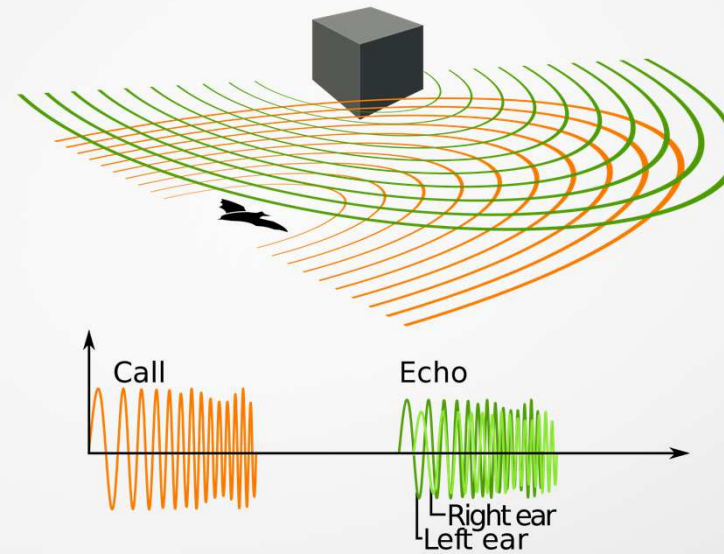


Bentornati ragazzi, questa volta tramite l'utilizzo della scheda [Arduino](#) e di una serie di componenti ricreeremo il sistema che i pipistrelli usano per navigare l'ambiente che li circonda, cioè l'[ecolocalizzazione](#).



Come funziona l'ecolocalizzazione?

Gli ultrasuoni sono quella gamma di frequenze che va da 20kHz in sù



Il pipistrello emette **ultrasuoni** attraverso la bocca e ne raccoglie l'eco con le orecchie. Il suo cervello è in grado di calcolare la distanza dell'ostacolo che ha provocato l'eco.

Vediamo adesso come ricostruire questa capacità del pipistrello con **Arduino**

Verifichiamo se abbiamo tutto il necessario ed
identifichiamo i componenti:

Resistenze



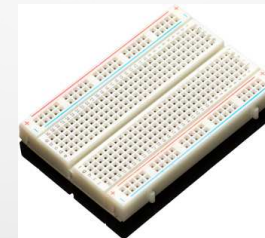
Connettori



Cavo USB



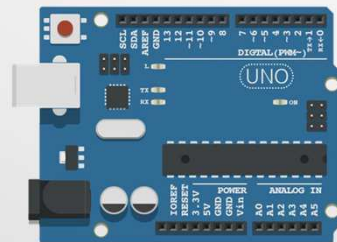
Led



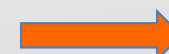
Breadboard



Arduino



Sensore ad ultrasuoni



Cavo alimentazione



2X

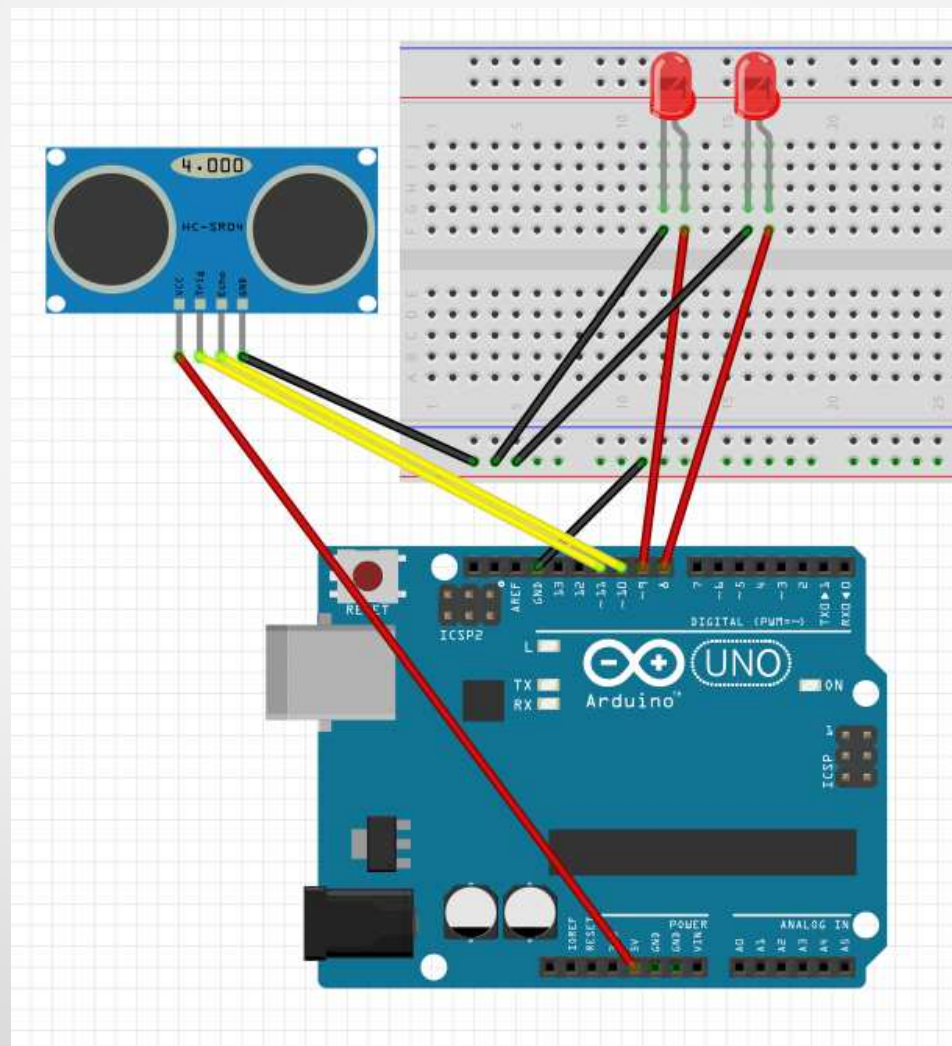
Il sensore è dotato di due capsule, la prima emette gli ultrasuoni, la seconda ne raccoglie l'eco.



