CSE 450/453

Shooter Game Controller System

Game Development Team

Team Members:
Herbert Acheampong
Nathan Fox
Osmond Wan

GitHub Repo

Table of Contents

GitHub Repo

How the system works

There are two controllers that are part of the system:

1. Movement Pad
    - Controls the character's movement input actions such as walking/sprinting forward, backward, left, right, crouching, and jumping.

2. Gun Controller
    - Controls the player's camera and crosshair. Also controls the player's inventory management, interactions with the environment, and shooting.

Usability:

1. Movement Pad
    - Players step on the movement pad panels to move the in-game character.

2. Gun Controller
    - Players hold the gun controller using both hands. One hand on the barrel of the gun and the other holding the handle of the gun.

    - Players move the camera left and right by manipulating the yaw of the gun.

    - Players move the camera up and down by manipulating the pitch of the gun.

    - Players press the buttons on the grip of the gun to do non-movement character interactions.

GitHub Repo

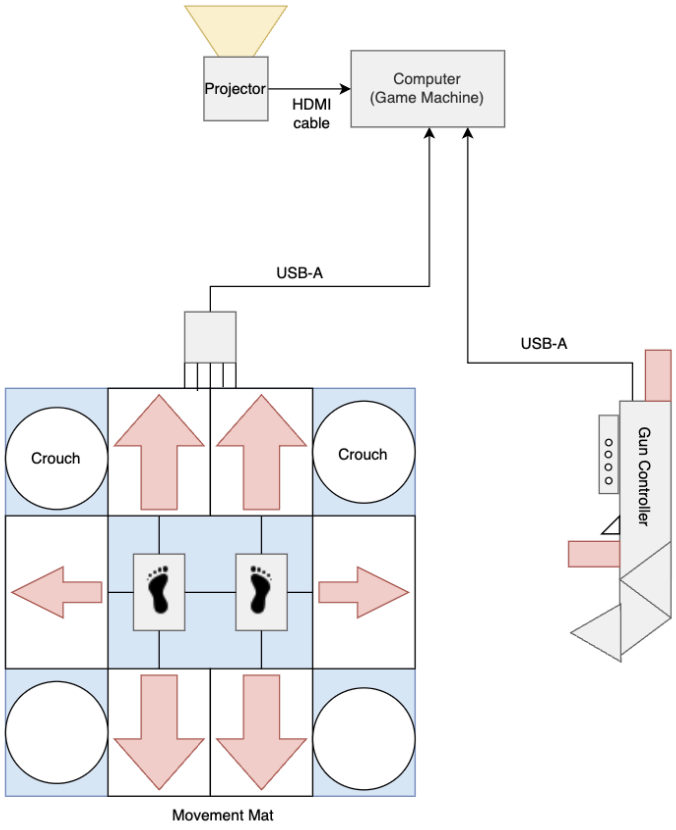
Connectivity and Inputs:

- Both controllers are connected to the computer with their own separate USB A cable.

- Movement pad uses 10 force sensitive resistors that are activated and translated to keyboard inputs corresponding to movement controls such as directional movement, sprinting, crouching, and jumping.

- Gun controller uses a gyroscope to translate physical movements to move the mouse cursor and player camera. Eight buttons on the body of the gun correspond to keyboard inputs that manage inventory management, quick swapping, and reloading. Three microswitches handle the aiming, shooting, and recalibration.

Design



GitHub Repo

Schematics

Gun Controller

1. **Microcontroller:**
    a. The circuit features an **Arduino Leonardo**, which is connected to various input/output components. The Arduino handles the main logic of the system.

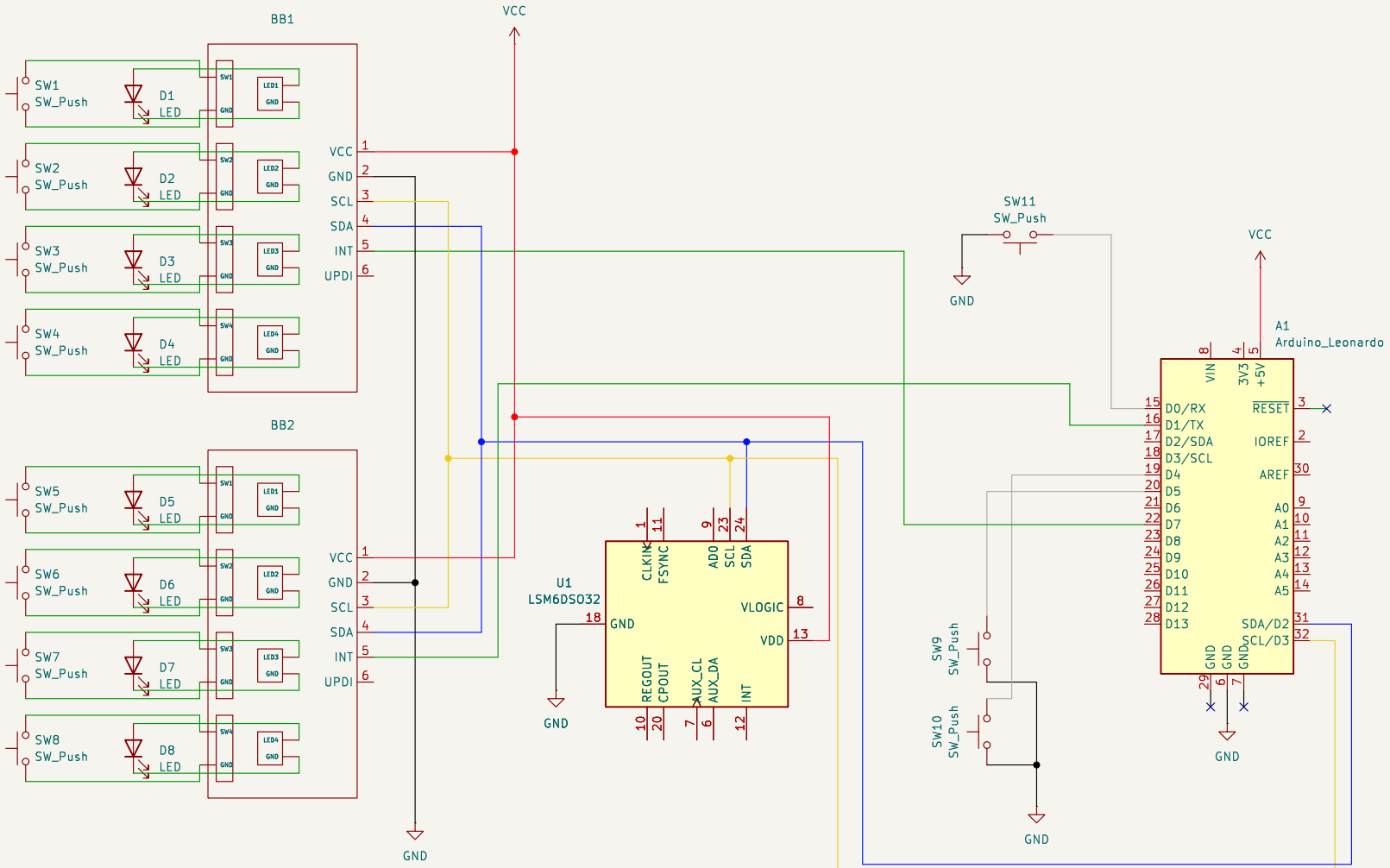
2. **Standalone Microswitches:**
    a. SW9, SW10, SW11 are standalone microswitches connected directly to the Arduino. These are designated for specific trigger functionalities, such as:
        - **SW10**: Serves as a primary trigger button for **shooting**, connected to digital **pin D4** on the Arduino.
        - **SW9**: Acts as the secondary trigger button for **aiming down sights (ADS)**, connected to digital **pin D5** on the Arduino.
        - **SW11:** The recalibration button that helps various functions such as recalibrating the cursor, sensitivity adjustments, and software resetting the gun. It is connected to digital **pin D0** on the Arduino.

