Sound Guided Bluetooth Car System

Ionut Catalin Dondera, Ioan Filip Gut, Politehnica University of Timisoara June, 2018

1 Repository

The project history, schematics, diagrams and codebase are contained under the following git repository:

https://github.com/CatalinID/Sound-Guided-Bluetooth-Car

2 User requirements

1. The system must follow a sound provided in a certain area.
2. The system should locate the sound source.
3. The system should run each time you turn on the robot and a loud sound is played.
4. Microphones should be placed somewhere on the robot or nearby and should record where the sound(noise) comes from .
5. The robot has to go in the direction where the sound comes from.

3 System overview

The overview of the system is depicted in Figure 1.

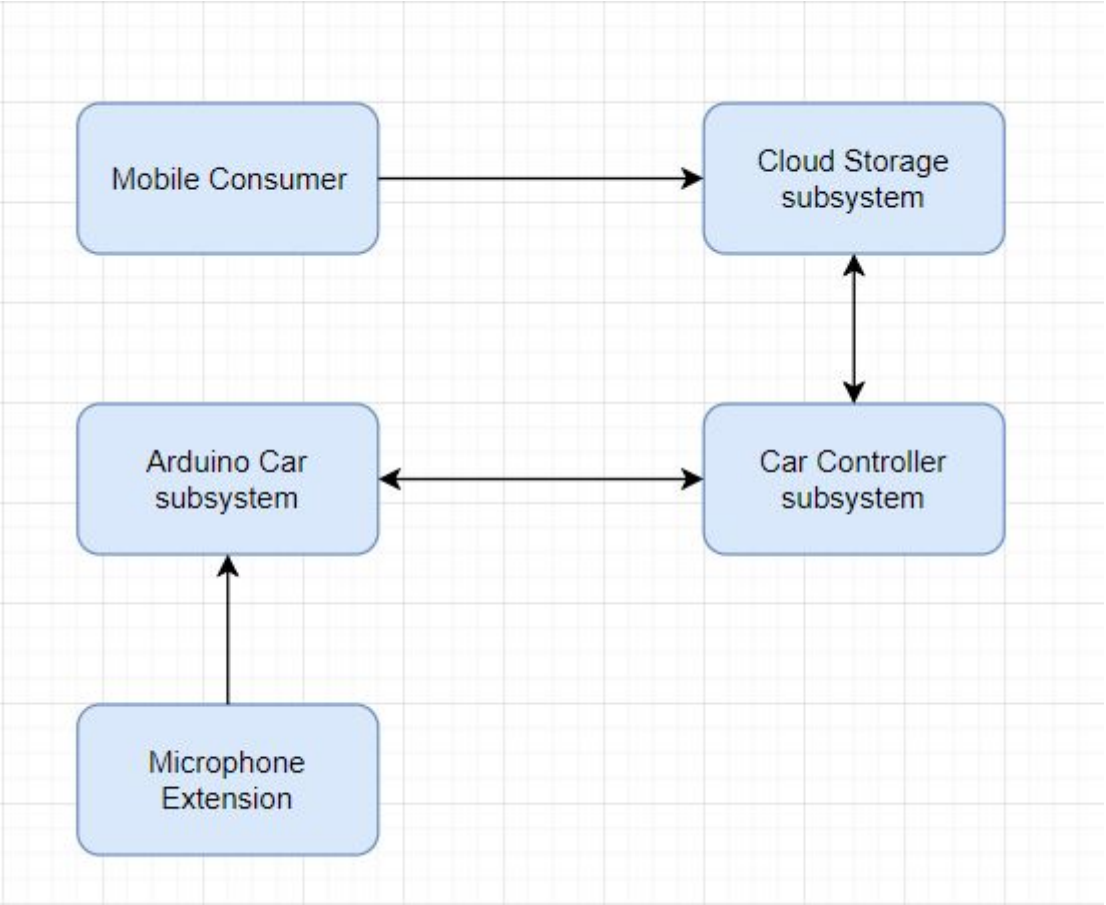

Figure 1: System overview diagram

Base (Car Controller subsystem) Subsystem encompasses the communication between the Arduino Car subsystem and the Mobile consumer. It also has the sole purpose to acquire information from the extension part, Arduino Car subsystem, more exactly from the microphones sensors and base on that information send the direction in which the car has to drive.

Speakers provide sound samples for the car. It must provide a specific type of sound that will be recorded and the car will react properly.

Firebase subsystem extension updates the real time database provided by the Base subsystem. Additionally, it offers the possibility for Mobile consumer to interpret that data and use it.

Mobile consumer also provides a UI interpretation for the stored data. The view will be design as a radar that will access data from cloud and will point where the sound is.

4 Circuit design

The hardware view of the system is depicted in Figure 2.

