

POLITECHNIKA WARSZAWSKA

Instytut Elektrotechniki Teoretycznej i Systemów
Informacyjno-Pomiarowych

PRACA DYPLOMOWA MAGISTERSKA

na kierunku INFORMATYKA



Piotr Błądek

Nr ind. 252536

Wykorzystanie internetu rzeczy oraz bezzałogowych statków powietrznych przy wspomaganiu prowadzenia akcji ratunkowej

Zakres pracy:

1. Internet rzeczy
2. Bezzałogowe statki powietrzne
3. Zadanie bezzałogowych statków powietrznych w ratownictwie
4. Architektura systemu
5. Problemy praktyczne zaobserwowane podczas testów

(Podpis i pieczętka kierownika zakładu dydaktycznego)

Kierujący pracą: dr hab. inż. Piotr Biczel

Termin wykonania: 15 grudnia 2017

Praca wykonana i zaliczona pozostaje własnością instytutu
i nie będzie zwrócona wykonawcy.

Warszawa, dnia roku
Politechnika Warszawska
Wydział Elektryczny

OŚWIADCZENIE

Świadomy odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa magisterska pt. *Wykorzystanie internetu rzeczy oraz bezzałogowych statków powietrznych przy wspomaganiu prowadzenia akcji ratunkowej*:

- została napisana przeze mnie samodzielnie
- nie narusza niczych praw autorskich
- nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej.

Jestem świadom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

(data i podpis dyplomanta)

Streszczenie

Temat pracy:

Wykorzystanie internetu rzeczy oraz bezzałogowych statków powietrznych przy wspomaganiu prowadzenia akcji ratunkowej

Niniejsza praca przedstawia proces tworzenia systemu wykorzystującego internet rzeczy i bezzałogowe statki powietrzne, przy wspomaganiu akcji ratowniczej. Zaczynam od wstępu teoretycznego, opisu technologii i sposobu w jaki chcę połączyć obie technologie, później przechodzę do swojej implementacji, opisu zastosowanych przez mnie rozwiązań, problemów i ich rozwiązań, na koniec przedstawiam wyniki testów systemu i wnioski jakie wyciągnąłem z tego projektu.

Temat jest trudny, trzeba połączyć dwie, dość nowe technologie, bezzałogowe statki powietrzne oraz internet rzeczy, tak żeby ze sobą współpracowały tworząc system, który z założenia ma pomagać ratownikom. Po drodze napotkałem wiele problemów, począwszy od bezpieczeństwa, przez autonomiczne sterowanie, krótki czas lotu UAV itd. Starałem się opisać jak sobie z nimi radziłem.

W procesie tworzenia systemu powstało kilka programów, program na system Android, został zmodyfikowany program kontrolera lotu Pixhawk, oraz program stacji naziemnej w framework'u Django. Nie załączam tutaj kodu źródłowego, jest on natomiast dostępny w publicznych repozytoriach na platformie Github <https://github.com/bladekp>.

Na koniec opisuję testy systemu w warunkach symulowanej akcji ratunkowej, które miały miejsce podczas konkursu "Droniada". Konkurs wypadł dobrze, natomiast samo zastosowanie internetu rzeczy i UAV przy prowadzeniu akcji ratunkowej, nie jest moim zdaniem dobrym pomysłem, o czym szerzej piszę na samym końcu pracy.

Słowa kluczowe: IoT, UAV, RSSI, beacon, dron

(data i podpis dyplomanta)

(data i podpis opiekuna)

Abstract

Topic:

Use of internet of things and unmanned aerial vehicles as assistance in the rescue operations

This paper describes the process of creating a system, using the internet of things, and unmanned aerial vehicles in support of the rescue operations. I start with theoretical introduction, a description of the technologies, and the way I combine both technologies, then I dive into my implementation, description of used solutions, problems, at the end I present the results of whole system tests and conclusions that I have learned from this project.

The topic is difficult, I need to combine two, fairly new technologies, unmanned aerial vehicles and the internet of things, to create one system, which is intended, to help people. I encountered a lot problems on my way, starting from security, autonomous flight, UAV's short flight time etc. I tried to describe, how I dealt with them.

In the process of creating the system, several programs were created, the program for Android operating system, program for Pixhawk flight controller was modified, and ground station program in the Django framework was done from scratch. I do not include the source code here, but everything is available online in public Github repositories <https://github.com/bladekp>.

Finally, I describe the system tests under the simulated rescue operation conditions, these tests take place during the 'Droniada' competition. The competition went well, but use of the internet of things and UAVs in rescue operations, is not good idea in my opinion, as I am writing at the very end of the paper.

Keywords: IoT, UAV, RSSI, beacon, drone

(date and signature of graduate student)

(date and signature of mentor)

Spis treści

Wykaz ważniejszych skrótów i oznaczeń	viii
1. Wstęp	10
1.1. Cel i zakres pracy	10
1.2. Tło projektu	10
1.3. Wprowadzenie do zagadnienia	10
2. Internet rzeczy	12
2.1. Koncepcja pomysłu internetu rzeczy	12
2.2. Infrastruktura, sprzęt i BLE [5]	14
2.3. Komponenty IoT wykorzystane w tworzonym systemie	15
3. Bezzałogowe statki powietrzne	16
3.1. Czym są UAV	16
3.2. Rola UAV w projektowanym systemie	17
3.3. Misja autonomiczna	19
3.4. Bezpieczeństwo stosowania UAV w misji ratunkowej	20
4. Koncepcja techniczna systemu	22
4.1. Schemat blokowy rozwiązania	22
4.2. Algorytm pracy	25
4.3. Podłączenie wielu statków do jednego systemu	26
4.4. Sterowanie autonomiczne	27
4.5. Bezpieczeństwo użytkowania	28
4.6. Aspekty mechaniczne systemu	28
5. Problemy i próba ich rozwiązania	29
5.1. Krótki czas lotu i zasięg UAV	29
5.2. Zawodność linku telemetrycznego	30
5.3. Wrażliwość systemu na sytuacje wyjątkowe	31
5.4. Problemy sprzętowe	31
5.5. Sygnał beacona	31
6. Testy i misja konkursowa	33

7. Podsumowanie	35
7.1. Wnioski	35
7.2. Plany na przyszłość	37
Literatura	39
Spis rysunków	40
Spis tablic	41

Wykaz ważniejszych skrótów i oznaczeń

ADS-B	System automatycznego nadzoru - rozgłoszeniowy (ang. Automatic dependent surveillance – broadcast) - system podający położenie „własnego” statku powietrznego innym statkom powietrznym oraz kontroli ruchu lotniczego (ADS-B out), a także odbierający sygnały od innych uczestników ruchu lotniczego.
BVLOS	(ang. Beyond Visual Line of Sight) - licencja uprawniająca do lotów poza zasięgiem wzroku operatora.
EASA	Europejska Agencja Bezpieczeństwa Lotniczego (ang. European Aviation Safety Agency) - agencja zajmująca się problemami bezpieczeństwa lotniczego w Europie.
ICAO	Organizacja Międzynarodowego Lotnictwa Cywilnego (ang. International Civil Aviation Organization) - agencja odpowiedzialna za wdrażanie międzynarodowych przepisów regulujących bezpieczeństwo w ruchu powietrznym.
IoE	Internet wszechrzeczy (ang. Internet of Everything) - koncepcja która rozszerza internet rzeczy (IoT) i kładzie nacisk na komunikację maszyna-maszyna, opisując bardziej skomplikowany system który obejmuje również ludzi i procesy.
IoT	Internet rzeczy (ang. Internet of Things) - koncepcja, wedle której przedmioty mogą gromadzić, przetwarzać, oraz wymieniać dane za pośrednictwem szeroko rozumianej sieci.
PAŻP	Polska Agencja Żeglugi Powietrznej (ang. Polish Air Navigation Services Agency PANSA) - agencja zarządza przestrzenią powietrzną, przepływem ruchu lotniczego i zapewnieniem służb ruchu lotniczego na terenie Polski.
RPA	Samolot pilotowany zdalnie (ang. Remotely-piloted aircraft)
RSSI	Wskaźnik siły sygnału w urządzeniu odbiorczym (ang. Received Signal Strength Indication) [16].
UAS	Bezzałogowe systemy statków powietrznych (ang. Unmanned Aircraft Systems) - szeroko rozumiane systemy obsługujące statki powietrzne, działające na pokładzie statków powietrznych w całości lub części.

UAV	Bezzałogowy statek powietrzny (ang. Unmanned Aerial Vehicle) - dron, który nie wymaga do lotu załogi obecnej na pokładzie oraz nie ma możliwości zabierania pasażerów.
UAVO	Operator bezzałogowego statku powietrznego (ang. Unmanned Aerial Vehicle Operator) - w polskim prawie oznacza świadectwo kwalifikacji operatorów dronów, uprawniające do ich wykorzystywania w celach innych niż loty sportowe i rekreacyjne.
VLOS	(ang. Visual Line of Sight) - licencja uprawniająca do lotów w zasięgu wzroku poz warunkiem zachowania bezpiecznej odległości i odpowiedniego oznakowania. Dron musi mieć tabliczkę identyfikacyjną, a operator kamizelkę.

1. Wstęp

1.1. Cel i zakres pracy

Sprawdzenie możliwości wykorzystania internetu rzeczy, oraz bezzałogowych statków powietrznych, przy wspomaganiu prowadzenia akcji ratowniczej. Fizyczne zaprojektowanie i wykonanie kompletnego systemu, części działającej na pokładzie statku powietrznego, oraz części działającej na ziemi. Przetestowanie systemu w warunkach symulowanej akcji ratunkowej. Potwierdzenie lub zaprzeczenie użyteczności projektowanego systemu, wykazanie wad i zalet systemu.

1.2. Tło projektu

Pomysł projektu nasunęli organizatorzy konferencji "Parada Robotów 2017" i zarazem konkursu "Droniada" [1]. Zadaniem uczestników konkursu było wspomaganie ewakuacji medycznej 10 osób po ataku huraganowego wiatru, przy użyciu bezzałogowców w locie autonomicznym, lub ewentualnie półautonomicznym, oraz tzw. beaconów [2]. W tym dokumencie opiszę system lokalizowania poszkodowanych osób, który stworzyłem na potrzeby wspomnianego wyżej konkursu. Przejdę od projektu systemu, przez jego implementację, testy, oraz to jak sprawdził się w zadaniu konkursowym, na koniec przedstawię wnioski jakie wyciągnąłem z tego projektu i czy może on mieć jakąkolwiek przyszłość komercyjną. Poniżej na zdjęciu znajduje się część naszego zespołu która brała udział w konkursie "Droniada", wraz z płatowcem użytym w misji konkursowej 1.1.

