BLE Demo Design Output

Table of Contents

[Overview 3](#_Toc194955730)

[Features 4](#_Toc194955731)

[Board Stackup and Pours 5](#_Toc194955732)

[STM32 Architecture and Toolchains 7](#_Toc194955733)

[STM32WB Microcontroller 7](#_Toc194955734)

[STM32 Developer Tools 7](#_Toc194955735)

[STM32WB Minimum Configuration 8](#_Toc194955736)

[Power Supply 8](#_Toc194955737)

[High-Speed External Clock Source 8](#_Toc194955738)

[Low-Speed External Clock Source 9](#_Toc194955739)

[Unused Pins 9](#_Toc194955740)

[Reset Pin 9](#_Toc194955741)

[Boot Mode and Boot0 Pin 9](#_Toc194955742)

[STM32 Peripheral Connections 11](#_Toc194955743)

[QSPI 11](#_Toc194955744)

[SPI 11](#_Toc194955745)

[I2C 11](#_Toc194955746)

[I2S 11](#_Toc194955747)

[UART 12](#_Toc194955748)

[RF Design 13](#_Toc194955749)

[Matching Network 13](#_Toc194955750)

[50Ω Matched RF Trace Geometry 14](#_Toc194955751)

[Low-Pass Filter 15](#_Toc194955752)

[Antenna Design 15](#_Toc194955753)

[Ceramic Chip Antenna 16](#_Toc194955754)

[DMA Utilization 17](#_Toc194955755)

[High-Capacity Non-Volatile Memory 18](#_Toc194955756)

[Serial Memory Flash 18](#_Toc194955757)

[microSD Card 19](#_Toc194955758)

[EEPROM 20](#_Toc194955759)

[JTAG 21](#_Toc194955760)

[Serial Wire Debug 21](#_Toc194955761)

[Serial Wire Output 22](#_Toc194955762)

[Power Configuration 23](#_Toc194955763)

[USB-C Bus and PD Controller 23](#_Toc194955764)

[DC-DC Regulator for Main Bus 24](#_Toc194955765)

[3.3V LDO for Main Bus 27](#_Toc194955766)

[3V Battery for Real-Time Circuit 28](#_Toc194955767)

[Audio Power Regulation 28](#_Toc194955768)

[Character Display 30](#_Toc194955769)

[Antenna Tuning 31](#_Toc194955770)

[Audio Output 32](#_Toc194955771)

[Audio Chain Considerations 32](#_Toc194955772)

[Integrated DAC and Amplifier 33](#_Toc194955773)

[LC Filter Output 34](#_Toc194955774)

[Analog Input 35](#_Toc194955775)

[Headphone Output 35](#_Toc194955776)

[Standby and Shutdown Modes 35](#_Toc194955777)

[Speaker Selection 36](#_Toc194955778)

[USB 38](#_Toc194955779)

[Current Monitoring 40](#_Toc194955780)

[Temperature Monitoring 42](#_Toc194955781)

[Bootloader from Serial Memory 43](#_Toc194955782)

[EMI/EMC 44](#_Toc194955783)

[Signal Integrity Practices 44](#_Toc194955784)

[DFM Check 47](#_Toc194955785)

[Excluded Features 48](#_Toc194955786)

[WiFi and TCP Client 48](#_Toc194955787)

# Overview

This document serves as the hardware design output document for the project. It contains details of the design choices to match the given specifications in the BLE Demo Specifications document. In the case where a specification was not met the cause and ways to address it are discussed.

The final circuit design features a microcontroller from the STM32WB line from STMicroelectronics, which supports several wireless protocols including Bluetooth® Low Energy, Zigbee®, and Thread® connectivity. A chip antenna along with a 50Ω feedline is included to enable wireless functionality.

An integrated Audio DAC and Class-D amplifier for a 1W, 8Ω speaker is included to allow for mono playback using the I2S interface supported by the STM32 microcontroller. To support playback, a QSPI Flash memory IC is available for sample storage and retrieval.

Bulk storage is provided by including an MicroSD housing. The SPI bus from the STM32 IC is configured to perform read/writes to the card.

The power input to the device is a USB-C connector with downstream DC-DC buck converters to generate both 3.6V and 5V rails. A USB Power Delivery IC is available to perform automatic negotiation of power levels to request the minimum/maximum voltage and current capabilities. A DIP switch allows for changing the capabilities requested upon a power reset.

A coin cell holder is included to provide power for the Real-Time Clock.

An onboard connector is available to interface to a small LCD character display. The connector and additional mounting holes are specifically laid out according to a reference LCD screen so that it can be mounted to the backside of the device during operation.

The board has headers for JTAG and SWD for the primary programming and debugging interface.

# Features

* STM32WBx wireless microcontroller in UFQFPN68 package
* 2.45 GHz ceramic chip antenna for Bluetooth Low Energy use
* USB Power Delivery controller
  + Adjustable voltage and current setpoints through DIP switches
  + 4 status LEDs
* Two DC-DC buck converters to generate required power levels
  + +3.6V rail for main system usage
  + +5V rail for audio circuit
* Two LDOs to filter DC-DC buck output
  + +3.3V rail for main digital IO and chip VDD levels
  + +1.8V rail for audio VDD levels
* Current consumption monitoring through sense resistor
* Mono audio output to 8 ohm speaker
  + I2S audio interface
  + 2-pin connector for external wired 1W speaker
* MicroSD card port
  + SPI mode
* Flash
  + QSPI bus for fast transfer
* EEPROM memory
  + I2C bus
* Ambient temperature sensor
  + Onboard with PCB cutouts for thermal isolation
* Boot mode switch
* Microcontroller reset switch
* LCD character display header
* 3V battery holder for STM32 battery component
* 10-pin JTAG/SWD connector
* 6-pin USART header for serial communication on/off board
* USB 2.0 supported
* 2 status LEDs

# Board Stack-up and Pours

The stack-up of the board is driven primarily by the RF circuit centered around the STM32WB microcontroller, which recommends a 4-layer board to accommodate impedance matching and ground pour requirements. The top layer is used to place components and most signals, particularly the more critical ones like the RF trace. Layers 2 and 3 are reserved for ground planes. The bottom layer is also used for signals.

Because of the large ground pour near the RF circuitry on the top layer, I have added additional ground pours to the top and bottom layers to open areas where I can periodically stitch around the edge of the pour to the inner ground layers. This is done to both balance the copper between the top and bottom layers, for manufacturing concerns, and avoid small ground pours or fingers that are not well connected to ground, which could pose EMI concerns.

It is noted by Zachariah Peterson Altium [article](https://resources.altium.com/p/copper-pour-and-stitching-do-you-need-them-pcb-layout), “Copper Pour and Via Stitching: Do You Need Them in a PCB Layout?” that thin copper fingers between signal traces could actually increase cross-talk if the finger is not grounded through vias periodically. However, there does not seem to be a strong consensus on whether to fill empty areas with ground (or power) pours as there are manufacturing concerns with unbalanced copper and there are claims that if used correctly it can decrease EMI issues. The Texas Instruments [video presentation](https://www.ti.com/video/6307563213112) and [companion PDF](https://www.ti.com/content/dam/videos/external-videos/en-us/9/3816841626001/6307563213112.mp4/subassets/crosstalk-on-pcb-layouts-presentation-quiz.pdf) “Crosstalk on PCB Layouts” by Art Kay provides a good overview of this, ultimately suggesting that haphazard use of ground pour will decrease circuit performance.

Another [article](https://resources.altium.com/p/shaky-ground-arguments-against-copper-pours) on Altium by Kella Knack, “ The Copper Ground Pours Problem in PCB Designs” agrees that pours generally are not advantageous but that there are some cases where pours on signal layers can be beneficial as they increase plane capacitance. In his [video presentation](https://youtu.be/0RyBCnowLsI?si=an82p3J144d2EuKn&t=2913) “Secrets of PCB Optimization”, Rick Hartley describes the issue of overplating metal in open areas when there is imbalance in metal distribution on the top layers. Having worked with metal plating myself during my time as an R&D engineer, I do recall that having significantly different features sizes or open voids could pose problems with quality.

Considering the points made in the referenced articles and presentations, I opted to ultimately try to balance the required RF top-layer ground pour with specific pours in large open areas that could be well connected to internal plane layers.

The stackup geometry is based on a stackup from JLCPCB, specifically their impedance-controlled JLC04161H-7628 Stackup. This one was chosen as it is a standard stackup for JLCPCB and allowed the RF trace to be matched to 50 ohms while having a decent width that was closer to 0402 components.

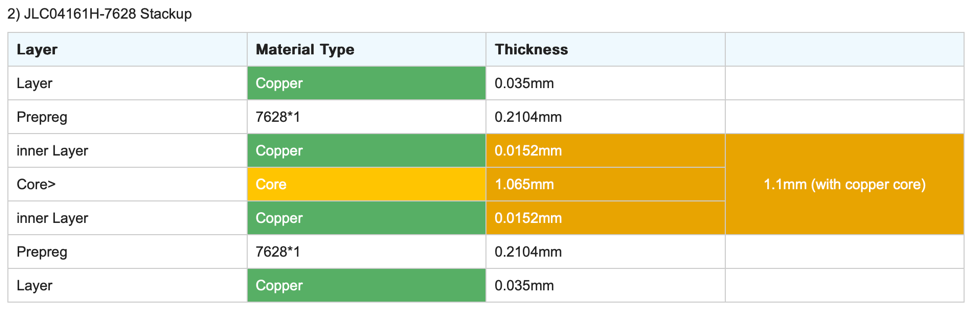


Figure 1: JCLPCB Stack-up

# STM32 Architecture and Toolchains

Since a main requirement of this project was to utilize a STM32 microcontroller with wireless capabilities, the lines of chips that could be selected from are the STM32 WBA, WB, and WB0 lines. These support short-range wireless protocols including BLE, Zigbee, and Thread. None of the STM32 chip lines support WiFi. This issue will be discussed shortly as it was an initial specification of the project that should have been supported if possible.

## STM32WB Microcontroller

The WB0 chipset is the simplest of the three lines and targets lightweight applications with long battery power lifetimes. It only supports Bluetooth. The WBA line is the most robust with higher output power and core operating frequency than the WB line. To balance cost and the feature set, the WB line is selected. Based on an initial estimate of the required number of pins for the planned peripheral usage the 68-pin variants are ideal. This limits the selection to the STM32WB55Rx variants as they are the only chips in the VFQFPN68 package. Other parameters that determine the variant include the flash memory size (C = 256 Kbytes, E = 512 Kbytes, G = 1 Mbyte) and temperature range (6 for range up to 85C, 7 for up to 105).

Based on these parameters, the [STM32WB55RCV6](https://www.mouser.com/ProductDetail/STMicroelectronics/STM32WB55RCV6?qs=vLWxofP3U2y%2F21YKkPUpfQ%3D%3D) is a good choice to get started on embedded development and is what is used as reference for the rest of the design. Mouser shows that it is readily available starting at $5.57 per chip.

