SDSU

College of Engineering
Electrical and
Computer
Engineering

EE/COMPE 491 Senior Design

Team 08 (Team Boom)

EE/COMPE 491W

Team Boom

Prototype Activity Report

May 2, 2024

Team Boom

Antonio Balestrino

Software Lead

Blake Chanhmisay

Software Team

Rogelio M Hernandez

Team Lead/Hardware

Faith Murray

Software Team

Tristan Storbeck

Hardware Lead

Executive Summary

Overview

This document outlines Team 8's final report for our Senior Design Project: a powered Bluetooth speaker, codenamed "Boom Tweet Tweet." Below is a brief outline of the sections covered in this document.

A product overview is first given in the system description section with a graphic detailing the proposed design. A system test procedure is then detailed. The initial identification of risks section highlights the most significant risks determined after the risk management meeting but before prototyping activities began. Following this, the selected prototype activity is described in detail. Test setups, procedures, results and an analysis of the results are included. Finally, after having completed the prototyping activity, the risk analysis is updated to reflect the updated project risks with the remaining risks being addressed by specific strategies to combat them moving forward.

Changes in Functionality

A pairing button and led has been added to the system diagram after risk mitigation for added functionality and system testing.

Contents

System Description

Proposed Design

Team Boom is working to create a Bluetooth speaker called Boom Tweet Tweet. It is powered by an external DC power supply that plugs in to J1 on the back of the speaker. Turning on the speaker with the power switch (SW1) will illuminate LED1 to show that the speaker is on and ready to connect. The speaker will connect via Bluetooth to another Bluetooth enabled device to allow audio to be played wirelessly. Volume may be controlled by the speaker volume knob (POT1) or by the mobile device's volume control.

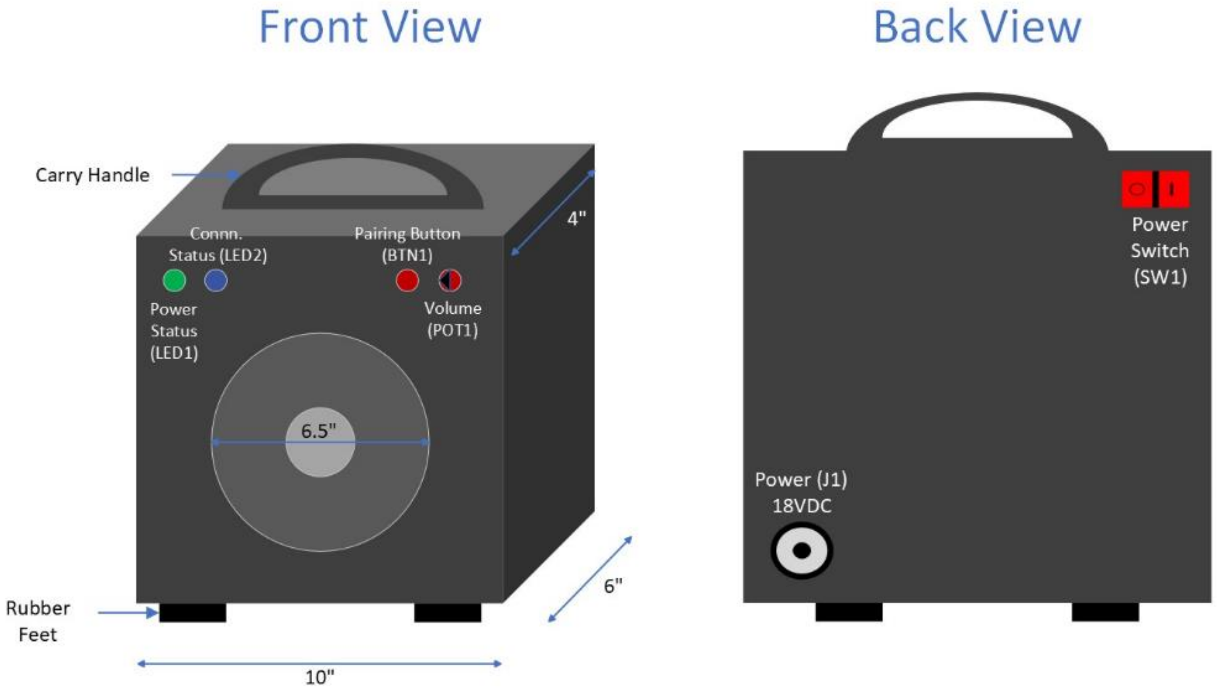


Figure 1 - System Description

System Level Block Diagram

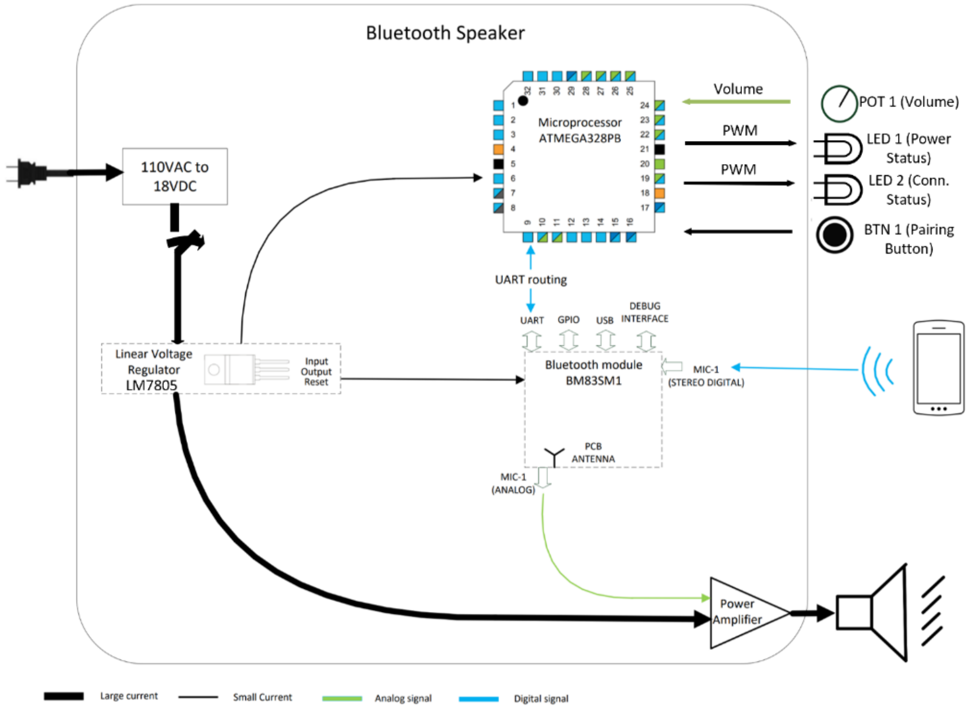


Figure 2 - System Block Diagram

Boom Tweet Tweet will utilize components such as an 18VDC power supply together with an LM7805 voltage regulator to supply 5V to the ATmega 328PB microcontroller, and BM83 Bluetooth module. A class AB amplifier will supply a maximum power output of 16W supplied to a 6.5” speaker for audio output. Figure 2 demonstrates the system level layout of Boom Tweet Tweet.

Figure 3 below shows the system state diagram which covers the various modes of operation for our powered Bluetooth speaker. The states cover the functions of the BM83 Bluetooth module which encompasses: Standy Mode, On, Play audio, and communications with the ATMEGA328PB microcontroller.

