BLE Demo Specifications

Table of Contents

[Overview 2](#_Toc164350179)

[Feature Request Summary 3](#_Toc164350180)

[Specifications 4](#_Toc164350181)

[STM32 Architecture and related toolchains 4](#_Toc164350182)

[BLE and Inverted-F Antenna 4](#_Toc164350183)

[WiFi and TCP Client 4](#_Toc164350184)

[DMA Utilization 5](#_Toc164350185)

[High-Capacity Non-Volatile Memory 5](#_Toc164350186)

[Serial Memory 5](#_Toc164350187)

[microSD Card 5](#_Toc164350188)

[Memory Encryption 6](#_Toc164350189)

[JTAG 6](#_Toc164350190)

[DC-DC Converter Design 6](#_Toc164350191)

[USB-C Power Input, Power Delivery 6](#_Toc164350192)

[USB Communication 7](#_Toc164350193)

[Multiple Power Sources 7](#_Toc164350194)

[Local Battery 7](#_Toc164350195)

[Real-Time Clock 7](#_Toc164350196)

[Display Capable 7](#_Toc164350197)

[EMC/EMF Testing Ready 7](#_Toc164350198)

[Antenna Tuning Ready 8](#_Toc164350199)

[Bootloader from Serial Memory 8](#_Toc164350200)

# Overview

In this document, I will define the initial feature request list and final specifications for the BLE Demo project. This project is to be carried out by Jake Kapala as a personal project to work primarily with the STM32 architecture and wireless technologies such as BLE and WiFi. Design, assembly, and testing stages all shall be done by Kapala using tools in his lab space.

In the Feature Request Summary, the features are listed via bullet point to initially capture what the project should integrate. This list is not exhaustive but used to generate the initial survey of technologies that should be considered.

The specifications detail what technologies should be included and any stipulations regarding how they should be used. These should directly serve as inputs to the design phase and referenced in the implementation document.

# Feature Request Summary

The following features are requested by the client. These hit a majority of the goals of the project. A crucial aspect of the project is to utilize STM32 Architecture to expand my experience with other chipsets besides those from Microchip, namely their core PIC chips. However, given the nature of embedded chips, not all requested features may be available with one microcontroller. Most features

1. STM32 Architecture and related toolchains
2. BLE and Inverted-F Antenna
3. WiFi and TCP Client
4. DMA Utilization
5. High-Capacity Non-Volatile Memory
6. Memory Encryption
7. JTAG
8. DC-DC Converter Design
9. USB-C Power, Power Delivery
10. USB Communication
11. Multiple Power Sources
12. Real-Time Clock
13. LCD Capable

# Specifications

The following specifications shall detail the technologies selected and what features are approved or unavailable. Since this device will serve as a development board, the specifications are defined rather loosely. It is generally not imperative that any one feature is included. Instead, I have decided to include as many technologies as are relevant to maximize development potential with the board. The main technologies that should be integrated are the STM32 architecture and BLE / WiFi protocols.

## STM32 Architecture and related toolchains

The project should utilize an STM32 controller that provides access to the typical STM32 architecture, features, software, and support. Voltage rails should be standard 3.3V to minimize logic translation but other standard rails such as 1.8V and 5.0V are acceptable if they reduce complexity in overall system design. Some translation for external tools shall be required if other logic levels are used in order to support serial readout to a local computer because the tools available use 3.3V logic.

Other than a STM32 chip, no other specifications shall limit the design and practical designs such as RAM or flash size shall be chosen to balance cost with estimated program size. The number of I/O should be sufficient enough to support the requested features but will most likely be limited by the STM32 lines supporting wireless stacks. Thus, if I/O does not prove to be sufficient to support the communication buses or other GPIO functions, it should be noted if a particular feature cannot be included and the reasoning for prioritizing certain features.

## BLE and Inverted-F Antenna

BLE should at least be 4.0 or greater. Version 5.0+ is preferred to work with modern standards. Since stack development can be a time-consuming process, it is often a good idea to utilize available frameworks and or example projects to get basic firmware up and running. The stack support primarily will be determined what STM offers alongside their hardware selection.

The antenna that should be utilized should first and foremost consider any recommendations from STM for a given controller. Beyond that, it is recommend to consider an inverted F antenna because of their use in IoT solutions and benefits. It should be detailed the pros/cons of the final antenna design.

## WiFi and TCP Client

A simple TCP client option should be considered and integrated in the design. In the case that BLE and WiFi cannot be supported at the same time due to hardware reasons, BLE should be prioritized. If the firmware image becomes too large to support both a TCP client and BLE client, different projects could explore the protocols separately.

If WiFi is supported, potential applications include a MQTT client. Additionally, TLS should be designed for to ensure standard security is included.

## 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. Transfer size should be flexible. The exact usage is not necessarily important, the value is in using DMA itself to handle transfers.

## High-Capacity Non-Volatile Memory

The board should both be able to utilize a surface mount memory controller but also have an MicroSD housing to allow the use of external memory chips. This is to provide access to local memory as well as provide the ability to design with common technologies in mind.

