

Parkovací senzor pro automobil s výstupem na mobilní telefon

Veronika Cichrová

Bakalářská práce
2017



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Veronika Cichrová**
Osobní číslo: **A14298**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Parkovací senzor pro automobil s výstupem na mobilní telefon**

Téma anglicky: **A Car Parking Sensor Linked to Mobile Phone Output**

Zásady pro vypracování:

1. **Prostudujte a popište funkci parkovacích senzorů a existující řešení s výstupem na mobilní telefon.**
2. **Navrhněte vlastní zařízení s využitím mikropočítače a vhodných senzorů.**
3. **Realizujte prototyp navrženého zařízení.**
4. **Vytvořte programové vybavení pro použitý mikropočítač.**
5. **Vytvořte programové vybavení pro mobilní telefon s operačním systémem Android.**

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy – linkery, práce s ATMEGA AVR a MSP430, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN, 2003. ISBN 80-730-0077-6.**
2. **MCCONNELL, Steve. Dokonalý kód: umění programování a techniky tvorby software. Vyd. 1. Brno: Computer Press, 2005, 894 s. ISBN 802510849x.**
3. **NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. Praha: BEN – technická literatura, 2005. ISBN 80-7300-141-1.**
4. **PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN – technická literatura, 2004. ISBN 80-7300-110-1.**
5. **SELECKÝ, Matuš. Arduino. Praha: COMPUTER PRESS – technická literatura, 2016. ISBN 978-80-251-4840-2.**

Vedoucí bakalářské práce:

Ing. Jan Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

24. února 2017

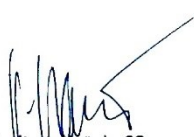
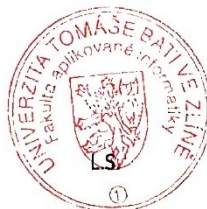
Termín odevzdání bakalářské práce:

24. května 2017

Ve Zlíně dne 24. února 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Název bakalářské práce: Parkovací senzor pro automobil s výstupem na mobilní telefon

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí. Že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s tím, že vyrovnaní případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považuji se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 23. 5. 2014


.....
podpis diplomanta

ABSTRAKT

Cílem této bakalářské práce je vytvoření parkovacího asistenta s ultrazvukovými senzory pro osobní automobil. Systém je řízen mikropočítačem a je bezdrátově propojen s mobilním zařízením pracujícím na platformě Android, které slouží jako výstupní zařízení. V teoretické části jsou popsány principy a vlastnosti jednotlivých potřebných prvků. V druhé části se nachází samotný návrh a praktické ztvárnění parkovacího asistenta.

Klíčová slova: Arduino, Android, parkovací asistent, bezdrátová komunikace, mikropočítač

ABSTRACT

The aim of this bachelor thesis is to create a parking assistant with ultrasonic sensors for a passenger car. The system is microprocessor-controlled and wirelessly connected to an Android mobile device that serves as an output device. The theoretical part describes the principles and the characteristics of the individual elements needed. In the second part there is the proposal and practical representation of the parking assistant.

Keywords: Arduino, Android, parking assistant, wireless communication, microcomputer

Chtěla bych poděkovat panu Ing. Janu Dolinayovi, Ph.D. především za veškerý věnovaný čas, odbornou pomoc, cenné rady a celkově za to, že odpověď na otázku nikdy nebyla stručná, ale naopak vyčerpávající a užitečná. Dále bych chtěla poděkovat Viktoru Schwarzovi za pomoc při zorientování se v Android studiu. Také bych chtěla poděkovat své rodině a hlavně mamince za veškerou podporu při psaní bakalářské práce a v průběhu celého studia.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PARKOVACÍ SENZORY	11
1.1 ULTRAZVUKOVÝ SENZOR	12
1.1.1 Další typy senzorů	13
1.2 BEZDRÁTOVÝ PŘENOS DAT.....	15
1.2.1 WLAN.....	15
1.2.2 Bluetooth	16
1.3 ZOBRAZOVACÍ JEDNOTKY	17
1.3.1 LED ukazatel.....	18
1.3.2 Palubní počítač	18
1.3.3 Se základním LCD displejem.....	18
1.3.4 S grafickým LCD displejem.....	18
1.3.5 Ve zpětném zrcátku	18
1.3.6 Telefon/tablet	18
1.3.7 Zvukové upozornění.....	19
1.4 PŘEHLED KOMERČNÍCH ŘEŠENÍ	19
1.4.1 Drátová komunikace	19
1.4.2 Bezdrátová Wi-Fi komunikace	19
1.4.3 Bezdrátová bluetooth komunikace	20
2 OPEN SOURCE	21
2.1 SOFTWARE	21
2.2 HARDWARE	21
3 PLATFORMA ARDUINO	22
3.1 TYPY DESEK	22
3.2 ARDUINO SHIELDY	26
3.2.1 Typy základních Arduino Shieldů [17].....	26
3.3 DALŠÍ ARDUINO MODULY A PERIFERIE.....	27
3.3.1 Druhy modulů dle určení	28
3.4 ARDUINO KLONY	29
3.5 NAPÁJENÍ	30
4 OPERAČNÍ SYSTÉM ANDROID	31
4.1 VERZE OS ANDROID	31
5 VÝVOJOVÁ PROSTŘEDÍ	32
5.1 PROSTŘEDÍ ARDUINO IDE.....	32
5.1.1 Popis vývojového prostředí Arduino IDE.....	33
5.2 ANDROID STUDIO	34
5.2.1 Základní prvky projektu	35
5.2.2 Emulátor.....	35
II PRAKTICKÁ ČÁST	37
6 NÁVRH ŘEŠENÍ HARDWARU	38

6.1	ARDUINO MEGA2560.....	38
6.2	ULTRAZVUKOVÉ SENZORY JSN-SR04T	39
6.3	BLUETOOTH HC-05.....	40
6.4	NÁVRH ZAPOJENÍ	41
7	PROGRAMOVÁ ČÁST ARDUINA	43
7.1	VÝVOJOVÝ DIAGRAM	43
7.2	PROGRAM.....	44
8	PROGRAM PRO ZAŘÍZENÍ S OPERAČNÍM SYSTÉMEM ANDROID	48
8.1	VÝVOJOVÝ DIAGRAM	48
8.2	PROGRAM.....	49
8.2.1	Grafický návrh	49
8.2.2	Oprávnění	50
8.2.3	Aktivity	51
9	PRAKTICKÉ OVĚŘENÍ	52
9.1	ZPROVOZNĚNÍ PARKOVACÍHO ASISTENTA	52
9.2	VYUŽITÍ VÝSTRAŽNÝCH PRVKŮ.....	52
9.3	TESTOVÁNÍ.....	52
	ZÁVĚR	54
	SEZNAM POUŽITÉ LITERATURY.....	55
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	59
	SEZNAM OBRÁZKŮ	61
	SEZNAM TABULEK.....	63

ÚVOD

V dnešní době plné vymožeností usnadňujícími život je možné téměř vše. Nápaditosti se meze nekladou a trh je otevřen v podstatě všemu. Všude ubývá drátových komunikací a jsou nahrazeny bezdrátovými. Pohodlnější a taky na pohled příjemnější, než pokládání kabeláže je bluetooth nebo Wi-Fi. Obzvláště pro parkovacího asistenta není ideální vést kabely napříč celým autem. Složitá montáž, nutnost rozebrání částí automobilu, aby byly kabely dobře ukryté nebo je mít jen tak položené v autě není hezké ani praktické. Proto je bezdrátové zařízení tou správnou volbou.

Parkovací asistent vyžaduje vhodné umístění výstupního zařízení. Jsou různé možnosti, ale většina z nich vyžaduje montáž na palubní desku. Jsou případy, kdy takový zásah není žádoucí. V tom případě může být vhodnou volbou použití svého smartphonu. Mobilní telefony jsou čím dál více využívaným zařízením ke všem aktivitám. Můžeme si na nich hlídat svůj příjem a výdej kalorií, řeknou nám, kdy po požití alkoholu můžeme sednout za volant nebo zjistit, kdy nám jede nejbližší vlak. Nedílnou součástí mobilních zařízení se stalo využívání GPS navigace v automobilu, tak proč ne parkovací senzory.

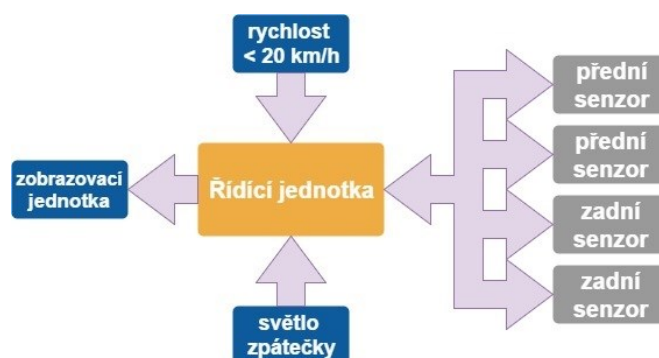
Tato bakalářská práce se zabývá využitím mobilního telefonu s operačním systémem Android jako výstupního zařízení ultrazvukových parkovacích senzorů. V teoretické části je nastíněna problematika jednotlivých využívaných částí. Možnost využití jiných typů senzorů, zobrazovacích zařízení nebo možnosti využití Arduina. V praktické části je popsán samotný návrh zařízení. Je předveden jak návrh hardwaru zařízení, tak jednotlivých softwarových částí. Vše je podrobně popsáno v příslušných částech.

I. TEORETICKÁ ČÁST

1 PARKOVACÍ SENZORY

Parkovací senzory slouží jako pomůcka řidiče při parkování auta. Senzory reagují na zařazení zpátečky, tím se připojí k napájecímu napětí a začnou fungovat. Poté ukazují vzdálenost cca od 30 do 300 cm, pokud jde tedy o ultrazvukové senzory. Vzdálenost se může lišit v závislosti na typu senzoru. Senzory obvykle doprovází zvukový signál s ukazatelem vzdálenosti nebo kamerou. Frekvence pípání zvukového signálu se zvyšuje s přiblížením se k objektu.

Blokové schéma parkovacího systému můžeme vidět na obrázku:



Obrázek 1: Blokové schéma parkovacího systému

Obrázek uvažuje případ, kdy jsou nainstalovány přední i zadní parkovací senzory. Přední parkovací senzory obvykle sepnou, při rychlosti menší než 20 km/h. Zadní senzory reagují na zařazení zpátečky a jsou připojeny na světlo zpátečky. Tyto informace do řídicí jednotky přichází jako vstupy. Senzory jsou vstupně výstupní. Splněním podmínky, že auto couvá, nebo je dopředná rychlost menší jak 20 km/h, se sepne řídicí jednotka. Řídicí jednotka dodává senzorům potřebnou energii a senzory jí naopak posílají naměřená data, která jsou v řídicí jednotce dále zpracována. Konečná zpracovaná data jsou vysílá do jediného čistě výstupního zařízení, kterým je zobrazovací jednotka. U některých systémů je možné k zprovoznění systému zmáčknout tlačítko. Tyto metody spuštění mohou být kombinovány. Některé parkovací asistenty jsou vybaveny parkovací kamerou s IR nočním viděním, tím by do schématu přibyl blok se snímací kamerou.

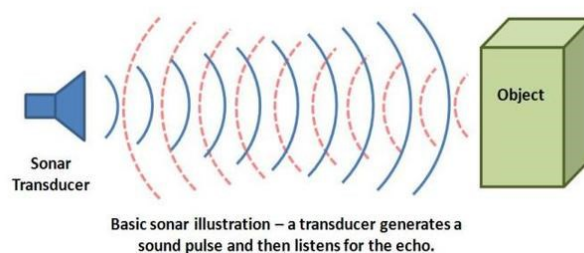
Nejčastěji používané senzory jsou ultrazvukové, méně časté ale přitom účinné jsou senzory elektromagnetické. Výhoda Elektromagnetických senzorů je ta, že nevznikají mrtvá místa měření. Nevýhod je však po více. Největší nevýhodou je detekce pouze pohyblivých překážek, takže v případě že auto stojí i když u zdi, překážka není viditelná. Také nedokáže rozpoznat, kde se překážka nachází. Funguje pouze do vzdálenosti 70 cm.

1.1 Ultrazvukový senzor

Ultrazvuk není pro lidské ucho slyšitelný, protože překračuje hranici spektra slyšitelnosti. Ovšem živočichové s lépe vyvinutým sluchem např. delfini či netopýři ho používají k běžnému životu v podobě komunikace nebo echolokace. [1], [3]

Ultrazvuk je také používán v lékařství, čištění, zvlhčování vzduchu a další. V lékařství je ultrazvuk používán třeba v echokardiografii a ultrasonografii, kde vlny o frekvenci v rozmezí 1 až 18 MHz sice projdou tělem, ale odráží se od orgánů díky tomu, že každá tkáň má jinou akustickou impedanci a dochází k částečnému odrazu. [1], [2]

Princip ultrazvukového senzoru je takový, že senzor generuje vysokofrekvenční zvukové vlny nad 20 000 Hz, které se v případě nárazu do vzdáleného předmětu (cca 20-400 cm) odrazí a vrátí zpět k senzoru. Zvukové vlny pro měření vzdálenosti se generují nejčastěji za pomoci elektrostriktce. To je opačný piezoelektrický jev. Jev je způsoben přivedením napětí k piezoelektrickému krystalu, kde vytvoří napětí na jeho povrchu tím, že se krystal deformuje. Zvukové vlny tedy vznikají tím, že po přivedení napětí, se krystal smršťuje a rozpíná a tvoří vibrace. Při nárazu vlny do předmětu, se odrazí a vrací zpět k senzoru, kde je vyhodnocována za pomoci vygenerovaného napětí elektronikou v senzoru. Pro využití ultrazvukových vln je potřeba vhodné prostředí. Tyto senzory není možné využít např. ve vakuu. Nejvhodnější použití je v prostředí vzduchu či kapaliny. [1], [3]



Obrázek 2: Odraz ultrazvukového senzoru [3]

Senzory jsou sami o sobě velmi odolné. Uzavřením v plastovém pouzdře jsou odolné proti vodě, teplu, nárazu. Jsou těsněny a všechny namáhané části jsou upevněny gumovými kroužky, které zabraňují pohybu v pouzdře. Senzory jsou tepelně testovány v rozmezí cca -40 až +90 °C, během kterých by měly bezchybně pracovat. [1], [36]

Je mnoho faktorů, které mohou výsledky měření zkreslovat. Ovlivňující faktory jsou například silný déšť, vítr a teplota vzduchu.

