Blocks of Code (v3)

Reimagine the Building Blocks of Programming

Group 28 Authors:

Jordan Merkel	Camilla Torres	Annesley Kolb	Destiny Ellenwood
Computer Engineering	*Electrical Engineering*	*Electrical Engineering*	*Computer Engineering*

Reviewer Committee:

Dr. Mike Borowczak	ECE	Associate Professor
Dr. Shady Elashhab	ECE	Lecturer
Dr. Sonali Das	ECE	Lecturer
Dr. Nazanin Rahnavard	ECE	Lecturer

Advisors:

Dr. Mike Borowczak	ECE	Associate Professor
Dr. Arthur Weeks	ECE	Senior Lecturer
Dr. Sreeram Sundaresh	ECE	Lecturer

Chapter 2 – Project Description

2.1 Motivation and Background

Block-based programming has become one of the most popular and effective ways to introduce children to coding. Platforms such as Scratch and Blockly emphasize drag-and-drop blocks over typed syntax, reducing barriers to entry and allowing beginners to focus on problem solving and creativity. This approach has been widely adopted in classrooms because it transforms programming into a more visual and approachable activity. Learners can see coding as the arrangement of colorful blocks that fit together like puzzle pieces, which helps them understand sequencing, logic, and cause-and-effect relationships without being overwhelmed by technical details.

Our project builds on this same philosophy but extends it into the physical world. Rather than dragging blocks on a screen, children will be able to snap together tangible coding blocks to form programs. Each block represents a different programming concept such as input, output, or control flow, and the process of assembling them becomes a literal, hands-on way of coding. Research in tangible programming interfaces (TPIs) shows that physically embodied coding tools can be especially effective in classrooms, giving children a more intuitive and engaging way to learn. This tangible approach lowers barriers even further, particularly for younger children and those with disabilities who may benefit from tactile interaction over screen-based systems.

In addition to accessibility, our design emphasizes multi-sensory feedback. When a child assembles a sequence of blocks, the outputs will be seen through LED lights and visual screen displays and heard through sound and music. By combining physical coding with multisensory outputs, our system provides immediate reinforcement of programming concepts in ways that can be understood and enjoyed by children with different abilities.

Comparable products in the market, such as LEGO Education SPIKE, Cubelets, and Cubetto, demonstrate the growing demand for screen-free coding experiences. However, most of these focus on robotics or visual problem solving, with limited attention to accessibility or multisensory inclusiveness. Our project fills this gap with a disco party theme, turning coding into a playful, accessible, and inclusive experience. By merging physical block programming with lights, music, and real-time app visualization, the system offers children a unique way to explore computational thinking by literally holding their programs in their hands while experiencing the results in a fun, multi-sensory way.

2.2 Existing Products, Past Projects, and Prior Related Work

2.2.1 Prior Related Work

Blocks o' Code Version 1 (V1).

The first version of Blocks o' Code was developed by Erebus Labs in 2015. Unlike the worksheet prototype, this version used real hardware. Each block was a four-layer PCB enclosed in acrylic, containing an ATtiny461a microcontroller, indicator LEDs, and a selector wheel connected to a potentiometer. Blocks communicated with each other using SPI for local connections and a TWI bus for global communication, with a BeagleBone Black acting as the main processor. The processor board ran a lexer, parser, and interpreter to process the block arrangements, and could drive different outputs like an LCD screen, RGB LEDs, a piezo buzzer, or a relay. This made V1 a significant step toward tangible programming because it was interactive, but the system was expensive, somewhat fragile, and limited in scalability.

Blocks o' Code Version 2 (V2).

The second version was created by UCF Senior Design Group 13 in Fall 2023–Spring 2024. V2 took the worksheet idea and transformed it into physical blocks with embedded electronics. Each block had its own function, and when connected, the blocks formed programs that could be executed. Another big step was the introduction of a companion app. The app allowed users to visualize the block arrangements, see how they translated into commands, and interact digitally. This combination of hardware and software was a strong milestone, but V2 still struggled with connection reliability, scalability, and an app that was basic compared to what learners are used to today.

2.2.2 Existing Commercial Products

MatataLab Musician Add-On.

MatataLab created a kit that teaches coding through music. With this add-on, students arrange blocks to compose songs, turning coding lessons into playful and creative activities. This shows the power of mixing logic with art and is especially important to our project since we are also focusing on music as a way to teach coding. The MatataLab musician kit proves that students are excited to learn when coding is combined with music, and this directly supports the direction we are taking with our project V3.