3. **Arcade Push Buttons with LEDs:**
    a. The breakout boards (BB1 and BB2) feature buttons SW1–SW8, each paired with an LED (D1–D8) for visual feedback.
    b. SDA pin on the breakout boards connects to **pin SDA/D2**
    c. SCL pin on the breakout boards connects to **pin SCL/D3**
    d. Functionality:
        - **SW1** (Inventory): Hold to open the inventory menu.
        - **SW2** (Hands): Switching to hand mode.
        - **SW3** (Pistol): Selects the pistol.
        - **SW4** (Rifle): Selects the rifle.
        - **SW5** (Interact): Used for general interaction with objects or the environment.

- **SW6** (Reload): Reloads the active weapon.
- **SW7** (Rope): Takes out rope.
- **SW8** (Flashlight): Select the flashlight on.

4. **Accelerometer and Gyroscope:**
    a. **LSM6DSO32**: A 6-DoF motion sensor used for detecting movement and orientation in space. It connects to the Arduino via the SDA and SCL **pins (D2 and D3)** for I2C communication.

5. **Power Source:**
    a. The Arduino provides 5V power to the circuit through the USB connection from the computer.



Sheet: /
File: guncontroller.kicad_sch

Title: Gun Controller

Size: A4	Date:	Rev:
KiCad E.D.A. 8.0.6		Id: 1/1

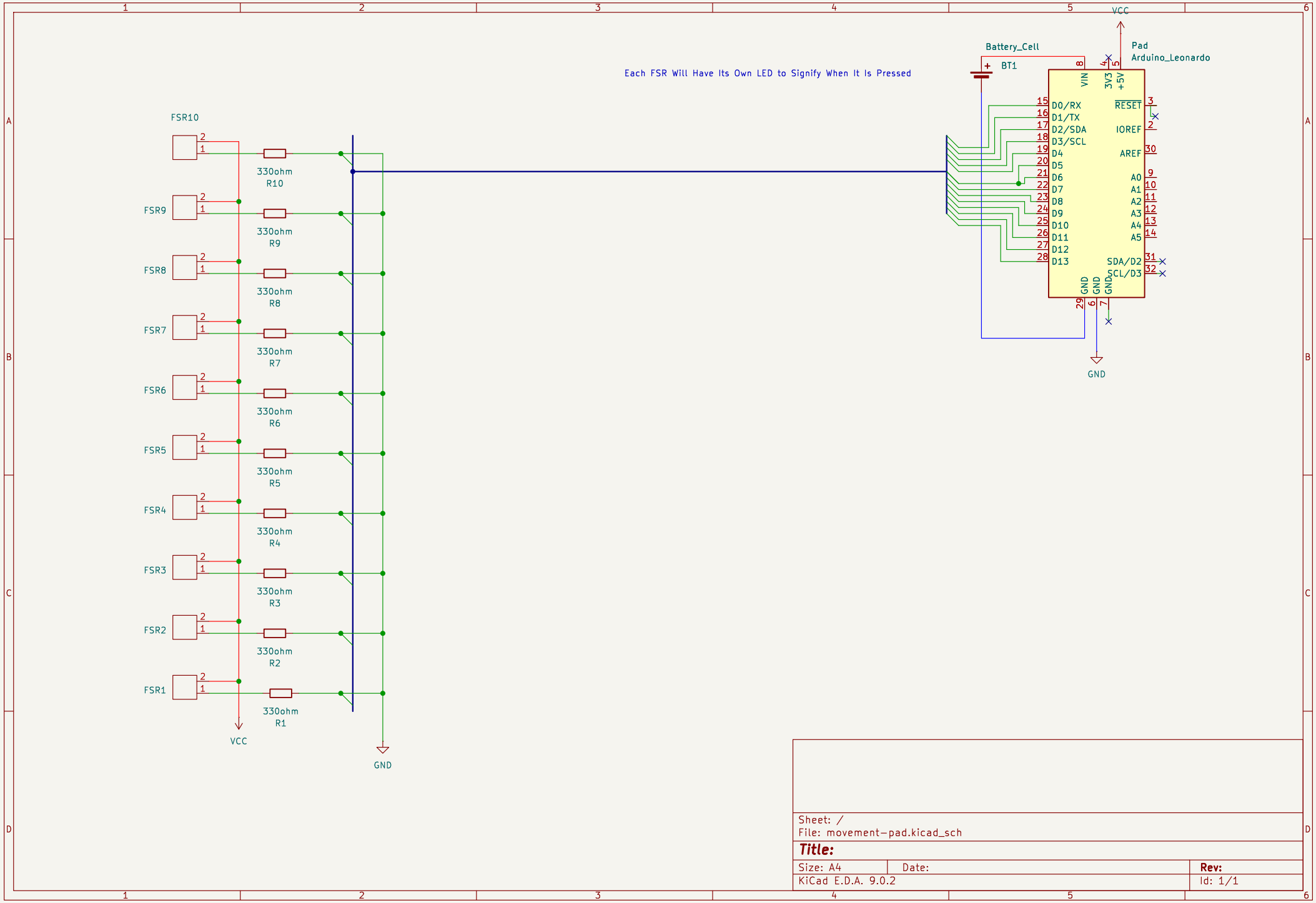
GitHub Repo

Movement Pad

1. **Microcontroller:**
    a. The circuit features an **Arduino Leonardo**, which is connected to various input/output components. The Arduino handles the main logic of the system.

2. **FSR and Connections:**
    a. A **Force-Sensitive Resistor** (FSR) is a sensor whose resistance decreases as pressure increases.
        - One terminal of each FSR is connected to analog **pins A0 to A10** of the Arduino Leonardo.
        - The other terminal of the FSR is connected to a 330-ohm resistor, which is then connected to GND.

3. **Power Source:**
    a. The Arduino provides 5V power to the circuit through the USB connection from the computer.


Each FSR Will Have Its Own LED to Signify When It Is Pressed
FSR10
FSR9
FSR8
FSR7
FSR6
FSR5
FSR4
FSR3
FSR2
FSR1
330ohm
R10
330ohm
R9
330ohm
R8
330ohm
R7
330ohm
R6
330ohm
R5
330ohm
R4
330ohm
R3
330ohm
R2
330ohm
R1
VCC
GND
Battery_Cell
BT1
Pad
Arduino_Leonardo
VIN
3V3
+5V
D0/RX
D1/TX
D2/SDA
D3/SCL
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
RESET
IOREF
AREF
A0
A1
A2
A3
A4
A5
SDA/D2
SCL/D3
GND
GND
GND
Sheet: /
File: movement-pad.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. 9.0.2
Id: 1/1

GitHub Repo

Overview of the Code

Gun Controller

Code found here.

