Il Bastone Sensibile per Ipovedenti [Light]

Obiettivo: Il **Bastone Sensibile per Ipovedenti** rappresenta un'attività di stampo inclusivo utilizzabile in un contesto scolastico al fine di compensare la cecità e permettere agli alunni non vedenti di superare ostacoli ed impedimenti ambientali. Tale attività permette di sviluppare attraverso un **processo meta-inclusivo** l'empatia necessaria per comprendere le difficoltà affrontate da un ragazzo ipovedente all'interno di un contesto scolastico.
Tale strumento, rappresenta l'evoluzione del più comune e tradizionale bastone assistivo per compensare la cecità e permettere agli alunni non vedenti di superare ostacoli ed impedimenti ambientali. Un sensore di prossimità, montato sul telaio, conferisce la sensibilità al bastone. A seconda della distanza calcolata attraverso il sensore, vengono generati degli impulsi sonori con durata variabile. A differenza di quanto avviene con l'uso di un bastone tradizionale, con il quale gli ostacoli vengono individuati attraverso il tocco, in questo caso basterà concentrarsi sulla durata sonora per capire la posizione di eventuali impedimenti al percorso, proprio come avviene con i sensori di parcheggio delle automobili. Durante le attività, gli alunni sono chiamati in prima persona ad affrontare e cercare di superare gli ostacoli dovuti alla cecità. Nello stesso tempo si rendono conto delle diverse-abilità che un loro pari, non vedente o ipovedente, deve necessariamente sviluppare per compensare la mancanza della piena percezione visiva.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Sensore Ultrasuoni (e.g., HC-SR04)
- 1 Buzzer attivo

Pre-requisiti:

Il Sensore a Ultrasuoni

Buzzer Attivo

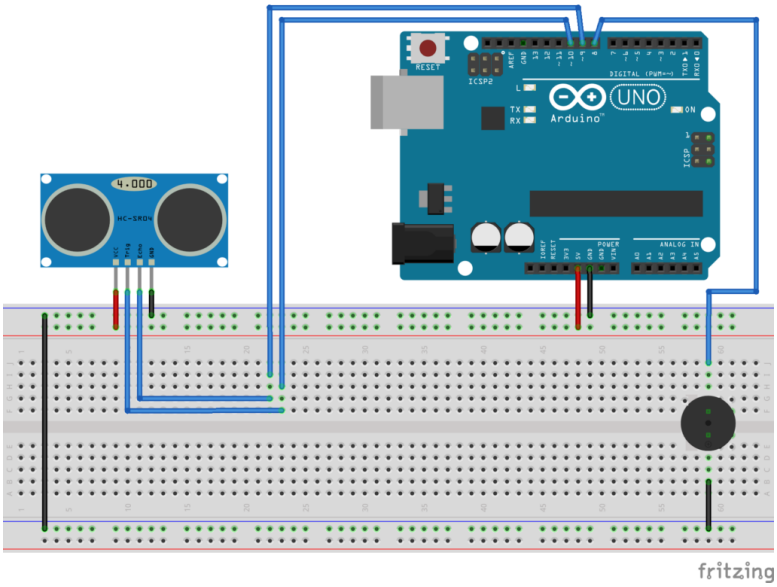
Teoria: Il **bastone sensibile per ipovedenti** rappresenta dal punto di vista tecnologico la naturale evoluzione del tradizionale sensore di parcheggio, ovvero un dispositivo ampiamente utilizzato in ambito automobilistico per favorire l'operazione del una delle moderne tecnologie che permettono al guidatore di un autoveicolo di essere a conoscenza della distanza tra la propria automobile ed un eventuale altro veicolo.
Nel caso specifico, il bastone sensibile si basa su un sensore di prossimità ad ultrasuoni (INPUT) ed un buzzer (OUTPUT) utilizzato come segnalatore acustico.
Nel dettaglio, maggiore è la distanza maggiore sarà il ritardo tra una segnalazione acustica e la successiva, analogamente, minore è la distanza minore sarà il ritardo tra una segnalazione acustica e la successiva. In particolare, la relazione ingresso uscita che lega la distanza al ritardo impiegato nella segnalazione acustica è funzione dei seguenti quattro parametri:

- DistanzaMassima: la distanza massima dopo la quale non viene più segnalato un’ostacolo.
- DistanzaMinima: la distanza minima per la quale il buzzer emette un tono continuo
- RitardoMassimo: Il ritardo tra una segnalazione acustica e la successiva nel caso di massima distanza.
- RitardoMinimo: Il ritardo tra una segnalazione acustica e la successiva nel caso di minima distanza.

Questi valori vengono utilizzati al fine di determinare l’equazione fondamentale per il calcolo del ritardo:

ritardo = distanza*((RitardoMax-RitardoMin)/(DistanzaMax-DistanzaMin))

<u>Collegamento Circuitale</u>:



Collegamento Circuitale

<u>Codice:</u>

Personalizzazioni: E’ possibile introdurre un secondo sensore ad ultrasuoni per evidenziare la presenza di un ostacolo a destra oppure a sinistra dell’utilizzatore.

