

Sistema per l'analisi dell'acqua in un acquario

di

Gianluigi Nozza

Ho conosciuto Paolo Bonelli per caso, navigando in rete alla ricerca di ispirazione per la realizzazione di un sensore "fai da te" di conducibilità elettrica. Ne avevo acquistato uno commerciale che aveva funzionato per qualche tempo e poi si era rotto proprio quando mi accingevo ad impiegarlo in fase operativa. E' stato un maledetto colpo di fortuna, e non scherzo. Se il sensore non si fosse rotto, non avrei potuto approfondire l'argomento nè acquisire l'esperienza necessaria a gestire questo tipo di sensori.

Girovagando in rete infatti, sono capitato sul sito di elettronica e domotica di Mauro Alfieri, in una sezione che illustrava la realizzazione di un sensore di conducibilità elettrica, che sarebbe dovuto servire di supporto alla gestione di un orto idroponico, <http://www.mauroalfieri.it/elettronica/orto-idroponico-verticale-con-arduino-seconda-parte.html>. L'articolo in questione fa riferimento ad un sensore di conducibilità realizzato per l'appunto da Paolo Bonelli e conteneva anche il codice per gestirlo, una vera e propria miniera di informazioni utilissime che mi ha fatto fare un salto di qualità nella gestione di Arduino.

Il nostro amico Paolo che avevo contattato inizialmente per qualche chiarimento, e a conclusione del mio progetto, per ringraziarlo della sua disponibilità, mi ha poi chiesto, nello spirito di condivisione e dell'open source di scrivere un contributo da pubblicare sul sito di Coscienza Ambientale, ciò che mi accingo a fare con qualche precisazione.

La ragione per la quale mi sono interessato alla realizzazione di un sensore EC, è che la conducibilità elettrica, era uno dei parametri che mi accingevo a monitorare con una centralina per la gestione del mio acquario. Cosa c'entra un acquario col sito di Coscienza Ambientale ?

Forse non molto, se non fosse che la gestione di un acquario può essere fatta da persone con coscienza ambientale, oppure senza e io cerco di appartenere alla prima categoria. Cosa significa per me avere coscienza ambientale nella gestione di un acquario ? Ad esempio rinunciare al marino in favore dell'acquario di acqua dolce in quanto il prelievo in natura di pesci marini tropicali, che non si riproducono in cattività, se non in circostanze eccezionali, determina un commercio nella maggior parte dei casi illegale, che depauperava le barriere coralline. Per ogni esemplare che viene venduto nei negozi specializzati è molto probabile che siano centinaia gli individui che non sopravvivano alla cattura e al trasporto.

Questo problema può essere presente anche nell'acquisto di pesci di acqua dolce, ma da decenni ormai esistono vere e proprie industrie di allevamento per soddisfare le richieste degli acquariofili e, sapendo cosa acquistare, rinunciando agli esemplari cosiddetti wild, o rilevando gli esemplari nati in cattività da altri appassionati allevatori, si può godere di un pezzo di natura in casa con la certezza di non aver depauperato l'ambiente naturale e alimentato attività illegali.

Inoltre l'esperienza che sto per descrivere, mi auguro possa servire in qualche altro campo magari con un interesse ed un'impronta più ecologica di quella che può avere un acquario domestico.

Vi riassumo quindi come tutto è iniziato.

Era da molto tempo ormai che sentivo parlare di questo Arduino e francamente avevo anche accarezzato l'idea di approfondire l'argomento, ma il fatto di non avere basi di elettronica mi ha sempre un po' frenato. Un giorno, frequentando un forum di acquariofili come me, ho conosciuto Alessandro che in un post illustrava

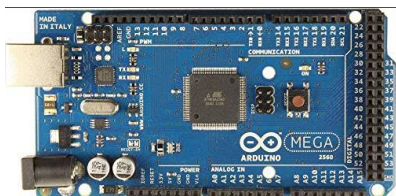
la centralina di controllo del suo acquario, costruita, non senza difficoltà, data la sua scarsa esperienza di programmazione.

Ho approfittato della generosità di Alessandro che ha reso disponibile integralmente il suo codice, e ha illustrato la sua centralina con una copiosa documentazione fotografica, e dopo 5 minuti di analisi del codice di Alessandro, di cui riuscivo a cogliere gli elementi essenziali di controllo dei diversi moduli, avevo già deciso che anche io mi sarei cimentato nell'impresa, quantunque il groviglio di fili che collegavano i diversi moduli della sua centralina, mi incutessero un certo timore. Insomma il codice mi sembrava intelligibile, molto meno l'aspetto hardware.

