EDUCATION AND SCIENCE MINISTRY OF UKRAINE
NATIONAL AVIATION UNIVERSITY
DEPARTMENT OF COMPUTER INTEGRATED COMPLEXES

ADMIT TO DEFENSE
Head of department
Viktor M. Sineglazov
“____” _______________ 2022

QUALIFICATION WORK

(EXPLANATORY NOTE)

GRADUATE OF EDUCATIONAL AND QUALIFICATION LEVEL
“ MASTER”

THEME: Control and Monitoring System of an Unmanned Aerial Vehicle. Monitoring Subsystem (complex)

Executor: **Tsoba.A.O**

Supervisor: **D.t.s., Professor Sineglazov V.M.**

Advisor on environmental protectio Ph.D., Associate Professor Iavniuk A.A.

Advisor on labor protection: Senior Lecturer Kozlitin O.O.

Norms inspector: Ph.D., Professor Filyashkin M.K.

Kyiv 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ КОМПЛЕКСІВ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
В.М. Синєглазов
"_____" ____________2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ
"МАГІСТР"

Тема: Система управління та моніторингу безпілотного літального апарату. Підсистема моніторингу (компл)

Виконавець: **Цьоба А.О.**

Керівник: **д.т.н., професор Синєглазов В.М.**

Консультант з екологічної безпеки: **к.т.н., доцент Явнюк А.А.**

Консультант з охорони праці: **старший викладач Козлітін О.О.**

Нормоконтролер: **к.т.н., професор Філяшкін М.**

Київ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп'ютерно - інтегрованих комплексів

Освітній ступінь: магістр

Спеціальність 151 "Автоматизація та комп'ютерно-інтегровані технології"

Освітньо-професійна програма "Комп'ютерно-інтегровані технологічні процеси і виробництва"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Синєглазов В.М.

" ____ " __________2022 р.

ЗАВДАННЯ
на виконання дипломної роботи студента

Цьоба Артура Олександрович

1. Тема роботи: «Система управління та моніторингу безпілотного літального апарату. Підсистема моніторингу (компл)».
2. Термін виконання роботи: з 19.08.2022р. до 15.11.2022р.
3. Вихідні дані до проекту (роботи): структурні схема НСУ**,** структурна схема БПЛА**,** протокол MavLink, середовище програмування QtCreator.
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): 1. Аналіз безпілотних літальних апаратів (БПЛА) та їх особливості, аналіз наземної станції управління (НСУ); 2. Аналіз підсистем управління та проблем захисту лінії управління ; 3. Аналіз існуючих рішень захисту управління польотом та захисту кібер лінії; 4. Розробка та дослідження лінії захисту управління БПЛА; 5. Розробка та дослідження інтерфейсу користувача (оператора); 6. Розробка додаткових підсистем моніторингу.
5. Перелік обов'язкового графічного матеріалу: 1. Структурні схеми комплексів БПЛА та НСУ. 2. Структурна схема БПЛА; 3. Типовий амплітудно-частотний спектр сигналу з імпульсно-кодовою модуляцією; 4. Існуючі рішення захисту лінії управління БПЛА ; 5. Узагальнена схема навігації БПЛА; 6. Результати досліджень синтезованих контурів управління.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Відмітка про виконання
1	Аналіз актуальності проблеми	09.08.2022-26.08.2022	
2	Аналіз характеристик безпілотних літальних апаратів та їх застосування	26.08.2022-02.09.2022	
3	Дослідження інформаційного забезпечення систем моніторингу безпілотними літальними апаратами	02.09.2022-16.09.2022	
4	Дослідження підсистем моніторингу та управління наземної станції управління	16.09.2022-23.09.2022	
5	Дослідження лінії управління безпілотним літальним апаратом	23.09.2022-07.10.2022	
6	Розробка та дослідження підсистеми моніторингу та управління безпілотного літального апарату	07.10.2022-21.10.2022	
7	Розробка графічного інтерфейсу користувача (оператора)	21.10.2022-04.11.2022	
8	Висновки по роботі та підготовка презентації і роздаткового матеріалу	04.11.2022-15.11.2022	

7. Консультанти зі спеціальних розділів

Розділ	Консультант (посада, П. І. Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці	Старший викладач, Козлітін О.О,		
Охорона навколишнього середовища	к.б.н., доцент, Явнюк А.А.		

8. Дата видачі завдання ______________________________

Керівник: ________________________ Синєглазов В.М.

(підпис)

Завдання прийняв до виконання: ________________________ Цьоба А.О.

(підпис)

NATIONAL AVIATION UNIVERSITY

Faculty of aeronavigation, electronics and telecommunications

Department of Aviation Computer Integrated Complexes

Educational level: master

Specialty 151 " Automation and computer-integrated technologies"

Educational and professional program "Computer-integrated technological processes and production"

APPROVED BY

Head of department

Victor M. Sineglazov

"_____"______________ 2022

Graduate Student's Diploma Thesis Assignment

Tsioba Artur Oleksandrovych

1. The thesis title: «Control and Monitoring System of an Unmanned Aerial Vehicle. Monitoring Subsystem (complex).».

2.The thesis to be completed between: from 19.08.2022 to 15.11.2022.

3. **Output data for the thesis:** structural diagram of the NSU, structural diagram of the UAV, MavLink protocol, QtCreator programming environment.

4. **The content of the explanatory note (the list of problems to be considered):** 1. Analysis of unmanned aerial vehicles (UAVs) and their features, analysis of the ground control station (GCS); 2. Analysis of control subsystems and control line protection problems; 3. Analysis of existing solutions for flight control protection and cyber line protection; 4. Development and research of UAV control line protection; 5. Development and research of the user interface (operator); 6. Development of additional monitoring subsystems.

5. List of compulsory graphic material: 1. Structural diagrams of UAV and NSU complexes. 2. Structural diagram of the UAV; 3. Typical amplitude-frequency spectrum of the signal with pulse-code modulation; 4. Existing solutions for UAV control line protection; 5. Generalized scheme of UAV navigation; 6. Results of studies of synthesized control loops.

6. Planned schedule:

№	Task	Execution term	Execution mark
1	Analysis of the relevance of the problem	09.08.2022-26.08.2022	
2	Analysis of characteristics of unmanned aerial vehicles and their application	26.08.2022-02.09.2022	
3	Research of information support of monitoring systems by unmanned aerial vehicles	02.09.2022-16.09.2022	
4	Research of monitoring and control subsystems of the ground control station	16.09.2022-23.09.2022	
5	Research of the unmanned aerial vehicle control line	23.09.2022-07.10.2022	
6	Development and research of the monitoring and control subsystem of the unmanned aerial vehicle	07.10.2022-21.10.2022	
7	Development of graphical user interface (operator)	21.10.2022-04.11.2022	
8	Conclusions on the work and preparation of presentation and handouts	04.11.2022-15.11.2022	

7. Special chapters' advisors

Chapter	Advisor (position, name)	Date, signature	
		Assignment issue date	Assignment accepted
Labor protection	Senior lecturer, Kozlitin O. O.		
Environmental protection	Ph.D, Associate Professor, Iavniuk V.F.		

8. Date of task receiving: ______________________

Diploma thesis supervisor: ____________________ Victor M. Sineglazov
(signature)

Issued task accepted: ____________________ Artur O. Tsoba
(signature)

ABSTRACT

For the thesis " Management and monitoring system of an unmanned aerial vehicle. Monitoring subsystem (complete). »

Keywords: GROUND CONTROL STATION, DATA EXCHANGE PROTOCOL, CONTROL SYSTEM, UNMANNED AIRCRAFT, CONTROL LINE, PROTECTION LINE, USER INTERFACE.

Explanatory note: number of pages - , number of figures - , number of used sources - .

The object of the research is the UAV monitoring system, the data exchange protocol and the UAV control line.

The purpose of the work is to develop the UAV monitoring and control subsystem, to modernize the protection of the UAV control line, to develop the user interface.

A study was conducted, based on the results of which the UAV protection protocol and criteria were selected. An algorithm has been developed that ensures the establishment of the fact that the received radio radiation belongs to the class of radio signals of UAV remote control systems with pulse-position and pulse-code modulations.

The results of practical testing of possible solutions are presented and the results of the functioning of the developed communication components are given .

The user interface was developed, and the hybrid data exchange protocol was implemented in software.

РЕФЕРАТ

на роботу «Система управління та моніторингу безпілотного літального апарату. Підсистема моніторингу (компл.). »

Ключові слова: НАЗЕМНА СТАНЦІЯ УПРАВЛІННЯ, ПРОТОКОЛ ОБМІНУ ДАНИХ, СИСТЕМА КЕРУВАННЯ, БЕСПІЛОТНИЙ ЛІТАЛЬНИЙ АПАРАТ, ЛІНІЯ УПРАВЛІННЯ , ЛІНІЯ ЗАХИСТУ, ІНТЕРФЕЙС КОРИСТУВАЧ, .

Пояснювальна записка: кількість сторінок – , кількість рисунків – , кількість використаних джерел – .

Об’єктом дослідження є система моніторингу БПЛА, протокол обміну даних та лінія управління БПЛА.

Мета роботи – розробка підсистеми моніторингу та управління БПЛА , модернізувати захист лінії управління БПЛА, розробити інтерфейс користувача.

Проведено дослідження, за результатами якого було вибрано протокол та критерії захисту БПЛА. Розроблено алгоритм, який забезпечує встановлення факту належності прийнятого радіовипромінювання до класу радіосигналів систем дистанційного керування БПЛА з імпульсно-позиційною та імпульсно-кодовою модуляціями.

Наведено результати практичного тестування можливих рішень та наведено результати функціонування розроблених компонентів зв'язку.

Розроблено інтерфейс користувача, та програмно реалізовано гібридний протокол обміну даних.

CONTENT

GLOSSARY

UAV – Unmanned Aerial Vehicle.

CO – Control Object

ACS – Automated Control System.

OBDC - On-board digital computer.

GCS - Ground control station.

IAW - Instructor's automated workplace.

GPS - Global Positioning System.

ADS-B - Automatic dependent surveillance-broadcast.

TCP - Transmission Control Protocol

INTRODUCTION

Recently, unmanned aviation has been rapidly developing. The development of unmanned aerial systems (UAS) based on unmanned aerial vehicles (UAVs) is currently carried out by almost all industrialized countries of the world. Until recently, UAVs had a military purpose, now the use of UAVs is effective both in military and civilian tasks, for example, in combating the consequences of emergencies, natural disasters, agricultural applications, reconnaissance and aerial photography.

The impetus for the development of unmanned aviation worldwide was the need for light, relatively cheap aircraft with high maneuverability characteristics and capable of performing a wide range of tasks. Unmanned aerial vehicles are successfully used in military operations around the world, and at the same time they successfully perform civilian tasks. Today, most of the existing unmanned aerial vehicles are piloted manually, using remote controls operating on radio channels. When manually piloting UAVs, there are difficulties associated with pilot training, insufficient operating range, and weather restrictions. UAV control is the task of a well-trained professional. For example, in the U.S. Army, UAV operators become active duty Air Force pilots after a year of preparation and training. In many aspects, it is more difficult than piloting an aircraft and, as is known, most accidents of unmanned aircraft are due to pilot-operator errors and mechanical failures. According to the official data provided for 2012, 70 unmanned aircraft crashed in the US Air Force.

Purpose of work: Develop UAV control interface and information systems monitoring interface. Implement the data transfer protocol. Improve the protection of the UAV control line.

CHAPTER 1. UAVs AND THEIR CHARACTERISTICS, ANALYSIS OF UAVS AS AN OBJECT OF RESEARCH.

Drone technology has long been used by defense organizations and tech-savvy consumers. However, the benefits of this technology go beyond these sectors.

With the increasing availability of drones, many of the most dangerous and highest paying jobs in the commercial sector are ripe for the transition to drone technology. Use cases for secure, cost-effective solutions range from data collection to delivery. And as autonomy and collision avoidance technologies improve, so does the ability of drones to perform increasingly complex tasks

1.1. UAVs and features of their use.

According to the global market for business services generated by drones, it is valued at more than $150 billion. And as more corporations try to exploit these commercial opportunities, investment in drones has grown.

Drone or UAV (unmanned aerial vehicle) usually refers to an unmanned aerial vehicle operating with a combination of technologies, including computer vision, artificial intelligence, object avoidance technology, and others. But drones can be land or sea vehicles that operate autonomously.

UAVs are devices that are operated without a crew. The main components of the UAV are: an air platform with a special landing system, a power plant, a power source for it, a power system, on-board radio-electronic equipment (on-board control equipment and electronic elements of the UAV). target load). The on-board equipment consists of an on-board electronic computer or special processors, a radio navigation system signal receiver, an altimeter, a vertical gyroscope, an on-board communication and data transmission system, and a steering gear.

To ensure the tasks of the UAV must contain in its composition:

- Devices for obtaining view information:
- Satellite navigation system (GLONASS/GPS);
- Radio link devices for view and telemetric information;
- Devices for command and navigation radio link;
- Command information exchange device;
- Information exchange device;
- On-board digital computer (OBDC);
- Image information storage device

For the practical application and development of UAVs, it is important to study the issue of their classification. The main classification features are: by the scale of tasks to be solved by mass, by flight duration, by practical flight ceiling, by type of aircraft, by basing, by use, by type of control system, by flight rules, by type of wing, by direction, by type , by fuel system, by fuel tank type, by number of uses and range. The classification of known UAVs is given in Table. 1.1

Sign	Kinds
According to the scale of the tasks being solved	• Operational and strategic • Tactical • Operational and tactical
By mass	• Small-sized • Medium-sized • Large-sized • Heavy
By flight duration	• Short-term • Medium duration • Of great duration
Behind the practical ceiling of flight	• Short • Medium height • Heights • Stratospheric

By type of aircraft	• According to the aircraft aerodynamic diagram • According to the helicopter aerodynamic scheme • Lighter than air
By basing	• On the ground • Marine • The cosmic
By use	• Military • Civil (state, private, commercial)
By type of control system	• Automatic • Remotely piloted • Remotely controlled • Remotely controlled by the aviation system
According to the rules of the flight	• Visual • Instrumentation • Visual and instrumental
By wing type	• Fixed • Floating
By direction	• By lifting direction (horizontal, vertical, multi-lifting) • By landing direction (horizontal, vertical, parachute, mast, non-stop, multi-launch)
By type	• By lift (aerodrome, launch, deck, water, manual, atypical lift, multi-lift) • By landing (airport, point, deck, water, non-stop, atypical landing, multi-landing)
By fuel system	• Single filling stations • Filling stations (ground, platform (marine, on-board))
By type of fuel tank	• Basic • Basic and reserve

By the number of uses	• Disposable • Reusable
By radius of action	• Near radius • Small radius • Medium radius • Long radius • Long flight range

1.2 Ground control stations of UAVs

UAV ground control station (GCS) is a land or sea control center that provides means of human control Unmanned aerial vehicles (UAVs or "drones"). It can also refer to a missile control system within or above the atmosphere, but it is usually described as a Mission Control Center.

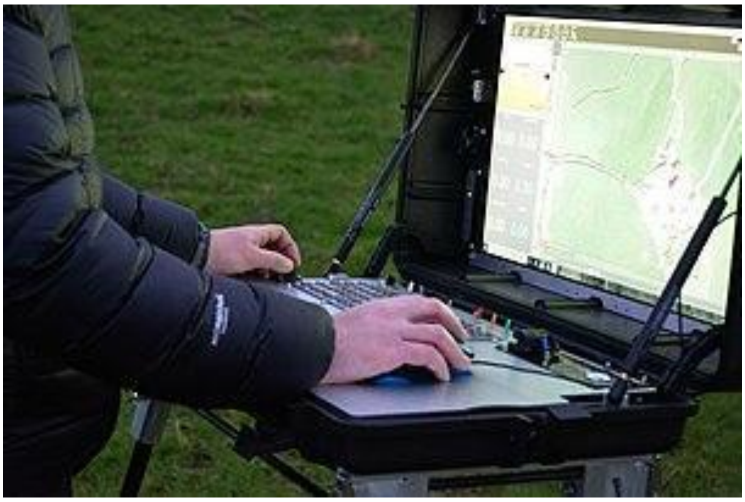


Fig. 1.1. An example of a portable ground control station of the UAV

Main purpose:

1. Plan the mission — after receiving information about the UAVs environment and the details of the mission, the GCS performs the appropriate

calculations to determine the trajectory of the course and the time frame of the maneuvers.

2. Provide navigation and positioning - the GCS reacts to changes in the environment in case of unforeseen situations during the flight of the UAV in relation to the planned trajectory.

3. Establish communication and data collection - during the mission, data from the UAV is transmitted to the GCS for their analysis and reporting to the operator.

The GCS supports network operation, and thanks to its architecture and compatibility with Ethernet, it can be easily and efficiently integrated into existing network systems, distributing the necessary information between users, for example, transmitting intelligence information from the GCS to the headquarters and receiving a combat mission from the headquarters.

Appointment:

- training of UAV and target load operators (TL), support of their skills and abilities in UAV piloting, navigation, operation of UAV systems and equipment in flight;
- carrying out objective supervision of floorings.

The simulator is a hardware and software complex with special software integrated into the aviation simulator for full-time automated workplaces of UAC operators and an instructor, which functions under the control of the Linux operating system, and is designed according to the modular principle, which ensures the possibility of its use for training operators of various UACs, taking into account the necessary the number of jobs for students.

Features of the simulator:

- training with full and abbreviated UAC calculations;
- automated evaluation of actions of UAC calculation persons;
- preparation, input, storage, display, adjustment and recording of the flight task;

- registration, processing, storage, display of parametric and species information obtained during the flight simulation of UAVs;
- UAV management during flight simulation;
- support and determination of the coordinates of the selected objects in the frame of the video stream on the control center operator's computer;
- introduction and simulation of UAV failures, the influence of external factors on UAV flight from the instructor's ARM;
- display of pilotage and navigation, parametric and species information during flight simulation, as well as operator actions on the instructor's IAW;
- processing, storage, display of information obtained as a result of real UAV flights;
- automated and manual processing of materials of objective control of real flights based on registered telemetry information of UAC for its further analysis and documentation.

To improve the skills of UAV operators, a course of training tasks has been implemented in the simulator. The instructor can simulate the occurrence of various emergency situations in the simulator when performing training tasks.

Composition of the simulator:

- instructor's automated workplace (IAW);
- IAW of the operator of the unmanned aerial vehicle;
- IAW of the operator of the target load;
- IAW of the head of the HAC calculation;
- IAW for flight data analysis;
- power supply systems;
- a set of operating documentation with general and special software.

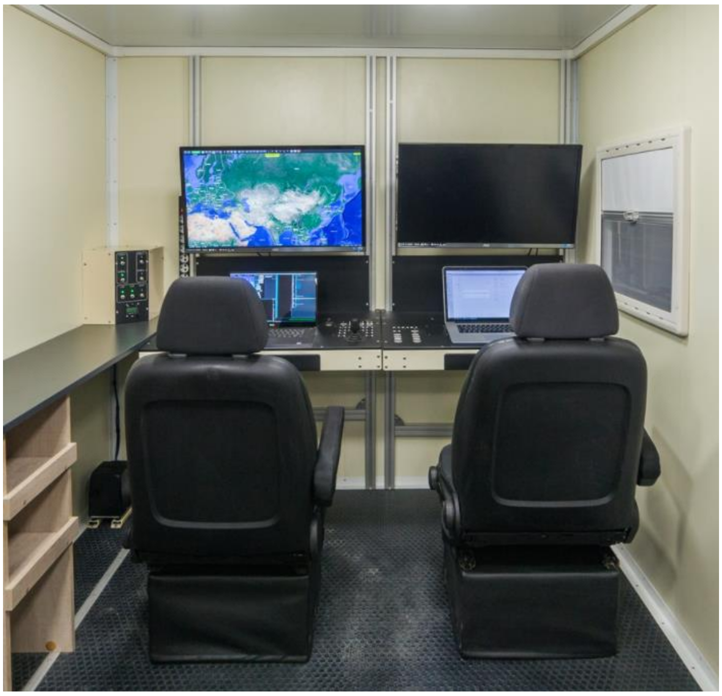


Fig. 1.2. An example of a full-fledged UAV ground control station

1.3. UAV monitoring subsystem.

The protocol is designed for binary telemetry. It can be used for systems with limited resources and limited bandwidth of communication channels. It describes the interaction between the small unmanned aerial vehicle (SUAV) system and the ground control station (GCS), as well as their components.

A system has a unique ID, and each component within that system has its own unique ID. These IDs will be used for routing/addressing.

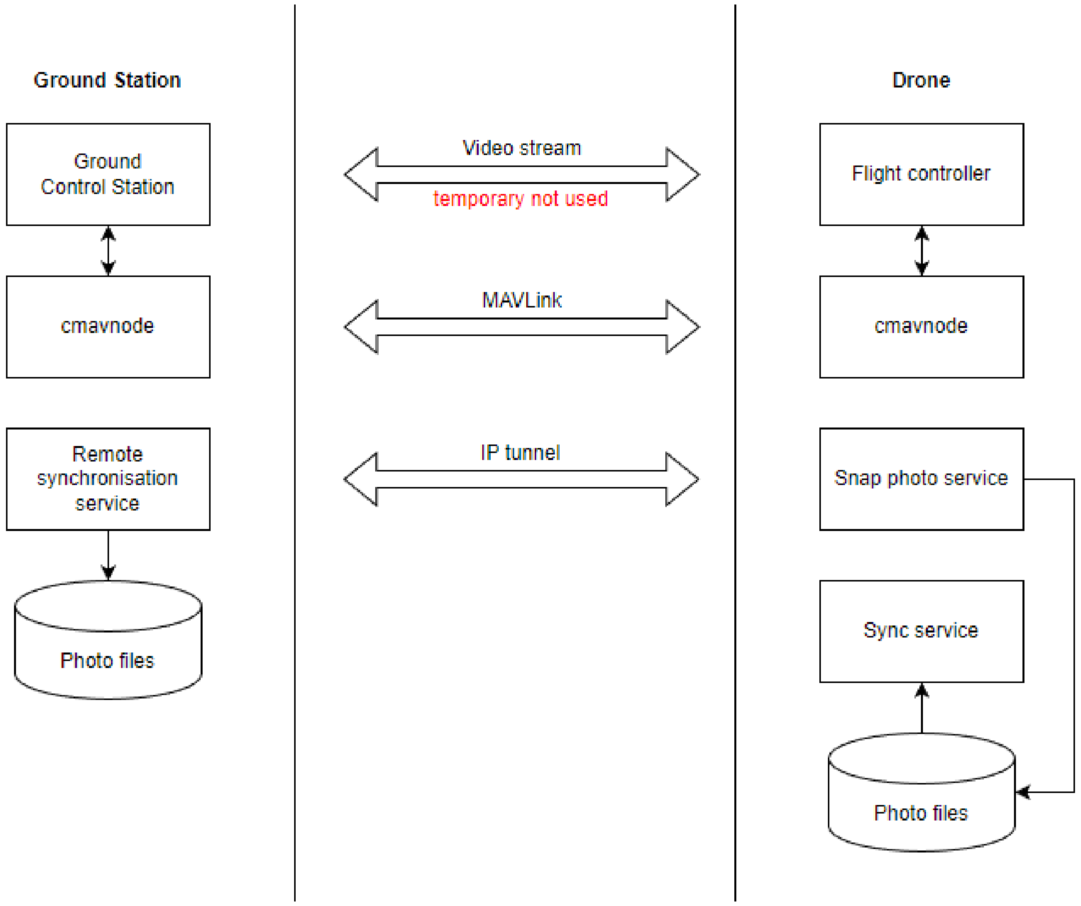


Fig. 1.3 – Functional diagram of the UAV control subsystem

MavLink is the main communication protocol between GCS and UAV. It allows communication over various transport and physical layer protocols such as UDP and TCP, WiFi and Ethernet.

MavLink is also a binary serialization protocol, which means first converting the data into a sequence of bytes and then converting it back to data for the receiver.

This quality makes the average message size of this protocol very low [1] due to high compression of the output data. Each MavLink v2 message consists of a header and a message body, weighing from 25 to 270 bytes. The messages themselves are divided into two main types [15]: 1. Status messages - sent from the UAV to the GCS and contain information about the position of the UAV, data from sensors and additional information required by the GCS. 2. Management messages - sent from the GCS to the UAV and contain requests for certain actions.

