Design Document

MIDI Music Box

Jeremy Lee [jeremyl6]

Sean Liang [shliang2]

Tyler Shu [tmshu2]

Introduction

Problem and Solution

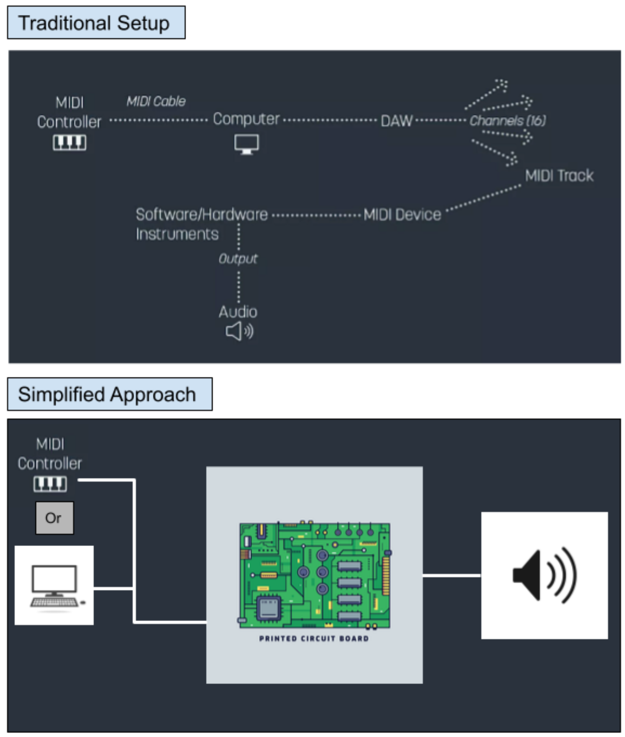
Making music using MIDI Devices commonly found in modern day music devices can tend to be inaccessible to beginners - with a large amount of equipment (and subsequently research) required to listen to music created on MIDI Devices, which typically do not include any forms of playback options, access to music making becomes difficult. Creating a Cost AND User-Friendly attachment specifically geared towards playback would eliminate the need to purchase expensive software and hardware needed to process and translate MIDI signals into usable data, enabling users to more easily break into music making at any level.

We want to implement a form of music box (playback device) that would be able to interface with standard USB ports to either play a recording provided by the user via a file or play signals live from a MIDI device. The project will achieve such using four subsystems, connected together to receive, process, and output relevant data:

1. **The Signal Processing Processing and Conversion Subsystem**
   This subsystem will primarily be responsible for taking in MIDI data via USB input directly into the microcontroller/processor, which will be a Raspberry Pi 3. The data will be processed and translated into a digital signal to be passed into a Digital-to-Analog converter, which will convert the Digital Signal into an Analog Signal for usage within the Output Control Subsystem.
2. **The Power Output and Conversion Subsystem**
   This subsystem will be primarily responsible for powering the rest of the board. The subsystem will employ conversion techniques to provide multiple levels of voltage in order to power various modules throughout the project.
3. **The Output Control Subsystem**
   This subsystem will be primarily responsible for providing variability to the output for user control. The subsystem will provide a method for the user to determine what type of signal will be output, and provide a method for the user to adjust the intensity of the noise produced by the project. Tactile inputs (like switches and dials) will be provided to achieve such.
4. **The Output Subsystem**
   This subsystem will be primarily responsible for amplifying the sound signals into perceivable noise. The subsystem will consist of an amplifier circuit connected to the Output Control Subsystem, which will determine the amount of amplification done to the waveform. The subsystem will also produce the waveform via a speaker.

Visual Aid

The following visual aid provides an idea of what the device will look like when utilized. The image contrasts the traditional method of setting up a MIDI system with our simplified version. In the image, the device is represented by the “Printed Circuit Board”:



High-Level Requirements List

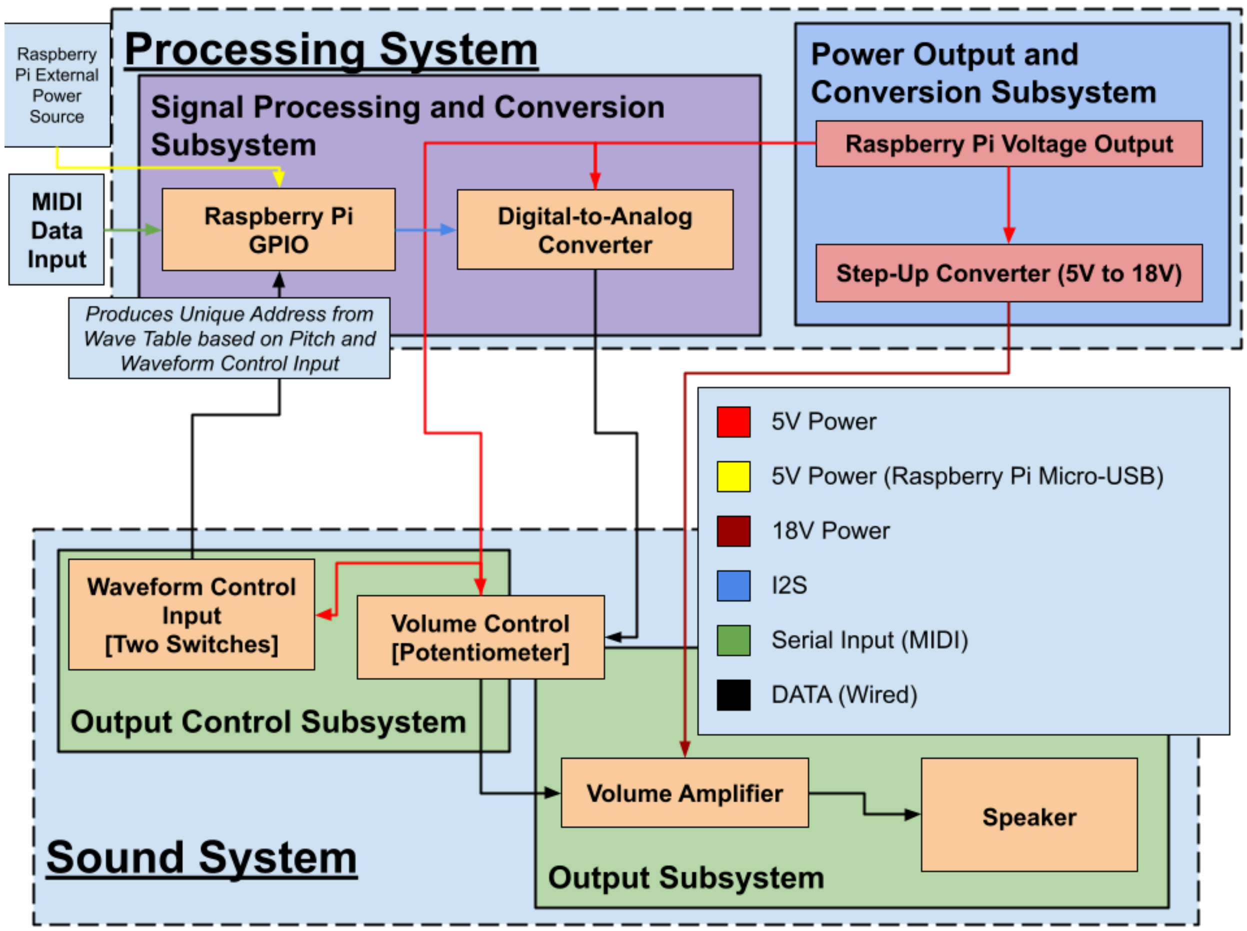
The project must be able to exhibit the following characteristics in order to solve the problem. These requirements are included to provide realistic expectations and goals for the project. The requirements are listed below:

1. The project must be able to synthesize at least **four different tones** (waveforms), including (but not restricted to): **Triangle, Square, Sine,** and **Sawtooth**.
2. The project must be able to produce at least **eight note polyphony** - that is, the project must have the ability to combine at least eight different musical parts together to form a harmony.
3. The project must be able to produce **pitch** in the frequency range **C2 (65.4 Hz) - C5 (523.5 Hz)**, with additional capability of reaching up to **15,000 Hz** including harmonics for synthesized tones.
4. The project must be able to drive a speaker up to **20W**.

Design

Block Diagram

The following is the Block Diagram for the project described. The board will consist of two main systems and four subsystems. A legend is included in the Block Diagram dictating the types of connections expected within the project.