Cubroid Coding Blocks.

Cubroid is a wireless coding block system where each block has its own function, like motors, sensors, or lights. It demonstrates how coding can be hands-on and interactive in a classroom setting. Cubroid's strength is that it makes learning fun and immediate, but its reliance on proprietary modules makes it less customizable and harder to scale for more advanced engineering use.

Tynker Sound Blocks.

Tynker is a popular digital coding platform. One of its features is sound blocks, which let students add music, sound effects, or voice into their programs. Even though Tynker does not use physical blocks, it is highly relevant to our project because it shows how a companion app can keep learners engaged through multimedia features. Since V2 already had a basic app, Tynker inspires us to expand ours with real-time audio feedback and stronger integration with the physical blocks.

LEGO Education SPIKE.

LEGO SPIKE combines LEGO bricks with programmable hubs, sensors, and motors, all controlled through a block-based app. It is widely used in STEM classrooms and competitions to teach problem solving and coding through play. SPIKE proves the effectiveness of combining hands-on construction with digital coding, and it serves as an important reference for how our project combines physical coding blocks with a companion app.

Cubelets.

Cubelets are small robot blocks that connect magnetically, with each cube representing a different behavior such as sensing, logic, or output. By snapping them together, students can build robots without needing wiring or screens. Cubelets highlight the appeal of simple, tactile system, though they lack deeper coding features and do not integrate music or app-based feedback like our project does.

Cubetto.

Cubetto is a wooden robot designed for early learners that is programmed using physical blocks placed on a board. It is screen-free and introduces sequencing and logic in an approachable way for young children. Cubetto shows how tangible programming can reach very young learners, but it is limited to very basic coding concepts and does not expand into multimedia experiences.

2.2.3 Existing Coding Platforms

Scratch.

Scratch is one of the most popular visual coding platforms for kids. Students drag and drop blocks to create animations, games, and even music. It emphasizes creativity and storytelling, which aligns with our vision for using music and playful interactions in our project V3.

Blocky.

Blocky is Google's visual programming library that powers many educational apps. It shows how block-based coding can be directly translated into real code, which inspires how our companion app will display the logic of physical arrangements.

2.2.4 Summary and Relevance

Looking back at earlier versions of Blocks o' Code and exploring commercial systems gave us a clear direction for Version 3. V1 and V2 established the foundation, showing that tangible blocks combined with an app can be an effective way to teach coding. MatataLab demonstrates how music can make coding playful and engaging, which matches directly with our vision of using music to teach. Cubroid shows how interactive block systems can succeed in classrooms but also reveals the limits of proprietary, closed systems. Tynker highlights the importance of companion apps with multimedia features, which motivates us to make our app more advanced and engaging.

By combining the lessons learned from V1 and V2 with inspiration from these commercial products, our Version 3 project will be more reliable, expandable, and fun. With stronger block hardware, integrated music features, and an enhanced app, we aim to create a system that not only continues the Blocks o' Code legacy but also pushes it further into a more creative and interactive direction.

Our team is developing the third version of Blocks o' Code. We are improving hardware reliability, making the system more expandable, and redesigning the companion app to be more powerful and fun. One of our biggest focuses is bringing music into action. While V2 has a basic app, our V3 app will provide real-time audio and visual feedback, inspired by how kids already enjoy music-based coding. By combining stronger block hardware with an app that integrates multimedia like sound and music, V3 will create a richer and more engaging experience than past iterations.

2.3 Goals

2.3.1 Basic Goals

- Design and implement at least 25 fully functional blocks, each containing a low-cost MCU (ESP32-C3) with sufficient GPIO pins for peripherals and I^2C communication capability.
- Support up to 25 blocks per configuration connected via magnetic connectors carrying power (3.3-5V) and I^2C data lines.
- Develop magnetic snap-together physical coding blocks that provide power and half-duplex data transfer to a master "Brain" block MCU.
- Support core control-flow logic at the firmware level: Start/Stop execution flag, If/Then Boolean branching, Loop counters, and Delay timers.
- Provide visual indication of block functionality on each block.
- Include at least two output modalities:
  - LED driver block with PWM control (1-10 kHz)
  - Piezo speaker block with tone generation (100 Hz-2kHz)
- 3D-printable enclosures with embedded magnet alignment and PCB mounting.

