Operational Assembly Manual

Robotics Traveling Van (RTV) Robot 1 and Robot 2

Created by:

Florence Fasugbe
Andres Gonzales
Colin Parsinia
Freddy Rivera

Project Completion Date: April 30, 2026

Contents

1. Disclaimer

This report was prepared by students as part of a university course requirement. While considerable effort has been put into the project, it is not the work of licensed engineers and has not undergone the extensive verification that is common in the profession. The information, data, conclusions, and content of this report should not be relied on or utilized without thorough, independent testing and verification. University faculty members may have been associated with this project as advisors, sponsors, or course instructors, but as such they are not responsible for the accuracy of results or conclusions.

2. Introduction

The Acro-Bots Operational and Assembly Manual is intended to serve as a comprehensive reference for anyone responsible for assembling, operating, maintaining, or troubleshooting the **Robotics Traveling Van** (RTV) robot systems developed as part of the Northern Arizona University (NAU) Mechanical Engineering Capstone program for the 2025–2026 academic year.

This manual covers two robots developed for client Dr. Michael Shafer of Northern Arizona University (NAU). Both robots are designed to demonstrate principles of feedback control to K-12 students in an engaging, hands-on outreach setting. Robot 1, known as the **Inverted Pendulum Cart**, demonstrates dynamic mobility using a self-balancing cart system. Robot 2, known as the **Ball-on-Beam**, demonstrates closed-loop PID control through a beam-tilting system that balances a ping-pong ball at a user-defined position.

This manual is written for a technically capable reader who may not have a formal engineering background, someone comfortable with basic hand tools, 3D printing, and common soldering practices. No prior robotics experience is assumed, and each section is written in numbered steps wherever possible so that assembly, operation, and maintenance can be followed sequentially without referring back to other sections.

> **WARNING: Always power down the robot and disconnect the charger before performing any mechanical or electrical work. Never attempt to disassemble the robot while the rocker switch is in the *ON* position.**

3. Robot 1 — Inverted Pendulum System

Robot 1 is an inverted pendulum robot system. A pendulum composed of 3D printed joints and machined aluminum rods is held by a 3D printed frame. Four L-type 520 Encoder Reduction Motors mounted to the bottom frame move the system back and forth based on the commands from a PID control loop running on a Raspberry Pi Pico (RP2040) microcontroller. A Magnetic Potentiometer (AS5600) affixed to a 3D printed mount is held parallel to a small magnet mounted in the central joint of the pendulum, reading the pendulum's current angle with less than 5% error, and is concealed by a TPU shell screwed onto the Top Frame. The system is powered by a 14.8 Volt LiFePO4 battery array and operates fully untethered for approximately 65 minutes on a single charge.

Major Sub-Assemblies:

- Bottom Frame – Mounting for BMS, L-type Motors, Top frame; contains RP2040, LiFePO4 battery pack, BMS, acrylic window.
- Top Frame – Mounting for Pendulum and bearings, Magnetic Potentiometer (AS5600) and 3D printed mount.
- Pendulum – Comprised of 2 3D printed shoulder brackets, a central U-shaped 3D printed Bracket, 4 segments of machined 6mm aluminum rod, and a small magnet.
- TPU Pendulum Shell – Provides protection against damage to structure/internal systems from falls, encloses magnetic potentiometer and associated wiring.

Required Tools:

- Soldering iron (set to 200°C for heat-set inserts)
- M3 hex drivers or Allen keys
- Needle-nose pliers

- Wire cutters and strippers
- 3D printer (FDM, PLA-capable) with slicer software
- Ruler or calipers

WARNING: LiFePO4 batteries must never be punctured, shorted, or submerged. Keep fingers and loose clothing away from moving parts when the robot is powered on. Do not operate on the edge of a table.

3.1 Maintenance

3.1.1 Routine Maintenance

Perform the following checks before each use:

1. **Fastner Inspection –** Check all M3 screws and inserts on Touchscreen Mounts, Motor Brackets, Motors, Wheels, Top Frame, Magnetic potentiometer mount, and TPU shell. Tighten any that feel loose. Do not overtighten. Threaded Inserts can detach from PLA Frame. If Inserts detach from frame, re-press and re-set the threaded insert using soldering iron at 200°C.
2. **Pendulum Arm Inspection –** Check that pendulum is rotating without resistance and that pendulum arms are aligned. In event of resistance to rotation, inspect contact point between shoulder brackets and top frame for obstructing materials or friction, inspect sealed bearings for damage, and aluminum beam segments for deformation. In event of pendulum arm misalignment, inspect 3D printed brackets for deformation in sockets or damage to machined aluminum beam segments.
3. **Magnetic Potentiometer Inspection –** Check that Magnetic Potentiometer mount for any damage or bending. System will not read pendulum angle properly if magnetic potentiometer is not held parallel to magnet mounted
4. **Structural Integrity Inspection –** Examine 3D printed frame and acrylic window for cracks/deformation. Preexisting damage to frame can result in irreparable damage to electrical components and total system failure in event of fall during operation.

3.1.2 Electrical Maintenance

The following covers electrical maintenance checks.

1. **Cable routing inspection** — Check that the magnetic encoder inside of the top section housing is still in place, and is connected.
2. **Charging port inspection** — Confirm the charging cable is secure, and the port is not loose in its mounting hole. The charging cable connects directly to the P+ and P- terminals on the BMS.
3. **Battery voltage check** — Before each use, verify that the battery is adequately charged. Do not operate the robot if battery voltage has dropped below 12.0V under load. Do not allow individual cells to drop below 3.0V to 3.2V per cell. The BMS is installed to help prevent over-discharge but monitor regularly.
4. **BMS visual check** — Confirm the BMS board is installed upright inside of the main body section of the robot. Ensure no visible burn marks, swelling, or corrosion on the board or terminals.
5. **Firmware check** — Verify the Raspberry Pi Pico is running the latest firmware. The source code and default PID configuration are available in the project GitHub repository at src/configuration.hpp as BEAM_KP, BEAM_KI, and BEAM_KD. Current factory defaults: Kp = 0.75, Ki = 0.30, Kd = 0.05.

3.2 Assembly

<u>3.2.1 3D Printing Specifications</u>

Unless otherwise specified all components are made of standard PLA filament using [20]% Infill.

Frame Base:

Print with bottom of Frame Base (Large Flat side with threaded insert holes and pass through holes for motor wires) facing printing bed. **Warning:** support structures may cause threaded insert holes to become blocked by base of support structures. In event of blockage use sharp implement/drilling tool to puncture support structure base for removal with tweezers.

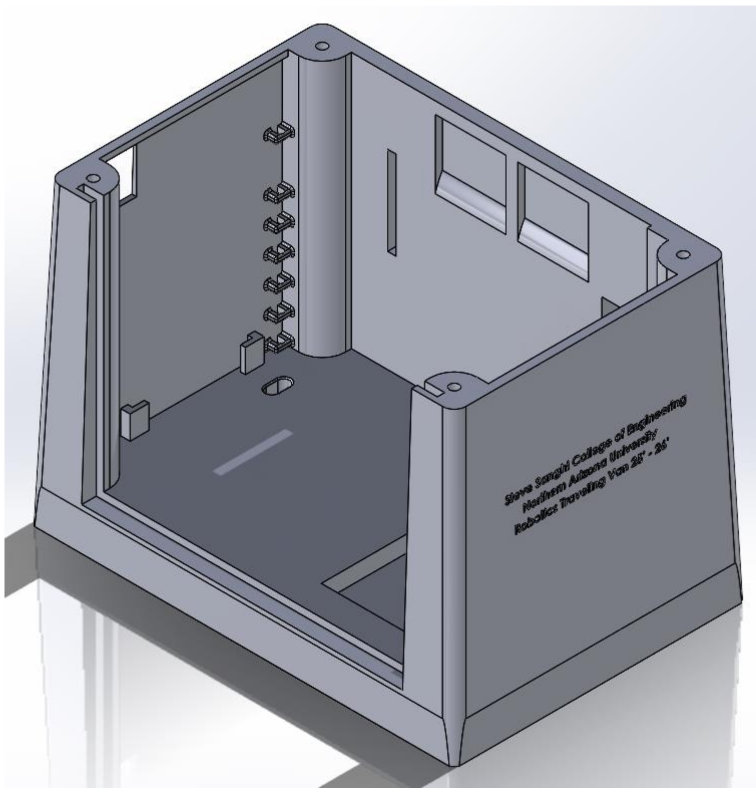

Figure 1: Frame Base of Robot 1 In upright position

Frame top:

Print with rounded pendulum support arms oriented upwards. Support structures required for large rectangular divot on face in contact with printer bed.

TPU Pendulum Shell:

Printed with TPU filament, Flat top of shell in contact with printer bed. Support structures required due to angled edges of top.

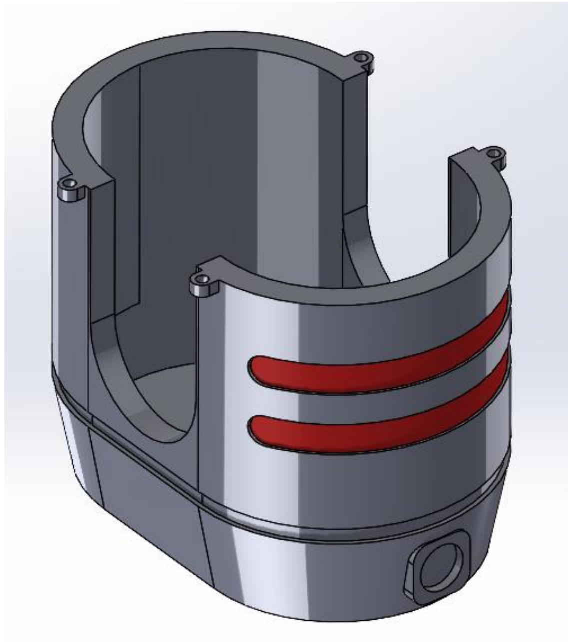

Figure 2: TPU Pendulum Shell placed upside down for printing

Pendulum Shoulder Bracket:

Print with D hole facing upwards and flat hexagonal surface with extended rectangle at one point in contact with printer bed. No support structures required.

Pendulum Center Bracket:

Print with flat surface of hexagonal sockets opposite of U structure in contact with printer bed. Supports required.

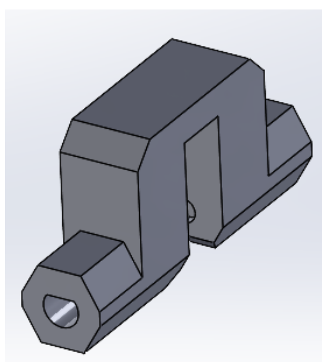

Figure 3: Pendulum Center Bracket in Proper 3D Printing position

Magnetic Potentiometer Mount:

Print with back of large square socket in contact with printer bed. Supports required for overhanging screw holes.

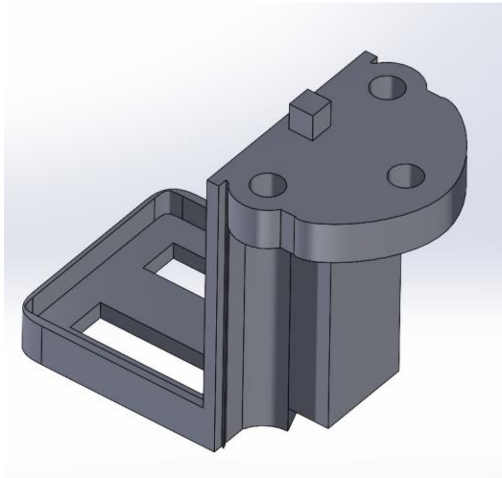

Figure 4: Magnetic Potentiometer Mount in 3D Printing Position

3.2.2 Pre-Assembly Checklist

Before beginning assembly, install all brass threaded inserts into the printed parts. Use a soldering iron set to 200°C. Let gravity assist in the insert and allow 1 full minute to cool before applying any mechanical load.

1. M3 x 5.7 mm threaded inserts: Install into all round holes on bottom and on top of each support column of Frame Base to affix motors and frame top respectively
2. M3 x 4.0 mm threaded inserts: Install into 4 small holes near base of pendulum support arms and on inner surface side of pendulum support arm with 3 holes of Frame Top, and on outer wall opposite of open side of Frame Base for touchscreen
3. 6010 6 mm bearings: position frame top so exterior side of pendulum support arm is supported by solid surface, press-fit into sockets on inner sides of pendulum support arms.
4. 3 mm cylindrical magnet (packaged with Magnetic Potentiometer) – affix into round socket on inner surface of U-shaped structure of Pendulum Center Bracket with plastic compatible adhesive

Once threaded inserts and bearings are installed, ensure all following parts and hardware are present and inspected.

