

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

CHYTRÉ ŘEŠENÍ PRO NOUZOVÉ VOLÁNÍ V AUTOMOBILECH

SMART SOLUTION FOR EMERGENCY CALL IN CARS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Haniška

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Pokorný

BRNO 2021

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Student: Jan Haniška

ID: 211254

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Chytré řešení pro nouzové volání v automobilech

POKYNY PRO VYPRACOVÁNÍ:

Práce se zabývá doplňkovým řešením pro automobily, které z výroby nejsou vybaveny bezpečnostním systémem eCall. Cílem práce bude vytvořit funkční vzorek zařízení komunikující s OBD-II jednotkou automobilu, které bude vyhodnocovat data z automobilu nebo z vlastního akcelerometru a v případě nehody zavolá na tísňovou linku. Součástí zařízení budou tyto, příp. další komponenty: řídicí jednotka - mikrokontrolér nebo mikropočítač, GSM modul, GPS modul, reproduktor, mikrofon, akcelerometr, akumulátor. V praktické části bude realizován emulátor OBD-II jednotky automobilu na libovolné platformě, dále bude realizován funkční vzorek zařízení. Pro tvorbu funkčního vzorku je možné využít vývojových kitů. V rámci bakalářské práce bude realizována alespoň jedna z těchto klíčových částí: (i) komunikační část využívající volání na tísňovou linku, (ii) zjišťování polohy z GPS přijímače nebo (iii) vyhodnocení dat z akcelerometru a implementace akumulátoru. Emulátor včetně funkčního vzorku bude pevně umístěn na tvrdou podložku a bude obsahovat popisky jednotlivých částí. Bude vytvořen návod k jeho obsluze.

DOPORUČENÁ LITERATURA:

[1] MCCORD, Keith. Automotive Diagnostic Systems: Understanding OBD I and OBD II. North branch: CarTech, 2011. ISBN 978-1-934709-06-1.

[2] OORNI, Risto a Ana GOULART. In-Vehicle Emergency Call Services: eCall and Beyond. IEEE Communications Magazine [online]. 2017, 55(1), 159-165 [cit. 2020-09-15]. DOI: 10.1109/MCOM.2017.1600289CM. ISSN 0163-6804. Dostupné z: <http://ieeexplore.ieee.org/document/7823355/>

Termín zadání: 1.2.2021

Termín odevzdání: 31.5.2021

Vedoucí práce: Ing. Jiří Pokorný

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce se zaměřuje na návrh a realizaci prvotního prototypu zařízení, které bude schopno dovybavovat starší vozy systémem tísňového volání podobným zařízením použitým ve službě eCall. Toto dovybavení je provedeno zapojením zařízení do diagnostického konektoru OBD-II (On Board Didagnostics). První část práce popisuje základní technologie a postupy použité při návrhu zařízení, jako jsou sběrnice CAN bus, diagnostický konektor OBD-II, komunikační protokoly používané v automobilovém průmyslu, způsob vyhodnocení nehody a další. Také jsou zde popsány obecné požadavky pro toto zařízení vzhledem ke směrnicím Evropské unie. Praktická část práce se zabývá nejdříve realizací emulátoru řídicí jednotky vozidla potřebného pro vývoj. Dále je zde popsán návrh zařízení, kde jsou vybírány jeho jednotlivé komponenty, jako modul pro komunikaci přes GSM, modul Global Positioning System (GPS), akcelerometr a vývojová deska pro řízení celého zařízení. Poslední část práce je věnována realizaci prototypu zařízení. Jako hlavní stavební blok celého zařízení byla zvolena vývojová deska NUCLEO-F103RB od firmy STMicroelectronics. Ve vývoji softwaru pro desku došlo k použití knihoven HAL od této firmy. Došlo k vytvoření dvou nových knihoven. První s názvem CAN_data_load sloužící ke komunikaci s emulátorem řídicí jednotky pomocí rozhraní CAN bus a následnému zpracování získaných dat a druhá GPS_load_control k získávání dat z GPS modulu a jejich zpracování. Nakonec je popsáno zprovoznění a obsluha vytvořeného experimentálního pracoviště s realizovaným prototypem zařízení. K těmto procedurám byl dodatečně vytvořen i návod.

KLÍČOVÁ SLOVA

eCall, OBD-II, CAN-BUS, NUCLEO, GPS, akcelerometr, Arduino, ELM327, MCP2551, diagnostika vozu

ABSTRACT

This bachelor thesis is focused on design and realization of initial device prototype, which will be able to retrofit older vehicles with the system for emergency calling similar to eCall. Retrofitting is done by connecting the device to the diagnostic connector OBD. First part of thesis describes basic technologies used during device design, as CAN bus, diagnostic connector OBD-II, communication protocols used in automotive industry or evaluation method for crash etc. Also there are described general requirements for this device according to European Union directives. Practical part of thesis is aimed at realization of vehicles electronic control unit emulator needed for development. Then there is described device design, where are chosen components like GPS module, GSM modem, accelerometer and development board for control of entire device. Last part of thesis is focused on realization of device prototype. As main component of entire device was chosen development board NUCLEO-F103RB, made by manufacturer STMicroelectronics. During software development were used HAL libraries from this manufacturer. Also were created two new libraries. First library called CAN_data_load is used for communication between vehicles electronic control unit emulator by CAN bus interface and subsequent data processing and second one called GPS_load_control for data acquisition from GPS module and their processing. Finally there is described commissioning and service of created experimental workplace with realized prototype device. For this procedures were created additional instructions.

KEYWORDS

eCall, OBD-II, CAN-BUS, NUCLEO, GPS, accelerometer, Arduino, ELM327, MCP2551, vehicle diagnostic

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Chytré řešení pro nouzové volání v automobilech“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Velmi rád bych poděkoval vedoucímu bakalářské práce panu Ing. Jiřímu Pokornému za jeho ochotnost pomoci, odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	12
1 Úvod do problematiky	13
1.1 Motivace a funkce zařízení	13
1.2 Konkurenční řešení	13
1.2.1 Zařízení zapojená do 12 V zásuvky zapalování	13
1.2.2 Zařízení zapojená do OBD konektoru	15
1.3 Popis OBD	17
1.3.1 Protokoly OBD-II	17
1.3.2 Sběrnice CAN-BUS	18
1.3.3 Konektor pro OBD-II	18
1.4 eCall	19
1.4.1 Příklady architektury použité pro nasazení v Evropě	20
1.4.2 Požadované parametry	20
1.5 Indikace nehody	23
1.5.1 Zpracování dat z akcelerometru	24
1.5.2 Načtení dat z řídicí jednotky pomocí OBD-II	26
2 Praktická část	27
2.1 Emulace OBD-II jednotky	27
2.2 Návrh zařízení	29
2.2.1 Platforma pro řídicí část zařízení	31
2.2.2 Mikrofon a reproduktor	31
2.2.3 Akcelerometr	32
2.2.4 GSM modem	33
2.2.5 GPS modul	33
2.2.6 Záložní akumulátor	34
2.3 Analýza realizace dílčích součástí zařízení	35
2.4 Realizace prototypu IVS	36
2.4.1 Vyčítání dat z řídicí jednotky pomocí Nuclea	37
2.4.2 Získání dat pro vývojovou desku z GPS modulu	43
2.4.3 Finální podoba zařízení	49
Závěr	56
Literatura	58
Seznam symbolů, veličin a zkratk	60

Seznam příloh	61
A Obsah přílohy	62
B Návod k obsluze vytvořeného pracoviště	63
B.1 Zapojení pracoviště	63
B.1.1 Datové kabely	64
B.1.2 Napájení	64
B.2 Konfigurace obslužného PC	64
B.2.1 Nastavení terminálového emulátoru Tera Term	65
B.2.2 Nastavení programu ArduinoOBDSimulator	65
B.3 Ovládání pracoviště z obslužného PC	66

Seznam obrázků

1.1	Fotografie zařízení Michelin Safe & Drive	14
1.2	Fotografie zařízení Bosch Telematics eCall Plug (TEP)	15
1.3	Fotografie zařízení T-Mobile Chytré auto	16
1.4	Fotografie zařízení PACE	16
1.5	Konektor pro OBD-II	18
1.6	Schéma architektury užitá ve Finsku	20
1.7	Schéma architektury užitá v České republice	21
1.8	Komunikační standardy pro eCall	23
1.9	Zobrazení os akcelerometru vůči vozidlu	25
2.1	Schéma zapojení emulátor OBD-II	28
2.2	Obsluha programu ScanMaster-ELM	28
2.3	Náhled na grafické rozhraní programu k obsluze emulátoru a ukázka signálu na sběrnici CAN	29
2.4	Fotografie realizace experimentálního pracoviště	29
2.5	Blokové schéma IVS.	30
2.6	Fotografie použité vývojové desky NUCLEO-F103RB.	37
2.7	Zapojení části zařízení sloužící ke komunikaci s emulátorem pomocí CAN bus.	38
2.8	Ukázka grafického rozhraní programu STM32CubeMX.	39
2.9	Fotografie desky osazené GPS modulem L80-M39 a stabilizátorem napětí AMS1117-3.	44
2.10	Schéma zapojení desky s GPS modulem.	45
2.11	Fotografie finální podoby zařízení po usazení na pevnou podložku.	50
2.12	Snímek obrazovky PC obsluhující zařízení.	51
2.13	Náhled do Správce zařízení v operačním systému Windows 10.	52
2.14	Konfigurační okno sériové linky v programu Tera Term.	53
2.15	Mapa zobrazující přesnost získaných dat.	55
B.1	Fotografie zařízení usazeného na pevnou podložku.	63
B.2	Konfigurační okno sériové linky programu Tera Term.	65
B.3	Náhled do Správce zařízení v operačním systému Windows 10.	66
B.4	Snímek obrazovky PC obsluhující zařízení.	67

Seznam tabulek

1.1	Označení pinů konektoru pro OBD-II	19
1.2	Struktura zprávy minimum set of data (MSD)	22
1.3	Parametry pro vyhodnocení čelního nárazu	25
1.4	Parametry pro vyhodnocení bočního nárazu	26
2.1	Porovnání typů vývojových desek	31
2.2	Porovnání typů akcelerometrů	33
2.3	Porovnání typů GSM modemů	34
2.4	Porovnání typů GPS modulů	35

Seznam výpisů

2.1	Funkce CAN_MsgSetup pro nastavení hlavičky a dalších parametrů CAN zprávy, která se nachází v souboru CAN_data_load.c.	40
2.2	Funkce CAN_TxMessage pro nastavení datové části zprávy a její odeslání, která se nachází v souboru CAN_data_load.c.	41
2.3	Část kódu ze souboru main.c, která slouží k obsluze CAN komunikace.	42
2.4	Ukázka dat ve tvaru National Marine Electronic Association (NMEA) řetězců, jenž jsou uložena v bufferu po jejich přijetí od Global Positioning System (GPS) modulu.	46
2.5	Kód k obsluze GPS modulu v souboru main.c - část 1.	48
2.6	Kód k obsluze GPS modulu v souboru main.c - část 2.	49
2.7	Příklad výstupu programu nahraného ve vývojové desce Nucleo do terminálu v PC.	54

Úvod

Nehody v silniční dopravě jsou bohužel častým úkazem dnešní doby, zvláště v částech světa jako je Evropa, kde je drtivá část obyvatelstva motorizovaná. Dle dat Českého statistického úřadu se událo na českých silnicích za rok 2019 přes 107 tisíc dopravních nehod. Tyto nehody měly za následek těžká zranění u 2110 osob a u 618 byla následkem smrt[1]. Logicky je cílem všech institucí, jež mohou tato čísla nějak ovlivnit, je snížit. Jedním z výsledků tohoto snažení je i následující prohlášení.

„Všechny nové modely automobilů a dodávek uvedené na trh po 31. březnu 2018 budou muset být vybaveny zařízením tísňového volání eCall, které v případě nehody automaticky vyrozumí záchranné složky.“ (Tisková zpráva Europarlamentu [2]) Vzhledem k tomuto nařízení je v dnešní době ve výbavě většiny nových vozů výše zmíněný eCall (výjimkou jsou modely vozů, jejichž schválení pro provoz bylo provedeno před uvedeným datem). Po silnicích se však nepohybují pouze nové vozy, ale také vozy staršího data výroby, jež tento systém nemají a právě na tyto případy se zaměřuje tato práce. Cílem je vyvinutí doplňkového zařízení, které tuto funkci do těchto vozů implementuje.

Zařízení bude zastupovat eCall, tudíž bude jeho úkolem automaticky kontaktovat záchranné složky pomocí tísňové linky volání 112 v případě vážné nehody. Automatizace tohoto úkonu by měla urychlit reakci záchranných složek, tím zkrátit čas jejich příjezdu a zvýšit šanci na přežití účastníků nehody. Hlavním cílem pro funkci zařízení je tedy záchrana životů, anebo minimalizace zdravotních následků způsobených dopravní nehodou.

1 Úvod do problematiky

Na úvod práce bylo nutné bližší seznámení s užitými technologiemi. Především bylo nutné projít si možnosti komunikace s řídicí jednotkou pomocí On Board Diagnostics (OBD). Dále nás také zajímaly parametry systému eCall, který již výrobci instalují do vozů.

1.1 Motivace a funkce zařízení

Cílem práce je vytvoření zařízení splňující v co největší možné míře definované požadavky k bezpečnostní funkci eCall. Splněním požadavků na naše zařízení by mělo být dosaženo kompatibility s dalšími články tohoto systému, což je samozřejmě žádoucí. Finální produkt by měl být implementovatelný do jakéhokoliv vozu disponujícího rozhraním OBD. Toto rozhraní bude využito jednak jako napájení, ale také jako zdroj informací, jenž pomůže s detekcí nehody a poskytne další použitelná data. Ta mohou být také uložena a zařízení je schopno následně plnit funkci tzv. černé skříňky. Je zde také snaha o nalezení ideálních řešení pro splnění definovaných funkcí, jako například způsob určení počtů osob nacházejících se ve voze. Opodstatněním pro vývoj tohoto zařízení a dovybavování vozů jeho pomocí je skutečnost, že v krátkém časovém horizontu zřejmě nehrozí taková obměna vozového parku v Evropě, aby zde byla většina vozů takto vybavena. Vzhledem k tomu, že průměrné stáří automobilů užívaných v Evropské unii v roce 2018 činilo 10,8 let, můžeme předpokládat, že ještě další desetiletí budou na Evropských silnicích převládat vozy, které nebudou mít od výroby implementovanou funkci eCall [3].

1.2 Konkurenční řešení

Na trhu již existují alternativy In Vehicle System (IVS) zařízení eCall. Tato zařízení implementují některé funkce, jenž jsou požadovány, ale ve většině nedokáží splnit úplné požadavky na IVS eCall. Tato zařízení je možné rozdělit dle způsobu jejich zapojení ve voze na ty, jenž zapojíme do 12 V zásuvky zapalování a ty, jenž se připojují k OBD konektoru.

1.2.1 Zařízení zapojená do 12 V zásuvky zapalování

Tato zařízení jsou si podobná jak vzhledem, tak také principem jejich funkce. Není zde navázáno žádné spojení s řídicí jednotkou vozu, zásuvka zapalování poskytuje pouze napájení. Nejpodstatnějším komponentem těchto zařízení je vestavěný akcelerometr, jenž má za úkol detekovat nehodu zjištěním velkých akcelerací při nárazu.

Data jsou následně pomocí Bluetooth zaslána do chytrého telefonu vybaveného aplikací pro jejich vyhodnocení. V případě, že bude detekována nehoda, tak by mělo být navázáno spojení s tísňovou linkou.

Michelin Safe & Drive

Toto řešení od francouzského výrobce pneumatik bylo představené v roce 2017. V případě nehody by měl být spojen hovor s call centrem společnosti Michelin, jejíž pracovník se pokusí uživatele kontaktovat, v případě žádné reakce předá známé informace a jeho pozici GPS záchranným složkám. Mimo tuto funkcionalitu by měla být také zajištěna nonstop individuální telefonická asistence, která má uživateli pomoci například s nalezením nejbližší čerpací stanice nebo sdělit nejbližší předpověď počasí. Poslední službou, jež zařízení poskytuje je hledání vozu dle jeho poslední známé pozice. Momentálně tento produkt na stránkách firmy Michelin není dohledatelný.



Obr. 1.1: Fotografie zařízení Michelin Safe & Drive [4].

Bosch Telematics eCall Plug (TEP)

Řešení vyvinuté firmou Bosch. Taktéž zde probíhá vyhodnocení nehody pomocí dat z interního akcelerometru. Data jsou vyhodnocena za pomoci algoritmů, jež byly vyvinuty pro airbagy touto firmou. Spojení s tísňovou linkou nebo případné odeslání dat o nehodě je v podstatě totožné jako u předchozího produktu. Kromě funkce tísňového volání je zde také funkce telemetrického vyhodnocování chování řidiče. Firma Bosch dodává pouze zařízení, ale neprovozuje službu k jeho funkci. K této činnosti vybízí potencionální partnery, avšak v současné době žádní provozovatelé této služby nejsou dohledatelní.



Obr. 1.2: Fotografie zařízení Bosch TEP [5].

1.2.2 Zařízení zapojená do OBD konektoru

Na rozdíl od předchozích, jsou tato zařízení z automobilu nejen napájena, ale jsou schopna také komunikovat s řídicí jednotkou vozu a číst z ní data. Většina těchto zařízení je určena především k vytvoření přístupového bodu Wi-Fi a funkce tísňového volání je pro ně jen doplňková.

T-Mobile Chytré auto

Jedná se o produkt zastupující skupinu zařízení především určených k poskytnutí Wi-Fi hotspotu ve vozidle. Zařízení je mimo to schopno základní diagnostiky vozu, určit jeho polohu, snímat otřesy a náklony nebo detekovat jeho odpojení od vozu. Funkce tísňového volání je zde pouze okrajová. Zašle předdefinovanou zprávu spolu s polohou GPS na nanejvýše 5 určených kontaktů, k tomu dojde při stisknutí SOS tlačítka v mobilní aplikaci nestane se tak automaticky. Získání zařízení a jeho funkce je možná pouze při sjednání dlouhodobého smluvního závazku s firmou T-Mobile o využívání jejích služeb.

PACE

Toto zařízení vůbec neimplementuje funkci Wi-Fi hotspotu, ale specializuje se na zpracování dat získaných z řídicí jednotky. Jsou zde poskytovány funkce jako vyhledání vozu podle GPS, vedení elektronické knihy jízd nebo zobrazení dat z vozu



Obr. 1.3: Fotografie zařízení T-Mobile Chytré auto [6].

(rychlost, otáčky motoru, aktuální spotřeba atd.) v reálném čase na obrazovce chytrého telefonu uživatele. Pro nás je však nejzajímavější funkcí možnost automatického tísňového volání. Princip je zde podobný, jako u dříve popsaných řešení zapojitelných do zásuvek zapalování. Zařízení po detekci havárie pošle tuto informaci pomocí Bluetooth do chytrého telefonu uživatele, kde se na 30 sekund zobrazí výstražná zpráva. Pokud uživatel na tuto zprávu nezareaguje a nezruší tísňové volání, tak bude spojen hovor s call centrem výrobce, který se jej pokusí kontaktovat. V případě, že se nedočká odezvy, tak je kontaktována záchranná služba a jsou jí předána data o pozici GPS, jež zařízení poskytne. Tato funkce je však prozatím dostupná pouze v Německu.



Obr. 1.4: Fotografie zařízení PACE [7].

