tipu91

INIZIAMO CON... ARDUINO: IL SENSORE DI PARCHEGGIO

27 March 2013

PREMESSA

Natale 2012. Come per molti universitari, le vacanze di Natale significano “la quiete prima della tempesta” (la tempesta sarebbe la sessione di esami invernale). Mio padre, da sempre patito di hi-tech ed affini,mi spiazza completamente: come prima cosa, essendo ormai “grandicello” non scartavo più regali da un pezzo, e in secondo luogo, non mi sarei mai aspettato quel regalo: UNO STARTER KIT CON ARDUINO!! Si proprio quella “schedina” che vedevo tutti i mesi su WIRED (spero non si offenda nessuno per la pubblicità) e che mi ha sempre attratto per le potenzialità che offre. Premetto che non sono un programmatore esperto (ho sostenuto un solo esame di programmazione all’università),tanto meno di elettronica (dato che il primo corso di elettronica l’ho sto affrontando ora), ma mi ritengo una persona curiosa e vogliosa di imparare, soprattutto in questo ambito, in modo da ampliare il mio bagaglio tecnico-culturale (extra-accademico)di aspirante ingegnere.

“Purtroppo” la sessione di esami non mi ha permesso di sbizzarrirmi con il mio nuovo “amico”, ma adesso, che gli esami sono alle spalle (anche se fra poco ci saranno i prossimi uff) posso iniziare a testare le mie e le “sue” (di ARDUINO) capacità. In questo articolo (e spero anche nei prossimi) ho l’intenzione di condividere con voi il mio primo progetto, in modo da aiutare chi sta in iniziando (come me) e farmi aiutare da chi è più avanti nella scoperta di questo bellissimo “giochino”!!

INTRODUZIONE

in questo articolo spiegherò uno miei primi progetti, un“sensore di parcheggio”. Questi progetti sono facili da realizzare e soprattutto sono delle rappresentazioni di applicazioni reali (ma semplificate)che sfruttano alcune delle potenzialità di Arduino.

SENSORE DI PARCHEGGIO

Siamo già da qualche anno entrati nel 21° secolo, nonché nel 3° millennio, tempi in cui la tecnologia fa parte della quotidianità e la si trova in quasi ogni oggetto

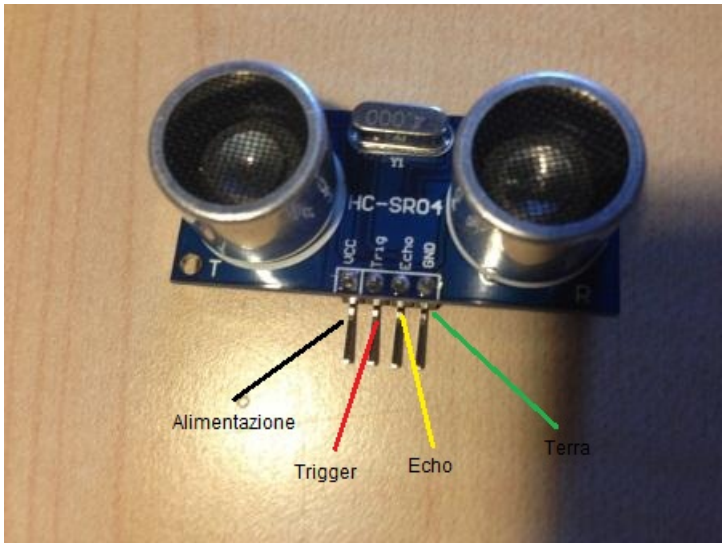
che ci circonda. Da qualche anno, nel campo automobilistico (civile) è entrata in gioco una tecnologia (ormai quasi “vecchia”e banale) che agevola l’automobilista: il SENSORE DI PARCHEGGIO :“Strumento ad onde ultrasoniche in grado di calcolare e visualizzare la distanza intercorrente fra il veicolo sul quale esso è installato e l'ostacolo a lui frontalmente più vicino.” (FONTE:brevetto per invenzione industriale n. 1196650 presentato a Roma al Ministero dell'Industria del Commercio e dell'Artigianato il 13 dicembre 1984 dall'Ing.Massimo Ciccarello e dall'Arch. Ruggero Lenci, e rilasciato il 16 novembre1988. Certificazione allegata). In parole povere è un sistema che,grazie agli ultrasuoni, comunica tramite vari metodi (suoni, luci, ecc) la distanza del proprio veicolo da un ostacolo circostante, utile per esempio in fase di parcheggio. Con questo progetto quindi, mi sono proposto di “ricreare” uno strumento simile, per poter sperimentare alcune funzioni di Arduino e dei sensori compresi nel kit.