1. Initialize the pins, interrupts, and breakout boards in setup().

2. In the main loop, loop(), the arcade breakout board interrupts are checked to see if any button was pressed. If the interrupt has been set, go into the helper function checkButtons().

3. In checkButtons(), the states of the buttons on the breakout board are read and depending on the button pressed and which breakout board triggered the interrupt, the corresponding keyboard input is pressed.
    - If a button is pressed, turn on its LED for the duration that it is being held.

4. Back to loop(), if the recalibration button is held, don't move/update the mouse cursor.
    - The arcade buttons are still functional in this state.
    - In this state, the primary and secondary fire triggers are used to control the sensitivity of the gun.
        - Default sensitivity is 0.25 and increases/decreases in increments of 0.05.
        - The minimum sensitivity is 0.05 and the maximum is 3.
    - If all three buttons (recalibration, primary, and secondary) are pressed down, the controller will perform a software reset. Call the reboot() function to do this.

5. In reboot(), set up a watchdog timer to wait and reset the Arduino. Software reset the arcade button breakout boards and read from the GPIO register INTFLAG to clear the remaining interrupts that may be left over. Call setup() to make sure all components are set up again.

6. If the recalibration button is **NOT** held down, go into the helper function updateMouseMovement() to move the mouse based on the movement of the gun.

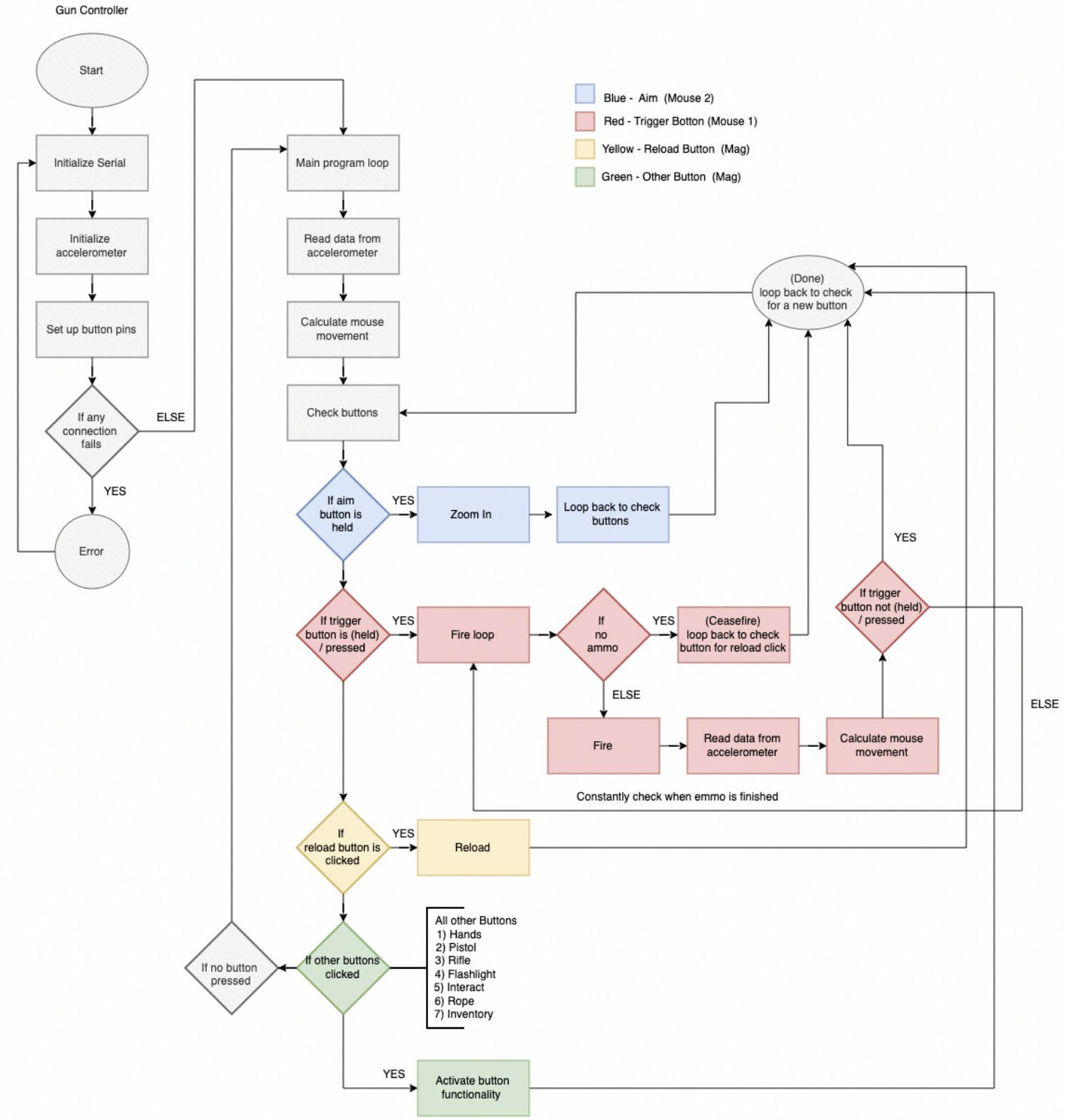
7. In updateMouseMovement(), read the raw gyroscope data from the IMU, apply deadband filtering to smooth the cursor from the shakiness of holding the controller, apply scaling and the sensitivity, and then move the mouse.

8. Back to loop(), check if primary and secondary triggers are pressed and press mouse1 or mouse2 accordingly.

9. End of loop(). Restart loop() and check again, polling for changes in the buttons and triggers.

[GitHub Repo]

Pseudo Code Flowsheet - Gun Controller



GitHub Repo

Movement Pad

Code found here.

Input scanning

The code scans through all 10 FSRs, and stores the state in an integer intended to be read in binary or hex. If the analogue input of the FSR is above a specific threshold, defined in conf.hpp. The number has a total of 10 bits that can change, the 10th bit corresponding to the FSR connected to A9, and the 1st bit corresponds to A0. Below is a diagram that shows what FSR corresponds to what panel on the board, where the top of the diagram is the front of the movement pad. There are also markings identifying what each cell is below the acrylic panels and on the underside of the pad.

Unfortunately, there are only 6 analogue pins that are labeled on the Arduino Leonardo even though it supports up to 12 analogue inputs. This is because some of the digital ports double as analogue ports. Specifically, digital pin 4 is A6, pin 6 is A7, pin 8 is A8, pin 9 is A9, pin 10 is A10 and pin 12 is A11. Only pins 4 to 9 are used[1].

1: **ADDENDUM:** Due to the wire breaking inside of Pin 9, Cell 9 has been moved to port 10/A10.