Printed Parts:

- Frame Base – printed, supports removed, heat-set inserts installed
- Frame Top – Printed, supports removed, heat-set inserts installed, bearings press-fit into pendulum support arms
- TPU Pendulum shell – Printed with TPU filament, supports removed.
- Pendulum Shoulder Brackets (×2) – printed.
- Pendulum Center Bracket – Printed, supports removed, magnet affixed
- Magnetic Potentiometer mount – Printed, supports removed

Hardware:

- Yahboom L-type 520 Encoder Reduction Motors (×4)
- Yahboom L-type motor fixing bracket (×4)

- Yahboom 65 mm high friction rubber tire (×4)
- Yahboom 6 mm hexagonal coupling (×4)
- AS5600 Magnetic Potentiometer
- M3 x 5.7 mm heat-set threaded inserts
- M3 x 4.0 mm heat-set threaded inserts
- M3 x 8 mm screws
- M3 x 12 mm screws
- LiFePO4 Battery Cells (×4)
- Acrylic window panels (cut to dimension per CAD package)
- Raspberry Pi Pico (RP2040)
- BMS (Battery Management System)
- Buck converter (step-down to 3.3V)
- Rocker switch
- 4" touchscreen (ST7796S display driver)
- Wiring harness / hookup wire
- DRV8871 H-Bridge Drivers
- Short machined aluminum rod segment
- Long machined aluminum rod segment
- Pendulum Arm Rods (×2)

3.2.3 Mechanical Assembly

Starting with the Frame base

1. **Yahboom L-type 520 Encoder Reduction Motors**: start by screwing the motor fixing brackets to the bottom of the frame, then place the motors on each bracket with each shaft positioned in the bracket hole closest to the front and back of the frame and screw onto brackets. Place hexagonal coupling in each wheel slot, place onto each motor shaft, and screw until wheel is secure. Pull motor wires up through rounded slots in base of frame

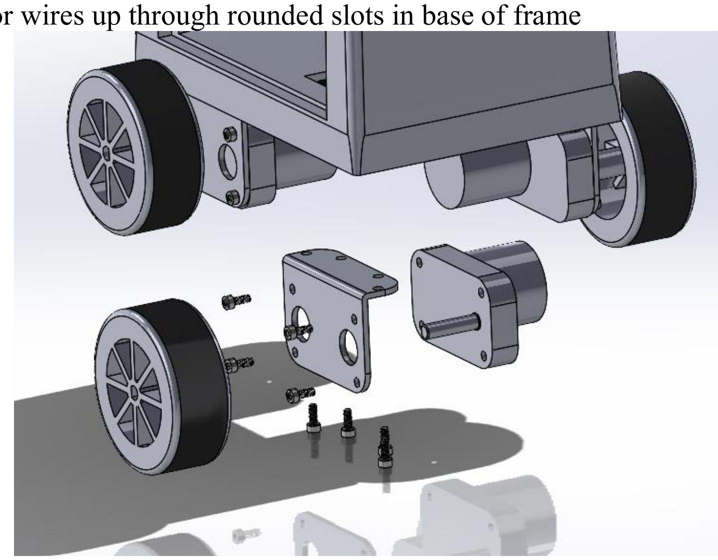

Figure 5: Single Motor Assembly Breakdown and proper wheel arrangement for Robot 1

2. **BMS and LiFePO4 battery pack** – place battery pack into square slot inside Frame Base, slide BMS into small rectangular holding arms on side opposite of battery slot, ensure all BMS sockets are accessible.

3. **Rocker Switch, Acrylic Window, and Touchscreen** – press fit rocker switch into rectangular slot located on top right of back wall of frame (same wall as BMS), slide acrylic window into thin rectangular slot on open end of open wall, screw touchscreen onto outer wall opposite of open wall with acrylic window.

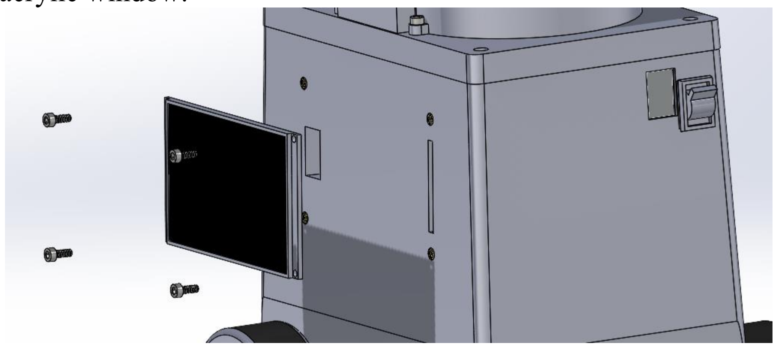

Figure 6: Touchscreen and Rocker Switch Mounting Positions

4. **Frame Top, PCB**– place PCB into rectangular slot on underside of Frame Top, then place Frame Top on Frame Bottom and screw together using 4 corner holes.
5. **Pendulum** – Start with 3D printed shoulder bracket and insert short machined aluminum segment, then insert the aluminum segment into the pendulum support arm on the top frame with 3 holes arranged in a triangle with square hole in the middle on its inner surface, then place Pendulum Center Bracket on free end of short aluminum rod. Take long machined aluminum segment and pass-through outer hole of opposite pendulum support arm and socket into Pendulum Center bracket, cap free end with other Pendulum Shoulder Bracket and insert pendulum side arms into round sockets on both Pendulum Shoulder Brackets.

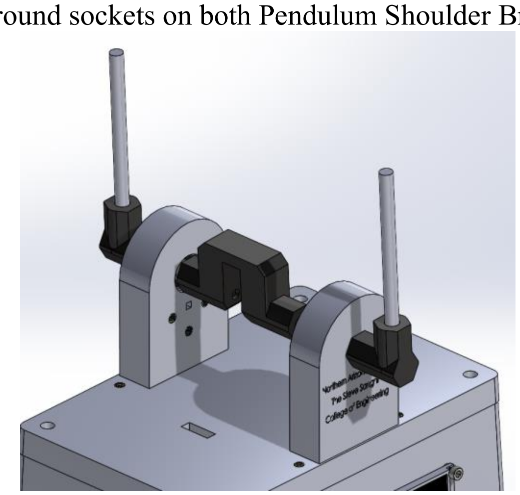

Figure 7: Fully assembled Pendulum Structure

6. **Magnetic potentiometer –** Insert magnetic potentiometer into square slot of magnetic potentiometer mount, routing wires through open slots in back of square slot, then screw magnetic potentiometer mount into inner side of pendulum support arm with 3 holes and square hole, ensuring square slot fits within U channel of Pendulum Center Bracket. Route wires of magnetic potentiometer through wire slot on Frame Top

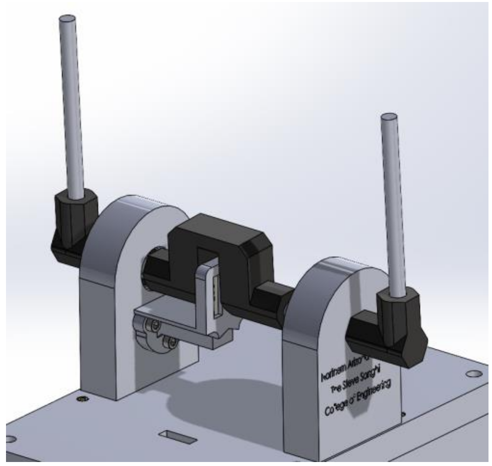

Figure 8: Magnetic Potentiometer Properly Mounted in Tandem with Pendulum

7. **TPU Shell –** slide TPU shell over Pendulum support arms of Frame Top, screw in using holes next to each side of pendulum support arms

3.2.4 Electrical Integration

This section describes the complete electrical wiring of **Robot 1 (Inverted Pendulum)** based on the confirmed pin configuration from the project GitHub repository. All GP numbers and signal names referenced below are printed directly on the physical boards; they match the labels visible on each component to the corresponding entry in the tables. GP numbers (e.g., **GP4, GP19**) are the GPIO numbers used by the firmware. Physical pin numbers (e.g., **pin 6, pin 25**) are the numbered positions on the **Raspberry Pi Pico (RP2040)** board counted from the top-left corner; for example, **GP6 is located at physical pin 9**. Signal names such as **SDA, SCL, MOSI, and IN1/IN2** come from component datasheets and describe the function of each connection for the **AS5600 Magnetic Encoder** and **DRV8871 motor drivers**.

The circuit schematic (ipr_circuit_diagram.pdf) and PCB layout are attached as screenshots below for quick reference, please refer to the full-resolution files for detailed verification.

Circuit Schematic Reference:

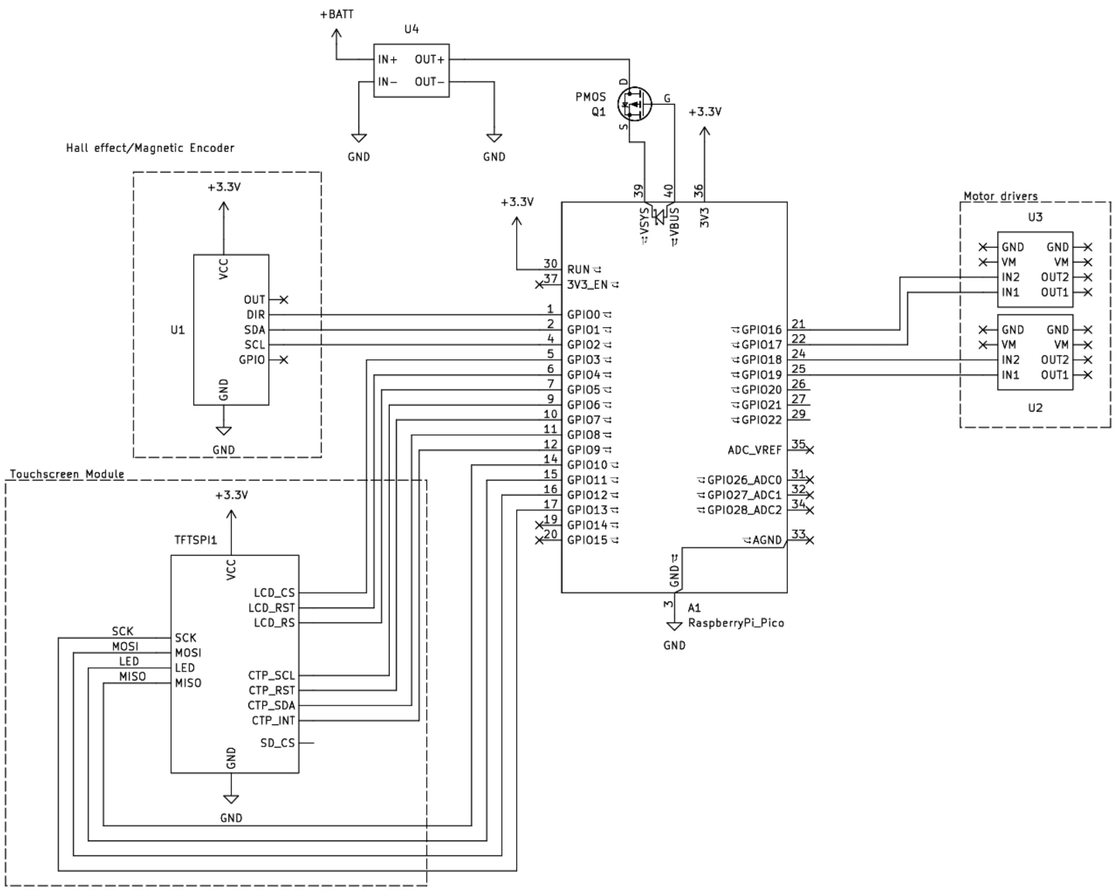

Figure 9: Circuit Schematic for Robot 2 Ball-on-Beam General Layout (BBBSchematic.pdf)

PCB Layout Reference:

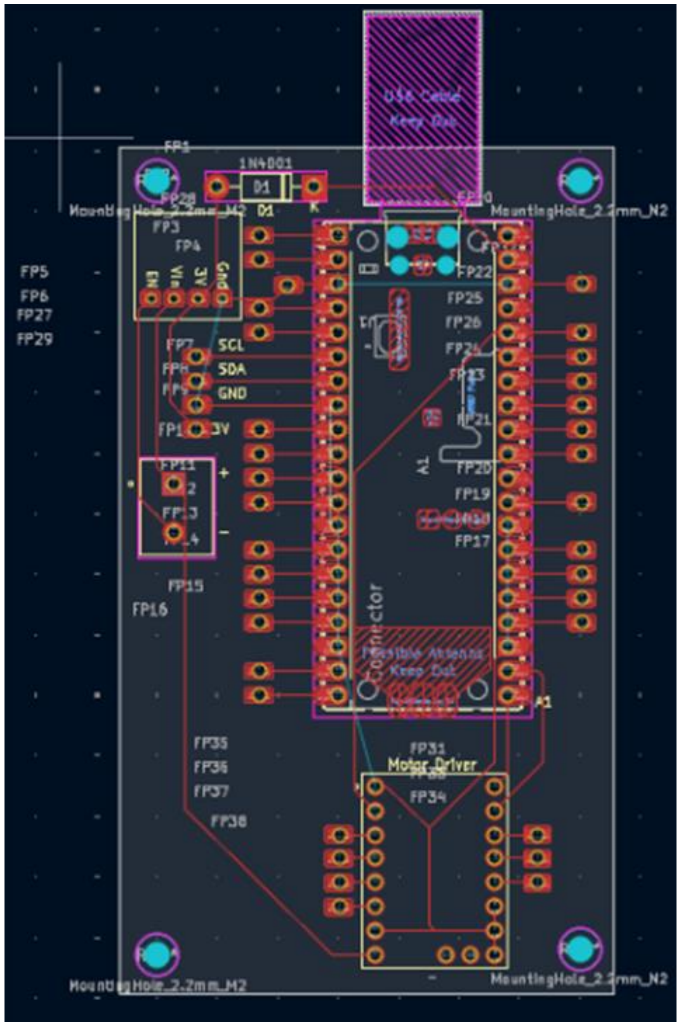

Figure 9: PCB Layout Component Footprint Map with FP Reference Labels

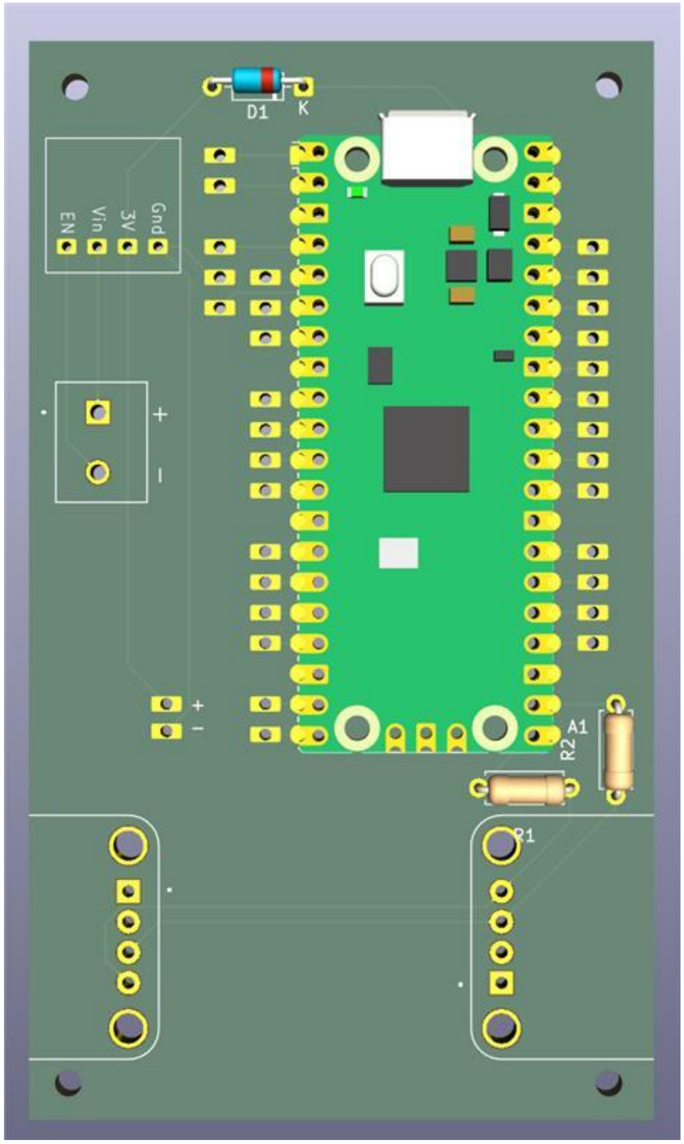

Figure 10: PCB 3D Render Populated Board with RP2040

Note: Circuit schematic shown above is a general layout reference only. For full-resolution detail, refer to the project GitHub repository or website.

Power Flow:

The **LiFePO4 battery array** (4 cells in series, approximately 13V total) connects via a 2-pin connector to the **Battery Management System (BMS)**. The BMS output feeds two primary paths: the **DRV8871 motor driver** power pins receive the full 12V rail to drive the four **GM3865-520 DC gear motors**, and a secondary path routes voltage to the **3.3V buck converter**. This buck converter is a critical component for Robot 1, as it is designed to **suppress regenerative back-EMF voltage spikes up to 24V** during hard stabilization stops, preventing them from resetting the **Raspberry Pi Pico (RP2040)**.

The buck converter's 3.3V output feeds the shared rail on the protoboard, which provides logic power to the **AS5600 Magnetic Encoder**, the **ST7796S touchscreen**, and the **DRV8871 logic supply**. To ensure system stability, the **LED backlight** and **VCC** for the touchscreen are tied directly to this external regulated 3.3V rail rather than the Pico's internal regulator to avoid overdrawing current. Common ground

(GND) is shared across the buck converter, BMS, and Pico to ensure signal integrity for the PID control loop.

DRV8871 Motor Drivers (H-Bridge → GM3865-520 DC Motors)

1. **GP6 (Pin 9) IN1**: PWM Forward control signal from Pico.
2. **GP7 (Pin 10) IN2**: PWM Reverse control signal from Pico.
3. **VM (Voltage Rail) 12V+**: Direct power from the 4S LiFePO4 battery array.
4. **GND Pico GND**: Common ground for signal integrity.
5. **OUT1 / OUT2 M+ / M-**: High-power output to the wheel motors.
    - *Logic Check:* (IN1=1, IN2=0) Forward; (IN1=0, IN2=1) Reverse; (IN1=1, IN2=1) Brake.

See below table for details

Driver Pin	**Connection to Pi Pico**	**Logic Function**
IN1	**GP6** (Pin 9)	PWM Signal (Forward)
IN2	**GP7** (Pin 10)	PWM Signal (Reverse)

AS5600 Magnetic Encoder (Pendulum Angle)

1. **GP5 (Pin 7) SCL**: I2C clock signal for real-time angle polling.
2. **GP4 (Pin 6) SDA**: I2C data signal providing position data.
3. **VCC 3.3V (Pico)**: Logic power.
4. **GND GND (Pico)**: Common ground.

See below table for details

Pin	**Connection to Pi Pico**	**Wire Color**
VCC	3.3V	Red
GND	GND	Black
DIR	HIGH or LOW (currently Low)	Black
SCL	**GP5** (Pin 7)	Blue
SDA	**GP4** (Pin 6)	White

GM3865-520 Motor Encoders (Cart Odometry)

1. **GP2 (Pin 4) Phase A**: Pulse signal from the motor encoder for speed tracking.
2. **GP3 (Pin 5) Phase B**: Pulse signal from the motor encoder for direction tracking.
3. **VCC / GND 3.3V / GND (Pico)**: Powers the Hall-effect sensors inside the motors.

See below table for details

Motor Pin	**Connection to Pi Pico**	**Wire Color**
VCC	3.3V	Yellow
GND	GND	Green

Phase A	**GP2** (Pin 4)	Red
Phase B	**GP3** (Pin 5)	Black

System Safety Note

The **3.3V Buck Converter** is wired in parallel with the **DRV8871 VM rail** to step down the 13V battery voltage. This provides a stable logic rail that suppresses back-EMF spikes up to **24V** generated during hard 10-inch geofence stops, ensuring the RP2040 does not crash during high-kinetic demonstrations.

Touchscreen (ST7796S display with capacitive touch controller):

All GP numbers and signal names are printed on the touchscreen board and the Pico. Match each label to connect correctly.

Display Pin	Pico Pin	Function	Wire Color
VCC	External 3.3V	Power for display	Red
GND	Common GND	Ground	Black
LCD_CS	GP21	TFT chip select	Orange
LCD_RST	GP20	TFT reset	Yellow
LCD_RS	GP22	TFT data/command	Green
SDI (MOSI)	GP19	SPI data input	Blue
SCK	GP18	SPI clock	Purple
SDO (MISO)	GP0	SPI data output	Gray
CTP_SCL	GP3	I2C1 clock (touch)	Orange
CTP_RST	GP6	Touch reset	Yellow
CTP_SDA	GP2	I2C1 data (touch)	Green
CTP_INT	GP7	Touch interrupt	Blue
SD_CS	—	SD card chip select — not connected	—
LED	External 3.3V	Backlight power — see WARNING below	White

WARNING: The LED backlight pin must connect to external 3.3V power — do NOT connect it to a Pico GPIO data pin. Connecting LED to a data pin will overdraw the Pico and may damage it.

Firmware Installation (fresh Pico):

If installing firmware on a new Raspberry Pi Pico for the first time, follow these steps:

1. Hold the BOOTSEL button on the Pico while plugging it into a computer via USB. Release the button once connected. The Pico will appear as a mass storage device on your computer.
2. Drag and drop the compiled firmware.uf2 file (found in the build/ folder of the project GitHub repository) directly onto the Pico drive. The Pico will automatically reboot and run the firmware.
3. Alternatively, use picotool: picotool load -f -x flash.uf2
4. Or use the upload script included in the repository: scripts/upload.sh

Note: If you have modified src/configuration.hpp, a simple make command is not sufficient to rebuild. Run the full build sequence or use scripts/buildFresh.sh to ensure changes take effect before flashing.

Robot 1: System Startup Sequence

On power-on, the software runs the following sequence automatically:

1. **Initial boot** — The Raspberry Pi Pico (RP2040) initializes, booting the firmware to handle high-speed control math.
2. **Pins and sensor setup** — All GPIO pins are configured for PWM motor control; the **AS5600 Magnetic Encoder** is initialized via the I2C bus.
3. **Zero-point calibration** — The system polls the frictionless magnetic sensor to establish the digital vertical reference (0°).
4. **Main loop begins** — The control loop repeats continuously at high frequency:
    - **Read Magnetic Encoder** — Fetch the real-time angular position of the pendulum arm.
    - **Read Wheel Encoders** — Monitor cart odometry to enforce the **10.0-inch geofence kill-switch**.
    - **Write a frame to the touchscreen** — Update the GUI with live sensor data and system status.
    - **Run PID calculation** — Compute the required base acceleration based on K_p, K_i, K_d, and coefficients to maintain verticality.
    - **Adjust PWM Output** — Generate digital drive signals for the **DRV8871 drivers** to accelerate the four DC gear motors.
    - **Loop restarts from step 4.**

Note: If the cart travels exactly **10.0 inches** from its starting point, the odometry logic will automatically trigger an **Emergency Stop** to prevent the robot from driving off a tabletop.

PCB Footprint Reference (key FP labels):

FP Label	Component / Function
A1	Raspberry Pi Pico (main microcontroller) — Handles PID calculations and the touchscreen GUI.
D1 / FP28	Schottky diode — Power protection at VIN to prevent USB backfeeding [23, Previous Chat].
MD1 / MD2	DRV8871 H-Bridge motor drivers — Provides high-current PWM control for the four DC motors.
FP3, FP4	3.3V Buck converter header — Essential for suppressing 24V back-EMF spikes from the DC motors.
FP35–FP38	DC motor output headers (M1–M4) — Connections for the Yahboom 520 motor power leads.

FP19, FP20	ST7796S Touchscreen SPI Pins — Dedicated to the SCK and SDI (MOSI) signals for the interactive interface [75, Previous Chat].
FP1, FP30	M2 PCB mounting holes — Secure the populated board to the internal electronics shelf.

3.2.5 Final Fit & Function Checks

After assembly is completed, perform the following series of checks before powering on for the first time

1. **Pendulum Rotation** – Bring pendulum to resting position on one side of robot, then rotate pendulum to resting position on other side, ensuring motion is smooth and there are no obstructions.
2. **Magnetic Potentiometer Alignment –** Unscrew TPU shell and make sure magnetic potentiometer mount is parallel to magnet affixed to Pendulum Center Bracket, ensure that both parts are not in direct contact. Reattach shell when completed
3. **Motor Check** – Hold Robot frame up and ensure all motor fixing brackets, motors, and wheels are securely screwed in, rotate each wheel one full rotation in each direction to ensure motors are capable of full range of motion.
4. **Top Frame and Touchscreen** – Ensure Frame top is properly secured to bottom frame, and Touchscreen is secured to side of Frame Base
5. **Wiring Check** – Inspect all wiring to ensure all systems are properly connected to each other, no stripped wires in contact with each-other or battery, not wires are snagged on surfaces and subject to tension.

3.3 Disassembly: Robot 1 (Inverted Pendulum)

3.3.1 Preparation

1. **Power off** the robot by flipping the rocker switch to the **OFF** position.
2. Carefully **catch the pendulum arm** and allow it to lower to its vertical rest position [User Update].
3. **Disconnect the LiFePO4 charger** if it is currently connected.

