



Leonardo da Vinci
istituto tecnico industriale



Via Peruzzi, 9 – Carpi (MO)
Tel. +39 059 695241
Fax + 39 059 643028
Cod. Fisc. 81004250361

itivinci@itivinci.mo.it - motf030004@istruzione.it - motf030004@pec.istruzione.it - www.itivinci.mo.it - MIUR: MOTF030004

Costruzione di un pH-metro

Simone Vacondio

Tesina di maturità 2013

Costruzione di un pH-metro

Autore:

Simone Vacondio

E-mail:

simpe94@gmail.com

Tutti i diritti riservati. © 2013 - Simone Vacondio

Quest'opera è rilasciata sotto licenza CC BY-NC-SA 3.0. L'enunciato integrale della licenza è reperibile all'indirizzo internet <http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>

Sei libero:

- di **condividere**: copiare, distribuire, trasmettere l'opera;
- di **modificare**: adattare l'opera.

Alle seguenti condizioni:

- **Attribuzione**: devi attribuire la paternità dell'opera nelle modalità specificate dall'autore o dal licenziatario, senza suggerire in alcun modo che essi avvallino te o il tuo uso dell'opera;
- **Non commerciale**: non puoi usare quest'opera per fini commerciali.
- **Condividi allo stesso modo**: se alteri, trasformi o ti basi su quest'opera, puoi distribuire il lavoro finito solo sotto la stessa, o simile, licenza dell'opera.

Un ringraziamento a Mirco, che mi ha aiutato in questo progetto; ai miei amici Chiara, Elena, Eugenio, Federica e Giulia, che se non mi mettesero di buon umore non riuscirei a cimentarmi in certe esperienze; ai miei insegnanti di questi cinque anni di superiori, per tutto ciò che mi hanno trasmesso; ai miei genitori, che mi sostengono sempre.

Premessa

La scelta di progettare e costruire un piaccametro, documentando dettagliatamente tutti i passi richiesti, è stata spinta dalla volontà di cimentarsi in un'esperienza che potesse coniugare organicamente le varie conoscenze acquisite nel corso di studi FASE.

A tal proposito, questo fascicolo mette in relazione tutte le diverse discipline delle quali è necessaria una conoscenza di base per cimentarsi nel progetto. Esso può costituire, inoltre, un modo per tentare un primo approccio (non senza, però, armarsi di buona volontà e voglia di imparare) con gli argomenti presentati, in modo che una fase applicativa succeda immediatamente a quella nozionistica.

Chi scrive e, in genere, chi si iscrive ad un istituto tecnico, riconosce l'importanza dell'aspetto pratico nell'apprendimento: vedere, provare, "toccare le cose con mano" sono esperienze in grado, più di qualsiasi lezione frontale ed enunciazione teorica, di stimolare la curiosità, sciogliere un nodo critico nella comprensione di un concetto e creare nello studente una conoscenza solida e organica.

Solitamente si pone l'accento sulla tesi per cui gli studi tecnici siano per chi cerchi un accesso immediato al mondo del lavoro, in virtù dell'importanza che essi attribuiscono alle abilità pratiche. Questo discorso molto inflazionato tende a essere fuorviante nel momento in cui porta alle errate conclusioni per cui il perito tecnico sia una figura affidabile negli ambiti pratici del lavoro, ma da cui si può pretendere poco a livello concettuale. La scelta dello studio tecnico dovrebbe, semmai, essere fondata sull'adesione ad una filosofia di apprendimento, ispirata al metodo scientifico, che lega inestricabilmente la conoscenza al risvolto concreto. Un simile approccio ha anche l'indiscutibile qualità di stimolare la creatività e favorire un'introduzione alla via ingegneristica di risolvere certi problemi di progettazione.

Per queste ragioni il progetto, che mira a collegare tra loro diverse discipline, è prima di tutto un'esperienza pratica e tangibile, che segue all'assimilazione di elementi di base fondamentali.

Il percorso prevede un'escursione nella chimica e nella fisica, per quanto concerne la definizione di pH, acidità e basicità e il funzionamento dell'elettrodo a vetro; nell'elettronica per la parte di condizionamento del segnale inviato dall'elettrodo a vetro; nell'informatica, per la parte di programmazione del microcontrollore, allo scopo di dotare il dispositivo di diverse funzionalità.

È opportuno sottolineare che alcune conoscenze richieste per questo progetto, in particolare quelle di informatica, non sono

fornite dal corso FASE. Come già accennato, tuttavia, non si possono intraprendere simili esperienze senza voglia di imparare, di compiere un approfondimento personale o anche di chiedere aiuto a persone competenti in un particolare argomento. La passione deve essere il sentimento che guida nel processo creativo, altrimenti tutto il lavoro si rivelerà un percorso sterile e non lascerà in chi lo compie quella consapevolezza, soddisfazione, gratificazione e arricchimento che conseguono a questo tipo di esperienze.

Simone Vacondio



Prove in laboratorio

L'autore di questo fascicolo esegue un test sul suo sudato progetto.

Sommario

Parte 1: Chimica 1

L'importanza del pH e della sua determinazione 1

L'equilibrio chimico 2

Equilibri acido-base 4

Reazioni di neutralizzazione e titolazioni 6

Elettrodo a vetro 8

I potenziali degli elettrodi a vetro 10

La calibrazione e gli errori irrimediabili 12

Parte 2: Elettronica 15

Il sistema d'acquisizione dati 15

Arduino: come ridurre le difficoltà di progettazione di un SAD 16

Come interfacciarsi ad Arduino 17

Il circuito di condizionamento 19

Pulsanti, display e lettore per schede di memoria SD 21

Parte 3: Programmazione 23

Funzionamento generale 23

La lettura del segnale 24

Calibrazione polinomiale 25

Curva di titolazione 28

Allarme 29

Monitoraggio 29

Appendice A 31

Appendice B 33

Appendice C 43

Bibliografia 45

Siti utili 45

Parte 1

Chimica

L'importanza del pH e della sua determinazione

Il pH-metro è uno strumento che compie misure di pH in soluzioni, in genere, acquose. Le ragioni per cui si vuole misurare il pH possono essere svariate e riguardare ambiti industriali e sanitari. In generale, il pH:

- deve essere mantenuto entro certi intervalli nelle acque naturali per la salvaguardia dell'ambiente e, indirettamente, della salute umana, poiché è un parametro che influenza notevolmente le possibilità di sviluppo di comunità di esseri viventi, sia che essi in acqua vivano, sia che semplicemente la assumano;
- è importante per garantire la qualità dei prodotti e la sicurezza dei consumatori nei processi produttivi di carta, alimenti, bevande e in generale di industrie che fanno uso di biotecnologie;
- può essere utilizzato per determinare la concentrazione di specie chimiche attraverso titolazioni.

Ma cosa esprime il pH? E perché è così importante?

Il pH è definito secondo la seguente relazione:

$$pH = -\log([H^+])$$

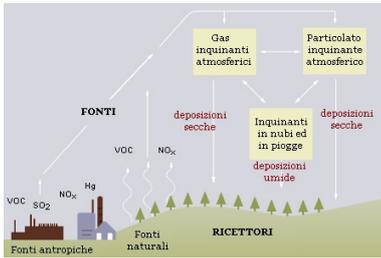
ovvero è il logaritmo in base 10 del reciproco della concentrazione di ioni H^+ in soluzione. L'importanza del pH nei campi industriali e sanitari è dovuta al fatto che esso è un modo di esprimere l'acidità o la basicità di una soluzione. Le soluzioni acide e basiche hanno la proprietà di poter essere coinvolte in reazioni (o più correttamente equilibri) acido-base. Queste reazioni avvengono anche naturalmente: si pensi che la grande presenza di ione bicarbonato nelle acque naturali è dovuta all'azione di erosione sulle rocce carbonatiche compiuta dall'anidride carbonica atmosferica, e che tale azione è proprio una reazione acido-base.

Il problema sorge quando le reazioni acido-base interferiscono con lo svolgimento dei processi vitali degli organismi viventi, ad esempio attraverso l'ingestione di un'acqua che, a causa di un pH basico o acido, è fortemente reattiva. Addirittura la semplice sopravvivenza, in ambienti acquatici con pH alterati, può risultare



Il pH negli acquari

Il pH ha un'influenza molto importante, ad esempio, sulla vita dei pesci teleostei. Essi possono essere soggetti all'effetto Root, per il quale una maggiore concentrazione di anidride carbonica o ioni H^+ nel sangue (quindi un minor pH) causa un abbassamento della capacità dell'emoglobina di trasportare ossigeno. Questo fenomeno è la ragione per cui è spesso necessario tenere monitorato il pH degli acquari.



Piogge acide

Lo schema mostra i processi coinvolti nella formazione di una pioggia acida. Le molecole determinanti, in questi fenomeni, sono quelle dei tipi SO_x e NO_x .

minacciata.

I fenomeni che portano alla variazione di pH nelle acque naturali sono vari. Relativamente alle attività umane, lo scarico di acque a pH non neutro, cioè né acido, né basico, è sicuramente uno dei più rilevanti. L'alterazione può essere causata dall'uomo anche indirettamente, con le piogge acide, che si formano con l'immissione in atmosfera di composti di zolfo e azoto del tipo SO_x ed NO_x .

Come per i processi vitali dei viventi, le reazioni acido-base possono essere indesiderate nei processi industriali. Per queste ragioni anche in questo campo vi è l'interesse ad avere acque con pH intorno a certi valori definiti. pH che esulano dalla neutralità possono altresì essere necessari alla produzione di determinati beni, come le batterie d'automobile o i prodotti per la pulizia casalinga.

Infine, in campo strettamente chimico, la misura del pH può essere utilizzata nelle titolazioni acido-base, nelle quali si determina la concentrazione di una specie chimica in soluzione modificando l'equilibrio acido-base con l'immissione di una soluzione standard. Il pH-metro che si vuole realizzare con questo progetto disporrà di particolari funzionalità per facilitare questo tipo di pratica di laboratorio.

Quelle elencate finora sono le ragioni per cui può essere necessario compiere determinazioni di pH e per cui, di conseguenza, può risultare utile un pH-metro. Prima di discutere la progettazione di questo strumento è bene chiarire tutti gli elementi di chimica relativi agli equilibri acido-base, in modo da avere un'idea dei processi che determinano l'acidità o la basicità di una soluzione, della relazione che il pH ha con questi parametri, dell'utilità delle titolazioni e del funzionamento di un elettrodo a vetro. Per introdurre questo discorso si inizierà richiamando il concetto di equilibrio chimico.

L'equilibrio chimico

Come già accennato, il pH è una grandezza che dipende dalla concentrazione di ioni H^+ in acqua. La presenza di questi ioni è determinata da un equilibrio chimico che sicuramente coinvolge l'autoionizzazione dell'acqua, poi in soluzioni non neutre anche la dissociazione di altre specie chimiche. Si tratterà in seguito dei concetti di dissociazione e di autoionizzazione, nonché di quali specie chimiche producano o consumino ioni H^+ ; la trattazione si concentrerà ora sulla disamina dei sistemi in equilibrio, indispensabile per capire il funzionamento di acidi e basi.

Il fatto che un sistema sia in equilibrio significa che in esso sono presenti sia i reagenti, sia i prodotti di una reazione, in quantità stabili. Ciò implica che la reazione non proceda fino al completo consumo dei reagenti (o almeno di quello limitante), ma si stabilizzi quando essi sono ancora tutti presenti, senza essersi esauriti. In queste situazioni si parla di equilibrio chimico e si può esprimere la rea-

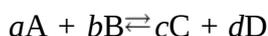
zione nel seguente modo:

reagenti \rightleftharpoons prodotti

È bene non farsi trarre in inganno dalla sensazione di staticità che evoca il concetto di equilibrio: la reazione chimica all'equilibrio non si arresta e le specie chimiche coinvolte non sono immobili e invariabilmente formate, bensì continuano a trasformarsi in entrambi i sensi della reazione. Accade, però, che la velocità con cui esse si trasformano in un senso sia uguale alla velocità con cui si trasformano nell'altro, e che subiscano perciò una variazione nulla in quantità.

Ogni reazione di equilibrio ha anche l'importante proprietà di avere una costante di equilibrio, che mette in relazione la quantità di reagenti con la quantità di prodotti. Grazie alla costante di equilibrio, si è quindi potenzialmente in grado di calcolare la quantità di varie specie chimiche coinvolte in un equilibrio, una volta note le quantità di altre specie.

Per una generica reazione:



la costante di equilibrio è così definita:

$$K = \frac{(a_c)^c \cdot (a_D)^d}{(a_A)^a \cdot (a_B)^b}$$

ed ha la proprietà di avere sempre lo stesso valore per quella reazione, posto però che essa avvenga alla stessa temperatura. La temperatura, infatti, può spostare l'equilibrio chimico: in generale, nelle reazioni esoenergetiche un aumento di temperatura sposta l'equilibrio verso i reagenti, mentre una diminuzione verso i prodotti; nelle reazioni endoenergetiche avviene l'inverso. Questi comportamenti sono una conseguenza del principio di Le Châtelier, secondo cui introducendo in un sistema in equilibrio una perturbazione, il sistema reagirà opponendosi a tale variazione.

Si noti anche che l'equazione della costante di equilibrio non si riferisce espressamente alla quantità o alla concentrazione in cui sono presenti le specie chimiche, bensì alla loro attività. L'attività è definita come la tendenza a reagire di una determinata specie chimica. È intuitivo come essa possa aumentare all'aumentare della concentrazione di un reagente: maggiore è la sua quantità, maggiore è la sua tendenza a reagire.

Di fatto, quando si ha a che fare con soluzioni e soluti, è possibile prendere come attività di una specie chimica la sua concentrazione, a patto che si trovi disciolta in una soluzione "a diluizione infinita" (più concretamente, poco concentrata). Questa semplificazione è possibile perché si possono trascurare tutti gli effetti che la presenza di altre specie può avere su una specie chimica. Tali effetti tendono, infatti, a crescere man mano che la concentrazione aumenta, rendendo la quantità di reagente effettivamente disponibile a reagire minore di quella complessiva. Si pensi, ad esempio, a soluzioni di ioni: ogni ione tende a circondarsi di cariche con segno opposto,



Il concetto di equilibrio

Se Paperino lucida le monete di Paperone alla stessa velocità con cui esse si ossidano, il sistema Deposito si può considerare in equilibrio.

compromettendo, a causa del "mascheramento" subito, la sua capacità di partecipare ad una reazione. Questo effetto si fa più marcato quanto maggiore è il numero di ioni presenti, mentre si fa sempre più trascurabile al suo diminuire.

Si potrà, allora, scrivere per soluzioni diluite:

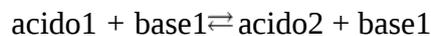
$$K = \frac{[C]^c \cdot [D]^d}{[A]^a \cdot [B]^b}$$

Appare chiaro, a questo punto, il senso di considerare gli ioni H^+ in equilibrio con le specie chimiche che li producono o che li consumano una volta poste in acqua: a seconda di come l'equilibrio è spostato nella reazione, gli ioni H^+ possono essere prodotti o consumati in misura maggiore o minore. Questi particolari equilibri sono proprio gli equilibri acido-base e la loro costante, come si vedrà, determina la forza di acidi e basi.

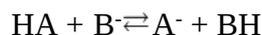
Equilibri acido-base

Secondo la teoria di Brønsted-Lowry, una reazione acido-base è una reazione di scambio di uno ione H^+ . Per definizione, l'acido è una specie chimica in grado di donare uno ione H^+ ad un'altra specie accettrice, la base. A seguito dello scambio, l'acido si trasforma nella base coniugata, diventando accettore, mentre la base si trasforma nell'acido coniugato, diventando donatrice.

Schematicamente, le reazioni acido-base sono descrivibili con la seguente equazione:



in cui sono coppie coniugate acido1/base1 e base2/acido2. Come esempio generale delle reazioni in cui avviene lo scambio di un solo ione H^+ , invece, si consideri la seguente reazione:



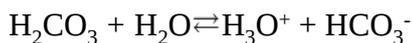
Come si è visto, il pH di una soluzione acquosa dipende dalla quantità di ioni H^+ in essa presenti. La particolarità dell'acqua è che essa è sia in grado di accettarne, sia in grado di donarne, e quindi può comportarsi da acido o da base a seconda dei casi. Il comportamento che assumerà in ogni caso dipenderà dalle specie chimiche con cui reagirà, che rispetto ad essa possono essere acide oppure basiche. Questo è dovuto anche al fatto che nella teoria di Bronsted-Lowry non esistano acidi e basi in senso assoluto, ma che più sottilmente una specie chimica si può classificare in uno dei due modi solo in relazione al ruolo che ha in una determinata reazione acido-base. Ciò avviene in particolare per sostanze come l'acqua, che si dicono anfotere per l'elevata interscambiabilità dei due ruoli.

Quando una specie acida o basica immette o assorbe ioni H^+ in acqua, si è in presenza di una dissociazione: le reazioni che producono ioni H^+ si dicono dissociazioni acide, mentre quelle che ne causano la diminuzione sono dissociazioni basiche. Per esempio, in

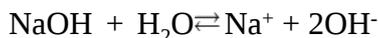
Teorie su acidi e basi

La teoria di Brønsted-Lowry è uno dei tanti modelli con cui descrivere acidi e basi. Tra gli altri, e più importanti, il più moderno è la teoria di Lewis, che individua l'acido come una sostanza in grado di accettare un doppietto elettronico da un'altra sostanza, detta base; il più antico è il modello di Arrhenius, secondo cui un acido è una sostanza che, dissociandosi in acqua, libera ioni H^+ , mentre una base ioni OH^- .

questo caso:

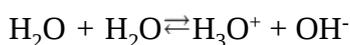


si ha dissociazione acida. In questo caso, invece:



si ha dissociazione basica. Si possono considerare gli ioni OH^- , infatti, come in grado di strappare uno ione H^+ dall'acqua, inglobandolo in una nuova molecola d'acqua.