CHAPTER 2. INFORMATION INTERACTION SUBSYSTEM OF THE GROUND CONTROL STATION AND THE UNMANNED AIRCRAFT

2.1 Problems that arise with the management and security of UAVs

The main characteristics of drones include small size, low cost, and ease of maneuverability and maintenance. This, in turn, made them the best choice for criminals. In addition, terrorists have begun to look at the use of these drones to carry out attacks, mainly due to the nature of drones, which makes them less prone to detection.

In fact, drones can be armed and modified to carry dangerous chemicals or be armed with explosives to attack critical infrastructure. Moreover, drones carrying explosives can be detonated around people who are gathering in hard-to-reach places. This makes it easier for the terrorist to complete the task, especially since drones provide suicide bomber stealth with aircraft range. Military analysts are concerned about the use of drones against the US for espionage purposes. This is due to the fact that ISIS was able to rearm commercial drones and make them suitable for combat operations over Iraq and Syria.

Therefore, security does not always mean security and vice versa. Outside the military home, civilian drones/UAVs can also malfunction and crash into a neighboring home or group of people, causing property/property damage and human injury/fatalities ranging from trauma/blunt force trauma, deep cut injuries. As a result, the main security concerns are listed below:

· Lack of safety design features: This can cause drones to get out of control and fly away aimlessly and autonomously without being able to disable or regain control.

· Lack of technology and operational standards: especially with regard to accident prevention mechanisms, which will result in the inability of the UAV to recognize and identify aircraft and airborne objects and avoid them.

· Prevent signal bending: This makes the UAV prone to hacking, hijacking and GPS/Signal Jamming as part of cyber terrorism or cybercriminal activities, mainly due to the fact that the UAV's command and control operations center is susceptible to exploitation.

· Lack of government regulation and awareness: especially in terms of security practices and functions that ensure the secure integration of UAs into the national airspace domain [21].

UAVs and drones are perceived as viable and vital information security threats. Many UAVs have serious design flaws and most of them are designed without wireless security protection and channel encryption.

· Tendency to spoofing. Analysis of the configuration and flight controllers of UAV models with multiple rotors revealed many weaknesses. They are associated with telemetry links that transmit data to/from the drone via a serial port connection, especially due to the weak nature of the connection, which in most cases is not encrypted. Experiments performed have shown that with the help of GPS spoofing, information can be easily hijacked, changed or introduced [22]. This vulnerability in the data link allows interception and moonlighting, giving hackers full control over the drone.

· Tendency to be infected with malware. Communication protocols are included in the UAVs to allow users to control drones with a wireless remote control such as tablets, laptops and mobile phones. However, this technique has proven to be dangerous, as it allows hackers to create a loopback TCP payload by injecting it into the drone's memory, which will covertly install malware on systems running at ground stations.

· Tendency to interfere and intercept data. Telemetry channels are used to monitor vehicles and facilitate the transmission of information through open, unsecured wireless transmission, making them vulnerable to various threats. This includes data interception, malicious data entry, and alteration of pre-established flight paths. This allows many infected digital files (videos, images) to be installed and pasted from the drone to the ground station. Another vulnerability was discovered and associated with the UAV communication module, which uses wireless communication to exchange data and commands with the ground station [23].

· Tendency to manipulation. Because drones fly pre-programmed and predetermined routes, manipulation can occur that can have serious consequences. Ranging from hijacking high value cargoes to redirecting the UAV to deliver explosives, bioweapons, or other terrorist payloads via RF or GPS spoofing, allowing the attacker to gain control of the drone by sending out fake signals, or jamming its target to malfunction. · Tendency to technical problems. Many drones suffer from various technical problems. This includes application errors such as a connection failure between the user's device and the aircraft, causing it to either crash or take off. Other issues are related to the lack of a stable connection, especially for complex natural reasons; battery life, resulting in a very limited flight time before cooking again.

· Prone to Wi-Fi interference. Drones can also be hijacked by sending a deauthentication process between the access point and the device controlling the drones, which can be done temporarily or permanently, such as interfering with the drones' intended frequency and luring it into connecting to a Wi-Fi hacker; this can be done by installing and configuring raspberry-pi to do this.

2.2 Overview of existing solutions and comparative analysis

Wireless networks suffer from several security threats. Recently, Intrusion Detection Systems (IDS) have been deployed to detect malicious UAV/UAV activities and detect suspicious attacks that may be targeted at them.

Typically, an IDS monitors incoming and outgoing network traffic and analyzes them for anomaly detection. Their goal is to detect and identify cyber-attacks by examining data checks (footprints) collected in different parts of the network. Various IDS approaches to protect drone networks from intruders are presented below. 1. Rule based intrusion detection. The UAV domain uses intrusion detection systems based on the rules for protecting communication between the aircraft and the ground station. The goal is to detect spoofed data entry attacks, especially those targeting signal strength. They proved that attackers can be detected within 40 seconds. [24]

Mitchell and Cheng presented a specification-based detection technique to protect the UAV system from various types of cyber attacks. The authors relied on UAV-IDS based rules of conduct. The rules of conduct were built around specific attack patterns, including reckless, random, and opportunistic attacks. This minimized detection errors, including false positives and false negatives, with a critical trade-off between safety and UAV performance. In [25], Mitchell et al. introduced BRUIDS, an adaptive behavioral rule-based behavioral system that detects malicious UAVs in airborne systems.

The authors also examined the effectiveness of BRUIDS on reckless, random, and opportunistic attacker behavior to quickly assess UAV survivability against malicious attacks. The simulation results showed that BRUIDS achieves a higher level of detection compared to the multi-objective anomaly-based IDS approach and a lower false positive rate.

However, rule-based IDSs suffer from managing their complexity, requiring human intervention in the configuration of the rules. Also, this type of IDS is not capable of detecting unknown attacks. 2. Intrusion detection based on signatures. In [26], Kacem et al. introduced the ADS-B intrusion detection system to protect the aircraft from cyber-attacks directed at the ADS-B message. This structure is based on signature detection methods that analyze the GPS position of the aircraft. In [27] Casals et al. developed a detection scheme using biological inspiration to detect cyberattacks targeting overhead networks. However, like a rule-based IDS, a signature-based IDS cannot detect unknown attacks or attacks using dynamic signatures. 3. Anomaly based detection. Anomaly-based IDS in the UAV domain is mainly used to prevent jamming attacks. In [28] Rane et al. introduced an anomaly-based learning algorithm to protect UAV nodes from DoS and DDoS attacks.

In [29] Lu et al. introduced a learning-enhanced motor temperature anomaly detection system for UAVs that prevents drone motors from running at abnormal temperatures, using DS18B20 sensors for temperature recording and a raspberry-pi processor for processing. This system offers the possibility of avoiding engine damage by landing the drone in case of overheating; however, this does not completely prevent the problem. Experimental results reveal the possibility of safe control of a drone based on sensor information.

In [30] Condomines et al. introduced a hybrid IDS based on traffic spectrum analysis and a reliable controller for anomaly estimation in UAV networks in the Flying Ad Hoc Network (FANET).

This technique was targeted at distributed DoS attacks and its effectiveness was tested on real-time traffic. The results showed accurate detection of different types of anomalies. However, further testing is still required to ensure its effectiveness. In [31] Sedjelmaci et al. introduced the Intrusion Detection and Response System (IDRF) to protect the UAV network from data integrity and network availability attacks, and to protect the UAV-supported VANET from harmful threats. The authors noted that the proposed structure is unique as a hybrid detection technique for UAV networks.

In [32] Lauf et al. introduced a decentralized anomaly-based detection technique using peak detection and cross-correlation methods. In fact, Maxima detection systems (MDS) provide characterization of one or zero suspicious hosts, while cross-correlation detection (CCD) methods detect multiple intrusions. However, their approach suffers from inaccuracies about false positives and false negatives. In addition, Sedjelmaci et al. introduced a hierarchical intrusion detection and response scheme to improve the security of UAV networks from destructive cyber attacks such as false information propagation, GPS spoofing, black hole and gray hole substitution and attacks.

This scheme works at the UAV and ground station level to detect malicious network anomalies. The simulation results showed a high detection rate of 93.3% and a low false positive rate of less than 3% with low communication costs. In [33], Mitchell et al. presented a specification-based IDS for protecting sensors and actuators embedded in UAC. To evaluate the effectiveness of the IDS solution, it was tested on UAVs to investigate the impact of the attacker's behavior

Given that drone network gateways can operate with some limitations (fog nodes), there is a need for a lightweight host-based anomaly detection technique requiring mini-computing resources. This can be achieved with a simple machine learning technique or a statistical approach with as few features as possible. The structure of a robust IDS should be based on a hybrid approach where rule or signature based approaches are used for known attacks and an anomaly based approach to detect anomalous behavior. Such a system would depend on experts in machine learning and human security

Due to the increase in the number of drone/UAV frame intercepts, various UAV communication solutions have been introduced. In [34] Zhang et al. the security issues of the physical layer in UAV communication systems were considered and an iterative algorithm based on block descent of coordinates and successive convex optimization methods was presented. The simulation results showed a significant improvement in the level of secrecy of UAV communication systems. Written by Zhang et al. applied these algorithms to solve the broadcast, line-of-sight and wireless ground-to-ground problems associated with fifth-generation (5G) wireless networks. The simulation results showed an improvement in the level of privacy of UAV communications to the ground (U2G) and terrestrial communications to the UAV (G2U). In [35] Cui et al. also considered the transmission nature of air-to-ground direct view wireless channels and solved it based on the physical layer using the UAV mobility trajectory design. The authors presented an iterative suboptimal algorithm using the block coordinate descent method, the S-procedure, and the sequential convex optimization method.

The simulation results showed a significant improvement in the average worst case safety. Zhao et al. introduced a cached UAV secure transmission scheme in ultra-dense small cell base stations (SBS) based on interference equalization to offload traffic through wireless backhaul and improve coverage and speed by generating jamming signals to thwart any potential attempt at listening. The simulation results showed the effectiveness of their methods. Liu et al. investigated UAV-enabled secure communication with cooperative UAV jamming and presented an iterative algorithm that provides an efficient solution to the minimum privacy maximization problem by jointly optimizing transmission power, UAV trajectory, and user planning variables. Numerous results have shown that the algorithm outperforms the main methods.

Liu et al. studied the problem of security in UAV-supported communication systems and presented a scheme for secure transmission of an UAV listening channel using a multi-antenna source transmitting to one UAV antenna in the presence of a full-duplex active eavesdropper. The multi-antenna source transmits artificial noise signals along with data signals to prevent full duplex eavesdropping and jamming capabilities.

In addition to modulation techniques, it is important to encrypt communications between drones and UAVs. In this context, various cryptographic solutions have recently been introduced, including message encryption and authentication. Since most drone standards are required to provide secure communications, the focus has been on how to develop a lightweight authentication and message encryption algorithm. It can also be done in a way that preserves source authentication in addition to the integrity and confidentiality of the transmitted data. Existing cryptographic algorithms to secure drone communications can be applied to secure communications for drones used for civilian applications. Moreover, secure communication protocol (eCLSC-TKEM) between drones and intelligent objects is preferred by 1.3, 1.5 and 2.8 times over other protocols, including protocols in CLSC-TKEM from Seo, CL-AKA from Sun and CL -AKA by Yang.

In addition, in [36] Sharma et al. introduced a highly secure Functional Encryption (FE) technique to protect the UAV-supported Heterogeneous Network (HetNet) in dense urban areas from harmful activities, and protect users' critical data through encryption; however, this solution requires further improvements. In addition, the creator of Chen et al. introduced an authentication scheme that traces and protects privacy (TPPA) for UAV communication control systems. TPPA integrates Elliptic Curve Cryptography (ECC), digital signature, hashing and other cryptography mechanisms for UAV applications.

It provides confidentiality, confidentiality, integrity, availability, anonymity and repudiation, especially against DoS and replacement, with low computational and communication costs. Working over long distances on battery-powered devices, the security of drone communications requires lightweight cryptographic algorithms and protocols. Recently, new cryptographic algorithms with single round functions or few iterations have been introduced. Moreover, existing privacy-preserving authentication protocols can use such lightweight cryptographic algorithms for minimal latency. Also, physical layer parameters can be used for multi-factor authentication.

2.3 Protocol structure and message description

The basic entity of the protocol is a packet that has the following format (Fig. 2.1):

Range 8-263 bytes

pstx	*pseq*	*psysId*	*pcompId*	*plen*	*pmsgId*	*ppayload*	*pchecksum*

The elements of the package are:

- pstx - start b package; message start character;
- pseq - packet counter to detect message loss;
- psysId - identifier of the sending system;
- pcompId - ID of the sending component;
- plen is the length of the payload;
- pmsgId - message identifier, which determines what data will be in the payload of the packet;
- ppayload - payload of the package;
- pchecksum - lower and upper b. Contains the checksum of the packet.

More details are shown in Table 2.1

Table 2.1: Package components

Byte ID	**Marker**	**Type**	**Content**	**Value**
0	pstx	*uint*8	Market launch package	0*xFE*
1	pseq	*uint*8	Package number	0 − 255
2	psysId	*uint*8	System ID	1 − 255
3	pcompId	*uint*8	System component ID	1 − 255
4	plen	*uint*8	Payload length	0 − 255
5	pmsgId	*uint*8	Message ID	0 − 255
n байт корисного навантаження	ppayload	[ ]*uint*8	Payload data	Масив значень 0 − 255

(*n* + 1) to (*n* + 2)	pchecksum	*uint*16	Checksum (low byte, high byte)	0 – 65555

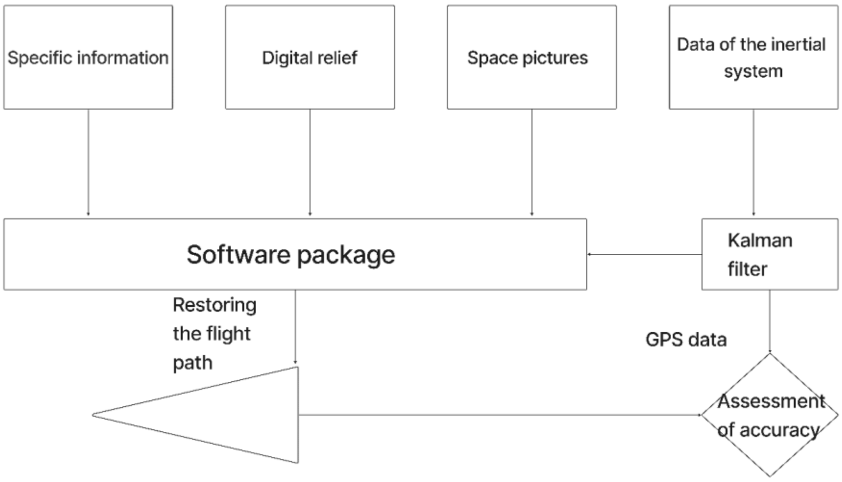


Fig. 2.1: General diagram of the monitoring subsystem

The protocol describes the algorithm and structure of the exchange of GCSand UAV.

The protocol describes:

1. Operations to clear, download and jump missions;
2. Setting/receiving a message about changing the current element of the mission;
3. Types of messages and enumerations for the exchange of mission elements;
4. Mission Elements (ME), which are general commands for network systems.

Each message has its own ID and data set. The identifier is placed on the packet in the message identifier section (msgId), while the data is entered in the payload section.

Messages

Messages mission RequestList

Initializing the loading of a list of mission elements with a system or component. Requests a complete list of mission elements from the system/component. The structure is shown in Table 2.2

Table 2.2: MissionRequestList message structure

Field Name	Type	Value	Description
targetSystem	uint8		System ID
targetComponent	uint8		Component ID.

MissionCount message

Sending the number of mission items. Used to initialize mission loading or by requesting a missionRequestList when loading a mission The structure is shown in Table 2.3

Table 2.3: Structure of the missionCount message

Field Name	Type	Value	Description
targetSystem	uint8		System ID
targetComponent	uint8		Component ID.
missionCount	uint8		The sequence number of the mission element

missionRequest message

Request for mission item data for a specific message sequence number sent by the recipient using the missionItem message. Used to download and download the mission. The structure is shown in Table 2.4

Table 2.4: MissionRequest message structure

Field Name	Type	Value	Description
targetSystem	uint8		System ID

targetComponent	uint8		Component ID.
qqsequence	uint8		Serial number of the sequence

MissionItem messages

Messages that encode the mission element (command) (defined in the cmdEnum enum) . Used to download and download the mission. The structure is shown in table 2.5.

Table 2.5: MissionItem message structure

Field Name	Type	Value	Description
targetSystem	uint8		System ID
targetComponent	uint8		Component ID.
qqsequence	uint16		Waypoint ID (serial number). It starts from scratch. Monotonically increases for each route point, without breaks in the sequence (0,1,2,3,4).
frame	uint8	frameEnum	Waypoint coordinate system
current	uint8		The flag is used during download. Determines whether the mission item is current.
autocontinue	uint8		The flag is used to automatically move to the next waypoint when the current command completes.
cmdId	uint16		Team ID.
param1	float		Parameter №1.
param2	float		Parameter №2
param3	float		Parameter №3.
param4	float		Parameter №4.

param5 (x)	int32		X coordinate in local coordinate system or longitude in earth system coordinates for cmdNav*.
param6 (y)	int32		Y coordinate in local coordinate system or height in earth system coordinates for cmdNav*
param7 (z)	float		Z coordinate in local coordinate system or latitude in earth system coordinates for cmdNav*.

missionResponse message

Response - The response of a system or component. When the system completes a mission operation (for example, after loading all mission elements, engaging the autopilot, and reaching the first waypoint). Encodes the success or failure of the execution of the command described in the missionResult enum. The structure is shown in table 2.6

Table 2.6: Structure of the missionResponse message

Field Name	Type	Value	Description
targetSystem	uint8		System ID
targetComponent	uint8		Component ID.
type	uint8	missionResultEnum	Mission result

missionCurrent message

A message containing the sequence number of the current mission element. Used when setting or changing a mission element. The structure is shown in Table 7

Table 2.7: Structure of the missionResponse message

Field Name	Type	Value	Description
qqsequence	uint8		The sequence number of the mission element

MissionSetCurrent messages

Sets the current mission element by serial number (goes to the given mission element along the shortest path). The structure is shown in Table 8.

Table 2.8: Structure of the missionSetCurrent message

Field Name	Type	Value	Description
targetSystem	uint8		System ID
targetComponent	uint8		Component ID.
qqsequence	uint16		The sequence number of the mission element

StatusText messages

This message is sent to inform systems when a request to set the current mission element fails the missionSetCurrent command. The structure is shown in Table 9.

Table 2.9: Structure of the statusText message

Field Name	Type	Value	Description
severity	uint8		Degree of seriousness.
text	char[50]		A textual status message, with no null terminating character.
id	uint16		A unique (opaque) identifier. Can be used for reassembling a logical message with a long status text from a sequence fragments. A value of zero indicates that it is

			the only fragment in the sequence, and the message can be sent immediately.
chunkQqsequence	uint8		Sequence number of the message part; indexing from scratch. Any null character in the text field means that this is the last fragment.

MissionClearAll message

Message sent to clear/delete all mission items stored in the system. The structure is shown in Table 2.10.

Table 2.10: MissionClearAll message structure

Field Name	Type	Value	Description
targetSystem	uint8		System ID
targetComponent	uint8		Component ID.

missionItemReach message

The message sent by the system when a new waypoint is reached. Used to track progress. The structure is shown in Table 2.11.

Table 2.11: MissionIteamReach message structure

Field Name	Type	Value	Description
qqsequence	uint8		Sequence number

Enumeration

The missionResultEnum enum

The result of executing the mission element in the missionResponse message. Used to indicate the reasons for the success or failure of an operation (for example, to upload or download a mission). The elements of the enumeration are listed in Table 2.12

Table 2.12. MissionResultEnum enum structure

Value	Field Name	Description
0	MissionAccepted	Mission element
1	missionError	General error (Not accepting mission command right now).
2	missionUnsupportedFrame	The coordinate system is not supported.
3	missionUnsupported	The command is not supported.
4	missionNoSpace	The mission item has exceeded the available storage space
5	missionInvalid	One of the parameters has an invalid value.
6	missionInvalidParam1	Parameter 1 has an invalid value.
7	missionInvalidParam2	Parameter 2 has an incorrect value.
8	missionInvalidParam3	Parameter 3 has an incorrect value.
9	missionInvalidParam4	Parameter 4 has an incorrect value.
10	missionInvalidParam5X	Parameter 5 (X) has an incorrect value.
11	missionInvalidParam6Y	Parameter 6 (Y) has an incorrect value.
12	missionInvalidParam7Z	Parameter 7 (Z) has an incorrect value.
13	missionInvalidQqsequence	Mission element received out of turn.
14	missionDenied	Do not accept any mission commands from this comms partner.
15	missionOperationCancelled	The current mission operation has been canceled (for example, loading a mission item, loading mission element).

The frameEnum enum

Coordinate systems for the speed/position/acceleration data in the message. The elements of the enumeration are listed in Table 2.13.

Table 2.13: FrameEnum enum structure

Value	Field Name	Description
0	gpsFrame	Global coordinate system (WGS84) along with altitude. x-coordinate: longitude, y-coordinate: altitude, z-coordinate: latitude.
1	earthFrame	Terrestrial coordinate system with a fixed reference point. x-coordinate: north, y-coordinate: down to the center of the earth, z-coordinate: east.
2	bodyFrame	The coordinate system is linked to the aircraft. x-coordinate: Forward, y-coordinate: up, z-coordinate: right. The longitudinal axis of the coordinate system is aligned with the aircraft body.

Mission elements

The main interaction between the GCS and the SUAVis carried out through the mission element. Each mission element has its own unique identifier and data set. The miss-this element is the GCS team in the SUAV, and is divided into the following types:

- Preflight command (cmdPreflight*). Used only before takeoff of the SUAV;
- Nav commands (cmdNav*) navigation and movement;
- Do commands (cmdDo*) for immediate actions such as changing the speed or activating the steering gear.
- Condition commands (cmdCondition*) to change the execution of the mission depending on some condition (for example, pause the mission until the command is executed).

ME are encoded/transmitted in the missionItem message shown in Table 2.5. This message includes the command identifier, 7 command parameters and 5 message service parameters. The first four parameters are used as the main parameters of commands and depend on their structure. The following parameters (x, y, z) 3 parameters are positional information and are used mainly for navigation commands, but can be used in other messages as needed.

Operations

Loading mission elements

The process is shown in Fig. 2.1 consists of the following sequence:

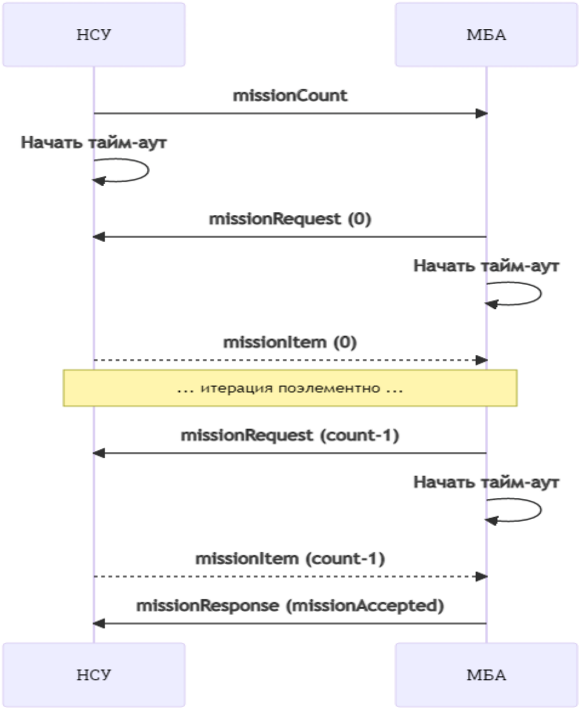


Figure 2.2: Timing diagram for loading parts of the mission.

1. The ground station requests the missionCount on the SUAV. The request determines the number of loaded mission elements (missionCount parameter). After sending the message to the GCS, a timeout is started to wait for a response from the SUAV.

2. The SUAVreceives the message and responds with a missionRequest message, requesting the first element of the mission (qqsequence == 0). Drone starts timeout to wait for missionItem station ground response message

3. The ground station receives the missionRequest message and responds with the requested mission item with a missionItem message.

