

Tennis Ball Collector

ECE 445 Design Document - Fall 2026

Project # 13

Mohammed Alnasser, Yousef Khojah, Zeyad Alruwaily

Professor: Cunjiang Yu

TA: Junyan Bai

Contents

1	Introduction	3
1.1	Problem	3
1.2	Solution	3
1.3	Visual Aid	5
1.4	High Level Requirements	6
2	Design	7
2.1	Physical Design	7
2.2	Block Diagram	9
2.3	Functional Overview & Block Diagram Requirements	10
2.3.1	Vision & Navigation Subsystem	10
2.3.2	Power, Drive & Communication Subsystem	12
2.3.3	Intake & Storage Subsystem	13
2.4	Hardware Design	15
2.4.1	Operating Voltage & Regulation	15
2.4.2	Motor Drive, Enable, and Fault Interfaces	16
2.4.3	Battery Protection & Monitoring	16
2.4.4	Ball Counting & Hopper Sensing	17
2.5	Software Design	18
2.5.1	Navigation & Motor Control	18
2.5.2	Odometry & Encoder Sensing	21
2.5.3	Pixy Communication	22
2.5.4	Wireless Communication	23
2.6	Commercial Component Selection	24
2.6.1	Drive & Intake Motors	24
2.6.2	Vision Module	25
2.6.3	Range Sensors	25
2.6.4	Battery	26
2.7	Tolerance Analysis	26
2.7.1	Intake Blade Kinematics	26
2.7.2	Obstacle Stopping Distance	28
2.8	Cost Analysis	29
2.9	Schedule	31
2.10	Risk Analysis	31
3	Societal Impact, Engineering Standards, Ethics, and Safety	34
3.1	Societal Impact	34
3.2	Engineering Standards	34
3.3	Ethics	34
3.4	Electrical and Mechanical Safety	35
3.5	Safety Mitigation Procedure	35

1 Introduction

1.1 Problem

Tennis practice sessions spend a surprising fraction of their time collecting balls rather than hitting them. During feeding drills a coach or player may use several dozen balls before every one of them is lying somewhere on the court, at which point play stops and someone walks the court picking them up. This is especially inconvenient when a coach is running a practice alone, managing several players, or working across more than one court. Manual ball hoppers make carrying and picking up balls easier, but they still require a person to push them around and collect every ball individually. Autonomous ball-collecting robots do exist commercially; for example, Tennibot currently lists its autonomous Rover collector at \$2,275 [17]. This places commercial autonomous collection well above the target cost of a course-built prototype and motivates a lower-cost design for individual players, schools and small clubs.

An autonomous collector introduces several engineering challenges that must all work together. The robot must reliably identify tennis balls, cover a defined area without missing regions, avoid obstacles, remain inside the court boundary, physically lift balls into storage, and run for a useful amount of time on a portable battery. If any of these functions is unreliable, operating the robot becomes more effort than collecting the balls by hand. Because the robot contains drive motors, a rotating intake, a lithium-polymer battery and autonomous navigation, a failure could also lead to a collision with a person or equipment, or to an overheating battery. Safety therefore has to be part of the design rather than an optional feature.

1.2 Solution

To address this problem we will build a compact, battery-powered robot that searches a defined area for tennis balls, approaches each detected ball, collects it through a motorized intake, and stores it in an onboard hopper. The prototype will be evaluated in a 6 m $\times$ 6 m indoor test area bounded by white tape and containing standard yellow tennis balls under

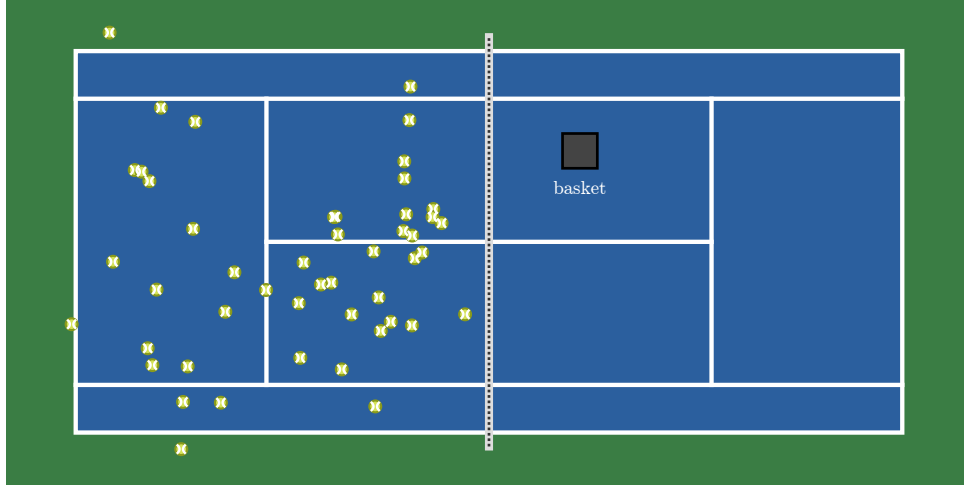


Figure 1: Balls left scattered across a court after a feeding drill (illustration)

normal room lighting.

The robot is divided into three subsystems: Vision and Navigation; Power, Drive and Communication; and Intake and Storage. A Pixy (CMUcam5) smart camera detects tennis balls on board and sends only object coordinates to an STM32F405 microcontroller, which serves as the primary controller. An MPU-6050 IMU, encoders on the 107 rpm drive motors, downward-facing Pololu reflectance sensors and forward range sensors give the controller information about the robot’s motion and surroundings. The robot normally follows a systematic back-and-forth search pattern; when the camera detects a ball, the controller interrupts the pattern, aligns the robot with the ball, collects it, and then resumes searching where it left off.

Two independently controlled rear wheels provide differential steering. At the front of the robot, a horizontal drum containing two long curved blades rotates at approximately 10 rpm and pushes balls up a fixed shallow ramp into a low, rear-sloping hopper that holds 10 balls. An infrared beam-break sensor counts balls as they enter the hopper, and a separate proximity sensor confirms when the hopper is full. An ESP8266 Wi-Fi module hosts a web page through which the user starts and stops the robot and views its battery level, ball count and state.

The project does not claim that autonomous tennis-ball collection is itself novel. Its improvement is a low-cost, course-buildable architecture that combines systematic cover-

age, local safety sensing, encoder-based motion control, and a direct-drive two-blade intake around a reusable plastic-basket chassis. The engineering contribution is therefore the integration and quantitative validation of navigation, custom power/control electronics, and a mechanically simple collection mechanism rather than a new object-detection algorithm.

1.3 Visual Aid

Figure 2 shows how the finished robot is used. The operator starts or stops the robot from a nearby phone or laptop, while the robot autonomously moves through the taped test area, detects balls, and stores them onboard. The wireless link is supervisory only; local sensing and motor shutdown remain on the robot.

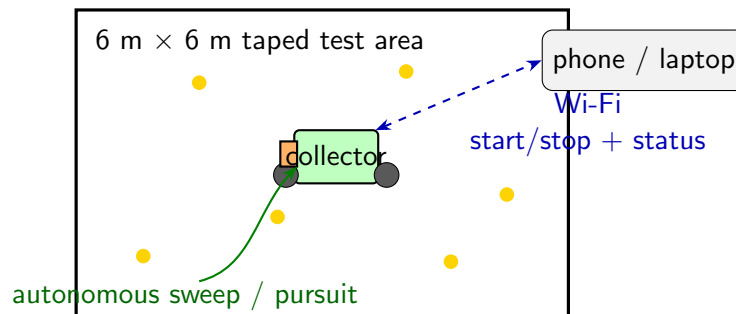


Figure 2: Use-context visual aid: the operator supervises the autonomous collector from outside the taped test area.

Figure 3 illustrates the intended ball path through the intake. A ball in front of the robot is swept by the rotating curved blades (orange) over the fixed shallow ramp (teal) and into the storage area. The final prototype uses a plastic basket as its main body, with the drive hardware supported below it and the intake attached at the front. The concept figure is therefore used to show intake operation rather than the final chassis construction.

On a typical run, the user places the robot in a corner of the taped area, opens the robot’s web page on a phone or laptop, and presses Start. The robot sweeps the area in lanes, diverts to pick up each ball it sees, and stops collecting once 10 balls are aboard, returning to its starting corner. At any time the user can press Stop on the web page, and the robot stops automatically if an obstacle or court boundary is detected or if it loses its communication link.

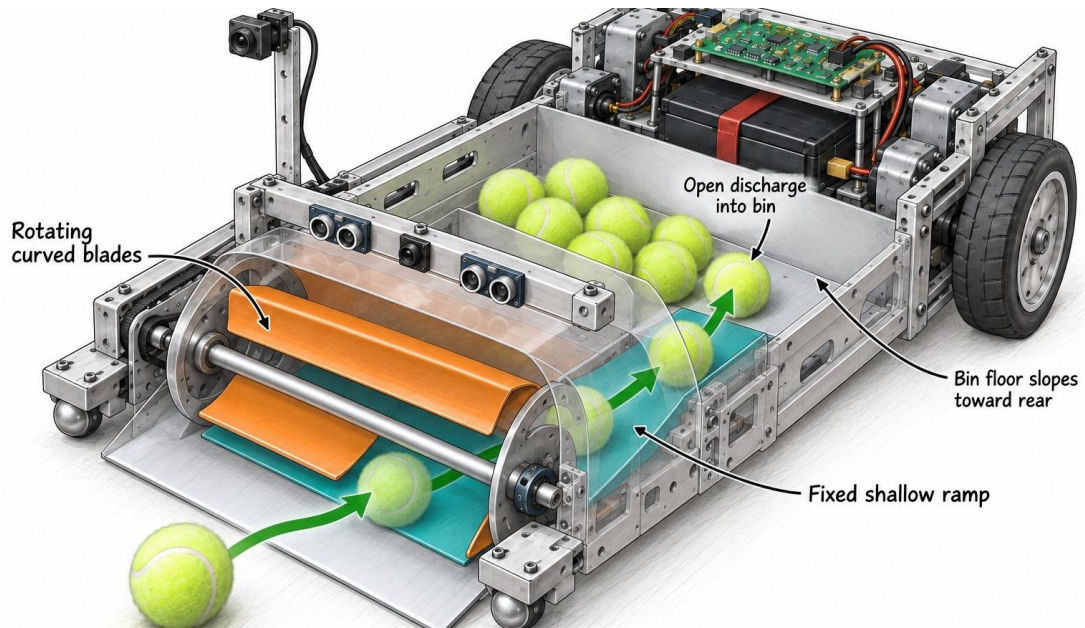


Figure 3: Conceptual intake cutaway and ball path. The final chassis uses the basket-centered construction shown in Figure 4.

