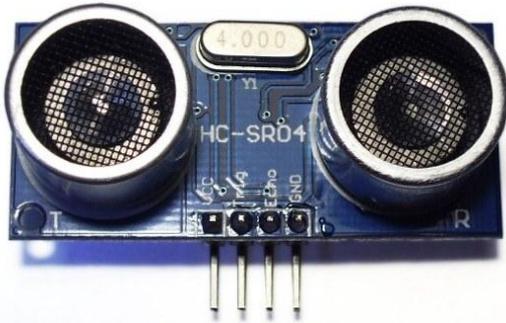


B - sensore di parcheggio



Questo prototipo, peraltro non originale nella sua idea di base, e' di fatto un'evoluzione dell'esercizio 21 dal quale ha derivato gran parte del programma.

Visti il suo basso costo, la sua semplicita' costruttiva e la disponibilita' di staffe su cui fissare il sensore, il prototipo puo' essere facilmente trasformato in un'apparecchiatura fissa da montare su di un veicolo privo di una visibilita' posteriore come ad esempio un camper o un furgone.

Il prototipo e' dotato di un cicalino ed un led che entrano in funzione quando il sensore rileva ostacoli ad una distanza inferiore ai due metri. Il cicalino ed il led si attivano con frequenze via via crescenti quanto piu' vicino e' l'ostacolo.

Attenzione: il sensore intercetta i segnali di ritorno con un "angolo di visuale" di circa 15 gradi per cui intercetta anche l'eventuale segnale di ritorno proveniente dalla strada. Per ottenere una valida misurazione bisogna quindi tenere il modulo ad una altezza da terra o da ostacoli laterali sufficiente ad evitare interferenze. In linea di massima per calcolare la distanza massima misurabile in funzione della posizione del sensore, puo' essere utilizzata la seguente formula:

posizione del sensore = distanza massima misurabile * 0,26 (0,26 e' il seno di 15 gradi)

Questo significa che se vogliamo "vedere" ostacoli ad una distanza massima di due metri dobbiamo tenere il sensore ad una altezza di almeno 52 centimetri da terra.

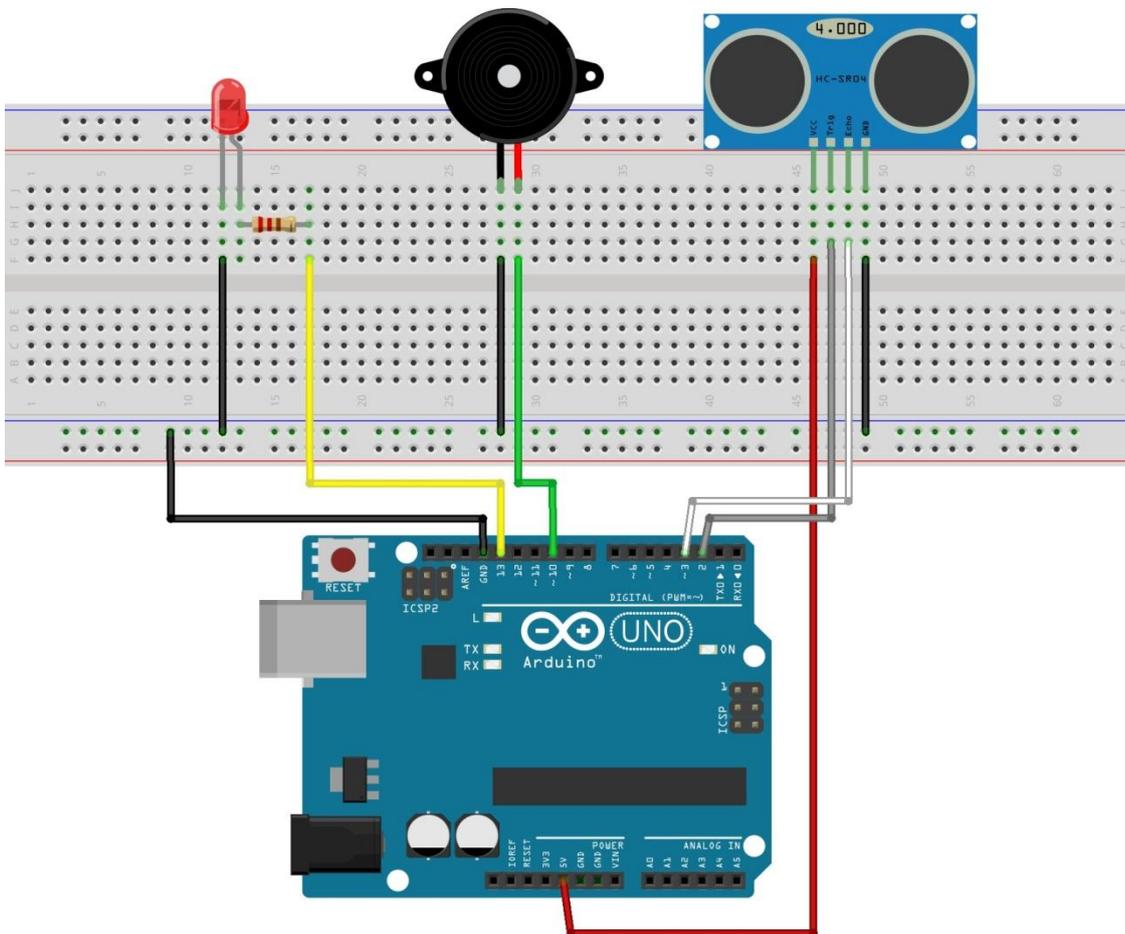
Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili](#)
- [Breve manuale di programmazione](#)

