

REACTION TIME MEASUREMENT AND TRAINING DEVICE

By

Sahil Aggarwal (sahil6)

Michael Griegel (griegel2)

Design Document for ECE 445, Senior Design, Fall 2026

TA: Zhuoer Zhang

2 October 2026

Project No. 46

Abstract

This document describes the design of a standalone reaction-time trainer: a 3×3 grid of illuminated buttons, a buzzer, an OLED and BLE logging, built on a custom ESP32-S3 PCB. A gearmotor-driven encoder wheel turning at 30 rpm sets a 1.9–4.3 s stimulus delay that the user cannot predict. The stimulus and response edges are timestamped by the same hardware capture timer and brought out to an external test port, so the reported reaction time can be checked with an oscilloscope against a ± 2 ms requirement. The document covers the subsystem designs and schematics, supporting calculations, requirements and verification, a tolerance analysis, cost, schedule, and ethical and safety considerations.

Contents

1 Introduction.....	1
1.1 Problem and Solution.....	1
1.2 Visual Aid.....	1
1.3 High-Level Requirements.....	2
2 Design.....	3
2.1 Block Diagram.....	3
2.2 Physical Design.....	3
2.3 Power Subsystem.....	5
2.4 Control and Timing Subsystem.....	7
2.4.1 Hardware timestamping.....	7
2.4.2 Support circuitry and pin assignment.....	8
2.4.3 Firmware.....	10
2.5 Electromechanical Delay Subsystem.....	12
2.5.1 Gearmotor selection.....	12
2.5.2 Motor driver.....	12
2.5.3 Encoder wheel and optical sensing.....	13
2.5.4 Speed control.....	14
2.6 Response Subsystem.....	16
2.7 Stimulus Subsystem.....	18
2.8 Display and Wireless Subsystem.....	19
2.9 Tolerance Analysis.....	21
2.9.1 Stimulus delay versus motor-speed tolerance.....	21
2.9.2 Reaction-time measurement error budget.....	22
3 Cost and Schedule.....	24
3.1 Labor.....	24
3.2 Parts.....	24
3.3 Total Cost.....	26
3.4 Schedule.....	27
4 Ethics, Safety, Societal Impact and Standards.....	28
4.1 Societal Impact.....	28
4.2 Engineering Standards.....	28
4.3 Ethics.....	28
4.4 Safety Concerns.....	29
4.5 Mitigation Procedures.....	29
References.....	30

1 Introduction

1.1 Problem and Solution

Reaction time matters in sports, competitive gaming, driving and rehabilitation, and most people measure it with a phone or computer app. Those platforms add their own latency and jitter. A 2025 study of Android phones found that a 120 Hz touchscreen alone adds about 8 ms of timing uncertainty, that audio stimuli arrived roughly 13 ms late on average, and that input/output buffer latencies ranged from 30 ms to 110 ms depending on the model [1]. Because this delay is inside the device, a user cannot separate their own reaction time from the latency of the phone. Dedicated light-and-button trainers avoid the operating system, but a four-pod BlazePod starter kit costs \$263 to \$329 [2], and commercial systems do not expose their stimulus and response signals for independent checking.

We will build a standalone, low-cost reaction-time trainer whose timing can be checked with a bench instrument. A 3×3 grid of illuminated arcade buttons and a buzzer present visual and audio stimuli, a 0.96-inch OLED shows the result, and a custom PCB built around an ESP32-S3-WROOM-1 module handles timing. The random wait before each stimulus comes from a physical mechanism instead of a software random-number generator: a gearmotor spins a two-track encoder wheel at 30 rpm, and the stimulus fires on the second index pulse after the trial is armed. This gives a 1.9 s to 4.3 s delay that the user cannot learn. The stimulus and response edges are timestamped in hardware, and the same two signals are brought out to an external test port, so the reported reaction time can be checked directly with an oscilloscope. Compared with phone tests, this removes display, touchscreen and OS latency from the measurement. Compared with commercial trainers, it costs roughly half as much in parts and its accuracy can be checked independently.

Changes since the proposal. We switched from the ESP32-WROOM-32E to the ESP32-S3-WROOM-1, which the ECE Electronics Shop stocks and whose native USB lets one USB-C connector both power and program the board. The H-bridge was replaced with a single low-side MOSFET because the encoder only turns in one direction. Timestamps now come from the ESP32-S3 hardware capture timer rather than being read inside an interrupt. A diode-OR "any button" signal was added so the test port shows the response edge for whichever button is pressed. The LED-current requirement now follows the purchased illuminated buttons (Section 2.7).

1.2 Visual Aid

Figure 1 shows the device in use. The user presses the center button to arm a trial, waits for one of the nine buttons to light (or for the buzzer), and presses the lit button. The result appears on the OLED, is stored locally, and is then sent over Bluetooth Low Energy (BLE) to a phone or laptop. An oscilloscope or logic analyzer can be connected to the external test port to measure the same interval.

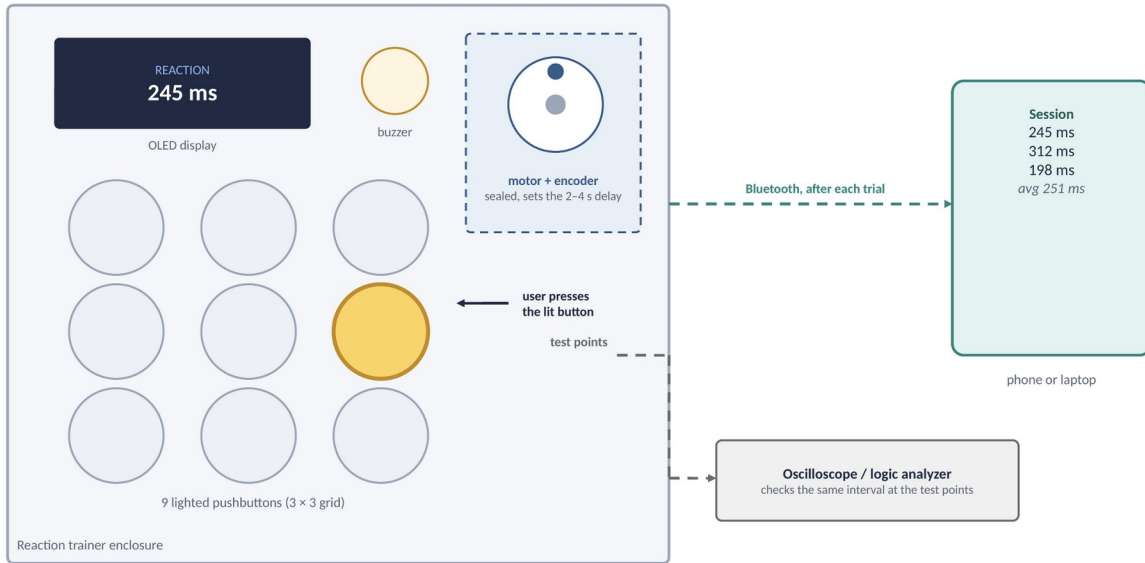


Figure 1: The device in use. The motor-driven encoder inside the enclosure sets when the stimulus occurs, the OLED reports the reaction time, BLE logs completed trials, and the test port lets an oscilloscope measure the same interval.

1.3 High-Level Requirements

The project solves the problem only if all three of the following requirements are met:

- The arming-to-stimulus delay shall be produced by the rotation of the motor-driven encoder wheel (the second index pulse after arming), shall fall between 1.9 s and 4.3 s, and exactly one stimulus shall occur in at least 95 of 100 consecutive armed trials.
- The device shall report reaction time with 1 ms display resolution, and the reported value shall agree with an oscilloscope or logic-analyzer measurement between the TP_STIM and TP_RESP test-port edges to within ± 2 ms in at least 95 of 100 valid trials.
- The device shall present all nine visual stimulus positions and the audio stimulus, and with Bluetooth disconnected it shall complete, display and store at least 95 of 100 consecutive valid trials without a reset.

2 Design

2.1 Block Diagram

Figure 2 shows the six subsystems. The **power subsystem** takes 5 V from a USB-C adapter, protects it with a fuse and TVS array, and produces a 3.3 V logic rail. The **control and timing subsystem** (ESP32-S3) timestamps the stimulus and response edges, runs the motor speed loop, selects the stimulus, and drives the display and BLE link. The **electromechanical delay subsystem** spins a two-track encoder wheel at 30 rpm and returns speed and index pulses; the index pulse is the only event that can start a stimulus. The **response subsystem** conditions the nine arcade switches and merges them into one RESP_ANY_N signal. The **stimulus subsystem** drives the nine button LEDs and the buzzer. The **display and wireless subsystem** shows results on the OLED and sends completed records over BLE. TP_STIM and TP_RESP go to an external test port. The interfaces between blocks are listed in Section 2.4 (Tables 3 and 4), so each block can be built and tested separately.

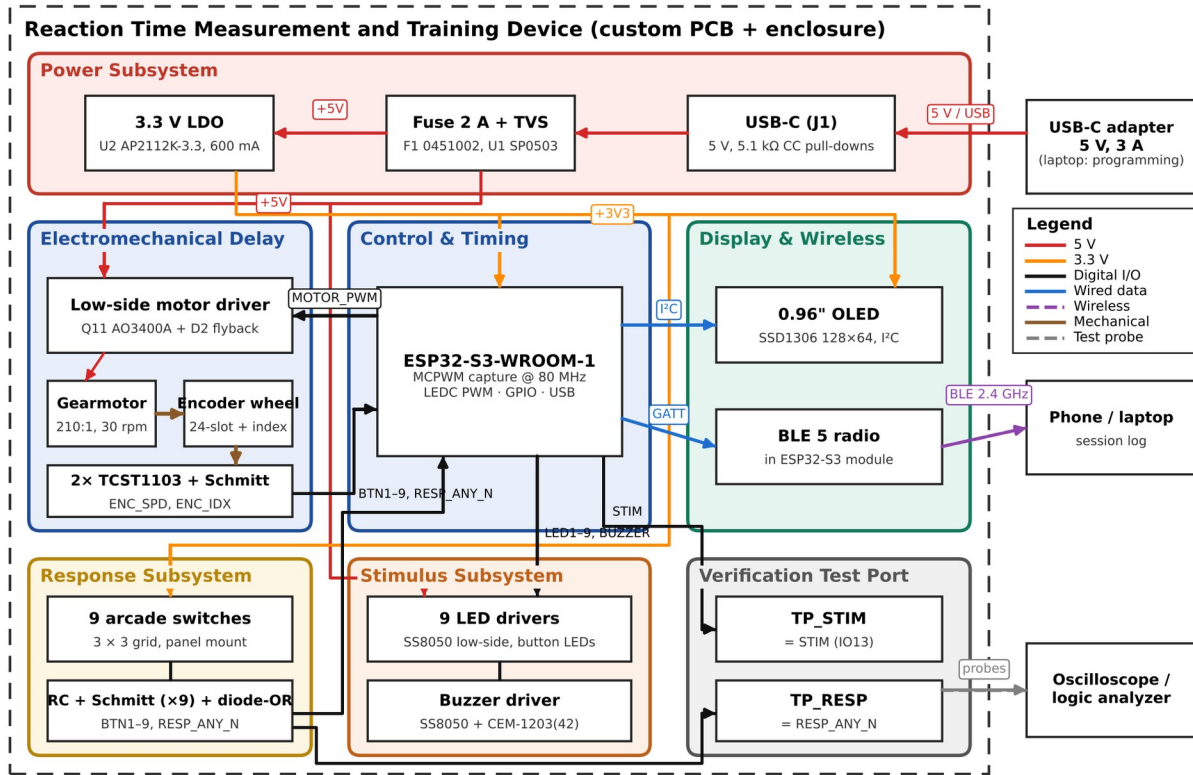


Figure 2: System block diagram. Red lines are 5 V, orange lines are 3.3 V, black lines are digital I/O, blue lines are wired data, the purple dashed line is wireless, brown lines are mechanical couplings and grey dashed lines are test-probe connections.

2.2 Physical Design

Figure 3 shows the enclosure, a 260 mm × 170 mm × 90 mm box made from 3 mm laser-cut acrylic. The nine 30 mm arcade buttons sit on a 45 mm pitch on the left of the lid, and the OLED and buzzer

grille are on the right. The 100 mm $\times$ 100 mm main PCB is mounted on 1/2 in standoffs under the right half of the lid, clear of the 35 mm deep button bodies. The PCB's right edge is 10 mm from the right wall, so the USB-C connector (J1) and the J8 test port, both on that edge, can be reached through slots in the wall without opening the box. The gearmotor and encoder wheel are mounted on the PCB inside a 3D-printed guard held on with screws. The guard keeps fingers and hair away from the moving parts and stops the user from seeing the wheel.

