

Misuratore portatile ad ultrasuoni

Gianpiero Gaeta, Andrea Fuso, Stefano Fusari

30 giugno 2016

Indice

1 Scopo	2
2 L'idea del progetto	2
3 Strumenti utilizzati	3
4 Progettazione case	14
5 Analisi tecnica HC-SR04	17
5.1 Construzione	17
5.2 Funzionamento	17
6 Progettazione sistema elettrico	18
7 Taratura dello strumento	21
7.1 Programmazione codice di taratura	21
7.1.1 Codice Arduino per la taratura	21
7.1.2 Spiegazione codice	24
7.1.3 Taratura angolo	25
7.2 Elaborazione dati	29
8 Programmazione codice strumento	49
8.1 Spiegazione parti di misurazione	57
9 Conclusioni	61

1 Scopo

Il nostro progetto si pone l'obiettivo di realizzare un dispositivo portatile che sia in grado di fornire misurazioni di distanze accurate e che permetta all'utente di decidere il livello di confidenza della misura e riporti in tempo reale l'incertezza, tutto questo racchiuso in dispositivo dalla dimensioni compatte.

2 L'idea del progetto



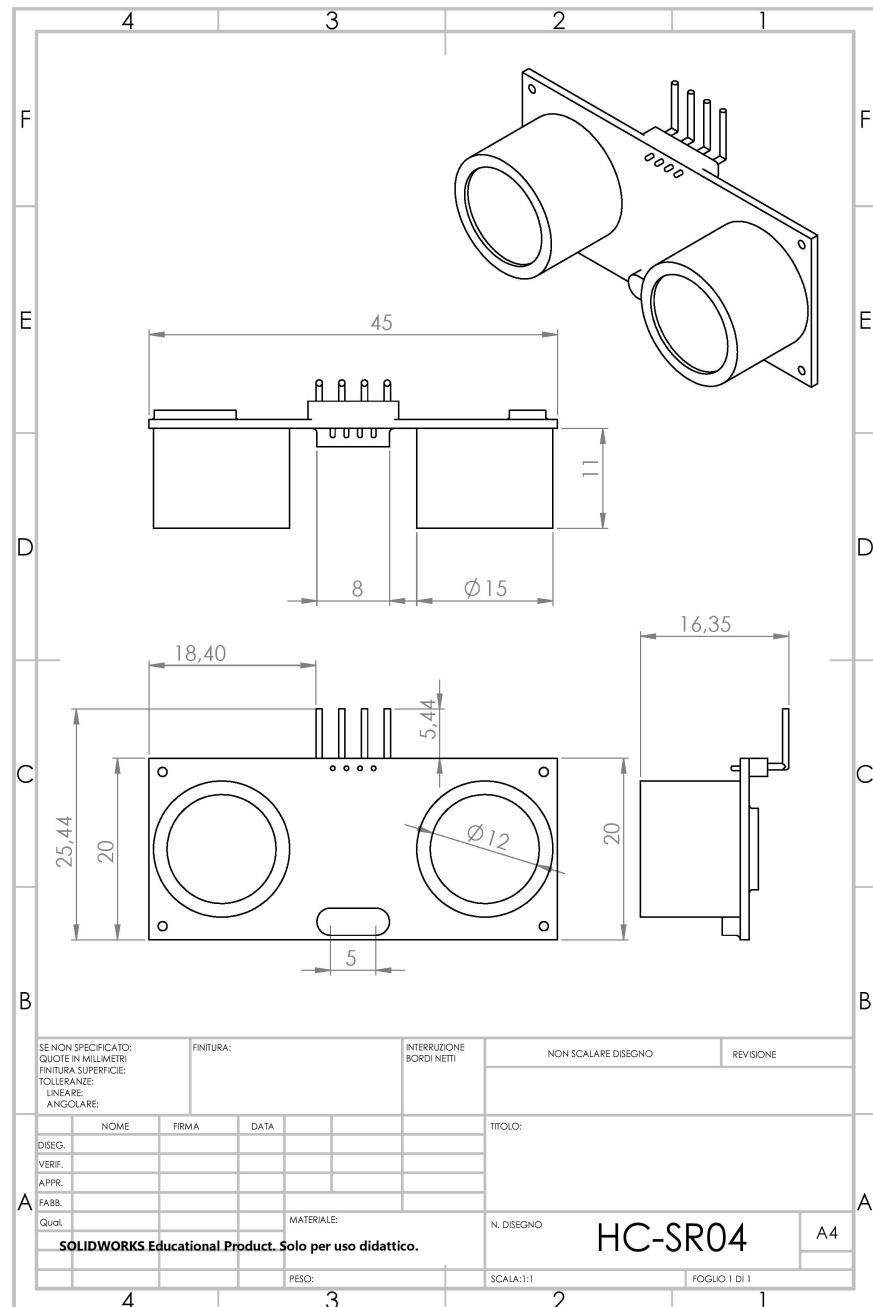
L'idea di creare questo dispositivo è nata dopo aver scoperto la piattaforma «Arduino», che ci ha subito appassionato. Grazie ad una scheda economica è infatti possibile collegare diversi sensori e programmare il tutto per creare progetti d'ogni tipo. Inizialmente dopo i primi progetti per comprenderne il funzionamento ci siamo cimentati nella costruzione di un robot a guida automatica che usava i sensori ad ultrasuoni «HC-SR04» per orientarsi (<https://www.dropbox.com/sh/46z8ah0egx1o5m2/AACuXsWGradj6vkoSQPtNBLea?dl=0>).

Dopo aver intrapreso il corso di «Misure meccaniche e termiche» abbiamo pensato di provare a tarare questi sensori e capirne a fondo il funzionamento e l'angolo che descrive il campo di misura in quanto durante la costruzione del robot, ci siamo imbattuti in diversi problemi riguardanti questi sensori da pochi euro.

Avendo effettuato un laboratorio sulla taratura e essendo interessati all'argomento abbiamo avuto questa idea.

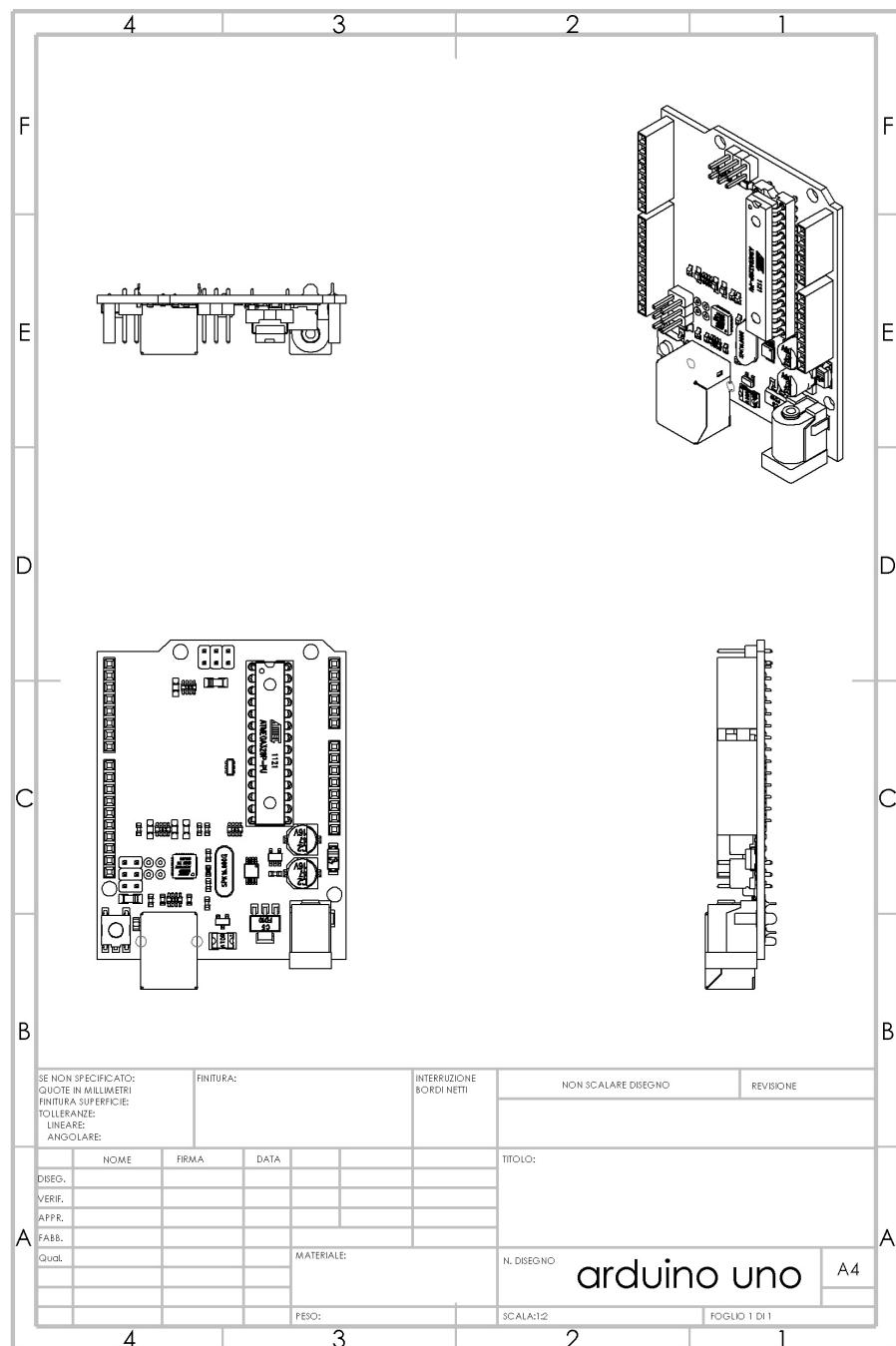
3 Strumenti utilizzati

- Sensore ad ultrasuoni HC-SR04



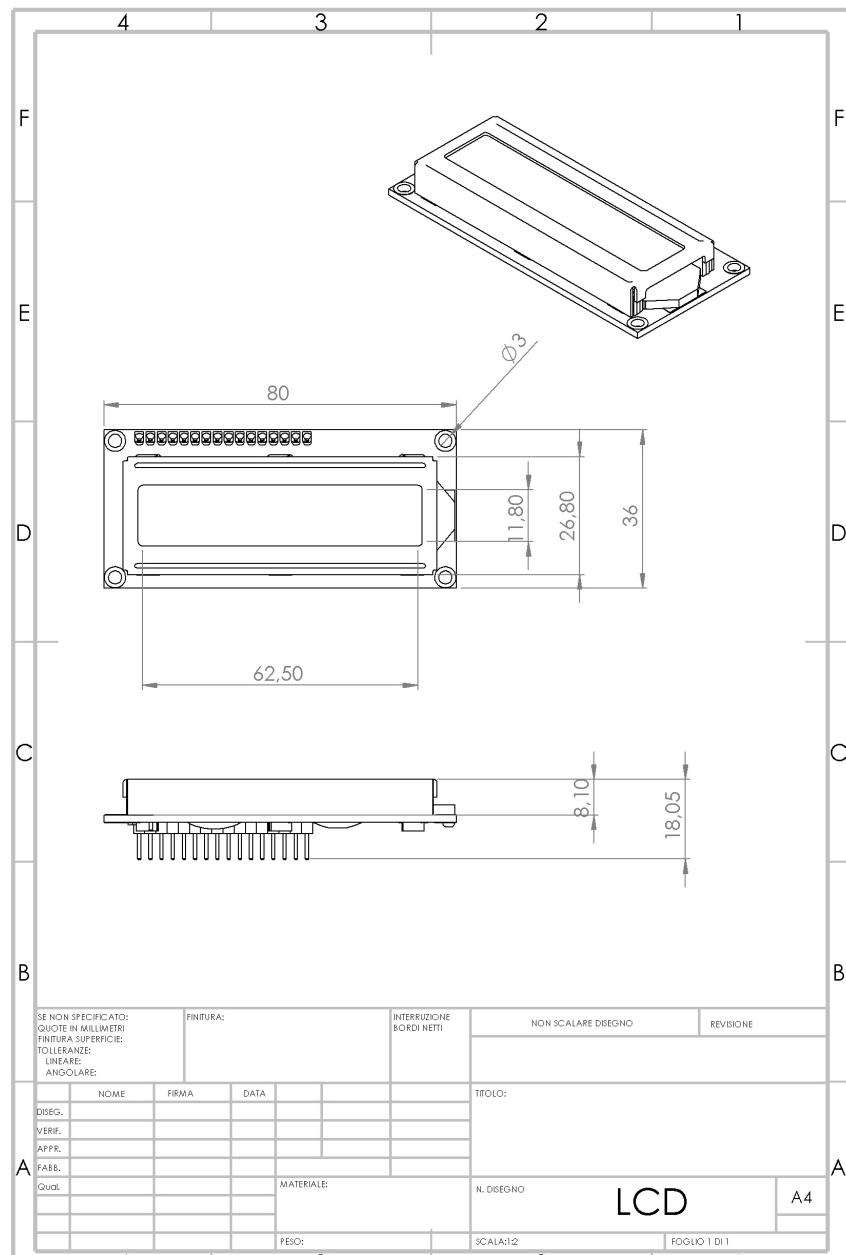
Tensione di alimentazione	DC 5 V
Corrente di alimentazione	15 mA
Frequenza di funzionamento	40 Hz
Trigger Input Signal	10 us TTL pulse

- Arduino Uno



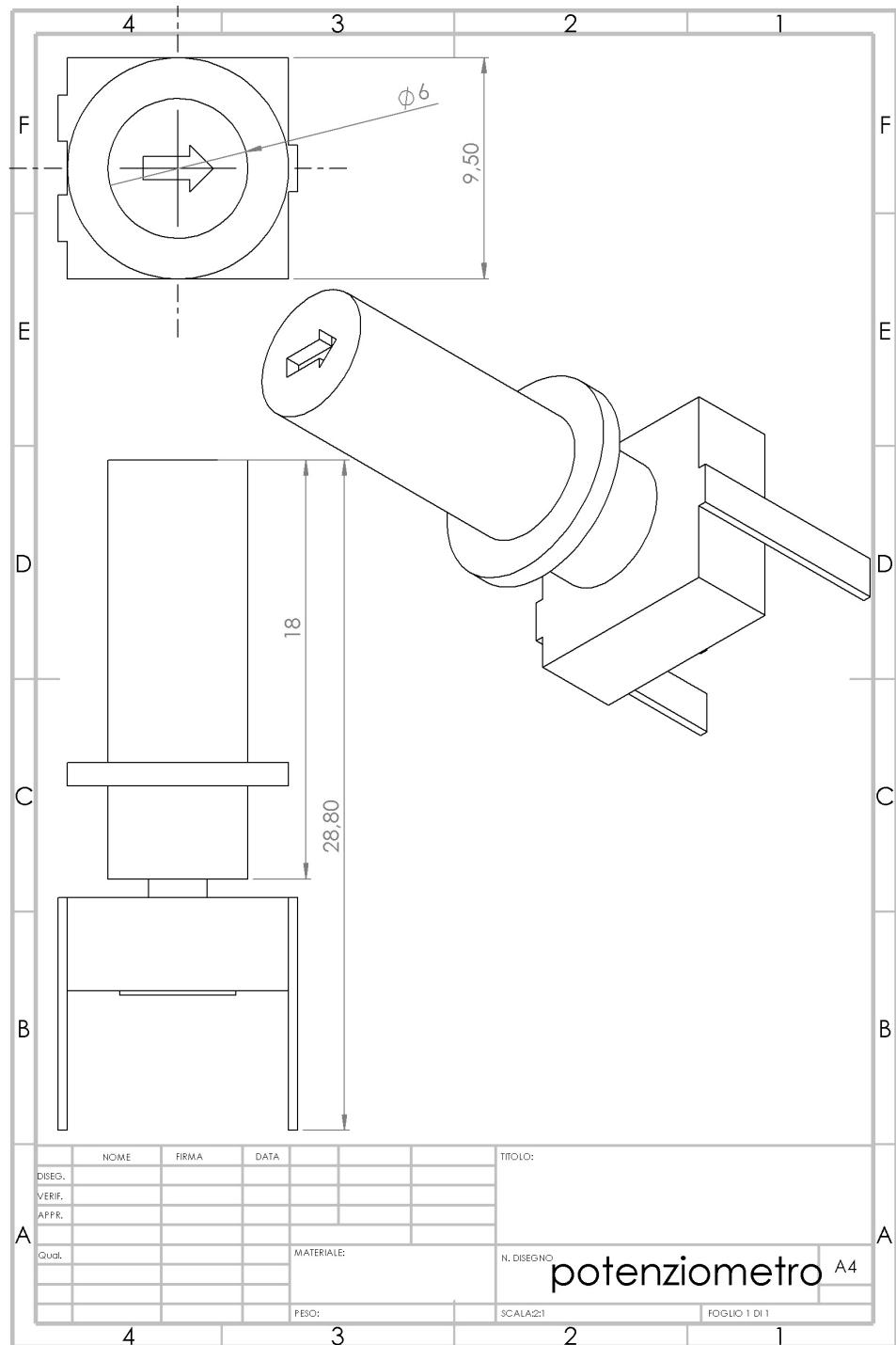
Tensione di esercizio	5V DC
Numero totale di Pin	20
Pin digitali	8 I/O
PWM	6
Analogici	6
Corrente per pin	40mA
Alimentazione consigliata	6-20 V DC
N° bit	8

- Display LCD 16x2



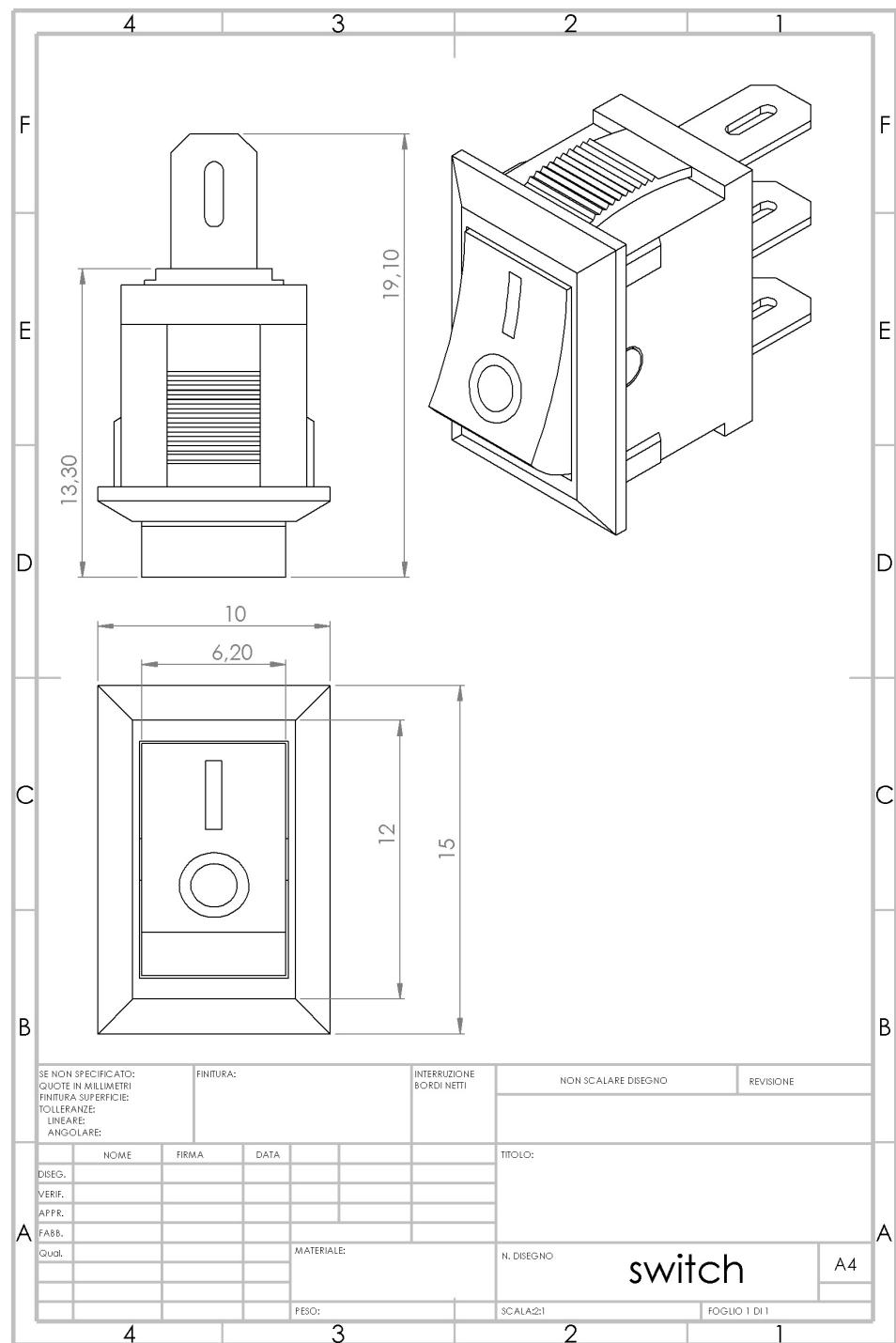
Tensione di alimentazione	5 V
Prontezza display	200-250 ms
Numero caratteri massimo	32

