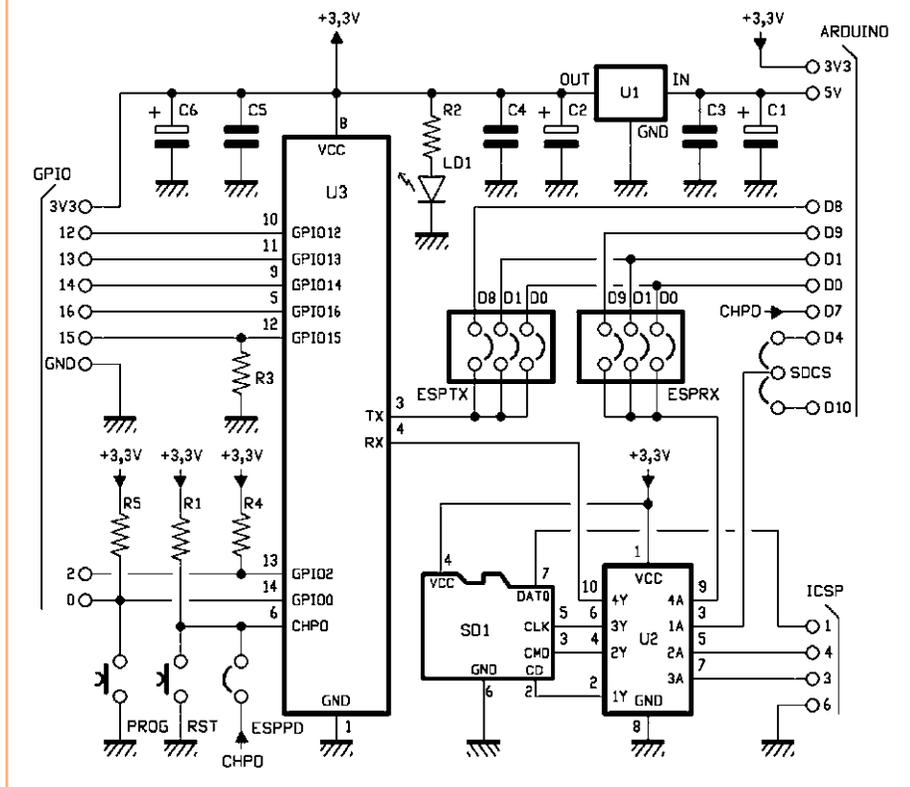
Le caratteristiche e le funzioni implementate sono:

- compatibilità 802.11 b/g/n;
- WiFi Direct (P2P), soft-AP;
- supporto stack TCP/IP;
- TR switch, balun, LNA, amplificatore RF e accoppiamento alla rete integrati;
- PLL e gestore dell'alimentazione integrati;

Tabella 1 - Funzione dei pin del chip ESP8266.

GPIO	INPUT	OUTPUT	PWM
GPIO0	sì	no	no
GPIO2	sì	no	no
GPIO12	sì	sì	sì
GPIO13	sì	sì	sì
GPIO14	sì	sì	no
GPIO15	sì	sì	sì

Schema elettrico



- potenza in trasmissione in 802.11b di +19,5 dBm;
- sensore termico integrato;
- supporto per antenna diversity;
- CPU a 32 bit;
- SDIO 2.0, SPI, UART;
- STBC, 1x1 MIMO, 2x1 MIMO;
- A-MPDU, A-MSDU;
- solo 2 ms per connessione e trasferimento di pacchetti dati;
- corrente assorbita in stato di off minore di 10 µA;
- consumo in standby minore di 1 mW (DTIM3).

L'ESP8266 nasce per fornire la connettività WiFi

in dispositivi mobili o indossabili alimentati a batteria, implementa modalità a basso consumo di energia che sono: *active mode*, *sleep mode* e *deep sleep mode*, ognuna delle quali prevede l'attivazione di un certo numero di periferiche. Nella *sleep mode* il modulo radio è spento mentre rimangono attivi il timer e il watchdog e l'assorbimento è di soli 12 µA; in questa modalità è possibile programmare il timer interno affinché attivi il modulo radio a intervalli regolari per inviare o ricevere dati. All'occorrenza è possibile attivare il modulo radio e abilitare le trasmissioni, tramite segnali dall'esterno. Da software è anche possibile ridurre la potenza di emissione e ridurre i con-

