



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**NÁVRH ŘÍDÍCÍHO SYSTÉMU PRO MODEL VOZIDLA
POHYBUJÍCÍHO SE V AUTONOMNÍM KONVOJI**

SINGLE BOARD COMPUTER BASED CONTROL DESIGN FOR MODEL OF AUTONOMOUS
CONVOY VEHICLE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Maxmilián Kvoch

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Stanislav Věchet, Ph.D.

BRNO 2017

ZADÁNÍ VŠKP 1

(tento list nahradíte oficiálním zadáním práce)

ABSTRAKT

Tato bakalářská práce se zabývá návrhem řídicího systému pro vozidlo jedoucí v autonomním konvoji. Řídicí systém slouží k ovládání vozidla na základě příkazů z vyšší řídicí jednotky. Nízko-úrovňová řídicí jednotka bude příkazy přijímat a ovládat aktuátory řízení a pohonu pomocí dvou PWM signálů. Nízko-úrovňová řídicí jednotka bude zpětně odesílat naměřená data o stavu baterie a vzdálenosti z ultrazvukového senzoru. Komunikace mezi nadřazenou a podřízenou řídicí jednotkou bude realizována pomocí rozhraní UART

KLÍČOVÁ SLOVA

Autonomní konvoj, ultrazvukový sonar, USART, PWM, MBED, ARDUINO, Raspberry Pi

ABSTRACT

This bachelor thesis is about development of controlling system for vehicle in autonomous convoy. Controlling system controls movement of the vehicle. Low-level controll unit takes commands from high-level unit and controls actuators of the vehicle using two PWM signals. Low-level unit also measures battery level and range from ultrasonic sensor. Measured data are sent to high-level unit. Control units are communicating through USART.

KEYWORDS

Autonomous convoy, ultrasonic range sensor, USART, PWM, MBED, ARDUINO, Raspberry Pi

BIBLIOGRAFICKÁ CITACE

KVOCH, M. *Návrh řídicího systému pro model vozidla pohybujícího se v autonomním konvoji*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 61 s.
Vedoucí bakalářské práce doc. Ing. Stanislav Věchet, Ph.D

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucímu práce, doc. Ing. Stanislavu Věchetovi, Ph.D za cenné rady a konzultace během vypracování práce. Dále chci poděkovat mým Rodičům za jejich podporu, kterou mi poskytovali během studia.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením doc. Ing. Stanislav Věcheta, Ph.D a s použitím literatury uvedené v seznamu literatury.

V Brně dne 25. 5. 2017

.....

Maxmilián Kvoch

OBSAH

1	ÚVOD.....	15
2	EMBEDDED SYSTÉMY VE SVĚTĚ	17
2.1	ARDUINO	19
2.1.1	Vývojové prostředí Arduino	20
2.1.2	Arduino UNO	21
2.1.3	Mikroprocesor ATmega328P	22
2.1.4	Arduino Shieldy.....	23
2.1.5	Programování vývojových desek ARDUINO	24
2.1.6	Přehled vývojových desek Arduino.....	24
2.2	MBED.....	26
2.2.1	Online vývojové prostředí MBED	26
2.2.2	mbed NXP LPC1768	27
2.2.3	Mikroprocesor LPC 1738	28
2.2.4	Přehled vývojových desek MBED	29
2.3	RASPBERRY PI	30
2.3.1	Verze a provedení počítače Raspberry pi	30
2.3.2	Raspberry Pi model 3	31
2.3.3	Operační systém	32
3	NÁVRH ŘÍDÍCIHO SYSTÉMU	33
3.1	Robotic Operating System	34
3.2	Požadavky na řídicí systém	34
3.3	Návrh struktury řídicího systému	35
3.3.1	Vysokourovňová řídicí jednotka	35
3.3.2	Nízkoúrovňová řídicí jednotka	36
3.3.3	USART	37
4	REALIZACE EXPERIMENTÁLNÍHO VOZIDLA.....	39
4.1	EXPERIMENTÁLNÍ MOBILNÍ PLATFORMA	39
4.1.1	Ackermanův podvozek	39
4.1.2	Aktuátor natočení přední nápravy	40
4.1.3	Stejnoseměrný motor.....	41
4.1.4	Regulátor Stejnoseměrného motoru	41
4.1.5	Pulzně šířková modulace	42
4.2	SENZORY	44
4.2.1	Senzor pro měření vzdálenosti	44
4.2.2	Měření stavu akumulátoru	45
4.2.3	Kamera.....	46
4.3	FIRMWARE PRO ŘÍDÍCI JEDNOTKU MBED	47
4.3.1	Popis komunikačního protokolu	47
4.3.2	Komunikace s nadřazenou jednotkou	47
4.3.3	Získávání dat ze senzorů.....	48
4.3.4	Ovládání aktuátorů	49
5	TESTOVÁNÍ PROTOTYPU VOZIDLA	51
6	ZÁVĚR	53

