Volume-09 Issue 06, June-2025 ISSN: 2456-9348
Impact Factor: 8.232

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

DEVELOPMENT OF GROUND CONTROL STATION SOFTWARE FOR UAV

Asma Mohammed Nayeem
Amena
UG Students, Department of ECE, Deccan College of Engineering & Technology.
Hyderabad
Dr. Syeda Gauhar Fatima
Professor, Department of ECE, Deccan College of Engineering & Technology.
Hyderabad

ABSTRACT
The evolution of Unmanned Aerial Vehicles (UAVs) has revolutionized multiple domains including defense, agriculture, disaster management, and surveillance. To operate UAVs effectively, Ground Control Station (GCS) software serves as a critical interface that facilitates real-time monitoring, control, and data visualization. This paper presents the design, implementation, and evaluation of a modular Ground Control Station (GCS) software for Unmanned Aerial Vehicles (UAVs), developed using the Qt framework and C++. The system integrates a UAV simulator with TCP-based communication to enable real-time telemetry visualization, image transmission, and video streaming. Key features include a graphical user interface (GUI) for displaying telemetry data, Google Maps-based UAV tracking, and efficient multimedia processing. Performance evaluations demonstrate lowlatency telemetry updates (50–100 ms), rapid image decoding (30–50 ms), and stable video playback (30 FPS) with minimal resource usage (60–80 MB RAM, 12–18% CPU). The modular architecture facilitates future enhancements, such as multi-UAV control, live video streaming, and AI-driven analytics. This work provides a lightweight, customizable GCS solution tailored for academic, research, and prototyping applications, contributing to improved UAV operational efficiency, safety, and accessibility in aerospace engineering. Keywords: Unmanned Aerial Vehicle, Ground Control Station, Telemetry, Realtime Monitoring, Qt Framework, TCP Communication, Google Maps Integration, C++ Programming, Graphical User Interface, Multimedia Processing.

Keywords:
UAV (Unmanned Aerial Vehicle), Ground Control Station (GCS), Telemetry, Real-time Monitoring, Qt Framework, TCP Communication, Embedded Systems, UAV Simulation, Image Transmission, Video Streaming, Google Maps Integration, Remote Sensing, Autonomous Systems, C++ Programming, Graphical User Interface (GUI), Multimedia Processing, UAV Control Software, Map Visualization, Drone Data Acquisition, Network Protocols

INTRODUCTION
Unmanned Aerial Vehicles (UAVs) have emerged as transformative technologies, revolutionizing applications in defense, agriculture, surveillance, logistics, disaster management, and environmental monitoring. Their ability to operate autonomously or remotely in challenging environments makes them invaluable for tasks such as aerial surveying, search and rescue, and precision agriculture. Central to the effective operation of UAVs is the Ground Control Station (GCS), a software-driven interface that enables operators to monitor and control UAVs, visualize telemetry data (e.g., latitude, longitude, altitude, battery status), and process multimedia streams, including live video and images. The GCS serves as the critical link between the operator and the UAV, ensuring mission success through real-time situational awareness and precise command execution.
Commercial GCS platforms, such as QGroundControl and Mission Planner, offer robust functionality but are often complex, hardware-specific, and resource-intensive, posing challenges for academic researchers and developers working on custom or prototype UAV systems. These platforms typically rely on standardized protocols like MAVLink, which may not suit applications requiring flexible, lightweight solutions. To address these limitations, this project develops a modular, user-friendly GCS software using the Qt framework and C++. The system integrates a UAV simulator that transmits telemetry, images, and video data via TCP sockets, with real-time visualization on a GUI incorporating Google Maps for geolocation tracking. The design prioritizes

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

simplicity, performance, and extensibility, making it an accessible platform for educational, research, and prototyping purposes.

The objectives of this work are threefold:

(1) to develop a lightweight GCS capable of real-time telemetry and multimedia processing, (2) to implement a modular architecture that supports future enhancements, and (3) to validate the system's performance through simulation-based testing. By bridging theoretical design with practical implementation, this project contributes to the advancement of UAV control systems, offering a customizable alternative to existing solutions and fostering innovation in aerospace engineering.