Ho scoperto dopo che Arduino non è una piattaforma hardware e software per esperti di elettronica; anzi probabilmente è vero il contrario vista la disponibilità di moduli hardware per ogni tipo di applicazione; inoltre esiste moltissimo codice open source, già disponibile in rete per ognuno dei moduli hardware in commercio, pronto per essere adattato alle proprie esigenze.

Dopo qualche giorno di pianificazione decisi che la mia centralina di controllo avrebbe monitorato ad intervalli di tempo regolari, i seguenti parametri: temperatura, pH, conducibilità elettrica e la portata di una pompa. I dati acquisiti dai sensori sarebbero stati scritti su una scheda SD, inviati ad una pagina Webserver accessibile da remoto e scritti su di un display LCD (data, ora, Temp, pH, EC, Total Dissolved Solid calcolato, portata della pompa principale dell'acquario). Ho poi anche aggiunto un sistema di allertamento mediante trasmissione email attraverso il modulo wifi e la rete di casa, qualora uno dei parametri monitorati risultasse fuori da determinati limiti prestabiliti. Infine, per risparmiare un po' di energia, ho predisposto lo spegnimento dell'LCD dopo 5 minuti dall'accensione, che viene attivata mediante un sensore ad ultrasuoni, all'apertura di un'anta del mobile di supporto dell'acquario.

Passo ad illustrare ora i componenti hardware utilizzati, si trovano nei siti di commercio più conosciuti sia europei che cinesi. Non starò ad illustrare le modalità di cablaggio di questi componenti. Si trovano facilmente in rete



Arduino Mega – alcuni test di compilazione falliti con Arduino UNO mi hanno convinto di impiegare la versione più carrozzata e performante di Arduino in commercio. Arduino mega rev 3.

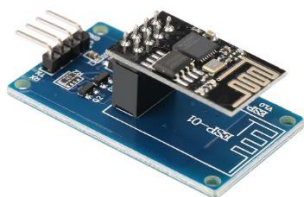


RTC Clock – Ho impiegato il modulo DS3231. Si tratta di un modulo Real Time Clock che impiega batterie CR2032. Ha però un problema, ha un circuito di ricarica della batteria e se si usa una CR2032 normale c'è il rischio che alla lunga la batteria si deformi o esploda. Purtroppo anche l'impiego dell' LIR2032 cioè della versione ricaricabile della batteria sembra non risolva il problema in quanto il voltaggio di ricarica sembra non sia ottimale. Insomma dopo aver cercato un po' in rete mi sono convinto che la cosa migliore era disabilitare il circuito di ricarica, cosa che ho effettuato eliminando il resistore 102 (cerchiato in rosso) con un saldatore elettrico a punta fine. Ho quindi impiegato una normale

CR2032 che dovrebbe avere una autonomia di parecchi anni senza bisogno di aggiornare la data. Una volta che la data e l'ora siano state impostate correttamente, il modulo è in grado di ritenere correttamente data e ora, fino a che rimane collegato alla batteria, anche se è staccato da Arduino,



Il modulo data logger che ho utilizzato è quello illustrato nella figura a sinistra, impiega una scheda micro SD, io ne ho usato una da 16 Gb. Di fatto consente ad Arduino di disporre di un HD in miniatura dove stoccare tutti i dati di interesse.



Il modulo wifi che ho impiegato è un ESP8266, ESP-01 con il suo adattatore che consente di alimentare il modulo a 5 volt. Questo modulo consente l'interfacciamento alla rete domestica e quindi anche un controllo ed una operatività da remoto.

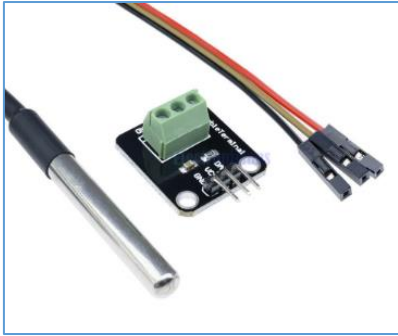
Il modulo pH impiegato è il seguente. Consente di impostare un valore di voltaggio a pH 7 pari a 2.5 Volt che si modifica in funzione della variazione di pH. Necessita ovviamente di taratura iniziale. Si trovano numerosi documenti in rete su come effettuare una taratura corretta. Si tratta di un modulo con connettore BNC al quale si deve collegare la sonda pH.