4. SUAVE and ground repeat the missionRequest/missionItem cycle, iterating qqsequence until all items have been loaded (qqsequence == count - 1).

5. The SUAVE , after receiving the last mission element, matches the GCS missionResponse messages with the type field equal to missionAccepted, which corresponds to the successful loading of all mission elements. (SUAV must set the new mission as the current mission, removing the original data; SUAV considers the download complete).

6. The ground station receives a mission according to the mission adopted to indicate that the operation is feasible.

Note:

• A timeout is set for each message that requires a response (eg missionRequest). If the timeout expires and no response is received, the request must be resent.

• ME must be obtained in order. If a mission element is received out of turn, the expected mission element must be re-requested by the GCS (the out of turn mission element is discarded).

• An error can be reported in response to any request with a MissionResponse message containing the error code. This should cancel the operation.

• The process above shows that a mission item is packaged in a missionitem message.

Downloading mission elements.

The process shown in Fig. 2.2 is similar to loading the mission element. The main difference is that the ground station sends a missionRequest which triggers the SUAVE autopilot to respond with the current number of mission items. This starts a loop where the ground station requests ME and the drone delivers them.

Note:

• A timeout is set for each message that requires a response (eg missionRequest). If the timeout expires without a response, the request must be sent. howl again.

• ME must be obtained in order. If a mission element is received out of order, the expected element must be re-requested (the mission element out of order is discarded).

• An error can be reported in response to any request with a MissionResponse message containing the error code. This should cancel the operation.

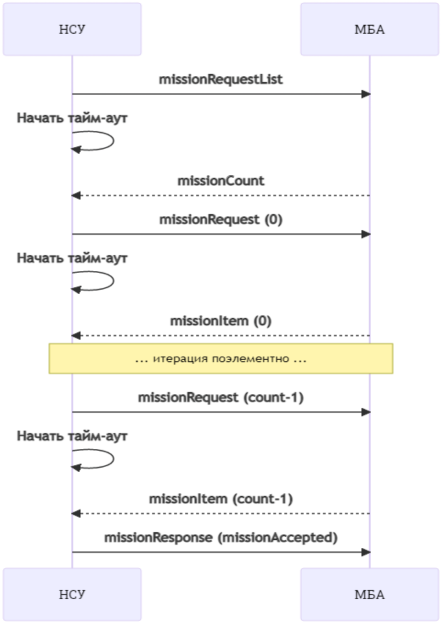


Fig. 2.3: Time chart for loading mission elements.

Setting the current mission element.

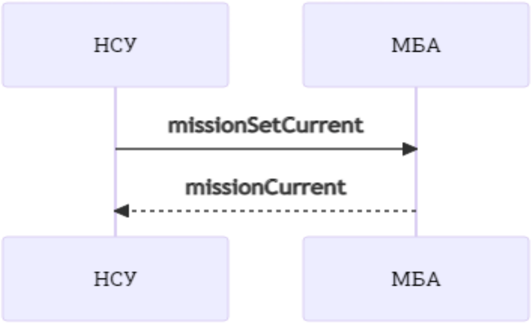


Fig. 2.4: Timeline of setting the current mission element.

The process is shown in Fig. 2.3 consists of the following steps:

1. The GCS sends a missionSetCurrent message with the sequence number of the mission element (qqsequence).

2. SUAVE receives the message and tries to update to the new mission element's serial number:

- On success, SUAVE should start broadcasting a missionCurrent message containing the mission element sequence number (qqsequence).
- On failure, SUAVE should start broadcasting a statusText message, for the error reason described.

Note:

- There is no specified timeout for the missionSetCurrent message.
- Message acknowledgment is done by broadcasting the mission/system status unrelated to the outgoing message. Accordingly, this approach differs from errors in other operations. This is because success/failure is important to all clients on a mission.

Tracking mission progress.

GCS can track progress by processing response messages from SUAVE:

- The vehicle must broadcast a missionItemReach message whenever a new mission item is reached. The notification contains the serial number of the current mission element.
- The vehicle must also send missionCurrent messages if the current mission element has been changed.

Clearing mission elements.

The sequence of operations looks like:

1. GCS sends missionClearAll

- The GCS timeout begins to wait for a missionResponse message from SUAVE.

2. SUAVE receives the message and purges ME from storage.
3. SUAVE matches missionResponse messages with a missionaccepted result in the type field over the missionresultEnum enum shown in the Table. 2.12.
4. The GCS receives the missionResponse message and clears its storage of mission information.

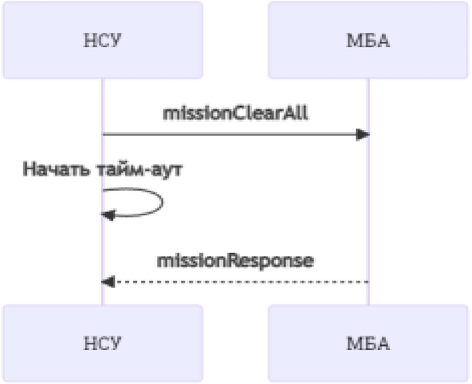


Fig. 2.5: Timeline for clearing mission elements.

Note:

- A timeout is set for each message that requires a response (eg missionClearAll). If the timeout expires and no response is received, the request must be resubmitted.
- An error can be reported in response to any request (in this case simply missionClearAll) with a missionResponse message containing the error code. This should cancel the deal. The GCS mission record must be kept.

Cancellation of mission elements.

Timeouts and attempts. A timeout must be set for all messages requiring a response. If the expected response is not received before the timeout expires, the message must be resent. If no response is received after several attempts, the client should cancel the operation and return to the pending state.

Recommended values for timeout before retry and number of retries:

- Timeout (default): 1500ms;
- Timeout (mission elements): 250 ms;
- Retries (max.): 5.

Errors/Completion. All operations end with a missionResponse message containing the result of the operation (missionResultEnum) in the type field.

On success, the message must contain the missionAccepted type; this is sent by the system that receives the command/data (e.g. SUAVE to download a mission or GCS to download a mission).

The operation can also fail - missionResponse.type is set to missionError or another error code in missionResultEnum. This can happen in response to any message / anywhere in the sequence.

Errors are considered uncorrectable. When an error message is sent, both ends of the system must reset themselves to the pending state, and the mission state on the vehicle must be unchanged.

Note:

- Timeouts are not considered an error.
- ME upload/download out-of-sequence messages resume and are not treated as errors.

Commands.

ppcmdNavWaypoint - Navigate to a marker (waypoint).

ppcmdNavReturnToLunch - Return to the starting point.

ppcmdNavLand - Landing.

ppcmdNavTakeOff - Takeoff.

ppcmdNavLandLocal - Landing on a local point.

ppcmdNavTakeOffLocal - Take off from a local position.

ppcmdNavFollow - Follow a vehicle.

ppcmdNavContinueAndChangeAltitude - Continue moving and ascend or descend to the specified altitude.

ppcmdNavPathPlanning - Offline planning management.

ppcmdNavDelay - Postpone the command for a time (seconds).

ppcmdNavSetYawSpeed - Specifies the angle the vehicle is turning.

ppcmdDoFollow - Follow the target.

ppcmdDoJump - Jump to the specified task (mission).

ppcmdDoChangeSpeed - Change speed or thrust setpoint.

ppcmdDoSetHome - change the start point to the current one.

ppcmdDoSetParameter - Set system parameters.

ppcmdDoSetRelay – set a relay to a condition.

ppcmdDoRepeatRelay - Turn on or turn off the relay for the desired amount with the desired period.

ppcmdDoSetServo - Set the servo to the desired PWM (Pulse Width Modulated) value.

ppcmdDoSetServo - Switches between nominal and desired PWM for the desired number of cycles with the desired period.

ppcmdDoFlightTermination - Terminate flight immediately.

ppcmdDoChangeAltitude - Change the given altitude value.

ppcmdDoSetActuator - Sets the actuator to the desired value.

ppcmdDoLandStart - Command to land.

ppcmdDoGoAround - Autonomous Landing Abort Mission Command.

ppcmdDoReposition - Move UAV to specific WGS84 global position

ppcmdDoMotorTest - Command to test the motor.

ppcmdDoAutotuneEnable - Enable or disable system autotuning.

ppcmdDoGuidedMaster - set master controller ID.

ppcmdDoGuidedLimits - Set managed limits for the joystick.

ppcmdDoSetMissionCurrent - Set the mission element with sequence number qqsequence as the current element. This means that the IBA will continue to move to this mission pointe by the shortest route (without following the intermediate points of the mission).

ppcmdConditionDelay - descent/ascent to the specified height at the specified speed.

ppcmdConditionChangeAltitude - Command to delay the mission status machine.

ppcmdConditionChangeDistance - Postpone the automatic mission to the next waypoint by the desired distance.

ppcmdConditionYaw - Reach the desired yaw angle.

ppcmdPreflightCalibration - Start calibration, this command will only be accepted in preflight mode.

ppcmdPreflightSetSensorOffsets - Set sensor offset.

ppcmdPreflightStorage - give requests for storing parameter values and logging logs.

ppcmdPreflightRebootShutdown - Request to reboot or shut down system components.

ppcmdMissionStart - Start the mission.

Under the manual control protocol (joystick)

The manual control sub-protocol allows the system to be controlled using a "standard joystick" (or a joystick-like input device supporting the same axis nomenclature). The subprotocol implemented by a single manualControl message is shown in Table 2.14. It provides an API for manual control of a vehicle using the standard joystick axis nomenclature, as well as a joystick-like input device. Unused axes can be disabled, and button states are transmitted as separate on/off bits of the bitmask.

Table 2.14: Structure of the manualControl message

Parameter	Type	Description
target	uint8	Managed system.
x	int16	The X axis is normalized to the range [-1000, 1000]. An int16_MAX value indicates an invalid value. Usually corresponds to the movement of the joystick forward (1000) - backward (-1000) and roll of the vehicle.
y	int16	Here is Y normalized to the range [-1000, 1000]. An int16_MAX value indicates an invalid value. Usually corresponds to the movement of the joystick to the right (1000) - to the left (-1000) and to search the vehicle.
z	int16	Here is Z normalized to the range [-1000, 1000]. An int16_MAX value indicates an invalid value. Usually corresponds to the movement of the joystick up (1000) - down (-1000) and the pitch of the vehicle.
r	int16	Here is R normalized to the range [-1000, 1000]. An int16_MAX value indicates an invalid value. Usually corresponds to the movement of the slider with a maximum value (1000) - and a minimum value (-1000). Positive values - positive thrust, negative values - negative thrust.
buttons1	uint16	A bit field corresponding to the current state of the joystick buttons from 0 to 15, 1 - pressed, 0 - released. The least significant bit corresponds to button 1.
buttons2	uint16	The bit field corresponding to the current state of joystick buttons 16-31, 1 - pressed, 0 - released. The least significant bit corresponds to button 16.

It defines the controlled target system, movement along the four main axes (x, y, z, r), as well as two 16-bit fields to represent the states of up to 32 buttons (buttons1, buttons2).

Parameters subprotocol

The subprotocol is used to exchange configuration settings between network components. Each parameter is represented as key \ value. A key is usually a parameter name (16 characters) and a value, which can be of one or more types.

2.3.8 Enumeration

2.3.8.1 ParamTypeEnum list

Sets the data type of the parameter (Table 2.15)

Table 2.15: Structure of the paramTypeEnum message

Value	Field Name	Description
1	paramUInt8	An 8-bit unsigned integer
2	paramInt8	8-bit integer with familiar
3	paramUInt16	16-bit unsigned integer
4	paramInt16	16-bit integer with a familiar
5	paramUInt32	32-bit unsigned integer
6	paramInt32	32-bit integer with a familiar
7	paramUInt64	64-bit unsigned integer
8	paramInt64	64-bit integer with a familiar
9	paramReal32	A 32-bit floating-point number
10	paramReal64	64-bit floating point number
11	paramCustom	User type

Messages

= paramValue messages

Printing the value of a built-in parameter. Including paramCount and paramIndex in the message allows the receiver to keep track of received parameters and re-request missing parameters after a loss or timeout. The structure is shown in Table 2.16

The paramRequestList message

Sets all parameters of the selected component or system as a whole. Once requested, all parameters must be sent by the system or component. The structure is shown in Table 2.17.

paramRequestRead messages

A request to read a built-in parameter with the string identifier paramId. Built-in parameters are stored as key [const char*] -> value [float]. This allows you to send a parameter to any other component (for example, GCS) without the need for prior knowledge of possible parameter names. Thus, the same GCScan store different parameters for different autopilots. The structure is shown in Table 2.18.

Table 2.16: Structure of the paramValue message

Parameter	Type	Value	Description
paramId	char[16]		Built-in parameter identifier, disabled by NULL if less than 16 human-readable characters, and NO NULL byte if length is exactly 16 characters - applications must provide 16+1 bytes of storage if identifier is stored as a string
paramValue	float		The value of the on-board parameter
paramType	uint8	paramTypeEnum	Onboard parameter type.
paramCount	uint16		Total number of on-board parameters
paramIndex	uint16		The index of this on-board parameter

Table 2.17 : The structure of the paramRequestList message

Field Name	Type	Description
targetSystem	uint8	System ID
targetComponent	uint8	Component ID

Table 2.18: The structure of the paramRequestRead message

Field Name	Type	Description
targetSystem	uint8	System ID
targetComponent	uint8	Component ID
paramId	char[16]	The built-in parameter identifier is NULL if less than 16 human-readable characters, and NO NULL bytes if the length is exactly 16 characters - applications must provide 16+1 bytes of storage if the identifier is stored as a string
paramIndex	int16	Parameter index. Send -1 to use the paramId field as an identifier (otherwise paramId will be ignored)

paramSet messages

Setting a parameter value (writing a new value to permanent storage). The receiving component must confirm the new parameter value by broadcasting paramValue messages (wide language broadcast ensures that all multiple GCS have an up-to-date list of all parameters). If the sending GCS has not received the paramValue within its timeout, it shall forward the paramSet message

Table 2.19: The structure of the paramSet message

Field Name	Type	Value	Description
targetSystem	uint8		System ID

targetComponent	uint8		Component ID
paramId	char[16]		The built-in parameter identifier is NULL if less than 16 human-readable characters, and NO NULL byte if the length is exactly 16 characters - applications must provide 16 + 1 bytes of storage if the identifier is stored as a string
paramValue	int16		The value of the on-board parameter
paramType	int16	paramTypeEnum	Onboard parameter type.

2.3.10 Message encoding

Parameter names/identifiers are specified in the paramId field of the messages in which they are produced. The paramId string can be up to 16 characters long. The string is terminated with a NULL character if there are less than 16 human-readable characters, and with no null termination byte if it is exactly 16 characters long. The values are encoded byte by byte in the paramValue field, a 4 byte IEE754 single precision floating point value.

The paramType parameter (paramTypeEnum) is used to specify the actual data type so that the receiver can decode it. Supported types are 8, 16, 32, and 64-bit signed and unsigned targets, and 32- and 64-bit floating point numbers. The exchange of large integers requires a byte-by-byte conversion rather than a simple cast (for example, 1E7 scalable integers can be useful for encoding some data types, but lose precision when converted directly to floating point numbers).

Message caching

The GCS or other component may maintain a cache of parameter values for connected components/systems to reduce the time it takes to display values and reduce network traffic. First, the cache can, perhaps, be filled by reading the full list of parameters at least one

Message operations

Reading all parameters

The read operation starts by sending the paramRequestList message. The target component must begin broadcasting parameters individually in paramValue messages upon receipt of this message. The drone must be able to pause after sending each parameter to ensure that the operation does not consume all of the available bandwidth on the link (approximately positively 30 to 50 percent of the bandwidth).

The sequence of transactions is shown in Fig. 2.5 the following:

- GCS (client) sends a paramRequestList specifying the target system/component.

- broadcast addresses can be used. all target components must match the parameters (or ignore the request if they don't).

– The GCS is expected to collect parameters from all responsible systems. The timeout/retry behavior depends on the GCS.

The target component(s) must respond by sending all parameters individually in paramValue messages.

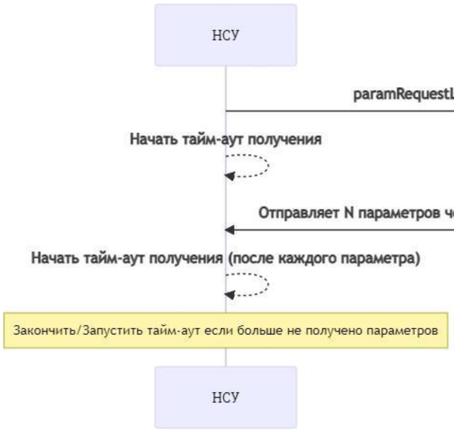

НСУ
МБА
paramRequestList
Начать тайм-аут получения
Отправляет N параметров через paramValue
Начать тайм-аут получения (после каждого параметра)
Закончить/Запустить тайм-аут если больше не получено параметров
НСУ
МБА

Fig. 2.6: Time diagram of reading all parameters

Read one parameter

One parameter can be read by sending a paramRequestRead message, as shown in Figure 1. 7.

The sequence of operations is as follows:

1.GCS (Client) sends paramRequestRead , specifying either the identifier (name) of the parameter, or the index of the parameter.

2.GCS triggers a confirmation wait timeout (in the form of a paramValue message).

3.SUAV matches paramValue containing the value of the parameter. This is a broadcast message (sent to all systems).

SUAV may restart the sequence if the paramValue confirmation is not received within the timeout.

The SUAV has no formal way to signal when an invalid parameter is requested (that is, for a parameter name or identifier that does not exist). In this case, the drone should return a statusText. GCS can monitor a particular message, but otherwise fails the request after any timeout/resubmission cycle completes

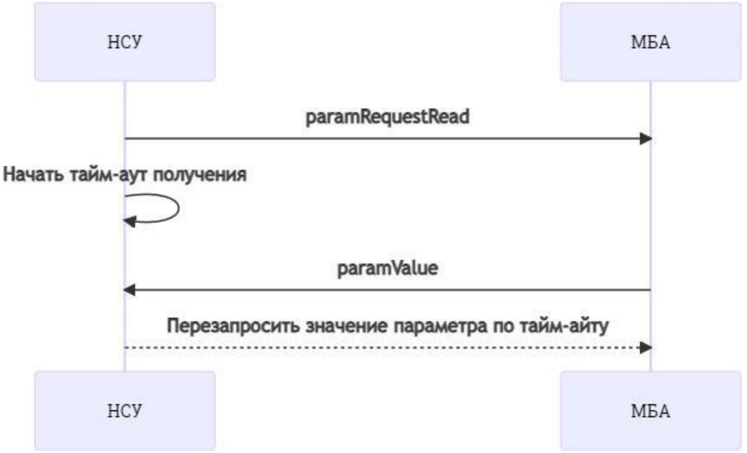

НСУ
МБА
paramRequestRead
Начать тайм-аут получения
paramValue
Перезапросить значение параметра по тайм-айту
НСУ
МБА

Fig. 2.7: Time diagram of reading one parameter

Write one parameter

Parameters can be written individually by sending the parameter name and value pair to the GCS as shown below (Figure 2.7):

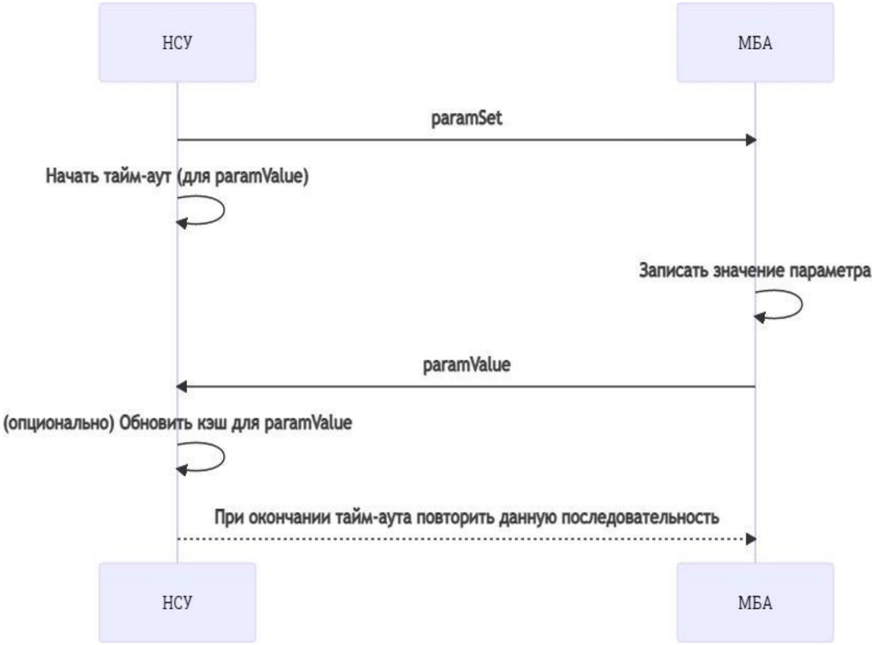

НСУ
МБА
paramSet
Начать тайм-аут (для paramValue)
Записать значение параметра
paramValue
(опционально) Обновить кэш для paramValue
При окончании тайм-аута повторить данную последовательность
НСУ
МБА

Fig. 2.8: Timing diagram of one parameter recording

The sequence of operations is as follows:

1.GCS (Client) sends a paramSet , which specifies the name of the parameter to update and its new value (also target system/component and parameter type).

2.GCS triggers a confirmation wait timeout (in the form of a paramValue message).

3.SUAV writes the parameter and responds by passing paramValue containing the updated parameter value to all components/systems.

4.GCS should update the parameter cache (if used) with the new value.

5.GCS may restart the sequence if the expected paramValue values are not received within the timeout or if the write operation fails (the value returned in paramValue does not match the set value).

SUAV must acknowledge the paramSet by broadcasting the paramValue even if the write operation fails. In this case, paramValue will be the current/unchanged value of the parameter.

2.4 Subsystem structure

In simple form, the UAV mission consists of a set of at least two waypoints in three-dimensional space, where 1 point is the beginning of the mission, and the last end of the mission, all other points are intermediate stages to the completion of the last mission. Based on this, in a generalized form, to establish a mission, you just need to make a chain of missions (waypoints).

The whole subsystem consists of

1. MavLink Protocol (Improved by myself).
2. Hybrid system of nested structures (software implementation)
3. Identifier validation system (software implementation).
4. Program interface (manually developed).
5. Interaction interface (handcrafted).
6. Additional subsystem algorithms (software implementation).
7. Visualization of UAV characteristics (Graphical implementation).

8. Topography (Map in different display modes, satellite, road and so on)

9. Channel protection algorithm (Theoretical implementation).

10. Data Exchange Algorithm (Borrowed by Razernet, redesigned for the task at hand)

11. Data logging subsystem (tracking and diagnostics of UAV characteristics).

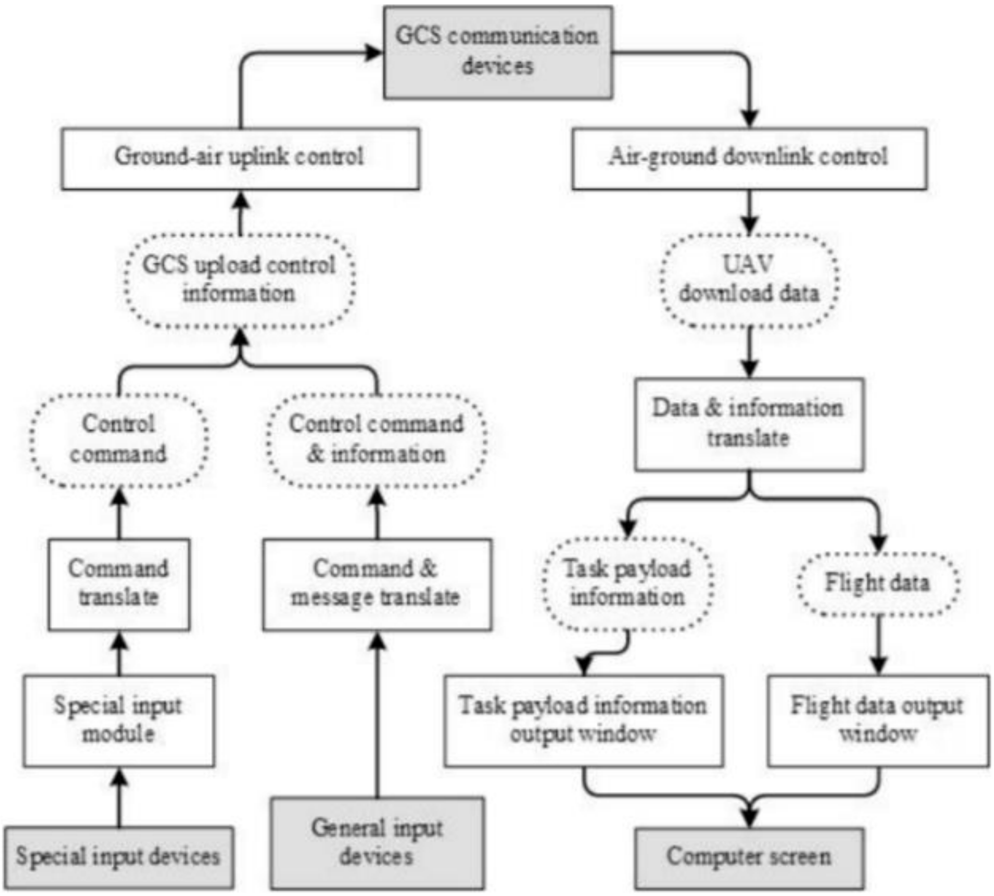

GCS communication devices
Ground-air uplink control
Air-ground downlink control
GCS upload control information
UAV download data
Control command
Control command & information
Data & information translate
Command translate
Command & message translate
Task payload information
Flight data
Special input module
Task payload information output window
Flight data output window
Special input devices
General input devices
Computer screen

Fig. 2.9: Architecture of the generalized subsystem.

