KLE Technological University
Creating Value, Leveraging Knowledge

Dr. M. S. Sheshgiri Campus, Belagavi

KLE TECHNOLOGICAL UNIVERSITY

Raspberry Pi Camera Tank

Department of Electronics and Communication Engineering

Name and USN

Amogh M. Gadad	02FE22BEC006
Amruta R. Biradarpatil	02FE22BEC008
Ashtami Hosapeti	02FE22BEC012
Naman Timmapur	02FE22BEC039

Under the Guidance of

Prof. Anita Shirol

Department of Electronics and Communication Engineering

Contents

1 Abstract

The Raspberry Pi Camera Tank is a versatile, remote-controlled robotic vehicle designed for live video streaming and real-time surveillance applications. Built using a Raspberry Pi single-board computer, a camera module, and a motorized tank chassis, this project integrates wireless control and video feed streaming over a local network or the internet. Key features include motor drivers for precise movement, a portable power supply, and an intuitive web interface. This project serves as an educational platform for embedded systems, robotics, and IoT applications.

2 Introduction

The Raspberry Pi Camera Tank is an ingenious robotics project that combines vehicle mechanics and live streaming, thereby making it a feasible system for many uses. Fundamentally, the system harnesses the capable Raspberry Pi single-board computer which serves as the control center for the tank, controlling its actions, streaming video feeds and operating superior features.

Being an improved version with a motorized tank chassis and a camera module, users can control the vehicle through a web interface wirelessly. To manage the motors and hierarchically control other peripherals, Raspberry GPIO Pi pins are employed so that Pi can be adapted easily as per requirements.

This project can be most useful in home security; the tank can oversee regions of a home that are normally inaccessible. Also, its mobility to all weathers assists it to undertake the exploration work, including investigation of dangerous or isolated region. The educational value of the tank is as important, as people get the experience in embedded systems, robotics, and computer vision within the base of students and hobbyists.

In general, the integration of the Raspberry Pi Camera Tank Hardware and Raspberry Pi Camera Tank Software & Applications make it possible to deliver a flexible and expandable platform for complex robotic ideas. The potential for future improvements, including self-controlled mobility and AI aided object recognition, will only amplify broaden its capabilities, making for an enticing project regardless of the expertise level of the developer, applying to robotics and IoT.

3 Literature Survey

3.1 Remote-Controlled Robots

Remote controlled robots refer to machines that are controlled by other individuals with out having any direct operator input. A lot of research concerns the evolution of remote operated vehicles or ROVs. Hasan et al. [1] has built a web-controlled Robot utilizing Raspberry Pi and Flask FW. The system allowed for remote navigation and live streaming, however it did not contain an integrated firing mechanism or high level motor control options. This project is an extension of these inputs and adds further functionality into the system.

3.2 Video Streaming in Robotics

Video streaming in real-time is a vital aspect in the advanced robotics to monitor and to supervise. Kumar et al. [2] employed OpenCV and MJPEG streaming for capturing and detecting real-time video in a robot. The main Critique which was apparent with their approach was that for shorter distance it did work well but due to missing multi- threading their response time was not satisfactory. In this paper, the proposed system has incorporated the threading in an attempt to enhance the frame capture and streaming.

Motor control using GPIO Motor control without the use of PWM signals is achievable using GPIO. One of the key issues in robotics is motor control Each robotic system includes some sort of motor control. The work done by Gupta et al. [3] employed the L298N motor driver to manage the movement of DC motors in terms of their direction and velocity by the aid of PWM signals produced by Raspberry Pi. Points such as the significance of accurate PWM control towards smooth motion as adopted in this project were well illustrated by their discovery.

3.3 Considered Integration of firing mechanisms

To date, there is little research undertaken into the incorporation of electromechanical firing systems into robots. Furthermore, in Wang and colleagues' work, they proposed a GPIO signal-controlled firing mechanism in [4]. This though was good, it offered little flexibility in control of the direction. These considerations form the basis of this project that seeks to allow for two-way interaction with the firing motor.

3.4 Web-Based User Interfaces

Web-Based User Interfaces are reference models that are applied for executing and displaying the user interfaces of interactive systems on the World Wide Web (WWW) technology. The application of Web interfaces for surveying has been researched before. Alam et al. [5] developed a Flask based web application for autonomous robot navigation with focus on non-complexity and minimal delay. However, their system did not include complex control functions such as being able to remotely control the filming of an image or a firing system. This work builds on their research by offering a fuller user interface for the program.

