Smart Bottle BLE Integration

Joshua Rizzolo

Ben Hawkins

1 Grand Ave, San Luis Obispo, CA, 93410, USA

June 10, 2022

Table of Contents:

List of Tables:

List of Figures:

List of Acronyms

ARM – Advanced RISC Machines are computers using a particular Reduced Instruction Set Architecture capable of running most common operating systems.

B2B – Board to Board connectors interface separate circuit boards for data transfer.

BLE – Bluetooth Low Energy is a version of Bluetooth optimized for minimal average power consumption by trading bandwidth and signal strength.

CA – Common-Anode is a configuration of LEDs in which the anodes of every LED are connected together, and the cathodes are left separate.

CC – Common-Cathode is a configuration of LEDs in which the cathodes of every LED are connected together, and the anodes are left separate (opposite of CA)

I2C – Inter-Integrated Circuit is a serial communication protocol using only a data and clock line. With I2C, the host first sends the address of the specific device it will communicate with to ensure that no other module acts on the data; then the data is sent.

LED – Light-Emitting Diodes are semiconductors that emit light of a desired color when a positive voltage is present between the anode and cathode.

MCU – Microcontroller Units are small computers on individual integrated circuits. They have many peripherals to facilitate data collection and processing in small systems.

PCB – Printed Circuit Boards are plates of silicon substrate with copper wires inlaid to connect the individual pieces of a circuit. They are noise-resistant custom boards on which circuits operate.

RGB – RGB simply stands for Red Green Blue. It is used to talk about LEDs that can emit each of these three colors of light in many combinations.

RTC – Real-Time Clocks are highly precise integrated circuits to track time and relay that information to other modules.

SPI – Serial Peripheral Interface is another serial communication protocol like I2C. However, SPI uses 4 wires: 2 separate data wires for bidirectional communication, a clock line, and a select line. The select line allows the server to select which peripheral to communicate with at the same time as data is transmitted.

SWD – Serial Wire Debugging pins are two designated pins to facilitate debugging of a microcontroller that has a limited number of pins. This setup is specific to ARM devices, but it can use the JTAG debugging protocol that is standard in many devices.

I. Abstract

In 1975, four percent of children aged five to nineteen were categorized as overweight or obese. As of 2016, this figure climbed above 18 percent [1]. Researchers at California Polytechnic State University San Luis Obispo (Cal Poly) want to investigate the effect of overeating in early childhood on later childhood obesity. This research requires collecting feeding pattern data on infants, which proves challenging. Parents cannot be relied on to regularly collect clean data due to factors including work schedule, multitasking, and general exhaustion. Thus, we have developed a tool to automatically collect data on feeding frequency and duration, as well as bottle angle during feeding.

The Smart Bottle is an attachment for baby bottles made of a flexible 3D-printed sleeve that slips on to the bottom of the bottle. The attachment itself is a two-inch (5cm) tall plastic housing with a diameter of seven centimeters in diameter to fit typical baby bottles. To collect data, this attachment houses a 3-dimensional accelerometer, a 3-dimensional gyroscope, and a real-time clock (RTC), and a volume sensor (left vague to protect intellectual property). For data storage and transmission, it includes an SD card reader, and a Bluetooth Low Energy (BLE) module. Finally, for power, it has a Lithium-Polymer (LiPo) battery, and a coin-cell battery. Using a neural network, measurement instruments identify feeding events and measure an infant's feeding duration, feeding angle, and amount eaten. All measurement data is then recorded on the SD card and broadcast to researchers or parents via Bluetooth. The added Bluetooth functionality will increase the drain on the LiPo battery but using a BLE-capable module minimizes this.

After manufacturing and assembly, testing found that necessary systems were all functional. The accelerometer returned numbers that changed with movement, which allows the

neural network to detect feeding events. The volume sensor was accurate within 0.5ml as tested with volumes ranging from 10 to 85 ml. The SD card correctly stored data without issue. The BLE module accurately transmitted data to connected devices. The RTC did not lose a second in the duration of testing. Finally, the board-to-board connector was successful in flashing the microcontroller unit (MCU) and the Bluetooth Low Energy module while the microUSB port facilitated successful uploading of Arduino code to the MCU.

II. Introduction and Background

A. Clients

The Smart Bottle is a tool for data collection, and its main users are parents of infants. Thus, researchers and parents comprise the two main groups of clients for this project. Both groups have different expectations for the Smart Bottle which must be considered.

Researchers at Cal Poly envisioned the Smart Bottle to collect their data. Furthermore, the collected data must be accurate in all dimensions, easily accessible, and uncorrupted. To meet accuracy needs, we use an accelerometer, and a gyroscope with sensitivities of 0.0 milli-gravities per least significant bit and 4.375 millidegrees per second per least significant bit [2]. To meet accessibility requirements, we have an SD card reader and a BLE module allowing wired and wireless data transmission. Further, the attachment comprising the Smart Bottle only requires two steps to remove and open in order to access data. Data purity needs will be discussed in the next section: “Motivation for New Method of Data Collection.”

Like researchers, parents also require data to be easily accessible. However, parents performing data collection are more interested in ease of measurement than researchers. For parents to regularly use this device, it must be simple to use, and it must tolerate being forgotten.

The device requires little knowledge to operate. It has no parts that parents must remove, nor does it have a power switch. Further, the bottle absolves parents of the need to record any data or remember specific rules. Parents pick up the bottle to begin a measurement, and they leave the Smart Bottle on a flat surface when not in use. In this state, the Smart Bottle goes into a low-power mode until it is lifted from where it was placed. In low power mode, the bottle can wait to be used for at least 300 days before needing a charge. Power consumption will be discussed in further detail in the "Requirements, Specifications, and Results" subsection of "Results."

TABLE I:

CUSTOMER REQUIREMENTS

Customer Requirement	**Engineering Specification**	**Justification**
a) Easy to use	1) 2 or fewer steps to turn on	The simpler the device is to use the more likely parents are to use it, the more likely they will gather good data for research.
b) Data is accurate	2) Registers a feeding event when lifted from a flat surface	Inaccurate data invalidates conclusions based on analysis of said data. If the Smart Bottle is to be used in research, it must accurately collect all the data that researchers have requested.
	3) Feeding angle is accurate within 10º	
	4) Volume measurement accurate to 1ml	
	5) Time measurement accurate within 10 s	
c) Data is accessible	6) There are 3 or fewer steps to access data	Fewer steps means fewer points for user error. Users are not engineers, and thus are more likely to get frustrated with the system not working according to expectations.
	7) Data storage solution is standard	The ability to access data from computers or smartphones gives researchers freedom in using and processing data.
d) Long Battery Life	8) LiPo Battery lasts more than a week on one charge	Infrequent charging means infrequent disassembly of the Smart Bottle. Those who would charge the bottle are

		unlikely to be engineers, so minimizing frequency of charging minimizes possibility for user error
	9) Coin cell battery lasts over 3 months on one charge	Coin cell batteries are not always rechargeable, so longer life means less electronic waste. Further, Accessing the coin cell to replace it requires removing the PCB from the housing and removing the battery from a tight enclosure. This process has many opportunities for damage to occur to the board. Less opportunity for damage is better.
List of Customer Requirements: a) Easy to use b) Data is accurate c) Data is accessible d) Long battery life		

TABLE II:

ENGINEERING SPECIFICATIONS

Spec #	Parameter	Target (units)	Tolerance	Risk (H,M,L)	Compliance	Test Equipment Needed
1	Steps to turn on	2	<	L	A, T, I	None
2	Feeding Event Accuracy	100%	5%	M	A, T	Computer, SD card reader
3	Feeding Angle Accuracy	5 º	±10º	M	A, T	Computer, SD card reader
4	Feeding Volume Accuracy	1ml	<	M	A, T	Computer, SD Card Reader
5	Feeding Time Accuracy	10 sec	<	L	A, T	Computer, SD Card Reader
6	Steps to Access Data	3	<	L	A, T, I	None
7	Standard Data Storage Solution	N/A	N/A	L	A, T, I	None
8	LiPo Battery Life	7 Days	>	M	A, T, S, I	Wattmeter
9	Coin Cell Battery Life	3 Months	>	M	A, T, S, I	Wattmeter

B. Motivation

Besides the Smart Bottle, there are several ways data collect could have been performed. However, some methods can significantly influence results, corrupting the data. If the data is to be used to research long-term effects of infant feeding patterns, it must not be corrupted. This section discusses the main alternative data collection methods and how they introduce inaccuracies in data.

The simplest collection method would involve parents recording feeding duration, amount, and angle themselves with the aid of video recordings. However, this approach puts pressure on the parents to measure, record, and track the data. Any error that parents make in any step of the process will corrupt the resulting data, possibly without any indication to the researchers that the data is incorrect. Thus, while it would be simpler for researchers to rely on parents to perform data collection, manual data collection introduces too many points for error.