Nepřesnost měření ve velkém horku lze kompenzovat přidáním teploměru vzduchu. Řídicí jednotka do výpočtů zahrne i teplotu a tím by bylo možné získat skutečnou rychlost vzduchu namísto zkrácené teplotou.

Tvar překážky může činit také značný problém, především pokud je objekt kulatý či šikmý. Problém se projevuje v tom, že vlna po odrazu může nabrat špatný směr a nikdy se nevrátí zpět, nebo může několika odrazy o jiné předměty prodloužit zpáteční cestu k senzoru. Oba případy měření znehodnotí.

U všech druhů senzorů je třeba dbát toho, kam senzory umístíme. Většina senzorů má v rozptylu tvar kuželu, a proto v této dráze nemůže být žádná rušivá překážka, která by znehodnotila měření. Také je důležité uvážit kolik senzorů je nutné použít, aby nevznikla slepá (nepokrytá) místa. A však ani počet senzorů nám nemůže zajistit 100 % spolehlivosti, když budou senzory ve špatné poloze či úhlu. Nevyhnutelnou překážkou může být tažné zařízení nebo zavěšená rezerva na pátých dveřích vozu. Dnešní moderní zařízení jsou na to připravená a mají funkci, která tyto překážky dokáže ignorovat.

1.1.1 Další typy senzorů

Pro měření vzdálenosti je možné využití i jiných typů senzorů než ultrazvukových. Tyto ostatní senzory jsou však méně používané z důvodu velké pořizovací ceny, malého rozpětí paprsku, krátké vzdálenosti a další...

Elektromagnetické senzory:

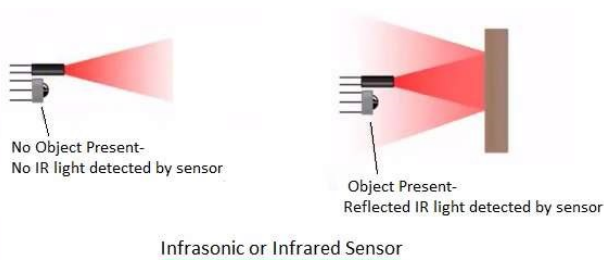
Elektromagnetické senzory se řadí do snímačů indukčních, generovaných. To znamená že naměřené hodnoty jsou převedeny na změnu magnetického toku budícího obvodu. Tím je vyvoláno indukované napětí. Neelektrická veličina je u tohoto druhu senzoru působícími prvkem na rychlost změny magnetického toku. Magnetický tok je obvykle měněn změnou impedance v magnetickém obvodu. [4]



Obrázek 3: Ukázka rozsahu elektromagnetického vlnění [30]

IR senzory:

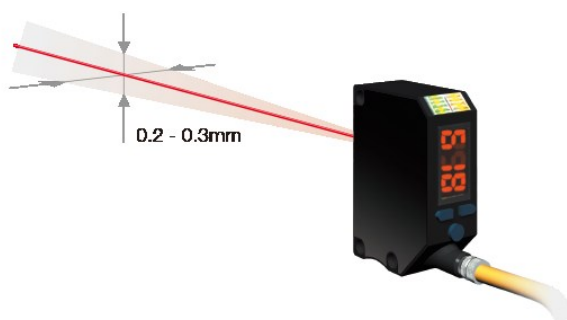
Tyto senzory fungují za využití IR (infračervené) diody. Tato dioda vyšle IR signál a při odrazu je snímán úhel dopadu CMOS senzorem. Z této informace je vypočítána vzdálenost objektu. Signál je rychlejší, než u ultrazvukového senzoru a však infračervený senzor má menší rozsah rozptylu kuželu paprsku. U ultrazvukového senzoru se rozptyl pohybuje kolem 75°, zatím co u IR senzorů je to kolem pouhých 60°. Maximální dosah paprsku senzoru se pohybuje kolem 80 cm. [28]



Obrázek 4: Odraz IR senzoru [32]

Laserové senzory:

Tyto senzory mají velmi velký dosah měření a přesnosti, proto jsou využívány spíše v průmyslu a stavebnictví ve formě bezkontaktního metru. Laserový paprsek nemá tvar kuželu, ale jedná se o úzký sdružený svazek. Tedy není nejvhodnější volbou k využití parkovacích senzorů. Nevýhodou je proti předchozím senzorům vyšší pořizovací cena. Světelný paprsek je vyslán k objektu měření, od kterého se odrazí a putuje zpět k senzoru, kde projde čočkou do detektoru. [29]



Obrázek 5 Ukázka laserového senzoru [31]

1.2 Bezdrátový přenos dat

Většina bezdrátových parkovacích senzorů využívá ke komunikaci drátové spojení. V případě bezdrátového je nejčastěji používáno Wi-Fi. Tato komunikace je rychlejší než Bluetooth a pro to i oblíbenější. Výhoda Bluetooth je zase nízká spotřeba energie a lepší funkčnost v prostředí s vyšším rušením. V případě parkovacích senzorů, jsou však tyto klady i zápory lehce zanedbatelné a je možné tyto druhy komunikací zaměnit. [5]

1.2.1 WLAN

WLAN neboli Wi-Fi je v dnešní době nejpoužívanější bezdrátovou sítí pro přenos dat, pro běžné uživatele. Je tomu tak především díky své rychlosti a dosahu. Nyní můžeme k internetu připojit téměř každé přenosné zařízení. Síť také slouží pro komunikaci mezi jednotlivými zařízeními. Původně bylo Wi-Fi navrženo jako bezdrátová náhrada USB kabelu. Mezi výhody také patří jednoduchost zavedení Wi-Fi sítí i tam, kde dříve nebylo možné internet zavést kvůli pokládání kabeláže. Nevýhodou je relativně snadná možnost odposlechu a nábourání sítě ke zneužití dat. Tuto nevýhodu můžeme správným zabezpečením eliminovat. Wi-Fi je standard lokálních bezdrátových sítí vycházejících z IEEE 802.11. Původní záminkou této sítě byla vzájemná komunikace mezi přenosnými zařízeními a využitím především v lokálních firemních sítích LAN. Později se stala nejvíce využívanou pro připojení k síti Internet. Oblíbenost byla tak velká, že bylo zpřístupněno 8 bezlicenčních pásem v rozpětí od 2,4 do 5 GHz. Největší problém je v zahlcení nejnižšího pásma, které využívají i jiné technologie, jako třeba Bluetooth nebo různé periferie od klávesnic a myši po sluchátka a dál. Tohle mělo negativní dopad ve formě silného rušení z důvodu zahlcení pásma. [6]

Podle účelu využití sítě se rozhoduje o tom, jakým způsobem bude vybudována. Pro propojení sítí je důležitá frekvence a SSID (Service Set Identifier). SSID je identifikátor sítě obsahující až 32 ASCII znaků. Propojení zařízení vysílající SSID a připojujícího zařízení se říká Access Point. Access point se stará o routování provozu drátových a bezdrátových sítí. [6]

Síť Ad-hoc:

Tyto sítě jsou propojeny na bázi peer-to-peer a zařízení jsou si rovné. Levná výstavba, rychlost, jednoduchost jsou výhody tohoto zapojení. Nevýhody mohou vzniknout při neaktivitě některého ze zařízení tvořících síť. Zapojení Ad-hoc vyžaduje, aby všechna zařízení, která spolu mají komunikovat, byla v radiovém dosahu. Tedy jsou vhodné do malých prostor.

Tyto sítě jsou především využívány nárazově v případě potřeby, kdy je nutnost převést data z jednoho zařízení do druhého. Velmi oblíbeným využitím je také při LAN párty. Je to jednoduché a rychlé. Rychlost dosahuje až 11Mb/s. V trvalých sítích se prakticky nevyužívá. [7]

Infrastrukturní síť:

Sítě obsahují jeden či více komunikačních prvků (Access point). Každý komunikační prvek vysílá své SSID na úrovni broadcastu, podle kterého si klient vybírá kam se připojí. Komunikační prvky zastávají vlastnosti mezi bezdrátovou a drátovou sítí pomocí síťového mostu (bridge). Klient vždy požádá Access point o připojení do sítě a podle oprávnění klienta bude přístup povolen nebo zamítnut. I v případě, kdy spolu komunikují pouze dvě zařízení, vždy je komunikace vedena přes prostředníka v podobě Access pointu. Každý klient smí být připojen pouze k jednomu přístupovému bodu, ale na tento bod může být prakticky připojeno desítky dalších zařízení. [8]

1.2.2 Bluetooth

Jedná se o nejrozšířenější bezdrátovou technologii využívanou pro přenos dat na krátké vzdálenosti. Poprvé byla technologie ve zkoumání v roce 1994, kdy měla nahradit drátovou komunikaci telefonu s jiným zařízením. Během roku 1998 se spojilo pět firem (Nokia, Ericsson, Toshiba, Intel, IBM) tvořících Bluetooth Special Interest Group (BSIG) s cílem vytvoření univerzálního standardu. [9], [10]

První propracovaná verze technologie přišla na svět v červenci 1999 a rovnou se těšila velké oblibě. To jak u obyčejných uživatelů, tak v průmyslu. Jenom během prvních deseti let BT na trhu, se vyrobilo nejméně 1,5 miliardy mikropočítačů. Netrvalo dlouho a skupina BSIG byla rozšířena o další členy. V dnešní době čítá nad 10 000 členů. [9]

Standard samozřejmě nezůstal u první verze 1.0a. Druhá verze přišla dříve než se čekalo, a to ještě na konci téhož roku ve formě 1.0b. Tyto první verze měly spoustu nevychytaných chyb a nepřesností. Jedna z nich byla špatná kompatibilita.

Standard 1.1 fungoval od začátku roku 2001, kdy představil první solidní produkt, vhodný pro komerční prodej. [9]

Další verze nesla označení 1.2 a objevila se v listopadu 2003. Byla navržena transparentně a rozšířena o technologii Adaptive Frequency Hopping (AFHSS) a možnost vytvoření rychlého připojení. Přenosová rychlost až 721 kbit/s. [9]

Verze 2.0 pochází z roku 2004. Je rozšířený o Enhanced Data Rate (EDR) s kterou dosahuje přenosové rychlosti až 2,2 Mbit/s. Jedná se o nejčastěji používaný typ. Dále byl v červenci roce 2007 rozšířen o podporu Near Field Communications s umožněním rychlejšího párování zařízení, pod označením 2.1. [9]

Verze 3.0 podporuje teoretickou rychlost až 24 Mbit/s. Avšak tato rychlost je použita k navázání spojení a vysokorychlostního přenosu se souběžným zapojením protokolu 802.11 (Wi-Fi). [9]

Další verze byla vydána pod označením 4.0 v červenci 2010. Tato verze však nemá nahradit předchozí verzi 3.0. Má jen umožnit výběr mezi větší výdrží baterie u typu 4.0 nebo rychlejší přenos dat u 3.0. Nová nízkoenergetická náročnost je vhodnou volbou především pro handsfree, sluchátka a jiná bezdrátová zařízení, kde je prioritní výdrž baterie na jedno nabití. Starší verze nabízející rychlejší přenos dat za větší energetické spotřeby, by byla zvolena např. při přenosu většího objemu dat. [9]

Poslední specifikace byla vydána v prosinci 2014 s označením 4.2 a zahrnuje protokol 6LoWPAN, který je využíván např. chytrými žárovkami. [9]

Z důvodu zabezpečení spojení dvou zařízení je nutné jejich spárování. Což znamená rozpoznání konkrétního zařízení, ke kterému je potřebné se připojit klidně i opakovaně. Příkladem mohou být BT sluchátka, ke kterým se po spárování telefon připojí automaticky bez zásahu uživatele vždy, když je to možné (tzn. Když mají obě zařízení zapnuté BT a jsou v dosahu). Proces párování může proběhnout buď žádostí od uživatele, který žádá připojení, nebo je spuštěn automaticky při prvním připojení, kde je nutná kontrola zařízení na druhé straně. Párování se v budoucnu již se stejným zařízením neopakuje. Zařízení uchovává seznam spárovaných zařízení. Tento seznam je možné upravovat a zařízení promazávat. [9], [10]

1.3 Zobrazovací jednotky

Možných druhů výstupu zobrazení aktuálního stavu vozidla při parkování je spousta. Některé jsou vizuální, jiné zvukové. Mohou se lišit zařízením, kde je vizuální výstup zobrazen i umístěním zařízení.

1.3.1 LED ukazatel

Nejzákladnějším možným ukazatelem vzdálenosti u parkovacích senzorů je LED ukazatel, obvykle doprovázený sedmi-segmentovým displejem. Sedmi-segmentový display se nachází uprostřed zařízení a číselně ukazuje vzdálenost od nejbližšího objektu. LED diody jsou poté na obě strany směrem ven poskládány v pořadí červená, pro velmi blízkou vzdálenost. Žlutá, pro větší vzdálenost a zelená pro největší vzdálenost.

1.3.2 Palubní počítač

Tato metoda zobrazení v palubním počítači je obvykle využívána v automobilech, které obsahují parkovací senzory již při sjezdu z výrobní linky. Naměřené výsledky jsou zobrazovány v zabudovaném zařízení automobilu.

1.3.3 Se základním LCD displejem

U většiny zobrazovacích jednotek k parkovacím panelům je využíván jednobarevný LCD display s RGB podsvícením. Na těchto displejích je obvykle k vidění vzdálenost od nejbližšího objektu a sloupcový ukazatel vzdálenosti.

1.3.4 S grafickým LCD displejem

Některé dražší asistenty obsahují klasický barevný LCD displej, na kterém je vše podrobněji a barevně zobrazeno. Obsahují vše, co základní plus něco navíc včetně oživením barvami. Někdy dokonce obsahují parkovací kameru, v tom případě je výstup kamery zobrazován na tomto displeji.

1.3.5 Ve zpětném zrcátku

Je možné si zakoupit parkovacího asistenta s výstupem ve zpětném zrcátku. U tohoto typu se jedná obvykle o asistenta s kamerou upevněnou na pátých dveřích vozu. Záznam v zrcátku vidíme z pohledu první osoby. Zobrazení můžeme vidět na polovině prostoru zrcátka, nebo na celém, dle zvoleného typu. Zrcátko při běžné jízdě se chová jako klasické, kamera se zapne zařazením zpátečky.

1.3.6 Telefon/tablet

Jedná se o nejméně obvyklý způsob. Výstup senzoru v mobilních zařízeních lze zobrazovat pomocí speciálních aplikací příslušících ke konkrétnímu senzoru.

1.3.7 Zvukové upozornění

Zvukové upozornění ve formě pípání, nebo dlouhého tónu při maximálním přiblížení obsahují téměř všechny parkovací asistenty a obvykle lze tuto možnost i vypnout.