The subsystem is divided into various modules, each of which is responsible for a separate functionality as a whole. It can be divided into three main categories:

1. This is a set of modules responsible for user interaction with the UAV and GCS.
2. A set of modules that implements the functionality of the exchange and management of UAV systems (Server and protocol).
3. Modules responsible for additional checks (conversion algorithms, dynamic data creation, logging).

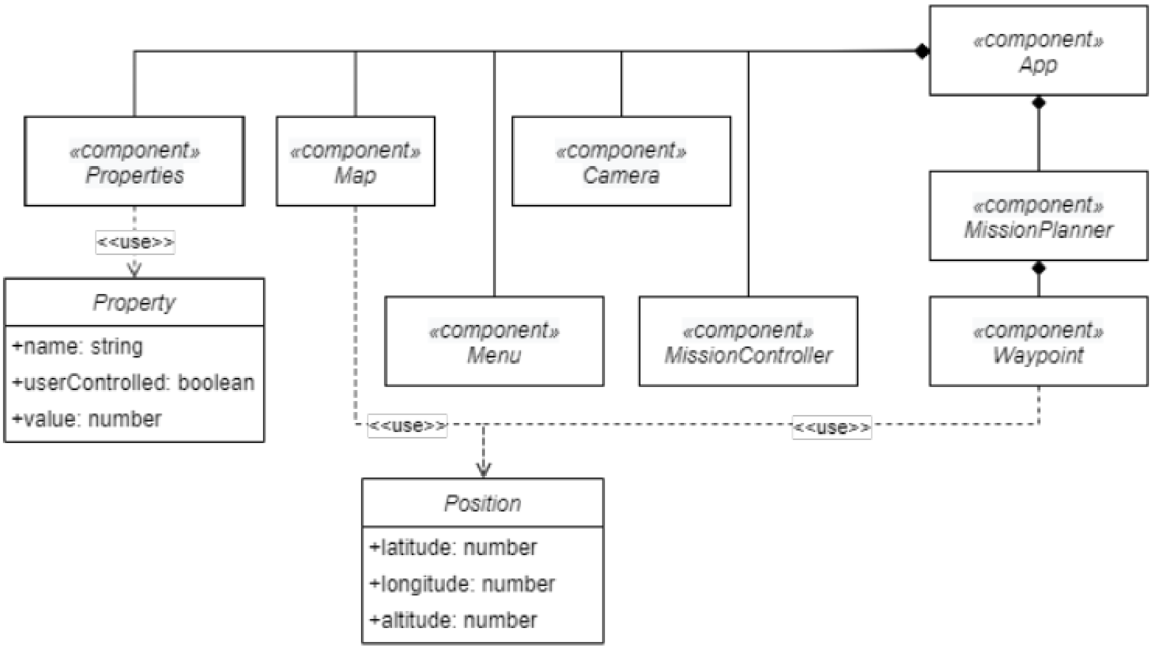


Fig. 2.10: Program architecture.

Most of the algorithms will be discussed in chapter 3.4. But until that time, it is necessary to consider the module (server), the generalized work and structure of this system.

The main algorithm of work is laid down on the Client - server architecture, the client-server architecture model is determined by the distribution of responsibilities between the client (UAV) and the server.

For our program to work, you need to run a virtual server that goes in the files to the diploma. Which is a wired web server in such a way that when accessed by a local address, this request is processed by resources related to the directory with the assembled server.

To decrypt data using the MavLink protocol, a data protocol controller “InformProtocol.h” is written on the server, which is responsible for decrypting the protocol and the hybrid system of structures.

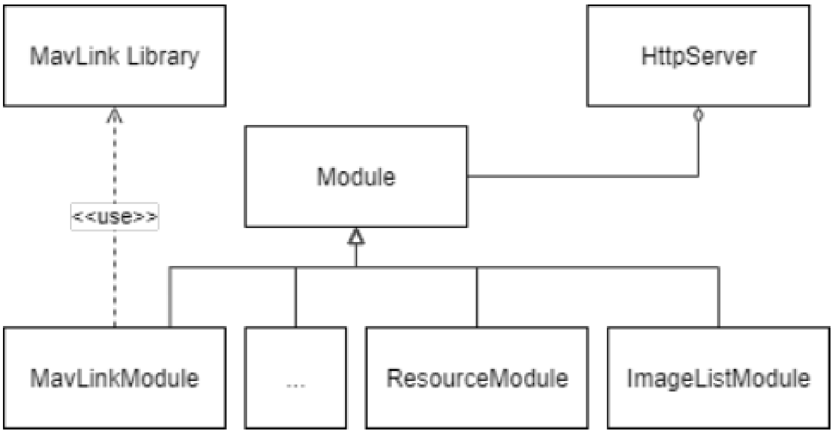


Fig. 2.11: Architecture of the server part.

2.5 Statement of the problem

In this work, we analyzed:

1. Structural features of the UAV.
2. Features of the Ground Control Station and its purpose.
3. The main subsystems for monitoring and controlling the UAV.
4. Features of UAV control and defense lines.
5. Existing problems of control and protection of the UAV control line
6. Existing solutions to ensure the protection of the UAV control line
7. Available solutions for UAV-Operator (GCS) interaction interface

It was found that the existing solutions in the protection of the line of defense and the hierarchy itself the structure of the UAV-GCS complex needs improvement. The following problems were identified:

1. Standard data exchange protocols have significant insecurity due to the openness of the data package and small structure.
2. The solutions that provide the protection line of the control channel are largely outdated, and knowing the algorithm of the exchange line, you can easily sew your data packets into the structure of the UAV.
3. The system of internal software monitoring does not allow full data analysis, so it will be difficult to diagnose the system on the fly.
4. Most of the interfaces are difficult to learn and the more functionality the system has, the more time it takes for the operator to master the system.

Therefore, the following tasks were set in the further work:

1. Modernize the level of protection of the data exchange protocol, and write a software implementation of the protocol that will make the system more secure (partially described in section 2.3, continued in section 4)

2. To analyze the features of signal formation in remote control systems of drones

3. Write a new algorithm for the line of protection that will allow the control line to receive more protection.

4. Develop a system by which it is possible (in the presence of an operator and a specialist) to quickly diagnose the state of the UAV and its systems (Software failures).

5. To develop a light and maximally optimized UAV control interface, which will not have a heavy appearance and will be clear to use.

At the time of writing, the UAV flight control application has not yet been finalized, which does not allow testing the application on a real system. A personal computer with the Ubuntu 20.04 operating system, 16 GB of RAM and an AMD

Ryzen 7 5800H with Radeon Vega Mobile Gfx 2.30 GHz processor was used as a platform. Another personal computer located in the same WiFi network acted as a client.

Performance measurements are presented in Table 3. Measurements were made using the "htop" process monitor. The CPU percentages indicated in the table are considered from one core. Since the main task of the processor is flight control, slightly more than 5%-10% of the on-board computer's power should be allocated to communication. Many target platforms—such as the Jetson Nano or the Raspberry Pi 4—have a quad-core processor at 1.5 GHz, so to approximate the satisfaction of this criterion, the CPU percentage was multiplied by 2.2 (the sample difference in core power23), divided by 4 (the number of cores), and then divided by 0.1 to obtain a percentage of the maximum load of 10%.

Due to the lack of data from the device, the application constantly generates various simulations of data, be it readings from meters or images. To consider the impact of these generators, they were additionally run without a web server, CPU readings were measured, and then the CPU percentage readings of the web application being used were adjusted accordingly.

CHARTER 3. PROBLEMS OF PROTECTION OF THE UAV CONTROL LINE AND METHODS OF NOTIFICATION OF THE LEVEL OF PROTECTION

3.1 Analysis of existing solutions

Most UAVs use radios to remotely control and share video and other data. Early UAVs had only a narrowband communication channel. Downlinks came later. These two-way narrowband radios transmit command and control (C&C) data and telemetry data about the status of the aircraft's systems to the remote operator. For very long-distance flights, military UAVs use satellite receivers as part of satellite navigation systems. In cases where the transmission of a video signal is required, UAVs implement a separate analog radio communication for video communication.

Most modern UAV applications require video transmission. Thus, instead of two separate channels for C&C, telemetry and video traffic, a wideband channel is used to transmit all types of data over a single radio channel. These broadband channels can use quality of service techniques to optimize C&C traffic to reduce latency. Typically, these broadband channels carry TCP/IP traffic that can be routed over the Internet.

The radio signal from the operator can come from:

ACIC DEPARTMENT				*NAU 22 11 86 000 EN*		
Performed	*A.O. Tsoba*			*Control and Monitoring System of an Unmanned Aerial Vehicle. Monitoring Subsystem* (complex).	*N.*	*Page*	*Pages*
Supervisor	*V.M.Sineglasov*						
					225 151		
S. controller	*M.K. Filyashkin*						
Dep. head	*V.M.Sineglazov*						

1. Ground Control - The person operating the radio transmitter/receiver, smartphone, tablet, computer, or military ground control station (GCS). Control from wearable devices, recognition of human movements, and human brain waves were also recently demonstrated.

2. A remote network system, such as satellite duplex data links for some military states. Downlink digital video over mobile networks has also reached consumer markets, while direct uplink control from UAVs over cellular and LTE has been demonstrated and tested.

3. Another aircraft that acts as a relay or mobile control station is the military manned unmanned aerial system (RPA).

4. The MAVLink protocol is becoming increasingly popular for transmitting commands and control data between the ground controller and the vehicle.

To date, it has become possible to control the aircraft using the autopilot in the complete absence of communication between the aircraft and the GCS. In this case, the flight task is performed offline. However, this does not allow us to claim that command and telemetry radio communication can be excluded from the composition of the UAV. Due to the increased complexity and cost of the complex during its operation, constant monitoring of the state of the aircraft in the air is necessary. In addition, sometimes there is a need to adjust the flight parameters of the UAV.

An urgent task is also the transfer of data on the payload of aircraft to the GCS. In this case, it is necessary to ensure the transfer of a large amount of data with specified requirements for bandwidth, bit error probability, etc. When creating small and ultra-small UAVs, requirements are put forward to minimize the size of the receiving-transmitting and antenna-feeder equipment. Increased requirements for

fault tolerance are placed on the UAV equipment for navigation and navigation, which provides manual landing modes (if necessary), on the servo drive and the automatic rescue system (ARS). The listed equipment belongs to the first classification group and ensures the reliability of the UAV complex as a whole. The failure of any element of the equipment of the first group leads to the immediate termination of the flight task and the return of the aircraft to the base. If this is not possible, the ARS is activated and the parachute ejects.

The rest of the aircraft belongs to the second classification group. When the equipment of this group fails, the decision on further actions is taken by the management staff of the complex. The interaction of the equipment of the first and second groups is carried out through control interfaces. During the operation of the communication system, the probability of bit errors of each communication channel is evaluated and a decision is made about the distribution of the flow of command and telemetry data between the channels. The use of several communication channels increases the reliability of the data transmission system and, at the same time, is redundant from the point of view of efficient use of the radio frequency spectrum. One of the ways to increase the efficiency of the communication system is the adaptive operation of the system, which involves the transmission of a part of useful data via command-telemetry communication channels, the volume of which changes depending on the current conditions of the radio signal. method of transmission.

As a rule, the maximum distance for direct radio communication between a civilian UAV and GCS today does not exceed 100 km. For command and telemetry communication over long distances, it is possible to use satellite communication. In this case, the data flow is limited to the minimum necessary information about the state of the UAV, the transmission interval of which can be from 30 to 300 seconds. A promising direction for the development of communication systems with UAVs is the use of frequency ranges above 5 GHz. In this case, it becomes possible to transmit a large amount of payload data in real time (for example, it can be images of radiation sensors of different wavelengths). The factors that sharply limit the range of the radio communication system when using these bands are the strong dependence of the propagation conditions of electromagnetic waves on weather conditions, the need for direct visibility, and the influence of multiradiation.

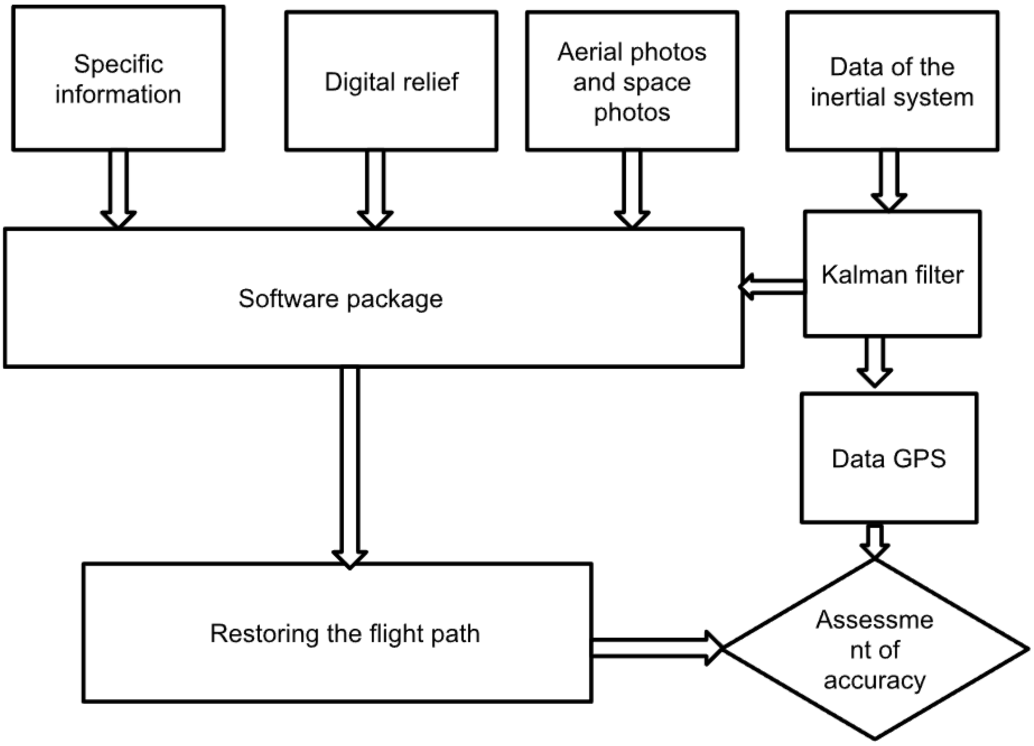


Fig. 3.1 Generalized scheme of UAV navigation

3.2 Improvement of the UAV control line protection method.

The tasks of identifying the type and class of radio signals necessary for the identification of the type of objects have been solved for many years. At the same time, the main attention was focused on certain conditions with previously known parameters of the signal and its internal structure.

Most of the known algorithms for sequential or simultaneous verification of several conditions do not take into account the structural component of the signal. In addition, a general drawback characteristic of most known algorithms is the fragmentation of the results, which complicates practical (program) implementation and reduces their effectiveness.

Identification of signals of UAV control systems is called the establishment of the fact that the received radio radiation belongs to the class of signals of UAV control systems.

The range of radio remote control systems available on the world market for engineering models (cars, ships, airplanes, etc.) is extremely large, and detailed information about the types of signals used, modes and protocols is a trade secret of production. enterprises and is not available for free access [42]. Based on the results of the analysis, it was established that UAV control systems are conditionally divided into two groups:

- systems operating at fixed frequencies in the meter wave range use signals with frequency or amplitude manipulation and relatively simple protocols for packet transmission of control commands;
- systems using adaptive operating frequency adjustment modes in the 2400...2483.5 MHz range, Gaussian frequency signals, phase or quadrature amplitude manipulation, as well as various spectrum expansion algorithms.

Based on the results of the market analysis of remote control radio systems, it was established that the quantitative ratio between the available samples belonging to the groups defined above is approximately the same. At the same time, the market value of the systems of the second group is about an order of magnitude higher than that of the first. Despite the fact that the model range of samples of both groups has approximately the same quantitative ratio, the popularity of their samples is related approximately as 80% to 20%. Therefore, the first group of UAV control systems was selected for further research.

3.2.1 Analysis of features of signal formation in remote control systems of drones/

The most widely used protocols for transmitting control commands from the operator to the drone include pulse-position and pulse-code modulation. It is important to note that the signal is generated during transmission. Signal conditioning on the transmit side using PPM modulation and includes the following steps:

- adding a synchronization interval;

- creating a sequence of single pulses (taking into account the number of channels). The position of each must clearly correspond to the value of the control variable;
- manipulation of the carrier (subcarrier) frequency of the transmitter by the received pulse packet.

The duration of a single pulse is equal to $\tau_S = 0.3 - 0.5$ ms, and synchronization $\tau_c -$ до 22 ms. At the same time, the change in frequency indicators F_M is from 3 to 10 kHz. Taking into account that the average duration of the pulse packet is approximately 0.08–0.2, asymmetric frequency manipulation occurs, which corresponds to the same asymmetric shape of the amplitude-frequency spectrum, as shown in Fig. 4.1 for the case of subsonic modulation. In this case, instead of the central frequency of the spectrum, there is the so-called dominant frequency f_D, which for the considered signal in Fig. 4.1 is 1100 Hz.

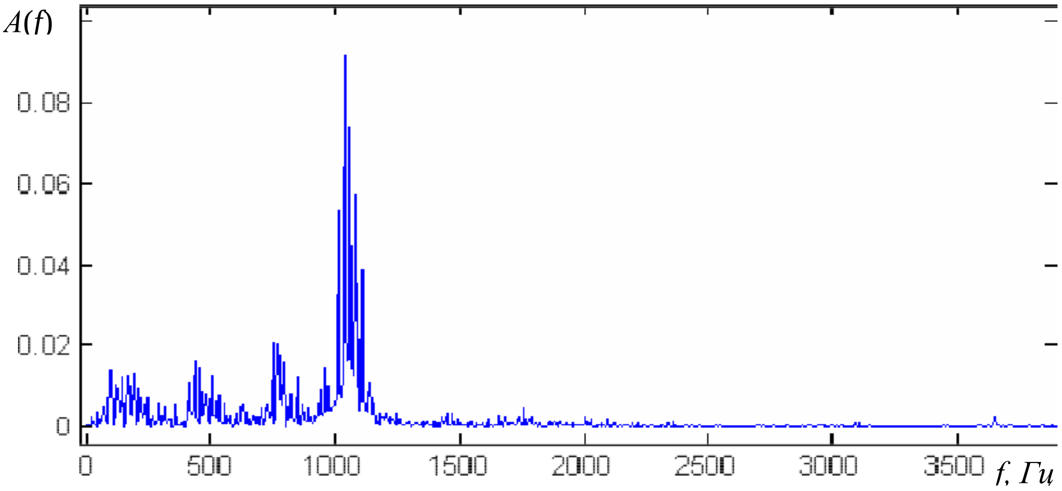


Fig.4.1. Typical amplitude-frequency spectrum of a signal with pulse-position modulation.

Signals with pulse code modulation (PCM) are formed by two-position frequency shift keying of bit sequences, which include eight-bit code words (bytes) in the following typical sequence: preamble (6 - 12 bytes), information channels (4 - 8 channels of 2 bytes each) , checksum (2 or 4 bytes). Unlike PPM, the usual symmetrical frequency shift keying occurs. A typical amplitude-frequency spectrum of such a signal is shown in Figure 4.2.

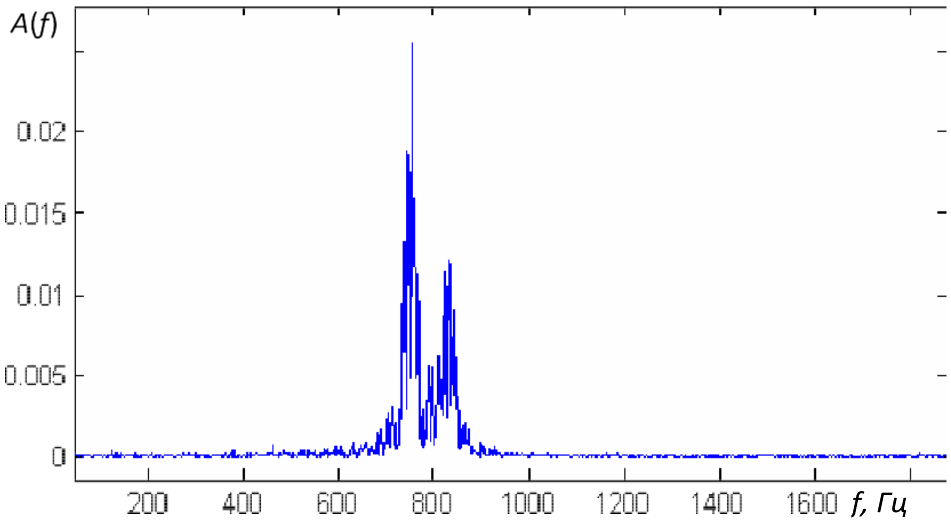


Fig. 4.2. A typical amplitude-frequency spectrum of a signal with pulse-code modulation

The conditions for detecting signals from UAV control systems are [43]:

- energy (electromagnetic) availability of the radio emission source; finding the amplitude-frequency spectrum of the signal within the bandwidth
- radio receiving device (RRD) during the time allocated for analysis; being in the off state of automatic frequency control systems and automatic gain control RRD;
- operation of RRD in the mode of signal transfer to a low-frequency subcarrier (USB or LSB), or conversion to quadrature components.

3.1 Identification of a system of signs and criteria for signal recognition/

According to the classical theory of signal detection with partially known modulation parameters, the first sign S_1 for our case, there is an energy one, according to which the signal energy:

$$E(t) = \frac{1}{t}\int_0^t U^2(\tau)d\tau. \tag{4.1}$$

Within a certain time interval t, limited start time t_s and completion t_f аналізу, analysis, must exceed the threshold value E_0:

$$S_1 = \begin{Bmatrix} E(t) > E_0 \\ t \in (t_s, t_j) \end{Bmatrix}, \qquad (4.2)$$

where $U^2(r)$ —actual analyzed signal. Threshold value E_0 it is established based on the results of the analysis of the background electromagnetic situation during the operation of non-dangerous radio-electronic devices.

In general, the detection should be carried out in a certain frequency range, therefore E_0 should be a function of frequency, the discrete values of which are taken with a step equal to the typical width of the signal spectrum.