1.4 High Level Requirements

To consider our project successful, the robot must fulfill the following:

- **Autonomous Collection:** Starting with an empty hopper, the robot shall collect and retain at least 8 of 10 stationary tennis balls within 10 minutes of the web start command, without human assistance, in at least 2 of 3 test layouts.
- **Obstacle Avoidance:** When approaching a stationary $30\text{ cm} \times 30\text{ cm} \times 30\text{ cm}$ obstacle at 0.20 m/s , the robot shall stop or steer around the obstacle without any part of the robot making contact in 5 of 5 trials.
- **Battery Operation:** On one fully charged battery, the robot shall operate for at least 15 minutes without recharging or experiencing an uncommanded controller reset while driving, operating the intake, and maintaining its Wi-Fi connection.

2 Design

2.1 Physical Design

The prototype is built around a rigid plastic basket that serves as the main body and the central mounting reference for the robot. Rather than constructing the entire chassis from aluminium plate, two aluminium extrusion rails are mounted beneath the basket to form the lower drivetrain structure. The two 12 V encoder gearmotors attach to these rails with metal brackets, and the drive wheels mount outside the basket so that the wheel loads are carried by the aluminium structure instead of the plastic wall. The battery, PCB and wiring are arranged within or immediately around the basket footprint to keep the mass concentrated near the centre of the robot. Figure 4 shows the basket, wheels, gearmotors, caster hardware and aluminium extrusion being used for the prototype.

The intake remains one horizontal drum spanning the front of the robot. Two continuous curved blades, approximately 300 mm long and mounted 180° apart on a common shaft, sweep an approximately 180 mm diameter envelope. A partial hood covers the upper part of the drum while leaving the lower front inlet and rear discharge open. A shallow fixed ramp beneath and behind the drum guides the ball into the basket. The blade profile and drum height remain adjustable during prototype testing so that the blades contact the ball without striking the floor or ramp.

The 10 rpm intake gearmotor is mounted directly on the drum axis and drives the drum shaft through a direct coupling. There is no belt, pulley or secondary reduction stage. This removes belt tensioning and alignment from the mechanism and makes the measured encoder speed directly equal to the drum speed. The drum shaft is still supported mechanically so that radial loads are carried by the shaft supports rather than by the motor bearings alone.

The Pixy camera and forward range sensors are mounted on brackets attached to the basket or its supporting structure with an unobstructed view ahead of the intake. The two downward-facing reflectance sensors are mounted near the front underside so they can detect the white boundary tape before the drive wheels cross it. Exact bracket spacing and hole locations are transferred from the physical basket, motors and sensor modules during

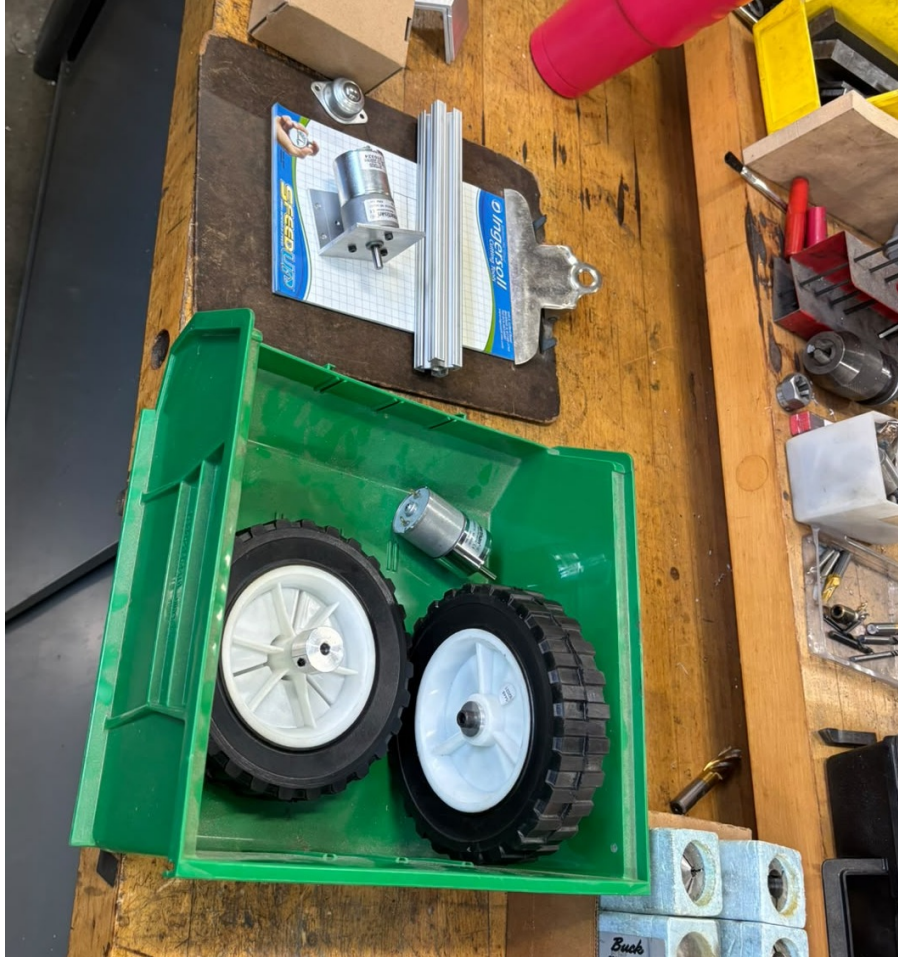


Figure 4: Major prototype hardware. The plastic basket is the main body; aluminium extrusion beneath it supports the drive motors and wheels.

assembly rather than being based on the superseded all-aluminium chassis dimensions.

Table 1: Physical design quantities fixed before assembly

Feature	Design value
Main body	Existing rigid plastic basket; hole locations transferred from hardware
Lower drivetrain	Two aluminium extrusion rails beneath basket
Drive wheels	2, approximately 90 mm diameter
Intake active width	approximately 300 mm
Intake swept diameter	approximately 180 mm
Intake blades	2, 180° apart on one horizontal shaft
Intake drive	12 V, 10 rpm gearmotor; direct coaxial coupling
Storage target	10 standard tennis balls

2.2 Block Diagram

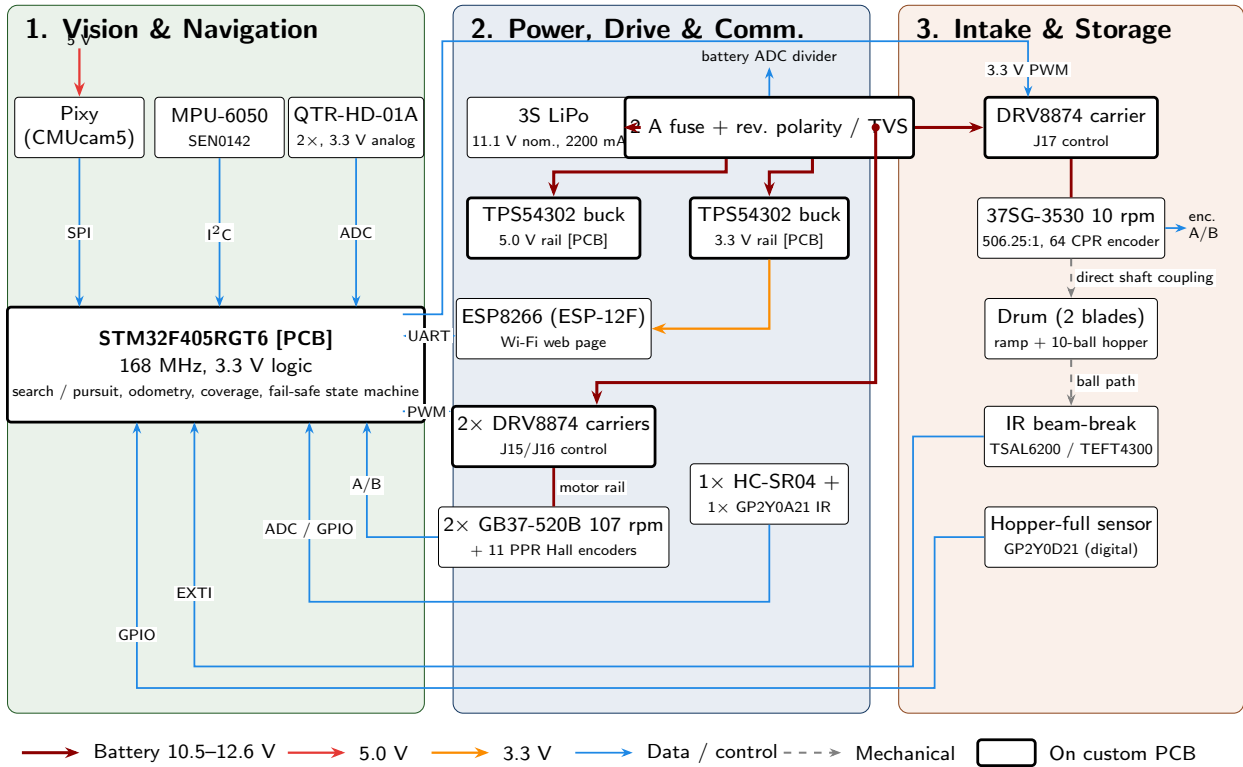


Figure 5: Tennis Ball Collector block diagram. The 5 V rail supplies the Pixy and 5 V range/IR sensors; encoder and boundary interfaces are referenced to 3.3 V. The intake encoder A/B lines go to an STM32 timer.

Figure 5 divides the robot into three independently testable subsystems: Vision & Navigation, Power/Drive/Communication, and Ball Intake & Storage. The STM32 is the

coordinating controller. Sensor data enter through SPI, I²C, ADC, and timer/encoder interfaces; the controller commands the external wheel and intake motor drivers through PWM/direction/enable signals and exchanges supervisory status with the Wi-Fi module over UART. Electrical power is distributed from the 3S battery through separate protected logic and motor branches, while the mechanical interface between the intake subsystem and storage is the directly driven two-blade drum feeding the basket.

2.3 Functional Overview & Block Diagram Requirements

2.3.1 Vision & Navigation Subsystem

The Vision and Navigation subsystem is responsible for locating tennis balls, estimating the robot's motion, detecting the boundary, and generating motion commands. It consists of the Pixy (CMUcam5) camera, the STM32F405 microcontroller, the MPU-6050 IMU, the two Pololu QTR-HD-01A reflectance sensors used for boundary detection, and the firmware that runs the search, pursuit, odometry and coverage functions. The Pixy performs colour-blob detection on board at 50 frames per second and sends the position and size of each detected ball to the STM32 over SPI, so the microcontroller never has to store an image frame. The IMU (a DFRobot SEN0142 MPU-6050 module) communicates over I²C, the boundary sensors are read by the STM32 ADC, and the wheel encoders return quadrature A/B signals from the drive system.

