Nerd Box

Group 11

Charles Nasser, Cesar Ordaz-Coronel, Anthony Bruno

Background

The issue addressed:

- Portable Gaming + Open Source Hardware + Game Creation = One Solution
- Values of a Solution: Educational, Creativity, Enjoyment

Today’s Solutions

- High-end solutions = $$$ for individual, Discontinued, moves quickly
- Low-end solutions = Discontinued, sold out, Outdated, other

Steam Deck Retail: $400 starting

Meggy Jr RGB - Outdated HW

ArduBoy - Too small form factor

Our Approach

Concept of Operations:

- User Input: device buttons, QWIIC (I2C) sensors
- Microcontroller: translates inputs to programmed outputs
- MVP Device Output: Display game animations/graphics
- Device programmed using Arduino IDE or Scratch programming
- USB for charging the battery

Objective: Deliver an open source hardware handheld, programmable gaming console to the video game console market that enables game development/creation and game hardware modification.

Requirements (Approach continued….)

MUSTS:

- Be Handheld, Portable
- Battery Powered
- User Interface (UI)
- Display Response to inputs
- Menu/Navigation
- Programmable for games

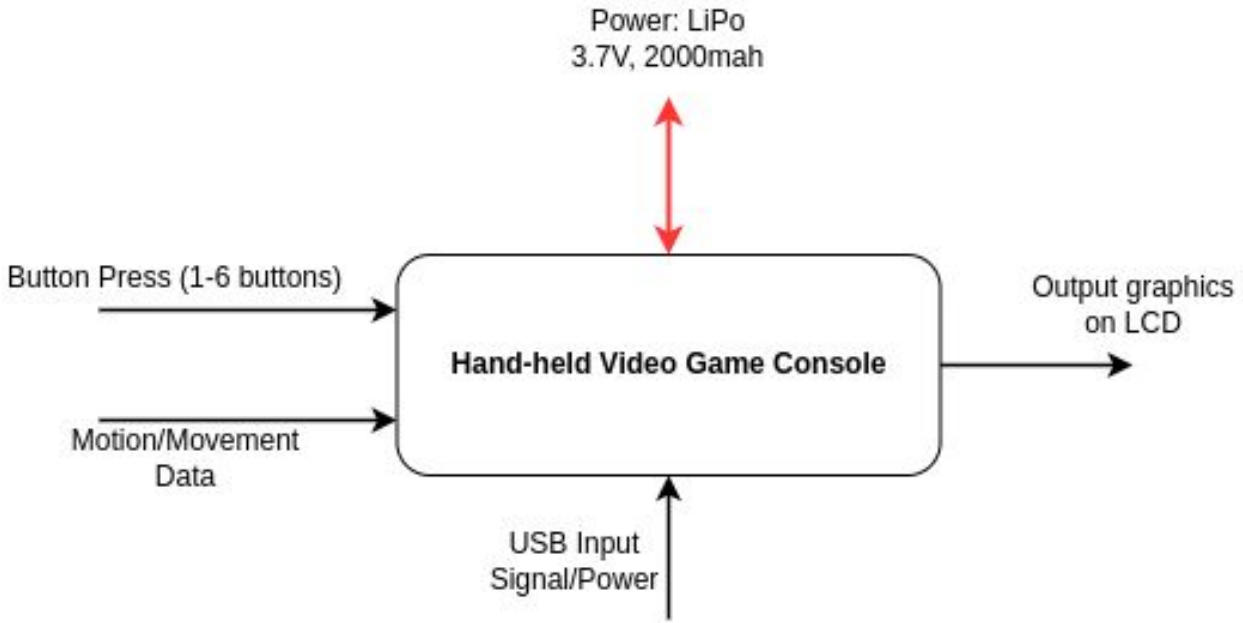
SHOULDS:

- Non-button/switch input (motion, CV, etc.)
- 3D Printed Enclosure
- Other non-display outputs (sound, rumble)
- Battery charging