7	SEZNAM POUŽITÉ LITERATURY	55
	SEZNAM OBRÁZKŮ.....	57
	SEZNAM TABULEK	58
	SEZNAM PŘÍLOH	59
	PŘÍLOHY	60

1 ÚVOD

Autonomní konvoj je takové uspořádání vozidel, kde první vozidlo konvoje je řízeno člověkem a všechna další vozidla jedoucí v koloně, jsou řízena automatickými řídicími systémy. Autonomní vozidla používají ke snímání okolního prostředí data ze senzorů, pomocí kterých sledují vozidlo před sebou i výskyt případných překážek na cestě. Typem snímačů mohou být radarové systémy, kamery, laserové dálkoměry nebo sonary. Vozidla také komunikují mezi sebou a využívají GPS k určování polohy.

Autonomní konvoje naleznou uplatnění především v silniční dopravě. Kolona 3-5 kamionů jedoucích za sebou v těsném rozestupu dokáže ušetřit spoustu paliva tím, že všechna vozidla konvoje jedou ve společném větrném vaku. Dalším možným využitím je skupina spolupracujících mobilních robotů, která by se mohla přesouvat mezi pracovními stanovišti v autonomním konvoji. Abychom byli schopni autonomní konvoj realizovat, musíme nejprve prozkoumat vhodné řídicí systémy a z nich vybrat systém, na kterém budeme stavět. V dnešní době existuje široký výběr vestavěných počítačů vhodných k realizaci řídicího systému.

Nejdříve se seznámíme s aktuálně používanými jednodeskovými počítači a jejich vlastnosti budou porovnány na několika příkladech. Tato práce se bude dále zabývat rozbořem dvou vývojových platform Arduino a mbed, které budou popsány v kapitolách 2.1. a 2.2. O návrhu struktury řídicího systému bude pojednávat kapitola 3. Následovat bude praktická část, ve které se budeme věnovat konstrukci autonomního vozidla. Platformu robotického systému bude tvořit rádiem řízený model offroad vozidla 1:10. Rádiové řízení bude nahrazeno novým řídicím systémem, založeným na počítači Raspberry Pi a subsystému s mikropočítačem mbed. Oba systémy budou propojeny komunikačním rozhraním USART.

V kapitole 4. bude popsán způsob řízení aktuátorů vozidla pomocí PWM signálů i způsob měření vzdálenosti pomocí ultrazvukového sonaru. Vozidlo bude také vybaveno automatickou diagnostikou v podobě měření stavu akumulátoru. Pro mikropočítač mbed bude nutné naprogramovat firmware, který obstará výše uvedené činnosti. Vývoj firmware bude popsán v kapitole 4.3. Aby byla prokázána funkčnost vozidla, bude vozidlo otestováno.





2 EMBEDDED SYSTÉMY VE SVĚTĚ

Embedded systém je druh počítače, který je zabudován do stroje nebo zařízení a podílí se na jeho řízení. Na rozdíl od osobních počítačů, jsou embedded počítače většinou jednoúčelové a slouží k vykonávání předem určených činností. Často bývají využity k řízení procesů v reálném čase, z čehož plynou vysoké nároky na odezvu, spolehlivost a robustnost. Vestavěný počítač je obvykle jednodeskový a je tvořen mikroprocesorem, který je jeho neoddělitelnou součástí. Vzhledem k tomu, že jednotlivé komponenty počítače jsou umístěny na jedné desce, odpadá použití slotů, které mohou být poruchové. Jednodeskové počítače jsou tedy oproti klasickým počítačům, spolehlivější, mají menší spotřebu energie a menší velikost. Nelze je však vylepšovat, např. výměnou paměti RAM za větší [1].