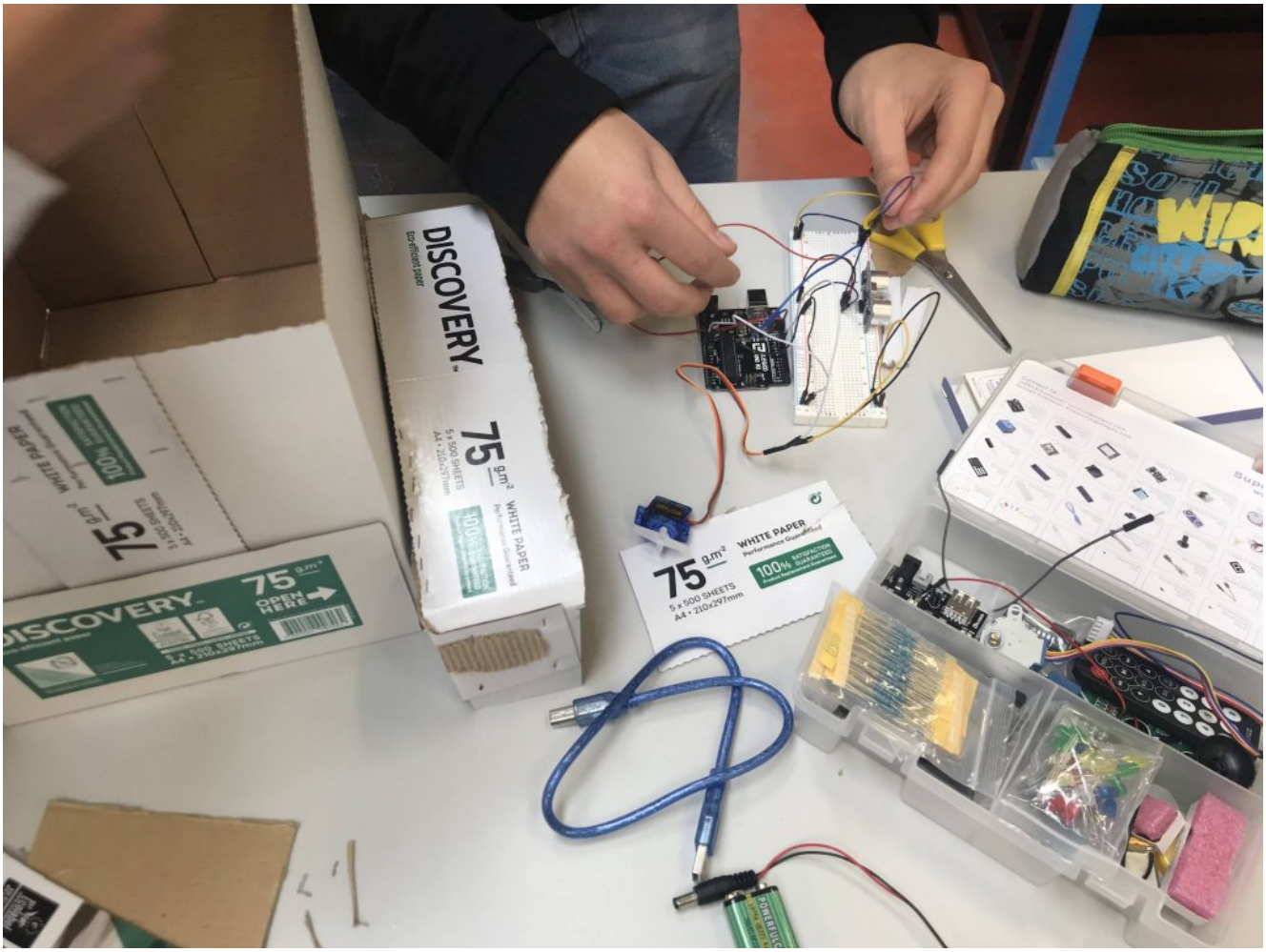
Il Cestino Smart scomposto

L’ultimo decennio è stato contrassegnato da un aumento delle problematiche legate al rispetto dell’ambiente ed al risparmio energetico. Il riscaldamento del pianeta, la guerra in Ucraina e la conseguente crisi energetica, impongono tutti a ripensare alle abitudini quotidiane.

La scuola ha il dovere di intervenire sulle nuove generazioni, fornendo loro le competenze e la sensibilità necessaria per rispettare il nostro pianeta.

Noi di Arduino Facile siamo convinti che i piccoli gesti, quelli concreti, alla portata di tutti, siano quelli che maggiormente ci avvicinano alla realtà, che, a volte, ci sembra lontana da noi.

