

Universitatea “Politehnica” din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

***Comunicație wireless între o dronă
terestră și un UAV***

Proiect de diplomă

prezentat ca cerință parțială pentru obținerea titlului de *Inginer*
în domeniul *Electronică, Telecomunicații și Tehnologia Informației*,
programul de studii *Electronică Aplicată*

Profesor coordonator

Prof. Dr. Ing. Corneliu BURILEANU

Student

Cătălina Daiana TOMESCU

2017

Aprobat Director de Departament:
Prof. Dr. Ing. Sever PASCA



TEMA PROIECTULUI DE DIPLOMĂ
a studentei Tomescu Al. Cătălina Daiana, grupa 443B

1. Titlul temei: **Comunicație wireless între o dronă terestră și un UAV**
2. Contribuția practică, originală a studentului va consta în (în afara părții de documentare):

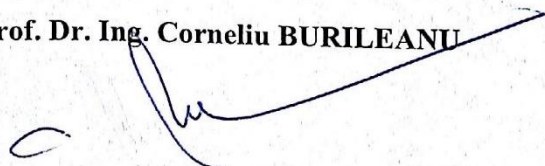
Proiectarea și implementarea practică a unei comunicații wireless între o stație de control terestră și un UAV. Prin această comunicație wireless stația de control poate transmite dronei UAV semnale pentru controlul traseului și a duratei zborului dronei, locul de aterizare al acesteia precum și waypointurile în care UAV-ul să captureze imagini. La fiecare semnal de control primit de la stația terestră UAV-ul trebuie să transmită înapoi stației terestre un semnal de Acknowledgement. De asemenea, imaginile capturate de UAV vor fi transmise stației terestre căreia îi va fi atașat un display pe care să poată fi vizualizate imaginile respective.

De asemenea, se vor face teste, se va analiza și se vor trage concluzii cu privire la influența distanței și a obstacolelor dintre stația terestră și UAV asupra comunicației wireless (delay în comunicare, pachete pierdute).

3. Proiectul se bazează pe cunoștințe dobândite în principal la următoarele 3-4 discipline: *Microcontrolere, Arhitectura Microprocesoarelor, Arhitectura Sistemelor de Calcul*
4. Proprietatea intelectuală asupra proiectului aparține: *UPB, Tomescu Cătălina Daiana*
5. Locul de desfășurare a activității: *UPB, Centrul de cercetări CAMPUS*
6. Realizarea practică rămâne în proprietatea: *UPB, Centrul de cercetări CAMPUS*
7. Data eliberării temei: *14.11.2016*

CONDUCĂTOR LUCRARE:

Prof. Dr. Ing. Corneliu BURILEANU



STUDENT:

Cătălina Daiana TOMESCU



Declarație de onestitate academică

Prin prezenta declar că lucrarea cu titlul "Comunicație wireless între o dronă terestră și un UAV", prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității "Politehnica" din București ca cerință parțială pentru obținerea titlului de Inginer în domeniul Inginerie Electronică și Telecomunicații, programul de studii Electronică Aplicată, este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referințe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referință la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referință la sursă. Înțeleg că plagiatul constituie infracțiune și se sancționează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurărilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sancționează conform regulamentelor în vigoare.

București, 20.06.2017

Cătălina Daiana TOMESCU



Cuprins

Introducere	15
Motivația lucrării.....	15
Aplicații ale UAV-urilor	15
Obiectivele lucrării	16
Capitolul 1. Descriere generală.....	17
1.1. Funcționalitate	17
1.2. Arhitectura generală – modelul client-server	18
Capitolul 2. Realizarea comunicației din punct de vedere al rețelei	21
2.1. Modelul TCP/IP	21
2.2. Protocolul TCP	24
2.3. Concepte Wireless	26
2.4. Asigurarea securității informației	31
Capitolul 3. Lehmann Aviation LA300 UAV	35
3.1. Descrierea dronei	35
3.2. Interfata wireless SN8200	36
3.3. Transmiterea setărilor către controller	41
Capitolul 4. Tehnologia software utilizată pe stația de control terestră	45
4.1. Prezentarea aplicației	45
4.2. Arhitectura aplicației	46
4.3. Implementare.....	50
Capitolul 5. Testarea comunicației	53
5.1. Interferențe	53
5.2. Influența distanței	53
5.3. Influența obstacolelor	54
Concluzii	57
Bibliografie	59
Anexa 1.....	61

Lista de figuri

Figura 1.1 – Arhitectura client – server

Figura 2.1 [8] - Modelul TCP/IP

Figura 2.2 [1] – TCP Three-Way-Handshake

Figura 2.3 – Parametri de asociere wireless

Figura 3.1 [2] – Componentele electronice ale UAV-ului LA300

Figura 3.2 [3] – Diagrama Bloc a Modulului SN8200

Figura 3.3 [4] – Simple Network Interface Controller

Figura 3.4 [5] – Framework EZ Web Wizzard pentru servicii web simple

Figura 3.5 – Setările transmise UAV-ului reprezentate vizual

Figura 4.1 – Inițiere conexiune cu UAV-ul

Figura 4.2 – Citire setări din fișier

Figura 4.3 – Transmitere setări către UAV

Figura 4.4 – Închidere conexiune cu UAV-ul

Figura 4.5 – Captură Wireshark comunicație wireless

Lista de tabele

Tabelul 3.1 [3] – Specificațiile electrice DC ale modulului SN8200

Tabelul 3.2 [3] – Caracteristicile RF pentru standardul IEEE802.11g ale SN8200

Tabelul 4.1 – Influența distanței în realizarea comunicației wireless

Lista de acrônimo

ADC = Analog to Digital Converter

AES = Advanced Encryption Standard

AP = Access Point

API = Application Programming Interface

DAC = Digital to Analog Converter

DHCP = Dynamic Host Configuration Protocol

DNS = Domain Name System

DoS = Denial of Service

EC = European Compliant

EHF = Extremely High Frequency

GPIO = General-Purpose Input/Output

HTTP = Hypertext Transfer Protocol

IC = Integrated Circuit

IEEE = Institute of Electrical and Electronics Engineers

IMAP = Internet Message Access Protocol

JSON = JavaScript Object Notation

MAC = Medium Access Control

MSL = Moisture Sensitivity Level

OFDM = Orthogonal Frequency-Division Multiplexing

P2P = Peer-to-peer

POP = Post Office Protocol

PSK = Pre-Shared Key

RADIUS = Remote Authentication Dial-In User Service

RF = Radio Frequency

RoHS = Restriction of Hazardous Substances

SHF = Super High Frequency

SMTP = Simple Mail Transfer Protocol

SNIC = Simple Network Interface Controller

SPI = Serial Peripheral Interface

SSID = Service Set Identifier

SWS = Simple Web Services

TCP = Transmission Control Protocol

TKIP = Temporal Key Integrity Protocol

UART = Universal Asynchronous Receiver/Transmitter

UAV = Unmanned Air Vehicle

UDP = User Datagram Protocol

UHF = Ultra High Frequency

UWP = Universal Windows Platform

WEP = Wired Equivalent Privacy

Wi-Fi = Wireless Fidelity

WiMAX = Worldwide Interoperability for Microwave Access

WLAN = Wireless Local Area Network

WPA = Wi-Fi Protected Access

WPAN = Wireless Personal-Area Networks

WWAN = Wireless Wide-Area Network

XAML = Extensible Application Markup Language

INTRODUCERE

Motivația lucrării

Un UAV (Unmanned Air Vehicle) reprezintă o aeronavă care, deși nu are pilot sau pasageri la bord, poate zbura pentru perioade lungi de timp la o altitudine și cu o viteză controlate, oferind astfel roluri importante în diverse arii de interes ale aviației civile și militare. De cele mai multe ori un UAV este o dronă automatizată sau care poate fi controlată de la distanță în timp real.

Primele prototipuri de acest tip au fost construite în timpul Primului Război Mondial de către armată în scopuri militare. Acestea erau lansate cu ajutorul unei catapulte și erau controlate de la distanță prin unde radio. Primul Război Mondial s-a terminat fără ca aceste prototipuri să fie efectiv folosite, însă în perioada interbelică dezvoltarea și testarea acestora a continuat astfel încât aceste aeronave au început să fie folosite în Războiul din Vietnam în scopuri de recunoaștere militară, ulterior având și alte roluri precum bombardarea unor ținte fixe.

Dezvoltarea dronelor și UAV-urilor a devenit de atunci chiar mai sofisticată și s-au făcut îmbunătățiri semnificative din punct de vedere al anduranței, altitudinii la care acestea pot ajunge și scopurilor în care sunt folosite astfel încât în ultimul timp a luat o foarte mare amploare folosirea UAV-urilor în multe arii de interes civile, de la fotografie la monitorizarea schimbărilor climatice.

Cu toate acestea, scopurile militare în care acestea erau inițial folosite nu au dispărut. Chiar dimpotrivă, utilizarea UAV-urilor s-a dezvoltat mult și în acest sens. Din acest motiv, precum și din cauza faptului că o comunicație wireless în scop personal cu un UAV poate produce interferențe și perturbă spațiul aerian, cel puțin în România este nevoie de o autorizație specială de zbor pentru a putea lansa drona de la sol.

Acestea fiind spuse, lucrarea de față este motivată de actualitatea și varietatea de aplicații și domenii în care UAV-urile pot fi folosite în societatea modernă în scopuri comerciale, aplicații pe care le voi descrie în detaliu în continuare.

Aplicații ale UAV-urilor

UAV-urile pot fi folosite în scopuri diverse, dintre acestea cele mai cunoscute fiind:

- Fotografie și înregistrări video;
- Geomapping;
- Agricultură;
- Networking;
- Monitorizarea schimbărilor climatice;

- Construcții etc.

Probabil cel mai cunoscut domeniu de către populație în care dronele și UAV-urile sunt folosite în societatea modernă este cel al fotografiei și înregistrărilor video. Tot mai multe evenimente de diverse tipuri sunt fotografiate și filmate cu ajutorul unui UAV căruia îi este atașată o cameră (GoPro sau alte tipuri) .

De asemenea, prin fotografierea unei zone și apoi corelarea acestor fotografii cu coordonatele geografice prin intermediul GPS-ului, pot fi întocmite hărți de mare precizie și acuratețe a unor zone care pot fi inaccesibile în alte moduri.

Atașând unui UAV senzori de presiune, umiditate, temperatură etc. și monitorizând o zonă pentru mai mult timp, în urma analizei datelor colectate se pot determina schimbări climatice pe termen lung. În același mod dronele pot fi folosite în scopuri meteorologice de scurtă durată.

În networking UAV-urile pot fi folosite ca gateway-uri între diverse rețele mai mici sau pot substitui pe termen scurt un echipament de telecomunicații care a cedat, evitând astfel costuri ce pot apărea din cauza inactivității echipamentului respectiv. Spre exemplu, în cazul unui furnizor de servicii de telefonie căruia îi pică un echipament din cauza furtunilor într-o zonă mai greu accesibilă (munte), dacă în zona respectivă nu există redunanță, clienții vor rămâne fără serviciul respectiv și astfel furnizorul poate suferi pierderi financiare și de reputație semnificative. Temporar, scenariul descris mai sus ar putea fi evitat folosind un UAV care să redirecționeze traficul către un alt echipament.

În industria construcțiilor, cu ajutorul unui UAV se poate monitoriza în mod permanent dacă acea construcție se desfășoară după arhitectura, măsurătorile și cu materialele pentru care au fost obținute autorizațiile de construcție respective.

Obiectivele lucrării

În această lucrare mi-am propus realizarea controlului de la distanță al unui UAV prin intermediul unei comunicații wireless cu o stație de control terestră. În acest mod dronei îi sunt specificate punctele de decolare și aterizare, punctele traseului pe care trebuie să îl urmeze precum și cele în care să captureze imagini.

Aceste puncte sunt transmise sub formă de coordonate geografice, latitudine și longitudine. Din punct de vedere practic mi-am propus ca pe stația de control terestră (laptop, telefon) să creez o aplicație software cu ajutorul căreia să trimit setările cu coordonatele geografice dronei.

De asemenea, mi-am propus în această lucrare să fac teste asupra influenței interferențelor, distanței și obstacolelor dintre drona și stația de control asupra comunicației wireless.

CAPITOLUL 1.

DESCRIERE GENERALĂ

1.1. Funcționalitate

Comunicația wireless dintre stația de control terestră și UAV-ul LA300 produs de compania Lehmann Aviation a fost realizată pe modelul unui sistem de tip client – server în care controller-ul wireless de la bordul UAV-ului este serverul, iar clientul este reprezentat de o aplicație software ce rulează pe stația de control terestră. UAV-ul LA300 dispune de două controllere: o interfață prin care poate comunica wireless și un controller care acționează motoarele dronei. Stația de control terestră este cea care inițiază comunicația și transmite interfeței wireless setările dorite, setări care sunt parsate, interpretate și transmise mai departe pe porturile de ieșire ale interfeței wireless către controllerul care acționează motoarele UAV-ului.

Modul în care sunt transmise dronei setările dorite se face prin intermediul unui obiect JSON în care fiecare nod reprezintă o setare. Un exemplu al formatului acceptat de UAV-ul LA300 este prezentat în Anexa 1. Printre setările care pot fi transmise și interpretate de dronă putem număra:

- Punctele de decolare și aterizare ale UAV-ului;
- Traseul pe care UAV-ul trebuie să îl urmeze, determinat ca o mulțime finită de waypointuri;
- Punctele traseului în care UAV-ul va captura imagini;
- Altitudinea fiecărui punct al traseului;
- Modul în care se va face aterizarea.

Toate punctele traseului enunțate mai sus sunt determinate prin coordonatele sale geografice – latitudine, respectiv longitudine. Având încorporat o antenă și funcționalitate GPS, UAV-ul poate localiza punctele specificate.

Din punct de vedere al succesiunii evenimentelor în timp, comunicația wireless este realizată în următorii pași:

1. Stația de control terestră solicită conectarea și asocierea cu AP-ul dronei;
2. Dacă credențialele de autentificare folosite de client sunt corecte, atunci AP-ul permite asocierea cu aplicația software ce rulează pe client;
3. Clientul inițiază o sesiune de comunicare;
4. Clientul transmite interfeței wireless a UAV-ului setările dorite;
5. Interfața wireless răspunde clientului ca a primit setările;
6. La sfârșitul transmisiei sesiunea de comunicare este închisă;
7. Interfața wireless parsează și interpretează setările primite de la client;
8. Setările sunt transmise pe porturile de ieșire ale modulului wireless și aplicate pe controllerul care acționează motoarele UAV-ului.

1.2. Arhitectura generală – modelul client-server

Realizarea comunicației wireless dintre UAV și stația de control terestră a fost făcută pe baza arhitecturii client-server, în care UAV-ul reprezintă serverul iar stația terestră este clientul. În Figura 1.1 fiecare sistem implicat în comunicația wireless are asociat rolul corespunzător arhitecturii client – server și este prezentat modul în care se face comunicația între cele două componente.