## STM32 Developer Tools

Being that the STM32 devices are ARM Cortex-M core-based microcontrollers, several IDEs can be used to develop firmware for the chips. STMicroelectronics has their own IDE named [STM32CubeIDE](https://www.st.com/en/development-tools/stm32cubeide.html) that is built upon the Eclipse IDE to provide a C/C++ development platform with extra features such as peripheral configuration, code generation, code compilation and more. It seems to be held in good standings by its userbase, though you can find criticism of it on various forums as well. Coming from Microchip’s IDE, MPLABX, I am familiar with buggy experiences with library management and slow applications, so I am not that concerned and plan to utilize STM32CubeIDE to evaluate the vendor’s default tool. My limited experience of it while configuring the project has been satisfactory so far. One additional positive is that CubeIDE is supported on Windows, MacOS, and Linux.

To program the microcontroller, an ST Link V2 or V3 is needed in order to connect to the JTAG/SWD pins. A bootloader can be used to program the device using various methods.

# STM32WB Minimum Configuration

The STM32WB has requirements for a basic configuration regarding power, crystal clocks, RF layout, and pin connections. These are discussed in the following sections.

## Power Supply

In general, most components of the STM32WB should be sourced by a stable 3.3V supply. Section 6.1.6 of the [device datasheet](https://www.st.com/resource/en/datasheet/stm32wb55cc.pdf) shows the recommended decoupling capacitor values for each of the power pins. These capacitors are placed as close as to their respective pins as was reasonable. For ease of assembly and testing, resistor and capacitor components were limited to package sizes 0603 and up.

In general, passive component sizing was determined using the following ranges, which I have loosely based on component cost and typical maximum voltage ratings for a given capacitance.

* 0603: 1 pF to 100 nF
* 0805: 1 uF to 4.7 uF
* 1206: 10 uF to 22 uF
* 12010: >22uF

## High-Speed External Clock Source

The external clock source that drives the main High Speed External (HSE) clock for the STM32WB must be a 32 MHz crystal oscillator or sine or square wave. The crystal recommended by the device datasheet is the EXS00A-CS06654 from NDK from the NX2016SA series, which has an 8pF load capacitance. Additional 0603 capacitors are included to tune the board capacitance to the load capacitance, though the STM32 does have internal adjustable load capacitors that could prove sufficient. The required load capacitance can be calculated using , where and is an upper estimate. If C1 = C2, then . A lower estimate of yields that ­ *.*

The application note [AN5042](https://www.st.com/resource/en/application_note/an5042-how-to-calibrate-the-hse-clock-for-rf-applications-on-stm32-wireless-mcus-stmicroelectronics.pdf) “Precise HSE frequency and startup time tuning for STM32 wireless MCUs” provides specific information for the main HSE clock for the STM32WB line. It is noted that HSE configuration is very important for correct RF operation. The HSETUNE register sets the added load capacitance, ranging from 12-16 pf. Depending on the stray capacitance of the system, the internal load capacitance range of 12-16 pf could be sufficient.

The recommended crystal specifications by STM ([video source](https://www.youtube.com/watch?v=qNJR9vL1D0c)) are as follows:

|  |  |
| --- | --- |
| Parameter | Value |
| Load Capacitance | 8 pF |
| Frequency tolerance |  |
| Frequency vs temperature (ref. to +25C) |  |

I have opted to use a crystal from ECS before as I have used them before and they are readily available and are reasonably priced. The device most closely matching the given parameters at 32 MHz would be the [ECS-320-8-33B2-CTN-TR3](https://www.mouser.com/ProductDetail/ECS/ECS-320-8-33B2-CTN-TR3?qs=Jm2GQyTW%2Fbiv%2FbrQ%2F81iXQ%3D%3D), an 8 pF load capacitance crystal is in a 3.2mm x 2.5mm size. The device also matches the frequency tolerance recommendation and has better frequency stability at 20 PPM. It is cost-effective at around $0.42/u in low quantities and is available at Mouser or Digikey.

The clock signal can be output through the MCO pin to expose the master clock signal for timing analysis. This is preferable to measuring the clock lines directly as that would disturb the circuit.

## Low-Speed External Clock Source

The LSE clock takes in a 32.768 kHz clock signal and powers a few low-speed domains, including the Real-Time Clock (RTC). It uses the OSC32\_IN and OSC32\_OUT ports to connect directly to a crystal, resonator, or oscillator. Since there are no internal capacitors for load capacitance matching, external capacitors are included.

The application note [AN2867](https://www.st.com/content/ccc/resource/technical/document/application_note/c6/eb/5e/11/e3/69/43/eb/CD00221665.pdf/files/CD00221665.pdf/jcr:content/translations/en.CD00221665.pdf) “Guidelines for oscillator design STM8AF/AL/S and STM32 MCUs/MPUs” from STM has a large table listing recommended crystals for the LSE clock for various chip lines, including the STM32WB. One recommendation is the [ECS-.327-9-12R-TR](https://ecsxtal.com/products/crystals/surface-mount-crystals/ecs-327-9-12r-c-tr/), a 2-pin crystal in a 2.0mm x 1.2mm package. Its load capacitance is listed as 9pf. Using the previous equations, it is calculated that ­depending on the stray capacitance. An initial value of 12pF, corresponding to 3pF stray capacitance, is specified in the schematic.

## Unused Pins

AT0 and AT1 must be left floating, according to the datasheet. These are reserved for RF functions.

## Reset Pin

The STM32 has a reset pin (NRST) that is used to reset the controller. It should be connected to JTAG connector for use in programming/debugging. A small tactile switch (PTS526SK15SMTR2 LFS) has been added to provide a manual way to reset the controller. The switch provides a way to pull the NRST pin to ground, the reset state. A small capacitor is placed as close to the pin as possible for stability, per the datasheet. There is an internal pullup resistor to make the default state be the active state.

## Boot Mode and Boot0 Pin

The STM32 can boot into the main application or into a bootloader mode. This is controlled by the state of the BOOT0 pin and the BOOT1 bit. The BOOT0 pin has an optional resistor pad to tie it to ground, making the default application the user-loaded code. A small switch (DS04-254-2-01BK-SMT) is also included to allow for manual setting of the BOOT0 pin. The statue of the BOOT pin and bit are used to select booting from 1) user flash, 2) system memory, and 3) embedded SRAM. Note that user flash stores the main application and is used most of the time. System memory launches the bootloader so that firmware updates can be installed. The embedded SRAM is mainly used for debugging or special purposes.

The BOOT0 pin is used to boot from various peripherals such as USB, UART, I2C, and SPI.

# STM32 Peripheral Connections

## QSPI

QSPI is used to communicate with Flash memory controller. The STM resources on QSPI should be consulted in configuring the peripheral. The standout pinout using the four data lines, clock line, and chip select are routed to the sole QSPI device on the board using lines as short as possible. Length matching was used on the data lines so that all lines are roughly the same length.

There are three modes. These include 1) indirect, 2) status-polling, 3) memory-mapped. Single data rate (SDR) and dual data rate (DDR) are supported. When the Cortex M4 frequency is below 50 MHz, the Quad-SPI block can use the same clock frequency for the bus. If higher, the prescaler must ensure a clock division of 2.

Indirect mode is when all operations are performed through registers, like classical SPI. Status-polling mode is when there is automatic periodical read of the flash memory status registers and interrupts are triggered when there is a match for certain conditions. Memory-mapped mode is when external flash memory is read like it is internal for read operations.

QSPI commands contain up to 5 configurable phases. Each phase can be disabled/enabled, length adjusted, and a selectable number of lines. DMA support and interrupts flag can be used for data FIFO management. This will need to be investigated during the programming stage to determine the optimal setup.

## SPI

SPI is used to connect with the MicroSD card slot. A second pin is routed to be used as a custom chip select if an external device is connected.

## I2C

The I2C bus is used to communicate with several devices on the board. It can be configured in STM32Cube to optimize operation. The required pullups to 3.3V are included on the board. Device addresses are discussed in the relevant sections of this document. They are summarized here.

* USB PD IC: 00100 000 or 00100 001
* Audio IC: 0011 000
* EEPROM: 1010 000
* Temp sensor: 1001 000

## I2S

The STM32 uses the Serial Audio Interface (SAI) to connect to audio devices using I2S. STM32Cube has a few configuration settings and must be set before using the interface. The required signals including the channel select, master clock, and data line are mapped to their own pins. An additional master clock that is used to sync secondary devices is also routed for convenience.

## UART

The UART peripheral is routed to a standard 2.54mm connector for use in communication or debugging with external devices. The RTS and CTS signals were not routed due to not generally being required as well as more critical signals using their pins.

## RF Design

The RF element of this design is driven by the design and capabilities of the STM32WB. The device datasheet notes to refer to [AN5165](https://www.st.com/resource/en/application_note/an5165-how-to-develop-rf-hardware-using-stm32wb-microcontrollers-stmicroelectronics.pdf) “Development of RF hardware using STM32WB microcontrollers” to properly design the RF circuit. It notes that special care should be given for the layout of an RF board compared to a conventional circuit as it is more sensitive to design factors.

It is important to impedance match from both from the 1) antenna to the input of the chip and 2) chip output to the antenna. Poor matching introduces *lower sensitivity* and *lower signal amplitude* of the transmit signal. Maximum power is transferred when the internal resistance of the source (e.g. STM32WB) equals the resistance of the load (e.g. antenna). For a frequency-dependent signal, the load impedance must be the complex conjugate of the source impedance. Recall the complex conjugate has the same real part and an imaginary part with the same magnitude but opposite sign.

The STM32WB radio is based on a direct modulation of the carrier in Tx and uses a low IF ([intermediate frequency](https://www.allaboutcircuits.com/textbook/radio-frequency-analysis-design/selected-topics/the-benefits-of-an-intermediate-frequency-in-rf-systems/#:~:text=Summary,known%20as%20a%20heterodyne%20receiver.)) architecture in Rx mode. Recall that the [baseband frequency](https://en.wikipedia.org/wiki/Baseband) is a frequency that has not been modulated to higher frequencies, such as the output of a microphone for example. The carrier frequency is typically the higher frequency signal that carries the information in a modulated form, 2.45 Ghz in the case of BLE and WiFi. An intermediate frequency is somewhere between the baseband and carrier frequencies. The benefits of IF are more relevant for RX receivers. It is noted that a benefit of IF receivers is that they allow the design to be optimized for one frequency because it first shifts down the received spectrum down to a band centered on the intermediate frequency. A lower frequency in practice also is kinder on circuit elements such as transistors and passive components, which deviate from their low frequency models. Transmission line effects also have more of an impact on the system.

At this time, the output impedance of the RF driver is not readily available online.

### Matching Network

The design should utilize a Pi-style matching network to match the output RF1 trace to 50 ohms.

The AN5165 app note recommends to use a Pi matching network and a low-pass filter, they use the [DLF162500LT-5028A1](https://www.digikey.com/en/products/detail/tdk-corporation/DLF162500LT-5028A1/4321991). However, the device datasheet does note that Integrated Passive Devices (IPD) are available to match the output to 50Ω. However, due to the size of the component, 2-layer stackup application, and the recommended use of a placement system, I am opting to use a Pi matching network.