All’ITIS “E.Mattei” di Urbino (https://www.itisurbino.edu.it/), gli studenti della 3BMC, dell’indirizzo di Meccanica e Meccatronica, si sono cimentati nella realizzazione di un cestino “intelligente”, che, alle pratiche igieniche, unisca ad esempio la raccolta differenziata dei rifiuti.



I gruppi al lavoro

Il punto di partenza è stato il nostro articolo https://www.arduinofacile.it/2021/05/13/il-cestino-smart/

Successivamente la classe è stata divisa in gruppi di 4/5 studenti che hanno lavorato in modalità jigsaw: ogni componente si è specializzato in una particolare competenza, che poi ha condiviso con i compagni.

Il resto è stato un mix di fantasia ed immaginazione…

Ogni gruppo, infatti, ha interpretato a suo modo la consegna, ed ha elaborato una propria versione del *cestino smart*. Ne è nata una competizione agguerrita in cui ha avuto la meglio il

gruppo denominato “Marrakesh” che ha realizzato un cestino con apertura a due ante, e una spia ad indicare il riempimento.

Le istruzioni di montaggio sono disponibili nella seguente presentazione preparata dai componenti del gruppo

cestino smartScarica

Al secondo posto si è piazzata un’altra versione, ancora più amica dell’ambiente: un cestino formato maxi, con due ante, per differenziare i rifiuti

Infine, sul podio un cestino più classico con la segnalazione di apertura.

Utilizzare e Creare una Libreria per il Sensore ad Ultrasuoni

Obiettivo: Utilizzare e creare una libreria (file header e cpp) per un Sensore a Ultrasuoni (HC-SR04) utilizzato per misurare la distanza.

Puoi scaricare i file di libreria cliccando nel seguente link: http://www.arduinofacile.it/wp-content/uploads/2020/10/UltrasonicSensor.zip
I file scaricati devono essere inseriti all'interno della cartella di progetto insieme al file .ino

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Sensore Ultrasuoni (e.g., HC-SR04)

Pre-requisiti:

Il Sensore a Ultrasuoni

Teoria: la realizzazione di funzioni di libreria permette di facilitare l'operazione di **riutilizzo del codice** rendendo più veloce e più rapido lo sviluppo. Nel caso specifico la funzione di libreria implementata sarà costituta da un file header (.h) e da un file sorgente (.cpp).

Un file header è un file di testo che contiene i prototipi dei metodi (funzioni) definite nel relativo file sorgente. Nel caso in questione il file header contiene anche la dichiarazione della classe "UltrasonicSensor" utilizzata per modellare il sensore ad ultrasuoni.
Tale classe sarà caratterizzata da 2 attributi:

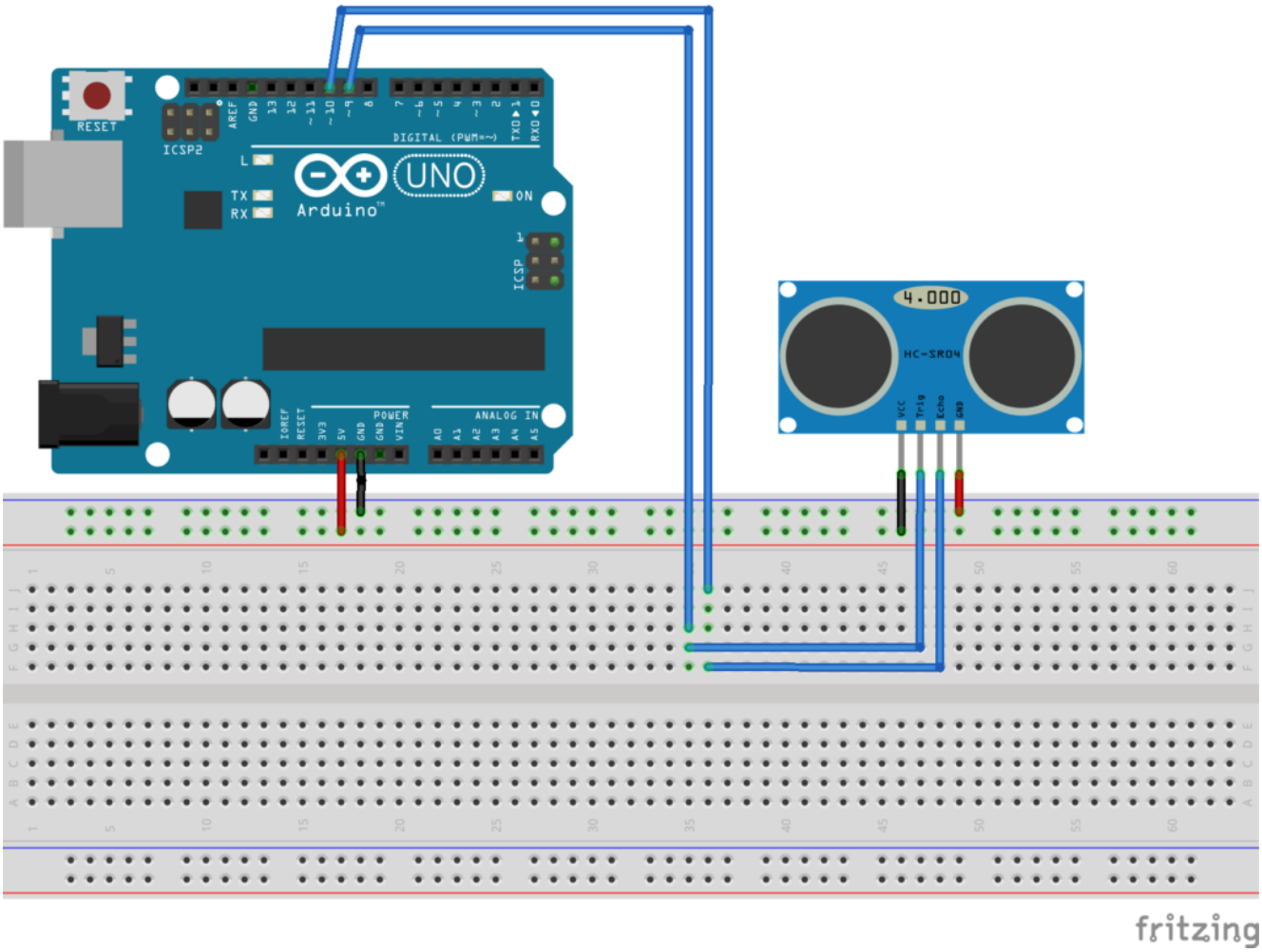
- **int pinEcho**: il pin di echo
- **int pinTrigger**: il pin di trigger

