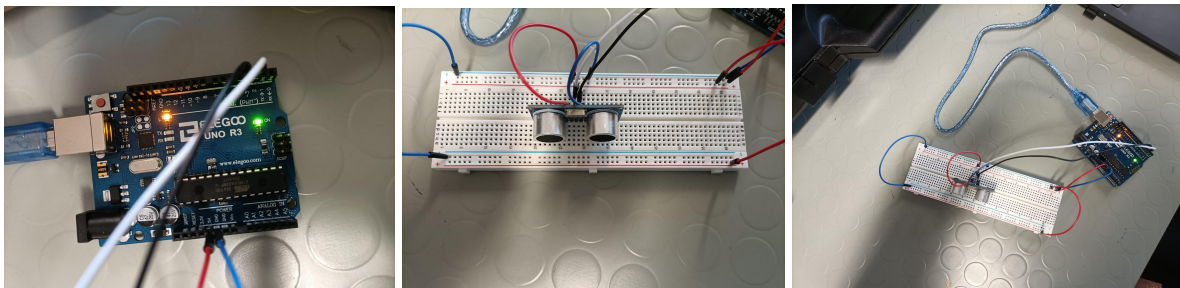


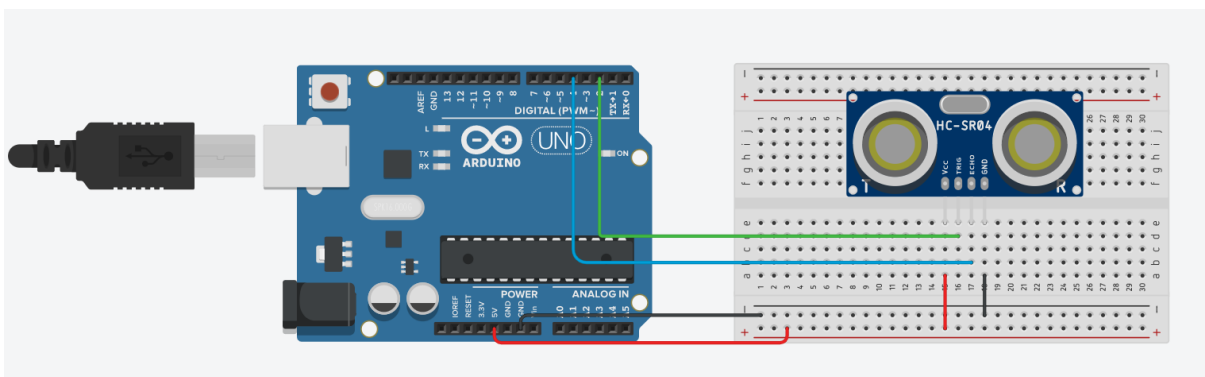
Alberto Aliberti
Michel Rossi
Michele Taschetta

Relazione esperienza di laboratorio: misura della velocità del suono

- **OBIETTIVI:**
 - Imparare il funzionamento di Arduino
 - Imparare a costruire un circuito
 - Misurare la velocità del suono
- **STRUMENTI UTILIZZATI:**
 - UNO R3 Controller Board
 - 830 Tie-Points Breadboard
 - Breadboard Jumper Wires
 - Ultrasonic Sensor
 - Computer (Software: Arduino, Excel)
 - Metro a nastro (Sensibilità di 1 mm)



- **SCHEMA ELETTRICO:**



Qua sopra è riportato lo schema elettrico utilizzato.

- **PROCEDIMENTO:**

Assemblaggio Breadboard con ausilio di Breadboard Jumper Wires e Ultrasonic Sensor. Collegamento 830 Tie-Points Breadboard a UNO R3 Controller Board.

Collegamento tramite cavo USB al computer.

Modifica del codice "Arduino" registrante misurazioni temporali in microsecondi di un Ultrasonic Sensor inviante onde ultrasoniche contro bersagli a distanze calcolate attraverso metro a nastro con sensibilità 1 mm.

La distanza è stata misurata con un metro. Il suo valore non è stato raddoppiato in quanto tutti i calcoli sono stati eseguiti dimezzando i valori dei tempi.

Calcolo attraverso i dati relativi a distanza e tempo della velocità del suono.

