Technical Design Document

Battlebots Relay Maze Robots

Platform: Arduino Nano (ATmega328P) · By Team C

BATTLEBOTS®

Version	Date	Status	Authors	Description
1.0	12/02/26	Draft	Annemiek Veld & Herbert Dorothea	Initial draft
2.0	26/02/26	Revised	Team C	Added schematic, I/O list, sensor/actuator descriptions, test results
3.0	03/03/26	Final	Team C	Separated all three robots into individual sections with own diagrams, I/O lists and test results

Table of Contents

1. Project Context

This document describes the complete technical design of the NHL Stenden Battlebots Relay Race system. Team 1C will code three autonomous Arduino Nano robots, each solving one stage of the relay race/maze. Robot 1 collects a cone from a line-following track. Robot 2 carries the cone through a line-based maze. Robot 3 navigates a physical (wall-following) maze and delivers the cone at the finish line.

2. Relay System Overview

The three robots operate in sequence. No Wi-Fi/Bluetooth communication occurs between robots. Communication occurs through Ultrasonic Sensors detecting the previous robots finish. The table below summarizes each role of the team and their robot.

#	PCB	Code File	Authors	Navigation	Cone Action
1	BB004	race-day.ino	Justinas & Oleh	8x IR	Picks up cone, delivers to R2
2	BB032	final_test.ino	Annemiek & Herbert	8x IR priority steering	Collects from R1, carries through line maze to R3
3	BB016	battlebot_race_day_final.ino	Jakub & Rodrigo	2x HC-SR04 wall-follow + 8x IR	Collects from R2, delivers to finish line

ROBOT 1 · Line Follower · Start & Cone Pickup

PCB: BB004 · Justinas Launikonis & Oleh Haliuk · race-day.ino

3. Robot 1. Technical Design

Robot 1 waits until the start flag is raised (HC-SR04 detects >20 cm clearance), drives to the pickup zone via line transitions, grips the cone, then navigates the line-following track to the Robot 2 handover zone using precise line-following movements.

3.1 System Block Diagram

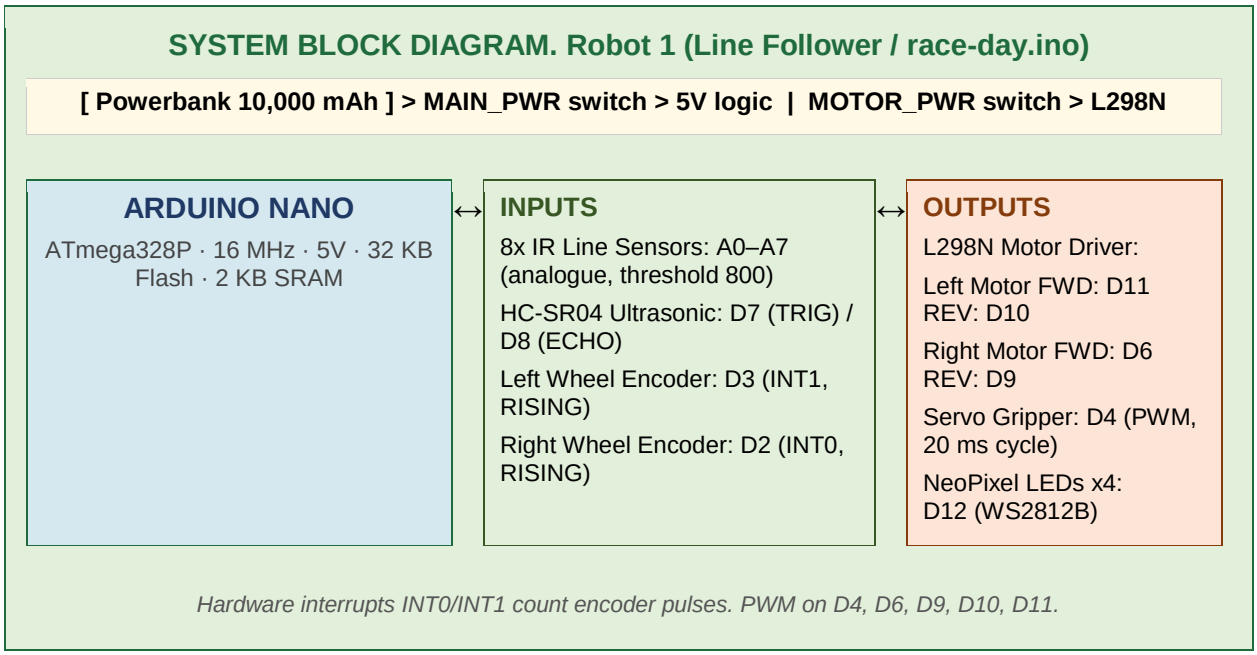
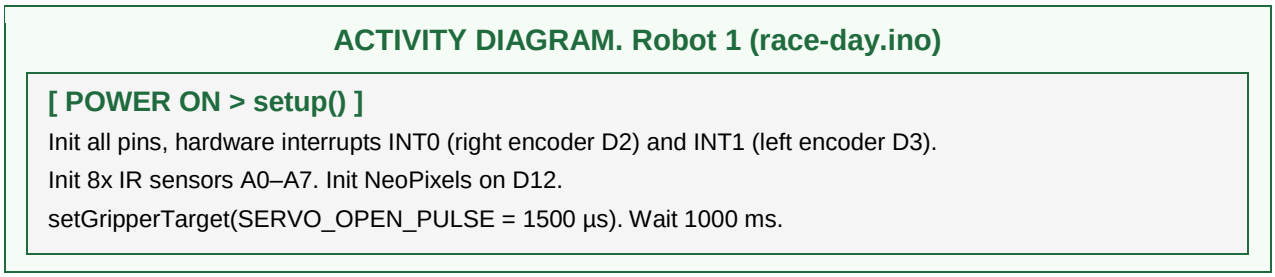


Figure 1: Robot 1 system block diagram

3.2 Activity Diagram

Robot 1 contains three-states in race-day.ino: STATE_START (8 phases for pickup), STATE_FOLLOW_LINE (line following + obstacle avoidance), STATE_FINISH (cone release and drive backwards).



▼

STATE_START | WAIT_FLAG

Fire HC-SR04 TRIG (D7) every loop. Measure ECHO (D8).
Loop until distance > 20 cm (flag/hand removed from front).