Ho poi acquistato un display lcd da 20 caratteri e 4 righe per rappresentare i valori istantanei dei parametri monitorati. Per semplicità ho acquistato la versione I2C che consente il collegamento ad Arduino con solo 4 fili invece che 10.



Un sensore di distanza ad ultrasuoni, consente il monitoraggio della distanza di un'anta posta davanti alla centralina e in caso di apertura dell'anta stessa, attiva l'accensione del backlight del display LCD. Dopo 5 minuti dall'accensione, se il sensore non rileva attività nei pressi della centralina, il backlight viene disattivato, riducendo i consumi.



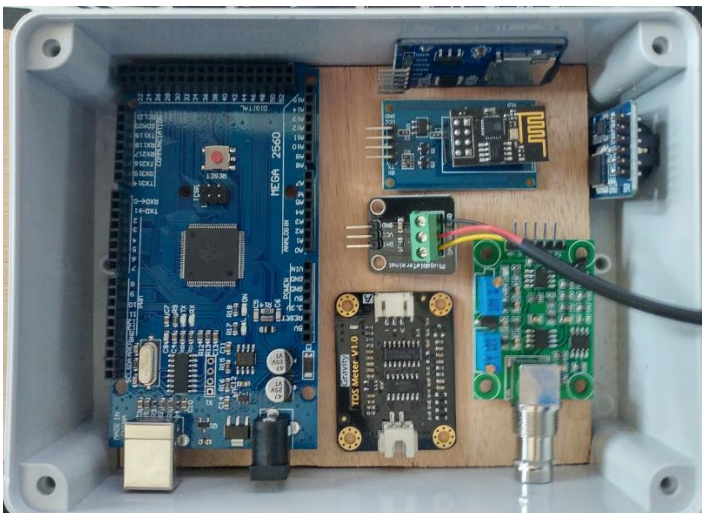
Il sensore di temperatura che ho impiegato è quello illustrato dalla fotografia a sinistra, arriva già con una schedina di collegamento ad Arduino e con i relativi jumpers.



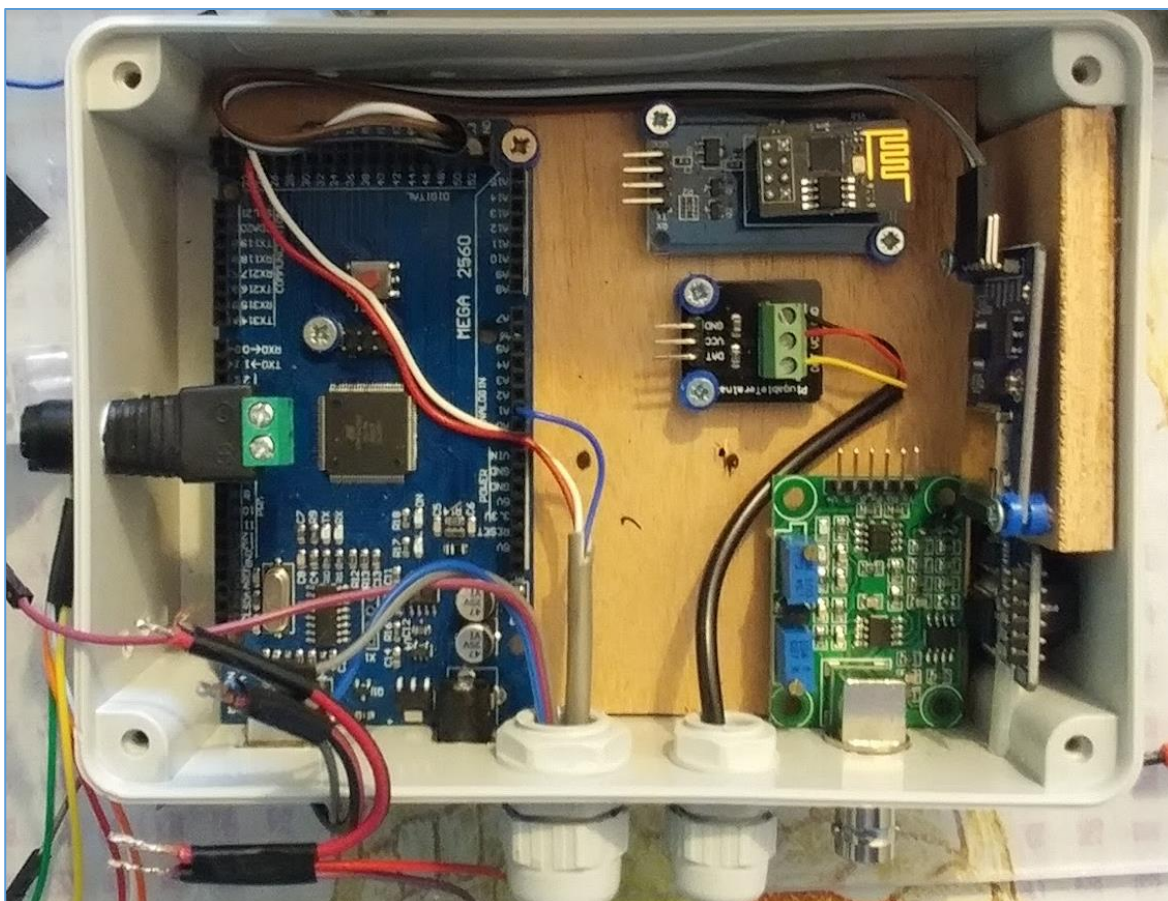
Quello illustrato nella foto è invece un sensore di portata. E' costituito in pratica da un micromulinello la cui velocità di rotazione, dipende dal flusso idrico in ingresso. Maggiore è la portata idrica transitante, più velocemente ruota il micromulinello.



