Sound Localization System

Bíró Mátyás-Péter

Student, Technical University of Cluj-Napoca, Computer Science (2021-2025)

E-mail: biromp6@proton.me

**Soós Dávid**

Student, Technical University of Cluj-Napoca, Computer Science (2021-2025)

E-mail: soos.sa.david@student.utcluj.ro

**Sprencz Róbert**

Student, Technical University of Cluj-Napoca, Computer Science (2021-2025)

E-mail: sprencz.zs.robert@student.utcluj.ro

**Szabó Loránd**

Student, Technical University of Cluj-Napoca, Computer Science (2021-2025)

E-mail: [szabo.zo.lorand@student.utcluj.ro](mailto:szabo.zo.lorand@student.utcluj.ro)

**Abstract.** The purpose of our work was to create a system that can locate the origin of a sound signal. It is assumed that there is a single sound source during the localization process. This implementation uses an Arduino Uno board and R.22112 microphones.

1. Introduction

In the following paper we will discuss an implementation of a Sound Localization System, together with observations, ideas, and experiences. We also share our data and the assembly method of our circuit.

The assembly follows a simple logic: we use four microphones as sensors and four LEDs as actuators, all of them being connected to the main Arduino Uno board. To be able to monitor the data read from the sensors, the Arduino maintains a serial connection to a computer through a USB-port.

|  |
| --- |
| Diagram  Description automatically generated |
| **Fig. 1.1 – Block Diagram** |

During each data read cycle, the Arduino measures the average voltage returned by each microphone for a given period and sends this information to the computer. The built-in serial plotter from the Arduino IDE plots these results, letting us obtain a live feed of the sound intensities. This data is later processed to determine the location of the sound source.

|  |
| --- |
|  |
| **Fig. 1.2 – Sound Intensity Measured by the Sensors** |

1. Hardware

The required hardware elements consist of:

* 1 x Arduino Uno – *Figure 2.*
* 1 x USB A – USB B cable – *Figure 2.2*
* 4 x 22112 microphone modules – *Figure 2.3*
* 4 x LEDs, together with resistors – *Figure 2.4*
* 1 x breadboard for prototyping
* jumper wires
* a computer for the programming of the Arduino board

|  |  |  |  |
| --- | --- | --- | --- |
|  | MC002722 - Multicomp - USB Cable, Type A Plug to Type B Plug, 1.8 m |  | Which Resistor Should I Use with my LED? - Kitronik Ltd |
| Fig. 2.1 | Fig. 2.2 | Fig. 2.3 | Fig. 2.4 |

1. Functionality

blah

1. Preparation and Configuration

The process of creating a Sound Localization System consists of the following steps:

1. Install Arduino IDE
2. Connect the microphones and the LEDs to the Arduino board
3. Connect the Arduino to the computer
4. Upload the code to the Arduino and run it

* During this stage, the measured volumes can be read from the display of the computer
* The position of the sound source can be determined graphically using the provided MATLAB code

1. Physical approach
   1. Scientific background

The process

The symbols in the above equation represent:

* – the measured distance
* – the speed of sound in air
* – the time needed for the sound waves to return to the sensor
  1. The sensor

Our work

|  |
| --- |
|  |
| **Fig. 6.** The diagram |

*5.3 The Relation Between the Sound Intensity and the Distance*

As you move away from a sound source, the sound gets quieter- especially when you are outdoors. For the relation between the two quantities the inverse square law holds true, more specifically:

I∝1/d2

We have used this for our graphical representation.

1. Conclusion

This work

1. Appendix

This section provides information for the better understanding of this work.

* 1. *The code*

The

|  |
| --- |
| //code |

* 1. *The final setup*

The

|  |
| --- |
|  |
| **Fig. 7x.** The final setup of the Sound Localization System |
|  |
|  |

1. Bibliography

[1] https://en.wikipedia.org/wiki/Sound\_localization

[2] <https://www.sigmanortec.ro/Modul-microfon-senzor-sunet-p126025149>

[3] https://pressbooks.pub/sound/chapter/intensity-and-distance-april-2019-version/