

**Slovenská technická univerzita v Bratislave**  
**FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ**  
**Študijný program: Počítačové a komunikačné systémy a siete**

---

Bc. Virkler Róbert  
**PARKOVACÍ ASISTENT**  
Diplomový projekt III

Vedúci práce: Ing. Matej Jurikovič  
máj, 2012

# ANOTÁCIA

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Počítačové a komunikačné systémy a siete

Autor: Bc. Róbert Virkler

Diplomová práca: Parkovací asistent

Vedúci diplomovej práce: Ing. Matej Jurikovič

máj, 2012

Cieľom práce bolo analyzovať existujúce riešenia parkovacích asistentov, dostupné hardvérové komponenty, z ktorých pozostáva samotné riešenie. Podrobne navrhnuť programové riešenie parkovacieho asistenta. Ďalším cieľom bola implementácia a testovanie parkovacieho asistenta.

Práca analyzuje možnosti vnoreného systému od spoločnosti Funtoro, pre osobné automobily a autobusy. Analýza sa zameriava na periférne zariadenia ako sú parkovacie senzory a parkovacie kamery. Zameriava sa tiež na analýzu existujúcich riešení, ktoré sú využívané vo svetových automobilkách. Vzájomne ich porovnáva v rôznych aspektoch. Práca podrobne popisuje návrh riešenia, požiadavky systému, výber implementačného prostredia a rovnako popisuje najdôležitejšie implementované funkcie. Aplikácia pracuje vo viacerých nezávislých režimoch, spracúva informácie získané z periférnych zariadení. Režim kamery, v ktorom sa premieta video z parkovacej kamery. Režim ultrazvukových senzorov, ktoré poskytujú používateľovi rozumne spracované výstupné informácie o vzdialenosti vozidla od prekážky. Kombinovaný režim, ktorý pracuje s oboma perifériami súčasne a poskytuje potrebné informácie o dianí sa za vozidlom. Riešenie podporuje použitie štyroch nezávislých senzorov, z ktorých sú informácie spracované a poskytované používateľovi v rozumnej forme.

Práca obsahuje používateľskú príručku a technickú dokumentáciu.

# ANNOTATION

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Computer and communication systems and networks

Author: Bc. Róbert Virkler

Master`s Thesis: Parking assistant

Supervisor: Ing. Matej Jurikovič

2012, May

The aim of this paper was to analyze existing solutions of parking assistants. Paper also analyzes hardware components which uses parking assistant. Others aims was creation, implementation and testing software solution of parking assistant.

Paper analyzes the possibility of embedded system from Funtoro Company for cars and buses. The analysis describes peripheral devices like parking sensors and parking cameras. Paper analyzes existing solutions used in the automobile world too. It compares them with various aspects. The paper describes creation of solution, system requirements, and selects implementation environment too. The created solution provides work in one of more independent modes, which uses the above mentioned peripheral devices. The camera mode previews video captured from parking camera. Mode ultrasonic sensors, provides for reasonably processed output information on the distance from obstacles. The mixed mode works with both parallel peripherals and provides the necessary information. The solution provides using four ultrasonic sensors independently too. This mode is called "Súhra". It provides information about distance from each sensor and also provides the shortest distance from all sensors showed on LED`s display.

Paper contains user`s guide and technical documentation.

# Obsah

<b>0</b>	<b>ÚVOD.....</b>	<b>11</b>
0.1	ÚČEL A ROZSAH DOKUMENTU.....	11
0.2	PREHLAD DOKUMENTU .....	12
0.3	POUŽITÉ SKRATKY .....	12
0.4	POUŽITÁ NOTÁCIA .....	13
0.4.1	<i>Diagram prípadov použitia.....</i>	<i>13</i>
0.4.2	<i>Stavový diagram .....</i>	<i>13</i>
<b>1</b>	<b>ANALÝZA.....</b>	<b>14</b>
1.1	ZARIADENIA SPOLOČNOSTI FUNTORO .....	14
1.1.1	<i>Koncové riešenia využívajúce zariadenia spoločnosti Funtoro .....</i>	<i>14</i>
1.2	EXISTUJÚCE RIEŠENIA .....	18
1.2.1	<i>Parkovací asistenti Volkswagen .....</i>	<i>18</i>
1.2.2	<i>Parkovací asistenti LEXUS.....</i>	<i>20</i>
1.2.3	<i>Alternatívne riešenia parkovacích asistentov .....</i>	<i>21</i>
1.3	ULTRAZVUK .....	25
1.3.1	<i>Ultrazvuková energia.....</i>	<i>25</i>
1.3.2	<i>Vplyv prúdenia zvuku na vlastnosti ultrazvukových senzorov.....</i>	<i>26</i>
1.3.3	<i>Princípy ultrazvukového merania.....</i>	<i>26</i>
1.3.4	<i>Vyžarovací diagram ultrazvukového senzora .....</i>	<i>28</i>
1.4	ULTRAZVUKOVÉ SENZORY.....	29
1.4.1	<i>Programovateľné ultrazvukové senzory P42.....</i>	<i>29</i>
1.4.2	<i>Microsonic HPS 340.....</i>	<i>30</i>
1.4.3	<i>Ultrazvukový senzor SRF08.....</i>	<i>30</i>
1.4.4	<i>Ultrazvukový senzor SRF02.....</i>	<i>31</i>
1.4.5	<i>Porovnanie analyzovaných ultrazvukových senzorov.....</i>	<i>32</i>
1.5	KOMUNIKAČNÉ ROZHRANIA .....	33
1.5.1	<i>RS-232.....</i>	<i>33</i>
1.5.2	<i>Rozhranie I<sup>2</sup>C.....</i>	<i>34</i>
1.6	ZÁVER ANALÝZY .....	36

<b>2</b>	<b>OPIS RIEŠENIA.....</b>	<b>38</b>
2.1	ŠPECIFIKÁCIA RIEŠENIA .....	38
2.1.1	<i>Funkcionálne požiadavky</i> .....	38
2.1.2	<i>Nefunkcionálne požiadavky</i> .....	39
2.1.3	<i>Diagram prípadov použitia pre kameru a senzor</i> .....	39
2.1.4	<i>Diagram prípadov použitia pre štyri senzory</i> .....	41
2.2	VÝBER HARDVÉRU.....	43
2.3	VÝBER IMPLEMENTAČNÉHO PROSTREDIA .....	43
2.3.1	<i>Java</i> .....	44
2.3.2	<i>C++</i> .....	44
2.3.3	<i>Platforma .NET</i> .....	44
2.3.4	<i>C#</i> .....	44
2.4	NÁVRH ARCHITEKTÚRY SYSTÉMU.....	45
2.4.1	<i>Štruktúra systému</i> .....	45
2.4.2	<i>Návrh aplikácie pre stolový počítač</i> .....	47
2.4.3	<i>Návrh aplikácie pre server</i> .....	58
2.5	IMPLEMENTÁCIA SYSTÉMU .....	66
2.5.1	<i>Implementácia funkcií kamery</i> .....	66
2.5.2	<i>Štruktúra systému</i> .....	67
2.5.3	<i>Funkcie hlavného okna</i> .....	67
2.5.4	<i>Funkcie režimu kamery</i> .....	70
2.5.5	<i>Funkcie režimu ultrazvukového senzora</i> .....	70
2.5.6	<i>Trieda senzor</i> .....	71
2.5.7	<i>Funkcie duálneho režimu</i> .....	71
2.5.8	<i>Funkcie režimu súhra</i> .....	72
2.6	OVERENIE RIEŠENIA .....	72
<b>3</b>	<b>ZÁVER .....</b>	<b>75</b>
3.1	MOŽNOSTI VYLEPŠENIA SYSTÉMU .....	76
<b>4</b>	<b>TECHNICKÁ DOKUMENTÁCIA.....</b>	<b>77</b>
4.1	DOKUMENTÁCIA K IMPLEMENTÁCII.....	77
4.1.1	<i>Hlavná trieda public partial class Form1 : Form</i> .....	77
4.1.2	<i>Funkcia public Form1()</i> .....	79

4.1.3	<i>Funkcia private void senzor_Click(object sender, EventArgs e).....</i>	80
4.1.4	<i>Funkcia private void setupCommPort(String comport) .....</i>	81
4.1.5	<i>Funkcia private void vyhodnot_senzor(double x, double set) .....</i>	81
4.1.6	<i>Funkcia senzor vrat_min(double x1, double x2, double x3, double x4).....</i>	84
4.1.7	<i>Funkcia private void snimac_Tick(object sender, EventArgs e) .....</i>	85
4.1.8	<i>Funkcia private double generator(int min, int max).....</i>	86
4.1.9	<i>Funkcia private void back_sens_Click(object sender, EventArgs e).....</i>	86
4.1.10	<i>Funkcia private void suhra_Click(object sender, EventArgs e).....</i>	87
4.1.11	<i>Funkcia private void camera_Click(object sender, EventArgs e).....</i>	88
4.1.12	<i>Funkcia private void dual_Click(object sender, EventArgs e).....</i>	88
4.1.13	<i>Funkcie private void mute_on_Click(object sender, EventArgs e) a private void mute_off_Click(object sender, EventArgs e).....</i>	89
4.1.14	<i>Funkcia public void start_cam_cam().....</i>	89
4.1.15	<i>Funkcia public void Obraz_cam(object sender, NewFrameEventArgs eventArgs).....</i>	90
4.1.16	<i>Funkcia public void Exit_zdroj().....</i>	90
4.1.17	<i>Funkcia private void stopky_timer_Tick(object sender, EventArgs e).....</i>	90
4.2	<b>POUŽÍVATEĽSKÁ PRÍRUČKA .....</b>	<b>91</b>
4.2.1	<i>Minimálne požiadavky .....</i>	91
4.2.2	<i>Inštalácia .....</i>	91
4.2.3	<i>Používateľské rozhranie.....</i>	92
<b>5</b>	<b>LITERATÚRA.....</b>	<b>100</b>
	<b>PRÍLOHA A : OBSAH ELEKTRONICKÉHO MÉDIA.....</b>	<b>103</b>
	<b>PRÍLOHA B : ELEKTRONICKÉ MÉDIUM .....</b>	<b>104</b>

## 0 Úvod

Kapitola popisuje zadanie projektu, účel a rozsah dokumentu. Rovnako popisuje použité skratky a výrazy a použitú notáciu pre jednotlivé diagramy.

### 0.1 Účel a rozsah dokumentu

Tento dokument vznikol ako dokumentácia k Diplomovej práci. Automobil je veľkým pomocníkom každého človeka, ktorý potrebuje prekonávať dlhé vzdialenosti v krátkom čase. Procesy spojené s riadením automobilu môžeme rozdeliť na jednoduchšie a rovnako tak aj na zložitejšie činnosti. Parkovanie môžeme považovať za jednu z najzložitejších činností. Náročné je hlavne v oblastiach, kde je počet automobilov na rozlohu oblasti veľký. Vtedy sa najčastejšie jedná o parkovanie v úzkych priestoroch, ktoré sú ťažko dostupné. Cieľom tejto práce bolo navrhnúť, implementovať a otestovať riešenie parkovacieho asistenta, ktorý zjednoduší samotný proces parkovania.

Riešenie parkovacieho asistenta využíva zariadenia od spoločnosti Funtoro[9]. Práca tiež analyzuje zariadenia, ktoré sú vhodné pre zostavenie parkovacieho asistenta. Jedná sa o server, LCD dotykový displej, parkovaciu kameru a parkovacie ultrazvukové senzory.

Na trhu je množstvo riešení parkovacích asistentov, počnúc továrenskými riešeniami alebo alternatívnymi riešeniami, ktoré sú vhodné do akéhokoľvek automobilu. Práca analyzuje továrenské riešenia inštalované do automobilov spoločnosti Volkswagen[7] a Lexus[2]. Jedná sa o riešenia, ktoré sú finančne náročnejšie. Analyzované sú však aj lacnejšie komerčné riešenia.

Práca podrobne popisuje návrh riešenia, kladie dôraz na funkcionálne ako aj na nefunkcionálne požiadavky systému. Porovnáva implementačné prostredia a programovacie jazyky. Popisuje podrobný návrh jednotlivých častí systému, ktorý presne definuje štruktúru a správanie sa systému. Práca podrobne popisuje najdôležitejšie implementované funkcie.

Súčasťou práce je technická dokumentácia s používateľskou príručkou. Dokument je určený najmä pre odbornú verejnosť.

## 0.2 *Prehľad dokumentu*

Kapitola 0: Úvod – kapitola popisuje účel a rozsah dokumentu, prehľad dokumentu. Definuje skratky a výrazy použité v dokumente ako aj notácie pre diagramy.

Kapitola 1: Analýza – kapitola analyzuje existujúce riešenia parkovacích asistentov, hardvér, ktorý môže byť použitý v zapojení parkovacieho asistenta.

Kapitola 2: Opis návrhu riešenia – kapitola opisuje výber implementačného prostredia, funkcionálne a nefunkcionálne požiadavky, podrobný návrh riešenia, implementáciu systému, overenie riešenia.

Kapitola 3: Záver – kapitola hodnotí dosiahnuté výsledky, funkcionálnosť aplikácie.

Kapitola 4: Technická dokumentácia – kapitola sa zaoberá opisom základných funkcií, ktoré boli implementované, časti kódov, premenné, rovnako však poskytuje používateľskú príručku.

Kapitola 5: Použitá literatúra

## 0.3 *Použité skratky*

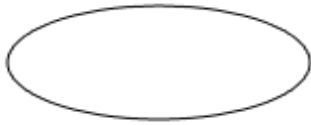
ACK (Acknowledgement)	- potvrdenie
CS (Chip Select)	- výber podriadeného zariadenia
DCE (Data Communications Equipment)	- dátové komunikačné vybavenie
DVB-T (Digital Video Broadcasting - Terrestrial)	- digitálne video vysielanie - pozemné
DVD (Digital Versatile Disc)	- digitálny univerzálny disk
DTE (Data Terminal Equipment)	- dátové koncové vybavenie
GB (Giga Bajt)	- giga bajt
GPS (Global Positioning System)	- celkový pozičný systém
GPRS (General Packet Radio Service)	- všeobecný balík rádiových služieb
I <sup>2</sup> C (Inter-Integrated Circuit)	- Integrované komunikačné rozhranie
LCD (Liquid Crystal Display)	- obrazovka s tekutých kryštálov
MOD (Media On Demand)	- médiá na požiadanie
MSI (Micro-Star Int'lCo.)	- MSI spoločnosť
PC (Personal Computer)	- Osobný počítač
SCL (Clock Line)	- hodinová linka
SDA (Data Line)	- dátová linka
SD (Secure Digital)	- digitálna bezpečnosť
TTL (Transistor Transistor Logic)	- tranzistor, tranzistor, logika
UART (Universal Asynchronous receiver/transmitter)	- univerzálny, asynchrónny prijímač/vysielač
USB (Universal Serial Bus)	- univerzálna sériová zbernica
VGA (Video Graphic Array)	- grafická karta
WIFI (Wireless Fidelity)	- bezdrôtové pripojenie



## 0.4 Použitá notácia

Notácia použitá pre stavové diagramy a pre diagramy prípadov použitia.

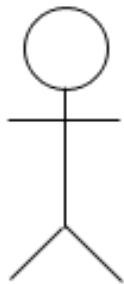
### 0.4.1 Diagram prípadov použitia



Prípád použitia



Priradenie prípadu použitia k hráčovi



Hráč

### 0.4.2 Stavový diagram



Začiatok, koniec režimu



Proces v diagrame



Podproces v diagrame



Rozhodovanie v diagrame



Reprezentácia senzora v diagrame



Priradenie v diagrame

# 1 Analýza

Analýza problému vychádza zo zadania, ktoré je špecifikované v nasledujúcich riadkoch. Analyzujte možnosti vnoreného systému od spoločnosti Funtoro[9], pre osobné automobily a autobusy. Automobilový priemysel napreduje a rovnako aj požiadavky zákazníkov sú čoraz vyššie. Smerovanie spoločností sa zameriava na uľahčenie činností spojených s vedením motorových vozidiel. Pri analýze sa zamerajte na periférne zariadenia ako parkovacia kamera a parkovacie senzory. Porovnajte tieto systémy s existujúcimi zariadeniami, v čom sú lepšie a v čom zaostávajú za konkurenciou. Navrhnite softvérové riešenie, ktoré bude v maximálnej miere spolupracovať a spracovávať informácie z týchto periférnych zariadení. Poskytnite podrobné a prakticky využiteľné informácie, ktoré môžu byť vhodné pre ďalšie spracovanie. Systém bude spracovávať a zobrazovať video prenos z parkovacej kamery a parkovacie senzory budú poskytovať informácie o vzdialenosti automobilu od prekážky. Riešenie bude poskytovať prácu vo viacerých profiloch s využitím oboch periférií alebo každej osobitne. Navrhnuté riešenie potom prakticky overte v reálnych podmienkach. Snažte sa ho doladiť do reálne využiteľnej podoby.

Kapitola analyzuje existujúce produkty od spoločnosti Funtoro[9], spôsoby zapojenia v reálnych podmienkach. Rovnako analyzuje existujúce riešenia parkovacích asistentov, továrenské riešenia ako aj alternatívne riešenia, ktoré je možné inštalovať do akéhokoľvek automobilu.

## ***1.1 Zariadenia spoločnosti Funtoro***

Spoločnosť Funtoro[9] bola založená v roku 2005. Zaoberá sa technológiami využívanými v automobilovom priemysle. Ponúka multimediálne riešenia pre autobusy, automobily a iné dopravné prostriedky.

V roku 2007 sa súčasťou MSI[12] a tak rozšírila svoje portfólio o rôzne druhy produktov a to nielen multimediálneho zamerania.

Navrhovaný systém využíva komponenty spoločnosti Funtoro. Medzi také komponenty patrí server a LCD dotykový monitor.

### **1.1.1 Koncové riešenia využívajúce zariadenia spoločnosti Funtoro**

Podkapitola popisuje dostupné produkty spoločnosti Funtoro využívané v zapojeniach pre automobily alebo autobusy. Rovnako popisuje spôsoby zapojenia jednotlivých produktov pre využitie v rôznych situáciách.

### ***Digitálny MOD server***

Funtoro MOD systém umožňuje používanie multimediálneho obsahu až pre 54 nezávislých cestujúcich. Tento systém môže byť použitý v autobusoch, rovnako však aj vo vlakoch, vo vozidlách mestskej hromadnej dopravy a iných, kde sa prepravuje väčšie množstvo cestujúcich.

Systém umožňuje cestujúcim prezerat' si filmy, rôzne videá, počúvať hudbu, rádio a podobne. Prezeranie multimediálneho obsahu je nezávislé a vzťahuje sa na každý monitor osobitne. Veľká výhoda spočíva v ovládaní systému, ktoré je nezávislé a nie je nutné sa prispôbovať ostatným používateľom. Počas prezerania obsahu si používateľ môže obsah posúvať dopredu, zastaviť si ho, alebo ho posunúť späť. Prezeranie prebieha na multimediálnom dotykovom displeji, ktorý poskytuje používateľovi ľahké a vkusné ovládanie.

Rovnako je možné sledovať televízne vysielanie z externého zdroju vďaka DVB-T tuneru, sledovanie cesty z čelnej kamery vozidla, ako aj sledovanie GPS zariadenia vodiča.

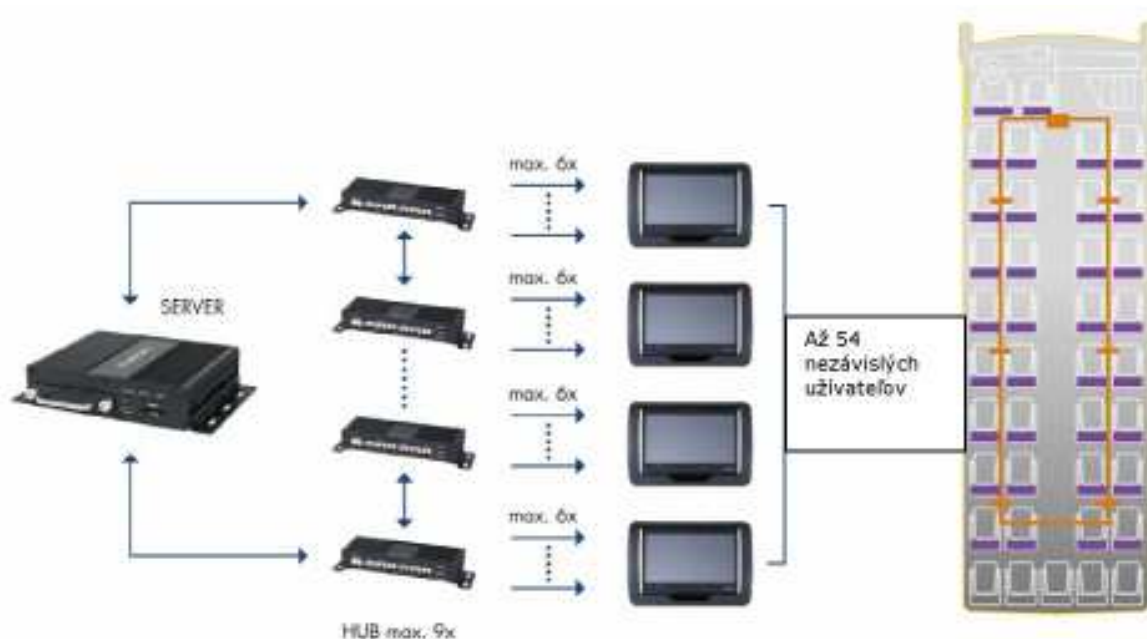
Na displeji je možné vykonať nastavenia a následne sledovať zmenu vplyvom týchto aplikovaných nastavení. Ovládanie je intuitívne dotykmi na obrazovku a výberom vhodnej kategórie. V prípade nejasností v ovládaní je dostupný prehľadný návod.

V prípade vypnutia systému a jeho opätovnom zapnutí si systém pamätá predošlú konfiguráciu ako aj poslednú konfiguráciu. Z tohto dôvodu nie je nutné opätovné nastavenie aplikácie, filmu, pesničky na daný čas.

V prípade, že systém narazí na závažnú chybu, používateľ nie je obťažovaný chybovými hláseniami, ale sám sa reštartuje a opätovne sa inicializuje na mieste, kde bola chyba zachytená.

Systém je zabezpečený sekundárnym zavádzačom, ktorý sa aktivuje v prípade zlyhania primárneho zavádzania systému. Zabezpečuje tým bezproblémový štart systému aj v prípade zlyhania prvého štartovania.

Príklad zapojenia systému je zobrazený na *obrázku 1*. Jedná sa o zapojenie pre autobusy, vlaky, mestskú hromadnú dopravu, ktoré pozostáva z jedného servera, rady prepájačov a dotykových displejov.



**Obr. 1** Základná architektúra s digitálnym MOD serverom[9]

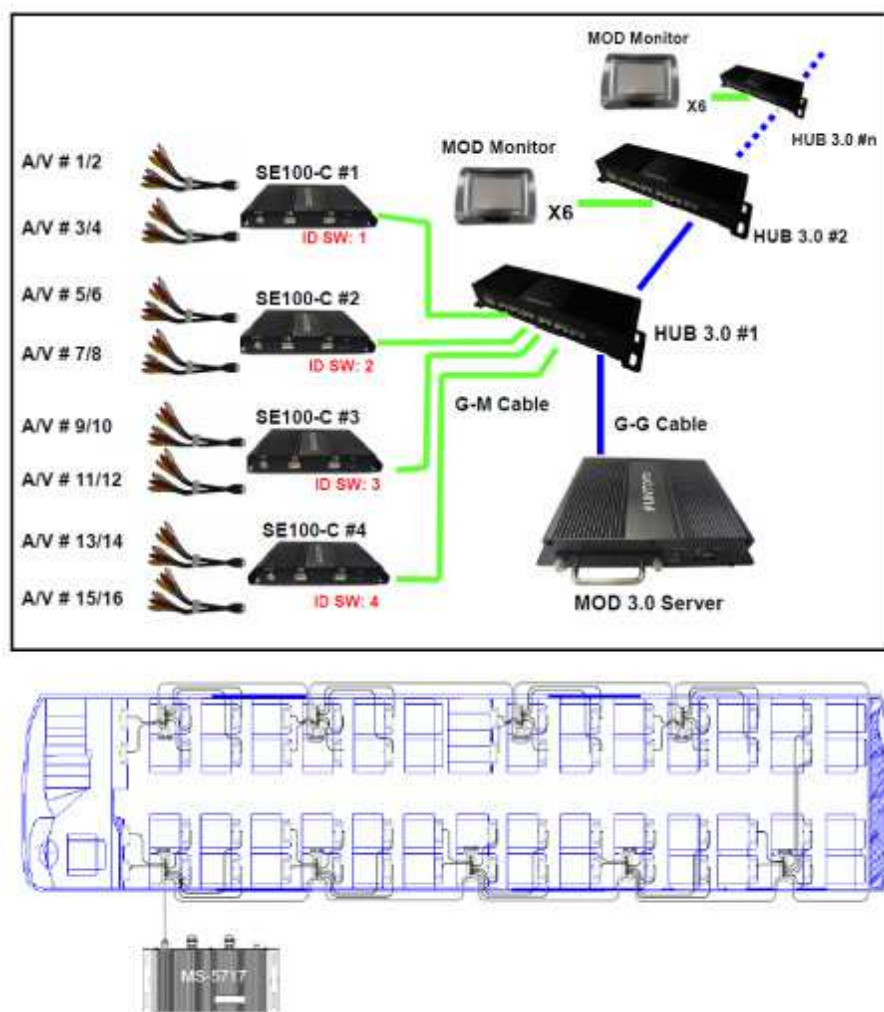
### *Vysielač server*

Riešenie s vysielačím serverom poskytuje podobné vlastnosti nezávislého sledovania multimediálneho obsahu ako predošlé riešenie využívajúce digitálny MOD server. Poskytuje 8 až 32 kanálov, na ktorých môže používateľ sledovať filmy, televízne vysielanie alebo počúvať hudbu.

Základné vybavenie servera poskytuje 8 audio kanálov, 8 video kanálov a 4 nezávislé vstupy pre externé zariadenia ako DVD prehrávač, kamery, TV vysielanie a podobne. Multimediálny obsah je štruktúrovane uložený na servery na SD karte o kapacite 8 až 32 GB.

Vďaka svojej univerzálnosti je možné server rozšíriť o ďalších dvanásť vysielačích kanálov. Je však možné pripojiť akýkoľvek audio alebo video zdroj a tak systém rozšíriť o ďalšie možnosti.

Obraz je možné sledovať na dotykovom displeji rovnako ako v riešení s digitálnym MOD serverom. Na *obrázku 2* je zobrazené zapojenie architektúry, ktorá poskytuje pripojenie ľubovoľného externého zdroja audio/video signálu a následnú interpretáciu na LCD monitor.



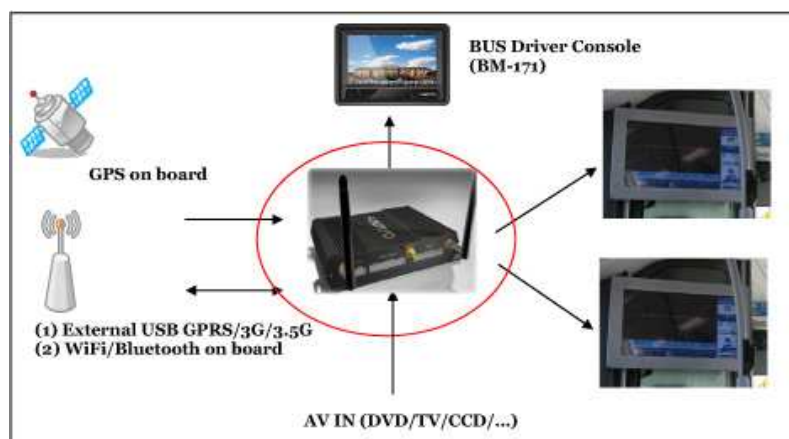
**Obr.2** Základná architektúra s vysielačím MOD serverom[9]

### *Telematický systém*

System, ktorý v sebe spája možnosti zobrazenia reklamných blokov, aktuálnych informácií a zábavy. Využíva štandardné centrálné monitory vo vozidle. Vysielaný obsah riadi vodič alebo sprievodca na svojom dotykovom monitore. Multimediálny obsah – video, hudba, obrázky, je uložený na SD karte a navyše je k telematickému systému možné pripojiť ďalší audio - video zdroj (DVD prehrávač, TV tuner, kamery a ďalšie).

Základnú verziu riešenia je možné doplniť o GPS navigáciu, Bluetooth a Wifi modul alebo externý USB GPRS modul.