If parents introduce error when performing data collection, then bring the parent and their infant to the lab. There, the researchers can collect the data while the baby eats. However, changing the environment in which the baby eats could drastically alter the baby's eating patterns, thus corrupting the data. While this reduces the effect of the parent on the experiment, it still introduces the possibility of significant error.

These two alternate methods illustrate well the constraints on this type of data collection. Manual data collection, and changes in environment can both influence how infants eat. Thus, data collection should be automatic and easy to perform anywhere to allow infants to eat normally while data is collected in real time. The Smart Bottle is a portable, unintrusive, and automatic device which satisfies the two requirements laid out for data collection.

C. Current State

The Smart Bottle project was started five years ago by Dr. Ventura and Dr. Murray. The project has been taken over by several groups of students in the years since, with each group adding functionality and addressing bugs from previous versions.

I began work on the Smart Bottle project in Fall 2021; by this point, the Smart Bottle included the functionality necessary to collect data. On the board were the measurement devices (or their connectors), the SD card reader, the real-time clock (RTC), the microcontroller unit (MCU), the coin cell and LiPo battery connectors, and the components comprising each subsystem. However, it was not fully functional.

When I began work on the Smart Bottle, I was told that the previous team found the accelerometer to be nonfunctional. While investigating the schematic for a possible cause, I found that the power pins (V_{cc}) for both the accelerometer and the RTC were connected to each other and nothing else. Beyond this, the terminal on the RTC labelled VBAT was connected to the LiPo battery and the coin cell. These two errors in the schematic left the accelerometer unpowered and allowed the microUSB to charge the coin cell with the LiPo. The coin cell charges at 3.3V where the LiPo charges at 3.7V, and that small difference is hazardous. Resolving these connection issues was my first priority after which I made my additions.

D. Scope

As stated in the abstract, previous versions of the Smart Bottle were not BLE capable. In these versions, data is only accessible via the SD card which requires removal and disassembly of the Smart Bottle attachment. While this is viable for a small number of devices, in larger pools of subjects data retrieval quickly becomes cumbersome and time-consuming. The addition of BLE technology to the Smart Bottle reduces the time researchers must spend retrieving data by eliminating the physical process of accessing the SD card and replacing it in the attachment. Further, the BLE allows parents to access the data themselves. This creates flexibility for the parents in multiple ways, including by allowing them to remotely send data to researchers without specialized knowledge of the device.

As is common practice with integrated systems, each version of the Smart Bottle has steadily fixed old subsystems and added new features and subsystems to be tested. This version is no different. The previous version had several errors including incorrectly labelled power connections for several subsystems. This BLE capable version seeks to resolve those issues present in the previous version, while also adding BLE functionality. The current version also introduces more mode indicator RGB LEDs around the circumference of the PCB to make current device mode more apparent to the user.

With these goals established, the scope of this project can be better defined in three sections: inspection of the old, small functional changes, and big additions. Testing the previous version revealed that the accelerometer was non-functional. Thus, the first task of this project was to inspect the accelerometer and related subsystems, where incorrectly labeled power connections were found. The previous version also showed the single mode-indicator LED to be insufficiently bright to properly display the Smart Bottle's current mode. Thus, more LEDs were added around the PCB. Finally, the focus of this project was the addition of the BLE module. This necessitated further small changes including increasing the PCB's diameter by a centimeter to accommodate the BLE module's "keep-out" zone for its antenna. Each of these tasks will be discussed in further detail in the "Methods" section

E. Necessary Functionality

With the scope of this project defined, we can now discuss the functionality from previous versions as well as the functionality this version adds.

Beginning with past versions, the main functions of the device were differentiating between feeding events and idle, measuring length of feeding events, measuring amount eaten, and storing all these data points. Auxiliary functions of the device included displaying status (idle, feeding, charging), supplying correct voltages from batteries to components, permitting reprogramming of the MCU, and remaining idle for extended periods of time without charging.

In the previous version of this project, most of this functionality was present. However, neither the RTC nor the accelerometer in this device were connected to power in this design. Thus, detection of feeding events and their durations was impossible. Further, while the device did indeed display its mode of operation, the RGB LED used as an indicator was too dim to make the mode obvious from all angles. The current iteration of this project seeks to remedy these issues with corrected power connections and three more RGB LEDs around the edge of the PCB.

While fixing errors from old versions is a necessary part of progress, the main purpose of the current version of the Smart Bottle is the addition of Bluetooth Low Energy communication. This addition of will simplify the process of retrieving data for analysis. It will not replace or in any way impede data storage on the SD card, leaving data physically accessible. The BLE module must be accessible for debugging and reprogramming if that becomes necessary. Thus, the board-to-board connector was modified to connect to either the MCU or the BLE module at a time. This allows reprogramming of both the MCU and the BLE module from the same debugging connection.

The next section will discuss the various constraints on this project, how they impact possible implementations, and how they were addressed.

F. Constraints

The main constraints on this project are power consumption, complexity of operation, board size, reserved spaces, and number of free MCU pins. Of these constraints, power consumption, complexity of operation, and board size come from the clients' needs. In contrast, reserved spaces and free MCU pins were both established in previous versions of the project. Each new version that adds new functionality will also use more pins on the MCU. Further, mounting holes and power connections have fixed positions on the PCB to minimize necessary changes to the housing between versions.

Power consumption is the most important constraint because long battery life is essential to the bottle's user experience (UX). Thus, the additions in this version of the project should require as little power as possible. For low-power wireless communication, there are few options: BLE, Wi-Fi, ANT, ZigBee, RF4CE, Nike+, IrDA, and NFC are all standard for wireless communication. However, Bluetooth and Wi-Fi are the most common and well-known technologies in consumer electronics [3] with both present in every new smart-device. Further, with the popularization of wearable and smart-home devices the average consumer is more likely to have experience connecting devices through both technologies. Further, none of the alternatives to Bluetooth or Wi-Fi can achieve the same power consumption per bit of information sent. Thus, six of the eight options can immediately be eliminated, as they would increase complexity of use and power consumption over Wi-Fi or BLE.

In terms of power consumption, BLE is clearly superior to Wi-Fi. On average, BLE consumes 72 µW [4] where Wi-Fi consumes 210 mW (the other 6 wireless communication technologies consume minimum 183μW). This massive difference stems partly from the difference in intended uses: Wi-Fi is designed for high throughput data transfer where BLE is designed for infrequent small bursts of data. When not transmitting, BLE modules enter a sleep mode, consuming no power until receiving an interrupt from their controller. The module is only out of sleep mode for seconds at a time in typical applications, meaning it consumes no power during the majority of the Smart Bottle's operation. This is how the BLE module achieves such low average power consumption.

BLE is also superior to Wi-Fi in that it is easier to use. Connecting two devices via Wi-Fi requires a both devices to connect to the same Wi-Fi network. This yields a few possibilities. The devices can both be connected to the same public Wi-Fi network while in use; but this introduces the possibility of third parties obtaining access to the Smart Bottle's data. Alternatively, both devices could be connected to the same private network while in use. To satisfy this, users could perform data transfer while connected to their home network, or the Smart Bottle could generate its own network for the user to access. One of these options constrain users to a specific location, which could corrupt data as discussed previously. The other option requires more knowledge and work from the user, which runs counter to project goals. Overall, Wi-Fi would introduce a minimum of one extra step of preparation before use as compared to BLE.

In terms of complexity and power consumption, BLE is clearly the superior wireless communication solution because it achieves minimal power consumption, it requires no new hardware to facilitate connection, and it is the technology with which the public is most familiar. For these reasons we chose BLE as the wireless communication technology for this project.

While power consumption restricts what technologies can be used in this project, space restricts the amount of functionality the project can include. There is little space for a PCB in the

Smart Bottle housing because it has a similar diameter to the bottom of a baby bottle. Specifically, the interior of the housing is 6.5 centimeters (cm) in diameter. This limits the size of the PCB to 6 cm of diameter to leave space for connections to power and measurement instruments. Designing a board of this size is made simplest by choosing small packages for passive components. Further, these components should be surface mount to avoid through-hole components taking up space on both sides of the PCB. Not all components come in a variety of packages, however. Power connectors and important data-management modules are fixed sizes, thus further restricting the space left for passive components and traces.

To address space constraints, we first expanded the PCB. The previous version of the Smart Bottle utilized a PCB with a diameter of 5 cm. Visual inspection made it clear that fitting the BLE module on such a small board would be highly challenging, especially when considering the "keep out" zone around the BLE module's antenna. Even after this expansion, free space is rare. Thus, the first components we placed were those with fixed positions relative to the center of the board. It was key to ensure that the mounting holes especially did not move or conflict with other components. However, once they were in place, routing traces around and through their footprints was simple.

Given the size of the board prior to its expansion, previous versions utilized small footprints for passive components. All passive components (besides three resistors) are either 0805 or 0603 packages. There are smaller packages available, but these are the smallest we believed we could accurately hand-place. These components took up little space in previous versions of the Smart Bottle, so no changes were made to component packages.

