

Rotore con Arduino.

PREMESSE PROGETTUALI

Con ogni probabilità l'acquisto di un oggetto commerciale fa risparmiare tempo e denaro, ma per chi come me ha la passione per l'autocostruzione forse questo articolo può dare degli spunti.

L'uso previsto è con antenne HF/VHF che non richiedano estrema precisione di puntamento, non ho provato a fare inseguimento di satelliti od altro, quindi non sono in grado di sapere se questo rotore è adatto a tale utilizzo.

Anni fa quando ho installato la mia stazione avevo realizzato un supporto d'antenna in cui il rotore era costituito da un motoriduttore comandato da un motorino di tergitristallo, mentre un semplice potenziometro multigiri calettato sull'asse era utilizzato per leggere la posizione dell'antenna.

Dopo alcuni anni spinto dalle possibilità del programma di log che utilizzo, ho deciso di costruire qualcosa un po' più al passo dei tempi che andasse oltre il classico controllo con galvanometro e pulsanti.

In internet ho trovato vari progetti ma nessuno mi è sembrato facilmente realizzabile, vuoi per l'uso di strani uP, che l'utilizzo di moduli bussola od altro.

Non volevo poi stravolgere la parte sul tetto per evitare importanti interventi meccanici.

Avevo iniziato a studiare l'uso dei PIC quando a pranzo dei colleghi mi hanno parlato di Arduino, si tratta di un progetto italiano open source che vuole avvicinare chiunque al mondo dell'autocostruzione evitandogli di studiare il linguaggio macchina dei microcontrollori e permettendo all'utilizzatore di focalizzarsi solo sulla sua idea.

Analizzando le caratteristiche del modello 2009 rilevavo la presenza di ingressi analogici costituiti da ADC a 12 bit ed escursione di ingresso 0-5V, potevano quindi fare al caso mio per leggere la posizione del potenziometro del rotore.

Da un primo semplice giro di calcoli c'era un problema; ipotizzando di applicare i 5V agli estremi del potenziometro multigiri sull'asse, avrei ottenuto una risoluzione pari a:

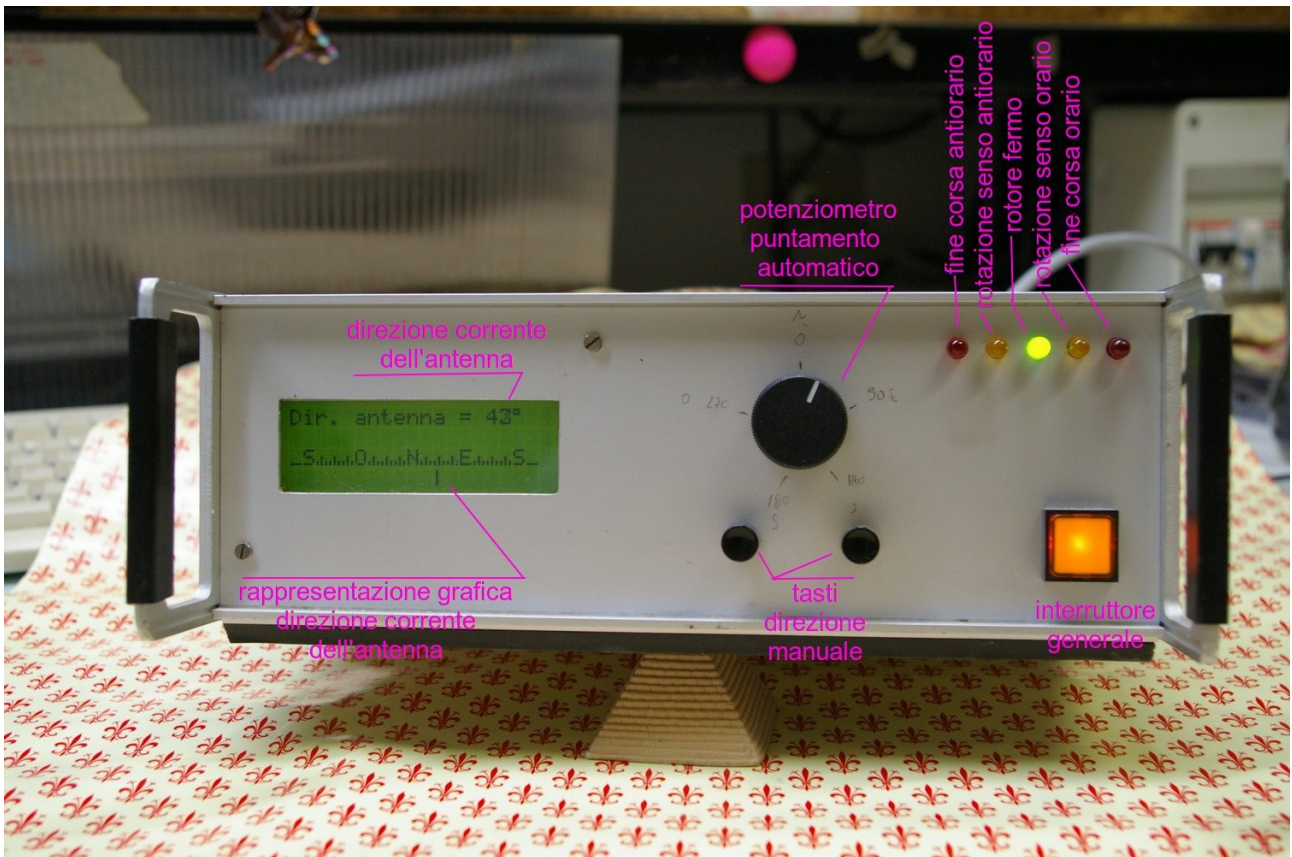
$$2^{12}=4092 \text{ step}$$

$$4092/10=409 \text{ step/giro}$$

$$409/360=1,1 \text{ step/}^\circ$$

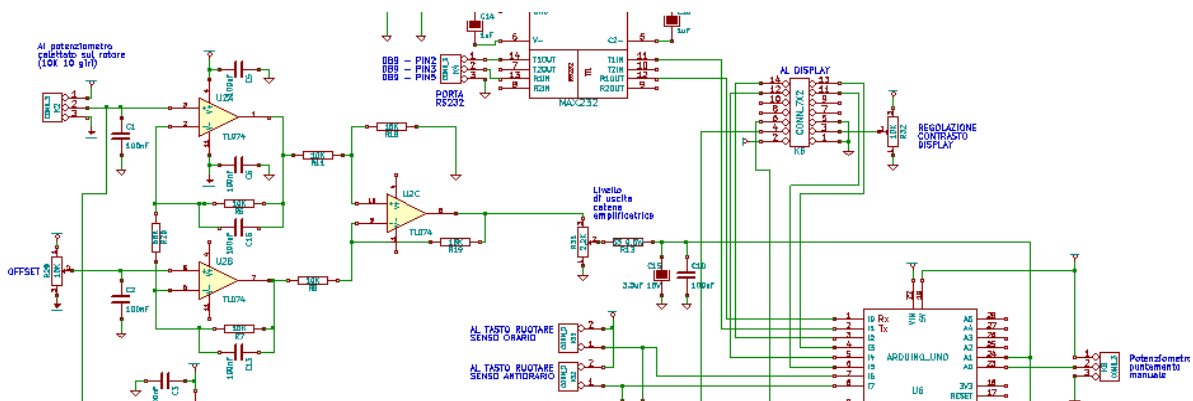
In queste condizioni tra ogni singolo livello dell'ADC si ha una differenza di 1,22mV, considerando la lunghezza del cavo che collega il controllo con il rotore questo avrebbe facilmente captato del rumore, quindi la cifra da visualizzare avrebbe "ballato" notevolmente e sul display non avrei ottenuto un'indicazione molto precisa.

Volevo inoltre che l'escursione totale dell'antenna fosse dei classici 360° più un "sormonto" di 45°+45° ai due estremi sud, quindi un totale di 450° di rotazione.



LA SCHEDA MADRE

Era quindi necessario ottenere una risoluzione migliore quindi dovevo in qualche modo amplificare il segnale proveniente dal potenziometro.

