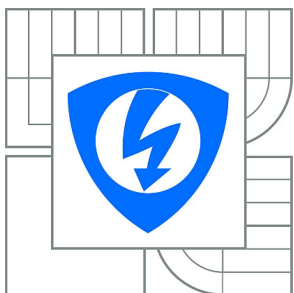


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# MONITOROVÁNÍ DOPRAVNÍ SITUACE S VYUŽITÍM RASPBERRY PI

TRAFFIC MONITORING USING RASPBERRY PI

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

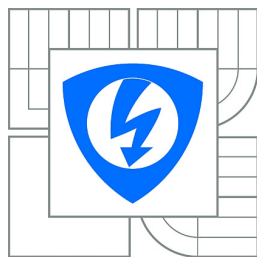
Bc. MICHAL ZACPAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETER HONEC, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Diplomová práce

magisterský navazující studijní obor  
Kybernetika, automatizace a měření

**Student:** Bc. Michal Zacpal

**ID:** 134663

**Ročník:** 2

**Akademický rok:** 2014/2015

## NÁZEV TÉMATU:

### Monitorování dopravní situace s využitím Raspberry PI

## POKYNY PRO VYPRACOVÁNÍ:

Cílem práce bude vytvořit systém pro monitorování dopravní situace v reálném provozu s využitím hardwaru Raspberry PI.

1. Proveďte rešerši.
2. Navrhněte hardwarovou koncepci zařízení.
3. Sestavte a oživte modul včetně nutných periférií.
4. Vytvořte softwarové vybavení - GUI a algoritmy detekce jízdních pruhů.
5. Ověřte funkcionalitu a vyhodnoťte spolehlivost.

## DOPORUČENÁ LITERATURA:

Hlaváč, Šonka, Počítačové vidění.

Šonka, Hlaváč, Boyle - IMAGE PROCESSING, ANALYSIS, AND MACHINE VISION,

**Termín zadání:** 9.2.2015

**Termín odevzdání:** 18.5.2015

**Vedoucí práce:** Ing. Peter Honec, Ph.D.

**Konzultanti diplomové práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

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

## ABSTRAKT

Práce se zabývá návrhem a následnou realizací jednotky pro monitorování dopravní situace v reálném provozu s využitím hardwaru Raspberry PI. V první části je uveden stručný přehled asistenčních systémů využívajících detekci jízdních pruhů od různých výrobců. Dále jsou popsány dva odlišné způsoby detekce jízdních pruhů. Poté je popsána snímaná scéna s ohledem na jízdní pruhy. Následně se již práce zabývá praktickou částí, a to návrhem a realizací podpůrné elektroniky, výběrem jednotlivých komponent včetně potřebné úpravy kamer, mechanickým návrhem a zpracováním jednotky. V další části je popsán výběr a instalace vhodných softwarových komponent nutných pro chod jednotky a také výběr vývojových nástrojů pro vytváření uživatelské aplikace. Po popisu vytvořeného grafického uživatelského rozhraní se práce zabývá vytvořenými algoritmy pro detekci jízdních pruhů. Na konci práce je vyhodnocena spolehlivost jednotky v reálné dopravní situaci. V příloze se nachází technická dokumentace vytvořené jednotky.

## KLÍČOVÁ SLOVA

Asistenční jednotka, detekce jízdních pruhů, dotykový displej, IPM, Raspberry PI, Raspberry PI kamerový modul, Raspbian, OpenCV, QT Creator, TVTM, zpracování obrazu

## ABSTRACT

This thesis describes the design and subsequent implementation of a unit for traffic monitoring using Raspberry PI. First section provides a quick overview of assistance systems, which use a road lane detection techniques. Next there is a description of two different methods for road lane detection. Follow the description of monitoring scene. Then the work describe the practical part including the design and realization of supporting electronics, selecting of each components, including the modifying of cameras, mechanical design and creating of unit. Another section is about selection and installation of appropriate software components necessary for running of the unit and the selection of development tools for creating user application. After description of graphical user interface, there is a description of road lanes detection algorithm. At the end of the thesis is summarized a reliability of unit in real traffic situation. At the appendix there are technical drawings, describing the unit.

## KEYWORDS

Assistance unit, image processing, IPM, Raspberry PI, Raspberry PI camera module, Raspbian, OpenCV, QT Creator, road lane detection, touchscreen, TVTM

ZACPAL, Michal *Monitorování dopravní situace s využitím Raspberry PI*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2015. 106 s. Vedoucí práce byl Ing. Peter Honec, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Monitorování dopravní situace s využitím Raspberry PI“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Peteru Honcovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Brno .....

.....

(podpis autora)

# OBSAH

Úvod	12
<b>1 Asistent sledování vybočení z jízdního pruhu</b>	<b>13</b>
1.1 Přehled současných asistentů sledování vybočení z jízdního pruhu . . .	13
1.1.1 Dodavatelé . . . . .	13
1.1.2 Výrobci vozidel . . . . .	14
<b>2 Detekce jízdních pruhů v obraze</b>	<b>16</b>
2.1 Inverzní perspektivní mapování IPM [10],[14] . . . . .	16
2.2 Top - View Transformation model . . . . .	17
2.2.1 Estimace parametrů $H$ , $f$ , $\Theta$ . . . . .	18
2.2.2 Interpolace jasových hodnot . . . . .	19
2.2.3 Hranové filtry, vyhlazování obrazu . . . . .	20
2.2.4 Lokalizace jízdních pruhů . . . . .	21
2.2.5 Segmentace jízdních pruhů za pomoci prahování . . . . .	22
2.3 Algoritmus CHEVP - Canny/Hough Estimation of Vanishing Points) [32] . . . . .	24
2.3.1 Matematický model jízdních pruhů . . . . .	24
<b>3 Popis snímané scény</b>	<b>27</b>
3.1 Značení jízdních pruhů . . . . .	27
<b>4 Hardware</b>	<b>29</b>
4.1 Blokové schéma . . . . .	29
4.2 Napájení . . . . .	30
4.3 Raspberry Pi . . . . .	31
4.3.1 GPIO . . . . .	31
4.4 Displej . . . . .	32
4.5 Rozšiřující modul pro periferie jednotky . . . . .	34
4.5.1 Napájení asistenční jednotky . . . . .	34
4.5.2 Informačně komunikační . . . . .	36
4.5.3 Převodník SPI/CAN . . . . .	37
4.5.4 FM Radio přijímač . . . . .	37
4.6 Popis snímací aparatury . . . . .	37
4.6.1 Raspberry Pi kamerový modul . . . . .	37
4.6.2 Objektiv . . . . .	37
4.6.3 Cam extender . . . . .	39
4.6.4 Jednotka pro monitorování dopravní situace před vozidlem . . . . .	40

4.6.5	Web kamera - monitorování dopravní situace za vozidlem . . .	41
<b>5</b>	<b>Mechanické zpracování jednotky</b>	<b>42</b>
<b>6</b>	<b>Softwarové vybavení</b>	<b>44</b>
6.1	Raspbian . . . . .	44
6.1.1	Instalace . . . . .	44
6.1.2	Základní nastavení, první spuštění . . . . .	45
6.1.3	Vzdálená správa . . . . .	46
6.1.4	Přístup k GPIO konektoru . . . . .	47
6.1.5	Měření teploty s DS18B20 . . . . .	47
6.1.6	Raspberry Pi kamerový modul . . . . .	48
6.1.7	Dotyková obrazovka . . . . .	48
6.1.8	Virtuální klávesnice . . . . .	49
6.2	QT Creator . . . . .	49
6.3	OpenCV . . . . .	50
6.4	Test knihoven OpenCV, RaspiCam a WiringPi . . . . .	51
6.5	Porovnání rychlosti Raspberry Pi kamerového modulu a USB webka- mery . . . . .	53
6.6	Softwarový přechod z Raspberry Pi verze 1 na verzi 2 . . . . .	53
<b>7</b>	<b>Grafické uživatelské rozhraní - GUI</b>	<b>54</b>
<b>8</b>	<b>Algoritmus detekce a rozpoznání jízdních pruhů</b>	<b>61</b>
8.1	Vytvořené třídy, struktury . . . . .	61
8.2	Výpočet TVTM . . . . .	62
8.2.1	Estimace parametrů $H$ , $f$ , $\Theta$ - OptimizeTVTM . . . . .	63
8.2.2	Transformace obrazu - LUT . . . . .	63
8.3	Detekce jízdních pruhů . . . . .	63
8.3.1	Detekce hran - ImEge . . . . .	63
8.3.2	Horizontální histogram - HorizHist . . . . .	64
8.3.3	Prahování - ImageThresholdIter . . . . .	64
8.3.4	Rozpoznání a klasifikace podélných dopravních čar . . . . .	65
<b>9</b>	<b>Vyhodnocení spolehlivosti systému pro detekci a rozpoznání jízdních pruhů</b>	<b>70</b>
<b>10</b>	<b>Závěr</b>	<b>74</b>
	<b>Literatura</b>	<b>77</b>

Seznam symbolů, veličin a zkratk	81
Seznam příloh	82
1 Program pro test knihoven OpenCV, RaspiCam a WiringPi	83
2 Program pro měření rychlosti FPS kamer	84
3 Obsah přiloženého CD	86
4 Výkresová dokumentace monitorovací jednotky	87



# SEZNAM OBRÁZKŮ

1.1	Ukázka kamer pro LDWS. [7], [8], [5], [6], [30]	15
2.1	Ilustrační obrázek k inverznímu perspektivnímu mapování IPM	17
2.2	Příklad TVTM, pro vysvětlení estimace parametrů $H$ , $f$ , $\theta$	19
2.3	Výsledky TVTM transformace pro různé hodnoty parametrů $H$ , $f$ , $\theta$	20
2.4	Ukázka detekce jízdních pruhů za pomoci hranových filtrů: A - filtr ef1, B - filtr ef2, C - filtr ef1 + filtr ef2	21
2.5	Vyhranovaný snímek proložený horizontálním histogramem vypočteným dle vzorce 2.14.	22
3.1	Ukázky snímané scény před vozidlem	27
3.2	Ukázky snímané scény za vozidlem	27
4.1	Konektor ISO 10487	29
4.2	Schema zapojení stabilizátoru s <i>LM2576T-5</i>	30
4.3	Popis konektorů rozšiřujícího modulu a jeho umístění na Raspberry Pi s USB propojkou	34
4.4	Propojovací konektor Raspberry Pi - rozšiřující deska	35
4.5	Schéma zapojení rozšiřující desky - napájecí část	35
4.6	Zapojení snímačů DS18B20	36
4.7	Doplňující obrázek k výpočtu ohniska (A) a rozlišení počtu px na šířku čáry (B)	38
4.8	Schéma zapojení převodníku Cam extender	40
4.9	Složení snímací aparatury	41
4.10	Kamera pro monitorování dopravní situace před vozidlem	41
5.1	3D model asistenční jednotky vymodelovaný v programu SketchUP (vlevo) a skutečná jednotka (vpravo)	43
5.2	3D model zezadu s odstraněným plechovým krytem (vlevo) a skutečná jednotka (vpravo)	43
6.1	Konfigurační menu Raspberry Pi	45
6.2	Zapojení tlačítka k programu pro otestování knihoven OpenCV, Raspicam a WiringPi	52
7.1	Grafické uživatelské rozhraní - popis jednotlivých funkcí	54
7.2	Roletové menu pro maximalizaci / minimalizaci aplikace a aktivaci / deaktivaci zvukové signalizace	55
7.3	Ukázka zobrazení okna pro obraz z přední i zadní kamery	56
7.4	Ukázka zobrazení okna pro zadní kameru	57

7.5	Ikony pro vizuální upozornění řidiče. Zleva: vše je v pořádku ( $iAlarm = 0$ ), upozornění na námrazu (ikona se objeví pokud vnější teplota klesne pod $0\text{ }^{\circ}\text{C}$ , $iAlarm = 1$ ), přestávka (ikona momentálně není implementovaná, $iAlarm = 2$ ) . . . . .	57
7.6	Ukázka zobrazení okna pro zobrazující výsledky detekce a rozpoznání dopravních čar . . . . .	59
7.7	Ukázka zobrazení okna pro nastavení asistenční jednotky . . . . .	59
8.1	Použitý „vertikální“ Gausův filtr pro filtraci šumu a zvýraznění vertikálních hran . . . . .	64
8.2	Ukázka prahování: A, B - Určení prahu iterativní metodou. C, D - Určení prahu Otsuovou metodou . . . . .	65
8.3	Vývojový diagram metody <code>SeparateLane</code> třídy <code>RoadLane</code> . . . . .	67
8.4	Ukázka vyhranovaných snímků pro různé hodnoty osvětlení snímání scény . . . . .	67
8.5	Průběh algoritmu pro detekci jízdních pruhů . . . . .	68
8.6	Vývojový diagram pro funkci <code>OptimizeTVTM</code> . . . . .	69
9.1	Ukázka detekce a rozpoznání vodorovného dopravního značení . . . . .	72
9.2	Ukázka detekce a rozpoznání vodorovného dopravního značení v noci. . . . .	72
9.3	Ukázka detekce a rozpoznání vodorovného dopravního značení. Druhý řádek zleva: silnice bez dopravních čar, špatná detekce, přerušená čára vlivem nečistoty. Třetí řádek zleva: Přerušování stěračem, praskliny ve vozovce a opravená vozovka. Přerušovaná světle modrá čára znázorňuje střed vozidla. Některé snímky obsahují zobrazené hodnoty deskriptorů ( <code>Desc1</code> a <code>Desc2</code> ). . . . .	73
10.1	Ukázka zabudované jednotky pro monitorování dopravní situace ve vozidle . . . . .	74

## SEZNAM TABULEK

4.1	Parametry stabilizátoru LM2576T-5 . . . . .	30
4.2	Srovnání Raspberry Pi 1 B+ a Raspberry Pi 2 B+ . . . . .	32
4.3	GPIO konektor Raspberry pi . . . . .	33
4.4	Parametry CMOS senzoru OmniVision OV5647, použitého v Raspberry Pi (RPi) kamerovém modulu. [18] . . . . .	38
4.5	Parametry vhodných objektivů . . . . .	39
6.1	Porovnání FPS Raspberry Pi kamerového modulu a USB webkamery pro $n = 10\,000$ snímků při rozlišení $640 \times 480$ px . . . . .	53
8.1	Soupis funkcí vytvořené knihovny <i>imageprocessing</i> . . . . .	61
9.1	Úspěšnost detekce a rozpoznání vodorovného dopravního značení pro dálnice a silnice I. třídy . . . . .	70
9.2	Úspěšnost detekce a rozpoznání vodorovného dopravního značení pro pozemní komunikace mimo dálnice a silnice I. třídy obsahující vodorovné dopravní značení . . . . .	70
9.3	Úspěšnost detekce a rozpoznání vodorovného dopravního značení pro dálnice a silnice I. třídy bez započtených snímků s poškozenými dopravními čarami. . . . .	71
9.4	Úspěšnost detekce a rozpoznání vodorovného dopravního značení pro pozemní komunikace mimo dálnice a silnice I. třídy obsahující vodorovné dopravní značení, bez započtených snímků s poškozenými dopravními čarami. . . . .	71

# ÚVOD

Asistenční systémy v automobilech už nejsou jen výsadou automobilů vyšší třídy. Některé systémy typu ABS (Anti-lock Brake System), ASR (Anti-Slip Regulation) nebo ESP (Electronic Stability Program) jsou již dodávány ve standardní výbavě. Naopak komplexnější systémy typu AFS (Adaptive Front-light System), automatické parkovací asistenty či asistent pro udržení vozidla v jízdním pruhu jsou zatím výsadou automobilů vyšší střední třídy. A právě v posledním jmenovaném asistentu nacházejí uplatnění algoritmy pro detekci jízdních pruhů. Tato diplomová práce se zabývá návrhem a realizací modulu monitorující dopravní situaci s detekcí jízdních pruhů. Jako řídicí modul byl určen malý počítač Raspberry Pi.

Cílem této práce je návrh a následně realizace modulu pro monitorování děje v okolí vozidla. Vytvořený modul by měl být především schopen detekce jízdních pruhů. Systém bude kromě již zmiňovaného Raspberry Pi vybaven i zobrazovací jednotkou a grafickým uživatelským rozhraním (GUI). Pro snadnější interakci a ovládání modulu bude displej vybaven dotykovou vrstvou. Vozovka a jízdní pruhy budou snímány za pomoci Raspberry Pi kamerového modulu a webkamery. U obou snímacích zařízení dojde k výměně jejich stávajících objektivů za objektivy splňující podmínky snímané scény. Celý systém bude pro snadné zabudování do vozidla o velikosti 2DIN autorádia dle normy ISO 7736, připojení (napájení, informace o klíčku v zapalování, . . .) bude realizováno konektorem dle ISO 10487. Výsledné algoritmy pro detekci jízdních pruhů by měly být z hlediska bezpečnosti a použitého hardwaru robustní a výpočetně nenáročné.

Jízdní pruh je dle § 2 písm. t) zákona č. 361/2000 Sb definován jako „část vozovky dovolující jízdu vozidel jiných než dvoukolových (motocyklů) v jednom jízdním proudu za sebou“. Podle druhu pozemní komunikace musí, případně nemusí být jízdní pruh vyznačen vodorovným dopravním značením. Tato práce se věnuje detekci jízdních pruhů obsahujících vodorovné dopravní značení, někdy také nazývané jako podélné dopravní čáry. Jízdní pruhy budou detekované ze snímku pořízeného kamerou. Pro zpracování obrazu bude použita multiplatformní knihovna OpenCV.

# 1 ASISTENT SLEDOVÁNÍ VYBOČENÍ Z JÍZDNÍHO PRUHU

Systémy, které využívají detekce jízdních pruhů se obvykle dělí na dva druhy, a to na systémy Lane departure warning system (LDWS), které pouze varují řidiče při neúmyslném vybočení z jízdního pruhu, a na systémy Lane keeping system (LKS), které řidiče varují při neúmyslném vybočení z jízdního pruhu a v případě, že nedojde od řidiče ke zpětné reakci, nebo je příliš pomalá, systém automaticky navrátí vozidlo do správného směru.

Asistent varování před vybočením z jízdního pruhu, dále jen LDWS, je asistenční systém vozidla, který slouží k varování řidiče před nechtěným vybočením z jízdního pruhu. Minimalizuje tak riziko nehody způsobené nepozorností řidiče. Pokud je systém informován řidičem o změně směru, pak nereaguje. Obvykle je systém aktivní při rychlosti vozidla nad 60 km/h a při dobře rozpoznatelném nepoškozeném vodorovném dopravním značení. Jeho hlavní částí je kamera snímající vodorovné dopravní značení. Při neúmyslném vybočení vozidla z jízdního pruhu o tom systém informuje řidiče vizuálně na přístrojové desce, akusticky, anebo vibracemi volantů či sedačky. Vzhledem ke snímání kamerou mají systémy omezenou až nulovou funkčnost při hustém dešti, sněžení nebo mlze. Některé systémy mají problém s vícenásobným značením nacházejícím se v opravovaných úsecích cest. V dnešní době jsou ve většině případů vozidla opatřena systémem LKS. Tyto systémy je obvykle možné vypnout. První systémy LDWS byly nasazovány od roku 2000, a to nejprve v nákladních vozidlech a až posléze jimi začaly být vybavovány osobní automobily.

## 1.1 Přehled současných asistentů sledování vybočení z jízdního pruhu

Během této práce bylo osloveno několik výrobců automobilů za účelem poskytnutí základních informací o jejich systémech LDWS a LKS. Materiály byly poskytnuty pouze firmami BMW a Daimler. Informace od jiných výrobců byly nalezeny na internetu nebo byly získány od řidičů těchto vozidel a mohou se tím pádem lišit od skutečnosti. Přehled je uveden pouze pro výrobce automobilů, kteří mají zastoupení na českém, případně evropském trhu.

### 1.1.1 Dodavatelé

Mezi hlavní dodavatele systémů pro sledování dopravní situace před vozidlem patří Bosch, Continental, Delphi, TRW a Valeo.

### **Bosch Víceúčelová kamera (Multi purpose camera) [7]**

Může být použita nejen pro detekci vybočení z jízdního pruhu, ale také pro ostatní asistenční systémy jako např. rozpoznávání dopravních značek, inteligentní ovládání světlometů nebo detekce předmětů a překážek. Kamera obsahuje CMOS senzor s rozlišením 1280 x 960 pixelů. Kamera obsahuje interní řídicí jednotku pro zpracování obrazu.

### **Bosch Stereo videokamera (Stereo video camera) [8]**

Rozšiřuje monokameru o trojrozměrnou detekci okolí. Obsahuje dva čipy CMOS s rozlišením 1280 x 960 pixelů. Zaznamenává (ve 3D) horizontální viditelnou oblast v úhlu 45° s dosahem 50 m.

Firma **Continental** nabízí stejně jako Bosch multifunkční mono kameru (Multi Function Mono Camera – MFC) [3] a multifunkční stereo kameru (Multi Function Stereo Camera – MFS) s blíže nespécifikovanými parametry.

### **Delphi radarový a kamerový systém (RACam) [4]**

Sdružuje funkci kamery a radaru do jednoho zařízení o velikosti 123 x 68 x 38 mm.

### **Delphi IFV-100 (Delphi Intelligent Forward View Camera (100 Series)) [5]**

Má integrovaný procesor pro zpracování obrazu. Obsahuje CMOS senzor. Úhel záběru kamery je 45° v horizontálním směru a 29° ve vertikálním směru. Umožňuje použití pro LDW, automatické ovládání světlometů, varování před kolizí s překážkou před vozidlem a rozpoznávání dopravních značek.

### **Delphi IFV-200 (Delphi Intelligent Forward View Camera (200 Series)) [6]**

Rozšiřuje IFV-100 o detekci chodců, v kombinaci s Delphi electronically Scanning Radar poskytuje lepší detekci a klasifikaci objektů.

### **TRW kamera [30]**

Úhel záběru je 21°, s dosahem 80 m. Kamera je aktivní až do rychlosti 260 km/h. Je určena pro použití detekce vodorovného dopravního značení, rozpoznávání dopravních značek, detekce a klasifikace objektů, detekce chodců a automatické ovládání světlometů.

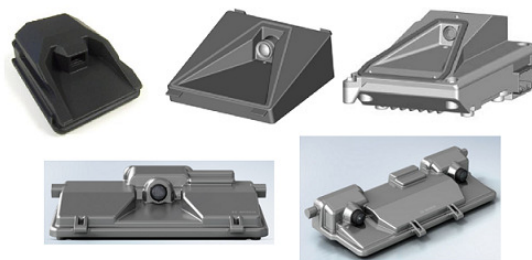
### **Valeo Laneguide® [1]**

Kamera obsahuje CMOS senzor. Kromě funkce LDW umožňuje jednotka rozpoznávání dopravních značek a varování při nedodržení bezpečné vzdálenosti vepředu jedoucího vozidla.

## **1.1.2 Výrobci vozidel**

### **BMW - Lane departure warning**

Systém je aktivní při rychlosti vozidla nad 70 km/h. Registrovatelná vzdálenost je



Obr. 1.1: Ukázka kamer pro LDWS. [7], [8], [5], [6], [30]

až do vzdálenosti 50 m. Při vybočování vozidla z jízdního pruhu informuje řidiče lehkými vibracemi ve volantu a informací na displeji. Systém je možné zapnout / vypnout tlačítkem na přístrojové desce. Systém se deaktivuje v případě nesprávného rozpoznání dopravních čar, které může být způsobené hustým sněžením nebo deštěm, poškozenými čarami nebo při detekci více čar. Detekce více čar nastává zpravidla při průjezdu stavbou, kde se nachází původní značení a dočasné značení. Je to jistá forma bezpečnosti, kdy systém raději nedetekuje nic, než aby řidiče informoval nesprávně.

#### **Daimler - Active lane keeping assist**

Systém je aktivní při rychlosti nad 60 km/h. V případě, že se řidič blíží k plné čáře bez upozornění směrovými světly, systém zlehka přibrzdí kola na protilehlé straně. V případě, že se řidič blíží k přerušované čáře bez upozornění směrovými světly, je na to upozorněn vibracemi ve volantu. V případě, že by mělo dojít ke kolizi, jsou přibržděna kola na protilehlé straně, čímž dojde k navrácení vozidla do jízdního pruhu. Systém využívá stereo kameru. Daimler je mimo jiné také první, kdo implementoval tento systém do nákladních automobilů.

#### **Ford - Lane Departure Warning**

Upozornění na vybočení je signalizováno vibracemi ve volantu a zobrazením na displeji. Úroveň vibrací je nastavitelná.

#### **Mazda- Lane Departure Warning**

Systém umožňuje detekci bílých i žlutých dopravních značení. Možnost vypnutí.

#### **Škoda - Lane assist**

Systém se aktivuje při rychlosti nad 65 km /h a je možné jej deaktivovat za pomoci tlačítka. Systém je opět omezen hustým sněžením, deštěm, špatně rozpoznatelnými čarami nebo špatně nasvícenou scénou.

#### **VW - Lane assist**

Systém je aktivní při rychlosti nad 60 km/h. Tento systém je aktivní i v případě, že se na vozovce nenachází dělící středová čára, ale pouze jen krajní. Systém opět může být vypnut tlačítkem.

## 2 DETEKCE JÍZDNÍCH PRUHŮ V OBRAZE

Metody pro detekci jízdních pruhů lze rozdělit do dvou skupin, a to na metody využívající a nevyužívající transformace obrazu. Transformace obrazu k potlačení perspektivního zobrazení může být také použita např. pro potlačení zkreslení objektivu.

### Perspektivní zobrazení:

Perspektivní zobrazení nasnímaného obrazu způsobuje, že vzdálenější objekty vypadají menší než objekty blízké. Dále také způsobuje, že rovnoběžné linie se směrem k horizontu opticky zúžují do jednoho bodu - tzv. úběžníku, to je právě případ jízdních pruhů.

Pro potlačení perspektivního zobrazení se používá metoda inverzního perspektivního zobrazení. Perspektivu je také možné potlačit za pomoci speciálních objektivů tzv. tilt-shift objektivů. Tyto objektivy jsou schopny posunout čočku relativně k senzoru, čímž se vhodně upraví zorný úhel. Dalšími objektivy potlačující perspektivní zobrazení jsou telecentrické objektivy používané především v průmyslu pro přesná měření za pomoci kamery.

### 2.1 Inverzní perspektivní mapování IPM [10],[14]

Nasnímaná scéna kamerou je v perspektivním zobrazení. Inverzní perspektivní mapování (*Inverse perspective mapping*), dále jen IPM, je metoda transformace obrazových souřadnic na tzv. souřadnice reálného světa, neboli v prostoru. Daný problém zobrazuje obrázek 2.1, kde X, Y, Z určují osy souřadnicového systému v prostoru. Souřadnice d, l, h jsou souřadnice polohy kamery v prostoru. Dále u,v reprezentují obrazový souřadnicový systém. Úkolem IPM je tedy transformace souřadnic u, v na souřadnice x, y, z. Úhel  $\gamma$  udává úhel mezi optickou osou kamery a osou y.  $\Theta$  představuje úhel mezi optickou osou kamery a osou z.  $2\alpha_u$  a  $2\alpha_v$  představují zorné pole kamery, a to v horizontálním a vertikálním směru. Pro IPM lze dle [14] sestavit rovnice 2.1 a 2.2.

$$x(u, v) = h * \cot\left(\frac{2\alpha_v}{R_v - 1} * v - \alpha_v + \Theta\right) * \sin\left(\frac{2\alpha_u}{R_u - 1} * u - \alpha_u + \gamma\right) + d \quad (2.1)$$

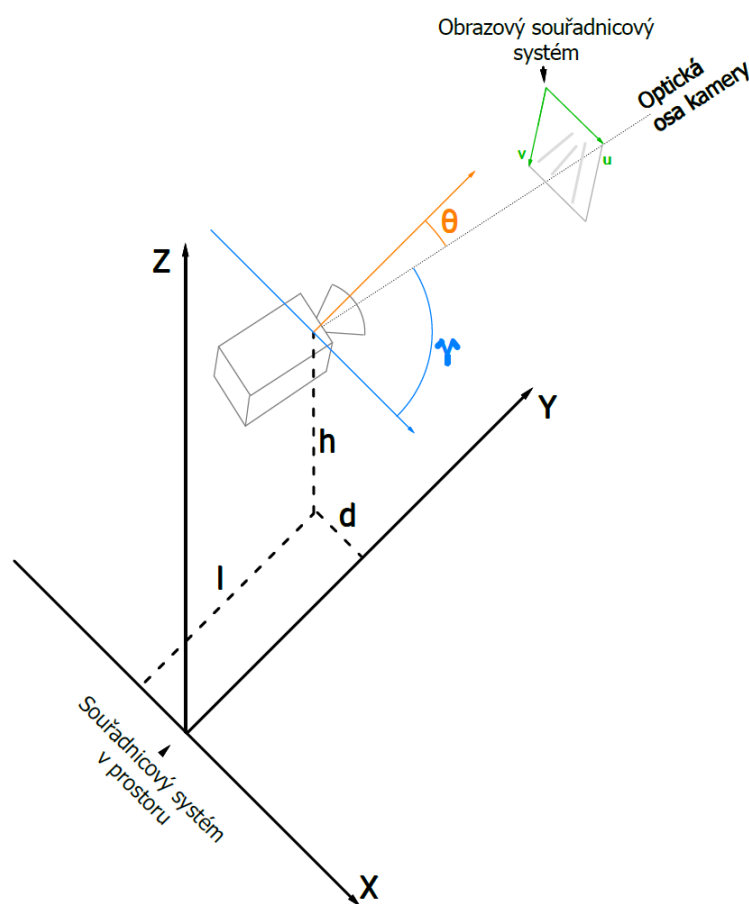
$$y(u, v) = h * \cot\left(\frac{2\alpha_v}{R_v - 1} * v - \alpha_v + \Theta\right) * \cos\left(\frac{2\alpha_u}{R_u - 1} * u - \alpha_u + \gamma\right) + l \quad (2.2)$$



Nicméně použitím těchto rovnic 2.1 a 2.2 vzniknou v transformovaném obraze prázdné pixely, protože se jedná o nelineární jednosměrnou transformaci. Proto se rovnice 2.1 a 2.2 používá pro získání rozsahu x a y a poté pracujeme s rovnicemi 2.3 a 2.4 [14]. V tomto případě lze zaručit, že ke každému pixelu ve výsledném obraze lze najít korespondující pixel v obraze původním.

$$v = \frac{(R_v - 1) * \left( \arctan \frac{h}{\sqrt{(x-d)^2 + (y-l)^2}} + \alpha_v - \Theta \right)}{2\alpha_v} \quad (2.3)$$

$$u = \frac{(R_u - 1) * \left( \arctan \frac{(x-d)}{(y-l)} + \alpha_u - \gamma \right)}{2\alpha_u} \quad (2.4)$$



Obr. 2.1: Ilustrační obrázek k inverznímu perspektivnímu mapování IPM

## 2.2 Top - View Transformation model

Metoda IPM vyžaduje znalost koeficientů, které se v praxi velmi špatně určují, případně se v dynamické scéně mohou měnit. Top-View Transformation Model

(TVTM) je v podstatě varianta IPM obsahující algoritmy pro automatické určení těchto parametrů. Metoda tedy nevyžaduje žádné interní ani externí parametry kamery a je schopna se tak adaptovat například při změně polohy kamery.

Úkolem metody je poskytnout pohled shora na snímanou scénu nebo oblast zájmu ze snímané scény jejím natočením. Matice rotací  $R_\omega$ ,  $R_\phi$ ,  $R_\kappa$  kolem kartézského souřadnicového systému pro TVTM jsou uvedeny v rovnici 2.5 [17]. Všechny rotace jsou ve směru hodinových ručiček.  $\Omega$  udává úhel otočení osy X,  $\Phi$  kolem osy Y a  $\kappa$  kolem osy Z. Pro rovnici  $R_\Omega$  jsou souřadnice osy X invariantní a osy Y a Z jsou pootočený o úhel  $\Omega$ . Podobně je tomu u matic  $R_\Phi, R_\kappa$ .

$$R_\omega = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\omega & \sin\omega & 0 \\ 0 & -\sin\omega & \cos\omega & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_\phi = \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_\kappa = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 & 0 \\ -\sin\kappa & \cos\kappa & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Vyjdeme - li z obrázku 2.1 a předpokladu, že úhel  $\gamma = 0$  (tedy, že kamera se bude dívat rovně před sebe), potom na základě této informace a matic rotací budou mít rovnice pro výpočet nových souřadnic dle 17 následující tvar:

$$x^* = H \frac{x \sin(\theta) + f \cos(\theta)}{-y \cos(\theta) + f \sin(\theta)}; y^* = H \frac{y \sin(\theta) + f \cos(\theta)}{-y \cos(\theta) + f \sin(\theta)} \quad (2.6)$$

- $x, y$  ... souřadnice původního obrazu,
- $x^*, y^*$  ... nové souřadnice pro transformaci obrazu,
- $H$  ... vzdálenost mezi kamerou a povrchem země,
- $f$  ... ohnisková vzdálenost objektivu kamery,
- $\theta$  ... úhel naklonění mezi optickou osou kamery a povrchem země.

