Marshall Shriner - shriner3@illinois.edu - ECE 120

Tulika Gupta - tulikag2@illinois.edu - ECE 120

Gauri Konanoor - gmk2@illinois.edu - ECE 120

Introduction

Statement of Purpose:

Oftentimes, at parties or in restaurants, people find themselves shouting at one another rather than having a relaxed conversation, because of loud music. In these situations, it is often very inconvenient to have someone abandon their job or interrupt their party experience to go and adjust the volume of the music. Our goal was to design a prototype of a speaker that measures the volume of the surroundings, and increases or decreases the volume of music in order to maintain a pleasurable atmosphere for any event.

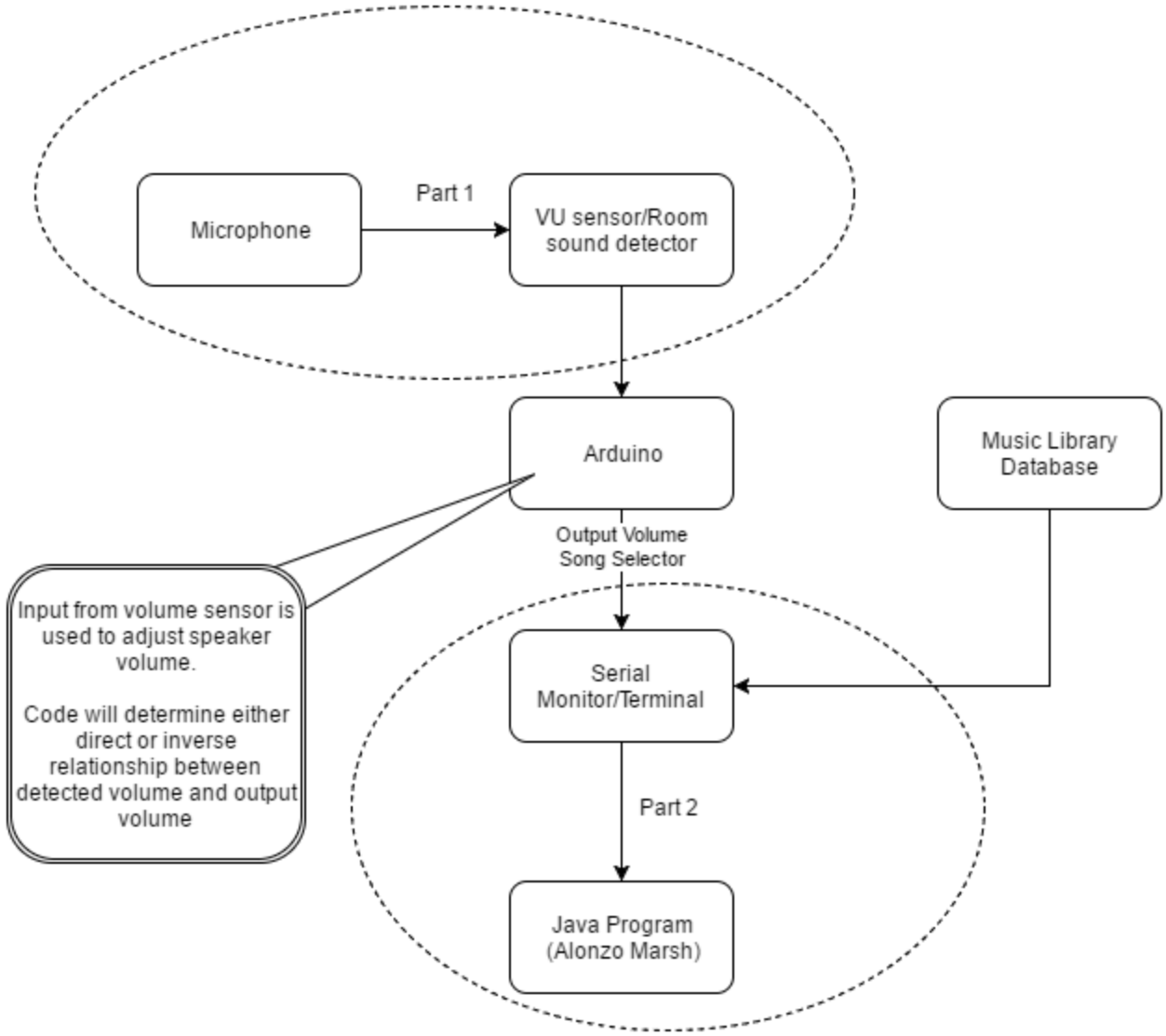
Features and Benefits:

Our project was designed to:

- Measure the loudness of the surrounding room
- Depending on the mode:
    - Increase the volume of the music when the ambient noise quiets down (inverse relationship -- currently unimplemented)
    - Or decrease the volume of the music when the ambient noise goes down (positive proportional relationship)

Design

System Overview:

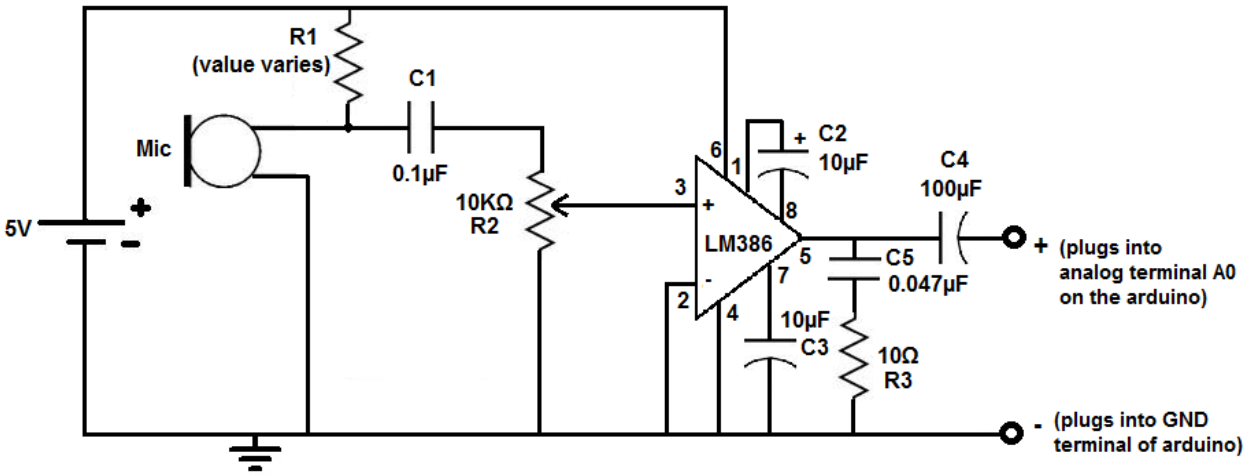


Design Details:

- Microphone
    - We used an Electret Microphone (COM-08635 ROHS) to sense our ambient room volume. This microphone didn't really do very well

what we wanted it to do. It worked for a while but at the end of the semester it suddenly stopped working.

- Sound Detector/Amplification Circuit
    - This circuit was theoretically needed to raise the amplitude of any detections from the microphone to arduino-readable levels.



- Arduino
    - For use in song selection, and interpretation of microphone data.
    - For Arduino Uno code, see Appendix
- Music Library Database
    - Folder containing .wav files of some sample songs for testing purposes.
    - Used by Java program when playing music

- Serial Monitor/Terminal
    - This was how the arduino and java programs communicated. The arduino program, called SoundSelector, printed a song index number and a value for music volume
- Java Program
    - Responsible for adjusting computer's volume, as well as playing music
    - Thanks to Alonzo Marsh for helping us out!

Results

We were not satisfied with how our project turned out. Our project was never fully together at one time, as once we got "part 2" of our project to work, portions of "part 1" were failing. This was, needless to say, very frustrating for us, because this situation provided us with very little time to be able to test and gather quantitative data. One of the weaknesses of our project was we spent too much time debating the conceptual and not enough time building and performing tests on the concrete.

Problems and Challenges

1) Microphones are very temperamental and ours failed the day before the presentation.
2) Due to limited coding experience, we faced many challenges while trying to alter the volume of the source.

3) Since the Arduino could not directly adjust the volume of the source, a separate code had to be written in Java, which again was a bit difficult for us to comprehend due to our limited knowledge in Java.
4) Through a period of time we tried to implement the volume control system with hardware but ended up using a software volume control.
5) Due to the change above, we were running a little short of time.

Future Plans

Based on problems faced while trying to execute the project, we have established that we need to work on:

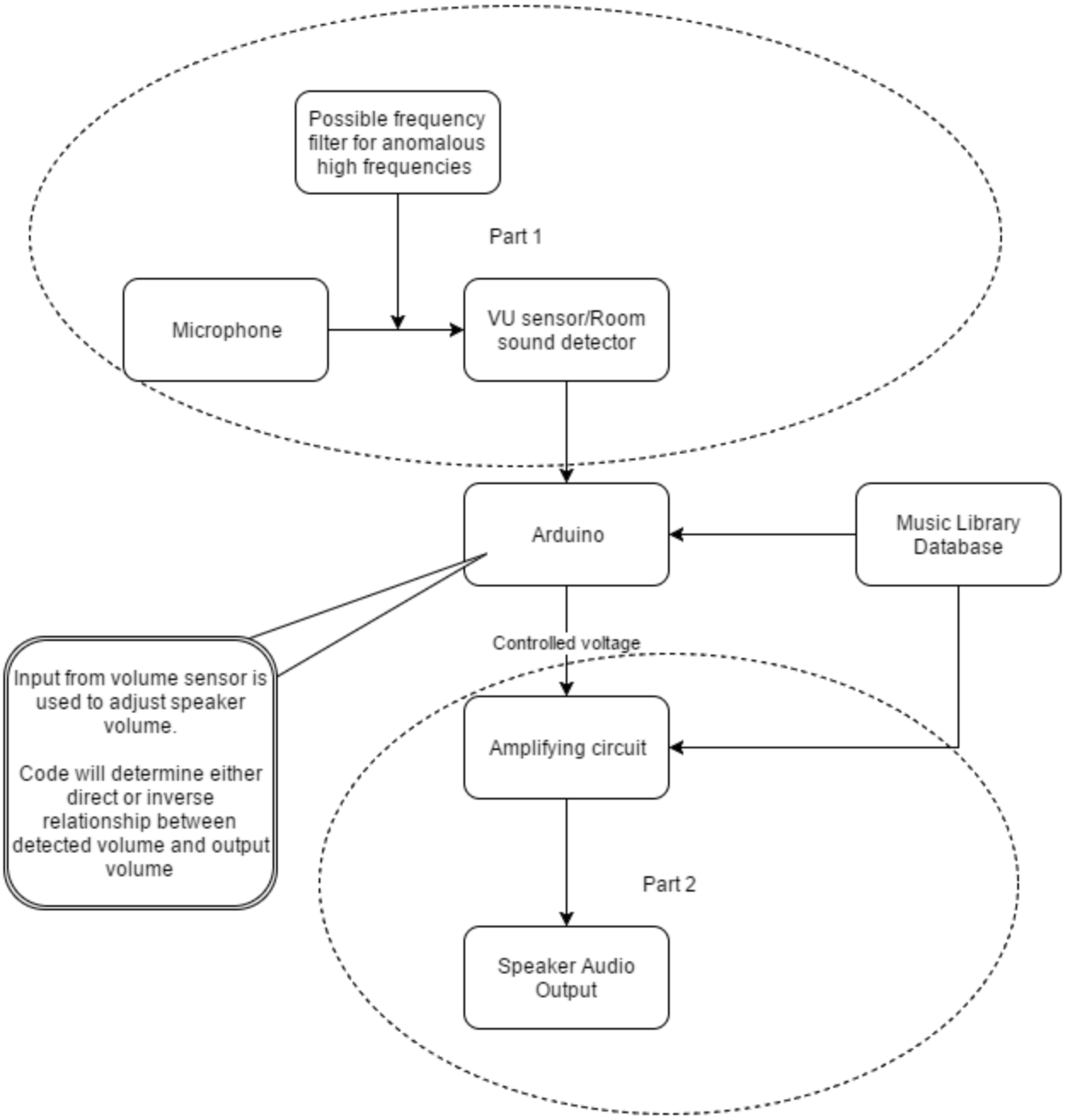
1) Debugging the program with a working microphone
2) Establishing a good distance between the source of sound and the microphone to prevent the microphone from reading the sound as ambient noise.

Once a working project is established, we plan to extend the project to account for an alternate setting in which we decrease the volume with an increase in ambient noise and vice versa. This setting would be useful for places that require a quieter surrounding.

The user would have the option to choose between either mode, thus giving our program more flexibility.

Appendix

Original schematic: Sought to use hardware volume control. Final result uses software volume control.



Microphone Demo Video:

https://drive.google.com/open?id=0B8b6CG1MZ1WQOE5jTGF0ZDhHY0U

Arduino Code:

```
#define MICPIN 0
#define DIFF 0.5f
int count, sound = 0, volumestore = 0;
float volume = 0.5;
/**
 * Method run at the beginning of the code.
 */
void setup() {
  Serial.begin(9600);
  count = 0;
}
/**
 * A loop that is iterated through continuously as part of the Serial.begin()
method.
 */
void loop() {
  count++;
  count = count % 11;
  Serial.print(count);
  Serial.print(" ");
  sound = analogRead(MICPIN);
  if(abs(sound - volumestore) > DIFF){
    if(sound > volumestore && volume <=1.0){
      volumestore = sound;
      volume = volume +0.1;
      Serial.println(volume);
      delay(10000);
      count++;
      count = count % 11;
      sound = analogRead(MICPIN);
      if(sound - volumestore > DIFF){
        volume = volume -0.1;
        Serial.print(count);
        Serial.print(" ");
        Serial.println(volume);
      }
    } else {
      volumestore = sound;
      if(volume >=0.0){
        volume = volume -0.1;
        Serial.println(volume);
      }
    }
  }
  delay(10000);
}
```

Bibliography:

"Amplifier Circuit". *Instructables.com*. N.p., 2016. Web. 11 Oct. 2016.

"Arduino Audio Input". *Instructables.com*. N.p., 2016. Web. 11 Dec. 2016.

"How To Build A Sound Detector Circuit". *Learningaboutelectronics.com*. N.p., 2016. Web. 11 Nov. 2016.

"Make A Simple Audio Amplifier". *Instructables.com*. N.p., 2016. Web. 7 Nov. 2016.

"Sound Detector Hookup Guide - Learn.Sparkfun.Com". *Learn.sparkfun.com*. N.p., 2016. Web. 9 Oct. 2016.