2.3.2 Advanced Goals

- Add more input options:
  - Microphone sensor block with ADC sampling (~8kHz)
  - Voice input keyword detection using an ML model
- Introduce data blocks: Boolean (yes/no), Integer (counters, values).
- Implement synchronized music + disco effects across blocks using I^2C broadcast packets from Brain MCU.
- Average cost of <$5 per block by using low-cost MCUs, minimal passive components, and standard PCB fabrication

- Ensure low-power operation with per-block consumption of <100mA during active mode

2.3.3 Stretch Goals

- Orientation-free block connectors with rotational symmetry to allow blocks to connect from any direction.
- Implement wireless charging via magnetic coupling for each block when stored in a charging drawer.
- Enable a companion app leveraging the ESP32 Wi-Fi module that shows a real-time visualization of block configuration and generates a live disco visualization.
- Add OLED displays to blocks on selected blocks for a richer output.
- Provide randomized sequence generation for music and LED patterns.

2.4 Objectives

2.4.1 Basic Objectives

- Select and program a low-cost MCU per block (Arduino R4 Minima for simplicity, or ESP32-C3 for wireless capability).
- Establish a reliable data-sharing protocol over I²C between blocks with the Brain MCU serving as master.
- Design and 3D-print magnetically connecting enclosures with pogo-pin data/power contacts.
- Implement firmware for core control blocks (Start, If/Then, Loop, Delay) and ensure correct sequencing across configurations.

2.4.2 Advanced Objectives

- Expand the block set to include audio input (microphone/voice), motion sensors, and data-handling blocks.
- Develop synchronized audio-light effects across multiple output blocks with millisecond-level timing accuracy.
- Optimize hardware BOM and design to achieve a unit cost <$5 per block.
- Implement power management strategies (sleep modes, duty cycling) to extend runtime.

2.4.3 Stretch Objectives

- Design orientation-independent magnetic connectors for maximum user flexibility.
- Integrate wireless charging coils + drawer station to simplify recharging.
- Extend firmware and app integration to a real-time visualization platform (via ESP32 Wi-Fi link).
- Add high-resolution OLED displays and optional randomized music/LED effect generation for extended engagement.

2.5 Description of Features and Functionalities

Blocks of Code will contain 25 or more 3D-printed physical programming blocks that serve different functionalities. These blocks will be contained in a drawer that will be able to wirelessly charge each block individually, as each block will have its own source of power. When assembled into a configuration, the blocks will work together to create an interactive “disco-party” experience. This combination will also introduce children to programming concepts such as sequencing, logic, and iteration, while the music-and-lights theme keeps the experience engaging and fun.

2.5.1 Physical Block Functionalities

Input Blocks

- Start “Brain” Block
    - This block is required for any block configuration. It indicates the start of the program and takes all the data from the other blocks via I2C communication.
- Note Block
    - This block takes an input of a button press (from a choice of 7) and will add a note into the disco sequence. These buttons correspond to music notes A-F.
- Microphone Block
    - This block takes an input of a voice or clap to be used with control flow blocks. This block will do different actions based on the user input. For example, if the user says “Go” or “Start” the music sequence will begin, or if the user claps lights will flash.

Control Flow Blocks

- If/Then
    - This block is used to introduce conditional logic into the sequence. It takes an input block and an output block and will perform the action of the output block if the input action happens.
- Loop
    - This block introduces repetition of actions within the music sequence. The user will specify a number of times for the action to repeat and choose what action will be repeated. This can be used to play a note repeatedly or flash LEDs multiple times.
- Delay
    - This block adds timing control to the sequence by pausing between actions. The user will specify a number that represents the number of seconds of delay.

Output Blocks

- LED Block
    - This block adds colorful visual feedback. It may include a single RGB LED or an array of LEDs capable of flashing, fading, or pulsing in sync with the sequence. This block helps create the disco party effect by lighting up in time with music or responding to user inputs.
- Speaker Block
    - This block produces audio output such as tones, melodies, or rhythmic beats. It works in combination with Note Blocks to play musical sequences and with control-flow blocks to respond to inputs.

Along with the physical blocks, Blocks of Code also includes a real-time companion app that mirrors the sequences created by the user. As blocks are added, removed, or rearranged, the app updates instantly to display the current configuration and visually represent the outputs. The companion app will not only display the real-time block configuration but will also generate a visual representation of the disco experience once the Start Block is activated. This means that while the physical blocks control lights, sounds, and vibrations in the real world, the app will simultaneously create a virtual party with animated lights, music cues, and effects that mirror the outputs.