1.4 Přehled komerčních řešení

V následujícím přehledu budou popsány některé parkovací senzory v kategoriích dle způsobu komunikace mezi řídicí a zobrazovací jednotkou.

1.4.1 Drátová komunikace

Steelmate PTS410V11

Jedná se o parkovacího asistenta disponujícími 4 senzory a jednobarevným LCD displejem. Obsahuje Antihook funkci. Tedy umí ignorovat rezervu zavěšenou na pátých dveřích, nebo tažné zařízení. Součástí je také zvuková signalizace a automatická detekce poškozeného senzoru, která je spuštěná ihned po zařazení zpátečky. [11]



Obrázek 6: Steelmate PTS410V11 [11]

1.4.2 Bezdrátová Wi-Fi komunikace

Steelmate PTS400T

Tento parkovací asistent využívá k zobrazování LCD display a k upozorňování zvukovou signalizací. Obsahuje automatický systém detekce funkčnosti senzorů, který je spuštěn vždy při zařazení zpátečky. Zaznamenává rozsah měření od 30 do 250 cm. Přiblížení je graficky znázorněno sloupcovým grafem, a to včetně vzdálenosti k překážce. Využívá technologii anti falešného poplachu. Data jsou přenášena pomocí Wi-Fi modulu. [12]



Obrázek 7: Steelmate PTS400T [12]

1.4.3 Bezdrátová bluetooth komunikace

FenSens

V San Franciscu (USA) se právě objevila novinka ve formě parkovacích senzorů, kvůli kterým není třeba vrtat části svého auta a k instalaci zařízení stačí pouhých 5 minut. Jde o připevněné senzory do rámu SPZ, tedy stačí vyměnit stávající rám za nový. K funkčnosti se nevyužívá žádných složitých kabelů. Senzor je napájený univerzální baterií velikosti AA, která vydrží nabitá 5 měsíců. Po vybití je třeba baterie dobít nebo vyměnit. Jako zobrazovací jednotka je využíván mobilní telefon s operačním systémem iOS nebo Android. Komunikace mezi mobilním telefonem či tabletem a senzorem je zprostředkována pomocí Bluetooth 4.0. Rám obsahuje alarm proti krádeži zařízení a bezpečnostní šrouby a dva ultrazvukové senzory. Značka udává že minimální požadavky pro operační systémy jsou u Androidu 4.4 (doporučené 5.0) a u iPhoneu iOS7.1. Bluetooth je i zpětně kompatibilní. Přístroj FenSens se spustí pomocí tlačítka v aplikaci. Samotná zobrazovací jednotka ve formě telefonu je včetně grafického znázornění situace doprovázená vibracemi a zvukovými signály. [13], [14]



Obrázek 8: FenSens [14]

2 OPEN SOURCE

2.1 Software

Open Source Software znamená, že se jedná o otevřený (volně dostupný) kód, který může kdokoli vidět, a dokonce jej modifikovat. V podstatě spolu všichni programátoři spolupracují na určitém vývoji. Termín Open Source se využívá i softwarech, které poskytují svou licenci zdarma. Jde zde o nekomerčnost, a proto jsou jiné způsoby, jak vydělat i na tomhle. Může se stát, že samostatný software je zdarma, ale další rozšiřující licence obohacující funkčnost softwaru jsou za příplatek. [15]

Běžně se používají Open Source technologie na webu. Jedná se třeba o Apache HTTP Server, PHP, My SQL, WordPress nebo Drupal. [15]

Mimo webové technologie je mezi Open source software řazen Linux, Android, Blender nebo třeba MediaWiki, na kterém běží Wikipedie. [15]

2.2 Hardware

Mimo Software může být také Open Source Hardware. Hardware, který se uživatel/společnost může upravovat, vylepšovat či kopírovat legálně dle libosti. Tak vzniká nové větvení originální linie. [15]

Zde můžeme zařadit v první řadě Arduino, Openmoko (mobilní telefony), LEON (mikroprocesory) a další. [15]

3 PLATFORMA ARDUINO

Vývojová deska Arduino je v dnešní době čím dál oblíbenější open-source platformou pro vývoj aplikací s mikropočítači. Mikropočítač je elektronická programovatelná součástka v podobě integrovaného obvodu [37]. Pro použití mikropočítače v nějaké aplikaci je potřeba vytvořit desku plošného spoje nebo použít hotové řešení ve formě vývojové desky, jako je právě Arduino.

V letech 2005-2013 bylo prodáno 700 000 kusů oficiálních zařízení platformy Arduina. Příjem za rok 2013 byl kolem jednoho milionu dolarů se získaným podílem na trhu cca 3-7 %. Přitom to začalo myšlenkou, vytvoření jednoduchého, levného vývojového setu pro studenty. Měla to být náhrada za desky BASIC Stamp, které byly drahé. Úspěch u studentů byl tak velký, že jeho distribuci rozšíří. Za roky vývoje vznikla spousta typů Arduina. Vzhledem k tomu že se jedná o open-source projekt, vznikala i spousta neoficiálních klonů. Některé jsou v rámci možností totožné kopie, jen za příznivější cenu. [16], [17]

Každé Arduino obsahuje mikropočítač firmy Atmel, spoustu elektronických komponent a podobné zpracování. Typů desek je mnoho a každá má svůj název, podle kterého je rozeznáme. Některé desky mají kromě názvu i označení verze (např. Rev3, R3). V různých verzích se může lehce lišit rozložení součástek a design. Nejedná se o zásadní změny, na to, aby se tvořil nový oficiální typ desky. Většina desek disponuje převodníkem pro komunikaci s PC pomocí rozhraní USB a mikropočítačem. Některým deskám převodník chybí. Nejčastěji kvůli úspoře místa, a proto je nutné použití externího převodníku pro komunikaci s PC. Další případ má převodník zabudovaný již v mikropočítači. [16], [17]

3.1 Typy desek

Arduino Mini

- Procesor Atmega328, 16 MHz (dříve využíván procesor Atmega168).
- Obsahuje 14 digitálních (z toho 6 PWM) a 8 analogových I/O pinů.
- Nejmenší verze Arduina, která pro svou velikost „zaplatila“ absencí USB portu, proto je nezbytné k programování použít externí převodník.
- Díky velikosti srovnatelnou s pěti korunovou mincí je vhodný do dálkových ovládání, vypínačů nebo třeba k zašití do oblečení a vytvoření multifunkční mikiny.

Arduino Nano

- Technické parametry jsou téměř totožné jako u Arduino Mini
- Rozměry nepatrně větší.
- Rozdílem proti Arduino Mini je především USB port s TTL převodníkem.

Arduino Micro

- Mikropočítač Atmega32u4, 16 MHz
- Obsahuje 20 digitálních (z toho 7 PWM) a 12 analogových I/O pinů.
- Převodník umístěný v mikropočítači, USB
- Výhodou tohoto mikropočítače je využití ve formě klávesnice či myši. Důvodem je možnost zasílání příkazů do počítače. Tato možnost je i u jiných typů mikropočítačů, ale je nutné přeprogramování převodníku, což nemusí být jednoduchá záležitost.
- Vhodná pro tvorbu vlastních herních ovladačů a klávesnic.

LilyPad Arduino

- Mikropočítač ATmega328 nebo Atmega32u4 v závislosti na tom, zda je po desce vyžadováno Bluetooth
- Neobvyklý kulatý tvar
- Spoje vytvořeny vodivou nití
- Uzpůsobená k oděvům, je možné si vytvořit blikající mikinu

Arduino Fio

- Deska uzpůsobená k připojení bezdrátových modulů – Xbee moduly
- Procesor Atmega328P, 8MHz
- Napětí sníženo z 5 V na 3,3 V, kvůli modulům

Arduino Uno

- Jedná se o nejpoužívanější typ
- Přímí nástupce hlavní vývojové linie
- Procesor ATmega328, 16 MHz
- 14 digitálních (z toho 6 PWM) a 6 analogových I/O pinů

- Obsahuje USB i napájecí konektor

Arduino Leonardo

- Designově totožný s Arduino Uno
- Změna je v procesoru ATmega32u4. 16 MHz
- 20 digitálních (z toho 7 PWM) a 12 analogových I/O pinů
- Díky této verzi mikropočítače, má vlastnosti jako již zmíněné Arduino Micro
- Vstupní napětí 7-12 V
- Obsahuje Bluetooth i napájecí konektor

Arduino Yún

- Kromě procesoru ATmega32u4 najdeme i Atheros AR9331, který je schopný zprovoznit i odlehčenou verzi linuxu – Linino
- Softwarový bridge zajišťuje komunikaci mezi procesory
- Deska obsahuje micro USB určené pro ATmega32u4, normální USB určené pro linux
- Ethernetový port a Wi-Fi pro zprostředkování připojení k síti.
- Obsahuje 20 digitálních (z toho 7 PWM) a 12 analogových I/O pinů.

Arduino Mega2560

- Předchozí typ byl označen pod číslem 1280
- Mikropočítač ATmega2560, 16 MHz
- 54 digitálních (z toho 15 PWM a 4 Hardwarové sériové porty) a 16 analogových I/O pinů
- Obsahuje USB i napájecí konektory

Arduino Due

- Pokračovatel verze Mega, jen s výkonnějším mikropočítačem, nesrovnatelným s dosud uvedenými typy desek.
- Mikropočítač Atmel SAM3X8E s taktovací frekvencí 84 MHz s 32 bitovým jádrem.

- 2x micro USB konektory s odlišným účelem. První slouží k programování a druhý k připojení periférií, jako klávesnice, myš nebo i mobilní telefon.
- 54 digitálních (z toho 12 PWM a 4 Hardwarové sériové porty) a 16 analogových I/O pinů a další...
- Nevýhodou může být, že tento typ desky běží pouze na 3,3 V a vyšším napětím by byla poškozena

Arduino Esplora

- Jedná se o „hybridní“ desku, která je u Arduina první svého druhu.
- Obsahuje potenciometr, tlačítka i joystick a ze součástí méně viditelných nalezneme také akcelerometr, teploměr, piezzo bzučák, mikrofon, snímač osvětlení a piny připravené k připojení LCD displeje.
- S tímto typem lze vytvořit konzoli k ovládání her, nebo dokonce herní set.
- Obsahuje procesor ATmega32u4, 16 MHz
- K desce je možné připojit i klávesnici nebo myš.

Arduino robot

- Slouží k vytvoření robota
- Obsahuje procesor ATmega32u4, 16 MHz
- Obsahuje kompas, potenciometr, mikrofon, display a klávesnici s 5 tlačítky
- 11 digitálních (z toho 6 PWM) a 8 analogových I/O pinů

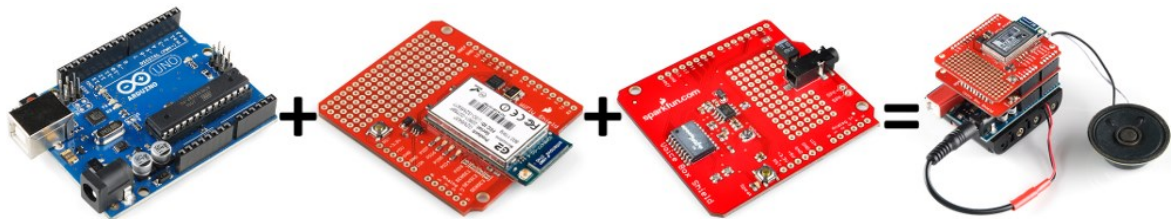
Arduino Intel Galileo

- Jak je již z názvu vidět, tak tento typ desky vznikl ve spolupráci s Intelem.
- Jedná se o první desku od Arduina fungující na procesoru Intel Quark SoC X1000 – 32 bitů, 400 MHz
- Obsahuje Ethernetový port, čtečku microSD karet, dvě USB a díky mini-PCI Express slotu je možné připojit přídatnou kartu

U výpisů druhů desek byl použit zdroj [17], [38].

3.2 Arduino Shieldy

Arduino Shieldy jsou doslova štíty. Jedná se modulární obvodové desky, které padnou přímo na základní Arduino desku (např. Mega, UNO). Většina těchto štítů lze na sebe stohovat. To znamená, že lze naskládat více štítů na sebe, jako na následujícím obrázku. U štítů je nutné, aby odpovídalo umístění napájecích, uzemňovacích a analogových pinů. Některé shieldy lze připojit přes konektor ICSP. Často je nutné využití pájky a připájení hradel. [18], [19]



Obrázek 9: Ukázka propojení Shieldu [19]

3.2.1 Typy základních Arduino Shieldů [17]

Wi-Fi Shield

- Umožňuje bezdrátovou Wi-Fi komunikaci
- Využívá standardu 802.11b/g na frekvenci 2.4 GHz
- Podporuje šifrování WEP a WPA2
- Na Shieldu je také umístěn slot pro microSD kartu, která může být využita k záznamu logování provozu sítě, nebo k ukládání jakýkoliv hodnot.

