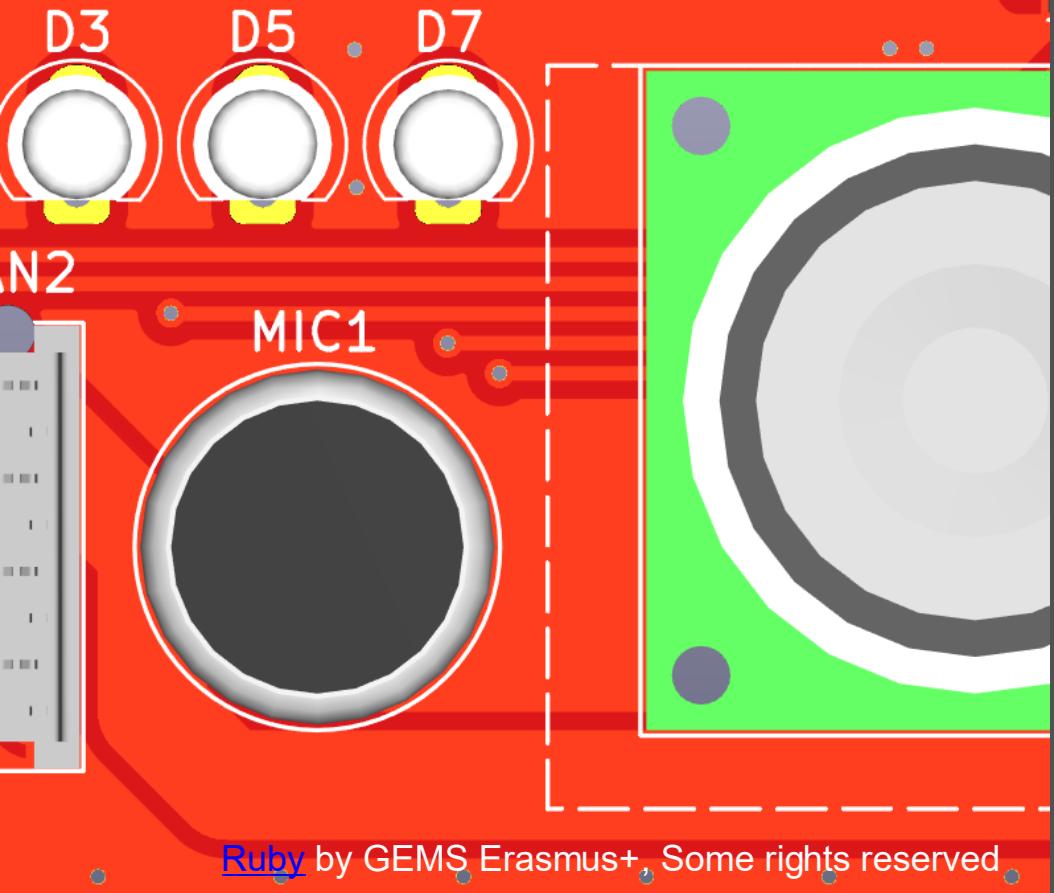


**ERASMUS+  
GEMS  
RUBY**



# Sound source localization

Tutorial

Daniel Pizarro Pérez (University of Alcalá)



# Index

Objectives

Basic concepts

Direction of Arrival (DoA) detection

DoA estimation example in Ruby

Conclusions

References

# Objectives

**This tutorial is focused on**

- Understanding the basic concepts of audio propagation.
- Sound source direction estimation with a pair of microphones.
- Example code in Arduino platform.

**This tutorial is not focused on**

- General multi-source localization
- Propagation in close-field conditions and reverberant scenarios

# Basic concepts (I)

## What is Sound?

**Definition:** Sound is a mechanical wave that results from the back-and-forth vibration of particles in a medium (such as air).

## Sound Waves:

**Type:** Longitudinal waves where particle displacement is parallel to the direction of wave propagation.

**Medium:** Air molecules transmit these pressure variations by colliding with neighboring molecules, transferring energy through the medium.

## Speed of Sound:

**Factors:** Depends on the medium's properties, such as temperature, pressure, and density.

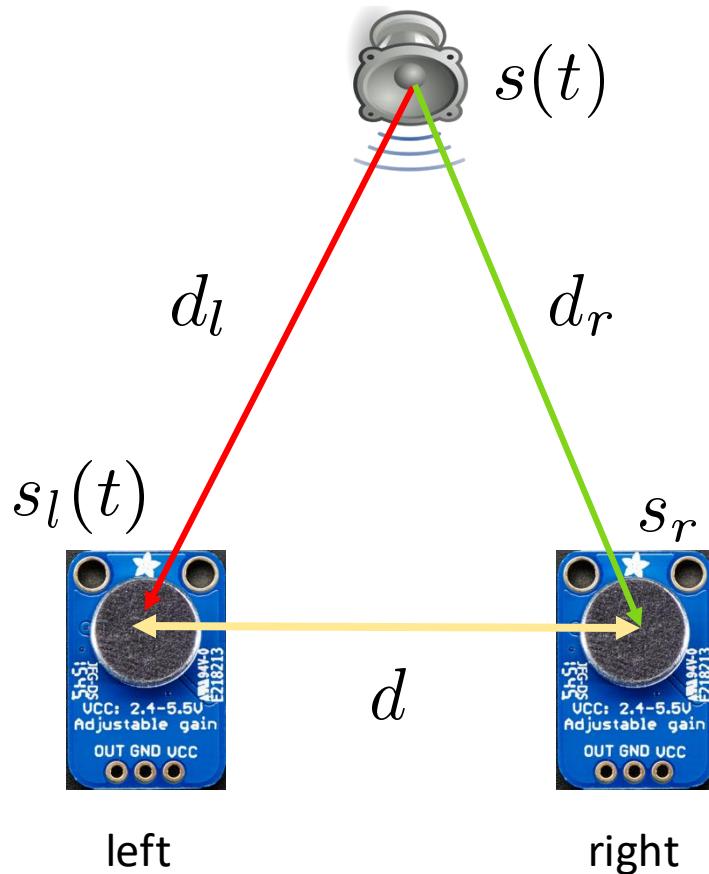
**In Air:** Approximately 343 meters per second (m/s) at room temperature (20°C).

