

Česká zemědělská univerzita v Praze

Technická fakulta



**Návrh automatického trasovacího navigačního
systému pro nevidomé v uzavřeném prostoru**

diplomová práce

Vedoucí diplomové práce: prof. Ing. Jaromír Volf, DrSc.

Diplomant: Bc. David Trejtnar

Konzultant: Ing. Viktor Novák

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. David Trejtnar

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Návrh automatického trasovacího navigačního systému pro nevidomé v uzavřeném prostoru

Název anglicky

Design of an automatic tracking navigation system for sightless people for indoor space use

Cíle práce

Cílem diplomové práce je navrhnout trasovací navigační systém pro nevidomé. Navigační systém bude primárně určen pro usnadnění pohybu nevidomé osoby v uzavřených prostorech (budovách). Systém bude založen na síti fixních radiomajáků umístěných v kontrolních bodech a mobilním přijímači signálu pro nevidomou osobu. Tento přijímač bude dávat nevidomé osobě hlasové pokyny umožňující její orientaci a navigaci v daném objektu.

Metodika

V teoretické části práce se student seznámí s problematikou zrakového postižení a s užívanými kompenzačními pomůckami pro nevidomé, se zaměřením na navigační systémy pro nevidomé. Následně bude navržen vlastní navigační systém založený na síti radiomajáků a mobilním přijímači pro nevidomou osobu. Funkce navrženého systému bude ve vybraných vnitřních prostorech následně ověřena. Závěrem bude diskutován přínos celého systému a možnosti jeho širšího využití.

Doporučený rozsah práce

50 – 60 stran

Klíčová slova

nevidomí, orientace v budově, navigace pro nevidomé

Doporučené zdroje informací

Cameron, N. 2021. Electronics projects with the ESP8266 and ESP32: building web pages, applications, and WiFi enabled devices. Springer, New York. ISBN 9781484263365

Häberle, H. a kolektiv. 2003. Průmyslová elektronika a informační technologie. Europa-Sobotáles. ISBN: 80-86706-04-4.

Karimi, H. A. 2015. Indoor Wayfinding and Navigation. CRC Press, London. ISBN 1482230844

Volf, J., Škeřík, F., Novák, V. 2018. Technical design of complex vision-tactile navigation system for using of blind persons navigation. Agronomy Research 16 (Special Issue 1), s. 1278-1288.

Předběžný termín obhajoby

2022/2023 LS – TF

Vedoucí práce

prof. Ing. Jaromír Volf, DrSc.

Garantující pracoviště

Katedra elektrotechniky a automatizace

Konzultant

Ing. Viktor Novák

Elektronicky schváleno dne 31. 1. 2023

doc. Ing. Monika Hromasová, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 31. 1. 2023

doc. Ing. Jiří Mašek, Ph.D.

Děkan

V Praze dne 31. 01. 2023

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Návrh automatického trasovacího navigačního systému pro nevidomé v uzavřeném prostoru" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2023

podpis autora

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce prof. Ing. Jaromíru Volfovi, DrSc., a konzultantovi Ing. Viktoru Novákovi za jeho vstřícný přístup vedení mé práce. Dále bych rád poděkoval Mgr. Taťaně Horehledové za korekci textu, Bc. Jirkovi Alexandrovičovi a Bc. Matějovi Kohútkovi za odborné rady při vývoji systému.

Abstrakt: Diplomová práce se zabývá navigací učenou pro lidi se zrakovým postižením. Navigace je navržena pro využití ve vnitřních prostorech budov. V rešeršní části je definováno zrakové postižení a její klasifikace do jednotlivých úrovní. Následně jsou popsány navigační systémy, které jsou popsány od počátku prvních navigačních systémů až po současnost. Zmíněna jsou také technická východiska potřebná pro tuto diplomovou práci. Konkrétně jsou popsány způsoby, kterými lze využít bezdrátové přijímače a vysílače k určení polohy, a technické možnosti použitého mikroprocesoru ESP8266.

Dále se práce zabývá vývojem samotného navigačního systému. Je navrženo a popsáno technické provedení, které by mělo být vytvořeno. Následuje výběr přenosového protokolu, testování komunikace mikroprocesoru se zvukovým modulem a popis nahrání programu pomocí programátoru CP2102. Poté následuje popis funkce tří programů pro tři různé typy zařízení. Hardware těchto zařízení je následně popsán. Celý systém je následně otestován a upraven z výsledku testování.

Klíčová slova: nevidomý, orientace v budově, navigace pro nevidomé

Design of an automatic tracking navigation system for sightless people for indoor space use

Abstract: The diploma thesis deals with learned navigation for people with visual impairments. The navigation system is intended for indoor use. The research part defines visual impairment and classifies it into different levels. Subsequently, the evolution of navigation systems is described from the first systems to the present day. The technical starting points needed for this thesis are also mentioned. The thesis outlines the use of wireless receivers and transmitters for location determination, as well as the technical capabilities of the ESP8266 microprocessor.

Furthermore, the work deals with the development and description of the technical design of the navigation system. This is followed by the selection of the transmission protocol, testing of the communication of the microprocessor with the sound module and a description of the navigation program upload using the CP2102 programmer. This is followed by a description of the function of the three programs for three different types of devices which are used in the navigation system. The hardware of these devices is subsequently described. The entire system is subsequently tested and adjusted based on the results of the testing.

Keywords: blind, orientation in the building, navigation for the blind

Obsah

1 Úvod.....	1
2 Cíl práce a metodika	2
2.1 Cíl práce	2
2.2 Metodika	2
3 Zrakové postižení	3
4 Přehled dosavadních pomůcek.....	4
4.1 Staré projekty pro navigaci nevidomých.....	4
4.1.1 Brunel Navigation System for Blind	5
4.1.2 MoBIC	5
4.1.3 Drishti	6
4.1.4 Noppa.....	7
4.2 Vodící pes.....	7
4.3 Bílá hůl	7
4.3.1 Vylepšení bílé hole z ČVUT.....	8
4.4 Navigačního středisko	9
4.5 Tyflosonar	10
4.6 Taktilní navigační systém	10
4.7 Haptické mapy	12
4.8 VPN – vysílačka pro nevidomé.....	13
4.9 Orientační maják	14
4.10 Programy pro smartphone	15
4.10.1 RightHear.....	15
4.11 Zhodnocení.....	17
5 Popis kritických komponent navrženého systému	18
5.1 Určení polohy v bezdrátové síti Wi-Fi.....	18
5.2 ESP8266.....	20
5.3 DY-SV17F (MP3 zvukový modul).....	23
6 Vývoj systému.....	25
6.1 Požadavky na systém	25
6.2 Návrh systému Verze1	25
6.2.1 Nastavení programu Arduino IDE	27
6.2.2 Komunikace mezi ESP a MP3 modulem.....	28
6.2.3 Nastavení sítě.....	31

6.2.4	Zapojení na univerzálním DPS	31
6.2.5	Program pro navigační modul.....	33
6.2.6	Program pro maják.....	41
6.2.7	Program pro maják s webovým serverem.....	42
6.2.8	Výpočet konstanty pro indikaci nízkého stavu baterie	45
6.2.9	Hardwarové zapojení pro testování	46
6.3	Vyzkoušení systému a ladění systému	49
7	Výsledky a diskuse	51
8	Závěr.....	53
9	Bibliografie	54
10	Přílohy	58

Seznam obrázků

Obr. 1	Funkce tlačítek na vylepšené holi [12]	8
Obr. 2	Schéma komunikace s navigačním centrem [12]	9
Obr. 3	Tyflosonar RAY	10
Obr. 4	Blokové schéma Taktilního navigačního systému [15].....	10
Obr. 5	Uchycení hmatového aktivátoru na paži [15].....	11
Obr. 6	Záznam kamery a aktivace elektromotorů [15].....	12
Obr. 7	Mapa čitelná hmatem [16].....	13
Obr. 8	Vysílačka VPN 01 [17].....	13
Obr. 9	zobrazení aplikace RightHear (25)	16
Obr. 10	schématické znázornění ToA (na levé straně) a TDoA (na pravé straně [22]).....	20
Obr. 11	Blokové schéma ESP8266 [24]	21
Obr. 12	ESP8266 verze 12-E pinout [28]	22
Obr. 13	Wemos D1 Mini pinout [28]	22
Obr. 14	Popis výstupních pinů modulu DY-SV17F [29]	24
Obr. 15	Blokové schéma majáku a navigačního modulu	26
Obr. 16	Rozložení sítě	26
Obr. 17	vývojové prostředí Arduino IDE [30]	28
Obr. 18	Schéma zapojení pro ověření funkčnosti komunikace s MP3 modulem.....	28
Obr. 19	Reálné zapojení pro ověření funkčnosti komunikace s MP3 modulem	29
Obr. 20	Program – testování MP3 modulu	30
Obr. 21	Schéma zapojení – test 1	32

Obr. 22 Schéma zapojení – test 2	32
Obr. 23 Program pro navigační modul – Setup	33
Obr. 24 Životní cyklus programu pro navigační modul	34
Obr. 25 Program pro navigační modul – hlavní programové smyčka část 1.....	35
Obr. 26 Program pro navigační modul – najitVzdalenost().....	35
Obr. 27 Program pro navigační modul – skenARSSI()	36
Obr. 28 Průměr hodnot	37
Obr. 29 Průměr hodnot a oříznutí špiček	37
Obr. 30 Program pro navigační modul – hlavní programové smyčka část 2.....	38
Obr. 31 Vývojový diagram pro navigaci	39
Obr. 32 Program pro navigační modul – přerušení	40
Obr. 33 Životní cyklus programu pro maják	41
Obr. 34 Programové definování datové struktury	41
Obr. 35 Program pro maják – světelná indikace.....	42
Obr. 36 Životní cyklus programu pro maják kombinovaný s web. serverem	42
Obr. 37 Program pro maják s web. serverem – výpis hodnot.....	43
Obr. 38 Pohled na obsah webového serveru.....	44
Obr. 39 Pohled na zapojení na univerzálním poli.....	44
Obr. 40 ESP maják.....	46
Obr. 41 ESP Maják s webovým serverem	47
Obr. 42 ESP-Navigační modul	47
Obr. 43 ESP12F s programátorem CP2102 a obvodem potřebným k nahrání programu ...	48
Obr. 44 Pohled na zapojení ESP majáku na univerzální desce	48
Obr. 45 Program pro navigační modul – úprava navigace	49
Obr. 46 Výpis do konzole po úpravě programu.....	50
Obr. 47 Znázornění testovací trasy	50

Seznam tabulek

Tab. 1 Klasifikace zrakového postižení podle WHO [1]	3
Tab. 2 MoBIC – rozdělení úrovní [5]	6
Tab. 3 Funkce tlačítek [17]	14
Tab. 4 Přehled standardů IEEE 802.11 [21]	18
Tab. 5 Přehled přesnosti bezdrátových technologií [22]	19
Tab. 6 konfigurace DY-SV17F [29]	23
Tab. 7 Funkce příslušných konfigurací [29]	24
Tab. 8 Přehled nahraných hlášek ve zvukovém modulu	29
Tab. 9 Definovaná datová struktura pro komunikaci ESP_NOW	31

Seznam použitých zkratk

České

AD	analog digital
DPS	deska plošných spojů
MS	Microsoft
ZP	zrakové postižení

Anglické

AP	access point / přístupový bod
DC	direct current /stejnsměrní proud
GIS	geographic information systém / geografický informační systém
GPS	global positioning systém / globální polohový systém
Iot	internet of things / internet věcí
IEEE	institute of electrical and electronics engineers / institut elektrických a elektronických inženýrů
LED	light-emiting diod / dioda emitující světlo
PC	personal computer / osobní počítač
RAM	random access memory / náhodně přístupná paměť
RF	radio frequency / vysokofrekvenční
RSS	received signal strenght / síla přijímaného signálu
TDOA	time difference of arrival / rozdílný čas přeletu
TOA	time of arrival / doba přiletu
UART	universal asynchronous receiver-transmitter / univerzální asynchronní přijímač-vysílač
USB	universal serial bus / univerzální sériová sběrnice

1 Úvod

Pro lidi s jakoukoli vadou zrakového aparátu je velmi těžké se orientovat v prostoru. V některých případech stačí zrak napravit dioptrickými brýlemi. Bohužel v některých případech ovšem tato náprava není možná. Zde už přichází řada pro jiné pomůcky, které mohou být naprosto jednoduché, jako třeba bílá hůl, nebo velmi komplexní, jako jsou například navigační systémy.

Nejčastěji nevidomého poznáme podle toho, že v ruce má uchopenou bílou hůl a vede ji před sebou po zemi. Dalším velmi výrazným znakem je, že dotyčná osoba nekouká tam, kam jde. Místo toho kouká takzvaně do prázdna. Případně nevidomý může mít černé brýle, které zakrývají jeho oči. Může také mít u sebe psího společníka, který je speciálně vycvičen a označen příslušným postrojem.

Existuje již celá řada pomocných systémů, jež dotyčnému jakýmkoliv způsobem sdělují informace, které on nemůže vidět. Většinou jde o hlasovou komunikaci, ale najdou se i systémy, které využívají hmatovou komunikaci. Nicméně tyto komunikace používá každý člověk dennodenně. Skvělým příkladem může být chytrý telefon. Hmatovou komunikaci může představovat vibrování telefonu, když uživateli někdo volá, nebo spuštění zvukové hlášky při příjmu SMS zprávy. Samotnou hlasovou komunikaci může představovat navigace, když uživatele navádí hlasovými povely.

Vývojáři mobilních aplikací se v poslední době zaměřili i na vyvíjení aplikací pro lidi se zrakovým postižením. Tyto aplikace jsou založeny na různých principech a lze je vždy využít ke specifickým účelům. Důležitou roli v tomto směru sehrály i dnešní moderní technologie, kterými jsou dnes vybaveny chytré telefony, nebo IoT prvky.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem diplomové práce je navrhnout navigační systém, který je převážně určen pro nevidomé osoby. Tento systém nevidomé osobě má usnadnit pohyb v budovách. Nicméně tento systém by mohly využívat i osoby, které jsou v budově poprvé.

Navržený systém bude složen z fixních radiomajáku umístěných v kritických bodech (např. u schodiště, na křížení chodeb, u výtahů, dveří) a z mobilních přijímačů signálu pro naváděnou osobu. Mobilní přijímač je vybaven reproduktorem, ze kterého jsou pouštěny uživateli hlasové povely, které ho navedou na místo určení. Systém je potřeba navrhnout tak, aby byl energeticky úsporný. Jednotlivé prvky mohou pracovat bez vnějšího zdroje elektrické energie.

2.2 Metodika

V rešeršní části je stručně definováno zrakové postižení a její hodnocení dle kritérií. Dále jsou popsány některé navigační pomůcky pro nevidomé nebo osoby se zhoršeným orientačním smyslem. U jednotlivých systémů jsou popsány jejich funkce. V poslední části jsou popsány komponenty, které jsou důležité pro navržení a sestavení navigačního systému.

Následně je navržen vlastní navigační systém založený na síti radiomajáků a mobilním přijímačem pro nevidomou osobu. Postup řešení navrženého systému je rozdělen do několika dílčích úloh, které budou v posledním kroku spojeny, a bude vytvořen celistvý systém.

První úlohou je připravit vývojové prostředí, otestovat komunikaci vybraných modulů po sériové lince s využitím knihovny a nahrání vlastních zvukových pokynů. Následně je vybrán přenosový protokol a definována struktura posílání dat mezi moduly. Pro modul nepřizpůsobený k vývoji bylo navrženo schéma, které následně bylo otestováno. Následoval vývoj tří softwarů pro tři různé režimy. První je popisován program pro navigační modul, i když programy byly částečně vyvíjeny souběžně. Následně je popsán program pro maják a maják s webovým serverem. Jako poslední je popis hardwarového zapojení.

Funkce navrženého systému byla ve vybraném vnitřním prostoru následně ověřena a byly popsány potřebné úpravy. Zbytek práce je věnován možnosti využití systému a různé varianty vylepšení.

3 Zrakové postižení

Rozlišovat tvary, barvy, odstíny nám umožňuje zrak. Zrakem vnímáme přibližně 85 % informací z dění kolem nás. Pokud dotyčný přijde právě o zrak, musí se spolehnout na další smysly, jako je hmat, sluch, popřípadě čich. [1]

Termínem zrakové postižení rozumíme, že poškození zraku nějakým způsobem ovlivňuje každodenní činnost člověka, u kterého optická korekce nepomáhá. Podle závažnosti jsou rozdělovány do 5 kategorií, viz. Tab. 1. Pokud člověk nosí dioptrické brýle a díky nim vidí dobře, nespadá do žádné kategorie zrakového postižení. [1]

Tab. 1 Klasifikace zrakového postižení podle WHO [1]

Kategorie	Druh zdravotního postižení	Popis
1	Střední slabozrakost	zraková ostrost s nejlepší možnou korekcí: maximum menší než 6/18 (0,30) - minimum rovné nebo lepší než 6/60 (0,10); 3/10–1/10
2	Silná slabozrakost	zraková ostrost s nejlepší možnou korekcí: maximum menší než 6/60 (0,10) - minimum rovné nebo lepší než 3/60 (0,05); 1/10–10/20
3	Těžce slabý zrak	a) zraková ostrost s nejlepší možnou korekcí: maximum menší než 3/60 (0,05) - minimum rovné nebo lepší než 1/60 (0,02); 1/20–1/50 b) koncentrické zúžení zorného pole obou očí pod 20 stupňů, nebo jediného funkčně zdatného oka pod 45 stupňů
4	Praktická slepota	zraková ostrost s nejlepší možnou korekcí 1/60 (0,02), 1/50 až světlocit nebo omezení zorného pole do 5 stupňů kolem centrální fixace, i když centrální ostrost není postižena
5	Úplná slepota	ztráta zraku zahrnující stavy od naprosté ztráty světlocitu až po zachování světlocitu s chybnou světelnou projekcí

Zraková ostrost je složená ze dvou čísel oddělené lomítkem, nejčastěji ve zlomku. První číslo je vzdálenost v metrech, ve které dotyčný čte. Druhé číslo je vzdálenost v metrech, ve které čte člověk s dobrým zrakem. [1]

4 Přehled dosavadních pomůcek

Již od 80. let jsou datovány první záznamy, které se zabývaly problematikou navigování osob v prostorech. Od této doby byla provedena spousta studií, návrh různých pomůcek a především systémů pro navigaci jak pro vnitřní prostor, tak pro vnější prostory. Mohou to být knihy napsané Braillovým písmem, modifikované různé stolní hry, slepecké hole nebo mobilní telefony, jejichž komunikační rozhraní je upraveno k potřebám nevidomého. Obecně pomůcky pro nevidomé se nazývají tyflopomůcky. [2]

V dnešní ovládání mobilních telefonů nevidomým je nejčastěji prováděno předčítání textu na obrazovce. Nevidomý různými gesty po obrazovce mění přepočítaný text, potvrzuje právě přečtenou akci, listuje mezi nimi, nebo se vrací zpět. Ne však všechny aplikace jsou pro tento účel odladěny.

Nejčastější pomůckou bývá bílá hůl, ke které bývají voleny různé dodatečné systémy. Tyto systémy rozšiřují vnímání uživatele takovým způsobem, který klasická bílá hůl nemůže nabídnout. Existuje vylepšená varianta bílé hole, jak je popisováno v kapitole 4.3.1. Velkou nevýhodou je, že novodobé systémy založené na bezdrátové technologii nejsou mezi sebou kompatibilní. Je tedy potřeba ke každému systému vlastnit specifickou část, aby ji dotyčný mohl využít.

4.1 Staré projekty pro navigaci nevidomých

Dle dostupných zdrojů jsou první záznamy o vnitřní navigaci datovány od 80. let, kdy proběhla řada důležitých studií. Článek, který publikovali Thorndyke a Hayes-Roth v roce 1982, se zaměřoval na znalosti získaných z mapy nebo navigace velkého vnitřního prostoru. Bylo zjištěno, že informace získané z map jsou velmi odlišné od znalostí získané procházkou ve zkoumaném prostoru. V roce 1983 Gräling a kol. vydali článek, ve kterém bylo popsáno zkoumání orientačních pomůcky a vizuální vnímání používané k orientaci v budovách. [3]

V následujících podkapitolách jsou popsány projekty, které měly jakýmkoliv způsobem pomoci zrakově handicapovaným k jejich větší nezávislosti na ostatních. Popisované projekty byly zahájeny ve 20. století nebo na přelomu 20. a 21. století. Projekty, které se také zabývaly touto problematikou a nejsou dále popsány v textu, jsou Trinetra, Navig, TANIA, Wayfinder access.

4.1.1 Brunel Navigation System for Blind

Tento systém z Bruselské univerzity v Londýně je založen na podobném principu jako navržený systém z ČVUT, který je popsán v kapitole 4.4. Uživatel má u sebe modul, který obsahuje GPS komunikátor pro určení pozice a modul pro hlasovou komunikaci. Tento systém však už od začátku je určen k přenosu videa do navigačního centra, a navíc je vybaven elektronickým kompasem. Pro tento systém byl navržen speciální navigační systém, kde mapy vypadají spíše jako plánek. [4]

4.1.2 MoBIC

Celé znění tohoto projektu v překladu je „mobilita nevidomých a starších lidí při interakci s počítači“. Tento projekt probíhal v letech 1994 až 1997. Lidé se zrakovým postižením mají omezenou možnost při plánování trasy, ať už jedu kamkoliv, a především na nová místa, kde ještě nebyli. Cílem tohoto projektu, jak je již patrné z názvu, bylo zaměření na poskytování informací k naplánování trasy.

Tento systém využíval k určení polohy geografický informačních systémů (GIS) a globální polohový systém (GPS). Systém je rozdělen na 2 podsystémy. První část systému se zaměřuje na pomáhání uživatelům při plánování cestování, pojmenován jako MOBIC Pre-journey (MOTA). Uživatel musel mít doma počítač s nainstalovaným příslušným softwarem a upraven tak, aby s ním mohl nevidomý pracovat. Pomocí tohoto softwaru si může sám připravit trasu. Jelikož mapy neobsahovaly všechny podrobné informace, jako např. značky, informace o křižovatkách, přechody, nebo nízké větve stromů, program umožňoval jim tyto informace vložit do jejich vlastní mapy.

Druhá část byla určena pro samotné navádění osoby ve venkovním prostředí, pojmenována jako MOBIC Outdoor System (MOODS). Jedná se o podpůrný prvek k bílé holi nebo vodícímu psovi. Dotyčný má na hlavě speciální sluchátka, která mu umožňují vnímat okolní ruch. U sebe musí mít „ruční klávesnici“ s hlasovým výstupem, která je potom připojena k navigačnímu systému.

Součástí projektu také bylo získání poznatků z průzkumu teoretických uživatelů. Dále pak byly vedeny rozhovory s lidmi, kteří pracují s pohybovým aparátem nebo v rehabilitačním středisku pro zrakově postižené.

Plánování trasy mělo zajišťovat bezpečnou a krátkou trasu. Mělo se tedy vyhýbat rušným silnicím, kde je větší riziko střetu chodce s vozidlem. Z průzkumu vyplynulo, že někteří chtějí být navigováni méně podrobněji, a jiní zase více podrobněji. Pokud se jednalo o novou trasu nebo oblast, kde si dotyčný nebude jistý svou polohou, bylo vyžadováno podrobné navigování. Proto úroveň poskytování informací byla rozdělena do několika úrovní: [5]

Tab. 2 MoBIC – rozdělení úrovní [5]

Úroveň	Název	Popis
1	Základní informace	směr cestování, sklon cesty, nejbližší křižovatka a všechny známé překážky
2	podrobné informace	zahrnutí informací z MOTA jako jsou například veřejné budovy
3	dopravní informace	informace typu: nejbližší autobusová zastávka, stanoviště taxi, stanice metra

4.1.3 Drishti

Na tomto systému se podílel Dr. Theral Moore, který od svých mladých let už byl téměř slepý. Pro tento projekt byly použity hardwarové komponenty a software běžně dostupné na trhu. Celý systém vážil přibližně 8 liber, což je přibližně 3,63kg, a byl instalován do batohu, což umožňovalo snadnou instalaci systému na uživatele. V batohu byl přenosný počítač, GPS přijímač a elektronický kompas. Uživatel musel mít nasazenou na hlavě soupravu s hlasovým vstupem (mikrofon) i výstupem (sluchátky). Systém byl dále vybaven tlačítkovým polohovacím zařízením a operačním systémem MS Windows 98. [6]

K určení polohy uživatele bylo zde použito diferenciální GPS (DGPS). Jedná se o způsob, kterým lze zlepšit přesnost polohy. Počítač měl přístup k internetu pomocí bezdrátové sítě wifi. Mohl tak být napojen na relační databázi systému GIS. [6]

4.1.4 Noppa

Jedná se o projekt finské vlády, který měl usnadnit cestování hromadnou dopravou zrakově postiženým osobám. Hledalo se úspornější řešení než namontování hlasových systémů na všechna vozidla veřejné dopravy, jelikož potenciálních uživatelů tvořilo zhruba 1,6% populace. Navržené zařízení bylo podobné mobilnímu telefonu. Zařízení bylo vybaveno výstupem pro sluchátka k hlasové komunikaci mezi přístrojem a uživatelem. Uživateli byla prezentována data o dopravním spojení, která již existují na internetu. [7]

4.2 Vodicí pes

Vodicí psi mají speciální výcvik k tomu, aby pomáhali zdravotně postižené osobě s orientací ve známém i neznámém prostoru. Pes se musí soustředit na svou práci, zároveň musí být schopný sám se rozhodovat, ale zároveň poslouchat svého pána. Ke správné funkci je důležitá souhra psovoda a samotného psa. Oba si k sobě musí vybudovat pevný vztah. První záznam o vodicích psech jsou z poloviny 18. století. Vhodná plemena k výcviku jsou flat coated retriever, labradorský retrívr, zlatý retrívr, německý ovčák, goldendoodle a královský pudl. [8] [9]

4.3 Bílá hůl

Nazývána také jako „slepecká hůl.“ Nevidomý používá hůl k získání informace o dotyku s předmětem ve svém okolí. Označení „bílá hůl“ vychází z anglického překladu „White cane“ nebo „Wisse stock.“ Z hlediska konstrukce je lze rozdělit na neskládací, skládací dvou a více dílnou, teleskopickou a kombinovanou. Na výrobu holí je použit různý materiál. Mohou být zhotoveny z hliníku, laminátu, grafitu, dřeva atp. Hole mohou plnit následující funkce [10]:

- a) Signalizační – upozorňuje kolemjdoucí, ale také motoristy na skutečnost, že osoba má jakékoliv poškození zraku.
- b) Ochranná – nevidomý získává informaci o svém okolí, tak chrání jeho např. před pádem či kolizí s cizím předmětem, který by ho mohl nějakým způsobem zranit.
- c) Orientační – získání hmatových informací k získání orientace v prostoru.
- d) Opěrná – hůl lze využít k podpěře osoby.