Questo discorso, però, ha come conseguenza che in acqua siano sempre presenti degli ioni H^+ e OH^- e che essa non sia esattamente sempre in forma molecolare. Effettivamente è così: l'acqua presenta il fenomeno dell'autoionizzazione, che avviene secondo il seguente equilibrio:



Tale reazione è una vera e propria reazione acido-base, in cui ciascuna delle due molecole d'acqua è rispettivamente acido e base (questo sempre grazie alla proprietà dell'acqua di essere anfotera). La sua costante di equilibrio è K_w ("kappa water", prodotto ionico dell'acqua), che a 25°C vale 10^{-14} , la quale fa sì che il prodotto tra le concentrazioni di ioni H^+ (che sono quelli della molecola H_3O^+) e ioni OH^- sia sempre costante. Ciò ha come conseguenza che all'aumentare della quantità di uno dei due ioni, la quantità dell'altro diminuisca, e questo spiega perché la dissociazione basica consumi ioni H^+ e, specularmente, la dissociazione acida consumi ioni OH^- .

Le dissociazioni acide e basiche intervengono su questo equilibrio modificando le concentrazioni delle due specie ioniche, che alla neutralità sono ciascuna uguale a 10^{-7} M. Il pH vuole allora essere un parametro che semplifichi l'espressione della concentrazione di ioni H^+ evitando l'impiego della notazione scientifica e di esponenti negativi. Esso, si ricorda, è così definito:

$$pH = -\log([H^+])$$

Alla neutralità, poiché gli ioni H^+ hanno concentrazione 10^{-7} M, il pH è 7. Per valori più vicini al massimo di 14, la concentrazione degli ioni H^+ è più piccola di quella alla neutralità e si ritiene basica la soluzione acquosa; per valori più vicini al minimo di 0, la concentrazione degli ioni H^+ è più grande di quella alla neutralità e la soluzione si ritiene acida.

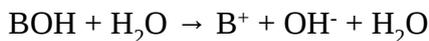
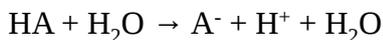
La tendenza di un acido o di una base a dissociarsi e a modificare il pH dipende dalla costante di equilibrio che regola tale reazione. Per acidi e basi "forti", la dissociazione si considera completa e viene trascurato l'equilibrio; per acidi e basi "deboli", invece, la costante di equilibrio è importante per determinare quanto il reagente è in grado di modificare il pH di un'acqua.

H^+ o H_3O^+ ?

Di fatto lo ione H^+ che viene scambiato nell'equilibrio va a legarsi alla molecola H_2O , formando lo ione H_3O^+ . Scrivere H^+ significa considerarlo separatamente, ma ci si riferisce sempre alla stessa specie chimica.

Reazioni di neutralizzazione e titolazioni

Nella pratica di laboratorio si ha a che fare spesso con miscele di acidi e basi. Gli effetti di produzione e consumo di ioni H^+ di queste specie chimiche, allora, si compensano, e il pH risulta influenzato solo dalla specie in eccesso, se presente. Si provi a sommare gli effetti di una dissociazione acida e di una dissociazione basica:



Le reazioni che portano a questa compensazione si dicono di "neutralizzazione". È importante ricordare che, indipendentemente dagli equilibri di dissociazione di acidi e basi, le neutralizzazioni si considerano, in genere, complete. Al consumo dei prodotti di una dissociazione, dovuto ad una neutralizzazione, per il principio di Le Châtelier l'equilibrio compenserà la perturbazione dissociando ulteriormente, finché la presenza di reagenti lo permette. Questo avviene soprattutto per neutralizzazioni di acidi o basi deboli con, rispettivamente, basi o acidi forti, mentre in caso di acidi forti con basi forti il discorso è inutile, essendo che si dissociano completamente. Ciò che cambia nei due casi, semmai, è il pH a neutralizzazione avvenuta, al cosiddetto punto di equivalenza: per titolazioni acido-base forti è neutro, mentre in caso di acidi deboli con basi forti sarà basico, a causa dell'idrolisi basica della base coniugata dell'acido debole (l'effetto, come sempre, è speculare per il caso acido forte-base debole).

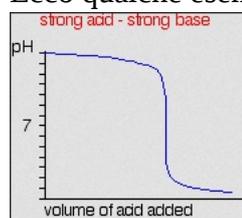
La peculiarità delle reazioni di neutralizzazione è che esse possono essere utilizzate, impiegando una soluzione standard a concentrazione nota, per determinare le concentrazioni di soluzioni in cui sono disciolti acidi o basi, attraverso pratiche di laboratorio chiamate "titolazioni".

Nelle titolazioni, si chiamano "titolante" la soluzione standard utilizzata per la determinazione, "titolato" la soluzione a concentrazione incognita. Si procede immettendo progressivamente piccole quantità di standard nel titolato finché non si arriva al viraggio, ovvero al punto in cui l'influenza sul pH della soluzione è data solo dal titolante. I prodotti della dissociazione della specie chimica incognita (H^+ o OH^-), infatti, sono stati completamente consumati e l'unica influenza sul pH della soluzione è ora quella del titolante in eccesso. A questo punto, si dispone del volume di titolante utilizzato per arrivare al viraggio e della sua concentrazione: conoscendo la stechiometria della reazione di neutralizzazione, si può calcolare la concentrazione della specie chimica dissociata nella soluzione.

Le titolazioni danno luogo a particolari curve che relazionano il pH col volume di titolante aggiunto, chiamate "curve di titolazione". Esse sono in grado di ben evidenziare il salto in termini di pH che compiono le soluzioni, una volta neutralizzate. È proprio in que-

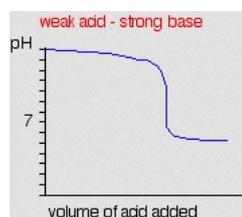
st'area del grafico che si individua il punto di equivalenza, ovvero il punto della titolazione in cui l'influenza sul pH di titolante e titolato è uguale: qui, infatti, non vi è eccesso né di acido, né di base.

Ecco qualche esempio di curva di titolazione:



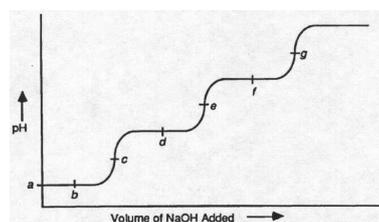
Titolazione acido forte-base forte

Nelle titolazioni con un acido ed una base forte, si ha una dissociazione completa di entrambi i reagenti. Si noti come in questo caso il punto di equivalenza corrisponda alla neutralità, ovvero pH 7.



Titolazione acido debole-base forte

Nelle titolazioni con un acido debole ed una base forte (e viceversa), il punto di equivalenza non è alla neutralità, ma a pH basico. La ragione di questo fatto è da ricercarsi nell'idrolisi basica data dalla base coniugata dell'acido debole, che diventa preponderante quando gli ioni H^+ dell'acido sono esauriti.



Titolazione di acidi poliprotici

Quando si titola un acido poliprotico, si può ben osservare nella curva il maggior numero di dissociazioni. Nel caso degli acidi diprotici, per esempio, la seconda dissociazione ha una forza molto minore della prima. Perché essa possa avvenire, deve innanzitutto esaurirsi la prima attraverso la titolazione, che consuma tutti gli ioni H^+ dell'acido indissociato; la seconda dissociazione si avrà per compensare la carenza di H^+ successiva all'esaurimento. Da qui l'andamento particolare delle curve di titolazione di acidi poliprotici.

Per seguire l'andamento di una titolazione, si può ricorrere a vari metodi. Uno dei più diffusi è l'utilizzo di un indicatore, ovvero una sostanza che, disciolta in soluzione, fornisce colorazioni diverse a seconda del pH a cui essa si trova. Nel corso di una titolazione, quindi, l'indicatore fornirà una certa colorazione nelle condizioni iniziali, un'altra dopo il viraggio. Il volume di titolante che si considera necessario alla neutralizzazione, nel caso generale, è quello immesso complessivamente quando, ad un'altra minima aggiunta, la soluzione avrà cambiato completamente colore (viraggio completo).

Si possono, allo stesso modo, condurre titolazioni potenziometriche, che invece prevedono l'utilizzo di un pHmetro. Questo è proprio il caso in cui si tracciano le curve di titolazione, dalle quali si può ricavare il volume di titolante aggiunto per arrivare al punto di neutralizzazione. Esso coincide con il punto di flesso: se si presta

attenzione all'andamento di queste curve, si noterà come il valore assoluto della loro derivata tenda inizialmente ad aumentare sempre di più, finché, al viraggio, non inizia bruscamente a diminuire. Il punto di flesso è il punto dove il valore assoluto della derivata è massima, o se si preferisce, dove la derivata seconda cambia di segno.

Il senso chimico di quest'astrazione matematica è piuttosto immediato: man mano che aumenta la presenza di titolante in soluzione, la sua influenza sul pH diventerà sempre maggiore, a causa della neutralizzazione dei prodotti di dissociazione della specie chimica da determinare. Il tasso di variazione del pH tenderà così all'aumento, che sarà massimo alla completa neutralizzazione, e porterà il valore del pH a essere influenzato solo dalla presenza di titolante. Il punto di equivalenza si trova, allora, proprio dove il tasso di variazione è massimo, cioè al punto di flesso. Procedendo ulteriormente si osserverà come le aggiunte siano sempre meno influenti, poiché il titolante, per quanto forte sia, dissocia sempre meno a causa della già grande presenza di suoi prodotti (si ricordi ancora una volta Le Châtelier).

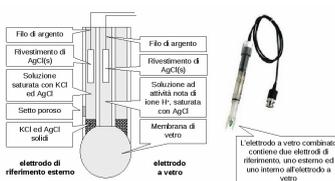
Le titolazioni potenziometriche hanno la qualità di fornire dati più attendibili rispetto a quelli forniti dalle titolazioni che usano indicatori chimici, e sono inevitabili quando si devono eseguire titolazioni in soluzioni colorate o torbide e per ricercare la presenza di specie insospettite. Anche se più lente delle titolazioni effettuate con i comuni indicatori, le titolazioni potenziometriche sono facilmente automatizzabili.

Elettrodo a vetro

Per effettuare titolazioni o determinare il pH di soluzioni si usano i piaccametri, punto focale di questa trattazione. Se l'esperienza che si vuole intraprendere si concentrerà maggiormente sulla progettazione della parte elettronica del pH-metro, qualche cenno sul funzionamento degli elettrodi a vetro, che compiono la funzione di trasduttore nel circuito che si andrà a realizzare, è comunque necessario.

Gli elettrodi a vetro sono gli strumenti di laboratorio che permettono di compiere misure di pH e sono parte integrante dei piaccametri. Il pH è misurato attraverso la differenza di potenziale che si crea tra una soluzione incognita e una soluzione di riferimento dell'elettrodo a vetro; la restante parte circuitale del pH-metro si occupa di convertire in unità di pH il segnale elettrico, renderlo leggibile all'operatore ed eventualmente fornire funzionalità che possano facilitare certe metodiche.

Un elettrodo a vetro è costituito da una sottile membrana (circa 0,1 mm) di vetro speciale, che è l'elemento dell'elettrodo effettivamente sensibile al pH, saldata all'estremità di un tubo resistente in plastica o vetro. La membrana di vetro si trova a contatto con due soluzioni:



L'elettrodo a vetro

Lo schema evidenzia gli aspetti costruttivi che accomunano la maggior parte degli elettrodi a vetro disponibili.

la soluzione della parte interna è una soluzione tampone ed ha pH costante, la soluzione esterna ha pH incognito ed è quella di cui si vuole determinare la concentrazione di ioni H^+ . Sulle due facce della membrana, tra le quali c'è una resistenza che può variare da 50 a 500 $M\Omega$, si crea una differenza di potenziale E_m dipendente solamente dalla differenza di pH delle due soluzioni ad esse affacciate. Dato che una delle soluzioni ha pH costante, E_m dipenderà solamente dal pH della soluzione esterna.

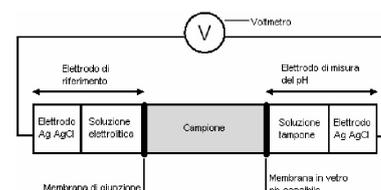
Il dato determinante per la misura del pH, quindi, è proprio E_m , il quale si può misurare con l'impiego di due elettrodi che, chiudendo il circuito, trasferiscano la differenza di potenziale ad un altro circuito che lo trasformi in dato leggibile dall'operatore. Uno dei due elettrodi è immerso nella soluzione tampone a contatto con la parte interna della membrana sensibile; l'altro elettrodo è situato in una soluzione isolata da quella interna, ed è collegato attraverso un setto poroso alla soluzione a pH incognito. È chiaro che, in questo modo, sull'elettrodo interno vi sarà il potenziale interno alla membrana, mentre sull'elettrodo esterno vi sarà il potenziale esterno alla membrana. Per far sì che sia possibile la conduzione, sia la soluzione interna, sia la soluzione esterna sono saturate di $AgCl$, il quale riveste anche i due elettrodi, di argento. Le reazioni di scambio elettronico tra Ag e $AgCl$, di ossidoriduzione, permettono il trasferimento del potenziale elettrico sugli elettrodi. Il potenziale che viene misurato è allora:

$$E_{vetro} = E_m + E_{Ag/AgCl} + E_k$$

in cui risulta incognito e variabile solo E_m . Il termine E_k è detto potenziale di asimmetria. La ragione di questo termine è legata al fatto che E_m risulta leggermente diverso da zero anche quando le concentrazioni interna ed esterna hanno lo stesso valore, perché non è possibile una perfetta equivalenza tra le superfici interna ed esterna della membrana. E_k è un parametro costruttivo dell'elettrodo e varia con il tempo e con l'usura.

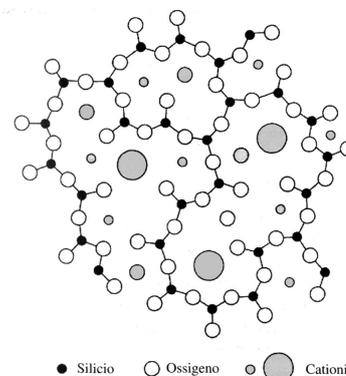
Nella maggior parte degli elettrodi a vetro oggi disponibili, anche l'elettrodo esterno è incorporato, e collegato ad un cavo coassiale assieme all'elettrodo interno. In questo caso si parla di "elettrodo a vetro combinato".

Un breve accenno, infine, al funzionamento della membrana sensibile, il cui fenomeno non è ancora del tutto chiarito: il vetro silicato usato per le membrane è costituito da un insieme infinito tridimensionale di tetraedri SiO_4^{4-} , in cui ogni atomo di ossigeno è condiviso fra due atomi di silicio. Le cariche negative sono bilanciate da cationi (es. Na^+ e Li^+), contenuti negli interstizi del reticolo. La membrana di vetro risponde al pH in quanto è in grado di scambiare ioni H^+ con le soluzioni con cui è a contatto: in questo processo i cationi Na^+ e Li^+ vengono scambiati con gli ioni H^+ della soluzione, con conseguente formazione di uno strato superficiale di "gel". Gli ioni H^+ entrati nel vetro si legano agli ioni O negativi formando gruppi OH (neutri). Dato che escono ioni positivi e



L'elettrodo a vetro come un "circuito elettrico"

L'elettrodo a vetro è di fatto un "circuito elettrico": la faccia della membrana a contatto con la soluzione campione è la massa, quella interna all'elettrodo invece porta il segnale. Entrambi i potenziali sono messi in comunicazione con un "voltmetro" attraverso soluzioni elettrolitiche ed elettrodi.



Il vetro silicato

Una rappresentazione della distribuzione di tetraedri SiO_4^{4-} e delle cariche positive in essa disposte. Preso complessivamente, l'insieme è neutro.



La membrana sensibile

Come si nota, gli ioni H^+ penetrano nei due sottili strati idratati, mentre lo strato centrale rimane anidro, assicurando comunque il contatto elettrico tra le due soluzioni grazie ai movimenti degli ioni più piccoli (Na^+ o Li^+) presenti nel vetro, che si spostano tra le lacune anioniche che caratterizzano la sua struttura interna. Gli H^+ quindi non attraversano la membrana di vetro; ciò è stato dimostrato usando una delle due soluzioni marcata con ioni trizio ($^3H^+$) radioattivi: l'altra non diventa radioattiva.

vengono sostituiti da gruppi neutri, i due strati di gel idratato si caricano negativamente, con un'entità di carica dipendente dalla concentrazione degli ioni H^+ .

Affinché tutto ciò che è stato descritto accada, il vetro deve essere idratato, e per questo motivo un elettrodo a vetro va mantenuto sempre immerso in una soluzione acquosa di composizione analoga all'elettrolita presente al suo interno; se tale soluzione andasse a secco, al riutilizzo dell'elettrodo occorrerebbe attendere molto tempo (qualche ora) prima che il sistema si stabilizzasse e avesse i requisiti per eseguire delle misurazioni.

I potenziali degli elettrodi a vetro

Una volta acquisite tutte le nozioni di base sul pH di soluzioni acquose e sulle modalità di misurarlo, arriva il momento di capire cosa dell'aspetto chimico è utile alla risoluzione del problema elettronico di progettazione. In particolare, bisogna entrare nello specifico e studiare quali sono le entità delle differenze di potenziale fornite da un elettrodo a vetro, come esse varino assieme al pH e quali fattori possano influenzarle.