Solids at room temperature		Liquids		Gases at 1 atm pressure
Titanium	6070 m/s	Sea water 25°C	1534 m/s	Hydrogen 27°C 1320 m/s
Nickel	6040 m/s	Water 25°C	1497 m/s	Helium 25°C 973 m/s
Steel	5940 m/s	Mercury 25°C	1450 m/s	Methane 27°C 450 m/s
Cast iron	4994 m/s	Ethanol 20°C	1159 m/s	Nitrogen 27°C 353 m/s
Brass	4700 m/s	Methanol 20°C	1116 m/s	Dry air 25°C 346 m/s
Silver	3650 m/s			Dry air 20°C 343 m/s
Platinum	3260 m/s			Oxygen 27°C 330 m/s
Gold	3240 m/s			Carbon dioxide 0°C 258 m/s
Polystyrene	2350 m/s			
Rubber	1600 m/s			

<https://theory.labster.com/sound-speed-table-dbs/>

# Basic concepts (II)

## Sound Source and a pair of microphones



$d_l$  Distance from source to left microphone

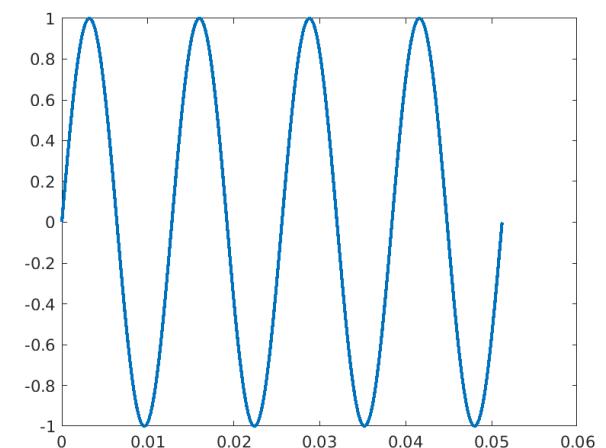
$d_r$  Distance from source to right microphone

$d$  Distance from left to right microphones

$s(t)$  Sound signal emitted by the source

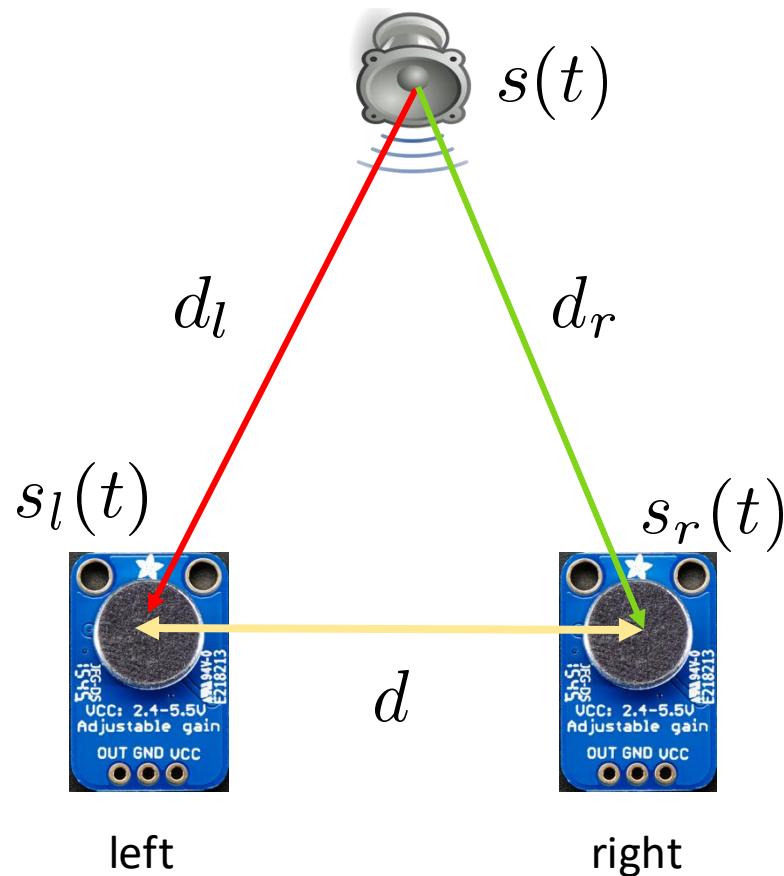
$s_l(t)$  Sound signal received at the left microphone

$s_r(t)$  Sound signal received at the right microphone



# Basic concepts (III)

## Sound Source and a pair of microphones



Assumption: The signals received at the microphones are delayed versions of the source signal

$$s_r(t) = s(t - \Delta_r)$$

$$s_l(t) = s(t - \Delta_l)$$

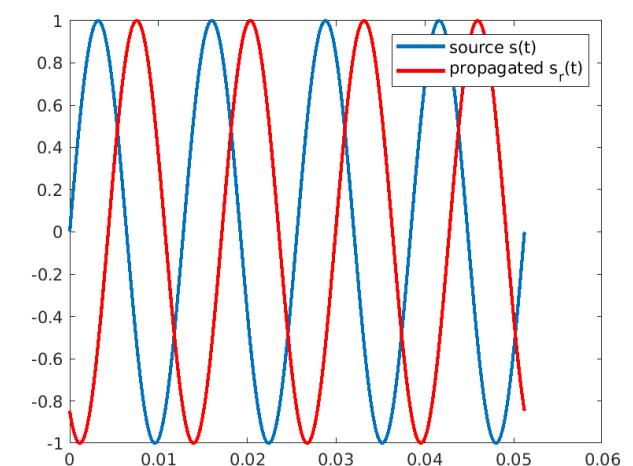
$v_s$  Speed of sound (334 m/s at 20°)

Example:

Source at 10 meters from the microphone

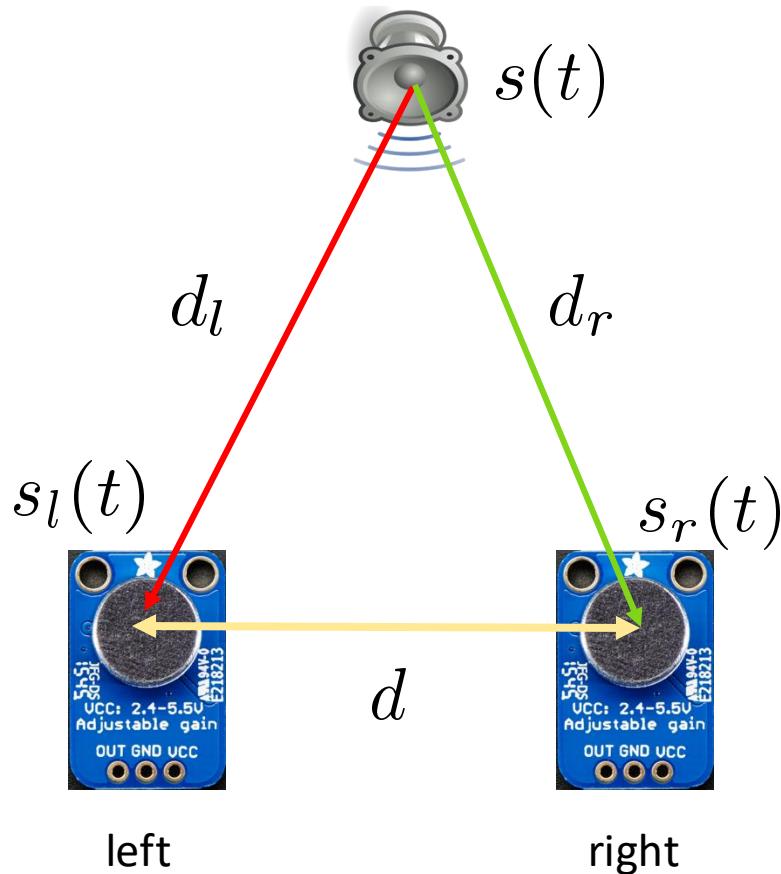
Time delays due to sound propagation

$$\Delta_r = \frac{d_r}{v_s} \quad \Delta_l = \frac{d_l}{v_s}$$



# Basic concepts (IV)

## Sound Source and a pair of microphones



Assumption: The signal received at the left microphone is a delayed version of the signal received at the right microphone.

