Electrical and Electronic Report – Team Rulo Bot – WRO Future Engineers 2025

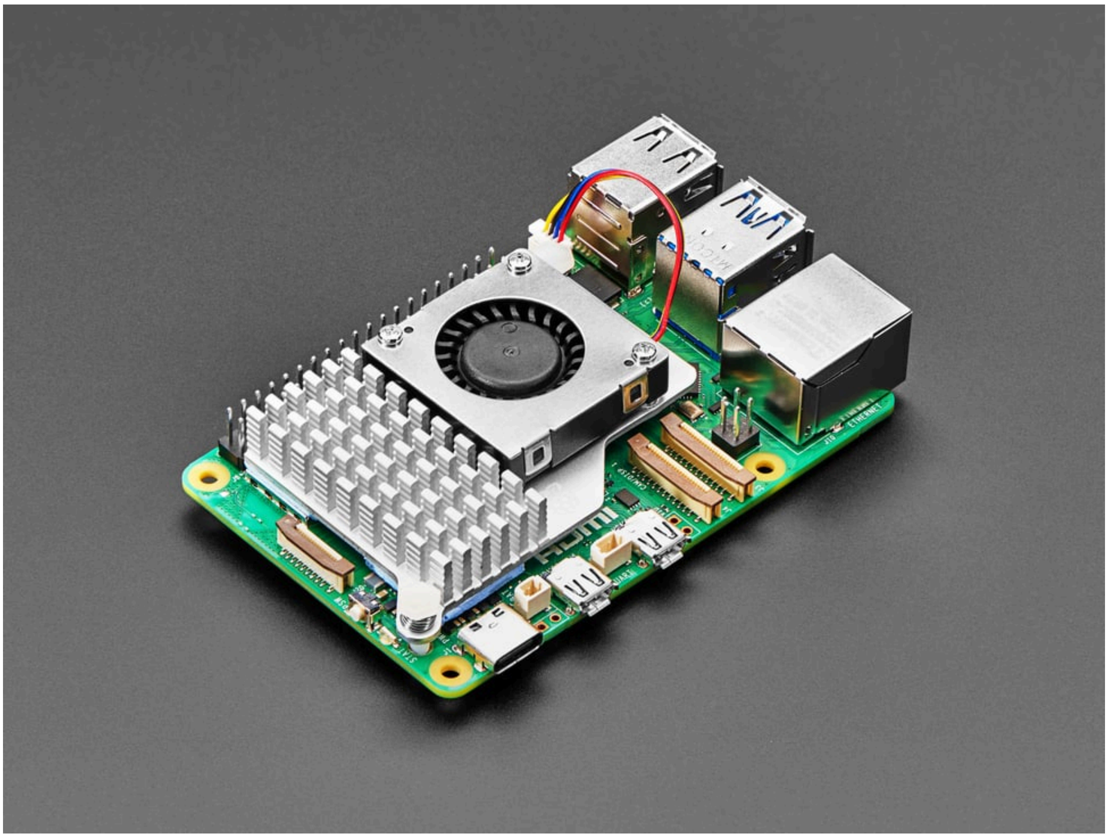
Introduction

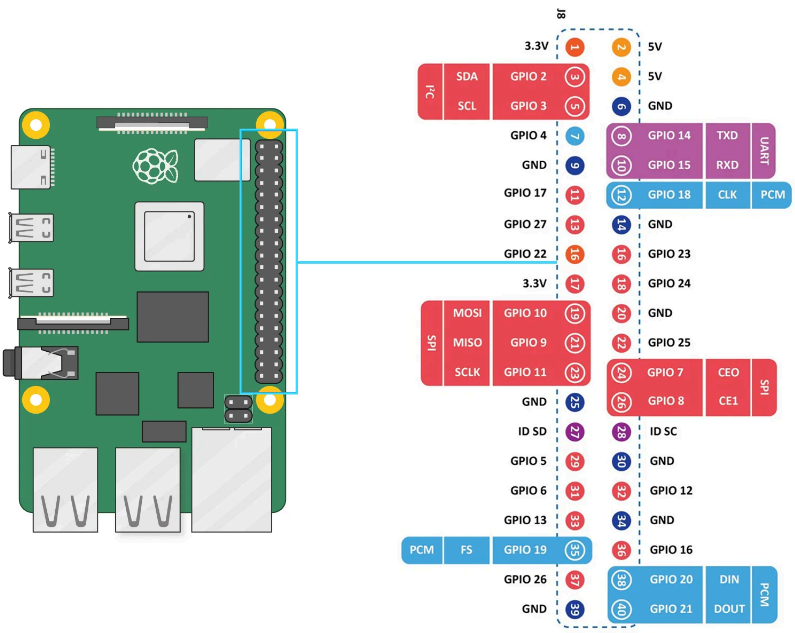
This report details the electrical and electronic architecture of the Team Rulo Bot robot developed for the WRO Future Engineers 2025 category. The robot's electrical/electronic system has the main objective of providing stable power and reliable control to all subsystems (computing, sensors, and actuators) required for the mission of an autonomous vehicle. This implies designing a robust power distribution capable of supplying different voltages (high current at 11.1 V for motors, regulated 5 V for logic and servomechanisms) and protecting sensitive components. In addition, the goal is to efficiently integrate the electronic controllers—one main computer and auxiliary microcontrollers—with the motors and sensors to achieve precise control of motion (Ackermann traction and steering) and data acquisition (vision and wheel encoder). In short, the objectives of the electrical system are reliability, safety, and optimal performance, ensuring that the robot can operate autonomously during competition rounds without electrical failures.

Main System Components

Below are listed and described in detail the electrical and electronic components used in the robot, including their relevant characteristics, functions within the system, and power requirements:

Raspberry Pi 5

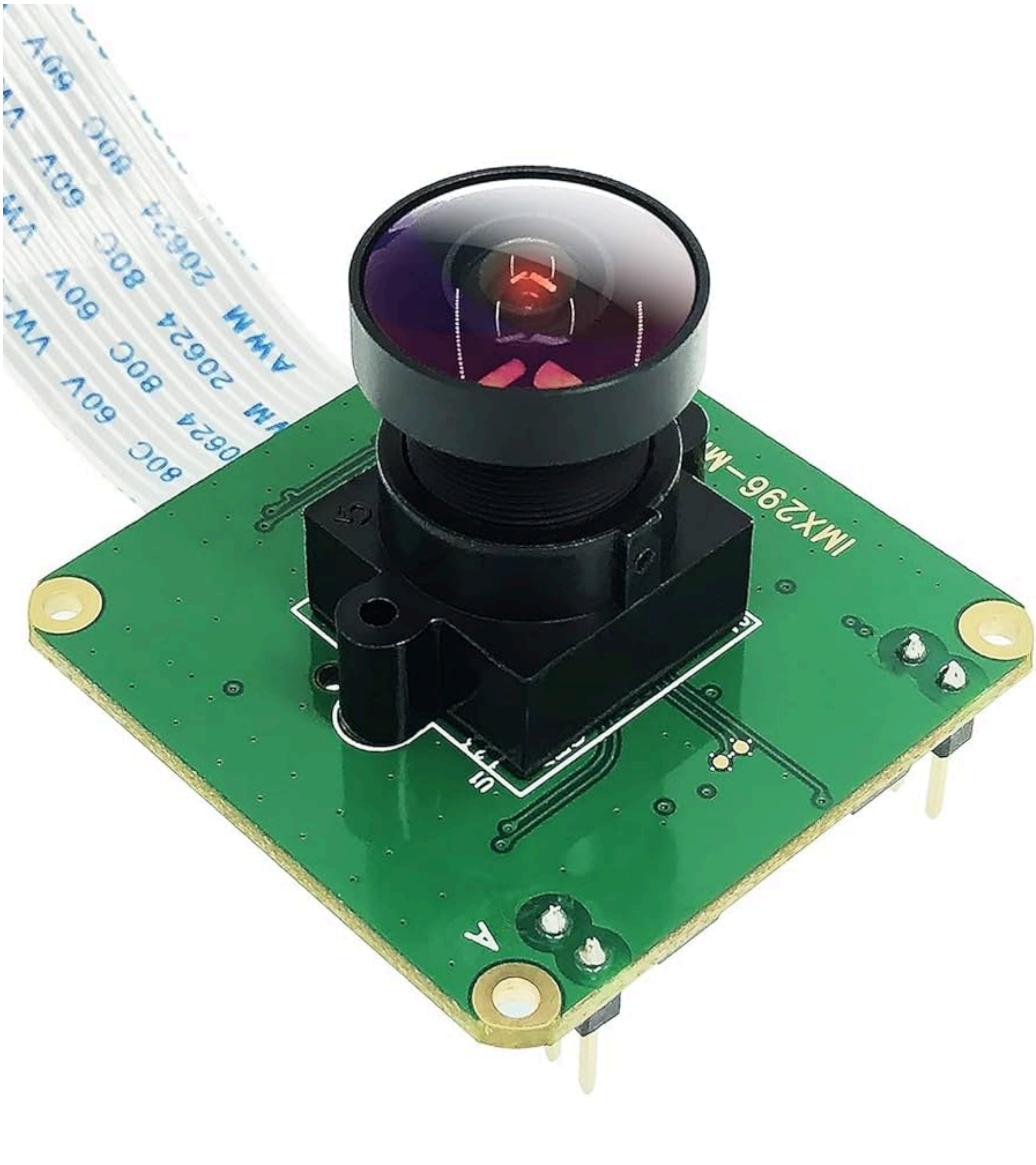




The Raspberry Pi 5 is the robot's main computer and is responsible for running computer vision (camera processing) and high-level decision-making. It is the latest generation of small single-board microcomputer, with a quad-core Arm Cortex-A76 processor at 2.4 GHz and memory options of 4 GB or 8 GB of LPDDR4X RAM. This powerful hardware provides a 2–3× performance increase over the previous generation, which is essential for running real-time pillar detection, lane following, and artificial intelligence algorithms. The Raspberry Pi 5 runs a Linux operating system and offers numerous expansion ports (40-pin GPIO, USB 3.0/2.0, microHDMI, CSI/DSI for camera and display, etc.), facilitating the integration of additional sensors and actuators.

Regarding its power supply, the Raspberry Pi 5 requires a stable 5 VDC source capable of supplying several amperes of current (up to ~5 A to achieve maximum performance with peripherals). In this design, the board is powered from the 5 V DC-DC regulator (described later), connected to the Pi's 5 V input. It is important to highlight that powering the Pi directly via its 5 V pin is valid as long as the voltage is well regulated at 5.0 V, since any excess above ~5.25 V could damage it. The source's available current can be much higher than what the Pi consumes (for example, a battery capable of 20 A); this is not a problem as long as the voltage remains constant and free of noise. For greater safety, the Pi 5 includes USB-C power protection circuitry with PD (Power Delivery) protocol, but when using the direct 5 V line, regulation must be guaranteed and, preferably, a fuse or current limiter should be added to avoid overloads. In summary, the Raspberry Pi 5 acts as the robot's "brain," receiving regulated 5 V power and sending control signals to the actuators (motor, servo) through its GPIO, as well as processing the USB camera input and communicating with the auxiliary microcontroller (Arduino) as needed.

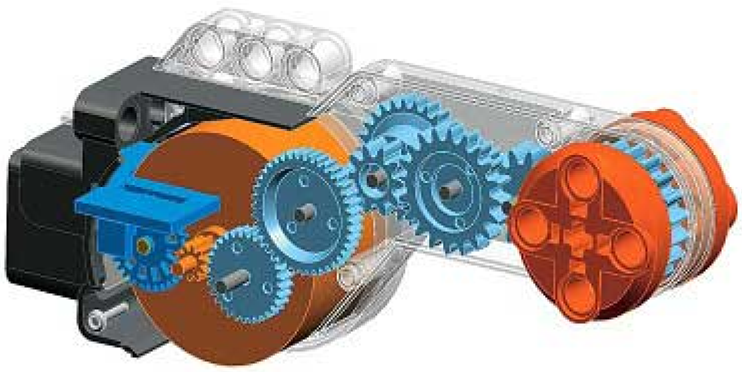
Innomaker USB UVC 2.0 Camera



The robot's vision system consists of a USB 2.0 digital camera from Innomaker, compatible with the UVC (USB Video Class) standard. This camera (UVC 2.0 model) is a plug-and-play module that the Raspberry Pi recognizes without special drivers, facilitating its use in machine vision applications. It features a 2-megapixel sensor capable of delivering video at a resolution of 1920×1080 (Full HD) at 30 FPS, encoded in MJPEG or YUYV (YUV2) formats. Some Innomaker UVC models incorporate a wide-angle lens (for example, ~95° diagonal field of view) to cover more of the environment, which is useful for lane following and object detection on the track.

The camera is powered directly through the Raspberry Pi 5's USB port, using the +5 V line it provides. Its consumption is relatively low (approximately 0.25 W typically, equivalent to ~50 mA at 5 V), so it does not represent a significant load for the Pi's power source. Since it complies with the UVC specification, the camera can send the video stream without saturating the CPU, leveraging generic drivers from the operating system. In the robot's logic, this camera is used to detect colored pillars and the circuit's lines, contributing to the autonomous navigation algorithm. The choice of a USB UVC device provides mounting flexibility (it is connected by cable to the Pi's USB port and can be positioned at the front of the robot) and reduces compatibility risks. It is a critical component, so ensuring proper power supply (good USB contact and voltages within range) and minimizing electrical interference on its USB line is essential to obtain stable images.