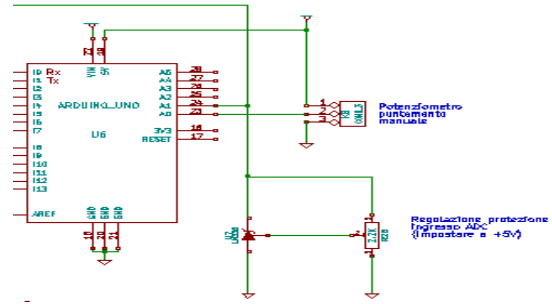


Questa è la funzione principale della scheda madre, per fare questo sono ricorso “all'amplificatore per strumentazione” costituito da tre operazionali (U2A,U2B ed U2C).

Ho così ottenuto l'amplificazione della tensione proveniente dal potenziometro del rotore e l'adattamento all'ingresso dell'Arduino sommandogli una tensione di offset tramite R29, in questo modo l'escursione del segnale di ingresso dell'ADC inizia da 0V (tutto in senso antiorario).

Per mezzo di R31 si limita invece la tensione a +5V (tutto in senso orario) andando a sfruttare l'intera escursione accettata dall'ingresso e quindi avere una risoluzione migliore pari a $4092/450 = 9$ step/°.

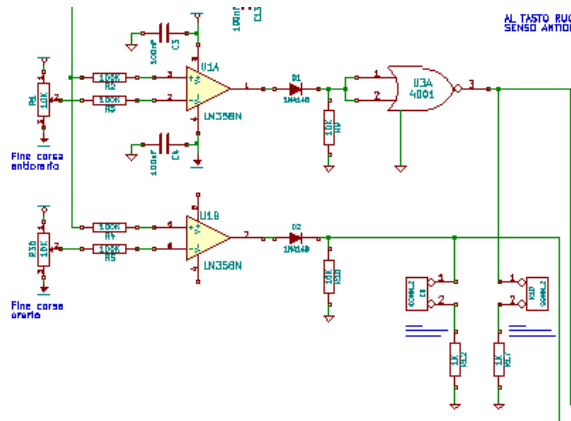
E' necessario fare attenzione al fatto che il microcontrollore non tollera tensioni maggiori a 5,5V, è stato così predisposto un circuito di protezione costituito da U7 ed R28 da tarare opportunamente.



Sebbene sia possibile realizzare dei finecorsa di tipo software ho deciso di mantenerli in logica cablata, questo evita che un blocco del programma possa portare ad una rotazione incontrollata con possibile danneggiamento ai cavi sul tetto.

Questa parte è costituita da due comparatori (U1A ed U1B) che leggono la tensione del potenziometro calettato sul rotore e tramite i due trimmer R1 ed R30 si fissano i fondocorsa.

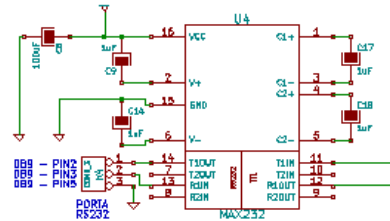
I nand U3A,U3B e U3C con i due transistori Q1 e Q2 costituiscono la logica che abilita o meno il motore a ruotare.



L'interfacciamento con il PC che controllerà il rotore può essere fatto sia tramite la porta USB della scheda Arduino che tramite la classica seriale RS232 (U4).

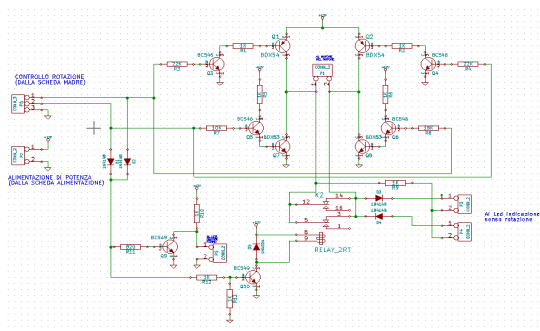
Nel primo caso però si avrà l'inconveniente che alimentando il computer questo accenderà una parte della circuitazione del rotore anche quando questo è spento, ovvero il display sarà attivo ma non retroilluminato ne sarà possibile girere le antenne.