### Serial Memory

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 in the 124 MB range would suffice.

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 seem to range around $10-20 for such larger capacity chips.

FRAM and MRAM are both better for endurance and size but are quite expensive. Would require specialized application to utilize benefits. In this project, NOR or NAND flash is probably most suitable.

### 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

In general, SD cards have 9 pins. 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.

SD cards may or may not have wear leveling available, it depends on the manufacturer. In the system design, this should be considered. A filesystem library will be needed to format data to the FAT-formatted card.

SD card integration should provide the ability to interface with larger capacity memories. It is not strictly necessary that data be updated from the computer, but the system design can utilize the removable aspect in applications. A filesystem most likely will be necessary but potential applications without it could be considered if that is supported by the SD card.

## Memory Encryption

To enable encryption on memory, the microcontroller should have some encryption hardware onboard to provide random number support. The exact encryption method is not specified, this depends on the data and use case. Potential types could include symmetric (AES) or asymmetric (RSA). In the design output file, a discussion on the integrity of the data stored at rest using the encryption method should detail the strength and weakness of the design.

## JTAG

The JTAG interface should be supported to allow for debugging, programming, short checking, etc etc to utilize the benefits that JTAG offers.

## DC-DC Converter Design

The main power bus should support a wide range of inputs. It is expected that power will come it at 5V through USB-C. The DC-DC should either be a buck topology or a buck-boost topology if is determined that a higher voltage bus (besides 3.3V) is needed. The buck topology should be able to handle a range of voltages expected from USB-C devices such as 5V-15V. Current requirements are estimated to be under 1A due to the device mainly being focused around the microcontroller with wireless capabilities. Low power status isn’t a necessity but should be considered, especially if a battery is added to the design.

Since the downstream circuit involves RF, power stability will be important. The design should consider if it is appropriate to feed the DC-DC output directly into the RF circuitry or if a LDO or similar regulator is required to provide a more stable output.

## USB-C Power Input, Power Delivery

The main power bus should be generated from a USB-C port and utilize off-the-shelf power bricks and cords. This should allow up to 3A and various voltage settings up to 24V. The actual voltage input shall be left to design choice based on required power rails. Care should be given to provide proper protection from transients and allow for power negotiation using Power Delivery. It could be beneficial to integrate a PD controller into the design to negotiate a 9V or 15V level if a different input voltage is needed to source the PDN.

## USB Communication

USB-C communication primarily comes down to the firmware stack and routing the USB data traces to the connector. The CDC class is probably most appropriate. If it is supposed, USB can be used to connect to a debug stream.

## Multiple Power Sources

The design should have at least 2 power sources, the main bus and a local battery for RTC operations. An additional battery source or input terminal, besides USB-C, can be included for convenience or evaluation. The main bus will be supplied by the USB-C terminal.

### Local Battery

Typically batteries for RTC applications operate at 3V, the CR2023 being one example. These are typically 20mm in diameter. Thus, the device should integrate the necessary battery holder to seat the battery and connect to the RTC circuit.

## Real-Time Clock

To have an RTC, an external battery must be included. The microcontroller must also have an RTC and battery-mode in order to setup up an RTC. If the microcontroller does not support a RTC, an IC should be added. A calendar is not required. Features like automatic daytime savings calculations or time zone support is not important.

## Display Capable

A simple display able to display characters is appropriate for this design. One display that could be appropriate for the application is a character LCD display from FocusLCDs (C204AS-FTW-LW63 - [focuslcds.com](https://focuslcds.com/product/c204as-ftw-lw63/)). This display is transreflective, meaning it should not require a backlight.

Character LCD displays contain 14 pins if no backlight is attached ([focuslcds.com](https://focuslcds.com/journals/intro-to-lcd-display-programming-character-lcds/#:~:text=Character%20LCD%20displays%C2%A0contain,displays%2C%20so%20pay%20attention!)). There are 8 data pins, an enable signal, a R/W pin, register select, and 3 power pins. If necessary, 4 pins can be used instead of 8 for data lines. While an SPI interface is acceptable if it is offered, the 8-bit parallel line seems standard for character displays.

## EMC/EMF Testing Ready

Without specialized tools, internal testing will be limited. The design should utilize best practices, use simulations where possible, and integrate test points if they are applicable. In lieu of test results from the assembled system, a route to perform tests should be provided. This includes any tests that could possibly be down internally with the addition of inexpensive tools and where to send the device for external tests.

## Antenna Tuning Ready

The device should support the ability to tune the antenna using data gathered from testpoints or usage. The recommended antenna to use is the inverted-F antenna, which will have a fixed design but might be able to be tuned using the matching components.

## Bootloader from Serial Memory

The design should plan to use a bootloader to flash new firmware. The firmware should ideally be in serial memory and use CRC-32 or alternatives to verify the image. An OTA update mode should be designed into the system, where Bluetooth is used to update the device. If the STM32 features an integrated BLE bootloader mode, it can be used. However it is a requirement to at minimum 1) store the firmware image to local memory and 2) write a custom bootloader to become familiar with writing to flash directly from an SPI or I2C connected memory.