The next feature is the modulation one, indicating that the signal to be analyzed belongs to the class of signals with frequency manipulation. The very sign S_2 is described by a set of partial signs in the expression:

$$S_2 = \{S_{2.1} \cap S_{2.2} \cap S_{2.3}\}, \qquad (4.3)$$

where the partial sign is calculated:

$$S_{2.1} = \left\{\max_i(F_i) << 2\pi \frac{1}{N_s} \sum_{i=0}^{N_s-1} |F_i|\right\}. \qquad (4.4)$$

where F_i, $i = 0, \dots N_s - 1$ — discrete Fourier transform with dimension N_s from the input signal, is responsible for its belonging to the class of signals with a constant amplitude;

$$S_{2.2} = \left\{\frac{1}{N_s-1} \sum_{i=0}^{N_s-1} (\theta_i)^2 > \frac{\pi^2}{16}\right\}. \qquad (4.5)$$

Expression in which $\theta_i = arctg(Q_i/I_i)$– readings of the absolute phase of the signal; I_i and

Q_i, $i = 0, \dots N_s - 1$ — the quadrature components of the signal, obtained by means of the Gilbert transformation, indicate that the signal belongs to the class of signals with angular modulation.

$$S_{3.2} = \left\{\sum_{i=1}^{N_S/2-l}(f_i - f_D)^2 \neq \sum_{i=N_S/2}^{N_S-1}(f_i - f_D)^2\right\}. \qquad (4.6)$$

The expression indicates that the modulating function (instantaneous signal frequency) $f_i = (\theta_i + \theta_{i-1}) * F_S$,де F_S – sampling frequency, has a pulse character. Dominant frequency f_D in the signal spectrum is calculated iteratively according to the expression.

$$\text{fD} = \text{k}^1{}_{\text{Ns}}\,\text{f}_\text{i} + (1\text{-k})\text{f'} \qquad (4.7)$$

where f_D – the value calculated by the previous iteration; k – dimensionless weighting factor chosen in the range of 0.05 - 0.1.

The final feature is the presence of a signal in the modulating function $F(t)$, subject to analysis, information about management teams (structural feature). That is, the modulating function itself must have a structure and appearance similar to the one formed on the transmitting side. Analytical sign S_3 is recorded in the following form:

$$S_3 = \left\{\left[\frac{|\hat{\tau}_c - \tau_c|}{\tau_c} < D_1\right] \cap \left[\frac{|\widehat{F_M} - F_M|}{F_M} < D_2\right] \cap \left[\frac{1}{T}\int_0^T F(t)dt < D_3\right]\right\}. \qquad (4.8)$$

where $\hat{r}_c$ and $\hat{F}_M$ –measured values of clock pulse duration and frequency deviation, respectively, dimensionless threshold values D_1, D_2та D_3 are set experimentally depending on the required values of the probabilities of correct detection and false alarm.

3.2.1 Improvement of the signal detection algorithm of UAV control systems.

It is based on successive checks of signs (4.2), (4.3), (4.8), which are preceded by the necessary calculations. The input data for the algorithm is an array of counts $r(k)$, $k = 0..K - 1$, obtained after discretization in time with frequency F_S, and quantization by signal mixture level $r(t)$ from the output of the radio receiving device:

$$r(t,U) = s(t,U) + n(t), \tag{4.9}$$

$s(t,U)$ −useful component; $n(t)$– noise; $U = [a, f, \theta, T]$ – vector of signal parameters; a –signal amplitude; f –central frequency of the spectrum; θ – time-invariant carrier frequency phase; T – symbolic period.

Sample rate valueF_S and the number of quantization levels must satisfy the requirements for the subsequent correct reproduction of the signal. The result of the algorithm should be the solution S about availability ($S = True$) or absence ($S = False$) signal of UAV control systems in the received signal mixture. The scheme of the developed algorithm is shown in Fig. 4.3.

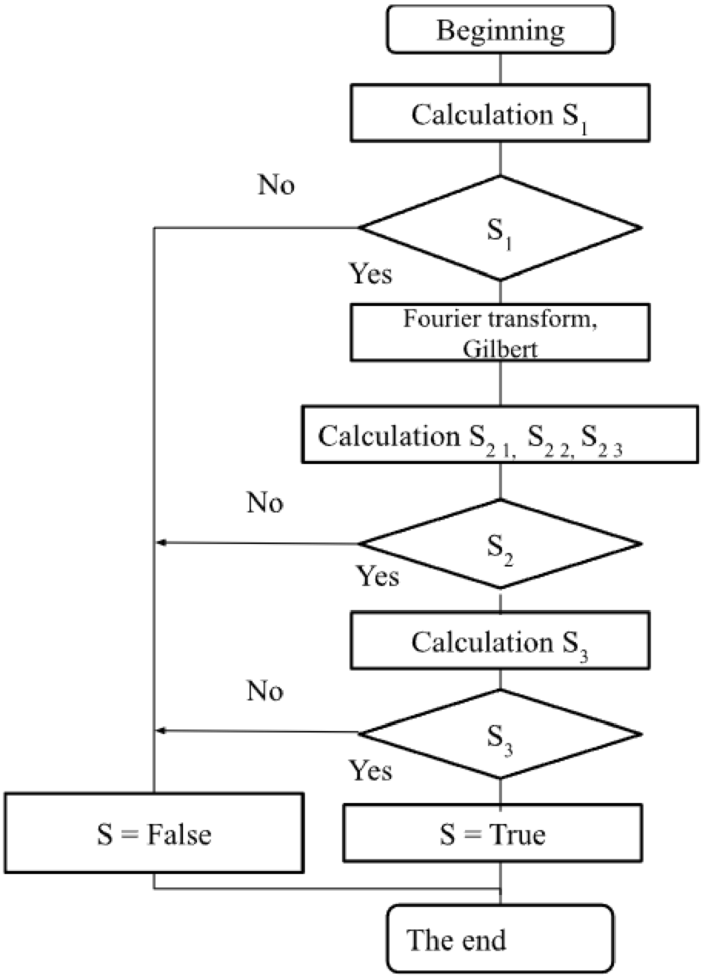


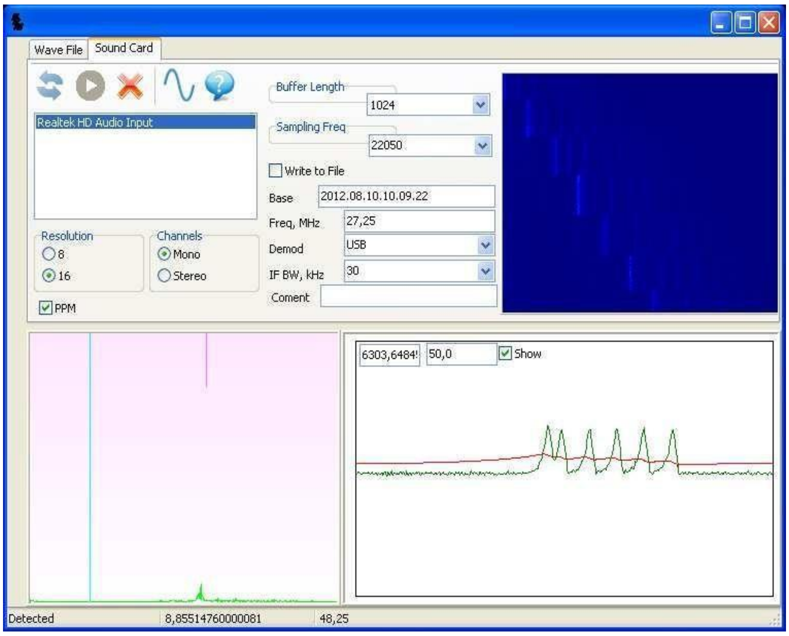
Fig. 4.3. Scheme of the signal detection algorithm of UAV control systems

The first stage of the algorithm is to check the energy feature S_1. Checking other signs is carried out only when S_1 = *True*, that is, the presence of any signal whose energy is sufficient for further analysis. Otherwise, a decision is made about the absence of any signal (including the signal of the UAV control systems) and the algorithm ends its work. Additionally, on this basis, the radio receiver can be rebuilt to analyze another part of the frequency range.

The second stage of the algorithm is to check the modulation feature S_2, which is preceded by intermediate calculations: Fourier and Gilbert transform, selection of the modulating function (vector of the instantaneous frequency of the signal), filtering and automatic tuning to the dominant frequency of the signal. When S_2 = *True* there is a case of a signal that, by the type of modulation, corresponds to the signal of the UAV control systems.

The third stage of the algorithm is to check the structural feature S_3. When it is executed, the final decision is made on the availability (S = *True*) or absence (S = *False*) signal of UAV control systems in the received signal mixture.

The practical implementation of the developed algorithm for detecting signals from UAV control systems is carried out in a software tool, the main window of which is shown in Fig. 4.4.



Realtek HD Audio Input

Buffer Length: 1024

Sampling Freq: 22050

□ Write to File

Base: 2012.08.10.10.09.22

Freq, MHz: 27,25

Demod: USB

IF BW, kHz: 30

Coment:

Resolution: ○ 8 ● 16

Channels: ● Mono ○ Stereo

☑ PPM

6303,6484! 50,0 ☑ Show

Detected 8,85514760000081 48,25

Fig. 4.4. Software implementation of the developed algorithm.

The software is designed for use in Windows 10 and above. It provides processing of sound files with a signal recording in WAW format or directly a stream of samples from an audio adapter of an electronic computer in real time. Signal analysis with automatic tuning to the center (dominant) frequency is performed in the bandwidth of the audio adapter, which can be up to 96 kHz. During operation, the main stages of processing are displayed: an oscillogram, an instantaneous spectrum and a sonogram of a signal, a graph of the modulating function. To detect the signal of the UAV control systems, an average of 0.5 - 1.2 s is required. In case of such detection, sound and visual indication is provided [44].

3.2.2 Proposals for the development of a UAV prototype with a secure communication channel.

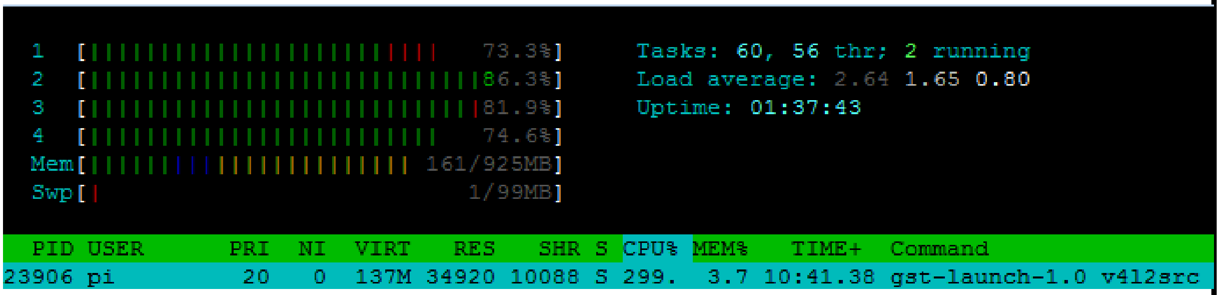
Main microcomputer. A microcomputer (Raspberry Pi 3 Model B) was proposed for further development and research. As the results of world researchers show, RPi 3 is able to fulfill all the functions and requirements that are formed for it. That is, RPi 3 can act as a powerful enough platform to implement the capabilities of a UAV prototype with a secure communication channel.

The main criteria for choosing a microcomputer were the presence of a large number of ports, sufficient power to perform stream encryption and other less expensive processes at the same time, and the built-in WiFi module and the presence of a GPU were considered useful features. RPi 3 fully meets the selection criteria: the number of ports is enough to connect all the necessary hardware components (DVB-T modem, video capture device, telemetry port and control port for the pilot controller, RTC, GPIO indicator and other situational ones), the processor copes with the load from all processes in full operation mode, there is a built-in WiFi module and VideoCore IV 3D.

Video systems. FPV Sony Super HAD color CCD with a maximum resolution of 752 x 582 (PAL) can be offered for video shooting. It is able to fully meet the resolution requirements of video traffic that needs to be captured and analyzed. Video traffic will be generated by the GStreamer framework, which consists of plugins.

A sufficiently large number of plugins provides a variety of possible options for building traffic and allows you to adjust the final result using various additional customization options. In addition, the source code of plugins is open, so you can modify plugins for specific solutions if the basic version does not have the necessary functionality.

For example, you can compare the processor load when using a standard plug-in for encoding video in H.264 format (Fig. 4.5) and the OMX plug-in (Fig. 4.6).



```
  1  [|||||||||||||||||||||||||     73.3%]     Tasks: 60, 56 thr; 2 running
  2  [||||||||||||||||||||||||||||86.3%]     Load average: 2.64 1.65 0.80
  3  [||||||||||||||||||||||||||||81.9%]     Uptime: 01:37:43
  4  [|||||||||||||||||||||||||     74.6%]
  Mem[|||||||||||||||||||||||  161/925MB]
  Swp[|                           1/99MB]

  PID USER       PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
23906 pi          20   0  137M 34920 10088 S 299.  3.7 10:41.38 gst-launch-1.0 v4l2src
```

Fig. 4.6. CPU load when using the OMX plugin

At the same time, with the second option, a significant part of the main processor is freed up due to the use of VideoCore IV 3D. Therefore, the average processor load when using x264 is about 300%, and when using the OMX plugin - 80%.

2. Communication line. For data transmission between the ground and air modules, the European standard for digital terrestrial broadcasting - DVB-T was chosen. DVB-T is designed to transmit a single MPEG-TS transport stream with digital services (multiplex), using COFDM modulation, at a rate of up to 31 Mbps. A compact full-duplex DVB-T transmitter was chosen, the parameters of which satisfy the conditions for using the communication line (it is necessary to

be able to separate the frequencies of the receivers on the UAV ground and air modules, and also to have a significant number of frequencies to which the receivers can be tuned in order to implement the EW countermeasure algorithm).

Table 4.1

Characteristics of the DVB-T modem

Parameter	**Value**	
Transmission bandwidth	Transmitter	2/3/4/5/6/7/8 MHz
	Receiver	5/6/7/8 MHz
Frequency range	Transmitter	50~950 MHz та 1200~1350 MHz with a step 1KHz
	Receiver	50~950 MHz with a step 1KHz
RF output level	0 dBm (108 dBuV)	
Digital amplification	Range: +6/-25 dB, with a step of 1 dB	

The software part of the communication line is the main and most complex component of the project: it includes stream multiplexing/demultiplexing, packet filtering, interfaces for interaction with the radio transmitter/receiver driver, statistics collection and other components.

Using the MavLink library, messages and control signals are exchanged between the flight controller and the ground control station. All data, including control signals, telemetry, video stream and other information, are combined into a single MPEG-TS stream, and additional data is attached to the packets to control the packets, which provides greater information protection.

Local messages within each of the modules are transmitted using the LCM library, which provides very fast data exchange and, as a result, low delays.

3. Communication Line Protection Since the AES algorithm is currently the encryption standard, the project uses the already implemented AES-128 algorithm from one of the many encryption libraries to implement data encryption in the project. The implementation of algorithms for countering EW

and protection against packet replacement is one of the main ideas of the project, therefore it is strictly confidential and cannot be described in this work.

However, a significant part of the research was the analysis and development of a time synchronization algorithm in a distributed system, namely between the ground and air communication module with the UAV. Synchronous clocks need to be used in two main areas: synchronous logging and synchronous change of radio frequencies in the EW countermeasure algorithm.

For logging, you need to set the corresponding time to real time with an accuracy of one second. For the effective functioning of the frequency change algorithm, much greater accuracy is required - up to 20 ms or better. It should also be noted that the communication system is based on the use of radio modems and has a delay in receiving data up to 50 ms (the delay is uneven and may differ from the delay in the opposite direction).

Based on the analysis of synchronization algorithms, requirements for the algorithm and hardware limitations, we will build our own algorithm that best satisfies the conditions for its use.

First, we will synchronize the time on the ground module using the NTP protocol. The ground module, unlike the air one, has the ability to connect to the network in pre-flight mode. An alternative time synchronization source, in the absence of a network, is an RTC connected to the ground unit, but the RTC has a lower timing accuracy and must be checked and adjusted from time to time.

Thus, there is a reference time source on the ground module. Now you need to synchronize the air module with the ground one.

Christian's algorithm was chosen as the basis of the synchronization algorithm because it is easy to implement and has high efficiency in small networks and networks with low load. The disadvantages of the algorithm are leveled as follows:

1. Requires an external accurate time source as above, the ground unit will be used as the reference time source.

2. There is no built-in protection against turning the clock in the opposite direction - we will synchronize the clocks once before the start of the flight. At present, key systems that depend on time synchrony will not be launched and the clock reset will not provoke ambiguity in the operation of the algorithms.

3. It does not allow to correctly set the time if the request and response are transmitted along different routes (different times are required for the transmission of these messages) - we will set the initial time as half the time of the full circle of transmission averaged over several iterations. To clarify the time at different times when transmitting in different directions, we correct the time by sending the current reference time to the air module and checking the calculated delay time with a delay when transmitting the reference time.

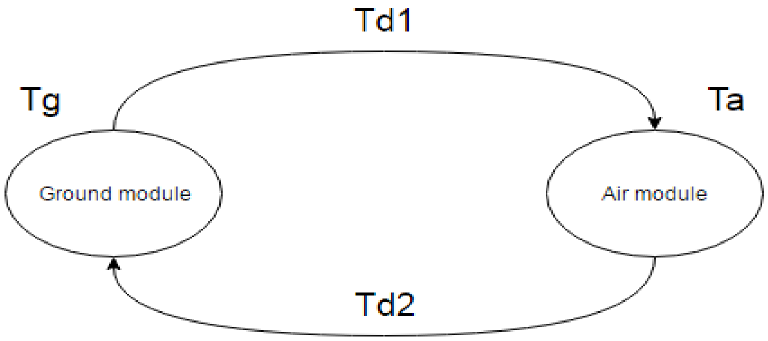


Fig. 4.7 Time shift

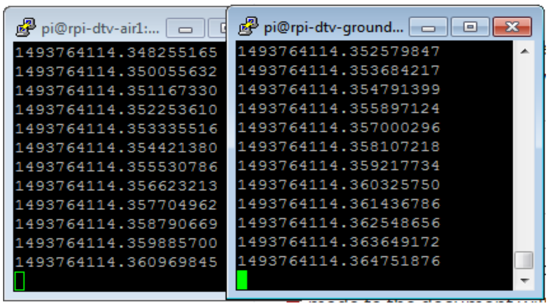
Delay in data transmission from ground to air module ($Td1$) and from air to ground ($Td2$) are unknown and may differ (Fig. 4.7). However, it is possible to determine the full circle delay ($Td1 + Td2$) and relying on Christian's algorithm to approximate $Td1 = Td2 = (Td1 + Td2)/2$.

For greater accuracy, several (currently 10) full circle delay calculations are performed. Of the 10 calculated delays, we choose the smallest one as the most accurate. After calculating the delay, change the time on the air module

$Ta = Tg + (Td1 + Td2)/2.$

The next step is to correct the delay chosen in the first step. Correction occurs by transferring timestamps from the ground module to the air module. On the air module, as in the first step, we find the slightest delay in the transmission of the timestamp. This delay is compared with the current time offset (in the first step $(Td1 + \ Td2)/2)$ at the ground station, if the delay has an error of synchronization accuracy falling into the rear delta, then the clock is considered synchronized, otherwise the time offset is set to be corrected by half the difference between the current delays on the air and ground module.

Thus, there is a clock synchronized with an accuracy of half the delay of a full circle, which theoretically is about 30 ms, but in fact, as shown by the practical results of use, due to the relative uniformity of the distribution of delay values in two directions, we have an error of about 5 ms, which is completely an acceptable result for the subsequent use of this algorithm.



```
1493764114.348255165
1493764114.350055632
1493764114.351167330
1493764114.352253610
1493764114.353335516
1493764114.354421380
1493764114.355530786
1493764114.356623213
1493764114.357704962
1493764114.358790669
1493764114.359885700
1493764114.360969845
```

```
1493764114.352579847
1493764114.353684217
1493764114.354791399
1493764114.355897124
1493764114.357000296
1493764114.358107218
1493764114.359217734
1493764114.360325750
1493764114.361436786
1493764114.362548656
1493764114.363649172
1493764114.364751876
```

Fig. 4.8. Time comparison after synchronization

3.3 Improving protection against cyber attacks

Working with any data is always associated with the potential for loss. Data can be lost as a result of various factors: human error (both users and network administrators), physical theft, destructive actions of malware, and data storage device failure. If personal data was lost (for example, an archive with photographs), then the damage is subjective and will be expressed in the negative emotions of the user. And, in case of loss of service information, the damage can manifest itself in the economic sphere - in financial damage, loss of competitive advantages, failures or failure to fulfill contracts, and even the ruin of the organization.

To protect against information loss, backup and data recovery systems (Backup & Recovery) are used. A data backup and recovery system is a software or hardware-software complex for creating copies of data at a certain frequency for their subsequent recovery. In addition to protecting against data loss, backup systems also allow you to ensure the continuity of work of employees by quickly updating the operating system (if you have an image) or restoring data on another computer.

How data backup and recovery systems work/

Creating a copy of the data is a fairly simple process, but the real needs of users are often very different and complex. For example, many users want to be able to backup from an arbitrary location or save very large amounts of data. For enterprises, the problem of managing a large amount of data, storing it and quickly recovering it is relevant. To solve each class of tasks, there are different systems for backing up and restoring data.

The main dividing lines between different data backup and recovery systems are in the areas of their use - for personal needs, in small companies and "home offices" (SMB / SOHO / ROBO) or in medium-sized (Enterprise) and large companies (Large Enterprise). Depending on this, the price of data backup and recovery systems, the types of storage used, the types of platforms, the functions provided, etc. differ. Let's look at some of these criteria.

One of the main differences for data backup and recovery systems is the type of storage media. Backups can be stored on tape, optical discs (CD, DVD, Blu-Ray, etc.), hard drives (HDD), solid state drives (SSD), network storage. Each of them has its own advantages and disadvantages. For example, storing data on tapes only at first glance seems to be an anachronism. Modern tape devices are quite cheap and guarantee long-term storage of data. But data recovery from such media can be very long. Therefore, they are more suitable for archiving data. "Hard" drives allow you to backup and restore fairly quickly, but they have a high price and not the longest life. An alternative to hard drives is the use of cloud storage, in which the type of storage systems is hidden from users. Of course, they use any disks as hardware, but the problem of saving disks falls on the service provider. Ensuring additional guarantees of preservation requires a lot of money for the maintenance of the "cloud" infrastructure (data duplication, "hot" replacement of disks, RAID arrays can be supported). However, the efficiency of using disk space can be higher, since the Cloud can be used by several clients and the efficiency of its use will be higher than that of a data backup and recovery system installed directly in the company. As a result, the effectiveness of a particular system is difficult to calculate a priori, so in each specific situation, the choice of a storage system should be preceded by an economic calculation.

Another difference is the type of platforms used. The data backup and recovery system can be implemented as software, hardware-software complex or as a service (software-as-a-service). The software costs less and requires separate storage systems. Therefore, such systems are suitable for personal use and small companies. For large companies, such systems can be used in conjunction with special data warehouses. For medium and large enterprises, backup and data update systems made in the form of software and hardware systems (PBBA, Purpose-Built Backup Appliance) are more suitable. These devices fall into two categories:

1. PBBA target systems (target systems). The data of the complex acts only as a target device for backup. Such a solution requires the use of additional software for automating, managing and consolidating backups, which in turn must be placed on additional server hardware with a deployed operating system to integrate all of the above components. These devices include EMC Data Domain, HP StoreOnce, etc.

2. PBBA integrated systems (integrated systems). This is a completely complete solution that does not require additional components for full-fledged work. These include servers, disk arrays, and backup software. Such systems have more integration between hardware and software and may include additional networking tools (such as load balancing). These solutions require no additional investment in infrastructure, lower deployment and integration costs, and are easier to maintain and administer. These devices include EMC Avamar, Symantec

Appliance BE+NBU, etc. Data backup and recovery systems also differ in the functions they provide. Conditionals can be divided into "basic" and "advanced" functions. The basic functions include scheduling, compressing and encrypting backups. Additional features are varied:

1. Duplication allows simultaneous copying to several sources, which increases the reliability of data storage.

2. Deduplication allows you to analyze and compress duplicated data. As a result, the load on data transmission channels and data storage is reduced.

3. Creation of system images. Periodic copying of not only data, but also system images allows you to quickly restore the employee's workplace even if the operating system or personal computer is damaged, which ensures the continuity of his work.

4. Load balancing. Allows you to optimize the load on multiple storages to quickly perform operations with backups.

5. Compatibility with software (operating systems and DBMS). Allows you to create snapshots of files and databases that can change during the backup process, for their correct holistic transfer and recovery.