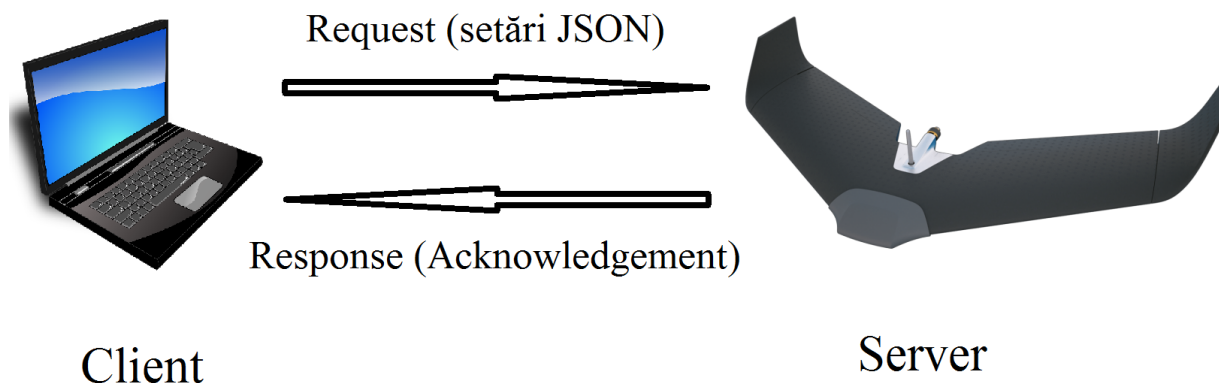


Figura 1.1 – Arhitectura client – server

Modelul client-server reprezintă o structură de sistem distribuit ce partiționează taskurile sau volumul de muncă între furnizori de servicii sau resurse numiți servere și solicitanții de servicii numiți clienți. De obicei clienții și serverele comunică prin intermediul unei rețele de calculatoare pe sisteme hardware diferite, însă atât serverul cât și clientul ar putea rula pe același sistem. Deoarece clientul solicită serviciile furnizate de către server, el este cel care inițiază sesiunea de comunicare cu serverul care așteaptă cererile primite. Printre exemplele de aplicații care folosesc modelul client-server se pot număra Email-ul sau World Wide Web.

Modelul de coordonare dintr-un sistem distribuit definește:

- Care dintre procese poate începe interacțiunea;
- Care dintre procese poate răspunde;
- Cum sunt administrate erorile ce pot apărea;
- Ambele aplicații implicate în comunicație nu pot aștepta fiecare ca un mesaj să sosească. Una dintre acestea trebuie să inițieze în mod activ interacțiunea în timp ce cealaltă așteaptă în mod pasiv.

Printre caracteristicile arhitecturii client-server putem număra scalabilitatea orizontală în care numărul de clienți poate crește și scalabilitatea verticală în care se poate migra ușor la un server mai puternic sau la o soluție multiserver.

O aplicație client sau server interacționează direct cu un protocol de nivel transport pentru a stabili sesiunea de comunicare și a trimite sau a primi informație. În cazul comunicației dintre stația terestră și dronă am folosit protocolul de nivel transport TCP deoarece, spre deosebire de UDP, este un protocol reliable. Urmează ca protocolul de nivel transport să folosească protocoale

de nivele mai joase pentru a trimite sau primi mesaje individuale. De aceea un calculator are nevoie de o întreagă stivă de protocoale pentru a rula o aplicație de tip client sau server.

TCP folosește valori întregi pe 16 biți pentru a identifica serviciile și asignează în mod unic un port fiecărui serviciu. De aceea, atunci când solicită un serviciu clientul specifică în request numărul portului serviciului pe care îl dorește.

În general un client software îndeplinește următoarele funcții:

- Este un program arbitrar care devine client temporar atunci când este nevoie de acces la distanță la diferite resurse, însă efectuează și alte operații de calcul local;
- Este invocat local de către utilizator și este executat doar pentru o singură sesiune de comunicare;
- Rulează local pe un calculator personal;
- Inițiază contactul cu serverul în mod activ;
- Poate accesa mai multe servicii dacă este nevoie, însă poate contacta în mod activ un singur server la un moment dat;
- Nu necesită hardware special sau un sistem de operare sofisticat.

Pe de altă parte, funcțiile îndeplinite de către server sunt următoarele:

- Este un program privilegiat cu scop special, dedicat furnizării unui singur serviciu, care poate însă suporta mai mulți clienți simultan;
- Rulează pe un computer distribuit;
- Așteaptă în mod pasiv contact de la clienți arbitrari;
- Acceptă contactul de la mai mulți clienți, însă oferă un singur serviciu;
- Necesită hardware cu resurse mari dedicate și un sistem de operare sofisticat;
- Trebuie să garanteze următoarele trei operații: autentificare – verificarea identității clientului; autorizare – verificarea posibilității unui client de a accesa un anumit serviciu; securitatea datelor – garanția că date specifice nu pot fi citite și/sau modificate de către utilizatori neautorizați.

CAPITOLUL 2.

REALIZAREA COMUNICAȚIEI DIN PUNCT DE VEDERE AL REȚELEI

2.1. Modelul TCP/IP

Modelul de protocoale TCP/IP pentru comunicații în rețea a fost creat în prima jumătate a anilor 1970 pornind de la necesitatea definirii unei suite de protocoale care să asigure livrarea end-to-end a informației între două calculatoare prin Internet. Acest model definește patru funcții ce trebuie îndeplinite astfel încât o comunicație să fie realizată cu succes, după cum este prezentat în Figura 2.1.

Suita de protocoale TCP/IP este un standard disponibil în mod gratuit publicului, ceea ce se traduce prin faptul că orice vendor este autorizat să implementeze aceste protocoale în hardware-ul sau software-ul pe care îl produce.

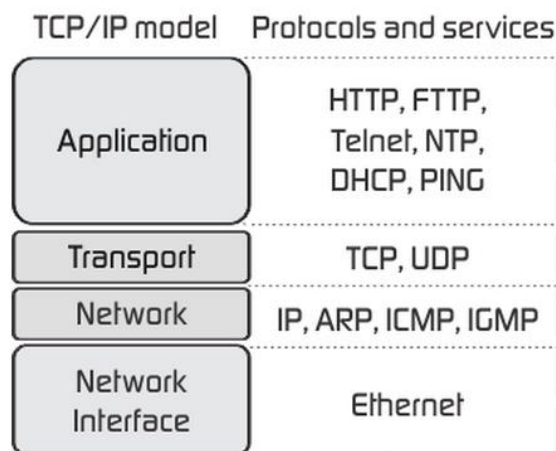


Figura 2.1 [8] - Modelul TCP/IP

2.1.1. Nivelul Aplicație

Nivelul Aplicație este nivelul de top al modelului TCP/IP și este cel mai apropiat de utilizator. Acest nivel include un set de protocoale ce îndeplinesc funcționalități specifice pentru o varietate de aplicații. Dintre acestea putem aminti:

- DNS – rezolvă numele folosite în Internet în adrese IP;
- Telnet – oferă acces de la distanță către servere și dispozitive folosite în rețelistică, precum rutere sau switchuri;
- SMTP – transferă mesajele și atașamentele mail-urilor;
- DHCP – protocol folosit pentru a asigna o adresă IP, masca de rețea, default gateway și adresa serverului DNS unui host în mod dinamic;

- HTTP – transferă fișiere ce aparțin unei pagini web;
- POP – protocol folosit de către clienții de email pentru a descărca emailurile de la un server;
- IMAP – un alt protocol pentru descărcarea emailurilor.

2.1.2. Nivelul Transport

Nivelul Transport este responsabil pentru stabilirea unei sesiuni de comunicare temporară între două aplicații și transmiterea datelor între ele. O aplicație ce rulează pe un host sursă generează date care sunt transmise către o altă aplicație ce aparține unui host destinație, fără a ține cont de tipul hostului destinație, tipul mediului pe care vor circula datele, calea pe care va circula informația sau dimensiunea rețelei. Nivelul Transport este cel care se asigură că datele sunt segmentate și pot fi reconstituite în mod corect la destinație. Astfel, principalele sale funcții sunt:

- Monitorizarea conexiunii între aplicațiile care rulează pe dispozitivele sursă, respectiv destinație;
- Segmentează datele primite de la nivelul Aplicație și reasamblează la destinație segmentele primite în stream-uri de date care sunt transmise mai departe nivelului Aplicație;
- Identifică aplicația căruia îi este destinat fiecare stream de date.

În modelul TCP/IP, procesele de segmentare și reasamblare de la nivelul Transport pot fi îndeplinite de două protocoale: TCP și UDP. Dintre acestea, pentru a implementa comunicația wireless între client și UAV eu am ales protocolul TCP din motivele prezentate în subcapitolul 2.2.

2.1.3. Nivelul Internet

Nivelul Internet furnizează patru servicii prin intermediul cărora să ofere dispozitivelor o comunicare eficientă end-to-end:

- Adresarea dispozitivelor: Un dispozitiv trebuie să fie configurat cu o adresă IP unică pentru a putea fi identificat și adresat într-o rețea. Putem numi host un dispozitiv din cadrul unei rețele ce este configurat cu o adresă IP;
- Încapsularea: Nivelul Internet primește o unitate de date numită segment de la nivelul Transport. Printr-un proces numit încapsulare, la acest nivel este adăugat un antet cu informații despre IP-ul sursă și destinație, formând astfel un pachet;
- Rutarea: Nivelul Internet furnizează serviciile necesare pentru a direcția pachetele către un host destinație aflat într-o altă rețea. Pentru a ajunge către alte rețele un pachet trebuie să fie procesat și redirecționat de către un ruter. Un pachet poate traversa mai multe astfel de

dispozitive intermediare în drumul său către destinație. Fiecare rută intermediară pe care un pachet este nevoit să o traverseze pentru a ajunge la host-ul destinație este numită hop;

- Decapsularea: Atunci când un pachet ajunge la nivelul Internet al host-ului destinație este verificat antetul pachetului respectiv. Dacă IP-ul conținut în acest antet este același cu propriul său IP, antetul este îndepărtat de pachet printr-un proces numit decapsulare. Unitatea de date astfel obținută este trimisă mai departe către nivelul Transport.

Unul dintre serviciile implementate de către suita de protocoale TCP/IP de la nivelul Internet este numit IP. Acesta furnizează doar funcțiile necesare pentru a transmite un pachet de la sursă la destinație într-un sistem interconectat de rețele. Protocolul nu a fost proiectat pentru a urmări și controla flow-ul de pachete, aceste funcții fiind furnizate de alte protocoale de la alte nivele. Principalele caracteristici ale protocolului IP sunt:

- Connectionless: Nu este stabilită o conexiune cu destinația înainte ca pachetele să fie trimise.
- Best effort (nesigur): Sosirea pachetelor la destinație nu este garantată.
- Independent de mediul pe care sunt transmise datele.

2.1.4. Nivelul Network Access

Nivelul Network Access este responsabil pentru transmiterea cadrelor între nodurile unei rețele printr-un mediu fizic (cu sau fără fir). Acest nivel acceptă pachetele primite de la nivelul superior, le împachetează în unități de date numite cadre pe care apoi le codează ca fiind o serie de biți ce este transmisă pe mediul fizic sub forma unor semnale.

Codarea reprezintă o metodă de conversie a unui flux de date într-un grup de biți – 0 sau 1 - folosiți pentru a furniza un tipar predictibil ce poate fi recunoscut atât de emițător cât și de receptor. În plus, metodele de codare de la nivelul fizic pot furniza și coduri de control, ca de exemplu pentru a marca începutul și sfârșitul unui pachet/cadru. Există numeroase tipuri de codare. Dintre acestea printre cele mai des folosite în rețelistică se numără:

- Codarea Manchester: Bitul 0 este reprezentat printr-o tranziție de la un nivel înalt de tensiune la un nivel de joasă tensiune, iar bitul 1 este reprezentat printr-o tranziție de la un nivel de joasă tensiune la un nivel de înaltă tensiune.
- Non-Return to Zero: Această metodă de codare este caracterizată de două stări numite „zero” și „unu”. Un 0 poate fi reprezentat ca un nivel de tensiune, iar un 1 poate fi reprezentat ca un alt nivel de tensiune diferit. Uzual, pentru 0 se folosește nivelul de tensiune de 0V, iar pentru 1 nivelul de tensiune de 5V.

Nivelul Network Access trebuie să genereze semnalele electrice, optice sau radio prin care sunt transmise datele. Standardele de la acest nivel trebuie să definească ce tip de semnal reprezintă un 1 și ce tip reprezintă un 0. Semnalele pot fi transmise în două moduri:

- Asincron: Semnalele sunt transmise fără un semnal de ceas asociat. Spațierea temporară dintre două blocuri de date poate fi de durată arbitrară, nefiind asociată unui semnal de ceas. De aceea, cadrele necesită indicatori de start și stop.

- Sincron: Semnalelor de datele este atașat un semnal de ceas, biții fiind transmiși la intervale egale de timp.

De asemenea, nivelul Network Access controlează modul în care datele sunt transmise și recepționate de pe mediul de transmisiune și execută detecții de erori pentru a asigura integritatea cadrelor transmise.

2.2. Protocolul TCP

Nivelul Transport este responsabil din punct de vedere al siguranței comunicației. La nivelul rețea, protocolul IP este preocupat doar de structura, adresarea și rutarea pachetelor. IP nu specifică în ce mod este efectuat transportul pachetelor, acesta fiind rolul nivelului Transport. Modelul TCP/IP ne pune la dispoziție două protocoale la nivelul Transport, TCP și UDP.

Am ales protocolul TCP pentru realizarea comunicației cu UAV-ul deoarece, spre deosebire de UDP, TCP este considerat un protocol fiabil și are toate caracteristicile nivelului Transport. TCP include procese care să ne asigure că informația transmisă între aplicația de pe stația de control terestră (laptop, telefon) și UAV ajunge la destinație, prin metode de livrare confirmată.

Folosind protocolul TCP ne asigurăm că sunt îndeplinite următoarele trei funcții definitorii unui protocol fiabil de nivel Transport:

- Urmărirea (trackingul) segmentelor de date transmise;
- Confirmarea (acknowledgementul) datelor primite;
- Retransmiterea oricăror date ce nu au fost confirmate ca fiind primite.

Protocolul TCP împarte un mesaj în unități de date denumite segmente. Segmentele sunt apoi numerotate în ordine și pasate mai departe către protocolul IP care la rândul său adaugă acestora antente cu adresele sursă și destinație, formând astfel pachete. TCP urmărește numărul de segmente transmise către un host specific de la o aplicație specifică. Dacă expeditorul nu primește înapoi un mesaj de confirmare într-o anumită perioadă de timp, acesta presupune că segmentul respectiv a fost pierdut și retrimite doar acel segment. Pe dispozitivul receptor, protocolul TCP este responsabil de reasamblarea segmentelor în ordine și transmiterea lor către nivelul Aplicație.

TCP a fost inițial descris în RFC 793 [6]. În adăuga suportării funcțiilor de bază de segmentare și reasamblare, TCP furnizează, de asemenea, și următoarele funcții:

- Conversații orientate pe conexiune (connection-oriented) prin stabilirea de sesiuni;
- Livrare reliable;
- Reconstrucție ordonată a segmentelor;

- Control al fluxului de date (flow control).

TCP este un protocol orientat pe conexiune. Un astfel de protocol negociază și stabilește o conexiune permanentă (sau sesiune) între sursă și destinație înainte de a trimite orice trafic. Prin stabilirea unei sesiuni dispozitivele negociază cantitatea de trafic ce poate fi transmisă la un moment de timp, iar datele ce sunt transmise între cele două dispozitive pot fi controlate. Sesiunea este terminată doar atunci când toate datele au fost transmise cu succes. Pentru a stabili o conexiune, fiecare dintre dispozitivele între care se realizează conexiunea trebuie să transmită un semnal de SYN (Synchronize) și să primească un mesaj de ACK (Acknowledgment) de la celălalt dispozitiv. De aceea, conceptual vorbind, trebuie să avem patru mesaje de control ce sunt schimbate între dispozitive. Totuși, este inefficientă transmiterea semnalelor SYN și ACK în mesaje separate când ar putea fi ambele transmise simultan. Acesta este motivul pentru care în succesiunea normală de evenimente din stabilirea conexiunii unul dintre semnalele SYN și unul dintre semnalele ACK sunt transmise împreună în același mesaj numit adesea SYN+ACK. Însurând, avem astfel un număr total de trei mesaje și de aceea acest proces poartă denumirea Three-Way-Handshake. Acest proces este prezentat conceptual în Figura 2.2 [7] de mai jos.