e da 4 metodi:

- **void SetPinEcho (int pinEcho)**: metodo utilizzato per settare il pin di echo
- **void SetPinTrigger(int pinTrigger):** metodo utilizzato per settare il pin di trigger:
- **long GetDistance():** metodo utilizzato per effettuare la misura di distanza (restituisce un valore di tipo long)
- **long GetAverageDistance(int numIteration)**: metodo utilizzato per effettuare la misura di distanza media su un numero dato di misure (restituisce un valore di tipo long)

Nel file sorgente viene invece riportata l'implementazione dei prototipi delle funzioni dichiarate nel file header.

<u>Collegamento Circuitale</u>:



Collegamento Circuitale

<u>Codice</u>:

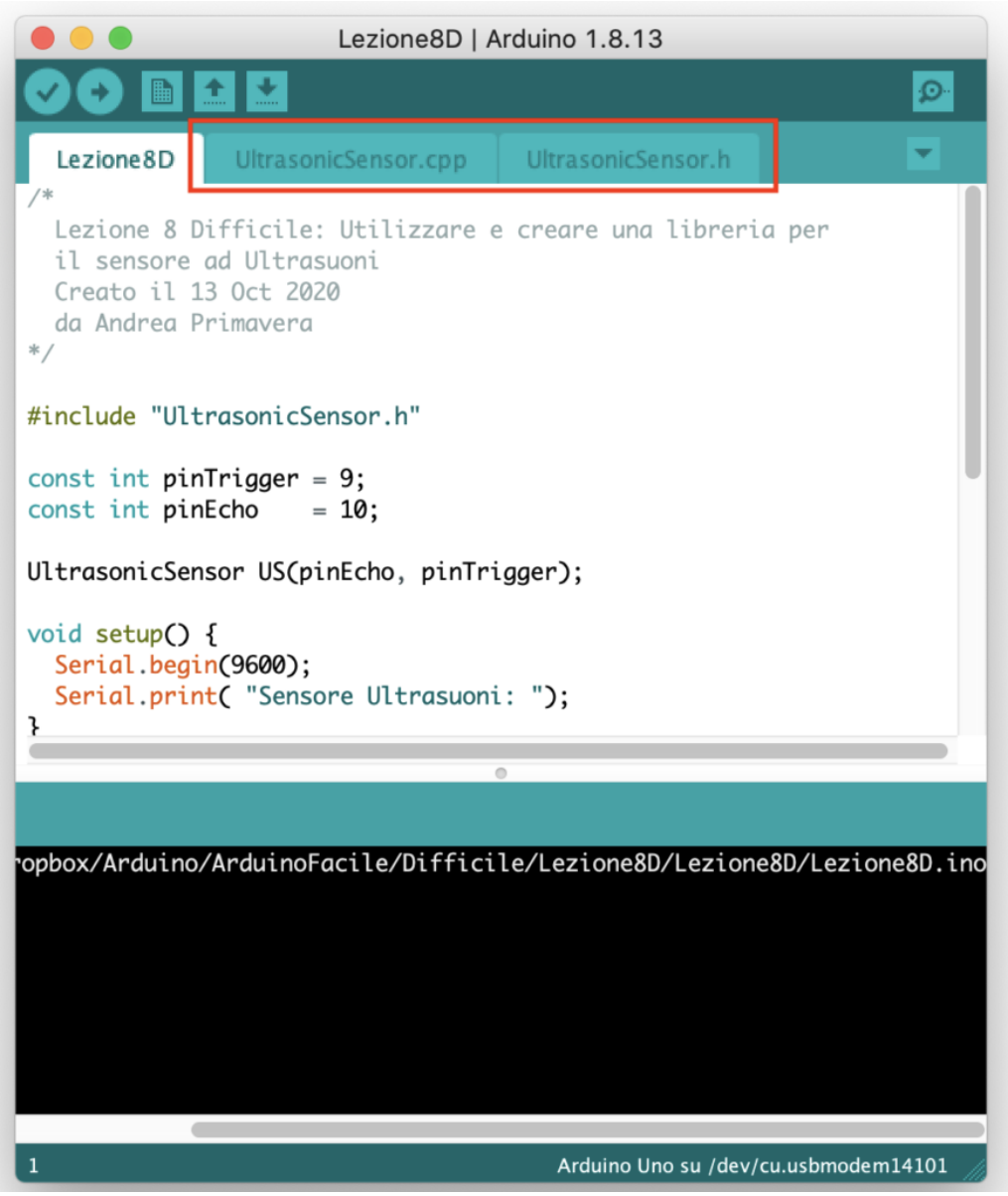
Vengono in seguito riportate le tre porzioni di codice utilizzate per creare la funzione di libreria relativa al sensore ad ultrasuoni HC-SR04.

