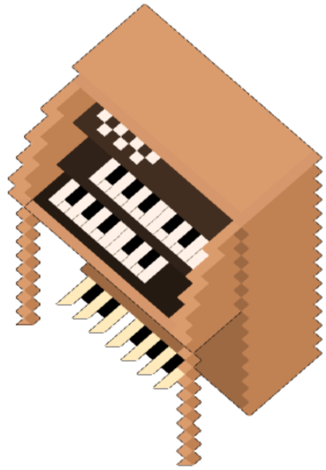


Project Plan

DEC13-09 Chiptune Synthesizer

Team Members:	Dustin Amesbury
	Wallace Davis
	Darren Hushak
	Brittany Oswald
	John Tuohy
Client:	Dr. Joe Zambreno
Advisor:	Dr. Nathan Neihart

Project Charter Version Control

Version	Date	Author	Change Description
0.0	19 Feb 2013	Dustin Amesbury	Document created
0.1	21 Feb 2013	Everyone	Group collaboration via GoogleDocs to fill out most of the sections
1.0	22 Feb 2013	Brittany Oswald	Assembled and formatted the first draft of the document
1.1	11 Apr 2013	Brittany Oswald, Darren Hushak, Dustin Amesbury	Implemented some of the feedback received from the first draft
1.2	12 Apr 2013	Brittany Oswald John Tuohy	Assembled and formatted second draft of the document

TABLE OF CONTENTS

Contents

1 Roles and Responsibilities

Dustin Amesbury -- *Project Leader* **--** As the project leader, Dustin is in charge of making sure the team stays on track throughout the semester as well as leading team meetings. Dustin will also be primarily involved with the implementing software in both Raspberry Pi modules.

Wallace Davis -- *Webmaster* -- Wallace is in charge of maintaining the public website and will be involved with implementing the songbox features. Wallace will also be assisting Darren with assembling and testing the electrical components of the chipophone.

Darren Hushak -- *EE Director* -- Darren will be the main advisor and executor of any hardware component of the project, including restoration of the organ console, its keyboards, and any of the original componentry that we wish to keep. Furthermore, Darren will be responsible for the conversion of the keyboards to MIDI controllers, addition of user input controls, and the interconnection between the MIDI controllers, the song playback and recording unit, and the sound modules. Lastly, as a music technology minor, Darren will advise and direct the group regarding standard music technology practices and music technology structures.

Brittany Oswald -- *User Experience Designer* -- Brittany is going to ensure the finished product is able to be used by users of all skill levels and abilities. Brittany will work on retrofitting and equipping the organ with the buttons and knobs and other user interface elements. She will also participate in writing code for the sound generation and ensure a quality experience for the user. Brittany's focus will be to see the project to completion.

John Tuohy -- *Communications Liaison* -- Along with communicating with the client and advisor, John is involved with designing and creating the software synthesizer on the Raspberry Pi.

2 Definitions and Acronyms

Term	Description
Channel (MIDI)	The MIDI specification allows for 16 MIDI “Channels,” which are used to direct different streams of data to different destinations. Each event has a MIDI channel associated with it, and will therefore be routed based upon its channel. The ‘broadcast’ channel is set per controller, and the ‘listen’ channel is set per module. All MIDI channels are broadcast to every module, but it is up to the module to pick the desired channel to listen to.
Chiptune/8-bit music	“Chiptune” music can be defined as a genre of music born out of the restrictive musical palette of 8-bit PCM audio used in the early popular gaming consoles. 8-bit refers to the bit depth of PCM audio, in that the digital form of the audio can only take on one of 2^8 values, which creates incredibly unique sounds.
Chipophone	The unofficial name of our chiptune synthesizer
Controller (MIDI)	A MIDI controller, most commonly a keyboard, is a device that takes user input and converts it to MIDI data.
Instrument/Voice (MIDI)	An instrument or voice is a software program or hardware component that takes MIDI data as an input and outputs either PCM or analog audio.
MIDI	MIDI, standing for Musical Instrument Digital Interface, is a standard for interconnection and communication between digital music devices. It is a serial protocol that only conveys events relating to musical performance, such as notes, note volume, and control values for manipulation of audio. MIDI itself does not convey any audio information.
MIDI CPU	A MIDI CPU is a programmable hardware circuit board that takes in analog or switched input from user interface devices (switches, potentiometers, encoders, etc), and outputs MIDI data.
Module	A MIDI module is a device that hosts an instrument/voice, which takes MIDI input and outputs audio. The module is the shell for the voice, and the voice refers to the algorithm or circuitry used to actually produce audio.
Raspberry Pi	Our chosen hardware platform for both our computational and signal processing needs. It is an incredibly small, yet powerful, bare-bones computer that runs a linux operating system.
Synthesizer	A synthesizer can refer to an analog or digital piece of equipment or software that can self-generate audio using various algorithms and techniques.