Power Output and Conversion Subsystem

The **Power Output and Conversion Subsystem** will be primarily responsible for generating the power necessary to drive the operations of the device. Components of the subsystem include the Raspberry Pi 3 and a Step-Up Converter (such as the QA053C DC DC CONVERTER). The subsystem will utilize the voltage pins on the Raspberry Pi to provide voltage that will either be directly supplied to other components, or stepped up to a higher voltage via a Step-Up Converter like the QA053C, and then directed to the appropriate components. This subsystem connects to all other subsystems in the design - lower voltage components include the DAC and Waveform Control Switches, and higher voltage components include the Volume Amplifier and Speaker.

The **Power Output and Conversion Subsystem** will be the primary power provider of the overall project. The subsystem powers all other subsystems in the project, making it essential to the operations of the project. The subsystem will utilize the Raspberry Pi's onboard GPIO 5V Pin to power its various components - the 5V pin will power the DAC within the SPCS, Waveform Control Input within the OCS, and be passed to the previously mentioned Boost Converter, which will convert the voltage from 5V to 15V in order to power the Volume Amplifier and Speaker contained within the OS. The essential requirements and verification of the subsystem are included below:

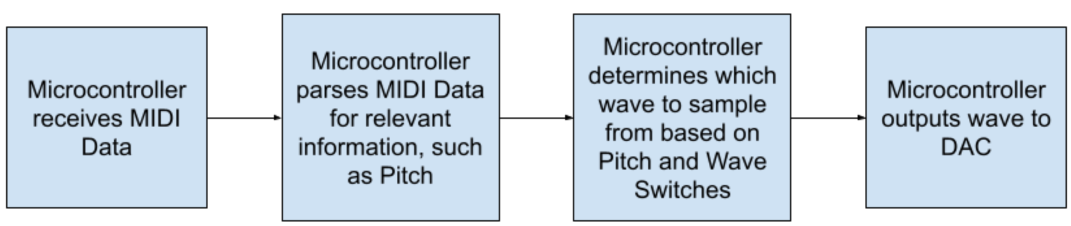
Requirements and Verification

Requirements	**Verification**
- 5V Power Supply should vary from 4.85V - 5.15V - 15V Power Supply should vary from 14.85V - 15.15V	- Voltage read from a multimeter should maintain the given range for 30 seconds

Signal Processing and Conversion Subsystem

The **Signal Processing and Conversion Subsystem** will be the primary processing unit used to interpret our input data, converting it from a Digital Waveform to an Analog Waveform. The input MIDI data will indicate to the processor the status of a given note (whether the note should be on or off), as well as 7-bit values for pitch and velocity. Using the EEPROM on the Raspberry Pi 3, 1024 samples of each waveform will be preloaded into the Raspberry Pi's memory, with 16-24 bit resolution, which will require up to 12.288 kB (24 Bits / Sample * 1024 Samples / Wave * 4 Waves = 98,304 Bits) of storage. The processor will employ a Ping-Pong buffer, which will simultaneously read and process MIDI input via the Serial Port and output I2S data to the DAC. For polyphony, up to 8 sample values would be obtained and then linearly added, sent as a digital signal into the DAC.

The sampling rate of the DAC was determined to be 32Khz, with the need for a 6.155MHz Clock to be used as the System Clock and a 1.536MHz to be used as the Bit Clock. These clock values were determined by analyzing the corresponding Datasheet for the PCM1753. The DAC will also need to contain 16-24 bit resolution, matching the resolution output by the Raspberry Pi, as well as requiring the output to be in I2S format. The SPCS will receive 5V power from the POCS, as well as power the Raspberry Pi via an external source which will supply the Raspberry Pi's Micro-USB port. The DAC will need to be able to output waveforms with frequencies up to 15KHz to the Volume Control of the OCS via wired data, as dictated by the High-Level Requirements. The essential requirements of the subsystem are included below:



Requirements	Verification
- The Raspberry Pi must be able to read Serial input from its serial ports utilizing the MIDI protocol, at the rate determined by the protocol (31250 bits per second) - The DAC must contain a resolution of a minimum of 24-bits - The DAC must be able to output waveforms with frequencies within	- Verify Serial reading by passing in test input with predefined waveform, and verifying based on output audio - Utilize all bits of DAC Components capable of 24-bits. Evaluate based on waveform clarity with Oscilloscope - Verify DAC Frequency Range and

the target range, up to 15KHz - The DAC must be able to produce 4 different waveforms (Sine, Square, Triangle, Sawtooth) - Signal to Noise Ratio (SNR) should be limited to 60 or lower	waveform shape using Oscilloscope and test input - Transfer data to software (MATLAB) to calculate SNR and Tone

Output Control Subsystem

The **Output Control Subsystem** will be the primary interface for the user to interact with the project's user, enabling the user to determine the type of waveform synthesized, as well as giving the user control over the level of volume output by the speaker. Analog signals will be received from the DAC within the SPCS and connected to the Volume Control, where the signal will be fed through a potentiometer and output to the Volume Amplifier within the Output Subsystem via wired connection. Additionally, the Waveform Control Input will output data via wire to the GPIO Pins of the Raspberry Pi with both a high and a low voltage, influencing the type of waveform output. The subsystem also takes 5V power input from the POCS. With the need to produce only four different tones, only a total of two switches is needed. The truth table representing the four waves is shown below:

Switch 1	Switch 2	Wave Produced
0	0	Square
0	1	Sine
1	0	Triangle
1	1	Sawtooth

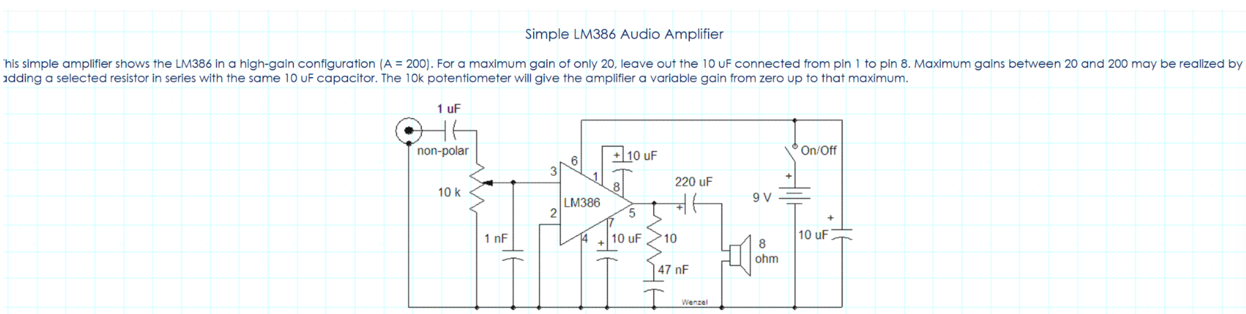
Thus only two switches are needed to represent our four states. The essential requirements of the subsystem and their verification methods are included below:

Requirements and Verification

Requirements	Verification
- Changing the switches changes the output sound and output waveform - Intensity of sound should be adjustable using a dial provided by the subsystem	- Measure waveform for shape and intensity using an Oscilloscope to verify both requirements

Output Subsystem

The **Output Subsystem** will be the primary sound production unit within the project. The subsystem will contain the Volume Amplifier and a Speaker. The Volume Amplifier must be able to take input via wired data from the Volume Control located in the OCS, and amplify the waveform. Additionally, the volume amplifier must be capable of outputting the inputted waveform with a gain ranging between 20 and 200, as dictated by the amplifier's design. The speaker must be able to output frequencies within the defined range. Notably, due to the Amplifier's higher voltage requirements, the subsystem is powered by the POCS's 18V connection. The subsystem is also connected to the **Power Output and Conversion Subsystem.** The following circuit diagram (Wenzel, *Audio Amplifiers*) represents the Output Subsystem. The circuit includes the speaker output, sound input, and the potentiometer which will allow volume adjustment. Note: although the volume variability functionality is a part of the Output Control Subsystem, it is innately built into the Volume Amplifier. Its requirements are included here.