3.3.2 Mechanical Disassembly Steps

1. **Remove TPU Covers:** Carefully detach the **TPU "crumple zone" covers** from the chassis corners [2, User Update].
2. **Remove Pendulum Arms:** Loosen the M3 clamping screws on the **3D-printed center bracket**. Slide the two **6mm aluminum rods** (the arms) out of the assembly. **Do not** disassemble the magnetic mount or the 4-link pivot unless deep maintenance is required.
3. **Remove Wheels:** Loosen the hub set screws on the **67.5 mm wheels** and slide them off the **Yahboom 520 encoder motor** D-shafts [59, User Update].
4. **Remove Acrylic Window:** Slide the **acrylic viewing window** upward out of its dedicated frame slot.
5. **Separate Chassis Halves:** Unscrew and remove the **M3 socket screws** connecting the Top and Bottom chassis halves. Carefully lift the top half, ensuring no internal wires (such as the encoder or switch leads) are snagged.

3.3.3 Electrical Disassembly Steps

1. **Disconnect Encoder:** Unplug the **AS5600 Magnetic Encoder** I2C cable from the Raspberry Pi.
2. **Remove Electronics:** Disconnect the motor leads from the **DRV8871 drivers** and slide the **Raspberry Pi** and internal electronics shelf out of their mounting rails.
3. **Battery Disconnection:** Disconnect the battery leads from the **Battery Management System (BMS)** and the **3.3V buck converter**.

4. **Remove Battery Array:** Slide the **4S LiFePO4 battery pack** out of the internal bottom compartment.

3.3.4 Components Storage and Labeling

1. **Fastener Management:** Place all **M3 and M2 screws** into a labeled zip-lock bag immediately. **Do not mix screw lengths** to prevent stripping the 3D-printed PLA threads during reassembly [32, User Update].
2. **Panel Protection:** Store the **acrylic viewing window** and **TPU covers** flat in a padded sleeve to prevent warping or scratching [User Update].
3. **Sensor Care:** Wrap the **AS5600 Magnetic Encoder** and its associated **magnet** in a soft cloth to protect the frictionless sensor face from debris [User Update].
4. **Battery Health:** Store the **LiFePO4 battery array** at approximately full charge. **Do not store below 3.0V per cell** to maintain safety and longevity (ER4) [7, User Update].
5. **Wiring Integrity:** Label all cable connectors (specifically the **RP2040 GPIO pins**) with masking tape and a marker before disconnecting for long-term storage [User Update].

3.4 Storage: Robot 1 (Inverted Pendulum)

Based on the technical specifications and mechanical design of the Inverted Pendulum system, here is the storage and maintenance procedure for **Robot 1**.

3.4.1 Short-Term Storage

For storage between uses or outreach events:

1. **Power off** the robot by flipping the main rocker switch to the **OFF position**.
2. Carefully **catch the pendulum arm** and allow it to lower to its natural rest position (hanging vertically) before setting the robot down to prevent high-velocity impacts with the chassis.
3. **Disconnect the LiFePO4 charger** if it is currently connected.
4. Place the robot on a **flat, stable surface**. Ensure the wheels are clear of any debris that could degrade the rubber grip.
5. **Do not stack anything** on top of the chassis or the aluminum pendulum arm assembly to avoid deforming the 3D-printed components.

3.4.2 Long-Term Storage

For storage of one week or longer, or before extended transport:

1. **Follow all Short-Term Storage steps** listed above.
2. **Charge the LiFePO4 battery array** to full before storage. Batteries should not be stored below **3.0V per cell** to maintain health and ensure compliance with electrical safety standards (ER4).
3. **If partially disassembling for transport:**
    - Remove the **acrylic viewing window** by sliding it upward through the dedicated frame slot.
    - Remove the 2 **pendulum arms** attached to the elbow brackets on the outside.
4. Place all **M3 and M2 screws** (from the chassis halves or arm assembly) into a **labeled zip-lock bag** immediately after removal. Do not mix screw lengths to avoid stripping the 3D-printed threads during reassembly.
5. **Protect the AS5600 Magnetic Encoder:** Wrap the center bracket mount in a soft cloth or foam to protect the magnet and sensor face from debris or misalignment.
6. **Label all internal cable connectors** (from the motor drivers and encoder to the Raspberry Pi) with masking tape and a marker before disconnecting anything to ensure correct re-wiring.

7. **Store the robot in a padded transport case.** Keep in a cool, dry location away from direct sunlight to protect the chassis, which can begin to warp if temperatures approach the 55°C glass transition limit.

3.5 Operation

3.5.1 System Startup (Inverted Pendulum)

1. **Clear the operating space** — Ensure the robot is placed in the center of a level surface with at least 12 inches of unobstructed space on either side to allow for linear balancing corrections and the **10.0-inch geofence kill-switch**.
2. **Ensure the pendulum arm is at rest** — The aluminum pendulum arm should be hanging freely in the downward position before powering on.
3. **Flip the main rocker switch to the ON position** — This will initialize the **Raspberry Pi controller** and the **4-inch LCD touchscreen**.
4. **Automatic Sensor Calibration** — The system will poll the **AS5600 Magnetic Encoder** to establish the digital zero-point and verify communication with the **four encoder motors**. This initialization takes approximately 3 seconds. **WARNING:** Do not move the cart or the pendulum arm during this initialization phase to ensure an accurate vertical reference.
5. **Engage Balancing** — Once the touchscreen displays the live sensor readout (showing the current angle and cart position), manually **raise the pendulum arm to a vertical position (0°)**. The PID loop will automatically activate and "catch" the arm as it enters the stable control region.
6. **Active Stability** — The robot will begin to drive the wheels to maintain the pendulum within a **±2.5° steady-state error margin**.

Note: If the cart travels more than **10.0 inches** from its starting point or the pendulum arm falls past a recoverable angle, the **geofence logic** will automatically cut motor power to prevent the robot from driving off a table. To reset, manually return the cart to the center of the workspace and raise the arm back to the vertical setpoint to re-engage the PID loop, or turn it off and back on.

3.5.2 Normal Operations

1. Once balancing is active, the **PID loop will continuously accelerate the four-wheel cart base** to keep the pendulum arm at the configured vertical setpoint.
2. The **default setpoint is perfectly vertical (0°)**. Following a physical perturbation, the system is designed to settle back to this position in **less than 5.0 seconds** under normal conditions.
3. **Adjust the setpoint** — Use the integrated touchscreen to change the data being shown (i.e. velocity data, distance from setpoint, etc.). This is a key interactive feature for K–12 students to visualize real-time feedback.
4. **Adjusting PID gains** — Advanced users can adjust $K_p, K_i, K_d,$ and coefficients via the touchscreen interface. **WARNING:** Do not set to extreme values without testing. Excessive integral gain, combined with motor back-EMF spikes (up to 24V), can cause violent cart acceleration and potential hardware resets.
5. **Safe operating boundaries** — The software enforces a **10.0-inch geofence kill-switch**. If the chassis travels more than 10 inches from its starting point, the odometry logic will reliably cut motor power to prevent the robot from driving off a tabletop.
6. **Student interaction** — Students are encouraged to introduce **physical perturbations (up to a 10° push at small forces)** to the pendulum arm during operation. Software-based **moving average filters** and anti-windup logic allow the system to recover cleanly and maintain verticality within a **±2.5° steady-state error margin**.

3.5.3 Shutdown Procedure: Robot 1 (Inverted Pendulum)

1. Navigate to the touchscreen GUI and select **"Stop"** to disengage the active PID balancing loop. Similarly, use the rocker switch to cut power and effectively "Stop" the robot (see step. 3).
2. Carefully **catch the pendulum arm** and allow it to lower to its rest position. Do not allow the arm to impact the chassis or the table at high velocity while the motors are active.
3. Flip the **rocker switch to the OFF position**. This will immediately de-energize the **Yahboom 520 encoder motors** and the Raspberry Pi controller.
4. If the robot is not used for more than 48 hours, connect the **LiFePO4 battery charger**. Testing verified a **1.4-hour continuous runtime**, but the cells must be regularly balanced via the BMS to ensure electrical safety (ER4) and battery longevity.
5. **For transport:** Ensure the **4-piece aluminum pendulum arm** assembly is secure. Place the robot in a padded transport case, taking care to protect the **AS5600 Magnetic Encoder** mount and the acrylic viewing window. Do not stack heavy objects on top of the chassis.

3.6 Troubleshooting

3.6.1 Startup Issues

Problem	Likely Cause	Fix
Robot does not power on	Battery discharged or rocker switch connection loose	Charge battery fully. Check rocker switch is fully seated in its cutout and wired correctly.
Touchscreen blank on power-on	RP2040 not booting or loose touchscreen cable	Check touchscreen cable connection to GPIO06–GPIO15 on RP2040. Re-flash firmware.uf2 via BOOTSEL method if screen remains blank after reseating cable.
Magnetic Encoder is not reading values	Magnet misaligned in the 3D-printed center bracket	Re-align the magnet so it is centered directly over the AS5600 sensor; verify continuity of I2C signal wires.
Wheels are not turning	Motor driver (DRV8871) thermal shutdown, loose motor wiring, or the E-Stop has been triggered.	Check if the E-Stop is engaged. Inspect wiring to DRV8871 drivers and ensure motor casing temperature is < 50°C.
Pendulum not balancing	Untuned PID gains, high raw sensor noise (jitter), or excessive friction in pendulum bearings.	Fine-tune Kp, Ki, and Kd coefficients in the control software. Verify that moving average filters are active to smooth sensor data. Inspect and lubricate bearings to ensure frictionless rotation.

3.6.2 Operational Issues

Problem	Likely Cause	Fix
Ball does not settle — oscillates continuously	Kd too low, allowing overshoot to compound	Increase Kd on touchscreen in increments of 0.1 until oscillation dampens. Default Kd = 0.05.

Problem	Likely Cause	Fix
Ball settles very slowly	Kp too low	Increase Kp in increments of 0.1. Default Kp = 0.75. If above 2.0, check ToF sensor calibration per Section 4.2.5.
Ball launched off beam on release after being held	Integral windup accumulated while ball was held manually	Verify anti-windup clamp is active in firmware. Temporarily reduce Ki. PID values reset on reboot — power cycle to restore defaults (Kp=0.75, Ki=0.30, Kd=0.05).
Robot is violently unstable after PID values were adjusted	Gains set to extreme values on touchscreen	Turn the rocker switch OFF, wait 5 seconds, and turn back ON. PID values reset to factory defaults on every boot. To view or restore defaults, open src/configuration.hpp in the project GitHub and locate BEAM_KP, BEAM_KI, and BEAM_KD.
ToF sensor reading jumps or drops out	Sensor misaligned or direct sunlight interference	Realign sensor so lens faces exactly down beam axis. Check SCL/SDA connections at GPIO02/GPIO03 on RP2040. Reduce direct sunlight if possible.
Ball rolls off end of beam	Setpoint beyond beam length or ball placed while PID is saturated	Place ball at center before engaging PID. Verify setpoint is within 20–270mm from sensor.
PID response sluggish after transport	ToF sensor offset shifted from vibration	Re-run sensor calibration per Section 4.2.5 step 3.

3.6.3 Mechanical Issues

Problem	Likely Cause	Fix
Acrylic window fell out	Slit tolerance too loose from print warping	Reprint Motor Holder in correct 90° orientation (Section 4.2.1). Temporary fix: small strip of clear tape inside the slit.
Pendulum grinds or catches during swing	6010 bearing misaligned or debris in the frictionless pivot.	Power off, remove TPU cover, inspect the pendulum pivot. Re-seat the 6010 bearings and ensure the 6mm aluminum rods are not rubbing against the chassis walls.
Pendulum arm has side-to-side wobble	Aluminum rods loose in the 3D-printed center bracket or bearing seats.	Ensure the 4-piece pendulum arm is fully seated in the center bracket. Re-tighten the M3 clamping screws; if the rod diameter is slightly under 6mm, apply a small shim of tape to the rod end.
Motors hum but wheels do not turn	Set screw slipping on the Yahboom 520 motor D-shaft or hub.	Power off. Access the internal motor level. Re-tighten the wheel hub set screws, ensuring they are locked against the flat side of the motor shaft.
Cracking sound during drop or hard stop	Internal layer delamination in the PLA+ chassis or fractured motor brackets.	Inspect the chassis for stress fractures, if structural integrity is compromised, reprint the damaged frame using high-fidelity PLA.

Problem	Likely Cause	Fix
Chassis feels loose or flexes	M3 socket screws connecting the top and bottom frame halves have backed out due to vibration.	Re-tighten all M3 socket screws. Consider using blue thread-locker or locking nuts to prevent future loosening during high-frequency balancing corrections.
Wheels detaching during operation	Loose hub fit or excessive vibration on the drive axle.	Inspect the hub-to-axle connection. Re-tighten lock nuts and ensure the 67.5 mm wheels are secure.