Innanzitutto si può osservare che il potenziale sulla membrana E_m si può esprimere con l'equazione di Nernst:

$$E_{\text{vetro}} = E_k + E_{Ag/AgCl} + E_m = E_k + E_{Ag/AgCl} + 2,3025 \frac{RT}{F} \log \left(\frac{[H^+]_{\text{int}}}{[H^+]_{\text{est}}} \right)$$

dove R è la costante universale dei gas, T la temperatura in scala assoluta e F la costante di Faraday. Si nota come il potenziale aumenti se la soluzione esterna è più basica di quella interna (meno H^+ e risultato del logaritmo positivo), diminuisca se più acida (più H^+ risultato del logaritmo negativo). Questa equazione è molto comune in elettrochimica, specie per il calcolo del potenziale di elettrodi nelle celle, e può essere adattata, come nella versione appena riportata, all'espressione dei potenziali generati dalle differenze di attività tra ioni H^+ .

Siccome si vuole trovare una relazione tra pH e differenza di potenziale, si può sfruttare il logaritmo presente nell'espressione:

$$\begin{aligned} E_{\text{vetro}} &= E_k + E_{Ag/AgCl} - 2,3025 \frac{RT}{F} [\log(H^+)_{\text{int}} - \log(H^+)_{\text{est}}] = \\ &= E_k + E_{Ag/AgCl} - 2,3025 \frac{RT}{F} (-pH_{\text{int}} + pH_{\text{est}}) \end{aligned}$$

In una forma semplificata, che raggruppi nel termine E_0 tutte le costanti, allora:

$$E_{\text{vetro}} = E_0 - \frac{RT}{F} (pH_{\text{est}})$$

con

$$E_0 = E_k + E_{Ag/AgCl} + 2,3025 \frac{RT}{F} pH_{\text{int}}$$

Trovata questa relazione, si possono evidenziare due considerazioni molto importanti per la misurazione del pH: la prima è che la differenza di potenziale inviata dall'elettrodo a vetro varia linearmente con il pH; ciò sarà fondamentale nella progettazione del circuito elettronico e semplificherà di non poco le cose. La seconda considerazione è che, a parità di pH, a diverse temperature il potenziale sulla membrana è diverso. Se esso fosse sempre costante, in tutti i piaccametri si saprebbe sempre a quale pH corrisponde un certo potenziale, mentre, in realtà, è necessario tenere conto dell'influenza della temperatura.

Questa influenza della temperatura sul potenziale generato non è da confondere con l'influenza che essa ha sull'attività degli ioni H^+ . Si è visto, infatti, come per Le Châtelier la temperatura sia influente sugli equilibri chimici: in effetti, l'autoionizzazione dell'acqua e le dissociazioni delle specie chimiche in essa presenti risultano influenzate dalla temperatura, senza contare che lo stato di agitazione del sistema ha un ruolo determinante anche sull'attività ionica. Tutte queste variabili rendono l'effetto della temperatura sul pH caratteristico di ogni soluzione.

Ciò che risulta importante ai fini progettuali, comunque, è la variazione del potenziale generato in base alla temperatura, mentre è giusto leggere pH diversi a diverse temperature. Il pH-metro dovrà, perciò, saper tradurre i potenziali in pH con riguardo della temperatura.

Per avere un'idea dei segnali inviati dall'elettrodo a vetro, si consideri che a 25°C:

$$E_{\text{vetro}} = E_0 - 0,05916 \cdot pH$$

la variazione del potenziale è di 59,16 mV per unità di pH. Trascurando, nel parametro E_0 , il potenziale di asimmetria e quello della reazione di scambio elettronico, e considerandolo sempre come:

$$E_0 = 2,3025 \frac{RT}{F} pH_{\text{int}} = (0,05916 \cdot 7) V = 0,414 V$$

si ha che a pH 7 il segnale è 0 V, perché la somma algebrica dei due termini che danno E_{vetro} è nulla. Questo avviene perché pH_{int} deve essere uguale a 7 per poter eseguire misurazioni su una scala più ampia possibile di pH; quando tale non è, si risolve il problema con la calibrazione (se ne discuterà in seguito).

Si possono, allora, calcolare gli estremi in termini di tensione, a cui la scala del pH può portare nell'elettrodo a vetro. A pH 14:

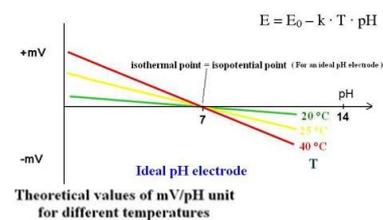
$$E_{\text{vetro}} = 0,414 V - 0,05916 \cdot 14 = -0,414 V$$

A pH 0, invece:

$$E_{\text{vetro}} = 0,414 V - 0,05916 \cdot 0 = +0,414 V$$

Si considereranno, allora, un massimo di 414 mV per pH acidi e un minimo di -414 mV per pH basici.

È bene sottolineare che la linearità tra pH e segnale in tensione inizia a non valere più a pH molto lontani dalla neutralità. Si parla di



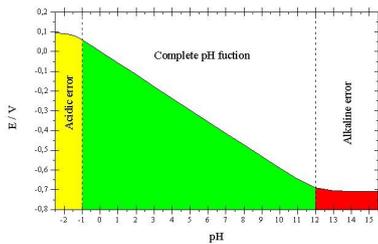
Theoretical values of mV/pH unit for different temperatures

$$E = E_0 - k \cdot T \cdot pH$$

- 0 °C = -54.2 mV/ per pH unit
- 20 °C = -58.2 mV/ per pH unit
- 25 °C = -59.2 mV/ per pH unit
- 50 °C = -64.1 mV/ per pH unit
- 75 °C = -69.1 mV/ per pH unit

La dipendenza della relazione pH-tensione dalla temperatura

Secondo la legge di Nernst, la temperatura influenza la dipendenza della tensione sulla membrana sensibile dal pH. Tarare e lavorare a temperatura costante, oppure implementare nel pH-metro una correzione elettronica in base alla temperatura possono essere soluzioni a questo fenomeno.



Errore di lettura

A pH estremi, viene a perdersi la linearità tra tensione sulla membrana sensibile e pH, con tanto di effetti indesiderati sulle misurazioni degli elettrodi a vetro.

“errore nel range alcalino” per pH alti, in genere maggiori di 12, quando i contributi dei metalli alcalini (come litio, sodio e potassio) sono comparabili a quelli degli ioni H⁺; si parla di “errore nel range acido” a pH molto bassi, quando le concentrazioni di ioni H⁺ sono così alte da risultare più difficilmente approssimabili con l'attività, la quale inizia a subire variazioni meno apprezzabili all'aumentare del pH. È necessario curarsi di questi errori, comunque, solo in casi estremi, mentre nelle comuni pratiche di laboratorio, come le titolazioni potenziometriche, non è generalmente richiesto considerarli.

La calibrazione e gli errori irrimediabili

Nella trattazione dell'elettrodo a vetro condotta finora, si sono individuate diverse potenziali cause d'errore nella misurazione del pH: se la linearità tra tensione e pH è un'indubbia comodità, non si possono comunque trascurare i problemi del potenziale di asimmetria e della dipendenza del potenziale per unità di pH dalla temperatura.

Per porre rimedio a queste cause d'incertezza, i pH-metri vengono sempre tarati prima dell'uso. La taratura agisce sulla circuiteria a cui gli elettrodi a vetro sono collegati, in modo da introdurre un offset che bilanci il potenziale di asimmetria portandolo a zero, e da modificare il guadagno del circuito per adattarsi alla temperatura. Per effettuare queste operazioni, si immerge l'elettrodo a vetro in appositi standard, in modo da sapere quali valori di pH bisogna raggiungere nella regolazione. È importante che il fissaggio del guadagno avvenga alla stessa temperatura a cui il pH dello standard usato è stato definito.

Ci sono, ad ogni modo, cause d'errore che non possono essere corrette, poiché intrinseche al funzionamento stesso dell'elettrodo a vetro: esse sono causate dal potenziale di giunzione. Il potenziale di giunzione è una piccola differenza di potenziale che si produce ogni volta che vengono a contatto soluzioni a differente composizione, come all'estremità di un ponte salino o in corrispondenza del setto poroso di un elettrodo. Si origina a causa della diversa mobilità degli ioni. Ad esempio, all'interfase fra una soluzione contenente NaCl e l'acqua, si crea una regione anteriore con un eccesso di carica negativa, ricca di ioni Cl⁻, seguita da una regione carica positivamente, impoverita di ioni Cl⁻. Questo perché lo ione Cl⁻ ha una mobilità maggiore dello ione Na⁺. Il risultato è una differenza di potenziale elettrico che contribuisce al potenziale complessivo della cella elettrochimica.



Il potenziale di giunzione

La diversa mobilità degli ioni fa sì che essi finiscano per disporsi in maniera non omogenea e che si creino delle differenze di potenziale. Esse hanno un'irrimediabile ripercussione sul potenziale complessivo in uscita dall'elettrodo.

Il potenziale di giunzione può essere ridotto utilizzando nel ponte salino un elettrolita ad elevata concentrazione e nel quale catione ed anione abbiano approssimativamente la stessa mobilità (KCl è spesso utilizzato nei ponti salini appunto perché K⁺ e Cl⁻ hanno circa la stessa mobilità).

In genere il potenziale di giunzione è relativamente basso (pochi

mV) è può essere trascurato. Nondimeno, dato che il potenziale di giunzione non è noto con precisione, esso pone un limite all'accuratezza delle misure potenziometriche, in particolare quelle dirette.

Parte 2

Elettronica

Si è visto finora cosa è il pH, quali fenomeni danno origine a sue modificazioni e come lo si può misurare. Le parti successive della trattazione si concentreranno su come poter progettare un circuito che, interfacciandosi con un elettrodo a vetro, possa fornire una lettura del pH e varie altre funzionalità. L'esposizione prenderà come riferimento la costruzione del pH-metro condotta dall'autore all'ITIS Leonardo da Vinci di Carpi (Modena) nell'anno scolastico 2012/2013.

Il Sistema d'Acquisizione Dati

La progettazione di un pH-metro richiede la realizzazione di un Sistema di Acquisizione Dati (SAD), ovvero un circuito in grado di monitorare, trasmettere ed elaborare una grandezza fisica in modo da renderla leggibile ad un operatore. Un Sistema d'Acquisizione Dati è un sistema costituito da un certo numero di canali di conversione analogico-digitale che interfacciano sensori e trasduttori ad un sistema di memorizzazione ed elaborazione. Il dispositivo d'elaborazione assicura la gestione del sistema e dei dati acquisiti, eseguendo particolari compiti come la visualizzazione dei segnali, l'archiviazione e l'elaborazione dei dati.

I SAD consentono l'acquisizione di segnali analogici mediante le seguenti fasi operative:

- condizionamento;
- campionamento;
- conversione.

I canali d'ingresso prevedono opportuni blocchi di condizionamento del segnale, quali attenuatori, amplificatori, filtri, convertitori corrente-tensione, amplificatori d'isolamento.

In un SAD sono spesso inseriti anche dei convertitori D/A per fornire in uscita dei segnali analogici con i quali pilotare attuatori, plotter, ecc.

Il funzionamento dei SAD, allora, si può schematizzare nelle se-



Schema di un SAD

guenti fasi:

- grandezza fisica;
- trasduttore;
- circuito di condizionamento;
- filtro;
- sample & hold;
- ADC;
- elaboratore.

In questo schema, la grandezza fisica è la grandezza che si vuole quantificare, mentre il trasduttore lo strumento che permette di tradurla in segnale elettrico. Come si è visto in precedenza, l'elettrodo a vetro è in grado di fornire un segnale in tensione che varia linearmente con il pH; di conseguenza se il pH è la grandezza che si vuole misurare, l'elettrodo a vetro è il vero e proprio trasduttore. Il circuito di condizionamento e il filtro sono, invece, fasi in cui il segnale subisce delle modifiche: esse servono a farlo rientrare in un range in cui l'ADC, cioè il convertitore analogico-digitale, possa effettuare una conversione. Gli ADC sono circuiti in grado di fornire parole binarie dato un certo segnale analogico, e lavorano in intervalli di tensione caratteristici, all'infuori dei quali danno risposte non significative. La conversione analogico-digitale è molto importante per avere un segnale facilmente processabile da un elaboratore, così da poter essere letto da un operatore o utilizzato per comandare altri dispositivi. Il filtro, infine, è più specificamente una parte del circuito in grado di eliminare i disturbi, mentre il sample & hold si occupa di trattenere il segnale analogico per il tempo necessario all'ADC per effettuare la conversione.

Arduino: come ridurre le difficoltà della progettazione di un SAD

Il progetto esposto in questo fascicolo vuole essere tanto didattico quanto facilmente accessibile. Non è, quindi, previsto l'impiego di conoscenze particolarmente approfondite di chimica, elettronica e matematica, preferendo semplificare il più possibile. La progettazione di un Sistema di Acquisizione Dati può diventare laboriosa nell'implementazione di tutti i suoi stadi, ma fortunatamente esistono sul mercato soluzioni che permettono di bypassare diversi suoi aspetti. Sicuramente ciò comporterà una maggiore ignoranza del funzionamento del circuito elettronico, ma permetterà una didattica applicativa e “a piccoli passi”, che aggiunge gradualmente nuovi aspetti al problema invece di costringere alla loro risoluzione in una sola volta. Del resto l'approfondimento è automatico con l'aumentare della complessità dei problemi da risolvere; se anche all'inizio sono nascosti al progettista apprendista alcuni meccanismi di funzionamento, egli avrà comunque imparato qualcosa nel realizzare il suo progetto e, nel realizzarne altri, si avvicinerà sempre di più alla comprensione dei meccanismi a più basso livello.



Una board Arduino Mega

Arduino Mega è la stessa board utilizzata per il progetto qui esposto. Il suo punto di forza è l'economicità, unita ad una grande disponibilità di pin analogici e digitali e ad un microcontroller con buone prestazioni.

Per questa serie di motivi la progettazione del pH-metro si servirà della piattaforma di prototipazione Arduino. Arduino ha conosciuto un notevole successo tra hobbisti e in ambienti didattici, grazie alla sua caratteristica di riuscire a facilitare l'apprendimento veloce dei principi fondamentali dell'elettronica e della programmazione. Tutto questo, appunto, grazie alla sua proprietà di evitare al progettista di ideare diverse parti del circuito: di fatto il funzionamento di Arduino si può semplificare, banalmente, alla lettura di segnali, analogici o digitali, e all'assunzione di un certo comportamento di conseguenza, modellato attraverso un programma. Ciò che è implementato in un solo componente hardware sono tutte le parti di lettura del segnale, come l'ADC e il sample & hold, e della sua elaborazione.

Queste funzionalità sono possibili grazie alla struttura di Arduino: esso è composto da un circuito stampato che integra un microcontrollore con pin connessi alle porte di input e output, un regolatore di tensione e quando necessario un'interfaccia USB che permette la comunicazione con il computer. A questo hardware viene affiancato un ambiente di sviluppo integrato multiplatforma (per Linux, Apple Macintosh e Windows). Questo software permette anche ai novizi di scrivere programmi con un linguaggio semplice e intuitivo derivato da C e C++ chiamato Wiring, liberamente scaricabile e modificabile.

Il successo di Arduino è probabilmente dovuto anche al fatto che le informazioni sull'hardware e soprattutto i progetti sono disponibili per chiunque: si tratta quindi di un hardware open source, distribuito nei termini della licenza Creative Commons Attribution-ShareAlike 2.5. In questo modo, chi lo desidera può legalmente auto-costruirsi un clone di Arduino o derivarne una versione modificata, scaricando gratuitamente lo schema elettrico e l'elenco dei componenti elettronici necessari.

Come interfacciarsi ad Arduino

Per implementare il comportamento interattivo, Arduino è fornito di funzionalità di input/output (I/O), grazie alle quali esso riceve i segnali raccolti da sensori esterni. In base a tali valori, il comportamento della scheda è gestito dal microcontroller, che attua le decisioni determinate dal particolare programma in esecuzione in quel momento sulla scheda.

Per ricevere e inviare segnali, Arduino è dotato di pin I/O collegati a connettori femmina, a cui si possono collegare facilmente fili ed altri connettori. L'I/O è sia digitale, sia analogico: i pin digitali possono essere utilizzati dal microcontroller per fornire segnali digitali oppure leggerli, mentre i pin analogici sono connessi ad un ADC che fornisce al microcontroller un livello discreto (tipicamente da 0 a 1023).

L'ambiente di sviluppo integrato (IDE) di Arduino, che permette di

scrivere il programma da eseguire sulla scheda, è un'applicazione multiplatforma scritta in Java, ed è derivata dall'IDE creato per il linguaggio di programmazione Processing e per il progetto Wiring. È concepita per iniziare alla programmazione artisti e altri neofiti, che siano a digiuno di pratica nello sviluppo di software. Per permettere la stesura del codice sorgente, l'IDE include un editor di testo dotato inoltre di alcune particolarità, come il `syntax highlighting`, il controllo delle parentesi, e l'indentazione automatica. L'editor è inoltre in grado di compilare e lanciare il programma eseguibile in una sola passata e con un solo click.

Come utilizzare tutti questi elementi al fine di progettare il pHmetro?

Si potrebbe inviare il segnale in tensione dell'elettrodo a vetro a un pin analogico, convertirlo in unità di pH (operazione concettualmente molto semplice, poiché come si è visto la tensione fornita dall'elettrodo varia linearmente col pH) e rendere leggibile il pH all'operatore attraverso un display (esistono librerie con cui comandare vari display LCD presenti sul mercato molto semplicemente con Arduino). Il programma caricato nel microcontroller potrebbe, poi, fornire diverse funzionalità aggiuntive comode in laboratorio, ma di questo si discuterà nella terza parte.