Il sensore di conducibilità elettrica era (perchè poi si è rotto) un TDS Gravity ver. 1 della DFRobot. E' costituito da una sonda con due elettrodi terminali, e da un modulo in grado di alimentare il sensore a corrente alternata per evitare fenomeni di polarizzazione e di elettrolisi.



Nella foto qui a sinistra sto facendo i primi tentativi di posizionamento dei vari moduli in una cassetta di derivazione.



Nella figura qui sopra i moduli sono stati posizionati definitivamente. Come potete vedere ho eliminato il modulo EC della DFRobot. Sulla sinistra si nota anche un connettore DC al quale è collegato un alimentatore esterno da 5 volt e 2A, che alimenta tutti i moduli della centralina ad eccezione di arduino che viene alimentato dal suo alimentatore a 9 Volt.

La mia versione della sonda EC di Paolo Bonelli.



Non avendo sottomano del plexiglass e non disponendo di attrezzatura sofisticata come quella di Paolo, mi riferisco in particolare al taglio laser, ho deciso che avrei realizzato la sonda EC, con materiali di più facile reperimento. I materiali utilizzati sono stati:

- vecchio cavo telefonico di recupero a tre conduttori
- un tubo di plastica cavo del diametro di circa 12 mm e della lunghezza di 15-16 cm.
- una resistenza da 470 Ohm
- delle guaine termoretraibili per l'isolamento degli elettrodi e delle connessioni, comprate al brico
- del filo al nichelchromo da 1 mm di diametro circa e 1,4 Ohm al metro di resistenza comprato su ebay.