Materiali:

- resitore da 100Ω
- speaker piezoelettrico da 8Ω
- sensore ad ultrasuoni HC-SRO 4
- Arduino (esempio Arduino UNO)
- Breadboard
- cavetti per collegamenti

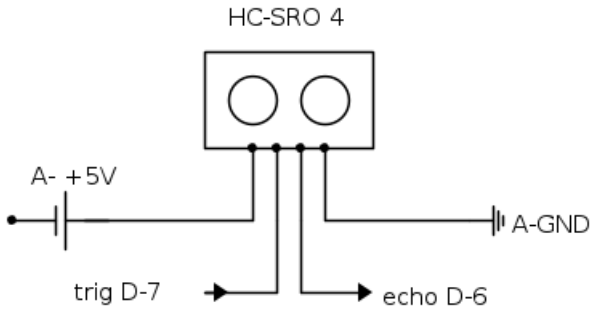
Assemblaggio

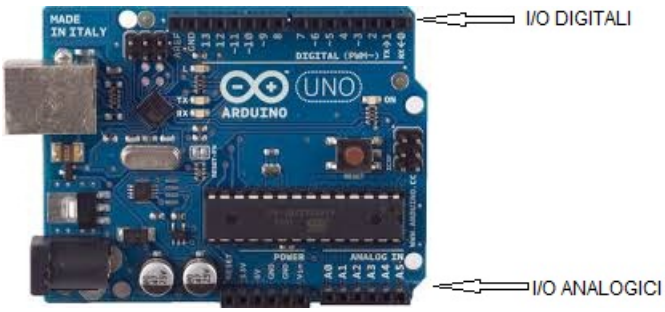
Il progetto, a livello circuitale, è suddiviso in due tronchi: la parte di rilevazione o di ingresso (il sensore a ultrasuoni) e la parte di trasmissione dei “risultati” (in questo caso uno speaker, ma chi vuole può sostituirlo da led o da uno schermo LCD), naturalmente connessi da un “mediatore”,il nostro Arduino. Come prima cosa prendiamo la breadbord e ci colleghiamo il sensore ad ultrasuoni:



IMG_0139.JPG

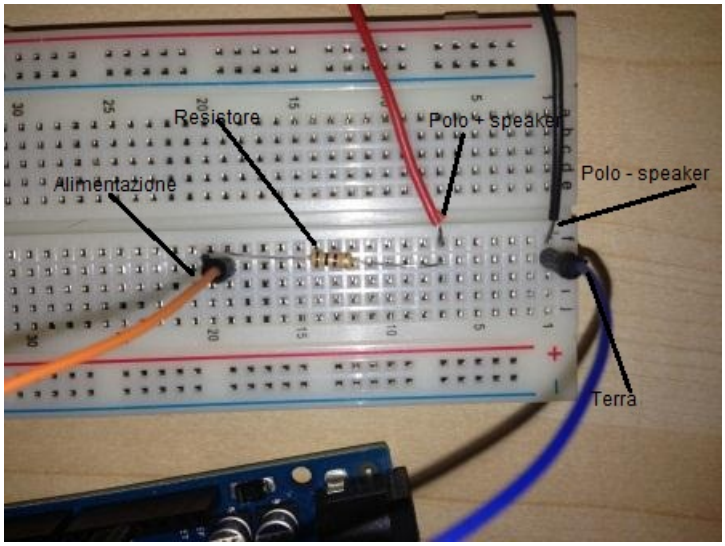
per quanto riguarda l'alimentazione del sensore, il pin VCC lo colleghiamo al pin di alimentazione ANALOGICO della scheda (quello a 5V) mentre il pin GND (la "terra") lo si collega a uno dei due GND della parte analogica di Arduino. Il pin TRIG (il trigger), quello che ci permetterà di inviare l'ultrasuono, lo colleghiamo a uno dei connettori I/O DIGITALI (io per esempio l'ho connesso al pin 7). Infine colleghiamo il pin ECHO (il sensore risponderà sul pin Echo con un impulso alto della durata corrispondente a quella di viaggio delle onde sonore) ad un'altra I/O DIGITALE (per esempio il pin 6).