Motor Shield

- Tento modul slouží k připojení motoru, který se vyžívá např. při pohybu robota, nebo pohonu
- Je možné připojit dva nezávislé DC motory nebo jeden krokový při maximálním proudu 2 A na jeden motor
- U motorů lze řídit směr i rychlost otáčení

Ethernet Shield

- Umožňuje komunikaci pomocí LAN sítě
- Obsahuje tedy koncovku RJ-45 s přenosovou rychlostí 10/100 Mb

- Shield je schopný pozorovat a zaznamenávat veškerou komunikaci na síti a využívat Webový server
- Stejně jako Wi-Fi Shield obsahuje slot pro microSD kartu

Relé Shield

- Umožňuje spínání nezávislých obvodů
- Některé moduly zvládají spínat i 230 V
- Shieldy se vyrábí v provedení od 1 až 8 na sobě nezávislých relé

GSM Shield

- Zajišťuje komunikaci pomocí GSM sítě nebo i GPRS s rychlostí maximálně 85,6 kb/s
- Shield může odesílat SMS, nebo s nimi být ovládán

3.3 Další Arduino moduly a periferie

Rozdíl mezi Arduino Shieldem a Arduino modulem je takový, že modul je něco, co je připojeno k Arduino za pomoci drátků, a proto je snazší přístup ke všem ostatním pinům základní desky. Zatímco štít sedí přímo na desce Arduino a celou jí chrání. Některé Shieldy obsahují třeba sloty na microSD karty. Na obrázku můžeme vidět některé druhy modulů. [20], [21]



Obrázek 10: Ukázka modulů [20]

3.3.1 Druhy modulů dle určení

Komunikační moduly

- Moduly sítě Ethernet pro připojení kabelu RJ-45
- GSM modul pro komunikaci SMS
- GPS modul k určení polohy
- Komunikační modul pro jednosměrnou nebo obousměrnou bezdrátovou komunikaci
- Infračervený modul pro komunikaci optickým signálem (např. dálkové ovladače)
- Bluetooth modul pro bezdrátovou komunikaci
- RFID modul pro bezdrátovou komunikaci na velmi malou vzdálenost

Vstupní moduly

- Modul s klávesnicí nebo tlačítky
- Různé senzory vlhkosti, tlaku, teploty.
- Detektory, optické závory, senzory intenzity barvy a osvětlení.
- Elektronické kompas, zvukové senzory
- Převodníky

Výstupní moduly

- Moduly s Relé a spínači
- Modul pro ovládání stejnosměrných nebo krokových motorů
- LED moduly a bzučáky, Modul pro SD karty

Displeje

- Alfanaumerické LCD displeje
- Sedmi-segmentové maticové displeje
- OLED a LCD displeje (grafické)
- Dotykové TFT displeje

3.4 Arduino klony

S oficiální řadou vzniká i spousta dalších desek, které oficiální nejsou. Některé i na první pohled vypadají, že se jedná o originály, včetně výbavy i funkčnosti. Mnohdy stačí nainstalovat jen správný ovladač a klon se chová totožně jako originál. Hlavní rozdíl je v ceně. Kvalita je srovnatelná, pokud nebereme do úvahy úplně ty nejlevnější „braky“. Klony se dokonce skládají i z těch samých součástek, a to včetně procesoru. Samozřejmě není podmínkou, že klony jsou vždy kopie. Mimo kopírování vznikají i nové desky, které jsou většinou jednoúčelové a slouží ke konkrétní věci. Není nereálné si své Arduino vytvořit i doma. Všechny součástky a nákresy desek jsou volně dostupné. [23]



Obrázek 11: Porovnání Originál vs. Klon [22]

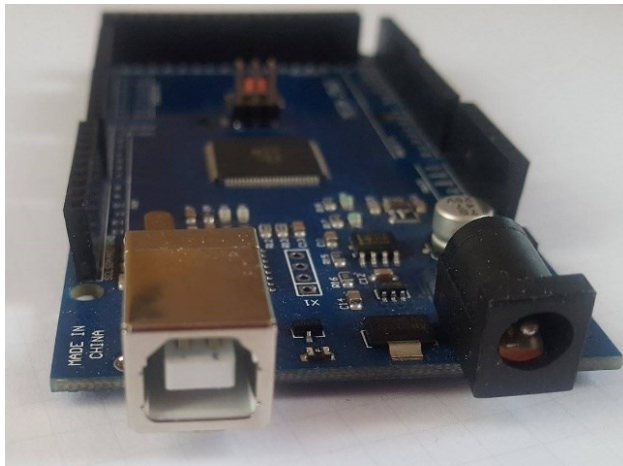
Klony jsou velmi oblíbené a možná se těší i větší popularity než samotný originál. Jak je psáno výše, tak zásadní rozdíl je hlavně v ceně, která je mnohdy méně než třetinová. Proto je pro lidi příznivější si raději koupit 3 kusy přímo z nějakého čínského e-shopu, kde je i poštovné zdarma než koupit jeden totožný originál. [23]

Jedinou zábranou v nekoupení klonu může být snad jen nějaký morální závazek, nebo svědomí s tím, že před několika lety si s vymyšlením a propracováním Arduina dal velkou práci a nyní je někdo okrádá a přizívuje se na nich. Pravdou však ale je, že oficiální distribuci Arduina drží na živu i klony, protože spousta držitelů klonů posunuje Arduino ku předu. Nadšenci přispívají komunitě svým duševním vlastnictvím, sdílí kódy a tvoří knihovny, které jsou použity jak u originálu, tak u klonu. [23]

3.5 Napájení

Platforma Arduino lze napájet hned několika způsoby:

- První je napájení přes USB. Jedná se konektor, který primárně slouží k sériové komunikaci a tím i nahrávání programu do Arduina. Tímto způsobem se může napájet skrze počítač, power-banku, zapalovač v autě nebo při použití adaptéru do zásuvky s přípojkou na USB.
- Druhé napájení je ze zdroje. Tuto možnost nemají všechny Arduino, jelikož nemají potřebný napájecí konektor. Lze napájet pomocí baterie či adaptéru.



Obrázek 12: Možnosti napájení

4 OPERAČNÍ SYSTÉM ANDROID

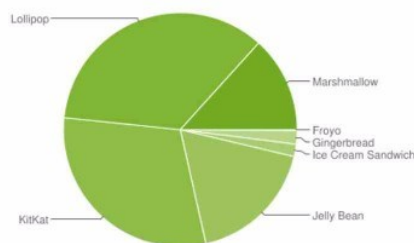
V dnešní době je nejrozšířenějším operačním systémem pro mobilním zařízením. Porazil dokonce Apple s jejich iOS. Společnost Android, Inc. Byla v srpnu roku 2005 koupena včetně jejich zaměstnanců společností Google. Jedná se open source platformu vytvořenou společností Google určenou především pro mobilní zařízení. Je založena na jádře Linuxu, stále se inovuje a přichází nové updaty. [24]

4.1 Verze OS Android

Za roky vyvíjení operačního systému vznikala spousta verzí. Všechny tyto verze byly zdarma dostupné a snadno instalovatelné. První oficiální verze vyšla v září 2008, pod názvem Android 1.0 a její revize 1.1 vyšla o půl roku později. Další přicházející verze již nesly za číselným označení i přívlastek ve formě sladkostí a dobrot v abecedním pořadí. [24]

Začalo se verzí 1.5 (Cupcake) a dále pokračovalo: 1.6 (Donut), 2.0/2.1 (Eclair), 2.2 (Froyo), 2.3/2.4 (Gingerbread), 3.0-3.2 (Honeycomb), 4.0.0-4.0.4 (Ice Cream Sandwich), 4.1-4.3 (Jelly Bean), 4.4 (KitKat), 5.0/5.1 (Lollipop), 6.0 (Marshmallow) a poslední dosud zveřejněnou verzí je Android 7.0/7.1 (Nougat). [25]

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.7%
4.1.x	Jelly Bean	16	6.4%
4.2.x		17	8.8%
4.3		18	2.6%
4.4	KitKat	19	30.1%
5.0	Lollipop	21	14.3%
5.1		22	20.8%
6.0	Marshmallow	23	13.3%



Obrázek 13: Statistika používání jednotlivých distribucí [25]

Každá z verzí v průběhu let přidala několik bezpečnostních opatření, aplikací, zjednodušení, a hlavně rozšíření možností práce na mobilním telefonu, či jiném zařízení.

5 VÝVOJOVÁ PROSTŘEDÍ

Vývojové prostředí (IDE – Integrated Development Environment) je software pracující v určitém programovacím jazyce, nebo jazycích. Vývojová prostředí se mohou značně lišit, a proto je důležitá volba toho správného.

IDE představuje software, v kterém je vyvíjena celá aplikace. Tento software obsahuje velké množství možných nástrojů pro usnadnění práce při vývoji, úpravách, překladu či ladění. Vývojové prostředí má za cíl snížení vynaloženého času programátora pro vývoj nového softwaru. Základní složky vývojového prostředí. [33]

Editor zdrojového kódu

Zde se sepisuje program ve zvoleném programovacím jazyce. Specializované Editory zdrojového kódu jsou na rozdíl od obyčejných textových editorů schopny zvýrazňovat syntaxi a její chyby nebo možnost nápovědy.

Kompilátor

Kompilátor (překladač) je softwarový nástroj pro překlad kódu z vyššího jazyka, do nižšího (strojového kódu).

Debugger

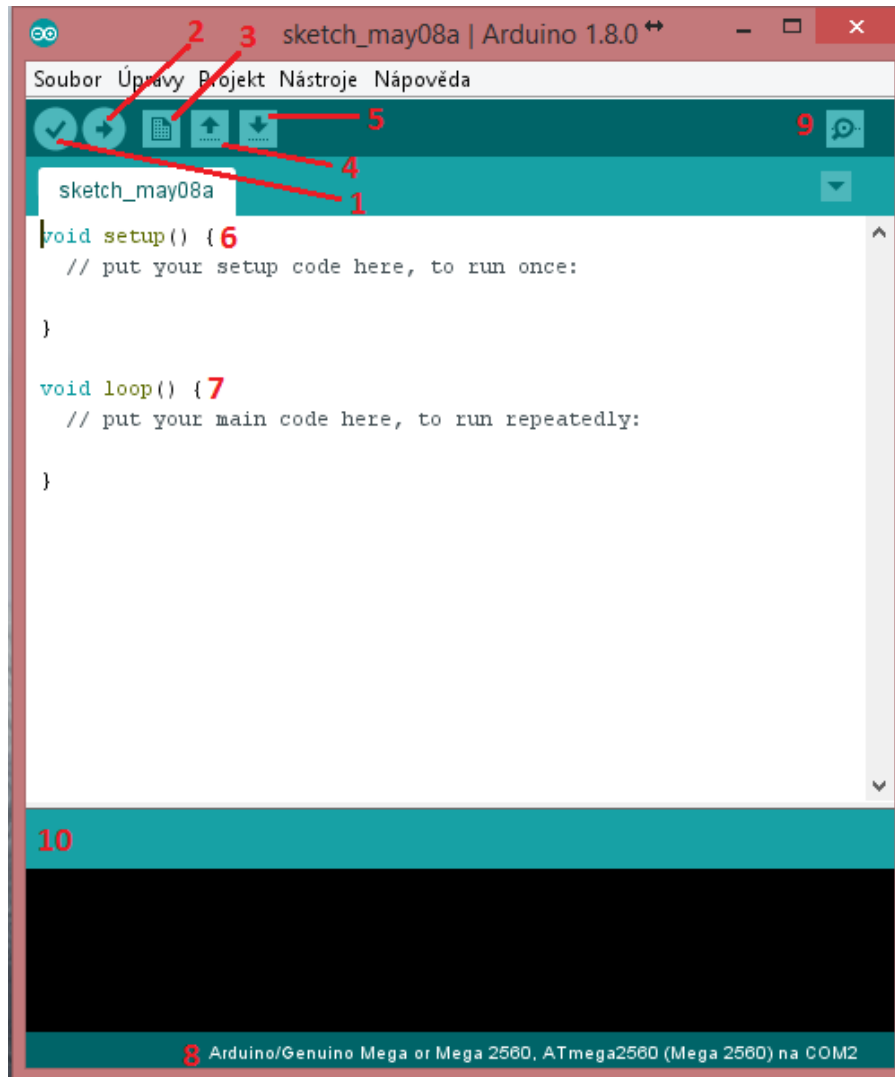
Jedná se o nástroj k hledání chyb v kódu při ladění softwaru. Program je možné krokovat, a tak chyby odstranit sledováním jednotlivých částí proměnných a samotného programu.

5.1 Prostředí Arduino IDE

Jedná se o open source vývojové prostředí navrhnuté speciálně pro Arduino. Prostředí usnadňuje psaní a nahrávání kódu přímo do Arduina. Software je dostupný pro Windows, Linux a OSX.

Prostředí nabízí krom editoru zdrojového kódu i sériovou linku, na které je možné si ověřit vstupy či výstupy sepsaného kódu v jazyce C nebo C++. Jazyk C je nejrozšířenějším programovacím jazykem. [34]

5.1.1 Popis vývojového prostředí Arduino IDE



Obrázek 14: Prostředí Arduino IDE

- První položka Soubor je prostá práce se soubory od ukládání, tisku, založení nového projektu až po vlastnosti.
- Druhá položka se nazývá Úpravy a stará se o vzhled textu. Nabízí kopírování, hledání, nastavení odsazení a další...
- V záložce Projekt najdeme především práci s knihovny
- Další položkou jsou nástroje. Zde jde hlavně o správu samotného Arduino. Lze změnit druh desky, typ procesoru nebo port na kterém se Arduino nachází.
- Poslední položkou je Nápověda. Tuto položku by měl obsahovat každý software.

1. Tlačítko slouží ke kompilaci s ověřením chyb. O kompilaci zdrojového kódu se starají nástroje Atmel.
2. Jedná se o tlačítko Nahrát. Jedním kliknutím bude kód nahrán do Arduina, která máte aktuálně připojené.
3. Kliknutím na obrázek listu, se vytvoří nový sketch. Nový projekt je defaultně pojmenován jako sketch_měsícDen. Tento název může být kdykoliv změněn.
4. Šipka nahoru nám otevře nabídku již hotových sketchů.
5. Šipka dolů zajistí uložení právě rozepsaného projektu.
6. Tato funkce v kódu musí být vždy, i když nebude použita. Bez ní by nebylo možné projekt zkompileovat. Jedná se o funkci void, která nemá návratovou hodnotu. Tato funkce se provede pouze při prvním spuštění. Jak je z názvu jasné, je vhodné zde nadefinovat a nastavit vše, čemu stačí jeden průběh. Lze zde nadefinovat I/O piny atd...
7. Stejně jako u předchozí funkce je nezbytně potřebná k funkčnosti. Při startu programu se provádí jako druhá a není opuštěna. Běží v cyklech dokola. Hlavně zde jsou volány další potřebné funkce, které má program napsané.
8. Zde je popsán typ právě nastaveného Arduina, jeho procesoru a na který port je připojeno.
9. Ikonou lupy spustíme sériovou linku
10. Zde jsou vypsány informace o právě prováděných aktivitách. V případě kliknutí na ikonu z 1. bodu, je zde napsáno *Kompiluji projekt...* V případě úspěšné kompilace je vypsána informace o dokončení a v černém poli jsou vytištěny informace o stavu paměti. V opačném případě, je vypsána chyba.

5.2 Android studio

Je oficiální vývojové prostředí pro Android od roku 2014. Je dostupný pro systémy Windows, Linux, OS X. Dříve se pro Android programovalo v prostředí Eclipse, a proto Google zajistil hladký přechod projektu do Android Studia. [27]

V Android studiu je velmi důležité věnovat pozornost již při zakládání nového projektu. Je nutné zvolit, pro jaké zařízení chceme software vyvíjet. Na výběr je ze 4 položek. První

položkou je telefon a tablet, druhou chytré hodinky, televize a poslední Android Auto. Poté je nutné u každé zvolené položky nastavit zpětnou kompatibilitu. Od které minimální distribuce Androidu žádáme funkčnost. Se staršími distribucemi se vytrácí možnosti vzhledu a funkčnosti. S novými verzemi se ztrácí pokrytí starších telefonů. Výhodou v rozhodování je, že při zvolení jednotlivých verzí se ukáže procentuální pokrytí telefonů. Toto číslo je zjišťováno z databáze aktivních uživatelů Google Play Store.