3 Executive Summary

Our challenge is to create a device (nicknamed "The Chipophone") that will be used as a display piece for the EE/CprE department. Our "chipophone" is a synthesizer that plays music using "chiptune" or "8-bit" sounds commonly associated with early console gaming systems. We are tasked with taking an existing electric organ (one which has already been donated to the project) and retrofitting it to play 8-bit music. All of the pedals, switches, and keys will be wired to create MIDI signals that will be interpreted and rendered into audio output. After the organ has been created, it will be placed in the TLA and will be used as a interactive showpiece to supplement the two arcade machines and the virtual pinball machine already located therein.

4 Project Overview

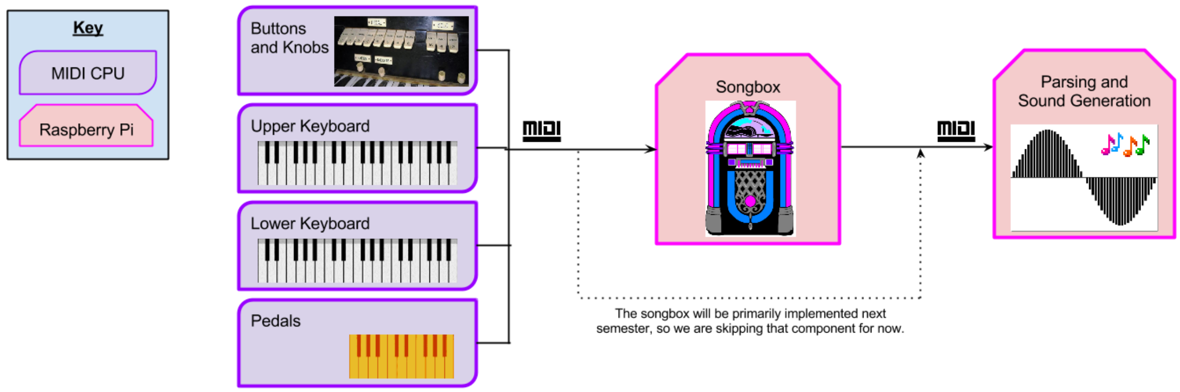
Our solution involves two stages: MIDI retrofitting the organ, and creation of an audio generation unit. We will be using MIDI CPUs to generate MIDI signals, and Raspberry Pi microcomputers to process and interpret said MIDI signals. To retrofit the organ, we will remove most of the internal mechanisms and circuitry of the organ and rewire the keyboards and button/switchboard interfaces into MIDI CPU chips. These chips will then be routed into the Raspberry Pi boards, whose audio generation code is our responsibility. The Chipophone will also allow for many features native to digital synthesizers For example, it will hold in memory various voices loadable into five different modules simultaneously; user creatable voices; mixing of multiple channels, etc. An additional third stage, to be implemented in the second semester of the project, will be to create a "Songbox" module, which will allow for recording and playback of both preloaded and user generated songs.