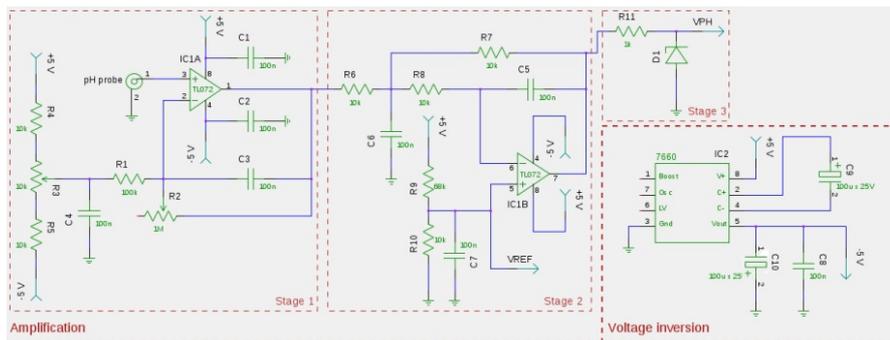
Si può schematizzare il progetto del pHmetro, allora, col seguente diagramma:



Per quanto semplice possa sembrare quanto esposto, purtroppo in realtà il problema è più complicato. Si è detto in precedenza che Arduino assolve alle funzioni di campionamento ed elaborazione dei segnali; se si è prestata un po' di attenzione, ci si sarà accorti che non è stata considerata la parte di condizionamento del segnale presentata nello schema sul funzionamento dei SAD.

In effetti, non si può inviare ad Arduino qualsiasi segnale analogico. La scheda è dotata di un ADC in grado di convertire tensioni limitatamente a certi range, che vanno da 0 V a una tensione positiva di riferimento variabile e non maggiore della tensione con cui è alimentato il microcontroller, cioè 5 V. Se si considera che l'elettrodo a vetro fornisce tipicamente segnali da -414 mV a +414 mV, ci si renderà conto che la conversione sarà tutt'altro che efficace. Ciò che si deve fare è trasformare il segnale dell'elettrodo a vetro in un segnale che sia convertito sfruttando tutta la risoluzione dell'ADC di Arduino, e ciò avverrà con un circuito di condizionamento.

Il circuito di condizionamento



Si propone qui un circuito di condizionamento che porti i segnali ad essere compresi nel range accettato dall'ADC di Arduino. Esso è formato da tre stadi: in breve, il primo stadio annulla l'offset del trasduttore e amplifica con un certo guadagno i segnali; il secondo stadio eleva le tensioni ottenute nel primo stadio in modo che siano tutte positive; il terzo stadio protegge l'ADC da tensioni negative o superiori a 5 V, poiché in grado di danneggiarlo.

Si può osservare che il primo stadio è configurato per poter eseguire la calibrazione: il trimmer R_3 è utilizzato per regolare la tensione di offset, con cui annullare il potenziale di asimmetria e tutti i potenziali indesiderati che impediscono all'elettrodo a vetro di fornire 0 V a pH 7; il trimmer R_2 , invece, permette di regolare il guadagno per adattare le tensioni in uscita dall'amplificatore operazionale alla temperatura, che si è visto influenzare la risposta in tensione dell'elettrodo a vetro. Si vedrà ora, più in dettaglio, il funzionamento di questo stadio.

La configurazione dell'amplificatore operazionale è non invertente: comunemente, un capo di R_1 sarebbe a massa; se però si collega il capo ad un nodo con un potenziale differente, si può introdurre un offset, regolabile utilizzando un trimmer. Per trovare la funzione che regola l'uscita del primo stadio, ci si può rifare all'uguaglianza tra le tensioni nei due ingressi degli amplificatori operazionali:

$$V_+ = V_- \Rightarrow V_{in} = \frac{R_1}{R_1 + R_2} V_{oi} + \frac{R_2}{R_1 + R_2} V_{os}$$

$$V_{in} - \frac{R_2}{R_1 + R_2} V_{os} = \frac{R_1}{R_1 + R_2} V_{oi}$$

$$V_{oi} = \left(1 + \frac{R_2}{R_1}\right) V_{in} - \frac{R_2}{R_1} V_{os}$$

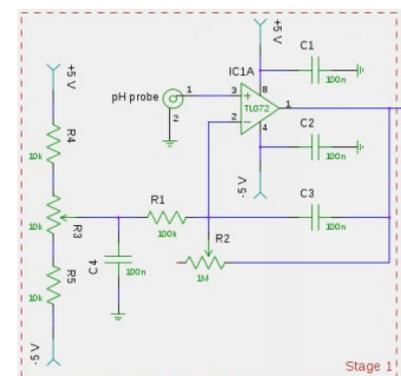
La tensione di offset V_{os} si può ridurre, in virtù delle alimentazioni con lo stesso valore assoluto, a questa espressione:

$$V_{os} = \left(\frac{R_3^I - R_3^{II}}{R_3 + 10k\Omega + 10k\Omega}\right) 5V$$

I comportamenti descritti da queste equazioni sono quelli richiesti per eseguire la calibrazione regolando i trimmer R_3 ed R_2 . La condizione che nella calibrazione a pH 7 si vuole ottenere, regolando l'offset col trimmer R_3 , è l'uguaglianza dei potenziali nei due ingres-

L'amplificatore operazionale

L'amplificatore operazionale è un componente che conosce svariati usi in elettronica. Nei casi qui presentati è utilizzato per amplificare tensioni, sfruttando la retroazione negativa. Il funzionamento con le configurazioni che prevedono tale retroazione è, sostanzialmente, molto semplice: applicato il segnale ad un ingresso, l'uscita adegua il suo valore di tensione in modo che, attraverso la retroazione, la tensione su entrambi gli ingressi sia uguale. L'uso di partitori di tensione permette di ottenere facilmente guadagni di tensione del valore richiesto.



Il primo stadio
Correzione dell'offset e amplificazione

Taratura

Basandosi sulla funzione che regola la tensione di uscita, si può simulare ciò che accade durante la taratura algebricamente, al fine di una migliore comprensione del funzionamento del circuito. Nel caso del pH-metro qui in progettazione, si devono mandare allo stadio successivo tensioni di 0 V a pH 7 e di circa 0,73 V a pH 4. L'elettrodo a vetro, si suppone, presenta un offset di 15 mV:

$$\text{pH} = 7: V_{in7} = 0,015 \text{ V } V_{out7} = 0 \text{ V}$$

$$\text{pH} = 4: V_{in4} = 0,236 \text{ V } V_{out4} = 0,731 \text{ V}$$

Si supponga di immergere l'elettrodo in una soluzione a pH 7 e di regolare il trimmer per la correzione dell'offset: la regolazione sarà tale per cui l'operatore leggerà in uscita 0 V.

$$0 = \left(1 + \frac{R_2}{100k\Omega}\right) V_{in} - \frac{R_2}{100k\Omega} V_{os}$$

$$V_{os} = \left(0,015 \text{ V} + \frac{R_2}{100k\Omega} \cdot 0,015 \text{ V}\right) \frac{100k\Omega}{R_2}$$

$$V_{os} = 0,015 \text{ V} + \frac{100k\Omega}{R_2} \cdot 0,015 \text{ V}$$

Come si può notare, V_{os} rimane incognita finché non si dà una regolazione de-

(continua)

(segue)

finitiva ad R_2 . Ciò non è sbagliato, perché la correzione dell'offset è sulla tensione in uscita, la quale dipende proprio dal guadagno. Sostituendo questa espressione dell'offset nella formula iniziale, infatti:

$$V_{o1} = \left(1 + \frac{R_2}{100k\Omega}\right) V_{in} - \frac{R_2}{100k\Omega} \left(0,015V + \frac{100k\Omega}{R_2} 0,015V\right)$$

$$V_{o1} = \left(1 + \frac{R_2}{100k\Omega}\right) V_{in} - \frac{R_2}{100k\Omega} 0,015V - 0,015V$$

$$V_{o1} = \left(1 + \frac{R_2}{100k\Omega}\right) V_{in} - \left(1 + \frac{R_2}{100k\Omega}\right) 0,015V$$

Il fattore di cui aumenta la tensione in entrata è lo stesso di cui aumenta la correzione dell'offset, come auspicato. Il passaggio successivo è immergere l'elettrodo in una soluzione a pH 4 e regolare il trimmer che fa da retroazione, con cui si adatta il guadagno:

$$0,731V = \left(1 + \frac{R_2}{100k\Omega}\right) 0,236V - \left(1 + \frac{R_2}{100k\Omega}\right) 0,015V$$

$$0,731V = \left(1 + \frac{R_2}{100k\Omega}\right) 0,221V$$

$$0,731V = 0,221V + \frac{R_2}{100k\Omega} 0,221V$$

$$R_2 = 100k\Omega \frac{0,731V - 0,221V}{0,221V} = 231k\Omega$$

A questo punto si dispone di una funzione che dà in uscita le tensioni richieste.

si: quando ciò si verifica, non vi è necessità della retroazione di riequilibrare il potenziale sull'ingresso invertente, e l'uscita dell'amplificatore fornirà pertanto un potenziale nullo. La regolazione del guadagno attraverso il trimmer R_2 , invece, è importante perché, come visto in precedenza, la tensione per unità di pH varia con la temperatura. Se si pensa che il microcontrollore dovrà fissare un livello a cui leggere pH 4, la regolazione del guadagno serve proprio a far sì che, immergendo l'elettrodo a vetro in una soluzione standard a pH 4, l'amplificatore operazionale fornisca quella tensione che dia quel livello. Il dimensionamento di trimmer e resistori è stato fatto prendendo come tensione massima dell'ADC 2.56 V, utilizzabile su Arduino Mega come tensione di riferimento attraverso un'istruzione specifica nel programma. Non è stato utilizzato il normale intervallo da 0 a 5 V perché gli amplificatori utilizzati, alimentati utilizzando il pin 5 V di Arduino, saturano a poco più di 4 V: ridurre il range dell'ADC permette di sfruttare tutta la risoluzione nella conversione.

Il secondo stadio opera un'elevazione delle tensioni, così che quelle negative inviate dall'elettrodo a pH basici rientrino in un range positivo. La configurazione dell'amplificatore, inoltre, è invertente: ciò permette di correggere la dipendenza tra pH e tensione, che sono così legati da una funzione lineare con coefficiente positivo, e non più negativo. I resistori impiegati sono uguali, poiché la fase di regolazione del guadagno è nel primo stadio e non sono più necessarie ulteriori amplificazioni.

L'elevazione della tensione è ottenuta, ancora una volta, riferendo il potenziale d'uscita non a massa, ma a una tensione di riferimento V_{ref} . Per avere idea di come venga elevata la tensione in uscita dal primo stadio, si ricorrerà ancora una volta all'uguaglianza tra le tensioni negli ingressi:

$$V_+ = V_- \Rightarrow \frac{R_7}{R_6 + R_7} V_{o1} + \frac{R_6}{R_6 + R_7} V_{o2} = V_{ref}$$

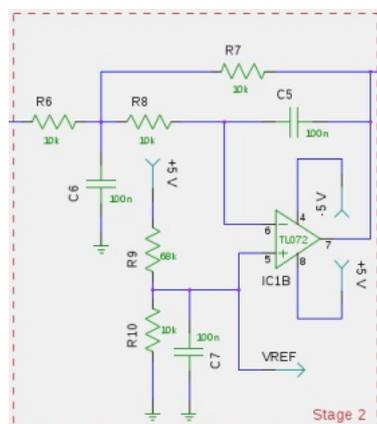
$$\frac{R_6}{R_6 + R_7} V_{o2} = V_{ref} - \frac{R_7}{R_6 + R_7} V_{o1}$$

$$V_{o2} = \left(1 + \frac{R_7}{R_6}\right) V_{ref} - \left(\frac{R_6 + R_7}{R_6}\right) \left(\frac{R_7}{R_6 + R_7}\right) V_{o1}$$

$$R_7 = R_6 \Rightarrow V_{o2} = 2V_{ref} - V_{o1}$$

Da quest'ultima equazione si può ben vedere come il segnale in uscita dal primo stadio sia invertito, e come venga poi elevato del doppio della tensione di riferimento. Anche in questo caso, il dimensionamento dei resistori, specialmente del partitore che fornisce V_{ref} , è pensato per il range di conversione da 0 a 2.56 V: la tensione di elevazione, infatti, è più vicina possibile alla metà di 2.56 V, ed è proprio la tensione che si viene a leggere quando dal primo stadio giungono 0 V. Questa tensione di riferimento è inviata anche a un pin analogico di Arduino, così che il programma possa avere il valore esatto a cui il pH deve essere 7.

Una volta condizionato il segnale, si passa al terzo stadio, che ha so-



Il secondo stadio
Elevazione a tensione positiva

stanzialmente il compito di proteggere l'ADC. Qui è stato posto un diodo Zener polarizzato inversamente, che svolge una duplice funzione di protezione: se dovessero arrivare delle tensioni negative, il potenziale di massa sarebbe allora più alto, la polarizzazione diventerebbe diretta e la corrente si scaricherebbe tutta a massa; se dovessero arrivare tensioni maggiori di 5 V, potenzialmente dannose per l'ADC e il microcontroller, il diodo andrebbe in breakdown (ovviamente bisogna munirsi di un diodo drogato per dare breakdown a circa 5 V). Le correnti che verrebbero a crearsi attraverso il diodo in questi casi sono tutte limitate dal resistore R_{11} in uscita dall'operazionale.

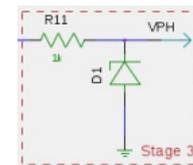
Per concludere lo studio del circuito di amplificazione, si vuole richiamare l'attenzione su alcuni brevi punti:

- il potenziale in uscita dall'elettrodo a vetro deve essere riferito alla massa del circuito: per fare ciò, si collega in entrata all'amplificatore operazionale il contatto centrale del cavo coassiale, a massa il contatto esterno;
- è importante che gli amplificatori scelti abbiano un'elevata impedenza d'ingresso (in questo caso sono dei TL072), poiché allo stesso modo è elevatissima la resistenza della membrana sensibile (tra 10 e 100 M Ω) e le correnti coinvolte sono pertanto molto basse;
- nel circuito di amplificazione è presente anche la parte di filtraggio, ottenuta con l'utilizzo di condensatori che filtrano le alte frequenze mantenendo stabili, come del resto è anche il pH di una soluzione, le tensioni;
- si può osservare che, per fornire l'alimentazione negativa agli operazionali e per dare la possibilità di regolare un offset anche negativo, è stato utilizzato un integrato 7660: esso è una pompa di carica, e tra i tanti utilizzi che se ne può fare vi è anche quello di invertitore di tensione (dai 5 V del pin "5 V" di Arduino, così, si può ottenere una linea di -5 V).

Pulsanti, display, e lettore per schede di memoria SD

Per la visualizzazione del pH misurato, si può utilizzare un display LCD 16 x 2, il quale può essere controllato molto facilmente attraverso apposite librerie scritte per Arduino. Ciò evita la necessità di un computer che riceva i dati dalla porta seriale e li stampi a schermo: il pH-metro diventa, così, perfettamente portatile ed indipendente.

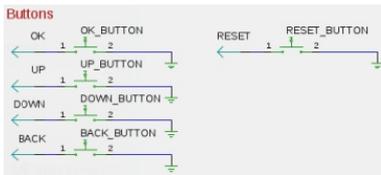
Siccome Arduino offre elevate potenzialità, è possibile sfruttarlo anche per disporre di diverse funzionalità utili: l'indipendenza, attraverso opportuni accorgimenti, si può accompagnare alla versatilità. Ecco, allora, che si possono aggiungere dei pulsanti con cui controllare il programma caricato sulla board e un lettore per schede



Il terzo stadio
Protezione dell'ADC



Display LCD
Un display LCD 16x2 interfacciato ad Arduino



I pulsanti

I bottoni sono collegati a massa da un capo, dall'altro a pin digitali a livello alto. Quando il bottone è premuto la tensione si scarica e il livello dei pin è basso. Siccome si vuole scrivere un programma che permetta di navigare in un menù per compiere varie operazioni, vi sono due bottoni per navigare tra le opzioni, uno per confermarle e uno per tornare indietro. Il pulsante "reset" è collegato all'omonimo pin di Arduino e permette, appunto, di resettare il programma.

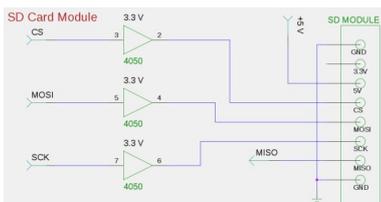
SPI

L'SPI (Serial Peripheral Interface) è un protocollo per la comunicazione di dati sincronizzata usato dai microcontroller per comunicare con uno o più dispositivi periferici velocemente e su piccole distanze. Può essere utilizzato anche per la comunicazione tra due microcontroller.

Con una connessione SPI c'è sempre un dispositivo master (padrone), di solito un microcontroller, che controlla i dispositivi periferici, detti dispositivi slave (servi). Di solito ci sono tre linee comuni a tutti i dispositivi:

- MISO (Master In Slave Out): la linea dello slave per mandare dati al master;
- MOSI (Master Out Slave In): la linea del master per mandare dati allo slave;
- SCK (Serial Clock): la linea con gli impulsi del clock per la sincronizzazione dei dati inviati dal master;

Se si deve comunicare via SPI con più dispositivi, questi si possono attivare alternativamente attraverso la linea SS (Slave Select): il master attiverà la linea SS del solo dispositivo con cui deve comunicare in quel momento.



Letture per schede SD

Alimentando un 4050 con 3.3 V, tutti i livelli logici alti vengono abbassati a quella tensione.

di memoria SD con cui salvare dei dati.