- Potenziometro



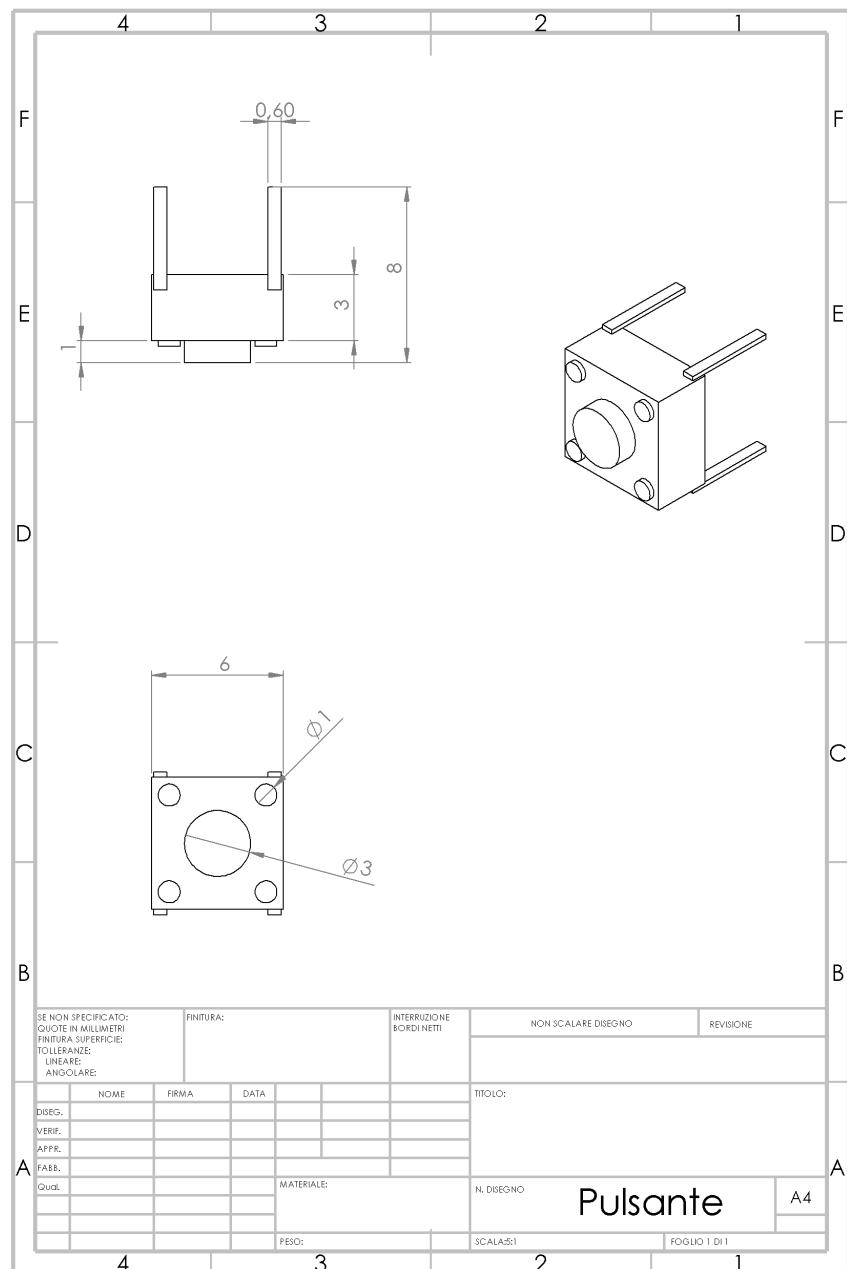
Tensione di funzionamento | 5V

- Switch

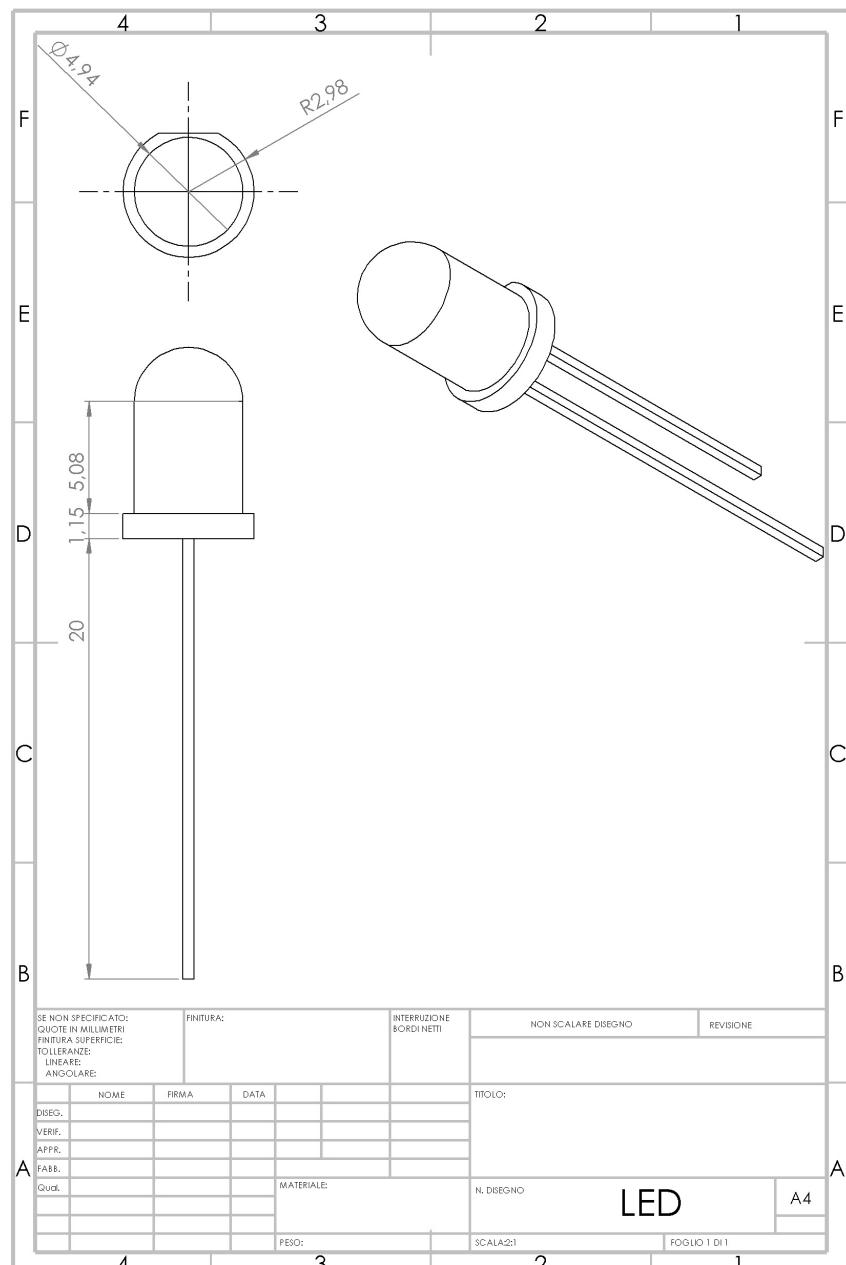


Resistenza	50 mΩ
Corrente di alimentazione	6A
Tensione di alimentazione	A.C. 125V

- Pulsante

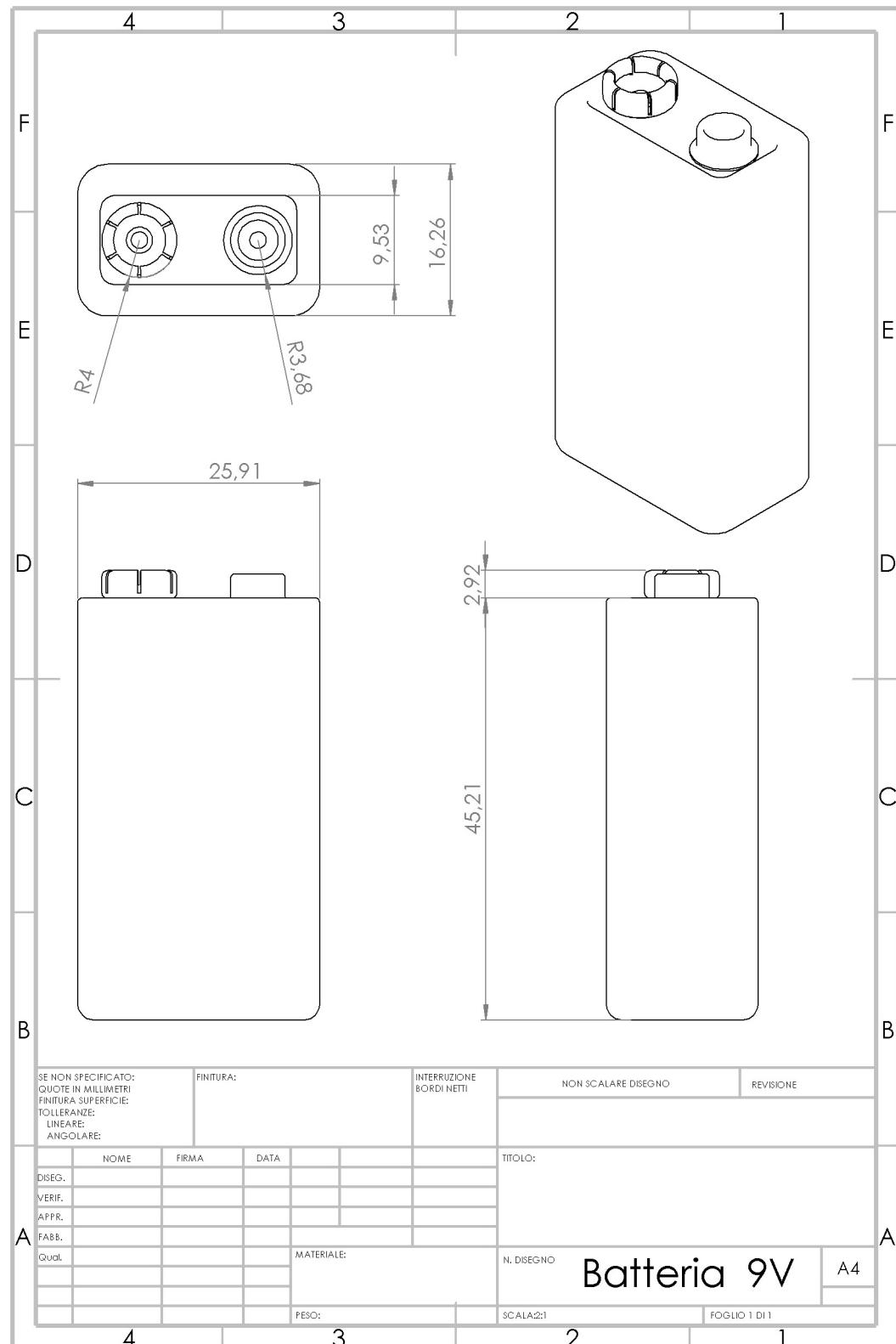


- Led



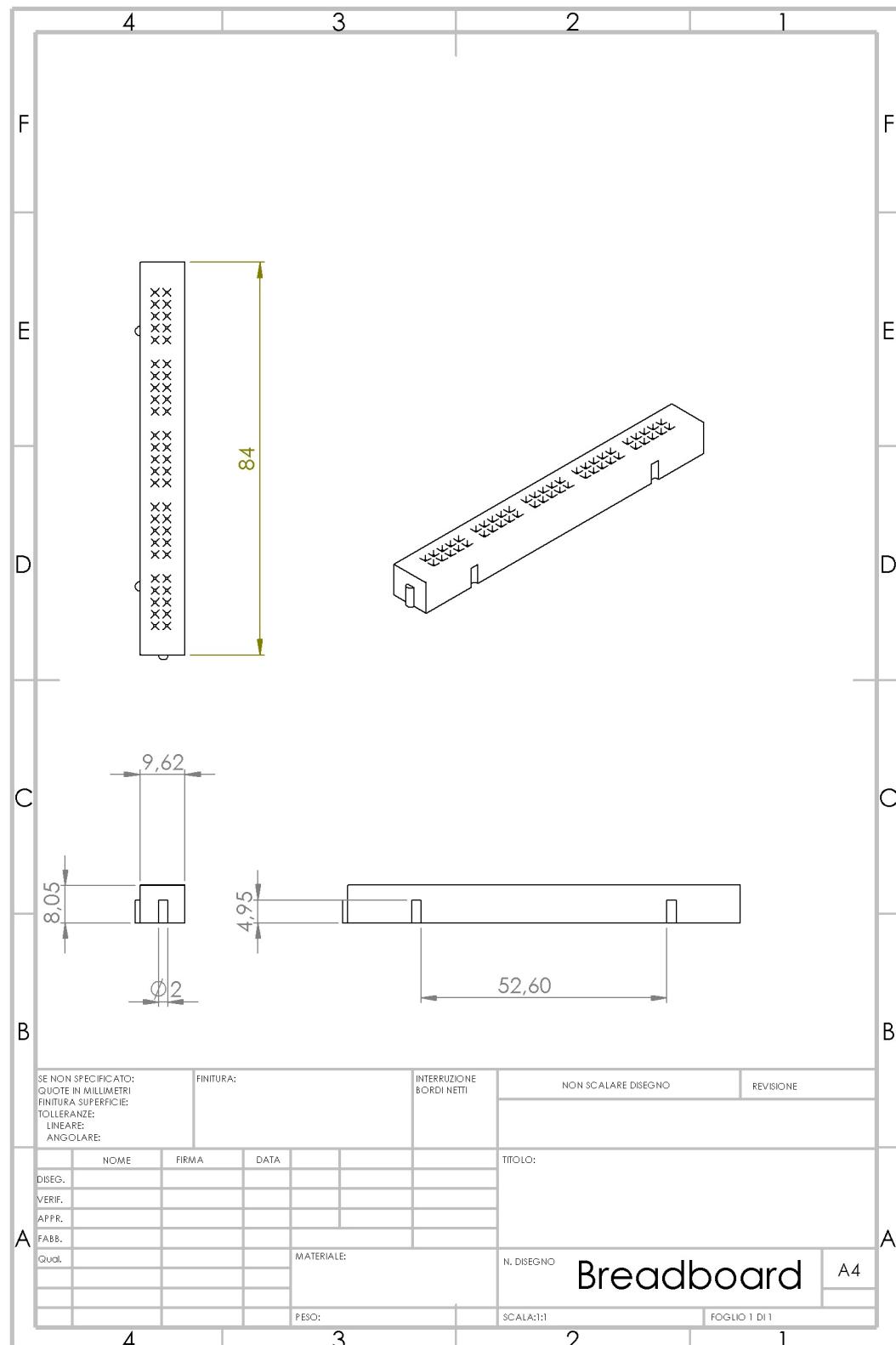
Tensione di alimentazione	5V
Corrente di alimentazione	30 mA
Resistenza	220Ω

- Batteria 9 V



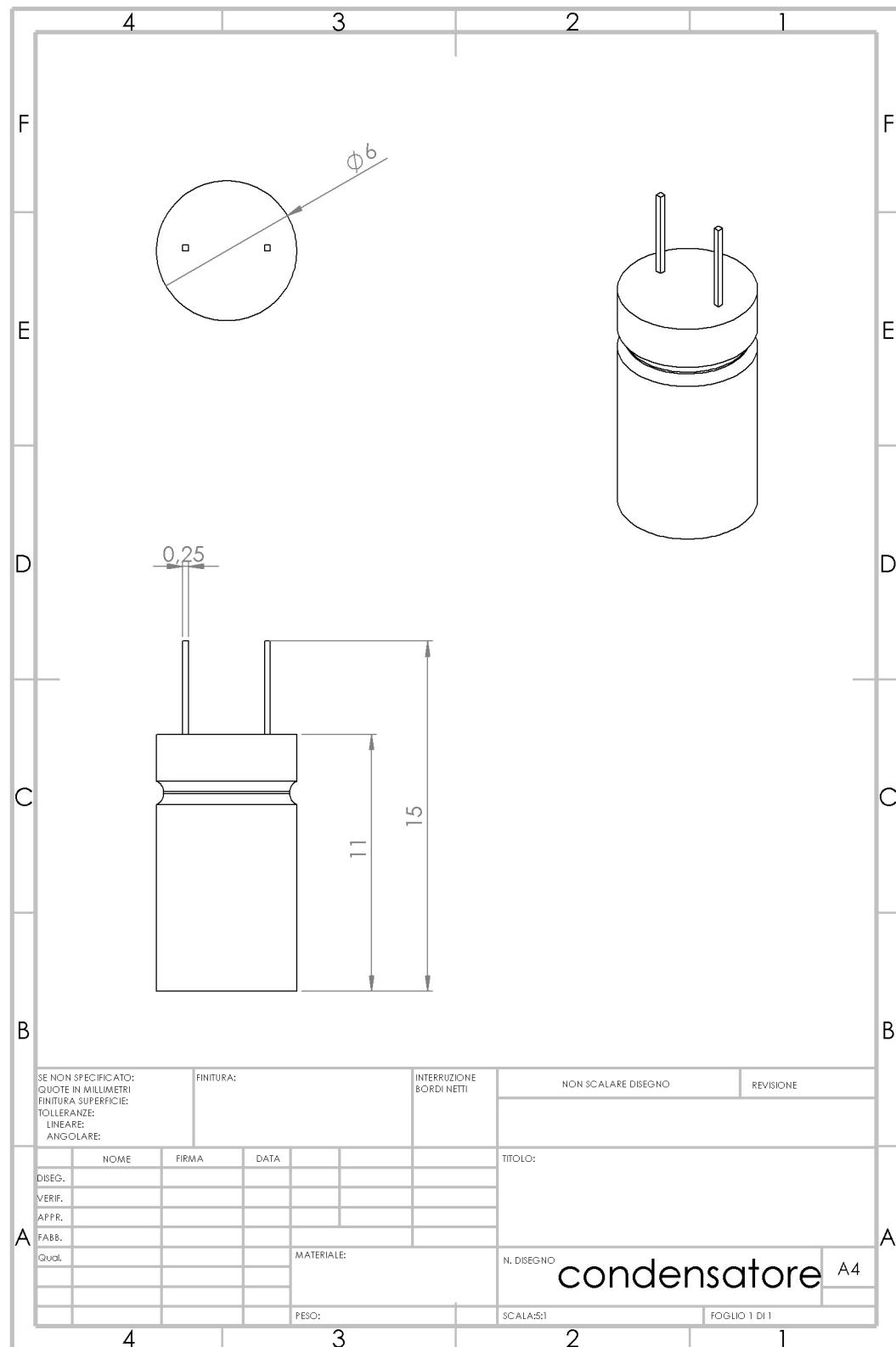
Voltaggio nominale	9V
Temperatura d'utilizzo	-18°C a 55°C