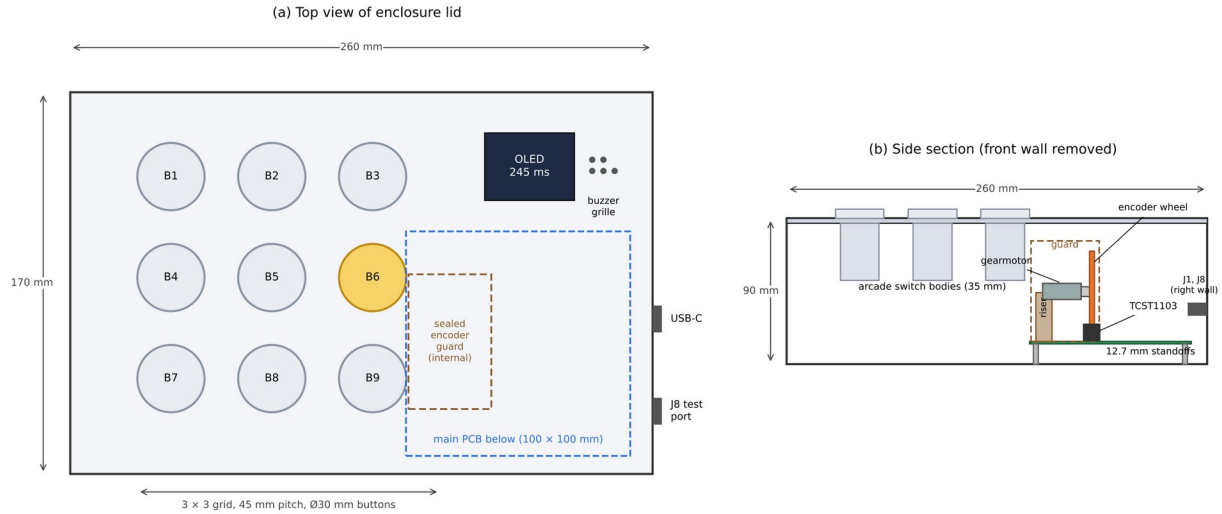


Figure 3: Physical design: (a) top view of the lid showing the button grid, OLED, buzzer, PCB outline and encoder guard; (b) side section showing the button bodies, standoff-mounted PCB, gearmotor on its riser, encoder wheel and interrupter.

Figure 4 shows the encoder wheel and the PCB floor plan. The wheel is 3D-printed as two $\varnothing 50$ mm, 1.6 mm thick disks, 14 mm apart on one hub that clamps onto the motor's 3 mm D-shaft through a Pololu #1079 mounting hub. Track A (speed) has 24 slots, each 7.5° wide and 3 mm deep. Track B (index) has one 14° notch. Each track passes through its own TCST1103 slotted interrupter (3.1 mm gap) soldered under the wheel. The datasheet does not give the exact height of the optical aperture, so we will set the wheel height with the printed motor riser and adjust it with shims during assembly. On the PCB, the ESP32-S3 antenna overhangs the top edge with a 15 mm copper keep-out, as Espressif recommends [3]. The motor and encoder zone is on the opposite side of the board, more than 20 mm from the antenna. J1, the power section and J8 are on the right edge, and the button-row connectors run along the bottom edge. If the routed board does not fit in 100 mm $\times$ 100 mm, the motor and encoder zone (J4, U6, U7) will move to a second small board joined by a JST-XH cable, which the course allows.

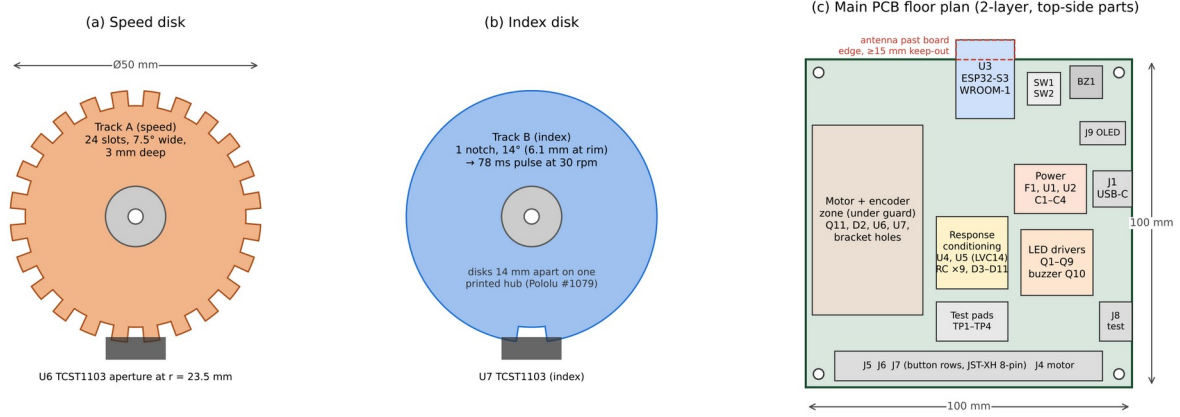


Figure 4: (a) Speed disk, 24 slots; (b) index disk, one notch; (c) floor plan of the 100 mm $\times$ 100 mm two-layer main PCB.

2.3 Power Subsystem

The power subsystem supplies every other block and supports all three high-level requirements, since a brown-out or reset during a trial would lose a measurement. It takes 5 V from a USB-C receptacle (J1, USB4085-GF-A, through-hole for mechanical strength). Two 5.1 k Ω pull-downs on CC1 and CC2 identify the board as a USB sink, so a standard USB-C adapter supplies 5 V at up to 3 A [4], [5]. VBUS passes through a 2 A fuse (F1) and an SP0503BAHTG TVS array (U1), which also protects the USB D+ and D- lines. The fused 5 V rail feeds the motor, button LEDs, buzzer and interrupter emitters directly. An AP2112K-3.3 LDO (U2) produces the 3.3 V logic rail [6]. We chose a linear regulator over a switching converter because the drop is only 1.7 V and a quiet supply is better for the timing logic. The same connector carries the ESP32-S3 USB data lines for programming. A laptop USB port supplies only 0.5–0.9 A, so during development we will program the board with the motor unplugged and use the 3 A adapter for normal operation.

Figure 5 shows the schematic, and Table 1 gives the load budget, built from datasheet maxima and the calculations in later sections.

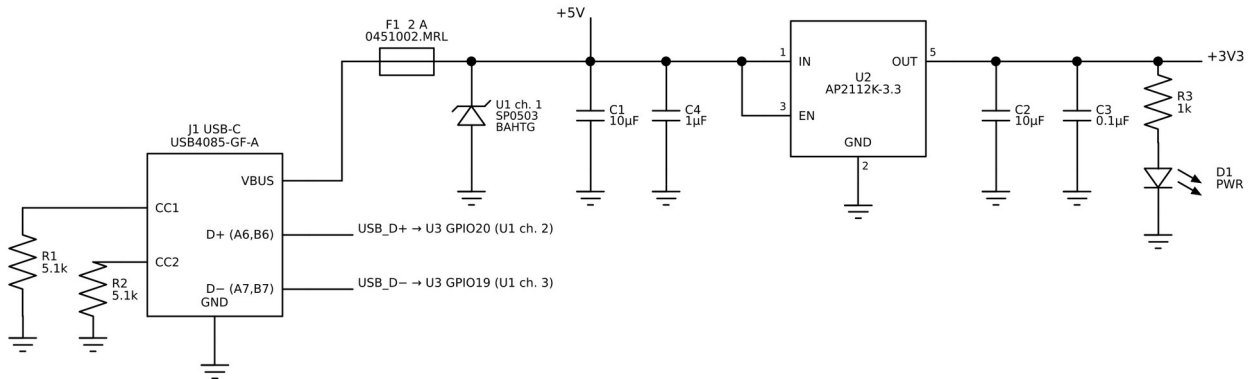


Figure 5: Power subsystem schematic: USB-C input with CC pull-downs, fuse, TVS, 5 V bulk capacitance and the AP2112K 3.3 V LDO.

Table 1: Power budget (worst case assumes all loads at their maxima simultaneously)

Load	Rail	Typical (mA)	Worst case (mA)	Basis
ESP32-S3-WROOM-1	3.3 V	≈80	355	Peak during RF transmit [7]
OLED module	3.3 V	≈10	25	≈20 mA typical, depends on pixels lit [8]
Pull-ups, power LED, 2 × 74LVC14	3.3 V	≈2	6	$11 \times 3.3 \text{ V} / 10 \text{ k}\Omega$, $(3.3 - 2) \text{ V} / 1 \text{ k}\Omega$
3.3 V subtotal (LDO output)	3.3 V	≈92	386	
Gearmotor (running / stalled)	5 V	≈60	300	0.36 A stall at 6 V, scaled to 5 V [9]
Button LEDs (one lit / all nine on)	5 V	≈10	113	Section 2.7
Buzzer	5 V	0	35	Section 2.7
Interrupter emitters (2)	5 V	≈42	50	Section 2.5
LDO input (= 3.3 V subtotal)	5 V	≈92	386	
Total drawn from USB-C	5 V	≈200	≈885	

Current margin. The worst-case 5 V draw of about 0.9 A is under a third of the 3 A adapter rating. The 2 A fuse sits above $1.25 \times$ the worst-case load, so it will not open in normal use but will open on a short. The LDO is rated for 600 mA against a worst case of 386 mA, a 55% margin, which exceeds the 25% we require.

Dropout. At the minimum adapter voltage of 4.75 V, and allowing about 0.1 V across the fuse and traces, the LDO input is at least 4.65 V. The AP2112K needs at most $3.3 \text{ V} + 0.4 \text{ V} = 3.7 \text{ V}$ at full load [6], which leaves almost 1 V of headroom.

$$P_{\text{LDO}} = (V_{\text{IN}} - V_{\text{OUT}}) \cdot I_{\text{OUT}} \approx (5.25 \text{ V} - 3.3 \text{ V})(0.15 \text{ A}) \approx 0.3 \text{ W} \quad (1)$$

Thermal check. Equation (1) estimates the sustained LDO dissipation, assuming an average load of about 150 mA (ESP32 with BLE active, plus the OLED). The 355 mA RF peaks last only milliseconds, so they do not set the temperature. With a conservative θ_{JA} of 250 °C/W for SOT-23-5 on minimal copper, the junction would run at roughly 100 °C at room temperature, below the 125 °C limit but with less margin than we would like. We will therefore place U2 on a copper pour stitched to the ground plane and measure its case temperature. If it exceeds 85 °C, we will switch to the AZ1117-3.3 in a TO-252 package, which the E-shop also stocks.

Decoupling and layout. Following the PCB-review guidance [5]:

- Every IC supply pin gets local decoupling, and there is 33 µF of bulk capacitance at the motor connector.
- Power traces are at least 0.5 mm (20 mil) wide, enough for 1 A on 1 oz copper per IPC-2221 [10].
- Both layers are filled with ground and stitched with vias.

- Test pads expose +5V, +3V3 and GND.

Table 2: Power Subsystem – Requirements and Verification

Requirement	Verification
1. With a 4.75–5.25 V input, the +3V3 rail stays within $3.3 \text{ V} \pm 5\%$ (3.135–3.465 V) for loads from 0 to 386 mA.	Power J1 from a current-limited bench supply. Load the +3V3 rail with an electronic load at 0, 100, 200, 300 and 386 mA, at input voltages of 4.75, 5.00 and 5.25 V, and measure +3V3 with a DMM. Pass: all 15 readings are within 3.135–3.465 V.
2. +3V3 stays within 3.135–3.465 V and the ESP32 does not reset during motor start, stall and stop.	With the full board running, monitor +3V3 on an oscilloscope while starting the motor 20 times and stalling the shaft for 2 s, with the supply limited to 1 A. Read the reset counter from the serial log. Pass: the rail stays within 3.135–3.465 V and the reset count is 0.
3. The supply and regulator handle the worst-case simultaneous load with $\geq 25\%$ current margin.	Run an "all-on" self-test for 60 s: all LEDs and the buzzer on, motor stalled, BLE transmitting. Measure the input current with a USB-C power meter and the LDO output current with a DMM. Pass: 5 V input $\leq 1.6 \text{ A}$ (2 A fuse / 1.25) and LDO output $\leq 480 \text{ mA}$ (600 mA / 1.25).
4. The device runs for $\geq 60 \text{ min}$ with the motor running, with no reset and no lost trials, and the U2 case stays below 85°C .	Run a 60-minute session with one trial every 10 s. B5 is pressed by an optocoupler driven from a function generator, and firmware answers each stimulus automatically. Afterwards, count the stored records and measure the U2 case temperature with a thermocouple. Pass: 0 resets, 360 ± 1 records stored, case below 85°C .
5. A 6.0 V supply fault (below the AP2112K's 6.5 V absolute maximum) causes no damage.	Apply 6.0 V to J1 for 10 s from a current-limited supply, return to 5.0 V, and repeat requirement 1 at 0 and 386 mA. Pass: +3V3 is still within 3.135–3.465 V.

