



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

LAUREA TRIENNALE IN INGEGNERIA ELETTRONICA

Sviluppo di sensori di distanza basati su ultrasuoni e rilevazione ottica

LAUREANDO

De Lazzari Riccardo

Matricola 1227135

Data di Laurea 17/07/2023

RELATORE

Prof. Meneghini Matteo

Universita degli Studi di Padova

ANNO ACCADEMICO
2022/2023

*Alla mia famiglia
e a tutte le persone a me care*

Abstract tesi

L'elaborato prevederà lo sviluppo di sensori per la misurazione di distanze attraverso l'utilizzo di ultrasuoni e rilevazione ottica. Sarà presente la progettazione hardware e software e una parte di confronto delle tecnologie, con i relativi pro e contro di ognuna. In particolare, con l'utilizzo dell'infrarosso, il candidato dovrà progettare interamente la circuiteria necessaria per rendere il segnale idoneo all'acquisizione. Verrà testata l'accuratezza e precisione in termini di distanza e l'angolo massimo di misurazione. Si affronteranno eventuali criticità dei sensori, come per esempio l'influenza nella misurazione del disturbo che introduce la luce solare e artificiale nella rilevazione ottica. Saranno presenti inoltre delle considerazioni sull'utilizzo del tempo di volo e dell'ampiezza dell'onda come tecnica di calcolo della distanza. Dai test emergerà che ogni tecnologia ha il suo settore di utilizzo e che in termini di costi, anche stando in fascia molto bassa i risultati saranno generalmente molto buoni. Globalmente si troverà che il sensore a ultrasuoni sarà il migliore a parità di prezzo, mentre le altre due tecnologie presenteranno più punti deboli che si attenueranno passando a dei dispositivi più pregiati.

Contents

1	Introduzione	5
2	Tecnologia a ultrasuoni	7
2.1	Progettazione hardware	7
2.2	Progettazione software	10
2.3	Test e raccolta dati	13
2.3.1	Rilevazione pratica dell'accuratezza e precisione in funzione della distanza	14
2.3.2	Rilevazione pratica dell'angolo massimo di misura	16
3	Tecnologia a laser	19
3.1	Progettazione hardware	19
3.2	Progettazione software	22
3.2.1	Rilevazione pratica dell'accuratezza e precisione in funzione della distanza	24
3.2.2	Rilevazione pratica dell'angolo massimo di misura	27
4	Tecnologia a infrarossi	29
4.1	Progettazione hardware	29
4.1.1	Calcolo RB e RC	30
4.1.2	Circuito rilevazione in modalità continua	32
4.1.3	Circuito rilevazione in modalità a impulsi	33
4.1.4	Calcolo proporzionalità distanza - tensione d'uscita in modalità continua	35
4.1.5	Calcolo proporzionalità distanza - tensione d'uscita in modalità a impulsi	36
4.2	Progettazione software	38
4.3	Test e raccolta dati	41

CONTENTS

4.3.1	Rilevazione pratica dell'accuratezza e precisione in funzione della distanza	41
4.3.2	Rilevazione pratica dell'angolo massimo di misura	43
5	Confronto	45
6	Conclusioni	47
	Bibliografia	53



Introduzione

I sensori di distanza sono uno strumento molto utile al giorno d'oggi che agevolano e permettono di fare misurazioni, anche di notevole entità in modo rapido, semplice e preciso. Ne esistono di svariati tipi e in questo elaborato si vogliono studiare tre tecnologie diverse: *ultrasuoni, laser e infrarossi*.

L'acquisizione e l'elaborazione dei segnali verrà affidata alla scheda *Arduino Uno* con microcontrollore *ATMega 328P*. Per la restante componentistica circuitale saranno a disposizione dei moduli *Ready to use*, già pronti per essere interfacciati e utilizzati con la scheda attraverso linee di comunicazione standardizzate, per le tecnologie laser e ultrasuoni.

La progettazione del modulo a infrarossi sarà affidata al candidato, che, attraverso l'uso di un diodo infrarosso e un fotodiodo (componente che genera una corrente elettrica in proporzione alla radiazione infrarossa ricevuta) dovrà utilizzare tutti gli strumenti circuitali necessari a trasformare la quantità di radiazione in una grandezza compatibile (tensione elettrica) con *Arduino*.

I sensori *Ready to use* anche se pronti per essere collegati alla scheda necessitano di una configurazione che permette loro di lavorare secondo le specifiche dettate dall'utente.

Lo sviluppo software comprenderà sia la parte di settaggio dei moduli che la parte in cui il segnale elettrico ricevuto dal singolo modulo verrà elaborato e reso idoneo alla lettura da parte dell'utente.

Seguirà una parte di test per tutte e tre le tecnologie, dove si verificherà in modo pratico il corretto funzionamento dei singoli progetti, comprendenti sia il software che la componentistica hardware. Verranno poi raccolti i dati in merito

alle caratteristiche funzionali di ogni circuito da cui il candidato potrà evidenziare eventuali pro e contro delle varie tecnologie.



Tecnologia a ultrasuoni

La misurazione attraverso gli ultrasuoni sfrutta il tempo di volo delle onde sonore ad alta frequenza per calcolare la distanza da un determinato oggetto. In questo progetto verrà utilizzato il modulo *HC-SR04*, composto da un circuito di adattamento del segnale e due capsule, una responsabile della trasmissione dell'ultrasuono e una responsabile della ricezione dell'onda riflessa.

HC-SR04 correttamente configurato in fase di sviluppo software è in grado di restituire al programmatore il tempo di volo totale (andata + ritorno) dell'onda sonora generata dalla capsula trasmittitrice. La parte software garantirà l'elaborazione di questo dato per generare la visualizzazione chiara della distanza del sensore da un determinato oggetto.

2.1 PROGETTAZIONE HARDWARE

HC-SR04 possiede 4 pin di collegamento:

Lista dei componenti	
Descrizione	Codice
Arduino Uno	/
Modulo a ultrasuoni	HC-SR04
Breadboard	/
Cavi di collegamento	/

Table 2.1: Lista componenti ultrasuoni

2.1. PROGETTAZIONE HARDWARE

1. Vcc: viene collegato alla tensione di alimentazione da 3 a 5.5V.
2. Trig: è il pin Trigger che deve essere portato alto per inviare il segnale ad ultrasuoni.
3. Echo: è il pin che produce un impulso che si interrompe quando viene ricevuto il segnale riflesso dall'ostacolo.
4. GND: viene collegata la massa.

Vediamo nel dettaglio che linee ci fornisce *Arduino*:

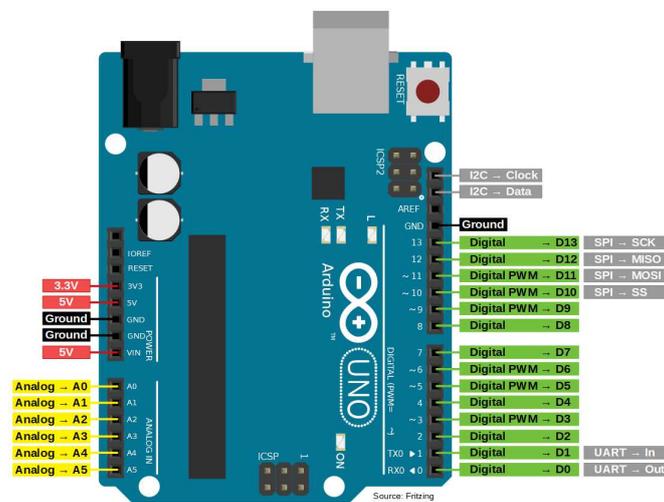


Figure 2.1: Dettaglio ingressi e uscite scheda Arduino

I pin D0 - D13 sono linee che possono assumere la funzione sia d'ingresso che d'uscita e sono digitali, per cui la tensione in output è impostabile solamente su due livelli (LOW = 0V e HIGH = VCC) ed è in grado di rilevare solamente segnali TTL se impostate come input. I pin analogici non saranno utilizzati.

Definizione 1 I segnali TTL sono segnali elettrici che portano una informazione di tipo binario. Il modo più comune è di portare queste informazioni e di associarle alla tensione. Se la tensione è elevata, si ha il livello alto (H), se è piccola abbiamo un livello basso (L). Nel caso di Arduino, leggerà un valore di tensione come alto se esso sarà almeno maggiore o uguale al 60% della tensione di alimentazione Vcc.

Per questo progetto saranno sufficienti le linee di alimentazione (VCC e GND) e due pin digitali che dovranno essere specificati nel software per permettere il corretto funzionamento.