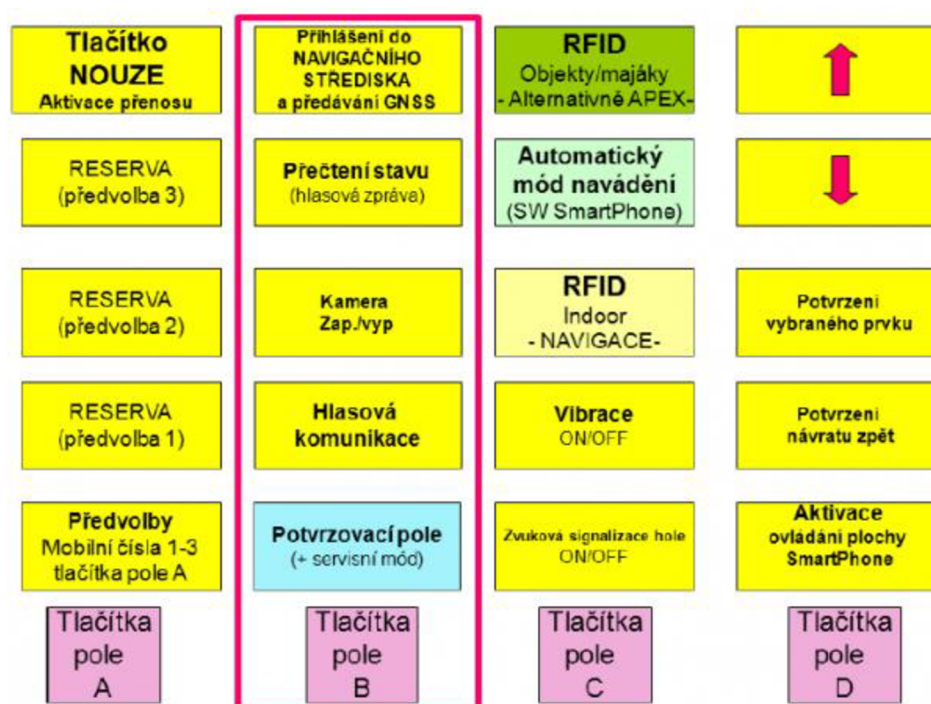
Dle § 21 odst. 3 vyhlášky č. 294/2015 Sb. bílá hůl signalizuje zrakově postiženou osobu. Pokud hůl má střídavě bílou a červenou barvu, signalizuje hluchoslepou osobu. Obě varianty těchto hůlí musí mít reflexní pásku. [11]

4.3.1 Vylepšení bílé hole z ČVUT

Bylo zjištěno, že přidáním různých senzorů, reproduktorů či radarů uživateli je spíše pro obtíž než k prospěchu. Proto vylepšení obsahuje pouze jednoduché ovládání s pár tlačítky, akumulátor, a výpočetní modul s bezdrátovou komunikací. Pro bezdrátovou komunikaci byla zvolena technologie Bluetooth, aby bylo možné se připojit k chytrému telefonu. Podle zakoupené varianty může také být součástí hole piezoelektrický měnič, který může např. pípat, když bude telefon mimo dosah. Stisk jakéhokoliv tlačítka je doprovázeno hlasovou informací o jeho stisknutí. Stejně tak hlášení stavu akumulátoru, potvrzení spárování s telefonem, nebo jakákoliv hlasová komunikace s navigačním střediskem, více v kapitole 4.4. [12]

Na holi se nachází 5 tlačítek, v případě rozšířené varianty pak jednou tolik. Testováním byl určen počet tlačítek k pohodlnému používání. Funkce tlačítek byly zvoleny výběrem testovaných nevidomých, ale funkce je možné i měnit a rozšiřovat. Základní rozložení je znázorněno na Obr. 1 Funkce tlačítek na vylepšené holi. [12]

Obr. 1 Funkce tlačítek na vylepšené holi [12]

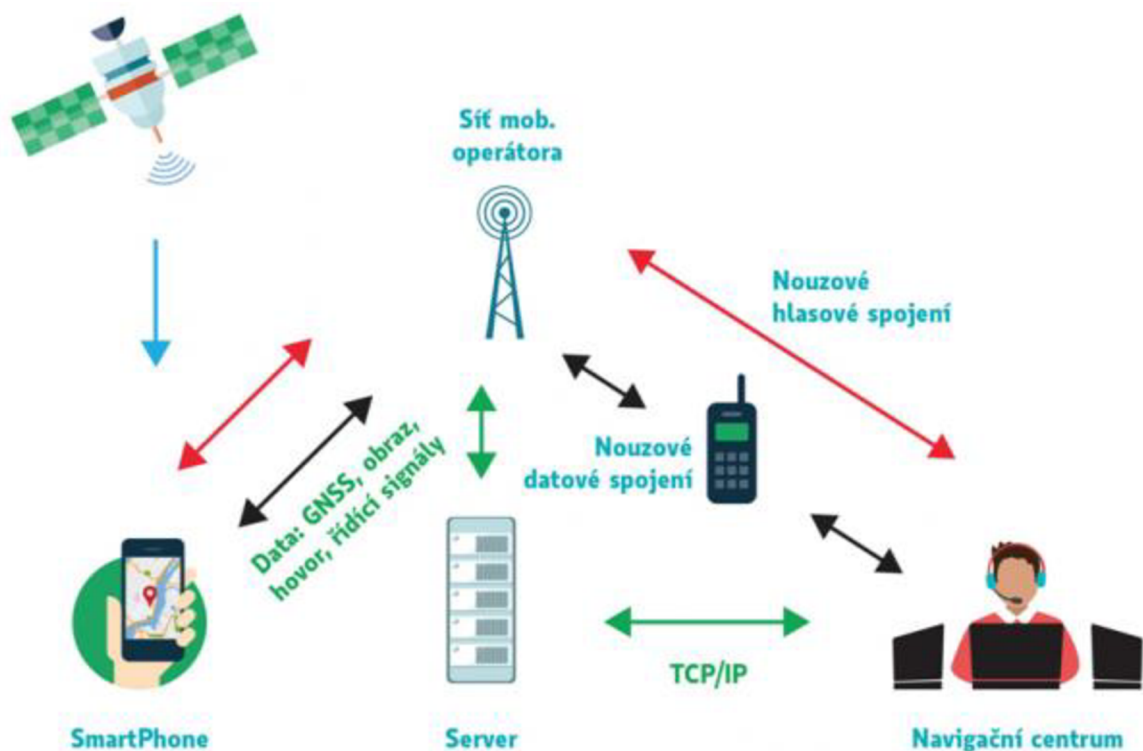


4.4 Navigačního středisko

Historie navigačního centra se datuje od roku 2006, kdy byl systém předán k používání od ČVUT FEL. První systém využíval údaje o poloze uživatele pomocí GPS, a GSM/EGPRS komunikační jednotku pro spojení se serverem. K tomuto serveru bylo připojeno navigační centrum, které v nouzovém případě mohlo kontaktovat uživatele pomocí mobilního telefonu. [12]

S postupem času a vývojem novějších technologií byl i obnovován tento navržený systém. Hlavní změnou oproti původnímu systému je nahrazení GPS a GSM/EGPRS komunikační jednotky, kterou má nevidomý u sebe, chytrým telefonem s operačním systémem, viz Obr. 2. Tento krok umožňuje poskytování více služeb, jako je například posílání obrazu, řídicí signály, nebo výpočet polohy na externím zařízení. Dále pak je možné k navigaci použít rodinného příslušníka nebo kamaráda. [12]

Obr. 2 Schéma komunikace s navigačním centrem [12]



4.5 Tyflosonar

Tyflosonar detekuje překážky ještě před střetem uživatele s touto překážkou. Hlavním prvkem je ultrazvukový senzor. Senzor vysílá krátké ultrazvukové vlny do okolního prostředí a při jejich odrazu od překážky se vrátí zpět. V případě Tyflosonaru RAY je možné detekovat překážku až do 2,85m a je také vybaven detekcí světla. Uživatel si může sám zvolit, jestli mu více vyhovuje detekce překážky akustická nebo vibrační. Tato pomůcka je pouze doplňkem k bílé holi. [13; 14]

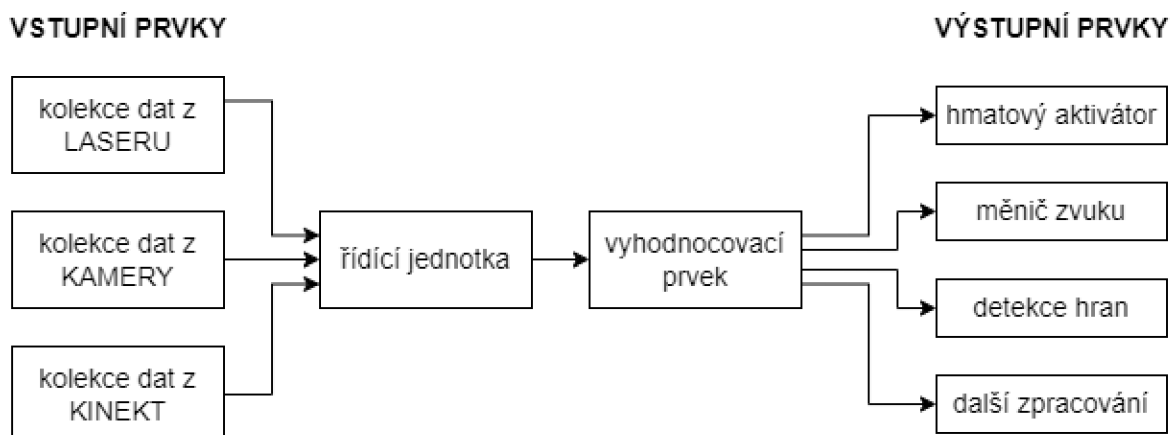
Obr. 3 Tyflosonar RAY



4.6 Taktilní navigační systém

Tento systém je určen pro navigaci v neznámém či téměř neznámém prostředí. Systém byl vyvinut na Technické fakultě ČZU. Řídicí jednotka vyhodnocuje data z laserového systému měření vzdálenosti, z kamerového systému, ale je také možné připojit různé podpůrné systémy, jako Kinect od Microsoft.

Obr. 4 Blokové schéma Taktilního navigačního systému [15]



Vyhodnocená data jsou následně upravená pro další komponenty systému. Propojení jednotlivých komponent je znázorněno na obrázku Obr. 4. Systém je napájen z baterie, která je umístěna v pouzdru v oblasti pasu uživatele. Při návrhu byl tedy brán zřetel na minimální spotřebu elektrické energie. [15]

Informace o prostředí jsou nevidomému předány hmatovým aktivátorem umístěným na předloktí, nebo akusticky. Uchycení hladového aktivátoru na předloktí je řešeno 2 pásy, viz Obr. 5. Pásy jsou dostatečně široké, aby nedocházelo k nepříjemnému vyvíjení tlaku na pláži. Hmatový aktivátor obsahuje 9 malých elektromotorů, na jejichž hřídeli je nasazen excentrický mechanismus, který zajišťuje vibrace při roztočení elektromotoru. Motory jsou rozmístěny v matici 3x3 a do budoucna je plánováno rozšířit počet elektromotorů na 5x5. Vibrace jsou následně vnímány hmatovými receptory uživatele.

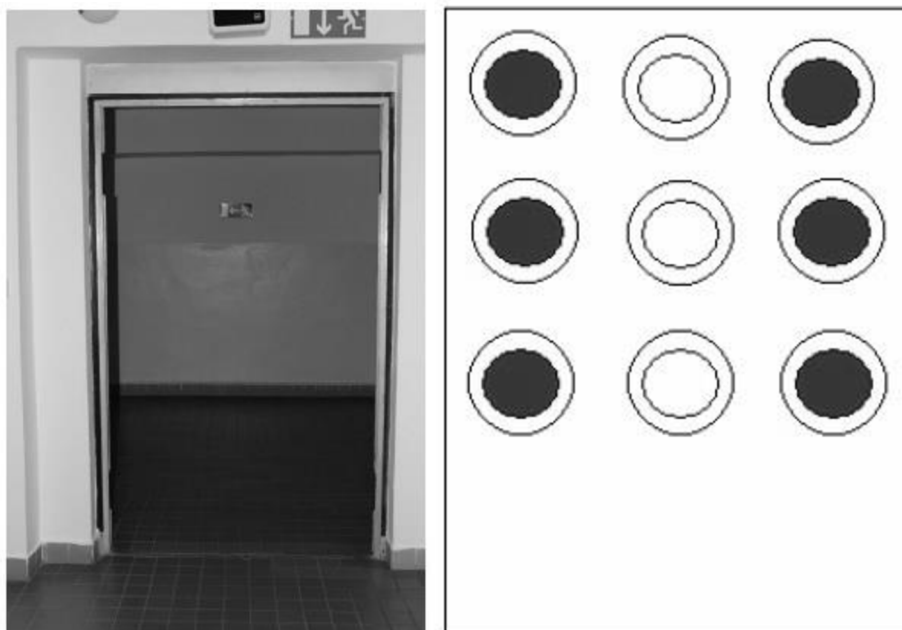
Další možností, jak předat informaci o okolním prostředí u toho systému, je použití předem nahraných hlasových povelů. Alternativně je také možné použít akustický tón s proměnnou frekvencí.

Obr. 5 Uchycení hmatového aktivátoru na paži [15]



Kamera je umístěna na hlavě uživatele a nasměrována tak, aby snímala jeho zorné pole. Může zaznamenávat video nebo fotografie podle nastavení mikroprocesoru. Snímky jsou následně odesílány řídicí jednotce, která následně pomocí speciálního softwaru detekuje hrany v prostoru a zpracuje informace pro výstupní zařízení. Aktivace elektromotorů ve hmatovém aktivátoru na základě snímaného obrazu je znázorněna na Obr. 6. Aktivní motory jsou znázorněny černým kolečkem.

Obr. 6 Záznam kamery a aktivace elektromotorů [15]



4.7 Haptické mapy

Tento systém je veřejnosti dostupný od roku 2014. Na tomto projektu pracovali Teiresiás, Elsa a Mapy.cz. Pomocí speciálního algoritmu jsou klasické mapy, na webovém portálu Mapy.cz, konvertovány na mapu haptickou. Na webové stránce je potřeba kliknout na „změnit mapu“ a vybrat možná „haptické mapy“. Po změně typu zobrazení bude mapa rozdělena na obdélníkové části. Oddalováním nebo přibližováním mapy lze přepínat mezi třemi měřítky. Následně je potřeba kliknout na jednotlivé rozdělené části, které dotyčný chce vytisknout. Vybrané části se stáhnou v souboru typu zip. Soubor obsahuje vybrané části mapy v obrázkovém formátu PNG. K mapám je také možné si stáhnout legendu, která je také v obrázkovém formátu PNG. [16]

Obr. 7 Mapa čitelná hmatem [16]



Tyto obrázky je možné vytisknout na jakékoliv tiskárně. Aby ovšem mohly být použitelné pro nevidomé, je třeba je vytisknout na mikrokapslový papír. Následně pak mapu vytištěnou na tomto speciálním papíře je potřeba vložit do speciálního stroje, který pomocí infračerveného záření udělá mapu hmatatelnou. Infračervené záření zahřeje inkoust na papíře a ten následně vystoupne. Toto speciální zařízení se nachází ve středisku Teiresiás nebo Elsa, ale i v dalších institucích. Mapy je též možné vytisknout na 3D tiskárně. Vytištěné a vystouplé mapy lze snadno skládat přiložením k sobě, jelikož na sebe navazují. Takto konvertovat mapy je možné v rámci Evropy a blízkého okolí, přibližně 17 milionů kilometrů čtverečních. Projekt je postupně rozšiřován i na další kontinenty. [16]

4.8 VPN – vysílačka pro nevidomé

Vysílač PN 01, viz Obr. 8, je vybaven šesti tlačítky. Každé z těchto tlačítek má předem definovanou funkci. Ke správné funkci je důležité, aby v dosahu vysílačky byl maják, který po zmáčknutí tlačítka spustí hlasové povely. Funkce jednotlivých tlačítek je popsána v tabulce Tab. 3. Vysílačka je vybavena akustickou zpětnou vazbou pro uživatele. Po stisku tlačítka zazní jasný krátký tón, aby nevidomý věděl o jejím stisku. Pokud tón byl složen ze dvou pípnutí, je stav baterie dobrý. Jestliže kapacita baterie je nízká a je potřeba její výměna, zazní po stisku tlačítka

Obr. 8 Vysílačka VPN 01 [17]



pouze jedno pípnutí. Dlouhým držením tlačítka 2 až tlačítka 6 je vyvoláno spuštění akustických informací majáku po dobu držení tlačítka. [17]

Dosah vysílačky je minimálně 40 m, v případě dobrých okolních podmínek může být dosah až 150 m. Pro tento komunikační systém je Českým telekomunikačním úřadem vyhrazená frekvence 86,790 MHz. [18]

Tab. 3 Funkce tlačítek [17]

Tlačítko 1	Úvodní znělka – obsahuje informace o poloze, budově - krátký stisk – spuštění znělky jednou - dlouhý stisk – spuštění znělky v nekonečné smyčce. K ukončení je potřeba stisk jakéhokoliv tlačítka na ovladači
Tlačítko 2	Doplňkové informací – jasný popis situovaného uspořádání dveří, eskalátoru, vypínačů, schodů atd.
Tlačítko 3	Informace o směru a čísle linky v MHD
Tlačítko 4	Je použito, když chce nevidomý nastoupit do MHD. Souprava otevře všechny dveře. Jsou jím vybaveny pražské tramvaje typu 14T a 15T a soupravy Českých drah řady 471 CityElefant, RegioNova a další.
Tlačítko 5	Akustická signalizace na přechodu a železničním přejezdu v obcích
Tlačítko 6	Akustická signalizace informačních panelů (příjezd a odjezd MHD, autobusů a vlaků)

4.9 Orientační maják

Orientační majáky se používají ke zvukové identifikaci polohy nevidomého. Nejčastěji jsou umístěny na veřejných budovách nebo jejich částech, jako jsou např. vchody do budovy, průchody v budově, vestibuly metra, a zastávky veřejné dopravy atp. [18]

Akustický orientační maják (AOM)

Tento typ orientačního majáku byl jeden z prvních používaných modelů. Oproti svým nástupcům je značně jednodušší. Byly umístěny nad střed průchodů a vydávaly pouze akustické trylky. Trylek „BRLM“ označoval vstup z ulice do podchodů, a tím pádem schodiště. Trylek „I-A“ označoval úrovně vstupy do objektů a Umístění turniketů v metru. Spuštění těchto majáků bylo vyvoláváno tlačítkem číslo 1 na dálkovém ovladači. [18]

Orientační zvukový modul (OZM)

Jedná se o novější provedení akustického orientačního majáku. Může být navíc vybaveno relé pro silový kontakt, nebo datovou sběrnici RS-485. [19]

Digitální hlasový maják (DHM)

Z tohoto typu majáku mohou být vyvolány až 2 hlasové fráze. Fráze vyvolávaná tlačítkem číslo 1 nejčastěji obsahuje název budovy a případně kde se daná budova nachází. Tlačítkem číslo 2 je pak vyvolána fráze obsahující stručný popis objektu nebo popis trasy z tohoto místa. [18]

Orientační hlasový maják (OHM)

Tento nejnovější typ majáků může obsahovat až 128 hlasových frází. Vzhledem k tomu, že vysílačky VPN mají pouze omezený počet tlačítek, tak se nejčastěji používají pouze 3 fráze. První i druhá fráze jsou shodné s frázemi DHM. [18]

4.10 Programy pro smartphone

V dnešní době spousta lidí používá chytrý telefon k různým aktivitám. Případně i k navigování, když jedeme někam na neznámé místo. V dnešní době existuje spousta aplikací k navigování. Navigovat zvládnou dokonce i aplikace jako Google mapy nebo mapy.cz. V následujícím odstavci je popsána aplikace RightHear, která dokáže navigovat nevidomého i uvnitř budov, kde jsou nainstalovány příslušné majáky. Není to ovšem jediná aplikace k navigování zrakově handicapovaných. Je možné také používat Cydalion, Lazarillo, Ariadne GPS, iMove, Loadstone GPS, nebo BlindSquare. Každá z těchto aplikací funguje trochu rozdílným způsobem a využívá různá podkladní data ke své funkci.

4.10.1 RightHear

RightHear byla spuštěna v roce 2015. Funkce systému je složená ze tří základních komponent. První komponentou je řídicí panel, který je dostupný online. Druhou komponentou jsou majáky, které využívají technologie bluetooth nízkoenergetických senzorů Beacons. Ty umožňují přehrávání hlasových informací v místě umístění. Poslední komponentou je aplikace pro chytré telefony, kterou nevidomý musí mít nainstalovanou k používání tohoto systému. Aplikace je pro systémy iOS a Android ke stažení zdarma. [20]

Obr. 9 zobrazení aplikace RightHear (25)



Celý systém je velmi podobný již zmíněnému jinému systému vysílaček VPN a majákům rozmístěných po městě s nahranými akustickými hláškami. Rozdílné jsou však použité technologie. Tento systém funguje na principu rozmístěných majáků, např. v restauraci, s technologií bluetooth. Chytrý telefon s technologií bluetooth potom rozezná majáky a pomocí integrované hlasové funkce přečte hlášky uživateli. Pro systém iOS je to VoiceOver a pro systém Android je to TalkBack. [20]

Na řídicím panelu mohou uživatelé upravovat informace, přidávat nahrávky v různých jazycích a získávat různé informace z analýzy. Mohou také poskytnout zpětnou vazbu, kterou mohou vylepšit orientaci dalším uživatelům. Může se jednat například o novou otevírací dobu, nové rozmístění stolů v restauraci, nebo nové uspořádání produktů v obchodě. Řídicí panel také obsahuje standard Wayfind pro psaní zvukových zpráv. Ten navádí uživatele k napsání zvukové zprávy dle standardů. [20]

Nízkoenergetické majáky mají výdrž baterie až 4 roky. Ke kontrole stavu baterie, kontrole dat, programování jsou připojené k sítí nepoužívající technologie wifi, GPS ani žádnou podobnou. Majáky jsou velmi malé, a mohou se tedy umístit téměř kamkoliv, aniž by narušovaly vizuální stránku interiéru. [20]

Pokud telefon zachytí signál některého z majáků, informuje uživatele vibrací. Dvojklikem se spustí zvuková zpráva, informace o poloze a informace o dané oblasti. Uživatel si může v případě nutnosti překlíknout na funkci kompasu. Tím získá informaci o natočení telefonu. Pro případ nouze je ve spodní části displeje Možnost spustit službu Call to Help. [20]

4.11 Zhodnocení

Většina historických systémů byla poměrně těžká a uživatel měl na sobě různě rozmístěné komunikační části daného systému. V dnešní době tyto staré systémy je možné nahradit chytrým telefonem, jelikož mají dostatečný výkon i technologie integrované v sobě. Jedná se především o GPS, jakékoliv formy připojení k internetu, ale také nízký příkon elektrické energie celého zařízení. Poměrně velmi mobilním zařízením bylo používáno v projektu NOPPA. To ovšem nebylo tak sofistikované jako předchozí systémy, a hlavně nedokázalo navigovat uživatele v prostoru.

Nejpoužívanější a pravděpodobně i nejpraktičtější pomůckou pro samostatný pohyb je bílá hůl. Je možné ji použít ve vnitřních i vnějších prostorách. Ke své funkci nepotřebuje žádný zdroj elektrické energie, kromě vylepšené varianty se zabudovanou elektronikou. Vylepšená varianta bílé hole ovšem otevírá nové a pohodlnější možnosti užívání moderních technologií. Při spárování s chytrým telefonem lze snadno ovládat funkce panelem tlačítek integrovanými na holi. Což přináší uživateli rychlejší reakce na podněty, které kolem něj mohou nastat. Velkým přínosem by bylo, kdyby bylo možné holí ovládat funkce různých aplikací, které jsou určeny pro navigování nevidomých osob. Uživatel by byl schopný relativně rychle přepínat mezi aplikacemi, jelikož existují aplikace určené pro navigování pouze ve vnitřních prostorech nebo venkovních prostorech.

5 Popis kritických komponent navrženého systému

Následující kapitole budou rozebrány teoretické základy pro určení polohy v bezdrátových sítích. Dále je popsán mikroprocesor ESP8266 a dvě provedení, které se budou používat v návrhu systému. Další důležitou komponentou je zvukový modul, který bude pouštět zvukové povely pro nevidomého.

5.1 Určení polohy v bezdrátové síti Wi-Fi

Bezdrátové sítě Wifi jsou obvykle tvořeny z pevných přístupových bodů (AP) a mobilních zařízení. Síť zprostředkovává komunikaci mezi zařízeními připojenými v jedné síti. Pevné přístupové body jsou síťové prvky, které jsou obvykle na jednom místě, příkladem může být router. Tento síťový prvek je velmi často používán v domácnostech, letištích, nemocnicích nebo obchodních centrech. Mobilní prvek je mobilní telefon, notebook, IP kamera atd. Wifi je standardizována pod označením IEEE 802.11. Od roku 1997, kdy byla vydána první verze, až do dneška bylo vydáno několik dalších standardů. Každá z těchto nových verzí je rozlišena až dvěma písmenky za označením standardu IEEE 802.11, viz. Tab. 4. V tabulce nejsou uvedeny všechny standardy, které již existují. Bezdrátové sítě Wifi jsou provozovány ve vysokofrekvenčním pásmu 2,4 GHz a 5 GHz. [3]

Tab. 4 Přehled standardů IEEE 802.11 [21]

Standard	Označení	Pásmo [GHz]	Max. rychlost [Mbit/s]
IEEE 802.11a	WiFi 1	5	54
IEEE 802.11b	WiFi 2	2,4	11
IEEE 802.11g	WiFi 3	2,4	54
IEEE 802.11n	WiFi 4	2,4 nebo 5	600
IEEE 802.11ac	WiFi 5	2,4 nebo 5	1 800

Tato navržená bezdrátová síť byla primárně určena k vysokorychlostnímu připojení k internetu, nikoliv k určování polohy. V tabulce Tab. 5 jsou uvedeny různé technologie a jejich přesnost při hledání polohy hledaného prvku. Výpočet vzdálenosti mobilního zařízení od AP je založeno na TOA nebo TDOA. Na zařízeních jsou většinou instalovány všesměrové antény, což ztěžuje určení polohy. [3]

Poloha na základě známé polohy

Jedná se o jednoduchý princip, kdy poloha AP je neměnná, a tudíž jeho poloha je známá. Předpokládá se, že připojené mobilní zařízení k tomuto AP je v jeho blízkosti, a tak mu je přiřazená přibližná poloha AP. Velikost chyby takto určené polohy vzrůstá s vyšším dosahem signálu z AP. [3]

Tab. 5 Přehled přesnosti bezdrátových technologií [22]

Technologie	Přesnost [m]	Využití v prostoru	Požizovací cena
Infračervené záření	0,57-2,3	místnost	nízká
Ultrazvuk	0,01-2	budovy	vysoká
Wifi	1,5	budovy	nízká
Bluetooth	0,3-1	budovy	nízká
ZigBee	0,25	budovy	vysoká
RFID	1-5	místnost	nízká

Poloha podle síly signálu

Dalším způsobem, jak získat polohu mobilního zařízení, je intenzita síly signálu (RSS) z a přístupových bodů. K velké přesnosti výpočtu polohy by bylo potřeba spousta malých AP, což by bylo velmi nákladné. Toto by však bylo velmi vhodné u technologií Bluetooth nebo RFID. [3]

Výpočet RSS je běžný pro většinu zařízení s bezdrátovým připojením, jelikož ve většině případů se z něj posuzuje kvalita signálu, čímž se i upravuje rychlost přenosu dat. Z jednoho vysílače průměrné RSS lineárně klesá se vzrůstajícím logaritmem vzdálenosti d . Zjednodušeně RSS lze definováno vztahem: [3]

$$RSS = P - K - 10\alpha \log_{10} d \text{ [dBm]} \quad (1)$$

P – vysílací výkon [dBm]

K – konstanta závislá na frekvenci a prostředí [dB]

α – ztrátový exponent dráhy [-]

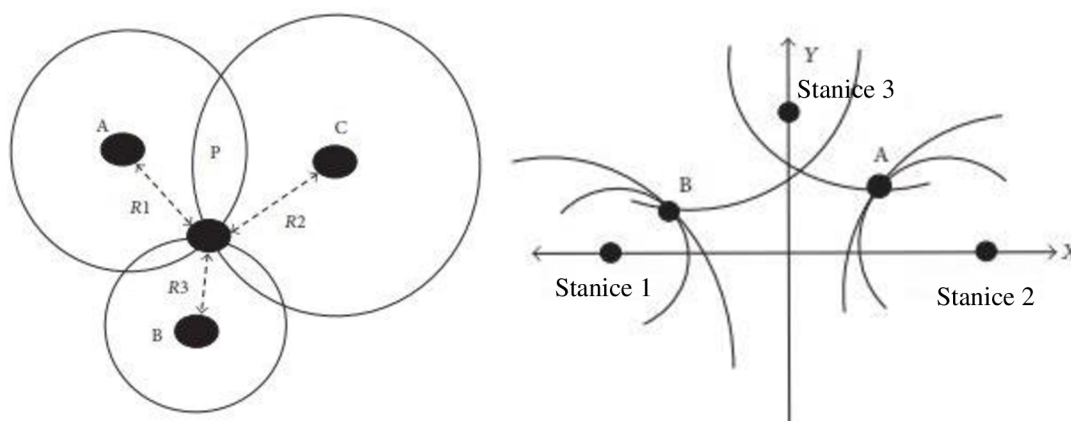
d – vzdálenost [m]

Geolokace TOA/TDOA

Metoda TOA používá k určení vzdálenosti pouze čas přenosu mezi přijímačem a vysílačem. Odhad polohy pomocí TOA je znázorněn na obrázku Obr. 10, kde černé body jsou stanice a kružnice kolem nich znázorňují šířící se signál od nich.

Metoda TDOA používá k určení polohy objektu rozdíl času vysílání dat a přijímání dat na každé stanici v páru. Každé dvě stanice jsou ohniskem hyperboly, jak je znázorněno na obrázku Obr. 10. Bod A a bod B jsou průniky těchto hyperbol. [22]

Obr. 10 schématické znázornění ToA (na levé straně) a TDoA (na pravé straně [22])

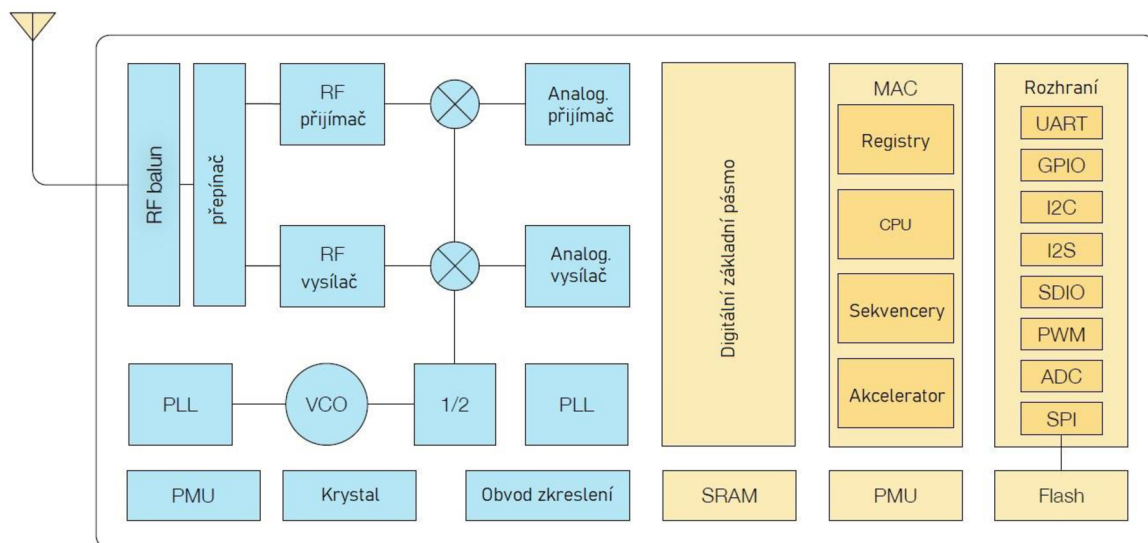


5.2 ESP8266

Firma Espressif Systems navrhla a vytvořila Esp8266 založený na levné platformě System-on-a-Chip (SoC). Jedná se o integrovaný obvod, ve kterém se nachází centrální procesorová jednotka, operační paměť, různé periferní obvody, vstupní i výstupní porty, a to jak digitální, tak analogové viz Obr. 11. Výhodou této platformy je úspora energie, místa a již zmíněných nákladů na výrobu. Jelikož je vše v jednom pouzdru, je zde i větší efektivita na jednotku spotřeby oproti procesoru v PC. [23]

Mikroprocesor obsahuje piny, které lze programově ovládat, označené jako GPIO, a některé nelze programově ovládat. Jsou to napájecí piny, pin pro restartování a pin pro povolení běhu procesoru. Některé piny je potřeba hardwarově nakonfigurovat ke spuštění mikroprocesoru a je potřeba k tomu správně navrhnout elektrický obvod. Existuje mnoho variant vývojových desek s ESP8266 a Pro tuto práci byl vybrán Wemos D1 a ESP12F.

Obr. 11 Blokové schéma ESP8266 [24]



Popis pinů mikroprocesoru

Napájecí napětí pro ESP8266 je 3,3V viz Obr. 12. Pokud je tedy umístěn na vývojové desce, jako je např. Wemos D1, je obvykle přítomen napěťový stabilizátor nebo DC-DC měnič ke snížení napětí z USB.

Sběrnice I2C je používána k připojení dalších periferních obvodů. Příklad periferních obvodů mohou být senzory teploty se sběrnici I2C a integrovaný obvod s digitálními či analogovými vstupy nebo výstupy. Je možné připojit až 128 zařízení. Pro tuto sběrnici jsou vyhrazeny piny s označením SCL, SDA. Jsou podporovány funkce Master/Slave. Piny SCLK, MISO, CS, MOSI lze použít pro sběrnici SPI v režimu Master/Slave.

Přítomen je i jeden pin pro analogový vstup, označen jako ADC0. ESP8266 je vybaveno jedním 10-bitovým AD převodníkem. AD převodník je schopen analogový signál vyjádřit až 1024 digitálními hodnotami.

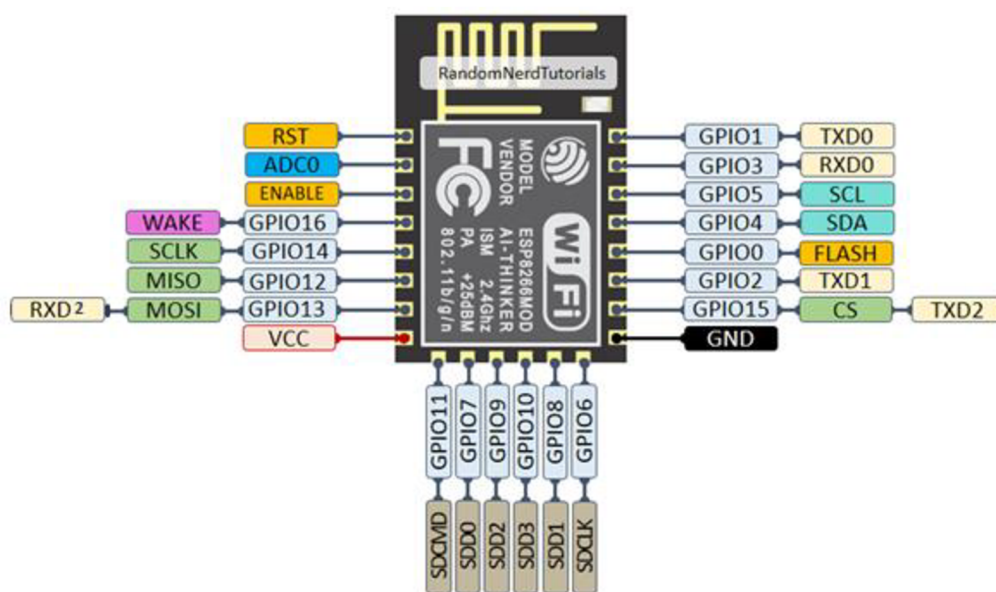
Mikroprocesor je vybaven dvěma sběrnicemi UART. Sběrnice UART0 je určená k asynchronní komunikaci (RS232 nebo RS485) s komunikační rychlostí až 4,5Mbps. Jsou k tomu vyhrazeny piny TXD0, RXD0, RST0 & CTS. Piny TXD0 a RXD0 je možné konfigurací přepojit na TXD2 a RXD2. Sběrnice UART1 má jenom jeden výstupní pin TXD1, je tedy určena pouze k odesílání dat (debugování).

K ESP8266 je možné takto připojit paměťovou kartu SD pomocí (SDIO v2.0) Secure Digital Input/Output Interface. Připojení paměťové karty jsou vyhrazené piny SDD0 až SDD03, SDCMD, SDCLK.

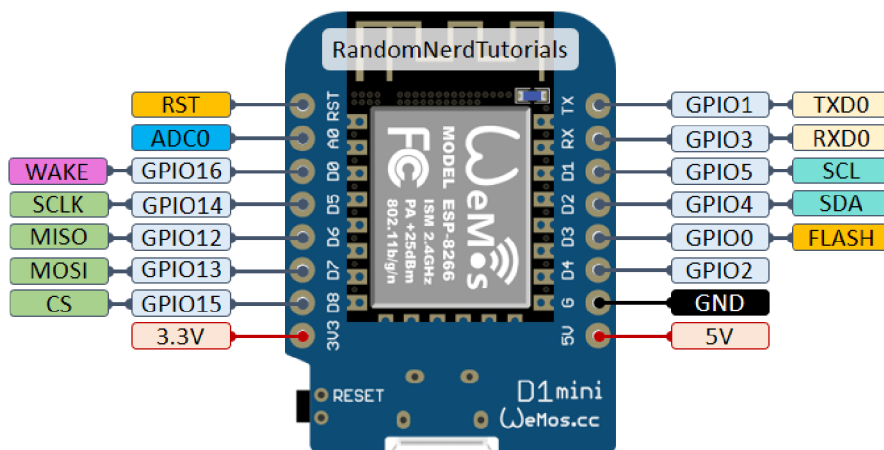
Piny GPIO0 až GPIO16 je možné nakonfigurovat pulzně šířkovou modulací (PWM) pomocí softwarové implementace. Podle dokumentace mikroprocesor je vybaven čtyřmi specializovanými obvody pro PWM, které jsou vyvedeny na piny GPIO12, GPIO15, GPIO14 a GPIO4. Pomocí této modulace je možné řídit otáčky stejnosměrného motorku, nebo jas led diody. Frekvenční rozsah je od 100HZ do 1kHz. [25; 26; 27; 28]

Provedení ESP8266

Obr. 12 ESP8266 verze 12-E pinout [27]



Obr. 13 Wemos D1 Mini pinout [27]



5.3 DY-SV17F (MP3 zvukový modul)

Hlavní částí modulu je mikroprocesor DY1703A viz Obr. 14, který obstarává konfiguraci, 24bit-AD převodník a celkovou funkci modulu. Dále pak modul obsahuje 32Mbits (4Mbytes) flash paměť a zesilovač. Do paměti je možné uložit zvukové soubory typu MP3, nebo WAV, které mohou být nahrány ve vzorkovací frekvenci 8/11.025/12/16/22.05/24/32/44.1/48 kHz. Soubory musí být uloženy pod číselným názvem 00001 až 00255. Nahrání souboru na příslušnou paměť je možné pomocí počítače. K propojení je na modulu Micro-USB port. Po připojení se modul přihlásí jako přenosné úložiště, a není tak potřeba speciální software. V paměti je nahráno 8 demo souboru typu MP3. [29]

Tab. 6 konfigurace DY-SV17F [29]

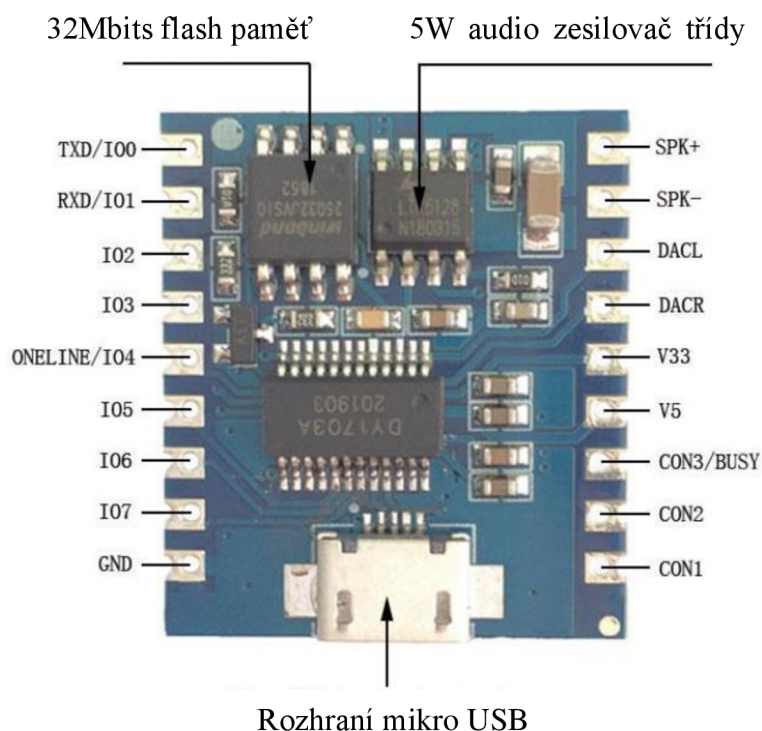
Nastavení konfigurace	Konfigurační piny		
	CON1	CON2	CON3
Kombinovaný mód 0	0	0	0
Kombinovaný mód 1	0	0	1
Samostatný mód 0	0	1	0
Samostatný mód 1	0	1	1
komunikace pomocí UART	1	0	0
komunikace pomocí One-Line			
Standartní MP3 mód	1	0	1

K výstupního zesilovači o výkonu až 5 W je možné připojit reproduktor o impedanci 4-8 Ω . Pro reproduktor jsou vyhrazeny piny SPK+ a SPK-. Spouštění nahraných souborů v paměti je možné několika způsoby, podle konfigurace pinů CON1, CON2 a CON3, viz Tab. 6. Konfigurace se nastavuje pull-up nebo pull-down rezistory. Modul je možné napájet 5 V pomocí mikro-USB konektoru, nebo pinu V5. Komunikace s mikroprocesorem, tedy konfiguračními piny, je možná pro napětí 3.3 V. Pin V33 je pouze výstupním pinem s výstupním napětím 3.3 V. Poslední dva piny DACR a DACL je výstupními linkami pro pravý a levý zvukový kanál. [29]

Tab. 7 Funkce příslušných konfigurací [29]

Nastavení konfigurace	Funkce							
	IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0
Kombinovaný mód 0	Adresace až 255 souborů, spuštění je zahájeno při sestupné hraně a následném vrácení zpět na logickou 1							
Kombinovaný mód 1	Adresace až 255 souborů, spuštění je zahájeno při držení logické 0 na vstupech.							
Samostatný mód 0	Adresace IO0 až IO7, spuštění je zahájeno při sestupné hraně a následném vrácení zpět na logickou 1							
Samostatný mód 1	Adresace IO0 až IO7, spuštění je zahájeno při držení logické 0 na vstupech.							
komunikace pomocí UART							RXD	TXD
komunikace pomocí One-Line				RXD				
Standartní MP3 mód				RPT	EQ	P/P/ MOD E	PREV/ V-	NEXT/ V+

Obr. 14 Popis výstupních pinů modulu DY-SV17F [29]



6 Vývoj systému

Tato kapitola se věnuje samotnému vývoji systému. Nejprve jsou popsány požadavky na navrhovaný systém. Následuje návrh samotného systému a popis dílčích komponent, ze kterého je systém složen. Postup je rozdělen do několika dílčích úloh, které jsou zaměřeny na konkrétní problematiku. Nejprve popisováno nastavení vývojového prostředí Arduino IDE. Následuje a testování komunikace mezi ESP8266 a zvukovým modulem. V dalším kroku je popisován elektrický obvod potřebný k nahrání programů do ESP12F. Následují kapitoly, které popisují vývoj softwaru a fyzickému zapojení elektrických obvodů. V posledním kroku je navržený systém otestován a upraven.

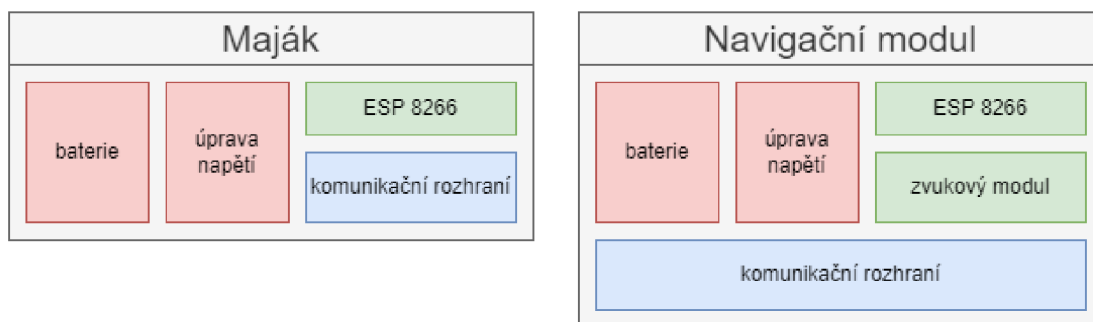
6.1 Požadavky na systém

Navržený systém je určen především pro osoby se zrakovým postižením. Systém ovšem mohou využít i osoby, které mohou být v dané budově poprvé a neorientují se v ní. Komunikační rozhraní s uživatelem musí být navrženo tak, aby bylo snadno použitelné a srozumitelné. Důležité je, aby systém byl vždy funkční a nevypadl ze sítě fixní maják nebo navigační modul. Každý modul bude vybaven hlídáním stavu baterie a komunikační rozhraní upozorní na tento stav. Pokud se tak stane, musí neprodleně dojít k výměně baterií. Navigační modul musí dovést uživatele na určené místo a následně také zpět k vrátnici.

6.2 Návrh systému Verze 1

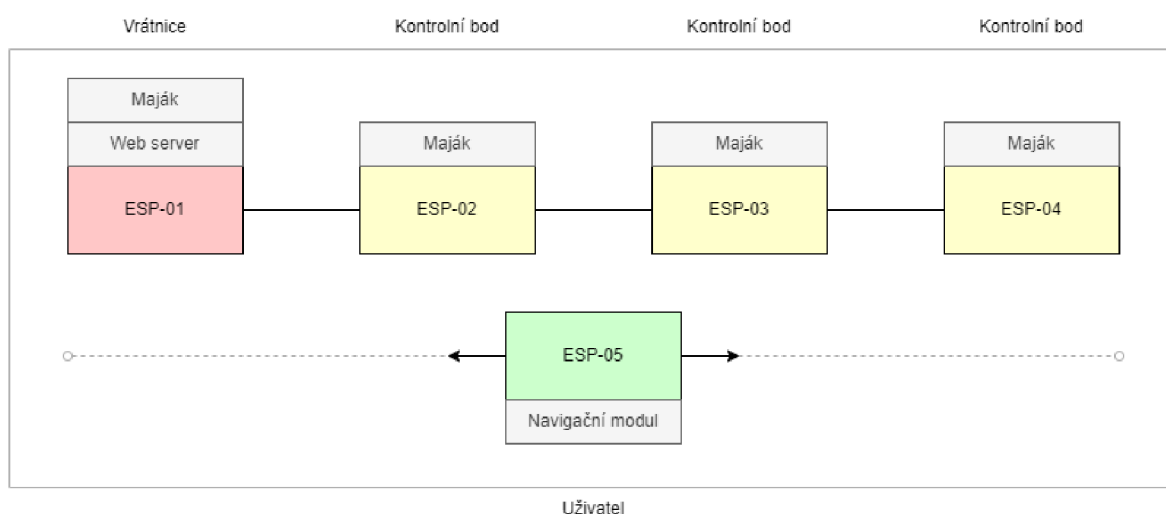
Komunikace mezi moduly, viz Obr. 15, bude vytvořena bezdrátovou sítí používající protokol ESP-NOW. Komunikační rozhraní pro navigační modul bude složeno z reproduktoru, tlačítka a LED. Panel bude napájen ze čtyř baterií AA. Snížení napětí bude zajišťovat stabilizovaný zdroj nebo měnič napětí. Reprodukce bude připojen ke zvukovému modulu na pinech pro výstupní zesilovač. Zvukový modul bude použit jednotlivé zvukové sekvence. Použití konkrétních zvukových sekvencí bude řízeno ESP8266. Tlačítko a LED budou připojeny přímo k ESP8266. Dalším vstupem bude vývod z odporového děliče, který bude určen k hlídání stavů napětí na baterii. Modul bude krátce rozsvěcet indikační diodu, pokud bude nízký stav baterie.

Obr. 15 Blokové schéma majáku a navigačního modulu



Maják budeme mít velmi podobné zapojení jako navigační modul viz Obr. 15. Ovšem nebude zvukový modul, tím pádem ani reproduktor a tlačítko. Speciálním případem bude maják, na kterém bude spuštěn webový server. Tento maják je určen k tomu, aby byl umístěn změn na vrátnici. Jeho komunikační rozhraní bude rozšířeno o několik LED diod, indikující různé stavy připojených majáků v síti. Zároveň bude mít i možnost být napájen externím zdrojem elektrické energie. Dále bude připojen reproduktor, který bude akusticky upozorňovat na problém.

Obr. 16 Rozložení sítě



Navigační modul se bude pohybovat mezi fixními majáky, viz Obr. 16, a jeho poloha bude určena pomocí RSS získané z majáků. Hodnota RSS nás bližších majáků bude vyšší než z majáků, které jsou dál od navigačního modulu. Síla signálu bude také závislá na překážkách v prostředí.

Navigační modul, který uživateli bude zapůjčen na vrátnici, spustí navigaci až po stisku tlačítka. Následně bude hlasovými povely nevidomý navigován k cílovému bodu. V daném bodě se navigace zastaví a bude čekat na opětovné stisknutí tlačítka. Spustí se zvuková

sekvence, která vyzve uživatele k potvrzení zpáteční trasy stisknutím toho samého tlačítka ještě jednou. Následně bude spuštěná navigace ke vrátnici, kde uživatel vrátí navigační modul zpět vrátnému. Pokud bude uživatel v průběhu navigace v nouzi, např. zabloudí, držením tlačítka informuje vrátnici, že potřebuje pomoc.

6.2.1 Nastavení programu Arduino IDE

Pro vývoj softwaru byl použit program Arduino IDE, který je ke stažení zdarma na stránkách „<https://www.arduino.cc/en/software>“. Software po nainstalování neobsahuje potřebné soubory pro vývoj již určených ESP modulů. Je možné stáhnout přímo v aplikaci kliknutím na Boards Manager v levém panelu nástrojů. Do vyhledávače je potřeba zadat ESP8266, a stáhnout příslušný rozšiřující balíček.

Pro komunikaci se zvukovým modulem DY–SV17F po sériové lince je použita knihovna DYPlayerArduino.h, které je možné zdarma stáhnout na webových stránkách „<https://github.com/SnijderC/dyplayer#esp-idf>“. Tato stránka obsahuje i dokumentaci potřebnou k používání této knihovny. Stažený soubor z webových stránek je ve formátu ZIP. Následně je třeba staženou knihovnu importovat do softwaru Arduino IDE (to lze provést otevřením záložky *Sketch/Include Library/Add .ZIP Library* a vybrat stažený soubor).

Napsaný program v tomto editoru je potřeba po dokončení zkompilovat. V okně „*Select board*“ je potřeba vybrat vývojový modul, viz Obr. 17, (Wemos D1 mini nebo ESP12) na který bude program nahrán a vybrat port ve kterém je vývojový modul připojen k počítači.

Do vývojového prostředí je také potřeba nainstalovat knihovny (painlessMesh.h, Ticker.h, ESP8266WiFiMulti.h a Arduino_JSON.h) z karty „Library manager“ knihovny, které budou použity během programování.

