OntarioTech
UNIVERSITY

Faculty of Engineering and Applied Science

# Arduino Audio Player

## Embedded Systems Final Project Report

Instructor: Dr. Akramul Azim

| | |
|---|---|
| Kaushal Patel | 100586212 |
| Darron Singh | 100584624 |
| Jasindan Rasalingam | 100584595 |
| Ashwin Kamalakannan | 100584423 |

# Project Overview

## Objective

Create a music player using an arduino and LM386 signal amplifier, and potentially add an alarm feature with display. Use this opportunity to do research on digital to analog conversion and signal amplification.

## Relation to Embedded Systems

- Micro-controller, low level code written for specific hardware
- System has a dedicated function, that is part of a larger system
    - In this case the speaker and arduino are part of the alarm clock
- Digital to Analog signal conversion
- Designed for a specific purpose, play audio files (.WAV)

# Design Decisions

We chose to base our system around the Arduino microcontroller and the LM386 audio amplification circuit. Our four main modules were as follows:

1. External storage device
2. Button inputs
3. Digital to analog conversion
4. Signal amplification and filtering

For our external storage, we decided on using an SD card reader module compatible with the Arduino. This was simple to implement and it allows our system to be easily integrated into more complex systems, since micro SD storage is common for portable devices.

The button inputs started as simple push buttons that were bound to inputs on the arduino circuit. We would then read the inputs when they were pressed and perform actions in our code accordingly. The problem with this was that there was no input filtering built into the arduino circuit and as a result we had to implement a debounce system. This system stops multiple inputs from registering and allows us to play and pause music a single time rather than register them multiple times.

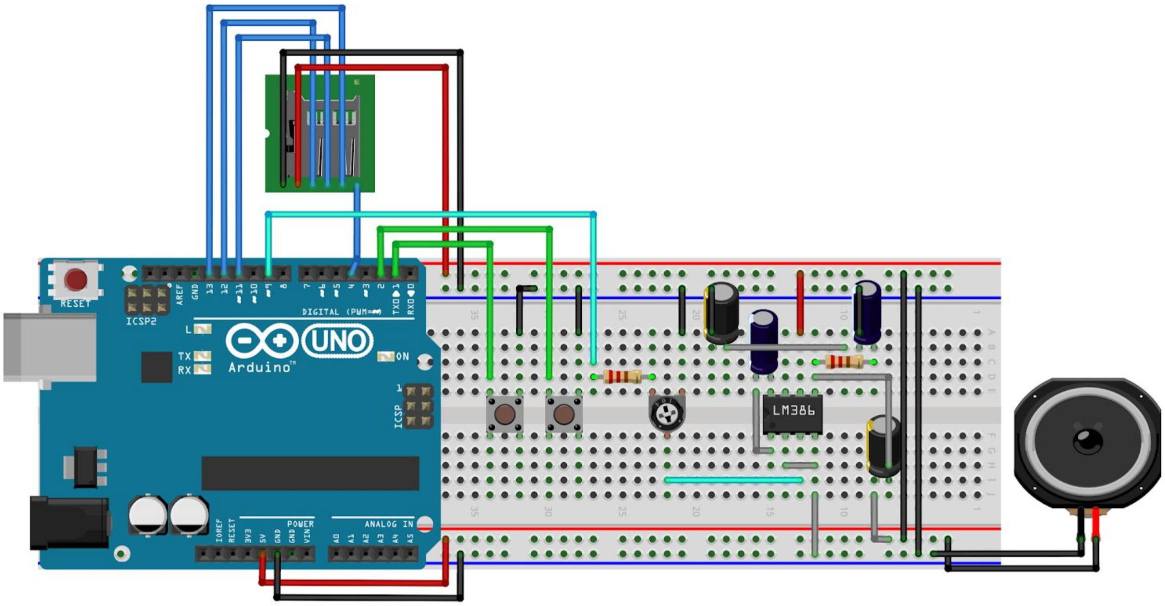
Since the arduino reads data from the SD card digitally, we need to convert this data to an analog signal that can be outputted to a speaker. We researched a few different methods of performing digital to analog conversion, both hardware and software based implementations. We decided to use a software library for the arduino called TMRpcm, which is capable of converting digital .WAV audio files to analog signals asynchronously. We simply include this open source library in our project and use it to play tracks.