It is common, and recommended, to use 0402 sized passive components to avoid parasitic inductances. The Pi matching network has pads for 0402 capacitors/inductors. The pads are placed in-line with the RF trace and thermal reliefs are not used to maintain the ground plane.

The UQFN48 variant is stated have a driver impedance 43-16J and has the same IPD recommendation (MLPF-WB-01E3) as the UQFN68 variant. It seems reasonable to assume that if the same IPD is used for both devices that the driver impedance should be similar.

## 50Ω Matched RF Trace Geometry

Transmission lines can be implemented on external layers using microstrips and coplanar waveguides or in buried internal layers with striplines. The typical material used in RF design is FR-4. STM indicates that 2-layer PCBs can be used but they must be limited to 0.8 to 1.0 mm to make 50-ohm lines without overly large traces. While this would pose an interesting design challenge to tackle, it is most likely adding unnecessary complexity for a first-pass designing with RF features. It is generally recommended that 4 layers be the minimum to have ample room to route as well as have full ground planes for RF coupling. As mentioned earlier, the RF requirements and following calculations were used to determine the board stackup.

The STM32WB’s RF design guide shows their reference circuits using Coplanar Waveguide with Ground (CPWG) topology instead of the common microstrip topology. Either topology can work and has some advantages or disadvantages over the other. The [short video](https://www.youtube.com/watch?v=yK1qCWjPA-o) from Altium, “RF Design in the PCB: Transmission lines (coplanar)” describes some of the benefits for using CPWG which include containing electric fields in the conductor better, small RF signal traces, and isolation from other nearby signal traces. Another major factor is that CPWGs operate at higher frequencies better with lower signal losses. For these reasons I have opted to use a CPWG routing style for the RF trace.

Since the gap between the center trace and the nearby ground planes will be less than the width of the center trace, the separation distance imposes a lower limit on the geometry. Specifically, because a PCB manufacturer, such as JLCPCB, has a minimum trace separation of around 5 mils this can serve as the minimum. To keep away from working with bare minimums, I have selected a separation of 6 mil. Using KiCad’s CPWG calculate and the information from JLCPCB for copper thickness, dielectric thickness and permittivity yields that center conductor width of 13.1 mils should deliver a 50 ohm impedance.

It is noted that larger separation widths eventually lead to a microstrip structure, so I am presuming that smaller separation widths are better. In the application note [AN91445](https://www.infineon.com/dgdl/Infineon-AN91445_Antenna_Design_and_RF_Layout_Guidelines-ApplicationNotes-v09_00-EN.pdf?fileId=8ac78c8c7cdc391c017d073e054f6227) “Antenna Design and RF Layout Guidelines” from Cypress (Infineon) it is stated that the separation width should be less than the thickness of the dielectric. In this stackup, that would be 0.2mm (7.83 mils). Using a separation of 6 mils adheres to this guideline. In a [video](https://www.youtube.com/watch?v=Wh_322rlEME) discussing CPWGs, Zach Peterson with Altium notes that Altium’s default width is 5 mils so it does appear that smaller distances are better given that he discusses that smaller via fencing pitches and distance from the center conductor is better. His main points for increased ground separation seemed to be more about manufacturability.

It is usually recommended to fill unused area on the top RF layer with a ground plane and connect that to the underlying ground layer (e.g. layer 2) and connect them with vias at 1/10th the wavelength of the RF signal. This is done to ensure that the ground remains well-connected and prevent emissions between connection points. Given that the wavelength of a 2.45 GHz signal is 12.2 cm, a maximum distance between vias would be about 1.2 cm.

An additional note to usually consider is the width of the coplanar ground conductors. They need to look enough like a plane to perform as intended, so using traces is not idea. Width recommendations seem to range, but the recommendation given by Cypress in their AN91445 note is that the ground plane widths need to be larger than the gap between them. In this case, the separation is . The plane used on the top layer more than spans this distance.

## Low-Pass Filter

Per STM’s recommendation in the device datasheet, a LPF is included after the 50 ohm matching network. The part used in the application note AN5165 is the DLF162500LT-5028A1.

## Antenna Design

There are a few methods to add an antenna to this design. These include 1) a PCB antenna, 2) chip antenna, and 3) external wired antenna. The original specifications recommend a PCB antenna, specifically the PIFA-type. These come in a different variety, with the primary antenna looking like an F and another variant called a MIFA (meandered) that winds the antenna length to save space. These are both relatively common in IoT applications. I have personally seen the MIFA antenna be used in small BLE modules, like the [ATWINC1500](https://www.microchip.com/en-us/product/atwinc1500) by Microchip. I have also utilized a BLE module that uses a chip antenna in a past project with no noticeable performance issues, though range was not tested. In short, either antenna is most likely suitable for the final application. An external antenna is most likely the most robust and flexible, offers the best performance, and costs the most.

A short comparison between PCB antenna and chip ceramics points out a few characteristics that determine what is most appropriate for a design. Chips are both more compact but also most costly, though cost is difficult to quantify across the low-high volume scale. This is due to the cost of the ceramic component, any matching components, and differences in PCB footprint for the PCB antenna. However, in the larger discussion of cost, the design and testing for PCB antenna boards can greatly increase cost through multiple iterations and time spent because each board variation needs to be tested and design requires simulation to be done.

It is noted in the article [“Six Hidden Costs in a Wireless SoC”](https://www.silabs.com/documents/public/white-papers/six-hidden-costs-in-a-wireless-soc-design.pdf) by Silicon Labs that application notes providing antenna designs for their RF circuits often do not guarantee sufficient results even when followed to the letter. There are too many factors, such as board or SMT component properties, that can vary from the original test setup to guarantee the measured performance.

Chip ceramics should be less susceptible to environmental changes as well and overall easier to use in a design. Since this is a project with a very limited scope, with potentially only 1-2 iterations, I reason that getting a PCB that “works well enough” to be able to program is most important. To properly tune an antenna requires additional tools and will be a time and financial investment. I will discuss potential ways to test my antenna shortly.

Cost aside, it seems that this project could benefit from a PCB antenna as the board is expected to be large enough to support a MIFA antenna and it is not for a wearable application. It is possible that once I test the first iteration of the board with the chip antenna that I design a 2nd version with a PCB antenna.

### Ceramic Chip Antenna

Antennas can be compared using a few properties. The Voltage Standing Wave Ratio (VSWR) is a measure of how efficiently input power is transmit through the device to the output, specifically by calculating the ratio of the maximum to the minimum voltage . This ratio can also be derived from impedances. Typically, you want a lower ratio that is [around 2 or less](https://www.antenna-theory.com/definitions/vswr.php#:~:text=In%20general%2C%20if%20the%20VSWR%20is%20under%202%20the%20antenna%20match%20is%20considered%20very%20good%20and%20little%20would%20be%20gained%20by%20impedance%20matching.).

Another factor is antenna gain, typically given in dBi. It is the ratio between the gain of the antenna compared to that of an isotropic antenna, which radiates uniformly in all directions. Thus, a gain of 5 mean that the antenna will emit more power in a given direction compared to the isotropic antenna. Johanson Technology notes in their “Understanding Chip Antennas” [handbook](https://www.johansontechnology.com/docs/3763/antenna-handbook.pdf) that chip antenna gain ranges from -6 dBi to +1.0 dBi.

My final selection is the [AMCA72-2R470G-S1F-T4](https://abracon.com/parametric/antennas/AMCA72-2R470G-S1F-T) from Abracon because of its cost versus performance balance. As well, I find the antenna datasheet to be straightforward and is sufficient at explaining how best to use the antenna. It also suited my antenna placement style, which was edge-centered instead of the corner mount.

#### Antenna Matching Network

Similar to how the output of the STM32WB needs a matchning network to achieve 50 ohm impedance, the antenna also needs to be tuned. This is discussed in the device datasheet, where a simple PI filter is used. To account for this, I have place the 0402 footprint pads for the capacitors and inductors.

# DMA Utilization

DMA can be used to move data between two locations quickly without CPU involvement. Priority can be programmed to handle multiple requests across the configurable channels. DMA should be used to configure peripheral-to-memory, memory-to-peripheral, or peripheral-to-peripheral transfers. Transfers size should be flexible.

The STM32WB supports transfer size of 2^16. It allows access to flash memory, SRAM, APB, and AHB peripherals. There are 7 independently configurable channels. As nothing additional is required to enable DMA, it should be ready to configure in STM32Cube during the programming stage.

# High-Capacity Non-Volatile Memory

The board was specified to utilize a surface mount memory controller but also have a MicroSD housing to allow the use of external memory chips. This is accomplished through a couple memory devices including a serial EEPROM IC, Flash IC, and MicroSD card slot.

## Serial Memory Flash

When evaluating which memory type to use, several types such as Flash (NAND and NOR), FRAM, and MRAM were considered as these are common types of memory readily available. Considering that the Flash memory is intended to store data such as audio samples, device updates, etc etc it is appropriate to use the QSPI bus to enable fast transfer and gain experience with the STM32’s QSPI peripheral.

NAND Flash offers high memory density for the lowest cost in general. Commonly available sizes include up to 512 MB. For this application, a chip with in the 10-100 Mb range would suffice to enable a sufficiently large capacity for audio samples and any other assets.

NOR Flash has a large range of products on Mouser and on Digikey. Common sizes range up to 1 Gbit / 125 megabytes. Micron Technology, Infineon, and Macronix are primary manufacturers of the technology. Like NAND, NOR is erased on blocks such as 4 kbyte. NOR flash is suggested to be more reliable over time and common for permanent storage (compared to SD-card like removeable memory). Costs seems to range around $10-20 for such larger capacity chips.

FRAM and MRAM chips are both better for endurance, read/write times but are more expensive when comparing to similar Flash ICs. Additionally, FRAM ICs do not seem to come in larger MB-like sizes. Everspin Technologoies Inc. does make some MRAM chips that range up to 1 Gbit capacities but these start to cost upwards of $100. Needless to say, to make the best use out of such ICs would require a more specialized application dependent on memory performance.

Based on the available selection of chips on Mouser/Digikey, unit price, and balance of capacity and performance, a 16 MB NOR Flash chip, [MX25L12833FM2I-10G](https://www.mouser.com/ProductDetail/Macronix/MX25L12833FM2I-10G-TR?qs=mELouGlnn3dBGERdOcUbNQ%3D%3D), from Macronix is selected. It is relatively low-cost at $1.25 and can utilize QSPI. The device ships with the Quad Enable bit set to 0 as default, which puts it in SPI mode and the QSPI pins are linked to other functions such as write protect, DNU, and reset. When in QSPI mode, the Hardware Protection Mode (HPM) and RESET feature are disabled (only in the 8-pin package). The QSPI mode can be entered by writing the “EQIO” command over the SIO0 line. This means that the QSPI bus will have to be compatible with SPI operation to initialize the memory.

A pull-up resistor is placed on the CS line to ensure the device is deselected upon startup conditions.