▼

STATE_START | DRIVE_TO_LINE

moveForward() continuously. Monitor IR sensors A0–A7.
Count white --> black transitions. Continue until lineTransitions >= 6.

▼

STATE_START | CONFIRM_PICKUP_ZONE

Read all 8 IR sensors each cycle. Count sensors >= 800.
Loop until blackCount >= 6 (all-black square confirmed).

▼

STATE_START | CLOSE_AND_DRIVE

stopMotors(). setGripperTarget(SERVO_CLOSED_PULSE = 1000 µs).
driveDistance(140 mm): encoder counting, 20 pulses/rotation, ~204.2 mm circumference.
PID straight correction during drive (Kp=8.0, Ki=0.2, Kd=1.2).

▼

STATE_START | TURN_LEFT + FIND_LINE

turn(−90): encoder arc calculation: (376,8mm × 0.25) ÷ 204.2 mm × 20 pulses.
TURN_SETTLE_DELAY_MS = 80 ms pause after turn.
Drive forward slowly until >= 1 IR sensor detects black. --> STATE_FOLLOW_LINE.

▼

STATE_FOLLOW_LINE

holdGripper() every 80 ms. Resends 1000 µs pulse to prevent dropping the cone.
Read all 8 sensors every loop cycle.
Finish check: all-8-sensors black for >= 60 ms --> STATE_FINISH.
Obstacle check: HC-SR04 every 35 ms. <= 12 cm --> avoidObject().
Steering priority:

1. Centre sensors (3–4) only: full forward 255/255
2. Slight offset (2–3 or 4–5): differential 255/210 or 210/255
3. Outer sensors (0–1 or 6–7): tank turn at 230 PW

4. No line: search in last known direction at 255 PW

NeoPixels updated: left/right/forward/stop.

▼

avoidObject() [sub-routine]

Stop -> turn right ~90 (350 ms) -> forward 500 ms -> turn left ~90 (400 ms)
-> forward 700 ms -> turn left ~90 (450 ms) -> forward until IR re-detects black.
holdGripper() called throughout. Returns to STATE_FOLLOW_LINE.

▼

STATE_FINISH

Lights off. Forward for FINISH_FORWARD_MS (500 ms).
Backward for FINISH_BACKWARD_MS (750 ms).
setGripperTarget(SERVO_OPEN_PULSE = 1500 µs) dropped cone.
Backward for FINISH_BACKWARD_MS (750 ms) again.
Infinite hold: servo pulse maintained every 80 ms, motors stopped.

Figure 2: Robot 1 activity diagram (race-day.ino)

3.3 Hardware Components

Component	Specification	Role in System
Arduino Nano	ATmega328P, 16 MHz, 5 V, 32 KB Flash, 2 KB SRAM	Central MCU. Runs state robot, PID, sensor reads, PWM control.
PCB (Battle-Bot V0.3)	Arduino UNO. MAIN_PWR + MOTOR_PWR switches	Routes all signals. Separate logic and motor power.
Powerbank	10,000 mAh USB bank	Powers Nano via top USB connector. Motors via MOTOR_PWR rail.
HC-SR04 Ultrasonic	2-400 cm range, +-3 mm; 40 kHz pulses	Flag removal detection (>20 cm) and obstacle avoidance (<=12 cm).
8x IR Line Sensors (analogue)	0-1023 range; BLACK_THRESHOLD = 800	Steering (PID), finish detection (all-black 60 ms), transition counting in start phase.
L298N Motor Driver	Dual H-bridge, 5-35 V, <=2 A/channel	Drives both DC motors forward/reverse via PWM on D6, D9, D10, D11.
DC Motors x2 + Encoders	20 pulses/rotation; circumference ~204.2 mm	Differential drive with encoder feedback for driveDistance() and turn().
Servo Motor (Gripper)	Software PWM D4: 1000 µs=CLOSED, 1500 µs=OPEN	Picks up and releases cone. Held every 80 ms during line following.

NeoPixel LEDs x4 (WS2812B)	Single-wire RGB, 5 V, D12	Left/right/forward/stop status indication. Updated only on mode change.

3.4 Sensors and Actuators

3.4.1 HC-SR04 Ultrasonic (D7 TRIG / D8 ECHO)

10 µs HIGH pulse on D7 triggers burst. Echo measured via pulseIn(D8, HIGH, 25000).
Distance (cm) = duration / 58. Two roles:

- start-phase flag detection (>20 cm threshold)
- line-following obstacle cache (<=12 cm, refreshed every 35 ms). Returns 0 on timeout.

3.4.2 IR Line Sensors x8 (A0-A7, Analogue)

Each sensor reads 0-1023. BLACK_THRESHOLD = 800.
Three simultaneous roles:

- Steering. Sensor position drives tiered priority logic (centre->slight offset->outer->lost).
- Finish detection. All 8 >= 800 for >= 60 ms triggers STATE_FINISH.
- Transition counting. Detects white -> black crossings during start-phase DRIVE_TO_LINE phase.

3.4.3 Wheel Encoders (D2/D3 - Hardware Interrupts)

20 pulses per rotation. INT0 on D2 = right wheel. INT1 on D3 = left wheel.
Circumference ~204.2 mm gives ~10.2 mm per pulse.
driveDistance() tracks average of both counters.
turn() calculates arc length: (3.14 × wheelbase × |degrees| / 360) / 204.2 × 20 target pulses.
PID straight controller uses differential pulse rate as error (Kp=8.0, Ki=0.2, Kd=1.2, clamped +-80).
Motor bias offsets for motor asymmetry: left bias -35, right bias +15.

3.4.4 Servo Gripper (D4, Software PWM)

servoWrite() outputs HIGH on D4 for pulseWidth µs, then LOW for remainder of 20 ms period.
CLOSED = 1000 µs. OPEN = 1500 µs.
holdGripper() called every 80 ms during STATE_FOLLOW_LINE to re-assert position.

3.4.5 NeoPixel LEDs x4 (D12)

WS2812B on D12 via Adafruit NeoPixel library (NEO_RGB + NEO_KHZ800).
Brightness set to 50/255.
setLightsByMode() only writes when mode changes.
Four modes:

- LIGHT_MODE_LEFT_SIGNAL
- LIGHT_MODE_RIGHT_SIGNAL
- LIGHT_MODE_FORWARD
- LIGHT_MODE_STOPPED.