2.6 Prototype Illustration

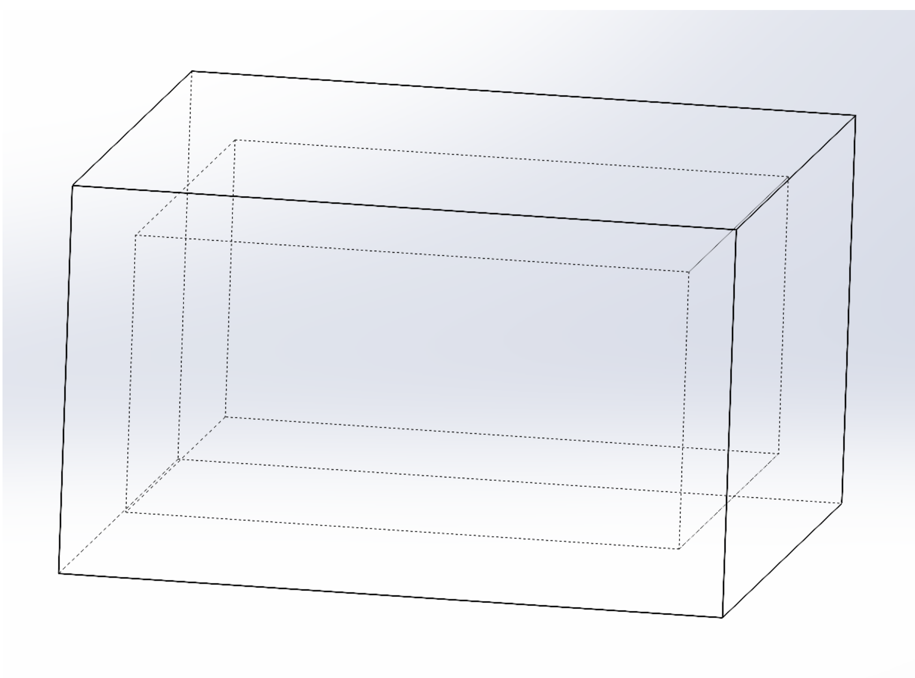

Figure 1: SolidWorks rendering of the cube prototype

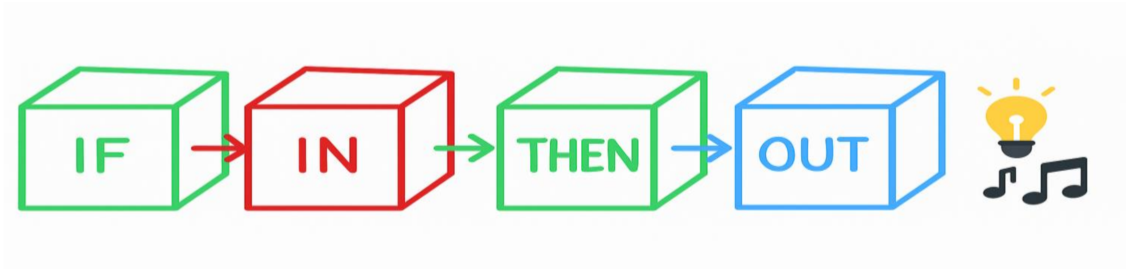

Figure 2: Conditional logic diagram

This rough draft of the prototype that we are developing for the Blocks of code (v3) system focuses on the design of a standardized cube enclosure. This will serve as the physical foundation for all block types. This cube was modeled in SolidWorks as a 50 × 50 × 50 mm hollow cube with 2mm wall thickness. This provides the appropriate balance between compactness while having enough room to house all the required electronics. While designing, the size was selected to ensure that the block provides internal volume for a printed circuit board (PCB), coin-cell battery, and other peripheral components, while staying compact and durable for repeated use.

This cube enclosure will be 3D-printed in a transparent PETG Filament. Transparent filament was chosen to allow users to visually connect physical electronics to the abstract programming functions. This also allows the visibility of the internal indicators such as LEDs or small displays to be visual. The goal is to get more kids interested in STEM due to visualization. To ensure that the assembly and durability of these cubes last, it is designed as a two-piece enclosure secured with screws threaded into reinforced bosses. This ensures that the blocks can be opened for maintenance while remaining mechanically safe for student handling.

The power for each block is supplied by an internal rechargeable battery, similar to mag-safe charging. The enclosures will be designed with consideration for this compatibility. In the final design, blocks will be recharged when placed into a dedicated storage and charging station. This station will contain drawer trays with embedded inductive charging coils allowing for the block to be recharged while they are stored away.

