FP(Key)A

Authors: Ben Sun, Korene Tu, Zhejia Yang
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—A keyboard with independently movable, programmable switches that can be configured into any layout the user desires.**

***Index Terms*—3D Printing, BLE, Mechanical Keyboard, Wireless Charging**

1 INTRODUCTION

Over the past 5 years, the mechanical keyboard market has exploded in popularity, with year over year growth of upwards of 10% since 2018 [2]. Such keyboards provide an outlet for self expression, with the user having the ability to customize the switches, keybindings, keycaps, and more to match their use case and aesthetic preferences. However, one area of customization that has remained prohibitively expensive to explore has always been the keyboard layout itself. Often times, mainstream keyboards and macropads are limited to the typical staggered QWERTY layout or a grid-like ortholinear layout. The only way to explore other, more ergonomic or project-specific layouts is to design your own or spend money on many different keyboards. With each keyboard possibly costing upwards of $200 factoring in switches, keycaps, and housing, many users are put off from exploring different layouts and simply learn to compromise with what they have.

To address this shortcoming in current keyboard offerings, our project is to create a keyboard whose keys are independently movable to any position the user desires. Each key on the keyboard is able to be programmed through a central configurator software which will remember the key assignments across devices. This project is aimed at video editors, streamers, gamers, and generally people who need flexibility in keyboard layouts due to using software where using different keys is more efficient or user-friendly than using a fixed staggered QWERTY keyboard.

Due to time constraints, we intend to create a proof of concept 16-key macropad. This will be sufficient to demonstrate the viability of BLE as a connection protocol between the different keys, the wireless charging capabilities, and individual configurability of the modules.

2 USE-CASE REQUIREMENTS

We have identified the following use case requirements:

1. Speed: Latency of less than 50 ms. Users such as gamers will need fast response times, and 50ms is what is typical for wireless gaming keyboards currently.
2. Longevity: Battery life of over 2 days per key. Users want their keyboards to work at least for the longest possible use sessions.
3. Expense: Cost of less than $300. Users would not pay for a customizable keyboard if they can buy two separate keyboards with different layouts for less.
4. Portability: weight of less than 1 pound. For users to want to carry the keyboard around, it needs to weigh around the same as current "portable" keyboards.
5. Convenience: mostly wireless operation. Users do not want to deal with the hassle of managing multiple wires on limited desk space.
6. Customizability: Independently adjustable keys. This is our most important requirement, as having the ability to configure the keyboard to any layout the user allows them to tailor their keyboard to the shape of their hands, reducing typing fatigue and increasing comfort. Additionally, having the ability to reconfigure the layout on the fly means that the user will need to buy fewer keyboards to try different layouts, potentially saving multiple keyboards from becoming e-waste due to user dissatisfaction.

3 ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

Provided below are the block diagrams for our keyswitch module and the wireless charging module. The keyswitch modules will each communicate with a central BLE receiver module (the controller) which is connected to the computer. The user will set keymaps by setting keybindings in the configurator software, and then flashing the resultant firmware to the controller board. Each keyswitch module will be able to be recharged on the wireless charging module, which will be able to recharge multiple keys at a time.

3.1 Keyswitch Module

The keyswitch module is the first half of our modular keyboard system. Each keyswitch module will contain a Seeed XIAO nrf52840 (hereon referred to as the Seeed) board which has BLE capabilities and an integraded BMS chip. This board enables us to keep track of battery levels and handle the recharging of the battery from the wireless charging inductive coil. The keyswitch modules will be identified uniquely by their bluetooth UUID, allowing them to be identified on the configurator. Each keyswitch module will have a custom designed 3D-printed enclosure which keeps everything together.