Il collegamento è il seguente:

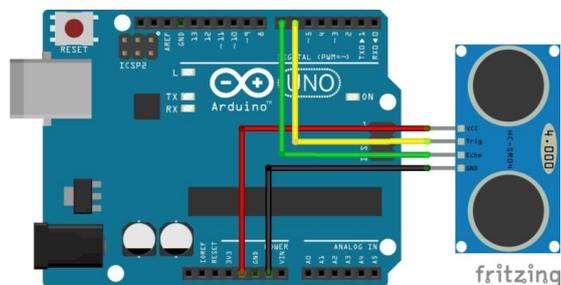


Figure 2.2: Collegamenti sensore ultrasuoni

1. Vcc: collegato al pin Vcc della scheda
2. Trig: pin D6
3. Echo: pin D7
4. GND: collegato al pin GND della scheda

Verrà utilizzata la *breadboard* (scheda forata che aiuta a eseguire i collegamenti tra componenti) per mettere il modulo in una posizione stabile adatta a eseguire le misurazioni nella fase di test.

2.2. PROGETTAZIONE SOFTWARE

Vediamo ora come funziona nel dettaglio il modulo *HC-SR04*:

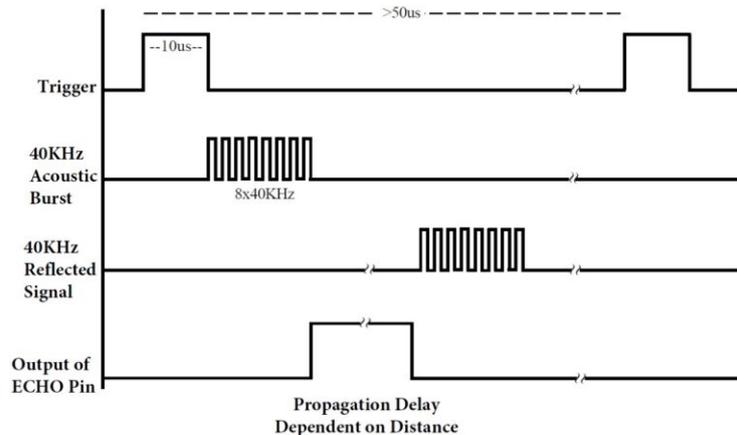


Figure 2.3: Diagramma dei tempi di un ciclo di misurazione

Il pin scelto come *Trigger* (pin D6) viene tenuto alto per 10 microsecondi. Da questo momento viene trasmessa una sequenza di impulsi a 40 KHz e ritroveremo subito il pin *Echo* a livello logico alto. Sarà sufficiente misurare il tempo che questo pin impiega per tornare basso e con la seguente formula calcolare la distanza misurata:

$$d = \frac{v * t}{2}$$

Dove v è la velocità del suono (343 m/s) e t il tempo misurato. La divisione per un fattore 2 deriva dal fatto che il tempo è complessivo di andata e ritorno dell'onda sonora.

2.2 PROGETTAZIONE SOFTWARE

Arduino impiega un linguaggio proprietario basato su C/C++ per dettare alla scheda le istruzioni da eseguire. Il codice è sostanzialmente composto da tre parti:

1. Sezione di inclusione delle librerie necessarie e inizializzazione delle eventuali componenti collegate alla scheda.
2. Sezione *setup*: segmento di codice che viene eseguito una sola volta, contenente settaggi delle modalità dei pin, variabili ed eventuali configurazioni dei componenti.

3. Sezione *loop*: questa parte contiene il vero e proprio programma dove vengono infinitamente eseguite le operazioni indicate al suo interno (es. calcolo distanza dal tempo misurato, ritardi, cicli for/while...).

Per visualizzare il dato finale utilizzeremo l'interfaccia seriale di Arduino.

Definizione 2 *La trasmissione seriale è una modalità di comunicazione tra dispositivi digitali nella quale i bit sono trasferiti lungo un canale di comunicazione uno di seguito all'altro e giungono sequenzialmente al ricevente nello stesso ordine in cui li ha trasmessi il mittente. Definendo la velocità (bit-rate) della trasmissione, Arduino è in grado di inviare dati in modo seriale al computer tramite appositi comandi che vengono visualizzati su una apposita finestra chiamata monitor seriale.*

Il codice Arduino è il seguente:

```
#define trigPin 6
#define echoPin 7

long durata, cm;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

2.2. PROGETTAZIONE SOFTWARE

```
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  durata = pulseIn(echoPin, HIGH);
  mm = durata * (343.1/2) * 1000

  Serial.print("Cm = ");
  Serial.println(cm);
  Serial.println();
  delay(60);
}
```

I comandi