- Breadboard



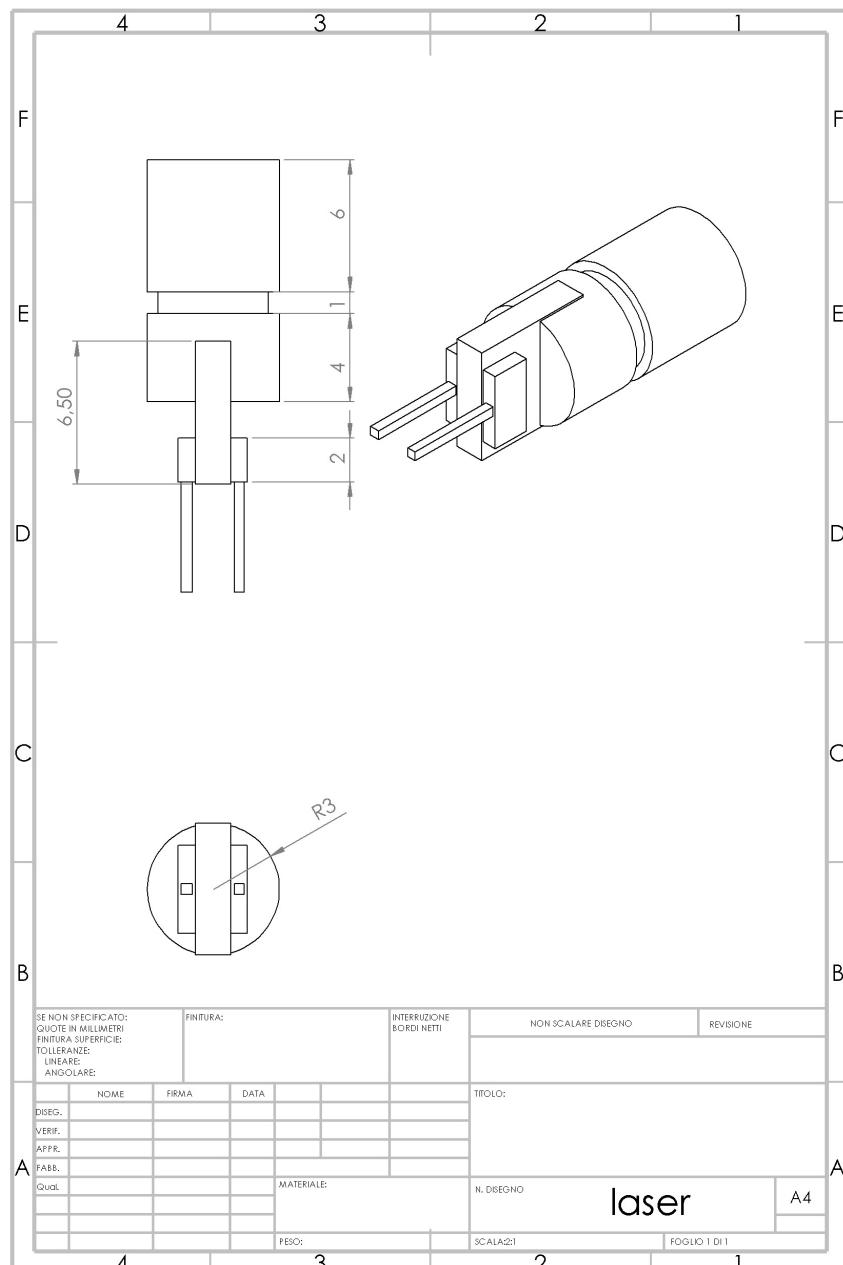
Tensione di alimentazione	5V
Range di funzionamento	300 V
Range di corrente	3-5 A

- Condensatore



Capacità $100 \mu F$

- Laser



Tensione	5 V
Tipo di onda	650nm 5mW
Colore	Rosso
Diametro esterno	6 mm

- Misuratore Laser Disto Classic 5 - LEICA

Range misure	0.2 m a 10 m
Tempo di misurazione	0.5 a 1 s
Accuratezza	$\pm 0.1\text{mm}$

- Resistenze da 220 Ω e 10 $k\Omega$

4 Progettazione case

Inizialmente avevamo in mente di realizzare un modello cad del case del nostro strumento per poi stamparlo con una stampante 3D. Dunque ci siamo subito messi all'opera riproducendo ogni componente con il software Cad «Solid Work 2016» con le misure effettuate nel modo più accurato possibile utilizzando un calibro cinquantesimale. (Nel capitolo precedente sono presenti tutti i disegni quotati realizzati).

Nella progettazione del case ci siamo preoccupati di introdurre nel minor spazio possibile tutti i componenti necessari per lo strumento e disponendoli in modo da evitare eventuali cortocircuiti, dunque isolando bene i contatti. Ci siamo inoltre predisposti lo scopo di rendere il tutto smontabile e quindi poter riutilizzare qualsiasi pezzo e cambiare la batteria qualora fosse necessario. Proprio per questo abbiamo inserito moduli con tolleranza tale da poter avvenire l'incastro e, dove necessario, abbiamo inserito fusi filettati per bloccare i componenti più a rischio.

Una volta terminata questa prima fase abbiamo iniziato a studiare come produrre il case rendendolo il più portatile e stabile possibile. Abbiamo inserito raccordi e due curvature sul fondo per renderlo più ergonomico ed abbiamo inserito su ogni pulsante una scritta con la sua funzione per facilitarne l'utilizzo.



Terminata la lunga progettazione cad abbiamo fatto un assieme per controllare le tolleranze di progetto e verificarne il montaggio.



Dopo aver completato il progetto e coretto i vari incastri con le tolleranze dovute, siamo andati a chiedere per far stampare il pezzo in una stamperia vicino al Politecnico, ma purtroppo il proprietario ci ha subito raggardato sulla difficoltà e l'alto costo di produzione.

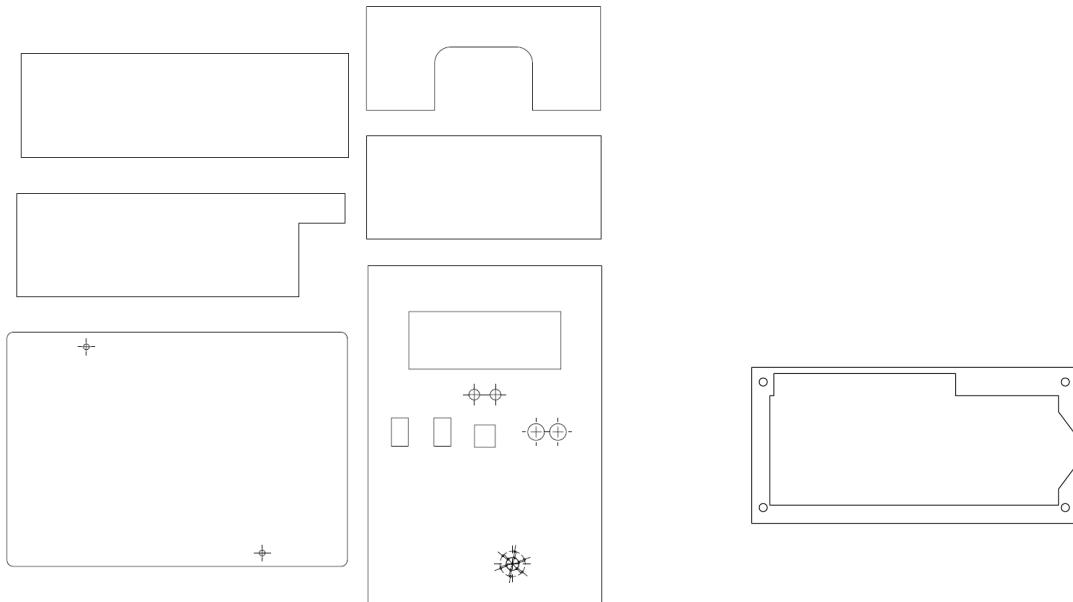
Un po' sconsolati ma mai abbattuti siamo ritornati nel negozio per chiedere che modifiche fossero necessarie per rendere il modello stampabile. Dopo questo colloquio siamo subito accorsi a modificare il modello togliendo gli elementi troppo sottili, le scritte e le due curvature sul fondo ottenendo un oggetto di questo tipo:



Il negoziante una volta visto il nuovo modello ci ha risposto in modo affermativo sulla produzione senza problemi di quest'ultimo e con la promessa di avere il prodotto in meno di una settimana. Dopo una settimana e mezza, chiamiamo per avere informazioni e lui ci dice che la macchina ha avuto un problema per la parte inferiore (la più facile, costituita solo da una piastra ed un parallelepipedo).

Alquanto arrabbiati abbiamo pensato che l'unica alternativa fattibile per la realizzazione del case fosse farlo in compensato, a quel punto il negoziante ci ha rincorato dicendoci che nel suo negozio era a disposizione il taglio laser.

Per la terza volta modifichiamo il modello aumentando ancora la semplicità e vanificando tutti gli sforzi fatti in precedenza. Il modello 3D adesso si trasforma in uno 2D dove spezziamo il modello originale in parti che poi verrano assemblate. È stato quindi mandato in stampa laser un disegno 2D da tagliare su vegetale da 3 mm e compensato da 5 mm.



Ritirati i pezzi abbiamo proceduto al montaggio effettuato con cura con l'ausilio di super attack e colla calda, sostituendo le parti che non possono essere stampate in 3D con una plastica che se scaldata può essere modellata e se immersa in acqua fredda riacquista la sua durezza. Infine dipingiamo il tutto con una vernice colorata. Nella modellazione delle parti mancanti ci poniamo l'obiettivo di rendere l'intera struttura montabile e smontabile, così da poter sostituire facilmente i pezzi in caso di rottura (cosa che si è rivelata poi fondamentale).

5 Analisi tecnica HC-SR04



5.1 Costruzione

- Il sensore HC-SR04 presenta sulla parte frontale due capsule ad ultrasuoni (trasmittente e ricevente) e un oscilloscopio al quarzo da 40 kHz. Il circuito presente su retro è costituito da 3 circuiti integrati:
 - MAX232 che si occupa della conversione del segnale da TIA-232 (RS-232) serial port a TTL-compatible digital logic circuits. (Entrambi sono diversi standard per la trasmissione dei dati)
 - LMC6034, un amplificatore operazionale
 - EM78P153, un microcontrollore di tipo OTP ad 8 bit

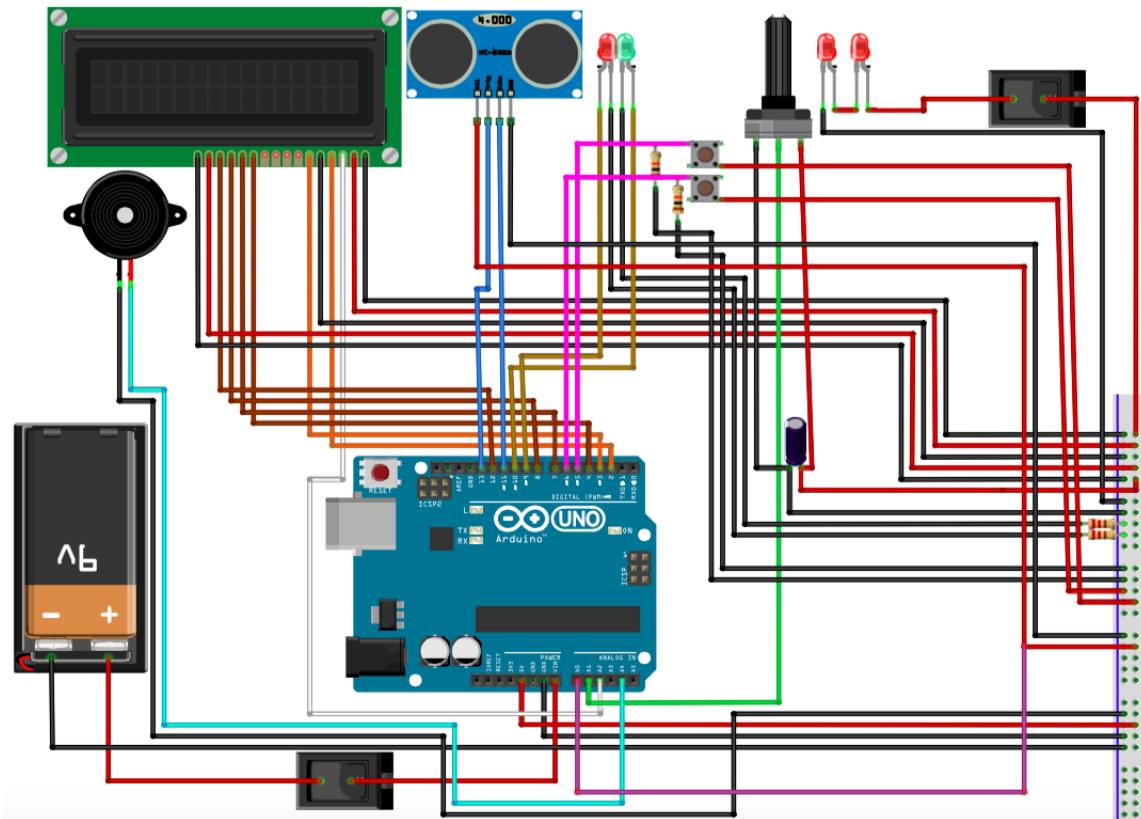
Tensione di alimentazione	DC 5 V
Corrente di alimentazione	15 mA
Frequenza di funzionamento	40 kHz
Trigger Input Signal	10 μ s TTL pulse

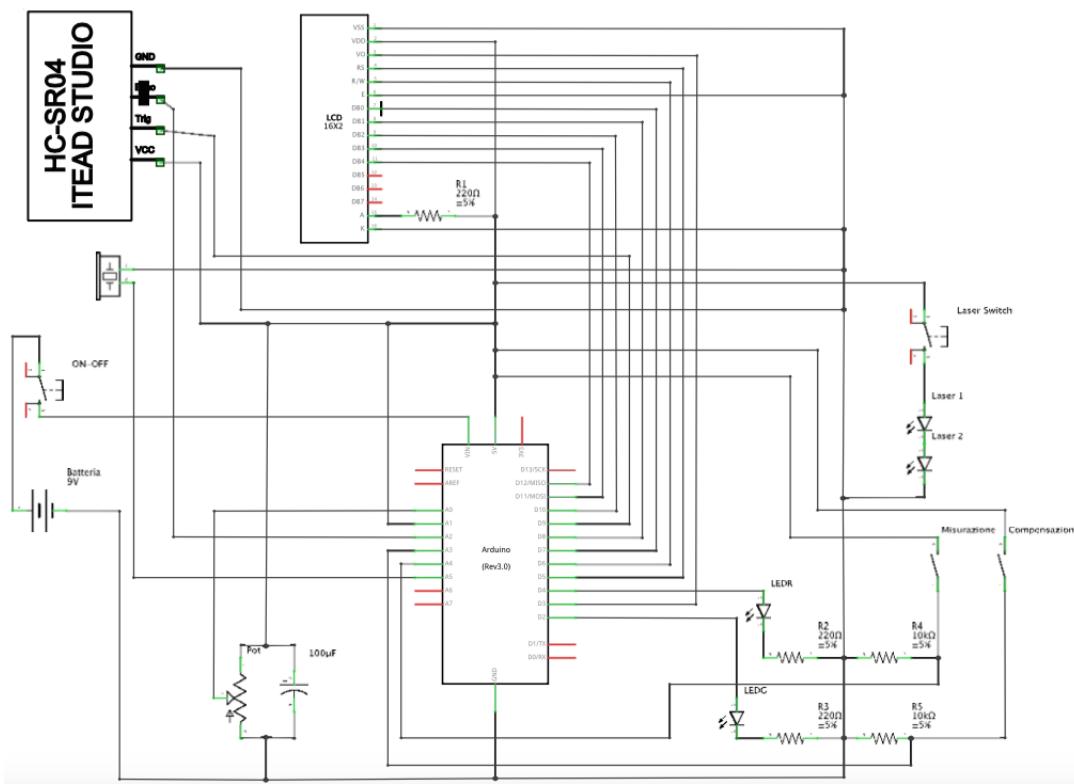
5.2 Funzionamento

- Si tiene il pin Trigger a livello basso, dopodichè si fornisce un impulso a livello alto (*HIGH*) della durata di almeno 10 μ s e lo si riporta a livello basso (*LOW*). In questo momento la capsula trasmittente emette un burst (una sequenza di livelli alti/bassi) a circa 40 kHz. L'onda ultrasonica colpisce il bersaglio, torna indietro e viene rilevata dalla capsula ricevente.
- Il microcontrollore a bordo del sensore calcola il tempo di andata e ritorno del segnale ed emette sul pin Echo, normalmente a livello basso, un segnale a livello alto di durata direttamente proporzionale alla distanza rilevata.
- Per misurare la distanza si ha quindi bisogno soltanto di due I/O di un microcontrollore qualsiasi: uno da settare come uscita (*OUTPUT*) al quale collegare il trigger, e uno da settare come ingresso digitale (*INPUT*) per misurare la lunghezza dell'impulso dell'eco.

6 Progettazione sistema elettrico

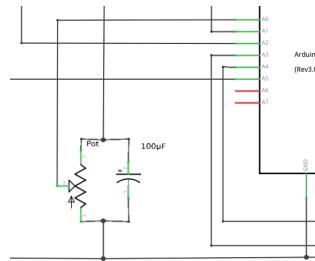
Nella progettazione del sistema elettrico avevamo originariamente scelto una scheda Arduino Nano (non originale), che si è rivelata essere di scarsa qualità, poiché in un brevissimo tempo se ne sono bruciate ben due (e di conseguenza è stato necessario smontare e rimontare l'intero circuito). Dopo l'ennesimo imprevisto, abbiamo deciso di cambiare il controllore, utilizzando una scheda Arduino Uno (originale), che fortunatamente non ha più dato alcun problema. La scheda è alimentata da una batteria da 9V e alimenta tutti gli altri componenti del circuito ad una tensione costante pari a 5V. I due puntatori laser sono collegati in serie a 5V e azionati da un interruttore dedicato. Per quanto riguarda il display LCD si è deciso di dedicare quattro degli otto canali disponibili per la trasmissione dei dati (poiché più che sufficienti). Qui di seguito è possibile vedere un modello del cablaggio all'interno del dispositivo e lo schema elettrico dello stesso:



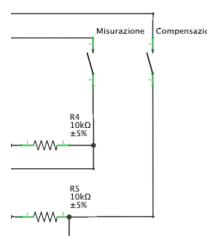


Le principali problematiche relative alla progettazione del circuito sono:

- Il corretto isolamento dell'intero sistema, per evitare cortocircuiti e l'errato funzionamento del dispositivo.
- Il posizionamento di un condensatore di $100\mu F$ messo in parallelo al potenziometro, per evitare che il repentino cambiamento della resistenza (dovuto alla rotazione della manopola del potenziometro), crei un calo di tensione che possa danneggiare la scheda:

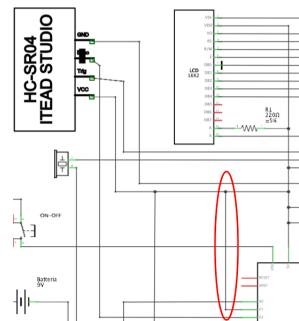


3. L'inserimento di due resistenze da $10k\Omega$ (una per ogni pulsante). L'alto valore di resistenza è motivato dal fatto che sono due resistenze di «pull-down», che servono per mettere a massa il cavo che verifica se è stato schiacciato o meno il pulsante (legge 0 se non è premuto e 1 nel caso in cui venga schiacciato) quando l'interruttore è aperto, evitando che un eventuale rumore di fondo falsifichi la lettura.