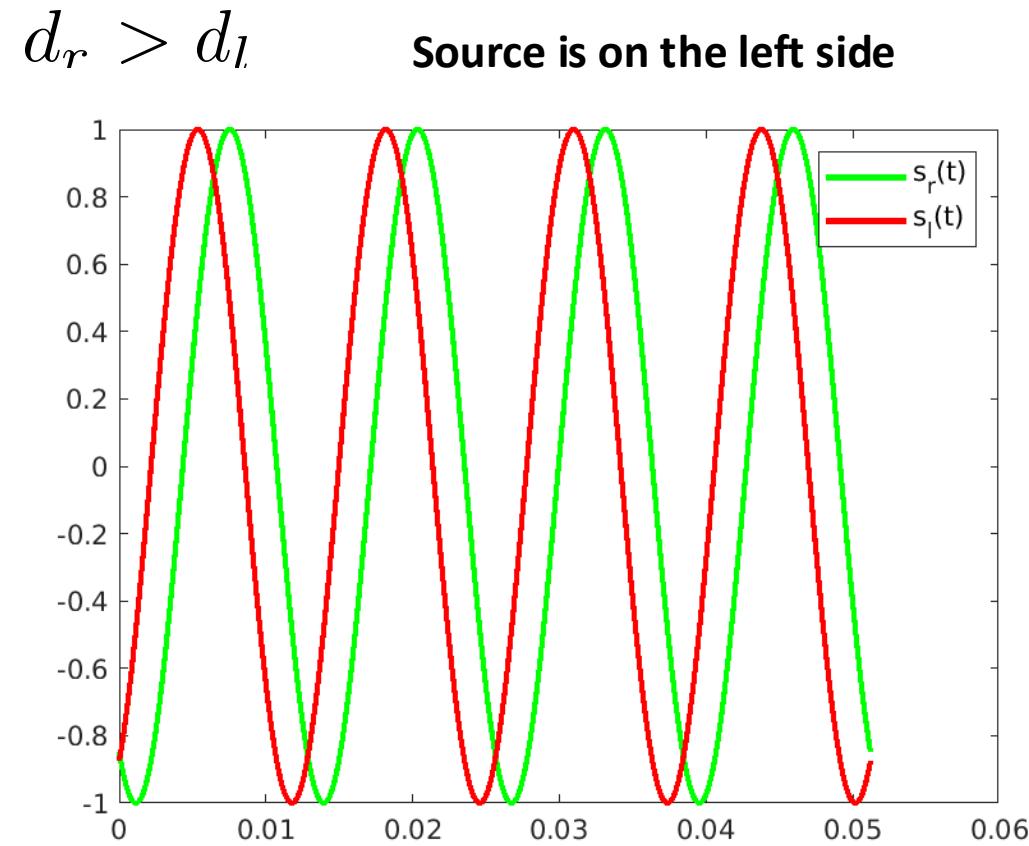
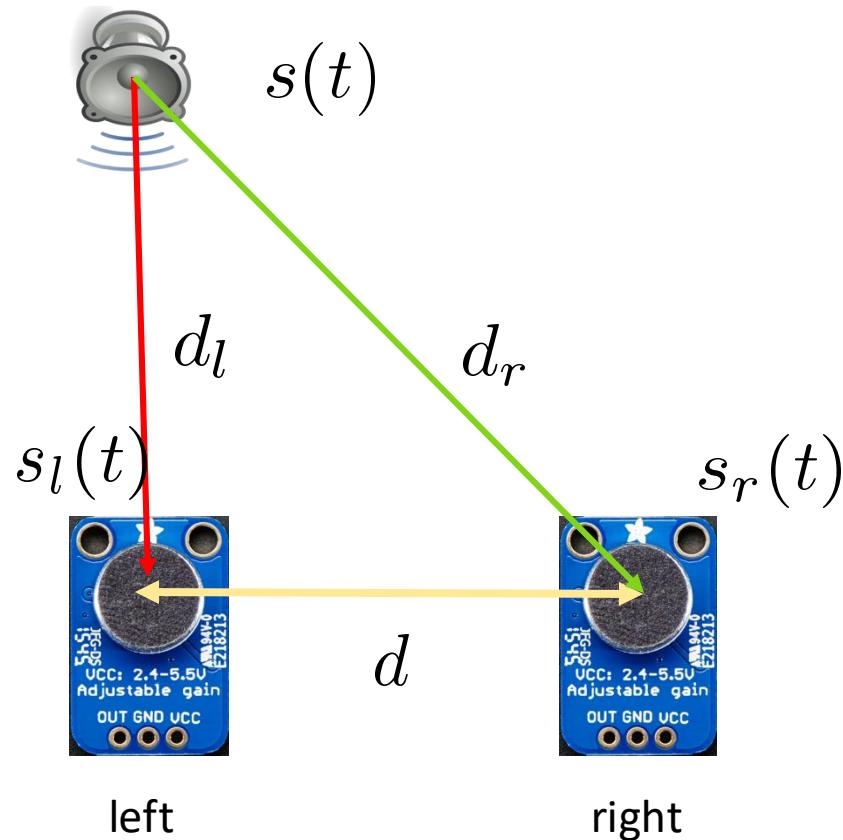
$$s_l(t) = s_r(t + \Delta_r - \Delta_l)$$

$$\Delta_r - \Delta_l = \frac{d_r - d_l}{v_s}$$

The delay is proportional to the difference of distances

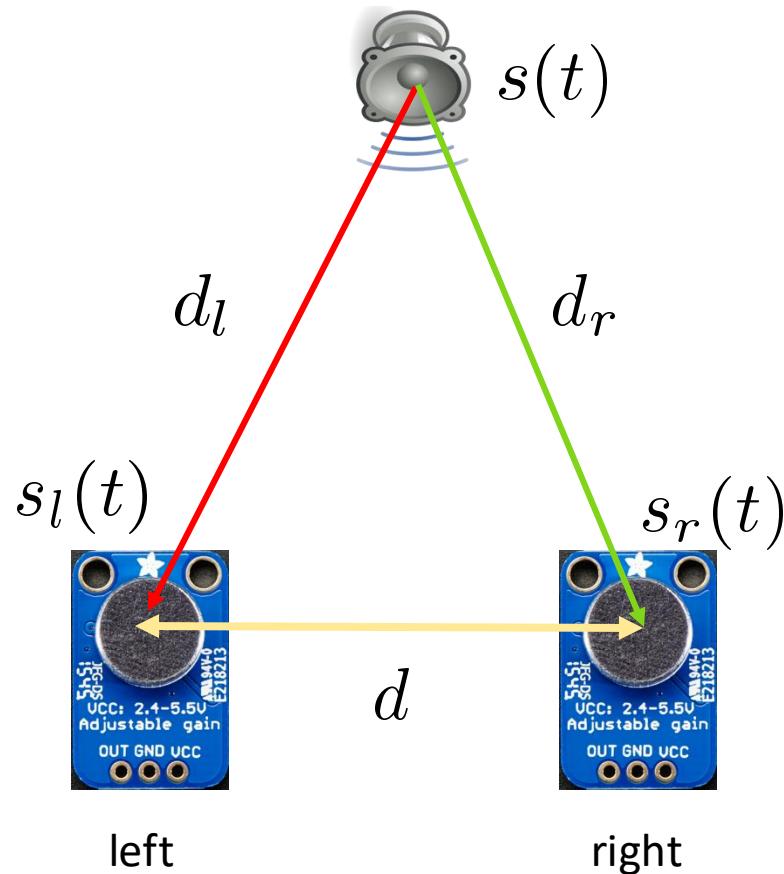
# Basic concepts (IV)