4 Functional Block Diagram

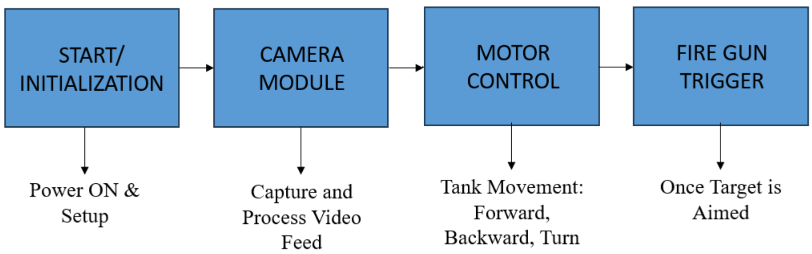

Figure 1: Functional Block Diagram of Raspberry Pi Camera Tank

4.1 Explanation of Blocks

- **Start/Initialization**: This block represents the startup process of the system. It involves powering on the Raspberry Pi, initializing hardware components, and setting up the software environment. This ensures all modules, such as the camera and motor drivers, are ready for operation.
- **Camera Module**: The camera module captures the live video feed, processes it in real time, and sends it to the target tracking algorithm. The video is essential for identifying and tracking targets, as well as for remote surveillance.
- **Target Tracking Algorithm**: This block detects and tracks the target using the video feed. It identifies objects of interest, calculates their location, and plans the path for the tank to approach or follow the target.
- **Motor Control**: Motor control manages the tank's movement based on the target tracking algorithm's path planning. It uses GPIO pins and PWM signals for precise control to move the tank forward, backward, or turn.

- **Fire Gun Trigger**: Once the target is properly aligned, this block activates the firing mechanism, triggering the gun to fire at the identified target. The action is controlled through the Raspberry Pi's GPIO interface.

5 Design Specification

5.1 Input

- **Webcam**: USB conglomerated video feed with live streaming for distant surveillance and imagery.
- **User Input (Flask Web App)**: Movement, firing mechanism, and frame capturing controls through internet connection.

5.2 Output

- **DC Motor**: Motor control of GPIO for the movement of the robot in forward, backward, left, and right directions.
- **Gun Firing Mechanism**: Electromechanical system designed to be fired by GPIO signals, with the ability to control direction of the motor.
- **Live Video Stream**: MJPEG format video feed for the purpose of real-time remote surveillance.
- **Image Capture**: Motion capture data recorded by the camera that can be downloaded through the web panel.

5.3 Protocols

- **GPIO Pins**: Control the direction, speed, and firing mechanism of the robot.
- **PWM**: Regulates the speed of a motor and maintains the accurate direction of the vehicle.
- **HTTP (Flask Framework)**: Supports user interaction by including web-based commands and video streaming support.
- **MJPEG Streaming**: Live video preview transmits over HTTP to allow for real-time surveillance.

6 Implementation

The implementation of this project is divided into three main components: hardware setup, software development, and integration of hardware with the Flask-based web application. Each component is designed to ensure real-time functionality and user interaction.

6.1 Hardware Implementation

The robot's hardware consists of the following components:
1. **Raspberry Pi**: Serves as the central processing unit for the system, handling motor control, video streaming, and user commands.
2. **L298N Motor Driver**: Controls the movement of two DC motors based on GPIO signals from the Raspberry Pi. The driver is connected to the robot's wheels for locomotion.
3. **DC Motors**: Provide forward, backward, left, and right movement. Pulse Width Modulation (PWM) signals regulate their speed and direction.
4. **Firing Mechanism**: An electromechanical system triggered by GPIO signals controls the firing motor for tactical operations.
5. **USB Camera**: Captures live video frames for remote monitoring. The camera is interfaced with the Raspberry Pi via the OpenCV library.

6.2 Software Implementation

The software stack is built to facilitate interaction between the hardware and the user:
1. **Flask Framework**: Provides the web interface for controlling the robot. The interface includes buttons for movement, firing, and image capture, as well as a video feed.
2. **OpenCV**: Handles video streaming by capturing frames from the USB camera and streaming them in MJPEG format over HTTP.
3. **RPi.GPIO Library**: Manages the GPIO pins on the Raspberry Pi, enabling motor control and firing mechanism operation.
4. **Multithreading**: A separate thread continuously captures frames from the camera to ensure efficient video streaming without interrupting other processes.