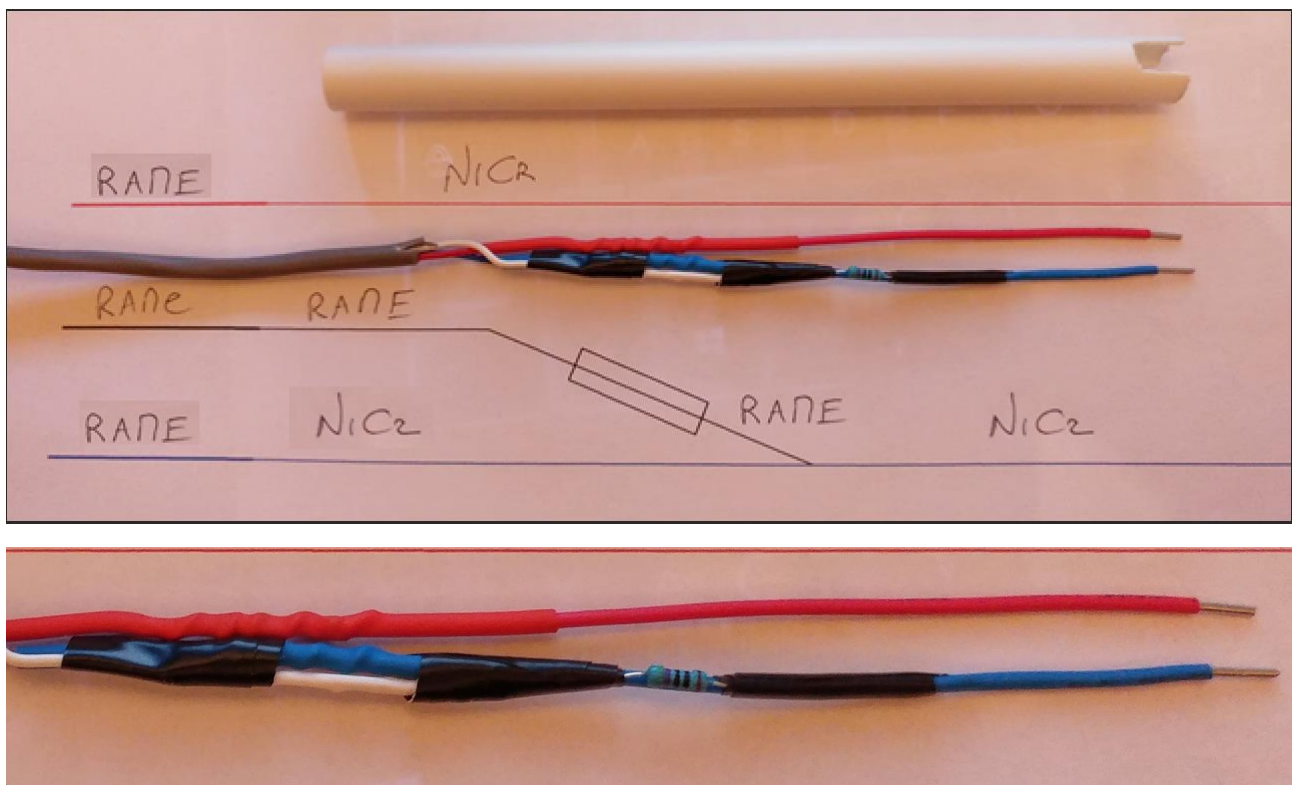
La scelta di usare del filo al nichel cromo (80/20) come elettrodi, è nata dalla necessità di impiegare un materiale resistente alla corrosione visto che la sonda è perennemente immersa in acqua 24 ore su 24. Si sarebbe potuto impiegare del filo d'oro, o palladio o platino ovviamente e forse con performance ancora superiori in riferimento alla resistenza alla corrosione, ma non mi sembrava il caso.

Lo spessore da 1 mm è tale che con le tensioni applicate (5 volt) e le lunghezze degli elettrodi (11 cm ciascuno), il riscaldamento degli elettrodi per effetto joule è praticamente trascurabile e quindi la resistenza degli elettrodi rimane sostanzialmente costante e così la risposta della sonda. Se si usasse filo più sottile gli elettrodi si riscalderebbero, (Il nichel cromo è materiale fortemente resistivo) e la sonda avrebbe una risposta affetta da deriva.

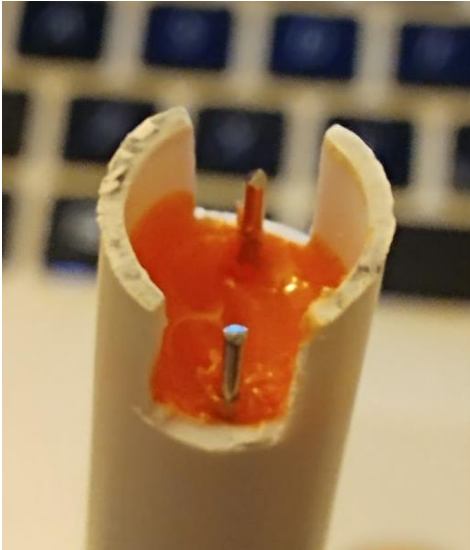
La figura seguente illustra la sonda EC prima dell'assemblaggio. In alto il tubo di plastica di circa 15 cm di lunghezza, in basso il filo elettrico con i terminali di nichel cromo. Il filo rosso (P1 di Paolo) è semplicemente un filo di rame alla cui estremità è saldato a stagno uno spezzone di 11 cm di filo al nichel cromo. La saldatura è stata ricoperta con guaina termoretraibile per elettricisti in maniera da isolare l'elettrodo ad eccezione del tratto terminale di circa 5-6 mm.