## microSD Card

There are different types of microSD cards, including:

1. SDSC (Standard Capacity)
   1. Max 2GB
   2. FAT16
   3. Typical max transfer speed of 12.5 MB/s
2. SDHC (High Capacity)
   1. Max 32GB
   2. FAT32
   3. Bus speeds of 12.5 MB/s to 25 MB/s typical, up to 3938 MB/s depending on bus interfaces
   4. Interfaces: UHS-I, UHS-II, UHS-III, SD-Express
3. SDXC (Extended Capacity)
   1. Max 2TB
   2. FAT32 or exFAT
   3. Bus speeds of 12.5 MB/s to 25 MB/s typical, up to 3938 MB/s depending on bus interfaces
   4. Interfaces: UHS-I, UHS-II, UHS-III, SD-Express

A filesystem library will be needed to format data to the FAT-formatted card. The filesystem needs to support the different FAT formats listed above for the different types of cards. The STM32WB has access to FatFS, which does not need RTOS to run. The FatFS middleware should support FAT32.

In general, SD cards have 9 pins and micro. A standard SD card can operate in 3 modes: 1) SPI, 2) one-bit SD, and 3) Four-bit SD. Cards historically have used 3.3V since their introduction in 2000. Recent additions to the protocol allow cards to use 1.8V levels but may require the controller to initialize the card using 3.3V logic.

MicroSD pinout is standardized, listed below:

|  |  |  |
| --- | --- | --- |
| Pin Number | SD Name | SPI Name |
| 1 | DAT2 | X |
| 2 | CD / DAT3 | CS |
| 3 | CMD | DI |
| 4 | VDD | VDD |
| 5 | CLK | SCLK |
| 6 | VSS | VSS |
| 7 | DAT0 | DO |
| 8 | DAT1 | X |

The mode cannot be switched while power is applied. In this case, the SPI interface is easier to implement. QSPI is not a supported interface. Some additional note are that SD cards must be read/written to in sectors (512 bytes) and that SD cards may or may not have wear leveling available, it depends on the manufacturer.

ESD protection is generally recommended to protect the connected circuitry from any ESD events created by the SD card, as any protections built in to connect microcontrollers or ICS are typically only rated for device-level ESD like the charge device model (CDM) and the human body model (HBM), not system level ESD mostly because of the size requirements to do so. The [article](https://www.ti.com/lit/sg/sszb130d/sszb130d.pdf?ts=1740453764443&ref_url=https%253A%252F%252Fwww.google.com%252F) “System-Level ESD Protection Guide” by Texas Instruments gives a brief overview of ESD terms, designs, and important considerations. Note that the working voltage (also known as reverse stand-off voltage) is the voltage at which the diode will begin to conduct significant current when off. Thus, the working voltage should be at the maximum working voltage of your device, an example being a working voltage of 3.4V for a 3.3V logic rail. The clamp voltage is where the ESD device limits the voltage on the rail while sinking a certain amount of excess current. For digital interfaces such as USB, HDMI, or an SD card, unidirectional ESD devices are ideal as they have better protection against negative voltages. As well, the capacitance of ESD devices should be minimized to maintain signal integrity and considered especially on high-speed interfaces as they are connected in parallel. Texas Instruments notes that the fastest speed of the SD speed class, 90 Mbps, is not fast enough to generally cause problems.

The ESD devices selected are ESD321DYAR (single channel) and PESD3V3X4UHMYL (quad channel). The STM32WB55 has a maximum limit of 5.5V on the I/O pins, so the ESD devices should reach breakdown well below this. The PESD3V3X4UHM is designed for SD card protection with a very low clamping voltage of 3.7V. The ESD321 diodes have a working voltage of 3.6V and breakdown voltage of 4.5V (minimum). The clamping voltage is higher at 6.8V at 16A TLP and is listed as being suitable for USB 2.0 applications. As such, it will be used in both the USB 2.0 Data lines and on the SD card lines to simplify the BOM given that the specs should be suitable for both circuits.

The [article](https://resources.altium.com/p/how-to-design-microsd-power-circuits-without-destabilizing-on-board-voltage-supply) “How to Design Stable Micro SD Card Power Circuits in PCB Layouts” by Altium recommends that a capacitance value of 45 uF or greater will do on the power rail. It also states that a typical MicroSD consumes up to 100 mA at 3.3V. Based off the Altium recommendation, two ceramic caps rated for 22 uF and a 0.1 uF cap for decoupling purposes have been included.

## EEPROM

An additional EEPROM is included on the board to store runtime settings and provide flexibility in terms of what/how data is stored. The design has an SOIC-8 footprint for an EEPROM memory chip. The schematic and layout are designed specifically for the CAT24C256 from Onsemi, which is a 256 kbit EEPROM. It uses I2C to communicate with the controller. Depending on the 3 address pins, the I2C address is set by . The address is set to be fixed at 1010 000 by grounding the pins.

The write protect pin inhibits write operations when pulled high but does have an internal pull-down resistor, making the default state “writeable”. The WP pin is connected to the STM32 for direct control of the pin state. Other variants of the CAT24 device using the same pinout with lower cost and lower capacity can be used with no change in the design.

# JTAG

The STM32WB55 supports the serial wire JTAG debug port (SWJ-DP) that enables either a serial wire debug or JTAG probe to be connected to the target. Debug is performed using only 2 pins instead of the five required by the normal JTAG setup. The JTAG TMS and TCK pins are shared with SWDIO and SWCLK and a specific sequence on the TMS pin is used to switch between the two modes, per the notes in section 3.28.1 in the device datasheet.

The pins required for JTAG are typically:

* TDI (Test Data In)
* TDO (Test Data Out)
* TCK (Test Clock)
* TMS (Test Mode Select)
* TRST (Test Reset) optional.

The STM32WB uses the reduced pin count protocol of JTAG (IEEE 1149.7) that only uses two pins, a TMSC (test serial data) and TSC (test clock) pin. This is also known as cJTAG, for compact JTAG. This allows a star topology instead of requiring devices to be connected in series.

The pins on the STM32 controllered are indicated to be:

* Pin 54: JTMS\_SWDIO
* Pin 56: JTCK\_SWCLK

To interface with the JTAG interface, the programmer could use the ST-LINK/V2 by STM which also supports SWD. The ISOL version has increased isolation between the computer and PCB but doesn’t seem to be required in most cases. It comes with a connector for the standard 20 pin JTAG interface. The ST-LINK V3 has a MIPI10, STDC14, and MIPI20 connector that sport a 1.27 mm pitch. To maintain flexibility in terms of how to connect to the JTAG pins, a 10-pin 1.27mm connector is provided as the main connector. However, I have also routed out the signals to a basic 2.54mm pitch DuPont header. This 2nd output is wired directly to the signal traces through a resistor network to route the signals out from the MIPI-10 connector. If not used, a resistor network can be used to short the traces to ground if there are performance issues.

# Serial Wire Debug

Serial Wire Debug is a similar interface to the JTAG protocol that is supported on ARM microcontrollers that only uses 2 wires, SWDIO and SWCLK. In the case of the STM32, both the data and clock signals for the 2-wire JTAG protocol and SWD are available on the same pin so that either protocol can be used using the same connector. It allows for programming and debugging of the device.

# Serial Wire Output

A third debugging feature that is frequently routed into the JTAG/SWD 10-pin connector is the Serial Wire Output (SWO) pin. The SWO pin can be used to gain access to data routed from printf style debug statements in code. The printf function ultimately calls a “print character” function that writes the data to the selected output such as SWO or UART. SWO can be left floating if not used and most debuggers should still work as intended. To provide maximum flexibility, the SWO pin has been routed to the MIPI-10 but with a few caveats. The SWO happens to share the pin with the SAI SCK pin, so unfortunately both functions will not be available at the same time. To deal with this, I have footprint for a jumper that connects the SCK/SWO pin from the STM32 to the 10-pin JTAG/SWD header. This allows for the SCK/SWO pin to be optionally connected to the SWO pin on the JTAG/SWD header in a debug stage but otherwise be connected solely to the audio IC during normal operation. This will limit what can be done with the audio circuit during debugging, but I suspect this is acceptable. There is also a UART connector that can be used solely for IO with external devices, such as a computer terminal.

# Power Configuration

The power requirements of the system are primarily driven by the RF components of the board since they are most sensitive. The STM32WB is designed for low-power applications running off coin-cell batteries, drawing up to 7.8 mA when the highest power level (+6 dBm). It has an onboard SMPS that can be optionally used given a high enough VDD. It is noted that the integrated LDOs that feed the system, including the RF and digital circuits, can be sourced either from the integrated SMPS or an external source. However, the other subsystems are fed directly from their source.

There are three embedded regulators that supply most of the digital and RF circuits, the three being the main regulator (MR), low-power regulator (LPR), and the RF regulator (RFR). The MR is used in the run and sleep modes as well as stop 0 mode, the LPR is used in a low-power run/sleep modes as well as stop 1 and stop 2 modes. It is used to power the SRAM2a in standby as well. The RFR supplies the RF analog part. All regulators are powered down in standby and shutdown modes and the outputs go to a high impedance mode. By default, the microcontroller is in run mode, range 1, after turning on.

In general, a 3.3V rail is sufficient to power the STM32. The audio circuit has its own power requirements of 1.8V and 5V with moderate levels of current, expected to be under 0.5A.

To generate these rails, USB Power Delivery is used to provide some flexibility in the input power. Levels such as 5V, 9V, 15V, and 20V can be requested with maximum current capabilities ranging from 3-5A. Since the board will not run off a battery, the USB PD controller needs to support “dead battery” mode which allows the negotiated power to be delivered downstream upon startup.

## USB-C Bus and PD Controller

The TPS25730 is a simple and decent option with good documentation as well as various features. It supports dead battery mode and uses a resistor divider to set the I2C address as well as voltage min, max, and current bounds. It also contains an integrated protection FET for VBUS. The TPS is powered by VBUS or VIN\_3V3, the latter being a typical supply when there is already power in the downstream circuit from another power input. Upon registering the dead battery condition, VBUS is used and VIN\_3V3 remains disabled until cleared using I2C. Thus, VIN\_3V3 could be hooked up to receive power from a 3.3V bus when it is available.