## Sound Source and a pair of microphones

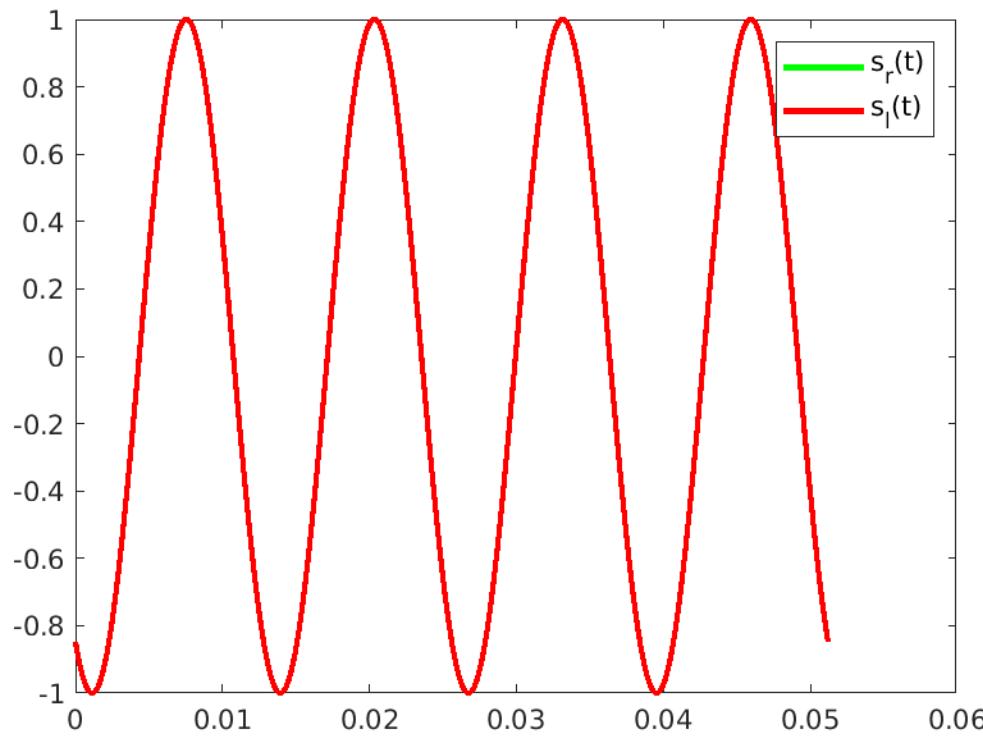


# Basic concepts (IV)

## Sound Source and a pair of microphones

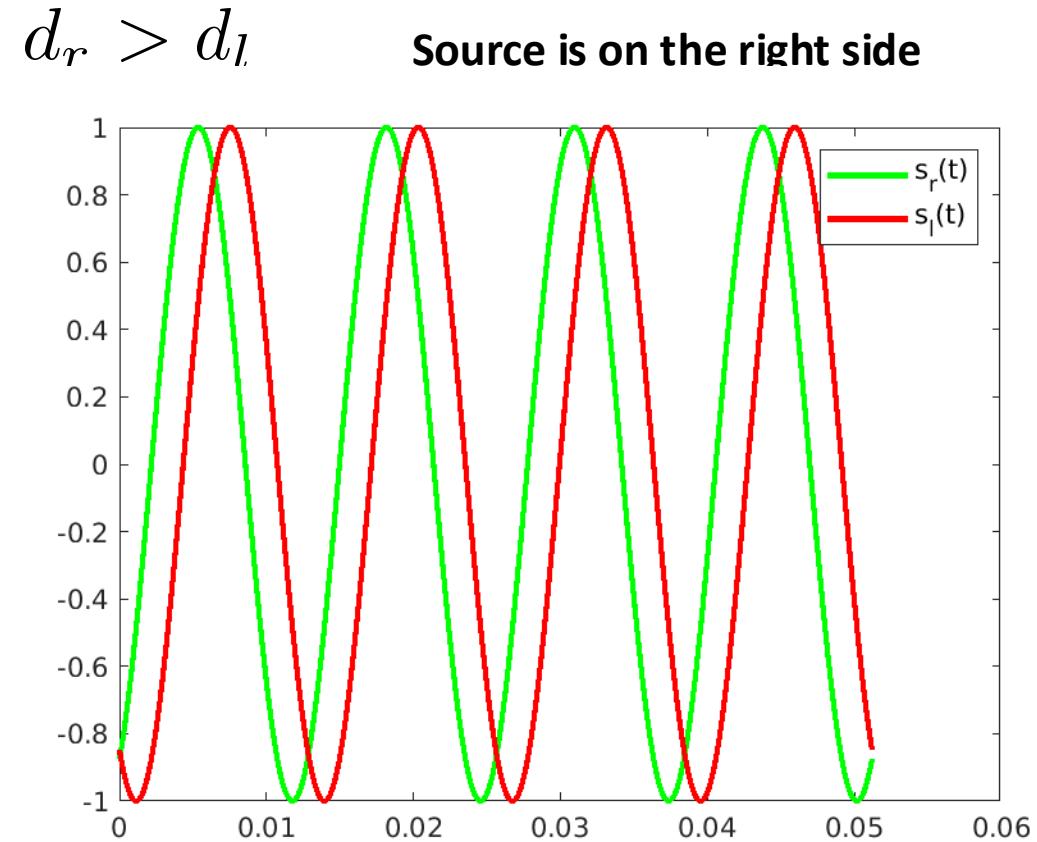
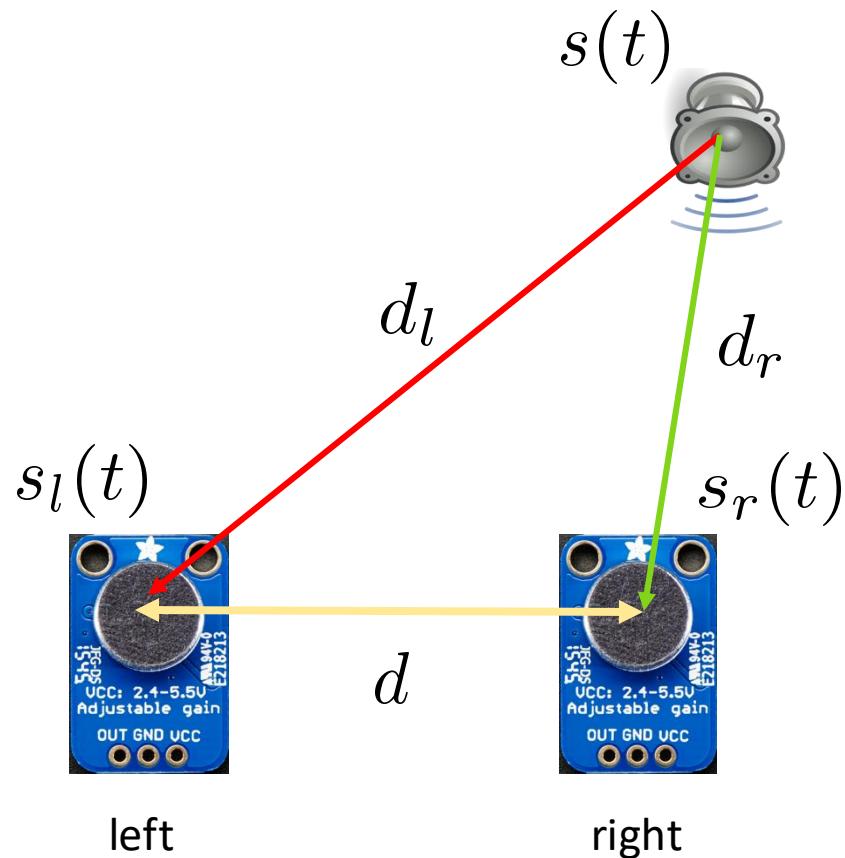