2.4 Control and Timing Subsystem

The control and timing subsystem runs each trial and makes the measurement behind high-level requirement 2. It also runs the motor speed loop that requirement 1 depends on, and the state machine and local storage that requirement 3 depends on. It is built around the ESP32-S3-WROOM-1-N16 module (U3), a dual-core MCU with BLE 5 and native USB [7]. We chose it for four reasons: the E-shop stocks it at no cost; it has enough GPIO (26 used) for nine buttons and nine LEDs without an I/O expander; its radio is pre-certified, which covers the BLE requirement; and its native USB removes the USB-to-UART bridge the original ESP32 would need.

2.4.1 Hardware timestamping

If we read a timer inside an interrupt routine, the measured reaction time would include interrupt latency, and that latency changes when BLE or the display is busy. Instead we use the capture unit in the ESP32-S3's MCPWM peripheral [11]. A free-running hardware counter clocked at 80 MHz runs continuously, and when a selected input edge arrives, the hardware copies the counter value into a capture register. STIM (the stimulus reference) and RESP_ANY_N (the conditioned response) are routed to two capture channels on the **same** counter, so the reaction time is given by (2).

$$t_{RT} = (C_{RESP} - C_{STIM}) / f_{clk}, \quad f_{clk} = 80 \text{ MHz} \quad (2)$$

Here C_{STIM} and C_{RESP} are the captured counts. Software only *reads* the values after the edges have been latched, so interrupt latency affects when the result is computed, not the result itself. The counter resolution (12.5 ns) is far finer than the 1 ms the device displays, and its 32-bit range covers far longer than the 2 s response window. STIM is a GPIO output that must also be fed into a capture input through the ESP32-S3 GPIO matrix. Espressif’s documentation indicates this routing is possible, but we have not tested it yet, so we will confirm the full capture configuration on an ESP32-S3 DevKit before the PCB layout is frozen.

The firmware lights the selected LED and raises STIM in the same GPIO register write, so the edge on TP_STIM marks the start of the visual stimulus. For audio trials, STIM is raised and the buzzer tone is started immediately afterward. TP_STIM and TP_RESP carry the same two signals that are captured, so an oscilloscope on the test port measures the same interval the device reports. The later sections refer back to this point instead of repeating it.

2.4.2 Support circuitry and pin assignment

Figure 6 shows the module schematic. EN has a 10 k Ω pull-up and a 1 μ F capacitor, giving the reset delay Espressif recommends [7], plus a RESET button (SW1). The boot-strapping pin GPIO0 has a 10 k Ω pull-up and a BOOT button (SW2). The other strapping pins (GPIO3, GPIO45, GPIO46) are left unconnected, so they keep their default states. GPIO19 and GPIO20 connect to the USB-C D– and D+ pins, and UART0 goes to a debug header (J10). Tables 3 and 4 define every signal and connector that crosses a subsystem boundary.

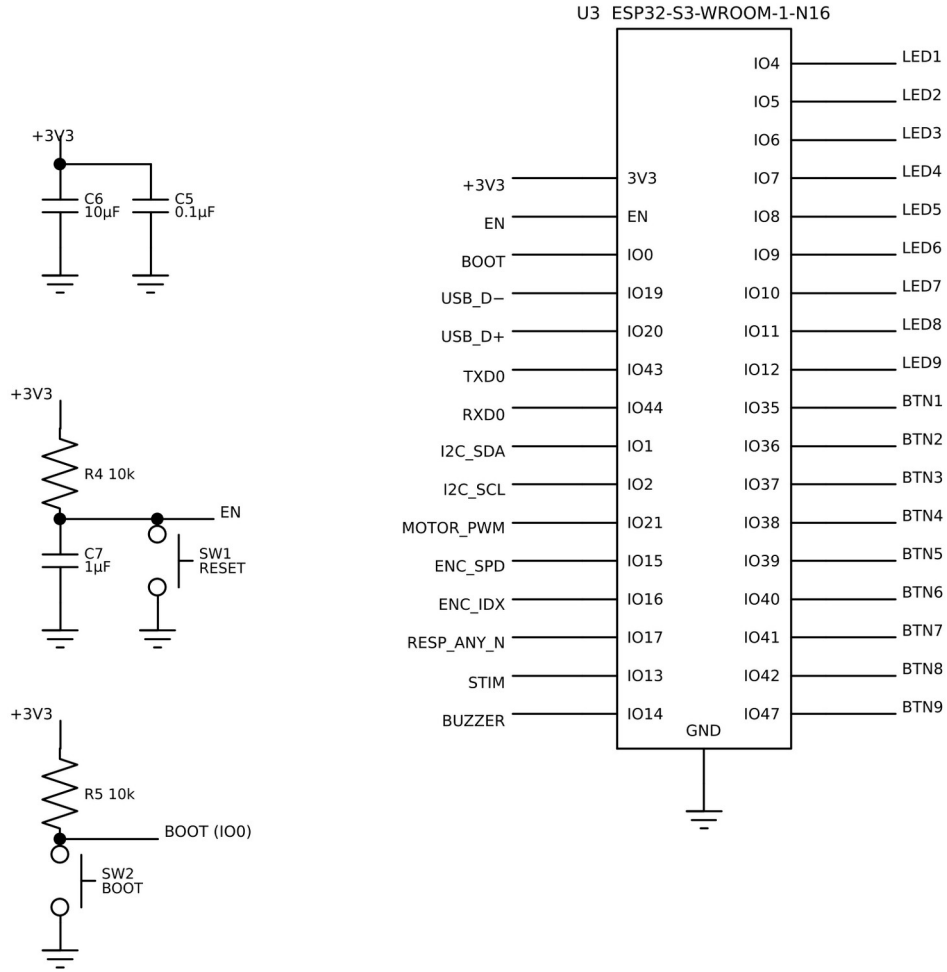


Figure 6: Control and timing subsystem schematic: ESP32-S3-WROOM-1 with reset and boot circuits, decoupling, and the net names shared with the other subsystems.

Table 3: Interface definition: ESP32-S3 pin assignment (all logic 0 V / 3.3 V)

Signal	GPIO	Dir.	Connects to	Notes
LED1–LED9	4–12	Out	Stimulus: Q1–Q9 base via 4.7 kΩ	≈0.5 mA per pin
STIM	13	Out (+capture)	TP_STIM (J8-1), capture input	Set together with the LED bit
BUZZER	14	Out (PWM)	Stimulus: Q10 base via 4.7 kΩ	2.048 kHz square wave
ENC_SPD	15	In	Electromech.: U5E output	≈12 Hz at 30 rpm; capture input
ENC_IDX	16	In	Electromech.: U5F output; J8-3	1 pulse/rev; capture input
RESP_ANY_N	17	In	Response: U5D output; TP_RESP (J8-2)	Active low on any press; capture input
MOTOR_PWM	21	Out (PWM)	Electromech.: Q11 gate via 150 Ω	20 kHz
I2C_SDA / SCL	1 / 2	I/O	Display: J9	400 kHz, 2.2 kΩ pull-ups
BTN1–BTN9	35–42, 47	In	Response: U4A–F, U5A–C outputs	High while pressed

Signal	GPIO	Dir.	Connects to	Notes
USB_D- / D+	19 / 20	I/O	Power: J1	Programming and debug
TXD0 / RXD0	43 / 44	I/O	J10 debug header	Serial log
DEBUG	18	Out	TP4 test pad	Spare / firmware marker
EN / BOOT	EN / 0	In	SW1 / SW2	RC reset; BOOT pulled high

Table 4: Off-board connectors

Ref.	Type	Pins	Purpose
J1	USB-C, USB4085-GF-A	VBUS, CC1/2, D+/D-, GND	5 V in, programming
J4	JST-XH 2-pin	+5V, MOTOR-	Gearmotor (switched low side)
J5, J6, J7	JST-XH 8-pin (one per button row)	+5V, GND, SW×3, LED_K×3	Arcade button switches and LEDs
J8	1×4 right-angle header, through the enclosure wall	TP_STIM, TP_RESP, ENC_IDX (each via 150 Ω , R54–R56), GND	External timing verification; series resistors protect against probe faults and ESD
J9	1×4 header	+3V3, GND, SCL, SDA	OLED module
J10	1×3 header	TXD0, RXD0, GND	Debug UART

2.4.3 Firmware

Figure 7 shows the trial state machine, and Figure 8 shows the timing of one trial:

- In READY the motor is speed-regulated and the OLED prompts the user. Releasing the center button B5 (after a 20 ms debounce) arms the trial.
- The first index edge is ignored. On the second index edge, the firmware starts the stimulus.
- Index edges that arrive much sooner than one revolution after the previous one are rejected as noise.
- When RESP_ANY_N is captured, the firmware reads the BTN1–BTN9 lines to identify the button, computes the reaction time, and only then updates the OLED and queues the BLE record.
- A press before the stimulus is a false start. No index within 2.5 s is an encoder fault. No press within 2.0 s, or the wrong button, is a no-response. Each is logged as an invalid trial.
- Completed records are kept in RAM and written to flash, so a reset does not erase trials that have already been recorded.

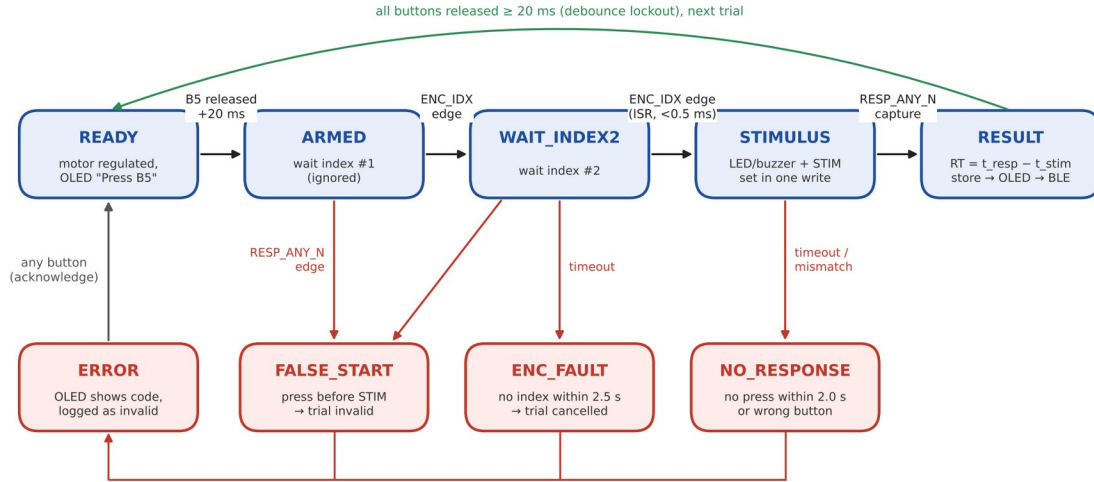


Figure 7: Trial state machine. Black arrows are normal transitions, red arrows are error paths, and the green arc returns to READY after all buttons have been released for at least 20 ms.

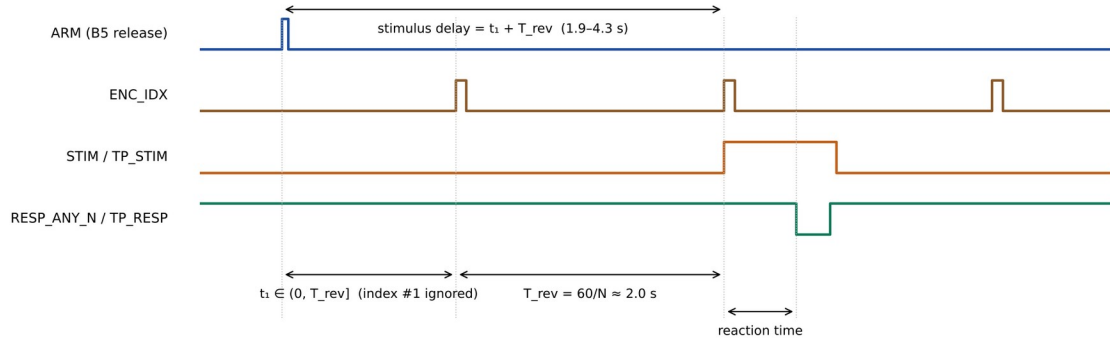


Figure 8: Timing of one trial. The stimulus delay is $t_1 + T_{rev}$; the reaction time is the interval between the STIM rising edge and the RESP_ANY_N falling edge, the same two edges available at TP_STIM and TP_RESP.