During normal operation the robot follows a back-and-forth search pattern across the test area. Once a ball is detected, the controller temporarily leaves the search path and steers toward the ball using its position in the camera image, slows down near the ball, and hands control to the intake. After collection it returns to the point where it left the search path. The boundary sensors provide an immediate stop when the robot reaches the tape, independent of what the camera reports. For more information on the software design of this subsystem, refer to Section 2.5.1. The requirements and verification table for this subsystem is below. Verification tests use the 6 m $\times$ 6 m taped test area.

Table 2: Vision & Navigation Subsystem – Requirements & Verification

Requirements	Verification
For a stationary tennis ball initially 1.0 m in front of the robot and within $\pm 20^\circ$ of the camera centreline, the subsystem shall guide the robot until the ball is 20 ± 5 cm ahead of the intake lip and within ± 5 cm of the intake centreline in at least 7 of 10 approaches.	- Mark ten ball positions on the floor, each 1.0 m from the intake lip, at angles of -20°, -10°, 0°, $+10^\circ$ and $+20^\circ$ (two trials each). - Load a firmware build that stops both wheels and the drum when the state machine enters COLLECT, so the final position is measured independently of the intake. - For each trial, start the robot in PURSUIT. After it stops, measure the distance from the intake lip to the ball centre and the lateral offset from the intake centreline with a tape measure, and confirm with overhead video. - Confirm that at least 7 of 10 trials are within 20 ± 5 cm and ± 5 cm.
The vision-processing interface implemented by the team shall deliver at least 20 checksum-valid tennis-ball object blocks per second to the STM32F405 while a ball is in view.	- Place a ball 1.0 m in front of the stationary robot. Log valid (checksum-correct) object blocks received by the STM32 over the debug UART for 10 s and confirm at least 200 were received. - Move the ball slowly from the left to the right edge of the image and confirm the reported x coordinate increases monotonically from near 0 to near 319.
The team-designed heading estimator, using IMU and encoder data, shall end within $\pm 5^\circ$ of the measured physical heading after a commanded 360° in-place rotation.	- Read the MPU-6050 WHO_AM_I register over I²C and confirm the value 0x68. - Mark the robot’s heading on the floor, command a 360° in-place rotation, and compare the final physical heading with the mark and the logged estimate. Confirm the error is within $\pm 5^\circ$ in 5 of 5 trials.
Wheel-encoder signals shall provide A/B quadrature information so that the odometry estimates a 2.00 m straight run within $\pm 3\%$ and reports the correct direction of travel.	- Lay a 2.00 m tape line on the floor. Drive the robot forward along it at 0.20 m/s and record the odometry distance from the debug log. Confirm 1.94–2.06 m. - Repeat while driving in reverse and confirm the reported distance is negative and within the same tolerance.
The navigation software shall transition between search, pursuit and collection without human control after the start command.	- Place one ball 2 m off the planned sweep path. Press Start on the web page and do not touch the robot. - From the state log and video, confirm the sequence SEARCH $\rightarrow$ PURSUIT $\rightarrow$ COLLECT $\rightarrow$ SEARCH occurs and that searching resumes from the point where the sweep was interrupted.
Boundary detection shall prevent both drive wheels from crossing the outside edge of the white-tape boundary during an autonomous collection run.	- Run three full autonomous collection runs with balls placed within 30 cm of the tape. - Review overhead video of every approach to the boundary and confirm that neither drive wheel crosses the outside edge of the tape in any run.

2.3.2 Power, Drive & Communication Subsystem

The Power, Drive and Communication subsystem supplies regulated logic power, commands robot motion, detects obstacles and provides the wireless user interface. Most of the custom PCB belongs to this subsystem. The Rev A controller board accepts the protected 3S LiPo logic branch (10.5–12.6 V) through an external 2 A fuse and on-board series reverse-polarity diode/TVS protection. Two independent TPS54302 buck converters generate the 5 V and 3.3 V rails. Motor power uses a separate fused battery branch and physical disconnect so motor return current does not flow through the controller board. The PCB monitors pack voltage with a calibrated resistor divider to an STM32 ADC input.

Three Pololu DRV8874 motor-driver carriers connect to J15–J17 through phase/PWM, hardware-enable, fault and conditioned current-feedback signals; two stages drive the GB37-520B 107 rpm wheel motors and one drives the 10 rpm intake motor. Each carrier accepts 4.5–37 V, is rated for 2.1 A continuous output, provides current-sense feedback, and has adjustable current limiting [16]. The limit is finalized during bring-up after measuring loaded and brief startup current. One HC-SR04 ultrasonic sensor and one Sharp GP2Y0A21 IR sensor provide forward obstacle detection. The ESP8266 communicates with the STM32 over UART and hosts a web page for start/stop control and status. For the hardware design of this subsystem, refer to Section 2.4.

Table 3: Power, Drive & Communication Subsystem – Requirements & Verification Pt. 1

Requirements	Verification
For an input voltage between 10.5 V and 12.6 V, the PCB shall simultaneously provide a regulated $5.0\text{ V} \pm 0.25\text{ V}$ rail at 1.0 A and a $3.3\text{ V} \pm 0.165\text{ V}$ rail at 0.50 A continuously for at least 15 minutes.	- Connect a bench supply to the battery input set to 10.5 V. Connect independent electronic loads to the 5 V rail at 1.0 A and the 3.3 V rail at 0.50 A simultaneously. - Log both rail voltages with a DMM every 30 s for 15 minutes and confirm all readings are within tolerance. - Repeat at 12.6 V.
The separated logic and motor power branches shall permit simultaneous wheel drive, intake operation and Wi-Fi traffic without an uncommanded STM32 reset.	- Probe the 3.3 V rail with an oscilloscope in single-shot mode triggered below 3.1 V. Start both wheels and the intake from rest at full command while the ESP8266 transmits. - Confirm the rail does not fall below 3.0 V, and that the STM32 reset-cause register and the uptime counter on the web page show no reset after 20 start/stop cycles.

Requirements	Verification
MCU-side digital interfaces shall remain within the STM32 input limits; every external signal presented to a non-5-V-tolerant input shall remain between 0 V and 3.6 V.	- With all sensors powered and active, measure the high level of every signal at the STM32 pin (ECHO, encoder A/B lines, hopper inputs and QTR-HD-01A analog outputs) with an oscilloscope. - Confirm every high level is between 2.3 V and 3.6 V.
The two wheel-driver channels shall accept independent PWM control from the STM32 so that the robot can drive forward and perform differential turns.	- Command 40% duty on the left wheel and 80% on the right and confirm from the encoders that the right wheel turns faster and the robot curves left. - Command equal and opposite duty cycles and confirm the robot rotates in place.

Table 4: Power, Drive & Communication Subsystem – Requirements & Verification Pt. 2

Requirements	Verification
The obstacle-detection system shall allow the robot to stop or steer around a $30 \times 30 \times 30$ cm stationary obstacle while travelling at 0.20 m/s without physical contact in 5 of 5 trials.	- Place a 30 cm cardboard cube 2 m ahead of the robot, centred on its path. Drive the robot toward it at 0.20 m/s with obstacle avoidance enabled. - Confirm from video and a strip of paper on the cube face that no contact occurs. Repeat for five trials, including centered and moderately offset obstacle positions within the forward sensor coverage.
The wireless supervisory interface implemented by the team shall maintain status communication for 15 minutes and execute user start/stop commands within 2 s.	- Connect a phone to the robot’s web page and run the robot for 15 minutes. Confirm the page updates the status for the whole test. - Press Start and Stop five times each while recording video of both the phone and the robot. Measure the delay between the button press and the robot’s response and confirm each is under 2 s.
The battery-monitoring circuit shall report protected-pack voltage to the STM32 within $\pm 2\%$ over 10.5–12.6 V and command a software stop at or below 10.5 V.	- Supply the protected battery input from a bench supply at 10.5 V, 11.1 V and 12.6 V. Compare the voltage shown on the web page with a calibrated DMM and confirm agreement within $\pm 2\%$. - Slowly reduce the supply and confirm that the controller disables motor commands when the reported pack voltage reaches 10.5 V or below.

2.3.3 Intake & Storage Subsystem

The Intake and Storage subsystem physically transfers tennis balls from the floor into the robot and keeps track of them. A Pololu DRV8874 motor-driver carrier connected through J17 drives an E-S 37SG-3530 gearmotor (506.25:1, 10 rpm) with a 64 CPR magnetic encoder. The motor is mounted coaxially with the drum and drives the drum shaft directly, so no

belt or pulley stage is required. The encoder therefore measures the drum speed directly. As the robot creeps up to a ball, a blade sweeps down behind it and pushes it over the fixed ramp into the rear-sloping hopper. An IR beam-break (TSAL6200 emitter and TEFT4300 phototransistor) spans the bin entrance and produces one pulse per ball, which the STM32 uses to maintain the ball count. A GP2Y0D21 digital proximity sensor watches the front row of the bin, which only fills when the hopper is at capacity, and provides an independent full signal. For the circuit design of the sensing, refer to Section 2.4.4.

Table 5: Intake & Storage Subsystem – Requirements & Verification

Requirements	Verification
At a robot approach speed of 0.03 m/s, the drum and fixed ramp shall transfer a stationary tennis ball whose centre is within ± 5 cm of the intake centreline completely into the hopper in at least 8 of 10 trials.	- Place a ball on a marked line in front of the robot at offsets of -5, -2.5, 0, $+2.5$ and $+5$ cm (two trials each). - Drive straight at 0.03 m/s with the drum running and record whether the ball ends up fully inside the hopper. Confirm at least 8 of 10 successes.
After entering the hopper, a ball shall remain inside for at least 5 seconds during each successful collection trial.	During each successful trial above, keep the drum running and the robot driving for 5 s after the ball enters, and confirm from video that the ball does not leave the hopper.
The intake drum shall rotate at 10 ± 1.5 rpm during collection.	- With a fully charged battery, mark one blade with tape. With the robot on the floor and the drum running, time 5 full revolutions with a stopwatch (or video) and confirm a period of 26–35 s, which corresponds to 8.5–11.5 rpm. - Confirm that the drum speed computed from the intake encoder agrees with the timed value within 0.5 rpm.
The hopper shall physically accommodate 10 standard tennis balls without preventing the intake from operating.	Place 9 balls in the hopper, then collect a 10th through the intake. Confirm all 10 remain inside and the drum rotates freely without contacting a ball.
The passage sensor shall provide a detectable signal to the STM32 each time a ball enters the hopper so that the ball count is correct for 10 of 10 single-ball entries.	Collect 10 balls one at a time and record the count shown on the web page after each. Confirm it matches the number of balls in the hopper after every entry.
The full-hopper sensing shall provide a digital status signal that allows the STM32 to stop collection within 5 s of the hopper reaching capacity.	- With 9 balls in the hopper, collect a 10th ball. Using video with a timestamp, confirm the drum stops within 5 s of the ball entering, in 5 of 5 trials. - Repeat with the beam-break disconnected and confirm the proximity sensor alone stops the drum.