Il filo blu (P3 di Paolo) è identico al filo rosso se non che a circa metà dell'elettrodo al nichel cromo, è saldato il terminale di una resistenza da 470 Ohm, che all'altro capo è saldata al terzo conduttore elettrico, quello di colore bianco (P2 di Paolo).

Come vedete lo schema ricalca esattamente il sensore EC di Paolo Bonelli, solo con materiali diversi. Il filo blu è collegato ad un Pin analogico (nel mio caso A1). Il filo rosso e il filo bianco a due pin digitali nel mio caso 24 e 25.



Una volta effettuate le saldature, isolati i conduttori con le guaine termoretraibili, ho provveduto ad inserire la sonda nel suo tubicino provvedendo ad isolare entrambe le estremità con del mastice per PVC (volendo si può utilizzare anche del normale silicone).



Dettaglio del terminale della sonda con i due elettrodi al nichel cromo scoperti. La sagomatura del tubo di plastica serve ad evitare che gli elettrodi vengano urtati accidentalmente e si possano piegare e contemporaneamente a non ostacolare il flusso idrico attorno agli elettrodi.

Aspetto finale della sonda EC



Quando mi sono imbattuto per la prima volta nel codice di Paolo, non avevo ancora realizzato che Arduino potesse alimentare piccoli dispositivi (come per l'appunto la sonda di conducibilità) mediante i pin digitali (poi ho scoperto che la cosa è possibile anche con i pin analogici). Immaginavo che per alimentare un modulo o un sensore fosse necessario collegarlo obbligatoriamente al pin dei 5 volt di Arduino o ad un alimentatore esterno. Nel codice di Paolo la sonda invece viene alimentata mettendo in "high" con l'istruzione "digitalwrite" uno dei due pin (es. P1) e in "low" l'altro pin (es. P3). In high il pin manda in uscita i 5 volt mentre in low il pin ha tensione nulla, esattamente come quando si collega un dispositivo ai 5 volt e al GND di Arduino.

Quando poi ho capito che addirittura Paolo simulava una alimentazione della sonda mediante corrente alternata AC invertendo all'interno del ciclo di lettura la polarità dei pin, con lo scopo di impedire l'insorgenza di fenomeni di polarizzazione continua e limitando i fenomeni di elettrolisi a livello degli elettrodi, ho ringraziato il cielo che mi si fosse rotto l'elettrodo TDS Gravity. Potevo farne tranquillamente a meno, anzi avevo in mano un dispositivo certamente superiore, più versatile, perchè potevo decidere io in maniera autonoma se alimentare il sensore, quando alimentarlo, con che frequenza invertire la polarità..... credetemi ero letteralmente al settimo cielo.

La taratura del sensore.

Ovviamente essendo la mia sonda completamente diversa nei materiali, nelle superfici esposte al fluido, nelle distanze tra gli elettrodi, è evidente che i parametri di taratura inseriti nel codice di Paolo, non potessero essere impiegati e che fosse necessaria una taratura ad hoc.

Poichè dispongo di un conduttivimetro professionale che ogni tanto taro, ho pensato di ricavarmi le soluzioni di taratura prendendo della comune acqua di rubinetto che ho provveduto a diluire con acqua distillata ottenendo 7 soluzioni a concentrazione e a conducibilità diverse. L'intervallo di conducibilità di mio interesse è inferiore a 300 uS, e dato che l'acqua di rubinetto è intorno ai 300 uS il metodo adottato va benissimo.

Ovviamente anche il metodo di acquistare 5 o 6 acque minerali a diversa conducibilità è valido ma non avevo voglia di fare un'indagine di mercato sulla conducibilità delle acque minerali in commercio al supermercato.

Nella tabella seguente sono elencati i valori di resistività misurati dalla mia sonda autocostruita, alle diverse conducibilità misurate dal conduttivimetro professionale. Ovviamente la resistività è inversamente proporzionale alla conducibilità elettrica misurata.

RS	uS
3.857	297
6.389	186
8.212	147
11.709	100
15.494	79
21.657	52
51.995	25

Quando si fa la taratura di un qualunque sensore particolare attenzione bisogna riservare a prevenire a qualunque forma di contaminazione, bastano poche gocce di una soluzione che passano da un bicchiere all'altro, per compromettere la bontà della taratura.