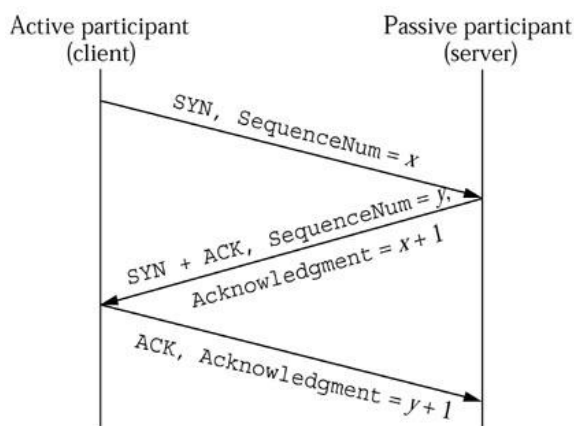


Figura 2.2 [1] – TCP Three-Way-Handshake

În termeni de rețelistică termenul de reliability înseamnă asigurarea că fiecare unitate de date transmisă de sursă ajunge la destinație. Din motive diverse este posibil ca o unitate de date să fie coruptă sau pierdută complet atunci când este transmisă prin rețea. TCP poate asigura că toate informațiile ajung la destinație prin retransmiterea de la sursă a unităților de date corupte sau pierdute.

Deoarece rețelele pot furniza mai multe rute pe care să circule pachetele, fiecare cu rate de transmisie diferite, datele pot ajunge la dispozitivul destinație în ordinea greșită. Prin numerotarea și secvențializarea segmentelor, protocolul TCP se asigură că segmentele sunt reasamblate la destinație în ordinea corectă.

Dispozitivele conectate la rețea au resurse limitate, precum memorie sau lățime de bandă. Atunci când TCP observă că resursele sunt suprasolicitate poate cere ca aplicația care transmite

date să reducă rata de transmisie, asigurând în acest mod controlul fluxului de date prin care pot fi prevenite pierderile de segmente pe rețea și evitarea retransmiterii segmentelor pierdute.

2.3. Concepte Wireless

Pentru a asigura flexibilitatea comunicării cu UAV-ul și controlul de la distanță al acestuia am ales ca mediul de transmisie de la nivelul Network Access al stivei TCP/IP să fie unul wireless.

Rețelele Wireless pot fi clasificate în [8]:

- Wireless Personal-Area Networks (WPAN) – Operează pe o distanță de ordinul metrilor. În acest tip de rețea sunt folosite dispozitive ce folosesc tehnologii precum Bluetooth sau Wi-Fi Direct pentru a comunica.
- Wireless LANs (WLANs) – Operează pe o distanță de câteva sute de metri, motiv pentru care am folosit acest tip de rețea în realizarea comunicației wireless cu UAV-ul.
- Wireless Wide-Area Networks (WWANs) – Operează pe distanțe mult mai mari, precum arii metropolitane.

Pentru conectarea dispozitivelor la aceste rețele există mai multe tipuri de tehnologii, printre care se numără:

- Bluetooth – Definit prin standardul WPAN IEEE 802.15, tehnologie ce folosește un proces de creare a perechilor de dispozitive pentru a comunica pe distanțe de până la 100 de metri.
- Wi-Fi – Definit prin standardul WLAN IEEE 802.11, folosit pentru transferul date, voce sau video pe distanțe de până la 300 de metri.
- WiMAX – Definit prin standardul WWAN IEEE 802.16, tehnologie folosită pentru comunicațiile pe distanțe mult mai mari, până la aproximativ 50 de km.
- Date mobile – 2G/3G/4G
- Comunicații prin satelit – Tehnologie ce furnizează accesul la rețea a zonelor izolate prin intermediul unui satelit geostaționar. Este de obicei o tehnologie mult mai scumpă și necesită o cale liberă, fără obstacole, între satelit și zona izolată care dorește acces la internet.

Comunicațiile wireless operează în domeniul undelor radio (cu frecvențe între 3Hz și 300GHz) ale spectrului electromagnetic. Domeniul undelor radio este subdivizat la rândul său în secțiunea frecvențelor radio și secțiunea microundelor. Tehnologiile Bluetooth, WLANs, datele mobile și comunicațiile prin satelit operează toate în secțiunea microundelor caracterizată de benzi frecvențe UHF, SHF și EHF.

Pentru a se putea conecta la rețeaua wireless, dispozitivele au incorporate sau atașate transmițătoare și receptoare reglate la o frecvență specifică a domeniului undelor radio. Concret, următoarele benzi de frecvențe sunt alocate rețelelor wireless descrise prin standardul 802.11:

- 2.4 GHz (UHF) – 802.11 b/g/n/ad

- 5 GHz (SHF) – 802.11 a/n/ac/ad
- 60 GHz (EHF) – 802.11 ad

Standardul WLAN IEEE 802.11 [9] definește cum sunt folosite benzile de frecvențe radio la nivelul Network Access al modelului TCP/IP. Variații ale acestui standard au fost dezvoltate de-a lungul timpului, printre care putem număra:

- 802.11 – Emis în anul 1997 și acum scos din uz, acest standard este versiunea originală a specificațiilor WLAN care opera la frecvența de 2.4 GHz și oferea viteze de până la 2 Mb/s. La momentul emiterii acestui standard rețelele LAN cu fir ofereau viteze de aproximativ 10 Mb/s, motiv pentru care această nouă tehnologie wireless nu a fost adoptată cu entuziasm.

- IEEE 802.11a – Emis în anul 1999, acest standard opera în banda de frecvențe de 5GHz, bandă mai puțin aglomerată decât cea de 2.4 GHz, și oferea viteze de până la 54 Mb/s. Deoarece acest standard operează la frecvențe mai înalte, are o arie de acoperire mai mică și este mai puțin eficient în străpungerea structurilor unei clădiri. Dispozitivele ce operează folosind acest standard nu sunt compatibile cu cele care folosesc standardele 802.11b și 802.11g.

- IEEE 802.11b – Emis în anul 1999, acest standard operează în banda de frecvențe de 2.4 GHz și oferă viteze de aproximativ 11 Mb/s. Dispozitivele ce folosesc acest standard au o arie de acoperire mai mare și sunt mai eficiente în folosirea lor în clădiri datorită faptului că structurile clădirilor nu atenuează așa tare semnalul.

- IEEE 802.11g – Emis în anul 2003, acest standard operează în banda de frecvențe de 2.4 GHz și oferă viteze de până la 54 Mb/s. Astfel, dispozitivele ce folosesc acest standard operează în aceeași bandă de frecvențe ca cele care folosesc standardul 802.11b, dar cu viteza celor care folosesc standardul 802.11a. Totuși, în momentul în care este suportat un client 802.11b, viteza de transmisie este redusă.

- IEEE 802.11n – Emis în anul 2009, acest standard operează atât în banda de transfer de 2.4 GHz, cât și în cea de 5 GHz. Vitezele de transfer variază de între 150 Mb/s și 600 Mb/s pe o distanță de până la 70 de metri. Acest standard este compatibil cu dispozitivele 802.11 a/b/g. Totuși, suportând un mediu mixt viteza de transfer este redusă.

- IEEE 802.11ac – Emis în anul 2013, operează în banda de frecvențe de 5GHz și furnizează viteze de transfer între 450 Mb/s și 1.3 Gb/s. Acest standard este compatibil cu versiunile 802.11 a/n.

- IEEE 802.11ad - Emis în anul 2014 și cunoscut sub numele WiGig, acest standard oferă o soluție ce operează în trei benzi de frecvențe: 2.4GHz, 5 GHz și 60 GHz și oferă viteze teoretice de până la 7 Gb/s. Este compatibil cu dispozitivele care folosesc toate celelalte versiuni ale standardului 802.11, descrise anterior.

În realizarea comunicației wireless între stația de control terestră și UAV am folosit standardul 802.11g deoarece reprezintă o soluție potrivită atât din punctul de vedere al vitezei de transfer, al distanței pe care poate opera, precum și al benzii de frecvențe în care funcționează.

Standardul IEEE 802.11 identifică două tipuri de topologii wireless:

- Ad-hoc – Modelul în care două dispozitive sunt conectate wireless între ele, fără nevoia suplimentară a unui echipament de infrastructură suplimentar, precum un router.

- Modelul de infrastructură – În acest model doi clienți wireless se conectează între ei prin intermediul unui echipament suplimentar.

Într-o rețea wireless de tip ad-hoc două dispozitive comunică între ele într-o manieră peer-to-peer (P2P) fără a folosi echipamente suplimentare, motiv pentru care am ales acest tip de topologie în realizarea comunicației între stația de control terestră și UAV. În vederea obținerii acestui tip de topologie am stabilit o legătură de tip client-server între cele două dispozitive, legătură în care UAV-ul reprezintă server-ul iar aplicația de pe stația de control terestră este clientul.

Pentru ca dispozitivele wireless să comunice între ele într-o rețea, acestea trebuie să se asocieze cu un AP sau cu un router wireless. O parte importantă a procesului 802.11 este reprezentată de descoperirea unui WLAN și apoi conectarea la acesta, process realizat în trei etape:

1. Descoperirea unui nou AP wireless;
2. Autentificarea la AP;
3. Asocierea cu AP-ul.

Pentru a se asocia, clientul wireless și AP-ul trebuie să stabilească de comun acord anumiți parametri care trebuie configurați pe AP și apoi pe client pentru a activa procesele enunțate mai sus:

- SSID – Un SSID reprezintă un identificator unic pe care clienții wireless îl folosesc pentru a distinge între mai multe rețele wireless în aceeași vecinătate. Numele SSID-ului apare în lista de rețele wireless disponibile pe client. Depinzând de configurarea rețelei, mai multe AP-uri pot avea același SSID care uzual este un nume ce cuprinde între 2 și 32 de caractere. În cazul aplicației mele SSID-ul AP-ului de pe UAV este LA300-41cc23. Astfel, pentru a conecta stația de control terestră la AP-ul de pe dronă trebuie să găsim acest nume în lista de SSID-uri disponibile de pe client.

- Parola – Deseori numită și cheie de securitate, parola este necesară clientului pentru a se autentifica la AP. Aceasta previne atacurile de securitate prin prevenirea accesului la rețeaua wireless a utilizatorilor nedorți. În cazul prezentei lucrări, am setat această parolă implicit “password”.

- Modul de rețea – Se referă la standardele WLAN 802.11 a/b/g/n/ac/ad. AP-urile și ruterele pot opera într-un mod mixt în care pot fi folosite mai multe standarde simultan. În realizarea comunicației wireless dintre client și dronă am folosit modul de rețea 802.11g.

- Modul de securitate – Se referă la setările parametrilor de securitate, precum WEP, WPA sau WPA2. Toate ruterele wireless și AP-urile trebuie securizate la cel mai înalt nivel. Tipurile None și WEP ar trebui evitate sau utilizate doar în cazul în care securitatea rețelei nu

este de interes. Pentru a oferi comunicației wireless cu drona securitatea necesară, am ales modul de securitate WPA-Personal.

- Setările de canal – Fac referire la banda de frecvență folosită pentru a transmite date wireless. Ruterele wireless si AP-urile pot alege setările de canal în mod dinamic sau acestea pot fi setate manual pentru a evita interferențele cu un alt AP sau router wireless. Cum implicit standardul 802.11g operează în banda de frecvențe 2.4 GHz, alegând acest standard și mod de rețea, implicit banda de frecvențe în care este realizată comunicația wireless dintre client și UAV este de 2.4 GHz.

Properties

SSID:	LA300-41cc23
Protocol:	802.11g
Security type:	WPA-Personal
Network band:	2.4 GHz
Network channel:	3
IPv4 address:	192.168.1.44
Manufacturer:	Intel Corporation
Description:	Intel(R) Dual Band Wireless-AC 8260 #2
Driver version:	18.40.0.12
Physical address (MAC):	A4-34-D9-5E-70-B8

Copy

Figura 2.3 - Parametri de asociere wireless

În Figura 2.3 sunt prezentați parametrii de asociere wireless așa cum apar pe client înainte de asocierea cu AP-ul de pe UAV.

Clienții wireless se conectează la AP folosind un proces de scanare care poate fi:

- Pasiv – AP-ul anunță public serviciul său prin trimiterea periodică a unor cadre broadcast ce conțin SSID-ul, standardele suportate și setările de securitate. Principalul scop al acestor cadre este acela de a permite clienților wireless să învețe ce rețele și AP-uri sunt disponibile într-o arie dată și astfel să aleagă ce rețea sau AP să folosească.

- Activ – Clienții wireless trebuie să știe numele SSID-ului. Clientul wireless inițiază procesul prin trimiterea broadcast a unui cadru cerere pe mai multe canale, cadru ce conține numele SSID-ului și standardele suportate. Modul activ poate fi necesar dacă un AP sau ruter wireless este configurat să nu trimită cadre broadcast prin care să se facă public.

În realizarea comunicației dintre UAV și stația de control terestră am folosit modul pasiv de scanare a rețelelor wireless disponibile, mod în care periodic AP-ul - drona -trimite cadre

broadcast în care anunță existența sa. Astfel, pe client va apărea în lista de SSID-uri disponibile și SSID-ul UAV-ului fără a mai fi necesară o cerere suplimentară din partea clientului – stația de control terestră.

După ce SSID-ul dronei îi este disponibil, pentru a se conecta la AP-ul UAV-ului clientul trebuie să se autentifice. Standardul IEEE 802.11 a fost inițial dezvoltat cu două tipuri de autentificare:

- Autentificare deschisă – Fundamental o autentificare de tip NULL în care clientul wireless spune AP-ului “autentifică-mă” iar AP-ul răspunde cu “da” fără a fi nevoie de o parolă. Acest tip de autentificare furnizează conectivitate wireless oricărui dispozitiv dorește să se conecteze și ar trebui folosit doar în situații în care securitatea datelor nu este de interes.
- Autentificare cu cheie – Tehnică bazată pe o cheie care este pre-distribuită între client și AP.

Din motive de securitate IT, pentru a nu lăsa utilizatori nedorți să se conecteze la AP-ul UAV-ului, evitând astfel incidente precum furtul de informație – traseul dronei, imagini sau video sau chiar schimbarea traseului dronei, am ales modul de autentificare cu cheie pre-distribuită, în care cheia de acces la AP este parola “password”.

Procesul de autentificare este următorul:

1. Clientul wireless – stația de control terestră – trimite un cadru de autentificare către AP-ul dronei.
2. AP-ul răspunde cu un text provocare clientului.
3. Clientul criptează textul primit de la AP folosind cheia pre-distribuită și trimite AP-ului textul criptat.
4. AP-ul decriptează textul criptat folosind aceeași cheie.
5. Dacă textul decriptat primit de la client corespunde textului provocare trimis de AP, atunci AP-ul autentifică clientul. Dacă cele două mesaje nu corespund, atunci clientul nu este autentificat și accesul wireless este blocat.