Four pins (ADCINx) set the voltage and current capabilities of the PDO to be requested. It is also used to set the I2C address. Since the device will be operating in dead battery mode and it appears the config lines are sampled only upon boot, an 8 position DIP switch (4-1825059-2 from TE Connectviity) is included to select two primary settings. The configuration involves sampling the voltage over a divider so the switch will select which resistor is used in the lower leg of the divider. The evaluation module [guide](https://www.ti.com/lit/ug/slvucp4/slvucp4.pdf?ts=1741359241704&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS25730) shows a similar resistor divider configuration setup and lists 100k as the upper leg resistor with a selectable resistor as the lower leg, with values recorded and verified by their DIV value when comparing the decode and DIV values in Table 8-1 of the TPS25730 datasheet. The decode value of 7 does not need a pulldown resistor and can be tired directly to LDO\_3V3, so if a decode value of 7 is needed then no pulldown resistor should be selected / populated.

|  |  |  |
| --- | --- | --- |
| Decoded Value | Resistor (ohm) | DIV Value |
| 0 | 1.0k | 0.0099 |
| 1 | 5.10k | 0.0485 |
| 2 | 12.1k | 0.1079 |
| 3 | 23.2k | 0.1883 |
| 4 | 43.2k | 0.3017 |
| 5 | 115k | 0.5349 |
| 6 | 412k | 0.8047 |
| 7 | 2M | 0.9524 |

For a PDO to be selected, it must fit within the bounds of the setpoints specified by the TPS25730. If a profile does not match, VBUS will remain at 5V and limited current can be drawn; 500 mA for USB 2 and 900 mA for USB 3. While 5V / 900 mA might be suitable power level for device operation, I assume 500 mA will not be with all features like BLE, audio, and character display. Thus, it is important that a PDO is selected. The following table describes the options selectable through the switch. Note that ADCIN3/4 are used together to define the current capability and are linked. With 4 total combinations, I will pick the two edge cases for low-current and high-current modes.

|  |  |  |
| --- | --- | --- |
| Name | Option A | Option B |
| Min Voltage (ADCIN1) | 0 | 5V | 1 | 9V |
| Max Voltage (ADCIN2) | 1 | 9V | 3 | 12V |
| Sink Current (ADCIN3/4) | IN3: 1 | 1Anom  IN4: 0 | 1.5Amax | IN3: 3 | 3Anom  IN4: 1 | 3Amax |

Note that the I2C address does change depending on the options selected for ADCIN1/2, so the STM32 will need to account for this by establishing which address the device is using upon startup.

At a minimum, the VBUS should see 5V minus a small factor from the sense resistor using in current sensing.

VBUS has an ESD protection diode, the TVS1800DRV, added for safety. It is [recommended](https://www.ti.com/lit/an/slvaf82b/slvaf82b.pdf?ts=1714998779059) by TI for a PD voltage of 15V.

## DC-DC Regulator for Main Bus

Since the voltage output from the USB PD port can be negotiated, and range from 5-20V, the system should be prepared to regulate the VBUS down to a level suitable for an LDO to feed the rest of the circuitry. This is due to the design feature that a higher input voltage may be negotiated so that other components or expansions can have access to higher power, such as in the case where a motor expansion board is to be added to the board. This is accomplished through a DC-DC buck converter. The logic bus should not require high levels of current but does need to provide a stable output. Thus, a 3A rated buck regulator could be sufficient.

Since 3.3V is the main system rail, it should be stable going into the system. Switching converters typically add switching noise to the system, so an LDO is placed after switching converter to help filter that. The chosen buck converter is the TPS62933 based on its cost, low external part requirement, and efficiency. The -F version has continuous current mode to keep low ripple in all conditions. The -O version operates out of the noise band. It has a max duty cycle of 98%, meaning it has very low dropout voltage. To achieve low frequency efficiency, the base version operates in either 1) continuous conduction mode (CCM), 2) discontinuous mode (DCM), or 3) pulse frequency mode (PFM). PFM mode achieves higher efficiencies at light loads by adjusting the switching frequency.

Because the different versions use the 7th pin either for a Soft Start (SS) or Power Good (PG) pin, I have included footprints to accommodate either architecture. With the SS pin, the appropriate capacitor can be placed with a jumper to ground. With the PG pin on the -P version, an 0603 diode and current limiting resistor can be placed for visual indication.

The output voltage is set by the voltage divider and . The is set to 10kohm resistor, . Solving for the top resistor yields the resistance needed:

The operating frequency of the device can be set through the resistor connected to the RT pin. If left floating, the switching frequency defaults to 500 kHz. If connected to ground, the frequency is set to 1200 kHz. For resistance values of , the switching frequency can range from 200 kHz to 2.2 MHz. The equation that governs the relationship is given by . Choosing a resistor value of 50 kOhm sets the frequency to 433 kHz. There are pros and cons to higher vs lower frequencies which come down to ripple size, response time of the system, size of the inductor, as well as ability of the LDO to filter ripple.

Calculating the recommended inductor size is done by the following equations given in the device datasheet. It is noted that the coefficient K is defined by and should range from 20 to 60%. A value of 40% is recommended and represents the ratio of maximum output current of the device to the inductor ripple.

In order to shift as much as the power conversion into the DC-DC converter and out of the LDO, an output voltage of 3.6V is targeted. The input voltage is expected to range from 5-12V. Solving for various values of L gives the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| **VIN (V)** | **IOUT MAX (A)** | **K** | **L (uH)** |
| 5 | 1.5 | 0.3 | 5.17 |
| 5 | 1.5 | 0.4 | 3.88 |
| 5 | 2 | 0.3 | 3.88 |
| 5 | 2 | 0.4 | 2.91 |
| 5 | 3 | 0.3 | 2.59 |
| 5 | 3 | 0.4 | 1.94 |
| 9 | 1.5 | 0.3 | 11.09 |
| 9 | 1.5 | 0.4 | 8.31 |
| 9 | 2 | 0.3 | 8.31 |
| 9 | 2 | 0.4 | 6.24 |
| 9 | 3 | 0.3 | 5.54 |
| 9 | 3 | 0.4 | 4.16 |
| 12 | 1.5 | 0.3 | 12.93 |
| 12 | 1.5 | 0.4 | 9.70 |
| 12 | 2 | 0.3 | 9.70 |
| 12 | 2 | 0.4 | 7.27 |
| 12 | 3 | 0.3 | 6.47 |
| 12 | 3 | 0.4 | 4.85 |

This shows that the optimal inductor value is different for various operating conditions. For the case where the device is drawing 9V power with 1.5A, an 6.8 uH or 8.2 uH inductor could be ideal. Essentially for all 5V scenarios, a 5.6 uH is sufficient. Higher inductor values increase cost and DCR but generally reduce ripple current.

The maximum current limit through the high-side MOSFET is from the datasheet. The calculated peak current is calculated as follow:

The current is calculated from the equation:

It follows that the inductor should have a saturation current of around 8-10 A, and RMS current rating of 5-6.5A. A shielded inductor will prevent less EMI and is preferable given the RF component of the board. The default inductor selected is the SRP6060FA-5R6M, which matches the requirements. Other variants come in 6.8 and 8.2 uH to support other power scenarios.

Regarding power dissipation, the buck should almost always be operating around 90% efficiency depending on the input and output voltage. With a 2A output, there should be a dissipation of around . This dissipated power leads to a junction temperature of . If a 0.5A load is drawn by the system, the heat rise should be around 11C. This is acceptable.

#### Notes about inductor and load currents

In fixed frequency mode and in the high-side switch on time, the SW pin voltage swings up to approximately and the inductor current increases with slope . Note that this does not depend on the output current. When the high-side switch is turned off and the LS switch is turned on, the inductor current discharges through the low-side switch with a slope of . Note again that the slope does not depend on the output current. When the current output is always above 0, the mode is CCM. Thus, a larger inductor will decrease the slopes of both phases.

When the load current, e.g. 300 mA, is lower than half of the peak-to-peak inductor current in CCM, the devices enter DCM. At an even lighter load, pulse frequency modification is used to maintain high-efficiency. When either the minimum high-side on time, or the minimum peak inductor current is reached, the switching frequency decreases to maintain regulation. It is noted that the current comparator in the TPS catches the peak inductor current only, the average load current entering PFM varies with the application and external LC filters. The high-side MOSFET is turned on in bursts to provide energy to the load. When zero current crossing is detected, the low-side MOSFET is turned off. The lowest selectable frequency possible is 200 kHz. The minimum switching frequency, presumably when in light current mode, is 30 kHz.

Under high load currents near the maximum, such as 2A or 3A, the switching frequency can be increased to 1 MHz or so to accommodate the load change and higher input voltages.

## 3.3V LDO for Main Bus

The STM32 needs a stable 3.3V reference for most peripherals. The internal SMPS can be used to feed the main and RF regulators but not the low power regulator, wakeup domain, LCD, or analog domain, or USB domain. Thus, a regulator is required to adjust the output from the USB port. A DC-DC buck boost converter is used to provide a fixed 3.6V output such. It is noted that the STM32 indicates it uses very low power to run, perhaps 20 mA or less, by looking at the Power Consumption Calculator (PCC) tool in the STM32Cube IDE.

It is estimated that major current draws could come from the character display backlight (100 mA) and MicroSD card (30-100mA). Essentially, the current requirements of the 3.3V bus are not that high and this allows for some flexibility. The LDO chosen to is the [**LD39200PU33R**](https://www.digikey.com/en/products/detail/stmicroelectronics/LD39200PU33R/5131797), which can output 2A with a maximum of 0.25V dropout and a PSSR of around 40-50 at 500 kHz. It is also a very cost-effective chip at around $1/unit. While it does limit the current supply for the 3.3V bus, its cost effectiveness and other performance qualities make it the desired choice for this design. With a max 0.25V dropout, a suitable input voltage should be or greater. This provides some headroom for the LDO and some room for the buck converter to scale down. This led to the selection of 3.6V as the intermediate bus.

## 3V Battery for Real-Time Circuit

When the 3.3V rail is not available to the STM32, VBAT operation is automatically activated. When the microcontroller is supplied only from VBAT, as it would be when USB-C is disconnected, external interrupts and RTC alarm/events do not exit it from VBAT operation. The VBAT pin should be connected to the battery. The valid voltage range is from . In section 6.3.2 of the datasheet it is recommended that VBAT fall within . Current consumption data is given for 1.8V, 2.4V, 3.0V, and 3.6V. A very common battery type is the CR2032, which is a 3V coin-cell battery. It is nominally 3.2mm thick and 20mm in diameter.

The [BAT-HLD-001](https://www.digikey.com/en/products/detail/te-connectivity-linx/BAT-HLD-001-THM/3044009?_gl=1*10no8pn*_up*MQ..&gclid=Cj0KCQjwztOwBhD7ARIsAPDKnkAVl-omgrW-V7a-AVtoaE2QZEPQHywOrfch-0X9evkTW-sQ3L3zPZgaApR6EALw_wcB) from TE Connectivity provides a cost-effective and simple battery holder.

## Audio Power Regulation

The audio component of this design requires at least one additional rail. The design of the audio component is discussed later in this document but, to continue discussion of power configuration, the audio chain requires at least 1.8V for analog and digital core components. To provide more power to the speaker, an additional rail that ranges from 3.6 – 5V also could be beneficial and potential configurations will be discussed.

The 1.8V could be generated relatively easily from the 3.6V rail using an LDO. Since the 3.3V rail feeding the RF circuit is generated from an LDO, this could be acceptable. However, since the speaker rail would benefit from a higher voltage, it could make sense to have a 2nd DC-DC switching regulator that regulates the USB PD voltage to 5V. The 1.8V rail could then be generated from this 2nd rail.

For the 1.8V rail, the [LDL112PV18R](https://www.mouser.com/ProductDetail/STMicroelectronics/LDL112PV18R?qs=Ok1pvOkw6%2FrJNgvWv3vE3g%3D%3D) LDO can output 1.8V from an input range of 1.6 to 5.5V with a max current of 1A. It has a low dropout voltage of 200 mV.

