Campanile-Carillon Model: Phase II

Design Document Version I - 3/26/19

sddec 19-12

Client: Dr. Tin-Shi Tam

Advisor: Gary Tuttle

Gabriel Stackhouse - Software Lead

Grant Mullen - Integration Manager

Kienan Otto - Report Manager

Ryan Roltgen - Meeting Scribe

Sam Habel - Meeting Facilitator

Yicheng Hao - Power Systems Lead

Email: sddec19-12@iastate.edu

Website: http://sddec19-12.sd.ece.iastate.edu/

Table of Contents

1 Frontal Materials

1.1 List of Figures

1.2 List of Definitions

SCLC - *Student Carillon Leadership Council - responsible for maintenance and transportation of the future campanile carillon model*

MIDI - *Musical Instrument Digital Interface- communication protocol, digital interface, and electrical connector standard*

Carillon - *A musical instrument composed of at least 23 carillon bells, arranged chromatically, and played by a keyboard that allows expression through variation of touch*

MCC - *Mobile Campanile Carillon Model*

2 Introduction

2.1 Acknowledgement

We would like to acknowledge Dr. Tin-Shi Tam and Gary Tuttle for their guidance and support as we work to improve the program and design of the systems to be utilized in the MCC. We would like to thank the multiple ME 415 senior design classes that have worked on the MCC through the past few semesters and providing us with help on space and mounting information. Finally, we thank the Stanton Memorial Carillon Foundation and all the alumni and friends of Iowa State University who are providing the financial support to make this project a reality.

2.2 Problem and Project Statement

The Iowa State Campanile has been a symbol of ISU pride since the tower's construction in 1897 to memorialize Margaret Stanton. However, alumni and friends of ISU can only admire the tower if they are on campus. Moreover, this audience may not be familiar with the beautiful sound of the 50 bells, as the carillon -- the instrument inside the campanile -- is typically only played at noon.

To remedy this, there has been an enormous effort to create a Mobile Campanile Carillon model (MCC). For the past 2 years many groups have been involved in the realization of this goal, and great progress has been made in that time. This model will in effect showcase the bells, display history and donor information, and implement a playable carillon at ground level. Our task is to design the electrical components of the model. In addition to the widescreen digital display to showcase a “documentary” slideshow, we will design a Guitar Hero style interface to make the carillon playable by a layperson.

2.3 Operational Environment

As the name suggests the intention of the model is to make the carillon mobile. In addition to mobility, it will need to be able to withstand extended periods of being both indoors and outdoors. This means that the components we select must be water resistant and able to handle temperatures in both the summer and winter months. Additionally, since the model will routinely be transported, it will also need to be durable and retain continuity during transport.

2.4 Intended Users and Uses

There are two distinct groups who will be interacting with the MCC: curious onlookers who aren’t familiar with playing an instrument, and musicians. Since the model will be displayed at many different venues, the client wants anyone to be able to interact with the model and make beautiful music, no matter what their musical background may be. To achieve this, we must design an interface that makes playing the instrument easy, intuitive, and most importantly, fun. However, because musicians will also be using the instrument, we need to be cautious that none of our solutions interfere with the expected response from the instrument that the Carillonneur expects.

The other users we must consider are the members of the Student Carillon Leadership Council -- the individuals responsible for upkeep and transportation of the MCC. Any system we implement must maximize reliability, so the model does not require constant maintenance. In addition, any components with reasonable chance of failure must be documented so that individuals with no electrical expertise can diagnose and replace the components in question.

2.5 Assumptions and Limitations

Assumptions

1. Although the MCC will need to withstand outdoor conditions. It will not be left outside during extreme conditions. Such as extreme cold, heavy rain/snow, or dangerous winds

2. We expect the MCC to be fabricated in April/May so that our designs can be implemented before the Summer of 2019

Limitations

1. Each component of the design shall be reliable and minimize failure rate.
2. Although no discrete financial limitations were specified, the project should minimize unnecessary costs while maximizing reliability and durability.
3. The design should be simple enough to be repaired and operated by a layperson. To assist in this, we will create documentation to help repairs and operation of the system.
4. The design should be in compliance with the goal to remain portable, and modular.
5. There will be limited software support from the team after we graduate.