$$d_r = d_l \quad \text{Source is centered}$$



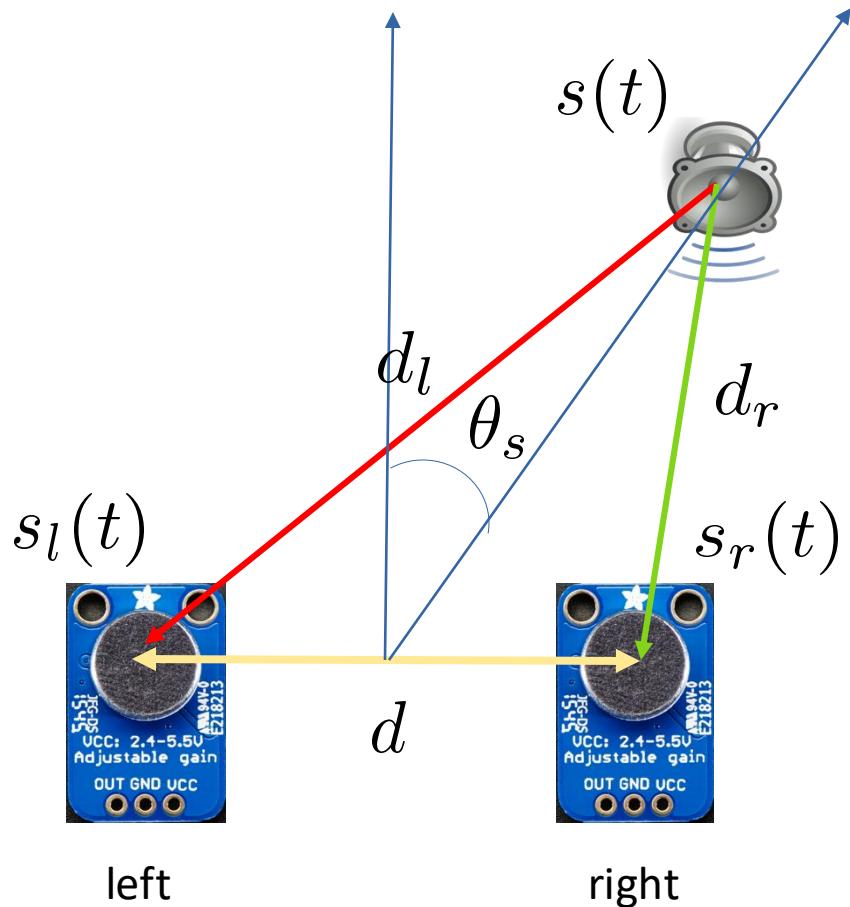
# Basic concepts (IV)

Sound Source and a pair of microphones



# Direction of Arrival (DoA) detection

## Direction of Arrival (DoA) [2]



$\theta_s$       **Source DoA angle.**

Direction of the source with respect to the microphones

$$\theta_s = \sin^{-1} \left( \frac{\Delta_r - \Delta_l}{d} v_s \right)$$

**Assuming**  $d_r, d_l \gg d$

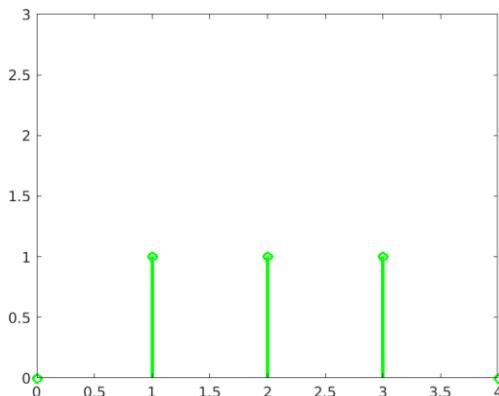
# Direction of Arrival (DoA) detection

What is the correlation between two sampled signals ?

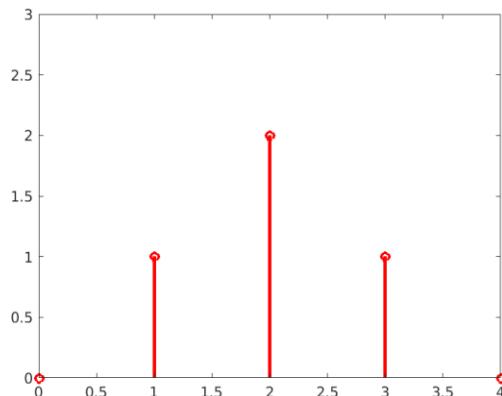
$$c(t) = \sum_{i=0}^{N_s} s_l(t_i) s_r(t + t_i)$$

$$c(0) = s_r(0)s_l(0) + s_r(1)s_l(1) + s_r(2)s_l(2) + s_r(3)s_l(3) + s_r(4)s_l(4) = 4$$