V druhém kroku zakládání projektu si volíme vzhled naší aplikace s různými vlastnostmi. Aplikace mohou mít zápatí s tlačítkem, jiné mohou obsahovat různé navigace, mapy, přihlášení nebo mohou využívat Fullscreen (celou obrazovku).

V třetím kroku zvolíme vhodný název a můžeme začít.

5.2.1 Základní prvky projektu

Manifest

Nutnou součástí kořenového adresáře je manifest. Předepsaný je i název, který musí být `AndroidManifest.xml` a ne jiný. Bez této části není aplikace spustitelná.

Slouží k jedinečnému identifikování aplikace. Popisuje jeho aktivity, služby, vše, co tvoří aplikaci včetně knihoven, tříd a další. Rovněž obsahuje práva a určuje minimální úroveň verze Androidu, kterou aplikace vyžaduje. [28]

Design activity

Design se ukládá do souboru s příponou `.xml`. Úpravy designu je možné provádět v grafickém editoru, nebo v kódu. V případě volby kódu se otevře náhled obrazovky zařízení a přeepsané změny jsou viditelné hned. Dokonce je možné otevření více typů zařízení na jednou a sledovat úpravy na různých zařízeních.

Main activity

Zde se nachází hlavní tělo kódu psané v jazyce Java.

5.2.2 Emulátor

Každý kód je možné testovat na emulátoru. Je to softwarové zařízení, které se chová jako fyzické. V Android studiu jsou některé emulátory již předvolené. Tedy je možné vyzkoušet svůj program pro Nexus S a další. Pokud není v nabídce ten správný typ telefonu, není nic

jednoduššího než si jej nakonfigurovat. Emulátor je dobrou alternativou, když není k dispozici reálné zařízení.

V případě reálného zařízení, lze testování provést propojením USB kabelu. Zařízení (např. telefon), se po připojení načte ve studiu a stačí ve výběru zvolit na místo emulovaného telefonu fyzický.

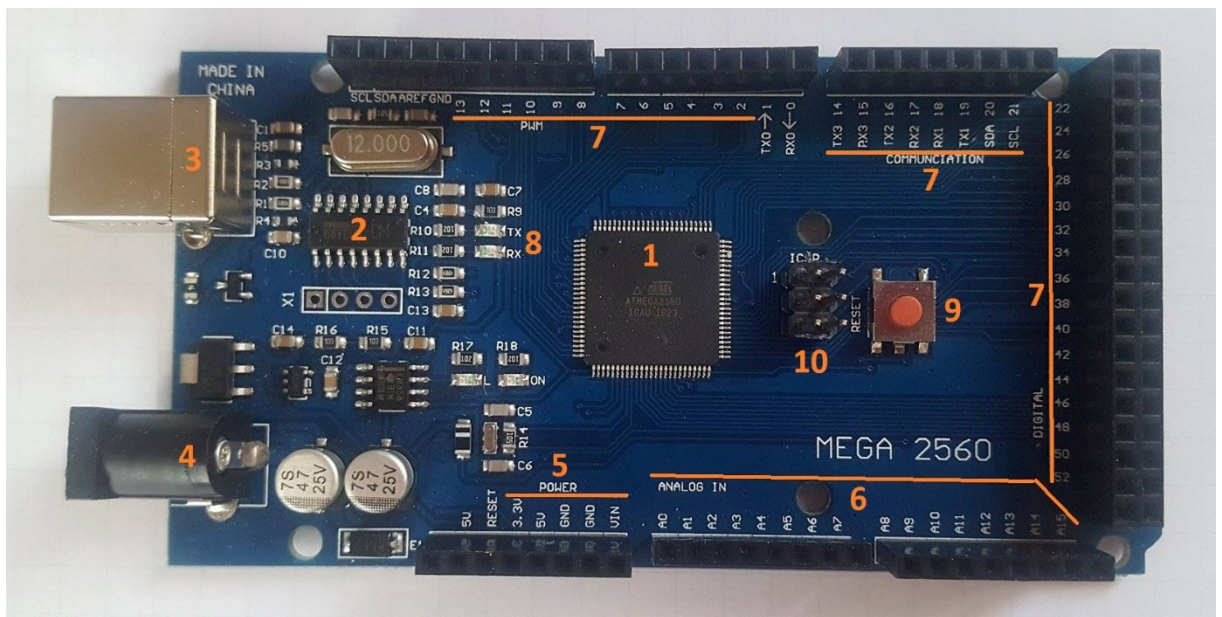
II. PRAKTICKÁ ČÁST

6 NÁVRH ŘEŠENÍ HARDWARU

Hlavním úkolem této bakalářské práce je návrh řešení parkovacích senzorů pomocí mikropočítače. Tuto práci je potřebné rozdělit na více částí. Jedná se o část hardwarovou a softwarovou. Softwarovou část se dělí na další dvě části. První část zajišťuje obsluhu mikropočítače a tím programové části Arduina. Druhá část se naopak stará o obsluhu aplikační části telefonu. Tyto jednotlivé dílčí části jsou vzájemně propojeny a tvoří výsledného parkovacího asistenta. Tato část práce je věnovaná hardwarovému návrhu řešení. K tomuto návrhu jsem volila následující součástky.

6.1 Arduino Mega2560

Jako základ mého parkovacího asistenta jsem zvolila Arduino Mega2560 R3. Přesněji jeho klon značky Funduino. Nejistila jsem, že by se nacházel zásadní rozdíl Čínského provedení, od toho originálního Italského.



Obrázek 15: Funduino Mega2560

Popis základních částí desky k obrázku 15:

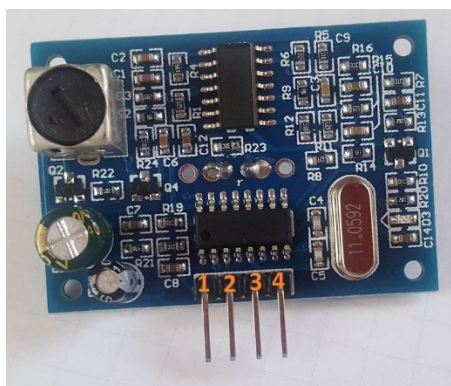
1. Mikroprocesor ATmega2560 od výrobce Atmel, 16 MHz, 8-bit, 5 V. Jedná se o hlavní mozek celé desky. Na různých typech Arduina můžeme nalézt různé typy procesorů Atmel a později dokonce Intel.
2. USB převodník, který zprostředkovává komunikaci PC a procesoru. Je to pomocník, při nahrávání programu do Arduina. Někdy je převodník umístěn v dalším

mikropočítači sloužícímu výhradně k převodu, jindy není vůbec a je nutné použití externího převodníku.

3. USB typu B. U některých jiných typů desek můžeme také najít micro USB, nebo vůbec žádný. Důvod absence USB portu bývá zapříčiněn úsporou místa.
4. Napájecí konektor uzpůsobený k připojení adaptéru. U této desky je možnost výběru napájení. Je možné napájet jak přes USB, tak přes napájecí konektor.
5. Napájecí vstupy Arduina. Zde si můžeme povšimnout možnosti napájení 3,3 V nebo 5 V. Zem je zde také víckrát, pro možnost napájení několika součástek 5 V a jiné součástky 3,3 V
6. Zde se nachází 16 analogových vstupů. Tyto vstupy jsou určené k získání analogových hodnot a také je lze využít jako digitální vstupy a výstupy.
7. Digitálních pinů na desce najdeme dohromady 54. Z toho 15 pinů podporuje PWM modulaci.
8. Jedná se o LED diody, které blikají v případě probíhající komunikace na sériové lince.
9. Resetovací tlačítko, slouží k resetu programu a jeho opětovnému spuštění.
10. Piny ICSP využívají některé Shildy, nebo je lze použít v případě absence převodníku k programování mikropočítače.

6.2 Ultrazvukové senzory JSN-SR04T

Těchto senzorů jsem použila 4 kusy. Z důvodu pokrytí celé zadní části automobilu. Jedná se o senzory uzpůsobené právě např. pro parkovací senzory. Minimální vzdálenost, při které jsou tyto senzory schopny měřit, je kolem 20 cm. Maximální měřitelná vzdálenost se blíží k 600 cm. Měrný úhel je 75°. Tyto senzory snášejí teplotu od -20°C do + 70°C a jsou voděodolné. Na obrázku můžeme vidět pouze desku senzoru. Samotný snímač se připojí z druhé strany této desky.

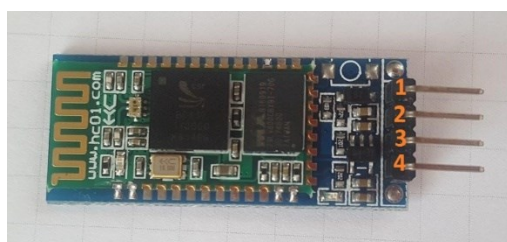


Obrázek 16: Ultrazvukový modul JSN-SR04T

1. GND – Pin slouží k zápornému proudu, tedy napájecí zem
2. Echo (TX) – Echo výstup (přijatá data)
3. Trig (RX) – Aktivační vstup (odeslaná data)
4. 5 V – Pin pozitivního napětí +5V

6.3 Bluetooth HC-05

Tato součástka využívá specifikace Bluetooth 2.0+EDR na frekvenci 2,4 GHz a slouží k bezdrátové komunikaci více zařízení. Disponuje přenosovou rychlostí 2,1 Mb/s. Dosah cca 10 m. Modul obsahuje LED indikátor s napěťovým regulátorem. Z důvodu úspory energie bych raději volila Bluetooth modul se specifikací 4.0, ale z důvodu poměr cena/výkon, jsem zvolila tuto starší verzi.

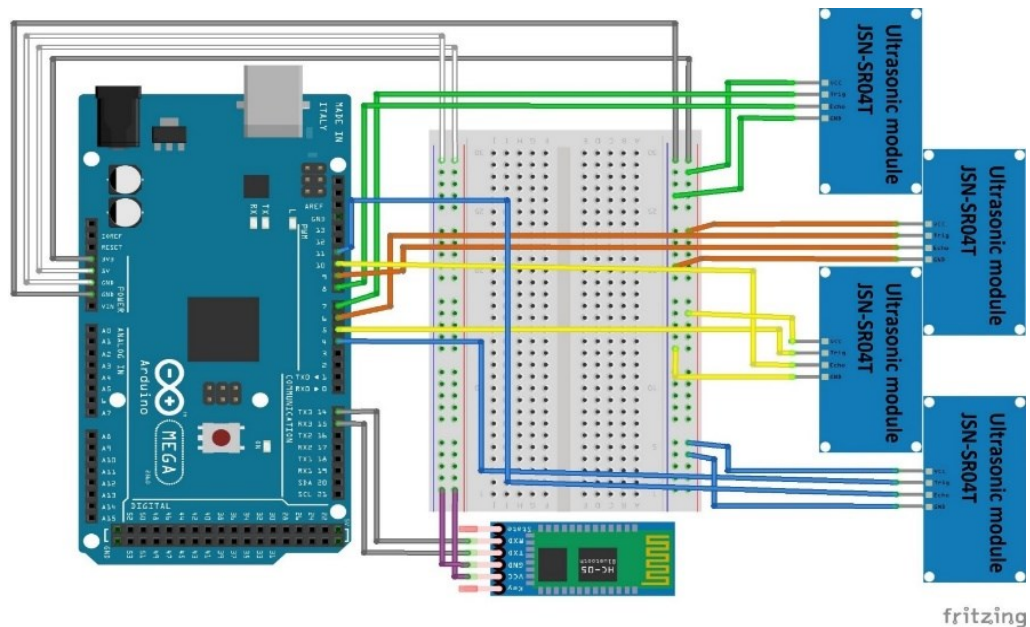


Obrázek 17: Bluetooth modul HC-05

1. VCC – Pin pozitivního napětí +3,3V
2. GND – Pin slouží k zápornému proudu, tedy uzemnění
3. TXD – Echo výstup (přijatá data)
4. TXR – Aktivační vstup (odeslaná data)

6.4 Návrh zapojení

Zde můžeme vidět propojení jednotlivých představených součástek. Na návrh propojení byl použit software Fritzing.



Obrázek 18: Návrh zapojení

Senzory jsou připojeny k napětí 5 V v pravé části nepájivého pole. Komunikační piny senzorů jsou připojeny drátky do digitálních pinů s PWM modulací na desce Arduino. Všechny echo výstupy (TX), jsou umístěny na pinech s čísly 8 až 11 a všechny aktivační vstupy (RX) se nachází na pinech 4 až 7. Bluetooth modul je připojen k napětí 3,3 V v levé části nepájivého pole a komunikační piny jsou připojeny ke komunikačním pinům TX3, RX3 na desce Arduino. Pro přehled propojení byla vytvořena následující tabulka, kde hodnoty ve sloupečku RX a TX odpovídají digitálním pinům na základní desce Arduino.

Modul	RX (Trig)	TX (Echo)
Senzor levý uprostřed (zelený)	7	8
Senzor levý (červený)	6	9
Senzor pravý uprostřed (žlutý)	5	10
Senzor pravý (Modrý)	4	11
Bluetooth modul	15	14

Tabulka 1: Přehled zapojení pinů modulů

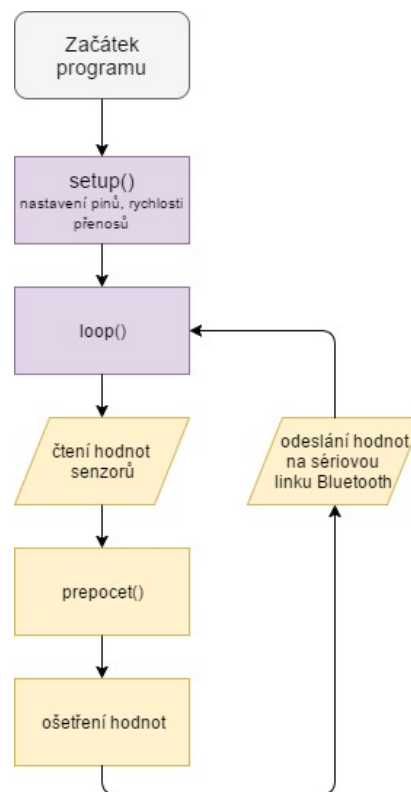
7 PROGRAMOVÁ ČÁST ARDUINA

Při návrhu programového vybavení, je třeba se zaměřit na části, které by měly být splněny ke správnému fungování. [35] Pro ujasnění struktury kódu je vhodné nakreslení stručného diagramu, splňujícího jednotlivé požadavky:

- Nastavení vstupních a výstupních pinů senzorů a bluetooth
- Čtení jednotlivých senzorů s přepočtem na cm
- Filtrace dat pro přesnější data
- Ošetření náhodných zkratů senzoru
- Odeslání na sériovou linku BT

7.1 Vývojový diagram

Vývojový diagram popisuje jednotlivé propojení částí kódu a popisuje funkci celého Arduina. Slouží k návrhu a stručnému popisu kódu.