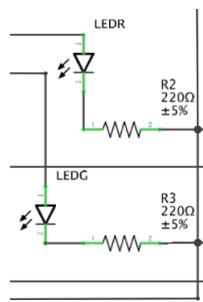


4. Il controllo che batteria, scheda e tutti gli elementi del circuito siano collegati alla stessa massa.

5. Il cavo che misura la tensione di alimentazione del sensore HC-SR04, posto perciò in parallelo a quest'ultimo:



6. Ognuno dei due led ha in serie una resistenza di 220Ω per evitare che si brucino:



7 Taratura dello strumento

Quando abbiamo avuto l'idea di creare lo strumento di misura abbiamo effettuato una taratura di prova del sensore HC-SR04 per testarne le capacità ed abbiamo notato una risposta lineare ad una distanza di 4 m, ma la presenza di oggetti all'interno di un raggio d'azione andava ad interferire con la sua misura. Altro problema a cui abbiamo pensato è stato che la tensione di alimentazione potesse influire sulla misura.

Dopo le prime ipotesi preliminari, la costruzione dello strumento e la progettazione del circuito elettrico, ci siamo cimentati nella taratura dello strumento. Abbiamo deciso di effettuare una taratura back-to-back con un misuratore laser «Disto classic 5» che ha una risoluzione di 0,1 mm .

Per effettuare la taratura ci siamo posizionati in un grande spazio privo di ostacoli e abbiamo utilizzato un'ampia superficie come riferimento per la misurazione. Abbiamo iniziato a misurare da una distanza di 20 cm poichè questa è la distanza di stand-off del misuratore laser.

Abbiamo deciso di misurare ad intervalli regolari una volta in andata ed una in ritorno.

7.1 Programmazione codice di taratura

7.1.1 Codice Arduino per la taratura

Per la taratura abbiamo scritto il seguente codice sul micro-controllore arduino:

Taratura strumento

```

1 //Pin//
2 const long trigger = 10;
3 const long echo = 11;
4 const long voltmetro = A0;
5 const long misuratore = A1;
6
7 void setup() {
8     pinMode(trigger, OUTPUT);
9     pinMode(echo, INPUT);
10    pinMode(voltmetro, OUTPUT);
11    Serial.begin(9600);
12 }
13
14 long times = 0;
15
16 void loop() {
17     if(times < 1){
18
19         for(long m = 0; m < 1; m++){
20
21             Serial.print("Tensione: ");
22             int valore = analogRead(A1);
23             int tensione = map(valore, 0, 1023, 0, 5);
24             Serial.println(tensione);
25             long array[14];
26
27             for (long i = 0; i < 14; i++){} //Creazione e inizializzazione array di 14 //
28                                         //misure quello da 12 e da 10//
29             array[i] = 0;
30         }
31
32         long array12[12];
33
34         for(long s = 0; s < 12; s++){
35             array12[s] = 0;
36         }
37
38         long array10[10];
39
40         for(long t = 0; t < 10; t++){
41
42             if(t == 0){ //Inizio misure
43                 Serial.print("Inizio misure");
44                 Serial.print(" Tensione: ");
45                 Serial.print(analogRead(A1));
46                 Serial.print(" ");
47                 Serial.print(" Distanza: ");
48                 Serial.print(distanza);
49                 Serial.print(" ");
50                 Serial.print(" Tensione: ");
51                 Serial.print(analogRead(A1));
52                 Serial.print(" ");
53                 Serial.print(" Distanza: ");
54                 Serial.print(distanza);
55                 Serial.print(" ");
56                 Serial.print(" Tensione: ");
57                 Serial.print(analogRead(A1));
58                 Serial.print(" ");
59                 Serial.print(" Distanza: ");
60                 Serial.print(distanza);
61                 Serial.print(" ");
62                 Serial.print(" Tensione: ");
63                 Serial.print(analogRead(A1));
64                 Serial.print(" ");
65                 Serial.print(" Distanza: ");
66                 Serial.print(distanza);
67                 Serial.print(" ");
68                 Serial.print(" Tensione: ");
69                 Serial.print(analogRead(A1));
70                 Serial.print(" ");
71                 Serial.print(" Distanza: ");
72                 Serial.print(distanza);
73                 Serial.print(" ");
74                 Serial.print(" Tensione: ");
75                 Serial.print(analogRead(A1));
76                 Serial.print(" ");
77                 Serial.print(" Distanza: ");
78                 Serial.print(distanza);
79                 Serial.print(" ");
80                 Serial.print(" Tensione: ");
81                 Serial.print(analogRead(A1));
82                 Serial.print(" ");
83                 Serial.print(" Distanza: ");
84                 Serial.print(distanza);
85                 Serial.print(" ");
86                 Serial.print(" Tensione: ");
87                 Serial.print(analogRead(A1));
88                 Serial.print(" ");
89                 Serial.print(" Distanza: ");
90                 Serial.print(distanza);
91                 Serial.print(" ");
92                 Serial.print(" Tensione: ");
93                 Serial.print(analogRead(A1));
94                 Serial.print(" ");
95                 Serial.print(" Distanza: ");
96                 Serial.print(distanza);
97                 Serial.print(" ");
98                 Serial.print(" Tensione: ");
99                 Serial.print(analogRead(A1));
100                Serial.print(" ");
101               Serial.print(" Distanza: ");
102               Serial.print(distanza);
103
104             if(t == 1){ //Fine misure
105                 Serial.print("Fine misure");
106             }
107         }
108     }
109 }
110
111 long distanza;
112
113 void setup() {
114     Serial.begin(9600);
115 }
116
117 void loop() {
118 }
```

```
41     array10[t] = 0;
42 }
43
44 long cont = 0;
45 long durata = 0;
46
47 for (long j = 0; j < 15; j++){
48
49     do {
50         digitalWrite(trigger, LOW);
51         delayMicroseconds(2);
52         digitalWrite(trigger, HIGH);
53         delayMicroseconds(10);
54         digitalWrite(trigger, LOW);
55         durata = pulseIn(echo, HIGH);
56     } while (durata == 0);
57
58     if(cont == 0){
59         cont = 1;
60     } else {
61         array[j-1] = durata;          //Si fanno 15 misurazioni ma se ne salvano //
62                                //solo 14 (scartando la prima)//
63     }
64
65     delay(500);
66 }
67
68     long massimo = array[0];
69     long minimo = array[0];
70
71     for(long x = 0; x < 14; x++){           //Si trovano massimo e minimo//
72                                //nell'array di 14//
73         massimo = max(massimo, array[x]);
74         minimo = min(minimo, array[x]);
75     }
76
77     long posMax = 0;
78     long posMin = 0;
79
80     for(long y = 1; y < 14; y++){           //Si determinano le posizioni del//
81                                //massimo e del minimo nell'array14//
82         if (array[y] == massimo){
83             posMax = y;
84         } else if (array[y] == minimo){
85             posMin = y;
86         }
87     }
88
89     long prev = 0;
90     long contMax = 0;
91     long contMin = 0;
92
93     for(long q = 0; q < 14; q++){           //Si riempie l'array 12 //
94
95         if(((array[q] != massimo) && (array[q] != minimo)) ||
96             ((contMax ==1) && array[q] == massimo) || ((contMin==1)&& array[q] ==minimo)){
97             array12[prev] = array[q];
98             prev++;
99         } else if (array[q] == massimo) {
100             contMax = 1;
101         } else if (array[q] == minimo) {
```

```

103         contMin = 1;
104     }
105 }
106
107 long massimo2 = array12[0];
108 long minimo2 = array12[0];
109
110 for(long x = 1; x < 12; x++){ //Si trovano massimo e minimo nell'array di 12//
111     massimo2 = max(massimo2, array12[x]);
112     minimo2 = min(minimo2, array12[x]);
113 }
114
115 long posMax2 = 0;
116 long posMin2 = 0;
117
118 for(long z = 0; z < 12; z++){ //Si determinano le posizioni del massimo e //
119                                         //del minimo nell'array12//
120     if (array12[z] == massimo2){
121         posMax2 = z;
122     } else if (array12[z] == minimo2){
123         posMin2 = z;
124     }
125 }
126
127 long prev2 = 0;
128 long contMax2 = 0;
129 long contMin2 = 0;
130 for(long k = 0; k < 12; k++){ //Si riempie l'array di 10//
131
132     if(((array12[k] != massimo2) && (array12[k] != minimo2)) ||
133         ((contMax2 == 1) && (array12[k] == massimo2)) || ((contMin2 == 1)&&
134         (array12[k] == minimo2)) ){
135         array10[prev2] = array12[k];
136         prev2++;
137     }
138
139     else if (array12[k] == massimo2) {
140         contMax2 = 1;
141     } else if (array12[k] == minimo2) {
142         contMin2 = 1;
143     }
144
145     long somma = 0;
146     long media = 0;
147     Serial.println("Distanze: ");
148
149     for(long c = 0; c < 10; c++){
150         Serial.println(array10[c]);
151         somma = somma + array10[c];
152     }
153     Serial.println(" ");
154     media = somma/10;
155     Serial.println("Media: ");
156     Serial.println(media);
157
158     }
159 }
160 times++;
161 }
162 }
163 }
```

7.1.2 Spiegazione codice

Attraverso un ciclo *for* il circuito esegue un totale di 15 misurazioni.

Come possiamo vedere per effettuare la misura è necessario prima di tutto tenere il pin «trigger» a livello *LOW*, successivamente dare un impulso di $10 \mu s$ e riportare il «trigger» a *LOW*. Il valore di $10 \mu s$ non è stato scelto a caso, ci siamo infatti preoccupati di cercare il datasheet del trasduttore, dove era consigliato di utilizzare questo valore come trigger input. Infine è presente la funzione *pulseIn()* che legge un valore di tipo basso o alto su un pin. In questo caso legge un valore sul pin «echo» di un segnale a livello alto di durata direttamente proporzionale alla distanza rilevata e questo segnale viene salvato in una variabile «durata» che verrà riscritta ogni volta dopo aver salvato il suo contenuto nell'array delle misure.

```

50  digitalWrite(trigger, LOW);
51  delayMicroseconds(2);
52  digitalWrite(trigger, HIGH);
53  delayMicroseconds(10);
54  digitalWrite(trigger, LOW);
      durata = pulseIn(echo, HIGH);

```

Questa serie di misure viene fatta con un intervallo di $50 ms$ in quanto, dopo una preliminare analisi di come effettuare la taratura, abbiamo registrato come valore massimo di misura utilizzabile $35381 \mu s$ dunque, il valore scelto, permette di far tornare il segnale e farne partire uno nuovo senza che questi si accavallino e vadano in interferenza modificando i valori della misura.

La prima viene scartata poichè abbiamo notato che la prima misurazione risulta affetta da errori e spesso compare il valore 0. Purtroppo non siamo riusciti a capire il perchè di questo, ma abbiamo voluto comunque eliminare questo dato che risulta oggettivamente errato.

Per evitare la presenza di valori pari a zero, frutto di qualche errore di tipo sistematico o casuale, abbiamo inserito il codice di misurazione all'interno di un ciclo *do while* che ripete la misura fino a quando il valore *durata* è uguale a zero.

Salvate le 14 misure attraverso un passaggio ad un nuovo array di dimensioni minori (poichè in C, a differenza di Matlab, non ci sono array di tipo dinamico) si passa all'eliminazione di due massimi e di due minimi. Questa scelta è stata dettata dal fatto che abbiamo notato, dopo una serie di prove, come il sensore fosse sensibile alla presenza di oggetti estranei o sporgenze lungo il suo raggio d'azione. Emettendo un segnale ad ultrasuoni è possibile che una di queste onde vada a colpire un oggetto non voluto e che se questo è inclinato nel giusto modo, possa ritornare allo strumento fornendo una misura errata; dunque togliendo questi 4 valori abbiamo cercato di ridurre il più possibile gli errori sistematici e cercando di migliorare la media della misura restituita dallo strumento.

Infine vengono stampati i valori a schermo per poterli copiare e trasferire facilmente su Excel (abbiamo provato a collegare Arduino a Matlab ma dopo una serie di tentativi vani ci siamo rassegnati ad utilizzare il metodo appena spiegato).

Abbiamo eseguito questa operazione di taratura tenendo sempre sott'occhio la tensione di alimentazione, importante poichè il nostro scopo è verificare se la risposta del nostro sensore vari o meno al variare dell'alimentazione. Per far questo abbiamo collegato un pin analogico direttamente al cavo di alimentazione del sensore, in modo da essere certi di leggere un valore corretto. Proprio per questo ci siamo preoccupati di fare la taratura solo dopo l'ultimazione del progetto e con tutti i componenti alimentati, così da verificare l'effettiva alimentazione entrante nel sensore.

```

22  Serial.print("Tensione: ");
23  int valore = analogRead(A1);
24  int tensione = map(valore, 0, 1023, 0, 5);
      Serial.println(tensione);

```

Per mostrare i valori di tensione in Volt, abbiamo mappato il valore registrato da Arduino nella porta analogica (legge valori da 0 a 1023) con valori in Volt da 0 a 5 (massima alimentazione che Arduino può fornire, e tensione a cui il trasduttore deve essere alimentato, come spiegato nel datasheet). In questo modo ad ogni misura abbiamo potuto vedere il valore effettivo in Volt che arrivava al sensore. Poichè abbiamo notato che per un valore di $4.5 V$ il sensore non funzionava più, abbiamo deciso di effettuare una seconda taratura ad una tensione di $4.6 V$, per studiarne l'andamento ed adattarne il diagramma di taratura.

7.1.3 Taratura angolo

Come detto in precedenza, ci siamo accorti che nel caso in cui vi sia un oggetto (o una sporgenza) all'interno di una determinata area (compresa tra il sensore e l'oggetto di cui si vuole misurare la distanza), le misurazioni perdevano di significato o andavano in contrasto tra di loro. Ci siamo posti quindi l'obiettivo di determinare un angolo, che potesse garantire il più possibile il corretto funzionamento dello strumento di misura, per far sì che fornisse una misura attendibile. Per fare questo abbiamo posizionato un cartone ($100 \times 150\text{ cm}$) ad una determinata distanza, procedendo poi alla misurazione della distanza, ottenendo il vettore con le 10 misurazioni e calcolandone il massimo e il minimo (*ValoreMinimo* e *ValoreMassimo*) con il seguente codice:

Taratura angolo

```
//Pin//
2 const long trigger = 10;
3 const long echo = 11;
4 const long voltmetro = A0;
5 const long misuratore = A1;
6 const long ValoreMinimo=11950;
7 const long ValoreMassimo= 12050;
8
9 //Parametri//
10 long valoriTensione[3] = {1023, 800, 500};
11
12 void setup() {
13     pinMode(trigger, OUTPUT);
14     pinMode(echo, INPUT);
15     pinMode(voltmetro, OUTPUT);
16     Serial.begin(9600);
17     pinMode(A4, OUTPUT);
18 }
19
20 long times = 0;
21
22 void loop() {
23     for(long m = 0; m < 1; m++){
24         Serial.print("Tensione: ");
25         int valore = analogRead(A1);
26         int tensione = map(valore, 0, 1023, 0, 6);
27         Serial.println(tensione);
28         analogWrite(voltmetro, valoriTensione[m]);
29         long array[14];
30         for (long i = 0; i < 14; i++){
31             array[i] = 0;
32         }
33         long array12[12];
34         for(long s = 0; s < 12; s++){
35             array12[s] = 0;
36         }
37         long array10[10];
38         for(long t = 0; t < 10; t++){
39             array10[t] = 0;
40         }
41         long cont = 0;
42         long durata = 0;
43         for (long j = 0; j < 15; j++){
44             do {
45                 digitalWrite(trigger, LOW);
46                 delayMicroseconds(2);
47                 digitalWrite(trigger, HIGH);
48                 delayMicroseconds(10);
49                 digitalWrite(trigger, LOW);
50                 durata = pulseIn(echo, HIGH);
```

```

    } while (durata == 0);
52   if(cont == 0){
      cont = 1;
54   } else {
      array[j-1] = durata;
      }
      delay(50);
58 }
long massimo = array[0];
60 long minimo = array[0];
for(long x = 0; x < 14; x++){
  massimo = max(massimo, array[x]);
  minimo = min(minimo, array[x]);
}
64 long posMax = 0;
66 long posMin = 0;
for(long y = 1; y < 14; y++){
  if (array[y] == massimo){
    posMax = y;
  } else if (array[y] == minimo){
    posMin = y;
  }
}
72 }
74 long prev = 0;
76 long contMax = 0;
78 long contMin = 0;
for(long q = 0; q < 14; q++){
  if(((array[q] != massimo) && (array[q] != minimo))
     || ((contMax ==1) && array[q] == massimo)
     || ((contMin==1)&& array[q] ==minimo)){
    array12[prev] = array[q];
    prev++;
  } else if (array[q] == massimo) {
    contMax = 1;
  } else if (array[q] == minimo) {
    contMin = 1;
  }
}
88 long massimo2 = array12[0];
90 long minimo2 = array12[0];
92 for(long x = 1; x < 12; x++){
  massimo2 = max(massimo2, array12[x]);
  minimo2 = min(minimo2, array12[x]);
}
94 }
96 long posMax2 = 0;
98 long posMin2 = 0;
for(long z = 0; z < 12; z++){
  if (array12[z] == massimo2){
    posMax2 = z;
  } else if (array12[z] == minimo2){
    posMin2 = z;
  }
}
104 long prev2 = 0;
106 long contMax2 = 0;
108 long contMin2 = 0;
for(long k = 0; k < 12; k++){
  if(((array12[k] != massimo2) && (array12[k] != minimo2))
     || ((contMax2 ==1) && (array12[k] == massimo2))
     || ((contMin2 == 1)&& (array12[k] == minimo2)) ){
    array10[prev2] = array12[k];
    prev2++;
  }
}

```

```

114 }
115     else if (array12[k] == massimo2) {
116         contMax2 = 1;
117     } else if (array12[k] == minimo2) {
118         contMin2 = 1;
119     }
120     long somma = 0;
121     long media = 0;
122     Serial.println("Distanze: ");
123     for(long c = 0; c < 10; c++){
124         Serial.println(array10[c]);
125         somma = somma+ array10[c];
126     }
127     Serial.println(" ");
128     media = somma/10;
129     delay(20);
130     if (media<=ValoreMinimo || media>= ValoreMassimo){
131         tone(A4, 80, 100);
132         delay(50);
133     }
134 }
```

Come si può notare, l'algoritmo che calcola la distanza è sempre lo stesso, tuttavia in questo caso il dispositivo continuava ad effettuare delle misurazioni, le cui medie venivano poi confrontate con il *ValoreMinimo* e il *ValoreMassimo* precedentemente calcolati; alla fine è stato aggiunta una parte di codice che permettesse di avere un feedback sonoro, nel caso in cui la misura in uscita non fosse compresa tra il massimo e il minimo.

```

130 if (media<=ValoreMinimo || media>= ValoreMassimo){
131     tone(A4, 80, 100);
132     delay(50);
133 }
134 }
```

Abbiamo sfruttato questo algoritmo per avvicinarci lateralmente con una lamina di dimensioni inferiori ($50 \times 50\text{ cm}$), segnando per terra il punto in cui lo strumento rilevava l'uscita della misurazione dal range di tolleranza:



Una volta ottenuto un quantitativo soddisfacente di punti, abbiamo "interpolato" con una retta, posizionando due strisce di scotch, misurando l'angolo tra le due (30°):



Ci siamo infine accorti che la misurazione veniva influenzata non solo dall'ingresso o meno di un oggetto nell'area descritta dall'angolo di misura, ma anche dall'inclinazione della superficie di tale oggetto, per questo motivo si è deciso di aggiungere nella parte di codice che effettua la misura, un controllo della compatibilità dell'incertezza della misurazione reale con quella calcolata nella fase di taratura (condizione ideale, con una superficie di grandi dimensioni, ortogonale allo strumento).