Dle rovnic 2.6 mohou být hodnoty souřadnic menší nebo rovny nule. Z toho důvodu se zavádí konstanta  $d$  zajišťující, že souřadnice původního obrazu budou namapovány na souřadnice výsledného obrazu. Konstanta se vypočte následovně:

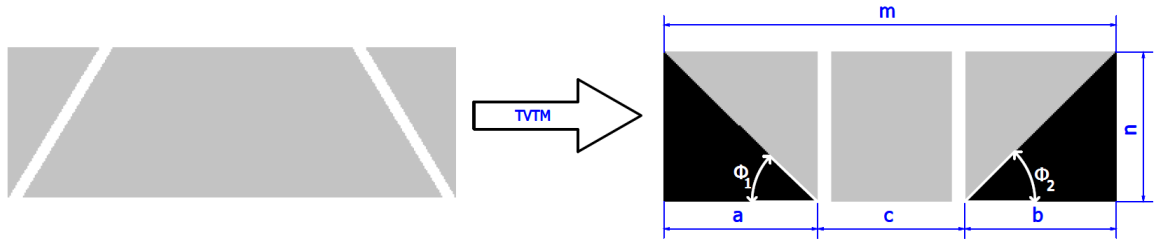
$$d = \frac{H \sin(\theta) + f \cos(\theta)}{f \sin(\theta) - \cos(\theta)} + 1 \quad (2.7)$$

rovnice pro výpočet nových souřadnic pak budou následujícího tvaru:

$$x^* = H \frac{x \sin(\theta) + f \cos(\theta)}{-y \cos(\theta) + f \sin(\theta)} + d; y^* = H \frac{y \sin(\theta) + f \cos(\theta)}{-y \cos(\theta) + f \sin(\theta)} + d \quad (2.8)$$

### 2.2.1 Estimace parametrů $H, f, \Theta$

Estimace je prováděna za pomoci inicializačních hodnot parametrů  $H, f$  a  $\Theta$ . Hodnoty parametrů je možné určit i bez inicializace, avšak za delší výpočetní dobu.



Obr. 2.2: Příklad TVTM, pro vysvětlení estimace parametrů  $H$ ,  $f$ ,  $\theta$

Na obrázku 2.2 vlevo je vstupní obraz před transformací, vpravo je výstupní obraz po transformaci. Obsah barevné oblasti ve výstupním obraze odpovídá obsahu barevné oblasti ve vstupním obraze. Rozměry snímku jsou  $m \times n$  pixelů, nechť  $a$ ,  $b$  a  $c$  jsou délky tří segmentů v obraze, aby platilo, že  $a = b$  musí zároveň platit, že  $\phi_1 = \phi_2$ .  $\phi_1$  a  $\phi_2$  [17] se vypočtou následovně:

$$\phi_1 = \cos^{-1} \frac{a}{\sqrt{n^2 + a^2}}, \phi_2 = \cos^{-1} \frac{b}{\sqrt{n^2 + b^2}} \quad (2.9)$$

Obrázky 2.3(B-F) zobrazují výsledky TVTM transformace pro různé hodnoty parametrů  $H$ ,  $f$  a  $\theta$ . Pro obrázek 2.3 B) platí, že  $a < b$ , to je způsobeno nízkými hodnotami parametrů  $H$  a  $\theta$ . U obrázku 2.3 C) je naopak  $a > b$ , to je způsobeno vysokými hodnotami parametrů  $H$  a  $\theta$ . Pro obrázek 2.3 D) platí, že  $a = b$  a  $a < c$ , tato nerovnost je způsobena velkou hodnotou parametru  $f$ . Naproti tomu obrázek 2.3 E) ilustruje situaci, kdy  $a = b$  a  $a > c$ , což je způsobeno nízkou hodnotou parametru  $f$ . Pro optimální zobrazení transformovaného snímku musí platit následující:

$$a = b = c = \frac{m}{3} \quad (2.10)$$

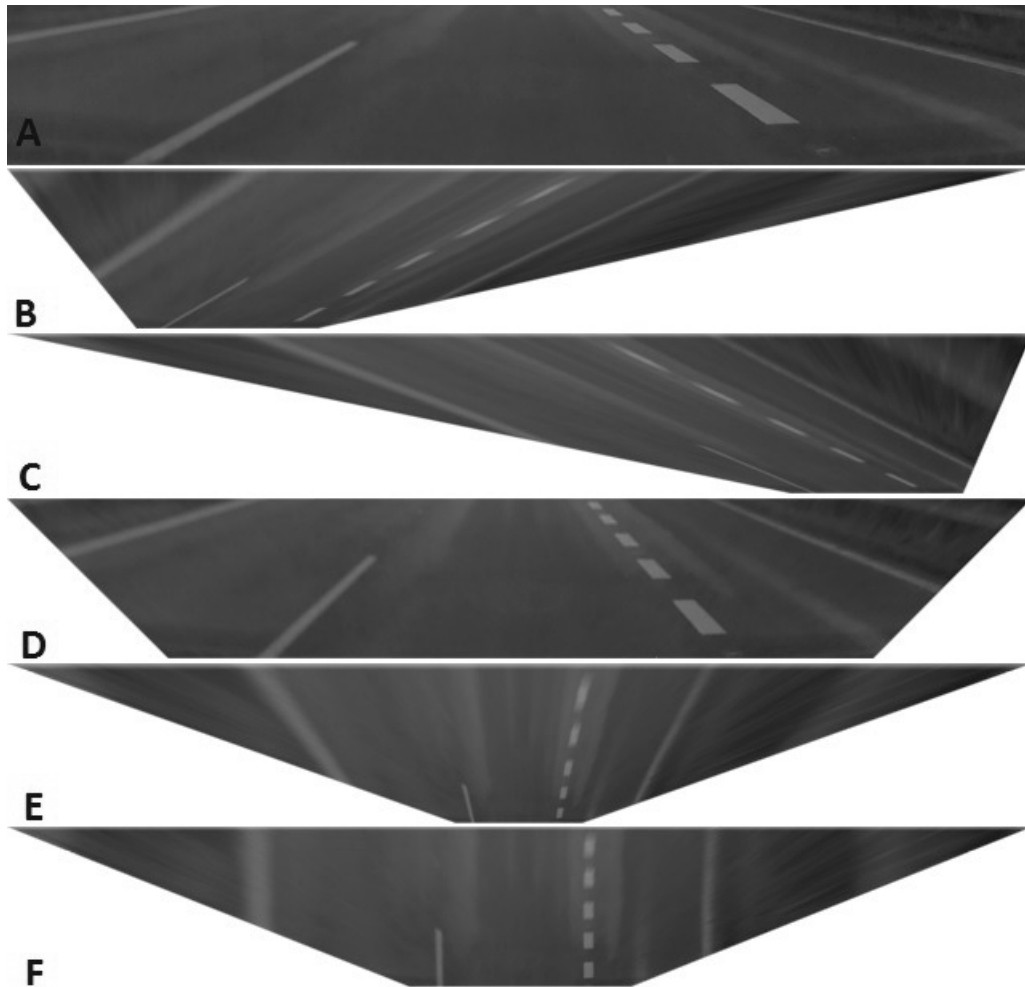
v reálném použití jsou pak povoleny malé odchylky a musí platit:

$$|a - b| < \epsilon; \frac{m}{3} - \omega < c < \frac{m}{3} + \omega \quad (2.11)$$

Transformované obrazy na pohled seshora jsou výhodné především v tom, že potlačují perspektivní zkreslení a umožňují pohled na vozovku z takových míst, do kterých by kameru nebylo možné umístit. Díky tomu je pak také usnadněna následná lokalizace a rozpoznání jízdních pruhů v obraze, který není zatížený perspektivním zkreslením.

### 2.2.2 Interpolace jasových hodnot

Při transformaci obrazu dochází k následujícímu problému. Celočíslným souřadnicím ve vstupním obraze odpovídají neceločíslné souřadnice ve výstupním obraze,



Obr. 2.3: Výsledky TVTM transformace pro různé hodnoty parametrů  $H$ ,  $f$ ,  $\theta$

tím ve výstupním obrazu vznikají prázdná místa, jejichž jasovou hodnotu je potřeba interpolovat. Mezi interpolační metody patří např. interpolace nejbližším sousedem, průměrnou hodnotou  $n$  nejbližších sousedů, lineární interpolace nebo kubická interpolace. Před dalším zpracováním transformovaného obrazu je tedy potřeba nejdříve provést interpolaci jasových hodnot.

### 2.2.3 Hranové filtry, vyhlazování obrazu

Za vhodných světelných podmínek v transformovaném obraze jsou jízdni pruhy reprezentovány formou bílých obdélníků na šedém podkladu. Pro detekci takovýchto objektů je tedy výhodné použití hranového detektoru. Hrana je místo v obraze, které vykazuje vysokou prostorovou frekvenci, tedy skokovou změnu jasové funkce. Každý obraz pořízený kamerou obsahuje určitou míru šumu. V některých případech se může stát, že určitá oblast šumu se může chovat jako hrana. Aby nedocházelo k

falešné detekci hran v obraze, je potřeba šum eliminovat.

K eliminaci šumu v obraze se používá diskretní konvoluce s vhodně zvolenou konvoluční maskou. Základní metodou pro vyhlazování a eliminaci šumu je průměrování. Konvoluční masky pro průměrování mohou být následující:

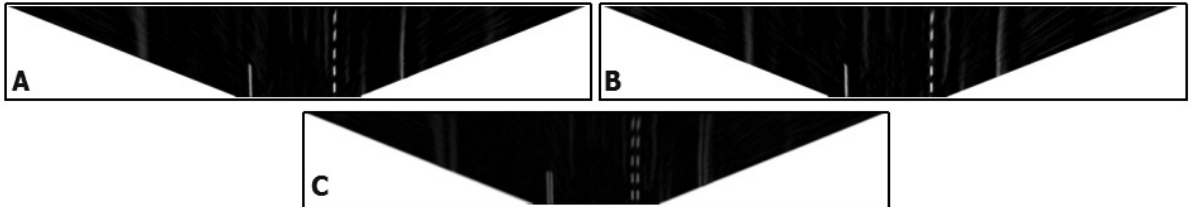
$$h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; h_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; h_3 = \frac{1}{10} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.12)$$

Masky  $h_2$  a  $h_3$  pak upravují váhy jednotlivých bodů masky. Filtraci šumu je také možné provést za pomoci Gaussova filtru. Pro vyhovující filtraci šumu se již provádí detekce hran opět za pomoci diskretní konvoluce.

Vycházíme - li z toho, že jízdní pruhy jsou v transformovaném obraze reprezentovány jako vertikální pruhy, pak jsou k jejich detekci použity následující filtry:

$$ef1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}; ef2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (2.13)$$

Filtr  $ef1$  lokalizuje hranu čáry z levé strany a filtr  $ef2$  zase z druhé strany. Výsledkem je tedy obraz vertikálních čar viz obrázek 2.4.



Obr. 2.4: Ukázka detekce jízdních pruhů za pomoci hranových filtrů: A - filtr  $ef1$ , B - filtr  $ef2$ , C - filtr  $ef1 +$  filtr  $ef2$

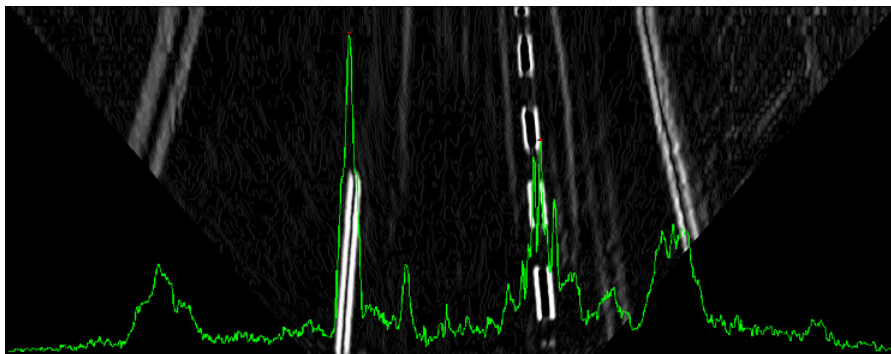
## 2.2.4 Lokalizace jízdních pruhů

Lokalizace jednotlivých jízdních pruhů v obraze je poté prováděna vypočtením horizontálního histogramu z vyhranovaného obrazu. Horizontální histogram se vypočte dle vzorce 2.14. V tomto histogramu jsou jízdní pruhy reprezentovány jako lokální maxima viz 2.5. Pozice lokálních maxim odpovídají středům jízdních pruhů v transformovaném obraze. Za pomoci rozšíření této hodnoty o offset na obě strany jsou poté pruhy vyřiznuty po celé své délce z transformovaného obrazu. Za předpokladu, že se kamera nachází uprostřed vozidla, jsou nalezena nejbližší lokální maxima na

obě strany od souřadnic středu snímku pořízeného kamerou. Vyříznuté snímky jsou následně vyprahovány.

$$h_h(i) = \sum_{j \in \langle 0; n \rangle} g(i, j) \quad (2.14)$$

- $i \in \langle 0; m \rangle$
- $m$  ... šířka obrazu v pixelech
- $n$  ... výška obrazu v pixelech
- $g(i, j)$  ... hodnota jasové funkce na souřadnicích  $i, j$



Obr. 2.5: Vyhranovaný snímek proložený horizontálním histogramem vypočteným dle vzorce 2.14.

### 2.2.5 Segmentace jízdnic pruhů za pomoci prahování

Segmentací se myslí oddělení objektů zájmu, v našem případě vodorovné dopravní značení, od pozadí, tedy cesty. Segmentace prahováním znamená, že pixelům příslušícím objektu zájmu je ve výsledném obraze přiřazena hodnota 1 a pozadí hodnota 0. Výstupem prahování je tedy binární obraz. Popsaný způsob prahování je prahování prosté. Existují i jiné druhy prahování například prahování s více prahy, částečné nebo adaptivní. Pro segmentaci jízdnic pruhů postačí prahování prosté.

U prahování je obecně problém určení hodnoty prahu. Tedy mez určující jaké jasové hodnoty přísluší objektu a jaké pozadí. Hodnota prahu může být pevná, určená experimentálně. Vzhledem k tomu, že snímaná scéna nemá statické osvětlení, není možné tento postup použít. Další metody určení prahu vychází z analýzy histogramu. Jednou z možností je využití metody iterativního nalezení prahu.

#### Metoda iterativního nalezení prahu [12]

Metoda pracuje tak, že v každé iteraci upřesňuje aktuální práh až do okamžiku stability, tedy že hodnota prahu se ve dvou po sobě jdoucích iteracích nezmění. Metoda využívá faktu, že aktuální práh od sebe dělí právě dvě množiny pixelů, tedy pixely

objektů a pozadí. V každé iteraci je stanovena průměrná hodnota pozadí  $E^i(bg)$ , průměrná hodnota objektů  $E^i(fg)$  a na základě těchto hodnot je stanoven práh další iterace jako aritmetický průměr.

$$T^{i+1} = \frac{E^i(bg) + E^i(fg)}{2} \quad (2.15)$$

$$E^i(bg) = \frac{h_w^c(T^i)}{h_c(T^i)} \quad (2.16)$$

$$E^i(fg) = \frac{h_w^c(255) - h_w^c(T^i)}{h_c(255) - h_c(T^i)} \quad (2.17)$$

Metoda iterativního nalezení prahu využívá histogram prostý  $h$ , vážený  $h^w$ , kumulativní  $h_c$  a kumulativní vážený  $h_w^c$ , ty jsou dle [12] dány vztahy:

$$h(q) = h(0), h(1), \dots, h(255) \quad (2.18)$$

$$h(q) = 0 * h(0), 1 * h(1), \dots, 255 * h(255) \quad (2.19)$$

$$h_c(q) = \sum_{p=0}^q h(p) = h_c(q-1) + h(q) \quad (2.20)$$

$$h_w^c = \sum_{p=0}^q p * h(p) = \sum_{p=0}^q h^w(p) = h_w^c(q-1) + h^w(q) \quad (2.21)$$

### Otsuova metoda nalezení prahu [20],[9],[21]

Otsuova metoda předpokládá, že v obraze se nachází dvě třídy pixelů (pixely pozadí a objektů), tedy že histogram obrazu je bimodální. Bimodální histogram je takový histogram, který obsahuje dvě zřetelně oddělitelná lokální maxima. Na základě histogramu pak metoda vypočítá optimální hranici oddělující třídy pixelů tak, aby odchylky těchto dvou tříd byly minimální.

$$\sigma_\omega^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad (2.22)$$

- $\omega_i$  ... váhy pravděpodobnosti dvou tříd oddělené prahem  $t$ ,
- $\sigma_B^2$  ... odchylky těchto tříd.

Problém minimalizování odchylek dvou tříd může být dle Otsua vyjádřeno jako maximalizování těchto odchylek.

$$\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (2.23)$$

kde  $\omega_1(t)$  se vypočte z histogramu  $h$  následovně:

$$\omega_1(t) = \sum_0^t h(i) \quad (2.24)$$

a střední hodnota třídy  $\mu_1(t)$  jako:

$$\mu_1(t) = \frac{[\sum_0^t h(i)x(i)]}{\omega_1} \quad (2.25)$$

kde  $x(i)$  je hodnota středu  $i$ -té složky histogramu. Obdobně lze vypočítat  $\omega_2(t)$  a  $\mu_2$  na pravé straně histogramu pro složky větší než  $t$ .

### Algoritmus

1. Spočítat histogram  $h$  a pravděpodobnost pro každou složku  $\omega_i$ .
2. Nastavení počátečních hodnot  $\omega_i(0)$  a  $\mu_i(0)$ .
3. Pro všechny hodnoty prahu  $t = 1 \dots 255$ .
  - Aktualizovat hodnoty  $\omega_i$  a  $\mu_i$ .
  - Vypočíst  $\sigma_b^2(t)$ .
4. Požadovaná hodnota prahu odpovídá maximu  $\sigma_b^2(t)$ .
5. Vypočíst dvě maxima (pro dvě maximální složky histogramu  $t_1$  a  $t_2$ ).  $\sigma_{b1}^2(t_1)$  je maximum a  $\sigma_{b1}^2(t_2)$  je větší nebo rovno maximum.
6. Práh je pak dán vztahem:

$$t = \frac{t_1 + t_2}{2} \quad (2.26)$$

## 2.3 Algoritmus CHEVP - Canny/Hough Estimation of Vanishing Points) [32]

Jedná se o algoritmus pro určení úběžníku, je odolný vůči šumu, stínům a jasovým změnám v pořízeném snímku. Je možné jej aplikovat na cesty s vyznačenými jízdními pruhy i bez nich. Tento algoritmus na rozdíl od předcházejícího nepoužívá k lokalizaci jízdních pruhů transformaci pořízeného snímku, vychází z toho, že jízdní pruhy jsou dvě paralelní čáry, které jsou v krátkém horizontálním pásmu přibližně rovnoběžné a na horizontu se zúžují do úběžníku.

### 2.3.1 Matematický model jízdních pruhů

Horizontální vzdálenost mezi dvěma body je dána vztahem  $D = (x_r - x_l)$ , kde  $x_r$  a  $x_l$  jsou souřadnice dvou korespondujících bodů  $P_l(x_l, y)$  a  $P_r(x_r, y)$  v souřadnicích



reálného světa. Po projekci bodů do obrazové roviny bude platit  $d = (c_r - c_l)$ ,  $p_l(c_l, r)$  a  $p_r(c_r, r)$ .

$$d = \frac{f^2 D(r - hz)}{H(f^2 + hz^2)} \quad (2.27)$$

- $f$  ... ohnisková vzdálenost objektivu,
- $H$  ... výška kamery nad zemí,
- $hz$  ... souřadnice úběžníku,
- $r$  ... vertikální souřadnice v obrazovém prostoru (řádek obrazu).

Vzdálenost  $d$  může být také dána vztahem  $d = k(r - hz)$ , kde

$$k = \frac{f^2 D}{H(f^2 + hz^2)} \quad (2.28)$$

Nechť  $L_s(c_s, r_s)$  jsou souřadnice středu jízdního pruhu, pak levá část jízdního pruhu bude  $L_l(c_l, r_l)$  a pravá  $L_r(c_r, r_r)$ , kde

$$c_l = c_s - \frac{1}{2}d = c_s - \frac{1}{2}k(r_l - hz); r_l = r_s \quad (2.29)$$

obdobně pravá strana:

$$c_r = c_s + \frac{1}{2}d = c_s + \frac{1}{2}k(r_r - hz); r_r = r_s \quad (2.30)$$

V následující části je popsáno určení parametru  $k$  i bez znalosti ohniskové vzdálenosti objektivu přímo z obrazových dat. Algoritmus CHEVP lze shrnout do následujících bodů:

1. Detekce hran v obrazu za použití Cannyho detektoru.
2. Detekce rovnoběžných hran za použití Houghovy transformace. V této části je obraz rozdělen na několik krátkých úseků v horizontálním směru, na které je postupně aplikována Houghova transformace.
3. Detekce úběžníku v každé sekci obrázku na základě spárovaných paralelních čar. Pozice úběžníku je průnik spárovaných přímek získaných z Houghovy transformace, je spočítána pro každou sekci obrázku zvlášť. Poloha řádku, ve kterém se nachází úběžníky z každé sekce, se pak nazývá úběžníková linie. Každému úběžníku v úběžníkové linii přísluší dvě přímky, které v příslušném úseku obrazu představují jízdní pruh.
4. Odhad středu jízdního pruhu z detekovaných čar. Jednotlivé sekce obrazu jsou procházeny postupně a jsou z nich vybírány čáry nejbližší středu. V případě, že sekce neobsahuje žádné čáry, je přeskočena. Následně je spojen úběžník zvolených přímek se středem mezi přímkami. Parametr  $k$  je pak vypočten:

$$k = \frac{c_r - c_l}{r_s - hz}$$

- $c_r$  ... souřadnice sloupce pravé čáry,
- $c_l$  ... souřadnice sloupce levé čáry,
- $r_s$  ... souřadnice řádku středu mezi přímkami,
- $hz$  ... horizontální souřadnice úběžníku.

Na základně parametru  $k$  je možné dopočítat souřadnice levé a pravé strany jízdního pruhu v obraze viz rovnice 2.29 a 2.30

### 3 POPIS SNÍMANÉ SCÉNY

Snímaná scéna je dynamická. Charakter scény určují měnící se podmínky, a to počasí, osvětlení scény světly vozidla nebo externími zdroji světla, jako jsou pouliční lampy. Vlivem počasí může dojít k přesvětlení scény nebo naopak k nedostatku světla při jízdě za tmy, či v mlze. Osvětlení scény lze do jisté míry ovlivnit vlastními světly vozidla za použití dálkových, tlumených nebo mlhových světel, v případě snímání scény za vozidlem zpátečním světlem.



Obr. 3.1: Ukázky snímané scény před vozidlem



Obr. 3.2: Ukázky snímané scény za vozidlem

#### 3.1 Značení jízdních pruhů

Následující informace vychází ze: Zásad pro vodorovné dopravní značení na pozemních komunikacích (technické podmínky TP133) vydaných Ministerstvem dopravy - obor pozemní komunikace [29]. Okraje jízdních pruhů jsou vyznačené tzv. vodíčovými čarami nacházejícími se na okrajích vozovky. Oddělení jízdních pruhů se provádí oddělovacími čarami vodorovného dopravního značení o šířce 0,12,5 nebo 25,0 cm. V případě, že je šířka vozovky mezi vodíčovými čarami menší než 6,0 m, tak se jízdní pruhy neznačí. Ve zvláštních případech je možné vyznačit jízdní pruh o minimální

šířce 2,75 m. Vyznačení jízdních pruhů o šířce menší než 2,75 m je možné pouze v oblasti stavby. Minimální šířka vyznačeného jízdního pruhu je tedy 3,0 m a maximální šířka je 3,75 m.

### **Podélné souvislé čáry**

Slouží k oddělení protisměrných jízdních pruhů a zákazu vjetí do protisměrného jízdního pruhu. Pro zdůraznění zákazu se používá dvojitě podélné souvislé čáry. Šířka čáry je 12,5 cm stejně jako mezera mezi čarami.

Podélná souvislá čáry se také používá pro oddělení odbočovacího nebo připojovacího pruhu od průběžného jízdního pruhu, oddělení vyhrazeného jízdního pruhu. Šířka čáry je 25,0 cm. Délka čáry je v případě odbočovacího pruhu 1,5 m stejně jako délka mezery. Délka čáry v případě připojovacího pruhu je 0,5 m a délka mezery je 1,5 m.

### **Podélné přerušované čáry**

Používají se opět pro oddělení jízdních pruhů tentokrát s povolením vjetí do protisměrného jízdního pruhu. Délka čáry je variabilní a závisí na druhu pozemní komunikace. V případě dálnice a silnice pro motorová vozidla je délka 6,0 m, jinak je délka 3,0 m. Délka mezer je dvojnásobná délka čáry, šířka čáry je 12,5 cm.

Podélné přerušované čáry se také používají pro oddělení odbočovacího nebo připojovacího pruhu od průběžného jízdního pruhu, vyznačení okraje jízdního pruhu ve směru hlavní pozemní komunikace, šířka čáry je 25,0 cm.

## 4 HARDWARE

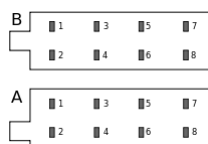
### 4.1 Blokové schéma

Blokové schéma propojení jednotky je uvedeno v příloze 4B. K automobilu je jednotka připojena pomocí konektoru XS1 (ISO 10487 (viz obr. 4.1)), ten zajišťuje jednotce přístup k 12V stálým a spínaným ze spínací skříňky a kostře automobilu (GND). Hlavní řídicí jednotkou je Raspberry Pi 2 (dále jen RPi) (A1) model B+, ke kterému jsou připojeny ostatní periférie. Práce byla zpočátku vytvářena pro model Raspberry Pi verze 1. V průběhu práce byl uveden na trh model Raspberry Pi verze 2. Novější verze je výkonnější, viz kapitola 4.3 a mechanicky kompatibilní s verzí 1 modelem B+.

Displej s odporovou dotykovou vrstvou (A3) má svoji řídicí jednotku (A2), ke které je připojena deska s tlačítky pro ovládání displeje. Deska umožňuje stejné nastavení jako tlačítka na klasickém LCD displeji, tj. nastavení jasu, kontrastu, volba vstupu, atd. Jelikož je displej schopný si uchovat toto nastavení i po odpojení od napájení, byl nastaven a deska byla uschována do jednotky. Čtyř-vodičová odporová dotyková vrstva displeje je připojena k RPi za pomoci převodníku (A4) přes USB.

Původně byla pro snímání vozovky zvolena web-kamera Logitech C310 (CAM2). Tato kamera byla zvolena s ohledem na dostupnost a podporu operačního systému (seznam zařízení, která jsou na RPi ověřena lze nalézt na [elinux.org](http://elinux.org)<sup>1</sup>). Vzhledem k delší době přístupu ke kameře přes USB byla vyměněna za kamerový modul RPi (CAM1). Přístup k tomuto modulu je podstatně rychlejší než ke kameře připojené přes USB. CAM2 byla přemístěna do zadní části vozidla a je využita ke snímání scény za vozidlem.

Pro připojení ostatních periférií k RPi byla vytvořen modul (A2). Jednotka dále obsahuje stabilizátor napětí 5V (TF1) pro napájení RPi a jeho periférií. Dále pak ventilátor (M1) pro odvod tepla, které je nejvíce generováno displejem a zčásti pak také stabilizátorem.



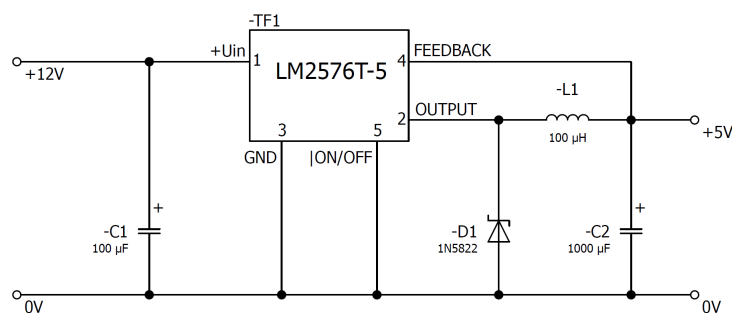
Obr. 4.1: Konektor ISO 10487

<sup>1</sup>[http://elinux.org/RPi\\_VerifiedPeripherals](http://elinux.org/RPi_VerifiedPeripherals)

## 4.2 Napájení

Jednotka je napájena 12V DC z automobilu. Přívod napětí je realizován za pomoci ISO konektoru 10487 (viz obr. 4.1), kde pro napájení slouží piny: A.4 - trvale pod napětím 12V DC (vodič červený), A.7 - spínané napětí 12V DC (vodič žlutý), A.8 - kostra 0V (vodič černý). Stálé napětí na pinu A.4 je jištěno pojistkou FU1.

Napětí 12V je použito pro napájení řídicí jednotky displeje, která obsahuje vlastní stabilizátor napětí a pro stabilizátor napětí 5V DC určený k napájení Raspberry Pi a jeho periférií. Jako napěťový stabilizátor byl zvolen integrovaný obvod *LM2576T-5*, jeho vstupní napětí je 7 - 40V DC. Výstupní napětí stabilizátoru je 5V s výstupním proudem maximálně 3A. Tato hodnota je dostačující pro napájení Raspberry Pi, kamerového modulu, webkamery, dotykové vrstvy na displeji, případně jiné vstupně výstupní periférie v podobě klávesnice nebo myši. Parametry stabilizátoru jsou uvedeny v tabulce 4.1. Obvod je zapojen dle doporučení výrobce (viz obr. 4.2). Hodnota odebíraného proudu při plném zatížení jednotky je 0,9 A.



Obr. 4.2: Schema zapojení stabilizátoru s *LM2576T-5*

Tab. 4.1: Parametry stabilizátoru *LM2576T-5*

Vstupní napětí	$U_{in} = 40 \text{ V}$
Výstupní napětí	$U_{out} = 5 \text{ V}$
Max. odebíraný proud	$I_{out} = 3 \text{ A}$
Provozní teplota	$-40 \text{ }^\circ\text{C} - 125 \text{ }^\circ\text{C}$
Účinnost ( při $U_{in} = 12 \text{ V}$ , $I_{out} = 3 \text{ A}$ )	$\eta = 77 \%$
Vnitřní tepelný odpor	$R_{\theta} = 2 \text{ }^\circ\text{C/W}$

## 4.3 Raspberry Pi

Raspberry Pi je malý počítač o velikosti kreditní karty (85,6 mm x 56,0 mm x 21,0 mm). Byl vyvinut *Raspberry Pi Foundation* původně k vyučování základům programování na školách. Díky své všestrannosti se ale rozšířil i mezi širokou veřejnost, a to poměrně rychlým tempem.

RPi je možné využít jako plnohodnotný počítač po připojení ostatních periférií. Napájen je 5V přes MicroUSB konektor. Displej je k němu možné připojit přes HDMI konektor nebo kompozitní RCA. Pro připojení LCD panelu obsahuje RPi DSI (Display Serial Interface) konektor. Problémem tohoto rozhraní je, že k němu existuje pouze velmi málo periférií, které by s ním byly schopny komunikovat po pouhém připojení bez složitých softwarových zákroků. Audio výstup je ve formě 3,5 mm jacku případně přes HDMI. Dále RPi obsahuje ethernet konektor (ne všechny modely) pro připojení do sítě a USB konektor pro připojení různých periférií komunikujících po této sběrnici. To, co dělá RPi výjimečným oproti standardnímu PC, je kromě jeho velikosti a spotřebě bezesporu GPIO (General Purpose Input/Output) konektor. Ten obsahuje kromě vstupů a výstupů také sběrnice UART, I2C a SPI. Připojení kamery k RPi je možné přes USB nebo za pomoci CSI (Camera Serial Interface), k tomuto rozhraní je momentálně možné připojit pouze Raspberry Pi kamerový modul. RPi v sobě nemá zabudovaný žádný pevný disk či flash paměť pro operační systém a data. Místo toho má slot pro SD kartu (novější modely pro microSD), ze které je spouštěn operační systém a zároveň také slouží pro uchování programů a dat.

