

Alpha Build – Prototroller

Britton M., Caleb O., Evan Z., Merrick R., Yu-yang H.

Usability

Interface

The primary interfaces are the modules the user may interact with, connected to the Prototroller chassis, that drives a gamepad report to the host through USB (Universal Serial Bus) connection. Additionally, serial logging is available, and can be viewed with a program like PuTTY. For our vertical slice, we have button and joystick modules. We have prepared the schematics for many more modules such as a D-pad, slider, XYAB button array, and accelerometer/gyroscope.

Navigation

The features of the controller will be discoverable through testing and manipulation of the swappable modules. Navigation with the device is intuitive to anyone who has used a handheld controller before.

Perception

An example of perception is watching the mouse move according to user inputs. Other modules will also indicate changes in application by reflecting a user input on the connected host. The state of connected modules can only be perceived by the connection of serial output. If a module is not connected correctly, or a module is inserted and no rescan occurred, the module will be in “disconnected” state (interaction proves futile). Additional modules that receive output from the controller and the connected computer will be able to provide output through a graphical interface on the Adafruit Featherwing 128x32 -- potentially displaying a mapping of what modules are currently connected to the board – or other sensory experiences such as haptic tactile feedback or audible responses from a simple buzzer device. All the outputs would be defined by the connected computer or the firmware running on the controller.

Responsiveness

The Alpha Build of the Prototroller will be running with a 12MHz crystal, which is standard for any RP2040-based hardware. This clock speed will enable sufficient controller responsiveness. However, we will continue to optimize the system for minimal latency by utilizing PLLs. Our goal is to match modern controller responsiveness of around 4ms.

Build Quality

Robustness

Modules will conjoin with the chassis using magnets to make the event of modules falling out unlikely. The casings (on both chassis and modules) will be engineered to protect against drops from heights from 5 feet and below. The quality of inputs to the master, and hence, the host, is entirely dependent on the quality of connections in the PCB and peripherals used.

In terms of programs and drivers, SPI speed should be sufficient to routinely poll all 25 modules for new data without issue. The transactions will send packets of data between the module and master and employ error-checking to determine that the data is formatted correctly.

Our schematics have ESD protection circuitry on power rails, as well as using decoupling capacitors and TVS diodes filter out voltage spikes, as well as generic RLC filters to create a barrier against undesirable frequencies in signals.

Consistency

The controller is a real-time predictable system that should not have any issues with inconsistent inputs. Our SPI data transmissions and HID reports will contain predictable information with a set buffer size.

The system will be connected through USB and shall not draw more than 500mA through the port.

A rescan button will be visible and pushing it will trigger a module rescan on all slots. Doing so will have all modules synced up with the master board. A reset button will restart the currently flashed program for the respective module. All modules will have a respective output in respect to their input due to their simple functionality and should consistently yield the same results — barring hardware failures that cannot be controlled.

Aesthetic Rigor

The Prototroller Alpha Build allows for inputs to be handled simultaneously (i.e., with both hands) to give the same feeling of using a controller. For future builds, these modules will be assembled in a more ergonomic manner to feel more akin to using modern controllers and mounted more securely to the 3D printed chassis. Additionally, grips will be included to complete the gamepad aesthetic.

Vertical Features

External Interface

A major use case is the reconfigurability of modules on the Prototroller chassis. To disconnect, the user may grab a module and pull it away (severing the physical and electrical connections). To connect, the user may insert a module into a slot and press the “rescan” button to update the persistent state. The other major use case is the I/O interaction, from which the user interacts with the mounted component while the module is connected to the chassis.

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. The data store of connected modules is routinely iterated to transmit and receive fresh data to/from the modules over SPI.

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 two 4-16 decoders 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.