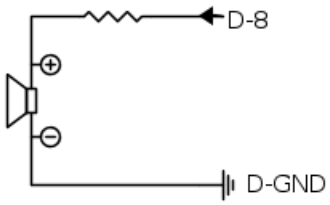


images.jpg

Adesso passiamo all'assemblaggio della parte di circuito comprendente lo speaker. Si collega il polo positivo dello speaker piezoelettrico, tramite un resistore da 100Ω ad un pin I/O DIGITALE: il pin 8. Mentre il polo negativo dello speaker viene collegato direttamente alla "terra" che si trova sempre sul lato dei connettori I/O DIGITALI.

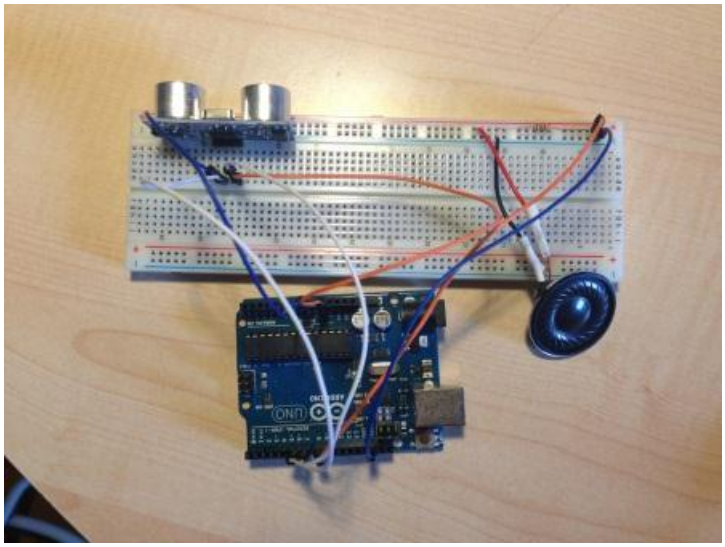


IMG_0151.JPG



A questo punto, la parte di assemblaggio circuitale è finita, quindi non ci resta che

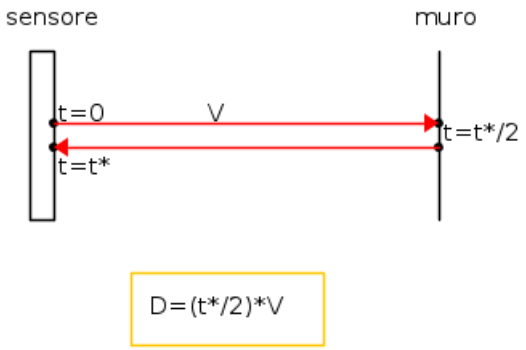
passare alla programmazione della scheda. Se tutto è andato per il verso giusto dovremmo avere una cosa del genere:



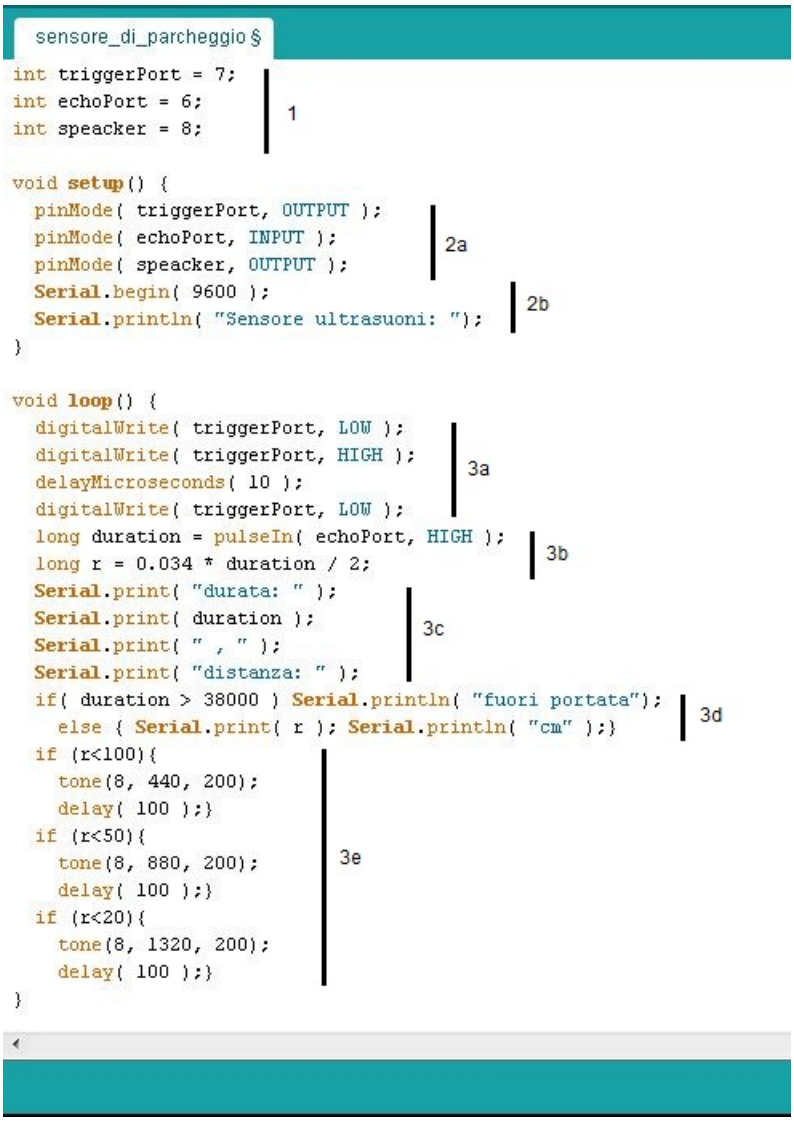
IMG_0133.JPG

Programmazione

Prima di tutto vorrei spiegare a "parole" (quindi senza codice) quello che dobbiamo andare a fare: dobbiamo inviare un'impulso tramite il TRIGGER ogni periodo (il periodo verrà stabilito successivamente) e si deve andare a leggere la durata del "tragitto" dell'ultrasuono sulla porta ECHO. Qui entra in gioco un pochino di fisica elementare: dato il tempo di percorrenza della distanza (che nel nostro caso sarà doppia perchè l'ultrasuono deve fare andata-ritorno) e la velocità (quella del suono) si deve ricavare la distanza percorsa dall'ultrasuono. Infine dobbiamo trasmettere la misura tramite lo speaker (o quello che si vuole).



Passiamo quindi alla scrittura, o meglio, alla spiegazione del codice!