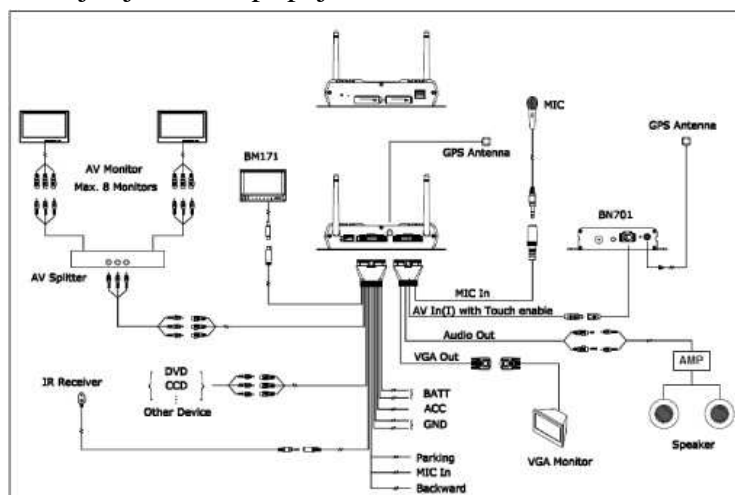
Zobrazovanie informácií prebieha na 7 palcovom dotykovom monitore, ktorý je však ovládaný len vodičom alebo sprievodcom. Na obrázku 3 je zobrazená základná architektúra a pripojenie monitorov a GPS, Wifi, GPRS k telematickému boxu.



**Obr. 3** Základná architektúra pripojenia monitorov k boxu [9]

Na *obrázku 4* je zobrazená schéma pripojenia externých audio a video zdrojov.

Rovnako tak zobrazuje aj káblové pripojenie LCD monitorov alebo VGA monitorov.



**Obr. 4** Schéma pripojenia jednotlivých multimediálnych prvkov k telematickému boxu[9]

## 1.2 Existujúce riešenia

Kapitola existujúcich riešení opisuje existujúce systémy parkovacích asistentov rôznych spoločností, ktoré sú bežne dostupné a využívané. Poskytuje prehľad a porovnanie jednotlivých produktov. Rovnako ich však porovnáva s navrhovanými riešeniami, ktoré je výsledkom tejto práce.

### 1.2.1 Parkovací asistenti Volkswagen

Spoločnosť Volkswagen[7] využíva riešenia parkovacích asistentov od americkej spoločnosti Valeo[28]. Ponúka riešenia parkovacích asistentov rôznych typov v závislosti od ceny. Najjednoduchšie riešenia pozostávajú z ultrazvukových senzorov inštalovaných v zadnej časti vozidla. Jedná sa o šesť ultrazvukových senzorov. Informácie z týchto

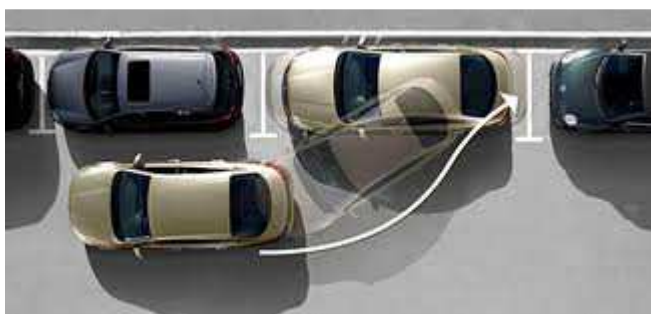
senzorov sa spracované zobrazujú na multifunkčnom monitore, zobrazené na *obrázku 5*. Poskytujú informácie o vzdialenosti prekážky od vozidla.



**Obr. 5** Multifunkčný displej [6]

Lepšia výbava pozostáva z parkovacieho asistenta, ktorý okrem senzorov poskytuje parkovacie kamery umiestnené v zadnej časti vozidla, rovnako aj v spätných zrkadlách automobilu. Parkovanie za pomoci parkovacieho asistenta je zobrazené na *obrázku 6*. Informácie sú prenášané na multifunkčný monitor, ktorý poskytuje možnosť prepínania medzi pohľadmi z jednotlivých kamier ako aj kombinované informácie z ultrazvukových senzorov a kamier.

Parkovacích asistentov môžeme rozdeliť do dvoch kategórií z pohľadu obsluhy parkovania. Jedná sa o manuálne parkovanie a samočinné parkovanie. Firma Volkswagen poskytuje aj riešenia automatického parkovacieho asistenta. Automatické parkovanie automobilu Volkswagen Tiguan[13] je zobrazené na *obrázku 7*.



**Obr. 6** Parkovanie s využitím zadných senzorov a kamery [7]

Samočinné parkovanie je podrobne popísané v nasledujúcich riadkoch. Systém v automobile Tiguan umožňuje nájsť voľné miesto a zaparkovať doň pozdĺžne. Samočinné parkovanie sa aktivuje tlačidlom, umiestneným pri radiacej páke. Sensory odmerajú veľkosť voľného miesta a vyhodnotia, či sa tam vozidlo zmestí. Vodič je po nájdení vhodného miesta upozornený na multifunkčnom monitore na palubnej doske. Po zaradení



spiatocky systém preberie riadenie, vodič používa iba brzdový a spojkový pedál. Počas celého manévru vodič kontroluje okolie, pomáhajú mu pri tom aj zvukové signály parkovacích senzorov. Vodič má pri parkovaní ruky pokojne na kolenách, vozidlo pracuje s volantom samostatne. Na záver parkovania je nutné zaradiť prvý rýchlostný stupeň a vozidlo sa dorovná s chirurgickou presnosťou k okraju obrubníka. Výhodou spomínaného systému je precíznosť parkovania, kedy je presnosť až na 5cm [1]. Mechanizmus je možné kedykoľvek vypnúť a vodič môže zaparkovať manuálne.



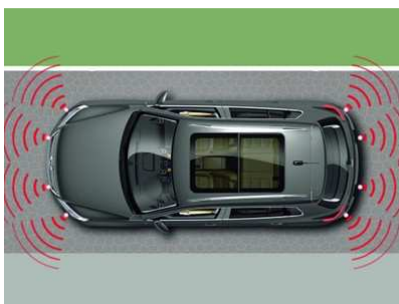
**Obr. 7** Automatické parkovanie [8]

### **1.2.2 Parkovací asistenti LEXUS**

Parkovací asistenti v autách spoločnosti Lexus[2] využívajú podobné riešenia ako automobily spoločnosti Volkswagen. Poskytujú základné typy parkovacích asistentov, ktoré pracujú len s ultrazvukovými senzormi. Vo svojej ponuke však poskytujú možnosť samočinného parkovania.

Samočinné parkovanie funguje principiálne rovnako ako v modeloch spoločnosti Volkswagen. V prípade, že chce vodič zaparkovať, spomalí na rýchlosť 50 km/h. Automobil začne snímať veľkosti parkovacích miest, ktoré by boli vhodné. V prípade, že nájde vyhovujúce parkovacie miesto, vodič zastaví, zaradiť spiatocku a tým dá pokyn pre samočinné parkovanie. Vodič sa nestará o nič viac, len reguluje rýchlosť vozidla plynovým pedálom. Rozdiel oproti riešeniam spoločnosti Volkswagen spočíva v tom, že nie je nutné aktivovať mechanizmus samočinného parkovania. Riešenie je porovnateľné s riešením spoločnosti Volkswagen. Parkovací asistenti sú pre automobily Lexus dodávané firmou Toyota[2]. Pôvodne mali byť určené len pre japonský trh, ale prenikli aj na svetový trh. Na *obrázku 8* je zobrazený automobil pri samočinnom automatickom parkovaní.





**Obr. 8** Riešenie parkovania pre Lexus[5]

Predchádzajúce podkapitoly *1.2.1 Parkovací asistenti Volkswagen* a *1.2.2 Parkovací asistenti Lexus* popisujú továrenské riešenia, ktoré sú montované do automobilov. Tieto riešenia sú finančne náročnejšie o rádovo stovky eur. Analyzované boli dve továrenské riešenia, ktoré sa značne podobajú a sú založené na podobných princípoch aj keď sa jedná o výrobky od dvoch nezávislých spoločností. Z tohto dôvodu nie je nutná ďalšia analýza továrenských riešení. Obe riešenia využívajú multifunkčné zobrazovacie monitory a poskytujú rôzne režimy pre prácu s periférnymi zariadeniami. Existujú však aj riešenia, ktoré nie sú finančne tak náročné, ceny sa pohybujú v desiatkach eur. Prehľad jednotlivých produktov je zobrazený v *tabuľke 2* v *podkapitole 1.2.3*. Jedná sa o základné typy parkovacích asistentov, ktoré pozostávajú z ultrazvukových senzorov a zobrazovacieho elementu, na ktorom sú zobrazované informácie získané z týchto senzorov.

### **1.2.3 Alternatívne riešenia parkovacích asistentov**

Podkapitola opisuje riešenia komerčných parkovacích asistentov, ktoré je možné inštalovať do akéhokoľvek automobilu.

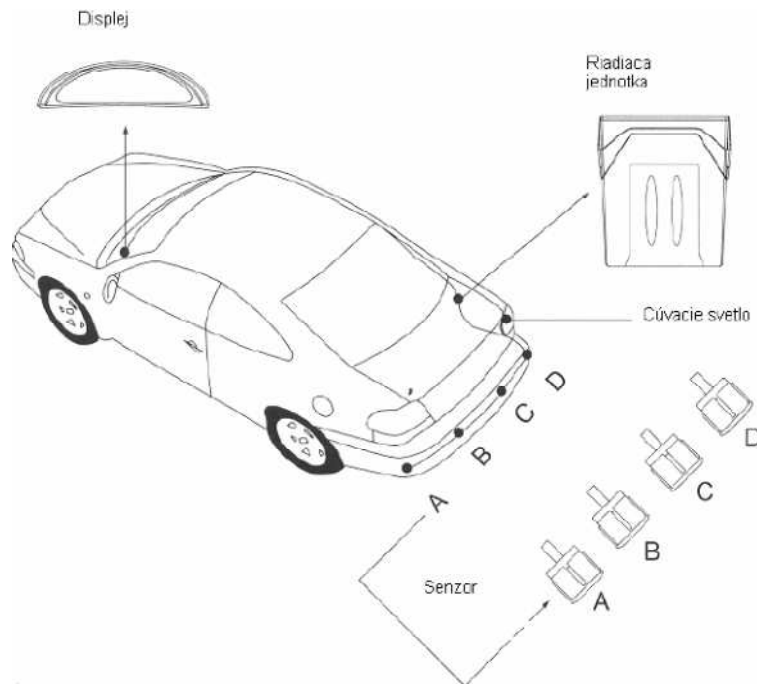
#### ***Bezdrôtový parkovací systém 4000***

Systém pozostáva zo skupiny štyroch ultrazvukových senzorov, diódového zobrazovacieho zariadenia a malého servera, do ktorého sú tieto senzory zapojené. Zapojenie je zobrazené na *obrázku 9*. Komunikácia so zobrazovacou jednotkou prebieha bezdrôtovo. Parkovací asistent sa aktivuje pri zaradení spiatočky[3].

Analyzované riešenie je finančne dostupné a funkcionalitou postačuje pre základné potreby vodičov, kedy akusticky a vizuálne informuje vodiča o vzdialenosti prekážky za vozidlom.

Výhodou je cena a bezdrôtová komunikácia zobrazovacieho zariadenia so serverom, kedy je možné server umiestniť do zadnej časti automobilu, aby nebolo potrebné robiť náročnú kabeláž pri prepojení so senzormi. Nevýhodou riešenia je jednoduchosť v

zobrazovaní informácií, kedy máme len minimálne informácie o prekážkach za vozidlom. Cena analyzovaného riešenia je 49.99 €.



**Obr. 9** Schéma zapojenia ultrazvukových senzorov[3]

### ***Parkovací asistent PS-082A***

Parkovací asistent PS-082A[4] predstavuje komplexnejšie riešenie, ktoré pozostáva z ultrazvukových senzorov, parkovacej kamery, zobrazovacieho zariadenia a servera. Systém je zobrazený na *obrázku 10*. Riešenie je bezdrôtové rovnako ako riešenie PS4000.

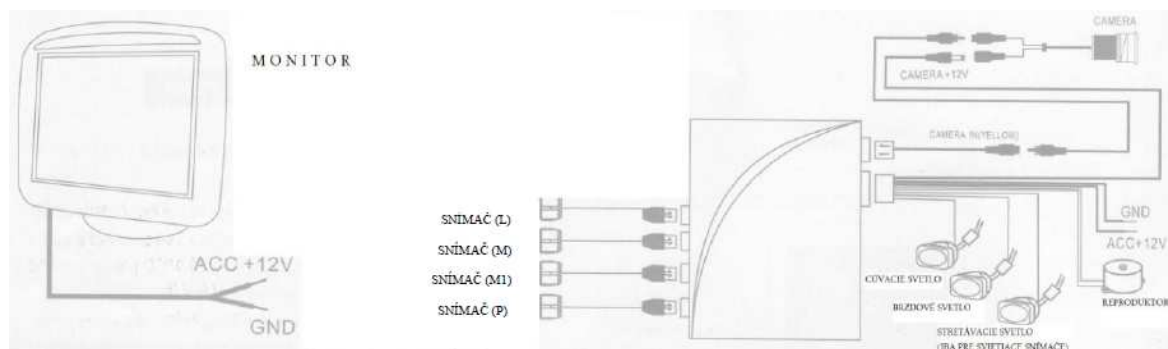
Systém je doplnený o parkovaciu kameru, ktorá je veľmi nápomocná, hlavne pri nákladných automobiloch alebo autobusoch. Kamera poskytuje zobrazovanie situácie za vozidlom aj v tme. Zobrazovacie zariadenie komunikuje so serverom bezdrôtovo. Zobrazovacie zariadenie je realizované v podobe spätného zrkadla, čo je veľké plus. Na monitore sa dá prepínať medzi režimom zobrazenia kamery alebo zobrazenie informácií z parkovacích senzorov. Parkovací asistent sa aktivuje zaradením spiatočky. Technické parametre sú uvedené v *tabuľke 1*.

Riešenie je finančne náročnejšie, čo však predstavuje daň za komplexnejšiu funkcionality v podobe parkovacej kamery. Cena systému je 217 €.

## Technické parametre:

**Tab. 1** Technické parametre dodávané k parkovaciemu asistentovi

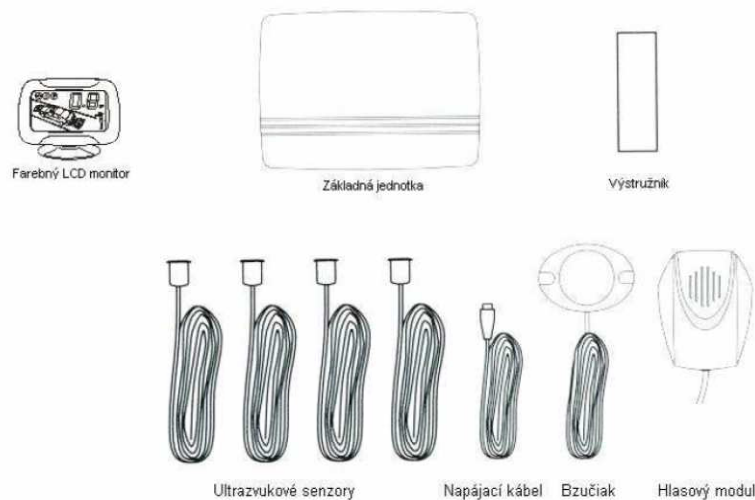
Napájanie	10 – 16 V js.
Prúdový odber pri aktivácii	200 – 850 mA
Optimálna detekčná vzdialenosť	0,4 – 1,5 m
Maximálna detekčná vzdialenosť	1,5 m
Akustické upozornenie na prekážku	áno, premenlivý tón
Optické upozornenie na prekážku	obraz + číselný a grafický údaj
Presné určenie prekážky	obraz + grafický údaj (vľavo / vpravo)
Počet snímačov / kamier	4 / 1
Pracovná frekvencia snímačov	40 kHz
Inštalčná výška snímačov	50 – 80 cm
Priemer otvoru na snímač / kameru	19 / 28 mm
Vŕtacie frézky súčasťou výrobku	áno
Lakovanie snímačov podľa farby vozidla	áno
Pracovná teplota displeja	0 až 60 °C
Pracovná teplota	-30 až 70 °C
Rozmer zrkadla / displeja	262 x 87 x 41 mm / 3,5“



**Obr. 10** Schéma zapojenia parkovacieho asistenta s kamerou a senzormi[4]

### ***Predné a zadné parkovacie senzory s LCD monitorom KL30928***

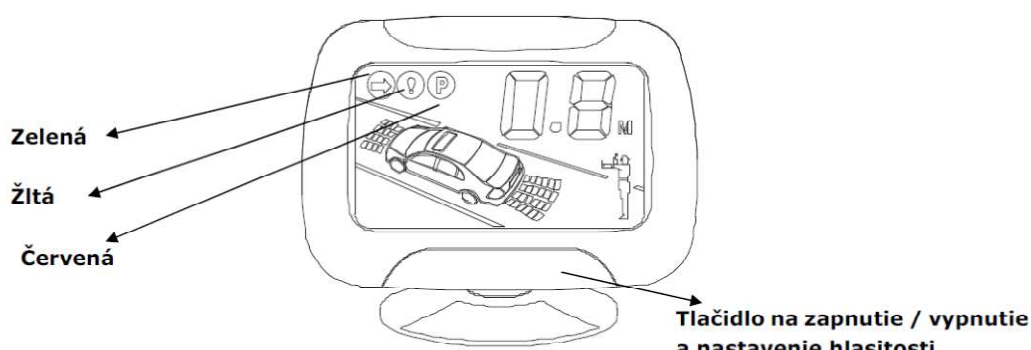
System predstavuje parkovací asistent od spoločnosti Molpir[10], ktorý využíva len ultrazvukové senzory. Na rozdiel od riešenia *parkovacieho asistenta 4000*, systém od spoločnosti Molpir využíva aj predné parkovacie senzory. Ako zobrazovacia jednotka slúži LCD monitor. Jednotlivé komponenty systému sú zobrazené na *obrázku 11*. Všetky komponenty sa pripájajú na server alebo k základnej jednotke.



**Obr. 11** Prvky parkovacieho asistenta [10]

Predné senzory sa aktivujú stlačením brzdového pedálu. V prípade, že sa pred vozidlom nenachádza prekážka monitor nereaguje. V prípade, že sa pred vozidlom objaví prekážka monitor zareaguje a zobrazí vzdialenosť vozidla od prekážky. Vizuálna interpretácia nameraných hodnôt je sprevádzaná zvukovou signalizáciou. Akustickú signalizáciu je možné kedykoľvek vypnúť.

Zadné senzory sa aktivujú pri zaradení spiatočky. Fungujú rovnakým spôsobom ako predné senzory. Zisťujú vzdialenosť vozidla od prekážky za vozidlom. V prípade, že sa za vozidlom nenachádza prekážka systém nereaguje. Avšak, ak je zaznamenaná prekážka, systém upozorní používateľa.



**Obr. 12** Zobrazovacie zariadenie parkovacieho asistenta[10]

Na obrázku 12 je znázornená prehľadnosť LCD monitora, ktorý poskytuje dôležité informácie. Jedná sa o vzdialenosť vozidla od prekážky, aktívne senzory, stav hlasitosti.

Popísané alternatívne riešenia parkovacích asistentov tvoria dobrý a postačujúci prehľad o aktuálnych ponukách na trhu. Popisujú technické možnosti, spôsoby realizácie ako aj cenové relácie systémov, v ktorých sú na trhu dostupné.

Porovnanie jednotlivých parkovacích asistentov je zobrazené v *tabuľke 2*. Továrnske modely parkovacích asistentov využívajú podobné technológie a riešenia sú realizované rovnakým spôsobom. Cenové relácie riešení sú rovnako veľmi podobné. Na rozdiel od toho, alternatívne riešenia poskytujú množstvo využiteľných informácií spracovaných z periférnych zariadení a sú o rády stoviek eur lacnejšie.

**Tab. 2** Porovnanie analyzovaných parkovacích asistentov

Parkovací asistent	Senzory	Kamera	Typ zobrazovania	Prenos signálu	Cena
Volkswagen(len senzory)	✓	✗	LCD/LED diódy	Kábel/WIFI	cca 500 €
Volkswagen(len kamery)	✗	✓	LCD	Kábel/WIFI	cca 700 €
Volkswagen(senzory/kamery)	✓	✓	LCD/LED diódy	Kábel/WIFI	cca 900 €
Volkswagen(automatické)	✓	✓	LCD/LED diódy	Kábel/WIFI	cca 1200 €
Lexus(len senzory)	✓	✗	LCD	Kábel/WIFI	cca 400 €
Lexus(len kamera)	✗	✓	LCD	Kábel/WIFI	cca 600 €
Lexus(senzory/kamera)	✓	✓	LCD	Kábel/WIFI	cca 800 €
Lexus(automatické)	✓	✓	LCD	Kábel/WIFI	cca 1000 €
PS4000	✓	✗	LED diódy	WIFI	44.9 €
PS-082A	✓	✓	spätné zrkadlo	WIFI	217 €
KL30928	✓	✗	LCD	Kábel	-
Navrhované riešenie	✓	✓	LCD	Kábel	-

### 1.3 Ultrazvuk

Ultrazvukom je nazývané akustické vlnenie, ktorého frekvencia leží nad hranicou počuteľnosti[11]. Hraničná hodnota frekvencie pre dané akustické vlnenie je udávaná hodnota 20KHz. Vlastnosti ultrazvuku, ktoré môžu byť ovplyvniteľné spadajú do závislosti na intenzite, amplitúde výchylky a frekvencii kmitočtov. Ultrazvuk je však ovplyvnený aj prostredím v ktorom sa šíri.

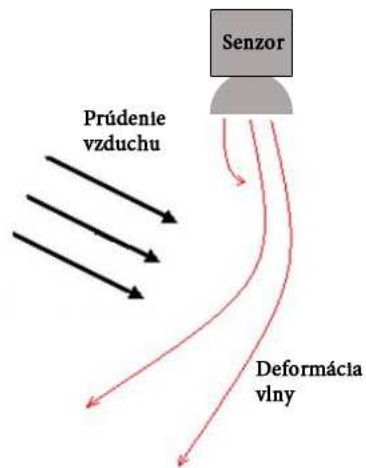
#### 1.3.1 Ultrazvuková energia

Aktívny ultrazvuk pracuje s veľkými intenzitami, približne nad  $0,5 \times 10^4 \text{W/m}^2$ [11]. Využíva sa na činnosti, ktoré vyvolávajú fyzikálne zmeny v prostredí. Jedná sa o činnosti ako vŕtanie, zváranie a iná úprava materiálov. Rovnako sa však používa aj v lekárskej terapii.

Pasívny ultrazvuk zahrňuje všetky oblasti použitia, kde ultrazvukové vlnenie dosahuje takú intenzitu, ktorá nevyvoláva žiadne fyzikálne zmeny v prostredí. Pasívny ultrazvuk je vďaka svojim malým intenzitám vhodný pre meracie a diagnostické účely.

### 1.3.2 Vplyv prúdenia zvuku na vlastnosti ultrazvukových senzorov

Rýchlosť šírenia zvuku v suchom prostredí pri teplote 0°C má hodnotu 331,6 m/s. S narastajúcou teplotou sa rýchlosť zvuku zvyšuje. Pri teplote 20°C rýchlosť zvuku narastie na hodnotu 343,8 m/s, čo je nárast približne o 4% oproti rýchlosti zvuku pri 0°C[14]. Pokiaľ je pri meraní teplota vzduchu rovnaká, stačí teplotu odmerať a chybu korigovať. V dnešnej dobe však majú niektoré senzory zabudovanú korekciu chyby vplyvom teploty.



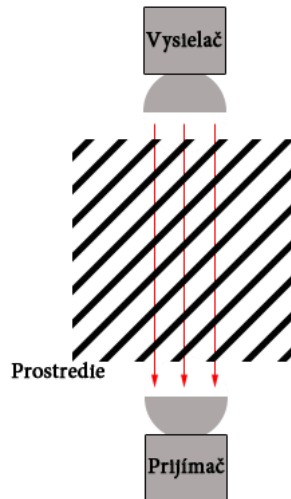
**Obr. 13** Vplyv prúdenia vzduchu na ultrazvukové vlny

Aj keď majú niektoré senzory zabudovanú korekciu chyby vplyvom teploty, ďalšie vplyvy vylúčiť nie je možné. Napríklad keď je senzor vystavený priamemu slnečnému žiareniu a vnútorný korektor senzora sa ohreje nad teplotu okolia, potom je výstupný údaj skreslený. Nepresnosti merania môžu rovnako vzniknúť ak teplota v priestore medzi senzorom a meranou prekážkou prudko rastie alebo prudko klesá. Vplyv prúdenia vzduchu na ultrazvukové vlnenie je znázornené na *obrázku 13*.

### 1.3.3 Princípy ultrazvukového merania

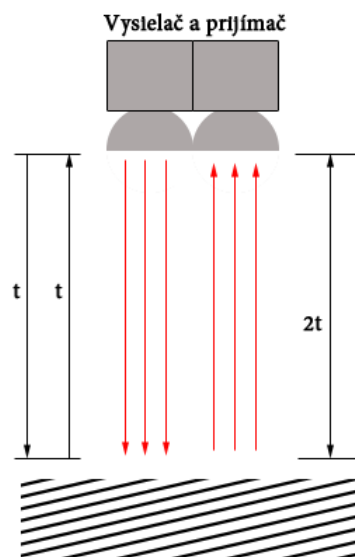
Ultrazvukové meranie je bezdotykové meranie. Princíp merania vzdialenosti k určitému objektu alebo k prekážke je založený na princípe merania doby medzi vyslaním ultrazvukovej vlny a následným prijatím tejto vlny. Doba medzi vyslanou a prijatou vlnou je priamo úmerná vzdialenosti snímača od meraného predmetu. Pri meraní vzdialenosti sa využívajú dva spôsoby merania. Prvý spôsob je meranie, kedy je oddelený vysielateľ a prijímač ultrazvukovej vlny. Druhý spôsob má integrovaný vysielateľ aj prijímač v jednom senzore.

Najstarší spôsob merania je meranie priechodové s oddelenou vysielačou a prijímacou sondou. Jedná sa o zisťovanie doby priechodu ultrazvukových vln meraným prostredím. Meranie je definované dvomi sondami, medzi ktorými sa nachádza merané prostredie. Pripojené sú v jednej osi na protíahlých stranách. Meranie je zobrazené na *obrázku 14*.



**Obr.14** Meranie ultrazvukom s oddeleným vysielačím a prijímaním

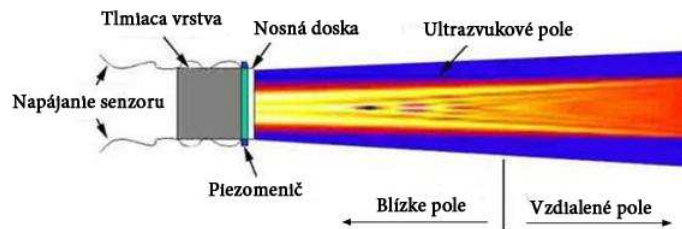
Druhý spôsob merania predstavuje metóda impulzová, odrazová, kedy je v jednej sonde zabudovaný vysielač a prijímač ultrazvukovej vlny. Vysielač posiela v určitých časových intervaloch ultrazvukové vlny. Tie sa po odrazení od predmetu vrátia do prijímača v tej istej sonde. Vzďialenosť je vypočítaná z času, ktorý je definovaný dobou vyslania ultrazvukovej vlny až po prijatie odrazenej ultrazvukovej vlny. Senzor je znázornený na *obrázku 15*.



**Obr.15** Modul vysielača a prijímača v jednej sonde

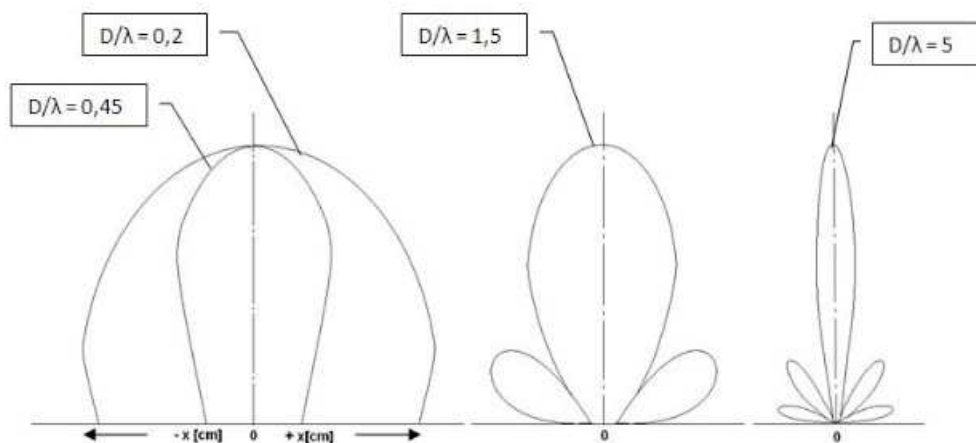
### 1.3.4 Vyžarovací diagram ultrazvukového senzora

Jedným z najdôležitejších parametrov ultrazvukového senzora, ktorým sa senzor podieľa na konečnej geometrickej rozlišovacej schopnosti ultrazvukového systému, je tvar vyžarovacieho diagramu senzora. Ultrazvukové pole, ktoré je generované meničom je možno z obecného pohľadu rozdeliť na dve základné oblasti. Tieto sú blízke vyžarovacie pole a vzdialené vyžarovacie pole. Rozdelenie poľa na blízku (Fresnelovu) a vzdialenú (Fraunhoferovu) oblasť je dané priebehom akustického tlaku v osi meniča. Ultrazvukové pole je vyžarované celou plochou piezokryštálu. Vyžarovanie je zobrazené na *obrázku 16*.



