



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra měření**

Bakalářská práce

Návrh a realizace monitorovacího systému pro bezpilotní prostředky

Richard Šimeček
Elektronika a komunikace

Květen 2022

Vedoucí práce: Ing. Martin Šipoš, Ph.D.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šimeček** Jméno: **Richard** Osobní číslo: **492032**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektromagnetického pole**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a realizace monitorovacího systému pro bezpilotní prostředky

Název bakalářské práce anglicky:

Design and Realization of a Monitoring System for Unmanned Aerial Vehicles

Pokyny pro vypracování:

Navrhněte a zrealizujte monitorovací systém pro bezpilotní prostředek (UAV) s přenosem vybraných parametrů do pozemní základnové stanice. Bakalářská práce bude zahrnovat následující kroky a splní vybrané klíčové vlastnosti:

- Návrh koncepce systému a použité platformy s ohledem na využitelnost systému, velikost, hmotnost, atd.
- Výběr a otestování komunikačního kanálu pro bezpilotní prostředek se zaměřením na spolehlivost přenosu dat a dosah.
- Monitorování GNSS pozice, výšky, rychlosti letu, napětí baterie, atd.
- Možnost připojit avionický systém přes rozhraní RS-232, CAN a odesílání vybraných parametrů do základnové stanice.
- Realizaci HW, který bude propojovat dílčí moduly, za účelem zvýšení robustnosti řešení a odolnosti na vibrace.
- Návrh a realizaci SW (případně modifikaci stávajícího SW) pro vizualizaci parametrů přenášených z paluby UAV.
- Praktické ověření celého systému.

Seznam doporučené literatury:

- [1] Monk, S: Electronics Cookbook: Practical Electronic Recipes with Arduino and Raspberry Pi, 2017 Simon Monk, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, ISBN 978-1-491-95340-2.
- [2] Karvinen, K; Karvinen, T.: Make: Getting Started with Sensors, Measure the World with Electronics, Arduino, and Raspberry Pi, Maker Media, Sebastopol, CA, 2014, ISBN: 978-1-4493-6708-4.
- [3] Cicolani, J.: Beginning Robotics with Raspberry Pi and Arduino Using Python and OpenCV, Apress, Pflugerville, Texas, USA, 2018, ISBN-13 (electronic): 978-1-4842-3462-4.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Martin Šipoš, Ph.D. katedra měření FEL (13138)

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **31.01.2022**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Martin Šipoš, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Rád bych poděkoval panu Ing. Martinu Šipošovi, Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích týkajících se mé bakalářské práce. Také děkuji svým rodičům za podporu při studiu. Dále děkuji Českému vysokému učení technickému v Praze za nabyté vědomosti a také fakultě měření za poskytnutí modulů a hardwaru.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 20. 05. 2022

.....

Abstrakt / Abstract

Tato bakalářská práce se zaměřuje na vytvoření systému pro spolehlivý přenos dat na delší vzdálenosti, zobrazení a uchování dat z avionických systémů leteckého prostředku. Provedl jsem porovnání 433 MHz rádiových modulů pro přenos dat přes UART a také 3G/2G přenos dat přes mobilní síť. Na základě výsledků měření jsem ke stavbě systému použil mikrokontrolér Arduino MKR 1400 společně s několika shieldy a základovou deskou vlastního návrhu. Dále pak Open Source databázi pro sběr dat v čase InfluxDB společně s Telegraf agentem pro příjem UDP paketů z Arduino a předání dat databázi. Databáze a Telegraf agent běží na Raspberry Pi 4 se statickou IP adresou a nainstalovaným operačním systémem Ubuntu Desktop 22.04 64-bit. InfluxDB má integrované vizuální rozhraní pro zobrazení dat v reálném čase a umožňuje data stáhnout pro jejich pozdější zpracování například v prostředí Matlab. Výsledkem práce je kompaktní jednotka pro sběr a posílání dat, dále pak základnová stanice RPi 4 pro příjem dat.

Klíčová slova: sběr dat, UAV, LTE, RPi 4, databáze, Arduino MKR 1400, GSM, C++

My bachelor work focuses on testing and comparing two types of wireless data transmission methods, one utilizing 433 MHz UART transmitters, the other one using 3G/2G module. I am comparing their performance and utilizing the better one for telemetry acquisition, transmission and graphical representation in real time as well as exporting the data for further research. For base station i used Raspberry Pi 4 with Ubuntu Desktop 21.10 64-bit OS running InfluxDB time based database and useragent Telegraf which handles incoming UDP packets for the database. It is possible to show values in real time thanks to integrated InfluxDB Dashboard and export stored values for further research. The system is comprised of flight unit based on Arduino MKR 1400 with several shields, enabling it to store collected data from said shields onto SD card as a backup. Arduino is mounted in motherboard of my own design to provide power and additional features such as sensor expansion headers and bus expansion ports.

Keywords: data acquisition, UAV, LTE, RPi 4, time series database, Arduino MKR 1400, C++, GSM

Title translation: UAV monitoring system

/ Obsah

1 Úvod	1
2 Výběr bezdrátové technologie	2
2.1 Popis rádiových modulů	
UART 433MHz	2
2.1.1 Základnový modul	3
2.1.2 UAV modul	3
2.2 Evaluace modulů	4
2.3 Popis experimentu	5
2.4 Výsledky měření	6
3 Přenos dat založený na mobilní komunikační síti	9
3.1 Arduino MKR 1400	9
3.2 Rozšiřující desky	11
3.3 Testování přenosu a výsledky měření	11
4 Realizace systému pro přenos telemetrie	12
4.1 UAV část	13
4.2 Použité rozšiřující desky	13
4.2.1 MKR ENV Shield	13
4.2.2 MKR CAN shield	14
4.2.3 MKR GPS Shield	17
4.3 Základní deska	18
4.3.1 Napájení	19
4.3.2 RS232	20
4.4 Základnová stanice	21
4.4.1 Serverová část	22
4.4.2 databáze InfluxDB	23
4.4.3 Telegraf	24
5 Závěr	26
Literatura	28
A Zkratky	33

Tabulky / Obrázky

2.1 Výsledky měření přenosu s původními anténami od výrobce modulů	7
2.2 Výsledky měření přenosu s alternativním párem antén	7
3.1 Pásmo a přenosové standardy podporované modemem U-blox SARA-U201	10
3.2 Výsledky měření provedené přes ICMP pakety	11
2.1 Nastavení obou 433MHz modulů pomocí programu Mission Planner	3
2.4 Ukázka testovacího Python skriptu	4
2.6 Druhý set antén použitých pro měření 433MHz modulů	6
2.7 Snímek obrazovky se zobrazeným spektrem při testování modulů 433MHz	8
4.1 Struktura zprávy CAN	15
4.2 Testovací zapojení komunikace s IMU jednotkou	16
4.3 Ukázka příjmu dat z inerciální jednotky přes sběrnici CAN a jejich výpis na sériovou linku	16
4.4 Data z inerciální jednotky zobrazené jako časové řady v databázi	17
4.5 GPS shield propojený s Arduino MKR 1400 přes sběrnici Display Data Channel	18
4.6 Základní deska pro UAV část systému	19
4.7 Ukázka duplexního přenosu pomocí čipu pro RS232 sběrnici	21
4.8 Sestavená UAV část	21
4.9 Dashboard s ukázkou zobrazení dat v reálném čase	24
4.10 Snímek obrazovky při testování UDP příjmu dat	25

Kapitola 1

Úvod

Moderní letectví je velice rozsáhlou a komplexní oblastí lidského technologického vývoje. Letadla zahrnují spoustu propracovaných systémů, které přispívají k bezpečnosti a komfortu posádky a cestujících. Umožňují přesun na dlouhé vzdálenosti spolehlivě a rychle za prakticky jakéhokoliv počasí a jsou nepostradatelnou součástí globální novodobé společnosti.

K nejdůležitějším systémům v leteckém odvětví patří ty radarové, poziční, hydraulické, elektromechanické a především systémy telekomunikační. Je nutné, aby piloti a posádka mohli kontaktovat letiště a jiná letadla v případě nouze nebo při rutinních činnostech a celkové koordinaci leteckého provozu. Jedna z možností spojení je Aircraft Communications Addressing and Reporting System, tedy zkráceně ACARS[1]. ACARS umožňuje zasílat krátké zprávy mezi letouny a pozemními stanicemi přes rádiové vlny. Další možností je pak dnes již klasická rádiová AM vysílačka pro spojení na kratší vzdálenosti.

Ovšem pro bezpilotní letecké prostředky se nároky na telekomunikační spojení liší, a to už z podstaty konstrukce samotného UAV (Unmanned Aerial Vehicle). V tomto dokumentu je zkratkou UAV myšlen bezpilotní letecký prostředek, tj. vzdáleně pilotovaný prostředek jako například RC letadla či koptéry různých variant. Pro velké UAV je kladen mnohem větší důraz na spolehlivost, rychlost a kapacitu přenosového kanálu. V případě výpadku by totiž v nejhorším případě mohlo dojít ke ztrátě kontroly nad UAV a posléze tak i ztrátě prostředku samotného. Pro malé UAV běžně dostupné pro širokou veřejnost jsou však nároky na přenos mnohem menší. Z toho vyplývá, že dimenzování komunikačního kanálu se odvíjí od požadavků leteckého prostředku na monitorování a ovládání. Například pro pasivní odesílání malého datového packetu telemetrie jednou za několik sekund budou nároky na kapacitu kanálu mnohem menší, než pro přenos videostreamu z kamery pro pilotování prostředku. [2]

Cílem této práce je zhodnotit spolehlivost, rychlost a kapacitu dvou bezdrátových technologií, a to přenos přes USB sériové rozhraní 433 MHz rádiovými moduly a dále pak přenos přes pokročilejší bezdrátovou technologii pro přenos dat založené na mobilní síti. Součástí cíle této práce je na zvolené bezdrátové technologii poté vybudovat systém pro odesílání a příjem dat s možností sledování hodnot v reálném čase. Systém by měl být schopný přijímat data od více leteckých prostředků najednou a základnová stanice by měla fungovat jako mobilní centrum pro monitorování UAV.

Kapitola 2

Výběr bezdrátové technologie

2.1 Popis rádiových modulů UART 433MHz

Jedná se o pár rádiových modulů určených pro přenos telemetrických dat přes UART a USB sériové rozhraní. Moduly jsou kompatibilní s UAV základovými deskami PixHawk a Ardupilot. Rádiové moduly vysílají na frekvenci 433 MHz, jež se nachází na pásmu ISM pro přenos dat, vyčleněné Českým telekomunikačním úřadem (ČTÚ)[3]. Antény byly součástí balení a jsou připojeny pomocí SMA konektoru. Oba moduly jsou napájeny 5 V, základní přenosová rychlost portu je nastavena na 115200 baud. Výpis technických specifikací ze stránky výrobce[4]:

```
Very small size
Air module weight (with antenna): 18gr
Frequency: 433MHz
Receiver sensitivity to -121 dBm
Transmit power up to 20dBm (100mW)
Transparent serial link
Air data rates up to 250kbps
MAVLink protocol framing and status reporting
Frequency hopping spread spectrum (FHSS)
Adaptive time division multiplexing (TDM)
Support for LBT and AFA
Configurable duty cycle
Built in error correcting code (can correct up to 25% data bit errors)
Open source firmware
AT commands for radio configuration
RT commands for remote radio configuration
Adaptive flow control when used with APM
Based on the HopeRF HM-TRP radio module, featuring an SiLabs Si1000 RF
microcontroller.
```

V předchozím seznamu jsou uvedeny některé důležité informace o fungování modulů, které ovlivňují, jak jsou data přenášena. Jedna z přenosových metod, kterou tyto moduly používají, je časový multiplex. Další metoda je FHSS (Frequency hopping spread spectrum), určena pro posílání dat přes rozšířené radiové spektrum. Data jsou vysílačem posílána přes relativně úzké pásmo, ale vysílač toto úzké pásmo přesouvá po určité větší šířce pásma ve velice krátkých časových intervalech. Přijímač musí vědět kam, se v danou chvíli naladit. Na druhou stranu FHSS poskytuje také řadu výhod. Tím, že vysílání nezůstává na jediné frekvenci, je přenos odolný proti úzkopásmovému rušení a se správně nastavenou sekvencí skoků je možné provést plně duplexní přenos. [5]

Oba moduly mají integrovanou LED diodu indikující přítomné napájecí napětí a dále zelenou indikační LED diodu, která při úspěšném navázání spojení mezi moduly bliká.

Blikání neindikuje přenos dat, pouze potvrzuje, že spojení je aktivní a připraveno k přenosu mezi moduly. Pokud spojení není navázáno, svítí zelená indikační dioda bez přerušení.

Obrázek 2.1 je snímek obrazovky programu Mission Planner, který slouží k nastavení parametrů modulu [6]. Já jsem ponechal tovární nastavení, pouze jsem změnil výkon vysílače tak, aby splňoval podmínky využití volného pásma 433 MHz. [3]



Obrázek 2.1. Nastavení obou modulů pomocí programu Mission Planner

2.1.1 Základový modul

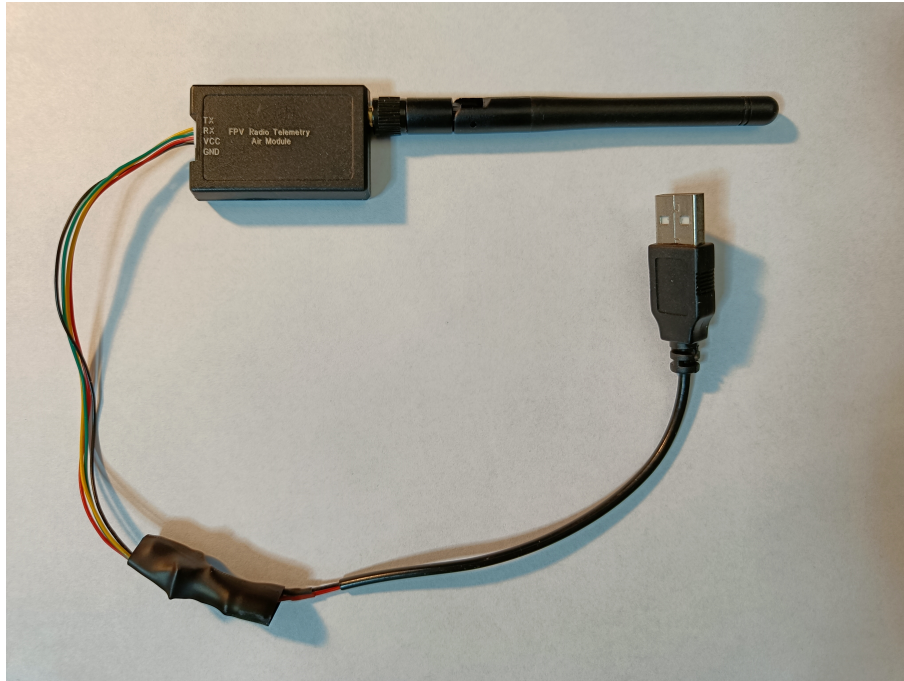
Základový modul je vybaven USB konektorem typu A a po připojení do počítače se zobrazí jako sériový COM port, rychlost přenosu je nastavena na 115200 baud.



Obrázek 2.2. Základový modul

2.1.2 UAV modul

UAV modul je vybaven JST konektorem, ke kterému jsem vytvořil adaptér pro napájení přes USB. UAV modul byl napájen z powerbanky, piny RX a TX byly zkratovány. Modul jsem tímto nastavil tak, aby příchozí data odesílal zase zpět do základového modulu. Při testování spojení na krátkou vzdálenost v místnosti se ukázalo, že powerbanka nedokáže dost rychle reagovat na náhlé proudové špičky odebírané zátěží. Aby tedy nedocházelo k poklesu napájecího napětí náhlým proudovým zatížením při vysílání, přidal jsem na napájecí vývody paralelně jeden keramický 220 nF kondenzátor a dva elektrolytické kondenzátory s celkovou kapacitou 320 uF, což problém s napájením zcela vyřešilo.



Obrázek 2.3. UAV modul s upraveným napájecím kabelem.

2.2 Evaluace modulů

Hlavními kritérii pro evaluaci byly rychlost přenosu, dosah a spolehlivost. Pro účely testování jsem sepsal jednoduchý program v Pythonu, jež odesílá daný počet libovolných zpráv po zvolené sériové lince, přijatá data uloží do .txt souboru. Skript byl navržen tak, aby byl co nejvíce univerzální a daly se měnit různé parametry přenosu, tedy délka zprávy, její obsah a počet odesílaných zpráv. Z těchto parametrů je počítána celková chybovost a rychlost přenosu.

Popis výstupu z testovacího Python programu a Obrázek 2.4 pro ukázkou:

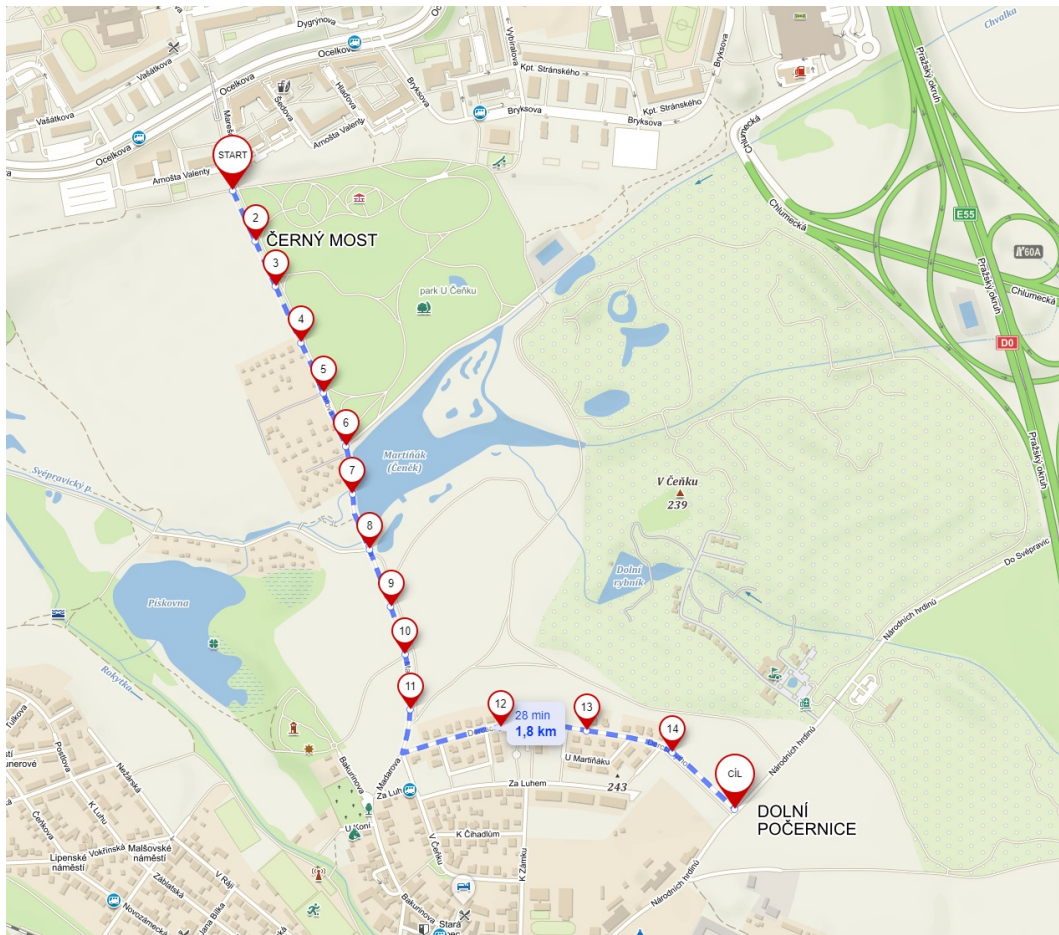
```
[Zpráva přijata v bez chyb] [Rychlost přenosu]B/s ([číslo zprávy],
[počet odeslaných zpráv v %]) [[časový rozdíl mezi odesláním
a přijetím zprávy]] [čas odeslání zprávy ve formátu
hh:mm:ss] [text zprávy]
```

```
COM7: Silicon Labs CP210x USB to UART Bridge (COM7) [USB VID:PID=10C4:EA60 SER=0001 LOCATION=1-8]
Select port: 0
COM7
Type name of file: test
Number of messages: 10
Message to send: test message
Yes 104B/s (1, 10.0%) [0.21011114120483398]: 22:43:11 test message
Yes 124B/s (2, 20.0%) [0.1768631935119629]: 22:43:12 test message
Yes 99B/s (3, 30.0%) [0.22198486328125]: 22:43:12 test message
Yes 109B/s (4, 40.0%) [0.2006089687347412]: 22:43:12 test message
Yes 109B/s (5, 50.0%) [0.20004606246948242]: 22:43:12 test message
Yes 136B/s (6, 60.0%) [0.16110491752624512]: 22:43:13 test message
Yes 104B/s (7, 70.0%) [0.21008896827697754]: 22:43:13 test message
Yes 105B/s (8, 80.0%) [0.2080824375152588]: 22:43:13 test message
Yes 121B/s (9, 90.0%) [0.18108034133911133]: 22:43:13 test message
Yes 102B/s (10, 100.0%) [0.21476030349731445]: 22:43:13 test message
Average speed: 111.3B/s
Total errors: 0.0%
>>> |
```

Obrázek 2.4. Názorný běh Python skriptu pro měření přenosových vlastností radiových modulů. Moduly byly umístěny vedle sebe na stole.

2.3 Popis experimentu

Rádiové moduly jsem testoval venku na plochem, otevřeném prostranství. Na statickém bodě START, od kterého se měření provádělo, byl PC s připojeným základnovým modulem. Další body měření byly rozvrženy tak, aby mezi nimi byl vždy rozestup 100m. Na tyto body se po každé sérii měření postupně přesouval UAV modul.



Obrázek 2.5. Mapa stanovišť měření

Pro účel kontroly a pozorování rádiového spektra jsem připojil do dalšího USB portu notebooku DVBT přijímač sloužící s programem SDR Sharp jako SDR přijímač. K SDR přijímači jsem připojil širokopásmovou anténu dodávanou v balení DVBT přijímače. Pozorování rádiového spektra jsem provedl za pomoci programu SDR Sharp. To mi umožnilo sledovat průběh posílání zprávy a ujistit se, že nedochází k rušení vnějšími vlivy. Vzhledem k tomu, že za místo testování jsem zvolil rozsáhlý park, bylo rádiové spektrum volné a na pracovní frekvenci modulů 433 MHz jsem nezaznamenal žádné vnější přenosy jež by narušovaly experiment.

Protože jsem chtěl nezávisle otestovat i vysílací schopnost modulů, udělal jsem druhé kolo experimentu s anténami, které mi byly zapůjčeny a u kterých máme prakticky vyzkoušené dobré přenosové vlastnosti.

K testování modulů byly využity dva sety antén:

- Antény dodávané s moduly

Unipólové antény od výrobce modulů. Bohužel není nikde k dohledání o jaký typ antén se jedná. Dle vzhledu a délky by odpovídala konstrukce typu unipólové antény

■ Alternativní antény

V druhé sérii měření byly použity antény přizpůsobené na frekvenci 433 MHz od jiných výrobců, jež byly v praxi spolehlivě otestovány v rámci různých předchozích projektů jak v laboratoři, tak i ve venkovním prostředí. Jednalo se o magnetické antény SW433-XP1M¹ a SENDA RG433M².



Obrázek 2.6. Druhý set antén použitých pro měření 433MHz modulů, nalevo anténa SW433-XP1M použitá pro UAV modul, napravo anténa SENDA RG433M použitá pro základnový modul

2.4 Výsledky měření

Tabulky 2.1 a 2.2 zobrazují výsledky měření radiových modulů. Jako data se odesílala následující zpráva o velikosti 86 bytů, její délka a struktura byla zvolena tak, aby simulovala telemetrická data.

```
16:08:11 123.12345648,15.1238446513,25.6,1231456,123415645,123456156,
3650,test message
```

Pokud se přijatá zpráva lišila od odeslané v jakémkoliv ze znaků, započítala se jako chybová. Podíl chybových zpráv ku úspěšně přeneseným zprávám je na konci měření vyjádřený v procentech jako **chybovost**. Dále se měří časový úsek mezi odesláním a přijetím zprávy. Z podílu počtu znaků ve zprávě a časového rozdílu Δt je pro danou zprávu vypočtena rychlost přenosu v B/s. Časový rozdíl Δt v tabulce byl vypočten jako průměr všech naměřených časových rozdílů. Výsledný průměr všech rychlostí v_p byl vypsán na konci běhu programu stejně tak jako **chybovost**. U chybových zpráv se rychlost přenosu nezapočítává. s udává vzdálenost v metrech mezi rádiovými moduly. Pokud došlo ke ztrátě spojení mezi moduly, bylo měření ukončeno a následně bylo určeno kolik procent zpráv z celkového počtu se podařilo odeslat a přijmout. Tato hodnota byla označena *up*. Počet odesílaných zpráv v jednom měření je značen n .

První měření bylo vždy provedeno jako kontrolní, radiové moduly byly umístěny přibližně jeden metr od sebe. Zprávy byly odesílány s frekvencí 1 Hz.

Pokud měření neproběhlo až do konce a nevrátily se všechny zprávy, Python program nepočítal chybovost ani průměry časového rozdílu a rychlosti přenosu.

¹ <https://dratek.cz/arduino/2132-nicerf-433mhz-antena-stojankova-sw433-xp1m-116mm.html>

² <https://www.shop3000.cz/uhf-433868-mhz/3432-senda-rg433m.html>

s [m]	up [%]	n [-]	chybovost [%]	Δt [ms]	v_p [B/s]
1	100	100	5.0	238.04	361.28
100	100	100	1.0	228.62	376.17
100	38.2	500	-	-	-
100	100	250	0.8	229.17	375.252
200	0	-	-	-	-
200	0	-	-	-	-
200	0	-	-	-	-
200	0	-	-	-	-
200	0	-	-	-	-
200	100	250	6.4	242.46	354.688

Tabulka 2.1. Výsledky měření přenosu s původními anténami od výrobce modulů

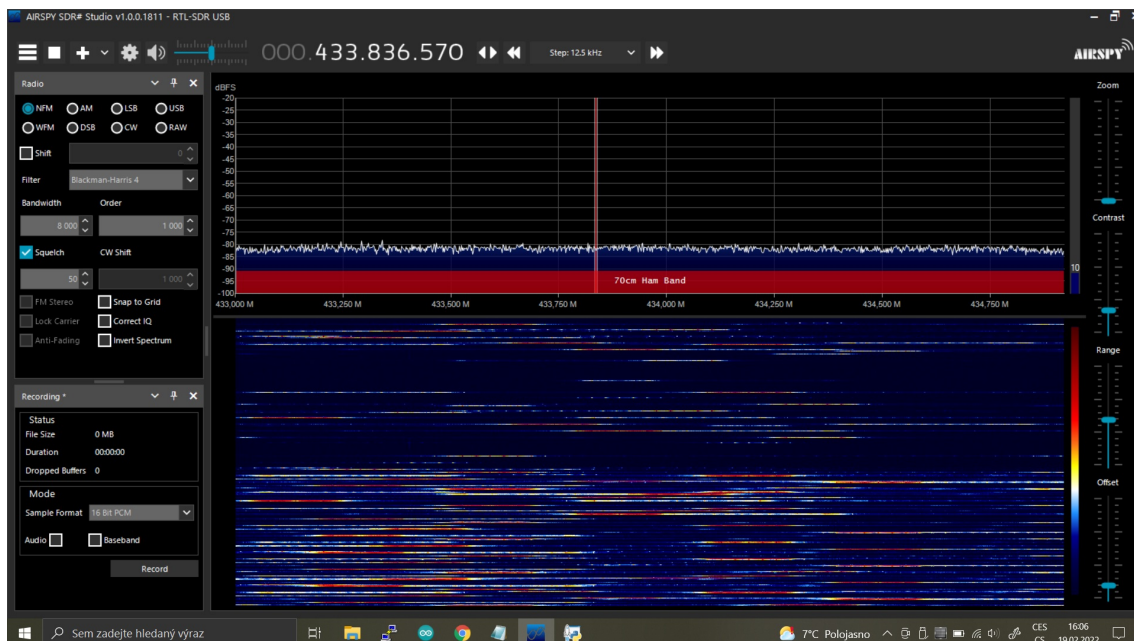
s [m]	up [%]	n [-]	chybovost [%]	Δt [ms]	v_p [B/s]
1	100	250	1.2	229.97	373.948
100	43.2	250	-	-	-
100	100	250	2.4	231.72	371.128
200	32.4	250	-	-	-
200	56.0	250	-	-	-
200	26.0	250	-	-	-
200	25.6	250	-	-	-

Tabulka 2.2. Výsledky měření přenosu s alternativním párem antén

V tabulce 2.1 můžeme vidět, že už při vzdálenosti 100 m došlo k výpadku spojení. Ve 200m vzdálenosti bylo navázání spojení velice obtížné, kompletní měření v tomto případě proběhlo pouze jednou ze šesti pokusů.

Jak můžeme vidět z tabulky 2.2, kdy byly pro experiment použity antény SW433-XP1M a SENDA RG433M, změna antén nijak zásadně přenos nezlepšila. Spojení se podařilo navázat ve všech případech, ale ve vzdálenosti 200 m se žádný přenos nepodařilo dokončit.

Obrázek 2.7 ukazuje snímek obrazovky s programem SDR Sharp, pořízený v průběhu měření. Je zde vidět FHSS metoda přenosu dat probíhající ve spodní části spektrogramu. Poté přenos najednou ustal, a to byl zároveň moment, kdy Python program přestal přijímat data. Jedná se o moment přibližně v polovině spektrogramu na časové ose, kdy se graf zbarvil tmavě modře a není zde vidět žádné vysílání ani příjem signálu. Následně opět můžeme vidět, že dochází k aktivitě na rádiovém spektru, ale signál je slabší. Ověřil jsem, že k tomuto slabšímu vysílání dochází při spárování vysílače a přijímače. Jedná se pravděpodobně o způsob, jak moduly udržují spojení (indikováno zelenou LED diodou na každém z modulů).



Obrázek 2.7. Program SDR Sharp zobrazující spektrum na okolí frekvence 433 MHz

Z tabulek je patrné, že moduly nejsou velice spolehlivé, a to již na krátkou vzdálenost do 100 m. Nejedná se o problém špatných antén, protože problémy se opakovaly i při jejich záměně za spolehlivé ověřené antény. Z tabulek a sledování spektra vyplývá, že dochází k náhodným výpadkům přenosu a ztrátě spojení mezi moduly, a je tedy tento způsob přenosu telemetrie není vhodný pro implementaci do systému přenosu dat.

Navíc jsou moduly prodávány jako pár určený pro přenos dat pouze mezi dvěma zařízeními, není tedy zaručeno, zda by bylo možné připojit v systému více leteckých zařízení najednou.

Kapitola 3

Přenos dat založený na mobilní komunikační síti

Druhou možností pro přenos dat je využití rozsáhlé telekomunikační infrastruktury pokrývající prakticky celé území ČR. Zajímavá je tato varianta především díky vysokým rychlostem přenosu dat v rámci LTE připojení, dále dostupností služby a také vysokou spolehlivostí. Mobilní internetová síť je mnohem komplexnější a pokročilejší systém přenosu dat než jednoduchý radiový vysílač a přijímač, umožňuje tedy pokročilejší způsoby přenosu, zabezpečení a kontroly přenosu.

Nevýhody této metody připojení jsou vyšší cena komponent kvůli jejich vyšší komplexitě, na což navazuje i vyšší složitost ovládání radiového modemu. Další nevýhodou je nutnost platit za přenos dat.

3.1 Arduino MKR 1400

V rámci své bakalářské práce jsem se rozhodoval, která vývojová platforma by byla vhodná pro realizaci UAV části monitorovacího systému. Jako hlavní kritéria jsem pro analýzu zvolil velikost, hmotnost a univerzálnost. V úvahu připadaly minipočítače Raspberry Pi od společnosti Raspberry. Ty by nabízely velký výpočetní výkon společně s celým operačním systémem, který by se dal dobře využít pro běh složitějších programů. Hardware však není příliš rozvinutý jako v případě jiných vývojových desek s mikrokontrolery. Minipočítače Raspberry Pi sice nabízí základní periferie vyvedené přes dva řádky kolíků s roztečí 2,54 mm, bylo by ale nutné stavět či zakoupit rozšiřující desky plošných spojů, které bývají rozměrné a dražší, než desky pro mikrokontrolery. Tyto desky by musely obsahovat nejen senzory a rozšiřující sběrnice, ale také napájecí obvody a popřípadě možnost připojení bezdrátového modemu buď přes sériové UART rozhraní nebo přes USB. Vstupní napájecí napětí počítačů Raspberry Pi je 5 V a je přiváděno pouze přes USB port¹. Nutnost přidat všechny tyto funkce by značně zvýšila rozměry a hmotnost celého palubního systému, což není pro malé letecké prostředky žádoucí.

Rozhodl jsem se proto využít vývojových desek Arduino MKR s mikrokontrolem, integrovaným modemem a napájením na jedné desce plošného spoje, která je velikostně menší, než nejmenší minipočítač Raspberry Pi Zero. Všechny desky MKR sdílí jeden formfaktor a umožňují montáž rozšiřujících modulů, takzvaných shieldů, pro přidání dalších funkcionalit. Stále je tedy potřeba zakoupit či vyrobit rozšiřující desky, avšak ty slouží pouze pro sběr dat. Těmito shieldy výrobce dosáhl vysoké modulárnosti, která umožňuje sestavit UAV část tak, aby splňovala požadavky na sběr avionických dat s minimální hmotností.

Pro testování a možnost implementace do monitorovacího systému jsem tedy vybral mikrokontrolér z rodiny MKR od italské světoznámé společnosti Arduino, která se proslavila svými široce dostupnými, levnými a jednoduše programovatelnými vývojovými

¹ <https://www.raspberrypi.com/documentation/computers/getting-started.html#setting-up-your-raspberry-pi>

deskami. Arduino MKR je jejich nová platforma mířící především na IoT aplikace. Tato platforma zahrnuje několik desek s procesory Arm® Cortex®-M0 32-bit SAMD21 a různými možnostmi pro připojení desky k internetu, například přes WiFi, LTE NB, GSM/3G nebo nízkenergetický nízkobjemový rádiový protokol Lo-Ra s dlouhým dosahem. Původně jsem chtěl použít desku MKR s možností připojení na LTE, bohužel však vzhledem k probíhající křemíkové krizi nebyla dostupná k zakoupení. Pro svou práci jsem tedy použil desku Arduino MKR 1400[7] s modemem podporujícím 3G/2G standardy, kterou jsme zakoupili již v roce 2021 před tím, než došlo k upuštění od standardu 3G. Pro svou práci jsem byl tedy limitován standardem 2G pro přenos dat. Díky tomu, že tento standard je používán dlouhou dobu, je 2G síť přístupná prakticky odkudkoliv na území ČR[10] a z výsledků měření v sekci 3.3 by měla poskytnout dostatečně rychlé připojení.

Arduino MKR 1400 je deska s integrovaným GSM/3G modemem a slotem pro Micro SIM. Jedná se o modem SARA-U201 od společnosti u-blox[8]. Modem podporuje připojení do kterékoliv mobilní sítě na světě. Jeho pracovní pásma a přenosové standardy jsou v tabulce 3.1.

Standard	3G	2G
	UMTS	GSM
	HSDPA	GPRS
	HSUPA	EDGE
Pásma		
	Band 19 (800 MHz)	GSM 850 MHz
	Band 5 (850 MHz)	E-GSM 900 MHz
	Band 8 (900 MHz)	DCS 1800 MHz
	Band 2 (1900 MHz)	PCS 1900 MHz
	Band 1 (2100 MHz)	
	Band 19 (800 MHz)	

Tabulka 3.1. Pásma a přenosové standardy podporované modemem U-blox SARA-U201 jak uvedeno v datasheetu [9].

K dispozici jsem obdržel SIM kartu od operátora Vodafone, který provozuje 2G síť na pásmech E-GSM 900 MHz a DCS 1800 MHz. [11]

Za účelem práce s modemem byla společností Arduino vydána knihovna MKR-GSM[12], jež zjednodušuje komunikaci s modemem. Modem je ovládán pomocí AT příkazů přímo z mikrokontroléru přes sériové rozhraní. Pro nastavení správných pásem na modemu jsem použil ukázkový program `BandManagement.ino`, který konfiguruje modem dle volby uživatele na sériovém rozhraní. Konfigurace pásma zůstane uložena v paměti modemu i po jeho vypnutí. Po nastavení nas pásma 900 MHz a 1800 MHz se modem v pořádku připojil k mobilní datové síti.

Balení Arduino MKR 1400 neobsahuje od výrobce anténu, deska je však vybavena konektorem micro UFL pro připojení externí antény. V průběhu své práce jsem vyzkoušel dvě antény, obě laděny na stejná pásma 700 MHz až 960 MHz a 1710 MHz až 2700 MHz. První anténa byla GSM-ANT-FPC02[13], vyrobená z měkkého materiálu s jednou samolepicí stranou pro jednoduché permanentní uchycení. Druhá anténa DB-A0125-GSM&4G byla výrobcem vyrobená na desce plošného spoje[14]. Jediný rozdíl, který jsem v rámci své práce na projektu zaznamenal byl, že přihlášení k mobilní síti bylo

rychlejší s druhou anténou, pravděpodobně díky její vyšší citlivosti. Jinak se antény z praktického hlediska ničím nelišily.

3.2 Rozšiřující desky

Pro platformu Arduino je velice populární funkcí možnost připojit rozšiřující desky, takzvané shiely. Jedná se o desky plošných spojů různých tvarů a velikostí, které poskytují rozšiřující periferie, senzory, řadiče, displeje a mnoho dalšího. Velice populární se staly díky jednoduché montáži k desce Arduino společně s širokým výběrem knihoven potřebných pro práci s nimi. V případě rodiny MKR jsou shiely průchozí, je tedy možné poskládat několik desek vertikálně na sebe. Tyto desky jsou propojeny tak, aby odebíraly napájecí napětí z desky mikrokontroleru a zároveň využívají sběrnic pro komunikaci, nejčastěji SPI, I2C a sériové rozhraní UART. Průchodnosti těchto desek jsem využil pro tvorbu vlastní desky, která slouží jako základní deska pro montáž do leteckého prostředku.

3.3 Testování přenosu a výsledky měření

Pro otestování rychlosti a spolehlivosti přenosu nebylo v tomto případě nutné tak rozsáhlé přípravy jako v předchozím případě, a to díky tomu, že modem pracuje na vyšší vrstvě ISO/OSI modelu a bylo možné využít příkazu ping.

Pro účely testování jsem napsal program pro arduino, založený na ukázce `GPRSPing.ino` z knihovny `MKRGSM`[12] nainstalovaný do Arduino IDE. Program se připojil k mobilní síti a následně začal posílat ICMP paket s velikostí 86 bytů na adresu `www.google.com`. V době testování spojení jsem ještě neměl k dispozici USB modem se statickou IP adresou pro serverovou část systému a ICMP paket nemohl projít přes mého internetového poskytovatele až do domácí sítě, proto jsem jako alternativu zvolil cílovou adresu serveru google. Velikost zprávy byla stejná jako délka zpráv odesílaných při testování radiových modulů. Program funguje velice obdobně jako program vytvořený v Pythonu pro testování radiových modulů. Je zaznamenán počet úspěšných a neúspěšných přenosů, vypočtena průměrná doba odezvy a průměrná rychlost přenosu. Jelikož se jedná o pakety vytvořené v modemu a neposílají se žádná skutečná data, program netvoří soubor se zprávami, ale pouze zapisuje výsledky na sériovou linku.

Vzhledem k dobrému plošnému pokrytí mobilním signálem prakticky po celém území ČR jsem test provedl z domova, ICMP paket se odesílal každou půl vteřinu, celkově měření trvalo přibližně 8 minut.

V následující tabulce jsou výsledky měření, kde průměrná doba přenosu je značena jako Δt , průměrná rychlost přenosu jako v_p a počet zpráv n :

Číslo měření	n	chybovost [%]	Δt [ms]	v_p [B/s]
1	1000	0	191.77	456.10
2	1000	0	288.88	321.20
3	1000	0	151.90	595.95
4	1000	0	289.91	304.93
5	1000	0	261.74	356.66
6	1000	0	151.38	648.49

Tabulka 3.2. Výsledky měření provedené přes ICMP pakety

Kapitola 4

Realizace systému pro přenos telemetrie

Vzhledem k výsledkům měření v předchozích kapitolách jsem se rozhodl postavit systém přenosu telemetrie na mobilní síti za pomoci vývojové desky Arduino MKR 1400. To mi umožnilo přenos dat na prakticky neomezenou vzdálenost díky dobrému pokrytí mobilním signálem. Rychlost je omezena pouze na 2G, protože pásma 3G byla uvolněna v rámci renovace radiového pásma a implementace o poznání rychlejšího standartu 4G mobilními operátory. Vypnutí sítě 3G proběhlo v průběhu roku 2021. [15]

Systém se skládá ze dvou částí. První zařízení popsané v sekci 4.1 je umístěné na palubě leteckého prostředku a má za úkol vyčítat, shromažďovat a odesílat data ze senzorů. Druhá část je pozemní stanice s popisem v sekci 4.4 je určena pro příjem dat, jejich uložení pro pozdější zpracování a zároveň zobrazení v reálném čase.

UAV část sbírá a odesílá následující hodnoty

- GNSS pozice,
 - zeměpisná délka,
 - zeměpisná šířka,
- nadmořská výška,
- rychlost,
- napětí baterie.

Ke sběru těchto dat jsem použil rozšiřující desky MKR popsané v sekci 4.2. Z těchto shieldů jsem navíc sbíral také následující data

- vlhkost,
- teplota,
- okolní osvětlení,
- GNSS čas.

Dále je možné získávat data z externích periférií přes sběrnici RS232 implementovanou na základní desce UAV části nebo přes sběrnici CAN díky shieldu popsáném v sekci 4.2.2.

Maximální frekvence odečítání a zasílání dat do základnové stanice byla uvažována 2Hz, což je pro monitorovací systém relativně hodně. U většiny monitorovacích systémů je perioda odesílání dat i několik sekund. Velikost jednoho paketu s odesílanou hodnotou je okolo 20 bytů. Odhad velikosti zprávy je zde nižší než délka testovacích zpráv z důvodu jiné struktury zprávy danou Telegraf aplikací popsanou v sekci 4.4.3. Minimální počet paketů n je dán počtem odesílaných proměnných. Pro odesílání pouze základních hodnot uvedených v prvním seznamu je $n=5$. Jednoduchým výpočtem získáme minimální rychlost přenosu v_m :

$$v_m = 20 * 5 * 2 = 200 \quad [B/s, B/s, B/s]$$

Arduino modem využívá síť 2G, která i přes obecně nižší přenosové rychlosti, poskytuje dostatečně rychlé připojení pro přenos malých objemů dat používaných pro tento systém při teoretické minimální potřebné rychlosti 200 B/s a nejnižší naměřené minimální rychlosti 304.93 B/s. Minimální naměřená přenosová rychlost je tedy dostatečná i s rezervou pro data z dalších senzorů. Reálná naměřená data se posílají v UDP paketech.

Data se dále také ukládají na mikro SD kartu pro případné rychlé zpracování v místě měření na počítači a SD karta takto funguje i jako záloha.

4.1 UAV část

UAV část je zařízení umístěné na palubě leteckého prostředku, které sbírá avionická data a odesílá je do základnové stanice. Celá tato část je založena na Arduino MKR 1400, jež bylo popsáno v kapitole 3.1. Tato část sestává ze základní desky, jež poskytuje způsob uchycení do leteckého prostředku, dále rozšiřuje možnost připojit více externích periférií za pomoci převodníku ze sériové linky s napěťovými úrovněmi mikrokontroleru na RS232 komunikační rozhraní, které se v letectví běžně využívá. Další možnost připojení externí periferie je přes rozhraní CAN, také často využívané v dopravních prostředcích a všude tam, kde je potřeba robustní sběrnice odolná vůči vlivům elektromagnetického rušení.

4.2 Použité rozšiřující desky

Jak již bylo uvedeno, rodina vývojových desek Arduino MKR má možnost připojit další desky, které přidávají funkce, periférie a další prvky. Každý shield je stavěn tak, aby se dal přišroubovat ve čtyřech bodech a to buď ze shora nebo ze spod. Použit lze jakoukoliv kombinaci distančních sloupků, šroubků a matek se závitem M2. Elektrické propojení je realizováno průchozími dutinkovými lištami.

Pro sběr dat o vlastnostech prostředí jsem použil MKR ENV shield s integrovanými MEMS senzory, shield je více popsán v kapitole 4.2.1. Vyčítání dat probíhá za pomoci knihovny MKRENV[16], která se stará o komunikaci se senzory na ENV shieldu. K dispozici je také slot pro mikro SD kartu. Zápis na ní probíhá přes sběrnici SPI pomocí knihovny SD.

4.2.1 MKR ENV Shield

Tento shield je určen pro sběr dat o okolním prostředí za pomoci elektronických senzorů a možnosti zápisu či čtení z SD karty[17]. Deska obsahuje následující senzory:

- senzor tlaku LPS22HB

MEMS senzor od firmy ST s napájecím napětím v rozmezí 1.7 V až 3.6 V. Komunikace je možná přes SPI či I2C s tím, že knihovna MKRENV využívá komunikaci přes I2C, na kterou je senzor nastavený propojením na desce. Senzor využívá pro měření tlaku membránu vyrobenou z piezoeletrického krystalu a napětí odpovídající tlaku je vyčítáno 24-bitovým analogově-digitálním převodníkem. Pro vyčítání teploty je v senzoru 16-bitový převodník. Frekvence odečítání hodnot je nastavitelná v rozmezí mezi 1 Hz až 75 Hz. Knihovna MKRENV inicializuje senzor v módu ONE_SHOT, senzor v tomto módu data neposílá sám od sebe s určitou frekvencí, ale pouze provede měření na vyžádání mikrokontroleru. Absolutní chyba pro měření tlaku v rozmezí 800 hPa až 1100 hPa je ± 0.1 hPa při 25°C, při přepočtu na barometrickou výšku by

toto odpovídalo absolutní chybě ± 0.8 m a výškovému rozsahu přibližně -700 m až 2000 m. [18]

- přesný senzor vlhkosti a teploty HTS221

Přesný senzor pro měření vlhkosti a teploty s přesností ± 0.5 °C. Senzor má vestavěný 16-bitový převodník jež převádí data o teplotě a vlhkosti pro odeslání přes sběrnici. Čip je na desce plošného spoje nastaven na komunikaci přes I2C sběrnici a může automaticky odesílat data s frekvencí od 1 Hz do 12.5 Hz. Čtení je nastaveno při inicializaci na One-shot, odečítání hodnoty proběhne pouze na požadavek mikrokontroleru. Senzor při měření aplikuje automaticky kalibrační data nahraná do senzoru výrobcem.[19]

- senzor okolního osvětlení TEMT6000

Jedná se o senzor určený pro detekci okolního světla. Umožňuje například sledovat osvětlení během dne a určit, zda je zataženo. Tento senzor neposkytuje data přes sběrnici, ale jako analogovou hodnotu. Konstrukčně se jedná o fototranzistor uzavřený v průhledném pouzdře. Dle výrobce je nejvyšší citlivost na vlnové délce 570 nm. Závislost citlivosti na vlnové délce by se dle výrobce měla shodovat s lidským okem. Rozlišení senzoru je dáno analogově digitálním převodníkem Arduino MKR 1400, který má rozlišení 12-bitů. Pro tento senzor je dedikován analogový pin A2.[20]

Výpočet barometrické výšky je dán vzorcem

$$h = \frac{T_1}{\alpha} \cdot \left(\frac{p}{p_1} \right)^{\frac{-\alpha R}{g_0}} - 1 + h_1$$

kde T_1 je teplota ve počáteční výšce h_1 ve °C, g_0 je konstanta gravitačního zrychlení s hodnotou $9,81ms^{-2}$, p_1 je počáteční tlak ve výšce h_1 , α je konstanta teplotního gradientu s hodnotou $-0.0065Km^{-1}$, R je univerzální plynová konstanta s hodnotou $287.06Jkg^{-1}K^{-1}$. Pro výpočet výšky nad mořem jsem za hodnotu h_1 jsem dosadil 0, p_1 je normální tlak $1013.25hPa$ při teplotě T_1 rovné 15 °C. [21]

Rozšiřující deska je napájena z Arduino MKR 1400. Obsahuje také slot pro Mikro SD kartu, jež je připojena přes SPI sběrnici. Na ni pomocí knihovny `SPI.h` a `SD.h` ukládám veškerá data jako zálohu ve formátu CSV. Tento formát jsem zvolil kvůli jeho jednoduché struktuře, který umožňuje jednoduše přidat a odebrat proměnné zapisované mikrokontrolerem do souboru. Je však možné zapisovat i do souboru `.bin` nebo `.txt` a struktura zápisu dat je libovolná dle naprogramování mikrokontroleru.

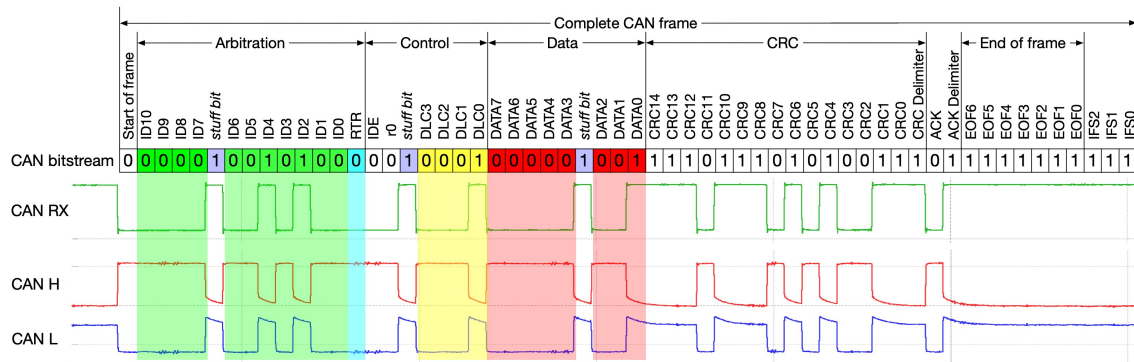
■ 4.2.2 MKR CAN shield

Tento shield zprostředkovává rozhraní pro připojení CAN zařízení. [22] CAN je velice rozšířená robustní sběrnice. Uplatnění našla především v automobilovém průmyslu, ale využívána je i v jiných odvětvích, především tam, kde je potřeba spolehlivé spojení mezi zařízeními na delší vzdálenost v řádu metrů až desítek metrů.

Sběrnice využívá dva vodiče pro přenos dat značené CANH a CANL. Napětové úrovně jsou symetrické a dominantní bit je logická 0, ustupující neboli recesivní bit je logická 1. Tímto je sběrnice zabezpečena proti kolizím, protože pokud se na sběrnici nachází logická 1, vysílač posílající logickou nulu zkratuje napětí na sběrnici přes rezistor a tím je odeslána logická 0.

Každé zařízení připojené ke CAN sběrnici má svoje identifikační číslo. Zpráva obsahuje identifikátor daného zařízení pro které je určena, tento identifikátor je prvních 11 bitů zprávy. Poté ve zprávě následuje RTR bit, který udává, zda-li se jedná o datovou zprávu či žádost o zaslání datové zprávy. Následuje kontrolní část o 6 bitech,

dva jsou k dispozici k budoucímu rozšíření komunikačního protokolu, další 4 udávají délku zprávy. Dalších 8 bitů je vyhrazeno pro přenášená data. Pro kontrolu celistvosti a správnosti přijaté zprávy používá sběrnice cyklický redundantní součet CRC-15. Následující dva bity slouží pro potvrzení přijetí zprávy příjemcem. Zpráva je zakončena sedmi recesivními bity. [23]



Obrázek 4.1. Struktura zprávy CAN¹

Na desce shieldu jsou od výrobce implementovány následující integrované obvody:

- kontrolér převodu SPI na CAN MCP2515

Čip umožňující Arduino MKR použít svou sběrnici pro posílání a příjem CAN zpráv. CAN kontrolér má rozsah napájecího napětí 2.7 V až 5.5 V. Nejedná se ale o vysílací čip, pouze o řadič s integrovanou logikou pro správné vytvoření struktury CAN zprávy, kontrolu integrity zprávy a práci s vnitřními buffery. Dále umožňuje příjem zpráv a jejich překlad tak, aby zprávy mohli být přeposlány přes SPI sběrnici do mikrokontroleru. Kontrolér disponuje výstupním interrupt pinem pro obsluhu přerušování a předání přichozících dat přes SPI. Tím, že je na desce použit tento kontrolér je ušetřen výpočetní výkon na Arduino MKR. [24]

- Vysílač/přijímač TJA1049

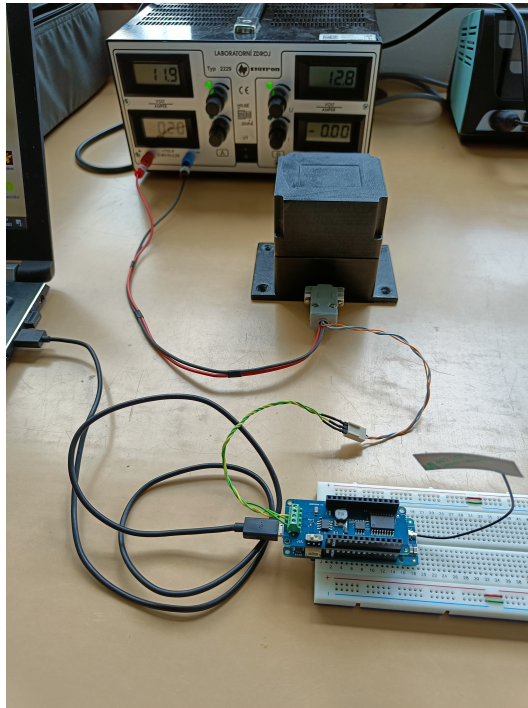
Jedná se o čip určený pro vysílání a příjem signálů po CAN sběrnici. Na rozdíl od MCP2515 je schopný přijímat na vstupu až ± 27 V. Čip se dále stará o časování signálů a jejich obousměrný převod mezi CAN sběrnici a CAN kontrolerem MCP2515. Pro napájení obvodu je použito napětí 3.3 V, pro CAN signál je do čipu přivedeno napětí 5 V. [25]

- Step-down měnič TPS54232 Na desce CAN je také malý zdroj jehož vstup je připojen na jednu ze svorek pro CAN sběrnici. Zdroj je zapojen jako Step-down, tedy mění větší vstupní napětí na menší výstupní napětí. V případě tohoto zdroje a tedy i desky CAN může být vstupní napětí až 28 V, výstupní napětí je pevně nastaveno na 5 V a připojeno na Arduino MKR vstupní pin Vin. Spínací frekvenci 1MHz a dokáže dodat výstupní proud až 2 A, což je i se všemi shieldy pro Arduino více než dostačující.[26]

Pro přijímání dat z externího senzoru přes CAN sběrnici mi byla zapůjčena inerciální jednotka s integrovaným komunikačním rozhraním CAN. Jednotka odesílá data na vyžádání tím, že se na zařízení z identifikátorem 1 odešle prázdná zpráva. Jednotka následně odešle na sběrnici data ze svých senzorů s nastavenou rychlostí 1 Mbit/s.

¹ https://kentindell.github.io/2020/01/03/canframe_py_tool/

Naprogramoval jsem arduino tak, aby odeslalo zprávu s ID 1 a četlo zprávy do paměti. Po ukončení příjmu jsem data vypsal na sériovou linku počítače. Ukázka příjmu dat je na obrázku 4.3 a celé testovací zapojení je na obrázku 4.2.



Obrázek 4.2. Testovací zapojení komunikace s IMU jednotkou přes sběrnici CAN

Jednotka je napájena 12V pomocí laboratorního zdroje. I přes možnost propojit sběrnici pouze dvěma vodiči jsem použil ještě třetí vodič k propojení zemí.

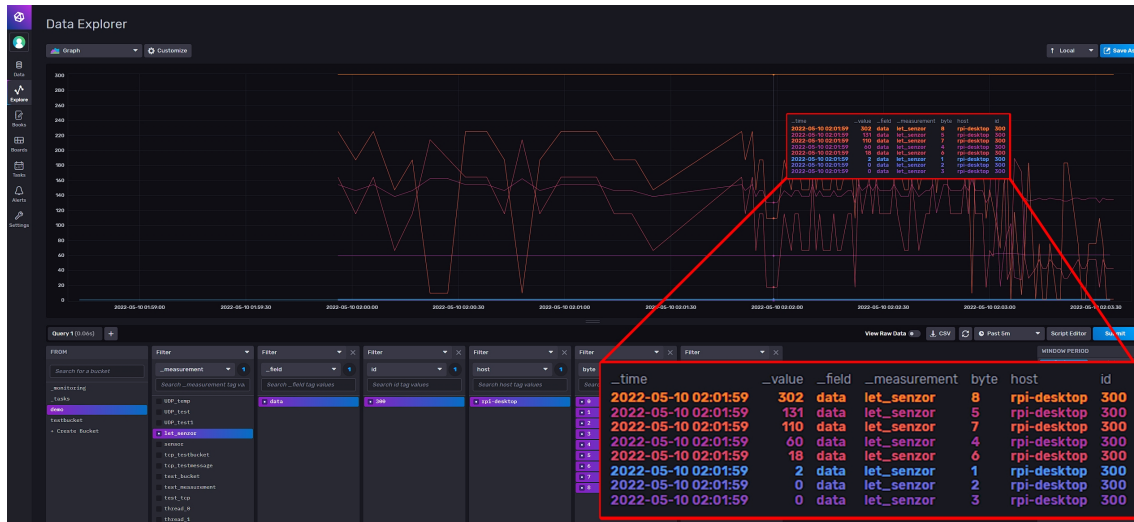
```

COM4
12:56:20.568 -> CAN Receiver
12:56:20.568 -> CAN initialized
12:56:21.584 -> Requesting data.
12:56:21.584 -> Recieved data:
12:56:21.584 -> 300 1 2 0 0 188 3 18 110
12:56:21.584 -> 302 1 2 0 0 191 127 190 118
12:56:21.584 -> 303 1 2 0 0 189 204 204 204
12:56:21.584 -> Done
12:56:21.584 -> -----
12:56:22.573 -> Requesting data.
12:56:22.573 -> Recieved data:
12:56:22.573 -> 300 1 2 0 0 188 19 116 188
12:56:22.573 -> 302 1 2 0 0 191 127 190 118
12:56:22.573 -> 303 1 2 0 0 190 51 51 51
12:56:22.573 -> Done
12:56:22.573 -> -----
12:56:23.604 -> Requesting data.
12:56:23.604 -> Recieved data:
12:56:23.604 -> 300 1 2 0 0 188 3 18 110
12:56:23.604 -> 302 1 2 0 0 191 127 190 118
12:56:23.604 -> 303 1 2 0 0 189 230 102 102
12:56:23.604 -> Done
12:56:23.604 -> -----
  
```

Automatické scrollování
 Zobrazit časové razítko
 Obojí NL & CR
 250000 baudů
 Vymazat výstup

Obrázek 4.3. Ukázka příjmu dat z inerciální jednotky přes sběrnici CAN a jejich výpis na sériovou linku

Jako další krok jsem tento výčet použil pro tvorbu zpráv pro databázi popsanou v sekci 4.4.2. ID zprávy slouží jako tag pod kterým se ukládá 8 separátních časových řad, každá řada obsahuje hodnoty z dané pozice bytu ve zprávě.



Obrázek 4.4. Data z inerciální jednotky zobrazené jako časové řady v databázi, jeden graf vždy odpovídá jednomu byte ze zprávy od inerciální jednotky. V pravém dolním rohu je přiblížení na přijmutá data.

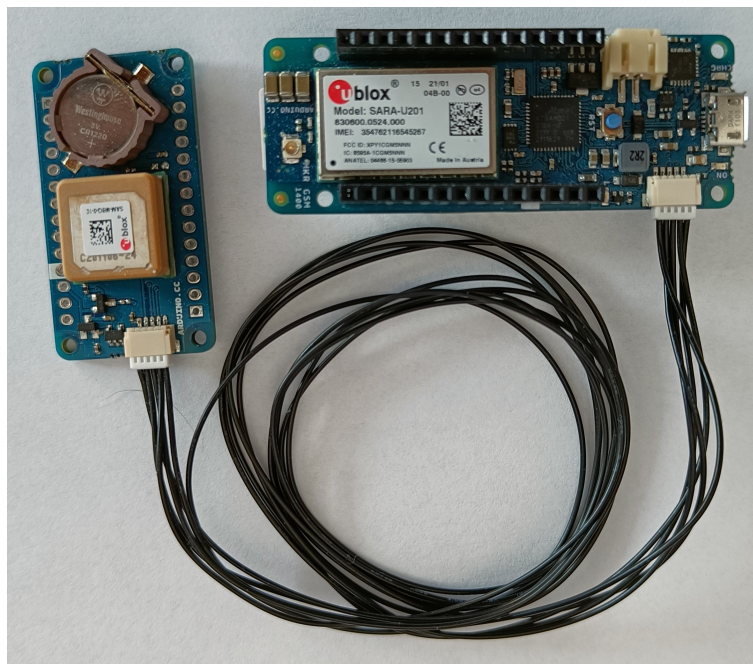
4.2.3 MKR GPS Shield

Jedná se o desku pro integraci GNSS funkce pro Arduino MKR.[27] Deska plošného spoje obsahuje miniaturní GNSS přijímač SAM-M8Q [28] od společnosti U-Blox. Tento modul má integrovanou anténu s nízkošumovým zesilovačem a SAW filtr pro zaručení kvalitního příjmu na GNSS frekvencích v okolí 1.575 GHz. Jako mnoho jiných GPS přijímacích modulů, má i tento modul studený a horký start. Studený start nastává, když je modul zapnut bez předchozího napájení. Při tomto startu může trvat až 30 sekund než modul začne posílat data o poloze přes sběrnici. Přijímač podporuje poziční systém GPS, GLONASS a Galileo. Při horkém startu, například při krátkodobé ztrátě signálů ze satelitů, je modul schopen opětovně získat svou pozici do 1-2 sekund. [29]

Přesnost horizontální polohy je 2.5 m pro systém GPS, 8 m pro systém GLONASS a 2.5 m pro kombinaci GPS a GLONASS. Přesnost pro systém Galileo zatím není známá, jelikož je tento systém stále budován. [29]

Shield dále obsahuje slot pro jednu baterii typu CR1220. Baterie je připojena na záložní vstup GNSS přijímače, ale neslouží k přímému napájení. Je použita pro běh interního oscilátoru a napájení RAM, což umožňuje rychlý náběh modulu po jeho vypnutí ztrátou napájecího napětí.

Modul nabízí dva druhy komunikace, a to přes sériovou linku UART nebo přes Display Data Channel, který je založen na protokolu I2C. Deska disponuje JST ESLOV konektorem sloužícím pro přivedení napájení, SDA, SCL a EXTINT pro komunikaci po sběrnici I2C. Výhoda spočívá v možnosti upevnit modul v místě lepšího signálu, protože shield nedisponuje konektorem pro připojení externí antény. [29]



Obrázek 4.5. GPS shield propojený s Arduino MKR 1400 přes sběrnici Display Data Channel pro jednodušší montáž modulu

Pro ovládání modulu jsem využil Arduino MKR knihovnu `MKRGPS`, která nabízí základní konfiguraci a výčet dat o poloze, rychlosti, čase a počtu satelitů. Tyto hodnoty jsem vyčítal pro další zpracování.

4.3 Základní deska

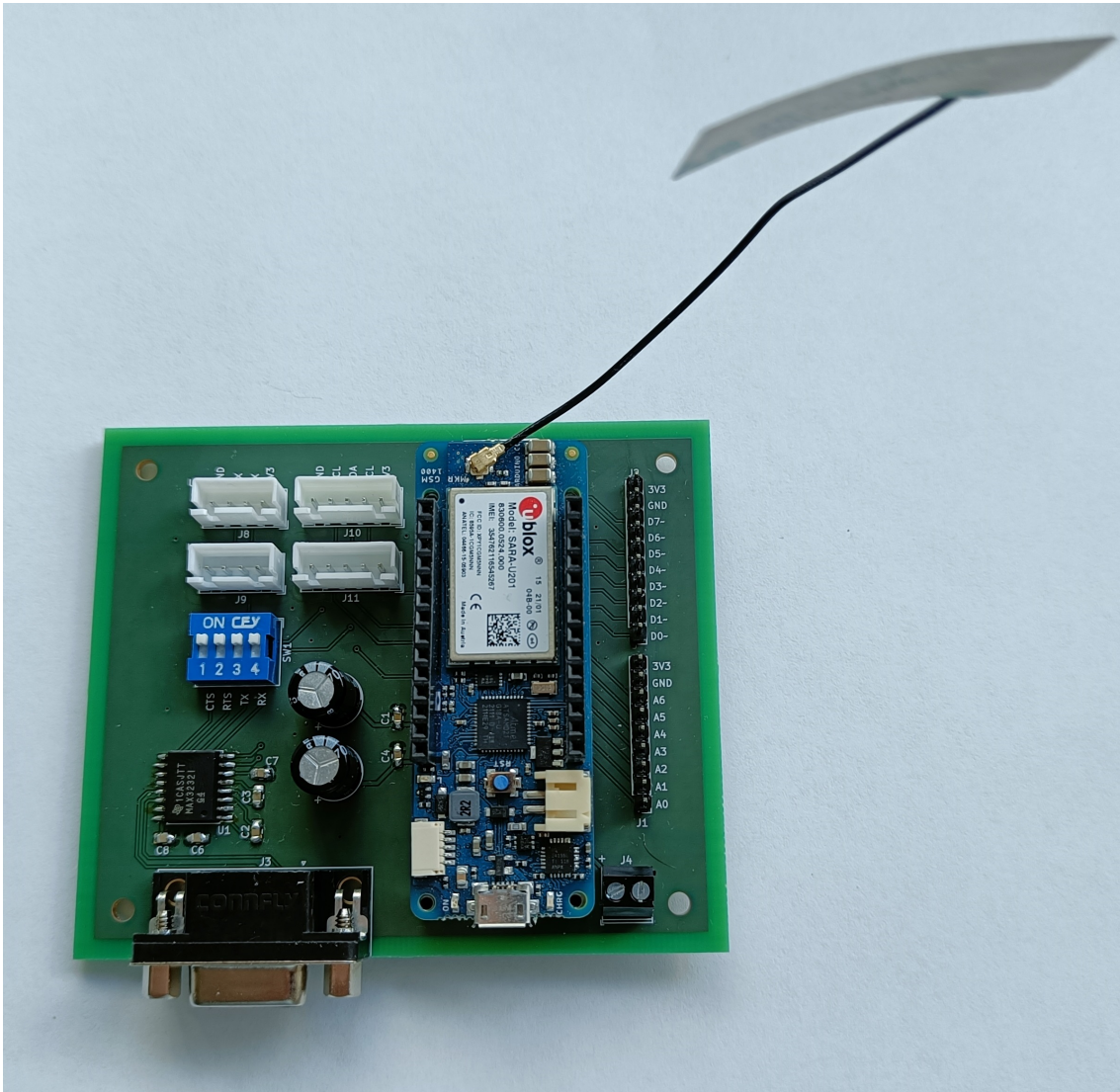
Pro UAV část jsem navrhl dvouvrstvou desku plošného spoje. Tato deska bude sloužit k montáži do UAV prostředku a dále bude sloužit jako rozšiřující platforma. Deska má velikost 94 mm na 80 mm a pro montáž je v každém rohu umístěna díra pro šroub s M3 závitem.

Deska dále obsahuje čtyři díry pro M2 šrouby umístěné tak, aby bylo možné přichytit na desku Arduino MKR 1400. Dále jsou zde umístěny dvě dutinkové lišty pro elektrické propojení z Arduino MKR do ostatních částí desky. Na pravé straně desky jsou umístěny dvě řady kolíkových pinů vedle sebe. Jedna řada má vyvedeny analogové piny Arduino MKR, druhá řada digitální vstupně výstupní rozhraní. Obě řady navíc obsahují jeden pin se zemí a jeden napájecí pin s napětím 3.3 V. Deska dále obsahuje dva pětipinové JST konektory se SPI sběrnici a napájecím napětím 3.3 V. Pro rozšíření možností komunikace jsem také na desku přidal dva čtyřpinové JST konektory pro sériovou komunikaci, zahrnul jsem opět napájecí napětí 3.3 V.

Pro další možnost napájení jsem na desku umístil svorkovnici s roztečí 3,5 mm. Napětí je přivedeno na pin Arduino MKR s názvem Vin.

V rámci testování jsem narazil na problém, kdy registrace modemu v mobilní síti vyžaduje značné množství energie. Komunikace po přihlášení již tolik energie nevyžaduje, ale občas se při přihlašování stalo, že náhlý proudový odběr způsobí pokles vnitřního napájecího napětí a Arduino MKR se restartuje. Umístění elektrolytického kondenzátoru paralelně k napájecímu napětí 5 V tento problém odstranilo, proto jsem na desku umístil elektrolytický a keramický kondenzátor na napájecí napětí 5 V i 3.3 V. Tato kombinace byla zvolena kvůli faktu, že elektrolytické kondenzátory mají sice větší kapacitu, ale

také větší vnitřní odpor. Keramické kondenzátory mají tyto vlastnosti přesně naopak, proto se v mnoha elektronických návrzích používá tato paralelní kombinace. Alternativně funkci kondenzátorů může zastat baterie připojená do JST konektoru umístěném na Arduino MKR.



Obrázek 4.6. Osazená základní deska s Arduino MKR 1400

■ 4.3.1 Napájení

Celou UAV část systému lze napájet několika způsoby:

- Svorkovnice na základní desce

Tento vstup je přiveden na spínaný zdroj ovládaný čipem BQ24195L umístěný na desce Arduino MKR. Na tento vstup je možné přivést pouze napětí v rozmezí 5 V až 17 V.
- CAN shield

Jak bylo popsáno v kapitole 4.2.2, je možné použít CAN shield pro napájení UAV části. Vstupní napětí je až 28 V. Výstupní napětí tohoto shieldu je přivedeno na vstupní napájecí pin V_{in} .
- JST konektor pro baterii

Konektor je určen pro připojení lithiové baterie s nominálním napětím 3.7 V. Baterie je nabíjena již zmiňovaným čipem BQ24195L, který se stará o kontrolu podvybití a přebití. Baterie je nabíjena i v případě, kdy je UAV část napájena přes CAN shield. Baterie je interně připojena na odporový dělič s výstupem přivedeným na analogový pin ADC_BATTERY. [31]

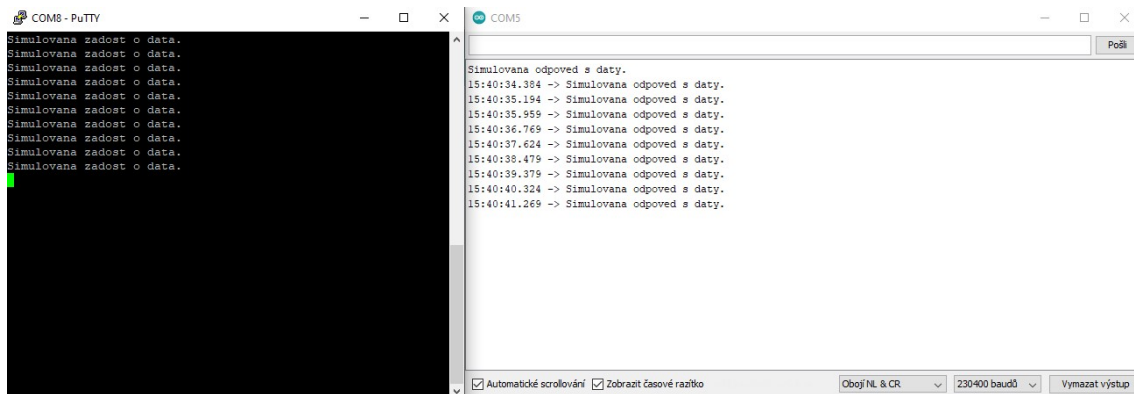
4.3.2 RS232

Jedná se o plně duplexní, sériový typ přenosu dat mezi dvěma zařízeními. Oba směry přenosu probíhají přes vlastní vodiče, nikdy tady nemůže dojít ke kolizi dat. Dále jsou k dispozici další vodiče pro další funkce jako oznámení, že zařízení je připraveno k přijetí dat, ale tyto vodiče se dnes moc nevyužívají. Díky pokrokům v technologickém vývoji a modernizacím komunikačních protokolů postačí k přenosu dat vodiče RX, TX a zem pro napěťovou referenci. Napětí na vodičích RX a TX se může pohybovat kdekoliv mezi $\pm 15\text{V}$, Arduino tedy nelze připojit k vodičům přímo i přes to, že má vývody pro sériovou komunikaci. Logická 1 je v RS232 definována jako napětí mezi hodnotami -3 V a -15 V , logická 0 jako napětí mezi 3 V a 15 V .

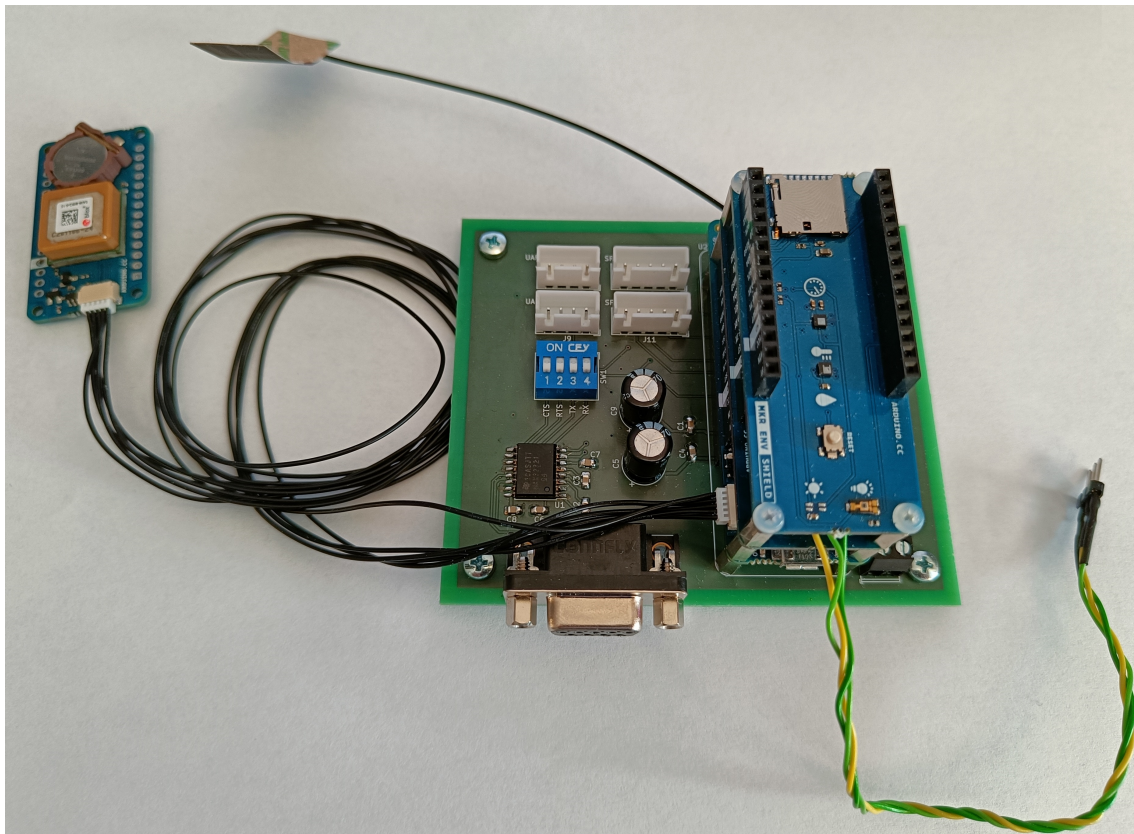
Pro napěťový převod signálů jsem použil čip MAX232, který převádí RS232 napěťové úrovně na TTL a naopak. Konkrétně jsem použil čip od společnosti Texas Instruments s celým názvem MAX3232IDW. Má volba konkrétního čipu byla omezená pracovním napětím mikrokontroleru Arduina MKR. Napájecí napětí pro tento čip jsem použil 3.3 V . U většího napětí by mohlo dojít při komunikaci k poškození mikrokontroleru. Pro tento převodník není nutné přivádět symetrické napájecí napětí, protože má integrovanou nábojovou pumpu a výstupní RS232 napěťové hodnoty je schopný si vytvořit. Je pouze nutné k tomuto čipu připojit několik keramických kondenzátorů o velikosti 100 nF , jak je uvedeno v datasheetu.

V rámci návrhu základní desky jsem zahrnul u RS232 také vodiče pro RTS a CTS. Dále jsem mezi Arduino a MAX232 umístil DIP přepínač pro každý vodič, aby bylo možné odpojit komunikaci s MAX232 v případě, kdy komunikace přes RS232 není používána. K fyzickému propojení jsem na desku umístil konektor DSUB-9. Šroubovací uchycení na stranách zajistí, že nedojde k náhodnému odpojení v průběhu letu například kvůli vibracím leteckého prostředku.

Přenos dat přes RS232 sběrnici jsem otestoval v laboratoři pomocí převodníku RS232-USB, který se po připojení k počítači zobrazí jako sériový COM port. Arduino MKR 1400 má dvě sériová rozhraní, jedno je použito pro komunikaci přes USB, druhé je vyvedeno jako dva vodiče s napěťovými úrovněmi 3.3 V , které jsou propojeny s čipem MAX232. Komunikace byla úspěšně otestována na následujících baud rychlostech: 9600, 19200, 38400, 57600, 74880, 115200, 230400. Při testování na rychlosti 250000 baud docházelo k občasným chybám v přenášených znacích.



Obrázek 4.7. Nalevo program PuTTY s otevřeným sériovým portem zobrazující přijatou zprávu odeslanou přes arduino. Napravo sériový monitor Arduino IDE s přijatou zprávou odeslanou z programu PuTTY.



Obrázek 4.8. Zkompletovaná a sestavená UAV část s Arduino MKR 1400 a dvěma shieldy společně s GNSS deskou připojenou přes sběrnici

4.4 Základnová stanice

Tato část systému pro přenos dat a telemetrie se stará o zobrazení v reálném čase, uložení a export pro další analýzu. Původní nápad byl, že se mělo jednat o program běžící na počítači, do kterého by byla data odesílána rovnou. Připojení by probíhalo přes internet v reálném čase a aplikace na počítači by zobrazovala příchozí hodnoty, prováděla výpočty a ukládala data na disk.

na sebe vázanou pevnou IP adresou, která je nutné k tomu, aby na ni UAV část mohla posílat data.

Pro práci s více linuxovými terminály najednou jsem použil balík `screen`[38]. Ten umožňuje běh více příkazových řádků najednou, a to i na pozadí. Přístup k datům v databázi je přes webové rozhraní, ale k pokročilejšímu nastavení databáze a prostředníka je potřeba příkazová řádka, do které se také vypisují konkrétnější problémy či chyby, které by popřípadě nastaly.

Pro nastavení firewallu jsem použil příkaz `ufw` pro otevření potřebných portů[39]. Server má otevřený port 22 pro možnost připojení přes SSH. Dále má otevřený port 8094 pro příjem UDP paketů a port 80, který je využíván pro webové rozhraní InfluxDB.

■ 4.4.2 databáze InfluxDB

Jako databázi pro uložení dat jsem zvolil Open-source řešení InfluxDB[40]. Jedná se o databázi časových řad, která je optimalizována pro příjem dat v čase, jejich zobrazení v grafické podobě a export pro další zpracování.

Uložení dat v InfluxDB je v jistém ohledu složitější díky velice univerzální stromové struktuře uložení dat zvané `time-structured merge tree` [41] se zkratkou `TSM Tree`. Struktura tabulky v InfluxDB je stavěna tak, že obsahuje pět základních sloupců se jmény začínajícími podtržítkem.[42]

■ `table`

Název tabulky, v databázi se používá název `Bucket` neboli kbelík

■ `_time`

Čas ve kterém byl záznam pořízen. Tato hodnota může být zahrnuta v příchozí zprávě ve formě unix časového formátu. Databáze dokáže přijmout časové razítko s přesností až na nanosekundy. Je však doporučeno používat hrubší časové rozlišení pokud je to možné. Pro své nastavení databáze jsem použil rozlišení jedné sekundy. Pokud není čas zahrnut ve zprávě, databáze přiřadí záznamu čas přijetí.

■ `_value`

Hodnota pro uložení do databáze. Databáze podporuje 64-bitové hodnoty typu `int`, `uint`, `IEEE-754 float`, dále pak `String` a `Bool`.

■ `_field`

Pole slouží jako název pro hodnoty `_value`. Podle `_field` s tvoří jednotlivé časové řady, popřípadě grafy. Jedno pole je vždy jedna časová řada.

■ `_measurement`

Měření je další název, ale pro skupinu polí.

Dále je možné ještě využít `tag`, které umožňují rozšířit strukturu dat a lépe konkretizovat hodnoty ve složitějších tabulkách. Tato výše popsaná struktura přesně odpovídá `Line Protocol`, což je textově založený formát zpráv pro zápis do databáze. [43]

V databázi jsem tedy vytvořil tabulku s názvem `MKR_data`. Nyní jsem mohl v tabulce `MKR_data` tvořit jednotlivé měření pro každý let a v každém měření (tedy letu) jednotlivé pole, kde se budou uchovávat naměřené hodnoty například teploty, výšky, rychlosti a pozice. Je samozřejmě možné ukládat mnohem více proměnných.

Další výhodou InfluxDB je integrovaný `Dashboard`, tedy webová stránka zobrazující hodnoty v reálném čase. Je zde na výběr mnoho možností jak data zobrazit, jaký typ grafu zvolit či zda zobrazovat pouze poslední hodnotu.



Obrázek 4.9. Dashboard s ukázkou zobrazení dat v reálném čase, přijímány jsou data o teplotě, tlaku, osvětlení, vlhkosti a dále pak data s GPS shieldu o poloze, rychlosti, výšce a počtu satelitů.

Po přihlášení jménem a heslem do webového rozhraní je možné si prohlédnout uložená data v Data Explorer nebo využít Dashboard. Toto je výhodné, protože základnovou stanicí tedy může být jakékoliv zařízení s webovým prohlížečem. Může se jednat o mobilní telefon, notebook nebo například tablet či jakékoliv jiné zařízení s webovým prohlížečem. Uživatel tedy není vázán na jedno konkrétní zařízení a je možné si prohlédnout nebo stáhnout data odkudkoliv.

4.4.3 Telegraf

Existují tři hlavní způsoby jak posílat data do databáze InfluxDB [44]

■ REST API

Jednoduchá možnost jak rychle zasílat data pomocí REST API POST příkazu do databáze. Data musí být ve formátu Line protocol

■ Knihovna InfluxDB

Pro nejpopulárnější programovací jazyky byly vytvořeny knihovny, které umožňují jednoduše zasílat data do databáze. Existuje knihovna i pro Arduino IDE, bohužel však podporuje pouze ESP32 a ESP8266.

■ User agent Telegraf

Jedná se o program vytvořený v programovacím jazyce Rust a je poskytován volně ke stažení. Telegraf slouží jako prostředník a dokáže získávat data mnoha různými způsoby aby je předal databázi. Pro příjem UDP paketů jsem využil právě Telegraf.

Jako první jsem se pokusil zasílat data do databáze pomocí REST API. Databáze ale přijímá data pouze přes HTTPS protokol, takže pokaždé, když jsem se pokusil poslat data z modemu, obdržel jsem chybu 308 o permanentním přesměrování. Modem na desce Arduino MKR 1400 nepodporuje přesměrování a jeho funkce v této oblasti jsou limitovány.

Obrátil jsem se tedy na spolehlivější a univerzálnější způsob příjmu dat přes Telegraf prostředníka[45]. Po otevření UDP portu ve firewallu jsem provedl následující nastavení v konfiguračním souboru Telegrafu:

```
[[outputs.influxdb_v2]]
  urls = ["http://localhost:8086"]
  token = "<token získaný při tvorbě Bucketu>"
```



```

organization = "mkr_system"
bucket = "demo"

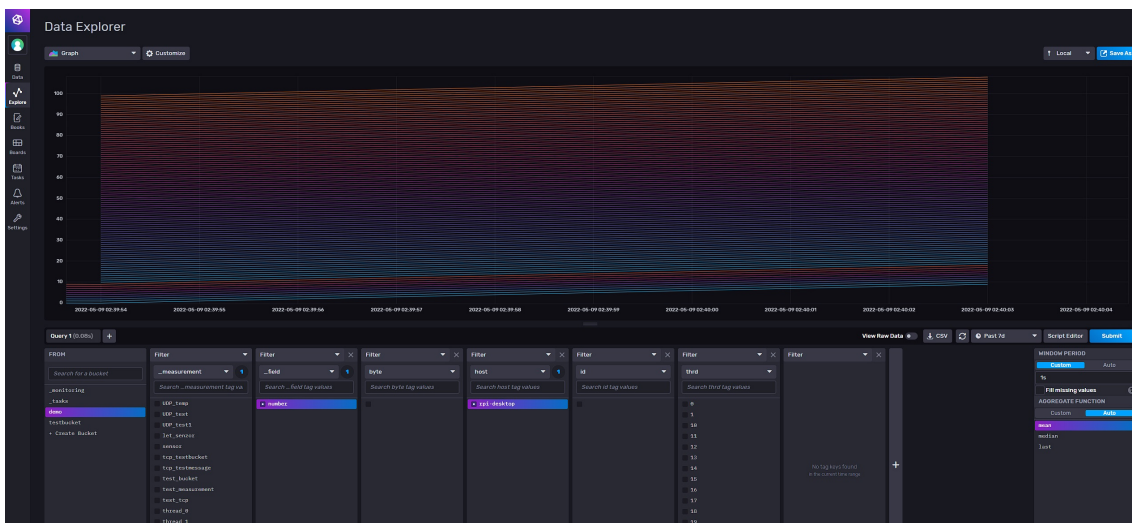
[[inputs.socket_listener]]
  service_address = "udp://:8094"
  data_format = "influx"
  content_encoding = "identity"

```

První odstavec definuje výstup do databáze, kam bude prostředník předávat data. Protože databáze běží na stejném zařízení, nastavil jsem adresu na `localhost`. Token je autorizační klíč generovaný automaticky při tvorbě tabulky a slouží k přístupu pro zápis a čtení dat. `Organization` slouží k rozlišení uživatelů databáze, název je volitelný při tvorbě uživatele. `Bucket` udává, do které tabulky se bude přistupovat.

Druhý odstavec definuje vstup pro Telegraf. `Service_address` definuje vstup dat, v mém případě UDP pakety na portu 8094. `Data_format` jsem nastavil na `influx`, tím Telegraf ví, že má očekávat data ve formátu Line Protocol popsany v sekci 4.4.2. `Content_identity` udává, zda-li jsou data nějak komprimovaná. V mém případě komprimovaná nejsou, ale je možnost přijímat data komprimovaná přes `gzip`.

Příjem dat jsem otestoval Python programem, který simuloval 100 klientů odesílající data přes UDP pakety současně každou vteřinu. Test proběhl úspěšně jak je vidět na snímku obrazovky 4.10. V `Data Explorer` databáze se správně zobrazilo 100 datových řad s postupně narůstajícími hodnotami ve správném pořadí.



Obrázek 4.10. Snímek obrazovky při testování UDP příjmu dat od 100 simulovaných klientů najednou

Kapitola 5

Závěr

V první části bakalářské práce jsem otestoval rychlost, spolehlivost a dosah dvou různých bezdrátových spojení pro přenos dat. Z výsledků experimentálního měření ve venkovním prostředí vyplývá, že radiové UART moduly pro přenos telemetrie na frekvenci 433MHz mají velice krátký dosah v řádu několika desítek metrů. Po více jak 150m je přenos již značně nespolehlivý. Tento typ bezdrátového spojení se ukázal jako nevhodný pro stavbu monitorovacího systému.

Dále jsem obdobným způsobem otestoval stejné parametry u vývojové desky Arduino MKR 1400 s integrovaným 3G/2G modemem na mobilní datové síti 2G. Výsledky měření ukázaly, že tento přenos je velice spolehlivý a má vyšší přenosovou kapacitu než radiové moduly. Dále má také mnohem větší dosah díky skvělému pokrytí území ČR mobilním 2G signálem.

V druhé části bakalářské práce jsem se věnoval stavbě monitorovacího systému založeném na bezdrátovém spojení zvoleném na základě výsledků z první části. Monitorovací systém používá topologii hvězda, ve které server přijímá data od jednoho klientského zařízení, ale je možné mít klientských zařízení více. Server byl postaven na minipočítači Raspberry Pi 4 s operačním systémem Ubuntu 22.04 LTS 64-bit a tento přístup jsem zvolil na základě několika zkušeností, které jsem při vývoji monitorovacího systému získal. Původně jsem stavěl server na Raspberry Pi 3B+, které má o něco nižší výkon, ale je cenově dostupnější. Výběr operačního systému je podmíněn faktem, že databáze vyžaduje 64-bitovou architekturu. Na RPi 3B+ jsem nainstaloval poslední verzi OS Raspbian 64-bit, který byl shodou okolností vydán několik týdnů před začátkem mé bakalářské práce. Dále bylo nutné, aby fungoval mobilní USB modem se statickou IP adresou pro příjem dat. Ten se mi částečně podařilo zprovoznit pomocí nástroje `modem manager`[47] do stavu, kdy se modem připojil k internetu. Dalším testováním se však ukázalo, že není možné obdržet odpověď od statické IP adresy přes příkaz `ping` ani po instalaci a konfiguraci firewallu na RPi 3B+. Přeinstaloval jsem tedy operační systém na Ubuntu Server 64-bit, kde se mi podařilo nainstalovat databázi InfluxDB. Bohužel jsem opět narazil na problém s řadiči k USB modemu, tentokrát nebyl úspěšný ani `modem manager`, protože se USB zařízení nepodařilo systému detekovat. Jako další možnost jsem tedy vyzkoušel Ubuntu 22.04 LTS 64-bit, kde se USB modem ihned zobrazil a bylo možné velice jednoduše zařízení nastavit, byť přes desktopové rozhraní. Instalace a konfigurace databáze InfluxDB proběhla bez problému.

Je nutné podotknout, že databáze InfluxDB nebyla mou první volbou. Jako velice slibná se jevila nová platforma Arduino IoT Cloud[48] určená pro rodinu vývojových desek MKR a běžící na cloud serveru společnosti Arduino. Základní členství je po registraci zdarma a umožňuje do IoT Cloud připojit dvě klientská zařízení. IoT Cloud nabízí uložení dat a také Dashboard pro zobrazení dat v reálném čase. Arduino MKR je možné programovat přes webové rozhraní stejným způsobem jako v Arduino IDE. Narazil jsem však na problém, že odesílání více jak jedné proměnné v jeden okamžik způsobí restart vývojové desky Arduino.

Zabezpečením monitorovacího systému jsem se zabýval okrajově nad rámec zadání. V sekci 4.4.3 jsem uvedl svou snahu přistupovat k databázi přímo přes HTTPS protokol, bohužel ani po dvoutýdenním řešení problému a práci s modemem přímo za pomoci AT příkazů, se mi nepodařilo k databázi připojit. A to i přesto, že na jiné HTTPS stránky, u kterých jsem do modemu naimportoval jiný certifikát bylo možné přistoupit.

Další možné oblasti, které by mohly být rozvinuty jako součást tohoto projektu, by mohly být zaměřeny jak na software, tak i na hardware monitorovacího systému. Momentálně je webové rozhraní databáze přístupné na jméno a heslo, ale používá protokol HTTP. Databáze nabízí možnost šifrovaného spojení, pro mé testovací účely jsem ho nepotřeboval a zabýval jsem se hlavně přenosem dat. Dále bych rád použil přenos dat pomocí TCP paketů. Během mého testování přenosu dat se mi nestalo, že by UDP paket nedorazil nebo by se změnilo po cestě jejich pořadí. TCP pakety by však přispěly k větší robustnosti přenosového kanálu. Použití TCP paketů by vyžadovalo použití AT příkazů přímo s modemem. Z hardwarového hlediska bych UAV jednotku do budoucna rozvinul o sofistikovanější a zmenšenou základnu. Server bych ponechal na platformě Raspberry Pi, ale vyměnil bych ho za desku verze 3B+ s operačním systémem Raspbian se statickou IP adresou udělenou pro Ethernetové rozhraní.

Výsledkem mé práce na monitorovacím systému je ověřený funkční koncept, skládající se ze serverové části pro příjem a zobrazení dat, dále pak jednotka pro sběr avionických dat za pomoci integrovaných senzorů či pomocí sběrnice CAN nebo sériové linky RS232 z externích senzorů. Serverová část je přístupná na velkém množství zařízení díky webovému prohlížeči, je schopna přijímat data mnoha způsoby, zobrazit je v reálném čase a exportovat je pro další zpracování. Jednotka pro sběr dat na UAV disponuje sběrnici SPI, CAN, I2C, sériovým rozhraním s napětovými úrovněmi 3.3 V a sériovou linkou RS232. Veškeré kódy jsou přístupné na mém GitLabovém účtu¹. Celá koncepce monitorovacího systému je vysoce modulární a má rozsáhlé možnosti sběru avionických dat za pomoci klientského zařízení Arduino MKR s možností připojení mnoha dalších klientských zařízení různého typu přes internetovou síť.

¹ <https://gitlab.fel.cvut.cz/simecric/uav-monitorovaci-system>

Literatura

- [1] SUN, S. ACARS Data Identification and Application in Aircraft Maintenance. In: *2009 First International Workshop on Database Technology and Applications* [online]. IEEE, 2009, s. 255-258. [cit. 2021-12-09]. ISBN:978-0-7695-3604-0. DOI: 10.1109/DBTA.2009.93. Dostupné z: <https://ieeexplore.ieee.org/document/5207766>
- [2] M. EDRICH, R. SCHMALENBERGER. A Combined DSSS/FHSS approach to interference rejection and navigation support in UAV communications and control. In: *IEEE Seventh International Symposium on Spread Spectrum Techniques and Applications* [online]. Praha, 2002, s. 687-691. [cit. 2022-05-14]. ISBN:0-7803-7627-7. DOI: 10.1109/ISSSTA.2002.1049347. Dostupné z: <https://ieeexplore.ieee.org/document/1049347>
- [3] Všeobecné oprávnění č. VO-R/10/12.2019-9 ze dne 17. prosince 2019 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu. In: *všeobecné oprávnění č. VO-R/10/12.2019-9 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu*. 2019, s. 3. ISSN ČTÚ-43 165/2019-613. Dostupné také z: <https://www.ctu.cz/sites/default/files/obsah/ctu/vseobecne-opravneni-c.vo-r/10/12.2019-9/obrazky/vo-r10-122019-9.pdf>
- [4] *AltiGator Unmanned Solutions* [online]. AltiGator, 2021. [cit. 2021-12-09]. Dostupné z: <https://drones.altigator.com/radio-telemetry-kit-a-for-pixhawk-433mhz-p-42334.html>
- [5] *frequency-hopping spread spectrum* [online]. TechTarget, 2000-2022. [cit. 2022-05-14]. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/frequency-hopping-spread-spectrum>
- [6] *Configuring a Telemetry Radio using Mission Planner* [online]. ArduPilot Dev Team, 2021. [cit. 2022-05-14]. Dostupné z: <https://ardupilot.org/copter/docs/common-configuring-a-telemetry-radio-using-mission-planner.html>
- [7] *MKR GSM 1400* [online]. Arduino, 2022. [cit. 2022-05-14]. Dostupné z: <https://docs.arduino.cc/hardware/mkr-gsm-1400>
- [8] *SARA-U201 module* [online] u-blox, 2022. [cit. 2022-05-14]. Dostupné z: https://content.u-blox.com/sites/default/files/SARA-U2_ProductSummary_%28UBX-13004142%29.pdf
- [9] *SARA-U201 module datasheet* [online]. u-blox, 2022. [cit. 2022-05-14]. Dostupné z: https://content.u-blox.com/sites/default/files/SARA-U2_DataSheet_UBX-13005287.pdf#page=7&zoom=100,68,741
- [10] *Mapa pokrytí mobilním internetovým signálem* [online]. Vodafone, 2022. [cit. 2022-05-14]. Dostupné z: <https://www.vodafone.cz/mapa-pokryti/>

- [11] *Pokrytí pro mobilní služby* [online]. Vodafone, 2022. [cit. 2022-05-14].
Dostupné z: <https://www.vodafone.cz/pece/internet-data/pokryti/sit-vodafone/>
- [12] *Knihovna pro ovládání GSM modemu na desce Arduino MKR* [online]. Arduino, 2022. [cit. 2022-05-14].
Dostupné z: <https://www.arduino.cc/reference/en/libraries/mkrgsm/>
- [13] *Tenká anténa GSM-ANT-FPC02 od společnosti SR Passives* [online]. TME Electronics Components, 2022. [cit. 2022-05-14].
Dostupné z: <https://www.tme.eu/Document/729ebed8a4988fc959c5e50ae6faf086/GSM-ANT-FPC02.pdf>
- [14] *PCB anténa DB-A0125-GSM&4G* [online]. Dinobell, 2022. [cit. 2022-05-14].
Dostupné z: <https://www.dinobell.com/5dbi-gsm-2g-3g-4g-lte-pcb-antenna-db-a0125-gsm4g.html>
- [15] HRON, Lukáš. Konec jedné mobilní éry. 3G je v Česku až na výjimky minulostí. In: *idnes* [online]. 30. 11. 2021. [cit. 2022-05-14]. Dostupné z: https://www.idnes.cz/mobil/mobilni-operatori/o2-t-mobile-3g-mobilni-sit-vypnuti.A211130_162918_mobilni-operatori_LHR
- [16] *Knihovna MKRENV* [online]. Arduino, 2022. [cit. 2022-05-14].
Dostupné z: https://www.arduino.cc/reference/en/libraries/arduino_mkrenv/
- [17] *MKR ENV shield* [online]. Arduino, 2022. [cit. 2022-05-14]. Dostupné z: <https://docs.arduino.cc/hardware/mkr-env-shield>
- [18] *Datasheet senzoru LPS22HB* [online]. STMicroelectronics, 2017. [cit. 2022-05-14].
Dostupné z: <https://www.st.com/resource/en/datasheet/dm00140895.pdf>
- [19] *Datasheet senzoru LPS22HB* [online]. STMicroelectronics, 2016. [cit. 2022-05-14].
Dostupné z: <https://www.st.com/resource/en/datasheet/dm00140895.pdf>
- [20] *Datasheet senzoru TEMT6000X01* [online]. VISHAY INTERTECHNOLOGY, INC, 2021. [cit. 2022-05-14].
Dostupné z: <https://www.vishay.com/docs/81579/temt6000.pdf>
- [21] *Altitude Calculations* [online]. NAROM, 2018. [cit. 2022-05-14].
Dostupné z: <https://www.narom.no/undervisningsressurser/the-cansat-book/the-primary-mission/using-the-sensors/altitude-calculations/>
- [22] *MKR CAN shield* [online]. Arduino, 2022. [cit. 2022-05-14]. Dostupné z: <https://docs.arduino.cc/hardware/mkr-can-shield>
- [23] *Specifikace CAN sběrnice* [online]. ROBERT BOSCH GmbH, 1991. [cit. 2022-05-14].
Dostupné z: <http://esd.cs.ucr.edu/webres/can20.pdf>
- [24] *Datasheet CAN kontroleru* [online]. Microchip Technology Inc., 2003-2019. [cit. 2022-05-14].
Dostupné z: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>
- [25] *Datasheet CAN vysílače a přijímače* [online]. NXP Semiconductors N.V., 2018. [cit. 2022-05-14].
Dostupné z: <https://www.nxp.com/docs/en/data-sheet/TJA1049.pdf>
- [26] *Datasheet spínaného zdroje na CAN shieldu* [online]. Texas Instruments Incorporated, 2022. [cit. 2022-05-14].
Dostupné z: <https://www.ti.com/lit/ds/symlink/tps54232.pdf>

- [27] *MKR GPS Shield* [online]. Arduino, 2022. [cit. 2022-05-14].
Dostupné z: <https://docs.arduino.cc/hardware/mkr-gps-shield>
- [28] *SAM-M8Q GNSS modul společnosti u-blox* [online]. u-blox, 2022. [cit. 2022-05-14].
Dostupné z: <https://www.u-blox.com/en/product/sam-m8q-module>
- [29] *Datasheet SAM-M8Q GNSS modulu společnosti u-blox* [online]. u-blox, 2022. [cit. 2022-05-14].
Dostupné z: https://content.u-blox.com/sites/default/files/SAM-M8Q_DataSheet_%28UBX-16012619%29.pdf
- [30] *Schéma vývojové desky Arduino MKR 1400* [online]. u-blox, 2022. [cit. 2022-05-14].
Dostupné z: <https://docs.arduino.cc/static/eddbb43ab9e19e77d1ff2bb7e19e5506/schematics.pdf>
- [31] *Schéma vývojové desky Arduino MKR 1400* [online]. u-blox, 2022. [cit. 2022-05-14].
Dostupné z: <https://docs.arduino.cc/static/eddbb43ab9e19e77d1ff2bb7e19e5506/schematics.pdf>
- [32] *Převodník UART na RS232* [online]. Texas Instruments Incorporated, 1995-2022. [cit. 2022-05-14].
Dostupné z: https://www.ti.com/store/ti/en/p/product/?p=MAX3232IDW&utm_source=google&utm_medium=cpc&utm_campaign=ocb-tistore-promo-asc_opn_en-cpc-storeic-google-wwe&utm_content=Device&ds_k=MAX3232IDW&DCM=yes&gclid=Cj0KCQjwpv2TBhDoARIsALBnVnnSlhvpvS5oCdzzpirZZ5agsPPW4YzsiN8ockGJydnWhGgqS4UXoa8UaAliwEALw_wcB&gclsrc=aw.ds
- [33] *ArduPilot* [online]. ArduPilot.org, 2022. [cit. 2022-05-14].
Dostupné z: <https://ardupilot.org/>
- [34] HERCEG, Tomáš. Úvod do jazyka sql. In: *dotnetportal* [online]. 3. 9. 2007. [cit. 2022-05-14]. Dostupné z: <https://www.dotnetportal.cz/clanek/50/Uvod-do-jazyka-SQL>
- [35] *Time series database (TSDB) explained* [online]. InfluxData Inc., 2022. [cit. 2022-05-14]. Dostupné z: <https://www.influxdata.com/time-series-database/>
- [36] *Mikropočítač Raspberry Pi 4* [online]. Raspberry Pi, 2022. [cit. 2022-05-14].
Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- [37] *Ubuntu 22.04 LTS 64-bit* [online]. Canonical Ltd. Ubuntu and Canonical are registered trademarks of Canonical Ltd., 2022. [cit. 2022-05-14].
Dostupné z: <https://ubuntu.com/download/raspberry-pi>
- [38] *GNU Screen* [online]. Free Software Foundation, Inc., 2010. [cit. 2022-05-14].
Dostupné z: <https://www.gnu.org/software/screen/>
- [39] *Balík UFW* [online]. Canonical Ltd., 2004-2022. [cit. 2022-05-14].
Dostupné z: <https://launchpad.net/u fw>
- [40] *Databáze InfluxDB* [online]. InfluxData Inc., 2022. [cit. 2022-05-14].
Dostupné z: <https://www.influxdata.com/>
- [41] *Time structured merge tree* [online]. InfluxData Inc., 2004-2022. [cit. 2022-05-14].
Dostupné z: <https://www.influxdata.com/blog/new-storage-engine-time-structured-merge-tree/>
- [42] *InfluxDB data schema* [online]. InfluxData, Inc., 2022. [cit. 2022-05-14].
Dostupné z: <https://docs.influxdata.com/influxdb/v2.2/reference/key-concepts/data-schema/#annotation-rows>

-
- [43] *Line protocol* [online]. InfluxData, Inc., 2022. [cit. 2022-05-14].
Dostupné z: <https://docs.influxdata.com/influxdb/v2.2/reference/syntax/line-protocol/>
- [44] *Write data to InfluxDB* [online]. InfluxData, Inc., 2022. [cit. 2022-05-14].
Dostupné z: <https://docs.influxdata.com/influxdb/v2.2/write-data/>
- [45] *Telegraf prostředník* [online]. InfluxData, Inc., 2022. [cit. 2022-05-14].
Dostupné z: <https://github.com/influxdata/telegraf>
- [46] *Modem manager pro Ubuntu* [online]. Canonical Ltd. Ubuntu and Canonical are registered trademarks of Canonical Ltd., 2022. [cit. 2022-05-14].
Dostupné z: <https://snapcraft.io/install/modem-manager/ubuntu>
- [47] *Modem manager pro Ubuntu* [online]. Canonical Ltd. Ubuntu and Canonical are registered trademarks of Canonical Ltd., 2022. [cit. 2022-05-14].
Dostupné z: <https://snapcraft.io/install/modem-manager/ubuntu>
- [48] *Arduino IoT Cloud* [online]. Arduino, 2022. [cit. 2022-05-14].
Dostupné z: <https://docs.arduino.cc/cloud/iot-cloud>

Příloha A

Zkratky

Seznam zkratek

SDR	Software Defined Radio - Softwareově definované radio
UART	Universal asynchronous receiver-transmitter - univerzální asynchronní přijímač-vysílač
LTE	Long Term Evolution - vysokorychlostní mobilní internet
UDP	User Datagram Protocol - protokol transportní vrstvy pro přenos dat
UAV	Unmanned Aerial Vehicle - nepilotovaný letecký prostředek
GSM	Groupe Spécial Mobile - globální systém pro mobilní komunikace
ACARS	Aircraft Communications Addressing and Reporting System - letecký systém pro přenos krátkých zpráv
AM	Amplitudová Modulace
ISM	Industrial, Scientific and Medical - rádiové pásmo uvolněné pro vysílání bez licence, pouze s omezeným výkonem a šířkou pásma
SMA	SubMiniature version A - koaxiální konektor pro střídavé signály
ČTÚ	Český telekomunikační úřad
FHSS	Frequency hopping spread spectrum - metoda přenosu dat v rozšířeném spektru
USB	Universal Serial Bus - univerzální sériová sběrnice
COM	Communication Port - komunikační rozhraní
JST	Japan Solderless Terminal - Název konektoru přejat od názvu společnosti vyrábějící tyto konektory
RX	Zkratka pro příjem signálu
TX	Zkratka pro vysílání signálu
DVBT	Digital Video Broadcasting Terrestrial - standard digitálního pozemního vysílání televizního signálu
ČR	Česká republika
IoT	Internet of Things - Internet věcí
LTE NB	Long Term Evolution Narrow Band - úzkopásmový vysokorychlostní mobilní internet pro IoT aplikace
3G	Datová mobilní síť třetí generace
2G	Datová mobilní síť druhé generace
E-GSM	Extended Global System for Mobile Communications - rozšířený globální systém pro mobilní komunikace
DCS	Označení pro datové pásmo 1800MHz
PCS	Personal Communications Services - osobní komunikační služba
SoC	System on Chip - Systém na čipu, například desky minipočítačů typu Raspberry Pi
I2C	Inter-Integrated Circuit - sběrnice pro přenos dat za pomoci dvou vodičů mezi několika zařízeními
ISO	International Organization for Standardization - mezinárodní organizace pro standardizaci

OSI	Open Systems Interconnection model - vrstvený model pro řešení komunikace mezi zařízeními
ICMP	Internet Control Message Protocol - protokol pro sdílení služebních informací v internetové síti
IP	Internet Protocol - internetový protokol
GNSS	Global Navigation Satellite System - globální družicový polohový systém
RS232	Označení fyzického řešení sériové linky
CSV	Comma-separated values - soubor pro uložení dat kde jsou data oddělena čárkou
RTR	Vzdálené vyžádání zprávy
ID	Identifikátor
SAW	Surface Acoustic Wave - povrchová akustická vlna
RX	příjem dat
TX	vyšlání dat
TTL	Tranzistorově-tranzistorová logika s napětovými logickými úrovněmi mezi 0 V až 5 V
RTS	Request to Send - Signál zde značí, že zařízení chce vysílat data
CTS	Clear To Send - Signál zde značí, že sběrnice je volná pro přenos dat
CTS	Dual in-line package - značí standardizované pouzdro komponenty se dvěma řádky fyzických propojek
RC	Radio Controlled - Dálkově ovládaný přes rádiové vlny
SQL	Structure Query Language - strukturovaný dotazovací jazyk
LTS	Long Term Support - dlouhodobě podporovaný
REST	Representational state transfer - architektura webového rozhraní
API	Application Programming Interface - rozhraní mezi aplikacemi
HTTP	Hypertext transfer protocol - internetový protokol pro komunikaci webových serverů a klientů
HTTPS	Hypertext transfer protocol secured - internetový protokol pro komunikaci webových serverů a klientů se šifrováním
TCP	Transmission Control Protocol - protokol transportní vrstvy pro spolehlivý přenos dat
AT	zvaný také Hayes command set - komunikační protokol vyvinutý pro ovládání modemů