4. Robot 2 — Ball-on-Beam System

Robot 2 is a ball-on-beam balancing system where a ping-pong ball sits inside a 3D-printed U-channel beam that can tilt left or right. The NEMA 17 stepper motor is mounted at the center of the chassis and the beam tilts in real time based on commands from a PID control loop running on a Raspberry Pi Pico (RP2040) microcontroller. The VL53L0X Time-of-Flight laser sensor is mounted at one end of the beam, and it tracks the ball's position to 1.0mm resolution. The robot is powered by a 14.8V LiFePO4 battery array and operates fully untethered for approximately 85 minutes on a single charge.

Major Sub-Assemblies:

- Center Hub — main electronics housing; contains the RP2040, electronics shelf, and hard-stop walls at ±20°
- Motor Holder — houses the NEMA 17 stepper motor, battery array, BMS, and charging port
- Beam — U-channel ball track; houses the 686 bearing on one end and the Time-of-Flight (ToF) sensor mount on the other
- Railroad Tie Plates — left and right structural connectors locking the two chassis halves together
- Acrylic Windows — clear side panels providing visibility into electronics while protecting internal components

Required Tools:

- Soldering iron (set to 200°C for heat-set inserts)
- M3 and M2 hex drivers or Allen keys
- Needle-nose pliers
- Wire cutters and strippers
- 3D printer (PLA-capable) with slicer software
- Ruler or calipers

WARNING: LiFePO4 batteries must never be punctured, shortened, or submerged. Keep fingers and loose clothing away from the beam and motor area when the robot is powered on. Do not operate on the edge of a table.

4.1 Maintenance

4.1.1 Routine Maintenance

Perform the following checks before each use or outreach event:

1. **Fastener inspection** — Check all M3 and M2 screws on the touchscreen mount, motor face, railroad tie plates, and ToF sensor mount. Tighten any that feels loose. Do not overtighten — PLA threads can strip. If a screw no longer holds, re-press the heat-set insert using the soldering iron at 200°C.
2. **Beam surface inspection** — Run a fingertip along the inside of the U-channel beam trough. The surface should be smooth and free of scratches, warping, or debris. Sand lightly with 400-grit sandpaper if needed or reprint the beam (approximately 2–3 hours).

3. **Acrylic window check** — Inspect the acrylic side panels for cracks or chips. Wipe clean with a dry microfiber cloth. Do not use solvent-based cleaners as they will cloud the acrylic.
4. **Beam pivot check** — Rotate the beam by hand with the robot powered off. It should pivot smoothly with no grinding or wobble. If resistance is felt, check that the Shaft Plug is fully seated in the bearing.
5. **Ball inspection** — The ping-pong ball should be round and undented. A dented ball causes erratic PID behavior. Replace with a standard ITTF-specification ping-pong ball.
6. **D-shaft set screw** — Confirm the NEMA 17 motor's D-shaft is fully engaged in the beam's printed sleeve with the flat side facing up. If the beam slips without the motor moving, re-seat the shaft and tighten the set screw.

4.1.2 Electrical Maintenance

The following covers electrical maintenance checks.

7. **Cable routing inspection** — Check that the ToF sensor wires routed through the beam slit are not pinched or under tension when the beam tilts to ±20°. Allow the wire must have enough slack to follow the full range of motion.
8. **Charging port inspection** — Confirm the charging cable is secure, and the port is not loose in its mounting hole. The charging cable connects directly to the P+ and P- terminals on the BMS.
9. **Battery voltage check** — Before each use, verify that the battery is adequately charged. Do not operate the robot if battery voltage has dropped below 12.0V under load. Do not allow individual cells to drop below 3.0V to 3.2V per cell. The BMS is installed to help prevent over-discharge but monitor regularly.
10. **BMS visual check** — Confirm the BMS board is seated flat in its cutout with no visible burn marks, swelling, or corrosion on the board or terminals.
11. **Firmware check** — Verify the Raspberry Pi Pico is running the latest firmware. The source code and default PID configuration are available in the project GitHub repository at src/configuration.hpp as BEAM_KP, BEAM_KI, and BEAM_KD. Current factory defaults: Kp = 0.75, Ki = 0.30, Kd = 0.05.

4.2 Assembly

4.2.1 3D Printing Specifications

All components for Robot 2 are printed using standard PLA filament with 20% gyroid infill. Enable Auto-Tree Supports in your slicer for all parts unless otherwise specified below.

Note: All slicer screenshots below include an XYZ axis indicator that represents the bottom left corner of the print bed (red = X, green = Y, blue = Z). Use this as a visual reference to confirm each part is oriented correctly before slicing.

Center Hub:

Print with the front face (the side with the touchscreen opening) flat down on the print bed.

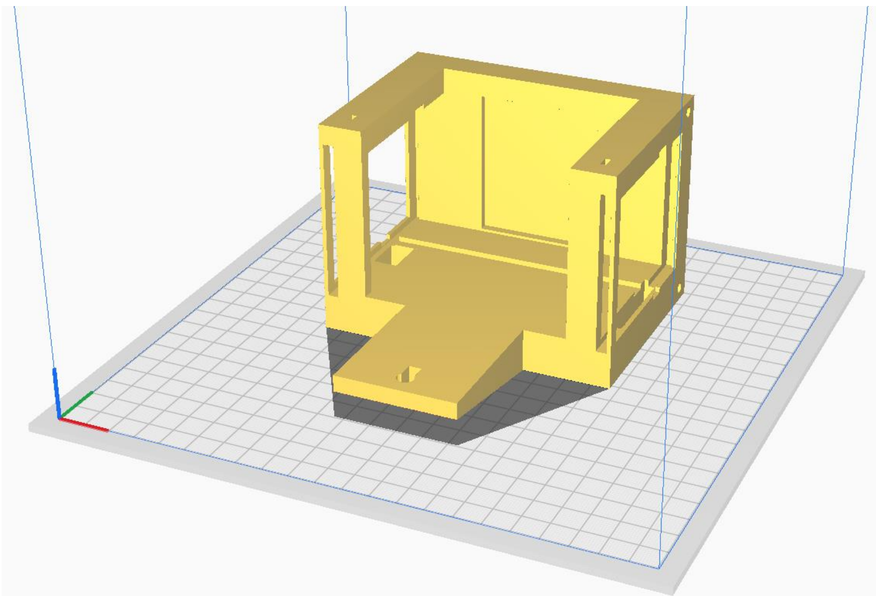

Figure 11: 3D Print Orientation: Center Hub (Front Face Down)

Motor Holder (CRITICAL ORIENTATION):

Print with the back wall facing flat on the bed. Ensure the top motor cutout is oriented exactly 90 degrees to the left or right of the bed. This ensures the nozzle moves lengthwise along the acrylic window slits, preventing wobbling and visual imperfections at higher Z-heights.

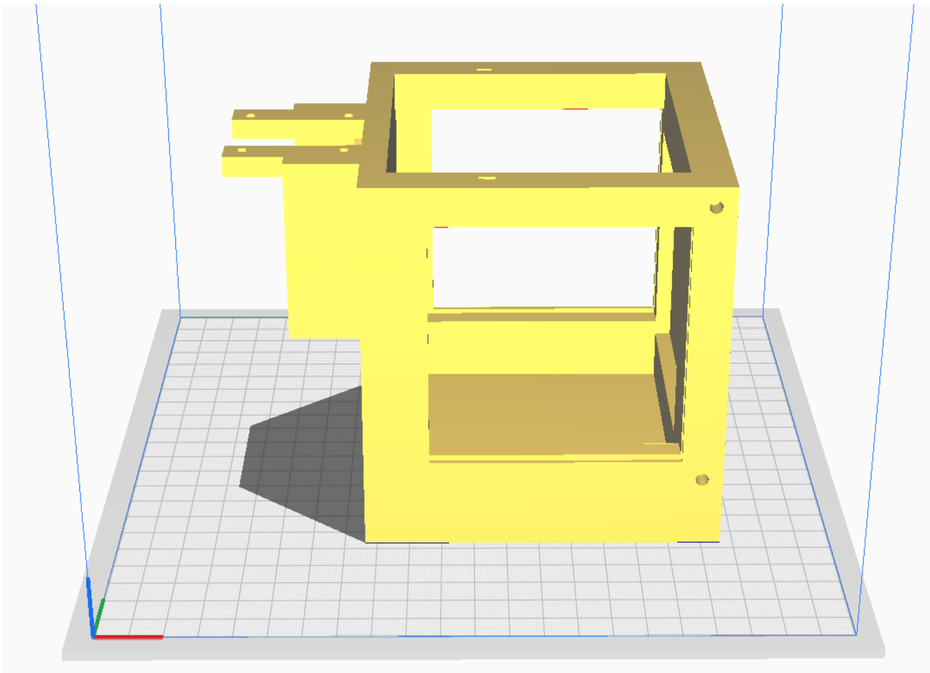

Figure 12: 3D Print Orientation: Motor Holder (90° Rotation, Back Wall Down)

Beam:

Print with the flat bottom directly on the build plate, rising upward. Auto-tree supports are required to ensure the internal motor shaft sleeve holds the correct dimension.

Shaft Plug:

Print as a standard cylinder rising upward from its flat circular base.

Railroad Tie Plates:

Lay flat on their sides. The circular screw holes must be drawn directly onto the bed to ensure accurate circular geometry.

Electronics Shelf:

Print normally with no special orientation required.

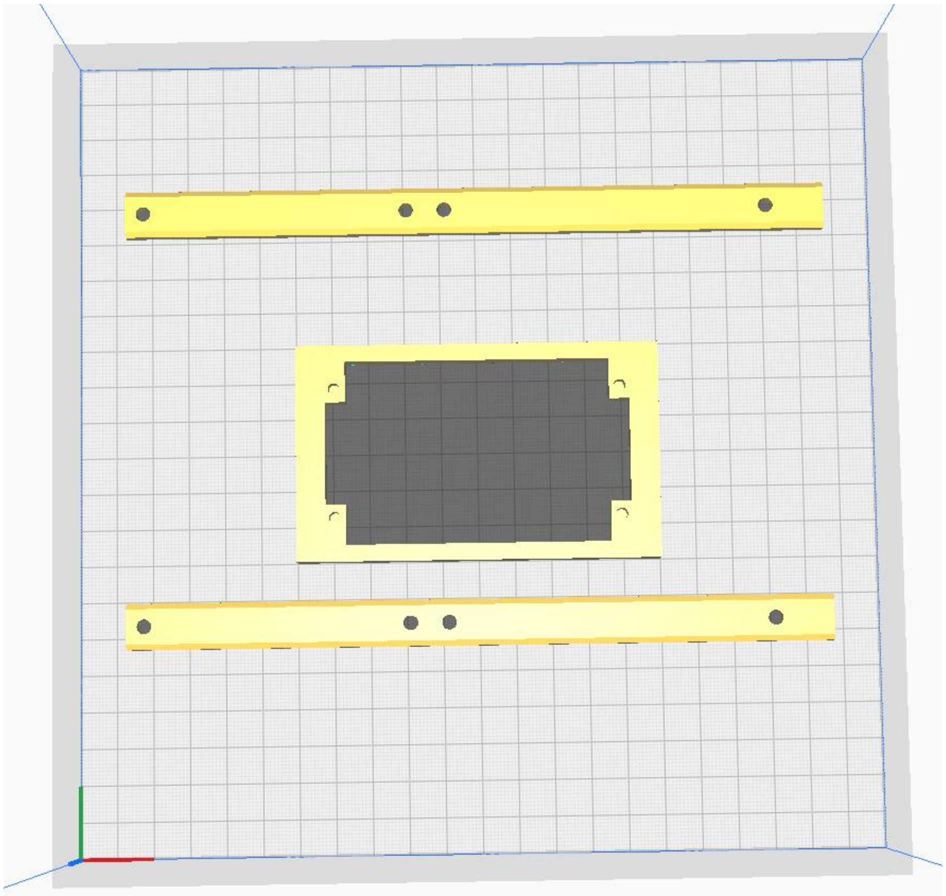

Figure 13: 3D Print Orientation: Electronics Shelf and Railroad Tie Plates (Flat on Bed)

4.2.2 Pre-Assembly Checklist

Before beginning assembly, install all brass threaded inserts into the printed parts. Use a soldering iron set to 200°C. Let gravity assist in the insert and allow 1 full minute to cool before applying any mechanical load.