7.2 Elaborazione dati

Acquisiti 10 dati manipolabili per ogni distanza più il riferimento del trasduttore laser, abbiamo inserito tutti i dati in un foglio Excel. Per ognuna delle 10 misurazioni abbiamo calcolato: media, massimo, minimo, deviazione standard ed incertezza per capire, una volta effettuata la misura, se affetta da errori strumentali e/o compatibile con le precedenti. Grazie a questo approccio è anche stato possibile comprendere a che distanza terminare la misura. Abbiamo notato che all'aumentare della distanza si verifica un aumento dell'incertezza delle singole misure. Dato che ci aspettavamo e che avevamo messo in conto. Qui di seguito sono riportati tutti i dati acquisiti durante l'operazione di taratura.

	ANDATA		5 V		ANDATA		5 V
Laser [mm]	HC-SR04 μS			Laser [mm]	HC-SR04 μS		
201	1164	Max	1164,00	1328	7615	Max	7615,00
	1164	min	1164,00		7582	min	7581,00
	1164	media	1164,00		7615	media	7603,80
	1164	dev. Sta	0,00		7609	dev. Sta	13,94
	1164	incertezza	0,00		7615	incertezza	4,41
	1164	supeiore			7590		
	1164				7608		
	1164				7615		
	1164				7581		
	1164				7608		
393	2266	Max	2267,00	1554	8883	Max	8936,00
	2267	min	2266,00		8884	min	8881,00
	2267	media	2266,90		8891	media	8890,30
	2267	dev. Sta	0,32		8882	dev. Sta	16,43
	2267	incertezza	0,10		8881	incertezza	5,19
	2267				8891		
	2267				8936		
	2267				8884		
	2267				8887		
	2267				8884		
575	3294	Max	3294,00	1788	10253	Max	10254,00
	3294	min	3288,00		10254	min	10226,00
	3294	media	3293,40		10253	media	10243,20
	3294	dev. Sta	1,90		10254	dev. Sta	13,33
	3294	incertezza	0,60		10253	incertezza	4,22
	3294				10254		
	3288				10229		
	3294				10229		
	3294				10227		
	3294				10226		
846	4864	Max	4864,00	2024	11578	Max	11581,00
	4863	min	4863,00		11580	min	11574,00
	4863	media	4863,20		11577	media	11578,20
	4863	dev. Sta	0,42		11577	dev. Sta	2,39
	4863	incertezza	0,13		11581	incertezza	0,76
	4863				11580		
	4863				11580		
	4863				11575		
	4864				11580		
	4863				11574		

ANDATA				5 V		ANDATA		5 V
Laser [mm]	HC-SR04 μS				Laser [mm]	HC-SR04 μS		
1085	6218	Max	6218,00		2256	12956	Max	12980,00
	6218	min	6194,00			12955	min	12929,00
	6218	media	6215,00			12934	media	12956,00
	6218	dev. Sta	7,62			12929	dev. Sta	16,31
	6218	incertezza	2,41			12959	incertezza	5,16
	6194					12980		
	6212					12959		
	6218					12955		
	6218					12980		
	6218					12953		
2482	14278	Max	14280,00		3535	20300	Max	20327,00
	14253	min	14252,00			20299	min	20276,00
	14256	media	14261,89			20301	media	20302,90
	14253	dev. Sta	12,64			20299	dev. Sta	14,73
	14253	incertezza	4,00			20300	incertezza	4,66
	14254					20327		
	14252					20300		
	14278					20300		
	14280					20327		
	14280					20276		
2724	15633	Max	15666,00		3718	21316	Max	21358,00
	15666	min	15614,00			21332	min	21316,00
	15666	media	15639,90			21332	media	21340,70
	15633	dev. Sta	15,58			21333	dev. Sta	15,66
	15639	incertezza	4,93			21331	incertezza	4,95
	15633					21358		
	15639					21358		
	15638					21358		
	15614					21331		
	15638					21358		
2945	16893	Max	16893,00		4065	23543	Max	23609,00
	16874	min	16867,00			23592	min	23518,00
	16867	media	16876,80			23609	media	23565,50
	16873	dev. Sta	11,01			23540	dev. Sta	32,02
	16872	incertezza	3,48			23539	incertezza	10,13
	16867					23584		
	16867					23594		
	16892					23542		
	16871					23594		
	16892					23518		
3115	17876	Max	17876,00		4484	26083	Max	26109,00
	17855	min	17848,00			26084	min	25988,00
	17853	media	17855,20			26109	media	26043,40
	17853	dev. Sta	8,02			26043	dev. Sta	40,08
	17856	incertezza	2,54			26012	incertezza	12,67
	17849					26020		
	17858					25988		
	17848					26019		
	17855					26067		
	17849					26009		

	ANDATA		5 V
Laser [mm]	HC-SR04 μS		
3261	18707	Max	18715,00
	18685	min	18685,00
	18690	media	18693,70
	18685	dev. Sta	11,33
	18686	incertezza	3,58
	18686		
	18707		
	18688		
	18688		
	18715		
5286	30660	Max	30660,00
	23902	min	23902,00
	30535	media	29928,60
	30654	dev. Sta	2117,87
	30584	incertezza	669,73
	30586		
	30587		
	30609		
	30560		
	30609		
6098	35384	Max	35387,00
	35386	min	35316,00
	35363	media	35358,30
	35367	dev. Sta	27,03
	35334	incertezza	8,55
	35316		
	35333		
	35332		
	35387		
	35381		

	RITORNO		5 V		RITORNO		5 V
Laser [mm]	HC-SR04 μS			Laser [mm]	HC-SR04 μS		
6099	35808	Max	35809,00	3530	20431	Max	20482,00
	35785	min	35393,00		20431	min	20408,00
	35809	media	35723,30		20408	media	20427,10
	35736	dev. Sta	123,75		20482	dev. Sta	22,36
	35808	incertezza	39,13		20427	incertezza	7,07
	35393				20433		
	35760				20408		
	35711				20409		
	35686				20433		
	35737				20409		
5287	30693	Max	30716,00	3263	18907	Max	18937,00
	30692	min	30664,00		18888	min	18856,00
	30665	media	30692,30		18862	media	18891,40
	30669	dev. Sta	20,57		18856	dev. Sta	28,61
	30664	incertezza	6,50		18860	incertezza	9,05
	30716				18880		
	30713				18887		
	30708				18906		
	30689				18931		
	30714				18937		
4482	23891	Max	23982,00	3111	17925	Max	17944,00
	23885	min	23832,00		17943	min	17918,00
	23865	media	23894,20		17918	media	17928,70
	23859	dev. Sta	45,48		17944	dev. Sta	10,58
	23891	incertezza	14,38		17925	incertezza	3,35
	23832				17919		
	23889				17922		
	23885				17944		
	23963				17924		
	23982				17923		
4061	23562	Max	23586,00	2942	16921	Max	16971,00
	23538	min	23528,00		16926	min	16920,00
	23562	media	23552,80		16970	media	16942,40
	23537	dev. Sta	21,42		16954	dev. Sta	18,97
	23561	incertezza	6,77		16945	incertezza	6,00
	23534				16945		
	23528				16926		
	23535				16946		
	23585				16971		
	23586				16920		
3718	21518	Max	21544,00	2720	15619	Max	15650,00
	21519	min	21492,00		15620	min	15617,00
	21493	media	21513,80		15617	media	15627,00
	21492	dev. Sta	16,48		15623	dev. Sta	11,34
	21521	incertezza	5,21		15638	incertezza	3,59
	21520				15621		
	21493				15650		
	21544				15622		
	21518				15640		
	21520				15620		

	RITORNO		5 V		RITORNO		5 V
Laser [mm]	HC-SR04 μS			Laser [mm]	HC-SR04 μS		
2481	14258	Max	14283,00	1325	7568	Max	7574,00
	14282	min	14255,00		7567	min	7546,00
	14283	media	14264,60		7567	media	7565,50
	14256	dev. Sta	12,53		7546	dev. Sta	10,42
	14283	incertezza	3,96		7567	incertezza	3,29
	14255				7573		
	14256				7547		
	14258				7574		
	14256				7573		
	14259				7573		
2254	12961	Max	12961,00	1082	6209	Max	6209,00
	12942	min	12935,00		6209	min	6182,00
	12936	media	12940,70		6182	media	6201,20
	12942	dev. Sta	7,60		6209	dev. Sta	12,57
	12936	incertezza	2,40		6183	incertezza	3,97
	12936				6209		
	12939				6184		
	12941				6209		
	12935				6209		
	12939				6209		
2017	11570	Max	11570,00	842	4848	Max	4854,00
	11544	min	11542,00		4848	min	4848,00
	11570	media	11558,00		4848	media	4848,60
	11543	dev. Sta	13,12		4848	dev. Sta	1,90
	11543	incertezza	4,15		4848	incertezza	0,60
	11564				4848		
	11570				4848		
	11570				4848		
	11564				4848		
	11542				4854		
1780	10178	Max	10183,00	570	3265	Max	3265,00
	10183	min	10178,00		3265	min	3265,00
	10180	media	10179,60		3265	media	3265,00
	10180	dev. Sta	1,71		3265	dev. Sta	0,00
	10178	incertezza	0,54		3265	incertezza	0,00
	10181				3265		
	10181				3265		
	10178				3265		
	10178				3265		
	10179				3265		
1553	8877	Max	8884,00	385	2225	Max	2226,00
	8884	min	8877,00		2220	min	2220,00
	8877	media	8879,10		2225	media	2224,70
	8877	dev. Sta	2,96		2225	dev. Sta	1,70
	8877	incertezza	0,94		2226	incertezza	0,54
	8877				2225		
	8883				2225		
	8883				2225		
	8878				2226		
	8878				2225		

	RITORNO		5 V
Laser [mm]	HC-SR04 μS		
204	1179	Max	1180,00
	1179	min	1179,00
	1179	media	1179,30
	1180	dev. Sta	0,48
	1179	incertezza	0,15
	1179		
	1179		
	1180		
	1180		
	1179		

	ANDATA		4.6 V		ANDATA		4.6 V	
Laser [mm]	HC-SR04 μS				Laser [mm]	HC-SR04 μS		
205	1158	Max	1158,00		1328	7587	Max	7587,00
	1158	min	1158,00			7587	min	7559,00
	1158	media	1158,00			7560	media	7576,60
	1158	dev. Sta	0,00			7581	dev. Sta	12,00
	1158	incertezza	0,00			7580	incertezza	3,80
	1158					7586		
	1158					7586		
	1158					7580		
	1158					7559		
	1158					7560		
388	2234	Max	2235,00		1556	8916	Max	8916,00
	2234	min	2229,00			8916	min	8895,00
	2234	media	2233,90			8915	media	8911,30
	2234	dev. Sta	1,79			8895	dev. Sta	8,60
	2234	incertezza	0,57			8916	incertezza	2,72
	2229					8915		
	2235					8915		
	2235					8915		
	2235					8895		
	2235					8915		
568	3245	Max	3245,00		1784	10228	Max	10256,00
	3245	min	3239,00			10255	min	10202,00
	3245	media	3244,40			10256	media	10229,80
	3245	dev. Sta	1,90			10255	dev. Sta	23,98
	3245	incertezza	0,60			10231	incertezza	7,58
	3245					10204		
	3245					10255		
	3245					10204		
	3245					10202		
	3239					10208		
842	4843	Max	4891,00		2019	11551	Max	11578,00
	4843	min	4843,00			11545	min	11545,00
	4849	media	4867,60			11551	media	11565,40
	4891	dev. Sta	21,99			11571	dev. Sta	13,43
	4867	incertezza	6,95			11578	incertezza	4,25
	4867					11554		
	4891					11577		
	4891					11577		
	4843					11572		
	4891					11578		
1087	6225	Max	6226,00		2255	12924	Max	12976,00
	6225	min	6198,00			12923	min	12922,00
	6199	media	6217,40			12924	media	12939,70
	6225	dev. Sta	12,49			12922	dev. Sta	22,57
	6225	incertezza	3,95			12976	incertezza	7,14
	6225					12955		
	6198					12976		
	6225					12924		
	6201					12950		
	6226					12923		

	ANDATA		4.6 V		ANDATA		4.6 V
Laser [mm]	HC-SR04 μS			Laser [mm]	HC-SR04 μS		
2479	14219	Max	14246,00	3535	20350	Max	20375,00
	14218	min	14218,00		20351	min	20349,00
	14240	media	14231,00		20350	media	20359,70
	14246	dev. Sta	11,28		20374	dev. Sta	12,33
	14239	incertezza	3,57		20374	incertezza	3,90
	14218				20350		
	14218				20351		
	14239				20375		
	14236				20349		
	14237				20373		
2727	15633	Max	15660,00	3718	21512	Max	23882,00
	15655	min	15633,00		23882	min	21492,00
	15636	media	15645,30		21516	media	21752,30
	15654	dev. Sta	12,12		21519	dev. Sta	748,44
	15633	incertezza	3,83		21492	incertezza	236,68
	15660				21542		
	15654				21495		
	15634				21518		
	15660				21521		
	15634				21526		
2948	16974	Max	16975,00	4060	23528	Max	23528,00
	16975	min	16921,00		23501	min	23454,00
	16954	media	16947,20		23501	media	23493,10
	16948	dev. Sta	21,62		23477	dev. Sta	23,89
	16927	incertezza	6,84		23454	incertezza	7,55
	16973				23454		
	16928				23504		
	16921				23504		
	16924				23504		
	16948				23504		
3114	17953	Max	17978,00	4483	25950	Max	25997,00
	17959	min	17903,00		25971	min	25946,00
	17927	media	17932,40		25978	media	25967,30
	17907	dev. Sta	27,26		25946	dev. Sta	17,45
	17978	incertezza	8,62		25979	incertezza	5,52
	17907				25973		
	17953				25947		
	17934				25980		
	17903				25997		
	17903				25952		
3263	18827	Max	18999,00	5310	30620	Max	30697,00
	18881	min	18822,00		30697	min	30599,00
	18999	media	18864,50		30671	media	30647,20
	18875	dev. Sta	52,22		30644	dev. Sta	34,85
	18842	incertezza	16,51		30650	incertezza	11,02
	18868				30599		
	18823				30599		
	18868				30648		
	18822				30647		
	18840				30697		

	ANDATA		4.6V
Laser [mm]	HC-SR04 μS		
6082	35304	Max	35329,00
	35324	min	35296,00
	35321	media	35311,80
	35305	dev. Sta	12,93
	35323	incertezza	4,09
	35321		
	35297		
	35329		
	35298		
	35296		

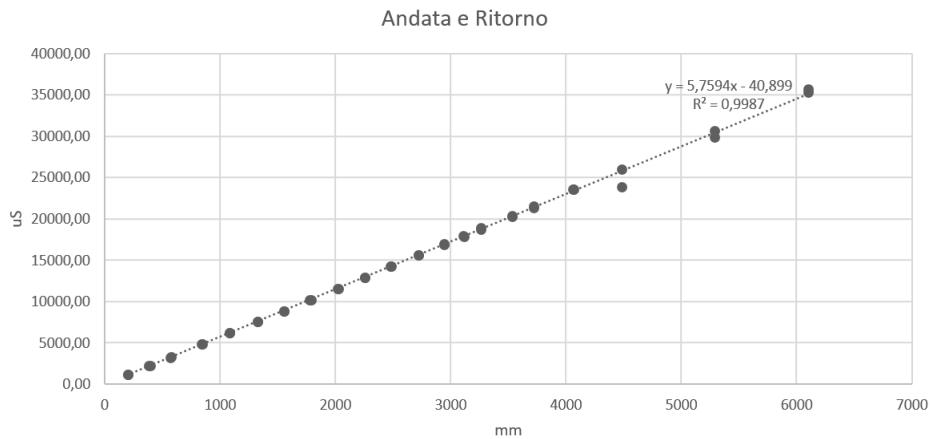
	RITORNO		4.6 V		RITORNO		4.6 V
Laser [mm]	HC-SR04 μS			Laser [mm]	HC-SR04 μS		
6081	23952	Max	35323,00	3528	20471	Max	20497,00
	35323	min	23952,00		20472	min	20418,00
	35274	media	34137,50		20495	media	20456,10
	35253	dev. Sta	3578,89		20444	dev. Sta	29,64
	35245	incertezza	1131,74		20418	incertezza	9,37
	35272				20473		
	35269				20422		
	35271				20423		
	35274				20446		
	35242				20497		
5281	30601	Max	30644,00	3262	18836	Max	18868,00
	30577	min	30577,00		18867	min	18813,00
	30577	media	30604,90		18837	media	18842,70
	30602	dev. Sta	21,40		18817	dev. Sta	19,11
	30627	incertezza	6,77		18813	incertezza	6,04
	30599				18868		
	30600				18863		
	30625				18844		
	30644				18838		
	30597				18844		
4483	25894	Max	25919,00	3112	18024	Max	18024,00
	25866	min	25866,00		17977	min	17951,00
	25888	media	25884,10		18024	media	17984,20
	25869	dev. Sta	17,18		17972	dev. Sta	28,85
	25866	incertezza	5,43		17973	incertezza	9,12
	25867				17952		
	25887				18024		
	25892				17972		
	25893				17951		
	25919				17973		
4062	23436	Max	23465,00	2941	16983	Max	17017,00
	23465	min	23412,00		16989	min	16959,00
	23435	media	23442,20		16959	media	16990,10
	23462	dev. Sta	16,50		17012	dev. Sta	20,10
	23463	incertezza	5,22		17017	incertezza	6,36
	23412				16959		
	23437				17011		
	23437				16992		
	23439				16988		
	23436				16991		
3721	21619	Max	21669,00	2723	15804	Max	15867,00
	21594	min	21572,00		15810	min	15784,00
	21643	media	21627,00		15856	media	15818,20
	21619	dev. Sta	32,37		15812	dev. Sta	24,87
	21645	incertezza	10,24		15806	incertezza	7,86
	21669				15784		
	21596				15808		
	21665				15867		
	21572				15821		
	21648				15814		

	RITORNO		4.6 V		RITORNO		4.6 V
Laser [mm]	HC-SR04 μS			Laser [mm]	HC-SR04 μS		
2480	14339	Max	14363,00	1326	7606	Max	7654,00
	14363	min	14312,00		7633	min	7603,00
	14338	media	14336,10		7606	media	7619,80
	14312	dev. Sta	14,53		7606	dev. Sta	16,99
	14313	incertezza	4,59		7603	incertezza	5,37
	14339				7628		
	14339				7606		
	14339				7626		
	14339				7630		
	14340				7654		
2256	13002	Max	13054,00	1084	7606	Max	7654,00
	13028	min	13001,00		7633	min	7603,00
	13034	media	13018,60		7606	media	7619,80
	13001	dev. Sta	18,17		7606	dev. Sta	16,99
	13027	incertezza	5,74		7603	incertezza	5,37
	13028				7628		
	13054				7606		
	13006				7626		
	13004				7630		
	13002				7654		
2016	11668	Max	11719,00	845	4857	Max	4881,00
	11689	min	11663,00		4881	min	4854,00
	11669	media	11685,20		4861	media	4858,90
	11663	dev. Sta	19,99		4854	dev. Sta	8,29
	11693	incertezza	6,32		4860	incertezza	2,62
	11714				4854		
	11719				4854		
	11663				4854		
	11687				4854		
	11687				4860		
1779	10266	Max	10266,00	571	3272	Max	3295,00
	10240	min	10215,00		3272	min	3271,00
	10240	media	10230,70		3271	media	3278,60
	10216	dev. Sta	17,46		3272	dev. Sta	10,11
	10240	incertezza	5,52		3295	incertezza	3,20
	10216				3295		
	10216				3272		
	10216				3289		
	10215				3274		
	10242				3274		
1552	8929	Max	8982,00	380	2194	Max	2194,00
	8960	min	8929,00		2194	min	2170,00
	8962	media	8957,20		2188	media	2188,70
	8932	dev. Sta	17,23		2194	dev. Sta	9,78
	8955	incertezza	5,45		2194	incertezza	3,09
	8961				2194		
	8981				2194		
	8955				2194		
	8955				2170		
	8982				2171		