System State Diagram

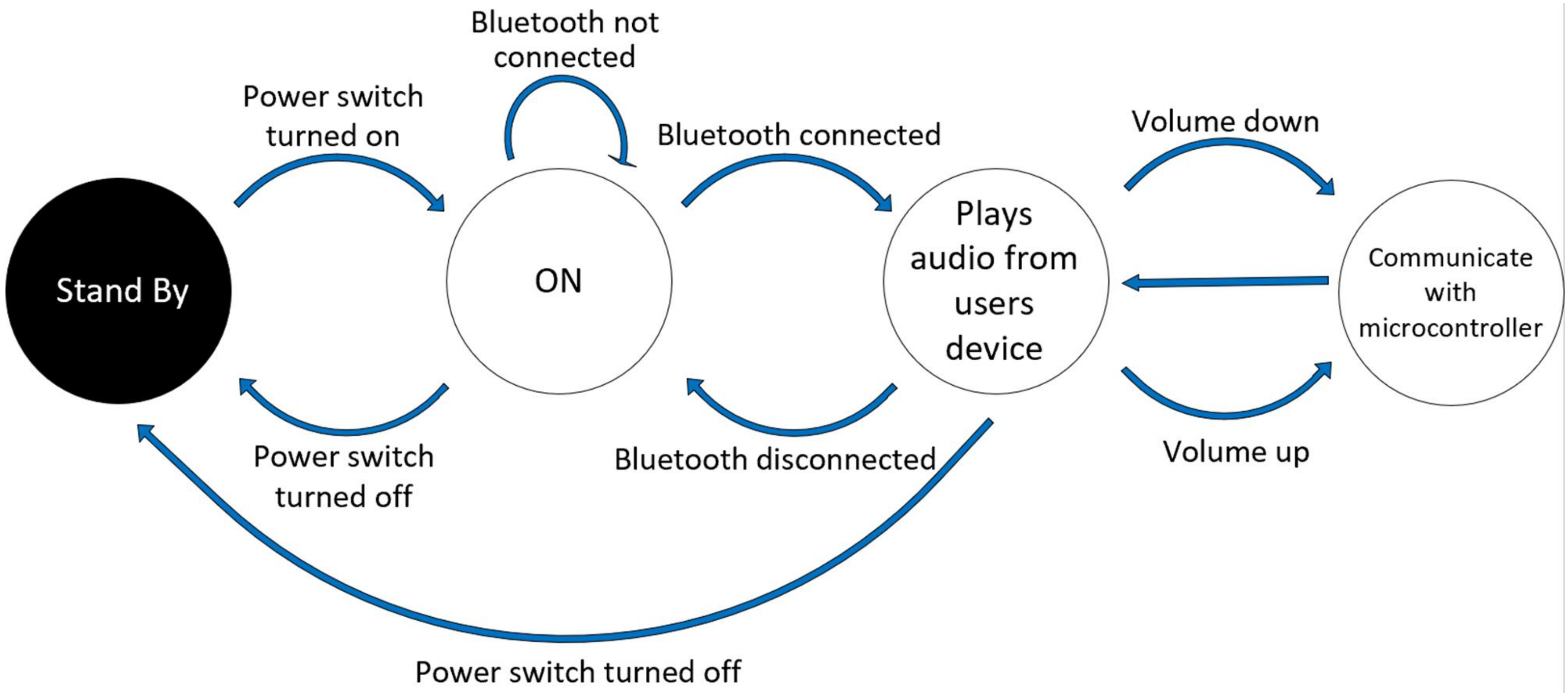
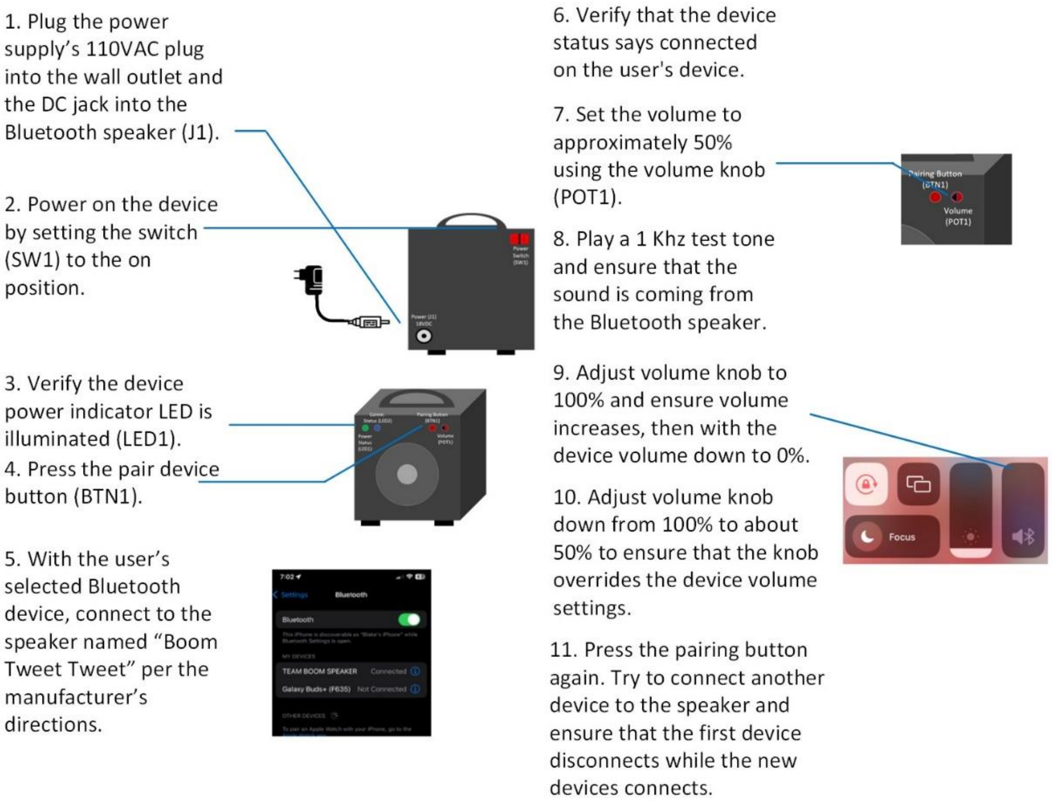


Figure 3 - System State Diagram

System Test - Functionality

Note: This test assumes the user can play a sound file on their chosen Bluetooth-enabled device. The desired sound should be played on the device's built-in speaker before the test to eliminate the possibility of the user's device being in error. This test should also be performed in a quiet setting such that the ambient noise levels do not jeopardize the results of the audio output level test. To complete the test, a sound pressure level meter (SPL) will also be required. The infographics below in Figure 4 and Figure 5 detail the steps to complete the system test.



1. Plug the power supply's 110VAC plug into the wall outlet and the DC jack into the Bluetooth speaker (J1).

2. Power on the device by setting the switch (SW1) to the on position.

3. Verify the device power indicator LED is illuminated (LED1).

4. Press the pair device button (BTN1).

5. With the user's selected Bluetooth device, connect to the speaker named "Boom Tweet Tweet" per the manufacturer's directions.

7:02

Settings Bluetooth

Bluetooth

This iPhone is discoverable as "Blake's iPhone" while Bluetooth Settings is open.

MY DEVICES

TEAM BOOM SPEAKER Connected

Galaxy Buds+ (F635) Not Connected

OTHER DEVICES

To pair an Apple Watch with your iPhone, go to the

6. Verify that the device status says connected on the user's device.

7. Set the volume to approximately 50% using the volume knob (POT1).

8. Play a 1 Khz test tone and ensure that the sound is coming from the Bluetooth speaker.

9. Adjust volume knob to 100% and ensure volume increases, then with the device volume down to 0%.

10. Adjust volume knob down from 100% to about 50% to ensure that the knob overrides the device volume settings.

11. Press the pairing button again. Try to connect another device to the speaker and ensure that the first device disconnects while the new devices connects.

