Mobile Robot with sound-based obstacle avoidance and echolocation

Louis Leroux, Jesper Rindom Jensen, Usama Saqib

Audio Analysis Lab, CREATE, Aalborg University, Denmark, July 2019

***Abstract* - Echolocation is a method which send sound and hear their echoes to localize or analyse the elements of environment to detect obstacle or to map a place. It is used by some animals like bats or artificially with sonars. In this article, the evolution of a robot which be able to move and record sound to map a room will be introduced. Indeed, Mems Microphones are recording sounds thanks to a Nucleo board programmed in Arduino. A kobuki robot is controlled by a user trough a python interface connected to an Arduino serial. Each electronic board are centralized into a program in python.**

1 Introduction

Robots evolved a lot over the past decade. They are used to assisting humans with different tasks, like moving in challenging and complex environments. The kobuki robot using here, is a robot which be able to move randomly with an external program available on their website or with an embedded program. The Mems microphones using here are SPW2430, there are produced and used by Adafruit and a lot of experiences has been made to record data with them. Aim of this project is to use the kobuki robot to move all around a room and sometimes stop to make sound using the SPW2430 microphones and record the echolocation. Subsequently, the user will analyse the data to determine where the obstacles are.

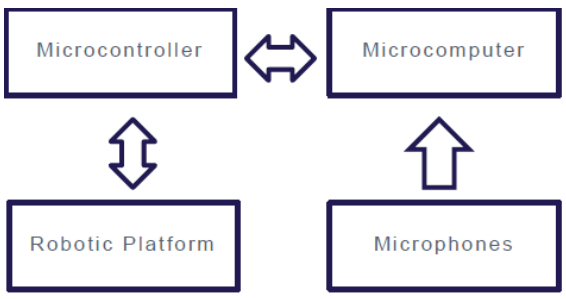

Figure 1: Overall System Architecture

"The robotic platforms discussed above has an on-board Arduino microcontroller that is connected to a proximity sensor, an accelerometer, a gyroscope as well as the motor control unit. These sensors would help us determine robot's orientation and speed. Moreover, these sensors and motor units could also be accessed from other microcontrollers that will help us navigate our robot based on the parameter we measure." - *Usama Saqib*. ROS could be also used to have access to these types of sensors and to control some abilities of the robot.

In this article, you will see how microphones were implemented, how the robot was programmed and the conclusion of everything which is done.

2 Mems microphones

To test the microphones we wanted to use it with I2S program of the Adafruit library, but it works only with their board. So, test was first on a simple Arduino but we did not have a good signal, that was very noisy and low because the sampling rate of the Arduino board was not high enough.

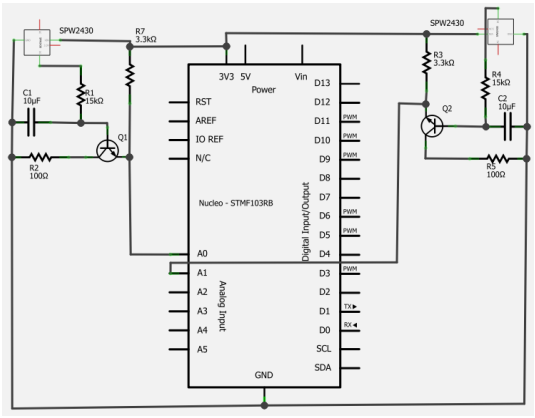

Figure 2: Mems microphones diagram

Therefore, we tried to use an amplifier, the LM358, but on the oscilloscope the signal was to low compare to a transistor with two resistors. So, we choose this type of configuration which have the role of amplifier. It is a linear amplifier, and he multiplied the signal by around 3/4 of the input signal. After that, we added a 10KHz high-pass filter because we would like to have only the audible sound and avoid noise. After all of this stuff, we are able to hear our voice with the microphones. We are using two Mems microphones.