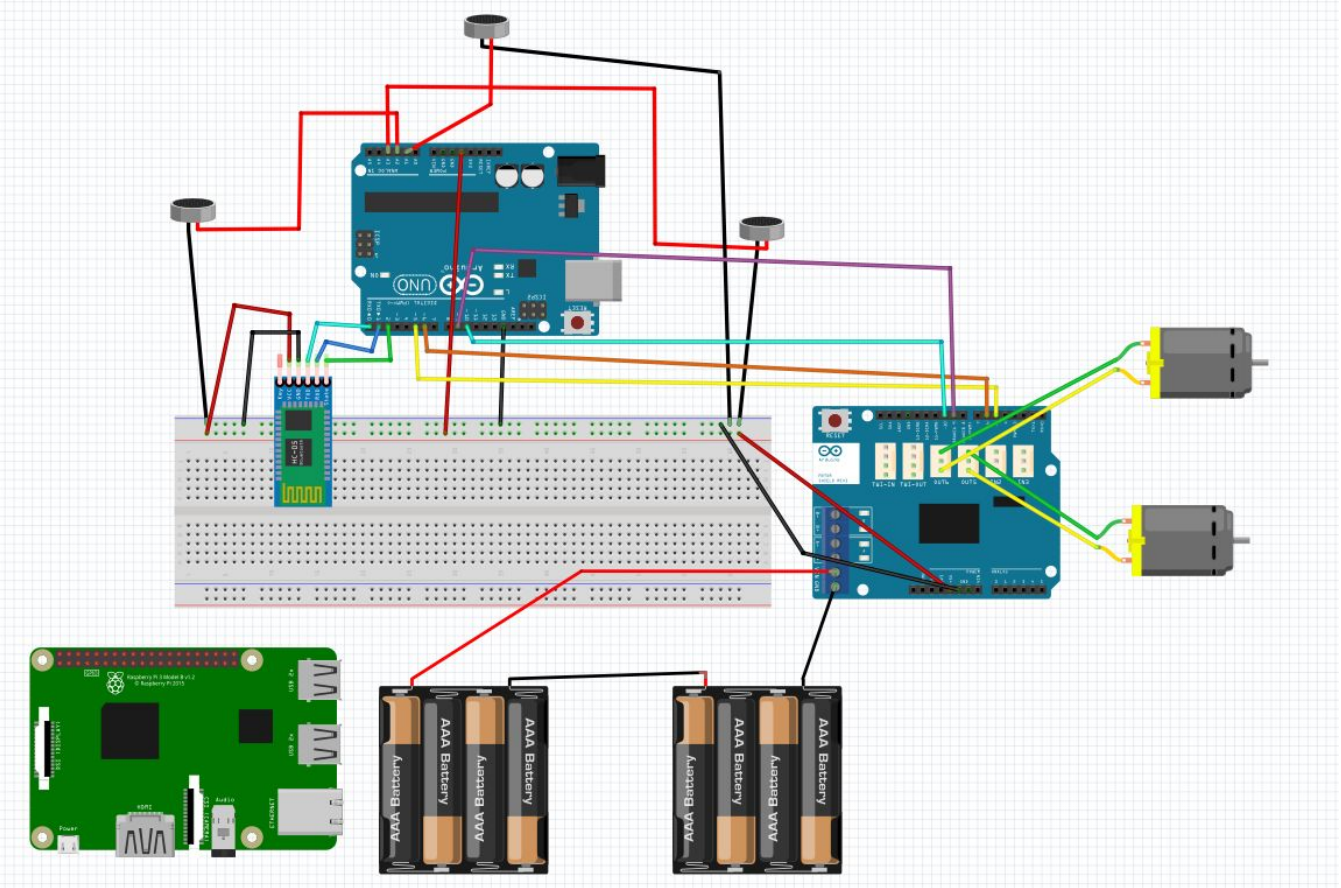

Figure 2: Circuit schematic

Raspberry Pi 3 provides support for quick prototyping. That makes it a perfect choice for quick prototyping and adequate for real-time applications. We will use the Raspberry Pi 3 as a brain for our system. It receive data about the position of the speakers, does the computations and push into Firebase the coordinates of the speaker. It also does the connection with the Arduino Car subsystem that will send and receive data/direction via Bluetooth.

Raspberry Pi 3 has both Wifi and Bluetooth module and based on this we don't need to make any wired connection to communicate with our subsystems.

Arduino Uno provides support for reading the analog values of the microphones (easy to implement) and it does properly the movement towards the target.

5 Software design

The software components and data flow directions are depicted in Figure 3. Each of these will be presented in the following subsections.