The most complex module was the amplifier for the speaker. We initially wanted to use a simple buzzer to play the audio but we decided that it was too simple and wouldn't sound as detailed as a real speaker. In order to get a basic 10Ω speaker to work, we needed to amplify the signal that was generated by the arduino. The LM386 is a low voltage, audio amp IC that is commonly used in consumer electronics such as radios and multimedia players. We use the IC to amplify the signal to a gain of 200. This is accomplished by connecting a 10uf capacitor between pins 1 and 8 of the ic. Without this capacitor, we would only have a gain of 20, resulting in a much lower volume output. We use several other capacitors to perform filtering. A 470uf cap is placed between the + and - input signal to filter radio interference. We use a 10uf and 100uf capacitor on the power rail to decouple it, suppressing any high frequency noise created by power surges. We also had the option to decouple the input but we determined it was unnecessary since our input signal is coming from the arduino and not another analog source.

## Components

| Name | Qty | Description |
|---|---|---|
| Arduino UNO Rev3 | 1 | Main controller for the system. Will run all of the software. |
| SD Card Reader module | 1 | Used to read data from the SD card and transfer to the Arduino main board. |
| SD card | 1 | Used to store the music files to be read by the system. |
| LM386 Audio Amplifier | 1 | Used to convert the digital audio data from the arduino to analog |
| 10uf Capacitor, 100uf Capacitor | 2 | Used in building the circuit |
| 1K,10K Resistor | 1 | Used in building the circuit |
| Push buttons | 2 | Play pause button, next song button. |
| Breadboard | 1 | Used for prototyping the system without soldering. |
| Connecting Wires | 1 | Connecting the various components together. |
| Speaker | 1 | Play audio output from arduino |

# Implementation (Schematic)



The above schematic was created in fritzing, an open source circuit design software. Some of the parts in the diagram are either missing or replaced with common parts since the software did not contain all the necessary components. Building the circuit on this software was very helpful because it allowed us to optimize the wiring and component placement as much as possible before we started purchasing parts. This also helps our end product since audio circuits should be shorter and can operate much cleaner when there is less resistance between the parts.

# Implementation (Code)

All source code and schematics are available here:
https://github.com/AshwinK97/Arduino-Music-Player

For our code, we chose to use the inbuilt arduino programming language. This language is very easy to build quick prototypes with and has a decent amount of support and external libraries. Although we may have benefitted from using a lower level language like C, the end result was very similar in terms of performance since the project put much stress on the Arduino hardware.

# Implementation Challenges

- Faulty ICs
    - All modules we purchased and implemented on our bread-board worked except the LM386 Audio Amplifier.
    - Not getting the correct sounds from analog audio signals. therefore, figured that the LM386 Audio Amplifier wasn't working.
    - Possible issue with filtering out noise
- Difficult working with analog signals
    - Testing can only be performed manually, very little debugging options
    - Audio circuits should be as short as possible to minimize resistance during signal amplification
- Prototyping with a breadboard and components can be difficult without appropriate tools
    - Hardware and budget limitations caused issues during construction.

## Lessons Learned

We learned quite a few skills related to developing embedded systems, namely how to prototype using components and a microcontroller, issues you may face with a physical medium and debugging.

To create the logic for the audio player, we had to use the arduino specific language. Learning this was crucial to being able to further develop prototypes in the future. Additionally, it gave us an idea of how to implement I/O and libraries. We also learned how to build a complete circuit using the components we had and experience from the lab.

Lastly, we ran into some issues with our LM386 Amplifier Module. We are not sure of the exact reason behind its failure, but it is highly likely a dead IC. A lesson we learned from this is to test components early on into the build to ensure liveness. This will give you more time to replace any components that are dead.

## Final Remarks

Overall, our group was able to complete the majority of the features that we originally wanted to do in comparison to our proposal. By applying the various concepts that we learned throughout the course and doing extensive research, we were able to create a solution that met the requirements as well as one that we were satisfied with. We appreciated all the support we received throughout this course.