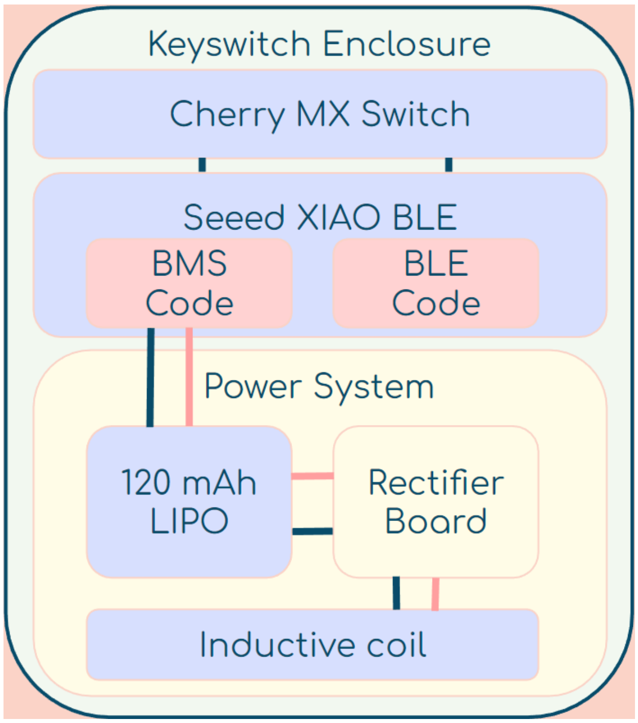

Figure 1: Keyswitch Module

3.2 Wireless Charging Module

The wireless charging module consists of multiple inductive pads with markers over them delineating to the user where to place the individual keyswitch modules to charge the keyboard. Each inductive pad will be an off the shelf coil with a corresponding coil located inside each keyswitch module. We will be using an **XKT-510** wireless transmission chip to transmit power to the switch modules, and a **T3168** wireless receiver board to receive power in the switch modules. This way, the user will be able to charge each keyswitch without much hassle or the need to plug in wires. Our goal is to have up to 4 keys being able to be charged simultaneously, but depending on parts availability, we may only be able to get a single wireless charging pad working as a proof of concept.

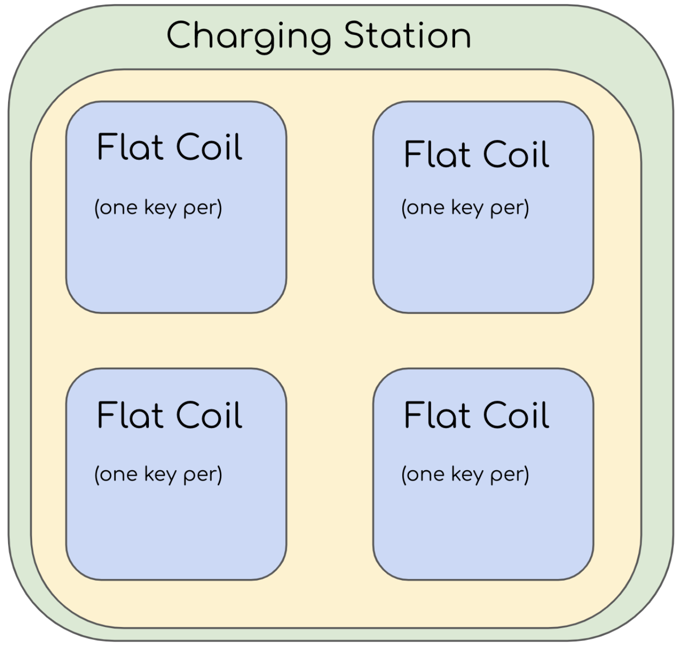

Figure 2: Charging Module

4 DESIGN REQUIREMENTS

We have identified the following design requirements as a result of our use case requirements

1. Dimensions of less than 22x22x22mm for the base (not including switch height). This requirement comes from the portability aspect, as if the keys themselves were any larger, then the keys would not be able to be placed side by side, thus limiting layout possibilities. The height limit comes from the convenience factor. We want the keys to be easily gripped, but not so tall that the typing experience is hampered by key wobble.

2. 100mAh battery capacity. This stems from the longevity requirement. Based on previous experience with LIPO batteries in wireless keyboards, 100mAh should be enough to last for 2 days of use without needing to be recharged, while being small enough to fit within the footprint of the dimensions [4].

3. BLE wireless capabilities. Based on the latency and customizability requirement, the keys must be able to communicate wirelessly to the central controller. BLE meets the requirements on latency and also has good library support with the Seeed boards that we are planning on using. Additionally, the LE (low energy) part of BLE will help with extending the battery life even more.