Per utilizzare i pulsanti si possono abilitare tramite programma i resistori di pull-up interni ai pin digitali. Collegando un capo del pulsante al pin e l'altro capo a massa, si avrà un livello logico alto in quel pin se il circuito è aperto, un livello logico basso se il circuito è chiuso. Arduino, allora, può essere programmato per assumere diversi comportamenti a seconda dei pulsanti premuti e dei livelli logici presenti. Di seguito si propongono quattro bottoni, più uno per il reset collegato all'apposito pin di Arduino, con cui inviare segnali digitali a quattro pin di Arduino:

La scheda di memoria SD, invece, necessita di comunicare attraverso un particolare protocollo, detto SPI, con Arduino. Ci sono dei pin specificamente denominati, su ogni board Arduino, per comunicare attraverso questo protocollo; essi devono essere collegati ai pin con la stessa denominazione del dispositivo a cui ci si deve interfacciare, in questo caso una scheda di memoria SD. È importante tenere conto del fatto che la maggior parte delle board Arduino opera a livelli logici alti di 5 V, mentre le schede SD a 3.3 V: per evitare di danneggiarle, è consigliato l'uso di un level shifter (se non già presente nel modulo per scheda SD), ovvero un integrato che abbassi la tensione del livello logico alto; ad esempio, utilizzando un 4050 e alimentandolo col pin 3.3 V di Arduino, tutti i livelli logici in entrata compariranno nelle uscite abbassati a 3.3 V quando alti. Si presti attenzione al fatto che la linea MISO è l'unica che non ha bisogno di shifting, perché in quel caso è Arduino a leggere un livello logico inviato dalla scheda SD.

Si può, infine, inserire anche un buzzer, in serie ad un resistore da 100 kΩ, collegato ad un pin digitale: con un'istruzione apposita esso si può attivare da programma, ad esempio per realizzare un allarme.

All'Appendice A si trovano gli schemi elettrici completi per collegare display, pulsanti, buzzer e lettore per schede di memoria SD.

Si vedrà nell'ultima parte del fascicolo, relativa alla programmazione, quali funzionalità si vogliono implementare impiegando i componenti qui descritti.

Parte 3

Programmazione

È possibile consultare all'Appendice B l'intero programma caricato su Arduino per dare vita al pH-metro, con tanto di commenti esplicativi. L'interfacciamento ai pulsanti e al display, nonché l'utilizzo delle istruzioni di base, non sono trattati in questa sede, per via della semplicità con cui li si può apprendere consultando i tutorial sul sito ufficiale (sempre all'Appendice si possono trovare i link).

Si preferirà trattare gli aspetti fondamentali del programma, come la modalità di lettura del pH e le funzionalità di calibrazione polinomiale, salvataggio delle curve di titolazione, allarme e monitoraggio del pH nel tempo.

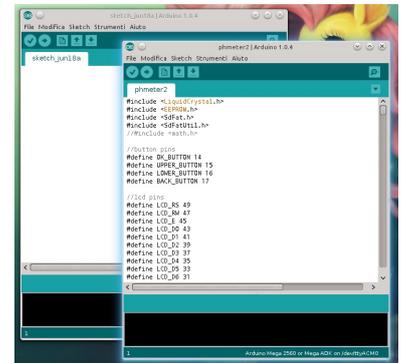
Funzionamento generale

Appena avviato il pH-metro, viene eseguito il controllo della presenza della scheda SD. Se essa non è presente, è rimpiazzata dall'utilizzo della EEPROM (la quale è, però, meno funzionale).

Dopo l'avvio, appare la schermata principale, gestita da un ciclo `while`: qui viene visualizzato il pH, il potenziale corrispondente sulla membrana e anche i byte letti sui due pin analogici, quello per il pH e quello per la tensione di riferimento.

Oltre alle funzioni per aggiornare questi dati, di cui si parlerà in seguito, è continuamente chiamata la funzione `bState()`, per il monitoraggio dei pulsanti. Essa restituisce un valore a seconda del pulsante che è premuto in un certo momento: i tasti upper e lower permettono la navigazione nei menù, il tasto ok la conferma delle opzioni e il tasto back di tornare indietro. La funzione è studiata anche per non restituire in due chiamate consecutive il valore dello stesso pulsante: i cicli si ripetono, infatti, molto velocemente, e siccome l'operatore molto probabilmente non è così veloce da rilasciare il pulsante tra una chiamata e l'altra della funzione, alla chiamata successiva la funzione restituisce 0, in modo da non compiere azioni indesiderate oltre a quelle che si intendeva effettuare.

Nel programma sono utilizzate diverse istruzioni `switch` per ottenere



L'IDE di Arduino

Una schermata dell'ambiente di sviluppo integrato di Arduino, disponibile per GNU/Linux, Microsoft Windows ed Apple Macintosh. È caldamente consigliato ai neofiti, per un interfacciamento il più semplice possibile col dispositivo.

un comportamento a seconda del valore restituito da `bState()`. Molto spesso gli switch vanno a modificare alcune variabili di controllo (`mainList`, `mainControl`, `ctrl`), le quali, a seconda del valore, attivano un determinato ciclo `while`. Per navigare nei menu, nello specifico, viene modificata la variabile `mainList` tramite la pressione dei tasti `upper` e `lower`, così che al posto della schermata principale appaiano le varie sessioni con cui modificare il comportamento del pH-metro; la variabile `mainControl`, invece, è utilizzata dai vari cicli `while` per scegliere in quale routine il programma deve entrare; la variabile `ctrl`, infine, ha una funzione simile a quella di `mainControl`, ma condiziona i cicli all'interno delle stesse sessioni principali. Questi comportamenti, come è lecito immaginare, sono ottenuti imponendo adeguate condizioni ai cicli `while` e alle istruzioni `if`.

Tra le funzioni importanti si annoverano anche `selectNumber()` e `composeNumber()`. La prima è chiamata quando si vuole richiedere all'operatore di inserire un numero, anche decimale non intero: essa torna utile, ad esempio, quando si vuole regolare la temperatura a cui si trova la soluzione, o quando facendo la curva di titolazione si vogliono immettere i millilitri aggiunti. La funzione, che dev'essere chiamata ripetutamente in una routine, gestisce sia l'input sia la visualizzazione sul display, e quando l'immissione è completata modifica una variabile di controllo (di solito `ctrl`) passata come puntatore, in modo che il programma capisca quando andare avanti. Le cifre che l'operatore modifica una a una, sono salvate in un array, anch'esso passato come puntatore, così possa essere modificato (se fosse semplicemente passato come argomento, ciò non avverrebbe).

Una volta completata l'immissione, l'array è passato alla funzione `composeNumber()`, la quale, a partire dalle cifre in esso presenti, restituisce il numero sotto forma di variabile `double`, in virgola mobile.

La lettura del segnale

Il valore del pH è restituito dalla funzione `readPh()`, la quale converte il byte letto sul pin analogico in unità di pH. La modalità di conversione di basa sulla retta passante per due punti: al byte letto sul pin analogico a cui è inviata la tensione di riferimento corrisponde il pH 7, mentre a pH 14 corrisponde il byte massimo, 1023.

pH = 7 byte = `byteref`

pH = 14 byte = 1023

La funzione con cui il byte è convertito si può calcolare di conseguenza, con la formula della retta passante per due punti:

$$\frac{pH - 7}{byte - byte_{ref}} = \frac{14 - 7}{1023 - byte_{ref}}$$

$$\frac{pH - 7}{byte - byte_{ref}} = \frac{14 - 7}{1023 - byte_{ref}}$$

$$pH = \frac{7}{1023 - \text{byte}_{ref}} \cdot (\text{byte} - \text{byte}_{ref}) + 7$$

Questa è la funzione con cui è calcolato il pH visualizzato sul display LCD; ciò che avviene durante la taratura è l'adeguamento dei segnali inviati dal circuito di condizionamento a ciò che si aspetta il programma col calcolo appena trovato. Se però è stata eseguita o caricata una calibrazione elettronica (di cui si parlerà in seguito), il programma utilizza una diversa funzione polinomiale che passi per i punti byte-pH impostati dall'operatore. È sempre la funzione readPh() a fare il discernimento.

Essendo possibile definire a quale temperatura si trova la soluzione, sul display è visualizzato anche il potenziale dell'elettrodo. Esso si può facilmente ricavare dal pH utilizzando la legge di Nernst:

$$V(mV) = 0,00019842 \cdot T \cdot (7 - pH) \cdot 1000$$

Calibrazione polinomiale

La calibrazione polinomiale, configurabile con la sessione “Calibration”, è utile se si vogliono ottenere misure che ricoprano un range dall'acido al basico, poiché quella manuale si considera affidabile solo nel range dello standard usato per regolare il guadagno. Per contro, la calibrazione polinomiale tende a fornire funzioni sempre più inaffidabili all'infuori del campo di valori usato per la calibrazione.

La calibrazione polinomiale si svolge immergendo l'elettrodo a vetro in uno standard e comunicando al pH-metro il suo pH. Viene, così, raccolta una serie di punti, che hanno come coordinate il byte in ingresso e il corrispondente pH impostato dall'operatore; alla funzione che restituisce il pH dal byte d'ingresso è imposto di passare per questi punti, così che, risolvendo un sistema di equazioni, si ottengano i coefficienti del suo polinomio.

Nello specifico, si vuole trovare una funzione $f(x)$ che dia il pH dato un byte (x):

$$f(x) = ax^0 + bx^1 + cx^2 + dx^3 + \dots + zx^n$$

Per trovarla si danno dei punti per cui essa deve passare:

$$P_1(x_1; y_1), P_2(x_2; y_2), P_3(x_3; y_3), \dots, P_n(x_n; y_n)$$

Attraverso un sistema di equazioni, si impone il passaggio della funzione per i punti dati; maggiore è il numero di punti, maggiore è il numero di equazioni, di monomi della funzione e relativi coefficienti da trovare:

$$\begin{cases} y_1 = ax_1^0 + bx_1^1 + cx_1^2 + dx_1^3 + \dots + zx_1^n \\ y_2 = ax_2^0 + bx_2^1 + cx_2^2 + dx_2^3 + \dots + zx_2^n \\ y_3 = ax_3^0 + bx_3^1 + cx_3^2 + dx_3^3 + \dots + zx_3^n \\ \dots \\ y_n = ax_n^0 + bx_n^1 + cx_n^2 + dx_n^3 + \dots + zx_n^n \end{cases}$$

Esempio di risoluzione di un sistema

È dato un sistema di equazioni lineari a più incognite:

$$\begin{cases} x + y - z = 9 \\ y + 3z = 3 \\ -x - 2z = 2 \end{cases}$$

Si dispongono ordinatamente in una matrice i coefficienti e i termini noti:

$$\begin{pmatrix} 1 & 1 & -1 & 9 \\ 0 & 1 & 3 & 3 \\ -1 & 0 & -2 & 2 \end{pmatrix}$$

Si procede al passaggio alla matrice a scalini; si somma alla terza riga la prima:

$$R_1 + R_3 \Rightarrow R_3$$

$$\begin{pmatrix} 1 & 1 & -1 & 9 \\ 0 & 1 & 3 & 3 \\ 0 & 1 & -3 & 11 \end{pmatrix}$$

Si sottrae alla terza riga la seconda:

$$-R_2 + R_3 \Rightarrow R_3$$

$$\begin{pmatrix} 1 & 1 & -1 & 9 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & -6 & 8 \end{pmatrix}$$

La matrice è ridotta a scalini; si porta a 1 il coefficiente di z nella terza riga:

$$-\frac{1}{6}R_3 \Rightarrow R_3$$

$$\begin{pmatrix} x & y & z & \\ 1 & 1 & -1 & 9 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & 1 & -\frac{4}{3} \end{pmatrix}$$

Con la sostituzione, si possono calcolare le incognite:

$$z = -\frac{4}{3}$$

$$y + 3z = 3 \Rightarrow y = 3 - 3\left(-\frac{4}{3}\right) = 7$$

$$x + y - z = 9 \Rightarrow x = -7 - \frac{4}{3} + 9 = \frac{2}{3}$$

Per semplicità è stata usata la sostituzione, ma lavorando sulla matrice applicando il metodo inversamente, si potevano isolare in ogni riga le incognite col loro risultato, proprio come fa l'algoritmo usato nel programma.

Un algoritmo con cui risolvere facilmente un simile sistema in un programma è quello ricavabile dal metodo di eliminazione di Gauss, che si può usare in matematica per trovare le soluzioni di un sistema di equazioni lineari. Il metodo prevede di disporre i coefficienti delle incognite (le ascisse dei punti) e i termini noti (le ordinate dei punti) in una matrice secondo la seguente modalità:

$$\begin{pmatrix} x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^n & y_1 \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^n & y_2 \\ x_3^0 & x_3^1 & x_3^2 & \cdots & x_3^n & y_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_n^0 & x_n^1 & x_n^2 & \cdots & x_n^n & y_n \end{pmatrix}$$

A questo punto, si può agire sulla matrice attraverso le mosse di Gauss, le quali possono essere:

- scambio di due righe;
- moltiplicazione di una riga per un numero diverso da zero;
- somma di una riga ad un multiplo di un'altra riga.

È semplice notare che le mosse di Gauss applicate alla matrice completa dei coefficienti corrispondono, nella classica maniera di esprimere un sistema di equazioni (con parentesi graffa), a:

- scambiare l'ordine di scrittura di due equazioni;
- moltiplicare entrambi i membri di un'equazione per un numero diverso da zero;
- sommare ad ogni membro di un'equazione la stessa quantità a sinistra e a destra.

L'obiettivo dell'algoritmo, che sfrutta quindi le mosse di Gauss, è ottenere una matrice a scalini, cioè una matrice in cui il primo elemento diverso da zero di una riga sia più a destra del primo elemento diverso da zero della riga precedente. È chiamata così perché gli zeri formano una specie di scala. Il primo elemento di una riga diverso da zero è detto pivot. Ecco un esempio di matrice a scalini, con pivot 3, -1 e 5:

$$\begin{pmatrix} 3 & 0 & 4 & 7 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Di seguito, il funzionamento dell'algoritmo con cui ottenere una matrice a scalini:

1. Se la prima riga ha il primo elemento nullo, scambiala con una riga che ha il primo elemento non nullo. Se tutte le righe hanno il primo elemento nullo, vai al punto 3.
2. Per ogni riga A_i con primo elemento non nullo, eccetto la prima ($i > 1$), moltiplica la prima riga per un coefficiente scelto in maniera tale che la somma tra la prima riga e A_i abbia il primo elemento nullo (quindi coeffi

ciente $= -\frac{A}{A}$). Sostituisci A_i con la somma

appena ricavata.

3. Adesso sulla prima colonna tutte le cifre, eccetto forse la prima, sono nulle. A questo punto ritorna al punto 1 considerando la sottomatrice che ottieni cancellando la prima riga e la prima colonna.

Nella variante Gauss-Jordan, dopo aver ridotto una matrice a scalini, è possibile usare l'algoritmo in senso inverso, cioè dal basso verso l'alto, per ottenere una matrice che in ogni colonna abbia solo il pivot (se presente) come numero non nullo, (questa matrice risultante è anche detta matrice a scalini in forma ridotta): basta usare ogni riga, partendo dall'ultima, per eliminare tutte le cifre diverse da zero che stanno sopra al pivot di questa riga. Infine, sempre con mosse di Gauss (moltiplicando righe), si può ottenere che ogni pivot abbia valore 1.

Si noti che nella riduzione a scalini bisogna considerare solo la parte di matrice coi coefficienti, anche se ovviamente anche i termini noti sono affetti dalle operazioni che si compiono. In questo modo, alla fine ci si ritroverà con una matrice con un solo 1 in ogni riga e un solo 1 in ogni colonna (saranno disposti, cioè, diagonalmente); l'1 corrisponde ad una certa incognita a seconda della colonna in cui si trova, e quell'incognita avrà il valore contenuto nell'ultima colonna sulla stessa riga dell'1. Ad esempio, con una matrice simile:

$$\begin{pmatrix} x & y & z & \\ 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 6 \end{pmatrix}$$

si ha che x vale 4, y vale 0 e z vale 6.

Il programma scritto per il pH-metro utilizza questo algoritmo per il calcolo dei coefficienti della funzione polinomiale. Una volta effettuata la taratura dall'operatore, la variabile booleana globale polyn passa allo stato true e la funzione readPh() passa di conseguenza all'uso della funzione appena trovata.

È anche possibile salvare le calibrazioni sulla EEPROM di Arduino, per poi caricarle in un'altra sessione. Essendo che i blocchi di memoria di questa unità arrivano ad un valore intero di solo 255, i salvataggi sono attuati con uno stratagemma: a essere salvati, innanzitutto, sono i punti inseriti dall'operatore, mentre i coefficienti sono ricalcolati ogni volta che si ricarica la calibrazione; nel salvataggio, quindi, i pH inseriti vengono scomposti nella loro parte intera e decimale e memorizzati su due blocchi di memoria, mentre i byte corrispondenti, che possono arrivare ad un massimo di 1023, divisi in 5 addendi non più grandi di 255 in 5 blocchi di memoria. Quando la calibrazione viene caricata, i punti vengono ricomposti e i

coefficienti calcolati. Un blocco di memoria è utilizzato anche per salvare il numero di punti utilizzati, così che in fase di lettura il programma sappia quanti punti utilizzare e quando fermarsi a leggere.

Naturalmente è possibile disattivare la calibrazione polinomiale in qualsiasi momento e tornare alla funzione lineare.