LEGO NXT Motor with Encoder



The vehicle's main traction actuator is a LEGO NXT DC motor (LEGO 9842) with an internal encoder. This motor is originally part of the Mindstorms NXT set and is designed to operate at approximately 9 V DC nominal, incorporating a reduction gearbox and an integrated quadrature optical encoder. Its angular resolution stands out: the internal rotation sensor records approximately 360 pulses per revolution, equivalent to ~1° resolution per pulse. This means the motor can be used not only for propulsion but also as a wheel rotation sensor, measuring traveled distances and speeds with high accuracy. In fact, LEGO's technical manual indicates that NXT servomotors include an integrated rotation sensor with ±1° accuracy on the shaft, commonly used for PID control in the NXT brick. In our case, that quadrature encoder (two channels offset by 90°) is read by the Arduino microcontroller, allowing pulse counting and determining both speed and direction of motor rotation (e.g., to implement an odometer).

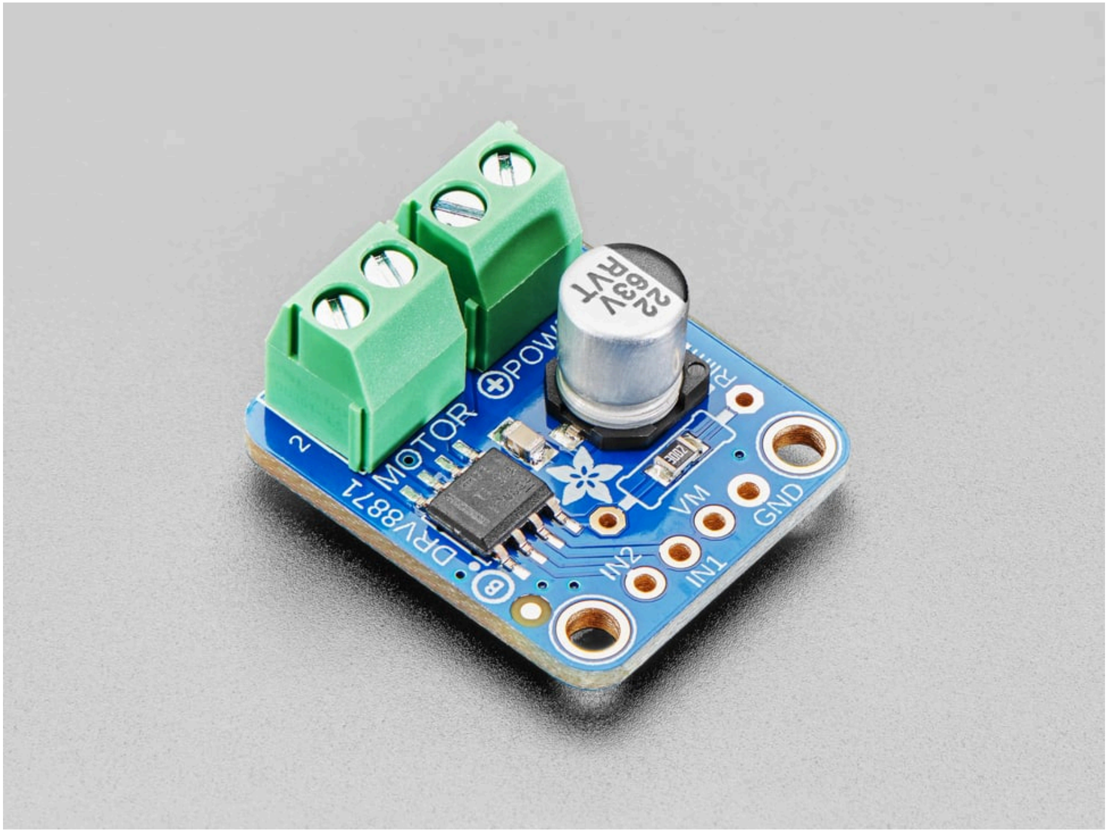
Regarding power, the NXT motor is connected to a DRV8871 driver that provides variable voltage for speed control. The maximum voltage it receives is that of the battery (11.1 V) or a nearby regulated value (depending on the configuration used; see the power section). Although the motor is specified at 9 V, in practice it can run at higher voltages for short periods; powering it directly with 11.1–12.6 V (fully charged 3S LiPo) increases its speed and power, but care must be taken not to exceed currents or cause overheating. The driver used allows modulating the effective voltage via PWM, so the motor will typically operate with duty cycles that deliver an average voltage close to 9–10 V under nominal conditions. In addition, the driver limits current (configurable) to protect both the motor and the source (battery). The NXT motor includes six pins in its connector: two for the motor coil (which receive power from the driver) and four for the encoder (encoder power at 5 V and two quadrature outputs A/B). These latter connect to the Arduino Nano, which conditions and counts the encoder signals. In summary, the LEGO NXT motor acts as the robot’s drive wheel, delivering motion to the tires, while its internal encoder provides essential feedback information for distance and speed control.

MG995 Servomotor (Ackermann Steering)

To actuate steering (the turning mechanism of the front wheels in an Ackermann configuration), the robot uses a Tower Pro MG995 hobby servomotor. This is a standard-size servo known for its high torque and metal gears. The MG995 can rotate ~180° (±90° from center) in its standard rotation version and delivers torque on the order of 9.4 kg·cm at 4.8 V and up to ~11 kg·cm at 6.0 V, sufficient to turn the robot’s wheels under load. Its response speed is approximately 0.20 s/60° at 4.8 V (0.16 s/60° at 6 V), which ensures reasonably fast changes of direction according to control commands. Internally, the MG995 uses a DC motor with reduction and a potentiometer for position control; it receives a PWM control signal at ~50 Hz from the microcontroller and automatically maintains the indicated angle within its mechanical range.

In the robot's electronics, the MG995 servo is powered from a regulated rail dedicated to actuators. Its consumption varies according to load: at rest it consumes a few mA, but under load it can demand currents on the order of 0.5–1 A, and even peaks of ~2 A at start-ups or mechanical stall. For that reason, it is considered an important power element in the electrical distribution. The MG995 shares ground with the Raspberry Pi and the Arduino; the servo's control pin is connected to the Arduino Nano, which generates the PWM signal needed to position the steering according to the algorithm's calculations (for example, outputs of the vision-based PD steering controller). It is worth mentioning that the MG995 servo is "economical but high torque," as described by its manufacturer, making it ideal for mobile robotics projects due to its robustness and cost-benefit ratio. Its inclusion enables the robot to have proportionally controlled steering, unlike a simple on/off mechanism, thus achieving more precise maneuvers along the vehicle's route.