Table 5: Control and Timing Subsystem – Requirements and Verification

Requirement	Verification
1. The reported interval agrees with the oscilloscope TP_STIM-to-TP_RESP measurement within ± 1 ms for intervals from 1 ms to 5 s, with ≤ 1 μ s internal resolution.	In a test mode, a function generator supplies the stimulus and response edges at 20 known delays between 1 ms and 5 s. Compare each reported value (μ s resolution in the stored record) with an oscilloscope measurement between TP_STIM and TP_RESP. Pass: all 20 within ± 1 ms.
2. Timing does not depend on software load: results still agree within ± 1 ms while BLE is streaming and the OLED is updating.	Repeat the row 1 test at 100 ms and 1 s, with BLE notifications and continuous OLED refresh running. Pass: all readings within ± 1 ms of the oscilloscope.
3. The stimulus edge occurs ≤ 0.5 ms after the accepted (second) ENC_IDX edge.	Probe ENC_IDX (J8-3) and TP_STIM (J8-1) on a two-channel oscilloscope for 100 trials and measure the delay with cursors. Pass: maximum ≤ 0.5 ms.
4. Every button and both encoder inputs are identified independently: pressing button k records button k in 90 of 90 presses.	Press each of the nine buttons 10 times in diagnostic mode, which logs the button bitmap at each response. Spin the wheel by hand and watch ENC_SPD and ENC_IDX. Pass: 90/90 records show only the pressed button, and each encoder input toggles only for its own track.

Requirement	Verification
5. A press before the stimulus cancels the trial as a false start in 20 of 20 attempts.	Arm 20 trials and press a button before the stimulus. Pass: 20/20 show FALSE START, are stored as invalid, and no stimulus follows.
6. ≥ 100 consecutive trials run with 0 resets and 0 corrupted records, and the display and BLE update only after both timestamps are stored.	Run 100 trials and compare the stored records (dumped over USB) with the oscilloscope intervals and the CRC check. Confirm in code review and the serial log that the OLED and BLE updates follow the "timestamps stored" event. Pass: 0 resets, 100/100 records intact.

2.5 Electromechanical Delay Subsystem

This subsystem produces the stimulus delay from physical motion, which is high-level requirement 1. It has four parts: the gearmotor and its low-side driver, the two-track encoder wheel, two optical interrupters, and their Schmitt-trigger outputs. Its interfaces are MOTOR_PWM in (3.3 V logic, 20 kHz), ENC_SPD and ENC_IDX out (0 V / 3.3 V digital pulses), and the +5V and +3V3 rails.

2.5.1 Gearmotor selection

We need a small motor that turns at 30 rpm from 5 V with speed headroom to spare and draws little current. We chose the Pololu 210:1 Micro Metal Gearmotor LP 6V (#1096), rated at 65 rpm no-load and 0.36 A stall at 6 V [12], [9]. A simple DC-motor model fitted to these ratings (armature resistance $R = 6 \text{ V} / 0.36 \text{ A} \approx 17 \text{ } \Omega$, back-EMF constant $K_e \approx 0.78 \text{ V} \cdot \text{s/rad}$ at the output shaft, plus a friction term from the 40 mA no-load current) gives the PWM duty needed at 30 rpm ($\omega = 3.14 \text{ rad/s}$) in (3).

$$D = (K_e \omega + V_{\text{friction}}) / V_s \approx (2.5 \text{ V} + 0.7 \text{ V}) / 5 \text{ V} \approx 0.6 \quad (3)$$

At the lowest supply (4.75 V) the same model gives a maximum speed of about 50 rpm, so 30 rpm leaves roughly 60% headroom for the controller. This is why we chose the 210:1 ratio. The 298:1 and 380:1 versions would need more than 80% duty at 5 V, leaving little room to correct for load or supply changes. The high-power (HP) versions stall at 1.6 A, which would push the worst-case 5 V load close to the 2 A fuse rating. The wheel and hub are light enough that the motor reaches speed in a few milliseconds, much faster than the speed loop runs.

2.5.2 Motor driver

The motor only turns one way, so a single low-side N-MOSFET replaces the H-bridge from the proposal (Figure 9):

- **MOSFET:** we chose the AO3400A because its on-resistance is guaranteed at a 2.5 V gate drive (48 m Ω maximum) [13], so the 3.3 V GPIO turns it fully on. At the 0.3 A stall current it dissipates only a few milliwatts. We rejected the E-shop IRLML0030 because it is only specified at 4.5 V gate drive and its threshold can reach 2.3 V.
- **Gate network:** a 150 Ω gate resistor limits the GPIO current, and a 100 k Ω pull-down keeps the motor off while the ESP32 boots.
- **Flyback:** a CDBA540 Schottky diode (D2) clamps the motor's inductive kick to about 5.5 V, far below the 30 V MOSFET rating.

- **Noise:** a 0.1 μF capacitor across the motor and 33 μF of bulk capacitance at J4 keep brush noise off the supply rails.
- **PWM:** the PWM frequency is 20 kHz, above the audible range.

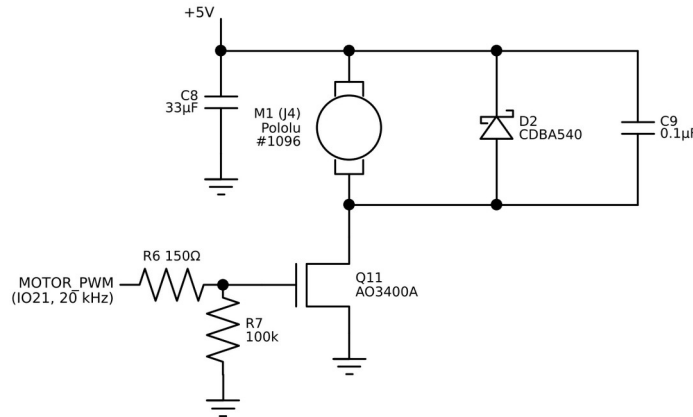


Figure 9: Motor driver: AO3400A low-side switch, CDBA540 flyback diode, snubber capacitor and gate network.

2.5.3 Encoder wheel and optical sensing

Each track is read by a Vishay TCST1103 slotted interrupter [14] (Figure 10). The component values come from three checks:

- **Emitter current:** the 180 Ω emitter resistor sets $I_F = (5\text{ V} - 1.25\text{ V}) / 180\text{ } \Omega \approx 21\text{ mA}$. Across the supply, resistor and forward-voltage tolerances it stays between about 17 mA and 25 mA, well under the 60 mA maximum.
- **Output current:** at the minimum I_F , the guaranteed 10% current-transfer ratio (CTR) gives at least 1.7 mA of collector current. The 10 k Ω pull-up needs only 0.33 mA to pull the output low, so there is about 5 $\times$ margin for LED aging.
- **Logic levels:** with the beam unblocked, the output is at most 0.4 V (the saturation voltage), below the 74LVC14's minimum low threshold of 0.6 V [15]. With the beam blocked, it rises to 3.3 V.

The phototransistor edges are slow (about 10 μs), so a Schmitt inverter squares them up so that each slot produces one clean edge. ENC_SPD and ENC_IDX are high while light passes through a slot or notch.

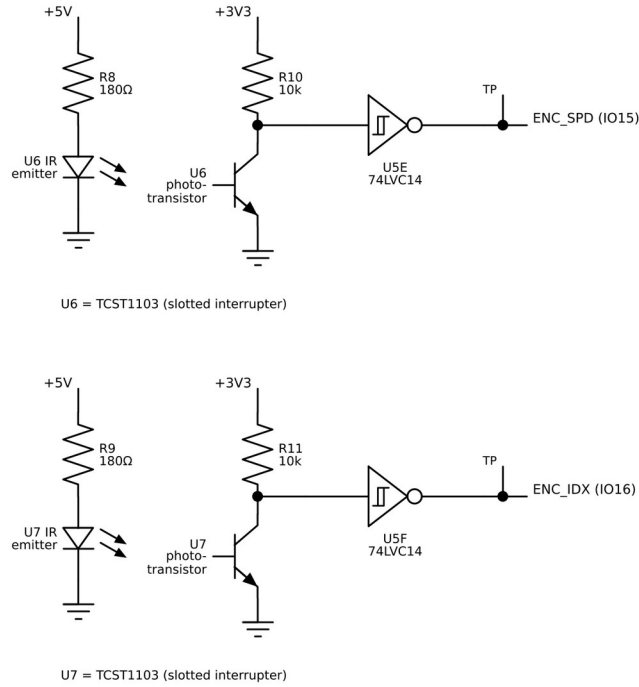


Figure 10: Encoder sensing: TCST1103 emitter and phototransistor, 10 k Ω pull-up, and 74LVC14 Schmitt inverter for the speed (top) and index (bottom) tracks.

At 30 rpm the wheel turns 180°/s:

- The speed track gives a pulse every 83 ms (12 Hz), each about 42 ms long.
- The index notch gives one pulse of about 78 ms every 2.0 s.
- Both openings are several times wider than the sensor's 1 mm aperture.

Small slot-placement errors from 3D printing make individual 15° periods slightly uneven. The 24 periods always add up to exactly one revolution, though, so the error averages out over each turn.

2.5.4 Speed control

Closed-loop control is needed because an open-loop PWM duty would let the speed drift with supply voltage, friction and motor temperature, and the delay range in requirement 1 depends directly on the speed (Section 2.9.1). The firmware measures speed from the time between successive ENC_SPD capture edges (12 updates per second at 30 rpm). A PI controller then adjusts the PWM duty, starting from the ≈ 0.6 feed-forward value in Section 2.5.1. Initial gains will be chosen from the motor model and tuned experimentally on the assembled motor and wheel, to hold every revolution at 30 rpm $\pm 5\%$. If a slot edge is missed, the next period is about twice as long, so the firmware rejects any period more than 1.5 $\times$ the previous one. Figure 11 is a model-based feasibility check, not a measured result. With preliminary gains, the simulated loop holds the speed inside the $\pm 5\%$ band through a supply step from 5.00 V to 4.75 V and a 50% friction increase.

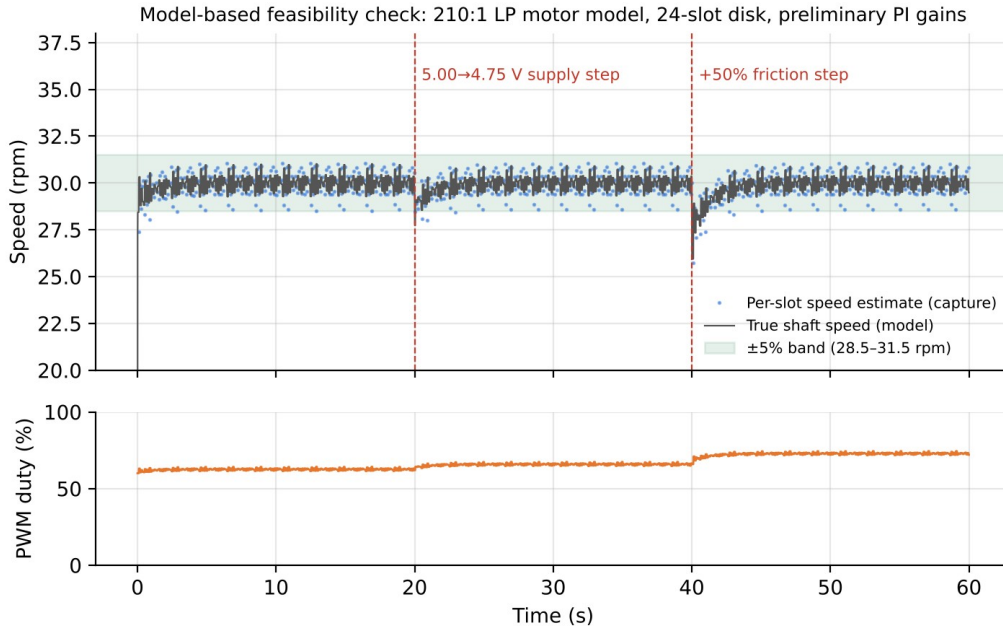


Figure 11: Model-based feasibility check of the speed loop (motor model fitted to datasheet ratings, simulated slot-placement error, preliminary PI gains). Final gains will be tuned on hardware.