Calcolo dell'errore relativo.

Formule utilizzate:

$$s/t = v$$

$$e = \Delta a/a + \Delta b/b$$

Variabili:

$2t = 2b$ = tempo registrato dall'apparecchiatura

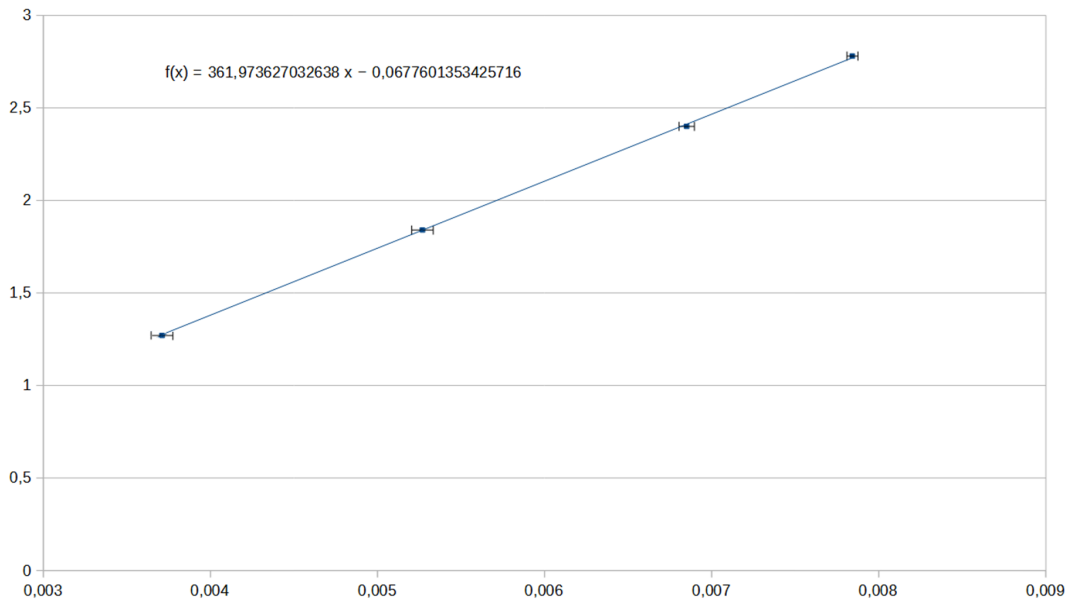
$s = a$ = distanza misurata con metro a nastro

s/t = velocità del suono misurata

e = errore relativo

- **ELABORAZIONE DATI:**

Media tempi (s) x 10 ⁻³	Distanza (m)	Valori massimi (s) x 10 ⁻³	Valori minimi (s) x 10 ⁻³	Errore (s) x 10 ⁻⁵
7,842	2,78	7,887	7,822	3,3
6,851	2,40	6,900	6,808	4,1
5,270	1,84	5,299	5,231	3,4
4,417	1,27	4,435	4,382	2,7



Osservando i dati si può subito notare che le medie dei tempi sono distanziate in modo sufficientemente regolare al variare della distanza tra il rilevatore e il bersaglio come giusto che sia trattandosi di una frazione.

Inizialmente sono stati riscontrati dati per valori minimi non coerenti con la media. Sono stati dunque creati dei programmi su Fogli Google che permettessero, attingendo dai dati. Il problema nel programma Arduino è purtroppo tuttora presente in quanto non siamo stati in grado di trovarlo e risolverlo.

- **CONCLUSIONI**

Tramite i dati registrati si può giungere alla conclusione per cui nello spazio ove si sono svolte le misurazioni la velocità del suono si aggira attorno ai 361.97 m/s con un errore delle misure stimate al $\approx 0.8\%$, plausibile dato l'ambiente non convenzionale per sperimentazioni e le apparecchiature utilizzate. Non sono, purtroppo, state prese misure dell'ambiente ove è stato svolto l'esperimento.

- **PROPOSTE PER IL FUTURO**

Sarebbe interessante analizzare come varie fonti di interferenza vadano a variare i dati (Umidità maggiore, onde di varia frequenza a varie distanze dal sensore, interferenza dei liquidi...).