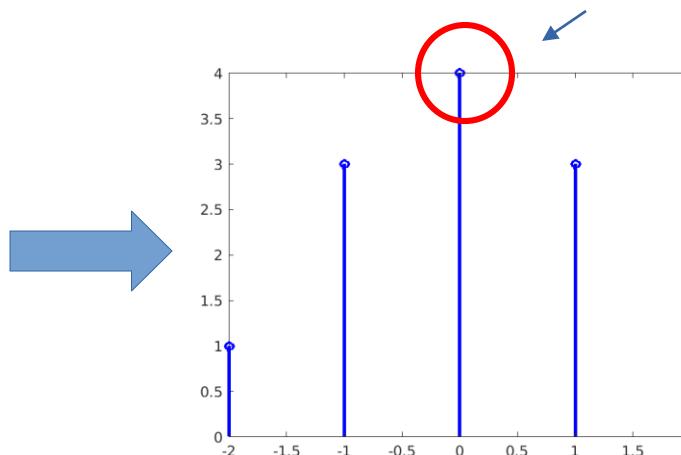
Example



$$s_l(t)$$



$$s_r(t)$$

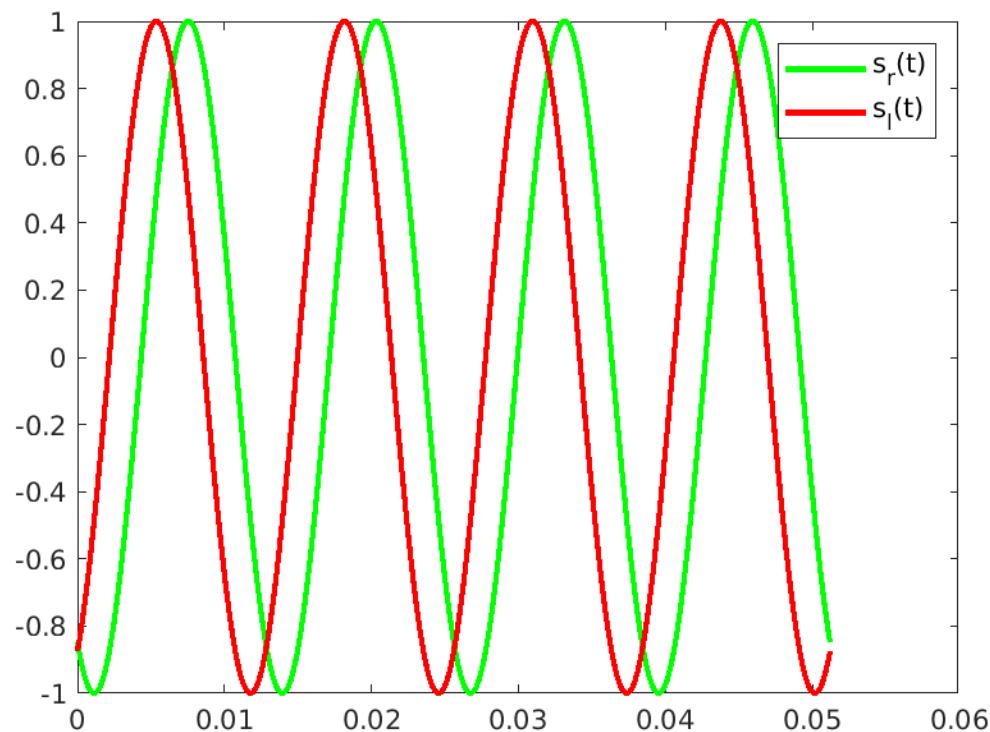


$$c(t)$$

# Direction of Arrival (DoA) detection

Measuring the delay difference with correlation

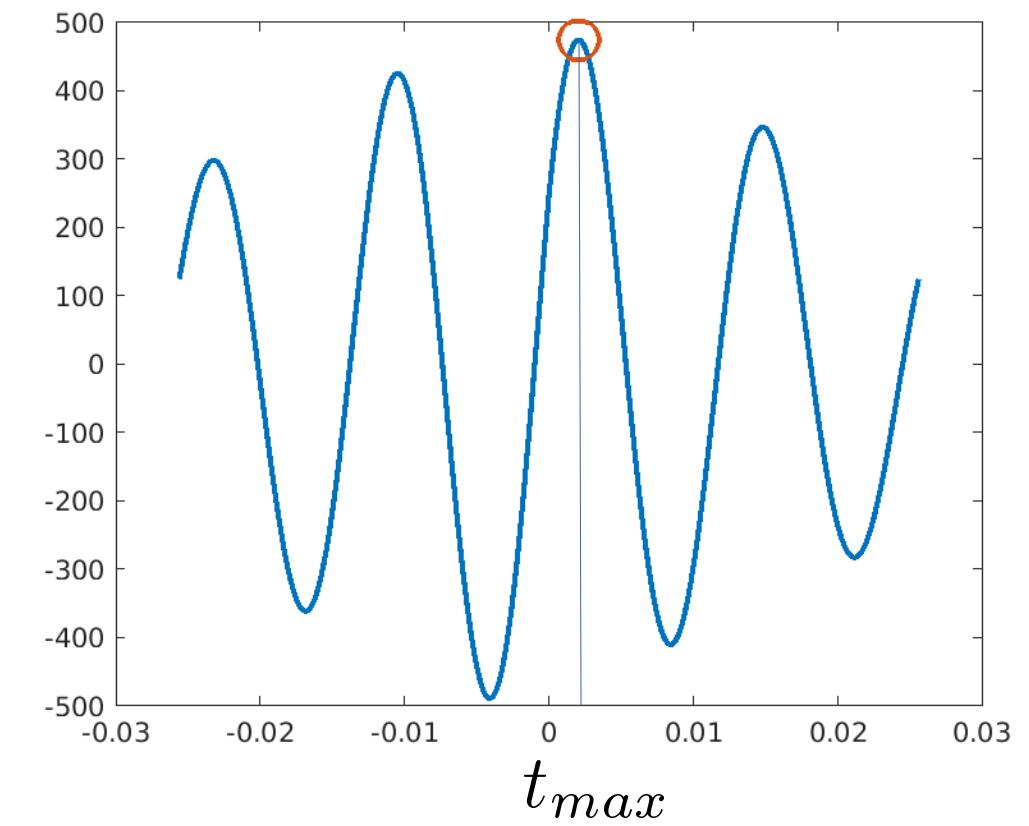
$$d_r > d_l$$



correlation

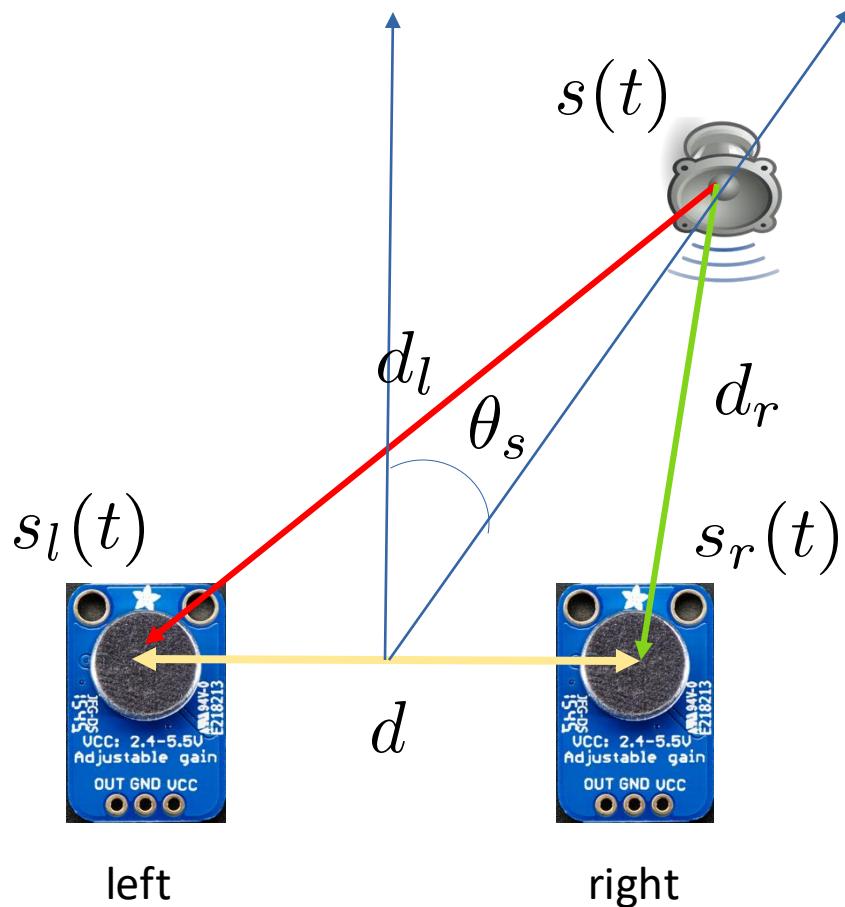


$$\Delta_r - \Delta_l = t_{max}$$



# Direction of Arrival (DoA) detection

Sound Source and a pair of microphones



Algorithm to find the DoA [2]