După ce clientul este autentificat, AP-ul dronei continuă procesul de asociere, la sfârșitul căruia setările sunt finalizate iar legătura de date este stabilită între stația de control terestră și UAV. Etapa de asociere constă în următorii pași:

1. Clientul wireless trimite un cadru cerere de asociere ce include adresa sa MAC.
2. AP – ul trimite un răspuns de asociere ce include BSSID-ul AP-ului – adresa MAC a AP-ului dronei.
3. AP-ul mapează clientului un port logic cunoscut ca identificator de asociere. Identificatorul de asociere este echivalent unui port pe un switch și permite urmărirea cadrelor destinate clientului, precum și transmiterea acestora.

La finalul acestor procese, după ce clientul a fost asociat cu AP-ul, informația poate fi transmisă între stația de control terestră și UAV.

2.4. Asigurarea securității informației

Securitatea informației a fost mereu o preocupare mai mare în cazul Wi-Fi deoarece limita rețelei a fost mutată. Este nevoie ca problemele de securitate ce pot apărea în cazul comunicației cu o dronă – furt de informație, schimbarea traseului acesteia, denial of service etc. – să fie adresate.

Pentru a înțelege mai bine riscurile existente din punct de vedere al securității datelor, pot fi enumerate următoarele tipuri de atacuri informatice aplicabile comunicației wireless dintre stația de control terestră și UAV:

- **Recunoașterea** - Acesta este procesul de descoperire folosit pentru a găsi informații despre rețea și a determina potențiale vulnerabilități și reprezintă de obicei primul pas într-un atac de securitate. Acest proces poate include scanări ale rețelei pentru a afla ce adrese IP răspund, precum și scanări ulterioare pentru a vedea ce porturi sunt deschise.

- **Escalarea Privilegiului** - Acesta este un proces de obținere a unui nivel de acces autorizat sau nu în urma căruia poate fi obținut un nivel de acces și mai mare. Un exemplu ar fi acela în care atacatorul obține acces la rețeaua dintre stația de control terestră și UAV în urma căruia poate escala accesul la toate informațiile deținute de stația de control terestră. În cazul în care această stație controlează mai multe drone, atacatorul poate obține acces și controla toate celelalte drone. O rețea este pe atât de sigură pe cât este cel mai slab dispozitiv conectat la aceasta.

- **Man-in-the-Middle** – Este acel tip de atac în care atacatorul se poziționează pe linia dintre două dispozitive ce comunică între ele cu intenția de a performa recunoaștere sau a manipula datele ce sunt transmise. Principalul obiectiv este acela ca atacatorul să vadă tot traficul. În cazul comunicației cu UAV-ul un exemplu ar putea fi acela că datele transmise de UAV către stația de control – imagini/video – să nu fie cele reale, capturate de dronă, ci falsificate și să nu reflecte realitatea. Un alt exemplu ar fi acela în care traseul transmis UAV-ului ar fi manipulat, pierzându-se astfel complet controlul asupra dronei.

- **DoS și DDoS** – Denial of Service și Distributed Denial of Service, în funcție de câte dispozitive sunt implicate în atac, unul sau mai multe. Ambele tipuri de atacuri urmăresc același scop, iar în cazul comunicației dintre stația terestră și drona aceste tipuri de atac se pot traduce în pierderea completă a controlului asupra dronei.

Din punct de vedere securitate IT există următoarele trei principale obiective [10]:

- **Confidențialitate** – Există două tipuri de date: date în mișcare care traversează rețeaua și date în stare de repaus ce sunt stocate local, pe servere, în cloud etc. Confidențialitatea se referă la faptul că doar indivizii sau sistemele autorizate pot avea acces la informații sensibile sau confidențiale. Acest lucru presupune ca niciun alt individ neautorizat nu ar trebui să aibă niciun tip de acces la aceste date. În ceea ce privește datele aflate în mișcare care traversează rețeaua, principala metodă de a le proteja este criptarea tuturor datelor înainte de trimiterea acestora în rețea. O altă opțiune adițională criptării ar fi aceea de a folosi o rețea separată pentru

transmiterea informațiilor condifiențiale. Acest obiectiv al securității datelor poate fi afectat de atacuri precum Recunoașterea sau Escalarea Privilegiului.

- Integritate – Se referă la faptul că doar indivizii sau sistemele autorizate pot face modificări asupra datelor. Coruperea datelor reprezintă o eșuare a menținerii integrității datelor. Acest principiu poate fi încălcat de atacuri precum Man-in-the-Middle.

- Disponibilitate – Acest concept se aplică atât sistemelor cât și datelor. Dacă rețeaua sau datele nu sunt disponibile utilizatorilor sau sistemelor autorizate atunci când este nevoie de ele – spre exemplu din cauza unui atac de tip DoS sau poate pur și simplu din cauza căderii rețelei – impactul poate fi semnificativ incluzând pierderea comunicației cu UAV-ul. Acest obiectiv poate fi complet pierdut în urma atacurilor de tip DoS sau DDoS.

Pentru a îndeplini obiectivul confidențialității în realizarea comunicației wireless, clientul – stația de control – trebuie să se autentifice la AP-ul UAV-ului folosind parola setată pe UAV, iar pentru îndeplinirea obiectivului integrității datelor, datele transmise între cele două dispozitive sunt criptate.

Standardul IEEE 802.11 introduce trei tehnici de autentificare folosind o cheie pre-distribuită:

- WEP – Specificație 802.11 originală, această tehnică a fost proiectată să furnizeze un nivel de acces asemănător conectării la o rețea folosind o conexiune cu fir. Datele sunt securizate folosind o metodă de criptare RC4 cu o cheie statică. Cum cheia nu se schimbă niciodată, această metodă nu este foarte sigură.

- WPA – Un standard Wi-Fi Alliance care folosește WEP, dar care securizează datele folosind un algoritm de criptare mult mai puternic – Temporal Key Integrity Protocol. TKIP schimbă cheia la fiecare pachet, ceea ce face algoritmul mai greu de spart.

- IEEE 802.11i/WPA2 – IEEE 802.11i este un standard industrial pentru securizarea rețelelor wireless. Versiunea Wi-Fi Alliance este numită WPA2. Ambele tehnici folosesc metoda de criptare AES.

WPA și WPA2 suportă două tipuri de autentificare:

- Personal – Pentru rețele destinate folosirii personale acasă, sau în birouri de dimensiuni mici, utilizatorii se autentifică folosind o cheie pre-distribuită – PSK. Nu este necesar un server de autentificare special.

- Enterprise – Pentru rețele de dimensiuni mari, în general în întreprinderi și birouri mari, însă este nevoie de un server suplimentar de autentificare numit RADIUS, ceea ce face acest tip de autentificare mai greu de implementat.

Standardele 802.11i și Wi-Fi Alliance WPA și WPA2 folosesc următoarele tipuri de protocoale:

- TKIP – Metoda de criptare folosită de WPA. Această tehnică de criptare folosește WEP, însă criptează datele folosind TKIP și are un câmp suplimentar pentru Message Integrity Check în pachetul criptat pentru a asigura faptul că mesajul nu a fost alterat.

- AES – Metoda de criptare folosită de WPA2. AES efectuează aceleași funcții ca și TKIP, însă folosește un protocol suplimentar prin care dispozitivul destinație poate recunoaște dacă biții criptați sau necriptați au fost alterați.

Metoda de autentificare folosită în comunicația wireless dintre cele două dispozitive – stația terestră și dronă – este de tip WPA Personal. Astfel, datele transmise între stația de control terestră și UAV sunt criptate folosind protocolul TKIP.

CAPITOLUL 3.

LEHMANN AVIATION LA300 UAV

3.1. Descrierea dronei

LA300 este un UAV complet autonom produs de compania Lehmann Aviation cu aplicații în special în domeniul agriculturii și realizării hărților de mare precizie, precum și în domeniul construcțiilor și mineritului.

Această dronă face parte din categoria dronelor de mici dimensiuni cu care se pot face poze de la altitudini joase de aproximativ 50-100 de metri cu ajutorul unei camere GoPro (Hero 3 sau succesoarele sale), Canon sau MicaSense ce poate fi atașată acesteia.

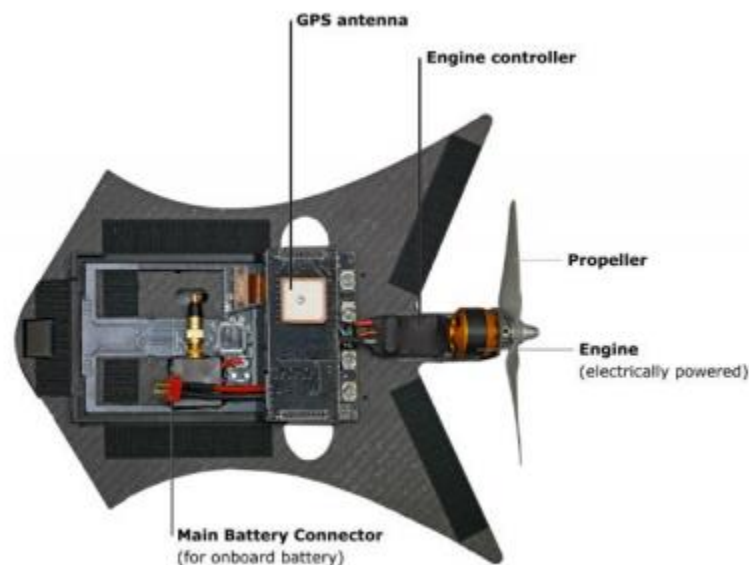


Figura 3.1 [2] – Componentele electronice ale UAV-ului LA300

Din punct de vedere electronic, drona LA300 este compusă în principal din următoarele componente, după cum este prezentată și în Figura 3.1:

- Baterie LiPo;
- Antena GPS;
- Interfața Wi-Fi SNIC SN8200;
- Controller folosit pentru acționarea motorului;
- Motor electric.

Bateria este de tip Lithium Polymer (LiPo) cu o capacitate totală de 2600 mAh și o tensiune nominală de 11.1V, suficient cât să asigure dronei un timp de zbor de aproximativ 30 de minute sau să parcurgă o distanță de aproximativ 15km.

Antena GPS încorporată oferă dronei un sistem de radionavigație prin care aceasta găsește cu ajutorul sateliților geolocația punctelor traiectoriei prin care trebuie să treacă.

Interfața wireless SN8200 este un controller produs de compania Murata ce este special conceput pentru a oferi suport în comunicațiile wireless. Prin acest controller stația de control terestră poate comunica cu UAV-ul și îi poate trimite setările cu coordonatele traiectoriei.

Controllerul de motor citește setările primite pe interfața wireless și acționează motorul electric astfel încât drona să parcurgă traiectoria dorită.

3.2. Interfața wireless SN8200

Interfața wireless SN8200 reprezintă o soluție wireless încorporată cu consum redus de energie ce adresează cerințele de conectivitate în aplicațiile Machine to Machine. Acest modul integrează un micro-controller ARM Cortex M3, Wi-Fi MAC/RF IC, memorie flash, generatoare de ceas și o antenă. Modulul SN8200 poate fi controlat de un dispozitiv printr-o interfață serială, însă poate servi, de asemenea, și ca o stație Wi-Fi sau controller de rețea de sine stătător. De aceea, acest modul permite obținerea unei conectivități wireless unei game variate de produse cu minimul de resurse de energie.

Interfața SN8200 furnizează suport pentru standardele IEEE 802.11 b/g/n și poate fi folosit pentru o varietate de aplicații diferite precum un dispozitiv de emisie-recepție serial - Wi-Fi, controller de rețea Wi-Fi, gateway Wi-Fi sau server de rețea.

Atunci când modulul SN8200 este controlat de către un dispozitiv de la distanță, interfața serială este foarte ușor de integrat. Folosit într-un sistem necontrolat de către un alt dispozitiv, micro-controllerul ARM Cortex-M3 poate fi folosit pentru a rula o gamă variată de aplicații.

Modulul Wi-Fi SN8200 poate fi folosit ca gateway Wi-Fi dacă este combinat cu un alt modul radio. Spre exemplu, un gateway Wi-Fi ZigBee poate fi implementat prin combinarea modulului SN8200 cu modulul ZigBee SyChip SN3020. Această proiectare furnizează o soluție cu costuri scăzute și consum de energie redus pentru aplicații precum Home Area Networks.

Sumarul modulului, după cum este prezentat și în Diagrama Bloc din Figura 3.2: [3]

- Tehnologie radio cu suport pentru standardele IEEE 802.11 b/g/n la frecvența 2.4GHz;
- Dimensiuni 30.5 mm x 19.4 mm x 2.8 mm;
- Antenă încorporată;
- Puterea de transmisie +18 dBm @ 11b/11Mbps;
- MCU: ARM Cortex-M3;
- Interfețe seriale: UART, SPI;
- Suport pentru aplicații cu senzori: ADC, DAC, I2C, GPIO;
- Temperatura de operare: de la -30 la 85 grade Celsius;

- Conform RoHS, MSL Level 3, EC.

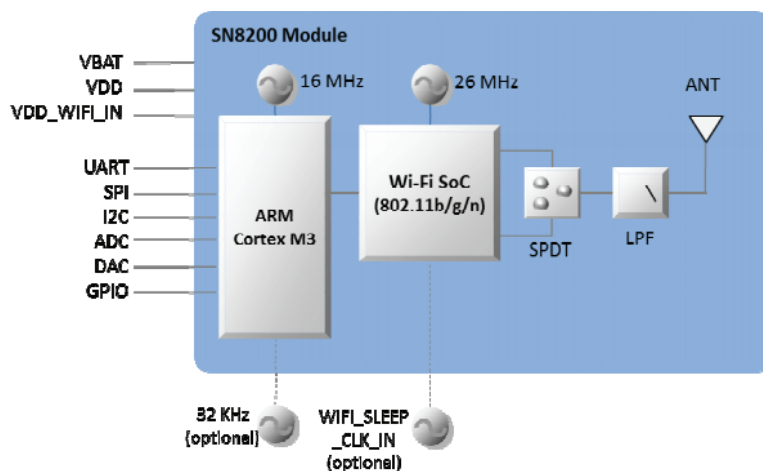


Figura 3.2 [3] – Diagrama Bloc a Modulului SN8200

Din punct de vedere electric, consumul tipic de energie al acestui modul, la temperatura ambiantă de 25 de grade Celsius, este prezentat în Tabelul 3.1:

Item		Condition	Values			Units
			Min	Typ	Max	
11b	Receive mode	11Mbps		110		mA
	Transmit mode (18dBm/ 100% Duty Cycle)			370		mA
11g	Receive mode	54Mbps		110		mA
	Transmit mode (14.5 dBm/ 100 % Duty Cycle)			290		mA
11n	Receive mode	MCS7		110		mA
	Transmit mode (13.5dBm/ 100% Duty Cycle)			280		mA
Standby Mode with IEEE802.11 Power Save		DTIM 1, Telnet session established and idling		3.15		mA
Standby Mode with IEEE802.11 Power Save		DTIM 3, Telnet session established and idling		1.28		mA

Tabelul 3.1 [3] – Specificațiile electrice DC ale modulului SN8200

Din considerente ce țin de viteza de transmisie a datelor, banda de frecvență în care operează, precum și compatibilitatea cu dispozitivele ce folosesc celelalte tipuri de standarde IEEE, dintre cele trei standarde pe care le pune a dispoziție interfața wireless SN8200 în realizarea comunicației wireless dintre stația de control terestră și UAV am ales să folosesc

standardul IEEE 802.11g. Caracteristicile RF oferite de modulul SNIC SN8200 pentru acest standard sunt prezentate în Tabelul 3.2:

Parameters	Specification			
Standard conformance	IEEE 802.11 g			
Modulation	OFDM			
Data rate	6, 9, 12, 18, 24, 36, 48, 54Mbps			
RF Characteristics	Min	Typ.	Max.	Unit
Frequency range	2400	--	2483.5	MHz
Carrier frequency error	-20	--	+20	ppm
Transmit output power	12.5	14.5	16.5	dBm
Spectrum mask				
9MHz to 11MHz (0dB ~ -20dB)	0		-	dB
11MHz to 20MHz (-20dB ~ -28dB)	0		-	dB
20MHz to 30MHz (-28dB ~ -40dB)	0		-	dB
30MHz to 33MHz (-40dB)	0		-	dB
Constellation Error (EVM)	--	--	-25	dB
Outband spurious emissions				
30MHz to 1GHz (BW=100KHz)			-96	dBm
1GHz to 12.75GHz (BW=1MHz)			-41	dBm
1.8GHz to 1.9GHz (BW=1MHz)			-65	dBm
5.15GHz to 5.3GHz (BW=1MHz)			-85	dBm
Received Minimum Sensitivity				
6Mbps (PER ≤ 10%)	-87	-89	--	dBm
54Mbps (PER ≤ 10%)	-72	-74	--	dBm
Maximum input level (PER ≤ 10%)	-13	--	--	dBm
Adjacent channel rejection (PER ≤ 10%)	-1			dB

Tabelul 3.2 [3] – Caracteristicile RF pentru standardul IEEE802.11g ale SN8200

După cum este prezentat în Figura 3.3, comunicarea cu SN8200 ca Simple Network Interface Controller se poate face pe:

- Interfața UART folosind Socket API;
- Interfața SPI folosind Peripheral API.