4. Key stability. For the convenience factor, we need the keys to be stable while mounted to the baseplate.

Any major wobble will detract from the user experience, and make it distracting to use.

5. Individual Key Configurability. For the customizability aspect, each key must be able to be moved anywhere on the baseplate, and have its keybinding changed to be a different key or a combination of keys.

5 DESIGN TRADE STUDIES

5.1 I^2C

Originally, we wanted to have each key module connect to each other via I^2C using contact strips on the side of the module. Each key would have it's own address and the central device would poll each key to determine its state. However, we ultimately decided against this approach as it limited customizability too much and could hamper the user experience. Having the requirement for every key to be touching every other key does not allow for as many layouts as if each individual key was independent of the rest. However I^2C would have offered the ability to more easily identify the keys, as each key's address would be known to the central controller, and the general programming would have been easier as there would be no need to deal with wireless communication protocols. Additionally, each keyswitch module would have been cheaper to produce since it would require no active components such as microcontrollers, antennas, or receivers.

5.2 Creating or Buying: Microcontrollers, Keys, Charging

A large part of our design trade studies is understanding what comes with designing your own electronics versus when to buy the product on the market. A large part of decided what parts to buy and what to design ourselves was quite difficult to organize. First we were unsure how we wanted to implement BLE into the keys and whether we wanted to make our own PCBs. Ben had found the Seeed Xiao board which is perfect for our use case of having a small but reliable and convenient to insert into 3D housing. Buying 16 key microcontrollers would be much easier than first creating test boards and then figuring out if they could transmit or not. Another few things on debating whether to buy or not were flat coil inductors as well, as Korene had found a method of making flat coils and suggested making them could save money, however Zhejia suggested back that mass produced inductors would be much more consistent and stable in what the inductance values would be and would have less room for error. While it may save money to create coils, it saves a lot more time to buy the inductors instead.
The microcontrollers for the keys to relay data to and then for the microcontroller to relay to the computer were also debated on whether to buy an ESP32 or not. However, the main argument is that because we had decided on 4 client BLE chips to collect data from the 16 keys, we decided it would be much easier to lay out a microcontroller to have 4 receiver boards and then connect them directly to a USB-C connection for ease. Finally, the situation that our team had debated for a long time was whether or not to create the charging circuit or buy an IC chip that follows standard charging protocols and has an ecosystem. While creating the charging circuit was possible, especially with 18-220 knowledge, there are many more things to consider in the circuit such as how a sine wave could be generated or how to properly increase the current to have a sufficient power delivery and be consistent in its charging to ensure the safety of the LiPo battery as well. In the end, we've decided on using the wireless transmission chips of the XKT-510 and T3168 where it has it's own ecosystem and is already small enough to our advantage and the use case requirement of the keys to be small.

5.3 Keys

We had shifted our original plan of making a Ten Key Less (TKL, no number pad) keyboard which would have 87 keys to a 32-key keyboard with only the letters and a few special keys like Enter, Tab, and Spacebar to a 16-key macro keyboard. While having a full keyboard with all of the keys used in daily life would be ideal and incredible for the number of possible usages there could be, we had narrowed down that for a full keyboard, most people would not need to constantly change 87 keys on the regular and there are already difficulties in whether our connections could work properly with the BLE modules which are reported to support 8 devices. As a whole, it is also much more cost efficient and productive to ensure a set few keys are able to work properly to also be proof of concept that a full keyboard could be made in the future. In addition with a macro keyboard, while it is fewer keys to use, it is tailored towards the specific market of keyboard enthusiasts who are more willing to spend more for a keyboard and open to novelty ideas such as the infinite keyboard layout. The board is tailored to a more specific audience who would find use cases of a macro pad, which is commonly used in gaming and shortcut configurations.

