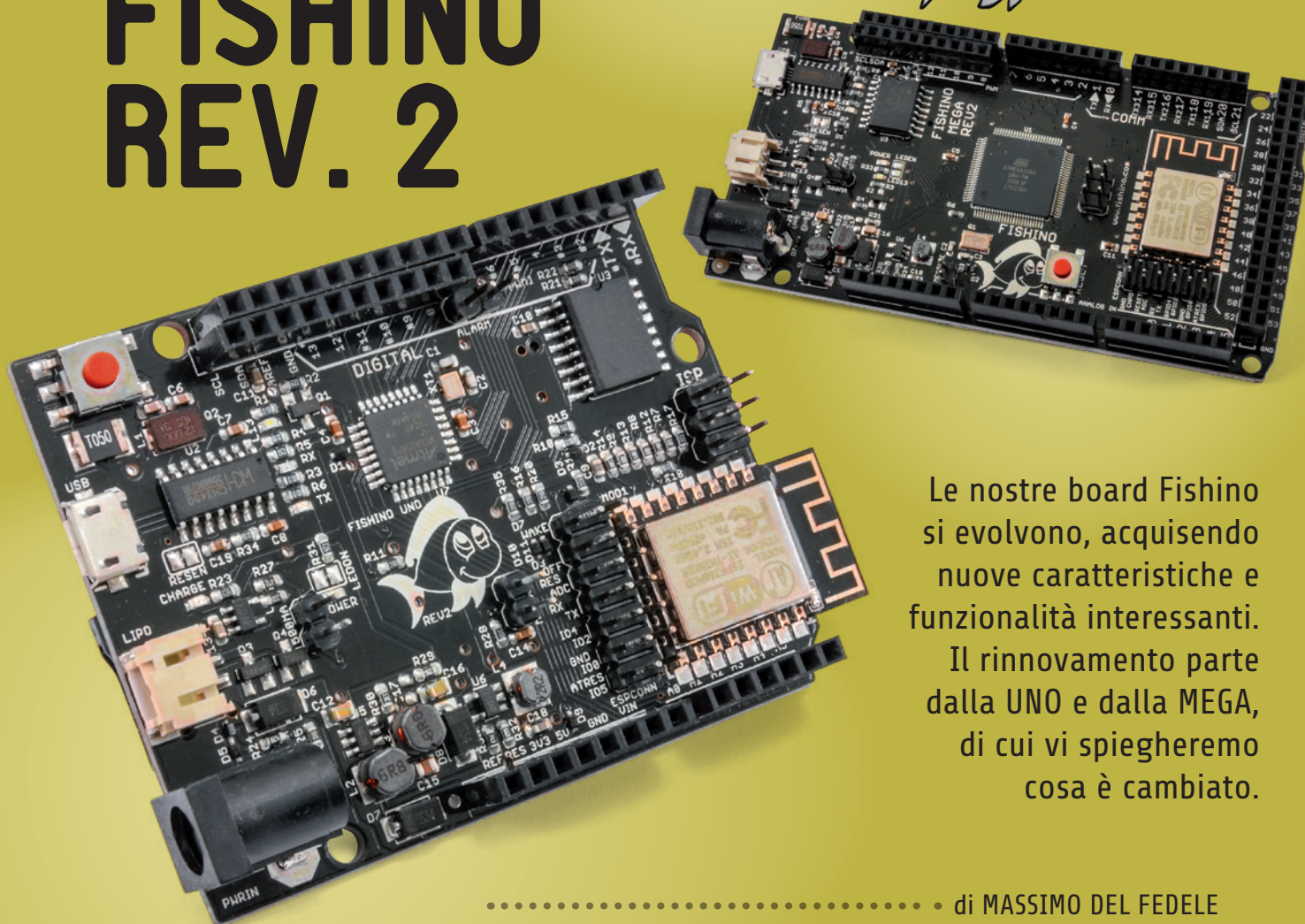


# ARRIVANO LE FISHINO REV. 2



Le nostre board Fishino si evolvono, acquisendo nuove caratteristiche e funzionalità interessanti. Il rinnovamento parte dalla UNO e dalla MEGA, di cui vi spiegheremo cosa è cambiato.