5 Project Scope

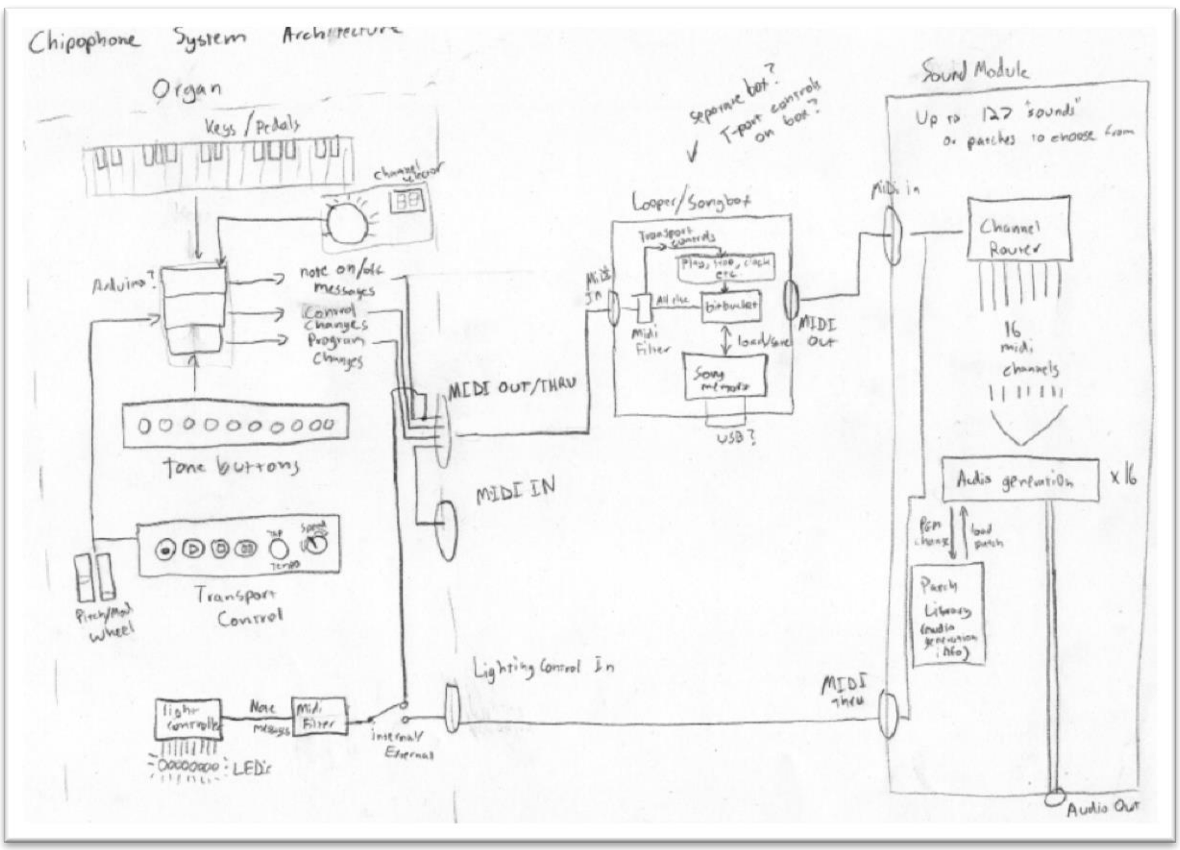
5.1 Goals and Objectives

Our expected product at the end of the year is to have a fully functioning organ that takes input from the keyboard, switches, and pedals of the organ and outputs "chiptune" sounds. The product will be designed with a broad user base in mind, be it students' recreational use or staff members using it as a recruitment tool to demonstrate the capabilities of undergraduate students in the EE/CprE departments. We want the entire system to be very usable and easy to understand. The system will also be designed to be very low maintenance, with minimal upkeep and longevity in mind.

5.2 Concept Art



5.3 System Overview Diagram



5.4 User Interface Description

The user will be able to interact with the chipophone in a similar manner as they would with any electric organ, with a selection of sounds and parameters available to them. The buttons, knobs, and display will be designed with the user in mind so that any new user will be able to learn how to use all the settings with a minimal learning curve, regardless of their musical ability.

5.5 Target Users

The intent of the project was to create an interesting recruitment tool, to be shown at campus visits and departmental functions. As such, the chipophone will be moved from room to room, and even across campus. In between presentations, we are assuming that it will be on unguided public display for students in a common area, for recreational playing. Because of this, we may have to augment the physical organ case to be robust yet mobile, in order to sustain longevity. Regarding users, we expect a wide variety of students, primarily enrolled and potential EE and CPRE students. Various other students will also use it at recruitment or departmental presentations. Between the wide range technical abilities of the user base and the unguided public display, we will need to be very conscious of robustness of programming, ease of use, and provide clear instructions.