2.4 Hardware Design

2.4.1 Operating Voltage & Regulation

The robot uses a 3S LiPo whose protected operating range is 10.5–12.6 V. The Rev A controller board is a 98 mm $\times$ 98 mm, two-layer PCB. Its logic branch is protected by an external 2 A fuse mounted near the battery, a series 1N5822 reverse-polarity diode and an SMBJ15A transient suppressor. Motor power is routed on a separate fused branch with a physical disconnect, while controller and motor grounds join at the distribution point so high motor-return current does not flow through sensor wiring or the logic PCB.

Two independent TPS54302 synchronous buck converters generate the regulated rails. U2 produces nominally 5.077 V for the Pixy and 5 V sensors, while U3 produces nominally 3.293 V for the STM32, ESP8266 and 3.3 V logic. The PCB validation targets are 5 V at 1.0 A and 3.3 V at 0.50 A simultaneously; these are board-level verification targets rather than claims that the converter IC headline current can be delivered indefinitely. The power stages use the component values from the released KiCad Rev A BOM: SRP1038A-100M and SRP1038A-6R8M inductors, 22 μ F output capacitors, and 0.1% feedback resistors. JP1 and JP2 allow each regulator output to be isolated for bring-up and external-rail testing.

Table 6: Rev A regulated-rail design targets

Rail	Nominal design output	Continuous verification load
5 V	5.077 V	1.0 A
3.3 V	3.293 V	0.50 A

All interfaces are checked at the STM32 pins during bring-up. The camera SPI and controller logic use 3.3 V. The HC-SR04 trigger path is level-shifted to 5 V by the AHCT logic on the PCB, while its echo path is conditioned before the STM32 timer input. The QTR-HD-01A boundary sensors are powered from 3.3 V and feed ADC channels. The GP2Y0A21 analog input is scaled by the board’s 10 k Ω /10 k Ω divider so a nominal 0–5 V sensor signal maps to 0–2.5 V at the ADC.

2.4.2 Motor Drive, Enable, and Fault Interfaces

The three motor power stages are external to the Rev A controller PCB. Rev A uses three Pololu DRV8874 single-channel brushed-DC motor-driver carriers (item #4035), one per motor. The carrier operates from 4.5–37 V, is rated for 2.1 A continuous output and approximately 4.4 A peak/current-limit operation, and exposes phase/enable control, fault status and analog current sensing [16]. J15, J16 and J17 carry ground, PWM/phase, active-high hardware-enable, active-low fault and conditioned 0–3.3 V current-feedback signals. The current-limit setting is adjusted during bring-up after measuring each motor’s loaded and brief startup current; this protects the motor and driver without changing the controller PCB.

A common permission signal prevents motion unless both firmware arm and the normally-closed stop loop are valid: $\text{HW_ENABLE} = \text{MOTOR_ARM} \wedge \text{STOP_OK}$. External pull-downs keep the driver adapters disabled during reset and connector unplugging. Firmware uses a watchdog and requires a deliberate restart after a fault. The drive PWM signals share a timer period, allowing synchronized left/right control, while each direction and fault channel remains independent. During bring-up, one driver is tested at a time with wheels raised before any loaded test.

2.4.3 Battery Protection & Monitoring

The LiPo pack uses an external BMS/balance charger and must never rely on converter-enable thresholds for cell protection. The controller measures the protected pack with a 100 k Ω /10 k Ω divider, so the ADC node is approximately $V_{BAT}/11$. At 12.6 V the ADC sees about 1.145 V, safely below the 3.3 V ADC rail. Firmware calibrates the divider and series-diode drop against a DMM, warns at 10.8 V, and disables motor commands at 10.5 V.

For runtime, a conservative planning estimate uses 0.5 A average for each wheel motor, 0.48 A for the intake and 0.35 A equivalent battery current for control electronics, for about

1.83 A total. Using 80% of the 2.2 Ah nominal pack gives

$$t_{run} = \frac{0.8(2.2 \text{ Ah})}{1.83 \text{ A}} \approx 58 \text{ min}, \quad (1)$$

so the 15-minute high-level requirement has substantial estimated margin. The formal verification still uses a timed loaded run because motor current and usable battery capacity must be measured on the assembled robot.

2.4.4 Ball Counting & Hopper Sensing

The ball counter is an infrared beam that spans the full 300 mm width of the bin entrance at ball-centre height (about 33 mm above the bin floor). A TSAL6200 940 nm emitter is driven at 50 mA through a 75Ω resistor, $(5 - 1.35 \text{ V})/50 \text{ mA} \approx 73 \Omega$, which dissipates 0.19 W, so a 0.5 W part is used. On the opposite bin wall, a TEFT4300 phototransistor mounted in a short shroud to block room light is pulled up to 3.3 V through 10 k Ω . A 74LVC1G17 Schmitt-trigger buffer cleans the edge before it reaches an STM32 external interrupt (Figure 6a). A ball rolling into the bin blocks the beam for a few hundred milliseconds. The firmware counts a ball when the beam has been blocked for at least 20 ms, and re-arms the counter after the beam has been clear for 100 ms.

Because two balls entering side by side could block the beam as a single event, the count alone does not decide when the hopper is full. A GP2Y0D21YK0F digital proximity sensor is mounted on the left bin wall at the front row, aimed across the bin. The opposite wall at 300 mm is beyond its fixed detection distance of about 24 cm [9], so its output only asserts when balls occupy the front row. Since balls settle toward the rear, this happens only when the hopper is at capacity. The STM32 stops collecting when either the count reaches 10 or the proximity sensor asserts.

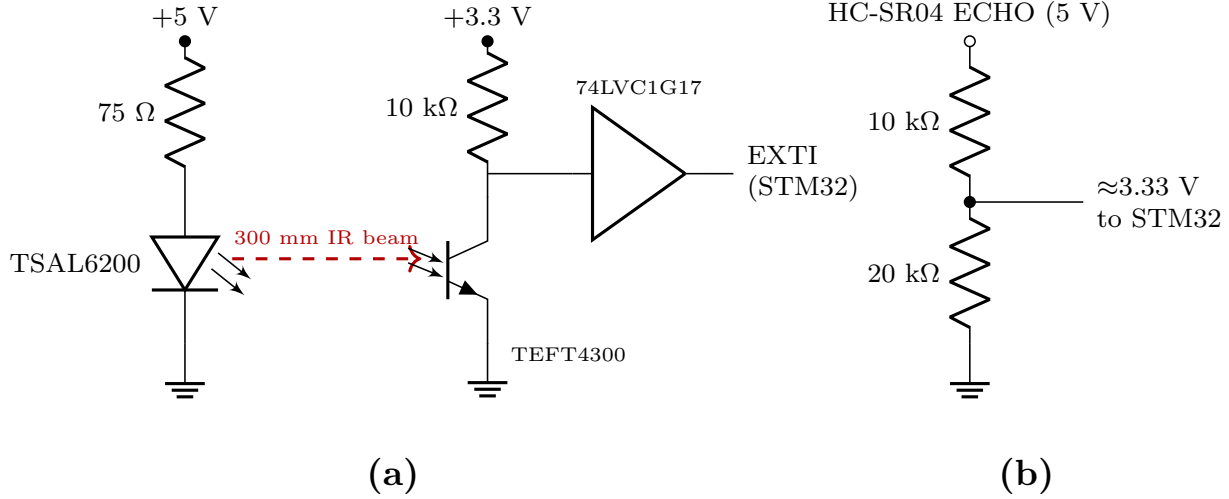


Figure 6: (a) Ball-count beam-break and (b) 5 V-to-3.3 V signal divider

2.5 Software Design

2.5.1 Navigation & Motor Control

The core of the project is the navigation state machine running on the STM32. Every 10 ms (100 Hz) it reads the web command, the battery voltage, the boundary inputs, the range sensors, the hopper sensors and the latest Pixy data, and selects a state of operation. Higher-priority checks pre-empt lower ones, as shown in Figure 7. The states are:

- **IDLE:** Wheels and drum off. Entered at power-up, after a Stop command, when the UART link to the ESP8266 times out, or when the battery falls below 10.5 V.
- **RECOVER:** Entered when a boundary sensor sees the tape. The robot stops, reverses 10 cm, turns away from the tape, and rejoins the sweep at the next lane.
- **AVOID:** Entered when the nearest range reading is below 25 cm. The robot stops, then turns toward the side whose angled sensor reports more free space and continues once the path is clear.
- **RETURN:** Entered when the hopper is full. The drum stops and the robot drives back to its starting corner using odometry, then enters IDLE.
- **SEARCH:** The normal state. The robot follows the back-and-forth sweep at 0.20 m/s,

holding its heading with the IMU and measuring lane length with the encoders.

- **PURSUIT:** Entered when the Pixy reports a ball. The robot stores its position on the sweep and steers toward the ball, slowing as it gets closer.
- **COLLECT:** Entered when the ball is 20 ± 5 cm ahead of the intake and within ± 5 cm of the centreline. The drum turns and the robot creeps forward at 0.03 m/s (scaled with the drum speed measured by the intake encoder, Section 2.7.1) until the beam-break counts the ball or 6 s (two blade passes) elapse, and then resumes SEARCH from the stored point.

Search pattern. Because the camera sees balls well to either side of the robot, the sweep lanes do not need to be as narrow as the intake. We use 1.0 m lane spacing (Figure 8), which covers the $6 \text{ m} \times 6 \text{ m}$ area in six lanes. At 0.20 m/s, the six 5.3 m lanes plus five turns take about $36.8 \text{ m} / 0.20 \text{ m/s} \approx 3.1$ minutes. Allowing roughly 30 s per ball for the approach and the slow COLLECT creep adds 5 minutes for 10 balls, for a total near 8 minutes, inside the 10-minute high-level requirement. A coverage grid of 25 cm cells (576 cells, one bit each, 72 bytes of RAM) records which parts of the area have passed through the camera’s view, which lets us report coverage on the web page.

Ball pursuit. The original Pixy reports object centres in a 320×200 coordinate frame over a 75° horizontal field of view [1]. The equivalent focal length in pixels is $f = 160 / \tan(37.5^\circ) = 208.5$ px, so the heading error to a ball at image position x is

$$\theta_{err} = \arctan\left(\frac{x - 160}{208.5}\right) \quad (2)$$

and the camera-to-ball distance can be estimated from the reported blob width w and the 67 mm ball diameter as $d \approx (208.5)(0.067)/w = 13.97/w$ m. A ball 20 cm ahead of the intake lip is about 0.40 m from the camera and appears roughly 35 px wide. During PURSUIT the wheel command is a forward speed $v = \min(0.10, 0.25(d - 0.40))$ m/s plus a turn rate $\omega = K_p \theta_{err}$, and COLLECT is entered when $d \leq 0.40$ m and the lateral offset $d \sin \theta_{err}$ is within 5 cm. The distance mapping will be calibrated against measured ball positions. Each wheel’s speed is held by a PI loop using encoder feedback.

