Proposal for Original Features

B17

February 13th, 2025

1 Rationale

The additional features in our self-balancing robot project are designed to enhance functionality, safety, and interactivity. The inclusion of a Raspberry Pi Zero facilitates data collection, enabling a camera feed to be displayed on the app. Our design adds a color sensor (VEML3328) to make the robot act differently as it detects different colors near it. A distance sensor (SNS-HCSR04) ensures collision avoidance, enhancing the robot's safety by stopping it from getting too close to obstacles. The RFID tag provides a security layer, preventing unauthorized operation by requiring the correct tag to be scanned. These features work together to create a more intelligent and reliable self-balancing robot.

2 Feature Goals, Requirements, and Constraints

2.1 Feature Goals

The proposed features aim to enhance the functionality, interactivity, and security of the self-balancing robot. The main goals include:

- **Obstacle Avoidance:** Utilize an ultrasonic distance sensor (SNS-HCSR04) to detect obstacles and prevent collisions by stopping or altering the robot's path.
- **Color Detection:** Implement a VEML3328 color sensor to allow the robot to respond differently based on detected colors.
- **RFID Authentication:** Add an RFID module to ensure only authorized users can control the robot via an app.
- **Camera Integration:** Use a Raspberry Pi Zero to process and transmit a real-time camera feed to the app, enabling remote visual monitoring.
- **Reliable Communication:** Establish robust communication between the Arduino Nano 33 BLE and Raspberry Pi Zero for seamless sensor data processing and command execution.

2.2 Requirements

2.2.1 Functional and Performance Goals

- The SNS-HCSR04 must reliably detect obstacles within a 5–200 cm range and stop the robot within 3 cm of an obstacle.
- The VEML3328 must differentiate between at least three predefined colors (e.g., red, blue, green) and trigger a response within 200 ms.
- The RFID module must authenticate an RFID tag within 1 second and prevent unauthorized operation.
- The Raspberry Pi Zero must process and transmit a camera feed at least 1 FPS
- The Arduino Nano 33 BLE must correctly interpret sensor inputs and send necessary commands to the Raspberry Pi Zero and motor controller.
- The robot must maintain stability while executing commands and responding to sensor inputs.

2.2.2 Technical Requirements

- The Arduino Nano 33 BLE must handle real-time sensor data acquisition and control outputs while communicating with the Raspberry Pi Zero.
- The Raspberry Pi Zero must process image data and handle high-level computations without excessive delay.
- Communication between the Arduino and Raspberry Pi should be implemented via UART, SPI, or I2C, ensuring reliable data transfer.
- Power consumption should be optimized to allow the robot to function for at least 20 minutes on a full charge.

2.3 Constraints

2.3.1 Financial

The financial constraints for the extra features are as follows,

1. There is a maximum of $65 (not including shipping and taxes) which can be used for extra features
2. The $65 will only be paid if it is through Digikey
3. Any "out of pocket" expenses will not be paid for by the ECE department
4. Any hardware provided in the second-year kits does not count for the $65 budget

2.3.2 Hardware

1. All the circuitry must be powered by 8 1.5V 1.9Ah batteries (12V 1.9Ah)
2. Must use the Arduino Nano 33 BLE Rev2 (for gyroscope and accelerometer), which comes with 64kHz clock speed, 1MB flash and 256kB RAM

3 Accomplishments and Visual Aid

Describe what your feature(s) will accomplish. Include a sketch or other visual representation to convey the idea.

- **Raspberry Pi Zero:** Send user commands to Arduino microcontroller and acts as a lightweight local server for a software user interface with signal plotting and live camera feed.
- **Freenove 5MP Camera:** Captures visual feed of robot surroundings to enable user feedback.
- **STM32 Microcontroller:** Reads sensors to respond to environment conditions and send commands to Arduino microcontroller.
- **VEML3328 RGB Color Sensor**: Scans set color combinations to facilitate predefined robot response.
- **HCSR-04 Distance Sensor:** Ensures safe navigation by measuring the distance to nearby objects and enabling obstacle detection.
- **RFID Module:** Prevents unauthorized access to robot operation.
- **LCD Display:** Provides basic visual feedback of system diagnostics.
- **Speaker:** Produces audio alerts for critical events, including authentication and environment detection.

4 Functional Diagram

We have included a high-level functional diagram for the end-user operation and control of the robot. Note that these include their experience through controlling the base device, with the additional functionality.

