LLM BASED AUTONOMOUS DRONE (TALK2FLY) DATABOSS ANALYTICS & SECURITY

PROJECT TEAM

Selin Ataş

Kutay Kaplan

Öykü Özbirinci

Efe Özdilek

Ömer Tuğrul

Muhammed Serkan Yıldırım

COMPANY MENTOR

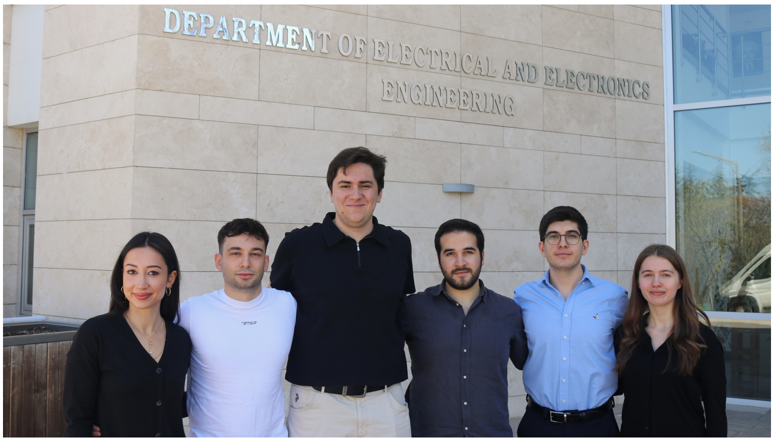
İsmail Balaban

ACADEMIC MENTOR

Prof. Süleyman Serdar Kozat

TEACHING ASSISTANT

Mehmet Efe Lorasdağı



Abstract. This project presents the design and implementation of an autonomous drone control system powered by Large Language Models (LLMs) and real time speech recognition, aimed at enhancing the usability and accessibility of UAV operations via natural voice interfaces. The system interprets Turkish voice commands and converts them into executable mission plans for drones operating in both simulated and real world environments. The architecture is composed of three main components: a mobile speech-to-text (STT) interface on Android with Flutter, a locally hosted Gemma 2 LLM pipeline for command classification and mission generation, and a drone control module. Communication across modules is managed via MQTT and JSON output of LLM is published to PX4 via MAVSDK to ensure structured, low latency message exchange. The system enables dynamic mission updates, emergency overrides, and telemetry feedback. Experimental evaluation demonstrates a command input latency of under five seconds and a 90% command classification accuracy. The final deployment supports full drone integration and lays the groundwork for potential improvements with visual recognition features. Designed for portability, edge deployment, and multilingual applicability, the system is suitable for use in domains such as defense, disaster response, and field inspection.

PROJECT DESCRIPTION

Our project focuses on developing an autonomous drone system powered by LLMs, which responds to Turkish voice commands to carry out real-world navigation and mission tasks, addressing the growing need for intuitive, hands-free control in autonomous vehicles. The company **DataBoss** seeks to solve the problem of complex UAV/UGV control interfaces by enabling non-technical users—including defense personnel, field workers, and consumers—to command drones using natural speech. This eliminates the barrier of technical knowledge and improves mission efficiency in sectors like security, logistics, and disaster response. The motivation stems from the increasing demand for accessible and intelligent autonomous systems; current solutions like **OmniDrive** [1], **TypeFly** [2], **GPT-Driver** [3], **LanguageMPC** [4], and **LLM4Drive** [5] provide partial solutions through cloud-based LLMs or vision reasoning systems but often require custom languages, expensive infrastructure, or lack offline autonomy. These products typically depend on cloud services, lack multilingual support, and require high-end hardware, which increases cost and limits scalability in resource-constrained environments. In contrast, our solution is novel in that it integrates a real-time STT system (using Whisper or Flutter's mobile STT), a Turkish-compatible LLM for command understanding and mission planning, and a drone control subsystem via PX4 and MAVSDK. All connected to the same MQTT client. The network between STT app and LLM is directly connected MQTT with text-based approach, and connection to the flight card is based on telemetry data via MAVSDK interface. This approach eliminates reliance on cloud APIs, and provides an edge-optimized, cost-effective solution that operates autonomously and adaptively even in dynamic or noisy field conditions. Our system is designed to meet strict specifications: $<10\%$ Word Error Rate (WER) in STT in standard outdoor conditions, command response times below 500 ms for short commands, <2 m navigation accuracy in real-world testing, drone power draw <6 A, and total system cost under 90000 TL. The design includes a modular architecture—STT on mobile, LLM on a PC, and drone control onboard—all connected via a lightweight MQTT protocol, allowing each component to work asynchronously.

The **Big Picture** involves a microphone-equipped mobile device transcribing speech, a PC-hosted LLM interpreting the text and generating mission commands, and a MAVSDK-controlled drone executing tasks and returning telemetry. Unlike other systems, our solution processes complex commands in Turkish, and includes structured feedback loops for adaptive mission control. This integration of LLM reasoning, low-latency STT, and real-time autonomous navigation forms a novel, robust, and patentable system suitable for both enterprise and field applications.

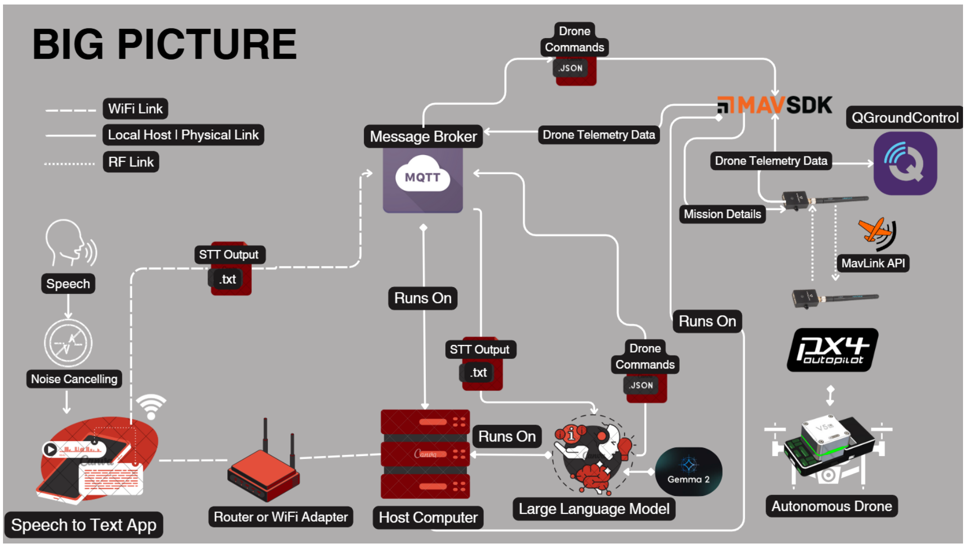


FIGURE 1. Big Picture of the Project

System overview showing the integration of the STT app, LLM, and PX4-based autonomous drone. Voice commands are transcribed on a mobile app and transmitted via MQTT to a host computer running the LLM. The LLM generates structured mission commands in JSON format, which are sent to the drone via MAVSDK. Real-time telemetry is fed back into both the LLM and QGroundControl for adaptive mission monitoring and execution. Communication is handled asynchronously over Wi-Fi and MQTT.

MILESTONES

Milestone 1: Parallel Development of Core Systems (Dec 15, 2024) The first milestone included literature review and research, followed by the development of STT, LLM, drone control systems in parallel. The explored STT models were FasterWhisper, wav2vec2, Flutter plugin. Real time voice capture was implemented using fasterwhisper-large model in the final simulation product based on WER, latency, and edge performance evaluations. LLaMA 3.1, Mistral, Cohere, Gemma 2 were explored for the LLM operations and Gemma 2 is used in the final product. In the drone control development, MAVSDK, PX4, Ardupilot, QGroundControl, Mission Planner were tested as control software alongside simulation software Gazebo and Airsim. The final product uses Gazebo as drone simulation software combined with PX4 firmware and MAVSDK communication. QGroundControl is the virtual Ground Control Station for the MAVLink protocol in the final product.

Milestone 2: Integration in Simulation (Jan 10, 2025) To integrate the core systems: STT, LLM and drone control in simulation, communication was conducted by publishing JSON file outputs from each system to MAVSDK via MQTT. Gazebo SITL executed missions and QGroundControl was used for feedback and log analysis.

Milestone 3: LLM Enhancement + Real Drone Integration (Mar 10, 2025) For the physical drone integration, CUAV V5+ open source flight control system was used with PX4 drone firmware. Sensor calibrations, GPS integration and telemetry setup were completed. LangGraph was used for LLM workflow by using a mission classification algorithm to generate missions. Vespa database was integrated for location retrieval in the LLM using BM25, E5, ColBERT embeddings. LLM output was formatted using Pydantic and Outlines structured generation libraries. Additionally, a mobile application was designed on Flutter and connected to the MQTT client for STT part of the project.

Milestone 4: Field Testing and Final Deployment (Apr 20, 2025) Outdoor tests were run with various missions on the physical drone to validate mission stability. Edge cases were tested to streamline the final product.

DESIGN DESCRIPTION

The project follows a milestone-driven solution strategy structured around the modular development and integration of STT, LLM, and drone control systems. In **Milestone 1**, each core subsystem was developed in parallel. The STT module was implemented using Whisper with noise cancellation techniques applied to enhance transcription accuracy.

The LLM subsystem, built using Gemma 2 (9B), employed prompt engineering, schema-based JSON formatting (via Pydantic and Outlines), and a retrieval-augmented generation (RAG) pipeline using ColBERT and E5 embeddings to enable robust mission planning. The drone control logic was developed in PX4 and tested using Gazebo and MAVSDK for basic commands and mission execution.

FIGURE 2. Example Mission Plan of LLM

In **Milestone 2**, these systems were integrated through an MQTT-based asynchronous communication interface, enabling real-time interaction across components. The STT outputs were transmitted to the LLM, which parsed them and generated structured mission plans in JSON format, executed via MAVSDK within the Gazebo simulation.

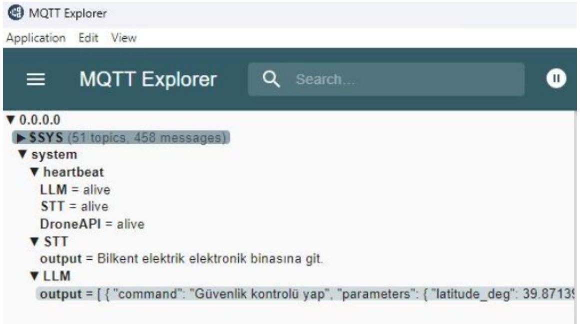


FIGURE 3. STT-LLM-Drone Connection Interface

In **Milestone 3**, the system was ported to physical hardware using the Pixhawk 6C-based Holybro X500 V2 drone kit. Voice input was captured from a mobile device and transmitted over Wi-Fi; the resulting JSON missions were executed by the real drone, with telemetry data streamed back for live reasoning and correction. The mobile app was developed for the part of STT on Android environment

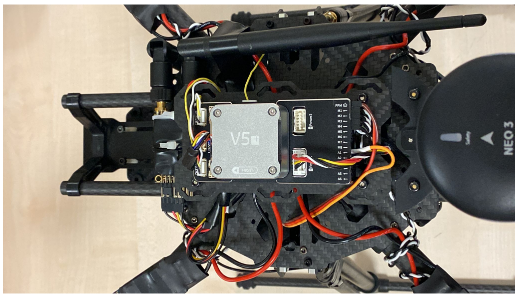


FIGURE 4. Physical Drone

Finally, in **Milestone 4**, full field testing was conducted. Emergency commands and safety-critical features (e.g., maximum altitude cap, return-to-home failsafe) were validated through live testing. The finalized system demonstrated low-latency response, mission accuracy, and robustness in noisy, dynamic environments—meeting the performance metrics defined at project outset.

Table 1 outlines the key tools, platforms, and libraries used throughout the project.

Component	**Tool/Platform**	**Purpose**
STT	Whisper / Flutter STT	Real-time Turkish transcription
LLM	Gemma 2 (9B) via vLLM/Ollama	Command generation and reasoning
MQTT Broker	Mosquitto	Asynchronous message passing
Simulation	PX4 + Gazebo + QGroundControl	Drone physics simulation and control
Drone Control	MAVSDK / MAVLink	Mission execution and telemetry
Drone Hardware	Pixhawk 6C (Holybro X500 v2)	Real-world navigation
Mobile App	Flutter + Android Studio	Speech input and UI
Data Handling	Pydantic, Outlines, LangChain	JSON formatting and schema validation

TABLE 1. Tools, libraries, and hardware used throughout the project

Whisper and Flutter STT were employed for speech recognition, Gemma 2 LLM handled command interpretation, and Mosquitto enabled inter-process communication. PX4 with Gazebo and QGroundControl supported simulation, while MAVSDK and Pixhawk 6C facilitated real-world drone control. Flutter-based mobile applications collected voice input, and Pydantic with LangChain managed structured data exchange.