[piano di **MONTAGGIO**]

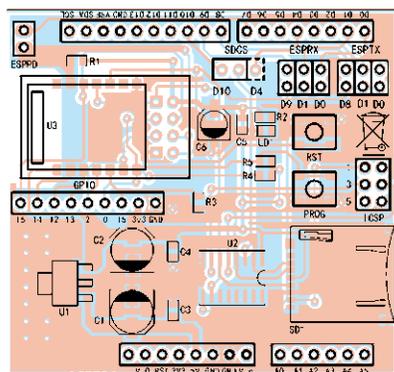
Elenco Componenti:

R1: 10 kohm (0805)
 R2: 330 ohm (0805)
 R3÷R5: 10 kohm (0805)
 C1, C2: 220 μ F 6,3 VL
 elettrolitico (\varnothing 6mm)
 C3÷C5: 100 nF ceramico (0805)
 C4, C5: 100 nF ceramico
 (0805)
 C6: C6: 22 μ F 6,3 VL
 elettrolitico (\varnothing 4mm)
 U1: TC1262-3.3VDB
 U2: 74HC4050D
 U3: ESP03
 RST: Microswitch

PROG: Microswitch
 SDA: Connettore micro SD-Card
 LD1: LED verde (0805)

Varie:

- Strip maschio 2 vie
- Strip maschio 3 vie (5 pz.)
- Jumper (4 pz.)
- Strip maschio 9 vie
- Strip Maschio/Femmina 3 vie (2 pz.)
- Strip Maschio/Femmina 6 vie
- Strip Maschio/Femmina 8 vie (2 pz.)
- Strip Maschio/Femmina 10 vie (1 pz.)
- Circuito stampato S1192



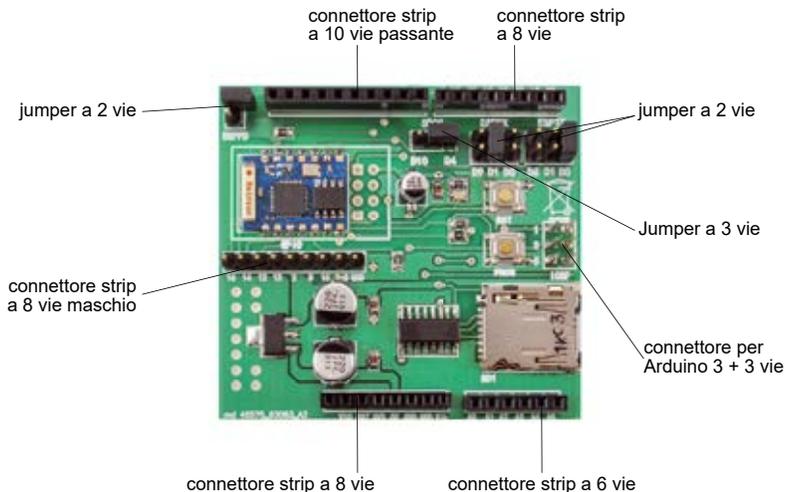
sumi in applicazioni in cui sono richieste comunicazioni a brevi distanze come in dispositivi indossabili. La linea CHIP_PD disponibile sul connettore esterno permette di portare il modulo nello stato di deep sleep mode con un assorbimento di corrente quasi zero.

L'assorbimento di corrente durante l'active mode raggiunge un massimo di 215 mA alla massima potenza di trasmissione con protocollo 802.11b a 11 Mbps e scende a 145 mA con protocollo 802.11g a 54 Mbps (diviene soli 60 mA per l'invio di un pacchetto di 1.024 byte con potenza in antenna di -65dBm). Esistono

vari moduli WiFi basati sull'ESP8266, che si distinguono per form-factor; in questo shield prevediamo l'ESP03, ma è possibile montarvi anche l'ESP01.

Montaggio connettori e jumper.

Lo shield viene fornito già assemblato. Le uniche parti da montare sono i jumper e i connettori. Saldare i 6 jumper a 2 vie in corrispondenza di D9-D1-D0 e D8-D1-D0 e il jumper a 3 vie in corrispondenza di D10-D4 come indicato sul PCB. Saldare poi il modulo ESP e i connettori strip a 10, 8, 6 vie nelle posizioni corrispondenti. Per il



connettore di Arduino procedere come indicato di seguito: inserire ora il connettore 3+3 vie sui pin ICSP di Arduino. Posizionare lo shield sopra la board Arduino facendo in modo che si inserisca perfettamente nel connettore ICSP; a questo punto saldare i pin del connettore.