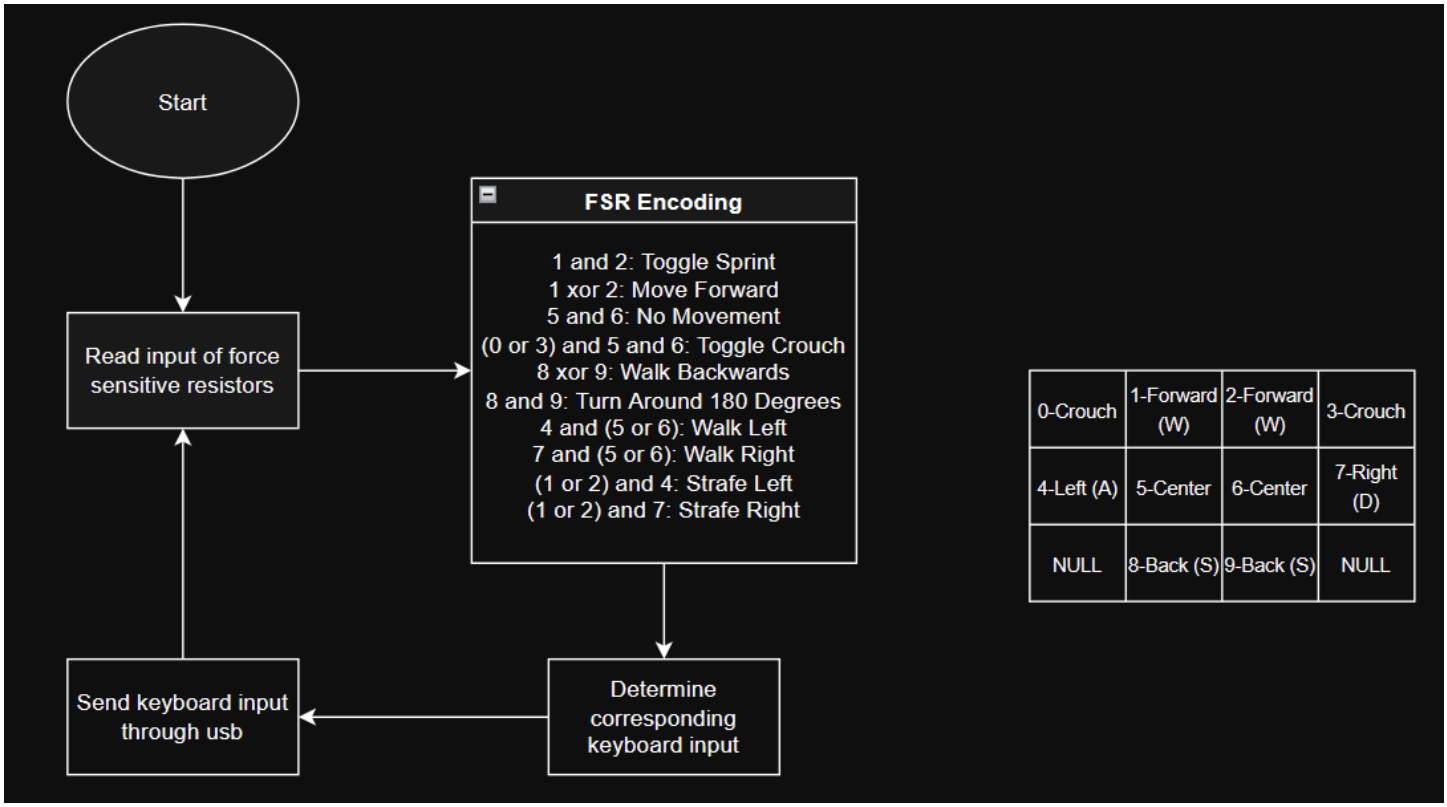
0-Crouch	1-Forward (W)	2-Forward (W)	3-Crouch
4-Left (A)	5-Center	6-Center	7-RIght (D)
NULL	8-Back (S)	9-Back (S)	NULL

GitHub Repo

Configuration

Basic configuration can be done in conf.hpp. The threshold value can be configured, and the cells needed to cause a specific action can be changed as well within the define statements. The code is mostly device agnostic as well. To define a new device to compile the code to will require updating the platformio.ini file. This document will not discuss the myriad options provided by Platformio, but you can find that here.

Pseudo Code Flowsheet - Movement Pad



GitHub Repo

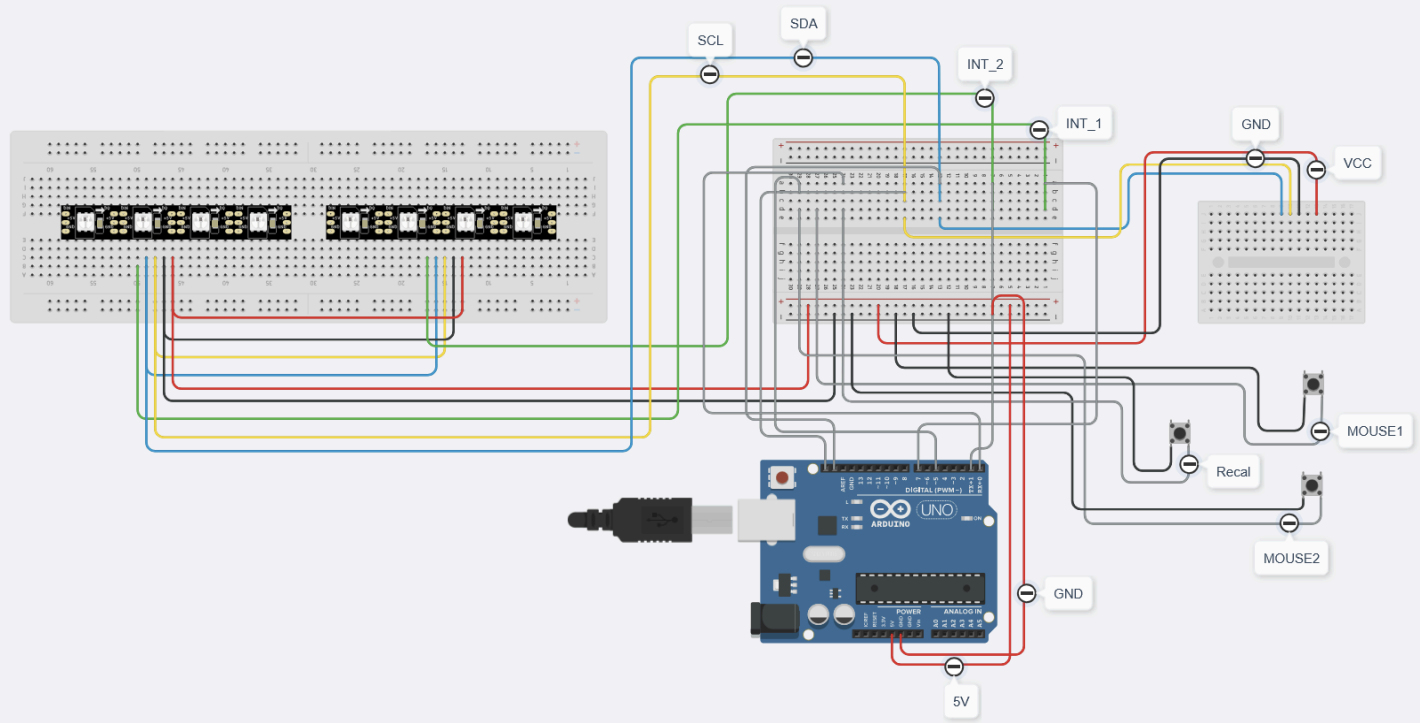
Wiring

Gun Controller Wiring

Wiring for the gun controller is a little bit more complicated than the movement pad due to the gun using more components. For this reason, included below is a wiring diagram of the components in the gun following the wire color convention table below. **Please note that the wire color scheme is specific for the gun controller**.