1.3. Wprowadzenie do zagadnienia

Projektowany system składał się z kilku części, schemat ideowy systemu przedstawia rysunek 1.2. Część pierwsza to poszkodowane w katastrofie osoby, a właściwie beacony które ratownicy medyczni wręczyli poszkodowanym osobom, beacony nadają uzgodniony sygnał reprezentujący pewien kolor, i tak kolor czerwony jest przyporządkowany do osób co do których wymagana jest natychmiastowa ewakuacja, kolorem żółtym oznaczone zostają osoby, które wymagają pilnej ewakuacji (mniejszy priorytet), kolorem zielonym osoby które chodzą o własnych siłach, natomiast kolorem czarnym zgony, osoby których ciała zostaną usunięte dopiero po zakończeniu akcji ratowania życia. Jest to tzw. system znaczników Triage [3], rysunek 1.3 przedstawia przykładowy, papierowy, używany przez ratowników znacznik systemu Triage. Moim zadaniem jest próba udowodnienia że taki papierowy znacznik można zamienić na beacon, który nadając dookolnie zakodowany sygnał pomoże ratownikom w akcji ratunkowej.

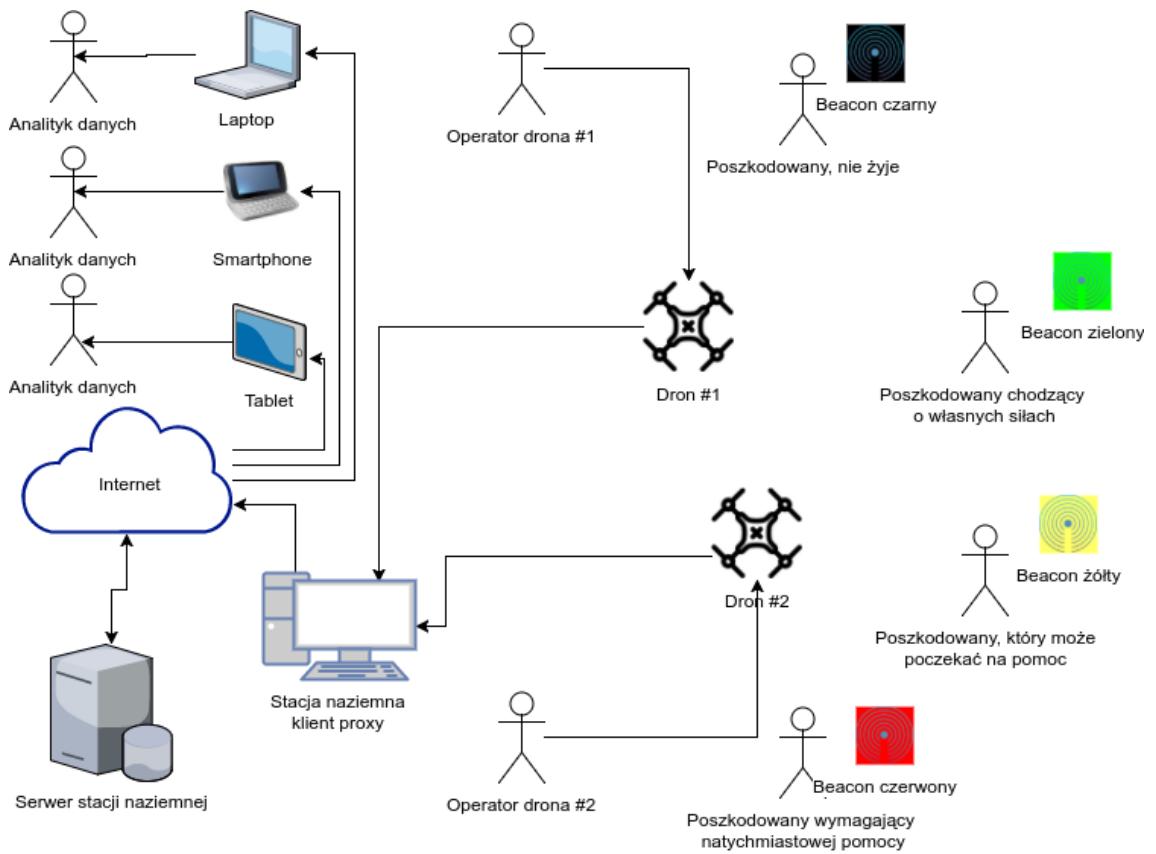


Rys. 1.1. Część naszego zespołu na konkursie "Droniada", osoby które brały udział w konkursie głównym

Kolejnym komponentem systemu był statek powietrzny, który miał za zadanie zbieranie nadawanych przez beacony sygnałów i umiejscawianie ich na płaszczyźnie mapy. Taki statek musiał zostać wyposażony w odbiornik bluetooth, w wersji co najmniej 4.0 i odpowiednie oprogramowanie zbierające sygnały.

Ostatnim elementem systemu był system naziemny (ang. Ground Station). System ten zbierał zlokalizowane przez statek powietrzny sygnały i wyświetlał je na mapie, dodatkowo aproksymując pozycję realną beaconów na mapie. System ten składał się z kilku komponentów w architekturze klient-serwer.

Ogólny diagram BPMN mojego rozwiązania przedstawia rysunek 1.4. Na pierwszym torze widzimy beacona, na drugim statek powietrzny, na ostatnich trzech przedstawiona jest stacja naziemna. Stacja naziemna mogła być dowolnie dzielona na fizyczne maszyny, w moim przypadku połączona była stacją naziemną z klientem proxy i z klientem odbiorczym, stacją naziemną z oprogramowaniem serwera była na innej maszynie fizycznej. W kolejnych rozdziałach będę przybliżał kolejne komponenty systemu.



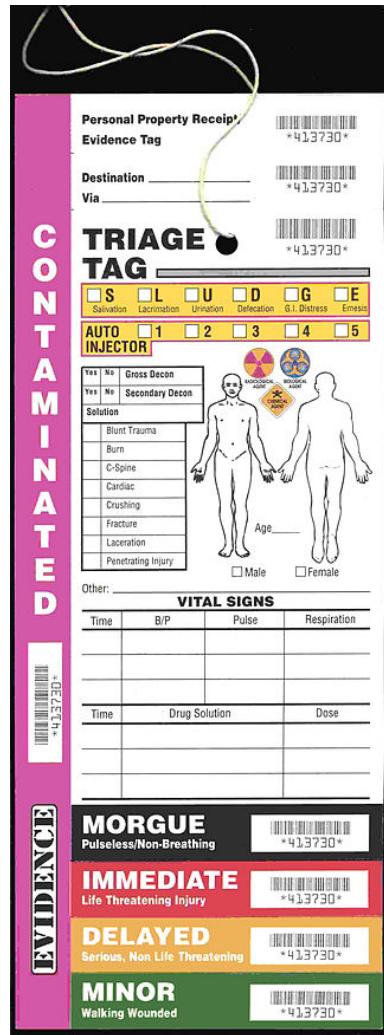
Rys. 1.2. Schemat ideowy systemu, widać na nim wszystkie elementy systemu, począwszy od poszkodowanych, przez drony i ich operatorów, a skończywszy na wszystkich elementach stacji naziemnej i analitykach zebranych danych

2. Internet rzeczy

2.1. Koncepcja pomysłu internetu rzeczy

Internet rzeczy (ang. *Internet of Things - IoT*) to idea która zakłada że wszystko można podłączyć do sieci internet [4], wszystko to znaczy pojazd, budynek i inne rzeczy wyposażone w elektroniczny system wbudowany z sensorami, aktorami oraz interfejs sieciowy, dzięki czemu możliwa jest wymiana danych.

Podstawowym celem idei internetu rzeczy jest stworzenie inteligentnych rzeczy, budynków, miast itp. Takie inteligentne rzeczy połączone w sieci, mogą dać ludzkości możliwość nieustannego monitorowania zmieniających się warunków i przystosowywania tych inteligentnych rzeczy do wymagań użytkowników. Dla przykładu sieć czujników, które zbierają dane pogodowe takie jak temperatura, ciśnienie, wysokość poziomu rzek itp. mogłyby w czasie rzeczywistym wykrywać anomalie i sama na nie reagować. Taka szybka reakcja na zagrożenie, znacząco zwiększa szanse na uratowanie ludzkiego życia i zmniejsza straty materialne.

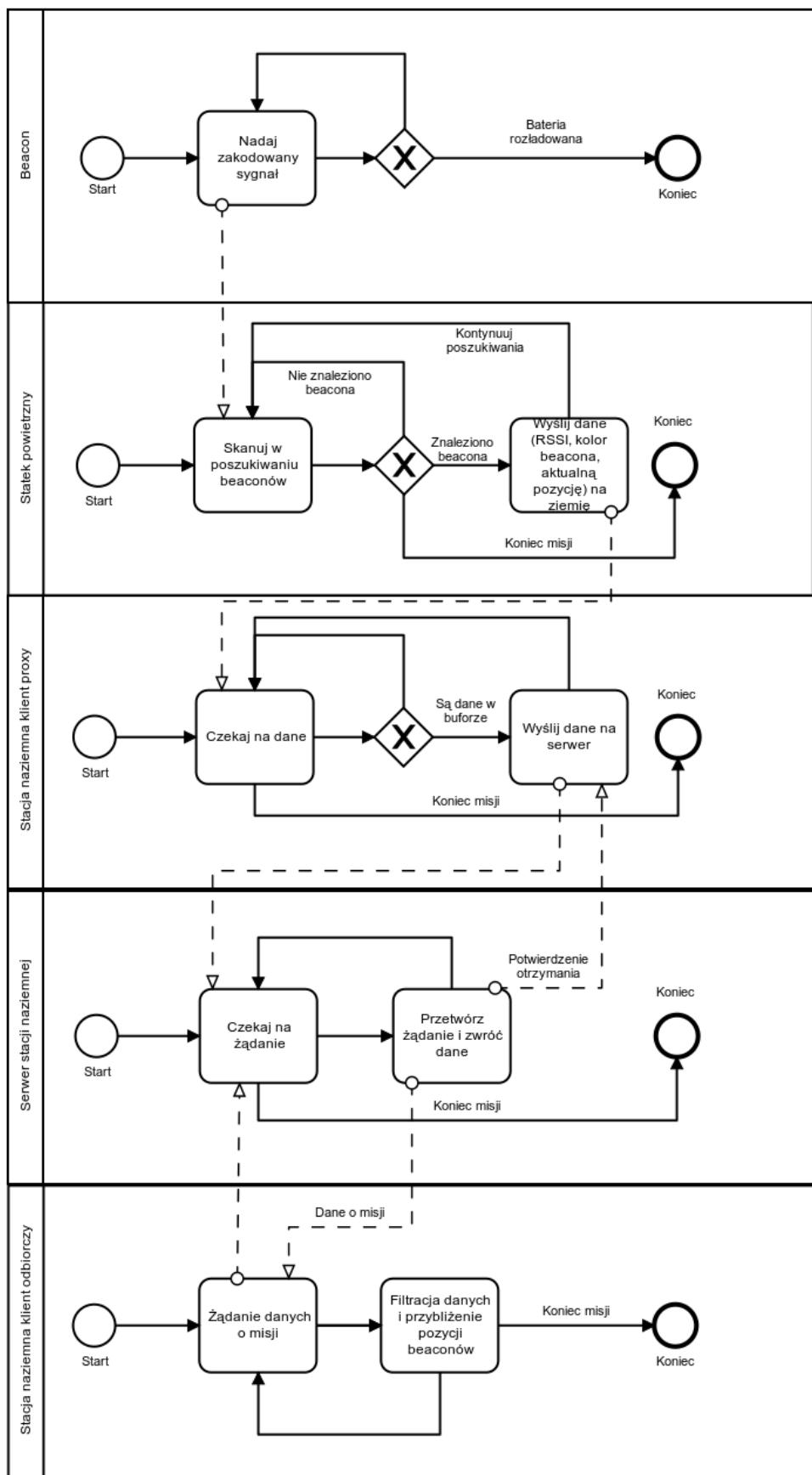


Rys. 1.3. Typowy znaczek systemu Triage, używany w akcjach ratowniczych [3]