Utilizzo pratico

La gestione del chip ESP8266 avviene con semplici comandi AT da inviare serialmente; anche l'invio e la ricezione dei dati avviene serialmente. Occorre però porre attenzione alla versione del modulo in possesso, in quanto i primi moduli commercializzati con firmware versione 080 comunicavano alla velocità di 115 kbps, mentre quelli più recenti (con firmware versione 09xx) funzionano a 9.600 bps. La possibilità di comunicare a una velocità di 9.600bps permette di gestire il modulo con la seriale software, facilmente implementabile

con l'apposita libreria disponibile nell'IDE di Arduino, lasciando libera la seriale hardware di comunicare con il PC, permettendo così una facile funzione di Debug.

Per questi primi esempi faremo in modo di poter comunicare con il chip ESP direttamente dall'applicativo Serial Monitor di Arduino, lasciando inutilizzato il microcontrollore ATMEL328 della scheda.

Caricare su Arduino uno sketch che non usi la seriale, *Blink.ino* va benissimo, e impostare i jumper in modo che il modulo possa comunicare con il PC (riga 3 della **Tabella 2**).

Aprire Serial Monitor di Arduino ed impostare una comunicazione a 9.600bps attivando sia il fine linea che il ritorno carrello (sia NL che CR) quindi inviare la stringa AT ed attendete risposta; se tutto è a posto il modulo risponde con "OK". Questo primo invio non fa assolutamente nulla ma permette di capire se il modulo è atti-

Tabella 2 - Disposizione dei jumper per la comunicazione.

COMUNICAZIONE DI ESP8266	JUMPER
seriale software pin 8 e 9 di Arduino	ESPTX=D8 e ESPRX=D9
UART di Arduino	ESPTX=D0 e ESPRX=D1
Comunicazione con PC tramite convertitore USB/seriale di Arduino	ESPTX=D1 e ESPRX=D0

Tabella 3 - Collegamento hardware dello shield.

PIN	FUNZIONE	PIN ARDUINO
SD_CS	SD chip select	D4 o D10
SD_MOSI	SD MOSI	D11
SD_MISO	SD MISO	D12
SD_CLK	SD clock	D13
ESPTX	ESP-8266 linea TX	D0 (RX0) o D1 (TX1) o D8
ESPRX	ESP-8266 linea RX	D0 (RX0) o D1 (TX1) o D9
ESPPD	ESP-8266 chip select (power down control)	3V3 o D7

vo e funzionante; all'accensione il modulo invia comunque la stringa "ready...".

Inviando la stringa AT+RST si esegue un reset software del dispositivo, il quale risponderà con la stringa "OK" seguita dal nome del fabbricante ed alcuni dati interni al chip. La stringa AT+GMR permette di conoscere la versione del firmware installato (Fig. 4). Con questa configurazione il modulo ESP8266 potrà comunicare con qualsiasi applicativo del PC come Processing o qualsiasi monitor seriale.

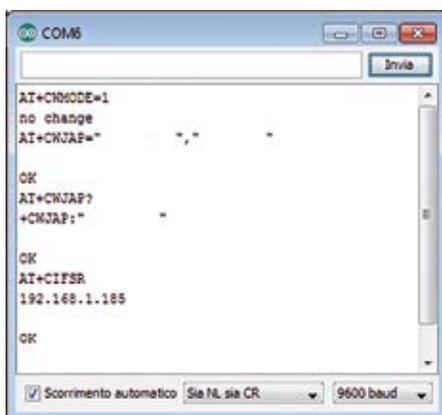
Possiamo sempre più prendere confidenza con il modulo provando a connetterci alla rete Wi-Fi di casa; il comando AT+CWMODE=1 permette di impostare l'accesso ad un router, mentre il comando AT+CWJAP="nomerete","password" ne permette l'accesso. Per sapere se la connessione è riuscita possiamo chiedere al mo-

dulo lo stato della connessione con il comando AT+CWJAP?, mentre per conoscere l'indirizzo IP assegnato dal router il comando è il seguente: AT+CIFSR.