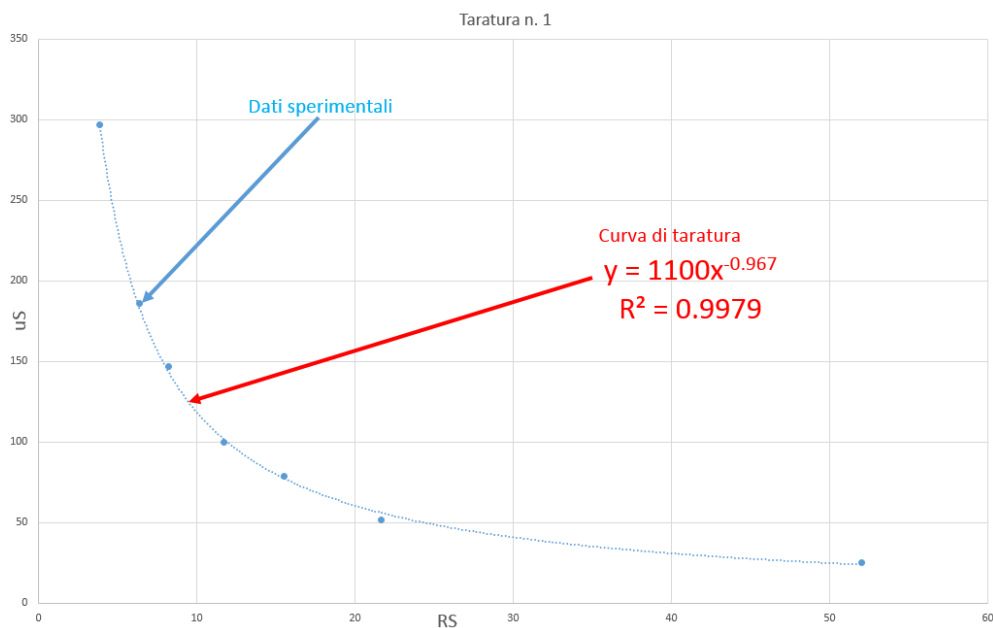
Nel grafico seguente invece sono stati inseriti in ascisse i valori di RS (media delle letture di resistività rilevate) e in ordinate i valori di conducibilità elettrica EC delle soluzioni di taratura impiegate (rilevate da conduttivimetro esterno). Si è quindi sovrapposta la linea di tendenza che meglio era in grado di interpolare i risultati sperimentali. La curva interpolatrice è una elevazione a potenza di equazione $y=1100x^{-0.967}$ e presenta un ottimo coefficiente di correlazione R^2 pari a 0.9979.

$A = 1100$ e $B = -0.967$ sono quindi i parametri di taratura della mia sonda e differiscono da quelli di Paolo che sono i seguenti: per inciso Paolo riporta due diverse serie di parametri di taratura perchè il suo codice gestisce due sensori di conducibilità mentre il mio uno solo.

```
float A[] = {870.43,822.71};
```

```
float B[] = {-1.194,-1.190};
```

Ovviamente chiunque dovesse realizzare una sonda ex novo, dovrebbe effettuare la taratura perchè questa dipende da una serie di fattori che sono difficilmente riproducibili: lunghezza, spessore, distanza e superficie degli elettrodi....



Il codice di gestione della sonda.

Dovendo integrare il codice di Paolo nel mio, ho evidentemente apportato delle modifiche. Si tratta di modifiche veramente minori, in primis il mio codice gestisce un solo sensore, e non due come quello di Paolo, poi ho inserito alcune istruzioni condizionali per attivare la lettura del sensore ad intervalli di un ora, ma solo se non è attivo il sensore di pH per evitare interferenze legate a microcorrenti spurie vaganti.

Nelle ultime pagine c'è il codice relativo alla lettura della sonda.

Poichè le mie letture avvengono non troppo spesso, ho preferito uscire dal ciclo di lettura del sensore, mettendo in low entrambi i pin digitali che alimentano la sonda.

Così facendo non c'è differenza di potenziale tra i due elettrodi della sonda quando questa non è in uso e quindi gli elettrodi sono meno sollecitati.

Spero che questa mia personale interpretazione della sonda EC di Paolo Bonelli possa servire a qualcun altro, non necessariamente nel campo dell'acquariofilia. Grazie ancora Paolo.