Table 6: Electromechanical Delay Subsystem – Requirements and Verification

Requirement	Verification
1. Every revolution during a 60 s run stays at 30 rpm $\pm$ 5% (index-to-index period 1.905–2.105 s), at both 4.75 V and 5.25 V supply.	Record ENC_IDX for 60 s on a logic analyzer at each supply voltage and compute the speed of every revolution (60 / period). Pass: every period is within 1.905–2.105 s.
2. The wheel produces exactly 24 speed pulses and exactly 1 index pulse per revolution.	Capture ENC_SPD and ENC_IDX for 10 revolutions at 30 rpm and count the ENC_SPD pulses between consecutive index pulses. Pass: exactly 24 in all 10 revolutions, and no extra index pulses.
3. ENC_SPD and ENC_IDX are clean 3.3 V logic: high $\geq$ 2.9 V, low $\leq$ 0.4 V, one edge per slot.	Probe both outputs with an oscilloscope and count edges over 100 pulses. Pass: the levels are within limits and the edge count equals 2 $\times$ the pulse count.
4. The stimulus is triggered only by the second ENC_IDX edge after arming, never by a software timer.	Log ARM, ENC_IDX and TP_STIM on a logic analyzer for 50 trials. Then cover the index track and arm 10 more trials. Pass: in 50/50 trials TP_STIM rises at the second index edge after ARM; with the track covered, 0 of 10 trials produce a stimulus.
5. The arming-to-stimulus delay is 1.9–4.3 s, and $\geq$ 95 of 100 armed trials produce exactly one stimulus.	Measure ARM-to-TP_STIM for 100 trials on the logic analyzer and count the stimuli per trial. Pass: all delays are within 1.9–4.3 s and $\geq$ 95 trials have exactly one stimulus. Plot a histogram of the delays.
6. If no index edge arrives within 2.5 s of the expected time, the trial is cancelled as an encoder fault with no stimulus.	Arm 10 trials and unplug the motor right after arming each one. Pass: 10/10 trials report ENC_FAULT and TP_STIM stays low.
7. The flyback clamp keeps the Q11 drain $\leq$ 7 V during PWM and at turn-off.	Probe the Q11 drain with an oscilloscope while the motor runs and when PWM stops from full duty. Pass: peak $\leq$ 7 V (MOSFET rating 30 V).

2.6 Response Subsystem

The response subsystem turns nine mechanical arcade switches into clean 3.3 V edges for the ESP32 and the test port. It defines the response edge in high-level requirement 2, and together with the stimulus subsystem it makes all nine positions usable (requirement 3). The switches inside the Adafruit 30 mm LED arcade buttons [16] connect through J5 to J7 and pull node A to ground when pressed. Its outputs are BTN1–BTN9 (high while pressed) and RESP_ANY_N (low while any button is pressed), all 0 V / 3.3 V.

Figure 12 shows one of the nine identical channels:

- **Input:** a 10 k Ω pull-up holds node A at 3.3 V.
- **Filter:** a 1 k Ω series resistor and 10 nF capacitor form a low-pass filter at node B that removes ESD and noise spikes picked up on the button wiring.
- **Conditioning:** a 74LVC14 Schmitt inverter turns the filtered voltage into one clean edge.
- **Combining:** the nine BTN outputs go to the ESP32 and are also combined by BAT54 Schottky diodes [17] into an "ANY" node with a 10 k Ω pull-down. A twelfth Schmitt gate inverts ANY to RESP_ANY_N, which goes to both the capture input and TP_RESP (Section 2.4.1).

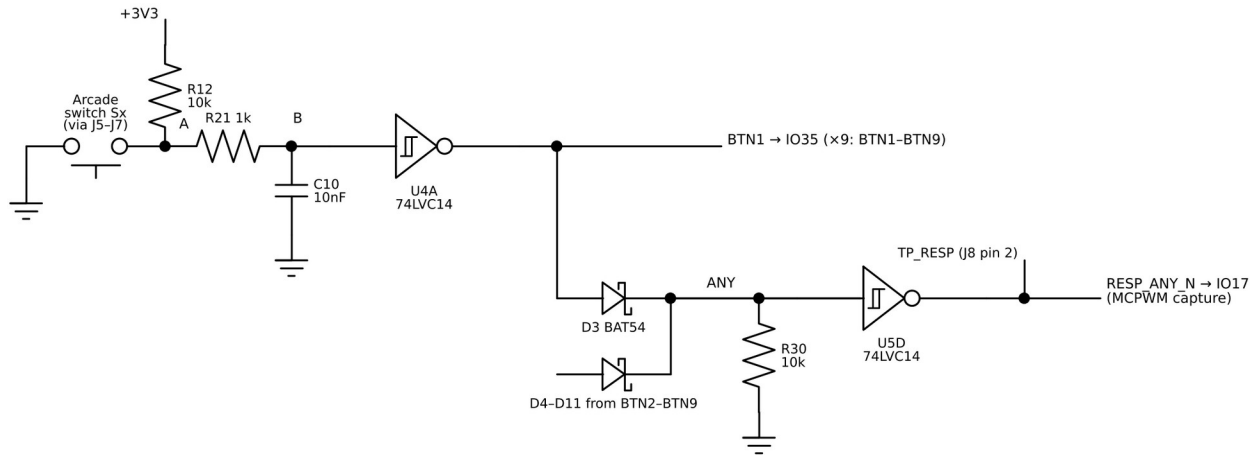


Figure 12: Response conditioning (one of nine channels shown) and the diode-OR that produces RESP_ANY_N for the capture input and the TP_RESP test point.

Press delay. When the switch closes, C discharges through R_s , and the output switches when V_B falls below the Schmitt trigger's lower threshold V_{T-} , as given by (4).

$$t_{\text{press}} = R_s C \cdot \ln(V_{CC} / V_{T-}) \quad (4)$$

With $R_s C = 10 \mu\text{s}$, this is about 11 μs at a typical V_{T-} , and under 20 μs across the full V_{T-} range and component tolerances. That is at least 50 times below our 1 ms limit, so the filter does not add meaningful delay. The delay also appears on both the capture input and TP_RESP, so it does not affect the comparison with the oscilloscope.

Bounce. When the switch bounces open, C recharges through $11\text{ k}\Omega$ ($\tau = 110\text{ }\mu\text{s}$). Openings shorter than about $35\text{ }\mu\text{s}$ cannot reach even the lowest upper threshold V_{T+} , so the filter removes them in hardware. Longer bounces are handled in firmware: the first edge is timestamped and the button is ignored for the next 20 ms. Figure 13 simulates a press with bounce.

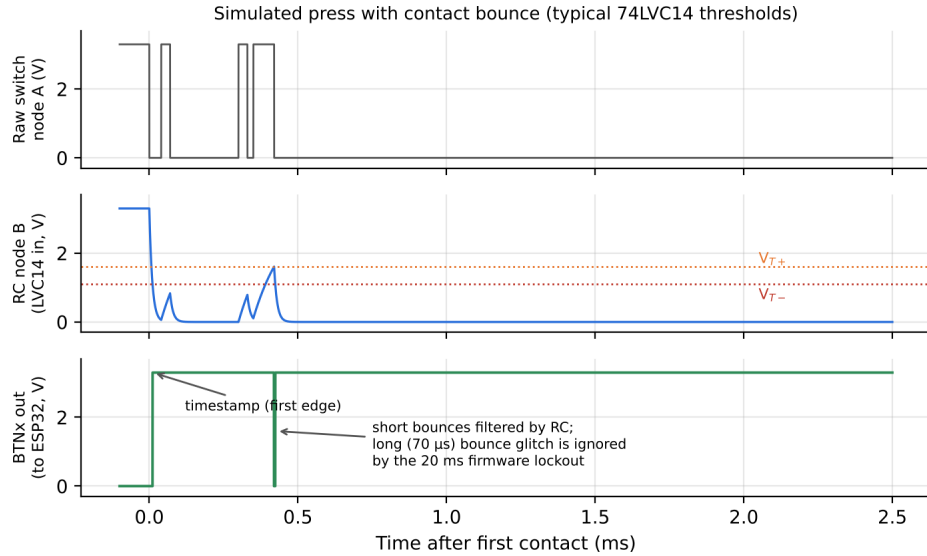


Figure 13: Simulated switch bounce at node A, the filtered node B, and the conditioned BTN output (typical 74LVC14 thresholds at 3.3 V).

Diode-OR levels. With one button pressed, $ANY \approx 3.3\text{ V}$ minus one Schottky drop ($\leq 0.32\text{ V}$ at 0.3 mA [17]), which is about 3.0 V , above the 2.0 V maximum upper threshold V_{T+} . With no button pressed, ANY is pulled to 0 V ; the only error is the gate input leakage ($\leq 5\text{ }\mu\text{A} \times 10\text{ k}\Omega = 50\text{ mV}$), far below the 0.6 V minimum V_{T-} . We chose a $10\text{ k}\Omega$ pull-down, rather than a larger value, so that the node discharges quickly when a button is released and is harder for noise to disturb.

Table 7: Response Subsystem – Requirements and Verification

Requirement	Verification
1. All nine buttons are detected independently; pressing one never asserts another BTN line (0 cross-talk errors in 90 presses).	Connect BTN1–BTN9 and RESP_ANY_N to a logic analyzer and press each button 10 times. Pass: in 90/90 presses only the matching BTN line goes high, and RESP_ANY_N goes low each time.
2. The delay from switch closure (node A) to the RESP_ANY_N edge is $< 1\text{ ms}$.	Probe node A and TP_RESP on an oscilloscope for 20 presses on each of three buttons. Pass: every delay $< 1\text{ ms}$.
3. One physical press produces exactly one accepted response, and edges within 20 ms after the first are ignored.	Make 100 presses in diagnostic mode, which logs raw and accepted response edges. Pass: 100/100 presses produce exactly one accepted response.
4. Conditioned outputs stay within logic limits: high $\geq 2.9\text{ V}$, low $\leq 0.4\text{ V}$, and never outside -0.3 V to 3.6 V .	Capture one BTN output and RESP_ANY_N on an oscilloscope during 20 presses and releases. Pass: all levels within the limits.

Requirement	Verification
5. TP_RESP is reachable at J8 with the enclosure closed and matches RESP_ANY_N at the ESP32 within 1 μ s.	With the enclosure closed, probe J8-2, and on the bring-up board probe the GPIO17 pad at the same time. Pass: J8 is reachable through the wall slot and the edge difference is $< 1 \mu$ s.

2.7 Stimulus Subsystem

The stimulus subsystem lights one of the nine button LEDs or sounds the buzzer, and provides the STIM reference edge. It supplies the nine visual positions and the audio stimulus in high-level requirement 3, and the stimulus edge in requirement 2. Its inputs are LED1–LED9, BUZZER and STIM (3.3 V logic). Its outputs are the button LED currents through J5–J7 and the buzzer sound.

LED drivers. Each Adafruit LED arcade button contains its own LEDs and a 200 Ω series resistor, and Adafruit specifies it at about 10 mA from 5 V [16]. The anode goes to +5V and the cathode (LED_Kx) to an SS8050 NPN low-side switch (Q1–Q9), as shown in Figure 14. A 4.7 k Ω base resistor gives about 0.5 mA of base current. That is roughly 20 times what the ≈ 10 mA LED needs given the transistor's minimum gain of 120, so the transistor is fully saturated, and each GPIO sources only about 0.5 mA. A 100 k Ω base pull-down keeps the LEDs dark while the ESP32 boots. The LED current follows (5).

$$I_{LED} = (V_S - V_{CE(sat)} - V_{F,eff}) / R_{int} \quad (5)$$

$V_{F,eff}$ is the effective LED voltage implied by Adafruit's rating (about 3.0 V; we allow 2.8–3.2 V because we have not measured it). Over the 4.75–5.25 V supply range, this gives roughly 6–13 mA. We therefore revised the proposal's 12–18 mA target to **6–13 mA**, because the purchased buttons set their own current. That level is still clearly visible indoors, and the requirement will be checked by measurement. Because the LED and STIM are switched together (Section 2.4.1), TP_STIM marks the start of the visual stimulus.