Requirements and Verification

Requirements	**Verification**
- The amplifier must be able to receive and amplify waveforms provided by the **Output Control Subsystem,** and output an amplified waveform with a gain between 20 and 200 to the speaker - The Speaker must be able to output frequencies within the range defined by the high-level requirements, which reach up to 15KHz - The Speaker must be able to output sound with a power of 0W to 20W	- Utilize Oscilloscope to analyze spectrum of sound, reading the original unamplified wave and the amplified wave and visually determining the gain - Utilize Oscilloscope to analyze the range of the Speaker, with test input - Utilize Multimeter to assess power output of Speaker, with test input

Cost and Schedule

Cost Analysis

Labor is calculated by summing up all the costs for each task. Costs for each task is found by multiplying the rate, members partaking in tasks and the time needed.
Labor: $20/hour * 475 hours = $9500
Parts:

Description	Manufacturer	Part #	Quantity	Cost ($)
Raspberry Pi 3 Model B Board	Raspberry Pi		1	65
SSOP-16 TO DIP-16 SMT ADAPTER	Chip Quik Inc.	PA0182-ND	3	11.07
SMOOTH FLOW LOW TEMP SOLDER PAST	Chip Quik Inc.	315-NC191LT10-ND	1	7.95
SPEAKER 8OHM 800MW TOP PORT 88DB	Soberton Inc.	433-1104-ND	3	5.25
IC AMP CLASS AB MONO 325MW 8SOIC	Texas Instruments	LM386MX-1/NOPBCT-ND	3	3.33
IC DAC/AUDIO 24BIT 200K 16 SSOP	Texas Instruments	296-41373-1-ND - Cut Tape (CT)	5	9.60
IC AMP CLASS AB MONO 700MW 8DIP	Texas Instruments	296-43959-5-ND	1	1.28
SPEAKER 8OHM 800MW TOP PORT 88DB	Soberton Inc.	433-1104-ND	1	1.75
IC DAC 10BIT	Microchip	MCP4911-E/P-ND	1	1.97

V-OUT 8DIP	Technology			
CAP CER 10UF 50V X5R 1206	Murata Electronics	490-12456-1-ND	10	4.05
Varying Capacitors and Resistors	N/A	N/A	30	3

Sum: $9500 + $114.25 = $9614.25

Schedule

Below is a schedule that contains the tasks needed to complete, by week until the end of the project is completed:

PROJECT TIMELINE

	OCTOBER					NOVEMBER				DECEMBER
	2	9	16	23	30	6	13	20	27	4
PROJECT WEEK	1	2	3	4	5	6	7	8	9	10
Encode Raspberry Pi								Fall Break		
Need to output digital waveforms for DAC Debugging and Proof-of-Concept. Also need to implement Serial Protocol Reading and Other Processing.	Encode DAC Communication Protocol		Encode DAC Communication Protocol to work with 24-bit DAC		Test DAC Output			Fall Break		
	Encode Custom Test Waveform							Fall Break		
		Encode Wave Table Lookup and Test using Mock MIDI Input						Fall Break		
		Encode USB Serial Protcol to take in MIDI Input from External Device						Fall Break		
COURSE DATES								Fall Break		
Notable Course Due	Project Proposal Rewrite (due 10/9 11:59p)						Mock Demo During Weekly	Fall Break	Final Demo With Instruct	Final Presentation With

Dates							TA Meeting		or and TAs	Instructor and TAs
	First Round PCBWay Orders (MUST PASS AUDIT BY DATE) (due 10/10 4:45p)		Third Round PCBWay Orders (MUST PASS AUDIT BY DATE) (due 10/24 4:45p)				Team Contract Fulfillment (due 11/18 11:59p)	Fall Break	Final Papers (due 12/6 11:59p)	
	Teamwork Evaluation (due 10/11 11:59p)			Individual Progress Reports (due 10/25 11:59p)				Fall Break	Mock Presentation with Comm and ECE TAs	Lab Checkout (12/7 3:00p-4:30p With TA)
		Second Round PCBWay Orders (MUST PASS AUDIT BY DATE) (due 10/17 4:45p)						Fall Break		Lab Notebook Due (12/7 11:59p)
Get the DAC Working								Fall Break		
Need to test and validate input and output. Depends on Raspberry Pi Encoding.	Solder DACs onto Breakout Boards		Build Oscillator Circuit					Fall Break		
		Test DAC Output with Raspberry Pi Input						Fall Break		
		Build Piezoelectric Clocks and Test Related Frequencies						Fall Break		