1. M3 × 5.7mm Inserts: Install into the left and right-side connection holes of the Center Hub and Motor Holder, and into the front face holes of the Center Hub (for the touchscreen).
2. M3 × 4.0mm Inserts: Install two into the side of the beam at the positions spaced for the time-of-flight sensor.
3. M2 Inserts: Insert into the mounting holes on the electronics shelf.
4. 686 Bearing: Place the bearing flat on a sturdy table and position the designated hole of the Beam over the bearing. Firmly press down wi the body weight to friction-fit the bearing into the plastic pocket.

Once inserts are installed, confirm that all the following parts and hardware are present and inspected:

Printed Parts:

- Center Hub — printed, supports removed, heat-set inserts installed

- Motor Holder — printed in correct 90° orientation, supports removed, heat-set inserts installed
- Beam — printed, supports removed, 686 bearing pressed in, heat-set inserts installed for ToF sensor
- Shaft Plug — printed
- Railroad Tie Plates (×2) — printed flat, screw holes clean
- Electronics Shelf — printed

Hardware:

- NEMA 17 stepper motor
- VL53L0X ToF sensor with cable
- 686 ball bearing
- M3 × 8mm screws (×8 minimum)
- M3 × 12mm screws (×4 for railroad ties)
- M3 × 5.7mm brass heat-set inserts
- M3 × 4.0mm brass heat-set inserts
- M2 brass heat-set inserts
- M3 nuts (×2, for ToF sensor spacers)
- LiFePO4 battery cells (×4)
- Acrylic window panels (cut to dimension per CAD package)
- Ping-pong ball (ITTF standard)
- Raspberry Pi Pico (RP2040)
- TMC2209 stepper motor driver
- BMS (Battery Management System)
- Buck converter (step-down to 3.3V)
- Rocker switch
- 4" touchscreen (ST7796S display driver)
- Wiring harness / hookup wire

4.2.3 Mechanical Assembly

Begin with the chassis halves. Press the Center Hub and Motor Holder together.

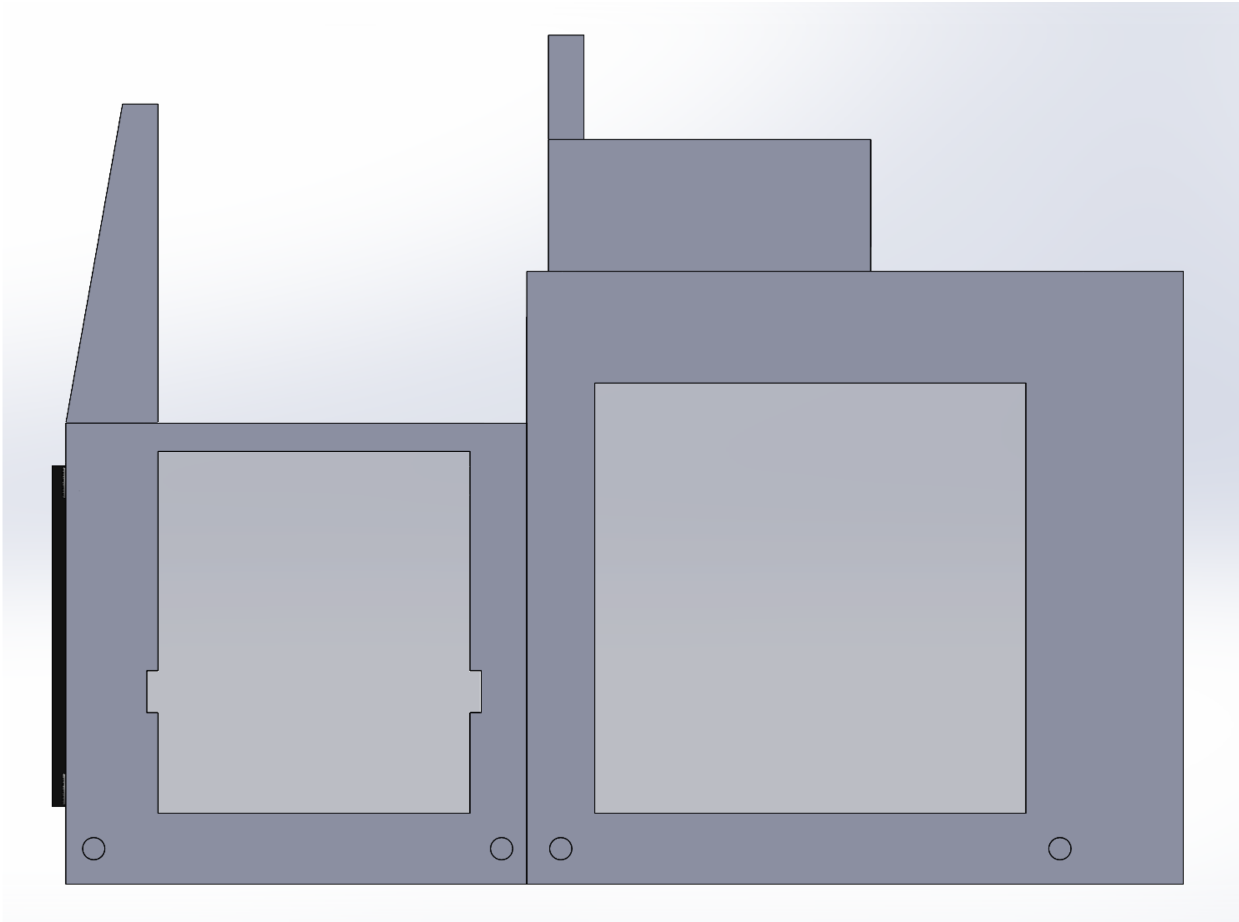

Figure 14: Chassis Assembly Center Hub and Motor Holder Mating View (Pre-Assembly)

Beam and Shaft Plug: Position the Beam directly above the Center Hub. Slide the Shaft Plug through the aligned slit in the Center Hub wall directly into the 686 bearing on the Beam.

1. Stepper Motor: Slide the stepper motor into the cutout on the Motor Holder, routing the motor cables down through the designated base hole. Press the motor's D-shaft directly into the printed sleeve on the Beam. Ensure the flat side of the D-shaft is facing UP.
2. Secure Motor: Using M3 × 8mm screws, bolt the front face of the stepper motor to the Motor Holder. Tighten fully to ensure zero mechanical play.
3. Railroad Ties: Place the printed railroad tie plates on the left and right sides spanning both chassis halves. Secure with M3 × 12mm screws. Tighten fully.
4. ToF Sensor: Place an M3 nut in front of the threaded inserts on the Beam to act as a spacer. Mount the sensor with the black lens facing into the beam channel. Secure with M3 × 8mm screws. Route the sensor wires through the beam slit and down through the bottom circular opening.
5. Electronics Shelf: Secure the PCB or breadboard to the shelf using M2 screws. Slide the shelf through the Center Hub slit so it sits directly beneath the Beam. Connect the routed sensor wires to the board.
6. Power and BMS: Slide the LiFePO4 batteries into the Motor Holder slot. Place the BMS in the cutout under the shelf. Route the charging cable along the left side, behind the batteries, through the hook piece, and out the top hole. Secure the back wall plate with M3 × 8mm screws to lock the charging port in place.

7. Acrylic Windows: Cut acrylic to the dimensions specified in the CAD package. Slide the panels downward into the printed slits on the Motor Holder and Center Hub.

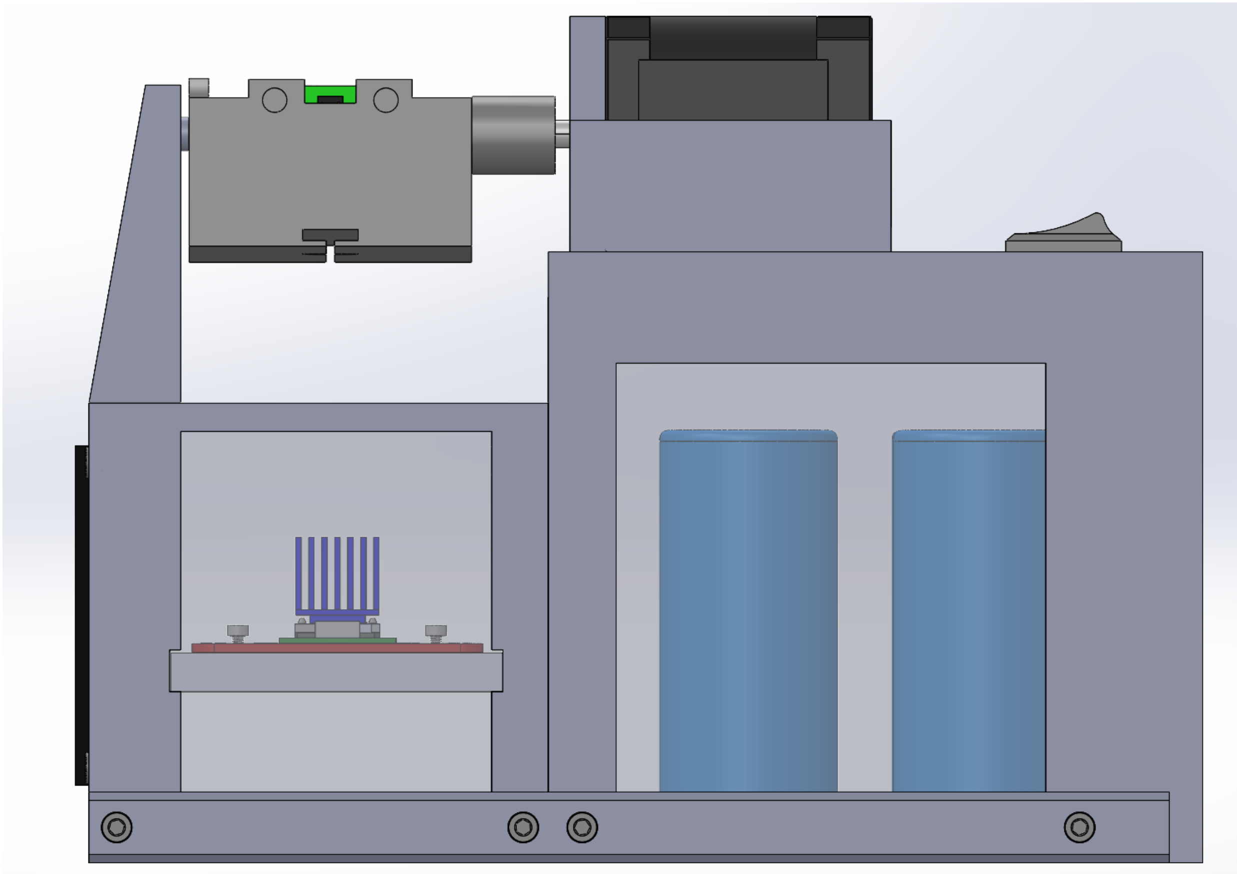

Figure 15: Chassis Assembly Full Side View with Sub-Assemblies Installed

4.2.4 Electrical Integration

This section describes the complete electrical wiring of Robot 2 based on the confirmed pin configuration from the project GitHub repository. All GP numbers and signal names referenced below are printed directly on the physical boards, they match the label visible on each component to the corresponding entry in the tables. GP numbers (e.g. GP6, GP21) are the GPIO numbers used by the firmware. Physical pin numbers (e.g. pin 9, pin 16) are the numbered positions on the Raspberry Pi Pico board counted from the top-left corner; for example, GP6 is located at physical pin 9. Signal names such as SDA, SCL, MOSI, and STEP come from component datasheets and describe the function of each connection.

The circuit schematic (BBBSchematic.pdf) and PCB layout are attached as screenshots below for quick reference, please refer to the full-resolution files for detailed verification.