Buzzer driver. The CEM-1203(42) magnetic transducer from the self-service drawers is rated 85 dB at 10 cm when driven at 3.5 V and 35 mA at 2.048 kHz [18], which implies an impedance of about 100 Ω . From 5 V, the series resistor needs to drop about 1.3 V at 35 mA ($\approx 37 \Omega$), so we use the stocked 43.2 Ω . Q10 (SS8050) switches it from a PWM output, and a BAT54 diode clamps the coil's flyback. Free-field spreading from 10 cm to 50 cm costs about 14 dB, so we expect roughly 70 dB at 0.5 m, inside the 60–85 dBA requirement, and lowering the PWM duty makes it quieter. An audio stimulus also includes about 3 ms of acoustic travel and transducer rise time. This offset is consistent, so we will measure it and report it with audio results. It does not affect requirement 2, which compares electrical edges.

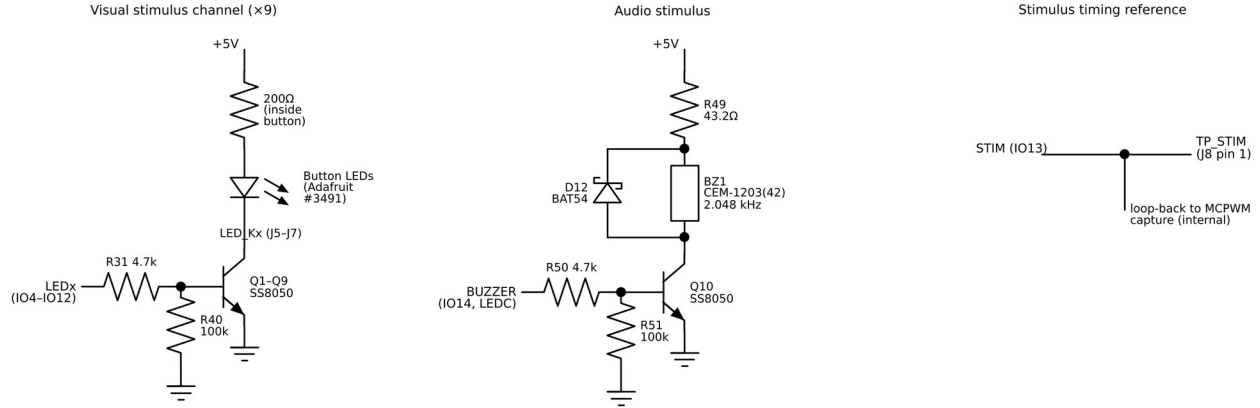


Figure 14: Stimulus subsystem: one of nine LED low-side drivers (the 200 Ω resistor and LEDs are inside the button), the buzzer driver with flyback diode, and the STIM timing reference routed to TP_STIM.

Table 8: Stimulus Subsystem – Requirements and Verification

Requirement	Verification
1. Each of the nine visual positions can be controlled independently, and a visual trial lights exactly one LED.	Turn LED1 to LED9 on one at a time in diagnostic mode, then run 90 random visual trials while logging the Q1–Q9 collectors. Pass: each command lights only its own button, and 90/90 trials light exactly one LED.
2. Each lit button draws 6–13 mA with a 4.75–5.25 V supply.	Measure the LED_K current of buttons 1, 5 and 9 with a DMM at 4.75, 5.00 and 5.25 V. Pass: all 9 readings within 6–13 mA.
3. The STIM edge and the LED drive edge occur within 10 μ s of each other.	Probe TP_STIM and the selected LED's transistor collector for 20 trials. Pass: difference $\leq 10 \mu$ s in 20/20 trials.
4. TP_STIM is reachable at J8 with the enclosure closed, with a high level ≥ 2.9 V.	Close the enclosure, probe J8-1 and run 5 trials. Pass: reachable without tools and high level ≥ 2.9 V.
5. The audio stimulus measures 60–85 dBA at 0.5 m.	In a quiet room, place a sound level meter (A-weighted) 0.5 m from the buzzer grille and trigger 10 audio stimuli. Pass: all readings within 60–85 dBA.
6. Each LED/BUZZER GPIO sources ≤ 1 mA.	Measure the voltage across a base resistor while its LED is on and compute $I = V / 4.7 \text{ k}\Omega$. Pass: ≤ 1 mA.

2.8 Display and Wireless Subsystem

This subsystem shows trial status and results, and optionally sends completed records to a phone or laptop. It runs only after a measurement is stored, so it cannot affect timing, and the device works fully with Bluetooth off, which is high-level requirement 3. Its interfaces are I²C (3.3 V, 400 kHz) to J9 and BLE 5 from the ESP32-S3 radio.

OLED. The Adafruit #326 0.96-inch 128 $\times$ 64 SSD1306 module [8] runs from +3V3, so its logic levels match the ESP32 (Figure 15). The I²C pull-up value is set by the 300 ns Fast-mode rise-time limit [19]. With about 100 pF of bus and cable capacitance, 4.7 k Ω gives about 0.4 μ s (too slow), while 2.2 k Ω gives about 0.2 μ s, so we use 2.2 k Ω . A full screen update takes about 25 ms at 400

kHz, well inside the 200 ms display requirement. The screen shows READY, ARMED, the result (for example "245 ms") and error codes (FALSE START, NO RESPONSE, ENC FAULT).

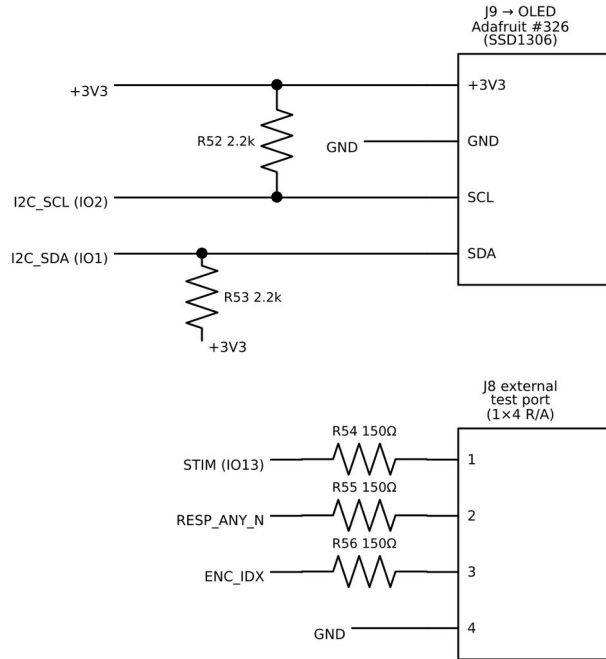


Figure 15: OLED connector J9 with I²C pull-ups (top) and the external test port J8 with series resistors (bottom).

BLE. The ESP32-S3 provides a custom GATT service with one notify characteristic (a value the device pushes to the connected host) [20]. After each completed trial, the 12-byte record in Table 9 is queued and sent. If no host is connected, records stay in device storage and are sent when one connects. A CRC byte lets the host confirm that the received record matches the stored one. A Python script using the *bleak* library [21] (or the nRF Connect phone app) logs sessions to CSV.

Table 9: BLE trial record (little-endian, 12 bytes)

Bytes	Field	Type	Description
0–1	session_id	uint16	User-chosen session number (no personal data)
2–3	trial_idx	uint16	Trial number within the session
4	trial_type	uint8	1–9 = visual position, 10 = audio
5	status	uint8	0 valid, 1 false start, 2 no response, 3 encoder fault, 4 wrong button
6–9	rt_us	uint32	Reaction time in μ s
10	delay_20ms	uint8	Arming-to-stimulus delay in 20 ms units
11	crc8	uint8	CRC-8 over bytes 0–10

Table 10: Display and Wireless Subsystem – Requirements and Verification

Requirement	Verification
1. The reaction time is displayed with 1 ms resolution within 200 ms of the response edge.	For 20 trials, measure the time from the TP_RESP edge to the end of the I ² C result transfer (logic analyzer on TP_RESP and SDA/SCL). Pass: whole-ms display and every delay ≤ 200 ms.
2. The display shows at least four distinguishable states: READY, ARMED, RESULT, ERROR.	Step through each state, including a deliberate false start, and photograph the screen. Pass: 4 distinct screens.
3. Each BLE record contains the session ID, reaction time and trial type, and ≥ 95 of 100 records arrive matching the stored records at 3 m.	Run 100 trials with the logging laptop 3 m away in line of sight, then compare its log with the device's stored records (dumped over USB). Pass: ≥ 95 records match field-for-field with a valid CRC.
4. With BLE disconnected, ≥ 95 of 100 valid trials complete, display and are stored.	Run 100 trials with no host connected, then dump the stored records over USB. Pass: ≥ 95 completed, displayed and stored.
5. Losing the BLE connection does not erase or change stored records.	Run 20 trials while connected, 10 with the host's Bluetooth off, then reconnect. Pass: all 30 records reach the host and match the device dump.

2.9 Tolerance Analysis

With our TA, we identified the electromechanical delay path as the riskiest part of the design. It combines motor control, optical sensing and moving parts, and it alone decides when a stimulus can occur, so it determines high-level requirement 1. We analyze it first, then summarize the error budget for the timing measurement in high-level requirement 2.

2.9.1 Stimulus delay versus motor-speed tolerance

For a shaft speed N (rpm), one revolution takes the time given by (6).

$$T_{\text{rev}} = 60 / N \quad (6)$$

When a trial is armed, the wheel can be at any angle, so the time to the next index edge, t_1 , is uniformly distributed on $(0, T_{\text{rev}}]$. The firmware ignores that edge and fires on the next one, which gives (7).

$$t_{\text{delay}} = t_1 + T_{\text{rev}}, \quad T_{\text{rev}} < t_{\text{delay}} \leq 2 T_{\text{rev}} \quad (7)$$

The requirement $1.9 \text{ s} \leq t_{\text{delay}} \leq 4.3 \text{ s}$ holds for every possible t_1 if $T_{\text{rev,min}} \geq 1.9 \text{ s}$ and $2 T_{\text{rev,max}} \leq 4.3 \text{ s}$. Substituting (6) gives the allowed speed range in (8).

$$60 / 2.15 = 27.9 \text{ rpm} \leq N \leq 60 / 1.9 = 31.6 \text{ rpm} \quad (-7.0\% / +5.3\%) \quad (8)$$

Our controller target is $30 \text{ rpm} \pm 5\%$ for every revolution (28.5–31.5 rpm), which lies inside this window. At the edges of that band, the delay limits are $60/31.5 = 1.905 \text{ s}$ and $2 \times 60/28.5 = 4.211 \text{ s}$, both within the requirement. The speed only has to hold over the at most two revolutions that make up one delay, and Electromechanical R&V 1 checks exactly that by measuring every index-to-index period.

Figure 16 shows a Monte Carlo check with 100 000 simulated trials. When the speed is spread anywhere in the $\pm 5\%$ band (worst case), every delay still falls between 1.9 s and 4.3 s. The delays are spread roughly evenly between about 2 s and 4 s (mean 3 s), so the user cannot predict the stimulus to better than a 2 s window.

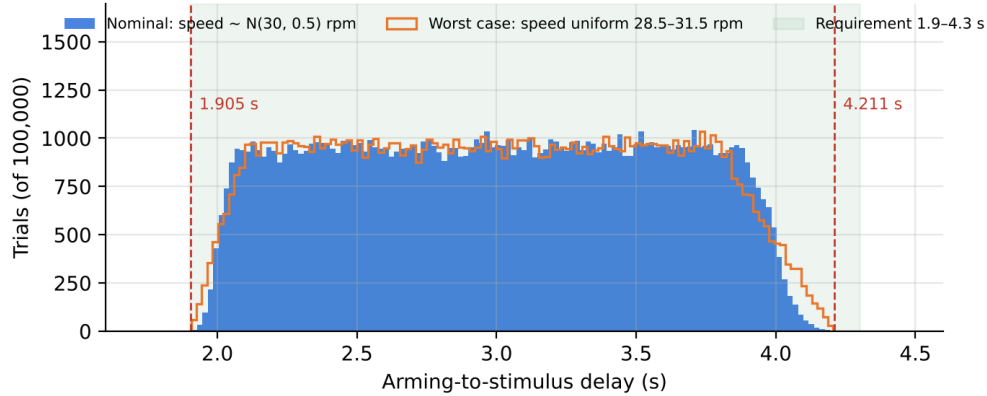


Figure 16: Monte Carlo distribution of the arming-to-stimulus delay (100 000 trials) for a nominal speed spread and for the worst case allowed by the $\pm 5\%$ speed requirement.

Exactly one stimulus per trial. Three failure modes could add or remove a stimulus, and each has a design response:

- A noise glitch on ENC_IDX: the Schmitt input and the firmware's rejection of index edges that arrive too early prevent it.
- A missed index edge: the 2.5 s watchdog records it as an encoder fault instead of firing late from a software timer.
- A stalled motor: the same watchdog catches it.