5.6 Project Deliverables

Milestone	Deliverable
MIDI Controller	The original keys and pedals of the organ will be retrofitted with a MIDI CPU in order to convert the key presses into MIDI data. Each keyboard and the pedals will have their own user-selectable channel. We will add encoders and other forms of user input as time goes on.
Songbox	The songbox will be a unit to be placed in between the MIDI controller and the audio module. It will be capable of loading pre-sequenced songs from memory, and playing them. Users will be able to select songs, and whether or not they play the melody (allowing users to play along with the songs). Users will also be able to record and play back their own compositions via the songbox.
Audio Module	Receives MIDI data and interprets it. Based on the interpretations, it will render the appropriate audio.
Functional Blocks	We'll be splitting our Chipophone into two major functional blocks. The first block will be done in the first semester, and it involves a basic audio generation unit. The goal is to be using the keyboard as the sole input device at the end of the first semester with the ability to create basic sounds, without functionality such as modulation and various effects. During the second semester, we'll be starting on a long list of features that can be implemented as time permits.

5.7 Technology Choices

5.7.1 Raspberry Pi

We chose to use two Raspberry Pi modules for our system. The Raspberry Pi board offers a powerful, inexpensive platform that allows the system to be controlled by a Linux distribution. We will be using one Raspberry Pi to process standard MIDI input to render audio. The second Raspberry Pi will be used to implement the "songbox" module, a conglomeration of several synthesizer functionalities such as sequencing, recording and playback, and local song memory.

5.7.2 MIDI Standard

We’ll be using the MIDI standard as our communication protocol. We chose MIDI because it is easy to create a data stream that is simple to process within our Raspberry Pi module, and reduces risk inherent with designing and implementing our own protocol.

5.7.3 MIDI CPU

The MIDI CPU is a controller that generates digital MIDI and packs it into a MIDI data stream that will be processed by the Raspberry Pi. We’ll be using four MIDI CPUs in total; one for each keyboard, one for the pedals, and one for the controllers. The four MIDI CPU’s allow for daisy chaining, which ultimately means a single MIDI stream output to our Raspberry Pi audio generation and MIDI processing boards.

5.7.4 Existing Organ

One generous group member was able to contribute an organ he had available to him at his disposal. By using this donated organ, we are now able to expand features that we might not otherwise been able to due to budget constraints.

5.7.5 Operating System

We chose to use Linux as our operating system platform due to its ability to interact with a sound card at a very low level and general compatibility with the Raspberry Pi. We hope to achieve fast boot times and low latency, also common to Linux, giving it the feel of a typical synthesizer.

5.8 Project Estimated Costs

Product	Purpose	Price Per Item	Quantity	Total
MIDI CPU Encoder	MIDI output from keyboard	$44.95	4	$179.80
Raspberry Pi	Synthesis "OS" board and songbox	$53.85	2	$107.70
Blue LEDs	Status lights	$0.21	10	$2.10
Diodes	Key latch componnents	$0.04	150	$6.26
Contact Switches	Key latch componnents		150	
SD card	Memory for RPi	$7.49	6	$44.94
Encoders and other UI Controllers	User Interface controllers (estimate)			$150.00
				$490.80

6 Project Conditions

6.1 Project Risks

6.1.1 Raspberry Pi Maximum Run Time

Raspberry Pi has a known issue to become unresponsive after around 12 continuous hours of use. This issue could surface with users forgetting to turn off the Chipophone after use. We plan on addressing this issue by implementing an auto shutdown feature after a predetermined period of time sitting idle.

6.1.2 Raspberry Pi Boot Time

We want the time between pressing the power button to outputting sound to be as fast as possible. We don't want the organ to feel like a computer.

6.1.3 Splitting into Functional Blocks

Splitting the project into modules has many benefits, but in doing so we would be exposing the project to risk of not being able to plan the communication between modules correctly. Maintaining a loosely coupled system is the goal, while making sure our software modules have high cohesion. If the modules are not able to communicate, the finished project will suffer.

6.1.4 Organ Condition

The organ we are using is used, and thus will require some tender loving care to bring it up to date and ensure that it looks appealing and high-tech when we are done with it. It will also be a challenge fastening the hardware and retrofitting the organ to have all of the technology we will be using. Fortunately, there is plenty of space available inside the organ after all of the old innards are removed. The exterior of the organ is more of a concern than the interior.

6.1.5 Lack of Experience

Most members lack experience with music technology despite their enthusiasm for the topic.

6.2 Non-Functional Requirements

6.2.1 Usability

Synthesizers in general can greatly range in complexity and appear intimidating to new users. Our goal is to make our synthesizer easy to use for everyone, both advanced users who have specific goals when they play music and for a beginner user who wants to make some cool sounds.

6.2.2 Portability

The chipophone will be easily transported from event to event. It will have casters so it can be moved. The only cable will be a single power cable which plugs into an normal outlet.

