UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica e dell'Automazione

Studio e sviluppo di un sistema acustico differenziale subacqueo per l'inseguimento di delfini durante la predazione

Study and implementation of a low-cost differential underwater system for dolphin acoustical tracking during an attack

Relatore:
Prof David Scaradozzi

Tesi di Laurea di:
Nicola Ronchini

A.A. 2021/2022

SOMMARIO

ELENCO DELLE FIGURE 4

INTRODUZIONE 6

CAPITOLO 1 STATO DELL'ARTE 10

1.1 Principali tecniche di localizzazione 10

CAPITOLO 2 MATERIALI E METODI 12

2.1 M5StickC Plus 12

2.1.1 Microfono SPM1423 15

2.2 M5Atom Echo 16

2.3 M5Stack 17

2.4 I2S 18

2.5 I2C 19

2.6 Arduino IDE 20

CAPITOLO 3 STRATEGIE ED ALGORITMI DI ORIENTAMENTO SONORO 22

3.1 Cross-Correlazione 22

3.2 Generalized-Cross-Correlation 23

3.3 TDOA 24

CAPITOLO 4 PROGETTAZIONE DEL SISTEMA ED ANALISI DEI RISULTATI 25

4.1 Collegamento 25

4.2 Microfoni 26

4.3 M5Stack 26

4.4 Python/Matlab 27

4.5 Prove con dati scaricati dalla rete 31

4.6 Prove con dati registrati 33

CONCLUSIONI 36

BIBLIOGRAFIA 37

ELENCO DELLE FIGURE

Figura 0.1: esempio di onda sonora 6
Figura 0.2: mappatura fondale marino con radar 7
Figura 0.3: primo dispositivo di localizzazione acustica chiamato telescopio acustico 8
Figura 0.4: primi impieghi militari 8
Figura 2.1: microcontrollore M5StickC Plus 12
Figura 2.2: dispositivi presenti all'interno dell'M5StickC Plus 13
Figura 2.3: composizione M5StickC Plus 14
Figura 2.4: Interfaccia UiFlow 14
Figura 2.5: caratteristiche microfono 15
Figura 2.6: risposta in frequenza microfono 15
Figura 2.7: M5Atom ECHO 16
Figura 2.8: interno dell'M5Atom 16
Figura 2.9: microcontrollore M5Stack 17
Figura 2.10: dispositivi presenti all'interno dell'M5Stack 17
Figura 2.11: composizione M5Stack 18
Figura 2.12: schema I2S 19
Figura 2.13: I2C master-slave 20
Figura 2.14: schema I2C 20
Figura 2.15: interfaccia ArduinoIDE 21
Figura 3.1: schema posizionamento microfoni per TDOA 24
Figura 4.1: collegamento tra gli M5StickC e l'M5Stack 25
Figura 4.2: plot segnali registrati M5StickC Plus 28
Figura 4.3: file .txt con i dati dei segnali precedenti 29
Figura 4.4: spettro dei due segnali precedenti 30
Figura 4.5: grafico segnali scaricati 31
Figura 4.6: cross-correlazione segnali precedenti 31
Figura 4.7: cross-correlazione normalizzata segnali precedenti 32
Figura 4.8: risultato algoritmo TDOA 32

Figura 4.9: utilizzo M5Atom ECHO come microfoni 33
Figura 4.10: esempio di dati raccolti.....34
Figura 4.11: GCC dati precedenti 34
Figura 4.12: errore direzione 35

INTRODUZIONE

I primi studi della percezione del suono negli esseri viventi risalgono ai greci ma solo dopo lo sviluppo dell' equazione delle onde si riuscì a comprendere meglio il meccanismo che regola questo fenomeno.

Il suono può essere descritto come un'onda che si propaga come vibrazione in aria o in altri mezzi elastici e che può essere captato, per esempio, dal nostro meccanismo uditivo.

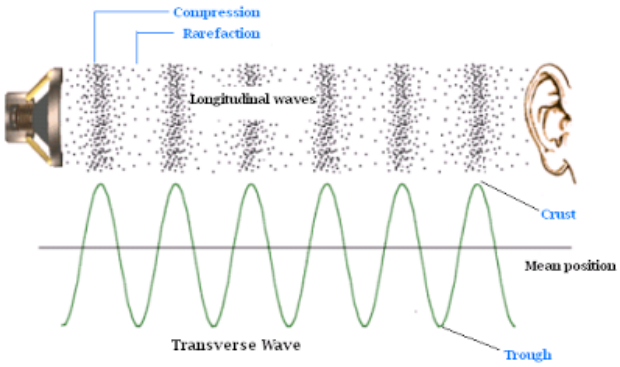

Figura 0.1: esempio di onda sonora

Le onde sonore sono descritte dalla seguente equazione (semplificata ad una dimensione): $\frac{\partial^2 p}{\partial x^2} - \frac{1}{v^2}\frac{\partial^2 p}{\partial t^2} = 0$ (dove p è la pressione acustica e v è la velocità di propagazione).

Supponendo v costante la soluzione generale è data da: p = f(ct-x) + g(ct+x); ovvero la somma di una onda progressiva ed una regressiva.

Grazie a questo e allo sviluppo di strumenti matematici, quali le serie e le trasformate di Fourier(utilizzate in gran parte nel nostro mondo moderno, non solo per l'acustica, ma anche per tutte le tecnologie di telecomunicazione, analisi dei segnali, compressione dei dati, ecc…), è stato possibile approfondire il problema della localizzazione sonora; sia studiando gli esseri viventi come noi, ma anche altri tipi di animali, come pipistrelli e delfini, che hanno una percezione del suono molto più accentuata della nostra, che cercando di replicare questi complessi meccanismi utilizzando strumenti ed attrezzature costruite dall'uomo stesso.

Lo studio di tecniche di localizzazione sonore in ambienti dove non è possibile utilizzare strumenti come il GPS e similari, come sotto la superficie dell'acqua (dove non sono di semplice utilizzo le onde elettromagnetiche), sta crescendo molto rapidamente.

Per esempio si stanno cercando dei metodi per utilizzare robot domestici in grado di rispondere alla chiamati di una persona, particolarmente importante nel campo della domotica assistita.

Si possono trovare tecniche di localizzazione sonora anche in ambiente industriale; attraverso un array di microfoni e un sistema di riconoscimento si possono identificare suoni riconducibili a problemi di alcuni macchinari e, consecutivamente, si può stimare la posizione del dispositivo difettoso.

Lo studio del suono si sta utilizzando anche nella realtà virtuale dove è importante avere una distribuzione coerente con ciò che si vede.

Utilizzo di tecniche sonore per il rilevamento di ostacoli o la mappatura di ambienti sono anche i sonar dei sottomarini e delle navi.

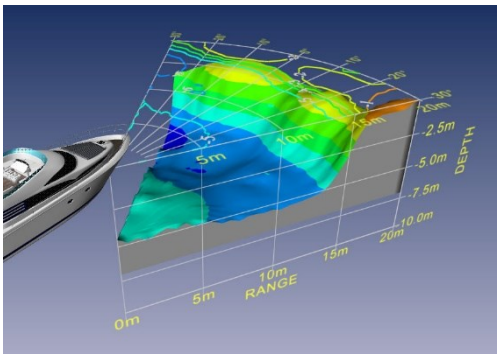

Figura 0.2: mappatura fondale marino con radar

Tecniche più avanzate puntano anche ad aggiungere un sistema di visione artificiale per semplificare il rilevamento del suono, infatti la localizzazione sonora è estremamente difficile, in particolare in ambienti dove è presente eco, sovrapposizione di diverse fonti sonore o fonti in movimento.

Storicamente la prima applicazione di localizzazione sonora risale a fine 1800 dove si creò un “telescopio acustico” per localizzare navi in condizioni nebbiose.

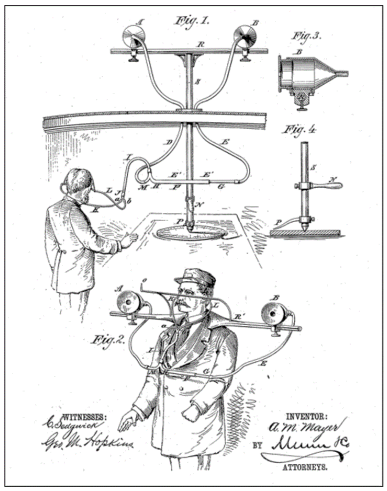

Figura 0.3: primo dispositivo di localizzazione acustica chiamato telescopio acustico

Le prime applicazioni di localizzazione acustica in ambito militare risalgono invece a metà della prima guerra mondiale dove venivano utilizzate per stimare la posizione di un cannone che sparava o dei motori degli elicotteri; successivamente, durante la seconda guerra mondiale, venne sviluppato il radar che soppiantò in parte la localizzazione sonora in aria, mentre venne continuato ad usare e a migliorare il sonar per applicazioni sottomarine.