- File Header: contiene la definizione della classe con i propri attributi (i.e., pinEcho e pinTrigger) ed i prototipi dei relativi metodi.

- File Sorgente: contiene le implementazioni dei metodi riportati nel file header.

- File Arduino: Utilizzato per fornire un esempio di come utilizzare la libreria per la gestione del sensore ad ultrasuoni.

Se tutti i file sono correttamente posizionati sullo stesso livello all’interno della cartella di progetto, due nuove tab compariranno nell’ambiente di sviluppo utilizzato per programmare Arduino. Attraverso queste tab sarà possibile visionare e modificare il file sorgente (.cpp) ed il file header (.h)



Il Sensore di Parcheggio

Obiettivo: Realizzare un sensore di parcheggio

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Sensore Ultrasuoni (e.g., HC-SR04)
- 1 Buzzer

Pre-requisiti:

Il Sensore a Ultrasuoni

Buzzer Attivo

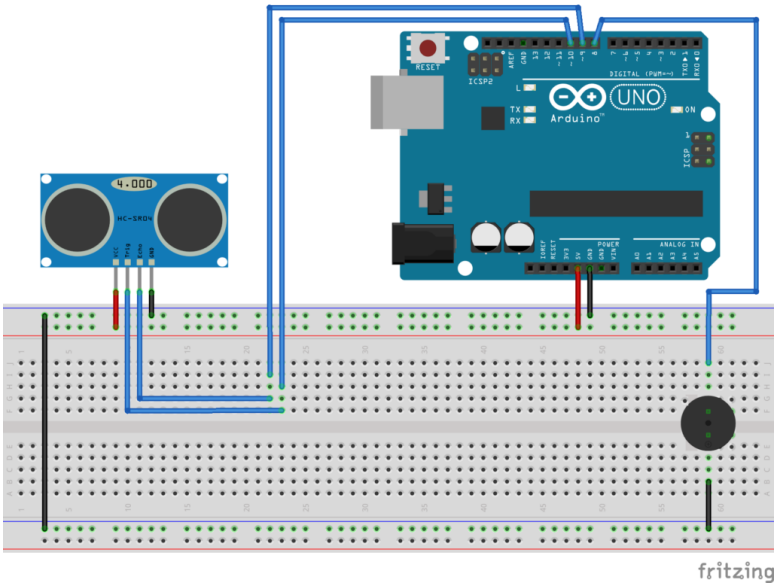
Teoria: Il **sensore di parcheggio** rappresenta una delle moderne tecnologie che permettono al guidatore di un autoveicolo di essere a conoscenza della distanza tra la propria automobile ed un eventuale altro veicolo. Questo permette di gestire in modo facile e rapido l'operazione di parcheggio evitando danneggiamenti o urti agli autoveicoli in questione.
Nel caso specifico, il sensore di parcheggio si basa su un sensore di prossimità ad ultrasuoni (INPUT) ed un buzzer (OUTPUT) utilizzato come segnalatore acustico.
Nel dettaglio, maggiore è la distanza maggiore sarà il ritardo tra una segnalazione acustica e la successiva, analogamente, minore è la distanza minore sarà il ritardo tra una segnalazione acustica e la successiva. In particolare, la relazione ingresso uscita che lega la distanza al ritardo impiegato nella segnalazione acustica è funzione dei seguenti quattro parametri:

- DistanzaMassima: la distanza massima dopo la quale non viene più segnalato un’ostacolo.
- DistanzaMinima: la distanza minima per la quale il buzzer emette un tono continuo
- RitardoMassimo: Il ritardo tra una segnalazione acustica e la successiva nel caso di massima distanza.
- RitardoMinimo: Il ritardo tra una segnalazione acustica e la successiva nel caso di minima distanza.

