## Aims and Objectives

### Aims

According to the proposal given by the project tutors (Appendix A), the main goal is to develop 4 new robot instruments that will be controlled by a conductor. This should form a new standalone robot orchestra that can play at least 2 pieces of music and should be flexible and allow for new additions of songs or instruments in the future. As the title implies, this is a robot orchestra and it should operate at a high level of autonomy where the only interaction should be the user of the orchestra picking the songs using the GUI designed to interact with the conductor. The MIDI format, which is a standard format for electronic music files, will be used as a common format for all the songs played in this project.

The orchestra should stick to a certain engineering aesthetic as it is meant to promote STEM aspects and to appeal to audiences that are not traditionally associated with engineering. Other goals outlined by the proposal are to make sure the orchestra is easily transportable in order to take it to different places and events.

### Objectives

The objectives of the project are as follows:

1. Selecting four new instruments:

* Looking at different songs that could incorporate 4 robot instruments then breaking down the notes needed for each instrument.
* Picking two songs as well as the 4 instruments that will be playing those songs.

1. Designing and constructing the four instruments:

* Proposing designs for the instruments as well as ideas for their communication with the conductor while considering transportation and modularity.
* Using CAD to design hardware and then producing prototypes.
* Designing and testing breadboard circuits and converting to PCBs.
* Printing the PCBs and making sure they work with the software like the breadboard circuits.
* Constructing the hardware.
* Testing the instruments and making sure they play all the notes at the right tempo.

1. Designing a conductor:

* Deciding on an embedded system for the conductor.
* Designing the conductor which will communicate with the instruments.
* Communicating with each instrument using the conductor.
* Designing the GUI.
* Receiving “Play” and “Stop” from the GUI and sending those commands to the instruments.

1. Synchronising the 4 instruments to play coherent music.

### Potential instruments

The panpipes were one of the instruments considered. They consist of 15 pipes would be played by blowing air across the openings of the pipes in order to generate notes. According to research done in [32], the pipes would require about 8.5 litres per minute [32] of airflow to play notes the octaves above C6 (if C1 is the first C).

Research was done on other wind instruments such as the Waseda Flutist Robot [33] and the Saxophone Playing Robot [34] seen in Figures 2.9 and 2.10 respectively, but due to the complexity of some components such as the artificial lips and lungs this was considered an unfeasible option due to the time constraints of the project as well as the budget.

|  |  |
| --- | --- |
| Figure 2.9 Waseda Flutist Robot [33]. | Figure 2.10 Saxophone Playing Robot [34]. |

Proof of concept can be seen in video [35] where a set of panpipes is being played by a robot that moves the panpipes across a nozzle that blows air across the pipes producing the corresponding notes. The panpipes are easier to automate since there are less moving parts and Degrees of Freedom (D.o.F) involved compared to the other wind instruments that were feasible.



Figure 2.11 Robot plays the panpipes by moving the panpipes across an air nozzle [35]

The guitar was another instrument that briefly researched as a potential instrument. There are few research papers on 6-stringed guitars therefore videos were the primary source of analysis. A paper on the history of robot instruments [36] shows a guitar with “72 finger left hand” [36], and a video of the robot band *Compressorhead* preforming the *Ace of Spades* [37].

|  |  |
| --- | --- |
| Figure 2.12 Sergi Jorda’s Afasia from [36] | Figure 2.13 Compressorhead 6-String guitar robot preforming Ace of Spades [37] |

Both robots have a relatively simple plucking system but a complex fret mechanism consisting of multiple solenoids or other moving parts which could be well take up a large portion of the budget and time which needs to be given to other instruments. This instrument was later discarded due to these reasons.

### Piezo Sensors

Piezo sensors were put on top of the xylophone keys to detect the solenoids hitting the keys. They were connected to a comparator circuit that compared the voltage from the piezo sensor and the reference voltage of the voltage divider at the non-inverting input. The output of the comparator circuit was connected to the MyRIO so whenever the output was high, the LabView program on the MyRIO played the corresponding note through a speaker. The MyRIO was the preferred microcontroller since it already had a built-in audio output.

|  |  |
| --- | --- |
| Figure 3.18 Piezo sensors on the keys connected to the comparator | Figure 3.19 LM311 Comparator PCB |

### Xylophone Transistor Circuit

A similar transistor circuit is used in the xylophone as was used in the keyboard. The VIN= 3.3 V, VBE = 1.6 V, gain = 4000 when IC = 2A. Therefore, the base resistor needed is, R = (3.3-1.6)/(2/4000) = 3.4 kΩ. Using the E24 resistor series a 3.3kΩ resistor was selected.

