Design and Development of a Low-Cost Quadruped Robot Using ESP32 and Raspberry Pi for Terrain Navigation and Surveillance

[1]Samyak R. Ghangale, [2]Sumeet Adsul and [3]Prathamesh J. Mawale,[4]Nitin K.Kamble, [5]Rohit P.Jadhav
Robotics and Automation Engineering Department, D.Y. Patil College of Engineering, Akurdi

***Abstract*—Quadruped robots inspired by Boston Dynamics' Spot are revolutionizing surveillance, inspection, and search-and-rescue applications. However, commercial models are expensive and not accessible to student researchers.We present the design and implementation of a cost-effective quadruped robot using ESP32 and Raspberry Pi controllers, MG996 servos, MPU6050 IMU, ultrasonic sensors, and a 3D-printed frame powered by a 3S 4000mAh Li-ion battery.The robot demonstrates stable locomotion, real-time posture control using sensor fusion, and image transmission via ESP32-CAM. Control is managed using a custom web interface hosted on the Raspberry Pi.This project provides a practical platform for quadruped robotics research and education, with open-ended scope for computer vision and autonomous navigation upgrades.**

I. INTRODUCTION

The need for agile and terrain-adaptive mobile robots is increasing, especially in defense, disaster relief, and inspection scenarios. Boston Dynamics' Spot robot showcases these capabilities but is cost-prohibitive. This project aims to develop a budget-friendly quadruped robot that mimics similar functionality for academic and prototyping purposes.[1]

While prior efforts in hobbyist robotics have explored two or four-legged locomotion, very few projects have focused on merging ESP32 and Raspberry Pi for distributed control. This setup offers a real-time low-level controller (ESP32) and a high-level interface (Raspberry Pi), enabling advanced functionalities such as vision processing and network communication while maintaining affordability.

II. METHODS AND MATERIALS

The quadruped robot was designed and built with a focus on low cost, modularity, and adaptability. The following sub-sections describe the mechanical design, hardware components, control architecture, and software environment used for the development and operation of the robot.

.

1.Mechanical Design and Fabrication: The mechanical structure of the robot consists of a central torso and four articulated legs. Each leg has two segments (thigh and shank), connected by rotational joints to allow movement in both horizontal and vertical planes. The entire structure was designed using Fusion 360 and exported in STL format for 3D printing.The robot parts were printed using PLA+ filament with a layer height of 0.2 mm and 20% infill. PLA+ was chosen for its improved mechanical strength and durability compared to standard PLA. The modular design of the frame allows easy assembly and disassembly, which is beneficial for prototyping and maintenance.

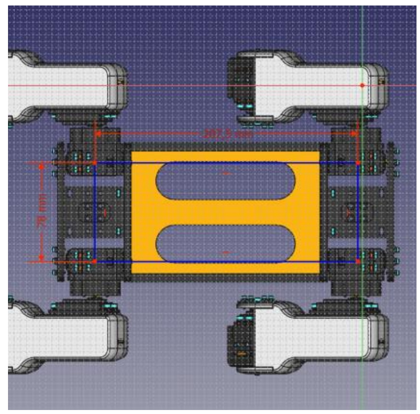

Fig. 1. CAD Model of the Quadruped Robot Frame [11]

This figure shows the top view of the 3D CAD model of the quadruped robot. It highlights the overall dimensions of the body structure, which measures approximately 207.5 mm in length and 78 mm in width. The model was created using FreeCAD and designed for easy 3D printing and modular assembly.

2.Actuators: Each leg is actuated by MG996R[5] servo motors, known for their affordability and decent torque output (~10 kg·cm at 6V). Two degrees of freedom are assigned to each leg: one at the hip for forward-backward motion, and one at the knee for vertical movement. A total of 12 servo motors are used (3 per leg), providing sufficient flexibility for executing basic gait patterns such as crawl and trot.

Servos are mounted using custom-designed brackets integrated into the 3D-printed frame. The servo horns are fixed to the leg linkages using M3 bolts, and thread-lock was applied to minimize loosening during movement.

3.Microcontrollers: The ESP32 microcontroller [2] was used to manage low-level control such as PWM signal generation for servos and sensor data acquisition. The Raspberry Pi 4 Model B [3] was employed to handle computationally intensive tasks such as image processing, streaming, and web UI hosting.

4.Sensors: The MPU6050 IMU provides 6-axis orientation data for posture control.[4] Two HC-SR04 ultrasonic sensors [6] at the front aid in basic obstacle detection. An ESP32-CAM[7] module streams real-time video to a web dashboard.

5.Power Supply:The robot is powered by a 3-cell (3S) 4000mAh Li-ion battery pack. The battery provides a nominal voltage of 11.1V and a maximum current discharge rating of

20C, sufficient to drive the servos and electronics.Voltage regulation is achieved using a buck converter to step down the voltage to 6V for servos and 5V/3.3V for logic circuits. Each component is protected with inline fuses, and power distribution is handled by a custom PCB that includes capacitor filtering to reduce electrical noise.

Fig. 2. Electronics Control Board Module[11]

This figure presents a close-up of the custom electronics control board that integrates the ESP32, voltage regulator, relay modules, and servo controller on a compact platform mounted at the center of the robot.

6. Software Architecture: The robot's motion control is implemented in the ESP32 firmware using Arduino IDE. Gait sequences are defined in arrays, and servo angles are updated in real time. The firmware also handles sensor fusion using data from the MPU6050, aling the robot to maintain balance during movement. On the Raspberry Pi, a Flask web server hosts a user interface accessible via Wi-Fi. The interface provides manual control buttons, joystick mapping, and a live video feed from the ESP32-CAM. Serial UART communication is used between ESP32 and Raspberry Pi, with Wi-Fi as a backup option.[8][11]

7.Control Interface: A Flask-based web server hosted on the Raspberry Pi allowed the user to control the robot via keyboard or joystick from a mobile or PC. Live video from the ESP32-CAM was embedded into the dashboard.