2.6 Expected End Product and Deliverables

We expect to develop two independent final products requested by Dr. Tin-Shi Tam. Each will be used as marketing material for Iowa State University to take to various events to represent the school. The deliverables are:

1. A functional tutorial guide on how to play a song on a 27-key carillon. The hardware will be located inside the campanile model and will have a battery life of at least 8 hours. This program will be able to read and display songs imported by the SCLC, and will allow for different levels of difficulty depending on the skill of the operator.
2. A standalone, battery operated information station which will display information about the university as selected by Dr. Tin-Shi Tam. The station will also be able to display interactive 3D models of structures on campus, such as the Iowa State University Campanile. The battery life of this station will match the campanile system's, lasting at least 8 hours.

2.7 Nature of Content

As our program is intended to help users play music on a mobile carillon, we had to take certain precautions. Our software needs to be simple enough to not encounter issues when attempting to run in a non-development environment, and our hardware systems need to be low-powered enough to avoid losing power in the middle of an event. Our available space in the campanile is very low, so we are unable to include larger or additional batteries to prolong battery life on more powerful hardware.

3 Specifications and Analysis

3.1 Proposed Design

Our plan is to have a screen on the MCC that will display falling notes onto a keyboard that mimics the layout of the carillon batons (keys of the carillon). We have a program from the previous group that we are working on debugging to run reliably and efficiently. There is also an LED bar that perfectly lines up with the batons on the carillon that will turn on the light over the note that should be played. This is for the user to have a secondary method to see what note should be played next. To control the LEDs and run the software we will use a Windows-based machine that is powerful enough for the program to run properly.

The group has made a decision on a monitor that meets the specifications of the available space while still being wide enough to have a one-to-one layout for each baton on the carillon. To interact with the program, we are planning on getting a touchpad or keypad for the device to move through the menu screens and to select what song to play. Once the song starts playing, the user doesn't need to provide any additional input until they decide to choose a new song.

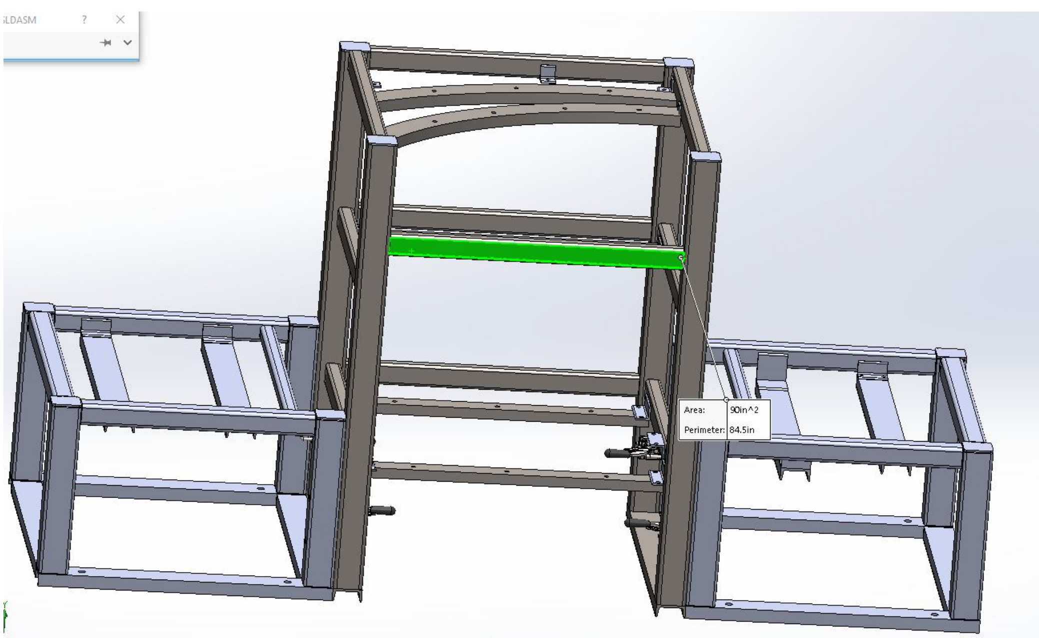

Figure 1: Monitor mount location on structure

We have recently discovered that the Raspberry Pi used by the previous group wasn't powerful enough to run the program at the desired resolution. The CPU was consistently overheating, and the program was suffering significant slowdowns due to inadequate hardware. For now we are proceeding with the backup Linux laptop that was purchased for the project. In the future, once our program is finalized and finalize the necessary hardware specifications, we will order a new device to run the program. It was also recently requested by the client that we port the program to Windows to provide more ease-of-use, so the new device we choose will likely be running Windows.