	RITORNO		4.6 V
Laser [mm]	HC-SR04 μS		
200	1156	Max	1156,00
	1156	min	1156,00
	1156	media	1156,00
	1156	dev. Sta	0,00
	1156	incertezza	0,00
	1156		
	1156		
	1156		
	1156		
	1156		

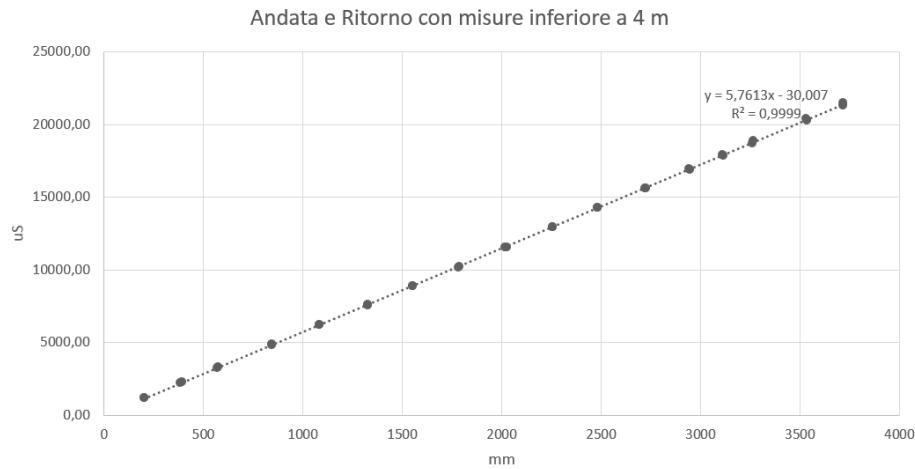
Una volta acquisiti i dati abbiamo preso di ogni serie le medie calcolate e con queste abbiamo iniziato l'elaborazione dei dati:

Abbiamo plottato sull'asse delle x le misure effettuate dal trasduttore laser in mm e sulle y quelle effettuate dal sensore ad ultrasuoni in $HC - SR04$ costruendo di fatto una curva di taratura. Questo procedimento è stato effettuato prima per le sole misurazioni in andata, poi per quelle in ritorno. Abbiamo poi effettuato un'interpolazione con una curva di regressione lineare e per entrambi i grafici abbiamo analizzato il valore di R per comprendere da che misura in poi lo strumento non rispondesse più in modo corretto. Come avevamo immaginato, dopo i $4\ m$ il sensore effettua misure con un'incertezza elevata dunque abbiamo deciso di scartare per il diagramma di taratura tali misure ponendo la portata dello strumento a $4\ m$.

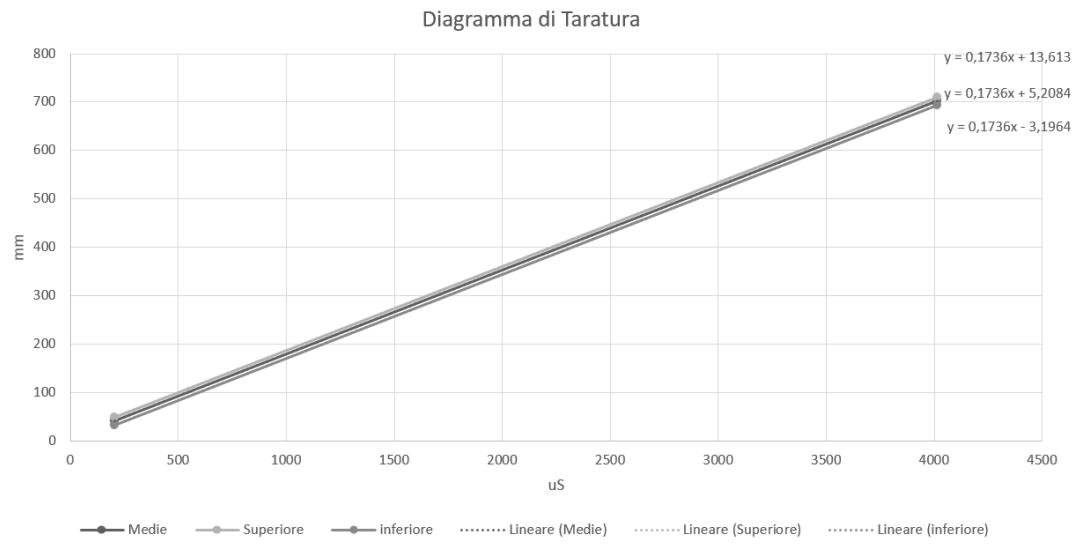


Dopo aver scartato i valori registrati sopra i $4\ m$ abbiamo unito i dati in andata e in ritorno e abbiamo notato l'assenza dell'effetto di isteresi e interpolando i dati abbiamo calcolato grazie alla funzione Excel *reg.line* il valore della linea di regressione notando una grande linearità: $R = 0.9999$ e uno scarto quadratico medio di $48,42\ \mu s$

Per ottenere questi dati abbiamo utilizzato il metodo dei minimi quadrati



Soddisfatti del risultato ottenuto abbiamo ricavato il diagramma di taratura



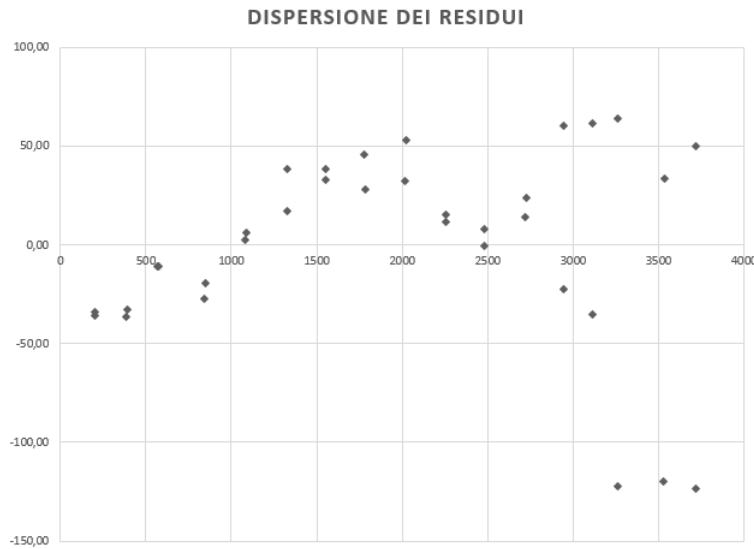
Trasduttore laser	HC-SR04	Metodo <i>minimi quadrati</i>	<i>d</i>	<i>d</i> ²
<i>mm</i>	μs	<i>Y_i</i>		
201	1164,00	1128,01	-35,99	1294,97
393	2266,90	2234,18	-32,72	1070,34
575	3293,40	3282,74	-10,66	113,62
846	4863,20	4844,05	-19,15	366,62
1085	6215,00	6221,00	6,00	36,04
1328	7603,80	7621,00	17,20	295,82
1554	8890,30	8923,05	32,75	1072,77
1788	10243,20	10271,20	28,00	783,85
2024	11578,20	11630,86	52,66	2773,52
2256	12956,00	12967,49	11,49	131,92
2482	14261,89	14269,54	7,65	58,53
2724	15639,90	15663,77	23,87	569,98
2945	16876,80	16937,02	60,22	3626,63
3115	17855,20	17916,44	61,24	3750,64
3261	18693,70	18757,59	63,89	4082,23
3535	20302,90	20336,19	33,29	1108,12
3718	21340,70	21390,51	49,81	2480,68
3718	21513,80	21390,51	-123,29	15201,31
3530	20427,10	20307,38	-119,72	14332,40
3263	18891,40	18769,11	-122,29	14953,65
3111	17928,70	17893,40	-35,30	1246,28
2942	16942,40	16919,74	-22,66	513,58
2720	15627,00	15640,73	13,73	188,49
2481	14264,60	14263,78	-0,82	0,68
2254	12940,70	12955,96	15,26	232,97
2017	11558,00	11590,54	32,54	1058,53
1780	10179,60	10225,11	45,51	2070,89
1553	8879,10	8917,29	38,19	1458,62
1325	7565,50	7603,72	38,22	1460,42
1082	6201,20	6203,72	2,52	6,35
842	4848,60	4821,01	-27,59	761,34
570	3265,00	3253,93	-11,07	122,46
385	2224,70	2188,09	-36,61	1340,04
204	1179,30	1145,30	-34,00	1156,12

Scarto quadratico medio:	48,42 μs
Sensibilità (<i>m</i>)	5,76 $\mu s/mm$
Intercetta	-30,00 μs

Livello di confidenza	Fattore di copertura	Incertezza estesa [μs]	Incertezza estesa [mm]
68,27%	1	48	8,4
90%	1,64	79	14
95%	1,96	95	16
99%	2,56	$1,2 \cdot 10^2$	21

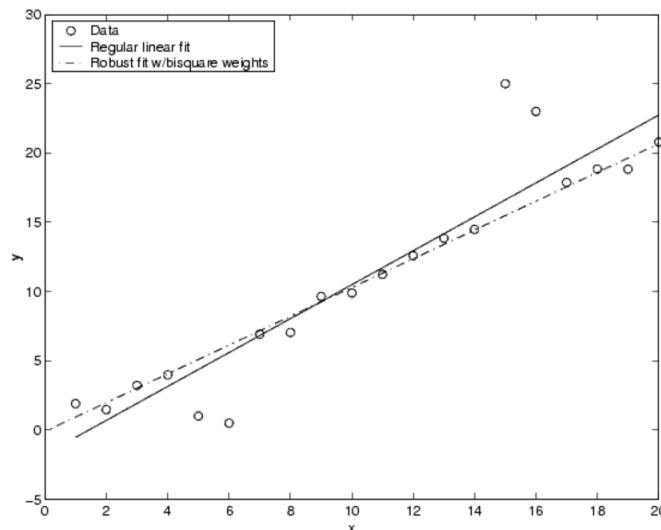
(I seguenti livelli di confidenza con i fattori di copertura sono stati presi da Accredia)

Una volta ricavata la il diagramma di taratura e la curva da utilizzarre nello strumento : $y = 0,1736x + 5.2084$ abbiamo fatto un controllo del risultato ottenuto andando a plottare i residui.



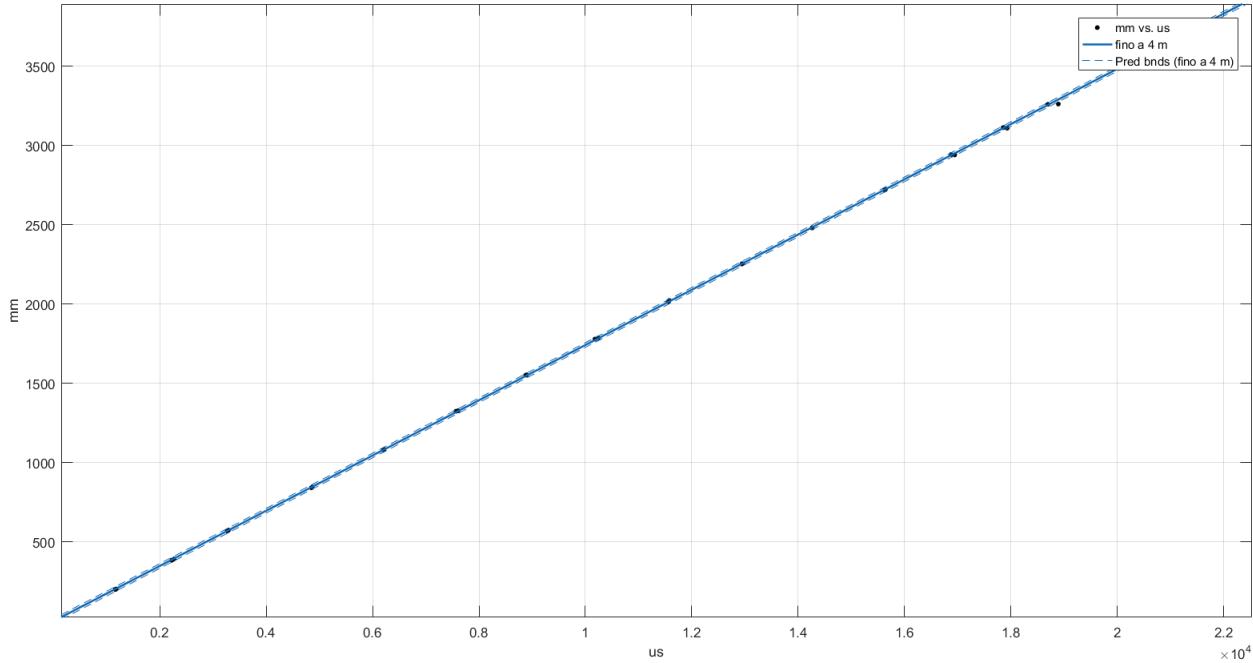
Se questi sono disposti in ordine causale sopra e sotto lo zero vuol dire che il modello utilizzato risulta corretto. In questo caso però abbiamo avuto il sospetto di un'autocorrelazione dei residui, avendo un andamento non del tutto casuale. Per cercare di capire se questa sensazione fosse giusta o meno, abbiamo cercato sul web qualche test, ma purtroppo non avendo nessuna competenza in statica non siamo riusciti a comprendere a pieno il test di Durbin-Waston il quale ha restituito un valore molto piccolo, ma tale valore risultava anche da un'esperienza fatta nel corso di un laboratorio, dove al contrario l'andamento era casuale, ne abbiamo dedotto quindi che magari le condizioni per l'applicazione di questo metodo non erano rispettate.

Dunque non contenti del risultato abbiamo scaricato l'estensione Matlab «Curve Fitting». Prima di tutto abbiamo creato su Matlab due vettori: uno con tutte le misure ordinate in mm e tutte le corrispondenti in μs . Attraverso questo utilissimo tool, sono stati plottati i valori e abbiamo apprezzato come software ci restituisse subito i valori del coefficiente angolare m della retta di taratura e dell'intercetta q , inoltre era possibile visualizzare subito i valori del RSME cioè dello scarto quadratico medio; inoltre grazie a questo programma possiamo decidere con che curva effettuare il fitting e facendo un po' di prove abbiamo confermato il fatto che la migliore per semplicità e accuratezza è la polinomiale di primo grado, cioè la retta. Successivamente abbiamo usato l'opzione «Robust fitting» grazie alla quale, usando la «Bisquare» abbiamo ottenuto una riduzione del RMSE che è passato da $8.4\ mm$ a $5.2\ mm$. Questo schema minimizza il peso della somma dei quadrati, dove il peso dato ad ogni punto dipende da quanto sono lontani dalla linea di fitting. Punti vicino alla linea hanno un peso pieno, i punti lontani dalla linea hanno un peso ridotto. I punti che sono più lontani dalla linea e che vengono interpretati come casuali hanno un peso nullo. Qui di seguito è riportato un esempio di come questa funzione operi.



Come possiamo vedere dal grafico superiore di comparazione tra il linear fit e il Robust fit usando bisquare weights, quest'ultimo segue la maggior parte dei dati e non è fortemente influenzato dagli outliers,

dunque grazie al tool di fitting di Matlab abbiamo ottenuto un nuovo diagramma di taratura riportato in seguito con i vari valori dei livelli di confidenza:



Livello di confidenza	Fattore di copertura	Incognita estesa [mm]
68,27%	1	5,2
90%	1,64	8,5
95%	1,96	10
99%	2,56	13

Successivamente, abbiamo però di migliorare ancora di più il nostro modello. Per far questo ci siamo impegnati in un'analisi di questo tipo: plottando su Matlab prima i valori che vanno da 0 a 500 mm, poi da 0 a 1000 mm e così via, e per ognuno di questi calcoliamo RMSE, coefficiente angolare e intercetta della linea di regressione.

Distanza riferimento [m]	m	q	fc	RMSE [mm]
4	0,1742	1,043	1	5,261
3,5	0,1742	0,7864	1	4,9608
3	0,1743	0,4769	1	4,049
2,5	0,1746	-1,231	1	3,507
2	0,1751	-3,367	1	1,96
1,5	0,1751	-3,3414	1	2,114
1	0,1748	-2,566	1	1,805
0,5	0,1756	-3,834	1	1,388

Come era prevedibile è stato notato un aumento dell'incertezza per un valore di misura sopra i due metri ed un valore del RMSE tendente a 2 per valori inferiori a due metri. Altro elemento che si può cogliere è la diminuzione del coefficiente angolare e l'aumento dell'intercetta all'aumentare della distanza. Dopo aver riflettuto sul significato di questo andamento abbiamo deciso di migliorare ulteriormente il nostro diagramma di taratura. In questo caso quello che è stato fatto è aver considerato un'incertezza costante di 2 mm fino ad una distanza di 2 m e successivamente avere un andamento che porti a congiungere tale valore con l'incertezza massima raggiunta a 4 m con il valore di 5,3 mm. Questo modello ci sembra il migliore finora presentato, in quanto:

- Per approssimare la linea di regressione è stato usato il metodo Bisquare di Matlab che ritorna un valore più accurato rispetto al metodo usato in precedenza.
- Inoltre l'adozione di un'incertezza variabile lungo il tratto di curva a maggior distanza rispecchia in pieno l'andamento dello strumento, che anche in fase di prima taratura, dove avevamo effettuato 10 misurazioni per ogni distanza, risultava meno accurato a distanza maggiori.

Per decidere il valore del coefficiente angolare da dare alla retta abbiamo eseguito una media dei coefficienti angolari pesata sul reciproco dell'incertezza, in modo tale che i contributi dei punti a minor incertezza siano maggiori rispetto a quelli con maggior incertezza. Per far questo abbiamo scritto un semplice script in matlab sotto riportato.

Media pesata inversa

```

1 | function [mediapesata]=mediaPesata(m, rmse)
2 | RMSE=10-rmse;
3 | somma=0;
4 | for a=1:length(m)
5 |     somma=m(a)*RMSE(a)+somma;
6 | end
7 | denominatore=sum(RMSE);
8 | mediapesata=somma/denominatore;
```

Da questo calcolo si prende il valore di m che risulta uguale a 0,1748. Per l'intercetta si applica la stessa funzione ai valori in esame risulta di -1,887. Grazie a questi dati si può quindi scrivere l'equazione della curva:

$$y = 0.1748x - 1.887$$

A questo punto siamo passati a descrivere le curve dell'incertezza. Come precedentemente detto, fino a 2 m si hanno due rette parallele alla principale e che hanno un'incertezza di 2 mm con fattore di copertura 1. Dunque le equazioni di queste rette sono:

$$\begin{cases} y_s = 0.1748x_s + 0.113 & 240 < x_s < 11425 \\ y_i = 0.1748x_i - 3.887 & 240 < x_i < 11425 \end{cases}$$

Anche qui come in precedenza è possibile fare una tabella per l'incertezza estesa con i vari livelli di confidenza:

Livello di confidenza	Fattore di copertura	Incercetza estesa [mm]
68.27%	1	2.0
90%	1,64	3.3
95%	1,96	3.9
99%	2,56	5.1

Dunque applicando un livello di confidenza del 99 % ho le due rette per l'incertezza diventano:

$$\begin{cases} y_s = 0.1748x_s + 3.213 & 240 < x_s < 11425 \\ y_i = 0.1748x_i - 6.987 & 240 < x_i < 11425 \end{cases}$$

Successivamente abbiamo calcolato l'equazione della retta per quanto riguarda il secondo tratto di interesse. Per far questo abbiamo semplicemente utilizzato la formula per la retta passante per due punti.