6. Various tools for remote administration. This is a fairly diverse set of functions that allow you to automate the work of the administrator. These may include remote installation of agents on user computers, verification of created archives, manual or automatic merging of backups, etc.

7. Working with virtual devices.

8. Work with "cloud" storages.

9. Algorithms for updating data.

In case of data loss, various algorithms are used to increase the data update speed, allowing you to restore only the necessary data, eliminate duplication during recovery, etc.

Conclusions to the third chapter

An algorithm has been developed that ensures the establishment of the fact that the received radio emission belongs to the class of radio signals of UAV remote control systems with pulse-position and pulse-code modulation.

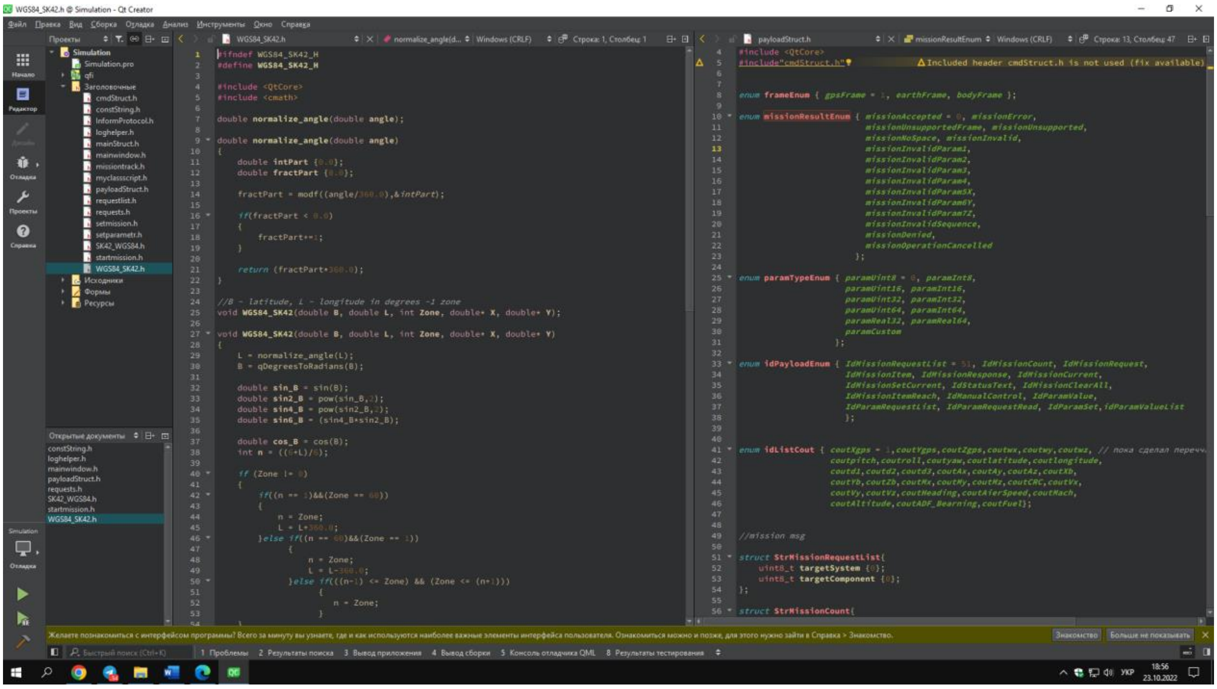
It is noted that the algorithm is based on successive checks of the energy, modulation and structural features of the signal, which provides for the possibility of automatic detection of its frequency tracking signal. Specific hardware and software solutions for building a UAV prototype are proposed.

The results of practical testing of possible solutions to problems on the finished prototype are given and the results of the functioning of the developed communication components are presented.

CHAPTER 4. SOFTWARE STRUCTURE

4.1 Project architecture

The user interface is implemented through the Qt toolkit , which is a cross-platform software development toolkit in the C++ programming language. Helps launch the program – on most modern operating systems, by simply compiling the program text for each operating system without changing the written code .



```
#ifndef WGS84_SK42_H
#define WGS84_SK42_H

#include <QtCore>
#include <cmath>

double normalize_angle(double angle);

double normalize_angle(double angle)
{
    double intPart {0.0};
    double fractPart {0.0};

    fractPart = modf((angle/360.0),&intPart);

    if(fractPart < 0.0)
    {
        fractPart+=1;
    }

    return (fractPart*360.0);
}

//B - latitude, L - longitude in degrees -1 zone
void WGS84_SK42(double B, double L, int Zone, double* X, double* Y);

void WGS84_SK42(double B, double L, int Zone, double* X, double* Y)
{
    L = normalize_angle(L);
    B = qDegreesToRadians(B);

    double sin_B = sin(B);
    double sin2_B = pow(sin_B,2);
    double sin4_B = pow(sin2_B,2);
    double sin6_B = (sin4_B*sin2_B);

    double cos_B = cos(B);
    int n = ((6+L)/6);

    if (Zone != 0)
    {
        if((n == 1)&&(Zone == 60))
        {
            n = Zone;
            L = L+360.0;
        }else if((n == 60)&&(Zone == 1))
            {
                n = Zone;
                L = L-360.0;
            }else if(((n-1) <= Zone) && (Zone <= (n+1)))
                {
                    n = Zone;
                }
```

```
#include <QtCore>
#include"cmdStruct.h"                    Included header cmdStruct.h is not used (fix available)


enum frameEnum { gpsFrame = 1, earthFrame, bodyFrame };

enum missionResultEnum { missionAccepted = 0, missionError,
                         missionUnsupportedFrame, missionUnsupported,
                         missionNoSpace, missionInvalid,
                         missionInvalidParam1,
                         missionInvalidParam2,
                         missionInvalidParam3,
                         missionInvalidParam4,
                         missionInvalidParam5X,
                         missionInvalidParam6Y,
                         missionInvalidParam7Z,
                         missionInvalidSequence,
                         missionDenied,
                         missionOperationCancelled
                       };

enum paramTypeEnum { paramUint8 = 0, paramInt8,
                     paramUint16, paramInt16,
                     paramUint32, paramInt32,
                     paramUint64, paramInt64,
                     paramReal32, paramReal64,
                     paramCustom
                   };

enum idPayloadEnum { IdMissionRequestList = 51, IdMissionCount, IdMissionRequest,
                     IdMissionItem, IdMissionResponse, IdMissionCurrent,
                     IdMissionSetCurrent, IdStatusText, IdMissionClearAll,
                     IdMissionItemReach, IdManualControl, IdParamValue,
                     IdParamRequestList, IdParamRequestRead, IdParamSet,idParamValueList
                     };


enum idListCout { coutXgps = 1,coutYgps,coutZgps,coutwx,coutwy,coutwz, // пока сделал перечч
                  coutpitch,coutroll,coutyaw,coutlatitude,coutlongitude,
                  coutd1,coutd2,coutd3,coutAx,coutAy,coutAz,coutXb,
                  coutYb,coutZb,coutMx,coutMy,coutMz,coutCRC,coutVx,
                  coutVy,coutVz,coutHeading,coutAierSpeed,coutMach,
                  coutAltitude,coutADF_Bearning,coutFuel};


//mission msg

struct StrMissionRequestList{
    uint8_t targetSystem {0};
    uint8_t targetComponent {0};
};

struct StrMissionCount{
```

Желаете познакомиться с интерфейсом программы? Всего за минуту вы узнаете, где и как используются наиболее важные элементы интерфейса пользователя. Ознакомиться можно и позже, для этого нужно зайти в Справка > Знакомство.

Fig . 4.1 General view of the software interface

The product is used in complex projects . It is popular for development on the Android platform . But its use requires significant resources.

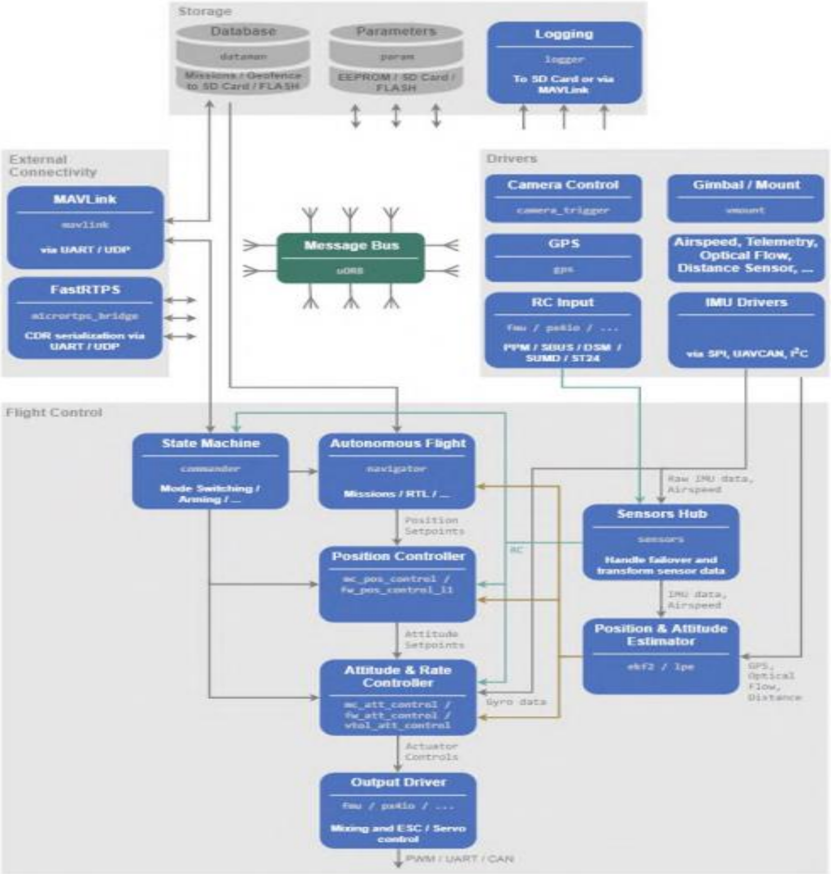
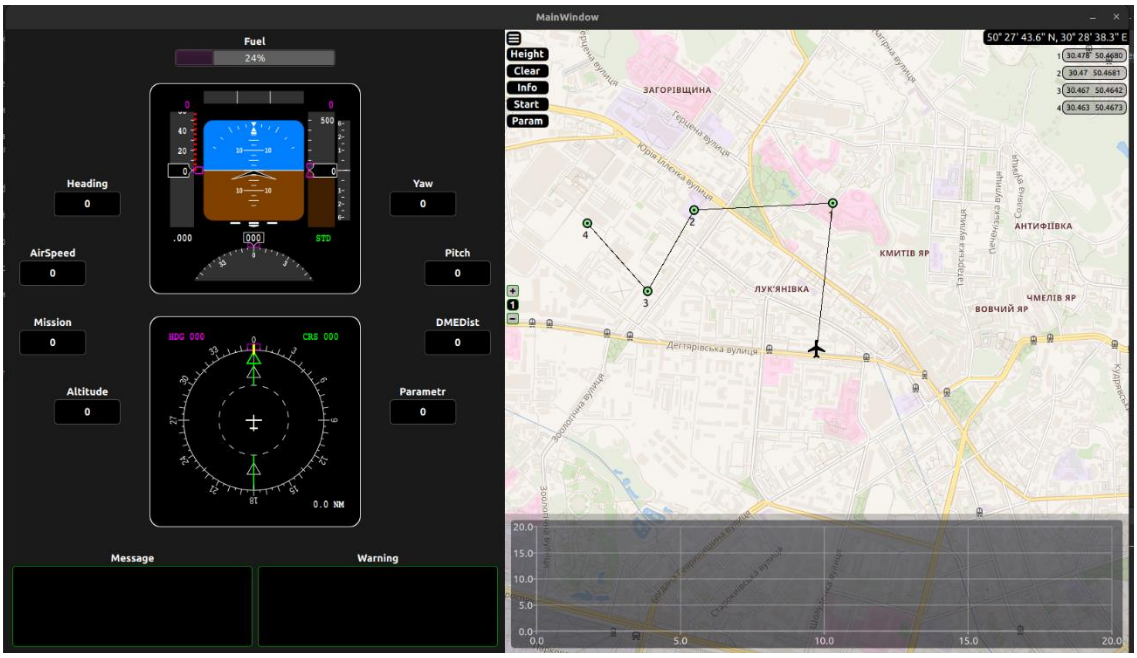


Fig 4.2 Project architecture

The project is divided into 3 main modules, this is the Interface implemented on the basis of the interpreted QTqml language , the software part where all the operations and action algorithms are described (the writing language is C++), and the server in which the UAV parameter transmission is simulated.

4.2 User (operator) interface

Let's start with the main tools for displaying and visualizing UAV parameters, which is implemented as follows:



MainWindow

Fuel 24%

Heading 0

AirSpeed 0

Mission 0

Altitude 0

Yaw 0

Pitch 0

DMEDist 0

Parametr 0

Message

Warning

50° 27' 43.6" N, 30° 28' 38.3" E

1	30.478	50.4680
2	30.47	50.4681
3	30.467	50.4642
4	30.463	50.4673

Fig. 4.3 Graphic implementation of UAV parameters monitoring

In the left part of the graphical interface, we can observe all the necessary parameters that are used in modern piloting realities.

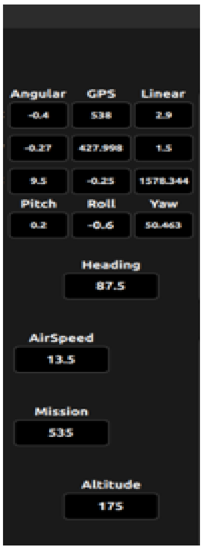
Two main digital instruments (QFlightinstruments) were taken as a basis:

- Electronic Attitude Direction Indicator (EADI)
- Electronic Horizontal Situation Indicator (EHSI)

As additional indicators, separate elements were added, which were also added manually:

1. GPS coordinates
2. Angular velocities
3. Angles of orientation
4. Latitude and longitude
5. delta controls
6. Linear acceleration
7. BINS coordinates
8. Magnetometer parameters
9. GPS course

10. Velocity vectors by GPS



Angular	GPS	Linear
-0.4	538	2.9
-0.27	427.998	1.5
9.5	-0.25	1578.344

Pitch	Roll	Yaw
0.2	-0.6	50.463

Heading 87.5

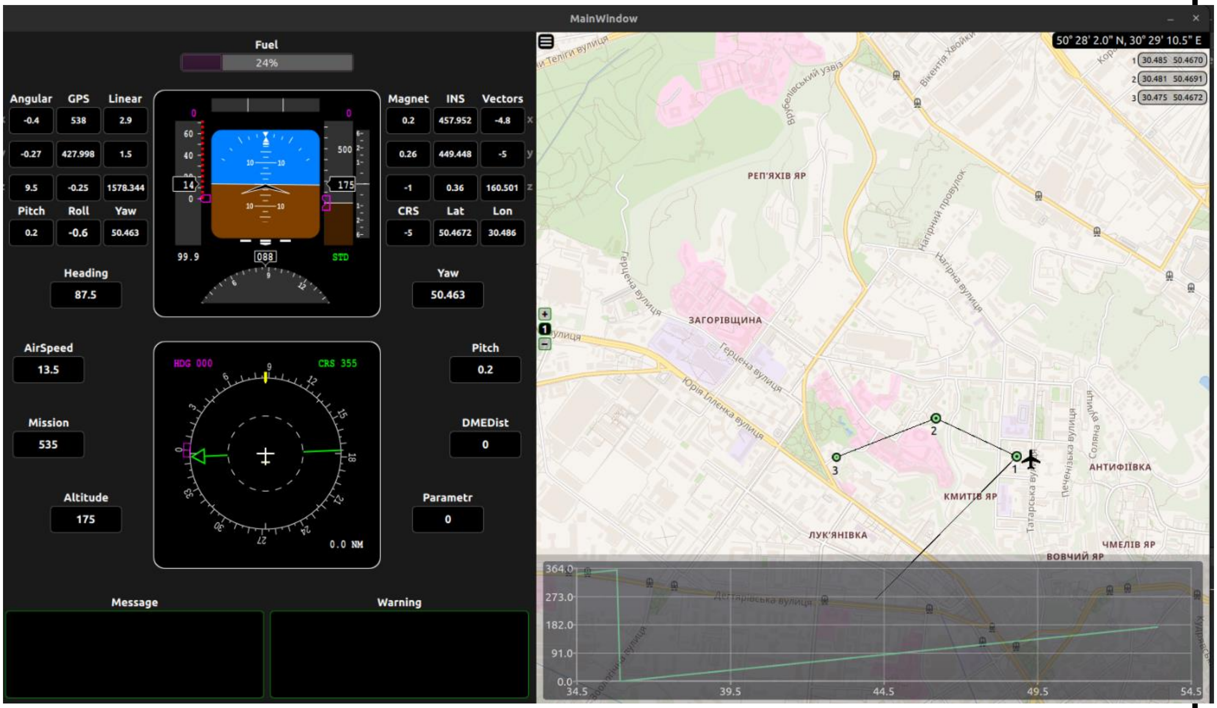
AirSpeed 13.5

Mission 535

Altitude 175

Fig. 4.4 Additional parameters of the UAV

The expanded parameters interface is shown in Fig. 4.5



Fuel 24%

Angular	GPS	Linear
-0.4	538	2.9
-0.27	427.998	1.5
9.5	-0.25	1578.344

Pitch	Roll	Yaw
0.2	-0.6	50.463

Magnet	INS	Vectors	
0.2	457.952	-4.8	x
0.26	449.448	-5	y
-1	0.36	160.501	z

CRS	Lat	Lon
-5	50.4672	30.486

Heading 87.5

Yaw 50.463

AirSpeed 13.5

Pitch 0.2

Mission 535

DMEDist 0

Altitude 175

Parametr 0

Message

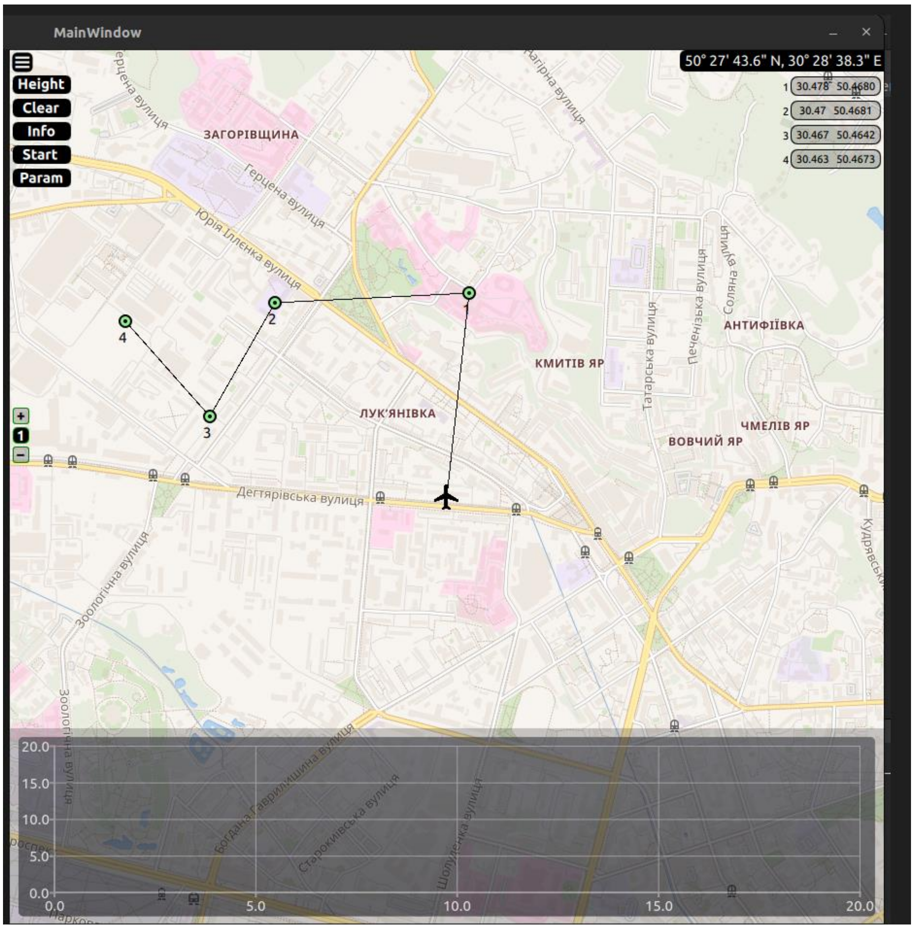
Warning

Fig. 4.5 Expanded interface of additional parameters

In the lower part of the interface there are additional messages , namely Message and Warning , which serve as additional information for control operators.

Fig. 4.5 Additional information messages

Now you can go to the right part of the interface (map and additional options).



50° 27' 43.6" N, 30° 28' 38.3" E

1	30.478	50.4680
2	30.47	50.4681
3	30.467	50.4642
4	30.463	50.4673

Fig. 4.6 Interface of the graphic position of the UAV on the map

This window contains:

1. Map;
2. Graph of dependence of speed and distance;
3. Route points (Missions);
4. Current location (Longitude, Latitude);

During the development of this interface, all the necessary indicators, cartographic subjects, a toolbar and dependency graphs were added, which will fully allow to have all the necessary and up-to-date information about the state of missions and UAV indicators.

4.3 Description of the software package

4.3.1 Logging system

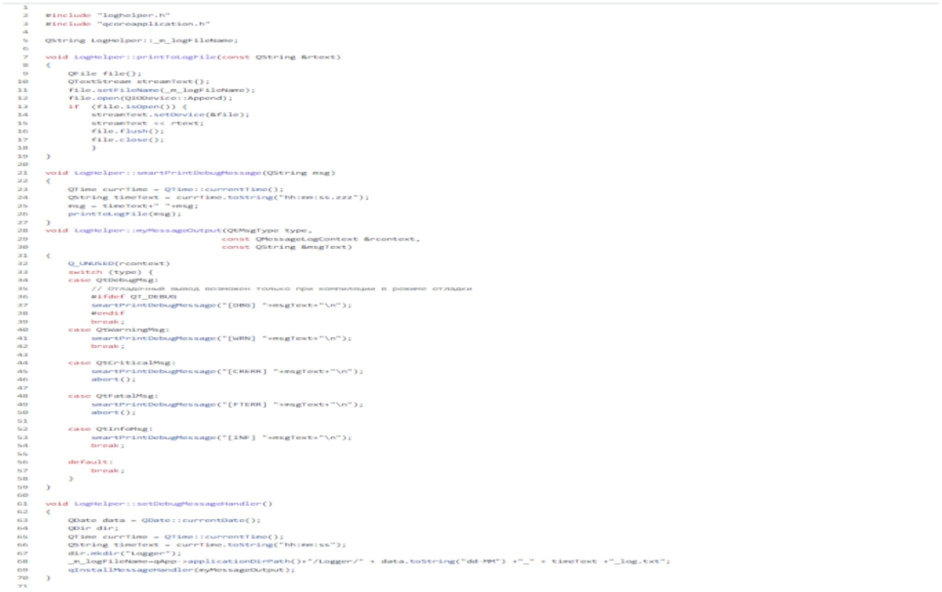
Logs (log files) are logs that contain information about system operation and certain user actions.

Their purpose is to record the operations performed on the machine for later analysis of these operations for system improvement or system diagnostics.