În comunicației wireless cu UAV-ul LA300 setările sunt trimise către interfața UART a modulului SN8200 folosind Socket API, secvența comenzilor care trebuie executate pe client pentru a putea comunica cu interfața wireless fiind următoarea:

1. Cerere Init SNIC;
2. Cerere configurare IP SNIC;
3. Cerere creare socket TCP;

4. Cerere creare conexiune TCP;
5. Cerere trimtere date pe socket;
6. Cerere închidere socket;
7. Cerere cleanup SNIC;

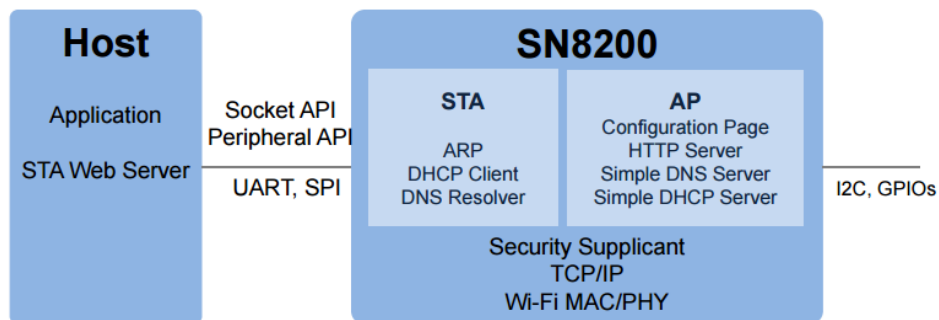


Figura 3.3 [4] – Simple Network Interface Controller

Aplicația client ce rulează pe dispozitivul care controlează UAV-ul nu trebuie să trateze în niciun fel orice control Wi-Fi, această funcție fiind îndeplinită în totalitate de server-ul găzduit de AP. Există totuși API pentru controlul Wi-Fi opțional.

Framework-ul EZ Web Wizzard suportă servicii web simple SWS - o metodă prin care aplicațiile pot fi executate pe chip-ul SN8200, aplicații ce controlează modulul, precum și alte periferice sau senzori și îndeplinesc următoarele funcții:

- Executarea de acțiuni specifice Wi-Fi precum scanarea AP-urilor disponibile;
- Configurarea și managementul IP al rețelei;
- Configurarea și controlul perifericelor conectate la SN8200 prin GPIO, I2C, UART.

Interfața wireless SNIC SN8200 poate funcționa atât ca AP, cât și ca stație wireless simultan. Clienții au posibilitatea de a comunica cu web-serverul încapsulat prin una dintre interfețele AP sau STA ale modulului. Platforma EZ Web Wizzard permite dezvoltatorilor de aplicații limitarea accesului la o pagină web specifică la una dintre cele două interfețe. Diagrama bloc pentru framework-ul încapsulat este prezentată în Figura 3.4.

EZ Web Wizard furnizează un API pentru a oferi posibilitatea realizării de configurații specifice, prin care pot fi setați următorii parametri:

- Parametri AP:
 - Parametri WLAN precum SSID, modul și cheia de securitate;
 - Parametri serverului DHCP precum masca de rețea, adresa IP a AP-ului etc.;
 - Domeniul DNS.
- Parametri STA:
 - Parametri WLAN precum SSID, modul și cheia de securitate.
 - Parametri IP precum configurarea adresei prin IP static sau DHCP.

- Hostname-ul publicat serverului DHCP, dacă este cazul.

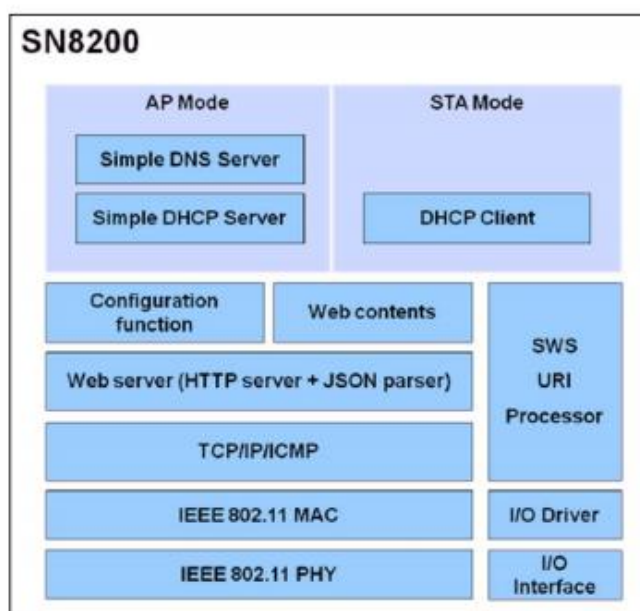


Figura 3.4 [5] – Framework EZ Web Wizzard pentru servicii web simple

Modul AP încorporează atât un server DHCP, cât și un server DNS. Serverul DHCP este utilizat pentru a asigura adrese IP dispozitivelor asociate cu AP-ul, iar serverul DNS permite clienților HTTP să se conecteze la EZ Web Wizzard fără a cunoaște efectiv adresa IP a interfeței. Un scenariu tipic de utilizare al modului AP ar putea fi următorul:

1. Utilizatorul asociază clientul cu AP-ul interfeței wireless SN8200 și obține o adresă IP;
2. Utilizatorul se conectează într-un web browser la EZ WEB Wizzard folosind domeniul configurat pentru modul, “sn8200.com”;
3. Utilizatorul configurează parametrii WLAN sau celelalte funcții ale SN8200;
4. Utilizatorul accesează interfața STA după aderarea cu succes la WLAN.

Interfața wireless SN8200 poate funcționa și ca un STA astfel încât să poată adera și să se poată conecta la un alt WLAN. Modul STA pune la dispoziție un client DHCP pentru ca modulul să obțină o adresă IP.

De asemenea, SN8200 pune la dispoziție și un web-server încorporat ce suportă următoarele funcții:

- Server HTTP și un JSON parser pentru procesarea request-urilor HTTP.
- Funcții specifice pentru controlul perifericelor conectate la interfața wireless.

Serverul HTTP rulează pe interfața wireless și poate procesa request-urile primite de la interfețele AP sau STA ale modulului SN8200, cu următoarele constrângeri:

- Conexiunile persistente nu sunt suportate.

- Setările specifice pentru controlul perifericelor conectate la modul trebuie transportate sub formatul unui JSON.

Setările specifice pentru controlul perifericelor conectate la interfața sunt preluate de JSON parser cu condiția ca obiectul JSON transferat prin operația POST să nu conțină spații inutile între componentele obiectului.

3.3. Transmiterea setărilor către controller

Setările dronei sunt transmise către interfața UART a modului wireless SN8200 în formatul unui obiect JSON ce va fi parsat de către modul, setările primite și interpretate fiind apoi transmise de către interfața wireless pe porturile de ieșire către controller-ul care acționează motoarele UAV-ului.

Formatul obiectului JSON folosit în cadrul aplicației pe care am creat-o este disponibil în Anexa 1 și în acest mod pot fi transmise și aplicate UAV-ului următoarele setări:

- Punctul de decolare;
- Traseul pe care îl va urma UAV-ul, determinat printr-o mulțime finită de puncte;
- Punctele în care o cameră atașată UAV-ului ar putea captura imagini/clipuri video;
- Altitudinea la care va zbura UAV-ul în fiecare dintre punctele traseului, relativă la nivelul punctului de decolare;
- Altitudinea absolută a fiecăruia dintre punctele traseului, față de nivelul mării;
- Punctul central al traseului;
- Raza zonei în care este indicat să nu fie niciun obstacol înainte de decolarea, precum și în timpul zborului UAV-ului;
- Modul în care se va face aterizarea – în spirală deasupra punctului de aterizare sau pe o traiectorie asimptotică asemănătoare aterizării unui avion;
- Punctul de aterizare.

Pozițiile tuturor punctelor setate prin obiectul JSON – punctele de aterizare/decolare, punctele traseului, precum și punctele în care camera atașată UAV-ului va captura imagini sunt specificate prin coordonate geografice – latitudine, respectiv longitudine. Pentru a putea distinge între cele două coordonate, fiecăruia îi este asociat un tip. Astfel, latitudinea va fi interpretată ca având tipul 0, iar longitudinea tipul 1. Pentru o precizie cât mai bună ambele coordonate geografice sunt numere fracționare, ce pot avea până la 15 zecimale.

Altitudinile punctelor, atât cele relative, cât și cele absolute, sunt specificate în metri și reprezintă numere întregi. Punctele de decolare și aterizare ale dronei vor avea obligatoriu înălțimea relativă 0. Dacă între două puncte consecutive ale traseului există o diferență ce depășește un anumit prag, atunci UAV-ul va trebui să știe dacă să își crească, respectiv să scadă altitudinea la care zboară înainte și după ce a trecut de punctul prin care zboară. Aceste informații îi pot fi transmise UAV-ului prin câmpurile conținute de obiectul JSON denumite "IsPostAscHelical", "IsPostDescHelical", "IsPreAscHelical", "IsPreDescHelical".

În Anexa 1 am creat o serie de setări valide care pot fi transmise UAV-ului astfel încât să realizeze un zbor în siguranță deasupra Campusului Politehnica. Setările expuse sub forma obiectului JSON din Anexa 1 sunt reprezentate vizual în Figura 3.5.



Figura 3.5 – Setările transmise UAV-ului reprezentate vizual

Traseul prezentat conține punctul de decolare marcat pe hartă cu T – “TakeOff”, 10 waypointuri pe care UAV-ul trebuie să le urmeze în timpul zborului marcate pe hartă în ordine de la 1 la 10, precum și punctul de aterizare marcat cu L – “Landing”. Dacă se dorește ca UAV-ul să aterizeze în același loc din care a decolat, punctele T și L pot fi suprapuse. Pentru exemplificare, punctele au fost alese diferite dar destul de apropiate, astfel încât UAV-ul să poată fi recuperat de către operator fără necesitatea deplasării acestuia între momentele de decolare, respectiv aterizare ale dronei.

Pentru siguranța zborului și integritatea UAV-ului în momentul zborului dronei nu ar trebui să existe o altă aeronavă care să își efectueze zborul pe o rază mai mică decât limita determinată prin nodul "SelectedFlyZoneRadius" al obiectului JSON. Această limită a fost aleasă și reprezentată pe hartă ca fiind 500 de metri.

Punctele trasului urmat de către UAV în care o cameră atașată poate captura imagini îi sunt transmise UAV-ului prin setarea nodului “IsPhoto” cu valoarea booleană true. Astfel, în

exemplul prezentat în Anexa 1 există patru puncte în care vor fi capturate imagini, și anume punctele marcate pe hartă cu numerele 1, 4, 6 și 9.

Deoarece punctele 4, 6 și 9 au o altitudine semnificativ mai mică decât restul punctelor traseului, UAV-ul trebuie să își modifice poziția eleroanelor astfel încât să își diminueze altitudinea la care zboară astfel încât să ajungă la altitudinea dorită în aceste puncte, după care eleroanele trebuie acționate în sens invers pentru ca drona să revină la altitudinea mai mare dorită de punctele consecutive ale traseului. Folosind setările "IsPostAscHelical", "IsPostDescHelical", "IsPreAscHelical", "IsPreDescHelical" amintite mai sus, există mai multe combinații de realizare a acestui lucru. Soluția exemplificată în Anexa 1 este prin setarea nodului "IsPostDescHelical" corespunzător waypointului 3 și nodurile "IsPreDescHelical" corespunzătoare waypointurilor 6 și 9 cu valoarea booleană true. Acest lucru este marcat pe harta exemplificată folosind culoarea roșu pentru waypointurile 3, 6, respectiv 9. Este de remarcat că deși waypointul 4 este cel care are altitudinea mai mică și nu punctul 3, existența nodurilor "IsPostDescHelical" și "IsPostAscHelical" permit specificarea schimbării de altitudine cu un punct în avans.

CAPITOLUL 4.

TEHNOLOGIA SOFTWARE UTILIZATĂ PE STAȚIA DE CONTROL TERESTRĂ

4.1. Prezentarea aplicației

În funcție scopul realizării comunicației cu UAV-ul, pe stația de control terestră poate fi nevoie de portabilitatea sistemului și/sau resurse hardware mari – putere de procesare, memorie. UAV-ul fiind un sistem mobil, este nevoie ca și sistemul care îl controlează wireless să se adapteze la această cerință. Deoarece distanța pe care poate funcționa comunicația wireless este limitată, un exemplu ar putea fi cel în care ne dorim să lansăm UAV-ul din locații geografice diferite, iar înainte de lansarea acestuia dorim să îi transmitem comenzile fără a fi nevoie ca UAV-ul să fie deplasat către un eventual sistem fix care să îl controleze, ci sistemul care îl controlează să poată fi deplasat în orice punct din care am dori decolarea dronei. Astfel, avem nevoie ca stația de control terestră să poată fi portabilă. Pe de altă parte, dacă presupunem că acest sistem trebuie să controleze simultan sau să stocheze informațiile primite de la mai multe UAV-uri, este nevoie ca stația de control să dispună de resurse hardware mari.

Pentru a îndeplini ambele cerințe enunțate mai sus a fost nevoie ca aplicația software prin care este realizată comunicația cu drona să poată fi rulată atât pe un telefon mobil cât și pe un laptop/desktop. Din acest motiv am ales să implementez o aplicație UWP.