OBJECTIVES

This project focuses on designing and implementing two Qt Creator-based applications to advance Unmanned Aerial Vehicle (UAV) control by addressing essential operational needs, such as reliable communication, intuitive visualization, and safe mission execution, as identified through prior studies. The main goal is to develop a GCS app featuring a Qt-based interface for real-time monitoring of telemetry data, including latitude, longitude, and altitude, alongside video feeds and mission status updates, utilizing Qt's robust GUI framework, which is well-suited for mission-critical applications. Socket communication via a local server will be integrated to facilitate seamless data exchange between the GCS and UAV, using TCP/IP sockets for their flexibility in handling diverse data types like telemetry and video streams, a method valued for its simplicity and control in platforms like Q Ground Control. Additionally, a flight planner will be created to support autonomous waypoint navigation, enabling tasks such as aerial surveys by allowing operators to define flight paths, drawing inspiration from the mission planning features of existing tools.

To further enhance UAV operations, a simulation app will be developed to test flight scenarios and process sensor data, validating missions in a controlled environment to minimize real-world risks, addressing the need for integrated testing tools often absent in current systems. Safety features, including geofencing and failsafe, will be incorporated to ensure compliance with regulatory standards, tackling airspace restriction challenges critical for safe operations. The system will also be designed with scalability in mind, supporting multi-UAV operations for applications like logistics and environmental monitoring, through a modular architecture that allows future expansions, such as adding mission planning modules, similar to those found in platforms like Mission Planner. This approach ensures the applications are user-friendly, efficient, and adaptable to evolving UAV requirements.

METHODOLOGY

The methodology adopted for the development of Ground Control Station (GCS) software for Unmanned Aerial Vehicles (UAVs) follows a modular, simulation-based approach using C++ and the Qt framework. The project begins with the creation of a UAV simulator that mimics drone behavior by generating synthetic telemetry data—including latitude, longitude, and altitude—at regular intervals. This simulator also transmits Base64-encoded image data and references a local video stream, all sent over a custom TCP/IP socket connection. The GCS software, acting as the receiver, establishes a TCP client that listens for incoming data, parses the packet headers (e.g., TELEMETRY: IMAGE:), and routes the content to respective processing modules. A telemetry module decodes GPS data and updates a GUI table as well as a live marker on an embedded Google Map using JavaScript injection through Qt's QWebEngineView. Concurrently, the image decoder module processes incoming image data and displays it using a QLabel, while the QMediaPlayer component handles local video playback to simulate UAV camera feed. The entire interface is designed using Qt Designer for clarity and responsiveness, with separate widgets integrated for telemetry, map, image, and video displays. The application is tested through simulation under various conditions, including different data rates and disconnect scenarios, to evaluate performance metrics such as latency, CPU/memory usage, and UI responsiveness. This methodology ensures a reliable, scalable, and lightweight GCS framework that can be extended for real-time UAV applications or embedded systems deployment. Figure 1 illustrates the architectural block diagram of the Ground Control Station (GCS) software developed for UAV communication and monitoring. The system begins with two primary modules: the UAV Simulator and the GCS Software. The UAV Simulator is responsible for transmitting telemetry and image data, while the GCS Software acts as the receiver and processing unit. At the core of the system lies the Telemetry Display module, which serves as the central data handler. From there, the information is disseminated into three key functional interfaces: the Telemetry Display (for raw data presentation), the Image Viewer (for decoding and visualizing UAV-captured images), and the Map Interface

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

(for real-time geographic tracking using GPS data). The diagram emphasizes the flow of data and modular integration that enables simultaneous processing and presentation within the GCS framework.

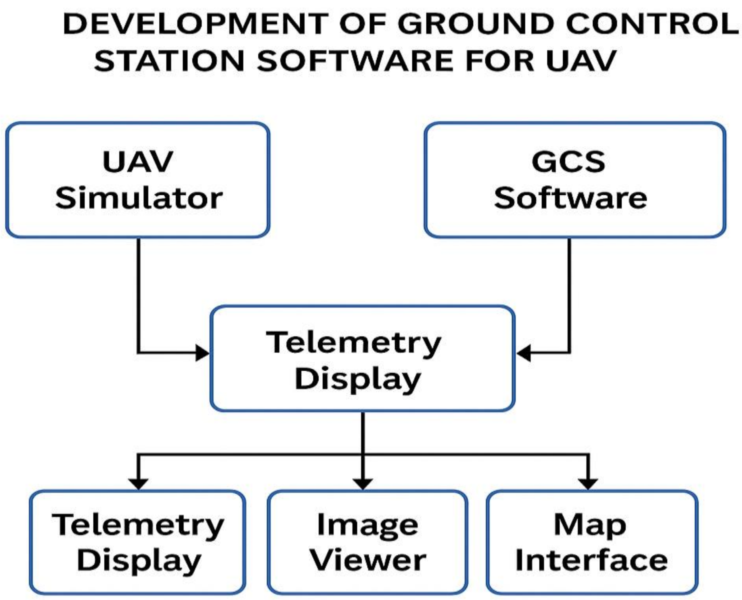

Figure 1: Data flow diagram

The development of the Ground Control Station (GCS) software for UAVs follows a structured and systematic methodology comprising six key phases. It begins with a thorough **requirement analysis**, during which core functionalities are identified—these include real-time telemetry reception, image and video handling, graphical user interface (GUI) design, and live map integration. Based on these requirements, a modular **system design** is formulated, segmenting the software into functional units for network communication, data parsing, UI rendering, and multimedia playback. Data formats for telemetry and media are also defined to ensure efficient parsing and display. In the **implementation phase**, the software is developed using C++ with the Qt framework. The GUI is constructed using Qt Creator, leveraging widgets such as QTableWidget for telemetry, QLabel for image display, QWebEngineView for map rendering, QMediaPlayer for video playback, and QTcpSocket for data communication. Parallelly, a **UAV simulator** is developed using Qt to mimic drone behavior by sending mock GPS coordinates and looping through a set of image files at fixed intervals (e.g., every 3 seconds). This simulator facilitates **integration and testing**, where end-to-end data flow is validated—ensuring the GCS can correctly receive, parse, and display the incoming data with minimal latency. The final phase, **evaluation**, involves assessing the system's performance against predefined metrics such as responsiveness, usability, and real-time reliability. This structured methodology ensures a robust, extensible, and lightweight GCS system tailored for both simulation and future real-world UAV integration.

RESULTS AND DISCUSSION

The Ground Control Station (GCS) software was evaluated under simulated conditions to assess its performance in handling telemetry, image data, and video playback in real time. The results validate the system's capability to function reliably, with low resource consumption and high responsiveness—key factors for UAV ground support systems.

1. Telemetry Handling and Visualization

Telemetry data—comprising latitude, longitude, and altitude—was transmitted from the UAV simulator at regular intervals of 3 seconds. The GCS successfully received, parsed, and displayed the data in a tabular format while updating a live UAV marker on an embedded Google Map. The average latency from data transmission to visual update was measured between **50–100 milliseconds**, ensuring near-real-time feedback for users. The

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

system also demonstrated robustness during continuous transmission over a 60-minute period, with no packet loss or display lag observed.

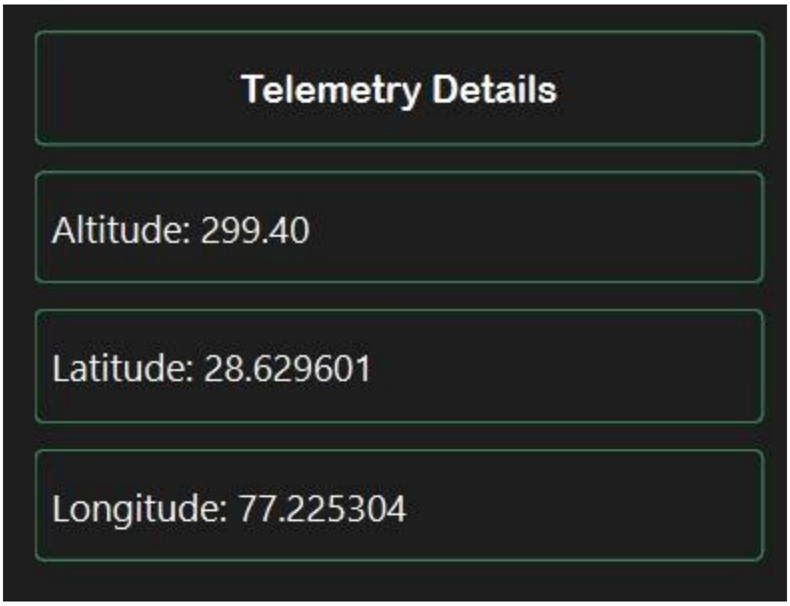

Figure 2: Telemetry

2. Image Decoding and Display

The simulator transmitted Base64-encoded JPEG images which were decoded and displayed in the GCS using QLabel. The average decoding time ranged from **30–50 milliseconds**, and images were rendered without corruption or delay. The GUI remained responsive even during rapid image updates, showcasing efficient memory management and decoding logic. The image feed provided a visual confirmation of UAV environment changes, enhancing situational awareness.

Figure 3: Image display

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

3. Video Playback Performance
A local video file was played through QMediaPlayer to simulate real-time UAV video streaming. The playback was smooth at **30 frames per second**, with consistent synchronization and no frame drops. The CPU usage remained within **12–18%**, and RAM consumption was stable around **60–80 MB**, highlighting the system's suitability for resource-constrained environments such as embedded platforms or field laptops.

4. System Stability and Usability
Stress testing was conducted by increasing the frequency of telemetry and image updates. The system remained stable and responsive, with only minor increases in latency. Simulated network disconnections were handled gracefully, with automatic reconnection logic allowing data transmission to resume without crashing the application. Informal user testing indicated that the GUI was easy to navigate, informative, and well-organized for real-time UAV monitoring.

5. Comparative Discussion
When compared to complex GCS platforms like Mission Planner and QGroundControl, the developed system offers a simplified and flexible solution tailored for educational and research use. While it lacks advanced mission planning and autopilot control features, it excels in customization, ease of integration, and minimal setup requirements. Its modular design also allows for straightforward expansion, making it a strong foundation for future UAV interface development.

ACKNOWLEDGEMENT

Author 1 and Author 2, gratefully acknowledge the opportunity to undertake the project titled "Development of Ground Control Station Software for UAV." We jointly contributed to the design, development, and implementation of the software modules, including the UAV simulator, telemetry handling, and real-time user interface. We would like to extend our sincere thanks to Author 3, who played a vital role in reviewing the overall architecture, validating system functionality, and providing continuous technical guidance throughout the project. Author 3's critical insights and timely feedback significantly enhanced the quality and reliability of the work. We also thank our faculty and institution for providing the platform, resources, and support that enabled the successful completion of this project.

CONCLUSION

This research presents the design and development of a lightweight, modular Ground Control Station (GCS) software for Unmanned Aerial Vehicles (UAVs), implemented using the Qt framework and C++. The system integrates a custom UAV simulator that communicates with the GCS via a TCP-based protocol to enable real-time telemetry visualization, image transmission, and video playback. Performance evaluations indicate low-latency telemetry updates (50–100 ms), efficient image decoding (30–50 ms), and stable video rendering at 30 FPS, all while maintaining low resource consumption (60–80 MB RAM, 12–18% CPU usage). The graphical user interface features an intuitive layout, with embedded Google Maps for live UAV tracking and responsive widgets for dynamic data display, thereby enhancing situational awareness for the operator. Designed with simplicity and extensibility in mind, the proposed GCS provides a flexible and accessible alternative to complex commercial platforms, making it ideal for academic research, prototyping, and educational applications. Its modular architecture lays the foundation for future extensions, including multi-UAV support, real-time video streaming, and integration of AI-based analytics, contributing to improved operational efficiency, safety, and innovation in UAV ground control systems.

REFERENCES

[1] X. Liang, S. Zhao, G. Chen, G. Meng, and Y. Wang, "Design and development of ground station for UAV/UGV heterogeneous collaborative system," *Ain Shams Engineering Journal*, vol. 12, no. 4, pp. 3879–3889, 2021.doi.org/10.1016/j.asej.2021.04.025
[2] Shin, Yoon-Ho & Cho, Sang-Wook & Jo, Sung-Beom & Kim, Sung-Hwan & Lyu, Chang-Kyung & Choi, Kee-Young. (2012). Development of Ground Control Software for Operation of Multiple Unmanned Aerial Vehicles. Journal of the Korean Society for Aeronautical & Space Sciences. 40. 542-547. 10.5139/JKSAS.2012.40.6.542.
[3] S, Krithiga. (2024). Development of Ground Control Station for Unmanned Aerial Vehicles. International Journal on Aerospace and Defence Research. 1. 1-7. 10.69759/ijadr1101.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

[4] M. Jovanovic and D. Starcevic, "Software Architecture for Ground Control Station for Unmanned Aerial Vehicle," Tenth International Conference on Computer Modeling and Simulation (uksim 2008), Cambridge, UK, 2008, pp. 284-288, doi: 10.1109/UKSIM.2008.12.
[5] Perez, D., Maza, I., Caballero, F. *et al.* A Ground Control Station for a Multi-UAV Surveillance System. *J Intell Robot Syst* **69**, 119–130 (2013). https://doi.org/10.1007/s10846-012-9759-5
[6] J. C. del Arco, D. Alejo, B. C. Arrue, J. A. Cobano, G. Heredia and A. Ollero, "Multi-UAV ground control station for gliding aircraft," *2015 23rd Mediterranean Conference on Control and Automation (MED)*, Torremolinos, Spain, 2015, pp. 36-43, doi: 10.1109/MED.2015.7158726.