Questi valori vengono utilizzati al fine di determinare l’equazione fondamentale per il calcolo del ritardo:

ritardo = distanza*((RitardoMax-RitardoMin)/(DistanzaMax-DistanzaMin))

<u>Collegamento Circuitale</u>:



Collegamento Circuitale

<u>Codice:</u>

Personalizzazioni: E’ possibile inserire un led al fine di avere un segnalatore visivo che indichi la distanza dall’ostacolo.

Collegare più Sensori Ultrasuoni ad Arduino

Obiettivo: Utilizzare simultaneamente due sensori a ultrasuoni (HC-SR04). Nel dettaglio, al fine di dimostrare il corretto funzionamento del circuito, ad ogni sensore è associato un led. Quando la distanza misurata dal sensore è inferiore ad una data soglia, il led associato si accende, mentre quando la distanza è superiore il led si spegne.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 2 Sensori Ultrasuoni (e.g., HC-SR04)
- 2 Led (utilizzati per dimostrare il funzionamento del circuito)
- 2 Resistenze (100 Ohm) (indispensabili per il corretto funzionamento dei led)

Pre-Requisiti:

Il Sensore a Ultrasuoni

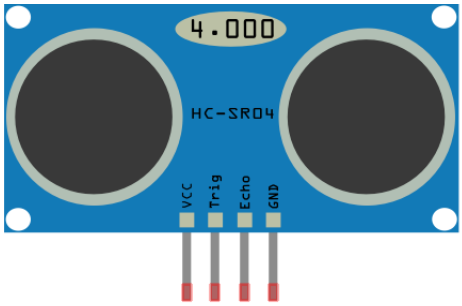
Teoria: Il sensore di prossimità è un dispositivo che permette di rilevare la presenza di oggetti nelle immediate vicinanze, senza che vi sia un effettivo contatto.
Nel caso specifico, il sensore di prossimità ad ultrasuoni sfrutta il principio del Sonar. Degli impulsi sonori (ultrasonici) vengono emessi dal dispositivo il quale attraverso l'eventuale eco di ritorno permette di rilevare la presenza di un oggetto all'interno della portata nominale.
Esempi pratici di sensori ad ultrasuoni sono i sensori di retromarcia e di parcheggio utilizzati nelle moderne automobili.
Nel dettaglio, le principali caratteristiche tecniche del sensore ad ultrasuoni HC-SR04 sono:

- Alimentazione: +5V DC
- Angolo di misura: < 30°
- Distanza di rilevamento: da 2cm a 400cm
- Risoluzione: 1cm
- Frequenza: 40kHz

Il sensore, dispone di 4 pin: Vcc (+5V), Trigger, Echo, GND.

- VCC: +5V DC
- Trigger: Genera l'impulso ultrasonico
- Echo: Rileva il segnale ultrasonico di ritorno