6.3 Integration

1. **Motor Control**: The Flask web application defines routes for controlling the motors (/left_side, /right_side, /up_side, /down_side, /stop). These routes send GPIO signals to the L298N motor driver, enabling movement.
2. **Firing Mechanism**: The /fire route triggers the firing mechanism by sending GPIO signals to the motor controlling the gun.
3. **Live Video Streaming**: The /video_feed route streams live video by continuously capturing frames and encoding them in MJPEG format. The video feed is accessible via the web interface.
4. **Image Capture**: The /capture_image route saves a snapshot of the current frame to the Raspberry Pi's storage, which users can download through the web interface.

6.4 System Workflow

1. The user accesses the web interface hosted by the Flask server.
2. Commands for robot movement, firing, or image capture are sent to the Raspberry Pi via HTTP requests.
3. The Raspberry Pi processes the commands, sends GPIO signals to the motor driver or firing mechanism, and streams live video to the user.
4. The user receives real-time feedback, including the robot's live location and the captured images.

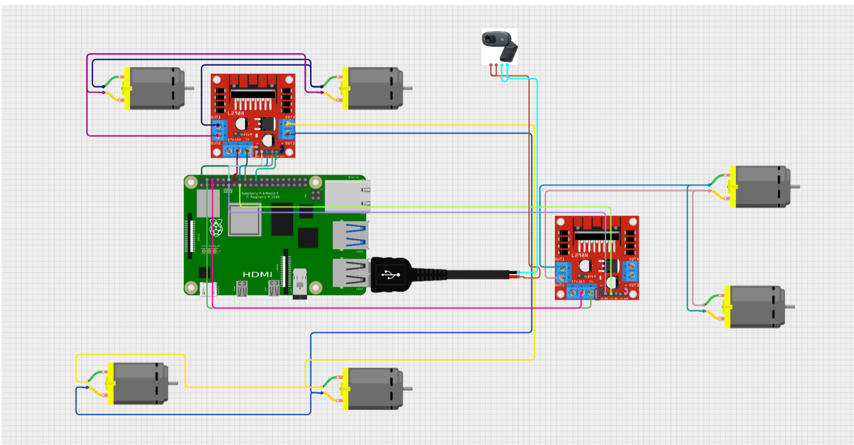

Figure 2: Schematic of Raspberry Pi Camera Tank

7 Results

The Raspberry Pi Camera Tank successfully demonstrated:

- Real-time video streaming via a web interface.
- Responsive remote control for movement and gun firing.

Figure 3: Raspberry Pi Camera Tank

Future enhancements include integrating autonomous navigation and AI-driven object recognition.

8 Conclusion

In conclusion, the project offers a scalable and modular platform ideal for both robotics and IoT applications. It serves as an effective educational tool, helping users explore concepts ranging from basic robotics to advanced automation. The platform's flexibility ensures its adaptability for a variety of use cases, particularly in surveillance and defense scenarios.

By integrating IoT, it supports remote monitoring and control, enhancing real-time operations. The system can evolve to include AI-driven navigation, enabling autonomous movement in dynamic environments. This capability is crucial for surveillance tasks in areas where human presence is limited.

The project highlights the intersection of education, innovation, and practical application. Its design allows for future expansion, enabling users to tackle complex challenges. Ultimately, it bridges theory and real-world solutions, fostering both learning and problem-solving.

References

[1] M. Hasan, R. Ahmed, and S. Khan, "Web-controlled robot using Raspberry Pi," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, no. 5, pp. 23–29, 2021.

[2] P. Kumar, S. Singh, and R. Sharma, "Real-time video streaming for surveillance robots," *Journal of Robotics and Automation*, vol. 9, no. 2, pp. 45–50, 2020.

[3] S. Gupta, A. Mehta, and N. Jain, "DC motor control using Raspberry Pi and PWM," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1200–1205, 2019.

[4] T. Wang, J. Li, and K. Zhou, "Electromechanical firing mechanisms for tactical robots," *IEEE Transactions on Mechatronics*, vol. 12, no. 3, pp. 190–198, 2020.

[5] M. Alam, A. Roy, and H. Chowdhury, "Flask-based web interface for robotic navigation," *IEEE Access*, vol. 8, pp. 30040–30050, 2020.