Personalmente utilizzo un adattatore RS232/USB in modo anche da "disaccoppiare" anche se non galvanicamente i due dispositivi.



LA SCHEDA CONTROLLO MOTORE

Il controllo del motore è effettuato tramite un ponte ad H (Q1,Q2,Q7 e Q8) ed un relè che effettua la frenatura in controcorrente. Questa parte del circuito è ospitata da una scheda separata in modo da permettere futuri sviluppi e variazioni senza dover riprogettare l'intero controll box.

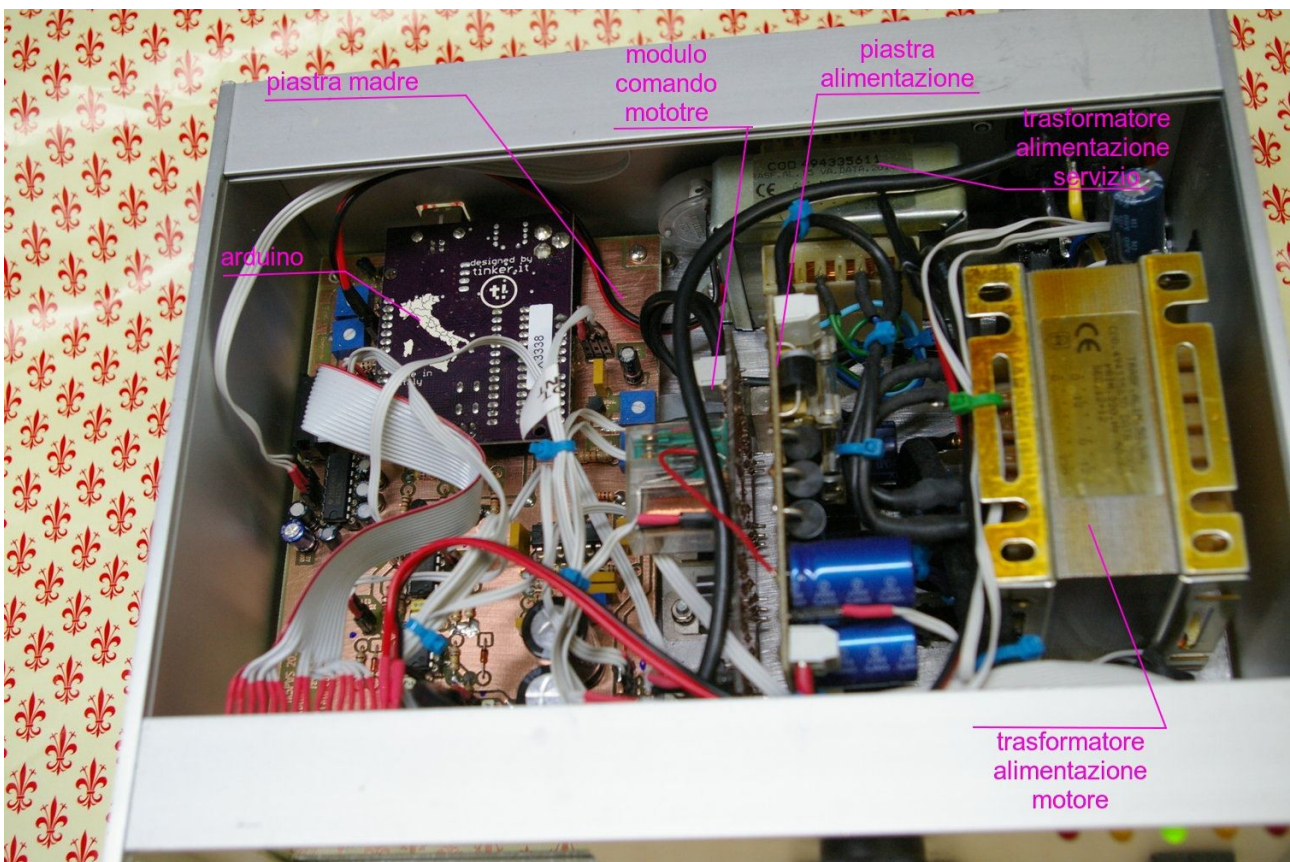
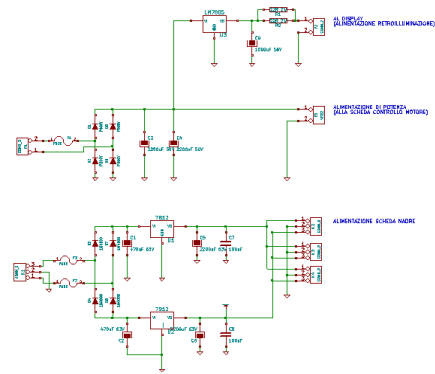