2.7 House of Quality

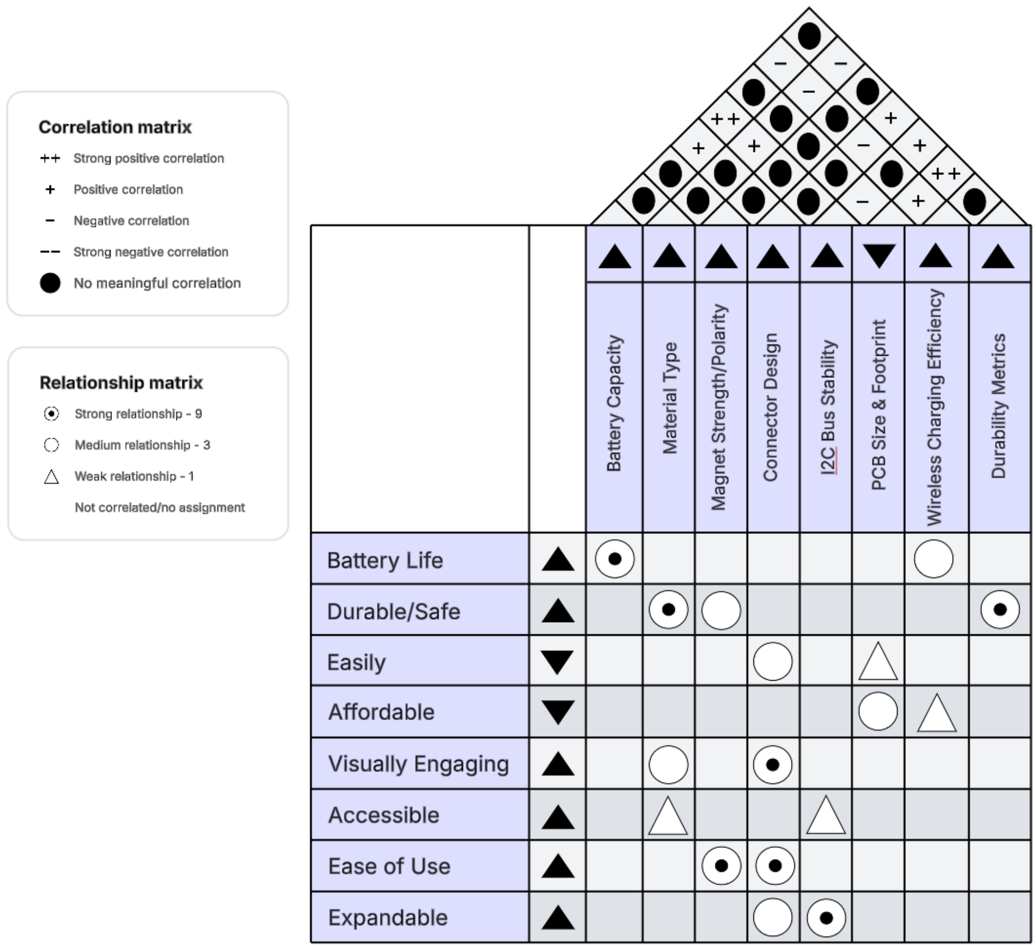

Figure 3: House of Quality by Annesley Kolb

2.8 Engineering Specifications

Component/ Block Level Specifications

Component/ Block	Specification	Quantitative Measure	Verification Method
Gateway Block	MCU	ESP32-C3, 160 MHz, Wi-Fi + BLE	Functionality test
Function Block (OLED +Mic)	Display size	0.96” I^{2}C OLED, 128x64	Power-on + I^{2}C scan

Function Block (Button)	Switch reliability	≥ 50,000 cycles	Press-count testing
Function Block (Speaker)	Audio output	0.3 W into 8Ω @3.3V	Oscilliscope + SPL meter
Power Module	Battery	200 mAh Li-ion, 3.3V	Charge/discharge test
Magnetic Connectors	Retention force	≥ 1.2 N per face	Pull test with scale
Contact Resistance	Pad-to-pad	≤ 150 mΩ per interface	4-wire ohmmeter

Overall System Specifications

Specification	Quantitative Measure	Verification Method
Average cost per block	≤ $5.00 (averaged across full set)	Bill of Materials analysis
Maximum chain length	≥ 25 blocks (I2C @ 100kHz)	Repetition test with gateway block
End-to-end latency (button to LED response)	≤ 50 ms	Logic analyzer on INT and LED lines
Idle lifetime	≥ 7 days (no user intereaction)	Quiescent current measurement
Active runtime	≥ 60 minutes typical use (music + LEDs)	Timer to battery cutoff
Drop survivability	Survive 1.0m drop onto plywood, 3 times	Standard drop test
Operating tempature	0-40 °C indoor	Chamber or monitored

*Demonstrable specs: chain length, latency, and drop survivability.