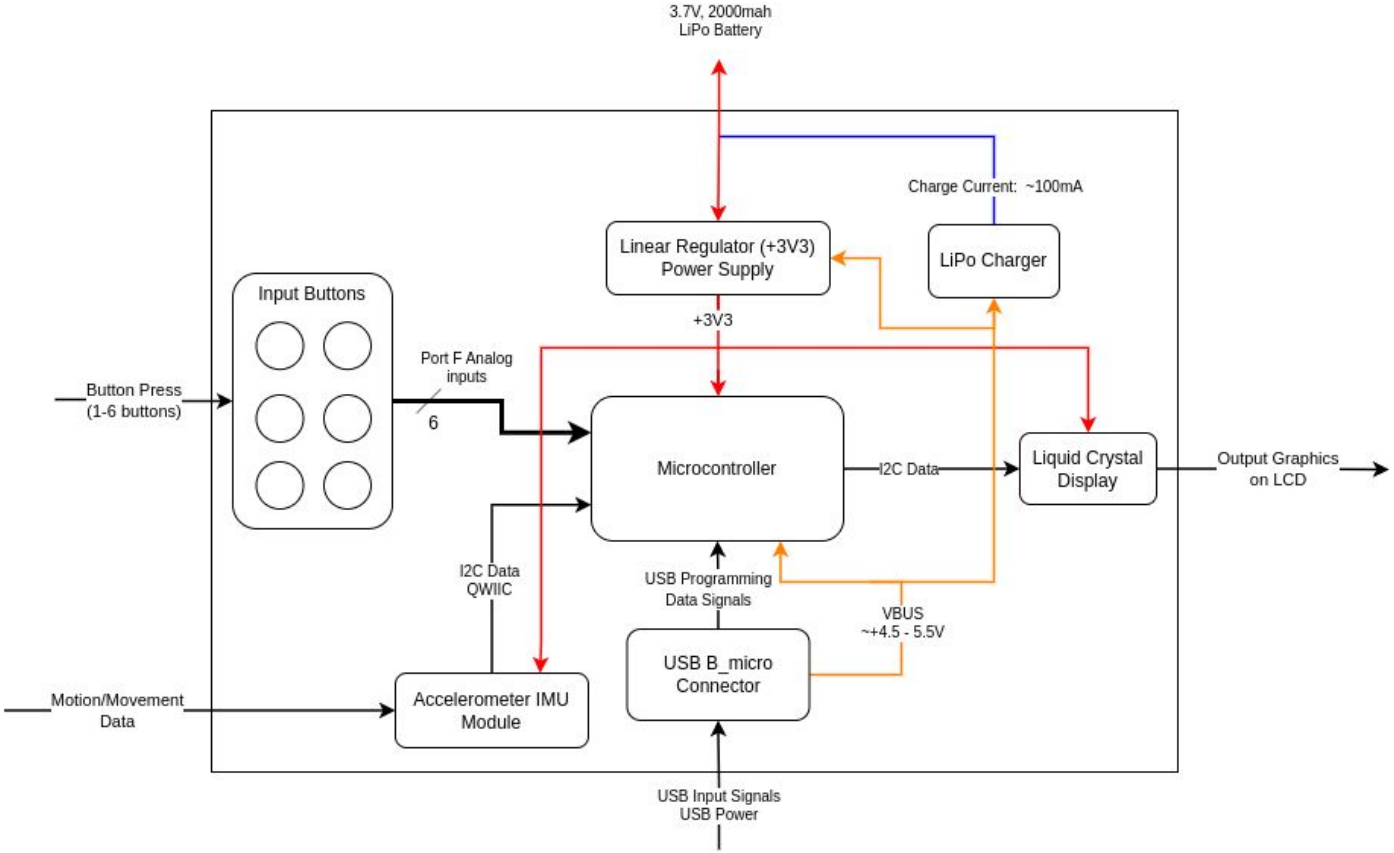
MAYS:

- Memory device for storage of user settings, more memory-heavy games
- Battery Level LED indicators
- Use another monitor to display games on larger displays

Design Overview - L0

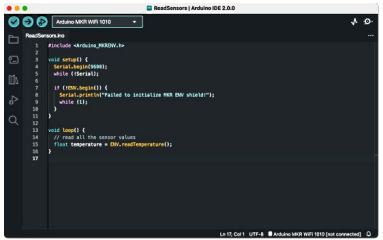


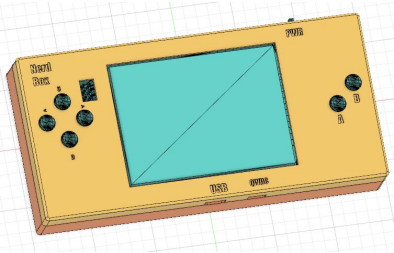
Design Overview Continued…. L1

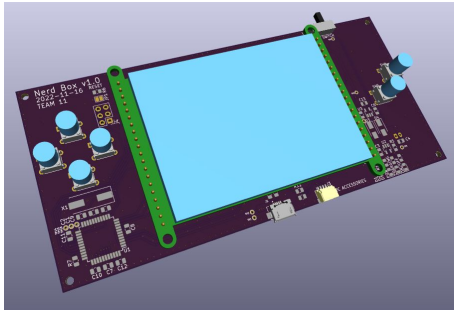


Implementation

- Look at Design Schematics
- Look at Design Board
- Look at Code
- Tools: Arduino IDE, Fusion 360, kiCAD







- Look at BOM - Bill of Materials

IP License & Existing Work

License Chosen: Creative Commons Zero V1.0 Universal

Other IP used:

- Based on Arduino Leonardo, Lilypad, Andrew Greenberg's Example schematic (ATmega32U4 µC interface) - For Hardware
- Adafruit library documentation - For code
- Based on example game coded in processing - Example code reference

Testing

- In Design/Pre-test setup
  - Placement of Test Points (TP), ~ 16 TP
  - Code simple test software (Serial monitor printing, button presses, etc.)