Non appena connessi alla rete, è possibile testare alcune funzionalità del modulo facendolo ad esempio funzionare da Server nei confronti del PC; per fare questo è necessario avviare un semplicissimo client sul PC utilizzando ad esempio un software portabile di nome Socket-Test, una specie di monitor seriale ma per reti ethernet. Invieremo in sequenza i comandi:

```
AT+CWJAP="nomerete", "password"
AT+CIPSTART="TCP", "192.168.1.128", 9999
AT+CIPSEND=4
```

inviare una stringa di 4 caratteri come ad esem-



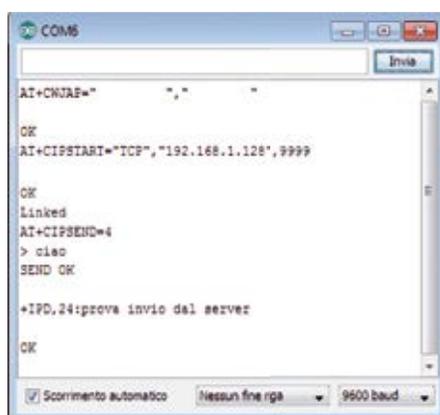
```
COM5
AT+GMR
no change
AT+GMR=" " " "

OK
AT+GMR?
+GMR:" "

OK
AT+CIFSR
192.168.1.128

OK
```

Serial Monitor per connessione alla rete Wi-Fi di casa.



```
COM5
AT+CWJAP=" " " "

OK
AT+CIPSTART="TCP", "192.168.1.128", 9999

OK
Linked
AT+CIPSEND=4
> ciao
SEND OK

+IPD,24:prova invio dal server

OK
```

Serial Monitor per il test come client.

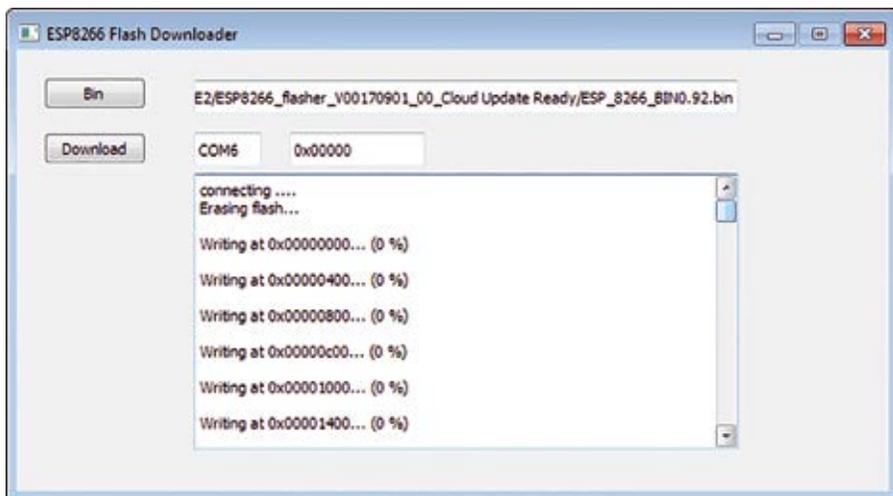


Fig. 3 - Installazione nuovo firmware.

pio "ciao" disabilitando l'invio del fine linea ed il ritorno carrello.

La ricezione di una stringa dal server sarà segnalata dalla stringa "+IPD" seguita dal numero di caratteri ricevuti e quindi il testo ricevuto.

Nei vari listati disponibili, troverete sempre una prima parte relativa alla modalità di comunicazione con il modulo ESP ed alla modalità di *debug*, infatti in questi primi esempi è sempre molto comodo prevedere una modalità per visionare i dati in transito dal modulo. Collegando

il jumper ESPTX alla linea D8 e ESPRX alla linea D9 sarà possibile gestire il modulo utilizzando una seriale software, lasciando libero il modulo UART dell'ATmega328 di comunicare con il PC per scopi di debug e programmazione; questa modalità funziona solo con i moduli ESP che comunicano a 9.600bps, modalità che comunque presenta i suoi limiti qualora si comincino a gestire flussi dati importanti contenenti oltre un centinaio di caratteri, come ad esempio avviene con la gestione di pagine web.

Tutti gli esempi proposti devono essere personalizzati introducendo il nome della vostra rete Wi-Fi e la password nei rispettivi campi SSID e PASS.