The 5V rail could be generated using a buck-boost converter or simply a buck converter with 100% duty cycle to allow for the default case of 5V coming from the USB. Since buck-boost converters are generally more costly compared to buck converters and it is not a strict requirement that a precise 5V rail be generated, as only the speaker power will see the 5V directly, a buck converter will be used.

The construction of the 5V rail must consider the 5V, 9V, 15V, and 20V cases. If the buck converter is rated for the higher voltages, generally there should not be any problems designing around the parameters. As the speaker may enter low-power states, light load efficiency is important. The ripple factor also will presumably be a significant factor, as audio reference levels should be noiseless and stable. The 5V input case could pose some issues with a dropout voltage attributed to the maximum duty cycle supported by a buck converter.

Most buck converters are limited by their maximum duty cycle relating to their architecture. In general, there needs to be switching to make sure the bootstrap capacitor remains charged. This ultimately limits the maximum duty cycle, limiting the upper limit of the output voltage given a certain input voltage. While some devices can support up to a 98% duty cycle, like the TPS62933 used for the 3.6V rail generation, this would cause a rail. While this is pretty close to 5V and acceptable given that it should only affect the maximum speaker output, I want to utilize a buck converter that supports 100% duty cycle, otherwise known as low dropout mode (LDO mode) by some manufacturers.

One option is the AP63300 (or AP63301), which supports LDO mode, has an efficiency of around 90% for 5V output with 12V input. The AP63301 does not have great light load efficiency because it operates in PWM mode but does sport a better ripple output. Both devices have a fixed switching frequency of 500 kHz.

The AP6330x has an undervoltage lockout voltage of 3.06V and disables the output if the input falls below that. It also has overvoltage protection, overcurrent protection, and thermal shutdown features.

The output voltage is set using a voltage divider, governed by the relationship . Diodes Incorporated provides recommended values for common outputs. For 5V output, and . Additionally, an inductor of 6.8 uH is recommended. and capacitors, , and .

The equation given to determine inductor value is . If using the 9V profile and keeping the inductor ripple current smaller by setting it to be 30% of the maximum load current, 3A, this yields that . The datasheet notes that an inductor in the 2.2 uH to 10 uH range with a DC current rating at least 35% higher than the max load current should be selected. The DC resistance should be minimized, ideally less than . A larger inductor improves efficiency under lighter loads, though this typically means internal resistance increases which leads to increase power loss. The inductor’s saturation current rating should be selected according to the peak current, calculated using . For load current of 3A, . If there is a set 1A max current, .

# Character Display

The ability to power and write to a small character display is made available by exposing the pins needed. The display chosen as a reference is the [C204A-FTW-LW63](https://focuslcds.com/product/c204a-ftw-lw63/) from FocusLCDs, which is a 20x4 character display with backlight. It uses a 3.3V input and a 4 or 8-bit interface, which is very common for character displays. It is roughly 3.9 by 2.4 inches. The pinout is given in the datasheet and uses a 1 x 16 pin header for all data and power pins. The pitch given is 2.54 mm.

In 4-bit mode, only 4 lines are needed to set the character display. However, data is then sent in nibbles then so two transactions are required to write data or a command. The FocusLCD [website article](https://focuslcds.com/journals/break-down-of-a-parallel-interface/?srsltid=AfmBOopbbiziXuHUgvugt_7WjOh3LS8rwjkJNCKrJj8tgkYwGZSwBcBg) links the [embeddedcraft.org](http://www.embeddedcraft.org/lcd4bit.html.) tutorial on 4-bit operation. In general, it is indicated that to write two nibbles the data must be set and followed with a pulse on the enable line a total of two times.

The pinout below is taken from the datasheet.

A screenshot of a computer

Description automatically generated

The V0 pin is used to modify the contrast of the screen. The [application note](https://focuslcds.com/journals/application-notes/adjusting-the-contrast-of-an-lcd-module/) by FocusLCDs indicates that V0 should be wired to a voltage divider or potentiometer of 10k - 20kOhms. The [23BR10KLFTR](https://www.digikey.com/en/products/detail/tt-electronics-bi/23BR10KLFTR/2408592?s=N4IgTCBcDa4MwCEBKBGADAaQDIDEAqSIAugL5A) by TT Electronics is selected for its small footprint and low price. Additionally, two pads for a voltage divider are included to allow for fixing of the voltage if that is preferred on some units. A third option considered, but ultimately not used is that of a digital potentiometer with an op amp which could allow for control of the contrast through the STM32. This could be relatively straightforward to add in a future revision.

# Antenna Tuning

The device should be able to be tuned using the onboard U.FL connector connected to the 50 ohm RF line.

# Audio Output

Audio was not part of the initial specification but has been added as a nice value add to allow for development with audio-related functions and allow for a fun project utilizing flash memory to store samples, the I2S interface for audio transfer, potential transfer of source files over BLE, and the DMA peripheral for handling the data.

Enabling audio output can be done in various ways with various degrees of complexity. At a most basic level, a piezo buzzer can be used with a PWM input to generate sound. A more middle-ground solution would be to utilize the STM to handle some audio processing with custom audio samples and generate the analog signal directly to an amplifier to a speaker. The most complex solution would be to have an audio codec chip on board, which handles a wide array of analog inputs/outputs as well as settings and controls relating to mixing, filtering, etc etc. Additional factors to consider are the amplifier’s topology (class A, B, AB, or D) and the speaker’s impedance and power ratings.

A piezo buzzer is very simple and can be integrated with little additional work, but also with very little configurability and does not introduce typical audio handling/processing stages. Thus, a solution using the STM32 chip to play short audio samples / beeps / messages seems most appropriate. The simplest implementation using the STM32WB would be to only support a mono file and single speaker. Supporting stereo would require more power, components, as well as DAC outputs, and memory to store samples. As the audio circuit is an add-on, I am looking to use a low-cost implementation with minimal features.

The memory requirement of storing samples is calculated using the bit depth, length of the sample, and sample rate. An example 16-bit mono track sampled at 44.1 kHz for 1 second takes up about 86 kB. Thus, a FLASH chip to store the data is required to store even short samples as this would otherwise require a large percentage of memory in a microcontroller. A high bitrate is not needed for this project, so a target of 16 bits is preferred but ultimately depends on what the DAC component can support.

It is noted that it seems to be the recommendation for the audio amplifier stage to have a power output that is at or just above the rating of the speaker. If the amp is underpowered, it could cause clipping and damage to the speakers. Likewise, too high a power output rating can also damage speakers. A general strategy is to use an amp that is rated for 75% to 150% of the speaker’s power output.

## Audio Chain Considerations

The audio chain has some important design factors to consider. These include the digital-to-analog conversion stage, amplification stage, and speaker selection. In general, the conversion happens via a DAC either inside the microcontroller or via an external one. Since the STM32WB does not have an onboard DAC, an external DAC must be used. It should be noted that audio DACs will typically have an Inter-Integrated Circuit Sound (I2S) to streamline communication of the audio sample data to the DAC.

The I2S protocol is a serial interface protocol that is used for transferring two-channel, digital audio as pulse-code modulation (PCM) between two ICs. It was first designed by Philips Semiconductor, now known as NXP Semiconductors, in 1986. It uses three lines to transfer data, 1) a data clock, 2) a left-right channel / word select, and 3) a serial data line. A value of 0 on the word select line will send data to the left channel, a value of 1 sends data to the right channel.

After the signal is generated via the DAC, the audio must be amplified to support the power needs of the downstream speaker. This is done via an audio amplifier, which typically fall within the A, B, AB, or D classes. The different classes provide different benefits that relate to power efficiency, noise generation, EMI, and supply configurations. Class D amplifiers are the most efficient generally but does add switching noise due to the architecture. For battery-powered applications, Class D is probably the most appropriate.

Browsing on Digikey or Mouser shows that some low-end DACs and amplifiers can be acquired or $1-3, but options are few. One such DAC the [ES9023P](https://www.mouser.com/ProductDetail/ESS-Technology/ES9023P?qs=sPbYRqrBIVn%252BcPkDS1h3%252BQ%3D%3D&srsltid=AfmBOorCMcTxVYMyqf4S6Ji0nHnvsG7kguJZSmwmWppDsx8XpjyiHKZu) from ESS Technology, which is cost-effective and simple to integrate for audio applications. However, its ground-centered output emphasizes a key factor in the audio chain: if the signal is ground-centered or not. In simple terms, the DAC employs an internal charge pump to generate both positive and negative voltages centered around ground to create the analog signal. A major benefit of this is that a DC component is *not* added to offset the analog signal for systems that are operating strictly above ground. It appears this reduces components required in the audio chain, offers better noise performance, but also requires that downstream amplifiers use a bipolar supply. Generating a bipolar supply is not trivial and adds additional cost.

As pointed out, it is important that the DAC and amplifier use the single-supply constraint with a DC offset or bipolar supply. Available low-cost amplifiers like the TS4962IQT appear geared towards the single supply setup, leading to a mismatch at the input. If the ES9023 is used and the output is centered around ground, the TS4962 will be incompatible because the VCC range would be from 0-3.3 or 5V.

Another important characteristic is whether the analog lines are differential or single-ended. It appears that some DACs or codecs could provide differential signaling to handle interference better. Amplifiers like the TS4962 allow for both configurations.

When choosing a speaker, the impedance of the speaker and the power rating must be matched to the amplifier to ensure proper operation. It is recommended that the amp supports a higher power rating, one that is less is said to cause audible defects like clicks or pops and at worst could cause damage.

## Integrated DAC and Amplifier

Due to the minimal requirements of the audio system, cost and ease of implementation led me to selecting the [TAS2505](https://www.ti.com/product/TAS2505-Q1) from Texas Instruments as DAC + amplifier block. It supports mono audio and receives data over analog or I2S, providing up to 2W via the class D amplifier. I2S and SPI can both be used to interface with internal controls and is packaged in a 24-pin VQFN chip. Bit depths of 16, 20, 24, or 32 bits are supported. A single-ended headphone output is also available.

It does require a 1.8V rail to supply AVDD and DVDD. The IO level is set by the IOVDD pin and can range from 1.1 to 3.6V, allowing for compatibility with the STM32WB IO. An additional power pin, SPKVDD, sets the power of the final stage and ranges 2.7 to 5.5V. Since the power output depends directly on the speaker VDD pin, I have opted to include a 2nd DC-DC convert to scale the USB input voltage to 5V. The exact implementation is discussed in the Power Configuration section. Since the 5V DC-DC converter should be sourcing a major of the current into the audio block, an LDO is added to convert to 1.8V. The device datasheet indicates that both the digital and analog rails can use the same rail. Since the rail will only be feed the audio circuit, I have opted to not use two separate rails even though it is generally good practice to do so.

The DAC requires a CLK signal fed in from the MCLK, BCLK, or GPIO/DOUT pins. This is used for syncing on-chip activity. The on-chip PLL can also be used to generate the required clocks from a wide range of fractional multiplication values to set the main clock. Since I had the MCLK pin available in the STM32, I have opted to connect it to the TAS2505.