1) Capture the left and right microphone signals  $s_l(t)$   $s_r(t)$

2) Compute correlation

$$c(t) = \sum_{i=0}^{N_s} s_l(t_i) s_r(t + t_i)$$

3) Compute the time index where the correlation is maximum

$$t_{max} = argmax(c(t))$$

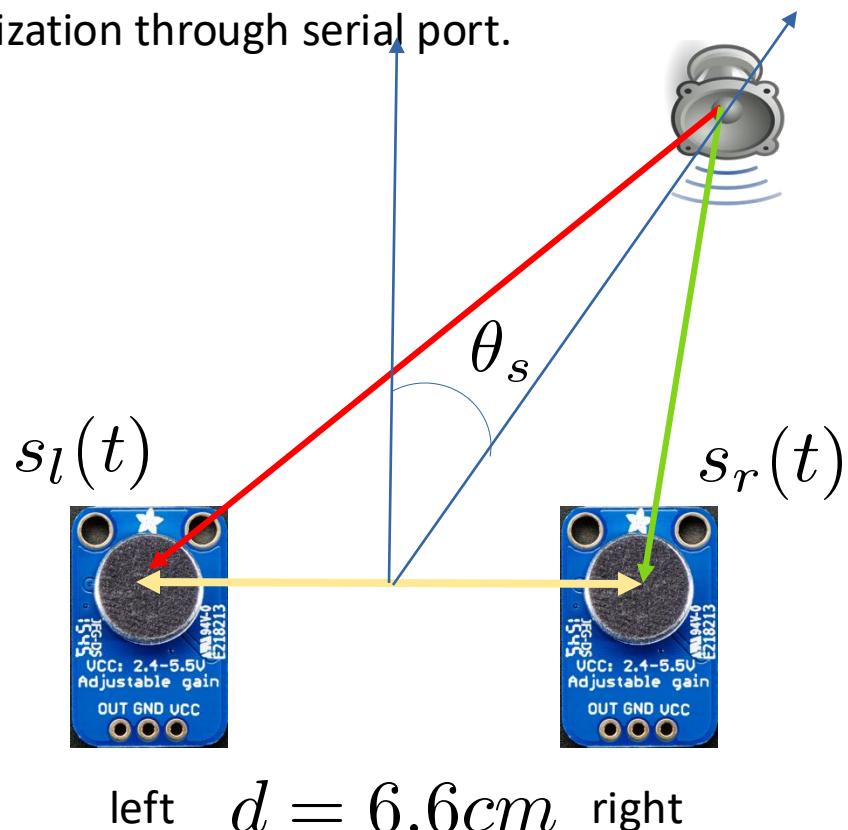
4) Compute the DoA angle

$$\theta_s = arcsin \left( \frac{t_{max}}{d} v_s \right)$$

# DoA estimation example in Ruby

## Objective:

- Capturing audio signals from two microphones connected to ADC1 channels 0 and 1.
- Resolution: 12 bits Sampling frequency: 30 KHz
- Compute Correlation and its maximum
- Compute DoA angle.
- Visualization through serial port.



## Environment:

Arduino + ESP32 libraries

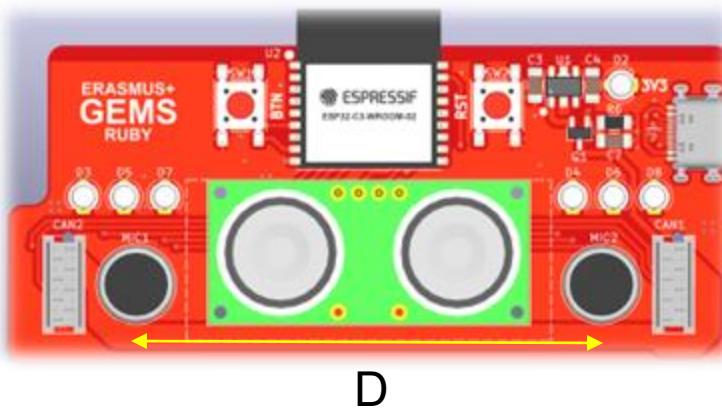
A screenshot of the Arduino IDE interface. The title bar shows "ESP32C3 Dev Module". The left sidebar lists files: "sketch\_jul22a.ino" (selected), "sketch\_jul22a.ino", and "sketch\_jul22a.ino". The main code area contains the following:

```
File Edit Sketch Tools Help
✓ → ↻ USB ESP32C3 Dev Module ...
sketch_jul22a.ino
1 void setup() {
2 // put your setup code here, to run once:
3
4
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9
10 }
```

# DoA estimation example in Ruby

## Code breakdown: Initialization and Setup

- Constants and Variables:
  - CONVERSIONS\_PER\_PIN: Number of conversions per pin.
  - Nsamples: Number of samples to collect.
  - LAGMAX: size of correlation buffer
  - Fs: sampling frequency
  - Vs: speed of sound
  - D: distance between microphones
  - Arrays sleft and sright for storing samples.
  - ADC Pin Configuration:
  - Correlation buffer
  - Tmax and DoA angle variables



```
1 #define CONVERSIONS_PER_PIN 1
2 uint8_t adc_pins[] = {1, 0};
3 #define Nsamples 725
4 #define LAGMAX 30
5 #define PI 3.1415926
6 #define D 0.066
7 #define vs 334.0
8 #define fs 30000
9 int correlation[2*LAGMAX+1];
10 int n;
11 int sleft[Nsamples];
12 int sright[Nsamples];
13 double tmax=0;
14 double angle=0;
15 uint8_t adc_pins_count = sizeof(adc_pins) / sizeof(uint8_t);
16 volatile bool adc_coversion_done = false;
17 adc_continuous_data_t *result = NULL;
18
```

# DoA estimation example in Ruby

## Code breakdown: Setup Function

- Serial Communication: Initialized at 115200 bits per second.
- ADC Configuration:
  - Set resolution to 12 bits.
  - Set attenuation to 11db.
  - Configure continuous ADC with pins, conversion count, frequency (fs Hz x 2 channels), and ISR callback.
  - Start continuous ADC conversions.

```
1 void setup() {  
2   Serial.begin(115200);  
3   analogContinuousSetWidth(12);  
4   analogContinuousSetAtten(ADC_11db);  
5   analogContinuousadc_pins, adc_pins_count, CONVERSIONS_PER_PIN, fs*2, &adcComplete);  
6   analogContinuousStart();  
7 }
```