Figure 4 - System Functionality Test

System Test - Performance

1. In a quiet room, using a SPL device, place the audio level testing device 1 meter from the speaker and play a 1Khz sound file from the connected device. (Using YouTube or an app).

2. Once again set the volume to 100%.

3. Ensure that the sound level meter reads a minimum of 87 dB corresponding to an average output power of 1W and a peak power of 16W.

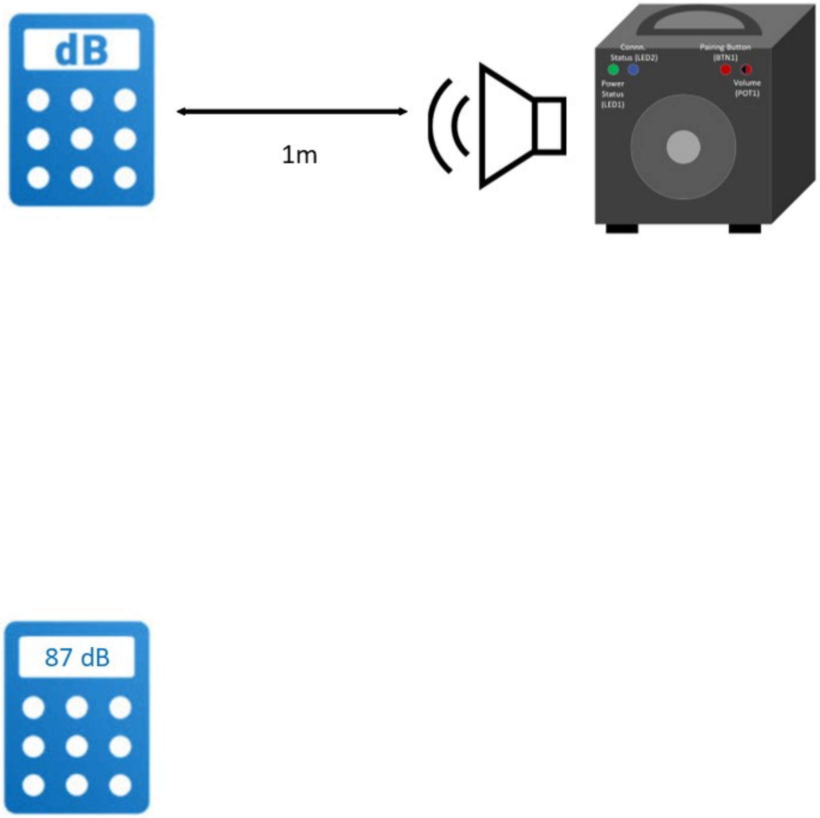


Figure 5 - System Performance Test

Key Technological Challenges and Strategies

First Challenge

We identified three key challenges in this project that we will have to address as we move forward. One of these challenges was regarding the design of the Printed Circuit Board (PCB). There are a variety of considerations to concern one's self with in this task. Component layout should be considered such that devices perform in an optimal manner. This includes placing audio circuitry away from noise-generating devices and keeping radio frequency (RF) devices from being placed above ground planes that would attenuate the signal. You must also consider how to route the traces so that there is enough clearance between them to prevent arcing and short circuits. Lastly, you must also consider the size of the trace which is a factor dependent on current, length of the trace, and thickness of the trace.

Second Challenge

The next challenge we have started to address is UART communication. This communication protocol is essential for our project's completion because without this line working properly, the microcontroller and Bluetooth module would not be able to send and receive signals between each other. One issue that we have addressed during our prototype activity was the difference between logic high values on the ATmega328 and BM83. We found that the ATmega has a logic high value of 5 V, while the BM83 has a logic high value of 3.3 V. To remedy this issue, we created a voltage divider to step down the voltage being driving to the Rx pin on the BM83. After further research, we found that the 3.3V that was being sent back to the ATmega was within range of its logic high, and no step up was needed. Another issue we faced, and we will need to continue prototype testing with, is understanding and transmitting data according to the BM83 UART protocol. In this protocol, the BM83 expects to receive six bytes from the microcontroller to properly execute a function. We must continue to test sending different OP code and parameter values to make sure that the BM83 responds the way we desire.

Third Challenge

The key challenges of configuring the BM83 require three documents provided by Microchip Technology. The Stereo Audio Module Data Sheet has a table of the functions that the pins provide and offer several configuration modes to allow communication between a microcontroller and the Bluetooth module itself. Firmware update procedures and speaker modes are detailed in the Application Designs Guide. The Host MCU Firmware Development Guide gives the reader a host of op codes and parameters sent as commands through a Host MCU to configure and control the BM83.

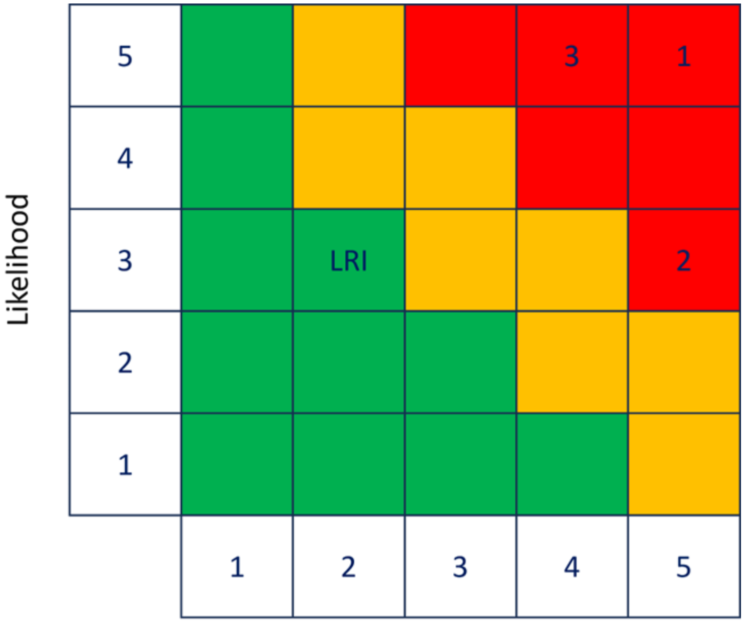
One key challenge was configuring the BM83, which was finding a configuration mode that allows a computer to communicate with the BM83. Referring to the Stereo Audio Module Data Sheet. We found that Test Mode would allow a computer to update the firmware of the BM83. Afterwards, we had to put the BM83 into Application Mode to send commands over to the BM83. Using software provided by Microchip, we flashed firmware onto the module using three files. The Config GUI tool allowed us to edit the firmware, the isUpdate tool was used for updating the BM83, and the SPKCommandSetTool was used to send commands.

The last key challenge was finding out how to make the UART connections to allow the PC to update, configure, and send commands over to the BM83. A 3.3V logic level UART-to-USB device was used so that a computer could interact with the BM83. The BM83 was put into Test mode using pin 22 to send power to the BM83, pin 50 to ground the BM83, and pin 31 to change the mode of the BM83. Pin 30 and pin 29 were connected to the RX and TX of the UART-to-USB device. After making these connections, we powered on the module and successfully paired a Bluetooth device to the BM83.

Identification of Risks

The following demonstrates a detailed list of all risks identified for prototyping. The risks are organized in order of severity. The top 3 greatest risks are shown first followed by a list of the Low-Risk Items (LRI).

The Amplifier is identified as the top risk since we don't have a schematic, layout, specifications, or physical component. The intent is to design an amplifier and as a last resort to acquire a Class AB amplifier from Prof. Dorr. Since there is no tangible material to work from, the amplifier is deemed as the top risk. The Atmel ATmega Microcontroller was identified as our second risk due to not having a means to directly program the Integrated Chip (IC). For our prototype activity, an Atmel Xplained board was used which is shown in our Prototype Activity section. Our third risk is the BM83 Bluetooth module with uncertainties such as connecting with a device and updating the firmware to communicate with the microcontroller. Our initial risk table is shown below.