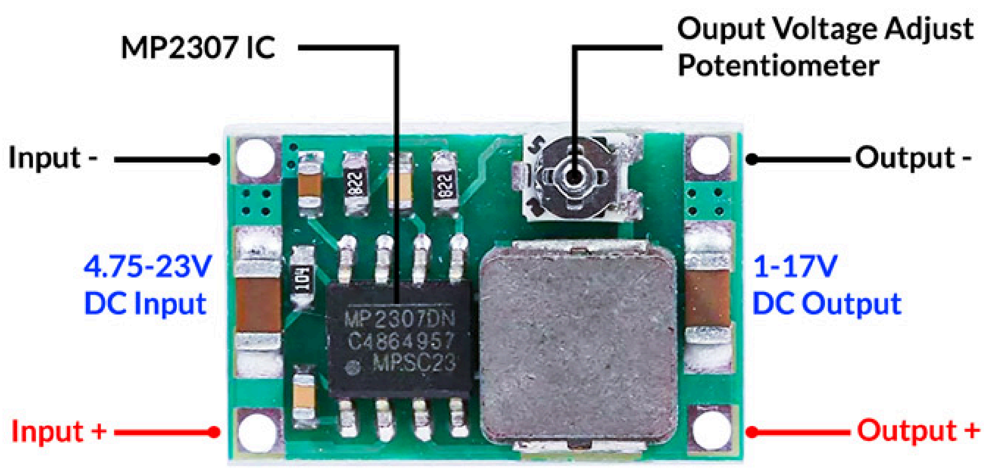
DRV8871 Motor Driver



To control the LEGO NXT traction motor from the Raspberry Pi/Arduino, a TI DRV8871 H-bridge DC motor driver mounted on a module is used. This controller is a monolithic H-bridge that supports up to 45 V motor voltage and provides up to 3.6 A peak current, more than enough to handle the NXT motor (whose typical consumption is below 1.5 A, with start-up peaks possibly near 2–3 A). The DRV8871 accepts logical inputs (IN1, IN2) of 3.3 V or 5 V to define the motor's rotation; these inputs can be modulated by high-frequency PWM to regulate speed. In our design, either the Raspberry Pi 5 (GPIO) or the Arduino Nano is used to send control signals to the driver, allowing control of the traction motor's direction and speed. A standout feature of this chip is that it incorporates programmable current limiting: via an external resistor, a current threshold is set, and the DRV8871 actively regulates to avoid exceeding it (no separate current sensors needed). This feature is valuable to protect the motor and prevent peaks that could reset the source; for example, the maximum current can be set to ~2 A, so that if the motor stalls or demands more, the driver will limit it, preventing damage.

The DRV8871 is powered directly by the 11.1 V LiPo battery on its VM pin (motor voltage). Its wide input range (from 6.5 V up to 45 V) allows it to work comfortably with a 3S LiPo. Internally, it consists of four power MOSFETs forming the full bridge, with a low internal resistance (~565 mΩ total), minimizing voltage drop and heat dissipation. The package incorporates a thermal pad to aid in heat dissipation; even so, given the intermittent use of the motor (the robot does not apply maximum continuous current for long periods) and the moderate current levels, the driver operates within safe ranges. Additionally, the DRV8871 offers a robust set of built-in protections: undervoltage lockout (UVLO), overcurrent protection (OCP), and thermal shutdown (TSD), with automatic retry when the fault condition disappears. This provides great safety, since in the event of a short in the motor or a prolonged jam, the chip will limit or cut the output, preventing a catastrophe. In summary, this driver acts as the power intermediary between the control logic and the traction motor, allowing bidirectional and safe control of the motor. In practice, its configuration on the board follows the schematic's wiring (battery → driver → motor) with low-resistance paths; adding bulk decoupling on VM is recommended as a future enhancement (see Opportunities for Improvement).

Mini360 DC-DC Converters (5.1 V Logic Rail and 6.0 V Servo Rail)

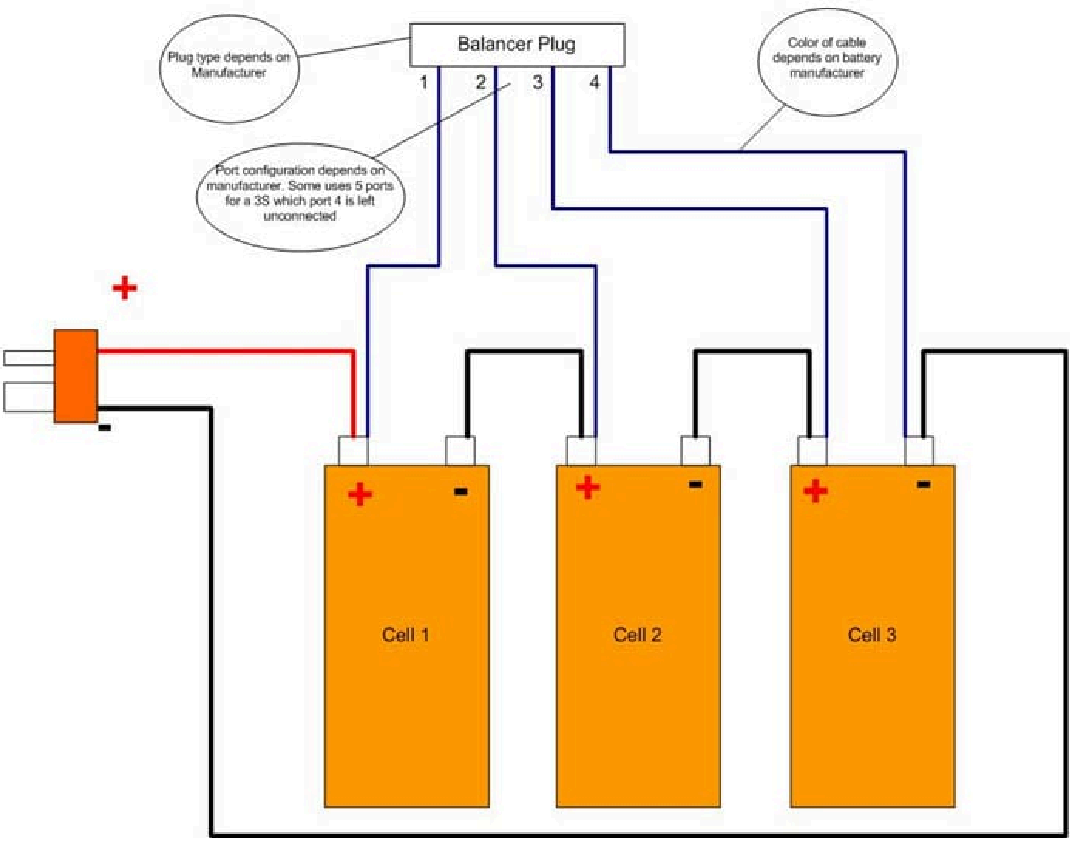


To obtain the required voltages from the main battery (11.1 V), the robot uses **two** Mini360 buck (step-down) DC-DC converters:

- **Mini360 @ 5.1 V (Logic Rail):** Feeds the Raspberry Pi 5, Arduino Nano, and low-power peripherals. The Mini360 module is based on the MP2307DN switching regulator, which offers high efficiency (up to ~95%) and accepts ~4.75–23 V input with an adjustable output of ~1.0–17 V via onboard potentiometer. In this design it is set to deliver **+5.1 V** to keep the Pi above 5.0 V under load. Operates around ~1.8 A continuous capability. Optional bulk decoupling on the 5 V bus and near the Pi is recommended—see Opportunities for Improvement.
- **Mini360 @ 6.0 V (Servo Rail):** Dedicated supply for the MG995 servo. This isolates servo demand from the logic rail. The converter accepts the same input range and is adjusted to **~6.0 V** to maximize MG995 torque within its rating. Local filtering at the servo connector is optional and recommended for a future revision—see Opportunities for Improvement.