RESULTS AND PERFORMANCE EVALUATION

The finalized system was tested both in Gazebo-based simulation and in real-world conditions using the Pixhawk 6C drone platform. Voice commands were issued through a mobile STT application, processed by the LLM, and executed via MAVSDK with telemetry feedback. Quantitative performance metrics for the system are summarized below:

- **STT** achieved a WER of $\leq 10\%$ in standard outdoor conditions and maintained above 85% accuracy in noisy environments using integrated noise cancellation.
- **Mission success rate** exceeded 90% in both simulation and physical field trials, even under parameter-omitted commands.
- **Navigation accuracy** remained within $\leq 1\,\text{m}$ in simulation and $\leq 2\,\text{m}$ during physical tests.

(A) Circular motion execution with real-time LLM tracking and correction

(B) Zig-zag motion execution with real-time LLM tracking and correction

FIGURE 5. Drone trajectories under LLM-guided control in different motion patterns

Figure 5 shows the accurate execution of mission plans derived from natural language input.

Compared to systems such as OmniDrive [1], TypeFly [2], and GPT-Driver [3], our solution exhibits several practical advantages. While OmniDrive and TypeFly rely on cloud-based vision-language models and custom scripting languages, our system performs end-to-end voice-based planning using a local LLM and standard communication protocols. Unlike cloud-dependent approaches, we achieve real-time response without external API calls, improving both latency and privacy.

While the system meets its design goals for accuracy, and autonomy, certain limitations remain. STT performance may degrade under extreme environmental noise despite adaptive filtering. Additionally, the current system is used with Turkish only, which limits broader applicability unless multilingual support is added. Latency values in LLM due to current hardware is also another limitation.

CONCLUSIONS AND FUTURE DIRECTIONS

The primary goal of this project was to develop an LLM-based autonomous drone system capable of executing complex missions based on Turkish voice commands. The system integrates a mobile STT interface, a LLM for command understanding and mission planning, and a MAVSDK-controlled drone for execution. Throughout the project, we successfully implemented all major milestones: real-time voice command recognition, LLM-based structured mission generation, system integration in simulation and on physical hardware, and real-world testing with telemetry feedback and failsafe mechanisms. To evaluate system performance, we measured key metrics such as WER for STT (targeting $\leq 10\%$), mission success rate ($\geq 90\%$), and navigation accuracy (within $\leq 1\,\text{m}$ in simulation, $\leq 2\,\text{m}$ in real-world testing). Additional qualitative evaluation included real-time command interpretation and emergency handling robustness.

In the future, the system can be extended to support multi-language capabilities, operate fully on embedded hardware, and handle more complex missions through improved reasoning and context awareness.

Furthermore, by integrating a depth camera, the system can record detected objects along with their spatial coordinates in a database, enabling the generation of new mission plans based on the updated dataset.

Potential applications include autonomous security patrols, logistics, agriculture, and disaster response. With its modular architecture and real-time performance, the system demonstrates strong potential for integration into DataBoss' broader AI-driven autonomous platforms.

REFERENCES

[1] H. Zhang et al., "OmniDrive: Generalizable Decision-Making for Autonomous Driving with Large Language Models," arXiv:2312.00436, 2023.
[2] Z. Luo et al., "TypeFly: Program-Aided Autonomous Aerial Navigation via Natural Language Instructions," arXiv:2310.06995, 2023.
[3] M. Liu et al., "GPT-Driver: Language Models as Driving Planners," arXiv:2309.00610, 2023.
[4] Y. Chen et al., "LanguageMPC: Large Language Models for Model Predictive Control," arXiv:2307.02088, 2023.
[5] Y. Zhang et al., "LLM4Drive: A Survey on Large Language Models for Autonomous Driving," arXiv:2401.02467, 2024.

BEHIND THE SCENES