È stata prevista, oltre alla classica applicazione come Web Server, anche un'applicazione come Web Client, per dimostrare la facilità con la quale possono essere richieste informazioni ai siti internet. Gli unici dati da fornire sono l'indirizzo del server ed il nome della pagina alla quale accedere; nel listato scaricabile dal nostro sito trovate (come commenti) diversi esempi di impostazioni che abbiamo verifica-

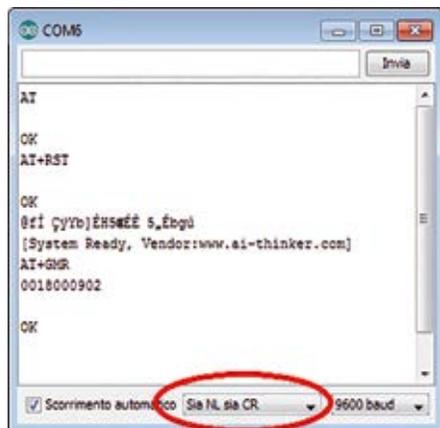
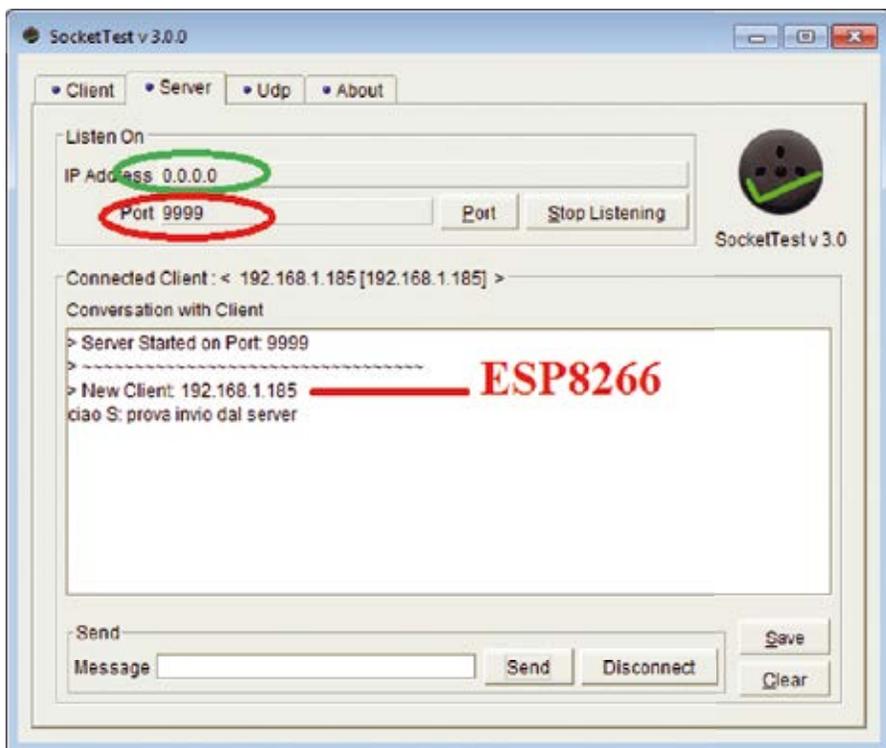


Fig. 4 - Serial Monitor delle comunicazioni con ESP.

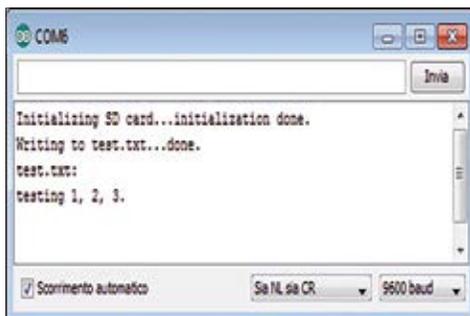


Server con ScketTest.

to personalmente, tra i quali l'interrogazione del server di Google oppure la lettura dei dati meteo dal server openweathermap. È possibile conoscere le condizioni meteo interrogando la più vicina stazione meteo senza la necessità di utilizzare alcun sensore. Esiste anche la possibilità di richiedere le previsioni meteo per i prossimi giorni; l'unico dato da inserire è il codice della città di cui si vogliono i dati ed è reperibile a questo indirizzo: http://openweathermap.org/help/city_list.txt. Similmente, è possibile interrogare il server di Yahoo per conoscere la data e l'ora attuale.