- **PROGRAMMA FOGLI**

Ecco in seguito il semplice programma utilizzato su Fogli utilizzato per calcolare il tempo minimo partendo dalla stringa di dati relativi ai tempi.

```
=MIN("INTERVALLO DATI")/2
// Questo codice prende semplicemente il valore minimo del
tempo registrato e successivamente lo divide per 2 poiché si
tratta di un tempo preso come andata e ritorno delle onde
```

● PROGRAMMA ARDUINO

Inizialmente il codice stampava 50 misurazioni di tempi mentre, dopo le nostre modifiche, il codice chiedeva in input la distanza e dava, in output, per 10 volte le 50 misurazioni; alla fine di quest' ultime stampava: la media dei tempi, il valore massimo, il valore minimo e la velocità calcolata con la distanza data in input fratto la media dei tempi.

In questo modo ci siamo risparmiati moltissimi calcoli.

Le aggiunte che abbiamo effettuato nel codice sono le seguenti:

```
[\code]
```

```
float distanza;
String distanza_string = "";
/*
  La variabile distanza_string è necessaria per leggere l'input
  del monitor seriale, in modo tale da poter calcolare la velocità
*/
```

```
[\code]
```

```
void loop()
{
  if (Serial.available() > 0)
  {
    /*
      IL metodo .available() della classe Serial restituisce il
      numero di caratteri presenti nell'input del serial monitor,
      la condizione si verificherà solo se il numero dei caratteri
      è maggiore di 0
    */

    char ch = Serial.read();
    if ((ch == '\n') || (ch == '\r'))
    {
      /*
        il metodo .read() legge un carattere per volta dell'input del
        serial monitor; quando viene premuto il tasto invio, Arduino
        aggiunge alla fine dell'input il carattere '\n' o '\r' o
        entrambi, a seconda del sistema operativo, il che vuol dire
        che se il carattere letto è uguale ad o \n o \r allora ci
        troviamo alla fine della stringa e sarà dunque possibile
        procedere con le misurazioni, in caso contrario il carattere
        verrà 'accodato' alla stringa
      */

      distanza = distanza_string.toFloat();
    }
  }
}
```

```

distanza_string serve per calcoli matematici, quindi
bisogna convertirla in float, per fare ciò basta usare
il metodo .toFloat() della classe String
*/

[\code]

for (j = 0; j < 10; j++)
{
    int minimo = 9999999;
    int massimo = 0;

    float somma = 0;
    // La somma servirà per calcolare la media, verrà aumentata
    // ad ogni misurazione

    for (i = 0; i < 50; i++)
    {
        // Per ogni distanza inserita il codice eseguirà
        // 10 volte 50 misurazioni

        [\code]

        somma += t;
        // incremento della somma
        // la variabile t rappresenta il tempo misurato

        if (t < minimo)
            minimo = t;
        /*
        Per aggiornare il minimo, ad ogni misurazione si
        controlla se il tempo è minore del minimo, se si, allora
        la variabile minimo verrà aggiornata a tale tempo
        */

        if (t > massimo)
            /*
            Per aggiornare il massimo, ad ogni misurazione si
            controlla se il tempo è maggiore del massimo, se si la
            variabile massimo verrà aggiornata a tale tempo
            */
            massimo = t;
    }
    float media = somma / 50; // calcolo della media

    Serial.print(" - - - - Media dei tempi: - - - - ");
    Serial.println(media);

```

```

Serial.print(" - - - - Minimo: - - - - ");
Serial.println(minimo);
Serial.print(" - - - - Massimo: - - - - ");
Serial.println(massimo);
Serial.print(" - - - - Distanza: - - - - ");
Serial.println(distanza);
Serial.print(" - - - - Velocità: - - - - ");
Serial.println(distanza / media * 1000000);
Serial.print("- - - - - - - - - - - - ");
Serial.println("\t");

/*
  In questa parte di codice stampiamo il massimo, il minimo e
  la media attraverso il metodo .println() della classe Serial
*/
}

distanza_string = ""
// Riportiamo la stringa della distanza a "" per poter
// accettare nuove distanze
}

else
  distanza_string += ch;
}
}
else
  Serial.println("Scrivi una distanza");
}

```