Likelihood

5	🟩	🟨	🟥	🟥 3	🟥 1
4	🟩	🟨	🟨	🟥	🟥
3	🟩	🟩 LRI	🟨	🟨	🟥 2
2	🟩	🟩	🟩	🟨	🟨
1	🟩	🟩	🟩	🟩	🟨
	1	2	3	4	5

Figure 6 - Initial Risk Cube

1. **Amplifier**
2. **Device Programming**
3. **Bluetooth Configuration**
4. **Low Risk Items (LRI)**

Risk	Name	Description	Probability	Impact	Mitigation
1	Amplifier	The amplifier shall receive a voltage of 18V ± 5% to supply an 8Ω speaker with a peak power output of 16W.	With only a preliminary hand calculation there is no spec or schematic of the amplifier circuit since it is a design procured from Prof. Dorr. (5)	Without a concrete design or finalized specs there will be no way to amplify the speaker's signal. (5)	Acquire schematic & spec from Prof. Dorr.
2	Device Programming	The microcontroller will have to be programmed. Atmel Studio and C language to be used. We have never programmed between ICs before.	Configurations will have to be made to initialize/set-up UART, logic control, and audio processing. Must become familiar with Atmel ICE since we have never used it before. (5)	Being the heart of the project, the microcontroller being programmed correctly is essential for its success. (4)	We will purchase the Atmel ICE debugger to program the microcontroller.
3	Bluetooth Configuration	The Bluetooth device must be able to connect to another Bluetooth-enabled device.	The Bluetooth device must be able to connect to another Bluetooth-enabled device. (3)	If no connection can be made to the speaker, the project is unsuccessful. (5)	Bluetooth is upgraded using OTA DFU, then put into host mode and test mode. Once in host mode and test mode, UART commands can be sent over.

4	Time Management	Everyone has various commitments and knowing how to prioritize time is important to project success.	LRI	LRI	Everyone understands the time commitment required: 10 hrs. each/weekly.
5	Component Acquisition & Damaged Parts	Hardware being ordered through suppliers.	LRI	LRI	Order extras & minimum 2 weeks ahead of Prototype Activities.
6	Printed Circuit Board (PCB) Design/Layout	Computer Aided Design of PCB utilizing software.	LRI	LRI	KiCAD Tutorials – Prof. Dorr: YouTube.
7	SMD Soldering, ESD	Soldering components onto PCB.	LRI	LRI	Mark Bruno assistance. Grounding straps, rubber mats, ESD protective bags.
8	Debugging	Troubleshooting code.	LRI	LRI	Atmel ICE.
9	Enclosure Construction	Wooden Enclosure.	LRI	LRI	Construct Ourselves or send to Woodshop.
10	Total Cost	Cost of Project.	LRI	LRI	Everyone splitting costs equally.

Table 1 - Assessed Risks Before Prototyping

Prototype Activity

Hardware team

Due to time constraints and no amplifier module, the amplifier did not undergo a prototype activity. The focus of the Prototype Activity was to ensure that the BM83 Bluetooth module would be set up properly along with having the ATmega 328PB microcontroller established with tangible code to communicate with the Bluetooth module. To begin prototyping with our Bluetooth module and microcontroller, the Bluetooth module must first be soldered onto its carrier board.

The integration of the Bluetooth module onto the carrier board alongside the capacitor was our first step in prototyping. Despite encountering challenges due to our lack of experience with such small components, we navigated through the initial hurdles, including unintended bridging of connections. Through meticulous rework and careful application of soldering techniques, we successfully rectified these issues, ensuring proper isolation of connections and restoring the integrity of the assembly.

For easier prototyping, we soldered female headers onto the board, enhancing modularity and facilitating future iterations of the design. In conclusion, our team's dedication to precise soldering led to an operational prototype, ready for thorough testing and refinement.

Software team

In our initial risk assessment, we identified three critical risks whose mitigation was essential for our project's advancement: the amplifier, device programming, and Bluetooth Configuration. In this prototype activity, we decided to target the mitigation of device programming risks and Bluetooth connectivity. These two risks were prioritized because of their significant role in the project; without these risks being mitigated, we would be significantly hindered in our ability to progress the project’s functionality and features.

Bluetooth Configuration was selected as our initial prototyping activity to ensure the BM83 Bluetooth module is set up for later prototyping activities down the road. This step introduces the COMPE team to the BM83's configuration modes. It enables the team to understand how the BM83 transmits commands over UART using its pins and to gain proficiency with the configuration tools provided by Microchip Technologies.

The COMPE Team decided to use the Over-the-Air Device Firmware Upgrade procedure in the BM83 Applications Design Guide (page 76) to update the BM83. Electrical specifications were researched using Audio Module Data Sheet (page 57). The BM83 was kept in embedded mode by supplying 5V DC to pin 22 and grounding pin 50 on the BM83. The OTA DFU procedure outlines the three hex images used to generate a firmware image for updating the BM83 module wirelessly via Bluetooth. After rehexing the images using the isUpdate tool, the BM83 module was updated using the Microchip Bluetooth App, as illustrated in Figure 7. Unfortunately, this update process did not proceed as expected and resulted in failure. Initially, the COMPE team planned to utilize the command prompt feature within the app (see Figure 8) to transmit commands to the BM83 over Bluetooth. However, following the update, the app was unable to detect the BM83, preventing any further communication with the BM83. The COMPE team was hoping to use OTA DFU procedure as an easier alternative to updating and sending commands to the BM83, but there was no documentation on the problems they had. These results elevated the risk for Bluetooth Configuration.

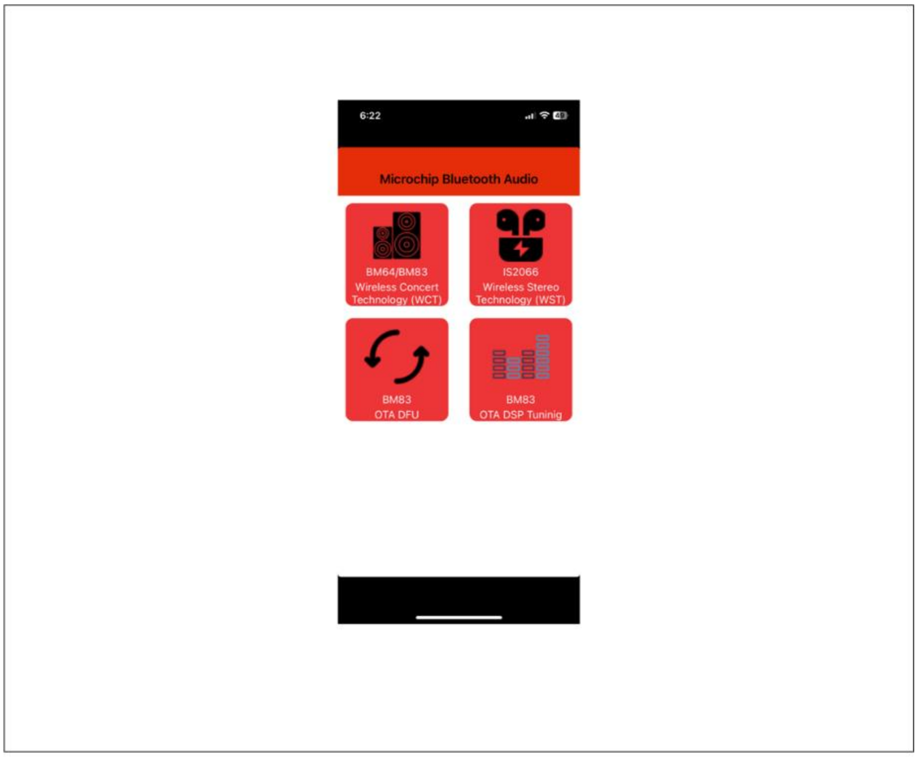
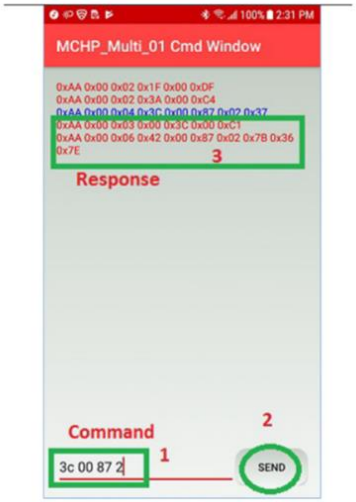