LA SCHEDA DI ALIMENTAZIONE

L'alimentatore è costituito da 2 sezioni, la scelta è stata imposta dalle forti escursioni di corrente e quindi di tensione che si hanno alimentando il rotore.

Queste farebbero variare la tensione di alimentazione del potenziometro che legge la posizione dell'antenna rendendone aleatoria la lettura. Così la parte "di potenza alimenta" il motore tramite un normale ponte a doppia semionda e con un piccolo stabilizzatore la retroilluminazione del display.

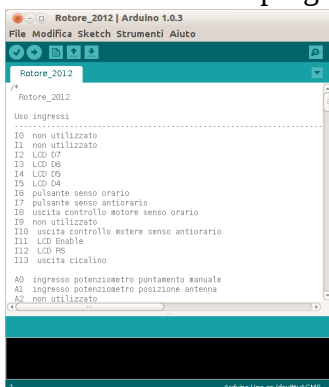
L'altra, di servizio, con tensione una tensione duale alimenta la scheda madre.



IL SOFTWARE E LE FUNZIONI DEL ROTORE

L'Arduino si programma con estrema facilità tramite la porta USB di cui dispone, questa può essere utilizzata anche per alimentarlo insieme ad alcune periferiche a basso consumo, si deve tener presente che prendendo l'alimentazione dall'USB del PC andiamo a caricare quest'ultima esponendola anche a possibili danni. Il software come tutto il progetto hardware e la documentazione è liberamente scaricabile da www.arduino.cc, esiste sia in versione per pc Linux che Windows.

E' necessario il sorgente (sketch), aprirlo con il programma di sviluppo (IDE), configurarlo per il modello di arduino corretto e la porta



a cui questo è collegato, fare l'upload sulla scheda. Fatto.

Le funzioni che ho implementato sono:

- Classici tasti di direzione oraria ed antioraria
- Potenziometro che permette di impostare un angolo compreso tra 0 e 360° verso cui il rotore provvederà a indirizzare automaticamente l'antenna.
- Controllo della direzione tramite porta seriale.
- Scala graduata sotto cui scorre un cursore che da la rappresentazione grafica della direzione corrente dell'antenna.
-



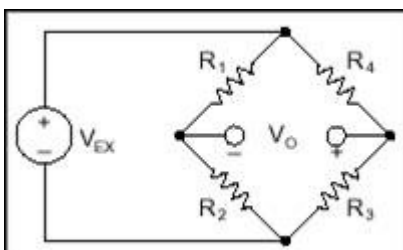
LA TARATURA

In primis è necessario impostare la soglia della protezione dell'ingresso analogico dell'Arduino, senza quest'ultimo inserito, si inserisce una tensione 6-7V tra il cursore di R31 e la resistenza R13 e si regola R28 per avere una tensione massima tra il pin A1 e massa di 5,0V.

Si passa adesso alla taratura del controll box vero e proprio in modo che legga correttamente la posizione dell'antenna.

Per poterlo fare agevolmente ed in sicurezza senza dover armeggiare sul tetto mi sono costruito un circuito che simula le varie posizioni dell'antenna, questo permette di fare un solo accesso alle antenne e successivamente di fare la messa a punto comodamente in laboratorio.

Il controll box applica agli estremi del potenziometro del rotore una tensione e per saperne la posizione va a misurare la tensione presente sul cursore.



Ho sfruttato il principio di funzionamento del ponte di wheatstone, ovvero, se abbiamo 4 resistenze collegate come in figura, quando la V_0 è nulla la tensione su R1 e quella su R4 saranno identiche come pure per l'altra coppia R2 ed R3.