Curva di titolazione

Nella sessione “Titration curve” si dispone di qualche funzionalità utile a velocizzare il processo di creazione di una curva di titolazione. Quando si esegue una titolazione, si può avviare questa sessione. Una volta attivata, il programma stampa a schermo il pH letto in quel momento e chiede all'operatore, ad ogni aggiunta di titolante, di selezionare quanti millilitri sono stati immessi. Essi vengono sommati a quelli aggiunti precedentemente e salvati col pH su un file nella scheda SD. Si procede in questo modo fino al completamento della titolazione: a quel punto si può uscire dalla sessione e il file viene salvato definitivamente. Vi si troveranno, in esso, tutti i punti della curva di titolazione, aventi come coordinate il totale di millilitri aggiunti ad un certo punto e il pH ivi misurato.

Durante la sessione viene svolto anche il calcolo dei punti di flesso: per farlo, si possono individuare i punti in cui il valore assoluto della derivata raggiunge il massimo, oppure in cui la derivata seconda cambia di segno. Il programma qui in esame utilizza la prima soluzione. L'algoritmo per individuare i punti di flesso si può sintetizzare nei seguenti passi:

1. Trova il valore assoluto della derivata in un punto facendo il rapporto incrementale utilizzando le coordinate del punto precedente.
2. Se non è la prima derivata che calcoli, confrontala con quella precedente. Se ti trovi:
 - nella fase di incremento, non fare nulla se il valore assoluto è aumentato, individua il punto di flesso se è diminuito e passa alla fase di decremento;
 - nella fase di decremento, non fare nulla se il valore assoluto è diminuito, passa alla fase di incremento se è aumentato;

I punti di flesso sono quindi salvati nel file assieme alla curva di titolazione e, per indicarli, si usano i millilitri presso cui sono individuati.

L'interfacciamento alla scheda SD avviene con la libreria per Arduino `stdfatlib`, che supporta le funzioni I/O standard del C++. Se la scheda SD non è presente, i dati sono salvati su EEPROM (con una modalità simile a quella presentata per il salvataggio dei dati di calibrazione) e sono inviabili tramite la porta seriale, ad esempio ad un PC che dispone di monitor seriale, utilizzando la funzione “Do-

wnload data”.

Allarme

L'allarme, configurabile alla voce “Alarm”, fornisce un segnale visivo e sonoro al superamento o alla discesa al di sotto di un certo limite di pH, a discrezione dell'operatore. Il segnale visivo compare sul display, mentre il segnale sonoro è ottenuto con l'uso del buzzer collegato a un pin digitale, con in serie un resistore.

La funzione che si occupa della gestione dell'allarme è `alarmCheck()`, che viene ciclicamente chiamata durante le sessioni di idle, di composizione della curva di titolazione e di monitoraggio. Il controllo è svolto solo se l'operatore ha impostato come true la variabile booleana `alarm`, altrimenti la funzione non fa nulla. In quel caso, è operato il confronto tra il pH limite e il pH misurato: a seconda della scelta effettuata (contenuta in un'altra variabile booleana), limite superiore o inferiore, i segnali visivo e sonoro sono attivati per qualche secondo se avviene il superamento. A quel punto l'allarme si disattiva.

La funzione di allarme può essere utile in diverse fasi per ricevere un segnale se il pH oltrepassa certi valori (i pH-metri sono spesso utilizzati, ad esempio, per controllare che gli acquari mantengano il pH entro certi limiti), oppure in una curva di titolazione si può impostare l'allarme per suonare all'approssimarsi della fase in cui è necessario moderare le aggiunte, perché ci si sta avvicinando al punto di equivalenza.

Monitoraggio

La sessione di monitoraggio, attivabile alla voce “Measure over time”, permette di scegliere un periodo di campionamento e iniziare a monitorare il pH nel tempo. Il tempo trascorso dall'inizio e il pH misurato in quell'istante vengono salvati sulla scheda SD. Se essa non è presente, sono salvati sulla EEPROM e, ancora una volta, sono accessibili utilizzando la funzione “Download Data”.

L'utilità di questa sessione è sicuramente poter vedere come cambia il pH di un sistema nel corso di un tempo prolungato, ma si potrebbe usare con successo anche per le titolazioni: se si riescono a sincronizzare la aggiunte di titolante, che devono essere sempre uguali, col periodo di campionamento, si può ricavare a ogni tempo quanto titolante è stato aggiunto e ottenere una vera e propria curva di titolazione. Con un po' meno di precisione, ma sempre con aggiunte uguali tra loro, potrebbe bastare anche semplicemente dividere il titolante totale per il numero di campionamenti e farlo incrementare della quantità trovata ad ogni istante. Qualunque modalità si decida di usare, una titolazione che si fa praticamente da sé è una gran comodità :D .

Appendice A

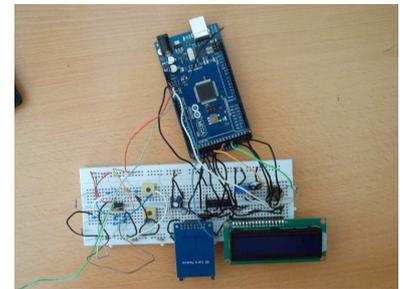
Schemi elettrici

Il progetto presentato in questo fascicolo è basato su *phduino*, un progetto di Carlos A. Neves, realizzato all'*Instituto de Química de São Carlos, Universidade de São Paulo, Brazil*. Il progetto è rilasciato sotto licenza Creative Commons, come anche il progetto derivato qui presentato, e ha una pagina Google Code reperibile all'indirizzo <http://code.google.com/p/phduino/>.

Si riporta, qui, la lista dei componenti necessari alla realizzazione del dispositivo descritto nel corso del fascicolo, e nella pagina successiva gli schemi elettrici.

Lista componenti

- Arduino Mega
- IC₁ TL072
- IC₂ 7660
- IC₃ 4050
- Resistore R₁ 100k
- Trimmer R₂ 1M
- Trimmer R₃ 10k
- Resistore R₄ 10k
- Resistore R₅ 10k
- Resistore R₆ 10k
- Resistore R₇ 10k
- Resistore R₈ 10k
- Resistore R₉ 10k
- Resistore R₁₀ 10k
- Resistore R₁₁ 1k
- Condensatori C₁÷C₈ 100n
- Condensatori elettrolitici C₉, C₁₀ 100µx25V
- Diodo Zener D₁, tensione di breakdown 5,1 V
- 5 pulsanti
- LC Studio SD Card Module
- Display LCD 16x2
- Trimmer 10k per regolare il contrasto del display



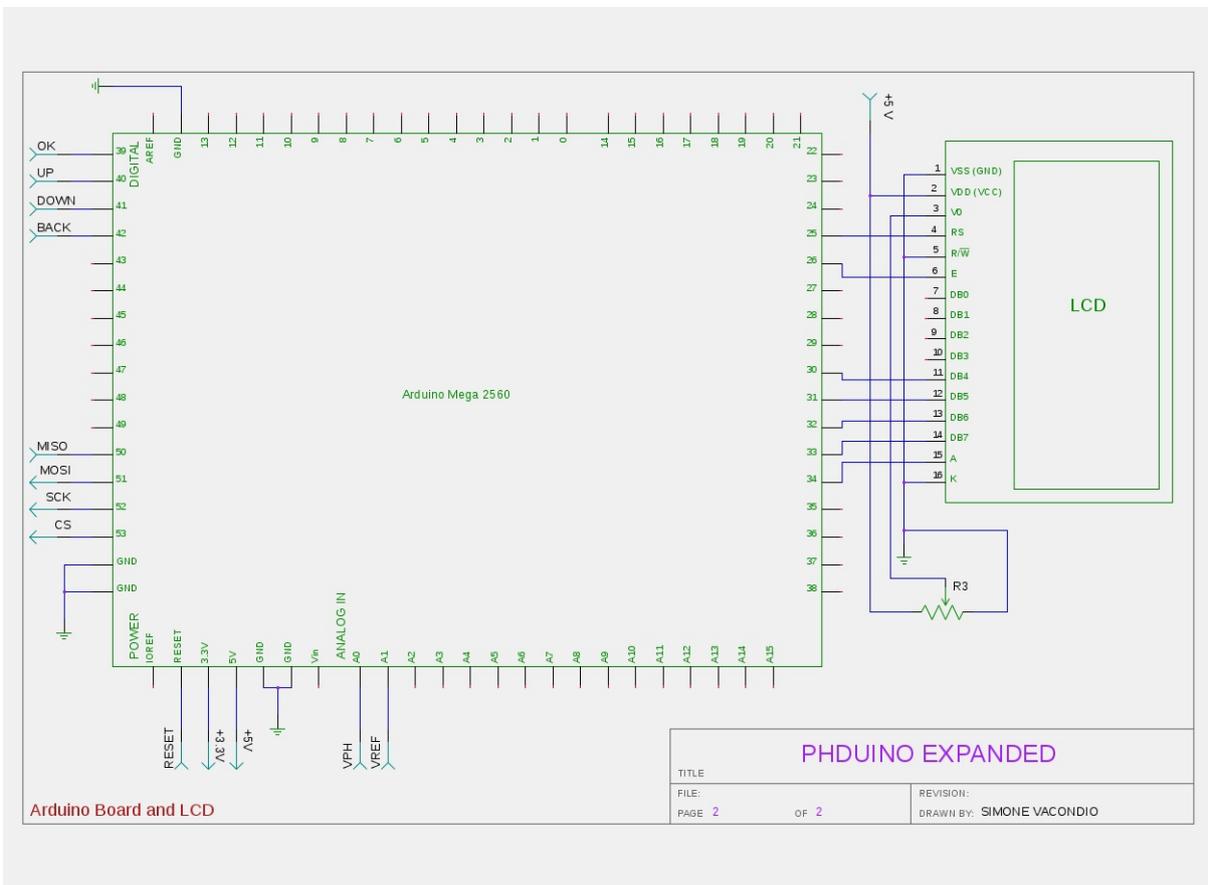
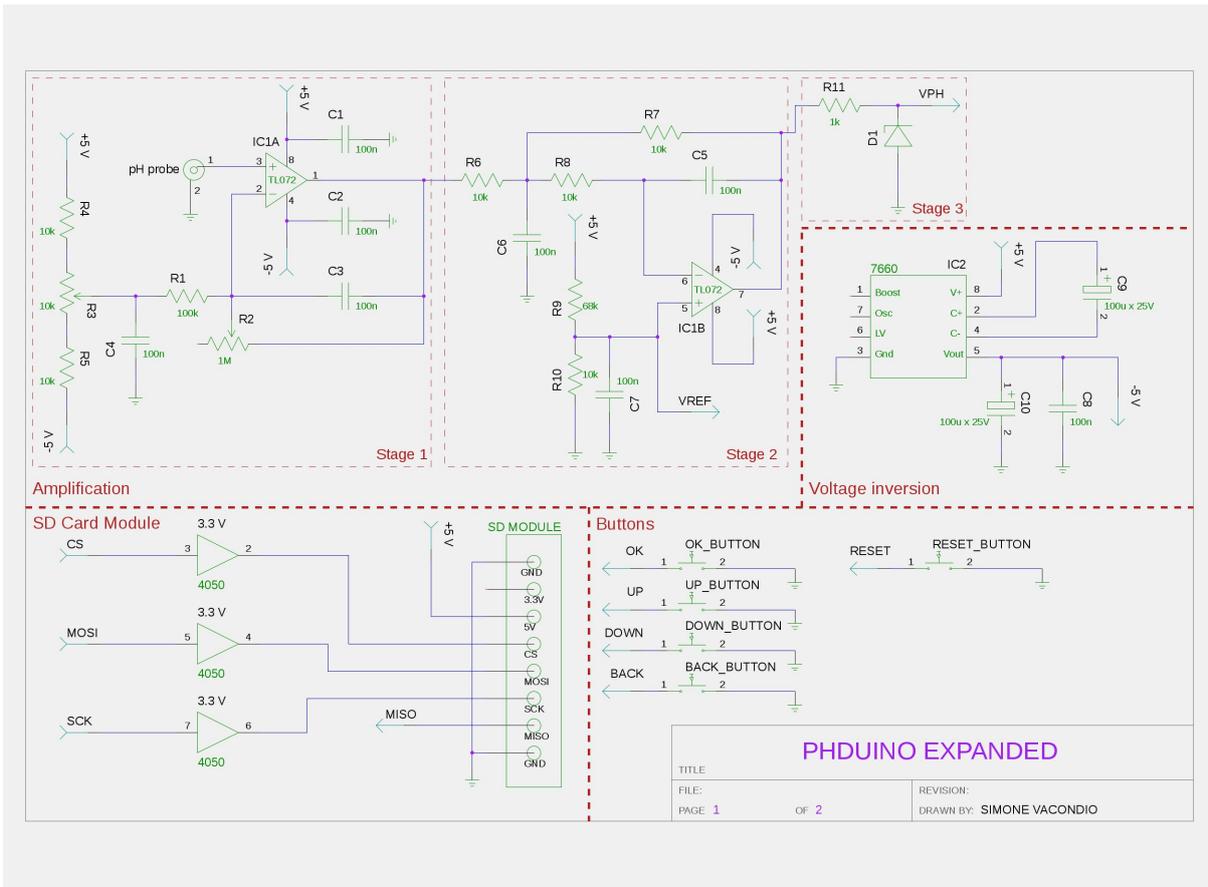
Un prototipo su breadboard

Può essere conveniente realizzare il prototipo di un progetto su breadboard. Inoltre mandando segnali analogici utilizzando un trimmer (facendo finta che siano in uscita dal circuito di condizionamento), si possono compiere tanti test preliminari senza bisogno dell'elettrodo a vetro e di soluzioni.



Un prototipo con millefori

Una realizzazione del pH-metro su millefori, la quale è stata collegata ad Arduino tramite cavi flat. Con opportuni accorgimenti (ad esempio utilizzando una Arduino ProtoShield) si può creare un circuito che si incastrerà comodamente sui pin di Arduino.



Appendice B

Programma

Imparare a programmare è un processo che può diventare più o meno lungo a seconda della passione e delle attitudini. Ciò che è certo è che nella programmazione non si finisce mai di imparare; è anche vero che programmare Arduino è piuttosto facile.

Per queste ragioni il lettore è invitato a documentarsi innanzitutto sulla programmazione in C++, la cui conoscenza di base apre tantissime porte, poi anche su come applicarla ad Arduino.

Un buon sito su cui studiare il linguaggio è:

<http://www.learncpp.com/>

mentre per Arduino si consiglia di consultare le pagine del sito ufficiale con l'indice di riferimento:

<http://arduino.cc/en/Reference/HomePage>

e di provare a seguire qualche tutorial, in particolare quelli introduttivi e quelli relativi all'uso del display LCD e della scheda di memoria SD; i tutorial sono reperibili all'indirizzo:

<http://arduino.cc/en/Tutorial/HomePage>

Dalla pagina seguente in avanti si propone la versione del programma caricata su Arduino scritta dall'autore, con tanto di commenti. Ovviamente esistono infinite soluzioni di programmazione migliori e più eleganti per far compiere al pH-metro le stesse funzioni; si spera che questo sia uno stimolo per il lettore a trovarne.