III. Methods

A. Design

With these constraints in mind, there were few choices to make for this project. After fixing issues from the previous version, the focus of my work was adding Bluetooth functionality. The previous team had already chosen the nrf51822 as the BLE module to be used as it comes with an antenna and an MCU built in. Further, the module's built in MCU uses the same architecture of the main MCU on the board meaning that both can be programmed with the same equipment and software. Third, the nrf51822 is sold by Adafruit on a breakout board to facilitate testing prior to design. Adafruit is also the company that makes many of the ARM M0-based microcontroller boards that are compatible with Arduino. This means that Adafruit's bootloader and flashing software work with both the MCU and the BLE module which greatly simplifies the programming process. I found several alternatives to the nrf51822, but none were significantly better in aspects crucial to the project. Thus, the project continued with the nrf51822 given the lack of better options and the fact that we had extra units from previous iterations.

The next choice to make was between communication protocols. The nrf51822 (henceforth "BLE module") was capable of I2C and SPI communication. However, it could only use four wire I2C because its data pins were unidirectional. This meant that communication with the MCU would require four wires irrespective of communication protocol. Inspection of the Smart Bottle schematic revealed that the SD card reader communicates with the MCU via SPI. This means that using SPI for the BLE module as well would only require the use of one more pin on the MCU for the chip-select line where I2C would need four. The MCU does use I2C to communicate with the measurement devices. However, the MCU must take data in and process it before sending it to the SD card. Thus, placing the BLE module on the same communication network with the SD card reader means that both modules receive the processed data at the same time with the fewest commands and MCU pins possible. Thus, the BLE will communicate with the MCU over SPI. The choice of BLE-MCU communication protocol was the last prerequisite to finalizing the schematic.

Adafruit provides the below example configuration for the BLE module which required minimal adjustment to adapt for this project. The main changes were the allocation of 4 digital I/O pins as SPI pins, the removal of the crystal oscillator, and the removal of the diode on the CS pin. Then the software debugging (SWD) pins were connected to the board-to-board (B2B) connector via 0Ω resistors. Zero-ohm resistors were also inserted between the MCU's SWD pins and the B2B connector. These resistors allow the B2B connector to connect to the BLE module or the MCU instead of using a separate B2B connector for each module. To change which module the B2B connects to, simply bridge the three 0 Ω resistors that correspond to the desired module and ensure that the other three 0 Ω resistors are left open. These connections are not identical between the MCU and the BLE module because the BLE module has a RESET pin where the MCU has a !RESET (read "not reset") pin. This was addressed by inserting an inverter between the 0 Ω resistor and the BLE's RESET pin.

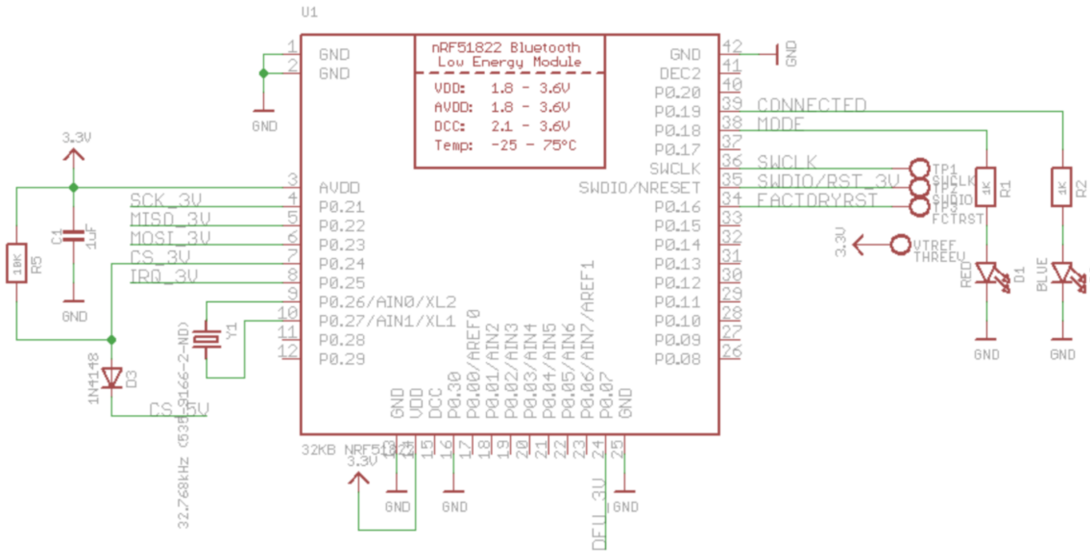


Fig. 1. Adafruit’s example BLE schematic

Finally, testing previous versions of the Smart Bottle showed that the mode-indicator LED was not bright enough. To remedy this, the singular through-hole common-cathode (CC) RGB LED was changed to 4 surface-mounted CC RGB LEDs around the circumference of the PCB. This change necessitated recalculation of the current-limiting resistors which was simple with the information provided by the LED datasheet [5].

Once the RGB LEDs were connected in with the proper current-limiting resistors, the schematic was complete. From there, designing the PCB itself was simple. The components with fixed positions were placed first. Then, the components with the most connections were placed close to the center of the board, and those with fewer connections were placed near the components that dominated their subsystems. After that, the traces were autorouted to ensure that the component locations were workable. From there, I manually routed each trace and manually placed each via. This was not necessary, but I chose to do it to learn what mistakes I may have made in placing components. Once the traces were routed, the design and electrical rules checks were run in a final search for potential errors. Neither check returned anything of note meaning that the design was finished.

B. Manufacture

With the design finished, the CAM files for the PCB and the solder stencils were sent to OSH Park for manufacturing. While these were being manufactured, all the necessary components were found and ordered from Mouser, Adafruit, DigiKey, and AliExpress. It would have been convenient to order all the parts from one site, but no site could be found with all the components listed and in stock.

Once the PCBs, stencils, and components all arrived, assembly began. Beginning with the layer that Eagle designated as the top, the board was secured in place and the stencil was aligned with the metal pads. This process was difficult due to the small size of the pads on the MCU and the accelerometer. Once the stencil was aligned, it too was secured in place to ensure it stayed stationary during application of solder. From there, solder paste was applied to the larger pads and spread to the smaller pads with a solder spreader. When every pad had solder paste on it, the stencil was carefully removed to disturb as little solder paste as possible. Nevertheless, removal of the stencil spread paste between the smaller pads. This excess paste could bridge important pads in the oven, so it was removed with the dull side of a razor blade under a microscope.

Once it was clear that no pads were bridged, components were placed on the board. Passive components were simple to place as their directionality was irrelevant. However, components including the MCU and had multiple possible orientations given the layout of their pads. Their pinouts were cross-referenced with the Eagle board file to determine the correct orientation before their placement. Once each component was placed on its corresponding pads in the correct orientation, the board was placed into the reflow oven and the oven was started.

When the oven finished the top side, the board was removed. The process from the top side would be replicated on the bottom side. However, the board would no longer lay flat for solder paste to be applied to the other side because the newly soldered components were now on the bottom, and their heights varied. To remedy this, holes were cut in the cardboard to fit the two tallest components: the B2B connector and the volume sensor connector. With these holes below the board, it could sit flat on the cardboard. Thus, the board and stencil were secured to the cardboard, and solder paste was applied.

However, before components were placed on the second side, the components on the first side had to be addressed. The second time through the oven, the board would have to be placed upside down with all the previously soldered components on the bottom so that the unsoldered components would not fall off. However, the oven would melt the solder on both sides of the board. This meant that previously soldered components could fall off while in the oven if they were heavy enough to break the surface tension of molten solder. To prevent this, a small sheet of tin foil was wrapped over the previously soldered components with only small corners touching the unsoldered side to hold the foil in place. With the foil covering the bottom, components were placed on the top. Then, the board was put back into the oven and the oven was started again.

Once the board was removed from the oven a second time, the first side was immediately inspected to ensure nothing had fallen off. Only one pin on one component detached from the board: the leftmost pin on the volume sensor connector. This pin was hand-soldered back to its pad, finalizing the board.

IV. Results

A. Hardware

With the board assembled, an error immediately became obvious: the RGB LED footprints had an incorrect pinout. The Eagle libraries used in creating the schematic did not have CC RGB LEDs. Thus, the surface mounted RGB LEDs were common anode (CA) in the schematic. I had assumed that CC and CA LEDs would have the same pinout because they used the same footprint, so I neglected to check the datasheets to make sure. This error did not significantly impact the functionality of the Smart Bottle, but the LEDs were the only source of feedback from the MCU to the user. Therefore, some wire was used to connect the pins on an LED to the correct pads of the footprint. This was a sufficient solution to get feedback from the board for testing, but it was an error that could not stand for future versions. Thus, a CC RGB LED footprint was created using the existing CA footprint, and it replaced the CA LEDs in the schematic.

With a source of feedback acquired, the first piece to test was the MCU. Once the necessary programs were installed, the 0 Ω resistors corresponding to the MCU were bridged, and the flashing process began. However, Microchip Studios reported that the reference voltage on the board was 1.2V as opposed to the desired 3.3V. Measuring voltage at the V_{CC} node yielded the same result. Eventually, through more voltage measurements and cross-referencing the schematic, the issue was revealed: the voltage regulator controlling the V_{CC} node was outputting 1.2V. Upon inspecting the datasheet, I found that I had ordered the wrong regulators. This was simple to fix as we had extra units from previous versions. Once I exchanged the regulator on the board for the correct unit, V_{CC} was measured at 3.3V as desired. From there, flashing commenced once again.

B. Software

After setting up the software, the MCU was successfully flashed with its bootloader. This indicated that the B2B connector was correctly implemented, which meant that flashing the BLE module would be just as simple. Once the 0 Ω resistors were changed to connect the B2B to the BLE module, the BLE firmware was successfully flashed without issue. At this point, everything was ready to begin testing the functionality of various subsystems using Arduino code. With the bootloader on the MCU, the B2B connector was no longer necessary to upload code. Instead, all code could be uploaded from Arduino through the microUSB port.

As the only source of feedback from the MCU, the RGB LED was the most important component to verify first. The simple test script turned on the green and blue parts of the LED without issue, but the red part remained off. The pins in the code were correct, so this appeared to be a hardware issue. After probing for continuity and voltage at every node related to the red part of the LED, no errors were found. Now, the Arduino IDE requires users to specify what proprietary board they have connected. Given that the Smart Bottle uses a custom PCB, I had set Arduino to search for a Feather M0 Express board as it uses the same MCU (ATSAMD21G18A) as the Smart Bottle. This allowed Arduino to program the board, but it could have been restricting functionality behind the scenes. I searched the internet for anyone who had similar problems, and that led me to the "variants" folder that Arduino installs. This folder contained a sub-folder for the Feather M0 Express, which in turn contained a variant.cpp file. This file held all information regarding each pin on the MCU, their corresponding attributes, and their pin number in Arduino. In this file, I found that the pin connected to the red LED was not set as an analog pin or connected to a pulse-width modulation circuit. This meant that the analogWrite() function used in the testing script would not work on the red LED. After changing this pin to be digital and changing the test script to use digitalWrite() for the red LED, the test script finally turned all three colors of the RGB LED on without issue.

With the LED working, the next step was to test BLE data transmission. The simple test script for this would read user input from the serial monitor in Arduino and transmit it to a connected device over Bluetooth. Adafruit's "Bluefruit Connect" app allows data reception and display on any smartphone, so I used my phone to receive and display data. Once the code was uploaded to the board, anything typed into the Arduino serial monitor appeared in the app and vice versa. This meant that BLE data transmission was functional, leaving only the measurement systems to test.

The accelerometer did not function in the previous version, and it was hypothesized that this was due to an incorrect power connection. It had functioned in older versions, making this hypothesis seem likely. Thus, this was the first measurement system to test. Running a test script from previous versions of the project only produced zeros on the serial monitor. This indicated that the accelerometer might still be nonfunctional. However, this was because the script was not configured to test the accelerometer. Once the accelerometer initialization code was added, the serial monitor showed that the accelerometer was returning data that changed based on orientation of the board. Quality of this data is reliant on the quality of the module, not the quality of the Smart Bottle PCB because the only connections to the accelerometer are for power and data transfer. Given that the module returned data, it must have power and data transfer must be functional if not perfect. Further, no changes were made to the accelerometer in this version of the Smart Bottle save the adjustment of the power connection. This means that data transfer should function identically to older versions from which usable data was collected. Thus, in the scope of this project the accelerometer functions properly.

C. Calibration

The other measurement system to test was the volume sensor that monitors the amount a baby eats. Each sensor is different, so they must be individually calibrated to ensure accuracy. The sensor outputs a voltage level which must be converted through an offset and a calibration factor into a real volume. The goal of calibration is to put a known volume of fluid into the bottle and receive a measurement from the sensor that is within 1ml of the correct volume. The eventual goal is to be accurate within 0.1ml.

The first test was to gather data with a calibration factor of 1 and an offset of 0 to establish a baseline. All test data was output to the SD card for easy analysis. A graph of the results of this test can be found in figure 2 below. The Arduino library for the sensor has functions to calibrate the sensor using a known volume which was used in the second test. After the second test, the file on the SD card reported a calibration factor of -2053.41 and an offset of -147302. The third test utilized these values via the set_calibration() and set_offset() commands, but the data it returned was highly incorrect. Subsequent tests involved setting the offset back to 0 and modifying the calibration factor in between tests to increase accuracy. After many iterations, a calibration factor of -1903.41 returned data accurate within 0.5ml for every known volume tested (results pictured in figure 3 below). This was well within the wider margin of error. Thus, several more tests were performed to achieve the more stringent margin of error. Unfortunately, the volume sensor reads between -0.2ml and 0.2ml with no liquid present. Thus, despite thorough testing, the lower margin of error was not achieved. While this is not ideal, the sensor is within tolerance. Further, the graph of calibrated outputs reveals the slope of the trend line and the R^2 value are 1. This shows measurements are highly accurate and that the trendline fits the data points. While the slope and R^2 value might decrease with more datapoints, their high accuracy is promising regardless.

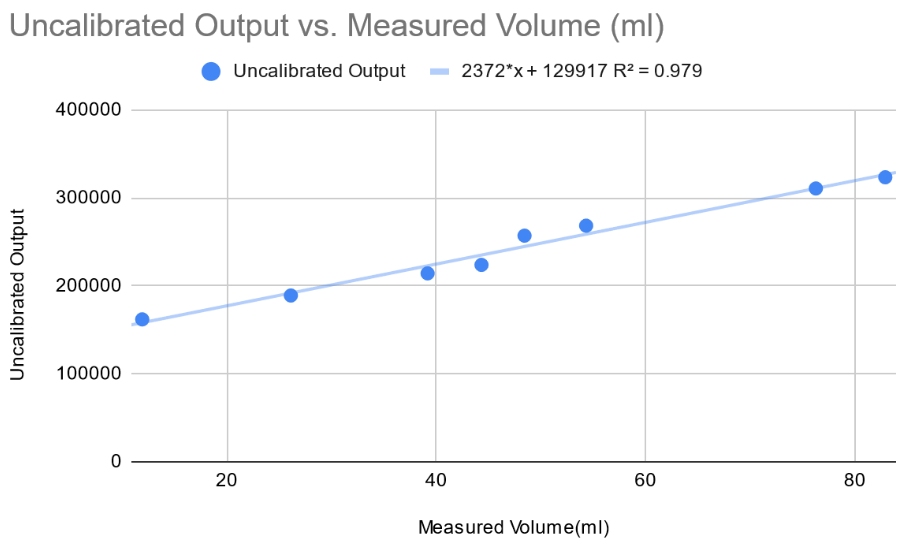


Figure 2: Uncalibrated Volume Sensor Outputs for Known Volumes

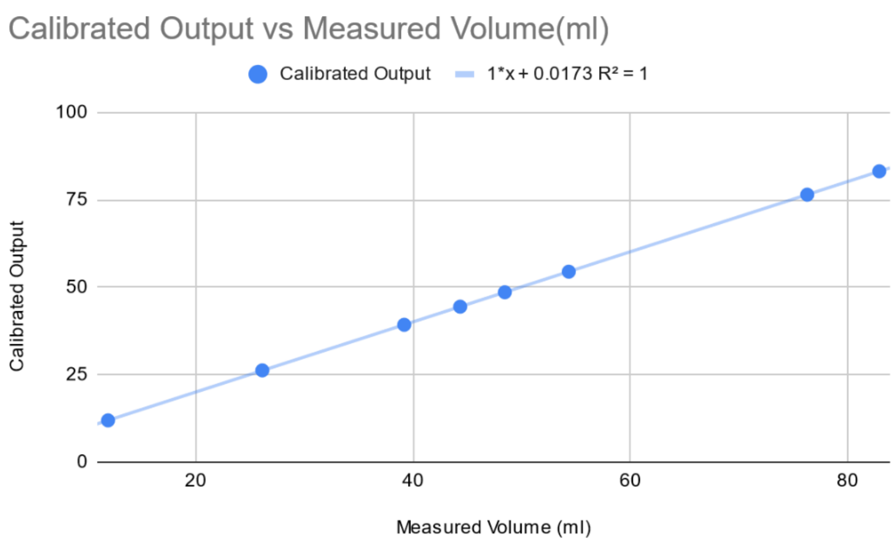


Figure 3: Calibrated Volume Sensor Outputs for Known Volumes

D. Requirements, Specifications, and Results

In the "Necessary Functionality" subsection of the introduction and background, the main functions of the Smart Bottle were defined as differentiating between feeding and idle, measuring length of feeding events, measuring amount eaten, storing data, allowing reprogramming, and remaining idle for long periods. Each of these functions save the last can be connected to a single module on the PCB.

The Smart Bottle utilizes a neural network to process accelerometer data to differentiate between feeding and idle states. While this does require training the neural network, the only hardware it requires is the accelerometer and the MCU. If the accelerometer is properly connected and powered and the MCU can be flashed with code, then the hardware aspect of this function works. As discussed in the "Hardware" subsection of "Results," the accelerometer outputs data that changes as the board moves. The accelerometer functioned in older versions of the project, and the only change made to it in this version was the correction of a power connection. Thus, the accelerometer functions as it did in old versions in which it was verified (excluding the version immediately before the addition of BLE). Further, the MCU was demonstrated to accept new code in each test. Thus, training the neural network is the last task to ensure the Smart Bottle can perform this main function. Given that the neural network is outside the scope of this project, differentiation between feeding and idle states is complete for this version of the PCB.

Changing the mode of the bottle is necessary, but it does not yield much information unless the duration and time of the feeding mode are tracked. This data gives researchers more context to each feeding event which could yield valuable insight. The MCU requests date and time data from the RTC at the beginning and end of each feeding event as determined by the neural network. Storing this information, then, only requires the MCU sending said data to the SD card. The SD card was fully functional in the last version of the Smart Bottle, and no changes were made to it in this version. Moreover, it has been shown to work in the testing discussed in the "Results" section previously. While the RTC was nonfunctional in the last version, it worked in previous versions with no issue. Once again, the only change to this system in this version was the correction of an erroneous power connection. Thus, it should work just as it used to. This was found to be true in testing when the RTC continually output correct times in each test after only being set in the very first test. No direct testing of the RTC was performed because its data was output with data from other subsystems during previous tests and the times were always found to be correct within 3 seconds. Some of this error can be attributed to human error in crosschecking RTC data and time.gov. Regardless of the source of the error, this was well within the engineering specification of 10 seconds, so the Smart Bottle can successfully track time and duration of feeding events.

Tracking the amount of formula eaten during a feeding event is done via a volume sensor and relies on the first main function of the Smart Bottle. As discussed in the previous "Calibration" subsection, the volume sensor initially returned unusable data, even after automatic calibration. However, after repeatedly adjusting the calibration factor, volumes between 10 and 85ml were measured with less than 0.5ml of error on each measurement. While the desired margin of error was 0.1ml, 0.5ml is still within the larger margin of error of 1ml. As long as the Smart Bottle is properly initialized after it is fully assembled but not yet full of formula, the volume obtains correct data. Thus, the Smart Bottle is capable of accurately tracking amount eaten.

As was seen throughout volume sensor testing, SD card data storage works as intended. The SD card reader functioned to specification in the previous version of the project; no changes were made to this subsystem from the previous version. Further, all the volume sensor calibration data was stored on the SD card and now the volume sensor works. Calibration to the accuracy achieved in testing would not be possible if the SD card reader was nonfunctional as this testing must be done with the Smart Bottle fully assembled which prevents the output of data to Arduino's serial monitor. While it would have been possible to output the data instead to the BLE module to read from a smartphone, this would have made data analysis significantly more cumbersome and less accurate than it was via .csv files on the SD card. These factors demonstrate that the SD card reader functions to specification in this version as it did in the previous one.

Reprogramming the Smart Bottle relies only on the microUSB port as the initial flashing via the board-to-board connector only needs to be done once. The ability to reprogram the board was necessary for performing any software testing or calibration. Given that both phases of testing were successful, the microUSB port must be functional. Further, no changes were made to the microUSB port or its connections between the previous version of the project and this one. Thus, since the previous version was fully capable of being reprogrammed, this version is as well.

Finally, the last necessary function of the Smart Bottle is long battery life. Power for major active components like the accelerometer and BLE module is controlled by the MCU. When the MCU enters low power mode it disables power to these other systems and it reenables them upon leaving low power mode. Thus, in low power mode the bottle consumes a maximum of 12.8uA for the MCU[6] and 3.5uA for the RTC [7] as stated in their respective datasheets. However, both modules draw power from separate batteries. The LiPo is rated for 400mAh at 3.7 volts where a CR1025 coin cell is rated for 30mAh at 3.3 volts. From these figures, the Smart Bottle can stay in low power mode for 1460 days before the MCU drains the LiPo, and 357 days before the RTC drains the coin cell. These numbers assume no other power draw from the LiPo including leakage current, which is not feasible. However, even tripling the power consumption of the MCU in low power mode to account for power draw from components like the voltage regular that will always be on, the LiPo still lasts about 487 days in low power mode. Further, only the RTC draws power from the coin cell. Thus, its battery life will not vary much. Given that the Smart Bottle spends most of the time in low power mode, its batteries lasts weeks on a single charge, even if power draw in its active state is twenty times what it is in low power mode.

All six of the necessary functions of the Smart Bottle have been verified in testing and are functional. Have the engineering specifications been met as well? There is much overlap between necessary functionality and engineering specifications. Specifications for battery life, measurement accuracy, and standard data storage have all been discussed in the discussion of necessary functionality. That leaves two specifications to consider: number of steps needed to turn the Smart Bottle on, and number of steps required to access the data.

Turning the Smart Bottle on and waking it from low power mode are not the same. Turning the bottle on requires opening the housing and plugging in the LiPo battery; that’s two steps. Waking the bottle, which from a parent’s point of view could be seen as turning it on, only requires lifting the bottle off of a flat surface (provided it is turned on). It should take no more than two steps for users to turn on or wake the bottle to ensure that the device is simple to use, and this has been achieved.

Data can be accessed in one of two ways: from the SD card or from the BLE module. Both methods have three steps. To access data via the SD card, the housing must be disassembled by unscrewing the top, the SD card must be pressed to trigger the release mechanism, and the SD card must be inserted into a computer. To access data via the BLE module, one must connect to the Smart Bottle from their phone, request the data, and open the downloaded file on their phone. However, while the hardware is fully capable of this, BLE file transfer is not yet implemented in the software. Thus, the only method currently available to access the data is through the SD card in three steps. This still meets the specification of three or fewer steps to access data.

That concludes the discussion of results of testing. Each necessary function is implemented and working, and all of the engineering specifications derived from customer requirements have been met. Thus, the next section will conclude this report by discussing how the Smart Bottle meets the needs of its clients and how it will facilitate collection of uncorrupted data.

V. Conclusion

Before the Smart Bottle project began, several nonintrusive data collection methods existed or were hypothesized alongside the Smart Bottle. The standard method prior to this project was to weigh infants in a controlled setting before and after feeding. However, feeding in a clinical setting and weighing the infant could both heavily influence infant feeding behaviors.

Another approach involves parents tracking their infant's feeding habits in a diary. This is less likely to influence infant behavior as it does not need to change their feeding environment. However, infants can feed between eight and twelve times per day including during the night or during work when parents are distracted, tired, or otherwise preoccupied. Parents recording incorrect data or forgetting to record data can corrupt data changing apparent patterns [8], [9].

A third approach also involved a baby bottle monitoring system similar to the Smart Bottle, but one that measures suction instead of volume. This is the most promising of the three approaches, but it requires the use of an unfamiliar bottle and nipple which could disturb the infants normal feeding patterns. The goal of the Smart Bottle project is unintrusive data collection to avoid corrupting data on collection. Each of the three methods above change the environment that the infant is used to in order to collect data, thus possibly influencing how the infant eats [10], [11]. This bottle system also takes pictures of the infant at constant intervals for analysis of suck count, burst duration, and pauses in sucking. However, this method is not practical for assessment of many meals, or large groups, or for studies conducted at home.

The Smart Bottle system improves upon previous instrumented infant feeding bottles as it is an attachment instead of a bottle. This means that it can be attached to a bottle and nipple that an infant is used to eating from to minimize changes to the infant's environment. Further, The Smart Bottle also improved upon older methods by eliminating the need to change feeding location, remember the data, or analyze video of feeding events. The Smart Bottle takes feeding volume, duration, time, and angle data and stores it all on the SD card. Further, in testing previous versions the Smart Bottle was found to detect "on table" and "feeding" events with 99.3% and 99.8% accuracy respectively [12]. Margins of error for collected data were discussed in previous sections, and all collected data fell within these margins after calibration. Thus, the Smart Bottle automatically collects accurate infant feeding data in a manner that is less intrusive or constrained than previous methods.

While this is the first version of the Smart Bottle that performs all its intended functions, it is not finished. Therefore, some suggestions for future improvements are listed below.