**Obr.16** Vyžarovací diagram

Tvar teoretických smerových charakteristík kruhových meničov závisí na pomere  $D/\lambda$ [14]. Jedná sa o priemer meniča  $D$  k jeho hrúbke. Smerová charakteristika meniča bude tým viac smerovejšia, čím bude väčšia hodnota pomeru  $D/\lambda$ . Poloha a tvar postranných lalokov smerovacej charakteristiky závisí na zvolenom spôsobe tlmenia meniča v samostatnom senzore a kvalite jeho akustickej väzby v sledovanom prostredí.



**Obr.17** Smerové vyžarovacie charakteristiky



Hlavný lalok smerovej charakteristiky je vymedzený uhlom  $\pm \theta$ , pre ktorý platí vzťah 1.

$$\sin\theta = 1.22 \frac{\lambda}{D}$$

**Vzťah 1** Fraunhoferova formula[14]

Vzťah 1 je využiteľný pri výbere vhodnej ultrazvukovej sondy. Pri definovanom priemere v rozsahu 1-30 mm, je možné smerovú charakteristiku ovplyvniť voľbou vhodnej pracovnej frekvencie v rozsahu 1-16 MHz použitých meničov.

## **1.4 Ultrazvukové senzory**

Kapitola popisuje ultrazvukové senzory dostupné na trhu, ktoré sú využiteľné pre rôzne typy riešení. Popisuje jednotlivé vlastnosti sensorov a vzájomne ich medzi sebou porovnáva.

### **1.4.1 Programovateľné ultrazvukové senzory P42**

#### **Vlastnosti[15]:**

- integrovaná elektronická jednotka
- rozsah merania 10cm – 600cm
- frekvencia signálu 4-16 Hz
- uhol vyžarovania signálu je 8°
- napájanie 19–30 V jednosmerne
- analógový výstup: 0–10 V alebo 4–20 mA
- programovateľné cez rozhrania R-S232 alebo RS-485
- kompenzácia merania vplyvom teploty
- presnosť 0,4 % alebo 1 mm
- vysoká linearita
- vysoká presnosť opakovaného merania
- pripojenie 8 pinovým konektorom
- pracovná teplota -15 až +70 °C
- cena cca 400 €

Senzor je často využívaný v automatizačnej technike. Poskytuje široký rozsah využitia. Hlavnou výhodou senzora je presnosť merania. Poskytuje zmenu meracích vlastností za pomoci počítača. K počítaču sa pripája pomocou štandardného sériového rozhrania RS-232. Výhodou je automatická korekcia chyby merania vplyvom teploty. Poskytuje dostatočný merací rozsah, maximálne 600 cm. Nevýhodou senzora je interpretácia

výstupných údajov. Poskytovaná je v analógovej forme v podobe prúdu alebo napätia. Ďalšou nevýhodou senzora je jeho vysoké napájacie napätie, čo by sa však mohlo riešiť transformáciou napätia. Pre použitie v aplikácii parkovacieho asistenta je najväčšou nevýhodou senzora jeho pomerne vysoká cena a nevyhovujúce základné požiadavky.

### **1.4.2 Microsonic HPS 340**

#### **Vlastnosti[16]:**

- rozsah merania 10cm – 1000 cm
- frekvencia signálu 120KHz
- uhol vyžarovania signálu je 15°
- napájanie 20–30 V jednosmerne
- analógový výstup
- kompenzácia merania vplyvom teploty
- presnosť 0,8 % alebo 5 mm
- vysoká presnosť opakovaného merania
- pripojenie 5 pinovým konektorom
- pracovná teplota -10 až +80 °C
- cena cca 150 €

Senzor HPS 340 sa využíva v automatizačnej technike. Medzi hlavné výhody senzora patrí maximálny merací rozsah, ktorý spĺňa požiadavky pre použitie do riešenia parkovacieho asistenta. Ďalšou výhodou je relatívne dobrá presnosť merania a výborná presnosť pri opakovanom meraní. Senzor disponuje vstavanou reguláciou merania vplyvom teploty.

Nevýhodou senzora je neštandardizovaný konektor a analógový výstup, ktorý by bol problematický pri zbere informácií do počítača. Senzor má vysoké napájacie napätie, čo je tiež nevýhodou z dôvodu, že automobily disponujú napájacím napätím 12V jednosmerne. Rovnako ako pri senzore P42 aj pri senzore HPS 340, je veľkou nevýhodou vysoká nákupná cena.

### **1.4.3 Ultrazvukový senzor SRF08**

#### **Vlastnosti[17]:**

- rozsah 3 cm - 6 m
- výstup na I<sup>2</sup>C, RS-232
- frekvencia 40 kHz
- napájacie napätie 5 V jednosmerne

- kompenzácia merania vplyvom teploty
- presnosť merania 3 mm
- vysoká presnosť opakovaného merania
- cena 57.90 €

Na rozdiel od senzorov P42 a HPS 340 sa senzor SRF08 využíva v robotických technológiách. Veľkou výhodou senzora, ale aj senzorov P42 a HPS 340, je ich presnosť merania. Disponujú korekciou merania pri rôznych teplotách. Výhodou senzora SRF08 je napájacie napätie, ktoré vyhovuje požiadavkám pre použitie v aplikácii parkovacieho asistenta. Pri napájacom napätí 5V postačuje napájanie zo zbernice, alebo z 5V stabilizátora. Jednou z najväčších výhod senzora SRF08 je digitálny výstup. Senzor komunikuje s nadradeným zariadením cez komunikačné rozhranie I<sup>2</sup>C.

Nevýhodou senzora je vyššia cena, nie je však taká vysoká ako pri senzoroch P42 a HPS 340, preto by bol senzor vhodný pre využitie v aplikácii parkovacieho asistenta.

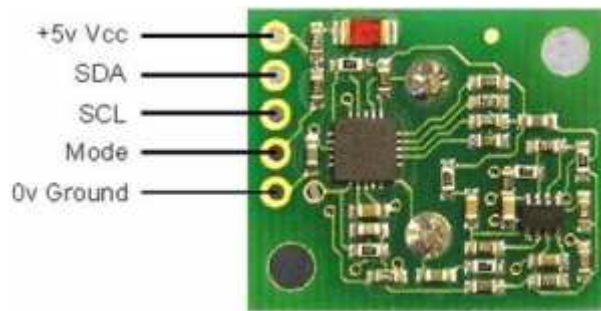
#### **1.4.4 Ultrazvukový senzor SRF02**

##### **Vlastnosti[18]:**

- rozsah 15 cm - 6 m
- výstup na I<sup>2</sup>C, RS-232
- frekvencia 40 kHz
- napájacie napätie 5 V jednosmerne
- kompenzácia merania vplyvom teploty
- presnosť merania 3 mm
- vysoká presnosť opakovaného merania
- cena 21.90 €

Senzor SRF02 má veľmi podobné vlastnosti ako senzor SRF08. Hlavným rozdielom je merací rozsah, v ktorom senzor meria. SRF02 meria pri spodnej hranici do 15cm čo by však pre využitie pre aplikáciu parkovacieho asistenta nebol až taký problém. Napájacie napätie je vyhovujúce, rovnako tak aj typ výstupného signálu. Spomedzi všetkých senzorov je cena najpriateľnejšia.

Senzor môže komunikovať cez komunikačné rozhrania I<sup>2</sup>C alebo RS-232. Výber komunikačného rozhrania sa definuje nastavením signálu *MODE*, ktorý je zobrazený na *obrázku 18*. Použitie interných funkcií senzora umožňuje vysielanie impulzu bez očakávania potvrdenia od nadradeného systému a bez očakávania na príjem odrazeného signálu.



**Obr. 18** Spodná časť senzoru SRF02

V prípade, že má senzor pracovať a komunikovať cez komunikačné rozhranie RS-232 je nutné nastaviť signál *MODE* na hodnotu 0V alebo ho prepojiť so signálom *GROUND*. Ak má senzor pracovať a komunikovať cez komunikačné rozhranie I<sup>2</sup>C, signál *MODE* ostane nezapojený. Na signál *SCL* sa pripájajú externé taktovacie hodiny a na signál *SDA* sa pripájajú dáta. Senzor obsahuje 5 registrov.

**Popis registrov SRF02:**

Reg 0 – z registra nie je možné čítať, slúži len na spustenie meracej sekvencie

Reg 1 – register nie je používaný

Reg 2 a 3 – sú dva 8 bitové registre, obsahujú informácie z každého posledného merania. V registri 2 je uložený vyšší bajt a v registri 3 je uložený nižší bajt.

Reg 4 a 5 – tieto registre slúžia na ukladanie minimálnej hraničnej meranej hodnoty. Register sa aktualizuje automaticky podľa okolitej teploty a definovaných algoritmov.

Senzor sa aktivuje poslaním hodnoty 0x51 na adresu 0xE0 do registra 0. Nutnosť je vyčkat' určitý časový úsek, po ktorom je možné čítať namerané hodnoty z registrov 2 a 3.

**1.4.5 Porovnanie analyzovaných ultrazvukových senzorov**

**Tab. 3** Tabuľkové porovnanie senzorov

Názov	Rozsah merania	Frekvencia	Napájacie napätie	Výstup	KMVT*	Presnosť	Cena
<b>P42</b>	10cm - 6m	4 - 16 Hz	19 - 30V	analog.	✓	1mm	400 €
<b>HPS 340</b>	10cm - 10m	120KHz	20 - 30V	analog.	✓	5mm	150 €
<b>SRF08</b>	3cm - 6m	40KHz	5V	I2C, RS232	✓	3mm	57.90 €
<b>SRF02</b>	15cm - 6m	40KHz	5V	I2C, RS232	✓	3mm	21.90 €

\*KMVT – korekcia merania vplyvom teploty

Tabuľka 3 poskytuje prehľadné porovnanie vlastností jednotlivých opisovaných senzorov. Na základne vlastností bol vybraný a použitý senzor SRF02, ktorý má vyhovujúci rozsah

merania, ktorý pre riešenie parkovacieho asistenta maximálne vyhovuje. Vďaka napájaniu 5V je možné napájať senzor priamo z komunikačného rozhrania a zo servera BV-105. Senzor komunikuje cez sériové rozhrania I<sup>2</sup>C alebo cez RS-232, čo je jeho veľkou výhodou. Presnosť merania senzora SRF02 nie je najlepšia spomedzi porovnávaných senzorov, ale pre použitie v aplikácii parkovacieho asistenta je maximálne vyhovujúca. Jedným z rozhodujúcich faktorov je aj cena senzora, vďaka ktorej sa náklady nenavýšia vo veľkej miere.

## **1.5 Komunikačné rozhrania**

Kapitola analyzuje komunikačné rozhrania, ktoré je možné použiť na komunikáciu senzora so serverom. Výstupom kapitoly je výber komunikačného rozhrania, ktoré najviac vyhovuje požiadavkám riešenia. V predchádzajúcej kapitole 1.4 *Ultrazvukové senzory* bol vybraný senzor SRF02, ktorý podporuje komunikáciu pomocou RS-232 a I<sup>2</sup>C rozhraní. Z toho dôvodu je jasné, že budú analyzované komunikačné rozhrania RS-232 a I<sup>2</sup>C.

### **1.5.1 RS-232**

Komunikačné rozhranie RS-232, je rozšírené komunikačné rozhranie, ktoré sa používa na komunikáciu osobných počítačov a inej elektroniky[19]. Pri sériovej komunikácii sa dáta posielajú jednotlivou sa sebou po jednom vodiči.

Štandard RS-232 definuje napätia a prenosové rýchlosti medzi zariadeniami, ktoré ho používajú. Podľa štandardu, RS-232 sú definované vzájomne sériovo komunikujúce zariadenia typov DTE a DCE. V tomto prípade skratka DTE reprezentuje server a skratka DCE zariadenie, ktoré je s ním prepojené prostredníctvom sériovej linky, čiže senzor. Štandard RS-232 definoval že DTE zariadenia používali 25-pinový konektor (zástrčku) a DCE zariadenia 25-pinový konektor (zásuvku). Konektory, ktoré používali 25 pinov boli postupne nahradené konektormi používajúcimi 9 pinov. Konektory na strane DTE majú vždy kolíky (samec).

Pri sériovej komunikácii je nutné nastaviť parametre komunikácie ako:

- číslo COM portu
- rýchlosť prenosu (baud rate)
- počet bitov
- parita
- stop bity

### ***Synchronný prenos***

Pri synchronnom prenose sa na vodičoch nastaví úroveň signálu a informácia sa potvrdí impulzom, alebo zmenou úrovne synchronizačného signálu. Synchronizačný signál je vysielaný aj keď sa neprenášajú žiadne dáta. Synchronizácia umožňuje rýchlejší prenos dát ako asynchronný prenos, keďže nevyžaduje vysielanie dodatočných znakov vyznačujúcich začiatok a koniec prenášaného rámca. Vysielacia a prijímacia strana musia mať časové základne, generátory taktovacích impulzov, ktoré musia byť zosynchronizované. Značky sú vysielané ako nepretržitý reťazec bitov. V medzerách sa automaticky vkladajú stavové bity. Začiatky a konce všetkých bitov musia korešpondovať s časovou základňou. Nevýhodou je zložité synchronizovanie, veľkou výhodou však je efektívne využitie kanála a možnosť zabezpečenia prenosu proti chybovosti. Prenos sa používa pre veľké objemy dát aj tam, kde je potrebné počas prenosu zabezpečiť šírku pásma.

### ***Asynchronný prenos***

Pri asynchronnom prenose sa dáta prenášajú v sekvenciách (rámcoch) danou rýchlosťou s úvodnou štartovacou sekvenciou vyslaním štart bitu. Hneď ako bol štart bit vyslaný, vysielateľ posielá dátové bity, ktorých môže byť 5, 6, 7, alebo 8, v závislosti od konfiguračnej voľby. Nasleduje paritný bit (nie je povinný). Prenášaný rámec je ukončený stop bitom. Dĺžky trvania prenosu jednotlivých bajtov sú rôzne, rôzne sú i medzery medzi dvoma blokmi. Prijímač nerozozná dva nasledujúce dátové bloky. Objem prenášaných dát sa pri tomto type prenosu rozdelí na menšie časti, každá časť sa posielá nezávisle od ostatných, časové intervaly medzi odosielaním jednotlivých bajtov sa môžu líšiť.

Nevýhodou tohto rozhrania je to, že nie je priamo podporované Funtoro serverom, preto nie je celkom vhodné ho použiť. Zbernica poskytuje pomalé prenosové rýchlosti, ktoré nie sú vhodné pre použitie v aplikácii parkovacieho asistenta.

### **1.5.2 Rozhranie I<sup>2</sup>C**

Zbernicu I<sup>2</sup>C vyvinula spoločnosť Philips Semiconductor[20] pôvodne pre svoje TV prijímače v roku 1980 za účelom komunikácie medzi integrovanými obvodmi na jednej doske plošných spojov pri použití minimálneho množstva signálových pripojení. Špecifikácia zbernice I<sup>2</sup>C je založená na jednoduchých hardvérových štandardoch, nie sú potrebné špeciálne konektory ani kabeláž. Obvody, ktoré používajú I<sup>2</sup>C protokol zahŕňajú pamäte EEPROM a RAM, senzory teploty, expandéry portov, hodiny reálneho času,

rovnako tak aj ultrazvukové senzory. Používa sa tiež ako riadiaca zbernica v obvodoch spracovania signálov, ktoré majú oddelenú dátovú zbernicu, napr. RF tunery, video dekodéry, enkodéry, audio procesory a kódeky. Zbernica I<sup>2</sup>C môže pracovať pri troch prenosových rýchlostiach:

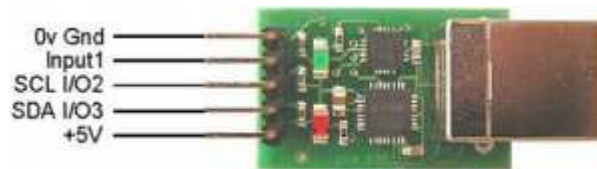
- Pomalá (pod 100Kbps)
- Rýchla (400Kbps)
- Vysokorýchlostná (3,4Mbps)

Vzdialenosť komunikujúcich zariadení je limitovaná, z dôvodu udržania komunikačnej rýchlosti, na približne 4 m a maximálna kapacita prenosového vedenia je 400pF[20]. Zbernica používa dva vodiče a to dátovú linku (SDA) a taktovaciu linku (SCL). Vodiče a rozhranie sú znázornené na *obrázku 19*. Synchronna komunikácia viacerých nadriadených zariadení znamená to, že aj podriadené zariadenie môže byť konfigurované tak, aby mohlo začať komunikáciu. Každý integrovaný obvod alebo čokoľvek pripojené na zbernici je identifikované svojou adresou, ktorá je v rámci siete jedinečná, preto zbernica I<sup>2</sup>C nevyžaduje signály zabezpečujúce identifikáciu zariadení na zbernici ani ďalšiu logiku. Dátové aj taktovacie signály sú pripojené na napájacie napätie pomocou odporov.

Nadriadené zariadenie začína a končí dátový prenos generovaním štart bitu a stop bitu. Generuje hodinový signál. Vysiela adresu podriadeného zariadenia, pre ktorého budú dáta určené, rovnako tak určuje smer prenosu dát.

Podriadené zariadenie odpovedá iba v prípade, že rozpozná svoju adresu. Časovanie komunikácie a prenosu dát je riadené hodinovým signálom z nadriadeného zariadenia.

Komunikácia spočíva v nastavení hodinového signálu nadriadeným zariadením a nastavením adresy zariadenia s ktorým chce nadriadené zariadenie komunikovať. Pošle štart bit a čaká. Podriadené zariadenie po prijatí bitu pošle naspäť potvrdenie, čím potvrdí, že je aktívne a schopné komunikovať. Nadriadené zariadenie začne komunikovať a posielat' dáta. Podriadené zariadenie pracuje s dátami a vykonáva požadované úkony. Prebieha komunikácia. Pre ukončenie komunikácie nadriadené zariadenie za posledným bajtom dát pošle jeden stop bit. Ak podriadené zariadenie prijme stop bit, potvrdí ho potvrdzovacím bitom. Nadriadené zariadenie už neodpovedá, že prijalo potvrdzovací bit. Nakoniec nastaví druhý stop bit, čím ukončí komunikáciu.



**Obr.19** Rozhranie I<sup>2</sup>C[20]

Hlavnou výhodou rozhrania je jeho komunikačná rýchlosť a menšia komunikačná réžia, keďže sa jedná o synchronizovaný prenos. I<sup>2</sup>C je pomocou jednoduchého zapojenia, ktoré využíva procesor FTDI, možné prerobiť na podporu USB. Čo je priamo podporované na Funtoro servery.

Výsledkom porovnania komunikačných rozhraní RS-232 a I<sup>2</sup>C je výber komunikačného rozhrania I<sup>2</sup>C z dôvodu vyhovujúcich vlastností a parametrov komunikácie, ktorými sú rýchlosť a možnosť pripojenia na server BV-105.

### **1.6 Záver analýzy**

Hlavným cieľom *kapitoly 1 Analýza riešenia* bola analýza existujúcich riešení, kde bolo nutné zistiť ako dané riešenia parkovacích asistentov pracujú, v akom výstupnom tvare poskytujú vodičom informácie. Ďalšia časť pozostáva zo samotného porovnania parkovacích asistentov, v čom sú dobré a v čom majú určité nedostatky. Na základe týchto porovnaní bolo nutné navrhnúť parkovací asistent, ktorý bude vyplňať tieto nedostatky a bude poskytovať vodičom plné a komplexné informácie pri parkovaní.

Po definovaní funkcionality, ktorú má vytváraný parkovací asistent realizovať, bolo nutné analyzovať jednotlivé komponenty, z ktorých bude pozostávať. Hlavným problémom bol preto výber vhodného servera, na ktorom parkovací asistent pracuje. Analýza podrobne opisuje rôzne riešenia serverov od spoločnosti Funtoro. Výstupom analýzy Funtoro komponentov je výber najvhodnejšieho servera a to servera BV-105. Hlavnú podmienku, ktorú spĺňa sú vstupné USB porty, vstupné USB porty a Windows CE. Parkovací asistent pracuje aj s parkovacím senzorom, ktorý je jeho neoddeliteľnou súčasťou. Podrobná analýza popisuje rôzne druhy senzorov, ktoré sú dostupné na trhu. Po dôkladnom porovnaní je vybraný najvhodnejší senzor. Senzor SRF02, ktorý maximálne spĺňa požiadavky pre aplikáciu parkovacieho asistenta. Hlavnými požiadavkami, na ktoré sa prihliadalo boli merací rozsah, napájacie napätie, tvar výstupných údajov, kompatibilita so serverom a samozrejme aj cena. Podrobnejšie informácie sú uvedené v prehľadnej tabuľke v *kapitole 1.4 Ultrazvukové senzory*.



Nutnou požiadavkou na senzor je kompatibilita so serverom. Kompatibilitu však nie je možné zaručiť na sto percent na továrenském výrobku. Nutnosťou bolo preto integrovať medzi server a senzor najvhodnejšie rozhranie, vďaka ktorému by boli schopné tieto dva komponenty komunikovať. Vybrané bolo rozhranie I<sup>2</sup>C, ktoré maximálne vyhovuje na prepojenie týchto komponentov. Jedná sa o sériovú komunikáciu vo vyšších rýchlostiach.

Samozrejmou riešením je využívanie parkovacej kamery. Tento doplnok neposkytuje špeciálnu funkcionálnu funkciu, ale je vhodným spĺnením procesu parkovania. Pri vhodnej interpretácii údajov zo senzora a obrazu z parkovacej kamery bude parkovanie pre vodiča ľahkou činnosťou a nie nočnou morou.

Výstupom analýzy je dostatočný prehľad o dostupných parkovacích asistentoch ako aj výber vhodných komponentov, vďaka ktorým je možné zostrojiť kvalitný parkovací asistent.

## 2 Opis riešenia

Kapitola popisuje podrobný návrh riešenej aplikácie parkovacieho asistenta. Zaoberá výberom implementačného prostredia vhodného pre implementáciu aplikácie. Popisuje požiadavky na systém, popisuje najdôležitejšie implementované funkcie.

### 2.1 Špecifikácia riešenia

Podkapitola opisuje špecifikáciu riešenia, funkcionálne požiadavky ako aj nefunkcionálne požiadavky, ktoré sú kladené na systém.

#### 2.1.1 Funkcionálne požiadavky

Cieľom diplomovej práce je vytvorenie aplikácie parkovacieho asistenta, ktorý pracuje s parkovacou kamerou a s ultrazvukovým senzorom. Veľmi dôležité je, aby bol program používateľsky jednoduchý a poskytoval informácie dôležité pre správne zaparkovanie automobilu, keďže sa jedná o vysoké finančné straty v prípade zlyhania systému. Aplikácia pracuje vo viacerých režimoch, kedy poskytuje rôzne druhy informácií z periférnych zariadení. Technicky by mala byť vyriešená inicializácia systému pri zaradení spiatočky, čo však nie je riešením tejto práce. Aplikácia však môže byť inicializovaná manuálne používateľom pred parkovaním. Po inicializovaní systému bude na výber práca v jednom z režimov, ktoré sú podrobne popísané v kapitole 2.4 *Návrh architektúry systému*.

V prvom režime aplikácia pracuje s periférnym zariadením a to s parkovacou kamerou. Na monitore sú zobrazované informácie z tejto kamery v podobe videa, ktoré je zobrazované na celej obrazovke. V prípade dostupnosti viacerých kamier je možné prepínanie obrazov medzi nimi. Rovnako je dostupné tlačidlo pre ukončenie režimu a vrátenie sa do počiatočného stavu kedy chceme zmeniť režim asistenta. Tento režim jednoznačne uľahčí parkovanie vozidla.

V druhom režime systém pracuje s ultrazvukovým senzorom. Na obrazovke je zobrazený automobil, v ňom je naznačený ultrazvukový senzor. Za vozidlom sú naznačené zóny vzdialenosti. Pri cúvaní sa tieto zóny farebne menia, čím vodiča informujú o zmene vzdialenosti vozidla od prekážky. Rovnako je cieľom vytvorenie prepočtu vzdialenosti vozidla od prekážky v rozumnej mierke, v metroch, respektíve pre lepšiu predstavu v kratšej vzdialenosti aj v centimetroch. Na obrazovke sa zobrazuje tlačidlo, ktoré umožňuje vrátenie sa do počiatočného stavu a umožní výber ktoréhokolvek z režimov.

Tretí režim poskytuje informácie kombinované. V menšom okne ponúkne video z parkovacej kamery. V ďalšej časti obrazovky, sú naznačené diódy, ktoré sa rozsvetujú podľa vzdialenosti od prekážky a poskytujú informácie o vzdialenosti.

Štvrtý režim poskytuje podporu pre pripojenie štyroch nezávislých senzorov. Informácie sú spracovávané s každým senzorom osobitne. V prehľadnej podobe sú zobrazené na obrazovke a používateľovi poskytujú dostatočné informácie o vzdialenosti vozidla od prekážky. Sensory poskytujú presnú vzdialenosť, ktorú namerali.

Parkovací asistent poskytuje základné nastavenie senzorov, ktoré umožňuje nastaviť merací rozsah každého senzoru osobitne. Poskytuje možnosť zapnutia alebo vypnutia merania daného senzora. Aplikácia rieši aj kontrolu pripojenia senzorov. Doplnková funkcionálna predstavuje meranie dĺžky parkovania pomocou parkovacích stopiek.

### **2.1.2 Nefunkcionálne požiadavky**

Hlavným cieľom je vytvoriť program, ktorý bude spĺňať nasledujúce požiadavky:

- intuitívna práca s programom
- prehľadné používateľské rozhranie
- jednoduchosť a rýchla dostupnosť požadovanej funkcie
- interpretácia údajov v jednoznačnom zrozumiteľnom tvare
- rýchla aktualizácia údajov

Program musí byť intuitívny, aby sa ho používateľ nemusel dlho učiť, ale aby mu bolo hneď po spustení jasné, čo sa udeje po kliknutí na niektoré z ikoniek. Musí byť prehľadný a zrozumiteľný.

### **2.1.3 Diagram prípadov použitia pre kameru a senzor**

**Číslo prípadu použitia:** UC1

**Názov:** Výber režimu s parkovacou kamerou

**Hráč:** Používateľ

**Popis:** Používateľ chce používať režim s parkovacou kamerou a parkovať za pomoci videa z parkovacej kamery. Používateľ po inicializácii programu alebo po vrátení sa na hlavnú obrazovku, vyberie režim parkovacej kamery.

**Číslo prípadu použitia: UC2**

**Názov:** Výber režimu s ultrazvukovým senzorom

**Hráč:** Používateľ

**Popis:** Používateľ chce používať režim s ultrazvukovým senzorom a parkovať za pomoci informácií o vzdialenosti od prekážky. Používateľ po inicializácii programu alebo po vrátení sa na hlavnú obrazovku, vyberie režim ultrazvukových senzorov.

**Číslo prípadu použitia: UC3**

**Názov:** Výber duálneho režimu

**Hráč:** Používateľ

**Popis:** Používateľ chce používať režim s ultrazvukovým senzorom a parkovacou kamerou. Chce mať informácie o vzdialenosti vozidla od prekážky a rovnako tak aj chce sledovať ako môže daná prekážka vyzeráť. Používateľ po inicializácii programu alebo po vrátení sa na hlavnú obrazovku, vyberie kombinovaný režim, pracujúci so senzormi a kamerou.

