IoT Enabled Autonomous RC Car

SE - 322

saleh al-ghannam - 2203

2024

**Technical Report: IoT-Enabled RC Car**

Introduction The IoT-enabled RC car project aims to demonstrate the capabilities a simple micro controller can have on a standard RC car by integrating a Pi Pico, ultrasonic sensors for obstacle avoidance, IR sensors for line tracking, and Bluetooth communication for remote control. The project demonstrates the potential of IoT and robotics in consumer applications.

System Overview

- Hardware components-

* Pi Pico microcontroller
* N20 motors
* IR sensors
* Ultrasonic sensors
* OLED screen
* Bluetooth module

-Software Components-

* Motor calibration logic
* Line tracking algorithm
* Obstacle avoidance algorithm
* Bluetooth communication protocol

[](https://github.com/misae717/IoT-RC-Car)

[***Repo***](https://github.com/misae717/IoT-RC-Car)

Implementation Details

-Motor Calibration-

The code performs an automated calibration of the motors at startup. - The wheel circumference is calculated based on the wheel diameter (44mm). - The approximate RPM of each motor is measured by timing the completion of a fixed distance. - The calibration logic adjusts the motor speeds based on these approximations, however this turned out to propose no value to the overall functionality of the car as the motors themselves were failing and no calibration or lubrication would help syncing up both motors to each other, eventually the motors were replaced and the gear ratios were adjusted to a much higher value contributing greater torque and stability at the cost of speed.

- Line Tracking-

IR sensors are used to detect high-contrast lines on the ground. - The Pi Pico processes the sensor data and adjusts the car's steering and speed to follow the line. - The line tracking algorithm is on the TO-DO list of stuff to refactor and improve in the future, currently the algorithm is functional but due to how the IR sensor grid is placed certain situations can cause the car to lose control and be stuck in a constant loop of adjustments, a software solution in the form of a delay was introduced to ease the issue, however any higher values on the delay would introduce inaccuracies in the line tracking and reducing overall functionality, TO-DO is to build a new IR grid array with the center sensors close to the ground and increase the distance between the edge and center sensors, this hardware solution will increase the accuracy and help mitigate edge scenarios where the tracking can run into failures as-well as increasing flexibility at the software level introducing different algorithms which can be introduced to deal with certain edge cases.

-Obstacle Avoidance-

Ultrasonic sensors continuously emit sound waves and measure the echo return time. - The Pi Pico calculates the distance to nearby obstacles based on the sensor data. - The obstacle avoidance algorithm adjusts the car's trajectory in real-time to navigate around detected obstacles, this currently provides many inaccuracies as any obstacle in the way no matter how small would give a reading, in the future this is aimed to be replaced with a LiDAR camera since unfortunately this is an inherent characteristic of the ultrasonic sensor. Along with the improvement and upgrade of the microcontroller, a pivot into the newer AMD embedded processors will greatly increase the processing power and memory, the current limitations of the Pi Pico 266KB of ram are forcing heavy constraints on the potential of the project.

-Bluetooth Remote Control-

A Bluetooth module is integrated into the RC car. - A mobile application or computer interface is developed to send control commands to the car via Bluetooth, currently a standalone application is being used, any application that can communicate with the Bluetooth module is supported as long as the entry points for the functions match as the code is structured in decoupled modules. - The Pi Pico receives the commands and translates them into appropriate motor control signals or entries into one of the autonomous modes.

-OLED Display-

An OLED screen is connected to the Pi Pico. - The display shows relevant information such as sensor readings, distance measures, and relevant data for operating modes. - The OLED screen enhances the user experience and provides real-time feedback, and can be freely adjusted to provide real-time feedback to the user.

Results and Testing

* + The motor calibration process ensures accurate and synchronized movement of the motors of the RC car if the motors are not malfunctioning.
  + Line tracking functionality has been successfully implemented, allowing the car to follow predefined paths autonomously.
  + Obstacle avoidance has been tested in various scenarios, demonstrating the car's ability to detect and navigate around obstacles.
  + The Bluetooth remote control feature enables seamless user control of the car's movement and functions.
  + The OLED display accurately displays relevant information, enhancing the user interface.

Conclusion and Future Work The IoT-enabled RC car project successfully integrates a Pi Pico microcontroller, sensors, and wireless communication to enhance the capabilities of a traditional RC car. The implemented features, including line tracking, obstacle avoidance, and Bluetooth remote control, demonstrate the potential of IoT and robotics in consumer applications. Future work may include the integration of advanced sensors like LiDAR and the usage of more advanced microcontrollers and the replacement of Bluetooth with high speed WiFi, the incorporation of machine learning algorithms for improved decision-making and full autonomy, and the expansion and design of the car's chassis and PCB for specific application domains such as environmental monitoring or autonomous vehicle development. The project serves as a foundation for further exploration and innovation in the field of IoT and robotics, providing valuable insights and a hands-on learning experience for enthusiasts, students, and researchers alike.