Color	Function
Red	VCC
Black	Ground
Blue	SDA
Yellow	SCL
Green	Interrupts
White	Normally Closed (NC) terminals on the microswitches

[GitHub Repo]



Looking at the wiring diagram above, take note of the positions of the breadboards and microswitches. These components correspond to how they are placed in the actual controller. The long breadboard refers to the longer protoboard that holds the arcade button breakout boards. The medium sized breadboard refers to the medium protoboard that holds all the wire connections from all the components in the gun. The smaller breadboard refers to the small protoboard that holds the IMU.

A quirk you may have noticed from the wiring diagram is that GND and 5V from the Arduino Leonardo (represented as an UNO in the diagram) are red and the wires coming out from the non-power pins are white. This is the case because the only colors of solid-core wire that was available at the time were red and white. Using stranded wire inside the Arduino's pin headers was not secure enough so therefore 20 AWG solid-core wire was preferred and used.

The wiring for all the components in the wiring diagram is accurate and exactly the same as inside the gun. To get a better view at which wires are connected to which rows/positions inside the gun, open this Tinkercad project and zoom in until the row numbers are visible.

GitHub Repo

Movement Pad Wiring

Each cell has a red wire that corresponds to the Arduino Leonardo's data pin for that cell. For example, there is a red wire labeled with a number on some masking tape that says it is from Cell 8. Therefore, it must go to A8. For testing and debugging this can either be connected to the Arduino or a separate battery like the 12V Dewalt battery in which you would use a multimeter to see the voltage value increase as more pressure is added on the FSR. All the white wires go to ground.

As a summary, the wire coloring scheme is shown below:

Color	Function
White	Ground
Red	FSR to Arduino analog pin

GitHub Repo

Build

Gun Controller

- The gun shaped shell used was the Nerf Fortnite IR Motorised Blaster.
- The flywheel and the electrical components of the blaster were removed to make space for wires, protoboards, and Arduino.
- The magazine and magazine well was removed so the space could be used for the Arduino.
- Plastic support posts were removed to make sure there was enough space for the 3D printed parts to fit so that there was no obstruction for the glue to adhere.
- The secondary trigger lock mechanism was removed so that the primary trigger could be pressed in without needing to hold down the secondary trigger. This also allowed the primary trigger to be pressed whenever the “chamber door” was pulled back.
- The mag release, primary trigger, and secondary trigger had their functionalities removed and replaced with the microswitches.

Movement Pad

- The movement pad is constructed from a 3’ by 4’ slab of particle board and supported underneath by multiple 2” by 3” pieces of wood, placed roughly under the center of where the acrylic panels would go and under where the force sensitive resistors are placed. Half-inch holes were drilled 5.5 inches from each side into where the 11 inch by 11 inch acrylic would sit to create a channel for the tail of the FSR to be tucked into.
- Additionally, several small wooden rectangles were cut to fit in between each acrylic panel, and several small wooden squares were cut to be placed on top the panels to secure them to the board.
- To mitigate pressure loss between the acrylic panel and the FSR, 1/8th inch wooden rectangles were cut and taped on top of each FSR.
- Channels were cut in the 2 by 3s to provide wire management

GitHub Repo

Future Development

If this project was to continue development, below are features that the team thinks could be added or issues that need to be fixed.

Gun Controller

- Using absolute mouse movement rather than relative mouse movement.
- Incorporating an IR sensor and IR light bar to help tracking and keep the mouse cursor calibrated.
- Switch to another microcontroller that supports absolute mouse positioning and relative mouse positioning at the same time.
- Switch to another microcontroller that supports better interrupts and has better tools to deal with interrupt conflicts.
- Use quaternion data from a 9-DOF IMU such as the BNO-088. This will help with implementing minimal drift absolute positioning.

Movement Pad

- Needs a complete redesign from the construction angle.
    - Pressure on each FSR is too low and too inconsistent, causing missed inputs or fake inputs to occur.
    - Issue can be somewhat mitigated in software by adjusting threshold values or by choosing different resistor values, but the problem lies fundamentally with the acrylic panels and the large wooden base

GitHub Repo

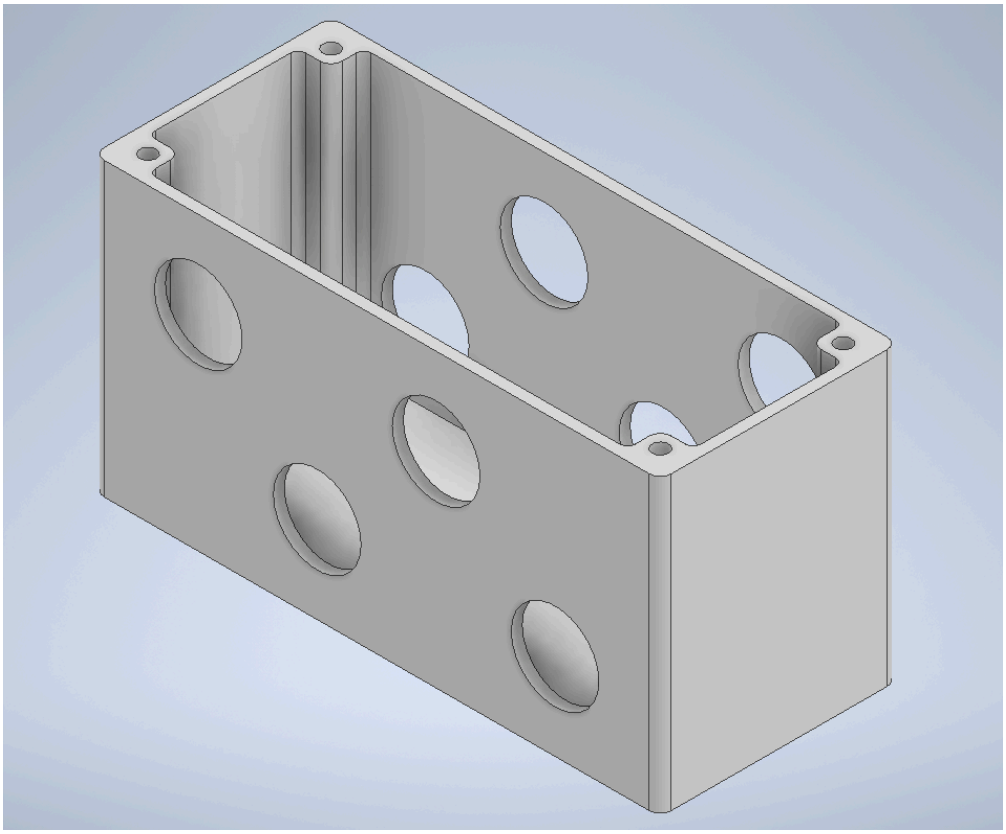
Miscellaneous

To secure the components inside the nerf gun, supports and enclosures were 3D-printed so that the parts can be attached firmly but also be removable if there is any need to diagnose or replace the parts.