1.3 Popis OBD

OBD je schopnost řídicí jednotky automobilu diagnostikovat stav a případné závady u jeho částí a systému, jež jsou monitorovány různými čidly. V dnešní době se k řídicí jednotce připojujeme pomocí standardizovaných protokolů pro OBD-II (ISO9141) a stejně tak standardizovaného konektoru (ISO 15031-3) pro toto užití (viz Obr. 1.5). Mimo výše zmíněnou kontrolu stavu a zjišťování závad umožňuje i provádění změn. K maximalizaci funkčnosti diagnostiky pomocí OBD jsou potřebné specializované diagnostické programy pro danou značku automobilu např. BimCOM (BMW,Mini), HiCOM (Hyundai,Kia) apod.

1.3.1 Protokoly OBD-II

Rozhraní pro OBD-II v dnešní době může využívat 5 různých signálních protokolů. Většina automobilů využívá pouze jeden z nich.

ISO15765-4 (CAN-BUS)

Nejmodernější protokol, v USA používán u všech automobilů vyrobených po roce 2008. Ke komunikaci užívá piny 6 a 14. Tento protokol má 4 varianty, jež se liší délkou hlavičky a rychlostí sběrnice [8].

- ISO 15765-4 CAN (11 b hlavička, rychlost 500 Kbaud)
- ISO 15765-4 CAN (29 b hlavička, rychlost 500 Kbaud)
- ISO 15765-4 CAN (11 b hlavička, rychlost 250 Kbaud)
- ISO 15765-4 CAN (29 b hlavička, rychlost 250 Kbaud)

ISO14230-4 (KWP2000)

Velmi rozšířený protokol u vozů vyrobených od roku 2003. Využívá K-line podle ISO 9141-2, jenž je v konektoru na pinu 7. Jedná se o starší protokol dnes jej nahrazuje výše zmíněný CAN-BUS. Tento protokol má dvě varianty. Tyto varianty se liší pouze v metodě inicializace komunikace. Rychlost je u obou totožná a to 10400 b/s [9].

- ISO 14230-4 KWP (5 Bd inicializace, 10,4 kBd)
- ISO 14230-4 KWP (rychlá inicializace, 10,4 kBd)

ISO9141-2

Starší protokol používaný zejména u automobilů vyrobených v Evropě mezi lety 2000 až 2004 nebo u vozů asijských výrobců. Užívá pin 7 (K-Line) a volitelně ještě i pin 15 (L-Line). Rychlost přenosu je totožná jako u KWP2000 tedy 10400 b/s [9].

SAE J1850 VPW

Protokol definovaný organizací Society of automotive engineers (SAE). VPW značí **v**ariable **p**ulse **w**idth tedy proměnou šířku pulzu. Využíván především u vozů výrobce General Motors. Využívá pin 2 a napětí pro logickou úroveň 1 je od 3,5 V do 7 V. Délka zprávy je omezena na 12 B včetně CRC. Protokol volí rychlost podle parametrů fyzické vrstvy mezi 10400 b/s a 41600 b/s. Protokol nejdříve ověřuje, jestli není medium pro přenos používáno jiným účastníkem a předchází tak kolizi [8].

SAE J1850 PWM

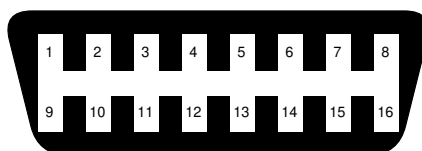
Od předchozího protokolu se liší v několika parametrech. PWM značí **p**ulse-**w**idth **m**odulation tedy modulace šířkou pulzu. Pro komunikaci využívá jak pin 2 tak i pin 10. Logickou úroveň 1 zde představuje napětí 5 V. Protokol komunikuje pouze rychlostí 41600 b/s. Obvykle jej užívají vozy značky Ford [8].

1.3.2 Sběrnice CAN-BUS

Jedná se o datovou sběrnici s jejímž vývojem začala firma Bosch již v roce 1983. Je určena pro komunikaci mezi senzory a funkčními jednotkami v automobilech, ale mimo ně je užívána také v jiných průmyslových odvětvích. Sběrnice sama o sobě je realizovaná jako symetrický nebo asymetrický dvou vodičový obvod, jenž může být také odstíněn. K definování elektrických parametrů pro fyzický přenos na sběrnici slouží norma ISO 11898. CAN protokol pracuje na síťové vrstvě ISO/OSI modelu. Umožňuje konfiguraci a centrální diagnostiku systému. Data vysílána do sběrnice nemají přiřazenou adresu pro jejich doručení, ale obsahuje identifikátor (ID), jenž definuje obsah a prioritu zprávy, pro její odeslání na sběrnici. Čím nižší hodnota identifikátoru tím má zpráva vyšší prioritu. Jednu zprávu odeslanou na sběrnici může přijímat i více zařízení. Maximální rychlost sběrnice je pak 1 Mb/s [10].

1.3.3 Konektor pro OBD-II

Pro připojení k řídicí jednotce pomocí OBD-II je zapotřebí k tomuto účelu standardizovaného konektoru. Jeho popis můžete vidět níže na Obr. 1.5 a Tab. 1.1.



Obr. 1.5: Znázornění konektoru pro užití OBD-II.

Tab. 1.1: Zapojení pinů konektoru

PIN	popis funkce	PIN	popis funkce
1	vyhrazen výrobci	9	vyhrazen výrobci
2	kladný pól sběrnice J1850	10	negativní pól sběrnice J1850
3	vyhrazen výrobci	11	vyhrazen výrobci
4	uzemnění k šasi	12	vyhrazen výrobci
5	uzemnění pro signál	13	vyhrazen výrobci
6	kladný pól pro CAN	14	záporný pól pro CAN
7	ISO 9141-2 K-Line	15	ISO 9141-2 L-Line
8	vyhrazen výrobci	16	kladný pól baterie

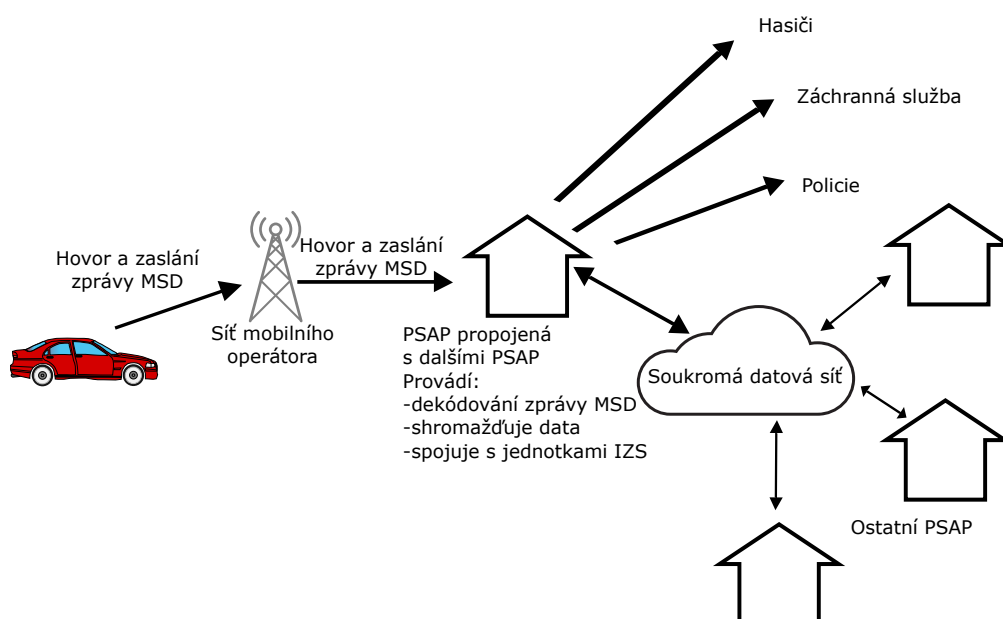
1.4 eCall

Jak již bylo zmíněno, systém eCall je povinnou součástí všech modelů nově vyrobených automobilů, které byly schváleny pro provoz v Evropské unii po datu 31. března 2018. eCall je evropským systémem pro tísňové volání zabudovaný do automobilu, jehož parametry určuje norma EN 16062. Pokud nastane autonehoda, tak systém automaticky spojí hovor s nejbližším public safety answering point (PSAP), jenž poskytne informace složkám integrované záchranné služby (IZS), pomocí celoevropské tísňové linky 112. Před spojením hovoru by měl být odeslán packet minimum set of data (MSD), jenž obsahuje informace o nehodě, jako je pozice nehody, čas nebo Vehicle identification number (VIN). Po odeslání těchto informací, dochází k navázání hovoru mezi PSAP a osobami nacházejícími se ve voze. Měla by zde také být možnost aktivace systému manuálně tlačítkem bez nutnosti detekce nehody. Zavedením systému eCall v Evropské unii se zabývá projekt Harmonised eCall European Pilot (HeERO) [11].

Podobné systémy jsou však vyvíjeny i jinde ve světě například v Rusku se jedná o systém ERA-GLONASS. Tento systém v současnosti není kompatibilní s eCall, avšak není vyloučeno, že tato kompatibilita v budoucnu bude navázána. Ekvivalent eCall byl také vyvinut v Malajsii, kde je zajímavostí způsob řešení odesílání MSD. Zpráva MSD je programem eSpeak převedena do hlasové podoby a ve smyčce přehrávána do hovoru s tísňovou linkou [12].

1.4.1 Příklady architektury použité pro nasazení v Evropě

Pro nasazení eCall v rozsahu, jenž byl popsán, bylo nutné, aby členské státy programu HeERO přizpůsobily svoje PSAP pro funkci tohoto zařízení. Budou uvedeny dva příklady řešení této problematiky. První příklad popisuje řešení architektury systému ve Finsku, kde je pouze omezený počet vzájemně propojených PSAP, kvůli soukromým datovým sítím. Z tohoto důvodu mají všechny Finské PSAP implementovány technologie, pro příjem eCall dat. Přehledněji viz Obr. 1.6. Jako druhý

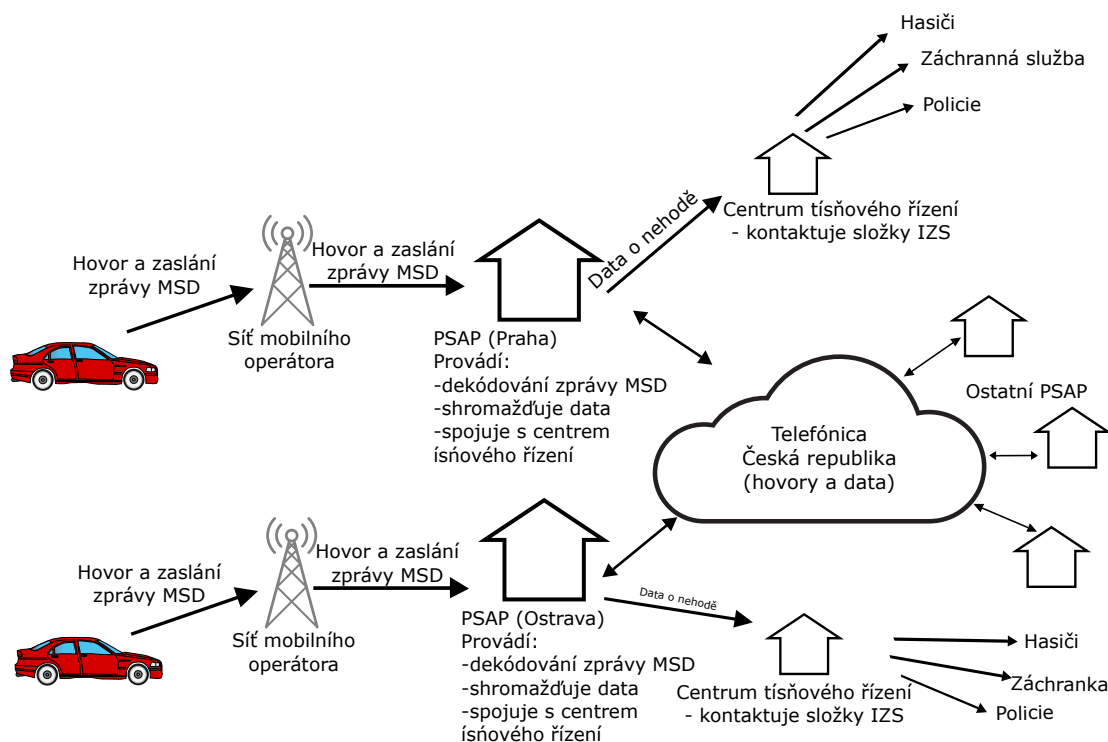


Obr. 1.6: Schéma architektury pro implementaci eCall ve Finsku

příklad bude popsána implementace v České republice. Zde bylo zpracování eCall dat centralizováno do dvou PSAP (v budoucnu se počítá s navýšením počtu PSAP zpracovávajících eCall). PSAP dekóduje zprávu MSD a odešle data o nehodě dále centru tísňového řízení, jenž dále informuje jednotky IZS [11]. Schéma této architektury vizte na Obr. 1.7.

1.4.2 Požadované parametry

Pro zařízení fungující jako IVS pro eCall je definováno několik parametrů a testů, které musí být splněny, aby jej bylo možné certifikovat a následně jím dovybavovat



Obr. 1.7: Schéma architektury pro implementaci eCall v České republice

vozy. Certifikace by měla deklarovat adekvátní spolehlivost a vybavenost. Přesně jsou tyto parametry a testy určeny následujícími normami:

- EN 15722 definuje zprávu MSD
- EN 16062 Vysokoúrovňové aplikační požadavky na eCall (HLAP) s využitím GSM/UMTS sítí s přepojováním okruhů
- EN 16072 Provozní požadavky na Panevropský eCall
- TR 16405 Nepovinné údaje stanovené pro těžká nákladní vozidla eCall
- EN 16102 Provozní požadavky na podporu eCall třetí stranou

Podrobný soupis potřebných norem udává jejich sada od Evropského výboru pro normalizaci CEN TC278 [13].

Obsah zprávy MSD

Zaslání této zprávy je ve své podstatě nejdůležitějším úkolem zařízení. Lze totiž předpokládat, že pokud nastane závažná nehoda, tak nebudou členové posádky kvůli zraněním pravděpodobně schopni komunikovat s dispečerem tísňové linky po spojení hovoru s ním, tudíž bude tato zpráva zásadním a hlavně jediným zdrojem informací pro záchranné složky. Proto je nutné, aby v ní byly sděleny informace [14], které

jsou uvedeny v Tab. 1.2.

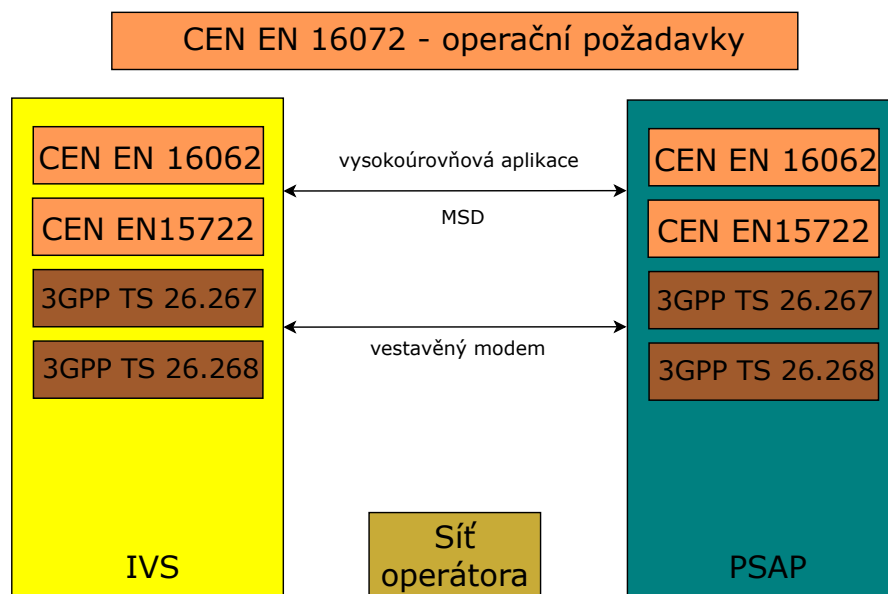
Tab. 1.2: Struktura zprávy MSD

Pořadí	Jméno	Popis
1	Verze formátu	Nastaveno na 1 pro odlišení od pozdějších formátů
2	Identifikátor zprávy	Navyšován s každým opakovaným přenosem
3	Ovládání	Automatické spuštění nebo manuální, indikace důvěryhodnosti pozice, třída vozidla
4	Identifikátor vozu	Číslo VIN dle ISO 3779
5	Typ paliva	Benzín, nafta, atd.
6	Časová známka	Čas kdy nastala nehoda
7	Pozice vozidla	Zeměpisná délka/šířka dle ISO 6709
8	Azimut vozidla ¹	Kroky po 2°
9	Nedávná pozice vozidla č. 1	Zeměpisná délka/šířka
10	Nedávná pozice vozidla č. 2	Zeměpisná délka/šířka
11	Počet osob ve vozidle	Minimálně počet zapnutých bezpečnostních pásů, pokud informace není známá, tak vynecháno
12	Volitelná dodatečná data	např. data o pasažérech

Komunikační standardy

Pro zajištění interoperability v komunikační rovině je potřeba splnit několik standardů. Co se týče komunikace na úrovni GSM a UMTS, tak zde je potřeba se řídit standardy ETSI/3GPP. Další požadavky pro tísňové volání nám určuje sada norem CEN TC278 WG15 “eSafety”. Tyto normy popisují požadavky na vestavěný modemu IVS a PSAP pro vyslání zprávy MSD [13]. Užití standardů v komunikační části systému ukazuje Obr. 1.8.

¹Napomáhá hlavně k lokalizaci nehody na dálnici. Díky azimutu víme, v kterém jejím směru se vůz nachází.



Obr. 1.8: Blokové schéma komunikačních standardů užitých pro eCall.

1.5 Indikace nehody

Pro IVS musí být otestována schopnost detekce 3 druhů nehody. Těmito druhy jsou čelní náraz, boční náraz a náraz zezadu. Hlavním nástrojem pro detekci nehody je akcelerometr. Pro správnou funkci indikace nehody je potřeba analyzovat chování vozu při reálných nehodách. Z této analýzy potřebujeme získat hodnoty akcelerace a úhlové rychlosti, které jsou nutné k určení, zda nastala nehoda. Pokud známe řádové hodnoty těchto parametrů, tak můžeme následně provést kalibraci zařízení, kdy bude možno odfiltrovat provozní otřesy vozu nebo prudké brzdění od srážky. I v případě nehody však nemusí být nutné kontaktování tísňové linky, systém by měl rozlišit, zda nastala nehoda natolik vážná, že vedla k závažným zraněním osob nacházejících se ve voze, jenž vyžadují okamžitou lékařskou pomoc. Od zařízení je požadována nejen detekce kolize, kdy bude pouze určeno, zda nastala nebo ne, ale je požadováno i určení, který ze 3 výše zmíněných typů nehody nastal. Data se také liší v závislosti na třídě vozu vzhledem k jejich různým velikostem a hmotnostem. Na základě toho dělí European New Car Assessment Programme (Euro NCAP) vozy do 10 tříd:

- Supermini
- Malý rodinný vůz
- Velký rodinný vůz

- Vůz vyšší střední třídy
- Malé MPV
- Velké MPV
- Sportovní roadster
- Malý Off-Road 4x4
- Velký Off-Road 4x4
- Pick-up

V okamžiku nárazu dochází k tzv. akceleračním pulzům. Tyto pulzy jsou naším hlavním zájmem při testování IVS. Jejich testování probíhá jak za pomoci vozu, tak i umístěním zařízení na pohyblivou podložku nebo připevněním ke kyvadlu, jsou také definovány rychlosti a úhly srážek. Jako referenční hodnoty jsou brána data Euro NCAP, jelikož zařízení bude určeno pro Evropu a tato organizace zodpovídá za bezpečnost vozů zde prodávaných [13].