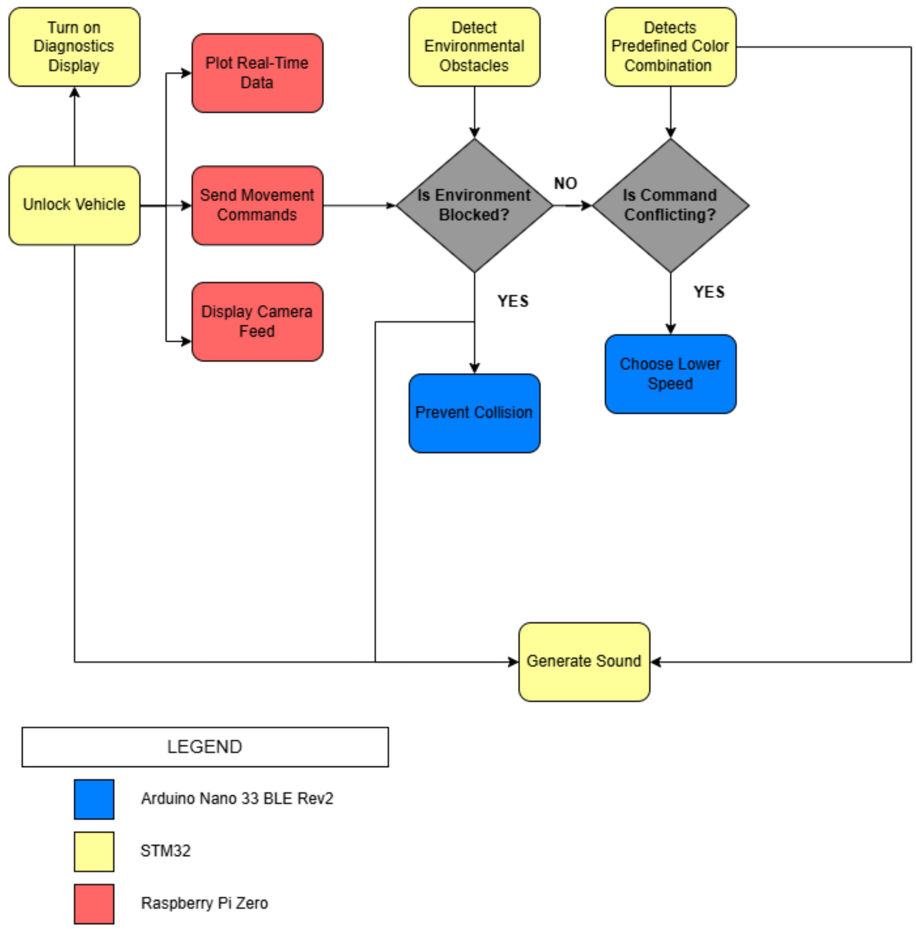

Figure 1: High-level Functional Diagram

4.1 Workflow

The user must follow the workflow to unlock the device, respond to sensor feedback, and follow constraints imposed by the system logic. An ideology this adheres to are our safety principles. If an obstacle is detected, the robot system will prevent commands from leading to collisions. Likewise, conflicting commands for movement (i.e. from Raspberry Pi hosted user interface and the color sensor) will cause the output to default to the command with slower speed.

This ensures that:

- A firmware-defined "red light" is respected at all times.
- A speeding robot can always be slowed down by the user.

5 Architecture Diagram

We have designed the electrical interfaces so the project components interact as shown in the following architecture diagram.