```
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
```

definiscono la modalità di funzionamento dei pin digitali: il pin *trig* (6) sarà un'uscita e il pin *echo* (7) sarà un ingresso. Nella parte di *loop* abbiamo sequenzialmente

- I comandi `digitalWrite(trigPin, LOW)` e `delayMicroseconds(2)` sono utilizzati per porre il *trig* in uno stato di partenza definito.
- Con `digitalWrite(trigPin, HIGH)` si pone l'uscita ad un livello alto (fronte di salita dell'impulso).
- Si tiene questo pin alto per 10 microsecondi con `delayMicroseconds(10)` per poi farlo tornare ad un livello basso con `digitalWrite(trigPin, LOW)`.
- `durata = pulseIn(echoPin, HIGH)` serve ad immagazzinare nella variabile *durata* l'intervallo di tempo che intercorre tra quando il pin *echo* si pone alto dopo aver mandato l'impulso fino a quando torna basso perchè l'onda riflessa è stata ricevuta dalla capsula RX.

- Segue il calcolo della distanza utilizzando la formula definita in precedenza; per passare da metri a centimetri è necessario moltiplicare per un fattore 100.

Attraverso il programma apposito, verrà poi trasferito il codice nella scheda.

2.3 TEST E RACCOLTA DATI

Il datasheet del modulo *HC-SR04* ci fornisce dei dati importanti sulle caratteristiche del sensore:

Caratteristiche HC-SR04	
Distanza minima misurabile	2 cm
Distanza massima misurabile	400 cm
Massimo angolo rilevazione ostacolo	30°
Accuratezza	0.3 cm
Corrente di funzionamento	15 mA

Il limite inferiore di misurazione è dettato dal fatto che l'onda riflessa deve ritornare alla capsula ricevitrice solo dopo che la sequenza d'impulsi è stata inviata, la distanza troppo ridotta non darebbe tempo al modulo di passare in modalità di "ascolto".

Il limite superiore deriva dal fenomeno fisico di propagazione dell'onda: tanto più distante è l'oggetto, tanto più l'onda trasmessa si allargherà a ventaglio e a parità di potenza di trasmissione avremo che solo una minima frazione verrà riflessa. Anche nel percorso di ritorno ci sarà una propagazione a ventaglio, il che porterà ad un'ulteriore riduzione della potenza effettiva dell'onda che inciderà sul ricevitore. Per garantire l'accuratezza dichiarata, il costruttore dichiara un massimo di 4 metri.

L'area di lavoro prevederà la presenza di un oggetto mobile rispetto ad un punto fisso dove sarà posizionato il sensore. Sarà presente inoltre un secondo dispositivo che verrà utilizzato per avere un riferimento di distanza molto preciso (misuratore laser commerciale).

2.3. TEST E RACCOLTA DATI

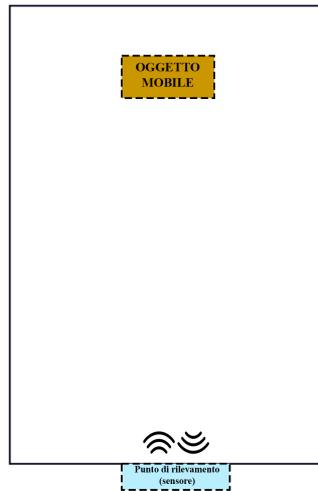


Figure 2.4: Area di lavoro per test

2.3.1 RILEVAZIONE PRATICA DELL'ACCURATEZZA E PRECISIONE IN FUNZIONE DELLA DISTANZA

Per la rilevazione di accuratezza e precisione si procederà alla cattura di 100 misurazioni consecutive a 5 distanze diverse, distribuite nel range di misura del sensore.

Con l'aiuto del programma excel, sarà poi possibile calcolare la precisione e l'accuratezza del sensore, sia globalmente che singolarmente per ogni lunghezza.

Definizione 3 *La precisione è definita come la media del valore assoluto della differenza di ogni misurazione con la media delle 100 misurazioni effettuate. L'accuratezza della singola misura, in percentuale è data da questa formula:*

$$Acc\% = 100 - \left(\frac{|Val_{effettivo} - Val_{misurato}|}{Val_{effettivo}} * 100 \right)$$

Per ottenere l'accuratezza totale sarà sufficiente fare la media delle singole accuratezze.

Si riportano i valori di precisione e accuratezza ottenuti dalle misurazioni e calcoli:

Sensore ultrasuoni 50 mm		Sensore ultrasuoni 175 mm	
Valore effettivo	50 mm	Valore effettivo	174,80 mm
Media	49,7419 mm	Media	174,54 mm
Precisione	0,44 mm	Precisione	0,145 mm
Accuratezza	99,29 %	Accuratezza	99,85 %

Sensore ultrasuoni 50 cm		Sensore ultrasuoni 1 mt	
Valore effettivo	500 mm	Valore effettivo	990 mm
Media	503,77 mm	Media	983,06 mm
Precisione	6,08 mm	Precisione	2,49 mm
Accuratezza	98,76 %	Accuratezza	99,29 %

Sensore ultrasuoni 4mt (max range)	
Valore effettivo	4000 mm
Media	4016,9 mm
Precisione	82,4 mm
Accuratezza	98,3 %

Risultati globali	
Precisione media	15,35 mm
Accuratezza media	99,1 %

Table 2.2: Test sensore ultrasuoni

I risultati ottenuti sono molto buoni considerando il prezzo e l'uso hobbystico a cui è destinato questo dispositivo; solo la misurazione a 4mt ha riportato, in proporzione, dei valori più distanti dal dichiarato.

Tutto ciò è comunque accettabile poichè ci si trova vicino al limite superiore del range dove è ragionevole pensare che la precisione e accuratezza diminuiscano drasticamente.

Non considerando la misurazione alla distanza limite la precisione diventa notevolmente più alta:

2.3. TEST E RACCOLTA DATI

Risultati globali (senza misura a 4mt)	
Precisione media	2,26 mm
Accuratezza media	99,3 %

2.3.2 RILEVAZIONE PRATICA DELL'ANGOLO MASSIMO DI MISURA

Per la rilevazione del massimo angolo di misura si procede a far avvicinare l'oggetto dal lato verso il centro dell'asse di misurazione finchè il sensore ne registra la presenza.

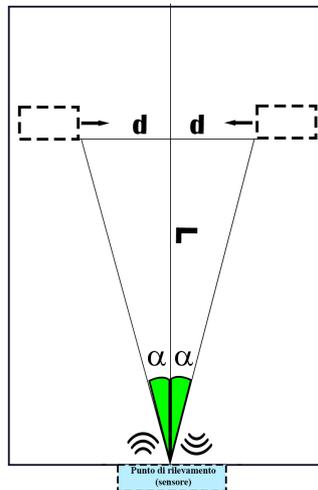


Figure 2.5: Area di lavoro per misurazione angolo massimo

Si esegue la stessa operazione sia da sinistra che da destra e successivamente misurando d e conoscendo L possiamo ricavare α :

$$\alpha = \arctan\left(\frac{d}{L}\right)$$

Si esegue questo test per 3 valori di L diversi ottenendo:

Angolo massimo di misura sensore ultrasuoni			
Distanza (L)	50 mm	100 mm	250 mm
Valore angolo SX	12.56°	11.20°	11.31°
Valore angolo DX	12.40°	11.53°	11.12°
Valore full cone (SX + DX)	24.96°	22.73°	22.43°

Table 2.3: Angolo massimo sensore ultrasuoni

3

Tecnologia a laser

La misurazione attraverso il modulo laser sfrutta il tempo di volo di un fascio di luce laser per determinare la distanza da un oggetto. In questo progetto verrà utilizzato il modulo *VL53L0X* che si interfaccia con *Arduino* tramite protocollo I²C.

Definizione 4 *Il protocollo di comunicazione I²C, scritto anche come I2C, è un protocollo di comunicazione seriale sincrono che permette a molti dispositivi di scambiarsi dati fra loro utilizzando solo due fili, uno per i dati ed uno per sincronizzare la comunicazione. È presente almeno un dispositivo master, responsabile della gestione e della ricezione dei dati e n dispositivi slave, responsabili dell'invio dati secondo le richieste del master.*

VL53L0X viene venduto già calibrato ma vengono fornite all'utente finale le istruzioni per un'eventuale ricalibrazione in caso di malfunzionamenti.

La libreria software che ci permette di utilizzare il modulo con *Arduino* ci restituisce il dato della distanza in millimetri già pronto per essere visualizzato. Ulteriori approfondimenti verranno fatti nella sezione 3.2

3.1 PROGETTAZIONE HARDWARE

Facendo riferimento al *datasheet*, il modulo in questione utilizza come emettitore un diodo laser con lunghezza d'onda pari a 940 nm, fuori dallo spettro visibile all'occhio umano, mentre per la rilevazione del raggio riflesso utilizza uno *SPAD*.

3.1. PROGETTAZIONE HARDWARE

Lista dei componenti	
Descrizione	Codice
Arduino Uno	/
Modulo a laser	VL53L0X
Breadboard	/
Cavi di collegamento	/

Table 3.1: Lista componenti laser

Features

- Fully integrated miniature module
 - 940 nm laser VCSEL
 - VCSEL driver

The VL53L0X integrates a leading-edge SPAD array (single photon avalanche diodes) and embeds ST's second generation FlightSense patented technology.

Figure 3.1: Specifica emettitore laser

Figure 3.2: Dettaglio ricevitore laser

Definizione 5 *I rivelatori microelettronici SPAD Single-Photon Avalanche Diodes sono componenti in grado di dare impulsi elettrici (solitamente in corrente) macroscopici in risposta a singoli fotoni, utilizzando la moltiplicazione a valanga per ottenere un funzionamento a scatto.*

VL53L0X possiede 6 pin di collegamento:

1. Vin: viene collegato alla tensione di alimentazione da 3 a 5.5V.
2. GND: viene collegata la massa.
3. SCL: è il pin Trigger che deve essere portato alto per inviare il segnale ad ultrasuoni.
4. SDA: è il pin che produce un impulso che si interrompe quando viene ricevuto il segnale riflesso dallostacolo.
5. GPIO1: Pin di interrupt (non utilizzato)
6. XSHUT: non utilizzato.

Sempre con riferimento al datasheet osserviamo che il modulo può misurare in 4 modalità diverse. Oltre ad un modo *standard*, quello *long range* ci permette di

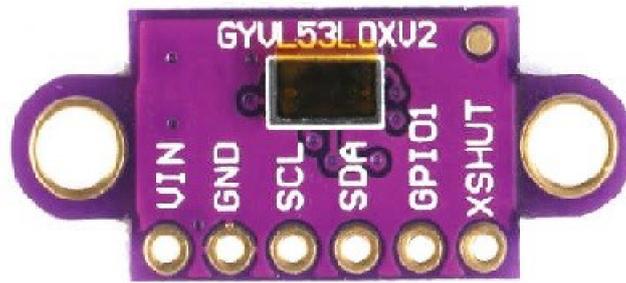


Figure 3.3: Pinout modulo VL53L0X

misurare distanze fino a 2m (contro gli 1.2m dello standard); la modalità *high speed* effettua misurazioni molto veloci diminuendo però l'accuratezza mentre quella *high accuracy* aumenta la precisione a discapito del tempo. Per la prova pratica dell'accuratezza dei vari profili utilizzeremo la *misurazione temporizzata*, che ci permetterà di raccogliere i valori in modo preciso e con discreta velocità. Vediamo i collegamenti:

1. Vin: collegato al pin Vcc della scheda.
2. GND: collegato al pin Gnd della scheda
3. SCL: Pin A5
4. SDA: Pin A4
5. GPIO1: Non collegato.
6. XSHUT: Non collegato.

I pin A4 e A5 sono per costruzione le linee di *Arduino Uno* dedicate alla comunicazione I²C.

Board	I2C/TWI pins
UNO, Ethernet	A4 (SDA), A5 (SCL)

Figure 3.4: Riferimento pin I2C

3.2 PROGETTAZIONE SOFTWARE

Per realizzare il programma utilizzeremo l'apposita libreria di supporto che fornisce tutte le API per l'utilizzo del modulo. Il codice è il seguente:

```
#include <Wire.h>
#include <VL53L0X.h>

VL53L0X sensor;

//#define LONG_RANGE
//#define HIGH_SPEED
//#define HIGH_ACCURACY

void setup()
{
  Serial.begin(9600);
  Wire.begin();

  sensor.setTimeout(500);
  if (!sensor.init())
  {
    Serial.println("Failed to detect and initialize sensor!");
    while (1) {}
  }

  #if defined LONG_RANGE
    // lower the return signal rate limit (default is 0.25 MCPS)
    sensor.setSignalRateLimit(0.1);
    // increase laser pulse periods (defaults are 14 and 10 PCLKs)
    sensor.setVcselPulsePeriod(VL53L0X::VcselPeriodPreRange, 18);
    sensor.setVcselPulsePeriod(VL53L0X::VcselPeriodFinalRange, 14);
  #endif
}
}
```

```

#if defined HIGH_SPEED
    // reduce timing budget to 20 ms (default is about 33 ms)
    sensor.setMeasurementTimingBudget(20000);
#elif defined HIGH_ACCURACY
    // increase timing budget to 200 ms
    sensor.setMeasurementTimingBudget(200000);
#endif
}

void loop()
{
    Serial.print(sensor.readRangeSingleMillimeters());
    if (sensor.timeoutOccurred()) { Serial.print(" TIMEOUT"); }

    Serial.println();
}
}

```

All'inizio è presente l'inclusione della libreria con annessa inizializzazione del componente *VL53L0X* e una parte in cui è possibile attivare singolarmente i profili di misura in base alla riga su cui effettuiamo l'operazione di *uncomment*.

La parte di *setup* contiene l'inizializzazione della comunicazione seriale per la visualizzazione sul monitor virtuale mentre il comando

```
Wire.begin();
```

è necessario per far funzionare la comunicazione I2C tra la scheda Arduino e il sensore. La restante parte di *setup* regola i parametri interni del modulo a seconda del profilo selezionato (come intensità minima del segnale di ritorno, larghezza degli impulsi e tempo disponibile per la misurazione).