Materiali

- Un modulo HC-SR04
- Un led
- Un buzzer attivo
- Una resistenza da 220 ohm
- Una breadboard

Schema



fritzing

Programma

/ Il cuore del dispositivo e' il modulo HC-SR04 che lancia un fascio di onde ad alta frequenza (ultrasuoni) e rileva eventuali segnali di ritorno dovuti ad un ostacolo posto ad una distanza massima di 2 metri. Il tempo intercorso tra il momento del lancio ed il momento della ricezione del segnale di ritorno viene poi convertito in centimetri e quindi comunicato all'utente tramite l'accensione di un led e l'attivazione di un suono aventi frequenza tanto piu' elevata quanto piu' vicino e' l'ostacolo.*

Lo schema vede il positivo del led collegato ad una resistenza da 220 ohm a sua volta collegata alla porta 13, il positivo del buzzer collegato alla porta 10, il trigger del modulo alla porta 2 e l'echo alla porta 3. Va da se che i negativi del led, del buzzer e del modulo debbano essere collegati alla terra (gnd) e che il positivo del modulo debba essere collegato all'alimentazione da 5 volt/*

```
float cm; // variabile in cui verra' inserita la distanza dall'ostacolo, in centimetri
long tempotrascorso = 0; // variabile utilizzata per misurare il tempo trascorso dal momento di
// accensione della scheda
long precedente = 0; // variabile utilizzata per memorizzare il momento della precedente
// accensione del led e del buzzer
long durata; // variabile in cui sara' inserita la durata (in millisecondi) della pausa del
```

Arduino: sensore di parcheggio

```
// led e del buzzer

void lucesuono (void)      // ***** routine di attivazione del led e del buzzer *****
{
  tempotrascorso = millis();      // acquisisce il tempo trascorso (in millisecondi)
  // dall'accensione di arduino
  if(tempotrascorso - precedente > durata) // verifica il tempo trascorso dall'ultima accensione
  // del led e del buzzer. Se il tempo trascorso dall'ultima accensione e' maggiore della durata
  // decisa dalla routine chiamante, lancia l'accensione del led e del buzzer, in caso contrario ritorna
  // alla routine chiamante senza attivare alcunché
  {
    // accensione del led e del buzzer, eseguiti solo per condizione vera
    precedente = tempotrascorso;      // salva il momento temporale (in millisecondi) in cui
  // vengono accesi led e buzzer
    digitalWrite(13, HIGH);          // attiva il led
    digitalWrite(10, HIGH);         // attiva il buzzer
    delay (50);                      // attende 50 millesimi di secondo
    digitalWrite(13, LOW);          // spegne il led
    digitalWrite(10, LOW);          // spegne il buzzer
  }
}

void setup()
{
  pinMode(2, OUTPUT); // definisce la porta digitale 2 (il trigger) come porta di output
  pinMode(3, INPUT);  // definisce la porta digitale 3 (l'echo) come porta di input
  pinMode(13, OUTPUT); // definisce la porta digitale 13 (il led) come porta di output
  pinMode(10, OUTPUT); // definisce la porta digitale 10 (il buzzer) come porta di output
}

void loop()
{
  digitalWrite(2, LOW); //disattiva il lancio del fascio di ultrasuoni (qualora fosse attivo)
  delayMicroseconds(2); // attende 2 microsecondi
  digitalWrite(2, HIGH); // attiva il lancio del fascio di ultrasuoni
  delayMicroseconds(10); // attende 10 microsecondi (il tempo richiesto dal modulo HC-SR04)
  digitalWrite(2, LOW); // disattiva il lancio del fascio di ultrasuoni
  cm = pulseIn(3, HIGH) / 58.0; // rileva il segnale di ritorno e lo converte in centimetri
  // (il divisore 58.0 e' una costante suggerita da altri programmi e sperimentalmente verificata)
  if (cm < 200) // se l'ostacolo e' a meno di 2 metri di distanza attiva la routine di allarme
  {
    durata = cm*5; // Pone la durata della pausa di silenzio pari ad un multiplo della distanza
  // (e quindi proporzionale alla stessa), il moltiplicatore 5 e' stato empiricamente definito e puo'
  // essere ovviamente modificato per aumentare o diminuire la frequenza di attivazione dell'allarme
    lucesuono (); // lancia la routine di attivazione di luce e suono
  }
  delay(50); // attende 50 millisecondi prima di rilanciare il ciclo
}
```