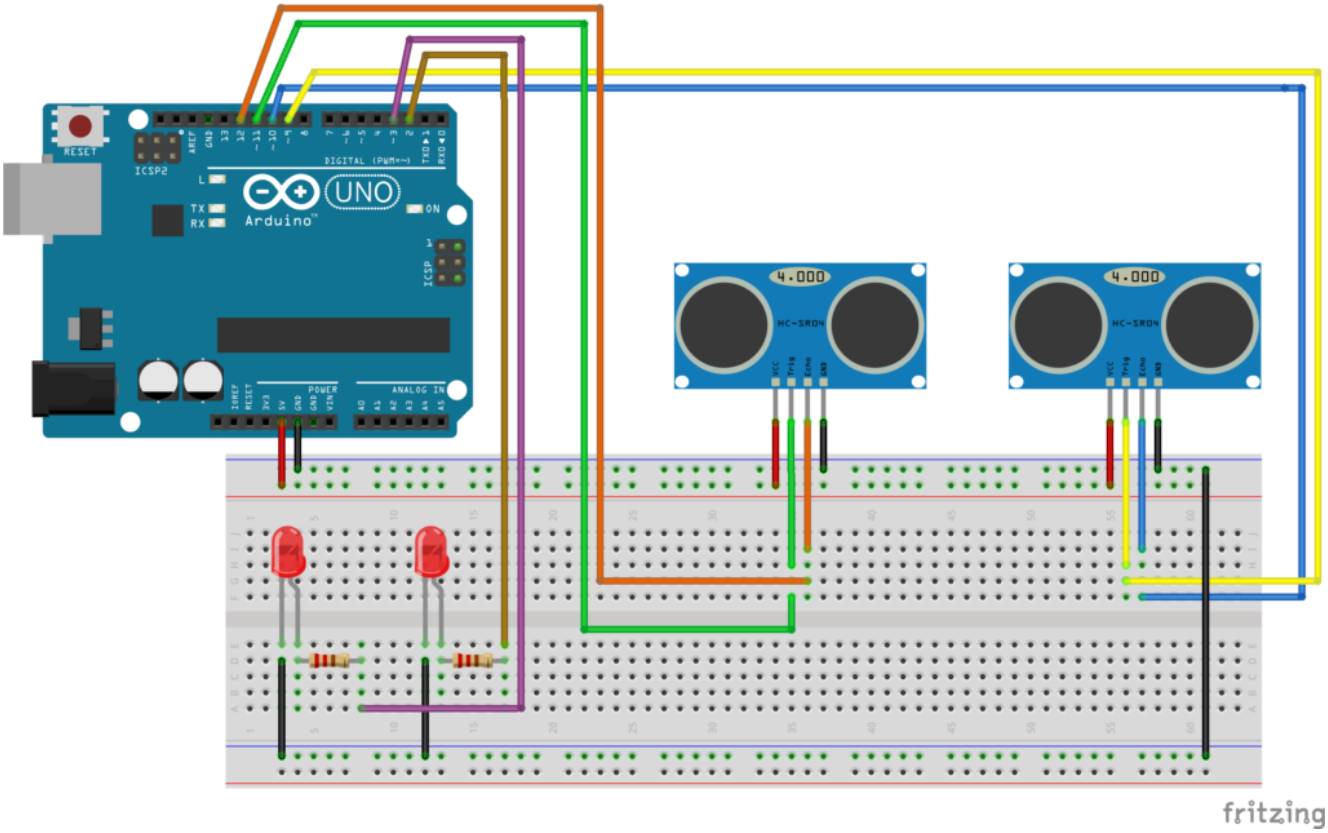
- GND: 0V Ground



Modello Sensore HC-SR04

<u>Collegamento Circuitale</u>:

La realizzazione hardware di circuiti che utilizzano più sensori ad ultrasuoni è relativamente facile. Ognuno dei pin di Echo e di Trig del sensore deve essere semplicemente collegato ai pin di input/output di Arduino.



Collegamento Circuitale

Codice:

Nel caso specifico, in cui si vogliano utilizzare più sensori ad ultrasuoni per misurare simultaneamente la distanza in più punti è opportuno gestire i sensori in modo corretto attraverso il relativo codice. Nel dettaglio, è importante considerare che inizialmente entrambi i sensori generano l'impulso utilizzato per determinare la distanza. In seguito le letture di echo (effettuate mediante l'istruzione pulseIn) sono eseguite in modo sequenziale (una dopo l'altra).

Il Sensore a Ultrasuoni

Obiettivo: Utilizzare un Sensore a Ultrasuoni (HC-SR04) per misurare la distanza.

Componenti elettronici:

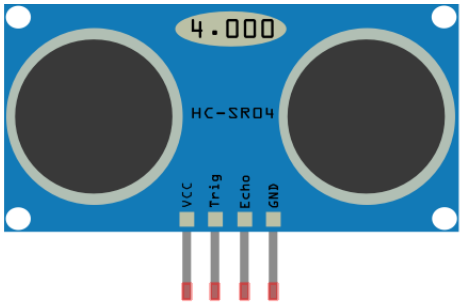
- Arduino UNO
- Breadboard
- 1 Sensore Ultrasuoni (e.g., HC-SR04)

<u>Teoria</u>: Il sensore di prossimità è un dispositivo che permette di rilevare la presenza di oggetti nelle immediate vicinanze, senza che vi sia un effettivo contatto.
Nel caso specifico, il sensore di prossimità ad ultrasuoni sfrutta il principe del Sonar. Degli impulsi sonori (ultrasonici) vengono emessi dal dispositivo il quale attraverso l'eventuale eco di ritorno permette di rilevare la presenza di un oggetto all'interno della portata nominale. Esempi pratici di sensori ad ultrasuoni sono i sensori di retromarcia e di parcheggio utilizzati nelle moderne automobili.
Nel dettaglio, le principali caratteristiche tecniche del sensore ad ultrasuoni HC-SR04 sono:

- Alimentazione: +5V DC
- Angolo di misura: < 30°
- Distanza di rilevamento: da 2cm a 400cm
- Risoluzione: 1cm
- Frequenza: 40kHz

Il sensore, dispone di 4 pin: Vcc (+5V), Trigger, Echo, GND.