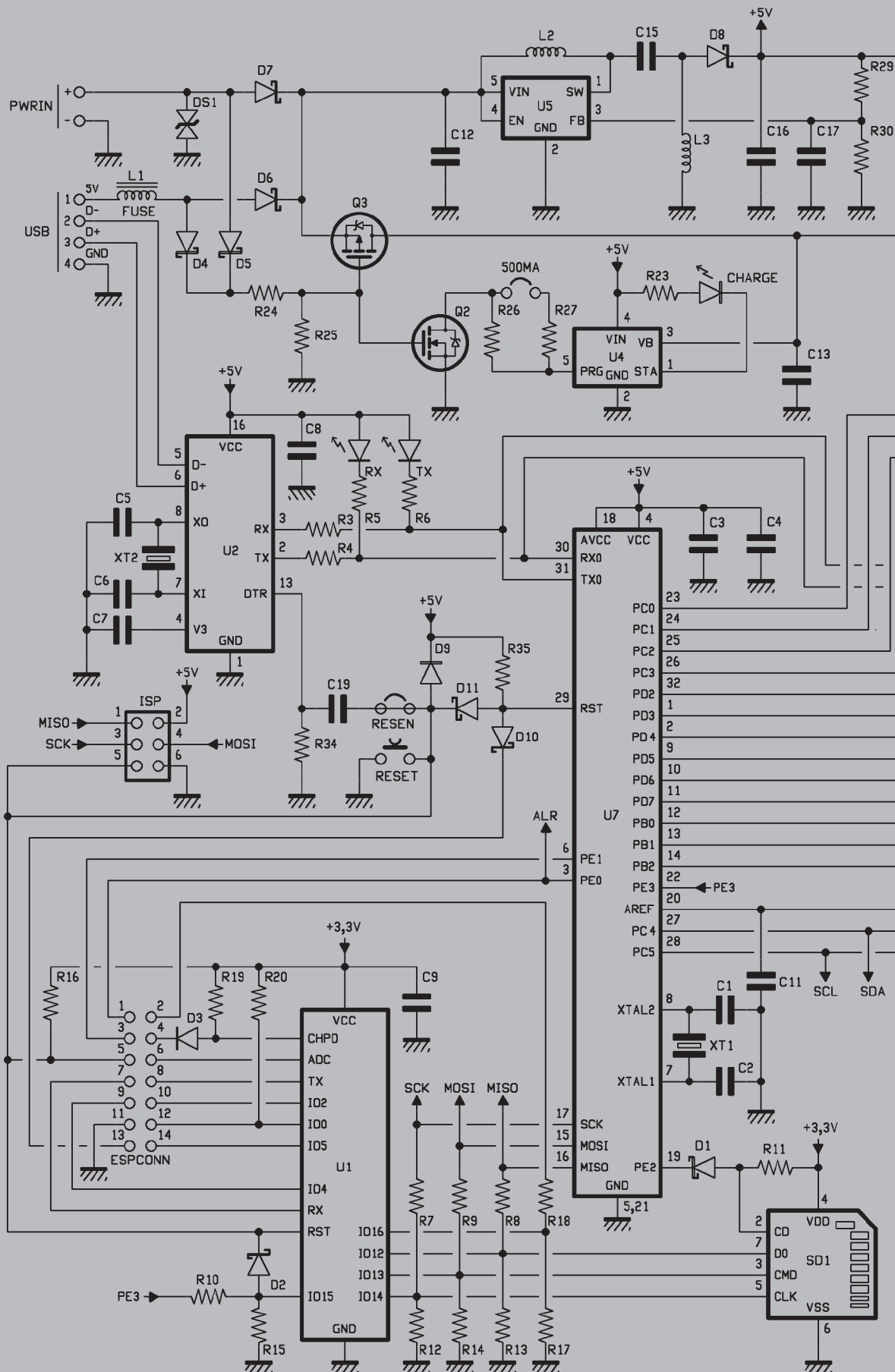
..... di MASSIMO DEL FEDELE

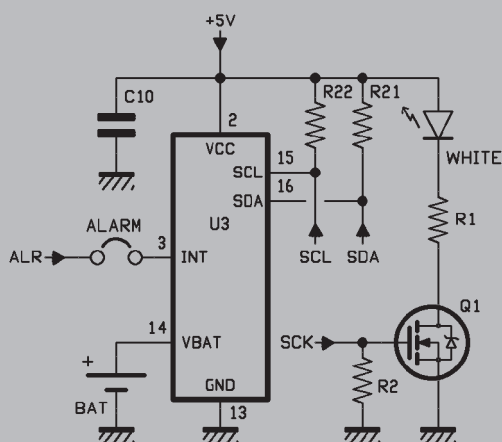
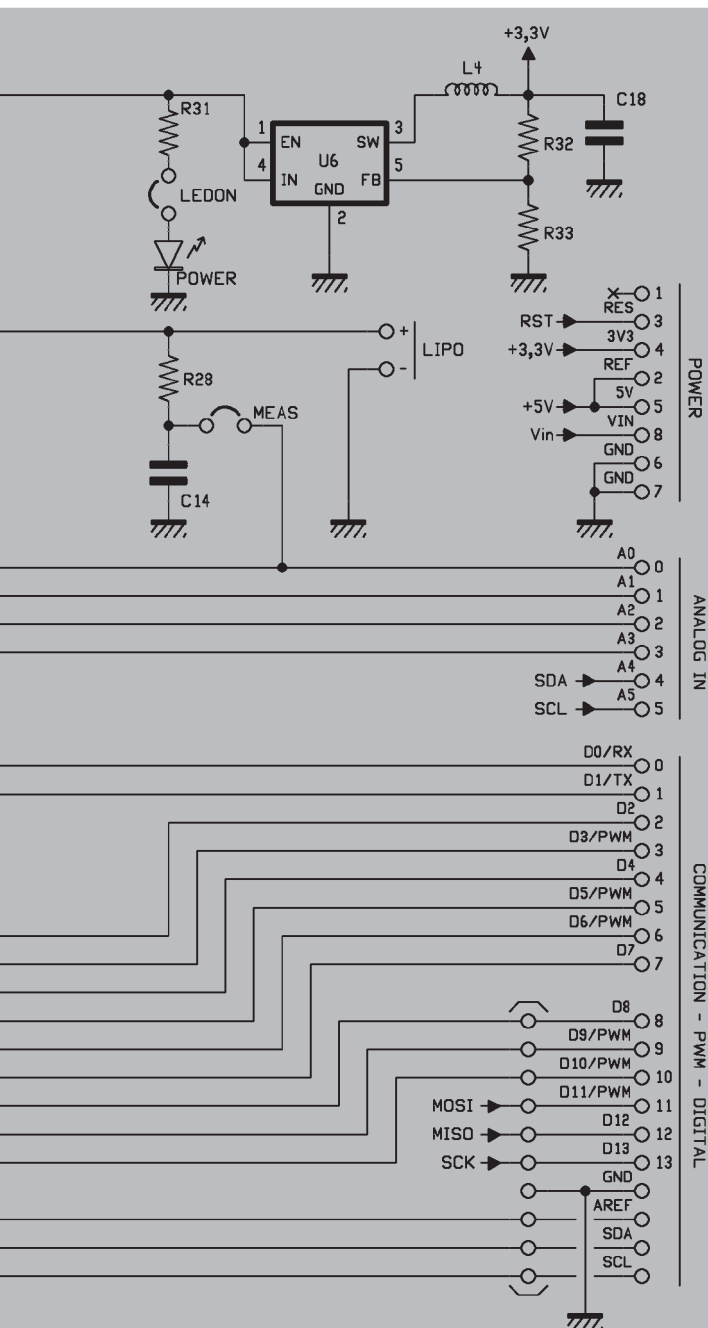
**I**n settembre 2015, abbiamo presentato quella che sarebbe diventata la capostipite di una serie di schede innovative compatibili con le famose Arduino: la Fishino UNO. Questa è stata seguita in breve tempo dalla Fishino Mega e, successivamente, dalla Guppy e da due schede a 32 bit, la Fishino32 e la Piranha. Quando abbiamo progettato la Fishino UNO e la Mega abbiamo voluto mantenere la compatibilità totale con le corrispondenti Arduino, utilizzando quindi gli stessi controller e, per le periferiche aggiuntive, i medesimi I/O usati dagli analoghi Shield disponibili sul mercato. Col passare del tempo i microcontrollori si evolvono e le nostre schede non possono rimanere

insensibili all'evoluzione, perciò abbiamo iniziato ad ammodernarle, dando origine alle versioni 2 (o REV 2, se preferite) della Fishino UNO e della Mega, con interessanti modifiche che ne migliorano ulteriormente l'utilizzo e risolvono alcuni problemi, principalmente relativi ad alcuni I/O utilizzati, appunto, dalle periferiche interne. Iniziamo quindi dalla Fishino UNO, che è stata completamente riprogettata, come avremo modo di esporvi in queste pagine.

## FISHINO UNO REV2

La nostra capostipite era rimasta l'unica non dotata di alimentazione switching e senza la possibilità





di essere alimentata tramite una batteria ricaricabile, quindi la prima cosa che abbiamo fatto è stata dotarla dello stesso alimentatore presente nella Mega, ovvero un doppio switching costituito da un convertitore SEPIC per ottenere i 5V da un ingresso che può andare dai 3,5 ai 25 volt, con un'efficienza di quasi il 90%, senza i problemi di surriscaldamento e i limiti di corrente di un regolatore lineare, seguito da un ulteriore convertitore buck sincrono per i 3,3 volt necessari al modulo WiFi e alla scheda MicroSD.

Lo schema elettrico visibile in queste pagine mostra tale sezione, composta da U4 per la ricarica della batteria e gli ingressi di alimentazione, che portano tre tensioni:

- LIPO, che fa capo al connettore per la batteria ricaricabile ai polimeri di litio;
- USB, che fa capo al connettore microUSB;
- PWRIN, che fa capo al plug di alimentazione.

Le due tensioni solitamente più elevate, ossia quella dell'USB e PWRIN vengono fatte passare attraverso i due diodi Schottky di potenza D6 e D7, mentre la tensione "più bassa", per motivi di efficienza, viene instradata verso il MOSFET a canale P siglato Q3. Novità della REV2 è la presenza del soppressore di sovratensioni DS1 sulla linea PWRIN.

D6 e D7, insieme al diodo di protezione del Q3, formano una porta OR di potenza, la quale permette di ottenere in uscita la tensione maggiore tra quelle presenti agli ingressi; supponiamo ad esempio di avere solo la tensione LIPO, di circa 3,7V: in questo caso il diodo interno al MOSFET risulta polarizzato direttamente portando verso il convertitore la medesima tensione, che non può ritornare verso gli altri due ingressi a causa della presenza dei diodi D6 e D7 polarizzati inversamente.

Ora, se colleghiamo il connettore microUSB, la tensione VUSB di 5 volt polarizza direttamente il diodo D6 e raggiunge il convertitore mentre il diodo interno al MOSFET si trova con una tensione di 4 volt all'anodo e di 5 volt al catodo e risulta quindi polarizzato inversamente, interdicendosi.

Se, infine, alimentiamo anche l'ingresso PWRIN con una tensione superiore ai 5 volt, ad esempio di 15 volt, il diodo D7 risulterà polarizzato direttamente, portando la tensione VIN al convertitore, mentre i diodi D6 e quello interno al MOSFET verranno polarizzati inversamente interdicendosi.

Lo stadio in questione raggiunge quindi l'obiettivo di effettuare una commutazione totalmente automatica della sorgente di alimentazione.