Momentálně je RPi dostupné v několika modelech, a to ve verzi 1 A, B, A+, B+ a verzi 2 model B+. Přičemž modely A+ a B+ jsou novějšími ekvivalenty původních A a B. Verze 2 model B+ je aktuálně nejnovější a nejvýkonnější ze všech uvedených modelů. Porovnání verze 1 a verze 2 (modely B+) je uvedeno v tabulce 4.2. Jak již bylo zmíněno, práce byla zpočátku vytvářena pro verzi 1 B+, díky zpětné kompatibilitě jak mechanické tak softwarové nebyl problém s přechodem na vyšší verzi 2 B+.

### 4.3.1 GPIO

GPIO neboli General Purpose Input / Output je konektor obsahující piny sloužící RPi pro komunikaci s okolním světem na nižší úrovni. V práci je tento konektor použit především ke snímání stavu spínací skříňky, připojení snímačů teploty a pro rozšíření funkcí asistenční jednotky. Přes tento konektor je k RPi připojen rozšiřující modul. Model B+ má na rozdíl od svých předchůdců 40-ti pinový GPIO konektor namísto 26-ti pinového, avšak prvních 26 pinů je kompatibilních se starší verzí.

Konektor u starších modelů obsahuje 26 pinů, z toho je 8 vstupně výstupních, další jsou určeny pro sběrnice UART, I2C a SPI. Na zbylých pinech je napájení 3,3 V a 5V a také GND. Doplněných 14 pinů obsahuje 9 vstupně výstupních, 2 jsou určeny pro přístup k EEPROM paměti a zbylé 3 piny jsou GND. Rozložení konektoru je uvedeno v tabulce 4.3.

**Pro práci s GPIO konektorem je potřeba dát pozor na několik věcí:**

1. Maximální napětí, které je možné na pin připojit, je 3,3 V.
2. Maximální proud, který je RPi schopno dodat na výstup, je 8 mA.
3. Pro komunikaci po sběrnicích, které využívají jiné napěťové úrovně než 3,3 V, je třeba tyto úrovně oddělit.

Při nedodržení těchto pravidel dojde ke zničení či poškození RPi.

Tab. 4.2: Srovnání Raspberry Pi 1 B+ a Raspberry Pi 2 B+

	Pi 1 B+	Pi 2 B+
SOC	BroadCom BCM2835	BroadCom BCM2836
CPU	ARM1176JZF-S	ARM CortexA-7
Počet Jder	1	4
Frekvence CPU	700 MHz	900 MHz
GPU	Broadcom VideoCore IV @ 250 MHz	
SDRAM	512 MB	1 GB
USB 2.0	4	4
Video výstup	HDMI, TRRS	
Video vstup	15-pin MIPI camera interface (CSI), USB	
Audio výstup	HDMI, 3,5 mm jack	
Datové uložení	MicroSD	
Ethernet	10/100 Mbit/s	
Napájecí napětí	5V DC přes MicroUSB nebo GPIO	
Proudový odběr	600 mA	800 mA
Rozměry	85,6 mm x 56,5 mm	

## 4.4 Displej

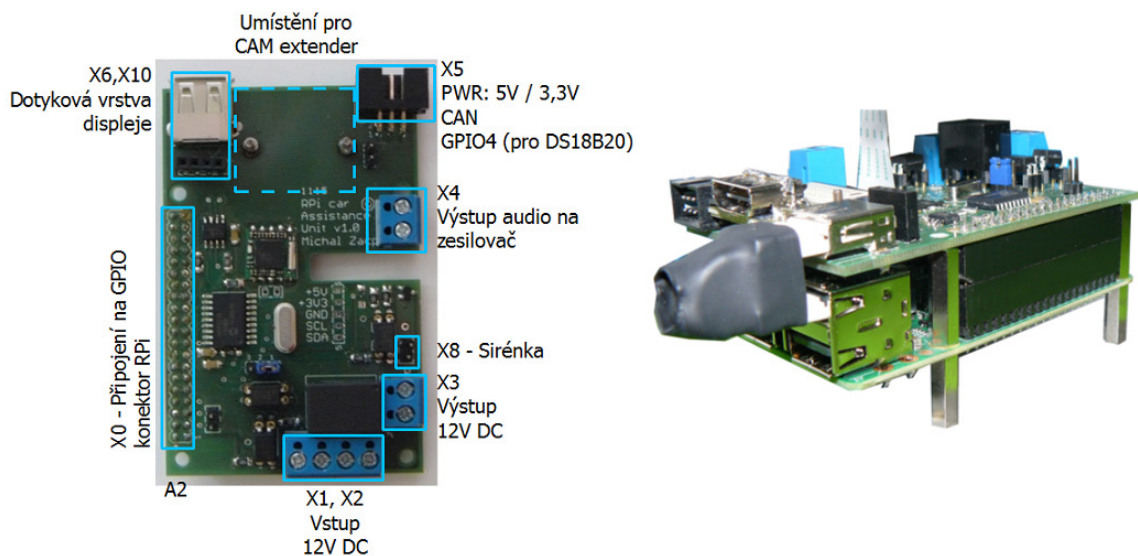
Při výběru displeje byl kladen požadavek na to, aby byl s dotykovou vrstvou (z důvodu snadného ovládání jednotky) a zároveň připojitelný nejlépe přes HDMI. Vzhledem k zabudování do panelu o velikosti 2DIN autorádia byl také velikostně omezen na maximálně 7".



Tab. 4.3: GPIO konektor Raspberry pi

GPIO#	Funkce	pin#	pin#	Funkce	GPIO#
-	+3V3	1	2	+5V	-
GPIO2	SDA1 (I2C)	3	4	+5V	-
GPIO3	SCL1 (I2C)	5	6	GND	-
GPIO4	GCLK	7	8	TXD0 (UART)	GPIO14
-	GND	9	10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11	12	GEN1	GPIO18
GPIO27	GEN2	13	14	GND	-
GPIO22	GEN3	15	16	GEN4	GPIO23
-	+3V3	17	18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19	20	GND	-
GPIO9	MISO (SPI)	21	22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23	24	CE0_N (SPI)	GPIO8
-	GND	25	26	CE1_N (SPI)	GPIO7
(Modely A a B končí pinem #26)					
EEPROM	ID_SD	27	28	ID_SC	EEPROM
GPIO5	-	29	30	GND	-
GPIO6	-	31	32	-	GPIO12
GPIO13	-	33	34	GND	-
GPIO19	-	35	36	-	GPIO16
GPIO26	-	37	38	-	GPIO20
-	GND	39	40	-	GPIO21

Z dostupných displejů bylo uvažováno nad 2,8" TFT displejem s rozlišením 320x480 pixelů ( PiTFT - Assembled 320x240 2.8"TFT+Touchscreen ). Tento displej se k RPi připojuje za pomoci GPIO pinů a je dodáván jako celý modul, který se pouze nasune na RPi. Vzhledem k malým rozměrům byl nakonec zvolen 7" LCD displej AT070TN92 s dotykovou vrstvou. Tento displej je propojen se svou řídicí jednotkou přes flex kabel a k RPi je připojen přes HDMI. Balení displeje dále obsahuje i převodník čtyř-vodičové dotykové vrstvy na USB a desku s tlačítky pro ovládání nastavení displeje. Displej díky své řídicí jednotce umožňuje také připojení přes VGA či RCA.



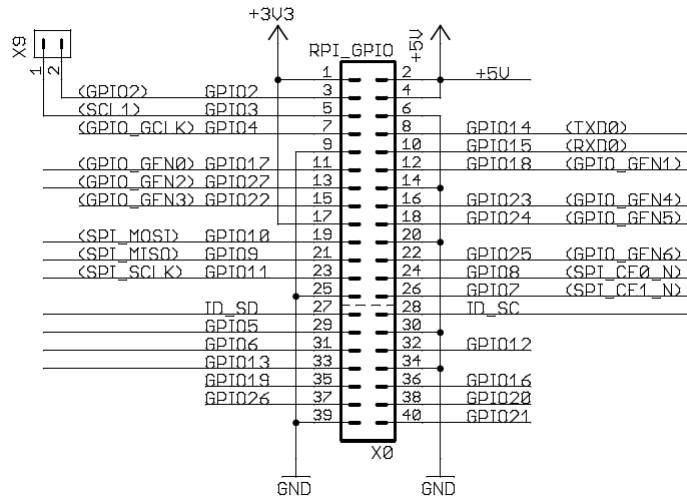
Obr. 4.3: Popis konektorů rozšiřujícího modulu a jeho umístění na Raspberry Pi s USB propojkou

## 4.5 Rozšiřující modul pro periferie jednotky

Modul byl vytvořen z důvodu propojení RPi a vozidla, a také pro připojení ostatních periférií. Rozměry včetně montážních otvorů vytvořené desky plošných spojů jsou shodné s RPi. Připojení modulu k RPi je realizováno nasunutím na jeho GPIO konektor. Modul je rozdělen na část zajišťující napájení jednotky, připojení kamery, připojení dotykové vrstvy displeje, akustické signálky a do budoucna je osazen SPI/-CAN převodníkem a FM rádiem. Jedná se pouze o přípravu a tyto obvody budou v rámci diplomové práce implementovány pouze hardwarově. Na obrázku 4.4 je znázorněno zapojení konektoru RPi. Názvy signálů korespondují se signály v dalších kapitolách. Konektor X9 slouží pro připojení periferních zařízení k I2C sběrnici. Celkové schéma zapojení je uvedeno v příloze 4C.

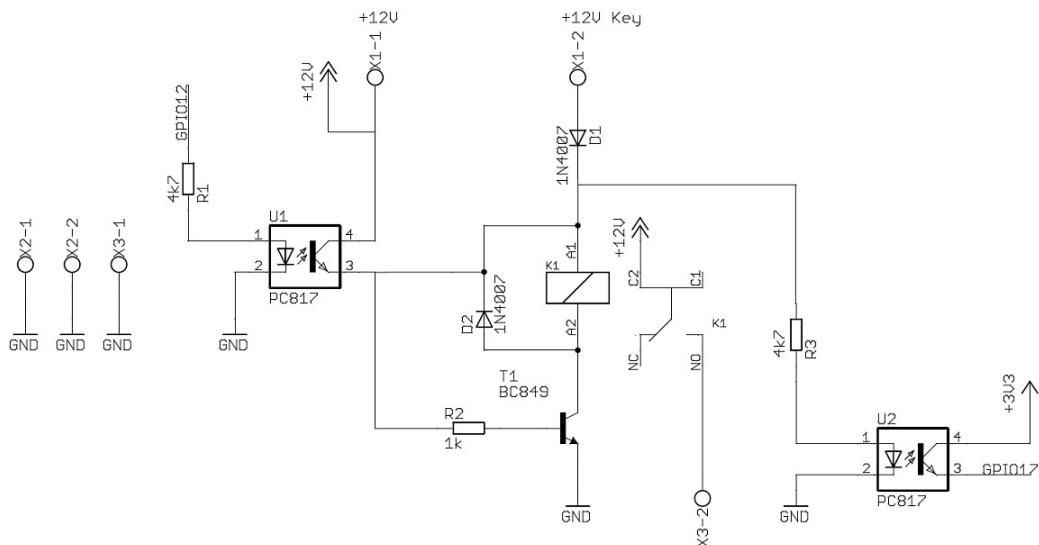
### 4.5.1 Napájení asistenční jednotky

Schéma zapojení této části je na obrázku 4.5. Ke svorce X1-1 je připojeno stálých 12V, ty jsou přiváděny do desky přes pojistku jednotky FU1. Na svorce X1-2 je spínané napětí spínací skříňky vozidla. To je určeno k sepnutí relé K1, které přivede stálých 12V do jednotky. Konkrétně na výstupní svorky X3-1, X3-2, tedy ke stabilizátoru TF1, displeji A3 a ventilátoru M1. Toto napětí zůstane stále připojené i po vytažení klíčku ze spínací skříňky, protože relé je udržováno stále zapnuté za pomoci výstupu GPIO12, který je sepnut okamžitě po startu jednotky. Protože RPi



Obr. 4.4: Propojovací konektor Raspberry Pi - rozšiřující deska

je schopné dodat na výstup maximálně 8 mA, je tento výstup oddělen za pomoci optočlenu U1. Aby nedošlo k vybití baterie po opomenutí vypnutí jednotky, je signál ze spínací skříňky přiveden na vstup GPIO17 a opět galvanicky oddělen přes optočlen U2. Tím je kontrolována přítomnost klíčku v zapalování. Po vypnutí RPi je výstup GPIO12 v log. 0 a je-li současně i klíček vytažen ze zapalování, relé K1 se rozepne a zamezí tak přívodu proudu do jednotky.



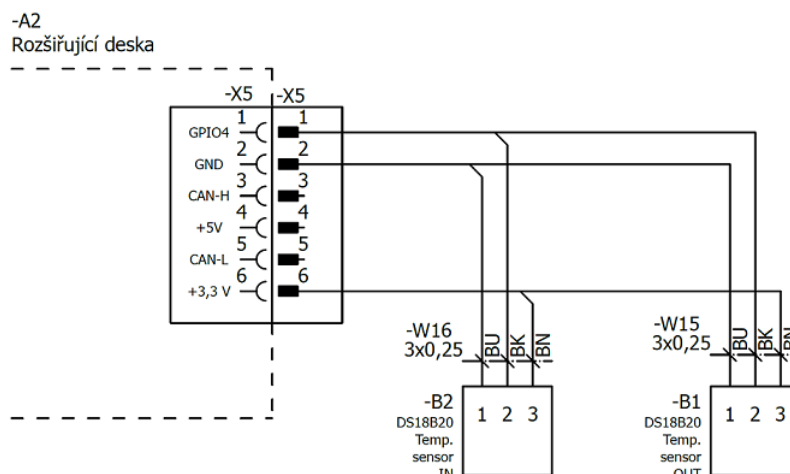
Obr. 4.5: Schéma zapojení rozšiřující desky - napájecí část

## 4.5.2 Informačně komunikační

Do této části spadá: konektor X6 pro připojení převodníku dotykové vrstvy displeje A5. Protože RPi má všechny USB konektory vyvedeny na zadní část jednotky, musela být vytvořena propojka zajišťující propojení převodníku s USB konektorem RPi bez nutnosti vedení kabelů mimo jednotku. LED dioda LED1 slouží indikaci napájecího napětí +5V. Konektor X5 je určen pro připojení ke sběrnici CAN a připojení snímačů teploty.

### Snímače teploty DS18B20

DS18B20 jsou digitální snímače teploty měřící v rozsahu od  $-55^{\circ}\text{C}$  do  $+125^{\circ}\text{C}$  s odchylkou  $0,5^{\circ}\text{C}$ . S nadřazeným mikroprocesorem komunikují po jedno-vodičové sběrnici 1-Wire. Každý snímač obsahuje unikátní 64-bitový kód, díky kterému může být připojeno více snímačů na jeden vodič a každý z nich je rozpoznán. Snímače jsou napájeny 3,3V. Připojení snímačů k jednotce je uvedeno na obrázku 4.6. Mezi napájecím napětím a datovým vodičem musí být vřazen odpor  $4,7\text{ k}\Omega$ , ten je již součástí desky. Teplota může být ze snímače vyčítána s rozlišením 9 nebo 12 bitů.



Obr. 4.6: Zapojení snímačů DS18B20

### Akustická signalizace

Sírénka H1 je určena ke zvukové signalizaci. Je připojena ke konektoru X8. Za pomoci propojky X7 je možné určit napájecí napětí, a to 5V nebo 12V. Je ovládána pomocí výstupu GPIO27 galvanicky odděleného optočlenem U3 a tranzistorového spínače.

### 4.5.3 Převodník SPI/CAN

Pro plánované připojení jednotky na CAN sběrnici vozidla byla deska vybavena SPI/CAN převodníkem skládajícím se z obvodů MCP2515 a MCP2551. Zapojení obvodů je uvedeno v příloze 4C. MCP2515 je samostatný SPI / CAN převodník, jehož funkcionality jsou integrovány v linuxovém jádře. Je napájen 3,3 V z RPi, zatímco obvod MCP2551 je napájen 5V. Aby nedošlo k poškození prvního obvodu, při komunikaci mezi sebou slouží napěťový dělič složený z odporů R9 a R11. Ten zajišťuje převod logických úrovní 0-5V na 0-3,3V. MCP2551 slouží jako rozhraní mezi fyzickou vrstvou sběrnice a komunikujícím zařízením.

### 4.5.4 FM Radio přijímač

Vytvořená jednotka je určena k nahrazení autorádia, aby nedošlo k úplnému odstranění této funkce z vozidla. Modul byl dovybaven obvodem TEA5767, který je určený pro příjem FM radiového signálu. Zapojení tohoto obvodu je uvedeno opět v příloze 4C. Jako přijímací anténu lze s výhodou použít anténu ve vozidle. Výstup obvodu bude připojen přes zesilovač a konektor XS1 na reproduktory ve vozidle. Obvod komunikuje po I2C sběrnici.

## 4.6 Popis snímací aparatury

Snímací aparatura pro monitorování dopravní situace před vozidlem se skládá z několika částí, které dohromady tvoří jeden celek v podobě kamery (viz obr. 4.10).

### 4.6.1 Raspberry Pi kamerový modul

Jedná se o kamerový modul přímo určený pro RPi, ke kterému se připojuje přes CSI konektor za pomoci 15 žilového plochého kabelu. Modul o rozměrech 25 x 24 mm obsahuje CMOS senzor OmniVision OV5647. Parametry tohoto senzoru jsou uvedeny v tabulce 4.4.

Modul již obsahuje mini objektiv s ohniskovou vzdáleností 3,60 mm, vertikálním úhlem záběru 41,41 ° a horizontálním úhlem záběru 53,50°. Objektiv je dle Raspberry Pi Foundation [24] pevně zaostřen na vzdálenost od 1 m do nekonečna.

### 4.6.2 Objektiv

Požadavky na objektiv byly následující: objektiv musí být na velikost čipu alespoň 1/4", vzhledem k použitému držáku objektivu musí být připojovací závit M12. Z kapitoly 3 víme, že je potřeba snímat vozovku o šířce alespoň 4 m, a to ve vzdálenosti

Tab. 4.4: Parametry CMOS senzoru OmniVision OV5647, použitého v RPi kamerovém modulu. [18]

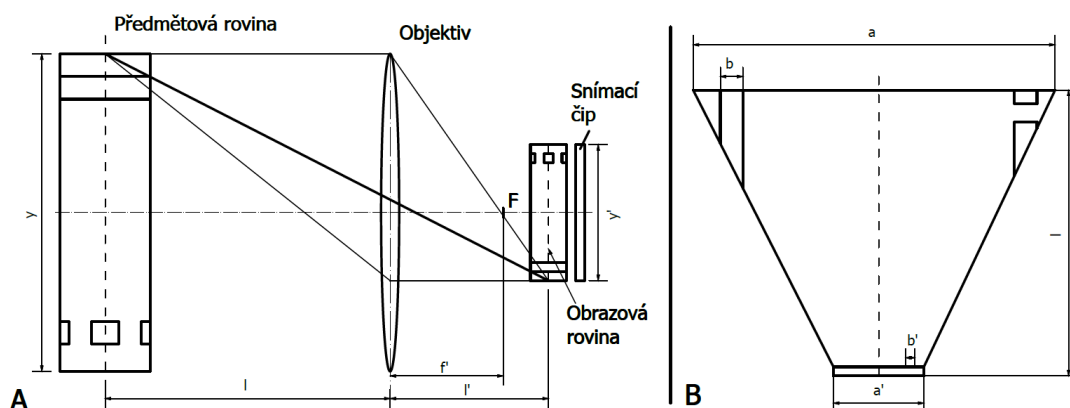
Velikost senzoru	3.76 x 2.74 mm
Velikost pixelu	1.4 μm x 1.4 μm
Optická velikost čipu	1/4"
Max. rozlišení	2592 x 1944 px (5 MPx)
Uspořádání obrazových bodů	Bayerova mřížka
Rozhraní	MIPI (Mobile Industry Processor Interface)

5 m od kamery. V takovém případě bylo nutné zvolit objektiv s malou ohniskovou vzdáleností. Minimální šířce podélné čáry jízdního pruhu by mělo odpovídat alespoň 10 px v pořizovaném snímku. Skutečná minimální šířka je 12,5 cm. Ohnisková vzdálenost objektivu se vypočte následovně:

$$f = \frac{y' * l}{y + y'} = \frac{3,76 * 5 * 10^3}{4 * 10^3 + 3,76} = 4,7mm \quad (4.1)$$

- $f$  ... ohnisková vzdálenost objektivu [mm]
- $l$  ... vzdálenost předmětu od kamery [mm]
- $y$  ... velikost snímaného předmětu [mm]
- $y'$  ... velikost snímacího čipu [mm]

vše je znázorněno v obrázku 4.7 A. Z řady dostupných objektivů splňujících dané parametry byly vybrány dva, jejichž parametry jsou uvedeny v tabulce 4.5. Po řadě testů byl nakonec zvolen objektiv 1 s proměnnou ohniskovou vzdáleností v rozmezí 2,6 - 6,0 mm. Tento objektiv má lepší světelnost, což se projeví zejména při jízdě ve tmě. Objektivy se dále liší v ohniskové vzdálenosti. Protože není požadavek ostření



Obr. 4.7: Doplnující obrázek k výpočtu ohniska (A) a rozlišení počtu px na šířku čáry (B)

na velké vzdálenosti, je plně dostačující objektiv s menším rozsahem (to se navíc pozitivně projevilo i v ceně tohoto objektivu). Při šířce záběru 4 m ve vzdálenosti 5 m odpovídá nejmenší šířce čáry zhruba 84 px. Vzhledem k tomu, že snímky jsou z důvodu rychlosti a postačující přesnosti pořizovány s menším rozlišením 640 x 480 px, kterému dle [15] odpovídá bining 4 x 4, bude výsledná nejmenší šířka čáry 21 px. Tím bylo dosaženo dokonce lepšího rozlišení, než bylo původně stanoveno. Postup výpočtu je následující:

Nejprve je vypočteno, kolik px na čipu odpovídá 12,5 cm při záběru o šířce 4 m.

$$b' = \left( \frac{b}{a} * a' \right) / p = \left( \frac{125,00}{4000,00} * 3,76 \right) / 1,40 * 10^{-3} = 83,93 \doteq 84px \quad (4.2)$$

- a ... ohnisková vzdálenost objektivu [mm],
- a' ... vzdálenost předmětu od kamery [px],
- b ... velikost snímaného předmětu [mm],
- b' ... velikost snímacího čipu [px],
- p' ... velikost pixelu snímacího čipu [px].

Bining 4 x 4 znamená, že jeden obrazový bod je složen ze čtyř skutečných pixelů na snímacím čipu. Tedy skutečný počet pixelů připadající na šířku čáry bude:

$$b'^* = \frac{b'}{Binning} = \frac{84}{4} = 21px \quad (4.3)$$

- b'\* ... ohnisková vzdálenost objektivu [mm]
- Binning ... Binning v daném směru (v tomto případě horizontálním) [-]

viz obrázek 4.7 B.

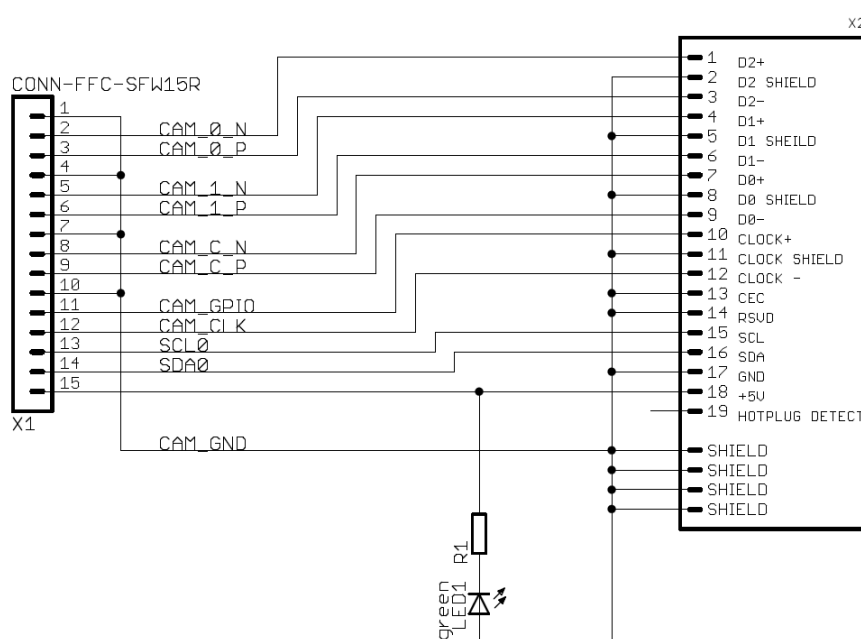
Tab. 4.5: Parametry vhodných objektivů

Parametry	Objektiv 1	Objektiv 2
Ohnisková vzdálenost	2,6 - 6,0 mm	2,8 - 12,0 mm
Obrazový formát	1/3"	1/3"
Připojovací závit	M12	M12
Clona	pevná	pevná
Světelnost	f/1.6	f/1.8
Horizontální úhel záběru	48,3 - 115,1 °	28,7 - 93,0 °

### 4.6.3 Cam extender

RPi kamerový modul se propojuje za pomoci plochého 15 žilového kabelu. Tyto kabely jsou vyráběny v maximální délce 60 cm a nejsou nijak odolné proti mechanickému poškození. Vzdálenost asistenční jednotky od kamery je přibližně 2,5 m,

pokud nemá být kabel v zorném poli řidiče. Z uvedených skutečností vyplývá, že plochý kabel není možné použít. Pro propojení asistenční jednotky a kamery bylo tedy potřeba vytvořit převodníky, které převedou signály z plochého kabelu od asistenční jednotky na propojovací kabel a zpět na plochý kabel ke kamerovému modulu. Aby odpadla výroba propojovacího kabelu, byl zvolen HDMI kabel, který obsahuje dostatečný počet vodičů a je určen pro vysokorychlostní přenos dat. Schéma převodníku je uvedeno na obrázku 4.8. Je to plošný spoj o stejné velikosti jako RPi kamerový modul, tedy 25 x 24 mm a se stejným rozmístěním otvorů. Toto rozvržení umožňuje případné připojení k modulu z druhé strany za pomoci distančních sloupků. Převodník je dále vybaven LED diodou pro indikaci napájecího napětí.



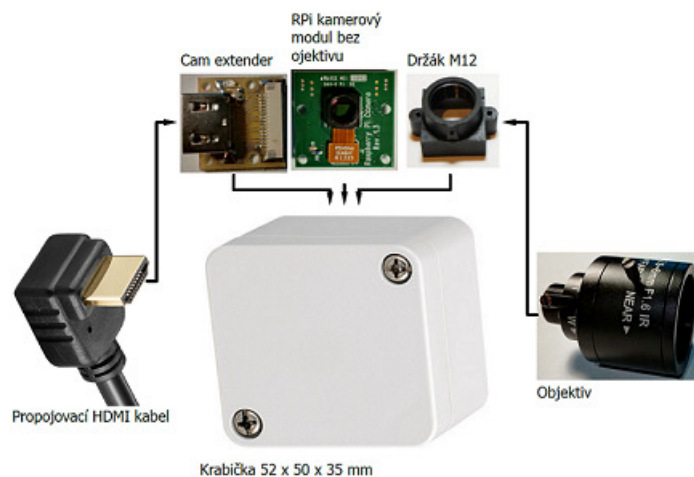
Obr. 4.8: Schéma zapojení převodníku Cam extender

#### 4.6.4 Jednotka pro monitorování dopravní situace před vozidlem

Objektiv je ke kamerovému modulu připojen pomocí M12 připojovacího závitu. Modul takovýmto závitem není vybaven a je potřeba na něj umístit příslušný držák. Před tím je potřeba opatrně odšroubovat původní mini objektiv. Všechny zmíněné komponenty, kromě objektivu, byly umístěny do malé krabičky o rozměrech 50 x 52 x 35 mm, která byla z původní bílé barvy přestříkána na černou.

Vytvořená snímací aparatura je přichycena pod zpětné zrcátko za pomoci přísavky (viz obr. 4.10).





Obr. 4.9: Složení snímací aparatury



Obr. 4.10: Kamera pro monitorování dopravní situace před vozidlem

#### 4.6.5 Web kamera - monitorování dopravní situace za vozidlem

Další kamerou použitou v této práci je USB webkamera Logitech C310. Tato kamera byla původně určena jako primární kamera pro snímání vozovky v přední části vozidla. Vzhledem k pomalému přístupu k této kameře (viz kapitola 6.5) byla nahrazena rychlejším RPi kamerovým modulem. Kamera byla následně přemístěna do zadní části vozidla jako zpětná, parkovací kamera. Byla doplněna o objektiv s krátkou ohniskovou vzdáleností 1,8 mm. Objektiv byl zvolen pro jeho velký zorný úhel 170°. Obě kamery se nachází uvnitř vozidla.

## 5 MECHANICKÉ ZPRACOVÁNÍ JEDNOTKY

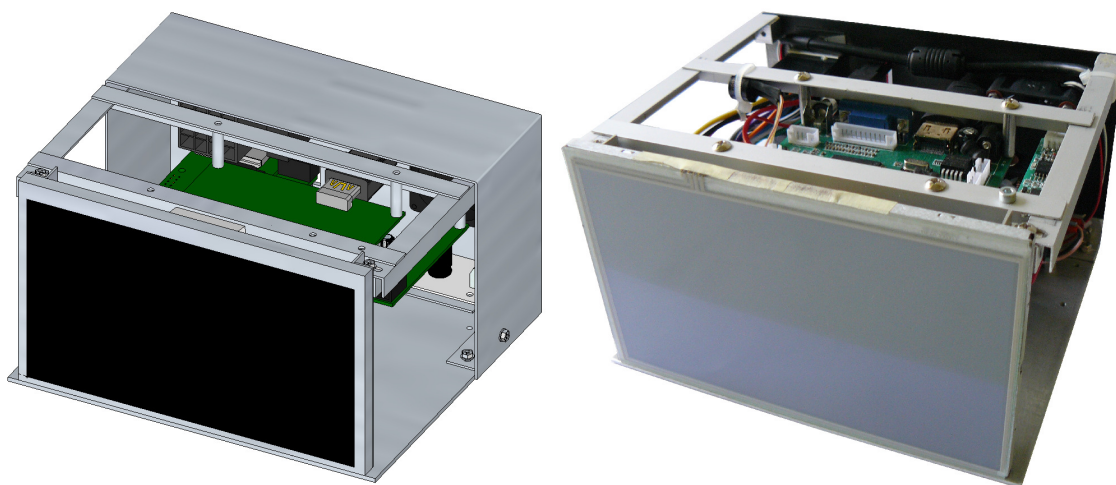
Protože bylo zvoleno zabudování celého systému do velikosti 2DIN rádia, musely být všechny komponenty umístěny do krabičky o rozměrech 180x100 mm, hloubka není standardizována, a proto byla zvolena 150 mm.

Model konstrukce byl nejprve vymodelován v programu SketchUp Make (verze 15.1.106), můžete jej vidět na obrázku 5.1. Tato konstrukce se vsune do 2DIN redukce příslušného automobilu a je k ní přichycena. Jediné, co bude potřeba vytvořit, je rámeček kolem displeje. Tím bylo zajištěno jisté univerzálnosti a přenositelnosti mezi různými druhy automobilů.