Figura 0.4: primi impieghi militari

Lo scopo principale di questa tesi è verificare se è possibile captare e localizzare una sorgente sonora in un ambiente sottomarino utilizzando due microfoni per simulare le orecchie di un essere vivente.

L’utilizzo di due microfoni ci limita all’identificazione in sole due dimensioni; durante la tesi si svolgeranno prove sia in aria che in acqua in presenza di una singola fonte sonora fissa trascurando l’eco.

Se si vuole espandere la localizzazione ad uno spazio tridimensionale è necessario aumentare il numero di microfoni; per sorgenti in movimento sono necessari algoritmi più raffinati.

La tesi sarà divisa in 5 capitoli:

1. Stato dell’arte: principali tecniche studiate per la localizzazione sonora utilizzando microfoni o array di microfoni.

2. Materiali e metodi: elenco dei dispositivi hardware, protocolli e software utilizzati durante le varie prove e principali funzionalità.

3. Algoritmi di localizzazione: algoritmi utilizzati per la localizzazione sonora.

4. Progettazione del sistema ed analisi dei risultati: connessione e programmazione dei dispositivi hardware ed analisi di segnali, sia ricevuti sia utilizzando database scaricati da internet.

5. Conclusioni: ricapitolazione dei risultati ottenuti e sviluppi futuri.

Capitolo 1
STATO DELL'ARTE

Nel campo della localizzazione acustica le ricerche sono molte ed in continuo sviluppo.

Si trovano studi e tecniche per un vasto insieme di applicazioni, dalla robotica alla sicurezza, dalla gestione di processi industriali alla domotica assistita.

Molti algoritmi menzionati sotto prendono spunto dal meccanismo uditivo di molti esseri viventi, tra cui anche gli stessi esseri umani:

- Binaurale: si cerca di determinare la posizione di una sorgente attraverso la differenza del segnale captato tra due orecchie.
- Monoaurale: si cerca l'angolo d'incidenza relativo ad un singolo orecchio.

Nelle ultime due/tre decadi la ricerca di una sorgente sonora attraverso l'uso di microfoni ha ricevuto molta attenzione, soprattutto nei campi della robotica, della medicina e del processamento dei segnali; al momento però questo tipo di localizzazione è per la maggior parte teorica a causa della complessità del problema; alcuni dei maggiori problemi riscontrati sono, per esempio, l'eco e l'effetto Doppler causato da sorgenti sonore in movimento.

Ultimamente si uniscono alla localizzazione sonora anche telecamere ed algoritmi di visione artificiale in grado di riconoscere la sorgente.

Insieme a problemi di localizzazione vengono studiati anche algoritmi per un filtraggio ottimale dei dati.

1.1 **Principali tecniche di localizzazione**

- TOA (time of arrival): si utilizzano le informazioni riguardanti l'istante di ricezione del segnale da parte dei microfoni.
- TDOA (time difference of arrival): si utilizza il tempo di ritardo della ricezione del segnale fra due o più microfoni. Il tempo di ritardo viene calcolato, per esempio, utilizzando la correlazione fra due segnali. La sincronizzazione è necessaria tra i vari microfoni.

- DOA (direction of arrival): ogni microfono calcola indipendentemente la direzione di arrivo della onda sonora; questo metodo non ha strettamente bisogno della sincronizzazione, almeno fino a che la sorgente si muove a basse velocità, però richiede un elevato costo computazionale.
- IID (inter microphone intensity difference): attraverso due microfoni si comparano i due segnali per verificare quale dei due è stato più "forte", così facendo si ottiene una approssimazione di dove sia la sorgente; per aumentare la risoluzione si può pensare di aumentare il numero di microfoni (alcuni dei robot domestici in commercio utilizzano questo tipo di algoritmo).
- Beamforming acustico: utilizza un insieme di microfoni con posizionamento scelto ad hoc e calcola tutti i vari ritardi fra di essi, successivamente li somma insieme per localizzare il segnale. In questa tecnica è molto importante la geometria del problema e il numero di microfoni impiegati(maggiore il numero meglio è). Il risultato finale è quello di ottenere una mappa acustica del luogo scelto.
- Localizzazione in 3 dimensioni: non è una vera e propria tecnica, ma il continuo degli algoritmi precedenti per ambienti in 3 dimensioni. In questo caso la geometria del problema diventa di notevole importanza e la complessità aumenta significativamente a causa dell'aumento del numero di microfoni necessari; tutti gli algoritmi precedenti sono estendibili al caso in tre dimensioni.
- Machine learning e reti neurali: con l'espansione di algoritmi di intelligenza artificiale è possibile anche localizzare le sorgenti sonore. Utilizzabili soprattutto in problemi ad alta complessità dove una soluzione numerica diventerebbe troppo costosa computazionalmente. È prima necessario addestrare questi algoritmi con situazioni di cui si conosce già la soluzione (possibile elevato costo computazionale nel momento di Training); ciò comporta un probabile malfunzionamento in altri ambienti che non siano quello dove è stata addestrata la rete.

Capitolo 2
MATERIALI E METODI

2.1 **M5StickC Plus**

L'M5stickC Plus è un microcontrollore che comprende un processore ESP32 dual core, 520KB di SRAM ed una connessione Wi-Fi.

Figura 2.1: microcontrollore M5StickC Plus

All'interno dell'M5 sono presenti una varietà di dispositivi per applicazioni di robotica e IoT come, ad esempio, giroscopio ed accelerometro, interfacce per gestire un Cloud, reti Wi-Fi, ecc .

Nella foto sottostante sono elencati i principali dispositivi presenti:

ESP32	240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi
Flash Memory	4MB
Power Input	5V @ 500mA
Port	TypeC x 1, GROVE(I2C+I/O+UART) x 1
LCD screen	1.14 inch, 135*240 Colorful TFT LCD, ST7789v2
Button	Custom button x 2
LED	RED LED
MEMS	MPU6886
Buzzer	built-in buzzer
IR	Infrared transmission
MIC	SPM1423
RTC	BM8563
PMU	AXP192
Battery	120 mAh @ 3.7V
Antenna	2.4G 3D Antenna
PIN port	G0, G25/G36, G26, G32, G33
Operating Temperature	0°C to 60°C
Net weight	15g
Gross weight	21g
Product Size	48.2*25.5*13.7mm
Package Size	65*25*15mm
Case Material	Plastic (PC)

Figura 2.2: dispositivi presenti all'interno dell'M5StickC Plus

Tra i dispositivi presenti si può notare che è presente anche un microfono (MIC: SPM1423). Grazie a questo dispositivo sarà possibile usare due M5, indipendenti tra loro, come “orecchie” del sistema.

Sono presenti anche dei GPIO (General Purpose Input Output) nella parte alta del dispositivo che permettono il collegamento ad altri dispositivi.

Nella parte bassa invece è presente una porta utilizzabile come porta I2C o normale GPIO ed una porta seriale utilizzata per caricare il codice dal computer (ed anche inviare e trasferire dati dal computer al dispositivo e viceversa).

Nella parte centrale del dispositivo, sotto lo schermo, è presente anche un pulsante.

Figura 2.3: composizione M5StickC Plus

Il microcontrollore è programmabile sia in MicroPython che in c (Arduino).

Un altro modo per programmare è attraverso M5Flow (o UiFlow), una interfaccia grafica che permette di generare codice in modo automatico in MicroPython, utilizzabile quando la complessità non è eccessiva.

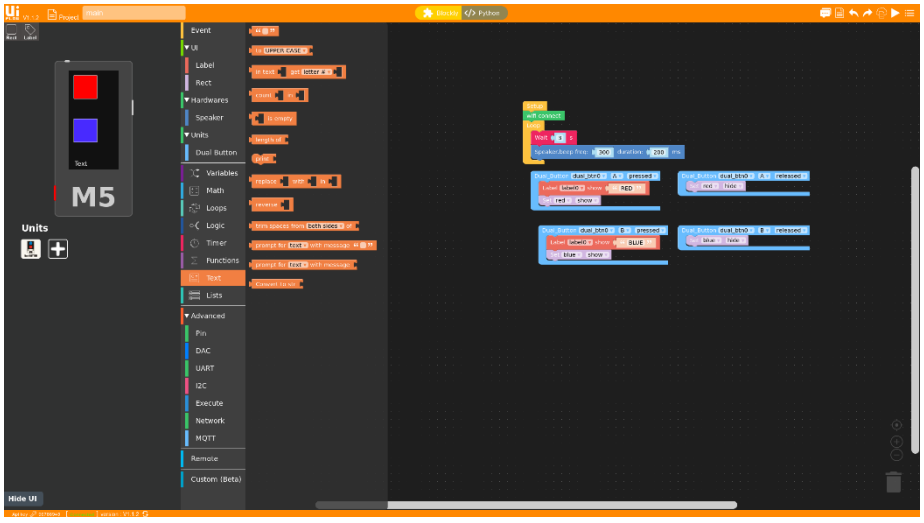

Figura 2.4: Interfaccia UiFlow

2.1.1 *Microfono SPM1423*

Questo è il microfono integrato nell'M5, è possibile accederci tramite software grazie alla porta I2S, ed è collegato ai PIN 0 (Clock) e 34 (Dati).