**Číslo prípadu použitia: UC4**

**Názov:** Zmena režimu asistenta

**Hráč:** Používateľ

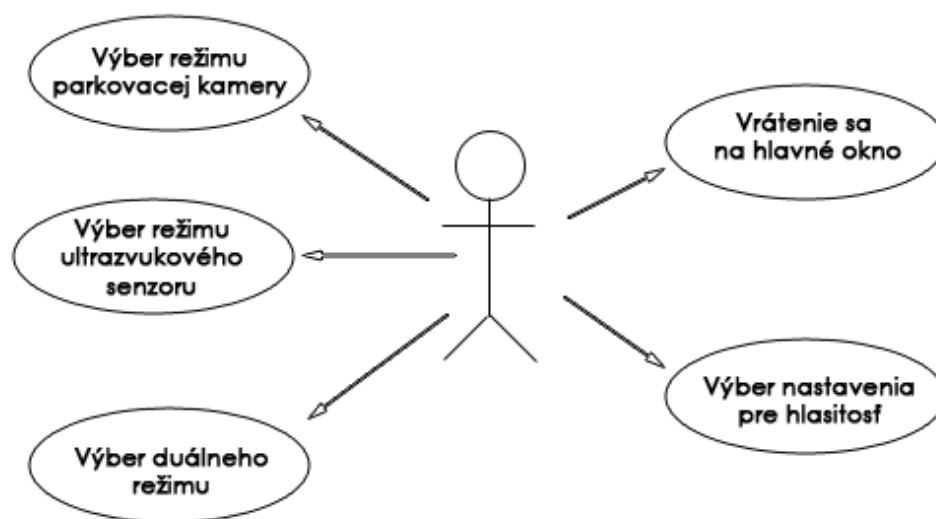
**Popis:** Používateľ pracuje v jednom z režimov, ale z určitého dôvodu chce tento režim zmeniť. Klikne na tlačidlo späť a vyberie si iný požadovaný režim.

**Číslo prípadu použitia: UC5**

**Názov:** Zmena zvukov

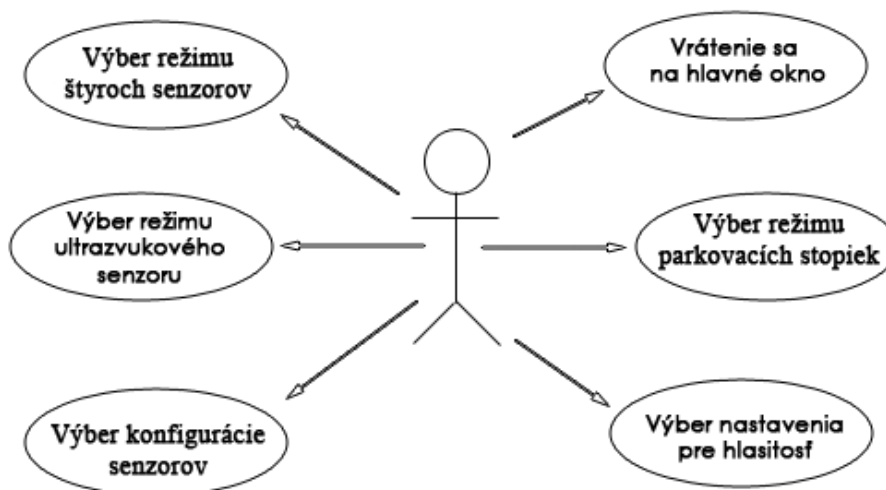
**Hráč:** Používateľ

**Popis:** Používateľ je na hlavnej obrazovke a chce zapnúť alebo vypnúť zvuky v programe tak klikne na príslušnú ikonu a tým zmení nastavenie.



Obr. 20 Diagram prípadov použitia pre kameru a senzor

#### 2.1.4 Diagram prípadov použitia pre štyri senzory



Obr. 21 Diagram prípadov použitia pre štyri senzory

**Číslo prípadu použitia:** UC6

**Názov:** Výber režimu štyroch senzorov

**Hráč:** Používateľ

**Popis:** Používateľ chce používať režim s podporou merania štyroch senzorov. Používateľ po inicializácii programu alebo po vrátení na hlavnú obrazovku, vyberie režim štyroch senzorov.

**Číslo prípadu použitia:** UC7

**Názov:** Výber režimu s ultrazvukovým senzorom

**Hráč:** Používateľ

**Popis:** Zhodné s UC2.

**Číslo prípadu použitia:** UC8

**Názov:** Výber konfigurácie senzorov

**Hráč:** Používateľ

**Popis:** Používateľ chce konfigurovať ultrazvukové senzory. Po inicializácii programu alebo po vrátení na úvodnú obrazovku vyberie konfiguráciu senzorov.

**Číslo prípadu použitia:** UC9

**Názov:** Zmena režimu asistenta

**Hráč:** Používateľ

**Popis:** Zhodné s UC4

**Číslo prípadu použitia:** UC10

**Názov:** Zmena nastavenia hlasitosti

**Hráč:** Používateľ

**Popis:** Zhodné s UC5

**Číslo prípadu použitia:** UC11

**Názov:** Spustenie parkovacích stopiek

**Hráč:** Používateľ

**Popis:** Používateľ si chce odmerať dobu za akú je schopný vykonať parkovací manéver. Po inicializácii programu alebo po vrátení sa na hlavnú obrazovku klikne na ikonu parkovacích stopiek a spustí meranie.

## 2.2 Výber hardvéru

Pri výbere hardvérových komponentov budeme vychádzať z analýzy, konkrétne z kapitoly 1.6 Záver analýzy, ktorá presne popisuje dôvod výberu hardvérových komponentov. Ako server je použitý BV-105 od spoločnosti Funtoro. Na tomto servery pracuje operačný systém Windows CE.

Program je zobrazovaný na dotykovom monitore MOD rovnako od spoločnosti Funtoro. Použitie monitora tohto typu má výhodu v tom, že monitor je dotykový a má integrované výstupné zvukové zariadenie, ktoré je využité pri akustickej signalizácii.

Meranie vzdialenosti vozidla od prekážky zabezpečuje ultrazvukový senzor SRF02, ktorého výber je zdôvodnený v *kapitolách 1.6 Záver analýzy* a *1.4 Ultrazvukové senzory*.

Na zachytávanie obrazu za automobilom je použitá USB kamera. USB kamera je z dôvodu, že použitý server obsahuje dostupné porty pre pripojenie.

### **Zoznam hardvérových komponentov:**

- Funtoro server BV-105 s operačným systémom Windows CE
- ultrazvukový senzor SRF02
- USB parkovacia kamera
- Funtoro MOD monitor

## 2.3 Výber implementačného prostredia

Výber implementačného prostredia je dôležitý krok pred samotnou implementáciou systému. Keď sa zvolí zle, môže to ovplyvniť nielen náročnosť implementácie, ale aj konečný program. Treba si zvážiť a porovnať, ktoré implementačné prostredie je najvhodnejšie.

Program musí byť intuitívny a používateľsky príjemný. Keďže bude obsahovať aj grafický výstup, jednoznačne musí mať grafické rozhranie. V analyzovaných existujúcich riešeniach výstup predstavoval aj diódový monitor, ktorý nepotrebuje grafický výstup, len riadenie signalizácie diódového zobrazovača, prípadne akustickú signalizáciu. Aplikácia musí byť multifunkčná. Primárnym cieľom diplomovej práce je vytvoriť programové riešenie schopné pracovať vo viacerých režimoch, ktoré budú súčasne využívať ultrazvukový senzor a kameru alebo každé zariadenie osobitne. Hlavným problémom, ktorý je treba vyriešiť je návrh rozhrania, ktoré zabezpečí komunikáciu aplikácie s periférnymi zariadeniami a samotné spracovanie prijatých údajov.

### 2.3.1 Java

Objektovo orientovaný jazyk od spoločnosti Sun (teraz patrí už pod spoločnosť Oracle)[21]. Je platformovo nezávislý – takisto beží pod operačným systémom Windows ako pod Linuxom. Existuje veľa foriem javy, napríklad pre webové aplikácie, mobilné zariadenia alebo aplikácie pre stolové počítače. Vychádza z jazyka C++[21]. Jeho nevýhoda je, že kvôli multiplatformovej podpore je vykonávanie programov pomalšie a neefektívne.

### 2.3.2 C++

Je rozšírením jazyka C o triedy a objekty, ktorý si zachoval procedurálnu stránku jazyka C[22]. Poskytuje širokú škálu možností, ako napr. dedenie, zapuzdrenie, šablóny, predlohy a podobné vlastnosti. Poskytuje podporu grafického rozhrania pomocou triedy MFC[22]. Programovací jazyk by bol alternatívou pre programovanie parkovacieho asistenta.

### 2.3.3 Platforma .NET

V skutočnosti sa ani nejedná o programovací jazyk, ale skôr o programovaciu techniku. Základ tvorí .NET Framework, ktorý poskytuje možnosť objektovo orientovaného programovania[23]. Podporuje viacero programovacích jazykov ako C++, C#, Visual Basic a ďalšie. Výhodou je, že tieto programovacie jazyky sa líšia len syntaxou, ale v podstate používajú rovnaké knižnice alebo dátové typy[23]. Preto aj všetky tieto jazyky sú rovnako výkonné a efektívne. Automaticky podporuje triedy, metódy, vlastnosti, udalosti, polymorfizmus[23].

### 2.3.4 C#

Je objektovo orientovaný programovací jazyk vyvinutý spoločnosťou Microsoft[27] ako časť ich iniciatívy .NET[24]. Microsoft si za základ pre nový jazyk C# zbral C++ a jazyk Java. C# bolo navrhované s úmyslom vyvážiť silu jazyka C++ a tú spojiť s možnosťou rýchleho programovania, ktoré ponúkali jazyky ako napríklad Visual Basic, Delphi.

Ciele pri tvorbe jazyka C#:

- C# bol vytváraný tak, aby bol jednoduchý, moderný, objektovo orientovaným jazykom pre všeobecné použitie.
- Jazyk a jeho implementácie by mali poskytovať podporu pre nasledovné princípy softvérového inžinierstva ako silná typová kontrola, kontrola ohraničenia polí,



detekciu pokusov na využitie neinicializovaných premenných a automatickú správu pamäte. Dôležitými vlastnosťami je tiež robustnosť, odolnosť a produktivita.[24]

- Jazyk je tvorený so zreteľom na vývoj softvérových komponentov, ktoré sú vhodné pre nasadenie v distribuovaných prostrediach.
- Prenositeľnosť zdrojového kódu je mimoriadne dôležitá, rovnako ako aj prenositeľnosť znalostí programátorov so skúsenosťami s jazykmi C a C++.
- C# je jazyk zameraný na tvorbu aplikácií pre stolové počítače, ako aj pre vnorené systémy s ohľadom na veľkú škálovateľnosť od veľmi veľkých systémov používajúcich sofistikované operačné systémy po veľmi malé zariadenia pre špecializované úlohy[24].
- Aj keď je jazyk C# plánovaný na ekonomické využívanie pamäte a procesorového času, nie je kompletne zameraný na výkonnosť a veľkosť výsledného binárneho kódu ako jazyky C alebo assembler.

Aplikáciu parkovacieho asistenta je možné implementovať vo viacerých programovacích jazykoch. Medzi vhodné implementačné jazyky patrí Java, C++, C#. Pre implementáciu systému som vybral jazyk C#. Dôvodom je, že mám s programovacím jazykom najväčšie skúsenosti a myslím si, že bude vhodný pre implementáciu systému. Vývojové prostredie, v ktorom sa bude aplikácia vyvíjať, je Visual Studio 2008.

## ***2.4 Návrh architektúry systému***

Navrhovaný systém pracuje vo viacerých režimoch, kedy v každom režime sú využívané iné periférne zariadenia a spracovávané ich dáta.

### **2.4.1 Štruktúra systému**

Prvotná predstava zaoberajúca sa návrhom systému definovaná v zadaní hovorí o systéme, ktorý pracuje s parkovacou kamerou a ultrazvukovým senzorom na servery. Avšak počas testovania periférnych zariadení sa vyskytol problém, ktorý nedovoľuje komunikáciu medzi parkovacou kamerou so serverom BV-105. Problém je podrobne popísaný v kapitole *2.5.1 Implementácie funkcií kamery*. Na základe vzniknutého problému sú navrhnuté architektúry dvoch aplikácií, ktoré poskytujú požadovanú funkcionálnosť. Prvá aplikácia pracuje na stolovom počítači a pracuje s parkovacou kamerou a ultrazvukovým senzorom. Druhá aplikácia pracuje na servery a umožňuje prácu s ultrazvukovým senzorom a poskytuje podporu pre pripojenie ďalších troch senzorov. Nasledujúce kapitoly detailne popisujú návrh architektúry oboch aplikácií.

#### ***2.4.1.1 Aplikácia pre stolový počítač***

Aplikácia pre stolový počítač poskytuje prácu v troch nezávislých režimoch. Prvý režim pracuje s dvomi kamerami, z ktorých prenáša obraz. Kamery sa medzi sebou prepínajú tlačidlami.

Druhý režim pracuje s ultrazvukovým senzorom. Spracováva namerané informácie a zobrazuje ich na diódovom poli. Rovnako poskytuje číselnú informáciu o nameranej vzdialenosti v centimetroch.

Tretím režimom je duálny režim, ktorý pracuje s oboma periférnymi zariadeniami spoločne. Prenáša obraz z kamery a informuje o nameranej vzdialenosti na diódovom paneli, ktorý pozostáva z desiatich diód.

##### **Režimy aplikácie:**

- Režim parkovacej kamery
- Režim ultrazvukového senzora
- Duálny režim
- Nastavenie zvukov

#### ***2.4.1.2 Aplikácia pre server BV-105***

Aplikácia pre server pracuje v dvoch základných režimoch. Poskytuje podporu pre prácu štyroch nezávislých senzorov. Prvý režim poskytuje najkratšiu vzdialenosť zo štyroch nameraných vzdialeností. Najkratšia vzdialenosť je zobrazovaná na diódovom poli. Najkratšia vzdialenosť je číselne vyjadrená v centimetroch.

Druhý režim je režim Súhra a poskytuje informácie z každého senzoru osobitne. Informácie sú spracovávané a vyjadrená je každá nameraná vzdialenosť. Rovnako však poskytuje aj informáciu o najkratšej vzdialenosti, ktorá je zobrazená na poli desiatich diód.

Aplikácia poskytuje aj režim konfigurácie, keďže poskytuje prácu štyroch senzorov a to si vyžaduje nutnosť možnosti konfigurácie každého z nich. Režim poskytuje možnosť manuálneho vypnutia alebo zapnutia daného senzora. Umožňuje zmenu meracích rozsahov ako aj kontrolu odpojeného senzora.

Aplikácia disponuje doplnkovou funkcionalitou v podobe parkovacích stopiek, ktoré slúžia vodičovi ako merací nástroj na meranie dĺžky parkovania.

### **Režimy aplikácie:**

- Režim ultrazvukového senzora
- Režim Súhra
- Režim konfigurácie
- Režim parkovacích stopiek
- Nastavenie zvukov

Nasledujúce kapitoly podrobne popisujú návrhy jednotlivých režimov oboch aplikácií.

### **2.4.2 Návrh aplikácie pre stolový počítač**

Podkapitola presne popisuje návrh jednotlivých režimov aplikácie pre stolový počítač.

#### ***2.4.2.1 Hlavné okno aplikácie pre stolový počítač***

Po spustení programu pre stolový počítač sa na obrazovke objavia tri ikony, ktoré reprezentujú každý s režimov (režim parkovacej kamery, režim ultrazvukového senzora a duálny režim), rovnako obsahuje ikonu pre nastavenie hlasitosti. Jedná sa o zapnutie alebo vypnutie zvukov. Na obrazovke je rovnako ikona pre vypnutie aplikácie. Hlavné okno je zobrazené na *obrázku 22*.

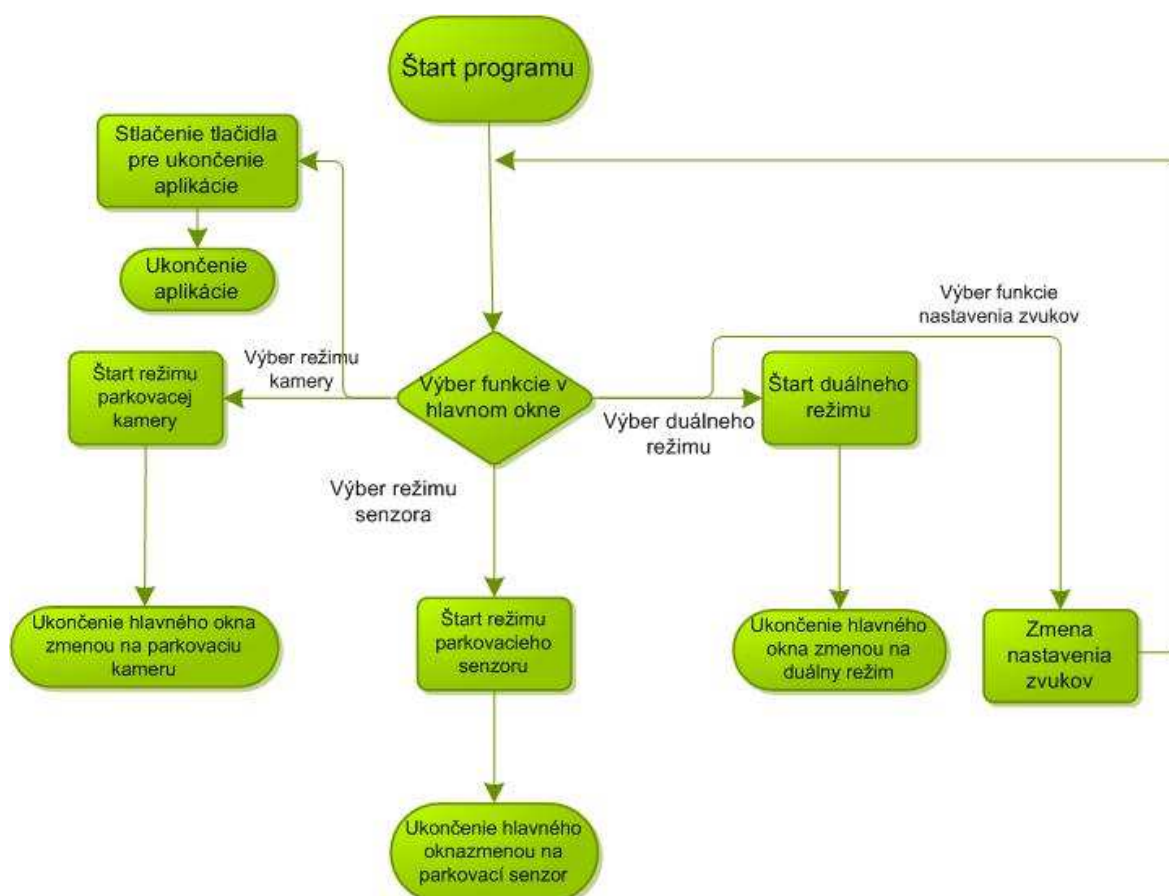
Pri návrhu hlavného okna sa kladie dôraz na intuitívne a jednoznačné ovládanie. Ikony sú dostatočne veľké a svojím vzhľadom predstavujú funkcie, ktoré spúšťajú. Prvá ikona z ľavej strany predstavuje ikonu pre spustenie režimu parkovacej kamery. Stredná ikona spúšťa režim parkovacích sensorov. Prvá ikona z pravej strany spúšťa duálny režim, kedy sa do činnosti uvedie kamera spolu so sensorom.

V hlavnom okne sa nachádza aj ikona na vypnutie alebo zapnutie zvukov v programe. Ikona je umiestnená v ľavom dolnom rohu. Ikona pre vypnutie aplikácie sa nachádza v pravom dolnom rohu. Hlavné okno má základné rozlíšenie 720 x 480 pixlov.

Diagram na *obrázku 23* predstavuje funkcionality hlavného okna aplikácie pre stolový počítač.



Obr. 22 Hlavné okno programu pre stolový počítač



Obr. 23 Stavový diagram hlavného okna pre stolový počítač

#### 2.4.2.2 Režim parkovacej kamery

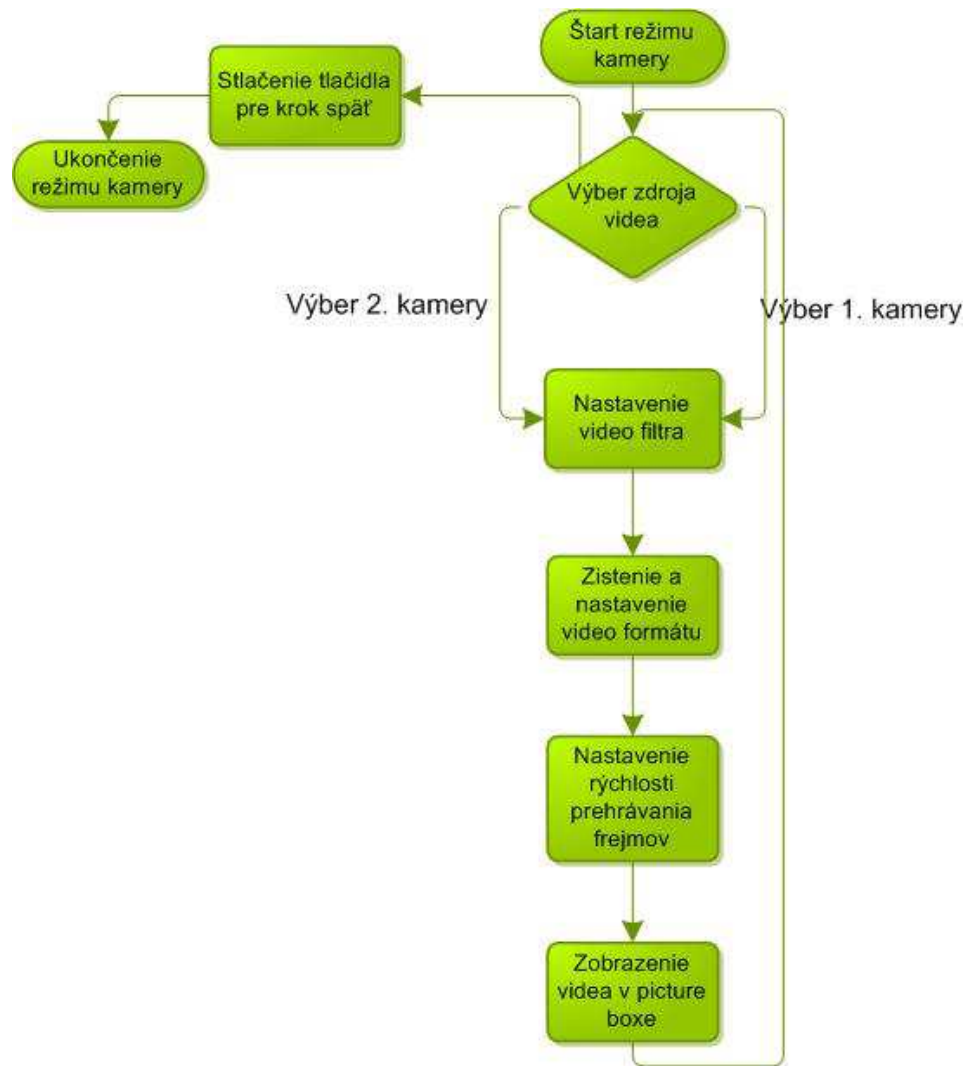
Režim parkovacej kamery sa aktivuje kliknutím na ľavú ikonu hlavného okna na *obrázku 22*. Po aktivovaní režimu sa začne prehrávanie videa na zobrazovacom poli z kamery 1. Po kliknutí na ikonu *kamera 2* sa vypne prehrávanie videa z kamery 1 a spustí sa prehrávanie z kamery 2. V režime parkovacej kamery je dostupné aj tlačidlo kroku späť. Po kliknutí na tlačidlo späť sa ukončí prehrávanie videa a zobrazí sa hlavné okno.



**Obr. 24** Okno režimu parkovacej kamery

*Obrázku 24* presne popisuje návrh okna režimu kamery. V ľavej hornej časti obrazovky sa nachádza identifikácia režimu. Väčšiu časť obrazovky predstavuje zobrazovacie pole, ktoré slúži na prehrávanie videa, umiestnené v strednej časti obrazovky. Pod zobrazovacím poľom v strednej spodnej časti obrazovky sa nachádzajú prepínače kamier. Pre ukončenie režimu a vrátenie sa na hlavné okno je v pravom dolnom rohu ikona pre krok späť.

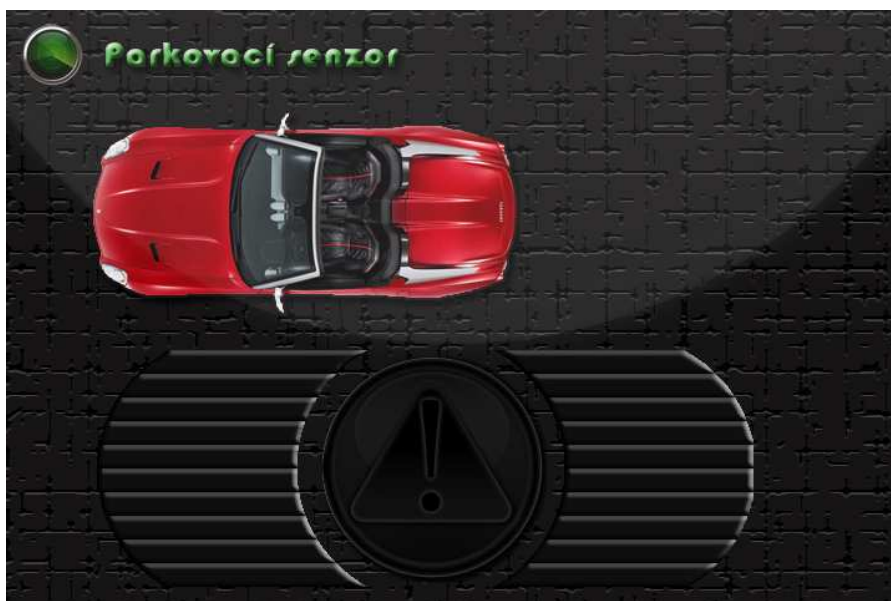
Po spustení režimu parkovacej kamery sa automaticky zistia pripojené zdroje videa. Plánuje sa s prácou dvoch kamier. Aplikácia primárne nastavuje ako zdroj videa prvú kameru, kameru 1. Po výbere zdroja videa sa nastaví grafické filtre, ktoré nastaví parametre obrazu, jedná sa o parametre ako kontrast, gama korekcia, korekcia farieb. Po nastavení jednotlivých filtrov sa zistí dostupné rozlíšenie, ktoré kamera poskytuje a nastaví sa vhodné rozlíšenie pre zobrazenie do pripraveného zobrazovacieho poľa na *obrázku 24*. Rovnako sa musí nastaviť rýchlosť prehrávania jednotlivých obrazov. Rýchlosť nastavenia závisí od rýchlosti spracovania videa počítačom. Posledným krokom je zobrazenie videa v pripravenom zobrazovacom poli, čo predstavuje výstup režimu. Režim poskytuje vizuálny prehľad o situácii za vozidlom. Režim je možné opustiť v akomkoľvek čase, kliknutím na ikonu späť. Prehrávanie videa sa tým ukončí a adaptér sa uzatvorí.



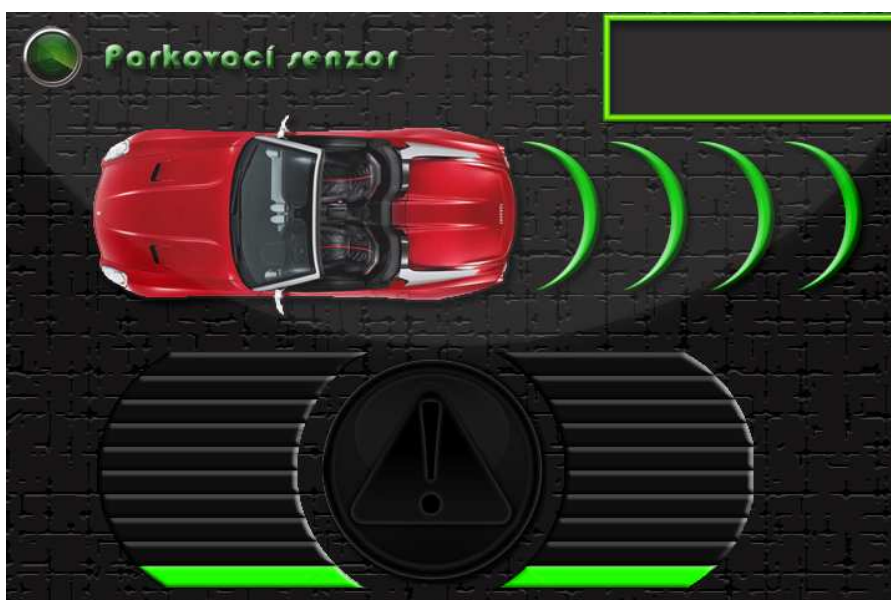
**Obr. 25** Stavový diagram režimu parkovacej kamery

### 2.4.2.3 Režim ultrazvukových senzorov

Funkcionalita a návrh režimu sú spoločné pre aplikáciu na stolovom počítači ako aj pre aplikáciu určenú pre server. Režim ultrazvukových senzorov sa aktivuje kliknutím na prostrednú ikonu hlavného okna aplikácie pre stolový počítač zobrazeného na *obrázku 22*. Po spustení režimu sa na obrazovke zobrazí model automobilu a diódový panel, ktoré slúžia na interpretáciu nameraných údajov z ultrazvukového senzora. Vizuálna informácia je sprevádzaná akustickou signalizáciou ako aj varovnou signalizáciou v prípade potreby. Rozmiestnenie jednotlivých zobrazovacích elementov je zobrazené na *obrázku 26* a *obrázku 27*.



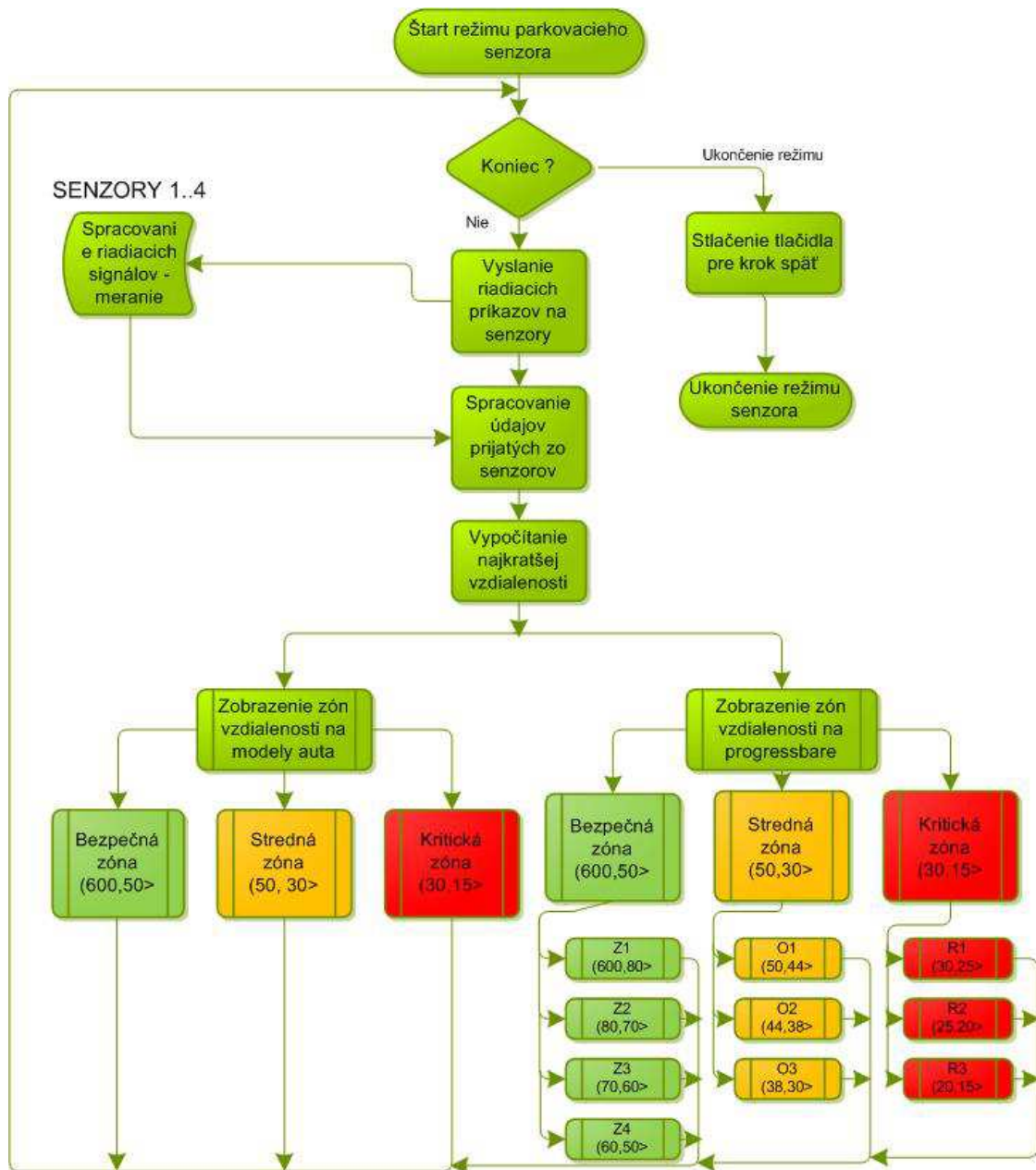
**Obr. 26** Zobrazenie okna pre prácu so senzorom(pasívny senzor)



**Obr. 27** Zobrazenie okna pre prácu so senzorom(aktívny senzor)

Obrázok 27 navyše znázorňuje pohľad v prípade kedy sa senzor aktivuje a sníma vzdialenosť. Aktívny režim je charakterizovaný vlnami, ktoré sa zobrazujú za modelom automobilu. Na diódovom paneli sa rozsvetujú diódy v závislosti na vzdialenosti. V pravom hornom rohu pribudlo okno, v ktorom sa zobrazuje vzdialenosť od vozidla prekážky v centimetroch. Jednotlivé meracie rozsahy ako aj kritické hlásenia sú podrobne popísané v nasledujúcich riadkoch. Na obrázku 28 je zobrazený stavový diagram režimu ultrazvukového senzora.





**Obr. 28** Stavový diagram režimu parkovacieho asistenta

Po aktivácii režimu pre prácu s ultrazvukovým senzorom sa nastaví parametre komunikácie. Nastaví sa port, rýchlosť komunikácie a synchronizácia. Ďalej sa nastaví adresa pre podriadené zariadenie, čiže pre senzor, na 0xE0. Na adresu sa následne pošle štart bit pre nadviazanie komunikácie. Senzor odpovie potvrdením. Následne sa do registra R0 pošle riadiaci príkaz, ktorým sa inicializuje senzor pre meranie vzdialenosti. Senzor následne pošle údaje potrebné pre spracovanie. Údaje získané zo senzora sa spracovávajú pre dva typy zobrazení. Prvý typ zobrazenia je farebné rozlíšenie vln za modelom automobilu. Druhý typ zobrazenia je rozsvetovanie diód na diódovom paneli v spodnej časti obrazovky. Diódový panel je zobrazený na *obrázku 29*. Simulácia ďalších



troch senzorov je zabezpečená algoritmom, ktorý generuje náhodné čísla v požadovanom rozsahu. Tie sú následne priradené k ostatným senzorum ako namerané vzdialenosti. Všetky namerané a vygenerované hodnoty sa vyhodnotia a výstupom bude najkratšia vzdialenosť. Funkcie sú však využívané v aplikácii pre server BV-105.

Program rozlišuje tri hlavné zóny vzdialenosti:

- zelená zóna (vzdialenosť väčšia ako 50cm)
- oranžová zóna (vzdialenosť medzi 50cm a 30cm)
- kritická červená zóna (vzdialenosť medzi 30cm a 15cm)

Farba zobrazovaných vln za modelom automobilu sa zobrazuje v závislosti od vzdialenosti podľa jednotlivých zón. Každá zóna sa však delí na jednotlivé menšie zóny, ktoré sa zobrazujú na diódovom paneli a poskytujú lepší prehľad o vzdialenosti prekážky od vozidla.

Delenie zelenej zóny:

Zelená zóna sa delí na menšie štyri zóny. Zelená zóna je znázornená na *obrázku 29*.

- Z1 (vzdialenosť väčšia ako 80cm)
- Z2 (vzdialenosť medzi 80cm a 70cm)
- Z3 (vzdialenosť medzi 70cm a 60cm)
- Z4 (vzdialenosť medzi 60cm a 50cm)



**Obr. 29** Zelená zóna

Delenie oranžovej zóny:

Oranžová zóna sa delí na tri menšie zóny. Oranžová zóna je zobrazená na *obrázku 30*.

- O1 (vzdialenosť medzi 50cm a 44cm)
- O2 (vzdialenosť medzi 44cm a 38cm)
- O3 (vzdialenosť medzi 38cm a 30cm)



**Obr. 30** Oranžová zóna

Upozornenie o vzdialenosti je reprezentované nie len vizuálne ale rovnako akustickou signalizáciou. Pre zelenú a oranžovú zónu je akustická signalizácia reprezentovaná zvukom pípania, ktoré sa vykonáva s väčšou časovou medzerou.

Delenie červenej zóny:

Červená zóna sa rovnako delí na tri menšie zóny ako oranžová zóna. Červená zóna je zobrazená na *obrázku 31*.

- R1 (vzdialenosť medzi 30cm a 25cm)
- R2 (vzdialenosť medzi 25cm a 20cm)
- R3 (vzdialenosť medzi 20cm a 15cm)

Upozornenie je v červenej zóne rovnako reprezentované aj akustickou signalizáciou, ktorá je však omnoho rýchlejšia ako v predchádzajúcich dvoch zónach. Zvýšenú pozornosť vodiča má navodiť upozornenie, ktoré sa zobrazuje v strednej časti diódového pola. Upozornenie je znázornené na *obrázku 32*.



**Obr. 31** Červená zóna



**Obr. 32** Upozornenie

Navrhnuté riešenie režimu ultrazvukového senzora poskytuje dostatočné informácie o vzdialenosti prekážky od vozidla a vodiča dostatočne a v dostatočnom čase na ňu upozorní.

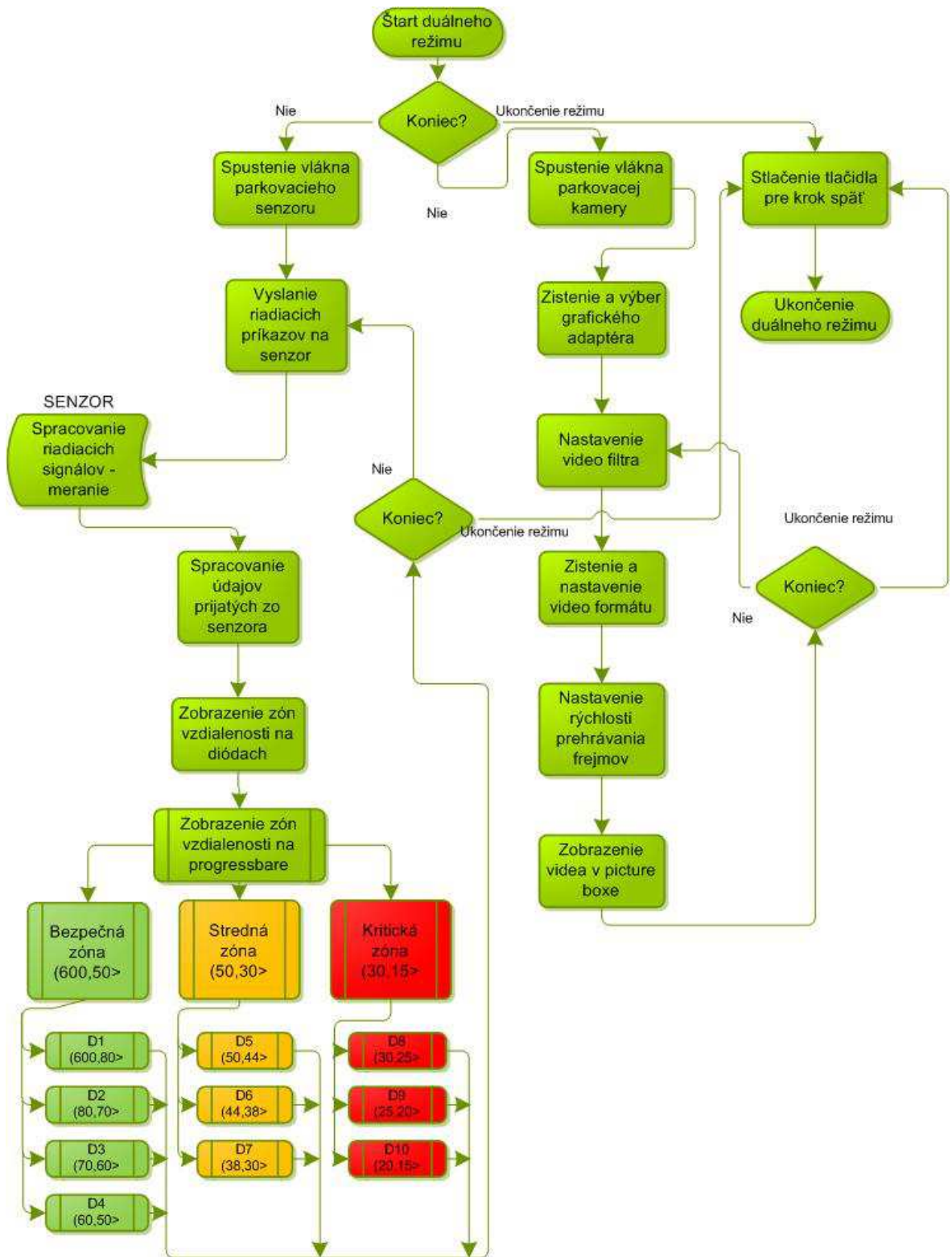
Ukončenie režimu sa uskutoční stlačením tlačidla späť v pravom dolnom rohu, kedy sa ukončí komunikácia so sensorom a nastaví sa hlavné okno.

#### **2.4.2.4 Duálny režim**

Duálny režim sa aktivuje kliknutím na pravú ikonu hlavného okna aplikácie pre stolový počítač. Duálny režim je zložený už z existujúcich režimov, čo vlastne vyplýva aj z jeho názvu. Jedná sa o kombináciu parkovacej kamery a parkovacieho senzoru.

Okno duálneho režimu pozostáva zo zobrazovacieho poľa, ktoré slúži na zobrazenie videa a sledovanie situácie za vozidlom a z desiatich diód, ktorých úlohou je informovanie vodiča o vzdialenosti vozidla od prekážky. Okno duálneho režimu je zobrazené na *obrázku 34*.

Režim je navrhnutý z dôvodu maximálneho využitia funkcionality programu a maximálnej schopnosti informovať vodiča o situácii za vozidlom. Pri spustení režimu sa spustia dve nezávislé vlákna. Každé vlákno pracuje s periférnym zariadením. Na prvom vlákne sa inicializuje kamera, nastaví sa filtre, rozlíšenie videa a všetky potrebné nastavenia. Video sa spustí. Súčasne sa na druhom vlákne nastaví parametre komunikácie so sensorom a začne sa komunikácia a meranie vzdialenosti. Informácie z parkovacieho senzora sú interpretované na desiatich diódach, čo je zrejmé z *obrázku 34*.



Obr. 33 Stavový diagram duálneho režimu



**Obr. 34** Okno duálneho režimu

Jednotlivé meracie zóny, v ktorých sú rozsvetované jednotlivé diód sú rovnaké ako v režime ultrazvukového senzora v kapitole 2.4.2.3 *Režim ultrazvukových senzorov*.

- Zelená zóna (vzdialenosť väčšia ako 50cm)
  - D1 (vzdialenosť väčšia ako 80cm)
  - D2 (vzdialenosť medzi 80cm a 70cm)
  - D3 (vzdialenosť medzi 70cm a 60cm)
  - D4 (vzdialenosť medzi 60cm a 50cm)
- Oranžová zóna (vzdialenosť medzi 50cm a 30cm)
  - D5 (vzdialenosť medzi 50cm a 44cm)
  - D6 (vzdialenosť medzi 44cm a 38cm)
  - D7 (vzdialenosť medzi 38cm a 30cm)
- Červená zóna (vzdialenosť medzi 30cm a 15cm)
  - D8 (vzdialenosť medzi 30cm a 25cm)
  - D9 (vzdialenosť medzi 25cm a 20cm)
  - D10 (vzdialenosť medzi 20cm a 15cm)

Upozorňovanie vodiča je rovnako vykonávané akustickou signalizáciou ako v režime ultrazvukového senzora.

Ukončenie režimu sa uskutočňuje stlačením ikony kroku späť v pravom dolnom rohu obrazovky. Ukončí sa komunikácia so senzorom a rovnako sa uzavrie adaptér videa a nastaví sa hlavné okno.

#### **2.4.2.5 Zmena nastavenia zvukov**

Zmena nastavenia zvukov je dostupná pre obe aplikácie. Pre aplikáciu stolového počítača ako aj pre aplikáciu na server. Nastavenie hlasitosti pozostáva len v možnosti zapnúť/vypnúť zvuky, ktoré sprevádzajú režimy ultrazvukového senzora, duálny režim a režim súhry. Funkcia je navrhnutá z dôvodu, že nie vždy je vhodné akusticky ohlasovať prekážku za vozidlom. Je to hlavne v situáciách, kedy sa v automobile vezú malé deti, ktoré spia alebo akákoľvek iná osoba, ktorú by to rušilo. Nastavenie sa mení jednoduchým kliknutím na ikonu reproduktora v ľavom dolnom rohu hlavného okna. Ikony nastavenia sú znázornené na *obrázku 35*.



**Obr. 35** Ikony pre nastavenie zvukov

### **2.4.3 Návrh aplikácie pre server**

Podkapitola presne popisuje návrh jednotlivých režimov aplikácie pre server BV-105.

#### **2.4.3.1 Hlavné okno aplikácie pre server**

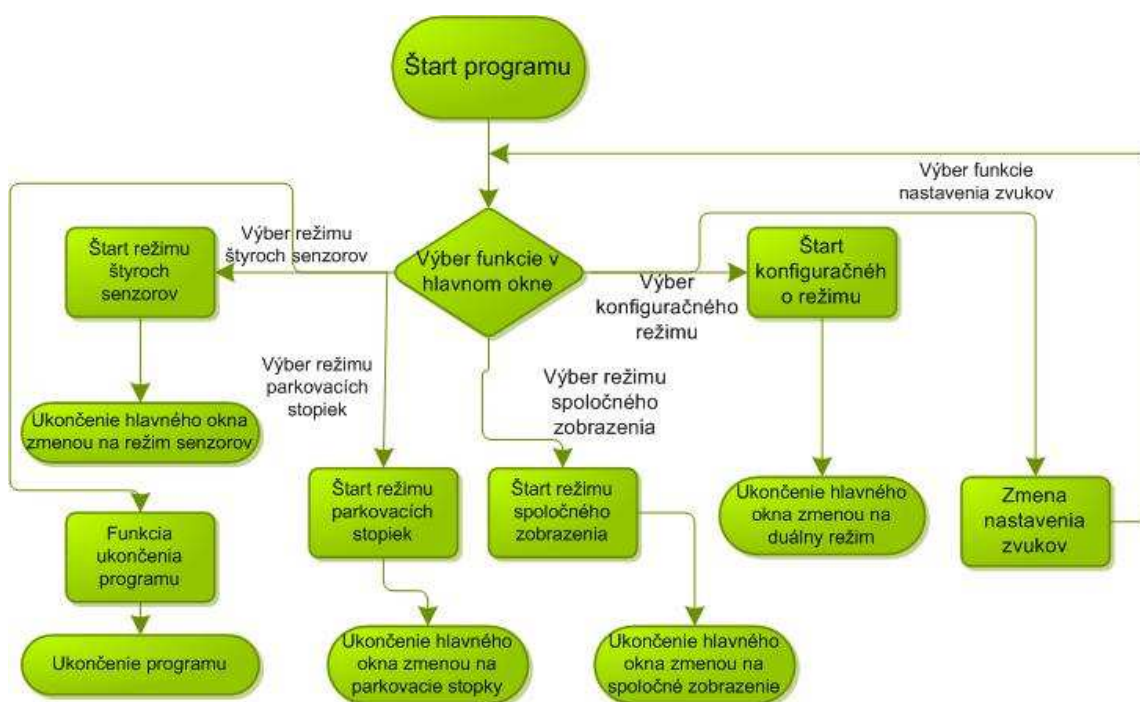
Na *obrázku 36* je zobrazené hlavné okno aplikácie na server. Oproti návrhu hlavného okna aplikácie pre stolový počítač pracuje v dvoch režimoch, ako bolo popísané v kapitole *2.4.1.2 Aplikácia pre server*. Prvý režim, režim ultrazvukových senzorov pracuje s ultrazvukovým senzorom a poskytuje podporu pre pripojenie ďalších troch senzorov. Výstupom režimu je najkratšia nameraná vzdialenosť zobrazovaná na diódovom paneli. Druhý režim, režim súhra podporuje prácu štyroch nezávislých senzorov. Výstupom režimu je zobrazenie štyroch nameraných hodnôt. Informácie sú zobrazované individuálne z každého senzora.

Hlavné okno poskytuje možnosť nastavenia zvukov v podobe zapnutia a vypnutia hlasitosti. Aplikácia bola doplnená o možnosť konfigurácie senzorov a možnosť merania doby parkovania pomocou parkovacích stopiek. Na *obrázku 37* je zobrazený stavový diagram hlavného okna programu pre server. Ukončenie aplikácie sa vykoná kliknutím na ikonu v pravej spodnej časti obrazovky.





Obr. 36 Hlavné okno programu pre server



Obr. 37 Stavový diagram hlavného okna pre server

#### 2.4.3.2 Režim ultrazvukových senzorov

Režim ultrazvukových senzorov je rovnaký v aplikácii pre stolový počítač ako aj pre aplikáciu na server. Rozdielom je však podpora štyroch ultrazvukových senzorov v aplikácii pre server. Rozloženie okna je totožné s režimom ultrazvukového senzora popísaného v kapitole 2.4.2.3 Režim ultrazvukového senzora na obrázkami 26 a 27. Zachované ostali aj meracie rozsahy a vizuálna aj akustická signalizácia.

### 2.4.3.3 Režim súhra

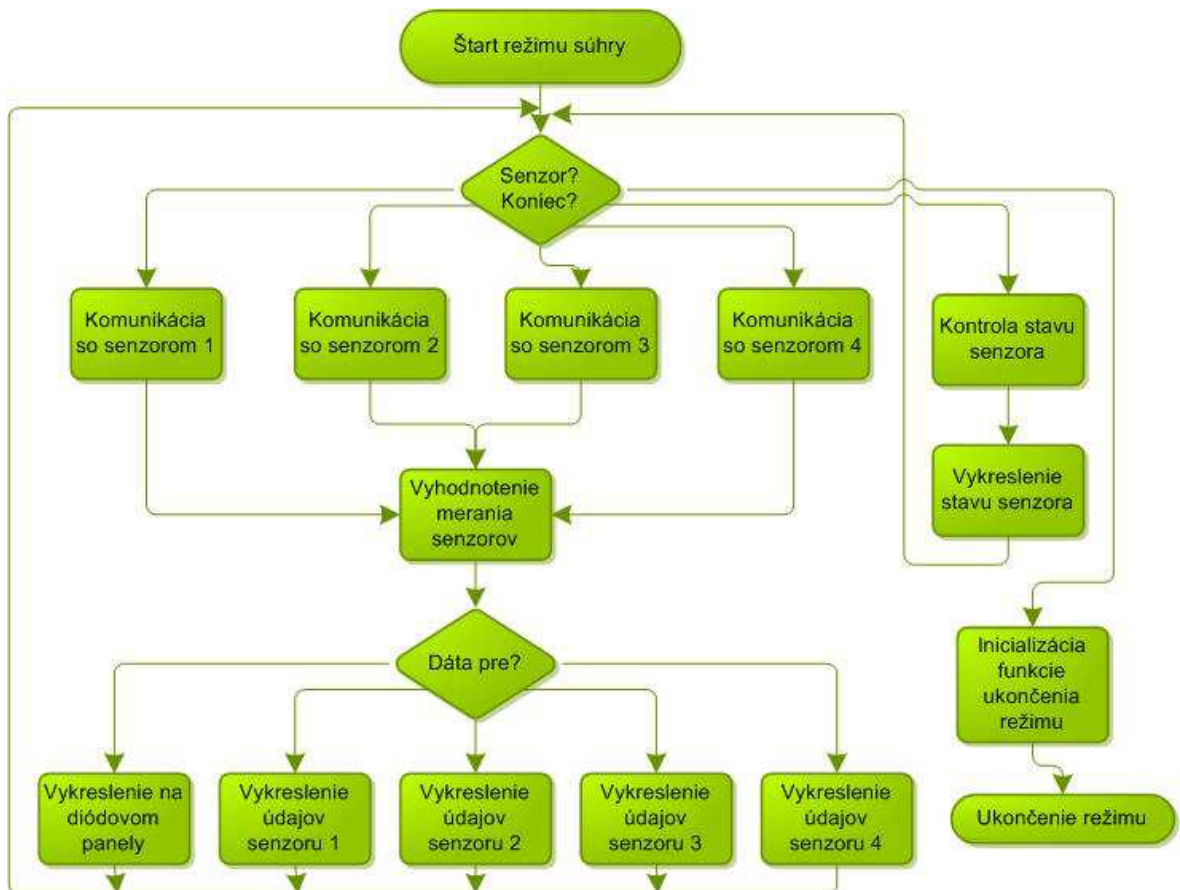
Režim sa aktivuje kliknutím na pravú ikonu hlavného okna pre aplikáciu na server na *obrázku 36*. Režim poskytuje zobrazenie štyroch ultrazvukových senzorov na vozidle. Každý zo senzorov poskytuje informáciu o nameranej vzdialenosti od prekážky a poskytuje ju v prehľadnej a zrozumiteľnej podobe. Tento pohľad poskytuje vodičovi maximálny prehľad nad situáciou, ktorá sa naskytá za vozidlom počas parkovania. Avšak okrem individuálneho zobrazenia informácie o vzdialenosti je poskytovaná aj komplexne najkratšia vzdialenosť zobrazovaná na diódovom paneli, ktorý pozostáva z desiatich diód. Jednotlivé rozsahy v základnom nastavení sú rovnaké ako v režime ultrazvukových senzorov v *kapitole 2.4.2.3 Režim ultrazvukových senzorov*. Okno režimu je zobrazené na *obrázku 38*.

Režim rovnako poskytuje pohľad na stav senzorov. Každý senzor môže nadobudnúť tri stavy. Prvý stav je stav aktívny, kedy je senzor v činnosti. Dióda stavu svieti na zeleno. Druhým stavom, ktorý môže nadobúdať senzor je stav odpojený používateľom, kedy senzor nevykonáva meranie a dióda stavu svieti na červeno. Posledným stavom, ktorý môže senzor nadobudnúť je stav odpojený, kedy nevykonáva činnosť a nie je ho ani možné zapnúť. Tento stav vychádza z poznatku, kedy pri inicializácii systém nenájde príslušný senzor. Dióda stavu v tomto prípade svieti na modrú farbu. Príklad zobrazenia každého zo stavu senzorov je zobrazený na *obrázku 38* v ľavej strednej časti. Ukončenie činnosti režimu sa inicializuje stlačením tlačidla pre návrat.



**Obr. 38** Grafické okno režimu súhra



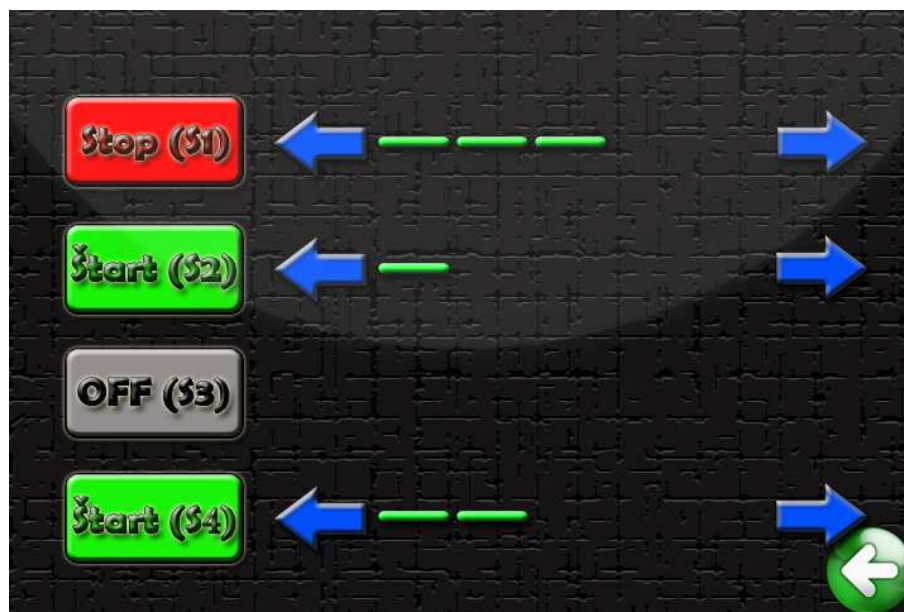


**Obr. 39** Stavový diagram režimu súhry

Po inicializácii režimu sa spustí komunikácia s každým senzorom osobitne kedy sa vyšlú riadiace signály na každý senzor a čaká sa informáciu o nameranej vzdialenosti. Po prijatí informácií o nameranej vzdialenosti sa tieto pošlú ako parametre funkcií vyhodnotenia a spracovania. Následne sa výsledky pošlú ako riadiace údaje pre funkcie, ktoré riadia rozsvetovanie diódových panelov pre jednotlivé senzory ako tak aj pre pole diód pre najkratšiu vzdialenosť. Spôsob činnosti režimu je znázornený na obrázku 39 v podobe stavového diagramu.