6.2.3 Performance

The chipophone will have no noticeable latency between keypress and sound. The chipophone will also be able to boot and be ready to play in under 30 seconds.

6.2.4 Maintainability

If anything goes wrong with any of the Raspberry Pi boards, they will be able to be easily swapped out for a new one.

6.2.5 Security

There will be a lock on the back of the chipophone, preventing anyone from stealing the boards or other equipment.

6.3 Project Constraints

- Budget of $1000
- Fully functional pristine result by December 2013

6.4 Timeline

ID	Project Name	Owner	Days	Start	End	28-Jan	4-Feb	11-Feb	18-Feb	25-Feb	4-Mar	11-Mar	18-Mar	25-Mar	1-Apr	8-Apr	15-Apr	22-Apr	29-Apr	6-May
1.0	Dec13-09	Dec13-09	101	21-Jan	2-May	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
1.1	Scope Definition Phase					■	■	■												
1.1.1	Research MIDI and other	Dustin A, Brittany O,	17	22-Jan	8-Feb	■	■													
1.1.2	Define Design Scheme	Darren H.	4	28-Jan	1-Feb	■														
1.1.3	Determine Desired Features	Brittany O. and Dustin	10	29-Jan	8-Feb	■	■													
1.1.4	Aquire Organ	Darren H.	1	15-Feb	16-Feb			■												
1.1.5	Create Purchasing List	John T, Darren H,	5	1-Feb	6-Feb	■	■													
1.1.6	Purchase Items	Dustin A.	7	6-Feb	13-Feb	■	■	■												
1.2	Research and Design Phase							■	■	■	■	■	■							
1.2.1	Complete Raspberry Pi Tutorials	Dustin A. and John T.	10	13-Feb	23-Feb			■	■											
1.2.2	Determine how to minimize Linux	Dustin A. and John T.	30	18-Feb	20-Mar			■	■	■	■	■	■							
1.2.3	Identify and Mitigate Risks	John T.	2	21-Feb	23-Feb				■											
1.2.3	Clean and Repair Organ	Darren H. and Brittany	10	18-Feb	28-Feb			■	■	■										
1.2.4	Create software architecture	Dustin A, Brittany O,	7	19-Feb	26-Feb				■	■										
1.3	Implementation Phase									■	■	■	■	■	■	■	■	■	■	
1.3.1	Implement Software Architecture	Dustin A, Brittany O, Wallace D. and John T.	60	26-Feb	27-Apr					■	■	■	■	■	■	■	■	■		
1.3.2	Produce sine, triangle, square	Dustin A, Brittany O,	15	26-Feb	13-Mar					■	■	■								
1.3.3	Produce combinations of waves	Dustin A, Brittany O,	7	1-Mar	8-Mar					■	■									
1.3.4	Create 5 sound modules	Darren H.	30	1-Mar	31-Mar					■	■	■	■	■						
1.3.5	Adjust pitch and octave	Dustin A.	7	1-Mar	8-Mar					■	■									
1.3.6	Implement Arpeggio	Dustin A.	14	15-Apr	29-Apr											■	■	■	■	
1.3.7	Design user interface	Brittany O.	20	1-Apr	21-Apr									■	■	■	■			
1.3.8	Modify/Add buttons and knobs to	Brittany O.	30	28-Mar	27-Apr									■	■	■	■	■		
1.3.9	Implement 30 minute inactivity	John T	5	1-Apr	6-Apr									■	■					
1.3.10	Install hardware into the organ	Darren H.	20	18-Mar	7-Apr							■	■	■	■					
1.3.11	Repaint the organ	Brittany O.	2	25-Apr	27-Apr													■		
1.3.12	Create and include instructions	Brittany O.	2	1-May	3-May														■	

7 Project References

Description	Location
A similar project inspiring this one	http://www.linusakesson.net/chipophone/
Guidance on MIDI and Raspberry Pi interfacing	http://www.siliconstuff.com/
Raspberry Pi documentation, downloads, and information	http://www.raspberrypi.org
MIDI CPU information and documentation	http://www.highlyliquid.com
NES Audio Chip programming and architecture information	http://emureview.ztnet.com/developerscorner/SoundCPU/spc.htm
Audio Programming information and guidelines	The Audio Programming Book - Richard Boulanger and Victor Lazzarini

8 Appendices

TODO Fill in later if we need one.