The specific parts that were modeled were the box that the arcade buttons are attached to, supports for the three sized protoboards, support for the Arduino Leonardo, and supports for some of the microswitches to raise their elevation to the height where the nerf gun triggers can interact with them.

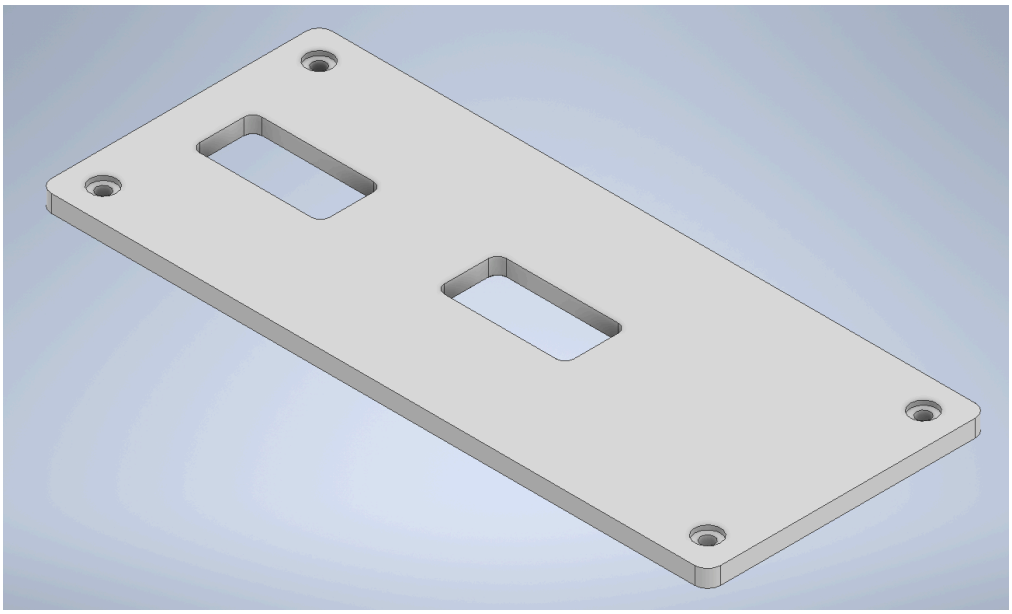
Below are pictures of the CAD models made. To find more information about these CAD models, the STEP and STL files are located in the “3D_Models” directory in the GitHub repository.

GitHub Repo



arcade_box_iteration3.stp

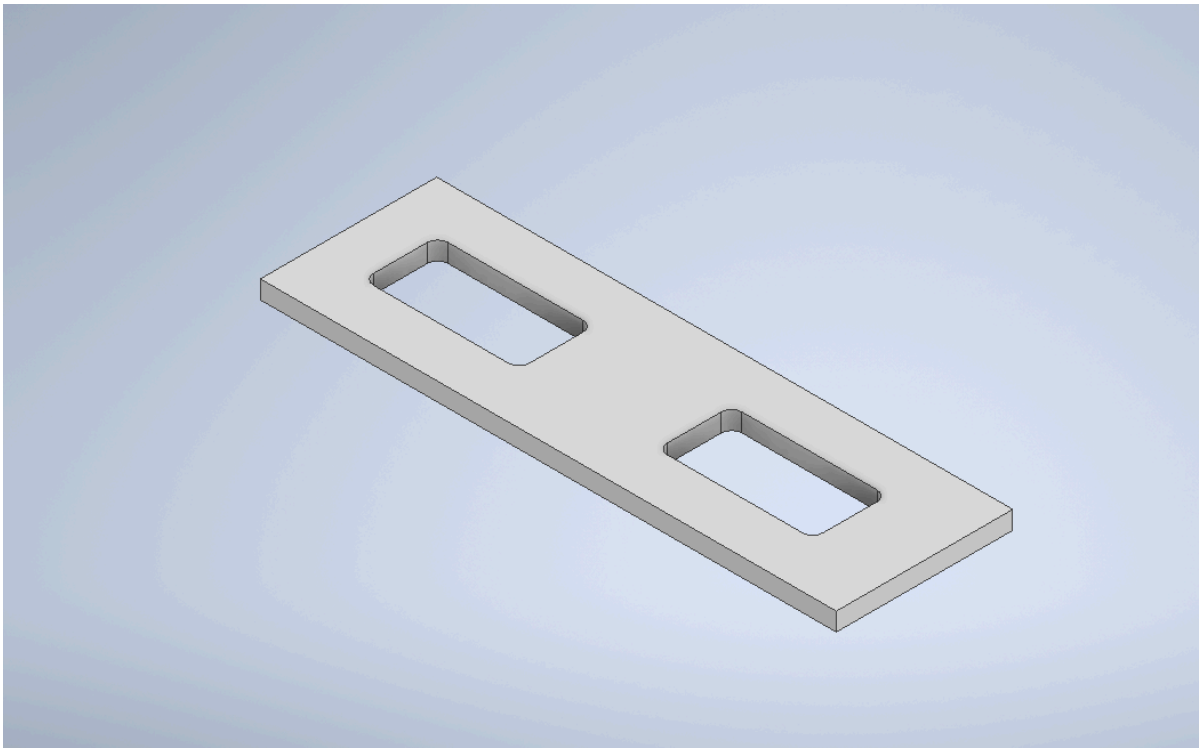
Box to hold the arcade buttons with holes for M3x4x5 heat inserts



arcade_box_lid_iteration3.stp

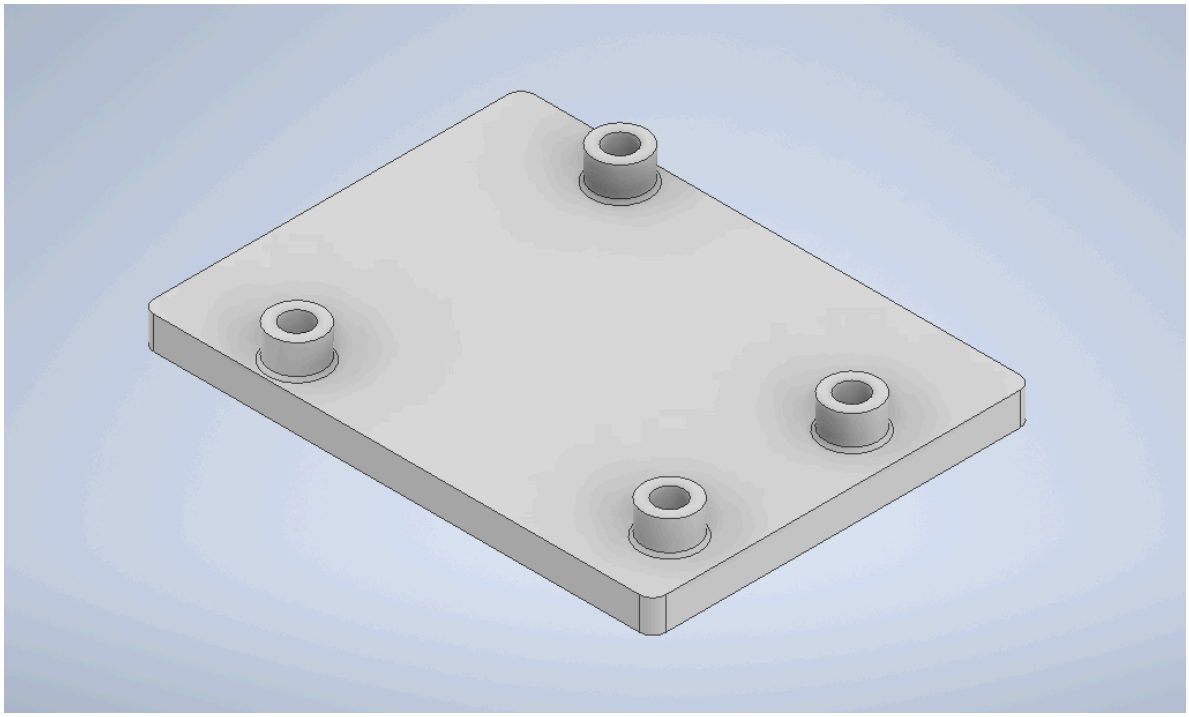
Lid to the arcade button box that has holes for M3 screws