2.9 Budget and Financing

Item	Price (USD)	Qty	Total Cost	Details
ESP32 MCU	$5.00	2	$10.00	Core MCU for Gateway block, Wi-Fi + I²C
RGB LED Matrix (4x4)	$4.00	2	$8.00	Visual output on Gateway
OLED Display	$3.00	2	$6.00	Display-based function blocks

Electret Microphone + Op-Amp	$1.50	3	$4.50	Mic-based function block for beat detection
Button Switches (6x6 mm)	$0.05	20	$1.00	Used in ladder/button function blocks
Class-D Audio Amp (PAM8302)	$1.00	3	$3.00	Speaker blocks
Mini 8 I2C Speaker	$1.50	3	$4.50	Speaker output for audio blocks
Vibration Motor (Coin Type)	$2.00	2	$4.00	Haptic feedback blocks
Rechargeable Li-ion Battery (200 mAh)	$3.00	3	$9.00	Power modules
Buck/Boost Regulator (MCP1640)	$1.20	3	$3.60	Battery regulation for Power Puck
PCB Fabrication & Assembly	$25.00	4	$100.00	JLCPCB prototypes, assembly for Gateway and Function PCBs
Magnets (6 mm x 2 mm)	$0.10	50	$5.00	Magnetic block alignment + contacts
Gold-Plated Contact Pads	$0.05	50	$2.50	ENIG finish for bus pads
PETG Transparent Filament (1 kg spool)	$20.00	1	$20.00	3D printing for block enclosures
Miscellaneous (adhesives, wiring, shipping)	$15.00	1	$15.00	General project consumables
TOTAL			$196.10	

2.10 Milestones

Project Milestones (Senior Design 1)				
Milestone number	**Milestone Description**	**Start Date**	**End Date (estimated)**	**Advisors and requirements**
1	Brainstorming	08/27/2025	09/04/2025	Meeting virtually to pitch ideas
2	Sponsor Meeting	08/27/2025	08/27/2025	Discuss project requirements with Dr. Mike and various ideas
3	Initial Document	08/26/2025	09/05/2025	First 10 pages of Divide and Conquer
4	Advisor Meeting	09/08/2025	09/08/2025	First meeting with Dr. Sundaresh reviewing Divide & Conquer paper and feedback
5	Revised Document	09/08/2025	09/12/2025	Final revision of Divide and Conquer paper
6	Sponsor Meeting	09/10/2025	09/10/2025	Reviewing project progress with Dr. Mike and feedback
7	Advisor Meeting			Second meeting with Dr. Sundaresh
8	Reviewer Meeting	09/24/2025	09/24/2025	Meet with reviewers for feedback
9	60-Page Milestone	10/28/2025	11/03/2025	First half of final document finished
10	Final Document	08/26/2025	11/25/2025	Final document of SD1
11	Mini Demo Video	TBD	11/25/2025	Demo of prototype