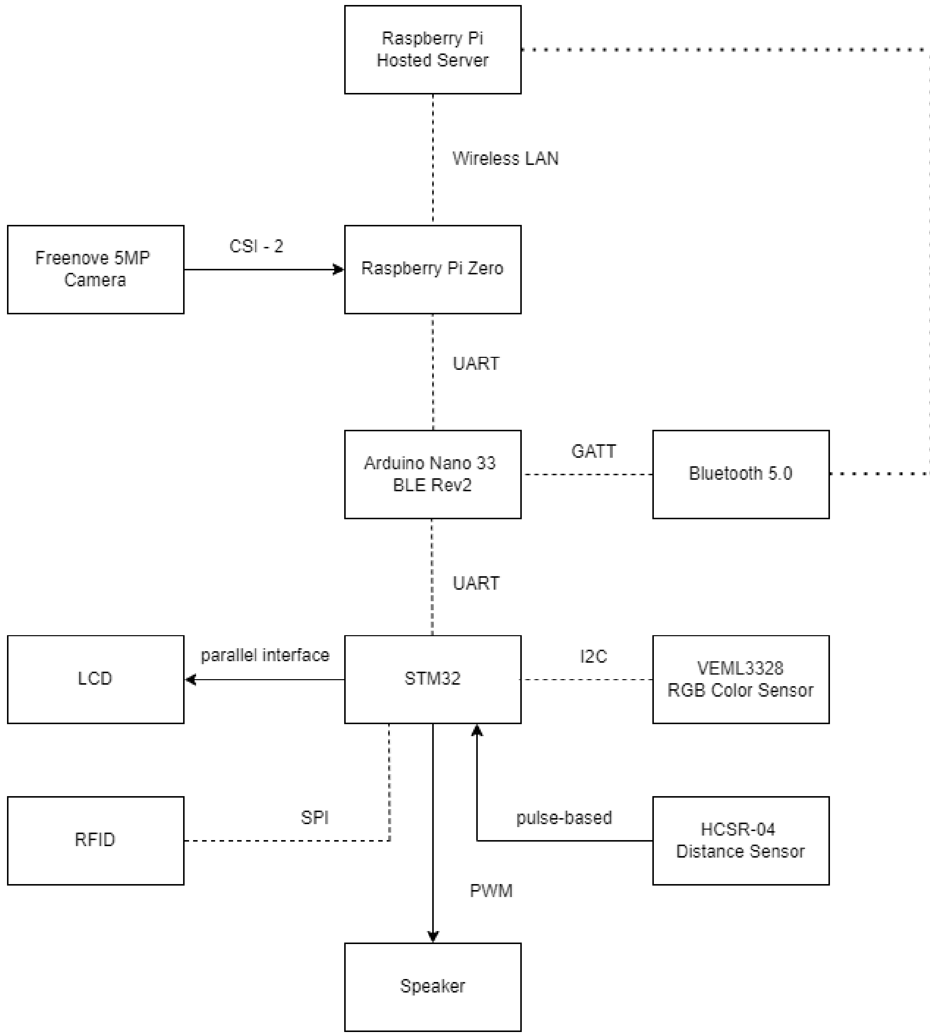

Figure 2: Architecture Diagram

Note that both wired and wireless communication interfaces are used to transmit information between our three boards.

- The Arduino Nano 33 BLE Rev2 acts as the robot controller and transmits diagnostic data to the adjacent boards (for display).
- The STM32 acts as the peripheral interface to collect raw data and provide commands based on environment input.
- The Raspberry Pi Zero is our dedicated on-board computer for post-processing and hosting the software application for the user-interface.

6 Criteria for Achieving Success

The success of the proposed features will be evaluated based on the following measurable criteria:

6.1 Performance Benchmarks

- **Obstacle Avoidance:** The distance sensor (SNS-HCSR04) should reliably detect obstacles within a 5–200 cm range, stopping the robot within 3 cm of an obstacle.
- **Color Detection:** The VEML3328 color sensor should accurately differentiate between at least three predefined colors (e.g., red, blue, and green) and trigger a corresponding response in the robot's behavior within 200 ms of detection.
- **RFID Authentication:** The RFID module should authenticate a valid tag within 1 second and deny access to invalid tags.

6.2 Functional Correctness

- The Raspberry Pi Zero should successfully process and transmit camera feed data to the app without excessive lag (less than 1 second delay).
- The Arduino Nano 33 BLE should correctly interpret sensor inputs and send relevant commands to the Raspberry Pi Zero and motor controller.
- The robot should remain balanced while responding to external commands and sensor inputs.

6.3 Reliability

- The system should function continuously for at least 5-7 minutes without requiring a reset or encountering communication failures.
- If an unexpected sensor failure occurs, the robot should enter a safe mode (e.g., stopping motion) rather than behaving unpredictably.
- Power consumption should allow the robot to operate for a minimum of 10 minutes on fully charged batteries.

7 Demonstration to Teaching Staff

The following features will be demonstrated to the TAs, with validation conducted as follows:

- **RFID Authentication:** On startup, the app's input controls should remain disabled until an RFID tag is scanned and successfully validated.
- **Obstacle Avoidance:** The robot will be placed near a wall and intentionally driven towards it. It should detect the obstacle and stop to prevent a collision.
- **Color Detection:** A specific color (e.g., red) will be presented to the robot. When driven towards the color, it should halt until the color is no longer detected.
- **Communication & Processing:** The Raspberry Pi Zero should receive data from the Arduino Nano 33 BLE. It will also perform additional computations such as capturing images, _____, and displaying relevant information on the app.
- **Feature Verification:** Each test will include a clear indication of success, such as a specific message displayed on the LCD or an audio signal from a speaker.

8 Accompanying Documentation

The following documentation will be provided to support the implementation and operation of the proposed features:

- **Datasheets:** Technical references for the key components used in the project:
  - Arduino Nano 33 BLE Rev2 Datasheet
  - Raspberry Pi Zero Datasheet
  - VEML3328 Color Sensor Datasheet
  - SNS-HC-SR04 Distance Sensor Datasheet
  - RFID Module Datasheet
- **Other Documentation:** Additional references for software development and library usage:
  - Arduino Library Documentation