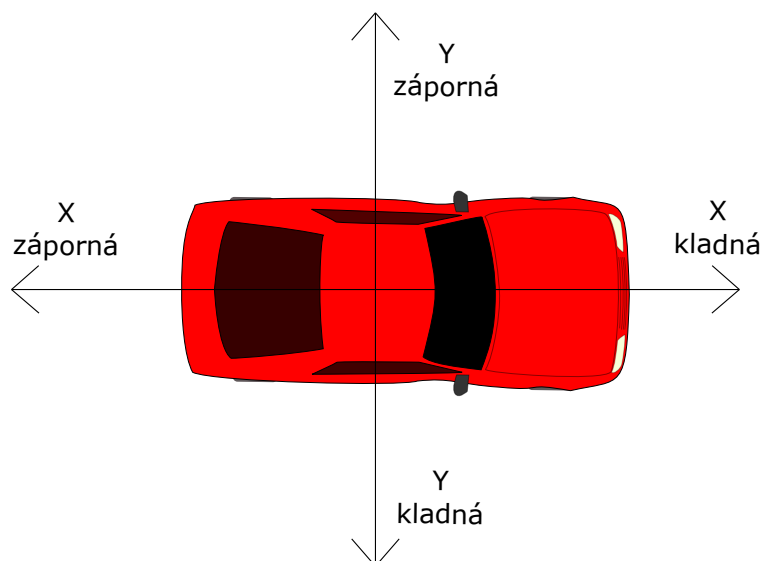
1.5.1 Zpracování dat z akcelerometru

Pro rozlišení typu nehody a jeho závažnosti nás budou zajímat data hlavně data os X a Y z tříosého akcelerometru implementovaného v IVS. Daty z osy X indikujeme čelní náraz při záporných hodnotách a náraz zezadu při kladných, daty z osy Y pak nárazy boční, a to zleva při záporných hodnotách a zprava při kladných (vše můžete názorně vidět na Obr. 1.9). Během nárazu zapříčiní změna zrychlení vozu vytvoření pulzu na výstupu akcelerometru, IVS má následně za úkol tento pulz analyzovat a vyhodnotit. Pulz vždy dosahuje určité amplitudy, tato hodnota je prvním ukazatelem, zda je potřeba se jím zabývat, pokud přesáhne tato hodnota předdefinovanou hranici, tak se posuzuje další parametr, kterým je dostatečné trvání pulzu.

Jestliže obě hodnoty přesáhly požadovanou hranici, tak následuje posun k dalšímu kroku, což je integrace tohoto pulzu. Integrace je provedena postupnou aproximací, kdy je rozdělena křivka na malé úseky a počítá se jejich obsah, jako obsah obdélníku a tyto obsahy jsou následně sečteny. Čím menší jsou tyto úseky, tím přesnější aproximace je dosaženo. Tato operace je popsána vzorcem:

$$\int_{x_0}^{x_n} f(x)dx \approx \sum_{i=1}^n (x_i - x_{i-1}) \cdot f_{avg}[x_{i-1}, x_i] \quad (1.1)$$

Funkcí $f(x)$ je zde myšlena křivka pulzu na výstupu akcelerometru indikovaná při nárazu. Dle takto vypočteného parametru je následně posuzována závažnost incidentu pro určitý typ vozu dle předdefinovaných tabulek [13]. U čelního nárazu je rozlišována závažnost nehody na mírnou a závažnou, tabulku přesných parametrů pro jejich rozlišení můžeme vidět v Tab. 1.3. Při nárazu z boku je detekováno pouze zda incident nastal nebo ne, jeho závažnost již nikoli (viz Tab. 1.4). U nárazu zezadu



Obr. 1.9: Znázornění polohy os vůči vozidlu

také není rozlišována závažnost nehody, navíc již není rozlišována ani třída vozidla, ale pro všechny vozy platí totožné parametry a to:

- Počáteční amplituda zrychlení: 4 G
- Délka trvání pulzu: 110 ms
- Prahová hodnota integrace pro detekci incidentu: 350

Tab. 1.3: Výpis parametrů pro vyhodnocení čelního střetu

Třída vozu	Počáteční amplituda zrychlení	Délka trvání pulzu	Prahová hodnota integrace pro mírný incident	Prahová hodnota integrace pro vážný incident
Velký rodinný vůz	6 G	130 ms	700	1500
Velké MPV	6 G	130 ms	600	1550
Velký Off-Road	6 G	130 ms	600	1500
Pickup	6 G	130 ms	600	1550
Malý rodinný vůz	6 G	130 ms	800	1500
Malé MPV	6 G	130 ms	800	1550
Malý Off-Road	6 G	130 ms	800	1500
Supermini	6 G	130 ms	850	1500

Tab. 1.4: Výpis parametrů pro vyhodnocení bočního střetu

Třída vozu	Počáteční amplituda zrychlení	Délka trvání pulzu	Prahová hodnota integrace pro detekci incidentu
Velký rodinný vůz	3 G	90 ms	300
Velké MPV	3 G	90 ms	250
Velký Off-Road	3 G	90 ms	250
Pickup	3 G	90 ms	300
Malý rodinný vůz	3 G	90 ms	350
Malé MPV	3 G	90 ms	350
Malý Off-Road	3 G	90 ms	350
Supermini	3 G	90 ms	350

1.5.2 Načtení dat z řídicí jednotky pomocí OBD-II

K tomuto úkonu je využita CAN sběrnici. Při čtení dat z ní jsou rozlišena požadovaná data pomocí tzv. PID nacházejícího se na počátku CAN rámce. PID představuje identifikátor daného parametru, který je reprezentován 11 b za start bitem (u rozšířeného CAN 29 b). Tento identifikátor řeší hlavně prioritizaci jednotlivých zpráv. U CAN sběrnice je nízká napěťová úroveň dominantní, což má za následek, že jakmile začne více zařízení vysílat na sběrnici, tak bude upřednostněna ta zpráva, jejíž PID bude mít nejnižší hodnotu. Na sběrnici je možné setkání se čtyřmi druhy rámců [10]:

- Data frame: datová zpráva
- Remote frame: žádost o data, chybí zde bitové pole pro data, je zde PID požadované zprávy
- Error frame: chybová zpráva signalizující chyby na CAN sběrnici
- Overload frame: zpráva o přetížení, oddaluje vyslání další datové zprávy nebo žádosti o ni

2 Praktická část

Tato kapitola se zabývá samotným návrhem a procedurou vývoje zařízení zastupující funkci IVS pro eCall. První částí je zde realizace emulátoru OBD-II jednotky potřebné pro vývoj zařízení. Dále je zde popsán návrh zařízení, realizace prvotního prototypu a jeho obsluha v rámci vytvořeného experimentálního pracoviště.

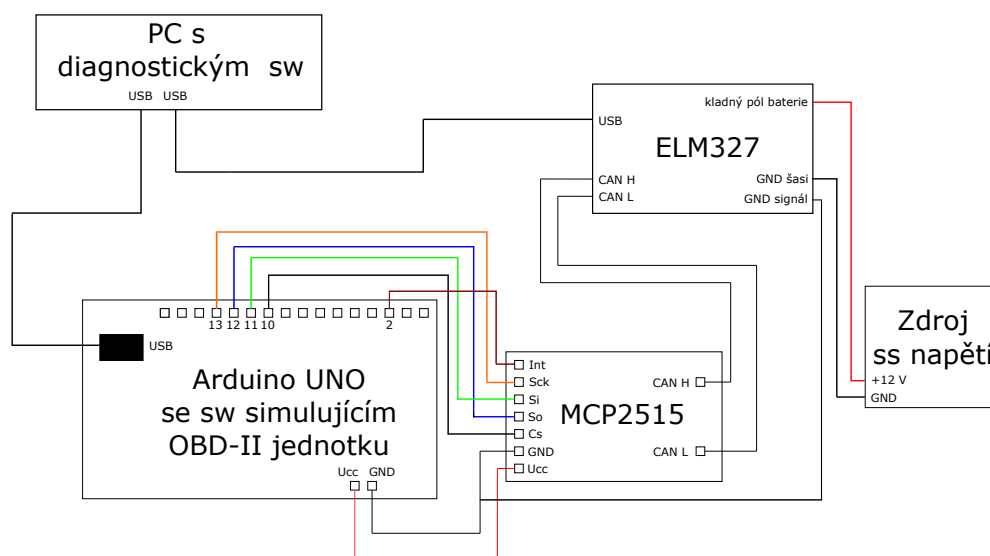
2.1 Emulace OBD-II jednotky

Kvůli výraznému usnadnění práce na zařízení bude emulována OBD-II jednotka, jelikož by se bez tohoto kroku musel všechny vývoj realizovat vevnitř vozu. Tento způsob vývoje by byl však značně nepohodlný a zdlouhavý.

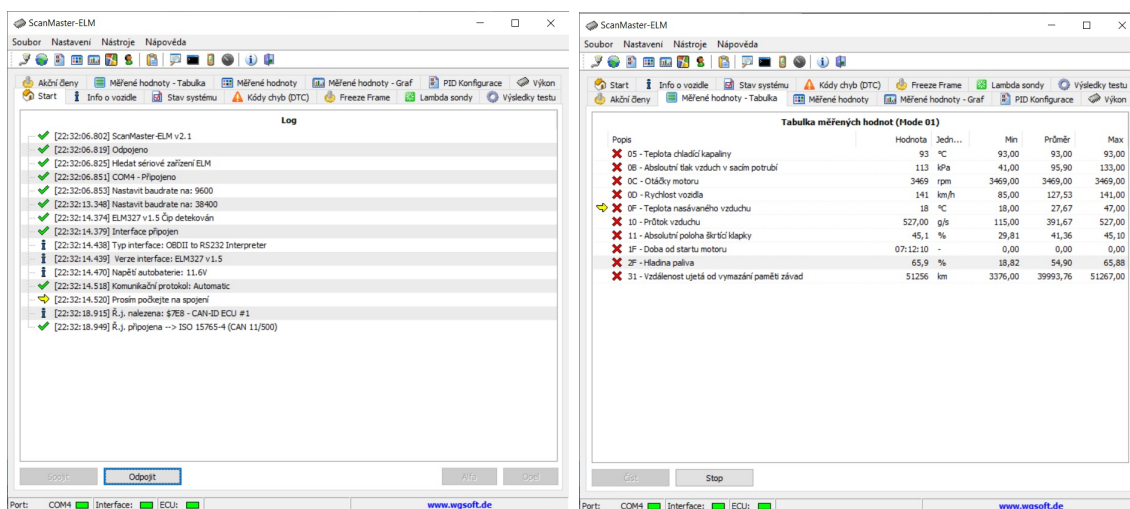
Pro realizaci emulátoru byla využita vývojová platforma Arduino UNO a její rozšiřující modul s čipem MPC2515. Tento čip funguje jako modul pro zasílání a přijímání dat po sběrnici CAN. Je složen ze tří částí. První je CAN modul obsahující implementaci CAN protokolu, provádí filtraci signálu a také obsahuje vyrovnávací paměti pro přenos a příjem. Druhou částí je blok řídicí logiky, kterým jsou řízeny všechny bloky a je zde také ovládán přenos informací mezi jednotlivými bloky a registry. Poslední je blok SPI rozhraní, pomocí kterého je obstarána komunikace s vývojovou deskou Arduino UNO. Lze provádět čtení i zápis do všech registrů modulu za použití standardních příkazů pro SPI, případně lze využít i specializované příkazy.

Čtení emulovaných dat je obstaráno diagnostikou ELM327, pro jejíž chod je nutné nasimulovat také napětí, které by indikovala v automobilu, k tomu nám slouží stejnosměrný zdroj napětí. Po zapojení celého experimentálního pracoviště dle schématu na Obr. 2.1 může být z PC emulátor ovládán pomocí grafického rozhraní programu určeného k jeho obsluze, jenž je vyobrazen na Obr. 2.3.

K obsluze diagnostiky ELM327 byl využit program ScanMaster-ELM. V programu musely být nastaveny požadavky na čtení všech PID podporovaných emulátorem v Arduinu. Na Obr. 2.2 je vidět průběh připojení k řídicí jednotce (k emulátoru) a také čtení jednotlivých parametrů z ní. Ve výpisu je vidět užitý typ protokolu (viz 1.3.1). Dále je na Obr. 2.3 zobrazeno grafické rozhraní pro obsluhu emulátoru. Program emulátoru, který byl nahrán do Arduina a také program s grafickým rozhraním, pomocí kterého je emulátor ovládán jsou vyvinuty Dhananjayem Khairnarem [15]. U kódu pro Arduino došlo k mírné modifikaci, kdy byla do kódu připsány nové parametry (typ použitého paliva, VIN). Jako poslední je zde na Obr. 2.3 vyobrazen snímek obrazovky osciloskopu, na kterém jsou viditelné signály CAN High a CAN Low, které představují část rámce. Je zde zřetelná jejich symetrie. Funkčnost komunikace přes CAN sběrnici byla analyzována za pomoci osciloskopu, především kvůli počátečním problémům s jejím zprovozněním. Nejdříve zde nebyl detekován

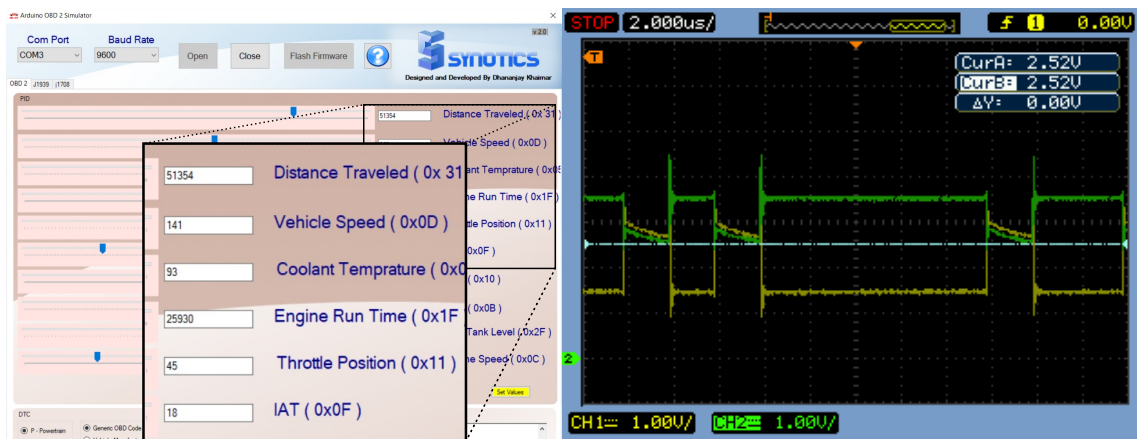


Obr. 2.1: Schéma pro zapojení emulátoru OBD-II jednotky.



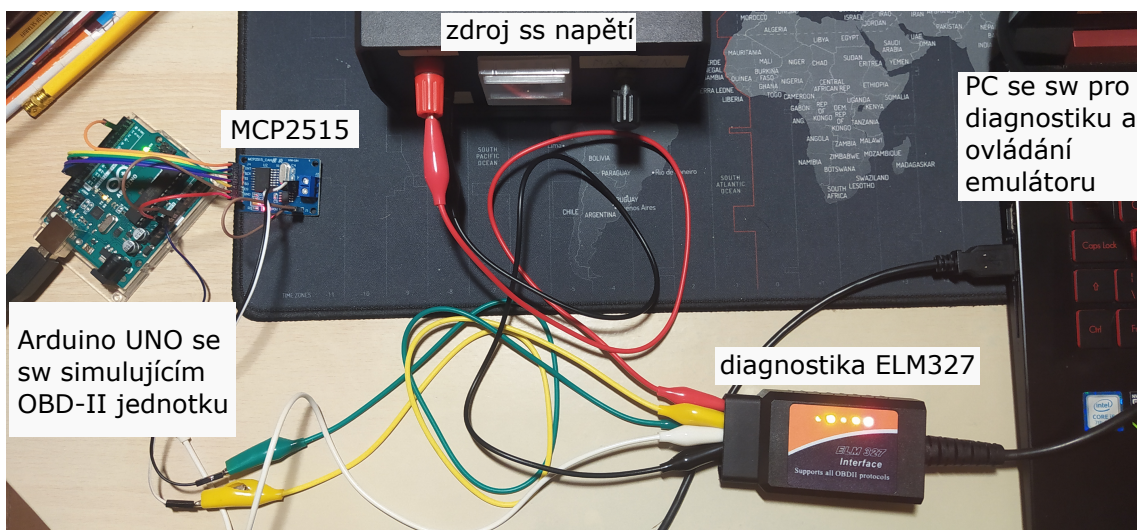
Obr. 2.2: Připojení k emulátoru (vlevo) a čtení dat z ní (vpravo) pomocí programu ScanMaster-ELM.

žádný signál, jelikož nebyla připojena diagnostika k zdroji stejnosměrného 12 V napětí, které by simulovalo napětí akumulátoru ve voze. Z tohoto důvodu diagnostika nezačala navazovat komunikaci s emulátorem, a proto se připojení k emulátoru nedařilo. Po připojení 12 V zdroje však již diagnostika navázala spojení s emulátorem, což mohlo být viděno jak ve výpisech programu ScanMasterELM, tak také na průběhu



Obr. 2.3: Grafické rozhraní programu k obsluze emulátoru (vlevo) a náhled na část CAN rámce (vpravo).

signálu zobrazeném na osciloskopu. Nakonec je zde fotografie realizace celého emulátoru viz Obr. 2.4. Na ní lze vidět zleva postupně Arduino UNO, rozšiřující desku MCP2515, zdroj stejnosměrného napětí, diagnostiku ELM327 a počítač, kterým je celé pracoviště obsluhováno.

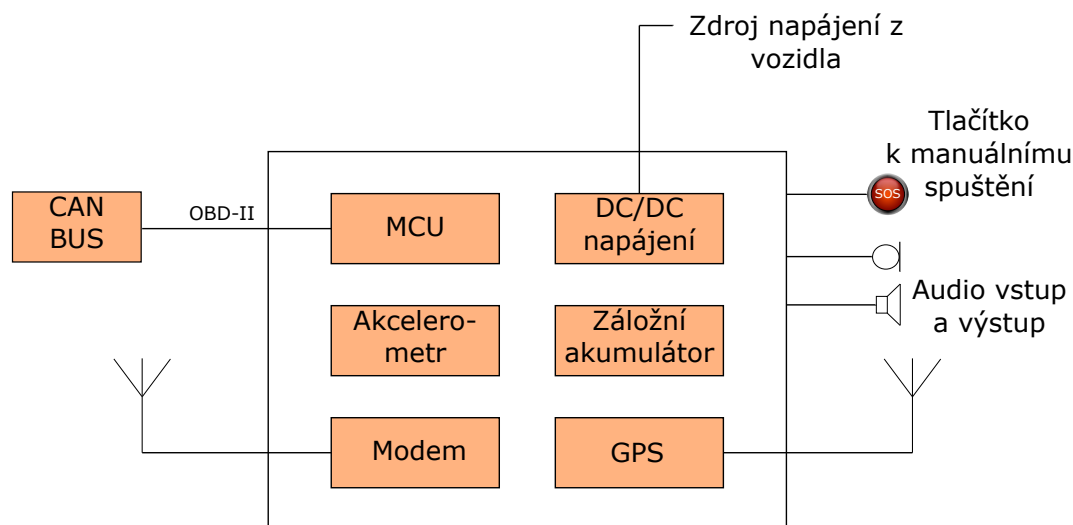


Obr. 2.4: Fotografie realizace experimentálního pracoviště.

2.2 Návrh zařízení

Při návrhu IVS zařízení pro službu eCall byla snaha o dosažení požadovaných parametrů zařízení za využití cenově přijatelných součástí. Základním požadavkem

je možnost chodu IVS nezávisle na dalších součástech automobilu. Vše by mělo být koncipováno tak, aby byla zaručena funkčnost v případě nehody. Na blokovém schématu Obr. 2.5 je vidět, které součásti musí být v IVS obsaženy.