Nella seguente foto sono riportate le caratteristiche principali:

Digital microphone	SPM1423
Sensitivity	94dB SPL@1KHz Nom.:-22dBFS
Signal to Noise Ratio	94dB SPL@1KHz,A-weighted Nom.:61.4dB
Total Harmonic Distortion	100dB SPL@1KHz Nom.:1%
Working Current	600μA

Figura 2.5: caratteristiche microfono

Di notevole importanza è come varia la sensitività rispetto alle frequenze ricevute; si può avere un'idea grazie al seguente grafico (scaricato dal DataSheet, reperibile nel sito ufficiale dell'M5StickC Plus):

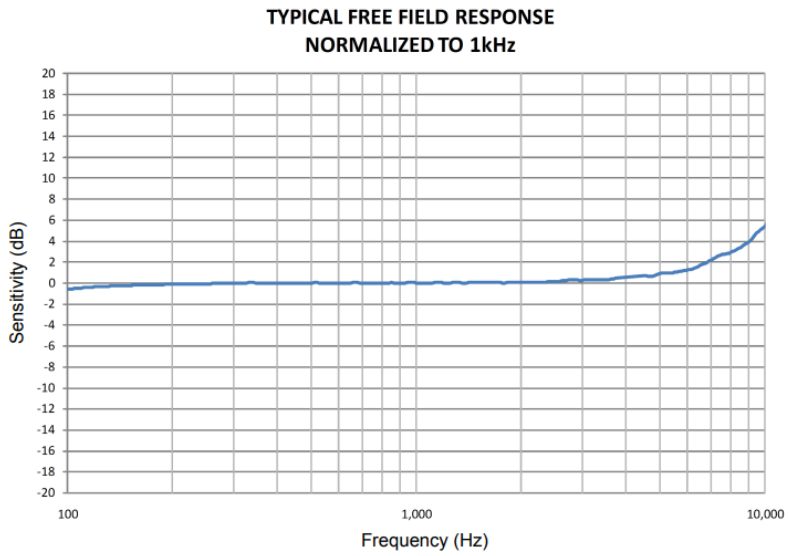

Figura 2.6: risposta in frequenza microfono

Osservando il grafico si può notare come la sensitività aumenta ad alte frequenze, ciò significa che segnali oltre 10 kHz di frequenza dovranno essere molto amplificati per fare in modo che il microfono riesca a percepirli correttamente.

Grazie a questo microfono è stato possibile catturare il suono per poi analizzarlo.

2.2 **M5Atom Echo**

Questo dispositivo fa sempre parte della famiglia degli M5 ed è molto simile, come specifiche, al M5StickC base.

Figura 2.7: M5Atom ECHO

Rispetto al precedente si presta molto bene ad essere usato come microfono in quanto è di dimensione sensibilmente ridotte.

Figura 2.8: interno dell'M5Atom

Anche questo microcontrollore è programmabile sia con UiFlow che in ArduinoIDE (in quest'ultimo caso sono utilizzate le stesse librerie dell'M5StickC, non plus).

È stato utilizzato durante alcune prove al posto degli M5StickC Plus principalmente per le dimensioni più contenute che facilitano il posizionamento dei microfoni.

2.3 **M5Stack**

L'M5Stack è un altro tipo di microcontrollore, sempre con un processore ESP32 dual core, 520KB di SRAM ed una connessione Wi-Fi, che sarà incaricato di gestire e sincronizzare i due M5StickC Plus.

Figura 2.9: microcontrollore M5Stack

Anch'esso presenta al suo interno molti dispositivi di cui i principali si possono visualizzare nella seguente foto:

Specifications	Parameters
ESP32-D0WDQ6-V3	240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi
Flash	16MB
Input Voltage	5V @ 500mA
Host Interface	TypeC x1, POGO PIN x1, I2C x1, GPIO x1, UART x1
IPS Screen	2 inch, 320x240 Colorful TFT LCD, ILI9342C, 853nit max brightness
Keys	Custom Keys x 3
Speaker	1W-0928
Microphone	Analog BSE3729 Microphone
IMU	6-axis MPU6886
USB Chip	CH9102F
LED	SK6812 RGB LED x 10
Antenna	2.4G 3D antenna
Battery	500 mAh @ 3.7V
Operating Temperature	0°C to 40°C
Net Weight	56.4g
Gross Weight	228g
Product Dimensions	54 x 54 x 21 mm
Package Size	147 x 90 x 40 mm
Cover Material	Plastic (PC)

Figura 2.10: dispositivi presenti all'interno dell'M5Stack

Di notevole importanza è la presenza di una flash di 16Mb su cui è possibile salvare i segnali audio ricevuti dai due microfoni per studiarli in un secondo momento.

Anche in questo microcontrollore sono presenti dei GPIO (di notevole utilità in quanto serviranno per dare il segnale per sincronizzare i due M5Stick); sono anche presenti due porte I2C e due porte seriali (di cui una è utilizzata soltanto per comunicare con il computer come nel caso dell'M5StickC Plus).

Sono presenti anche tre pulsanti nella parte superiore del dispositivo, al di sotto dello schermo.

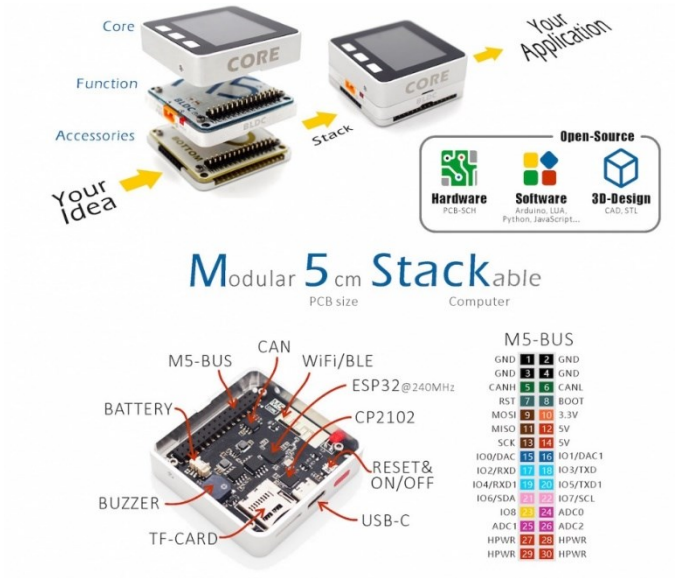

Figura 2.11: composizione M5Stack

Questo tipo di microcontrollore è altamente personalizzabile, infatti in commercio si possono trovare dei moduli aggiuntivi per aggiungere delle funzionalità al dispositivo. Nella figura precedente (fig. 2.7), per esempio, è presente un modulo LoraWan(di colore blu) che può essere utilizzato per una rete in stile IoT (Internet of Things).

Anche l'M5Stack può essere programmato con gli stessi programmi dell'M5StickC Plus.

2.4 **I2S**

L'I2S (Inter-IC Sound), da non confondere con l'I2C che si vedrà successivamente, è un collegamento seriale specializzato nella gestione di segnali audio ricevuti da microfoni.

Il collegamento è formato da 3 fili:

- Una linea di trasmissione dati (serial Data).

- Una linea chiamata “World-Select” che si occupa di gestire, nei microfoni stereo, la linea dati sinistra e destra.
- Una linea di Clock che sarà utilizzata per determinare la frequenza di campionamento. Durante il corso di questa tesi la frequenza di campionamento sarà impostata a 44100 Hz o a 22050 Hz in quanto il microfono sopra i 10 KHz non riceve più in maniera ottimale.