#### 2.4.3.4 Režim konfigurácie senzorov

Účel režimu konfigurácie senzorov je poskytnúť možnosť používateľovi zmeniť základné nastavenia senzorov. Umožňuje meniť rozsahy meraných vzdialeností každého senzoru individuálne. Rovnako tak umožňuje vypnúť alebo zapnúť senzor, ktorý si používateľ vyberie. Režim poskytuje informácie o senzoroch, ktoré nie sú v danom momente dostupné v systéme a automaticky je zablokovaná ich akákoľvek konfigurácia. Pohľad na režim je zobrazený na obrázku 40.



**Obr. 40** Grafické okno konfiguračného režimu

Ovládanie senzorov sa vykonáva štyrmi veľkými ikonami v ľavej časti obrazovky. Sensory môžu nadobudnúť tri základné stavy. Aktívny stav, kedy je tlačidlo zobrazené červenou farbou a ponúka možnosť zmenu stavu, na stav vypnutý používateľom. Príkladom je prvé tlačidlo. Ďalším stavom je stav, kedy je senzor vypnutý používateľom. Pri takom stave je možnosť znovu spustenia senzora, tlačidlá svietia zelenou farbou. V našom prípade senzor 2 a senzor 4. Posledným stavom je fyzické odpojenie senzora, ktoré je zistené pri spustení systému a v takom prípade nie je možné vykonávať žiadne nastavenia a tlačidlo senzora svieti šedou farbou.

Konfigurácia ponúka zmenu základných meraných rozsahov jednotlivých senzorov. Každý rozsah je možné meniť tlačením modrých tlačidiel v podobe šípk pri príslušnom senzore. Po kliknutí na ľavé šípky sa znižuje meraný rozsah a celý rozsah sa posúva k nulovej hodnote. Po kliknutí na pravé šípky sa zväčšuje meraný rozsah a celý sa posúva k maximálnej hodnote. Každý prírastok hodnoty k základnému rozsahu predstavuje nárast o 10cm v základe. V základe preto, lebo ku každému rozsahu sa pripočíta určitá časť tejto hodnoty a v konečnom dôsledku je prírastok celého rozsahu exponenciálny. Veľkosť prírastu je znázornená v *tabuľke 4*. Premenná „X“ predstavuje základnú hodnotu prírastku. Minimálne 10cm a maximálne 50cm.

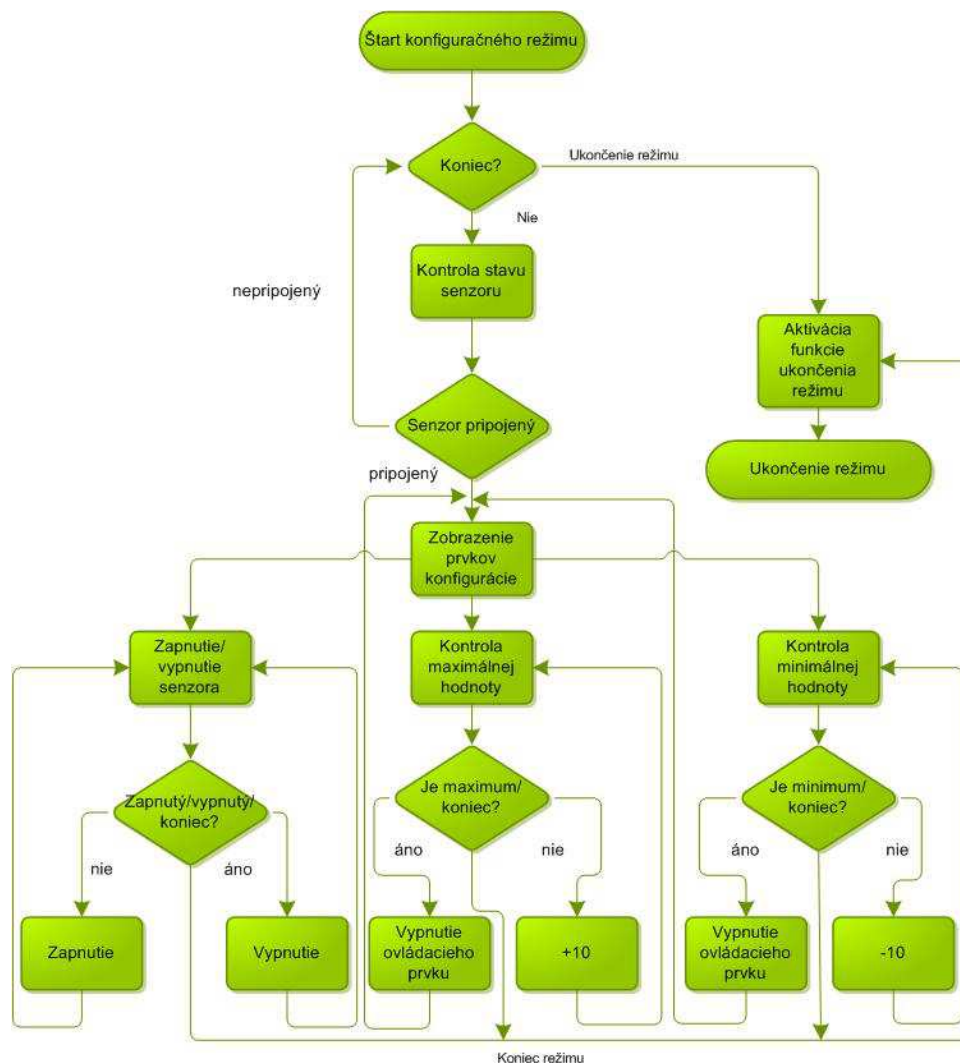
**Tab. 4** Zmena rozsahu meranej vzdialenosti

Horná hranica	Dolná hranica	Zóna
-	$80 + (X*1,2)$	Z1
$80 + (X*1,2)$	70	Z2
70	$60 + (X/2)$	Z3
$60 + (X/2)$	$50 + (X/3)$	Z4
$50 + (X/3)$	$44 + (X/4)$	O1
$44 + (X/4)$	$38 + (X/5)$	O2
$38 + (X/5)$	$30 + (X/6)$	O3
$30 + (X/6)$	$25 + (X/7)$	Č1
$25 + (X/7)$	$20 + (X/8)$	Č2
$20 + (X/8)$	0	Č3

Z tabuľky 4 je teda zrejmé, že ak si používateľ nakonfiguruje senzor na meranú vzdialenosť tým, že nastaví tri paličky, v podstate nastaví základ na 30cm. Tie sú následne prepočítané na priradenie do príslušnej zóny. Ak by sme si to rozpisali postupne tak pre zónu „Č3“ by sme dostali prírastok  $30/8$  čo sa rovná 3,75cm. Zóna „Č2“ má prírastok  $30/7$  čo je 4,28cm a zóna by sa posunula nahor o  $30/8$ . Zóna „Č1“ má prírastok  $30/6$  a celá sa posunie o  $30/7$ . Takto podľa tabuľky 4 vykonáme posun a prírastok až k zóne „Z1“, kde máme prírastok až o  $30*1,2$ . Na tomto príklade vidíme, že prírastok nie je konštantný. Kým sa rozsah zóny „Č3“ navýši len o 3,75cm tak zóna „Z1“ sa navýši o 36cm. Tento spôsob je zvolený z dôvodu, aby nedošlo k veľkému rozťahovaniu červených zón a aby sa zachovala citlivosť týchto zón. Ale na druhej strane, aby sa zväčšil aj meraný rozsah ostatných zón.

Stavový diagram na obrázku 41 popisuje spôsob činnosti konfiguračného režimu. Diagram je vytvorený pre ukážku konfigurácie jedného senzora. Pre každý ďalší senzor je spracovanie rovnaké. Po inicializácii režimu sa kontroluje pripojenie senzora, od ktorého sa následne odvíja ďalšie spracovanie. Ak je senzor neaktívny, ďalšie nastavenia nie sú podporované. Avšak ak je senzor pripojený, zobrazia sa prvky konfigurácie. Počas možnosti konfigurácie sa kontrolujú stavy, či je senzor aktívny alebo neaktívny. Kontrolujú sa aj nastavenia prírastkov a ich maximálne hodnoty, aby nedošlo k prekročeniu povolených hodnôt.

Režim sa ukončuje stlačením tlačidla pre návrat späť rovnako ako v ostatných režimoch.

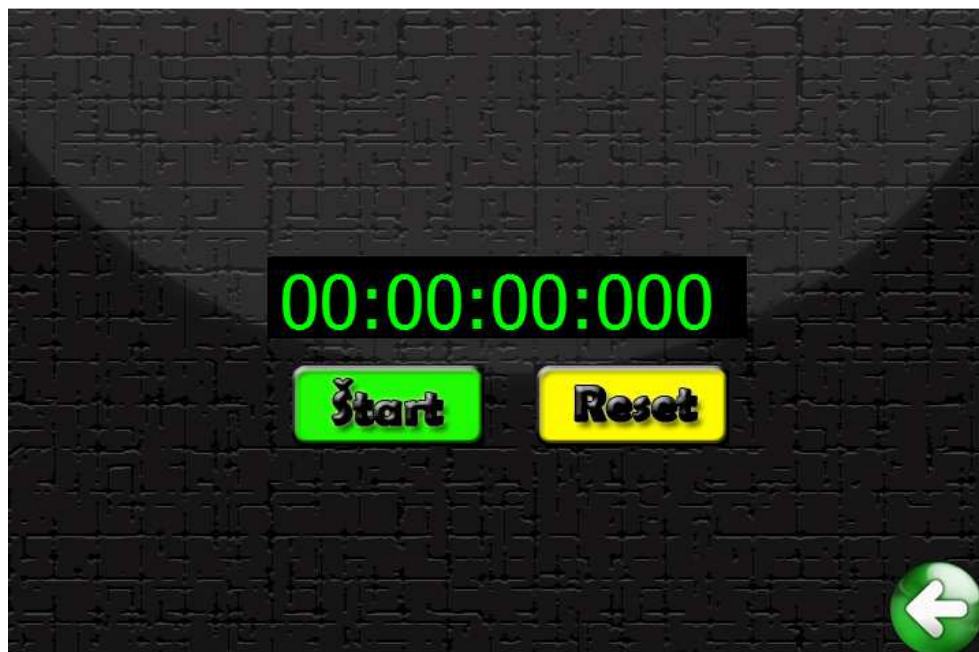


**Obr. 41** Stavový diagram konfiguračného režimu pre jeden senzor

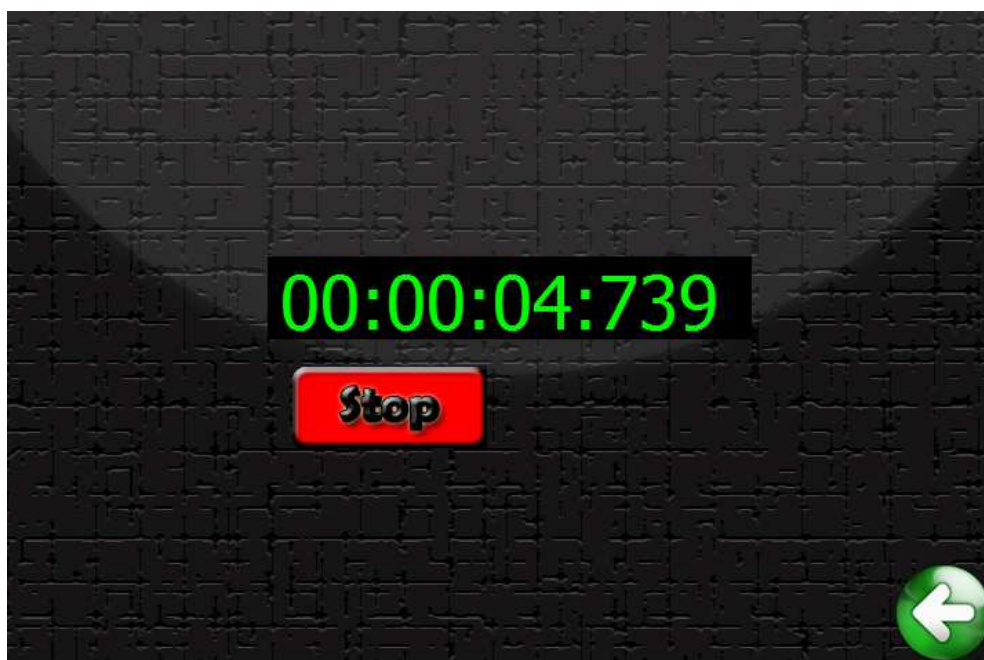
#### 2.4.3.5 Parkovacie stopky

Funkcia parkovacích stopiek nie je nevyhnutná pre samotné parkovanie, ale predstavuje doplnkovú funkcionalitu, ktorá používateľovi poskytne presnú informáciu o tom, ako dlho mu trvá zaparkovanie na určité miesta.

Po spustení režimu sa na obrazovke zobrazí časomiera a dve tlačidlá. Po kliknutí na tlačidlo „Štart“ sa spustí sekvencia počítania, ktoré sa zobrazuje na časomiere. Následne sa režim opustí a využívajú sa funkcie parkovania s tým, že meranie beží na vlastnom vlákne. Po ukončení parkovania sa vrátíme späť do režimu stopiek a zastavíme počítanie tlačidlom „Stop“. Na časomiere nám ostane výsledok počítania. Ten môžeme následne vynulovať tlačidlom „Reset“ alebo pokračovať ďalej v počítaní. Na *obrázkoch 42 a 43* môžeme vidieť stopky v režime inicializácie ako aj počas počítania. Režim parkovacích stopiek sa opúšťa kliknutím na ikonu opustenia režimu.



Obr. 42 Stopky v čase resetu



Obr. 43 Stopky v čase počítania

## 2.5 Implementácia systému

Kapitola popisuje najdôležitejšie funkcie, ktoré zabezpečujú hlavné a špecifické vlastnosti systému, rovnako podrobne popisuje jednotlivé časti systému. Systém je vyvíjaný v prostredí Visual Studio 2008 v programovacom jazyku C#. Systém je vyvíjaný pre typ projektu, ktorý podporuje mobilné zariadenia. Pracuje so základnými knižnicami, ktoré sú dostupné vývojovým prostredím a programovacím jazykom. Základné knižnice, ktoré podporujú prácu s COM portami, knižnice podporujúce tvorbu vlákien, knižnice pracujúce s médiami, knižnice podporujúce tvorbu grafického používateľského rozhrania.

Implementácia vychádza z návrhu riešenia, ktorý je podrobne popísaný v kapitole 2.4 *Návrh architektúry systému*.

### 2.5.1 Implementácia funkcií kamery

Prvotná predstava, ktorá vychádzala zo zadania diplomového projektu, zahŕňala implementáciu kamery na samotný server. Poznatky získané počas analýzy problémovej oblasti kamier tvrdia, že kamera musí byť typu UVC (USB VIDEO CLASS) z dôvodu, aby umožňovala prácu s funkciami, ktoré ju dokážu v systéme nájsť a pristupovať k nej. Server pracuje s operačným systémom Windows CE verzie 6. Vyskytol sa problém, ktorý predstavoval nefunkčnú komunikáciu kamery so serverom. Pri testovaní komunikácie kamery a servera boli použité ovládače odporúčané spoločnosťou Microsoft[27], tie však nepracovali korektne. Použité boli taktiež ovládače, ktoré vytvorili komerčné spoločnosti venujúce sa danej problematike, no neúspešne. Každý ovládač bol testovaný na mnou vytvorených funkciách. Hlavným cieľom bolo otestovanie a zabezpečenie funkčnej komunikácie kamery a servera. Neúspešne.

Na základe vzniknutých problémov sme sa rozhodli implementovať funkcionality kamery na stolový počítač. Kamera je pre operačný systém Windows 7 plne podporovaná a dokáže s ním pracovať korektne. Počas implementácie boli použité štandardné knižnice poskytujúce funkcie spracovania videa. Takzvané „Direct Show“ knižnice. Funkcie, ktoré sú použité počas implementácie sú popísané v kapitole 2.5.4 *Funkcie režimu kamery*.

Rozhodnutie implementovať funkcie kamery na stolový počítač viedlo k vytvoreniu dvoch aplikácií. Aplikácia pre stolový počítač a aplikácia pre server. Návrh oboch aplikácií je popísaný v kapitole 2.4 *Návrh architektúry systému*.

## 2.5.2 Štruktúra systému

Systém je implementovaný s ohľadom na jednoduchosť, ovládateľnosť a intuitívnosť. Pracuje vo viacerých režimoch, v ktorých využíva periférne zariadenia kamery a ultrazvukového senzora. Do každého režimu sa dostaneme z hlavného okna. Vyvinul som dve nezávislé aplikácie, z dôvodu nefunkčnosti kamery na servery ako som spomenul už počas návrhu v kapitole 2.4 *Návrh architektúry systému*. Problém je vysvetlený v kapitole 2.5.1 *Implementácia funkcií kamery*. Prvá aplikácia je vyvinutá pre stolový počítač a poskytuje tri režimy a možnosti nastavenia.

### *Hlavné okno (PC aplikácia)*

- Režim parkovacej kamery
- Režim ultrazvukového senzora
- Duálny režim
- Zmena nastavení zvukov
- Ukončenie aplikácie

Druhú aplikáciu som vyvinul pre mobilné zariadenie, čo vyplýva zo špecifikácie a zadania projektu. Aplikácia poskytuje podporu pre možnosť pripojenia štyroch nezávislých ultrazvukových senzorov. Pracuje v dvoch režimoch a poskytuje rôzne možnosti nastavení senzorov ako aj možnosť využitia parkovacích stopiek.

### *Hlavné okno (server aplikácia)*

- Režim ultrazvukového senzora
- Režim súhra
- Zmena nastavení zvukov
- Konfigurácia senzorov
- Parkovacie stopky

Základná a hlavná trieda, ktorá obsahuje všetky funkcie sa nazýva `public partial class Form1 : Form`.

## 2.5.3 Funkcie hlavného okna

Hlavné okno definuje funkcionálnosť, vďaka ktorej je možné ovládať funkcie, ktoré vykonávajú špecifickú komunikáciu s periférnymi zariadeniami a riadia funkcie spracovania prijatých údajov:

- `void camera_Click(object sender, EventArgs e)`
- `void senzor_Click(object sender, EventArgs e)`
- `void dual_Click(object sender, EventArgs e)`



- void mute\_on\_Click(object sender, EventArgs e)
- void mute\_off\_Click(object sender, EventArgs e)
- void config\_Click(object sender, EventArgs e)
- void stopky\_btn\_Click(object sender, EventArgs e)
- void setupCommPort(String comport)
- void exit\_Click(object sender, EventArgs e)

Funkcia void camera\_Click(object sender, EventArgs e) zabezpečuje hlavnú komunikáciu s pripojenými kamerami. Po inicializácii funkcie sa hlavné okno prepne na režim parkovacej kamery. Systém nájde dostupné kamery, ktoré sa následne uložia do systému pre ďalšie spracovanie. Prvá kamera sa začne využívať ako zdroj videa, kedy sa nastaví video filter a začnú sa vytvárať video obrázky, ktoré sú následne premietané do pripraveného zobrazovacieho poľa. Za jednu sekundu sa vytvorí a zobrazí 25 obrazov, čo plne postačuje pre prácu parkovacieho asistenta. Režim poskytuje dve tlačidlá, ktoré umožňujú prepínať medzi uloženými kamerami. Po kliknutí na aktiváciu druhej kamery sa ukončí vytváranie obrazov prvej kamery, zruší sa video filter. Ten sa nastaví na druhú kameru a začne vytvárať obrázky a premietat' video. Režim je možné ukončiť kliknutím na tlačidlo ukončenia.

Funkcia void senzor\_Click(object sender, EventArgs e) zabezpečuje hlavnú funkcionálnosť pri komunikácii ultrazvukového senzora so serverom. Po inicializácii funkcie sa vypne zobrazenie hlavného okna a zobrazia sa nástroje pre zobrazenie meranej vzdialenosti. Otvorí sa COM port, nastaví sa riadiace príkazy, ktoré sú poslané do senzora (register 0) pre inicializáciu merania. Zo senzora nie je možné čítať 65μs dáta pokiaľ senzor nespracuje prijaté inštrukcie a nevykoná merania. Po uplynutí tejto doby sú do senzora poslané riadiace príkazy, ktoré zabezpečia posielanie nameranej vzdialenosti v požadovaných jednotkách. Po prijatí nameraných údajov sú tieto spracované funkciami pre porovnanie a vyhodnotenie a následne sú informácie interpretované v podobe diódového poľa. Diódové pole je znázornené na *obrázku 29*. Režim sa ukončuje kliknutím na tlačidlo ukončenia režimu, kedy sa komunikácia ukončí a zobrazí sa režim hlavného okna.

Funkcia void dual\_Click(object sender, EventArgs e) inicializuje režim, pri ktorom pracujú obe periférne zariadenia spoločne v aplikácii pre stolový počítač. V režime sa využíva len jedna parkovacia kamera na rozdiel od režimu kamery. Režim však poskytuje informácie o vzdialenosti vozidla od prekážky. Na senzor sa vyšlú riadiace



príkazy, čím sa začne komunikácia a na diódovom paneli sa začne zobrazovať stav vzdialenosti. Vytvorí sa nové vlákno, ktoré na seba preberie funkcie spracovania videa. Získa sa zdroj videa, na ktorý sa nastaví video filter. Následne sa začnú vytvárať video obrázky, ktoré sú zobrazované na zobrazovacom poli. Ukončenie režimu sa vykoná funkciou pre ukončenie režimu, kedy sa ukončí komunikácia senzora so serverom a ukončí sa prenos videa a zobrazia sa ovládacie prvky hlavného okna.

Táto funkcia v aplikácii pre server predstavuje a spúšťa režim súhra. Poskytuje podporu pre pripojenie a spracovanie údajov zo štyroch senzorov súčasne. Pri inicializácii režimu sa spustí komunikácia senzora so serverom. V dôsledku, že je dostupný len jeden senzor, ostatné tri sú simulované funkciami, ktoré generujú požadované sekvencie čísel. Následne po získaní dát zo všetkých senzorov sa tieto vyhodnotia a zobrazia sa individuálne informácie z každého jedného. Na diódovom paneli sa zobrazí najkratšia vzdialenosť. Ukončenie režimu sa realizuje funkciou ukončenia.

Funkcia `void mute_on_Click(object sender, EventArgs e)` slúži na zapnutie zvukov, ktoré sú aktivované v režimoch, ktoré pracujú s ultrazvukovými senzormi.

Funkcia `void mute_off_Click(object sender, EventArgs e)` slúži na vypnutie zvukov.

Funkcia `void config_Click(object sender, EventArgs e)` slúži na konfiguráciu senzorov. Inicializácia funkcie zobrazí ikony zobrazujúce stav senzorov. Režim poskytuje funkcie, ktoré umožnia vypnúť alebo zapnúť používanie jednotlivých senzorov. Rovnako poskytuje možnosti zmeny meraných rozsahov jednotlivých senzorov. Zmeny rozsahov sa vykonávajú na základe hodnôt *tabuľky 4* v *podkapitole 2.4.3.4 Režim konfigurácie senzorov*. Režim sa ukončuje inicializáciou funkcie pre ukončenie režimu.

Funkcia `void setupCommPort(String comport)` sa inicializuje pri spustení systému. Zabezpečuje vytvorenie objektu typu COM port, vďaka ktorému je možné nastaviť základné parametre komunikácie s ultrazvukovým senzorom. Prenosová rýchlosť je nastavená na 19200 znakov za sekundu. Paritný bit nie je nastavený, pretože nevyžaduje spätnú kontrolu. Kontrola je vykonávaná samotným senzorom, ktorý ak nemá pripravené dáta, posiela hodnotu 0xE1. Funkcia následne nastavuje dva stop bity, pre zabezpečenie oddelenia dátovej časti od časti režie. Po nastavení všetkých parametrov sa otvorí port a je pripravený na komunikáciu.

Funkcia `void stopky_btn_Click(object sender, EventArgs e)` slúži na inicializáciu parkovacích stopiek, ktoré predstavujú doplnkovú funkcionálnosť. Merajú dobu

za akú dokáže vodič zaparkovať vozidlo. Stopky pracujú na vlastnom vlákne nezávisle od hlavnej časti programu.

Funkcia `void exit_Click(object sender, EventArgs e)` slúži na ukončenie aplikácie. Funkcia zatvára otvorený COM port.

#### 2.5.4 Funkcie režimu kamery

`void start_cam_cam()` - funkcia slúži na detekciu kamier v systéme a zabezpečuje nastavenie zdroja videa. Na tento zdroj sa nastaví video filter a začnú sa vytvárať obrázky, ktoré sú zobrazované do obrazového poľa.

`void CloseVideoSource()` - funkcia slúži na ukončenie zobrazovania obrazov do obrazového poľa. Odstráni video filter zo zdroja videa a korektne ukončí adaptér.

`void cam1_on_Click(object sender, EventArgs e)` - funkcia slúži na vyvolanie funkcie zapnutia kamery 1.

`void cam2_on_Click(object sender, EventArgs e)` - funkcia slúži na vyvolanie funkcie zapnutia kamery 2.

`void back_cam_Click(object sender, EventArgs e)` - funkcia slúži na ukončenie režimu parkovacej kamery. Vypína ovládacie prvky režimu a aktivuje ovládacie prvky hlavného okna.

#### 2.5.5 Funkcie režimu ultrazvukového senzora

`void snimac_Tick(object sender, EventArgs e)` - funkcia spúšťa časovač, ktorý zabezpečuje neustále sa opakujúcu komunikáciu ultrazvukového senzora so serverom a obsahuje funkcie, ktoré zabezpečujú vyhodnotenie a vykreslenie údajov získaných zo senzora. Časovač sa vykonáva každých 200ms, čo zabezpečí včasnú a dostatočnú aktualizáciu údajov.

`double ReadValue_senzor()` - funkcia spracuje dátový dok prijatý zo senzora a spracuje ich do potrebnej formy. Funkcia vracia nameranú vzdialenosť v centimetroch.

`void vyhodnot_senzor(double x, double set)` - prvým parametrom funkcie je nameraná vzdialenosť v centimetroch, druhým parametrom je základ, ktorý slúži pre konfiguráciu rozsahu senzora. Funkcia definuje desať základných zón, ktoré sú definované a presne popísané v kapitole 2.4.2.3 *Režim ultrazvukových senzorov*. V závislosti od parametra *x* sa rozsvetia príslušné diódy. Rovnako sa na základe parametra *x* spustí hlasová signalizácia, pokiaľ je povolená v nastaveniach. V prípade, že je nastavený

parameter *set* v rozsahu <10, 50>, zmena základných rozsahov sa vykoná na základe tabuľky 4 popísanej v podkapitole 2.4.3.4 Režim konfigurácie senzorov.

senzor vrat\_min(double x1, double x2, double x3, double x4) – hlavnou úlohou tejto funkcie je porovnanie údajov získaných z jednotlivých senzorov (generátorov) a vyhodnotenie najkratšej vzdialenosti. Funkcia je typu *senzor*, predstavuje vlastnú triedu, ktorá definuje číslo senzora a meranú vzdialenosť. Funkcia vracia najmenšiu vzdialenosť zo všetkých nameraných vzdialeností a senzor, ktorý ju namerá. Táto funkcia sa využíva v prípade ak pracujeme so štyrmi senzormi.

double generator(int min, int max) – funkcia generátora nahradzuje reálne údaje, ktoré by boli získané zo senzorov. Využíva sa pre simuláciu reálnej činnosti troch chýbajúcich senzorov. Parametre funkcie predstavujú uzavretý interval, v ktorom sa generujú výstupné údaje. Generátor sa využíva rovnako len pri režime použitia štyroch senzorov, v režime súhra.

void back\_sens\_Click(object sender, EventArgs e) – funkcia ukončuje spojenie s ultrazvukovým senzorom, vypína zobrazovacie prvky režimu. Zobrazuje a sprístupňuje funkcie hlavného okna.

## 2.5.6 Trieda senzor

```
public class senzor
{
    public double min;
    public double ktory;
}
```

Pri spracovaní údajov zo senzorov bolo nutné definovať najkratšiu vzdialenosť a presne definovať, ktorý senzor ju namerá. Z toho dôvodu bolo nutné vytvoriť triedu, ktorá obsahuje premenné týchto parametrov. K parametrom sa prístupuje cez objekty príslušnej triedy. Triedu *senzor* využíva funkcia *vrat\_min()*.