6 SYSTEM IMPLEMENTATION

6.1 Power

In order to supply power to our switches, as seen in 2, each keyswitch will have a 18mm x 14mm x 7mm (length x width x height) rechargable 100mAh LIPO battery which acts as the power supply for the **Seeed Studio XIAO nrf52840 board**, which will be referred to as the seeed board from now on. We chose this LIPO since it has a small enough footprint to fit within our keyswitch housing as has a enough power to last an entire standard wireless keyboard about 36 hours under typical use, allowing our keyboard to meet our battery life requirements.

We will be using **inductive charging** which is a type of wireless charging to deliver rechargable power to the keys wirelessly to reduce hassle for the user.

As seen in 2, there will be a flat and round inductive coil along with a thin ferrite sheet above it to minimize power loss though magnetic leakage. This coil with sheet attatchment will be at the very bottom of the keyswitch housing as depicted in 2, and will be connected to a rectifier board. These 3 things: *coil, sheet, and rectifier board* make up the **inductive power receiver**. This power reciever will be connect to the seeed board via the exposed 5V and GND pins, which feeds into the seeed board's built in power management system and charges the 100mAh LIPO battery supplying power to the seeed board.

For our **inductive power transmitter**, we will have a power station where the user would put their keys in the station and leave them there to charge overnight. While ideally, the user would not have to take off the keys at all, for this first prototype, we are using the more well researched and consistent inductive charging, which requires close coupling and has faster power delivery, and since the keyboard will last though 2+ days of continuous use during waking hours (12 hours a day), taking off the keys before bed and charging them overnight should not be too much of an inconvenience. The power station will be made of a transmitter board and transmitter coils with a ferrite sheet underneath the coils. We will be using the **XKT-510** wireless transmission chip on our transmitter board and the **T3168** on our reciever board, which are simple and cheap Qi wireless charging compliant wireless charging chips. Ideally, in a finished charging station, there will be one transmitter for each key, making a 1:1 correlation between transmitters and recievers, but due to supply issues like cost and delivery time of wireless charging chips and boards, we will only be demoing at minimum 1 transmitter as a proof of concept for this prototype. More specifically, the chips we listed would have to ship from China and either be extremely expensive to quickly ship or arrive very late, so for our protype, we buy a set of receiver and transmitter boards off amazon that use these chips and desolder capacitors and resistors and replace them with our tuned values.

6.2 KeySwitch Communication

To communicate our keyswitch states (pressed, unpressed) to our central micro-controller without using wires since wires would overly complicate the process of moving switches around, we are using the Bluetooth Low Energy protocol or BLE, which is often used in smart home environments where many nonconnected sensors need to communicate wirelessly. BLE is both low power, meaning it reduces power consumption, extending our battery life, and low latency. It's latency is even lower than traditional bluetooth 5.0 due to short data send preparation times, but it has lower throughput than bluetooth 5.0, which fits our application since sending 1 binary state (on or off) per keywitch does not require high throughput. We are using the **Seeed Studio XIAO nrf52840 board** for it's built in BLE capabilities and programming the seeed board using the Arduino and the **Arduino BLE library** since that is what the board already supports without re-flashing the boot-loader. Our main micro controller, which is temporarily another seeed board, will simply scan the state of the keys using BLE and communicate that information to a computer via a **USB-C** connection just like a normal keyboard [1]. Other options for the main microcontroller are being researched, but we will stick to using another seeed board unless we find a better option.

6.3 Keyboard Housing

The housing to hold both the components of each keyswitch together and the charging station will be created using CAD and 3D-printed based on the dimensions of the components listed in 2. The baseplate that provides the freely adjustable keyswitch area will be made of a **thin acrylic plate** with a **Velcro pad layer** on top for securing each keyswitch whereever the user wishes to put it. We chose acrylic for the support plate since it is both light and strong, making it resistant to long term use and still portable. To prevent the sliding of the baseplate while typing due to it's light weight, we will be adding *rubber feet* to the bottom of the baseplate.

6.4 Keyboard Configuration

To make the keybinds - the letters and shortcuts associated with each key - customizable, we will be creating a desktop application that allows the user to set different layers of keymaps, maps of letters, symbols, and shortcuts to each key to allow for standard custom keyboard customizability. We are currently planning on using python UI libraries for the front end UI, and the backend processing of the configuration software will generate arduino code associated with the keybinds and flash them onto the main microcontroller using a terminal command.