Universal Windows Platform este platforma de aplicații pentru Windows 10. Aplicațiile UWP pot fi dezvoltate folosind un singur set API și pot fi descărcate din Windows Store pe orice dispozitiv ce are ca sistem de operare Windows 10 – PC, tabletă, telefon, Xbox, HoloLens, Surface Hub și altele. Aplicațiile create folosind această platformă suportă o mulțime de dimensiuni, modele și rezoluții ale ecranului, precum și o varietate de modele de interacțiune - fie că este mouse, tastatură, touch etc. În spatele aplicațiilor UWP există ideea că utilizatorii își doresc ca aplicația să poată rula pe toate dispozitivele pe care le au și să aibă disponibilă opțiunea de a alege dispozitivul care este cel mai potrivit sau productiv pentru sarcina respectivă. [11]

UWP folosește Windows Runtime, un API nativ al sistemului de operare. Acest API este implementat în limbajul C++ și suportat în limbajele C#, Visual Basic, C++ și JavaScript. Astfel, câteva opțiuni de dezvoltare a aplicațiilor în UWP includ:

- Interfață grafică XAML susținută de descrierea funcțională a aplicației într-unul dintre limbajele C#, VB sau C++;
- Interfață grafică DirectX competată de un backend C++;
- JavaScript și HTML.

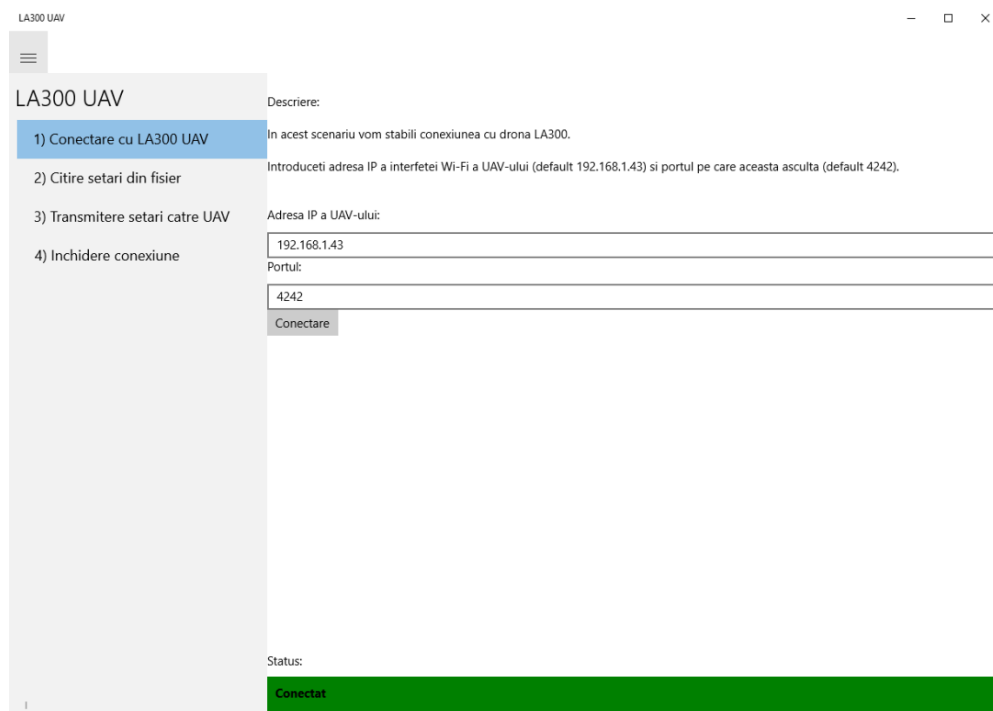
Implementarea aplicației software folosită pe stația terestră pentru a comunica cu UAV-ul a fost realizată în mediul de dezvoltare Microsoft Visual Studio folosind limbajul C++, iar interfața grafică a fost realizată folosind XAML.

Microsoft Visual Studio este un mediu de dezvoltare integrat folosit pentru crearea de aplicații și programe pentru sistemul de operare Microsoft Windows, precum și pentru site-uri, aplicații și servicii web și utilizează mai multe platforme de dezvoltare software, printre care și Universal Windows Platform. Visual Studio suportă 36 de limbaje de programare diferite, cele mai comune fiind C, C++, VB.NET, Visual C#, XML, HTML, JavaScript sau CSS. Suport pentru alte limbaje precum Python sau Ruby poate fi obținut prin instalarea de pachete adiționale. Pentru dezvoltarea aplicațiilor UWP, Visual Studio pune la dispoziție mai multe funcționalități precum implementarea codului sursă, realizarea designului aplicației, integrarea cu servicii cloud, debug și analiza, optimizarea și testarea aplicației.

4.2. Arhitectura aplicației

Din punct de vedere structural, aplicația software este constituită din patru scenarii ce urmăresc respectarea pașilor descriși în subcapitolul 3.2. impuși de către controllerul SN8200 cu care trebuie să comunice wireless:

1. Crearea sesiunii de comunicare – Figura 4.1. Prin specificarea adresei IP și a portului UAV-ului, apăsând butonul “Conectare” în acest scenariu poate fi deschis socketul pe care se va face comunicația;



The screenshot displays the 'LA300 UAV' application window. On the left is a sidebar menu with four options: '1) Conectare cu LA300 UAV' (highlighted in blue), '2) Citire setari din fisier', '3) Transmitere setari catre UAV', and '4) Inchidere conexiune'. The main area on the right contains a description: 'In acest scenariu vom stabili conexiunea cu drona LA300. Introduceti adresa IP a interfetei Wi-Fi a UAV-ului (default 192.168.1.43) si portul pe care aceasta asculta (default 4242)'. Below this are two input fields: 'Adresa IP a UAV-ului:' with the value '192.168.1.43' and 'Portul:' with the value '4242'. A 'Conectare' button is positioned below the port field. At the bottom, a 'Status:' label is followed by a green bar with the text 'Conectat'.

Figura 4. 1 – Inițiere conexiune cu UAV-ul

2. Citirea/scrierea setărilor care urmează a fi transmise către UAV dintr-un fișier de pe disk – Figura 4.2. Pentru a putea fi interpretate în mod corect setările trebuie să fi în format JSON, fără niciun spațiu inutil între noduri;

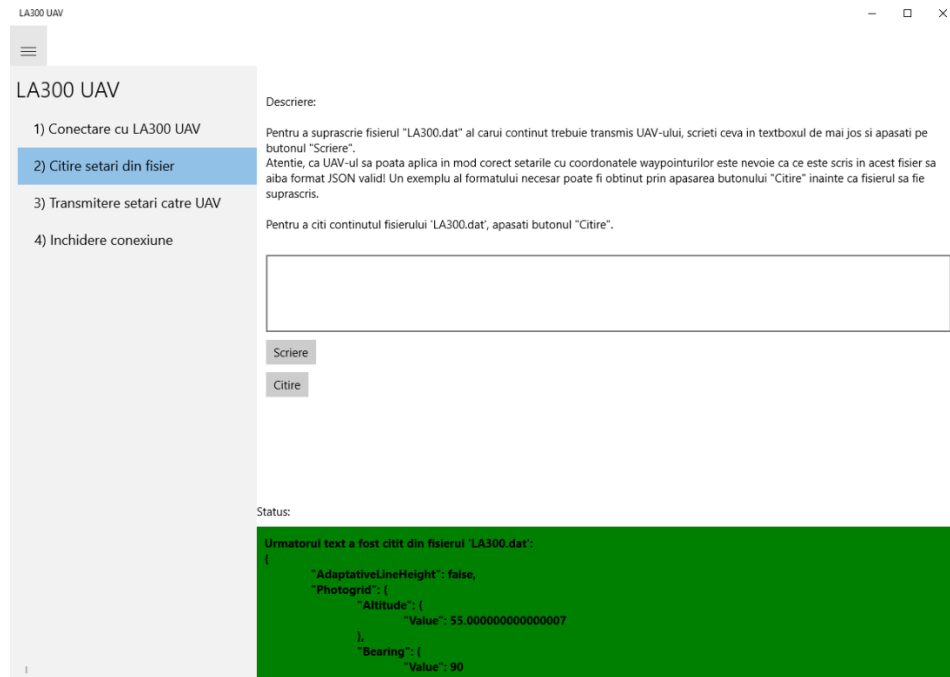


Figura 4.2 – Citire setări din fișier

3. Transmiterea setărilor către UAV – Figura 4.3. În acest scenariu setările citite în scenariul anterior sunt transmise pe socketul creat in Scenariul 1 către UAV;

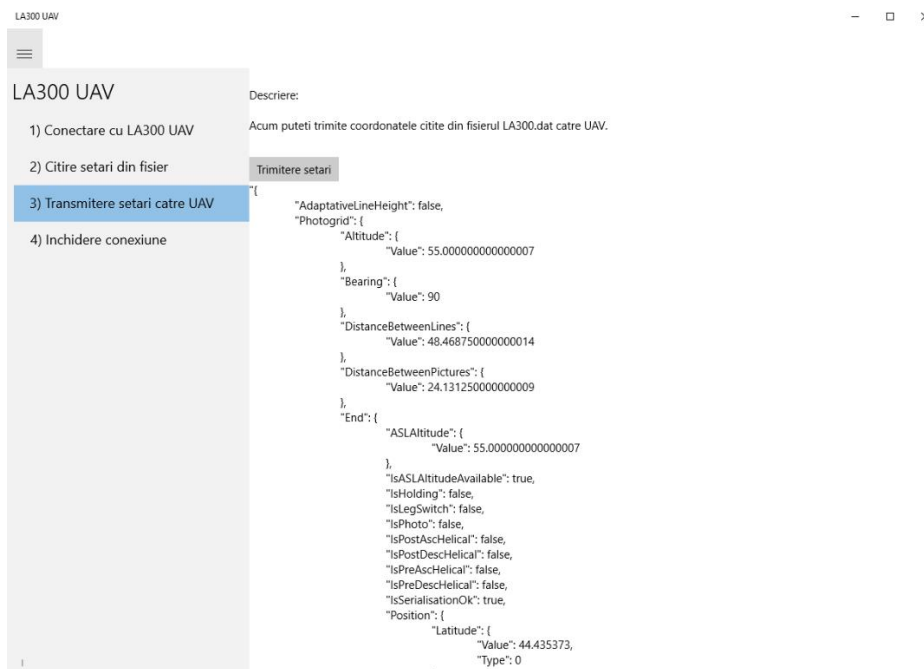


Figura 4.3 – Transmitere setări către UAV

4. Închiderea sesiunii de comunicare – Figura 4.4. Din motive de securitate a datelor la sfârșitul comunicației trebuie închis socketul pe care s-a realizat conexiunea. Acest lucru este realizat în Scenariul 4.

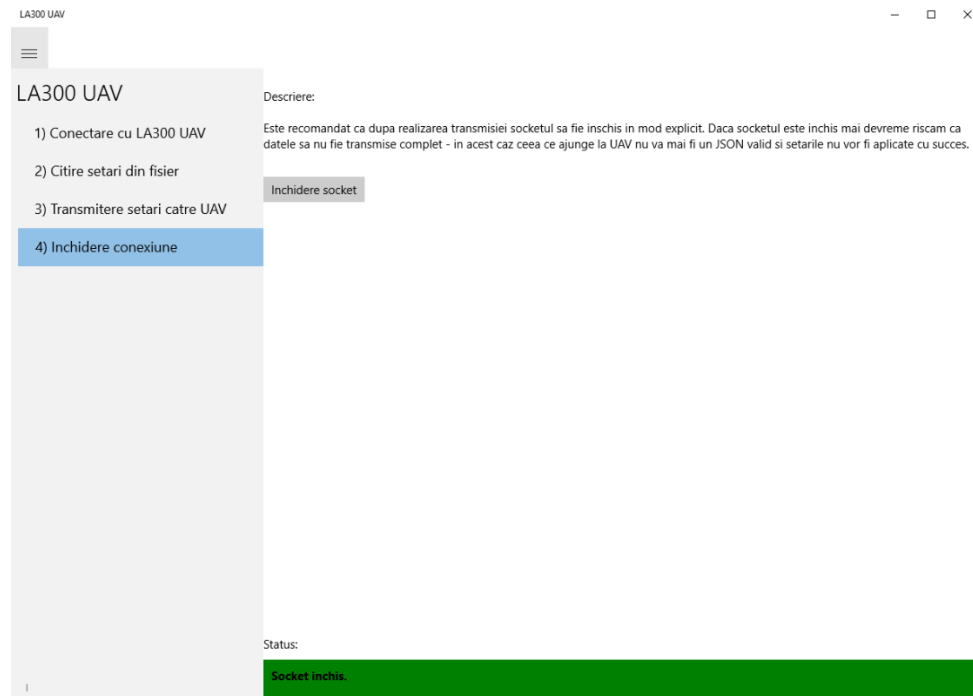


Figura 4.4 – Închidere conexiune cu UAV-ul

Proiectul aplicației UWP implementată conține următoarele fișiere:

- App.xaml, App.xaml.h, App.xaml.cpp: Reprezintă obiectul Aplicației. În fișierul App.xaml pot fi adăugate stiluri ale interfeței și alte elemente grafice ce se doresc a fi accesibile din orice pagină. Codul din spatele acestor fișiere conține manipulatori pentru evenimentele OnLaunched și OnSuspending. Tipic, aici este adăugat cod pentru inițializarea aplicației în momentul în care este lansată sau pentru realizarea cleanup-ului în cazul în care se dorește terminarea execuției aplicației sau în cazul în care procesul aplicației intră în starea suspended;
- MainPage.xaml, MainPage.xaml.h, MainPage.xaml.cpp: Aceste fișiere conțin codul din spatele paginii principale, de start, a aplicației. Nu oferă suport pentru navigație sau controale built-in;
- Pch.h, pch.cpp: Un fișier header precompilat, respectiv fișierul care include fișierul precompilat în proiect. În pch.h pot fi incluse orice headere care nu sunt schimbate des și nu sunt incluse în alte fișiere ale proiectului;
- Package.appxmanifest: Un fișier XML care descrie capacitățile pe care trebuie să le aibă dispozitivul pe care rulează aplicația, informații despre versiunea aplicației sau alte metadata.

- <NumeleProiectului>_TemporaryKey.pfx: O cheie prin care aplicația poate fi deployată pe mașina pe care rulează mediu de dezvoltare Visual Studio;
- Scenario<nr.>.xaml, Scenario<nr.>.xaml.h, Scenario<nr.>.xaml.cpp: Fiecare dintre cele patru scenarii implementate în aplicație conține câte trei fișiere de acest tip. În fișierul cu extensia .xaml este realizată interfața grafică prin adăugarea elementelor de tip butoane, textbox-uri etc. Descrierea funcțională este realizată în limbajul C++ în fișierele cu extensia .xaml.h, respectiv .xaml.cpp;
- Global.h: Un fișier header în care a fost declarată o variabilă globală folosită în mai multe scenarii.

Fiecare fișier .xaml din folderul proiectului are fișierele corespunde .xaml.h și .xaml.cpp în același folder, respectiv fișierele .g și .g.hpp în folderul Generated Files care se află pe disk, însă nu este parte din proiect. Fișierele XAML sunt folosite pentru a crea elemente de interfață și pentru a le conecta la surse de date. Fișierele .h și .cpp sunt modificate pentru a realiza logica manipulatorilor de evenimente. Fișierele autogenerate reprezintă transformarea XAML în limbaj C++. La build codul scris de dezvoltator și cel autogenerat fuzionează într-o definiție a unei clase care apoi este compilată. [12]

În Figura 4.5 este prezentată o captură Wireshark prin care poate fi observată realizarea cu succes a comunicației wireless dintre stația de control terestră și UAV-ul LA300 după efectuarea tuturor pașilor descriși mai sus. Wireshark reprezintă un analizor de pachete disponibil open source și este folosit pentru analiza și depanarea rețelei, dezvoltarea aplicațiilor software și a protocoalelor de comunicație, precum și în scop educativ.