The trigger latency from the index edge to STIM is one interrupt, and we budget it at under 0.5 ms (Control R&V 3). That is negligible compared with the 1.9 s minimum delay.

2.9.2 Reaction-time measurement error budget

High-level requirement 2 compares the device's reported value with an oscilloscope measurement between the same two nets (Section 2.4.1). Table 11 lists the main contributors over the longest possible interval, the 2 s response window. Each is marked as a datasheet/analytical bound, a design estimate, or a quantity we will verify by test.

Table 11: Error budget for the device-versus-oscilloscope comparison (2 s interval)

Error source	Size	Type	Notes
Capture-timer quantization and input synchronization	$< 1 \mu\text{s}$	Analytical	80 MHz counter; affects both edges equally
ESP32 crystal and oscilloscope timebase	$\approx 0.1\text{--}0.2 \text{ ms}$	Estimate	Assumes tens of ppm for each clock over 2 s; to be confirmed against the scope
Signal path from capture pin to test port	$< 1 \mu\text{s}$	Analytical	Same net through a 150Ω resistor

Error source	Size	Type	Notes
Rounding the displayed value to 1 ms	± 0.5 ms	Analytical	Only when the displayed value is compared; the stored record has μ s resolution
Firmware/configuration errors (e.g. wrong edge captured)	—	Verified by test	Caught by Control R&V 1–2 and the 100-trial comparison

The analytical error budget is well below the ± 2 ms requirement, even with display rounding included. The main risk is not analog error but a logic or configuration mistake, which the oscilloscope comparison in Section 2.4 is designed to catch. Final accuracy will be established by that measurement, not by this analysis. The difference between the true human reaction time and the reported value (the 10–20 μ s conditioning delay for visual trials, plus about 3 ms of acoustic delay for audio trials) is discussed in Sections 2.6 and 2.7.

3 Cost and Schedule

3.1 Labor

Illinois ECE reports average starting salaries of \$87,429 for Electrical Engineering and \$119,064 for Computer Engineering graduates (2024–25) [22]. Their mean, \$103,000 per year over 2,080 working hours, is about \$50/hour, which we use for both team members. We estimate 150 hours per person over the remaining 10 weeks of the semester (about 15 h/week). Equation (9) gives the labor cost per partner.

$$\$50/\text{h} \times 2.5 \times 150 \text{ h} = \$18,750 \text{ per partner} \quad (9)$$

Table 12: Labor cost

Team member	Rate	Hours	Multiplier	Total
Sahil Aggarwal	\$50/h	150	2.5	\$18,750
Michael Griegel	\$50/h	150	2.5	\$18,750
Total labor		300		\$37,500

3.2 Parts

Table 13 lists every part with its manufacturer, part number, quantity and unit price. Parts marked E-shop or Self-service are free to the team through the ECE Electronics Shop, but they are costed at distributor list prices so that the total reflects the true cost of the device. Prices were checked on October 2, 2026; where a distributor did not show a quantity-1 price, the value is an estimate from comparable listings.

Table 13: Bill of materials (source names are purchase links)

Description	Manufacturer	Part number	Qty	Unit	Ext.	Source
ESP32-S3 Wi-Fi/BLE module, 16 MB flash	Espressif	ESP32-S3-WROOM-1-N16	1	\$6.10	\$6.10	E-shop
LDO 3.3 V 600 mA, SOT-23-5	Diodes Inc.	AP2112K-3.3TRG1	1	\$0.45	\$0.45	E-shop
USB-C receptacle, USB 2.0, through-hole	GCT	USB4085-GF-A	1	\$1.00	\$1.00	E-shop
Fuse 2 A, SMD	Littelfuse	0451002.MRL	1	\$0.70	\$0.70	E-shop
TVS array 5.5 V, SOT-143	Littelfuse	SP0503BAHTG	1	\$0.55	\$0.55	E-shop
Hex Schmitt inverter, TSSOP-14	Texas Instruments	SN74LVC14APWR	2	\$0.63	\$1.26	Digi-Key
Schottky diode 30 V 200 mA, SOT-23	onsemi	BAT54LT1G	10	\$0.10	\$1.00	Digi-Key
NPN transistor 25 V 1.5 A, SOT-23	Comchip	SS8050-G	10	\$0.10	\$1.00	E-shop
N-MOSFET 30 V 5.7 A, logic level, SOT-23	Alpha & Omega	AO3400A	1	\$0.52	\$0.52	Digi-Key
Schottky diode 40 V 5 A (flyback)	Comchip	CDBA540-HF	1	\$0.45	\$0.45	E-shop

Description	Manufacturer	Part number	Qty	Unit	Ext.	Source
Slotted optical interrupter, 3.1 mm gap	Vishay	TCST1103	2	\$1.96	\$3.92	Digi-Key
Magnetic transducer 3.5 V 2.048 kHz	CUI Devices	CEM-1203(42)	1	\$1.50	\$1.50	Self-service
Tactile switch (BOOT, RESET)	C&K	PTS645SL43SMTR92 LFS	2	\$0.20	\$0.40	E-shop
Green LED 0805 (power)	Lite-On	LTST-C171GKT	1	\$0.20	\$0.20	E-shop
Capacitor 0.1 μ F 16 V X7R 0603 (C3, C5, C19, C20)	KEMET	C0603C104K4RAC7867	4	\$0.05	\$0.20	E-shop
Capacitor 10 nF 16 V X7R 0603	KEMET	C0603C103K4RAC7867	9	\$0.05	\$0.45	E-shop
Capacitor 1 μ F 25 V X7R 0805	Samsung	CL21B105KAFNNNE	2	\$0.08	\$0.16	E-shop
Capacitor 10 μ F 10 V X5R 0603	—	GMC10X5R106M10NT	3	\$0.10	\$0.30	E-shop
Capacitor 33 μ F 10 V X5R 0805	TDK	C2012X5R1A336M125AC	1	\$0.40	\$0.40	E-shop
Capacitor 0.1 μ F 50 V X7R 0805	Yageo	CC0805KRX7R9BB104	1	\$0.05	\$0.05	E-shop
Resistor 10 k Ω 0603 (R4, R5, R10–R20, R30)	Yageo	RC0603FR-0710KL	14	\$0.02	\$0.28	E-shop
Resistor 1 k Ω 0603 (R3, R21–R29)	Yageo	RC0603FR-071KL	10	\$0.02	\$0.20	E-shop
Resistor 4.7 k Ω 0603 (R31–R39, R50)	Yageo	RC0603FR-074K7L	10	\$0.02	\$0.20	E-shop
Resistor 100 k Ω 0805 (R7, R40–R48, R51)	Stackpole	RMCF0805JT100K	11	\$0.02	\$0.22	E-shop
Resistor 2.2 k Ω 0805 (R52, R53)	Stackpole	RMCF0805JT2K20	2	\$0.02	\$0.04	E-shop
Resistor 5.1 k Ω 0805 (R1, R2)	Stackpole	RMCF0805JT5K10	2	\$0.02	\$0.04	E-shop
Resistor 180 Ω 1206 (R8, R9)	—	1206 180 Ω 5%	2	\$0.03	\$0.06	E-shop
Resistor 150 Ω 0603 (R6, R54–R56)	—	0603 150 Ω 1%	4	\$0.03	\$0.12	E-shop
Resistor 43.2 Ω 0603 (R49)	—	0603 43.2 Ω 1%	1	\$0.03	\$0.03	E-shop
JST-XH 8-pin header + housing + 8 contacts	JST	B8B-XH-A / XHP-8 / SXH-001T-P0.6	3	\$0.75	\$2.25	Digi-Key
JST-XH 2-pin header + housing + contacts	JST	B2B-XH-A / XHP-2	1	\$0.30	\$0.30	Digi-Key
2.54 mm straight pin header strip (J9, J10)	Sullins	PRPC040SAAN-RC (strip)	1	\$0.90	\$0.90	Self-service
2.54 mm right-angle pin header strip (J8)	Sullins	PRPC040SBAN-M71RC	1	\$1.20	\$1.20	Digi-Key
0.110 in (2.8 mm) insulated female quick-disconnects, 22–18 AWG (button wiring)	—	assorted, 50-pack	1	\$7.00	\$7.00	Amazon
Test point loops (TP1–TP4)	Keystone	5000	4	\$0.25	\$1.00	Digi-Key
Gearmotor 210:1 LP 6 V, 3 mm D-shaft	Pololu	#1096	1	\$23.95	\$23.95	Pololu
Micro metal gearmotor bracket pair	Pololu	#989	1	\$2.95	\$2.95	Pololu
Aluminum mounting hub, 3 mm shaft (2-pack)	Pololu	#1079	1	\$9.95	\$9.95	Pololu

Description	Manufacturer	Part number	Qty	Unit	Ext.	Source
30 mm arcade button with LED, clear	Adafruit	3491	9	\$2.50	\$22.50	Adafruit
0.96" 128×64 OLED, SSD1306, I ² C	Adafruit	326	1	\$17.50	\$17.50	Adafruit
USB-C power supply 5.1 V 3 A	Raspberry Pi	SC0213 (Adafruit 4298)	1	\$8.74	\$8.74	Adafruit
Hook-up wire 22 AWG stranded, 25 ft (black, red)	—	WH22-00-25 / WH22-02-25	2	\$6.04	\$12.08	ECE Supply Center Stock # 410182191 / 410182192
4-40 × 1/2 in brass hex standoff	—	1450C	4	\$0.63	\$2.52	ECE Supply Center Stock # 830145250
M3 × 8 mm screw + M3 hex nut (guard, riser, bracket)	—	M3 × 8 / 90591A121	12	\$0.16	\$1.92	ECE Supply Center Stock # 830717052 / 830820100
PLA filament for wheel, riser and guard (est. 150 g)	—	1.75 mm PLA	1	\$4.00	\$4.00	Amazon
3 mm cast acrylic sheet, 24 × 18 in (enclosure)	—	—	1	\$25.00	\$25.00	Amazon
2-layer PCB, 100 × 100 mm, qty 5	PCBWay	—	1	\$5.00	\$5.00	PCBWay (course order)
Parts subtotal					\$170.56	

Machine shop. The ECE Machine Shop will laser-cut the acrylic enclosure and drill the button, OLED and connector holes. We estimate 6 hours of shop labor (quote pending), valued at \$50/h = \$300. The encoder wheel, motor riser and guard will be 3D-printed by the team.

3.3 Total Cost

Table 14: Grand total

Item	Cost
Labor (Section 3.1)	\$37,500.00
Parts subtotal (Table 13)	\$170.56
Shipping (5% of parts)	\$8.53
Sales tax (10% of parts)	\$17.06
Machine shop (6 h × \$50/h)	\$300.00
Grand total	\$37,996.14

The parts cost of one device (excluding labor) is about \$171, roughly half the price of a four-pod commercial trainer [2].

3.4 Schedule

Table 15 assigns one specific task per member per week. Sahil Aggarwal owns the hardware (schematic, PCB, power, assembly and mechanical work), and Michael Griegel owns the firmware (timing, motor control, user interface and BLE). Course milestones are shown in bold; their exact dates follow the Fall 2026 ECE 445 calendar. Each PCB order wave takes 7 to 10 days from submission to delivery [5], so the first order is scheduled for week 2 to leave time for a second revision.

Table 15: Weekly schedule and division of labor

Week of	Sahil Aggarwal (hardware)	Michael Griegel (firmware)
Oct 5	Design Review. Finish the KiCad schematic from the Chapter 2 schematics, run ERC, and assign footprints (ESP32-S3, USB4085, TCST1103, JST-XH).	On an ESP32-S3 DevKit: confirm the STIM/RESP hardware-capture configuration (including STIM loop-back) before the PCB layout is frozen; 20 kHz PWM; BLE notify test.
Oct 12	PCB layout (antenna keep-out, ground pour, test points), DRC with the PCBWay rules, submit the first PCB order ; place the Digi-Key, Pololu and Adafruit orders.	Breadboard demo: one button channel (RC + 74LVC14) and the trial state machine on the DevKit; check timing against the scope.
Oct 19	Breadboard the motor driver and the TCST1103 sensors; CAD and print the encoder wheel, motor riser and guard.	Motor PI speed loop with ENC_SPD capture on the breadboard; index counting, blanking and ENC_FAULT watchdog.
Oct 26	PCB arrives: assemble and verify the power section (Power R&V 1, 3, 5) before fitting U3.	OLED driver and screens (READY, ARMED, RESULT, ERROR); record format, ring buffer and CRC-8.
Nov 2	Fit U3, the logic and the drivers; flash a blink test; verify the response and stimulus hardware. Progress demo.	Port the firmware to the PCB pin map; run the Control R&V 1–3 with a function generator.
Nov 9	Enclosure (machine shop), wire the button harness to J5–J7, mount the motor and wheel; fix any PCB errors and send a revision order if needed.	BLE GATT service and Python <i>bleak</i> logging script; Display/Wireless R&V 3–5.
Nov 16	Electromechanical R&V (60 s speed, 24/1 pulse count, flyback) and the 60-min power test. Mock demo.	High-level requirement tests: 100 trials scope versus device (± 2 ms), 100-trial delay histogram, 100 trials with BLE off.
Nov 23	Fall break: buffer for hardware rework; take the U2 thermal measurement.	Fall break: buffer for firmware bug fixes; audio offset measurement with a microphone.
Nov 30	Final safety check (guard, fuse, standoffs); prepare the demo bench with the scope on J8. Final demo.	Demo script and data plots; final presentation slides.
Dec 7	Final report: hardware chapters; lab checkout.	Final report: firmware and verification chapters; lab notebook.