### Comparator circuit

After making the initial comparator circuit using a LM741 [42] and a voltage divider, several values of resistors were tested and it was found that a sensitive reference voltage was found to be around 0.2V that would detect the impact of the solenoid upon the piezo sensor. An LED was also placed at the output as an indicator. This circuit is seen in Figure 3.20.

R1 = 8000 Ohms and R2 = 330 Ohms.

Where V is the supply voltage (5 V) there for .

The circuit worked very similar to the final PCB and circuit design shown in Figure 3.19 and Figure 3.21, the only important difference being that the LM741 is powered by -15 and +5 volts whereas the LM311 can be powered by Ground and +5 volts. There were 12 of these circuit connected to each other and to the 12 xylophone keys and in turn connected to 12 individual inputs on the MyRIO.

After tests were done, there were problems with the piezo sensors when they were used for around 20 seconds, the LEDs indicated that the comparators stopped responding to the solenoid hits. This was fixed by adding a resistor in parallel with the piezo sensor to get rid of any build-up of static charge. Later it was noticed that there was -15 V at the output of the comparator when it was in the off state, which could damage the MyRIO, therefore the circuit was slightly modified, and a new comparator was added: LM311. This comparator could function of off ground as negative voltage supply so the danger of damaging the MyRIO was no longer there. The new circuit can be seen in Figure 3.21.

This leads to the LabView code that was used on the MyRIO to play the different sounds for each individual key.

|  |  |
| --- | --- |
| Figure 3.20 LM741 comparator circuit | Figure 3.21 LM311 comparator circuit |

### Xylophone Software for Piezo Sensors

The code to play the different notes was made in LabView. The MyRIO cannot directly output a wav. or mp4 format files to the audio output therefore the wav. files of the xylophone notes, which were obtained from the University of Iowa Electronic Music Studios [46], were converted to a waveform and stored as the one-dimensional array, Y seen in Figure 3.23. This was all done offline since the MyRIO cannot do the converting in real-time [47].

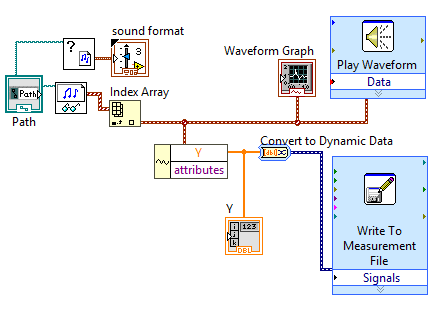


Figure 3.23 Converting wav. files to waveforms with help from [47].

This array was then converted to a constant and moved to the main program seen in Figure 3.24.

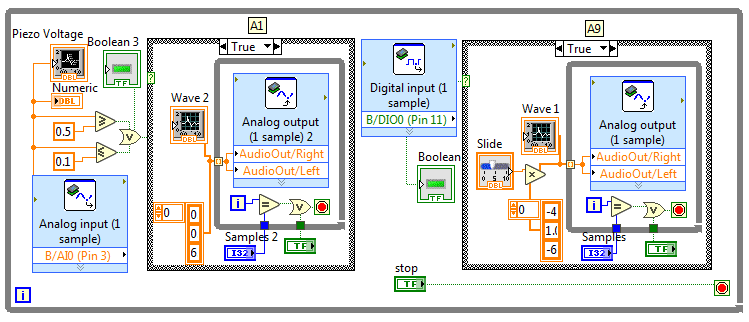


Figure 3.24 LabView code for analogue and digital piezo inputs

This code looked at the input from the pin to which the comparator was connected to and whenever it was high, it played the waveform from the 1D array through the audio output. The waveform was played at 44,400 samples a second, which was the speed that was specified when converting the sound files to an array, the number of samples played could be set to any number and the volume was changed by multiplying the amplitude of the waveform. The front panel (interface) for testing can be seen in Appendix I**.**

This code was then expanded to 12 inputs and 12 waveforms for all the xylophone keys. The real-time module of the MyRIO was then used to store the code onboard and execute it every time the MyRIO was started up (Appendix J).

### Speaker

A speaker which would play the xylophone notes was installed and an amplifier circuit connected it to the MyRIO output. This circuit allowed the user to tune the volume and gain of the speaker since it did not come with any inbuilt controls.