2.2. Infrastruktura, sprzęt i BLE [5]

Warstwa odpowiedzialna za zbieranie danych ze świata rzeczywistego, wyposażona jest przede wszystkim w sensory i elektroniczne identyfikatory. Sensory już od bardzo dawna używane są w przemyśle samochodowym, służbie zdrowia, zakładach produkcyjnych czy przemyśle kosmicznym, są coraz tańsze i mniejsze, dzięki temu mogą być dodawane jako standardowe wyposażenie do dowolnych urządzeń. Większość współczesnych telefonów jest wyposażona w akcelerometr, magnetometr, żyroskop, czujnik zbliżenia, czujnik oświetlenia, a nawet czujnik wilgotności, czy barometr. Projekt CeNSE (ang. Central Nervous System of the Earth) ma na celu rozwój technologii w taki sposób, aby umożliwić oczujnikowanie całej naszej planety, tworząc swoisty układ nerwowy [6].

Bardzo ważną warstwą sieci internetu rzeczy, jest warstwa transportowa. Warstwa transportowa łączy wszystkie sensory i aktory do sieci internet. Prowadzenie sieci kablowej jest bardzo drogie i czasochłonne, dlatego urządzenia implementujące



Rys. 1.4. Ogólny diagram BPMN przepływu danych w systemie

ideę internetu rzeczy najczęściej wykorzystują różnorakie technologie bezprzewodowe, na przykład:

- Sieci komórkowe 2G, 3G, 4G - wymagają one doposarzenia urządzenia w kartę SIM
- WiFi - sieci małego zasięgu, dla zapewnienia dostępu do internetu wymagany jest punkt dostępowy
- Bluetooth - protokół dedykowany sieciom osobistym, stosunkowo duże przedkości przesyłu
- ZigBee - niska przepustowość, energooszczędność, spory zasięg w porównaniu z dwiema wyżej wymienionymi pozycjami
- Z-Wave - protokół dla domowej automatyki, wykorzystuje radio małej mocy

Urządzeniem które powstało specjalnie dla internetu rzeczy jest tzw. beacon, jest to urządzenie małej mocy które, w z góry określonych interwałach czasowych, nadaje dookoła ramki danych przy wykorzystaniu bluetooth. Ważne jest że beacon ma zaimplementowany bluetooth w wersji 4.0 (lub wyższej) potocznie nazywany BLE (ang. *Bluetooth Low Energy*). Ramki te, mogą odbierać urządzenia znajdujące się w pobliżu beacona, zasięg typowego beacona to około 100m, na podstawie odebranego sygnału urządzenia mogę podejmować akcje. Ważnym parametrem sygnału nadawanego przez beacony, w naszym przypadku było tzw. RSSI, jest to po prostu miara siły sygnału odebranego przez urządzenie odbiorcze. Im większy parametr RSSI tym mocniejszy sygnał, dzięki temu można, z dość dużym błędem niestety, powiedzieć jak daleko znajdujemy się od nadajnika sygnału.

2.3. Komponenty IoT wykorzystane w tworzonym systemie

W tworzonym systemie wykorzystaliśmy beacony, jako urządzenia którymi oznaczone były osoby poszkodowane. Beacony to małe nadajniki sygnału radiowego, które najczęściej są używane do wysyłania danych na nasze telefony, w ogólności odbiornikiem może być dowolne urządzenie wyposażone w bluetooth w wersji 4.0 lub wyższej. Beacon często nazywany jest latarnią, działa on bowiem jako taka latarnia, zazwyczaj jest umieszczony w jednym, konkretnym miejscu, gdy wejdziemy w zasięg nadawanego przez niego sygnału możemy odbierać dane przez niego nadawane i na nie reagować. Beacony konkursowe nadawały dane przy wykorzystaniu protokołu iBeacon^[7], stworzonego przez firmę Apple, innym równie popularnym protokołem jest Eddystone^[8], jednak ja w tej pracy skupiłem się na protokole iBeacon. To co nas najbardziej interesowało to bajty z danymi Major i Minor, oznaczającymi kolejno kolor i numer beacona według specyfikacji w tabeli 2.1.

Tab. 2.1. Kodowanie kolorów na dane Minor i Major

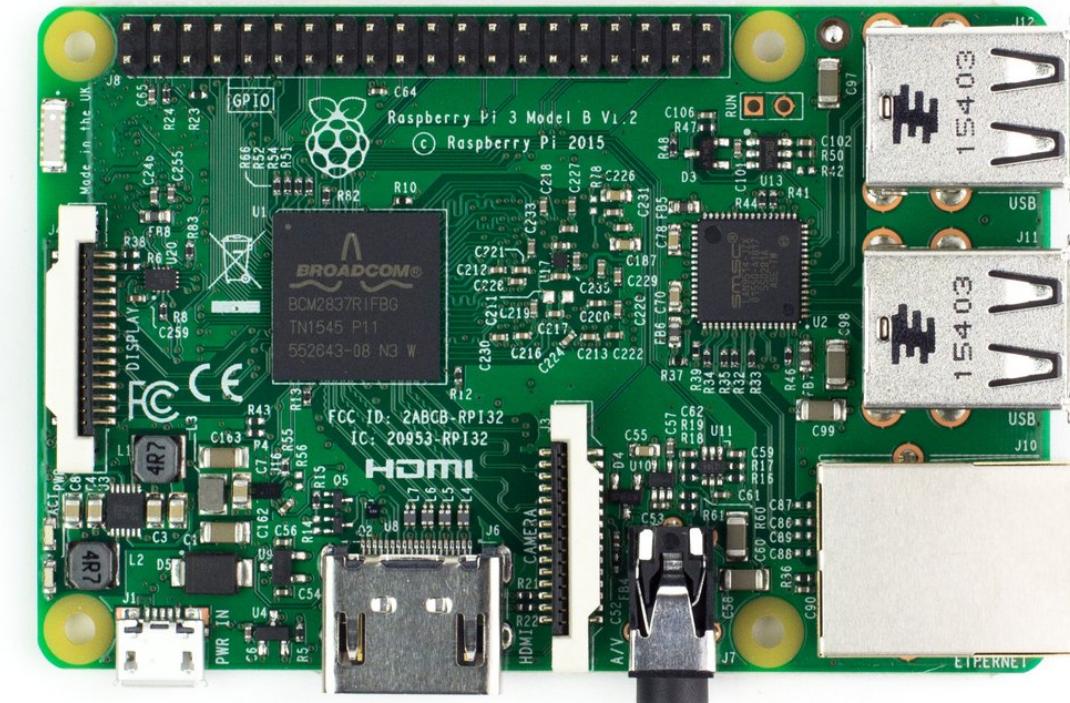
Major.Minor	Kolor	Znaczenie
1.1	Czarny	Katgoria BLACK, stwierdzono zgon, ciało zostanie zabrane po akcji ratunkowej.
2.1 2.2	Zielony	Odpowiednik kategorii GREEN, poszkodowany chodzi o własnych siłach - zdalny nadzór.
3.1 3.2 3.3	Żółty	Odpowiednik kategorii YELLOW, pilna ewakuacja medyczna.
4.1 4.2 4.3 4.4	Czerwony	Odpowiednik kategorii RED, natychmiastowa ewakuacja medyczna poszkodowanego.

Do odbioru sygnału beaconów, organizatorzy konkursu proponowali użycie gatewya produkcji firmy kontakt.io [9]. Problem był taki że wspomnianego gatewya trzeba było zintegrować z komputerem pokładowym bezzałogowca, co wydawało się być zadaniem trudnym, ze względu na to że gateway był wyposażony w interfejsy głównie bezprzewodowe, co mijało się z celem przy kilkucentymetrowych odległościach. Postanowiłem więc użyć do tego celu komputera Raspberry Pi w wersji 3 model B 2.1 [11], który to wspiera bluetooth 4.1, jest stosunkowo małych rozmiarów (56,5mm x 85,6mm x 17mm), niewielkiej masy (45g), dodatkowo jest dość tani, wszystkie te cechy idealnie pasują do urządzenia, które można podłączyć do bezzałogowego statku powietrznego. Na Raspberry Pi zainstalowałem system RTAndroid[10], tak żeby możliwe było skorzystanie z API ułatwiającego komunikację pomiędzy beaconami i odbiornikami, utrzymywanyego przez firmę kontakt.io, dzięki temu API pisanie aplikacji odbierającej dane było o wiele prostsze.

3. Bezzałogowe statki powietrzne

3.1. Czym są UAV

Bezzałogowe statki powietrzne, UAV (ang. *Unmanned Aerial Vehicle*) potocznie nazywane dronami to statki powietrzne, które nie wymagają do lotu załogi obecnej na pokładzie, oraz nie mają możliwości zabierania pasażerów, pilotowane są zdalnie lub wykonują lot autonomiczny. Niegdyś wykorzystywane jedynie w wojsku, stały się jednym z symboli walki z terroryzmem, pionierami w tego typu konstrukcjach były Stany Zjednoczone oraz Izrael. Ostatnio drony za sprawą spadającej ceny kom-



Rys. 2.1. Komputer Raspberry Pi 3 Model B, użyty w moim projekcie w charakterze odbiornika bluetooth danych z beaconów

ponentów z których są konstruowane, stały się dostępne praktycznie dla każdego, najczęstsze zastosowania dronów to fotografowanie i filmowanie z powietrza imprez masowych, wesel, promocja firm, turystyka, fotografowanie nieruchomości z powietrza, inwestycji typu droga szbkiego ruchu, most. Poza branżą zdjeciową, która w tym momencie jest głównym powodem zakupu dronów przez małe firmy usługowe, ale też osoby prywatne, drony znajdują zastosowanie w geodezji, ochronie, straży pożarnej, interesują się nimi firmy kurierskie, dostawcy jedzenia, w leśnictwo i wiele więcej. Drony posiadają możliwość zamontowania na ich pokładzie różnorodnych sensorów, gdzie masa sensora jest oczywiście ograniczeniem, ale dron do kwoty którą jest za niego w stanie zapłacić statystyczny Kowalski może unieść już pokaźnych rozmiarów sensory. Również mój projekt zakładał montaż takich dodatkowych sensorów na pokładzie statków powietrznych. Mój plan zakładał użycie trzech statków

powietrznych do wykonania misji, jednym z nich był płatowiec widoczny na zdjęciu 3.1, dwa pozostałe to czterowirnikowce, jeden komercyjny Phantom 3, drugi zbudowany przez nasz zespół od podstaw 3.2.