Obrázek 20: Vývojový diagram pro Arduino

7.2 Program

Programová část Arduina je zpracována ve vývojovém prostředí Arduino IDE.

```
#include <SoftwareSerial.h>
```

Obrázek 21: Include u Arduina

Přidáním knihoven do programu získáme větší programové možnosti. Pro funkci odesílání dat pomocí Bluetooth je potřeba knihovna SoftwareSerial.h, která nám dopomůže komunikaci, kterou by bylo nutné složitě programovat.

```
SoftwareSerial BT = SoftwareSerial(14, 15);  
const int RxMiddleLeftSensor = 7;  
const int RxLeftSensor = 6;  
const int RxMiddleRightSensor = 5;  
const int RxRightSensor = 4;  
const int TxMiddleLeftSensor = 8;  
const int TxLeftSensor = 9;  
const int TxMiddleRightSensor = 10;  
const int TxRightSensor = 11;  
long LastMiddleLeftSensor = 500;  
long LastLeftSensor = 500;  
long LastRightSensor = 500;  
long LastMiddleRightSensor = 500;  
int Counter = 0;  
long AvgMiddleLeftSensor, AvgLeftSensor, AvgMiddleRightSensor, AvgRightSensor;
```

Obrázek 22: Inicializace proměnných u Arduina

Další částí kódu je deklarace jednotlivých proměnných a nastavení jejich hodnot. Jako první vidíme nastavení sériové komunikace pomocí bluetooth. Je nutné nastavení příslušných pinů, kde modul najdeme. Prvním prvkem je Rx, které se nám nachází na digitálním pinu číslo 14, druhým prvkem je tedy Tx, které najdeme pod číslem 15. Dále můžeme najít konstanty vstupů a výstupů jednotlivých senzorů. Jsou označeny slovem *const*, protože jejich hodnota je stále stejná a nebude se měnit. Ostatní proměnné jsou potřebné pro jednotlivé části programu. Všechny tyto proměnné jsou globální. To znamená, že je můžeme využít ve všech funkcích bez jejich opětovné deklarace a vymazání uložené hodnoty.

```
void setup()
{
  Serial.begin(9600);
  BT.begin(9600);
  delay(100);

  pinMode(RxMiddleLeftSensor, OUTPUT);
  pinMode(RxLeftSensor, OUTPUT);
  pinMode(RxRightSensor, OUTPUT);
  pinMode(RxMiddleRightSensor, OUTPUT);

  pinMode(TxMiddleLeftSensor, INPUT);
  pinMode(TxLeftSensor, INPUT);
  pinMode(TxRightSensor, INPUT);
  pinMode(TxMiddleRightSensor, INPUT);
}
```

Obrázek 23 Funkce setup() u Arduina

Funkce setup() je typu void, to znamená že nemá návratovou hodnotu. Funkce proběhne jednou na začátku běhu programu nebo po restartu a nastaví vše co se nachází uvnitř. Obsahem funkce je nastavení rychlosti sériové linky a rychlost odesílání na sériovou linku bluetooth. Obě tyto komunikace jsou nastaveny na rychlost 9600 bitů za sekundu. Delay je zpoždění běhu kódu o hodnotu udanou v závorkách. Tato hodnota je udána v milisekundách. Dále zde máme nastavení jednotlivých pinů senzorů. Zde místo čísel využíváme deklarované konstanty. První čtveřice s parametrem output nastavuje piny jako výstupní, druhá čtveřice input nastavuje piny jako vstupní.

```
void loop()
{
  long MiddleLeftSensor, LeftSensor, RightSensor, MiddleRightSensor;
  long cmMiddleLeftSensor, cmLeftSensor, cmRightSensor, cmMiddleRightSensor;

  digitalWrite(RxMiddleLeftSensor, LOW);
  delayMicroseconds(2);
  digitalWrite(RxMiddleLeftSensor, HIGH);
  delayMicroseconds(5);
  digitalWrite(RxMiddleLeftSensor, LOW);
  MiddleLeftSensor = pulseIn(TxMiddleLeftSensor, HIGH);
  cmMiddleLeftSensor = MicrosecondsToCm(MiddleLeftSensor);
}
```

Obrázek 24: Funkce loop()

Funkce loop() typu void je smyčka. Jedná se o stále se opakující cyklus. Tedy s každým průběhem cyklu jsou opětovně deklarovány proměnné umístěné uvnitř funkce. Funkce digitalWrite nastavuje hodnotu proudu při zápisu. V případě parametru nastaveném na high, bude na výstupu 5 V, v opačném případě LOW, bude hodnota při zápisu 0 V. PulseIn(pin, hodnota) vrací v milisekundách délku pulsu. Impuls je vyslaný do modulu senzoru a čeká na odpověď pulzu od modulu. Odpověď je v podobě mikrosekund. Toto číslo je dále přepočítáno na cm ve funkci *Prepocet(mereni)*, s která bude později vysvětlena. *DelayMicroseconds*

udává čas zpoždění v mikrosekundách. Tato část kódu je opakována pro každý senzor zvlášť. Tím získáme jednotlivá měření pro každý senzor.

```

if (Counter < 5)
{
  Counter++;
  AvgMiddleLeftSensor = AvgMiddleLeftSensor + cmMiddleLeftSensor;
  AvgRightSensor = AvgRightSensor + cmRightSensor;
  AvgMiddleRightSensor = AvgMiddleRightSensor + cmMiddleRightSensor;
  AvgLefttSensor = AvgLefttSensor + cmLeftSensor;
}
else {
  Counter = 0;
  cmMiddleLeftSensor = AvgMiddleLeftSensor / 5;
  cmRightSensor = AvgRightSensor / 5;
  cmMiddleRightSensor = AvgMiddleRightSensor / 5;
  cmLeftSensor = AvgLefttSensor / 5;
  AvgMiddleLeftSensor = 0;
  AvgRightSensor = 0;
  AvgMiddleRightSensor = 0;
  AvgLefttSensor = 0;
}

```

Obrázek 25: Filtrace dat Arduina

Pro spolehlivé měření je vhodné jednotlivá data filtrovat. Zde naměřená data filtruji tím způsobem, že vezmu 5 posledních naměřených hodnot jednotlivých senzorů a tyto hodnoty jsou průměrovány. To nám zamezuje skákání nahodilých hodnot přicházejících z modulu senzoru. Po pátém průchodu podmínky počítadlo vejde do části *else*, která počítadlo vynuluje, spočítá jednotlivé průměry a také vynuluje ostatní proměnné.

```

if (cmMiddleLeftSensor != 582) predchoziG = cmMiddleLeftSensor; else cmMiddleLeftSensor = predchoziG;
if (cmLeftSensor != 582) predchoziR = cmLeftSensor; else cmLeftSensor = predchoziR;
if (cmRightSensor != 582) predchoziY = cmRightSensor; else cmRightSensor = predchoziR;
if (cmMiddleRightSensor != 582) predchoziB = cmMiddleRightSensor; else cmMiddleRightSensor = predchoziB;

```

Obrázek 26: Ošetření dat u Arduina

Další ošetření senzorů jsem provedla tím způsobem, že jsem zakázala nastavení nejvyšší možné hodnoty. Důvod byl takový, že při jakékoliv nejistotě modulu senzoru byla jeho odpověď nejvyšší možné číslo 582. Řešení je takové, že když se naměřená hodnota rovná tomuto číslu, tak je do této proměnné vložena předchozí naměřená hodnota, která se nerovná hodnotě 582.

```

BT.print("zG");
BT.print(cmMiddleLeftSensor);
BT.print("GR");
BT.print(cmLeftSensor);
BT.print("RY");
BT.print(cmRightSensor);
BT.print("YB");
BT.print(cmMiddleRightSensor);
BT.print("Bz");
BT.println();

```

Obrázek 27: Odeslání dat na sériovou linku BT u Arduina

Zde jsou již data odesílána přes sériovou linku Bluetooth v jednom řetězci. Proto jsou zde použity pomocné znaky, které dále v programu pro aplikaci na Android pomáhají rozpoznání zasílaného řetězce. Toto je poslední část cyklické funkce loop.

```
long MicrosecondsToCm(long microseconds)
{
    return (microseconds * (340 / 2)) / 10000;
}
```

Obrázek 28: Přepočet dat u Arduina

Posledním bodem programu pro Arduino je funkce přepočtu výstupu senzorů v mikrosekundách na cm. Tato funkce má jako vstupní parametr výstup senzoru. Jedná se o funkci s návratovou hodnotou typu long, tedy výstupem funkce bude číslo typu long v podobě vzdálenosti v cm. Použitý vzorec v části return je udán výrobcem.

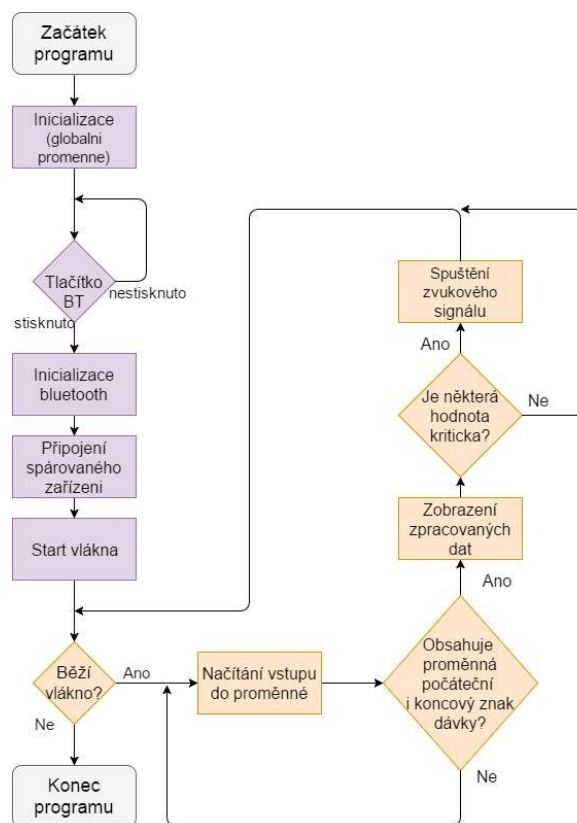
8 PROGRAM PRO ZAŘÍZENÍ S OPERAČNÍM SYSTÉMEM ANDROID

V této části se zaměřuji na aplikaci pro Android a její jednoduché ovládání. Také je nutné provést analýzu problému a přesný návrh, jakým směrem by se aplikace měla vydávat. Potřebné body programu jsou:

- Grafický návrh
- Nastavení oprávnění manifestu
- Nastavení komunikace Bluetooth
- Příjem a zpracování dat z Bluetooth
- Zobrazení konečných hodnot a jejich zobrazení

8.1 Vývojový diagram

Vývojový diagram představuje grafický pohled na problematiku kódu. Tento diagram představuje stručný popis k pochopení.



Obrázek 29: Vývojový diagram pro Android aplikaci

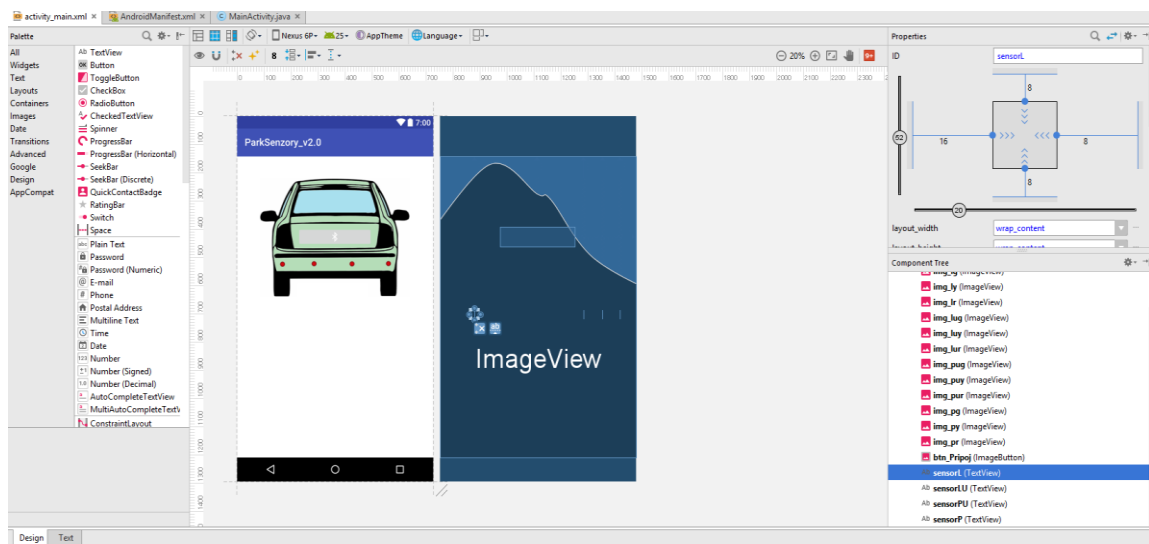
8.2 Program

Programová část aplikace pro Android je vytvořena ve vývojovém prostředí Android Studio. Tato aplikace využívá dat zasílaných z Arduina na sériovou linku bluetooth. Data jsou přebrána a dále zpracována. Výstupem je grafické rozhraní informující o aktuálním stavu automobilu za pomoci barevných kuželů a také zvukového upozornění v podobě pípnání.

8.2.1 Grafický návrh

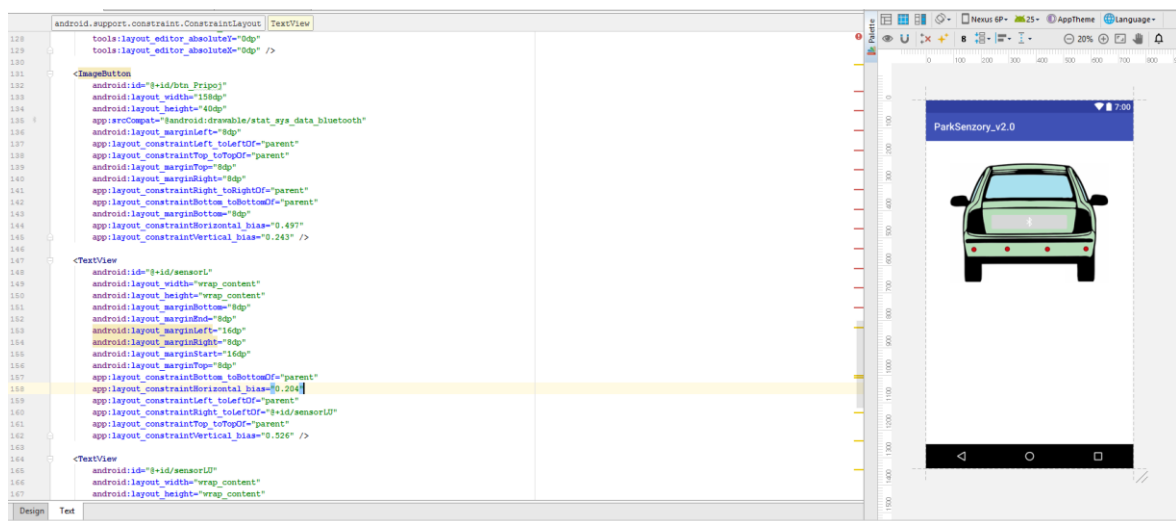
Základní vzhled aplikace se volí již při zakládání projektu, společně s minimální verzí operačního systému Android, na kterém chceme aplikaci použít. Já zvolila design pod názvem *Empty Activity*. Je to prázdná plocha s barevným pruhem, kde je umístěn název programu.