Note that it is standard for current customizable keyboard software to generate binary/hex files and flash it either using a UI or requiring the user to execute commands on the terminal themselves. For our application, the user will not have to flash it themselves, and instead a click of a button will execute the flash commands for them.

7 TEST & VALIDATION

Our testing methods will evaluate how our final project will hold against the set design requirements that will seek to fulfill our use case requirements as well. In order to have a high-quality keyboard with novelty and practicality, we have noted previously that we need a high-latency keyboard that has a prolonged battery life with charging ease and it is cost-efficient. The keyboard should be stable to use, has easy portability, and fulfills its primary goal of being an infinite layout freedom keyboard.

7.1 Tests for Latency

Latency is the speed test of the keyboard where we observe how fast the response from the press of the wireless keyboard is registered on the computer. A fast response is necessary for applications like gaming and ensuring that the user will not see a significant delay in the usage of the keyboard. For this, we shall use the slow-motion video camera on the iPhone and count how many frames there are for the keypress to be registered. The FPS on the slow-mo camera on an iPhone 12 is 240FPS, which is 4.2 ms per frame. So in order to reach our goal of 50ms latency, there must be 12 frames or less in the recorded video test.

7.2 Tests for Battery Life and Charging

Battery life is crucial as it determines how long without interruptions can the user use the keyboard for a prolonged period of time and how convenient is the charging mechanism so the user would not be burdened with waiting a long time for their keyboard to charge. For this test, we use time as our metric to fulfill where the charging time for all of the keys must go from 0%to 100% in 8 hours or less, which will be tested by observing the battery progress on the software we will create for the keyboard configuration which will gain the data from the BMS on the SEEED Xiao boards. If the charging current is too low and thus not charging fast enough, we will use contact pads at the bottom of the switch to directly charge the keys instead of inductive charging.
For how long our keyboard lasts, we'll record how much battery life has drained in one hour of continued usage and the metric goal is for it to drop less than or equal to 5% in order for the estimated battery life to be around 20 hours, which is around a full day worth of continued use. If the battery life fails, we will increase the size of the LiPo to fit our design requirements.

7.3 Tests for Cost

The average cost of a quality keyboard is about $200. This keyboard will be aimed to be less than $300 in order to be worth less than buying 2 new keyboards to test out a different layout. The cost calculation will include the total cost of the bill of materials used in the product plus an assembly fee of $15 an hour to assemble the parts together. This assembly fee only includes the time of active assembly of each key into the 3D housing and %15 per hour is slightly higher than the Pennsylvania average pay of a factory worker which is approximate $12.90 an hour. If the cost of the board is still a bit high we will value our labor less. The absolute lowest we should go is around $5 as that is still higher than the $3.62 USD that Chinese factory workers are paid.

7.4 Tests for Stability

It's important that the keyboard feel steady when typing, especially at fast speeds, and for many different people to be able to type with ease on the keys. In order to account for this variety of potential users, we will have a survey of many users typing on the keyboard for around 30 minutes in order to get used to the keyboard and be able to use it in a setting they would like it for, be it for keyboard short cuts, an online game, or ergonomic use. The survey will take note of whether the keyboard feels comfortable and stable despite the velcro board or tall keys. If there wobble in key usage is more noticeable, then we will search for stronger alternatives in the keyboard mat material such as the 3M Dual-Lock.

7.5 Tests for Portability

Portability is important and the best thing about a wireless keyboard is that it could be used anywhere and is convenient to move around. The metric for portability will be that the weight of the total keyboard for the keys and board should be less than a kilogram total, which will be measured on a scale. The risk here is that if the modules are too heavy we can lighten up the 3D housing to have fewer infills or change to use a different key switch that is lighter or use lighter keycaps.

7.6 Tests for Wireless Typing

The wireless typing test is necessary to ensure the keyboard can work wireless and without issue connecting to multiple BLE receivers. We must ensure the whole system can support 16 keys to reach our goal of a successful macro pad with 16 usable keys.