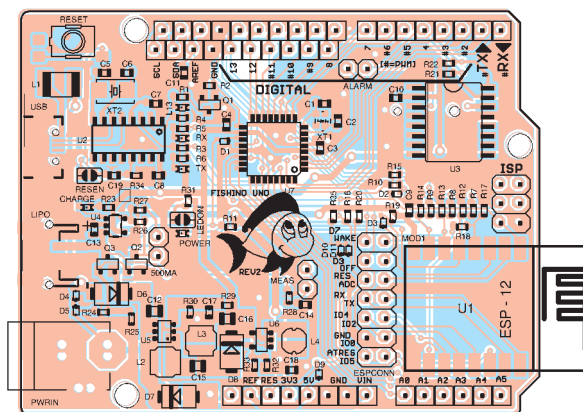
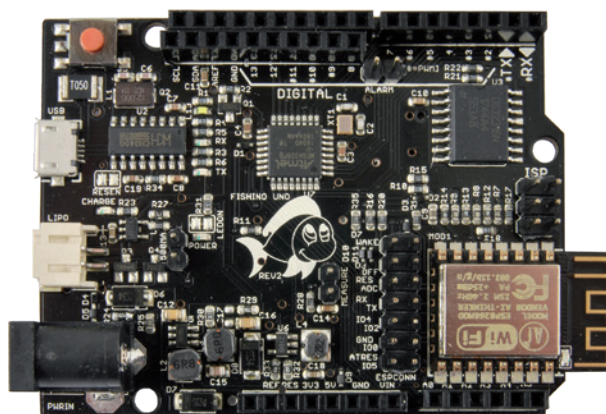
Torniamo ora al MOSFET Q3, che ci serve per



## Elenco Componenti:

R1, R3, R4: 1 kohm	C16: 22 $\mu$ F ceramico
R2: 220 kohm	C17: -
R5: 2,4 kohm 1%	C18: 10 $\mu$ F ceramico
R6: 1 kohm	C19: 1 $\mu$ F ceramico
R7÷R9: 680 ohm	CHARGE: LED rosso
R10, R11: 10 kohm	WHITE: LED bianco
R12÷R14: 1,5 kohm	RX: LED giallo
R15: 3,3 kohm	TX: LED blu
R16: 10 kohm	POWER: LED verde
R17: 3,3 kohm	D1÷D5: RB521S-30TE61
R18: 1 kohm	D6÷D8: SS34 DO-214AB
R19÷R22: 10 kohm	D9÷D11: RB521S-30TE61
R23: 470 ohm	DS1: SMBJ24CA
R24: 2,7 kohm	Q1, Q2: 2N7002
R25: 100 kohm	Q3: NTR4171P
R26: 10 kohm	U1: Modulo ESP8266
R27: 2,7 kohm	U2: CH340G SOT23-5
R28: 10 kohm	U3: DS3231 SOT23-6L
R29: 97,6 kohm 1%	U4: MCP73831T-2
R30: 13,3 kohm 1%	TQFP100
R31: 470 ohm	U5: SX1308
R32: 475 kohm	U6: LC3406
R33: 105 kohm	U7: ATMEGA328PB-AU
R34: 1 kohm	(MF1390)
R35: 10 kohm	XT1: Quarzo 16 MHz
C1, C2: 22 pF ceramico	XT2: Quarzo 12 MHz
C3, C4: 100 nF ceramico	L1: Fusibile 500mA (1210)
C5, C6: 22 pF ceramico	L2, L3: 6,8 $\mu$ H
C6: 22 pF ceramico	L4: 2,2 $\mu$ H
C7: 1 $\mu$ F ceramico	BAT: Porta Batteria
C8: 100 nF ceramico	CR1220 da CS
C9: 10 $\mu$ F ceramico	RESET: Microswitch
C10, C11: 100 nF ceramico	SD1: Connettore micro-SD
C12: 22 $\mu$ F 25 VL tantalio	USB: Connettore micro-USB
C13: 4,7 $\mu$ F ceramico	
C14: 100 nF ceramico	
C15: 4,7 $\mu$ F 25 VL tantalio	

Varie:  
- Plug alimentazione



- Connettore JST 2 vie passo 2 mm
- Strip maschio 2 x 3 vie
- Strip maschio 2 vie (3 pz.)
- Strip femmina 6 vie
- Strip femmina 8 vie (2 pz.)
- Strip femmina 10 vie (2 pz.)
- Batteria CR1220
- Circuito stampato S1390 (70 x 54 mm)

bypassare la pur piccola caduta di tensione dello Schottky, così da ridurre le perdite di potenza, che divengono rilevanti ad alte correnti.

La tensione selezionata dalla OR a diodi raggiunge i piedini 4 e 5 dell'U5, che è la base del convertitore SEPIC, il cui vantaggio è principalmente la possibilità di ottenere in uscita una tensione sia inferiore che superiore a quella in ingresso, unendo così i vantaggi di un convertitore Buck e di un Boost, ma utilizzando un solo integrato switching.

La tensione in uscita è regolata dal partitore costituito da R29 ed R30, che forniscono una tensione di riferimento di 0,6 volt all'ingresso FB dell'integrato quando in uscita sono presenti 5 volt.

All'uscita del convertitore SEPIC troviamo quindi una tensione continua di 5 volt. Il LED verde POWER funge da spia di accensione ed è disattivabile tagliando il ponticello SMD LEDON, in modo da ridurre i consumi al limite se si vuole alimentare il Fishino MEGA a batteria. La tensione di 5 volt in uscita dal SEPIC entra in un ulteriore convertitore switching, questa volta un semplice convertitore Buck (o step-down, ossia abbassatore) operante a una frequenza superiore al MHz che, per motivi di efficienza, è stato realizzato tramite un convertitore sincrono, ovvero dotato all'interno di un secondo MOSFET al posto del diodo Schottky utilizzato normalmente. Da questo convertitore escono i 3,3 volt

necessari ad alimentare il modulo WiFi ed il lettore di microSD. La ricarica della batteria è affidata all'U4, il solito MCP73831 nella classica configurazione occorrente a mantenere carica una cella ai polimeri di litio. Il piedino PROG viene utilizzato per impostare la corrente di carica, che dipende dalla resistenza collegata fra PRG e massa; nel nostro circuito può essere scelta tra due valori: circa 100 mA con il ponticello siglato 500 MA aperto (viene inserita la sola resistenza R9 da 10 kohm) oppure circa 500 mA con il ponticello chiuso.

Il LED CHARGE indica lo stato di carica della batteria, anche se va notato che a causa dell'imprecisione del rilevatore interno, soprattutto a basse tensioni di alimentazione, può far accendere il LED anche a batteria carica (mentre deve restare illuminato solo durante la carica); non ve ne preoccupate perché è tutto normale.

Notate che utilizziamo il MOSFET Q2 per chiudere o lasciare aperto il pin PRG dell'U4, in modo da disattivare il caricabatterie quando il circuito non è alimentato né dall'USB, né dal jack PWRIN. Il solito ponticello MEAS permette di collegare l'ingresso analogico ADC0 del microcontrollore alla batteria se ci serve verificare lo stato della carica.

Lasciamo lo stadio d'alimentazione e passiamo al microcontrollore, che qui, nella Fishino UNO REV2, non è più l'ATmega328pb-au, ma la sua evoluzione ATmega328pb-an. Quest'ultimo, pur restando compatibile al 100% con il predecessore, dispone di alcune periferiche ed I/O aggiuntivi da noi prontamente sfruttati per gestire il modulo WiFi, la scheda microSD ed altro, senza impegnare gli I/O disponibili sugli header esterni. Grazie a questa scelta abbiamo potuto liberare gli I/O 4, 7 e 10, destinati in precedenza alle suddette periferiche, evitando quindi qualsiasi interferenza con shield esterni; abbiamo inoltre sfruttato un paio di I/O interni per nuove funzioni, utilizzabili in alternativa l'una con l'altra:

- gestione allarme dall'RTC;
- spegnimento del modulo WiFi e risveglio del microcontroller;
- porta I<sup>2</sup>C aggiuntiva.

Prima di vedere come sfruttare queste funzionalità, facciamo una premessa: per utilizzare la REV2 delle nostre board è indispensabile installare il package **Fishino** nell'IDE di Arduino; non facendolo, la scheda è comunque utilizzabile come una Arduino UNO, ma non sarà possibile sfruttare le periferiche incorporate quali il modulo WiFi e la scheda microSD, che la UNO non ha e che utilizzano le porte aggiuntive del microcontrollore.