Rys. 3.1. Nasz płatowiec, na bazie kadłuba skyhunter, podczas wykonywania misji konkursowej

3.2. Rola UAV w projektowanym systemie

Zadaniem statków powietrznych, które wykorzystaliśmy w projekcie było znalezienie w terenie nadajników sygnału bluetooth. Teren był z góry określony i wyznaczał go wielokąt wypukły zaprezentowany na rysunku 3.3. Do odbioru danych, tak jak pisałem wcześniej, użyliśmy Raspberry Pi 3 Model B, po jednym komputerze do każdego statku powietrznego, mieliśmy trzy statki, więc były trzy komputery Raspberry Pi. Statki powietrzne musiały się poruszać w taki sposób, żeby po pierwsze pokryć swoim przelotem cały teren, a po drugie, zmieścić się w wyznaczonym na 30 minut oknie czasowym misji. Ważne było żeby pozycje znalezionych beaconów podać z dużą dokładnością, błąd powyżej kilku metrów oznaczał stratę punktów. Nasz plan zakładał rozpoczęcie misji płatowcem, z zaplanowaną misją, która pokrywałaby cały teren poszukiwań, do momentu znalezienia pierwszego beacona przez płatowiec, czterowirnikowce miały pozostać uziemione, w momencie nadania informacji o znalezieniu pierwszego beacona, pierwszy z czterowirnikowców podrywałyby się i kierował w pobliże miejsca znalezienia pierwszego beacona, po zadowalającym



Rys. 3.2. Nasz czterowirnikowiec, zbudowany od podstaw, wyposażony w kontroler lotu Pixhawk

dla nas przybliżeniu pozycji pierwszego beaconsa, czterowirnikowiec poruszałby się do kolejnego znalezioneego beaconsa. Drugi czterowirnikowiec był swego rodzaju za- bezpieczeniem gdyby pierwszy uległ awarii, lub gdyby pierwszy nie dawał sobie rady ze znajdowaniem kolejnych beaconsów, czterowirnikowiec był sterowany manualnie przez wyszkolonego operatora, posiadającego świadectwo kwalifikacji UAVO.

3.3. Misja autonomiczna

Jednym z kryteriów oceny stosowanym przez jury konkursowe, było wykonanie misji w trybie autonomicznym, to znaczy że statkami powinien sterować program komputerowy, nie człowiek. Planowanie takiej misji nie jest większym problemem, jako że stosowany przez nas kontroler lotu, Pixhawk, miał możliwość zaprogramowania mu takiej właśnie misji, planowanie misji polega na podaniu programowi punktu startu, punktu końca, oraz punktów pośrednich z wysokościami barometrycznymi, czyli wysokościami nad poziomem gruntu, na jakich ma się odbywać misja. Przykład takiej misji znajduje się na rysunku 3.4. My wykonywaliśmy misję w trybie częściowo autonomicznym a częściowo manualnym, pątowiec miał wgraną misję, natomiast czterowirnikowce były sterowane przez operatorów. Początkowo były plany wykonania całej misji autonomiczne, byłoby to możliwe przy wykorzystaniu narzędzia dronekit [12], które to udostępnia wygodny interfejs programistyczny ułatwiający



Rys. 3.3. Mapa terenu w obrębie którego mogły znajdować się beacony konkursowe, pomarańczową linią wyznaczona jest granica zewnętrzna terenu

sterowanie statkami powietrznymi. Wymagałoby to jednak ode mnie napisania programu, który w odpowiedzi na znalezienie beacona przez pilotowca, wysyłałby w odpowiednie miejsce czterowirnikowiec w celu dokładniejszego zbadania pozycji beacona. Było to możliwe, ale bardzo trudne, ponieważ i tak miałem sporo pracy z innymi komponentami systemu, porzuciłem tą część projektu na samym początku. Program taki wymagałby ponadprzeciętnej obsługi sytuacji wyjątkowych, które przy takich misjach pojawiają się bardzo często (mocniejszy podmuch wiatru, przeszkoda statyczna na przykład drzewo, przeszkoda dynamiczna na przykład człowiek lub ptak) a to z kolei wymagałoby setek godzin testów i dodatkowego oprzyrządowania statku powietrznego (czujniki zbliżeniowe, zwiększenie widoczności poprzez dodatkowe lampki i ewentualne sygnały dźwiękowe w celu ostrzeżenia ludzi i odstraszenia ptaków). W rezultacie takiego programu sterującego czterowirnikowcem, nie otrzymalibyśmy niestety niczego lepszego niż jest nam w stanie zapewnić operator, żeby program przerósł operatora w tej kwestii potrzebne byłoby tysiące godzin testów, których nie byliśmy w stanie zapewnić.

3.4. Bezpieczeństwo stosowania UAV w misji ratunkowej

Może się niestety tak zdarzyć że w misji ratunkowej ucierpi sam ratownik, albo jego sprzęt, mieliśmy przykład takiego incydentu podczas samego konkursu, gdzie jednej z drużyn zapaliła się bateria zainstalowana w statku powietrznym, przez co



Rys. 3.4. Przykład misji autonomicznej w programie Mission Planner

statek stanął w ogniu, sytuację przedstawiają zdjęcia 3.5 oraz 3.6.



Rys. 3.5. Dron jednej z drużyn spłonął na zawodach

Dużą wagę trzeba przyłożyć do bezpieczeństwa samego systemu, po to żeby to ratownik ratował, a żeby sam nie potrzebował pomocy. Rynek dronowy to stosunkowo nowa dziedzina i jest jeszcze dużo do zrobienia jeżeli chodzi o bezpieczeństwo, które często odkładane jest na dalszy plan, a w konsekwencji w ogóle pomijane przy



Rys. 3.6. To co zostało z drona po pożarze baterii

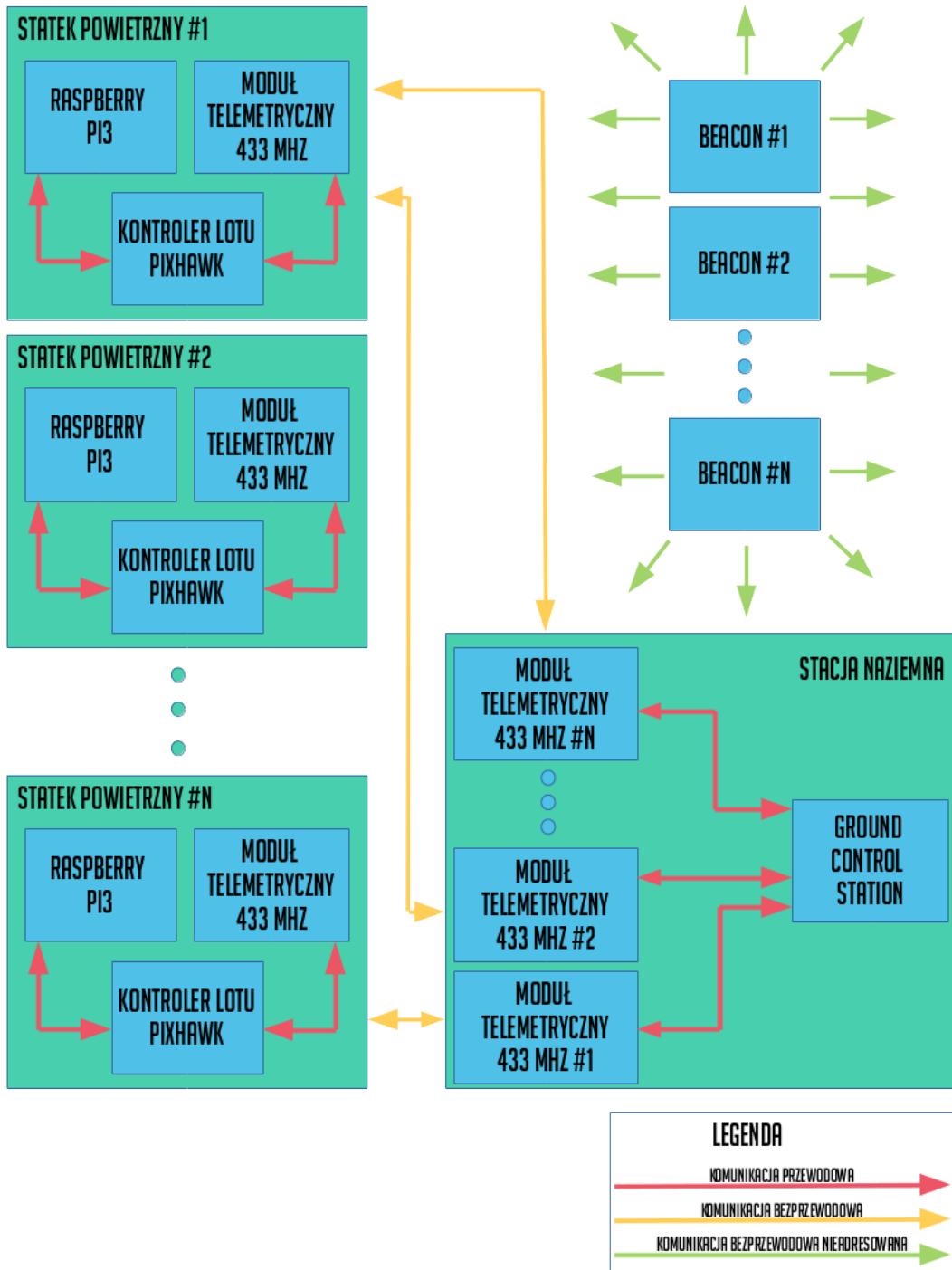
projektowaniu systemów. Należy zadbać o to żeby operatorzy dronów byli dobrze widoczni, na przykład powinni mieć na sobie kamizelki w jaakrawym kolorze. Sprzęt powinien być sprawdzony przed startem, przy pomocy tzw. checklisty, gdyby drużyna której spłonął dron miała taką checklistę, może nic by się nie stało. Ważne jest też żeby operator miał ważne świadectwo kwalifikacji UAVO. Ale najważniejszy jest sam system. W dronie który spłonął prawdopodobnie brakowało systemu ostrzegania przed przegrzaniem baterii, gdyby taki system zadziałał, prawdopodobnie nic by się nie stało, drużyna nie straciłaby drona, nie doszłoby do niebezpiecznej sytuacji.

4. Koncepcja techniczna systemu

4.1. Schemat blokowy rozwiązania

Na rysunku 4.1 znajduje się schemat blokowy zaproponowanego przeze mnie rozwiązania.

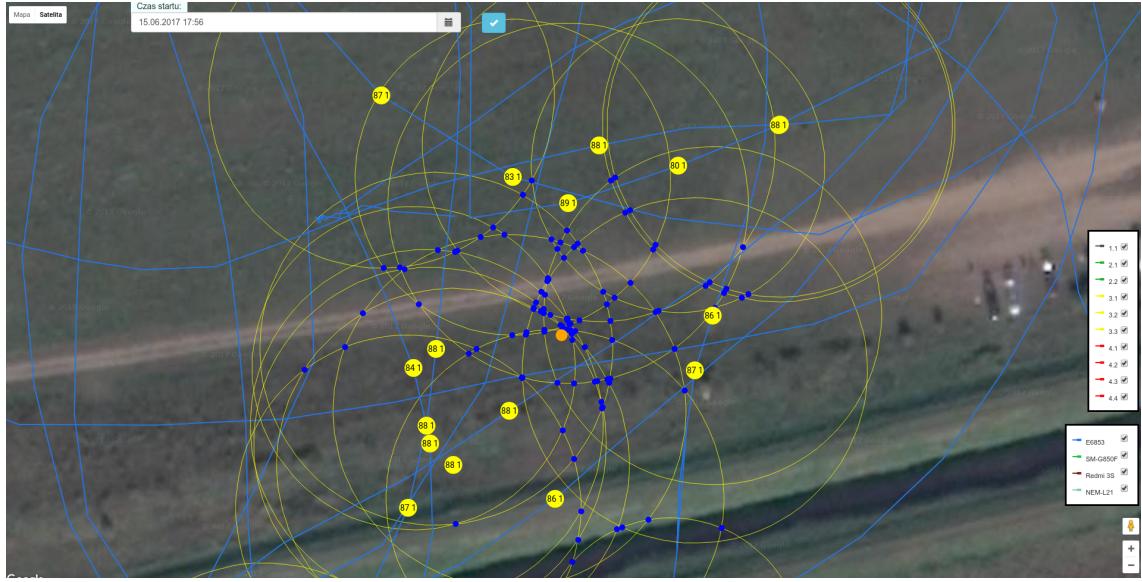
Stacja naziemna, to komputer z zainstalowanym oprogramowaniem i modułami telemetrycznymi, do odbierania komunikatów ze statków powietrznych. Moduły telemetryczne nie bez powodu pracują na częstotliwości $433MHz$, jest to bowiem częstotliwość na użytkowanie której nie potrzebne jest zezwolenie, według rozporządzenia Urzędu Komunikacji Elektronicznej [13]. Stacja naziemna niekoniecznie



Rys. 4.1. Schemat blokowy systemu

jest miejscem przy którym siedzi operator systemu. System został zaprojektowany w architekturze klient-serwer, gdzie funkcje klientów może pełnić dowolne urządzenie, które ma dostęp sieciowy do serwera, przy czym nasz serwer znajdował się w bezpiecznym pomieszczeniu, daleko od miejsca wykonywania misji, podczas gdy my przy pomocy klientów w postaci różnych urządzeń zasilaliśmy serwer danymi, a przy

pomocy innych klientów dane z serwera były pobierane i prezentowane operatorowi w wygodnej i zrozumiałej dla niego postaci, której przykład stanowi rysunek 4.2.



Rys. 4.2. Widok ekranu operatora systemu, w centralnym miejscu mapa z zaznaczoną trasą przelotu, próbkami znalezionejgo sygnału, oraz wyznaczoną aproksymacją

Na rysunku widzimy dokładnie to co widzi operator systemu, zaprezentowany zrzut pochodzi z testów systemu przeprowadzanych jeszcze przed zawodami. Niebieska linia to trasa przelotu statku powietrznego, użytego do testu. Żółte kropki to próbki sygnału jakie udało się zebrać. Na kropkach pokazana jest siła sygnału (RSSI), oraz numer beacona w kolejności (w tym wypadku numer 1). Żółte okręgi wyznaczają, gdzie prawdopodobnie może znajdować się beacon (im słabszy sygnał tym większy okrąg, moja aproksymacja działa podobnie jak ta która działa w systemie GPS). Tam gdzie żółte okręgi się przecinają stawiamy granatowy punkt, ale nie zawsze, wybieramy tylko te punkty które są najbliżej prawdopodobnego rozwiązania, w końcu pomarańczowy punkt stanowi rozwiązanie aproksymacji. W przedstawionym przykładzie system pomylił się około 2 metry na południe, co mogło być wynikiem dokładności systemu GPS i małej ilości próbek sygnału w zaprezentowanym przykładzie.

Dodatkowo operator, po prawej stronie ekranu, ma możliwość włączenia lub wyłączenia ścieżek pojazdów, oraz poszczególnych beaconów, co może być przydatne w przypadku dużej ilości zebranych danych. Na górze jest też możliwość wybrania z systemu danych, od konkretnego momentu w czasie, na przykład możemy wybrać tylko punkty, które zostały utworzone dzisiaj po godzinie 14:00, opcja ta była przydatna podczas przeprowadzania testów.

Wracając do schematu blokowego, kolejnym jego elementem są statki powietrzne, ich ilości w projektowanym systemie nie ograniczało nic. W praktyce oczywiście, ograniczeniem mógłaby być przepustowość łączą jaką dysponowaliśmy, ilość opera-

torów potrzebnych do obsługi statków i na pewno same statki, które nie są przecież tanie. Wymaganie jakie było postawione naszym statkom powietrznym, to posiadanie kontrolera lotu pixhawk, miejsce na umieszczenie modułu Raspberry Pi 3, oraz posiadanie linku telemetrycznego $433MHz$. Od konieczności posiadania Pixhawka poczyniliśmy jednak odstępstwo, ponieważ daliśmy możliwość pracy ze statkiem komercyjnym, jakim jest Phantom 3 **4.3**, nie ma on na wyposażeniu kontrolera lotu Pixhawk, a jedynie co było wymagane żeby Phantom mógł zasilać nasz system danymi, to podczepienie mu urządzenia z zainstalowaną aplikacją, urządzeniem mogło być Raspberry Pi 3, lub na przykład dowolny telefon, wyposażony w bluetooth 4.0 lub większy.



Rys. 4.3. Komercyjny dron Phantom 3, użyty przez nas w misji

Ostatnim elementem zaprezentowanym na schemacie są beacons. Są to urządzenia wyposażone w nadajnik bluetooth, który ma możliwość wysyłania dookolnych, niezaadresowanych danych. Dzięki temu że dane nie są zaadresowane, nie trzeba parować urządzeń (tak jak miało to miejsce przed pojawiением się bluetooth 4.0). Sygnały nadawane przez wspomniane beacons były zbierane przez statki powietrzne, a właściwie przez aplikację zainstalowaną na Raspberry Pi 3, lub innym urządzeniu i wysyłane do stacji naziemnej.

4.2. Algorytm pracy

Algorytm pracy systemu jest następujący. Pierwszym krokiem jest rozmieszczenie włączonych beaconów, nadających dane w ustalonym wcześniej protokole 2.1, na jakimś z góry ustalonym obszarze. Beacons, przypominam symulują osoby które ucierpiały w katastrofie.

Kolejnym krokiem jest uruchomienie klientów, czyli urządzeń które będą sygnały zbierać, i umieszczenie ich na statkach powietrznych.

Następnie podłączamy się klientem który będzie dane odbierał, jako że moja aplikacja została napisana w technologii webowej, klientem odbiorczym mogło być dowolne urządzenie posiadające dostęp do serwera i przeglądarkę internetową, na przykład laptop.

Od tego momentu rozpoczyna się misja właściwa, statki wznoszą się w powietrzu i oblatują wskazany teren, mogą to robić autonomiczne, z pomocą zaprogramowanej wcześniej przez operatora misji, lub operator sam może kierować ich ruchem, można też oczywiście stosować mieszaną taktykę, konfiguracja jest dowolna, statki są od siebie niezależne. Aplikacja zainstalowana na urządzeniu podczepionym do statku, zbiera znalezione beacons i przy pomocy linku telemetrycznego wysyła je do stacji naziemnej, która transmisuje je do serwera, a z serwera dowolny klient odbiorczy może je pobrać.

Klient odbiorczy ma po swojej stronie zaimplementowany algorytm aproksymacji punktu w którym prawdopodobnie znajduje się poszukiwany beacon. Doświadczony operator systemu, empirycznie na podstawie próbek sygnałów i aproksymacji, wybiera miejsce znalezienia beacona i przekazuje je do zainteresowanych osób lub innych systemów.

Algorytm pracy został przedstawiony przy pomocy diagramu BPMN na rysunku 1.4

4.3. Podłączenie wielu statków do jednego systemu

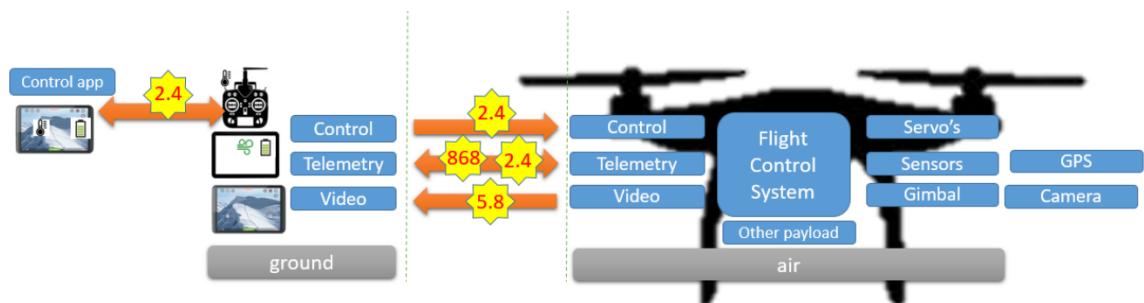
Projekt systemu od początku skupiał się na możliwości skorzystania z wielu statków powietrznych. Zostało to umożliwione, dzięki zastosowaniu infrastruktury typu klient-serwer, klientów może być dowolnie wielu i nie muszą o sobie wiedzieć, z tego punktu widzenia nie ma problemu z wieloma statkami. Również samo przybliżanie położenia beaconów, nie jest wrażliwe na ilość statków w przestrzeni, a to dzięki temu że aproksymacja dokonywana jest już po stronie klienta odbiorczego, im więcej punktów znalezionych, nawet jeżeli punkty znalazły różne statki, tym lepiej.

Może się natomiast pojawić problem innej natury, statki będą się poruszać po z góry zdefiniowanym obszarze i jak to bywa, mogą na siebie wpaść wykonując za-

danie, może być to błąd operatora, błąd aparatury sterującej, błąd w programie kontrolera, usterka sprzętowa, może to też być coś z zewnątrz, ptak, drzewo, człowiek. Możliwości jest wiele, wyeliminować wszystkich zagrożeń na pewno się nie da, ale można je ograniczyć, stosując się do przyjętych procedur, przepisów prawa lotniczego, dla UAV też już są specjalne przepisy, które regulują tą coraz szybciej rozwijającą się branżę.

Ostatnią kwestią są zakłócenia. Radia statków nie są dużych mocy, raz ze względu na ograniczenia regionalne, w Polsce obowiązuje bardzo rygorystyczne prawo, jeżeli chodzi o moc nadajników, dwa ze względu na to że im mocniejsze radio, tym więcej energii konsumuje, a na statku powietrznym nie ma chyba nic cenniejszego niż energia, trzeba ją oszczędzać. Dodatkowym problemem jest ilość urządzeń korzystających z tych samych częstotliwości, typowe częstotliwości stosowane dla UAV znajdują się w tabeli 4.1.

Jeden dron często używa do komunikacji z ziemią trzech a nawet czterech i więcej pasm częstotliwości, przykład pasm wykorzystywanych przez drona Phantom 3 znajduje się na rysunku 4.4.



Rys. 4.4. Częstotliwości wykorzystywane podczas lotu przez drona Phantom 3 [14]

Należy też pamiętać, że tych częstotliwości używają również inne technologie i rządzenia, takie jak wi-fi, bluetooth, sterowanie bram i automatyka budynkowa, modelarze itd. Nie ma dużego prawdopodobieństwa sytuacji w której dron odbierze i przetworzy jakiś sygnał który nadał punkt dostępowy sieci Wi-Fi na ziemi, ze względu na to, że każdy z sygnałów ma inny protokół i po prostu się nie zrozumieją.

Tab. 4.1. Typowe częstotliwości stosowane dla UAV

Częstotliwość	Opis
433MHz	sygnał telemetryczny w Europie
915MHz	sygnał telemetryczny w USA
1,2GHz	sygnał audio i wideo
2,4GHz	sygnał sterujący
5,8GHz	sygnał audio i wideo

Problemem jednak jest nadawanie na podobnych pasmach, w bliskiej odległości, sygnały które się na siebie nałożą, mogą prowadzić do zakłóceń i znaczącego spadku przepustowości transmisji, dlatego kilka dronów mających nadajniki sterujące na podobnej częstotliwości, może się zakłócać a ich zasięg może się przez to znaczco zmniejszyć [14].

4.4. Sterowanie autonomiczne

System zakładał możliwość lotu autonomicznego przez statki powietrzne, oczywiście zawsze pozostawała też możliwość lotu manualnego. Oprogramowanie którego użyliśmy udostępniało możliwość lotu autonomicznego bez żadnych modyfikacji kodu, miałem więc po części ułatwione zadanie. Po części, ponieważ latając kilkoma statkami po tym samym obszarze, zawsze istnieje możliwość kolizji, nasze oprogramowanie i sprzęt nie udostępniało nam żadnych narzędzi do omijania przeszkód, musieliśmy być pewni trasy przelotu która planowaliśmy. Samoczynne omijanie przeszkód, byłoby zdecydowanie zbyt szerokim tematem żeby mogły go zaimplementować, w rezultacie większa odpowiedzialność spadła na operatorów, którzy musieli dbać o bezpieczeństwo w tym zakresie. Ze względu na te ograniczenia zdecydowaliśmy, że tylko jeden z dronów będzie wykonywał misję w trybie autonomicznym, podczas gdy pozostałe będą sterowane manualnie.

4.5. Bezpieczeństwo użytkowania

Ratownik udzielający pomocy przy użyciu naszego systemu, powinien pamiętać przede wszystkim o swoim bezpieczeństwie. Tak jak ma to miejsce przy okazji wypadku samochodowego, którego świadkiem jesteśmy, najpierw dbamy o własne bezpieczeństwo, zabezpieczamy miejsce zdarzenia i dopiero wtedy przystępujemy do akcji ratunkowej. Podobnie tutaj, przed rozpoczęciem misji trzeba zadbać o bezpieczeństwo, sprawdzić tzw. checklistę startową, sprawdzić stan statku powietrznego, czy na przykład nie uległ uszkodzeniu w transporcie, zorientować się w kwestii pogody, która ma tutaj niebagatelne znaczenie. Ważne żeby operator był zaopatrzony w kamizelkę odblaskową w jaskrawym kolorze, dzięki temu każdy będzie go widział z daleka. Do startu trzeba przygotować miejsce, nie można wystartować jeżeli miejsca jest zbyt mało, szczególnie jest to ważne jeżeli ma się statek powietrzny o starcie poziomym, ze startem pionowym jest zdecydowanie łatwiej, ale też trzeba uważać.

Operatora nie należy rozpraszać, powinna być też jedna osoba odpowiedzialna za zwracanie uwagi osobom postronnym, które przeszkadzają operatorowi, zadają pytania, nawet nieświadomie mogą doprowadzić do wypadku.

Należy również pamiętać o zabezpieczeniu samego sprzętu, do elementów elektronicznych nie powinna mieć dostępu woda, głównie chodzi tu o deszcz, czyli musimy

ochronić statki co najmniej z góry i boków. Przechowywanie baterii i innej elektroniki również musi być robione w sposób przemyślany, co prawda napięcie na jakie są projektowane tego typu statki to najczęściej 24V, rzadziej, ale zdarza się 36V, więc w większości przypadków jest to napięcie uznawane za bezpieczne. Natomiast najczęściej stosuje się baterie z rodziny litowo-jonowych lub litowo-polimerowych, należy pamiętać że baterie tego typu potrzebują odpowiedniego układu elektronicznego tzw. BMS (ang. Battery Management System), który nie tylko dba o ich optymalne zużywanie ale przede wszystkim o bezpieczeństwo użytkownika.

4.6. Aspekty mechaniczne systemu

W systemie który projektowałem większość funkcjonalności była realizowana programowo, nie było elementów ruchomych, a jedyne o co musiałem zadbać, to mocowanie odbiornika sygnału beaconów, do statku powietrznego i zadbanie o to żeby nie uległ zamoczeniu, zostało to zrealizowane przy pomocy specjalnej obudowy do Raspberry Pi. W przypadku pątowca Raspberry Pi zostało zamocowane w kokpicie, obok kontrolera lotu, w przypadku naszego czterowirnikowca, pomiędzy baterią a kontrolerem pokładowym, w przypadku Phantoma nie było zbyt dużo możliwości, jako że wewnątrz jego korpusu nie ma miejsca na nic większego, a na pewno nie na komputer rozmiarów Raspberry Pi, dlatego Raspberry Pi zostało przymocowane do nóżek drona a zewnętrz.

5. Problemy i próba ich rozwiązania

5.1. Krótki czas lotu i zasięg UAV

Oczywistym problemem dronów zasilanych energią elektryczną, jest relatywnie krótki czas lotu i zasięg. Poniżej znajduje się tabela, gdzie dokonano szacunkowej wyliczenia czasu lotu i zasięgu przy umiarkowanych, porównywalnych dla każdego drona, warunkach pogodowych 5.1.

Nazwa	Zdjęcie	Zasięg [m]	Czas lotu [min]
Parrot AR drone		50	11
MJX Bugs 3		300	15

Upair One		500	20
Traxxas Aton		200	20
DJI Phantom 3 Stand		1500	25
Yuneec Q500+		2000	22
3DR Solo		500	22
Autel X-Star		2000	23
DJI Phantom 3 Pro		3000	23
DJI Phantom 4 Pro		7000	26
DJI Inspire 2		7000	27

DJI Mavic Pro		7000	27
---------------	--	------	----

Tab. 5.1. Szacunkowy czas lotu dronów seryjnych [15]

W tabeli zostały wymienione seryjne drony czterowirnikowe, do kwoty 3000\$ a więc takie które są dostępne od ręki, dla każdego, w sklepie. Można oczywiście latać droższym sprzętem, z baterią o większej gęstości energii, z lepiej zoptymalizowanym sprzętem. Jeżeli zależy nam na zasięgu można też użyć pątowca, pątowcem z takim samym osprzętem i masą startową, polecimy dalej i będziemy latać dłużej. Dla nas czas lotu nie był dużą przeszkodą, misja miała trwać maksymalnie 30 minut, zakładając że czterowirnikowcami nie startujemy od razu, spokojnie wystarczyło nam baterii. Zasięg też nie był problemem, maksymalnie oddalony od miejsca startowego punkt, w którym było jeszcze możliwe znalezienie beaconów, był w odległości około 900m mieliśmy więc pokrycie całego terenu.

5.2. Zawodność linku telemetrycznego

Jednym z poważniejszych problemów na jaki się natknąłem, było notoryczne zrywanie linku telemetrycznego pomiędzy dronem a stacją naziemną. Spowodowane to było przede wszystkim tym, że urządzenia które stosowaliśmy były wątpliwej jakości, sytuację pogarszał fakt że musieliśmy mieć jednocześnie kilka linków, do kilku dronów. Link telemetryczny nie jest konieczny do lotu, jest jednak konieczny do wykonania misji, ponieważ po linku telemetrycznym wysyłałem pozycje znalezionych beaconów. Problem rozwiązałem tak, że umożliwiłem wysyłanie pozycji beaconów do stacji naziemnej, przy pomocy sieci telefonii komórkowej GSM, czas takich żądań był dłuższy, ale zwiększała się niezawodność rozwiązania.

5.3. Wrażliwość systemu na sytuacje wyjątkowe

Tak jak wspomniałem wcześniej, nasz system był wrażliwy na różne wyjątkowe sytuacje, takie jak nagła zmiana pogody, opady deszczu, silny wiatr, nie posiadaliśmy również czujników zbliżeniowych na dronach, każdy więc błąd operatora mógł doprowadzić do kolizji z przeszkodą.

5.4. Problemy sprzętowe

Nie ominęły nas również problemy sprzętowe, wspominałem już o modułach telemetrycznych wątpliwej jakości. Podczas konkursu mieliśmy też problem z modułem GPS w jednym z dronów. Zawodziły też inne mniejsze podzespoły, mieliśmy jednak dużo sprzętu na wymianę.

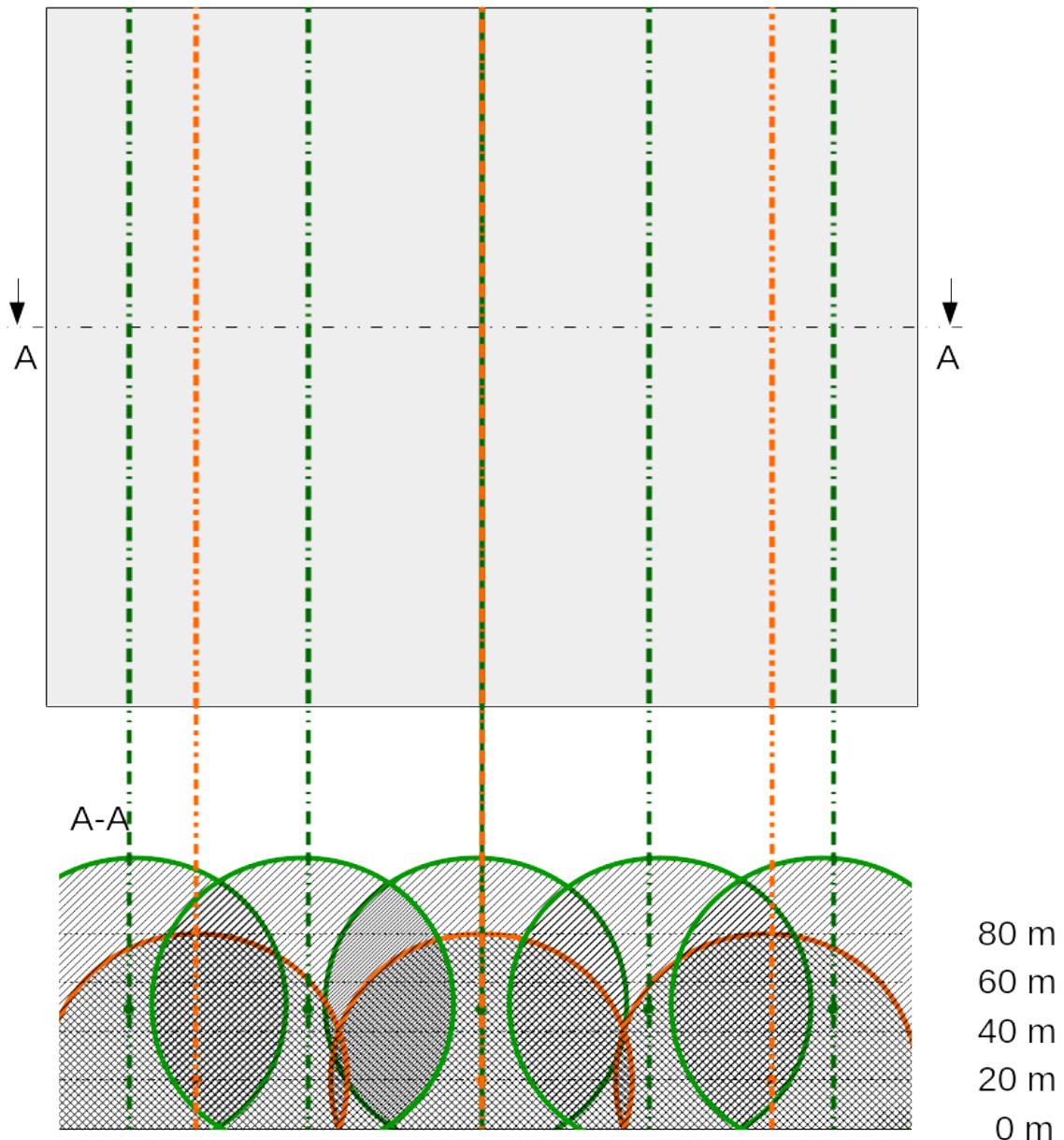
5.5. Sygnał beacona

Pierwsza sprawa jest taka, że sygnał nadawany przez beacony miał doświadczalnie przez nas zmierzony zasięg maksymalny około $80m$, przy dobrych warunkach pogodowych i bezpośredniej widoczności, bez żadnych przeszkód pomiędzy odbiornikiem i nadajnikiem. Dodatkowo opieraliśmy się tutaj na parametrze sygnału, RSSI [16], który wskazuje na siłę sygnału jaką widzi urządzenie odbiorcze, wskaźnik ten bardzo zależy od warunków pogodowych, przeszkód pomiędzy urządzeniami, wreszcie od samego wykonania urządzeń, szczególnie odbiorczego, na jego podstawie jedynie w przybliżeniu mogliśmy określić czy znajdujemy się w pobliżu beacona, czy jesteśmy daleko. Żeby więc znaleźć takiego beacona na ziemi z powietrza musimy po pierwsze lecieć nisko, a czasami nie jest to możliwe, są drzewa, budynki, linie energetyczne, które nie pozwolą nam lecieć nisko, założymy więc że na wysokości powyżej 20 metrów mamy wolną przestrzeń. Żeby jednak znaleźć wszystkie beacony, musimy lecieć tak, żeby każdy punkt oblatywanego obszaru był od nas najdalej o te założone kilkudziesiąt metrów, skoro maksymalnie widzieliśmy beacona z $80m$ założyliśmy że musimy lecieć co najdalej $60m$ od każdego punktu na przeszukiwanym obszarze. Im wyżej lecimy tym musimy lecieć ciasniej, czyli nasze ścieżki muszą być bliżej siebie. Problem ilustruje rysunek 5.1.

Na rysunku widzimy przykładowy obszar który musimy zbadać i dwie proponowane wysokości przelotu, wysokość $50m$ oznaczona jest na zielono, wysokość $20m$ na pomarańczowo, widać że aby przelecieć zadany obszar na wysokości $50m$ musimy przelecieć go w sumie 5 razy, w przypadku lotu na wysokości $20m$ wystarczy 3 razy, lot na wysokości $60m$ lub wyżej, przy naszych założeniach nie miałby już sensu.

6. Testy i misja konkursowa

Testy pojedynczych modułów przeprowadzane były na bieżąco, testów integracyjnych było jednak stosunkowo mało, ze względów logistycznych, ciężko znaleźć odpowiednie miejsce do latania, trzeba zabierać ze sobą dużo sprzętu. Tak na prawdę pierwszym prawdziwym sprawdzianem z wykonania misji była misja konkursowa. Poszła nam ona całkiem dobrze, nie wystrzegliśmy się jednak błędów, wylecieliśmy poza dozwolony obszar, nie udało nam się znaleźć jednego beacona, a dokładność



Rys. 5.1. Schematy przelotu dla różnych wysokości nad gruntem

naszych pomiarów pozostawiała wiele do życzenia. Dodatkowo aplikacja przez jakiś czas odpowiadała ze sporym opóźnieniem i musieliśmy sobie z tym radzić, przyczyny opóźnienia nie udało mi się ustalić.

Aplikacja do aproksymacji musiała znać przybliżoną charakterystykę sygnału beaconów, testy takie zostały przez nas wykonane, a wyniki zaimplementowane w funkcji aproksymującej. Aproksymacja działała całkiem dobrze, nie dało się jednak zaufać jej w pełni, operator systemu musiał na bieżąco weryfikować dane, pojawiające się na pulpicie systemu.

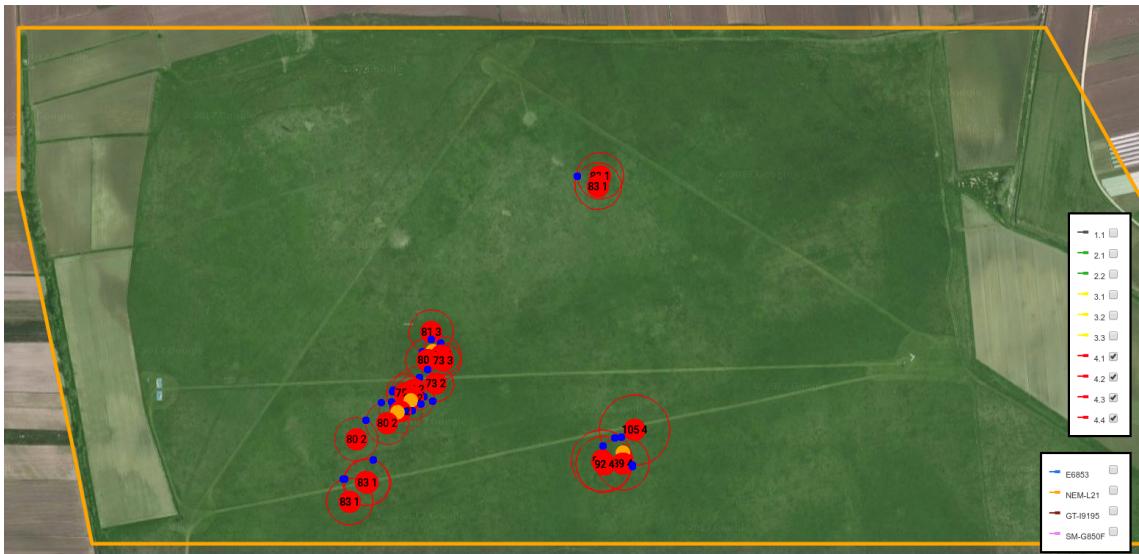
Na pierwszym obrazku w tym rozdziale [6.1](#) widzimy miejsca w których organizatorzy umieścili beacony, oczywiście nie znaliśmy ich położenia przed konkursem, dopiero po zakończeniu konkursu otrzymaliśmy te pozycje. Gwiazdkami w kolorach systemu triage, oznaczone są kolejne beacony, etykieta pod gwiazdką wskazuje na bajty Major i Minor nadawane przez beacony.



Rys. 6.1. Pozycje beaconów rozmieszczonych przez organizatorów podczas konkursu

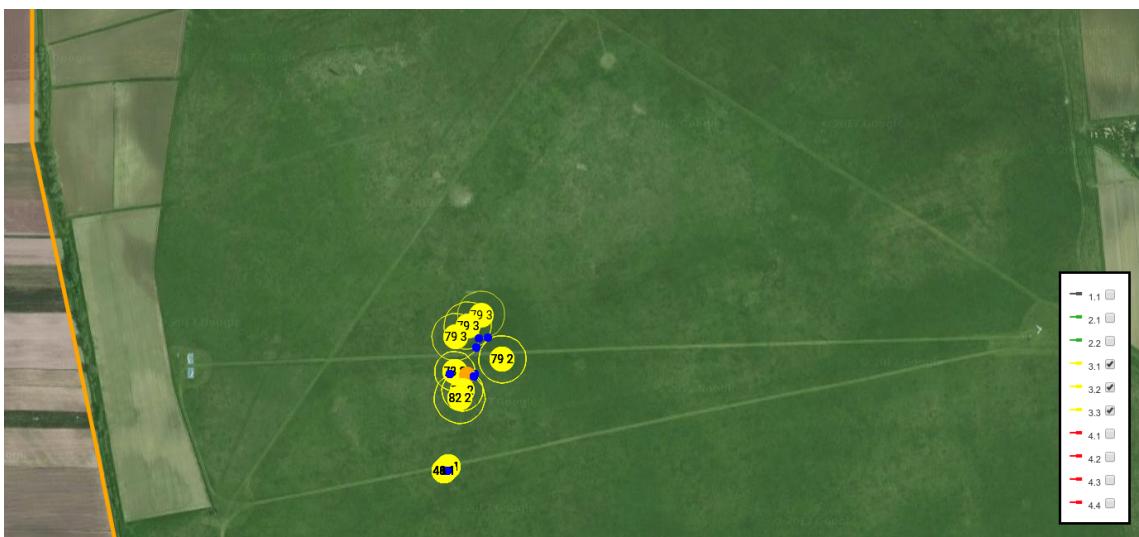
Kolejne zdjęcie [6.2](#) przedstawia, jak wyglądało wskazanie systemu na miejsce położenia beaconów nadających bajt Major równy 4. Udało nam się znaleźć wszystkie czerwone beacony, z dość dobrą dokładnością, chociaż nie ukrywam że niektóre pomiary były mylące, a nasz system nie miał niestety zbyt dużych możliwości filtracji danych, widać to na przykładzie beacona 4.1, otrzymaliśmy sprzeczne dane, jakoby beacon nadający te dane znajdował się w dwóch miejscach, oddalonych od siebie o około $300m$, co nie było możliwe, bo tak jak wspominałem, zasięg sygnału beaconów to maksymalnie około $80m$.

Następny rysunek [6.3](#) pokazuje wskazanie na żółte beacony, jak przy czerwonych mieliśmy w miarę dobrą dokładność, w najgorszym przypadku $22,37m$ tak w przypadku żółtego z numerem 3 pomyliliśmy się aż o $126,74m$, to był nasz najgorszy wynik, żółty z numerem 2 miał błąd $81,58m$, widać że potrzebna byłaby dodatkowa



Rys. 6.2. Wskazanie naszego systemu na miejsca położenia beaconów czerwonych

praca nad polepszeniem dokładności pomiarów.



Rys. 6.3. Wskazanie naszego systemu na miejsca położenia beaconów żółtych

Ostatnie są beacony zielone, czarnego niestety nie znaleźliśmy, na rysunku 6.4 widzimy nasze wskazanie na nie. Dokładności w tym przypadku były w miarę dobre, w okolicach 15m.

Na ostatnim rysunku 6.5 mamy porównanie tego gdzie były beacony do tego co wskazał mój system.

Na koniec tabela 6.1 z położeniem beaconów, naszym wskazaniem i naszym błędem pomiaru.



Rys. 6.4. Wskazanie naszego systemu na miejsca położenia beaconów zielonych



Rys. 6.5. Nałożenie naszych szacunkowych pozycji na obraz z pozycjami realnymi

7. Podsumowanie

7.1. Wnioski

Podstawowym wnioskiem z przeprowadzonych testów i zawodów jest to że system w takim stanie w jakim był, nie może mieć żadnego poważnego zastosowania, w

Tab. 6.1. Położenie beaconów konkursowych oraz naszych i błąd naszego pomiaru

Beacon	Pozycja realna [lat, lon]	Pozycja zmierzona [lat, lon]	Błąd [m]
1.1	50.0888040, 20.2041863		
2.1	50.0889409, 20.1968885	50.088867, 20.197039	13,46
2.2	50.0916493, 20.1993463	50.091776, 20.199251	16,65
3.1	50.0898048, 20.1971456	50.089800, 20.197217	4,73
3.2	50.0896552, 20.1971186	50.089036, 20.197759	81,58
3.3	50.0908825, 20.1989062	50.089979, 20.197821	126,74
4.1	50.0916549, 20.1998945	50.091767, 20.199826	13,61
4.2	50.0897875, 20.1975500	50.089981, 20.197523	22,37
4.3	50.0900354, 20.1976396	50.090101, 20.197607	8,21
4.4	50.0890267, 20.2006400	50.089051, 20.200665	3,23

szczególności nie może służyć ratownikom przy prowadzeniu akcji ratowniczej. Drony które mieliśmy do dyspozycji miały krótkie czasy lotu, mały zasięg, a prawdziwa misja ratunkowa nie odbywa się na powierzchni kilku kilometrów kwadratowych, gdzie można całą akcję przeprowadzić ręcznie, bez użycia niepotrzebnego dodatkowego sprzętu. Można oczywiście użyć droższego sprzętu, o większym zasięgu, ale znów czy beacony to dobry wybór? Moim zdaniem nie, ratownik który oznacza poszkodowanego beaconem, równie dobrze może sam zaznaczyć na mapie gdzie go zostawił, nie są tu potrzebne drony do skanowania obszaru. Kolejną sprawą jest to, że nasz system zakłada że beacony leżą statycznie w jednym miejscu, w prawdziwym życiu człowiek może się ruszać i przechodzić w inne miejsce, a to eliminuje ten system z użytku.

7.2. Plany na przyszłość

Planów na przyszłość z tym konkretnym projektem nie mam, wydaje mi się że to co tutaj robiliśmy, w tej formie, nie ma przyszłości w ratownictwie, co najwyżej komponenty tego systemu mogłyby mieć zastosowanie w innym projekcie. Samo wyszukiwanie beaconów mogłyby mieć zastosowanie na przykład w biurach do sprawdzania kiedy pracownik przeszedł, wyszedł, gdzie mniej więcej się znajduje, a można by to było realizować przy pomocy beacona takiego jak na rysunku 7.1.

Podobnie na imprezach masowych można przy pomocy takiego beacon śledzić osoby. Gdyby zasięg sygnału takiego beacona był większy, na przykład kilka kilometrów, takie beacony można by było rozdawać ludziom którzy idą w góry, dzięki temu łatwiej byłoby ich później znaleźć przy pomocy drona, w przypadku gdyby zabłądzili.



Rys. 7.1. Beacon w postaci karty, rozmiarowo zbliżony do karty kredytowej

Literatura

- [1] *Parada Robotów Droniada 2017[online]*, dostęp w internecie w dniu 24 czerwca 2017: <http://www.5zywiolow.pl/droniada2017/>
- [2] *Bluetooth low energy beacon[online]*, dostęp w internecie w dniu 1 października 2017: https://en.wikipedia.org/wiki/Bluetooth_low_energy_beacon
- [3] *Znacznik systemu Triage[online]*, dostęp w internecie w dniu 24 czerwca 2017: https://en.wikipedia.org/wiki/Triage_tag
- [4] *Internet rzeczy[online]*, dostęp w internecie w dniu 24 czerwca 2017: https://en.wikipedia.org/wiki/Internet_of_things
- [5] *Raport Obserwatorium ICT — Internet przedmiotów [online]*, dostęp w internecie w dniu 2 października 2017: http://www.obserwatoriumit.pl/site/assets/files/1059/internet_of_things.pdf
- [6] *Central Nervous System of the Earth [online]*, dostęp w internecie w dniu 2 października 2017: <https://en.wikipedia.org/wiki/CeNSE>
- [7] *Protokół iBeacon[online]*, dostęp w internecie w dniu 8 lipca 2017: <https://developer.apple.com/ibeacon/>

- [8] *Protokół Eddystone[online]*, dostęp w internecie w dniu 21 października 2017: <https://developers.google.com/beacons/eddystone>
- [9] *Gateway producji firmy kontakt.io[online]*, dostęp w internecie w dniu 8 lipca 2017: <https://store.kontakt.io/next-generation/33-gateway.html>
- [10] *Strona domowa projektu RTAndroid[online]*, dostęp w internecie w dniu 8 lipca 2017: <https://rtandroid.embedded.rwth-aachen.de>
- [11] *Raspberry Pi models comparison[online]*, dostęp w internecie w dniu 21 października 2017: <https://socialcompare.com/en/comparison/raspberrypi-models-comparison>
- [12] *Dronekit[online]*, dostęp w internecie w dniu 21 października 2017: <http://dronekit.io/>
- [13] *Rozporządzenie Urzędu Komunikacji Elektronicznej ws. dozwolonych częstotliwości nie wymagających zezwolenia na terenie RP[online]*, dostęp w internecie w dniu 18 września 2017: https://www.uki.gov.pl/files/?id_plik=12366
- [14] *Research into the Radio Interference Risks of Drones[online]*, dostęp w internecie w dniu 20 września 2017: https://www.agentschaptelecom.nl/sites/default/files/research_into_radio_interference_risks_of_drones.pdf
- [15] *Drones with the best flight times[online]*, dostęp w internecie w dniu 27 września 2017: <http://www.dronesglobe.com/guide/long-flight-time/>
- [16] *Received signal strength indication[online]*, dostęp w internecie w dniu 30 września 2017: https://en.wikipedia.org/wiki/Received_signal_strength_indication

Spis rysunków

1.1. Część naszego zespołu na konkursie ”Droniada”, osoby które brały udział w konkursie głównym	11
1.2. Schemat ideowy systemu, widać na nim wszystkie elementy systemu, począwszy od poszkodowanych, przez drony i ich operatorów, a skończywszy na wszystkich elementach stacji naziemnej i analitykach zebranych danych	11
1.3. Typowy znacznik systemu Triage, używany w akcjach ratowniczych [3]	12
1.4. Ogólny diagram BPMN przepływu danych w systemie	13
2.1. Komputer Raspberry Pi 3 Model B, użyty w moim projekcie w charakterze odbiornika bluetooth danych z beaconów	17
3.1. Nasz pątowiec, na bazie kadłuba skyhunter, podczas wykonywania misji konkursowej	18
3.2. Nasz czterowirnikowiec, zbudowany od podstaw, wyposażony w kontroler lotu Pixhawk	18
3.3. Mapa terenu w obrębie którego mogły znajdować się beacony konkursowe, pomarańczową linią wyznaczona jest granica zewnętrzna terenu	19
3.4. Przykład misji autonomicznej w programie Mission Planner	20
3.5. Dron jednej z drużyn spłonął na zawodach	21
3.6. To co zostało z drona po pożarze baterii	21
4.1. Schemat blokowy systemu	23
4.2. Widok ekranu operatora systemu, w centralnym miejscu mapa z zaznaczoną trasą przelotu, próbami znalezionej sygnału, oraz wyznaczoną aproksymacją	24
4.3. Komerencyjny dron Phantom 3, użyty przez nas w misji	25
4.4. Częstotliwości wykorzystywane podczas lotu przez drona Phantom 3 [14]	27
5.1. Schematy przelotu dla różnych wysokości nad gruntem	32
6.1. Pozycje beaconów rozmieszczonych przez organizatorów podczas konkursu	34
6.2. Wskazanie naszego systemu na miejsca położenia beaconów czerwonych	34
6.3. Wskazanie naszego systemu na miejsca położenia beaconów żółtych .	35
6.4. Wskazanie naszego systemu na miejsca położenia beaconów zielonych	35
6.5. Nałożenie naszych szacunkowych pozycji na obraz z pozycjami realnymi	36
7.1. Beacon w postaci karty, rozmiarowo zbliżony do karty kredytowej .	37

Spis tabelic

2.1. Kodowanie kolorów na dane Minor i Major	16
4.1. Typowe częstotliwości stosowane dla UAV	27
5.1. Szacunkowy czas lotu dronów seryjnych [15]	30
6.1. Położenie beaconów konkursowych oraz naszych i błąd naszego pomiaru	36