Obr. 2.5: Blokové schéma IVS.

Požadované prvky v IVS zařízení pro službu eCall

Vzhledem k nárokům uvedeným v normách a také vzhledem k požadovaným informacím pro zprávu MSD (viz 1.4.2) je známo, že naše zařízení musí obsahovat určité součásti. Pro svou funkci bude zařízení muset komunikovat přes OBD-II s řídicí jednotkou mimo to však bude potřebná implementace dalších součástí. Pro získání údajů o poloze a směru jízdy vozidla bude nutný GPS modul. K odeslání MSD a spojení hovoru mezi tísňovou linkou a osobami ve voze bude potřeba GSM/UMTS modul. Dále je pak pro indikaci nehody nutná implementace akcelerometru. Pro manuální spuštění tísňového volání je nutná přítomnost tlačítka určeného k tomuto účelu, mimo něj také reproduktoru a mikrofону sloužících pro hovor s dispečerem tísňové linky. Během nehody může dojít i k takovému poškození vozidla, kdy dojde k přerušení napájení systému z vozu, proto je nutné, aby zařízení disponovalo záložním zdrojem (akumulátorem). IVS musí být samozřejmě také odolné vůči mechanickému poškození, aby nedošlo při nehodě k jeho zničení, čímž by ztratilo svůj význam. Jednotlivé komponenty musí také vyhovovat evropským standardům a tím zaručovat interoperabilitu [13]. Hlavním komponentem bude vývojová platforma sloužící k řízení zařízení.

2.2.1 Platforma pro řídicí část zařízení

Jako základní stavební kámen k stavbě IVS musela být zvolena platforma s implementovaným microcontroller unit (MCU), která bude fungovat jako řídicí článek celého zařízení. Tato řídicí jednotka zařízení musí jednotlivé periferie ovládat, vyčítat z nich data, vyhodnocovat tato data nebo data daným periferiím zasílat. Pro zjednodušení sestavování celého zařízení bylo voleno mezi celými vývojovými deskami, vzhledem k možnosti rychle realizovat prototyp zařízení. Vybíráno bylo mezi různými vývojovými deskami osazenými čipy různých výrobců, s přihlédnutím k obecné podpoře desek postavených na procesorech od firmy STMicroelectronics a s přihlédnutím k lepším parametrům v porovnání s dalšími vybranými deskami, jako jsou kmitočet, velikosti paměti SRAM a flash, však byla zvolena deska s tímto procesorem. Porovnání desek z pohledu několika základních parametrů je sepsáno v Tab. 2.1.

Tab. 2.1: Porovnání několika parametrů u různých typů vývojových desek

Parametr	FRDM-KL25Z	DM320119	NUCLEO-F103RB
Výrobce	NXP	Microchip	STMicroelectronics
Frekvence	48 MHz	32 kHz	72 MHz
Napájecí napětí	3,3 V	1,7–3,6 V	3,3 V, 5 V, 7–12 V
Velikost flash paměti	128 kB	8kB	128 kB
SRAM	16 kB	4kB	20 kB
Cena	387 Kč	386 Kč	298 Kč

2.2.2 Mikrofon a reproduktor

Oba komponenty by měly být situovány co nejbližší k sedadlu řidiče, jelikož u tohoto sedadla je jistá jeho obsazenost v momentě incidentu. Typickým umístěním mikrofonu je prostor blízko zpětného zrcátka, pro reproduktor pak oblast středové konzole popřípadě u řídicí páky. U nových vozů je často využíváno pro tyto účely již vestavěného handsfree, avšak naše zařízení, zabývající se tato práce, je myšleno jako plně autonomní, tudíž nepočítá s využitím systémů implementovaných v automobilu.

Mikrofon

Jako vhodný pro naše užití byl zvolen Micro Electro Mechanical Systems (MEMS) mikrofon s označením IMP23ABSU. Jeho předností je vysoká citlivost s její typickou hodnotou -38 dB. Bude také plnit stanovený požadavek na odchylky od původního signálu, kterým je odchylka od původního spektra na rozsahu 10 Hz až 10 kHz maximálně o 3 dB. Další výhody jsou například nízký odběr nebo malé rozměry. Několik parametrů je uvedeno ve výčtu níže.

- Napájecí napětí: 1,52 - 3,6 V
- Maximální odebíraný proud: 150 μ A
- Citlivost mezi 10 Hz až 10 kHz: od cca 50 Hz do 13 kHz je odchylka nulová u horní hranice je odchylka o 2 dB u dolní pak o 3 dB
- Rozsah pracovní teploty: -40 °C až +85 °C
- Odstup hluku od signálu (SNR): 64 dB(A)

Reproduktor

Pro IVS byl zvolen reproduktor s označením CES-3990-158PM-67. K jeho výběru napomohli malé rozměry, kterými jsou výška 9 mm a průměr 39 mm. Jsou zde vyvedeny dva drátky (červený pro kladný a černý pro záporný pól) zakončené konektorem k snadnému připojení. Základní parametry můžete opět vidět ve výčtu níže.

- Maximální výkon: 1,5 W
- Impedance: 8 Ω
- Maximální frekvence: 20 kHz
- Rozsah pracovní teploty: -40 °C až +70 °C

2.2.3 Akcelerometr

IVS požaduje ke své funkci 3-osý akcelerometr. Dalším potřebným parametrem je možnost měřit zrychlení větší než 6G, což je nejvyšší počáteční hodnota, která spouští proces indikace nehody. Avšak při těžkých nehodách se dostáváme k mnohem vyšším hodnotám a to až k desítkám G, tudíž pro přesnost měření a správné určení závažnosti bude vhodné použít akcelerometr, jenž dokáže vyhodnocovat i takto vysoké hodnoty. S přihlédnutím k těmto požadavkům byl vybrán akcelerometr s označením KX134-1211. Maximální měřitelné zrychlení je zde až 64G do obou směrů každé osy. Na výstupu může být očekáváno 16 datových bitů pro každou osu, což nám následně poskytuje rozlišení na $0,488 \cdot 10^{-3}G$. Porovnání parametrů s dalšími zvažovanými variantami je vypsáno níže v Tab. 2.2.

Tab. 2.2: Porovnání několika parametrů u různých typů akcelerometrů

Parametr	IIS3DWBTR	ADXL372BCCZ-RL7	KX134-1211
Výrobce	STMicroelectronics	Analog Devices	Kionix
Napájecí napětí	2,1–3,6 V	1,6–3,5 V	1,7–3,6 V
Rozsah	16G	200G	64G
Sběrnice ¹	SPI	SPI, I ² C	SPI, I ² C
Pouzdro čipu	LGA-14L	LGA-16	LGA-12
Citlivost	2049 úrovní/G	10 úrovní/G	4096 úrovní/G
Výstup	16 b	12 b	16 b
Rozměry	2,5x3 mm	3x3,25 mm	2x2 mm
Pracovní teploty	-40 °C až +105°C	-40 °C až +105°C	-40 °C až +105°C
Cena	384 Kč	324 Kč	287 Kč

2.2.4 GSM modem

Jedná se v podstatě o jednu z nejpodstatnějších komponentů, jelikož bez jeho přítomnosti by celé zařízení nebylo schopno plnit svůj účel, což je při indikaci nehody zaslání zprávy MSD a spojení hovoru s tísňovou linkou. Při výběru se ukázaly jako nejzajímavější varianty moduly řady M od firmy Quectel, u kterých dokonce byla výrobcem avizována možnost použití v zařízeních eCall. Porovnali jsme tři modely (viz Tab. 2.3) a nakonec jsme zvolili model MC90CA-04-STD, ve většině parametrů jsou na tom všechny modely stejně, avšak výhodou tohoto oproti ostatním je především již vestavěný modul GPS. I přes o něco větší cenu oproti ostatním modemům dává jeho využití smysl také z ekonomického hlediska, jelikož jsou zde integrovány dvě součásti IVS do jednoho pouzdra, což ve výsledku snižuje výslednou cenu oproti dvěma samostatným řešením. Porovnání několika jeho parametrů se samostatnými řešeními z pohledu modulu GPS můžete vidět v kapitole 2.2.5.

2.2.5 GPS modul

K získání nejpodstatnější části zprávy MSD bude muset být v IVS implementován GPS modul, kterým je poskytována informace o pozici havarovaného vozu a původním směru jeho jízdy. Přesněji by mohl být modul nazýván jako Global Navigation Satellite System (GNSS), jelikož americký systém GPS je pouze jedním z

¹Sběrnice jež podporují plnou funkcionalitu akcelerometru.

Tab. 2.3: Porovnání několika parametrů u různých typů GSM modemu

Parametr	M66FB-03-STD	M95FA-03-STD	MC90CA-04-STD
Výrobce	Quectel	Quectel	Quectel
Čtyřpásmový	ano	ano	ano
Napájecí napětí	3,3–4,6 V	3,3–4,6 V	3,3–4,6 V
Citlivost GSM	-109 dBm	-109 dBm	-109 dBm
Sběrnice	UART	UART	UART
Pouzdro čipu	LLC	LLC	LLC
Rozměry	17,7x15,8 mm	23,6x19,9 mm	25,6x15 mm
Vestavěný modul GPS	ne	ne	ano
Pracovní teploty	-40 °C až +85°C	-40 °C až +85°C	-40 °C až +85°C
Cena	149 Kč	156 Kč	282 Kč

množiny těchto systémů, dále je zde například ruský GLONASS nebo evropský Galileo. Důležitým parametrem při výběru vhodného modulu je čas, za který je schopen určit polohu, ten totiž přímo ovlivňuje i rychlost odeslání zprávy MSD. Jak bylo již uvedeno v předchozí kapitole, tak bylo zvoleno řešení, kde jsou GSM modem i GPS modul sloučeny v jednom pouzdře, v Tab. 2.4 je uvedeno několik parametrů srovnávajících toto řešení s řešeními dedikovanými.

2.2.6 Záložní akumulátor

Vzhledem k tomu, že během nehody může nastat situace, kdy bude akumulátor vozu zničen nebo následky nárazu dojde k jeho odpojení, především při čelním střetu (u některých vozů je baterie umístěna v kufru, tudíž zde při nárazu zezadu), tak je nutné zajistit funkci IVS tím, že bude vybaveno záložním akumulátorem. Dle maximálních odběrů všech součástí IVS bylo zjištěno, že odebíraný proud nepřesáhne 302 mA (reálně se bude pohybovat zřejmě výrazně níže). Vzhledem k doporučeným hodnotám napětí pro všechny součásti IVS byl zvolen jako vhodná varianta Li-Pol akumulátor s označením YJ 952040-700mAh. Jeho napětí 3,7 V je vyhovující pro všechny součásti zařízení a s kapacitou 700 mAh je schopný udržet samostatně zařízení v chodu přinejmenším 2 hodiny a 20 minut. Udržení zařízení v provozu po tuto dobu by mělo být naprosto dostačující, nejen k odeslání zprávy MSD, ale také k tomu aby operátor tísňové linky případně komunikoval s posádkou vozu, která by byla při vědomí, až do příjezdu jednotky IVS na místo.

Tab. 2.4: Porovnání několika parametrů u různých typů GPS modulů

Parametr	M10578-A2	Teseo-LIV3F	MC90CA-04-STD
Výrobce	Antenova	STMicroelectronic	Quectel
Napájecí napětí	2,8–4,3 V	2,1–4,3 V	3,3–4,6 V
Čas do získání dat při startu studeném, teplém a horkém	<35 s <25 s <1 s	<32 s <35 s <1,5 s	<35 s <30 s <1 s
Sledovací citlivost	-165 dBm	-163 dBm	-167 dBm
Sběrnice	UART	UART, I ² C	UART
Rozměry	9x9 mm	9,7x10,1 mm	25,6x15 mm
Pouzdro čipu	LGA	LCC-18	LCC
Pracovní teploty	-40 °C až +85°C	-40 °C až +85°C	-40 °C až +85°C
Cena	333 Kč	325 Kč	282 Kč

2.3 Analýza realizace dílčích součástí zařízení

V zadání práce jsou uvedeny 3 klíčové části funkcionality zařízení, z nichž je nutno zrealizovat alespoň jednu, aby bylo splněno zadání práce. Těmito dílčími součástmi zařízení jsou:

- komunikační část zajišťující volání na tísňovou linku
- zjišťování polohy z GPS přijímače
- vyhodnocení dat z akcelerometru a implementace akumulátoru

Komunikační část (spojení s tísňovou linkou)

Je potřeba navázat komunikaci mezi Nucleem a modemem. K tomu bude sloužit sběrnice UART. Pro zjednodušení testování modemu MC90 existuje vývojový kit MC90CA-TE-A (stojí 1800 Kč), který se používá v kombinaci GSMEVB kitem (1300 Kč). Na desce jsou konektory RS-232. Komunikace probíhá pomocí AT příkazů. Test zasílání příkazů do modemu lze provést např. pomocí softwaru HyperTerminal.

Tato možnost je dostatečná pro testování modemu, ve finální verzi IVS už však musí být modem kompletně obsluhován řídicí částí zařízení osazené MCU. K tomu bude potřeba vyvinout software schopný tuto obsluhu provádět.

Zjišťování polohy pomocí GPS přijímače

Při využití kitů MC90CA-TE-A a GSMEVB je možno i vyčítání polohy GPS. Komunikace je totožná jako u GSM, takže pomocí AT příkazů, dokonce je možná komunikace stejným portem, pokud bude kit nastaven do módu „All-in-one Sollution“. Test funkčnosti GNSS je možno provést například za pomoci softwaru PowerGPS.

Vyhodnocování dat z akcelerometru a implementace akumulátoru

K připojení akcelerometru je nutno použít sběrnici SPI. K jednoduššímu využití akcelerometru při vývoji by bylo vhodnější použití zkušební desky KX134-1211-EVB110. K vyhodnocování dat z akcelerometru bude potřeba vyvinout software, který bude schopen tuto činnost provádět, dle postupů udaných v kapitole 1.5. Řešení implementace akumulátoru bude vhodné až jako poslední krok, kdy budou jasné požadavky na napájení všech dílčích prvků zařízení. Následně bude možné navrhnout komplexní řešení napájení zařízení.

2.4 Realizace prototypu IVS

V rámci této práce bude provedena realizace prototypu, který sice nebude zahrnovat všechny funkcionality navrhovaného zařízení, ale bude nastiňovat podobu případného finálního zařízení. Vzhledem k náročnosti realizace a hlavně testování komunikační části zařízení není dostatečná pracovní kapacita k tomu, aby byla v této práci obsažena. Namísto toho dostanou přednost další části, které bude možno uvést do provozu v daném časovém horizontu s danou pracovní kapacitou.

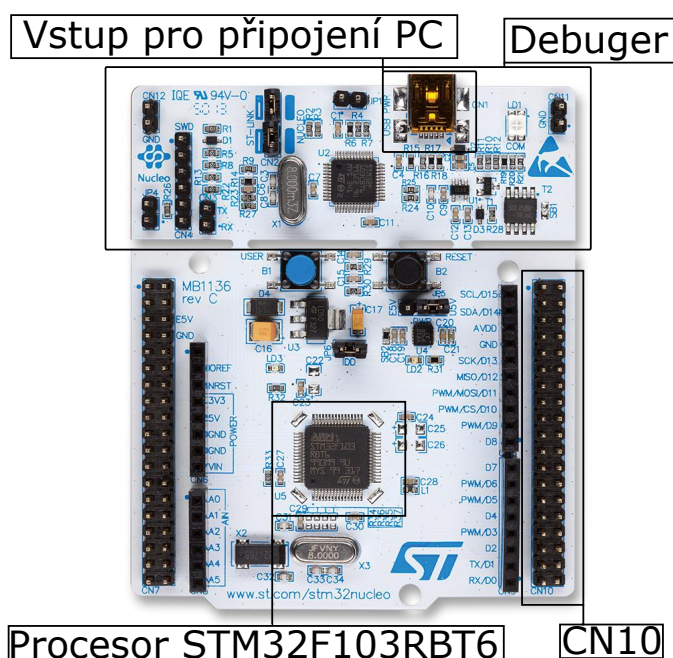
Prvním bodem, který bude řešen je navázání komunikace mezi vývojovou deskou Nucleo a již dříve zhotoveným emulátorem (případně řídicí jednotkou reálného vozu) pomocí konektoru pro OBD-II. Cílem navázání komunikace pak bude, aby bylo Nucleo schopno z řídicí jednotky vozu vyčíst data pro zprávu MSD, jako jsou číslo VIN nebo druh paliva. Dalším řešeným bodem při realizaci prototypu bude získání údajů o poloze z modulu GPS, jenž jsou také součástí dat potřebných pro kompletaci zprávy MSD. Vzhledem k tomu, že u tohoto prvotního prototypu zařízení nebude implementovaná část obstarávající komunikaci prostřednictvím GSM, nebyl použit zamýšlený modem MC90CA-04-STD, který má v sobě i funkci GPS modulu, proto je nutno použít samostatný GPS modul. K tomuto účelu byl vybrán jiný modul od výrobce Quectel a to ten s označením L80-M39.

2.4.1 Vyčítání dat z řídicí jednotky pomocí Nuclea

Pro výčet dat z řídicí jednotky vozu bylo potřeba, aby byla zprovozněna komunikace pomocí sběrnice CAN mezi řídicí jednotkou a vývojovou deskou Nucleo. Proces zprovoznění komunikace mezi vývojovou deskou a emulátorem nebo řídicí jednotkou vozu lze rozdělit na dvě základní části, a to na hardwarovou a softwarovou. Jako první bude popsána část hardwarová.

Hardwarová část

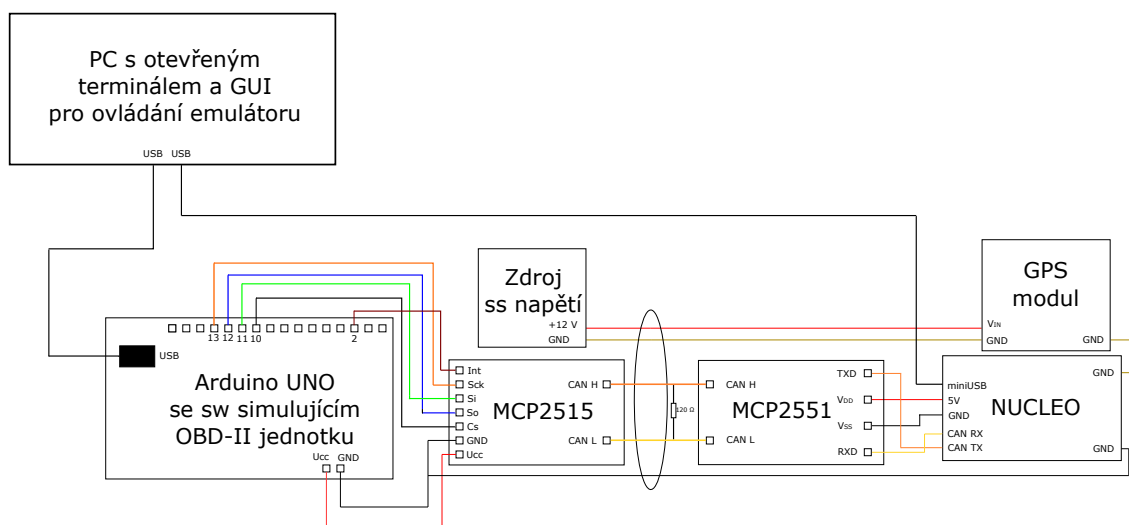
Při tvorbě hardwaru byl brán zřetel nejen na možnost otestování zařízení v laboratorních podmínkách pomocí emulátoru (viz kap. 2.1), ale také bylo pomyšleno na možnost testování zařízení při připojení k řídicí jednotce reálného vozu, z tohoto důvodu byl jako rozhraní mezi emulátorem řídicí jednotky a Nucleem zvolen konektor OBD-II. Přesněji byl konektor typu samec napojen vodiči na čip MCP2551 (viz další odstavce) a z něj na Nucleo do příslušných pinů jeho konektoru CN10, který je pravým konektorem na desce zobrazené na Obr. 2.6. Konektor pro OBD-II typu samice byl napojen na emulátor, což je rozložení, které odpovídá i v případě, kdybychom měli záměr naše zařízení testovat i na reálném voze. Napojením pinů z emulátoru do konektoru OBD-II bylo ulehčeno testování jeho funkčnosti pomocí diagnostiky v počáteční fázi vývoje, jelikož její připojení již bylo díky konektoru zjednodušeno.



