U.P.B. Sci. Bull., Series C, Vol. 83, Iss. 2, 2021 ISSN 2286-3540

GROUND CONTROL STATION FOR MULTI-UAV SYSTEMS

Emilian VLASCEANU[1], Dan POPESCU[2]

Abstract – This paper proposes an original implementation for a GCS (Ground Control Station) software in the context of a multi-GCS and multi-UAV (Unmanned Aerial Vehicle) system. The ground system is built around a GDT (Ground Data Terminal), which is the gateway between ground system and aerial platform that is Hirrus UAV. The software is written in C++ programming language, is based on SDL (Simple DirectMedia Library) and OpenGL for implementing the GUI (Graphical User Interface), and STL (Standard Template Library) and BOOST Library for threading and asynchronous operations.

Keywords: UAV, GCS, multi-agent system, IoT.

1. Introduction

This paper studies the implementation of a multi-operator – multi-UAV (Unmanned Aerial Vehicle) UAS (Unmanned Aerial System) from the ground system point of view. UAV systems are frequently used for specific missions like intelligence, surveillance, reconnaissance, communication relay, photogrammetry, search and rescue or even for atmospheric measurements. The advantage of using the UAS over conventional aircraft is given by the lower costs, shorter operator training period, shorter deployment time, high flexibility and low risks for human operator.

Right now, there are many GCS software implementations under continuous development and some of them are already commercialized with or without the autopilot hardware. One of the most popular solution is Mission planner [1]. This software is only compatible with Windows OS (Operating System) but is very flexible considering the platform used. The operator could perform a mission using a plane, copter or even a rover. Another great solution is the Visionair GCS software [2]. They propose one of the most complex implementations for a GCS software that is compatible with almost all commercialized autopilots. Also, their software solution can be used simultaneously from multiple GCSs in a multi-UAV mission paradigm. Like Mission Planner, the Visionair GCS software works only on Windows OS.

[1] PhD Student, Faculty of Automatic Control and Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: emilian.vlasceanu@aft.ro

[2] Prof., Faculty of Automatic Control and Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: dan_popescu_2002@yahoo.com

The GCS (Ground Control Station) software development relies on the following functional requirements [3]:

1. Real-time UAV control [4]: The human operator should be able to control the UAV anytime during the mission. This means the datalink (uplink and downlink) between the aerial platform and the ground system should be up and running while de UAV is deployed in a specific mission.
2. Payload manipulation: The UAV could be equipped with a specific payload during the mission. From the GCS point of view, the human operator should be able to control each type of payload the aerial platform is equipped with. The GCS software must implement a specific protocol for data encapsulation (commands and telemetry data) for each type of payload the UAV could be equipped with.
3. Mission planning [5]: One of the most important actions during the preflight session (or even during flight) is the mission planning. The human operator should be able to define the UAV flight path, the safety zone (home point) and the secondary flight parameters.
4. Preflight checks: To check the UAS functionality, during the preflight session, the human operator will follow a specific set of procedures. Many of them are automated tests that will run recursively during the mission, so the human operator will be notified if the UAS incurs problems.
5. Data logger and video recorder: The GCS is responsible for recording the entire telemetry stream captured and if desired, the video recording can be enabled.
6. Operator training: Our goal is to train the UAV operator to handle the aerial platform, monitoring, and controlling the actions, if this is required.
7. Multi-UAV monitoring and control [6]: the possibility to monitor the entire UAV team and the capacity of intervention, if needed.
8. Control over the internet: This feature is useful when the operator is working in a mission control center.

From an operator point of view, the human mind is mostly a “single task device”, so the intervention over the UAS should be done on one aerial platform at a time.

In order to increase the system robustness, the telemetry should be available to multiple human operators. Our goal is to sketch a ground system architecture based on Hirrus UAV that can accommodate a variable number of GCSs, controlled, each one, by a human operator. Our software solution should run on multiple operating systems, using the same hardware. In order to achieve this objective, we are developing this software using C++ programming language and cross-platform libraries: STL (Standard Template Library), Boost, SDL (Simple DirectMedia Library) and OpenGL.