```

///define LONG_RANGE
///define HIGH_SPEED
///define HIGH_ACCURACY

```

Se nessuna di queste 3 righe è trasformata da commento a codice il sensore opererà in modalità di misura standard. La parte funzionale del loop è composta essenzialmente dal comando

3.2. PROGETTAZIONE SOFTWARE

```
Serial.print(sensor.readRangeSingleMillimeters());
```

che effettua la stampa del valore della distanza in millimetri rilevata dal sensore richiamando l'apposita funzione della libreria. Passato un certo tempo definito in cui il sensore non riceve nessun segnale di ritorno il programma stampa un TIMEOUT sul monitor seriale.

3.2.1 RILEVAZIONE PRATICA DELL'ACCURATEZZA E PRECISIONE IN FUNZIONE DELLA DISTANZA

Per la rilevazione di accuratezza e precisione si procederà alla cattura di 100 misurazioni consecutive a 5 distanze diverse, distribuite nel range di misura del sensore.

In particolare per questo modulo laser verranno testate le sue performance sia in condizioni di buio che in presenza di luce solare e/o artificiale, data la sensibilità ad esse.

La modalità *long range* verrà testata separatamente in quanto specifica solamente per le distanze più lunghe.

DISTANZA 50 mm

Sensore laser 50mm (condizioni di buio)			
	Mod. Standard	Mod. High speed	Mod. High accuracy
Valore effettivo	50 mm		
Media	50,22 mm	52,5 mm	50,06 mm
Precisione	1,31 mm	1,73 mm	0,47 mm
Accuratezza	97,44 %	94,28 %	99,12 %

Sensore laser 50mm (condizioni di luce)			
	Mod. Standard	Mod. High speed	Mod. High accuracy
Valore effettivo	50 mm		
Media	49,96 mm	53,34 mm	50,80 mm
Precisione	1,24 mm	1,69 mm	0,64 mm
Accuratezza	97,52 %	93,28 %	98,06 %

DISTANZA 140 mm

Sensore laser 140mm (condizioni di buio)			
	Mod. Standard	Mod. High speed	Mod. High accuracy
Valore effettivo	140 mm		
Media	142,3 mm	148,96 mm	140,5 mm
Precisione	1,44 mm	1,66 mm	0,9 mm
Accuratezza	98,24 %	94,88 %	99,10 %

Sensore laser 175mm (condizioni di luce)			
	Mod. Standard	Mod. High speed	Mod. High accuracy
Valore effettivo	175 mm		
Media	144,7 mm	149,62 mm	143,4 mm
Precisione	1,54 mm	2,09 mm	1,53 mm
Accuratezza	96,7 %	94,5 %	97,5 %

DISTANZA 50 cm

Sensore laser 50 cm (condizioni di buio)			
	Mod. Standard	Mod. High speed	Mod. High accuracy
Valore effettivo	500 mm		
Media	508,62 mm	511,49 mm	499,41 mm
Precisione	2,73 mm	4,60 mm	0,70 mm
Accuratezza	98,3 %	97,82 %	99,7 %

Sensore laser 50 cm (condizioni di luce)			
	Mod. Standard	Mod. High speed	Mod. High accuracy
Valore effettivo	500 mm		
Media	511,38 mm	510,36 mm	497,97 mm
Precisione	7,75 mm	5,20 mm	1,61 mm
Accuratezza	97,58 %	97,87 %	99,53 %

DISTANZA 1 mt

Sensore laser 1 mt (condizioni di buio)			
	Mod. Standard	Mod. High speed	Mod. High accuracy
Valore effettivo	1000 mm		
Media	993,23	912,15 mm	1000,28 mm
Precisione	2,54 mm	12,98 mm	0,8 mm
Accuratezza	98,69 %	91,21 %	97,33 %

3.2. PROGETTAZIONE SOFTWARE

Sensore laser (riepilogo generale)	
Precisione	2,94 mm
Accuratezza	98,23 %

Table 3.2: Test sensore laser

Sensore laser 1 mt (condizioni di luce)			
	Mod. Standard	Mod. High speed	Mod. High accuracy
Valore effettivo	1000 mm		
Media	998,36 mm	914,31 mm	1000,43 mm
Precisione	2,94 mm	21,58 mm	1,45 mm
Accuratezza	98,23 %	91,43 %	97,5 %

Sensore laser condizioni di buio (recap modalità)			
	Mod standard	Mod. High Speed	Mod. High Accuracy
Precisione	2,01 mm	5.24 mm	0.71 mm
Accuratezza	98,16 %	94,54 %	98.82%

Sensore laser condizioni di luce (recap modalità)			
	Mod standard	Mod. High Speed	Mod. High Accuracy
Precisione	3,36 mm	7,64 mm	1,30 mm
Accuratezza	97,5 %	94,27 %	98,2%

MODALITA' LONG RANGE

Sensore laser 1,90 mt (condizioni di buio)	
	Mod. Long Range
Valore effettivo	1900 mm
Media	1802,5 mm
Precisione	6,46 mm
Accuratezza	94,84 %

Globalmente il risultato è buono, visto il costo del sensore (c.ca 7€) ma presenta comunque instabilità nelle misurazioni. Certe catture sono state ripetute in quanto i risultati erano totalmente alterati. Numerose volte il modulo andava in timeout e faceva difficoltà nell'iniziare la comunicazione I2C. Si può notare nella tabella di recap come le diverse modalità facciano la differenza su precisione e accuratezza

e anche la miglior performance in condizioni di buio. Nonostante il costruttore dichiarò 200 cm come distanza massima in modalità *long range*, la verità è che effettuando le misure la maggior parte delle volte si ricadeva in un timeout o in un dato falsato. Per ottenere delle misure accettabili si è dovuto ridurre la distanza a 190 cm.

3.2.2 RILEVAZIONE PRATICA DELL'ANGOLO MASSIMO DI MISURA

La procedura è la medesima del sensore a ultrasuoni:

Angolo massimo di misura sensore laser			
Distanza (L)	50 mm	300 mm	1000 mm
Valore angolo SX	10.34°	11.10°	9.51°
Valore angolo DX	10.40°	11.22°	9.12°
Valore full cone (SX + DX)	20.74°	22.32°	18.63°

Table 3.3: Angolo massimo di misura sensore laser

I valori sono globalmente minori rispetto all'ultrasuoni, ciò dovuto al fatto la diffusione del laser è focalizzata su un'area più stretta.



Tecnologia a infrarossi

La misurazione con infrarossi utilizza la rilevazione ottica di un fascio di luce IR per determinare la distanza di un oggetto. Andrà progettato il circuito responsabile della generazione della luce infrarossa e della rilevazione della radiazione di ritorno.

A differenza della tecnologia a laser che sfrutta il tempo di volo, questo sensore utilizzerà l'ampiezza di un opportuno segnale fornito in ingresso ad *Arduino* per determinare la distanza dell'oggetto.

Verranno sviluppate due versioni, la cui differenza sarà la minor sensibilità al rumore luminoso ambientale con lo scopo di migliorare l'utilizzabilità del sensore.

4.1 PROGETTAZIONE HARDWARE

La base del progetto è un diodo a infrarossi, modello *OSRAM SFH 4550-S*, responsabile della generazione fascio IR e un fotodiodo, modello *OSRAM SFH 203*, in grado di generare una corrente in base alla radiazione ricevuta.