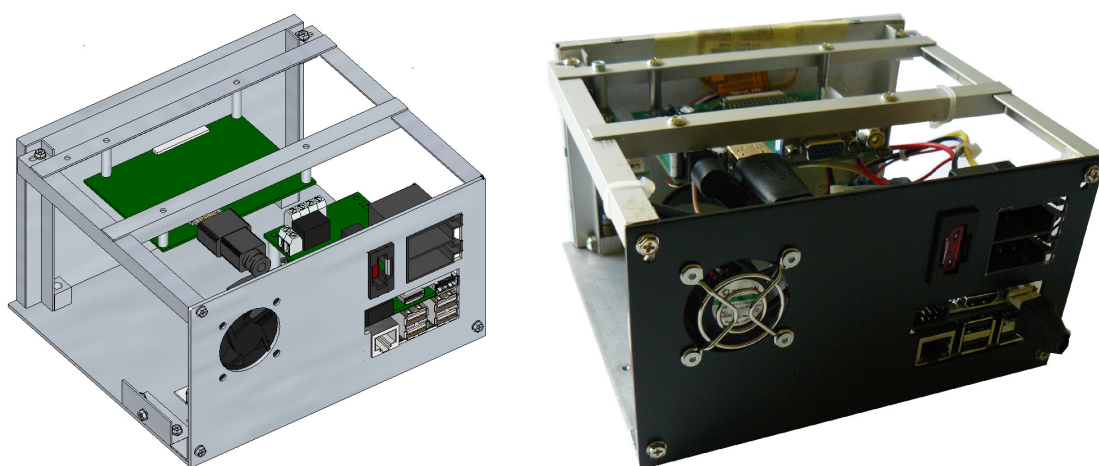
V této práci je konstrukce vsazena do univerzálního 2DIN rámečku pro vozy Seat, Škoda a VW. Protože je zobrazovací plocha displeje menší než otvor pro čelní panel 2DIN rámečku, byl vytvořen rámeček kolem displeje. Rámeček byl vytvořen z bílého plastu. Takto vytvořený rámeček byl přilepen k univerzálnímu 2DIN rámečku, přestříkán plničem a následně finální barvou.

Všechny části konstrukce jsou vyrobeny z hliníku, hliníkového plechu nebo duralu. Pro spojení bylo využito hliníkových L profilů a šroubků. Spodní plechové dno je z hliníkového plechu tloušťky 2,0 mm, stejně tak zadní plechová stěna, ta je využita pro chlazení stabilizátoru LM2576T-5, který je se stěnou propojen za pomoci hliníkového pásku. Vzhledem k celohliníkové konstrukci by neměl nastávat problém s přehříváním stabilizátoru. Plechový kryt je vyroben ze slabšího plechu tloušťky 1,0 mm.

Ke konstrukci byla vytvořena výkresová dokumentace, kterou je možné nalézt v příloze 4A. V této dokumentaci jsou uvedeny veškeré rozměry jednotlivých částí, rozmístění jednotlivých komponent atd.



Obr. 5.1: 3D model asistenční jednotky vymodelovaný v programu SketchUP (vlevo) a skutečná jednotka (vpravo)



Obr. 5.2: 3D model zezadu s odstraněným plechovým krytem (vlevo) a skutečná jednotka (vpravo)

## 6 SOFTWAREVÉ VYBAVENÍ

### 6.1 Raspbian

Operační systémy dostupné pro RPi:

1. RASPBIAN - Debian Wheezy
2. Snappy Ubuntu Core
3. PIDORA - Fedora Remix
4. OPENELEC - XBMC Media Center
5. RASPBMC - XBMC Media Center
6. RISC OS - nejedná se o linuxovou distribuci

Ne všechny operační systémy jsou podporovány oběma verzemi RPi. Pro tuto práci byl zvolen nejrozšířenější operační systém, a to RASPBIAN, který je podporován 1. i 2. verzí RPi. Tento systém byl vybrán hlavně z důvodu, že se jedná o primární operační systém pro RPi, čímž by měla být zajištěna plná kompatibilita s hardwarem i s použitými aplikacemi. Jelikož se jedná o nejpoužívanější operační systém, lze využít zkušeností ostatních uživatelů. Všechny dostupné operační systémy jsou volně stáhnutelné z oficiálních stránek Raspberry Pi <sup>1</sup> ve formátu zip, který po rozbalení obsahuje image operačního systému. Za zmínku také stojí NOOBS, který obsahuje zavaděč pro instalaci všech zmíněných operačních systémů. Při bootování z karty si tedy uživatel může zvolit systém, který chce využívat, a ten si nainstaluje. Na kartě může být vždy pouze jeden operační systém, protože instalace nového automaticky přemaže ten starý.

#### 6.1.1 Instalace

Jak již bylo zmíněno dříve, operační systém RPi se nachází na (micro)SD kartě. Před prvním spuštěním je tedy potřeba námi zvolený operační systém na kartu nahrát z jiného PC. Následující text popisuje postup nahrání v operačním systému Windows.

Pro nahrání \*.img souboru je potřeba program *Win32 Disk Imager* (v případě OS Windows). V programu vyberete \*.img soubor s operačním systémem. Zařízení je (micro)SD karta, na kterou chcete systém nahrát. Operační systém se nahraje kliknutím na tlačítko Write. Naopak za pomoci tlačítka Read je možné vytvářet zálohy celé paměťové karty. To je výhodné především v případě, že plánujete zásadnější změny v systému, například v jeho jádře. Další výhodou je, že systém je již nastaven podle vašich preferencí. Není tak potřeba instalovat vše od začátku, ale stačí si pouze vytvořit image karty, který pak nahrajete na druhou kartu a můžete jí použít ve druhém RPi. Nevýhoda je pouze v tom, že pokud máte 8GB SD kartu,

---

<sup>1</sup><http://www.raspberrypi.org/downloads/>

na které máte zaplněné 3GB, image bude mít velikost celé karty, tedy 8GB. Doporučená velikost karty je 2GB, to je ale opravdu hraniční hodnota. V práci byla použita microSD karta s kapacitou 16GB.

### 6.1.2 Základní nastavení, první spuštění

Po prvním spuštění RPi není zobrazeno grafické rozhraní, ale příkazový řádek, kde je potřeba provést přihlášení uživatele. Uživatelské jméno: pi, heslo: raspberry. Po přihlášení je potřeba provést základní nastavení příkazem:

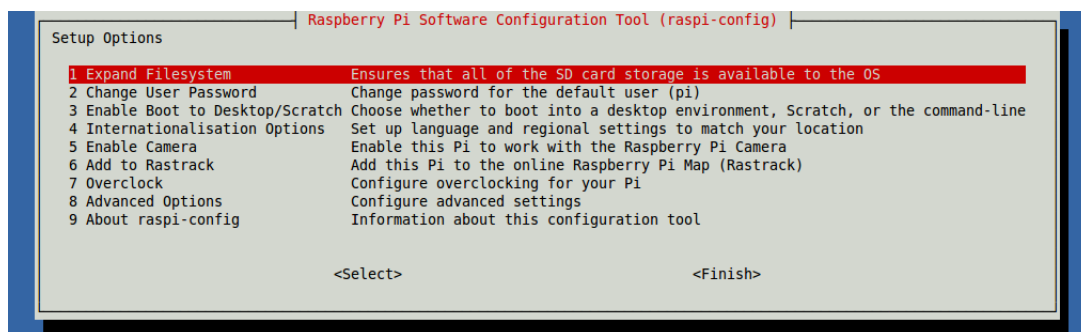
```
> sudo raspi-config
```

Tím se dostaneme k nastavení RPi (viz obr. 6.1). Zde je možné zvolit, zda má systém po spuštění zobrazit grafické rozhraní či příkazovou řádku, aktivovat kameru připojenou přes CSI, nebo přetaktovat procesor, který standardně běží na 900MHz. Pro vyšší frekvence se však doporučuje použití alespoň pasivních chladičů na RPi. Po instalaci Raspbianu je karta rozdělena na dva oddíly (FAT a Ext3). Protože Raspbian automaticky nenastavuje microSD kartu na plnou velikost, byl jako první příkaz spuštěn Expand Filesystem. Dále bylo nastaveno bootování systému do grafického rozhraní, frekvence procesoru na 1000 MHz. Po restartu byl systém aktualizován z terminálu následujícími příkazy:

```
> sudo apt-get update
```

```
> sudo apt-get upgrade
```

Dále bylo potřeba upravit rozlišení obrazovky na 720p (1280x720px), a to následují-



Obr. 6.1: Konfigurační menu Raspberry Pi

cím způsobem. Je potřeba upravit soubor config.txt, který se nachází ve složce boot. Zde je potřeba odkomentovat (umazáním znaku #) a upravit řádky:

```
hdmi_group = 1
```

```
hdmi_mode = 4
```

Na [elinux.org](http://elinux.org)<sup>2</sup> je seznam všech módů, které RPi podporuje, ne všechny jsou ale

<sup>2</sup>[http://elinux.org/RPi\\_config](http://elinux.org/RPi_config)

podporovány displejem použitým v této práci. Je potřeba mít na paměti, že soubor `config.txt` je systémový soubor a je potřeba k němu přistupovat s právy roota. Stejně tak při instalaci nových programů a knihoven. Před instalací programů je vždy vhodné provést aktualizaci systému, a to příkazy:

```
> sudo apt-get update
> sudo apt-get upgrade
```

Změna lokalizace, rozložení klávesnice a časového pásma se provádí opět za pomoci terminálu:

```
dpkg-reconfigure locales cs utf-8
dpkg-reconfigure keyboard-configuration Generic 105key -> Other -> Czechia
dpkg-reconfigure tzdata Europe -> Prague
```

Aby se změny projevily je nutné RPi restartovat.

### 6.1.3 Vzdálená správa

RPi umožňuje vzdálené připojení přes SSH (Secure Shell) protokol. Přihlašovací údaje jsou stejné jako ty, které byly uvedeny výše (pokud je tedy uživatel nezmění). Protokol SSH je ve výchozím nastavení zapnutý. Vypnout ho je možné pomocí konfiguračního menu RPi (příkaz `raspi-config`) v nabídce 8 Advanced Options.

Pro přístup k RPi je také výhodné mít zprovozněnou vzdálenou plochu, protože práce na malém 7" displeji je neefektivní. Ta se také vyplatí pro úpravy systému, až bude jednotka zabudovaná ve vozidle a nebude k ní možné snadnou cestou připojit myš s klávesnicí, protože pro ovládání se bude používat pouze dotyková obrazovka. Pro vzdálený přístup je v RPi použit VNC (Virtual Network Computing) server. Jeho instalace se provede následujícím příkazem:

```
> sudo apt-get install x11vnc
```

a spustí se příkazem:

```
> x11vnc -forever
```

Tento příkaz nezaručí, aby se VNC server spouštěl automaticky po každém spuštění RPi. K tomu je potřeba do adresáře `/etc/init.d/` nakopírovat soubor `vncserver`. Ten je možné nalézt v příložených souborech k práci. Tomuto souboru je potřeba nastavit atribut spustitelnosti:

```
> sudo chmod 755 /etc/init.d/vncserver
```

a dále:

```
> sudo update-rc.d vncserver defaults
```

Po restartu systému je možné se k RPi připojit za pomoci VNC klienta. Pro přenos souboru je možné použít například SFTP (Secure File Transfer Protocol) nebo SCP (Secure Copy) protokol.

## 6.1.4 Přístup k GPIO konektoru

K tomu, aby bylo možné softwarově přistupovat k GPIO konektoru, je potřeba pro něj nainstalovat ovladače. Aktuální verzi je možné nalézt na <sup>3</sup>. Ovladače stáhneme a nainstalujeme příkazem:

```
> sudo dpkg -i nazev_ovladacu.deb
```

Tím je GPIO konektor přístupný za pomoci příkazů z jazyka Python. Pro přístup k GPIO pinům z jazyka C je určena knihovna *wiringPI*. Její instalace se provede následující sekvencí příkazů z příkazové řádky:

```
> sudo apt-get install git-core
> git clone git://git.drogon.net/wiringPi
> cd wiringPi
> git pull origin
> ./build
```

Následujícími příkazy lze ověřit, zda instalace proběhla v pořádku.

```
> gpio -v
> gpio readall
```

## 6.1.5 Měření teploty s DS18B20

Jak již bylo popsáno v kapitole 4, měření teploty je realizováno za pomoci snímačů DS18B20. Ovladače pro komunikaci se snímačem jsou již součástí operačního systému a není nutné je instalovat. Je potřeba je pouze aktivovat přidáním řádku:

```
dtoverlay=w1-gpio,gpiopin=4
```

do souboru */boot/config.txt*. Měření se spustí příkazy:

```
> modprobe w1-gpio > modprobe w1-therm
```

Příkazem *lsmod* je možné ověřit, zda je měření aktivní. V případě, že jsou k RPi připojeny snímače DS18B20, v adresáři */sys/bus/w1/devices/* je vytvořena složka, jejíž název je shodný s unikátním kódem daného snímače. Tato složka obsahuje textový soubor nazvaný *w1\_slave*. Tento soubor obsahuje kontrolní součet určující správnost přečtení teploty (crc) a hodnotu teploty. Ukázka výpisu souboru pro snímač s unikátním kódem: 28-0000059d81c7,

```
2d 00 4b 46 ff ff 08 10 fe : crc=fe YES <- CRC souhlasí
```

```
2d 00 4b 46 ff ff 08 10 fe t=22250 <- Teplota: 22,250° C
```

```
b3 41 4b 46 7f ff 0d 30 9a : crc=7e NO <- CRC nesouhlasí
```

```
b3 01 4b 46 7f ff 0d 10 9a t=1051187 <- Nesprávná hodnota teploty
```

Vyčítání teploty daného snímače je realizováno čtením příslušného souboru.

---

<sup>3</sup><https://code.google.com/p/raspberry-gpio-python/downloads/list>

## 6.1.6 Raspberry Pi kamerový modul

Pro použití RPi kamerového modulu je potřeba jej aktivovat, to bylo provedeno při základním nastavení (*raspi-config*). Tím je modul přístupný pro používání z příkazové řádky (shell), nebo jazyka Python. Příklad pořízení snímku za pomoci příkazové řádky (jako správce):

```
> raspistill -o test_funkcnosti.jpg
```

Pořídí obrázek a uloží jej pod názvem *test\_funkcnosti.jpg* do aktuálního adresáře (za předpokladu, že do adresáře má uživatel povoleno zapisovat).

```
> raspivid -o test_video.h264 -t 10000
```

Pořídí záznam o délce 10 000 ms, který uloží pod názvem *test\_video.h264* opět do aktuálního adresáře. Jelikož je zpracování obrazu prováděno za pomoci knihovny OpenCV v jazyce C++, je potřeba ještě nainstalovat knihovnu *RaspiCam*, která umožňuje přístup k modulu za pomoci jazyka C++. Knihovna obsahuje třídu *RaspiCam* poskytující přístup ke kamerovému modulu a třídu *RaspiCam\_Cv* poskytující přístup ke kamerovému modulu ve spolupráci s knihovnou OpenCV. Instalace knihovny se provede následovně:

Nejprve je potřeba stáhnout knihovnu <sup>4</sup> a rozbalit. Pro verzi *raspicam-0.1.3*:

```
> cd raspicam-0.1.3
```

```
> mkdir build
```

```
> cd build
```

```
> cmake ..
```

```
> make
```

```
> sudo make install
```

```
> sudo ldconfig
```

## 6.1.7 Dotyková obrazovka

Po připojení převodníku do USB konektoru je dotyková vrstva sice aktivní, ale není zkalibrovaná. To je platné pro Raspbian s verzí jádra 3.18, u starších verzí nebyl převodník podporován a bylo zapotřebí překompilovat jádro. Pro kalibraci dotykové vrstvy je potřeba nainstalovat kalibrační program. Ten je potřeba stáhnout například za pomoci příkazu:

```
> wget http://github.com/downloads/tias/xinput_calibrator/xinput_calibrator-0.7.5.tar.gz
```

---

<sup>4</sup><https://sourceforge.net/projects/raspicam/files/>



rozbalit a nainstalovat.

```
> ./configure
```

```
> make
```

```
> sudo make install
```

Příkazem `xinput_calibrator` se spustí program pro kalibraci dotykové vrstvy. Na obrazovce se postupně objeví čtyři kalibrační body, které je potřeba zvolit. V terminálu se objeví informace o kalibraci. Abychom nemuseli kalibraci provádět po každém spuštění RPi, zkopírujeme úsek označený `-> Making calibration permanent` `<-` do souboru, který vytvoříme:

```
> sudo nano /usr/share/X11/xorg.conf.d/01-input.conf
```

V našem případě obsahoval úsek následující hodnoty:

```
Option "Calibration" "1976 69 187 1900"
```

### 6.1.8 Virtuální klávesnice

Jak již bylo zmíněno, zabudovaná jednotka ve vozidle nebude mít připojenou klávesnici ani myš, jejím jediným ovládacím prvkem bude dotyková obrazovka. Z toho důvodu byla do systému doinstalována jednoduchá softwarová klávesnice příkazem:

```
> sudo apt-get install matchbox
```

## 6.2 QT Creator

Celá aplikace pro Monitorování dopravní situace byla zpočátku vytvářena na PC mimo RPi. Protože kompilace aplikace na RPi byly zdlouhavé, bylo lepší je naprogramovat na PC a poté odladit na RPi, z důvodu, že PC neobsahuje GPIO konektor, RPi ani kamerový modul. Po přechodu na RPi verzi 2 tato činnost, díky jeho vyššímu výkonu, odpadla. Pro vytváření aplikace byl zvolen Qt Creator, vzhledem k tomu, že je multiplatformní. Jeho instalace se provede příkazem:

```
> sudo apt-get install qtcreator
```

QT Creator ke své funkci vyžaduje doinstalování kompilátoru, unixového emulátoru terminálu, systému správy verzí a systém pro správu a verzování zdrojových kódů.

Instalace se provede příkazy:

```
> sudo apt-get install gcc
```

```
> sudo apt-get install xterm
```

```
> sudo apt-get install git-core
```

```
> sudo apt-get install subversion
```

Nakonec je potřeba QT Creator ještě nastavit:

Tools -> Options -> Build & run -> záložka Tool chain -> Add -> GCC

Compiler path: /usr/bin/arm-linux-gnueabi-gcc-4.6

Debugger : /usr/bin/gdb

Mkspec : default

Help -> About plugins: Device support - odebrat Remote linux

Restartovat QT Creator a

Tools -> Options -> záložka Build & run -> QT version -> add “/usr/bin/qmake-qt4”

Následně je QT creator nastaven a připraven k použití.

## 6.3 OpenCV

Pro zpracování obrazu byla zvolena multiplatformní knihovna OpenCV (Open Source Computer Vision). Knihovna je svobodná a obsahuje mnoho funkcí a algoritmů pro transformaci obrazu, kompletní analýzu obrazu jako je jeho filtrování, úprava formátu, detekce objektů v obraze, dále také obsahuje funkce pro kalibraci kamery, práci s videem a jednoduchou práci s uživatelskými okny. Její výhodou je snadná komunikace se standardním hardwarem. Při použití kamery není potřeba její složité nastavování. Pro komunikaci využívá knihovna systémových ovladačů. Knihovna je pod BSD licenci a je možné jí využít v programovacích jazycích C, C++, Java a Python pod operačními systémy Android, Linux, Mac OS X a Windows. OpenCV má modulární strukturu a obsahuje následující moduly:

- **core** - Defnuje základní datové struktury *Mat* a základní funkce používané ostatními moduly.
- **imgproc** - Modul pro zpracování obrazu. Filtrace obrazu geometrické transformace, histogram.
- **calib3d** - Modul pro kalibraci jedné nebo stereo kamery, stereo korespondenční algoritmy a prvky pro 3D rekonstrukci.
- **video** - Modul pro zpracování videa obsahující například algoritmy pro sledování objektů.
- **features2d** - Příznaky, deskriptory.
- **objdetect** - Modul pro detekci objektů a prvků z předdefinovaných tříd např, obličej, části obličej, lidé, auta, atd.
- **highgui** - Obsahuje video kodeky a slouží pro pořizování snímků a videa z kamer a také pro jednoduché uživatelské rozhraní.
- **gpu** - Akcelerované algoritmy, z různých OpenCV modulů pro grafickou kartu.

Instalace knihovny se provede následovně: Nejdříve je potřeba doinstalovat do-datečné součásti:

```
>sudo apt-get -y install build-essential cmake cmake-curses-gui pkg-config
```

```
>sudo apt-get -y install libjpeg8 libjpeg8-dev libjpeg8-dbg
libjpeg-progs ffmpeg libavcodec-dev libavcodec53 libavformat53
libavformat-dev libgstreamer0.10-0-dbg libgstreamer0.10-0
libgstreamer0.10-dev libxine1-ffmpeg libxine-dev libxine1-bin
libunicap2 libunicap2-dev swig libv4l-0 libv4l-dev python-numpy
libpython2.6 python-dev python2.6-dev libgtk2.0-dev
```

V práci je použita verze 2.4.9, která se stáhne následovně:

```
>wget http://sourceforge.net/projects/opencvlibrary/files/opencv-
unix/2.4.9/opencv-2.4.9.zip/download opencv-2.4.9.zip
```

Rozbalení a příprava pro zpracování:

```
>unzip opencv-2.4.9.zip
>cd opencv-2.4.9
>mkdir release
>cd release
>ccmake ../
```

Celá instalace knihovny včetně kompilace trvá na RPi ve verzi 1 zhruba pět až šest hodin, ve verzi 2 je tento čas podstatně kratší a do jedné hodiny je celá knihovna nainstalovaná a plně funkční.

## 6.4 Test knihoven OpenCV, RaspiCam a WiringPi

Pro odzkoušení funkčnosti knihoven OpenCV, RaspiCam a zároveň i připojených kamer (web kamery a kamerového modulu) byl v QT Creatoru vytvořen program, jehož výpis je obsažen v příloze 1. Vytvořený program na začátku inicializuje webkameru a RPi kameru. V případě, že se mu to nepovede, je zobrazeno příslušné hlášení, že kamera nebyla připojena. V takovém případě k webkameře již přistupuje jiný program anebo přistupujeme ke špatné kameře (nepřipojené, neexistující) a je potřeba nadefinovat jinou hodnotu *CAM\_ID*. U RPi kamery je potřeba zkontrolovat, zda je správně připojená. V případě, že jsou kamery připojeny, pak jsou inicializovány GPIO piny. RPi je možné softwarově nastavit interní pull up, nebo pull down rezistor. Vzhledem k zapojení tlačítka (viz obrázek 6.2) byl nastaven vnitřní pull up rezistor. V nekonečné smyčce se kontroluje stisknutí tlačítka, pokud je tlačítko stisknuto, jsou pořizovány snímky z kamer a zobrazovány na displeji do příslušného okna. Dále se kontroluje, zda nebyla stisknuta nějaká klávesa na klávesnici, pokud ano, dojde k uzavření okna a ukončení programu. Před samotnou kompilací programu je potřeba v QT Creatoru nastavit cesty ke knihovně. To se provede v souboru s příponou *\*.pro*. Do souboru se vloží následující řádky:

```
INCLUDEPATH += /usr/local/include/opencv
```

```
LIBS += -L/usr/local/lib
```

```
-lopencv_core\
```

```
-lopencv_imgproc\
```

```
-lopencv_highgui\
```

```
-lopencv_calib3d\
```

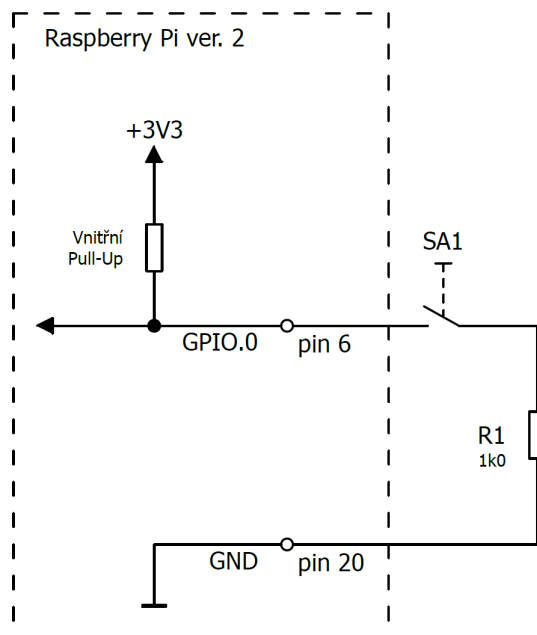
Jedná se v podstatě o namapování jednotlivých knihoven OpenCV, aby QT Creator věděl, kde se knihovna nachází. Pro následující ukázkou nejsou všechny využity, ale pro další práci s knihovnou budou potřeba. Namapování knihovny RaspiCam:

```
-lraspicam\
```

```
-lraspicam_cv\
```

a wiringpi pro ovládání GPIO konektoru.

```
-lwiringpi\
```



Obr. 6.2: Zapojení tlačítka k programu pro otestování knihoven OpenCV, RaspiCam a WiringPi

## 6.5 Porovnání rychlosti Raspberry Pi kamerového modulu a USB webkamery

Pro porovnání rychlosti obou kamer byl vytvořen program, jehož výpis je uveden v příloze 2. Počet pořízených snímků je volitelný přes konstantu MAX\_CAPTURE. Porovnání spočívá ve výpočtu FPS. Měří se doba  $t$ , za kterou bylo pořízeno  $n$  snímků.

$$FPS = \frac{t}{n}[FPS] \quad (6.1)$$

Průměrné hodnoty pro  $n = 10\,000$  snímků s rozlišením  $640 \times 480$  px jsou uvedeny v tabulce 6.1. Podle očekávání je z uvedených hodnot patrné, že USB webkamera je pomalejší.

Tab. 6.1: Porovnání FPS Raspberry Pi kamerového modulu a USB webkamery pro  $n = 10\,000$  snímků při rozlišení  $640 \times 480$  px

Kamera	Průměrná doba pořízení $n$ snímků [s]	FPS
RPi kamerový modul	334,0	29,9
USB webkamera Logitech C310	690,0	14,5

## 6.6 Softwarový přechod z Raspberry Pi verze 1 na verzi 2

Softwarový přechod z verze 1 na verzi 2 byl vzhledem ke zpětné kompatibilitě, kterou se výrobce snažil zachovat, bezproblémový. Programy, které byly vytvořené na RPi verze 1 stačilo pouze opětovně zkompileovat v novější verzi. Vyšší výkon se projevil především v odezvách systému, který je na novém RPi podstatně rychlejší. Výhodou je čtyřjádrový procesor podporující multivláknové operace. Dle výrobce je výkon přibližně šestkrát vyšší než u předchozí generace. To ovšem platí při využití všech jader procesoru.

## 7 GRAFICKÉ UŽIVATELSKÉ ROZHRAŇÍ - GUI

Pro interakci s uživatelem bylo vytvořeno grafické uživatelské rozhraní (Graphical User Interface (GUI)), to můžete vidět na obrázku 7.1. Toto rozhraní se automaticky spustí po startu RPi.



Obr. 7.1: Grafické uživatelské rozhraní - popis jednotlivých funkcí

### Popis ovládání GUI:

1. Datum a čas.
2. Zobrazení / skrytí roletového menu.
  - (a) Maximalizace okna aplikace.
  - (b) Minimalizace okna aplikace.
  - (c) Zapnutí / vypnutí akustické signalizace.
3. Vizualizace dopravních čar.
4. Obraz z přední kamery.
5. Obraz ze zadní kamery.
6. Upozornění pro řidiče.
7. Informační panel.
8. Teplota uvnitř vozidla.
9. Teplota vně vozidla.
10. Aktivace / deaktivace zaznamenávání snímků.
11. Aktivace / deaktivace přední kamery.

12. Aktivace / deaktivace zadní kamery.
13. Aktivace / deaktivace detekce jízdních pruhů.
14. Nastavení.
15. Vypnutí jednotky.
16. Oznamovací lišta.

## Datum a čas

Slouží k zobrazení systémového času. Pro vyčtení systémového času byla vytvořena funkce `GetSystemTime(int iFormat);`, která vrátí řetězec data a času v požadovaném formátu. Pro získání času je využita třída `QTime`. Funkce se nachází v `basefunction.h`, `basefunction.cpp`, kde jsou uvedeny i formáty data a času, které funkce vrací.

## Zobrazení / skrytí roletového menu

Po startu systému je automaticky spuštěno GUI na celou obrazovku. Za pomoci tlačítek maximalizace a minimalizace je možné se dostat přímo do systému Raspbian nebo opět aktivovat režim plné obrazovky. Tlačítko zapnutí / vypnutí akustické signalizace zapíná nebo vypíná zvukovou signalizaci.



Obr. 7.2: Roletové menu pro maximalizaci / minimalizaci aplikace a aktivaci / deaktivaci zvukové signalizace

## Vizualizace dopravních čar

Tato část slouží k zobrazení dopravních čar podle toho, jaké čáry se nachází na pravé nebo na levé straně od vozidla. Nově detekované dopravní čáry jsou ukládány do fronty FIFO o čtyřech prvcích. Zobrazená dopravní čára je ta, která je ve frontě obsažena nejvícekrát po sobě.

## Obraz z přední kamery

Zobrazuje obraz z přední kamery v případě, že je kamera aktivní. Pokud je kamera aktivní, po stisku do zobrazované plochy dojde k jejímu zvětšení. Zároveň pokud je aktivní i zadní kamera, je zobrazován i její obraz viz obrázek 7.3 (ukázka i s GUI v pozadí, v následujících ukázkách již bude bez GUI). Opuštění se provede tlačítkem zpět.



Obr. 7.3: Ukázka zobrazení okna pro obraz z přední i zadní kamery

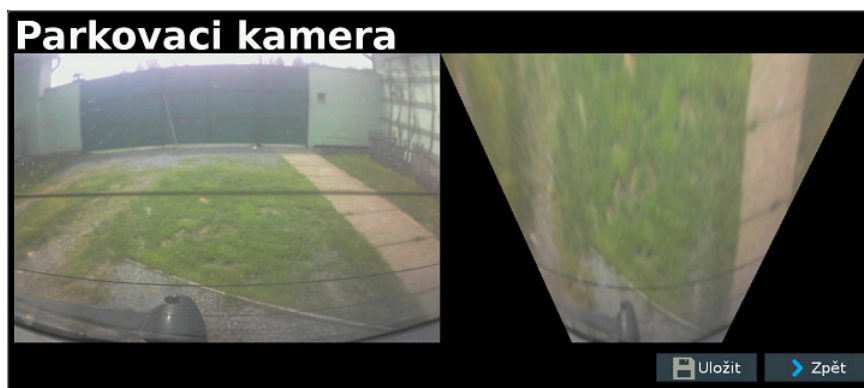
## Obraz ze zadní kamery

Zobrazuje obraz ze zadní kamery v případě, že je kamera aktivní. Pokud je kamera aktivní, po stisku do zobrazované plochy dojde k jejímu zvětšení viz obrázek 7.4. V pravé části je zobrazen pohled z ptáčích perspektivy pro usnadnění při parkování vozidla. Pro zobrazení pohledu z ptáčích perspektivy je využit algoritmus TVTM použitý také pro detekci jízdních pruhů. Při spuštění aplikace je vypočítána LUT pro zadní kameru a následně jsou za pomoci této tabulky transformovány všechny tři složky barevného obrazu ze zadní kamery. TVTM není počítán z celého pořízeného snímku, ale pouze z jeho spodní poloviny.

## Upozornění pro řidiče

Slouží jako vizuální upozornění pro řidiče. Aktuálně dostupné motivy pro zobrazení jsou uvedeny v obrázku 7.5. Pro zobrazení slouží funkce `InfoAlarm(int iAlarm);`.





Obr. 7.4: Ukázka zobrazení okna pro zadní kameru



Obr. 7.5: Ikony pro vizuální upozornění řidiče. Zleva: vše je v pořádku ( $iAlarm = 0$ ), upozornění na námrazu (ikona se objeví pokud vnější teplota klesne pod  $0\text{ }^{\circ}\text{C}$ ,  $iAlarm = 1$ ), přestávka (ikona momentálně není implementovaná,  $iAlarm = 2$ )

### Informační panel

Slouží k výpisu a záznamu hlášení systému. Je možné podle něj například určit nefunkční nebo odpojenou kameru.

### Teplota uvnitř / vně vozidla

Slouží k zobrazení teplot ze snímačů B1 a B2. Princip získání hodnoty je popsán v kapitole 6.1.5. Vyčtení hodnoty ze souboru a zpracování řetězců zajišťuje funkce `ReturnTemperatureF(QString sSensor);`, nacházející se v `RaspiTemp.h`, `RaspiTemp.cpp`. Vstupem funkce je unikátní kód daného snímače. Snímače tedy není možné mezi sebou prohodit. Funkce nejprve zjistí, zda je hodnota přečtena správně tím, že v prvním řádku hledá řetězec „YES”, pokud jej najde, ve druhém řádku hledá znak „=” a řetězec znaků za „rovná se” se převede na hodnotu typu float. V případě, že soubor nebyl nalezen (poškozený nebo odpojený snímač) nebo byla hodnota přečtena nesprávně, vrátí hodnotu teploty 999,0. Teplota je aktualizována každých pět minut.