L'incertezza massima che si ha per 4 m è la seguente:

$x = 22894 \Rightarrow y = 4m$		
Livello di confidenza	Fattore di copertura	Incercetza estesa [mm]
68.27%	1	5,3
90%	1,64	8,7
95%	1,96	10
99%	2,56	14

Quindi ad una distanza di 4 m e con un livello di confidenza del 99% si ha che:

$$\begin{cases} y_s = 0.1748x_s + 12.113 & x_s = 22894 \\ y_i = 0.1748x_i - 15.887 & x_i = 22894 \end{cases}$$

Dunque si possono ricavare le equazioni delle due curve che descrivono l'incertezza per una misura sopra i 2 m :

$$\begin{cases} y_s = 0.1756x_s - 5.6954 & 11425 < x_s < 22894 \\ y_i = 0.1740x_i + 1.9214 & 11425 < x_i < 22894 \end{cases}$$

Abbiamo così caratterizzato le curve che descrivono l'incertezza.

La curva superiore è descritta da questo sistema:

$$\begin{cases} y = 0.1748x + 3.213 & 240 < x < 11425 \\ y = 0.1756x - 5.6954 & 11425 < x < 22894 \end{cases}$$

La curva inferiore da quest'altro:

$$\begin{cases} y = 0.1748x - 6.987 & 240 < x < 11425 \\ y = 0.1740x + 1.921 & 11425 < x < 22894 \end{cases}$$

Codice diagramma di taratura

```

function [mInferiore ,qInferiore ,mSuperiore ,qSuperiore]=
diagrammaTaratura(livelloConfidenza)

4 if livelloConfidenza==99
    incDueMetri=5.1;
    incQuattroMtri=14;
8 else if livelloConfidenza==95
    incDueMetri=3.9;
    incQuattroMtri=10;
10 else if livelloConfidenza==90
    incDueMetri=3.3;
    incQuattroMtri=8.7;
12 else if livelloConfidenza==68
    incDueMetri=2;
    incQuattroMtri=5.3;
14 end
16 end
18 end
end

20 // costanti //
22 m=0.1748;
q=-1.887;
24 max1=(4000-q)/m;
max2=(2000-q)/m;
26 min1=(40-q)/m;

28 // incertezza sotto i due metri //
x=min1:0.1:max1;
y=m*x+q;
plot(x,y,'b');
hold on;
xlabel(['$ \mu $'], 'interpreter', 'latex');
34 ylabel(['$ mm $'], 'interpreter', 'latex');
title('Diagramma di taratura');
36 x1=min1:0.1:max2;
y1=m*x1+q+incDueMetri;
38 y2=m*x1+q-incDueMetri;
plot(x1,y1, 'r')
40 hold on;
plot(x1,y2, 'r')
42 hold on;

44 // incertezza superiore sopra 2 m //
x3=max2:0.1:max1;
46 xs=max2;
ys=m*xs+q+incDueMetri;
48 x3s=max1;
y3s=m*x3s+q+incQuattroMtri;
50 m3=(y3s-ys)/(x3s-xs);
Ys=m3.* (x3-xs)+ys;
52 q3=m3*-xs+ys;
plot(x3,Ys,'y');
hold on;

56 // Incertezza inferiore sopra 2 m //
x0=max2;
58 y0=m*x0+q-incDueMetri;
x4=max1;
60 y4=m*x4+q-incQuattroMtri;

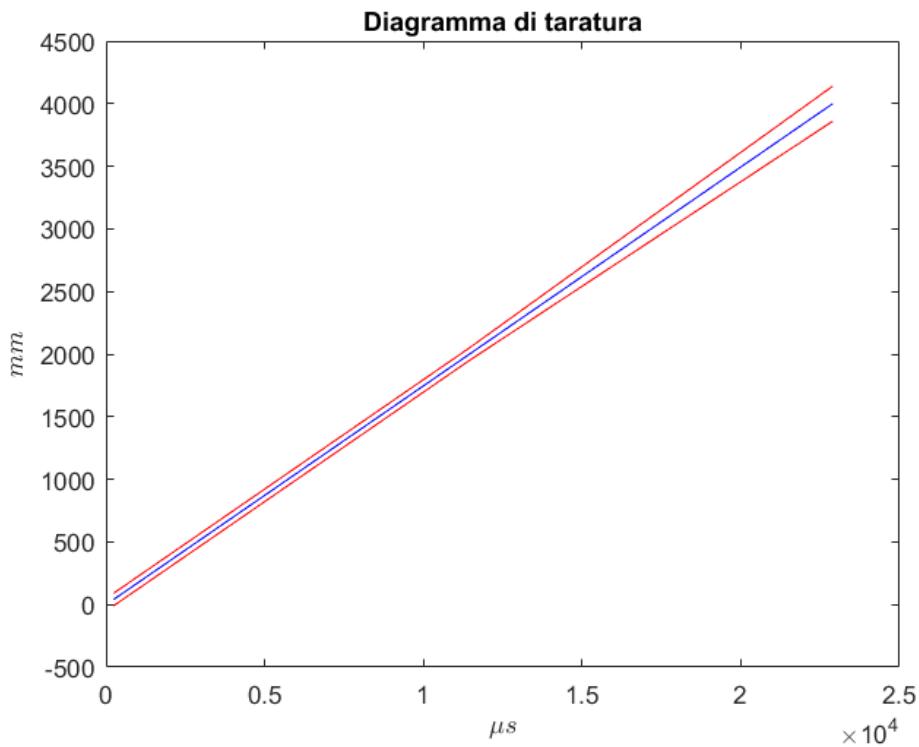
```

```

62 m4=(y4-y0)/(x4-x0);
Y=m4.* (x3-x0)+y0;
plot(x3,Y, 'y');
hold on
q4=m4*-x0+y0;
66
// risultati //
68 mInferiore=m4;
qInferiore=q4;
70 mSuperiore=m3;
qSuperiore=q3;
72 end

```

Il seguente codice Matlab è una funzione che prende in ingresso il livello di confidenza da usare e restituisce i valori dei coefficienti angolari e delle intercette che vanno a descrivere il diagramma di taratura per quel determinato livello di confidenza. Una volta descritte queste rette, sono state poi inserite nel codice dello strumento le corrispondenti equazioni per essere utilizzate. Questa funzione ritorna inoltre il diagramma di taratura seguente (l'asse dei mm è stato scalato per rendere maggiormente visibile le varie curve):



Come si può vedere dal grafico l'incertezza rimane costante fino ai due metri, successivamente aumenta all'aumentare della distanza misurata. In questo diagramma non partiamo dal valore zero in quanto, dopo diverse prove abbiamo stimato la distanza di stand-off dello strumento a 40 mm.

Infine ci siamo posti il problema dell'alimentazione e se la sensibilità cambiasse al cambiare dell'alimentazione. Per questo abbiamo effettuato una seconda taratura ad una tensione di alimentazione di 4.6 V .

Con nostra grande sorpresa i risultati sono risultati uguali a quelli prodotti dalle misurazioni eseguite con un'alimentazione di 5 V, dunque facendo una valutazione sull'incertezza abbiamo deciso di non andare a modificare il nostro diagramma di taratura. Inoltre per evitare a monte questo problema abbiamo inserito un cavo che legge la tensione di alimentazione data al sensore e sul display è infatti possibile vedere la percentuale di batteria rimanente, e nel caso non fosse sufficiente verrà visualizzato un avviso per il cambio batteria (una tacca di batteria). Inoltre l'alimentazione al sensore risulta costante anche al variare della carica della batteria, poiché questa alimenta Arduino a 9 V che a sua volta alimenta a 5 V il sensore (e gli altri componenti del circuito), quindi per un buon intervallo di funzionamento siamo sicuri del funzionamento del nostro apparecchio.

8 Programmazione codice strumento

Codice strumento

```

// Inserimento libreria //
2 #include<LiquidCrystal.h>

4 // Dichiarazione classe LCD //
LiquidCrystal LCD(2, 3, 4, 7, 8, 12);

6 // Dichiarazione Pin sensore HC-SR04 //
8 const int trigger = 13;
const int echo = 11;

10 // Pulsanti //
12 int switchStateMisura = 0;
int prevSwitchStateMisura = 0;
14 const int switchPinMisura = 6;
const int compensazione = 160;
16 int switchStateComp = 0;
int prevSwitchStateComp = 0;
18 const int switchPinComp = 5;
int distanzaCp;

20 // Piezo //
22 int buzzer = A4;

24 // Contatori vari //
int control = 0;
26 int frontDistance = 0;
int cont = 0;
28 int verCont = 0;

30 // LED //
const int LEDR = 9;
32 const int LEDG = 10;

34 // Ingressi analalogici //
const int batteriaMis = A0;
36 int batteria = 1023;
int batteriaLivello = 0;

38 // Variabili di misura //
40 const int confidenzaMis = A1;
int confidenza;
42 int arrayMisure[10];
int numeroMisurazioni = 11;
44 int misurazioni = numeroMisurazioni - 1;
int distanza = 0;
46 int incertezzaMis = 0;
int prevConf = 2000;
48 int conf = 0;
int contStart = 0;
50 const int minValue = 40;
const int maxValue = 4000;

52 // LCD //
54 const int contrasto = 110;
int contrastoPin = A2;

56 // Funzione misurazione //
58 int misura() {

```

```

long array[14];
for (long i = 0; i < 14; i++){      //Creazione e inizializzazione di tre array //
    array[i] = 0;
}
long array12[12];
for(long s = 0; s < 12; s++){
    array12[s] = 0;
}
long array10[10];
for(long t = 0; t < 10; t++){
    array10[t] = 0;
}
long cont = 0;
long durata = 0;
int contLed = 0;
digitalWrite(LED_R, HIGH);

for (long j = 0; j < 15; j++){
    do {
        digitalWrite(trigger, LOW);
        delayMicroseconds(2);
        digitalWrite(trigger, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigger, LOW);
        durata = pulseIn(echo, HIGH);
    } while (durata == 0);
    if(cont == 0){
        cont = 1;
    } else {
        array[j-1] = durata;           //Si fanno 15 misurazioni //
    }                               //ma se ne salvano solo 14 //
    delay(120);                   // (scartando la prima) //
    if (contLed == 0){
        digitalWrite(LED_R, LOW);
        contLed = 1;
    } else {
        digitalWrite(LED_R, HIGH);
        contLed = 0;
    }
}

long massimo2 = array12[0];
long minimo2 = array12[0];
for(long x = 1; x < 12; x++){      //Si trovano massimo e minimo nell'array di 12 //
    massimo2 = max(massimo2, array12[x]);
    minimo2 = min(minimo2, array12[x]);
}

long prev2 = 0;
long contMax2 = 0;
long contMin2 = 0;

for(long k = 0; k < 12; k++){      //Si riempie l'array 10 //
    if(((array12[k] != massimo2) && (array12[k] != minimo2)) ||
       ((contMax2 == 1) && (array12[k] == massimo2)) ||
       ((contMin2 == 1)&& (array12[k] == minimo2))){
        array10[prev2] = array12[k];
        prev2++;
    } else if (array12[k] == massimo2) {
        contMax2 = 1;
    } else if (array12[k] == minimo2) {
}

```

```

    contMin2 = 1;
122 }
}
124
long somma = 0;
126 long media = 0;

128 for(long c = 0; c < 10; c++){
    somma = somma + array10[c];
130 }

132 media = somma/10;
int misurazionemm = (0.1748 * durata)-1.887;
134 int sommaDev = 0;

136 for(int t = 0; t < misurazioni; t++){
    sommaDev= (media - array10[t])^2 + sommaDev;
138 }

140 int devSt = sqrt(sommaDev/10);
int incertezza = devSt/(sqrt(10));
142 int b = 0;
int valore = analogRead(confidenzaMis);
144 int lc = map(valore, 0, 1023, 0, 2);

146 if (incertezza <= LC[lc]){
    return misurazionemm;
148 } else {
    return 0;
150 }
152

// Funzione calcolo incertezza estesa //
154 int LC[3] = {2, 4, 5};
int incertezzaVal(int misurazione){
156     int valore = analogRead(confidenzaMis);
    int lc = map(valore, 0, 1023, 0, 2);
    int Micros = (misurazione + 1.887)/0.1748;
    int y1 = misurazione;
    int y2 = 0;
    switch (lc) {
162         case 0:
            if (misurazione <= 2000) {
164             return LC[0];
            } else {
166                 y2 = 0.1751*Micros - 3.1901;
                    return abs(y2-y1);
            }
        case 1:
            if (misurazione <= 2000) {
                return LC[1];
            } else {
                y2 = 0.1753*Micros - 4.0828;
                return abs(y2-y1);
            }
        case 2:
            if (misurazione <= 2000) {
178                 return LC[2];
            } else {
                y2 = 0.1756*Micros - 5.6954;
                return abs(y2-y1);
            }
    }
180
182 }
```

```

184     }
185 }
186 // Funzione stampa livello confidenza //
187 void incertezza() {
188     int valore = analogRead(confidenzaMis);
189     int lc = map(valore, 0, 1023, 0, 2);
190     if (lc == 0) {
191         LCD.print("No");
192     } else if (lc == 1) {
193         LCD.print("95%");
194     } else {
195         LCD.print("99%");
196     }
197 }
198
199 // Funzioni stampa errori //
200 void limiteInf() {
201     LCD.clear();
202     LCD.setCursor(4, 0);
203     LCD.print("Errore 01");
204     LCD.setCursor(0,1);
205     LCD.print("Ripetere misura!");
206 }
207
208 void limiteSup() {
209     LCD.clear();
210     LCD.setCursor(4, 0);
211     LCD.print("Errore 02");
212     LCD.setCursor(0,1);
213     LCD.print("Ripetere misura!");
214 }
215
216 void erroreMis() {
217     LCD.clear();
218     LCD.setCursor(4, 0);
219     LCD.print("Errore 03");
220     LCD.setCursor(0,1);
221     LCD.print("Ripetere misura!");
222 }
223
224 // Creazione caratteri speciali LCD //
225 byte inc[8] = {
226     B00100,
227     B00100,
228     B11111,
229     B00100,
230     B00100,
231     B00000,
232     B11111,
233     B00000,
234 };
235
236 byte level100[8] {
237     B11111,
238     B11111,
239     B00000,
240     B11111,
241     B11111,
242     B00000,
243     B11111,
244     B11111,

```

```

    };
246 byte level50[8] {
248     B00000,
249     B00000,
250     B00000,
251     B11111,
252     B11111,
253     B00000,
254     B11111,
255     B11111,
256 };
257
258 byte level10[8] {
259     B00000,
260     B00000,
261     B00000,
262     B00000,
263     B00000,
264     B00000,
265     B11111,
266     B11111,
267 };
268
// Funzione stampa batteria //
270 void stampaBatteria () {
271     if((batteria <= 1023) && (batteria >= 900)) {
272         LCD.print(char(1));
273         batteriaLivello = 100;
274     } else if (batteria == 500) {
275         LCD.print(char(2));
276         batteriaLivello = 50;
277     } else if (batteria == 100) {
278         LCD.print(char(3));
279         batteriaLivello = 10;
280     }
281 }
282
// Setup //
283 void setup() {
284     pinMode(trigger, OUTPUT);
285     pinMode(echo, INPUT);
286     pinMode(A5, OUTPUT);
287     LCD.begin(16, 2);
288     pinMode(LED_R, OUTPUT);
289     pinMode(LED_G, OUTPUT);
290     digitalWrite(LED_R, LOW);
291     digitalWrite(LED_G, LOW);
292     pinMode(contrastoPin, OUTPUT);
293     analogWrite(contrastoPin, contrasto);
294     pinMode(buzzer, OUTPUT);
295     pinMode(switchPinMisura, INPUT);
296     pinMode(switchPinComp, INPUT);
297     LCD.setCursor(3,0);
298     LCD.print("Benvenuto!");
299     delay ( 500 );
300
301     tone(buzzer, 660 ,100);
302     delay ( 150 );
303     tone(buzzer, 660 ,100);
304     delay ( 300 );
305     tone(buzzer, 660 ,100);
306 }
```

```

308     delay ( 300);
  tone(buzzer, 510 ,100);
  delay ( 100);
  tone(buzzer, 660 ,100);
  delay ( 300);
  tone(buzzer, 770 ,100);
  delay ( 550);
  tone(buzzer, 380 ,100);
  delay ( 575);
  tone(buzzer, 510 ,100);
  delay ( 450);
  tone(buzzer, 380 ,100);
  delay ( 400);
  tone(buzzer, 320 ,100);
  delay ( 500);
  tone(buzzer, 440 ,100);
  delay ( 300);
  tone(buzzer, 480 ,80);
  delay ( 330);
  tone(buzzer, 450 ,100);
  delay ( 150);
  tone(buzzer, 430 ,100);
  delay ( 300);
  tone(buzzer, 380 ,100);
  delay ( 200);
  tone(buzzer, 660 ,80);
  delay ( 200);
  tone(buzzer, 760 ,50);
  delay ( 150);
  tone(buzzer, 860 ,100);
  delay ( 300);
  tone(buzzer, 700 ,80);
  delay ( 150);
  tone(buzzer, 760 ,50);
  delay ( 350);
  tone(buzzer, 660 ,80);
  delay ( 300);
  tone(buzzer, 520 ,80);
  delay ( 150);
  tone(buzzer, 580 ,80);
  delay ( 150);
  tone(buzzer, 480 ,80);
  delay ( 500);

350
LCD.clear();
LCD.setCursor(0,3);
LCD.print(" Progetto S.Fusari, A.Fuso, G. Gaeta.");
354 for (int i = 0; i < 26 ; i++){
    LCD.scrollDisplayLeft();
    delay(250);
}
358 LCD.createChar (0, inc);
LCD.createChar (1, level100);
360 LCD.createChar (2, level50);
LCD.createChar (3, level10);
362 digitalWrite(LEDG, HIGH);
LCD.clear();
364 LCD.setCursor(0,0);
LCD.print("Per iniziare, ");
366 LCD.setCursor(0,1);
LCD.print("premi MIS!");
368 }

```

```

370 // Loop // void loop() {
371   batteria = analogRead(batteriaMis);
372   switchStateMisura = digitalRead(switchPinMisura);
373   conf = map(analogRead(confidenzaMis), 0, 1023, 0, 2);
374   if (switchStateMisura != prevSwitchStateMisura) {
375     contStart = 1;
376     LCD.clear();
377     LCD.setCursor(0,0);
378     LCD.print("Sto misurando");
379     LCD.setCursor(0,1);
380     LCD.print("Attendere...");
381     digitalWrite(LEDG, LOW);
382     confidenza = analogRead(confidenzaMis);
383     int dist = misura();
384     distanza = dist;
385     if (distanza <= minValue) {
386       limiteInf();
387       verCont = 0;
388     } else if (distanza >= maxValue) {
389       limiteSup();
390       verCont = 0;
391     } else if (distanza == 0) {
392       erroreMis();
393       verCont = 0;
394     } else {
395       digitalWrite(LEDG, HIGH);
396       LCD.clear();
397       LCD.setCursor(0,0);
398       LCD.print("Cp:");
399       if (control == 1) {
400         LCD.print("Si ");
401         distanzaCp = distanza+compensazione;
402         cont = 1;
403       } else if (control == 0) {
404         LCD.print("No ");
405         cont = 0;
406         distanzaCp = distanza;
407       }
408       LCD.setCursor(6,0);
409       LCD.print("LC:");
410       incertezza();
411       LCD.setCursor(15,0);
412       stampaBatteria();
413       LCD.setCursor(0,1);
414       LCD.print("Ans: ");
415       LCD.print(distanzaCp);
416       LCD.print(char(0));
417       confidenza = incertezzaVal(distanza);
418       LCD.print(confidenza);
419       LCD.print("mm");
420       delay(300);
421       verCont = 1;
422     }
423   }
424   switchStateComp = digitalRead(switchPinComp);
425   if ((switchStateComp != prevSwitchStateComp )&&(verCont==1)) {
426     if (control == 0) {
427       LCD.clear();
428       LCD.setCursor(0,0);
429       LCD.print("Cp:Si");
430       LCD.setCursor(6,0);