In a multi-GCS setup, having multiple operators focused on the same UAV **can** be a problem from the control point of view. We propose an algorithm for giving rights and ownership for each GCS over a specific set of controls.

2. Methodology

The main components of the UAS are:

1. Aerial platform – Hirrus UAV (Fig. 1) – a fixed wing mini UAV used in various applications (search and rescue, photogrammetry, surveillance, etc.). This UAV may accommodate three different types of payloads: day payload, night payload and a high-resolution camera.
2. GDT (see Fig. 2) – is part of the Datalink system and provides the link between the ground system (IPBox + GCSs) and the aerial platform. The Datalink System also contains the Airborne Data Terminal that is available onboard the aerial platform.
3. IPBox (Fig. 3) – is part of the GDT system but is mentioned separately because it acts like a dispatcher for ground system. It ensures the connection between GCSs and GDT. In a catastrophic situation (if all connected GCSs are down), the IPBox can replace the GCS minimal functionalities: emergency landing or return to home commands.
4. GCS (Fig. 4) – the interface between the human operator and the rest of the UAS. Its main scope is to plan and monitor the mission, but it also covers additional functions like reception and encoding of the video stream in STANAG 4609 TS (Transport Stream) format, using the telemetry data received from the aerial platform, and the post-mission analysis.
5. Launcher (Fig. 5) – the main purpose of this component is to deploy the UAV to the mission even if the weather conditions are not ideal. It can ensure a safe launch despite severe atmospheric manifestations.
6. Power supply system (Fig. 6) – it provides the electrical power for the ground system (GCS, GDT) and contains three modules (two of them are mandatory – Power Unit and Uninterruptible Power Supply module, and one is optional – Light Power Unit).

Fig. 1: UAV - Hirrus L

Fig. 2: GDT

Fig. 3: IPBox

Fig. 4: GCS

Fig. 5: Launcher

Fig. 6: Power supply system

The UAV system featured in this paper is capable of working in a flexible architecture, but for the scope of this document, we take into consideration/we consider a system based on 2 UAVs, 1 GDT, 1 IPBox and 2 GCSs. A third GCS can be connected to this system using a 3G modem, but the system complexity will increase because the system should accommodate a server. The server should intermediate the connection between remote GCSs and the IPBox (Fig. 8).

When we are talking about this network setup, the ground system is isolated (working in a LAN – Local Area Network setup), so the ground equipment should be configured by a network expert. The IP (Internet Protocol) addresses are statically allocated, and each GCS is configured to communicate with the IPBox, by setting up the IPBox config parameters from its network setup menu (Fig. 7).