## **Aktivace / deaktivace zaznamenávání snímků**

Slouží k zahájení / ukončení ukládání snímků ze zvolené kamery v případě, že je kamera aktivní. Výběr kamery, ze které se mají snímky ukládat, se provede v sekci nastavení. Snímky jsou ukládány do složky: /home/pi/TrafficMonitoring\_Gallery/ a dále podle toho, zda se jedná o přední nebo zadní kameru do složky Front\_ nebo Rear\_ s aktuálním datem. Název každého snímku je ve formátu "YYYYMM-DDThhmmss.png" (20150416T1801011.png). Čas je určen opět za pomoci funkce `GetSystemTime(4)`. Zaznamenávání je signalizováno modrým proužkem kolem tlačítka. Pro ukládání snímků byla vytvořena funkce `save_pic(QString path, Mat matImg, int iCam)`; jejím vstupem je cesta pro uložení snímku, snímek (který má být uložen) a z které kamery byl snímek pořízen. Výstupem funkce je hodnota `true` nebo `false` podle toho, zda se obrázek podařilo uložit. Funkce se nachází v `CameraTools.h`, `CameraTools.cpp`.

## **Aktivace / deaktivace přední, zadní kamery**

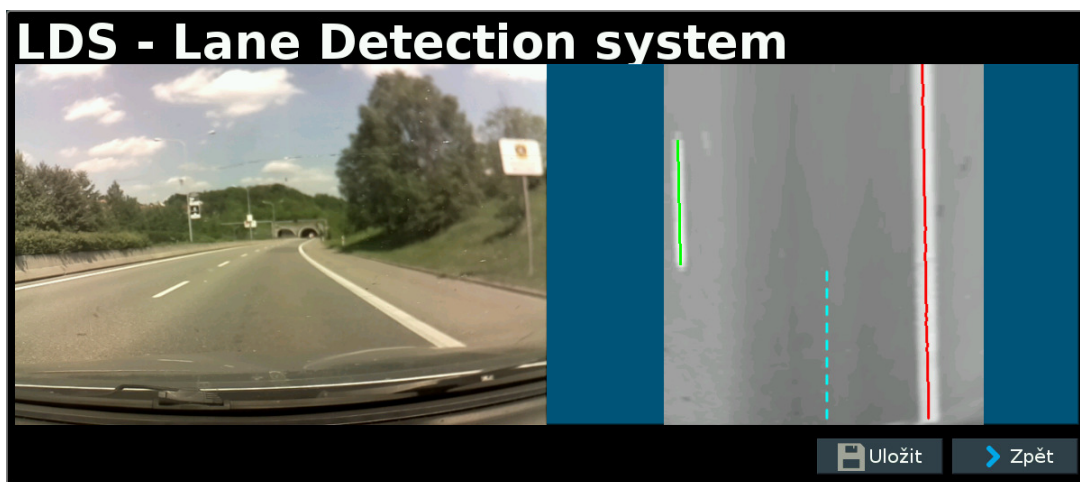
Tlačítko zapne nebo vypne příslušnou kameru. Aktivní kamera je signalizovaná modrým proužkem kolem tlačítka.

## **Aktivace / deaktivace detekce jízdních pruhů**

Slouží k vypnutí nebo zapnutí detekce jízdních pruhů. V nastavení je možné zvolit, zda se má vizualizace detekce provádět pouze do části Vizualizace dopravních čar nebo, do samostatného okna jako je na obrázku 7.6, kde je zobrazen obraz z přední kamery, pohled z ptáčích perspektivy, detekované jízdní pruhy a znázornění středu vozidla. Podmínkou je, aby se kamera nacházela uprostřed vozidla. Algoritmy detekce jsou popsány v kapitole 2. Při spuštění aplikace je vypočítaná LUT (stejně jako byla pro zadní kameru), protože pro detekci jízdních pruhů je opět použit TVTM. TVTM opět není počítán z celého snímku, ale pro určitou část, kterou je možné zvolit v nastavení.

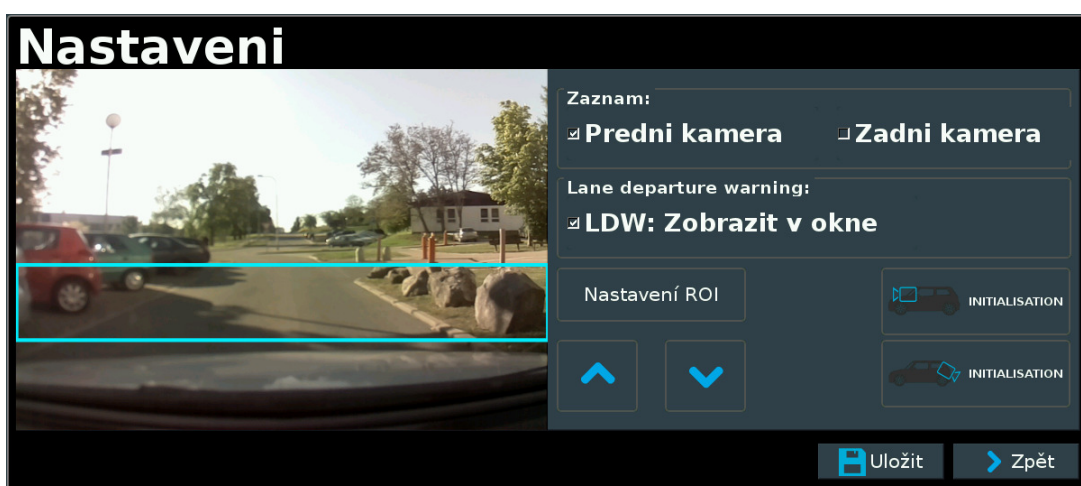
## **Nastavení**

Slouží k nastavení asistenční jednotky. Po stisknutí tlačítka se zobrazí okno viz obrázek 7.7. Zde je možné zvolit, z jaké kamery mají být ukládány snímky, možnost zobrazení detekce jízdních pruhů, zobrazovací plocha slouží k zaostření kamery a nastavení oblasti zájmu pro výpočet TVTM pro přední kameru. Obraz se spustí tlačítkem „Nastavení ROI“ a oblast zájmu se vybírá posunem nahoru a dolů zvýrazněným obdélníkem, posun se provádí za pomoci příslušných tlačítek. Obdélník určuje, z jaké části obrazu bude TVTM počítán. Je také možné nechat jednotku



Obr. 7.6: Ukázka zobrazení okna pro zobrazující výsledky detekce a rozpoznání dopravních čar

znovu přepočítat TVTM pro získání aktuální LUT přední a zadní. Tyto tabulky jsou počítány vždy při startu jednotky, kdy je kamerou pořízen jeden snímek, z kterého je LUT vypočítaná. V případě, že je kamera odpojena nebo nepřístupná během startu aplikace, je pro výpočet TVTM použit inicializační obrázek o stejné velikosti (640x480 px) jako obrázky pořízené kamerami, uložený u aplikace. V případě, že se uživateli obraz nezamlouvá, může použít tlačítka pro přepočítání. Tlačítkem uložit se nastavení uloží. Nastavení smí řidič, z důvodu bezpečnosti, provádět pouze za stojícího vozidla.



Obr. 7.7: Ukázka zobrazení okna pro nastavení asistenční jednotky

## **Vypnutí jednotky**

Po stisknutí tlačítka je uživatel ještě jednou dotázán, zda chce jednotku opravdu vypnout. Pokud od uživatele není žádná zpětná reakce na tuto výzvu, jednotka se po minutě z důvodu šetření baterie automaticky vypne. Situace je stejná i při vytažení klíče ze spínací skříňky.

## **Oznamovací lišta**

Zobrazuje aktuální hlášení systému, to je potom možné zpětně dohledat v informačním panelu 7.

## 8 ALGORITMUS DETEKCE A ROZPOZNÁNÍ JÍZDNÍCH PRUHŮ

Vzhledem k použitému hardwaru byl při vytváření algoritmů kladen důraz na co nejnižší výpočetní náročnost a nejvyšší spolehlivost. Dosažení obou parametrů je ve většině případů nemožné, a proto musel být vytvořen určitý kompromis. Pro detekci jízdních pruhů byl vytvořen algoritmus, který vychází z TVTM, tedy detekce čar z transformovaného obrazu. Algoritmus TVTM je doplněn o vlastní návrh detekce jízdních pruhů popsany v kapitole 2.2 a realizace v 8.3. Veškeré funkce, které byly vytvořeny pro zpracování obrazu a detekci čar, zdržuje knihovna *imageprocessing*. Knihovna obsahuje následující funkce:

Tab. 8.1: Soupis funkcí vytvořené knihovny *imageprocessing*

<code>computeTVTM</code>	Výpočet TVT
<code>CropLane</code>	Vyříznutí dopravní čáry
<code>FindLocalMin</code>	Nalezení minima mezi 2 maximy o definované vzdálenosti
<code>HorizHist</code>	Počet nenulových řádků ve sloupcích obrazu
<code>HorizHistU</code>	Suma jasových hodnot řádků ve sloupcích obrazu
<code>ImageThresholdIter</code>	Prahování iterativní metodou pro nalezení hodnoty prahu
<code>ImEdge</code>	Detekce hran v obraze. Dle kapitoly 2.2.3
<code>InterpolTVTM</code>	Vyplnění prázdných míst v obraze po TVT
<code>LaneBorder</code>	Nalezení souřadnic okrajů dopravní čáry v binárním obraze
<code>OptimizeTVTM</code>	Estimace parametrů $H$ , $f$ , $\Theta$
<code>RetTVImg</code>	TVT obrazu za pomoci LUT
<code>VertHist</code>	Počet nenulových sloupců v řádcích obrazu
<code>VertHistU</code>	Suma jasových hodnot sloupců v řádcích obrazu

### 8.1 Vytvořené třídy, struktury

Pro účely algoritmů byly vytvořeny struktury `LaneCoordinates`, `TVTM` a třída `RoadLane`.

#### Struktura `LaneCoordinates`

Tato struktura slouží pro uchování souřadnic levé a pravé strany detekované čáry. Její definice je následující:

```
typedef struct {
    vector<Point> L;
    vector<Point> R;
}LaneCoordinates;
```

## Struktura TVTM

```
typedef struct{
    Mat TvtmImg;
    int a;
    int b;
    int c;
    Point *lut;    // LookUpTable
}TVTM;
```

Tato struktura slouží pro uchování parametrů výpočtu TVTM, kde v proměnné `TvtmImg` je uložen transformovaný obraz, proměnné `a`, `b`, `c` jsou délky segmentů viz obrázků 2.2 a `lut` je tzv. Look Up Table neboli vyhledávací tabulka určená pro rychlou konverzi mezi netransformovaným a transformovaným snímkem, což je vysvětleno dále.

### Třída RoadLane

Tato třída obsahuje informace o jednotlivých dopravních čarách v obraze, tj. typ čáry, z kolika čar se skládá, pozici čáry. Třída také obsahuje metody pro separování čáry ze struktury `LaneCoordinates` a metody pro rozpoznání, o jaký druh čáry se jedná. Deklarace třídy je následující:

```
class RoadLane{
public:
    int Type;    // Typ cary
    int NumberLane;    // Pocet car, ze kterych je cara slozena (plna cara = 1,
    ...)
    vector<LaneCoordinates> Lanes; // Souradnice cary (leve a prave okraje)
    bool NoLane; // Informace jestli trida obsahuje caru
    double Desc1; // Deskriptor 1 - Delka(y) cary / vyska snimku
    double Desc2; // Deskriptor 2 - Delka(y) mezery / vyska snimku
    vector<vector<Point> > CenterOfLine; // Souradnice stredu cary
    // Konstruktory
    RoadLane(LaneCoordinates _LaneCoor, int iImageHeight);
    RoadLane( const RoadLane &obj); // Copy constructor
    RoadLane();
    // Desktruktor
    ~RoadLane();
private:
    double dCoeff;
    // Metody
    // Separace car ze souradnic pravdepodobnych car
    void SeparateLane(LaneCoordinates _LaneCoor, double dCoeff, int iImageHeight);
    void RecognizeLane(int iImageHeight); // Rozpoznani separovanych car
    void LineCenter(); // Vypocet souradnic stredu cary
};
```

## 8.2 Výpočet TVTM

Výpočet TVTM obrazu zajišťuje funkce TVTM `computeTVTM(Mat InImg, double H, double f, double theta)`. Vstupem funkce je obraz určený pro transformaci

parametry  $h$ ,  $f$ ,  $\Theta$ . Výstupem je struktura TVTM. Výpočet je prováděn za pomoci vztahů popsaných v kapitole 2.2. Výstupní obraz je ořezán a jsou určeny koeficienty  $a$ ,  $b$ ,  $c$ . Během výpočtu nových souřadnic je vytvářena LUT. Výstupní obraz obsahuje prázdná místa z důvodu popsaného v kapitole 2.2.2. K vyplnění prázdných míst slouží funkce `Mat InterpolTVTM(Mat InputImg)`; , která tato místa vyplňuje duplikací sousedních pixelů.

### 8.2.1 Estimace parametrů $H$ , $f$ , $\Theta$ - `OptimizeTVTM`

Estimaci parametrů provádí funkce `TVTM OptimizeTVTM(Mat TVTMImg, double H, double f, double theta)`; na základě algoritmu popsaného v kapitole 2.2.1. V případě, že jsou hodnoty  $a$ ,  $b$ ,  $c$  vzdálené od pevně stanovených odchylek, jsou hodnoty parametrů  $H$ ,  $f$  a  $\Theta$  měněny „nahrubo“. Pokud se k odchylkám přiblíží, jsou měněny s „jemným“ krokem. Postup estimace ilustruje vývojový diagram na obrázku 8.6.

### 8.2.2 Transformace obrazu - LUT

V případě, že nedojde k přemístění snímací kamery, případně ke změně jejího rozlišení, je možné výpočetně náročnou transformaci vypočítat pouze na počátku měření a transformaci ostatních snímků potom provádět za pomoci LUT. LUT neboli look up table je vyhledávací tabulka obsahující informaci o tom, jaké souřadnice netransformovaného obrazu připadají v transformovaném obraze. Pro převod mezi transformovaným a netransformovaným obrazem slouží funkce `Mat RetTVImg(Mat inputImg, TVTM TVTModel)`; . Jejím vstupem je obraz určený k transformaci a LUT, výstupem je transformovaný obraz.

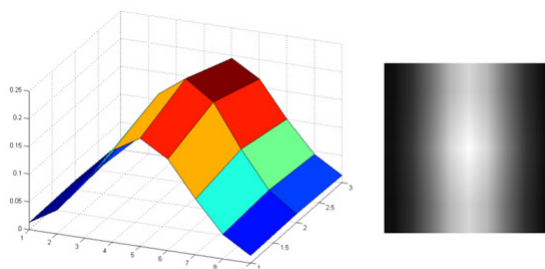
## 8.3 Detekce jízdnic pruhů

Celý průběh algoritmu pro detekci jízdnic pruhů je zobrazen v obrázkovém diagramu 8.5

### 8.3.1 Detekce hran - `ImEdge`

Pro hranování transformovaného obrazu byla vytvořena funkce `ImEdge(Mat InImg)`; , jejímž vstupem je transformovaný obraz a výstupem obraz vyhranovaný. Hranování je realizováno dle postupu popsaného v kapitole 2.2.3 za pomoci filtrů *ef1* a *ef2*. Před samotným hranováním je vstupní obraz filtrován „vertikálním“ gausovým filtrem (viz obrázek 8.1) pro potlačení šumu a zvýraznění vertikálních hran. Poté je

ve vyhranovaném obraze zvýšen jas pro potlačení šumu na pozadí a vyniknutí vertikálních čar. Na obrázku 8.4 je ukázka vyhranovaných snímků pro různé hodnoty osvětlení snímané scény.



Obr. 8.1: Použitý „vertikální“ Gausův filtr pro filtraci šumu a zvýraznění vertikálních hran

### 8.3.2 Horizontální histogram - HorizHist

Pro určení polohy dopravních čar z vyhranovaného obrazu slouží horizontální histogram, který vychází ze vztahu 2.14. Zjednodušeně jde o sumu jasových hodnot v jednotlivých sloupcích. Celý histogram je normalizován na rozsah počtu řádků obrazu. Od středu (střed snímku před transformací - za předpokladu umístění kamery uprostřed vozidla) vozidla jsou poté hledány lokální minima mezi blízkými maximy na obě strany obrazu. Čára je po vyhranování reprezentována dvěma rovnoběžnými přímkami o určité šířce, díky lokalizaci minima, je určení středu čáry přesnější než pouze z globálního maxima. Jako maximum je uznán vrchol, který je větší jak *počet řádků obrazu/4*, tedy potencionální čára je na výšku obsažena alespoň v 1/4 obrazu. Z vyhranovaného obrazu je vyříznuta část, jejíž výška odpovídá výšce obrazu a šířka je pozice vrcholu rozšířená o offset. Funkce pro výpočet horizontálního histogramu v binárním obraze se nazývá `HorizHist`, pro šedotónový obraz slouží funkce `HorizHistU`. Funkce pro hledání minim se nazývá `FindLocalMin`.

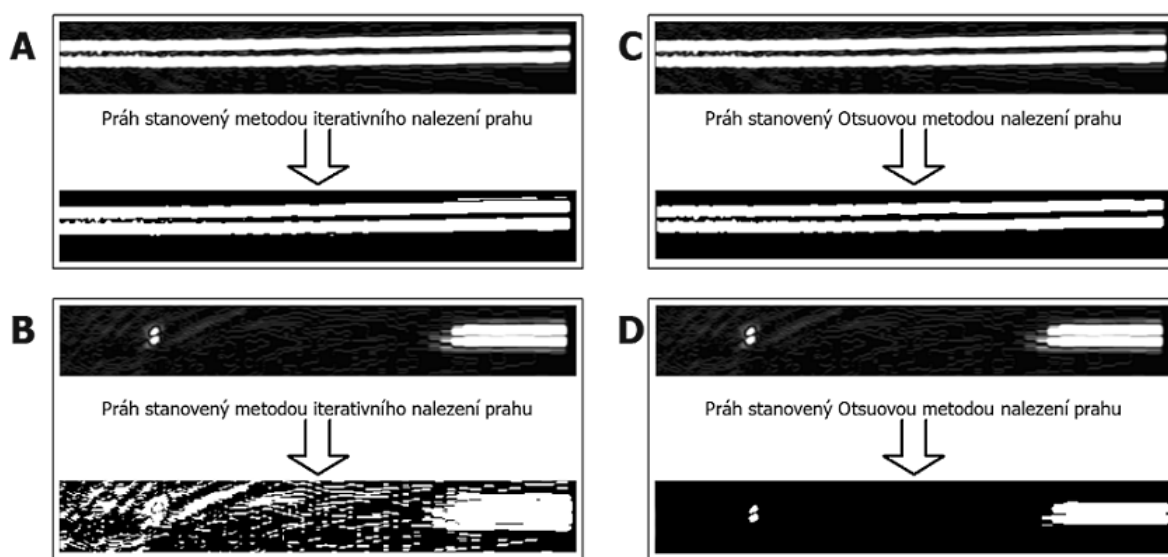
### 8.3.3 Prahování - ImageThresholdIter

Pro prahování vyříznuté oblasti s potencionální čarou byla použita metoda iterativního nalezení prahu popsána v kapitole 2.2.5. Funkce se nazývá `ImageThresholdIter` (`Mat inputImg, int iMaxIter`); a Jejím vstupem je obrázek, který má být vyprahován a maximální počet iterací. Tento počet slouží k omezení, aby se algoritmus nezastavil na prahování na příliš dlouhou dobu. Výstupem funkce je vyprahovaný obrázek. Metoda je založena na oddělení objektu od pozadí. Jak se ukázalo, aby byla čára oddělena od pozadí, musí být v obraze zastoupena ve větší míře než je



pozadí, to ale není vždy splněno a společně s čarou je vyprahováno částečně i pozadí viz obrázek 8.2 A, B.

Díky fitraci „vertikálním” gaussovým jádrem jsou vertikální pruhy jasně viditelné a obraz je bimodální, to znamená, že histogram obrazu obsahuje dva vrcholy. Obraz je tak nezávislý na osvětlení, a pro určení hodnoty prahu může být použita Otsuova metoda (kapitola 2.2.5) viz obrázek 8.2 C, D. Vyhranování snímků pro různé hodnoty osvětlení scény je uvedeno v obrázku 8.4, kde je vidět, že vyhranované snímky jsou opravdu bimodální. Prahování celého obrazu neprobíhá, protože vyřezané oblasti jsou rozměrově menší a prahování tak zabere méně času. Prahování je prováděno za pomoci funkce `threshold(vstup, vystup, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);`.



Obr. 8.2: Ukázka prahování: A, B - Určení prahu iterativní metodou. C, D - Určení prahu Otsuovou metodou

### 8.3.4 Rozpoznání a klasifikace podélných dopravních čar

Výstupem prahování je binární obraz, kde se s největší pravděpodobností nacházejí dopravní čáry. Úkolem klasifikace je určit, zda se opravdu jedná o dopravní čáru a následně o jaký typ čáry jde.

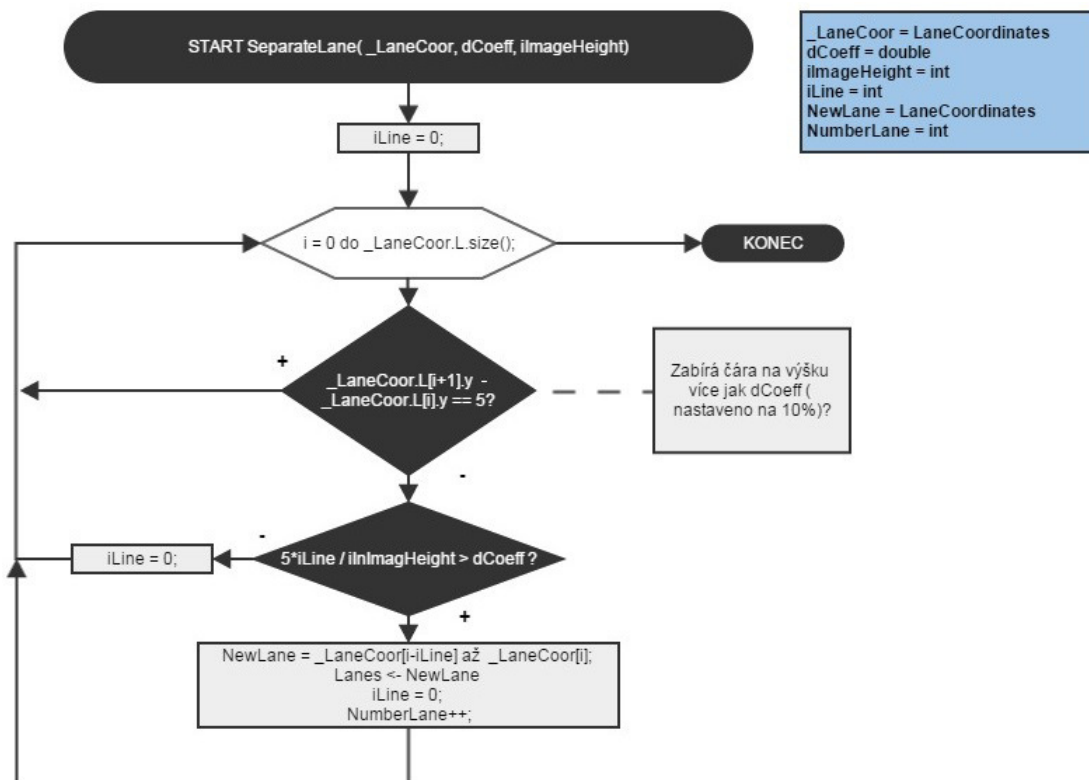
#### Rozpoznání podélných dopravních čar

Pro získání souřadnic potencionální čáry slouží funkce `LaneBorder(Mat InImg, int iMinLaneWidth, int iMaxLaneWidth, int iPosition, int iLaneOffset);`.

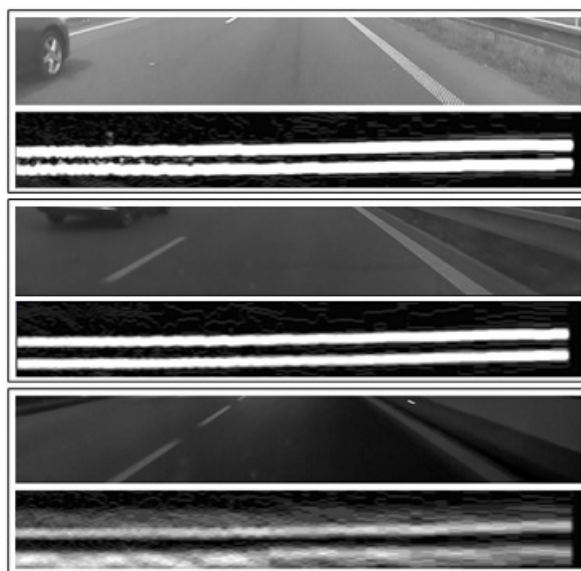
Vstupem funkce je binární obraz s vyříznutou potencionální čarou, minimální a maximální definovanou šířkou čáry v px, pozicí čáry a velikost offsetu vyříznuté čáry. Funkce prochází každý pátý řádek a zjišťuje, zda se na daném řádku nacházejí pixely s nenulovou hodnotou. Nejprve je nalezen první nenulový pixel z levé strany a poté je hledán nenulový pixel z pravé strany. Je - li vzdálenost nalezených pixelů v rozsahu minimální a maximální šířky čáry, jsou souřadnice nalezených pixelů uloženy do struktury `LaneCoordinates`. Výstupem funkce jsou tedy vektory souřadnic levé a pravé strany čáry. Tento algoritmus je opakován až do posledního řádku vyříznutého obrazu.

Pro rozdělení nalezených souřadnic na jednotlivé čáry slouží metoda třídy `RoadLane` nazvaná `SeparateLane`. Jejím vstupem je struktura `LaneCoordinates` a výška obrazu v px. Funkce prochází souřadnice y (řádky) a zjišťuje, jestli dvě sousedící souřadnice jsou od sebe vzdáleny 5 px, v případě že jsou, pokračuje dál, v případě že ne zjistí zda předešlé souřadnice zabírají v obraze na výšku alespoň 10 % plochy, pokud ano souřadnice jsou uloženy do proměnné `Lanes`. Proměnná `Lanes` obsahuje pole rozdělených čar. Funkci popisuje vývojový diagram na obrázku 8.3. Tímto mechanismem byly čáry rozseparovány a nyní zbývá určit, zda se jedná o čáru nepřerušovanou, přerušovanou, nebo přerušovanou připojovací, k tomu slouží metoda `RecognizeLane`. Ještě předtím jsou souřadnice levých a pravých okrajů čar přepočteny na souřadnice středu čar za (aritmetický průměr levé a pravé souřadnice) pomocí metody `LineCenter`.

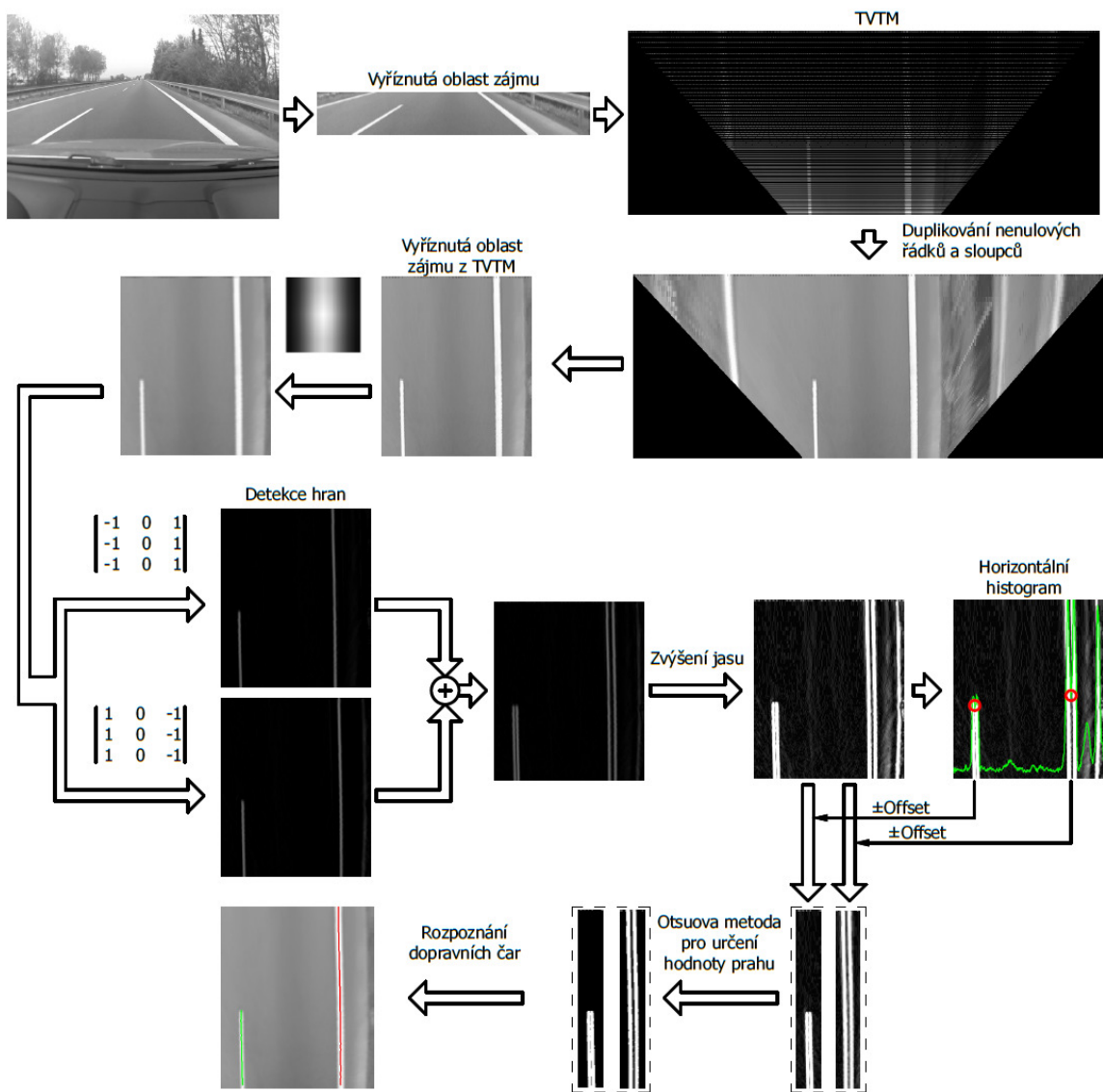
Metoda `RecognizeLane` na základě délek a mezer jednotlivých čar vypočte deskriptory popisující čáru. Deskriptor `Desc1` je poměr délky čáry ku výšce obrazu. Deskriptor `Desc2` je poměr délky mezery ku výšce obrazu. Je-li čára složena z více čar (přerušovaná) je nejdříve vypočtena průměrná délka čáry (mezery). Dle stanovených koeficientů je určeno o jakou čáru se jedná. Například čára nepřerušovaná bude v obraze zabírat alespoň 80% (zahrnuje i drobné přerušování) tedy její deskriptor `Desc1` by měl být vyšší než 0,8 a `Desc2` nižší než 0,2.