7.7 Tests for Layout Freedom

This is crucial to the use case of having a keyboard to be used in many different configurations are a variety of locations and distances apart. Each key can easily move in the x and y directions and the keys will continue to function properly no matter the configuration being used. We shall test this by putting the keyboard in a variety of common configurations of keyboards: split keyboard, circle cluster, arrow keys, and more configurations at a variety of distances as well to test the viability of how flexible the layout can be. The keyboard must be able to function in the most common configurations of keyboard layout at the regular distance of key spacing which is about 2mm to 4mm apart.

8 PROJECT MANAGEMENT

8.1 Schedule

The schedule Gantt chart is shown in Fig. 4. The tasks are broken down by week starting from week 2 (Feb 06), and finish by week 13 (the week before finals week).

8.2 Team Member Responsibilities

1. Ben - Programming each of the key modules and getting them to interface with a central controller via BLE. Helping design and test the 3D printed keyswitch housings.

2. Korene - Wireless charging controller research and design. Helping design and test 3D printed keyswitch housings

3. Zhejia - Wireless charging controller research and prototyping. Frontend app development of configurator.

8.3 Bill of Materials and Budget

Currently this is a bit of an over estimation as we are still looking at multiple other options to be purchased.

8.4 Risk Mitigation Plans

1. Module Weight Issues: If the modules end up being too heavy, we plan to modify the 3D printed housings to include less infill, or use low profile switches and keycaps.

2. Battery life insufficient: If the battery life ends up being insuffecient, we plan to buy larger LIPOs and redesign the 3D printed housing to accomodate them.

3. Stability: If the keyswitch modules end up being too unstable or wobbly, we plan to use a stronger fastener such as 3M Dual-Lock to hold the switches in place.

4. BLE connectivity issues: If a single BLE receiver is unable to handle the simultaneous connection of 16 different keys, we may opt to use multiple different BLE receivers to connect all of the keys to the computer.

9 RELATED WORK

Regular keybinding customizable keyboards with various fixed layouts have been on the market for a while and originated as an offshoot of the mechanical keyboard market, which was no longer being popularly manufactured with the new devices. More "ergonomic" layouts, however, did not really gain any traction until the introduction of the Planck keyboard around 2019. [3].
Bluetooth is a relatively new development in the keyboard space, with custom bluetooth keyboard not really existing until the introduction of bluetooth enabled Arduino pro micro devices such as the BlueMicro or nice!nano around 2020. Still today, however, the adoption and software support of custom Bluetooth keyboards is limited.

10 SUMMARY

Overall, our design uses seeed XIAO boards with BLE capabilities and wireless individual power supply to each key to create a low latency, long battery life, freely layout customizable keyboard, where the keys can be placed in whatever layout and configuration and still deliver speedy keyboard responses to user devices. Custom keyboard users no longer need multiple different keyboards for their different setups and use cases. Some challenges include optimizing for the hopping connection latency of BLE to preserve low latency, keeping the size of our wireless keyswitches with the house within normal keycap sizes, and integrating everything together without sacrificing too much convenience.

Glossary of Acronyms

- BLE - Bluetooth© Low Energy
- BMS - Battery Management System
- LIPO - Lithium Polymer Battery
- mAh - Milliamp-hour
- UUID - Universally Unique Identifier

References

[1] *Getting started with Seeed studio Xiao NRF52840 (sense): Seeed Studio Wiki*. URL: `https://wiki.seeedstudio.com/XIAO_BLE/`.

[2] Shubham Munde. "Mechanical Keyboard Market Report". In: *Market Research Future* (2020).

[3] *The early years of custom keyboards*. 2020. URL: `https://kbd.news/The-early-years-of-custom-keyboards-1440.html`.

[4] *ZMK power Profiler*. URL: `https://zmk.dev/power-profiler`.