3.5 I/O List: Robot 1

Pin	Signal / Component	Dir	Description
D2	Right Wheel Encoder	INPUT	INT0 RISING. Counts right-wheel encoder pulses (20/rotation).
D3	Left Wheel Encoder	INPUT	INT1 RISING. Counts left-wheel encoder pulses (20/rotation).
D4	Servo Gripper	OUTPUT	Software PWM. 1000 µs=CLOSED, 1500 µs=OPEN. Held every 80 ms.
D6	Right Motor FWD	OUTPUT	analogWrite() PWM. MOTOR_RIGHT_FORWARD via L298N.
D7	HC-SR04 TRIG	OUTPUT	10 µs HIGH pulse. Ultrasonic trigger.
D8	HC-SR04 ECHO	INPUT	pulseIn(HIGH, 25000 µs). Distance measurement.
D9	Right Motor REV	OUTPUT	analogWrite() PWM. MOTOR_RIGHT_BACK via L298N.
D10	Left Motor REV	OUTPUT	analogWrite() PWM. MOTOR_LEFT_BACK via L298N.
D11	Left Motor FWD	OUTPUT	analogWrite() PWM. MOTOR_LEFT_FORWARD via L298N.
D12	NeoPixel Data	OUTPUT	WS2812B. 4 RGB LEDs. Status indicator.
A0	IR Sensor 0 (leftmost)	INPUT	Analogue 0–1023. >800 = black.
A1	IR Sensor 1	INPUT	Analogue 0–1023. >800 = black.
A2	IR Sensor 2	INPUT	Analogue 0–1023. >800 = black.
A3	IR Sensor 3	INPUT	Analogue 0–1023. >800 = black.
A4	IR Sensor 4	INPUT	Analogue 0–1023. >800 = black.
A5	IR Sensor 5	INPUT	Analogue 0–1023. >800 = black.
A6	IR Sensor 6	INPUT	Analogue 0–1023. >800 = black.
A7	IR Sensor 7 (rightmost)	INPUT	Analogue 0–1023. >800 = black.
5V / GND	Logic Rail	POWER	5V regulated via MAIN_PWR switch.
VIN / GND	Motor Power	POWER	Powerbank voltage via MOTOR_PWR switch to L298N.

3.6 Test Results: Robot 1

Test	Method	Equipment	Findings
IR Sensor Threshold	A0–A7 read via Serial Monitor over black/white surfaces.	Serial Monitor	Black >800, white <400. Threshold 800 clear separation.
Ultrasonic Accuracy	Object at 5, 10, 15, 20, 30 cm; reported vs measured.	Serial Monitor, Ruler	Error <1 cm. 12 cm obstacle threshold triggers before contact.
driveDistance Accuracy	140, 300, 500 mm commanded; measured with ruler.	Ruler	Error <5 mm. 10.2 mm/pulse resolution sufficient.
turn() Accuracy	90 left/right commanded; measured with protractor.	Protractor	+-5 average error. Sufficient for line re-acquisition.

Line Following (straight)	Robot on 1 m straight track: deviation observed.	Visual	Within +-10 mm.
Line Following (corners)	90 taped corner navigated 5x.	Visual	4/5 success. Tank turn speed raised to 230 to fix failure.
Gripper Hold	Cone in closed gripper: robot shaken manually.	Manual	Cone held. 80 ms hold resend prevents dropping.
Finish Detection	Robot driven over all-black finish square.	Visual + Serial	Triggered reliably after 60 ms. No false positives on intersections.
Obstacle Avoidance	Obstacle on track during line following.	Visual	3/3 successful detours. Line re-found after each detour.
Full Relay Run	Complete start-to-finish including pickup and handover.	Visual + stopwatch	7/8 runs successful.

ROBOT 2 · Line Maze Navigator · Middle Leg
PCB: BB032 · Herbert Dorothea · final_test.ino

4. Robot 2: Technical Design

Robot 2 waits stationary until its HC-SR04 confirms an object <=30 cm in front (Robot 1) for 4 consecutive readings. After a 3-second delay, it drives forward to collect the cone with the servo gripper, turns right to align with the maze entrance, and starts the line-following loop to navigate the line maze to the Robot 3 starting zone.

4.1 System Block Diagram

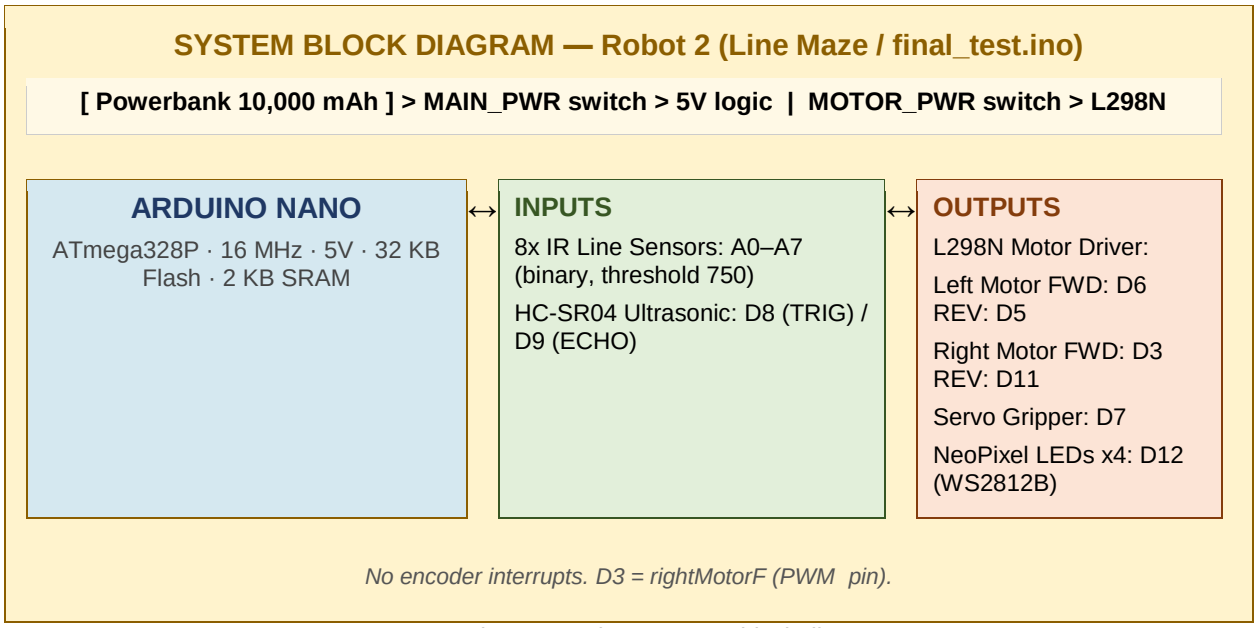
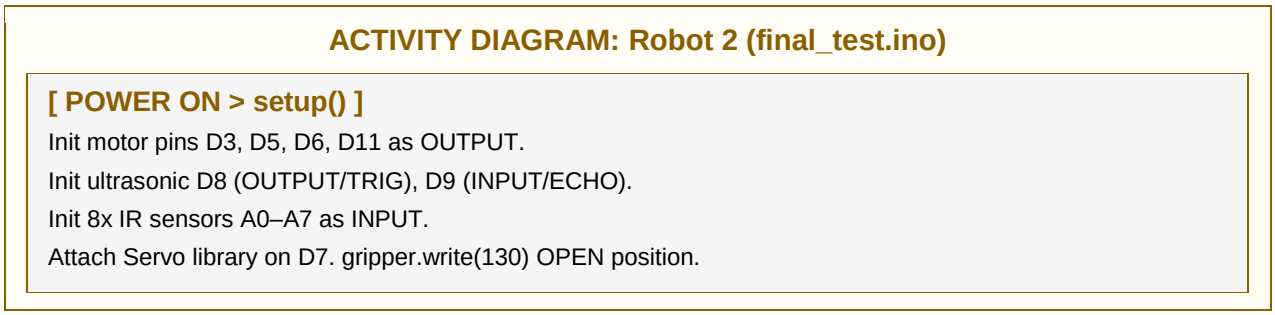