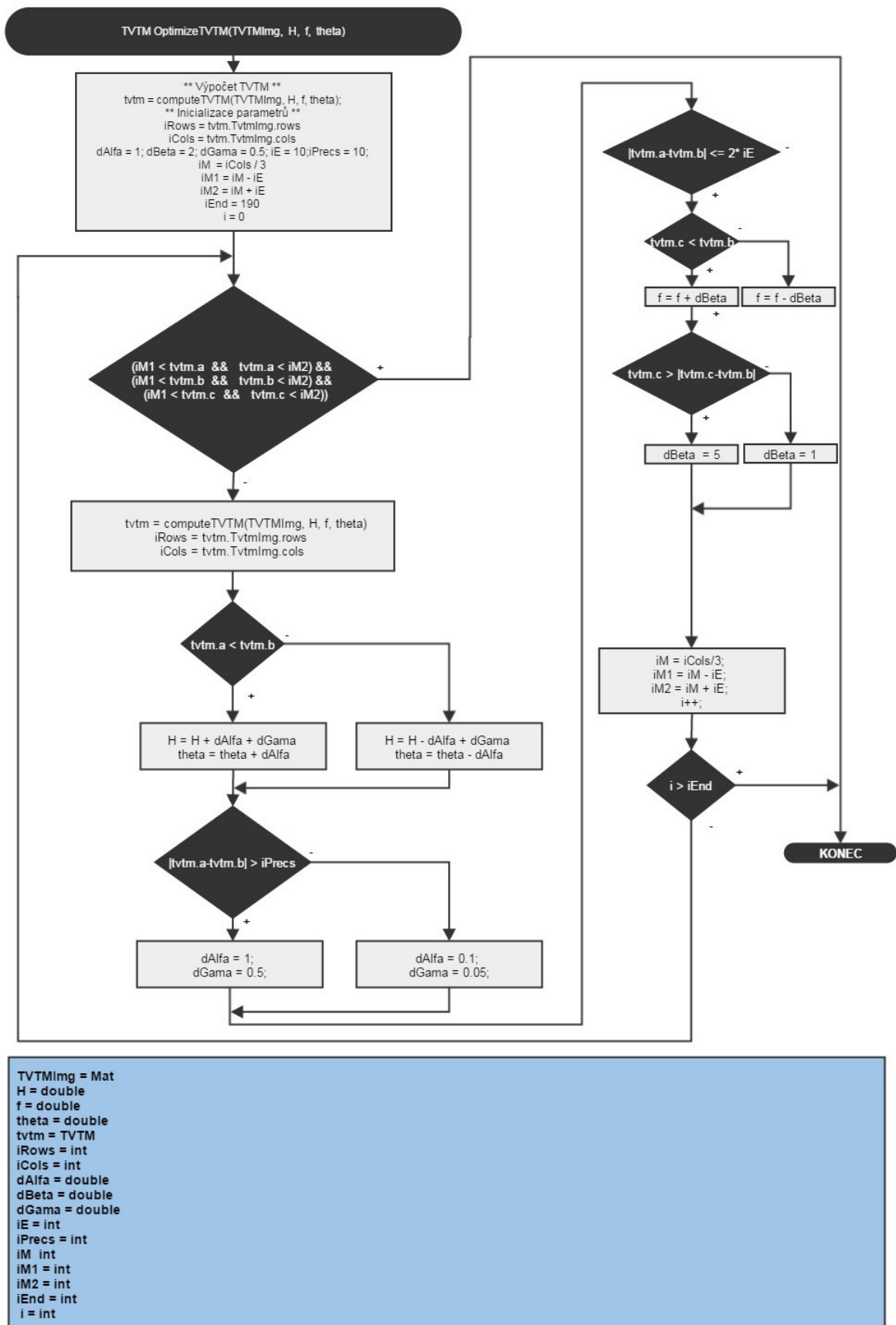
Obr. 8.3: Vývojový diagram metody SeparateLane třídy RoadLane



Obr. 8.4: Ukázka vyhranovaných snímků pro různé hodnoty osvětlení snímané scény



Obr. 8.5: Průběh algoritmu pro detekci jízdnic pruhů



Obr. 8.6: Vývojový diagram pro funkci *OptimizeTVTM*

## 9 VYHODNOCENÍ SPOLEHLIVOSTI SYSTÉMU PRO DETEKCI A ROZPOZNÁNÍ JÍZDNÍCH PRUHŮ

Pro ověření spolehlivosti detekce a rozpoznání jízdních pruhů byly ukládány snímky se zakreslenými dopravními čarami viz obrázek 9.1. Během každého měření bylo pořízeno alespoň tisíc snímků s periodou snímání 2 s při různých světelných podmínkách. Měření bylo dále rozděleno dle druhu pozemních komunikací. Vzhledem k tomu, že veškeré asistenční systémy komerčních výrobců nejsou aktivní pod rychlost 50 km/h, byly z měření vyřazeny snímky pořízené ve městech a obcích. Celkové shrnutí spolehlivosti je uvedeno ve dvou tabulkách. Tabulka 9.1 obsahuje vyhodnocení spolehlivosti pro dálnice a silnice I. třídy. Tabulka 9.2 obsahuje vyhodnocení spolehlivosti ostatních pozemních komunikací obsahujících vodorovné dopravní značení.

Tab. 9.1: Úspěšnost detekce a rozpoznání vodorovného dopravního značení pro dálnice a silnice I. třídy

Počasí a denní doba	Detekce jízdních pruhů	Rozpoznání vodorovného dopravního značení
Den - jasno	92 %	88 %
Den - zataženo	91 %	88 %
Noc	78 %	80 %

Tab. 9.2: Úspěšnost detekce a rozpoznání vodorovného dopravního značení pro pozemní komunikace mimo dálnice a silnice I. třídy obsahující vodorovné dopravní značení

Počasí a denní doba	Detekce jízdních pruhů	Rozpoznání vodorovného dopravního značení
Den - jasno	66 %	55 %
Den - zataženo	67 %	56 %
Noc	73 %	72 %

Z tabulek je patrné, že vytvořený systém je vhodný pro použití zejména na dálnicích a silnicích I. třídy, kde je vodorovné dopravní značení obvykle v dobrém stavu a bývá pravidelně obnovováno, aby nedošlo k jeho zániku. Poměrně dobrých výsledků dosahuje systém na těchto pozemních komunikacích i za tmy. Nesprávná detekce dopravní čáry byla způsobena nejčastěji u podélných souvislých čar jejich

přerušením. Tato přerušení jsou nejčastěji způsobena nánosem nečistot nebo opotřebením čáry.

Systém dosahuje mírného zhoršení při detekci jízdnic pruhů na ostatních pozemních komunikacích. Tyto silnice ne vždy obsahují vodící čáru a přechod vozovky v krajnici je v některých případech systémem mylně detekován jako dopravní čára. Další problém představují vozovky s poškozenými dopravními čarami často opravenými nánosem opravné směsi. Ukázka detekce a rozpoznání je uvedena v obrázku 9.3, některé snímky ještě obsahují zobrazené deskriptory (Desc1 a Desc2).

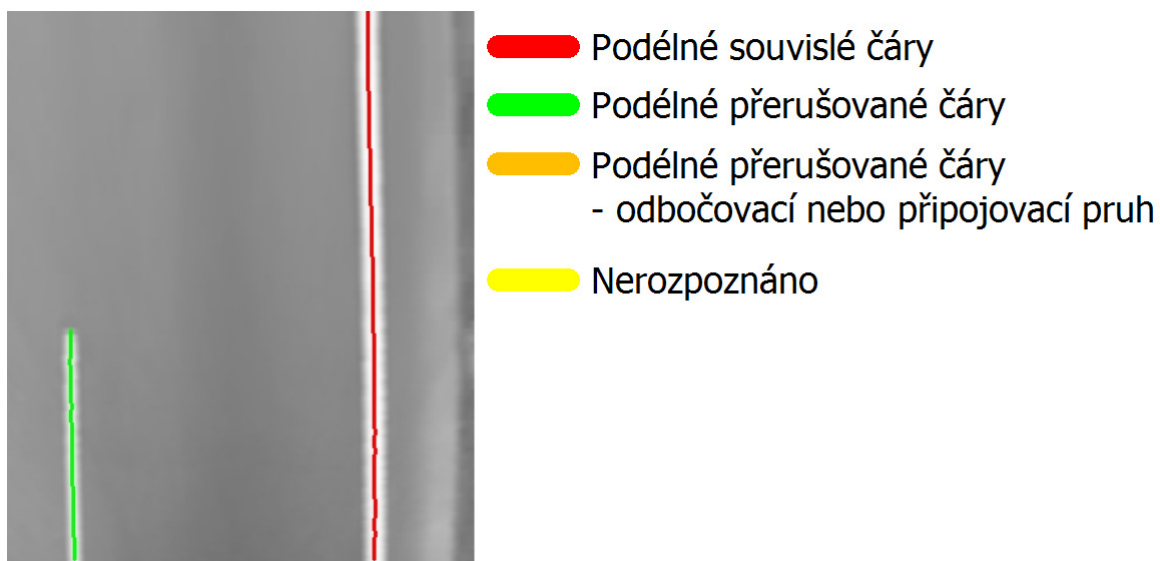
Pro kvalitnější posouzení algoritmů detekce a rozpoznání jízdnic pruhů byly vyřazeny snímky, které dokumentují špatný technický stav pozemních komunikací. Nově vyhodnocené spolehlivosti dokumentují tabulky 9.3 a 9.4. Při posouzení spolehlivosti při nezahrnutí poškozeného nebo velmi nevýrazného značení je úspěšnost algoritmů na velmi dobré úrovni.

Tab. 9.3: Úspěšnost detekce a rozpoznání vodorovného dopravního značení pro dálnice a silnice I. třídy bez započtených snímků s poškozenými dopravními čarami.

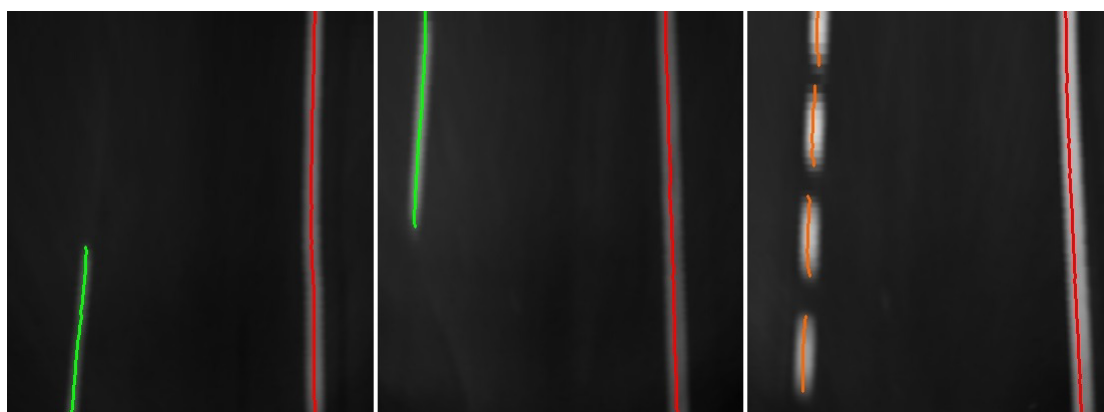
Počasí a denní doba	Detekce jízdnic pruhů	Rozpoznání vodorovného dopravního značení
Den - jasno	97 %	94 %
Den - zataženo	97 %	94 %
Noc	93 %	95 %

Tab. 9.4: Úspěšnost detekce a rozpoznání vodorovného dopravního značení pro pozemní komunikace mimo dálnice a silnice I. třídy obsahující vodorovné dopravní značení, bez započtených snímků s poškozenými dopravními čarami.

Počasí a denní doba	Detekce jízdnic pruhů	Rozpoznání vodorovného dopravního značení
Den - jasno	90 %	94 %
Den - zataženo	91 %	92 %
Noc	91 %	93 %

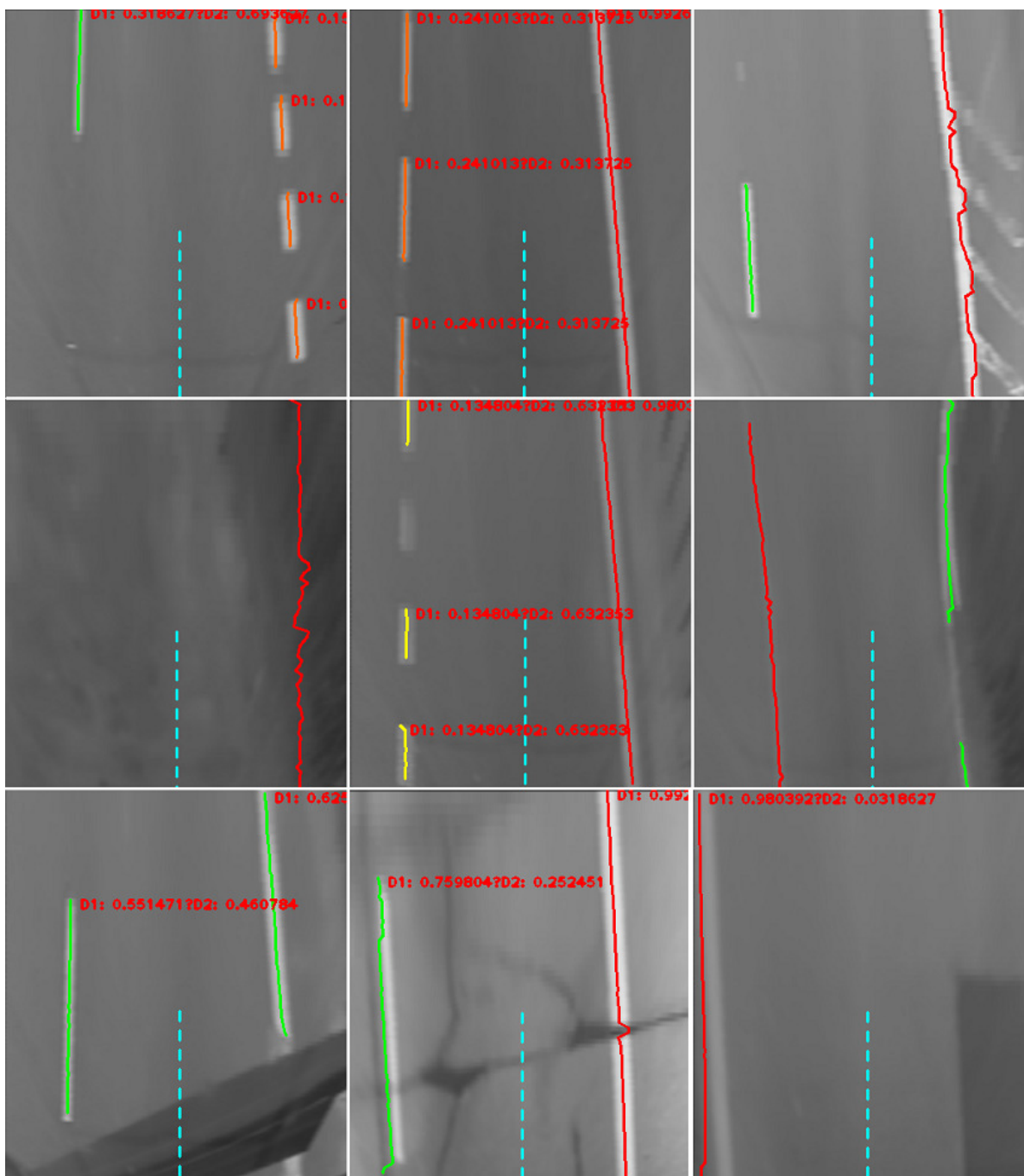


Obr. 9.1: Ukázka detekce a rozpoznání vodorovného dopravního značení



Obr. 9.2: Ukázka detekce a rozpoznání vodorovného dopravního značení v noci.





Obr. 9.3: Ukázka detekce a rozpoznání vodorovného dopravního značení. Druhý řádek zleva: silnice bez dopravních čar, špatná detekce, přerušovaná čára vlivem nečistoty. Třetí řádek zleva: Přerušování stěračem, praskliny ve vozovce a opravená vozovka. Přerušovaná světle modrá čára znázorňuje střed vozidla. Některé snímky obsahují zobrazené hodnoty deskriptorů (Desc1 a Desc2).

## 10 ZÁVĚR

V této práci byla navržena a vytvořena asistenční jednotka pro monitorování dopravní situace o velikosti 2 DIN [13]. Velikost jednotky umožňuje snadné umístění ve vozidle na místo autorádia, její zabudování ve vozidle je zobrazeno v obrázku 10.1. Řízení celého zařízení zajišťuje malý počítač Raspberry Pi 2 model B+, jehož čtyři USB konektory a ethernet port jsou vyvedeny na zadní straně asistenční jednotky. Propojení Raspberry Pi s vozidlem je realizováno pomocí konektoru ISO 10487 a rozšiřujícího modulu (kapitola 4.5). Modul zajišťuje napájení jednotky, připojení snímačů teploty a připojení kamery. Interakce jednotky s uživatelem je zajištěna přes 7 palcový dotykový displej.



Obr. 10.1: Ukázka zabudované jednotky pro monitorování dopravní situace ve vozidle

Jednotka je schopna monitorovat dopravní situaci před a za vozidlem (kapitola 4.6). Pro monitorování dopravní situace před vozidlem byla vytvořena kamera skládající se z Raspberry Pi kamerového modulu objektivu s proměnnou ohniskovou vzdáleností a pro propojení s Raspberry Pi navrženým převodníkem pro prodlužovací kabel. Jako prodlužovací kabel byl zvolen HDMI kabel, který obsahuje dostatečný počet vodičů a je určen pro vysokorychlostní přenos. Jelikož Raspberry Pi umožňuje připojení pouze jednoho kamerového modulu pro monitorování dopravní situace za vozidlem, je určena webkamera Logitech C310, připojená přes USB. Přístup ke kameře přes USB je pomalejší než přes CSI, ze zadní části vozidla není momentálně potřeba zpracovávat tolik snímků jako z přední kamery, proto je toto omezení

přijatelné. U webkamery byl vyměněn stávající objektiv za objektiv s ohniskovou vzdáleností 1,8 mm, z důvodu širokého úhlu záběru.

Co se softwarové části týče, tak operačním systéme je Raspbian. Grafické uživatelské rozhraní bylo vytvářeno v QT creatoru a algoritmy pro detekci jízdních pruhů využívají knihovny pro zpracování obrazu OpenCV. Grafické uživatelské rozhraní bylo svým ovládáním navrženo tak, aby řidiče neomezovalo v řízení, proto byla volena tlačítka dostatečně velká. Z důvodu bezpečnosti smí veškerá nastavení řidič provádět, pokud vozidlo stojí. Jednotka je schopna zobrazovat uživateli obraz z přední i zadní kamery a zaznamenávat snímky z obou kamer. Zadní kameru lze využít také jako parkovací. Po otevření příslušného okna se řidiči zobrazí obraz z kamery netransformovaný a vedle něj transformovaný na pohled z ptačí perspektivy, který řidiči poskytuje lepší odhad vzdálenosti od překážky. Dále je jednotka schopna uživateli zobrazit detekované jízdní pruhy. Jednotka je také vybavena snímači vnitřní a vnější teploty vozidla. Podrobnější popis ovládání GUI je uveden v kapitole 7.

Pro detekci jízdních pruhů obsahujících vodorovné dopravní značení byl navržen algoritmus vycházející z TVTM (kapitola 2.2), díky kterému je obraz transformován na pohled z ptačí perspektivy (obdobně je tomu i u zadní (parkovací) kamery). V takto transformovaném obraze je prováděna detekce jízdních pruhů (kapitola 8). K detekovaným dopravním čarám byl vytvořen i algoritmus pro jejich rozpoznání (kapitola 8.3.4) na základě procentuálního zastoupení čáry na výšku v obraze. Pro uchování, detekci a rozpoznání dopravních čar byla vytvořena třída Roadlane (kapitola 8.1).

Na závěr práce byla otestována funkcionalita a spolehlivost jednotky. Jednotka byla po celou dobu testování (a je i nyní) funkční. Během testování konečné verze aplikace nedocházelo ke stavům, kdy byl program ukončen či přerušen z důvodů sémantické chyby či neočekávané události. Z vyhodnocení detekce a rozpoznání jízdních pruhů (kapitola 9) je zřejmé, že detekční systém je vhodný zejména pro použití na dálnicích a silnicích I. třídy. Vzhledem k jednotvárnému počasí v průběhu testování byla detekce vyzkoušena pouze za jasného, oblačného počasí a za tmy.

Vytvořené práci bych se rád věnoval i v budoucnu. Plánuji zprovoznění SPI/CAN převodníku pro vyčítání informací zpočátku pouze o rychlosti a aktivitě směrových světel. S tímto cílem souvisí implementace plnohodnotného asistenčního systému pro upozornění řidiče na vybočení z jízdního pruhu (většina komponent je již připravena). Další úpravy budou spočívat ve zprovoznění FM radiového modulu, nahrazení webkamery za Raspberry Pi kamerový modul. Pro připojení více kamer bude potřeba zprovoznit případně navrhnout vlastní multiplexer pro přepínání mezi jednotlivými kamerami, protože Raspberry Pi momentálně umožňuje připojení pouze jedné kamery. Díky připojení více kamer bude možné jejich počet rozšířit o kamery

do zrcátek a vylepšit tak parkovacího asistenta. Vyšší počet kamer přináší vyšší počet snímků ke zpracování a tím i vyšší nároky na hardware. Z těchto požadavků plyne rozdělení aplikace do vláken a využít tak plného potenciálu čtyř-jádrového procesoru, kterým Raspberry Pi disponuje.

Vytvořená jednotka zdaleka nedosahuje komplexním asistenčním jednotkám v dnešních automobilech, ale za cenu materiálu cca 5000 Kč je v ní ukryt velký potenciál s rozsáhlými možnostmi rozšíření.

Tato diplomová práce pro mě byla velmi zajímavá a přínosná z hlediska získání znalostí a zkušeností. Jednalo se o multidisciplinární úlohu z oboru elektroniky, mechaniky, programování embedded systémů a zpracování obrazu.

## LITERATURA

- [1] Comfort and driving assistance systems: LaneGuide®. Valeo: Automotive technology, naturally [online]. 2013 [cit. 2015-04-20]. Dostupné z: <<http://www.valeo.com/en/our-activities/comfort-and-driving-assistance-systems/technologies/laneguide-80.html>>.
- [2] Connectors for car audio. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 11.12.2014 [cit. 2015-01-08]. Dostupné z: <[http://en.wikipedia.org/wiki/Connectors\\_for\\_car\\_audio#ISO\\_10487](http://en.wikipedia.org/wiki/Connectors_for_car_audio#ISO_10487)>
- [3] Continental Automotive: Drive and we'll keep you in the right lane. CONTINENTAL AG. Continental Automotive: Automotive Group [online]. 2015 [cit. 2015-04-20]. Dostupné z: <[http://www.conti-online.com/www/automotive\\_de\\_en/themes/passenger\\_cars/chassis\\_safety/adas/ldw\\_en.html](http://www.conti-online.com/www/automotive_de_en/themes/passenger_cars/chassis_safety/adas/ldw_en.html)>.
- [4] DELPHI AUTOMOTIVE LLP. ACTIVE SAFETY: Delphi Integrated Radar and Camera System [PDF]. [2014], 1 s. [cit. 2015-04-20]. Dostupné z: <<http://delphi.com/docs/default-source/old-delphi-files/61580fd3-7001-452e-a002-a54aa996628d-pdf>>.
- [5] DELPHI AUTOMOTIVE LLP. SAFETY: Delphi Intelligent Forward View Camera (100 Series) [PDF]. [2014], 1 s. [cit. 2015-04-20]. Dostupné z: <<http://delphi.com/docs/default-source/old-delphi-files/a72579e2-7c94-4a98-8d0e-68cb9bb64cd2-pdf>>.
- [6] DELPHI AUTOMOTIVE LLP. SAFETY: Delphi Intelligent Forward View Camera (200 Series) [PDF]. [2014], 1 s. [cit. 2015-04-20]. Dostupné z: <<http://delphi.com/docs/default-source/old-delphi-files/5f896c73-d605-4ae2-9aa0-d8e19be01a86-pdf>>.
- [7] Driver assistance systems: Multi purpose camera. ROBERT BOSCH GMBH. Bosch worldwide [online]. [2014] [cit. 2015-04-20]. Dostupné z: <[http://products.bosch-mobility-solutions.com/en/de/\\_technik/component/SF\\_PC\\_DA\\_Predictive-Emergency-Braking-System\\_SF\\_PC\\_Driver-Assistance-Systems\\_2567.html?compId=2880](http://products.bosch-mobility-solutions.com/en/de/_technik/component/SF_PC_DA_Predictive-Emergency-Braking-System_SF_PC_Driver-Assistance-Systems_2567.html?compId=2880)>.
- [8] Driver assistance systems: Stereo video camera . ROBERT BOSCH GMBH. Bosch worldwide [online]. [2014] [cit. 2015-04-20]. Dostupné z: <[http://products.bosch-mobility-solutions.com/en/de/\\_technik/](http://products.bosch-mobility-solutions.com/en/de/_technik/)>

component/SF\_PC\_DA\_Predictive-Emergency-Braking-System\_SF\_PC\_Driver-Assistance-Systems\_2567.html?compId=5443>.

- [9] GREENSTED, Andrew. [2009]. Otsu Thresholding. The Lab Book Pages: An online collection of electronics information [online]. [cit. 2015-05-08]. Dostupné z: <<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>>.
- [10] Hao Li, Fawzi Nashashibi. Lane Detection (Part I): Mono-Vision Based Method. [Technical Report] RT-0433, 2013. <hal-00781274> [cit. 2015-05-08]. Dostupné z: <<https://hal.archives-ouvertes.fr/hal-00781274/document>>.
- [11] HLAVÁČ, Václav a Milan ŠONKA. 1992. Počítačové vidění. Praha: Grada, 272 s. ISBN 80-854-2467-3.
- [12] HORÁK, Karel. 2010. Počítačové vidění: Segmentace objektů. Multimedia Interactive Didactic System [online]. [cit. 2015-05-13]. Dostupné z: <[http://midas.uamt.feec.vutbr.cz/POV/Exercise05/content\\_cz.php](http://midas.uamt.feec.vutbr.cz/POV/Exercise05/content_cz.php)>.
- [13] ISO 7736. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 9.9.2014 [cit. 2015-01-08]. Dostupné z: <[http://en.wikipedia.org/wiki/ISO\\_7736](http://en.wikipedia.org/wiki/ISO_7736)>.
- [14] JIAYONG, Deng, Kim JUNGHU, Sin HYUNCHUL a Han YOUNGJOON. 2013. Fast Lane Detection Based on the B-Spline Fitting. International Journal of Research in Engineering and Technology (IJRET) [online]. 2(4): 134-137 [cit. 2015-05-09]. ISSN 2277 – 4378 Dostupné z: <[http://journalsweb.org/siteadmin/upload/43881%20713035\\_N.pdf](http://journalsweb.org/siteadmin/upload/43881%20713035_N.pdf)>.
- [15] JONES, Dave. 2014. 7. Camera Hardware. Picamera: Picamera 1.10 documentation [online]. [cit. 2015-05-11]. Dostupné z: <<http://picamera.readthedocs.org/en/release-1.10/fov.html>>.
- [16] KRÚPA, Milan. Milan Krúpa: Raspberry Pi [online]. 17.11.2014 [cit. 2015-01-08]. Dostupné z: <<http://www.astromik.org/raspi/>>.
- [17] LIN, Chien-Chuan a Ming-Shi WANG. 2012. A Vision Based Top-View Transformation Model for a Vehicle Parking Assistant. Sensors [online]. 12(12): 4431-4446 [cit. 2015-05-09]. DOI: 10.3390/s120404431. ISSN 1424-8220. Dostupné z: <<http://www.mdpi.com/1424-8220/12/4/4431>>.
- [18] OMNIVISION TECHNOLOGIES, Inc. Datasheet OV5647: 1/4"color CMOS QSXGA (5 megapixel) image sensor with OmniBSI™ technology [PDF].

- 11.03.2009, 140 s. [cit. 2015-04-20]. Dostupné z: <[http://www.seeedstudio.com/wiki/images/3/3c/0v5647\\_full.pdf](http://www.seeedstudio.com/wiki/images/3/3c/0v5647_full.pdf)>.
- [19] OpenCV [online]. 2015. [cit. 2015-05-09]. Dostupné z: <<http://opencv.org/>>.
- [20] OTSU, Nobuyuki. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*. 9(1): 62-66. DOI: 10.1109/TSMC.1979.4310076. ISSN 0018-9472. Dostupné z: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4310076>>.
- [21] Otsu's method. 25 May 2005. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation [cit. 2015-05-08]. Dostupné z: <[http://en.wikipedia.org/wiki/Otsu%27s\\_method](http://en.wikipedia.org/wiki/Otsu%27s_method)>.
- [22] Qt Creator. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 6.1.2015 [cit. 2015-01-08]. Dostupné z: <[http://en.wikipedia.org/wiki/Qt\\_Creator](http://en.wikipedia.org/wiki/Qt_Creator)>.
- [23] Qt Documentation. Qt | Cross-platform application; UI development framework [online]. 2014 [cit. 2015-01-08]. Dostupné z: <<http://doc.qt.io/>>.
- [24] Raspberry Pi Documentation: Camera. [2013]. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [online]. [cit. 2015-05-09]. Dostupné z: <<https://www.raspberrypi.org/documentation/hardware/camera.md>>.
- [25] RASPBERRY PI FOUNDATION. Raspberry PI [online]. [2015] [cit. 2015-01-08]. Dostupné z: <<http://www.raspberrypi.org/>>.
- [26] RaspiCam: C++ API for using Raspberry camera with/without OpenCv. [2014]. Aplicaciones de la Visión Artificial [online]. [cit. 2015-05-09]. Dostupné z: <<http://www.uco.es/investiga/grupos/ava/node/40>>.
- [27] ROEBUCK, Kevin. 2011. ADASs - Advanced Driver Assistance Systems High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors. Dayboro: Emereo Publishing. ISBN 17-433-3886-4.
- [28] RPi Low-level peripherals: General Purpose Input/Output (GPIO). ELinux.org [online]. 10.10.2014 [cit. 2015-01-08]. Dostupné z URL: <[http://elinux.org/RPi\\_Low-level\\_peripherals#General\\_Purpose\\_Input.2FOutput\\_.28GPIO.29](http://elinux.org/RPi_Low-level_peripherals#General_Purpose_Input.2FOutput_.28GPIO.29)>.
- [29] SEIDL, Antonín. 2013. Revize TP 133 Zásady pro vodorovné dopravní značení na pozemních komunikacích [PDF]. [cit. 2015-05-08]. ISBN nemá. Dostupné z: <<http://www.pjpk.cz/TP%20133.pdf>>.

- [30] TRW AUTOMOTIVE. Scalable Camera Family [PDF]. 2013, 1 s. [cit. 2015-04-20]. Dostupné z: <[http://www.trw.com/sites/default/files/TRW\\_ge\\_scalablecamera\\_en.pdf](http://www.trw.com/sites/default/files/TRW_ge_scalablecamera_en.pdf)>.
- [31] Virtual Keyboard for the Raspberry Pi. Ozzmaker.com - Blip, bloop, bloop... [online]. [30.6.2014] [cit. 2015-01-08]. Dostupné z URL: <[http://ozzmaker.com/2014/06/30/virtual-keyboard-for-the-raspberry-pi/?utm\\_source=feedly&utm\\_reader=feedly&utm\\_medium=rss&utm\\_campaign=virtual-keyboard-for-the-raspberry-pi](http://ozzmaker.com/2014/06/30/virtual-keyboard-for-the-raspberry-pi/?utm_source=feedly&utm_reader=feedly&utm_medium=rss&utm_campaign=virtual-keyboard-for-the-raspberry-pi)>.
- [32] WANG, Yue, Eam KHWANG TEOH a Dinggang SHEN. Lane detection and tracking using B-Snake. Image and Vision Computing [online]. 2004, roč. 4, č. 22, s. 269-280 [cit. 2015-04-23]. DOI: 10.1016/j.imavis.2003.10.003. Dostupné z: <<http://www.sciencedirect.com/science/article/pii/S0262885603002105>>.
- [33] WATZKE, David. Grafické programy v Qt 4 - 3 (Qt Creator a Designer. AbcLinuxu.cz [online]. 23. 3. 2009 [cit. 2015-01-08]. Dostupné z: <<http://www.abclinuxu.cz/clanky/programovani/graficke-programy-v-qt-4-3-qt-creator-a-designer/>>.



## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CSI	Camera serial interface
FIFO	First in first out - Fronta (první dovnitř, první ven)
FPS	Frame per second - počet snímků za sekundu
GUI	Graphical User Interface
IPM	Inverse perspective mapping
LDWS	Lane departure warning system
LKS	Lane keeping system
LUT	Look up table - převodní tabulka
px	Pixel - obrazový bod
RPi	Raspberry Pi
TVTM	Top view transformation model

## SEZNAM PŘÍLOH

1	Program pro test knihoven OpenCV, RaspiCam a WiringPi	83
2	Program pro měření rychlosti FPS kamer	84
3	Obsah přiloženého CD	86
4	Výkresová dokumentace monitorovací jednotky	87

# 1 PROGRAM PRO TEST KNIHOVEN OPENCV, RASPICAM A WIRINGPI

```
#include <iostream>
#include <opencv2/opencv.hpp>
#include "wiringPi.h"
#include "raspicam_cv.h"
#define CAM_ID 0
#define PIN_0 0
using namespace std;using namespace cv;using namespace raspicam;
int main()
{
    cout << "Testovani_kamery..." << endl;
    VideoCapture cam_web(CAM_ID); // Web kamera
    RaspiCam_Cv cam_rpi; // RPi kamera
    // Zmena rozliseni u RPi kamera na 640x480 px
    cam_rpi.set(CV_CAP_PROP_FRAME_WIDTH, 640);
    cam_rpi.set(CV_CAP_PROP_FRAME_HEIGHT, 480);
    cam_rpi.open();
    // V pripade, ze kamera nebyla pripojena zobrazi hlaseni
    // a ukonci program. Potom je potreba zmenit cislo kamery CAM_ID
    if (!cam_web.isOpened())
    {
        cout << "Web_kamera_nebyla_pripojena." << endl;
        return -1;
    }
    if (!cam_rpi.isOpened())
    {
        cout << "RPi_kamera_nebyla_pripojena." << endl;
        return -2;
    }
    wiringPiSetup (); // Setup pro GPIO piny
    pinMode (PIN_0, INPUT); // Nastaveni pinu 0 jako vstupniho
    pullUpDnControl(PIN_0, PUD_UP); // Pull up
    // Promenne do, kterych se ukladaji snimky z kamer
    Mat img_web; Mat img_rpi;
    while(1) // Nekonecna smyka
    {
        if (digitalRead(PIN_0) == LOW) // kontrola stisknuti tlacitka
        {
            cam_web >> img_web; // Porizeni snimku z web kamery
            cam_rpi.grab(); // Porizeni snimku z RPi kamery
            cam_rpi.retrieve(img_rpi);
            imshow("web_kam",img_web); // Zobrazeni snimku
            imshow("rpi_kam",img_rpi); // Zobrazeni snimku
            // Ukoneni programu pi stisknut libovolne klvesy
        }
        if(waitKey(10) >= 0)
        {
            cout << "Ukoncuji_program." << endl;
            break;
        }
    }
    cam_web.release(); // Uvolneni kamery
    cam_rpi.release(); // Uvolneni kamery
    return 0;
}
```

## 2 PROGRAM PRO MĚŘENÍ RYCHLOSTI FPS KAMER

```
#include <ctime>
#include <fstream>
#include <iostream>
#include <raspicam/raspicam.h>
#include <raspicam/raspicam_cv.h>
#include <opencv2/opencv.hpp>
#include <opencv2/highgui/highgui.hpp>
#define CAM_ID 0
#define MAX_CAPTURE 10000
using namespace std;using namespace cv;using namespace raspicam;
int main()
{
    cout << "Testovani_kamery..." << endl;
    VideoCapture cam_web(CAM_ID); // Web kamera
    RaspiCam_Cv cam_rpi; // RPi kamera
    // Zmena rozliseni u RPi kamery na 640x480 px
    cam_rpi.set(CV_CAP_PROP_FRAME_WIDTH, 640);
    cam_rpi.set(CV_CAP_PROP_FRAME_HEIGHT, 480);
    cam_rpi.open();
    time_t timer_begin, timer_end;

    if (!cam_web.isOpened())
    {
        cout << "Web_kamera_nebyla_pripojena." << endl;
        return -1;
    }
    if (!cam_rpi.isOpened())
    {
        cout << "RPi_kamera_nebyla_pripojena." << endl;
        return -2;
    }
    Mat img_web; Mat img_rpi;
    cout << "Mereni_RPi_kamery_zahajeno" << endl;
    time ( &timer_begin ); // Start timer
    for (int i = 0; i < MAX_CAPTURE; i++)
    {
        cam_rpi.grab(); // Porizeni snimku z RPi kamery
        cam_rpi.retrieve(img_rpi);
    }
    time ( &timer_end ); // Stop timer
    cam_rpi.release();
    cout << "Mereni_RPi_kamery_ukonceno" << endl;
    double secondsElapsed = difftime ( timer_end, timer_begin );
    cout << "RPi_kamera:" << secondsElapsed << "sekunda" << MAX_CAPTURE << "snimku"
        << "FPS=" << ( float ) ( ( float ) ( MAX_CAPTURE ) / secondsElapsed ) <<
        endl;
    cout << "Mereni_WEB_kamery_zahajeno" << endl;
    time ( &timer_begin ); // Start timer
    for (int i = 0; i < MAX_CAPTURE; i++)
    {
        cam_web >> img_web; // Porizeni snimku z web kamery
    }
    time ( &timer_end ); // Stop timer
    cam_rpi.release();
```

```
cout << "Mareni WEB kamery - ukonceno" << endl;
secondsElapsed = difftime ( timer_end, timer_begin );
cout<< "WEB kamera:" <<secondsElapsed<<" sekunda" << MAX_CAPTURE<<" snimku:
FPS=" << ( float ) ( ( float ) ( MAX_CAPTURE ) /secondsElapsed ) <<endl;
return 0;
}
```

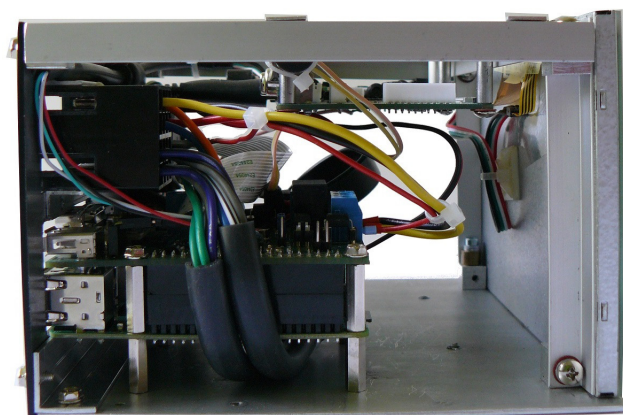
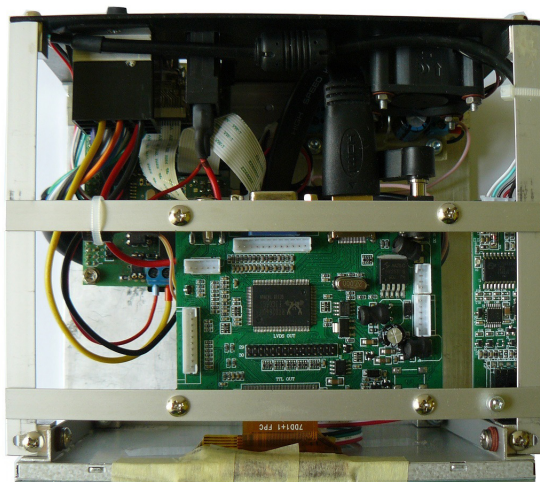
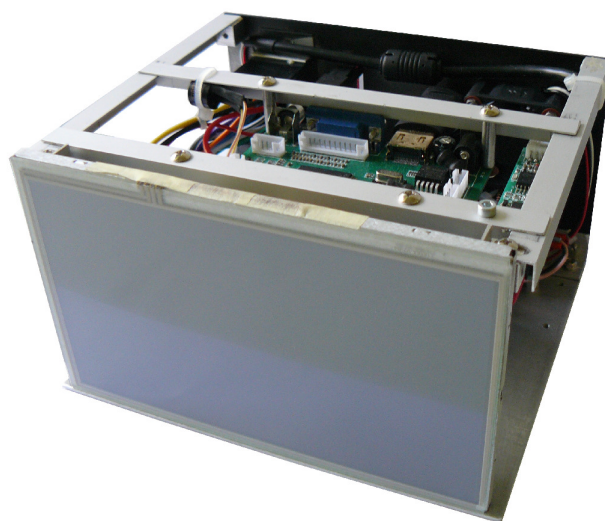
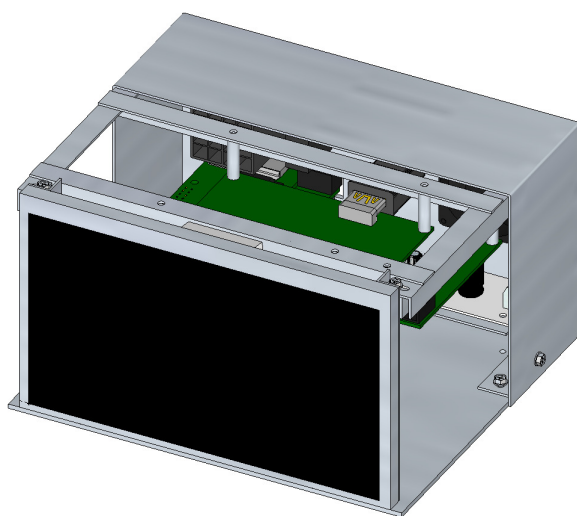
### **3 OBSAH PŘILOŽENÉHO CD**

- Elektronická verze diplomové práce
- Program pro test knihoven OpenCV, RaspiCam a WiringPi
- Program pro měření rychlosti FPS kamer
- Vytvořená aplikace pro ovládání jednotky
- Zdrojové kódy k vytvořené aplikaci, ve formě projektu z vývojového prostředí QT Creator

## 4 VÝKRESOVÁ DOKUMENTACE MONITOROVACÍ JEDNOTKY

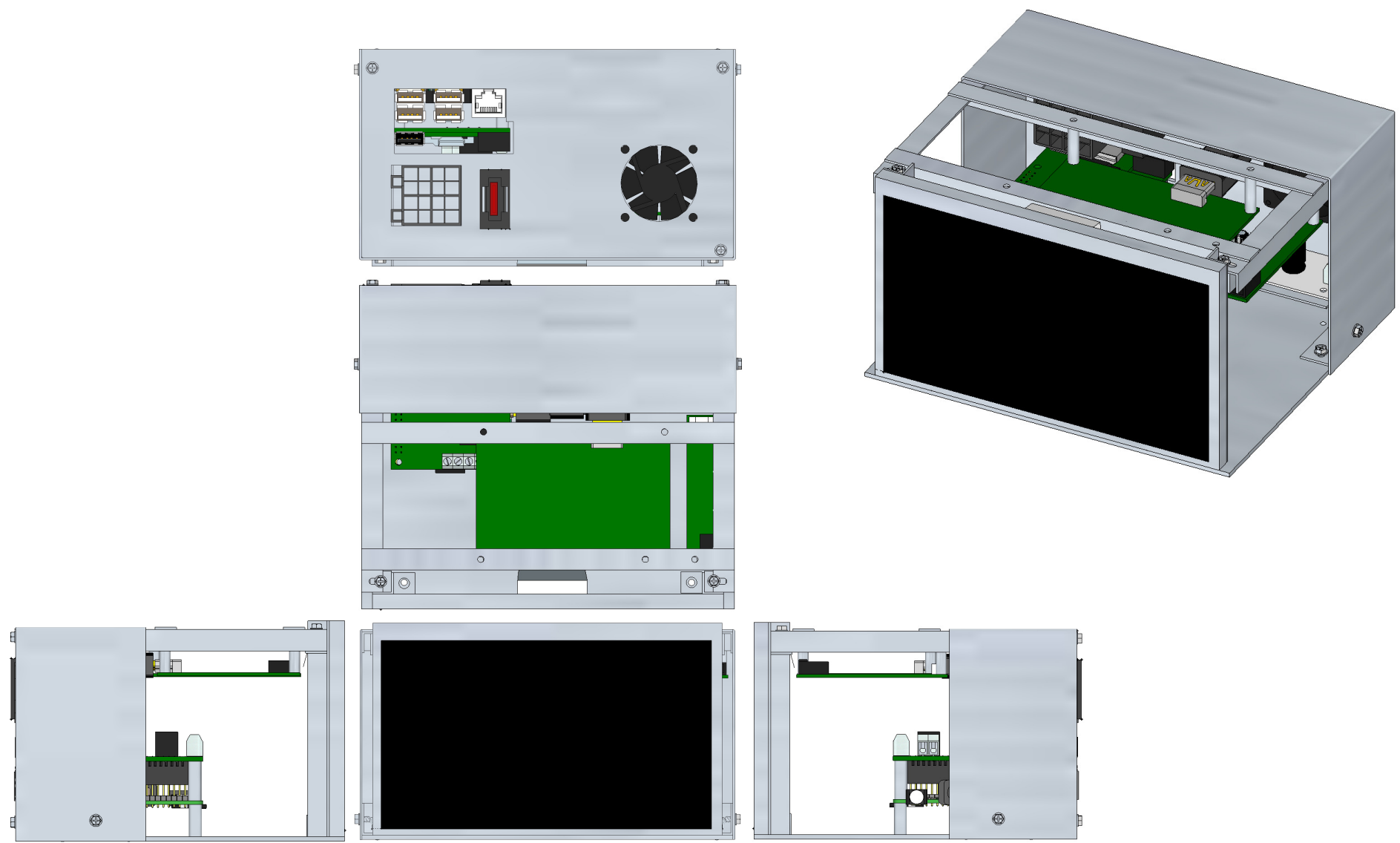
Výkresová dokumentace se nachází na vložených listech a obsahuje následující:

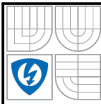
- A - Technické výkresy mechanické části
- B - Blokové schéma zapojení jednotky
- C - Schéma zapojení
- D - Kusovník artiklů



1 2 3 3 4 5 6 6 7

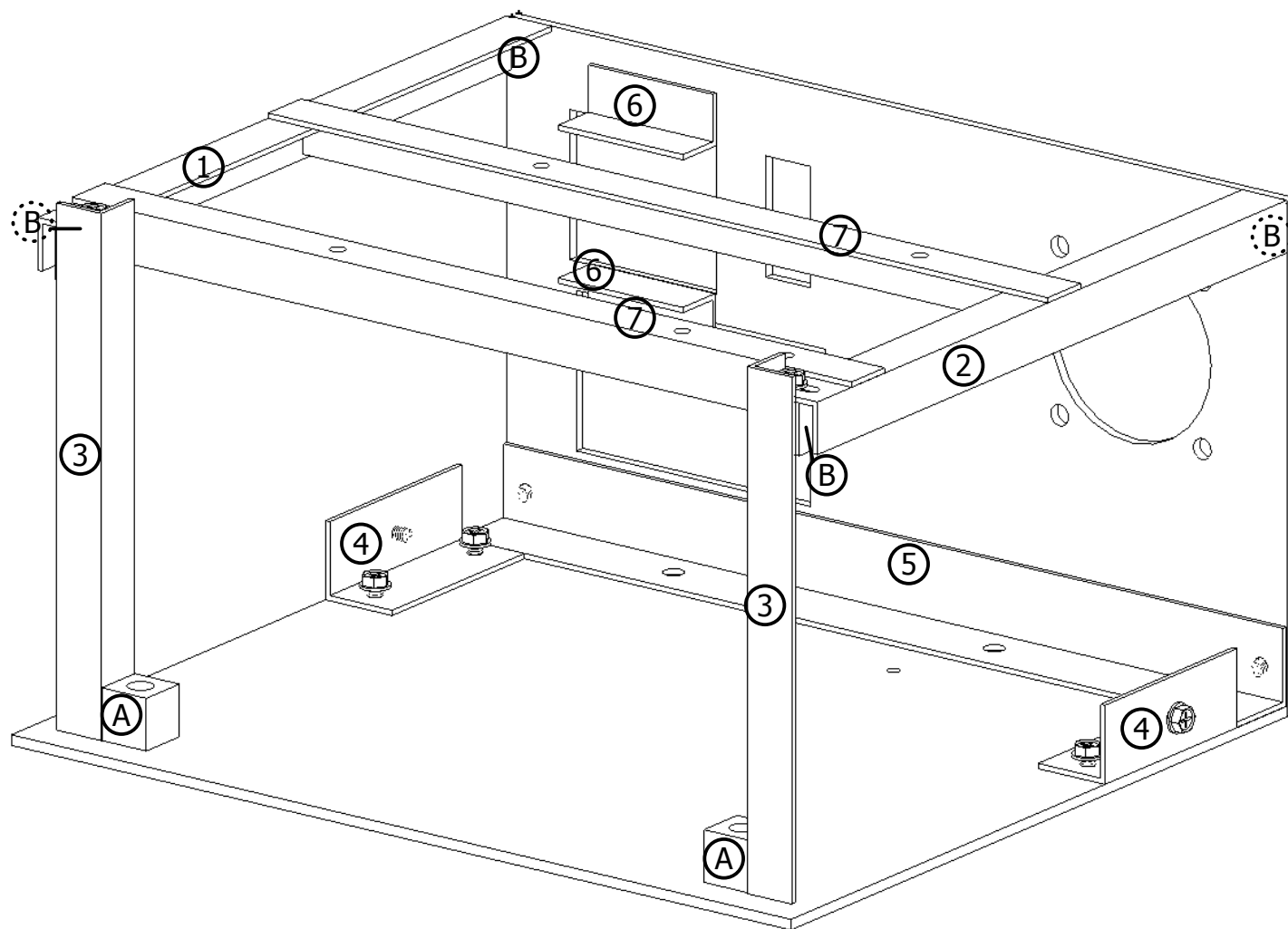
A  
B  
C  
D  
E  
F




	ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ		
	Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>		
Autor:	Bc. Michal Zecpal	Příloha:	A
Výkres:	Asistenční jednotka		
Datum:	3.1.2015	List č.:	<b>1 / 8</b>

1 2 3 3 4 5 6 6 7

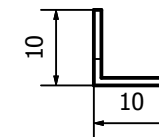
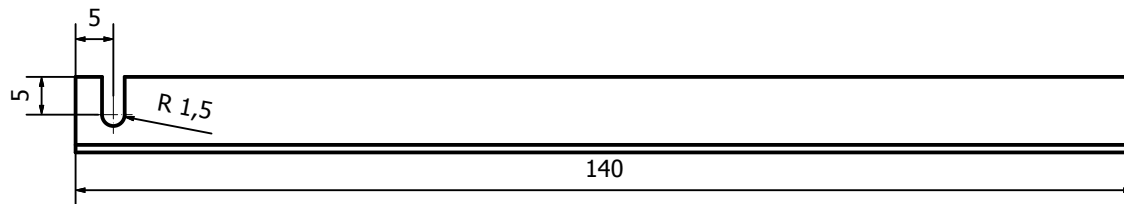





 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY  
 FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
 VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	A
Výkres:	3D model - popis		
Datum:	19.12.2014	List č.:	<b>2 / 8</b>

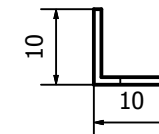
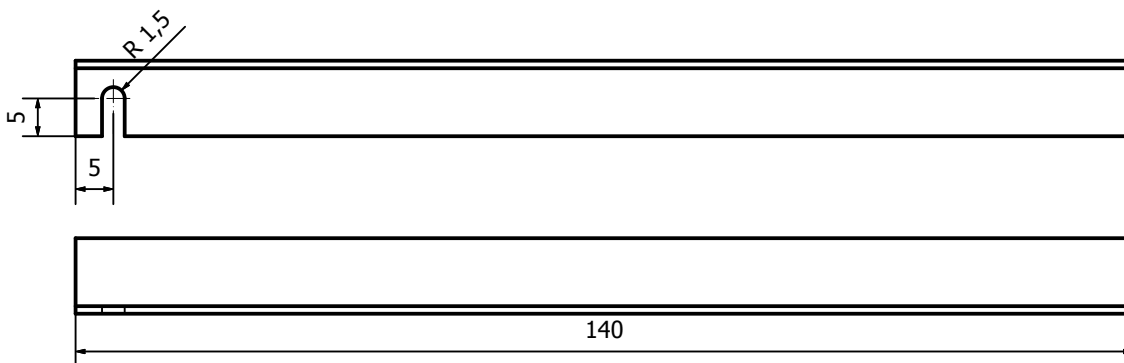
1



1 ks

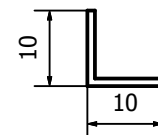
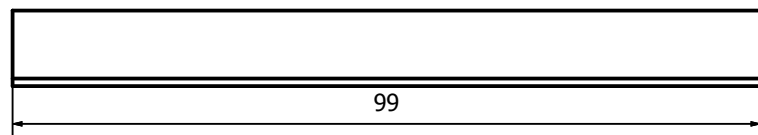
"L" profily  
tloušťka: 1mm  
materiál: hliník

2



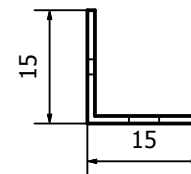
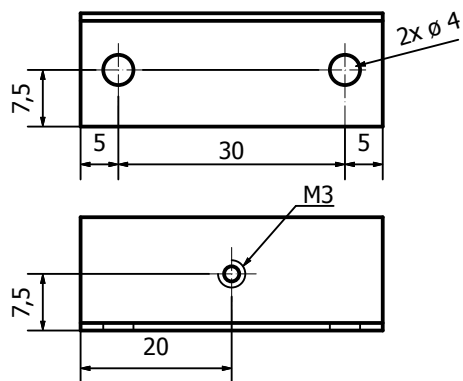
1 ks

3



2 ks

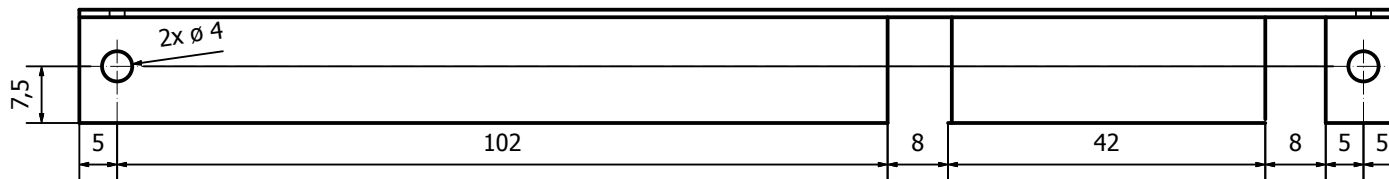
4



2 ks

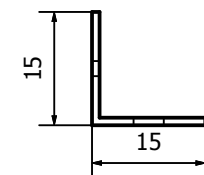
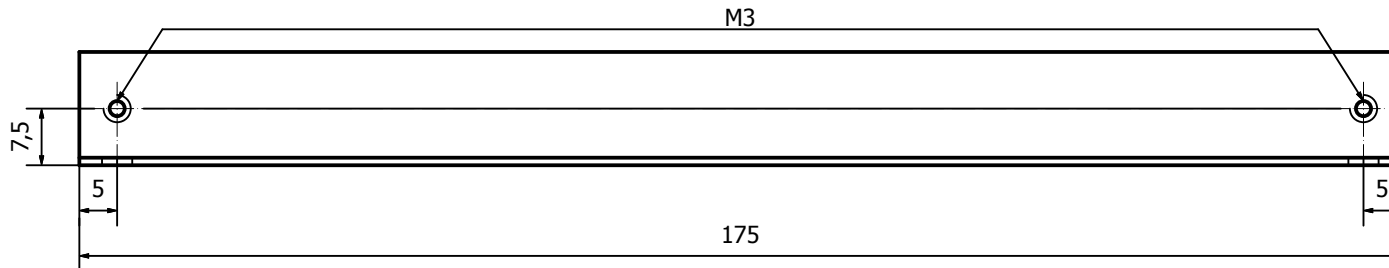
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	A
Výkres:	Rozměry "L" profilů	Měřítko:	1 : 1
Datum:	19.12.2014	List č.:	<b>3 / 8</b>

5



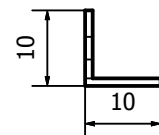
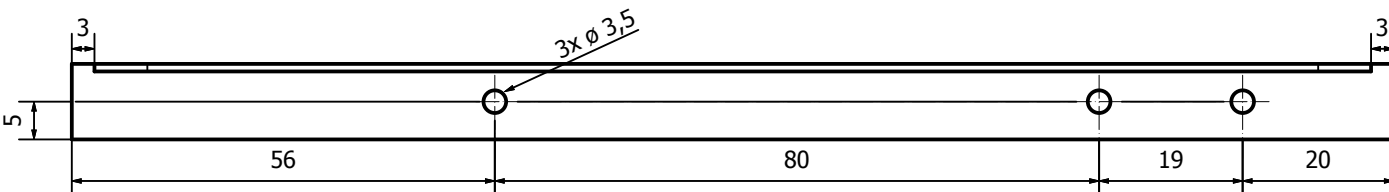
"L" profily  
tloušťka: 1mm  
materiál: hliník

6

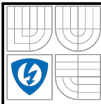


1 ks

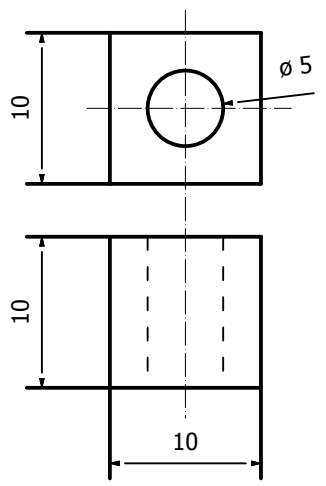
7



2 ks

 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	A
Výkres:	Rozměry "L" profilů	Měřítko:	1 : 1
Datum:	27.12.2014	List č.:	<b>4 / 8</b>

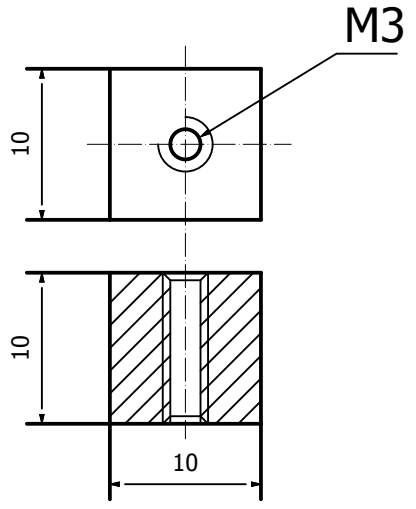
(A)



2 ks

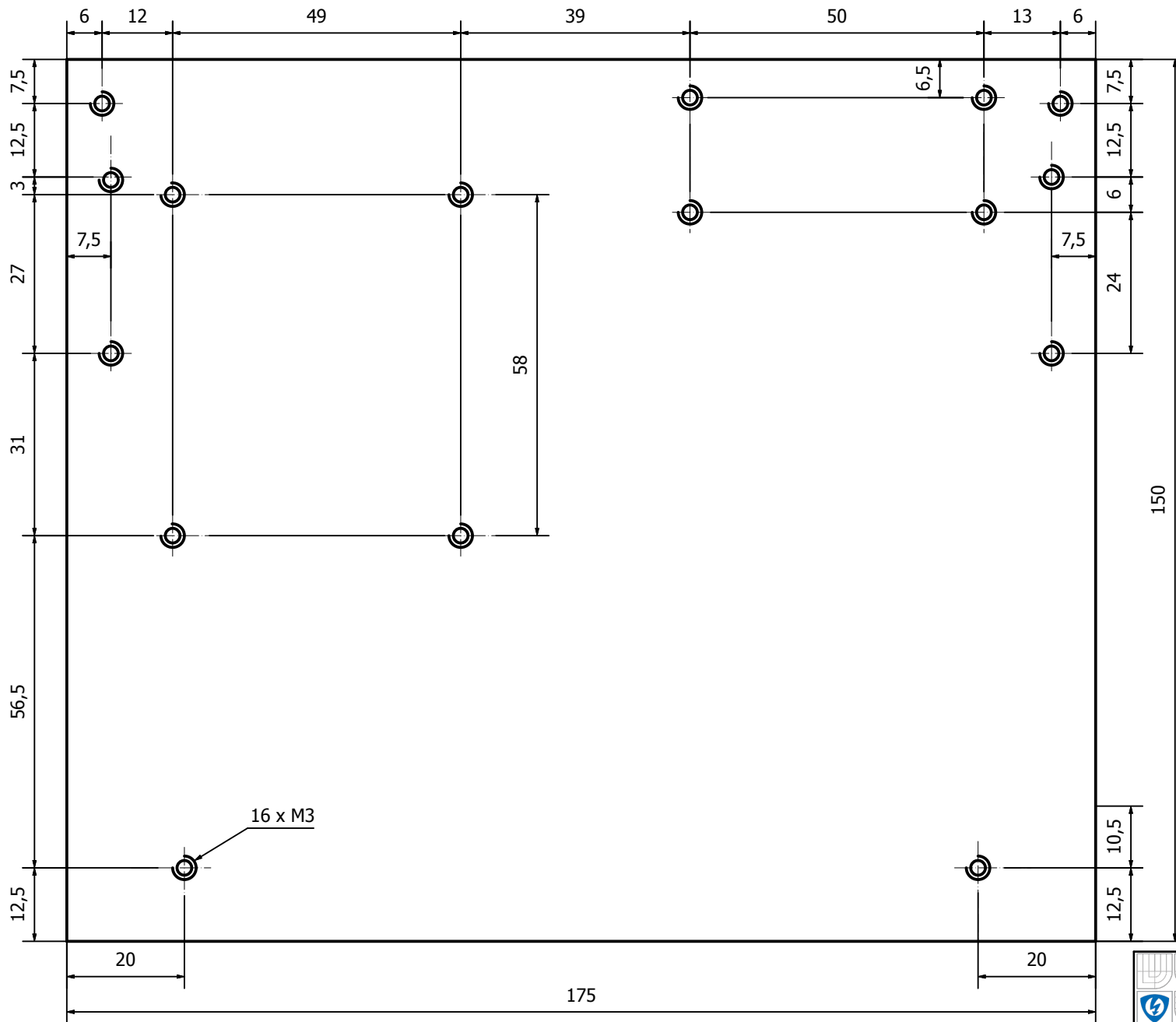
materiál: dural

(B)




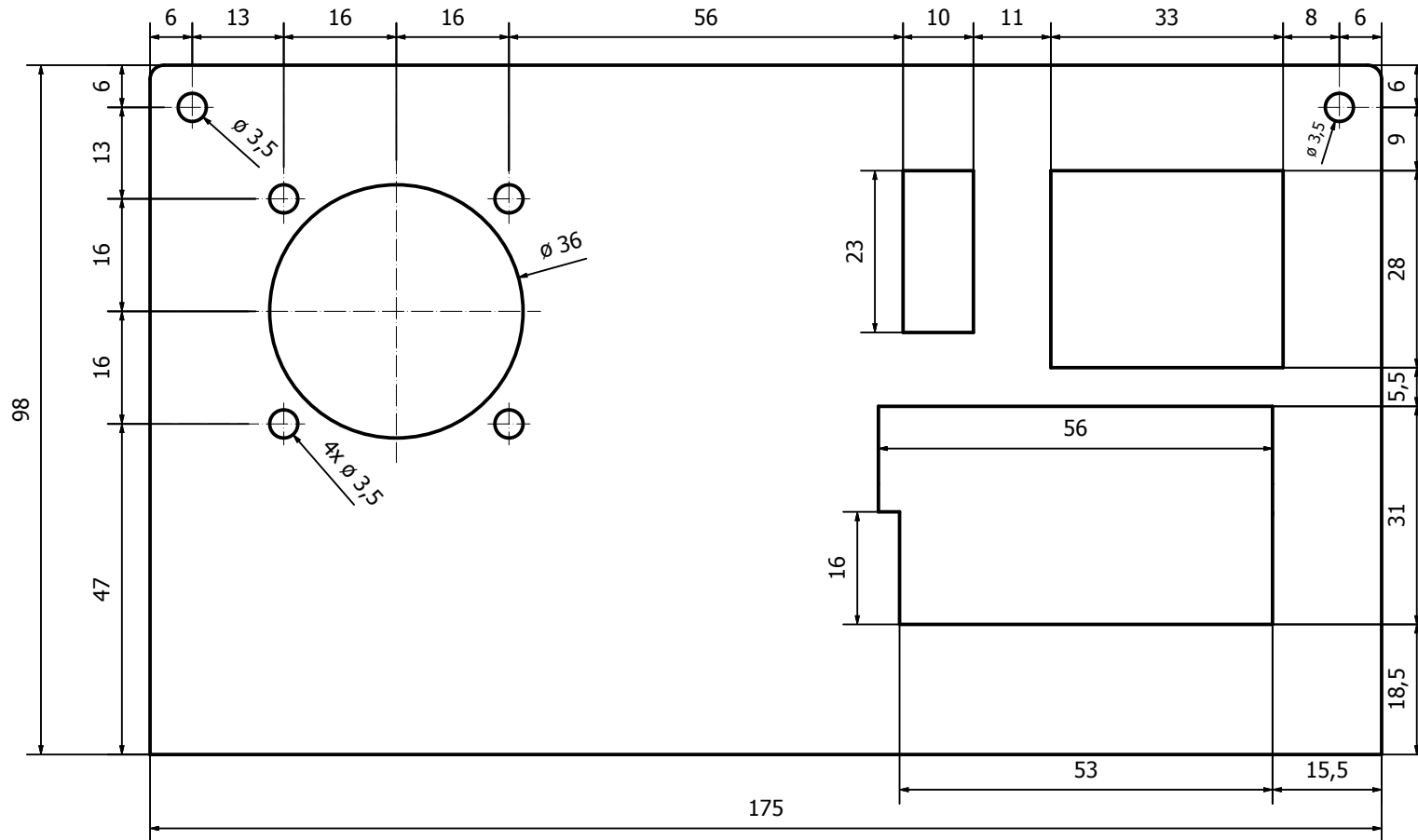
4 ks

	ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ		
	Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>		
Autor:	Bc. Michal Zecpal	Příloha:	A
Výkres:	Spojovací kostky	Měřítko:	1 : 0,5
Datum:	3.1.2015	List č.:	<b>5 / 8</b>