Obr. 2.6: Fotografie použité vývojové desky NUCLEO-F103RB.

Dalším potřebným krokem bylo zprovoznění komunikace pomocí CAN bus mezi Nucleem a emulátorem. Stejně tak jako u Arduina, které v našem zapojení obstarává funkci emulátoru řídicí jednotky vozidla, bylo nutno u Nuclea doplnit zapojení o CAN vysílač, který nám poslouží jako rozhraní pro CAN bus. Na rozdíl od Arduina, kde byl využit čip MCP2515, zde byl využit čip MCP2551, jelikož zde nedochází k převodu z SPI na CAN, ale z UART na CAN. Z MCP2551 jsou vyvedeny dva základní vodiče s označením CAN H(igh) a CAN L(ow), které jsou známy již z dřívějších kapitol. Vzhledem k funkci CAN bylo nutno mimo přímého spojení s vodiči na straně emulátoru také vložení uzavíracího rezistoru o hodnotě 120 Ω . Pro lepší manipulaci s konektorem a možnost dobré manipulace při případném terénním testování jsou vodiče z MCP2551 vyvedeny do OBD-II konektoru kabelem UTP/F, kde jsou vodiče CAN H a CAN L vedeny v jednom krouceném páru, což je nejčastěji používaným provedením přenosového média pro tuto technologii v automobilovém průmyslu [16], ač v samotném standardu pro CAN bus (ISO-11898) žádná doporučení ohledně fyzické vrstvy nefigurují.

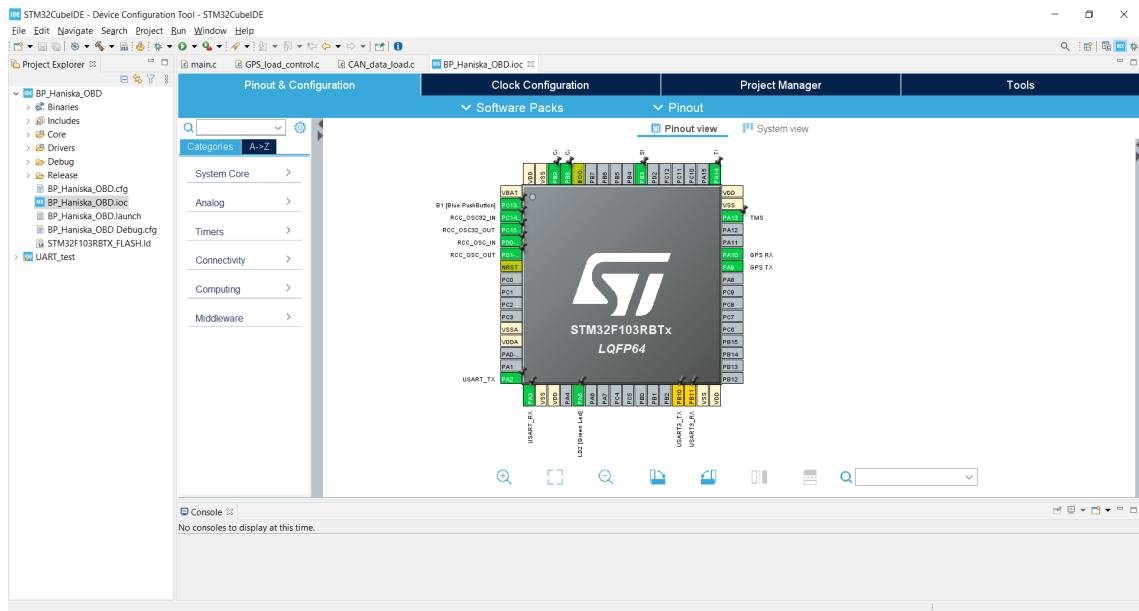
Posledním krokem k zkompletování hardwarové části komunikace mezi Nucleem a emulátorem je připojení MCP2551 do samotného Nuclea, čímž není myšleno pouze zapojení datových vodičů, ale také vodičů napájení pro čip. Zapojení celé části pro komunikaci pomocí CAN bus pak můžete vidět na Obr. 2.7.



Obr. 2.7: Zapojení části zařízení sloužící ke komunikaci s emulátorem pomocí CAN bus (bližší popis zapojení GPS modulu vizte v kap. 2.4.2).

Nejasným bodem v zapojení se může zdát zapojení vodičů pro TX a RX z Nuclea. Piny konektoru pro jejich zapojení byly zvoleny pomocí programového nástroje STM32CubeMX, který se svým grafickým rozhraním (viz Obr. 2.8) umožňuje konfi-

guraci jednotlivých pinů nebo celkovou konfiguraci jednotlivých rozhraní a zároveň je součástí vývojového prostředí STM32CubeIDE využitého pro vývoj softwaru zařízení.



Obr. 2.8: Ukázka grafického rozhraní programu STM32CubeMX (zde součást STM32CubeIDE).

Softwarová část

Jak již bylo zmíněno, tak bylo k vývoji softwaru použito vývojové prostředí s názvem STM32CubeIDE. Toto prostředí je produktem firmy STMicroelectronics, která je také výrobcem námi použité vývojové desky NUCLEO-F103RB, prostředí tedy bylo uzpůsobeno tak, aby byl uživatel s jeho pomocí schopen provádět vývoj a ladění programů i pro tuto vývojovou desku. Pomocí vývojového prostředí je možné provést všechna základní nastavení (rozhraní, přerušení, taktý apod.) pohodlně za pomoci grafického rozhraní a následně je vygenerován kód, do kterého už může uživatel rovnou zapisovat program s jeho funkcionalitou. Další výhodou tohoto vývojového prostředí jsou pak knihovny od firmy STMicroelectronics automaticky implementované po vytvoření nového projektu. Tyto knihovny obsahují mnoho užitečných funkcí, které tak nemusí uživatel vyvíjet sám a urychlí se tak práce na jeho programu.

Nejpoužívanější knihovnou při vývoji programu pro naše zařízení je knihovna *HAL*. Díky této knihovně je především možná komunikace pomocí různých rozhraní, zde se jedná o rozhraní CAN a o funkce s názvem začínajícím *HAL_CAN*, ale knihovna obsahuje také například funkci pro časovou prodlevu *HAL_Delay*.

Komunikace mezi simulátorem řídicí jednotky (nebo reálnou jednotkou) probíhá na principu dotazu a odpovědi. Prvním krokem k získání požadovaných dat je zaslání dotazu na tato data řídicí jednotce. Dotaz je zaslán jako klasická CAN zpráva s 11 b hlavičkou, podle které jsou data filtrována a tím pádem přijata nebo ne, další částí zprávy je datové pole, které obsahuje až 8 B. U zpráv určených k dotazu jsou obvykle alokovány pouze první 3 B datového prostoru (dle standardu SAE)[17]. První bajt určuje množství dodatečných data bajtů, druhý bajt určuje jeden z 10 možných módů (okamžitá data, vyčtení diagnostických chyb, čtení obecných informací o vozidle atd.), zde se obvykle setkáme s hodnotou jedna pro dotazování okamžitých hodnot (rychlost, otáčky apod.), třetí bajt nese číslo Parameter ID (PID), kterým je upřesněno, na která data (parametr, např. rychlost vozu) se dotazujeme. Pro odeslání zprávy s dotazem na daná data je tedy nutno, aby byla nastavena správná hlavička zprávy (dle SAE to je pro dotaz hodnota 7DF v hexadecimálním tvaru) a bylo provedeno nastavení datové části zprávy dle požadovaných dat. Pro zabezpečení funkční komunikace mezi řídicí jednotkou a vývojovou deskou Nucleo byl vytvořen zdrojový soubor `CAN_data_load.c` a hlavičkový soubor `CAN_data_load.h`, které obsahují funkce, jež jsou následně volány v hlavním projektovém souboru `main.c`. Nastavení hlavičky zprávy, délky datové části zprávy a dalších parametrů zprávy je provedeno pomocí funkce `CAN_MsgSetup`, do jejichž parametrů je nutno zadat délku datové části zprávy a hodnotu hlavičky. Nastavení je provedeno změnou parametrů instance `settings` (viz výpis 2.1). Před odesláním zprávy je ještě nutno doplnit datovou část zprávy pomocí funkce `CAN_TxMessage`. Zpráva je doplněna daty a umístěna do registru `TX_MAILBOX`, kde čeká na odeslání (provedeno funkcí `HAL_CAN_AddTxMessage`), které by mělo být provedeno hned, jakmile budou volné prostředky. Je provedena kontrola odeslání zprávy, zda byla odeslána nebo je v registru, což je indikováno pomocí LED osazené na vývojové desce Nucleo (funkcí `HAL_GPIO_TogglePin`). Celou funkci lze vidět ve výpisu 2.2.

Výpis 2.1: Funkce `CAN_MsgSetup` pro nastavení hlavičky a dalších parametrů CAN zprávy, která se nachází v souboru `CAN_data_load.c`.

```

1 void CAN_MsgSetup(uint32_t Length, uint32_t Identifier)
  {
2   settings.RTR = CAN_RTR_DATA;
3   settings.StdId = Identifier;
4   settings.ExtId = 0;
5   settings.IDE = CAN_ID_STD;
6   settings.DLC = Length;
7   settings.TransmitGlobalTime = DISABLE;
8 }

```

Výpis 2.2: Funkce CAN_TxMessage pro nastavení datové části zprávy a její odeslání, která se nachází v souboru CAN_data_load.c.

```
1 void CAN_TxMessage(uint8_t Data[8], CAN_HandleTypeDef
   hcan) {
2     uint32_t TxMailBox = CAN_TX_MAILBOX0;
3     HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
4     if (
5     HAL_CAN_AddTxMessage(&hcan, &settings, Data, &
       TxMailBox)
6     == HAL_OK) {
7         //Message sent to TxMailBox
8         if (
9         HAL_CAN_IsTxMessagePending(&hcan, CAN_TX_MAILBOX0))
10        {
11            HAL_Delay(2000);
12            //Still pending in TxMailBox
13            HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
14        } else {
15            //Already sent to CAN
16            HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
17            HAL_Delay(100);
18        }
19    } else {
20        printf("Request send failed\n");
21        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
22        HAL_Delay(100);
23    }
```

Po odeslání zprávy s dotazem na určitá data je nutno přijetí odpovědi, která byla za předpokladu, že je daný typ parametru podporován a jsou v pořádku i všechny hardwarové náležitosti, řídicí jednotka odeslána zpět. Jak již bylo zmíněno, tak jsou zprávy na straně příjemce filtrovány podle hlavičky, aby k tomu docházelo, tak je potřebné nastavení filtru. Odpovědi řídicí jednotky na dotaz o data jednotlivých PID přicházejí obvykle s hlavičkou o hodnotě 7E8 v hexadecimálním tvaru, avšak u některých výrobců jsou možné i jiné hlavičky, proto je nastaven filtr funkcí *CAN_SetFilterAutomotive* tak, že dojde k přijetí všech zpráv s hlavičkou v rozmezí 700 až 7FF. Po nastavení filtru je již možný samotný příjem zprávy, ten je obstarán funkcí *CAN_RxData*. Funkce zkontroluje FIFO registr určený k příjmu zpráv a

pokud se zde objeví zpráva, tak je uložena do bufferu.

Obě dříve popsané funkce, tedy funkce pro zasílání dotazů na daný parametr a pro přijetí odpovědi na tyto dotazy, jsou následně postupně použity ve funkcích *CAN_GetFuelType*, *CAN_GetVIN*, *CAN_GetSituation*. V těchto funkcích pak dále dochází ke zpracování přijatých dat a také k jejich uložení do instance storage, která je strukturou nově vytvořeného typu *CAN_DataStore*. Data jsou zde uložena po jejich zpracování již ve formě vhodné pro následný výpis do terminálu, ke kterému dochází zavoláním funkce *CAN_PrintOut*. Ve výpisu 2.3 je pak možno vidět část kódu ze souboru *main.c*, která slouží ke kompletní obsluze CAN komunikace. Postupně zde dochází k nastavení filtru pro příjem zpráv pomocí funkce *CAN_SetFilterAutomotive* a jsou načtena jednoduše data, která se v průběhu času již dále nemění, tedy druh paliva (funkce *CAN_GetFuelType*) a číslo VIN vozu (funkce *CAN_GetVIN*). Dále se dostáváme k funkcím, jež jsou už dále volány periodicky a to funkce pro získání všech dostupných proměnných parametrů (*CAN_GetSituation*), funkce pro uložení aktuálních dat do instance storage (*CAN_GetStorage*) a funkce pro výpis dat do terminálu (*CAN_PrintOut*).

Výpis 2.3: Část kódu ze souboru *main.c*, která slouží k obsluze CAN komunikace.

```
1  /* setting mask for receiving messages */
2  CAN_SetFilterAutomotive(sFilterConfig, hcan);
3
4  /* this data are not changed through the time,
5  so we can load it only once */
6  CAN_GetFuelType(hcan);
7  CAN_GetVIN(hcan);
8  .
9  .
10 .
11 /* Infinite loop */
12 /* USER CODE BEGIN WHILE */
13 while (1) {
14     printf("\n----CAN-COMMUNICATION-START----\n");
15     CAN_GetSituation(hcan);
16     storage = CAN_GetStorage();
17     CAN_PrintOut(PRINT_FUELTYPE, storage);
18     CAN_PrintOut(PRINT_VIN, storage);
19     CAN_PrintOut(PRINT_SITUATION, storage);
20     printf(
21     "-----CAN-COMMUNICATION-END-----\n\n-----GPS-CO...
```

2.4.2 Získání dat pro vývojovou desku z GPS modulu

Pro získání dat o poloze a kurzu pohybu vozidla pro kompletaci zprávy MSD je nutno tato data vyčíst z GPS modemu. K obstarání této funkcionality zařízení bylo nutno zabezpečit spojení mezi GPS modulem a vývojovou desku Nucleo. Řešení zde lze opět rozdělit na část hardwarovou a část softwarovou.

Hardwarová část

Jak již bylo zmíněno, tak pro naše zařízení byl nakonec zvolen GPS modul od firmy Quectel s označením L80-M39, protože zde není realizována část využívající GSM modul pro bezdrátovou komunikaci, tudíž nebyl potřebný multifunkční modul MC90CA-04-STD. Níže můžete vidět přehled základních parametrů použitého modulu L80-M39.

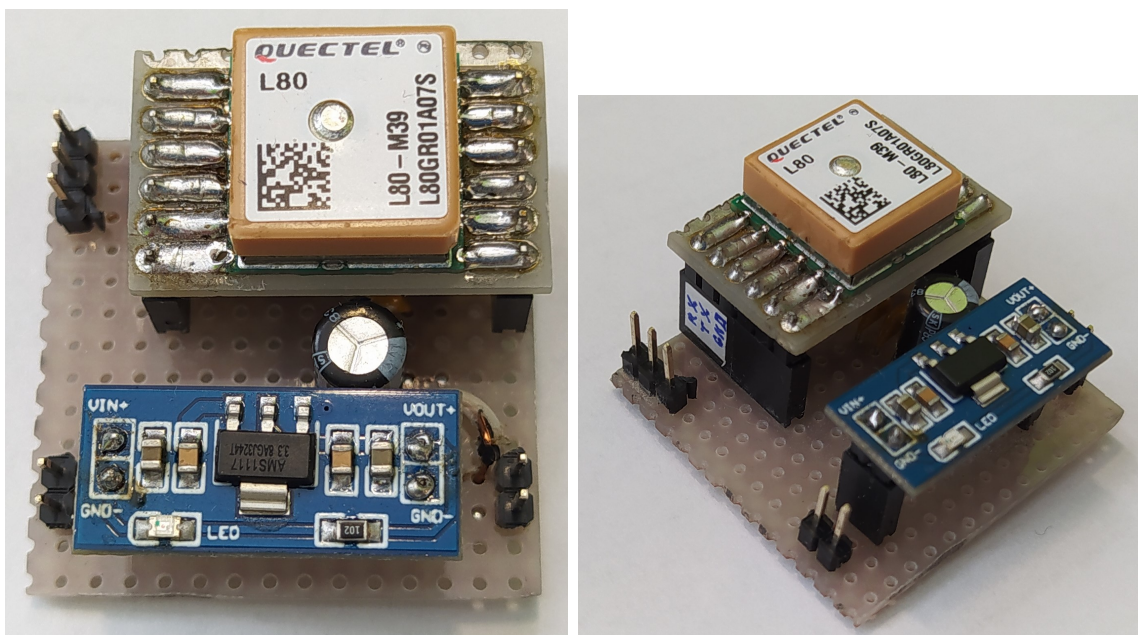
- Výrobce: Quectel
- Napájecí napětí: 2,8 až 4,3 V, typicky 3,3 V
- Čas do získání dat při startu
 - studeném: 35 s
 - teplém: 30 s
 - horkém: 1 s
- Sledovací citlivost: -168 dBm
- Sběrnice: UART
- Rozměry: 16x16x6,95 mm
- Pouzdro čipu: LLC
- Pracovní teploty: -40 °C až +85 °C
- Cena: 184 Kč

Prvním faktorem, který musel být řešen při nasazení tohoto modulu, bylo jeho napájení. Možnou variantou zde bylo napájení z vývojové desky Nucleo, jež je napájena z obslužného PC, což by nebylo problémem vzhledem k tomu, že maximální proud spotřebovávaný tímto modulem je při stavu akvizice dle výrobce typicky 25 mA a mezní výstupní proud Nucleo je 300 mA (proud požadovaný rozhraním pro CAN bus, tedy čipem MCP2551, je v maximu pouze 610 μ A). Byl však zvolen jiný způsob provedení napájení pro modul a to takový, kde je použita přímo energie z baterie vozu (připojení k ní je možné díky OBD-II konektoru, viz kapitola 1.3.3) respektive ze síťového adaptéru na 12 V, který ji na našem testovacím pracovišti supluje.