### *Gestione periferiche interne (WiFi e microSD)*

Come accennato, per la gestione di queste periferiche vengono ora utilizzati alcuni I/O interni, liberando quindi i pin 4, 7 e 10 utilizzati in precedenza, che vengono quindi resi disponibili per qualsiasi altro uso. Restano invece impegnati i 3 I/O destinati all'interfaccia SPI che sono comunque utilizzabili in contemporanea da altre periferiche SPI.

L'unico "inconveniente", se così si può chiamare, nell'utilizzo degli I/O interni aggiuntivi, è che i pins di selezione del WiFi (WIFICS, prima pin 10) e della microSD (SDCS, prima pin 4) cambiano e diventano, rispettivamente, gli I/O 23 e 22.

Negli sketch quindi non bisogna più utilizzare i numeri 10 e 4 ma le costanti WIFICS e SDCS, già definite correttamente nelle varie librerie.

Ad esempio, l'istruzione per l'inizializzazione della scheda SD passa dalla seguente:

```
card.init(SPI_FULL_SPEED, 4);
```

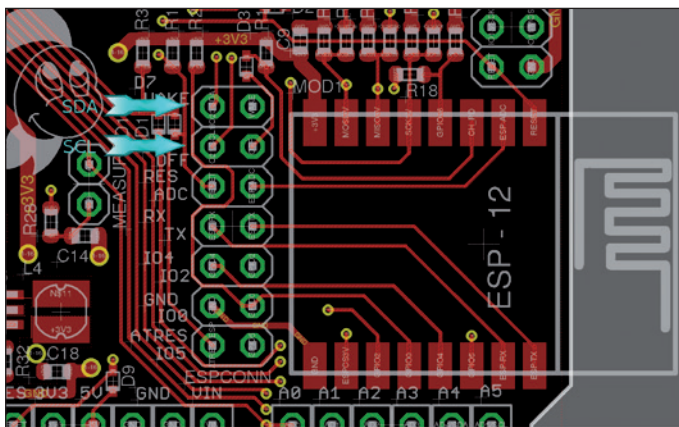
a questa:

```
card.init(SPI_FULL_SPEED, SDCS);
```

Il modulo WiFi viene gestito, invece, direttamente dalla libreria Fishino, quindi non occorre modificare nulla in quel caso.

### *Interfacce SPI, I<sup>2</sup>C e USB*

La sezione SPI è la stessa della Fishino UNO e serve ad adattare i livelli delle logiche a 5 volt dell'ATmega con quelle a 3,3 volt del modulo WiFi e della scheda microSD. L'adattamento viene realizzato semplicemente tramite partitori resistivi (resistenze da R7 a R17) nella direzione 5V -> 3,3V, mentre nella direzione inversa viene sfruttato il fatto che le logiche a 5 volt accettano come segnali alti valori ben inferiori a 3 volt, risultando quindi compatibili con le logiche a 3,3 volt. Il segnale MISO (Master In Slave Out) in teoria non richiederebbe un adattamento, visto che la direzione va dalla logica a 3,3 volt verso quella a 5 volt; l'abbiamo inserito comunque in previsione della futura estensione del firmware del modulo WiFi che permetterà il caricamento degli sketch attraverso il medesimo, cosa che richiede uno scambio di ruoli, diventando in quel caso il modulo WiFi il master e l'ATmega lo slave. L'interfaccia con la MicroSD funziona attraverso le linee SPI tra cui MOSI (dati dall'ATmega verso la SD, Master Out Slave In), MISO (dati dalla SD all'ATmega, Master In Slave Out) e SCK (clock). I



**Fig. 1** - La PC aggiuntiva localizzata sul connettore ESPCONN.

livelli verso la scheda SD sono ridotti dagli adattatori di livello di cui al paragrafo precedente. La selezione della scheda avviene tramite la linea PE2 del microcontrollore, che viene adattata al livello di tensione richiesto dal contatto 2 del lettore SD tramite R1 e D1 (uno Schottky); la SD si attiva con il PE2 a livello basso, mentre con 1 logico o in three-state (alta impedenza) è disabilitata.

Novità della REV2 è l'aggiunta di una seconda interfaccia I<sup>2</sup>C-Bus che è disponibile utilizzando la periferica **Wire0** (al contrario di **Wire** che è quella relativa all'interfaccia I<sup>2</sup>C principale); i relativi I/O sono disponibili sul connettore ESPCONN, aggiunto nella REV2, come mostrato nella **Fig. 1**.

Quest'interfaccia utilizza gli I/O 20 e 21, sovrapposti ad altre funzioni (ingresso allarme e gestione Low Power), quindi va utilizzata in ALTERNATIVA a queste; se volete utilizzare la **Wire0** non potete utilizzare l'allarme RTC e la gestione di spegnimento del modulo WiFi.

La libreria **Wire** contenuta nel package **Fishino** è già predisposta per gestire la I<sup>2</sup>C aggiuntiva.

Fate attenzione che nella **Fishino UNO REV2** non sono presenti le due resistenze di pull-up necessarie all'interfaccia I<sup>2</sup>C; se sulle vostre periferiche non ci sono (solitamente sono presenti...) dovete collegare due resistori da 4,7 ÷ 10 kohm tra tali pin ed il positivo (+5V) di alimentazione.

Passiamo all'interfaccia USB, che è la stessa della **Fishino UNO** originaria e si sviluppa intorno all'integrato CH340G; il chip è stato scelto sia per motivi di economicità che di semplicità circuitale, a parità di prestazioni.

L'integrato fornisce in uscita tutti i segnali di un'interfaccia RS232 standard, dei quali utilizziamo solo quelli di trasmissione/ricezione dati (RX e TX) ed il segnale DTR utilizzato per il reset automatico in fase di programmazione, come nell'Arduino originale, cosa che permette il caricamento degli

sketch senza dover premere pulsanti o azionare interruttori.

### Sezione RTC

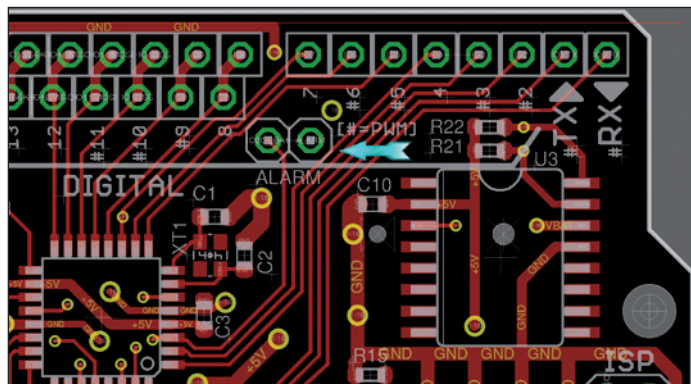
Anche il modulo RTC ha subito un'evoluzione; il relativo chip, un DS1307, è stato infatti sostituito con un DS3231, notevolmente più preciso, dotato di un'uscita di allarme e con il quarzo incorporato, il che permette una calibrazione precisa del medesimo in fase di produzione ed eventuali correzioni software per compensarne l'invecchiamento. L'uscita di allarme dell'RTC è collegabile all'ingresso 20 (uno degli I/O aggiuntivi) tramite un ponticello sulla scheda (**Fig. 2**) che permette l'attivazione dell'allarme RTC (nuova funzionalità della REV2). Inserendo questo ponticello all'ingresso digitale 20 (leggibile con un'istruzione **digitalRead(20)** o utilizzabile come ingresso di interrupt del tipo "pin-change") verrà generato un impulso ogniqualvolta l'allarme dell'RTC verrà attivato.

La libreria **FishinoRTC** è già predisposta per gestire questo ingresso e per correlare ad esso una funzione richiamata allo scadere dell'allarme; per dettagli potete leggere il file **FishinoRTC.h** della libreria stessa, in attesa che venga documentata.