GitHub Repo



arcade_lid_extrusion_3.5.stp

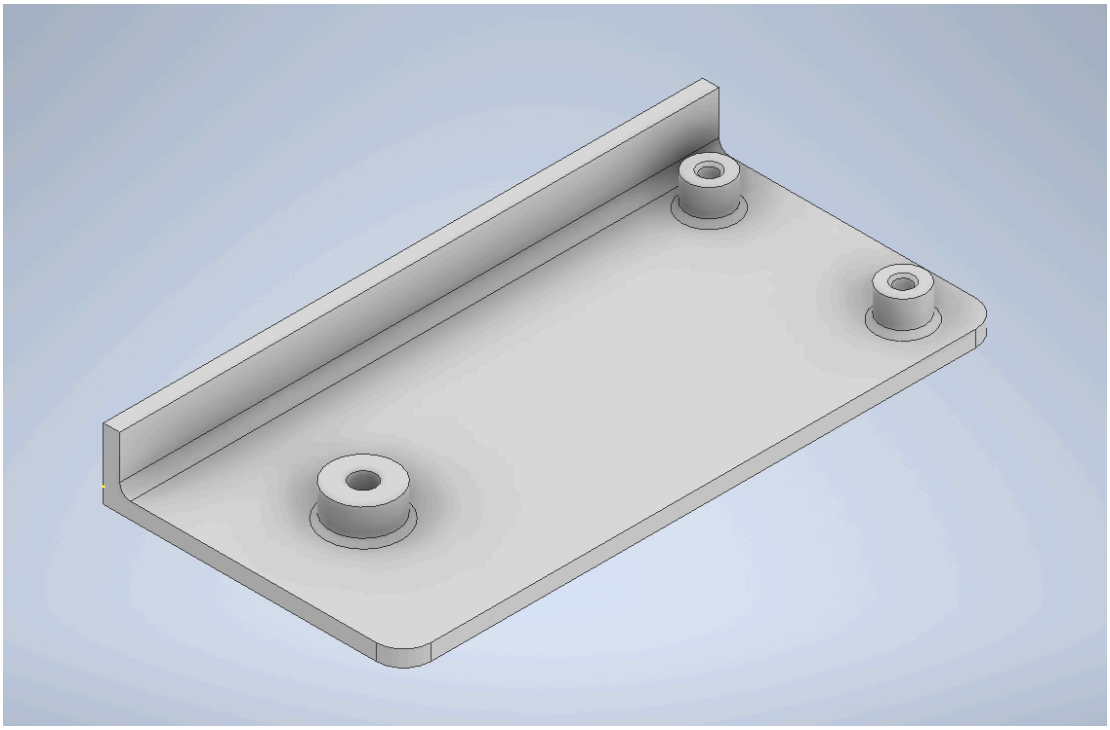
Placed on top of lid to increase gap between lid screws and body of nerf gun



leonardo_backplate.stp

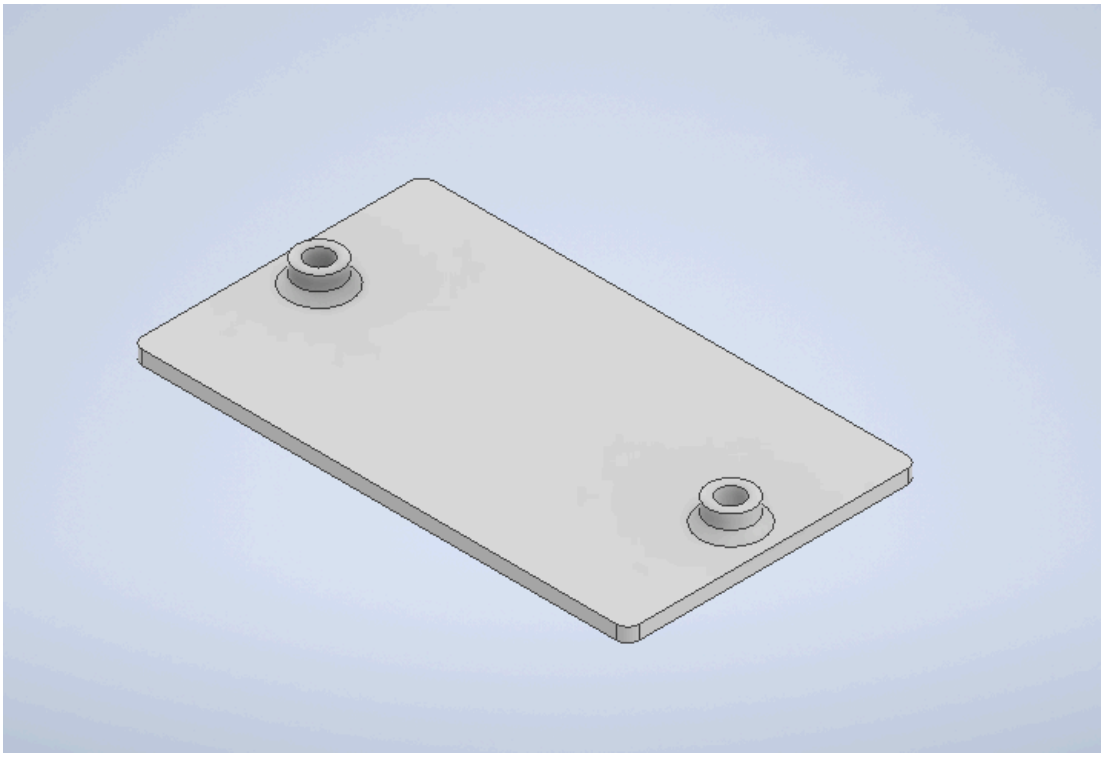
Arduino Leonardo backplate with holes for M3x4x5 heat inserts

