

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta



**Alternativní metoda dálkových odečtů elektroměrů
v příhraniční oblasti pomocí UAV**

Bakalářská práce

Tomáš Macák

Školitel: Ing. Rudolf Vohnout, Ph.D.

České Budějovice 2021

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Student: Tomáš Macák
(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika

Katedra/ústav PŘF JU, kde bude práce vypracována a obhájena: UAI

Školitel: Ing. Rudolf Vohnout, Ph.D.
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PŘF JU:
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant:
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma bakalářské práce: Alternativní metoda dálkových odečtů elektroměrů v příhraniční oblasti pomocí UAV

Dálkové odečty elektroměrů v málo osídlených oblastech Šumavy představují problematiku, která může mít negativní vliv na spolehlivost a optimální funkci Smart Gridu v dané oblasti. Je proto třeba zkoumat také alternativní možnosti sběru dat o skutečných spotřebách v těchto a podobných lokalitách.

Úkoly práce:

- Vytvořit literární rešerši na téma metody dálkových odečtů elektroměrů, používaných protokolů v technologii Smart Metering, protokolů na komunikaci na krátkou a středně krátkou vzdálenost a závislosti rychlosti sběru informací o spotřebách na optimalizaci distribuce.
 1. Úvod do problematiky dálkových odečtů a jimi používaných komunikačních protokolů, bezdrátových komunikačních protokolů využívajících wake-up a sleep princip.
 2. Výběr s využitím správně nastavených kritérií vhodného UAV, pro uskutečnění experimentu se zohledněním délky doletu a doby provozu na jedno nabití.
 3. Příprava podkladů pro návrhovou část s přehledem HW a SW komponent výsledného řešení se zdůvodněním finálního výběru.

Cíle práce:

- Návrh systému pro UAV, který bude zodpovědný za vyčítání dat z chytrých elektroměrů.

- Experimentální verifikaci navrženého systému v reálném prostředí (tj. nikoliv simulačně).

Základní doporučená literatura:

[1] S. S. Bhalshankar and C. S. Thorat, "MAHADISCOM electrical theft controlling mechanism: (Smart grid advanced metering infrastructure and drone operated technology for controlling theft by direct hooking)," *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, 2017, pp. 1647-1650. doi: 10.1109/WiSPNET.2017.8300040

Financování práce

Školitel prácepodpis:

U externích vedoucích fakultní garant prácepodpis:

Garant oboru bak. studia (nepožaduje se u oboru biologie)podpis:

Vedoucí katedry/ústavu PŘF JU, kde proběhneobhajoba.....podpis:

Případný souhlas vedoucího ústavu AVpodpis:

V Českých Budějovicích dne 2.7.2019

Podpis studenta 

Bibliografické údaje

Macák, T., 2021: Alternativní metoda dálkových odečtů elektroměrů v příhraniční oblasti pomocí UAV. [Alternative method of remote meter reading in the border area using UAV. Bc. Thesis, in Czech.] – 50 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic

Anotace

Cílem této bakalářské práce je navrhnout alternativní metodu pro dálkové odečty dat z chytrých elektroměrů, a to s využitím UAV. Po navržení systému bude tento systém vytvořen a následně otestován. Tento systém by měl sloužit ke sběru dat z oblastí, kde nasazení běžných technologií pro dálkové odečty není možné. V teoretické části bude pak seznámení s běžně používanými metodami dálkových odečtů a dále budou popsány bezdrátové technologie, které budou k tomuto projektu zvažovány.

Annotation

The aim of this bachelor thesis is to design an alternative method for remote meter reading of smart meters with use of UAV. After designing the system will be created and tested afterwards. This system should be useful for gathering of data from locations where the use of classical systems for remote meter reading is not possible. In teoretic part will be a brief introduction to common used methods of remote meter reading and description of wireless technologies which will be considered for this project.

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

V Českých Budějovicích dne: 13. 4. 2021

Podpis autora: Tomáš Macák

Poděkování

Rád bych zde poděkoval svému vedoucímu bakalářské práce panu Ing. Rudolfu Vohnoutovi, Ph.D., za čas věnovaný mně a této práci a poskytnuté odborné konzultace. Dále bych rád poděkoval panu Ing. Václavu Novákovi, CSc. a panu PhDr. Milanu Novákovi, Ph.D. za věnované odborné konzultace k této práci.

Obsah

1	Úvod.....	1
2	Cíle, a struktura práce.....	2
3	Metodologický postup.....	4
4	Teoretická část	6
4.1	Závislosti sběru informací o spotřebách na optimalizaci distribuce.....	6
4.2	Metody dálkových odečtů elektroměrů a používané protokoly	6
4.2.1	Technologie PLC	7
4.2.2	Bezdrátové (Rf) technologie	8
4.3	Použité Rf technologie	10
4.3.1	Použité antény	11
4.4	Použitý hardware	11
4.4.1	Arduino UNO	11
4.4.2	Raspberry Pi 3B.....	12
4.5	Použité programové vybavení	12
4.5.1	Operační systém	12
4.5.2	Další použitý software	13
4.5.3	Programovací jazyk	13
4.5.4	Použité knihovny v Pythonu	13
4.5.5	Použité knihovny pro Arduino	14
4.6	Faktory ovlivňující volbu zmíněných HW a SW prvků	14
5	Praktická část	15
5.1	Popis a upřesnění požadavků pro volbu vhodného UAV	15
5.1.1	Typy UAV ke zvažení	15
5.1.2	Další nutné vlastnosti UAV.....	16
5.2	Výběr vhodného hardwaru	18

5.2.1	Volba hardwaru pro UAV	18
5.2.2	Volba hardwaru k chytrým elektroměrům	19
5.3	Výběr vhodných modulů pro bezdrátovou komunikaci.....	20
5.4	Konfigurace RF modulů.....	21
5.4.1	Konfigurace modulů pro Raspberry Pi	22
5.4.2	Konfigurace modulu pro Arduino	24
5.4.3	Test konfigurace	25
5.5	Tvorba skriptů	26
5.5.1	Tvorba skriptu pro Arduino.....	26
5.5.2	Tvorba skriptů pro RPi.....	32
5.5.3	Dodatečné skripty	37
5.6	Testování navrženého systému.....	38
6	Diskuse a inspirace pro navazující práce.....	40
6.1	Návrhy na zlepšení.....	40
7	Závěr.....	42
8	Seznam použité literatury	44
9	Seznam obrázků	48
10	Seznam tabulek	49
11	Seznam příloh	50

1 Úvod

Chytré elektroměry jsou dnes stále více využívány a s tím rostou i možnosti jejich odečtů. Většina těchto elektroměrů je však umístěna v dobře dostupných oblastech, a tak zde není se sběrem dat o spotřebách větší problém. Často je tedy přenos informací prováděn pomocí různých druhů kabelů či pomocí jiných bezdrátových sítí. Problém však nastává, pokud chceme tyto typy elektroměrů použít v místech, kde nejsou standardní způsoby odečtů možné. Jsou to například málo osídlené či špatně přístupné oblasti. V těchto lokalitách totiž není dostupná ať už drátová, či bezdrátová síť. Tu zde zavést buď nelze, anebo by zde neměla žádné jiné využití. Odečty zde pak představují problematiku, která může mít negativní vliv na spolehlivost a optimální funkci Smart Gridu v dané oblasti. Je proto třeba zkoumat také alternativní možnosti sběru dat o skutečných spotřebách v těchto lokalitách.

Jednou z možností získávání dat z těchto míst je sběr pomocí UAV. Vhodné UAV s optimální nosností a dostatečnou vydrží baterie se nastaví tak, aby létalo po nejkratší trase chytrých elektroměrů v předem dané výšce v dané oblasti. Na zvolený dron se pak zavěsí zařízení, které konkrétní elektroměr na trase rozpozná a provede bezpečný sběr dat. UAV pak prolétne trasu v oblasti a sebere data od všech elektroměrů. Data budou moci být po navrácení dronu zobrazena a dále analyzována. Právě vývojem zmíněného systému pro UAV se tato práce zabývá. Systém bude následně testován v reálném prostředí.

Toto téma bylo zvoleno z důvodu potřeby odečtů chytrých elektroměrů v neosídlených a špatně přístupných oblastech Šumavy. Také z důvodu potřeby navržení podobného systému na katedře informatiky (UAI) Přírodovědecké fakulty na Jihočeské univerzitě. Tato práce navazuje na projekt, kde jsou data přímo vyčítána. Nezabývá se tedy konkrétním vyčítáním dat z elektroměrů.

2 Cíle, a struktura práce

Cíle této práce by se daly dělit následujícím způsobem:

Cíl 1) Volba vhodného UAV

Před výběrem samotného zařízení a následným vývojem systému je nutné provést volbu či specifikaci UAV, které je vhodné pro plnění funkce navrženého systému. Zde je třeba porovnat jednotlivé typy a vyhodnotit, který bude pro využití nejvíce vhodný. Také je nutné popsat vlastnosti, kterými musí vybíraný dron disponovat.

Cíl 2) Vývoj systému pro dálkové odečty chytrých elektroměrů

Hlavním cílem celé práce je vytvořit takový systém, který je možné připevnit na dříve vybrané UAV, a který je schopen rozpoznat dané elektroměry na trase a bezpečně od nich sbírat data. K splnění tohoto cíle je nutné zvolit vhodný hardware, který bude využit. Je také třeba vytvořit programové vybavení, které bude zprostředkovávat komunikaci mezi UAV a elektroměry.

Cíl 3) Testování funkčnosti vytvořeného systému

Posledním, ne méně důležitým cílem praktické části je vytvořený systém otestovat a doladit všechny nedostatky. Ověří se tak, zda je navržené zařízení opravdu možné využít k výše zmíněnému typu funkcionality. Typů testů bude provedeno více.

Práce je rozdělena na dvě hlavní kapitoly, a to kapitoly teoretická část a část praktická. Tyto kapitoly jsou dále členěny do dalších podkapitol, které slouží k ulehčení orientace v dokumentu:

1) Teoretická část

V této části je vytvořena rešerše, která je zaměřena na téma závislosti sběru dat od chytrých elektroměrů na optimalizaci distribuce a běžně používané metody dálkových odečtů elektroměrů. Je zde také popis bezdrátové technologie, hardwaru a programového vybavení použitého k vývoji systému, který tvoří hlavní cíl této práce.

2) Popis praktické části

Tato kapitola se zaměřuje na samotný vývoj systému pro dálkové odečty elektroměrů pomocí UAV. Na počátku jsou rozebrány faktory, ke kterým je nutné přihlížet při volbě dronu pro tvořený systém. Jsou zde také popsány požadavky na výběr hardwaru a rádiového modulu, dále konfigurace rádiových modulů a zejména vývoj skriptů, které zprostředkovávají samotnou komunikaci mezi elektroměrem a dronem. Závěr kapitoly tvoří popis testování navrženého systému.

3 Metodologický postup

1) Studium odborných materiálů

Nejprve je prostudována odborná literatura týkající se řešeného problému a popsány metody, které jsou běžně k odečtům používány. Jsou použity jak materiály knižní, tak materiály elektronické. Všechny využití zdroje jsou uvedeny do seznamu použité literatury na konci práce.

2) Sběr dalších informací

Dále je nutné získat informace o chytrých elektroměrech používaných v oblasti, a především o datech, která jsou z nich vyčítána, pomocí jakého hardwaru a v jakém formátu. Tyto informace jsou získány od kolegy, který vypracovává práci, ve které vyčítá konkrétní data.

3) Výběr vhodného UAV a zařízení

Následně je proveden výběr či specifikace UAV, které má dlouhý dolet na jedno nabití a je zde možnost zavěsit externí zařízení a následně je zvolen hardware, který je s již vybraným dronem a elektroměry použitelný. Je také volen rádiový modul, který zprostředkovává samotnou bezdrátovou komunikaci.

4) Volba programovacího jazyka a programového vybavení

V tomto kroku je třeba vybrat programovací jazyky, které jsou dále použity pro programování skriptů na již vybraná zařízení. Jazyky jsou zvažovány s přihlédnutím k vybranému hardwaru, počtu vhodných knihoven a dostupným recenzím. Je také zvoleno programové vybavení, které je při vývoji využito.

5) Vývoj

Po zajištění potřebných informací, výběru vhodného UAV a samotného zařízení je čas na programování. K tomu je využito již dříve popsaných programovacích jazyků a dalšího programového vybavení.

6) Testování

Po zhotovení finální verze je nutné navržený systém řádně otestovat. Testování probíhá ve více fázích, a to test konkrétního zvoleného rádiového modulu na počátku a test vytvořeného systému po vypracování. Testování samozřejmě probíhá také během vývoje.

4 Teoretická část

4.1 Závislosti sběru informací o spotřebách na optimalizaci distribuce

Přesné informace o spotřebě elektrické energie jsou velice důležité. Díky těmto informacím může poskytovatel elektrické energie sledovat skutečnou spotřebu na jednotlivých odběrných místech.

Po zavedení systému AMR (Automatic Meter Reading) můžeme data sbírat dálkově, častěji (třeba i každou hodinu) a pravidelně a nemusíme k provedení této činnosti vysílat fyzickou osobu, místo které k tomu využíváme různých komunikačních technologií. Je možné také sbírat data jiná než pouze celkovou spotřebu elektřiny za určité období. Díky tomuto systému má poskytovatel přesnější informace o spotřebách energie, závislé na krátkém časovém úseku a může tak lépe plánovat na trhu s elektrickou energií. Tyto informace jsou samozřejmě užitečné i pro spotřebitele, kteří energii odebírají. [1]

Dalším důležitým poznatkem je, že díky přesným informacím o spotřebě energie na daných odběrných místech v konkrétním čase máme možnost lépe odhalovat podvody a úniky dodávané elektrické energie. [1]

Inteligentní elektroměry umožňují již obousměrnou a téměř real-time komunikaci. Na základě toho je možné umožnit regulaci výroby a optimalizovat služby a dodávky zákazníkovi. [2] Tento fakt je velice užitečný, co se stability gridu týče. Díky znalostem o aktuálním využití je možné zvyšovat či redukovat výrobu elektrické energie dle aktuální potřeby. Zároveň je také možné udržovat dostatečné rezervy pro případy výpadků ve výrobě, nebo jiných problémů. S využitím inteligentních prvků je však nutné brát v potaz také bezpečnost. [3, 4] Díky obousměrné komunikaci lze například zákazníka vzdáleně odstavit od dodávek energie.

4.2 Metody dálkových odečtů elektroměrů a používané protokoly

Pro efektivní činnost elektroměrů je nutné zvolit vhodnou metodu dálkových odečtů. Smart metery dokonce umožňují komunikaci v obou směrech. V této kapitole jsou popsány technologie, které lze pro tento typ přenosu dat využít. Komunikace by měla probíhat, pokud možno, s co nejmenším zpožděním.

Při volbě přenosového systému přihlížíme zejména k dostupnosti jednotlivých technologií vzhledem k oblasti, kde jsou jednotlivá měřidla instalována. Ke komunikaci s nadřazenými prvky dnes můžeme využít buďto kabel, optická vlákna, či různé bezdrátové komunikační protokoly. [5]

4.2.1 Technologie PLC

Power Line Communication (PLC) je technologie pro přenos datových signálů přes již existující vedení elektrické sítě. Výhodou je, že není potřeba instalovat a spravovat další komunikační trasy, což zároveň snižuje finanční náklady. Moderní PLC umí dosahovat vysokých přenosových rychlostí. [6]

Elektrická distribuční síť využívá pro přenos frekvenci 50 Hz. Je tedy možné využití pásma vyšších frekvencí k přenosu datových signálů. Vzhledem k primárnímu účelu použití těchto sítí mohou nastat problémy, které vzniknou rušením elektrických spotřebičů připojených do stejné rozvodné sítě. Princip PLC komunikace spočívá v modulaci a demodulaci signálu. Do elektrické rozvodné sítě se injektuje modulovaný, vysokofrekvenční signál, který se dále šíří sítí. Přijímací zařízení získá poté data tak, že oddělí tyto signály v komunikačním pásmu a signál demoduluje. [7]

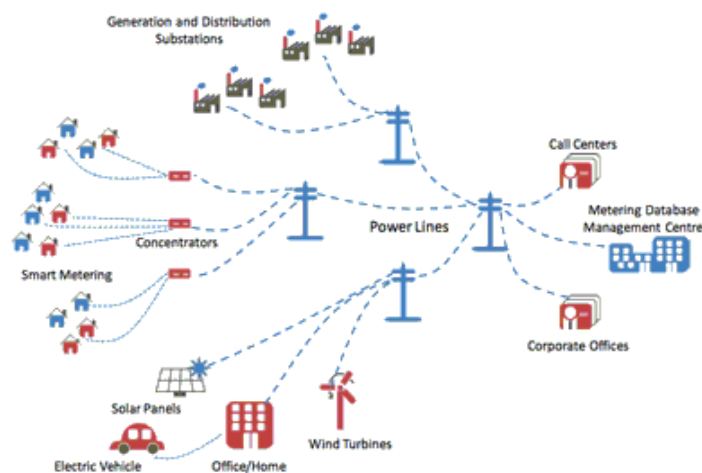
Dělíme na 2 druhy:

- **Narrowband PLC**

Tento typ pracuje na nižších frekvencích, 3 – 500 kHz, a nedosahuje takové přenosové rychlosti. Ovšem dosahuje větších vzdáleností až několik kilometrů. [8] Technologie se dále dělí na Low Data Rate (LDR) a High Data Rate (HDR). Rychlosti HDR dosahují až 500 kbps. [7]

- **Broadband PLC**

Zde jsou k přenosu využívány vyšší frekvenční pásma, 1,8 – 250 MHz, což umožňuje dosahovat vyšších přenosových rychlostí (stovky Mbps). Tento typ je využíván zejména v případech, kdy nám postačí komunikace na kratší dosah. [7, 8]



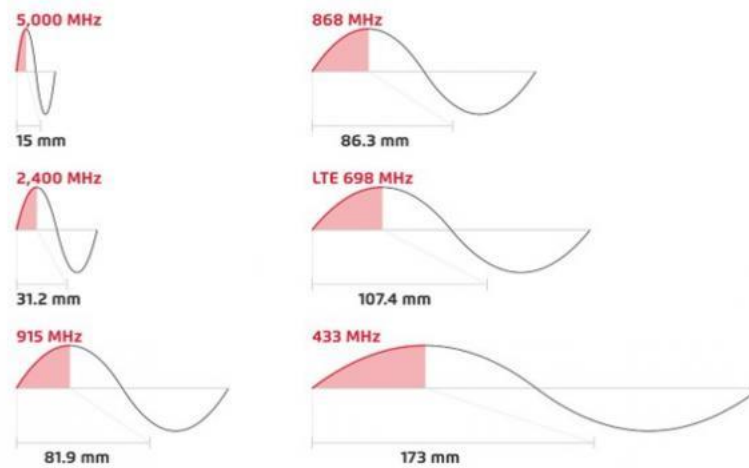
Obrázek 1 - Komunikace s využitím PLC [8]

4.2.2 Bezdrátové (Rf) technologie

Radiofrekvenční přenos je typem bezdrátové komunikace. [5] Tyto technologie využívají elektromagnetického vlnění, které se šíří bezdrátově. Dnes tyto technologie dosahují vysokých přenosových rychlostí. Výhodou této metody je, že k přenosu nepotřebujeme žádné vodiče. Jsme ovšem omezeni maximálním dosahem signálu a fyzické překážky dosah snižují. Vliv na kvalitu signálu mají také změny počasí, či počet zařízení v dosahu, vysílajících na stejné frekvenci. [9] Technologie využívající tento typ komunikace na krátkou vzdálenost jsou například WiFi, Bluetooth a další. Na delší vzdálenosti takto komunikují např. mobilní technologie, nebo třeba LoRa či Sigfox.

Vzorec pro výpočet vlnové délky frekvence (λ – délka vlny, v – rychlost šíření vlnění v prostředí, f - frekvence) [10]:

$$\lambda = \frac{v}{f}$$



Obrázek 2 – Příklad vlnových délek u častě používaných frekvencí [26]

Frekvenční pásma, ve kterých zařízení komunikují mohou být buď licencovaná, nebo nelicencovaná. Tyto pásma u nás stanovuje Český telekomunikační úřad. Pro komunikaci v nelicencovaném pásmu není třeba žádné specifické povolení. Tyto pásma jsou tedy hojně využívána. Z těchto důvodů může docházet k vzájemnému rušení v těchto pásmech, jelikož není zajištěna žádná ochrana proti rušení od dalších zařízení v tomto pásmu komunikujících. Licencovaná pásma naopak umožňují komunikaci pouze uživatelům, kteří mají povolení toto specifické pásmo využívat. Díky tomu zde k rušení nedochází. [9] Za licencovaná pásma se však platí a není tak jednoduché povolení k využívání získat. K typu komunikace na krátkou vzdálenost, využívané při vývoji systému v této práci, proto nejsou licencovaná pásma k použití vhodná.

Tabulka 1 - Využívání vymezených rádiových kmitočtů dle ČTU pro zařízení krátkého dosahu [25]:

27 MHz	Provozování je možné *
49 MHz	Provozování není přípustné (zařízení ruší rozhlasovou službu a necivilní aplikace)
230 – 400 MHz	Pásmo vyhrazeno pro účely obrany státu – žádný civilní provoz není přípustný
433 MHz	Provozování je možné * (Pouze přenos dat; bezdrátová sluchátka nejsou povolena)
470 – 694 MHz, 823 – 832 MHz	Provozování bezdrátových mikrofونů možné *
694 – 823 MHz, 832 – 862 MHz	Provozování mikrofونů již není povoleno
863 – 865 MHz	Provozování akustických aplikací je možné *
868 – 876 MHz, 915 – 921 MHz	Provozování je možné *
694 – 823 MHz, 832 – 862 MHz 876 – 915 MHz, 921 – 960 MHz	Pásmo provozu mobilních sítí – provozování jiných aplikací není přípustné
1,2 GHz	V Evropě není možné pro zařízení krátkého dosahu používat
1785 – 1805 MHz	Provozování bezdrátových mikrofونů je možné *
2,4 GHz	Provozování (RLAN, RFID, zařízení krátkého dosahu) je možné *

* Provoz možný dle všeobecného oprávnění k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu stanoveného ČTU (nyní č. VO-R/10/12.2019-9)

4.3 Použité Rf technologie¹

Moduly použity k rozšíření systému o možnost rádiové komunikace jsou od britského výrobce Rf Solutions, konkrétně model ZULU. Využit byl modul ZULU2-M89-SO s deskou ZULU-EVAL-M, díky které je možné tyto moduly jednoduše připojit pomocí USB či RS-232. Také je použit modul ZULU-M-ARDUINO, který slouží jako shield připojitelný k Arduinu.

¹ Technická dokumentace v přílohách (A, B)

Moduly mají následující parametry:

- Komunikace v bezlicenčním kmitočtovém pásmu 868 MHz
- Dosah komunikace až 2 km (při vhodných podmínkách a anténě)
- Přenosová rychlost až 115 kbps (56 kbps při použití shieldu pro Arduino)
- Nenáročnost na spotřebu el. energie
- Možnost jak vysílání, tak příjmu dat (Transceiver)
- Secure Data Protocol

4.3.1 Použité antény

Ke každému z modulů pro komunikaci je nutné připojit vhodnou anténu pracující v pásmu 868 MHz, jelikož toto frekvenční pásmo ke komunikaci využíváme. K výše zmíněným modulům byly využity antény FLEXI-M4-868 a FLEXI-SMA90-868 (modul pro Arduino), které jsou také k dostání od Rf Solutions. Anténa pro Arduino je polohovací, díky čemuž je možné směr antény upravit po zavěšení pod UAV.

4.4 Použitý hardware

V této kapitole je popsán hardware, který byl využit při tvorbě systému.

4.4.1 Arduino UNO

Arduino UNO je mikrokontrolerová deska vyvíjena Arduino.cc založena na Atmega328. Arduino UNO obsahuje rozhraní USB, napájecí konektor, 14 digitálních vstupně/výstupních pinů (z toho 6 může být použito pro PWM), 6 analogových pinů a Atmega328 mikrokontroler. Také podporuje sériovou komunikaci pomocí Tx a Rx pinů. Arduino je open-source platforma, to znamená, že desky si může každý upravit pro lepší funkcionalitu. [11] Arduino disponuje také vlastním softwarem (IDE), který je zdarma. Programuje se v jazyku C a C++.



Obrázek 3 - Arduino UNO [28]

4.4.2 Raspberry Pi 3B

Raspberry Pi je levný, malý, jednodeskový počítač vyvíjený společností Raspberry Pi Foundation, který je vybaven CPU, GPU, USB a dalšími porty a vstupně/výstupními piny a je schopen fungovat jako běžný počítač. Raspberry Pi první generace přišel na trh roku 2012. Od té doby vzniklo již několik novějších, vylepšených verzí. [12]



Obrázek 4 - Raspberry Pi [29]

4.5 Použité programové vybavení

Pro vývoj toho systému a následné testování a použití je nutné mít nainstalovaný software, který je dále rozepsán v jednotlivých podkapitolách.

4.5.1 Operační systém

Operačních systémů, které můžeme na RPi nainstalovat je více. Avšak doporučovaný přímo výrobcem je Raspberry Pi OS (Raspbian), což je systém založený na linuxové distribuci Debian. Tento systém jsem také zvolil.

4.5.2 Další použitý software

Terminal

Pro konfiguraci RF modulů je možné použít jakýkoliv terminálový emulátor pro sériové porty. Mnou použitá aplikace byla aplikace Terminal, která je přímo doporučovaná v technické dokumentaci k RF modulům. [13]

Arduino IDE

Pro programování Arduina bylo použito vývojové prostředí Arduino IDE, které je vyvíjeno přímo pro toto použití a je ke stažení na stránkách Arduina.

IDLE

K usnadnění programování v Pythonu slouží mnoho vývojových prostředí. Mnou zvolené prostředí bylo prostředí IDLE, které je nainstalováno přímo s instalací Pythonu.

4.5.3 Programovací jazyk

Při vývoji Arduina byl použit jazyk C a C++, který je využíván v Arduino IDE.

Další použitý programovací jazyk ke tvorbě skriptů byl jazyk Python 3, který je preferovaný jazyk na RPi (Lze ovšem použít i jazyky jiné). Jednou z jeho výhod je velké množství knihoven, kde některé z nich byly použity a vývoj usnadnily. Pro spuštění skriptů na RPi je tedy nutné mít Python 3 nainstalovaný a spouštět je jím (ve starších verzích Python skripty nemusí fungovat správně). Je také nutné mít instalovány všechny použité knihovny. Pokud nějaké nejsou k dispozici, je tedy nutné je doinstalovat.

4.5.4 Použité knihovny v Pythonu

V Pythonu byly z důvodu ulehčení práce a zvýšení efektivity využity tyto knihovny (nebo jejich části):

- **pyserial** – tato knihovna slouží k práci se sériovými porty
- **threading** – knihovna sloužící k práci s vlákny a umožňuje paralelismus
- **json** – knihovna k práci s formátem typu json
- **pycrypto** – obsahuje kolekci bezpečných hashovacích funkcí (jako SHA256 a další) a soubor šifrovacích algoritmů
- **base64** – knihovna ke kódování binárních dat na zobrazitelné ASCII znaky a naopak
- **binascii** – modul obsahující soubor metod k převodu mezi binárními daty a různými ASCII zakódovanými reprezentacemi

4.5.5 Použité knihovny pro Arduino

Na Arduino byly využity následující knihovny:

- **SPI** – knihovna umožňující komunikaci SPI (Serial Peripheral Interface) zařízením s Arduinem jako hlavním zařízením
- **SD** – tato knihovna umožňuje číst a zapisovat na SD karty (čtečka karet jako externí modul)
- **SoftwareSerial** – slouží k použití jiných digitálních pinů než primárně nastavených (0 a 1) k sériové komunikaci
- **PROGMEM (pgmspace)** – ukládání dat ve flash (programové) paměti místo v SRAM

4.6 Faktory ovlivňující volbu zmíněných HW a SW prvků

Požadavky, které vedly k výběru právě tohoto HW a SW jsou více rozebrány v praktické části v jednotlivých podkapitolách. Nicméně bylo třeba vybrat dostupný, kompatibilní hardware, který není příliš náročný na spotřebu elektrické energie a splňoval dále rozvedenou, potřebnou funkcionalitu. Programové vybavení bylo následně voleno, aby bylo s hardwarem kompatibilní a také dle dostupných recenzí a mých preferencí.

5 Praktická část

V této kapitole je popsán postup při návrhu a následné tvorbě komunikačního systému pro alternativní metodu dálkových odečtů v příhraniční oblasti pomocí UAV.

5.1 Popis a upřesnění požadavků pro volbu vhodného UAV

Základem spolehlivé funkčnosti celého systému je dobře zvolené UAV. Pro správnou funkčnost vytvořeného systému není nutné specifikovat konkrétní značku a typ dronu, nýbrž spíše uvést požadavky, které zvolený dron musí splňovat.

5.1.1 Typy UAV ke zvážení

Drony můžeme na základě mechanismu, který je udržuje ve vzduchu dělit do tří základních kategorií:

1. Více-rotorové drony

Tyto drony mají možnost vertikálního vzletu a přistání. Dalším benefitem při použití tohoto typu je, že můžeme během letu setrvat v zafixované pozici. Nevýhodou ovšem je, že tyto drony ke svému provozu spotřebují velké množství elektrické energie.[14] Vhledem k často krátkému doletu na jedno nabití je tento typ pro projekt využitelný jen tehdy, pokud sbíráme data pouze od menšího počtu elektroměrů, které od sebe nejsou příliš vzdálené.

2. Drony s fixním křídlem

Výhoda tohoto typu je, že ke svému letu spotřebuje oproti vrtulovému dronu značně méně energie. Ke svému pohybu totiž využívá plachtění. Tyto drony mají často také vyšší nosnost. Pro vzlet/přistání však potřebují určitý prostor a ve vzduchu musejí být udržovány stále v pohybu.[14] Z těchto důvodů, je tedy možné tento typ dronu využít, ovšem postrádá možnost setrvání nad elektroměrem při delší časové náročnosti vyčítání.

3. Hybridní drony (kombinace 1. a 2. typu)

Tento typ UAV využívá výhody dvou předchozích typů a kombinuje je dohromady. Je tedy možné využít vrtulí k vertikálnímu startu/přistání a následně využít křídla k plachtění. Je zde také možnost na určitých místech fixovat pozici. Právě proto se tento typ jeví jako nejvhodnější pro použití s navrženým systémem v této práci.

Tabulka 2 - Klady a zápory jednotlivých typů UAV

	Spotřeba el. energie	Fixace pozice	Vertikální start/přistání
Více-rotorové UAV	Vysoká	Ano	Ano
UAV s fixním křídlem	Nízká	Ne	Ne
Hybridní UAV	Průměrná	Ano	Ano

5.1.2 Další nutné vlastnosti UAV

Autopilot a GPS

Jedním z prvotních požadavků pro volbu vhodného UAV je zvolit dron, který je možné nastavit tak, aby byl na základě nastavených GPS souřadnic (místa s vyčítanými elektroměry) automaticky schopen tuto trasu prolétnout. Toto lze většinou nastavit v příslušné aplikaci.

Signalizace o dosažené pozici

Dalším požadavkem pro volené UAV je možnost výstupu PWM, který umožňuje předání pulsu externímu zařízení. Tyto pulsy se používají např. jako spoušť kamery zavěšené pod UAV.[15] Na základě obdržení tohoto pulsu bude následně vyslána žádost o data ke

konkrétnímu elektroměru. Tento výstup musí umožňovat nastavení takové, aby na konkrétní GPS souřadnici byl puls předán externímu zařízení.

Napájení externího zařízení

Následující nutnou výbavou dronu je možnost napájení externího zařízení. Pokud tuto možnost UAV nemá, je třeba přidat externí zdroj v podobě např. externí powerbanky a připočíst jeho hmotnost k celkové hmotnosti systému. Systém není, co se týče spotřeby energie, nijak náročný. Jak řídicí zařízení, tak rádiový modul jsou vhodné k použití pro nízkoenergetické účely.

Nosnost

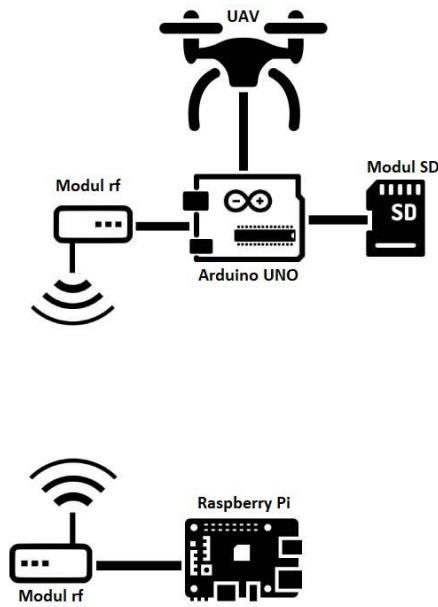
Dalším důležitým aspektem, který je třeba při volbě zvážit je maximální nosnost voleného UAV. Hardware, který je třeba na dron upevnit, váží bez přídavného napájení, upevňovacího materiálu a ochranného obalu pouhých cca 100 gramů. Jak již bylo zmíněno, je ale nutné připočítat minimálně ochranný obal a materiál použitý k upevnění.

Výdrž baterie

Jedním z důležitých požadavků, které musí zvolený dron splňovat, je délka doletu na jedno nabití vzhledem k počtu vyčítaných elektroměrů na trase, po které se UAV bude pohybovat. Vytvořený systém je univerzální, a přesně nespecifikuje počet elektroměrů, ze kterých budou data sbírána. Tato volba je individuální. Je tedy nutné určit trasu, na kterou bude tento systém nasazen a na základě toho zvážit, jak dlouho bude nutné UAV udržet ve vzduchu. Samotný sběr dat není nijak časově náročný (samozřejmě záleží zda načítáme soubor s již připravenými daty, či data přímo vyčítáme z elektroměru na základě žádosti a jaké atributy vyčítáme). Stačí, aby UAV pomaleji prolétlo nad daným elektroměrem, nejlépe nad ním na kratší dobu zafixovalo pozici.

5.2 Výběr vhodného hardwaru

Nejprve bylo třeba zvolit vhodný hardware, který je zodpovědný za samotné vyčítání a sběr dat. Každá ze dvou spolu komunikujících stran (strana UAV a strana elektroměrů) má však jiné požadavky. Z tohoto důvodu a z důvodu vyšší efektivity byl HW volen pro každou stranu zvlášť.



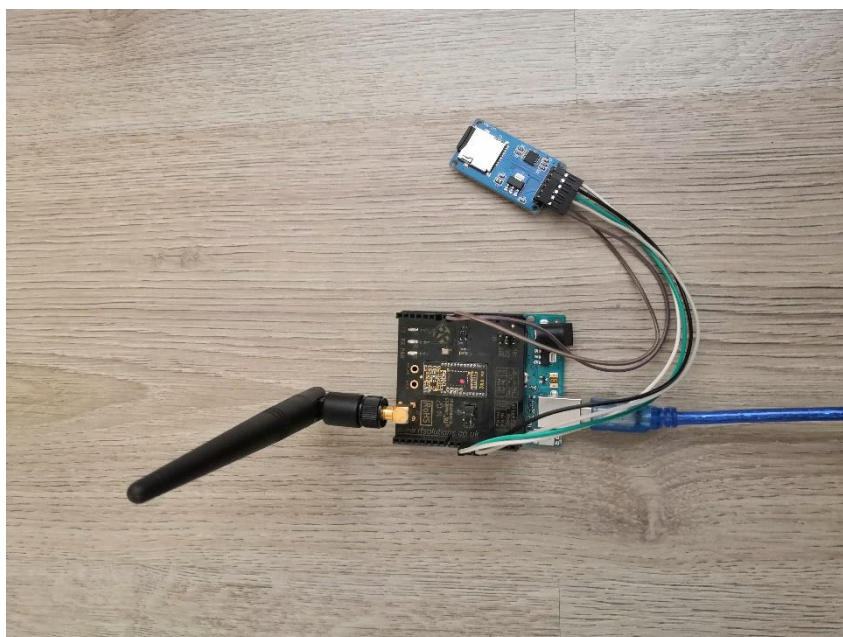
Obrázek 5 - Schéma zapojení vybraného hardwaru [Zdroj vlastní]

5.2.1 Volba hardwaru pro UAV

Při výběru hardwaru na stranu UAV byly zvažovány následující faktory:

- Možnost rozšíření o Rf komunikaci
- Vyslání žádosti o data k elektroměru
- Příjem vyčtených dat
- Ověření bezchybnosti příchozích dat (kontrolní součet)
- Uložení na paměťové médium
- Nenáročnost na spotřebu elektrické energie (napájení z externí baterie)
- Váha a velikost zařízení (bude zavěšeno pod UAV)

Z těchto výše zmíněných důvodů bylo zvoleno malé zařízení Arduino UNO, které splňuje požadavky. Nad toto zařízení byl usazen shield, který jej rozšířil o modul určený k rádiové komunikaci. Jelikož zařízení nemá paměťové médium, bylo nutné dokoupit rozšiřující modul pro čtení mikro SD karet, který je následně připojen na jednotlivé digitální a napájecí piny pomocí vodičů (v tomto případě na piny RF shieldu, který je propojuje s deskou Arduino). Do tohoto modulu byla následně umístěna micro SD karta (kapacita dle počtu vyčítaných elektroměrů).



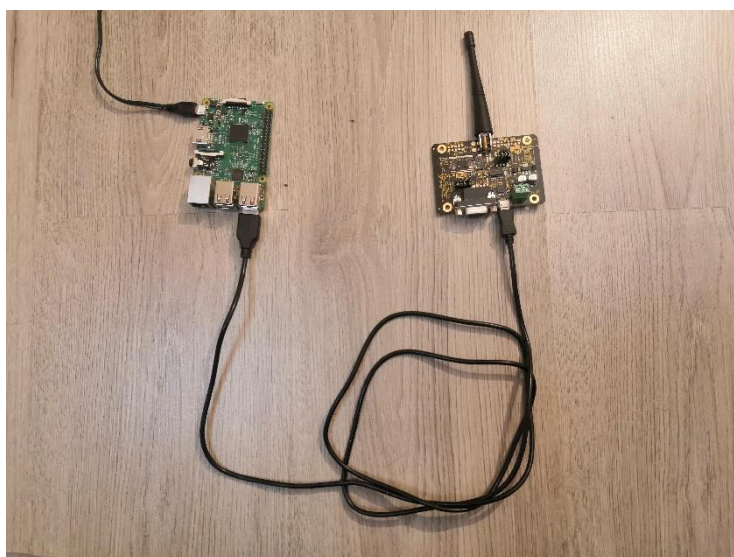
Obrázek 6 - Náhled na použité Arduino UNO s připojenými moduly [Zdroj vlastní]

5.2.2 Volba hardwaru k chytrým elektroměrům

Na straně chytrých elektroměrů bylo nutné zvažovat jiné faktory:

- Možnost připojení k elektroměru a schopnost vyčítat data
- Možnost rozšíření o rádiovou komunikaci
- Odesílání dat na základě přijaté žádosti
- Složitější operace s daty (šifrování dat)
- Úspora elektrické energie

Z těchto důvodů byl zvolen jednodeskový počítač Raspberry Pi konkrétně model 3B. Dalším důvodem pro volbu právě tohoto zařízení byl fakt, že tento projekt navazuje na práci, ve které jsou data z elektroměrů pomocí tohoto zařízení přímo vyčítána, a bylo proto vhodnou volbou i pro zajištění bezdrátové komunikace. Na trhu lze pořídit i novější verzi 4, avšak pro potřeby tohoto projektu je verze 3 dostačující. Tento malý počítač je napájen přes mikro USB a pomocí USB portů k němu můžeme připojit vybraný RF modul. Pomocí těchto portů je také možné zařízení připojit ke konkrétním elektroměrům.



Obrázek 7 - Raspberry Pi s připojeným rf modulem [Zdroj vlastní]

5.3 Výběr vhodných modulů pro bezdrátovou komunikaci

Pro výběr modulu pro zprostředkování bezdrátové komunikace byly zvažovány následující faktory:

- Dostatečný dosah komunikace
- Vhodnost pro použití v rámci oblasti Šumavy (průchod signálu přírodními překážkami)
- Nízká spotřeba elektrické energie
- Nízká hmotnost (modulu pro Arduino)
- Vhodné, bezlicenční frekvenční pásmo pro použití v ČR (Evropě)
- Možnost použití jak s Arduinem, tak s RPi
- Schopnost jak vysílání, tak příjmu dat (tedy transceiver)

Tyto zmíněné požadavky splňují moduly od Britského výrobce Rf Solutions a to konkrétně model Zulu. Tento typ je použit jako rozšiřující deska (shield) pro Arduino (ZULU-M-ARDUINO) a také na desce připojitelné pomocí USB (tedy možné připojit k Raspberry Pi)(ZULU-EVAL-M).

Moduly jsou opatřeny barevnou tečkou, která označuje verzi firmwaru. Verzi můžeme zjistit také v konfiguraci. Je proto třeba ověřit, že všechny použité moduly mají verzi stejnou, či zda jsou spolu jednotlivé verze schopny pracovat.



Obrázek 8 – ZULU-EVAL-M [Zdroj vlastní]



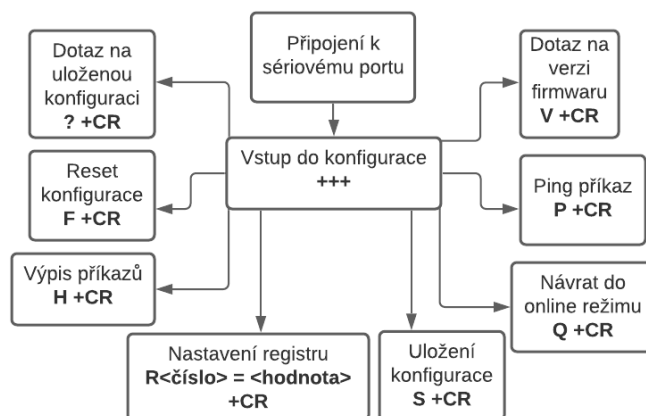
Obrázek 9 – ZULU-M-ARDUINO [Zdroj vlastní]

5.4 Konfigurace RF modulů

Aby byly moduly schopny vzájemné komunikace, bylo třeba je nejdříve nakonfigurovat. Konfigurace modulů pro RPi a Arduino je mírně odlišná, proto je konfigurace popsána ve dvou různých podkapitolách.

5.4.1 Konfigurace modulů pro Raspberry Pi²

Rf modul byl pomocí USB připojen k počítači s ovladačem pro převod na port sériový. Po spuštění aplikace Terminal (popsán v kapitole Použité programové vybavení) stačí pak připojit modul k PC, v aplikaci vybrat příslušný COM port a nastavit správné parametry pro připojení. Většinu z nich lze vyčíst z technické dokumentace, modulační rychlost (mezi PC a Rf modulem) je však nutné dle dokumentace nastavit hardwarově přímo na desce modulu pomocí propojení určitých pinů (např. 9600). Následně se připojíme k modulu. Po úspěšném připojení je možné začít s konfigurací. Konfigurace spočívá v odesílání určitých příkazů v podobě textových řetězců, které jsou opět vypsány v tech. dokumentaci.

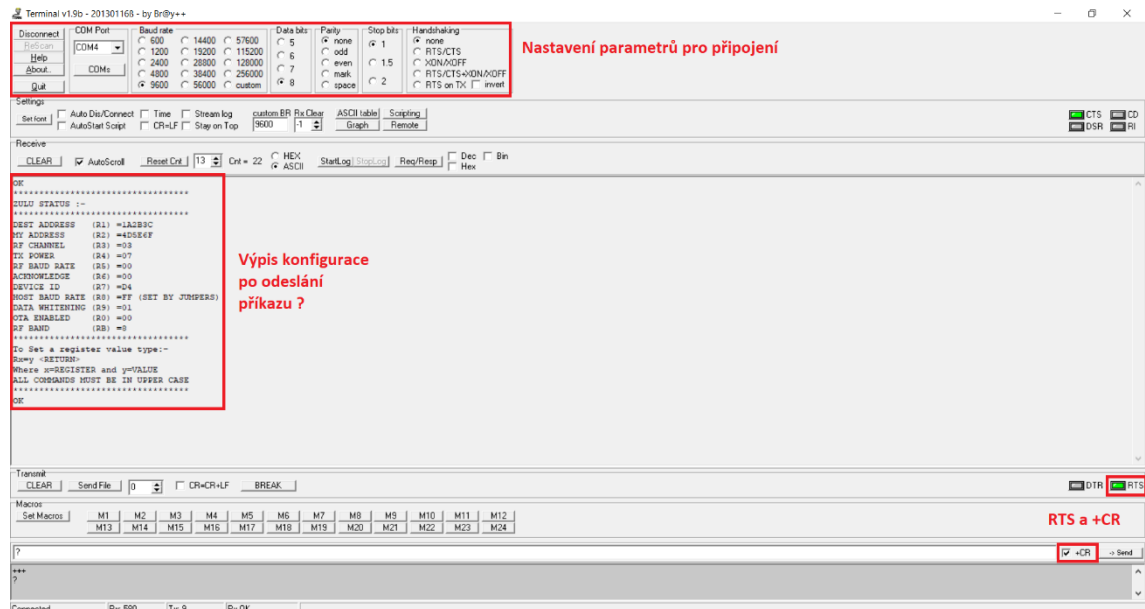


Obrázek 10 - Vývojový diagram popisující konfiguraci [Zdroj vlastní]

V aplikaci nejdříve zaškrtneme pole RTS (Request to Send). Pomocí příkazu +++ vstoupíme do konfiguračního módu. Následně zaškrtneme pole +CR, což nám umožní řetězce ukončovat znakem pro carriage return (nutno ukončovat všechny příkazy kromě +++) a můžeme začít nastavovat jednotlivé registry opět pomocí příkazů uvedených v technické dokumentaci (příkazy pro konfiguraci registrů jsou ve tvaru R<číslo registru> = <hodnota>). Nejprve nastavíme registr R1, který slouží k nastavení adresy zdrojové (zařízení, které právě konfigurujeme), následně R2 a adresu cílovou (zařízení, na které budeme odesílat). Je nutné, aby všechny moduly u elektroměrů (připojeny k RPi) měly

² Technická dokumentace v přílohách (A)

stejnou zdrojovou adresu a jako cílovou adresu pak adresu modulu u Arduina. Tím moduly zařadíme do jedné skupiny a vytvoříme komunikaci typu one-to-many, kterou následně budeme řídit pomocí napsaného skriptu v Pythonu. Následně pak nastavíme registr R3, kterým určíme kanál a určíme tak přesnou frekvenci (na všech modulech musí být nastaven stejný) a registr R5 pro nastavení přenosové rychlosti. Tímto způsobem můžeme konfigurovat i registry další (např. omezit vysílací výkon). Čtyři výše zmíněné jsou však povinné. Nakonec příkazem S uložíme konfiguraci a Q konfiguraci opustíme. Pokud bychom nechali modul v režimu konfigurace, nebyl by schopen komunikace. Pomocí příkazů se lze např. také zeptat na verzi firmwaru rf modulu (V), uloženou konfiguraci (?) a další.



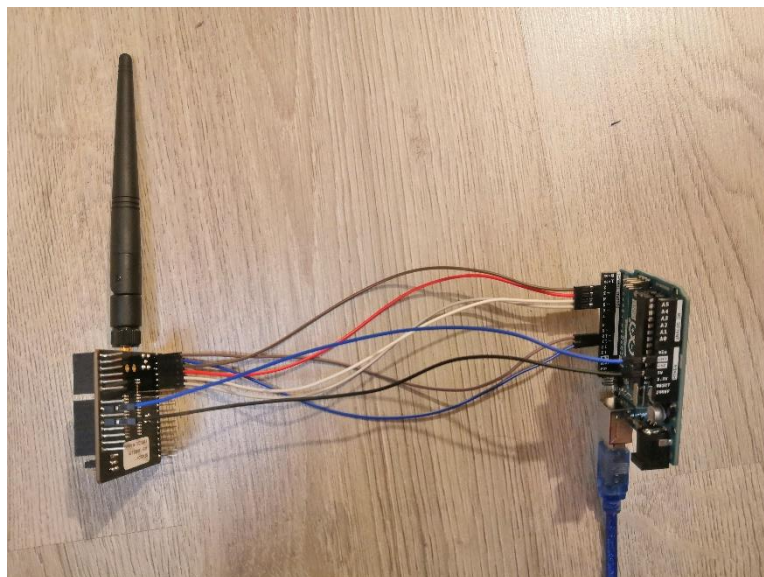
Obrázek 11 - Ukázka konfigurace v programu Terminal [Zdroj vlastní]

- Příklad konfigurace jednoho z modulů:
 - +++
 - ? <CR>
 - R1 = 1A2B3C <CR>
 - R2 = 4D5E6F <CR>
 - R3 = 1 <CR>
 - S <CR>
 - Q <CR>

5.4.2 Konfigurace modulu pro Arduino³

Ke konfiguraci shieldu pro arduino je třeba vytvořit konfigurační skript v Arduino IDE, který poté nahrajeme do arduina a shield nakonfigurujeme. Je možné buďto vytvořit skript takový, který rovnou předá modulu konfigurační příkazy odeslané ve správném pořadí, avšak praktičtější je vytvořit univerzální skript, který je nezávislý na momentálně použitých konfiguračních příkazech a můžeme ho použít kdykoli a podobně, jako Terminal pro konfiguraci modulů pro PC. Skript tedy bude čekat na řetězce zadané uživatelem pomocí sériového monitoru (lze spustit v Arduino IDE).

Jelikož při použití konfiguračního skriptu bude arduino zároveň připojeno sériově k PC, a bude od nás skrz sériový monitor přijímat konfigurační příkazy (popřípadě vypisovat odpovědi modulu), nebude možné k propojení s rf modulem využít defaultní piny určené k sériové komunikaci (Rx – příjem, Tx - vysílání), konkrétně piny 0 a 1. Je tedy nutné modul neusazovat nad arduino typicky jako shield, ale propojit ho s arduinem dle schématu v technické dokumentaci s tím rozdílem, že zvolíme pro vývody sloužící pro Tx a Rx jiné, volné digitální piny (např. piny 10 a 11). Zbytek použitých vývodů modulu propojíme s arduinem stejně, jak udává schéma na původně určené piny. Toho docílíme použitím propojovacích vodičů, či s využitím nepájivého pole. Následně importujeme knihovnu SoftwareSerial, která nám umožní použít i jiné, námi zvolené digitální piny jako piny Tx a Rx.



Obrázek 12 - Propojení pinů při konfiguraci modulu pro arduino [Zdroj vlastní]

³ Technická dokumentace v přílohách (B)

Po napsání a nahrání vytvořeného konfiguračního skriptu do arduina je možné spustit sériový monitor. Zde zvolíme správnou modulační rychlost, kterou je nutné mít nastavenou dle dokumentace pomocí propojených pinů na Rf shieldu vodiči (opět např. 9600 baud). Pod touto modulační rychlostí je také spuštěn sériový port v inicializační části skriptu. Následně můžeme začít se zasíláním konfiguračních příkazů dle technické dokumentace. Je však nutné si dát pozor na zakončení řádku před odesláním každého příkazu. Vstup do konfigurace (+++) se opět odesílá bez ukončovacího znaku. Pro další konfigurační příkazy je pak nutné zvolit ukončení opět znakem CR či znaky NL (nový řádek) & CR. Samotná konfigurace pak probíhá stejně, jak je popsáno u konfigurace modulů pro RPi (je však nutné postupovat dle dokumentace, jelikož část příkazů se může lišit). Vytvořený a použitý konfigurační skript je uveden v přílohách.

5.4.3 Test konfigurace

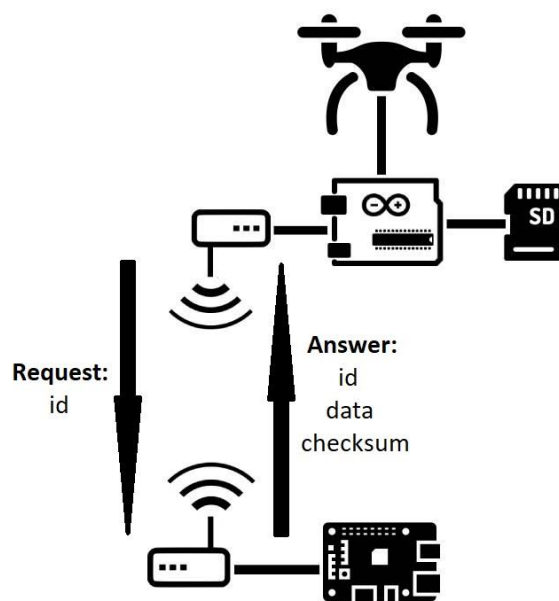
Po konfiguraci modulů je dobré otestovat, zda konfigurace proběhla správně. Stačí zapojit konfigurované moduly a opět pomocí aplikace Terminal zkusit odeslat zprávu z jednoho modulu na druhý. Pokud jsou nakonfigurovány správně, měli by být schopny komunikace (dle nastavených cílových adres).

Pokud chceme otestovat i shield pro Arduino, můžeme například napsat jednoduchý skript, který bude naslouchat a vypisovat přijaté řetězce v sériovém monitoru. Také můžeme napsat skript, který bude čekat na náš řetězec, který poté odešle na modul dle nastavené cílové adresy.

Je však nutné, aby byly moduly zpět v online režimu. Pokud budou ponechány v režimu konfiguračním, nebudou schopny komunikace.

5.5 Tvorba skriptů

Po sestavení a konfiguraci všech potřebných komponentů bylo využito již zmíněných programovacích jazyků pro tvorbu skriptů, které jsou zodpovědné za samotnou bezdrátovou komunikaci mezi UAV a elektroměry. Bylo třeba vytvořit skript zejména pro Arduino, které tvořilo základ tohoto projektu. Také musely být vytvořeny skripty, které jsou zodpovědné za navázání komunikace s Arduinem a odeslání vyčtených dat z elektroměrů pomocí RPi. Programování je rozděleno do následujících kapitol.



Obrázek 13 - Schéma komunikace použité k tvorbě skriptů [Zdroj vlastní]

5.5.1 Tvorba skriptu pro Arduino⁴

Arduino UNO nedisponuje příliš velkým množstvím paměti. Velikost paměti flash, kde je uložen napsaný sketch je 32 kB. Velikost SRAM (static random access memory), kde jsou ukládány vytvořené proměnné činí pouze 2 kB.[16] Je tedy nutné brát tyto „nedostatky“ během programování v potaz a nepoužívat zbytečně velké datové typy a nealokovat zbytečně velké prostory při deklaraci polí atp. Je také možné ukládat některé paměťově náročnější konstanty v paměti flash.

⁴ Skript v přílohách (C)

Jelikož arduino programujeme v jazyku c a c++, je možné využívat také datové typy proměnných z těchto jazyků. Ovšem pokud je to možné, je dobré se některým typům ve většině případů vyhnout (např. dynamicky alokovaný String lze nahradit polem typu char statické velikosti ukončovaným znakem \0 – c string). [17]

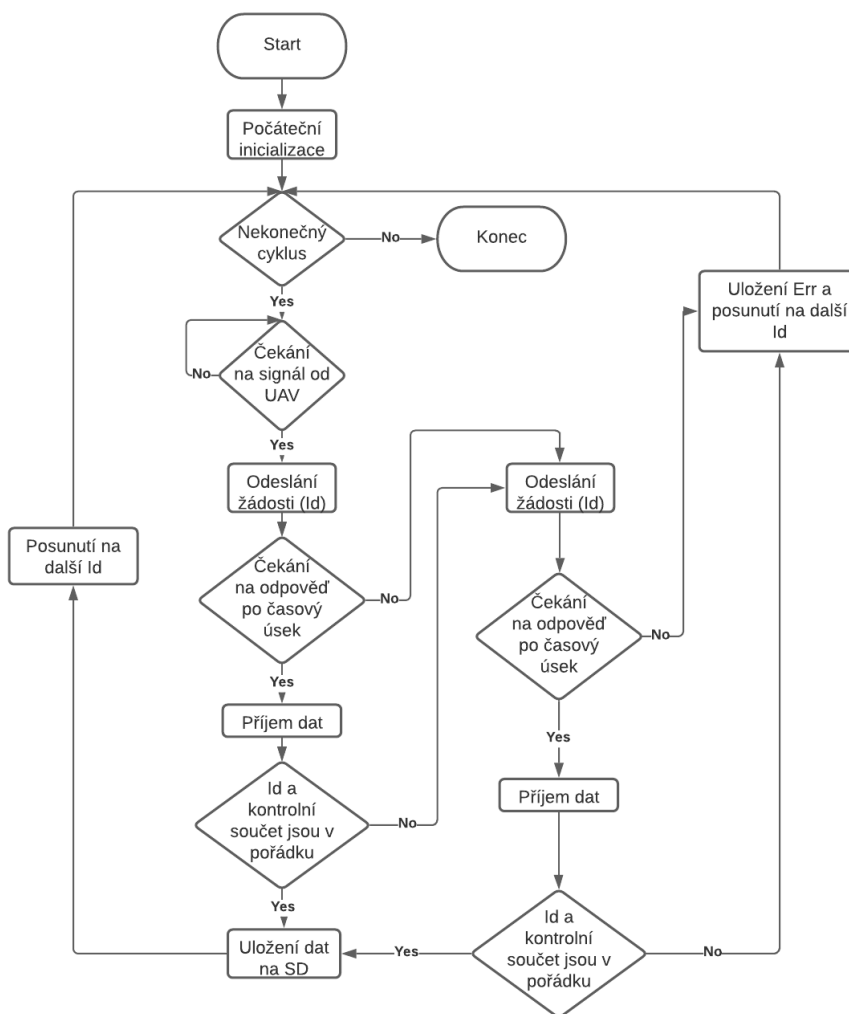
Funkce skriptu

Tento skript **UAV_Arduino.ino** tvoří hlavní část navrženého systému. Na vyzvání od UAV pomocí PWM pulsu je odeslána žádost s id elektroměru, který je na řadě pro vyčtení (tyto id jsou seřazeny v poli v pořadí, ve kterém budou vyčítány) a následně čeká po určité, předem nastavenou dobu na odpověď.

Pokud odpověď přijde, je celý přijatý stream rozdělen pomocí separátního znaku (zde zvolen znak „@“) na id modulu, ze kterého byl tento stream obdržen, samotná zašifrovaná data a kontrolní součet šifrovaných dat spočten pomocí funkce crc32. Pokud souhlasí id z příchozího streamu s id elektroměru, ke kterému byla vysílána žádost, spočte se kontrolní součet z přijatých, šifrovaných dat. Tento součet je následně porovnán se součtem, který byl součástí přijatého streamu. Pokud je shodný, pouze datová část z celého streamu je uložena na kartu SD do souboru DATA.txt (pokud soubor neexistuje, je vytvořen). Následně se opět vyčkává na puls od UAV pro vyčtení elektroměru dalšího.

Pokud se stane, že na žádost za předem nastavenou dobu nepřijde odpověď, nebo odpověď nepřišla od správného elektroměru (jiné id) či nesedí kontrolní součty (data nepřišla v pořádku), je odeslána žádost ještě jednou. Pokud i tato žádost selže, je do textového souboru zapsán řetězec „Err“ s id modulu, který značí chybu nejspíše na zařízení u elektroměru a je čekáno na puls pro vyčtení elektroměru dalšího.

Co se týče šifrovaných dat, ty zde rozšifrována nejsou, z důvodu bezpečnosti. Arduino je totiž zavěšeno pod UAV, které by v průběhu letu mohl někdo získat, tím pádem i data na SD, která by jinak byli v otevřené podobě. S konkrétními vyčtenými hodnotami tento skript nijak nepracuje. Data jsou tedy rozšifrována až po doručení na základnovou stanici a dále jsou využita k další analýze.



Obrázek 14 - Vývojový diagram funkce skriptu pro Arduino [Zdroj vlastní]

Struktura

1. Import použitých knihoven. Jedná se konkrétně o knihovny SPI, SD a PROGMEM. Jsou to knihovny, které jsou zde využity zejména k zápisu vyčtených dat na kartu SD a dále pro ukládání konstant do flash paměti arduina (nikoli SRAM).
2. Inicializace digitálních pinů použitých k propojení s rozšiřujícími moduly arduina (SD, CTS, RTS) a pinu analogového pro příjem pulsu PWM od UAV. Tyto piny jsou k nalezení v dokumentaci modulů či knihoven.

V další části jsou deklarovány (popřípadě inicializovány) veřejné proměnné a konstanty použité v projektu.

3. Ve funkci `setup()` následuje určení pinů pro vstupní či výstupní operace a spuštění funkcí `Serial.begin()` a `SD.begin()`.
4. Funkce `loop()` tvoří samotný cyklus zodpovědný za výše zmíněnou funkcionalitu algoritmu. Řídí vysílání a příjem dat a určuje podmínky ukládání či zahození přijatých dat. Dále se odkazuje na metody, které řeší konkrétní problémy.
5. Funkce `rec()` a `trans()` slouží k samotné komunikaci. Metoda `rec()` zprostředkovává příjem dat ze zásobníku. Data jsou zde čteny po bytech a ukládány do předem deklarovaného pole typu `char` (c string). Metoda `trans()` je naopak použita k vysílání předaných dat (zde konkrétně id elektroměru na který se dotazujeme). Při sériové komunikaci s modulem je také třeba kontrolovat piny pro CTS (clear to send) a RTS (request to send).

```
//Recieve data
void rec() {

    digitalWrite(CTS, LOW);

    Serial.readBytesUntil('\n', payload, 400);
    while (Serial.available()) {
        Serial.read();
    }

    digitalWrite(CTS, HIGH);

}
```

Obrázek 16 - Funkce `rec()` [Zdroj vlastní]

```
//Transmit data
void trans(int id) {
    digitalWrite(CTS, LOW);

    //empty buffer
    while (Serial.available()) {
        Serial.read();
    }
    digitalWrite(CTS, HIGH);

    //transmit
    while (digitalRead(RTS) == 1) {
    }
    Serial.println(id);
    digitalWrite(CTS, LOW);

}
```

Obrázek 15 - Funkce `trans()` [Zdroj vlastní]

6. Metoda `separate()` dělí obdržený stream na id, šifrovaná data a kontrolní součet na základě předem stanoveného separátního znaku.

```
void separate() {  
  
    recId = strtok(payload, "@");  
    recData = strtok(0, "@");  
    recCrc = strtok(0, "@");  
  
}
```

Obrázek 17 - Funkce `separate()` [Zdroj vlastní]

7. K ukládání dat do souboru na kartě SD slouží funkce `save()`, která soubor otevře a předaná data zapíše. Pokud nastala v přenosu chyba, zapíše error s číslem modulu, u kterého chyba nastala. Pokud ještě soubor `DATA.txt` neexistuje, je vytvořen.

```
//write to a file  
void save(char arrToSave[]){  
    myFile = SD.open("Data.txt", FILE_WRITE); //open or create file on SD  
    if (myFile){  
        myFile.println(arrToSave);  
        myFile.close(); //close file  
    }  
}
```

Obrázek 18 - Funkce `save()` [Zdroj vlastní]

8. Funkce `crc_update()` a `crc_string()` počítají kontrolní součet a byly převzaty z internetového článku.[18] Následně byla napsána funkce `checksum()`, která porovná vypočtený hash s hashem přijatým.

```
//Compare hash  
bool checksum(){  
  
    if (("0x" + String(crc_string(recData), HEX)) == String(recCrc)){  
        return true;  
    } else{  
        return false;  
    }  
}
```

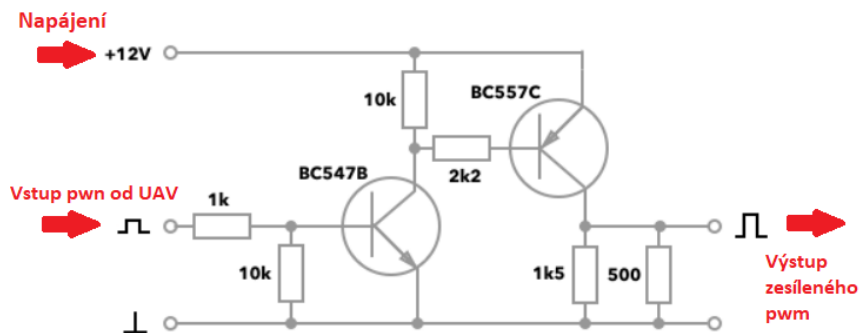
Obrázek 19 - Funkce `checksum()` [Zdroj vlastní]

Reakce na puls pwm od UAV

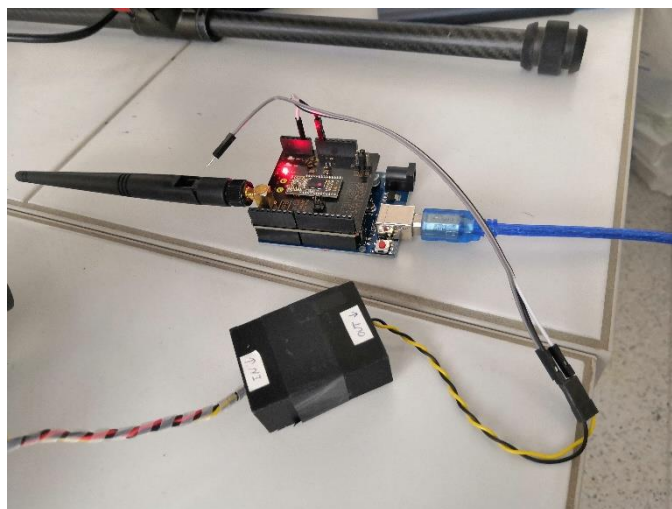
Puls od UAV je registrován použitím funkce `analogRead()`, pomocí které je na něj vyčkáváno ve smyčce. Aby bylo použití této funkce možné, je třeba připojit výstup pulsu pwm na jeden z analogových pinů Arduina. Druhý vývod od dronu je pak připojen na jeden z pinů označených jako GND (ground).

Při zkoušení odeslání žádosti k elektroměru na základě obdržení tohoto pulsu bylo použito UAV DJI Matrice 600 Pro, který právě nastavení těchto pwm výstupů umožňuje. Napětí pulsu je ovšem pouze 3,3 V. Při zprovoznování nastal problém, že arduino nebylo schopné tento puls registrovat. Proto byl použit zesilovač, který zmíněný problém vyřešil.

[19]



Obrázek 20 - Schéma použitého zesilovače k zesílení pulsu od UAV [19]



Obrázek 21 - Propojení arduina s UAV přes zesilovač [Zdroj vlastní]

Kontrolní součet

K výpočtu kontrolního součtu je vytvořeno spousta knihoven, ovšem při použití těchto knihoven nastal problém, že se vypočtený součet neshodoval, i když data přišla v pořádku. Tento problém spočívá buďto v použití rozdílných platforem, na kterých jsou součty počítány (python, Arduino), nebo byla chyba v implementaci na mé straně. Nakonec byly použity a upraveny metody, které byly nalezeny v článku na internetu, kde výpočet vychází stejně, jako při výpočtu crc32 použitého v mém skriptu v pythonu. [18]

Spuštění

Před nahráním do Arduina je nutné zadat id elektroměrů v pořadí, v jakém budou vyčítány do pole idArr[]. Rozšiřující moduly musí být připojeny tak, jak udává tento skript v počáteční inicializaci. Je také možné upravit dobu čekání na odpověď od elektroměrů přepsáním konstanty period, či modulační rychlost dle nastavení na rf shieldu. Poté je možné skript do Arduina nahrát (nahráváme bez připojených modulů).

```
/* Need to be connected to these pins */
const byte CTS = 3;
const byte RTS = 2;
const byte SDcard = 10;
const byte pulse = A0;

/* Inicialize before upload */
const unsigned int period = 2000;
const unsigned int baudRate = 9600;
int idArr[] = {28101, 28102};
```

Obrázek 22 - Příklad počáteční inicializace [Zdroj vlastní]

5.5.2 Tvorba skriptů pro RPi⁵

Veškeré skripty spouštěné na RPi byly vytvořeny v jazyce Python3. Tento jazyk obsahuje vysoké množství knihoven k využití.

⁵ Skript v přílohách (D)

Funkce skriptu

RPI_rf.py je skript, zodpovědný za přenos vyčtených dat z elektroměru na žádost arduina. Neslouží však přímo k vyčítání dat z elektroměru, jelikož navazuje na práci, ve které jsou data přímo vyčítána. Není však problém importovat potřebné knihovny a funkci pro přímé vyčítání pro určitý elektroměr přidělat.

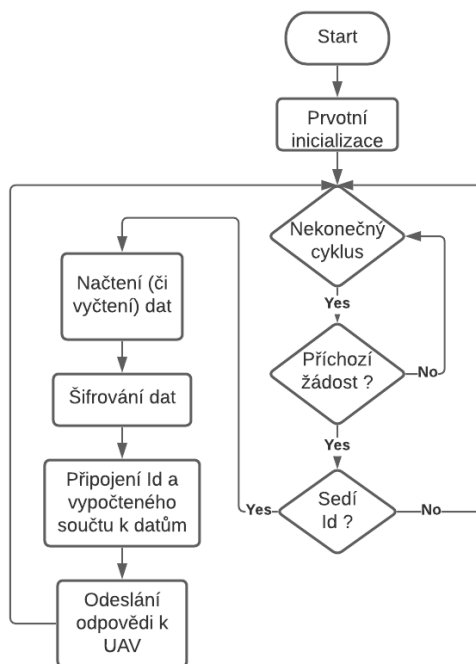
Je zde vyčkáváno na žádost od arduina k odeslání vyčtených dat. Pokud rf modul obdrží id od arduina, je porovnáno s id tohoto modulu, které je zadáno staticky před spuštěním skriptu. Pokud se id shoduje, znamená to, že žádost byla opravdu odeslána k tomuto elektroměru. Následně jsou tedy načtena data, z předem připraveného souboru typu json, který je obsahuje. Tento soubor může být například prepisován souborem s novými aktuálními daty každý určitý časový interval pomocí jiného skriptu. Je také možné přepsat připravenou funkci (`getData()`) tak, aby místo načítání souboru data přímo z elektroměru vyčetla do formátu json.

```
{
  "importedActivePower": 25612,
  "configMeterOnPlace": 11,
  "dateTime": "2020-10-30T14:00:45.470Z"
}
```

Obrázek 23 - Příklad načítaných dat formátu json [Zdroj vlastní]

Načtená data jsou poté zašifrována pomocí veřejného klíče uloženého v souboru typu key. K těmto datům je ještě přidán a oddělen separátním znakem (zde `@`) vypočtený kontrolní součet pouze z šifrovaných dat a opět id modulu, aby bylo možné ověřit, že je odpověď opravdu od tohoto elektroměru. Tento řetězec je poté odeslán zpět k Arduinu jako odpověď a čeká se na další žádost.

Pokud přijaté id nesedí s nastaveným id v tomto skriptu, nic se neděje a je vyčkáváno na další žádost (všechny moduly mají totiž stejnou adresu, ale rozdílné id).



Obrázek 24 - Vývojový diagram funkce skriptu pro RPi [Zdroj vlastní]

Struktura:

1. Import knihoven či jejich částí, které jsou dále ve skriptu používány. Jedná se o knihovny threading, serial, json, Crypto.PublicKey, Crypto.Cipher, base64 a binascii.
2. Otevření sériového portu s určitými parametry pro komunikaci s rf modulem, načtení souboru typu key a vytvoření veřejného klíče, stanovení cesty k souboru s průběžně vyčítanými daty a inicializace proměnné id_, která slouží k identifikaci modulu (elektroměru). Poté je spuštěno vlákno.
3. V metodě vlákna run() je spuštěn nekonečný cyklus, který vyčkává na žádost Arduina. Zde je také na základě přijatého id řízeno, zda je žádost přijata a data jsou načteny, nebo zda je žádost zahozena (žádost nesměruje k tomuto modulu). Je zde počítán kontrolní součet a složen řetězec k odeslání (id@šifrovanáData@kontrolníSoučet).

4. Funkce `getData()` se stará o samotné načítání dat ze souboru typu json. Jak již bylo zmíněno, je možné místo načítání dat ze souboru data z konkrétního elektroměru přímo vyčíst do formátu json upravením právě této metody.

```
def getData(self):
    try:
        with open(myFile) as myJson:
            loadedData = json.dumps(json.load(myJson))
    except IOError:
        loadedData = "err"
    return loadedData
```

Obrázek 25 - Funkce `getData()` [Zdroj vlastní]

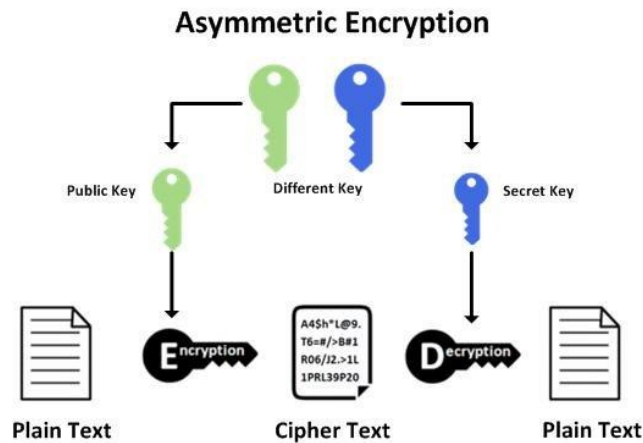
5. Funkce `encrypt()` slouží ke konkrétnímu šifrování dat veřejným klíčem pomocí asymetrického algoritmu RSA.

```
def encrypt(self, data):
    encryptor = PKCS1_OAEP.new(pubKey)
    encryptedData = encryptor.encrypt(data)
    return encryptedData
```

Obrázek 26 - Funkce `encrypt()` [Zdroj vlastní]

Šifrování dat

K šifrování dat je použit algoritmus RSA. Je to asymetrický šifrovací algoritmus, což znamená, že funguje na principu dvou rozdílných klíčů, a to klíč veřejný a soukromý. Veřejný klíč je sdílen se všemi uživateli (v našem případě se všemi RPi) a ty tímto klíčem data šifrují. Naopak klíč soukromý je nutné bezpečně uchovat a slouží k následnému dešifrování obdržených zašifrovaných dat.[20] Při použití symetrické šifry bychom riskovali prozrazení našeho jediného klíče při proniknutí do jakéhokoli ze zařízení u elektroměrů. Použití tohoto algoritmu je možné pomocí knihovny `pycrypto`.



Obrázek 27 - Schéma asymetrické kryptografie [27]

Při generování klíčů je nutné zvolit vhodnou délku klíče dle velikosti šifrovaných dat. Délka klíče, použitá v testování byla 2048 bitů. Pokud by nastala potřeba vyčítat dat více, a tudíž přesáhnout tuto velikost, je nutné zvolit klíč delší (4096 bitů by mělo být dostačujících). Poté je však nutné alokovat více paměti char poli payload[] ve skriptu pro Arduino, do kterého je ukládán příchozí řetězec (k velikosti klíče nutno připočíst id, kontrolní součet a oddělovací znaky).

Kontrolní součet

CRC (Cyclic redundancy check) je technika, která se používá k detekci chyb na datech. Je to vlastně hashovací funkce, která detekuje nechtěné změny dat používaná především při digitálním přenosu dat či zařízeními jako pevné disky. Nepoužívá se však proti úmyslným útokům (tj. neslouží jako algoritmus pro šifrování).[21] Ve zde popisovaném skriptu je použita verze crc32 z knihovny binascii pro python.

Spuštění

Před spuštěním tohoto skriptu je nutné zadat správný port, přes který je připojen rf modul a modulační rychlost, nastavenou na rf modulu. Dále je třeba do proměnné myKey zadat cestu k souboru typu key s uloženým veřejným klíčem, kterým budou data šifrována. Dalšími nutnými parametry je zadání cesty k souboru json s vyčítanými daty do proměnné myFile (pokud je tato metoda získání dat použita) a id elektroměru do proměnné id_. Poté je možné skript spustit v pythonu. Je nutné použít Python3.

```
sPort = "/dev/ttyUSB0"  
baudRate = 9600  
myKey = "/home/Tom/Desktop/pubKey.key"  
myFile = "/home/Tom/Desktop/values.json"  
id_ = "28101"
```

Obrázek 28 - Příklad inicializace parametrů [Zdroj vlastní]

```
[Tom@localhost ~]$ python3 /home/Tom/Desktop/RPi_rf.py
```

Obrázek 29 - Příklad spuštění skriptu [Zdroj vlastní]

5.5.3 Dodatečné skripty⁶

Pro použití a testování tohoto systému bylo nutné vytvořit ještě dodatečné skripty. Tyto skripty byly tvořeny taktéž v Pythonu3 či pro zařízení arduino.

Skript **AsGenerate.py** je jednoduchý skript sloužící ke generaci dvojice soukromého a veřejného klíče pomocí algoritmu RSA a uložení do souborů typu key. Veřejný klíč je poté distribuován na veškerá zařízení u elektroměrů (Rpi). Soukromý klíč je nutné naopak uchovat. Předem inicializovaná velikost klíče pro generování je 2048 bitů. Tato velikost lze samozřejmě upravit (např. zvýšit na 4096 bitů).

Další vytvořený skript **AsDecrypt.py** slouží k rozšifrování dat přijatých Arduinem a uložených na SD. Tento skript čte soubor po řádcích (každý řádek obsahuje data od

⁶ Vzorové skripty v přílohách (E, F, G)

jednoho elektroměru), které následně dešifruje a vypíše do souboru typu json. Pokud nastala chyba při sběru dat od konkrétního elektroměru, je vypsáno, že se vyskytla chyba.

Pro konfiguraci rf modulu pro Arduino UNO byl napsán sketch **config.ino**, který je více popsán v kapitole Konfigurace Rf modulů. Po nahrání tohoto skriptu do arduina a připojení konkrétního shieldu je možné rf modul konfigurovat (nevyužívat hardwarové serial, nutno zvolit jiné digitální piny pro rx a tx).

5.6 Testování navrženého systému

Testování vytvořeného systému probíhalo ve více fázích. Kromě níže popsaných testů bylo také testováno v průběhu vývoje.

Prvotní testování bylo provedeno ihned po konfiguraci rf modulů. Tento test byl proveden zejména ke zkoušce uložené konfigurace a zjištění vzdálenosti, na kterou je modul schopen komunikovat. Na přímou viditelnost a plný vysílací výkon byl modul testován na vzdálenost cca 620 metrů a obstál bez problémů. Dle technické dokumentace by však měl být schopen při vhodných podmínkách, s vhodnou anténou a vyhovující konfigurací (vysílací výkon, přenosová rychlost) takto dosáhnout až ke dvěma kilometrům (k takové vzdálenosti se ale v tomto projektu nepřiblížíme). Při komunikaci přes přírodní překážku (hliněný val) se vzdálenost zkrátila cca na 200 metrů. Přes 3 cihlové zdi byl dosah již značně omezen, proto je nutné komunikaci přes příliš mnoho stěn zamezit. Jelikož bude komunikace probíhat shora a UAV bude mít možnost dostat se téměř nad elektroměr, neměl by být proto s komunikací přes překážky takový problém. Kladný je také faktor, že moduly využívají frekvenci 868 MHz.

Druhý test byl proveden k vyzkoušení samotného navrženého systému, ovšem bez čekání na signál od UAV. Tento signál byl nahrazen časovým zpožděním mezi jednotlivými žádostmi o data. Tento test byl proveden se dvěma moduly pro RPi (simuluje moduly u dvou elektroměrů). Bylo zde testováno zejména funkčnost automatické komunikace a podmínky pro ukládání či zahození přijatých dat. Menší problém nastal s kontrolním součtem, což bylo opraveno.

Další test byl proveden již za přítomnosti UAV. Bylo testováno odeslání žádosti o data, až po obdržení signálu od autopilota dronu a následně vyčkání na signál další. UAV

použité pro testování bylo DJI Matrice 600 Pro. Puls pwm bylo nutné mezi dronem a Arduinem vést skrz zesilovač, který tento puls zesílil. Bez toho nebylo možné puls zaznamenat (Použití zesilovače více popsáno v kapitole Tvorba skriptu pro Arduino). Tento puls byl také simulován tlačítkem. Zbytek probíhal již stejně jak u předešlého testu.

Na závěr byl systém otestován s různými přenosovými rychlostmi. Nejdříve bylo testováno s přenosovou rychlostí 9600 baud (jak hostitelská, tak rf), která byla využívána i při vývoji. Následně bylo zvyšováno až na hostitelskou přenosovou rychlost 38400 baud a rf přenosovou rychlost 56000 baud (maximální rychlost modulu pro Arduino). I tento test probíhal bez problémů na vzdálenost cca 150 metrů.

6 Diskuse a inspirace pro navazující práce

Vývojem podobných systémů pro UAV se zabývá mnoho dalších prací, či studií a použití dronů není tedy žádnou novinkou. Drony začínají být hojně využívány například v zemědělství, kde mohou zajišťovat činnosti jako mapování, zalévání a další podobné funkce. [22] Drony jsou již využívány i v energetice. Existuje např. studie, která se zabývá vyhledáváním krádeží elektrické energie právě pomocí UAV. [23]

V této práci je navržen systém, který slouží jako alternativní metoda ke sběru dat od elektroměrů pomocí UAV. Systémů, které se k automatickým odečtům elektroměrů využívají je dostatek, ovšem tyto běžně užívané technologie nemají pokrytí ve všech lokalitách. [24] Automatický sběr dat je zde pak obtížný. Právě z tohoto důvodu se tato práce zabývá vývojem systému, který bude právě v těchto lokalitách možné využít. Dálkové odečty pomocí UAV by měly být možné ve všech, i špatně přístupných lokalitách. Nevýhoda oproti běžným metodám ovšem je, že sběr dat pomocí UAV je závislý na časovém intervalu, ve kterém UAV prolétá nad konkrétními elektroměry, jelikož se také musí dobíjet a nelze data sbírat tedy tak často, jako v případě technologií běžně používaných.

6.1 Návrhy na zlepšení

Systém, vytvořený v této práci je zodpovědný za sběr dat od elektroměrů a jejich následné uchování v zabezpečené podobě na paměťovém médiu, pro další zpracování po dorážení na „základnovou stanici“. Kartu SD je však nutné ručně vyjmout a data pomocí napsaného skriptu rozšifrovat a zařadit k další analýze (např. do databáze). Tyto kroky by se daly v budoucnu vylepšovat a automatizovat.

Po každém vyčtení všech elektroměrů na trase a přistání UAV na dobíjecí stanici by bylo možné realizovat automatické odeslání dat od Arduina opět na základě signalizačního pulsu od dronu k dalšímu rádiovému modulu. Po přijetí by data byla buďto automaticky rozšifrována a uložena k další analýze, nebo by byla ještě pomocí jiné, přenosové technologie na delší vzdálenosti (mobilní síť, Sigfox) dopravena na vzdálenou základnu a zpracována až následně.

Další možné vylepšení systému by se mohlo zaměřovat na automatizovanou výměnu veřejných klíčů používaných k šifrování dat před odesláním směrem k UAV. Nové klíče by mohly být po určité časové době vygenerovány, a následně opět pomocí UAV distribuovány k jednotlivým elektroměrům. Bylo by však nutné vyzkoušet, zda by k těmto operacím bylo zařízení Arduino dostačující.

7 Závěr

Na úvod byla zpracována rešeršní část, kde byla popsána důležitost získávání dat z elektroměrů. Dále byla vypracována rešerše, která se zaměřuje na běžně používané technologie pro vyčítání elektroměrů a některé z těchto metod byly více rozebrány. Rešerše je více zaměřena na sběr dat pomocí bezdrátových technologií, jelikož právě tento typ je nejbližší systému, který měl být vytvořen. Jsou také popsány hardwarové prvky a programové vybavení používané k tvorbě samotného systému.

Následně byl zvolen vhodný hardware a rádiový modul. Rádiové moduly byly vhodně nakonfigurovány a konfigurace testována. Pro zařízení Arduino Uno byl vytvořen skript, sloužící k vysílání žádostí o data vyčtená z elektroměrů a jejich následné přijetí a zpracování. Dále byl vytvořen skript v jazyku Python 3, který vyčkává na žádost a na základě té odesílá vyčtená data zpět k UAV. Byly také napsány jednoduché dodatkové skripty potřebné k zprovoznění a testování systému, a to konkrétně skript ke generaci RSA klíčů, skript pro následné dešifrování dat a konfigurační skript pro Arduino.

Nakonec se podařilo vytvořit funkční systém, který by bylo možné použít pro sběr dat z chytrých elektroměrů v oblastech, kde jiné, běžně používané metody dálkových odečtů nejsou možné. Pokud by se však mělo jednat o použití v lokalitách, kde je možné spolehlivě využít metody odečtů zmíněné v rešeršní části práce, není z efektivních důvodů použití tohoto systému příliš vhodné. Data totiž nemohou být sbírána tak často, jelikož UAV je nutné také nabíjet.

Systém byl následně několikrát testován, a to s příkladem dat, která jsou skutečně od elektroměrů vyčítána. Tato data poskytl kolega, který samotné vyčítání řeší. Testováno bylo také vysílání žádosti o data na popud UAV a samotný rádiový modul. Byly také zkoušeny různé přenosové rychlosti.

Co se týče funkce, systém umí data od elektroměrů sesbírat, ověřit, zda jsou ve správné podobě a od správného elektroměru a zašifrované je uložit na připojené paměťové medium. Samotné vyčítání je buď možné provádět nezávislým skriptem mezi jednotlivými sběry těchto dat, nebo jednoduše přepsat připravenou metodu a data přímo vyčítat po obdržení žádosti o tato data (více popsáno v kapitole „Tvorba skriptů“). Po navrácení UAV na „základnu“ je však nutné médium ručně vyjmout a následně pomocí

vedlejších skriptů data rozšifrovat a uložit k další analýze. Zde se tedy nabízí možnost dalšího vylepšení, a to automatizace těchto nedostatků, které jsou více popsány v kapitole „Diskuse a inspirace pro navazující práce“.

8 Seznam použité literatury

[1] KHALIFA, Tarek, Kshirasagar NAIK a Amiya NAYAK. *A Survey of Communication Protocols for Automatic Meter Reading Applications*. 2011, **13**(2), 168-182. ISSN 1553-877X. Dostupné z: doi:10.1109/SURV.2011.041110.00058

[2] HERZÁNOVÁ, Jessica. *ANALÝZA dat z pilotního nasazení smart meteringu PREDistribuce, a.s.* Praha, 2018. Dostupné také z: <https://dspace.cvut.cz/bitstream/handle/10467/80463/F3-DP-2019-Herzanova-Jessica-Diplomova%20prace%20-%20Herzanova%20Jessica.pdf?sequence=-1&isAllowed=y>. Diplomová práce. České vysoké učení technické v Praze.

[3] AYAR, Muharrem, Serhat OBUZ, Rodrigo D. TREVIZAN, Arturo S. BRETAS a Haniph A. LATCHMAN. A Distributed Control Approach for Enhancing Smart Grid Transient Stability and Resilience. *IEEE Transactions on Smart Grid* [online]. 2017, **8**(6), 3035-3044 [cit. 2021-04-13]. ISSN 1949-3053. Dostupné z: doi:10.1109/TSG.2017.2714982

[4] A. A. Khan, T. Wiens and M. Massoth, "Stability improvement solution of the smart power grid by an analysis of voltage variation in intelligent buildings", SMART 2014: The Third International Conference on Smart Systems Devices and Technologies, pp. 13-19, 2014

[5] KAPOUN, Vladislav. *Smart grids - chytré sítě v energetice*. Brno, 2013. Dostupné také z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=69209. Bakalářská práce. Vysoké učení technické v Brně.

[6] PENG, Wenjun, Weidong ZHANG a Bo AN. Radiation Measurement and Analysis of Power Line Communication System for Power Meter Reading. *IEEE Access*. 2020, **8**, 40989-40999. ISSN 2169-3536. Dostupné z: doi:10.1109/ACCESS.2020.2976520

[7] SKRÁŠEK, Tomáš. *Úzkopásmová PLC komunikace se standardy G3-PLC, PRIME a IEEE-1901.2*. Brno, 2015. Dostupné také z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=100768. Diplomová práce. Vysoké učení technické v Brně.

- [8] What is Power Line Communication? *EE Times* [online]. 2011 [cit. 2021-04-08]. Dostupné z: <https://www.eetimes.com/what-is-power-line-communication/>
- [9] KYML, David. *Přenosové charakteristiky bezdrátových sítí a jejich negativní ovlivnění přírodními jevy*. České Budějovice, 2015. Dostupné také z: <https://theses.cz/id/p6ddwk/15012387>. Bakalářská práce. Jihočeská Univerzita.
- [10] Frekvence. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021, 13. 1. 2020 [cit. 2021-04-04]. Dostupné z: <https://cs.wikipedia.org/wiki/Frekvence>
- [11] AQEEL, Adnan. *Introduction to Arduino Uno* [online]. 2018 [cit. 2021-04-03]. Dostupné z: <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html>
- [12] AQEEL, Adnan. *Introduction to Raspberry Pi 3* [online]. 2018 [cit. 2021-04-03]. Dostupné z: <https://www.theengineeringprojects.com/2018/04/introduction-to-raspberry-pi-3.html>
- [13] *Terminal: com port development tool* [online]. [cit. 2021-04-04]. Dostupné z: <https://sites.google.com/site/terminalbpp/>
- [14] FOTOUHI, Azade, Haoran QIANG, Ming DING, Mahbub HASSAN, Lorenzo Galati GIORDANO, Adrian GARCIA-RODRIGUEZ a Jinhong YUAN. *Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges* [online]. 2019, **21**(4), 3417-3442 [cit. 2021-04-08]. ISSN 1553-877X. Dostupné z: doi:10.1109/COMST.2019.2906228
- [15] Automatický trigger pro termokameru WIRIS PRO. *Mind blog* [online]. c2021, 27.12.2019 [cit. 2021-04-04]. Dostupné z: <https://www.voxcafe.cz/mindblog/clanky/drony/automaticky-trigger-pro-termokameru-wiris-pro.html>
- [16] Memory. *Arduino.cc* [online]. c2021, 5 February 2018 [cit. 2021-04-03]. Dostupné z: <https://www.arduino.cc/en/Tutorial/Foundations/Memory>

- [17] The Evils of Arduino Strings. *Majenko Technologies* [online]. 4 February 2016 [cit. 2021-04-03]. Dostupné z: <https://hackingmajenkoblog.wordpress.com/2016/02/04/the-evils-of-arduino-strings/>
- [18] Arduino CRC-32. *Excamera* [online]. February 2011 [cit. 2021-04-03]. Dostupné z: <https://excamera.com/sphinx/article-crc.html>
- [19] Konstrukce systému spektrálních kamer, termokamery a DJI A3. *Mind blog* [online]. c2021, 27.06.2018 [cit. 2021-04-04]. Dostupné z: <https://www.voxcafe.cz/mindblog/clanky/drony/konstrukce-systemu-spektralnich-kamer,-termokamery-a-dji-a3.html>
- [20] RSA Algorithm in Cryptography. *Geeks for Geeks* [online]. Noida, 05 Jan 2021 [cit. 2021-04-03]. Dostupné z: <https://www.geeksforgeeks.org/rsa-algorithm-cryptography/>
- [21] Cyclic Redundancy Check (CRC). *Techopedia* [online]. c2021 [cit. 2021-04-03]. Dostupné z: <https://www.techopedia.com/definition/1793/cyclic-redundancy-check-crc>
- [22] KIM, Jeongeun, Seungwon KIM, Chanyoung JU a Hyoung Il SON. Unmanned Aerial Vehicles in Agriculture: A Review of Perspective of Platform, Control, and Applications. *IEEE Access* [online]. 2019, 7, 105100-105115 [cit. 2021-04-13]. ISSN 2169-3536. Dostupné z: doi:10.1109/ACCESS.2019.2932119
- [23] BHALSHANKAR, Surekha S. a C. S. THORAT. MAHADISCOM electrical theft controlling mechamisum: (Smart grid advanced metering infrastructure and drone operated technology for controlling theft by direct hooking). *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* [online]. IEEE, 2017, 2017, , 1647-1650 [cit. 2021-04-13]. ISBN 978-1-5090-4442-9. Dostupné z: doi:10.1109/WiSPNET.2017.8300040
- [24] Okokpujie, Kennedy O. and Abayomi-Alli,, A. and Abayomi-Alli,, O. and Odusami,, M. and Okokpujie, Imhade P. and Akinola, O. A. (2017) AN AUTOMATED ENERGY METER READING SYSTEM USING GSM TECHNOLOGY. In: 2nd International Conference on Applied Information Technology.

- [25] Využívání vymezených rádiových kmitočtů. *Český telekomunikační úřad* [online]. c2018 [cit. 2021-04-11]. Dostupné z: <https://www.ctu.cz/vyuzivani-vymezenych-radiovyh-kmitoctu>
- [26] Jak zvolit správnou anténu ve 3 krocích. *Vyvoj.hw.cz* [online]. c1997-2014, 8. 10. 2020 [cit. 2021-04-04]. Dostupné z: <https://vyvoj.hw.cz/jak-zvolit-spravnou-antenu-ve-3-krocich.html>
- [27] Ons Jallouli. *Chaos-based security under real-time and energy constraints for the Internet of Things*. Signal and Image processing. UNIVERSITE DE NANTES, 2017. English.
- [28] Arduino Uno Rev3. *Arduino.cc* [online]. c2021 [cit. 2021-04-11]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>
- [29] Raspberry Pi 3 Model B. *Pi-shop.ch* [online]. c2012-2020 [cit. 2021-04-11]. Dostupné z: <https://www.pi-shop.ch/raspberry-pi-3>

9 Seznam obrázků

Obrázek 1 - Komunikace s využitím PLC [8]	8
Obrázek 2 – Příklad vlnových délek u častě používaných frekvencí [26]	9
Obrázek 3 - Arduino UNO [28]	12
Obrázek 4 - Raspberry Pi [29]	12
Obrázek 5 - Schéma zapojení vybraného hardwaru [Zdroj vlastní]	18
Obrázek 6 - Náhled na použité Arduino UNO s připojenými moduly [Zdroj vlastní] ...	19
Obrázek 7 - Raspberry Pi s připojeným rf modulem [Zdroj vlastní]	20
Obrázek 8 – ZULU-EVAL-M [Zdroj vlastní]	21
Obrázek 9 – ZULU-M-ARDUINO [Zdroj vlastní].....	21
Obrázek 10 - Vývojový diagram popisující konfiguraci [Zdroj vlastní].....	22
Obrázek 11 - Ukázka konfigurace v programu Terminal [Zdroj vlastní]	23
Obrázek 12 - Propojení pinů při konfiguraci modulu pro arduino [Zdroj vlastní]	24
Obrázek 13 - Schéma komunikace použité k tvorbě skriptů [Zdroj vlastní]	26
Obrázek 14 - Vývojový diagram funkce skriptu pro Arduino [Zdroj vlastní]	28
Obrázek 15 - Funkce trans() [Zdroj vlastní]	29
Obrázek 16 - Funkce rec() [Zdroj vlastní]	29
Obrázek 17 - Funkce separate() [Zdroj vlastní]	30
Obrázek 18 - Funkce save() [Zdroj vlastní].....	30
Obrázek 19 - Funkce checkSum() [Zdroj vlastní].....	30
Obrázek 20 - Schéma použitého zesilovače k zesílení pulsu od UAV [19].....	31
Obrázek 21 - Propojení arduina s UAV přes zesilovač [Zdroj vlastní].....	31
Obrázek 22 - Příklad počáteční inicializace [Zdroj vlastní]	32
Obrázek 23 - Příklad načítaných dat formátu json [Zdroj vlastní].....	33
Obrázek 24 - Vývojový diagram funkce skriptu pro RPi [Zdroj vlastní].....	34
Obrázek 25 - Funkce getData() [Zdroj vlastní].....	35
Obrázek 26 - Funkce encrypt() [Zdroj vlastní]	35
Obrázek 27 - Schéma asymetrické kryptografie [27].....	36
Obrázek 28 - Příklad inicializace parametrů [Zdroj vlastní]	37
Obrázek 29 - Příklad spuštění skriptu [Zdroj vlastní]	37

10 Seznam tabulek

Tabulka 1 - Využívání vymezených rádiových kmitočtů dle ČTU pro zařízení krátkého dosahu [25]:	10
Tabulka 2 - Klady a zápory jednotlivých typů UAV	16

11 Seznam příloh

Příloha A – Technická dokumentace k ZULU-M89-SO a ZULU-EVAL-M

Příloha B – Technická dokumentace k ZULU-M-ARDUINO

Příloha C – Skript **UAV_Arduino.ino**

Příloha D – Skript **RPI_rf.py**

Příloha E – Dodatkový skript **config.ino**

Příloha F – Dodatkový skript **AsGenerate.py**

Příloha G – Dodatkový skript **AsDecrypt.py**