2.11 Software Flowchart

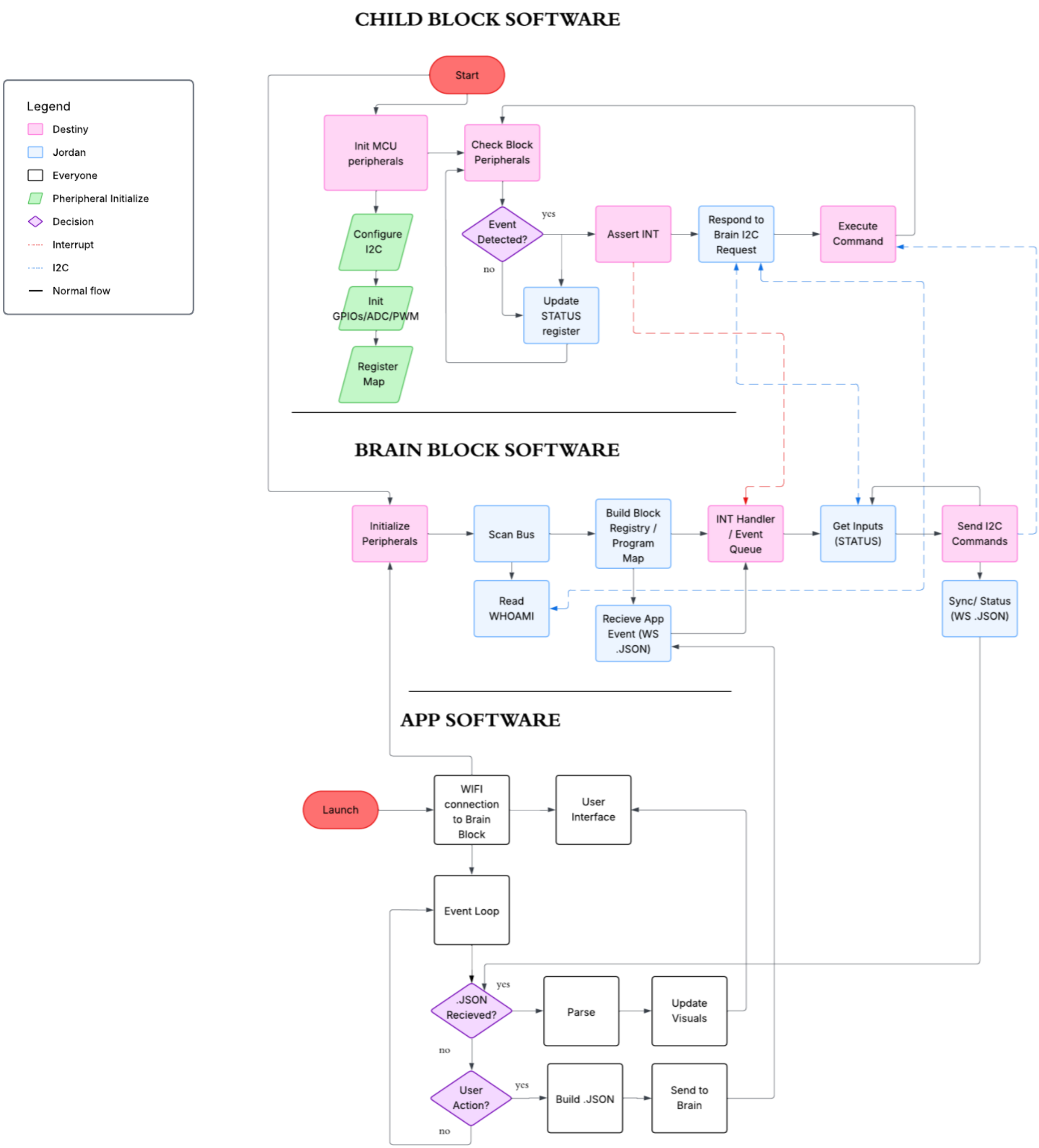

Figure 4: Software Flowchart by Destiny Ellenwood

2.12 Hardware Flowchart

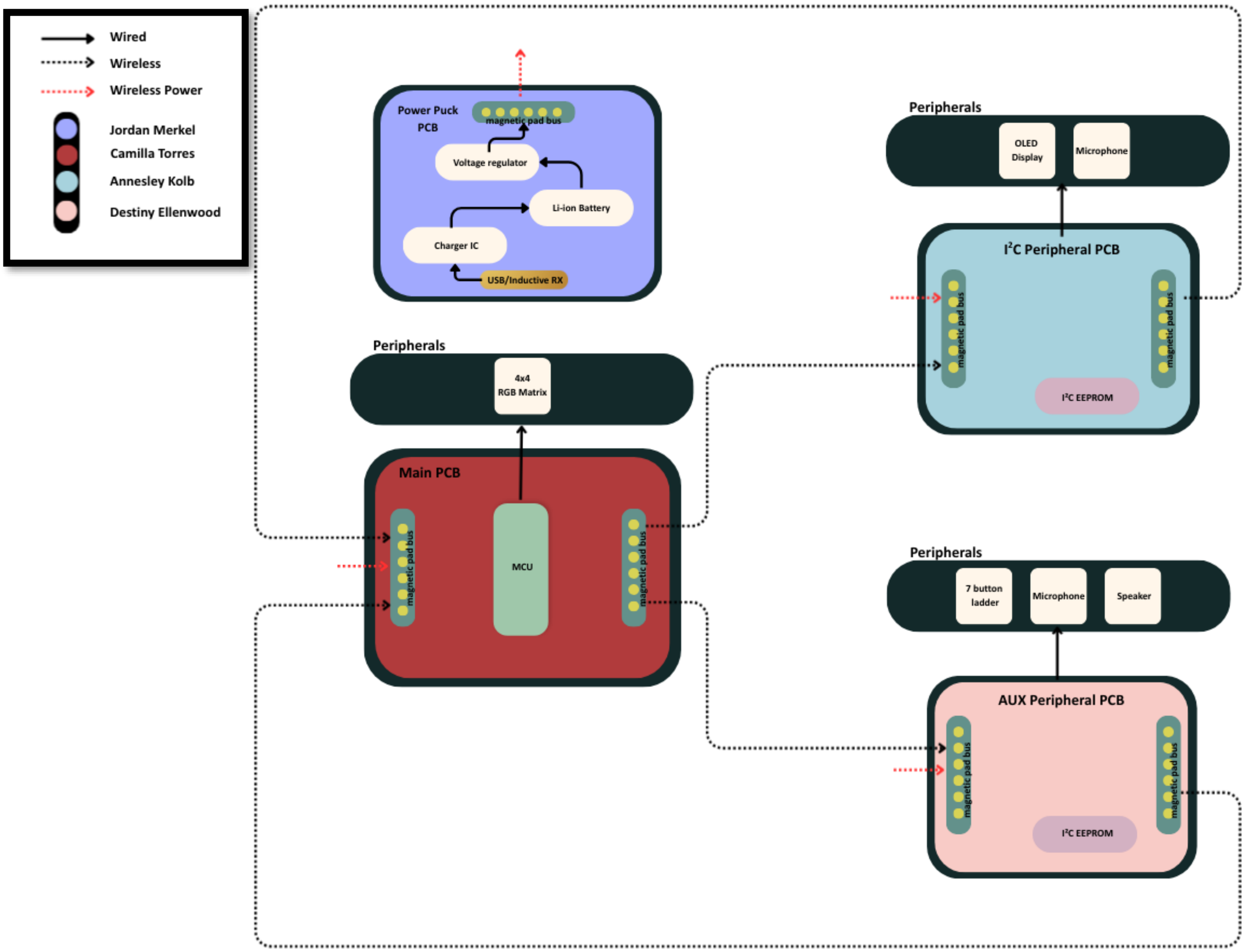

Figure 5: Hardware Flowchart by Camilla Torres

Appendices

Appendix A - References

- [1] Erebus Labs, “Blocks o’ Code V1 Worksheet.” [Online]. Available: https://github.com/erebus-labs/blocks-o-code/blob/master/Learning_Materials/Generic/HOWTO_WORKSHEET.pdf
- [2] University of Central Florida, “Senior Design Group 13: Blocks o’ Code V2.” [Online]. Available: https://www.ece.ucf.edu/seniordesign/fa2023sp2024/g13/
- [3] MatataLab, “Musician Add-On.” [Online]. Available: https://matatalab.com/en/musician-add-on
- [4] Cubroid, “Cubroid Coding Blocks.” [Online]. Available: https://www.cubroid.com/cubroid-coding-blocks
- [5] Tynker, “Tynker Toolbox: The Sound Blocks.” [Online]. Available: https://www.tynker.com/blog/tynker-toolbox-the-sound-blocks/
- [6] LEGO Education, “SPIKE Prime and SPIKE Essential.” [Online]. Available: https://education.lego.com/en-us/products/lego-education-spike-essential/