Circuit Schematic Reference:

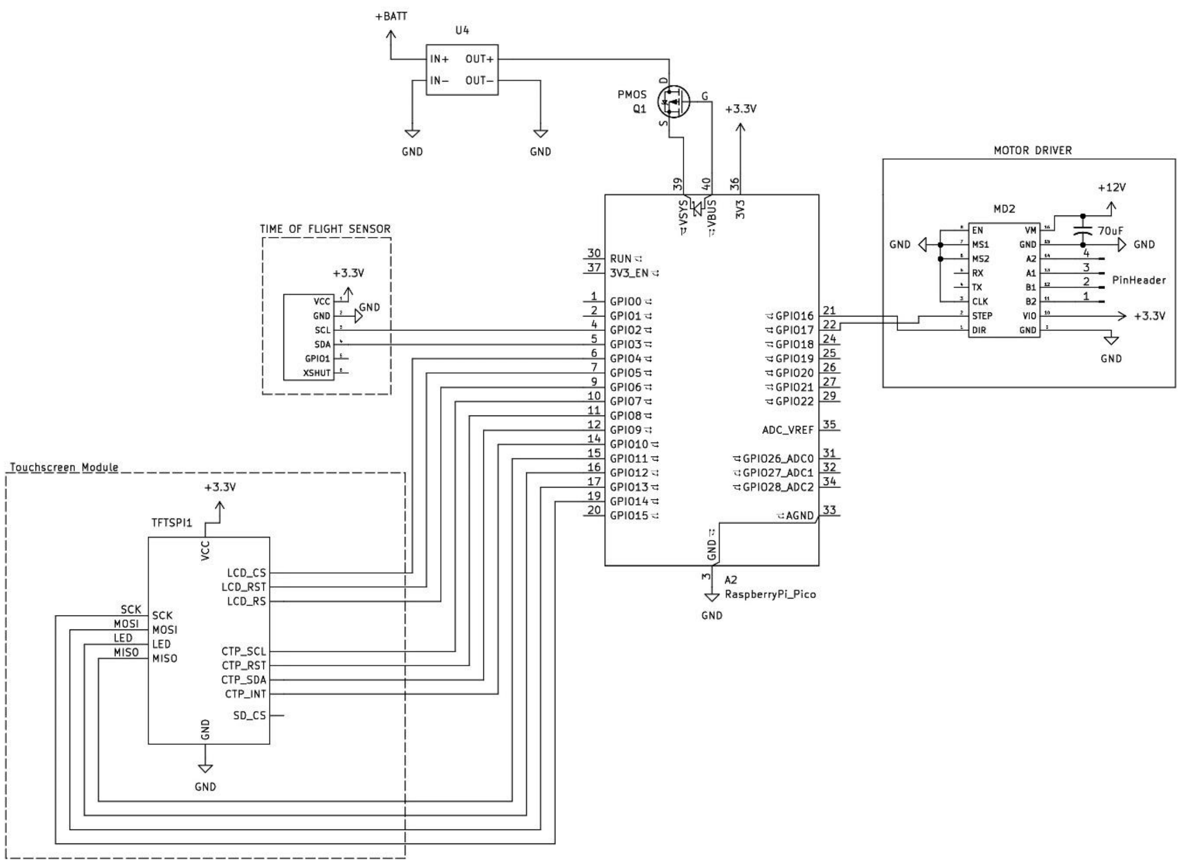

Figure 16: Circuit Schematic for Robot 2 Ball-on-Beam General Layout (BBBSchematic.pdf)

PCB Layout Reference:

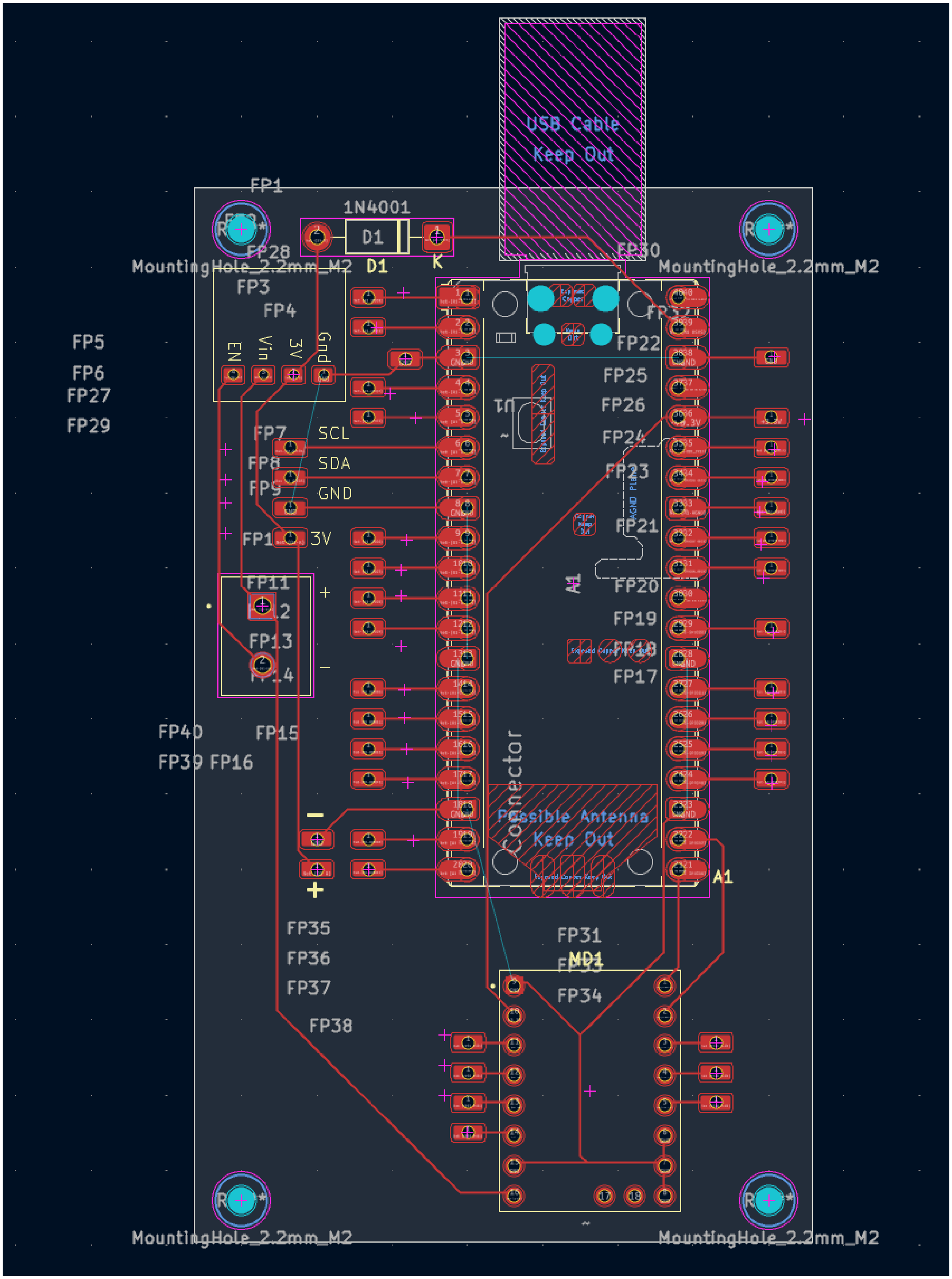

Figure 17: PCB Layout Component Footprint Map with FP Reference Labels

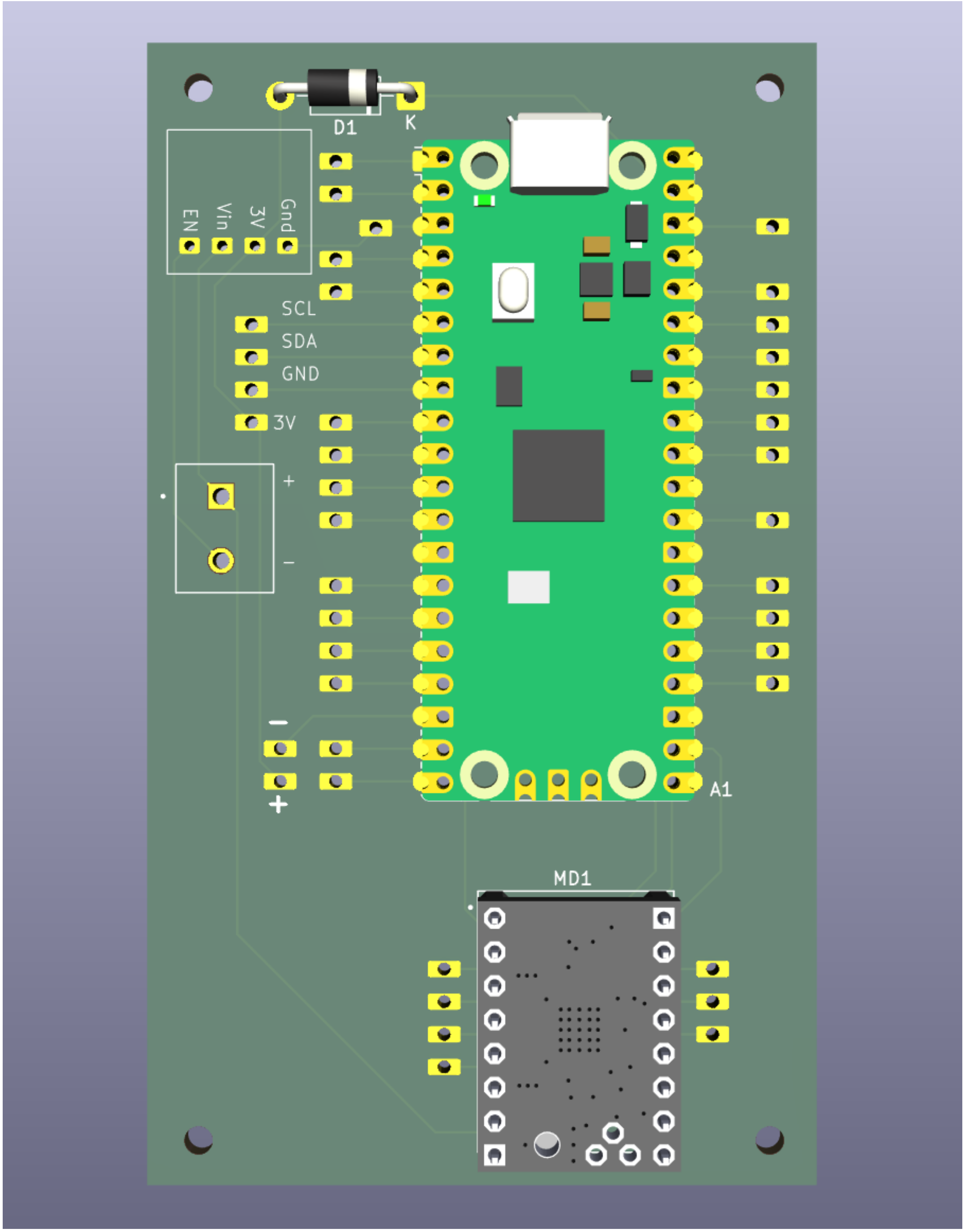

Figure 18: PCB 3D Render Populated Board with RP2040 and TMC2209 Stepper Driver

Note: Circuit schematic shown above is a general layout reference only. For full-resolution detail, refer to the project GitHub repository or website.

Power Flow:

The LiFePO4 battery array (4 cells in series, approximately 13V total) connects via a 2-pin connector to the BMS. The BMS output feeds two paths: the motor driver VM pin receives the full 12V rail for motor power, and a Schottky diode routes voltage to the Pico's VIN pin. The Pico's onboard 3.3V regulator then powers all logic components, the VL53L0X sensor, touchscreen, and TMC2209 logic supply. A step-down buck converter is present on the board as an alternative regulated 3.3V supply path. GND from the buck converter connects to the Pico GND, and the 3.3V output feeds the shared 3.3V rail on the protoboard.

Stepper Motor Driver (TMC2209 → NEMA 17):

1. Pin 16 (Pico) → STEP — sends step pulse signals to TMC2209
2. Pin 17 (Pico) → DIR — sets rotation direction
3. EN → GND (Pico) — active low, driver is always enabled, no GPIO control required
4. MS1 and MS2 → active low — sets 1/8 microstepping mode
5. VM → 12V power rail — motor voltage supply
6. VIO → 3.3V (Pico) — logic voltage
7. Motor coil wires connect to the 4-pin header on the PCB in the following order:

Terminal	Phase	Wire Color
A1	A+	Red
A2	A−	Black
B1	B+	Yellow
B2	B−	Blue

VL53L0X Time-of-Flight Sensor:

1. VCC → 3.3V rail
2. GND → GND rail
3. SDA → GP6 (Pico pin 9) — I2C1 data
4. SCL → GP7 (Pico pin 10) — I2C1 clock
5. XSHUT → leave disconnected or pull high — not actively controlled in this design
6. Interrupt pin → not required for basic ranging, leave disconnected

Touchscreen (ST7796S display with capacitive touch controller):

All GP numbers and signal names are printed on the touchscreen board and the Pico. Match each label to connect correctly.

Display Pin	Pico Pin	Function	Wire Color
VCC	External 3.3V	Power for display	Red
GND	Common GND	Ground	Black
LCD_CS	GP21	TFT chip select	Orange
LCD_RST	GP20	TFT reset	Yellow
LCD_RS	GP22	TFT data/command	Green
SDI (MOSI)	GP19	SPI data input	Blue
SCK	GP18	SPI clock	Purple
SDO (MISO)	GP0	SPI data output	Gray
CTP_SCL	GP3	I2C1 clock (touch)	Orange
CTP_RST	GP6	Touch reset	Yellow
CTP_SDA	GP2	I2C1 data (touch)	Green
CTP_INT	GP7	Touch interrupt	Blue
SD_CS	—	SD card chip select — not connected	—
LED	External 3.3V	Backlight power — see WARNING below	White

WARNING: The LED backlight pin must connect to external 3.3V power — do NOT connect it to a Pico GPIO data pin. Connecting LED to a data pin will overdraw the Pico and may damage it.