Using switching regulators instead of linear ones greatly reduces heat when converting from ~11 V down to logic/servo voltages, improving runtime and avoiding large heatsinks. Both modules operate around ~340 kHz switching frequency.

LiPo 3S 5200 mAh Battery (11.1 V)

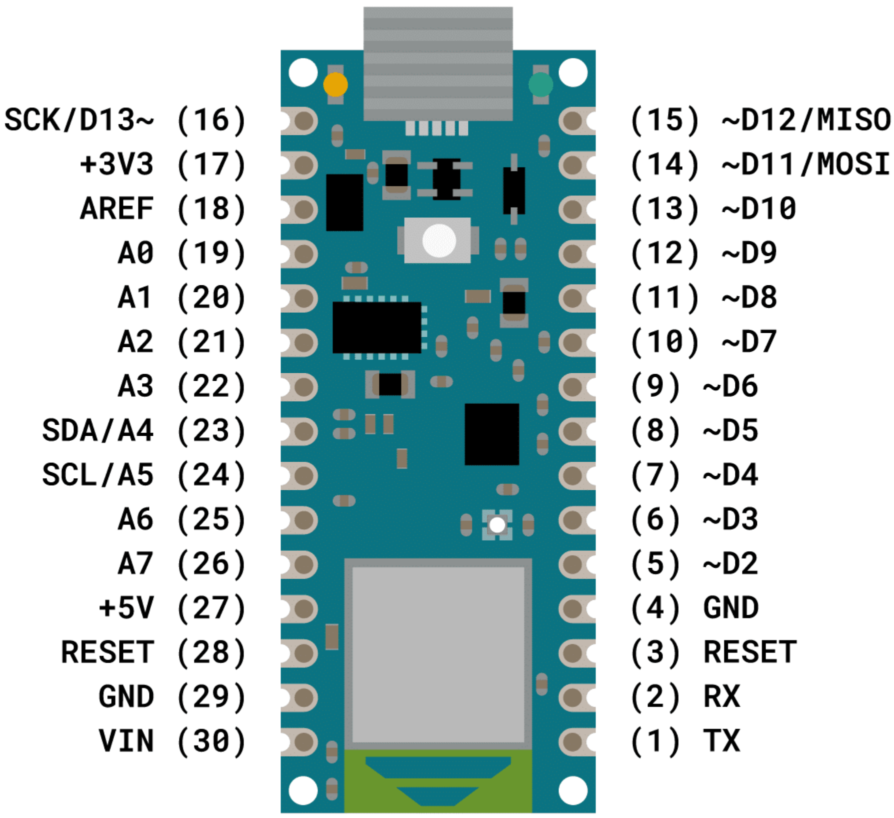


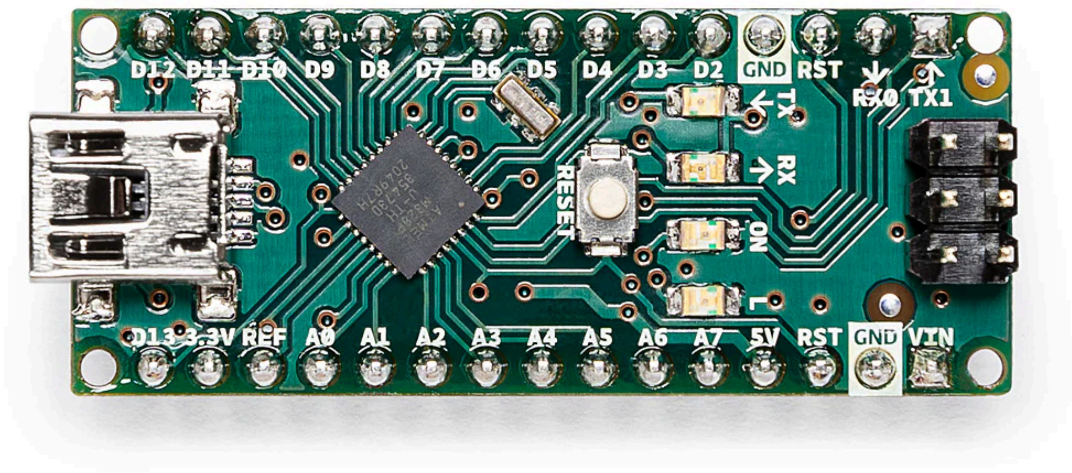
The robot's main power supply is a rechargeable lithium-polymer (LiPo) battery with 3 cells in series (3S), with a capacity of 5200 mAh and a nominal voltage of 11.1 V (each cell ~3.7 V nominal, totaling ~11.1 V). This type of battery is widely used in mobile robotics and modeling due to its high energy density and ability to deliver high discharge currents. A 3.7 V LiPo cell fully charged reaches 4.20 V; therefore, the 3S battery actually delivers around 12.6 V at 100% charge and drops to ~9.0 V when it is almost depleted (3.0 V per cell is usually considered the minimum safe threshold). In practice, the regulators and the motor driver are sized for the 9–12.6 V range, and the electronic system monitors the voltage to avoid over-discharge. With 5200 mAh, the battery can theoretically deliver 5.2 A for one hour, but its discharge capacity is usually indicated with a C rating (for example, 25C would mean up to 130 A instantaneous). In our robot, typical consumptions are on the order of 2–6 A, so the battery operates comfortably within its limits, providing sufficient autonomy for multiple competition rounds without recharging.

Given LiPo chemistry, safety precautions have been taken in its use. LiPo batteries are very efficient but sensitive to mishandling, and can ignite or explode if overcharged, short-circuited, or discharged below the minimum threshold. Therefore: (1) A suitable balancing charger is used that cuts off at 4.20 V per cell and balances the 3 cells. (2) In the robot system, the battery is protected with a main 10 A fuse in series at the output; this fuse acts as a safeguard against a short anywhere in the circuit, disconnecting the battery to prevent excessive currents. (3) The battery is connected to the rest of the system through a high-current master switch (SW2), which allows the robot to be turned off completely and also serves as a safety element (kill-switch). (4) A low-battery alarm was implemented (via software): the Arduino Nano measures, through a resistor divider, the battery voltage and alerts the Raspberry Pi when it drops below ~10 V, to proceed to stop operations and avoid over-discharge.