Figure 7 - Microchip Bluetooth Audio App



MCHP_Multi_01 Cmd Window

```
0xAA 0x00 0x02 0x1F 0x00 0xDF
0xAA 0x00 0x02 0x3A 0x00 0xC4
0xAA 0x00 0x04 0x3C 0x00 0x87 0x02 0x37
0xAA 0x00 0x03 0x00 0x3C 0x00 0xC1
0xAA 0x00 0x06 0x42 0x00 0x87 0x02 0x7B 0x36
0x7E
```

Command: 3c 00 87 2

Figure 8 - Command Prompt Feature

After encountering difficulties with the OTA DFU procedure, the COMPE Team shifted their focus to using UART for sending commands to the BM83. They set the Bluetooth module to test mode, following the set up detailed in Figure 9, and employed 3.3V logic levels for communication between the UART-to-USB device and the BM83. To eliminate the possibility of firmware corruption from the OTA DFU process, the module was re-updated using the isUpdate tool. The SPKCommandSetTool was then used to establish a connection between the BM83 and a computer, enabling the transmission of commands to the Bluetooth module. Using the system tab in the SPKCommandSetTool, we were able to send a power on and pairing command to the BM83. This method allowed the team to not only power on the BM83 but also to successfully pair it with a phone, as shown in Figure 10. These actions led to an understanding of how the BM83's transmits and receives commands via UART. The results of using UART via the SPKCommandSetTool to send commands to the BM83 had lowered the risk of Bluetooth Configuration from its initial risk identification and negated the elevated risk from the OTA DFU procedure.

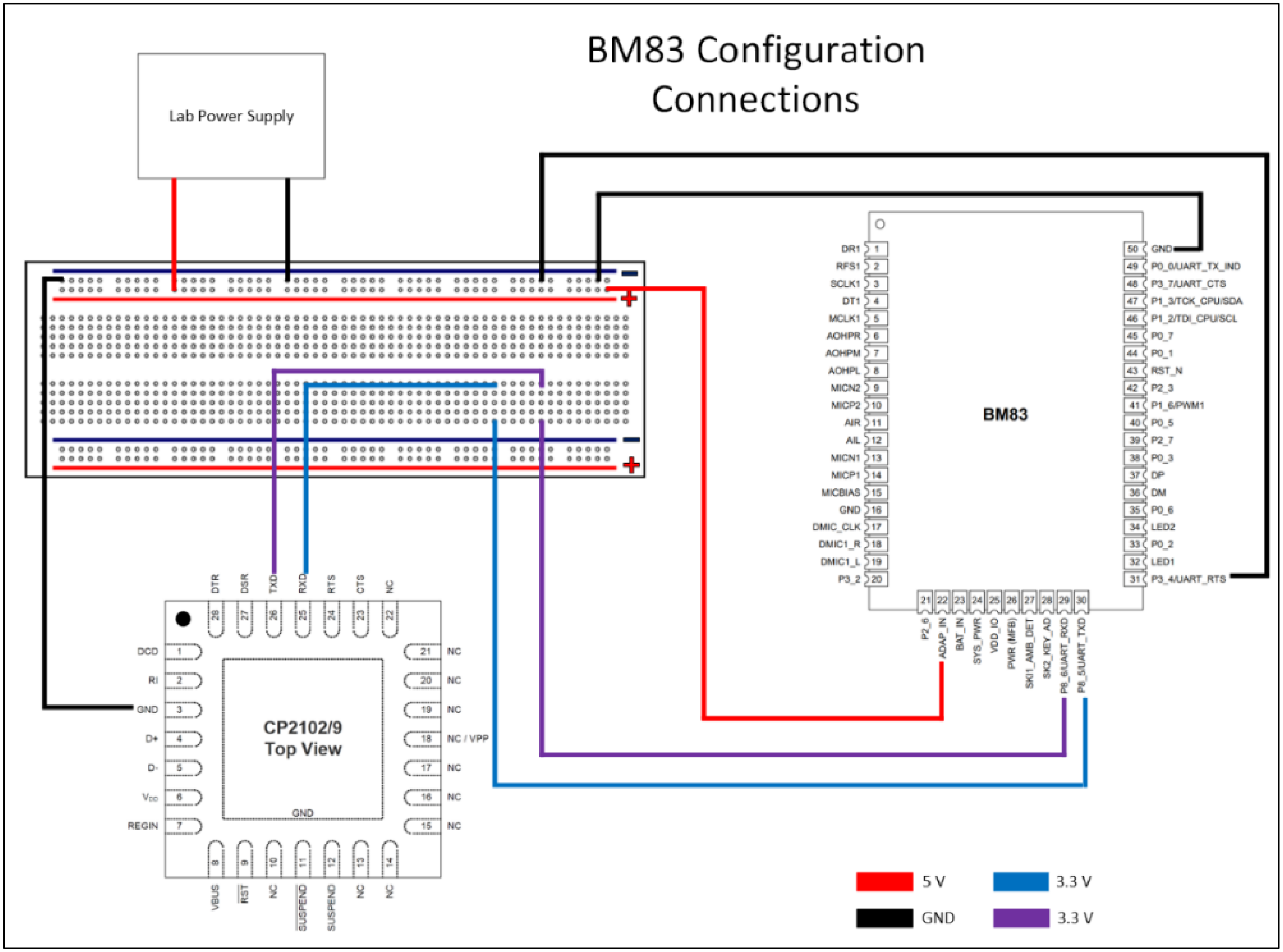
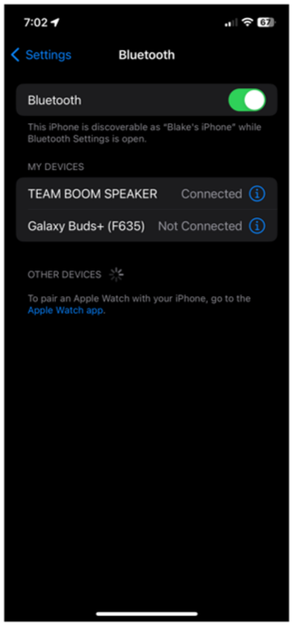


Figure 9 - BM83 Configuration Connections



Bluetooth

This iPhone is discoverable as "Blake's iPhone" while Bluetooth Settings is open.

MY DEVICES

TEAM BOOM SPEAKER Connected

Galaxy Buds+ (F635) Not Connected

OTHER DEVICES

To pair an Apple Watch with your iPhone, go to the Apple Watch app.

Figure 10 - Successful Pairing of a Bluetooth capable device, phone stays connected.