- Assembly Testing
  - **Solder [section] + Run with either Bench supply or Battery + measure TPs OR software testing**
  - Section Order:
    - device power supply, LiPo charger, Microcontroller, USB, Display/buttons
  - Take video of tests

- Running the Test Plans → *Show in the GDrive*

Results (for the PROJECT)

What **Worked** for us

- The output of the regulator showed +3V3
- Device display showing programmed graphics
- Device PASSED written test plans

What **didn't Work well** for us

- Bootloader failure → switched to +5V from +3V3
- Display wasn't functioning properly → Solder/close some pads on the 2.8" TFT display board to use SPI mode
- Problem accessing pins in software → Fixed pin naming in Arduino IDE

Results (for the TEAM)

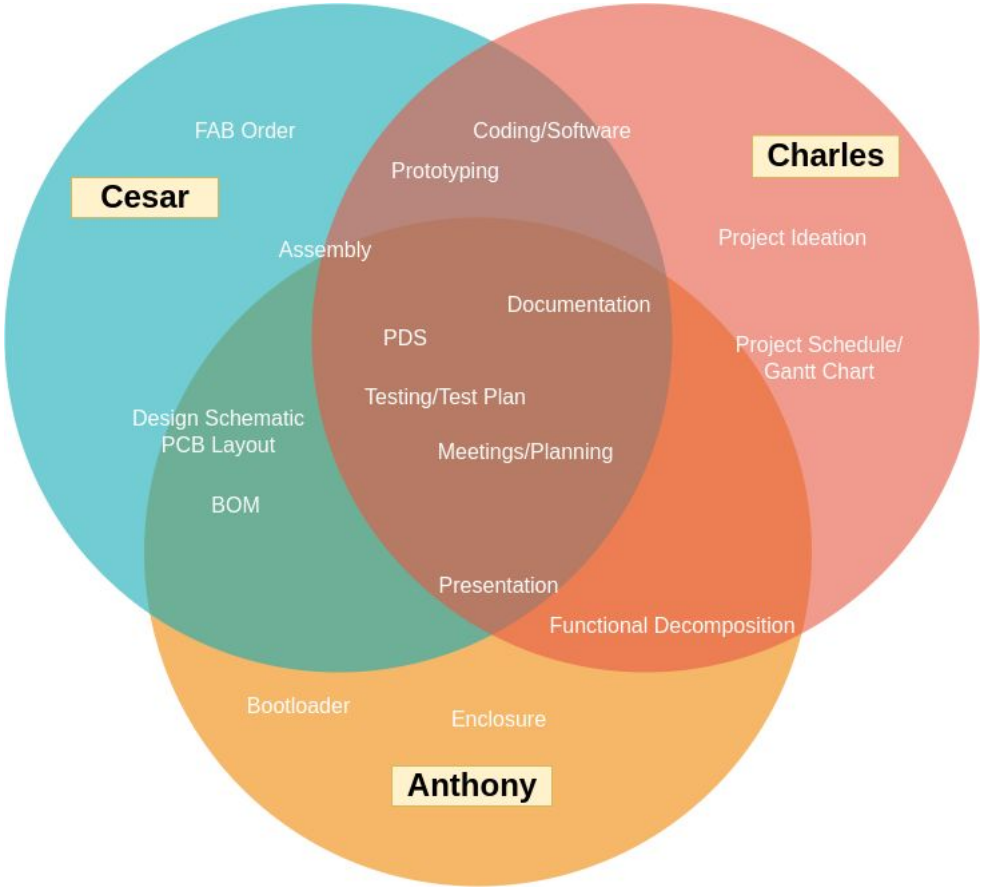
What **worked** for us

- Group communication messaging system (Discord)
- Regular scheduled meetings that fits with our schedules
- Checking each other's work

What **didn't work** so well for us

- Group of 3 (4 is better)
- Working on deliverable assignments day-before due date
- Extra meetings, difficult to schedule

Contributions



Lessons Learned

- Communication in a project with a short development time is **CRITICAL**
- Help check each other's work to verify
- Prototype early, Prototype with better parts

Time Machine - done differently……

- Add more memory w/ a different microcontroller
- Add more output features such as sound
- Add a battery indicator to the LCD
- Easier interface for users to add their code

Alright….. Let’s Demo
Demo Video if necessary

Let’s look at the Collaboration site

Link Here