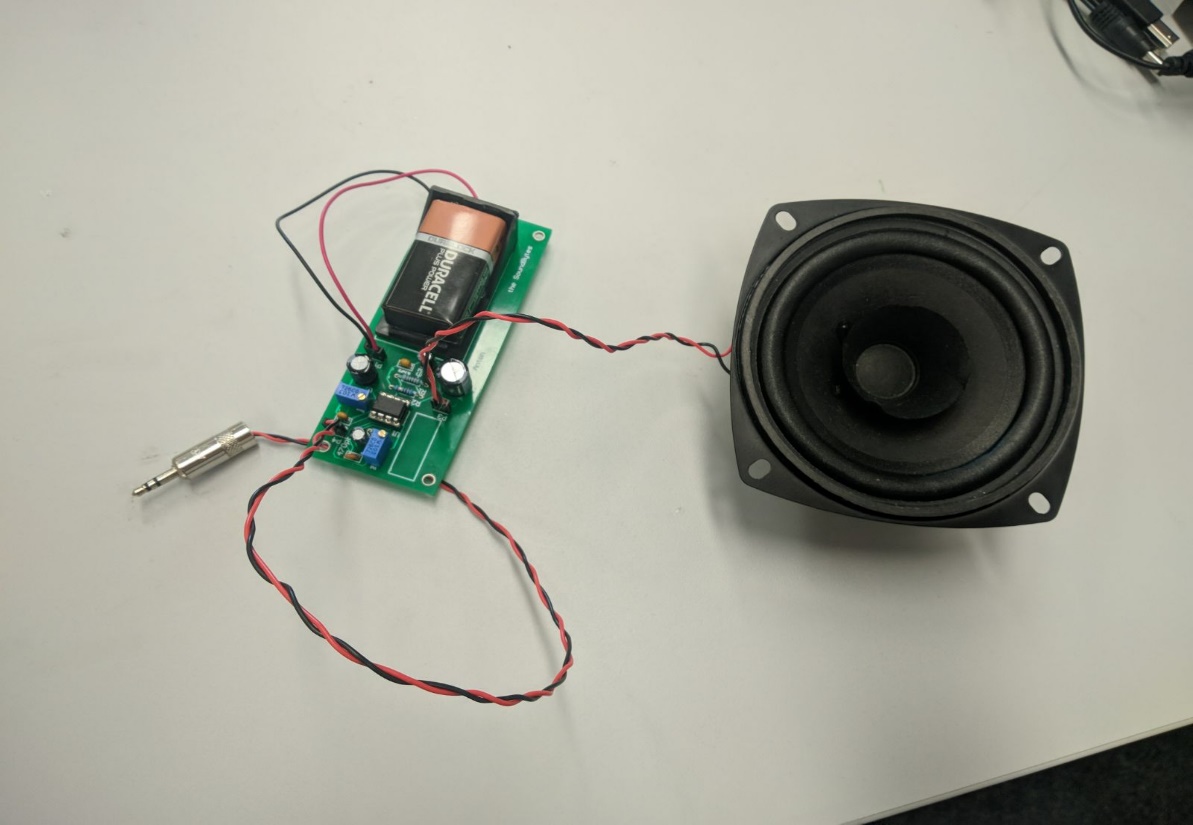


Figure 3.25 Visaton Speaker [48] connected to the amplifier circuit.

The datasheet [48] stated that the speaker was capable of outputting 86 dB between 80 Hz and 20 kHz frequencies, which was loud enough to play with the rest of the orchestra. To get to the higher sound levels, the signal from the MyRIO needed to be amplified.

An amplifier circuit was built following the instructions from [49]. This circuit allowed the user to control both volume and gain of the signal using potentiometers. This circuit can be seen below:

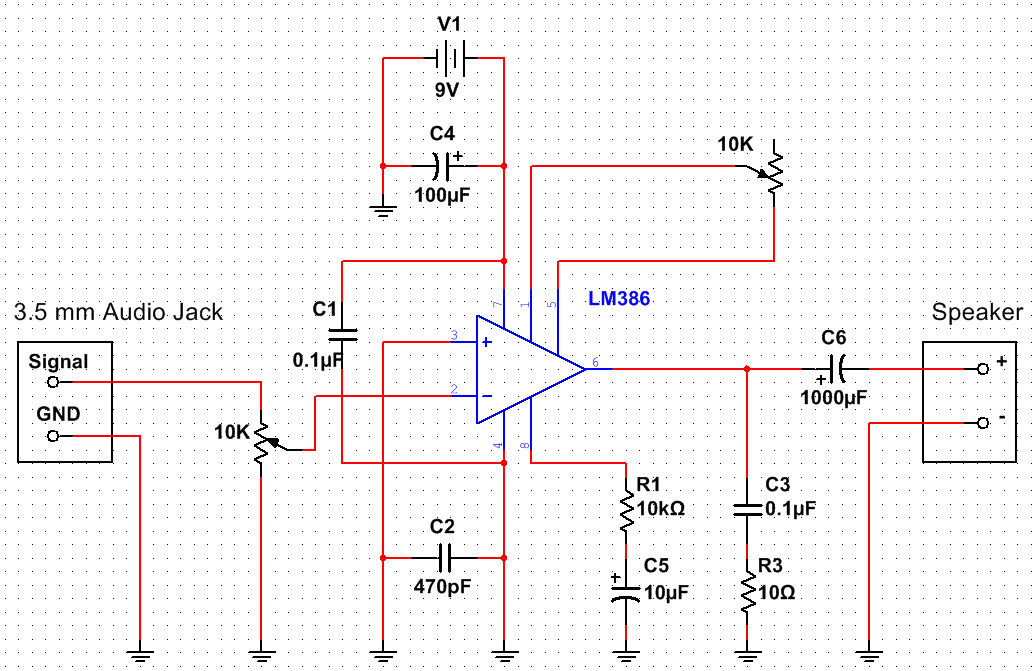


Figure 3.26 **.** Amplifier circuit with volume and gain controls based on [49].

This circuit used the LM386 chip for amplification. A 9 V battery was used as the power supply, but a MyRIO +15 V and ground could also be used as the LM386N-4 [50] chip can support up to 22 V as a power supply. The designed PCB can be found in Appendix K.

This circuit was later used to amplify the microphone input for the Tesla coil setup, but it was later found out that the circuit could pick up the signal from the Tesla coil wirelessly, without a microphone.

### Xylophone Testing

In order to determine the functionality of the xylophone, several tests were performed, concerning both the hardware and software. More specifically, each solenoid was tested individually by writing a program where the user is able to choose which solenoid to be played. The purpose of this test is mainly to check the response of each solenoid (i.e. whether they turn on or off) and the ability of the Bosch bar (which hold the solenoids above the keys) to preserve its initial condition and not be affected by the impact of the solenoids.

To test the software of the xylophone, each song played by the instrument, was timed and compared with the MIDI file version. By doing so, the accuracy of the software was tested and since the two timings match (of the MIDI file and of the instrument), then it means that the xylophone behaves as expected.

For the piezo sensor testing, the first step was to test each comparator individually by connecting them to a MyRIO and running the code for only one note using code shown in Figure 3.24. After making sure all the circuits work and produce a note, the circuits were all mounted and all the twelve keys with the piezo sensors were attached to the xylophone. The testing for twelve notes showed that each key produced different note, going from low to high as the keys were hit from one end of the xylophone to the other.

### Summary

The xylophone has twelve solenoids that are controlled by the MyRIO. The MyRIO receives the commands through the wireless module which dictate what song to play. Once the commands have been received, the solenoids start hitting the keys. The keys have piezo sensors on them that are connected to another MyRIO using comparator circuits. The second MyRIO interprets the high input and plays the note corresponding to the key that is being hit. The sound is played through the speaker that is connected to the audio output of the MyRIO.

### Panpipes

There were three main designs for the panpipes throughout the project considered during the feasibility period.

|  |  |  |
| --- | --- | --- |
| Figure 3.31 First design | Figure 3.32 Second design | Figure 3.33 Third design |

**First design**

The initial design seen in Figure 3.31 had the panpipes fixed to a base and the nozzle that directed the air from the pump (mattress pump at the time). The nozzle was on an elevated platform that would move from side to side to play different notes and had a stopper that would cut off the airflow when the platform would be moving from pipe to pipe. This platform would be powered by a stepper motor allowing it to move to individual pipes.

This design was later changed to the second design shown in Figure 3.32 which would simplify the moving mechanism.

**Second design**

Instead of a stepper powered track moving sideways, this design involved mounting the nozzle onto a servo motor which could rotate 360 degrees. The pipes would also be taken apart and mounted in a circle around the servo with the nozzle.

This design required very precise measurements regarding the angle of the nozzle (to make sure it was blowing air at the right angle over the panpipe) and the requirements of the stepper motor, since it needed to be able to rotate very quickly and accurately. Although this was possible, it was difficult to design this in such a way that it would be easy to transport as it did not easily dissemble and reassemble.

**Third design**

The final design seen in Figure 3.33 would have an air compressor providing the airflow since after tests were done it was established that the mattress pump did not provide enough pressure to produce a loud enough sound. There were to be 15 pipes coming from the compressor, providing airflow to each of the 15 panpipes. At the end of each airflow pipe there would be a solenoid valve that is normally closed, and when a specific note needs to be played, the valve would be turned on to let the air through. Plumbing fittings would be used to distribute the pressure and airflow between all the panpipes.

# 