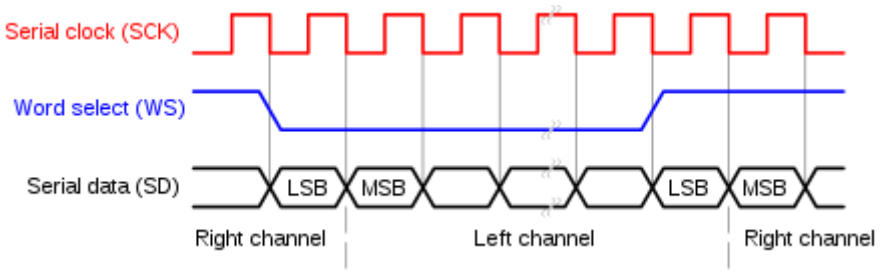

Figura 2.12: schema I2S

I2S è relativamente più veloce rispetto all’I2C (più lento, ma in grado di effettuare operazioni più complesse che si vedranno nel prossimo sotto-capitolo), infatti dovrà gestire segnali di frequenze elevate, anche dell’ordine dei KHz.

Una ulteriore differenza importante rispetto alla porta I2C è la unidirezionalità della I2S, quindi non si potranno modificare le specifiche del microfono (la frequenza di campionamento e il tipo di dati ottenuti sono specificati nel software durante l’inizializzazione dei driver I2S e dipendono dallo strumento collegato al microfono).

2.5 **I2C**

I2C (Inter Integrated Circuit) è un sistema di comunicazione seriale bifilare utilizzato per scambiare dati ed informazioni tra dispositivi e circuiti integrati.

Il protocollo hardware I2C è composto da due fili:

- SDA (Serial Data): utilizzato per trasferire dati
- SCL (Serial Clock): utilizzato per gli impulsi di Clock.

Vanno aggiunte anche una linea di riferimento ed una linea di alimentazione (che può variare avere tensioni tipiche di 3,3V o 5V) che possono essere condivise da tutti i dispostivi connessi all’I2C.

Questo protocollo di comunicazione può essere utilizzato tra due dispositivi: uno slave ed un master; però permette il collegamento anche MultiSlave e MultiMaster come in figura.

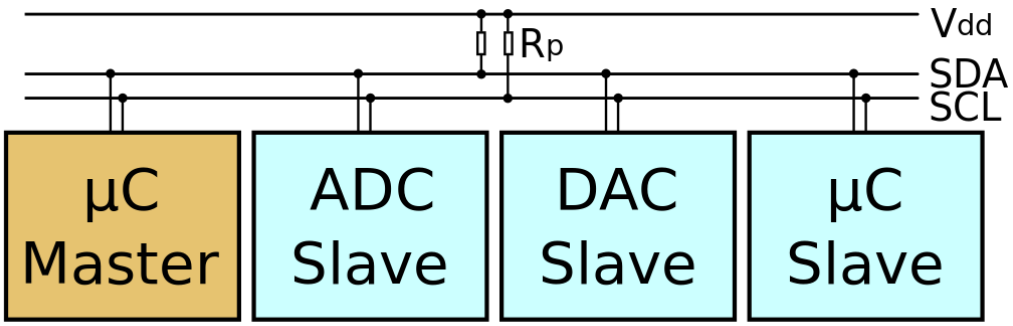

Figura 2.13: I2C master-slave

Questo permetterà di connettere i due M5StickC Plus(che funzioneranno da Slave) al M5Stack (Master) attraverso questo protocollo per inviare i dati del microfono.

Per identificare i dispostivi vengono assegnati degli indirizzi a 7 bit univoci (solitamente scritti in notazione esadecimale; nelle ultime revisioni sono stati aumentati i bit a 10) per permettere la trasmissione.

L'I2C può utilizzare diversi velocità; le tre principali sono:

- Low-Speed Mode: velocità base di 100 kbit/s (ma nulla impedisce di trasferire a velocità inferiori)
- Fast Mode: 400 kbit/s
- High-Speed Mode: 3,4 Mbit/s

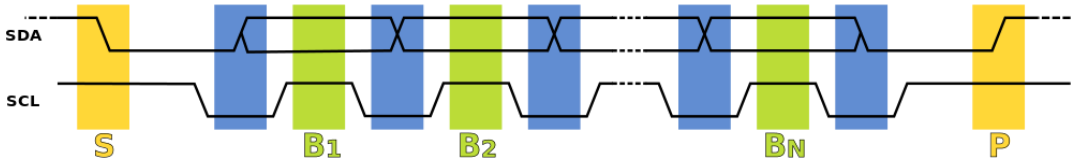

Figura 2.14: schema I2C

A differenza dell'I2S l'I2C è bidirezionale quindi si possono traferire informazioni in entrambe le direzioni.

2.6 **Arduino IDE**

Tutti i programmi scritti durante questa tesi sono stati creati nell'IDE di Arduino.

La scelta è ricaduta su questa IDE in quanto, utilizzando il linguaggio C, si è molto più liberi nello scrivere il codice, sono presenti più funzioni ed è molto più comodo utilizzare le librerie rispetto a MicroPython.

L'IDE di Arduino ed il sito GitHub di M5 mettono a disposizione delle librerie grazie alle quali è possibile programmare più semplicemente i dispostivi appartenenti alla famiglia degli M5.

Un'altra libreria molto importante include i driver del protocollo I2S, grazie alla quale è stato molto semplificato il processo di connessione con il microfono.

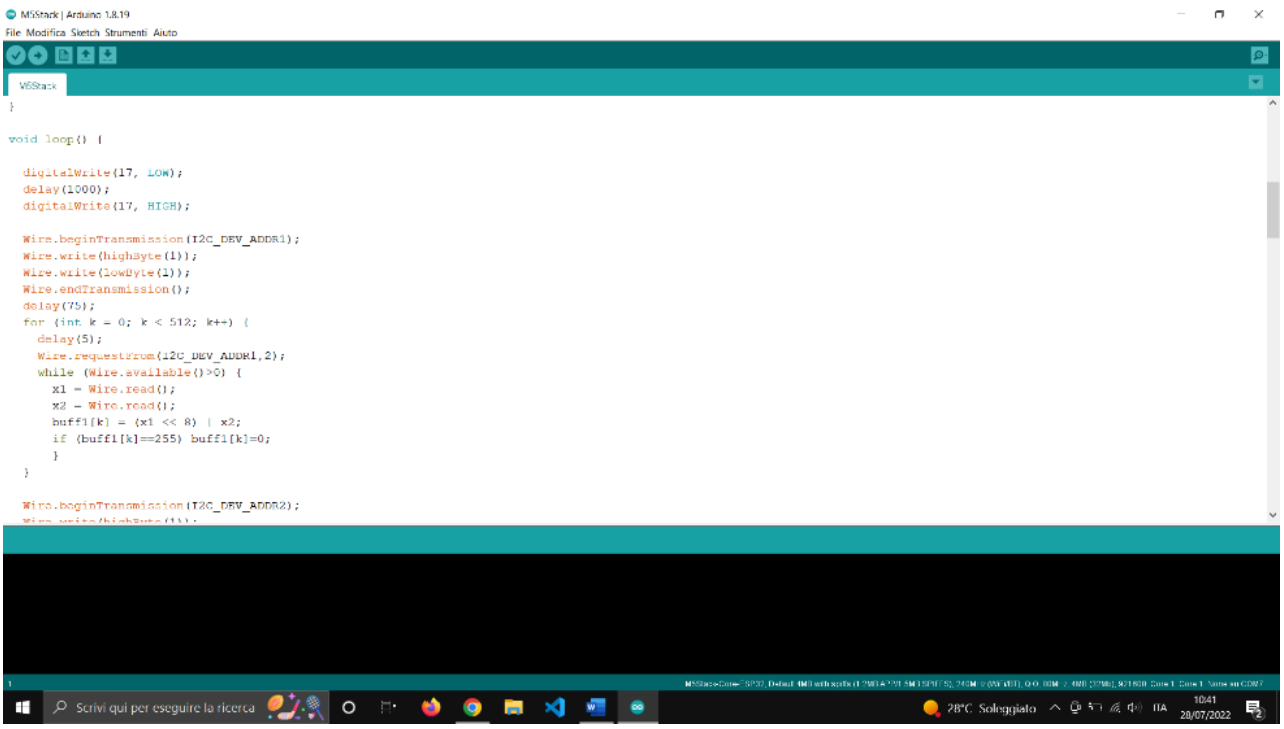

Figura 2.15: interfaccia ArduinoIDE

Capitolo 3
STRATEGIE ED ALGORITMI DI ORIENTAMENTO SONORO

In questo capitolo si andrà a vedere che tecniche sono state utilizzate per localizzare una sorgente sonora e come sono state implementate.

3.1 **Cross-Correlazione**

La tecnica più utilizzata per ottenere il tempo di ritardo tra due o più microfoni è chiamata Cross-Correlazione.