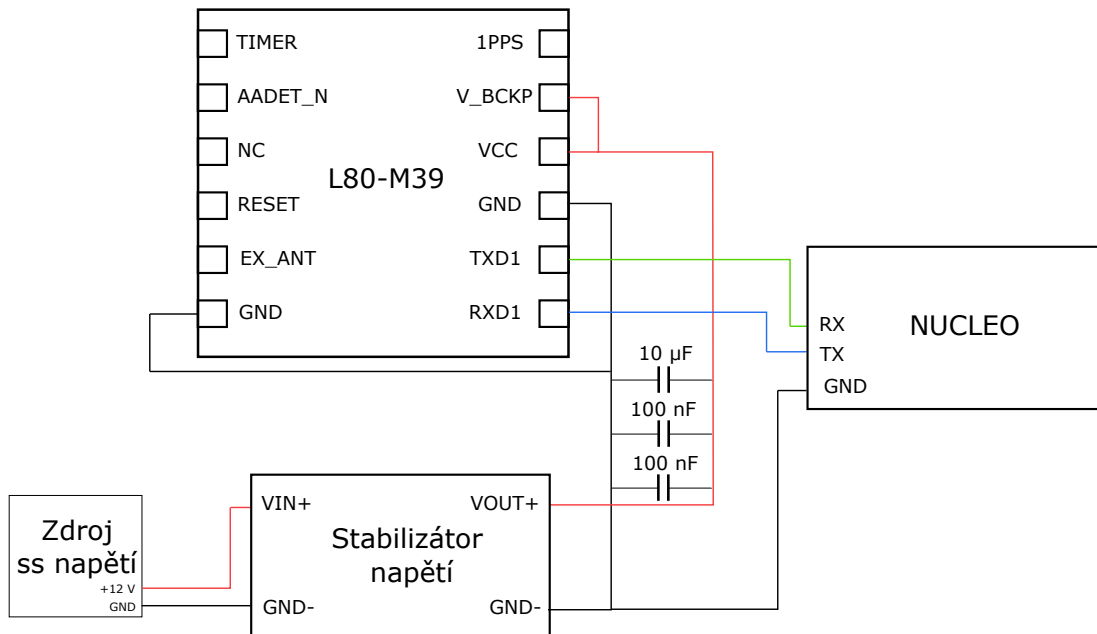
Jak bylo již dříve avizováno, tak typickou hodnotou napájecího napětí pro námi použitý modul je 3,3 V, proto bylo nutno, aby byla vstupní napěťová hladina změněna na tuto hladinu. K tomuto účelu byl do zapojení zakomponován modul osazený stabilizátorem napětí AMS1117-3.3, který je schopen přeměnit vstupní napětí 12 V na výstupních 3,3 V ideálních pro napájení našeho GPS modulu.

V momentě, kdy bylo obstaráno vhodné napájecí napětí, bylo možno přistoupit k připojení GPS modulu. Dle doporučení výrobce byly mezi póly napájení připojeny také dva blokovací kondenzátory o hodnotě 100 nF a jeden o hodnotě 10 μ F. Výstup ze stabilizátoru napětí VOUT+ byl pak připojen k pinům VCC a V_BCKP, zemnicí piny GND pak byly spojeny s výstupním konektorem GND- stabilizátoru. Pro zlepšení kompaktnosti celého zapojení došlo k jeho integraci na jednu desku. Podoba výsledné desky je vyobrazena na Obr. 2.9.

Mimo napájení bylo nutno také vyvedení kontaktů pro komunikaci pomocí rozhraní UART, k tomuto účelu slouží tři kontakty. Prvním a také nejdůležitějším je konektor TX, který je určen pro vysílání signálů, přesněji jsou jeho pomocí GPS modulem vysílány data ve formě NMEA zpráv. Dalším je konektor RX, který slouží k příjmu, pomocí něj je možno nastavovat modul zprávami definovanými v proprietárním protokolu firmy Quectel. Posledním konektorem je uzemnění GND, které slouží k vyrovnání potenciálu mezi komunikujícími stranami. Schéma celého zapojení desky s GPS modulem L80-M39 je pak možno vidět na Obr. 2.10.



Obr. 2.9: Fotografie desky osazené GPS modulem L80-M39 a stabilizátorem napětí AMS1117-3.



Obr. 2.10: Schéma zapojení desky s GPS modulem.

Softwarová část

Hlavním úkolem této části je zabezpečení schopnosti příjmu dat, která jsou GPS modulem vysílána a jejich následné zpracování. Jak již bylo zmíněno, modul komunikuje pomocí rozhraní UART, k jeho obsluze byly opět použity funkce z knihovny *HAL*, přesněji funkce *HAL_UART_Receive* a *HAL_UART_Transmit*. Všechny funkce pro obsluhu modulu jsou umístěny zvlášť v nově vytvořeném zdrojovém souboru *GPS_load_control.c* a byl také vytvořen hlavičkový soubor *GPS_load_control.h* k němu náležící. Funkce jsou opět jako u části pro obsluhu CAN bus volány v hlavním zdrojovém souboru *main.c*.

Prvním úkolem, jenž musí vývojová deska Nucleo provést, je příjem dat z modulu. Pro příjem dat za užití rozhraní UART slouží již dříve zmíněná funkce z knihovny *HAL*, *HAL_UART_Receive*, za jejíž pomoci jsou přijatá data uložena do bufferu. Přijatá data jsou ve tvaru řetězců standardu NMEA 0183 periodicky zasílána modulem (v základním nastavením je perioda rovna 1 s). Příklad řetězců můžete vidět ve výpisu 2.4. Každý začíná symbolem \$, po kterém následuje pětice velkých písmen, první dvě písmena značí typ přijímače (GPS, GLONASS, Galileo atd.), zbylá tři písmena prozrazují konkrétní typ zprávy[18].

Dále následují datová pole, jenž jsou od úvodního řetězce, ale i od sebe navzájem,

oddělena čárkami. Na samotném konci řetězce je pak hvězdička za níž následuje kontrolní součet pro určení správnosti zprávy a ukončovací znaky <CR> (návrat vozíku) a <LF>(přechod na novou řádku).

Výpis 2.4: Ukázka dat ve tvaru NMEA řetězců, jenž jsou uložena v bufferu po jejich přijetí od GPS modulu.

```
$GPRMC ,152103.000 , A ,4947.9018 , N ,01728.0236 , E ,6.54 ,
242.32 ,220421 , , , A*6A
$GPVTG ,242.32 , T , , M ,6.54 , N ,12.12 , K , A*0F
$GPGGA ,152103.000 ,4947.9018 , N ,01728.0236 , E ,
1 ,4 ,7.09 ,129.4 , M ,43.0 , M , , *5E
$GPGSA , A ,3 ,08 ,16 ,26 ,27 , , , , , , , ,7.16 ,7.09 ,1.00*03
$GPGSV ,2 ,1 ,07 ,16 ,65 ,253 ,46 ,23 ,52 ,129 ,21 ,26 ,51 ,204 ,
44 ,27 ,44 ,295 ,46*7F
$GPGSV ,2 ,2 ,07 ,10 ,33 ,167 , ,37 ,32 ,174 , ,08 ,10 ,292 ,34*4F
$GPGLL ,4947.9018 , N ,01728.0236 , E ,152103.000 , A , A*58
$GPTXT ,01 ,01 ,02 , ANTSTATUS=OPEN*2B
```

Po uložení dat do bufferu je dále nutné jejich zpracování do tvaru, jenž bude vhodný k uložení a výpisu do terminálu. Prvním krokem při zpracování je nalezení úseku surových dat uložených v bufferu, který obsahuje informace, o něž se zajímáme. Daný úsek nalezneme podle tří písmen řetězce, která určují typ zprávy. K tomu slouží funkce *GPS_FindIdx*, jenž vrací index prvního písmene hledaného výrazu v daném poli znaků (v tomto případě bufferu). Tímto indexem je dán orientační bod v bufferu, díky kterému je možno čtení dat z NMEA řetězců. Námi požadovanými daty jsou pozice (zeměpisná délka a šířka) a azimut. Pro získání pozice slouží funkce *GPS_LoadPosition* a pro získání azimutu pak funkce *GPS_LoadCourse*. Vyčtená data jsou následně uložena a případně vypsána do terminálu.

Kromě klasického režimu fungování modulu je však nutno rozeznat i jeho další stavy. Před zpracováním dat nejdříve probíhá kontrola zda vůbec nějaká data modul vysílá, tato kontrola je ošetřena pomocí logické proměnné *loadState*, která je přepisována podle toho, zda se v bufferu, promazaném předtím funkcí *GPS_ClearBufs*, nachází nějaká data. Pokud dojde k zjištění, že z nějakého důvodu nejsou přijímána data z GPS modulu, tak dochází k výpisu hlášky upozorňující na tento problém a následně k opětovné kontrole, zda nebyl problém odstraněn. Pokud problém odstraněn nebude, tak program uváže mezi těmito dvěma operacemi. Program bude opět pokračovat po vyřešení daného problému.

Po této kontrole dochází dále ke kontrole akvizičního stavu, modul totiž může vysílat i NMEA řetězce s prázdnými datovými poli, což značí nedokončený akviziční proces modulu. Tuto kontrolu provádí funkce *GPS_LoadPosition*, která vrací

logickou hodnotu NEPRAVDA, pokud při načítání dat zjistí, že jsou datová pole prázdná. Jakmile dojde k indikaci tohoto stavu, tak je následně kódem v `main.c` do terminálu vypsána informativní hláška (za předpokladu, že je splněna také podmínka vztažená k standby módu, viz další odstavec) a běh programu pokračuje dále.

Mimo získávání dat byla také zprovozněna funkce přepínání GPS modulu do standby (pohotovostního) módu. Motivací pro implementaci této funkcionality byla možnost snížení spotřeby energie modulem. Tato funkcionality má potenciál především při případné budoucí implementaci záložního bateriového napájení. Vzhledem k nižší spotřebě by bylo zařízení schopno pracovat déle v případech, kdy došlo při nehodě k vyřazení napájení z akumulátoru vozu samotného. Dle dat od výrobce je modulem při normálním módu odebírán proud 20 mA, avšak při standby módu je odebírán proud pouze 1 mA [19]. Tyto hodnoty byly ověřovány vlastním měřením, kdy byl naměřen proud 1,15 mA pro standby mód a 21,95 mA, bylo tedy zjištěno, že hodnoty uváděné výrobcem zhruba odpovídají skutečnosti.

Samotná softwarová implementace tohoto módu byla provedena za pomoci funkcí *GPS_Standby* a *GPS_WakeUp*. Funkce *GPS_Standby* přepíná modul do standby módu zasláním proprietárního příkazu „\$PMTK161,0*28“. Jakmile modul tento příkaz přijme, tak přestane vysílat NMEA řetězce přes rozhraní UART a zastaví se akviziční proces. Aktivní zůstane pouze jádro modulu a naslouchání na příjmu rozhraní UART (pin RXD1). K přepnutí modulu zpět do standardního módu pak slouží funkce *GPS_WakeUp*, ta zašle do modulu jeden znak mezerníku (" ") a tím jej opět probudí (pro probuzení modulu nemusí být zaslán pouze tento znak, ale stačí zaslání jakéhokoliv znaku).

Program vývojové desky pracuje s přepínáním GPS modulu do standby módu tak, že dojde k přepnutí až po ukončení akvizičního procesu. Nejdéle trvající akvizice modulu by měla při studeném startu trvat maximálně 35 s, proto je maximální čas, po který program čeká na její dokončení, definován touto hodnotou [19]. Pokud ani po uplynutí tohoto časového intervalu nedojde k úspěšné akvizici, tak je vypsána informativní hláška o tomto stavu a program pokračuje ve svém chodu. Tato situace může nastat typicky při testování modulu v budovách, kde dochází jejich stěnami k natolik velkému útlumu satelitního signálu, že pro něj není možné úspěšně dokončit akviziční proces. Část kódu ze souboru `main.c` sloužící k obsluze a prezentaci dat z GPS modulu je k nahlédnutí ve výpisech 2.5 a 2.6.

Výpis 2.5: Kód k obsluze GPS modulu v souboru main.c - část 1.

```

1  printf("...-----GPS-COMMUNICATION-START-----\n");
2  bool loadState = true;
3  bool acquState = false;
4  int stbcounter = 0;
5  unsigned char position[42];
6  unsigned char course[7];
7  while (loadState) {
8      GPS_WakeUp(&huart1); // wake up module
9      STBAGAIN: GPS_ClearBufs(buffer);
10     HAL_UART_Receive(&huart1, buffer, sizeof(buffer),
11     TIMEOUT);
12     loadState = GPS_Load(buffer[0]);
13     if (loadState == false) {
14         GPS_ClearBufs(position);
15         GPS_ClearBufs(course);
16         acquState = GPS_LoadPosition(position, buffer,
17         GPS_FindIdx(buffer, "GLL"));
18         if (acquState) {
19             GPS_Standby(&huart1); // back to standby mode
20             (only after successful acquisition)
21             GPS_LoadCourse(course, buffer, GPS_FindIdx(
22             buffer, "PRM"));
23             unsigned char posLab[] = "Position: ";
24             unsigned char couLab[] = "\nCourse: ";
25             HAL_UART_Transmit(&huart2, posLab, sizeof(
26             posLab),
27             SENDING_TIME);
28             HAL_UART_Transmit(&huart2, position, sizeof(
29             position),
30             SENDING_TIME);
31             HAL_UART_Transmit(&huart2, couLab, sizeof(
32             couLab),
33             SENDING_TIME);
34             HAL_UART_Transmit(&huart2, course, sizeof(
35             course),
36             SENDING_TIME);
37             printf(" deg\n");

```

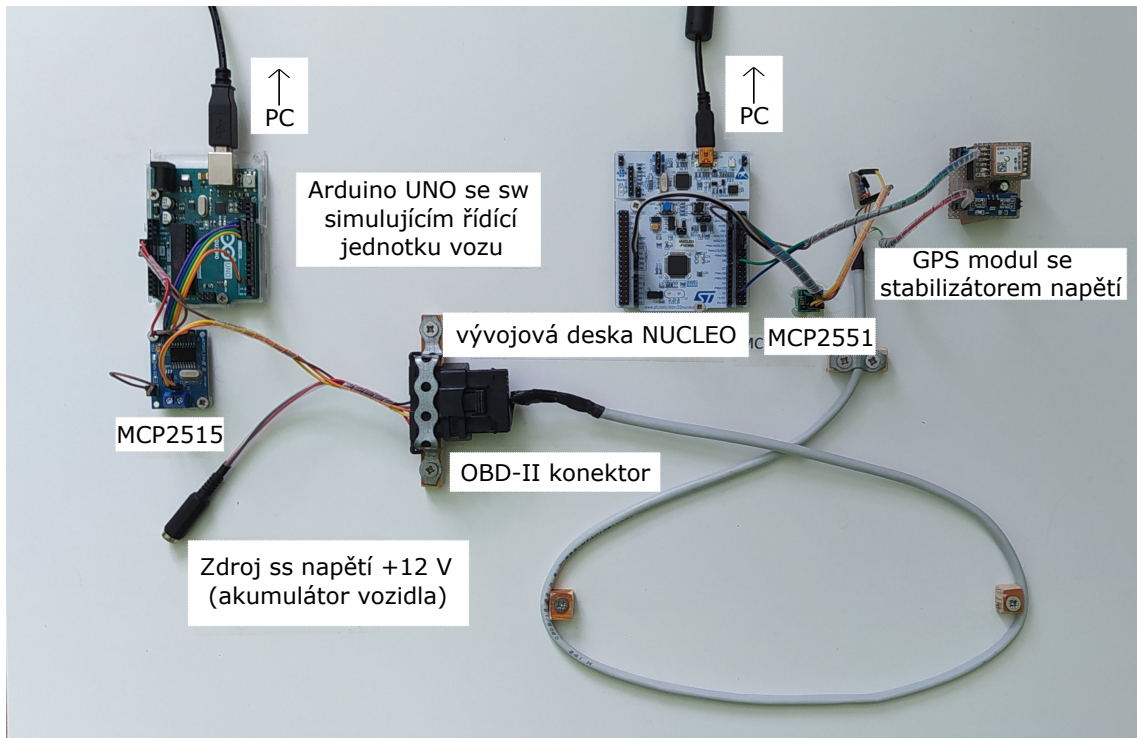
Výpis 2.6: Kód k obsluze GPS modulu v souboru main.c - část 2.

```
1      stbcounter = 0;
2      acquWait = false;
3  } else {
4      /* for case, when acquisition wasn't
5  successful because of short run time
6  since wake up from stand by mode */
7      if (acquWait) {
8          goto DNTWAIT;
9      }
10     stbcounter += 1;
11     if (stbcounter <= 14) { /* value 14 chosen
12 because of measuring of the time duration (14
13 repeats takes cca 35 seconds) */
14         HAL_Delay(1000);
15         goto STBAGAIN;
16     }
17     acquWait = true;
18     DNTWAIT: stbcounter = 0;
19     printf("GPS module connected, can't finish
20 acquisition\n");
21 }
22 } else {
23     printf("No data from GPS module\n");
24 }
25 }
26 printf("-----GPS-COMMUNICATION-END-----\n");
```

2.4.3 Finální podoba zařízení

Celé zařízení bylo usazeno na pevnou podložku, aby jej bylo možné přesouvat a pracovat s ním jako s experimentálním pracovištěm. Finální podobu zařízení (pracoviště) je možno vidět na Obr. 2.11. Pro komunikaci zařízení a PC je nutno připojení dvou kabelů do USB portů PC. První je zapojen do Arduina UNO a slouží k přenosu dat pro ovládání emulátoru z PC, která jsou vysílána obslužnou aplikací s grafickým rozhraním od Dhananjaye Khairnara, ArduinoOBDSimulator [15]. Směrem zpět do PC je zasílán log, který informuje o aktivitách emulátoru. Druhý kabel je veden z vývojové desky Nucleo, data jsou z ní zasílána do PC, kde jsou následně zobrazo-

vána v terminálu. Pro napájení GPS modulu je zde pohyblivý přívod pro síťový adaptér na 12 V stejnosměrného napětí, kterým je simulováno napětí akumulátoru reálného vozidla. Spojení mezi emulátorem a vývojovou deskou Nucleo je provedeno jako rozpojitelné s konektorem pro OBD-II, tudíž je možné i odpojení od emulátoru a připojení ke konektoru reálného vozidla.

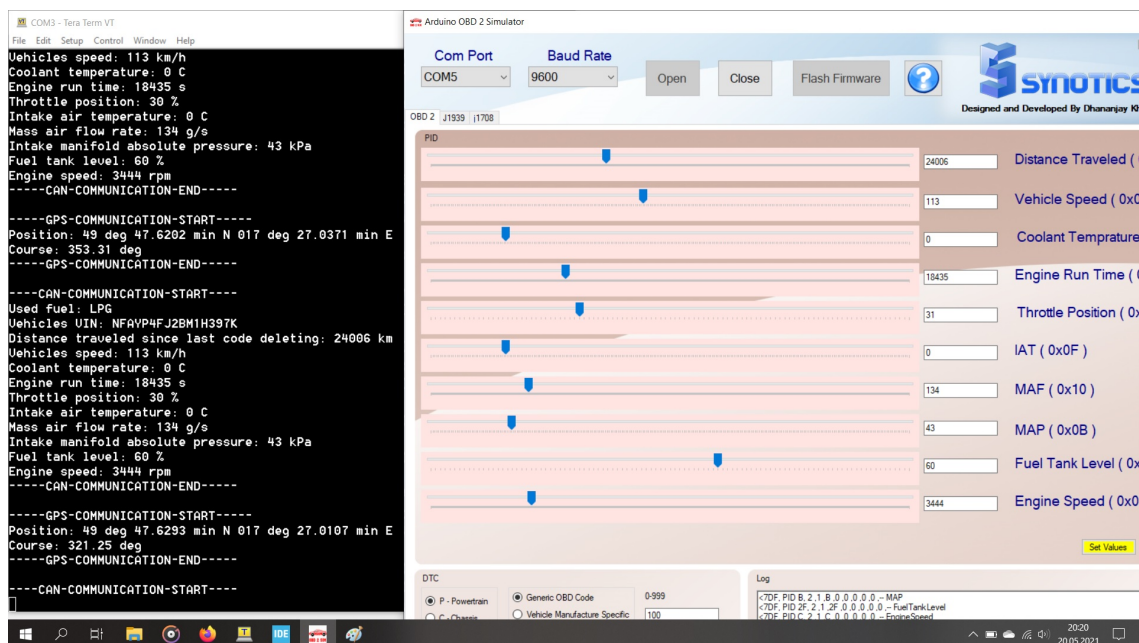


Obr. 2.11: Fotografie finální podoby zařízení po usazení na pevnou podložku.