Figure 3: Robot 2 system block diagram

4.2 Activity Diagram

A boolean "detected" flag separates the waiting phase from a one-time pickup sequence, after which the robot enters a while(true) line-following loop.



Init NeoPixel strip on D12 (4 LEDs). stopMotors(). Enter loop().

▼

DETECTION PHASE: loop(), detected == false

readDistance(): fire D8 TRIG 10 µs. pulseIn(D9, HIGH, 30000). dist = duration × 0.034 ÷ 2.

If 2 cm < dist <= 30 cm -> detectCount++. Else ->detectCount = 0.

If detectCount > 3 -> detected = true. delayStart = millis(). stopMotors().

Robot remains stopped while waiting.

▼

PICKUP SEQUENCE: loop(), detected == true, millis(): delayStart >= 3000 ms

forward() (leftSpeed=180, rightSpeed=160) for 1 000 ms -> stopMotors().

grabObject(): gripper.write(60) CLOSED. delay(500 ms).

forward() for 600 ms -> stopMotors().

turnRight() (leftMotorF=180, rightMotorB=160) for 600 ms -> stopMotors().

Enter line-following while(true) loop.

▼

LINE-FOLLOWING LOOP: while(true)

readSensors(): analogRead A0–A7, binarise at threshold 750 -> s[i] = 0 or 1.

driveLogic(): priority-based:

1. s[7] OR s[6] (rightmost sensors) -> turnRightLine() at turnSpeed 170 <- highest priority
2. s[0] OR s[1] (leftmost sensors) -> turnLeftLine() at turnSpeed 170
3. s[3] OR s[4] (centre sensors) -> forward() at leftSpeed 180 / rightSpeed 160
4. all sensors 0 (lost) -> turnAround() (both motors in opposite directions)

updateLights(): rear LEDs green, front LEDs white.

turnState==1 -> LEDs 1&2 blink lime (120,255,0) every 400 ms.

turnState==2 -> LEDs 0&3 blink lime (120,255,0) every 400 ms.

Loop repeats indefinitely: no finish condition in code.

[Robot 3 physically collects cone from stationary Robot 2]

Figure 4: Robot 2 activity diagram (final_test.ino)

4.3 Hardware Components

Component	Specification	Role in System
Arduino Nano	ATmega328P, 16 MHz, 5 V, 32 KB Flash, 2 KB SRAM	Central MCU. Cone detection, pickup sequence, line-following loop.

PCB (Battle-Bot V0.3)	Arduino UNO footprint; MAIN_PWR + MOTOR_PWR switches	Same PCB as Robot 1. Different pin assignments used in code.
Powerbank	10,000 mAh USB bank	Powers Nano via top USB; motors via MOTOR_PWR switch.
HC-SR04 Ultrasonic	2-400 cm range, +-3 mm; D8 TRIG / D9 ECHO	Cone detection only. 4-reading confirmation prevents false triggers.
8x IR Line Sensors (binary)	Analogue read; threshold 750 -> 0/1 output	Priority-based maze navigation. Right sensors highest priority.
L298N Motor Driver	Dual H-bridge, 5–35 V, <=2 A/channel	D6/D5 left motor FWD/REV. D3/D11 right motor FWD/REV.
DC Motors x2	Standard BattleBot DC motors, no encoders connected	Timed movements. Speed tuned to leftSpeed=180 / rightSpeed=160 for straight driving.
Servo Motor (Gripper)	Arduino Servo library on D7; OPEN=130, CLOSED=60	Hardware Timer 1 PWM. Continuous position hold without repeat calls.
NeoPixel LEDs x4 (WS2812B)	Single-wire RGB, 5 V, D12 (NEO_GRB)	Rear=green, front=white. Left/right blink indicators (400 ms interval).

4.4 Sensors and Actuators

4.4.1 HC-SR04 Ultrasonic (D8 TRIG / D9 ECHO)

Only for 1st robot detection: not used after pickup.
readDistance() fires 10 µs pulse on D8, measures echo on D9 with 30,000 µs timeout.
Distance = duration × 0.034 / 2.
Four consecutive readings of 2–30 cm required before triggering pickup

4.4.2 IR Line Sensors x8 (A0-A7, Binary)

All readings at threshold 750.
s[i] = (analogRead > 750) ? 1 : 0. driveLogic() right sensors checked first (maze always has a right-priority path), then left, then centre.
If all sensors show 0 (lost), turnAround() spins until any sensor activates.