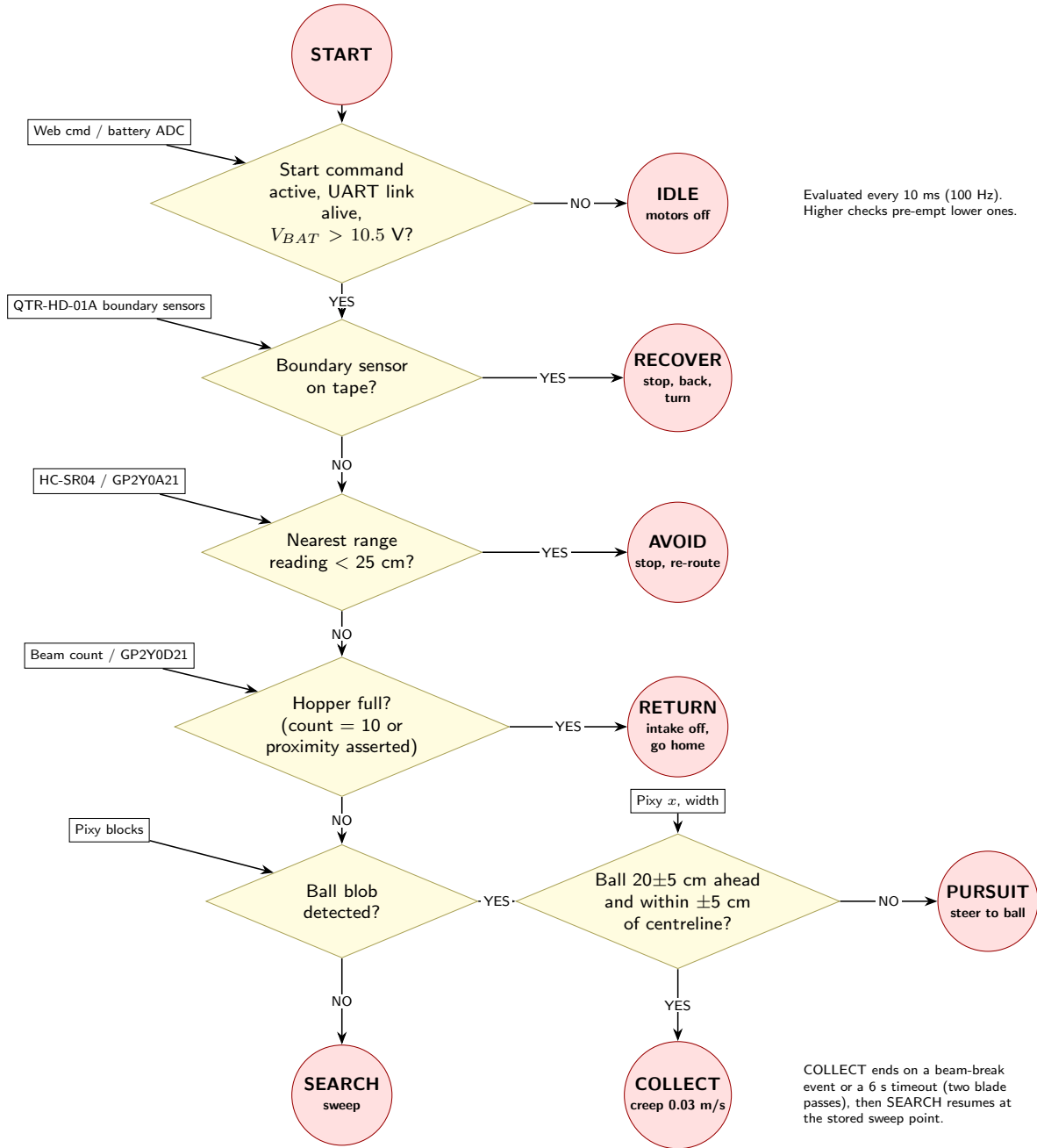


Figure 7: Navigation state-selection flowchart

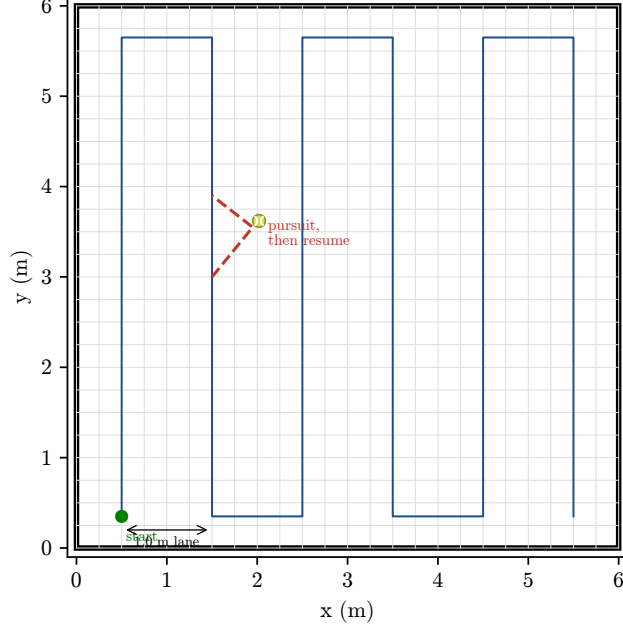


Figure 8: Planned sweep over the 6 m $\times$ 6 m test area with a pursuit detour; grid shows 25 cm coverage cells

2.5.2 Odometry & Encoder Sensing

Each drive motor has a quadrature Hall encoder with two outputs, A and B, offset by a quarter period. As shown in Figure 9, A leads B when the wheel turns forward and B leads A when it turns backward. Rather than decoding these in interrupts, we connect each pair to an STM32 timer in encoder mode, which counts every edge of both channels in hardware and counts up or down according to the phase. The firmware only reads the counter each 10 ms loop.

The GB37-520B drive motors use an 11 pulse-per-revolution Hall encoder on the motor shaft [12], which gives $11 \times 4 = 44$ counts per motor revolution when both edges of both channels are counted. With the nominal 1:56 reduction of the 107 rpm version, one wheel revolution produces $N = 44 \times 56 = 2464$ counts, so each count corresponds to

$$\Delta s = \frac{\pi D_{\text{wheel}}}{N} = \frac{\pi(90 \text{ mm})}{2464} = 0.115 \text{ mm} \quad (3)$$

The exact counts per wheel revolution will be confirmed during bring-up by turning each wheel ten revolutions by hand. At the full 107 rpm wheel speed this is $2464 \times 107/60 \approx 4.4 \text{ k}$

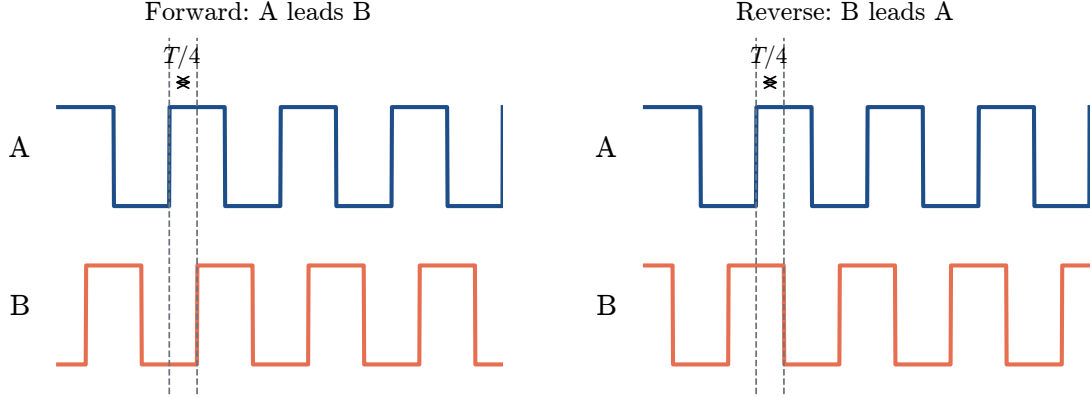


Figure 9: Quadrature encoder outputs for forward and reverse rotation

counts per second per wheel, which the hardware timers handle easily. A third timer decodes the intake motor's 64 CPR encoder, which produces $64 \times 506.25 = 32,400$ counts per drum revolution, or about 5.4 k counts per second at 10 rpm. The firmware uses this to measure the drum speed and to detect a jammed drum, whose count stops changing even though the driver is enabled. The robot's heading change from the wheels is $\Delta\theta_{enc} = (\Delta s_R - \Delta s_L)/b$ with a track width $b = 464$ mm. Because wheel slip makes this estimate drift during turns, it is blended with the integrated MPU-6050 gyroscope rate in a complementary filter, $\theta = 0.98(\theta + \omega_{gyro}\Delta t) + 0.02\theta_{enc}$. The position estimate is reset at each lane end when the boundary sensors detect the tape at a known coordinate.

2.5.3 Pixy Communication

The Pixy is an SPI slave clocked at 1 MHz. It sends each detected object as a 14-byte block made of seven 16-bit words, shown in Figure 10, and marks the start of a new frame with two consecutive sync words [2]. Signature 1 is taught to the yellow-green tennis-ball colour. The STM32 verifies each block's checksum, discards blocks that fail, and keeps the largest valid signature-1 block in each frame as the pursuit target. Even with six blocks per frame, the data rate is about $86 \times 50 = 4.3$ kB/s, around 3% of the 125 kB/s link capacity.

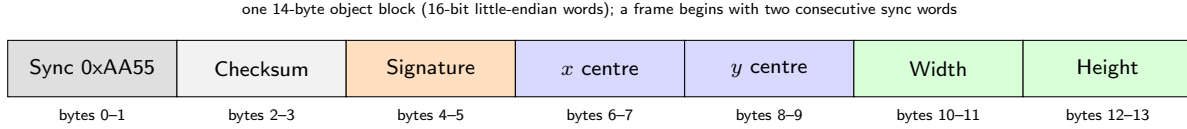


Figure 10: Pixy (CMUcam5) object block format received over SPI

2.5.4 Wireless Communication

The ESP8266 hosts a web page with Start and Stop buttons and a status panel. It talks to the STM32 over UART at 115200 baud. Every frame on the link has the form [0xA5 start byte][type][length][payload][CRC-8], and the STM32 discards any frame whose CRC does not match. Every 100 ms the STM32 sends a status frame whose 32-bit payload is shown in Figure 11: bits (3:0) hold the current state, bits (7:4) the ball count, bits (11:8) flags for hopper full, obstacle, boundary and low battery, bits (23:12) the battery voltage in 10 mV steps, and bits (31:24) the coverage percentage.