1. Add "Reset" and "Initialize Volume Sensor" buttons to the outside of the housing.
2. Create an Arduino variant.cpp file specifically for the Smart Bottle.
3. Create a system to automatically calibrate the volume sensor with at least 5 known volumes.

VI. References

[1] R. J. R. Levesque, "Obesity and Overweight," in *Encyclopedia of Adolescence*, New York, NY: Springer New York, 2011, pp. 1913–1915. doi: 10.1007/978-1-4419-1695-2_447.

[2] "iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope," Oct. 2018. Accessed: Jun. 07, 2022. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm6dsox.pdf

[3] Behr Tech, "6 Leading Types of IoT Wireless Tech and Their Best Use Cases," *BehrTech Blog*, Jan. 05, 2021. https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/ (accessed Jun. 07, 2022).

[4] P. Mannion, "Comparing Low-Power Wireless Technologies (Part 1) ," Oct. 26, 2017. https://www.digikey.com/en/articles/comparing-low-power-wireless-technologies (accessed May 19, 2022).

[5] Lite-On Technologies, "SMD LED Product Data Sheet LTST-N683GBEW," Jul. 2016.

[6] Microchip Studios, "Low-Power, 32-bit Cortex-M0+ MCU with Advanced Analog and PWM Features," 2021. Accessed: Jun. 07, 2022. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/SAM_D21_DA1_Family_DataSheet_DS40001882F.pdf

[7] Maxim Integrated, "Extremely Accurate I2C-Integrated RTC/TCXO/Crystal." Accessed: Jun. 07, 2022. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS3231.pdf

[8] K. M. Hurley, M. M. Black, B. C. Merry, and L. E. Caulfield, "Maternal mental health and infant dietary patterns in a statewide sample of Maryland WIC participants," *Maternal & Child Nutrition*, vol. 11, no. 2, pp. 229–239, Apr. 2015, doi: 10.1111/mcn.12004.

[9] J. O. Fisher *et al.*, "Overestimation of infant and toddler energy intake by 24-h recall compared with weighed food records," *The American Journal of Clinical Nutrition*, vol. 88, no. 2, pp. 407–415, Aug. 2008, doi: 10.1093/ajcn/88.2.407.

[10] E. Sazonov, M. H. Imtiaz, J. Bahorski, C. R. Schneider, and P. Chandler-Laney, "Design and Testing of an Instrumented Infant Feeding Bottle," in *2018 IEEE SENSORS*, Oct. 2018, pp. 1–4. doi: 10.1109/ICSENS.2018.8589888.

[11] G. Capilouto, T. Cunningham, D. Mullineaux, E. Tamilia, C. Papadelis, and P. Giannone, "Quantifying Neonatal Sucking Performance: Promise of New Methods," *Seminars in Speech and Language*, vol. 38, no. 02, pp. 147–158, Mar. 2017, doi: 10.1055/s-0037-1599112.

[12] J. Guan, R. Brewster, J. de La Fuente, A. K. Ventura, and B. G. Hawkins, "Artificial Neural Network for Identification of Infant Feeding Tracking Using the Smart Bottle

System," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Nov. 2021, pp. 7539–7543. doi: 10.1109/EMBC46164.2021.9629536.

APPENDIX A — ANALYSIS OF SENIOR PROJECT DESIGN

Please provide the following information regarding your Senior Project and submit to your advisor along with your final report. Attach additional sheets, for your response to the questions below.

Project Title: Smart Bottle BLE Integration

Student's Name: Joshua Rizzolo

Student's Signature: Joshua Rizzolo

Advisor's Name: Benjamin G. Hawkins

Advisor's Initials: B.G.H.

Date: 06/09/22

Summary of Functional Requirements

Describe the overall capabilities or functions of your project or design. Describe what your project does. (Do *not* describe how you designed it).

The Smart Bottle records data on infant feeding patterns including amount eaten, bottle orientation, feeding duration, and time of feeding. It also automatically differentiates between "feeding" and "on table" states with a neural network and an accelerometer. In the "on table" state, the system is in low-power mode awaiting an interrupt from the neural network to turn back on and enter "feeding" mode in which it performs data acquisition. All data is recorded on an SD card for future analysis by researchers. Data can also be transmitted via bluetooth to a user's phone or computer should parents want feedback on their infant.

Primary Constraints

Describe significant challenges or difficulties associated with your project or implementation. For example, what were limiting factors, or other issues that impacted your approach? What made your project difficult? What parameters or specifications limited your options or directed your approach?

Major limiting factors of this project included size, free board area, low power consumption, complexity of operation, and free MCU pins. The housing for the Smart Bottle PCB is only 6.5-7cm in diameter internally, which limits the radius of the PCB. Further, this project focussed on integrating BLE capability into an existing system. This meant that the PCB had little room left for the BLE module due to the space taken up by previous modules. Some components from previous versions had to stay in the same place on the PCB relative to the center, meaning that nothing new could be placed in those areas. Each module that existed in previous versions of this project also already took up MCU pins, leaving only 5 free for this project. The BLE module only required one additional MCU pin, but not all of the free pins could perform the same functionality, further limiting available pins. The addition of wireless communication could use lots of power, so we chose BLE to keep the average battery life of the system as long as possible. Finally, the system will be used predominantly by parents of infants, so the device must be simple to operate so that parents don’t miss a feeding event due to technical difficulties.

Economic

• **What economic impacts result? Consider**:

Human Capital – What people do.

Financial Capital – Monetary instruments.

Manufactured or Real Capital – Made by people and their tools.

Natural Capital – The Earth's resources and bio-capacity.

The Smart Bottle is manufactured capital given that I am previous teams made it. However, it will not be sold, lowering its direct financial impacts on the economy. The important financial impacts that the Smart Bottle will have on the economy are the potential changes in the market for infant formula if parents discover that their children are eating too much or too little. Similarly, its impact on the environment directly will be low (though certainly present) because it is a research tool that will not be massively manufactured or distributed. It is likely that no more than 20 Smart Bottle attachments will be created. However, once again the indirect impact that the device may have through the research it facilitates could be much larger. For example, if parents find out that their children are eating far too much food, there could be a large-scale reduction in demand for infant formula. This could result in much formula being wasted in the event that there is a surplus, which directly impacts the environment by using resources to make formula that will be thrown away to decompose slowly.

The largest economic impact I see this having is on human capital. The Smart Bottle is a tool to facilitate research, or the acquisition of knowledge. This betters our understanding of the world and allows us to establish better feeding habits for our children, making them healthier in the future.

• **When and where do costs and benefits accrue throughout the project's lifecycle**?

Most of the costs of the Smart Bottle accrue at the beginning and end of its lifecycle. At the beginning, the devices cost resources and time to manufacture and calibrate. At the end of their life cycle, the devices become waste and must be disposed of. Disposal of waste is a main source of pollution in our world, so disposal of the devices after the end of their lifecycles costs environmental health. The only cost of the Smart Bottle that accrues during the use phase of its life cycle is its electricity consumption.

In contrast, the benefits of the Smart Bottle mostly accrue during the use phase of its life. The device benefits humanity by furthering our knowledge and thus possibly improving average health. Research on such an important topic requires lots of data, so preliminary results will likely come out while data is still being collected. Further, in some sense the data itself is the direct benefit that the device provides, as analysis on preexisting data can be done with or without the device. Data collection counts as using the device, thus most of the benefit of the device occurs while it is in use.

• **What inputs does the experiment require? How much does the project cost? Who pays**?

Original estimated cost of component parts (as of the start of your project).

$35.20

Actual final cost of component parts (at the end of your project)

$226.02

Resistors			
Value	Quantity	805	603
36.5	3	x	
100	2		x
1k	3	x	
2k	1	x	
8.2k	1		x
10k	5	x	
20k	1		x
100k	3	2	1

Capacitors				
Value	Quantity	805	603	1210
22pF	2		x	
0.1uF	7	1	6	
1uF	5	2	3	
10uF	7	4		3

Diodes & LEDs		
Color	Quantity	Package
Red	1	805
Blue	1	805
Orange	1	805
Schottky	1	SOD-123
CC RGB	4	P-LCC-4-3

ICs			
Type	Value	Quantity	Package
Inverter	74AHC1G04DBV	1	SOT23-5
RTC	DS3231	1	SO16W
MCU	ATSAMD21G18_QFN	1	TQFN48_7MM
Regulator	SPX3819-3.3	1	SOT23-5
Amplifier	HX711	1	SO16
BJTs	PNPMMBT4403	1	SOT23-3
Accel	LSM6DSOXTR	1	PQFN50P250X300X86-14N
Fixed Inductor	3.3uH	1	805
XTAL OSC	32.768kHz, 9pF	1	XTAL3215
LiPo Charger	MCP73831T-2ACI/OT	1	SOT23-5
BLE	MDBT40	1	MDBT40
Buttons	SPST_TACT-KMR2	2	KMR2

Connectors			
Type	Value	Quantity	Package
Load	CONN HEADER	1	SMD R/A 5POS 1MM
Coin Cell	CR1025	1	10mm SM Coin Cell Clip
micro USB	N/A		MICRO USB, RIGHT ANGLE, SURFACE
Board to Board	10 pin	1	SWD 0.05 Pitch Connector
LiPo	2-pin (2.54mm pitch)	1	JST PH 2pin
Micro SD Card	MICROSD	1	MICROSD

Total cost: $132.70

• **Additional equipment costs (any equipment needed for development?)**

Additional costs totaled $93.72 for testing equipment. Prior to PCB manufacturing, a breakout board version of each of the main three modules (SD Card reader, BLE module, MCU) were ordered. These were to test the physical configuration and to test our ability to wipe and reprogram the MCU and BLE modules with the debugger (Segger JLink Mini).