Grafický návrh lze editovat dvěma způsoby. Je možné jej editovat v kódu, či klasicky v grafickém editoru. Příprava na můj grafický návrh probíhala již mimo Android studio. Připravila jsem si jednotlivé obrázky, které jsem přidala do složky se založeným projektem a tím byly naimportovány do programu. Obrázky je vhodné přidávat v grafickém režimu. Je zde mnoho náhledů. První náhled je pohled do telefonu, jak bude aplikace vypadat. Druhý náhled je vhodný při ukotvování a pozicování jednotlivých obrázků, textových polí, tlačítek a další...



Obrázek 30: Ukázka grafického editoru Android studia

Zde máme i nabídku možností statického nastavená parametrů použitých component. Úpravy provedené v designové části se projeví i v části textové. Textová část úprav obsahuje i náhled aplikace, u které je možné okamžitě vidět provedené změny.



Obrázek 31: Ukázka textového editoru designu Android studia

Na obrázku lze vidět i ukázkou kódu, kde jednotlivé komponenty mají nastaveny vlastnosti, vzhled a umístění.

Co není v tomto režimu viditelné, jsou obrázky s barevnými kužely, znázorňujícími vzdálenost senzoru od objektu. Tyto kužely mají nastavenou viditelnost na *false*, tedy jsou neviditelné. Jejich viditelnost je podmíněna naměřenými daty v různých částech kódu. Při velmi velké vzdálenosti senzory nedekují nic. Pouze vzdálenost v centimetrech. Při bližší vzdálenosti se rozsvítí zelený kužel u příslušného senzoru. V malé vzdálenosti od objektu je rozsvícen žlutý kužel a ve velmi malé vzdálenosti od objektu je viditelný kužel červený a rychlost zvukového upozornění se zrychlí.

Tlačítko se znakem Bluetooth slouží k bezdrátovému propojení zařízení a nastartování funkčnosti senzorů. V kódu je na toto tlačítko hadler pro obsluhu události kliknutí.

8.2.2 Oprávnění

Práva jsou nastavována v manifestu. Zde se o program dozvíme nejvíce. Manifest tohoto programu obsahuje včetně základních informací pro spuštění aplikace v systému Android i přidělená uživatelská povolení. Bez těchto povolení by nebylo možné bezdrátovou komunikaci provést. Jak taková povolení vypadají můžeme vidět na následujícím obrázku.

```
<uses-feature android:name="android.hardware.bluetooth" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED" />
```

Obrázek 32: Ukázka nastavení práv pro BT v manifestu

8.2.3 Aktivity

Hlavní částí programu pro Android je čtení a následné zpracování dat vysílaných z Arduina pomocí bluetooth. K tomuto účelu slouží vlákno, které je možné vidět na následujícím obrázku. Pomocí metody *read* třídy *InputStream* jsou načtena data do pole znaků a následně jsou připojeny do globální proměnné řetězce *tempProud*. K tomu dochází, když příslušný string neobsahuje celý vysílaný řetězec označeným počátečním i koncovým znakem ‚z‘.

Nyní je k dispozici celá informace o současném stavu jednotlivých senzorů a úkolem je rozdělit jednotlivé hodnoty pro zobrazení uživateli. Pro tyto účely jsou využity značky na začátku a na konci každé naměřené hodnoty pomocí metody *substring*. Zjištěná hodnota je uložena a vypsána do textového pole v layoutu. Všechny tyto kroky jsou ošetřeny podmínkami pro případ poškozených dat, kdy hodnota vzdálenosti neodpovídá očekávanému formátu, nebo řetězec neobsahuje potřebné znaky. Na základě hodnoty je dále rozhodnuto, jaké grafické znázornění pro případnou blížící se překážku má být zobrazeno. K značení jsou použity barevné kužely. O kuželech je rozhodováno pomocí ternárních operátorů, jež umožňují zkrátit jednoduchou podmínku do jednoho řádku. Pro zápis v tomto tvaru se nejprve uvádí zkoumaný výraz následovaný otazníkem. Za otazník je připojen výraz provedený v případě splnění podmínky a výraz pro případ nesplnění podmínky je oddělen dvojtečkou. Stejně nalezení hodnot a následné zpracování je provedeno pro každý senzor. Na závěr je volána funkce *ZvukoveUpozorneni*, která nalezne minimální hodnotu vzdálenosti a následně v případě nutnosti spustí zvukové upozornění.

```
int byteCount = inputStream.available();
if (byteCount > 0)
{
    //načítání dávek dat z arduina
    byte[] rawBytes = new byte[byteCount];
    inputStream.read(rawBytes);
    final String proud=new String(rawBytes,"UTF-8"); //uložení do stringu
    handler.post(new Runnable() {
        public void run()
        {
            tempProud += proud; //jednotlivé davky jsou přidány do glob. promenne
            //pokud se najde řetězec obsahující na začátku a nakonec písmeno 'z'
            if (tempProud.length() - tempProud.replace("z", "").length() > 1) {

                //pouhá kontrola kódu, zda jako další znak přichází R
                if (tempProud.contains("R")) {
                    //z řetězce je odkrojeno vše po R
                    hodnota = tempProud.substring(tempProud.indexOf("R") + 1);
                    if (hodnota.contains("R")) { //kontroluji řetězec
                        //je vzata část řetězce dokud nenarazím na další R
                        hodnota = hodnota.substring(0, hodnota.indexOf("R"));
                        //v hodnotě by nyní mělo zůstat už jen číslo - namerena vzdálenost v cm
                        //kontrola jestli je hodnota validní...
                        //měla by mít max 3 znaky a nesmí být null
                        if (hodnota != null && hodnota.length() < 3) {
                            ///přetypování na integer
                            int ciselnaHodnota = Integer.parseInt(hodnota);
                            hodnotaR = ciselnaHodnota; //ulození do globální promenne pro ZvukoveUpozorneni
                            //Nastavení kuželových zobrazení vzdálenosti
                            ilr.setVisibility((ciselnaHodnota <= maxRedVzdálenost) ? View.VISIBLE : View.INVISIBLE);
                            ily.setVisibility((ciselnaHodnota > maxRedVzdálenost && ciselnaHodnota <= maxYellowVzdálenost) ? View.VISIBLE : View.INVISIBLE);
                            ilg.setVisibility((ciselnaHodnota > maxYellowVzdálenost && ciselnaHodnota <= maxGreenVzdálenost) ? View.VISIBLE : View.INVISIBLE);
                            sensorl.setText(hodnota.concat(" cm"));
                        }
                    }
                }
            }
        }
    });
}
```

Obrázek 33: Obsah vlákna

9 PRAKTICKÉ OVĚŘENÍ

9.1 Zprovoznění parkovacího asistenta

Pro zprovoznění asistenta je nutné připojit Arduino k síti. V průběhu testování je k Arduino připojena solární power banka, kvůli mobilitě celého zařízení. Druhou věcí je zprovoznění výstupního zařízení. Na smartphonu se zapne aplikace *ParkSenzory_v2.0*. Je dobré si také ověřit, zda je povolené bluetooth. Poté stiskem tlačítka s ikonkou BT je spuštěno párování a následné propojení zařízení. Těmito kroky se senzory automaticky spustí a započne průběh měření.

9.2 Využití výstražných prvků

K upozornění řidiče, že se blíží k překážce jsou využity vizuální a zvukové prvky. Vizuální prvky jsou v podobě barevných kuželů různých velikostí.

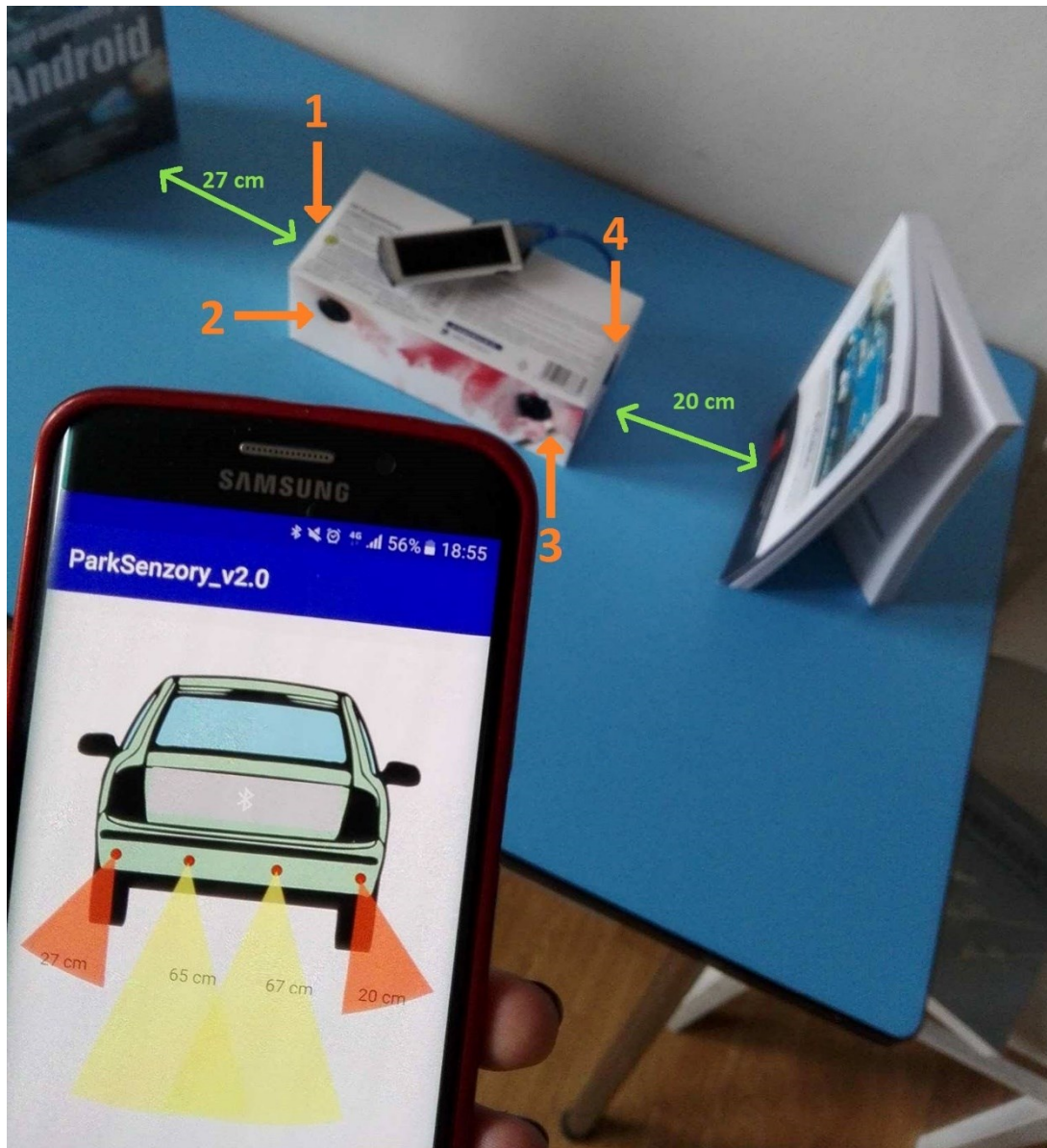
Rozsahy zobrazení kuželů a zvukových upozornění:

- Při vzdálenosti od objektu větší než 301 cm nejsou spuštěna žádná vizuální ani zvuková upozornění.
- Při vzdálenosti od objektu na 81–300 cm je zobrazen zelený kužel a zvuková signalizace není spuštěna, neboť nebezpečí srážky nehrozí.
- Při vzdálenosti od objektu 41-80 cm je zobrazen žlutý kužel s malou opakovací frekvencí zvukové signalizace. Upozornění pro řidiče, že by se měli mít na pozoru.
- Při vzdálenosti od objektu pod 40 cm je zobrazen kužel červený a frekvence opakování zvukové signalizace se zvětší. Jedná se o nejvyšší stupeň upozornění a je vhodné zastavení vozidla.

9.3 Testování

Při testování aplikace bylo nutné senzory připevnit k něčemu, kde v jejich rozptylu nebude žádný objekt a bude se s nimi snadno manipulovat, proto byla zvolena krabice, do které bylo možné všechny části uschovat. Na následující fotografii můžeme vidět průběh testu krajních senzorů, které jsou označeny číslem 1 a 4. K těmto senzorům byly přiloženy knihy na malé vzdálenosti. Výstup jednotlivých senzorů je viditelný na mobilním telefonu. Pro první senzor vykazuje vzdálenost od knihy 27 cm a čtvrtý senzor 20 cm. Tyto vzdálenosti byly reálně

o něco větší, ale vzhledem k rychlosti přibližujícího se automobilu k objektu srážky, je tahle nepřesnost užitečná. Na barvě a velikosti kuželů vidíme, že se jedná o vzdálenost kritickou pro oba senzory. Zbývající senzory označeny čísly 2 a 3 vykazují přiblížení.



Obrázek 34: Testování senzorů

Avšak při přiložení kulatého lesklého povrchu před senzor, nebylo měření přesné a neprobíhalo podle představ. Hodnoty vykazovaly značné výkyvy, protože se paprsek ultrazvukového senzoru nevracel od objektu stejnou cestou, jako k němu připutoval. Tato situace byla zmíněna již v teoretické části práce.

ZÁVĚR

Cílem bakalářské práce bylo vytvořit fungujícího parkovacího asistenta za pomoci mikropočítače platformy Arduino a telefonu s operačním systémem Android.