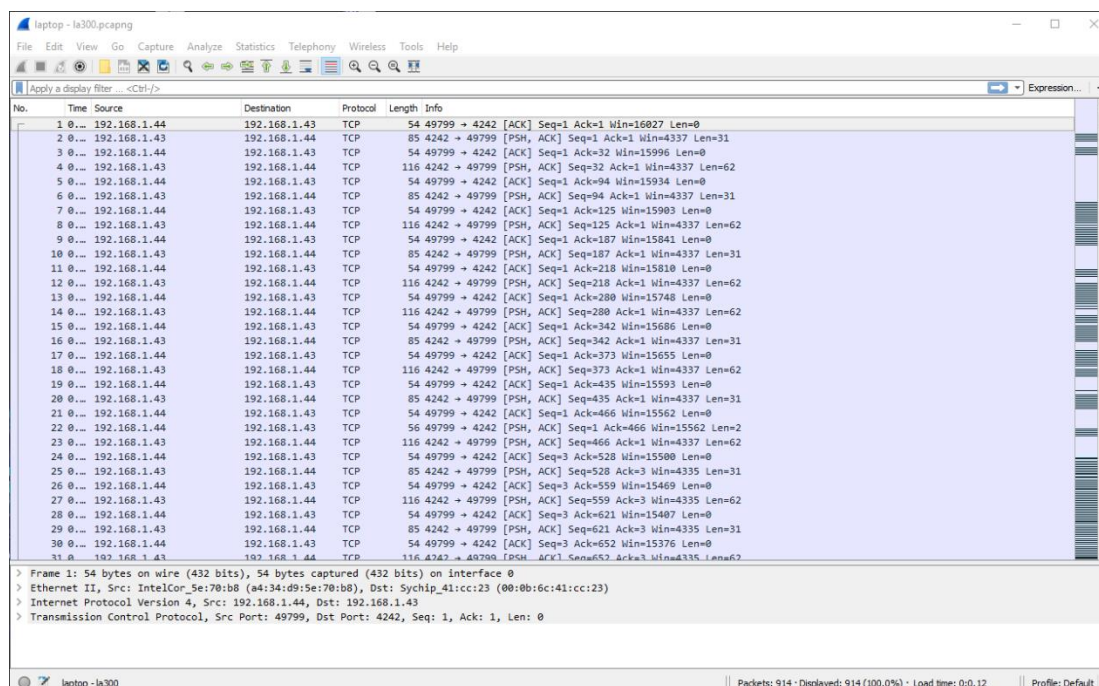


Figura 4.5 – Captură Wireshark comunicație wireless

4.3. Implementare

Pentru a urma pașii impuși de realizarea unei comunicații cu modulul wireless SN8200 aflat la bordul UAV-ului LA300, în aplicația UWP creată am ales ca pentru fiecare pas să am un scenariu diferit, la care am adăugat un scenariu suplimentar în care sunt citite, respectiv scrise setările ce urmează a fi transmise dintr-un fișier de pe disk. Astfel, aplicația UWP creată este compusă din următoarele patru scenarii, respectiv pași în realizarea comunicației cu UAV-ul:

1. Conectarea cu UAV-ul – practic inițializarea unei sesiuni de comunicare;
2. Citirea sau scrierea setărilor ce urmează a fi transmise către UAV dintr-un fișier de pe disk. Aceste setări trebuie să aibă formatul JSON;
3. Transmiterea setărilor în format JSON către modulul wireless SN8200;
4. Închiderea sesiunii de comunicare.

Fiecare dintre cele patru scenarii diferite ale aplicației UWP este implementat ca fiind o clasă diferită ce conține variabile și metode specifice scenariului respectiv, iar toată aplicația este implementată ca fiind un namespace. În programarea orientată pe obiecte o clasă reprezintă o porțiune de cod extensibilă dedicată creării de obiecte ce furnizează valori inițiale de stare (variabile) și implementări funcționale (metode), în timp ce un namespace reprezintă un concept ce permite organizarea variabilelor, a funcțiilor și a altor elemente ale programului. Un namespace încapsulează codul astfel încât pentru a accesa codul respectiv în primul rând trebuie referit namespace-ul în care este încapsulat.

Astfel, codul sursă este format din următoarele patru clase:

- Prima clasă implementează crearea unei sesiuni de comunicare și a unui socket pe care se va face transmiterea datelor. Pentru a crea un socket este nevoie de adresa IP și portul dispozitivului destinație, motiv pentru care utilizatorul trebuie să introducă aceste informații ale UAV-ului în textbox-urile respective. Cum adresa IP și portul pe care modulul wireless SN8200 așteaptă să primească informații sunt 192.168.1.43, respectiv 4242, am setat ca acestea să apară și implicit la startul aplicației, însă pot fi suprascrise de către utilizator în cazul în care UAV-ul își schimbă setările. Prin apăsarea butonului Conectare este stabilită conexiunea cu UAV-ul și este deschis socketul pe care se va face comunicarea. În cazul în care cel puțin una dintre informațiile furnizate are un format greșit, la apăsarea butonului Conectare utilizatorul va primi un mesaj de eroare prin care este avertizat că formatul nu este corespunzător. Dacă formatul adresei IP și a portului este corect, însă nu există în rețea un dispozitiv cu adresa respectivă, utilizatorul va primi, de asemenea, un mesaj de eroare prin care va fi înștiințat. În cazul în care conexiunea va fi realizată cu succes după apăsarea butonului Conectare, utilizatorul va primi un mesaj de informare corespunzător.

- Cea de-a doua clasă conține două metode. Prin apăsarea butonului Citire, prima metodă implementează citirea setărilor ce urmează a fi transmise către interfața wireless a UAV-ului dintr-un fișier de pe disk, LA300.dat și stocarea lor într-o variabilă globală ce va fi folosită și în clasa următoare pentru transmiterea setărilor. Dacă în urma citirii setărilor se dorește modificarea și salvarea ulterioară a acestora pe disk înainte de transmiterea datelor către dronă,

acest lucru poate fi realizat prin apăsarea butonului Scriere, iar codul ce susține acest comportament este implementat în a doua metodă a acestei clase.

- Următoarea clasă conține o metodă prin care setările citite la punctul anterior sunt transmise modului wireless pe socketul creat la primul scenariu. În cazul în care primii pași nu au fost parcurși va apărea un mesaj de eroare prin care utilizatorul va fi informat de acest lucru. Dacă totuși pașii anteriori au fost urmați corespunzător, însă din diverse motive transmiterea informațiilor nu s-a realizat cu succes (spre exemplu o posibilă deconectare a clientului de către UAV), va fi aruncată o excepție cu mesajul că transmiterea a eșuat cu eroarea excepției. De asemenea, dacă transmiterea setărilor s-a efectuat cu succes, utilizatorul va fi notificat de acest lucru.

- Ultima clasă conține o metodă care implementează terminarea conexiunii și închiderea socketului creat în primul scenariu și pe care a fost realizată transmiterea setărilor din al treilea scenariu. În urma parcurgerii acestui pas utilizator va fi notificat că socketul a fost închis cu succes.

CAPITOLUL 5.

TESTAREA COMUNICAȚIEI

5.1. Interferențe

Pentru testarea influenței interferențelor în realizarea cu succes a comunicației wireless dintre stația de control terestră și UAV am abordat două cazuri. Primul scenariu implică o comunicație într-un scenariu real în care UAV-ul se află în aer liber unde nu sunt cunoscute interferențele. Cel de-al doilea caz tratat este realizarea comunicației în condiții de laborator în care sunt cunoscute toate dispozitivele care comunică în banda de frecvențe folosită de 2.4 GHz.

În urma realizării testelor într-un scenariu real în care UAV-ul chiar ar putea zbura, mediul de testare fiind în aer liber în Campusul Universității Politehnica, nu am întâmpinat probleme de comunicare. Din 50 de încercări de a transmite setările UAV-ului, toate cele 50 au reușit, iar setările au fost aplicate cu succes. De menționat este faptul că în acest mediu au fost descoperite 3 AP-uri la care s-ar fi putut conecta stația de control terestră și care funcționau în banda de frecvențe de interes.

În cel de-al doilea scenariu, cel controlat în laborator, pentru a testa comunicația am pornit un router wireless la care am conectat șase dispozitive. Pentru a solicita mediul fiecare dintre cele șase dispozitive a fost pus să descarce și să încarce pe internet fișiere de ordinul a câțiva GB, urmând ca apoi dispozitivele să comunice două câte două între ele astfel încât să facă, de asemenea, atât trafic de download, cât și de upload. La fel ca în cazul mediului necontrolat, nici în acest caz nu au fost întâmpinate probleme de comunicare între stația de control și UAV datorate suprasolicitării benzii de 3.4 GHz în niciuna dintre cele 50 de încercări.

Pentru a determina dacă apar pierderi de pachete ar putea fi efectuate teste suplimentare, atât de anduranță, în care stația de control să comunice cu UAV un timp îndelungat, cât și teste în care să fie realizat un trafic mult mai mare în banda de frecvențe de interes. Pentru scopul prezentei lucrări, este suficient însă să știm că nu apar probleme de comunicare într-un mediu de lucru real solicitat în mod normal.

5.2. Influența distanței

Principala problemă care ar trebui pusă în ceea ce privește controlarea unui UAV de la sol este cea a stabilirii distanței maxime la care încă mai putem comunica prin wireless cu respectivul UAV. Acesta este cel mai important factor pentru asigurarea unui control al UAV-ului în timp real când acesta își desfășoară zborul în mod normal. Din acest motiv mi-am propus să determin distanța maximă la care comunicația wireless se realizează cu succes.

Pentru a stabili dacă UAV-ul poate într-adevăr fi controlat de la o anumită distanță nu este suficient să transmitem datele pentru a spune că a fost realizată comunicația cu succes,

deoarece pot apărea pierderi de pachete. De aceea am avut în vedere mai multe aspecte în realizarea testelor de influență a distanței, și anume: dacă UAV-ul poate fi sau nu descoperit și dacă stația de control se poate asocia la el, dacă sesiunea de comunicare poate fi deschisă, respectiv închisă, și nu în ultimul rând dacă setările au fost transmise fără pierderi de pachete. Pentru a determina dacă au apărut sau nu pierderi de pachete am folosit analizorul de trafic Wireshark.

Astfel, rezultatele testelor făcute de influență a distanței pe care controlul UAV-ului poate fi făcut cu succes sunt prezente în Tabelul 4.1. În mod empiric am determinat faptul că distanța maximă la care comunicația wireless este realizată cu succes este de aproximativ 50 de metri, deși standardul folosit de modulul wireless SN8200 cu care comunică stația de control este caracterizat de distanțe de comunicare mai mari, de aproximativ 200-300 de metri.

Distanța	Creare conexiune	Transmitere date	Închidere conexiune	Pierdere pachete
~ 1m	OK	OK	OK	Nu
~ 3 m	OK	OK	OK	Nu
~ 5 m	OK	OK	OK	Nu
~7.5 m	OK	OK	OK	Nu
~ 10 m	OK	OK	OK	Nu
~ 20 m	OK	OK	OK	Nu
~ 30 m	OK	OK	OK	Nu
~ 40 m	OK	OK	OK	Nu
~ 45 m	OK	NOT OK	OK	Da
~ 50 m	NOT OK	NOT OK	NOT OK	Da
~ 60 m	NOT OK	NOT OK	NOT OK	Da

Tabelul 4.1 – Influența distanței în realizarea comunicației wireless

5.3. Influența obstacolelor

O altă problemă ce poate apărea în realizarea comunicației wireless este reprezentată de apariția obstacolelor între stația de control terestră și UAV și modul în care aceasta poate reduce semnificativ distanța pe care cele două dispozitive pot comunica sau poate duce chiar la pierderea completă a legăturii cu UAV-ul.

Bineînțeles, în ceea ce privește obstacolele intervin foarte mulți factori care pot influența mai mult sau mai puțin comunicația. Printre cei mai importanți factori putem enumera materialul din care este construit obstacolul, grosimea obstacolului sau numărul de obstacole. De aceea, chiar și în urma testelor făcute nu putem spune cu siguranță că un obstacol reduce distanța de comunicare cu o valoare fixă.

Cu toate că acest subiect este destul de sensibil la factorii enunțați mai sus, am făcut totuși o serie de teste din care au rezultat următoarele rezultate:

- Un copac nu afectează suficient de mult comunicația wireless încât legătura dintre stația de control și UAV să fie pierdută, iar distanța maximă de comunicare rămâne aproximativ aceeași;
- Un perete de cărămidă cu grosimea de aproximativ 30 de cm nu determină pierderea conexiunii wireless, însă reduce distanța maximă de comunicare la aproximativ jumătate față de cea în care nu este niciun obstacol;
- O clădire din beton duce la pierderea completă a conexiunii, însă acest lucru depinde de dimensiunea clădirii, numărul de pereți ai clădirii, grosimea fiecărui perete în parte;
- În măsurători efectuate la sol prezența a patru mașini între stația terestră și UAV conduce la reducerea distanței de comunicare cu aproximativ 20 de metri.

CONCLUZII

În urma realizării practice a prezentei lucrări au fost îndeplinite cu succes obiectivele setate la începutul lucrării, acelea de a controla de la sol un UAV prin intermediul unei comunicații wireless. Modelul UAV-ului pe care am lucrat este LA300 produs de compania Lehmann Aviation, iar printre setările transmise wireless cu care îl putem controla se numără:

- Punctul de decolare;
- Traseul pe care îl va urma UAV-ul în zborul său, determinat ca o mulțime finită de puncte caracterizate prin coordonatele geografice latitudine și longitudine;
- Altitudinea la care va zbura UAV-ul, prin setarea înălțimilor absolute și relative a fiecărui punct al traseului;
- Punctele în care UAV-ul va captura imagini;
- Punctul de aterizare.

Printre problemele ce pot apărea în cazul unei astfel de comunicații se numără apariția interferențelor, distanța maximă la care mai pot comunica stația de control și UAV-ul, precum și influența obstacolelor dintre cele două sisteme. De aceea mi-am propus realizarea unei serii de teste prin care să pot stabili limitele la care controlul UAV-ului încă poate fi realizat cu succes în mod practic.

Astfel, din punct de vedere al interferențelor, deși comunicația wireless este realizată în banda de frecvențe de 2.4 GHz, bandă de frecvențe suficient de populată, nu am întâmpinat probleme. Comunicația a fost realizată cu succes atât în condiții de laborator în care au existat mai multe routere și dispozitive care funcționau în aceeași bandă de frecvențe, cât și în condiții apropiate de un scenariu real în care stația de control și UAV-ul se aflau în aer liber, mediu mai puțin solicitat în această bandă de frecvențe.

Din punct de vedere al influenței distanței, deși cel puțin în mod teoretic distanța maximă la care cele două dispozitive mai pot comunica cu succes este de aproximativ 200-300 de metri, în urma realizării practice a testelor această limită este mai mică, de doar aproximativ 50 de metri. De menționat este faptul că testele au fost realizate într-un mediu apropiat de un scenariu real, în aer liber și linie dreaptă fără obstacole între dispozitive. Din acest motiv o limitare, cel puțin temporară, ar fi aceea că UAV-ul nu poate fi controlat în timp real în cazul în care se află în zbor la o distanță mai mare de 50 de metri. Setările pot fi transmise și aplicate cu succes UAV-ului înainte de decolare urmând ca după aterizare să poată fi descărcate informațiile dorite, precum imaginile capturate, însă îmi propun ca în viitor această limită maximă a distanței să fie suficient de mare încât UAV-ul să poată fi controlat în timp real.