## 2.5.7 Funkcie duálneho režimu

- void start\_cam\_dual()
- void CloseVideoSource()
- void dual\_Tick(object sender, EventArgs e)
- double ReadValue\_senzor()
- void vyhodnot\_senzor(double x, double set)
- void back\_dual\_Click(object sender, EventArgs e)

Keďže duálny režim je kombináciou dvoch režimov, ktorých funkcie boli popísané v podkapitolách 2.5.4 *Funkcie režimu kamery* a 2.5.5 *Funkcie režimu ultrazvukových*

*senzorov*, v tejto kapitole sú funkcie len vymenované. Funkcie sa líšia len v názvoch, kde definujú príslušný režim. Funkcie pracujúce s kamerou sa spúšťajú na vlastnom vlákne, keďže sa jedná o paralelné vykonávanie operácií snímania videa a merania vzdialenosti.

### 2.5.8 Funkcie režimu súhra

- void dual\_Tick(object sender, EventArgs e)
- double ReadValue\_senzor()
- void vyhodnot\_dual(double x, double set)
- senzor vrat\_min(double x1, double x2, double x3, double x4)
- double generator(int min, int max)
- void vyhodnot\_s1(double x, double set)
- void vyhodnot\_s2(double x, double set)
- void vyhodnot\_s3(double x, double set)
- void vyhodnot\_s4(double x, double set)
- void back\_dual\_Click(object sender, EventArgs e)

Režim využíva podobné funkcie ako režim merania najkratšej vzdialenosti. Jedná sa o funkcie, ktoré zabezpečujú komunikáciu so senzorom. Vyhodnocujú najkratšie vzdialenosti za využitia triedy *senzor*. Rozdielom oproti režimu ultrazvukových senzorov je spracovanie informácií každého senzoru osobitne. V dôsledku toho boli zaradené do vyhodnotenia funkcie pre spracovanie údajov pre každý senzor osobitne.

## 2.6 Overenie riešenia

Overenie funkčnosti riešenia predstavuje jeden z najdôležitejších krokov počas samotnej implementácie ako aj na jej konci. Samotné testovanie aplikácie prebiehalo v dvoch fázach. Prvá fáza predstavovala testovanie počas implementácie jednotlivých funkcií. Počas samotného testovania sa postupne otestovali funkcie hlavného okna, funkcie parkovacích senzorov, režimu kamery, ako aj duálneho režimu či režimu konfigurácie.

System bol testovaný len v laboratórnych podmienkach, pretože testovanie na reálnom automobile by predstavovalo veľký zásah do kabeláže a napájania automobilu, čo by mohlo spôsobiť ako poškodenie zariadenia, rovnako tak poškodenie automobilu, z toho dôvodu som ostal pri testovaní v laboratórnych podmienkach. Počas testovania funkcií režimu kamery na servery sa vyskytol závažný problém. Problém predstavoval nefunkčnú komunikáciu kamery so serverom. Problém bol zapríčinený nevhodnými ovládačmi, ktoré neumožnili komunikáciu inicializovať. Problém bol vyriešený implementáciou funkcionality kamery na stolový počítač, z čoho vzišli dve nezávislé aplikácie. Na

stolovom počítači bola implementovaná funkcionálna kamera, ktorá je podrobne popísaná v kapitole 2.5.4 *Funkcie režimu kamery*, funkcionálna bola následne otestovaná.

Testovanie ultrazvukového senzora spočívalo v kontrole spojenia a správnosti komunikácie senzora so serverom a následne v presnosti merania. Samotná aktivita senzora bola otestovaná vizuálne, kedy na senzory blikajú diódy, tak aj meraniami. Presnosť merania bola testovaná na prípravku, ktorý bol vytvorený z papiera a metra. Porovnávala sa vzdialenosť, ktorá bola nameraná prípravkom so vzdialenosťou nameranou senzorom. Test potvrdil presnosť merania senzora.

Testovanie duálneho režimu spočívalo v kontrole, či dokáže kamera pracovať nezávisle od senzora v spoločnom režime. Otestované boli obe vlákna, ktoré zabezpečujú funkcionálnosť či už senzora alebo kamery.

Systém poskytuje podporu pre prácu so štyrmi senzormi. Reálne je však dostupný iba jeden senzor. Zostávajúce tri senzory získavajú dáta z generátorov, ktoré poskytujú požadované hodnoty. Generátory prešli radou nastavení a testov, ktoré zabezpečili generovanie požadovaných údajov. Následne boli integrované do systému. Po integrácii generátorov do systému sa vyskytol problém, ktorý nepodporoval prácu štyroch senzorov v jednom režime. Problém bol vyriešený vytvorením samostatného vlákna pre každý senzor (generátor).

Ďalším krokom bolo testovanie konfigurácie. Konfigurácia predstavuje rozsiahly zásah do merania a vyhodnotenia senzorov. Testovanie muselo byť podrobné a precízne. Prvou testovanou funkciou bola funkcia zabezpečujúca vypnutie alebo zapnutie senzorov. Test spočíval v kontrole nameraných dát, ktorých hodnota v odpojenom stave mala byť nulová. Kontrola preukázala správnosť funkcie odpojenia. Následne bolo vykonané testovanie funkcie pripojenia senzora. Kontrola nameraných dát preukázala správnosť funkcie. Výstupom merania bola nameraná vzdialenosť vozidla od prekážky a nie nulová hodnota. Náročnejšie testovanie spočívalo v kontrole zmeny meracieho rozsahu vplyvom nastavení rozsahov. Počas testovania bolo vykonané meranie pre každý zo senzorov a pre každý z desiatich rozsahov. Pri overení funkčnosti bolo vykonaných 200 meraní, ktoré preukázali správnosť zmeny meracích rozsahov, ktoré sú definované *tabuľkou 4* v kapitole 2.4.3.4 *Režim konfigurácie senzorov*.

Testovanie parkovacích stopiek predstavovalo porovnávanie nameraných hodnôt so stopkami obsiahnutými v mobilnom telefóne. Meranie preukázalo totožné výsledky. Je možné tvrdiť, že stopky merajú správne.

Posledným krokom čiastkového testovania bolo overenie kontroly, ktorá zabezpečuje detekciu nepripojeného senzora. Funkcia bola testovaná s pripojeným senzorom, kedy funkcia nemala vykazovať zmeny oproti bezporuchovej činnosti. Následne sa test opakoval s tým, že senzor nebol pripojený a systém dokázal túto zmenu zistiť vykonať príslušné opatrenia, ako napríklad vypnutie možnosti konfigurácie senzoru.

Čiastočné merania preukázali, že systém má predpoklad pracovať ako celok. Ďalším krokom bolo však dokázať toto tvrdenie. Po ukončení implementácie všetkých funkcií a vytvorení celkovej podoby systému, sa systém začal testovať ako celok. Na všetky funkcie boli aplikované rovnaké testovacie postupy a výsledkom testovania celého systému je funkčnosť všetkých častí a splnenie požiadaviek vyplývajúcich zo zadania diplomovej práce.

### 3 Záver

Na základe podrobnej analýzy vychádzajúcej z práce Diplomový projekt 1[25], ktorá analyzuje možnosti vnoreného systému spoločnosti Funtoro[9], boli použité zariadenia server BV-105 a MOD monitor. Výstupom podrobnej analýzy ultrazvukových senzorov, v kapitole 1.4, je výber a použitie ultrazvukového senzora SRF02. Pre zobrazovanie videa je použitá kamera LOGITECH PRO 9000 v dôsledku podpory knižnice USB VIDEO CLASS. Podrobné zdôvodnenie výberu jednotlivých hardvérových komponentov popisuje kapitola 1.6 *Záver analýzy*.

Počas testovania kamery sa vyskytol problém nefunkčnej komunikácie kamery a servera. Problém je spôsobený nevhodnými ovládačmi kamery pre operačný systém Windows CE. Problém sa nepodarilo odstrániť. Podrobný popis problému je uvedený v kapitole 2.5.1 *Implementácia funkcií kamery*. Náhradné riešenie spočívalo v implementácii funkcionality kamery na stolový počítač. Dôsledkom implementácie funkcionality kamery na stolový počítač bola nutnosť návrhu a implementácie dvoch samostatných riešení. Prvé riešenie pozostáva z aplikácie určenej pre server BV-105 a druhé riešenie pozostáva z aplikácie určenej pre stolový počítač.

Na základe podrobného návrhu, vychádzajúceho z práce Diplomový projekt 2[26], boli navrhnuté a implementované dve nezávislé aplikácie, ktoré poskytujú prácu vo viacerých nezávislých režimoch. Navrhnuté a implementované režimy zahŕňajú všetku požadovanú funkcionality. Režimy poskytujú prácu s parkovacou kamerou, ultrazvukovým senzorom, kombinovaný režim a režim konfigurácie. Architektúra systému a podrobný návrh je popísaný v kapitole 2.4 *Návrh architektúry systému*. Podrobný popis implementácie jednotlivých režimov je popísaný v kapitole 2.5 *Implementácia systému*.

Aplikácie prešli radou testov, ktoré preukázali správnosť jednotlivých funkcií. Testovanie je podrobne popísané v kapitole 2.6 *Overenie riešenia*. Testy prebiehali iba v laboratórnych podmienkach. Testovanie v reálnych podmienkach by si vyžadovalo doplnkové príslušenstvo v podobe automobilových káblov potrebných pre napájanie servera. Jednalo by sa o napájacie káble k autobatérii, ktoré by na jednej strane obsahovali svorky na pripojenie k autobatérii a na opačnej strane svorky na pripojenie servera. Montáž by si rovnako vyžadovala inštaláciu parkovacieho senzora do karosérie automobilu. Inštalácia by si vyžadovala špecializované nástroje a zásah do karosérie by spôsobil nenávratné zmeny. Na základe týchto dôvodov som sa rozhodol pre testovanie v laboratórnych podmienkach.

Obsahom práce je aj technická dokumentácia, ktorá pozostáva z používateľskej príručky a z doplnku k implementácii.

### ***3.1 Možnosti vylepšenia systému***

- Implementácia funkcií kamery na server BV-105
- Automatická inicializácia systému po zaradení spiatočky
- Rozšírenie o predné parkovacie senzory
- Zobrazenie kontroly stavu senzorov v režime ultrazvukového senzora
- Funkcia zabezpečujúca vyhľadanie pozdĺžneho parkovacieho miesta pomocou ultrazvukových senzorov
- Regulácia a kontrola vozidla, vytvorenie samočinného parkovacieho asistenta



## 4 Technická dokumentácia

Kapitola obsahuje technickú dokumentáciu k implementovanému riešeniu parkovacieho asistenta. Pozostáva z dvoch základných častí. Prvá časť predstavuje dokumentáciu k implementácii a druhá časť predstavuje používateľskú príručku.

### 4.1 Dokumentácia k implementácii

Dokumentácia predstavuje doplnok k časti implementácie. Opisuje najdôležitejšie funkcie spolu s časťami zdrojových kódov a premenných.

#### 4.1.1 Hlavná trieda *public partial class Form1 : Form*

Táto trieda je hlavnou triedou programu, ktorá obsahuje všetky základné funkcie. Zabezpečuje komunikáciu so senzorom, vyhodnocuje údaje získané zo senzoru, pracuje s kamerami.

```
public partial class Form1 : Form
{
    //hlavne premenne programu
    const byte I2C_CMD = 0x55;
    int n;
    byte[] sbuf;
    int readDelay = 10, sound_flag = 0, mute = 1, sinc = 1, ss1 = 1,
    ss2 = 1, ss3 = 1, ss4 = 1, disc = 0, stopky_stav = 0;
    double s1_stat = 1, s2_stat = 1, s3_stat = 1, s4_stat = 1;
    string os = "wince";
    SoundPlayer ok_ton, err_ton;
    private System.IO.Ports.SerialPort port;
    byte device_addr = 0xE0;
    Stopwatch sw = new Stopwatch();
    //funkcia zabezpecujuca inicializaciu hlavneho okna
    public Form1()
    //aktivacia rezimu ultrazvukoveho senzora
    private void senzor_Click(object sender, EventArgs e)
    //nastavenie parametru komunikacie cez COM port
    private void setupCommPort(String comport)
    //citanie hodnoty zo senzora
    private double ReadValue_senzor()
    //spracovanie udajov zo senzora v rezime ultraz. Rezimu
    private void vyhodnot_senzor(double x, double set)
    //vratenie najkratsej vzdialenosti
    senzor vrat_min(double x1, double x2, double x3, double x4)
    //casovac zabezpecujuci pravidelnu komunikaciu so senzorom
    private void snimac_Tick(object sender, EventArgs e)
    //aktivacia dualneho rezimu
    private void dual_Click(object sender, EventArgs e)
    //aktivacia rezimu suhry
    private void suhra_Click(object sender, EventArgs e)
    //spracovanie udajov so senzora v dualnom rezime a rezime suhry
    private void vyhodnot_dual(double x, double set)
    //funkcie zabezpecujuce vykreslenie jednotlivych senzorov v rezime
    suhry
    private void vyhodnot_s1(double x, double set)
```

```

private void vyhodnot_s2(double x, double set)
private void vyhodnot_s3(double x, double set)
private void vyhodnot_s4(double x, double set)
//funkcia generuje vstupne data pre simulaciu prace senzorov
private double generator(int min, int max)
//pocitadlo zabezpecujuce komunikaciu so senzorov v dual. rezime
private void dual_tik_Tick(object sender, EventArgs e)
//funkcia zatvara COM port a ukoncuje aplikaciu
private void exit_Click(object sender, EventArgs e)
//ukoncenie rezimu ultrazvuk. senzora
private void back_sens_Click(object sender, EventArgs e)
//ukoncenie dualneho rezimu
private void back_dual_Click(object sender, EventArgs e)
//funkcia zapnutia zvukov
private void mute_on_Click(object sender, EventArgs e)
//funkcia vypnutia zvukov
private void mute_off_Click(object sender, EventArgs e)
//aktivacia konfiguracneho rezimu
private void config_Click(object sender, EventArgs e)
//funkcie konfiguracie zabezpecujuce manualne spustenie senzorov
private void go_1_Click(object sender, EventArgs e)
private void go_2_Click(object sender, EventArgs e)
private void go_3_Click(object sender, EventArgs e)
private void go_4_Click(object sender, EventArgs e)
//funkcie konfiguracie zabezpecujuce manualne vypnutie senzorov
private void stop_1_Click(object sender, EventArgs e)
private void stop_2_Click(object sender, EventArgs e)
private void stop_3_Click(object sender, EventArgs e)
private void stop_4_Click(object sender, EventArgs e)
//funkcia ukoncuje rezim konfiguracie
private void back_conf_Click(object sender, EventArgs e)
//funkcie konfiguracie menia meracie rozsahy senzorov
private void s1_l_Click(object sender, EventArgs e)
private void s1_p_Click(object sender, EventArgs e)
private void s2_l_Click(object sender, EventArgs e)
private void s2_p_Click(object sender, EventArgs e)
private void s3_l_Click(object sender, EventArgs e)
private void s3_p_Click(object sender, EventArgs e)
private void s4_l_Click(object sender, EventArgs e)
private void s4_p_Click(object sender, EventArgs e)
//funkcia vykonava pocitanie pri aktivovanych stopkach
private void stopky_timer_Tick(object sender, EventArgs e)
//spustenie rezimu suhra
private void stopky_btn_Click(object sender, EventArgs e)
//ukoncenie rezimu suhra
private void stopky_back_Click(object sender, EventArgs e)
//spustenie sekvencie pocitania
private void stopky_start_Click(object sender, EventArgs e)
//ukoncenie sekvencie pocitania
private void stopky_stop_Click(object sender, EventArgs e)
//vynulovanie pocitania
private void stopky_reset_Click(object sender, EventArgs e)
//funkcia spustajuca prvu kameru v rezime kamery
private void start_cam_cam()
//funkcia spustajuca kameru v dualnom rezime
private void start_cam_dual()
//funkcia spustajuca druhu kameru v rezime kamery
private void start_cam_cam_2()
//funkcia vytvara a zobrazuje nove obrazy do obrazoveho pola, rezim kamery
private void Obraz_cam(object sender, NewFrameEventArgs eventArgs)

```

```

//funkcia vytvara a zobrazuje nove obrazy v dualnom rezime
private void Obraz_dual(object sender, NewFrameEventArgs eventArgs)
//ukoncenie zdrojov videa
private void Exit_zdroj()
//funkcia zabezpeci premietanie kamery 1
private void cam1_on_Click(object sender, EventArgs e)
//funkcia zabezpeci vysielanie kamery 2
private void cam2_on_Click(object sender, EventArgs e)
}

```

#### 4.1.2 Funkcia *public Form1()*

Funkcia zabezpečuje inicializáciu hlavného okna a prístupní funkcie, ktoré sa z neho volajú. Rovnako zabezpečuje nastavenie COM portu pre server ako aj pre 32b Windows. Zabezpečuje kontrolu pripojenia senzora v prípade absencie a nastavuje pomocné premenné. Nastavuje základné adresáre pre hľadanie pomocných súborov.

```

public Form1()
{
    InitializeComponent();
    FormBorderStyle = FormBorderStyle.None;
    WindowState = FormWindowState.Maximized;
    TopMost = true;
    hl_vrch.Visible = true;
    senzor.Visible = true;
    dual.Visible = true;
    exit.Visible = true;
    mute_off.Visible = false;
    mute_on.Visible = true;
    config.Visible = true;
    stopky_btn.Visible = true;
    if (os == "wince")
    {
        Cursor.Hide();
    }

    if (os == "win32")
    {
        ok_ton = new SoundPlayer(@"c:\beep_ok.wav");
        err_ton = new SoundPlayer(@"c:\beep_err.wav");
    }
    else if (os == "wince")
    {
        ok_ton = new SoundPlayer("Storage Card\\BEEP_OK.wav");
        err_ton = new SoundPlayer("Storage Card\\BEEP_ERR.wav");
    }

    if (os == "wince")
    {
        try
        {
            setupCommPort("$device/COM10");
        }
        catch (Exception e)
        {
            disc = 1;
        }
    }
    else if (os == "win32")

```

```

    {
        try
        {
            setupCommPort("COM3");
        }
        catch (Exception e)
        {
            disc = 1;
        }
    }
}

```

#### 4.1.3 Funkcia *private void senzor\_Click(object sender, EventArgs e)*

Funkcia sa aktivuje kliknutím na ikonu, ktorá spúšťa režim ultrazvukového režimu. Zabezpečuje zobrazenie prvkov režimu ako diódové pole, meracie vlny, displej s nameranou hodnotou. Nastavuje základné premenné pre komunikáciu so senzorom a spúšťa počítadlo, ktoré zabezpečuje pravidelnú komunikáciu so senzorom.

```

private void senzor_Click(object sender, EventArgs e)
{
    n = 0;
    sbuf = new byte[100];
    snimac.Enabled = true;
    hore_senzor.Visible = true;
    back_sens.Visible = true;
    car.Visible = true;
    warn1.Visible = true;
    warn2.Visible = true;
    warn1.Visible = true;
    warn2.Visible = true;
    zl1.Visible = true;
    zl2.Visible = true;
    zl3.Visible = true;
    zl4.Visible = true;
    zp1.Visible = true;
    zp2.Visible = true;
    zp3.Visible = true;
    zp4.Visible = true;
    ol1.Visible = true;
    ol2.Visible = true;
    ol3.Visible = true;
    op1.Visible = true;
    op2.Visible = true;
    op3.Visible = true;
    rl1.Visible = true;
    rl2.Visible = true;
    rl3.Visible = true;
    rp1.Visible = true;
    rp2.Visible = true;
    rp3.Visible = true;
    sede.Visible = true;
    hodnota.Visible = true;
    disp.Visible = true;
    hodnota.ForeColor = Color.LightGreen;
}

```

#### 4.1.4 Funkcia *private void setupCommPort(String comport)*

Funkcia zabezpečuje nastavenie základných parametrov sériovej komunikácie prostredníctvom COM portu. Jedná sa o nastavenie prenosovej rýchlosti, paritných bitov, stop bitov. Funkcia rovnako obsahuje kontrolu, ktorá predíde chybnému pokusu o otvorenie spojenia v prípade absencie senzora.

```
private void setupCommPort(String comport)
{
    port = new SerialPort();
    this.port.BaudRate = 19200;
    this.port.Parity = 0;
    this.port.StopBits = StopBits.Two;
    this.port.Handshake = Handshake.None;
    this.port.RtsEnable = false;
    this.port.ReadTimeout = 500;
    this.port.WriteTimeout = 500;
    this.port.StopBits = StopBits.Two;

    this.port.DataBits = 8;
    port.PortName = comport;
    if (disc == 0) { port.Open(); }
}
```

#### 4.1.5 Funkcia *private void vyhodnot\_senzor(double x, double set)*

Funkcia vykonáva hlavnú úlohu pri vyhodnocovaní prijatých údajov zo senzorov. Ako parametre funkcie vstupujú nameraná vzdialenosť v centimetroch a základ pre zmenu nastavení rozsahov. Nameraná vzdialenosť je následne porovnávaná radom rozhodovaní a na základe svojej hodnoty je na diódovom paneli rozsvietené príslušné množstvo diód príslušnej farby. Rovnako tak sa na základe zóny nastavuje aj zvuková signalizácia, v prípade, že je povolená. Funkcia zabezpečuje aj zobrazenie vzdialenosti na displeji v pravom hornom rohu a poskytuje presnú číselnú informáciu o vzdialenosti. V prípade potreby funkcia aktivuje kritickú signalizáciu kedy aktivuje výstražné hlásenia. Základné rozsahy zón, v ktorých sa nameraná hodnota porovnáva je možné zmeniť. Ku každému rozsahu sa pripočíta požadovaná hodnota v pomere definovanom *tabuľkou 4* v kapitole 2.4.6. Výstupom funkcie je teda obraz poskytujúci vizuálny prehľad o vzdialenosti vozidla od prekážky.

```
private void vyhodnot_senzor(double x, double set)
{
    sinc++;
    double cm = x;
    if (cm >= (50+(set/3)))
    {
        if (g.Visible == false)
            g.Visible = true;
        else g.Visible = false;
    }
}
```

```

o.Visible = false;
r.Visible = false;
warn1.Visible = false;
warn2.Visible = false;
hodnota.ForeColor = Color.LightGreen;

if (cm >= (80+(set*1.2)))
{
    z11.Visible = true;
    z12.Visible = false;
    ...
    rp1.Visible = false;
    rp2.Visible = false;
    rp3.Visible = false;
}
else if ((cm < (80+(set*1.2))) && (cm >= (70+set)))
{
    z11.Visible = true;
    z12.Visible = true;
    ...
    rp1.Visible = false;
    rp2.Visible = false;
    rp3.Visible = false;
}
else if ((cm < (70+set)) && (cm >= (60+(set/2))))
{
    z11.Visible = true;
    z12.Visible = true;
    ...
    rp1.Visible = false;
    rp2.Visible = false;
    rp3.Visible = false;
}
else if ((cm < (60+(set/2))) && (cm >= (50+(set/3))))
{
    z11.Visible = true;
    z12.Visible = true;
    ...
    rp1.Visible = false;
    rp2.Visible = false;
    rp3.Visible = false;
}
    sound_flag = 1;
}
else if ((cm < (50+(set/3))) && (cm >= (30+(set/6))))
{
    if (o.Visible == false)
        o.Visible = true;
    else o.Visible = false;

    g.Visible = false;
    r.Visible = false;
    warn1.Visible = false;
    warn2.Visible = false;
    hodnota.ForeColor = Color.Orange;

    if ((cm < (50+(set/3))) && (cm >= (44+(set/4))))
    {
        z11.Visible = true;
        z12.Visible = true;
        ...
    }
}

```

```

        rp1.Visible = false;
        rp2.Visible = false;
        rp3.Visible = false;
    }
    else if ((cm < (44+(set/4))) && (cm >= (38+(set/5))))
    {
        z11.Visible = true;
        z12.Visible = true;
        ...
        rp1.Visible = false;
        rp2.Visible = false;
        rp3.Visible = false;
    }
    else if ((cm < (38+(set/5))) && (cm >= (30+(set/6))))
    {
        z11.Visible = true;
        z12.Visible = true;
        ...
        rp1.Visible = false;
        rp2.Visible = false;
        rp3.Visible = false;
    }
    sound_flag = 2;
}
else if (cm < (30 + (set/6)))
{
    if (r.Visible == false)
        r.Visible = true;
    else r.Visible = false;

    g.Visible = false;
    o.Visible = false;
    hodnota.ForeColor = Color.Red;

    if ((cm < (30 + (set/6))) && (cm >= (25 + (set/7))))
    {
        z11.Visible = true;
        z12.Visible = true;
        ...
        rp3.Visible = false;
        warn1.Visible = false;
        warn2.Visible = false;
    }
    else if ((cm < (25 + (set/7))) && (cm >= (20 + (set/8))))
    {
        z11.Visible = true;
        z12.Visible = true;
        ...
        rp3.Visible = false;
        warn1.Visible = false;
        warn2.Visible = false;
    }
    else if (cm < (20 + (set/8)))
    {
        z11.Visible = true;
        z12.Visible = true;
        ...
        rp2.Visible = true;
        rp3.Visible = true;
    }
}

```

```

        if ((warn1.Visible == false) && ((sinc % 2) == 0))
        {
            warn1.Visible = true;
            warn2.Visible = false;
        }
        else if ((warn1.Visible == true) && ((sinc % 2) ==
0))
        {
            warn1.Visible = false;
            warn2.Visible = true;
        }
    }
    sound_flag = 3;
}
hodnota.Text = cm.ToString();

if ((sound_flag == 1) && (mute == 1) && ((sinc % 3) == 0))
{
    err_ton.Stop();
    ok_ton.Stop();
    ok_ton.Play();
}
else if ((sound_flag == 2) && (mute == 1) && ((sinc % 2) ==
0))
{
    err_ton.Stop();
    ok_ton.Stop();
    ok_ton.Play();
}
else if ((sound_flag == 3) && (mute == 1))
{
    err_ton.Stop();
    ok_ton.Stop();
    err_ton.Play();
}
}
}

```

#### 4.1.6 Funkcia *senzor vrat\_min(double x1, double x2, double x3, double x4)*

Funkcia slúži na spracovanie nameraných hodnôt senzorov a následné vypočítanie najmenšej hodnoty, teda najkratšej vzdialenosti. Funkcia je typu *senzor*, ktorá umožní vrátiť informácie o senzoroch poľom. Vstup funkcie sú štyri namerané vzdialenosti a výstupom je pole, ktoré poskytuje najkratšiu vzdialenosť a senzor, ktorý ju namerá. Funkcia je založená na vzájomnom porovnaní.

```

senzor vrat_min(double x1, double x2, double x3, double x4)
{
    senzor akt = new senzor();
    akt.min = 1000;
    akt.ktory = 1;

    if (x1 >= x2)
    {
        akt.min = x2;
        akt.ktory = 2;
    }
}

```



```

else
{
    akt.min = x1;
    akt.ktory = 1;
}

if (akt.min >= x3)
{
    akt.min = x3;
    akt.ktory = 3;
}

if (akt.min >= x4)
{
    akt.min = x4;
    akt.ktory = 4;
}

return akt;
}

```

#### 4.1.7 Funkcia *private void snimac\_Tick(object sender, EventArgs e)*

Funkcia zabezpečuje pravidelné posielanie riadiacich príkazov na senzor a následne číta namerané údaje. Namerané údaje sú priradené senzoru 3. Sensory 1, 2, 4 získavajú údaje z generátora náhodných čísel v požadovaných intervaloch. Následne sa volá funkcia pre vrátenie najkratšej vzdialenosti a nastaví sa hodnota pre zmenu meracieho rozsahu. Nakoniec sa vykoná vysvietenie príslušných diód v nastavenom rozsahu. Cyklovanie sa vykonáva každých 200ms. Za 1s je zabezpečených päť zberov a vyhodnotení dát, čo je dostačujúce.

```

private void snimac_Tick(object sender, EventArgs e)
{
    sbuf[0] = I2C_CMD;
    sbuf[1] = 0xE1;
    sbuf[2] = 0x00;
    sbuf[3] = 0x04;

    senzor vysledok = new senzor();
    double res = 0;
    if (disc == 0) { port.Write(sbuf, 0, 4); }
    Thread.Sleep(readDelay);

    if (disc == 0) { port.Read(sbuf, 0, 4); }
    double senzor_1, senzor_2, senzor_3, senzor_4;

    if (ss1 == 1) { senzor_1 = generator((int)generator(40, 79), 150); }
    else { senzor_1 = 1000; }
    if (ss2 == 1) { senzor_2 = generator(33, 150); } else { senzor_2 = 1000; }
    if ((ss3 == 1) && (disc == 0)) { senzor_3 = ReadValue_senzor(); }
    else { senzor_3 = 1000; }
    if (ss4 == 1) { senzor_4 = generator(40, (int)generator(41, 150)); }
    else { senzor_4 = 1000; }

    vysledok = vrat_min(senzor_1, senzor_2, senzor_3, senzor_4);
}