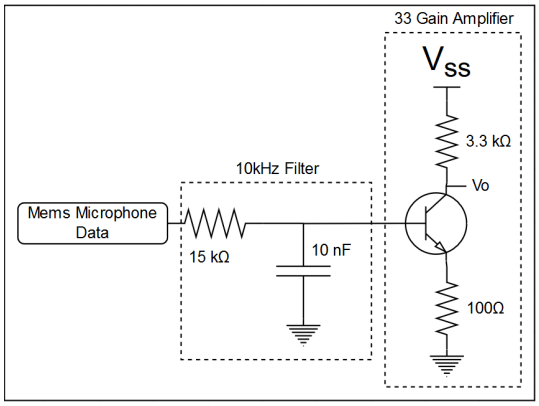

Figure 3: Filter and Amplifier

STM32F103RB board has been use after the Arduino one because we had a better output signal with this one. To record we are using 500000 baud-rate. After that we were able to record the data at the analog input of the STM32 board and put it in a text file using python. Indeed, we took the data from the serial of the Arduino, this one is stored inside an array which is clean by some functions to destroy noise or data which is not use for make a sound. Therefore, we were able to read the text file using Pyaudio on the data text recorded previously, we are able to split the two microphones and hear both of them separately. We are also able to register them into a wave file with the function winsound.

3 Kobuki robot

The aim of the kobuki robot was to control it using an external board. To achieve this we used a random walk available on the kobuki website[1] and we connected an external Arduino on the robot. We would like to use ROS on Ubuntu to have access on the gyroscope in the robot. It means that we should be able to know where the robot is in the room. However, we can't have access to the gyroscope with ROS now.

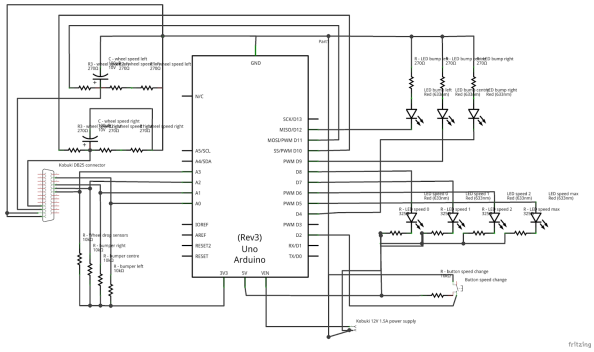

Figure 4: Kobukino diagram

The Arduino board is connect to the robot using the Serial port of the kobuki. The baud-rate used here is 115200. Thanks to this, we are able to control each wheels of the robot and also the push sensors. The three LED's on the top are used to know which sensors are pressed. The four LED's on the bottom are not all used, indeed only the first two are. If just the first LED is light, the robot will not move. Nevertheless, if the second is also light, he will move. Just after the robot was power on, we need to press "B1" button during two seconds. We also need to press the "BO" button to start the link between the program and the robot. To move the robot, the second led needs to be on. If its not, press the button on the breadboard.

The implementation of this program on Arduino was controlled by python. Indeed, Arduino and python are connected by the serial and we are able to send data for the serial and move the robot.

4 Implementation

To do the implementation we are using python. Each Arduino are connected with the Serial and we are able to take the data but also to send data. Indeed,to move the robot, when we press "8" the robot will go forward, "2" backward, "4" left, "6" right and we need to press "0" to stop the robot.

To record we just need to press "r" even though we are moving. Then, the robot will stop and start record data on a different text file for each recording. After each record, we may stop the program with "s" or continue moving with "m".

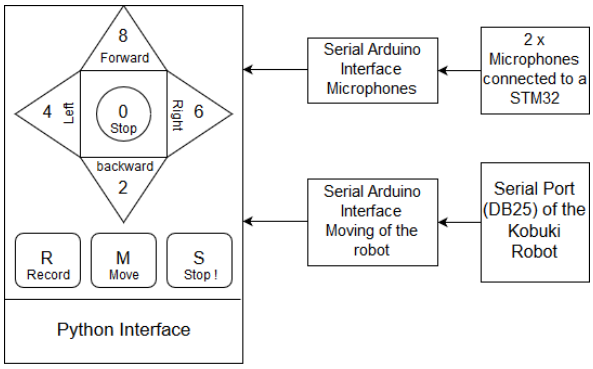

Figure 5: Block diagram show how to move the robot

5 Conclusion

The robot is able to move all around a room controlled by a user. When the user wants to record he just need to press a button and analyse the data recorded. To improve the robot, we might add a Bluetooth module which control the robot remotely and try to have access to the gyroscope of the robot using ROS or Arduino. The board for the microphone should be also changed for the STM32discovery1 to have a better output. Avoid totally the noise with a better approach should be a good evolution.

[1] kobuki website: http://kobuki.yujinrobot.com/kobukino/