```

```

432     LCD.print("LC:");
433     incertezza();
434     LCD.setCursor(15,0);
435     stampaBatteria();
436     LCD.setCursor(0,1);
437     LCD.print("Ans: ");
438     if (cont == 0) {
439         distanzaCp = distanza + compensazione;
440         LCD.print(distanzaCp);
441     } else {
442         LCD.print(distanza);
443     }
444     LCD.print(char(0));
445     confidenza = incertezzaVal(distanza);
446     LCD.print(confidenza);
447     LCD.print("mm");
448     control = 1;
449     delay(300);
450 } else if (control == 1 ){
451     LCD.clear();
452     LCD.setCursor(0,0);
453     LCD.print("Cp:No");
454     LCD.setCursor(6,0);
455     LCD.print("LC:");
456     incertezza();
457     LCD.setCursor(15,0);
458     stampaBatteria();
459     LCD.setCursor(0,1);
460     LCD.print("Ans: ");
461     if (cont == 1) {
462         distanzaCp = distanza - compensazione;
463         LCD.print(distanzaCp);
464     } else {
465         LCD.print(distanza);
466     }
467     LCD.print(char(0));
468     confidenza = incertezzaVal(distanza);
469     LCD.print(confidenza);
470     LCD.print("mm");
471     control = 0;
472     delay(300);
473 }
474 if ((prevConf != conf)&&(contStart ==1)&&(verCont==1)) {
475     LCD.clear();
476     LCD.setCursor(0,0);
477     LCD.print("Cp:");
478     if (control == 1) {
479         LCD.print("Si ");
480     } else if (control == 0) {
481         LCD.print("No ");
482     }
483     LCD.setCursor(6,0);
484     LCD.print("LC:");
485     incertezza();
486     LCD.setCursor(15,0);
487     stampaBatteria();
488     LCD.setCursor(0,1);
489     LCD.print("Ans: ");
490     if (control==1){
491         distanzaCp = distanza + compensazione;
492         LCD.print(distanzaCp);

```

```

494     } else {
495         LCD.print(distanza);
496     }
497     LCD.print(char(0));
498     confidenza = incertezzaVal(distanza);
499     LCD.print(confidenza);
500     LCD.print("mm");
501     delay(300);
502     prevConf = conf;
503 }

```

8.1 Spiegazione parti di misurazione

Per ottenere la misura di distanza abbiamo deciso di implementare una funzione *misura()* che procede secondo i seguenti passi:

1. Vengono creati e inizializzati (a zero) tre array di dimensioni diverse: 14, 12, 10. Quello di 14 elementi verrà poi riempito con le misurazioni (in microsecondi μs); quello da 12 costituisce un array temporaneo, mentre quello da 10 è l'array su cui viene effettutata la media (per ottenere quello che è l'output dello strumento).

```

50 int misura() {
51     long array[14];
52     for (long i = 0; i < 14; i++){
53         array[i] = 0;
54     }
55     long array12[12];
56     for(long s = 0; s < 12; s++){
57         array12[s] = 0;
58     }
59     long array10[10];
60     for(long t = 0; t < 10; t++){
61         array10[t] = 0;
62     }

```

2. Siccome ci siamo resi conto che molte volte come prima misurazione il sensore restituisse 0 (causa ignota), la funzione esegue 15 misurazioni ma solo 14 vengono salvate nel primo array (la prima è scartata), tutto questo avviene tramite il costrutto *do while*, che permette in questo caso di accettare soltanto valori che non siano pari a 0.
 - (a) La funzione *delayMicroseconds()* è utilizzata per mettere in pausa la scheda, evitando errori tra un'operazione e l'altra.
 - (b) La funzione *pulseIn()* ritorna la lunghezza del tempo dell'impulso in microsecondi: aspetta che il pin *echo* sia *HIGH*, inizia contare i millisecondi finché questo non diventa *LOW*. Il tempo è determinato empiricamente, e molto probabilmente genera errori con impulsi brevi (lavora con impulsi che vanno da una durata minima di $2\mu s$ ad una massima di tre minuti).

```

68 for (long j = 0; j < 15; j++){
69     do {
70         digitalWrite(trigger, LOW);
71         delayMicroseconds(2);
72         digitalWrite(trigger, HIGH);
73         delayMicroseconds(10);
74         digitalWrite(trigger, LOW);
75         durata = pulseIn(echo, HIGH);
76     } while (durata == 0);

```

3. Successivamente si calcolano massimo e minimo dell'array con 14 elementi, siccome è probabile che siano causa di errori questi valori vengono trascurati (l'algoritmo fa in modo che vengano eliminati solo una volta, evitando di avere dei valori pari a zero nell'array successivo, quello da 12). Il procedimento viene eseguito analogamente nel passaggio $array12[] \rightarrow array10[]$, quello finale.

```

92 | long massimo2 = array12[0];
93 | long minimo2 = array12[0];
94 |
95 | for(long x = 1; x < 12; x++){
96 |     massimo2 = max(massimo2, array12[x]);
97 |     minimo2 = min(minimo2, array12[x]);
98 |
99 |
100 |     long prev2 = 0;
101 |     long contMax2 = 0;
102 |     long contMin2 = 0;
103 |
104 |     for(long k = 0; k < 12; k++){
105 |         if(((array12[k] != massimo2) && (array12[k] != minimo2)) ||
106 |             ((contMax2 == 1) && (array12[k] == massimo2)) ||
107 |             ((contMin2 == 1) && (array12[k] == minimo2)) ){
108 |             array10[prev2] = array12[k];
109 |             prev2++;
110 |         } else if (array12[k] == massimo2) {
111 |             contMax2 = 1;
112 |         } else if (array12[k] == minimo2) {
113 |             contMin2 = 1;
114 |         }
115 |     }
116 |
117 | }
```

4. Una volta ottenuto l'array con le 10 misurazioni in ms , si procede al calcolo della media.

```

116 | long somma = 0;
117 | long media = 0;
118 |
119 | for(long c = 0; c < 10; c++){
120 |     somma = somma + array10[c];
121 | }
```

5. A questo punto si entra nella fase finale:

- (a) Si calcola la misura attraverso l'equazione che è stata ricavata nella fase di taratura.

```

int misurazionemm =
(0.1748 * durata)-1.887;
```

- (b) Si calcola la Deviazione Standard dell'array contenente le 10 misurazioni e la relativa incertezza.

```

126 int sommaDev = 0;

128 for(int t = 0; t < misurazioni; t++){
    sommaDev= (media - array10[t])^2+sommaDev;
}

132 int devSt = sqrt(sommaDev/10);
int incertezza = devSt/(sqrt(10));

```

- (c) Una volta conosciuta l'incertezza della misurazione effettuata si procede alla verifica della compatibilità con i dati raccolti nella fase di taratura dello strumento; la funzione restituisce quindi la misura in *mm* nel caso sia stata verificata la compatibilità, in caso contrario restituisce 0.

```

136 int valore = analogRead(confidenzaMis);
    int lc = map(valore, 0, 1023, 0, 2);

138 if (incertezza <= LC[lc]){
    return misurazionemm;
}
140 else {
    return 0;
}
142

```

Dopo aver eseguito varie misurazioni, ci siamo accorti che se la distanza dell'oggetto da misurare è inferiore ai *4 cm* si generano degli errori nel risultato, per questo motivo abbiamo deciso di effettuare un controllo sulla distanza, che permetta lo stampaggio a video di un messaggio di errore (*Errore 01*) nel caso in cui la distanza sia inferiore ai *4 cm* :

```

200 void limiteInf() {
201     LCD.clear();
202     LCD.setCursor(4, 0);
203     LCD.print("Errore 01");
204     LCD.setCursor(0,1);
205     LCD.print("Ripetere misura!");
206 }

```

Come spiegato in precedenza, l'incertezza dello strumento tende a crescere di molto per distanze superiori ai *4m*, fatto che ci ha portato a definire la portata a questa distanza. Si è quindi deciso di creare un controllo sulla misurazione, che nel caso in cui si esca dal campo di misura (superando i *4m*) venga stampato sul display *Errore02* :

```

208 void limiteSup() {
209     LCD.clear();
210     LCD.setCursor(4, 0);
211     LCD.print("Errore 02");
212     LCD.setCursor(0,1);
213     LCD.print("Ripetere misura!");
214 }

```

Il controllo viene eseguito per ogni misurazione nel loop:

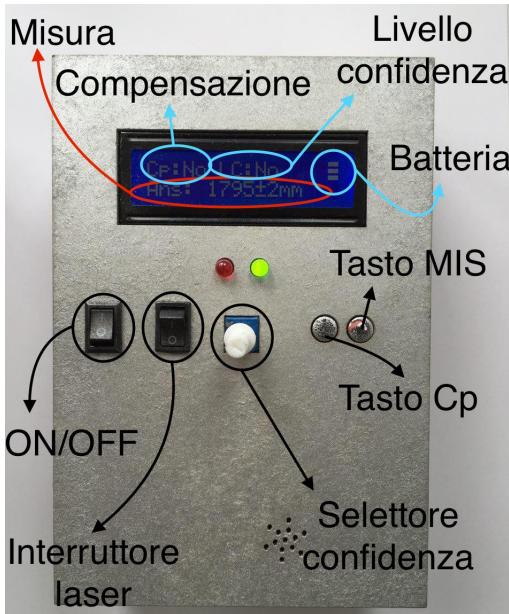
```
386 | if (distanza <= minValue) {  
387 |     limiteInf();  
388 |     verCont = 0;  
389 | } else if (distanza >= maxValue) {  
390 |     limiteSup();  
391 |     verCont = 0;  
392 | } else if (distanza == 0) {  
393 |     erroreMis();  
394 |     verCont = 0;  
395 | }
```

La funzione *erroreMis()* , che stampa a video il codice errore *Errore 03* , viene eseguita solo se la funzione *misura()* restituisce il valore 0, e ciò, come già spiegato in precedenza, avviene se l'incertezza delle misurazioni effettuate non è compatibile con quella ottenuta nel corso della taratura dello strumento.

9 Conclusioni

Ripercorrendo velocemente il percorso effettuato per la costruzione e tartatura dello strumento abbiamo prima di tutto progettato il circuito elettrico e il case, successivamente abbiamo montato il tutto e progettato il codice di tartatura e in seguito il codice di funzionamento dello strumento in se collegando i vari pulsanti e altro hardware. Dopo queste operazioni ci siamo concentrati sulla taratura dello strumento, effettuando prima una taratura campione, per studiarne il comportamento in prima approssimazione, successivamente ci siamo messi in condizioni più adatte ed abbiamo effettuato una serie di misurazioni con lo scopo di effettuare una taratura back-to-back con un trasduttore laser. Effettuata questa operazione delicata abbiamo analizzato scrupolosamente i dati ottenuti. In primo luogo abbiamo estrapolato i dati utili e li abbiamo graficati, in seguito abbiamo eliminato gli errori sistematici e stabilito la portata e campo di misura dello strumento ed infine plottato un primo diagramma di taratura. Non contenti del risultato ottenuto, abbiamo cercato una strada diversa per migliorare il nostro modello e grazie al programma di fitting di Matlab siamo riuciti ad effettuare un fitting più preciso ed abbiamo deciso dopo un'attenta analisi di utilizzare una curva dell'incertezza che variasse per tener conto della maggiore incertezza ad una maggiore distanza.

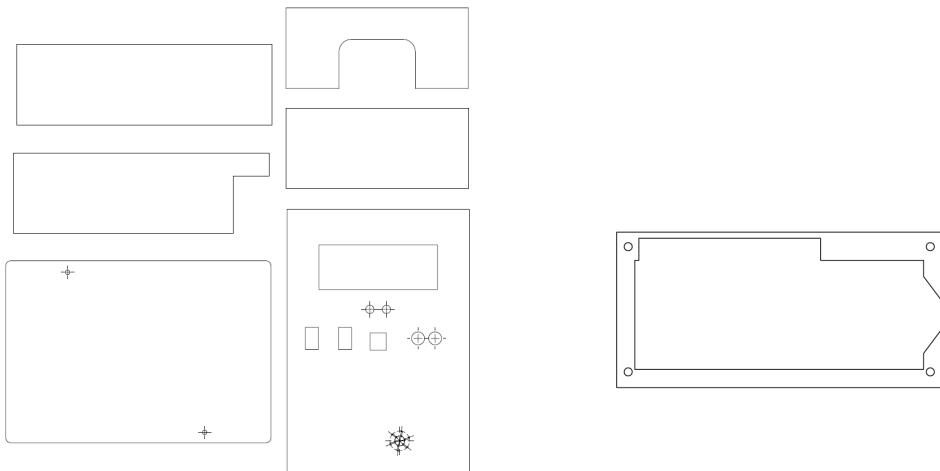
Dopo questa completa analisi possiamo fortunatamente avere un datasheet completo del nostro prodotto.



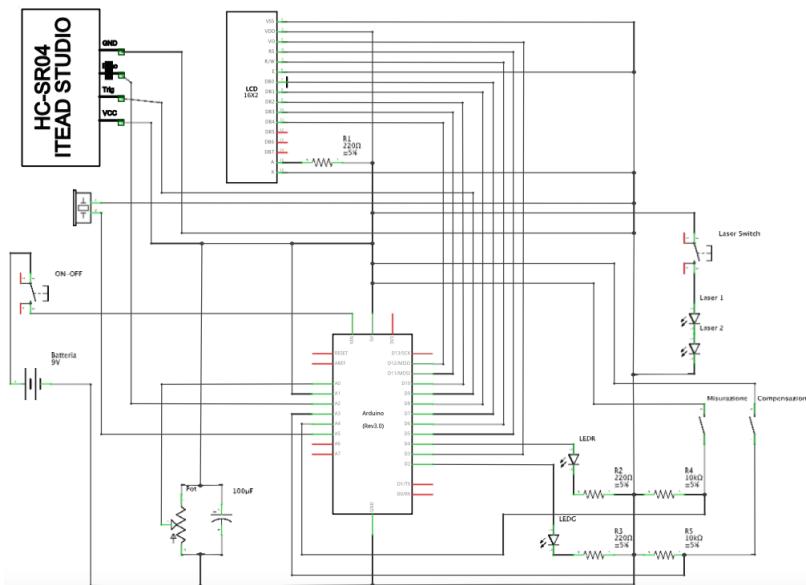
- Breve guida all'utilizzo:

- Per eseguire una misura premere il tasto *MIS* e attendere che il dispositivo abbia elaborato i dati.
- Se si vuole aggiungere la lunghezza del dispositivo a quella misurata (per misurare ad esempio la distanza tra pavimento e soffitto), è sufficiente premere il pulsante *Cp*; l'attivazione o meno della funzione compensazione rimane automaticamente in memoria nella misura successiva.
- Se si vuole scegliere il livello di confidenza della misura si deve ruotare il potenziometro, potendo scegliere tra tre diversi valori di livello di confidenza:
 1. Nessuno.
 2. 95%
 3. 99%
- Si consiglia di sostituire la batteria quando l'indicatore della stessa segna una sola tacca.
- Schermate di errori:
 1. *Errore 01* : si verifica quando la distanza è inferiore ai 4 cm
 2. *Errore 02* : si verifica quando si supera il limite superiore del campo di misura (4 m)
 3. *Errore 03* : viene mostrato nel caso in cui ci sia stato un errore nel calcolo della distanza

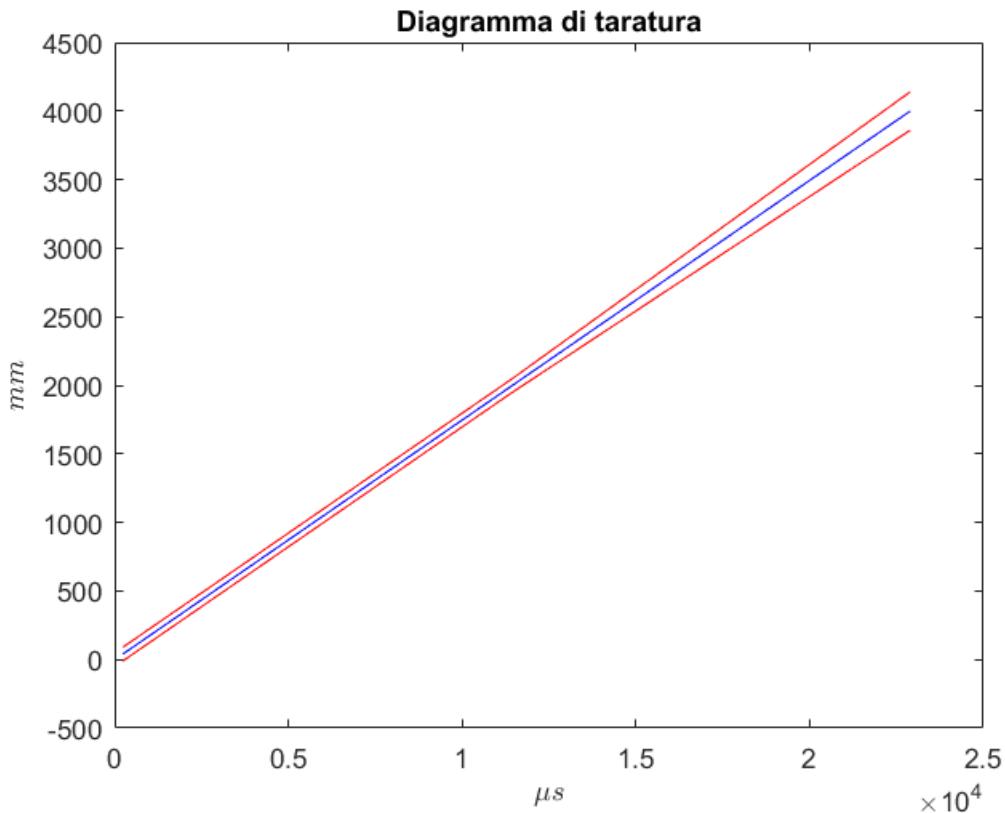
- Schema costruttivo:



- Schema elettrico:



- Diagramma di taratura:



Tensione di alimentazione	9 V
Portata	40 mm - 4000 mm
Sensibilità	5.72 μ s/mm
Equazione curva	$y = 0.1748x - 1.887$
Incertezza per distanza minore di 2 m (LC 99%)	5.1 mm
Incertezza per distanza maggiore di 2 m (LC 99%)	Variabile secondo (9.1) e (9.2)

Variazione incertezza distanza maggiore di 2 m

Superiore:

$$\begin{cases} y = 0.1748x + 3.213 & 240 < x < 11425 \\ y = 0.1756x - 5.695 & 11425 < x < 22894 \end{cases} \quad (9.1)$$

Inferiore:

$$\begin{cases} y = 0.1748x - 6.987 & 240 < x < 11425 \\ y = 0.1740x + 1.921 & 11425 < x < 22894 \end{cases} \quad (9.2)$$