Texas Instruments provides an application reference [guide](https://www.ti.com/lit/ug/slau472c/slau472c.pdf?ts=1740604691643) to aid in using the device. It will be helpful in the programming stage and was used to configure the hardware circuit.

### LC Filter Output

The footprints for an LC filter are included to address EMI concerns of Class-D amplifiers. The Application Report [“Post Filter Feedback Class-D Amplifier Benefits and Design Considerations”](https://www.tij.co.jp/jp/lit/an/sloa260/sloa260.pdf) notes that LC filters are typically included at the output to meet FCC/CE regulations for emissions. The Application Report by TI provides a good reference for the design of an LC filter for both AD and BD modulation schemes.

A diagram of a circuit

AI-generated content may be incorrect.Additional information is provided by an applications engineer on the Texas Instruments [forums](https://e2e.ti.com/support/audio-group/audio/f/audio-forum/703441/tas2505-output-lc-filter-calculation) when discussing the output filtering stage. The typical switching frequency is 300 kHz and the min/max is 250/350 kHz, using BD modulation. It is confirmed in 2nd [post](https://e2e.ti.com/support/audio-group/audio/f/audio-forum/812319/tas2505-q1-output-signal-question) that the output without a filter is a PWM signal, not the analog signal you might expect for audio. A third [post](https://e2e.ti.com/support/audio-group/audio/f/audio-forum/979784/tas2505-q1-low-pass-filter-for-8ohm-speaker-requirements) confirms that a low-pass filter is not mandatory to drive an 8-ohms load, but that it is possible to add an LC filter in order to eliminate EMI noise and provides a link to an Application Report [“LC Filter Design”](https://www.ti.com/lit/an/slaa701a/slaa701a.pdf?ts=1743985690293). The report indicates that for BD modulation, a type-2 filter can be used. Through some change of perspective of the schematic components, a differential speaker and filter can be modeled as a single-ended one where the load is halved.

Figure 2: BTL Differential Speaker

A diagram of a circuit

AI-generated content may be incorrect.Following the report, the component values for a filter are calculated. Let , where . The parameters from the system are , , and . Solving for L yields that . The capacitor value can be calculated via . The next closest standard value is 0.68 uF, so that will be chosen.

Figure 3: Equivalent Single-Ended Circuit

The document notes that both metalized film capacitors and ceramic capacitors are used in high-power designs. Metalized film types are recommended over ceramic caps due to performance across temperature, voltage, current, and frequency but it is noted that ceramics can be used in space-constrained designs. Agreeing with this sentiment, it is recommended by [EE Times](https://www.eetimes.com/understanding-output-filters-for-class-d-amplifiers/#:~:text=Ceramic%20capacitors%20experience%20large%20changes%20in%20capacitance%20as%20the%20voltage%20across%20them%20changes%2C%20which%20can%20result%20in%20distortion.) to not use ceramic capacitors because of their derating with increasing voltage applied. This factor is dependent on the applied voltage and capacitor voltage rating. Texas Instruments notes that a guideline equation of can be used to estimate the expected capacitor value in the system. Thus, a 100V rated capacitor experiencing 50V volts will derate by 50%. The total voltage experienced over a capacitor can be estimated using . Given the expected system parameters, , , . This gives an expected voltage over the capacitor, . Typical voltage ratings for 0603 packages include 16V, 25V, 35V, and 50V. This gives a range of derating starting from 29% to 9% percent. This derating will move the cutoff point for the LP filter to a lower frequency, attenuating higher frequency signals to a degree. An additional 0.1 uF in parallel with each main capacitor could add the capacitance lost due to derating.

### Analog Input

The TIDA-01529 is the evaluation board and shows that only a capacitor of 1 uF, 50V rating, is needed for AINL and AINR inputs. The evaluation board used RCA mono connectors, specifically the 971 and [972](https://www.digikey.com/en/products/detail/keystone-electronics/972/317317). Based on the schematic of the part, the signal is fed to the center socket pin. I have included a 2.54mm 3-pin header to minimize used space and allow an external connection to the inputs.

If the analog inputs are not used, they should be shorted together according to the datasheet. An 0603 footprint for a jumper is included to do this,

### Headphone Output

The headphone output could be used by wiring up an external connector jack, but the output will only be mono. The datasheets shows a 47 uF capacitor on the HP out signal, so this has been included.

### Standby and Shutdown Modes

The TAS2505 has both standby and shutdown modes available. When powered by two buses, one for the speaker and another for the 1.8V DVDD and AVDD inputs, standby mode can be triggered by disabling internal registers. These include the following register blocks, as indicated by a [post](https://e2e.ti.com/support/audio-group/audio/f/audio-forum/970509/tas2505-question-for-power-saving-mode) on the TI forums:

1. DAC Channels (page 0 / register 63 / bit D7)
2. Master reference (page 1 / register 1 / bit D4)
3. Headphone output and analog inputs (page 1 / register 9 / bits D5, D1, D0)
4. Speaker Class-D driver (page 1 / register 45 / bit D1)

A full shutdown can be achieved by turning of all external supplies ([source from TI forums](https://e2e.ti.com/support/audio-group/audio/f/audio-forum/849669/tas2505-about-standby-mode)) and IOVDD can be turned off before the rest of the power is cut. The [application reference guide](https://www.ti.com/lit/ug/slau472c/slau472c.pdf?ts=1742237505640) indicates that technically SPKVDD should be provided first, then IOVDD, then AVDD/DVDD but there is no required minimum time in between each stage. There is a 10 ns time between when DVDD is provided and the reset pin can be raised.

To accommodate a full shutdown of the audio circuitry, the STM32 has the ability to shut down the 5V SPKVDD rail, the 1.8V DVDD and AVDD rails, and the 3.3V IOVDD rail though control of the respective enable functions for each rail. The 5V DC-DC regulator and the 1.8V LDO both have enable signals that can be controlled directly. Since the 3.3V IOVDD rail is connected to the system’s 3.3V rail, a load switch is placed near the audio circuit to provide control of the power into the IOVDD pin.

The TAS2505 allows for some IO to be routed from the device. To take advantage of this, the DOUT pin is connected to a status LED. DOUT can be controlled through setting internal registers over I2C. The MISO pin can be remapped to provide an interrupt signal for certain conditions in the device.

## Speaker Selection

The speaker should be an 8 ohm-rated speaker with a small footprint to be compatible with the amplifier stage output, which can output up to 1.7W when SPKVDD is 5.5V. The [CMS-151135-078L100A](https://www.sameskydevices.com/product/audio/speakers/miniature-(10-mm~40-mm)/cms-151135-078l100a) is a 0.7W, 8 ohm speaker. In comes in a 15 x 11 x 3 mm package and is readily available at Mouser and Digikey for around $2.30. It supports a frequency range of 100-20 kHz, which covers most of the audible range for music. The bass range is lacking a bit, as the frequency range for audio is generally quoted as being from 20 Hz up to 20 kHz. This is acceptable, however.

The resonant frequency of a speaker also has an important part in the design. The [post](https://blog.miscospeakers.com/understand-resonant-frequency-and-speaker-design) “Understand Resonant Frequency and Speaker Design” by Misco provides a quick brief on how a speaker’s resonant frequency will affect design and performance. The speaker will operate the most efficiently close to its resonant frequency, any frequencies below this are not easily produced and may not come through. The CMS speaker lists a resonant frequency around 550-850 Hz, which should be fine.

The speaker comes in a few variants as to how it plugs into the system. One version has spring contacts, others include different connectors. To save space, the variant with Molex 51021-0200 connector has been designed for by adding the 0532610271 header.

Other speakers readily available on Amazon, Adafruit, Digikey, etc generally seem to either come with bare wire leads or use the JST-PH type of connector with pitches either at 1.25mm, 2.0mm, 2.5mm or 2.54mm. I have read that Pin 1 is usually the ground/negative wire for the small 2-pin connectors such as these.

# USB

The STM32WB series offers a full-speed (FS) USB 2.0 interface, 12 Mbit/s. Up to 16 endpoints are supported.

The connector appropriate for the design is a USB 2.0 connector, which does not route the additional USB 3.0 pins. This reduces pin count and soldering complexity. The primary pins that are routed are the standard D+, D-, CC1, and CC2 pins. The SBU pins, the sideband channels, that are available through the connector should be left floating as they are not used. These are used in other protocols supported by the USB-C technology such as Audio Adapter Accessory Mode, USB4, Thunderbolt, or HDMI.

The shield is connected to ground, according to USB-C specifications.

The USB data lines, D+ and D-. should be routed as a differential pair. The trace impedance should be matched to the cable impedance, which is set around 90 ohms. This [document](https://www.ti.com/lit/wp/slyy109b/slyy109b.pdf?ts=1743131061614&ref_url=https%253A%252F%252Fwww.google.com%252F) by Texas Instruments indicates that the specification allows for connecting of the D+ and D- pairs directly. Some designs elect to use a USB 2.0 Mux to improve signal integrity when merging the pairs. The article [“Routing Requirements for a USB Interface on a 2-Layer PCB”](https://resources.altium.com/p/routing-requirements-usb-20-2-layer-pcb#:~:text=Differential%20Skew,is%20very%20large!) by Altium notes that for High Speed USB, which is faster than the Full Speed, should allow for about 600 mils (15 mm) of length mismatch.

A screenshot of a computer

AI-generated content may be incorrect.USB data lines (D+, D-) should be impedance matched to 90 ohms differential. A trace width of 16.65 mils with spacing of 8 mils should accomplish this on the basic JLC04161H-7628 stackup.

The CC tracks should be 10 mil to provide enough current capacity for supported cables, per the datasheet (9.4.1.5).

As I have filled some sections of the top of the board with ground, but my USB data routing is using a coupled microstrip mode, the top-layer ground should be pulled back around the USB traces to avoid coplanar waveguide effects. The Altium article [“RF Microstrips and Ground Plane Clearance: How Close is Too Close?”](https://resources.altium.com/p/microstrip-ground-clearance-how-close-too-close) notes that a field solver is required to recommend a separation distance but concludes that a rule of 3W is generally too conservative. An example given is that for a microstrip trace of 12 mils, a separation distance of 36 mils should be used to pull back the ground planes surrounding the strip. Here my mode is a coupled microstrip where each line is 16.65 mils wide, so the separation from ground could be 48 mils.

The application note [“Introduction to USB hardware and PCB guidelines using STM32 MCUs”](https://www.st.com/content/ccc/resource/technical/document/application_note/group0/0b/10/63/76/87/7a/47/4b/DM00296349/files/DM00296349.pdf/jcr:content/translations/en.DM00296349.pdf) by STM indicates that no serial termination resistors are needed on the data lines, so none have been included. Other points to note about the hardware setup is that there is a USB\_NOE output port which indicates if the USB transceiver is active in the STM32. While useful, it is not necessary and not implemented because it would interfere with the JTAG pins. There is also a USB\_CRS\_SYNC output which is related to the clock recovery system and can be used as a trigger in a fine-tuning process for the 48 MHz frequency. It will not be utilized as other inputs include the 32 kHz crystal oscillator and USB Start of Frame signal.