MANI	APERTURA_ANTE	BACKLIGHT	CLOCK	CONTROLLO_PARAMETRI	EC\$	EMAIL	LCD_PRINT	LETTURA_SENSORI	PORTATA	SD_WRITE	WIFI	
1	*											
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												

```

1  * Conductivimetro. Codice originale Paolo Bonelli, modifiche minori di Gianluigi Nozza
2  Il codice utilizza un solo sensore, a differenza di quello di Paolo che era stato strutturato per due.
3
4  Collegamenti hardware
5  sensore rosso      blu      bianco
6  P1 -----O 0-----P2-----RRRR-----P3      Lato sensore
7  24 elettrodi A0 470 ohm 25      Lato Arduino
8  */
9  float Leggi_EC() {
10     if (((long)(millis() - last_EC_read) > samplingInterval_EC) && (state == 0)) { // tiene conto dell'overflow di millis() la lettura viene fatta solo se la sonda pH non è alimentata, per eliminare le eventuali interferenze,
11         float RS1 = 0.;
12         float RS2 = 0.;
13         float VCC = 5000.;
14         float Vconv = VCC / 1024.;
15         float V1 = 0.;
16         float V2 = 0.;
17         float VSG = 0.;
18         float runningAvg1 = 0.;
19         float runningVSG = 0.;
20         float runningAvg3 = 0.;
21
22         // ciclo di misura sul sensore
23         for (byte i = 0; i < Numb_EC_Readings; i++) {
24             digitalWrite(P1_EC, HIGH);
25             runningAvg1 = runningAvg1 + Vconv * analogRead(P2_EC);
26             digitalWrite(P1_EC, LOW);
27             runningVSG = runningVSG + Vconv * analogRead(P2_EC);
28             digitalWrite(P3_EC, HIGH);
29             runningAvg3 = runningAvg3 + Vconv * analogRead(P2_EC);
30             digitalWrite(P3_EC, LOW);
31         }

```

```

VSG = runningVSG / Numb_EC_Readings;
V1 = runningAvg1 / Numb_EC_Readings;
V2 = runningAvg2 / Numb_EC_Readings;
V3 = runningAvg3 / Numb_EC_Readings;
// compute RH and VS
if (V1 < 2.) {
    RS = 60000.;
    CS = 0.;
} else {
    // correppo con la galvanica
    V1 = V1 - VSG;
    V2 = V2 - VSG;
    // primo calcolo di RS
    RS1 = R1 * (VCC - V1) / V1;
    // secondo calcolo di RS e media tra i due (RS in kohm)
    RS2 = R1 * V2 / (VCC - V2);
    RS = (RS1 + RS2) / 2000.;
    // calcolo CS secondo i parametri di taratura
    if ( RS < 0.5) {
        CS = 1300;
    }
    else {
        CS = A * pow(RS, B);
    }
    // conducibilità normalizzata a 20 gradi centigradi.
    CS_20 = CS * (1 - (temperature - 20) * 0.0205);
}

// voltaggio medio letto su analogico P2 con P1 e P3 in low (correnti spurie)
// voltaggio medio letto su analogico P2 con P1 high e P3 in low
// voltaggio medio letto su analogico P2 con P1 low e P3 in high
// esiste un valore di V1 al di sotto del quale il liquido si può considerare completamente resistivo (es. acqua distillata) e la resistenza RS molto grande
// e la conducibilità zero. Questo valore al momento V1 = 2,0 dovrà essere scelto con la nuova sonda dopo taratura

// depurazione dei voltaggi letti, dalle correnti spurie eventuali

// R1 = 470 Ohm. Il circuito del sensore non è simmetrico, la resistenza è collegata solo al P3 quindi il calcolo di RS1 e RS2 è diverso
// il circuito del sensore non è simmetrico, la resistenza è collegata solo al P3 quindi il calcolo di RS1 e RS2 è diverso
// calcolo resistenza media in Kohm (/2000)
// taratura da fare sarà un valore diverso,
// taratura da fare sarà un valore diverso

// calcola la conducibilità. La funzione di interpolazione è una elevazione a potenza con esponente negativo. A*RS^B
// 2% di variazione per ogni grado centigrado Se la temperatura è > di 20 gradi CS è maggiore di CS_20 e la correzione negativa
// Se la temperatura è < di 20 gradi CS è minore di CS_20 e la correzione positiva

last_EC_read = null;

```