Appendice B

```
#include <LiquidCrystal.h>
#include <EEPROM.h>
#include <SdFat.h>
#include <SdFatUtil.h>
//#include <math.h>

//button pins
#define OK_BUTTON 14
#define UPPER_BUTTON 15
#define LOWER_BUTTON 16
#define BACK_BUTTON 17

//lcd pins
#define LCD_RS 49
#define LCD_RW 47
#define LCD_E 45
#define LCD_D0 43
#define LCD_D1 41
#define LCD_D2 39
#define LCD_D3 37
#define LCD_D4 35
#define LCD_D5 33
#define LCD_D6 31
#define LCD_D7 29
#define LCD_BLIGHT 27

//alarm pin
#define ALARM_PIN 18

//SD CS pin
#define SD_CS 53

/*EEPROM addresses
EEPROM addresses usage:
-0: number of calibration points;
-1 to 63: pH and analog values for calibration;

-100: number of titration curve points;
-101 to 900: pH and mL values for titration curve;

-1000: number of measure over time points;
-1001: number of measure over time points;
-1002: sample period in seconds;
-1003: to 4000: pH values

-4094: number used when saving a file with titration data to SD
-4095: number used when saving a file with measure over time data
to SD
*/
#define CALIB_N_ADDR 0
#define CALIB_PTS_ADDR 1

#define TITR_N_ADDR 100
#define TITR_PTS_ADDR 101
#define TITR_MAX_POINTS 200

#define SAMPLE_N1_ADDR 1000
#define SAMPLE_N2_ADDR 1001
#define SAMPLE_PERIOD_ADDR 1002
#define SAMPLE_PTS_ADDR 1003
#define SAMPLE_MAX_POINTS 1500

#define SD_TITR_NAME_ADDR 4094
#define SD_SAMPLE_NAME_ADDR 4095

//ph holding variable
double ph;

//linear/manual calibration parameters
double temperature = 298.15;

//polynomial calibration parameters
boolean polyn = false;
double coeff[10];
int nCoeff;

//alarm activator
boolean alarm = false;
boolean upperLimit;
double phLimit;

//listing and control variables
int mainList = 0;
int mainControl = 0;
char ctrl = '1';

//time holder
unsigned long currentMillis;

//backlight switch
boolean backlight=true;

//SD variables
SdFat sd;
boolean SD_av;

//lcd object
LiquidCrystal lcd(LCD_RS, LCD_RW, LCD_E, LCD_D0, LCD_D1, LCD_D2,
LCD_D3, LCD_D4, LCD_D5, LCD_D6, LCD_D7);

void setup () {
//sets max analog input to 2.56V
analogReference (INTERNAL2V56);

//serial monitor setup
Serial.begin(9600);

//button setup
pinMode (OK_BUTTON, INPUT_PULLUP);
pinMode (UPPER_BUTTON, INPUT_PULLUP);
pinMode (LOWER_BUTTON, INPUT_PULLUP);
pinMode (BACK_BUTTON, INPUT_PULLUP);

//alarm setup
pinMode (ALARM_PIN, OUTPUT);

//lcd setup
lcd.begin(16, 2);
pinMode(LCD_BLIGHT, OUTPUT);
digitalWrite(LCD_BLIGHT, HIGH);
lcd.clear();

//SD setup
if (!sd.begin(SD_CS, SPI_HALF_SPEED)) {
//sd.initErrorHalt();
SD_av=false;

lcd.print("SD unavailable");
lcd.setCursor(0, 1);
lcd.print("Using EEPROM");
delay (2000);
lcd.clear();
}
else {
SD_av=true;

lcd.print("Using SD");
delay(2000);
lcd.clear();
}

Serial.print("Free Ram: ");
Serial.println(FreeRam());
}

void loop () {
//main menu
while (mainControl==0) {
switch (bState()){
case UPPER_BUTTON:
if (mainList<7){
mainList++;
lcd.clear();
}
break;

case LOWER_BUTTON:
if (mainList>0) {
mainList--;
lcd.clear();
}
break;

case OK_BUTTON:
mainControl = mainList;
break;

case BACK_BUTTON:
mainList=0;
lcd.clear();
break;
}

switch (mainList){
case 0:
//idle screen, repeatedly printing pH and mV
ph = readPh();
lcd.home();
lcd.print("pH:");
lcd.print (ph);
lcd.print(" ");
alarmCheck(ph);

lcd.print("mV:");
lcd.print(temperature*0.00019842*(7-ph)*1000);
lcd.print(" ");

/*lcd.setCursor(0, 1);
* lcd.print("temp.: ");
* lcd.print(temperature);
*/
}
}
}
}
```

```

        *      lcd.print(" C");
        *      break;*/

//prints the analog input as well (only for development)
delay(200);
lcd.setCursor(0, 1);
lcd.print("A0:");
lcd.print(analogRead(A0));
delay(200);
lcd.print(" A1:");
lcd.print(analogRead(A1));
lcd.print(" ");

break;

case 1:
    lcd.home();
    lcd.print("Calibration");
    break;

case 2:
    lcd.home();
    lcd.print("Set temperature");
    break;

case 3:
    lcd.home();
    lcd.print("Set alarm");
    break;

case 4:
    lcd.home();
    lcd.print("Titration curve");
    break;

case 5:
    lcd.home();
    lcd.print("Measure");
    lcd.setCursor(0, 1);
    lcd.print("over time");
    break;

case 6:
    lcd.home();
    lcd.print("Download data");
    break;

case 7:
    lcd.home();
    lcd.print("Backlight");
    break;
}

//calibration session
while(mainControl==1) {
    if (ctrl == '0') {
        ctrl = '1';
        mainControl = 0;
        mainList = 0;
        lcd.clear();
        break;
    }

    int temp_nPoints = 2;
    int temp_analog[10];
    double temp_ph[10];

    int digits[] = {
        0, 7, 0, 0
    };
    boolean choice = false;

    if (polyn==true) {
        lcd.clear();
        lcd.print("Disable calibr.?");
        while (ctrl=='1') {
            if (choice==false) {
                lcd.setCursor(0, 1);
                lcd.print("No ");
            }
            else {
                lcd.setCursor(0, 1);
                lcd.print("Yes");
            }

            switch (bState()) {
            case UPPER_BUTTON:
                choice=!choice;
                break;
            case LOWER_BUTTON:
                choice=!choice;
                break;
            case BACK_BUTTON:
                lcd.clear();
            }
        }

        ctrl = '0';
        break;

        case OK_BUTTON:
            ctrl = '0';
            if (choice==true) polyn = false;
            break;
        }
    }

    lcd.clear();
    lcd.print("Load latest?");
    while (ctrl=='1') {

        if (choice==false) {
            lcd.setCursor(0, 1);
            lcd.print("No ");
        }
        else {
            lcd.setCursor(0, 1);
            lcd.print("Yes");
        }

        switch (bState()) {
        case UPPER_BUTTON:
            choice=!choice;
            break;
        case LOWER_BUTTON:
            choice=!choice;
            break;
        case BACK_BUTTON:
            lcd.clear();
            ctrl = '0';
            break;
        case OK_BUTTON:
            if (choice==false) ctrl = '2';
            else ctrl='7';
            break;
        }
    }

    lcd.clear();
    lcd.print("How many points?");
    while (ctrl=='2') {
        switch (bState()) {
        case UPPER_BUTTON:
            if (temp_nPoints<10) temp_nPoints++;
            break;
        case LOWER_BUTTON:
            if (temp_nPoints>2) temp_nPoints--;
            break;
        case BACK_BUTTON:
            lcd.clear();
            ctrl='1';
            break;
        case OK_BUTTON:
            ctrl = '3';
            break;
        }

        lcd.setCursor(0,1);
        lcd.print(temp_nPoints);
        lcd.print(' ');

        lcd.clear();
        lcd.print("Point");
        lcd.setCursor(8, 0);
        lcd.print("pH:");
        lcd.setCursor(0, 1);
        lcd.print("analog in: ");

        if(ctrl=='3') {
            for (int i=0; ; i++) {
                while(ctrl=='3') {
                    lcd.setCursor(5, 0);
                    lcd.print(i+1);
                    lcd.print(' ');

                    lcd.setCursor(11, 1);
                    lcd.print(readAnalog(0));

                    selectNumber(digits, 4, 2, 11, 0, &ctrl);
                }

                if(ctrl == 'f') {
                    temp_analog[i] = analogRead(0);
                    temp_ph[i] = composeNumber(digits, 4, 2);
                    if (i == temp_nPoints-1) {
                        ctrl = '4';
                        break;
                    }
                }
            }
        }
    }
}

```

```

    }
    ctrl = '3';
    continue;
}
if (ctrl == 'b') {
    if (i==0) break;
    ctrl = '3';
    i-=2;
    continue;
}
}
if (ctrl=='b') ctrl = '2';

if (ctrl == '4') {
    Serial.println("Points matrix:");
    for (int i=0; i<temp_nPoints; i++) {
        Serial.print(temp_analog[i]);
        Serial.print('\t');
        Serial.println(temp_ph[i]);
    }
    ctrl = '5';
}
}

lcd.clear();
lcd.print("Save to EEPROM?");
while (ctrl=='5') {
    if (choice==false) {
        lcd.setCursor(0, 1);
        lcd.print("No ");
    }
    else {
        lcd.setCursor(0, 1);
        lcd.print("Yes");
    }
}

switch (bState()) {
case UPPER_BUTTON:
    choice=!choice;
    break;
case LOWER_BUTTON:
    choice=!choice;
    break;
case BACK_BUTTON:
    lcd.clear();

    ctrl = '1';
    break;

case OK_BUTTON:
    if (choice==false) ctrl = '8';
    else ctrl='6';
}
}

//saves points to EEPROM
Serial.print("Free ram: ");
Serial.println(FreeRam());
if (ctrl == '6') {
    EEPROM.write(CALIB_N_ADDR, temp_nPoints);

    //saves analog values
    int diff;
    int addr = CALIB_PTS_ADDR;
    //int serAddr = 1;
    for (int i=0; i<temp_nPoints; i++) {
        diff = temp_analog[i];
        for (int j=0; j<5; j++) {
            if (diff >= 255) {
                EEPROM.write(addr, 255);
                diff = diff - 255;
                addr++;
            }
            else {
                EEPROM.write(addr, diff);
                diff = 0;
                addr++;
            }
        }
    }

    //serial print
    int sum=0;
    for (int j=addr-5; j<addr; j++) {
        int a = EEPROM.read(j);
        sum+=a;
        Serial.print(j);
        Serial.print("->");
        Serial.print(a);
        Serial.print('\t');
    }
    Serial.print("sum=");
    Serial.println(sum);
}

//saves pH values
for (int i=0; i<temp_nPoints; i++) {
    int intg = floor(temp_ph[i]);
    int dec = ceil((temp_ph[i]-intg)*100);

    int a;
    int b;
    EEPROM.write(addr, intg);
    a=EEPROM.read(addr);
    Serial.print(addr);
    Serial.print("->");
    Serial.print(a);
    Serial.print('\t');
    addr++;

    EEPROM.write(addr, dec);
    b=EEPROM.read(addr);
    Serial.print(addr);
    Serial.print("->");
    Serial.print(b);
    Serial.print("\tsum=");

    double sum=a+ (double)b/100;
    Serial.println(sum);
    addr++;
}
ctrl = '8';
}

//fills the arrays with data from EEPROM
if (ctrl=='7') {
    temp_nPoints = EEPROM.read(0);

    //reads saved analog inputs
    int addr=1;
    for(int i=0; i<temp_nPoints; i++) {
        int sum=0;
        for(int j=0; j<5; j++) {
            sum+=EEPROM.read(addr);
            addr++;
        }
        temp_analog[i]=sum;

        Serial.print(i);
        Serial.print("->");
        Serial.println(temp_analog[i]);
    }

    //reads saved pHs
    for(int i=0; i<temp_nPoints; i++) {
        temp_ph[i] = EEPROM.read(addr) + (double)EEPROM.read(addr+1)/100;
        addr+=2;

        Serial.print(i);
        Serial.print("->");
        Serial.println(temp_ph[i]);
    }
    ctrl='8';
}

lcd.clear();
lcd.print("Processing...");
//the marvelous gauss' elimination algorithm for calculating coefficients
if (ctrl=='8') {
    const int maxp = temp_nPoints;
    double matrix[maxp][maxp+1];

    //fills matrix with data
    for (int i=0; i<maxp; i++) {
        for (int j=0; j<maxp; j++) {
            matrix[i][j]=pow(temp_analog[i], j);
        }
        matrix[i][maxp]=temp_ph[i];
    }

    //prints matrix
    for (int k=0; k<maxp; k++) {
        for (int j=0; j<maxp+1; j++) {
            Serial.print(matrix[k][j]);
            Serial.print('\t');
        }
        Serial.print('\n');
    }
    Serial.print('\n');

    //gauss' elimination algorithm
    for (int i=0; i<maxp-2; i++) {

        //if all the column is 0, skips a cycle
        if(matrix[i][i]==0) {
            int a=0;
            for (int j=i+1; j<=maxp-1; j++) {
                if (matrix[j][i]!=0) {
                    a++;
                }
            }
        }
    }
}
}

```

```

        break;
    }
}
if (a==0) {
    Serial.println("skipping");
    continue;
}

Serial.println("swapping");
//if only the first element is 0, swaps the row with another one
for (int j=i+1; j<=maxp-1; j++) {
    if (matrix[j][i]!=0) {
        for (int k=i; k<=maxp; k++) {
            double buf = matrix[j][k];
            matrix[j][k] = matrix [i][k];
            matrix[i][k] = buf;
        }
        break;
    }
}

//zeroes a non-zero element by summing its row to a proper multiple of the first row
for (int j=i+1; j<=maxp-1; j++) {
    if (matrix[j][i]!=0) {
        double coeff = matrix[j][i]/matrix[i][i];
        for(int k=i; k<=maxp; k++) matrix[j][k]-
        =matrix[i][k]*coeff;
    }
}

//prints matrix
for (int k=0; k<maxp; k++) {
    for (int j=0; j<maxp+1; j++) {
        Serial.print(matrix[k][j]);
        Serial.print('\t');
    }
    Serial.print('\n');
}
Serial.print('\n');

//inverted gauss' elimination algorithm
for (int i=maxp-1; i>=1; i--) {
    if(matrix[i][i]==0) {
        int a=0;
        for (int j=i-1; j>=0; j--) {
            if (matrix[j][i]!=0) {
                a++;
                break;
            }
        }
        if (a==0) {
            Serial.println("skipping");
            continue;
        }

        Serial.println("swapping");

        for (int j=i-1; j>=0; j--) {
            if (matrix[j][i]!=0) {
                for (int k=i; k>=0; k--) {
                    double buf = matrix[j][k];
                    matrix[j][k] = matrix [i][k];
                    matrix[i][k] = buf;
                }
                double buf = matrix[j][maxp];
                matrix[j][maxp] = matrix[i][maxp];
                matrix[i][maxp] = matrix[j][maxp];

                break;
            }
        }

        for (int j=i-1; j>=0; j--) {
            if (matrix[j][i]!=0) {
                double coeff = matrix[j][i]/matrix[i][i];
                for(int k=i; k>=0; k--) matrix[j][k]-
                =matrix[i][k]*coeff;
                matrix[j][maxp]-=matrix[i][maxp]*coeff;
            }
        }
        //prints matrix
        for (int k=0; k<maxp; k++) {
            for (int j=0; j<maxp+1; j++) {
                Serial.print(matrix[k][j]);
                Serial.print('\t');
            }
            Serial.print('\n');
        }
        Serial.print('\n');
    }
}

}

//reduces all the coefficients to 1
boolean err=false;
for (int i=0; i<=maxp-1; i++) {
    if(matrix[i][i]!=1 && matrix[i][i]!=0) {
        matrix[i][maxp]=matrix[i][maxp]/matrix[i][i];
        matrix[i][i]=matrix[i][i]/matrix[i][i];
    }
    else if(matrix[i][i]==0) {
        err=true;
        break;
    }
}
if (err==true) {
    Serial.println("division by zero detected");
    lcd.clear();
    lcd.print("unexpected");
    lcd.setCursor(0, 1);
    lcd.print("failure");
    delay(2000);

    ctrl = '1';
    mainControl = 0;
    mainList = 0;
    break;
}

//prints matrix
for (int k=0; k<maxp; k++) {
    for (int j=0; j<maxp+1; j++) {
        Serial.print(matrix[k][j]);
        Serial.print('\t');
    }
    Serial.print('\n');
}
Serial.print('\n');

//assigns to the coefficient matrix the results and prints equation
for (int i=0; i<maxp; i++) coeff[i]=matrix[i][maxp];

//prints coefficients
Serial.print ("y = ");
for (int i=0; i<maxp; i++) {
    Serial.print (coeff[i], DEC);
    Serial.print("analog");
    Serial.print(i);
    Serial.print(" + ");
    Serial.print('\n');
}
polyn = true;
nCcoeff = temp_nPoints;
ctrl = '0';
}

//temperature setting session
while(mainControl==2) {
    int digits[] = {
        2, 9, 8, 1, 5
    };

    lcd.clear();
    lcd.print("Set temperature:");

    while (ctrl == '1') selectNumber (digits, 5, 2, 0, 1, &ctrl);

    if (ctrl == 'f') {
        temperature = composeNumber(digits, 5, 2);

        lcd.clear();
        ctrl = '1';
        mainList = 0;
        mainControl = 0;
    }

    if (ctrl == 'b') {
        lcd.clear();
        exit();
    }
}

//alarm setting session
while (mainControl==3) {
    int digits[] = {
        0, 7, 0, 0
    };
    boolean choice=false;

    lcd.clear();
    lcd.print("Activate alarm?");

    while (ctrl=='1') {
        if (choice==false) {

```

```

    lcd.setCursor(0, 1);
    lcd.print("No ");
}
else{
    lcd.setCursor(0, 1);
    lcd.print("Yes");
}

switch (bState()) {
case UPPER_BUTTON:
    choice=!choice;
    break;
case LOWER_BUTTON:
    choice=!choice;
    break;
case BACK_BUTTON:
    lcd.clear();
    alarm=false;
    mainControl = 0;
    mainList = 0;
    ctrl = '0';
    break;

case OK_BUTTON:
    if (choice==false) {
        alarm=false;

        lcd.clear();
        mainList=0;
        mainControl=0;
        ctrl = '0';
    }
    else {
        alarm=true;

        ctrl='2';
    }
}
if (ctrl=='0') {
    ctrl = '1';
    break;
}

lcd.clear();
lcd.print("Select limit:");
while (ctrl=='2') {
    if (choice==true) {
        lcd.setCursor(0, 1);
        lcd.print("Upper");
    }
    else {
        lcd.setCursor(0, 1);
        lcd.print("Lower");
    }
}

switch (bState()) {
case UPPER_BUTTON:
    choice=!choice;
    break;
case LOWER_BUTTON:
    choice=!choice;
    break;
case BACK_BUTTON:
    lcd.clear();
    ctrl = '1';
    break;

case OK_BUTTON:
    if (choice==true) {
        upperLimit=true;
        ctrl='3';
    }
    else {
        upperLimit=false;
        ctrl='3';
    }
}

lcd.clear();
lcd.print("Select ph limit:");
while(ctrl=='3') selectNumber(digits, 4, 2, 0, 1, &ctrl);

if (ctrl=='f') {
    phLimit = composeNumber(digits, 4, 2);
    ctrl = '1';
    mainList = 0;
    mainControl = 0;
}

if (ctrl=='b') ctrl = '2';
lcd.clear();
}

//titration curve session
if (mainControl==4) {
    int digits [] = {
        0, 0, 0
    };

    double ml=0;
    int nPoints=0;
    int addr=TITR_PTS_ADDR;

    double infl[3];
    int n_infl=0;
    boolean past_infl=false;

    double prev_deriv;
    double prev_ml;
    double prev_ph;

    ofstream logfile;
    char buf[80];
    char filename[] = "TITR000.CSV";
    if(SD_av==true) {
        int val = EEPROM.read(SD_TITR_NAME_ADDR);
        if(val!=255) EEPROM.write(SD_TITR_NAME_ADDR, val+1);
        else EEPROM.write(SD_TITR_NAME_ADDR, 0);

        int a, b, c;

        if (val!=0) {
            a = val/100;
            b = (val-a*100)/10;
            c = (val-b*10-a*100);

            filename[4] = '0' + a;
            filename[5] = '0' + b;
            filename[6] = '0' + c;
        }
        if(sd.exists(filename)) sd.remove(filename);
        logfile.open(filename);

        if (!logfile.is_open()) {
            SD_av = false;

            lcd.clear();
            lcd.print("Unexpected");
            lcd.setCursor(0, 1);
            lcd.print("failure");
            delay(2000);
            lcd.clear();
            lcd.print("Using EEPROM");
            delay(2000);
        }
    }
    if(SD_av==true) {
        ofstream bout (buf, sizeof(buf));
        bout << filename << "\nmL\tpH\n";
        logfile << buf << flush;
    }

    lcd.clear();
    lcd.print("ph: ");
    lcd.setCursor(0, 1);
    lcd.print("mL: ");

    while (1) {
        while (ctrl=='1') {
            ph = readPh();
            lcd.setCursor(4, 0);
            lcd.print(ph);
            selectNumber(digits, 3, 2, 4, 1, &ctrl);
            alarmCheck(ph);
        }

        if (ctrl=='f') {
            ml+=composeNumber(digits, 3, 2);

            if(SD_av==false){
                nPoints++;

                int intg;
                int dec;

                intg = floor(ml);
                dec = ceil((ml-intg)*100);
                EEPROM.write(addr, intg);
                addr++;
                EEPROM.write(addr, dec);
                addr++;

                intg = floor(ph);
                dec = ceil((ph-intg)*100);
                EEPROM.write(addr, intg);
                addr++;
                EEPROM.write(addr, dec);
                addr++;
            }
        }
    }
}