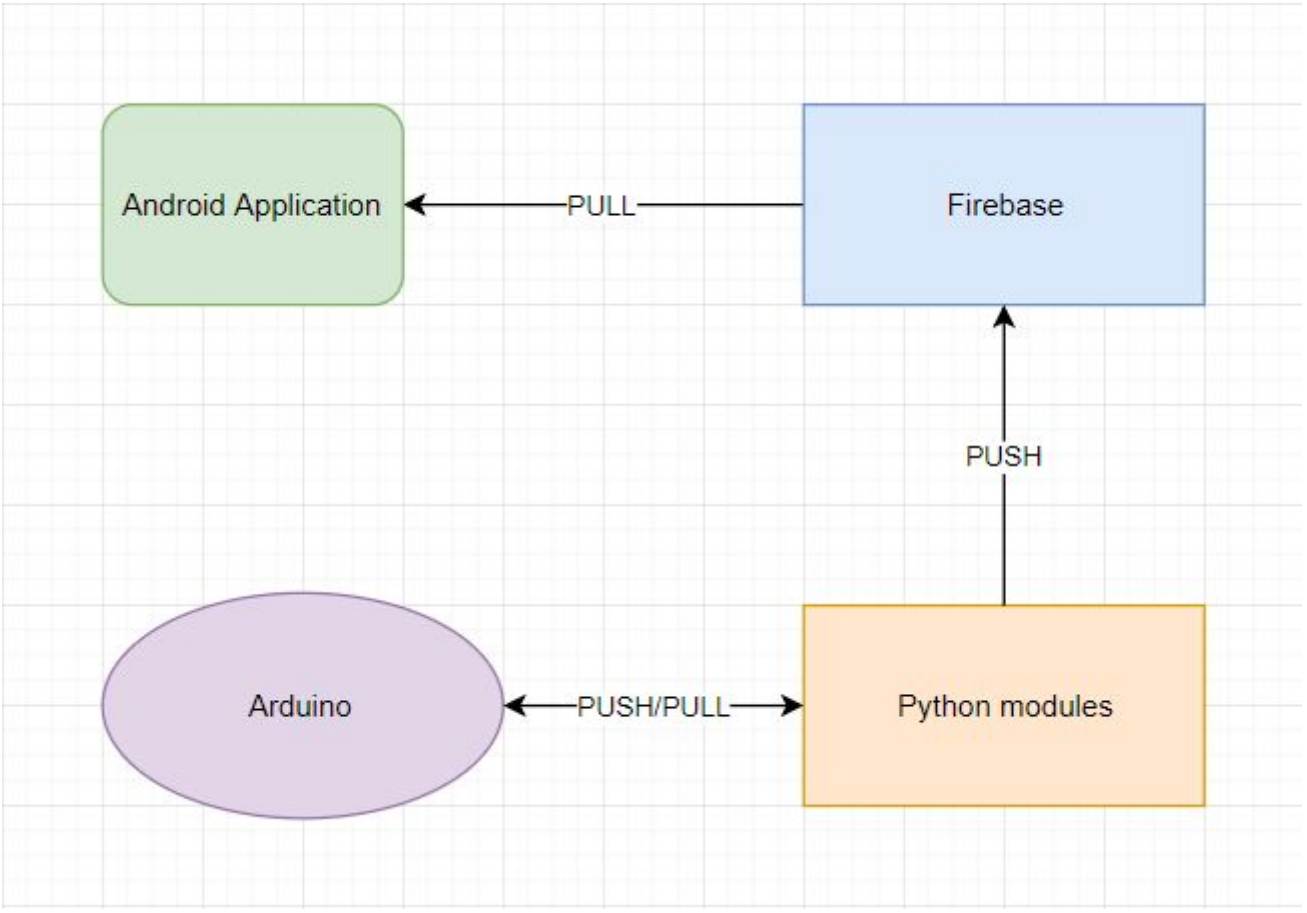

Figure 3: Software entities involved

5.1 Python modules

drive.py: it sends the start state to the Arduino and it retrieves the microphones data from it via Bluetooth, computes the directions of the sound and then pushes it to Firebase and the Arduino every 0.1 seconds.

```
while input_data.find("Done") :
    #This reads the incoming data
    i=i+1
    m[i]=input_data.decode()
    print(m[i])#These are bytes coming in so a decode is needed
    time.sleep(0.1) #A pause between bursts
    input_data=bluetooth.readline()
```

5.2 Android application

Firebase library: to connect the application the database and read the data from it

MainActivity: retrieves the direction of the sound from Firebase and displays the data on the screen related to the position of the Arduino car.

```
public void onDataChange(DataSnapshot dataSnapshot) {
    String text = dataSnapshot.getValue(String.class);
    displayLocation(text);
}
```

5.3 Arduino

Arduino: responds to the states received by Bluetooth from the Raspberry. It does an average from 100 analog reads from each microphone and sends the data to the raspberry to be processed.

6 Results and further work

The current version of the project supports the following functionalities:

- reading data from the microphones
- real-time communication to Firebase Database
- client implementations for retrieving in real-time the position of the speaker from FireBase Database (Android App)
- proper movement towards the target is perform when the sound is louder than a specific value

The following list of extensions and improvements was identified to be supported in the future:

- upgrade the microphones such that they can record a sound and compare that sound with a pattern in order to react to that sound.
- extend Android app functionality to display the approximate distance to the location where the sound is recorded.

7 References

1. Draw IO [last seen: May 2018], https://www.draw.io/
2. Arduino forum [last seen: May 2018], https://forum.arduino.cc/
3. Firebase Database [last seen: May 2018], https://firebase.google.com/docs/database/