8.Gait Planning and Motion Logic

Crawl Gait: One leg moves at a time while the other three maintain stability. This gait is used for slow, stable walking.Trot Gait: Diagonally opposite legs move together, allowing faster motion but requiring better balance control.ach gait cycle is divided into stance and swing phases. Servo positions for each phase are stored in lookup tables, and time synchronization is achieved using the ESP32's hardware timers. PID control logic is applied to orientation data to ensure the torso remains level during movement. This enhances stability and prevents servo overstrain.

In summary, the methods and materials used in the development of this quadruped robot focus on accessibility, low cost, and scalability for academic and hobbyist applications. The combination of 3D-printed mechanics, modular electronics, and real-time firmware makes this platform a versatile testbed for further research in legged locomotion and robotic perception.

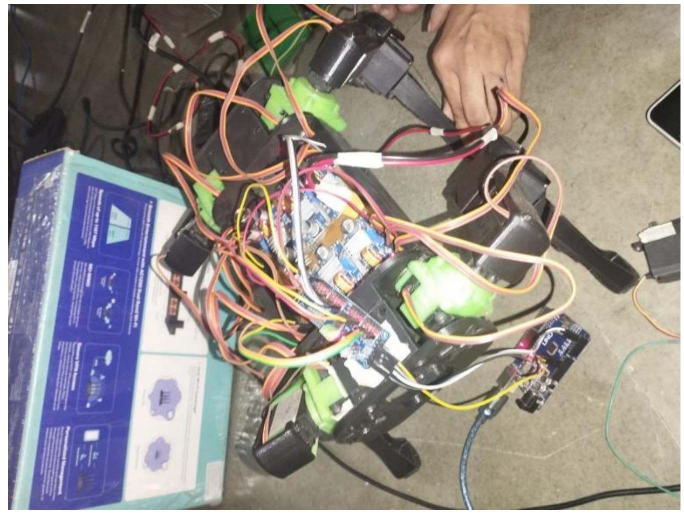

Fig. 3. Assembled Robot with Integrated Electronics

This image shows the partially assembled quadruped robot. MG996 servos are installed in each leg module, and the central frame houses the main control system with wiring, power modules, and sensors.

III. RESULTS

The quadruped robot was successfully assembled and tested under indoor conditions. The following observations were made: The robot achieved stable walking using crawl and trot gait sequences. Movements were smoother on flat surfaces such as tiles or wooden floors. The robot was able to recover from minor tilt disturbances thanks to the IMU-PID feedback loop.MPU6050 readings were used to maintain horizontal body orientation. The feedback system could correct pitch and roll deviations within ±5 degrees during movement.The ESP32-CAM module streamed video at 640x480 resolution at ~10 FPS, with acceptable latency when viewed through the Flask UI.On a full charge, the robot operated for approximately 30–40 minutes during mixed walking and idle periods. Current draw under load ranged from 1.5A to 2.5A.The web-based dashboard was responsive and allowed basic maneuvering commands including forward, backward, rotate left/right, and stop.These results demonstrate that the prototype is capable of performing basic quadruped locomotion and remote monitoring at a significantly reduced cost.[11].

IV. DISCUSSION

The design proved effective as an educational and experimental platform. Integrating the ESP32 and Raspberry Pi allowed for an efficient division of responsibilities between real-time control and computational overhead. This architecture can support future extensions such as:(a)Integration with computer vision models for object following (b) Use of LIDAR for SLAM and terrain mapping.Deployment of machine learning for adaptive gaits based on terrain feedback.ROS2 integration for multi-node robotic control and simulation.The robot's low cost (~INR 12000–14000) makes it accessible for robotics clubs, schools, and early-stage researchers.

V. CONCLUSION

This paper presented a cost-effective quadruped robot built using ESP32 and Raspberry Pi with open-source components and 3D-printed parts. The robot demonstrated stable gait, posture correction, and real-time video streaming. It offers a practical platform for students and researchers interested in legged robotics, real-time embedded systems, and autonomous mobility.

REFERENCES

[1] Boston Dynamics, "Spot: Agile Mobile Robot," [Online]. Available: https://www.bostondynamics.com/spot

[2] Espressif Systems, "ESP32 Technical Reference Manual," [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

[3] Raspberry Pi Foundation, "Raspberry Pi 4 Model B," [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[4] InvenSense, "MPU-6050 Product Specification," [Online]. Available: https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/

[5] TowerPro, "MG996R Servo Motor Datasheet," [Online]. Available: https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf

[6] Adafruit Industries, "HC-SR04 Ultrasonic Sensor Guide," [Online]. Available: https://learn.adafruit.com/ultrasonic-sonar-distance-sensors/overview

[7] Espressif Systems, "ESP32-CAM Development Board," [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-cam.html

[8] Arduino IDE Documentation, "ESP32 Servo and WiFi Libraries," [Online]. Available: https://docs.arduino.cc

[9] M. H. Raibert et al., "BigDog, the rough-terrain quadruped robot," in Proc. 17th World Congress Int. Federation Automatic Control, 2008, pp. 10822–10825.

[10] Y. Li, R. Wang, and M. Q.-H. Meng, "A Survey on Legged Robots," Chinese Journal of Mechanical Engineering, vol. 32, pp. 1–20, 2019.

[11] M. Kubina, "SpotMicroESP32 Project," GitHub repository, [Online]. Available: https://github.com/michaelkubina/SpotMicroESP32. [Accessed: 24-Apr-2025].