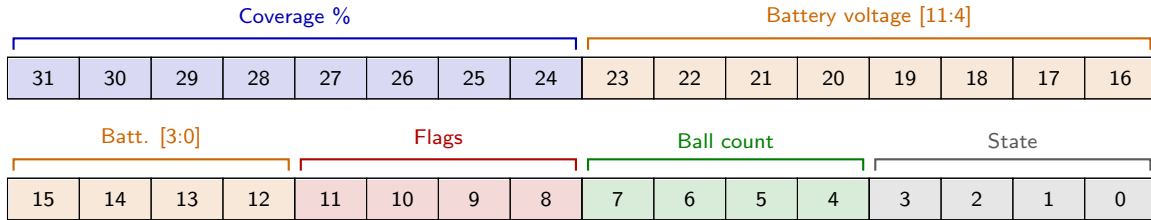


Figure 11: Bit breakdown of the status payload transmitted from the STM32 to the ESP8266

The ESP8266 sends a command frame every 100 ms with the 16-bit payload in Figure 12: bit (0) requests Start, bit (1) requests Stop, bit (2) is a heartbeat that toggles each frame, and bits (15:8) hold a sequence number.

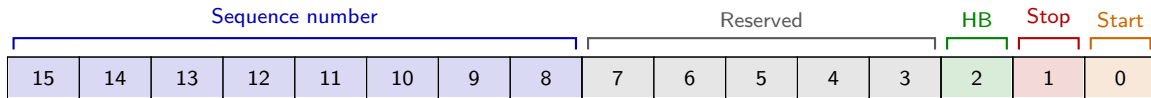


Figure 12: Bit breakdown of the command payload transmitted from the ESP8266 to the STM32

The link is designed to fail safe. If the STM32 receives no valid command frame for 1 s, it assumes the ESP8266 has failed and enters IDLE with all motors off. If the ESP8266 has not received a status poll from the web page for 3 s, it sends a Stop command, so the robot never keeps running after the operator loses the ability to stop it. Obstacle and boundary

handling stay entirely on the STM32 and never wait for the wireless link. A Start or Stop press reaches the robot within one 100 ms frame plus one 10 ms control loop plus network delay, typically well under 0.5 s compared with the 2 s requirement.

2.6 Commercial Component Selection

2.6.1 Drive & Intake Motors

For the wheels we will use two Dyna Engine GB37-520B-12-107 gearmotors, and for the intake one E-S Motor 37SG-3530-506.25K-EN gearmotor. Both are 12 V, 37 mm metal gearmotors with integrated quadrature encoders and 6 mm output shafts, so one set of hubs, brackets and encoder interface circuits serves all three. Their listed specifications are summarized in Figure 13.

Specification	GB37-520B-12-107 (drive)	37SG-3530-506.25K-EN (intake)
Rated voltage	12 V	12 V
No-load output speed	107 rpm	10 rpm
Gear ratio	1:56 (nominal)	1:506.25
Encoder	Hall, 11 PPR (44 counts/motor rev)	Magnetic, 64 CPR
Rated speed / current	–	8 rpm / 0.48 A
Rated / maximum torque	25 kg·cm max	26 kg·cm rated
No-load current	–	0.07 A
Output shaft	6 mm, 15 mm long	6 mm D-shaft, 21.5 mm long
Gearbox / motor size	37 mm gearbox, 33 mm motor	37D×29L / 35D×33L mm

Figure 13: Drive and intake gearmotor specifications from the vendor listings [12, 13]

With 90 mm wheels, 107 rpm corresponds to a no-load ground speed of

$$v = \pi(0.090)\frac{107}{60} \approx 0.50 \text{ m/s}.$$

This is well above our 0.20 m/s search speed, and the firmware caps the commanded speed at 0.5 m/s for safety. The intake’s 506.25:1 gearbox gives the 10 rpm drum speed directly at the motor output. Because the motor is coupled directly to the drum shaft, there is no additional belt or pulley ratio. Its 26 kg·cm rated torque is far more than the roughly 0.5 kg·cm needed to lift a 57 g ball at the 9 cm blade radius, so the drum runs close to its no-load speed. Both motors are operated through external current-limited driver stages connected

to the Rev A motor-control headers described in Section 2.4.2.

2.6.2 Vision Module

We will use the original Pixy (CMUcam5) from the course inventory. The Pixy pairs an image sensor with a dedicated dual-core processor that performs colour-blob detection on every frame and outputs only the object list, so the STM32 never stores image data. A single 320×240 RGB565 frame would occupy about 154 kB of the STM32F405's 192 kB of RAM, so offloading vision is what makes a single microcontroller practical. Its main specifications are shown in Figure 14. The Pixy draws its power from the PCB's 5 V rail through its I/O connector.

Specification	Pixy (CMUcam5)
Processor	NXP LPC4330, 204 MHz, dual core
Image sensor	Omnivision OV9715, 1280×800
Processing rate	50 frames per second
Lens field of view	75° horizontal, 47° vertical
Power	140 mA typical; 5 V (USB / I/O) or 6–10 V unregulated
Data outputs	UART, SPI, I ² C, USB, digital, analog

Figure 14: Pixy (CMUcam5) specifications [1]

2.6.3 Range Sensors

The forward obstacle sensors are one HC-SR04 ultrasonic module and one Sharp GP2Y0A21 IR sensor. The HC-SR04 is triggered by a pulse of at least $10 \mu\text{s}$, emits an eight-cycle 40 kHz burst, and returns an echo pulse whose width is proportional to distance (Figure 15) [6]. Its beam spreads about 15° , so mounted at 210 mm height the beam is still above 175 mm at a range of 25 cm. This means tennis balls on the floor do not trigger obstacle avoidance. The ultrasonic sensor is triggered no more often than once every 60 ms to allow echoes from the previous burst to decay. The Sharp sensor measures 10–80 cm with an analog output that updates continuously, and it covers the short-range region directly ahead.

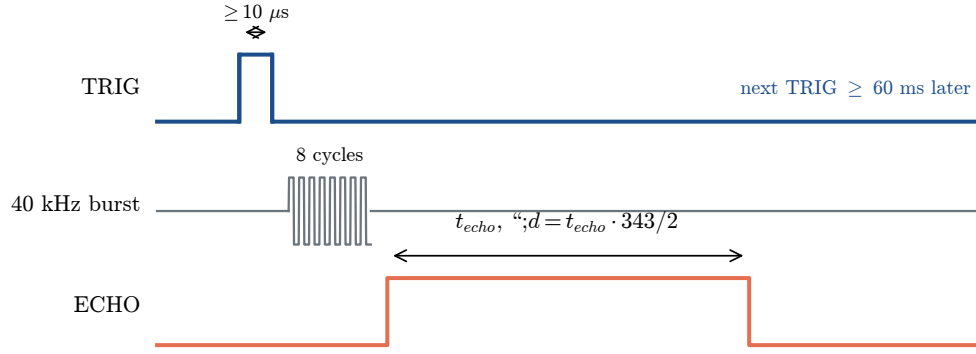


Figure 15: HC-SR04 trigger and echo timing, after the datasheet [6]

2.6.4 Battery

We will use a Turnigy 3S (11.1 V nominal) 2200 mAh 30C Shorty LiPo pack with XT60 connector (SKU 9067160023) [21]. Three cells give a 10.5–12.6 V operating range, which suits the selected 12 V motors and the TPS54302 controller-board converters. The selected pack fits within the basket-centered electronics area and gives a large runtime margin (Section 2.4.3). It will be charged only with a balance charger made for 3S LiPo packs.

2.7 Tolerance Analysis

2.7.1 Intake Blade Kinematics

The largest risk in the project is whether the drum reliably captures a ball while the robot is moving. If the blades move too slowly relative to the robot, the robot pushes the ball forward instead of the blade sweeping it onto the ramp. The drum’s angular velocity at 10 rpm is

$$\omega = (10 \text{ rev/min}) \left(\frac{2\pi \text{ rad}}{1 \text{ rev}} \right) \left(\frac{1 \text{ min}}{60 \text{ s}} \right) = 1.047 \text{ rad/s} \quad (4)$$

and with the 90 mm effective blade radius the tip speed is

$$v_{tip} = \omega r = (1.047)(0.090) = 0.094 \text{ m/s} \quad (5)$$

Compared with the 0.03 m/s approach speed, $v_{tip}/v_{robot} = 3.1$, so the blade sweeps around and behind the ball much faster than the robot advances into it. The blade tip only needs to match the robot's speed at

$$r_{min} = \frac{v_{robot}}{\omega} = \frac{0.03}{1.047} = 2.87 \text{ cm} \quad (6)$$

so the 9 cm design radius has a large margin. Figure 16 shows how the ratio varies with radius and approach speed.

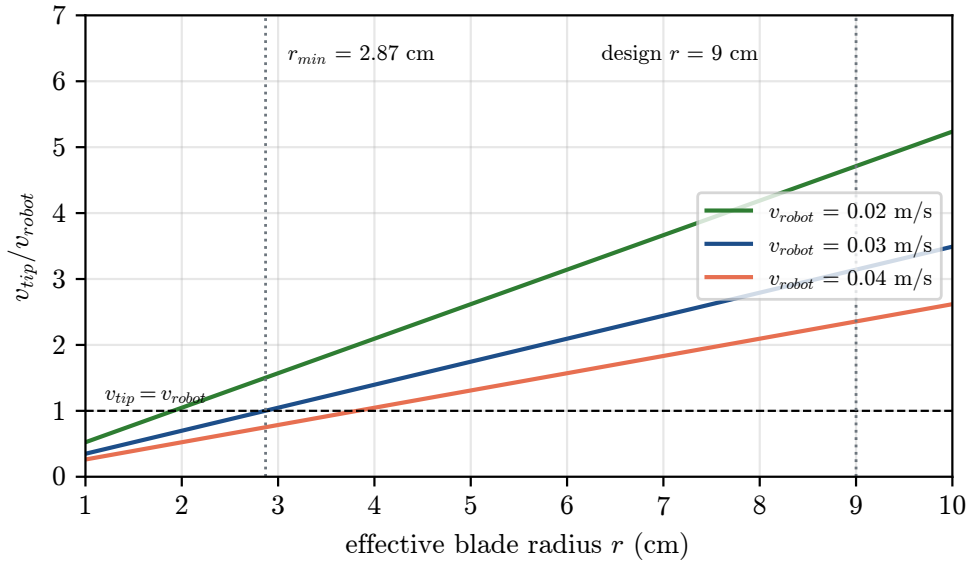


Figure 16: Blade-tip to robot speed ratio at 10 rpm versus effective blade radius

The second constraint is how far the robot travels between blade passes. With two blades at 10 rpm, a blade passes the front of the drum every 3 s. At 0.03 m/s the robot advances $0.03 \times 3 = 90$ mm between passes, which equals the depth of the drum's front sweep zone (from $y = -5$ mm to the shaft at $y = 85$ mm). A ball that enters the sweep zone is therefore always reached by a blade before the robot can overrun it. At 0.04 m/s the robot would advance 120 mm between passes and could push a ball past the sweep zone.