GitHub Repo



long_backplate.stp

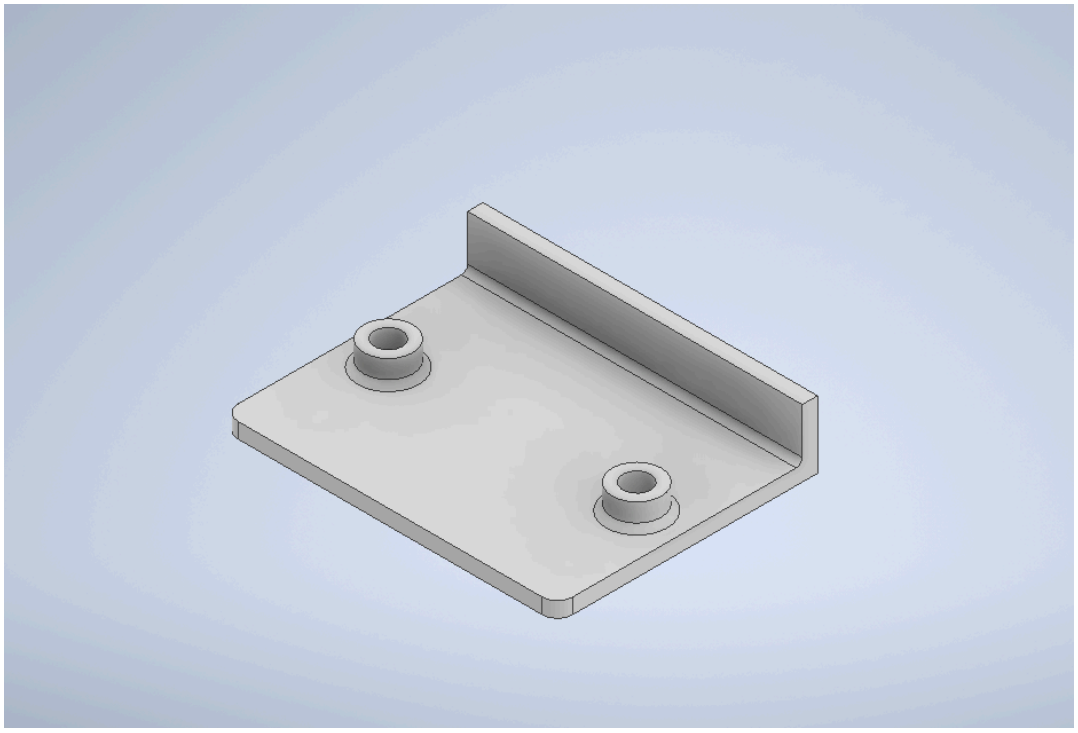
Long protoboard backplate with holes for M3x4x5 and M2x3x3.5 heat inserts



medium_backplate.stp

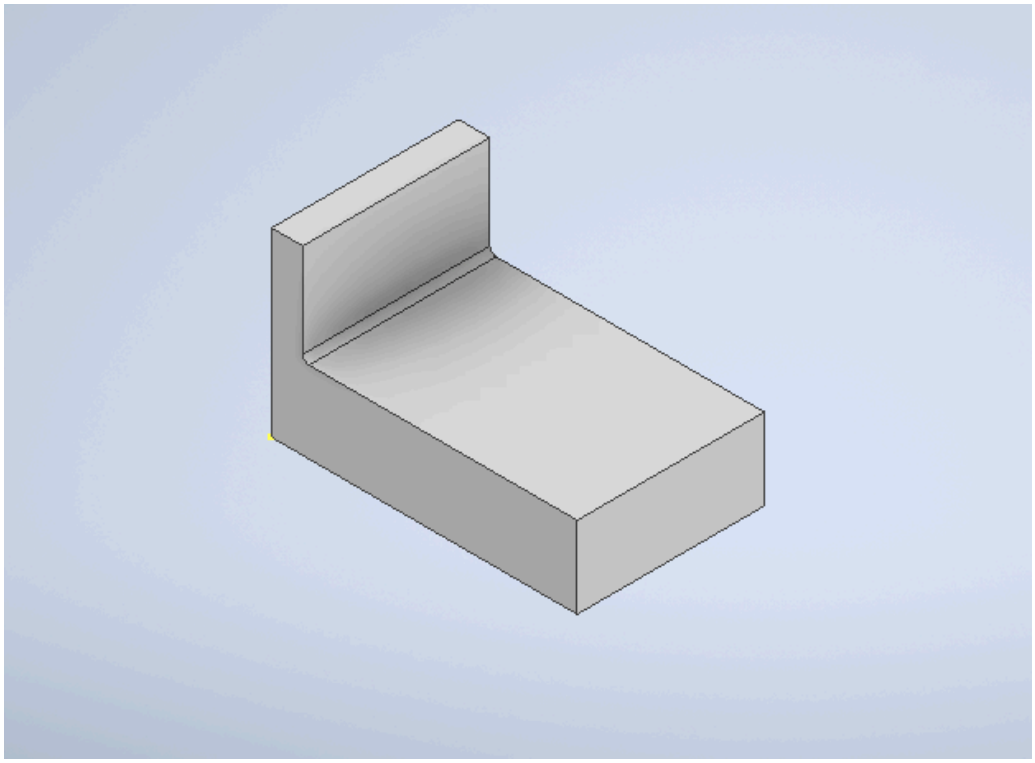
Medium protoboard backplate with holes for M3x4x5 heat inserts

GitHub Repo



small_backplate.stp

Small protoboard backplate with holes for M3x4x5 heat inserts



small_microswitch_holder.stp

Microswitch holder used to raise its height when secured in the nerf gun

GitHub Repo

Bill of Materials

Part Description	Quantity	Unit Cost	Cost *(excluding tax + shipping)*
Arduino Leonardo with Headers	2	$24.90	$49.80
24mm Mini LED Arcade Button - Clear	3	$2.50	$7.50
24mm Mini LED Arcade Button - Yellow	2	$2.50	$5.00
24mm Mini LED Arcade Button - Red	1	$2.50	$2.50
24mm Mini LED Arcade Button - Blue	1	$2.50	$2.50
24mm Mini LED Arcade Button - Green	1	$2.50	$2.50
Arcade Button 1x4 Breakout Board	2	$9.95	$19.90
Arcade Button Wires - 0.11”	2	$4.95	$9.90
Adafruit LSM6DSO32 6-DoF Accelerometer and Gyroscope - STEMMA QT / Qwiic	1	$12.50	$12.50
Lever Micro Switch	2	$1.50	$3.00
10PCS Mini Micro Switch 3Pin with Roller Limit Switch	1	$10.97	$10.97
Force Sensitive Resistors - Square	10	$5.99	$59.99
Pink DDR Arrow Panel	5	$39.99	$199.95
Blue DDR Arrow Panel	5	$39.99	$199.95
ElectroCookie Solderable Breadboard Large PCB Board for Electronics Projects Compatible for DIY Arduino Soldering Projects, Gold-Plated (3 Pack, Blue)	1	$9.49	$9.49

GitHub Repo

ElectroCookie Solderable Breadboard PCB Board for Electronics Projects Compatible for DIY Arduino Soldering Projects, Gold-Plated (5 Pack + 1 Mini Board, Blue)	1	$8.49	$8.49
Box Enclosure-PTQ-11055-C	1	$35.80	$35.80
15ft Micro USB cable	1	$7.99	$7.99
Strain Relief Cord Boot Protector	1	$6.59	$6.59
			Total
			$654.32