• **How much does the project earn? Who profits?**

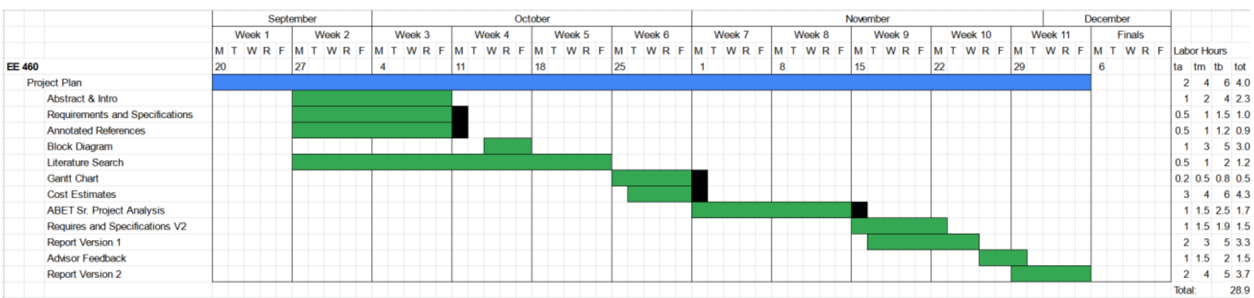
The Smart Bottle system is intended as a research tool for Cal Poly researchers. Thus, the project earns no money, and no one profits monetarily. All profit goes to researchers in the form of data.

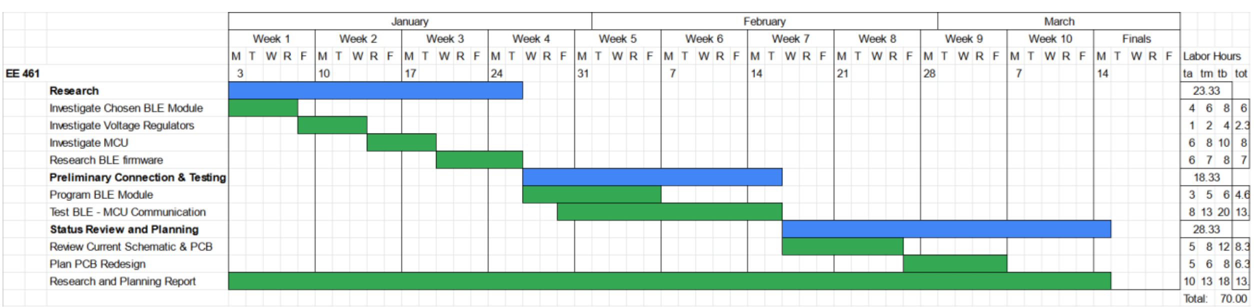
Timing

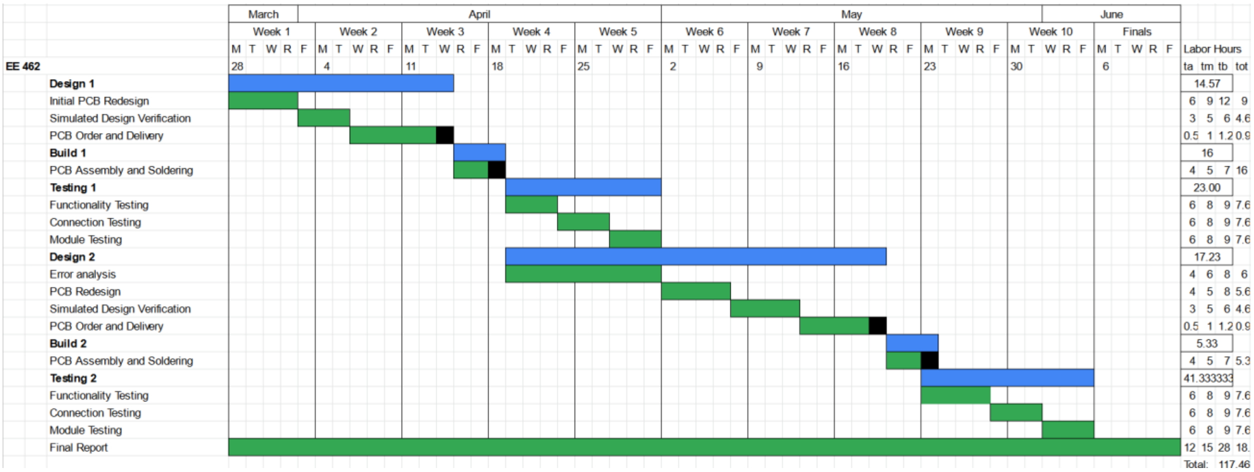
When do products emerge? How long do products exist? What maintenance or operation costs exist?

Products can begin emerging as soon as the small errors on the current version of the PCB are corrected. This would be in Fall of 2022 at the latest. Once created, the devices must be programmed, calibrated, and trained before they can be tested. This might take a day per bottle. Then, testing will take another day to ensure high accuracy. Finally, the devices can be used for research, which would likely last 12 months per child. However, it is unclear how long each bottle will continue to function. They would likely be in use for at least two rounds of data collection; assuming each round lasts the whole 12 months that a typical child relies solely on bottle or breastfeeding, each bottle would last about 732 days. In the whole course of this time, operations costs just amount to electricity needed to charge the device. Maintenance costs are unlikely to be high given the low-risk and low-contact environment that the bottle will be sharing with an infant. However, sensors are known to become less accurate with time; thus, these might need replacing if their lifecycle is longer than anticipated. Lastly, the flexible 3D-printed sleeve that attaches the PCB housing to a baby bottle may lose flexibility with time, thus needing to be refabricated.

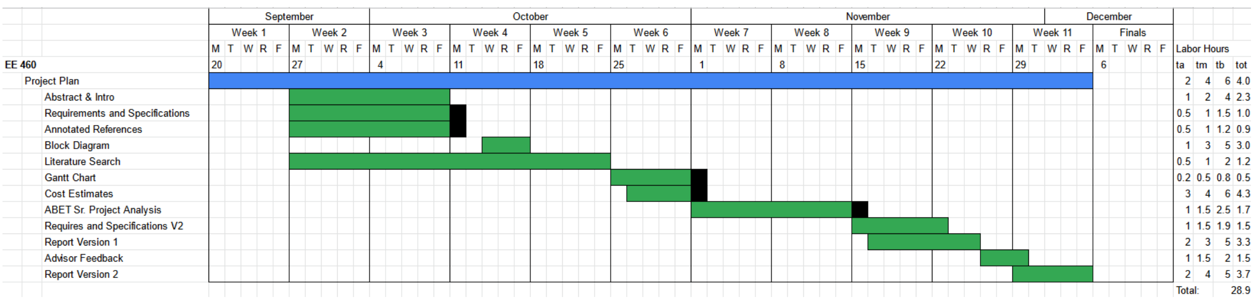
Original estimated development time (as of the start of your project), as Gantt chart