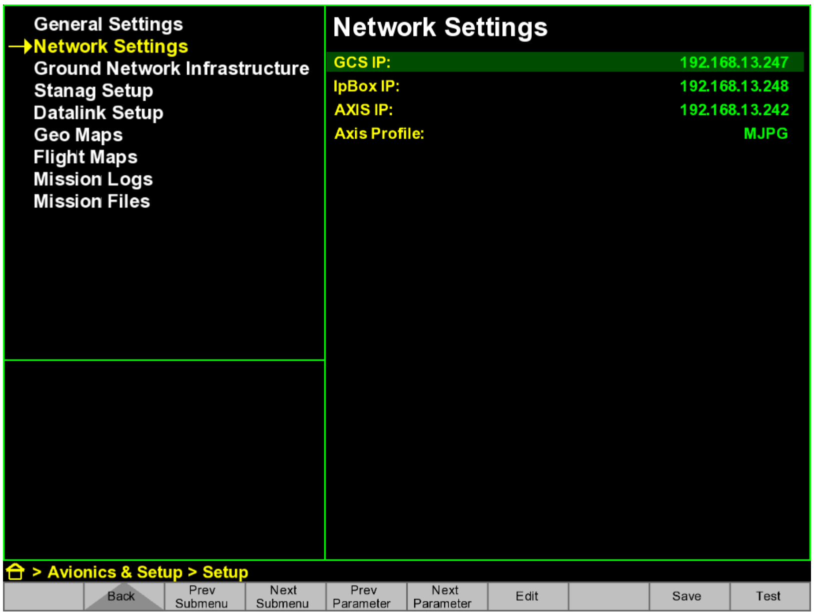

Fig. 7: Network config menu

If the ground system is connected to the internet, the router is the main network component of the ground system. The DHCP (Dynamic Host Configuration Protocol) is enabled and the ground network IPs will be dynamically allocated when a device is connected to the network. The router acts as a gateway between the local network and the remote GCS and it is mandatory to connect the server directly to the router.

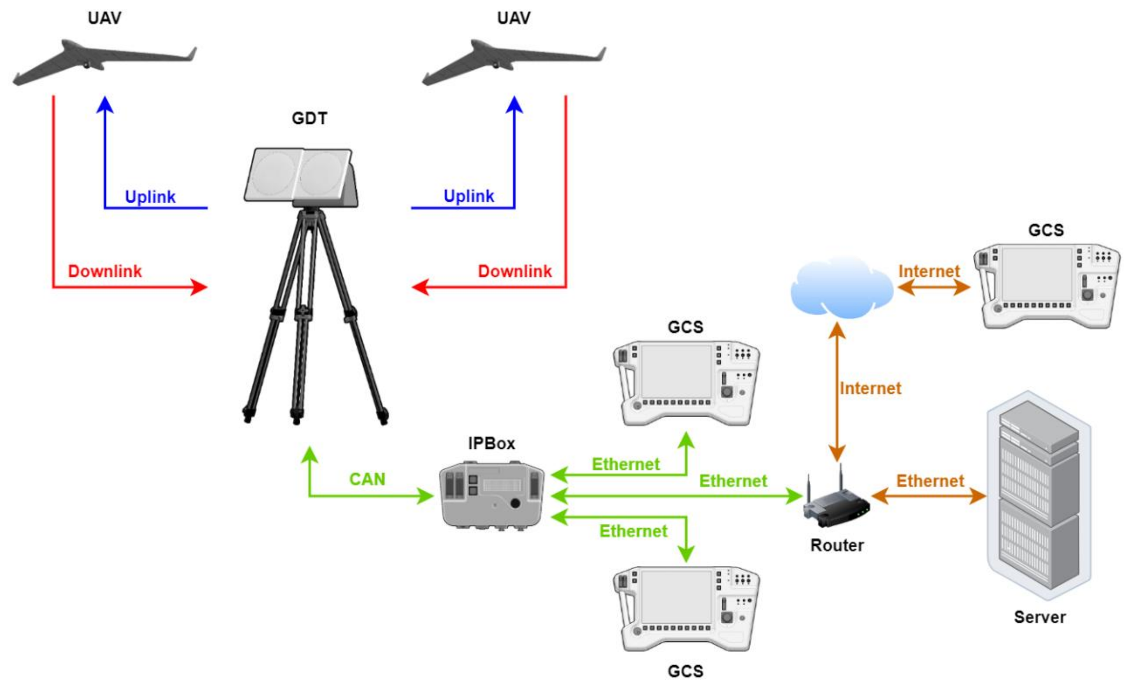

Fig. 8: Multi-UAV, multi-GCS with internet connection

By working in a multi-UAV and multi-GCS setup we get a MIMO system (Multiple-Input and Multiple-Output) which gives us the possibility to optimize the interaction with the resources.