L'ingresso di allarme è condiviso con le funzionalità Low Power (risparmio energetico), come spiegato nei paragrafi successivi ed è utilizzato anche dalla libreria **FishinoLowPower** per consentire al modulo WiFi di risvegliare la scheda dopo un periodo di sleep; quindi utilizzandolo per gestire l'allarme dell'RTC si dovrà rinunciare al risveglio dal low-power. L'ingresso è inoltre condiviso con l'interfaccia I<sup>2</sup>C aggiuntiva, come vedremo in seguito.

### Sezione WiFi

Il modulo WiFi è rimasto lo stesso della **Fishino UNO** originaria; la comunicazione con U7 avvie-



**Fig. 2** - Il jumper per la funzione allarme RTC.



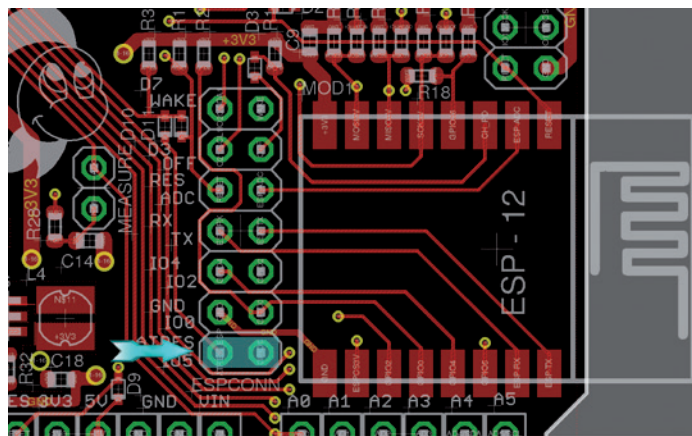
ne tramite interfaccia SPI e ricorrendo ad alcuni accorgimenti circuitali quali il diodo D2 utilizzato per forzare a livello basso il pin IO15 del modulo al reset, senza il quale il modulo stesso si avvierebbe nella modalità “caricamento da SD” che lo renderebbe inutilizzabile. Questo risulta necessario perché GPIO15 ha anche funzione di Slave Select (SS) del modulo e non può quindi essere collegata direttamente a massa.

Tutti i pin utili del modulo sono portati su un connettore (ESPCONN) come riportato qui di seguito.

- GPIO0: oltre ad essere utilizzabile come input/output digitale, serve per selezionare la modalità d'avvio al boot del modulo. Quest'ultimo può infatti essere avviato da Flash interna (funzionamento normale, GPIO0 a 1) o da interfaccia seriale, utilizzato per la riprogrammazione del firmware (GPIO0 a 0).
- GPIO2, GPIO4 e GPIO5 sono disponibili per l'uso come pin digitali, e sono sfruttabili tramite le apposite funzioni di libreria come fossero estensioni dei pin digitali di Arduino.  
Rx e Tx costituiscono la porta seriale hardware del modulo e sono utilizzati anche in fase di programmazione del firmware. Una prossima estensione del firmware ne permetterà l'uso come porta seriale aggiuntiva che consentirà a Fishino MEGA di avere un'ulteriore porta seriale.
- CH\_PD è il pin di abilitazione del modulo. Portandolo a livello alto il modulo risulta abilitato (impostazione predefinita), mentre un livello basso mette in standby l'ESP riducendone i consumi praticamente a zero.
- RESET è il reset hardware dell'ESP, attivo a livello basso.
- ADC è l'ingresso analogico dell'ESP, diretto verso un convertitore A/D da 10 bit (1.024 valori possibili).

La sezione di reset della Fishino UNO REV2 rimane quella della fishino UNO originaria che già conoscerete e prevede il reset simultaneo dell'U7 e del modulo WiFi U1 alla pressione del tasto di reset, all'avvio e alla richiesta di programmazione da parte dell'IDE; per poter eseguire la programmazione dell'Atmega tramite WiFi, il modulo ESP dev'essere in grado di resettare l'Atmega stesso senza a sua volta autoresettersi.

Iniziamo dal segnale DTR che esce dall'interfaccia USB/Seriale (U1/CH340G) e che, come anticipato, viene posto a livello basso quando la porta seriale viene aperta. Attraverso il condensatore C5 (1  $\mu$ F ceramico, contro i 100 nF dell'originale per allun-