E' necessario sviluppare un circuito che permetta a questi due componenti di interfacciarsi con la scheda *Arduino* in modo da poter effettuare una misurazione della distanza.

4.1. PROGETTAZIONE HARDWARE

Lo schema elettrico è il seguente:

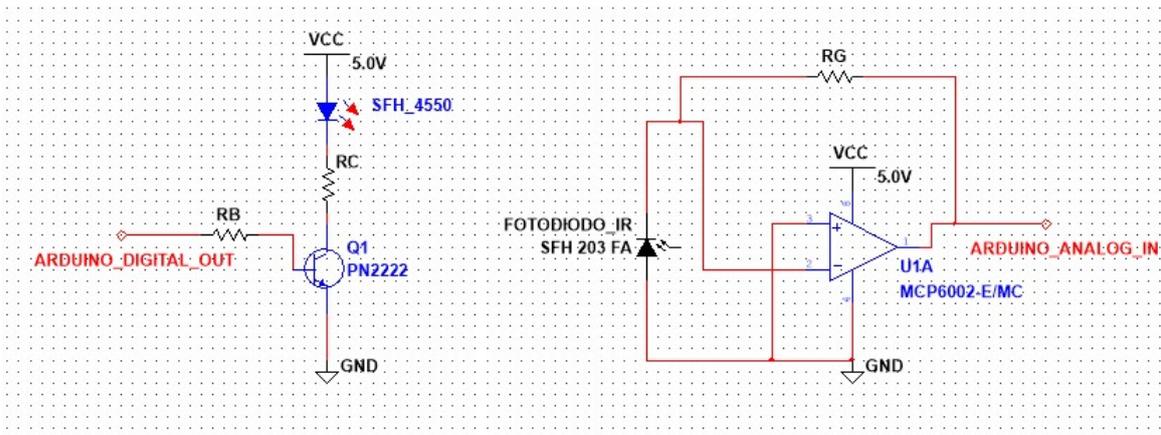


Figure 4.1: Schema elettrico circuito sensore infrarossi

Per la parte di generazione della luce infrarossa si utilizza un amplificatore a transistor BJT con resistenza di base R_B e resistenza di collettore R_C . Il tutto è alimentato a 5V dalla scheda *Arduino*.

4.1.1 CALCOLO R_B E R_C

Osservando la scheda tecnica del diodo possiamo notare che presenta una corrente massima di funzionamento pari a 100 mA .

Forward current	I_F	max.	100 mA
Forward current pulsed $t_p \leq 400\ \mu\text{s}; D \leq 0.005$	$I_{F\text{ pulse}}$	max.	1 A

Figure 4.2: Corrente massima da datasheet

Tuttavia, i dati presenti nel foglio tecnico ci fanno intendere che il dispositivo è progettato per lavorare in modo più efficiente a impulsi. Dobbiamo perciò limitare la corrente quando si utilizza il led in modo continuo; imponiamo quindi una I_D pari a circa 30 mA . Possiamo notare dalla figura sottostante che la caduta di tensione sul diodo rispetto alla corrente è di circa 1,5 V.

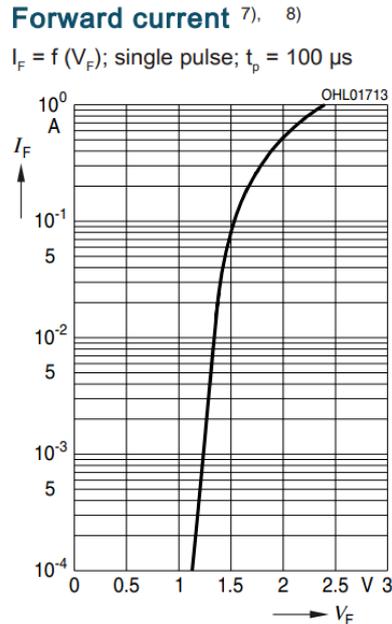


Figure 4.3: Tensione del diodo in base alla corrente di lavoro

Supponendo il transistor in saturazione, con riferimento al datasheet $V_{CEsat} \approx 0.3V$, $V_{BEsat} \approx 0.6V$ e $hFE \approx 100$

$$V_{CE} = V_{CC} - V_{LED} - RC * I_C > V_{CEsat}$$

$$RC < \frac{V_{CC} - V_L - V_{CEsat}}{I_C} < 110\Omega$$

Scegliamo $RC = 100\Omega$ e otteniamo, rieffettuando il calcolo una $I_C = 33mA$.

$$I_B > \frac{I_C}{\beta_F} > 33nA$$

$$R_B < \frac{V_{outarduino} - V_{BEsat}}{I_B} < 13333\Omega$$

Scegliamo $R_B = 10K\Omega$. Verifichiamo inoltre che la potenza dissipata dalla resistenza RC sia accettabile:

$$P = R_C * I_C^2 = 109mW$$

Ben al di sotto del limite per queste resistenze, $P_{MAX} = 250mW$

4.1. PROGETTAZIONE HARDWARE

Lavorando a impulsi e riferendoci al datasheet di figura 4.2 possiamo imporre una corrente decisamente più alta in modo da migliorare il rapporto segnale-rumore in ricezione. Supponendo $I_C \approx 300mA$ otteniamo

$$R_C = \frac{V_{CC} - V_L - V_{CEsat}}{I_C} = 11\Omega$$

Scegliamo $R_C = 12\Omega$ e ricalcolando la corrente risulta $I_C = 275mA$

$$I_B > \frac{I_C}{\beta_F} > 2.75mA$$

$$R_B < \frac{V_{outarduino} - V_{BEsat}}{I_B} < 1709\Omega$$

Scelgo $R_B = 1500\Omega$

Verifichiamo nuovamente che la dissipazione della resistenza R_C sia accettabile:

$$P = R_C * I_C^2 = 908mW$$

Tuttavia la potenza media dissipata sarà minore al valore trovato poichè non stiamo applicando un segnale continuo, bensì un'onda quadra. La potenza media dissipata sarà

$$P_{DISS} = R_C * (I_{RMS})^2 = R_C * \left(\frac{I_C}{\sqrt{2}}\right)^2 = 453mW$$

Possiamo scegliere un resistore con $P_{max} = 500mW$.

Il valore di I_C imposto avrebbe potuto essere ancora più alto osservando i valori massimi di funzionamento del diodo, tuttavia questo avrebbe comportato una dissipazione eccessiva per la resistenza R_C (si sarebbe dovuto utilizzare una resistenza fisicamente più grande andando a penalizzare la compattezza del circuito e di valore molto piccolo) e il transistor. Esiste inoltre un limite di corrente massima erogabile dalla porta USB a cui è collegata la scheda che si aggira intorno ai $\approx 350 - 400mA$.

4.1.2 CIRCUITO RILEVAZIONE IN MODALITÀ CONTINUA

Il circuito per la rilevazione della radiazione in figura 4.1 è composto dal fotodiode e da un convertitore corrente tensione fatto con amplificatore operazionale MCP6002. Questo modello in particolare permette di lavorare con tensione di al-

imentazione singola (0-5V) e quindi di poterlo collegare direttamente alla scheda *Arduino*.

L'uscita sarà

$$V_O = -I_F * R_G$$

Questo circuito ha il difetto di essere sensibile ai disturbi luminosi ambientali che verranno anch'essi amplificati dallo stadio con operazionale. Le performance saranno quindi molto scarse e sarà praticamente inutilizzabile se non in condizioni di scarsissima illuminazione.

4.1.3 CIRCUITO RILEVAZIONE IN MODALITÀ A IMPULSI

Nell'ambiente di lavoro può capitare che ci siano fonti di radiazione vicine alla lunghezza d'onda dell'infrarosso che contribuiscono ad immettere del rumore all'interno del circuito.

Per ovviare a questo problema è necessario inserire un condensatore C2 da 1uF dopo il fotodiodo in modo da eliminare l'eventuale componente continua indesiderata introdotto dall'ambiente e programmare *Arduino* in modo che la luce IR sia mandata a impulsi con frequenze dell'ordine dei 5 KHz. Questo valore rappresenta un limite dettato dall'istruzione di *Arduino* per catturare un dato analogico. Per eseguire una lettura infatti, sono necessari circa $T_i = 100\mu S$, quindi volendo acquisire il valore nella parte alta dell'impulso, si avrà che

$$F_{max} = \frac{1}{2 * T_i} = 5KHz$$

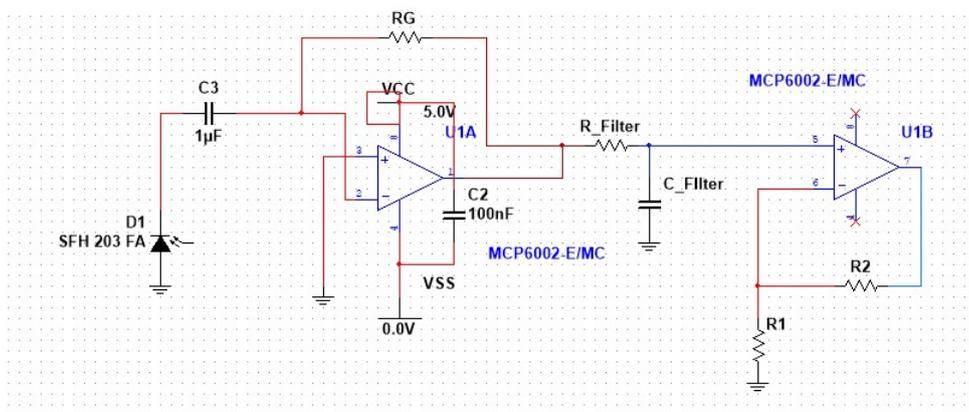


Figure 4.4: Schema elettrico circuito sensore infrarossi con filtraggio disturbi

4.1. PROGETTAZIONE HARDWARE

La presenza del doppio stadio è giustificata dal filtro passa basso che sta in mezzo. Esso serve a ridurre le componenti in alta frequenza indesiderate per aiutare a isolare ulteriormente il segnale utile. Successivamente il secondo stadio porterà il segnale a livello corretto per la lettura.

Inglobando il filtraggio nel convertitore, data la grande resistenza R_G (i fotodiodi sono in grado di generare correnti dell'ordine dei uA quindi anche se non noto, il valore di R_G sarà comunque molto elevato), si dovrebbe utilizzare un condensatore molto piccolo per ottenere frequenze di taglio compatibili (dell'ordine delle decine di KHz), ma ciò comporterebbe che il valore di capacità sia comparabile con le capacità parassite che ci sono sulla breadboard o la capacità equivalente del fotodiode ($\approx 6 - 8pF$) e questo andrebbe ad alterare in modo negativo le performance di filtraggio.