Supponiamo di avere due segnali:

$$s_1(t) = n_1(t) + e_1(t)$$
$$s_2(t) = n_2(t - \tau) + e_2(t)$$

Dove s_1 e s_2 sono i due segnali ricevuti dai microfoni composti dalla somma dei segnali n_1 e n_2, senza rumore sovrapposto e con n_2 ritardato di un certo tempo τ, e da $e_1(t)$ e $e_2(t)$ che saranno i rumori supposto bianchi con distribuzione gaussiana ricevuti dai microfoni.

La cross-correlazione può essere calcolata nel seguente modo, nel dominio della frequenza:

$$R_{S_1S_2} = \int_{-\infty}^{+\infty} S_1(\omega) S_2^*(\omega) e^{-j\omega t} d\omega$$

Dove $S_1(\omega)$ è la trasformata di Fourier del segnale $s_1(t)$ e $S_2^*(\omega)$ è la trasformata del segnale $s_2(t)$ coniugata (la coniugazione di un numero complesso $x = a + bj$, con a e b numeri reali, equivale a $x^* = a - bj$).

Il termine $e^{-j\omega t}$ deriva da una proprietà della trasformata di Fourier, ovvero:

$$F\big(x(t - \tau)\big) = F(x(t)) e^{-j\omega\tau}$$

Ora che si è calcolata la cross-correlazione nel dominio della frequenza è sufficiente calcolarne la trasformata inversa per riportarla nel dominio del tempo.

$$CC = F^{-1}(R_{S_1S_2})$$

Per trovare il ritardo tra i due segnali è sufficiente cercare il massimo della funzione CC.

$$\tau = \frac{argmax(CC)}{F_c}$$

Dove F_c è la frequenza con la quale sono stati campionati i due segnali.

Allo stesso risultato si sarebbe potuti arrivare anche calcolando la cross-correlazione nel dominio del tempo attraverso una funzione di convoluzione.

$$(f * g)(\tau) = \int_{-\infty}^{+\infty} s_1(t) s_2^*(t - \tau)\, dt$$

Durante le varie prove sono stati utilizzati entrambi i metodi utilizzando la FFT (Fast Fourier Transform) per la procedura nel dominio della frequenza e metodi numerici per calcolare la convoluzione nel dominio del tempo utilizzando software come Python e Matlab.

3.2 **Generalized-Cross-Correlation**

Questo tipo di cross-correlazione è una variante proposta da Knapp e Carter nel 1976 e consiste nel calcolare la seguente cross-correlazione:

$$R_{S_1 S_2} = \int_{-\infty}^{+\infty} \psi(\omega) S_1(\omega) S_2^*(\omega) e^{-j\omega t} d\omega$$

Dove $\psi(\omega)$ è una funzione peso calcolata come:

$$\psi(\omega) = \frac{1}{|S_1(\omega) S_2^*(\omega)|}$$

L'idea dietro a questa funzione peso è quella di eliminare le componenti del modulo della funzione per lasciare solo i contributi della fase (che sarà poi quello che darà il tempo di ritardo).

Come nel caso precedente anche qui è sufficiente trovare il massimo dell'inversa della funzione $R_{S_1 S_2}$ e dividerla per la frequenza di campionamento per trovare il tempo di ritardo tra i due microfoni.

Da studi precedenti sembra che questo tipo di cross-correlazione sia meno sensibile al rumore, quindi abbia una maggiore accuratezza, però presenta alcuni problemi quando la sorgente è a banda molto stretta, come per esempio una sinusoide di una sola frequenza, o in presenza di un SNR (Signal to Noise Ratio) relativamente basso.

Anche questo metodo, come la maggioranza degli algoritmi di localizzazione sonora o anche elettromagnetica, presenta problemi in casi di eco molto accentuato; questo fenomeno si può riconoscere come un numero di picchi nella correlazione maggiore di uno.

Durante le varia prove sono comunque state utilizzate entrambe per verificare quale delle due performa meglio e in che ambiente.

3.3 **TDOA**

Grazie al tempo di ritardo calcolato con gli algoritmi precedenti è ora possibile trovare l'angolo di arrivo della sorgente.

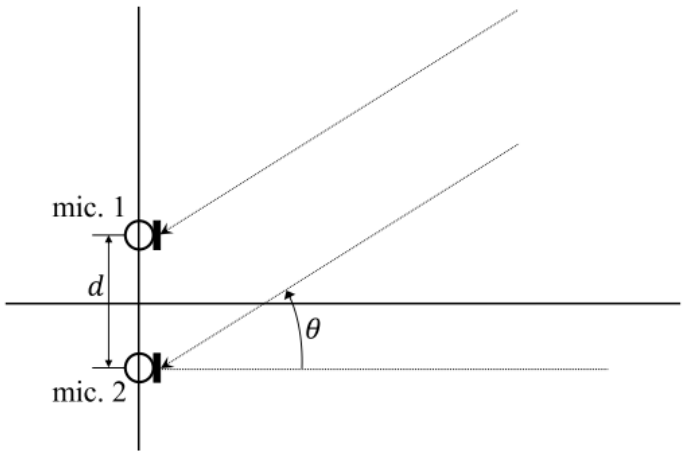

Figura 3.1: schema posizionamento microfoni per TDOA

L'angolo di arrivo si può calcolare come:

$$\theta = \sin^{-1}\frac{v\tau}{d}$$

Dove v è la velocità del suono (circa 343 m/s in aria e 1484 m/s in acqua), d è la distanza fissata tra i due microfoni e τ è il tempo di ritardo calcolato in secondi trovato con i precedenti algoritmi.

Utilizzando due microfoni singoli è possibile calcolare solo l'angolo θ nel piano a due dimensioni creato dai microfoni, per avere una descrizione più accurata della posizione, anche in tre dimensioni, è necessario aumentare il numero di microfoni.

Per migliorare la localizzazione in due dimensioni si potrebbero utilizzare array di microfoni al posto dei microfoni singoli, ciò consentirebbe di trovare le anche le coordinate della sorgente oltre all'angolo di arrivo, come anche aumentare il numero di microfoni.

Capitolo 4
PROGETTAZIONE DEL SISTEMA ED ANALISI DEI RISULTATI

4.1 **Collegamento**

Il sistema è composto da tre parti:

- I due microfoni utilizzabili dagli M5StickC Plus(o analogamente gli M5AtomEcho, il microfono presente all'interno è il medesimo).
- L'M5Stack che fungerà da controllore dei due microfoni.

I due microfoni sono collegati all'M5Stack attraverso due fili per il protocollo I2C grazie ai quali è possibile inviare e ricevere dati.

Oltre a questi due fili è presente anche un altro filo per inviare un segnale di sincronizzazione ai due microfoni.

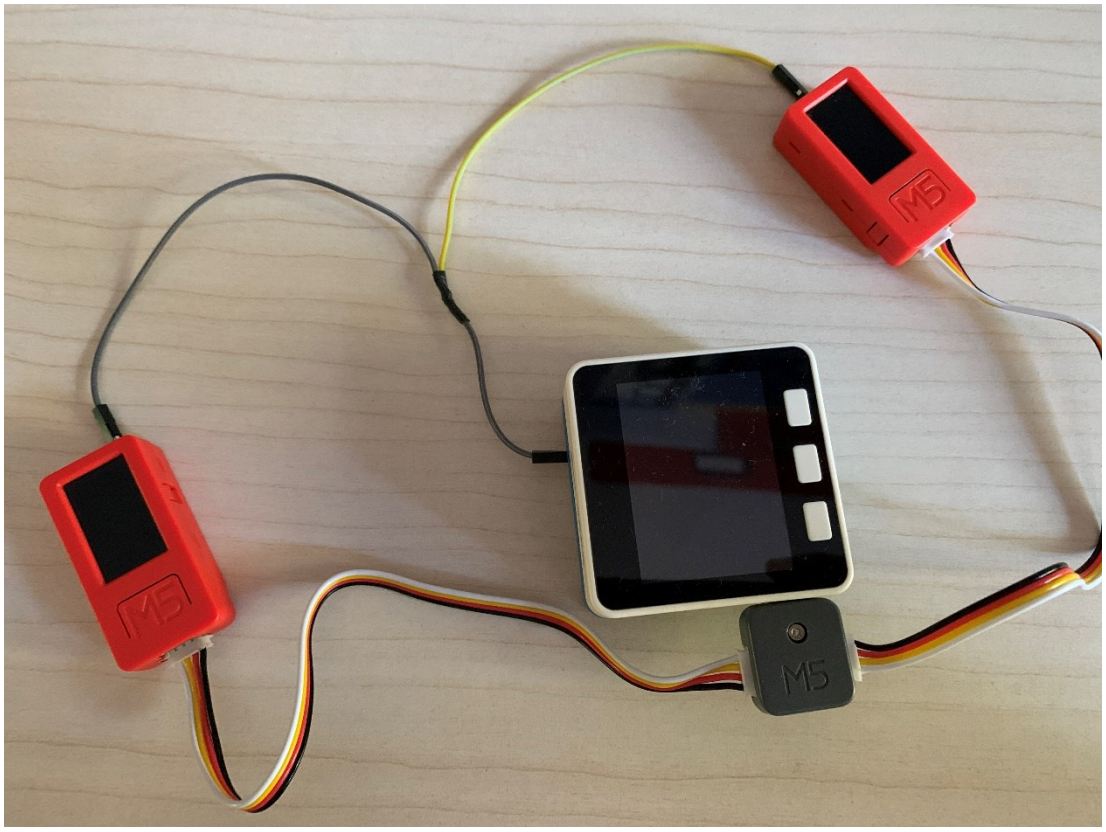

Figura 4.1: collegamento tra gli M5StickC e l'M5Stack

4.2 **Microfoni**

Il programma interno degli M5Stick C Plus è stato scritto in Arduino e comprende due parti principali (il programma per gli M5Atom Echo è identico a parte per il numero del GPIO che sarà 21):