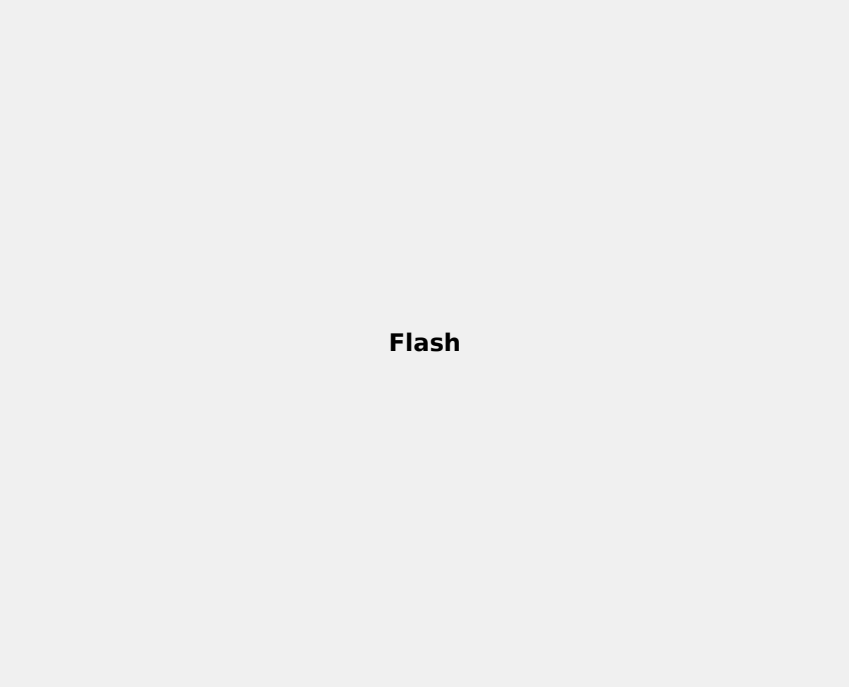
Cattura.JPG

Nel blocco 1 si inizializzano i pin digitali: quello del TRIG (7) lo chiamiamo *triggerPort*; quello del ECHO (6) lo chiamo *echoPort*; infine il pin relativo allo speaker (8) lo chiamo *speaker*.

Il blocco 2 è quello della funzione **setup**, la prima ad essere chiamata quando parte uno sketch. Viene utilizzata per inizializzare variabili, per impostare lo stato dei pin, per far partire le librerie da usare, per l'impostazione delle comunicazioni seriali. La funzione di setup() sarà la prima ad essere eseguita dopo ogni accensione o reset di Arduino. infatti nel blocco 2a si imposta lo stato dei pin: il pin TRIG e quello relativo allo SPEAKER vengono impostati come OUTPUT, mentre quello ECHO come INPUT. nel blocco 2b invece si imposta le comunicazioni seriali (si inizializza il monito seriale, sul quale si possono visualizzare i dati rilevati se si collega Arduino al computer).

Il blocco 3 è il blocco della funzione **loop** che fa proprio quanto suggerisce il proprio nome, eseguendo ciclicamente il programma definito al suo interno. Nella parte 3a si deve emettere l'impulso; come prima cosa si "spegne" il pin *triggerPort*, successivamente si "accende" e infine si rispegne dopo aver aspettato 10 millisecondi (i tempi sono misurati tutti in millisecondi). Il blocco 3b è quello adibito alla ricezione dell'impulso e al calcolo della distanza. Nella prima istruzione infatti definisce una variabile *duration* tramite il pino ECHO, mentre nella seconda istruzione definisce la variabile della distanza *r* colcolandola come detto precedentemente (ricordiamoci che la velocità è espressa in m/ms). Nel blocco 3c si riportano i dati rilevati sul monitor seriale: se intendiamo "lavorare" senza pc attaccato, queste istruzioni sono superflue. Anche il blocco 3d non è necessario per il funzionamento; infatti questa istruzione serve a ricordarci che se il tempo di percorrenza (quindi la distanza) è troppo elevata, la risposta sarà falsata: per questo scriverà sul monitor seriale "fuori portata". Infine il blocco 3e è quello relativo allo speaker: se la distanza rilevata (sono misurate tutte in centimetri) è inferiore a 100cm, lo speaker emetterà un suono che dura 200ms di frequenza 440Hz (il famoso LA 440) ogni 100ms; se la distanza è inferiore a 50cm lo speaker emetterà contemporaneamente due suoni (440Hz e 880Hz); mentre se la distanza scende sotto i 20cm lo speaker emetterà contemporaneamente tre suoni (440Hz, 880Hz e 1320Hz).

CONCLUSIONI



Naturalmente non mi ritengo assolutamente un esperto in materia, ma mi faceva piacere condividere con gli altri neofiti come me questa guida. Spero in futuro (non troppo lontano) di avere abbastanza tempo per pubblicare anche altri progetti, per

ora vi ringrazio per l'attenzione e auguro, a chi voglia cimentarsi in questa prova, buona fortuna!

PS per qualsiasi chiarimento, oltre che commentare, potete contattarmi anche tramite MP!! PPS chi volesse il codice (dato che non vi è la possibilità di mettere lo script) può chiedermelo direttamente!!

Estratto da "http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Tipu91:iniziamo-con-arduino"