There are some limitations in a multi-GCS system setup. Multiple GCSs can access the same specific resources at a time, and this could lead to a failed mission. We propose a lock access policy based on a network master GCS responsible with giving rights to the rest of the GCSs. Each GCS can get access right to many categories of resources, but at a given time, there are some categories where only one GCS can have ownership prerogatives.

The right policy is the first layer of access and relates to the entire system (including all UAVs).

The ownership is the second layer of access and relates to a specific resource (Table 1). For example, if the payload can be controlled by multiple GCSs (many GCSs received right to access payload resource), there will be only one GCS that can control the payload at a time, for a specific UAV. If the GCS discards the control, other GCS could take the ownership. On the other hand, if multiple GCSs could control the UAV (let’s assume we want to change the UAV altitude setpoint), any of these GCSs could change at the same time the parameter (the altitude is changed using a rotary controller so the value sent to the UAV is relative to current value). If one GCS increments the altitude with 1 meter, and another one decrements the same parameter with 1 meter, the result is that the UAV has the same altitude setpoint as before the GCSs intervention.

Table 1

Access rights for GCS

Category	Right	Owner
Network master	one	one
Mission planning	all	one
UAV control	all	all
Payload control	all	one
Checklist responsible	one	one
Ground system manager	all	one

Another important aspect in the internet connection setup is to forward the broadcast package over the network tunnel established between the local server and the remote GCS. The server should be connected to the IPBox via a 3G/4G router and should be a relay for the telemetry data and the STANAG 4609 transport stream. The server will act as a manager over the ground network, being capable of configure the rights policy.

3. Implementation

a. Hardware

The Ground Control Station hardware can be categorized into two groups:

i. Computer. Behind the shock resistant case, it is a standard PC setup: motherboard, CPU, RAM, SSD, monitor and power supply

ii. Custom designed controllers located on the front: joystick and buttons that are commanded by a specialized board, designed and developed by our team.

b. Software

The Ground Control Station software is developed using C++ programming language. It is based on STL (Standard Template Library) and BOOST library for backend and OpenGL for frontend.

The software is a multi-tasking application, based on Active Objects design pattern and Event Aggregator [8]. Each thread is an active object and the communication between threads is done using predefined events. Each thread subscribes on desired events and notifies other threads by publishing a specific event.

There are 3 threads:

1. Main thread, especially based on SDL and OpenGL, is the module where the data is displayed in specific indicators and the graphical controllers are implemented. This component is organized as a collection of menus, each menu having its own list of components:
    a. Aircraft instruments
    b. Buttons, labels and other GUI (Graphical User Interface), tools
    c. Map component
    d. Video player and photo player
    e. Aerial platform info layer
    f. Ground system layer

   The rendering module is a resource consuming one, so the usage of a high-performance set of libraries is a must. OpenGL [9] provides a collection of features that can accelerate the rendering performance.

   Unfortunately, OpenGL is not capable of creating and rendering a window. For this, we are using SDL [10] (it is able to create multiple windows) and also this library provides access for the input events (we are capturing mouse, keyboard and touchscreen events).

Main thread is also responsible for manipulating data received from the UAV and generating flight paths or other type of commands to be sent to aerial platforms.

2. Communication thread, that can be split in two submodules:
    a. Network communication: the network interface with the IPBox;
    b. Serial communication: the communication with the embedded module which interacts with hardware buttons and controllers.

   This thread is built around Boost.ASIO [11] library. We have used asynchronous methods to ensure a bidirectional communication with the endpoints.
3. Video link thread: this thread is used to connect the GCS with the UAV video source.

The video encoding thread is a complex mixture of algorithms that combines the telemetry data received from the aerial platform with the video stream transmitted between UAV and ground.

The transmission protocol used inside the UAS ground network is UDP (User Datagram Protocol) because it is suitable for real-time applications and multiple endpoints [12]. The IPBox sends a broadcast message to the network, 80 times per second, and each GCS responds with at least one message per second (alive message) or multiple messages if the operator is changing a set of parameters. The message is a collection of KLVs (Key-Length-Value) structures (see Table 2) which means the size of the message is variable.