Plech  
tloušťka: 2mm  
materiál: hliník

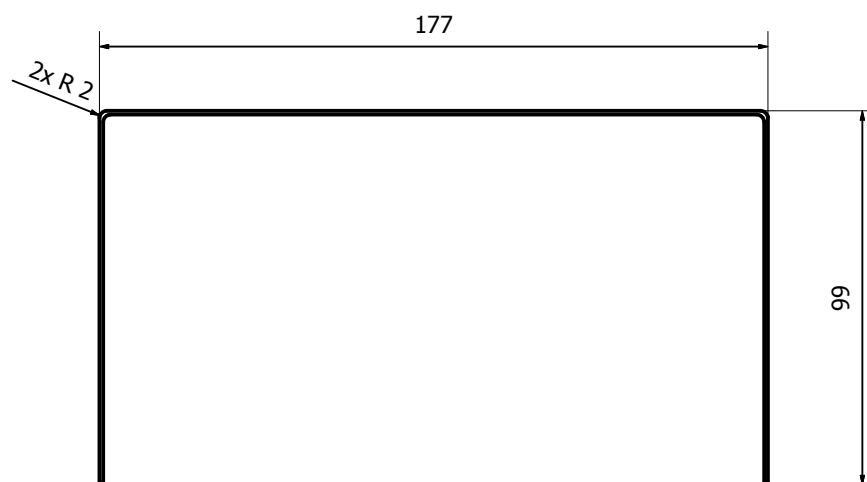
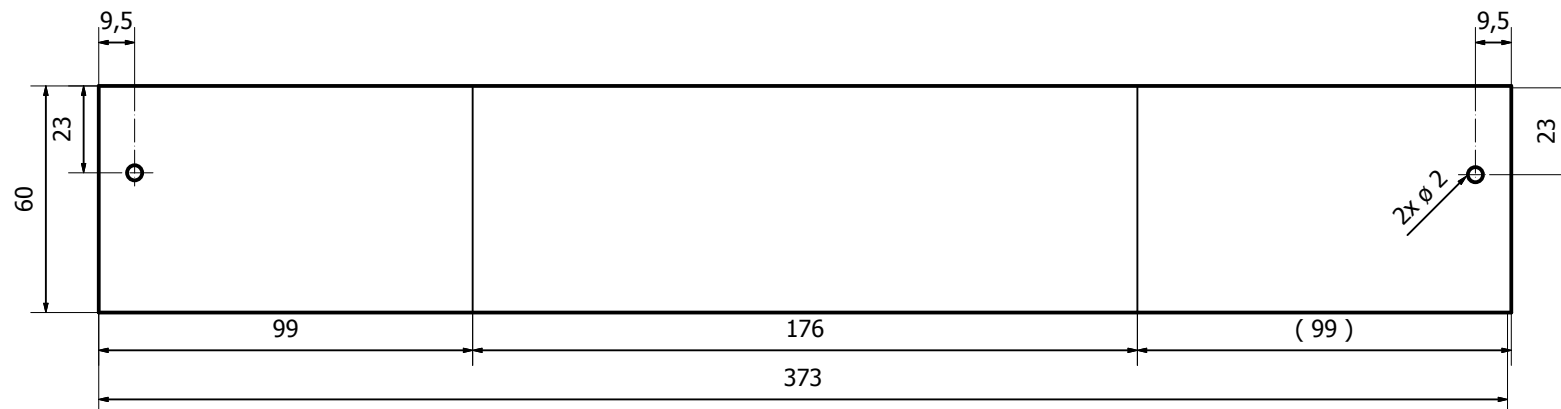
 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	A
Výkres:	Základní deska	Měřítko:	1 : 1
Datum:	27.12.2014	List č.:	<b>6 / 8</b>




Plech  
tloušťka: 2mm  
materiál: hliník

 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	A
Výkres:	Zadní deska	Měřítko:	1 : 1
Datum:	26.12.2014	List č.:	<b>7 / 8</b>

Plech  
tloušťka: 1mm  
materiál: hliník

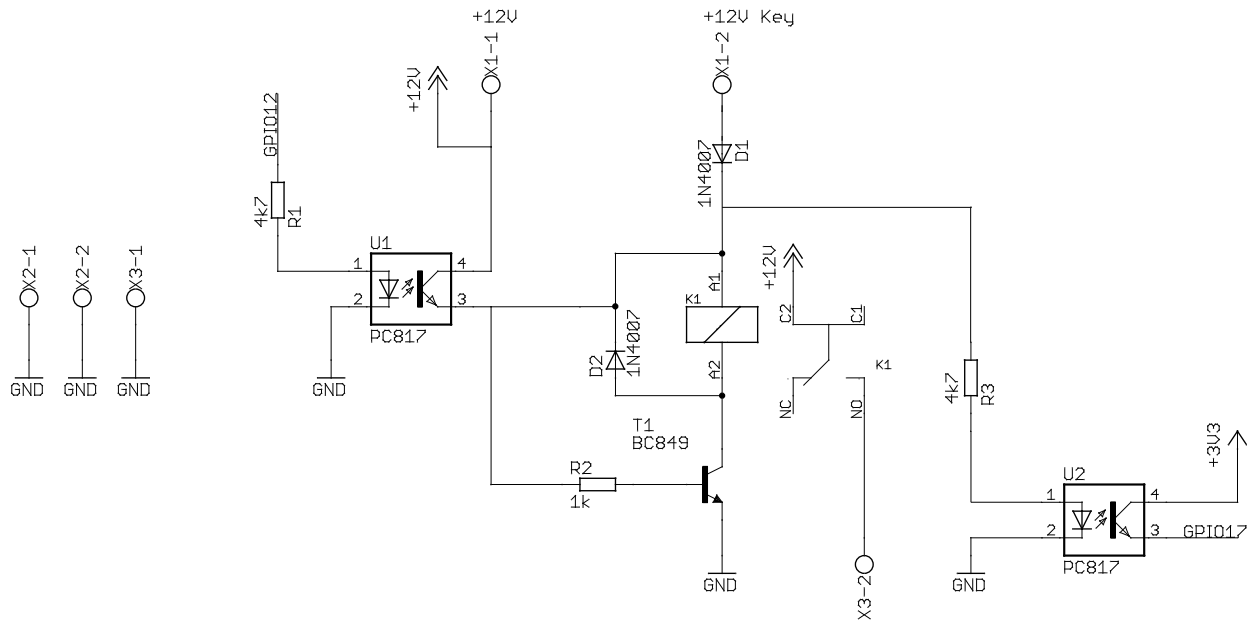


	ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ		
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	A
Výkres:	Plechový kryt	Měřítko:	1 : 2
Datum:	7.12.2014	List č.:	<b>8 / 8</b>

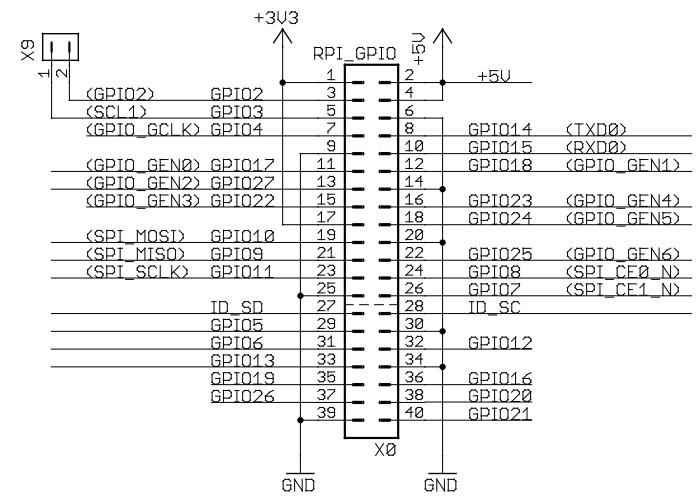




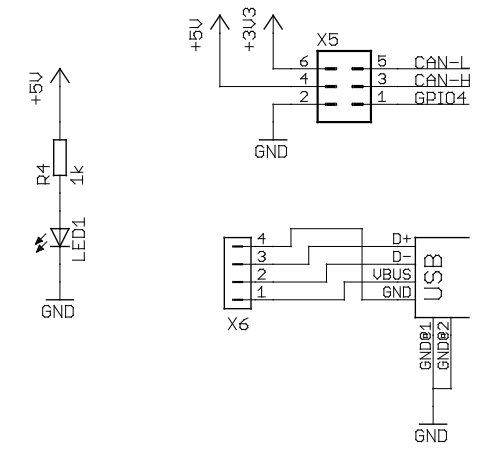
# Napájecí část



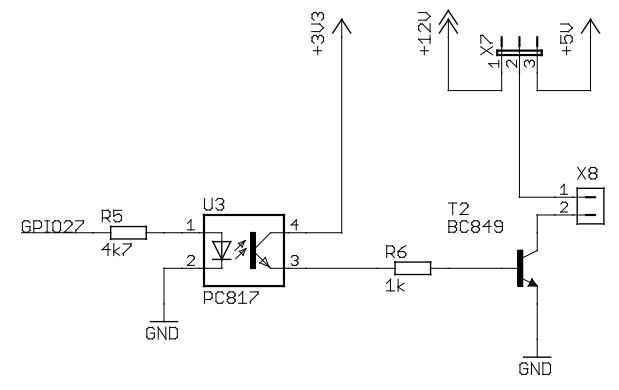
# Raspberry PI2 B+ GPIO konektor




# Informačně komunikační část



# Akustická signalizace



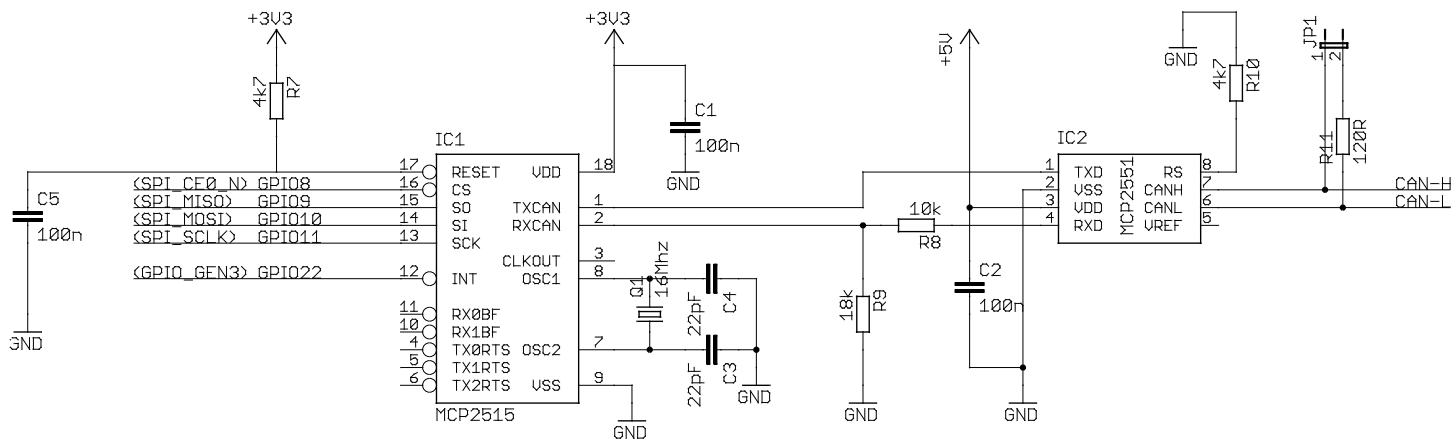


**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**  
**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ**  
**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

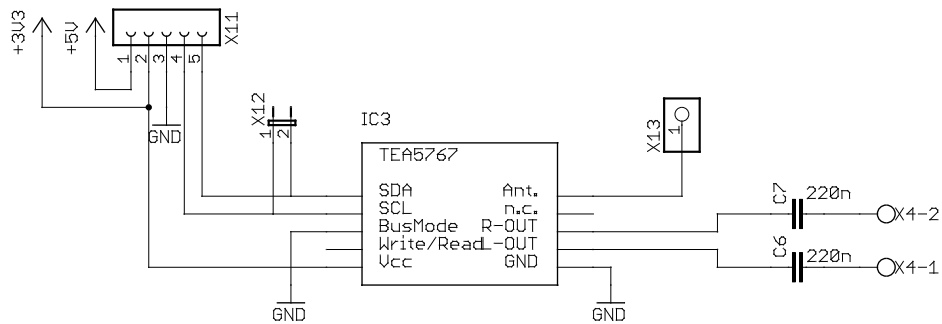
Název práce: **Monitorovní dopravní situace s využitím Raspberry PI**

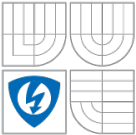
Autor:	Bc. Michal Zacpal	Příloha:	C
Výkres:	Rozšiřující jednotka 1/2		
Datum:	19.4.2015	List č.:	<b>1 / 8</b>

# SPI / CAN

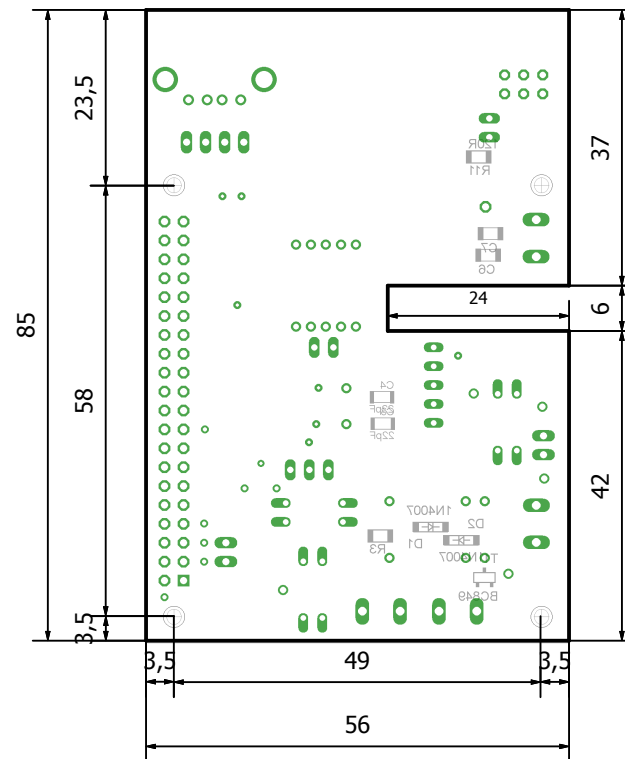
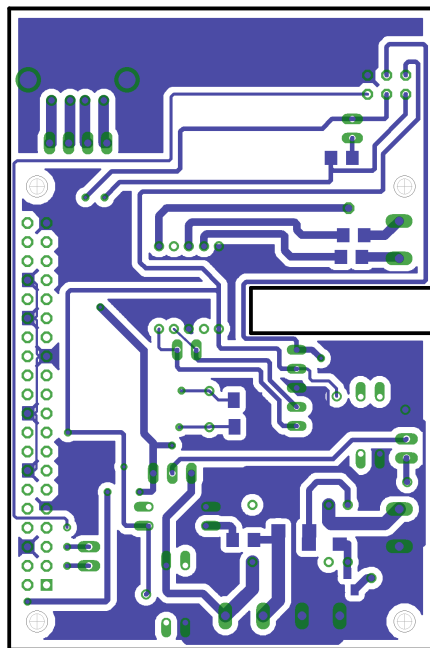



# FM Radio

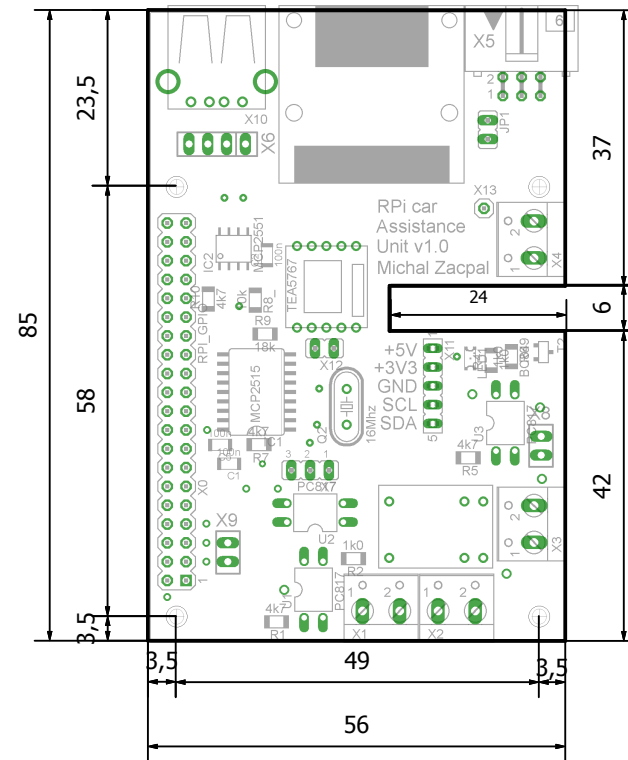
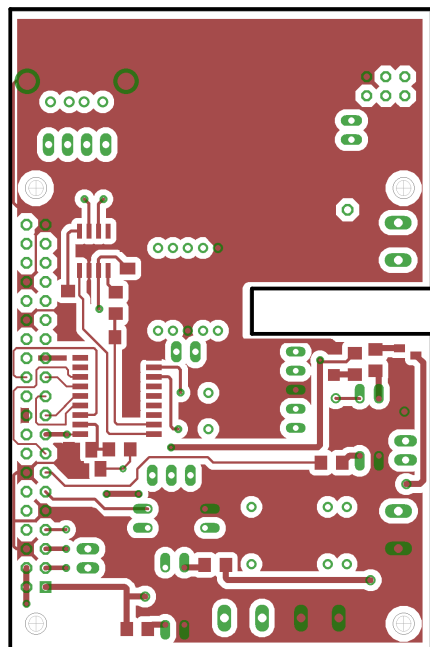




**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**  
**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ**  
**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

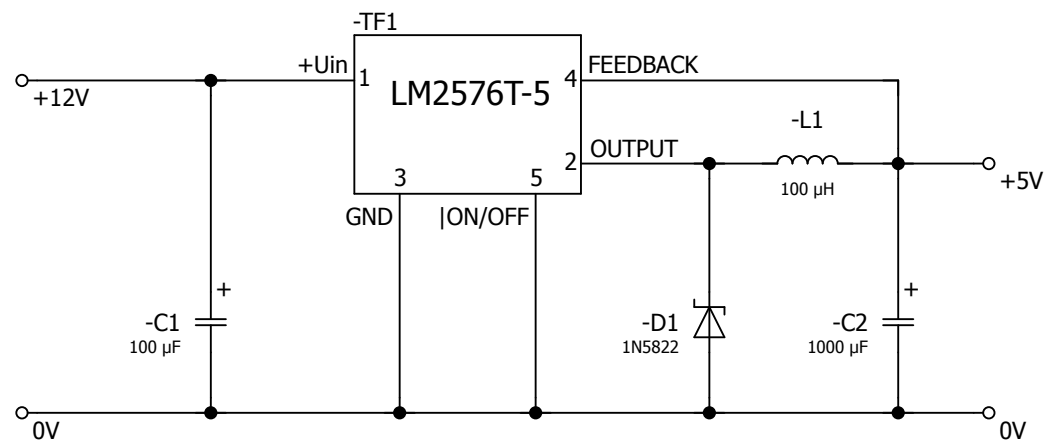
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zacpal	Příloha:	C
Výkres:	Rozšiřující jednotka 2/2		
Datum:	19.4.2015	List č.:	<b>2 / 8</b>




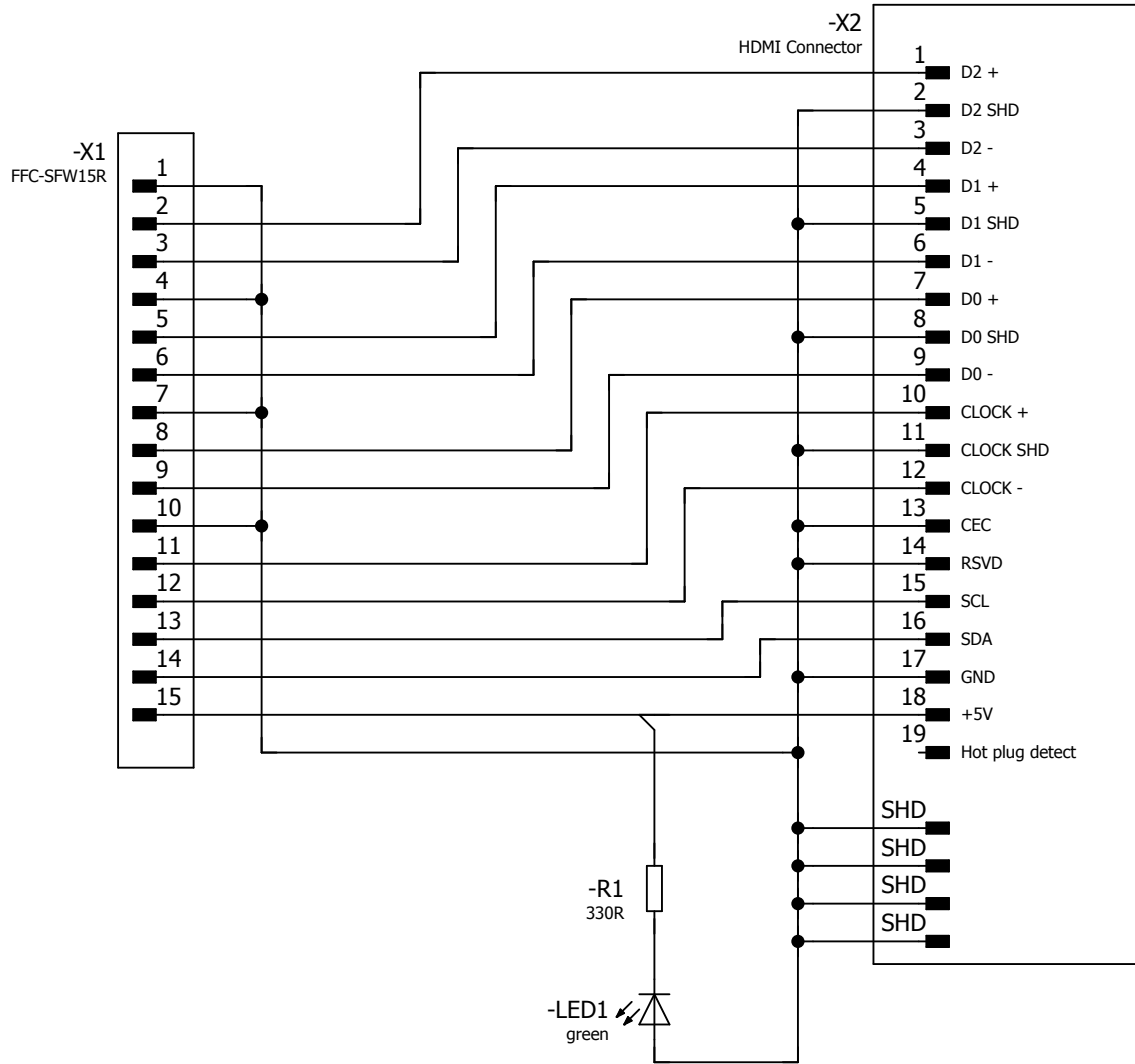
 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	C
Výkres:	Rozšiřující jednotka - Návrh DPS - bottom		
Datum:	12.5.2015	List č.:	<b>3 / 8</b>




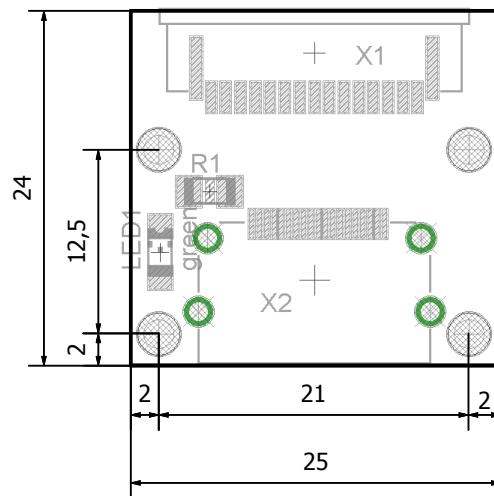
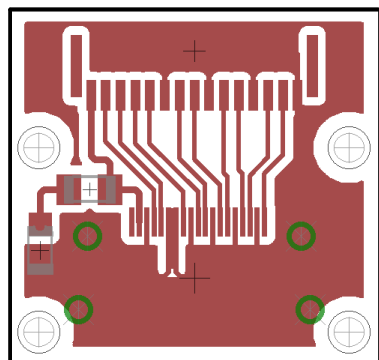
 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zaccpal	Příloha:	C
Výkres:	Rozšiřující jednotka - Návrh DPS - top		
Datum:	12.5.2015	List č.:	<b>4 / 8</b>




 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	C
Výkres:	Stabilizátor 5V		
Datum:	2.12.2014	List č.:	<b>5 / 8</b>

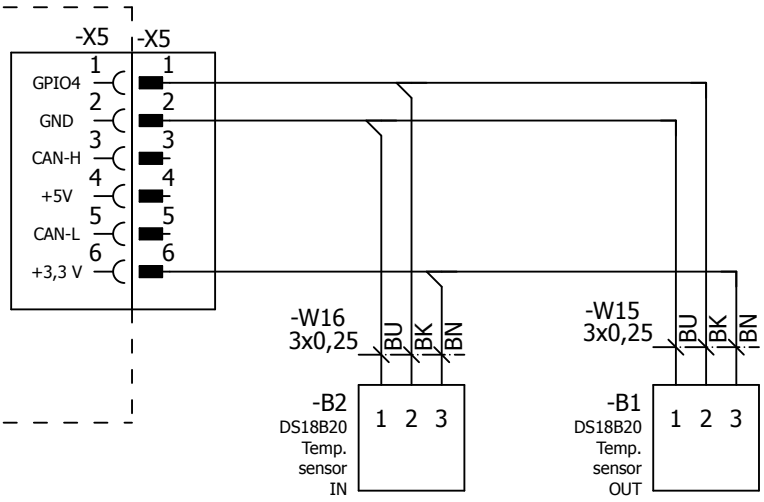



 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	C
Výkres:	Cam extender		
Datum:	1.5.2015	List č.:	<b>6 / 8</b>



	ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ		
	Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>		
Autor:	Bc. Michal Zecpal	Příloha:	C
Výkres:	Cam extender - Návrh DPS		
Datum:	13.5.2015	List č.:	<b>7 / 8</b>


-A2  
Rozšiřující deska



 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovní dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	C
Výkres:	Zapojení snímačů DS18B20		
Datum:	7.5.2015	List č.:	<b>8 / 8</b>



Označení:	Popis:	Typ:
A1	Raspberry Pi 2	Model B+ rev 1.x
A2	Rozšiřující deska	v1.0
A3	Řídicí jednotka pro displej	VS-TY2662-V1
A3.1	Ovladač k nastavení displeje	-
A4	Display	AT070TN92
A5	USB převodník pro dotykovou vrstvu displeje	-
A6.1	Camera extender	v1.1
A6.2	Camera extender	v1.1
B1	Snímač venkovní teploty	DS18B20
B2	Snímač vnitřní teploty	DS18B20
CAM1	Kamera pro monitorování dopravní situace před vozidlem	Raspberry Pi camera Rev 1.3
CAM2	Kamera pro monitorování dopravní situace za vozidlem	Logitech C310
F1	Pojistka + pojistkový držák	10A + KS-SH-FLA3
H1	Sirénka	KXG1212C
M1	Ventilátor	SUNON MB40201VX-F99
TF1	Stabilizátor napětí 5VDC	LM2576T-5
XS1	Připojovací konektor	ISO 10487

 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorování dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	D
Výkres:	Soupis položek		
Datum:	28.4.2015	List č.:	<b>1 / 2</b>

## Rozšiřující modul

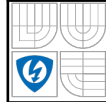
Označení:	Hodnota:	Součástka:	Pouzdro:
C1	100n	CON 1206	C1206
C2	100n	CON 1206	C1206
C3	22pF	CON 1206	C1206
C4	22pF	CON 1206	C1206
C5	100n	CON 1206	C1206
C6	220nF	CON 1206	C1206
C7	220nF	CON 1206	C1206
D1	1N4007	Dioda 1N4007	SMA
D2	1N4007	Dioda 1N4007	SMA
IC1	MCP2515	MCP2515-I/SOGRB2SO18L	SO18L
IC2	MCP2551	MCP2551/SN	SO08
IC3	TEA5767	TEA5767	TEA5767
JP1	-	JP1E	JP1
K1	12V0,2W	RELEM3S12T	N4100
LED1	green	SML1206	1206
Q1	16Mhz	CRYSTALHC49S	HC49/S
R1	4k7	RES 1206	R1206
R2	1k	RES 1206	R1206
R3	4k7	RES 1206	R1206
R4	1k	RES 1206	R1206
R5	4k7	RES 1206	R1206
R6	1k	RES 1206	R1206
R7	4k7	RES 1206	R1206
R8	10k	RES 1206	R1206
R9	18k	RES 1206	R1206
R10	4k7	RES 1206	R1206
R11	120R	RES 1206	R1206
T1	BC849	TRAN BC849	SOT23
T2	BC849	TRAN BC849	SOT23
U1	PC817	OPTO PC817	DIL04
U2	PC817	OPTO PC817	DIL04
U3	PC817	OPTO PC817	DIL04
X0	2x20pin	BL840GD	BL840GD
X1	-	SVORK ARK300V	ARK300V-2P
X2	-	SVORK ARK300V	ARK300V-2P
X3	-	SVORK ARK300V	ARK300V-2P
X4	-	SVORK ARK300V	ARK300V-2P
X5	2x3pin	MLW06A	MLW06A
X6	-	HDR-1X4-SIP-100	SIP-100-4V
X7	-	JP2E	JP2
X8	-	HDR-1X2-SIP-100	SIP-100-2V
X9	-	HDR-1X2-SIP-100	SIP-100-2V
X10	USB 2.0	USB-A-H	USB1X90
X11	-	FE05-1	FE05-1
X12	-	PROPOJKA	JP1
X13	-	PIN 1X1	1X01

## Stabilizátor 5V

Označení:	Hodnota:	Součástka:	Pouzdro:
C1	100 uF / 25V	Kondenzátor elektrolyt.	Elektrolyt. radiální
C2	1000 uF / 25V	Kondenzátor elektrolyt.	Elektrolyt. radiální
D1	1N5822	Schotkyho dioda	DO201AD
L1	100 uH	Tlumivka	Toroidní vertikální
TF1	5V	Stabilizátor napětí	TO220-5

## Cam Extender

Označení:	Hodnota:	Součástka:	Pouzdro:
LED1	green	SML1206	1206
R1	330R	RES 1206	R1206
X1	-	CONN-FFC-SFW15R	SFW15R
X2	-	CONN-HDMI	10029449-001RLF

 ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ			
Název práce: <b>Monitorovni dopravní situace s využitím Raspberry PI</b>			
Autor:	Bc. Michal Zecpal	Příloha:	D
Výkres:	Seznam součástek		
Datum:	1.5.2015	List č.:	<b>2 / 2</b>