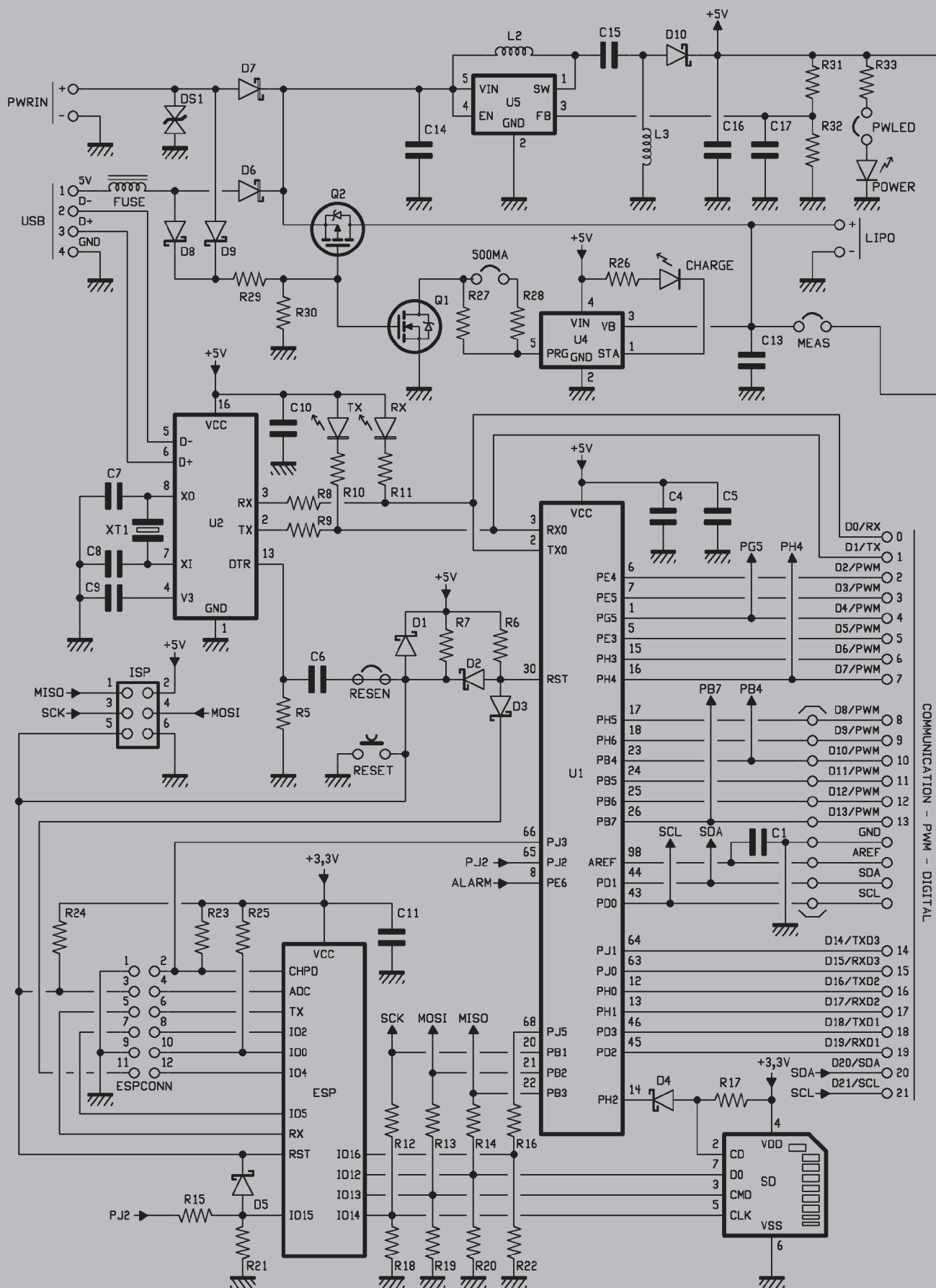


**Fig. 3 - Jumper per abilitare la funzione OTA.**

gare l'impulso di reset) viene generato un breve impulso che, passato attraverso il jumper SMD RESEN (tagliando il quale è possibile disattivare l'auto-reset), raggiunge la linea di “reset esterno”, alla quale sono connessi anche il pulsante di reset ed il pin 5 sul connettore di programmazione (ICSP). A differenza del circuito originale, nelle schede Fishino, tra la linea di RESET e il pin corrispondente dell'Atmega c'è il diodo D10, scopo del quale (e del diodo D10) è di resettare solo l'Atmega senza veicolare il segnale anche all'ESP. Tramite questo sistema abbiamo quindi dato la possibilità al modulo WiFi di controllare la linea di reset dell'Atmega che, unitamente all'interfaccia SPI, ne permette la riprogrammazione senza nemmeno la necessità di un bootloader precaricato. In pratica, una volta completato lo sviluppo nel firmware, sarà possibile non solo riprogrammare via WiFi l'Atmega, ma farlo utilizzando anche lo spazio normalmente riservato al bootloader. Come per la precedente versione è possibile caricare gli sketch tramite WiFi (aggiornamento firmware OTA), ma nella REV2 abbiamo riposizionato i connettori in modo da poter abilitare la cosa tramite un semplice ponticello, come vedete nella Fig. 3. Una volta installato questo ponticello ed attivata la modalità OTA durante l'aggiornamento del firmware potrete caricare comodamente gli sketch via WiFi utilizzando l'IDE di Arduino o il nostro ambiente di sviluppo FishIDE.

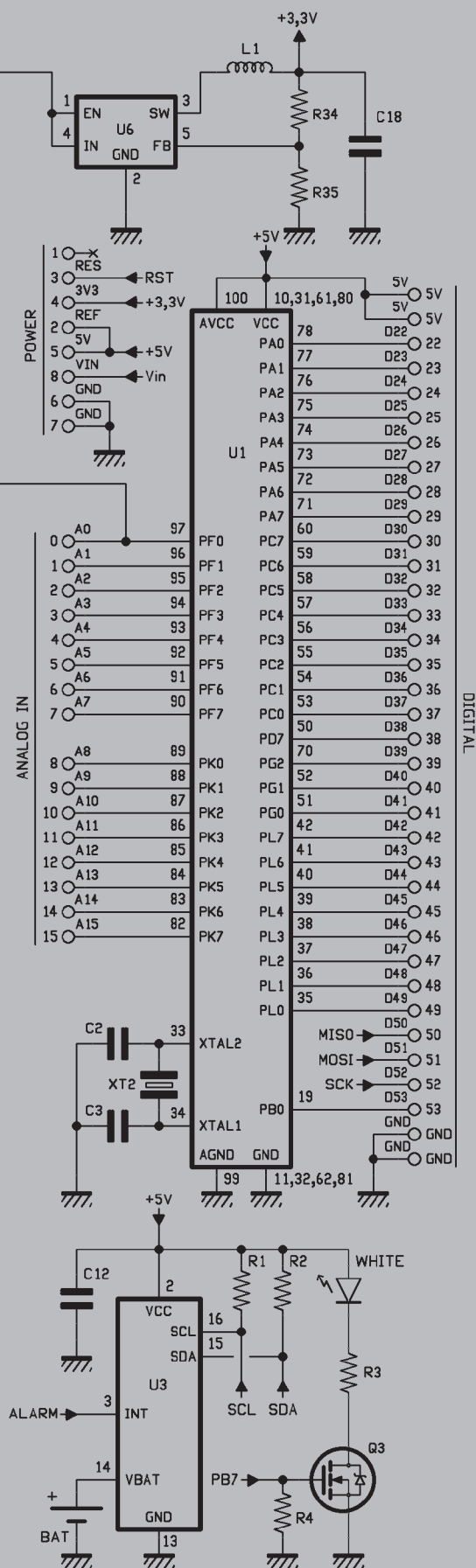
#### **Gestione caratteristiche di basso consumo**

Nella FishinoUNO REV2 sono stati predisposti alcuni I/O utilizzabili dalla libreria FishinoLowPower in modo da non occupare risorse esterne; in particolare, sono gestibili tramite gli I/O interni le funzionalità di reset del modulo WiFi e di risveglio automatico della scheda tramite il suo RTC interno.



COMMUNICATION - PWM - DIGITAL





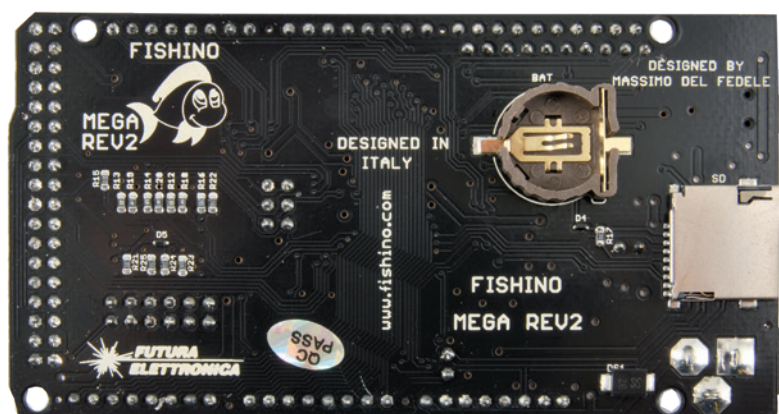
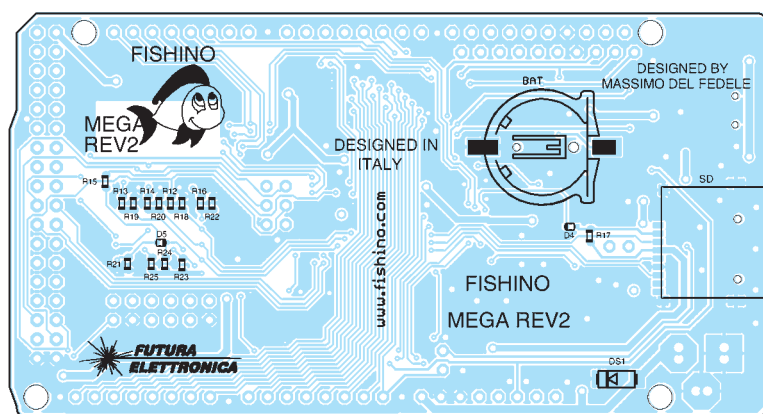
Per attivare queste funzionalità occorre inserire due ponticelli come mostrato nella **Fig. 4**. Una volta inseriti questi ponticelli la libreria **FishinoLowPower** sarà completamente funzionale, mentre non potranno essere utilizzati né l'interfaccia I<sup>2</sup>C aggiuntiva, né la gestione degli allarmi dell' RTC.

La procedura di aggiornamento è semplificata da un programma disponibile sia per la piattaforma Windows che Linux, che esegue l'operazione in modo completamente automatico e a prova di errore. I passi per l'aggiornamento sono i seguenti.

- 1) Caricare uno sketch che non utilizzi la porta seriale; l'esempio base BLINK (quello che fa lampeggiare il LED sulla scheda) è perfetto. Questo passaggio serve ad evitare interferenze tra lo sketch caricato ed il collegamento seriale tramite l'Atmega e l'ESP. Se il programma di flashing non rileva Fishino, quasi certamente è stato caricato uno sketch errato.
- 2) Sul connettore ESPCONN dovete connettere la linea TX di Fishino con la linea ESP-TX e la RX di Fishino con la linea ESP-RX (riferitevi alla **Fig. 1**):
  - connettere la porta GPIO0 a massa tramite un cavetto o un ponticello sempre sul connettore ESPCONN (**Fig. 1**);
  - collegare il Fishino al PC (o premere il pulsante di RESET se già connesso);
  - lanciare il programma FishinoFlasher, assicurandosi che il PC sia connesso ad Internet.