# 2-channel Audio Acquisition example

## Code breakdown: Data Processing in Loop Function

- ISR Flag Check: Processes data if conversion is complete.
- Data Reading and Storage: Stores mV values in sleft and sright.
- Cross-Correlation and DoA Calculation:
  - Computes cross-correlation.
  - Finds the maximum position to calculate time delay.
  - Computes the DoA angle using the time delay and distance between microphones.

```
1 void loop() {
2   if (adc_coversion_done == true) {
3     adc_coversion_done = false;
4     if (analogContinuousRead(&result, 0)) {
5       sleft[n] = result[0].avg_read_mvolts - 1200;
6       sright[n] = result[1].avg_read_mvolts - 1200;
7       n++;
8     } else {
9       Serial.println("Error occurred during reading data.");
10    }
11  }
12
13 if(n >= Nsamples) {
14   analogContinuousStop();
15   n = 0;
16   calculate_cross_correlation(sleft, sright, Nsamples, LAGMAX, correlation);
17   tmax = (((double)find_max_position(correlation, 2*LAGMAX+1))-LAGMAX+1)/((double)
fs);
18   double argument = (tmax/D)*vs;
19   if(argument>1.0) argument=1.0;
20   if(argument<-1.0) argument=-1.0;
21   angle = asin(argument)*180.0/PI;
22   Serial.printf("tmax=%2f,angle=%2f\n", tmax, angle);
23   delay(1000);
24   analogContinuousStart();
25 }
26 }
27 }
```

# 2-channel Audio Acquisition example

## Experiment

- 1 KHz pure tone audio source
- Visualization in Arduino Serial Plotter



tmax=0.00003, angle=9.98, arg=0.17323

tmax=-0.00010, angle=-31.31, arg=-0.51970



tmax=0.00020, angle=90.00, arg=1.03939



# Conclusions

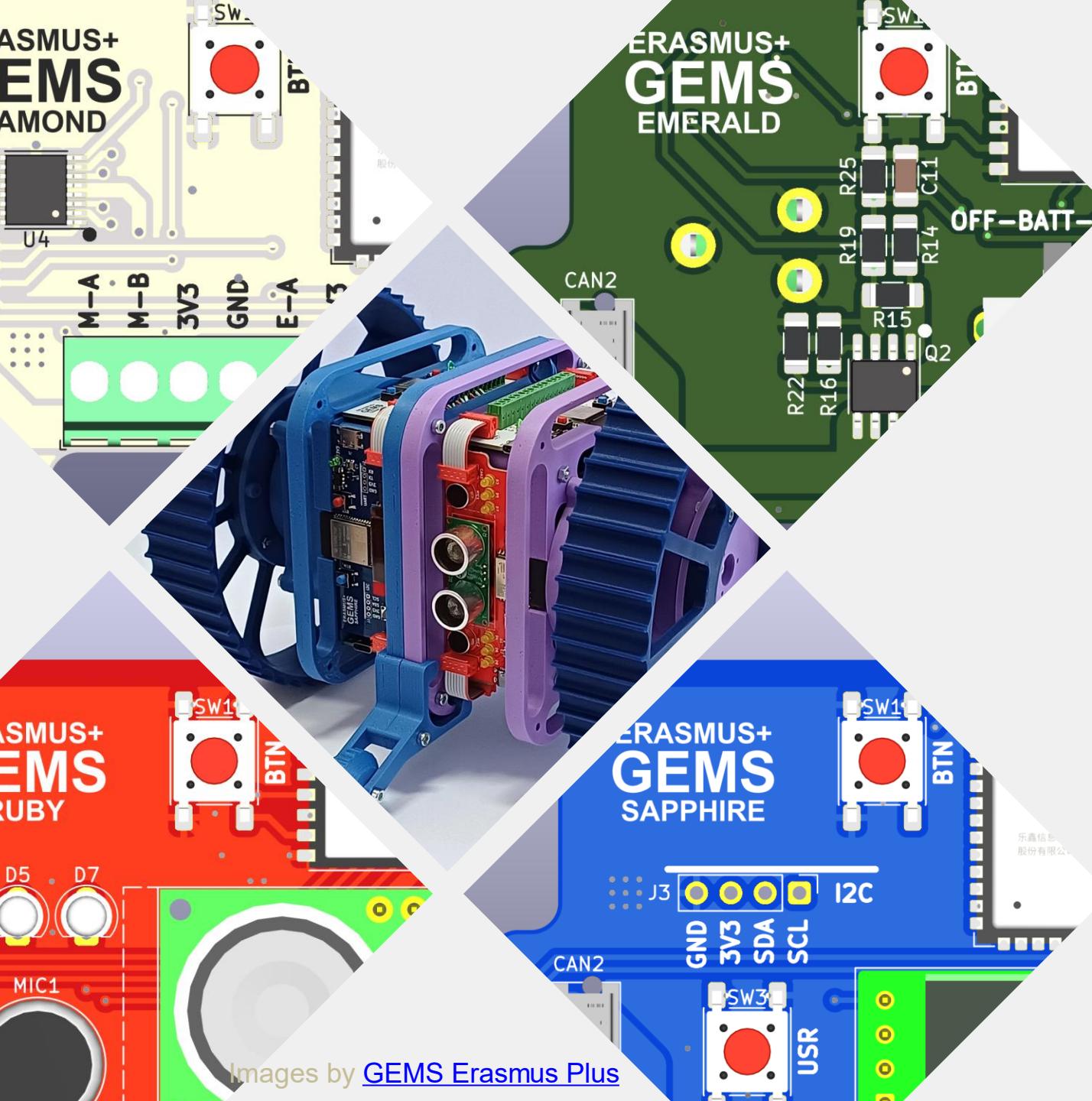
- Effective Signal Capture: Successfully captured audio signals from two microphones using ESP32's ADC in continuous mode.
- Cross-Correlation Analysis: Applied cross-correlation to determine the time delay between signals from two microphones.
- Direction of Arrival (DoA) Estimation: Calculated the DoA of the sound source using the time delay and physical separation of the microphones.
- Practical Applications: Techniques learned can be applied to real-world scenarios such as sound localization, robotic navigation, and acoustic monitoring.

# References

[1] Table of Sound Speeds.

<https://theory.labster.com/sound-speed-table-dbs/>

[2] Y. A. Huang, J. Benesty, and J. Chen, “**Time delay estimation and source localization**,” in Springer Handbook of Speech Processing. Springer, Berlin, Heidelberg, pp. 1043–1063, 2008. DOI:10.23919/Eusipco47968.2020.9287466



# Thank you for watching!

This video was created by University of Alcala for the GEMS Erasmus+ project (a collaboration between University of Ljubljana, University of Alcala, Teaching Factory Competence Center, and Delft University of Technology). It is released under a [Creative Commons Attribution Non Commercial Share Alike 4.0 International License](#)



**Co-funded by  
the European Union**

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CMEPIUS. Neither the European Union nor the granting authority can be held responsible for them.