Regarding device programming, specifically the configuration of UART for the ATmega328PB microcontroller, our objective was to determine if we could send commands to the BM83 Bluetooth module and receive corresponding return events. Given the immense role that this communication link will hold for the rest of our project, establishing this connection became the primary goal for the device programming portion of this activity. To validate that the microcontroller was receiving return messages from the BM83, a few simple lines of code were implemented within a continuous while loop, activating an LED that would only turn on when the UART Rx(receive) buffer was full (Figure 11). The activation of the LED, thus confirming successful data reception by the ATmega from the BM83, would significantly mitigate the risks associated with device programming. This is because most of the remaining programming tasks would deal with functionality and features rather than communication. On the other hand, failure to activate the LED would indicate that some aspect of the UART configuration was incorrect, greatly increasing the risk associated with device programming.

The initial step in this segment of the prototype activity was to write the configuration module for the ATmega328 microcontroller. Within this module we defined some key parameters and functions such as the baud rate, in this case 115200 to match the BM83, as well as the initialization of UART and the BM83.

Within the UART initialization function, we wrote to the Control and Status Register B to enable the transmitter (TXEN0) and receiver (RXEN0) of USART0. Also in this function, data bits for communication were set to 8 by writing to Control and Status Register C. Finally, the UBRR0H and UBRROL were written to, and these registers would hold the binary value of the baud rate. For the BM83 initialization function to operate effectively, which sent a power on command and a version request command to the BM83, we first had to create three essential functions: the transmit function, receive function, and a send command function. The send command function would take in an OP code and parameter value, calculate the checksum value, then send the whole packet to the transmit function one byte at a time. The transmit function would first check to make sure the transmit buffer was empty and once it was empty data would be put in the buffer to be sent. Similarly, the receive function would wait until the buffer was full and once that happened it would return its contents. Putting this all together, the BM83 initialization function would send the OP code and parameter value of the power on command and version request command to the send command function. This function calculated the checksum of the packet and sent it to the transmit function. When the BM83 sent a response back to the ATmega, the receive function was called to receive the data as well as activate the LED.

The next step for this segment of the prototype activity involved establishing an electrical connection between the ATmega, BM83, and power supply. Initially we connected the Rx of the ATmega to the Tx of the BM83, and vice versa. However, we found that the ATmega had 5V logic levels and the BM83 had 3.3V logic levels. To solve this issue and ensure that the ATmega was only sending 3.3V signals, we set up a voltage divider with 0.5kΩ and 1.0kΩ resistors. This voltage divider established an output voltage of 3.3V that we connected to the Rx pin of the BM83. Next, we established a ground connection for both devices. We then established a power connection to the BM83 through a power supply. Finally, we were ready to flash our code to the microcontroller to see if a UART connection was established. After flashing our code, the LED turned on which showed that a connection was established, and messages were being received back by the ATmega (Figure 11).

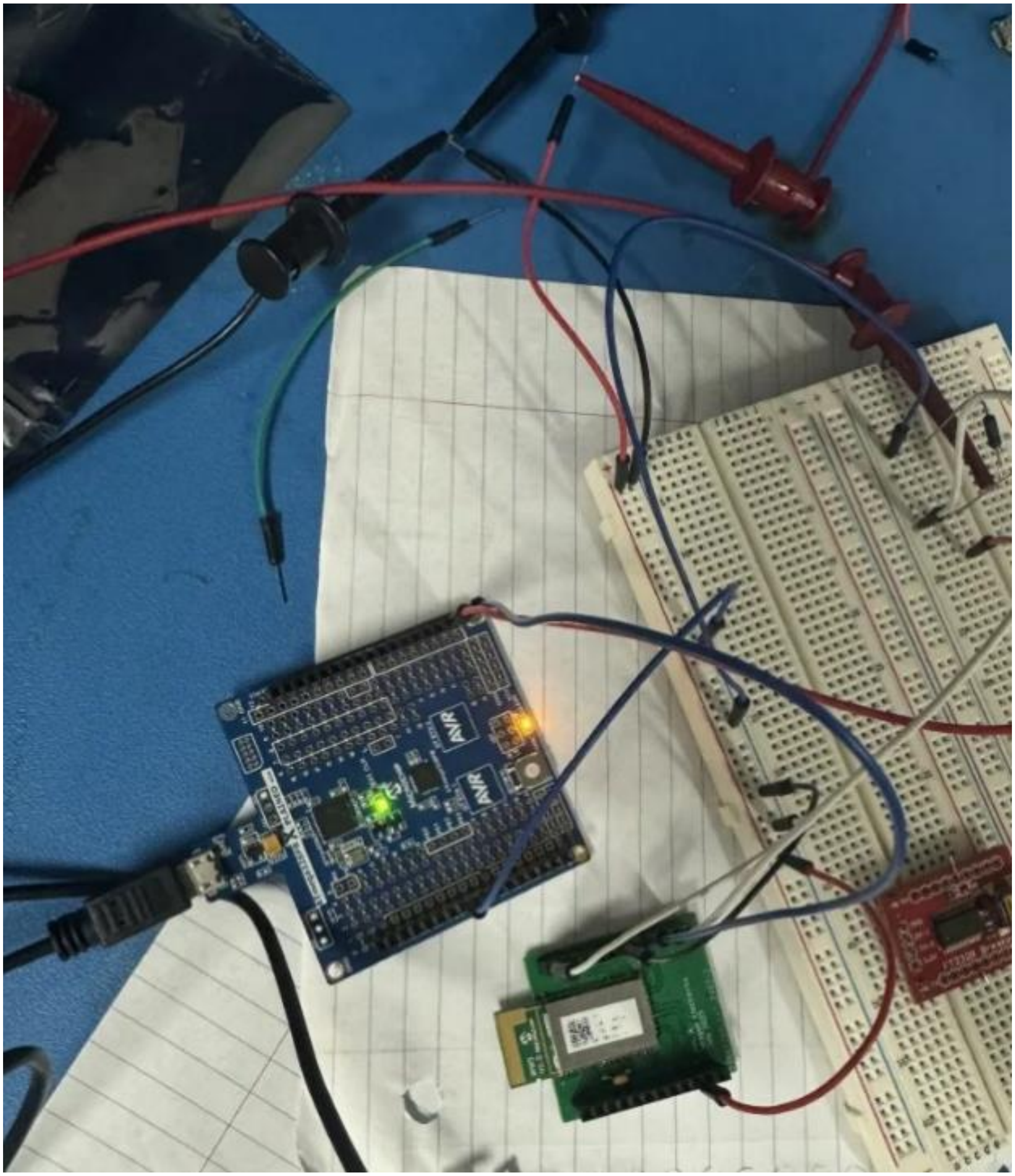


Figure 11 - The setup between the ATmega328 Xplained Mini Board and the BM83 Bluetooth Module. The bottom LED (orange light) is illuminated, indicating successful data reception and thus UART establishment.