1. Registrazione del suono: il suono viene registrato attraverso il microfono integrato SPM1423 ed inviato al processore dell'M5 grazie al protocollo I2S. L'I2S è stato inizializzato con un tempo di campionamento di 44100 Hz per permettere di far risaltare più chiaramente il possibile tempo di ritardo fra i due microfoni; inoltre i dati ricevuti dal microfoni sono stati settati per essere in formato int16_t ovvero un numero a 16 bit.
   I dati registrati sono salvati in un buffer di 512 campioni nel caso di utilizzo dell'M5Stack (per limiti tecnici alla velocità di invio) e 8192 in caso di invio diretto dei due M5StickC Plus al Pc tramite porta seriale.
   Il suono verrà registrato ogni qual volta il master (l'M5Stack o un semplice interruttore) invia un segnale nel filo collegato al GPIO 26 degli M5Stick.
2. Scambio di dati fra slave e master (nel caso dell'utilizzo dell'M5Stack): una volta registrato il suono il master invierà un altro segnale ai microfoni (questa volta attraverso i cavi dedicati all'I2C) per chiedere l'invio della registrazione; oppure scambio diretto tra M5Stick e il pc nel caso di invio diretto tramite seriale.

Per differenziare i due M5Stick è bastato dare due indirizzi I2C diversi al momento dell'inizializzazione, in questo caso sono stati usati 0x55 e 0x56; grazie a ciò i codice presenti nei due M5Stick sono gli stessi a parte l'inizializzazione.

4.3 **M5Stack**

L'M5Stack fungerà da master per i due M5Stick ed effettuerà operazioni sugli array ricevuti.

Il programma ha 3 fasi:

1. Invio di un segnale di durata di un secondo attraverso il GPIO 17 per comunicare agli M5Stick di cominciare a registrare.

2. Invio di un segnale sul canale I2C ai due M5Stick per cominciare a trasferire gli array registrati.
3. In questo terzo punto le strade possono essere due:
    - Continuare a lavorare all'interno dell'M5Stack calcolando la correlazione fra i due array per rilevare il tempo di ritardo e quindi l'angolo della sorgente.
      In questo caso si sono riscontrati problemi tecnici dovuti all'elevato costo computazionale necessario sia per calcolare la FFT e la IFFT che per calcolare la correlazione nel dominio del tempo. Successivamente si potrebbe pensare di utilizzare una scheda apposita come, per esempio, una Raspberry o un DSP (Digital Signal Processor) anche se ciò comporterà un aumento dei costi.
    - Utilizzare la porta seriale connessa al computer per passare i dati all'interno del pc grazie ad un programma scritto in Python o Matlab e lavorarci in un secondo momento. Per la maggior parte della tesi è stato usato questo approccio perché permette di studiare con più calma quali sono gli algoritmi più efficaci e in che condizioni e per sopperire ai limiti tecnici citati precedentemente.

4.4 **Python/Matlab**

Nonostante sia stato usato sempre Arduino per programmare l'hardware al momento dell'analisi dei dati è stato usato Python (soprattutto per interfacciarsi con la seriale) e Matlab (grazie ai suoi strumenti matematici semplici da usare).

Python è un linguaggio relativamente semplice e comprende una serie di librerie per l'interfacciamento seriale con altri dispositivi esterni al computer, oltre a librerie matematiche intuitive da utilizzare come, per esempio, la trasformata di Fourier o strumenti grafici per visualizzare i dati.

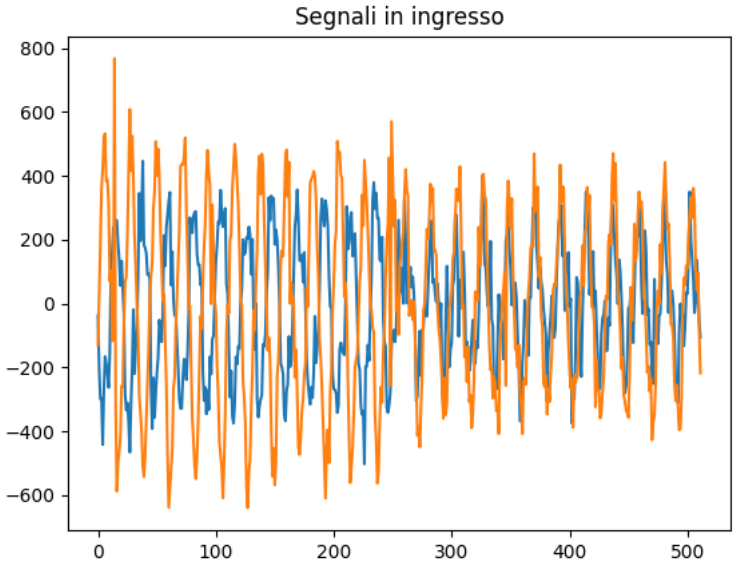

Figura 4.2: plot segnali registrati M5StickC Plus

Il programma principale utilizzato è stato creato in modo che vengono letti 1024 campioni (512 per ogni microfono, nel caso di M5Stack) o 16384 (8192 per microfono in caso di interfacciamento seriale diretto) e successivamente vengono salvati in un file, che può essere sia txt che csv, per permettere successive analisi come nella seguente foto.

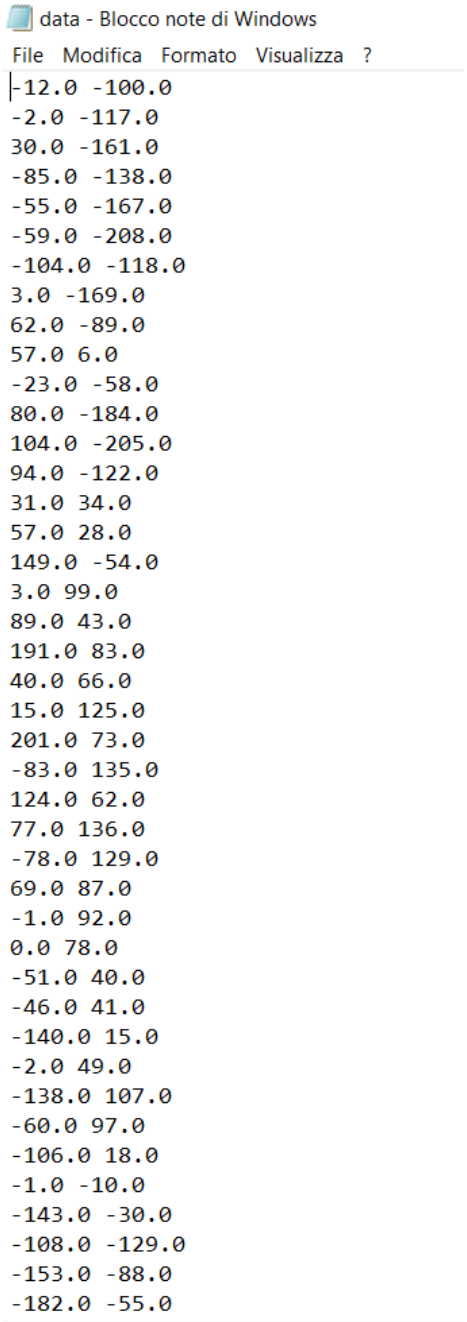

Figura 4.3: file .txt con i dati dei segnali precedenti

Grazie alle librerie presenti è stato possibile anche visualizzare correttamente lo spettro dei segnali registrati; relativamente alla immagine dei segnali sopra presente, per esempio, lo spettro è il seguente:

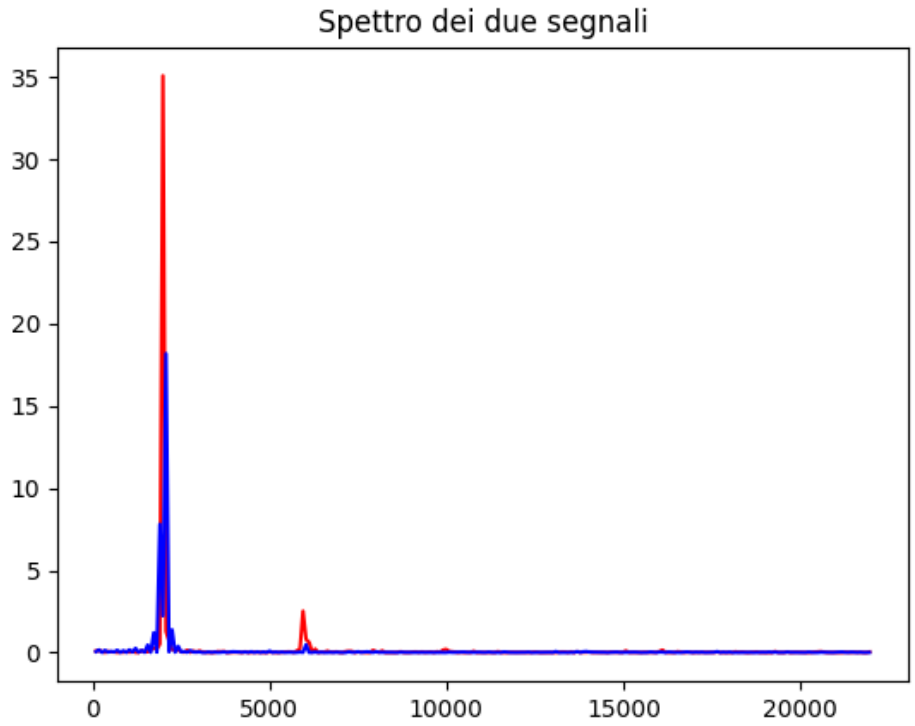

Figura 4.4: spettro dei due segnali precedenti

Si può notare come è presente un picco in coincidenza con 2000 Hz infatti i segnali erano due sinusoidi pure(con sovrapposto naturalmente un po' di rumore di fondo, si può infatti vedere un picco relativamente ridotto anche intorno ai 6 KHz) di 2 KHz.

4.5 **Prove con dati scaricati dalla rete**

Per verificare la correttezza degli algoritmi si è fatto uso inizialmente di un database reperito su GitHub in cui sono presenti diversi file WAV contenenti anche l'angolo di arrivo.

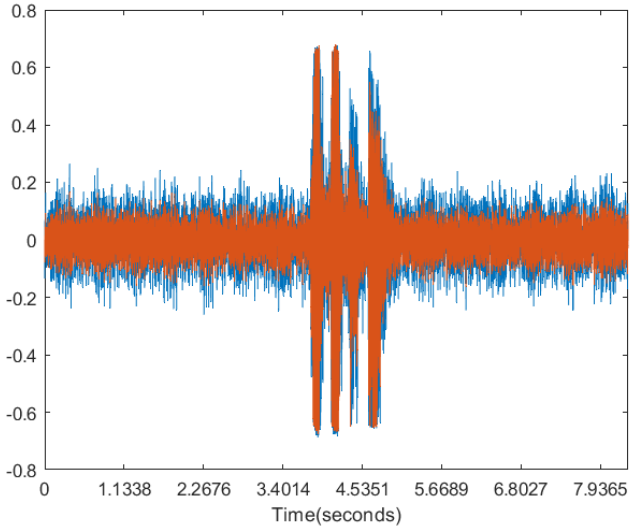

Figura 4.5: grafico segnali scaricati

Questi segnali sono stati registrati da due microfoni distanti 15 cm per una durata di 8 secondi ad una frequenza di 48 KHz; la sorgente è ad un angolo theta pari a 45 gradi.

Durante il codice sono stati esaminati i due diversi algoritmi spiegati nel capitolo 3.

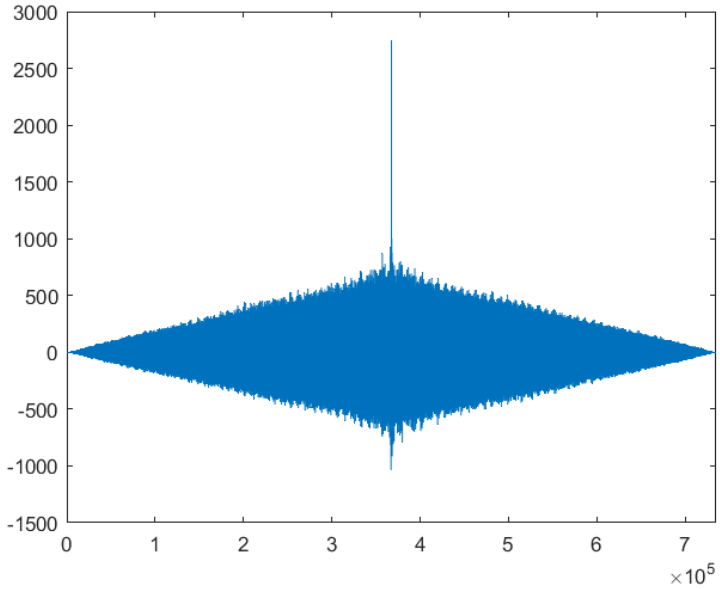

Figura 4.6: cross-correlazione segnali precedenti

Questa figura è la cross-correlazione normale(ovviamente calcolandola sia nel dominio del tempo che della frequenza il risultato non cambia) mentre la seguente è la cross-correlazione pesata con il metodo descritto precedentemente nel capitolo 3.2.

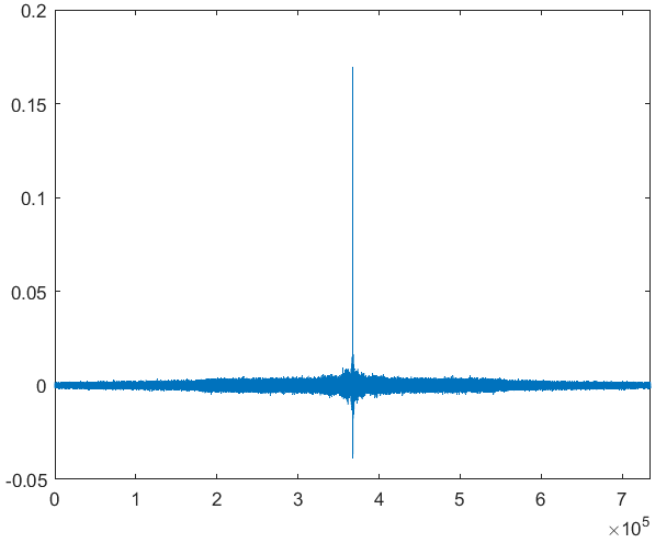

Figura 4.7: cross-correlazione normalizzata segnali precedenti

Il risultato è lo stesso ma si può notare come la correlazione pesata abbia un picco più evidente in questo caso.

Una volta trovato il picco è possibile utilizzare la formula del paragrafo 3.3 per calcolare l'angolo di arrivo.

```
[~, ind] = max(gcc);
tau = (ind-length(mic1))/f1
```

```
tau = 2.9478e-04
```

```
theta = asin(tau/max_tau)*180/pi;
disp(theta)
```

```
   46.2732
```

Figura 4.8: risultato algoritmo TDOA

Dove "max_tau" equivale al rapporto tra la distanza fra i microfoni e la velocità del suono in aria, "f1" è la frequenza utilizzata per il campionamento e "ind" è il valore in corrispondenza del picco della correlazione.

Il database segnava un angolo di arrivo di circa 45 gradi quindi la precisione è soddisfacente.

4.6 **Prove con dati registrati**

Durante i momenti finali del tirocinio sono stati eseguiti test utilizzando i materiali descritti negli scorsi capitoli.