Obsluha celého zařízení probíhá z PC, do kterého jsou připojeny pomocí rozhraní USB vývojové desky Arduino UNO (emulátor řídicí jednotky) a Nucleo (řídicí část našeho zařízení), jak již bylo popsáno dříve. Mimo fyzického připojení PC k emulátoru je nutno, aby bylo vybaveno vhodným softwarem. Kromě již dříve zmíněného programu pro ovládání emulátoru je nutný také terminálový emulátor. Při vývoji a testování zařízení byl použit volně šiřitelný program Tera Term, ale lze použít i jiné jako PuTTY, HyperTerminal a další. Na Obr. 2.12 je zobrazena obrazovka adekvátně softwarově vybaveného PC, kde je možno vlevo vidět terminál Tera Term s výstupem programu nahraného ve vývojové desce Nucleo a vpravo pak grafické rozhraní programu pro ovládání emulátoru.

Program pro ovládání emulátoru (ArduinoOBDSimulator)

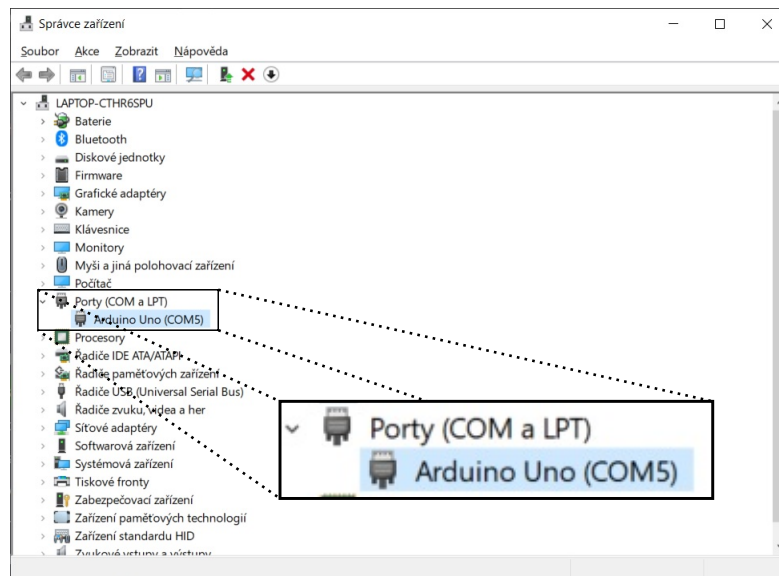
Aby mohl program plnit funkci „ovladače“ emulátoru řídicí jednotky, tak je nejdříve nutná jeho konfigurace. K tomu slouží dva rozevírací seznamy v levém horním rohu



Obr. 2.12: Snímek obrazovky PC obsluhující zařízení (vlevo Tera Term, vpravo ArduinoOBDSimulator).

okna programu s popisky „Com Port“ a „Baud Rate“. V prvním je nutno vybrat správný COM port, do kterého je připojeno Arduino UNO s nahraným programem emulátoru. Zjištění správného portu je možné v operačním systému Windows 10 pomocí systémového nástroje Správce zařízení, kde je uvedeno ke kterému COM portu patří zařízení Arduino UNO (viz Obr. 2.13). Ve vedlejší rozevíracím seznamu je volena datová rychlost přenosu, zde stačí ponechat původní hodnotu 9600 Baud, jelikož program v Arduino je na tuto rychlost nastaven, avšak za předpokladu, že bude tato rychlost stejně tak změněna i v něm, tak může dojít k přenastavení této hodnoty. Dále vpravo od tohoto rozevíracího seznamu se nacházejí tlačítka „Open“ a „Close“. Prvním dojde k navázání spojení mezi programem a Arduinem a druhým k jeho ukončení.

Po konfiguraci programu a následném spuštění programu je již možno přistoupit k samotnému ovládání emulátoru. Ovládání je obstaráno posuvníky, které náležejí jednotlivým parametrům, jež jsou vypsány vpravo od nich. Změny hodnot by se měly projevit na výstupu programu vývojové desky Nucleo v terminálu. Vpravo dole je možno pozorovat log, kde jsou zobrazena data o dění na sériové lince mezi Arduinem a PC. Pomocí logu je například možno zjistit, zda přicházejí zprávy se žádostmi o data jednotlivých parametrů a zda jsou na tyto žádosti odesílány odpovědi.

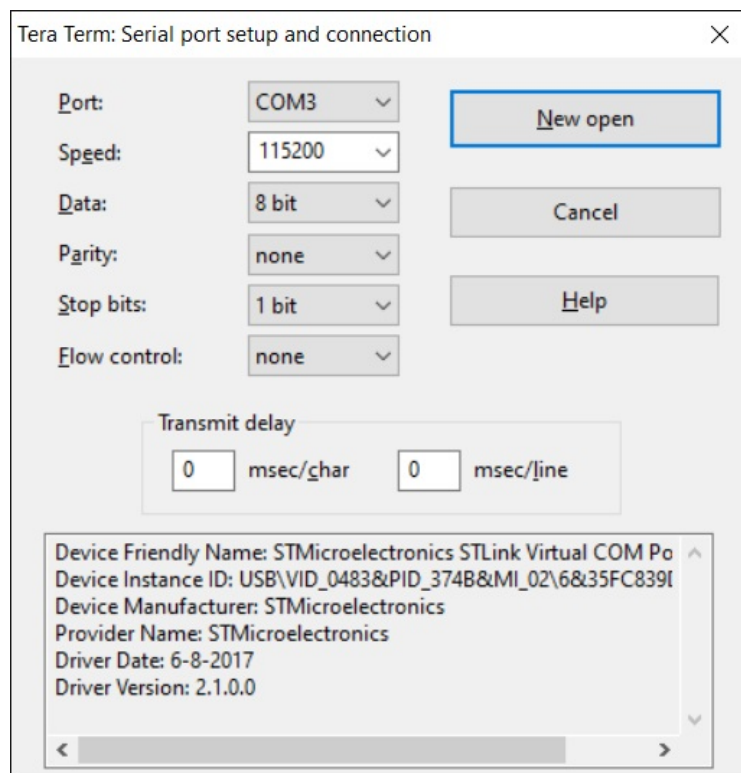


Obr. 2.13: Náhled do Správce zařízení v operačním systému Windows 10.

Tera Term

Pro použití tohoto terminálového emulátoru je stejně jako v předchozím případě potřebná nejdříve jeho konfigurace. Tento terminál má různé možnosti fungování, avšak v tomto případě je požadována pouze funkcionality terminálu pro sériovou linku. Pro nastavení sériové linky přejdeme v horní liště do kolonky „Setup“ a z rozevřacího menu zde vybereme položku „Serial port...“. V tomto momentě bylo zobrazeno konfigurační okno, které je zobrazeno na Obr. 2.14. Nachází se zde šest rozevřacích seznamů, pomocí kterých probíhá konfigurace připojení. První položkou je stejně jako u programu ArduinoOBDSimulator výběr požadovaného COM port. Na rozdíl od něj je zde ve spodní části okna možno vidět souhrn informací o daném portu, tudíž zde není potřeba využívat Správce zařízení jako v minulém případě. Další položkou je opět jako u předchozího programu přenosová rychlost, její hodnotu volíme dle nastavení komunikačního rozhraní UART zvoleného pro zasílání dat terminálu u desky Nucleo, u programu pro naše zařízení byla zvolena rychlost 115200 Baud. Dalšími položkami jsou velikost dat, parita, počet stop bitů a řízení toku. U našeho zařízení jsou všechna tato data nastavena shodně s výchozím nastavením v tomto konfiguračním okně, tudíž není potřeba je měnit, stejně tak není potřeba měnit hodnoty zpoždění při vysílání, které se nacházejí pod rozevřacími seznamy.

Po nastavení hodnot je možné spustit terminál a navázat spojení s vývojovou deskou Nucleo stisknutím tlačítka „New open“ nebo také celý proces zrušit tlačítkem „Cancel“. Po spuštění terminálu by mělo dojít k zobrazování výpisu programu běžící-



Obr. 2.14: Konfigurační okno sériové linky v programu Tera Term.

cího v Nucleu. Výpisy je možno kopírovat z terminálu nebo pomocí funkce „Log...“ vytvořit soubor do nějž budou všechna data průběžně ukládána dokud nedojde k přerušení zápisu uživatelem.

Výstup programu vývojové desky Nucleo

Výstupem programu vývojové desky Nucleo je přehledný výpis s daným formátem v terminálu obslužného PC, pomocí nějž jsou uživatelům sdělovány získaná data nebo případně detekované problémy při práci programu. Výpis se skládá ze dvou hlavních částí, jeho názorný příklad je možno vidět ve výpisu 2.7.

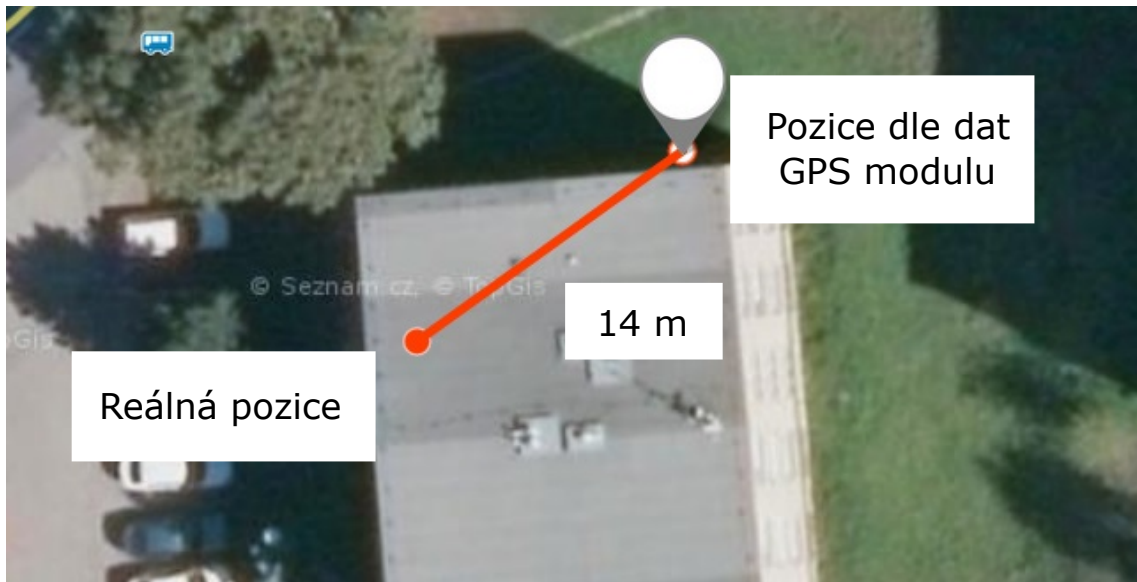
Výpis 2.7: Příklad výstupu programu nahraného ve vývojové desce Nucleo do terminálu v PC.

```
-----CAN-COMMUNICATION-START-----  
  
Used fuel: LPG  
Vehicles VIN: NFAYP4FJ2BM1H397K  
Distance traveled since last code deleting: 43758 km  
Vehicles speed: 110 km/h  
Coolant temperature: 118 C  
Engine run time: 46594 s  
Throttle position: 50 %  
Intake air temperature: 44 C  
Mass air flow rate: 204 g/s  
Intake manifold absolute pressure: 115 kPa  
Fuel tank level: 52 %  
Engine speed: 5672 rpm  
-----CAN-COMMUNICATION-END-----  
  
-----GPS-COMMUNICATION-START-----  
Position: 49 deg 47.6183 min N 017 deg 27.0217 min E  
Course: 70.650 deg  
-----GPS-COMMUNICATION-END-----
```

První část výpisu je zaměřena na data získaná z emulátoru řídicí jednotky za užití rozhraní CAN bus. Nejdříve jsou vypsány jednou dotazované neměnné parametry, tedy typ používaného paliva a číslo VIN. Následně jsou prezentována data proměnných parametrů, jako například rychlost vozidla (vehicles speed), pozice pedálu (throttle position), hladina paliva v nádrži (fuel tank level) a další.

Po dokončení části zabývající se daty získanými z emulátoru program přechází ke komunikaci s GPS modulem. Výstup se zde může být značně lišit, jelikož se odvíjí od aktuálního stavu modulu. V případě, že modul není připojen, ať z důvodu fyzického přerušení datového spojení nebo kvůli absenci jeho napájení, tak bude programem v tomto bodě periodicky vypisována výstražná hláška a dojde k uváznutí v tomto bodě. K opětovnému pokračování běhu programu je nutno daný problém napravit. Další možností je, že je sice komunikace s modulem navázána správně, ale ten není schopen dokončit akvizici. V tomto případě je pouze vypsána hláška, která na tento stav upozorňuje, ale program může dále pokračovat v chodu. Posledním a nejžádanějším stavem je ten, kdy dojde ke sdělení dat o poloze zařízení a směru pohybu. Tedy stav, při kterém probíhá správně ke komunikaci s GPS modulem a

došlo také k úspěšnému dokončení akvizičního procesu. Tento případ je zobrazen ve výpise 2.7. Data jsou vypsána ve tvaru, kdy jsou stupně zeměpisné délky či šířky udávány jako celá čísla a minuty jako desetinná čísla se čtyřmi desetinnými místy. Na příkladě ukázkového výpisu lze ověřit validitu získaných dat. Kontrolu dat lze provést vynesemím udané souřadnice do mapy a porovnáním jí se souřadnicí reálné pozice GPS modulu. Toto srovnání je vyobrazeno na Obr. 2.15.



Obr. 2.15: Mapa zobrazující přesnost získaných dat.

Po vynesení souřadnic do mapy lze konstatovat, že data udávající pozici zařízení mají od reálné polohy zařízení odchylku zhruba 14 m. Takováto přesnost určení pozice je vcelku uspokojivá, avšak například na dálnici by v případě nehody při této přesnosti nemuselo být jasné, v kterém jízdním směru k ní došlo, proto je také důležitý azimut, který udává směr pohybu modulu. Přesnější hodnota reálné orientace modulu při tomto testu nebyla měřena a zaznačena, ale azimut v ukázkovém výpise 2.7 je zhruba shodný s realitou v daném momentě.

Závěr

Práce zkoumala možnost dovybavování starších vozů, které nebyly z výroby vybaveny zařízením IVS pro službu eCall, zařízením, které by jej bylo schopno zastoupit. V úvodu práce byl vytvořen přehled zařízení implementujících do vozů funkci tísňového volání, ve kterém bylo postupně porovnáno několik konkurenčních zařízení. Bylo zjištěno, že ani jedno z nich není schopno poskytnout plnou funkcionalitu IVS, dle všech požadavků pro službu eCall. Na rozdíl od nich však zařízení, které bylo navrženo v rámci této práce, všechny tyto požadavky splňuje, což jej činí potenciálně velmi zajímavým pro trh.

K vývoji zařízení byl zhotoven emulátor, který umožnil simulaci řídicí jednotky automobilu, díky čemu nebylo nutno během vývoje testovat zařízení na reálném vozidle, ale postačovalo statické pracoviště. Tento emulátor byl vytvořen za užití hardwaru sestávajícího z vývojové desky Arduino UNO a jeho rozšiřující desky s čipem MCP2515, který poskytuje možnost komunikace s Arduinem pomocí sběrnice CAN. Po sestavení emulátoru byl do Arduina nahrán software od Dhananjaye Khairnara, který byl modifikován přidáním dvou dalších parametrů, na které je možno se emulátoru dotázat. Od stejného autora byl využit i software s grafickým rozhraním pro ovládání emulátoru. Funkčnost emulátoru byla ověřena za pomoci diagnostiky ELM327 a programu ScanMasterELM.

Dále byl sestaven také návrh budoucího zařízení IVS pro eCall. Bylo vytvořeno blokové schéma zařízení, kde je popsáno, které součásti by zařízení mělo obsahovat. Ke každé součásti byl následně vybrán konkrétní typ komponentu, který vzešel jako nejvhodnější po porovnání s dalšími možnými alternativami.

Po zpracování návrhu zařízení došlo k realizaci prvotního prototypu zařízení. U tohoto prototypu došlo k zaměření na zprovoznění dvou funkcionalit uvažovaného zařízení, a to na komunikaci s vozem (emulátorem) pomocí CAN bus a získávání dat o pozici zařízení a směru jeho pohybu za pomoci GPS modulu. Jako základní komponent pro vyčítání, zpracování a následnou prezentaci dat, byla zvolena vývojová deska NUCLEO-F103RB, pro kterou byl vyvinut software určený k těmto účelům. Vývoj softwaru proběhl za pomoci vývojového prostředí STM32CubeIDE a knihoven HAL od výrobce vývojové desky, firmy STMicroelectronics, v programovacím jazyce C.

Výsledný prototyp je dle určených cílů díky vyvinutému softwaru schopen komunikace s emulátorem řídicí jednotky, načtení dat z něj, jejich zpracování a následné prezentace prostřednictvím terminálového rozhraní v obslužném PC. Také byla realizována schopnost komunikace s GPS modulem a rovněž zpracování dat z něj a jejich prezentace. Mimo to byla také implementována funkce přepínání GPS modulu do pohotovostního režimu pro snížení jeho spotřeby energie. Výsledná verze zařízení

byla upevněna na pevnou podložku a popsána vysvětlujícími popisky, čímž bylo uzpůsobeno pro další experimentální činnost.

Prototyp vytvořený v rámci této práce je nastíněním podoby dílčích částí případného budoucího plně funkčního zařízení. V případě úplného dokončení vývoje zařízení a implementace zbývajících částí potřebných u IVS pro službu eCall, by se mohlo jednat o zajímavou alternativu pro vozidla tímto systémem nevybavená.