Il circuito RC in figura 4.4 invece ci permette di poter scegliere un condensatore di valore proporzionalmente molto più grande, escludendo il rischio di alterazione delle performance dato dalle capacità parassite.

Lavorando con impulsi a circa 5 KHz imponiamo una frequenza di taglio di 50 KHz per evitare di filtrare eccessivamente il nostro segnale utile. Risulta quindi

$$F_{cut} = \frac{1}{2 * \pi * R_{filter} * C_{Filter}} = 50KHz$$

Scegliendo un condensatore di valore pari a 10nF otteniamo

$$R_{filter} = \frac{1}{2 * \pi * C_{Filter} * F_{cut}} = 318\Omega$$

Scegliamo $R_{filter} = 330\Omega$ e risulta

$$F_{cut} = \frac{1}{2 * \pi * R_{filter} * C_{Filter}} = 48,2KHz$$

Segue poi un secondo stadio non invertente, le cui resistenze verranno calcolate successivamente insieme a R_G quando sarà nota la corrente generata dal fotodiode durante la ricezione dell'impulso luminoso.

Il condensatore C3 è richiesto da specifica del datasheet dell'operazionale.

4.1.4 CALCOLO PROPORZIONALITÀ DISTANZA - TENSIONE D'USCITA IN MODALITÀ CONTINUA

Un punto fondamentale per il funzionamento del sensore è associare la distanza di misurazione al valore di tensione che fornisce il circuito tramite una legge. Per esigenze costruttive imponiamo che il sistema possa rilevare un minimo di 3cm.

A questa distanza la I_D misurata è pari a $-2,62\mu A$, sapendo che il convertitore tensione-corrente restituisce un valore pari a

$$V_O = -R_G * I_D$$

imponiamo che la V_O debba essere uguale a 5V quando la distanza è minima per sfruttare tutto il range a disposizione. Di conseguenza avremo che:

$$R_G = -\frac{V_O}{I_{DMAX}} = 1.9M\Omega$$

Scegliamo $R_G = 1.8M\Omega$, per non incorrere nella saturazione dell'operazionale e perdere parte del range di misurazione. Effettuando la misurazione di V_O a delle distanze di riferimento otteniamo una tabella con grafico associato da cui ricaviamo la formula per il calcolo della distanza effettuando un fit dei punti. Il valore inserito rappresenta già la lettura della tensione eseguita da Arduino, con scala 0-1023.

4.1. PROGETTAZIONE HARDWARE

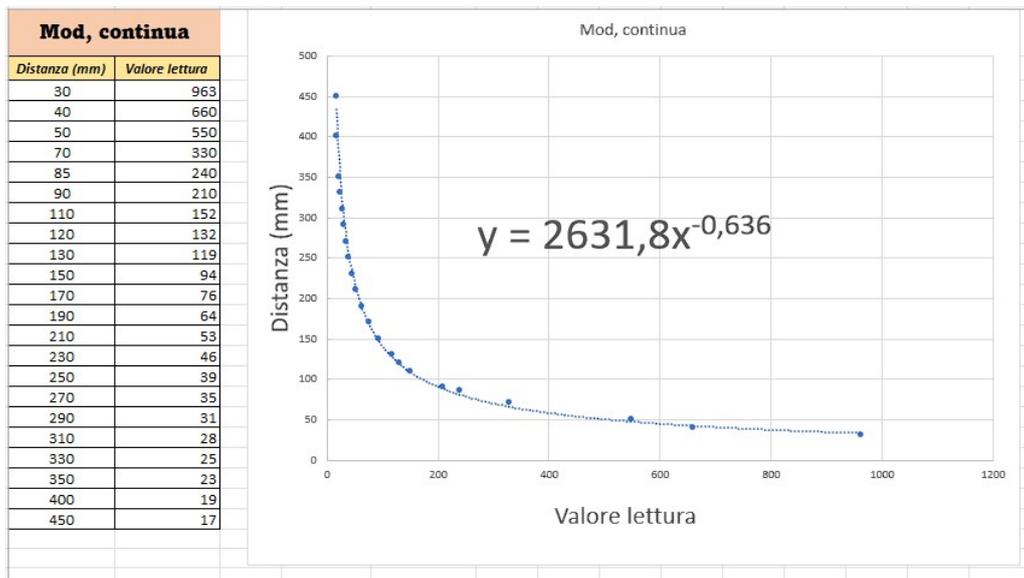


Figure 4.5: Tabella e grafico con fit dei dati

L'equazione mostrata ci permetterà di calcolare la distanza in base al valore che Arduino leggerà dalla porta analogica.

4.1.5 CALCOLO PROPORZIONALITÀ DISTANZA - TENSIONE D'USCITA IN MODALITÀ A IMPULSI

Misurando ora I_D alla minima distanza (3 cm) risulta $\approx -20\mu A$.

E' opportuno che la R_G faccia risultare $V_{O1} \ll 5V$ poichè eventuali componenti di frequenza indesiderate non andranno eccessivamente amplificate mantenendo un buon rapporto segnale-rumore.

Imponiamo che V_{O1} debba essere 500mV alla minima distanza. Risulta quindi:

$$R_G = \frac{V_{O1}}{-I_{D_{MAX}}} = 25K\Omega$$

Scegliamo $R_B = 27K\Omega$, quindi $V_{O1} = -I_{D_{MAX}} * R_G = 0.54V$

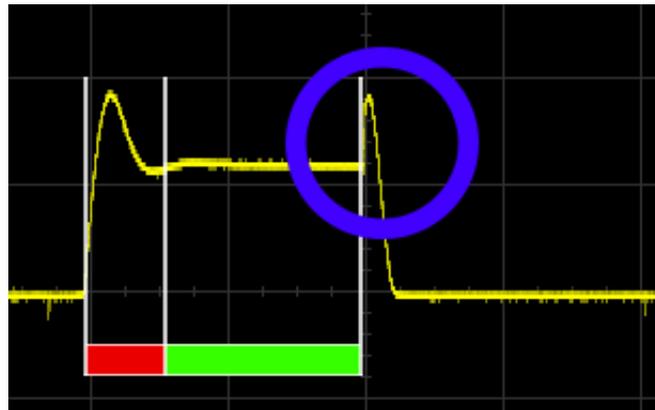


Figure 4.6: Dettaglio impulso post-filtraggio catturato da oscilloscopio

Inserendo il sistema passa basso e testando in modo pratico questa prima parte di circuito possiamo notare che è presente un fenomeno di overshoot. Questo è dato dalla banda limitata dell'operazionale che genera sovraelongazione nella risposta al gradino e dalla performance non ottimale del sistema di filtraggio, poichè la catena RC in figura 4.4 è del primo ordine. Si può scegliere di ridurre la frequenza di taglio sacrificando l'integrità dell'impulso quadro ma eliminando la sovraelongazione oppure passare ad un sistema attivo di secondo ordine, ma ciò implicherebbe una minor compattezza del circuito dovendo aggiungere altri componenti anche se comunque il fenomeno dell'overshoot è principalmente causato dall'operazionale. E' possibile bypassare in modo semplice ma comunque efficace questo problema iniziando l'acquisizione all'inizio del periodo verde visualizzato nell'immagine 4.7. Considerando che ogni divisione della griglia dei tempi è pari a 50 uS/div, la sezione rossa è circa uguale a 25 uS ed è la parte che è necessario saltare prima di iniziare l'acquisizione. Essendo la durata dell'impulso vincolata al tempo di lettura del dato ($\approx 100\mu S$) si avrà che aggiungendo un'istruzione di delay risulterà:

$$T'_i = t_{conversion} + t_{delay} = 100\mu S + 25\mu S = 125\mu S$$

Di conseguenza la frequenza di lavoro si ridurrà a:

$$F'_{max} = \frac{1}{2 * T'_i} = 4KHz$$

La porzione cerchiata in blu in figura 4.7 non altera la lettura del dato poichè in

4.2. PROGETTAZIONE SOFTWARE

quel momento la scheda ha già terminato l'acquisizione (istante in cui Arduino ha inviato il comando di spegnimento del led trasmettitore).

Calcoliamo $R1$ e $R2$ del secondo stadio:

$$A_v = \frac{V_{OMAX}}{V_{O1}} = 1 + \frac{R2}{R1} = \frac{5}{0.54} = 9.25$$

Scegliendo $R1$ pari a 15 Kohm risulta:

$$R2 = (A_v - 1) * R1 = 122K\Omega$$

Scegliamo quindi $R2 = 120K\Omega$. Ora che abbiamo dimensionato tutti i componenti eseguiamo nuovamente la procedura di fit dei dati:

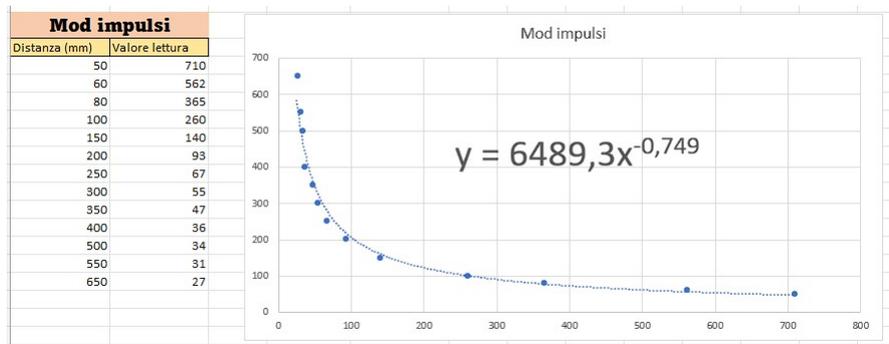


Figure 4.7: Tabella e grafico con fit dei dati modalità a impulsi

Anche in questo caso l'equazione mostrata ci permetterà di calcolare la distanza in base al valore che Arduino leggerà dalla porta analogica.

E' necessario considerare che tutto il processo di dimensionamento e taratura è stato effettuato con un oggetto a bassa riflettività. Utilizzando una superficie più lucida o più chiara, l'intensità del raggio riflesso aumenterà. In questo caso si avrà un aumento della distanza minima, per non incorrere nella saturazione positiva del circuito. Questo fattore verrà valutato nella sezione 4.3

4.2 PROGETTAZIONE SOFTWARE

Imponiamo che il pin D5 sia l'uscita per il comando del diodo mentre il pin analogico A0 l'ingresso per il segnale del fotodiode.

Il codice per il funzionamento in modalità continua è il seguente:

```

#define LED 5
#define RECEIVER A0

int distanza;

void setup() {
  Serial.begin(9600);
  pinMode(LED,OUTPUT);
  pinMode(RECEIVER,INPUT);
  digitalWrite(LED,HIGH);
}

void loop() {
  distanza=(2631,8 * pow((distanza),-0.636));
  Serial.println(distanza);
  delay(50);
}

```

Nel setup si inizializza la comunicazione seriale per la visualizzazione sul monitor virtuale e si imposta il pin D5 come OUTPUT e il pin A0 come INPUT. Successivamente si accende il LED con l'istruzione

```
digitalWrite(LED,HIGH);
```

Nella parte di loop si fa il calcolo della distanza utilizzando la formula ricavata con il fit del grafico e si stampa il valore sul monitor seriale inserendo poi un delay di 50ms solo per comodità di lettura dei dati.

Vediamo il codice leggermente più complesso per la versione migliorata a infrarossi:

4.2. PROGETTAZIONE SOFTWARE

```
#define LED 5
#define RECEIVER A0

int distanza[8];
int media;
long mediaDistanza;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(RECEIVER, INPUT);
  digitalWrite(LED, LOW);
  digitalWrite(5, LOW);
}
void loop(){
  mediaDistanza=getDistance();
  Serial.println(mediaDistanza);
  delay(50);
}
```