Se i collegamenti sono stati eseguiti correttamente, il programma rileverà la porta a cui è connesso Fishino, determinerà il modello e la versione del firmware attualmente installata, si collegherà ad un server remoto e scaricherà la lista dei firmware disponibili, mostrando l'ultimo e permettendo comunque la selezione delle versioni precedenti nel caso si voglia fare un downgrade (**Fig. 2**). Facendo clic sul pulsante "Flash" verrà avviata la procedura di aggiornamento, alla fine della quale apparirà un messaggio di conferma. Per terminare il programma occorre premere il pulsante "Esci". Nel caso Fishino non venga rilevato automaticamente, è possibile provare a selezionare la porta manualmente. È comunque probabile che siano stati commessi degli errori nei collegamenti. La selezione manuale risulta indispensabile nel raro caso in cui più di un Fishino sia connesso contemporaneamente al PC, nel qual caso il primo viene rilevato automaticamente ma resta la possibilità di sceglierne un

# [piano di montaggio **FISHINO MEGA rev.2**]



## Elenco Componenti:

R1: 4,7 kohm	R29: 2,7 kohm
R2: 4,7 kohm	R30: 100 kohm
R3: 1 kohm	R31: 97,6 kohm
R4: 220 kohm	R32: 13,3 kohm
R5: 1 kohm	R33: 470 ohm
R6: 10 kohm	R34: 475 kohm 1%
R7: 10 kohm	R35: 105 kohm
R8: 1 kohm	C1: 100 nF ceramico
R9: 1 kohm	C2, C3: 22 pF ceramico
R10: 2,4 kohm %	C4: 100 nF ceramico
R11: 1 kohm	C5: 100 nF ceramico
R12: 680 ohm	C6: 1 µF ceramico
R13: 680 ohm	C7: 22 pF ceramico
R14: 680 ohm	C8: 22 pF ceramico
R15: 1 kohm	C9: 1 µF ceramico
R16: 3,3 kohm	C10: 100 nF ceramico
R17: 10 kohm	C11: 1 µF ceramico
R18: 1,5 kohm	C12: 100 nF ceramico
R19: 1,5 kohm	C13: 4,7 µF ceramico
R20: 1,5 kohm	C14: 22 µF 25 VL ceramico
R21: 3,3 kohm	C15: 4,7 µF 25 VL tantalio
R22÷R25: 10 kohm	C16: 22 µF ceramico
R26: 470 ohm	C17: -
R27: 10 kohm	
R28: 2,7 kohm	

altro. Una volta terminata la procedura è sufficiente eliminare i tre collegamenti e Fishino sarà pronto.

### Misura della tensione della batteria

Per misurare tramite l'ADC interno all'ATmega la tensione della LiPo abbiamo previsto un ponticello che collega la batteria direttamente all'ingresso analogico AD0; inserendo tale jumper, mostrato nella Fig. 5, la tensione sarà misurabile semplicemente tramite una `analogRead(A0)`. Il microcontroller utilizzato, l'Atmega 328pb-an, dispone di ulteriori e notevoli caratteristiche aggiuntive, tra cui timer addizionali, PWM aggiuntivi, eccetera. Purtroppo lo spazio qui non basterebbe per elencarle tutte, quindi vi rimandiamo al datasheet del micro. I PWM aggiuntivi sugli header della scheda sono stati comunque marcati, come d'abitudine, dal carattere '#' (cancellito).

### FISHINO MEGA REV2

Passiamo ora alla nuova Fishino MEGA, della

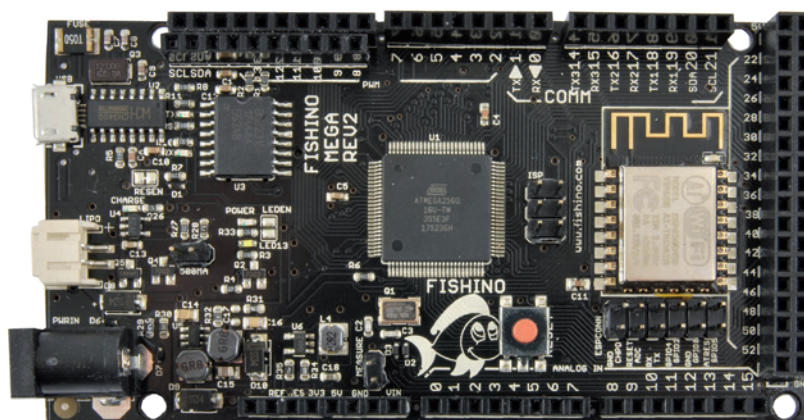
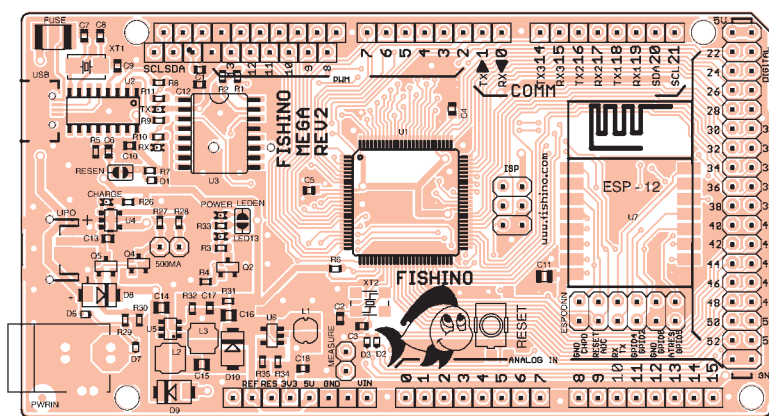
quale in queste pagine trovate lo schema elettrico; da esso vedete come le modifiche siano meno consistenti e si limitino allo sfruttamento di alcuni I/O del microcontroller, inutilizzati sulla prima MEGA, per gestire le periferiche interne ed alcune funzionalità aggiuntive. Anche in questo caso va ricordato che per utilizzare le periferiche interne (WiFi ed SD) ed alcune altre funzionalità è indispensabile installare il package **Fishino** nell'IDE di Arduino e selezionare la scheda Fishino MEGA REV2; utilizzando la scheda come se fosse una Arduino MEGA, le funzionalità estese non saranno accessibili. Il microcontrollore rimane lo stesso della Fishino MEGA originaria (ATmega 2560-16AU) e tale resta anche l'interfaccia USB, che vedete basata sul solito CH340G già descritto per la Fishino UNO REV2. Invariata anche la sezione di reset e la circuiteria correlata all'avvio del caricamento del bootloader, simile a quella della Fishino UNO REV2, dove spiccano il pulsante di reset manuale e il jumper (RESEN) che abilita il reset da DTR del converter USB/seriale.

C18: 10  $\mu$ F ceramico  
 CHARGE: LED rosso  
 WHITE: LED bianco  
 RX: LED giallo  
 TX: LED blu  
 POWER: LED verde  
 D1÷D7: RB521S-30TE61  
 D8÷D10: SS34  
 DS1: SMBJ24CA

Q1: NTR4171P  
 Q2: 2N7002  
 Q3: 2N7002  
 U1: ATMEGA2560-16AU  
 (MF1391)  
 U2: CH340G  
 U3: DS3231  
 U4: MCP73831T-2  
 U5: SX1308  
 U6: LC3406  
 U7: Modulo ESP8266  
 XT1: Quarzo 12 MHz  
 XT2: Quarzo 16 MHz  
 L1: 2,2  $\mu$ H  
 L2: 6,8  $\mu$ H  
 L3: 6,8  $\mu$ H

FUSE: Fusibile 500mA  
 (1210)  
 BAT: Porta Batteria  
 CR1220 da CS  
 RESET: Microswitch  
 SD: Connettore micro-SD  
 USB: Connettore  
 micro-USB

Varie:  
 - Plug alimentazione  
 - Connettore JST 2 vie  
 passo 2 mm  
 - Strip maschio 2 x 3 vie  
 - Strip maschio 2 vie  
 (3 pz.)  
 - Strip femmina 8 vie  
 (4 pz.)  
 - Strip femmina 10 vie  
 (2 pz.)  
 - Strip femmina 18 vie  
 (2 pz.)  
 - Batteria CR1220  
 - Circuito stampato  
 S1391 (101 x 54 mm)



### Modulo RTC

Iniziamo con la modifica più evidente, ovvero la sostituzione dell' RTC con il DS3231: questo dispositivo integra un quarzo molto preciso e calibrabile e un'uscita di allarme. A differenza della Fishino UNO, qui non è necessario utilizzare alcun ponticello per abilitare la funzionalità allarme, visto che gli I/O disponibili non utilizzati in precedenza sono già sufficienti per gestire tutte le funzionalità aggiuntive. La libreria FishinoRTC è già in grado di gestire tutte le funzionalità del nuovo RTC, incluso l'allarme.