Literatura

- [1] Český statistický úřad: Nehody v silniční dopravě. 2020, [online]. [cit. 5.12.2020].
URL https://www.czso.cz/documents/10180/122586371/nedcr100520_2.xlsx/39ef3dff-64db-442e-8fda-7b4ee8b97f07?version=1.1
- [2] Evropský parlament: Tisková zpráva. 2015, [online]. [cit. 20.11.2020].
URL <https://www.europarl.europa.eu/news/cs/press-room/20150424IPR45714/zarizeni-automatickeho-tisnoveho-volani-ve-vsech-novych-typech-aut-od-jara-2018?quizBaseUrl=https%3A%2F%2Fquizweb.europarl.europa.eu>
- [3] European Automobile Manufacturers Association: Average vehicle age. 2018, [online]. [cit. 5.12.2020].
URL <https://www.acea.be/statistics/tag/category/average-vehicle-age>
- [4] Ferrière, F.: Fotografie zařízení Michelin Safe & Drive. 2017, [online]. [cit. 21.11.2020].
URL <https://www.caradisiac.com/michelin-safe-drive-un-systeme-pour-alerter-les-secours-en-cas-d-accident-163166.htm>
- [5] Bosch Connected Devices and Solutions: Fotografie zařízení Bosch Telemetric eCall Plug (TEP). [online]. [cit. 21.11.2020].
URL https://www.bosch-connectivity.com/media/product_detail_tep/handy-stecker-16x9_en_res_1280x720.jpg
- [6] T-Mobile Czech Republic a.s.: Fotografie zařízení T-Mobile Chytré auto. [online]. [cit. 21.11.2020].
URL <https://www.t-mobile.cz/dcpublic/chytre-auto-obd-port.png>
- [7] PACE Telematics GmbH: Fotografie zařízení PACE. [online]. [cit. 21.11.2020].
URL <https://www.pace.car/en>
- [8] SECONS s.r.o.: OBD-II Protocols. [online]. [cit. 20.11.2020].
URL http://www.obdtester.com/obd2_protocols
- [9] McCord, K.: *Automotive Diagnostic Systems: Understanding OBD I and OBD II*. CarTech Inc, 2011.
- [10] CANLAB s.r.o.: CAN - Controller Area Network. [online]. [cit. 20.11.2020].
URL http://www.canlab.cz/cs/can_bus

- [11] Hadjidimitriou, S. N.; Öörni, R.: A Framework for Appraising European Member States' Readiness Level for eCall Deployment. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, s. 2299–2304, doi:10.1109/ITSC.2015.371.
- [12] Mushthalib, M. A.; Mansor, H.; Abidin, Z. Z.: Development of eCall for Malaysia's Automotive Industries. In *2019 7th International Conference on Mechatronics Engineering (ICOM)*, 2019, s. 1–5, doi:10.1109/ICOM47790.2019.8952044.
- [13] Fischer, F.: Procedures for eCall certification and execution. 2015.
- [14] Rybnik, M. A.; Omieljanowicz, M. M.; Tabedzki, M. M. S.; aj.: In-Car eCall Device for Automatic Accident Detection, Passengers Counting and Alarming. *Transactions on Computational Science XXXV: Special Issue on Signal Processing and Security in Distributed Systems*, ročník 11960, 2020: str. 36.
- [15] Khairnar, D.: Repozitář s použitým programem pro emulátor řídicí jednotky vozu. 2020, [online]. [cit. 11.12.2020].
URL <https://github.com/8-DK/ArduECUSimulator>
- [16] Juexiao, C.; Minjie, G.; Feng, L.; aj.: EMC comparison of unshielded twisted pair and shielded twist pair in automotive CAN-bus. In *2009 IEEE Intelligent Vehicles Symposium*, 2009, s. 1063–1066, doi:10.1109/IVS.2009.5164428.
- [17] Microchip Technology Inc.: Introduction to Controller Area Network (CAN). [online]. [cit. 17.5.2021].
URL https://www.microchip.com/stellent/groups/SiteComm_sg/documents/DeviceDoc/en558265.pdf
- [18] Langley, R.: NMEA 0183: A GPS receiver. *GPS world*, 1995.
- [19] Quectel Wireless Solutions Co., Ltd.: L80 Hardware Design. [online]. [cit. 18.5.2021].
URL https://www.quectel.com/download/quectel_l80_hardware_design_v1-5/

Seznam symbolů, veličin a zkratek

OBD	palubní diagnostika – On Board Diagnostics
SAE	společnost automobilních inženýrů – Society of automotive engineers
PSAP	veřejný bezpečnostní bod odpovědi – public safety answering point
IZS	integrovanná záchranná služba
MSD	minimální soubor dat – minimum set of data
VIN	Vehicle identification number – Identifikační číslo vozidla
HeERO	Harmonised eCall European Pilot
SP ID	identifikátor poskytovatele služeb – Service Provider Identification
GPS	globální poziční systém – Global Positioning System
GNSS	globální navigační satelitní systém – Global Navigation Satellite System
IVS	systém zabudovaný ve voze – In Vehicle System
MCU	jednočipový počítač – microcontroller unit
Euro NCAP	Evropský program pro posuzování nových automobilů – European New Car Assessment Programme
MEMS	mikroelektronický mechanický systém – Micro Electro Mechanical Systems
PID	identifikátor parametru – Parameter Identifier
NMEA	Národní sdružení pro loďní elektroniku – National Marine Electronic Association

Seznam příloh

A	Obsah přílohy	62
B	Návod k obsluze vytvořeného pracoviště	63
B.1	Zapojení pracoviště	63
B.1.1	Datové kabely	64
B.1.2	Napájení	64
B.2	Konfigurace obslužného PC	64
B.2.1	Nastavení terminálového emulátoru Tera Term	65
B.2.2	Nastavení programu ArduinoOBDSimulator	65
B.3	Ovládání pracoviště z obslužného PC	66

A Obsah přílohy

V příloze se nachází archiv s celým projektem programu pro použitou vývojovou desku NUCLEO-F103RB. Projekt byl vytvořen za pomoci vývojového prostředí STM32CubeIDE ve verzi 1.6.1. Struktura projektu je vyobrazena níže.

```
BP_Haniska_OBD ..... hlavní složka projektu
├── .settings ..... konfigurační soubory pro STM32CubeIDE
├── Core ..... jádro projektu
│   ├── Inc ..... hlavičkové soubory včetně vytvořeného kódu
│   │   ├── CAN_data_load.h
│   │   ├── GPS_load_control.h
│   │   ├── main.h
│   │   ├── stm32f1xx_hal_conf.h
│   │   └── stm32f1xx_it.h
│   ├── Src ..... zdrojové soubory včetně vytvořeného kódu
│   │   ├── CAN_data_load.c
│   │   ├── GPS_load_control.c
│   │   ├── main.c
│   │   ├── stm32f1xx_hal_msp.c
│   │   ├── stm32f1xx_it.c
│   │   ├── syscalls.c
│   │   ├── systemem.c
│   │   └── system_stm32f1xx.c
│   └── Startup
├── Drivers
│   ├── CMSIS ..... složka knihoven vztažených k vývojové desce
│   └── STM32F1xx_HAL_Driver ..... složka knihoven HAL
├── .cproject ..... soubor pro import projektu do STM32CubeIDE
├── .mxproject ..... konfigurační soubor pro STM32CubeMX
├── .project ..... soubor pro import projektu do STM32CubeIDE
├── BP_Haniska_OBD_Debug.cfg
├── BP_Haniska_OBD.cfg
├── BP_Haniska_OBD.ioc ..... soubor grafického rozhraní konfigurace desky
├── BP_Haniska_OBD.launch
├── README.md ..... stručný popis projektu pro GitLab
└── STM32F103RBTX_FLASH.ld
```

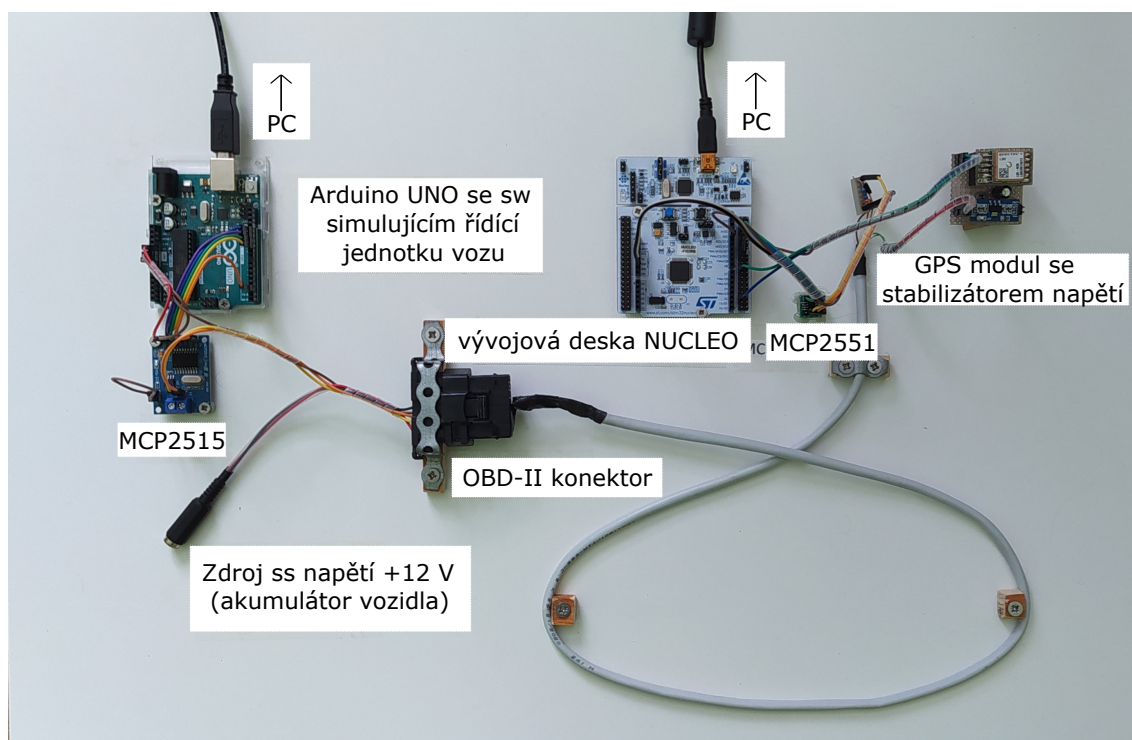
B Návod k obsluze vytvořeného pracoviště

Tento návod popisuje, jakým způsobem se uvádí do provozu experimentální pracoviště vytvořené v rámci této práce. Celá procedura zprovoznění pracoviště je rozdělena do tří hlavních částí, těmito částmi jsou:

- zapojení pracoviště
- konfigurace obslužného PC
- ovládání pracoviště z obslužného PC

B.1 Zapojení pracoviště

Pro zapojení pracoviště je nutno zapojení dvou datových kabelů pomocí rozhraní USB s obslužným PC. Mimo tyto kabely je nutné připojení napájení, které je použito pro modul GPS.



Obr. B.1: Fotografie zařízení usazeného na pevnou podložku.

Návod řeší situaci, kdy je pro práci s prototypem zařízení zastupujícím IVS pro službu eCall, využit emulátor. Mimo tuto variantu je však také možné odpojení OBD-II konektoru, tím i odpojení emulátoru a následné zapojení do OBD-II konektoru reálného vozidla. V tomto případě není nutno vykonávat procedury pro zprovoznění emulátoru zde uvedené.

B.1.1 Datové kabely

Připojení těchto kabelů je důležité pro přenos dat mezi obslužným PC a vývojovými deskami Arduino UNO a NUCLEO-F103RB. Pro obě desky je prostřednictvím těchto kabelů také dodáváno napájení.

Kabel z Arduina UNO

Kabel je ve vývojové desce zapojen pomocí konektoru UBS-B (na Obr. B.1 levý označený vývod do PC). Druhý konec kabelu je opatřen konektorem typu USB-A, který je nutno zapojit do volného portu v obslužném PC. Tento kabel umožňuje komunikaci mezi obslužným PC a programem emulátoru řídicí jednotky vozidla, což umožňuje jeho ovládání.

Kabel z NUCLEA-F103RB

Je zapojen ve vývojové desce pomocí konektoru miniUSB (na Obr. B.1 pravý označený vývod do PC). Druhý konec kabelu je taktéž opatřen konektorem typu USB-A a je nutno zapojit do volného portu obslužného PC. Kabel umožňuje komunikaci mezi obslužným PC a programem ve vývojové desce NUCLEO, což umožňuje prezentaci zpracovaných dat v okně terminálu PC.

B.1.2 Napájení

Je obstaráno připojením síťového adaptéru na 12 V stejnosměrného napětí. Adaptér je připojen pomocí sousedního napájecího konektoru 5,5/2,1 mm, který je možno vidět na Obr. B.1 vlevo dole. Pomocí něj je simulováno napětí 12 V na příslušném pinu OBD-II konektoru, na kterém by po připojení k tomuto konektoru v reálném voze bylo toto napětí přivedeno z akumulátoru vozidla. U tohoto zařízení je tento zdroj stejnosměrného napětí využit k napájení GPS modulu.

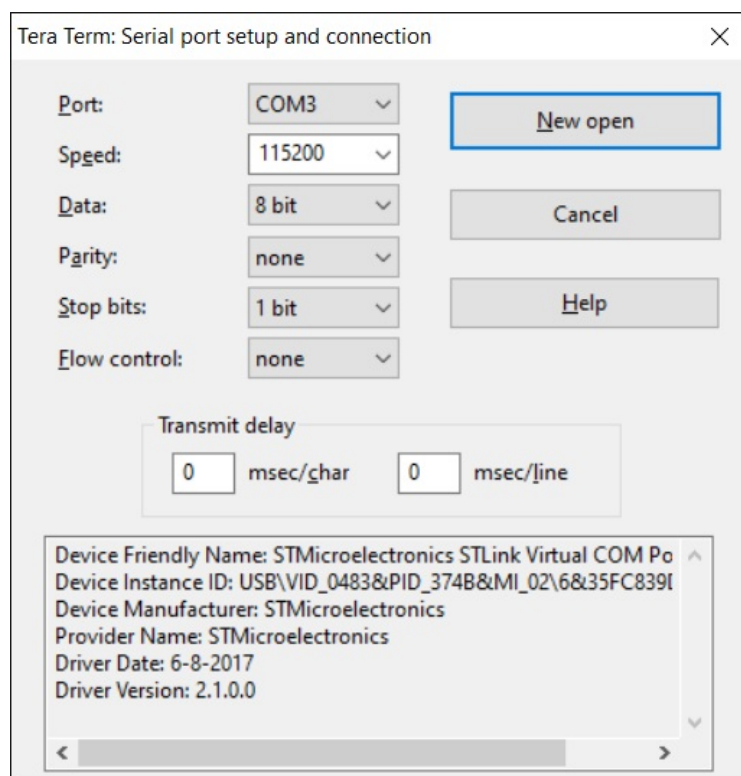
B.2 Konfigurace obslužného PC

Aby byla konfigurace obslužného PC možná, tak je nutno do něj nainstalovat určitý software. Nutným softwarem je terminálový emulátor a program ArduinoOBDSimulator (dostupný pouze pro operační systémy Windows). Jako terminálový emulátor můžou být použity různé programy, jako PuTTY, HyperTerminal a další, avšak v tomto návodu je popsána konfigurace v programu Tera Term.

B.2.1 Nastavení terminálového emulátoru Tera Term

Je nutno zde zvolit a nastavit danou sériovou linku, která náleží připojené vývojové desce NUCLEO-F103RB. Nastavení probíhá následným způsobem:

1. Spuštění programu Tera Term.
2. V horní liště klikneme na rozevírací menu „Setup“.
3. V otevřeném menu klikneme na položku „Serial port...“.
4. Otevřelo se konfigurační okno, kde v položce port vybereme COM port, který náleží vývojové desce NUCLEO-F103RB, k čemuž napomáhají informace o portu ve spodní části konfiguračního okna.
5. Provedeme nastavení všech ostatních položek dle hodnot na Obr. B.2.
6. Po nastavení všech položek navážeme spojení s vývojovou deskou kliknutím na tlačítko „New open“.

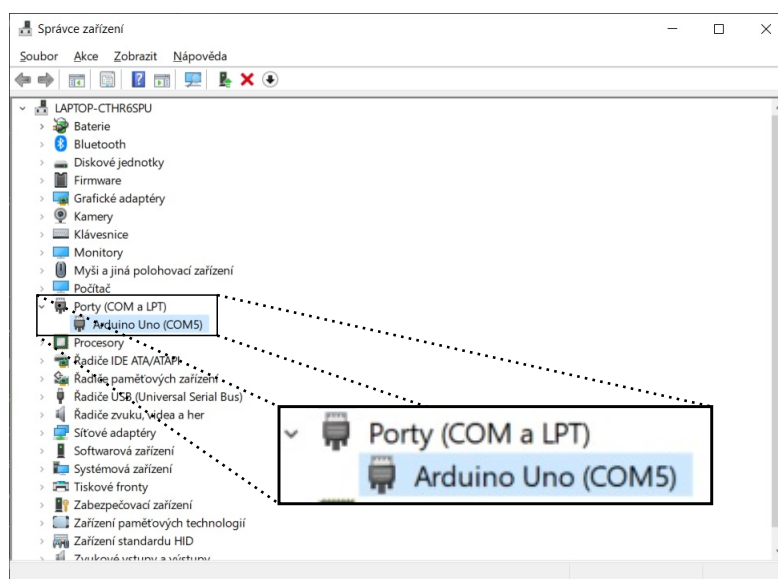


Obr. B.2: Konfigurační okno sériové linky programu Tera Term.

B.2.2 Nastavení programu ArduinoOBDSimulator

Je zde opět nutno zvolit správnou sériovou linku. Tentokrát se jedná o linku pro vývojovou desku Arduino UNO. Pro úspěšnou konfiguraci jsou nutné následující kroky:

1. Spuštění programu ArduinoOBDSimulator.
2. Spuštění systémového programu Správce zařízení (u systému Windows 10 stačí stisknout Windows klávesu, napsat „Správce zařízení“ a potvrdit).
3. V programu zjistíme číslo COM portu pro vývojovou desku Arduino UNO (viz Obr. B.3).
4. Zjištěný COM port zvolíme v příslušné položce programu ArduinOBDSimulator.
5. Položku „Baud Rate“ ponecháme na hodnotě 9600.
6. Kliknutím na tlačítko „Open“ navážeme spojení s vývojovou deskou.



Obr. B.3: Náhled do Správce zařízení v operačním systému Windows 10.

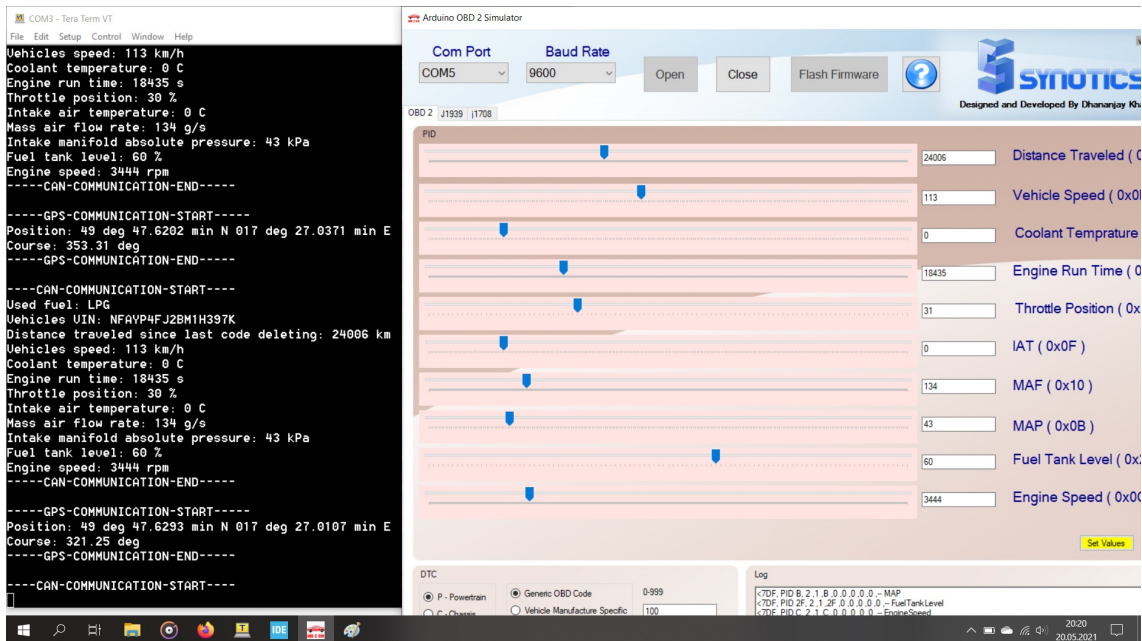
B.3 Ovládání pracoviště z obslužného PC

po úspěšné konfiguraci je již možno ovládat pracoviště z obslužného PC. Obrazovku je možno zorganizovat, tak jako na Obr. B.4. Pomocí posuvníku v programu ArduinoOBDSimulator, jsme schopni měnit jednotlivé parametry emulátoru. Data načtená z emulátoru jsou prezentována v okně programu Tera Term. Mimo hodnot parametrů emulátoru, které jsou přenášeny pomocí rozhraní CAN bus, jsou zde také prezentovány data z GPS modulu. Dle výpisu můžeme vidět, zda byla data o pozici a azimutu zařízení úspěšně načtena, zda nedošlo k úspěšnému dokončení akvizičního procesu nebo zda nebyla navázána komunikace s GPS modulem.

V případě, že nedošlo k navázání spojení mezi vývojovou deskou NUCLEO-F103RB a GPS modulem (je vypisována hláška „No data from GPS“), tak je nutno

tento problém odstranit. Odstranění problému proběhne pravděpodobně provedením těchto úkonů:

- Prověření zda nebylo přerušeno datové spojení mezi vývojovou deskou a GPS modulem (odpojení některého z datových vodičů RX <-> TX nebo vodiče GND <-> GND).
- Ověření zda je GPS modul správně napájen, viz kapitola B.1.2.



Obr. B.4: Snímek obrazovky PC obsluhující zařízení (vlevo Tera Term, vpravo ArduinoOBDSimulator).