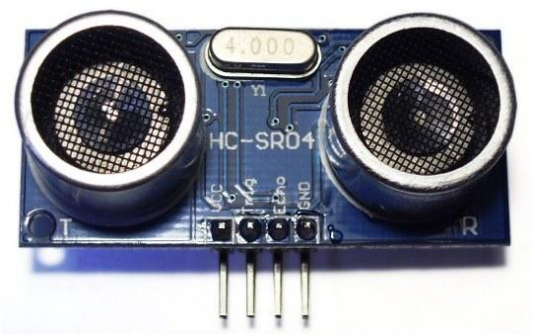
B - sensore di parcheggio – parking sensor (some notes at end of this section)



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 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 (PDF scaricato nell'area di download)
- Breve manuale di programmazione (PDF scaricato nell'area di download)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Here some notes about this project, translated by google translator

This prototype, not original in its basic idea, is an evolution of previous example 21, from which he derived a large part of the program. The prototype is equipped with a buzzer and a LED that starts flashing when sensor detects obstacles at less than two meters distance. It buzzes and flashes with increasing frequency, as more near is the obstacle.

Having regard to its low cost, its simplicity constructive and availability of brackets on which to secure the sensor, the prototype can easily be transformed into fixed equipment to be mounted on a vehicle.

Warning: sensor intercepts the return signals with a "viewing angle" of about 15 degrees. It intercepts also any return signal coming from the road. To get a valid signal you have to keep the module at a height from the ground sufficient to avoid interference. In principle, to calculate the

maximum measurable distance as a function of sensor position (altitude), can be used the following formula:

position sensor = maximum measurable distance * 0.26 (0.26 is 15 degrees sine)

This means that if we want see obstacles starting from a distance of two meters, we have to keep sensor at a height of at least 52 centimeters above ground.

Note: This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

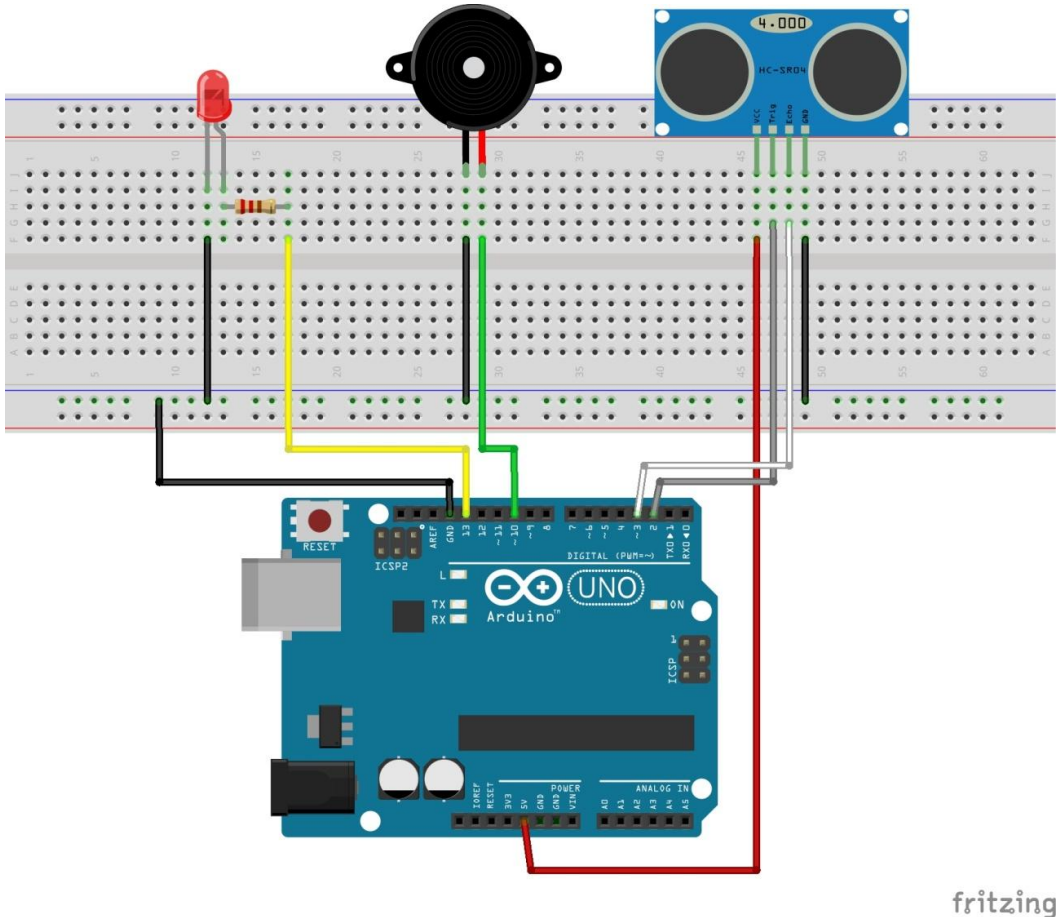
- Projects collection
- Movies on youtube
- About Arduino and components (italian; pdf will be downloaded in your download area
- Quick programming guide (almost english; pdf will be downloaded in your download area)

For any questions or suggestions about this note (and on its english translation), please write to giocarduino@libero.it (simple words and short sentences, please)

Materiali

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

Schema



Programma

Download del programma via Dropbox
Program download via Dropbox

```
/*  Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo.
Per rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
trasferimento nell'IDE, premendo CTRL+T.
*
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
*
*-----------------------------------------------------------------------------------------------
* Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
*
The device heart is the HC-SR04 appliance, that launches a beam of high-frequency waves (ultrasound)
and detects any return signals from an obstacle located at a maximum distance of 2 meters. The time
elapsed between launch and return signal is converted into centimeters and then communicated to user
by flashing a LED and generating a sound, having frequency more high as most near is the obstacle.
*
Schematic sees the LED positive leg connected to a 220 ohm resistor in turn connected to pin13, the
buzzer positive leg connected to the pin10, the "trigger" (on HCSR04) to pin 2 and the echo to pin
3. It goes without saying that the negative of LED, buzzer and HCSR04 should be connected to ground
(gnd), and that the positive of the HCSR04 should be connected to the 5 volt powered pin
*-----------------------------------------------------------------------------------------------
*/

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
//  led e del buzzer

//
// ********** routine di attivazione del led e del buzzer *********
// ****************managing led and buzzer *************************
//
void lucesuono (void)          {
  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 alcunche' - checks LED and buzzer frequency. If the
  time elapsed since the last power is greater than the duration decided by calling routine,
  activates the LED and buzzer, if not back to calling routine without activating anything

 */
  {     // 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)
  Detects the return signal and converts it to centimeters (the divisor 58.0 is a constant suggested
  by other programs and experimentally verified)
  */
   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
Calculates the pause duration equal to a multiple of distance (and therefore proportional to the
same), the multiplier 5 is been empirically defined and can be obviously modified to increase or
decrease the alarm frequency
*/
    lucesuono ();      // lancia la routine di attivazione di luce e suono
  }
  delay(50);           // attende 50 millisecondi prima di rilanciare il ciclo
}
```