```

```

    if (vysledok.ktory == 1)
    { res = s1_stat; }
    else if (vysledok.ktory == 2)
    { res = s2_stat; }
    else if (vysledok.ktory == 3)
    { res = s3_stat; }
    else if (vysledok.ktory == 4)
    { res = s4_stat; }

    vyhodnot_senzor((double)vysledok.min, res);
}

```

#### 4.1.8 Funkcia *private double generator(int min, int max)*

Funkcia zabezpečuje generovanie dát pre simulovanie činnosti troch senzorov, ktoré fyzicky nie sú dostupné. Parametrami funkcie je minimálna a maximálna hranica, v ktorej sa bude požadovaná hodnota generovať. Funkcia vracia simulované namerané dáta.

```

private double generator(int min, int max)
{
    Random random = new Random();
    return random.Next(min, max);
}

```

#### 4.1.9 Funkcia *private void back\_sens\_Click(object sender, EventArgs e)*

Funkcia slúži na ukončenie režimu ultrazvukového senzora. Po zavolaní funkcie sa vypne zobrazenie režimu, nastaví sa hlavné okno a sprístupnia sa funkcie hlavného okna. Ukončí sa počítadlo čím sa ukončí komunikácia so senzorom. Funkcia rovnako poskytuje kontrolu nastavení zvukov, kedy je schopná pamätať si predošlé nastavenie zvukov a zachovať toto nastavenie a prispôbiť nastavenie funkcie zvukov.

```

private void back_sens_Click(object sender, EventArgs e)
{
    //Hlavný obraz
    hl_vrch.Visible = true;
    exit.Visible = true;
    config.Visible = true;
    stopky_btn.Visible = true;
    if (mute == 1)
    {
        mute_on.Visible = true;
        mute_off.Visible = false;
        ok_ton.Stop();
        err_ton.Stop();
    }
    if (mute == 0)
    {
        mute_off.Visible = true;
        mute_on.Visible = false;
    }

    sound_flag = 0;
    snimac.Enabled = false;
}

```

#### 4.1.10 Funkcia *private void suhra\_Click(object sender, EventArgs e)*

Funkcia zobrazuje prvky režimu súhry a sprístupňuje funkcie režimu. Hlavnou úlohou je spustenie počítadla, ktoré ma rovnakú funkcionality ako počítadlo pre režim ultrazvukového režimu, avšak vyhodnocuje vzdialenosti každého senzora osobitne. Funkcia zabezpečuje zobrazenie stavu jednotlivých senzorov, kedy zobrazuje či sú pripojené, aktívne alebo neaktívne.

```
private void dual_Click(object sender, EventArgs e)
{
    if (ss1 == 1)
    {
        on_1.Visible = true;
        off_1.Visible = false;
    }
    else
    {
        on_1.Visible = false;
        off_1.Visible = true;
    }
    if (ss2 == 1)
    {
        on_2.Visible = true;
        off_2.Visible = false;
    }
    else
    {
        on_2.Visible = false;
        off_2.Visible = true;
    }
    if ((ss3 == 1) && (disc == 0))
    {
        on_3.Visible = true;
        off_3.Visible = false;
        dis_3.Visible = false;
    }
    else if ((ss3 == 0) && (disc == 0))
    {
        on_3.Visible = false;
        off_3.Visible = true;
        dis_3.Visible = false;
    }
    else
    {
        on_3.Visible = false;
        off_3.Visible = false;
        dis_3.Visible = true;
    }
    if (ss4 == 1)
    {
        on_4.Visible = true;
        off_4.Visible = false;
    }
    else
    {
        on_4.Visible = false;
        off_4.Visible = true;
    }
}
```

```

        n = 0;
        sbuf = new byte[100];
        dual_tik.Enabled = true;
    }

```

#### 4.1.11 Funkcia *private void camera\_Click(object sender, EventArgs e)*

Zavolanie funkcie ukončí zobrazenie hlavného okna a zobrazí okno režimu kamery. Zobrazí sa obrazové pole a ikony pre prepínanie kamier. Funkcia vyhodnotí, ktorá z kamier bola naposledy zapnutá a na základe tohto poznatku aktivuje príslušnú kameru a spustí premietanie videa. Funkcia poskytuje možnosť prepnutia sa na prehrávanie druhej kamery.

```

private void camera_Click(object sender, EventArgs e)
{
    back_cam.Visible = true;
    hore_cam.Visible = true;
    cam_view.Visible = true;
    if (scam == 1)
    {
        cam1_on.Visible = false;
        cam1_off.Visible = true;
        cam2_off.Visible = false;
        cam2_on.Visible = true;
        start_cam_cam();
    }
    else if (scam == 2)
    {
        cam1_on.Visible = true;
        cam1_off.Visible = false;
        cam2_off.Visible = true;
        cam2_on.Visible = false;
        start_cam_cam_2();
    }
}

```

#### 4.1.12 Funkcia *private void dual\_Click(object sender, EventArgs e)*

Funkcia zobrazuje prvky diódového pola a zobrazuje zobrazovacie pole pre prehrávanie videa. Aktivuje počítadlo, ktoré zabezpečí komunikáciu so senzorm a spracuje dáta pre desať diódový zobrazovací panel. Aktivuje sa funkcia pre prehrávanie videa do zobrazovacieho pola.

```

private void dual_Click(object sender, EventArgs e)
{
    back_dual.Visible = true;
    hore_dual.Visible = true;
    dioda_g1.Visible = true;
    dioda_g2.Visible = true;
    dioda_g3.Visible = true;
    dioda_g4.Visible = true;
    dioda_o1.Visible = true;
    dioda_o2.Visible = true;
    dioda_o3.Visible = true;
}

```

```

        dioda_r1.Visible = true;
        dioda_r2.Visible = true;
        dioda_r3.Visible = true;
        dioda_sede.Visible = true;
        cam_dual.Visible = true;
        start_cam_dual();
        n = 0;
        sbuf = new byte[100];
        tik_dual.Enabled = true;
    }

```

#### 4.1.13 Funkcie *private void mute\_on\_Click(object sender, EventArgs e)* a *private void mute\_off\_Click(object sender, EventArgs e)*

Funkcie zabezpečujú základné nastavenie zvukov, kedy sa jedná o zapnutie alebo vypnutie hlasitosti.

```

private void mute_on_Click(object sender, EventArgs e)
{
    mute = 0;
    mute_on.Visible = false;
    mute_off.Visible = true;
}
private void mute_off_Click(object sender, EventArgs e)
{
    mute = 1;
    mute_off.Visible = false;
    mute_on.Visible = true;
}

```

#### 4.1.14 Funkcia *public void start\_cam\_cam()*

Funkcia slúži na detekciu pripojených kamier a na následné premietanie videa do pripraveného zobrazovacieho poľa. V prvom rade funkcia vytvorí objekt, ku ktorému priradí zdroje videa, ktoré zistila. Ak je aspoň jeden zdroj, prejde sa kontrolou a práve na tento zdroj sa nastaví video filter. Následne sa začnú spracovávať obrázky, ktoré sú premietané do zobrazovacieho poľa.

```

public void start_cam_cam()
{
    try
    {
        zar = new
        FilterInfoCollection(FilterCategory.VideoInputDevice);

        if (zar.Count == 0)
            throw new ApplicationException();

        je = true;

        if (je)
        {
            zdroj = new VideoCaptureDevice(zar[0].MonikerString);
            zdroj.NewFrame += new
            NewFrameEventHandler(Obraz_cam);
        }
    }
}

```

```

        Exit_zdroj();
        zdroj.DesiredFrameSize = new Size(632, 313);
        zdroj.Start();
    }
}
catch (ApplicationException)
{
    je = false;
    MessageBox.Show("Kamera nenajdena");
}
}

```

#### 4.1.15 Funkcia *public void Obraz\_cam(object sender, NewFrameEventArgs eventArgs)*

Funkcia zabezpečuje sťahovanie rámcov z kamery a zobrazovanie ich v zobrazovacom poli režimu.

```

public void Obraz_cam(object sender, NewFrameEventArgs eventArgs)
{
    Bitmap img = (Bitmap)eventArgs.Frame.Clone();
    cam_view.Image = img;
}

```

#### 4.1.16 Funkcia *public void Exit\_zdroj()*

Funkcia zabezpečuje korektné ukončenie spojenia s kamerou a ukončenie vytvárania rámcov do zobrazovacieho poľa.

```

public void Exit_zdroj()
{
    if (!(zdroj == null))
        if (zdroj.IsRunning)
        {
            zdroj.SignalToStop();
            zdroj = null;
        }
}

```

#### 4.1.17 Funkcia *private void stopky\_timer\_Tick(object sender, EventArgs e)*

Funkcia zabezpečuje počítanie v stopiek a následné zobrazenie nameranej hodnoty na displeji. Akcia sa pri spustených stopkách vykonáva každú 1ms.

```

private void stopky_timer_Tick(object sender, EventArgs e)
{
    TimeSpan elapsed = sw.Elapsed;
    cas.Text = String.Format("{0:00}:{1:00}:{2:00}:{3:00}",
    Math.Floor(elapsed.TotalHours), elapsed.Minutes,
    elapsed.Seconds, elapsed.Milliseconds);
}

```

## 4.2 Používateľská príručka

Poskytuje informácie o minimálnych požiadavkách na systém, na ktorom bude program pracovať. Opisuje inštaláciu aplikácie, ako aj podrobný návod ako ju využívať optimálne.

### 4.2.1 Minimálne požiadavky

Minimálne požiadavky pre spustenie aplikácie na servery:

- Operačný systém Windows CE 6
- Procesor Samsung 6410
- RAM 512 MB
- Dotykový monitor
- 1xUSB port
- 10 MB voľného miesta na pevnom disku
- 1x Ultrazvukový senzor SRF02 (najviac 4)

Minimálne požiadavky pre spustenie aplikácie na stolovom počítači:

- Operačný systém Windows XP
- Procesor 1000 MHz
- RAM 512 MB
- 3xUSB port
- 10 MB voľného miesta na pevnom disku
- 1x Ultrazvukový senzor
- 2x USB kamera (1x USB kamera, 1x integrovaná kamera)

### 4.2.2 Inštalácia

Aplikáciu nie je nutné inštalovať. Stačí ju nakopírovať do požadovanej zložky, z ktorej sa bude spúšťať. K samotnej aplikácii je nutné pribalíť pomocné súbory *beep\_ok.wav*, *beep\_err.wav*, *AForge.Video.DirectShow.dll* a *AForge.Video.dll*. Do systému je však nutné nainštalovať ovládač pre rozhranie zabezpečujúce komunikáciu aplikácie so senzorom. Zabezpečuje vytvorenie virtuálneho COM portu, ktorý sa priradí k portu USB. Do zložky „Windows“ nakopírujeme súbor s názvom *FTD2XX.dll*. Následne vložíme senzor do USB portu a zadáme názov ovládača, ktorý sme nakopírovali do zložky „Windows“. Aplikáciu je týmto možné spustiť dvojklikom na súbor *parkovaci\_asistent.exe*.

### 4.2.3 Používateľské rozhranie

Poskytuje používateľovi návod, ako pracovať s aplikáciou a využívať všetky jej funkcie.

#### 4.2.3.1 Hlavné okno



Obr. 1 Hlavné okno aplikácie (PC)

Hlavné okno aplikácie pre stolový počítač obsahuje tri základné ikony, ktoré spúšťajú jeden z troch režimov. Rovnako poskytuje ikonu pre nastavenie zvukov a ikonu pre ukončenie aplikácie.

- Kliknutím na ľavú veľkú ikonu sa aktivuje režim parkovacej kamery
- Kliknutím na strednú veľkú ikonu sa aktivuje režim ultrazvukového senzora
- Kliknutím na pravú veľkú ikonu sa aktivuje duálny režim
- Kliknutím na ikonu reproduktora v ľavom dolnom rohu sa vypnú/zapnú zvuky
- Kliknutím na pravú dolnú ikonu sa ukončí aplikácia

Hlavné okno pre server je veľmi podobné ako okno pre stolový počítač. Na rozdiel od hlavného okna pre stolový počítač obsahuje dve veľké ikony, ktoré poskytujú prácu v jednom z dvoch režimov. Rovnako poskytujú menšie ikony, ktoré zabezpečujú nastavenia senzorov, zvukov a využitie parkovacích stopiek.

- Kliknutím na ľavú veľkú ikonu sa aktivuje režim parkovacieho senzora
- Kliknutím na pravú veľkú ikonu sa aktivuje režim súhra
- Kliknutím na ozubené koliesko sa aktivuje režim konfigurácie senzorov
- Kliknutím na stopky sa aktivuje režim parkovacích stopiek



Nastavenie zvukov a ukončenia aplikácie je totožné s *obrázkom 1*.



Obr. 2 Hlavné okno aplikácie pre server

**Zoznam tlačidiel:**



Ikona režimu parkovacej kamery



Ikona režimu ultrazvukového senzora



Ikona duálneho režimu



Ikona režimu súhra



Ikona zapnutej hlasitosti (vypnutie zvukov)



Ikona vypnutej hlasitosti (zapnutie zvukov)



Ikona parkovacích stopiek



Ikona konfigurácie senzorov



Ikona ukončenia aplikácie

#### 4.2.3.2 Režim parkovacej kamery



Obr. 3 Režim parkovacej kamery

Režim parkovacej kamery pozostáva z časti pre prehrávanie videa, ktorá sa nazýva zobrazovacie pole. Vrchná časť režimu obsahuje identifikáciu režimu. Režim sa skladá z troch hlavných tlačidiel.

- Kliknutím na tlačidlo „Kamera 1“ sa aktivuje prehrávanie videa z kamery 1
- Kliknutím na tlačidlo „Kamera 2“ sa aktivuje prehrávanie videa z kamery 2
- Kliknutím na tlačidlo ukončenia režimu program ukončí prehrávanie videa a nastaví sa hlavná obrazovka

#### Zoznam tlačidiel:



Ikona štartu premietania videa prvej kamery



Ikona signalizuje aktívnu prvú kameru



Ikona štartu premietania videa druhej kamery



Ikona signalizuje aktívnu druhú kameru



Ikona ukončenia režimu

#### 4.2.3.3 Režim ultrazvukového senzora



Obr. 4 Režim ultrazvukového senzora

Režim ultrazvukového senzora poskytuje spracované informácie získané z ultrazvukového senzora. Identifikácia režimu je zobrazená v ľavom hornom rohu. Presná vzdialenosť je zobrazená na displeji v pravom hornom rohu. Hodnota je vyjadrená v centimetroch. Nameraná vzdialenosť je rovnako zobrazená na diódovom paneli, kde sa v závislosti od vzdialenosti vozidla od prekážky aktivujú príslušné diódy.

Režim je automatický a nie je nutná žiadna aktivácia funkcií počas činnosti režimu. Obsahuje jediné tlačidlo a to ukončenie režimu.

#### 4.2.3.4 Duálny režim



**Obr. 5** Duálny režim

Duálny režim poskytuje informácie z parkovacej kamery ako aj z ultrazvukového senzora. V ľavej hornej časti obsahuje identifikáciu režimu. Väčšiu časť obrazovky tvorí zobrazovacie pole, ktoré slúži na premietanie videa z parkovacej kamery. Diódový panel v pravej časti obrazovky slúži pre zobrazenie informácie o najkratšej vzdialenosti prekážky od vozidla. Tento režim je rovnako automatický a po jeho aktivácii nie je nutná ďalšia obsluha. Zobrazenie videa sa aktivuje automaticky ako aj meranie vzdialenosti. Obsahuje jedinú ikonu a to ikonu pre ukončenie režimu, podobne ako režim ultrazvukového senzora na *obrázku 4*.

#### 4.2.3.5 Režim súhra

Režim súhra poskytuje spracovanie nameraných vzdialeností štyrmi nezávislými senzormi, pričom na diódovom paneli poskytuje zobrazenie najkratšej vzdialenosti zo všetkých sensorov. V ľavom hornom rohu je identifikácia režimu, rovnako ako v predošlých režimoch. Pravá strana pozostáva z diódového panela, ktorý zobrazuje najkratšiu vzdialenosť všetkých sensorov. Oblasť „Vyhodnotenie vzdialenosti senzormi“ predstavuje štyri nezávislé merania štyroch sensorov, kedy každý z nich zobrazuje svoje namerané vzdialenosti v podobe farebných prúžkov. Ľavá stredná časť obrazovky zobrazuje aktuálny stav jednotlivých sensorov. Režim je zobrazený na *obrázku 6*.





Obr. 6 Režim súhra

**Stavy senzorov:**



Senzor v režime merania



Senzor v neaktívnom režime (vypnutý používateľom)



Nepripojený senzor

Po aktivácii režim nevyžaduje ďalšiu konfiguráciu a pracuje v automatickom režime. Stav senzorov sa nastaví automaticky na základe kontroly počas štartu režimu. Režim obsahuje jediné tlačidlo a to ukončenie režimu.

***4.2.3.6 Konfiguračný režim***

Režim konfigurácie poskytuje nastavenie meracích rozsahov jednotlivých senzorov, ako aj možnosť zmeny stavu senzora. Zmenu senzora je možné uskutočniť kliknutím na požadovanú ikonu v ľavej časti obrazovky v červenom poli.

Zmenu základných rozsahov jednotlivých senzorov je možné vykonať klikaním na modré šípky pri danom senzore. Šípka smerujúca doľava znižuje meraný rozsah a šípka smerujúca doprava zväčšuje meraný rozsah. Nastavenia jednotlivých senzorov je možné sledovať v strednej časti obrazovky, kedy sú reprezentované zelenými pomlčkami. Každá pomlčka reprezentuje hodnotu 10cm. Režim konfigurácie je zobrazený na obrázku 7.



Obr. 7 Režim konfigurácie

**Ikony ovládania senzorov:**



Ikona aktivuje meranie senzora, zobrazenie ikony predstavuje deaktivovaný senzor



Ikona deaktivuje senzor, zobrazenie ikony predstavuje aktivovaný senzor



Ikona informuje o tom, že senzor nebol zistený počas inicializácie systému. V tomto stave nie je povolená žiadna konfigurácia senzora

**Ikony nastavenia rozsahov:**



Kliknutím na ikonu sa zmenší meraný rozsah



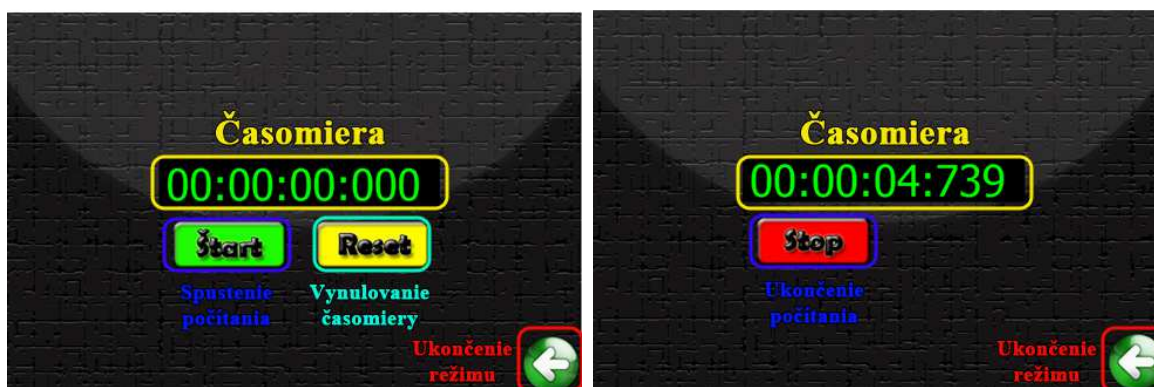
Kliknutím na ikonu sa zväčší meraný rozsah



Ikona reprezentuje hodnotu 10cm

Konfiguračný režim sa ukončuje kliknutím na ikonu ukončenia režimu, kedy sa všetky nastavenia uložia a sú aplikované na funkcie merania.

#### 4.2.3.7 Režim parkovacích stopiek



Obr. 8 Režimy parkovacích stopiek

Po štarte režimu parkovacích stopiek môžu nastať dva stavy, *obrázok 8*, v ktorých sa môžu stopky nachádzať. Prvý stav, ktorý môžu nadobudnúť je stav zastaveného počítania. Na obrazovke sa v tomto stave zobrazí časomiera, ktorá nepočíta a obsahuje dve tlačidlá. Tlačidlo štartu, ktoré umožňuje spustenie počítania. Druhým tlačidlom je tlačidlo reset, ktoré umožňuje vynulovanie časomiere.

Druhým stavom je stav počítania. Na obrazovke je v tomto stave zobrazená časomiera, ktorá počíta. Dostupné je len jedno tlačidlo a to na ovládanie stopiek a to vypnutie počítania.

Režim sa ukončuje stlačením na tlačidlo späť, ktoré je dostupné v každom stave stopiek.

#### Tlačidlá časomiere:



Ikona spúšťa meranie parkovacích stopiek



Ikona ukončuje meranie parkovacích stopiek



Ikona nastaví časomieru na nulovú hodnotu

## 5 Literatúra

- [1] ŠTEFANOVIČ, J., 2008, Parkuj, parkuj, vykrúcaj[online],Autoviny.zoznam.sk[cit. 7.4. 2011].Dostupné na internete:  
<http://autoviny.zoznam.sk/cl/100223/270131/Parkovaci-asistent--Parkuj--parkuj--vykrucaj--->
- [2] LEXUS, 2011, Inteligentný parkovací asistent[online],lexus.sk[cit. 7.4. 2011].Dostupné na internete:<http://www.lexus.sk/range/ls/key-features/interior/interior-intelligent-park-assist.aspx>
- [3] GSM ALARM, 2008, Bezdrôtový cúvací systém[online],gsmalarm.sk[cit. 7.4. 2011].Dostupné na internete:<http://www.gsmalarm.sk/bezdrotovy-cuvaci-system-4000-displej-p-55554733.html>
- [4] PARKOVACIE-SENZORY, 2009, PS-082A [online], parkovacie-senzory.sk[cit. 7.4. 2011].Dostupné na internete:<http://www.parkovacie-senzory.sk/ps082a-parkovaci-asistent-s-kamerou-a-35-displejom-v-spatno-p-738.html?osCsid=ecqc7qdlh5scedc5od4a9m3n82>
- [5] ANDREJČÁK, T.,2009, Zabudnite na stres, autá vedia parkovať aj samy[online], auto.pravda.sk[cit. 7.4. 2011].Dostupné na internete:  
[http://auto.pravda.sk/zabudnite-na-stres-auta-vedia-parkovat-aj-samy-f4t-/sk-amagazin.asp?c=A090226\\_131855\\_sk-amagazin\\_P39](http://auto.pravda.sk/zabudnite-na-stres-auta-vedia-parkovat-aj-samy-f4t-/sk-amagazin.asp?c=A090226_131855_sk-amagazin_P39)
- [6] GOLD, A., 2009, Parking assistance [online], cars.about.com[cit. 7.4. 2011]. Dostupné na internete: <http://cars.about.com/od/volkswagen/ig/2009-Volkswagen-CC-photos/2009-Volkswagen-CC-park-assist.htm>
- [7] VOLKSWAGEN, 2010, Parking assistance[online], vw.co[cit. 7.4. 2011].Dostupné na internete: <http://www.vw.co.za/en/redirects/about/technology/park-assist/.html>
- [8] BLOG AUTOMOBILE, 2010, Volkswagen: Park Assist est bien un nom... masculine[online], blogautomobile.fr[cit. 7.4. 2011]. Dostupné na internete:



<http://blogautomobile.fr/volkswagen-park-assist-est-bien-un-nom-masculin-44448#axzz1IpDBA5E9>

- [9] FUNTORO, 2010, Unique multimedia system of the future[online], funtoro.com[cit. 9.4. 2011], Dostupné na internete:[www.funtoro.com](http://www.funtoro.com)
- [10] MOLPIR, 2009, Predné a zadné parkovacie senzory s LCD displejom, 14s.
- [11] BÁLEK R.: ÚVOD DO AKUSTIKY – Ultrazvuk, text k přednáškám, ČVUT, Praha, 240s., 2007
- [12] MSI, 2011, MSI Profile[online], [eu.msi.com](http://eu.msi.com)[cit. 6.5. 2011]. Dostupné na internete: <http://eu.msi.com/product/mm/>
- [13] CAR RENTAL, 2010, Volkswagen Tiguan[online], [carrentaltoday.com](http://carrentaltoday.com)[cit. 6.5. 2011]. Dostupné na internete: <http://www.carrentaltoday.com/volkswagen-tiguan-for-rent/>
- [14] ŠVEHLA, Š., FIGURA, Z.: Ultrazvuk v technológii. Západoslovenské tlačiarne, Bratislava, 528 s., 1984, 1. Vydanie
- [15] REM TECHNIC, 2002, senzorika[online], [rem-technic.cz](http://rem-technic.cz)[cit. 10.11. 2011]. Dostupné na internete: <http://www.rem-technic.cz/senzorika/ultrazvukove-senzory/programovatelne-ultrazvukove-senzory-p42-335.html>
- [16] OEM AUTOMATIC, 2009, ultrazvukové senzory[online], [oemautomatic.cz](http://oemautomatic.cz)[cit. 10.11. 2011]. Dostupné na internete: [http://www.oemautomatic.cz/wps/wcm/connect/3bd6558043d0fbe3b182b9e01e85254b/Mic\\_MailingNews2010\\_CZ\\_01-09-10.pdf?MOD=AJPERES](http://www.oemautomatic.cz/wps/wcm/connect/3bd6558043d0fbe3b182b9e01e85254b/Mic_MailingNews2010_CZ_01-09-10.pdf?MOD=AJPERES)
- [17] EMZET ROBOTICS, 2005, srf08[online], [emzet-robotics.sk](http://emzet-robotics.sk)[cit. 10.11. 2011]. Dostupné na internete: <http://eshop.emzet-robotics.sk/predaj/97/SRF08/>

- [18] EMZET ROBOTICS, 2005, srf02[online], emzet-robotics.sk[cit. 10.11. 2011].  
Dostupné na internete: <http://eshop.emzet-robotics.sk/predaj/98/SRF02/>
- [19] STANDARDS, 1995, rs-232[online], camiresearch.com[cit. 10.11. 2011].  
Dostupné na internete:  
[http://www.camiresearch.com/Data\\_Com\\_Basics/RS232\\_standard.html](http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html)
- [20] ROBOT ELECTRONICS, 2008, i<sup>2</sup>c[online], robot-electronics.com[cit. 10.11. 2011]. Dostupné na internete: [http://www.robot-electronics.co.uk/htm/usb\\_i2c\\_tech.htm](http://www.robot-electronics.co.uk/htm/usb_i2c_tech.htm)
- [21] WIKIPEDIA, 2007, java[online], en.wikipedia.org[cit. 7.5. 2011]. Dostupná na internete: [http://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))
- [22] WIKIPEDIA, 2005, c++[online], en.wikipedia.org[cit. 7.5. 2011]. Dostupné na internete: [http://en.wikipedia.org/wiki/C\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/C_(programming_language))
- [23] WIKIPEDIA, 2005, .net framework[online], en.wikipedia.org[cit. 7.5. 2011]. Dostupné na internete: [http://en.wikipedia.org/wiki/.NET\\_Framework](http://en.wikipedia.org/wiki/.NET_Framework)
- [24] WIKIPEDIA, 2006, c sharp[online], en.wikipedia.org[cit. 8.5. 2011]. Dostupné na internete: [http://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [25] VIRKLER, R.: Parkovací asistent. Bratislava: FIIT STU, 2011, 29 s. Diplomový projekt 1.
- [26] VIRKLER, R.: Parkovací asistent. Bratislava: FIIT STU, 2011, 55 s. Diplomový projekt 2.
- [27] MICROSOFT, 2012, Microsoft[online], microsoft.com[cit. 21.4. 2012]. Dostupné na internete : <http://www.microsoft.com/en-us/default.aspx>
- [28] VALEO, 2008, Valeo[online], valeo.com[cit. 30.4. 2012]. Dostupné na internete: <http://www.valeo.com/en.html?L=223>

## **Príloha A : Obsah elektronického média**

./obsah.txt	- obsah CD média
./anotacia.pdf	- anotácia v slovenskom jazyku
./annotation.pdf	- anotácia v anglickom jazyku
./dokumentacia/dp3.pdf	- dokumentácia projektu
./program_pc/spustitelny program/*	- spustiteľný program spolu s potrebnými súborami
./program_pc/zdrojove subory/*	- zdrojové súbory program
./program_s/spustitelny program/*	- spustiteľný program spolu s potrebnými súborami
./program_s/zdrojove subory/*	- zdrojové súbory programu

## **Príloha B : Elektronické médium**