Quindi se R1 ed R2 sono i due resistori equivalenti del potenziometro calettato sull'asse del rotore, sostituendoli con la coppia R4 ed R3 (che equivalgono ad un trimmer) all'ingresso del controll box otterremo la stessa indicazione.

Ho predisposto con una semplice basetta "millefori" un circuito di taratura costituito da 7 trimmer che avrei selezionato singolarmente con un jumper.

A questo punto sono salito sul tetto ed ho puntato manualmente le antenne verso nord, ho fissato un goniometro intorno all'asse verticale del rotore e l'ho regolato per indicare 0° ottenendo un riferimento certo di quanto poi sarebbe ruotato il tutto.

Ho regolato il potenziometro della posizione in modo da avere la stessa ddp tra il cursore ed i due estremi e l'ho fissato definitivamente.

trimmer Si collega i pin 1 e 3 del K1 della scheda di taratura in parallelo agli estremi del potenziometro calettato sull'asse, si posiziona un multimetro impostato per Volt tra il pin 2 di K1 ed il cursore del potenziometro, infine si seleziona con il jumper RV4 (Nord, 0°).

Semplicemente adesso si regola il trimmer per avere una indicazione di 0V sul tester.

Si posiziona le antenne nelle varie altre direzioni (est, ovest, sud dx e sx, fine corsa sx e dx) spostando di volta in volta il ponticello per selezionare il relativo trimmer e si ripete la regolazione azzerando ogni volta la lettura sul multimetro.

A questo punto si può scendere dal tetto e procedere in comodità in laboratorio.

Per sicurezza si toglie al momento l'Arduino, si regola R31 portando il suo contatto mobile a contatto con il reoforo che va al pin8 di U2C, si collega il circuito di taratura al controll box al posto del collegamento verso il potenziometro calettato al rotore e si inserisce il jumper sul fine corsa antiorario.

Si regola R29 per avere pin 8 di U2C una tensione pari a 0V.

Si passa a regolare R1, se il led di raggiunto fine corsa antiorario è acceso si regola il trimmer sino a quando questo non si spegne, a questo punto si gira la vite di regolazione in senso opposto sino a riottenere l'accensione. Se invece è spento si regola il trimmer fino a farlo accendere.

Adesso si sposta il jumper sul fine corsa antiorario e si regola R31 per avere sul suo contatto mobile 5V, fare attenzione al fatto che è presente il circuito di protezione che abbiamo tarato in precedenza, quindi se la tensione in uscita da R31 fosse superiore alla soglia di intervento, il regolatore U7 bloccherebbe la tensione sul pin A1 di Arduino a 5V facendo nascere una d.d.p. su R13, questo porta ad una errata indicazione del rotore.

Si passa a regolare R30, se il led di raggiunto fine corsa orario è acceso si regola il trimmer sino a quando questo non si spegne, a questo punto si gira la vite di regolazione in senso opposto sino ad riottenere l'accensione. Se invece è spento si regola il trimmer fino a farlo accendere.

Si inserisce adesso l'Arduino (a circuito spento), si accende il controll box e si regola il contrasto in modo da leggere comodamente il display.

Se tutto è andato bene selezionando con il circuito di taratura le varie posizioni del rotore si avranno le relative letture sul display.

Possono esserci delle piccole differenze di lettura dovute alle non linearità dei componenti ed in particolare del circuito di protezione dell'ingresso, infatti U7 inizia a condurre un poco prima dei 5V canonici quindi abbasserà la tensione all'ingresso A1.

In questo caso selezionare sul circuito di taratura "nord 0°" e con il trimmer R29 regolare sino al leggere 0° sul display.

Ripetere quindi i passi iniziali per verificare che agli estremi della rotazione l'escursione rimanga tra 0 e 5,0V.

Sperando di aver fornito qualche spunto a qualcuno, tutto questo è a disposizione della comunità radioamatoriale sotto licenza creative commons.

I vari file contenenti lo schema elettrico e i circuiti stampati realizzati con il programma Kicad ed i sorgenti da caricare sull'Arduino sono disponibili sul sito www.arilucca.it o possono essere richiesti a me direttamente.