```
int getDistance() {
  // put your main code here, to run repeatedly:
  for (int i = 0; i < 8; i++) { //pulse at 4KHz
    digitalWrite(LED, HIGH); //Accensione led IR - fronte salita impulso
    delayMicroseconds(25); //Delay per lettura corretta e stabile del dato
    distanza[i] = analogRead(RECEIVER); // Lettura effettiva del dato
    digitalWrite(LED, LOW); //Spegnimento led IR - fronte discesa impulso
    delayMicroseconds(125); //Delay rendere il segnale un impulso stabile
    // con duty al 50% (100 uS lettura + 25 uS delay)
  }
  media=0;
  for (int i = 0; i < 8; i++) { //media letture
    media=media+distanza[i];
  }
  return(6489.5 * pow((media/8),-0.749)); //return del dato con calcolo
    // distanza tramite formula
}
```

Oltre alla parte di setup simile al codice precedente, per comodità è stata realizzata una funzione separata che restituisce come dato finale la distanza.

Vediamo il dettaglio di `int getDistance()`:

Il primo ciclo `for` serve a generare gli 8 impulsi con annessa acquisizione del dato posizionandolo dentro un vettore. Le istruzioni di `delay` sono necessarie per rendere la parte alta e la parte bassa dell'impulso uguali secondo i calcoli eseguiti in precedenza. Successivamente si fa la media delle 8 acquisizioni e si restituisce la distanza secondo la formula calcolata con il fit dei dati.

Nella parte di `loop` si esegue la semplice chiamata alla funzione.

4.3 TEST E RACCOLTA DATI

Le performance del sensore in modalità continua sono abbastanza mediocri, rendendo la misurazione sostanzialmente impossibile con la presenza di luce. A scopo di confronto eseguiamo comunque le rilevazioni.

4.3.1 RILEVAZIONE PRATICA DELL'ACCURATEZZA E PRECISIONE IN FUNZIONE DELLA DISTANZA

La procedura di test è la medesima degli altri sensori, per ottenere dei risultati un minimo accettabili si lavora al buio:

Riepilogo performance sensore infrarossi modalità continua						
Distanza	50 mm	100 mm	175 mm	300 mm	400 mm	600 mm
Media	49 mm	96 mm	173 mm	304 mm	409 mm	552 mm
Precisione	< 1 mm	< 1 mm	< 1 mm	< 1 mm	< 1 mm	< 1 mm
Accuratezza (%)	98 %	96 %	98,85 %	98,66 %	97,75 %	92 %

Table 4.1: Riepilogo performance sensore infrarossi modalità continua

Come si osserva dalla tabella il dato risulta molto stabile con ottima precisione, ma l'accuratezza diminuisce molto dopo i 400 mm.

Anche se nel resto del range, la performance non è scarsa, come detto in precedenza il sensore è sostanzialmente inutilizzabile in presenza di qualsiasi fonte di luminosa nell'ambiente.

Seguiamo lo stesso iter per la versione migliorata del rilevatore, questa volta

4.3. TEST E RACCOLTA DATI

posizionandosi in un ambiente con illuminazione mista (luce artificiale e/o solare):

Performance sensore infrarossi modalità impulsata			
Distanza	50 mm	100 mm	175 mm
Media	48.01 mm	97.96 mm	173.57 mm
Precisione	< 1 mm	< 1 mm	< 1 mm
Accuratezza (%)	96,02 %	97,97 %	99,18%

Table 4.2: Performance sensore infrarossi modalità impulsata 1

Performance sensore infrarossi modalità impulsata			
Distanza	300 mm	400 mm	600 mm
Media	303.64 mm	406.18 mm	587,4 mm
Precisione	< 1 mm	1.06 mm	1.07 mm
Accuratezza (%)	98,78 %	98,45 %	96,29 %

Table 4.3: Performance sensore infrarossi modalità impulsata 2

Precisione media	< 1 mm
Accuratezza media	97,78 %

Table 4.4: Riepilogo generale sensore infrarossi modalità impulsata

Le performance sono più che buone considerando che questo modulo è realizzato tramite circuiteria semplice, discreta e su breadboard. Osservando l'accuratezza media, è bene non eccedere i 600 mm poichè il dato letto sarà molto piccolo e sapendo che c'è una proporzionalità inversa tra tensione d'uscita e distanza, una grande variazione di essa comporterà una piccola variazione nella lettura (visto che l'ADC di Arduino ha solo 1023 livelli) causando quindi una diminuzione drastica dell'accuratezza finale.

E' stato rilevato inoltre che cambiando la riflettività dell'oggetto, l'accuratezza scende mediamente del 10%, rendendo abbastanza inaffidabile la misura, soprattutto sulle brevissime distanze, visto la proporzionalità inversa che esiste tra tensione e distanza.

4.3.2 RILEVAZIONE PRATICA DELL'ANGOLO MASSIMO DI MISURA

La procedura è la medesima del sensore a ultrasuoni:

Angolo massimo di misura sensore infrarossi			
Distanza (L)	50 mm	150 mm	300 mm
Valore angolo SX	6.21°	5.28°	3.5°
Valore angolo DX	6.4°	5.16°	3.8°
Valore full cone (SX + DX)	12.61°	10.44°	7.3°

Table 4.5: Angolo massimo di misura sensore infrarossi



Confronto

Ricapitolando questi sono i dati emersi dai vari test di ogni sensore:

Riepilogo generale sensori			
	Ultrasuoni	Laser	Infrarossi
Costo sensore	2 €	7 €	~ 1 €
Precisione globale	2,26 mm	2,94 mm	< 1 mm
Accuratezza globale	99,3 %	98,23 %	97,78 %
Massima distanza misurabile	400 cm	190 cm	60 cm
Massimo angolo di misura	23.37°	20.56°	10.11°

Table 5.1: Riepilogo generale sensori

Partendo dal costo, quello minore è del modulo a infrarossi, principalmente per il fatto che non è un modulo *ready to use* ma è stato costruito utilizzando i singoli componenti discreti e con poca complessità (pochi componenti, ottima compattezza). Utilizzando un sensore già pronto sul mercato (*Sharp GP2Y0A21YK0F*) si otterrebbe un costo in linea con gli altri.

Per la precisione il migliore è stato quello a infrarossi, con un'acquisizione molto stabile, anche se gli altri due tipi presentano comunque un valore più che accettabile.

Globalmente tutti i rilevatori hanno un'ottima accuratezza, spicca però quello a ultrasuoni, mentre invece il sensore a infrarossi si trova leggermente in difetto, dovuto sempre al fatto di essere realizzato mediante componentistica discreta e lavorando sull'ampiezza del segnale luminoso ricevuto, essendo esso inversamente

proporzionale alla distanza, risulta molto impreciso quando ci si avvicina al limite del suo range.

La distanza massima misurabile è nettamente migliore nel sensore a ultrasuoni. Sicuramente la tecnologia a laser potrebbe coprire una distanza maggiore, ma non in questa fascia di prezzo (<10 €). L'infrarosso, non lavorando sul tempo di volo, soffre di altissima imprecisione quando si aumenta la distanza. Questo lo rende adatto alla misurazione di brevi lunghezze dove gode di buona accuratezza.

Per quanto riguarda l'angolo di misurazione, si trova che il sensore a ultrasuoni ha il miglior FOV (Field of View), seguito dal laser che ha un campo un po' più ridotto. L'infrarosso ha una prestazione molto inferiore, in quanto anche cambiando di poco l'angolazione, si faceva variare la tensione in uscita del rilevatore e, vista la proporzionalità altamente non lineare, la misurazione risultava troppo imprecisa.

La tecnologia a ultrasuoni presenta inoltre il vantaggio di non essere sensibile ai disturbi che introduce la luce artificiale e solare, un fattore sicuramente rilevante in un comune utilizzo di questi apparecchi.

Ricapitolando, a parità di fascia di prezzo, il sensore a ultrasuoni risulta globalmente il migliore. Per ottenere delle prestazioni simili se non superiori con il laser, è necessario raddoppiare il budget per arrivare al modello *VL53L1X* dotato di performance ben superiori al fratello minore.

Passando a rilevatori più costosi, il laser è sicuramente in grado di operare su distanze più lunghe rispetto all'ultrasuono e all'infrarosso poichè per natura il fascio è focalizzabile su aree molto piccole potendo avere una potenza radiante molto alta senza eccessive dispersioni.

Per quanto riguarda la tecnologia a infrarossi, la tecnica di calcolo della distanza lo pone sull'utilizzo nella misurazione di brevi distanze. Difatti facendo riferimento al modello in commercio *Sharp GP2Y0A21YK0F* esso possiede tra le caratteristiche un massimo di misurazione corrispondente a 80cm. Per ottenere capacità di misurazione più ampia è necessario passare ad un modulo TOF (*Time Of Flight*) dotato di una circuiteria molto più complessa in grado di rilevare il tempo di volo della luce (attraverso l'utilizzo di moltiplicatori di frequenza e memorie aggiuntive) rendendo la proporzionalità da tensione e distanza lineare. La scheda Arduino, con il suo clock non è adatta a acquisire dati a livello di velocità della luce (300 mila chilometri al secondo, frequenze dell'ordine dei GHZ). In conclusione, in base all'utilizzo e al budget, possiamo collocarci sulla tecnologia che più soddisfa le esigenze dell'utilizzatore.



Conclusioni

I sensori di distanza sono uno strumento molto utilizzato al giorno d'oggi che vanno in molti casi a sostituire il metro tradizionale agevolando le misurazioni.

Sul mercato abbiamo a disposizione diverse tecnologie di realizzazione, ognuna con i suoi vantaggi e svantaggi, facilmente risolvibili con le dovute accortezze elettroniche (vedi infrarossi e luce ambientale)

Con l'evoluzione dell'elettronica si è in grado al giorno d'oggi di effettuare misurazioni con precisione e a distanze che fino a diversi anni fa erano impossibili da immaginare. Nelle fasce di prezzo basse è più conveniente l'ultrasuono, mentre il laser è molto più preciso su prodotti di fascia più alta, oltre ad essere l'unico a raggiungere distanze considerevoli, che possono raggiungere anche i 200 metri. Da precisare che quando si lavora con il laser sopra una certa potenza è necessario prestare attenzione durante l'utilizzo per evitare danni agli occhi.

La tecnologia a infrarossi invece, è più adatta nel funzionamento come sensore di prossimità o misuratore di piccole distanze, visto che gode di una buona accuratezza sul corto raggio. Come ampliamento dell'elaborato sarebbe interessante valutare l'utilizzo del laser accoppiato alla visualizzazione digitale dell'ambiente di misurazione. Attraverso l'elaborazione software infatti, è possibile associare le misurazioni, anche molto estese a delle riproduzioni 3D o viste aeree realizzati con droni. Questo permette ai progettisti di poter lavorare con visione molto realistica ed estrema precisione anche su ambienti molto molto grandi con tutti i vantaggi moderni della tecnologia digitale.

List of Figures

2.1	Dettaglio ingressi e uscite scheda Arduino	8
2.2	Collegamenti sensore ultrasuoni	9
2.3	Diagramma dei tempi di un ciclo di misurazione	10
2.4	Area di lavoro per test	14
2.5	Area di lavoro per misurazione angolo massimo	16
3.1	Specifica emettitore laser	20
3.2	Dettaglio ricevitore laser	20
3.3	Pinout modulo VL53L0X	21
3.4	Riferimento pin I2C	21
4.1	Schema elettrico circuito sensore infrarossi	30
4.2	Corrente massima da datasheet	30
4.3	Tensione del diodo in base alla corrente di lavoro	31
4.4	Schema elettrico circuito sensore infrarossi con filtraggio disturbi	33
4.5	Tabella e grafico con fit dei dati	36
4.6	Dettaglio impulso post-filtraggio catturato da oscilloscopio	37
4.7	Tabella e grafico con fit dei dati modalità a impulsi	38

List of Tables

2.1	Lista componenti ultrasuoni	7
2.2	Test sensore ultrasuoni	15
2.3	Angolo massimo sensore ultrasuoni	17
3.1	Lista componenti laser	20
3.2	Test sensore laser	26
3.3	Angolo massimo di misura sensore laser	27
4.1	Riepilogo performance sensore infrarossi modalità continua	41
4.2	Performance sensore infrarossi modalità impulsata 1	42
4.3	Performance sensore infrarossi modalità impulsata 2	42
4.4	Riepilogo generale sensore infrarossi modalità impulsata	42
4.5	Angolo massimo di misura sensore infrarossi	43
5.1	Riepilogo generale sensori	45

Bibliografia

- [1] Texas Instruments. *Op amp stability and input capacitance*. URL: https://www.ti.com/lit/an/slyt087/slyt087.pdf?ts=1687347593525&ref_url=https%253A%252F%252Fwww.google.com%252F.
- [2] MakersLab. *Sensore HC-SR04*. URL: <https://www.makerslab.it/sensore-di-distanza-ad-ultrasuoni-hc-sr04-con-arduino/>.
- [3] Microchip. *Datasheet MCP-6002*. URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP6001-1R-1U-2-4-1-MHz-Low-Power-Op-Amp-DS20001733L.pdf>.
- [4] STMicroelectronics. *Datasheet VL53L0X*. URL: <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>.
- [5] OSRAM. *Datasheet SFH-203*. URL: https://media.osram.info/media/resource/hires/osram-dam-5488337/SFH+203_EN.pdf.
- [6] OSRAM. *Datasheet SFH-4550*. URL: https://media.osram.info/media/resource/hires/osram-dam-5580407/SFH+4550_EN.pdf.
- [7] Pololu-GitHub. *VL53L0X Arduino library*. URL: <https://github.com/pololu/vl53l0x-arduino>.
- [8] Wikipedia. *Laser*. URL: <https://it.wikipedia.org/wiki/Laser>.