Dacă apar și alți factori perturbatori precum existența unor obstacole între stația de control și UAV, distanța maximă la care comunicația este realizată cu succes este chiar mai mică, depinzând de mai mulți factori precum materialul din care este construit obstacolul, grosimea obstacolului, numărul de obstacole etc. În urma efectuării testelor am constatat, spre

exemplu, că prezența unui copac între stația de control și UAV nu influențează în mod negativ, UAV-ul putând fi în continuare controlat cu succes, în timp ce prezența unei clădiri construită din beton limitează complet comunicația, legătura cu UAV-ul fiind pierdută. În mod empiric, cu măsurători efectuate la sol, prezența a patru mașini pe linia dintre cele două dispozitive reduce distanța la care poate fi realizată comunicația de la 50 la aproximativ 30 de metri.

Limitarea comunicației wireless la o distanță de aproximativ 50 de metri provine de la modulul wireless SN8200 al UAV-ului care suportă doar trei standarde wireless tradiționale, și anume IEEE 802.11 b/g/n. Chiar și în mod teoretic aceste standarde sunt caracterizate de o distanță de comunicare de aproximativ 200-300 de metri. Dacă se dorește controlarea UAV-ului în timp real trebuie aleasă o tehnologie mai dezvoltată precum WiMAX sau ZigBee.

Bibliografie

- [1] "Montana State University," [Online]. Available: <http://www.cs.montana.edu/~halla/csci466/lectures/lec25/lec25-tcp>. [Accessed 05 06 2017].
- [2] "L-A Series Drones User Guide," Lehmann Aviation, 2015.
- [3] "SN8200 Wi-Fi Network Controller Module User Manual and Datasheet," 2013.
- [4] "SN8200 Module Software," Murata, 2013.
- [5] "SN82xx SNIC EZ Web Wizzard Development Platform User Manual," Murata, 2013.
- [6] "rfc-793," Information Sciences Institute, Arlington, Virginia, 1981.
- [7] C. M. Kozierok, "TCPIPGuide," [Online]. Available: http://www.tcpipguide.com/free/t_TCPCConnectionEstablishmentProcessTheThreeWayHandsh-3.htm. [Accessed 07 05 2017].
- [8] W. Odom, CCNA Routing and Switching 200-125 Official Cert Guide, Cisco Press, 2016.
- [9] "IEEE 802.11™: Wireless LANs," IEEE Standards Association, 2012.
- [10] K. Barker and S. Morris, CCNA Security 640-554 Official Cert Guide, Cisco Press, 2013, pp. 8-19.
- [11] "Microsoft Docs," Microsoft, [Online]. Available: <https://docs.microsoft.com/en-us/windows/uwp/get-started/whats-a-uwp>. [Accessed 20 04 2017].
- [12] "Microsoft Docs," Microsoft, [Online]. Available: <https://docs.microsoft.com/en-us/windows/uwp/get-started/create-a-basic-windows-10-app-in-cpp>. [Accessed 20 04 2017].

Anexa 1

```
{
  "AdaptativeLineHeight": null,
  "Photogrid": {
    "Altitude": null,
    "Bearing": null,
    "DistanceBetweenLines": null,
    "DistanceBetweenPictures": null,
    "End": null,
    "FirstWaypointDistance": {
      "Value": 100
    },
    "ForwardOverlap": null,
    "Height": null,
    "LastWaypointDistance": {
      "Value": 100
    },
    "Lines": [],
    "NbPicturePerLine": 0,
    "Rotation": null,
    "SideLap": null,
    "Start": null,
    "Width": null
  },
  "SavedMissionType": 1,
  "SelectedCameraFocalIndex": 0,
  "SelectedForwardOverLapIndex": 11,
  "SelectedGroundDefinitionIndex": 0,
  "SelectedSideLapIndex": 8,
  "WaypointsMap": {
    "Center": {
      "Latitude": {
        "Value": 44.433288901527931,
        "Type": 0
      },
      "Longitude": {
        "Value": 26.047720134639121,
        "Type": 1
      }
    },
    "FlyZoneRadius": {
      "Type": 16,
      "Visibility": 0,
      "Waypoint": {
        "ASLAltitude": {
          "Value": 0
        },
        "IsASLAltitudeAvailable": false,
        "IsHolding": false,
        "IsLegSwitch": false,
        "IsPhoto": false,
        "IsPostAscHelical": false,
        "IsPostDescHelical": false,
        "IsPreAscHelical": false,
        "IsPreDescHelical": true,
        "IsSerialisationOk": true,
        "Position": {
          "Latitude": {
            "Value": 44.436930499589721,
            "Type": 0
          },
          "Longitude": {
            "Value": 26.046906874098227,
            "Type": 1
          }
        },
        "RelativeAltitude": {
          "Value": 0
        }
      }
    },
    "IsPostAscHelical": false,
    "IsPostDescHelical": false,
    "IsPreAscHelical": false,
    "IsPreDescHelical": false,
    "IsSerialisationOk": true,
    "Position": {
      "Latitude": {
        "Value": 44.435722588117819,
        "Type": 0
      },
      "Longitude": {
        "Value": 26.04648216903637,
        "Type": 1
      }
    },
    "RelativeAltitude": {
      "Value": 100
    }
  }
},
  "IsLandingDirectionEnabled": false,
  "Landing": {
    "Type": 2,
    "Visibility": 0,
    "Waypoint": {
      "ASLAltitude": {
        "Value": 84
      },
      "IsASLAltitudeAvailable": true,
      "IsHolding": false,
      "IsLegSwitch": false,
      "IsPhoto": false,
      "IsPostAscHelical": false,
      "IsPostDescHelical": false,
      "IsPreAscHelical": false,
      "IsPreDescHelical": true,
      "IsSerialisationOk": true,
      "Position": {
        "Latitude": {
          "Value": 44.436930499589721,
          "Type": 0
        },
        "Longitude": {
          "Value": 26.046906874098227,
          "Type": 1
        }
      },
      "RelativeAltitude": {
        "Value": 0
      }
    }
  }
}
```

```

    }
  },
  "LandingDirection": {
    "Type": 18,
    "Visibility": 1,
    "Waypoint": {
      "ASLAltitude": {
        "Value": 0
      },
      "IsASLAltitudeAvailable": false,
      "IsHolding": false,
      "IsLegSwitch": false,
      "IsPhoto": false,
      "IsPostAscHelical": false,
      "IsPostDescHelical": false,
      "IsPreAscHelical": false,
      "IsPreDescHelical": false,
      "IsSerialisationOk": true,
      "Position": {
        "Latitude": {
          "Value": 0,
          "Type": 0
        },
        "Longitude": {
          "Value": 0,
          "Type": 1
        }
      },
      "RelativeAltitude": {
        "Value": 80
      }
    }
  },
  "MultipleZones": {
    "Position": {
      "Latitude": {
        "Value": 0,
        "Type": 0
      },
      "Longitude": {
        "Value": 0,
        "Type": 1
      }
    },
    "Zones": []
  },
  "SecondaryWaypointsList": [],
  "SelectedFlyZoneRadius": {
    "Value": 500
  },
  "SpecialControlType": 0,
  "Takeoff": {

```

```

    "Type": 1,
    "Visibility": 0,
    "Waypoint": {
      "ASLAltitude": {
        "Value": 84
      },
      "IsASLAltitudeAvailable": true,
      "IsHolding": false,
      "IsLegSwitch": false,
      "IsPhoto": false,
      "IsPostAscHelical": false,
      "IsPostDescHelical": false,
      "IsPreAscHelical": false,
      "IsPreDescHelical": false,
      "IsSerialisationOk": true,
      "Position": {
        "Latitude": {
          "Value": 44.436919187963326,
          "Type": 0
        },
        "Longitude": {
          "Value": 26.046506069244288,
          "Type": 1
        }
      },
      "RelativeAltitude": {
        "Value": 0
      }
    }
  },
  "WaypointsEditionMode": 0,
  "WaypointsEditionRights": 2,
  "WaypointsList": [{
    "Type": 5,
    "Visibility": 0,
    "Waypoint": {
      "ASLAltitude": {
        "Value": 209.64268924123138
      },
      "IsASLAltitudeAvailable": true,
      "IsHolding": false,
      "IsLegSwitch": false,
      "IsPhoto": true,
      "IsPostAscHelical": false,
      "IsPostDescHelical": false,
      "IsPreAscHelical": true,
      "IsPreDescHelical": false,
      "IsSerialisationOk": true,
      "Position": {
        "Latitude": {
          "Value": 44.43619389052219,
          "Type": 0
        }

```

```

    },
    "Longitude": {
      "Value": 26.044286226422628,
      "Type": 1
    }
  },
  "RelativeAltitude": {
    "Value": 125.64268924123137
  }
}, {
  "Type": 0,
  "Visibility": 0,
  "Waypoint": {
    "ASLAltitude": {
      "Value": 212.8370421690785
    },
    "IsASLAltitudeAvailable": true,
    "IsHolding": false,
    "IsLegSwitch": false,
    "IsPhoto": false,
    "IsPostAscHelical": false,
    "IsPostDescHelical": false,
    "IsPreAscHelical": false,
    "IsPreDescHelical": false,
    "IsSerialisationOk": true,
    "Position": {
      "Latitude": {
        "Value": 44.435617857565489,
        "Type": 0
      },
      "Longitude": {
        "Value": 26.044255198332621,
        "Type": 1
      }
    },
    "RelativeAltitude": {
      "Value": 128.8370421690785
    }
  }
}, {
  "Type": 6,
  "Visibility": 0,
  "Waypoint": {
    "ASLAltitude": {
      "Value": 229
    },
    "IsASLAltitudeAvailable": true,
    "IsHolding": false,
    "IsLegSwitch": false,
    "IsPhoto": false,
    "IsPostAscHelical": false,

```

```

    "IsPostDescHelical": true,
    "IsPreAscHelical": false,
    "IsPreDescHelical": false,
    "IsSerialisationOk": true,
    "Position": {
      "Latitude": {
        "Value": 44.435329838934479,
        "Type": 0
      },
      "Longitude": {
        "Value": 26.044767161836262,
        "Type": 1
      }
    },
    "RelativeAltitude": {
      "Value": 145
    }
  }
}, {
  "Type": 0,
  "Visibility": 0,
  "Waypoint": {
    "ASLAltitude": {
      "Value": 190.86415642521524
    },
    "IsASLAltitudeAvailable": true,
    "IsHolding": false,
    "IsLegSwitch": false,
    "IsPhoto": true,
    "IsPostAscHelical": false,
    "IsPostDescHelical": false,
    "IsPreAscHelical": false,
    "IsPreDescHelical": false,
    "IsSerialisationOk": true,
    "Position": {
      "Latitude": {
        "Value": 44.435108285193337,
        "Type": 0
      },
      "Longitude": {
        "Value": 26.045573892180073,
        "Type": 1
      }
    },
    "RelativeAltitude": {
      "Value": 106.86415642521523
    }
  }
}, {
  "Type": 0,
  "Visibility": 0,
  "Waypoint": {

```

```

    "ASLAltitude": {
      "Value": 219.82060938468877
    },
    "IsASLAltitudeAvailable": true,
    "IsHolding": false,
    "IsLegSwitch": false,
    "IsPhoto": false,
    "IsPostAscHelical": false,
    "IsPostDescHelical": false,
    "IsPreAscHelical": false,
    "IsPreDescHelical": false,
    "IsSerialisationOk": true,
    "Position": {
      "Latitude": {
        "Value": 44.435097207519107,
        "Type": 0
      },
      "Longitude": {
        "Value": 26.046334080405018,
        "Type": 1
      }
    },
    "RelativeAltitude": {
      "Value": 135.82060938468877
    }
  }, {
    "Type": 6,
    "Visibility": 0,
    "Waypoint": {
      "ASLAltitude": {
        "Value": 194.39671327778396
      },
      "IsASLAltitudeAvailable": true,
      "IsHolding": false,
      "IsLegSwitch": false,
      "IsPhoto": true,
      "IsPostAscHelical": false,
      "IsPostDescHelical": false,
      "IsPreAscHelical": false,
      "IsPreDescHelical": true,
      "IsSerialisationOk": true,
      "Position": {
        "Latitude": {
          "Value": 44.435119362911962,
          "Type": 0
        },
        "Longitude": {
          "Value": 26.047187352933065,
          "Type": 1
        }
      }
    }
  },

```

```

    "RelativeAltitude": {
      "Value": 110.39671327778396
    }
  }, {
    "Type": 0,
    "Visibility": 0,
    "Waypoint": {
      "ASLAltitude": {
        "Value": 218.60328672221607
      },
      "IsASLAltitudeAvailable": true,
      "IsHolding": false,
      "IsLegSwitch": false,
      "IsPhoto": false,
      "IsPostAscHelical": false,
      "IsPostDescHelical": false,
      "IsPreAscHelical": false,
      "IsPreDescHelical": false,
      "IsSerialisationOk": true,
      "Position": {
        "Latitude": {
          "Value": 44.435174751427155,
          "Type": 0
        },
        "Longitude": {
          "Value": 26.048164737788966,
          "Type": 1
        }
      }
    },
    "RelativeAltitude": {
      "Value": 134.60328672221607
    }
  }, {
    "Type": 0,
    "Visibility": 0,
    "Waypoint": {
      "ASLAltitude": {
        "Value": 207.15234506994057
      },
      "IsASLAltitudeAvailable": true,
      "IsHolding": false,
      "IsLegSwitch": false,
      "IsPhoto": false,
      "IsPostAscHelical": false,
      "IsPostDescHelical": false,
      "IsPreAscHelical": false,
      "IsPreDescHelical": false,
      "IsSerialisationOk": true,
      "Position": {
        "Latitude": {

```



```

        "Value": 44.435640012807482,
        "Type": 0
    },
    "Longitude": {
        "Value": 26.048971468165405,
        "Type": 1
    }
},
"RelativeAltitude": {
    "Value": 123.15234506994055
}
}
}, {
    "Type": 6,
    "Visibility": 0,
    "Waypoint": {
        "ASLAltitude": {
            "Value": 178.1900693314231
        },
        "IsASLAltitudeAvailable": true,
        "IsHolding": false,
        "IsLegSwitch": false,
        "IsPhoto": true,
        "IsPostAscHelical": false,
        "IsPostDescHelical": false,
        "IsPreAscHelical": false,
        "IsPreDescHelical": true,
        "IsSerialisationOk": true,
        "Position": {
            "Latitude": {
                "Value": 44.436260355521256,
                "Type": 0
            },
            "Longitude": {
                "Value": 26.048723243411587,
                "Type": 1
            }
        }
    },
    "ZoomLevel": 15.862750900593744
}
}

```

```

        "RelativeAltitude": {
            "Value": 94.1900693314231
        }
    }, {
        "Type": 0,
        "Visibility": 0,
        "Waypoint": {
            "ASLAltitude": {
                "Value": 210.6033546751824
            },
            "IsASLAltitudeAvailable": true,
            "IsHolding": false,
            "IsLegSwitch": false,
            "IsPhoto": false,
            "IsPostAscHelical": false,
            "IsPostDescHelical": false,
            "IsPreAscHelical": false,
            "IsPreDescHelical": false,
            "IsSerialisationOk": true,
            "Position": {
                "Latitude": {
                    "Value": 44.436725608258143,
                    "Type": 0
                },
                "Longitude": {
                    "Value": 26.048025111366996,
                    "Type": 1
                }
            },
            "RelativeAltitude": {
                "Value": 126.6033546751824
            }
        }
    },
    "ZoomLevel": 15.862750900593744
}
}

```