- [7] Modular Robotics, “Cubelets Robot Blocks.” [Online]. Available: https://www.modrobotics.com/cubelets/
- [8] Primo Toys, “Cubetto.” [Online]. Available: https://www.primotoys.com/cubetto/
- [9] MIT Media Lab, “Scratch.” [Online]. Available: https://scratch.mit.edu
- [10] Google Developers, “Blockly.” [Online]. Available: https://developers.google.com/blockly
- [11] J. Flynt, “Should You Print with PETG Instead of PLA?,” 3D Insider, Jul. 10, 2022. [Online]. Available: https://3dinsider.com/pla-vs-petg/
- [12] GoEngineer, “How to Make a Cube in SOLIDWORKS,” Goengineer.com, 2019. [Online]. Available: https://www.goengineer.com/blog/how-to-make-a-cube-in-solidworks
- [13] M. S. Horn and R. J. K. Jacob, “Designing Tangible Programming Languages for Classroom Use,” in Proc. Tangible and Embedded Interaction Conf. (TEI ’07), 2007, pp. 159–162.
- [14] P. Wyeth, “How young children learn to program with sensor, action, and logic blocks,” J. Learn. Sci., vol. 17, no. 4, pp. 517–550, 2008.
- [15] D. H. Rose and A. Meyer, Teaching Every Student in the Digital Age: Universal Design for Learning. Alexandria, VA: ASCD, 2002.