V teoretické části najdeme jednotlivé prvky parkovacího asistenta. Od možnosti volby senzorů přes možnosti bezdrátového přenosu po jednotlivé druhy zobrazovacích prvků. Dále jsou předvedeny možnosti mikropočítače s různým příslušenstvím. Posledním bodem teoretické části jsou vývojová prostředí, která jsou potřebná k celému vývoji. Pro naprogramování funkcí mikropočítače je použito Arduino IDE a pro mobilní aplikaci je zvoleno Android Studio.

Praktická část se nejprve věnuje návrhu hardwaru zařízení. V tomto návrhu jsou podrobně vypsány jednotlivé použité součásti s následným návrhem propojení a ukázkou již zapojeného Arduina. Druhým bodem praktické části je návrh programu obsluhující Arduina. Tímto programem jsou získávána data senzorů, která jsou následně převedena na vhodnou veličinu. Již převedená data jsou filtrována a odesílána na sériovou linku bluetooth. V další části je řešena aplikace pro operační systém Android. Tato aplikace přebere data ze sériové linky jako řetězec, který je po převzetí ihned zpracován, rozdělen na jednotlivá data senzorů a vizuálně zobrazen na displeji mobilního zařízení. Konečnou fází je praktické testování, kde je testována přesnost senzorů. Sensory se při přiblížení objektu chovaly dle očekávání, až na jeden objekt s lesklým kulatým povrchem, kde začaly prokazovat chybovost.

Tento parkovací asistent může být přínosem pro lidi, kteří potřebují pomoc při parkování svého vozu a je pro ně nežádoucí pokládat napříč autem kabely, nebo připevňovat displej výstupního zařízení. Pouze stačí vlastnit mobilní zařízení s operačním systémem Android. Dále by aplikace mohla být rozšířena pro ještě snazší použití na chytrých hodinkách (smart watch) za pomoci aplikace pro zprostředkování komunikace mobilního telefonu s hodinkami Android wear.

SEZNAM POUŽITÉ LITERATURY

- [1] Introduction to Ultrasonic Testing. *NDT Resource center* [online]. [cit. 2017-05-14]. Dostupné z: https://www.nde-ed.org/EducationResources/CommunityCollege/Ultrasonics/cc_ut_index.htm
- [2] Label Sensor Types and Technologies. *LabelSensors* [online]. St. Paul, MN, USA: Lion Precision [cit. 2017-05-14]. Dostupné z: <http://www.labelsensors.com/choose/labelSensorTypes.html>
- [3] Ultrasonic Distance Sensor: Understanding the Science Behind the Ultrasonic Distance Sensor. *Wikispaces: Arduino-info* [online]. San Francisco, USA: Tangient, 2017 [cit. 2017-05-14]. Dostupné z: <http://arduino-info.wikispaces.com/Ultrasonic+Distance+Sensor>
- [4] Mikrosenzory a mikromechanické systémy. *VUT Brno* [online]. Brno: Fond Rozvoje Vysokých Škol, 2002 [cit. 2017-05-14]. Dostupné z: http://www.umel.feec.vutbr.cz/~adamek/uceb/DATA/s_5_3.htm
- [5] Cesta k bezdrátové komunikaci – které standardy jsou pro vás nejlepší? *Pandatron* [online]. Vysoké Mýto [cit. 2017-05-14]. Dostupné z: http://pandatron.cz/?4447&cesta_k_bezdratove_komunikaci_%96_ktere_standardy_jsou_pro_vas_nejlepsi?&mobile=1
- [6] Wi-Fi standard IEEE 802.11. *Praha WiFi* [online]. Praha, c2003-2017 [cit. 2017-05-14]. Dostupné z: <http://prahawifi.cz/wifi-standard-ieee-80211>
- [7] Sítě ad-hoc. *Marigold* [online]. 2012 [cit. 2017-05-14]. Dostupné z: <https://www.marigold.cz/wifi/doku.php/adhoc>
- [8] POČÍTAČOVÉ SÍTĚ (2. část): Infrastrukturní síť. *Outech haviro* [online]. 2012 [cit. 2017-05-14]. Dostupné z: http://www.outech-havirov.cz/chmiel/files/ovt_epo_ps/me/cast2_15b_network.pdf
- [9] Základy technologie Bluetooth: původ a rozsah funkcí. *PC word* [online]. Praha 5: David Čepička, 2009 [cit. 2017-05-14]. Dostupné z: <http://pcworld.cz/hardware/Zaklady-technologie-Bluetooth-puvod-a-rozsah-funkci-6635>

- [10] Bluetooth – modrozub pod drobnohledem (vědecké okénko). *Mobilizujeme* [online]. Praha 10: David Kovařík, 2011 [cit. 2017-05-14]. Dostupné z: <https://mobilizujeme.cz/clanky/bluetooth-modrozub-pod-drobnohledem-vedecke-okenko>
- [11] Parkovací asistent Steelmate PTS410V11. *Alarmax* [online]. Brno [cit. 2017-05-14]. Dostupné z: <https://www.alarmax.cz/parkovaci-asistent-steelmate-pts410v11>
- [12] Parkovací asistent Steelmate PTS400T. *JB tuning* [online]. Brno [cit. 2017-05-14]. Dostupné z: <http://www.jb-tuning.cz/www-jb-tuning-cz/eshop/40-1-Parkovaci-systemy/0/5/7918-Parkovaci-asistent-Steelmate-PTS400T>
- [13] Parking made SAFE AND SIMPLE. Fensens [online]. San Francisco, CA, United States: Fensens, ©2017 [cit. 2017-05-14]. Dostupné z: <https://www.fensens.com/>
- [14] The World's First Wireless Parking Sensor And App. *Indiegogo* [online]. San Francisco, CA, United States: Fensens, 2016 [cit. 2017-05-14]. Dostupné z: <https://www.indiegogo.com/projects/the-world-s-first-wireless-parking-sensor-and-app-car-gadgets#/>
- [15] CO JE OPEN SOURCE SOFTWARE A HARDWARE? *Arduino* [online]. OLDŘICH HORÁČEK, 2014 [cit. 2017-05-14]. Dostupné z: <https://arduino.cz/co-je-open-source-software-a-hardware/>
- [16] Čas open-source hardwaru nadešel. *Zpravy aktualne* [online]. Lubor Benda, 2011 [cit. 2017-05-14]. Dostupné z: <https://zpravy.aktualne.cz/ekonomika/technika/cas-open-source-hardwaru-nadesel/r~i:article:707240/?redirected=1494782866>
- [17] Průvodce světem Arduina. Bučovice: Martin Stříž, 2015. ISBN 978-80-87106-90-7.
- [18] Zaciname s arduinem: 4 dil o arduinu shieldy. *PHgame* [online]. [cit. 2017-05-14]. Dostupné z: http://phgame.cz/PHGame_serialy/serialy/zaciname-s-arduinem/4-dil-o-arduinu-shieldy
- [19] Arduino Shields. SparkFun [online]. *SparkFun Electronics* [cit. 2017-05-15]. Dostupné z: <https://learn.sparkfun.com/tutorials/arduino-shields>
- [20] Kuman 16 in 1 Modules Sensor Kit Project Super Starter Kits for Arduino. Toys for Science [online]. Toys for Science, ©2017 [cit. 2017-05-15]. Dostupné z: <https://www.toysforscience.com/shop/kuman-16-in-1-modules-sensor-kit-project->

super-starter-kits-for-arduino-uno-r3-mega2560-mega328-nano-raspberry-pi-3-2-model-b-k62/

- [21] Externí moduly pro vývojové desky Arduino – přehled. *Actrl: Arduino, a vše kolem* [online]. 2016 [cit. 2017-05-15]. Dostupné z: <http://actrl.cz/blog/index.php/2016/externi-moduly-pro-vyvojove-desky-arduino-prehled/>
- [22] Real vs clone 2: Real Arduino vs. Fake/Clone Uno Board. *Dark blue bit* [online]. ©2017 [cit. 2017-05-15]. Dostupné z: <https://darkbluebit.com/arduino/real-vs-clone-2/>
- [23] Klony vs. originály. *Arduinotech* [online]. Kunčice pod Ondřejníkem, 2016 [cit. 2017-05-15]. Dostupné z: <https://www.arduinotech.cz/inpage/klony-vs-originaly/>
- [24] UJBÁNYAI, Miroslav. Programujeme pro Android. Praha: Grada, 2012. Průvodce (Grada). ISBN 978-80-247-3995-3.
- [25] Android Distribuce: Marshmallow neustále roste, ukazují nejnovější statistiky Googlu. *Android market* [online]. Antonín Vach, 2016 [cit. 2017-05-15]. Dostupné z: <http://androidmarket.cz/ruzne/android-distribuce-marshmallow-neustale-roste-ukazuji-nejnovejsi-statistiky-googlu/>
- [26] Android má konečně své oficiální vývojové prostředí. *Business world* [online]. Alexander Lichý, 2014 [cit. 2017-05-15]. Dostupné z: <http://businessworld.cz/aplikace/android-ma-konecne-sve-oficialni-vyvojove-prostredi-12038>
- [27] App Manifest. *Android: Developer* [online]. [cit. 2017-05-15]. Dostupné z: <https://developer.android.com/guide/topics/manifest/manifest-intro.html#filestruct>
- [28] Osmikanálový IR detektor přiblížení s I2C sběrnici. *Babjak* [online]. Jan Babjak, 2016 [cit. 2017-05-15]. Dostupné z: <http://ir.babjak.cz/>
- [29] Lasery kolem nás. *CEZ* [online]. Jaroslav Kusala, 2004 [cit. 2017-05-15]. Dostupné z: <https://www.cez.cz/edee/content/microsites/laser/kap3.htm>
- [30] Volkswagen Parking Sensor Parking Dynamics PD1. *Parking Dynamics* [online]. Parking Dynamics, ©2008 [cit. 2017-05-15]. Dostupné z:

<http://www.parkingdynamics.co.uk/Vehicle-Sensors/Volkswagen/Volkswagen-Parking-Sensor-Parking-Dynamics-PD1>

- [31] Accurate Laser Sensor With CMOS Image Sensor Type D Series: Longer Sensing Distance and Better Repeat Accuracy. *Optex fa* [online]. OPTEX FA [cit. 2017-05-15]. Dostupné z: https://www.optex-fa.com/products/photo_sensor/laser/d/
- [32] Infrared sensor basics | Infrared sensor network working: Infrared Sensor working. In: *RF Wireless world* [online]. RF Wireless World, ©2012 [cit. 2017-05-15]. Dostupné z: <http://www.rfwireless-world.com/Terminology/Infrared-sensor-basics-and-Infrared-sensor-working.html>
- [33] Vývojové prostředí. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-15]. Dostupné z: https://cs.wikipedia.org/wiki/V%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD
- [34] MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN, 2003. ISBN 80-730-0077-6.
- [35] MCCONNELL, Steve. Dokonalý kód: umění programování a techniky tvorby software. Vyd. 1. Brno: Computer Press, 2005, 894 s. ISBN 802510849x.
- [36] NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. Praha: BEN - technická literatura, 2005. ISBN 80-7300-141-1.
- [37] PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004. ISBN 80-7300-110-1.
- [38] SELECKÝ, Matúš. Arduino. Praha: COMPUTER PRESS - technická literatura, 2016. ISBN 978-80-251-4840-2.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PC	Personal Computer
PWM	Pulse Width Modulation
I/O	Input/Output
TTL	Transistor Transistor Logic
PCI	Peripheral Component Interconnect
ICSP	In-Circuit Serial Programming
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
DC	Direct Current
SD	Secure Digital
GMS	Google Mobile Services
GPRS	General Packet Radio Service
SMS	Short message service
RFID	Radio Frequency Identification
OLED	Organic Light Emitting Diode
TFT	Thin Film Transistor
IDE	Integrated Development Environment
Rx	Received x
Tx	Transmitted x
XML	eXtensive Markup Language
BT	Bluetooth
IR	Infrared
CMOS	Complementary Metal–Oxide–Semiconductor
WLAN	Wireless Local Area Network

USB	Universal Serial Bus
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
SSID	Service Set Identifier
ASCII	American Standard Code for Information Interchange
AC	Access Point
BSIG	Bluetooth Special Interest Group
IBM	International Business Machines
AFHSS	Adaptive Frequency Hopping
EDR	Enhanced Data Rate
WPAN	Wireless Personal Area Network
LED	Light Emitting Diode
LCD	Liquid Crystal Display
RGB	Red Green Blue
SPZ	Státní poznávací značka
PHP	Hypertext Preprocessor
SQL	Structured Query Language
HTTP	Hypertext Transfer Protocol

SEZNAM OBRÁZKŮ

Obrázek 1: Blokové schéma parkovacího systému	11
Obrázek 2: Odraz ultrazvukového senzoru [3]	12
Obrázek 3: Ukázka rozsahu elektromagnetického vlnění [30]	13
Obrázek 4: Odraz IR senzoru [32]	14
Obrázek 5 Ukázka laserového senzoru [31]	14
Obrázek 6: Steelmate PTS410V11 [11]	19
Obrázek 7: Steelmate PTS400T [12]	20
Obrázek 8: FenSens [14]	20
Obrázek 9: Ukázka propojení Shieldu [19]	26
Obrázek 10: Ukázka modulů [20]	27
Obrázek 11: Porovnání Originál vs. Klon [22]	29
Obrázek 12: Možnosti napájení	30
Obrázek 13: Statistika používání jednotlivých distribucí [25]	31
Obrázek 14: Prostředí Arduino IDE	33
Obrázek 15: Fanduino Mega2560	38
Obrázek 16: Ultrazvukový modul JSN-SR04T	40
Obrázek 17: Bluetooth modul HC-05	40
Obrázek 18: Návrh zapojení	41
Obrázek 19: Ukázka zrealizovaného propojení podle návrhu	42
Obrázek 20: Vývojový diagram pro Arduino	43
Obrázek 21: Include u Arduina	44
Obrázek 22: Inicializace proměnných u Arduina	44
Obrázek 23 Funkce setup() u Arduina	45
Obrázek 24: Funkce loop()	45
Obrázek 25: Filtrace dat Arduina	46
Obrázek 26: Ošetření dat u Arduina	46
Obrázek 27: Odeslání dat na sériovou linku BT u Arduina	46
Obrázek 28: Přepočítání dat u Arduina	47
Obrázek 29: Vývojový diagram pro Android aplikaci	48
Obrázek 30: Ukázka grafického editoru Android studia	49
Obrázek 31: Ukázka textového editoru designu Android studia	50
Obrázek 32: Ukázka nastavení práv pro BT v manifestu	50

Obrázek 33: Obsah vlákna.....	51
Obrázek 34: Testování senzorů.....	53

SEZNAM TABULEK

Tabulka 1: Přehled zapojení pinů modulů	41
---	----