Pre-Alpha Build

Architectural Elements

External Interface

The primary external interface is the Host-Prototroller connection through USB (Universal Serial Bus). To convey information from the persistent state, we use serial logging through PuTTY. The serial logging is demonstrated by having two RP2040 microcontrollers (one master and one slave), communicating with each other via SPI by endlessly cycling data between each other, and outputting the buffer data over the serial connection.

Persistent State

The Prototroller is a real-time device that operates with little persistent state. As we integrate more swappable modules into the project, the host device will keep track of which modules have been inserted in an internal data store. The data store is persistent until the user presses a rescan button, which triggers a routine to check if the physical module configuration has been changed. This routine could also trigger during faults, such as an unresponsive module, to update the data store accordingly.

Internal Systems

When a module is initially connected, neither the host device nor master will be aware of the change until the rescan button is pressed. This allows a user to fully set up their controller layout before the host attempts to use it. Additionally, it eliminates the need to continually run the detection routine during normal operation. Once the master detects new modules and retrieves their associated IDs, it will store these objects in an internal data store. The master may then iterate over the connected modules and read/write data at a constant rate. The master will sequentially poll each module using a 5-to-32 decoder to pull each chip-select low, read/write serial data, and pull the chip-select high. Only one chip-select will be pulled low at a time. Regardless of whether input is registered, data will be transferred between MISO and MOSI lines between the master and slave.

Information Handling

Communication

Internally, the primary communication is done with SPI between the RP2040 master microcontroller and slave microcontrollers (onboard the modules). Connected modules will be polled sequentially by the master for new I/O states. Externally, the persistent state transmits to the host device over a wired USB connection as a Human-Interface Device (HID).

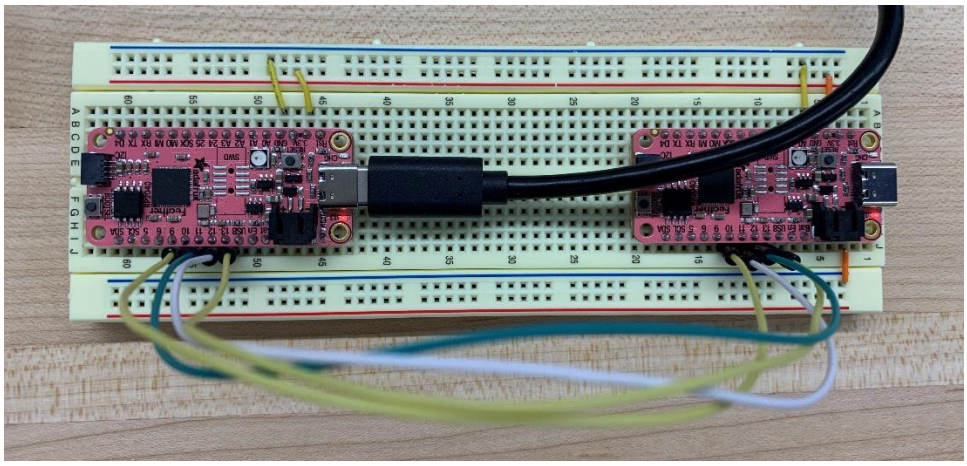

Figure 1: Physical deliverable of Pre-Alpha build demonstrating the project architecture.

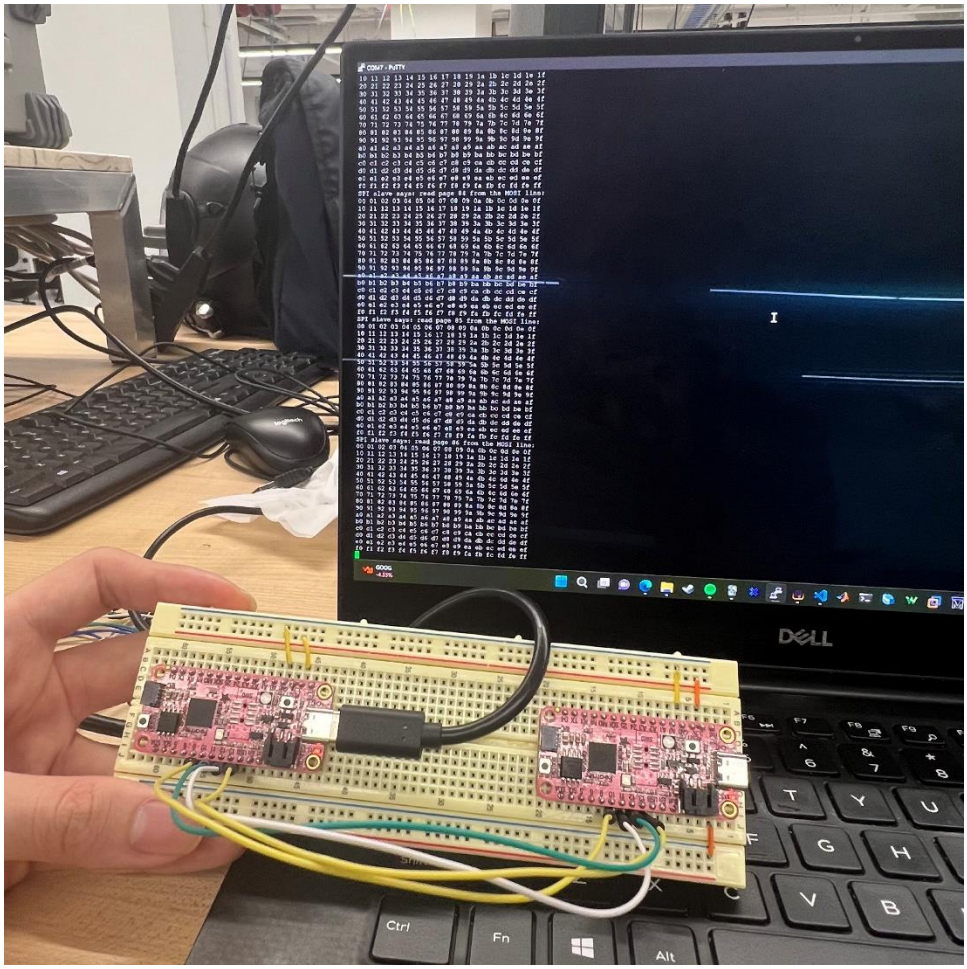

Figure 2: The external interface is shown via USB and PuTTY which gives an internal view of the entire system, demonstrating the internal data store and all major communication aspects.

Integrity & Resilience

If a module becomes unresponsive, the master will remove it from the data store of connected modules. The user will notice input is not being received, which will prompt them to check that the modules are correctly inserted and press the rescan button. Moreover, the host device can try to reset the module or send a new command before removing it from the data store.

We are not worried about malicious attacks for two primary reasons. One, sensitive data being passed through the system is not a design consideration. Two, the project is open-source and as such, our IP is freely available. For the pre-alpha build, all error logging took place through serial logging. For stretch goals in the future, we hope to provide a user interface for extra controller mapping and profile configuration (much like Dolphin emulator) and utilize the Adafruit FeatherWing OLED 128x32 (or similar) for error status messages and debugging when modules become unresponsive.