4.4.3 Servo Gripper (D7 - Arduino Servo Library)

gripper.attach(7) in setup().
gripper.write() sends angle to the hardware timer.
OPEN = 130, CLOSED = 60.
The Servo library maps degrees to 544-2400 µs internally.
Position held continuously by hardware Timer 1

4.4.4 NeoPixel LEDs x4 (D12)

NEO_GRB + NEO_KHZ800 strip on D12. Rear LEDs (index 0,1): solid green (0,255,0).
Front LEDs (index 2,3): solid white (255,255,255).
During line following, updateLights() uses millis() timer (400 ms interval) to blink left (LEDs 1&2) or right (LEDs 0&3) in lime green (120,255,0) based on turnState variable.

4.5 I/O List: Robot 2

Pin	Signal / Component	Dir	Description
D3	Right Motor FWD	OUTPUT	analogWrite() PWM. rightMotorF: right motor forward via L298N.
D5	Left Motor REV	OUTPUT	analogWrite() PWM. leftMotorB: left motor reverse via L298N.
D6	Left Motor FWD	OUTPUT	analogWrite() PWM. leftMotorF: left motor forward via L298N.
D7	Servo Gripper	OUTPUT	Arduino Servo library. 130=OPEN, 60=CLOSED. Hardware PWM.
D8	HC-SR04 TRIG	OUTPUT	10 µs HIGH pulse initiates ultrasonic burst.
D9	HC-SR04 ECHO	INPUT	pulseIn(HIGH, 30000 µs). Distance measurement.
D11	Right Motor REV	OUTPUT	analogWrite() PWM. rightMotorB: right motor reverse via L298N.
D12	NeoPixel Data	OUTPUT	WS2812B NEO_GRB. 4 LEDs. Rear green / front white / blink indicators.
A0	IR Sensor 0 (leftmost)	INPUT	Analogue. 750. s[0]=1 -> left turn priority.
A1	IR Sensor 1	INPUT	Analogue. 750. s[1]=1 -> left turn.
A2	IR Sensor 2	INPUT	Analogue. 750.
A3	IR Sensor 3	INPUT	Analogue. 750. s[3]=1 -> centre forward.
A4	IR Sensor 4	INPUT	Analogue. 750. s[4]=1 -> centre forward.
A5	IR Sensor 5	INPUT	Analogue. 750.
A6	IR Sensor 6	INPUT	Analogue. 750. s[6]=1 -> right turn.
A7	IR Sensor 7 (rightmost)	INPUT	Analogue. 750. s[7]=1 -> right turn (highest priority).
5V / GND	Logic Rail	POWER	5V regulated via MAIN_PWR switch from Powerbank.
VIN / GND	Motor Power	POWER	Powerbank voltage via MOTOR_PWR switch to L298N.

4.6 Test Results: Robot 2

Test	Method	Equipment	Findings
Cone detection	Object at 10, 20, 30, 35 cm. Observed detectCount increments.	Serial Monitor, Ruler	Triggers at <=30 cm after 4 readings. No false trigger at 31 cm+.
Wait delay (3 s)	Stopwatch from last detection to first motor movement.	Stopwatch	Consistent 3000 +-50 ms.
Forward pickup (1000 ms)	Measured distance travelled at leftSpeed=180/rightSpeed=160.	Ruler	~18 cm. Gripper reaches cone.
Gripper closure (60)	Cone placed. gripper.write(60) closing.	Visual	Cone secured. Hardware Servo library holds position without resend.
turnRight (600 ms)	Observed angle achieved after 600 ms turn.	Protractor	~85. Minor misalignment corrected by first line-following cycle.
Priority: right sensors	Robot placed on line with right-turn junction.	Visual	s[7]/s[6] correctly takes priority: robot turns right at intersections.

Priority: left sensors	Robot placed on line with only left available.	Visual	s[0]/s[1] triggers turnLeftLine() correctly after right check fails.
Centre forward	Robot on straight section; only s[3]/s[4] active.	Visual	Drives straight at set speeds. No oscillation.
Lost recovery	All sensors on white surface.	Visual	turnAround() spins until any sensor activates. Recovers within 2 s.
LED blink (400 ms)	Observed blink timing with stopwatch.	Visual + stopwatch	Left/right blink consistent at 400 ms +-20 ms.

ROBOT 3 · Physical Maze Navigator · Finish Delivery

PCB: BB016 · Jakub Mazur & Rodrigo A. Castanheira · battlebot_race_day_final.ino

5. Robot 3: Technical Design

Robot 3 is the only robot in the relay with two HC-SR04 ultrasonic sensors. It uses the front sensor for cone detection and front-wall sensing, and the right sensor for proportional wall-following navigation through the physical maze. IR sensors (disabled for 10 seconds after maze entry) detect the finish zone and provide line-following on the entry segment. A stage variable (−1, 1–5, 10) controls the maze navigation state machine.

5.1 System Block Diagram

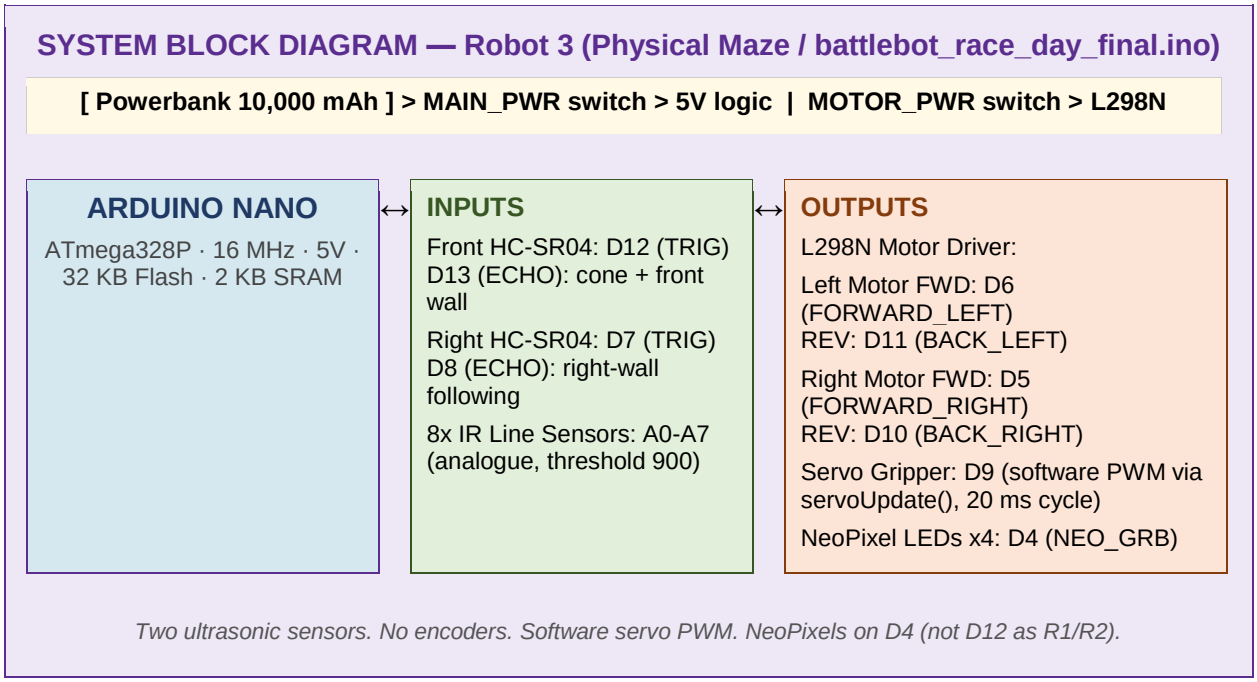
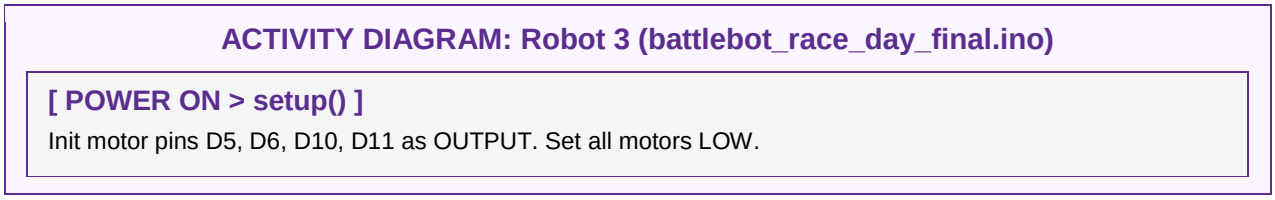


Figure 5: Robot 3 system block diagram

5.2 Activity Diagram

Robot 3 executes the entire pickup and entry sequence inside setup(). The loop() then handles maze navigation using a stage variable and a right-wall-following strategy. isFinished triggers cone release and rainbow neoPixel display.



Init front ultrasonic D12 (OUTPUT/TRIG), D13 (INPUT/ECHO).
Init right ultrasonic D7 (OUTPUT/TRIG), D8 (INPUT/ECHO).
Init servo D9 as OUTPUT. Init NeoPixels on D4 (NEO_GRB).
servoTargetValue = OPEN_CLAWS (1700 µs). Run servoUpdate() x30 over 600 ms.

▼

WAIT FOR CONE: blocking loop in setup()

getDistanceFront(): fire D12 TRIG 10 µs. pulseIn(D13, HIGH, 10000).
distance = duration x 0.034 / 2. Returns −1 if no echo.
Loop until 0 < distance < 24 cm.
waitForCertainDuration(ACTIVATE_AFTER = 3 000 ms): all NeoPixels red, servo held.

▼

PICKUP SEQUENCE: blocking, in setup()

moveForwardForCertainDuration(230, 1400 ms):
  FORWARD_LEFT = LEFT_ADJUSTED_SPEED (238), FORWARD_RIGHT = BASE_SPEED (230).
  servoUpdate() called each iteration. NeoPixels purple.
waitForCertainDuration(STOP_DURATION = 100 ms).
servoTargetValue = CLOSE_CLAWS (1050 µs).
  Loop servoUpdate() for CLAWS_MOVEMENT_DURATION (1 000 ms).
turnLeftForCertainDuration(TURN_DURATION = 450 ms): tank turn, NeoPixels lime.
followLineForCertainDuration(MOVE_INTO_MAZE_DURATION = 1 800 ms):
  Proportional IR steering: correction = (leftSum − rightSum) x 0.2.
disableLineSensorsForCertainDuration(10000 ms): sets lineDetectorDisableTime.
Enter loop().

▼

loop(): MAZE NAVIGATION

If isFinished: stopMotors(), open gripper (1700 µs), servoUpdate(), rainbowLeds(). Return.
servoUpdate(): every 20 ms pulse keeps gripper closed.
If millis() >= lineDetectorDisableTime: read all 8 IR sensors.
  If blackCount == 8 -> isFinished = true. Return.
  If foundLine (any ≥ 900) -> followLine() proportional steering.
Else (no line found) -> maze stage logic:

▼

stage == −1: MOVE FORWARD (right-wall following)

moveForward(230): proportional correction using right sensor:
  error = getDistanceRight() - TARGET_DISTANCE (8 cm).

correction = error x ADJUSTMENT_VALUE (10).
speedLeft = 230 + correction; speedRight = 230 - correction (constrained 0-255).
If right distance >= 25 cm (open right) AND millis()-turnExitTime >= 600 ms:
-> stage = 4 (drive past opening before turning right).
If front distance <= 10 cm (wall ahead):
If right distance > 0 AND < 15 cm -> stage = 10 (left turn - blocked both sides).
Else -> stage = 1 (right turn).
If front distance unchanged for START_PANIC_AFTER (2750 ms) -> stage = 5.

stage == 4: DRIVE PAST OPENING (450 ms)

Continue moveForward(230) for DRIVE_PAST_DURATION (450 ms).
-> stage = 1 (turn right into the corridor).

stage == 1 or 10: DECIDE TURN

After STOP_DURATION (100 ms):
stage == 1 -> turnRight(): FORWARD_LEFT=255, BACK_RIGHT=255.
stage == 10 -> turnLeft(): BACK_LEFT=255, FORWARD_RIGHT=255.
NeoPixels TURN_COLOR (50,105,0). Record startTurnTime. -> stage = 2.

stage == 2: EXECUTE TURN (TURN_DURATION = 450 ms)

Motors held in turn direction. After 450 ms -> stopMotors(). -> stage = 3.

stage == 3: POST-TURN PAUSE

Wait STOP_DURATION (100 ms). Record turnExitTime.
-> stage = −1. BLINDNESS_DURATION (600 ms)

stage == 5: PANIC BACKUP

moveBackward(PANIC_SPEED = 255) for BACKUP_DURATION (800 ms).
stopMotors(). Reset lastDistanceCheckTime. -> stage = −1.