Our plan is to design and build a power system that will power our display, the Windows laptop, and any other peripherals needed. To accomplish this, we will use lead acid batteries, an inverter to power the monitor and the laptop, and a high amperage battery charger. We will likely make use of 300W to 400W inverters in order to have plenty of headroom to power all system components without overheating. The capacity (in aH) of our lead acid battery will be determined by our desired battery life. The capacity could always be increased at a later date by wiring additional batteries in parallel. The batteries we choose will likely be deep cycle marine batteries, similar to many batteries used in solar systems. Last, we will make use of a high amperage battery charger in order to charge the batteries to full capacity in a short amount of time. We currently have a bill of materials, but we are waiting to order until we get the go-ahead from the client.

One of the peripherals we are considering implementing is a battery monitoring system that will take the battery voltage and estimate our current battery life. This could be put onto a custom pcb alone with our microcontroller.

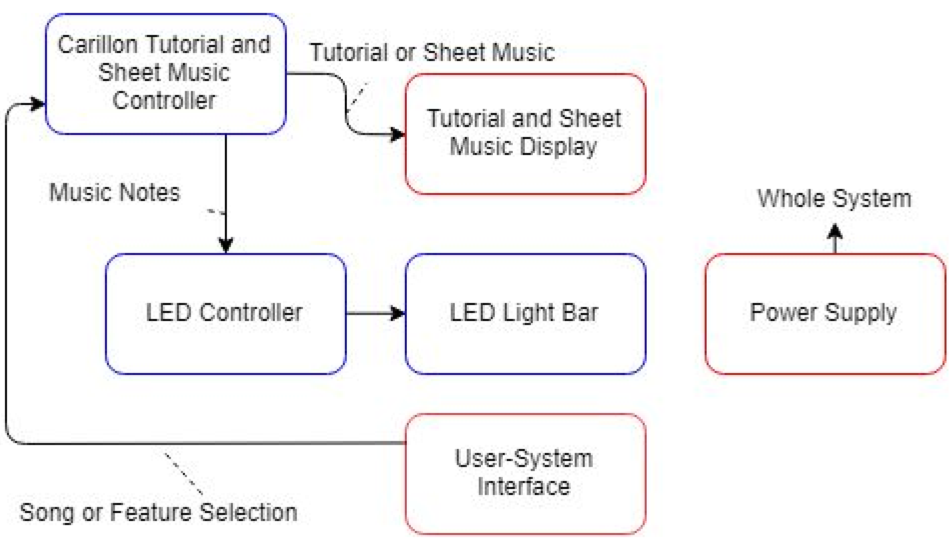

Figure 2: Block diagram of system

3.2 Design Analysis

We have working C++ code that can read MIDI files and play them in a Guitar-hero style, lighting up the corresponding LEDs on the light bar as the notes align. Despite this, the program was littered with bugs, and a good portion of the design thus far has been finding and eliminating the problems that we inherited. Ten weeks in, and we have a much more polished and reliable program, so we have been very successful in this regard.

We have recently been requested by the client to port the program to Windows 10, so work on that has begun in full. While the majority of the code has proven to port without any intervention, there are many sections that require full rewrites to support the new OS. Adding additional features to the software will be put on hold until the porting is complete, so as to avoid any further incompatibilities between operating systems.

Since the carillon model is designed to be mobile, we will have to be considerate of the space available for our system to fit in. This could cause problems if we rush on buying materials for the battery without ensuring that, when assembled, they will all fit in the allotted space. To help with this, we are meeting with the ME 415 team and referencing CAD files that have the exact dimensions of the model. Both will be extremely important as we continue to design and build the battery solution that will have the right balance between high capacity vs. taking up a small amount of space.

4 Testing and Implementation

4.1 Interface Specifications

The core of our group's carillon tutorial system will be a single computer. Initially, we will design everything to operate on a laptop which will provide power to the Arduino and light bar, as well as provide data to both the monitor and Arduino. All data will be processed onboard the central computer. Our original design called for a touch-screen monitor for user-interfacing, but because of the high cost of a touchscreen monitor that fits specifications, the design has been altered to use a small touchpad or keypad to navigate menus within the program.