PCB Development								Fall Break		
Need to develop PCB Schematics with defined parts. Will plan to use currently available parts.	Build First PCB Prototype - may not be printed for now but at least have a draft ready for inspection by second round.			Build Fourth PCB Prototype if necessary				Fall Break		
		Build Second PCB Prototype and Debug			Design PCB Housing			Fall Break		
			Build Third PCB Prototype and Debug					Fall Break		
Power Subsystem Design								Fall Break		
Subsystem Design can be the last thing we implement, should be pretty simple. Need to find the appropriate circuits to achieve target voltage.	Research Power Parts needed to build a Booster Circit	Test Buck Converter (Use first round PCB to do so)	Research and Build Step-Up Converter Circuit					Fall Break		
	Make First Iteration of PCB							Fall Break		

Discussion of Ethics and Safety

Ethics and Safety issues continue to be an important aspect of the project's development. While the project does not have any particularly glaring ethics violations, possible breaches will likely stem from the standards of integrity and ethical conduct practice by engineers working on the project. According to the IEEE Code of Conduct, engineers should "commit [themselves] to the highest ethical and professional conduct and agree to uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities". In the context of such, the personal integrity of the engineers as well as the conduct must be monitored, as requirements and deadlines may drive project members to breach the provided ethics through cutting corners. To avoid this, proper documentation and group accountability should be implemented, as well as a sufficient design plan and timeline should be implemented to discourage members from breaching the code of ethics.

A Safety issue that may stem from the completion of the project may be the misuse of the project to damage the physical wellbeing of another person - a person may misuse the project and damage their own hearing or another's, either through the prolonged exposure to the maximum volume the device can output, or through the installation of a much more powerful speaker in the place of the intended device. Methods to prevent this are innate in the device's design - the project is not designed to produce more than the allowed volume range to prevent the user from damaging their own hearing. OSHA regulations state, "Exposure to loud noise kills the nerve endings in our inner ear…the employer must reduce noise exposure through [various methods]…to attenuate occupational noise", indicating that exposure to loud noise should be avoided if possible.

An additional safety concern may stem from the possibility of electrocution - though highly unlikely, the operation of an electronic component without sufficient voltage regulation may result in the electrocution and subsequent injury of an individual. Voltage regulation and proper securement of the project will prevent the possibility of such, and subsequently will be included in the final design. Since the project proposal, the concern of electrical shock during development has arisen as prototyping has begun - warranting the use of proper safety procedures during testing.

The possibility of ethical and safety issues will always be present in the development process of projects, as well as after their development. By enforcing group accountability amongst members and working with ethics in mind, these issues can be addressed as they arise, in the proper setting and with the proper actions. Ethical and Safety issues will continue to be assessed with the development of the project.

Safety procedures to address the provided safety concerns are provided in the following text. While the issue of excessive volume is addressed through the design of the circuit, limiting the output volume, electrical safety during the development of the project requires appropriate procedures. Namely, standard electrical safety procedures should be followed during the development of the project. Developers should shut off voltage while working on the circuit, always work with another person to avoid major injury, ground items properly and limit voltage to safe ranges, and avoid leaving wiring and electrical components exposed (University of Washington Environmental Health and Safety, *Basic Electrical Safety*). These standards should be implemented to reduce the possibility of injury during the development process.

Citations

Basic Electrical Safety. EHS. (n.d.). https://www.ehs.washington.edu/fire-life/basic-electrical-safety#:~:text=Electrical%20safety%20basics,for%20an%20activity%20or%20event.

Dsadmin. (2020, June 29). *GPIO I/O clock speed*. Digital Shack. https://digitalshack.org/gpio-i-o-clock-speeds/#:~:text=Absolute%20Max%20speed%20on%20Raspberry,and%20most%20cases%20much%20less.

Raspberry pi 3. Components101. (n.d.). https://components101.com/microcontrollers/raspberry-pi-3-pinout-features-datasheet

Wenzel, C. (n.d.). *Audio Amplifiers*. Audio amplifiers. http://techlib.com/electronics/audioamps.html

Wright, G. (2023, January 24). *What is MIDI (Musical Instrument Digital Interface)? – TechTarget definition*. WhatIs.com. https://www.techtarget.com/whatis/definition/MIDI-Musical-Instrument-Digital-Interface#:~:text=The%20MIDI%20protocol%20uses%208,data%20rate%20and%20is%20asynchronous.