Physically, the 3S LiPo is mounted on the robot chassis with padding and zip ties, avoiding sudden movements or impacts. Its terminals are properly insulated to prevent accidental short circuits (for example, with high-current XT60 connectors). Additionally, the current-delivery characteristics were considered: the LiPo battery can supply bursts of tens of amperes, so the robot's main power lines (+11.1 V and ground) use appropriately gauged wiring (silicone 14 AWG) and lengths were minimized to reduce resistance and voltage drops. A good ground return from all subsystems to the battery is essential to avoid ground loops and potential differences. In summary, the 11.1 V 3S LiPo battery is the energy heart of the robot, and its high-capacity selection ensures that the robot has the necessary power during performance, while the protection measures guarantee safe and reliable operation.

Arduino Nano (Auxiliary Microcontroller)



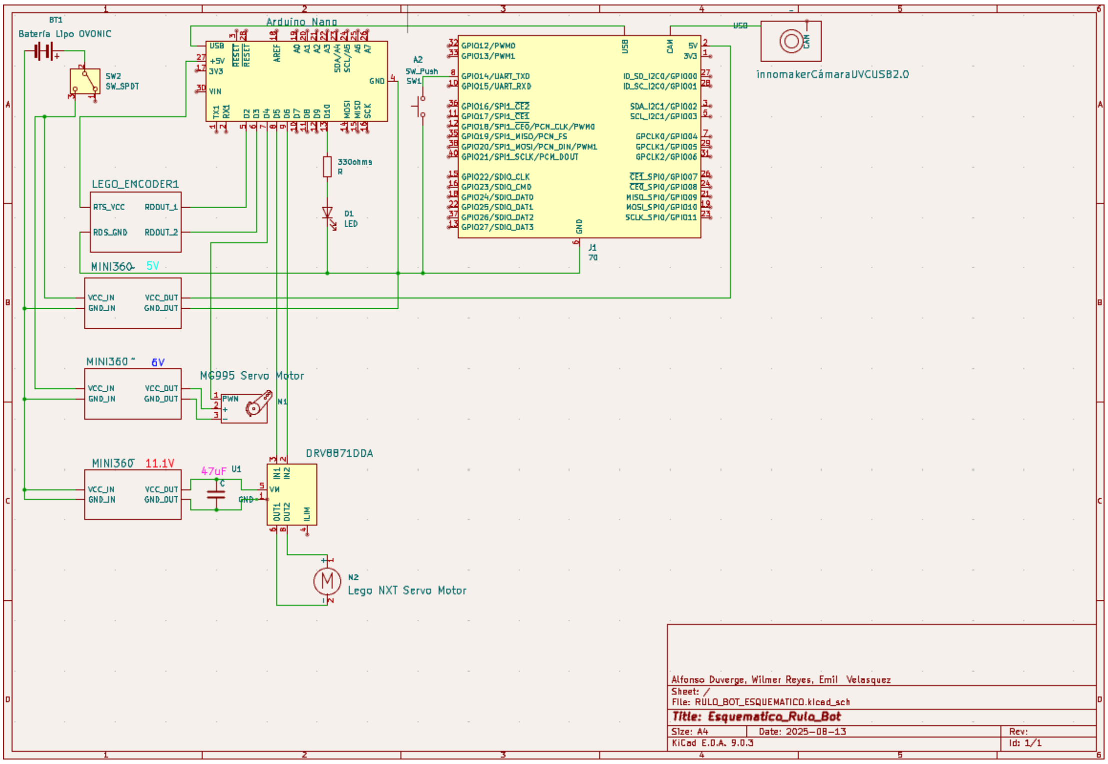


The Arduino Nano is the auxiliary microcontroller board incorporated into the robot's electronic system. It is based on the 8-bit AVR ATmega328P microcontroller running at 16 MHz, with 32 KB of Flash memory, 2 KB of RAM, and 1 KB of EEPROM. This small-sized board (approx. 18 × 45 mm) offers a considerable number of I/O pins (14 digital, 8 analog) and communicates easily with the Raspberry Pi via serial port (UART) or USB. In our design, the Arduino Nano performs sensor acquisition and real-time control functions that complement the Raspberry Pi. For example, it is in charge of reading the NXT motor encoders (counting A/B pulses with interrupts to obtain precise motion measurements) and generating the PWM control signal for the MG995 steering servo. This offloads time-critical tasks from the Raspberry Pi, since the Arduino (dedicated microcontroller) can handle microsecond interrupts and maintain precise timers for the servo PWM, while the Pi (running Linux) may not be as deterministic in response times. In addition, the Arduino can easily integrate other analog or pulse sensors (such as ultrasonics, gyroscope, etc. in future expansions) and provide filtered readings to the Pi.

In terms of power, the Arduino Nano operates at 5 V as the main logic and supply level. In the robot, it is powered directly from the 5.1 V logic rail (Mini360), connected to the Nano's 5V pin, thus sharing the same source as the Raspberry Pi. The board includes its own linear regulator in case it is powered via the Vin pin with 7–12 V, but in this case Vin is not used; instead, direct regulated 5 V is used, which improves efficiency and stability (by avoiding cascaded regulation). The Arduino Nano's consumption is very low (about 30–50 mA typically), so its impact on the 5 V load is minimal. All grounds (GND) of the Arduino, Raspberry Pi, driver, and others are connected to a common ground, guaranteeing a shared reference for the signals. Communication between the Raspberry Pi and the Arduino is done via serial UART (Arduino TX/RX pins to a USB adapter on the Pi, or directly to the Pi's GPIO UART pins), with 5 V↔3.3 V logic levels converted via a divider or interface, since the Pi uses 3.3 V on GPIO. Through this communication, the Pi sends commands (for example, to set the servo position or reset counters) and requests data (encoder counts, sensor status) from the Arduino. A user button is wired to **Arduino A2** (per schematic) for mode/kill or other functions in software. A status LED **D1** is included with a **330 Ω** series resistor for visual indication. In short, the Arduino Nano plays the role of real-time coprocessor, executing specific hardware-level tasks that complement the Raspberry Pi's high-level capability. Its inclusion adds redundancy and robustness to the control system, since even if the Pi experienced a momentary heavy load, the microcontroller can continue handling the basic aspects of motion temporarily.