Table 2

Example for UAV parameters encapsulation

Parameter	Key	Length	Value
UAV Latitude	1012	8 bytes	44.365766
UAV Longtiude	1013	8 bytes	25.912924
UAV Payload Power	5730	1 byte	0

A separate process can be attached to GCS software: STANAG 4609 video encoding software, developed in concordance with NATO digital motion imagery standard [13].

4. Experimental results and discussions

Lately, our team developed a new type of payload suitable for data transfer missions [14]. More precisely, this payload initiates the communication between the aerial platform and the ground CH (Cluster Head). On the other hand, the CH is radio-linked with a variable number of SN (Sensor Nodes) [15] and **collects** data from the endpoints, acting as a gateway between GSN (Ground Sensors Network) and the aerial platforms. The GCS Software should accommodate the new features of the recent developed payload. This means, the human operator should be able to control the payload and monitor the mission progress.

For a standard data acquisition mission our system setup is based on:

- 2 fixed wings UAVs from Hirrus family.
- 1 GDT
- 2 GCSs
- 2 Power units

- 1 Launcher
- 2 WSNs (each WSN is composed of 1 CH and 3 SNs)

Fig. 9: Multi-GCS setup

The proper way to reach the mission goals was to use 2 GCSs controlled by 2 human operators (Fig. 9). The mission started with the planning stage. At this point, the operator uploads the flight pattern to be followed by the UAV, the safety, the takeoff and landing parameters. The flight pattern should match the data transfer requirements, so each WP (Waypoint) is placed next to a CH geographical position (latitude and longitude) and the flight altitude is set to a safe value (at least 100 m above ground level). Nevertheless, these parameters can be adjusted during the mission.

After the planning stage ends, the preflight checks should be completed. The GCS software includes a collection of preflight tests to be accomplished before the UAV deployment. At this point, if there is a failed check, the mission could not begin safely.

The next step is to deploy each UAV to its own mission. The launcher used to release the aerial platforms to the mission is controlled by the operator in charge with the UAV to be launched. Each operator will send the launch command to the UAV at a time.

During the UAVs mission, the GCS receives real-time telemetry data which is displayed in a custom-made GUI (Fig. 10).

Fig. 10: GCS screenshot during a multi-UAV mission

After the first waypoint is reached, the data transfer begins automatically, as it was planned. As long as the transfer is not finished, the UAV will loiter over the specific waypoint. When the transfer is done, the UAV will go to the next waypoint and will repeat the scenario. If there is no other waypoints to fly over, the UAV will change its flight mode to landing procedure and will return to ground using a parachute.

The data collected during this mission is stored onboard and at the GCS level on the storage unit. After the mission ends, the data is analyzed by an expert.

The common GCS hardware used for obtaining experimental results is:

1. Intel Core i5 – 3.2 GHz CPU
2. 8 GB of RAM
3. 500GB SSD
4. Gigabit network card

Using the above setup, during the execution, the software CPU performance was almost all the time under 1% and the RAM footprint was under 150 MB.

During the beta release we were monitoring the network traffic generated by the ground system. For the telemetry and commands link, the bandwidth required was almost insignificant (about 60 to 80 Kbps). When we are talking about the video link, the necessary transmission capacity was higher (up to 4 Mbps when the video codec was H264 and the captured image was dynamic). In an over-the-internet setup, the H264 codec should be configured to fit 2 Mbps bandwidth. This means the FPS (Frames per Second) could be changed from 25 to a lower value, or the resolution could be reduced from 704 x 576 to 640 x 480 or even lower). There are also other parameters that can be altered to obtain the desired bandwidth, but the STANAG specialist should decide what the best option for reaching the mission objectives **is**.