Table 1: Bill of materials

Description #	Manufacturer	Quantity	Cost @	Total
Seeed Studio XIAO nRF52840	Digikey	20	$9.59	$153.44
Flat Coil Inductors	Digikey	40	$2	$80
LiPo Batteries	Amazon	5	$18.99	$75.96
Wireless Charging Kit	Adafruit	16	$8.96	$143.36
Microcontroller PCB fab	JCLPCB	1	$10	$10
3D Housing	RoboClub	16	$9	$0
Cherry MX Switches	Ben Sun	16	$0	$0
Velcro	Ben Sun	1 sheet	$0	$0
Circuit components	220 Lab, ECE kits, Physical Computation Lab	many	$0	$0
				$452.76

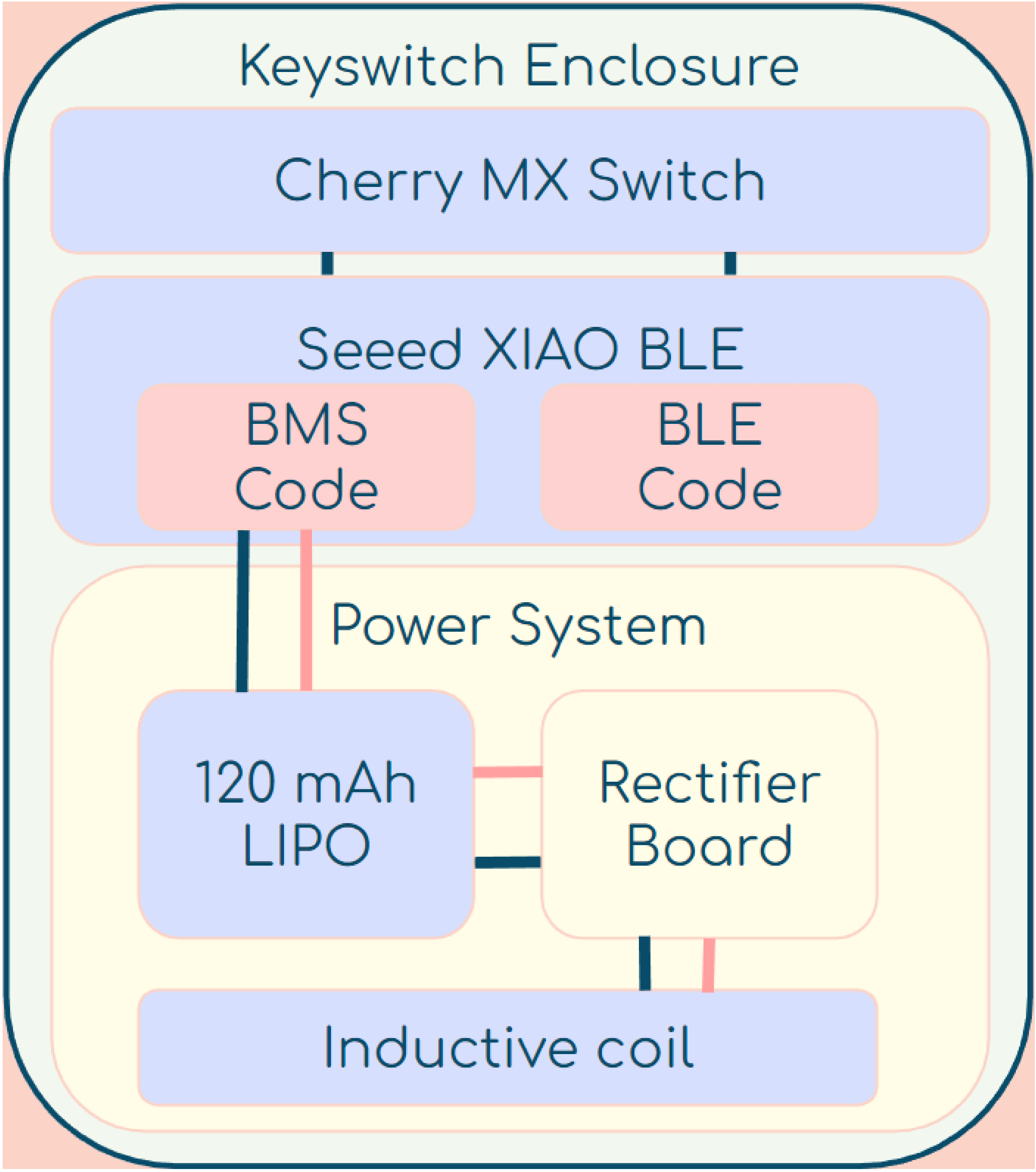

Figure 3: A full-page version of the same system block diagram as depicted earlier.