The drum speed is not fixed, however. A brushed motor's no-load speed scales roughly with its supply voltage, so as the battery discharges from 12.6 V to the 10.5 V cutoff the 37SG-3530 slows to about $10 \times 10.5/12 \approx 8.75$ rpm, and the motor's rated-load speed is 8 rpm [13]. At 8.75 rpm a blade passes every 3.43 s, so a fixed 0.03 m/s creep would advance

103 mm between passes, more than the 90 mm sweep depth. We use the intake encoder to remove this sensitivity. During COLLECT the firmware sets the creep speed from the measured drum speed n (in rpm) so that the robot advances exactly one sweep depth per blade pass:

$$v_{collect} = \frac{d_{sweep}}{t_{pass}} = (0.090 \text{ m}) \left(\frac{2n}{60 \text{ s}} \right) = 0.003 n \text{ m/s} \quad (7)$$

This gives 0.030 m/s at 10 rpm and 0.026 m/s at 8.75 rpm. Figure 17 compares the advance per blade pass for a fixed and an encoder-scaled creep speed.

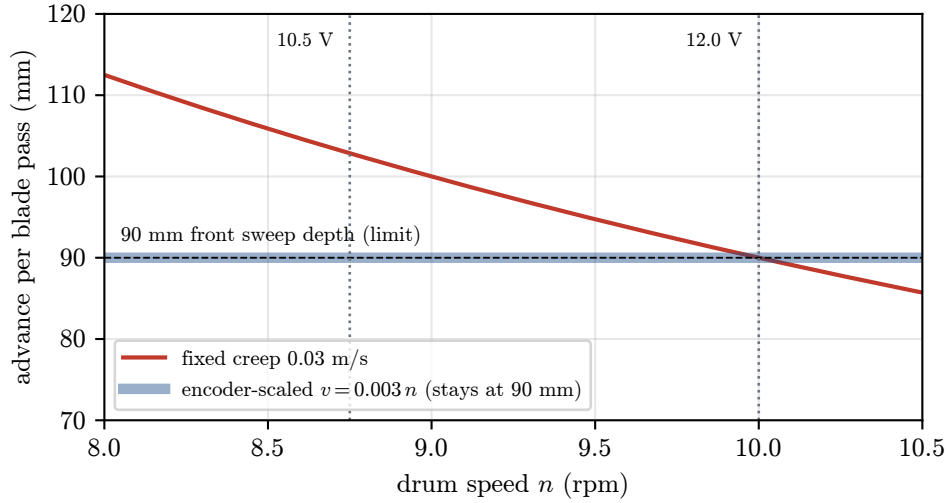


Figure 17: Robot advance per blade pass versus drum speed for a fixed and an encoder-scaled COLLECT speed

Because both the creep speed and the tip speed scale with n , the tip-to-robot speed ratio stays at 3.1 across the whole battery range. This analysis shows that the chosen speeds are kinematically sound. The actual capture rate also depends on blade shape, friction and the ramp lip, so it will be confirmed by the 8-of-10 intake test.

2.7.2 Obstacle Stopping Distance

To meet the obstacle-avoidance requirement, the robot must stop before reaching a 30 cm cube when travelling at 0.20 m/s. The worst-case delay between the obstacle entering the stop threshold and the motors receiving a stop command is one 60 ms HC-SR04 measurement

interval plus one 10 ms control loop:

$$t_{delay} = 60 \text{ ms} + 10 \text{ ms} = 70 \text{ ms}, \quad d_{delay} = (0.20)(0.070) = 14 \text{ mm} \quad (8)$$

After the stop command, the motor-control layer disables forward torque and requests active braking from the external driver stage where supported. Using a conservative measured-design target of 0.5 m/s^2 deceleration, the braking distance is

$$d_{brake} = \frac{v^2}{2a} = \frac{(0.20)^2}{2(0.5)} = 40 \text{ mm} \quad (9)$$

for a total stopping distance of 54 mm. The stop threshold is 250 mm from the sensor face, and the front of the hood is about 10 mm ahead of the sensors, so the robot stops with roughly 186 mm of clearance. This leaves margin for ranging error, a lower actual deceleration, and an occasional missed reading.

2.8 Cost Analysis

Labor. The course cost model is $(\$/\text{hour}) \times 2.5 \times \text{hours}$. Grainger reports an average 2024–2025 starting salary of \$87,429 for Illinois electrical-engineering graduates [20]; this corresponds to approximately \$42.03/hour using 2080 work hours/year. We budget 120 engineering hours per team member for design, fabrication, integration and verification:

$$(\$42.03/\text{h})(2.5)(120 \text{ h}) = \$12,609 \text{ per member.}$$

For three members, estimated labor is **\$37,827**.

Parts and services. Table 7 lists the complete system-level parts and the specialized Rev A PCB items. Course/shop-stock items have a project purchase cost of \$0; their replacement value is not counted. Prices for purchased parts are current planning prices or distributor one-piece prices; course/shop-stock items have zero acquisition cost. The detailed passive/reference-designator BOM remains in the KiCad project and is summarized here by function to keep the design document readable.

Table 7: Bill of materials and project acquisition cost

Description / source	Manufacturer	Part #	Qty	Cost
107 rpm encoder drive motor, RobotShop	Dyna Engine	GB37-520B-12-107	2	\$25.00
10 rpm encoder intake motor, RobotShop	E-S Motor	37SG-3530-506.25K-EN	1	\$15.58
HC-SR04 ultrasonic module, ECE shop loan	Various	HC-SR04	1	\$0.00
MPU-6050 IMU module, DFRobot	DFRobot	SEN0142	1	\$13.64
Analog IR range sensor, DigiKey	Sharp	GP2Y0A21YK0F	1	\$9.90
Analog reflectance sensor, Pololu	Pololu	QTR-HD-01A #4201	2	\$6.50
3S 11.1 V, 2200 mAh 30C LiPo, HobbyKing	Turnigy	9067160023	1	\$13.99
3S LiPo balance charger, lab equipment	ECE lab	3S balance charger	1	\$0.00
External 2 A logic fuse, motor-branch fuse/disconnect and power wiring	ECE shop stock	protected power harness	1 set	\$0.00*
Original Pixy camera, course inventory	Charmed Labs	CMUcam5	1	\$0.00
Plastic basket, wheels, casters, aluminium rails/brackets, shaft, blades, ramp and fasteners	ECE shop / team stock	shop-fabricated	1 set	\$0.00
External motor-driver carriers, Pololu	Pololu / TI	DRV8874 carrier #4035	3	\$35.82
ESP8266 Wi-Fi module, Adafruit	AI-Thinker / Adafruit	ESP-12F (PID 2491)	1	\$6.95
IR ball-count emitter, DigiKey	Vishay	TSAL6200	1	\$0.50
IR ball-count receiver, DigiKey	Vishay	TEFT4300	1	\$0.70
Hopper-full digital IR sensor, DigiKey	Sharp	GP2Y0D21YK0F	1	\$7.83
Beam-break Schmitt buffer, DigiKey	Texas Instruments	SN74LVC1G17DBVR	1	\$0.15
Beam-break resistor/shroud/wiring	ECE shop stock	75 Ω 0.5 W + 10 k Ω	1 set	\$0.00*
STM32F405 MCU, ST	STMicroelectronics	STM32F405RGT6	1	\$0.00*
5 V / 3.3 V buck converters, TI	Texas Instruments	TPS54302DDCR	2	\$0.00*
Power magnetics	Bourns	SRP1038A-100M / -6R8M	2	\$0.00*
Protection parts	Vishay / Littelfuse-equivalent	1N5822, SMBJ15A	2	\$0.00*
Precision regulator resistors and specified bulk/ceramic capacitors	Panasonic / Murata / Yageo	per KiCad BOM	1 set	\$0.00*
8 MHz crystal, reset switch, status LED and logic ICs	Abracon / Omron / TI	ABLS-8MHz; B3F-1000; SN74LVC14A; SN74AHCT04; SN74LVC2G00	1 set	\$0.00*
Headers, 0805 passives, terminal block and test hardware	ECE shop stock	per KiCad BOM	1 set	\$0.00*
Rev A bare PCB fabrication	ECE 445 course allocation	98 mm $\times$ 98 mm, 2-layer	1 lot	\$0.00
Machine-shop fabrication labor	ECE machine shop	course service	4 h planned	\$0.00
Purchased-parts subtotal				\$136.56

*Available through course/lab inventory or the course PCB process; if stock is unavailable, the team will update the BOM with the actual distributor quote before purchase.

The **grand project cost under the course labor model is \$37,963.56** (\$37,827 la-

bor + \$136.56 current acquisition cost). Shipping and tax are excluded until vendor checkout. This total distinguishes economic labor value from the team’s actual out-of-pocket/course purchase cost.

2.9 Schedule

The schedule below follows the Fall 2026 course calendar [14]. Mohammed leads the Power, Drive & Communication subsystem (including the PCB), Yousef leads Vision & Navigation, and Zeyad leads Intake & Storage. Course deadlines are shown in bold. Because the first PCBway round (Sep 30) has passed, the board targets the second round on Oct 8, leaving the third PCBway round (Oct 15) and the fourth round through JLCPCB (Oct 29) for revisions.

2.10 Risk Analysis

The component that poses the greatest risk to the project’s success is the intake. Every other subsystem can be tested on the bench, but whether the two curved blades and the fixed ramp reliably lift a ball into the bin can only be proven on a physical prototype, and the 4 mm nominal clearance between the blade sweep and the ramp leaves little room for blade flex or misalignment. To reduce this risk, the drum and its direct-drive motor are mounted on adjustable supports so the drum axis can be raised during tuning, the blades will be tried in both formed aluminium and plastic, and the intake is built and tested on its own rig before it is mounted on the chassis. The approach speed is also fixed at 0.03 m/s, which the tolerance analysis shows is the fastest speed at which a blade always reaches a ball.