- VCC: +5V DC
- Trigger: Genera l'impulso ultrasonico
- Echo: Rileva il segnale ultrasonico di ritorno
- GND: 0V Ground



Modello Sensore HC-SR04

Pertanto considerando la formula che lega velocità, spazio e tempo:

$$s = v*t$$

e la velocità del suono pari a 343 m/s (che espressa in microsecondi diventa 0,0343 m/uS), si ottiene:

$$s = 0,0343*t$$

Considerando inoltre che il suono percorrerà due volte la distanza da misurare (dal sensore all'oggetto e dall'oggetto al sensore); il tempo t ottenuto deve essere diviso per due, ottenendo:

$$s = 0,0343*(t/2)$$

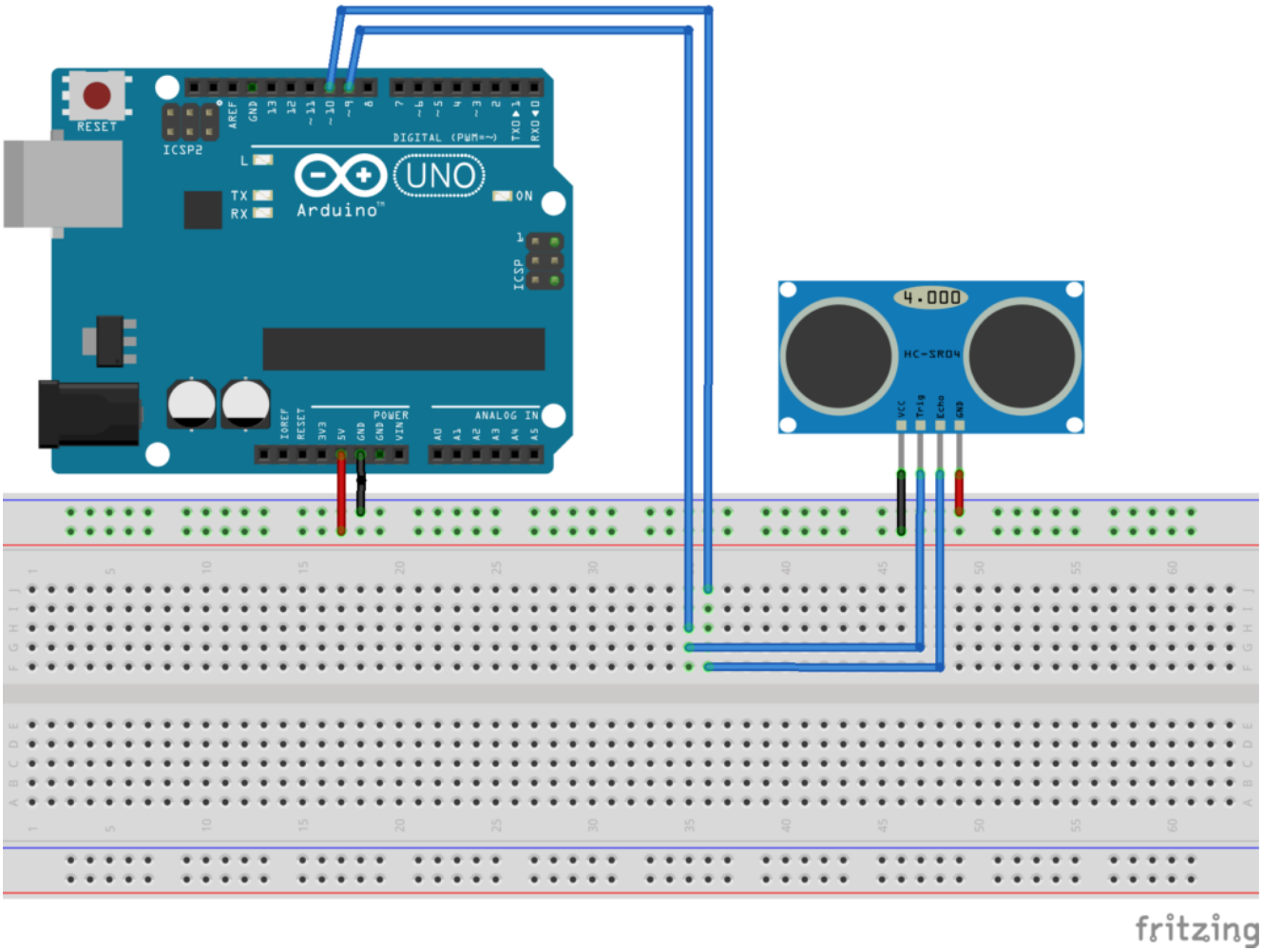
$$s = 0,01715*t$$

$$s = t/58,31$$

Per valutare la presenza di un oggetto è necessario rispettare il seguente “**protocollo**“:

- Il PIN Trigger deve essere dettato alto (HIGH value) per almeno 10microsecondi.
- In automatico il modulo HC-SR04 invierà 8 impulsi ultrasonici ad una frequenza pari a 40kHz.
- Il PIN Echo viene posto in ascolto. Calcolando il tempo di arrivo/ritorno dell’impulso ultrasonico.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

[crayon-66219a2e75fc6385914175/]

Personalizzazioni: E’ possibile modificare il contrasto del display intervenendo sul potenziometro.