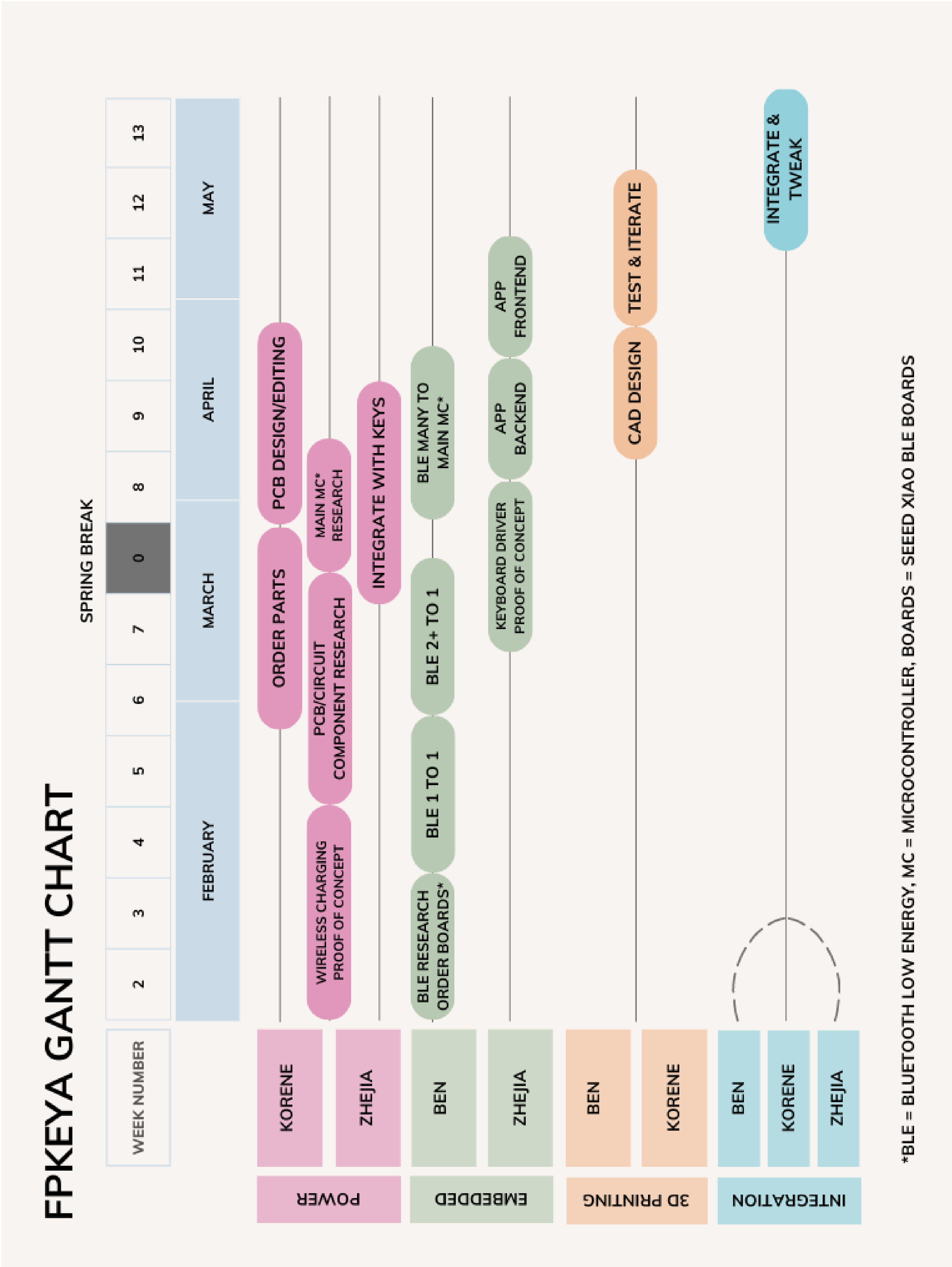

Figure 4: Gantt Chart