Power Supply and Voltage Distribution System

Figure: *Block diagram of the robot's electrical system, showing the 11.1 V LiPo battery as the primary source, a master switch (SW2) and main 10 A** fuse, and two Mini360 DC-DC converters generating the **5.1 V logic rail** and the **6.0 V servo rail**. Power is distributed to the subsystems: Raspberry Pi, Arduino Nano, MG995 servo, and DRV8871 DC motor driver. Control connections (PWM and data signals) between the Pi, Arduino, driver, and actuators are also illustrated. A status LED (D1 + 330 Ω) and a pushbutton to Arduino A2 are included.*



The general electrical schematic (see figure above) is organized around the 11.1 V 3S LiPo battery, from which **three** main power rails are derived:

11.1 V Power Bus (unregulated): The battery connects through **SW2 (master switch)** and a **10 A fast fuse** to the DRV8871 that powers the traction motor. This bus carries the battery's full nominal voltage (between ~9 V and 12.6 V depending on the state of charge). Since the NXT motor is the only element that uses this high-voltage line, its route is short: from the battery to the driver and from the driver to the motor. Positioning and wiring minimize resistance and loop area. Optional bulk decoupling on the VM node is recommended as a future improvement.

5.1 V Regulated Logic Rail (Mini360): Takes input from the 11.1 V battery and delivers a **+5.1 V** output. This rail powers in parallel the Raspberry Pi 5 (5 V input), the Arduino Nano (5 V pin), and low-power peripherals (including the USB camera via the Pi). Power distribution uses a central 5 V/GND node with short, appropriately gauged wiring. Powering the Pi via the 40-pin GPIO (5 V/GND) avoids USB-PD negotiation; because the Pi is sensitive to voltage droop, cable length and connectors are kept robust. It is recommended to include a **~5 A fuse** on the branch feeding the Pi to emulate USB-C inlet protection. Optional bulk decoupling at the source and near the Pi is recommended —see Opportunities for Improvement.

6.0 V Regulated Servo Rail (Mini360): Delivers a dedicated **~6.0 V** supply for the MG995 servo. This physical and electrical segregation reduces coupling of servo current spikes into the logic domain. The servo ground ties to the common ground at the distribution point. Local filtering at the servo connector is optional and recommended as an enhancement in a future revision.

Grounding uses a practical “star” topology: high-current returns for the motor/servo go directly to the battery/fuse node, while logic returns (Pi/Arduino) return to the distribution point, reducing shared impedance and preventing resets or sensor errors.

Summary of voltages and power for each subsystem: The following table summarizes the sources and consumptions of the main components:

Component	Power Supply	Approx. Consumption	Notes
LiPo 3S Battery	11.1 V nominal (9–12.6 V range)	— (5200 mAh capacity)	Main source (up to ~100 A instantaneous discharge depending on C).
Mini360 (Logic)	In: 11.1 V; Out: **5.1 V**	Up to ~1.5–2 A (depending on load)	Feeds Pi + Arduino + low-power peripherals; optional bulk decoupling recommended (future).
Mini360 (Servo)	In: 11.1 V; Out: **~6.0 V**	Up to ~1.5–2 A (peaks as demanded)	Dedicated to MG995; optional local filtering recommended (future).
Raspberry Pi 5	5 V (from logic rail)	~1 A average (peaks 2–3 A)	CPU + peripherals; requires stable 5 V; consider ~5 A fuse on Pi branch.
USB Camera	5 V (via Pi USB)	~0.05 A (0.25 W)	Pi’s USB load; low consumption.
Arduino Nano	5 V (from logic rail)	~0.05 A	Logic module, negligible consumption.
MG995 Servo	~6 V (from servo rail)	~0.1–0.5 A (peaks ~1–2 A)	Current varies with load; isolated on its own rail.

Component	Power Supply	Approx. Consumption	Notes
LEGO NXT Motor	11.1 V (battery via DRV8871)	~0.2–1 A (peaks 2 A)	~9 V nominal motor; driver limits max current.
DRV8871 Motor Driver	VM=11.1 V, Vcc=5 V (logic)	Self-consumption <0.01 A; per motor	3.6 A peak capability; built-in protections; optional VM bulk decoupling recommended (future).
Status LED (D1 + 330 Ω)	From logic rail (5.1 V)	Few mA	Visual indication.
Pushbutton	→ Arduino **A2** input (logic)	—	User input (mode/kill) handled in software.

The table shows that the most critical loads for regulation are the Raspberry Pi (especially at CPU/USB peaks) and the servo when moving under effort. Therefore, the design ensures the regulators can supply the necessary current without the logic voltage dropping below ~4.9 V. If heavier use is anticipated (e.g., adding more servos or heavy sensors), further splitting or upsizing regulators can be considered (see Improvements).

Lastly, power-up/power-down management was considered. When connecting the battery, inrush current can occur; using the master switch (SW2) mitigates arcing. On shutdown, the procedure must be followed: command the Raspberry Pi to shut down via software (to avoid SD corruption), then cut the master switch so the Pi does not suffer a sudden blackout. Since the Pi 5 includes a power button on the board, that pin could be integrated into a control circuit for safe shutdown from outside.

Electrical Safety and Robustness Analysis

An electrical design for a competition robot must not only fulfill basic functions but also anticipate failures and ensure that in any eventuality the damage is contained. Below are the electrical safety, protection, and redundancy measures implemented, along with recommendations:

Short-circuit protection – Fuses: As mentioned, the battery's main line incorporates a fast-acting **10 A** fuse. This protects against direct shorts in cables or inside a module. Additionally, it would be advisable to add a dedicated **~5 A fuse on the Raspberry Pi branch**. Fuses not only protect components but also prevent more serious risks like cable overheating or fires—especially crucial with high-current LiPo batteries. Fuse selection must balance not tripping on normal peaks (e.g., a servo that draws 2 A briefly) but tripping on sustained faults.

Programmed current limiting: Beyond physical fuses, electronic limitations are leveraged. The DRV8871 has a shunt resistor that sets the motor's maximum current; adjusting this value ensures that even if the motor locks, it will not draw more than, say, 2.5 A, protecting the motor from overheating and the battery from excessive drain. Likewise, each Mini360 has an effective current ceiling; under overload it will sag or current-limit rather than fail. This acts as implicit protection: if downstream loads try to draw beyond the budget, the voltage drop limits further stress (not ideal operationally, but preferable to damage).

Separation of power channels (independent regulation): The design already separates logic (5.1 V) and servo (~6.0 V) rails, which reduces coupling of noisy, high-di/dt loads into sensitive logic. Further isolation (filters, dedicated grounds) can be added as an enhancement.

Decoupling and filtering (recommended upgrades): For improved noise immunity and transient handling, it is **recommended** to add distributed decoupling: bulk capacitors on the 5 V logic rail and near the Raspberry Pi, local filtering on the ~6 V servo rail (e.g., electrolytic + 0.1 µF), and bulk on DRV8871 VM. TVS diodes (≈15 V on the battery line, ≈5.6 V on the 5 V rail) and ferrite elements on motor/servo leads are also **recommended** enhancements. These items are **not currently populated** per the provided schematic and are listed as **future improvements**.