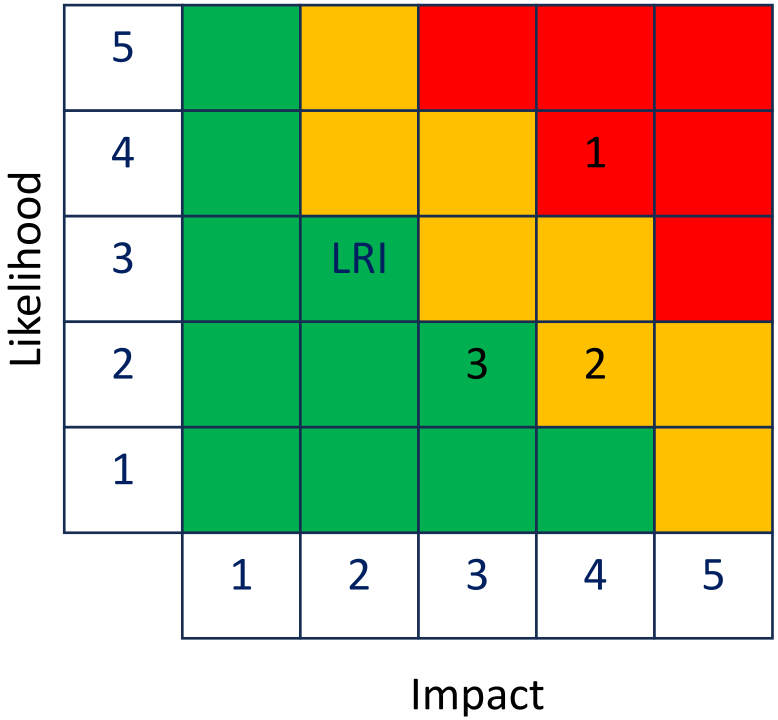
Final Risk Analysis

After having completed the prototyping activity, we believe we have reduced the risks of device programming and Bluetooth configuration. While there is still much work to be done in these areas, the progress made during this prototyping activity has demonstrated that we can communicate with the BM83 and configure it, communicate via UART between the ATmega 328 microcontroller and the BM83 Bluetooth module, and establish Bluetooth connections between a mobile device and the BM83 module. As a result, the following updated risk cube (Figure 12).

In Part B, our team plans to implement specific strategies to manage risks associated with Device Programming and Bluetooth Configuration. Our approach involves utilizing an oscilloscope to monitor the waveforms on pins 6 and 7 of the Bluetooth module following phone pairing while a song plays. By issuing volume up/down commands, we aim to analyze the waveform, verifying if the ATmega is successfully sending commands to the BM83.

While the prototyping activity was a success, the amplifier circuit remains to be designed and tested. The hardware team will work closely together to design and test schematics in LTSpice along with receiving feedback from meetings with Professor Dorr and other advisors. Once a successful build is modeled in LTSpice, we will build it in hardware to confirm the desired aspects. Finally, once the hardware tests pass, we will connect it to the output of the BM83 and to a speaker to confirm that the amplifier works as intended. As a final resort, if the amplifier circuit is unsuccessful, we will use an acquired amplifier module.

Updated Risk Cube



Likelihood

5	🟩	🟨	🟥	🟥	🟥
4	🟩	🟨	🟨	🟥 1	🟥
3	🟩	🟩 LRI	🟨	🟨	🟥
2	🟩	🟩	🟩 3	🟨 2	🟨
1	🟩	🟩	🟩	🟩	🟨
	1	2	3	4	5

Impact

1. Amplifier
2. Device Programming
3. Bluetooth Configuration
4. Low Risk Items (LRI)

Figure 12 - Updated Risk Cube

Key Integration Steps

The key integration steps are outlined in Figure 13 on the next page. There are two main paths defined by the hardware team's integration tests and the software team's integration tests.

On the hardware side, the amplifier is first built in a simulation software such as LTspice. In the simulation, a power gain of 16W will be confirmed as well as an analysis of total harmonic distortion (THD) to ensure acceptable audio quality. Next, the amplifier will be built in hardware in the lab. Here we will again test the gain, followed by a 1Khz test tone on a speaker to confirm audio output. Following this we will play an audio file to test the audio quality output of the amplifier. Parallel to this, the full system schematic will be designed so that when we are ready for breadboard integration, we will know exactly what connections to make.

On the software side, the first step that we must take would be to flash the BM83 with the latest firmware as well as update any settings through the SPK tool set. Next, UART configuration as well as user interface handling must be set up within the microcontroller by flashing our latest configuration code. After this step, we can now connect the ATmega328PB to the BM83 according to our wiring diagram. Once this is set up, we could use a simple LED on the transmission line to the Rx pin of the microcontroller to confirm that return signals are being sent back to the ATmega. After successful confirmation of UART communication, we will then shift our focus to testing user interface. We will properly connect a switch and button to its respective pins on the ATmega while also having an output pin connected to an LED which confirms that the microcontroller is reading the input changes properly. After UART communication and user interface testing is complete, we will test the final functionality of our project, audio amplification control for our class AB amplifier. We will do this by connecting a Bluetooth device to our speaker and sending a sine wave audio signal to verify that the amplifier is performing as designed.

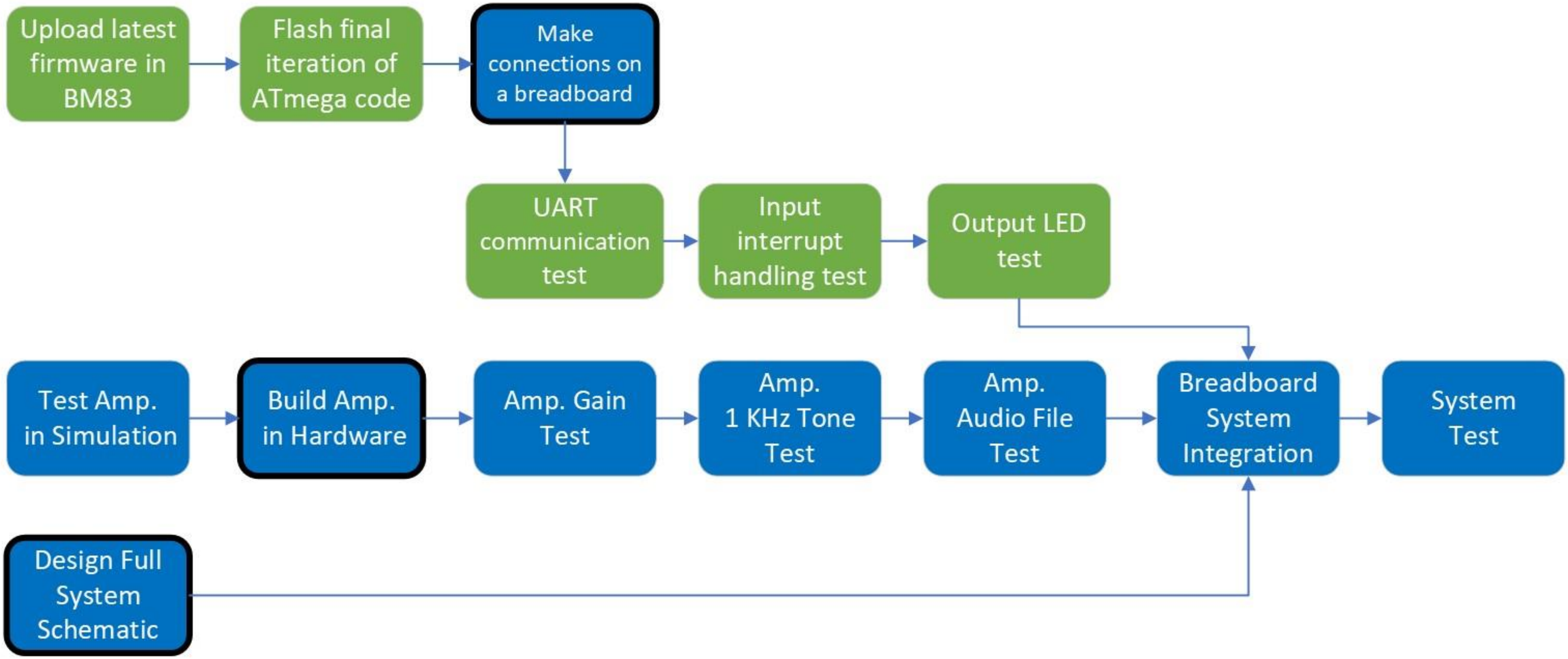

Upload latest firmware in BM83
Flash final iteration of ATmega code
Make connections on a breadboard
UART communication test
Input interrupt handling test
Output LED test
Test Amp. in Simulation
Build Amp. in Hardware
Amp. Gain Test
Amp. 1 KHz Tone Test
Amp. Audio File Test
Breadboard System Integration
System Test
Design Full System Schematic

Figure 13 - Key Integration Steps. Blue boxes indicate hardware integration. Green boxes indicate software. Black-outlined boxes indicate a setup prerequisite for the next step.