4.2 Hardware and Software

A low-power laptop is being used to test our code, which is being written entirely in C++. Our final design will not use a laptop to run the program, but our final hardware solution has not been decided yet. In the past, our team has used a Raspberry Pi V3 for testing, but it does not have enough processing power to run the program smoothly. To test the final hardware solution, it must be able to run the program smoothly at 30 frames per second or higher for all songs. Our minimum operable frames per second is 24.

The code will be tested mainly through edge cases and routine operational tests. The edge cases will be empty MIDI files as well as MIDI files with a large amount of notes to put strain on the system. The routine test involves uploading a MIDI file and displaying it without errors. The program must also be able to display uploaded sheet music as a PDF.

The battery will be tested by running the system for 8 hours or more, which is the requested battery life of the system.

4.3 Functional and Non-functional testing

Our team has a few major points for our functional testing:

We are focusing our testing on a few of the major components, with the chief focus being placed on the light bar. The light bar is custom built with addressable LEDs, so we are testing for accuracy and synchronization. The LEDs light up in conjunction with a note falling down the screen. Specifically, an LED will light up when a note in its corresponding column reaches the bottom of the screen and turn off when the note leaves the screen. Our tests must conclude that the LEDs accurately function in coordination with the on-screen notes.

A major concern for our system, as well as any system, is making sure all new changes don't break old, previously-functional code. This requires testing all existing functionality at each iteration of our program. We need to ensure no feature introduces any unintended effects to the program.

Our program should be able to run at a minimum of the full monitor resolution and 30 frames per second. It should remain constant at these specs and show no fluctuations to maintain a smooth, consistent appearance.

The monitor should be able to display the graphics appropriately in both light and dark settings. This will require a group of people to observe the display from various distances and angles in different brightness situations to determine the correct settings and any changes to the graphics.

The non-functional testing is more simple. We have a few major, overarching requirements for our end product:

The product shall:

- Have a tutorial which is usable by any person of any skill level
- Display sheet music
- Display Donors
- Include an interface for adding additional midi files and sheet music
- Be easily troubleshooted

4.4 Process

LED lights - Tested using the arduino

- Compatibility tests
- Connection of wires

Screen - Tested efficiently

- Size sufficient for keyboard
- Testing brightness settings

Software - Tested through the use of a survey

- Usability
- Reliability
- Aesthetic appeal

4.5 Implementation Results and Challenges

Much of the testing will need to be done without the final hardware available. Due to the amount of time needed to acquire the correct monitor and processor, our testing is limited to running the program on a laptop and displaying at a custom resolution on a simple desktop monitor. Debugging the program will be difficult because our team has inherited the majority of the code from the previous group.

If needed, fixing the light bar will be a huge roadblock. We have only one electrical engineer on the team, and the previous group did not provide documentation on their work on the light bar. Our integration with the ME group that is creating the frame for the carillon model may take more time than expected. Our group has struggled to analyze the 3D model of the frame, and we are limited in our ability to plan the space we will need for the various components of the system.

4.6 Simulation and modeling

Regarding simulation, we have done the initial math to calculate our battery life based upon our inverter efficiency, battery capacity, power requirements of our components, and our desired battery life. We also plan to simulate our battery monitoring system before drafting our PCB. This will be in addition to the breadboard and perfboard models we will make. We don’t currently have plans to simulate software, as the high level of interaction makes this difficult.

5 Closing Material

5.1 Conclusion

We have the majority of the code working properly and are continuing to debug the code and add features that the client wants us to work on adding. We plan on continuing our work while documenting the progress we make and producing the documents to give a proper tutorial for the people who will be in charge after we leave.

We are still tackling problems that exist within our code that has to do with the light bar not lighting up properly when the notes reach the bottom of the screen. In addition, we are also working on porting the current program from Linux over to Windows so that it is easier to get working for someone who is unfamiliar with technology and doesn’t want to set up a Linux machine by themselves.

Finally, we want to revamp the design of the program to look more professional and less grey, so that the user experience is more improved and looks more visually pleasing. This is a low-priority effort that we will only being work on once the bugs have been fixed and the program is running smoothly as expected.

5.2 References

Website of previous group: https://sddec18-01.sd.ece.iastate.edu/