Firmware Installation (fresh Pico):

If installing firmware on a new Raspberry Pi Pico for the first time, follow these steps:

7. Hold the BOOTSEL button on the Pico while plugging it into a computer via USB. Release the button once connected. The Pico will appear as a mass storage device on your computer.
8. Drag and drop the compiled firmware.uf2 file (found in the build/ folder of the project GitHub repository) directly onto the Pico drive. The Pico will automatically reboot and run the firmware.
9. Alternatively, use picotool: picotool load -f -x flash.uf2
10. Or use the upload script included in the repository: scripts/upload.sh

Note: If you have modified src/configuration.hpp, a simple make command is not sufficient to rebuild. Run the full build sequence or use scripts/buildFresh.sh to ensure changes take effect before flashing.

System Startup Sequence:

On power-on, the software runs the following sequence automatically:

1. Initial boot — system initializes
2. Pins and sensor setup — all GPIO pins configured, VL53L0X sensor initialized
3. Beam reset to center — beam moves to the zero level position
4. Main loop begins — repeats continuously:
    - Send desired angle to stepper motor
    - Write a frame to the touchscreen
    - Read Time-of-Flight sensor
    - Run PID calculation
    - Adjust output value → loop restarts from step 1

PCB Footprint Reference (key FP labels):

FP Label	Component / Function
A1	Raspberry Pi Pico (main microcontroller)
D1 / FP28	Schottky diode — power protection at VIN
MD1 / FP31, FP34	TMC2209 stepper motor driver module
FP3, FP4	Buck converter header (EN, Vin, 3V, Gnd)
FP35–FP38	Motor coil output headers (A1, A2, B1, B2 → NEMA 17 wires)
FP19, FP20	GPIO18/GPIO19 area — currently unconnected, reserved for future use
FP1, FP30, corner FPs	M2 PCB mounting holes (four corners)

<u>4.2.5 Final Fit and Function Checks</u>

After assembly is complete, perform the following checks before powering on for the first time:

1. **Beam pivot** — Rotate the beam through its full range manually. It should reach both ±20° hard-stop walls smoothly with no grinding or wobble.
2. **Motor shaft engagement** — With the robot powered off, manually rotate the NEMA 17 shaft carefully. The beam should tilt in direct proportion with no slipping. If not, re-seat the D-shaft with the flat side facing up and tighten the set screw.
3. **ToF sensor alignment** — Power on the robot. Slowly move the ping-pong ball from one end of the beam to the other. The distance reading on the touchscreen should change smoothly with no dropouts or jumps greater than 5mm. If jumps occur, loosen the sensor to mount screws, realign, and retighten.
4. **Hard-stop verification** — Manually tilt the beam to both limits. The Center Hub wall should stop the beam cleanly at ±20° without cracking sounds or visible deformation.
5. **Ball tracking test** — Place the ball at the center of the beam and engage the PID loop via the touchscreen. The ball should settle to the setpoint within 15 seconds. If not, see Section 4.6 of Troubleshooting.

4.3 Disassembly

WARNING: To prevent fracturing the 3D-printed PLA chassis, disassembly must follow the sequence below to relieve mechanical tension in the correct order.

4.3.1 Preparation

1. Power off the robot by flipping the rocker switch to the OFF position.
2. Allow the beam to return to level position on its own before proceeding.
3. Remove the ping-pong ball from the beam and set aside.
4. Disconnect the LiFePO4 charger if connected.

4.3.2 Mechanical Disassembly Steps

1. Remove the acrylic windows by sliding them upward out of the printed slits.
2. Unscrew and remove the left and right Railroad Tie plates (M3 × 12mm screws, ×4 total).
3. Loosen the stepper motor mounting screws. Carefully slide the motor's D-shaft out of the Beam's printed sleeve. Support the beam with one hand while pulling the motor back with the other. Do not yank.
4. Pull the Shaft Plug straight out of the Center Hub wall to free the other end of the Beam.
5. Carefully separate the chassis halves (Center Hub and Motor Holder), ensuring no wires routed between them are snagged or snapped.

4.3.3 Electrical Disassembly Steps

1. Disconnect the ToF sensor cable from GP6/GP7 on the RP2040 before removing the electronics shelf.
2. Slide the electronics shelf straight out of the Center Hub slit.
3. Disconnect the battery leads from the BMS (B+ and B- terminals) before removing the battery cells.
4. Slide the LiFePO4 battery cells out of the Motor Holder battery slot.

4.3.4 Components Storage and Labeling

1. Place all M3 and M2 screws into a labeled zip-lock bag immediately after removal. Do not mix screw lengths.
2. Store the acrylic panels flat to prevent warping.
3. Wrap the ToF sensor in a soft cloth or foam to protect the laser lens.
4. Store the LiFePO4 batteries at approximately full charge in a cool, dry location. Do not store below 3.0V per cell.
5. Label all cable connectors with masking tape and a marker before disconnecting if this is a long-term disassembly.

4.4 Storage

4.4.1 Short Term Storage

For storage between uses or outreach events:

1. Power off the robot by flipping the rocker switch to the OFF position.
2. Allow the beam to return to level position on its own before setting the robot down.
3. Disconnect the LiFePO4 charger if connected.
4. Remove the ping-pong ball from the beam and store separately to prevent rolling or loss.
5. Place the robot on a flat, stable surface. Do not stack anything on top of the beam.

4.4.2 Long Term Storage

For storage of one week or longer, or before extended transport:

1. Follow all Short-Term Storage steps above.
2. Charge the LiFePO4 battery to full before storage. LiFePO4 batteries store best at full charge and should not be stored below 3.0V per cell.
3. If partially disassembling for transport: remove the acrylic windows by sliding them upward, then remove the railroad tie plates (M3 × 12mm screws ×4). Store the acrylic panels flat to prevent warping.
4. Place all M3 and M2 screws into a labeled zip-lock bag immediately after removal. Do not mix screw lengths.
5. Wrap the ToF sensor in a soft cloth or foam to protect the laser lens if the beam assembly is removed.
6. Label all cable connectors with masking tape and a marker before disconnecting anything.
7. Store the robot in a padded transport case or bag. Store in a cool, dry location away from direct sunlight.

4.5 Operation

4.5.1 System Startup

1. Ensure the ping-pong ball is NOT on the beam before powering on.
2. Flip the main rocker switch to the ON position.
3. The system will run an automatic homing and calibration sequence. The stepper motor will rotate the beam to one side, then rotate back up to horizontal to establish the zero-level angle. This takes approximately 3 seconds.

WARNING: Do not touch the beam during the auto-home sequence.

4. Once the beam levels and the touchscreen display the live sensor readout, the robot is ready for operation.
5. Place the ping-pong ball on the beam; the robot will begin to actively balance the ball as the PID loop activates and takes over.

Note: If the ball gets stuck at either far end of the beam, or is removed for approximately 2 seconds, the robot will automatically reset and re-run the homing sequence. Place the ball at the center again to re-engage balancing. If the beam does not return to perfect level after a reset, manually move it to flat before placing the ball.

4.5.2 Normal Operations

1. Once balancing is active, the PID loop will continuously tilt the beam to keep the ball at the configured setpoint position.
2. The default setpoint is the center of the beam (approximately 145mm from the sensor). The ball will settle to this position within approximately 9 seconds under normal conditions.
3. **Adjust the setpoint** — Use the touchscreen to change the target position. The ball will respond immediately and move to the new target. This is the primary interactive feature for K-12 demonstrations as students can change the setpoint and watch the ball follow.
4. **Adjusting PID gains** — Advanced users can adjust Kp, Ki, and Kd through the touchscreen. Factory default values: Kp = 0.75, Ki = 0.30, Kd = 0.05. These values reset to defaults on every boot and changes made through the touchscreen are not saved permanently. To permanently change defaults, edit BEAM_KP, BEAM_KI, and BEAM_KD in src/configuration.hpp in the project GitHub repository. Increasing Kp makes response faster but may cause overshooting. Increasing Kd dampens oscillation.

WARNING: Do not set Ki above 2.0 without testing. Excessive integral gain can cause the beam to snap hard when the ball is released after being held manually.

5. **Safe operating boundaries** — The beam is physically limited to ±20° by the hard-stop walls in the Center Hub. The software also enforces a maximum angular acceleration of 33.7 rad/s^2. Do not override these limits in firmware.
6. **Student interaction** — Students may touch and reposition the ball during operation. The anti-windup clamp in the firmware prevents integral accumulation while the ball is held. The system will recover cleanly upon release.

4.5.3 Shutdown Procedure

1. Remove the ping-pong ball from the beam and set aside.

2. Allow the beam to return to level on its own. Do not force it manually while powered.
3. Flip the rocker switch to the OFF position and the stepper motor will de-energize immediately. The beam may drift slightly from level — this is normal.
4. If the robot is not used for more than 48 hours, connect the LiFePO4 charger and charge to full before storage. The battery is fully charged when the charger indicator light changes — refer to your specific charger model for the exact indicator behavior.
5. For transport: ensure the beam is near level. Place the robot in its transport case or padded bag. Do not stack anything on top of the beam.

4.6 Troubleshooting

4.6.1 Startup Issues

Problem	Likely Cause	Fix
Robot does not power on	Battery discharged or rocker switch connection loose	Charge battery fully. Check rocker switch is fully seated in its cutout and wired correctly.
Touchscreen blank on power-on	RP2040 not booting or loose touchscreen cable	Check touchscreen cable connection to GPIO06–GPIO15 on RP2040. Re-flash firmware.uf2 via BOOTSEL method if screen remains blank after reseating cable.
Motor does not auto-home on startup	TMC2209 driver not receiving power or VREF set too low	Check 12V rail from BMS to TMC2209 VM pin. Verify VREF potentiometer is set to correct holding current.
Auto-home runs but beam does not reach hard-stops	Motor losing steps during homing	Increase VREF slightly on TMC2209. Ensure D-shaft is fully engaged in beam sleeve.
Beam does not return to level after reset	Homing calibration did not complete cleanly	Manually move the beam to flat before placing the ball. The robot will re-engage balancing from the corrected position.

4.6.2 Operational Issues

Problem	Likely Cause	Fix
Ball does not settle — oscillates continuously	Kd too low, allowing overshoot to compound	Increase Kd on touchscreen in increments of 0.1 until oscillation dampens. Default Kd = 0.05.
Ball settles very slowly	Kp too low	Increase Kp in increments of 0.1. Default Kp = 0.75. If above 2.0, check ToF sensor calibration per Section 4.2.5.
Ball launched off beam on release after being held	Integral windup accumulated while ball was held manually	Verify anti-windup clamp is active in firmware. Temporarily reduce Ki. PID values reset on reboot — power cycle to restore defaults (Kp=0.75, Ki=0.30, Kd=0.05).

Problem	Likely Cause	Fix
Robot is violently unstable after PID values were adjusted	Gains set to extreme values on touchscreen	Turn the rocker switch OFF, wait 5 seconds, and turn back ON. PID values reset to factory defaults on every boot. To view or restore defaults, open src/configuration.hpp in the project GitHub and locate BEAM_KP, BEAM_KI, and BEAM_KD.
ToF sensor reading jumps or drops out	Sensor misaligned or direct sunlight interference	Realign sensor so lens faces exactly down beam axis. Check SCL/SDA connections at GPIO02/GPIO03 on RP2040. Reduce direct sunlight if possible.
Ball rolls off end of beam	Setpoint beyond beam length or ball placed while PID is saturated	Place ball at center before engaging PID. Verify setpoint is within 20–270mm from sensor.
PID response sluggish after transport	ToF sensor offset shifted from vibration	Re-run sensor calibration per Section 4.2.5 step 3.

4.6.3 Mechanical Issues

Problem	Likely Cause	Fix
Beam grinds or catches during pivot	Shaft Plug not fully seated or bearing misaligned	Power off. Remove railroad tie plates. Re-seat Shaft Plug fully into bearing. Reinstall tie plates.
Beam has side-to-side wobble	686 bearing loose in printed pocket	Press bearing back firmly. If it will not hold, apply a small amount of super glue around the outer ring and allow to cure fully before reassembling.
Motor hums but beam does not move	D-shaft slipping in beam sleeve	Power off. Re-seat D-shaft with flat side facing up. Tighten set screw.
Cracking sound at hard-stop	Center Hub hard-stop wall cracked	Inspect Center Hub for cracks. If cracked, reprint. Do not operate with a cracked hard-stop wall.
Acrylic window fell out	Slit tolerance too loose from print warping	Reprint Motor Holder in correct 90° orientation (Section 4.2.1). Temporary fix: small strip of clear tape inside the slit.
Chassis feels loose or flexes	Railroad tie plate screws backed out	Re-tighten all four M3 × 12mm railroad tie screws. Confirm tie plates are seated flat against both chassis halves.