The second risk is vision. The original Pixy filters by hue and saturation, which works well for the bright yellow-green ball but poorly for white tape, which has almost no saturation. We therefore do not rely on the camera for containment. The downward-facing QTR-HD-01A sensors provide the boundary stop, and odometry inside the known 6 m $\times$ 6 m area provides the planning. The ball signature will be tested under the actual lab lighting

Week	Task	Person
Sep 28 – Oct 2	Design Document due Fri 10/2 ; design review sign-up	Everyone
	Order drive/intake motors, sensors and battery	Everyone
	Finalize controller PCB and purchasing BOM	Mohammed
	Verify Pixy/IMU/range-sensor interfaces and test plan	Yousef
	Measure basket/rail hardware and freeze intake mounting datums	Zeyad
Oct 5 – Oct 9	Design Review (Mon 10/5 – Wed 10/7)	Everyone
	Finish power and motor-driver schematic and 2-layer layout; pass PCBway audit	Mohammed
	Second Round PCBway Order (must pass audit) – Thu 10/8, 4:45p	Mohammed
	Pixy bring-up: teach ball signature, read blocks over SPI on dev board	Yousef
Oct 12 – Oct 16	Basket rail/bracket layout plus drum and ramp drawings; intake test rig	Zeyad
	Breadboard Demo (Mon 10/12 – Wed 10/14) : Pixy tracking, motor drive, range and boundary sensors	Everyone
	Teamwork Evaluation I – Wed 10/14	Everyone
	Third Round PCBway Order (revisions) – Thu 10/15, 4:45p	Mohammed
	Encoder (TIM) and SEN0142 IMU drivers on STM32	Yousef
Oct 19 – Oct 23	Last day for machine shop revisions – Fri 10/16	Zeyad
	Solder PCB; verify 5 V / 3.3 V rails and DRV8874 current limits	Mohammed
	Odometry and heading filter; straight-line and 360° tests	Yousef
	Intake pickup trials on rig; tune drum height and blade material	Zeyad
Oct 26 – Oct 30	Fourth Round JLCPCB Order (final revisions) – Thu 10/29, 4:45p	Mohammed
	ESP8266 web page and UART protocol	Mohammed
	Search sweep, boundary stop and coverage grid	Yousef
	Mount direct-drive intake; beam-break and hopper-full sensors	Zeyad
Nov 2 – Nov 6	Progress Demo (Mon 11/2 – Wed 11/4)	Everyone
	Individual progress reports – Wed 11/4	Everyone
	Obstacle-avoidance tests	Mohammed
	Pursuit control and distance calibration	Yousef
Nov 9 – Nov 13	Ball counting, encoder-scaled COLLECT speed, full-hopper logic	Zeyad
	Full integration; R&V tests for each subsystem	Everyone
	Battery runtime and reset tests	Mohammed
	Vision/navigation R&V reruns and coverage analysis	Yousef
Nov 16 – Nov 20	Intake/capacity R&V reruns and jam testing	Zeyad
	Team Contract assessment – Fri 11/13	Everyone
	Mock demo (weekly TA meeting); Mock Presentation (Thu 11/19 – Fri 11/20)	Everyone
	Power/driver fault-injection and final hardware regression	Mohammed
Nov 23 – Nov 27	Full autonomous sweep/pursuit regression and log review	Yousef
	Intake endurance, ball-count and full-hopper regression	Zeyad
	Fall Break	–
Nov 30 – Dec 4	Final Demo (Mon 11/30 – Wed 12/2)	Everyone
	Demo power/PCB readiness and fault recovery	Mohammed
	Demo navigation, pursuit and coverage behavior	Yousef
	Demo intake, counting and hopper-full behavior	Zeyad
Dec 7 – Dec 10	Final Presentation (Thu 12/3 – Fri 12/4); Video Assignment – Fri 12/4	Everyone
	Final paper – Wed 12/9	Everyone
	Final power/PCB results and cost documentation	Mohammed
	Final navigation/vision results and plots	Yousef
	Final intake/storage results and mechanical drawings	Zeyad
	Lab checkout and lab notebook – Thu 12/10	Everyone

Figure 18: Schedule for project progression, aligned with the ECE 445 Fall 2026 calendar

in the first week.

Finally, the robot is an autonomous machine with a rotating intake and a LiPo battery, so failures must be handled conservatively. The DRV8874 motor-driver carriers provide adjustable hardware current limiting, obstacle and boundary sensors command stops through the STM32, loss of the wireless link or web client stops the robot, and the speed is capped well below 0.5 m/s. These measures reduce the chance that a single communication or sensing fault causes uncontrolled motion.

3 Societal Impact, Engineering Standards, Ethics, and Safety

3.1 Societal Impact

The project can improve the welfare of players and coaches by reducing repetitive ball-collection time and allowing practice sessions to spend more time on instruction. A compact, low-cost platform may be more accessible to schools, clubs and individual users than commercial autonomous collectors. Rechargeable operation avoids disposable batteries, while the LiPo pack and electronics must eventually be handled through appropriate battery and e-waste recycling channels. The prototype is validated only in a 6 m $\times$ 6 m indoor taped area under normal room lighting; we will not represent it as safe or reliable for full outdoor courts until those conditions are separately verified.

3.2 Engineering Standards

The project uses standards and published safety guidance where they are applicable. UL 2054 and IEC 62133 provide safety guidance for rechargeable battery systems [22, 23]. USB or Wi-Fi certification is not claimed because the robot is a course prototype rather than a marketed product. The custom PCB follows the ECE 445 PCB checklist and manufacturer layout/data-sheet requirements for the STM32 and power converters. If the design were commercialized, applicable FCC intentional-radiator requirements for the Wi-Fi radio and product-level machinery/battery standards would have to be reviewed before sale.

3.3 Ethics

The IEEE Code of Ethics asks engineers to hold paramount public safety and to be honest and realistic in technical claims [18]. The ACM Code similarly emphasizes avoiding harm and providing thorough evaluations of systems and their risks [19]. These principles require us to report measured collection, stopping and runtime results rather than presenting simulations or vendor specifications as demonstrated performance. The camera is used only for ball

detection: the Pixy outputs object coordinates rather than recording or transmitting images, reducing privacy concerns for players and bystanders.

3.4 Electrical and Mechanical Safety

The 3S LiPo reaches 12.6 V when fully charged, and short circuits, overcharging, puncture or excessive current can cause overheating or fire. The battery is secured against movement, charged only with a 3S balance charger, inspected before each use, and removed from service if swollen, damaged or abnormally hot. The logic branch has a fuse and reverse/transient protection; the motor branch is separately fused and has a physical disconnect. Exposed power connections are insulated, and motor wiring is kept separate from low-level sensor wiring.

The rotating intake presents pinch and entanglement hazards. A hood guards the blade sweep and the direct motor-to-drum coupling wherever practical. Adjustment, clearing a jam, or handling the drum is performed only with motor power physically disconnected. The robot is first tested with wheels raised and then at low speed in a controlled area before autonomous operation near people.

3.5 Safety Mitigation Procedure

Before each powered test the team (1) inspects the LiPo and verifies the correct fuse/disconnect, (2) checks that the intake guard and motor mounts are secure, (3) confirms the normally-closed stop loop disables hardware motor permission when opened, (4) powers the controller with motors disabled and checks 5 V/3.3 V rails, (5) tests one motor channel at a time with wheels raised, and (6) clears the test area before autonomous motion. During operation a person remains within reach of the physical power disconnect. Loss of valid supervisory communication, boundary detection, a driver fault, or a controller fault produces a motor-disabled state and requires deliberate restart. These procedures protect both developers and nearby users from foreseeable electrical, battery and mechanical hazards.

References

- [1] Charmed Labs and Carnegie Mellon University, “CMUcam5 (Pixy) overview and specifications.” [Online]. Available: <https://docs.openmv.io/dev/legacy/cmucam5/>
- [2] Charmed Labs, “Pixy (CMUcam5) porting guide: object block format.” [Online]. Available: https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:porting_guide
- [3] STMicroelectronics, *STM32F405xx/STM32F407xx Datasheet*, DS8626. [Online]. Available: <https://www.st.com/resource/en/datasheet/stm32f405rg.pdf>
- [4] Texas Instruments, *TPS54302 4.5-V to 28-V Input, 3-A Synchronous Step-Down Converter*. [Online]. Available: <https://www.ti.com/product/TPS54302>
- [5] Espressif Systems, *ESP8266EX Datasheet*. [Online]. Available: <https://www.espressif.com/en/support/documents/technical-documents>
- [6] “HC-SR04 Ultrasonic Sensor,” datasheet. [Online]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [7] MaxBotix, “Using multiple ultrasonic sensors.” [Online]. Available: <https://maxbotix.com/blogs/blog/using-multiple-ultrasonic-sensors>
- [8] Sharp, *GP2Y0A21YK0F Distance Measuring Sensor Unit* datasheet.
- [9] Sharp, *GP2Y0D21YK0F Distance Measuring Sensor Unit (Digital Output, 24 cm)* datasheet.
- [10] Pololu, “QTR-HD-01A Reflectance Sensor: 1-Channel, 5mm Wide, Analog Output (#4201).” [Online]. Available: <https://www.pololu.com/product/4201>
- [11] DFRobot, “SEN0142 6 DOF Sensor – MPU6050.” [Online]. Available: <https://wiki.dfrobot.com/sen0142/>
- [12] Dyna Engine, “12V 107RPM Brushed DC Gear Motor 37D w/ Encoder, GB37-520B-12-107,” RobotShop. [Online]. Available: <https://www.robotshop.com/products/dyna-engine-12v-107rpm-brushed-dc-gear-motor-37d-w-encoder>
- [13] E-S Motor, “GM37 Geared Motor w/ Encoder – 12V 10RPM, 37SG-3530-506.25K-EN,” RobotShop; and “GM37 Geared Motor w/ Encoder – 12V 56RPM, 37SG-3530-90K-EN” (stall current of the same motor family). [Online]. Available: <https://robotshop.com/products/e-s-motor-gm37-geared-motor-w-encoder-12v-10rpm>
- [14] ECE 445, “Calendar,” University of Illinois Urbana-Champaign, Fall 2026. [Online]. Available: <https://courses.grainger.illinois.edu/ece445/calendar.asp>

- [15] Vishay Semiconductors, *TSAL6200 High Power Infrared Emitting Diode, 940 nm* datasheet; *TEFT4300 Silicon NPN Phototransistor* datasheet.
- [16] Pololu, “DRV8874 Single Brushed DC Motor Driver Carrier, item #4035.” [Online]. Available: <https://www.pololu.com/product/4035>
- [17] Tennibot, “Rover autonomous tennis ball collector.” [Online]. Available: <https://www.tennibot.com/rover/>
- [18] IEEE, “IEEE Code of Ethics.” [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>
- [19] Association for Computing Machinery, “ACM Code of Ethics and Professional Conduct.” [Online]. Available: <https://www.acm.org/code-of-ethics>
- [20] The Grainger College of Engineering, University of Illinois Urbana-Champaign, “Electrical Engineering - post-graduate outcomes, 2024–2025.” [Online]. Available: <https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/electrical-engineering>
- [21] HobbyKing, “Turnigy 2200mAh 3S (11.1V) 30C Shorty LiPo Battery Pack w/XT60, SKU 9067160023.” [Online]. Available: https://hobbyking.com/en_us/turnigy-2200mah-3s-11-1v-30c-shorty-lipo-battery-pack-w-xt60.html
- [22] UL Solutions, *UL 2054: Household and Commercial Batteries*.
- [23] IEC, *IEC 62133: Secondary Cells and Batteries Containing Alkaline or Other Non-Acid Electrolytes – Safety Requirements for Portable Sealed Secondary Cells*.