Fig. 11: Multi-UAV mission

5. Conclusions

The Ground Control Station software of UAS is the main interface between the human operator and the rest of the UAS, including the ground system and the aerial platform(s). It is a mandatory component of the system. GCS is a collection of custom-made hardware components corroborated with off the shelf PC (Personal Computer) equipment, all of them, being controlled with our software solution. From the software perspective, the system is based on a particular architecture and custom-made implementation, designed especially for the GCS hardware. The proposed solution can run on Linux based operating systems as well as on Windows platform. It is very easy for us to implement new features as long as our team has access to resources such as UAV platform and ground system. This gives a high flexibility to our solution, compared to the other software products available on market.

Working with a limited budget and a small team like ours is challenging because it is hard to compete with those companies that can deliver new products faster than us. Also, the software developed by us is compatible only with our UASs so it is hard to integrate products from other companies. On the other hand, we have implemented the latest STANAG agreements to provide a common interface between our software and other NATO member's software solutions.

Our plan is to continue the development of the GCS software, including adding new features and flight path generation algorithms. Also, the log player feature is under development and will be an excellent tool for post-flight analysis.

Acknowledgment

Research supported by the Operational Program Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125.

R E F E R E N C E S

[1]. https://ardupilot.org/planner/docs/mission-planner-overview.html

[2]. https://www.uavnavigation.com/products/ground-control-station/visionair-gcs-software

[3] *Natarajan, G.. (2001).* Ground control stations for unmanned air vehicles (Review Paper). Defence Science Journal. 51. 229-237. 10.14429/dsj.51.2234.

[4]. *Romaniuk, Sławomir & Gosiewski, Zdzislaw & Ambroziak, Leszek. (2016).* A Ground Control Station for the UAV Flight Simulator. Acta Mechanica et Automatica. 10. 10.1515/ama-2016-0005.

[5]. *Jovanovic, Mladjan & Starcevic, Dusan. (2008).* Software Architecture for Ground Control Station for Unmanned Aerial Vehicle. 284 - 288. 10.1109/UKSIM.2008.12.

[6]. *Perez Rodriguez, Daniel & Maza, Ivan & Caballero, Fernando & Scarlatti, David & Casado, Enrique & Ollero, Anibal. (2012).* A Ground Control Station for a Multi-UAV Surveillance System. Journal of Intelligent & Robotic Systems. 69. 10.1007/s10846-012-9759-5.

[7]. *E. Vlasceanu, D. Popescu and L. Ichim*, "Ground Control Station for an Unmanned Aerial Vehicle Integrated in IoT," 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Iasi, Romania, 2018, pp. 1-6, doi: 10.1109/ECAI.2018.8679050.

[8]. *Jovanovic, Mladjan & Starcevic, Dusan & Jovanovic, Zoran. (2010).* Improving Design of Ground Control Station for Unmanned Aerial Vehicle: Borrowing from Design Patterns. Proceedings - 36th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2010. 65-73. 10.1109/SEAA.2010.31.

[9] khronos.org/opengl/

[10] http://wiki.libsdl.org/CategoryVideo

[11]. https://www.boost.org/doc/libs/1_74_0/doc/html/boost_asio.html

[12] https://en.wikipedia.org/wiki/User_Datagram_Protocol

[13]. STANAG 4609 JAIS (EDITION 3)-NATO DIGITAL MOTION IMAGERY STANDARD

[14]. E. *Vlasceanu, D. Popescu and L. Ichim*, "Aerial Robotic Team for Complex Monitoring in Precision Agriculture," 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini Island, Greece, 2019, pp. 167-169, doi: 10.1109/DCOSS.2019.00044.

[15]. *D. Popescu, E. Vlasceanu, M. Dima, F. Stoican and L. Ichim,* "Hybrid Sensor Network for Monitoring Environmental Parameters," 2020 28th Mediterranean Conference on Control and Automation (MED), Saint-Raphaël, France, 2020, pp. 933-938, doi: 10.1109/MED48518.2020.9183165.