### I/O interni per WiFi e microSD

Come per la Fishino UNO REV2 sono stati liberati gli I/O 4, 7 e 10, utilizzati precedentemente per la scheda microSD ed il modulo WiFi; restano impegnati i pin dell' interfaccia SPI (50, 51 e 52) che sono comunque utilizzabili contemporaneamente su altre periferiche SPI.

Per gestire SD e WiFi è sufficiente sostituire negli sketch, se non fatto in precedenza (in tutti i nostri esempi troverete sempre le costanti e non i numeri

direttamente) gli I/O 4 e 10 rispettivamente con SDCS e WIFICS; tutto il resto rimane invariato.

### I/O interni per le funzioni Low Power

Le funzionalità della libreria FishinoLowPower richiedono l'uso di 2 I/O per gestire lo spegnimento del modulo WiFi ed il risveglio dell'Atmega tramite l' RTC interno al modulo WiFi stesso; nella versione precedente della scheda erano necessari 2 ponticelli esterni ed il conseguente impegno di 3 I/O sulla scheda, che ora non sono più necessari.

La libreria FishinoLowPower è in grado di controllare, da subito e senza alcun ponticello su scheda, le funzionalità di risparmio energetico disponibili.

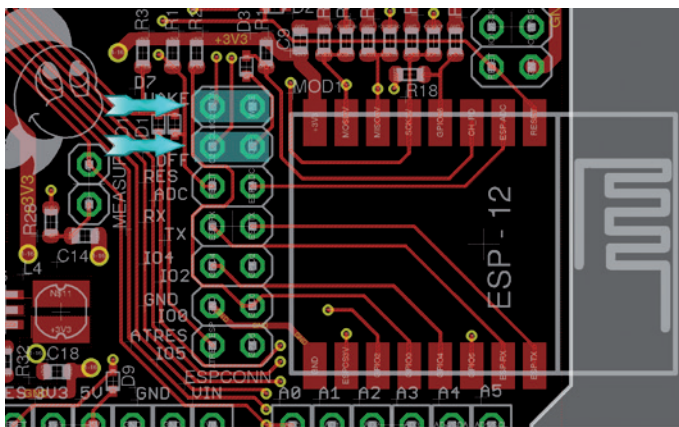
### Caricamento sketch via WiFi (OTA)

Come per la precedente versione è possibile caricare gli sketch tramite WiFi; nella REV2 abbiamo però riposizionato i connettori in modo da poter abilitare la cosa tramite un semplice ponticello, come vedete nella Fig. 6.

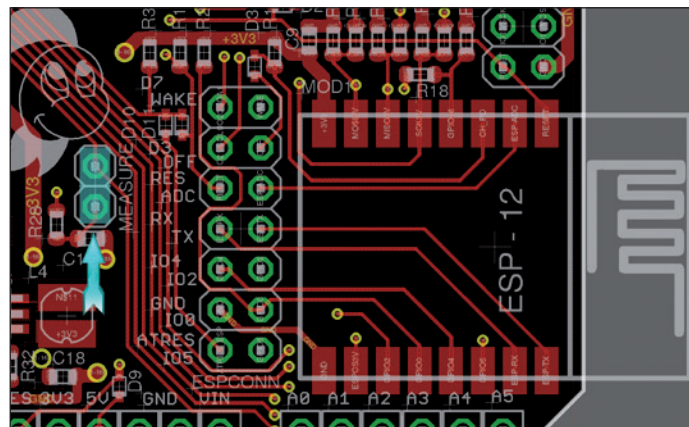
### Protezione dalle sovratensioni

Utilizzando cavi di alimentazione molto lunghi





**Fig. 4** - I jumper necessari ad attivare la modalità low-power.



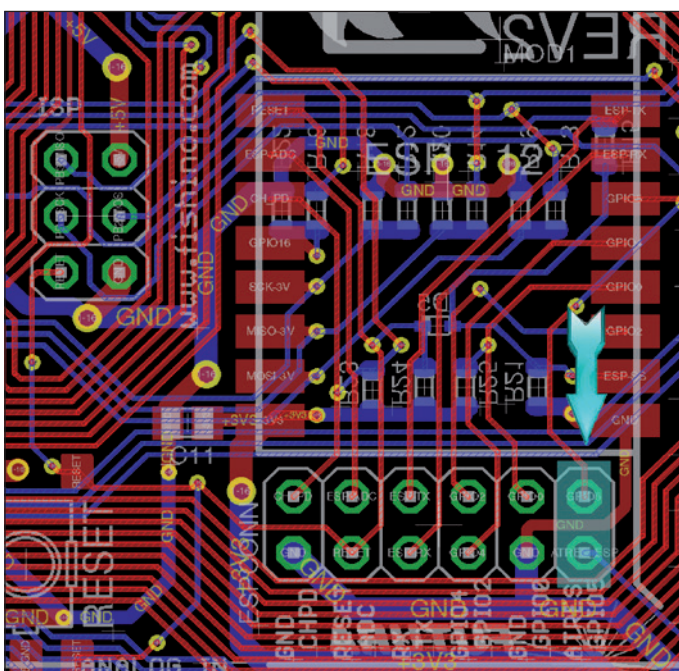
**Fig. 5** - Il jumper per la misura della tensione della LiPo.

sull'ingresso VIN (header) o sul plug di alimentazione è possibile che all'accensione e/o allo spegnimento dell'alimentatore vengano generati degli spike (impulsi) ad alta tensione dovuti all'eccessiva induttanza dei cavi stessi, in grado di bruciare il convertitore di ingresso.

Questa è un'eventualità piuttosto rara, ma è comunque successa in un paio di occasioni (su qualche migliaio di schede in circolazione); abbiamo quindi previsto un diodo soppressore di extratensioni sull'alimentazione esterna in grado di cortocircuitare a massa qualsiasi extratensione presente. Nella MEGA REV2 abbiamo inserito il soppressore DS1 a protezione della linea del jack di alimentazione PWRIN.

## CONCLUSIONI

Concludiamo qui la presentazione delle versioni aggiornate delle nostre schede Fishino ad 8 bit; tutte le librerie disponibili ed i package dell'IDE sono già stati aggiornati, ove necessario, per gestirle in modo ottimale. Non stiamo a descrivervi la realizzazione pratica perché per quanto in questo articolo riportiamo i piani di montaggio e la lista dei componenti, la costruzione delle schede è molto complessa perché fa uso di PCB doppia faccia a fori metallizzati densamente popolati, per di più con componenti SMD molto piccoli, il cui posizionamento richiede particolari attrezzature che non tutti posseggono. Nulla vieta, a chi se la sentisse, di realizzare da sé le REV2; per tutti gli altri, come al solito le board sono disponibili già montate e collaudate presso la Futura Elettronica, [www.futurashop.it](http://www.futurashop.it), che le fornisce complete di bootloader nel microcontrollore e quindi pronte per ricevere i vostri sketch, naturalmente da un IDE con caricato il package Fishino. ■



**Fig. 6** - Connettore ESPCONN della MEGA REV2.



## per il MATERIALE

Le board Fishino UNO REV2 e Fishino MEGA REV2 sono disponibili presso Futura Elettronica. Fishino UNO REV2 (cod. FISHINO) è disponibile a Euro 36,00 e Fishino MEGA REV2 (cod. FISHINOMEGA) è in vendita a Euro 49,90. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)  
Tel: 0331-799775 - Fax: 0331-792287 - [www.futurashop.it](http://www.futurashop.it)