Work Breakdown Structure (WBS)

To ensure a systematic approach towards its development, we've structured our project using a Work Breakdown Structure (WBS). This hierarchical framework provides a comprehensive overview of the project's scope, dividing it into manageable levels and sub-levels. At Level 1, we define the overarching goal of creating a Bluetooth-powered speaker. From this main objective, we must first achieve four level 2 outcomes: hardware, firmware, procurement, and system integration. Figure 14 on the page below shows our full WBS.

The Level 2 firmware outcome is broken down into three level 3 outcomes: LED Status, to show speaker being powered on and indicating pairing status from the BM83, ADC Volume Sampling to control the volume and send over the op codes to increase and decrease volume in the BM83, and Controls Functionality to send commands to pair Bluetooth as well as power on the BM83.

The level 2 System integration involves bringing together all the tested components to ensure seamless operation of the Bluetooth speaker system. The outcome of system integration is subsystem testing, which is level 3, composed of four level 4 outcomes: system breadboard testing, microcontroller and Bluetooth module testing, PCB design, and amplifier circuit testing.

The level 2 hardware outcome consists of five components. The Bluetooth module, the microcontroller, the amplifier, the speaker, and the power supply. We identified that these components make up the key subsystems of the Bluetooth powered speaker. Four of these level 3 outcomes have level 4 outcomes that define their function in the system. A Bluetooth module to pair a Bluetooth device, the microcontroller to control the BM83, an amplifier to amplify the waveform from the BM83, and a speaker to play the audio output from the amplifier.

By breaking down our project into these detailed components and subcomponents, we ensure thorough planning and execution, leading to the successful development and integration of our Bluetooth-powered speaker system.

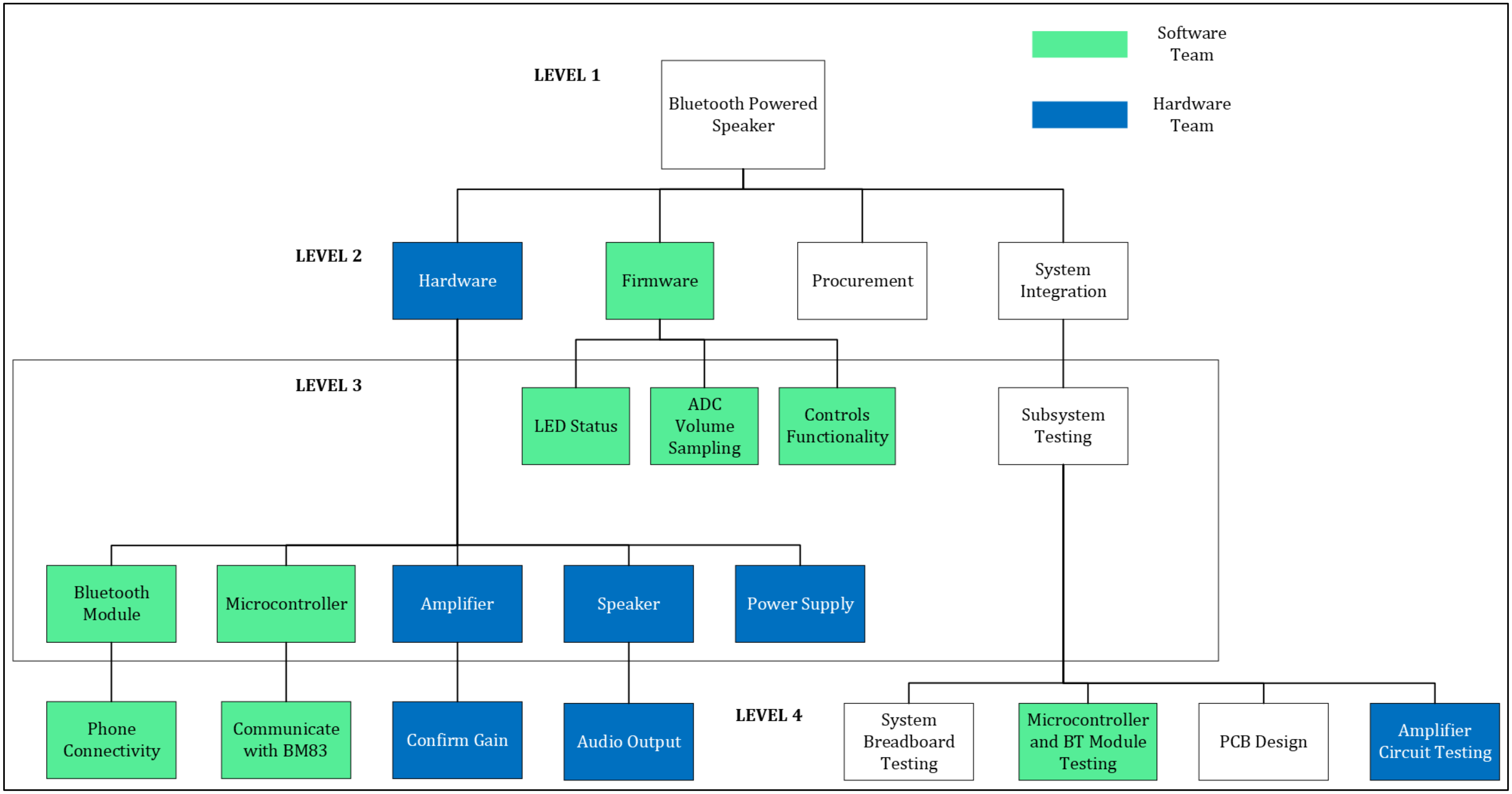


Figure 14 - WBS Chart

Task List & Schedule

Heading into the Fall semester, the team has identified the following milestones as shown in Table 2. The milestones have been listed in chronological order based on priority level. The Fall 2024 milestone have been chosen based on our Key Integration Steps.

For the schedule pertaining to the milestones and tasking. The link to our Team Gantt is provided for further inspection.

Team Boom Gantt Chart

Fall 2024 Milestones	Description
Amplifier Testing Complete	Amplifier design is complete, tested, and ready for integration
Software Configured	Software on microcontroller and Bluetooth module is complete, I/O tested
Functional Prototype (breadboard)	Fully functional speaker on breadboard, including all inputs and outputs
PCB Designed and Functional	PCB design has been finalized, received, and tested
Enclosure Built	Speaker enclosure is complete and ready for final assembly
Final Build	Speaker is fully assembled and tested

Table 2- Milestones

Cost & Financing

Item	Quantity	Price	Total
Power Jack	2	$0.69	$1.38
Microcontroller	3	$1.52	$4.56
Linear Voltage Regulator	4	$1.87	$7.48
Power Switch	2	$4.06	$8.12
Amplifier Parts	1	$20.00	$20.00
Pairing Button	2	$10.23	$20.46
Atmega XMINI Eval Board	2	$15.47	$30.94
Volume Potentiometer	2	$18.15	$36.30
LED 2	2	$18.49	$36.98
LED 1	2	$18.56	$37.12
Bluetooth Module	3	$13.18	$39.54
Power Supply	2	$20.71	$41.42
Enclosure Material	1	$46.00	$46.00
MISC Hardware	1	$50.00	$50.00
Speaker (8 Ohms)	2	$39.91	$79.82
Atmel Ice Programmer	1	$207.72	$207.72
TOTAL			$667.84

Table 3 - Cost Breakdown

*As mentioned in Risk 10, total cost will be split among members of Team Boom