Implementation of the logging system was a difficult task as it was not necessary to reboot the system by constantly writing and reading its parameters and errors. To optimize the entire process, it was decided to develop this system as follows:

1. Intercept the entire console of our program by redirecting the flow of information into a text stream (buffer)
2. Dynamic creation of directory and log files (.txt) .
3. Each session, the flow of information is written to files
4. Debugging proceeds with a frequency of 1000 Hertz

After the session ends, the file is saved and the buffer is cleared



```
#include "loghelper.h"
#include "qcoreapplication.h"

QString LogHelper::_m_logFileName;

void LogHelper::printToLogFile(const QString &rtext)
{
    QFile file();
    QTextStream streamText{};
    file.setFileName(_m_logFileName);
    file.open(QIODevice::Append);
    if  (file.isOpen()) {
        streamText.setDevice(&file);
        streamText << rtext;
        file.flush();
        file.close();
        }
}

void LogHelper::smartPrintDebugMessage(QString msg)
{
    QTime currTime = QTime::currentTime();
    QString timeText = currTime.toString("hh:mm:ss.zzz");
    msg = timeText+" "+msg;
    printToLogFile(msg);
}
void LogHelper::myMessageOutput(QtMsgType type,
                                const QMessageLogContext &rcontext,
                                const QString &msgText)
{
    Q_UNUSED(rcontext)
    switch (type) {
    case QtDebugMsg:
        // Отладочный вывод возможен только при компиляции в режиме отладки
        #ifdef QT_DEBUG
        smartPrintDebugMessage("[DBG] "+msgText+"\n");
        #endif
        break;
    case QtWarningMsg:
        smartPrintDebugMessage("[WRN] "+msgText+"\n");
        break;

    case QtCriticalMsg:
        smartPrintDebugMessage("[CRERR] "+msgText+"\n");
        abort();

    case QtFatalMsg:
        smartPrintDebugMessage("[FTERR] "+msgText+"\n");
        abort();

    case QtInfoMsg:
        smartPrintDebugMessage("[INF] "+msgText+"\n");
        break;

    default:
        break;
    }
}

void LogHelper::setDebugMessageHandler()
{
    QDate data = QDate::currentDate();
    QDir dir;
    QTime currTime = QTime::currentTime();
    QString timeText = currTime.toString("hh:mm:ss");
    dir.mkdir("Logger");
    _m_logFileName=qApp->applicationDirPath()+"/Logger/" + data.toString("dd-MM") +"_" + timeText +"_log.txt";
    qInstallMessageHandler(myMessageOutput);
}
```

Fig. 4.7 Implementation of the logging system (C++)

After checking the work, we came to the conclusion that such manipulations will allow us not to reboot the system and correctly keep the event log of our program.

4.3.2 Mission establishment subsystem

Having drawn up the data exchange protocol and missions, you can proceed to the software implementation, the structural diagram is shown in Fig. 4.8

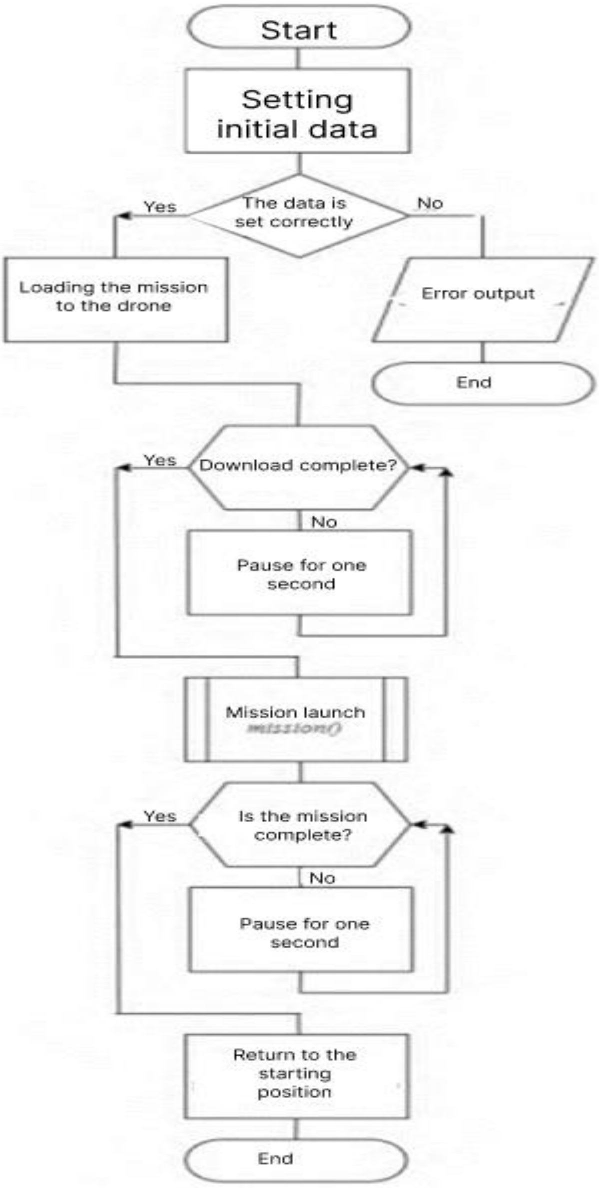


Fig. 4.8 Structural diagram of mission initialization

Any flight begins with the initialization of the initial parameters. Since we use our data packaging algorithm, which is a hybrid of the MavLink protocol and the created data structures (Fig. 4.9), which are multi-nested systems of identifiers, additional protection of our system against unauthorized access appears.

After each parameter initialization according to the developed transmission line protection algorithm, we include timers that check the delay and the correctness of identifiers, if everything is correct - the data is considered correct.

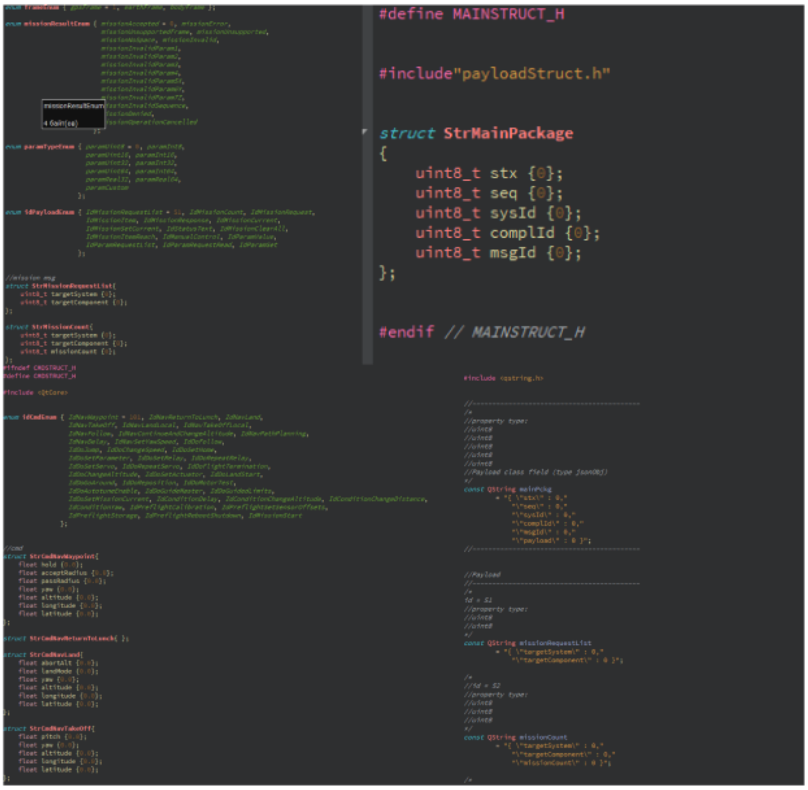
After the initial initialization, the mission is downloaded according to the same algorithm, after the NSU received a message about the mission being

downloaded, it sends a coded check for the correctness of the data recording. When the responses to the messages are correct, the final ready-to-execute signal is sent.

During the mission execution process, it is possible to monitor all subsystems of the UAV and log logs (which in the future makes it possible to check all security protocols).

After completing the mission, the UAV sends a signal about the completion of the mission, the NSU sends a response in which it gives a mission log, according to which the UAV checks whether all the points of the mission have really been completed, if so, the UAV sends a message about the final status of the mission.

Next, the waiting protocol for further commands is started, if the waiting time exceeds the norm, the aircraft returns to the previously set landing point.



```
enum frameEnum { gpsFrame = 1, earthFrame, bodyFrame };

enum missionResultEnum { missionAccepted = 0, missionError,
                         missionUnsupportedFrame, missionUnsupported,
                         missionNoSpace, missionInvalid,
                         missionInvalidParam1,
                         missionInvalidParam2,
                         missionInvalidParam3,
                         missionInvalidParam4,
                         missionInvalidParam5X,
                         missionInvalidParam6Y,
                         missionInvalidParam7Z,
                         missionInvalidSequence,
                         missionDenied,
                         missionOperationCancelled
                       };

enum paramTypeEnum { paramUint8 = 0, paramInt8,
                     paramUint16, paramInt16,
                     paramUint32, paramInt32,
                     paramUint64, paramInt64,
                     paramReal32, paramReal64,
                     paramCustom
                   };

enum idPayloadEnum { IdMissionRequestList = 51, IdMissionCount, IdMissionRequest,
                     IdMissionItem, IdMissionResponse, IdMissionCurrent,
                     IdMissionSetCurrent, IdStatusText, IdMissionClearAll,
                     IdMissionItemReach, IdManualControl, IdParamValue,
                     IdParamRequestList, IdParamRequestRead, IdParamSet
                   };

//mission msg
struct StrMissionRequestList{
    uint8_t targetSystem {0};
    uint8_t targetComponent {0};
};

struct StrMissionCount{
    uint8_t targetSystem {0};
    uint8_t targetComponent {0};
    uint8_t missionCount {0};
};
#ifndef CMDSTRUCT_H
#define CMDSTRUCT_H

#include <QtCore>

enum idCmdEnum { IdNavWaypoint = 101, IdNavReturnToLunch, IdNavLand,
                 IdNavTakeOff, IdNavLandLocal, IdNavTakeOffLocal,
                 IdNavFollow, IdNavContinueAndChangeAltitude, IdNavPathPlanning,
                 IdNavDelay, IdNavSetYawSpeed, IdDoFollow,
                 IdDoJump, IdDoChangeSpeed, IdDoSetHome,
                 IdDoSetParameter, IdDoSetRelay, IdDoRepeatRelay,
                 IdDoSetServo, IdDoRepeatServo, IdDoFlightTermination,
                 IdDoChangeAltitude, IdDoSetActuator, IdDoLandStart,
                 IdDoGoAround, IdDoReposition, IdDoMotorTest,
                 IdDoAutotuneEnable, IdDoGuidedMaster, IdDoGuidedLimits,
                 IdDoSetMissionCurrent, IdConditionDelay, IdConditionChangeAltitude, IdConditionChangeDistance,
                 IdConditionYaw, IdPreflightCalibration, IdPreflightSetSensorOffsets,
                 IdPreflightStorage, IdPreflightRebootShutdown, IdMissionStart
               };


//cmd
struct StrCmdNavWaypoint{
    float hold {0.0};
    float acceptRadius {0.0};
    float passRadius {0.0};
    float yaw {0.0};
    float altitude {0.0};
    float longitude {0.0};
    float latitude {0.0};
};

struct StrCmdNavReturnToLunch{ };

struct StrCmdNavLand{
    float abortAlt {0.0};
    float landMode {0.0};
    float yaw {0.0};
    float altitude {0.0};
    float longitude {0.0};
    float latitude {0.0};
};

struct StrCmdNavTakeOff{
    float pitch {0.0};
    float yaw {0.0};
    float altitude {0.0};
    float longitude {0.0};
    float latitude {0.0};
};
```

```
#define MAINSTRUCT_H


#include"payloadStruct.h"


struct StrMainPackage
{
    uint8_t stx {0};
    uint8_t seq {0};
    uint8_t sysId {0};
    uint8_t complId {0};
    uint8_t msgId {0};
};


#endif // MAINSTRUCT_H
```

```
#include <qstring.h>

//-----------------------------------------
/*
//property type:
//uint8
//uint8
//uint8
//uint8
//uint8
//Payload class field (type jsonObj)
*/
const QString mainPckg
        = "{ \"stx\" : 0,"
          "\"seq\" : 0,"
          "\"sysId\" : 0,"
          "\"complId\" : 0,"
          "\"msgId\" : 0,"
          "\"payload\" : 0 }";
//-----------------------------------------


//Payload
//-----------------------------------------
/*
id = 51
//property type:
//uint8
//uint8
*/
const QString missionRequestList
        = "{ \"targetSystem\" : 0,"
          "\"targetComponent\" : 0 }";

/*
//id = 52
//property type:
//uint8
//uint8
//uint8
*/
const QString missionCount
        = "{ \"targetSystem\" : 0,"
          "\"targetComponent\" : 0,"
          "\"missionCount\" : 0 }";

/*
```

Fig. 4.9 Part of the hybrid implementation of the protocol

CHAPTER 5. PROTECTION OF THE ENVIRONMENT

Environmental protection is a system of measures for the rational use of natural resources, the preservation of particularly valuable and unique natural complexes, and the provision of environmental safety. This is a set of state, administrative, legal, economic, political and social measures aimed at the rational use, reproduction and preservation of natural resources of the earth, limiting the negative impact of human activity on the environment.

The main principles of environmental protection are (Article 3 of the Law):

- prioritization of environmental safety requirements, mandatory compliance with environmental standards, regulations and limits on the use of natural resources when carrying out economic, managerial and other activities;
- guaranteeing an ecologically safe environment for people's life and health;
- the preventive nature of the measures for the protection of the natural environment;
- greening of material production based on the complexity of solutions in matters of environmental protection, use and reproduction of renewable natural resources, wide implementation of the latest technologies;
- mandatory environmental examination;
- openness and democracy in decision-making, the implementation of which affects the state of the natural environment, the formation of the population's ecological outlook;

ACIC DEPARTMENT				NAU 22 11 86 000 EN			
Performed	A.O. Tsoba			Control and Monitoring System of an Unmanned Aerial Vehicle. Monitoring Subsystem (complex).	N.	Page	Pages
Supervisor	V.M.Sineglasov						
Consultant	Lavniuk A.A.				225 151		
S. controller	M.K. Filyashkin						
Dep. head	V.M.Sineglazov						

- scientifically based regulation of the impact of economic and other activities on the surrounding natural environment;
- compensation for damage caused by violation of protection legislation
natural environment;
- establishment of environmental tax, fee for special use of water, fee for special use of forest resources, fee for subsoil use in accordance with the Tax Code of Ukraine.

The legislation of Ukraine establishes standards for the use of natural resources and other environmental standards.

Environmental standards establish the maximum permissible emissions and discharges into the environment of polluting chemical substances, the levels of permissible harmful effects of physical and biological factors on it (Article 33 of the Law).

Norms of maximum allowable concentrations of pollutants in the surrounding natural environment and levels of harmful physical and biological effects on it are uniform for the entire territory of Ukraine. Enterprises, institutions and organizations whose activities are associated with a harmful impact on the surrounding natural environment, regardless of the time of their implementation, must be equipped with structures, equipment and devices for cleaning emissions and discharges or their neutralization, reducing the impact of harmful factors, as well as control devices for quantity and composition of pollutants and characteristics of harmful factors (Article 51 of the Law).

5.1 Application UAVs for protection nature

Poaching and change climate sharply affect health _ wild nature in everything the world According to the data World Wildlife Fund of nature , every year are dying out thousands species _ In order to help fight this one trend , environmentalists apply innovative methods protection and study of our global ecosystems .

Combined with geospatial _ images now drones are used for surveillance and tracking animals . Team from Liverpool schools of Natural Sciences of John Murabuduye University an autonomous drone system that can monitor and transmit endangered species information about their well- being back researchersambDJI worked in nature protection spacious , offering services unmanned flying devices to _ help teams conduct research without disturbing natural environment existence _ The Ocean Alliance is an example of an organization that used drones (for example , marine SnotBot) to collect samples - in particular , mucus from whales _ Apart from relief of research ecosystems , drones also they can to allow environmentalists fight poachers . _

5.2 Waste control _

. Application BpLA for waste control Recycling and biodegradation improved global management waste _ However innovation in assembly waste still _ appear , in particular drones that _ help clean up oceans _ RanMarine manages unmanned marine transport with a device similar to Roomba , which used for collection waste in ports and harbors , while RedZone Robotics focuses on jobs that are used to support management systems wastewater .

5.3 Use of UAVs for tracking diseases

Animal tracking too allows researchers track diseases _ London School of Hygiene and Tropical of medicine used drones with thermal imaging cameras to track the movement of macaques in Palawan province in the Philippines - a region where malaria is an active threat . Possibility following these animals provided further insight _ possible transfer infectious disease and its jumps from animals to humans. Similarly , Microsoft uses technology unmanned flying devices for capturing and testing mosquitoes on infectious disease _

Ideally _ this one information can be used for protection local residents , and in the future maybe help to prevent epidemics even before they start. More one disease with which fight for help drones , there is schistosomiasis - a tropical disease that is called parasitic worms _ A team of researchers that consists of scientists from the University of Washington and Stanford , started experimental method for tracking distribution and forecasting transmission of schistosomiasis. Instead use of animals, approach teams uses unmanned and satellite snapshots for tracking presence " unconquered , floating vegetation "where snails , which transmit the disease, live in their own places - search these seats with the help of unmanned pictures allows those researchers to know which ones districts are located above risk infection with schistosomiasis

5.3 Use of UAVs in environmental monitoring

In terms of increase in man-made burden and increase risk man-made disasters are necessary modernization existing systems monitoring surrounding the environment , especially the atmosphere . To solve this one problems was a monitoring system has been developed pollution surrounding environment that uses helicopter (MBLA-VT) or airplane apparatus (MBLA-C) of a small UAV and equipped with a ground station management . Disadvantages of existing monitoring systems is a problem to determine level pollution on different heights from sources . Limited data , absence quantity points measurements to determine . Exploring dangerous territory and receiving level information _ pollution , the ground system reacts slower in emergency situations .

Method of using UAV in monitoring surrounding environment
is divided into several stages :
• tasks ecological monitoring ;
• choice technical means for monitoring surrounding environment
(to determine opportunities means measuring techniques , to determine number used drones , define the best route for

research);

- choice technical means collection data surrounding environment ;
- measurement parameters surrounding environment ;
- processing parameters surrounding environment ;
- comparison ecological parameters with environmental standards;
- analysis parameters surrounding environment ;
- forecasting the state of the environment in the future .

Surveillance complex maybe perform task ecological site supervision _ Location mining enterprises for the purpose of detection violations of protection zones health and borders . Withdrawal land plots for industrial enterprises .

The complex includes a drone from useful load in in the form built -in onboard cargo , which includes a methane detector, a gas analyzer , a radiometer , a thermal imager , pilemeters , a course camera and an air camera. The complex uses a recycling system transfers data to execute everyone tasks in automatic and semi-automatic modes. Now this product is the most attractive and different from similar products by that , in this case , he collects as much as possible ambulance equipment _ evaluations pollution air _

Use in solution questions education , research and applied programs are used complexes observation with a child by helicopter or flying device unmanned flying devices _ The complex provides remote aerial observation , video and aerial photography objects in the area and height from 50 to 600 m, monitoring thermal images , measurements pollution of atmospheric radiation , detection methane leakage , quantitative definition oxygen , carbon monoxide , dioxide carbon , nitrogen oxides , nitrogen dioxide , dioxide sulfur , hydrogen sulfide , measurement temperature and pressure . UAV and control station can be operated when it is 20 km from each other one when the radios are visible to each other. One of final MK products are research in real mode .

Get a large amount points measurement on several high altitude levels from 0 to 1000 m in steps of 50-100 m. With help these points measurement you can reproduce quickly three-dimensional model of transportation polluting substances and effectively determine the main ones sources pollution on different distances from sources

Using developed complex will bring unique opportunities for organizations that _ participate in monitoring surrounding environment because _ drones they can compared to all by existing methods of monitoring the atmospheric environment , to carry out control by direct measurement methods . Direct measurement method maybe to provide high precision monitoring volume air environment . Now no monitoring system can _ solve similar programs . Scientific development is a novelty innovative techniques research air that _ applies new ones measuring devices installed on drones _ flying devices . These devices decide urgent system problem monitoring fast ones natural and technical changes environment . This the innovative project is unique because _ _ theoretical research and practical experience teams developers allows create small devices with the most useful load and automatic transmission of the test information

What tasks of the State Inspectorate will help drones to solve?

- Monitoring and protection of environmental objects: lands, forests, water resources, nature reserves, plants and animals. Drones are equipped with high-resolution cameras and GPS navigators. This allows you to carry out aerial photography at any time and transmit up-to-date data with an accurate reference to the location of objects in real time.

– Detection of unauthorized use of territories (landfill, deforestation) and illegal hunting. Regular inspection will help find intruders and quickly take measures to eliminate incidents.

– Analysis of the level of air pollution and radiation. Control of the handling of waste and hazardous chemicals (agrochemicals, pesticides). Gas analyzers and dosimeters can be installed on board drones. The implementation of such technologies will optimize the work of the State Inspection team and protect it from potential threats.

Advantages of using UAVs

- Unmanned aerial vehicles are able to cover long distances in a short period of time, record information and instantly transmit it to the control point.

- The technical characteristics of drones allow round-the-clock inspection of hard-to-reach and dangerous areas from any distance.

- With the help of aerial survey data, you can create 3D models and terrain plans. This will simplify the analysis of the state of natural objects and make it possible to draw up an effective action plan.

– Saving financial costs and optimization of human resources. Environmental monitoring with a drone does not require physical effort and costs less than the work of a whole team.

Detection of forest fires, coordination of ground crews during their elimination

Patrol of the forest fund is carried out on an aircraft-type UAV equipped with a controlled gyro-stabilized video camera with a 28x magnification with an overview of the entire lower hemisphere (360 degrees, endless rotation). Patrol flight in normal visibility is carried out at an altitude of 600-800 meters. Noticing the smoke, the operator directs the UAV to the smoke point. To document forest fires, photographing of the places of their occurrence is carried out using a camera or video camera. Each forest fire is monitored from the air from the moment it is discovered until it is completely eliminated. The flyby of the fire is carried out 2-3 times a day. At each inspection, the boundaries of the fire and its area are plotted on the map to assess the dynamics of the spread. The operator determines the main direction of spread and the threats associated with it, the presence of separate combustion centers, dangerous areas, places where fire passes through mineralized strips, and identifies the location of people and equipment. A detailed inspection of an active forest fire (control over the work of forest fire teams) is carried out from a height of 200-400 m. Combined (optical and IR range) video cameras are used to detect hidden combustion sources.

Accurate aerial census of wild animals

Aerial census of wild animals is one of the most accurate ways to determine their abundance, spatial distribution, and places of concentration. Based on the photo, video and thermal imaging received from the UAV, the detected animals are counted in the inspection area. The resulting images are tied to geographic coordinates to determine the coordinates of the animals. Photographs and thermal images of animals are superimposed on each other for automatic accounting. Based on the data obtained, the number is recalculated according to approved state methods for the entire study area.

Ukrainian start-up Aerodron has created a multifunctional cargo drone that can process fields from the air. The founder of the startup Yuriy Pederii said this at the Pitch Day No. 1 competition of the Ukrainian Startup Fund in Kyiv.

"We developed this plane so that it would cultivate fields three times faster, ten times cheaper, and, of course, the plane does not damage plants or loosen the soil like a tractor," he said.

Pederius added that the plane also does not use water for flights and is completely ecological, unlike tractors.

Startup "Aerodron" entered the top three in the ranking based on the results of the Pitch Day No. 1 competition of the Ukrainian Startup Fund. A total of 12 startups made it to the finals of the competition, competing for a $75,000 grant.

Back in July 2019, the government presented the Ukrainian Startup Fund. On December 2, the foundation began accepting grant applications. In total, the Ukrainian Startup Fund received more than 300 applications for grants.

CHAPTER 6. LABOR PROTECTION

6.1 System of labor protection measures

The system of labor protection measures deals with the development of means to ensure the safety of life and health of employees in the course of their work, that is, this system includes measures that, individually or collectively, are aimed at creating working conditions that meet the requirements for preserving life and health employees in the process of work.

Labor protection is based on a complex of state legislative acts. The general laws of Ukraine that determine the main provisions on labor protection are the Constitution of Ukraine, the Code of Labor Laws, the Law of Ukraine "On Labor Protection", the Law of Ukraine "On Mandatory State Social Insurance against Industrial Accidents and Occupational Diseases that Caused Loss" working capacity" and by-laws on labor protection.

The technical operation of electrical equipment of airplanes and airports is associated with the danger of damage to engineering and technical personnel by electric current.