Negli esempi che proponiamo è stata utilizzata la sintassi classica html che è composta da tre sezioni: request (riga di richiesta), body (corpo del messaggio) e header (informazioni

aggiuntive). La sezione request a sua volta si compone di tre campi: il primo è il metodo di accesso e può essere di tipo GET (recupero



Web server con ESP8266.

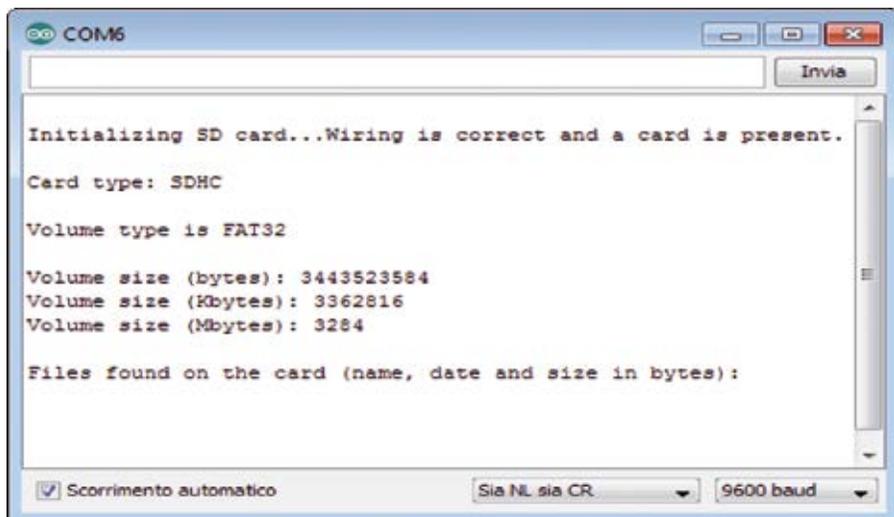


Fig. 5 - Test SD

informazioni), POST (invio informazioni), HEADER (come GET, ma riceve solo l'intestazione della pagina); il secondo è l'URI (Uniform Resource Identifier) cioè l'indirizzo della pagina cui si vuole accedere e il terzo è la versione del protocollo (nel nostro caso HTTP/1.1). La sezione body contiene semplicemente la richiesta vera e propria, mentre la sezione header contiene nel nostro caso l'header "HOST:".

Ecco che ad esempio la richiesta delle condizioni meteo può essere fatta inviando la richiesta: "GET /data/2.5/weather?id=3164699 HTTP/1.1 HOST: api.openweathermap.org" La richiesta deve essere inviata dopo che, ovviamente, ci saremo connessi al server con l'istruzione: "AT+CIPSTART = \"TCP\", \"DEST_HOST\" , \"DEST_PORT\" \r\n". Tutti gli esempi sono idonei con moduli aventi