```

```

if(nPoints==TITR_MAX_POINTS) {
    EEPROM.write(TITR_N_ADDR, nPoints);

    lcd.clear();
    lcd.print("Max point number");
    lcd.setCursor(0,1);
    lcd.print("achieved");

    exit();
    break;
}
else {
    if (nPoints==0) {
        prev_ph=ph;
        prev_ml=ml;
        prev_deriv=0;
    }
    else if (n_infl<3) {
        double deriv=(ph-prev_ph)/(ml-prev_ml);
        if (past_infl==false && abs(deriv) < abs(prev_deriv)){
            infl[n_infl] = prev_ml;
            n_infl++;
            past_infl = true;
        }
        else if (past_infl==true && abs(deriv) > abs(prev_de-
riv)) past_infl=false;

        prev_deriv = deriv;
        prev_ph = ph;
        prev_ml = ml;
    }
    nPoints++;

    logfile << ml << '\t' << ph << endl << flush;
    Serial.print("Free ram: ");
    Serial.println(FreeRam());
    if (!logfile) {
        lcd.clear();
        lcd.print("Unexpected");
        lcd.setCursor(0, 1);
        lcd.print("failure");
        delay(2000);
        lcd.clear();
        lcd.print("Using EEPROM");
        delay(2000);
    }
    exit();
    break;
}
}
lcd.clear();
lcd.print("Saved");
delay(1000);
lcd.clear();
lcd.print("ph: ");
lcd.setCursor(0, 1);
lcd.print("mL: ");

ctrl='1';
}

if (ctrl=='b') {
    if (SD_av==false) {
        EEPROM.write(TITR_N_ADDR, nPoints);

        lcd.clear();
        lcd.print("Data saved");
        lcd.setCursor(0, 1);
        lcd.print("to EEPROM");
        delay(2000);
    }
    else {
        ofstream bout(buf, sizeof(buf));
        bout << "Inflection point(s)\n";
        for (int i=0; i<n_infl; i++) {
            bout<<infl[i]<<endl;
        }
        logfile << buf << flush;
        logfile.close();

        lcd.clear();
        lcd.print("Data saved");
        lcd.setCursor(0, 1);
        lcd.print("to SD card");
        delay(2000);
    }
}

lcd.clear();
exit();
break;
}
}

//measure over time session
while (mainControl==5) {

int digits[] = {
    0, 0, 0
};

int period;
int n=0;
unsigned long int prevMillis;

int nPoints=0;
int addr=SAMPLE_PTS_ADDR;

ofstream logfile;
char buf[80];
char filename[] = "SAMP000.CSV";
if(SD_av==true) {
    int val = EEPROM.read(SD_SAMPLE_NAME_ADDR);
    if(val!=255) EEPROM.write(SD_SAMPLE_NAME_ADDR, val+1);
    else EEPROM.write(SD_SAMPLE_NAME_ADDR, 0);

    int a, b, c;

    if (val!=0) {
        a = val/100;
        b = (val-a*100)/10;
        c = (val-b*10-a*100);

        filename[4] = '0' + a;
        filename[5] = '0' + b;
        filename[6] = '0' + c;
    }
    if(sd.exists(filename)) sd.remove(filename);
    logfile.open(filename);

    if (!logfile.is_open()) {
        SD_av = false;

        lcd.clear();
        lcd.print("Unexpected");
        lcd.setCursor(0, 1);
        lcd.print("failure");
        delay(2000);
        lcd.clear();
        lcd.print("Using EEPROM");
        delay(2000);
    }
}
if(SD_av==true) {
    ofstream bout (buf, sizeof(buf));
    bout << filename << "\nseconds\tpH\n";
    logfile << buf << flush;
}

lcd.clear();
lcd.print("Sample period(s)");

while (ctrl=='1') selectNumber(digits, 3, 0, 0, 1, &ctrl);
delay(500);

if (ctrl=='f') {
    period = composeNumber(digits, 3, 0);
    if(period==0) {
        lcd.clear();
        lcd.print("Please set a");
        lcd.setCursor(0, 1);
        lcd.print("non-zero period");
        delay(2000);
        lcd.clear();
        lcd.print("Sample period(s)");

        ctrl = '1';
    }
    else if(period>255 && SD_av==false) {
        lcd.clear();
        lcd.print("Too big to");
        lcd.setCursor(0, 1);
        lcd.print("save to EEPROM");
        delay(2000);
        lcd.clear();
        lcd.print("Sample period(s)");

        ctrl = '1';
    }
    else {
        if (SD_av==false) EEPROM.write(SAMPLE_PERIOD_ADDR, period);
        ctrl='2';
    }
}

if (ctrl=='b') {
    lcd.clear();

    exit();
}

long unsigned int startMillis=millis();

```



```

static boolean state;
currentMillis = millis();

if (currentMillis-previousMillis>500) {
  previousMillis = currentMillis;
  state = !state;

  if (state==false) {
    lcd.setCursor(column, row);
    for(int i=0; i<=nDigits-nDecs-1; i++) lcd.print(p[i]);
    lcd.print(".");
    for (int i=nDigits-nDecs; i<=nDigits-1; i++) lcd.print(p[i]);

    if (currDigit>=nDigits-nDecs) {
      lcd.setCursor(column+currDigit+1, row);
      lcd.write(" ");
    }
    else {
      lcd.setCursor(column+currDigit, row);
      lcd.write(" ");
    }
  }
  else {
    lcd.setCursor(column, row);
    for(int i=0; i<=nDigits-nDecs-1; i++) lcd.print(p[i]);
    lcd.print(".");
    for (int i=nDigits-nDecs; i<=nDigits-1; i++) lcd.print(p[i]);
  }
}

/*returns a double type number starting from an array of digits,
 * giving the number of digits and of decimals as arguments.
 */
double composeNumber (int *p, int nDigits, int nDecs) {
  double num=0;

  for (int i=0; i<=nDigits-1; i++) num+=(double)p[i]*pow(10,
(nDigits-nDecs-1)-i);

  return num;
}

/*converts the signal to ph according to calibration parameters
 * and returns it;
 */
double readPh() {
  static unsigned long previousMillis;
  unsigned long currentMillis = millis();

  static double prevPh;

  if (currentMillis-previousMillis>500) {
    double ph;
    int sig1 = 2*analogRead(A15);
    int sig2 = 1023;
    int ph1 = 7;
    int ph2 = 14;
    if (polyn==false) ph = (double)(14-7)*(analogRead(A14)-
sig1)/(sig2-sig1) + 7;
    else {
      double sum = 0;
      double x = analogRead(A14);
      for(int i=0; i<Ncoeff; i++) sum += coeff[i]*pow(x, i);
      ph = sum;
    }
    previousMillis = currentMillis;
    prevPh = ph;
    return ph;
  }
  else return prevPh;
}

int readAnalog(int pin) {
  static unsigned long previousMillis;
  unsigned long currentMillis = millis();

  static double prevAnalog;

  if (currentMillis-previousMillis>500) {
    double analog = analogRead(pin);
    previousMillis = currentMillis;
    prevAnalog = analog;
    return analog;
  }
  else return prevAnalog;
}

/*returns which button is being pressed (same numbers as pins)
 * and returns 0 if it is being pressed without releasing it before.
 */
int bState () {
  static int prevStatus;

  if (digitalRead(UPPER_BUTTON)==LOW && prevStatus!=UPPER_BUTTON) {
    prevStatus = UPPER_BUTTON;
    return UPPER_BUTTON;
  }
  else if (digitalRead(LOWER_BUTTON)==LOW && prevStatus!=LOWER_
BUTTON){
    prevStatus = LOWER_BUTTON;
    return LOWER_BUTTON;
  }
  else if (digitalRead(OK_BUTTON)==LOW && prevStatus!=OK_BUTTON){
    prevStatus = OK_BUTTON;
    return OK_BUTTON;
  }
  else if (digitalRead(BACK_BUTTON)==LOW && prevStatus!=BACK_
BUTTON){
    prevStatus = BACK_BUTTON;
    return BACK_BUTTON;
  }
  else if (digitalRead(UPPER_BUTTON)==LOW && prevStatus==UPPER_
BUTTON){
    return 0;
  }
  else if (digitalRead(LOWER_BUTTON)==LOW && prevStatus==LOWER_
BUTTON){
    return 0;
  }
  else if (digitalRead(OK_BUTTON)==LOW && prevStatus==OK_BUTTON){
    return 0;
  }
  else if (digitalRead(BACK_BUTTON)==LOW && prevStatus==BACK_
BUTTON){
    return 0;
  }
  prevStatus = 0;
  return 0;
}

/*if alarm is activated, keeps the ph limits under control and
 * beeps if they are crossed
 */
void alarmCheck(double ph) {
  static unsigned long previousMillis;
  static boolean beep=true;
  static int duration=0;
  currentMillis = millis();

  if (alarm==true) {
    if (upperLimit==true && ph > phLimit){
      if (beep==true) {
        lcd.setCursor(15, 0);
        lcd.print('*');
      }
      else {
        lcd.setCursor(15, 0);
        lcd.print(' ');
      }
    }
    if (currentMillis-previousMillis>500){
      previousMillis=currentMillis;
      beep = !beep;
      if(beep==true) tone(ALARM_PIN, 2400);
      else noTone(ALARM_PIN);
      duration++;
    }
  }

  if (duration==10) {
    alarm=false;
    noTone(ALARM_PIN);
    duration=0;
  }
  return;
}

if (upperLimit==false && ph < phLimit) {
  if (beep==true) {
    lcd.setCursor(15, 0);
    lcd.print('*');
    digitalWrite(ALARM_PIN, HIGH);
  }
  else {
    lcd.setCursor(15, 0);
    lcd.print(' ');
  }
}

if (currentMillis-previousMillis>500){
  previousMillis=currentMillis;
  beep = !beep;
}

```

```
        duration++;
    }

    if (duration==10) {
        alarm=false;
        duration=0;
    }
    return;
}
lcd.setCursor(15, 0);
lcd.print('*');
}

}

//resets control variables
void exit() {
    ctrl='l';
    mainList=0;
    mainControl=0;
}
```

Appendice C

Istruzioni

Avvio

Per avviare il pH-metro, alimentare Arduino. Si possono utilizzare indistintamente un cavo USB o un alimentatore da 9 V e almeno 500 mA.

All'avvio il programma effettua il controllo della presenza della scheda SD. Se essa non è presente, viene utilizzata la EEPROM.

Navigazione

Per navigare nei menu si possono usare i pulsanti upper e lower.

Per confermare un'opzione si usa il tasto ok; per annullarla, il tasto back.

Schermata principale

La schermata principale visualizza il pH e la corrispondente tensione sulla membrana, calcolata sulla temperatura impostata dall'operatore (di default è 298,15 K).

Si può entrare nel menu principale in qualsiasi momento premendo il tasto upper, e scorrere le opzioni con upper o lower.

Menu principale

Il menu principale si compone delle seguenti voci:

- "Calibrazione": permette di effettuare la calibrazione polinomiale. Selezionare se caricare l'ultima calibrazione effettuata. Se se ne vuole fare una nuova, viene richiesto il numero di punti con cui calibrare (non più di 10). A questo punto, per ogni punto è richiesto di immergere la sonda in uno standard e di comunicare al pH-metro il suo pH. Per confermare il valore di pH e registrare il punto, si preme ok e si ripete la procedura per tutti i punti. Tra un punto e l'altro, si scolleghi l'elettrodo e lo si lavi. Alla fine sarà chiesto se si desidera salvare su EEPROM la calibrazio-

ne.

- "Temperature": permette di impostare la temperatura a cui si trova la soluzione, in modo che il calcolo della tensione sia corretto.
- "Alarm": permette di attivare o disattivare l'allarme. Se lo si attiva, chiede se impostare un limite superiore o inferiore, dopodiché di immettere il limite.
- "Titration curve": permette di comporre una curva di titolazione salvando i mL aggiunti e il pH su SD o EEPROM. All'apertura della sessione, comunicare una aggiunta di 0 mL per creare il primo punto con il pH iniziale. Una volta confermato il primo punto, è richiesto di immettere il secondo e così via. Ad ogni punto immettere una aggiunta di titolante nella soluzione e comunicare il volume immesso al pH-metro. Quando la titolazione è conclusa premere back per salvare i dati. Se si usa la scheda SD è effettuato anche il calcolo dei punti di flesso, inserito nel file coi dati.
- "Measure over time": permette di monitorare il pH nel tempo. All'entrata nella sessione è richiesto di immettere il periodo di campionamento, dopodiché inizierà il monitoraggio. Per concluderlo, premere back o ok; i dati saranno salvati su SD o EEPROM.
- "Download data": invia alla porta seriale i dati salvati su EEPROM della curva di titolazione e del monitoraggio. Per leggerli si consiglia l'installazione sul computer di un monitor seriale.
- "Backlight": permette di accendere o spegnere la retroilluminazione del display LCD.

Calibrazione manuale

Entrare nella schermata principale. Immergere l'elettrodo a vetro in una soluzione standard a pH 7 ed effettuare la regolazione del trimmer dell'offset, con un cacciavite a punta piatta, finché il pH letto non è 7.

Scollegare l'elettrodo e lavarlo.

Ricollegarlo e immergerlo in una soluzione standard a pH 4 o 10, a seconda del range in cui si desidera operare, e regolare il trimmer del guadagno finché il pH letto non è 4 o 10.

La calibrazione è da effettuare alla stessa temperatura a cui sono fissati i valori di pH degli standard.

Bibliografia

Autori di Wikipedia, *Arduino*, Wikipedia, L'Enciclopedia libera, 2013, consultato il 11/06/2013 all'indirizzo [http://it.wikipedia.org/wiki/Arduino_\(hardware\)](http://it.wikipedia.org/wiki/Arduino_(hardware))

Autori di Wikipedia, *Metodo di eliminazione di Gauss*, Wikipedia, L'Enciclopedia libera, 2013, consultato il 15/06/2013 all'indirizzo http://it.wikipedia.org/wiki/Eliminazione_di_Gauss

Autori di Wikipedia, *Root Effect*, Wikipedia, L'Enciclopedia libera, 2013, consultato il 17/06/2013 all'indirizzo http://en.wikipedia.org/wiki/Root_effect

Autori di Wikipedia, *Teoria acido-base di Lewis*, Wikipedia, L'Enciclopedia libera, 2013, consultato il 17/06/2013 all'indirizzo http://it.wikipedia.org/wiki/Teoria_acido-base_di_Lewis

A. Lombardi, *In laboratorio – V esperienza: le titolazioni potenziometriche: titolazione acido debole – base forte e determinazione della costante di acidità*, Università degli Studi di Napoli Federico II, 2013, consultato il 17/06/2013 all'indirizzo <http://www.federica.unina.it/smfncchimica-generale-ed-inorganica-smfn/titolazioni-potenziometriche-titolazione-acido-debole/>

Falk E., *all-about-ph.com*, 2013, consultato il 10/06/2013 all'indirizzo <http://www.all-about-ph.com>

Ferrari M., *Elettrochimica*, ITIS Leonardo da Vinci di Carpi, Modena, 2013

Ferrari M., *Tecniche analitiche potenziometriche*, ITIS Leonardo da Vinci di Carpi, Modena, 2013

Rubino C., Venzaghi I., Cozzi R., *Stechio & Lab, Le basi dell'analisi chimica 1. Stechiometria*, Bologna, Zanichelli, 2001

PatrickJMT, *Gaussian Elimination*, YouTube, 2012, visualizzato il 13/06/2013 all'indirizzo <http://www.youtube.com/watch?v=2j5Ic2V7wq4>

---, *I Sistemi d'Acquisizione Dati*, Dipartimento di Ingegneria Industriale e dell'Informazione dell'Università di Napoli, 2004/2005

Siti utili

<http://www.all-about-ph.com>

Consultato anche per la stesura di questo fascicolo, offre una trattazione del pH semplice e che non richiede troppe basi di chimica, comunque completa ed esauriente.

<http://www.arduino.cc>

Il sito è il punto di riferimento per chi vuole utilizzare Arduino: è presente un catalogo delle varie versioni di Arduino e delle shield supportate dal progetto ufficiale, nonché un'ampia documentazione comprendente tutorial che introducono all'uso della scheda con rara immediatezza.

<http://www.learncpp.com>

Un ottimo sito per iniziare a programmare in C++.