I due pin laterali permettono di alimentare il sensore

Il pin Trig è collegato alla capsula che emette gli ultrasuoni, mentre il pin Echo è collegato alla capsula che riceve.

Realizziamo adesso il circuito



Programmare Arduino...

Questi due comandi permettono ad Arduino di ricevere i dati dal sensore e convertirli in un unità di misura, in questo caso centimetri.

```
int trigPin = 11;
int echoPin = 10;
int ledRosso = 9;
int ledVerde = 8;

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledRosso, OUTPUT);
  pinMode(ledVerde, OUTPUT);
}

void loop () {
  long durata, distanza;
  digitalWrite(trigPin, LOW);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  durata = pulseIn(echoPin, HIGH);
  distanza = 0.034 * durata / 2;

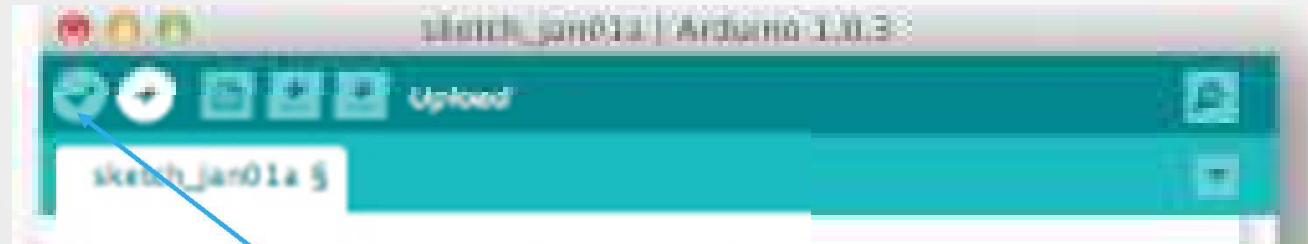
  if(distanza <= 20)
  {
    digitalWrite(ledRosso, HIGH);
    digitalWrite(ledVerde, LOW);
  }
  else
  {
    digitalWrite(ledRosso, LOW);
    digitalWrite(ledVerde, HIGH);
  }

  delay(1000);
}
```



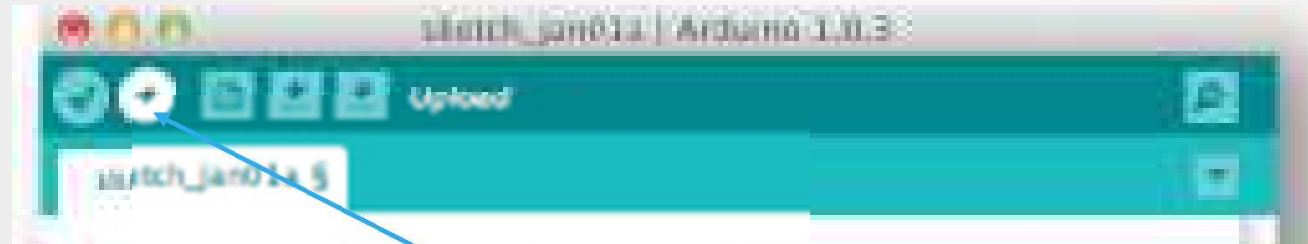

Ricordiamo adesso i due pulsanti presenti nella barra in alto sull'IDE

- ✓ Serve per controllare il codice scritto
- ➔ Serve per trasferire nella scheda il codice scritto



Serve per controllare il codice scritto


Premendo il primo pulsante IDE controlla che non ci siano errori nella scrittura del programma. Se questi sono presenti saranno segnalati in una finestra sistemata in basso sull'IDE, altrimenti verrà restituito il valore della memoria occupata sulla scheda



Se è tutto ok allora possiamo premere il secondo pulsante. Il codice verrà adesso trasferito nella scheda



Serve per trasferire nella scheda il codice scritto



Se i led non si accendono bisogna
ricontrollare il circuito verificando se i cavi
sono stati inseriti nei pin **Arduino** indicati
nel programma

```
int trigPin = 11;  
int echoPin = 10;  
int ledRosso = 9;  
int ledVerde = 8;
```

Alle variabili intere
trigPin, echoPin,
ledRosso, ledVerde
vengono assegnati i
valori 11, 10, 9, 8