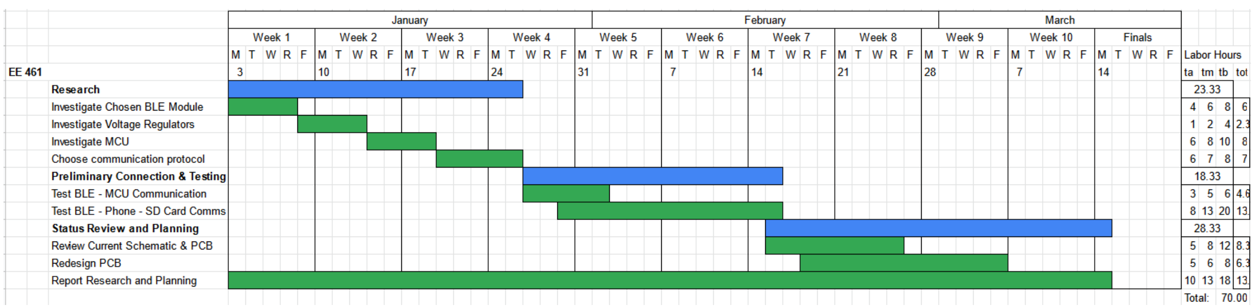


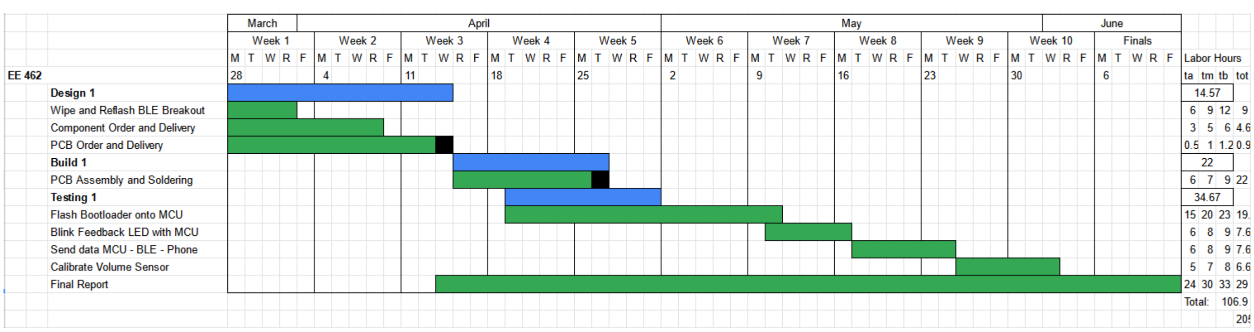




Actual development time (at the end of your project), as Gantt chart







What happens after the project ends?

Once I finish my changes to the PCB layout, a few PCBs will be manufactured, more components will be ordered, and Dr. Hawkins will assemble one or more to test. Once programming, calibration and testing are finished researchers could use the first generation of the fully functional Smart Bottles to gather data. There is certainly more functionality to add to the device; future students working with Dr. Hawkins will likely take on the project to make additions or fix problems as needed. But, after this version I believe the Smart Bottle is ready for its intended use.

If manufactured on a commercial basis: (N/A)

• Estimated number of devices sold per year

• Estimated manufacturing cost for each device

• Estimated purchase price for each device

• Estimated profit per year

• Estimated cost for user to operate device, per unit time (specify time interval)

Environmental

• **Describe any environmental impacts associated with manufacturing or use, explain where they occur and quantify**.

Use of the Smart Bottle should only impact the environment through its electricity use. It consumes extremely little power, but it will still draw power from San Luis Obispo or Cal Poly's grid. This impact will occur locally drawing power from fossil fuel plants and hydroelectric stations alike. As California progresses towards 100% renewable energy, the environmental impact of the use of these devices will decrease. Manufacturing impacts, however, will occur in

Oregon at the OSH Park factory and at the many locations which manufacture the various ICs, connectors, and passive components on the Smart Bottle PCB. Each of the components and boards also uses natural resources from all over the world. It is difficult to quantify the impacts of a project as small as this one. Using more of the Earth's resources and creating pollutants during manufacturing both certainly harm the planet. My best estimate is that this project has had an environmental impact comparable to buying several new electric RC cars.

• **Which natural resources and ecosystem services does the project use (in)directly?**

- Silica Sand, Limestone, Soda Ash, etc for Fiberglass for PCBs
- Copper for PCB traces
- Lead and Tin for Solder
- Rosin for Flux
- Whatever goes in transistors and ICs
- Nichrome for Resistors
- Metal and Ceramic for Capacitors
- Transport of pollutants away from factories (the wind)

• **Which natural resources and ecosystem services does the project improve or harm?**

The Smart Bottle is a tool to facilitate research on the connection between overeating in infants and later childhood obesity. Depending on the results of the research, this project could result in lower demand for infant formula. This would protect scarce water and the raw materials for infant formula.

The Smart Bottle project has contributed to global carbon emissions, thus harming the global ecosystem by contributing to climate change both in manufacturing and in use.

• How does the project impact other species?

Pollutants emitted in the course of manufacturing the Smart Bottle or in the production of energy to charge it harm the global ecosystem. Further, it also contributes to worldwide overuse of valuable resources. This project has no more direct impact on other species than this. It is possible that the device may facilitate protection of water resources by decreasing infant formula consumption, but this seems negligible.

• **Manufacturability**

The interior of the housing is small. This forces the PCB and components to be small as well. Accurately applying solder paste without bridging pads was difficult, as was placing the more complex components like the MCU. Finally, to prevent components from falling off the PCB on the second round through the reflow oven, the bottom of the PCB was wrapped in tin foil.

Sustainability

• **Describe any challenges associated with maintaining the completed device, or system.**

Care must be taken when opening and closing the housing as the power and data connections inside can obstruct the threads in the housing. If the wires are run over like this too many times, they could sever and require replacement. Care must also be taken in cleaning the housing of any liquid before opening it so as not to allow water to contact the active PCB. The flexible 3D printed sleave may also loosen over time and require replacement.

• Describe how the project impacts the sustainable use of resources.

Depending on the research findings, this project could diminish consumption of infant formula by informing parents about the proper amount to feed their children. This would save more water, and use fewer of the resources needed to make infant formula.

• **Describe any upgrades that would improve the design of the project.**

If a button to reset the system and a button to reinitialize the volume sensor were added to the outside of the housing, that would make the system significantly simpler to calibrate.

• **Describe any issues or challenges associated with upgrading the design.**

The upgrade proposed above would require feeding wires from into the housing. It would also require integrating two buttons into the 3D-printed housing. Any other proposed upgrades would likely add modules to an already tightly packed little PCB, which will make future component placement and trace routing more difficult.

Ethical

• **Describe ethical implications relating to the project's design, manufacture, use, or misuse.**

Misuse of the Smart Bottle could entail a parent or researcher causing a child to gain or lose weight instead of trying to keep them in the healthy range. If the PCB gets wet, that could also cause an electric shock. This would also constitute misuse of the device because it should not be opened around infants at all to avoid allowing them to bridge raw electrical connections. Either of these cases of misuse could endanger a child.

Health and Safety

• **Describe health/safety concerns associated with the project's design, manufacture, or use.**

Standard use of the device is completely safe for users as the housing is insulated and the system runs at low voltage and very low current. However, when the housing is open and the PCB is exposed, it should be kept away from infants to prevent them from bridging traces and burning or shocking themselves. Similarly, the exposed PCB should not be handled near water to avoid electrocution and destruction of the system.

Social and Political

• Describe social and political issues associated with design, manufacture, and use.

There is an obesity epidemic in the United States. Research into causes of obesity can help slow this epidemic and lead people towards healthier lives. Further, future research will cite or be inspired by current research into the subject, and research is the basis on which political policy should be made.

• Who does the project impact? Who are the direct and indirect stakeholders?

Direct: Dr. Hawkins, Dr. Ventura, Dr. Ventura's research students, Joshua Rizzolo, Paige Dolan

Indirect: Infants and their parents, infant formula manufacturers, biomedical engineering firms, Cal Poly EE & BMED departments

• How does the project benefit or harm various stakeholders?

It benefits all of its direct stakeholders by allowing other aspects of the project to move forward. This includes training of the neural network, data collection, and further calibration of the device.

Infants and their parents don't benefit immediately, but future groups will benefit from the new information yielded by the research that this project facilitates. Cal Poly's engineering department benefits with another project to display to prospective students and their families. Finally, the results of the research will determine whether the formula manufacturers and BMED firms benefit or are harmed. If a connection is found between early overeating and obesity, then formula manufacturers will be harmed and the BMED firms might look into making similar devices to give parents info about their child.

• To what extent do stakeholders benefit equally? Pay equally? Does the project create any inequities?

The Cal Poly BMED and EE departments pay for the whole project while only getting projects to discuss with prospective students. I get to pass senior design and, later, graduate for completing this project. Dr. Hawkins and Dr. Ventura receive a functional system which they can refine and use to collect data. I benefit the most from the completion of this project, but it also cost me the most hours of any group.

• Consider various stakeholders' locations, communities, access to resources, economic power, knowledge, skills, and political power.

Dr. Hawkins has all the power to decide the directions that the project will go and the tasks I need to complete. He is also certainly capable of completing the project with his knowledge, but Cal Poly requires me to display my knowledge through this project in order to graduate. Cal Poly can do this because it holds political and economic power in this scenario. I must pay to attend Cal Poly, and if I don't finish this project then I will not receive my degree thus precluding me from employment.

Development

• Describe any new tools or techniques, used for either development or analysis that you learned independently during the course of your project. Include a literature search.

This project gave me a much better understanding of microcontrollers and their interface with personal computers. I spent hours digging through Arduino configuration files for a specific board and editing the pin functions based on the datasheet and context clues to match the functions on our custom board. I had not even thought about the difficulties of interfacing

Arduino with a custom board before. I also learned a bit more about how many Arduino libraries are out there. I installed 6 separate ones for this project and tracking down documentation of some of their functions was frustrating.