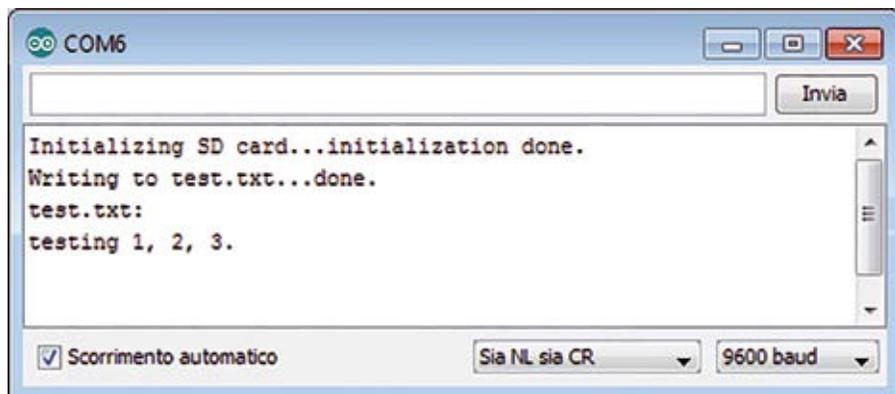


Fig. 6 - Test scrittura/lettura su SD.

```

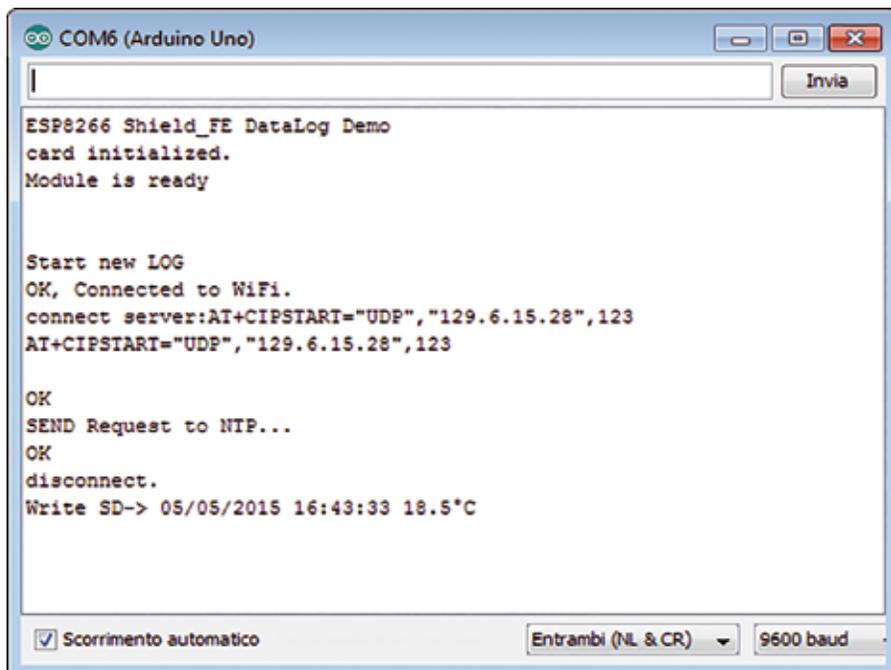
COM6
[Invia]
AT+RS1
OK
AT+CWMODE=1
AT+CWOAP
J' Èu ,u"  Ä  Ä" ÄAT+CWJAP="  ", "  "
AT+CIPMUX=0
OK
AT+CIPSTART="UDP", "129.6.15.28", 123
AT+CIPSTART="UDP", "129.6.15.28", 123
OK
AT+CIPSEND=48
ATE0
OK
>
Send Packet to NTP:ä i 1N14
response from ESP8266:
+++++
SEND OK
+IPD,48:ä ä ACTS@'@fOP_ @'@eÄi;X@'@eÄfcd
OK
*****
13
Seconds since Jan 1 1900 = 3635691689
Unix time = 1426702889
20:21:29
18/03/2015
[Scorrimto automatico] [Sia NL sia CR] [9600 baud]

```

firmware versione 09.02, che è la più stabile ed esente da *bug*; attualmente non esiste ancora una versione definitiva del firmware ed in rete si possono trovare versioni anche più recenti ma ancora tutte da testare. È comunque possibile installare una versione differente del firmware con una semplice procedura che ora descrive-

remo. Tra i file troverete una cartella di nome *Firmware*, che contiene un eseguibile di nome *ESP8266_flasher.exe*, utilizzato per caricare un nuovo firmware sul modulo; abbiamo messo a disposizione anche il firmware di versione 09.02 (Fig. 3).

Una volta avviato il software (non richiede in-



```
COM6 (Arduino Uno)
ESP8266 Shield_FE DataLog Demo
card initialized.
Module is ready

Start new LOG
OK, Connected to WiFi.
connect server:AT+CIPSTART="UDP", "129.6.15.28", 123
AT+CIPSTART="UDP", "129.6.15.28", 123

OK
SEND Request to NTP...
OK
disconnect.
Write SD-> 05/05/2015 16:43:33 18.5°C
```

Fig. 7 - Report relativo allo sketch.

stallazione) è sufficiente specificare la porta COM utilizzata per la comunicazione con il modulo ed aprire il file .bin, per sapere quale seriale utilizzare potete usare le informazioni fornite dall'IDE di Arduino quando è connessa la scheda Arduino.

Ponete il modulo nella modalità di comunicazione con il PC ed abilitate la modalità di boot mode del modulo ponendo la linea GPIO0 a massa, premendo e mantenendo premuto il pulsante PROG, dopodiché riavviare il modulo ponendo per un attimo la linea CHPD a livello basso tramite il pulsante RST. Dopo il riavvio, il modulo si pone in modalità di boot mode, pronto a ricevere il nuovo firmware, cliccate quindi sul pulsante Download per attivare il caricamento. Appena avviato il download del firmware, potete rilasciare il pulsante PROG e attendere la conclusione dell'operazione, che richiede in tutto alcuni minuti. Utilizzate il comando AT+GMR per verificare se il modulo

funziona correttamente e quale versione del firmware risulta installata.