Wire gauge and safe connectors: All cables that handle significant current (battery, motor, main 5/6 V rails) are of adequate gauge to avoid heating. XT60 connectors are used for the battery (rated up to 60 A continuous), robust terminal blocks for regulator outputs, and Dupont headers only for logic signals. The motor and battery cables are soldered or crimped firmly, minimizing contact resistance. Cable restraint prevents inadvertent shorts.

Redundancy and fault recovery: Using an Arduino Nano as a coprocessor adds redundancy: if the Raspberry Pi hangs or restarts, the Arduino can detect the loss of communication and, for example, brake the motor or center the steering for safety. Component choices (metal-gear MG995, robust LEGO motor, the Pi 5's PMIC) contribute to resilience. Software watchdogs complement hardware (e.g., if no command is received within X time, the Arduino cuts PWM to the driver).

Maintenance and periodic verification: Before each round, the team checks voltages with a multimeter: the battery ~12.5 V, the logic rail 5.1 V, the servo rail ~6.0 V, etc. Loose connections or pinched cables are inspected. A real-time voltage monitor (Arduino reports the battery voltage to the Pi during the run) detects unexpected drops and helps diagnose regulator issues or abnormal consumption. Electronics are fixed with standoffs and screws to resist vibration.

In summary, Team Rulo Bot's electrical system incorporates multiple levels of protection: fuses, electronic limitation, good wiring practices, isolation of actuator power, and sound grounding. These measures ensure that in the event of overloads or interference, the robot continues operating stably or fails safely (fail-safe) by stopping its movements. Electrical safety is critical not only for the robot's longevity but also for integrity during competition, avoiding risks to operators and other teams.

Technical Conclusions and Opportunities for Improvement

The development of the electrical and electronic system described has successfully provided the Team Rulo Bot with a reliable power platform and a capable electronic "brain," meeting the demanding challenges of WRO Future Engineers 2025. In tests and competitions carried out, the proposed power distribution kept operating voltages stable even during intensive maneuvers (sudden accelerations, quick turns) that imply consumption peaks. The integration between the Raspberry Pi 5 and the Arduino Nano proved effective: the Pi focuses on vision processing and strategy, while the Arduino ensures accurate encoder readings and smooth servo control, thus achieving closed-loop control over the robot's trajectory. Components such as the DRV8871 driver and the Mini360 converters have shown performance in line with specifications, without notable overheating since they operate within safe margins (supported by built-in protections). The choice of a 5200 mAh 3S LiPo battery provided an optimal balance between capacity and weight, offering sufficient autonomy and delivering the current needed for the actuators without worrisome voltage drops.

Nevertheless, every design can be improved. Below are identified improvement opportunities and future considerations that could increase the robustness and efficiency of the electrical/electronic system:

- **Higher-capacity or dual 5 V source:** While the Mini360 for logic fulfilled its role, future iterations could use a higher-current regulator (e.g., a 5 V 5 A UBEC). Alternatively, keep the present two-rail approach but upsize the logic converter and/or add a third converter for additional actuators/sensors. One could even explore using the Pi 5's USB-C PD port with a small PD negotiation board that delivers 5 V 5 A from the battery with high efficiency; some expansion boards accept 12 V input and negotiate PD 5 V 5 A for the Pi, cleanly solving the power supply within official specifications.

- **Advanced battery monitoring:** Incorporate a dedicated voltage and current sensor (such as an INA219 or similar) in the battery line to provide real-time consumption data, enabling energy-strategy optimization and anomaly detection (e.g., unexpected friction). This can drive behaviors (reduce speed on low SOC) and yield accurate remaining-battery estimation. A low-voltage cut-off circuit (e.g., MOSFET disconnect below 3.0 V/cell) can protect the pack during extended testing.

- **Add bulk decoupling on rails (recommended):**
  - Logic rail 5.1 V: ≥470–1000 µF at the source, plus ≥470 µF local near the Raspberry Pi.
  - DRV8871 VM: ≈47–100 µF low-ESR close to VM/GND.
  - Servo rail ~6.0 V: ≈220–470 µF + 0.1 µF ceramic at the MG995 connector.
- **EMC protections (recommended):** TVS diodes (≈15 V on battery bus; ≈5.6 V on 5 V rail), ferrite beads/rings on motor and servo leads, and optional small chokes on the servo supply.

- **Improvements in wiring and connectors:** Replace provisional connections with standardized secure connectors where helpful (e.g., Deans/MT60 for motor, XT30/MicroFit for board-level power distribution). Continue optimizing routing to separate power and signal paths; shield the USB camera cable path from inductors/motors if interference ever appears.

- **Include IMU or other sensors with proper isolation:** Add an IMU (gyro/accel) for better navigation. Feed it from a very clean 3.3/5 V with its own small filter. For ultrasonic sensors, consider a tiny local linear regulator and careful grounding to minimize ripple sensitivity.
- **Optimization of weight and space (electronics):** Consider a custom PCB "power/IO distribution board" that integrates the two Mini360s, fuses, connectors, and indicator LEDs. This reduces loose wiring, unifies grounds, minimizes parasitic inductances, and aids maintainability.
- **Cooling improvements:** If prolonged operation shows heating in any component (regulators, driver, Raspberry Pi), add heatsinks or a small fan. The Pi 5 has an official fan kit; the DRV8871 and Mini360s can also benefit from adhesive heatsinks if driven near limits.
- **Event logging:** Log low-voltage warnings, current limits, watchdog events, and any fuse trips. This feedback helps tune current limits and validate thermal margins.

In conclusion, the current electrical-electronic system of Team Rulo Bot has proven to more than meet the project's needs, offering safe power and control. The proposed improvements aim to reinforce reliability and prepare for future expansions, ensuring that the robot can adapt to greater challenges while maintaining electrical integrity. This technical report will serve as a reference both for judges and for the team itself in future iterations, facilitating understanding of the design and promoting good engineering practices in the competition. Each component and decision described here reflects a technical criterion oriented toward stable performance and safety, in line with the spirit of the Future Engineers category, where not only the result is evaluated but also the process and the solidity of the robot's construction.

References

Technical documentation of components and relevant datasheets, for example: Specifications of the Tower Pro MG995 servo (sandorobotics.com.mx), Texas Instruments DRV8871 motor driver application notes (ti.com, ti.com), and LEGO Mindstorms NXT reference manuals (lagos.udg.mx) for the motor with encoder.

Best-practice guides in embedded systems, including advice from the Raspberry Pi community in Spanish on correct powering of the board (forums.raspberrypi.com).

Electronics articles and resources (e.g., Components101) on the Mini360 module and its current capability (components101.com), as well as manufacturer information for the UVC camera used (spanish.module-camera.com) and the LiPo battery (cell characteristics and care) (dynamoelectronics.com).

Practical experience of the team and laboratory tests carried out during development, which complement theory with empirical data.

In [ ]: