



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MONITOROVÁNÍ VYUŽITÍ ULTRALEHKÝCH LETADEL

MONITOR OF EXPLOITATION IN ULTRAMICRO AVIATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MATĚJ KUDERA

VEDOUCÍ PRÁCE

SUPERVISOR

Prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2021

Zadání bakalářské práce



Student: **Kudera Matěj**
Program: Informační technologie
Název: **Monitorování využití ultralehkých letadel**
Monitor of Exploitation in Ultramicro Aviation
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte literaturu na téma provoz ultralehkých letounů, dále prostudujte možnosti "embedded systémů" a mobilních systémů v letadlech.
2. Navrhněte postup/koncepci aplikace pro monitorování využití letadla/vrtulníku se zaměřením na snadnou instalaci a nízké pořizovací náklady na techniku v letadle.
3. Popište postup zjišťování parametrů provozu a diskutujte dosažitelné vlastnosti řešení.
4. Implementujte systém a demonstруйте jeho funkčnost na vhodném příkladu použití.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

Abstrakt

Cílem této bakalářské práce bylo vytvořit systém, který bude zaznamenávat provoz ultralehkého letounu, do kterého bude nainstalován. V práci jsou rozebrány principy potřebné pro návrh tohoto systému a je vytvořen podrobný návrh, který je postaven na získaných znalostech o jednotlivých principech. Vytvořený systém se skládá ze dvou částí. Vestavěného zařízení, které je zabudováno do ultralehkého letounu a zaznamenává, zda je v pohybu a mobilní aplikace, která umožňuje jednoduchý přístup k záznamům pořízených vestavěným zařízením. Mobilní aplikace byla zhotovena pro operační systém Android za pomoci vývojového prostředí Android Studio a vestavěné zařízení bylo implementováno pomocí minipočítače Raspberry Pi, ke kterému byly připojeny potřebné periferie. Na závěr byla otestována funkčnost systému pomocí vytvořených testů a byly provedeny testovací lety ultralehkého letounu se zabudovaným vestavěným zařízením.

Abstract

The aim of this bachelor's thesis was to create a system that will record the operation of an ultralight aircraft in which it will be installed. The thesis analyzes the principles important for the design of this system and creates a detailed proposal, which is based on the acquired knowledge of individual principles. The created system consists of two parts. An embedded device that is built into the ultralight aircraft and records whether it is in motion and a mobile application that allows easy access to records made by the embedded device. The mobile application was made for the Android operating system using the Android Studio development environment and the embedded device was implemented using a Raspberry Pi minicomputer, to which the necessary peripherals were connected. Finally, the functionality of the system was tested using the created tests and test flights of ultralight aircraft with embedded device were performed.

Klíčová slova

záznamník letu, autonomní zařízení, minipočítač, vestavěné zařízení, Raspberry Pi, Raspbian, Python, periferie, GPS, mobilní aplikace, Android, Android Studio, Kotlin, Bluetooth

Keywords

flight recorder, standalone device, minicomputer, built-in device, Raspberry Pi, Raspbian, Python, peripherals, GPS, mobile application, Android, Android Studio, Kotlin, Bluetooth

Citace

KUDERA, Matěj. *Monitorování využití ultralehkých letadel*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Dr. Ing. Pavel Zemčík

Monitorování využití ultralehkých letadel

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka a uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Matěj Kudara
11. května 2021

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce prof. Dr. Ing. Pavlu Zemčíkovi za odbornou pomoc a cenné rady, které mi pomohly při řešení této práce.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 2 |
| 2 | Technologie vestavěných systémů | 4 |
| 2.1 | Platforma Raspberry Pi | 4 |
| 2.2 | Platforma Arduino | 11 |
| 2.3 | Možnosti komunikace s periferiemi | 16 |
| 2.4 | Družicové polohové systémy | 22 |
| 2.5 | Bezdrátové komunikační technologie | 25 |
| 3 | Provozování ultralehkých letounů | 30 |
| 3.1 | Popis ultralehkých letounů a způsobu záznamu letových časů | 30 |
| 3.2 | Podrobný popis provozu ultralehkých letounů | 31 |
| 3.3 | Určování letového času pomocí motohodin | 32 |
| 3.4 | Určování letového času stopkami | 32 |
| 3.5 | Zaznamenání letu sofistikovanými letovými přístroji | 33 |
| 4 | Zhodnocení provozu a specifikace zaznamenávacího systému | 34 |
| 4.1 | Zhodnocení aktuálních metod měření letového času | 34 |
| 4.2 | Rozhodnutí o realizaci systému | 35 |
| 4.3 | Podrobná specifikace systému | 35 |
| 5 | Tvorba zaznamenávacího systému a přístupové mobilní aplikace | 37 |
| 5.1 | Návrh struktury tvořeného systému | 37 |
| 5.2 | Výběr komponent pro vestavěné zaznamenávací zařízení | 38 |
| 5.3 | Zapojení hardwaru zaznamenávacího systému | 43 |
| 5.4 | Přízpůsobení napěťových úrovní zapalovacích pulzů motoru | 44 |
| 5.5 | Popis implementace softwaru vestavěného zařízení | 45 |
| 5.6 | Popis implementace softwaru mobilní aplikace | 47 |
| 5.7 | Experimentální montáž vestavěného zařízení | 48 |
| 5.8 | Testování systému | 50 |
| 5.9 | Návrhy na zlepšení | 55 |
| 6 | Závěr | 57 |
| | Literatura | 58 |
| A | Obsah přiloženého paměťového média | 61 |
| B | Podrobný popis implementace systému | 62 |

Kapitola 1

Úvod

Amatérské letectví bylo ještě před několika desítkami let vnímáno jako luxus, který si mohou dovolit jen bohatí lidé kvůli velkým pořizovacím cenám letadel a celkově velkým provozním nákladům, které jsou nezbytné pro létání a následný servis. Díky vývoji nových technologií a zdokonalování výrobních procesů se však cena potřebná pro vstup do amatérského letectví stabilně snižuje a celkově se tedy zvyšuje zájem o toto hobby. Zvyšující se počet pilotů zvyšuje i zájem o tvorbu nejrůznějších systémů, které by usnadnily pilotům a majitelům letadel starosti s provozem letadel. Tato práce se zabývá tvorbou systému, který by automaticky zaznamenával jednotlivé lety ultralehkých letounů a umožňoval by k nim přiřadit operujícího pilota.

Cílem práce bylo, aby tvořený systém šel následně použít na dvě hlavní věci, a to usnadnění vyplňování letového deníku jednotlivými piloty a pomoc při rozpočítávání letových nákladů, pokud je ultralehký letoun sdílen mezi více piloty. Letový deník slouží pilotům pro zápis letů, které uskutečnily, jednotlivé záznamy jsou pak důležité pro kontrolu letovými úřady a také umožňují pilotům určit, kolik letových hodin už mají nalétáno. Piloti si tato data běžně zaznamenávají jen okrajově a až následně je přepisují do svých letových deníků. Stává se tedy, že data nejsou vždy úplně přesná nebo dokonce mohou některá chybět. Systém bude záznamy pilotům tvořit automaticky a oni je pak mohou jen přepsat do svých deníků. Rozpočítávání letových záznamů je druhým velkým problémem, se kterým se amatérští piloti potýkají. Je běžné, že každý pilot nevládní svůj vlastní stroj, spíše ho vlastní jeden pilot, který ho následně pronajímá ostatním a nebo má více pilotů koupeno jeden dohromady do společného vlastnictví. S tím je spojena nutnost pravidelného sbírání letových záznamů od pilotů pověřenou osobou, která následně tato data zpracuje a určí, jaké náklady za provoz patří jednotlivým pilotům. Automatické pořizování letových záznamů tedy eliminuje nutnost sbírání letových záznamů od pilotů a také se omezí problémy s nepřesným rozpočítáním. Tuto práci jsem si vybral, protože mě zajímá tvorba vestavěných systémů a chtěl bych se v ní zdokonalit, mimo to mě také vždy fascinovala letadla.

Předpokladem tvorby systému je vybrat potřebné technologie, ze kterých půjde systém vytvořit a budou podporovat jeho lehkou instalaci a levnou pořizovací cenu. Vlastní systém musí umět vyhodnotit zda ultralehký letoun letí a následně uložit všechna potřebná data o proběhlém letu. Nakonec je nutné vytvořit přístupové zařízení, které umožní pilotům zaznamenané lety prohlížet a případně je editovat.

Tvorba systému je rozdělena do následujících kapitol. Po úvodu následuje kapitola 2, která popisuje technologie, které bylo nutné nastudovat k realizaci systému. Popsány jsou různé volby řídicího prvku jako Raspberry Pi a Arduino, způsoby připojení periférií k řídicímu prvku, družicové polohové systémy a nakonec bezdrátové přenosové technologie.

V kapitole 3 je probráno, jak se ultralehké letouny provozují a jak se aktuálně řeší problém zaznamenávání letů. V kapitole 4 jsou zhodnoceny aktuální principy provozu letadel a je vytvořena specifikace tvořeného systému. V kapitole 5 je uvedeno, jak byl výsledný systém zhotoven. Je zde popsán návrh systému, výběr použitých komponent, zapojení komponent a implementace softwaru vestavěného a mobilního zařízení. Po sekcích určených tvorbě systému se další sekce věnují experimentální montáži, testování systému a poslední sekce se věnuje návrhům na zlepšení, které jsou odvozeny z dosažených výsledků testování.

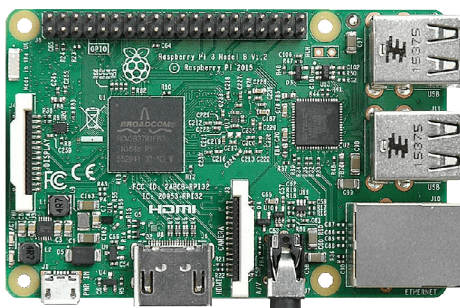
Kapitola 2

Technologie vestavěných systémů

Obsah této kapitoly popisuje technologie a poznatky, kterým je třeba porozumět, aby bylo možné vytvořit systém se všemi požadavky, které jsou na něj kladeny v zadání této práce. Je však potřeba zdůraznit, že následující popis poznatků nemá encyklopedický charakter a jsou zde popsány informace, nezbytně nutné k pochopení principů, na kterých je tato práce založena. Limit rozsahu práce totiž nedovoluje podrobnější popis.

2.1 Platforma Raspberry Pi

Raspberry Pi (viz obrázek 2.1) je jednodeskový počítač o velikosti platební karty, který byl vytvořen za účelem podpoření výuku informatiky ve školách a seznámit tak studenty s konceptem jak mohou počítače řídit různá fyzická zařízení [11]. Cena tohoto minipočítače se pohybuje v rozmezí od 150–1 500 Kč v závislosti na výkonu kupované verze.



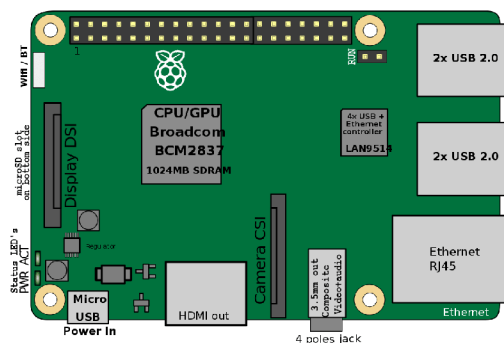
Obrázek 2.1: Raspberry Pi 3 (model B)¹

Raspberry Pi je koncipován jako malý stolní počítač, umožňující nainstalování operačního systému a obsahuje potřebné porty pro připojení periférií, které lze připojit k normálnímu stolnímu počítači. Mimo věci srovnatelné s všedním počítačem nabízí Raspberry Pi také GPIO piny (pin je zde jeden vývod mikropočítače), na které jde připojit periférie použitelné pro ovládání nejrůznějších zařízení. Vyvinuto bylo již několik generací, které se liší výkonem a zamýšleným použitím.

¹Převzato z: https://upload.wikimedia.org/wikipedia/commons/b/b4/Raspberry_Pi_3_Model_B.png

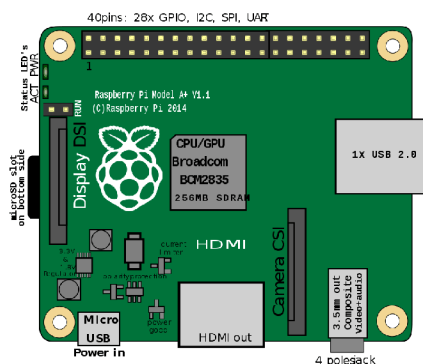
Modely Raspberry Pi

Raspberry Pi jde pořídit ve třech možných provedeních, které se liší výkonem a počtem nainstalovaných portů [19]. Uživatelům je tedy umožněno vybrat si konkrétní model, který nejvíce vyhovuje jejich aplikaci.



Obrázek 2.2: Raspberry Pi model B/B+²

Nejznámějším a nejvíc používaným modelem Raspberry Pi je model B/B+ (viz obrázek 2.2), který je koncipován jako plnohodnotná zmenšenina stolního počítače. Nabízí všechny potřebné konektory pro připojení nejdůležitějších periférií a umožňuje uživateli se zařízením pracovat jako s normálním počítačem. Oproti obyčejnému počítači je však ještě zařízení vybaveno GPIO piny, na které může uživatel připojit rozšiřující hardware pro tvorbu nejrůznějších projektů. Model B/B+ je nejvýkonnější a největší ze všech ostatních modelů Raspberry Pi. Mezi primární využití tohoto modelu patří například náhrada stolního pc, nejrůznějších serverů, vestavěných zařízení s potřebou většího výkonu a v neposlední řadě výukových platforem pro naučení se práce s vestavnými zařízeními.

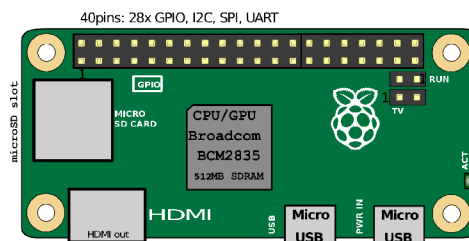


Obrázek 2.3: Raspberry Pi model A/A+³

²Převzato z: https://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/RaspberryPi_3B.svg/1280px-RaspberryPi_3B.svg.png

³Převzato z: https://upload.wikimedia.org/wikipedia/commons/thumb/b/b6/Drawing_of_Raspberry_Pi_model_A%2B_rev1.1.svg/1280px-Drawing_of_Raspberry_Pi_model_A%2B_rev1.1.svg.png

Raspberry Pi model A/A+ (viz obrázek 2.3) je velice podobný modelu B/B+, hlavní změnou je odstranění některých konektorů. Odstranění konektorů umožnilo, aby tento model měl menší profil a také zmenšení samotné desky. Primární využití je tedy směřováno na vestavěná zařízení, která potřebují být v rámci malá s větším výpočetním výkonem.

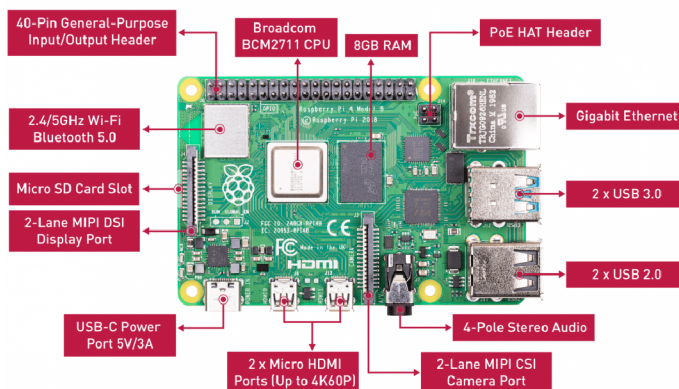


Obrázek 2.4: Raspberry Pi model Zero⁴

Posledním modelem z rodiny Raspberri Pi je model Zero (viz obrázek 2.4), který je nejmenší a nejméně výkonný. Obsahuje také nejmenší množství konektorů, které umožňují připojení jen nejzákladnějších periférií, které by mohlo vestavěné zařízení potřebovat. Důraz je kladen na to, aby tento model bylo možné použít v malých vestavěných zařízeních, kde je rozhodující velikost, váha a spotřeba.

Osazení desky Raspberry Pi a možnosti napájení

Jak již bylo zmíněno, platforma Raspberry pi je i z velké části konstruována tak, aby byla jednoduše použitelná ve výuce a naučila studenty jak fungují počítače a jak je mohou připojit k zařízení, které mají ovládat [8]. Z tohoto důvodu jsou veškeré komponenty na desce (viz obrázek 2.5) obnaženy a umožňují podrobnější pohled na všechny součásti, které takovéto zařízení musí obsahovat.



Obrázek 2.5: Raspberry Pi 4 přehled komponent⁵

⁴Převzato z: https://upload.wikimedia.org/wikipedia/commons/thumb/b/ba/Raspberry_Pi_Zero_-_Location_of_connectors_and_ICs.svg/1920px-Raspberry_Pi_Zero_-_Location_of_connectors_and_ICs.svg.png

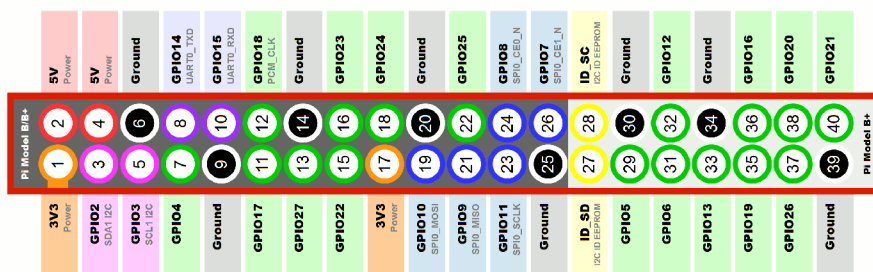
⁵Převzato z: https://blog.seeedstudio.com/wp-content/uploads/2020/05/44671590483676_pic_hd-1030x611.png

Předností Raspberry Pi je i možnost využití různých typů napájení [26], které umožňují vybrání neoptimálnějšího způsobu napájení v závislosti na zamýšleném použití zařízení.

- 5V DC skrze USB-C konektor
- 5V DC skrze napájecí GPIO piny
- PoE (neboli Power over Ethernet) umožňuje napájet zařízení přes Internetový port. Pro tuto funkci je však potřeba dokoupit Shield⁶, který tuto funkcionalitu poskytuje.

Konektor GPIO umožňuje připojení rozšiřujících periferií

GPIO konektor (viz obrázek 2.6) umožňuje k Raspberry Pi připojit nejrůznější hardware, který je možné naprogramovat pro dosažení požadované funkčnosti tvořeného zařízení [11]. Z pohledu vestavěných zařízení jsou GPIO piny nejzajímavější pro připojení hardwaru, který umožňuje pracovat se svým fyzickým okolím. Pinů je čtyřicet a jsou mezi sebou rozděleny podle nabízené funkčnosti.



Obrázek 2.6: Raspberry Pi přehled pinů⁷

Piny jsou v základu rozděleny na napájecí piny (5V, 3.3V a GND) a vstupně výstupní piny [12]. Vstupně výstupní piny lze použít k tomu, aby přijímaly či vysílaly logickou nulu (0V) nebo logickou jedničku (3.3V). A tím například posílaly hodnoty nebo přijímaly výsledky z nějakého zařízení. Některé ze vstupně výstupních pinů nabízí navíc rozšiřující funkčnost pro připojení složitějších periferií. Do rozšířené funkčnosti pinů se řadí:

- Pulzně šířková modulace (PWM),
- Synchronní sériová komunikace SPI,
- Komunikace po sériové sběrnici I2C,
- Asynchronní sériový přenos UART.

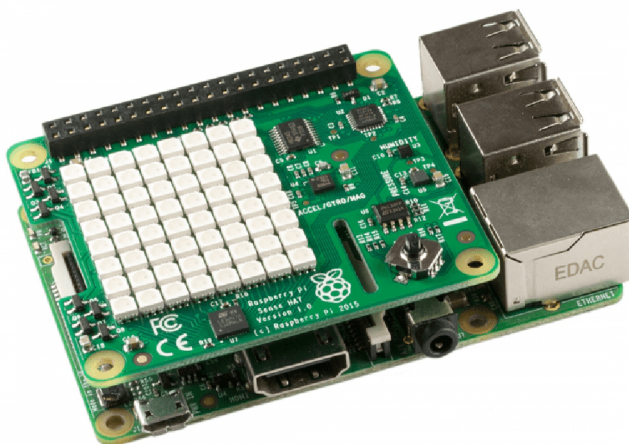
V neposlední řadě se GPIO piny používají i pro připojení takzvaných Shields nebo HATS, které rozšiřují Raspberry Pi o specifické funkčnosti.

⁶<https://rpishop.cz/zdroje/1040-raspberry-pi-poe-hat.html>

⁷Převzato z: <https://notenoughtech.com/wp-content/uploads/2016/04/Raspberry-Pi-GPIO-Layout-Model-B-Plus-rotated-2700x900.png>

Rozšiřující desky pro platformu Raspberry Pi

Pro jednodušší práci s nejrůznějšími hardwarovými perifériemi vznikly pro Raspberry Pi takzvané HATS nebo Shields [24] (viz obrázek 2.7). Což jsou rozšiřující desky, které už na sobě obsahují určitý rozšiřující hardware nebo poskytují rozhraní pro jednoduché připojení nějakého hardwaru. Pojmenování HATS získaly tyto desky hlavně díky tomu, že se pokládají přímo na Raspberri Pi, které pak vypadá jako kdyby na sobě mělo čepici. Rozšiřujících desek jde také dát více vrstev na sebe. Velký počet rozšiřujících desek je i díky povaze Raspberry Pi orientován na výukové účely.



Obrázek 2.7: Raspberry Pi s připojeným Shieldem Sence Hat⁸

Rozšiřující desky lze rozdělit do několika hlavních skupin:

- **Výukové desky:** slouží hlavně pro výuku základů z určité oblasti hardwaru,
- **Zvukové desky:** umožňují připojení pole mikrofونů nebo reproduktorů,
- **Obrazové desky:** obyčejné nebo dotykové displeje,
- **Rozšiřující konektorové desky:** tyto desky rozšiřují Raspberry Pi o piny nebo porty s rozšiřujícími funkcemi jako ADC, analogové vstupy, tranzistory, relé atd.,
- **Ostatní:** do této skupiny se řadí velice specifické desky, které byly pro Raspberri Pi vyrobeny (RTC, RS-485, Desky magnetického podle atd.).

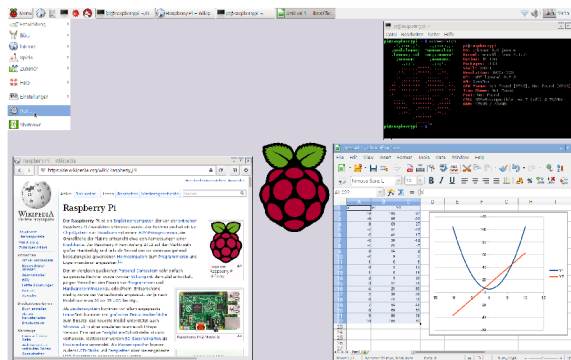
Operační systémy dostupné na Raspberry Pi

Nejnámější operační systém pro platformu Raspberry Pi je Raspbian⁹, který je založen na 32-bitovém Debianu a byl vytvořen speciálně pro tuto platformu [16]. Raspbian (viz obrázek 2.8) je navržený tak, aby šel používat jako běžný operační systém stolního počítače. Naproti operačním systémům stolních počítačů je však Raspbian podstatně jednodušší a neumožňuje spuštění všech programů, které jsou k dispozici na stolních počítačích. Značná

⁸Převzato z: https://grobotronics.com/images/detailed/13/dsc_0775-800x609_grobo.jpg

⁹<https://www.raspbian.org/>

část programů, které jsou na Raspbian k dispozici byla vytvořena převážně k podpoře učení práce s touto platformou.



Obrázek 2.8: Operační systém Raspbian¹⁰



Obrázek 2.9: Operační systém RetroPie¹¹

Kromě Raspbianu umožňuje Raspberry Pi nainstalování různých dalších operačních systémů, mezi kterými jsou i systémy specifické pro konkrétní použití (viz obrázek 2.9). Tím je umožněno využití Raspberry Pi plně pro konkrétní problém bez potřeby přetvářet standardní operační systém. Systémy použitelné pro Raspberry Pi je možné rozdělit do následujících kategorií:

- Operační systémy pro stolní PC,
- Operační systémy pro mobilní zařízení,
- Serverové operační systémy,
- Operační systémy pro vestavěné zařízení,
- Real-time operační systémy,
- Operační systém pro multimediální zařízení.

¹⁰Převzato z: https://upload.wikimedia.org/wikipedia/commons/f/fe/Raspbian-Desktop_Version_2015-09-25.png


¹¹Převzato z: <https://i.pinimg.com/originals/dd/04/1d/dd041da1fdbc0d47c38723f1cda2941d.jpg>

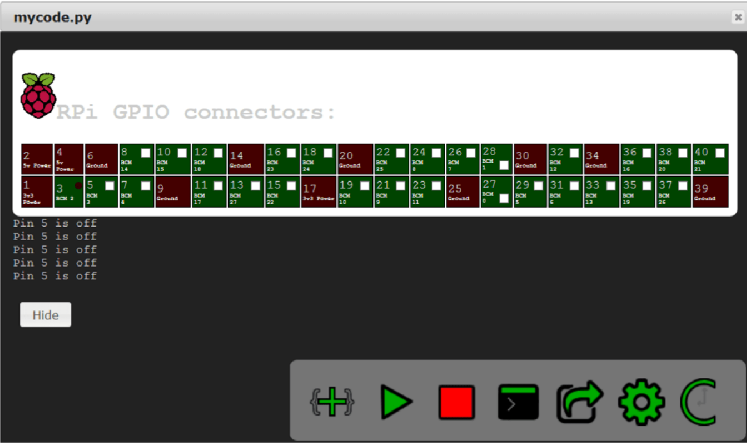
Možnosti programování na Raspberry pi

Raspberry Pi dokáže spouštět kód v celé řadě programovacích jazyků [11]. Mezi nejpoužívanější z nich se řadí Scratch a Python, protože v sobě mají implementovanou jednoduchou podporu pro ovládání GPIO konektoru a konektorů zařízení.

Scratch nabízí grafické programování a cílí tím převážně na školy. Umožňuje tak studentům jednoduše pochopit jak funguje programování a jak lze naprogramovat ovládání nejrůznějších zařízení, která lze k Raspberry Pi připojit.

Nejvhodnějším programovacím jazykem pro plnohodnotné programování je Python. Díky vysoké abstrakci tohoto jazyka je jednoduché udělat nejrůznější programy, které mají jen zpracovávat určitá data nebo ovládat připojené periferie (viz obrázek 2.10).

```
mycode.py 
1 # import modules
2 import RPi.GPIO as GPIO
3 import time
4
5 # setup pins
6 GPIO.setmode(GPIO.BCM)
7 GPIO.setup(3, GPIO.OUT)
8 GPIO.setup(5, GPIO.IN)
9
10 # loop 5 times
11 for i in range(5):
12
13     # flash output
14     GPIO.output(3, True)
15     time.sleep(0.5)
16     GPIO.output(3, False)
17     time.sleep(0.5)
18
19     # read input
20     if GPIO.input(5):
21         print("Pin 5 is on")
22     else:
23         print("Pin 5 is off")
24
```



Obrázek 2.10: Ovládání GPIO pomocí Pythonu¹²

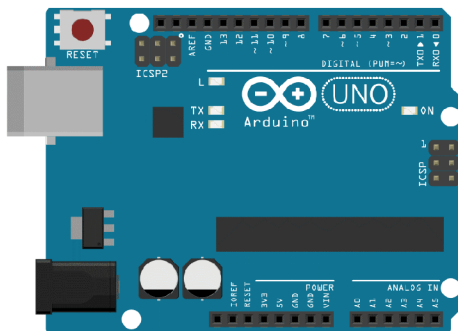
Jak již bylo zmíněno v základu je Raspberry Pi jen zmenšený stolní počítač. Není tedy problém použít jiné programovací jazyky. Je však nutné počítat s tím, že oficiálně podporovaným a doporučeným jazykem je Python. Díky tomu mohou nastat nečekané problémy při používání jiných programovacích jazyků, obzvláště při pracování s GPIO piny a porty zařízení. Často používané jsou například jazyky Java, JavaScript, C a C++¹³.

¹²Převzato z: <https://community.computingatschool.org.uk/system/images/2533/original/screenshot.png>

¹³<https://www.seeedstudio.com/blog/2020/02/25/which-raspberry-pi-programming-language-should-you-use-comparison-guide/>

2.2 Platforma Arduino

Arduino¹⁴ je jednodeskový mikropočítač (viz obrázek 2.11), který je koncipován pro výuku i implementaci projektů, které mají pracovat s fyzickým prostředím [7]. Veškeré součásti projektu Arduino jsou open-source, a proto existuje hned celá řada implementací desek tohoto mikropočítače. Také je možné poskládat celý mikropočítač ručně z jednotlivých dílčích součástí.



Obrázek 2.11: Mikropočítač Arduino¹⁵

Primárním cílem Arduina je ovládání periferií, které se k němu dají připojit. Proto obsahuje jen nezbytně nutné komponenty, které jsou potřebné k této činnosti. Díky této jednoduché činnosti je pak i jednoduchý vlastní cyklus použití Arduina, který se skládá z nahrání kódu, který následně provádí různé výpočty a čte či zapisuje hodnoty na vstupně výstupní piny.

Typy mikropočítačů Arduino

Arduino desky lze pořídit ve velkém množství různých typů (viz obrázek 2.12), které se od sebe liší svojí velikostí, výkonem a poskytovaným rozhraním [13]. Většina desek se od sebe liší i zamýšleným použitím, pro které byly konstruovány. Oficiálních desek Arduina je v současné době devatenáct a neoficiální několik stovek. Výhodou neoficiálních desek je, že jsou levnější, ale problémem může být, že ne všechny desky budou stoprocentně kompatibilní se vším co podporují oficiální desky.

Nejvýraznější typy oficiálních Arduino desek¹⁶:

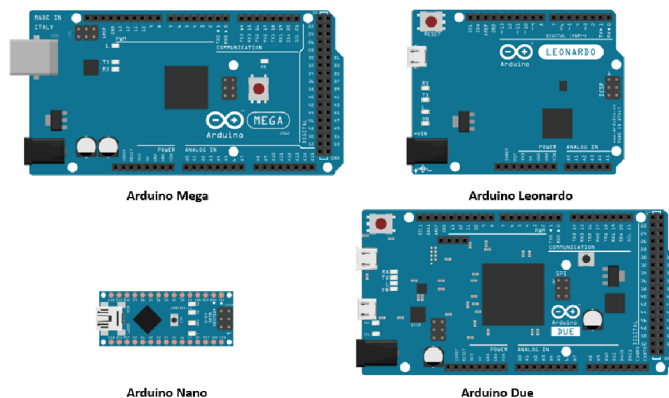
- **Arduino Uno:** standardní 8-bitové Arduino používané převážně pro výukové účely,
- **Arduino Due:** Arduino s 32-bitovým ARM procesorem,
- **Arduino Mega2560:** největší Arduino, obsahuje větší paměť a více portů pro připojení periferií,
- **Arduino Yún:** Arduino použitelné pro IoT zařízení, umožňuje nainstalování operačního systému který je založen na Linuxu, obsahuje zabudovaný Ethernet a Wifi,
- **Arduino Nano:** nejmenší Arduino, použitelný pro zařízení, která potřebují malou hmotnost, spotřebu a malý výkon,

¹⁴<https://www.arduino.cc/>

¹⁵Převzato z: https://www.pikpng.com/png1/m/248-2487756_arduinouno-arduino-port-clipart.png

¹⁶<https://www.arduino.cc/en/Main/Products>

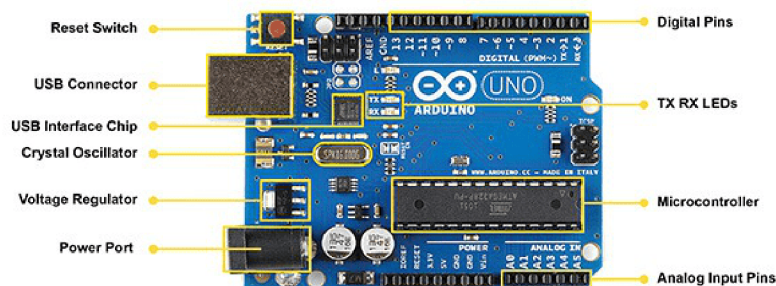
- **Arduino Pro:** návrh desky zaměřený na permanentní zasazení do tvořeného zařízení,
- **LilyPad Arduino:** návrh desky zaměřený na nositelná zařízení,
- **Arduino Esplora:** zaměřeno na tvorbu retro herních zařízení.



Obrázek 2.12: Přehled desek Arduino¹⁷

Komponenty umístěné na desce Arduino a možnosti napájení

Díky velkému počtu rozdílných druhů Arduino se často liší i konkrétní komponenty, které jsou na desce instalovány (viz obrázek 2.13) podle konkrétního využití Arduino [15]. Rozdílné bývají mikroprocesory, které jsou na desce instalovány (například 8-bitový AVR, 32-bitový ARM), počet vstupně výstupních pinů a rozšiřující hardware jako Ethernet nebo Wifi. Stejný vždy zůstává základ jako jsou desky vytvořené pro stejnou spolupráci mezi nimi a nebylo potřeba se učit nové postupy pro každé Arduino.



Obrázek 2.13: Arduino Uno přehled komponent¹⁸

Arduino také podporuje různé typy napájení¹⁹, aby byla co nejvíce usnadněna implementace do výsledného zařízení. Arduino lze napájet pomocí:

- 5 V USB konektor,

¹⁷Převzato z: <https://psalmchmart.store/wp-content/uploads/2020/03/boardss.png>

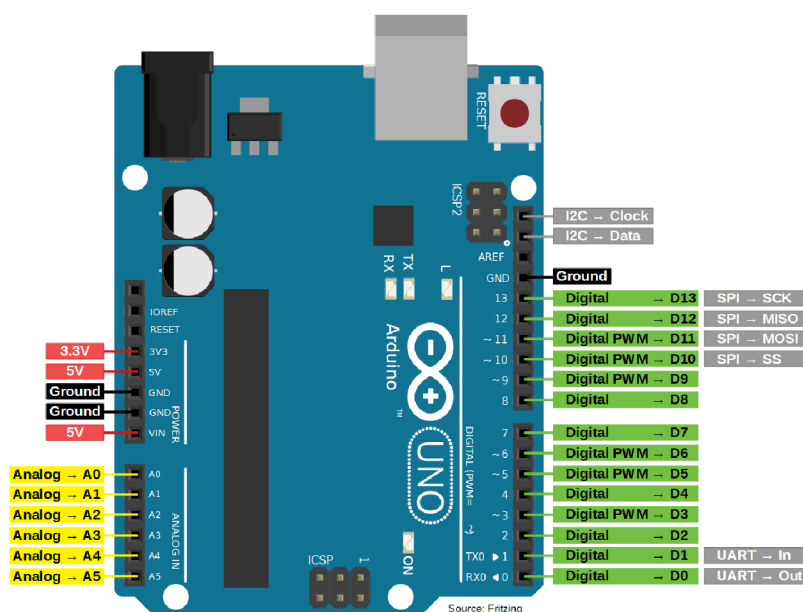
¹⁸Převzato z: <https://blog-c7ff.kxcdn.com/blog/wp-content/uploads/2016/10/board-01.jpg>

¹⁹<https://www.open-electronics.org/the-power-of-arduino-this-unknown/>

- Japan Jack 7 - 12 V,
- napájecí I/O piny 5 V,
- napájecí I/O piny 6 - 12 V,
- bateriový Shield.

Konektor vstupně výstupních pinů Arduina

Konektor vstupně výstupních pinů (viz obrázek 2.14) umožňuje k Arduinu připojení nejrůznějšího rozšiřujícího hardwaru [7]. Z pohledu vestavěných zařízení se jedná hlavně o zařízení, která mohou pracovat s okolním světem. Jak už bylo zmíněno, existuje mnoho druhů Arduina a to má za následek, že se liší i přesné piny, které každá deska nabízí.



Obrázek 2.14: Arduino Uno přehled pinů²⁰

Obecně však Arduina obsahují základní druhy pinů, které jsou mezi deskami stejné a liší se hlavně svým počtem a rozšiřujícími funkcemi. Základními druhy jsou:

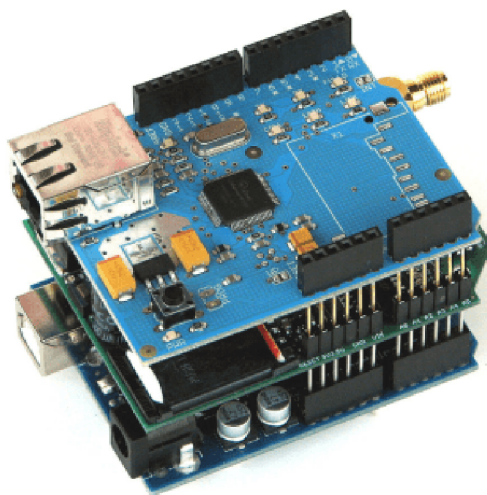
- **Napájecí piny:** je možné je použít dvěma způsoby. Napájení Arduina nebo pokud je Arduino napájeno jiným možným způsobem, tak je možné je použít pro napájení připojených součástí (napájení je možné pro 3.3 a 5 V připojená zařízení).
- **Analogové vstupní piny:** používané pro čtení hodnot z analogových senzorů. Analogová hodnota je čtena pomocí A/D převodníků. Pokud pro tvořený projekt nejsou tyto piny potřeba, je možné je přepnout do funkčnosti, kterou mají digitální piny.
- **Digitální piny:** mohou být nastaveny jako vstupní nebo výstupní. Komunikace probíhá pomocí logických nul a jedniček. Některé digitální piny mají i rozšiřující funkci, pro kterou je lze použít. Rozšiřující funkčnost může být například použití PWM nebo komunikace UART, SPI a I2C.

²⁰Převzato z: <https://diyi0t.com/wp-content/uploads/2019/08/Arduino-Uno-Pinout-1.png>

Vstupně výstupní piny slouží také pro připojení nejrůznějších rozšiřujících desek neboli Shields.

Rozšiřující desky pro platformu Arduino

Rozšiřující desky (nebo také Shields) jsou určeny k tomu, aby bylo možné jednoduše rozšířit funkčnost Arduina ve specifickém odvětví [25]. Shields se připojují na vstupně výstupní piny Arduina (viz obrázek 2.15) a rozšiřují ho o konkrétní hardware nebo poskytují rozhraní pro jednoduché připojení konkrétního rozšiřujícího hardwaru. V závislosti na povaze konkrétních rozšiřujících desek je u některých umožněno jejich propojení a tím lze rozšířit funkčnost ve více odvětvích.



Obrázek 2.15: Arduino s rozšiřujícími Shields²¹

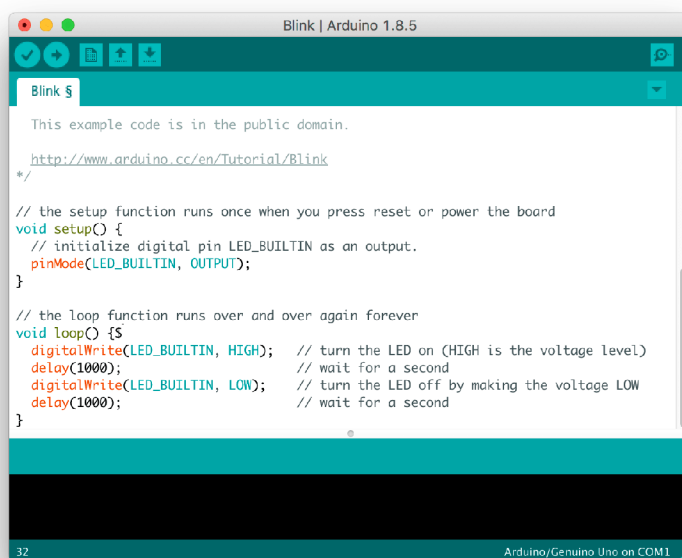
Typy rozšiřujících desek podle jejich využití:

- **Výukové:** slouží pro výuku používání komponent pro určité odvětví vestavěných zařízení.
- **Prototypovací:** prototypovací desky nepřinášejí funkčnost. Zlepšují již nabízené funkčnosti (například lepší vyvedení pinů, SD karta pro větší úložiště nebo baterka pro napájení).
- **Komunikační:** rozšiřují funkčnost Arduina o některý typ komunikace (například Ethernet, Wifi, GPS, Bluetooth, GSM).
- **Zvukové:** umožňují napojení reproduktorů pro přehrávání zvuků nebo analyzátorů pro analýzu zachycených zvuků.
- **Obrazové:** umožňují připojení displeje nebo kamery.
- **Ostatní desky:** desky určené pro některé specifické úkoly (například ovládání motorů, RTC nebo relé).

²¹Převzato z: <https://aprendiendoarduino.files.wordpress.com/2019/06/shields.png?w=364>

Způsob tvorby programů pro Arduino

Program pro Arduino může být napsaný v kterémkoli programovacím jazyce, který lze přeložit kompilátorem do binární podoby pro 8-bitové AVR nebo 32-bitové ARM procesory [15]. Nejpoužívanějším způsobem jak naprogramovat Arduino je pomocí Arduino IDE (viz obrázek 2.16). Toto vývojové prostředí bylo vytvořeno přímo tvůrci Arduina, kteří chtěli aby šli všechny jejich desky programovat stejným jednoduchým způsobem.



Obrázek 2.16: Arduino IDE²²

Programy v Arduino IDE jsou psány v programovacím jazyce Wiring, který je svým zápisem dost podobný jazykům C nebo C++. Tento programovací jazyk byl vytvořený pro jednoduché programování hardwaru bez potřeby porozumění podrobnostem, které by jinak byly potřeba k této činnosti.

Programu, který byl napsán v Arduino IDE se říká Sketch. Každý Sketch musí vždy obsahovat dvě povinné části:

- **Setup():** tato funkce je spuštěna vždy při startu a restartu Arduina, slouží pro inicializaci všech věcí, které jsou potřeba nastavit před spuštěním vykonávání hlavního kódu programu. Nastavují se zde například proměnné, piny nebo knihovny.
- **Loop():** funkce s hlavním kódem programu, která je vykonávána pořád dokola, dokud není Arduino vypnuto nebo restartováno.

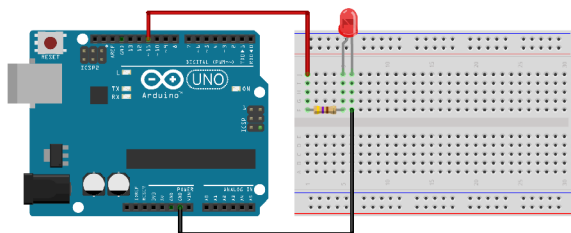
Nově je také k dispozici vývojové prostředí Arduino PRO, které nabízí rozšířenou funkčnost srovnatelnou s velkými vývojovými prostředími pro jiné programovací jazyky. Oproti obyčejnému Arduino IDE je k dispozici například debugger, git nebo možnost modulárního vývoje.

²²Převzato z: https://upload.wikimedia.org/wikipedia/commons/a/a1/Arduino_IDE_-_Blink.png

2.3 Možnosti komunikace s periferiemi

Nejpodstatnější funkcí vestavěných zařízení je možnost připojování rozšiřujícího hardwaru, který je potřebný k tomu, aby zařízení mohlo plnit svoji cílovou funkčnost [23]. Složitost samotného připojení se může výrazně lišit podle hardwaru, který je potřeba připojit.

Nejjednodušším způsobem jak připojit rozšiřující hardware k mikropočítači vestavěného zařízení je pomocí obecného vstupně výstupního pinu (viz obrázek 2.17). Tento způsob umožňuje s periferií komunikovat jen pomocí dvou stavů, logická nula nebo jednička. Jak lze pozorovat, možnosti tohoto způsobu jsou značně limitovány a použití je tedy možné například u ovládání jednoduchých elektronických obvodů. Příklad výstupního obvodu může být ovládání LED diody. Příklad výstupu pak získávání informace, zda bylo zmáčknuto některé tlačítko.



Obrázek 2.17: Ovládání LED diody²³

Pro reálná vestavěná zařízení je však potřeba komunikovat s periferiemi složitějšími způsoby než jen zasíláním dvou stavů. Je běžné zobrazovat data na displeji uživateli nebo číst data ze složitějšího senzoru, jako je například GPS. Pro tyto účely je potřeba posílat velké množství logických nul a jedniček (bitů), které jsou uspořádány takovým způsobem, aby nesly složitější informaci pro přijímací zařízení.

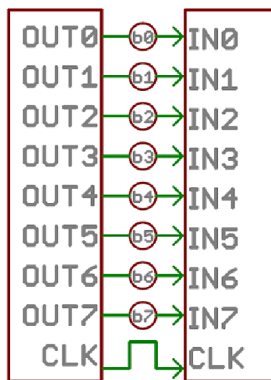
Paralelní komunikace

Jedním ze způsobů jak posílat komplexnější data mezi mikropočítačem a periferií je paralelní komunikace [10]. Paralelní komunikace posílá bity po několika kabelech naráz v takzvaných vlnách (viz obrázek 2.18), čas odeslání jednotlivých vln je určován podle řídicího hodinového signálu. Počet jednotlivých linek je běžně mocninou dvojky.

Výhoda paralelní komunikace je v rychlosti posílání dat, která je určena rychlostí interních hodin a počtem přenosových linek. Tento typ komunikace má však také značné nevýhody, a to například potřeba spojení někdy i celkem značných počtů kabelů a možný výskyt přerůků mezi jednotlivými linkami. Kvůli své rychlosti se paralelní komunikace používá primárně pro implementaci sběrnic, ve kterých je potřeba uplatnit vysoké přenosové rychlosti.

²³Převzato z: <https://raw.githubusercontent.com/BasOnTech/Arduino-Beginners-NL/master/E02-led-knipperen-breadboard/led-knipperen-breadboard.png>

²⁴Převzato z: <https://cdn.sparkfun.com/assets/c/a/c/3/a/50e1cca6ce395fbc27000000.png>



Obrázek 2.18: Osmibitová paralelní komunikace²⁴

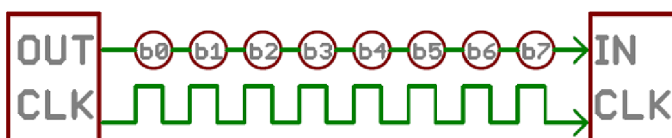
Sériová komunikace

Oproti paralelní komunikaci sériová komunikace používá jen jediný kabel pro přenos bitů, jednotlivé bity je tedy potřeba postupně posílat po jednom kabelovém spojení [20]. Přenos bitů po kabelu je řízen stejně jako u paralelní komunikace řídicím hodinovým signálem (viz obrázek 2.19).

Výhodou sériové komunikace je potřeba malého počtu kabelů pro přenos informací (může být použit i jediný kabel) a tím je potlačen výskyt přerůků v přenosu. Jako nevýhoda by mohla být považována rychlost přenosu bitů, většinou však není problém zvýšit rychlost řídicích hodin a tím získat rozumné přenosové rychlosti.

Pro vestavěné systémy je sériová komunikace nejpoužívanějším způsobem jak komunikovat s připojenými periferiemi. Sériová komunikace je totiž levná na implementaci a pro přenos dat mezi periferiemi a mikropočítačem jsou její přenosové rychlosti více než dostatečné. Asi největší výhodou sériové komunikace pro využití ve vestavěných systémech je malý počet kabelů, který je potřeba pro přenos informací. Mikropočítače mají většinou značně omezený počet pinů, a proto je potřeba piny šetřit, aby šlo všechny potřebné periferie k mikropočítači připojit.

Sériovou komunikaci jde najít ve značném počtu zařízení, nejznámějšími příklady může být například USB nebo Ethernet rozhraní. Ve vestavěných zařízeních je sériová komunikace používána v rozhraních jako UART, SPI a I2C.



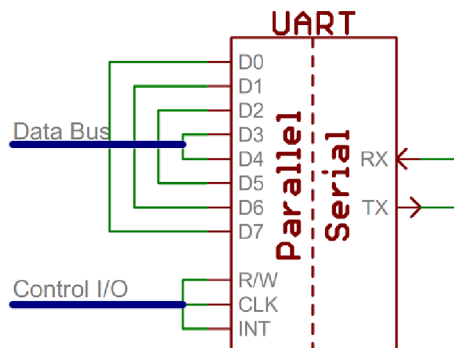
Obrázek 2.19: Osmibitová sériová komunikace²⁵

UART

UART (Universal asynchronous receiver/transmitter) [17] je hardwarový obvod, který pro mikropočítač implementuje rozhraní pro sériovou komunikaci (viz obrázek 2.20). Obecně

²⁵Převzato z: <https://cdn.sparkfun.com/assets/e/5/4/2/a/50e1ccf1ce395f962b000000.png>

slouží UART jako prostředník mezi paralelní sběrnicí, kterou přijímá/odesílá data mikro počítač a sériovým komunikačním rozhraním, které je potřebné pro připojení rozšiřujících periférií. Pro odesílání a přijímání dat přes sériové rozhraní jsou používány dva piny TX(transmitter) a RX(receiver).



Obrázek 2.20: Rozhraní UART²⁶

Pro přijímání a odesílání dat stačí UARTu jen dvě spojení, protože je to takzvané asynchronní sériové komunikační rozhraní. Jednodušeji lze říct, že řídicí hodinový signál mezi dvěma komunikujícími zařízeními není sdílený, jak by to bylo u synchronní komunikace. Tento způsob je vhodný pro minimalizování potřebných spojení, které by byly potřeba vytvořit mezi periférií a mikro počítačem. Nevýhoda je ve výsledném přenosovém protokolu, který musí být složitější, aby dokázal posílat spolehlivě data i přesto, že konkrétní doba přenosu není přesně definována společným hodinovým signálem.

Pro správný přenos dat po asynchronním sériovém rozhraní je potřeba dodržovat dva základní mechanismy, které eliminují potřebu pro sdílený řídicí hodinový signál.

- **Modulační rychlost (Baud rate):** modulační rychlost určuje jak rychle jsou přes přenosové rozhraní posílána data. Rychlost je udávána v bitech za sekundu. Tento údaj je potřeba nastavit stejně na obou zařízeních, která spolu komunikují, jinak nebude stejná perioda, za kterou se budou obě zařízení snažit číst/zapisovat data na přenosové kabely a všechna přenesená data by pak byla jen smetí.
- **Datový rámec:** všechna data, která jsou potřeba přes UART přenést se před odesláním dělí na takzvané datové rámce (viz obrázek 2.21). Každý vytvořený rámec je pak složen ze synchronizačních bitů, přenášených dat a dle implementace se může vyskytovat i paritní bit. V každém rámci je typicky jeden startovací bit, který udává přijímacímu zařízení, že začíná nový datový rámec, který obsahuje nějaká data. Následují datové bity, kterých je běžně osm, ale mohou se lišit dle implementace. Po datových bitech může následovat paritní bit, ze kterého lze zjistit jestli se data při přenosu nepoškodila a nakonec je rámec ukončen jedním či dvěma končícími bity, které příjemci říkají, že nastal konec aktuálně přenášeného rámce. Jak lze tedy pozorovat, absence synchronizačních hodin způsobuje potřebu posílat větší počet bitů, než jen samotná data, aby se zajistil správný přenos posílaných dat.

Pokud mikro počítač neobsahuje hardwarovou implementaci UARTu je možné použít takzvaný **Softwarový UART**. Softwarový UART funguje obdobně jako hardwarový UART,

²⁶Převzato z: <https://cdn.sparkfun.com/assets/d/1/f/5/b/50e1cf30ce395fb227000000.png>

²⁷Převzato z: <https://cdn.sparkfun.com/assets/f/9/c/0/2/50d2066fce395fc43b000000.png>



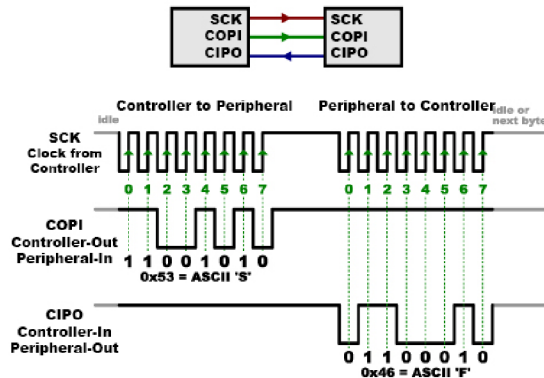
Obrázek 2.21: Rámec UART²⁷

jen přenos dat je řízen programem v mikropočítači, který limituje funkčnost hardwarového UARTu.

SPI

SPI (Serial Peripheral Interface) [22] je sběrnice používaná pro připojení malých periférií k mikropočítači vestavěného zařízení. Rozhraní je synchronní takže oproti datovým spojům používá i separátní spoj pro šíření řídicího hodinového signálu. Díky sdílenému hodinovému signálu je přesně dáno, kdy se mají bity číst/zapisovat a není potřeba uplatnění žádných dalších podpůrných mechanismů pro zajištění správného přenosu dat.

SPI funguje na principu ovladače a periferie, což znamená že jedno zařízení v SPI komunikaci se chová jako hlavní a ovládá všechny připojené periferie. Ovladač zajišťuje rozesílání řídicího hodinového signálu a určuje tak periferiím přesnou dobu, kdy se má komunikovat. Pro komunikaci informací se používají piny s názvy COPI (Controller Out Peripheral In) a CIPO (Controller In Peripheral Out). Výměna dat probíhá oběma směry naráz při jednom tiku hodinového signálu (viz obrázek 2.22). Interně je v zařízeních SPI výměna dat mezi zařízeními implementována pomocí jednoduchého posuvného registru. Což má za výsledek, že SPI rozhraní jsou jednoduché a levné na výrobu.



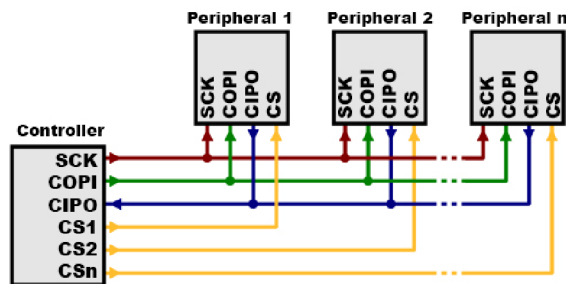
Obrázek 2.22: Komunikace dat mezi SPI rozhraními²⁸

Pokud je k SPI rozhraní připojeno více periférií, pak je potřeba zavést spoje označené CS (Chip Select), které slouží k tomu, aby si SPI ovladač mohl vybrat, se kterou periférií si chce aktuálně vyměňovat data a ostatní se do komunikace nezapojili. Platí, že pro každou periférii, která se má připojit k SPI ovladači je potřeba na mikropočítači vyhradit jeden volný pin, který bude použitý jako povolovací signál pro připojovanou periférii.

Nevýhody SPI:

²⁸Převzato z: https://cdn.sparkfun.com/assets/learn_tutorials/1/6/BasicSPI_Updated.jpg

²⁹Převzato z: https://cdn.sparkfun.com/assets/learn_tutorials/1/6/MultipleCS_Updated.jpg



Obrázek 2.23: SPI připojení několika periférií²⁹

- komunikace mezi ovladačem a periferií musí být od začátku dobře definovaná. Komunikaci vždy řídí ovladač a je tedy problematické když by chtěla periferie poslat něco urgentního ovladači. Tento protokol počítá jen s tím, že periferie může odeslat svoje data jen když si o ně zažádá ovladač a při této výměně rovnou dostane nová data od ovladače.
- problém je také s tím, že veškerá komunikace je směřována přes kontrolér, není tedy možné, aby si vyměnily data dvě periferie na přímo.
- obecně je také problém s tím, že SPI zabírá znatelný počet pinů na mikropočítači, a to hlavně když je připojeno více periférií současně (viz obrázek 2.23).

Nejlepší je SPI používat při připojování jednoduchých senzorů k mikropočítači. Při tomto zapojení není potřeba posílat data senzorům a jen se využívá toho, že si může mikropočítač přes jedno rozhraní zvolit, ze kterého senzoru si chce přečíst aktuální hodnotu.

I2C

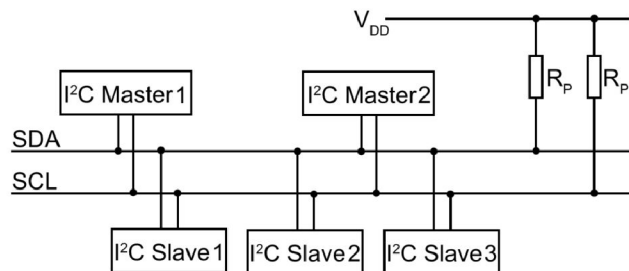
I2C je synchronní sériová sběrnice využívána pro připojení nízkorychlostních periférií k mikropočítači vestavěného zařízení [21]. Pro přenos dat a synchronizaci hodin stačí I2C jen dva spoje a protokol přenosu dat podporuje připojení až 1008 periférií. Oproti SPI podporuje I2C připojení více ovladačů (také označováno jako masters) na sběrnici a je tak možné, aby přenos dat zahajovaly postupně jiná zařízení.

I2C používá spoje označené jako SCL s SDA, kde SCL je spoj pro šíření hodinového signálu a SDA spoj pro přeposílání dat (viz obrázek 2.24). Řídící hodinový signál generuje vždy aktuálně komunikující ovladač, je však umožněno, aby komunikující periferie zpomalila hodinový signál když ještě nemá nachystaná data pro odeslání ovladači. Díky tomu, že jsou data posílána jen po jednom spoji, je celá komunikace poloduplexní a nejde tedy vyměňovat informace mezi periférií a ovladačem naráz jako tomu je i SPI.

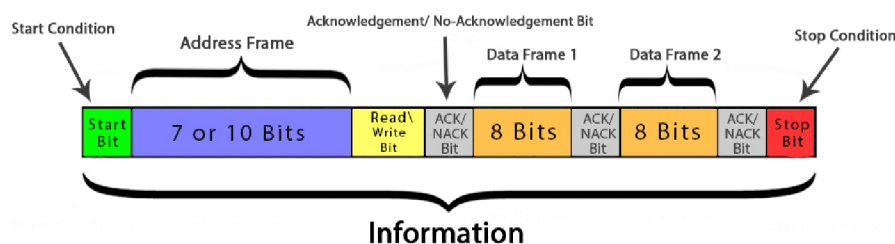
Z důvodu, že je možné připojit více ovladačů na jednu sběrnici, jsou vysílací obvody jednotlivých komunikujících zařízení implementovány jako takzvaný otevřený kolektor (open drain). Otevřený kolektor umožňuje zařízením nastavovat hodnotu na sběrnici jen na logickou nulu, nikdy ne na logickou jedničku. Tento mechanismus je potřebný, aby nemohl vzniknout zkrat, kdyby se snažily dva ovladače vysílat naráz. Změna na logickou jedničku je zajištěna pull-up rezistory, které jsou na sběrnici připojeny.

³⁰Převzato z: <https://i.stack.imgur.com/LcGZ8.jpg>

³¹Převzato z: <https://dcbestore.com/wp-content/uploads/2020/02/I2C-frame.jpg>



Obrázek 2.24: Zapojení I2C³⁰



Obrázek 2.25: Základní datový rámec I2C³¹

Pro zajištění kompletní komunikace, kterou I2C nabízí je celá komunikace řízena komplexním protokolem, který data rozděljuje do rámců (viz obrázek 2.25), která jsou následně posílána přes sběrnici a jsou řízena signalizačními bity. Celá komunikace se skládá z následujících dílů:

- **Startovací podmínka:** slouží ovladači pro indikaci ostatním, že chce začít komunikovat po sběrnici. Provádí se nastavení SCL na logickou jedničku a SDA na logickou jedničku. Pokud by se snažily naráz začít komunikovat dva ovladače, tak vyhraje ten, který zvládne splnit startovací podmínku jako první.
- **Adresový rámec:** po indikaci zahájení komunikace, vždy následuje adresový rámec, který udává adresu periferie, se kterou si bude ovladač vyměňovat data. Následuje read/write bit, který udává směr komunikace, zda budou data na sběrnici posílána ovladačem nebo periferií. Po každém rámcí (adresovém i datovém) následuje NACK/ACK bit, který dává možnost přijímacímu zařízení sdělit odeslateli jestli mu data dorazila.
- **Datový rámec:** datové rámce obsahují samotná data, která si zařízení vyměňují. V jednom navázaném spojení může být těchto rámců jakýkoli počet.
- **Ukončující podmínka:** po odeslání nebo přijetí všech dat ovladač vygeneruje ukončující podmínku, která rozpojí aktuální komunikaci a umožní navázání komunikace dalšími ovladači.

I2C je dobré používat v případech, kdy je potřeba připojit velké množství periférií k mikroprocesoru a celá implementace je limitována počtem využitelných spojů. A také v případech, kdy musí komunikaci na jedné sběrnici řídit více ovladačů. Z hlediska rychlosti přenosu a ceny implementace je I2C někde mezi SPI a UART.

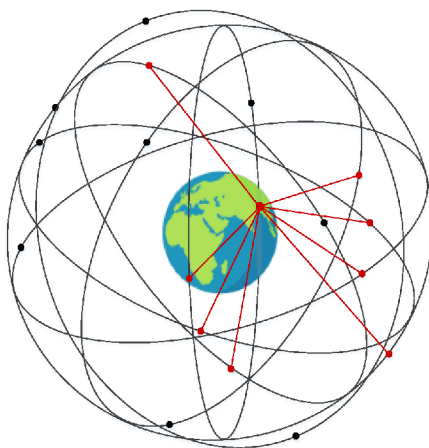
2.4 Družicové polohové systémy

Družicový polohový systém (nebo také Satnav) je služba, která umožňuje určit polohu (zeměpisnou šířku, délku a nadmořskou výšku) zařízení pomocí družic [27]. Pro určení polohy stačí malý rádiový přijímač, který přijímá údaje odesílané z družic a pomocí těchto údajů vypočítá přesnou polohu zařízení. Přesnost polohy se pro běžné aplikace pohybuje v jednotkách metrů, pro vědecké či vojenské účely je možné získat přesnost v jednotkách centimetrů či milimetrů. Hlavní způsoby využití těchto systémů jsou určení polohy, navigace, sledování zařízení vybaveného rádiovým přijímačem nebo synchronizace času.

Existují dva druhy družicových polohových systémů, globální (GNSS), který umožňuje získání polohy kdekoli na planetě zemi a lokální, který byl vyvinutý pro získávání polohy jen v konkrétní oblasti. U globálních systémů je pokrytí zajištěno 18 - 30 satelity na střední oběžné dráze země (takzvané medium earth orbit - MEO), která je ve výšce kolem 20 000 km nad mořem. Systémy Satnav nejsou spojeny s telefonní či internetovou sítí, údaje o poloze a času je tedy možné získat zcela nezávisle na těchto sítích.

Princip fungování globálních polohových systémů

Nejpoužívanější metodou pro určení pozice uživatelského zařízení je takzvaná **dálkoměrná metoda**, která ke svému fungování využívá měření vzdálenosti mezi uživatelským zařízením a viditelnými družicemi (viz obrázek 2.26).

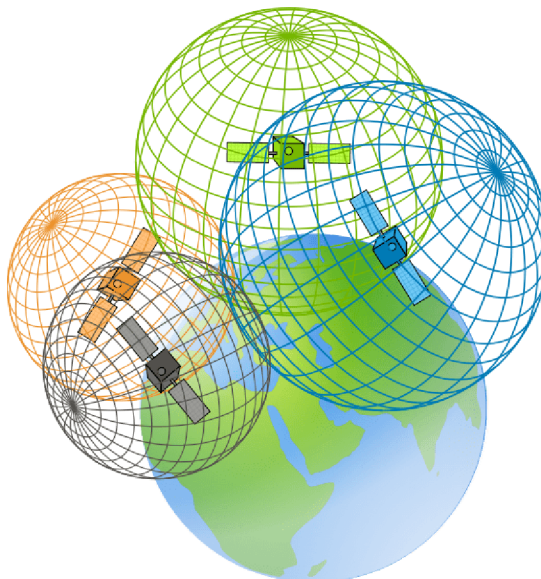


Obrázek 2.26: Vzdálenost mezi družicemi a uživatelským zařízením³²

Z pohledu implementace, tento systém funguje tak, že družice nepřetržitě vysílají zprávy, ve kterých je časové razítko odeslání zprávy a navigační parametry družice. Výpočet vzdálenosti mezi konkrétní družicí a příjemcem by se tedy měl vypočítat jako rozdíl času, kdy byla zpráva odeslána ze satelitu a kdy dorazila do přijímacího zařízení. Je však téměř nemožné zajistit dokonalou synchronizaci mezi hodinami satelitu a hodinami přijímacího zařízení. Z tohoto důvodu je potřeba pro určení polohy používat vždy zprávy z nejméně čtyř satelitů, s těmito informacemi je možné zajistit synchronizaci času na přijímacím zařízení a získat tak dobu jak dlouho trvalo signálu urazit vzdálenost mezi satelitem a přijímacím zařízením. Je však nezbytně nutné, aby byl synchronní čas mezi všemi satelity používaného

³²Převzato z: https://www.mdpi.com/applsci/applsci-10-00500/article_deploy/html/images/applsci-10-00500-g001.png

pozičního systému co nejvíce synchronní, jinak budou výpočty nepřesné. Časové razítko je potřebné i pro výpočet pozice družice v čase, kdy zprávu odeslala. K tomu se používají navigační parametry družice, ze kterých jde za pomoci časového razítka určit pozici družice v kartézském souřadnicovém systému, který má střed ve středu země.



Obrázek 2.27: Určení pozice pomocí satelitů³³

Získáním pozic družic, v kartézském systému souřadnic, je pak možné provést výpočet bodu, ve kterém se vzdálenosti ze všech dostupných družic protínají (viz obrázek 2.27). Výsledný bod je pak pozice, ve které se uživatel nachází vzhledem k satelitům. Výsledný bod je pak potřeba ještě převést na systém souřadnic (zeměpisná šířka, délka a nadmořskou výška), který určí pozici uživatele na mapě.

Rozlišování zpráv posílaných pozičními systémy

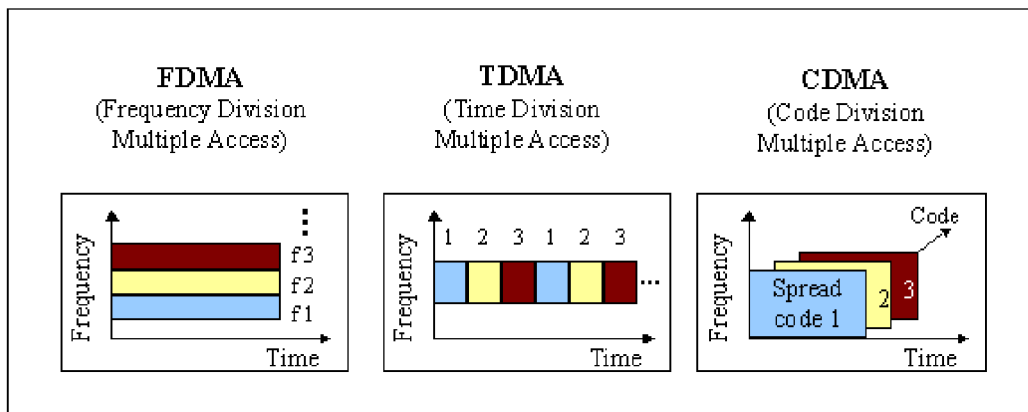
Systémů pro určení polohy uživatele existuje hned několik a každý systém má mezi 20 - 30 satelity. Je tedy potřeba zajistit, aby se informace, které satelity posílají, navzájem nerušily a byly od sebe navzájem rozlišitelné.

Pro rozlišení jednotlivých systémů se používá rozdílná frekvence, na které jsou zprávy od satelitů vysílány. V rámci jednoho systému je pak potřeba odlišit vysílání jednotlivých satelitů. Podle toho pak dokáže přijímací zařízení určit kolik satelitů s ním komunikuje a které zprávy patří kterému. Vysílání jednotlivých satelitů lze rozlišit následujícími metodami (viz obrázek 2.28):

- **CDMA (Code Division Multiple Access):** při této metodě vysílají družice zprávy na stejné frekvenci a rozlišují se tak, že je zakódují pomocí svého kódu. Při příjmu signálu uživatelským zařízením je pak využívání korelace, pomocí, které jde vyfiltrovat signál od jedné družice a signály od ostatních družic se poté jeví jako šum v pozadí.

³³Převzato z: <https://freesvg.org/img/GPS-3D-trilateration.png>

³⁴Převzato z: <http://www.itu.int/osg/spuold/ni/images/codedivision.gif>



Obrázek 2.28: Metody přístupu k vysílacímu médium³⁴

- **FDMA (Frequency Division Multiple Access):** tato metoda přiděluje každému satelitu jinou frekvenci v používaném frekvenčním pásmu. Každý satelit tedy vysílá svoje zprávy na jiné frekvenci.
- **TDMA (Time Division Multiple Access):** při této metodě posílají družice zprávy na stejné frekvenci a jednotlivé družice se od sebe liší, tím že každá komunikuje v jeden konkrétní čas, když žádná z ostatních družic nekomunikuje. Všechny družice se takto postupně střídají pořad dokola.

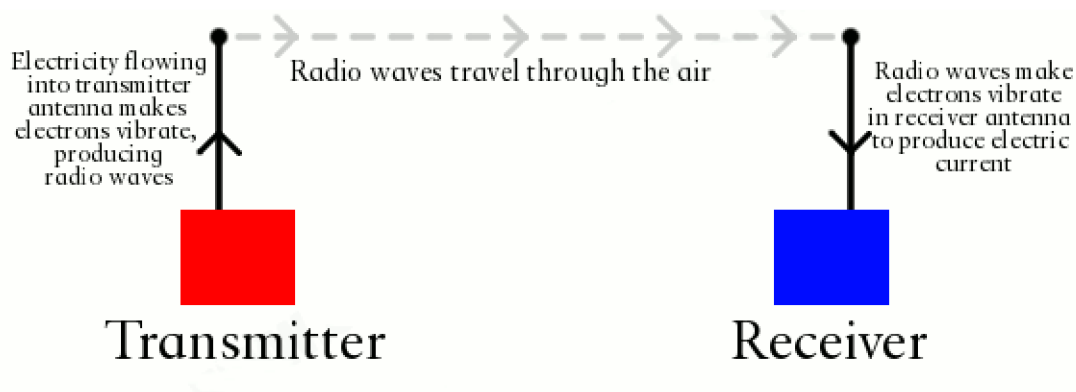
Přehled aktuálně provozovaných pozičních systémů

K dnešnímu datu jsou v provozu následující globální poziční systémy:

- **GPS (Global Positioning System):** nejstarší široce používaný poziční systém, který provozují Spojené Státy Americké. Systém používá 30 satelitů na orbitě ve výšce 20 180 km nad mořem. Pro odlišení vysílání jednotlivých satelitů se používá metoda CDMA. Udávaná přesnost systému je 0.3 - 5m.
- **GLONASS (Global'naya Navigatsionnaya Sputnikovaya Sistema):** poziční systém provozovaný Ruskou Federací. Používá 24 satelitů na orbitě ve výšce 19 130 km nad mořem. Pro odlišení vysílání jednotlivých satelitů se používá metoda FDMA i CDMA. Udávaná přesnost systému je 2 - 4m.
- **Galileo:** nejnovější poziční systém provozovaný Evropskou Unií. Systém využívá 24 satelitů na orbitě ve výšce 23 222 km nad mořem a zprávy jednotlivých satelitů jsou od sebe odděleny metodou CDMA. Udávaná přesnost je 1m pro civilní účely a 0.01m pro vojenské a vědecké účely.
- **BeiDou (BeiDou Navigation Satellite System):** poziční systém vlastněný Čínskou lidovou republikou. V prvních verzích byl koncipovaný jen jako regionální systém pro Asii, postupem času se rozšířil počet satelitů a dnes je použitelný globálně. Aktuálně je v provozu 28 satelitů z toho 24 na MEO, 3 na IGSO a 1 na GSO. Systém používá CDMA metodu pro odlišení zpráv jednotlivých satelitů. Udávaná přesnost určení pozice je 3.6m pro civilní účely a 0.1m pro vojenské a výzkumné účely.

2.5 Bezdrátové komunikační technologie

Bezdrátové technologie byly navrženy pro eliminaci nepraktičnosti kabelových spojů v aplikacích, kde jde hlavně o pohodlnost spojení a ne o maximální rychlost přenosu [14]. Bezdrátový přenos dat je možný zajistit hned přes několik typů nosného média. Přenos dat je možné zajistit opticky, radiově nebo sonicky. Dnes nejpoužívanějším médiem jsou radiové vlny, které se používají kvůli své dobré prostupnosti signálu jinými objekty a dlouhým dosahem spojení. Radiový přenos je realizován pomocí vysílačů a přijímačů, mezi kterými jsou radiové vlny posílány (viz obrázek 2.29).



Obrázek 2.29: Princip radiového vysílání³⁵

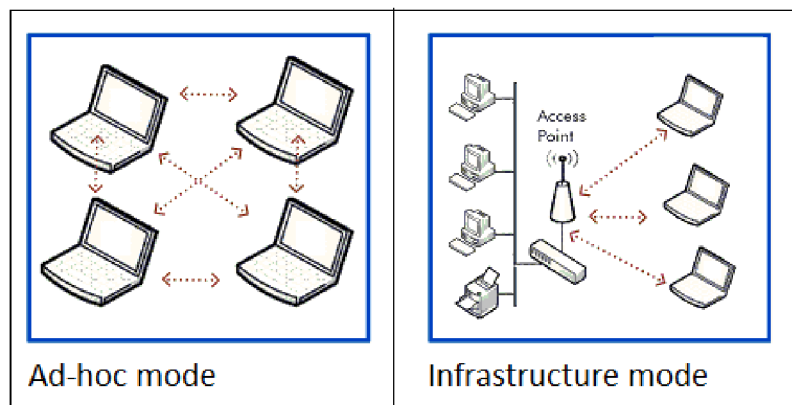
Bezdrátové technologie jsou ve vestavěných systémech hojně využívány, ať už je tomu jen pro náhradu spojů mezi hlavním zařízením a senzory, nebo pro možnost složitější komunikace zařízení, například přes internet. Technologií využitelných k těmto účelům existuje hned několik a liší se primárně tím, na které konkrétní využití cílí svojí hardwarovou implementací a nabízenými službami.

Wi-Fi

Wi-Fi (označována také IEEE 802.11) je technologie bezdrátového přenosu, která využívá radiové vlny na frekvencích 2,4 a 5 GHz [6]. Primárně je známá pro své využití v bezdrátovém připojování zařízení k internetu, z důvodu že umožňuje rychlý přenos dat na rozumné vzdálenosti. Síť tvořená technologií Wi-Fi se nazývá WLAN (bezdrátová LAN). Wi-Fi také nabízí dobrou bezpečnost poskytované sítě, využívá k tomu jak omezení připojení zařízení k síti, tak pak samotné šifrování přenášených dat (příklad WPA2).

Přenos dat pomocí Wi-Fi lze realizovat dvěma základními módy, infrastrukturním nebo takzvaným Ad-hoc módem (viz obrázek 2.30). Primárně využívaný je infrastrukturní mód, který funguje tak, že obsahuje jedno hlavní zařízení, které funguje jako přístupový bod a ke kterému se musí všechny ostatní zařízení připojit pokud chtějí přijímat/posílat data v dané síti. V tomto módu je přenos dat prováděn vždy jen mezi připojeným zařízením a řídicím zařízením, přímý přenos dat mezi jednotlivými připojenými zařízeními není možný. Ad-hoc mód oproti tomu umožňuje jednotlivým zařízením tvořit mezi sebou účelové sítě, ve kterých si přenos dat řídí sami zařízení.

³⁵Převzato z: <https://cdn4.explainthatstuff.com/transmitterandreceiver.png>



Obrázek 2.30: Módy Wi-Fi sítě³⁶

Wi-Fi prošlo v historii rozsáhlým vývojem, který postupně zlepšoval veškeré vlastnosti tohoto bezdrátového přenosu. Postupem času se zlepšovaly rychlosti přenosu, dosah spojení a zabezpečení dat. Nejnovější verze nabízí maximální rychlost 450 Mbit/s a dosah od 30 - 100 m u standardních vysílačů.

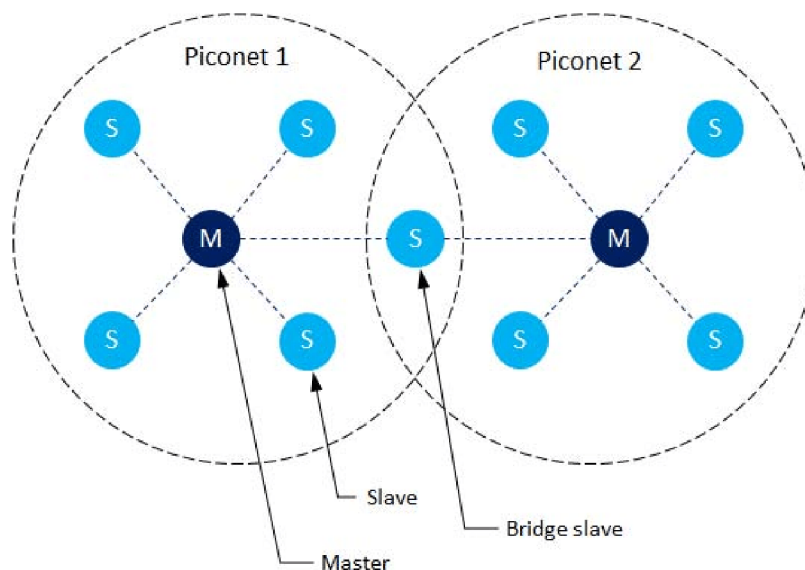
Pro účely vestavěných zařízení se bezdrátová technologie Wi-Fi používá u specifických případech. Může být použita pro připojení zařízení k internetu nebo například pro bezdrátové spojení se senzorem, který potřebuje odesílat či přijímat velké množství dat (příkladem může být kamera). Technologie Wi-Fi byla vyvíjena hlavně s důrazem na rychlost přenosu a z tohoto důvodu je její spotřeba vysoká. Pro některé typy zařízení je pak tato spotřeba nevýhodou.

Bluetooth

Bluetooth je bezdrátová technologie, která využívá frekvenci 2,4 GHz pro přenos dat. Známa je hlavně pro svoje využití v bezdrátovém připojování periférií [18]. Síť tvořená touto technologií se typově řadí do sítí PAN (Personal Area Network). Navázání spojení mezi zařízeními je zabezpečeno pomocí párování a samotná přenášená data jsou enkryptována pro zabezpečení jejich obsahu.

Síť vytvořená technologií Bluetooth se odborně nazývá Piconet. Komunikace v této síti je založena na modelu master/slave, to znamená že všechny zařízení jsou připojena na jedno hlavní, které řídí celou komunikaci. Slaves zařízení musí data posílat jen masterovi, nemůžou si je posílat mezi sebou, pokud si data mezi sebou potřebují vyměnit, musí je přenést přes mastera. Navíc také slaves zařízení nemůže komunikovat kdykoli, komunikace je vždy zahájena master zařízením, když si data vyžádá od nějakého slave zařízení. V jedné Piconet může být maximálně 8 zařízení, 1 master a 7 slaves. Technologie Bluetooth však navíc poskytuje funkčnost, kdy je možné připojit jedno slave zařízení ke dvěma masterům, kteří ovládají dvě rozdílné sítě a toto slave zařízení pak poskytuje výměnu dat mezi těmito dvěma sítěmi (viz obrázek 2.31).

³⁶Převzato z: <https://www.researchgate.net/profile/Viraj-Shah-5/publication/316175326/figure/fig1/AS:495727332610048@1495202014846/Ad-hoc-mode-vs-Infrastructure-mode-IEEE80211-introduced-many-types-of-the-Wi-Fi.png>



Obrázek 2.31: Spojení dvou Piconets mostovým slave zařízením³⁷

Pro adresování jednotlivých zařízení v síti se používají takzvané MAC adresy, které jednotlivým Bluetooth zařízením přidělují jejich výrobci. MAC adresa je tvořena 48 bity, kde spodních 24 bitů je označení výrobce a horních 24 bitů je unikátní číslo, které výrobce přidělil danému zařízení. Samotná adresa se pak zobrazuje v hexadecimálním tvaru (příklad: 000666422152).

Bluetooth také podporuje takzvané rozšiřující profily, které staví na protokolu Bluetooth a upravují ho pro použití k odesílání určitých specifických dat. Mezi nejpoužívanější profily se řadí:

- **Profil sériového portu:** profil používaný pro nahrazení sériové komunikace jako RS-232 nebo UART. Umožňuje zařízením odesílat data stejným způsobem jako kdyby používaly TX a RX kabely.
- **Zařízení lidského rozhraní:** umožňuje jednoduchý přenos informací pro zařízení jako myši, klávesnicemi, kontrolery atd.
- **Hands-Free profil:** umožňuje příjem zvuku i odesílání zvuku zaznamenaného mikrofonom. Navíc umožňuje jednoduché funkce pro správu hovorů (přijmutí, odmítnutí atd.)
- **Profil pokročilé distribuce zvuku:** používá se pro jednosměrné odesílání zvuku, ať už pro přehrávání do sluchátek nebo přenos zaznamenaného zvuku z mikrofону.
- **Profil dálkového ovládní A/V:** většinou se implementuje současně s profilem pro pokročilou distribuci zvuku a umožňuje dálkové ovládní přes Bluetooth zařízení. U bezdrátových sluchátek to může být například přeskočení skladby, zastavení přehrávání atd.

³⁷Převzato z: <https://www.researchgate.net/publication/284812098/figure/fig4/AS:390973147369474@1470226670192/Architecture-of-Bluetooth-network.png>

Bluetooth se vyrábí ve třech třídách výkonu. Zákazník si tedy může vybrat, tu která je pro jeho použití nejlepší. Třída 1 nabízí vysílací výkon 100 nW a dosah až 100 m. Třída 2 nabízí vysílací výkon 2.5 nW a dosah až 10 m. Třída 3 nabízí vysílací výkon 1 nW a dosah až 10 cm. V běžné spotřební elektronice je většinou implementována třída 2.

Aktuálně existují tři verze Bluetooth, kde každá verze je určena pro konkrétní využití.

- **Klasické:** Bluetooth označení 2.1 EDR (Enhanced data rate) s maximální rychlostí přenosu 2.1 Mbitů za sekundu.
- **Vysokorychlostní:** Bluetooth označení 3.0 s maximální rychlostí přenosu 24 Mbitů za sekundu. Tato verze využívá k samotnému přenosu dat technologii Wi-Fi, samotná technologie Bluetooth slouží jen pro navázání a spravování Wi-Fi spojení mezi komunikujícími zařízeními.
- **BLE (nízkoenergetické Bluetooth):** Bluetooth označení 4.0, které míří na co nejmenší spotřebu energie. Díky velké úspoře spotřebované energie je však snížen maximální dosah na 50 m ze 100 m a maximální rychlost datového přenosu je snížena na 0.27 Mbit za sekundu.

Ve vestavěných systémech je Bluetooth technologie hojně využívána pro připojení jednoduchých senzorů, u kterých jde hlavně o jednoduchost spojení, malé nároky na spotřebu a není potřeba přenášet velké objemy dat.

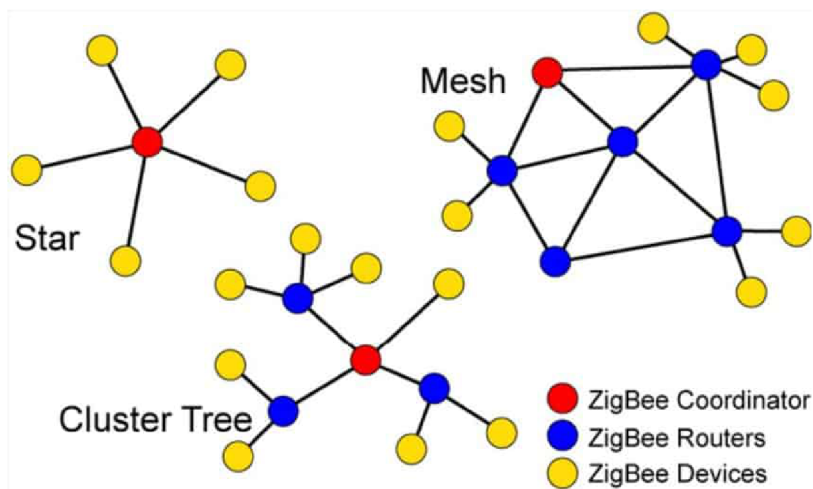
ZigBee

Bezdrátová technologie ZigBee pracuje na frekvencích 868 Mhz, 915 Mhz a 2,4 GHz [9]. Tato technologie byla vytvořena za jediným účelem a to pro domácí a industriální automatizaci. Proto je velký důraz kladen na nízké energetické nároky, bezpečnost, nízkou cenu a standardizaci. Tím že je ZigBee navrženo pro uplatnění v automatizaci, je očekáváno, že v síti bude připojeno převážně velké množství senzorů, které nepotřebují velké nároky na přenos dat. Síť tedy umožňuje připojení 65 000 zařízení a nabízí přenosovou rychlost 250 Kbitů za sekundu. Dosah bezdrátového spojení se pohybuje od 10 do 100 m. Protokol sítě ZigBee také obsahuje plnou kompatibilitu s protokolem IP, je tedy možné připojení celé sítě k internetu přes router a umožnit tak jednotlivým zařízením komunikovat přes internet.

Síť nabízená technologií ZigBee se skládá ze třech základních prvků:

- **Koordinátor:** koordinátor je hlavním zařízením celé bezdrátové sítě. Umožňuje připojení nových zařízení do sítě, ukládá si bezpečnostní klíče a nabízí možnost spojit dvě rozdílné sítě mezi sebou. V každé síti může být jen jeden koordinátor.
- **Směrovač:** směrovače zajišťují směrování posílaných dat v síti. V jedné síti může být několik směrovačů.
- **Koncové zařízení (uzel):** koncové zařízení může být například senzor, který má zaznamenávat data. Koncových zařízení může být v síti velký počet, ale nikdy nesmí být připojena dvě koncová zařízení přímo na sebe. Musí komunikovat vždy přes uzel typu směrovač nebo koordinátor.

Technologie ZigBee také umožňuje tvorbu celé řady typů sítí podle jejich výsledného využití (viz obrázek 2.32).



Obrázek 2.32: Typy sítí ZigBee³⁸

Technologii ZigBee ve vestavěných systémech je nejlepší využít pro malé senzory, které musí být připojeny k velké síti senzorů. Do této sítě posílají svá zaznamenaná data hlavnímu zařízení, které je pak využívá.

³⁸Převzato z: <https://gadget-freakz.com/wp-content/uploads/ZigBee-network-topologies.jpg>

Kapitola 3

Provozování ultralehkých letounů

Před započítáním práce na systému, který je tvořen v této práci, je nutné porozumět problematice ultralehkých letadel, pro kterou je systém tvořen. Je zde vysvětleno co to jsou ultralehké letouny, jak se provozují a jak se dnes tvoří záznamy o jejich využívání. Tyto údaje jsou nezbytné pro následný návrh jednotlivých částí systému. Je však potřeba zdůraznit, že následující popis poznatků nemá encyklopedický charakter a jsou zde popsány informace, nezbytně nutné k pochopení návrhu tvořeného systému. Limit rozsahu práce totiž nedovoluje podrobnější popis.

3.1 Popis ultralehkých letounů a způsobu záznamu letových časů

Letadla i vrtulníky, které mají kapacitu pro jednu až dvě osoby a mají vzletovou hmotnost do 600 kg (viz obrázek 3.1) se většinou řadí do kategorie ultralehké letouny [2]. Ultralehké letouny jsou populární mezi amatérskými piloty díky tomu, že na jejich pilotování stačí průkaz na ultralehká letadla (ULL), který je znatelně jednodušší na získání než standardní pilotní průkaz (PPL), pro který je nutné nalétat více letových hodin a mít obecně větší znalosti a výcvik. Výhodou ultralehkých letounů oproti standardním je také jejich nižší pořizovací cena, cena údržby a cena paliva.

Stále je však běžné že každý pilot nevlastní svůj ultralehký letoun. Ultralehký letoun si piloti běžně pronajímají nebo ho má koupeno více pilotů dohromady do společného vlastnictví. Tím, že se stroj půjčuje mezi více lidí, nastává problém s tím, že musí být zaznamenáváno kdy a kdo se strojem letěl pro následné rozpočítání provozních nákladů¹.

Tvorba záznamů o provedených letech je dnes primárně založena na manuální práci pilotů ultralehkých letounů. Každý pilot vždy musí zaznamenat čas, který letěl do provozního deníku ultralehkého letounu a do svého pilotního deníku. Pro rozpočítání výsledných nákladů je tedy většinou nutné všechny ručně vytvořené záznamy přepsat například do Excelu, ve kterém se následně povedou výpočty, kolik má, který pilot zaplatit za jeho odlítaný čas. Každý provedený let se běžně zaznamenává od vzletnutí ultralehkého letounu do jeho následného přistání na dráze, pokud by však pilot přistál a hned se chystal na další let, tak se vše počítá jako jeden uskutečněný let.

¹Dle tvrzení dotázaných pilotů z přistávací plochy v Kotvrdovicích. <https://www.lkkotv.cz>



Obrázek 3.1: Příklad ultralehkého letadla Aeroprakt A-22 Foxbat

3.2 Podrobný popis provozu ultralehkých letounů

V této sekci je popsán podrobný popis provozu ultralehkých letounů, který je usměrněn předpisy Letecké amatérské asociace [1]. Principy provozu jsou důležité pro zaznamenávací systém, který musí znát kdy má začít zaznamenávat let a kdy má let ukončit, také musí znát speciální případy provozu které mohou nastat.

Provoz každého ultralightu je složen z následujících bodů:

- **Vytažení ultralehkého letounu a předletová kontrola:** Před letem musí piloti nejdříve vytáhnout ultralehký letoun před hangár, kde provádí vizuální kontrolu trupu a kontrolu pohyblivosti řídicích ploch.
- **Nastartování, pojiždění a motorová zkouška:** Následuje nastartování motoru a popojíždění na stojánku, na kterém se provádí zahřívání provozních kapalin a motoru. Zahřívání se provádí na střední výkon po dobu přibližně 5 minut. Po nahřátí motoru se provede motorová zkouška při které se dá motor na dobu pár sekund do vysokého výkonu a testuje se zda má očekávaný výkon.
- **Vypnutí motoru a nástup cestujícího:** Tento bod nastává jen pokud musí pilot nabrat cestujícího. Před nástupem cestujícího je nutné vypnout motor, dále musí pilot cestujícího usadit a zajistit jeho bezpečné zajištění do sedačky. Nakonec může pilot znovu nastartovat motor a pokračovat dalším bodem.
- **Vzlet:** V tomto bodě musí pilot ohlásit odlet ostatním pilotům, po ověření volné dráhy musí dojet pilot na její začátek ze kterého bude startovat. Pro odstartování se provede nastavením výkonu motoru na skoro maximálního výkonu a čeká se na dosažení vzletové rychlosti, pak je možné s ultralehkým letounem vzlétnout. Letový čas piloti začínají počítat od momentu nastavení vzletového výkonu.
- **Let:** Během letu pilot naviguje ultralehký letoun po zvolené trase a přizpůsobuje výkon motoru a rychlost aktuálním letovým podmínkám. Během letu může být nastavován výkon motoru od volnoběhu do maximálních otáček.
- **Přistání:** Před zahájením přibližovacího manévru pilot ověří volnost dráhy. Pokud je dráha prázdná tak začne klesat nad dráhu s minimálním výkonem motoru a stabilně

sníží rychlost. Po přistání pilot dojde k hangáru, kde vypíná motor a uklízí ultralehký letoun do hangáru. Konec letu si piloti poznačují po přistání, když se rychlost sníží na rychlost pomalého poježdění. V tomto bodu může kromě normálního vypnutí a úklidu letounu, také nastat případ že pilot bude po přistání pokračovat v dalším letu. V tomto případě se po přistání časomíra nevypíná a počítá se i krátký čas, který ultralehký letoun strávil na zemi než znovu vzletl.

3.3 Určování letového času pomocí motohodin

Nejobyčejnějším způsobem jak určit letový čas ultralehkého letounu je přes počítadlo provozního času motoru [3], které bývá namontováno v nějaké podobě v takřka všech strojích (viz obrázek 3.2). Princip této metody spočívá v tom, že si musí pilot poznačit hodnotu před letem a po letu ultralehkého letounu a z rozdílu následně může vypočítat výslednou dobu letu, kterou si následně zapisuje do deníku.



Obrázek 3.2: Příklad čítače provozního času²

3.4 Určování letového času stopkami

Pohodlnější metodou pro piloty je zaznamenávání letového času pomocí stopek [4] (viz obrázek 3.3). Zde pilotovi stačí, aby při startu ultralehkého letounu zapnul časomíru a při ukončování letu ji manuálně vypnul. Stopky pilotovi poskytnou přesný čas letu, který už jen musí dále poznačit. Pro účely měření se používají jak digitální tak i mechanické stopky.

²Převzato z: http://www.tl-ultralight.cz/zh/ultralight-aircraft/tl-2000-sting-rg/_include-cockpit/kokpit01-10805

³Převzato z: <https://www.pooleys.com/media/3426/1218.jpg>



Obrázek 3.3: Příklad leteckých stopek³

3.5 Zaznamenání letu sofistikovanými letovými přístroji

Nejpokročilejší metoda pro záznam letů je dostupná jen v novějších ultralehkých letounech, do kterých jsou namontovány pokročilejší letové přístroje [5] (viz obrázek 3.4). Nové letové přístroje díky technologii GPS 2.4 dokážou samy zaznamenat, kdy ultralehký letoun letí a ke kterému letu si zaznamenají jeho trasu i čas. Tato metoda má tedy ze všech ostatních metod nejbližší k automatickému zaznamenávání letu.



Obrázek 3.4: Příklad kokpitu s moderními letovými přístroji⁴

Analýza provozu ultralehkých letadel a rozbor jednotlivých zaznamenávacích metod je rozebrán v následující kapitole 4.

⁴Převzato z: https://lh3.googleusercontent.com/proxy/n0Gi_rE85UjhcbVswpsKM3J8qQex6q7bqmXpBbBSqXCWgy0_4FDvopTeWfRiYvV7oUAgod2v0SX1LfmKRRireoDJY-yAD2YOPnpFF0bUkKNHrzBX0kp6_6PifqjqBPm4tRyDy1LS8d3dWOAmEPUmHUKIFg4SUFaIiXmNGUYd4WsKIUY_Xe50

Kapitola 4

Zhodnocení provozu a specifikace zaznamenávacího systému

Tato kapitola se zabývá vyhodnocením informací o způsobech aktuálního provozu ultralehkých letounů, rozhodnutím o funkcích tvořeného systému a vytvořením podrobné specifikace založené na požadavcích a potřebách pilotů a majitelů ultralehkých letounů. Získání relevantní a podrobné specifikace napomáhá k rozhodnutí o postupu tvorby systému v této bakalářské práci. Přesná specifikace limituje možné problémy způsobené nejistým popisem, které by se musely eliminovat až v průběhu samotné implementace.

4.1 Zhodnocení aktuálních metod měření letového času

Hlavním problémem zaznamenávání letových hodin je manuální pořizování záznamů piloty. Nevylučuje se totiž, že piloti vždy zapíší přesné hodnoty. Je tedy běžné, že vypočtené náklady k úhradě nejsou přesné. Samotný výpočet nákladů také není jednoduchý, protože se provádí přenášením z ručních záznamů pilotů například do Excelu, kde se provádě samotný výpočet.

Zhodnocení metody měření letového času pomocí motohodin

Je zřejmé, že není jednoduché si poznačit přesnou hodnotu počítadla při vzletu a při přistání letounu, výsledná hodnota letového času bývá tedy zkreslená. Problém je v tom, že ne každý letoun má podrobné počítadlo provozního času. Většina strojů má totiž možnost zaznamenávat jen hodiny, jen pár strojů má nižší jednotky. Díky nepřesným výsledkům, malému počtu letounů s přesnými počítadly a obecné nejednoduchosti měření se tato metoda běžně nepoužívá a lze ji použít pokud nemá pilot jinou možnost.

Zhodnocení metody měření letového času pomocí stopek

Tato metoda je mezi amatérskými piloty hodně rozšířená díky své jednoduchosti, pro starší typy ultralehkých letounů je to také takřka jediná možná metoda, jak měřit letový čas. Problém je v jejím manuálním charakteru, kde je vyžadováno, aby pilot nezapoměl, že stopky musí ve správný čas zapnout a vypnout a následně si tento údaj poznačit

Zhodnocení metody se sofistikovanými letovými přístroji

Tato metoda je velice blízká automatickému zaznamenávání a vyžaduje po pilotech, aby si data manuálně z přístroje přepisovali, protože tyto přístroje neumožňují stáhnutí dat v elektronické podobě. Problémem je, že se zde stále nevylučuje možnost, že si pilot data zapomene přepsat, většina přístrojů si data neukládá, takže po vynutí je není možné získat. Dále je stále potřeba zaznamenaná data přepsat do určitého programu, ve kterém se uskuteční kalkulace provozních nákladů.

Nevýhodou je také, že znatelné množství pilotů používá jednoduché ultralehké letouny bez pokročilých letových systémů, které díky absenci pokročilých systémů jsou jednodušší a levnější na údržbu. Pokud ultralehký letoun obsahuje moderní letové přístroje, pořád může nastat, že je systém nepřehledný a většina pilotů neví, kde má tato data hledat a používají jinou jednodušší metodu.

4.2 Rozhodnutí o realizaci systému

Z uvedených metod je viditelné, že mají svoje výhody i nevýhody. Hlavní společný problém je ten, že si pilot musí vždy výsledný čas letu manuálně poznačit. V této práci stačí využít jednoduché principy popsaných metod a je důležité, aby veškeré ukládání letových časů bylo prováděno automaticky, pak budou všechny záznamy o letech v elektronické podobě a nemůže se stát, že by pilot zapomněl záznam vytvořit.

Z popisu aktuálního stavu provozu ultralehkých letounů je patrné, že čas, který si piloti značí se počítá přibližně od doby zvýšení otáček motoru k maximálnímu výkonu po dobu několika desítek sekund (nebo zvýšená rychlosti k hranici vzletové rychlosti) převážně do doby než se vypne motor ultralehkého letounu. Veškerý ostatní čas, kdy se provádí zkoušky motoru a pojíždění se nepočítá. Je tedy výhodné tyto poznatky použít pro návrh zaznamenávacího systému.

Na základě analýzy provozu ultralehkých letounů a aktuálních metod měření letů je patrné, že v současné době neexistuje žádný systém, který by automaticky zaznamenával proběhlé lety ultralehkého letounu a poskytl by následně data pilotům pro jednoduchou kalkulaci provozních nákladů. Rozhodl jsem se tedy vytvořit nové zařízení, které tuto funkci umožní. V rámci této bakalářské práce vznikla experimentální podoba tohoto systému, která slouží k ověření zda je této funkčnosti možné jednoduše dosáhnout. Systém je dále možné zdokonalit až do komerční podoby.

Na základě zadání a nastudovaných informací jsem se rozhodl vytvořit jednoduché vestavěné zařízení, které automaticky zaznamenávat proběhlé lety. Logika záznamu letu bude založena na principech provozu a měření letů ultralehkých letounů. Všechny záznamy budou zaznamenávány do vestavěného zařízení a stahovat a editovat pomocí přístupového zařízení.

4.3 Podrobná specifikace systému

Po určení primární funkčnosti systému je nutné stanovit jednotlivé body specifikace podle kterých bude následně systém implementován. Je potřebné specifikovat podobu vestavěného zařízení, napájení, princip zaznamenávání letů, určení operujícího pilota a přístup k vytvořeným záznamům.

Jednotlivé body specifikace jsou uvedeny v následujících odrážkách:

- Řídící prvek vestavěného zařízení bude jednodeskový minipočítač,
- Rozměry vestavěného zařízení by měly být do 20x15x5 cm, aby zařízení nezabíralo moc místa a šlo jednoduše nainstalovat,
- Řídící prvek by měl umožňovat připojení nejrůznějších rozšiřujících komponent, které budou nutné pro automatický záznam letů,
- Systém by měl podporovat až 20 pilotů,
- Kapacita zaznamenaných letů by měla být v řádu tisíců,
- Vestavěné zařízení by mělo být možno umístit v okolí přístrojové desky ultralehkého letounu, kde je dobrý dosah ke všem spojům, které je k zařízení nutné připojit,
- Systém bude napájen v době provozu letadla. V ostatní okamžiky není potřeba, aby fungoval,
- Napájení vestavěného zařízení bude zajištěno z interní sítě ultralehkého letounu, která poskytuje napětí 12 V a proud do 3 A,
- Výběr pilota bude možný přes malé tlačítko a aktuálně vybraný pilot bude zobrazen na malém displeji. Oba tyto prvky budou umístěny na přístrojové desky letounu,
- Pokud se pilot zapomene před letem identifikovat, bude let uložen bez pilota,
- Ke každému letu je nutné uchovávat jeho dobu trvání a čas, kdy let proběhl, možné je i zaznamenání trasy provedeného letu,
- Přístup, editace i stahování dat z vestavěného zařízení bude umožněno uživatelům bezdrátově za pomoci mobilního telefonu,
- Přístup k datům bude rozdělen z důvodu bezpečnosti dvěma rolemi. Majitel bude moci prohlížet, editovat a stahovat všechna data. Piloti budou moci prohlížet, editovat a stahovat svá data.

Podrobný návrh a realizace zde specifikovaného systému je předmětem kapitoly 5.

Kapitola 5

Tvorba zaznamenávacího systému a přístupové mobilní aplikace

Následující kapitola detailně popisuje, jak byl vytvořen finální systém navržený pro plnění očekávané funkčnosti stanovené popisem této práce. Popsán je návrh jednotlivých částí systému, výběr komponent, které budou pro tento systém vhodné, přesný způsob zapojení jednotlivých komponent používaných v systému, které jsou potřebné k plnění dílčích úkolů. Následně jsou popsány nejdůležitější informace z implementace softwaru vestavěného zařízení namontovaného v letadle, plnicí funkci zaznamenávání letu a mobilní aplikace, která slouží pro práci se záznamy uloženými na vestavěném zařízení. Dále je popsána testovací montáž systému, otestování a vyhodnocení výsledků, které určují, jak dobře a za jakých podmínek je systém použitelný. Na závěr je uveden výčet možných budoucích vylepšení systému.

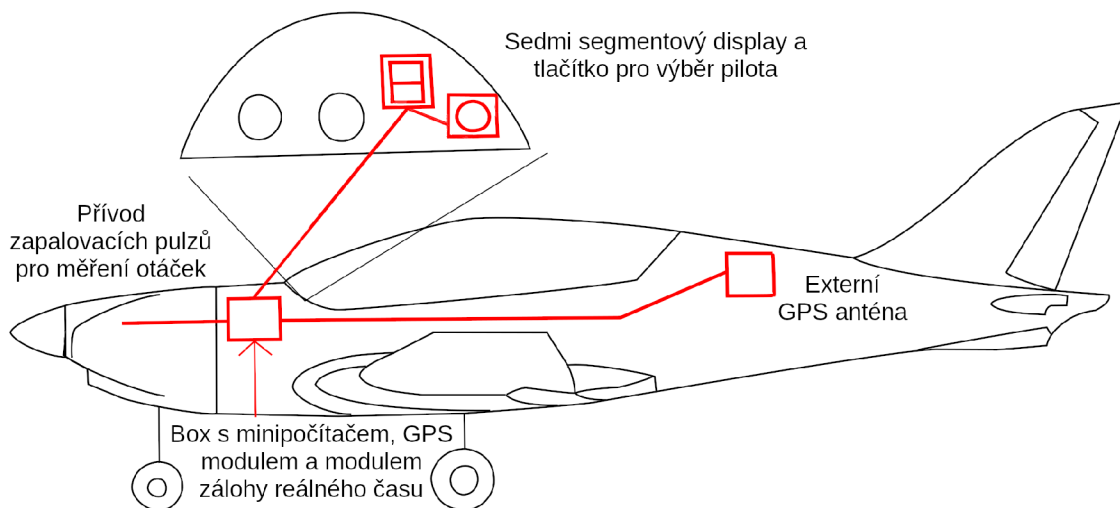
5.1 Návrh struktury tvořeného systému

Ze získaného podrobného popisu je vytvořen návrh struktury tvořeného systému, který podrobněji popisuje, které části systém obsahuje a jak mezi sebou komunikují. Struktura systému byla navrhována s ohledem na jednoduchou možnost použití v ultralehkých letounech a celkovou nenáročnost jejich konstrukce. Jednoduchá konstrukce také přispěje dobrému rozšíření systému mezi amatérskými piloty, kteří si budou moci tento systém do svého letadla implementovat a získat tak přehled nad využíváním jimi vlastněných letadel.

Tvořený systém (viz obrázek 5.1) je složen z bloku, ve kterém je uložen mikropočítač, který automaticky zaznamenává lety a komponent potřebných pro zaznamenávání letu. Je zde přítomný obvod, který umožňuje připojení zapalovacích pulzů motoru na minipočítač, GPS modul a modul zálohy reálného času. Pulzy otáček zde slouží jako nejjednodušší způsob určení stavu ultralehkého letounu pro který je následně potřebný modul reálného času který určuje kdy letoun provedl svůj let. Jako pokročilejší způsob zaznamenání letu byl použit GPS modul, který dokáže zaznamenat také trasu letadla a automaticky určit aktuální čas. Pro lepší signál GPS modulu byla přidána externí anténa.

Pro zobrazení operujícího pilota byl zvolen segmentový displej, který je jednoduché připojit na minipočítač a také podporuje 20 znaků, které jsou potřeba pro rozlišení pilotů. Pro volbu pilota stačí malé tlačítko. Oba tyto prvky jsou vyvedeny na palubní desku letadla, kde k nim má pilot jednoduchý přístup.

Pro bezdrátový přístup k datům a editaci dat jsem se rozhodl použít technologii Bluetooth. Bluetooth je přítomné ve většině moderních mobilních zařízení a nabízí jednoduchý způsob spojení dvou zařízení. Vytvořené spojení nemusí být moc rychlé, protože je potřeba posílat je základní textová data, která budou reprezentovat jednotlivé uložené záznamy. Jako cílový operační systém pro výslednou mobilní aplikaci byl zvolený operační systém Android. Vybrán byl kvůli své rozšířenosti mezi mobilními zařízeními, jednoduchou tvorbu a publikaci nových aplikací.



Obrázek 5.1: Schéma systému

5.2 Výběr komponent pro vestavěné zaznamenávací zařízení

Na základě návrhu systému je potřeba vybrat hardware, který splňuje očekávanou funkčnost a splňuje všechny požadavky potřebné pro jeho bezpečnou implementaci do ultralehkého letounu. U každého zvoleného prvku je uveden jeho obecný popis a důvod proč byl vybrán.

Raspberry Pi 3 Model B+

Jako řídicí prvek byl vybrán minipočítač Raspberry Pi 3 (viz obrázek 5.2), který je pro tuto práci více než dostatečně výkonný. Lze na něm bez problémů provádět testování a psaní kódu, případně bude možné i jednoduché dodělání případných vylepšení. Tímto mikropočítačem je také pokryt případ, pokud by systém měl mnohem větší nároky než bylo očekáváno. Výhodou je i dostatečný počet pinů s podporou sériových komunikačních protokolů, na které se dají připojit složitější rozšiřující periferie, malá velikost, ze základu zabudované bezdrátové přenosové technologie Wi-Fi a Bluetooth a možnost instalace paměťových mikro SD karet, na které lze nahrát velké množství záznamů.

Výhodou je také velké množství tutoriálu pro tuto platformu a možnost programování v programovacím jazyku Python, který je pro tvořenou logiku více než dostatečný.



Obrázek 5.2: Raspberry Pi 3B+¹

GPS modul NEO-7M

Modul NEO-7M (viz obrázek 5.3) umožňuje vestavěnému zařízení zjistit jeho aktuální polohu pomocí satelitů GPS. Tento modul umožňuje datové připojení pomocí technologie UART a je možné ho napájet 5V stejnosměrného napětí, které je možné brát přímo z výstupních pinů minipočítače. K celkové integraci také přispívá jeho malá velikost. Hlavním důvodem proč byl tento modul vybrán je možnost připojení externí antény pomocí SMA konektoru.



Obrázek 5.3: GPS modul NEO-7M²

¹Převzato z: <https://c1.neweggimages.com/ProductImage/13-142-011-Z01.jpg>

²Převzato z: https://dratek.cz/photos/produkty_gal/f/3/3414.jpg?m=1502871565

Externí GPS anténa

Externí anténa (viz obrázek 5.4) je důležitá pro zlepšení signálu přijímaného GPS modulem. GPS modul bude namontován blízko samotného řídicího prvku systému, který bude umístěn za přístrojovou deskou letadla. Lze tedy očekávat, že signál v této oblasti nebude nejlepší a může být rušený z okolních přístrojů, také samotná GPS anténa modulu není robustní a její signál za příznivých podmínek nemusí být nejlepší.

Pro získání co možná nejlepšího GPS signálu je nutné použít externí GPS anténu, která se k modulu připojí přes SMA konektor. Díky 5m dlouhému kabelu je možné si zvolit nejvhodnější místo k namontování na ultralehkém letounu, které zaručí jak dobrý signál tak snadnou instalaci.



Obrázek 5.4: Externí GPS anténa³

RTC modul DS3231

Modul reálného (viz obrázek 5.5) času slouží pro uchování hodnoty aktuálního času, který je potřebný pro tvorbu letových záznamů, i přes dobu kdy je zařízení odpojeno od elektrické energie. Samotný minipočítač totiž žádnou funkci uchování času neposkytuje a po jeho vypnutí se předem nastavená hodnota ztratí. Modul si uchovává aktuální čas díky připojené knoflíkové baterii LIR2032, která napájí oscilátor zařízení, když není připojeno vstupní napájecí napětí. Komunikace s tímto zařízením je zajištěna sériovou komunikační sběrnici I2C.

Zařízení obsahuje přesný teplotně kompenzovaný oscilátor, takže nedochází bez pravidelné synchronizace k tak velké odchylce od reálné hodnoty času. Tento problém je však možné eliminovat úplně, díky připojenému GPS modulu, který umí získat ze satelitů mimo aktuální polohy i aktuální čas v místě zařízení. To umožní aktuální čas pravidelně synchronizovat.

³Převzato z: https://www.ahifi.cz/files/products_images/big/s/smac1.jpg

⁴Převzato z: <https://dratek.cz/photos/produkty/f/1/1261.jpg?m=1502871291>



Obrázek 5.5: RTC modul DS3231⁴

Tlačítko 12mm

Pro změnu aktuálního pilota byl vybrán jednoduchý spínač (viz obrázek 5.6) s možností panelové montáže. Tento spínač nabízí jednoduchou montáž do palubní desky letadla, díky použitému závitu. Připojení s řídicím prvkem bude navázáno přes GPIO piny, které pro jejich nastavení vytvoří jednoduchý obvod, který bude spojovat stisk tlačítka.



Obrázek 5.6: Tlačítko s panelovou montáží⁵

Segmentový LED displej

Pro zobrazení aktuálně zvoleného pilota byl vybrán malý (14,2 mm) segmentový displej (viz obrázek 5.7), který bude jednoduše připevněn na přístrojovou desku ultralehkého letounu

⁵Převzato z: <https://dratek.cz/photos/produkty/f/5/5406.jpg?m=1543267540>

a nebude překážet ostatním důležitým ukazatelům. Ovládán bude standardními GPIO piny jednodeskového počítače.



Obrázek 5.7: Segmentový display⁶

Stepdown měnič LM2596 DC-DC

Pro účely napájení je použit měnič napětí (viz obrázek 5.8), který vygeneruje napětí, na kterém pracují komponenty systému. Tento modul podporuje vstupní napětí 4.5 - 40 V a toto napětí dokáže měnit na výstupní napětí v rozmezí 3 - 35 V. Maximální proud podporovaný tímto modulem jsou 3 A, což je více než dostatečné. Pro účely tvořeného systému je potřeba zmenšit napětí 12 V, které je v rozvodné síti letadla na 5 V, na kterých pracují všechny komponenty použité v systému.



Obrázek 5.8: Stepdown měnič napětí LM2596⁷

⁶Převzato z: https://www.dexhal.cz/data/tmp/0/3/8883_0.jpg?1586095461

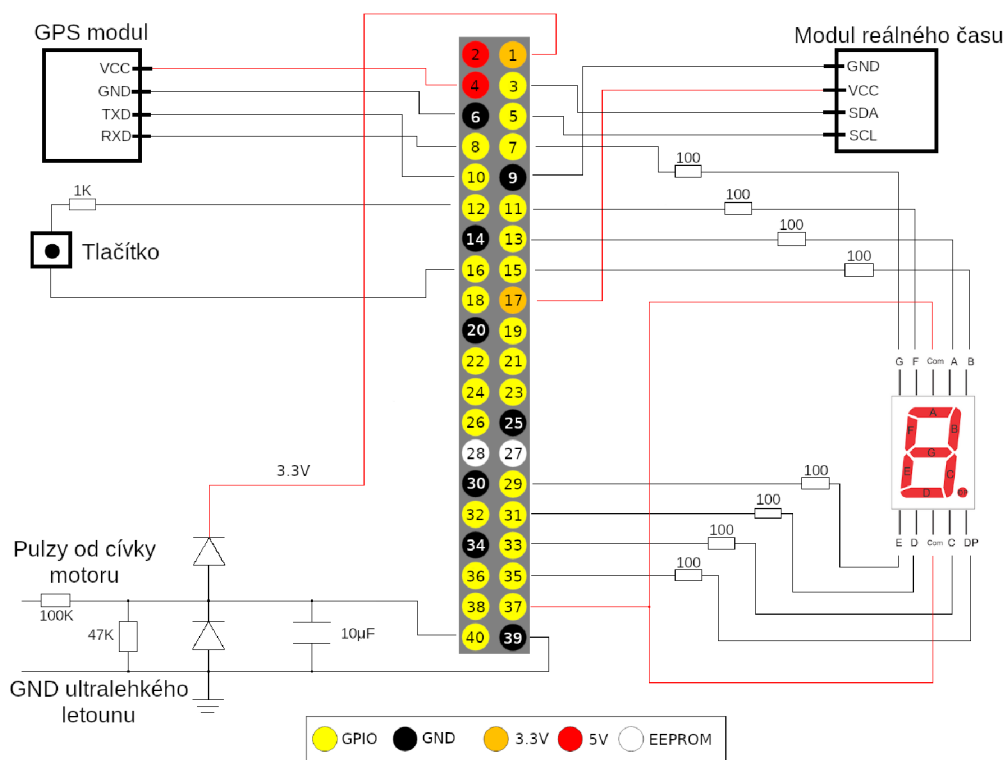
⁷Převzato z: <https://dratek.cz/photos/produkty/f/1/1303.jpg?m=1502871316>

5.3 Zapojení hardwaru zaznamenávacího systému

Před začátkem samotné práce na zapojení a implementaci tvořeného systému bylo nutné zakoupit veškerý potřebný hardware. Veškeré součástky je možné sehnat v českých internetových obchodech i za příznivé ceny (viz tabulka 5.1).

| | Cena v Kč | Obchod |
|----------------------------------|-----------|------------|
| Startovací sada Raspberry Pi 3B+ | 1822 | RPishop.cz |
| GPS modul NEO-7M | 525 | Drátek.cz |
| GPS anténa | 259 | ahifi.cz |
| RTC modul DS3231 | 66 | Drátek.cz |
| Tlačítko | 16 | Drátek.cz |
| Segmentový LED displej | 11 | Dexhal.cz |
| Akumulátor LIR2032 | 65 | Conrad.cz |
| Stepdown měnič LM2596 DC-DC | 45 | Drátek.cz |

Tabulka 5.1: Cena potřebných komponent. Ceny platné k 06.04.2021



Obrázek 5.9: Schéma zapojení ⁸

Pro vytvoření schématu zapojení vestavěného zařízení (viz obrázek 5.9) byly použity návody k daným komponentům, které popisují, co a jak je potřeba zapojit. Z důvodu toho, že systém je zamýšlen jen jako experimentální, jsem se rozhodl veškeré spoje provést

⁸Obrázek segmentového displeje převzat z: https://res.cloudinary.com/mynewsdesk-cld/image/upload/t_limit_1000/nxe09i6l6qi8ioi6cw6q.jpg

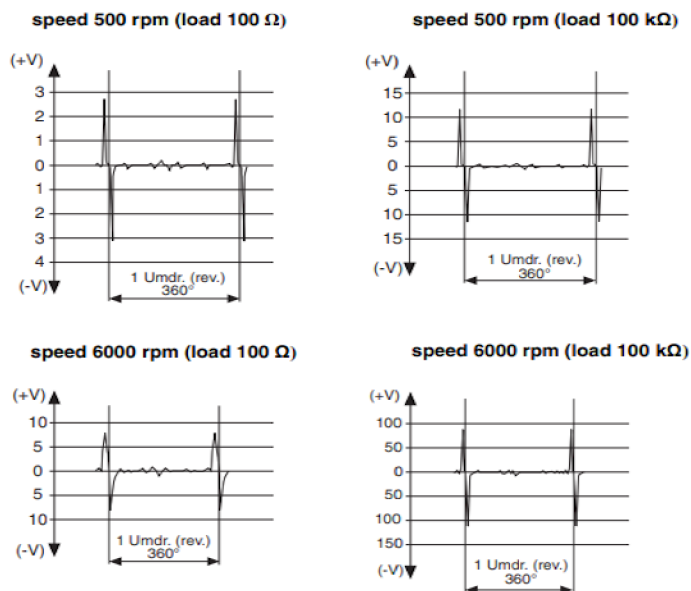
jen za pomoci kabelů s obyčejnými jednopinovými konektory, které na otestování stačí. Obvod přizpůsobení napěťových úrovní byl také vytvořen jen za pomoci svorkovnice. Důvod připojení zapalovacího pulzu motoru takovýmto způsobem je vysvětleno v sekci 5.4.

Raspberry Pi je možné napájet GPIO piny. Rozhodl jsem se však pro napájení přes USB, které bylo zamýšleno výrobcem jako hlavní způsob napájení a obsahuje tedy napěťovou ochranu, která není na GPIO pinech dostupná. Výhodou napájení systému je, že veškeré komponenty vyžadují buď 3.3 V nebo 5 V. Pro napájení všech komponent stačí tedy použít samotný minipočítač Raspberry Pi, který obě tyto hodnoty nabízí na svých GPIO pinech.

5.4 Přizpůsobení napěťových úrovní zapalovacích pulzů motoru

Podle četnosti zapalovacích pulzů lze určit aktuální počet otáček motoru za minutu, které jsou potřebné pro určení stavu ultralehkého letounu pro automatický záznamový systém. Schéma zapojení je koncipováno pro motor Rotax 912, který je umístěn v ultralehkém letadle, které bude použito na testy tohoto systému. Tento obvod bude dostatečný pro většinu ultralehkých letounů, protože téměř každý používá motor značky Rotax, není však vyloučeno, že pro některé ultralehké letouny bude potřeba tento obvod pozměnit podle motoru, který je v něm nainstalován.

Komponenty potřebné pro zachytávání zapalovacích pulzů motoru Rotax 912 byly odvozeny z průběhu ukázkových pulzů získaných z oficiální dokumentace motoru (viz obrázek 5.10). Použité komponenty zde omezují signál přicházející od motoru na hodnoty, které jsou použitelné řídicím počítačem a nijak ho neohroží. K tomuto účelu je tedy nutné velikost pulzů omezit na přípustnou hranici a odstranit nepotřebný záporný pulz.



Obrázek 5.10: Průběh zapalovacích pulzů motoru Rotax 912⁹

⁹Převzato z: <https://www.german-light-aircraft.com/attachments/article/172/d04967.pdf>

Vytvořený omezující obvod přicházející pulzy přetvoří do podoby viditelné na obrázku 5.11, která je vhodná pro použití řídicím počítačem. Na obrázku je vidět vždy jeden vysoký pulz, který má hodnotu do 4 V a označuje zážeh motoru. Zbytek průběhu je už jen šum přítomný na vodiči.



Obrázek 5.11: Průběh zapalovacích pulzů přizpůsobených vytvořeným obvodem

5.5 Popis implementace softwaru vestavěného zařízení

Kód vestavěného zařízení jsem se rozhodl psát v programovacím jazyce Python, protože k němu existuje mnoho dostupných návodů zabírajících se tématy, které byly nezbytné pro zprovoznění všech funkcí vestavěného zařízení. Python je také velice oblíbený programovací jazyk, používaný pro programování projektů pro platformu Raspberry Pi.

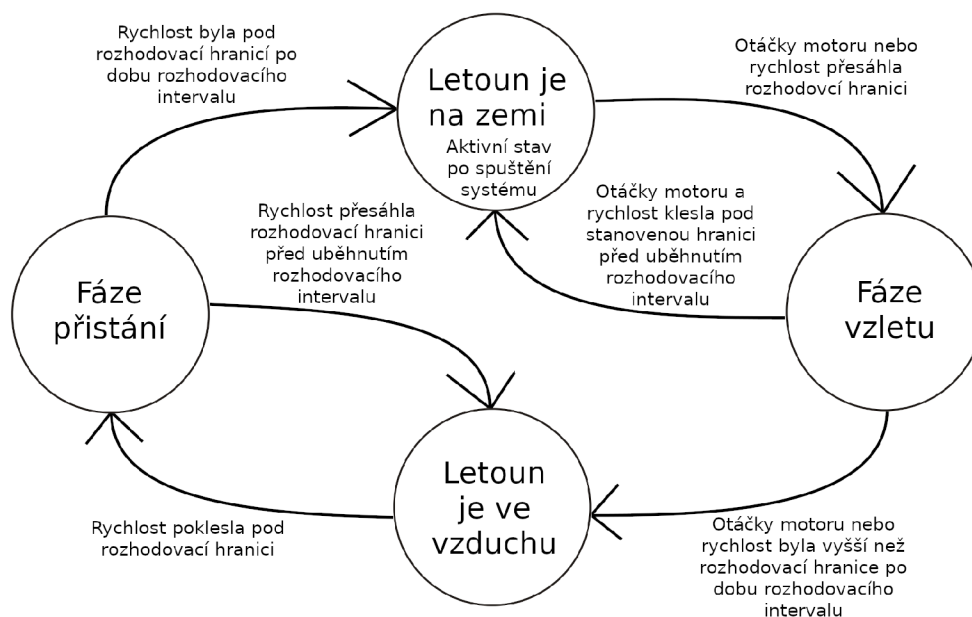
Celá funkčnost vestavěného zařízení je postavena na dvou úkonech, které je potřeba vykonávat. A to komunikace s mobilní aplikací přes Bluetooth a automatické zaznamenávání letových záznamů do lokální databáze. Z tohoto důvodu jsem tedy vytvořil dva rozdílné programy, kde každý z nich plní jen svoji přidělenou roli. Jediným společným bodem obou těchto programů je databáze, ve které jsou uloženy veškeré záznamy o pilotech a zaznamenaných letech. Pro implementaci databáze jsem zvolil SQLite¹⁰, které je pro Python velice jednoduché na integraci a poskytuje všechnu potřebnou funkcionalitu.

Hlavní částí celé implementace vestavěného zařízení je logika která určuje v jakém stavu se ultralehké letadlo nachází. Rozhodovací logika využívá k rozhodování informace které čte z připojených komponent. Hlavní podíl na určení stavu ultralehkého letounu má aktuální rychlost a aktuální počet otáček motoru za minutu. Vlastní rozhodovací logika se skládá ze čtyř částí (přechody mezi stavy jsou znázorněny v obrázku 5.12):

- **Letoun je na zemi:** Tento stav indikuje, že nejsou zvýšené otáčky motoru a ani není zaznamenána zvýšená rychlost letounu z GPS modulu. Když otáčky nebo rychlost překročí hranici stanovenou rozhodovacími parametry, tak systém přechází do fáze vzletu.

¹⁰<https://www.sqlite.org/index.html>

- **Fáze vzletu:** Tato fáze udává, že letoun zvyšuje svoji rychlost nebo otáčky motoru. Očekává se tedy, že letadlo začíná startovat. Když jsou otáčky nebo rychlost přes stanovenou rozhodovací hranici daný počet sekund, očekává se, že letadlo vzlétlo a systém přechází do fáze letu. Přechodem do fáze letu se začínají také tvořit záznamy o probíhajícím letu. Když během této fáze poklesne rychlost a otáčky motoru pod rozhodovací hranici je fáze vzletu zrušena a přechází se zpět do fáze na zemi. Tento případ nastane když se z ultralehkým letounem rychle pojíždí nebo byla prováděna motorová zkouška.
- **Letoun je ve vzduchu:** Letoun je ve stavu letu, pokud je jeho rychlost přes stanovenou rozhodující hranici. Během této fáze jsou průběžně tvořeny záznamy o probíhajícím letu. Otáčky se zde nepoužívají k rozhodnutí o stavu systému protože se mohou během letu měnit od volnoběhu do maximálního výkonu podle aktuální letové situace. Pokud má tedy systém dostupný jen údaj o otáčkách motoru, tak záznam letu skončí vypnutím systému.
- **Fáze přistávání:** Tato fáze nastává, když rychlost poklesne pod stanovenou rozhodovací hranici. Začátkem této fáze je pozastaveno tvoření záznamů o probíhajícím letu a čeká se zda bude pokles rychlosti přítomný déle než je stanovený rozhodovací čas. Když je tomu tak, je let považován za ukončený, vytvoří se záznam o letu do databáze a pokračuje se do stavu na zemi. Když rychlost v tomto stavu opět přesáhne rozhodovací hranici, pokračuje se dál ve fázi letu.



Obrázek 5.12: Diagram přechodů mezi stavy rozhodovacího systému

Posledním krokem bylo zajištění, aby zařízení pracovalo zcela autonomně a vyžadovalo jen minimální zásahy člověkem, například pro údržbu. Pro splnění tohoto problému jsem zvolil mechanismus operačního systému Linux, zvaný Servisy¹¹. Servis umožňuje nastavit

¹¹<https://www.linux.com/news/introduction-services-runlevels-and-rcd-scripts/>

automatické spuštění programu na pozadí. Poskytuje mu také možnost nastavení, kdy a jak se má který program spouštět. Pro zajištění nepřetržitého běhu programů jsem tedy vytvořil dva servery, které programy spustí po načtení potřebných dat v operačním systému a také je automaticky restartují, pokud by se nečekaně ukončily. Tímto je zajištěno, že zařízení dokáže pracovat zcela automatizovaně a vždy bude poskytovat svoji funkčnost uživateli, když bude zapnuté. Podrobný popis jednotlivých částí implementace softwaru vestavěného zařízení je uveden v příloze B.

5.6 Popis implementace softwaru mobilní aplikace

Pro tvorbu mobilní aplikace přístupující k datům na vestavěném zařízení jsem zvolil vývojové prostředí Android Studio¹², které je založené na IntelliJ IDEA. Toto vývojové prostředí se jeví jako nejlepší volba z důvodu, že toto prostředí je vyvíjeno společností Google, která také vede vývoj operačního systému Android. Veškeré nástroje jsou tedy na tuto práci uzpůsobeny a nabízí jednoduchou a intuitivní implementaci.

Pro programování aplikací v Android Studiu lze použít hned několik programovacích jazyků. Podporovány jsou jazyky Java, Kotlin a C++. Pro vývoj aplikace jsem si nakonec zvolil programovací jazyk Kotlin, který je od roku 2019¹³ preferovaným jazykem pro vývoj aplikací pro operační systém Android.

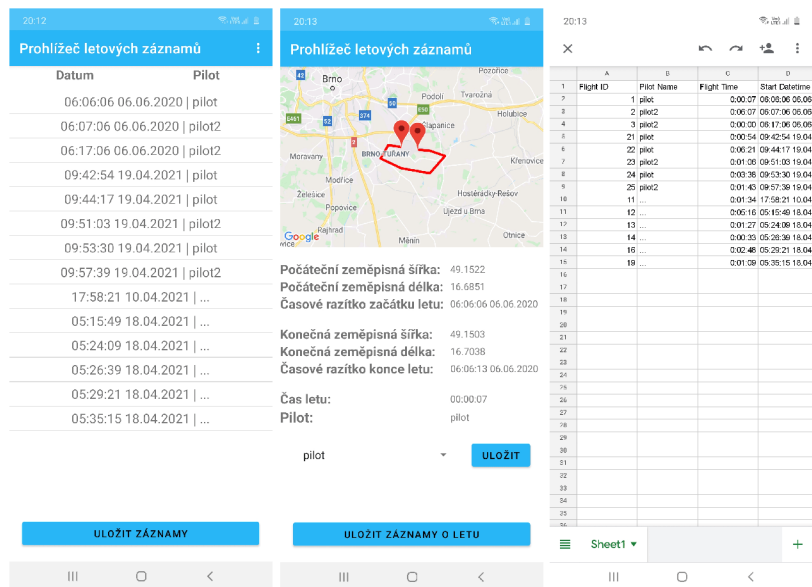
Pro aplikaci byl navrhnut systém zprávy uživatelů který umí rozlišovat dva rozdílné typy uživatelů, majitele letadla a piloty. Majitel letadla má přístup ke všem záznamům ve vestavěném zařízení a může je editovat, také jako jediný může tvořit nové piloty a měnit jejich údaje. Naproti tomu piloti vidí jen svoje data a lety bez pilotů, ke kterým se mohou přiřadit pokud si jsou jistí že to letěli oni a jen se zapomněli navolit na vestavěném zařízení. Dále mohou piloti editovat svoje osobní údaje. Role majitele letadla je vytvořená tak, že se nepovažuje za pilota, a není tedy možné tohoto uživatele navolit jako pilota ke konkrétnímu letu, pro zaznamenávání letových záznamů si musí majitel letadla vytvořit samostatný účet pilota.

Výsledná aplikace se skládá z následujících aktivit kde každá umožňuje uživateli specifickou funkčnost (ukázka vzhledu aplikace je vidět na obrázku 5.13):

- **Select device:** slouží k výběru Bluetooth zařízení, ke kterému se chce uživatel připojit,
- **Login:** slouží uživateli k přihlášení pod svým uživatelským jménem a heslem,
- **Flights:** zobrazuje seznam letů přihlášeného uživatele a lety bez přiřazeného pilota,
- **Flight detail:** zobrazuje uživateli podrobnější informace o konkrétním letu. který si vybere v aktivitě Flights,
- **Edit user:** slouží uživateli ke změně jeho přihlašovacích údajů. Pilot zde může změnit také svůj znak,
- **Edit pilots:** slouží majiteli letadla pro správu účtů jednotlivých pilotů,
- **Edit or save pilot:** slouží majiteli letadla ke změně přihlašovacích údajů vybranému pilotovi.

¹²<https://developer.android.com/studio>

¹³[https://cs.wikipedia.org/wiki/Kotlin_\(programovac%C3%AD_jazyk\)](https://cs.wikipedia.org/wiki/Kotlin_(programovac%C3%AD_jazyk))



Obrázek 5.13: Ukázka mobilní aplikace a stahování záznamů

Podrobný popis implementace softwaru mobilní aplikace je uveden v příloze B.

5.7 Experimentální montáž vestavěného zařízení

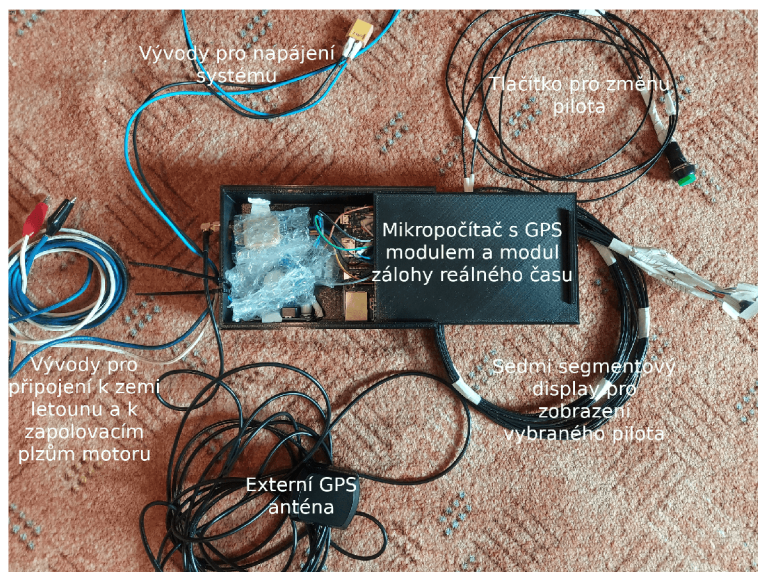
V následující sekci je popsán způsob, jakým se má systém namontovat do ultralehkého letounu. U každé montáže je potřeba vyřešit tyto záležitosti: možnosti napájení, uložení systému nebo připojení systému k otáčkoměru letadla. Pro experimentální montáž systému byl zvolen jednoduchý dvoumístný ultralehký letoun (viz obrázek 5.14) s motorem Rotax 912, který bude následně použit i na testování systému (viz sekce 5.8).



Obrázek 5.14: Testovací letoun s provedenou experimentální montáží vestavěného zařízení

Vestavěné zařízení potřebuje přístup k napájení a ke kabelu přicházejícímu do otáčkoměru letadla pro možnost snímání těchto pulzů. Nejvýhodnější je tedy zařízení umístit za panel s letovými instrumenty, kde bude jednoduché oba tyto spoje realizovat. Dále je nutné veškerý hardware uložit do boxu, ve kterém půjde jednoduše do letadla vložit a bude

izolovaný od okolních zařízení. Vytvořil jsem tedy jednoduchý 3D model krabičky, který lze vytisknout na 3D tiskárně. Tento model má v sobě dostatek místa pro veškeré použité komponenty a potřebné vývody pro vyvedení venkovních spojů (viz obrázek 5.15).



Obrázek 5.15: Ukázka uložení komponent s venkovními spoji

Dále je nutné napojit všechny potřebné venkovní spoje na vestavěné zařízení. V testovacím letadle nebyl problém vyřešit připojení k napájení a k pulzům pro otáčkoměr, otáčkoměr tohoto konkrétního letadla má totiž rovnou připojený 12 V vstup pro toto zařízení (viz obrázek 5.16). Stačilo se tedy připojit všemi potřebnými kabely na toto zařízení. U ostatních ultralehkých letounů může toto spojení vyžadovat vytvoření separátních spojů k potřebným kabelům dle schématu zapojení letadla.



Obrázek 5.16: Pohled na zadní stranu přístrojové desky testovacího ultralightu

Také je potřebné vybrat pozici GPS antény, kterou jsem v testovacím letadle umístil hned nad přístrojovou desku, kde je dostatečný signál přes přední sklo letadla. Při mon-

táži je možné díky 5m kabelu tuto pozici změnit pro dosažení lepšího signálu, možností je třeba montáž na křídlo letadla. Nakonec bylo nutné vyvést indikační prvek s tlačítkem pro možnost volby aktuálně letícího pilota. Pro finální montáž by bylo vhodné udělat malé díry do přístrojového panelu a prvky tam zalepit nebo vytvořit další krabičku, která by tyto prvky obsahovala a byla by přidělaná po okrajích přístrojové desky. V testovacím letadle jsem prvky přilepil na okraj přístrojové desky, pro účely testování to bylo dostačující.

Kromě pevné montáže systém umožňuje také montáž bez nutnosti jakéhokoli zásahu do systému letadla. Možné je někde umístit krabičku s komponentami a tu napájet 12 V autozásuvkou, která se ve většině ultralehkých letounů běžně nachází. Pro účely snímání letu pak budou používány jen informace z GPS modulu, může tedy nastat, že při špatném GPS signálu nebudou výsledná data dobrá.

5.8 Testování systému

Testováním systému jsem ověřoval zda systém splňuje specifikaci a obecně očekávanou funkcionalitu. Testy základní funkčnosti jednotlivých programů jsem prováděl už během samotné implementace.

Hlavní částí celého testování bylo ověřit program automatického zaznamenávání letů, u kterého bylo potřebné otestovat nejen přítomnost chyb v kódu, ale i jak kvalitní jsou data přijímaná programem od snímače otáček a GPS modulu. Pokud by byla data špatná, nemohl by se zbytek systému použít.

Pro otestování systému jsem vytvořil následující testy:

- Otestování kvality napájení vestavěného zařízení stepdown měničem,
- Ověření připojení mobilní aplikace k vestavěnému zařízení a přenosu dat mezi nimi,
- Ověření přesnosti měření otáček motoru,
- Otestování rychlosti zachycení platného GPS signálu,
- Ověření dat přijímaných od GPS modulu,
- Ověření zaznamenávacího programu jen na datech od GPS modulu,
- Ověření platného uložení dat při tvrdém vypnutí systému,
- Ověření platného zaznamenání letu v ultralightu,
- Ověření provedení záznamu letu jen z údajů o otáčkách motoru.

Zde vypsání testy byly vytvořeny s ohledem na testování základních principů funkčnosti experimentálního zařízení. Pokud by měl být systém reálně nasazený, bylo by potřeba jich provést mnohem více. Příkladem mohou být testy elektromagnetické kompatibility, elektromagnetického vyzařování, elektrických spojů a také dlouhodobé testování na více uživatelích a více ultralehkých letounech.

V následujících sekcích jsou popsány testy, které byly provedeny a jejich výsledky.

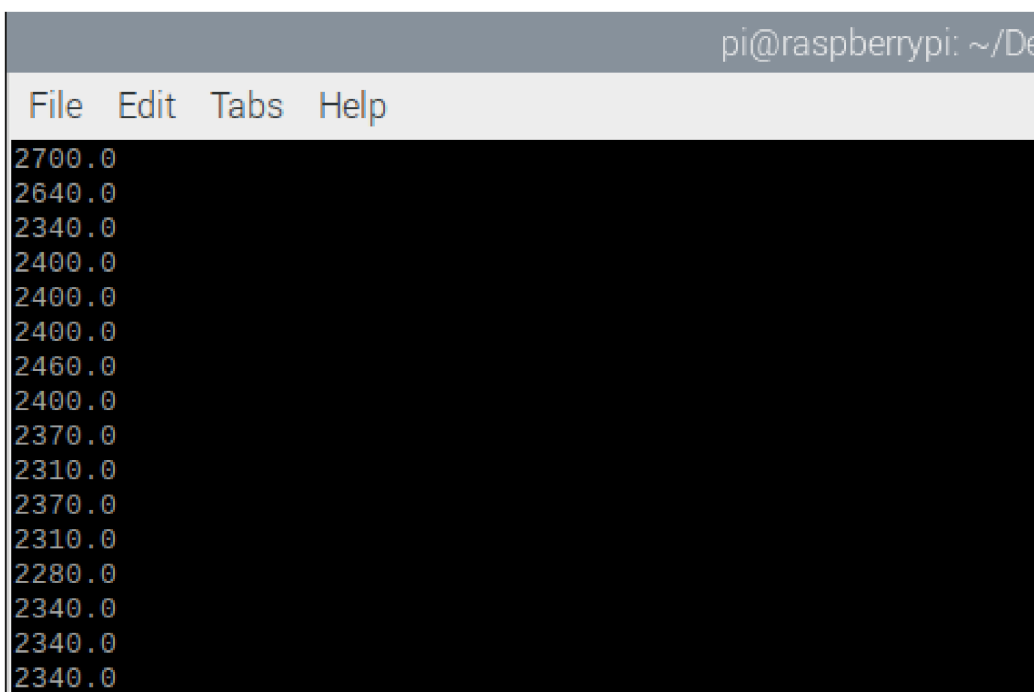
Testování připojení mobilní aplikace k vestavěnému zařízení a přenosu dat mezi nimi

Pro testování mobilní aplikace a programu pro přenos dat přes Bluetooth jsem vytvořil databázi s imaginárními daty. Jako testovací mobilní zařízení jsem použil Samsung Galaxy S10 Lite.

Během provádění testů se podařilo spárovat mobilní zařízení s vestavěným zařízením a podařilo se úspěšně přihlásit. Následné prohlížení a editace záznamů funguje dle očekávání.

Testy měření otáček motoru

U měření počtu otáček motoru za minutu bylo nutné ověřit, zda program dává stejnou hodnotu jako ukazuje otáčkoměr ultralehkého letadla. Kdyby byla tato hodnota příliš rozdílná od hodnoty udávané otáčkoměrem, tak by ji bylo těžké použít pro rozhodování o změně stavu letadla.



```
pi@raspberrypi: ~/De
File Edit Tabs Help
2700.0
2640.0
2340.0
2400.0
2400.0
2400.0
2460.0
2400.0
2370.0
2310.0
2370.0
2310.0
2280.0
2340.0
2340.0
2340.0
```

Obrázek 5.17: Ukázka měření otáček

Pro otestování jsem vytvořil jednoduchý program, do kterého jsem přepsal logiku zaznamenávání ze zaznamenávače letu. Jak lze vidět z obrázku 5.17, výsledné hodnoty otáček motoru jsou celkem stabilní a při porovnání s reálnými hodnotami, které udává otáčkoměr, se hodnoty liší o maximálně 100 otáček. Lze tedy říci, že pro zaznamenávání letu jsou tyto údaje dostatečné.

Testy GPS modulu

Pro použití dat přijímaných GPS modulem k rozhodování o stavu ultralehkého letounu je nejdříve nutné zachytit samotný GPS signál. Bez zachyceného GPS signálu se tento modul pro rozhodovací logiku nedá použít. První testy, které jsem provedl na tomto modulu, se tedy zabývají rychlostí, jakou dokáže modul zachytit platná GPS data. V následujících testech bylo provedeno vždy 5 měření pro každou lokaci a byla zaznamenána maximální zaznamenaná hodnota.

| Místo | Rychlost zachycení |
|---------------------------|--------------------|
| V domě u okna | Signál nezachycen |
| V zastavěné oblasti | max 30 min |
| Na otevřeném prostranství | max 15 min |

Tabulka 5.2: Rychlost zachycení platných dat GPS modulem na interní anténě

Jak lze z dosažených výsledků v tabulce 5.2 vyzorovat, tak využití interní antény tohoto modulu je pro použití v tomto systému zcela nedostatečné. Není možné, aby ultralehký letoun čekal příliš dlouho, než by bylo možné zaznamenávat údaje o trase letu. Jedinou možností je tedy použít externí anténu.

| Místo | Rychlost zachycení |
|---------------------------|--------------------|
| V domě u okna | max 5 min |
| V zastavěné oblasti | max 1 min |
| Na otevřeném prostranství | max 30 sek |

Tabulka 5.3: Rychlost zachycení platných dat GPS modulem na externí anténě

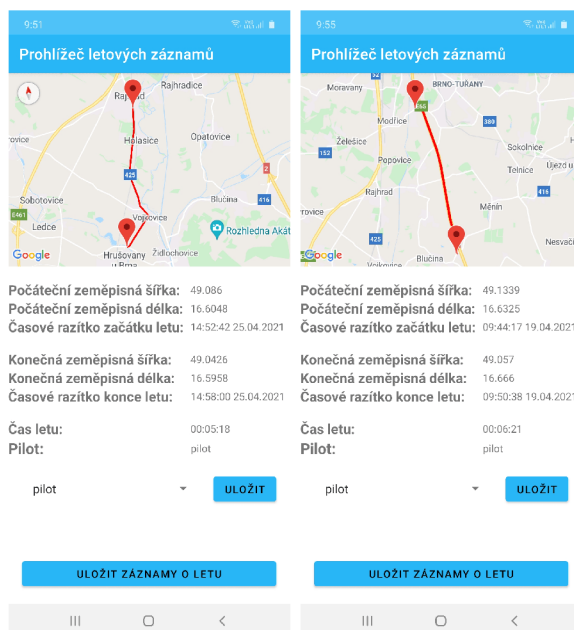
Rychlosti dosahované externí anténou (viz tabulka 5.3) jsou přívětivé a lze tedy očekávat, že u většiny letů by neměl být problém zachytit platný signál před odstartováním ultralehkého letounu a tím pádem bude let zaznamenán i s jeho trasou. Běžná příprava ultralehkého letounu k letu totiž trvá několik minut. Trasa by nemusela být zaznamenána u letů ve kterých je před nastartováním už ultralehký letoun plně připravený k letu. Pak by odstartování trvalo jen desítky sekund a mohlo by se stát, že se platný GPS signál před odstartováním nepovede zachytit.

Pro úplnost by bylo vhodné udělat testy v samotném ultralehkém letounu. Tyto testy nebyly provedeny z důvodu nedostatku času, který jsem měl na testování systému v letounu.

Testy tvorby záznamů s přiřazením pilota

Po ověření funkčnosti GPS modulu bylo nutné ověřit, zda dobře funguje samotný program zaznamenávání letů. K tomuto testu jsem použil automobil, ve kterém jsem mohl ověřit průchody mezi jednotlivými stavy rozhodovací logiky jen na základě rychlosti. Při stabilně zvýšené rychlosti se měl vytvořit záznam o proběhlém letu.

Po provedení několika testů (viz obrázek 5.18) se ukázalo, že rozhodovací logika funguje správně a tvoří přesné trasy s validními informacemi a správně navoleným pilotem.



Obrázek 5.18: Záznamy pořízené v autě

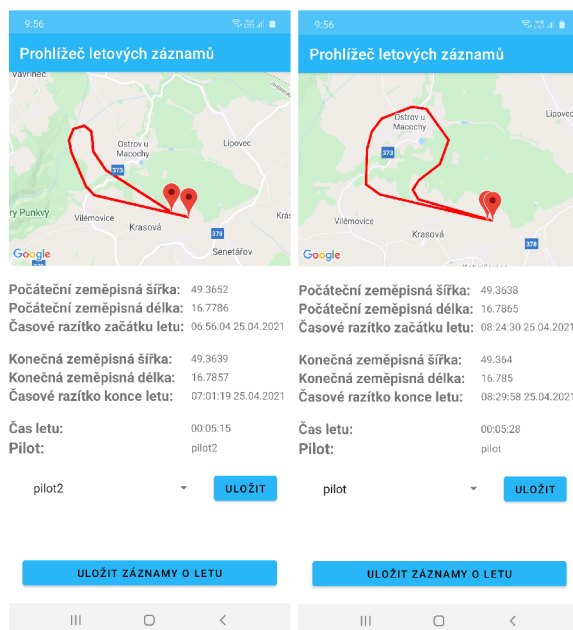
Testování tvrdého vypnutí systému

Za pomoci automobilu a GPS modulu jsem také ověřoval, zda je možné systém natvrdo vypnout během tvoření záznamu letu což je běžná věc když je let zaznamenáván jen za pomoci otáček motoru. Tato funkčnost se také ukázala jako funkční. Vytvořené logy z provedeného letu byly správně uloženy v pomocném souboru a při následném naběhnutí systému se správně přečetli a uložili do interní databáze zařízení.

Testování systému v letadle a určení přesnosti zaznamenaných letů

Nakonec bylo nutné otestovat, jestli systém jako celek funguje správně, když je namontovaný do ultralehkého letounu a jsou na něj připojeny všechny komponenty. Za tímto účelem bylo provedeno několik testovacích letů, ze kterých se vyhodnocovaly získané údaje.

Z provedených letů (viz obrázek 5.19) vyplývá, že systém správně zachytává všechny stavy letu a také správně zaznamenává trasu. Jedinou nesrovnalostí je čas doby trvání letu, který je díky času potřebnému k zaznamenání, že nastal let, a stanovenému rozptylu mezi tvorbou jednotlivých log záznamů vždy o něco menší než reálná hodnota, kterou naměřil pilot letadla. Průměrně tato hodnota byla menší o přibližně 45 sekund. Pro účely přesného rozpočítávání provozních nákladů by byly tyto hodnoty vždy o několik desítek sekund chybné. Stále je však výhodné, že jsou záznamy v rámci mezí přesné a tvoří se vždy automaticky v elektronické podobě.



Obrázek 5.19: Záznamy provedených letů

V testování systému nebyly provedeny všechny možné testy, kterým by šlo systém podrobit. Testy jsem zvolil tak, aby ukázaly zda fungují nejzákladnější prvky systému. Jak lze z provedených testů vidět, hlavní prvky systému fungují dle specifikace.

5.9 Návrhy na zlepšení

Vytvořený systém je koncipován značně otevřeně, aby si ho mohl každý uživatel přizpůsobit dle svých potřeb a pro jeho konkrétní ultralehký letoun. Systém je hodně přizpůsobitelný, avšak samotná montáž je pro každý letoun celkem podobná a jednoduchá. Kromě nejednoznačností v montáži však existuje pár obecných vylepšení, která by mohla pomoci systému k jeho lepší funkčnosti nebo pár rozšíření, které by přinesly užitek uživatelům systému.

Použití slabšího modelu Raspberry Pi

Raspberry Pi 3B+ je výkonný minipočítač a jak se ukázalo, tak vytvořený systém nikdy nevyužije jeho celý dostupný výkon. Primárním důvodem proč byl původně vybrán model B+ je jeho jednodušší využití pro experimentování. Díky jeho výkonu tedy nebyl problém psát kód přímo na něm a během testů šlo také rovnou pozorovat stavy systému. Pro reálné nasazení je možnost ho nahradit jeho menší a slabší verzí Raspberry Pi Zero WH, která má jen snížený výkon, ale jinak poskytuje všechny potřebné funkčnosti pro výsledné vestavěné zařízení. Nahrazení minipočítače by způsobilo zmenšení vyžadovaného prostoru a snížení pořizovací ceny. Průměrná cena Raspberry Pi Zero WH je ke dni 14.04.2021 500 Kč.

Výkonejší GPS modul

Pokud by v budoucnu byly dostupné kvalitnější GPS moduly, mohly by vyměnit stávající GPS modul za výkonnější. Lepším GPS modulem by bylo možné zvednout schopnost zaznamenávání letů systémem jen za pomoci tohoto modulu. Možná by pak nebylo potřeba zaznamenávat otáčky motoru pokud by modul dokázal vždy do pár sekund zachytit validní signál.

Zlepšení mobilní aplikace

Mobilní aplikace byla od základu vyvíjena, aby dokázala poskytovat nejzákladnější potřebnou funkčnost. Existuje tedy celá řada vylepšení, která by mohla jejím uživatelům pomoci. Mezi primární vylepšení by šlo zařadit vylepšení designu, nebo pokročilé filtry, které by umožnily lepší hledání v seznamu letů zobrazených na hlavní stránce po přihlášení uživatele.

Automatické ukládání záznamů přes Wi-Fi

Tím, že je k přístupu k datům na vestavěném zařízení využíváno Bluetooth, je značně omezený čas a prostor, kdy si uživatelé svoje záznamy mohou prohlédnout. V aktuální podobě mohou uživatelé prohlížet data jen, když je ultralehký letoun v provozu a jsou od něj přibližně 5 metrů. Toto omezení by šlo eliminovat pokud by v hangáru s ultralehkým letounem byla přístupná síť Wi-Fi. Při této skutečnosti by mohlo být nastavené vestavěné zařízení tak, aby se automaticky připojovalo a odesílalo své záznamy na server. Tato data by si pak mohli uživatelé kdekoli ručně prohlédnout nebo by se mohla upravit mobilní aplikace a bylo by možné využívat data ze serveru.

Automatická volba parametrů pro rozhodovací systém

Nutnost manuálního zadání parametrů pro rozhodovací systém do každého ultralightu by také šla omezit. Možné by bylo využít strojového učení které by se naučilo jak má tyto pa-

parametry volit. Pak by bylo možné, aby majitel nainstaloval systém do svého stroje a provedl testovací let, při kterém by se potřebné parametry sami nastavily.

Automatická rekognice pilota

Nejzajímavějším rozšířením z pohledu uživatelů by však byla jejich automatická rekognice při jejich vstupu do letadla. Tímto rozšířením by se eliminovala nutnost manuálního navolení pilota, které nyní musí uživatelé provádět při vstupu do letadla. Z podstaty věci také toto rozšíření není nijak lehké na implementaci a existuje také spousta možností, jak toto rozšíření implementovat. Nejméně náročným způsobem by bylo ověřování telefonu pomocí Bluetooth technologie, u této metody by však mohl nastat problém pokud by v ultralehkém letounu letěli dva piloti a oba měli zapnuté Bluetooth. Poté by se musel vyřešit rozhodnutí, koho zapsat jako operujícího pilota.

Pokud by nebyla proveditelná automatická rekognice, je možné vytvořit logiku, která by dokázala odhadnout, kterého pilota má k letu přiřadit na základě předem pořízených záznamů. Mohlo by se stát, že existují dva záznamy, které byly provedeny v těsné blízkosti za sebou a jeden z těchto záznamů má pilota přiřazeného a druhý ne. Program může na základě těchto záznamů usoudit, že je provedl stejný pilot, protože byly provedeny v těsné blízkosti za sebou. Nutné by však bylo, aby tito, automaticky přiřazení piloti, byli ukládáni zvlášť od reálně navolených pilotů pro možnost jednoduchého nalezení chyb v přiřazení, které by bylo následně nutné opravit.

Kapitola 6

Závěr

Cílem práce bylo vytvořit jednoduchý systém, který dokáže automaticky vytvářet záznamy o využití ultralehkých letounů a následně je prezentovat uživatelům pro jejich další zpracování. Tento cíl se podařilo splnit.

Před započítím práce na samotném systému bylo nutné prostudovat technologie, které je možné použít na tvorbu takového systému. Poté je probrán princip provozování ultralehkých letounů a jak se aktuálně měří zaznamenané lety. Dále byla vypracována specifikace udávající jak má systém fungovat. Následně byl na vytvořen podrobný návrh systému. Jako řídicí prvek systému byl vybrán jednodeskový počítač Raspberry Pi, který je umístěn v ultralehkém letounu s připojenými periferiemi. Pro zobrazení a editaci záznamů byla vytvořena aplikace na mobilní telefony s operačním systémem Android, která s vestavěným zařízením komunikuje pomocí Bluetooth.

Po vytvoření kompletního návrhu byly implementovány programy pro vestavěné zařízení v programovacím jazyce Python. První program má na starosti komunikaci z mobilní aplikací. Druhý se stará o čtení informací z nejrůznějších periferií, pomocí kterých automaticky určuje, kdy ultralehký letoun letí. Dále byla implementována aplikace pro Android telefony pomocí programovacího jazyka Kotlin, která umožňuje práci s daty na vestavěném zařízení. Závěrem byla veškerá funkčnost otestována deseti cestami v automobilu a dvěma lety v ultralehkém letadle, ze kterých byla vyhodnocena celková použitelnost systému.

Výsledný systém nabízí majitelům i pilotům ultralehkých letounů jednoduchou i levnou (celková cena systému byla částce 3000 Kč) platformu, pomocí které mohou monitorovat, kdy jaký ultralehký letoun byl ve vzduchu a kdo s ním letěl. Systém umožňuje správu uživatelů a prohlížení zaznamenaných letů na mobilním zařízení. Určení stavu ultralehkého letounu je prováděno na základě GPS dat a hodnoty aktuálních otáček motoru. Systém podporuje 20 rozdílných pilotů a umožňuje pro všechny dohromady zaznamenat 5000 záznamů o letech.

Práci na tomto projektu jsem získal spoustu nových informací, protože bylo nutné nastudovat data z různých odvětví, která byla následně použita pro tvorbu výsledného funkčního celku. Navrhovaný a otestovaný systém pro ultralehké letouny pracuje dle očekávaných specifikací a je možné ho reálně použít.

V budoucnu by mohl systém projít řadou vylepšení. Mobilní aplikace může být rozšířena o další funkce, které by byly pro uživatele důležité, může být vylepšen celkový vzhled aplikace, vytvoření nové krabice s přehlednějším uložením všech komponent, automatické ukládání dat na server přes Wi-Fi nebo automatickou rekognici pilota pro jednotlivé lety.

Literatura

- [1] *Pravidla provozu sportovních létajících zařízení* [online]. Letecká amatérská asociace, Czech Republic, 2010 [cit. 2021-03-21]. Dostupné z: <http://www.laacr.cz/SiteCollectionDocuments/predpisy/UL1.pdf>.
- [2] *Požadavky letové způsobilosti SLZ, Ultralehké letouny řízené aerodynamicky* [online]. 1. vyd. Letecká amatérská asociace, Czech Republic, 2019 [cit. 2021-03-21]. Dostupné z: http://www.laacr.cz/SiteCollectionDocuments/predpisy/UL2%20%C4%8D%C3%A1st%20I_26.3.2019.pdf.
- [3] *Návod k obsluze, M527 - Počítadlo motohodin* [online]. Hadex, spol. s.r.o, Czech Republic, 2021 [cit. 2021-03-21]. Dostupné z: <https://www.hadex.cz/navody/m527.pdf>.
- [4] *ASA Flight Timer Instruction Manual* [online]. Aviation Supplies & Academics Inc., United Kingdom, 2021 [cit. 2021-03-21]. Dostupné z: https://www.asa2fly.com/Assets/SupportFiles/ProductManuals/TIMER-2_Manual-2007.pdf.
- [5] *Garmin G3X Pilot's Guide* [online]. Garmin International Inc., United States, 2019 [cit. 2021-03-21]. Dostupné z: https://static.garmin.com/pumac/190-01115-00_q.pdf.
- [6] BANERJI, S. a CHOWDHURY, R. S. *On IEEE 802.11: Wireless LAN Technology* [online]. International Journal of Mobile Network Communications & Telematics, Volume 3, Issue 4, AIRCC Publishing Corporation, India, 2013 [cit. 2021-03-21]. Dostupné z: https://www.researchgate.net/publication/248381994_On_IEEE_80211_Wireless_Lan_Technology.
- [7] BANZI, M. *Getting Started with Arduino*. 2. vyd. Maker Media Inc., Switzerland, 2011. ISBN 1449309879.
- [8] CHAUDHARI, H. *Raspberry Pi Technology: A Review* [online]. North Maharashtra University, India, 2015 [cit. 2021-03-21]. ISSN 2394-3343. Dostupné z: <http://www.ijiere.com/FinalPaper/FinalPaper201532874333741.pdf>.
- [9] CHELLAPPA, M. R., MADASAMY, S. a PRABAKARAN, R. *Study on ZigBee technology* [online]. Anna University of Technology, India, 2011 [cit. 2021-03-21]. Dostupné z: https://www.researchgate.net/publication/261497749_Study_on_ZigBee_technology.
- [10] DOBKIN, R. R., MORGENSHEIN, A., KOLODNY, A. a GINOSAR, R. *Parallel vs. Serial On-Chip Communication* [online]. VLSI Systems Research Center, Israel, 2017 [cit. 2021-03-21]. Dostupné z: https://www.researchgate.net/publication/221031553_Parallel_vs_serial_on-chip_communication.

- [11] HALFACREE, G. *The Official Raspberry Pi Beginner's Guide 2018: How to use your new computer*. 1. vyd. Raspberry Pi Press, United Kingdom, 2018. ISBN 1912047683.
- [12] HUGHES, J., NUTTALL, B., SCHELLER, A., BEALE, C., CLAY, L. et al. *GPIO* [online]. Raspberry Pi Foundation, United Kingdom, 2021 [cit. 2021-03-21]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio/>.
- [13] KASWAN, K. S., SINGH, S. P. a SAGAR, S. *Role Of Arduino In Real World Applications* [online]. International Journal of Scientific & Technology Research, Volume 9, Issue 01, IJSTR, India, 2014 [cit. 2021-03-21]. Dostupné z: <http://www.ijstr.org/final-print/jan2020/Role-Of-Arduino-In-Real-World-Applications-.pdf>.
- [14] KUMARAN, D. N. M. a SONYA, R. *Principles of Wireless Communications*. 1. vyd. Scholars-Press Publication, Mauritius, 2017. ISBN 978-620-2-30432-0.
- [15] LOUIS, L. *Working Principle of Arduino and Using it as a Tool for Study and Research* [online]. International Journal of Control, Automation, Communication and Systems, Volume 1, Issue 2, Springer Nature, India, 2016 [cit. 2021-03-21]. Dostupné z: https://www.researchgate.net/publication/326316390_Working_Principle_of_Arduino_and_Using_it_as_a_Tool_for_Study_and_Research.
- [16] MAKSIMOVIC, M., VUJOVIC, V., DAVIDOVIĆ, N., MILOSEVIC, V. a PERISIC, B. *Raspberry Pi as Internet of Things hardware: Performances and Constraints* [online]. University of East Sarajevo, Serbia, 2014 [cit. 2021-03-21]. Dostupné z: https://www.researchgate.net/publication/272175660_Raspberry_Pi_as_Internet_of_Things_hardware_Performances_and_Constraint.
- [17] NANDA, U. a PATTNAIK, S. *Universal Asynchronous Receiver and Transmitter (UART)* [online]. Silicon Institute of Technology, India, 2016 [cit. 2021-03-21]. Dostupné z: https://www.researchgate.net/publication/308988751_Universal_Asynchronous_Receiver_and_Transmitter_UART.
- [18] NANDI, G. a TSIRA, R. V. *Bluetooth Technology: Security Issues and Its Prevention* [online]. International Journal of Computer Applications in Technology, Volume 5, Issue 5, Inderscience Enterprises Ltd., India, 2014 [cit. 2021-03-21]. ISSN 2229-6093. Dostupné z: https://www.researchgate.net/publication/268018259_Bluetooth_Technology_Security_Issues_and_Its_Prevention.
- [19] NAYYAR, A. a PURI, V. *Raspberry Pi-A Small, Powerful, Cost Effective and Efficient Form Factor Computer: A Review* [online]. KCL Institute of Management and Technology, India, 2015 [cit. 2021-03-21]. ISSN 2277 128X. Dostupné z: https://www.researchgate.net/publication/305668622_Raspberry_Pi-A_Small_Powerful_Cost_Effective_and_Efficient_Form_Factor_Computer_A_Review.
- [20] OSISIOGU, U. *Utilization of Serial Communication in Arduino* [online]. Nnamdi Azikiwe University, Nigeria, 2015 [cit. 2021-03-21]. Dostupné z: https://www.researchgate.net/publication/327285060_Seminar_Paper_on_Serial_Communication.
- [21] PATEL, S., TALATI, P. a GANDHI, S. *Design of I2C Protocol* [online]. International Journal of Technical Innovation in Modern Engineering & Science, Volume 5, Issue 3,

- IJTIMES, India, 2019 [cit. 2021-03-21]. ISSN 2455-2585. Dostupné z: https://www.researchgate.net/publication/332142672_Design_of_I2C_Protocol.
- [22] POLSANI, P., BRAHMAIAH, V. P. a SAI, Y. P. *Design & Verification of Serial Peripheral Interface (SPI) Protocol* [online]. International Journal of Recent Technology and Engineering, Volume 8, Issue 6, Blue Eyes Intelligence Engineering and Sciences Publication, India, 2020 [cit. 2021-03-21]. ISSN 2277-3878. Dostupné z: <https://www.ijrte.org/wp-content/uploads/papers/v8i6/F7356038620.pdf>.
- [23] STALLINGS, W. *Data and Computer Communications*. 10. vyd. Pearson Education Inc., United States, 2013. ISBN 0133506487.
- [24] WU, E. *10 Awesome Raspberry Pi HATs You Need to Try* [online]. Seeed Studio, Shenzhen, 2019 [cit. 2021-03-21]. Dostupné z: <https://www.seeedstudio.com/blog/2019/02/12/awesome-hats-shields-for-raspberry-pi-you-should-try/>.
- [25] WU, E. *Best 20+ Arduino Shields You Should Try* [online]. Seeed Studio, Shenzhen, 2019 [cit. 2021-03-21]. Dostupné z: <https://www.seeedstudio.com/blog/2019/09/25/best-20-arduino-shields-you-should-try/>.
- [26] WU, E. *Meet The Brand New Raspberry Pi 4 8GB* [online]. Seeed Studio, Shenzhen, 2020 [cit. 2021-03-21]. Dostupné z: <https://www.seeedstudio.com/blog/2020/05/28/meet-the-brand-new-raspberry-pi-4-8gb-ram/>.
- [27] ŠEBESTA, J. *Globální navigační systémy*. Czech Republic, 2012. Dizertační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií VUT. Dostupné z: http://www.urel.feec.vutbr.cz/~sebestaj/RAR/literatura/Globalni_navigacni_systemy.pdf.

Příloha A

Obsah přiloženého paměťového média

- `3d_print/` - Složka obsahuje díly ochranné krabičky vestavěného zařízení určené pro 3D tisk.
- `mobile_application/` - Složka obsahuje projekt Android Studia, ve kterém je vytvořena mobilní aplikace.
- `raspberry_pi/` - Složka obsahuje soubory potřebné k instalaci systému na jednodeskový počítač Raspberry Pi.
- `text_src/` - Složka obsahuje zdrojové soubory Latexu pro tvorbu textu práce.
- `system_testing/` - Složka obsahuje fotky a video, které byly pořízeny během testování systému na letišti v Kotvrdovicích.
- `raspberry_pi/README.txt` - Soubor obsahuje návod jak zprovoznit programy prováděné na Raspberry Pi.
- `mobile_application.apk` - Instalační soubor mobilní aplikace.

Příloha B

Podrobný popis implementace systému

Tato příloha obsahuje podrobný popis implementace vestavěného zařízení a přístupové mobilní aplikace

Podrobný popis implementace softwaru vestavěného zařízení

V následujících sekcích je podrobně popsána implementace jednotlivých částí, ze kterých je software vestavěného zařízení zhotoven.

Popis struktury Databáze uchovávající data vestavěného zařízení

Databáze slouží vestavěnému zařízení pro uchování všech záznamů o pilotech a letových záznamech ultralehkého letounu. Data v databázi jsou rozdělena do tří tabulek User, Flight a GPS_Log.

Tabulka User uchovává data o uživatelích využívajících systém. Ke každému uživateli je uchováváno jeho přihlašovací jméno, heslo, role a symbol. Role odlišuje dva podporované typy uživatelů potřebné pro rozdělení práv v mobilní aplikaci. Uživatelem může být majitel letadla, pak má přístup k veškerým datům a editaci, nebo pilot, který používá aplikaci jen pro prohlížení svých vlastních letových záznamů. Poslední uchovávanou hodnotou je jednoznakový symbol, který slouží pilotům pro jejich identifikaci na vestavěném zařízení před začátkem letu. Pro majitele letadla je tato hodnota vždy prázdná protože ho nejde použít jako pilota.

Tabulka Flight slouží pro namapování jednotlivých logovacích záznamů na konkrétního pilota, což představuje jeden konkrétní let. Odkaz na pilota může být prázdný, pokud během letu nebyl žádný stanoven, pak je tento let uložen bez této hodnoty a je očekáváno, že se k němu pilot později přiřadí přes mobilní aplikaci.

V tabulce GPS_Log jsou uloženy logovací záznamy, které vytvořil logovací program při zaznamenání, že je ultralehký letoun ve stavu letu. Každý log obsahuje odkaz na let, ke kterému patří i časové razítko v UTC, kdy byl záznam pořízen. Každý log má také místo pro zeměpisnou šířku a délku, tyto údaje se však vyplňují jen pokud je navázané validní GPS spojení, jinak je tento údaj nedostupný a je možné tvořit záznamy jen podle měřiče otáček.

Databáze je uložena v souboru airplane_data.db, který se nachází ve stejné složce jako všechny ostatní soubory potřebné pro běh systému. Problémem s implementací této data-

báze je, že k ní přistupují dva programy současně. Může tedy nastat, že tabulku databáze používá jeden program pro editaci a druhý program by pak tuto tabulku nemohl použít. Z tohoto důvodu oba programy při jakékoli změně hned ukončují započatou transakci, aby byla tabulka zamčená na co nejmenší možný čas. Práce s databází v zaznamenávací letu byla také omezena jen na krátký okamžik před začátkem nebo po konci letu, kdy je potřeba uložit pořizovaná data do databáze. Lze tedy očekávat, že výskyt problému je velice málo pravděpodobný a nastane jen zřídka.

Implementace programu Bluetooth komunikace s mobilním zařízením

Tento program slouží k tomu, aby poskytl mobilní aplikaci zaznamenaná data o pilotech a jejich letech, která získává z interní databáze vestavěného zařízení. V programu bylo nutné vyřešit automatické ovládání Bluetooth modulu a následně schéma přenosu dat mezi vestavěným zařízením a mobilní aplikací.

Program je interně rozdělen do dvou hlavních částí. První část, která se provádí hned po spuštění, se stará o nastavení Bluetooth modulu. Bluetooth modul je potřebné nastavit tak, aby se s ním mohli uživatelé kdykoli spárovat a nebyl potřebný žádný zásah na straně vestavěného zařízení pro uskutečnění nových spojení. Tato funkčnost je důležitá pro nové uživatele, kteří si stáhnou mobilní aplikaci a chtějí se podívat na svá data bez nutnosti nějakého nastavování na interním zařízení. Pro dosažení této funkčnosti jsou prováděny příkazy příkazové řádky, které nastaví viditelnost zařízení a automatické párování přes `bluetoothctl`¹. Pro umožnění posílání zpráv přes Bluetooth modul v Pythonu bylo potřeba do Raspberry Pi doinstalovat knihovnu `pybluez`² a balíček `bluetooth`³, které tuto funkčnost umožňují.

Dále program obsahuje hlavní část celé komunikace, která přijímá zprávy od mobilní aplikace a odesílání výsledky dotazů zpět mobilní aplikaci. Tato část je vytvořena ze dvou vnořených nekonečných smyček, kde první z nich nastaví Bluetooth do serverového módu, který čeká na navázání jakéhokoli spojení. Po úspěšném navázání spojení program přechází do druhé nekonečné smyčky, ve které přijímá příkazy od mobilního zařízení, podle kterých pracuje s interní databází a posílá odpovědi stanovené vytvořeným přenosovým protokolem. Po zrušení navázaného spojení program dorazí opět na začátek první nekonečné smyčky, kde čeká na navázání nového spojení. Pokud by se s vestavěným zařízením spárovalo nějaké zařízení bez požadované aplikace, bude jen blokovat spojení pro další zařízení, žádná data se však nepřenese, protože by zařízení muselo začít komunikovat přes stanovený protokol, aby mohl program nějak odpovědět.

Následuje popis protokolu využívaného pro přenos dat. Komunikace je ve směru od mobilní aplikace k vestavěnému zařízení:

- **Send table User:** Požadavek na odeslání tabulky User z databáze. Odpověď je odesílána po řádcích. První řádek obsahuje číslo, které udává kolik záznamů se celkem přeneslo, další řádky obsahují záznamy ze sloupců tabulky oddělených znakem `~`. Každá hodnota je odesílána ve formátu `název sloupce:hodnota`.
- **Send table Flight:** Požadavek na odeslání tabulky Flight z databáze. Odpověď je odesílána po řádcích. První řádek obsahuje číslo, které udává kolik záznamů se

¹<https://www.linux-magazine.com/Issues/2017/197/Command-Line-bluetoothctl>

²<https://github.com/pybluez/pybluez>

³<https://packages.ubuntu.com/bionic/bluetooth>

celkem přeneše, další řádky obsahují záznamy ze sloupců tabulky oddělených znakem ~. Každá hodnota je odesílána ve formátu název sloupce:hodnota.

- **Send table GPS_Log:** Požadavek na odeslání tabulky GPS_Log z databáze. Odpověď je odesílána po řádcích. První řádek obsahuje číslo, které udává kolik záznamů se celkem přeneše, další řádky obsahují záznamy ze sloupců tabulky oddělených znakem ~. Každá hodnota je odesílána ve formátu název sloupce:hodnota.
- **Delete Flight ID:::** Tento příkaz slouží majiteli ultralehkého letounu pro mazání letů uložených na vestavěném zařízení. Program přijme ID záznamu ke smazání a smaže ho z databáze. Po úspěšném smazání záznamu odpovídá program zprávou OK.
- **Delete User ID:::** Tento příkaz slouží majiteli ultralehkého letounu pro mazání pilotů uložených na vestavěném zařízení. Program přijme ID pilota ke smazání a smaže ho z databáze. Po úspěšném smazání pilota odpovídá program zprávou OK.
- **Change Pilot Flight_ID::~Pilot_Name::~:** Příkaz slouží pro přiřazení pilota k letovému záznamu. Program spojí jméno pilota s logovými záznamy přes tabulku Flight. Úspěšné přiřazení letu pilotovi indikuje program odesláním zprávy OK.
- **Edit User ID::~Values Name::~Password::~Symbol::~:** Tento příkaz slouží pro editaci záznamů o uživateli v tabulce User. Povinná je vždy hodnota ID, ostatní hodnoty se mohou namíchat dle toho, co je u daného uživatele zapotřebí změnit. Úspěšná změna údajů uživatele je indikována mobilní aplikaci zprávou OK.
- **Create User Values Name::~Password::~Symbol::~:** Tento příkaz slouží majiteli ultralehkého letounu pro tvorbu nových účtů pro piloty. V každé zprávě jsou tedy obsaženy údaje potřebné pro vykonání této činnosti. Úspěšné přidání nového pilota do databáze na vestavěném zařízení je indikováno odesláním zprávy OK.
- Pokud vestavěné zařízení dostane jinou zprávu než je výše uvedeno, tak odpovídá zprávou Invalid action, která udává připojenému zařízení, že zpráva není podporována a čeká se na další příchozí zprávu.

Jak lze z jednotlivých příkazů protokolu poznat, tak celá komunikace funguje na principu klient-server. Komunikaci vždy začíná mobilní zařízení a pokud dostane vestavěné zařízení validní příkaz, tak jej provede. Pokud má příkaz nějaký výsledek, tak se vytvořená data odešlou mobilnímu zařízení. Dále je také vidět, že příkazy jsou formulovány běžnou řečí a není využíváno žádných specifických kódů. Tento styl jsem zvolil z důvodu, že příkazů není velké množství a také je tento způsob přehledný pro účely debugování⁴. Data v jednotlivých zprávách jsou oddělována speciálním znakem, zde ~. Tento způsob byl zvolen pro jednoduchou identifikaci částí zprávy pomocí regexu⁵.

Implementace programu zaznamenávání letů

Funkcionalita tohoto programu je zodpovědná za ovládání všech připojených komponent k vestavěnému zařízení a podle údajů z nich rozhoduje o aktuálním stavu ultralehkého letounu, zda je na zemi a nebo ve stavu letu, pomocí tohoto stavu jsou nakonec tvořeny záznamy uchovávací proběhlé lety. K vytvoření tohoto programu bylo zapotřebí vyřešit

⁴[https://cs.wikipedia.org/wiki/Lad%C4%9Bn%C3%AD_\(programov%C3%A1n%C3%AD\)](https://cs.wikipedia.org/wiki/Lad%C4%9Bn%C3%AD_(programov%C3%A1n%C3%AD))

⁵https://cs.wikipedia.org/wiki/Regul%C3%A1rn%C3%AD_v%C3%BDraz

komunikaci s každou komponentou, zpracování jejich dat, vytvořit rozhodovací logiku, která určuje, v jakém stavu je ultralehký letoun a nakonec vyřešit zaznamenávání těchto získaných dat pro účel tvorby jednotlivých záznamů o proběhlých letech.

Tento program je, stejně jako program pro Bluetooth komunikaci, rozdělen na dvě hlavní části. V první části se provádí inicializace všech potřebných proměnných důležitých pro další běh programu a inicializují se spojení mezi GPIO piny vestavěného zařízení a připojenými komponentami. Většina vytvořených proměnných zde slouží pro uchování globálních parametrů, nebo pro uložení hodnot, které slouží programu pro rozhodování o aktuálním stavu systému a přechodu mezi nimi. Uloženy jsou zde například všechny podporované znaky pro segmentový displej, které jsem vybral 0,1,2,3,4,5,6,7,8,9,A,C,E,F,G,H,J,L,P,U. Mezi proměnnými je také malá sekce konstant, které jsou důležité pro funkčnost celého rozhodovacího systému. Tyto konstanty určují například rozhodovací rychlost nebo otáčky motoru, od kterých se začíná uvažovat, že systém přechází ze stavu na zemi do stavu letu a naopak. Hodnoty rozhodovacích konstant je nutné určit pro každý systém zvlášť dle charakteristik stroje, do kterého se bude systém umisťovat. Za proměnnými se nachází kompletní nastavení GPIO pinů, které je potřeba pro komunikaci s připojenými komponentami. Nastaveny jsou zde všechny samostatné piny, sériová komunikace UART pro GPS modul a I2C komunikace pro RTC modul. Pro komunikaci s RTC modulem jsem využil volně dostupnou knihovnu vytvořenou výrobcí modulu, která nabízí jednoduché rozhraní pro kompletní komunikaci s tímto modulem, uloženou na platformě Github⁶.

Druhou částí kódu je nekonečná smyčka, která čte data z připojených komponent a určuje stav systému. Každý průchod cyklem se stará o určení, zda byl vybrán pilot a podle toho zobrazuje danou hodnotu na segmentovém displeji. Pokud nebyl od spuštění programu vybrán žádný pilot, je pomocí globálních proměnných zajištěno, aby displej každou sekundu blikal všemi segmenty a upozornil tak pilota, aby se k letu navolil. Dále se cyklus stará o získání aktuálních hodnot otáček motoru a získání dat z GPS modulu. Otáčky motoru jsou získávány pomocí přerušení vyvolávaného logickou jedničkou přicházející od motoru přes vytvořený obvod přizpůsobení napěťových úrovní. V programu počítám, kolik přijde impulzů za sekundu a podle této hodnoty následně vypočítám odpovídající počet otáček motoru za minutu. Pro motor Rotax 912, namontovaný na testovacím letadle, udávají 2 pulzy jednu otáčku motoru. Tato hodnota se může lišit pro konkrétní ultralehký letoun, ve kterém bude systém namontován. Potom by bylo nutné předělat i rovnici výpočtu otáček. GPS data jsou vyhodnocována trochu komplikovanějším způsobem. GPS modul posílá data v NMEA⁷ formátu a jsou odeslána modulem každou sekundu. Pro možnost práce s těmito daty je tedy v každém cyklu hlavní smyčky nutné přečíst všechna NMEA data a vyhodnotit, zda jsou validní a nebo GPS modul zatím nezachytil GPS signál. Získaná data jsou uložena do lokálních proměnných a slouží dál rozhodovacímu algoritmu pro určení stavu systému. Pokud není zachycen signál GPS, data v lokálních proměnných jsou prázdná a aktuální rychlost nastavena na 0 uzlů.

Poslední částí hlavní smyčky je rozhodovací, kód který je vypsán v sekci 5.5. Jak lze na popisu jednotlivých stavů vidět, tak fáze letu a přistání se proti prvním fázím už nezabývají aktuální hodnotou otáček motoru, ale jen aktuální rychlostí. Tato úprava byla vytvořena proto, aby byl rozhodovací systém schopný o stavu systému rozhodnout i pokud má dostupnou jen jednu z hodnot. Může se stát, že se GPS spojení vůbec nenačte nebo se během letu přeruší. Pokud by pro rozhodovací systém byla dostupná jen hodnota aktuálních otá-

⁶https://github.com/switchdoclabs/RTC_SDL_DS3231

⁷<http://aprs.gids.nl/nmea/>

ček motoru za minutu, tak během letu není jednoduché určit, kdy má let skončit. Je totiž běžné, že se nastavení výkonu motoru mění od volnoběhu do maximálního výkonu. Pro zajištění spolehlivého měření letu když je dostupná jen aktuální rychlost motoru pracuje rozhodovací systém tak, že po odstartování se rozhodovací logika přepíná do stavu letu, ve kterém už ale zůstává a bez načtení údaje o rychlosti se nemůže z tohoto stavu přepnout jinam. Let je tedy počítán až do doby, než je vypnut přívod energie, který se stává většinou ve stejný okamžik, kdy se letadlu vypíná motor. Zařízení tedy umožňuje tvořit rozumně přesné záznamy, ať už je dostupná jakákoli jedna z hodnot.

Ukládání letových logů je řešeno pomocí pomocného souboru, který si tento program vytváří. Program si do něj ukládá zvoleného pilota a všechny vytvořené logy, po ukončení letu jsou data z tohoto souboru přečtena, uložena do databáze a následně smazána, aby se mohl zaznamenávat další let. Když se zaznamenané logy neuloží před vypnutím systému, je soubor zpracován při dalším spuštění systému. Tento způsob nakládání se záznamy je výhodný pro koncentrování času, kdy je nutné zapsat data do databáze, také poskytuje lepší práci s databází při vkládání pilota k letu. Při přímé práci s databází by bylo nutné při každé změně pilota tlačítkem změnit hodnoty v databázi, místo jednotného zapsání po ukončení letu.

Podrobný popis implementace softwaru mobilní aplikace

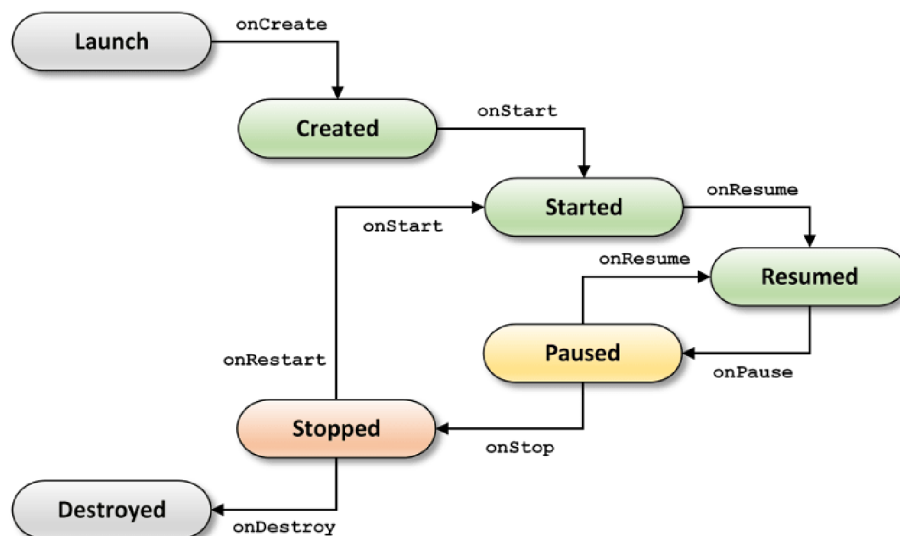
Při tvorbě nové aplikace je vždy zapotřebí zvolit minimální verzi operačního systému Android, na které dokáže aplikace stále běžet. Výběr minimální verze má za důsledek, že v programu nebude možné použít určité konstrukce které jsou dostupné jen na novějších verzích a bude potřebné požadovanou funkčnost vytvořit zastaralejšími způsoby. Je tedy nutné vybrat verzi, kterou podporuje rozumné množství zařízení na trhu a zároveň už není příliš stará na to, aby výrazně omezovala nová zařízení. Pro usnadnění tohoto výběru lze použít tabulku dostupnou v tomto vývojovém prostředí, která zobrazuje přibližné procento zařízení které dokáží spustit aplikaci pro konkrétní verzi Androidu (viz obrázek B.1). Dle této tabulky jsem tedy nakonec vybral verzi 5.0, která podporuje v dnešní době většinu zařízení.

| ANDROID PLATFORM VERSION | API LEVEL | CUMULATIVE DISTRIBUTION |
|--------------------------|-----------|-------------------------|
| 4.0 Ice Cream Sandwich | 15 | |
| 4.1 Jelly Bean | 16 | 99,8% |
| 4.2 Jelly Bean | 17 | 99,2% |
| 4.3 Jelly Bean | 18 | 98,4% |
| 4.4 KitKat | 19 | 98,1% |
| 5.0 Lollipop | 21 | 94,1% |
| 5.1 Lollipop | 22 | 92,3% |
| 6.0 Marshmallow | 23 | 84,9% |
| 7.0 Nougat | 24 | 73,7% |
| 7.1 Nougat | 25 | 66,2% |
| 8.0 Oreo | 26 | 60,8% |
| 8.1 Oreo | 27 | 53,5% |
| 9.0 Pie | 28 | 39,5% |
| 10. Android 10 | 29 | 8,2% |

Obrázek B.1: Tabulka vyjadřující přibližný poměr zařízení, která podporují aplikace po určité verze Androidu. Převzato z Android Studia. Platné k 07.04.2021

Samotná aplikace tvořená v Android Studiu se primárně skládá z aktivit, kde každá aktivita znázorňuje jedno okno mobilní aplikace. Pro vytvoření jedné aktivity je zapotřebí vytvořit dva soubory, které ji budou definovat, a to soubor .xml, ve kterém je uložen vzhled aktivity a soubor .kt, ve kterém je uložena funkčnost dané aktivity naprogramovaná v programovacím jazyce Kotlin.

Přecházení uživatele mezi jednotlivými aktivitami je řízeno takzvaným životním cyklem aktivity (viz obrázek B.2) ve kterém jsou znázorněny jednotlivé stavy, ve kterých může



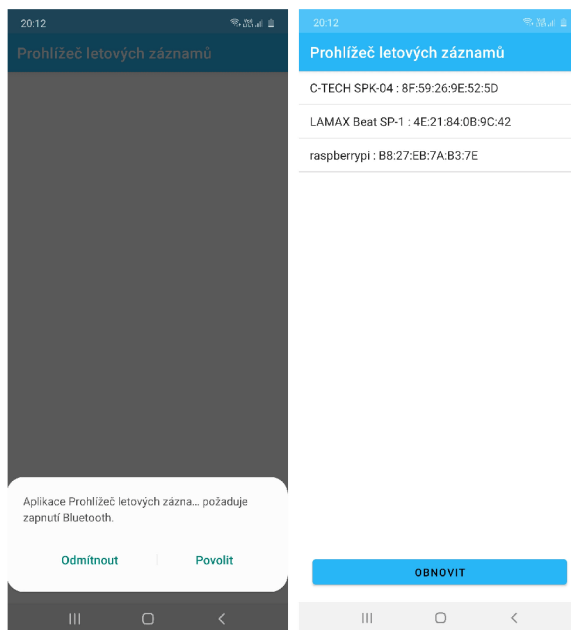
Obrázek B.2: Životní cyklus Android aktivity

aktivita být a jak aktivita mezi jednotlivými stavy přechází. Běžně v mobilních aplikacích funguje, že když uživatel zmáčkne tlačítko zpět, tak se dostane na poslední aktivitu, na které byl předtím. Pro účely mé aplikace bylo však zapotřebí tuto funkčnost změnit, protože návrat mezi některými okny je v aplikaci, ať už z důvodu navázaného Bluetooth spojení nebo přihlášení uživatele, nemožný. Příkladem může být, že je uživatel na aktuální stránce svých letů a následně by se odhlásil, při zmáčknutí zpětného tlačítka by se uživatel při normální funkčnosti dostal zpátky na okno se svými lety, jako kdyby byl přihlášený, to je ale z pohledu přihlašování uživatelů kompletně špatně. Podobný problém by nastal i u aktivity umožňující navázání Bluetooth spojení, zde by ale nejspíš celá aplikace při provedení uvedeného manévru spadla. Podrobněji je celý tento postup vysvětlen v textu příslušejícímu konkrétním aktivitám.

Kromě aktivit je také v projektu vytvořen soubor SQLiteHelper.kt, ve kterém je vytvořena třída uchováající databázi provozu letadla načtenou z vestavěného zařízení. To, že mobilní aplikace uchovává vlastní kopii letových dat, značně poníží závislost aplikace na nepřetržitém přenášení všech potřebných dat mezi mobilní aplikací a vestavěným zařízením. Implementace vnitřní databáze umožňuje jedno načtení dat, které se provede při navázání Bluetooth spojení, následně už pak mohou aktivity číst data z vnitřní databáze. Pro komunikaci s vestavěným zařízením jsou následně potřeba už jen jednoduché příkazy, které se využijí, pokud uživatel nějaká data mění či maže a je potřeba je propsat do hlavní databáze. Jinak by se po zavření mobilní aplikace ztratila a ostatním uživatelům by se nezobrazila. Lokální databáze také dovoluje ztrátu Bluetooth spojení během prohlížení záznamů uživatelem, v tomto případě může dál uživatel prohlížet všechna data, ale je nedostupná jakákoli editace.

Následuje popis aktivit využitých v aplikaci v přibližném pořadí, jak se zobrazují uživateli:

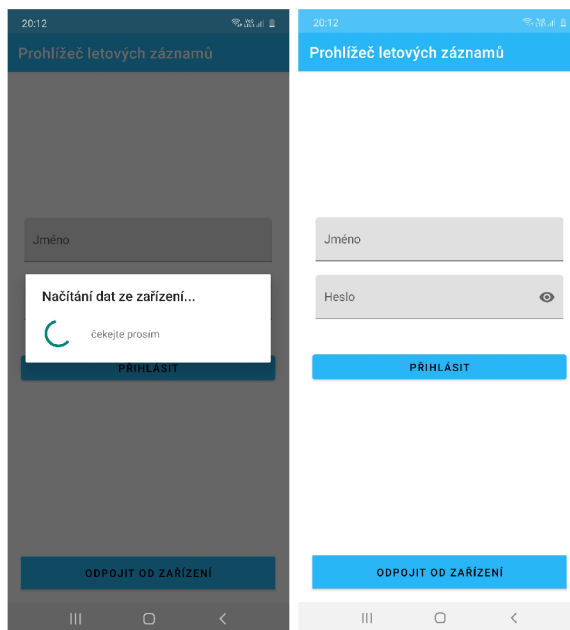
SelectDeviceActivity



Obrázek B.3: Ukázka vzhledu SelectDeviceActivity

Tato aktivita je první, která se uživateli zobrazí po spuštění aplikace a slouží uživateli k výběru konkrétního vestavěného zařízení, ke kterému se chce připojit, pokud by jich měl víc (viz obrázek B.3). V seznamu dostupných zařízení je vždy zobrazena zařízení, která jsou přes Bluetooth spárovaná s mobilním zařízením, pokud by se tedy uživatel připojoval poprvé, tak je důležité, aby nejdříve spároval zařízení v nastavení mobilního telefonu, jinak se mu zařízení nezobrazí. Platné spojení je navázáno, když uživatel vybere zařízení, které je v provozu a je to platné vestavěné zařízení, ne např. bezdrátová sluchátka. Po navázání spojení se aplikace automaticky přepne do přihlašovací aktivity.

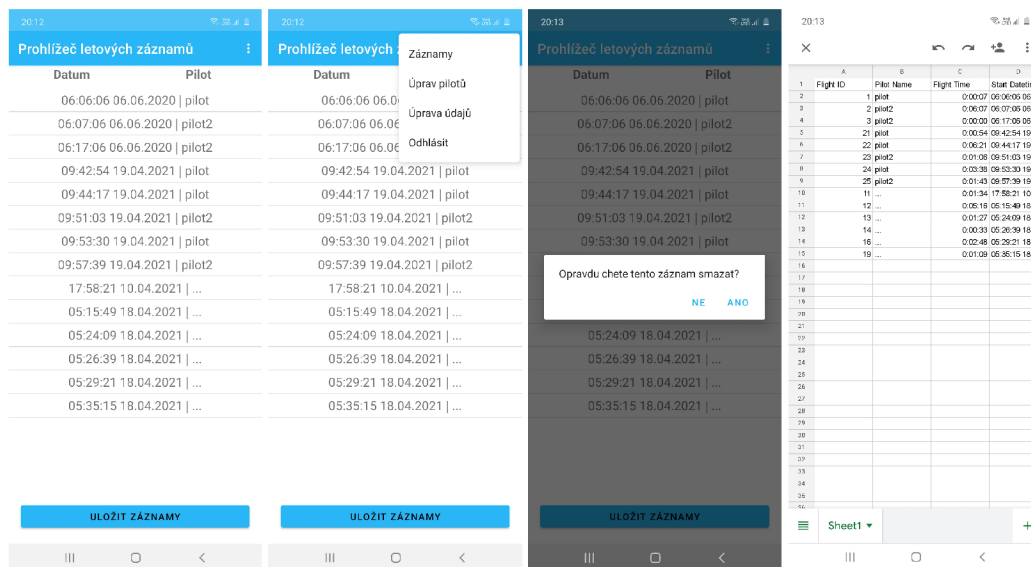
LoginActivity



Obrázek B.4: Ukázka vzhledu LoginActivity

LoginActivity, jak už název napovídá, slouží pro přihlášení konkrétního uživatele aby si mohl prohlédnout svá data na vestavěném zařízení (viz obrázek B.4). První věc která se spustí při zapnutí této aktivity, je stahování dat z vestavěného zařízení do interní databáze, která se následně využívá pro veškerou práci s daty. Aktivita obsahuje jednoduchý formulář, kam uživatel zadá své přihlašovací údaje, které se po stisknutí tlačítka k přihlášení ověří se záznamy v databázi a při zadání platných údajů je uživatel puštěn na další aktivitu, kterou je FlightsActivity. Aktivita také obsahuje tlačítko pro odpojení od zařízení pokud by se uživatel přihlásil ke špatnému vestavěnému zařízení, pokud jich má spuštěných víc naráz.

FlightsActivity

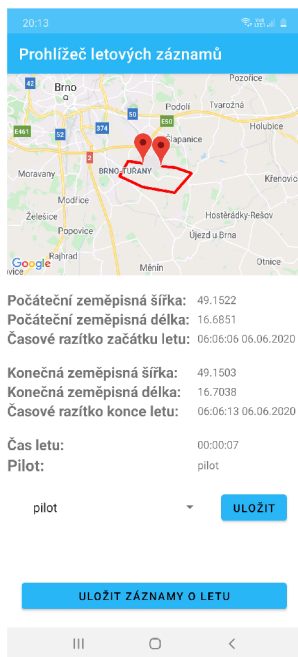


Obrázek B.5: Ukázka funkčnosti FlightsActivity

FlightsActivity slouží uživatelům pro zobrazení jejich letů, které jsou na vestavěném zařízení uložena, kromě letů příslušných jednotlivým pilotům jsou také zobrazeny lety, u kterých nebyl vybrán pilot pro možnost jejich zpětného přiřazení (viz obrázek B.5). Všechny záznamy lze seřadit vzestupně či sestupně podle data jejich pořízení a jména pilota. Kliknutím na konkrétní let se zobrazí FlightDetailActivity s podrobnějšími daty o letu. Záznamy o letech je také možné stáhnout pro následné zpracování, například na počítači. Data se stahují pomocí tlačítka přítomného na displeji a ukládají se do složky se staženými soubory mobilního zařízení. Pro jednoduché následné zpracování jsou data ukládána ve formátu CSV⁸. Pro další navigaci mezi aktivitami je v pravém horním rohu okna přítomné menu umožňující pohyb mezi FlightsActivity a EditUserActivity, nebo se zde může uživatel odhlásit, a dostane se zpět na LoginActivity.

Pro majitele letadla je funkčnost této aktivity rozšířena o možnost mazání letových záznamů dlouhým stiskem na konkrétním záznamu a o možnost přesunu na EditPilotsActivity dostupnou jako další položka v menu.

⁸<https://cs.wikipedia.org/wiki/CSV>



Obrázek B.6: Ukázka vzhledu FlightDetailActivity



Obrázek B.7: Ukázka vzhledu EditUserActivity



Obrázek B.8: Ukázka vzhledu EditPilotsActivity

FlightDetailActivity

FlightDetailActivity ukazuje uživateli všechny důležité informace o vybraném letu (viz obrázek B.6). Pokud byl celý let pořízený i s GPS záznamy, tak se zobrazí i jeho dráha na Google mapách. Uživateli je také umožněno stažení jednotlivých logů zaznamenaných zařízením k aktuálnímu letu, pokud by je k něčemu potřeboval. Záznamy si lze stáhnout ve formátu CSV příslušným tlačítkem. Poslední nabízenou funkcí této aktivity je výběrový box, ve kterém lze přiřadit pilota k vybranému letu. Pro piloty je tato funkčnost dostupná jen pro případ, pokud by chtěly přiřadit sami sebe k letu, který nemá pilota, měnit data dalších pilotů nemohou. Majitel letadla může přiřadit let, kterémukoli pilotovy.

EditUserActivity

Tato aktivita slouží uživatelům pro změnu údajů, kterými se do aplikace přihlašují. Pro piloty je zde také možnost změny jejich znaku, přes který se volí na vestavěném zařízení (viz obrázek B.7). Změna údajů je umožněna přes vytvořený formulář, který uživatel uloží zobrazeným tlačítkem.

EditPilotsActivity

EditPilotsActivity je dostupná jen pro majitele letadla a umožňuje mu spravovat účty jednotlivých pilotů kteří s jeho letadlem létají (viz obrázek B.8). Stiskem řádku s pilotem se otevře EditOrSavePilotActivity, ve které může majitel změnit přihlašovací údaje pilota a jeho znak. Podržením řádku s pilotem je možné ho také smazat. Dále je přítomné tlačítko umožňující tvorbu nového pilota, vyvoláním EditOrSavePilotActivity. Tvorba nového pi-

lota je tedy umožněna jen majiteli ultralehkého letounu, pilot si sám vytvořit účet nemůže. Tento způsob byl zvolen z důvodu bezpečnosti a kontroly všech účtů pilotů majitelem.

EditOrSavePilotActivity

EditOrSavePilotActivity slouží majiteli letadla pro změnu údajů o jednotlivých pilotů a tvorbu nových účtů pro piloty, kteří s letadlem létají.

Všechny aktivity jsou také dostupné ve světlém nebo tmavém provedení podle nastavení režimu mobilního telefonu.