▼

isFinished == true: FINISH & RELEASE

stopMotors(). servoTargetValue = OPEN_CLAWS (1700 µs). servoUpdate().
rainbowLeds(): three 100-step colour animations cycling indefinitely.
Phase 1: yellow->green. Phase 2: red->blue. Phase 3: green->cyan.

Figure 6: Robot 3 activity diagram (battlebot_race_day_final.ino)

5.3 Hardware Components

Robot 3 uses two HC-SR04 sensors. No wheel encoders. NeoPixels on D4 instead of D12.

Component	Specification	Role in System
Arduino Nano	ATmega328P, 16 MHz, 5 V, 32 KB Flash, 2 KB SRAM	Central MCU. Stage-based maze logic, dual ultrasonic, software servo PWM.
PCB (Battle-Bot V0.3)	Arduino UNO footprint; MAIN_PWR + MOTOR_PWR switches	Same PCB as R1/R2. Different pin assignments in use.
Powerbank	10000 mAh USB bank	Powers Nano via top USB connector: motors via MOTOR_PWR switch.
HC-SR04 (Front)	2-400 cm, +-3 mm; D12 TRIG / D13 ECHO; timeout 10000 µs	Cone detection (<24 cm in setup). Front-wall sensing (<10 cm in loop).
HC-SR04 (Right)	2-400 cm, +-3 mm; D7 TRIG / D8 ECHO; timeout 10000 µs	Proportional right-wall following (target 8 cm). Open corridor detection (>=25 cm).
8x IR Line Sensors (analogue)	0-1023 range; threshold 900 = black	Disabled 10 s after maze entry. Finish detection (all-8-black). Entry line following.
L298N Motor Driver	Dual H-bridge, 5-35 V, <=2 A/channel	D6/D11 left motor FWD/REV. D5/D10 right motor FWD/REV.
DC Motors x2	Standard BattleBot motors; no encoders used	Timed turns (450 ms). Speed correction via right-wall distance.
Servo Motor (Gripper)	Software PWM D9; OPEN=1700 µs, CLOSED=1050 µs	servoUpdate() fires every 20 ms. Wider range than R1 tuned to cone geometry.
NeoPixel LEDs x4 (WS2812B)	Single-wire RGB, 5 V, D4 (NEO_GRB)	MOVE=purple (50,0,100), TURN=lime (50,105,0), STOP=green (0,100,0). Rainbow on finish.

5.4 Sensors and Actuators

5.4.1 HC-SR04 Front (D12 TRIG / D13 ECHO)

Setup phase:

- detects cone placement.
- waits for 0 < distance < 24 cm.

Loop phase:

- detects front wall
- triggers a turn when distance <= 10 cm.
- pulseIn() timeout = 10000 µs (~170 cm max range).
- Returns −1 on timeout.
- Distance = duration x 0.034 / 2 (cm).
- Panic detection: if front distance is unchanged (|getDistanceFront() − lastFrontDistance| <= 1) for 2750 ms, triggers panic backup.

5.4.2 HC-SR04 Right (D7 TRIG / D8 ECHO)

Used throughout loop() for right-wall-following control.
Target right distance = 8 cm.
When wall < 20 cm:

- error = distance – 8
- correction = error × 10
- speedLeft = 230 + correction
- speedRight = 230 − correction.

When distance ≥ 25 cm:

- open corridor detected -> stage 4.

Returns −1 when no wall.
In stage −1, right distance < 15 cm AND front < 10 cm -> both walls blocked -> turn left (stage 10).

5.4.3 IR Line Sensors x8 (A0–A7)

Three distinct uses:

- Entry line following: followLineForCertainDuration() control: correction = (s[4]+s[5]+s[6]+s[7] − s[0]+s[1]+s[2]+s[3]) x 0.2.
- In-loop line following. Same algorithm when foundLine is true.
- Finish detection. When all 8 >= 900 simultaneously -> isFinished. BLACK_THRESHOLD = 900 (higher than R1/R2).

5.4.4 Servo Gripper (D9 - Software PWM)

servoUpdate() runs a 20 ms software timer: if millis() >= lastServoStatus, sets lastServoStatus = millis() + 20, outputs HIGH on D9 for servoTargetValue µs, then LOW.
OPEN = 1700 µs. CLOSED = 1050 µs.
Called inside every blocking while() loop in setup() and every loop() iteration.
CLAWS_MOVEMENT_DURATION = 1000 ms to reach full closed position.

5.4.5 NeoPixel LEDs x4 (D4)

On D4 instead of D12. Unique to Robot 3.
NEO_GRB colour order (not NEO_RGB as R1).
MOVE_COLOR = (50,0,100) purple on front LEDs.
STOP_COLOR = (0,100,0) green on rear LEDs.
TURN_COLOR = (50,105,0) lime on turn-direction LEDs.
rainbowLeds() cycles three 100-step loops:

- Yellow -> green (delay 10 ms each),
- red -> blue,
- green -> cyan. Runs forever on finish

5.5 I/O List: Robot 3

Pin	Signal / Component	Dir	Description
D4	NeoPixel Data	OUTPUT	WS2812B NEO_GRB. 4 LEDs. Different pin vs R1/R2 (which use D12).