4 Ethics, Safety, Societal Impact and Standards

4.1 Societal Impact

Public health and welfare. The device gives athletes, gamers, older adults and rehabilitation users an affordable way to train and track reaction time with honest, verifiable numbers. It is a training tool, not a medical device. Its results will not be presented as a diagnosis or as a measure of intelligence, athletic ability or fitness to drive. **Economic.** About \$171 in parts is roughly half the price of a four-pod commercial trainer [2], and because the design uses common parts and exposes test points it can be repaired rather than replaced. **Environmental.** The device has no batteries, is powered from a reusable USB-C adapter, and uses snap-in arcade buttons and connectorized wiring so that individual parts can be replaced. **Social, cultural and global.** Lights, a tone and numbers need no language or internet connection, and the device works without a phone, so access does not depend on owning a particular platform.

4.2 Engineering Standards

- **IEEE Code of Ethics and ACM Code of Ethics** [23], [24]: these govern our accuracy claims, privacy and human testing (Section 4.3).
- **IPC-2221C** [10]: trace widths (≥ 0.5 mm for 1 A on the 5 V rail) and conductor spacing on the PCB.
- **USB Type-C specification** [4]: 5.1 k Ω Rd pull-downs on CC1 and CC2 so that the board is a compliant 5 V sink; we draw at most 0.9 A from a 3 A source.
- **FCC 47 CFR Part 15** [25]: we use the pre-certified ESP32-S3 module, keep its antenna configuration unchanged and follow Espressif's layout keep-out [3]. We recognize that a certified module does not by itself certify the finished product.
- **Bluetooth Core Specification 5.0** [20] for the BLE GATT link, and **NXP UM10204 (I²C)** [19] for the OLED bus timing (Section 2.8).
- **ISO 12100:2010** [26]: risk assessment and fixed guarding of the rotating encoder wheel. **IEC 62368-1** [27]: we keep all accessible circuits in the lowest energy class (≤ 5.25 V DC) and keep accessible surfaces cool.
- **ANSI/ESD S20.20** [28]: we use grounded wrist straps and ESD mats when handling the ESP32 and 74LVC14 parts, and the TVS array (U1), the RC input filters and the 150 Ω series resistors on the J8 test port protect the inputs and outputs that users and probes can touch.

4.3 Ethics

Following the IEEE Code of Ethics [23], which asks engineers to be "honest and realistic in stating claims or estimates based on available data," every accuracy figure we publish will come from our own oscilloscope measurements (Section 2.4 and high-level requirement 2), not from datasheet numbers. We will keep the displayed **resolution** (1 ms) separate from the **accuracy** we verify against the oscilloscope (± 2 ms), and we will report the ≈ 3 ms acoustic offset of audio trials instead of hiding it. In line with the ACM Code's principles of avoiding harm and respecting privacy [24], the BLE record (Table 9) contains only a user-chosen session number, the trial type and timing

values, with no names, health or demographic data, and data are sent only to a host the user pairs. Testing validates the device, not people; if the project grows into systematic human-subject research, we will first get an IRB determination [29], and classmates who try the device do so voluntarily. AI tools were used to help draft parts of this document and to check calculations. The team reviewed every design decision and takes responsibility for it, and adapted reference circuits are cited.

4.4 Safety Concerns

Following the course safety guidelines [30], the project is low risk: it uses no batteries, no voltages above 5.25 V DC, no lasers and no flying parts, so a separate safety manual is not required. The following hazards remain, including two that arose after the proposal:

- **Rotating machinery.** The encoder wheel turns at only 30 rpm with 1.6 kg·cm stall torque, but fingers, hair or a loose drawstring could still be pinched or caught.
- **Electrical and thermal.** A short on the 5 V rail could overheat traces or the adapter, and the LDO can reach about 100 °C under worst-case load (Section 2.3).
- **Acoustic.** The buzzer must stay below levels that are uncomfortable at close range; the OSHA 8-hour limit is 90 dBA [31], and our design target is ≤ 85 dBA at 0.5 m with tones under 1 s.
- **New since the proposal: acrylic edges and laser-cut parts,** which could cut during assembly, and **USB-C hot-plugging** during bring-up, which could stress the ESP32 pins.
- **Assembly.** Soldering burns and flux fumes, and ESD damage to parts.

4.5 Mitigation Procedures

1. **Guarding:** the motor and wheel are fully enclosed in a 3D-printed guard fastened with M3 screws (tool needed to open), and the guard sits inside the closed enclosure. Unplug USB-C before servicing; the guard is off while powered only for the hand-stall test (Power R&V 2), done with a gloved hand and a 1 A supply limit.
2. **Current limiting:** first power-ups use a 0.5 A current-limited supply, rails are checked before U3 is fitted, and the 2 A fuse protects against shorts.
3. **Thermal:** U2 sits on a copper pour; we measure its case temperature during the 60-minute test, with an 85 °C limit and a planned fallback part.
4. **Acoustic:** buzzer duty and duration are capped in firmware (≤ 1 s per tone), and the sound level is verified with a meter (Stimulus R&V 5).
5. **Assembly and ESD:** we follow the DRS soldering guidance [32]: ventilated stations with fume extraction, eye protection, no soldering on powered boards, and washing hands after soldering. We use lead-free solder where practical, ESD straps and mats (ANSI/ESD S20.20), and we deburr acrylic edges and wear safety glasses when drilling.
6. **User data and claims:** no personal data are transmitted, and the device and documentation state that results are for training only.

References

- [1] U. Roussel, E. Fléty, C. Agon, I. Viaud-Delmon, and M. Taffou, "Using Android smartphones to collect precise measures of reaction times to multisensory stimuli," *Sensors*, vol. 25, no. 19, Art. no. 6072, 2025, doi: 10.3390/s25196072.
- [2] BlazePod, "BlazePod Starter Kit," product page. [Online]. Available: <https://www.blazepod.com/products/starter-kit>. [Accessed: Oct. 2, 2026].
- [3] Espressif Systems, "ESP32-S3 Hardware Design Guidelines: PCB Layout Design." [Online]. Available: <https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32s3/pcb-layout-design.html>. [Accessed: Oct. 2, 2026].
- [4] USB Implementers Forum, "Universal Serial Bus Type-C Cable and Connector Specification," Release 2.3, Oct. 2023. [Online]. Available: <https://www.usb.org/document-library>
- [5] J. Jung, "PCB Review," ECE 445 lecture slides, University of Illinois Urbana-Champaign, 2025.
- [6] Diodes Incorporated, "AP2112: 600mA CMOS LDO Regulator with Enable," datasheet. [Online]. Available: <https://www.diodes.com/assets/Datasheets/AP2112.pdf>
- [7] Espressif Systems, "ESP32-S3-WROOM-1 & ESP32-S3-WROOM-1U Datasheet," v1.8, Nov. 2023. [Online]. Available: https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf
- [8] Adafruit Industries, "Monochrome 0.96 inch 128x64 OLED Graphic Display – STEMMA QT," product 326. [Online]. Available: <https://www.adafruit.com/product/326>. [Accessed: Oct. 2, 2026].
- [9] Pololu Corporation, "Micro Metal Gearmotors," datasheet rev. 6.2. [Online]. Available: <https://www.pololu.com/file/0J1487/pololu-micro-metal-garmotors-rev-6-2.pdf>
- [10] IPC, *IPC-2221C: Generic Standard on Printed Board Design*. Bannockburn, IL, USA: IPC, 2023.
- [11] Espressif Systems, "ESP-IDF Programming Guide (ESP32-S3): Motor Control Pulse Width Modulator (MCPWM)." [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/api-reference/peripherals/mcpwm.html>. [Accessed: Oct. 2, 2026].
- [12] Pololu Corporation, "210:1 Micro Metal Gearmotor LP 6V (item #1096)," product page. [Online]. Available: <https://www.pololu.com/product/1096>. [Accessed: Oct. 2, 2026].
- [13] Alpha and Omega Semiconductor, "AO3400A 30V N-Channel MOSFET," datasheet. [Online]. Available: <https://www.aosmd.com/res/datasheets/AO3400A.pdf>
- [14] Vishay Semiconductors, "TCST1103, TCST2103, TCST2203: Transmissive Optical Sensor with Phototransistor Output," doc. 83764, rev. 2.0, Aug. 2011. [Online]. Available: <https://www.vishay.com/docs/83764/tcst1103.pdf>
- [15] Texas Instruments, "SN74LVC14A Hex Schmitt-Trigger Inverter," SCAS285AC, Apr. 2022. [Online]. Available: <https://www.ti.com/lit/ds/symlink/sn74lvc14a.pdf>
- [16] Adafruit Industries, "Arcade Button with LED – 30mm Translucent Clear," product 3491. [Online]. Available: <https://www.adafruit.com/product/3491>. [Accessed: Oct. 2, 2026].
- [17] onsemi, "BAT54LT1G Schottky Barrier Diode," datasheet BAT54LT1/D. [Online]. Available: <https://www.onsemi.com/download/data-sheet/pdf/bat54lt1-d.pdf>
- [18] CUI Devices, "CEM-1203(42): Magnetic Audio Transducer, Externally Driven, 3.5 V, 2.048 kHz," datasheet; listed in the UIUC ECE Electronics Shop self-service inventory, Fall 2026.
- [19] NXP Semiconductors, "UM10204: I2C-bus specification and user manual," Rev. 7.0, Oct. 2021. [Online]. Available: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>

- [20] Bluetooth SIG, "Bluetooth Core Specification," Version 5.0, Dec. 2016. [Online]. Available: <https://www.bluetooth.com/specifications/specs/>
- [21] H. Blidh, "Bleak: Bluetooth Low Energy platform agnostic client for Python," GitHub repository. [Online]. Available: <https://github.com/hbldh/bleak>. [Accessed: Oct. 2, 2026].
- [22] Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, "Salary Averages." [Online]. Available: <https://ece.illinois.edu/admissions/why-ece/salary-averages>. [Accessed: Oct. 2, 2026].
- [23] IEEE, "IEEE Code of Ethics," IEEE Policies, Sec. 7.8. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>
- [24] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," 2018. [Online]. Available: <https://www.acm.org/code-of-ethics>
- [25] Federal Communications Commission, "Radio Frequency Devices," 47 CFR Part 15. [Online]. Available: <https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15>
- [26] ISO, *ISO 12100:2010, Safety of Machinery – General Principles for Design – Risk Assessment and Risk Reduction*. Geneva, Switzerland: ISO, 2010.
- [27] IEC, *IEC 62368-1:2023, Audio/Video, Information and Communication Technology Equipment – Part 1: Safety Requirements*. Geneva, Switzerland: IEC, 2023.
- [28] EOS/ESD Association, *ANSI/ESD S20.20-2021: Protection of Electrical and Electronic Parts, Assemblies and Equipment*. Rome, NY, USA, 2021.
- [29] University of Illinois Urbana-Champaign Office for the Protection of Research Subjects, "Does my study need IRB approval?" [Online]. Available: <https://oprs.research.illinois.edu/>. [Accessed: Sep. 18, 2026].
- [30] ECE 445 Senior Design Laboratory, "Safety," course guidelines, University of Illinois Urbana-Champaign. [Online]. Available: <https://courses.grainger.illinois.edu/ece445/guidelines/safety.asp>
- [31] Occupational Safety and Health Administration, "Occupational Noise Exposure," 29 CFR 1910.95. [Online]. Available: <https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.95>
- [32] University of Illinois Urbana-Champaign Division of Research Safety, "Soldering Safety," DRS Safety Library. [Online]. Available: <https://drs.illinois.edu/>. [Accessed: Sep. 18, 2026].