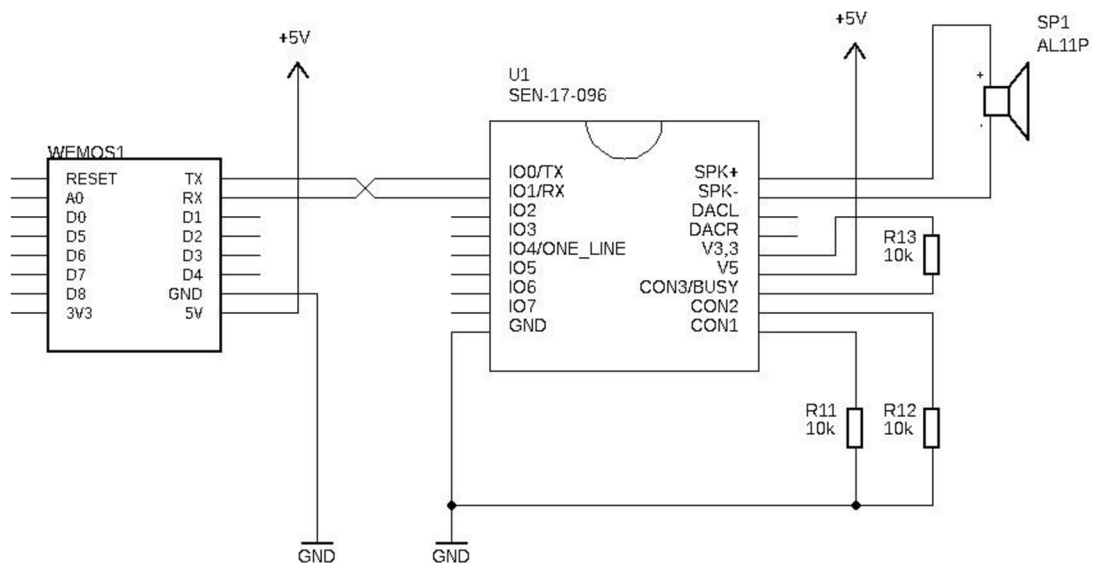
Obr. 17 vývojové prostředí Arduino IDE [30]



6.2.2 Komunikace mezi ESP a MP3 modulem

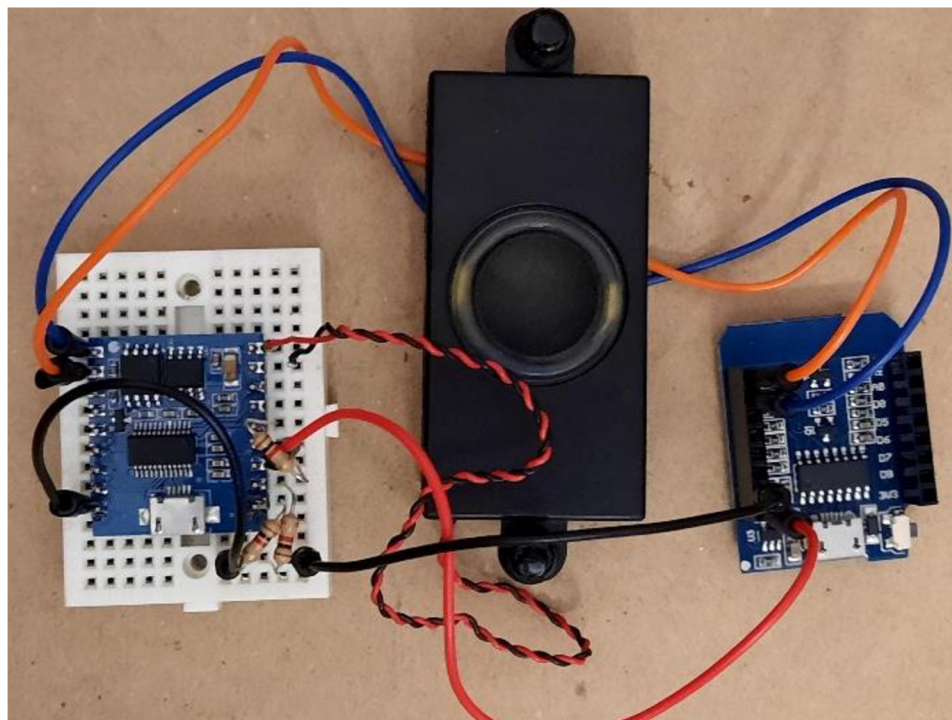
Moduly jsou napájeny 5 V z konektoru mikro USB na Wemos D1 mini viz Obr. 18. Dle dostupných dokumentací jsou oba moduly vybaveny měničem napětí na 3,3 V. Není tedy potřeba upravovat úroveň napětí. MP3 modul je nakonfigurován pro UART komunikaci třemi rezistory R13, R12 a R11. Jako elektroakustický měnič byl zvolen reproduktor ASE06003MR-LW150-R. Realizovaný elektrický obvod je zobrazen na Obr. 19

Obr. 18 Schéma zapojení pro ověření funkčnosti komunikace s MP3 modulem



Použitím importované knihovny je komunikace mezi moduly snadná. Nejprve je navázaná sériová komunikace rychlostí 9600 baudů. Toto je rychlost, na kterou je nastaven MP3 modul. Následuje nastavení hlasitosti, posláni příkazu pro postupné spuštění všech nahrávek v paměti a spuštění přehrávání. Hlavní program postupně spouští nahrané zvukové soubory z paměti viz Tab. 8. Testovací program je zobrazen na Obr. 20.

Obr. 19 Reálné zapojení pro ověření funkčnosti komunikace s MP3 modulem



Tab. 8 Přehled nahraných hlášek ve zvukovém modulu

Soubor	Přehrávaný text	Čas potřebný pro přehrávání [s]
1	jdete správně	3
2	zabočte vlevo	3
3	započte vpravo	3
4	jdete opačným směrem	3
5	spouštím hlasové navigování	4
6	jste v cíli	3
7	nízký stav baterie	3
8	baterie je vybita	3
9	vyhledávají se orientační body	5
10	pomoc přivolaná	3
11	pomoc je na cestě	3
12	prošli jste kontrolním bodem	3

V průběhu vývoje softwaru bylo zjištěno, že připojení zvukového modulu na UART 0 není příliš vhodné. Byl tedy připojen pomocí linky UART 1, kde je použit pouze jeden vodič k posílání zpráv do zvukového modulu. Programu bylo potřeba nakonfigurovat komunikaci na tuto sběrnici.

Obr. 20 Program – testování MP3 modulu

```
#include <Arduino.h>
#include "DYPlayerArduino.h"

// Inicializuje "player"
DY::Player player(&Serial1);

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  player.begin();
  player.setVolume(30); // 100% Volume
  //player.setCycleMode(DY::PlayMode::Repeat); // Prehrat vse
  //player.play();
}

void loop() {
  player.playSpecified(1);
  delay(500);
  player.stop();
  delay(100);
  player.playSpecified(2);
  delay(3000);
}
```


6.2.3 Nastavení sítě

První použitý protokol pro přenos dat mezi Wemos D1 moduly a ESP12 byl ESP-NOW. Komunikaci mezi zařízeními v takto nastavené síti je možno nastavit několika způsoby. Pro tento systém byl vybrán komunikační model, kde všechny kontrolní body (majáky kromě web serveru) a navigační modul jsou nastavené jako Slave. Posílají informace v datové struktuře, viz Tab. 9, zařízení na vrátnici, které je nastaveno jako Master.

Tab. 9 Definovaná datová struktura pro komunikaci ESP_NOW

Definice struktury dat	popisek
byte id;	unikátní označení ESP
byte batt;	stav baterie
bool tl_nouze;	volání o pomoc uživatelem

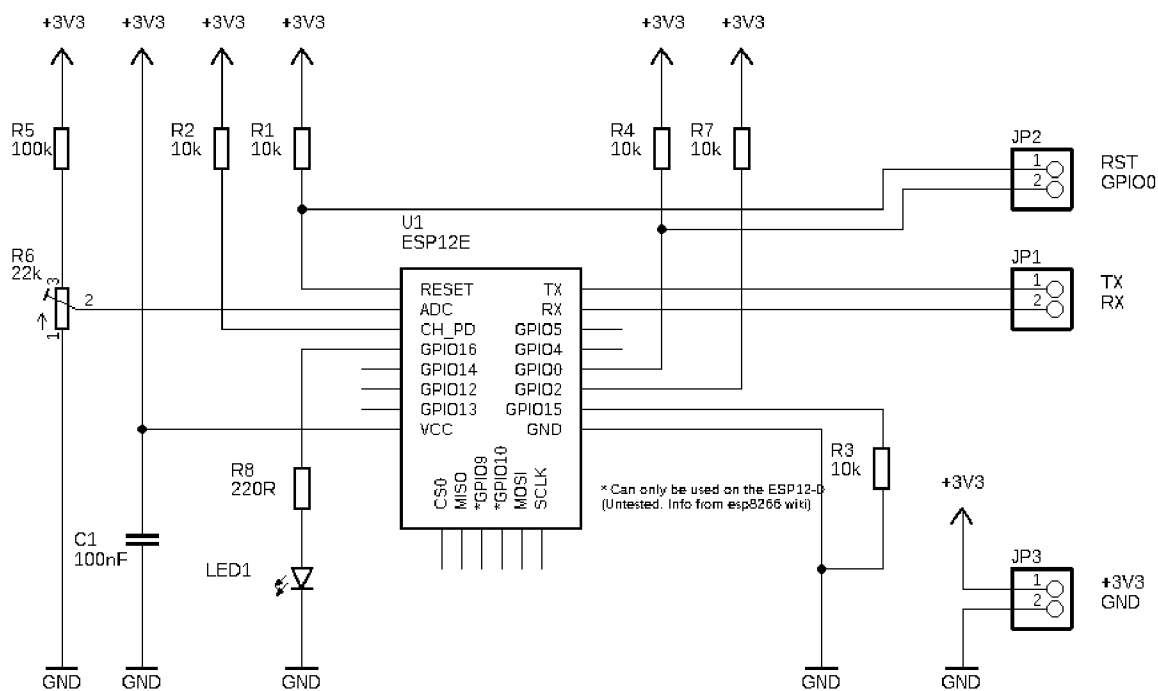
Každý maják kromě sítě ESP-NOW je konfigurován jako AP stanice. Každý maják má definované vlastní SSID, které se skládá z textového řetězce „ESP:“ a jeho MAC adresy. Tím je snadno rozeznatelný pro navigační modul od ostatních AP stanic v okolí. Tady akční modul pomocí vyhledávací funkce pro vyhledání SSID je schopen zjistit i sílu signálu RSSI, kterou by následně použil k určení polohy.

Testováním bylo zjištěno, že tento komunikační protokol není posílán skrz jednotlivé prvky k cílovému bodu. Tudíž pro tuto aplikaci není příliš vhodný a byl nahrazen protokolem ESP-MESH. Bylo nastaveno jiné složitější heslo i jiné SSID, které je stejné pro každé zařízení „NavPrNevidim“.

6.2.4 Zapojení na univerzálním DPS

Navržené schéma zapojení, viz Obr. 21, bylo testováno na univerzální DPS pro ověření funkčnosti. Nahrávání programů do RAM paměti ESP je možné pomocí linky UART. Při nahrávání programu je potřeba některé piny speciálně nakonfigurovat. Jedná se o piny RESET a GPIO0. Konfiguraci těchto pinů zajišťuje dvojice tranzistorů, které jsou na obrázku Obr. 22. Do převodníků CP2102 (USB na UART) jsou připojeny 4 vodiče TX, RX, RTS, DTR. Pro převodník bylo potřeba stáhnout a nainstalovat ovladače pro systém, které jsou dostupné na stránkách výrobce. Reálné zapojení je zobrazeno na obrázku Obr. 43.

Obr. 21 Schéma zapojení – test 1

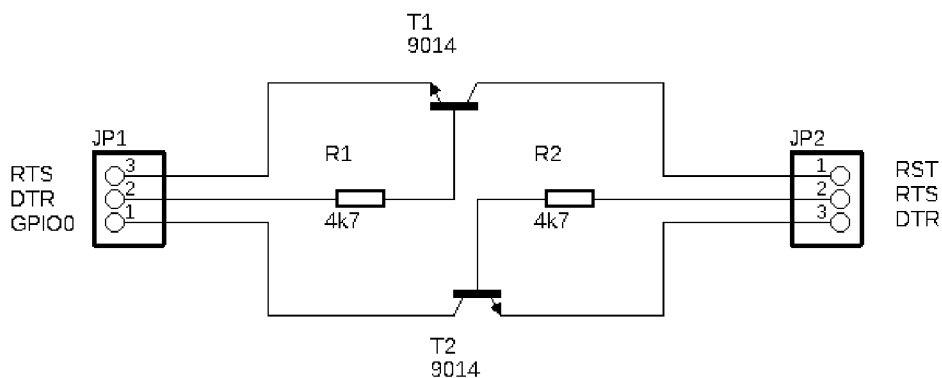


GPIO0:
1 - Running mode
0 - Download mode

INA0 rozmetí 0-1V

Stav baterie:
 $UA0 = (VCC/110\ 000) * 10\ 000 [V]$

Obr. 22 Schéma zapojení – test 2



6.2.5 Program pro navigační modul

Po zapnutí navigačního modulu nejprve dojde k inicializaci základních funkcí viz Obr. 23. Jde především o nastavení portů, sériových linek, přerušení a časovače.

Funkce `setTimer1()` je složená z několika jednoduchých příkazů. Nastavuje hodnotu časovače na 0, příznak přerušení při ukončení časovače, odkaz na podprogram, který bude spuštěn. Nedílnou součástí bylo nastavení počítadla na nejvyšší možnou hodnotu počítání.

Obr. 23 Program pro navigační modul – Setup

```
//-----Setup-----
void setup() {
  Serial.begin(115200); // UART0 - GPIO1 (TX) and GPIO3 (RX)
  Serial1.begin(9600); // UART1 - GPIO2 (TX)

  //nastaveni IN/OUT
  pinMode(pin_batt, INPUT);
  pinMode(tlacitko, INPUT);
  pinMode(led_R, OUTPUT);
  pinMode(led_O, OUTPUT);
  pinMode(led_Z, OUTPUT);
  pinMode(led_Debug, OUTPUT);
  pinMode(repro, OUTPUT);

  // definoovani pinu
  const byte pin_batt = A0;
  const byte led_Z = D7;
  const byte led_O = D6;
  const byte led_R = D5;
  const byte led_Debug = D0;
  const byte repro = D1;
  const byte tlacitko = D2;

  //nastaveni preruseni - spusteni pouze na vzestupnou hranu
  attachInterrupt(digitalPinToInterrupt(tlacitko),
  tlacitko_prepuseni_R, RISING);

  setTimer1(); //nastaveni casovace

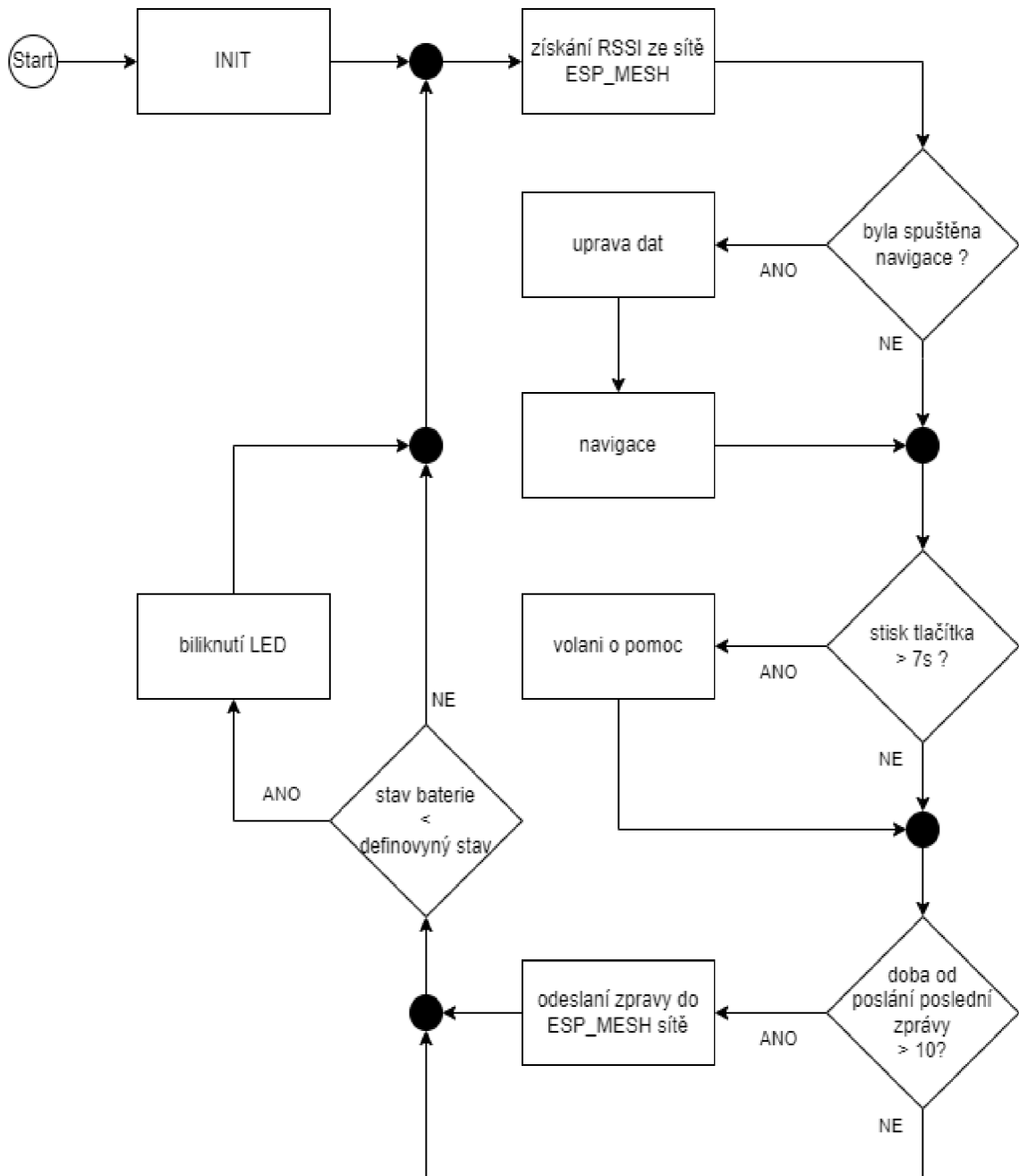
  // nastaveni zvukoveho modulu
  player.begin(); // zahajeni komunikace
  player.setVolume(30); // 100% Volume
}
```

Následuje spuštění hlavní programové smyčky, viz Obr. 24, ve které dochází ke skenování zařízení se specifickou MAC adresou. Dále zde dochází ke spuštění potřebných funkcí ke správnému běhu navigace, posílání potřebných dat do již vytvořené sítě majáků a detekci

dlouho stisknutého tlačítka, které následně informuje pověřenou osobu o její potřebnou pomoc.

Skenování wifi sítí okolí navigačního modulu je prováděno pomocí funkce scanNetworks(), viz Obr. 25, která je součástí knihovny ESP8266WiFi.h. Funkce vrátí hodnotu vyjadřující počet nálezů sítí. Tato hodnota je následně vložena do funkce skenARSSI.

Obr. 24 Životní cyklus programu pro navigační modul



Obr. 25 Program pro navigační modul – hlavní programové smyčka část 1

```
void loop() {  
    byte pocet_AP = WiFi.scanNetworks();    //spusteni skeni Wifi  
    skenARSSI(pocet_AP);                    //vyhledani ESP  
}
```

Nejprve je testováno, jestli nalezený počet sítí není nulový viz Obr. 27. Pokud nulový není, je spuštěn FOR cyklus k průchodu všech nalezených sítí. Při každém průchodu je porovnáváno SSID (název sítě) s hledanou Wifi sítí. V tomto případě název hledané sítě je „NavPrNevidim“. Pokud název odpovídá, je uložena BSSIDSrt (MAC adresa) tohoto zařízení a její RSSI (síla získaného signálu).

Při výpisu do konzole bylo zjištěno, že ukládaná hodnota MAC adresy je rozdílná oproti původní hodnotě. Proto ještě před jejím uložením je potřeba první hexadecimální část upravit. Nakonec jsou všechny naskenované sítě z paměti vymazány, aby zbytečně nezabíraly místo.

Obr. 26 Program pro navigační modul – najitVzdalenost()

```
int16_t najitVzdalenost(byte n, uint8_t b0, uint8_t b1, uint8_t  
b2, uint8_t b3, uint8_t b4, uint8_t b5, int16_t b6){  
    int16_t vzdalenost = 120;  
    for(int i = 0; i < n; ++i){  
        if((b0==majak[i][0]) && (b1==majak[i][1]) &&  
            (b2==majak[i][2]) && (b3==majak[i][3]) &&  
            (b4==majak[i][4]) && (b5==majak[i][5])){  
            // odedcteni hodnot RSSI -> -50 -(-120)= 70 -> priblizuji  
            // -60 - (-50) = -10 -> oddaluji  
            vzdalenost = majak[i][6] - b6;  
            // ulozeni pro dalsi vypocet  
            temp = majak[i][6];  
        }  
    }  
    return vzdalenost;  
}
```

```

//-----WIFI-sken a RSSI-----
void skenARSSI(byte n){
    pocet_majak = 0;           // nulovani pocitadla
    if (n == 0) {
        if(DEBUG)
            Serial.println("nebyla nalezena zadna sit");
    } else {
        if(DEBUG)
            Serial.println("site nalezeny");
        for (byte i = 0; i < n; ++i) {
            String SSID = WiFi.SSID(i);
            String BSSIDstr = WiFi.BSSIDstr(i);
            int16_t RSSI = WiFi.RSSI(i);
            delay(2);
            //vyhledani zarizeni v siti ESPnavigace
            if (SSID.indexOf(MESH_PREFIX) == 0){
                int mac[6];
                if( 6 == sscanf(BSSIDstr.c_str(),
"%02x:%02x:%02x:%02x:%02x:%02x", &mac[0], &mac[1], &mac[2],
&mac[3], &mac[4], &mac[5] ) ) {
                    for (int ii = 0; ii < 6; ++ii ) {
                        majak[pocet_majak][ii] = (uint8_t) mac[ii];
                    }
                    // KOREKCE: nekde dochazi z kamene MAC adresy
                    majak [pocet_majak][0] = majak[pocet_majak][0] - 2;
                    majak[pocet_majak][6] = RSSI;
                    pocet_majak = pocet_majak+1;// aktualizace pocitadla
                }
            }
        }
    }
    WiFi.scanDelete(); //vymazani siti z pameti
}

```

Program má uložené MAC adresy konkrétních majáků, kolem kterých musí projít. Proto k navigování jsou potřeba hodnoty síly signálu dvou majáků. Pokud uživatel půjde od vrátnice, pro navigování jsou vloženy hodnoty RSSI následujícího majáku a majáku u vrátnice. Aktuální hodnota RSSI je odečítána od hodnoty předchozí. Tato hodnota je nulová, když

uživatel navigačního přípravku se nepohybuje. Kladná hodnota znamená, že uživatel se pohybuje směrem k následujícímu majáku. Pakliže je hodnota záporná, uživatel se vzdaluje od nadcházejícího majáku.

V případě, že maják není nalezen, hodnota jeho RSSI je rovna hodnotě 120, viz Obr. 26. V takovémto případě v následující funkci pro úpravu dat je podmínka, která v případě této hodnoty nespustí navigování a nahlásí jejich chybu.

Při testování bylo zjištěno, že hodnota RSSI je s každým skenem rozdílná a v různých intervalech dochází ke skokové změně. Proto nejprve byla hodnota průměrována z posledních 5 hodnot. Tento postup ovšem neodfiltroval náhlé skoky. Proto program sleduje aktuální hodnotu RSSI, a pokud je skok větší než definovaná konstanta (PRUMER_SKOKY = 15), je nahrazena hodnotou předchozí. Výsledky těchto úprav dat jsou zobrazeny na obrázcích Obr. 29 a Obr. 28. Spuštění navigace je umožněno až po dokončení průměrování validních hodnot RSSI.

Obr. 28 Průměr hodnot

```
14:37:49.745 -> site nalezeny
14:37:49.745 -> DEBUG: prumer1 -5
14:37:49.745 -> DEBUG: prumer2 0
14:37:49.745 -> DEBUG: navigace: spatny smer
14:37:52.066 -> site nalezeny
14:37:52.066 -> DEBUG: prumer1 0
14:37:52.066 -> DEBUG: prumer2 0
14:37:52.066 -> DEBUG: navigace: poloha se nemeni
14:37:54.360 -> site nalezeny
14:37:54.360 -> DEBUG: prumer1 0
14:37:54.360 -> DEBUG: prumer2 0
14:37:54.360 -> DEBUG: navigace: poloha se nemeni
14:37:56.658 -> site nalezeny
14:37:56.705 -> DEBUG: prumer1 0
14:37:56.705 -> DEBUG: prumer2 0
14:37:56.705 -> DEBUG: navigace: poloha se nemeni
14:37:58.998 -> site nalezeny
14:37:58.998 -> DEBUG: prumer1 0
14:37:58.998 -> DEBUG: prumer2 0
14:37:58.998 -> DEBUG: navigace: poloha se nemeni
14:38:01.293 -> site nalezeny
14:38:01.339 -> DEBUG: prumer1 5
14:38:01.339 -> DEBUG: prumer2 0
14:38:01.339 -> DEBUG: navigace: spatny smer
14:38:03.604 -> site nalezeny
14:38:03.651 -> DEBUG: prumer1 0
14:38:03.651 -> DEBUG: prumer2 0
14:38:03.651 -> DEBUG: navigace: poloha se nemeni
```

Obr. 29 Průměr hodnot a oříznutí špiček

```
22:04:34.061 -> site nalezeny
22:04:34.094 -> DEBUG: prumer1 0
22:04:34.094 -> DEBUG: prumer2 0
22:04:36.381 -> site nalezeny
22:04:36.381 -> DEBUG: desekce spicky RSSI: -32 nahrazeno -1
22:04:36.417 -> DEBUG: prumer1 0
22:04:36.417 -> DEBUG: prumer2 0
22:04:38.699 -> site nalezeny
22:04:38.699 -> DEBUG: desekce spicky RSSI: 32 nahrazeno -1
22:04:38.699 -> DEBUG: prumer1 0
22:04:38.699 -> DEBUG: prumer2 0
22:04:40.989 -> site nalezeny
22:04:41.022 -> DEBUG: desekce spicky RSSI: -21 nahrazeno -1
22:04:41.022 -> DEBUG: prumer1 0
22:04:41.022 -> DEBUG: prumer2 0
22:04:43.278 -> site nalezeny
22:04:43.326 -> DEBUG: desekce spicky RSSI: 21 nahrazeno -1
22:04:43.326 -> DEBUG: prumer1 -1
22:04:43.326 -> DEBUG: prumer2 0
22:04:45.615 -> site nalezeny
22:04:45.615 -> DEBUG: prumer1 -1
22:04:45.615 -> DEBUG: prumer2 0
22:04:47.921 -> site nalezeny
22:04:47.921 -> DEBUG: prumer1 0
22:04:47.921 -> DEBUG: prumer2 0
22:04:50.226 -> site nalezeny
22:04:50.272 -> DEBUG: prumer1 0
22:04:50.272 -> DEBUG: prumer2 0
```

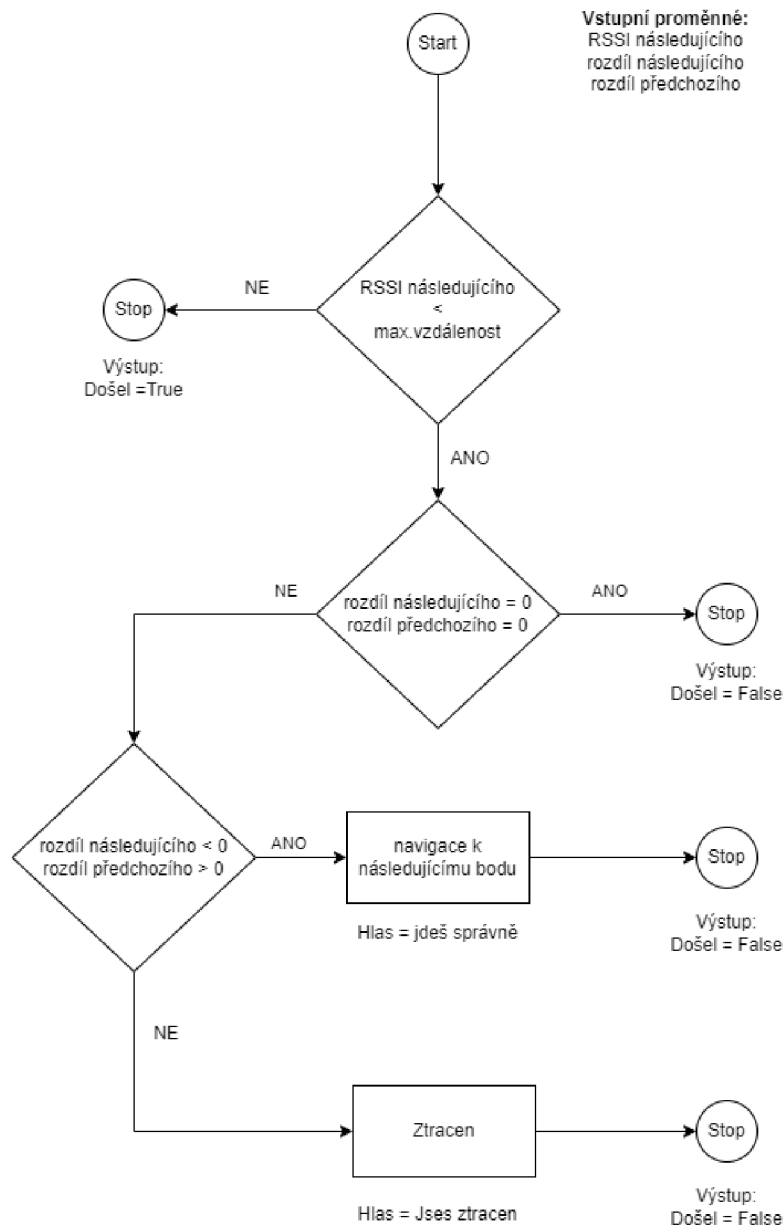
K vyhodnocení směru chůze je použita funkce navigace(). Do parametrů je zadán cílový bod a zpracované hodnoty RSSI, viz Obr. 30. Pokud tato funkce vyhodnotí, že uživatel ještě nedošel k cílovému majáku, výsledkem je FALSE a při dalším průchodu programu jsou do ní nahrány aktuální zpracované hodnoty RSSI. V případě, že se uživatel přiblížil k majáku

dostatečně blízko, jsou do této funkce předány nové hodnoty, nebo je navigace ukončena. Vývojový diagram pro navigaci je znázorněn na Obr. 31.

Obr. 30 Program pro navigační modul – hlavní programové smyčka část 2

```
// navigace k bodu2 -- bod2 predchozi, bod3 nasledujici
if((fin_bod2 == false) && (prumReady1 == true) && (prumReady2 ==
true)){
  fin_bod2 = navigace(bod3[6], prumerVzdalenosti2,
prumerVzdalenosti1);
  if(fin_bod2 == true){
    player.stop();
    delay(1000);
    player.playSpecified(6);
    hlaska = true;
    timer1_write(cas_povoleni_hlasky); //nasatveni casu
    timer_on = true;
    tlacitko_stisknuto = false; //zruseni navigace
    trasa_zpet= true; // nastaveni priznaku pro zpatecni cestu
    if(DEBUG)
      Serial.println("DEBUG: dokončili jste trasu");
  }
}
```


Obr. 31 Vývojový diagram pro navigaci



Interval mezi jednotlivými zvukovými hláškami je nastavován neprodleně po jejich přehrání. Do časovače je nahraná hodnota, jež po dosáhnutí 0 je vyvoláno přerušení kdy je hláška opět povolena. V programu jsou i speciální případy (dokončení trasy, zavolání pomoci), kdy je vynuceno zastavení probíhající hlášky a spuštěna hláška jiná.

Vyvolání stavu „volání o pomoc“ (dlouhé držení tlačítka) je nastaveno kombinací přerušení, viz Obr. 32. Při stisku tlačítka dojde k prvnímu přerušení, kdy je zapamatována hodnota z funkce millis(). Tato funkce vrátí aktuální stav počítadla, které je spuštěno od spuštění mikroprocesoru. Posledním příkazem je nastavení přerušení na sestupnou hranu toho

tlačítka. Pokud je tlačítko drženo po definovanou dobu, dojde ke splnění podmínky. Dojde ke spuštění zvukové hlášky a ukončení navigace. Po uvolnění tlačítka je důležité do proměnné „tlacitko_cas“ vložit maximální hodnotu, aby nedošlo k nechtěnému vyvolání stavu „volání o pomoc“. Jelikož tato proměnná bude v další iteraci programu používána k vyhodnocení podmínky a její nepřepsáním by podmínka byla splněna. Posledním příkazem v tomto stavu je spuštění funkce ZpravaMesh(). Tato funkce nakonfiguruje navigační modul k připojení do sítě NavPrNevidim. Po připojení jsou v krátkých intervalech odesílány zprávy o stavu zařízení v již definované datové struktuře, viz Tab. 9.

Obr. 32 Program pro navigační modul – přerušení

```
//-----Preruseni-----  
ICACHE_RAM_ATTR void tlacitko_prepuseni_R(){  
    tlacitko_stisknuto = true;  
    tlacitko_cas = millis();  
    if(DEBUG)  
        Serial.println("DEBUG:-----: bod:  
tlacitko_prepuseni_vzestup");    // DEBUG  
    attachInterrupt(digitalPinToInterrupt(tlacitko),  
tlacitko_prepuseni_F, FALLING); // nastaveni na sestupnou hranu  
}  
ICACHE_RAM_ATTR void tlacitko_prepuseni_F(){  
    tlacitko_cas = 0xFFFFFFFF;  
    if(DEBUG)  
        Serial.println("DEBUG:-----: bod:  
tlacitko_prepuseni_sestup");    // DEBUG  
    attachInterrupt(digitalPinToInterrupt(tlacitko),  
tlacitko_prepuseni_R, RISING); // nastaveni na nabeznou hranu  
}
```

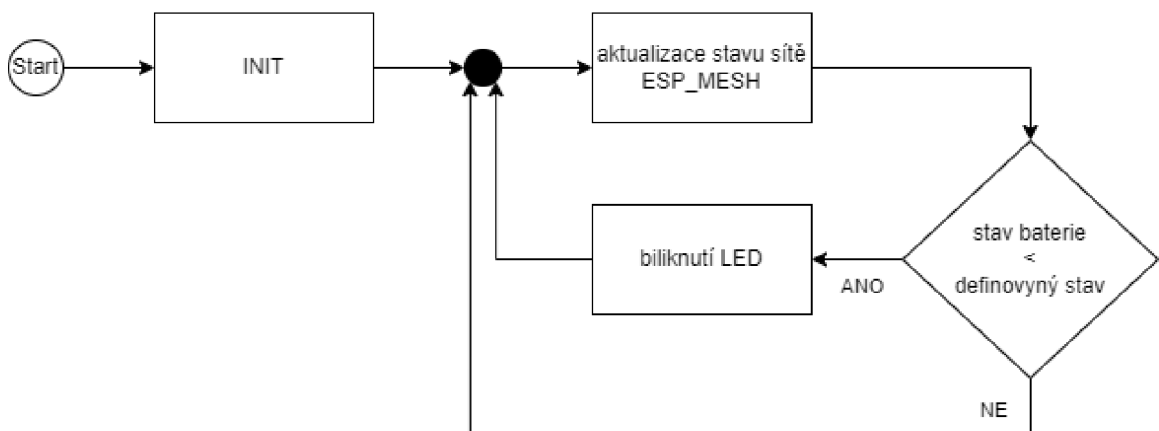
Předposledním stavem je „odesílání zpráv do ESP_MESH sítě“. Pro vnoření do této podmínky je hodnota z funkce millis() dělená konstantou, která je definována při spuštění programu. Takto je nastaven interval pro vnoření do podmínky. Pokud výsledek dělení je větší než stav počítadla, je navigační modul připojen do sítě pomocí funkce ZpravaMesh().

Poslední stav „bliknutí LED“ mění aktuální stav LED na opačný. Interval pro tento stav je řešen podobně jako na předchozím stavu „odesílání zpráv do ESP_MESH sítě“. V případě, že tato dioda svítí a je aktivní stav nouze, dioda svítí a přerušovaně zhasíná. V opačném případě dioda krátce blikne. Celý program je vložen v příloze G.

6.2.6 Program pro maják

Program pro maják je vcelku velmi jednoduchý, viz Obr. 33. Ve stavu „INIT“ jsou nastaveny vstupní a výstupní porty, stejně jako v programu pro navigační modul. Po nich následuje spuštění funkce „startMesh“ ve které je postup příkazů pro správné nastavení a spuštění mesh sítě. Nastaveno je i automatické posílání informací o modulu v definovaném intervalu. Přeposílaná data jsou v datové struktuře JSON, viz Obr. 34, takže při odesílání jsou převedeny na textový řetězec a při přijetí jsou převedeny zpět na proměnné.

Obr. 33 Životní cyklus programu pro maják



Obr. 34 Programové definování datové struktury

```
String getReadings () {  
    JSONVar jsonReadings;  
    jsonReadings["id"] = id_esp;  
    jsonReadings["batt"] = analogRead(pin_batt);  
    readings = JSON.stringify(jsonReadings);  
    return readings;  
}
```

V hlavní programové smyčce je volána funkce update(), která zajišťuje správný běh nastavené sítě. Kontrola stavu baterie je zde řešena obdobně jako u navigačního modulu. Jelikož zde výchozí stav LED je vypnuto, tak stačí diodu pouze krátce rozsvítit, viz Obr. 35. Celý program je vložen v příloze E.

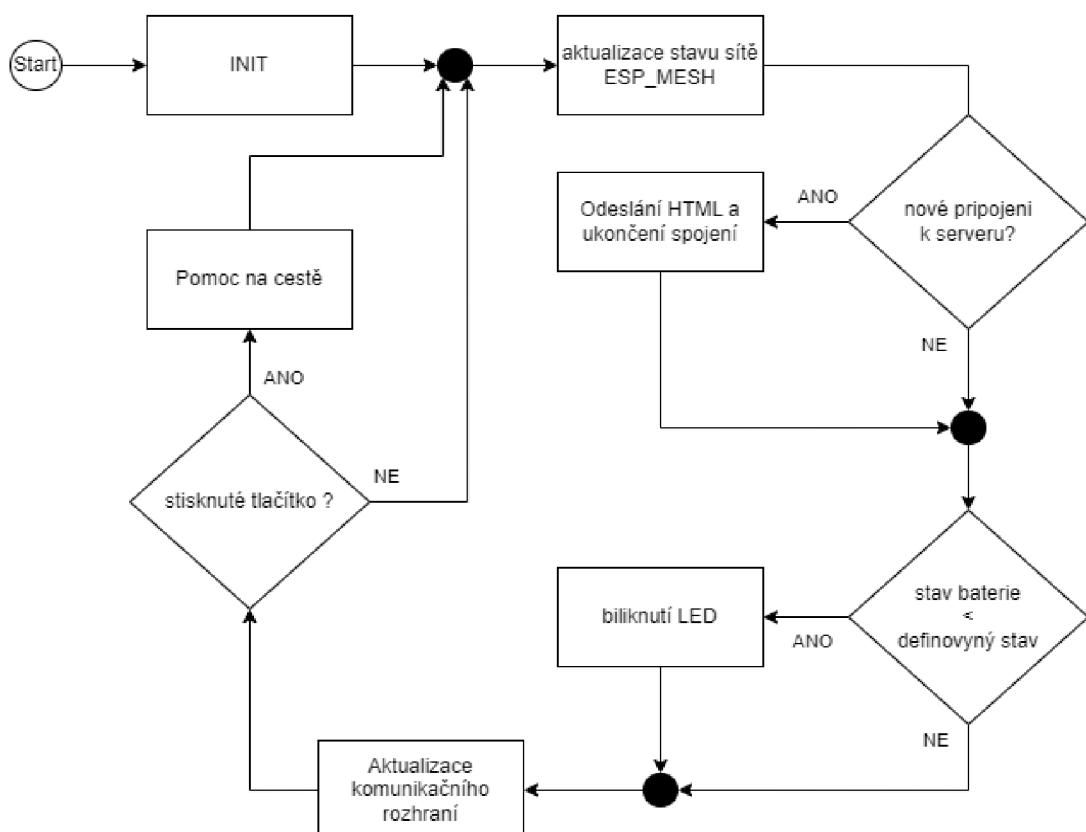
Obr. 35 Program pro maják – světelná indikace

```
if(((millis() - StavBatt) > STAV_BATT) &&
    (adBatt <= nizky_stav)){
    digitalWrite(led_R, HIGH);
    delay(20);
    StavBatt = millis();
    digitalWrite(led_R, LOW);
}
```

6.2.7 Program pro maják s webovým serverem

Nastavení vstupních a výstupních pinů je shodné jako u předchozích zařízení včetně sériové komunikace. Následuje vytvoření nebo zprovoznění komunikace s přednastavenou bezdrátovou sítí ESP-MESH, viz Obr. 36.

Obr. 36 Životní cyklus programu pro maják kombinovaný s web. serverem



Při testování webového serveru běžícího současně se sítí mesh bylo zjištěno, že webový server nebyl vždy dostupný. Ve většině případů byl dostupný, když byl prvním účastníkem (vytvořil síť) v síti mesh. Tento problém byl vyřešen použitím knihovny ESP8266WiFiMulti, která byla už nainstalována v kapitole 6.2.1. Nejprve dojde k pokusu o připojení do sítě pomocí této knihovny a poté spuštěná funkce startMesh(). Webový server je potom dostupný na adrese 10.134.33.1 nebo 10.190.10.100. Nicméně nejlepší výsledky byly dosaženy právě tehdy, když toto zařízení bylo prvkem vytvářecí mesh sítě.

Po stavu „INIT,“ stejně jako u majáku, je spuštěna funkce pro udržení spojení mezi moduly v síti. Následuje detekce připojení uživatele k webovému serveru. Pokud vznikne dotaz k připojení na server, který je spuštěn na portu 80, je splněna podmínka a dotyčným je odeslán jednoduchý HTML soubor. Při vytváření souboru je k vytvoření textu důležitých informací o stavu sítě je použit cyklus, viz Obr. 37. V cyklu jsou postupně procházena dvě různá pole hodnot, ve kterých jsou uloženy získané informace od majáků a navigačního modulu, viz Obr. 38. První pole hodnot obsahuje získanou analogovou hodnotu z AD převodníku použitého k měření stavu baterie. Druhé pole hodnot obsahuje informaci, jestli některým zařízením byl vyvolán stav pomoci.

Obr. 37 Program pro maják s web. serverem – výpis hodnot

```
client.println("<body><h1>ESP8266 Web Server</h1>");
for(byte i =1; i < POC_MAJAKU; ++i){
    client.print("modul "); client.print(i); client.print(" - stav
baterie: ");
    if(batt_majaku[i] <= nizky_stav){
        client.print("Potreba vymenit");
    }else{
        client.print("OK");
    }
    if(pomoc_majaku[i] == true){
        client.println(" - volani o POMOC !!!! <br>");
    }else{
        client.println("<br>");
    }
}
client.println("</body></html>");
```

Po úspěšném odeslání webové stránky bylo navázané spojení stále aktivní k webovému serveru. To způsobovalo zastavení běhu programu. Proto na konci této podmínky pro

odeslání webové stránky je spojení automaticky ukončeno. Následuje kontrola stavu baterie, která je řešená stejným způsobem jako u majáku.

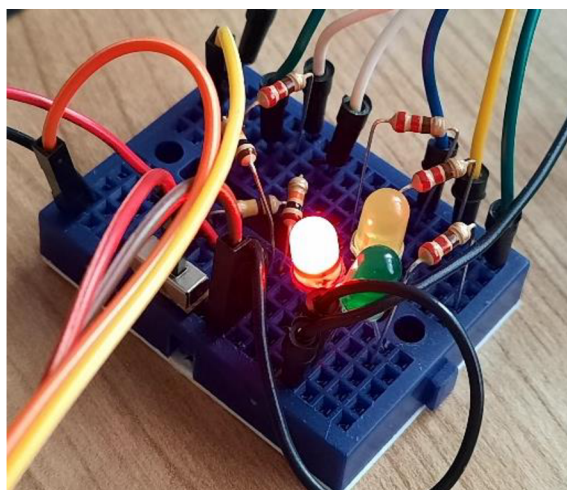
Ve stavu „aktualizace komunikačního rozhraní“ jsou použity dva FOR cykly, které procházejí již zmíněná pole hodnot. V poli hodnot s daty o stavech baterií všech zařízení v připojené síti jsou vyhledávány hodnoty, které jsou menší než definované konstantou „nizky_stav“. Pokud je nalezena alespoň jedna hodnota nižší, než je mezní hodnota, je rozsvícena žlutá dioda. Druhý cyklus prochází pole s hodnotami o volání pomoci. Pokud nalezne alespoň jednu hodnotu TRUE, tak rozsvítí červenou diodu a spustí akustickou signalizaci. Pokud nebyla rozsvícená ani jedna dioda, rozsvítí se zelená dioda indikující bezproblémový stav.

Při zachycení volání o pomoc navigačního modulu je možné stisknout tlačítko. Po jeho stisku bude odesílána zpráva a vyslání pomoci. Navigační modul poté bude přehrávat hlášku „pomoc je na cestě.“ Celý program je vložen v příloze F.

Obr. 38 Pohled na obsah webového serveru



Obr. 39 Pohled na zapojení na univerzálním poli



6.2.8 Výpočet konstanty pro indikaci nízkého stavu baterie

Dle dokumentace napájecí modul AMS1117-3.3 V je schopen pracovat od napětí 4,75 V do 12 V. Pro testovací účely hodnota nízkého napětí byla zvolena 4,8 V. Ve finální verzi by bylo vhodné toto napětí zvednout alespoň na 5 V.

Výpočet napětí z odporového děliče na vstupu A0:

$$U_{A0min} = \frac{UCC_{min}}{R1} \cdot R2 \quad (2)$$

$$U_{A0min} = \frac{4,8}{110000} \cdot 10000 \quad (3)$$

$$U_{A0min} \cong 0,436 V \quad (4)$$

Kde U_{A0min} je napětí nastavené odporovým děličem, UCC_{min} je hodnota napětí nízkého stavu baterie, hodnoty $R1$ a $R2$ jsou pevně definované hodnoty rezistorů.

Testováním bylo zjištěno, že použitý AD převodník v mikroprocesoru snímá hodnoty od napěťové úrovně GND až do napájecího napětí VDD. V dokumentaci bylo uveden rozsah AD převodníků od 0 V do 1 V.

Převedení na 10bit AD převodník:

$$A = \frac{U_{A0min}}{\frac{VDD}{2^{10}}} \quad (5)$$

$$A = \frac{0,436}{\frac{3,3}{1024}} \quad (6)$$

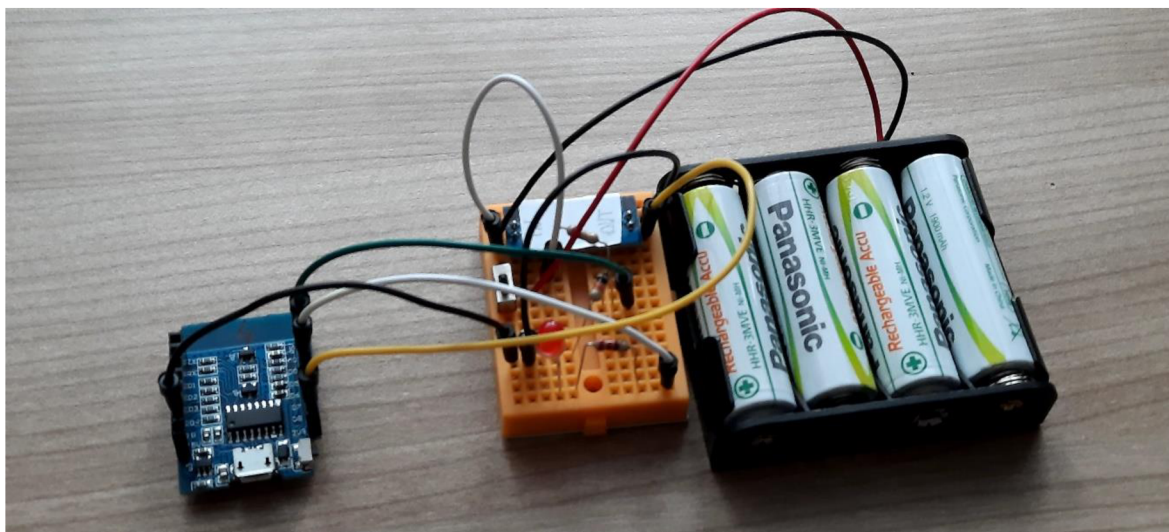
$$A \cong 135 \quad (7)$$

Kde A je digitální hodnota AD převodníku vyjadřující napětí U_{A0min} , VDD je hodnota napájecího napětí mikroprocesoru a 10 bit AD převodník může mít 2^{10} možných stavů.

6.2.9 Hardwarové zapojení pro testování

Navržený systém byl testován na třech Wemos D1 modulech a jednom napájeném modulu ESP12F na univerzální pájecí desce. Moduly ESP maják, viz Obr. 40, a ESP maják s webovým serverem, viz Obr. 41, jsou napájeny čtveřicí nabíjecích článků Panasonic HHR-3MVE Ni-MH nebo články GP ReCyko Pro 2000. Akumulátory jsou umístěny v pouzdru pro 4 články typu AA. Z pouzdra jsou vyvedeny dva vodiče. Vodič s červenou bužírkou vede kladné napětí ze sériově zapojených článků, a vodič s černou bužírkou vede nulové napětí. Kladné napětí je přivedeno na mechanický přepínač, kterým lze snadno obvod přerušit. Za přepínačem je přívod ke stabilizátoru napětí a odporovému děliči pro měření výstupního napětí akumulátorů. Těsně za přepínačem byla plánována dioda, jako ochrana proti přepólování. Bohužel úbytek na diodě by zapříčinil nízký stav napětí pro chod stabilizátoru. Schéma zapojení akumulátorů a modulu se stabilizátorem je v příloze D. Kapacity kondenzátorů nejsou nikde zaznamenané, proto jsou v tabulce uvedeny doporučené velikosti kapacit kondenzátoru. Napájení pro zbylé dva moduly jsou řešeny připojením k powerbance nebo počítači.

Obr. 40 ESP maják

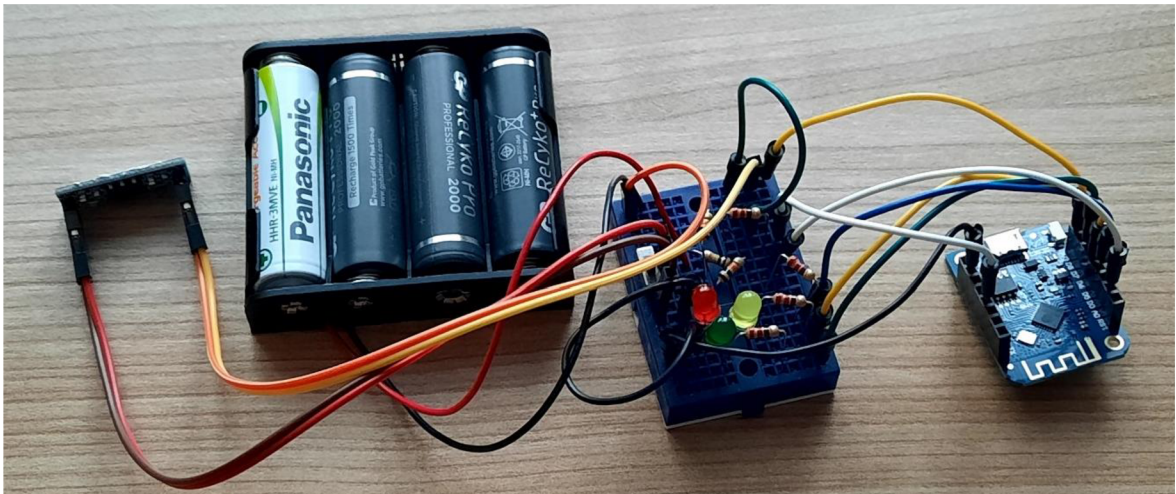


Modul ESP maják má pouze jeden výstup, na kterém je připojena nízko příkonová červená LED dioda přes rezistor R1 pro omezení protékajícího proudu. Tuto diodu mají všechny další moduly, stejně jako odporový dělič k měření stavu napětí akumulátorů. Více pinů pro funkci tohoto modulu není potřeba. Schéma pro tento modul je v příloze A.

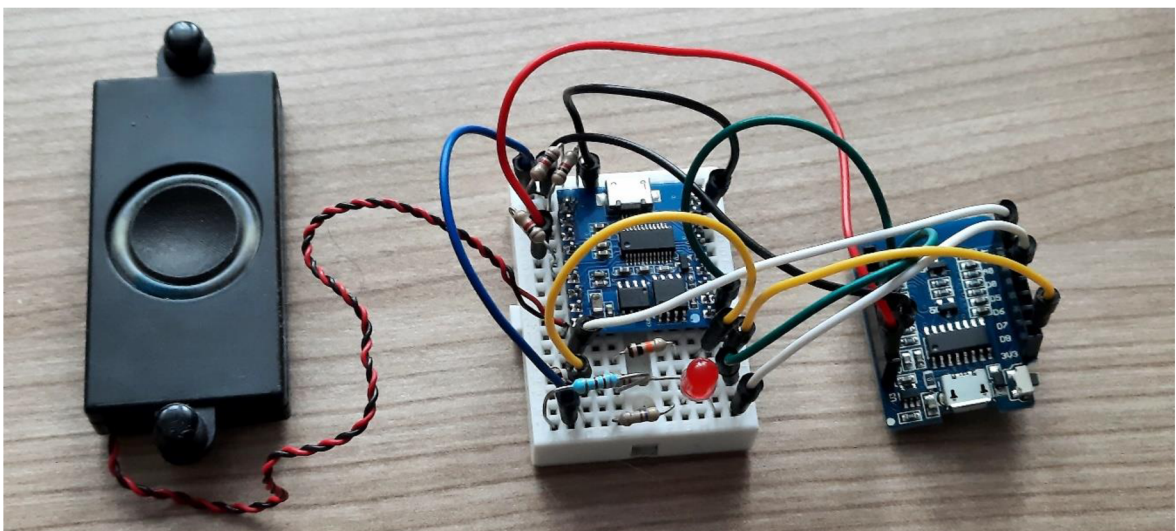
Modul ESP maják s webovým serverem je rozšířenou variantou ESP maják. Schéma pro tento modul je v příloze B. Je vybaven dalšími dvěma nízko příkonovými diodami (žlutá a

zelená) s příslušnými rezistory pro omezení proudu diodou. Diody jsou rozsvěcovány podle různých stavů, které jsou popsány v programu pro tento modul (Program pro maják s webovým serverem). Tlačítko S1 je připojeno k mikroprocesoru rezistorem R9 a napájecímu napětí. Rezistor je zde určen pro omezení proudu při případné špatné konfiguraci pinů, která může nastat při spouštění programu. Aby bylo možné vyhodnotit stav tlačítka je k tlačítku je připojen pull-down rezistor R4. Paralelně k tlačítku je zapojen kondenzátor k potlačení případných zákmitů kontaktů v tlačítku. Reprodaktor (případně může být nahrazen piezoelektrickým měničem) se zapojenou paralelní diodou je spínán tranzistorem Q1, který je ovládán mikroprocesorem. Rezistorem R7 je nastaveno napětí na gate tranzistoru a rezistorem R8 je nastaven maximální proud nabíjení kapacity elektrody tranzistoru. Pro testovací účely reproduktor nebyl zapojen.

Obr. 41 ESP Maják s webovým serverem

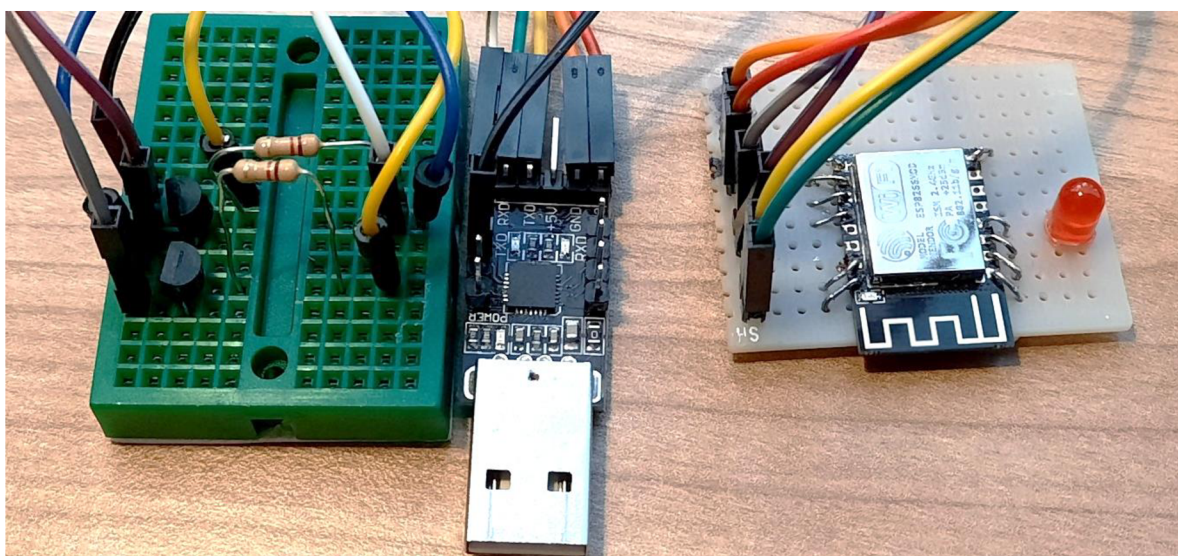


Obr. 42 ESP-Navigační modul



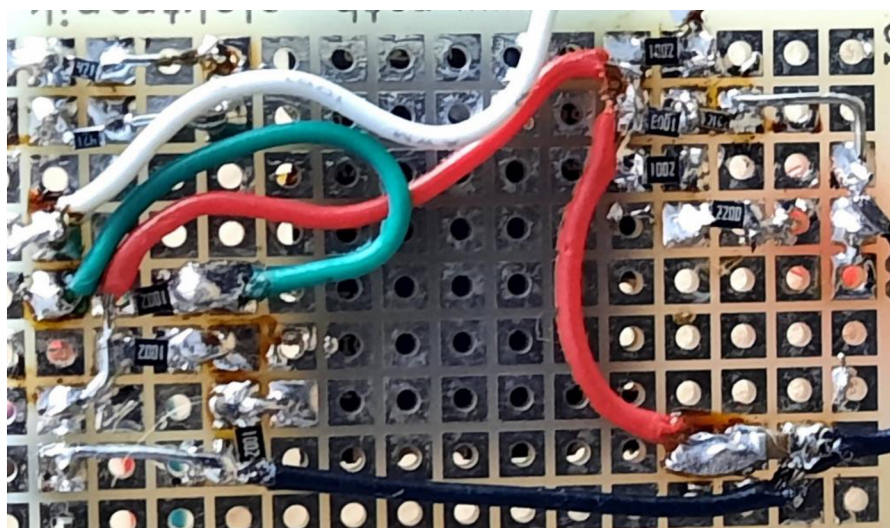
Navigační modul je vybaven tlačítkem k vyvolání stavu nouze nebo spuštění navigace, jak bylo popisováno v programu pro tento modul. Tlačítko je zapojeno stejným způsobem, jako u ESP majáku s webovým serverem. Na mikroprocesoru je vývod D4 nakonfigurován jako UART1. Po této lince je zprostředkována komunikace se zvukovým modulem. Konfigurace pinů zůstala stejná jako z testování popsané v kapitole 6.2.2. Celé schéma zapojení je v příloze C.

Obr. 43 ESP12F s programátorem CP2102 a obvodem potřebným k nahrání programu



Na obrázku Obr. 43 je zobrazen ESP maják napájený na univerzální desce zapojen s programátorem CP2102 a nepájivým polem. Na nepájivém poli je zapojena dvojice tranzistorů podle schématu na Obr. 22. Programátor je zároveň i zdrojem elektrické energie pro provoz tohoto modulu.

Obr. 44 Pohled na zapojení ESP majáku na univerzální desce



6.3 Vyzkoušení systému a ladění systému

Celý systém byl testován v domácích podmínkách, kdy majáky byly rozmístěny různě po místnostech viz Obr. 47. Maják s webovým severem byl umístěn v kuchyni na stole. Další maják byl umístěn v pokoji na stole a poslední maják byl umístěn na botníku v předsíni. Nejdříve bylo testováno připojení na webový server pomocí tabletu. Připojení z pokoje k webovému serveru dopadlo neúspěšně. Webový server byl dostupný po přemístění tabletu blíže k majáku s webovým serverem, tedy do kuchyně.

Obr. 45 Program pro navigační modul – úprava navigace()

```
if((nasl == 0) && (pred == 0)){...
}else{
    if(nasl > před /*(nasl > 0) && (pred <= 0)*/){...
    }
    }else{...
    }
}
}
```

V rané fázi testování byl navigační modul napájen z powerbanky a byl nesen figurantem od prvního majáku až k poslednímu. Následovala korekce programu ve vyhodnocování podmínky pro navigaci (změna polarity znamének). Dalším testování však výsledky nebyly příliš uspokojivé. Byla tedy provedena další korekce vyhodnocování pro navigaci. Podmínka pro vyhodnocení předchozí a následující zpracované hodnoty RSSI byla upravena na porovnávání mezi těmito hodnotami, viz Obr. 45. Po těchto úpravách byla spuštěná navigace úspěšnější v navigování figuranta, viz Obr. 46. Touto úpravou byl vyřešen problém, kdy RSSI jednoho majáku byla měněna více oproti druhé, když byl figurant v pohybu.

Neustále přetrvával problém po spuštění navigace. Proto maják zapojený v pokoji do notebooku byl přepojen do power banky a navigační modul byl připojen k notebooku. V notebooku byla spuštěna komunikace s navigačním modulem pro ladění programu. Následujícím průchodem navigačního modulu od prvního po poslední maják bylo zjištěno několik podstatných skutečností. Antény na DPS ESP modulů nejsou všesměrové. Při umístění v prostoru je tedy potřeba instalovat moduly anténou od stěny.

Po spuštění navigace program v několika málo iteracích zprůměruje hodnotu rozdílu síly signálu a následně je spuštěna navigace. Testováním bylo zjištěno, že tato metoda pomáhá při rozhodování, ale není dostačující. Hodnoty v některých případech se velmi výrazně

Obr. 46 Výpis do konzole po úpravě programu

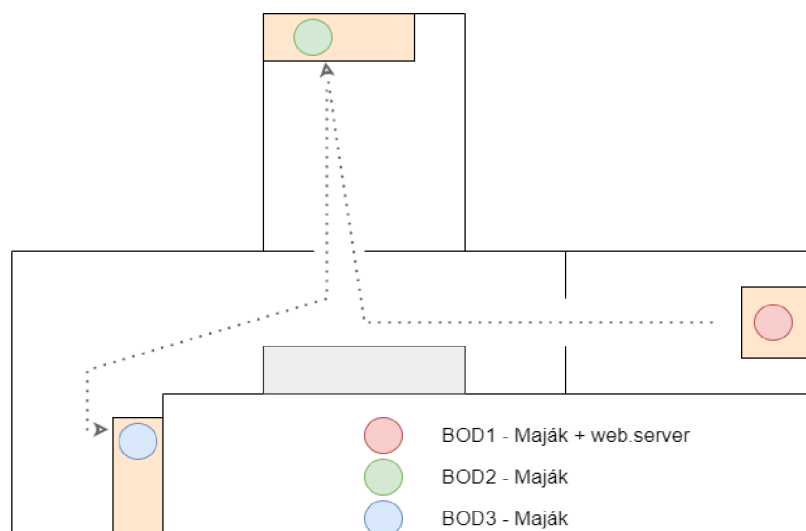
```

DEBUG: poloha2: 2
DEBUG: prumer1 -7
DEBUG: prumer2 1
DEBUG: navigace: spravny smer
DEBUG: navigace: k bod2
site nalezeny
DEBUG: poloha1: 9
DEBUG: poloha2: 1
DEBUG: prumer1 -3
DEBUG: prumer2 3
DEBUG: navigace: spravny smer
DEBUG: navigace: k bod2
site nalezeny
DEBUG: poloha1: -6
DEBUG: poloha2: 8
DEBUG: prumer1 -4
DEBUG: prumer2 4
DEBUG: navigace: spravny smer
DEBUG: navigace: k bod2
site nalezeny
DEBUG: poloha1: 1
DEBUG: poloha2: 18
DEBUG: prumer1 0
DEBUG: prumer2 7
DEBUG: navigace: dosli jste do cile
DEBUG: navigace: k bod2
    
```

měnily, když modul setrval na místě. Tyto výkyvy se oproti testování při vytváření programu neprojevovaly. Nejspíše se jednalo o vnější rušení nebo odrazy v malé místnosti. V některých případech však navigační modul dovedl figuranta až k cílovému majáku. Část výpisu v programu Arduino IDE z konzole sériové komunikace s modulem po úpravě programu je na obrázku Obr. 46.

Když navigační modul odesílá data do sítě není schopen skenovat okolní síť. Tento proces má nežádoucí účinky na spuštěnou navigaci. Při krátké vzdálenosti totiž nebylo vždy jasné, jestli figurant jde správně. Při testování nedostupnosti majáku navigační modul okamžitě zareagoval a nahlásil chybu sítě.

Obr. 47 Znárodnění testovací trasy



7 Výsledky a diskuse

Navržený navigační systém je určen především pro vnitřní prostory. Zřejmě by mohl být nasazen i ve vnějších prostorech. Hardware by musel být instalován do boxu s dostatečným krytím s odolností proti povětrnostním podmínkám. Musela by být vyvedena anténa, pro lepší vysílání signálu.

Majáky musí být umístěny na strategických místech, kde je potřeba sdělit uživateli, jakým směrem se má vydat. Těmito místy jsou křižovatky, schodiště, vstup do výtahu či označení konkrétních dveří, např. označení toalet, kanceláří, ordinačních místností. Využití systému je především ve veřejných budovách. Systém by nemusel sloužit jen nevidomým, ale i lidem pro snazší orientaci ve velkých komplexech, jako jsou například nemocnice. Překážkou systému by nemuselo být ani využití ve více patrech, jelikož síla signálu přes strop by byla utlumena dostatečně tak, aby nenarušovala správný běh navigačního modulu. Při vysoké hustotě majáků by mohly být rozděleny na orientační a cílové. Pokud by totiž trasa mezi majáky byla příliš krátká, program by obsahoval velké množství majáků, což by mohlo být méně efektivní z pohledu paměti.

Současná verze systému popisovaná v této práci má pevně definovanou trasu v programu podle které je uživatel navigován. Pro širší využití by bylo vhodné navrhnout algoritmus a komunikační rozhraní pro vyhledání cílového majáku a automatické vygenerování trasy. V případě malé sítě, by mohl navigační modul vyhledat trasu sám. Ve velké síti, by to mohl být problém z hlediska paměti a výpočetního výkonu. Tento problém by mohlo řešit nahrazení dosavadního navigačního modulu chytrým mobilním telefonem, nebo mikroprocesorem s vyšším výpočetním výkonem.

Webový server spuštěný na jednom majáku je především pro dispečera (vrátného), který dohlíží na stav systému, jako je stav baterií a volání o pomoc v případě nouze uživatele. Webový server by mohl být spuštěn na všech majácích, aby dispečer se mohl připojit kdekoliv v dosahu sítě. Výhodou by byla pohodlnější výměna baterií, jelikož by si dispečer nemusel pamatovat konkrétní moduly. Navíc by na serveru bylo možné zobrazit aktuální polohu uživatele, případně poslední polohu, kdyby opustil dosah sítě, pro snazší lokaci, jelikož uživatel nemusí vždy setrvávat na jednom místě. Výhodou webového serveru je snadná přístupnost z jakéhokoliv zařízení jako je počítač, chytrý telefon, tablet atd.

Skenování sítě je v aktuálně navrženém programu málo efektivní. Jelikož program čeká, než je skenování majáků dokončeno a potom teprve pokračuje dále. Program by bylo vhodné

upravit tak, aby skenování bylo spuštěno dříve a pak přistoupil rovnou k výsledkům vyhledávání. Tento proces je složitější na implementaci, a proto nebyl použit do současné verze navigačního systému.

Navigační modul se aktuálně orientuje pouze pomocí získané síly signálu orientačních majáků. Pro navigování jsou vždy použity dva specifické majáky se specifickou MAC adresou. Zakomponování akcelometru k hodnotám RSSI by mohlo vylepšit navigování. Navržené napájecí obvody jsou vhodné pro baterie nebo zásuvkové napájecí zdroje v definovaném rozmezí viz příloha D. Nahrazení stabilizátoru step-up nebo step-down měničem by přineslo vyšší efektivitu přeměny vstupního napájecího napětí na požadované napětí.

Navigační modul při připojování a vysílání dat do sítě mesh není schopen navigovat uživatele, což ve výsledku může uživatele náhle zmást. Při dalším vývoji tohoto systému by bylo vhodné i při této sekvenci dále vyhledávat maják a pokračovat v navigaci. Bohužel nebyl prostor vyzkoušet tuto myšlenku z důvodu nedostatku času, a těžko říct, jestli je to pro ESP8266 reálné.

8 Závěr

V rešeršní části jsou popsány různé navigační systémy určené pro nevidomé osoby nebo osoby se zhoršeným orientačním smyslem. Velmi zajímavé jsou orientační majáky pro VPN vysílačky, které jsou umístěny na strategických místech ve veřejných prostorech, jako jsou např. česká pošta nebo zastávky městské hromadné dopravy. Dalším inspirativním systémem je aplikace pro chytré telefony, které využívají technologii Bluetooth pro komunikaci s majáky. Majáky jsou různě rozmístěny po vnitřních prostorech budovy.

Navržený navigační systém je složen z orientačních majáků rozmístěných ve vnitřních prostorech budovy a navigačního modulu. Moduly využívají technologii Wifi ke vzájemné komunikaci mezi sebou. Navigační modul má navržené komunikační rozhraní určené pro nevidomé. Součástí je zvukový modul s nahranými zvukovými stopami, které jsou použity příkazem z mikroprocesoru ESP8266. Komunikace mezi mikroprocesorem a zvukovým modulem je realizována pomocí UART.

Bylo vytvořeno jednoduché webové prostředí pro předávání důležitých informací dispečerovi. Stránku není třeba neustále kontrolovat, stav systému je indikován třemi LED. Proto maják určený pro umístění na vrátnici má rozšířené komunikační rozhraní. Zelená dioda je rozsvícena, když není vyžadována pozornost dispečera a indikuje tak normální stav systému. Žlutá dioda indikuje nízký stav baterií. Současně se žlutou diodou může svítit i červená dioda, která indikuje uživatele v nesnazi, a je vyžádána pomoc. Každý orientační maják je vybaven červenou LED diodou, která při nízkém stavu baterie krátce bliká.

Testováním systému byly nalezeny chyby, které byly vyřešeny úpravou softwaru pro navigační modul. Po úpravách byl figurant doveden k cílovému majáku. V předchozí kapitole byli popsány možná vylepšení systému, které by mohla po jejich vyřešení vést k nasazení do provozního režimu.

9 Bibliografie

1. **SCHINDLER, Radek; PEŠÁK, Milan.** Kdo je zrakově postižený? *SONS.cz*. [Online] 6 2015. [Citace: 31. 1 2023.] <http://archiv.sons.cz/kdojezp.php>.
2. Tyflopomůcky - Prodej pomůcek pro zrakově postižené. *SONS*. [Online] [Citace: 17. 01 2023.] <https://eshop.tyflopomucky.cz/kategorie/hodinky-a-budiky/>.
3. **KARIMI, Hassan A.** *Indoor Wayfinding and Navigation*. První vydání. London: CRC Press, Taylor & Francis Group, 2015. ISBN 1482230852; 9781482230857.
4. **GARAJ, U.** The Brunel Navigation System for the Blind: Determination of the Appropriate Position to Mount the External GPS Antenna on the User's Body. *ResearchGate*. [Online] 1 2000. [Citace: 4. 2 2023.] https://www.researchgate.net/publication/259971603_The_Brunel_Navigation_System_for_the_Blind_Determination_of_the_Appropriate_Position_to_Mount_the_External_GPS_Antenna_on_the_User's_Body.
5. **PETRIE, Helen a kol.** MOBIC: Designing a Travel Aid for Blind and Elderly People. *ResearchGate*. [Online] 1 1996. [Citace: 3. 2 2023.] https://www.researchgate.net/publication/231835192_MOBIC_Designing_a_Travel_Aid_for_Blind_and_Elderly_People.
6. **MOORE, S.E.; HELAL, A.; RAMACHANDRAN, B.** Drishti: An Integrated Navigation System for. *cise.ufl.edu*. [Online] [Citace: 4. 2 2023.] <https://www.cise.ufl.edu/~helal/projects/publications/wearableConf.pdf>.
7. **VIRTANEN, Ari; KOSKINEN, Sami.** NOPPA NAVIGATION AND GUIDANCE SYSTEM FOR THE BLIND. *ResearchGate*. [Online] [Citace: 4. 2 2023.] https://www.researchgate.net/publication/266097542_NOPPA_NAVIGATION_AND_GUIDANCE_SYSTEM_FOR_THE_BLIND.
8. **FOJTÍKOVÁ, Šárka.** *Vodící pes jako faktor zvýšení kvality života*. [Internetový zdroj] Brno : autor neznámý, 2009.
9. Plemena psů. *Škola pro výcvik vodících psů*. [Online] 2018. [Citace: 26. 2 2023.] <https://guidedogs.cz/vodici-psi/plemena-psi/>.
10. Bílé hole - Kategorizace a názvosloví bílých holí. *SONS.cz*. [Online] [Citace: 20. 01 2023.] <https://archiv.sons.cz/docs/bilehole/01.php>.

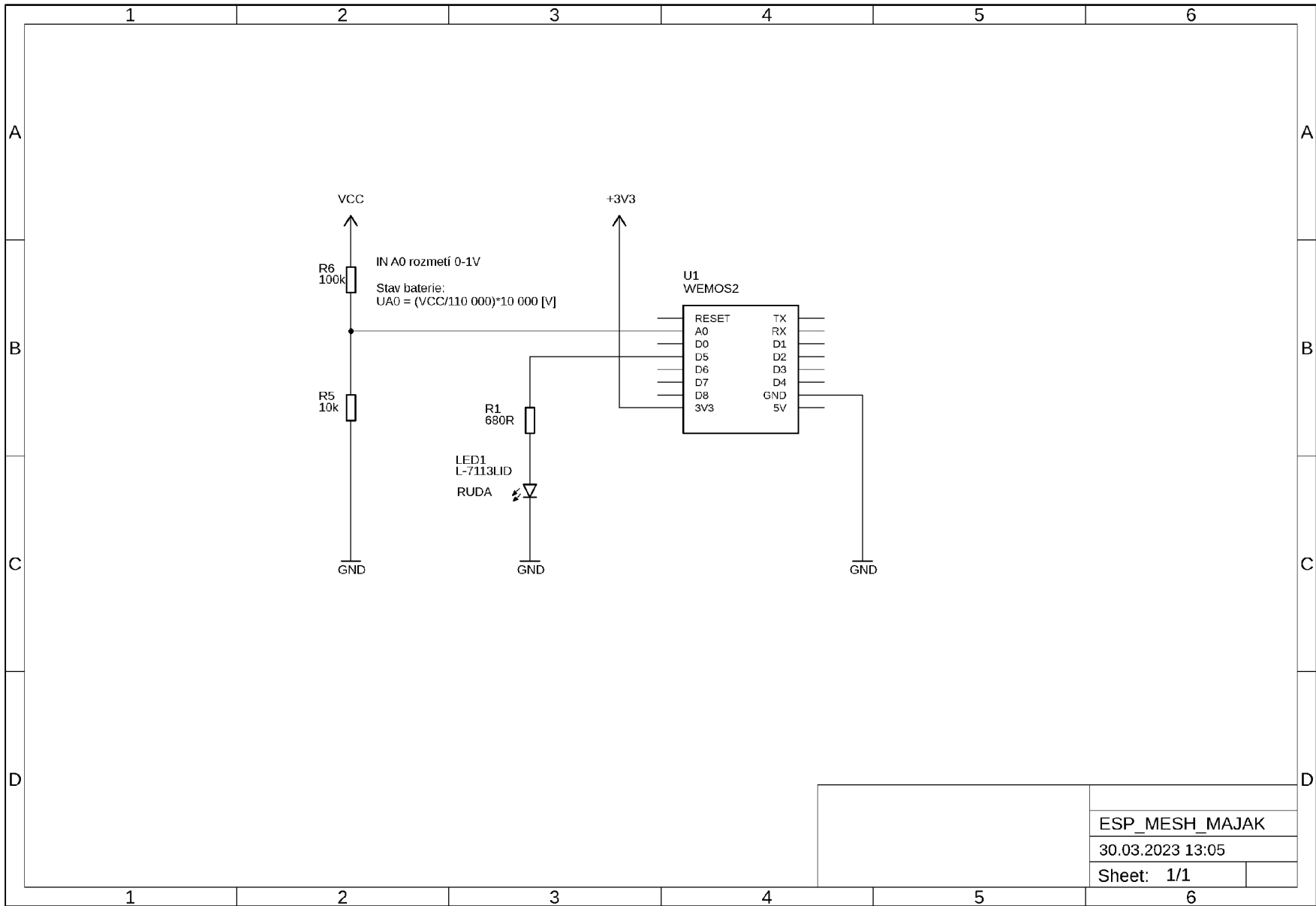
11. Vyhláška č. 294/2015 Sb. *Zákon pro lidi.cz*. [Online] AION CS, s.r.o. 2010-2023. [Citace: 20. 01 2023.] <https://www.zakonyprolidi.cz/cs/2015-294/zneni-20160101#p28-1-1>.
12. **ROBENEK, Jan.** Bílá hůl dostane tlačítka, rodina pokyny. Situaci zhodnotí očima kamery. *vyvoj.hw.cz*. [Online] HW server s.r.o., 18. 12 2015. [Citace: 22. 01 2023.] <https://vyvoj.hw.cz/zdravi-a-medicina/bila-hul-dostane-tlacitka-rodina-pokyny-situaci-zhodnoti-ocima-kamery.html>.
13. Tyflosonar RAY – ultrazvukový detektor překážek. *TYFLOCOMP*. [Online] [Citace: 23. 01 2023.] <https://tyflocomp.sk/produkty-php/pomocky-na-orientaciu/tyflosonar>.
14. **BENDO VÁ, Petra; JEŘÁBKOVÁ, Kateřina; RŮŽIČKOVÁ, Veronika.** Kompenzační pomůcky pro osoby se specifickými potřebami. *DOCPLOYER*. [Online] 2006. [Citace: 26. 2 2023.] <https://docplayer.cz/8496114-Kompenzacni-pomucky-pro-osoby-se-specifickymi-potrebami.html>. ISBN 80-244-1436-8.
15. **VOLF, J.; ŠKEŘÍK, F.; NOVÁK, V.** XXI IMEKO World Congress. *imeko.org*. [Online] 30. 8 2015. [Citace: 12. 2 2023.] <https://www.imeko.org/publications/wc-2015/IMEKO-WC-2015-TC18-354.pdf>.
16. **FOJTŮ, Martina.** Univerzita spolupracuje se Seznamem. Společně rozšiřují mapy pro nevidomé. *Zprávy z MUNI*. [Online] Masarykova univerzita, 7. 5 2016. [Citace: 31. 1 2023.] https://www.em.muni.cz/udalosti/11654-univerzita-spolupracuje-se-seznamem-spolecne-rozsiruji-mapy-pro-nevidome?fbclid=IwAR2H_DkaOxlvoHtFloZxJloqdVJPlrb_PGUfxseJkTy25rqbUCxGgo8vsU.
17. VPN návod a použití. *SONS*. [Online] [Citace: 17. 01 2023.] <https://www.sons.cz/VPN-navod-a-pouziti-P4002837.html#prettyPhoto>.
18. **KONEČNÝ, Josef.** AKUSTICKÉ ÚPRAVY PRO NEVIDOMÉ. *centrumpronevidome*. [Online] 12. 11 2018. [Citace: 23. 01 2023.] <https://www.centrumpronevidome.cz/subdom/bariery/bariery/akusticke.htm>.
19. ORIENTAČNÍ MAJÁČKY PRO NEVIDOMÉ. *Elektronické orientační pomůcky*. [Online] [Citace: 23. 01 2023.] <http://www.ok.cz/elvos/Majacky.html>.

20. A Technical Overview of the. *Righthear*. [Online] [Citace: 4. 2 2023.] https://righthear.com/wp-content/uploads/2018/11/Technial-Overview-Whitepaper_RightHear.pdf.
21. Co je WiFi 802.11ac? *Alza.cz*. [Online] 4. 4 2022. [Citace: 17. 2 2023.] <https://www.alza.cz/slovník/wifi-80211ac>.
22. **BETÜL, Aktaş; MAHMUT, Çavur**. A Study of Interior Positioning Technologie. *ResearchGate*. [Online] 12 2018. [Citace: 27. 2 2023.] https://www.researchgate.net/publication/329935669_A_Study_of_Interior_Positioning_Technologies.
23. What is a System on Chip (SoC)? *anysilicon*. [Online] [Citace: 12. 2 2023.] <https://anysilicon.com/what-is-a-system-on-chip-soc/>.
24. Datasheet ESP8266EX. *ESPRESSIF*. [Online] 2 2023. [Citace: 27. 2 2023.] https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
25. NodeMCU ESP8266 Detailed Review. *Make-if.ca*. [Online] [Citace: 22. 2 2023.] <https://www.make-it.ca/nodemcu-details-specifications/>.
26. ESP8266EX. *Espressif*. [Online] [Citace: 24. 1 2023.] https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
27. ESP8266 Pinout Reference: Which GPIO pins should you use? *Random nerd tutorials*. [Online] [Citace: 24. 1 2023.] <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>.
28. **GROKHOTKOV Ivan**. Reference. *ESP8266 Arduino Core*. [Online] [Citace: 20. 3 2023.] <https://arduino-esp8266.readthedocs.io/en/latest/reference.html#analog-output>.
29. DY-SV17F module functions introduction. *The backshed*. [Online] 17. 11 2022. [Citace: 2. 2 2023.] https://www.thebackshed.com/forum/uploads/Grogster/2022-11-17_193909_DY-SV17F%20module%20functions%20introduction.pdf.
30. Arduino Sketchbook. *Arduino.cc*. [Online] [Citace: 25. 2 2023.] <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2>.
31. **KŘÁPEK, Milan**. *Navigace nevidomých*. [Online] 2007. [Citace: 17. 1 2023.] https://is.muni.cz/th/bn6lv/diplomka_2007_05_11.pdf.

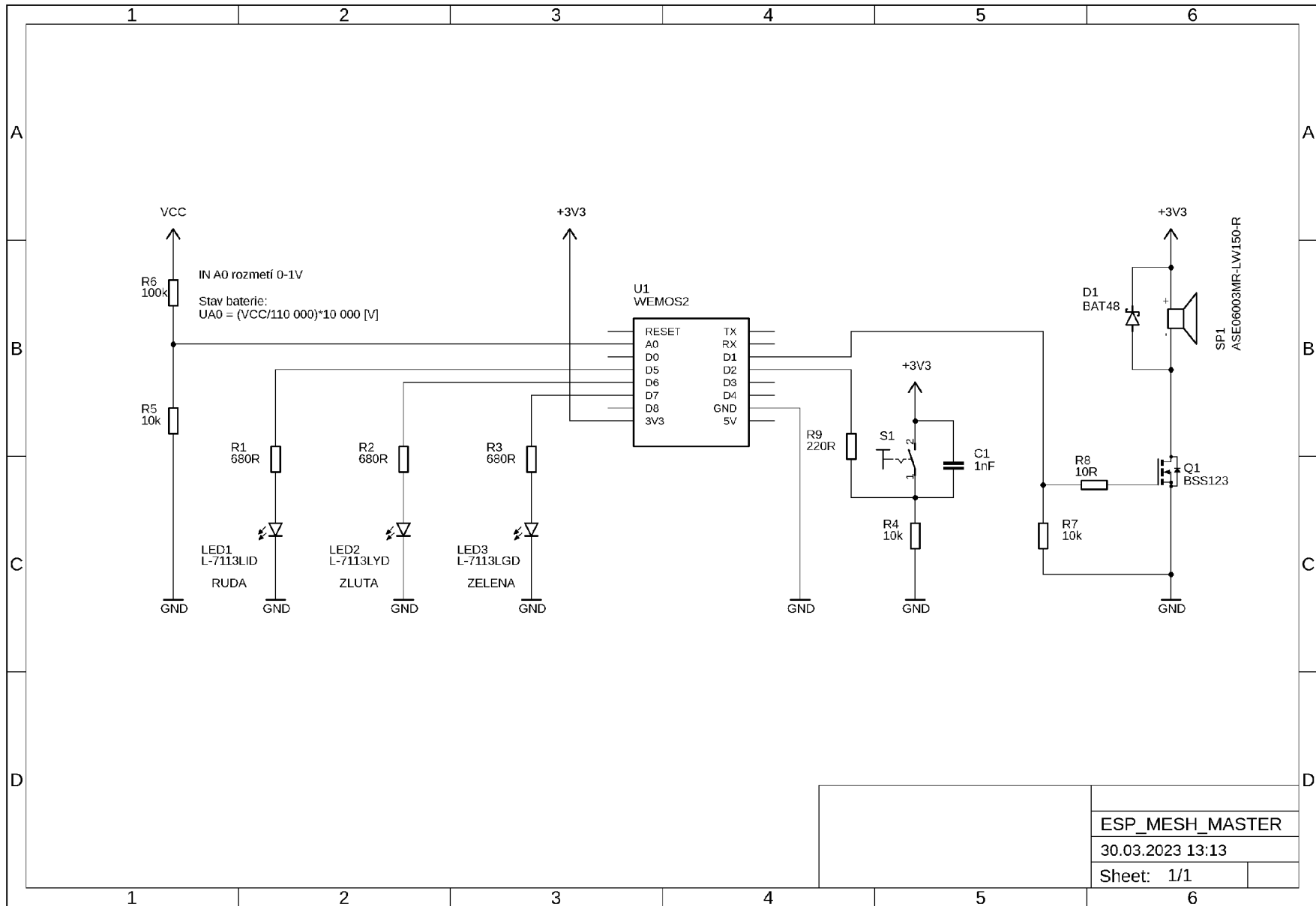
32. Domovská stránka. *RightHear*. [Online] [Citace: 26. 2 2023.] <https://www.right-hear.com/>.

10 Přílohy

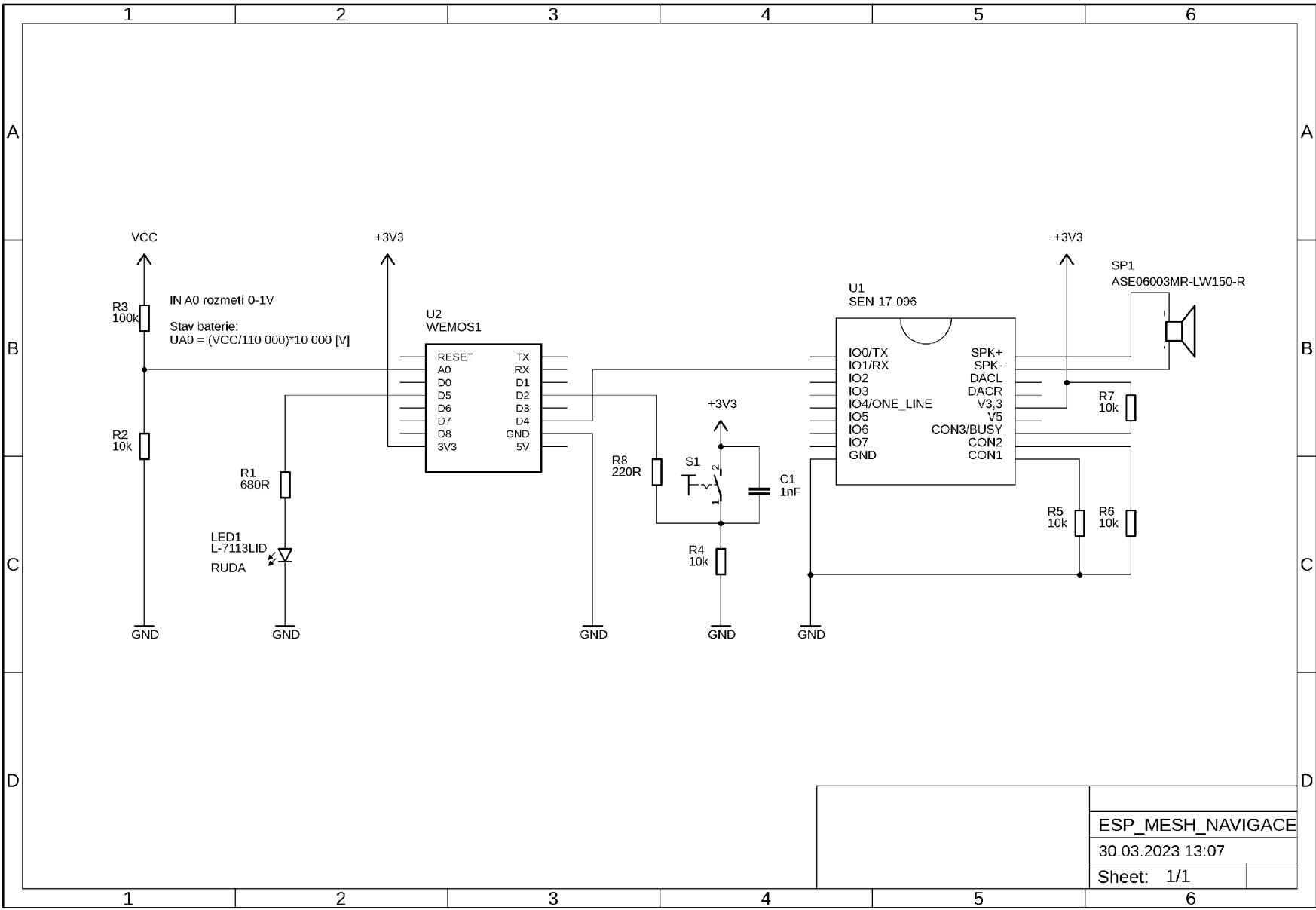
Příloha A – Schéma pro ESP maják	59
Příloha B – Schéma pro ESP maják s webovým serverem	60
Příloha C – Schéma pro ESP navigační modul	61
Příloha D – Schéma pro úpravu napětí	62
Příloha E – Program pro ESP maják.....	63
Příloha F – Program pro ESP maják s webovým serverem	66
Příloha G – Program pro ESP navigační modul.....	71

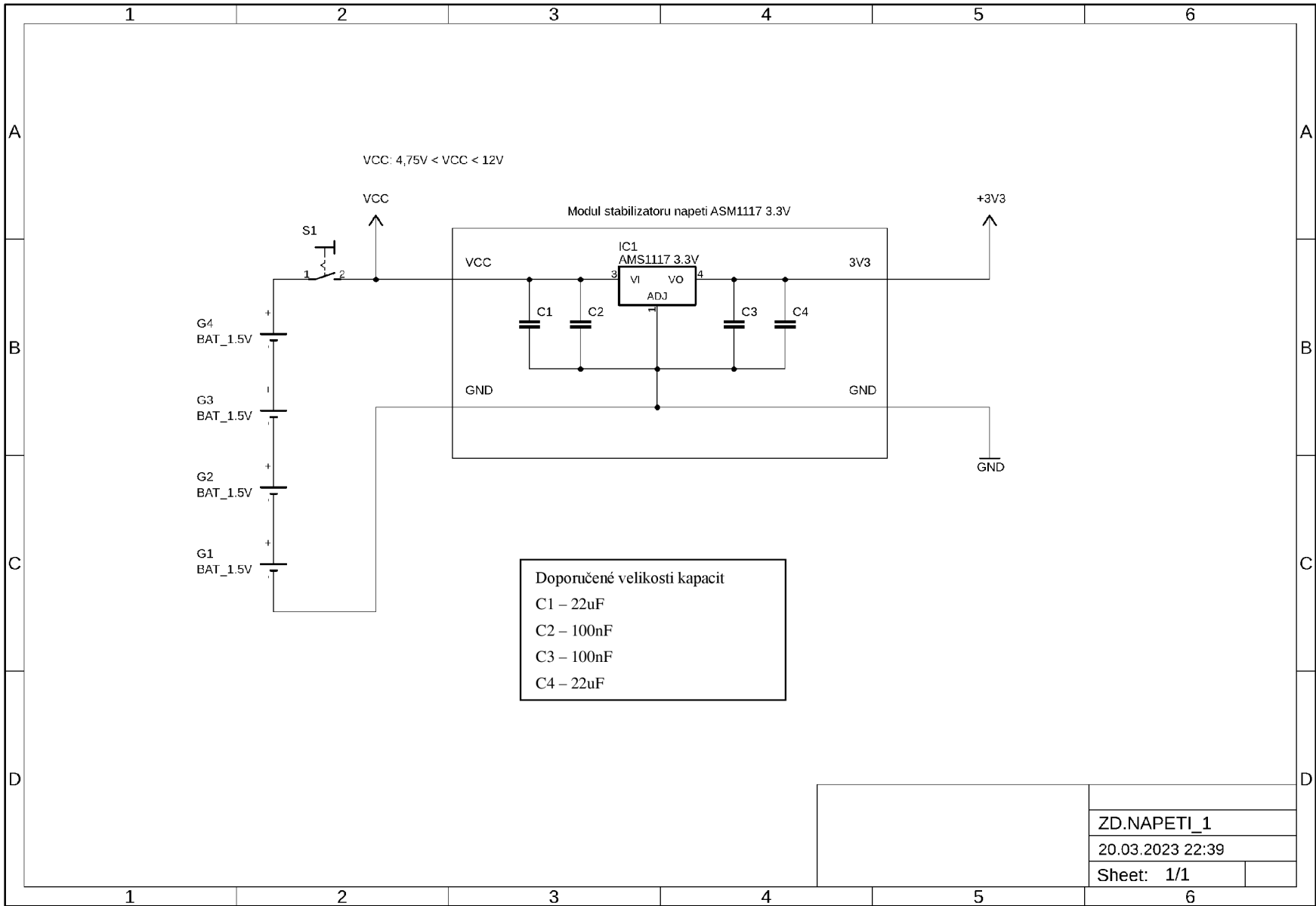


ESP_MESH_MAJAK	
30.03.2023 13:05	
Sheet: 1/1	



Příloha B – Schéma pro ESP maják s webovým serverem





Příloha E – Program pro ESP maják

```
#include "painlessMesh.h"
#include <Arduino_JSON.h>
#include <ESP8266WiFi.h>
#include <Ticker.h>

#define MESH_PREFIX "NavPrNevidim" //nazev site
#define MESH_PASSWORD "8695123!!AA" //heslo site
#define MESH_PORT 5555 //komunikacni TCP port

#define CHANNEL_SLAVE 1
#define STAV_BATT 10000 // ms

// definoovani pinu
const byte pin_batt = A0;
const byte led_Z = D7; // vse je OK
const byte led_O = D6; // nizky stav nejakeho modulu
const byte led_R = D5; // kritikcy stav nejakeho modulu / uzivatel v nesnazi
const byte repro = D1; // uzivatel v nesnazi
const byte tlacitko = D2; // zruseni stavu nesnaze

// definovani ID
const byte id_esp = 4;

// definovani promennych a konstant
const uint8_t nizky_stav = 135; //((4,8V/110 000R)*10 000R = 0,436/(3,3/1024) = 135
uint32_t StavBatt = 0;
uint16_t adBatt = 0;

byte id; //oznaceni ESP
short batt; //stav baterie
bool tl_nouze; //volani o pomoc

String readings; // string pro posilani zarizenim v siti
Scheduler userScheduler; // kontrola uloh
painlessMesh mesh; // vytvoreni objektu mesh

// User stub
void sendMessage() ; // Prototype so PlatformIO doesn't complain
String getReadings(); // Prototyp posilani dat

Task taskSendMessage( TASK_SECOND * 10 , TASK_FOREVER, &sendMessage ); //vytvori se uloha
pro posilani zpravy

// struktura pro odeslani informaci
String getReadings () {
  JSONVar jsonReadings;
  jsonReadings["id"] = id_esp; // id
  adBatt= analogRead(pin_batt); //
  jsonReadings["batt"] = analogRead(pin_batt); // asi predelat
  readings = JSON.stringify(jsonReadings); // prevedeno na retezec JSON
  return readings;
}
//-----ESP_MASH-----
// funkce pro odeslani zpravy do site
void sendMessage() {
```

```

    String msg = getReadings(); //vytevorí se text
    mesh.sendBroadcast( msg ); //odeslání zpravy do site mesh
}
// funkce pro tisk přijaté zpravy
void receivedCallback( uint32_t from, String &msg ) {
    Serial.printf("přijata zprava %u msg=%s\n", from, msg.c_str());
    JSONVar myObject = JSON.parse(msg.c_str());
    id = myObject["id"];
    batt = myObject["batt"];
    tl_nouze = myObject["tl_nouze"];
}
// funkce se spustí při identifikaci nového zarizrno v siti
void newConnectionCallback(uint32_t nodeId) {
    Serial.printf("Nove pripojeni, nodeId = %u\n", nodeId);
}
// funkce se spustí při změně pripojení v siti
void changedConnectionCallback() {
    Serial.printf("Zmena pripojeni\n");
}
// funkce se spustí při synchronizaci času v siti
void nodeTimeAdjustedCallback(int32_t offset) {
    Serial.printf("Ladeni casu %u. Offset = %d\n", mesh.getNodeTime(),offset);
}
void startMesh(){
    //mesh.setDebugMsgTypes( ERROR | MESH_STATUS | CONNECTION | SYNC | COMMUNICATION | GENERAL |
MSG_TYPES | REMOTE ); // all types on
    mesh.setDebugMsgTypes( ERROR | STARTUP ); // set before init() so that you can see startup
messages

    mesh.init( MESH_PREFIX, MESH_PASSWORD, &userScheduler, MESH_PORT ); // inicializace
nastevne site ESP
    mesh.onReceive(&receivedCallback);
    mesh.onNewConnection(&newConnectionCallback);
    mesh.onChangeedConnections(&changedConnectionCallback);
    mesh.onNodeTimeAdjusted(&nodeTimeAdjustedCallback);

    userScheduler.addTask( taskSendMessage ); // userScheduler
spousti ulohy ve spravny cas
    taskSendMessage.enable(); // povoleni vysilani
zprav do site
}
//-----Setup-----
void setup() {
    Serial.begin(115200);
    //nastaveni IN/OUT
    pinMode(pin_batt, INPUT);
    //pinMode(tlacitko, INPUT);
    pinMode(led_R, OUTPUT);
    pinMode(led_O, OUTPUT);
    pinMode(led_Z, OUTPUT);
    //pinMode(repro, OUTPUT);

    startMesh();
    Serial.println(WiFi.macAddress());
}
//-----Hlavni program-----
-
void loop() {
    mesh.update();

    // kontrola stavu baterie

```

```
if(((millis() - StavBatt) > STAV_BATT) && (adBatt <= nizky_stav)){  
    digitalWrite(led_R, HIGH);  
    delay(20);  
    StavBatt = millis();  
    digitalWrite(led_R, LOW);  
}  
}
```

Příloha F – Program pro ESP maják s webovým serverem

```
#include "painlessMesh.h"
#include <Arduino_JSON.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include "IPAddress.h"

#define MESH_PREFIX "NavPrNevidim" //nazev site
#define MESH_PASSWORD "8695123!!AA" //heslo site
#define MESH_PORT 5555 //komunikacni TCP port

#define STATION_SSID "ESP8266-Access-Point"
#define STATION_PASSWORD "123456789"

#define HOSTNAME "HTTP_BRIDGE"

#define CHANNEL_SLAVE 1 // definovani kanalu ESP_MESH
#define STAV_BATT 10000 // ms
#define POC_MAJAKU 5
#define POMOC_CESTA 120000 // ms

// definoovani pinu
const byte pin_batt = A0;
const byte led_Z = D7; // vse je OK
const byte led_O = D6; // nizky stav nejakeho modulu
const byte led_R = D5; // kriticky stav nejakeho modulu / uzivatel v nesnazi
const byte led_Debug = D0; // dioda urcena pro debug systemu
const byte repro = D1; // uzivatel v nesnazi
const byte tlacitko = D2; // zruseni stavu nesnaze
//const byte led_ESP = 2; // led na modulu ----> pin2 = D4 , UART1

// definovani ID
const byte id_esp = 1;

// konfigurace site
//const char* ssid = "ESP8266-Access-Point";
//const char* password = "123456789";
WiFiServer server(80); // nastaveni port serveru
String header; // vytvoreno nove stranky

// definovani promennych a konstant
const uint16_t nizky_stav = 135; //((4,8V/110 000R)*10 000R = 0,436/(3,3/1024) = 135
uint32_t StavBatt = 0;
uint16_t batt_majaku[POC_MAJAKU]; //vyska pole 10
bool pomoc_majaku[POC_MAJAKU]; //vyska pole 10
uint32_t web_cekani = 0;
uint32_t reset_priznaku = 0;
bool pomoc_na_ceste = false;

uint16_t connectTimeOutPerAP=5000;

typedef struct str_uzel {
    byte id; //oznaceni ESP
    short batt; //stav baterie
    bool tl_nouze; //volani o pomoc
} str_uzel;
```

```

String readings;          // string pro posilani zarizenim v siti
Scheduler userScheduler; // kontrola uloh
painlessMesh mesh;       // vytvoreni objektu mesh
ESP8266WiFiMulti wifi_multi;

// User stub
void sendMessage() ;      // Prototype so PlatformIO doesn't complain
String getReadings();     // Prototyp posilani dat

Task taskSendMessage( TASK_SECOND * 6 , TASK_FOREVER, &sendMessage ); //vytvori se uloha
pro posilani zpravy

// struktura pro odeslani informaci
String getReadings () {
  JSONVar jsonReadings;
  jsonReadings["id"] = id_esp; // id
  batt_majaku[id_esp] = analogRead(pin_batt); // asi predelat na NULL
  jsonReadings["batt"] = batt_majaku[id_esp]; // asi predelat na NULL
  jsonReadings["tl_nouze"] = pomoc_na_ceste; //
  readings = JSON.stringify(jsonReadings); // prevedeno na retezec JSON
  return readings;
}

// funkce pro odeslani zpravy do site
void sendMessage() {
  String msg = getReadings(); //vytevorí se text
  //msg += mesh.getNodeId(); //prida se ID zarizeni
  mesh.sendBroadcast( msg ); //odeslani zpravy do site mesh
  //taskSendMessage.setInterval( random( TASK_SECOND * 1, TASK_SECOND * 5 )); //po odeslani je
  //treba definovat interval odeslani dalsi zpravy
}

// funkce pro tisk prijate zpravy
void receivedCallback( uint32_t from, String &msg ) {
  Serial.printf("prijata zprava %u msg=%s\n", from, msg.c_str());
  JSONVar myObject = JSON.parse(msg.c_str());
  uint16_t id = myObject["id"];
  batt_majaku[id] = myObject["batt"]; //ulozeni dat do pole
  pomoc_majaku[id] = myObject["tl_nouze"]; //ulozeni dat do pole
}

// funkce se spusti pri identifikaci noveho zarizrno v siti
void newConnectionCallback(uint32_t nodeId) {
  Serial.printf("Nove pripojeni, nodeId = %u\n", nodeId);
}

// funkce se spusti pri zmene pripojeni v siti
void changedConnectionCallback() {
  Serial.printf("Zmena pripojeni\n");
}

// funkce se spusti pri synchronizaci casu v siti
void nodeTimeAdjustedCallback(int32_t offset) {
  Serial.printf("Ladeni casu %u. Offset = %d\n", mesh.getNodeTime(),offset);
}

//
void startMesh(){
  //mesh.setDebugMsgTypes( ERROR | MESH_STATUS | CONNECTION | SYNC | COMMUNICATION | GENERAL |
  MSG_TYPES | REMOTE ); // all types on
  mesh.setDebugMsgTypes( ERROR | STARTUP ); // set before init() so that you can see startup
  messages
}

```

```

    mesh.init( MESH_PREFIX, MESH_PASSWORD, &userScheduler, MESH_PORT ); // inicializace
nastevne site ESP
    mesh.onReceive(&receivedCallback);
    mesh.onNewConnection(&newConnectionCallback);
    mesh.onChangedConnections(&changedConnectionCallback);
    mesh.onNodeTimeAdjusted(&nodeTimeAdjustedCallback);

    userScheduler.addTask( taskSendMessage ); // userScheduler
spusti ulohy ve spravny cas
    taskSendMessage.enable(); // povoleni vysilani
zprav do site
}
//-----Setup-----
void setup() {
    Serial.begin(115200); // UART0 - GPIO1 (TX) and GPIO3 (RX)
    Serial1.begin(9600); // UART1 - GPIO2 (TX)

    //nastaveni IN/OUT
    pinMode(pin_batt, INPUT);
    pinMode(tlacitko, INPUT);
    pinMode(led_R, OUTPUT);
    pinMode(led_O, OUTPUT);
    pinMode(led_Z, OUTPUT);
    pinMode(led_Debug, OUTPUT);
    pinMode(repro, OUTPUT);

    batt_majaku[id_esp] = analogRead(pin_batt);

    IPAddress local_IP(10, 190, 10, 100);

    wifi_multi.addAP(MESH_PREFIX,MESH_PASSWORD);
    wifi_multi.addAP(STATION_SSID,STATION_PASSWORD);

    Serial.print("Pripojovani k Wi-Fi...");
    while(wifi_multi.run(connectTimeOutPerAP)!=WL_CONNECTED)
    {
        Serial.print(".");
        delay(1000);
    }
    Serial.println();
    Serial.print("Pripojeni k: ");
    Serial.println(WiFi.SSID());
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());

    startMesh();

    // WiFi.mode(WIFI_AP_STA);
    // WiFi.softAP(ssid, password);
    delay(10);
    Serial.print("ESP-MESH: IP adresa serveru: ");
    Serial.println(mesh.getAPIP());
    server.begin();
}

void loop() {
    mesh.update();
    //delay(100);

    WiFiClient client = server.available(); // naslouchani pro klienty

```

```

if (client && ((millis() - web_cekani) > 1000)) { // pokud je pripojen klient +
vyhozeni
Serial.println("New Client.");
String currentLine = "";
while (client.connected()) {
if (client.available()) {
char c = client.read(); // ulozeni klienta do c
Serial.write(c); // vypisk klienta do konzole
header += c;
if (c == '\n') {

if (currentLine.length() == 0) {

client.println("HTTP/1.1 200 OK");
client.println("Content-type:text/html");
client.println("Connection: close");
client.println();

// HTML hlavicka
client.println("<!DOCTYPE html><html>");
client.println("<head><meta name=\"viewport\" content=\"width=device-width,
initial-scale=1\">");
client.println("<link rel=\"icon\" href=\"data:,\>");

client.println("<style>html { font-family: Helvetica; display: inline-block;
margin: 0px auto; text-align: center;}");
client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor:
pointer;}");
client.println(".button2 {background-color: #555555;}</style></head>");

// Telo stranky
client.println("<body><h1>ESP8266 Web Server</h1>");
for(byte i =1; i < POC_MAJAKU; ++i){
client.print("modul "); client.print(i); client.print(" - stav baterie: ");
if(batt_majaku[i] <= nizky_stav){
client.print("Potreba vymenit");
}
else{
client.print("OK");
}

if(pomoc_majaku[i] == true){
client.println(" - volani o POMOC !!!! <br>");
}
else{
client.println("<br>");
}
}
}
client.println("</body></html>");

client.println();
break;
} else {
currentLine = "";
}
} else if (c != '\r') {
currentLine += c;
}
}
}
}

```

```

// vymazani header
header = "";
// ukonceni spojeni
client.stop();
Serial.println("Client disconnected.");
Serial.println("");
//delay(100);
web_cekani = millis();
}

// kontrola stavu baterie
if(((millis() - StavBatt) > STAV_BATT) && (batt_majaku[id_esp] <= nizky_stav)){
    digitalWrite(led_R, HIGH);
    delay(20);
    StavBatt = millis();
    digitalWrite(led_R, LOW);
}

//batt_majaku[id_esp] = analogRead(pin_batt);

// komunikacni rozhrani
for(byte i =1; i < POC_MAJAKU; ++i){
    if(pomoc_majaku[i] == true){
        digitalWrite(led_R, HIGH);
        break;
    }
    if(i == (POC_MAJAKU-1))
        digitalWrite(led_R, LOW);
}

for(byte i =1; i < POC_MAJAKU; ++i){
    if(batt_majaku[i] <= nizky_stav){
        digitalWrite(led_O, HIGH);
        analogWrite(repro,512);          //PWM 50% ON
        break;
    }
    if(i == (POC_MAJAKU-1))
        digitalWrite(led_O, LOW);
        analogWrite(repro,0);          //PWM 0% OFF
}

if(digitalRead(led_R) == false && digitalRead(led_O) == false)
    digitalWrite(led_Z, HIGH);
else
    digitalWrite(led_Z, LOW);

if(digitalRead(tlacitko) == true){
    for(byte i = 1; i < POC_MAJAKU; ++i){
        pomoc_majaku[i] =0;
    }
    pomoc_na_ceste = true;
}

if(((millis() - reset_priznaku) > POMOC_CESTA) && pomoc_na_ceste == true){
    reset_priznaku = millis();
    pomoc_na_ceste = false;
}
}