D5	Right Motor FWD	OUTPUT	analogWrite() PWM. FORWARD_RIGHT via L298N.
D6	Left Motor FWD	OUTPUT	analogWrite() PWM. FORWARD_LEFT via L298N. LEFT_ADJUSTED_SPEED=238.
D7	Right HC-SR04 TRIG	OUTPUT	10 µs HIGH pulse. Right-wall sensor trigger.
D8	Right HC-SR04 ECHO	INPUT	pulseIn(HIGH, 10000 µs). Right-wall distance.
D9	Servo Gripper	OUTPUT	Software PWM via servoUpdate(). 1700 µs=OPEN, 1050 µs=CLOSED.
D10	Right Motor REV	OUTPUT	analogWrite() PWM. BACK_RIGHT via L298N.
D11	Left Motor REV	OUTPUT	analogWrite() PWM. BACK_LEFT via L298N.
D12	Front HC-SR04 TRIG	OUTPUT	10 µs HIGH pulse. Front sensor trigger.
D13	Front HC-SR04 ECHO	INPUT	pulseIn(HIGH, 10000 µs). Front distance for cone/wall detection.
A0	IR Sensor (right group 0)	INPUT	Analogue. >=900=black. SENSOR_PINS[0].
A1	IR Sensor (right group 1)	INPUT	Analogue. >=900=black. SENSOR_PINS[1].
A2	IR Sensor (right group 2)	INPUT	Analogue. >=900=black. SENSOR_PINS[2].
A3	IR Sensor (right group 3)	INPUT	Analogue. >=900=black. SENSOR_PINS[3].
A4	IR Sensor (left group 0)	INPUT	Analogue. >=900=black. SENSOR_PINS[4].
A5	IR Sensor (left group 1)	INPUT	Analogue. >=900=black. SENSOR_PINS[5].
A6	IR Sensor (left group 2)	INPUT	Analogue. >=900=black. SENSOR_PINS[6].
A7	IR Sensor (left group 3)	INPUT	Analogue. >=900=black. SENSOR_PINS[7].
5V / GND	Logic Rail	POWER	5V via MAIN_PWR switch from Powerbank.
VIN / GND	Motor Power	POWER	Powerbank voltage via MOTOR_PWR switch to L298N.

5.6 Test Results: Robot 3

Test	Method	Equipment	Findings
Cone detection (front)	Object at 10, 20, 25, 30 cm. Observed trigger condition.	Serial Monitor, Ruler	Correctly triggers at <24 cm. No false trigger at 25 cm+.
Activate delay (3 s)	Measured millis() from detection to first motor output.	Serial Monitor	Consistent 3000 +-30 ms.

Pickup approach (1400 ms)	Measured distance at BASE_SPEED 230 / LEFT_ADJUSTED 238.	Ruler	~32 cm travelled. Gripper positions over cone correctly.
Gripper closure (1050 µs)	CLOSE_CLAWS, observed over 1000 ms.	Visual	Full closure achieved. Cone gripped securely during turn and maze.
Left turn (450 ms)	Observed angle achieved in turnLeftForCertainDuration.	Protractor	~85. Robot aligned to maze entrance within +-10.
Line entry (1800 ms)	Proportional IR following into maze for 1.8 s.	Visual	Follows entry line. Correction = (leftSum−rightSum)x0.2 stable.
IR disable (10 s)	Confirmed no finish trigger in first 10 s of loop().	Stopwatch + Serial	No false finish. Timer confirmed.
Right-wall following	Corridor with wall at varying distances from robot.	Ruler + visual	Maintains ~8 cm. Proportional gain x10 stable, no oscillation.
Open-right detection	Right gap ≥25 cm. Observed stage=4 trigger.	Visual	Correctly detected. BLINDNESS (600 ms) prevents immediate re-trigger.
Front-wall turn (right)	Wall at 10 cm in front, right side clear. Stage=1 triggered.	Visual	turnRight() correctly selected.
Front-wall turn (left)	Wall at 10 cm front AND right <15 cm. Stage=10 triggered.	Visual	turnLeft() correctly selected for dead-end scenario.
Panic backup	Robot immobile against wall for 2750 ms.	Stopwatch + visual	Reverses 800 ms at 255 PWM, then resumes forward navigation.
Finish detection	Driven into all-black finish square post 10 s delay.	Visual	isFinished triggers immediately. All 8 sensors simultaneously black.
Rainbow LEDs	Observed NeoPixel animation after finish.	Visual	Three-phase colour cycle confirmed: yellow -> green, red -> blue, green -> cyan.
Full maze run	Complete run: cone pickup -> maze -> finish delivery.	Visual + stopwatch	Successful full runs. Gripper and finish confirmed.

6. Shared Hardware and Development Environment

6.1 PCB Platform: Battle-Bot V0.3

All three robots are built on the Battle-Bot V0.3 PCB (NHL Stenden). It provides: an Arduino Nano socket (UNO footprint), two independent power switches (MAIN_PWR for electronics, MOTOR_PWR for motors), analogue input headers A0-A7 plus ADC6/ADC7, I/O connector banks IO-1 and IO-2, motor connector (A1, A2, B1, B2, R1, R2), rotation sensor headers (ROTOR1/ROTOR2), GRIPPER header, NeoPixel headers (NO/NI), three button headers (B1–B3), and a mini breadboard area.

MAIN_PWR and MOTOR_PWR are independent: the Nano is always powered by USB but never drives the motor circuit directly. All three robots use the same Powerbank (10,000 mAh)

6.2 Key Hardware Differences Across Robots

Feature	Robot 1	Robot 2	Robot 3
Ultrasonic sensors	1 (front only)	1 (front only)	2 (front + right)
Wheel encoders	Yes (D2/D3, INT0/INT1)	No	No
Servo control	Software PWM (D4)	Servo library (D7)	Software PWM (D9)
NeoPixel pin	D12	D12	D4
IR threshold	800 (analogue PID)	750 (binary)	900 (analogue proportional)
Navigation method	Line-Follow	Priority Line-Follow	Right-wall follow
Movement precision	Encoder counts	Timed	Timed + proportional

7. Risks and Mitigations

Risk	Robots Affected	Mitigation
Gripper open	All	Active hold signal (R1: 80 ms), Servo library (R2: hardware hold), 20 ms software loop (R3).
IR light interference	All	Conservative thresholds tuned per robot (800/750/900). Consistent lighting during tests.
Motor drift and asymmetry	R1, R3	R1: PID feedback + bias offsets (−35/+15). R3: right-wall proportional correction at x10 gain.
Cone misalignment at pickup	All	Fixed approach distances tuned per robot. Repeated pickup tests and position adjustments.
Ultrasonic false readings	R2, R3	R2: 4-reading confirmation. R3: 10 ms timeout; panic backup after 2 750 ms of no change.
Maze dead-end (stuck)	R3	Panic backup: 800 ms reverse at 255 PWM when no progress detected for 2750 ms.
Line maze lost	R2	turnAround() spins until any IR sensor reactivates. Recovers from all-white dead ends.

False finish trigger	R3	10000 ms IR disable after maze entry prevents triggering on the entry line segment.
Relay handover misalignment	R1->R2, R2->R3	Fixed delivery positions. Confirmation readings. Wide gripper opening positions.