La disponibilità di una SD per salvare dei dati ci permette di realizzare un interessante data logger che dispone della funzionalità di accesso alla rete Internet per reperire informazioni utili. Nel caso di un data logger un dato sicuramente importante è la data e l'ora in cui avviene il salvataggio dei dati, funzione di solito svolta da un RTC (Real Time Clock) interfacciato con Arduino. Disponendo di un accesso ad Internet, è semplice interrogare un server NTP per conoscere data e ora precisi senza bisogno di eseguire alcuna impostazione iniziale. Una prima semplice prova per verificare se tutto funziona correttamente consiste nell'inserire nello slot una SD (micro SD per la precisione) ed eseguire lo sketch *cardInfo.ino* disponibile negli esempi di sistema; assicuratevi che il jumper SDCS sia nella posizione D4 (Fig. 5). Potete vedere nelle immagini pubblicate la risposta

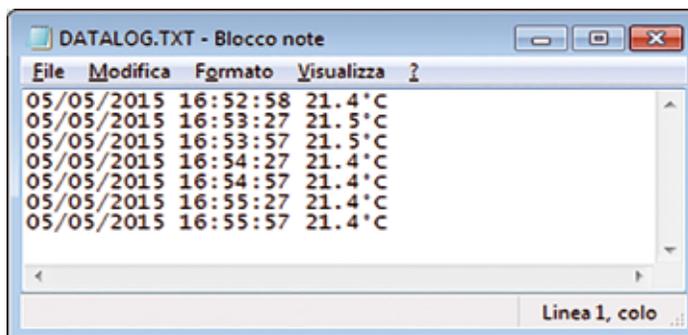


Fig. 8
Contenuto del file
di log dello sketch.

visualizzata su Serial Monitor relativa ad una SD HC cat4 da 8GB formattata FAT32 utilizzata per le nostre prove.

Con l'esempio *ReadWrite.ino* potete verificare se le funzioni di lettura funzionano correttamente (**Fig. 6**).

Vediamo ora come salvare il valore della temperatura ambiente misurata tramite un semplicissimo integrato come l'LM35. Lo sketch per la funzione di data logger si chiama *ESP8266_DataLogNTP.ino*. Per testare questo programma è necessario, per prima cosa, inserire il nome e la password della vostra rete Wi-Fi nei campi SSID e PASS. Il sensore di temperatura andrà alimentato da Arduino tramite i pin GND e +5V mentre la sua uscita va connessa all'ingresso analogico AN0. La modalità di default prevista consiste nel far comunicare il microcontrollore ATmega328 con il modulo Wi-Fi tramite una seriale software e di lasciare la seriale hardware per la funzione di debug. Nel caso vogliate gestire il modulo Wi-Fi con la seriale hardware è sufficiente porre i jumper in modo che i segnali ESPTX e ESPRX siano connessi rispettivamente a D0 e D1. Le seguenti righe di codice vengono utilizzate nello sketch per costruire la stringa che verrà scritta nella SD card:

```
String dataString = getDateString();
dataString += " ";
dataString += getTimeString();
dataString += " AN0=";
dataString += String(analogRead(A0));
```

Questa stringa potrà essere modificata secondo necessità aggiungendo o meno altri valori e altri separatori. Abbiamo previsto che il modulo rimanga acceso solo per il breve lasso di tempo necessario a connettersi alla rete e ricevere i dati dal server NTP, per tutto il resto del tempo tra un log ed il successivo il modulo è portato nello stato di power-down, grazie al segnale del pin D7 che comanda il pin CH_PD dell'integrato EPS8266.

Nella SD vi ritroverete un file di nome DATALOG.TXT contenente i valori acquisiti, come visibile nella **Fig. 8**. Questo sketch è compatibile con moduli in cui è installato il firmware versione 9.02 e 9.03 come quello montato nel lo shield.

L'articolo completo del progetto è stato pubblicato su: *Elettronica In n. 200*

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

supporto tecnico: www.futurashop.it/Assistenza-Tecnica