In the state standard of Ukraine DSTU 2293-99 "System of labor safety standards. Occupational Health. Terms and definitions" established terms and definitions of the main concepts of labor protection. Here are some of them:

Labor protection is a system of legal, socio-economic, organizational and technical, hygienic or medical and preventive measures and means aimed at preserving the health and working capacity of a person in the process of work;

Harmful (production) factor production factor, the impact of which can lead to deterioration of the health status and reduction of the employee's working capacity;

ACIC DEPARTMENT		NAU 22 11 86 000 EN	
Performed	A.O. Tsoba	Control and Monitoring System of an Unmanned Aerial Vehicle. Monitoring Subsystem (complex).	N. Page Pages
Supervisor	V.M.Sineglasov		
Consultant	Kozlitin O.O.		225 151
S. controller	M.K. Filyashkin		
Dep. head	V.M.Sineglazov		

A dangerous (production) factor is a production factor, the influence of which in certain conditions can lead to injuries or other sudden deterioration of the employee's health;

6.2 Analysis of working conditions at the workplace Organization of the workplace

In order to create favorable conditions for visual work, which would exclude rapid eye fatigue, the occurrence of occupational diseases and contribute to increasing labor productivity, industrial lighting must meet the requirements of SNiP II-4-79 "Estestvennoe and artificial lighting. Design standards", DBN V.2.5-28-2006 "Natural and artificial lighting", where the main requirement is the need to create lighting on the work surface that corresponds to the nature of visual work and is within the established norms. The lighting in the room with VDT should be combined, in which the insufficient natural lighting according to the standards is supplemented with artificial lighting. Natural lighting should be lateral, preferably one-sided. It is best when the windows are oriented to the north or northeast, this will make it possible to eliminate the unwanted blinding effect of the sun's rays. Windows must be equipped with adjustable devices (blinds, curtains, external visors, etc.). The surface of the floor of the room with VDT should be flat, non-slippery, convenient for cleaning and wet cleaning, have antistatic properties. The area on which one workplace with a computer is located should be at least 6.0 square meters, and the volume of the room should be at least 20.0 cubic meters. Premises with VDT must be equipped with first aid kits. Proper organization of workplaces helps to eliminate general discomfort, reduce worker fatigue, and increase productivity.

The organization of the workplace involves:

- correct placement of the workplace in the production premises; - selection of an ergonomically sound working position, industrial furniture with
taking into account anthropometric characteristics of a person;

- rational arrangement of equipment at workplaces; - taking into account the nature and peculiarities of labor activity.

The workplace includes a table, a chair, and a computer with a real-time monitor on the table.

The size of the room for VDT is length a = 7 m, width c = 4.5 m, height h = 3.5 m.

According to DNAOP 0.00-1.3 1-99, the location of premises intended for work with VDT in basements and basement floors is unacceptable. It is also prohibited to locate explosive premises of category A and B (ONTP 24-86) and factories with "wet" technological processes next to premises where computers are located, as well as above or below such premises. In addition, production premises for working with VDT should not adjoin premises in which the level of noise and vibration exceeds permissible values.

Since the area of the premises S= 7x4.5=5 sq.m., and the area on which one workstation with VDT is located, must be at least 6.0 sq.m., no more than five computerized worm places can be placed in this premises.

The volume of the room is Shh = 31.5 x 3.5 = 110.25 cubic meters. and the volume that falls for one workplace - 110.25: 5 = 22.05 cubic meters.

We plan the placement of computerized workplaces in the premises taking into account the following requirements:

- workplaces with VDT are located at a distance of at least 1 m from the wall with light openings (windows);
- the distance between the side surfaces of the VDT should be at least 1.2 m; - the distance between the back surface of one VDT and the screen of another should not
be less than 2.5 m;

- the passage between rows of workplaces must be at least 1 m; It is also necessary to take into account the size of furniture at computerized workplaces, in particular,

the desktop. According to DNAOP 0.00-1.31-99, the recommended dimensions of the table for a workplace with VDT are: height - 725 mm, width - 600-1400 mm, depth 800-1000 mm. We assume that the desktop has the following dimensions: width - 1200
mm, depth - 800 mm.

It is best to place computerized workplaces in rows along a wall with windows. This will make it possible to exclude the specular reflection of natural light sources (windows) on the VDT screen and the latter falling into the operators' field of vision, which impairs their visual performance.

Here is the plan of the production premises with computerized workplaces (Fig. 5.1).

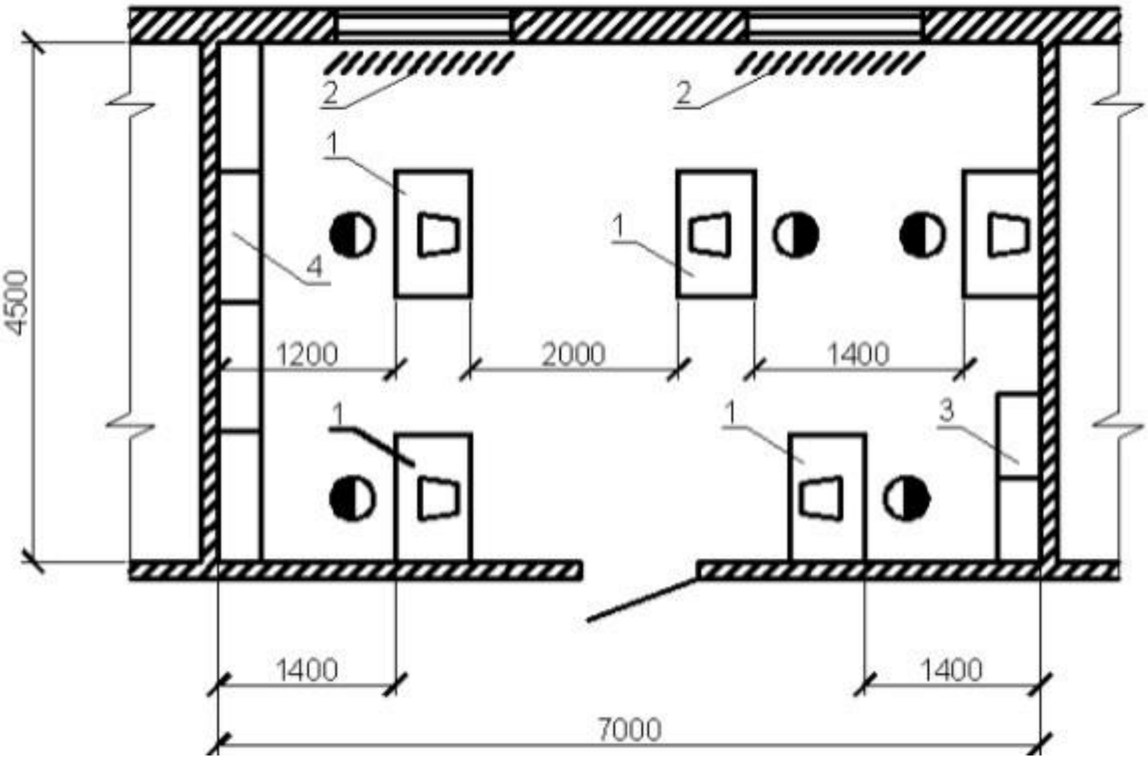


Fig. 5.1 Plan of a production facility with computerized workplaces

1 - computerized workplace with VDT; 2 - sun blinds;

3 - cabinets for storage of diskettes and software; 4 – cabinets for storing documentation and professional literature.

6.3 Analysis of harmful and dangerous production factors

The subject of labor protection within the laboratory and his workplace is affected by the following harmful and dangerous production factors:

1) non-ionizing electromagnetic fields and radiation; 2) production noise;

3) artificial lighting;

4) electrical network and increased value of voltage in the electrical circuit, the closing of which in certain situations can occur through the human body;

5) harmful substances in the air of the working area.

Non-ionizing electromagnetic fields and radiation

A research engineer is exposed to industrial frequency electromagnetic radiation from machinery and equipment and radio frequency electromagnetic radiation at his workplace.

According to DSNiP 3.3.6.096-2002 "State sanitary norms and rules when working with sources of electromagnetic fields", the intensity of the electromagnetic field of industrial frequency (50 Hz) acting on the research engineer should not exceed 5 kV/m.

Production noise

According to DSN 3.3.6.037-99 "Sanitary norms of industrial noise, ultrasound and infrasound", noise levels in the research laboratory acting on the research engineer should be 50 dB, and the actual value of the noise level is 55-65 dB. This is due to the presence in the premises of air conditioning, ventilation and air heating, which generate noise. At the workplace there are such types of noise as mechanical, ultrasonic, electromagnetic and aerodynamic; there is no infrasound.

Natural and artificial lighting

According to DBN-V.2.5.-28-2006 "Natural and artificial lighting", the standardized indicators of illumination at the workplace of a research engineer should be 300-500 lux, and the actual value of illumination is 250-430 lux. This is due to the obsolescence of the lighting system. The minimum illumination of work surfaces in production premises is determined mainly by the characteristics of visual work. Standardized indicators are of an interdisciplinary nature.

Local lighting is lighting added to the general, which is created by lamps that concentrate the light flow directly on workplaces. Alternate lighting – lighting in the absence of the main work process.

Emergency lighting is divided into safety lighting and evacuation lighting. Evacuation lighting: 0.5 lux indoors, 0.2 lux in open areas.

The workplace uses mixed lighting. As natural lighting in this room, one-way lighting with the help of three windows is used.

For artificial lighting in this case, light sources with a sufficiently high efficiency in general lighting fixtures, located evenly over the entire area of the room, are needed. LED lamps, which have one of the highest light output rates, are best suited in this case.

With correctly calculated and executed lighting of production premises, the worker's eyes for a long time retain the ability to distinguish objects well, without tiring. Such conditions contribute to the reduction of occupational injuries and occupational eye diseases.

6.4 Development of labor protection measures

The main regulatory document that defines the requirements for the safe operation of computing equipment is DNAOP 0.00-1.31 -99 "Labor safety rules during the operation of electronic computing machines".

Persons who have mastered the appropriate practical course, read the instructions and received instruction on occupational health and safety at the workplace, have passed a medical examination and have no contraindications to working with VDT may be allowed to work with computer equipment.

It is necessary to comply with the established safety requirements before starting work, for example, before turning on the computer to the network, it is necessary to make sure that the equipment is grounded, that the power cord and other conductors are in good condition. Check the reliability and correct installation of the equipment on the work table, turn on the air conditioning system of the workplace, adjust the lighting of the workplace, etc.

If any malfunctions are detected, the work cannot be started and the manager or other responsible person should be notified.

When performing work, it is forbidden to work without proper lighting, close the ventilation holes of the equipment, leave the equipment turned on unattended, and allow outsiders to work on the equipment.

And in order to prevent the negative impact on health of harmful production factors, it is necessary to observe the regimes of work and rest. After each hour of work at the display, you must take a 10-15 minute rest break. During scheduled breaks, it is recommended to perform special exercises and self-massage of the hands and eyes, as well as conduct a session of psychophysiological relief in a specially equipped room.

After finishing work, you should turn off the printer, display, processor and other equipment, pull out the plugs from the sockets, clean the workplace, carefully wash your hands with warm water and soap, also turn off the air conditioner, lighting and general power supply unit.

6.5 Fire safety of the production premises

The main regulatory document regulating fire safety requirements is the Law of Ukraine "On Fire Safety". This Law defines the general legal, economic and social foundations of ensuring fire safety on the territory of Ukraine, regulates the relations of state bodies, legal entities and individuals in this field regardless of their type of activity and forms of ownership.

According to the Law of Ukraine "On Fire Safety", ensuring the safety of the enterprise, institution is entrusted to the manager or an authorized person. Managers are obliged to:

1 Develop a set of measures to ensure fire safety (in this case, in premises with computing equipment);

2 Develop and approve instructions, provisions, rules regarding fire safety and control their implementation;

3 Organize training of employees on fire safety;

4 Keep the means of fire protection and communication, fire equipment, equipment and inventory in good condition, do not use them for purposes other than their intended purpose.

5 Conduct official investigation of fire incidents.

Persons who have not been instructed in fire safety cannot be allowed to work. Each employee is obliged to comply with fire safety requirements, as well as to take measures to eliminate violations of fire safety rules, eliminate fires and fires. Every employee must know the location of primary fire extinguishing equipment and be able to use them, employees must know the rules of behavior during a fire, evacuation routes. In the event of a fire, employees must immediately notify the fire department (call) and the management of the institution and start extinguishing the fire with all available means.

In addition to general fire safety requirements, special fire prevention measures are implemented

activities For buildings and premises in which video terminals and computers are operated, such measures are defined by the Rules on fire safety in Ukraine NAPB A.01.001-2004, the Rules on occupational safety during the operation of electronic computing machines DNAOP 0.00-1.31-99 and other regulations documents

Buildings and their parts, in which computers are located, must not be lower than П degree of fire resistance. Above and below the premises where computers are located, as well as in the premises adjacent to them, the location of premises of categories A and B is not allowed in order to prevent explosion and fire hazards. Category B premises should be separated from premises with computers by fire walls.

Rules for installation of electrical installations. Appropriate markings must be placed on the entrance door of the premises.

Stores of information media, important documentation, and spare equipment should be placed in separate rooms equipped with non-combustible racks and cabinets. In the premises where work computers are located, only those media that are necessary for current work should be kept.

Sound-absorbing cladding of walls and ceilings in computer rooms should be made of non-flammable or flame-resistant materials.

rooms with computers with carbon dioxide fire extinguishers at the rate of 2 pcs. for every 20 square meters of the premises, taking into account the maximum permissible concentration of the fire-extinguishing substance. The distance from the possible source of fire to the location of the fire extinguisher should not exceed 20 meters according to the regulations.

6.6 Conclusions from the section

The need to comply with the labor protection measures described in the section is necessary for implementation.

First of all, because the highest value is always a person, his life and health. Neither the amount of wages, nor the level of profitability of the enterprise, nor the value of the produced product can serve as a basis for disregarding safety rules and justifying existing threats to the life or health of employees. In addition, in this case it is also about the values of a specific person as an employee with his own knowledge, skills and experience.

Secondly, properly organized work to ensure labor safety increases the discipline of employees, which, in turn, leads to increased labor productivity, a decrease in the number of accidents, equipment breakdowns and other out-of-hours situations, that is, ultimately increases the efficiency of production.

Thirdly, labor protection means not only ensuring the safety of employees during the performance of their official duties. In fact, this also includes a wide variety of measures: for example, the prevention of occupational diseases, the organization of full rest and food for employees during work breaks, providing them with the necessary overalls and hygiene products, and even the implementation of social benefits and guarantees. The correct approach to the organization of occupational health and safety at the enterprise, the competent use of various non-material ways of motivating employees give the latter the necessary sense of reliability, stability and management interest in their employees. Thus, thanks to established labor protection, staff turnover also decreases, which also has a beneficial effect on the stability of the entire enterprise.

CONCLUSIONS

According to the purpose of the diploma, the first chapter analyzed the features of the UAV and its types. The control station of the UAV , as an object of research, and the data monitoring subsystem were considered.

In the second section, the following issues were considered:

1. Problems arising from the management of UAV security there .

2. Existing solutions and comparison of control line protection algorithms.

3. GCS\UAV data transfer protocol.

4. Subsystem structure

The full task of the project was formulated, and the task:

1. Modernize the level of protection of the data exchange protocol, and write a software implementation of the protocol that will make the system more secure.

2. To analyze the features of signal formation in remote control systems of drones

3. Write a new algorithm for the line of protection that will allow the control line to receive more protection.

4. Develop a system by which it is possible (in the presence of an operator and a specialist) to quickly diagnose the state of the UAV and its systems (software failures).

5. To develop a light and maximally optimized UAV control interface , which will not have a heavy appearance and will be easy to use.

In the third section, specific solutions and means of eliminating the weakness of the UAV defense line were proposed in the form of an algorithm. And the algorithm is based on successive inspections energy , modulation and structural signs of the signal, assumes the possibility of automatic detection of the accompanying signal its frequency. Offered specific hardware and software solutions for building a UAV prototype.

The results of practical testing are presented possible problem solutions on the finished prototype and the results are given functioning developed components connection _

The proposed specific solutions for the protection of the cyber network , which are based on the backup and recovery system, consist of the following sequence of functions :

1. Duplication allows simultaneous copying to several sources, which increases the reliability of data storage.
2. Deduplication allows analysis and compression of duplicated data. As a result, the load on data transmission channels and data storage space is reduced.
3. Creation of system images. Periodic copying of not only data, but also system images allows you to quickly restore an employee's workplace even in case of damage to the operating system or personal computer, which ensures the continuity of his work.
4. Load balancing. Allows you to optimize the load on several storages for the fastest performance of backup operations.
5. Compatibility with software (operating systems and DBMS). Allows you to create "casts" of files and databases, which can change during the backup process, for their correct and complete transfer and restoration.
6. Various tools for remote administration. This is a fairly diverse set of functions that allow you to automate the administrator's work. These may include remote installation of agents on users' computers, verification of created archives, manual or automatic merging of backup copies , etc.
7. Work with virtual devices.
8. Working with "cloud" storage.
9. Data recovery algorithms.

In the fourth chapter there was:

1. The considered architecture of the project.
2. Developed user interface.
3. The developed software and its systems are described

In the fifth and sixth chapters, the issues of ecology and labor protection were considered.

Thus, all assigned tasks were completed in full.

REFERENCES

1. Atherton K.D. The faa says there will be 7 million drones flying over america by 2020. Popular Sci. 2016

2. Vattapparamban E., Güvenç İ., Yurekli A.İ., Akkaya K., Uluağaç S. Wireless Communications and Mobile computing Conference (IWCMC), 2016тInternational. IEEE; 2016. Drones for smart cities: issues in cybersecurity, privacy, and public safety; pp. 216–221.

3. Dalamagkidis K., Valavanis K.P., Piegl L.A. On integrating unmanned aircraft systems into the national airspace system. Springer; 2012. Aviation history andunmanned flight; pp. 11–42.

4. Altawy R., Youssef A.M. Security, privacy, and safety aspects of civilian drones: a survey. ACM Trans. Cyber-Phys. Syst. 2017;1(2):7.

5. Marshall D.M., Barnhart R.K., Hottman S.B., Shappee E., Most M.T. Crc Press; 2016. Introduction to unmanned aircraft systems. 6. Chen M., Challita U., Saad W., Yin C., Debbah M. Machine learning for wireless networks with artificial intelligence: a tutorial on neural networks. arXiv Preprint arXiv:1710.02913. 2017

7. Dinger J., Hartenstein H. Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on. IEEE; 2006. Defending the sybil attack in p2p networks: taxonomy, challenges, and a proposal for self-registration; pp. 8–pp.

8. Fotouhi A., Qiang H., Ding M., Hassan M., Giordano L.G., GarciaRodriguez A., Yuan J. Survey on uav cellular communications: practical aspects, standardization advancements, regulation, and security challenges. arXiv Preprint arXiv:1809.01752. 2018

9. Uragun B. Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on. Vol. 2. IEEE; 2011. Energy efficiency for unmanned aerial vehicles; pp. 316–320.

10. Irizarry J., Gheisari M., Walker B.N. Usability assessment of drone

technology as safety inspection tools. J. Inf. Technol. Construct. (ITcon) 2012;17(12):194–212.

11. Abid M.E., Austin T., Fox D., Hussain S.S. Drones, uavs, and rpas: an analysis of a modern technology. Worcester Polytech. Inst., Worcester, Massachusetts. 2014

12. Kopardekar P.H. 2014. Unmanned aerial system (uas) traffic management (utm): Enabling low-altitude airspace and uas operations.

13. Motlagh N.H., Taleb T., Arouk O. Low-altitude unmanned aerial vehiclesbased internet of things services: comprehensive survey and future perspectives. IEEE Internet Things J. 2016;3(6):899–922.

14. Yang L., Qi J., Xiao J., Yong X. Intelligent Control and Automation (WCICA), 2014 11th World Congress on. IEEE; 2014. A literature review of uav 3d path planning; pp. 2376–2381.

15. Ueno S., Higuchi T. Numerical Analysis-Theory and Application. InTech; 2011. Collision avoidance law using information amount.

16. Brandt A.M., Colton M.B. Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on. IEEE; 2010. Haptic collision avoidance for a remotely operated quadrotor uav in indoor environments; pp. 2724–2731.

17. Hernandez-Hernandez L., Tsourdos A., Shin H.-S., Waldock A. Unmanned Aircraft Systems (ICUAS), 2014 International Conference on. IEEE; 2014. Multiobjective uav routing; pp. 534–542.

18. Lipsitch M., Swerdlow D.L., Finelli L. Defining the epidemiology of covid19-studies needed. N. Engl. J. Med. 2020 [PubMed]

19. . Mansfield K., Eveleigh T., Holzer T.H., Sarkani S. Technologies for Homeland Security (HST), 2013 IEEE International Conference on. IEEE; 2013. Unmanned aerial vehicle smart device ground control station cyber security threat model; pp. 722–728.

20. Jones A., Kovacich G.L. Auerbach Publications; 2015. Global Information Warfare: The New Digital Battlefield.

21. Carr E.B. Unmanned aerial vehicles: examining the safety, security, privacy

and regulatory issues of integration into us airspace. Natl. Centre Policy Anal. (NCPA).Retriev. September. 2013;23:2014.

22. Kerns A.J., Shepard D.P., Bhatti J.A., Humphreys T.E. Unmanned aircraft capture and control via gps spoofing. J. Field Rob. 2014;31(4):617–636.

23. Kovar D. 2016. Uavs, Iot, and Cybersecurity.

24. Mitchell R., Chen R. Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications. 2014;44(5):593–604.

25. Mitchell R., Chen R. Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications. IEEE Trans. Syst. Man Cybernet. 2013;44(5):593–604.

26. Kacem T., Wijesekera D., Costa P., Barreto A. Trustcom/BigDataSE/ISPA, 2016 IEEE. IEEE; 2016. An ads-b intrusion detection system; pp. 544–551.

27. Casals S.G., Owezarski P., Descargues G. Digital Avionics Systems Conference (DASC), 2013 IEEE/AIAA 32nd. IEEE; 2013. Generic and autonomous system for airborne networks cyber-threat detection; pp. 4A4–1.

28. Rani C., Modares H., Sriram R., Mikulski D., Lewis F.L. Security of unmanned aerial vehicle systems against cyber-physical attacks. J. Defense Model. Simul. 2016;13(3):331–342.

29. Lu H., Li Y., Mu S., Wang D., Kim H., Serikawa S. Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. IEEE Internet Things J. 2017;5(4):2315–2322.

30. Condomines J.-P., Zhang R., Larrieu N. Network intrusion detection system for uav ad-hoc communication: from methodology design to real test validation. Ad Hoc Netw. 2019;90:101759.

31. Sedjelmaci H., Senouci S.M., Ansari N. A hierarchical detection and response system to enhance security against lethal cyber-attacks in uav networks. IEEE Trans. Syst. Man Cybernet. 2017;48(9):1594–1606.

32. Lauf A.P., Peters R.A., Robinson W.H. A distributed intrusion detection system for resource-constrained devices in ad-hoc networks. Ad Hoc Netw. 2010;8(3):253–266.

33. Mitchell R., Chen I.-R. Proceedings of the first ACM MobiHoc workshop on Airborne Networks and Communications. ACM; 2012. Specification based intrusion detection for unmanned aircraft systems; pp. 31–36.

34. Zhang G., Wu Q., Cui M., Zhang R. GLOBECOM 2017-2017 IEEE Global Communications Conference. IEEE; 2017. Securing uav communications via trajectory optimization; pp. 1–6.

35. Cui M., Zhang G., Wu Q., Ng D.W.K. Robust trajectory and transmit power design for secure uav communications. IEEE Trans. Veh. Technol. 2018;67(9):9042– 9046.

36. Sharma D., Rashid A., Gupta S., Gupta S.K. A functional encryption technique in uav integrated hetnet: a proposed model. Int. J. Simul.–Syst. Sci. Technol. 2019;20

37. Clark D.R., Meffert C., Baggili I., Breitinger F. Drop (drone open source parser) your drone: forensic analysis of the dji phantom iii. Digital Invest. 2017;22:S3–S14.

38. Ganti S.R., Kim Y. Unmanned Aircraft Systems (ICUAS), 2016 International Conference on. IEEE; 2016. Implementation of detection and tracking mechanism for small uas; pp. 1254–1260.

39. Kosmowski K., Matyszkiel R., “Verification of the criterion and measures of interferences used in radio planning systems,” in Proc. SPIE 11055, XII Conference on Reconnaissance and Electronic Warfare Systems, 110550J (2019).

40. Sliwa J., Matyszkiel R., Jach J., “Efficient Methods of Radio Channel Access Using Dynamic Spectrum Access That Influences SOA Services Realization - Experimental Results,” in 2015 IEEE 81st Vehicular Technology Conference (VTCSpring).

41. Болховская О. В. Характеристики обнаружения пространственных сигналов для статистик обобщенного отношения правдоподобия в случае