# Current Monitoring

The board supports current monitoring of the main power input through the USB-C connector through a sense resistor, current sense amplifier, and built-in ADC of the STM32.

The accuracy of the current sensing comes down to maximizing the sense resistance while limiting the power dissipation in the resistor and offset error and reduced common-mode rejection due to bias currents. A recommendation from Texas Instruments for their INAx180 line is to keep sense resistors no larger than a few ohms.

Power dissipation in the resistor is governed by . 5A is the maximum allowed over USB-C lines. The predetermined settings discussed in the USB-C Bus and PD Controller section allow up to 3A sourced from USB, so current monitoring will be optimized according to a 3A max input. A minimum current will be estimated to be 50 mA. The full-scale range of the analog output should correspond to a current range of 0 to 3.2A to provide some headroom above the normal operating conditions. If the maximum dissipation is set to 0.2W, this limits the sense resistance to .

The current sense amplifier selected is Texas Instruments’ [INAx180](https://www.ti.com/lit/ds/symlink/ina180.pdf) series, which supports both high side and low side sensing. In this case, we are using high side sensing since the resistor is placed directly after the USB port and sees the VBUS voltage of 5-20V. The INAx180 has options for gains of 20, 50, 100, and 200 V/V and common mode voltages up to 26V. It does support a shutdown mode by removing the power to the power pin, though this isn’t required.

Typically, analog lines should be clamped to safe levels depending on connected circuits. In this case, the analog input feeds directly into the STM32WB’s ADC which has a range of 0 to 3.3V. A common approach includes using Schottky diodes, but this is not needed due to the design of the INAx180. The output of the device is limited by the voltage level at its VS pin, which is set at 3.3V so that its maximum level is within safe operating conditions for the STM32WB.

The main equations given by the amplifier’s datasheet to choose the gain for a sense are given below. Notae that and refer to the positive and negative swing values and are provided in the datasheet in section 7.5, limiting the range that can be used. Given the system design, and . The Total Offset Error is specified to be and the gain error is 1%.

Choosing a resistor of 10 mOhms and amplifier gain of 100 satisfies the rail requirements and gives a total error of 1.08%. At a max current of 3A, the voltage that should be output to the ADC input is 3V. The ERJ-8CWFR010V resistor is chosen as the sense resistor, which comes in a 1206 package with 1% tolerance and 1W power rating. The

The [INA180A3IDBVR](https://www.digikey.com/en/products/detail/texas-instruments/INA180A3IDBVR/8132988) is the appropriate variant to have a gain of 100 in an SOT-23-5 package. It supports a power supply voltage of 2.7 – 5.5V to power the device. The datasheet also indicates that the device can be powered down using a logic gate or switch to drive the VS pin. Current leakage when the device is off but common mode voltage is present is said to be dictated by the bus voltage and 500 kOhm impedance in the device to ground. Small leakage should be acceptable for the short amount of time the device is unpowered.

To filter out high frequencies, a simple RC filter is placed at the output of the INA180x. Default RC values of 1k and 100 nF should create a LPF with a cutoff frequency around 1.6 kHz.

# Temperature Monitoring

The footprint for a temperature monitor is included to provide additional functionality. The [TMP1075NDRLR](https://www.digikey.com/en/products/detail/texas-instruments/TMP1075NDRLR/15222310) from Texas Instruments has an alert pin that can operate in two modes to indicate significant states in the system. The comparator mode activates the alert pin if the temperature measured is out of the high and low bounds. The interrupt mode will sound the alert if temperature rises above the high setpoint and will sound again when the temperature falls below the low setpoint.

Note that the temperature measured could be greatly effect by heat generated from circuit elements. The application report [“Temperature sensors: PCB guidelines for surface mount devices”](https://www.ti.com/lit/an/snoa967a/snoa967a.pdf) by Texas Instruments discusses this issue and ways to address it. Based on the report, I have opted to both remove ground pour around the area to decrease thermal conduction and have put cuts around the IC to create an island. Evaluation will be needed to determine how effective the combined strategies are at isolating the sensor so that it measures ambient temperature.

The device supports up to four addresses depending on the state of the A0 pin. Tying it to ground sets the 7-bit address to 1001 000.

# Bootloader from Serial Memory

The design should be able to support a bootloader as is. USB, UART, I2C, and even the local Flash memory could be used to transfer the update into the device if the bootloader allows. The application note AN2606 [“STM32 microcontroller system memory boot mode”](https://www.st.com/resource/en/application_note/an2606-stm32-microcontroller-system-memory-boot-mode-stmicroelectronics.pdf) serves as the main resource from STM on bootloaders in their chip lines and could serve as a starting point when developing the firmware.

# EMI/EMC

Without specialized tools, internal testing will be limited. I must investigate what tools I could acquire to perform some rudimentary testing, but in general proper EMI testing looks to involve special equipment and room treatments to get valid data. The most I can do is to implement best Signal Integrity related practices for now and try to include some common test points so that I can analyze performance with what I have on hand.

## Signal Integrity Practices

To mitigate EMI, various techniques are used to target common sources of emissions and practice signal integrity. These are discussed below in various details depending on how specific they are to this design and what block they apply to.

#### Reference Change Vias

Vias are placed by nearby signal vias to allow the signal reference to change from the L2 to L3 ground plane whenever possible.

#### Block Organization

Once all the major components were identified, the board was sectioned to try to keep like-circuits in their own physical space on the board as much as possible. The power circuitry is all located around the USB-C connector, the RF components on the opposite side, and the analog audio and temp sensing circuits on located towards to middle. While I tried to place components in their own section, some such as the QSPI flash are placed as close to the STM as possible as it could use a very fast clock and edge rate.

#### Minimizing trace length

By using good component place and the flexibility of the peripheral pin mapping inside the STM32, trace length for most busses was minimized. While the USB data lines are longer than I would like, I opted it was more important to center the STM32 where it is to have enough area to fanout all the signals and keep the RF trace shorter, which is much faster and sensitive compared to the differential USB lines.

#### Trace spacing

In general, all traces are spaced equally away from each other to try to follow the “3W” guideline, which advises that a separation of three track widths significantly decreases cross talk. While this is a four layer board where the inner layers are ground, I have generally sought to avoid routing traces directly on above/below each other, as this would also lead to crosstalk.

#### Trace Width

A minimum trace width of 6 mils is used where signal traces are congested. When more than 3W of separation is available, I generally tried to route with 8 or 10 mils to minimize ohmic losses. Power traces are generally 20 mils at minimum, though the main busses utilize 40 or 60 mil wide traces in the power section of the board.

#### Serial Termination Resistors on High-Speed Lines

One common source of EMI is through mismatched impedances of a driver, the transmission line, and the load. A driver could be an output clock from a microcontroller and a load could either be another IC for memory or sensing. Typically, the driver will have an impedance in the range of 10s of ohms, the transmission line will be matched to around 50 ohms, and a load will have a high impedance in the kOhms/mOhms region. This mismatch creates reflections, which causes signal integrity and EMI issues. The reflection % is given by . If then a positive reflection occurs, where the reflection is added to the base signal, and overshoot occurs. The opposite case is undershoot, where reflection subtracts from the signal. In today’s digital domain and architecture, undershoot is more of the problem due to lower digital rails, noise margins, so generally it is recommended to err on the side of designing for overshoot.

When traces are sufficiently long and the frequencies on the trace are fast, antennas can be formed as the length of the antenna depends on the wavelength of the signal. Fast signal means shorter wavelength, allowing for shorter antennas. The issue can present itself in unexpected circumstances, as a relatively lower clock signal might not catch the designer’s attention if they neglect to consider the edge rate of the signal. In general, to create square pulses over a digital line there must be frequencies several times faster than the base frequency in order to create nice transitions. These higher frequencies should be accounted for when analyzing the length of the trace.

The application note [AN4760](https://www.st.com/resource/en/application_note/an4760-quadspi-interface-on-stm32-microcontrollers-and-microprocessors--stmicroelectronics.pdf) from STM references the [AN4661](https://www.st.com/resource/en/application_note/an4661-getting-started-with-stm32f7-series-mcu-hardware-development-stmicroelectronics.pdf) note for layout guidelines. The main takeaways are to design for 50 ohm traces and if the trace is over 120mm then termination is required. Match the data pin lengths to within 10mm of each other, an “S” style serpentine can be used. Do not use serpentine routing on the clock trace and provide at least 3x the trace width away from other signals.

For reference, Phil’s lab provides a brief overview of SI and EMI in this [video](https://www.youtube.com/watch?v=VtzPL8wQ8-E).

#### Audio Circuitry

The audio block follows this [article](https://www.ti.com/lit/an/snaa050a/snaa050a.pdf?ts=1740599253596) by Texas Instruments to mitigate EMI. The Class-D amplifier is efficient, a desired trait for low or battery powered systems, but also utilizes switching in the architecture which poses an EMI concern. A couple recommended techniques from the article to combat this include:

* Standard power decoupling practices
* Power planes are backed off from the edge of the PCB
* Adequate termination of all high-frequency clock lines
* Proper filtering of PCB connectors
* Avoid loop antennas, where a forward and return current is on a well-defined conducting path
* Keep audio traces as short as possible

#### Via Fences

This [article](https://www.nwengineeringllc.com/article/designing-your-rf-pcb-stackup.php) explains that via fences around different sections of circuitry can be beneficial. The RF circuit utilizes a via fence and the rest of the PCB uses a more moderate spacing to connect all the ground layers.

# DFM Check

DFM checks include using the built-in KiCad DRC with parameters acquired directly from the JLCPCB page of their manufacturing capabilities. Where possible, I generally avoided placing vias in pads. As well, I avoided 90 degree turns and sharp corners in copper layers to avoid acid traps. All copper should be well connected with no small floating copper sections. As discussed already, I used copper planes to balance copper placement amongst the top and bottom layers. Copper is pulled back from the edge of the board to avoid shorting between layers.

Another strategy I used was to place all larger components on the top side of the board and only place small components on the bottom. This was done as large components would require special treatment so that they don’t fall off during reflow. While I don’t have exact numbers right now to estimate what components can be reflowed in place due to the solder tension, I have read that normal 1206 passives should be fine.

# Excluded Features

## WiFi and TCP Client

While WiFi connectivity was originally specified in the original design, it has been excluded upon discovering the fact that no single STM32 device supports both BLE and WiFi. Additionally, WiFi connectivity is not supported by the active STM32 lines so building a board around an STM32 and WiFi would require either a secondary WiFi-enabled IC or dongle. Given that only working with BLE would create a project with significant scope allowed me to decide to do without any WiFi features.

Additionally, since STM32 does not manufacture WiFi chips I reckon that if I were to require WiFi in a project, I would most likely be best served by looking at manufacturers that target that. For example, the ESP32 is a very common family supporting WiFi and Bluetooth that could prove useful in future designs. Nordic Semiconductors also is well known for their wireless-enabled microcontrollers