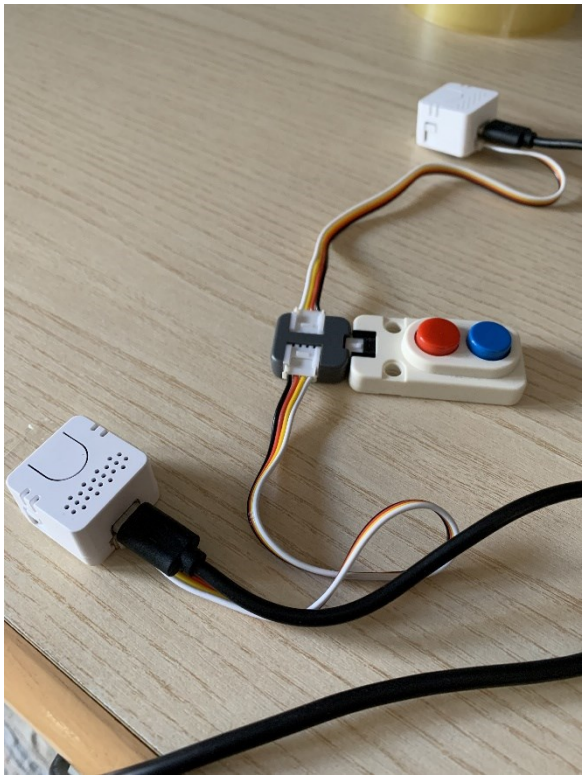

Figura 4.9: utilizzo M5Atom ECHO come microfoni

Molte prove sono state effettuate con la configurazione mostrata nella figura precedente; all'inizio sono state fatte anche con la configurazione presente in figura 4.1 ma ciò comportava utilizzare segnali con campioni non maggiori ai 500/1000 in quanto, da un lato, era dispendioso temporalmente il trasferire dati attraverso la porta I2C, dall'altro, come già accennato prima, il processo per calcolare la correlazione impiega un elevato utilizzo della potenza di calcolo dell'hardware.

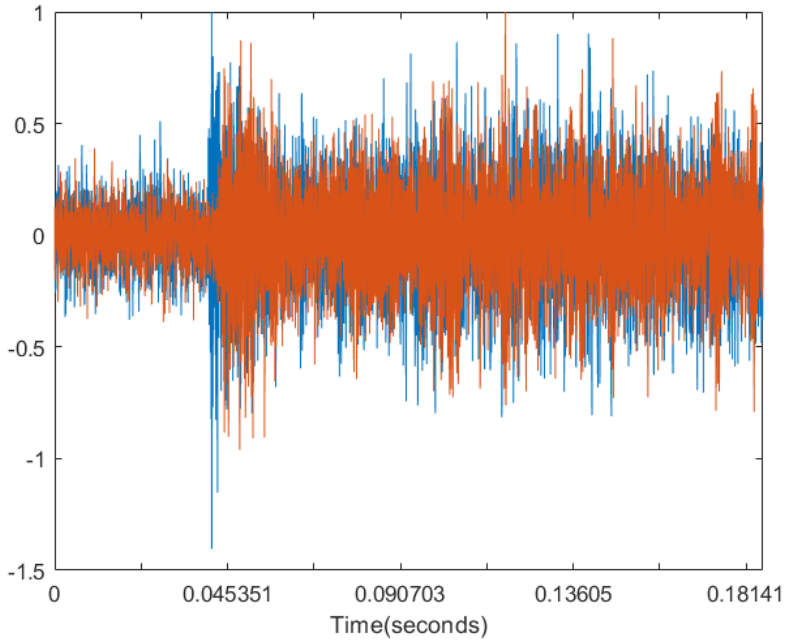

Figura 4.10: esempio di dati raccolti

Nella figura soprastante sono presenti i due segnali raccolti dai microfoni, si può notare come è effettivamente presente un ritardo.

L'asse delle ordinate indica l'ampiezza dei segnali (precedentemente è stata rimossa la media e sono stati normalizzati ai rispettivi valori massimi) mentre l'asse delle ascisse indica il tempo in secondi (i campioni sono 8192 con una frequenza di campionamento di 44100 Hz quindi l'intervallo temporale dei due segnali precedenti è di 8192 x 1 / 44100 = 0,1857 secondi circa).

Successivamente è stata calcolata la cross-correlazione.

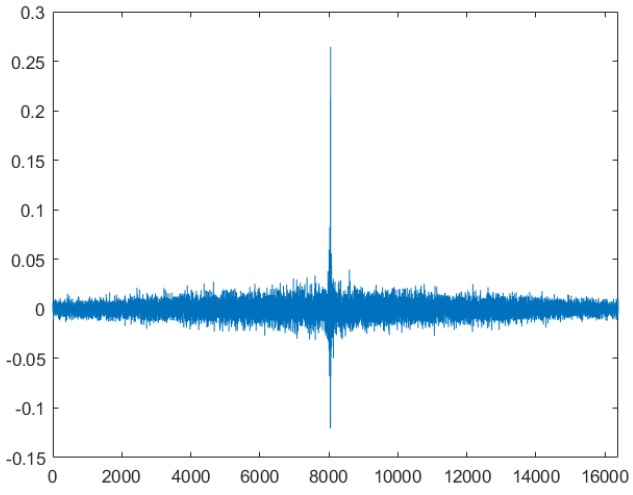

Figura 4.11: GCC dati precedenti

Anche in questo caso sembra che la correlazione sia giusta, in realtà è presente un problema che si può osservare nella prossima foto.

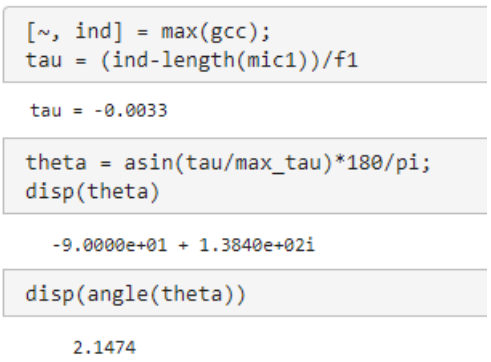

Figura 4.12: errore direzione

Questo accade perché è presente un ritardo dovuto all'hardware tra i due microfoni, quindi si ottiene un valore tau > max_tau per cui non è definito l'inversa del seno per valori maggiori di uno.

Il ritardo è causato dal fatto che i due M5StickC (o, analogamente, gli M5Atom ECHO) contengono due processori indipendenti quindi variazioni anche di pochi millisecondi possono inficiare la misurazione.

Dopo aver rilevato questo problema è stato eseguito un tentativo con gli interrupt per identificare subito il segnale di registrazione (è stato tentato anche di eseguire la read dell'I2S ma ciò non è stato possibile in quanto il processore non può rimanere fermo durante la registrazione dato che la frequenza di registrazione è di diversi ordini di grandezza rispetto a quella del processore).

CONCLUSIONI

Si è verificato che gli algoritmi sopra citati sono funzionanti per due o più microfoni indipendenti.

Si è notato come è necessario avere microfoni esterni ai microprocessori, come ad esempio l'M5StickC, per problemi dovuti al ritardo che si crea tra essi come, ad esempio, utilizzare più microfoni esterni collegati ad una Raspberry o un DSP per ottenere anche una potenza di calcolo più elevata.

Dopo aver costruito un sistema con i requisiti di cui sopra si potranno fare prove quasi real-time (il real-time è relativamente difficile da ottenere a causa dell'elevato numero di calcoli da effettuare) sia in aria che in acqua, dato che cambia solamente la velocità del suono, utilizzando gli algoritmi sopra descritti.

Una volta riusciti gli esperimenti con la cross-correlazione e GCC è anche possibile andare più a fondo utilizzando algoritmi più recenti e performanti.

Inoltre i microfoni che verranno utilizzati dovranno avere una risposta in frequenza che riesca ad amplificare anche le frequenze più elevati, per i delfini ad esempio è utile avere una frequenza di campionamento anche fino ai 100KHz.

Una volta riusciti ad identificare la direzione di arrivo del suono è possibile montare il sistema di localizzazione in un robot, terreste o sottomarino, ed attraverso un sistema di controllo fare in modo che segua una sorgente sonora.

BIBLIOGRAFIA

Chen L., Liu Y., Kong F. & He N., 2011, Acoustic Source Localization Based on Generalized Cross-correlation Time-delay Estimation.

Chaudhary K. N., Verma S., Aditya A., Sound Source Localization using GCC-PHAT with TDOA Estimation.

Liou F. J., Joshi K., Vador G., Real-Time Sound Source Localization

Chung M., Chou H. & Lin C., Sound Localization Based on Acoustic Source Using Multiple Microphone Array in an Indoor Environment

Liaquat M. U., Munawar H. S., Rahman A., Qadir Z., Kouzani A., Mahmud M. A. P., Localization of Sound Sources: A Systematic Review.

José F. Valente, José C. Alves, Real-Time TDOA measurements of an underwater acoustic source

Lee R., Kang M., Kim B., Park K., Lee S. & Park H., Sound Source Localization Based on GCC-PHAT with Diffuseness Mask in Noisy and Reverberant Environments

Parsayan A., Ahadi M. S., 2010, TDE-ILD-Based 2D Half Plane Real Time High Accuracy Sound Source Localization Using Only Two Microphones and Source Counting.