```


Příloha G – Program pro ESP navigační modul

```
#include "painlessMesh.h"
#include <Arduino_JSON.h>
#include <ESP8266WiFi.h>
#include <Ticker.h>
#include "DYPlayerArduino.h"

#define MESH_PREFIX "NavPrNevidim" //navez site
#define MESH_PASSWORD "8695123!!AA" //heslo site
#define MESH_PORT 5555 //komunikacni TCP port

#define CHANNEL_SLAVE 1 // definovani kanalu ESP_MESH
#define pocet_ESP 10 // max pocet ESP
#define DEBUG 1 // vypis na UART pri debug
#define RSSI_VZDALENOST -40 // rozhodovaci vzdalenost pri navigaci --> silnejsi signal = bliz
#define PRUMER_SKOKY 15 // rozhodovaci uroven nahlich skoku RSSI
#define TLACITKO_NOUZE_DELKA_STISKU 7000
#define POMOC_DELKA_VYSILANI 7000
#define STAV_ZARIZENI_CAS_OSESLANI 100000
#define STAV_BATT 10000 // ms

// definoovani pinu
const byte pin_batt = A0;
const byte led_Z = D7; // vse je OK
const byte led_O = D6; // nizky stav nejakeho modulu
const byte led_R = D5; // kritikcy stav nejakeho modulu / uzivatel v nesnazi
const byte led_Debug = D0; // dioda urcena pro debug systemu
const byte repro = D1; // uzivatel v nesnazi
const byte tlacitko = D2; // zruseni stavu nesnaze
//const byte led_ESP = 2; // led na modulu ----> pin2 = D4 , UART1

// definovani ID
const byte id_esp = 2;

// definovani promennych a konstant
bool wifi = 0;
bool show_hidden = true;
bool timer_on = false;
const unsigned long cas_na_wifi = 3*312500; // nastaveni 3s 1 tick = 3.2us
const unsigned long cas_povoleni_hlasky = 5*312500; // nastaveno 5s max: 26.8435424 s
312500= 1s
bool tlacitko_stisknuto = false;
bool trasa_zpet = false;
int16_t majak[pocet_ESP][7]; // pole vyska 10, sirka 7 //index od 0
byte pocet_majak= 0;
int16_t temp;
int16_t poloha1 = 0;
int16_t poloha1_s = 0;
int16_t poloha2 = 0;
int16_t poloha2_s = 0;
int8_t pol_pocet = 0;
int8_t prumerVzdalenosti1;
int8_t prumerVzdalenosti2;
bool hlaska = false;
int8_t tlacitko_pocet = 0;
uint32_t tlacitko_cas = 0xFFFFFFFF;
```

```

uint32_t cas_odeslani_stavu = 60000;
bool pomoc = false;
uint16_t pocitadloStavZarizeni = 0;
const uint8_t nizky_stav = 135; //((4,8V/110 000R)*10 000R = 0,436/(3,3/1024) = 135
uint32_t StavBatt = 0;
uint8_t pocHlasky = 0;
bool hlaska_5 = false;
bool rdy = false;
//-----
byte id; //oznaceni ESP
short batt; //stav baterie
bool pomoc_na_cestech; //volani o pomoc
//-----
//trasa 1
int16_t bod1[] = {0x48, 0x3F, 0xDA, 0x04, 0x86, 0x21, -120}; //modul 1;
int16_t bod2[] = {0xAC, 0x0B, 0xFB, 0xDA, 0xBE, 0x0A, -120}; //modul 3;
int16_t bod3[] = {0x18, 0xFE, 0x34, 0xD8, 0xBD, 0x32, -120}; //modul 4;

bool fin_bod1 = false; // priznak dosazeni bodu
bool fin_bod2 = false; // priznak dosazeni bodu
bool fin_bod3 = false; // priznak dosazeni bodu

//int16_t bodX[] = {0xbc, 0xDD, 0xC2, 0x6C, 0xF3, 0x4A, -120}; //modul 2; TOTO JE TENTO MODUL
//-----
#define prumPocePrvku 5
int16_t prum1[prumPocePrvku];
uint8_t prum1_ukazatel = 0;
int16_t prum2[prumPocePrvku];
uint8_t prum2_ukazatel = 0;
bool prumReady1 = false;
bool prumReady2 = false;
//-----
String readings; // string pro posilani zarizenim v siti
Scheduler userScheduler; // kontrola uloh
painlessMesh mesh; // vytvoreni objektu mesh
DY::Player player(&Serial1); // Inicializuje "player" na UART1

// User stub
void sendMessage(); // Prototype so PlatformIO doesn't complain
String getReadings(); // Prototyp posilani dat

Task taskSendMessage( TASK_SECOND * 1 , TASK_FOREVER, &sendMessage ); //vytvori se uloha
pro posilani zpravy

// struktura pro odeslani informaci
String getReadings () {
  JSONVar jsonReadings;
  jsonReadings["id"] = id_esp; // id
  jsonReadings["batt"] = analogRead(pin_batt); // asi predelat na NULL
  jsonReadings["tl_nouze"] = pomoc; //
  readings = JSON.stringify(jsonReadings); // prevedeno na retezec JSON
  return readings;
}
//-----ESP_MASH-----
// funkce pro odeslani zpravy do site
void sendMessage() {
  String msg = getReadings(); //vytevoru se text
  mesh.sendBroadcast( msg ); //odeslani zpravy do site mesh
  //taskSendMessage.setInterval( random( TASK_SECOND * 1, TASK_SECOND * 5 )); //po odeslani je
  treba definovat interval odeslani dalsi zpravy
}

```

```

}
// funkce pro tisk prijate zpravy
void receivedCallback( uint32_t from, String &msg ) {
    if(0)
        Serial.printf("prijata zprava %u msg=%s\n", from, msg.c_str());
    JSONVar myObject = JSON.parse(msg.c_str());
    id = myObject["id"];
    batt = myObject["batt"];
    pomoc_na_ceste = myObject["tl_nouze"];
}
// funkce se spusti pri identifikaci noveho zarizrno v siti
void newConnectionCallback(uint32_t nodeId) {
    Serial.printf("Nove pripojeni, nodeId = %u\n", nodeId);
}
// funkce se spusti pri zmene pripojeni v siti
void changedConnectionCallback() {
    Serial.printf("Zmena pripojeni\n");
}
// funkce se spusti pri synchronizaci casu v siti
void nodeTimeAdjustedCallback(int32_t offset) {
    Serial.printf("Ladeni casu %u. Offset = %d\n", mesh.getNodeTime(),offset);
}
void startMesh(){
    //mesh.setDebugMsgTypes( ERROR | MESH_STATUS | CONNECTION | SYNC | COMMUNICATION | GENERAL |
MSG_TYPES | REMOTE ); // all types on
    mesh.setDebugMsgTypes( ERROR | STARTUP );

    mesh.init( MESH_PREFIX, MESH_PASSWORD, &userScheduler, MESH_PORT ); // inicializace
nastevne site ESP
    mesh.onReceive(&receivedCallback);
    mesh.onNewConnection(&newConnectionCallback);
    mesh.onChangedConnections(&changedConnectionCallback);
    mesh.onNodeTimeAdjusted(&nodeTimeAdjustedCallback);

    userScheduler.addTask(taskSendMessage); // userScheduler spusti
ulohy ve spravny cas
    taskSendMessage.enable(); // povoleni vysilani
zprav do site
}
//-----Vypocet-----
int8_t prum1Vypocet(){
    int8_t prumCislo = ((prum1[0] + prum1[1] + prum1[2] + prum1[3] + prum1[4]) /
prumPocePrvku);
    return prumCislo;
}
int8_t prum2Vypocet(){
    int8_t prumCislo = ((prum2[0] + prum2[1] + prum2[2] + prum2[3] + prum2[4]) /
prumPocePrvku);
    return prumCislo;
}
// naplneni pole (rozdil RSSI)
void prum1Naplnit(int16_t pol){
    if(prum1_ukazatel >= prumPocePrvku){
        prum1_ukazatel = 0;
    }
    prum1[prum1_ukazatel]= pol;
    prum1_ukazatel = prum1_ukazatel + 1;
    if((prum1_ukazatel == prumPocePrvku) && prumReady1 == false)
        prumReady1 = true; //nastaveni priznaku pro validni hodnotu
= true
}
}

```

```

// naplneni pole (rozdil RSSI)
void prum2Naplnit(int16_t pol){
    if(prum2_ukazatel >= prumPocePrvku){
        prum2_ukazatel = 0;
    }
    prum2[prum2_ukazatel]= pol;
    prum2_ukazatel = prum2_ukazatel + 1;
    if((prum2_ukazatel == prumPocePrvku) && prumReady2 == false) //nastaveni priznaku pro
validni hodnotu = true
        prumReady2 = true;
}
// uprava dat
void upravaDatPolohy(){
    if((poloha1 >= 120) || (poloha2 >= 120)){
        Serial.println("DEBUG: kontrola site: chyba ");
        if(hlaska == false){
            player.playSpecified(9);
            hlaska = true;
            timer1_write(cas_povoleni_hlasky); //nasatveni casu
            timer_on = true;
        }
    }else{
        // oriznuti spisek nahlou zmenou signalu
        if((prumReady1 == true) && (prumReady2 == true)){
            int16_t roz_poloha = poloha1_s - poloha1;

            if((abs(roz_poloha) > PRUMER_SKOKY)){
                Serial.print("DEBUG: desekce spicky RSSI: ");Serial.print(poloha1);Serial.print("
nahrazeno ");Serial.print(poloha1_s); Serial.println(""); //DEBUG
                poloha1 = poloha1_s;
            }else{
                poloha1_s = poloha1;
            }
        }
        // oriznuti spisek nahlou zmenou signalu
        if((prumReady1 == true) && (prumReady2 == true)){
            int16_t roz_poloha = poloha2_s - poloha2;

            if((abs(roz_poloha) > PRUMER_SKOKY)){
                Serial.print("DEBUG: desekce spicky RSSI: ");Serial.print(poloha2);Serial.print("
nahrazeno ");Serial.print(poloha2_s); Serial.println(""); //DEBUG
                poloha2 = poloha2_s;
            }else{
                poloha2_s = poloha2;
            }
        }
        // uprava dat pro polohu1
        prum1Naplnit(poloha1);
        prumerVzdalenosti1= prum1Vypocet();
        if(DEBUG)
            Serial.print("DEBUG: prumer1 ");Serial.print(prumerVzdalenosti1);Serial.println("");
//DEBUG

        // uprava dat pro polohu1
        prum2Naplnit(poloha2);
        prumerVzdalenosti2= prum2Vypocet();
        if(DEBUG)
            Serial.print("DEBUG: prumer2 ");Serial.print(prumerVzdalenosti2);Serial.println("");
//DEBUG
    }
}
}

```

```

//-----navigace-----
// (hodnota RSSI nasledujiciho, rozdil nasledujiciho, rozdil predchoziho)
bool navigace(int16_t rss, int8_t nasl, int8_t pred){
  if(rss < RSSI_VZDALENOST){
    if((nasl == 0) && (pred == 0)){
      Serial.println("DEBUG: navigace: poloha se nemeni");
      return false;      //poloha se nemeni
    }else{
      if(nasl > pred/*(nasl > 0) && (pred <= 0)*/){
        Serial.println("DEBUG: navigace: spravny smer");
        if(hlaska == false){
          player.playSpecified(1);
          hlaska = true;
          timer1_write(cas_povoleni_hlasky);      //nasatveni casu
          timer_on = true;
        }
        return false;      //spravny smer
      }else{
        Serial.println("DEBUG: navigace: spatny smer");
        if(hlaska == false){
          player.playSpecified(4);
          hlaska = true;
          timer1_write(cas_povoleni_hlasky);      //nasatveni casu
          timer_on = true;
        }
        return false;      //spatny smer
      }
    }
  }
}
}
}
}
//-----WIFI_majak-----
// konfigurace AP SSID
void configDeviceAP() {
  WiFi.mode(WIFI_AP_STA);
  String Prefix = "ESP_MASH:";
  String Mac = WiFi.macAddress();
  String SSID = Prefix + Mac;
  String Password = "123456789";
  bool result = WiFi.softAP(SSID.c_str(), Password.c_str(), CHANNEL_SLAVE, show_hidden);
  if (!result) {
    Serial.println("AP nastaveni selhalo.");
  } else {
    Serial.println("AP nastaveni bylo uspesne, SSID: " + String(SSID));
  }
}

// funkce vyhledani WIFI stanic v dosahu
void vypis(byte n){
  Serial.println("Wifi sken spusteno");
  if (n == 0) {
    Serial.println("nebyla nalezena zadna sit");
  } else {
    Serial.print(n);
    Serial.println("site nalezeny");
    for (int i = 0; i < n; ++i) {
      // Print SSID a RSSI
      Serial.print(i + 1);
    }
  }
}

```

```

Serial.print(" ");

// MAC adresa
Serial.print(WiFi.BSSIDstr(i));
Serial.print(" ");

//silna signalu v dBm
Serial.print(WiFi.RSSI(i));
Serial.print("dBm");
Serial.print(" ");

// Nazev site
Serial.println(WiFi.SSID(i)); // SSID
//String BSSIDstr = WiFi.BSSIDstr(i);

    delay(10);
}
}
Serial.println("");
WiFi.scanDelete();
}

//-----WIFI-sken a RSSI-----
void skenARSSI(byte n){
    pocet_majak = 0; // nulovani pocitadla
    if (n == 0) {
        if(DEBUG)
            Serial.println("nebyla nalezena zadna sit");
    } else {
        if(DEBUG)
            Serial.println("site nalezeny");
        for (byte i = 0; i < n; ++i) {
            String SSID = WiFi.SSID(i);
            String BSSIDstr = WiFi.BSSIDstr(i);
            int16_t RSSI = WiFi.RSSI(i);
            delay(2);
            if(SSID.indexOf(MESH_PREFIX) == 0){ //vyhledani zarizeni v siti ESPnavigace
                int mac[6];
                if( 6 == sscanf(BSSIDstr.c_str(), "%02x:%02x:%02x:%02x:%02x:%02x", &mac[0], &mac[1],
&mac[2], &mac[3], &mac[4], &mac[5]) ) {
                    for (int ii = 0; ii < 6; ++ii) {
                        majak[pocet_majak][ii] = (uint8_t) mac[ii];
                    }
                    majak[pocet_majak][0] = majak[pocet_majak][0] - 2; // KOREKCE: nekde dochazi z
kamene MAC adresy
                    majak[pocet_majak][6] = RSSI;
                    pocet_majak = pocet_majak+1; // aktualizace pocitadla
                }
            }
        }
    }
    WiFi.scanDelete(); //vymazani siti z pameti
}

//-----Pruchod polem a vyhledani
vzdalenosti
// n -> index prochazeni pole (pocet_majak, bod[0], bod[1], bod[2], bod[3], bod[4], bod[5])
int16_t najitVzdalenost(byte n, uint8_t b0, uint8_t b1, uint8_t b2, uint8_t b3, uint8_t b4,
uint8_t b5, int16_t b6){
    int16_t vzdalenost = 120;
    for(int i = 0; i < n; ++i){

```



```

        vzda enost = majak[i][6] - b6;          // odedcteni hodnot RSSI -> -50 -(-120)= 70 ->
priblizuji // -60 - (-50) = -10 -> oddaluji
        temp = majak[i][6];                    // ulozeni pro dalsi vypocet
    }
}
return vzdalenost;
}
//-----Timer-----
void ICACHE_RAM_ATTR onTimerISR(){
    hlaska = false;          // povoleni hlasky
}
void setTimer1(){
    timer1_write(0);
    timer1_attachInterrupt(onTimerISR);
    timer1_enable(TIM_DIV256, TIM_EDGE, TIM_SINGLE);
}
//-----Preruseni-----
ICACHE_RAM_ATTR void tlacitko_prepuseni_R(){
    if(pomoc == false)
        tlacitko_stisknuto = true;
    tlacitko_cas = millis();
    if(DEBUG)
        Serial.println("DEBUG:-----: bod: tlacitko_prepuseni_vzestup");
// DEBUG
    attachInterrupt(digitalPinToInterrupt(tlacitko), tlacitko_prepuseni_F, FALLING);
// nastaveni preruseni na sestupnou hranu
}
ICACHE_RAM_ATTR void tlacitko_prepuseni_F(){
    tlacitko_cas = 0xFFFFFFFF;
    if(DEBUG)
        Serial.println("DEBUG:-----: bod: tlacitko_prepuseni_sestup");
// DEBUG
    attachInterrupt(digitalPinToInterrupt(tlacitko), tlacitko_prepuseni_R, RISING);
// nastaveni preruseni na nabeznou hranu
}
//-----Odeslani dat do MESH-----
-----
void ZpravaMesh(){
    startMesh();
    mesh.update();
    uint32_t save_millis = millis();
    while((millis() - save_millis) < POMOC_DELKA_VYSILANI){
        mesh.update();
    }
    delay(50);
    WiFi.disconnect();
    WiFi.mode(WIFI_OFF);
    delay(300);
    WiFi.mode(WIFI_STA);
}
//-----Setup-----
void setup() {
    Serial.begin(115200);    // UART0 - GPIO1 (TX) and GPIO3 (RX)
    Serial1.begin(9600);    // UART1 - GPIO2 (TX)

    //nastaveni IN/OUT
    pinMode(pin_batt, INPUT);
    pinMode(tlacitko, INPUT);
    pinMode(led_R, OUTPUT);
    pinMode(led_O, OUTPUT);
    pinMode(led_Z, OUTPUT);
}

```

```

pinMode( ed_Debug, OUTPUT);
pinMode(repro, OUTPUT);

//nastaveni preruseni - spusteni pouze na vzestupnou hranu (zakmit osetren kondenzatorem)
attachInterrupt(digitalPinToInterrupt(tlacitko), tlacitko_prepuseni_R, RISING);

setTimer1();          //nastaveni casovace

// nastaveni zvukoveho modulu
player.begin();      // zahajeni komunikace
player.setVolume(30); // 100% Volume
}
//-----Hlavni program-----
-
void loop() {
  byte pocet_AP = WiFi.scanNetworks(); //spusteni skeni Wifi
  skenARSSI(pocet_AP); //vyhledani ESP

  if(DEBUG){          //DEBUG
    for(byte i = 0; i < pocet_majak; ++i){
      Serial.print("DEBUG: ");
      Serial.print(majak[i][0],HEX);Serial.print(":");Serial.print(majak[i]
[1],HEX);Serial.print(":");Serial.print(majak[i][2],HEX);Serial.print(":");
      Serial.print(majak[i][3],HEX);Serial.print(":");Serial.print(majak[i]
[4],HEX);Serial.print(":");Serial.print(majak[i][5],HEX);Serial.print(" ");
      Serial.print(majak[i][6],DEC);Serial.println("");
    }
  }

  // navadeni k cily
  if((tlacitko_stisknuto == true && trasa_zpet == false)){
    if(fin_bod1 == false/* && trasa_zpet == false*/){
      // ziskani dat pro polohu1
      poloha1 = najitVzdalenost(pocet_majak, bod1[0], bod1[1], bod1[2], bod1[3], bod1[4],
bod1[5], bod1[6]); //vrati vzdalenost bodu1
      bod1[6] = temp;

      // ziskani dat pro polohu2
      poloha2 = najitVzdalenost(pocet_majak, bod2[0], bod2[1], bod2[2], bod2[3], bod2[4],
bod2[5], bod2[6]); //vrati vzdalenost bodu3
      bod2[6] = temp;

      Serial.print("DEBUG: poloha1: ");Serial.println(poloha1);
      Serial.print("DEBUG: poloha2: ");Serial.println(poloha2);

      upravaDatPolohy(); // FCE upravuje data z poloha1, poloha2 a vystupem jsou
prumerVzdalenosti1, prumerVzdalenosti2

      if((fin_bod1 == false) && (prumReady1 == true) && (prumReady2 == true)){
        // navigace k bodu1 -- bod1 predchozi, bod2 nasledujici
        fin_bod1 = navigace(bod2[6], prumerVzdalenosti2, prumerVzdalenosti1);
      }
    }

    if(fin_bod2 == false && fin_bod1 == true){
      // ziskani dat pro polohu1
      poloha1 = najitVzdalenost(pocet_majak, bod2[0], bod2[1], bod2[2], bod2[3], bod2[4],
bod2[5], bod2[6]); //vrati vzdalenost bodu1
      bod2[6] = temp;

      // ziskani dat pro polohu2

```



```

    if((b0==majak[i][0]) && (b1==majak[i][1]) && (b2==majak[i][2]) && (b3==majak[i][3]) &&
(b4==majak[i][4]) && (b5==majak[i][5])){
        vzdalenost = majak[i][6] - b6; // odceteni hodnot RSSI -> -50 -(-120)= 70 ->
priblizuji // -60 - (-50) = -10 -> oddaluji
        temp = majak[i][6]; // ulozeni pro dalsi vypocet
    }
}
return vzdalenost;
}
//-----Timer-----
void ICACHE_RAM_ATTR onTimerISR(){
    hlaska = false; // povoleni hlasky
}
void setTimer1(){
    timer1_write(0);
    timer1_attachInterrupt(onTimerISR);
    timer1_enable(TIM_DIV256, TIM_EDGE, TIM_SINGLE);
}
//-----Preruseni-----
ICACHE_RAM_ATTR void tlacitko_prepuseni_R(){
    if(pomoc == false)
        tlacitko_stisknuto = true;
    tlacitko_cas = millis();
    if(hlaska_5 == false){
        hlaska_5 = true;
        player.playSpecified(5);
        hlaska = true;
        timer1_write(cas_povoleni_hlasky); //nasatveni casu
        timer_on = true;
    }
    if(DEBUG)
        Serial.println("DEBUG:-----: bod: tlacitko_prepuseni_vzestup");
// DEBUG
    attachInterrupt(digitalPinToInterrupt(tlacitko), tlacitko_prepuseni_F, FALLING);
// nastaveni preruseni na sestupnou hranu
}
ICACHE_RAM_ATTR void tlacitko_prepuseni_F(){
    tlacitko_cas = 0xFFFFFFFF;
    if(DEBUG)
        Serial.println("DEBUG:-----: bod: tlacitko_prepuseni_sestup");
// DEBUG
    attachInterrupt(digitalPinToInterrupt(tlacitko), tlacitko_prepuseni_R, RISING);
// nastaveni preruseni na nabeznou hranu
}
//-----Odeslani dat do MESH-----
void ZpravaMesh(){
    startMesh();
    mesh.update();
    uint32_t save_millis = millis();
    while((millis() - save_millis) < POMOC_DELKA_VYSILANI){
        mesh.update();
    }
    delay(50);
    WiFi.disconnect();
    WiFi.mode(WIFI_OFF);
    delay(300);
    WiFi.mode(WIFI_STA);
}
//-----Setup-----
void setup() {
    Serial.begin(115200); // UART0 - GPIO1 (TX) and GPIO3 (RX)
}

```

```

    bod3[6] = temp;

    // ziskani dat pro polohu2
    poloha2 = najitVzdalenost(pocet_majak, bod2[0], bod2[1], bod2[2], bod2[3], bod2[4],
bod2[5], bod2[6]); //vrati vzdalenost bodu3
    bod2[6] = temp;

    upravaDatPolohy(); // FCE upravuje data z poloha1, poloha2 a vystupem jsou
prumerVzdalenosti1, prumerVzdalenosti2

    if((fin_bod1 == false) && (prumReady1 == true) && (prumReady2 == true)){
        // navigace k bodu1 -- bod1 predchozi, bod2 nasledujici
        fin_bod1 = navigace(bod2[6], prumerVzdalenosti2, prumerVzdalenosti1);
        Serial.println("DEBUG: navigace: k bod2");
    }
}
if(fin_bod2 == false && fin_bod1 == true){
    // ziskani dat pro polohu1
    poloha1 = najitVzdalenost(pocet_majak, bod2[0], bod2[1], bod2[2], bod2[3], bod2[4],
bod2[5], bod2[6]); //vrati vzdalenost bodu1
    bod2[6] = temp;

    // ziskani dat pro polohu2
    poloha2 = najitVzdalenost(pocet_majak, bod1[0], bod1[1], bod1[2], bod1[3], bod1[4],
bod1[5], bod1[6]); //vrati vzdalenost bodu3
    bod1[6] = temp;

    upravaDatPolohy(); // FCE upravuje data z poloha1, poloha2 a vystupem jsou
prumerVzdalenosti1, prumerVzdalenosti2

    if((fin_bod2 == false) && (prumReady1 == true) && (prumReady2 == true)){
        // navigace k bodu2 -- bod2 predchozi, bod3 nasledujici
        fin_bod2 = navigace(bod1[6], prumerVzdalenosti2, prumerVzdalenosti1);
        Serial.println("DEBUG: navigace: k bod1");

        if(fin_bod2 == true){
            player.stop();
            delay(1000);
            player.playSpecified(6);
            hlaska = true;
            timer1_write(cas_povoleni_hlasky); //nasatveni casu
            timer_on = true;
            tlacitko_stisknuto = false; //zruseni navigace
            trasa_zpet = false;
            hlaska_5 = false;
            if(DEBUG)
                Serial.println("DEBUG: dokončili jste zpáteční trasu");
        }
    }
}
}

// dojde při dlouhém držení tlačítka
if(digitalRead(tlacitko) == true && (millis() - tlacitko_cas) > TLACITKO_NOUZE_DELKA_STISKU)
{
    pomoc = true;
    digitalWrite(led_R, HIGH); // zapnutí led
    if(fin_bod2 != true){
        player.stop();
        delay(1500);
        if(pomoc_na_cestě == false){

```

```

        player.playSpecified(10);                // pomoc privolana
    }else{
        player.playSpecified(11);                // pomoc na ceste
    }
    hlaska = true;
    timer1_write(cas_povoleni_hlasky);           //nasatveni casu
    timer_on = true;
    tlacitko_stisknuto = false;                  //zruseni navigace
}
if(DEBUG)
    Serial.println("DEBUG:-----: bod: tlacitko_prepuseni: pomoc
privolana"); //DEBUG

    ZpravaMesh();
}

// poslani zpravy o stavu zarizrni
if((millis()/STAV_ZARIZENI_CAS_OSESLANI) > pocitadloStavZarizeni){
    pocitadloStavZarizeni = pocitadloStavZarizeni + 1;
    digitalWrite(led_Debug, HIGH);              // zapnuti led
    ZpravaMesh();
    digitalWrite(led_Debug, LOW);               // vypnuti led
}

// kontrola stavu baterie
if(((millis() - StavBatt) > STAV_BATT) && (analogRead(pin_batt) <= nizky_stav)){
    if(pomoc == true){
        digitalWrite(led_R, LOW);
        delay(200);
        StavBatt = millis();
        digitalWrite(led_R, HIGH);
    }
    else{
        digitalWrite(led_R, HIGH);
        delay(20);
        StavBatt = millis();
        digitalWrite(led_R, LOW);
    }
    if(pochlasky > 5){
        pochlasky=0;
        player.playSpecified(7);
        hlaska = true;
        timer1_write(cas_povoleni_hlasky);       //nasatveni casu
        timer_on = true;
    }
    pochlasky = pochlasky + 1;
}
}
}

```