Na trhu je široké spektrum výrobců jednodeskových počítačů. Můžeme si tak zvolit řídicí systém přesně na míru požadavků aplikace. Tím dosáhneme snížení energetických nároků i výsledné ceny za hotové zařízení. V dnešní době se odvětví embedded systémů rychle rozvíjí, protože vznikají nové technologické trendy. Z nich můžeme jmenovat například internet věcí, samoříditelné automobily, 3D tiskárny, drony nebo nositelnou elektroniku. Tato zařízení se bez počítačového řízení neobejdou, protože se od nich předpokládá jistá míra inteligence nebo autonomního chování.

V tabulce 1. je uveden přehled aktuálně dostupných embeded systémů. Byly vybrány čtyři mikropočítače, které jsou zajímavé pro svoji příznivou cenu a vysoký výkon a malé rozměry. Počítač Raspberry Pi bude podrobněji popsán v kapitole 2.3. Konkurentem počítače Raspberry Pi je počítač Odroid-XU4, který byl navržen v Jižní Koreji a nabízí 8-jádrový procesor s architekturou ARM big.LITTLE. V jednom pouzdře obsahuje relativně pomalejší a úspornější jádra (LITTLE) s výpočetně výkonnějšími a na spotřebu náročnějšími jádry (big) [2]. Čínský počítač LattePanda je založen na procesoru Intel Atom X5 a je navržen pro operační systém Windows 10. Zajímavostí je, že přímo na desce počítače je integrován 8 bitový mikroprocesor ATmega32U4 programovatelný v prostředí Arduino.

Poslední vybraný počítač je BeagleBone Blue. Na rozdíl od předchozích počítačů je speciálně vybaven pro použití v robotice. Na jeho desce nalezneme konektory pro připojení nejpoužívanějších sběrnic, rotačních enkodérů a servomotorů. Chybí však HDMI rozhraní. Deska také obsahuje akcelerometr, atmosférický barometr a obvod pro napájení z li-po akumulátoru. BeagleBone Blue je založen na stejném procesoru jako BeagleBone Black Wireless a je s ním částečně kompatibilní. Blue je možné použít s operačním systémem ROS, Linux nebo Ardupilot, který je určen k řízení kvadrokoptéry.

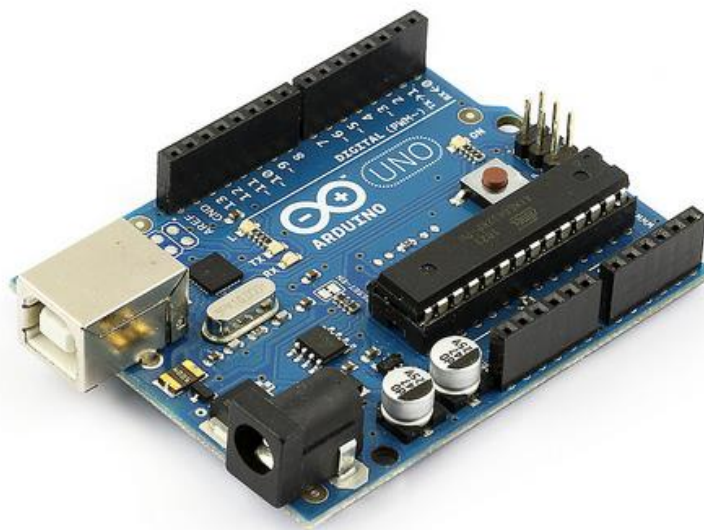
				
Název	Raspberry Pi 3 [3]	LattePanda [4]	BeagleBone® Blue [5]	ODROID-XU4 [6]
Procesor	Broadcom BCM2837 ARMv8	Intel Cherry Trail Z8350 (ATmega32u4 koprocessor)	Octavo Systems OSD3358 ARM Cortex-A8	Samsung Exynos5422 Cortex-A15 +Cortex-A7
Frekvence procesoru	1.2 GHz	1.44 GHz	1 GHz	2GHz
Počet jader	4	4	1	8
Grafická karta	VideoCore IV 3D graphics core	Intel HD Graphics	SGX530 graphics accelerator	Mali-T628 MP6
Paměť RAM	1 GB LPDDR2	4 GB DDR3L	512 MB DDR3	2 GB LPDDR3
Uživatelská paměť	SD karta (vyměnitelné)	64 GB eMMC	4 GB eMMC	eMMC 5.0 (vyměnitelné)
Vlastnosti	4 x USB 2.0 host 40 x GPIO HDMI Ethernet 3.5 mm Audio Jack Camera interface Display interface Bluetooth 4.1 Wi-Fi 802.11n	1 x USB 3.0 host 2 x USB 2.0 host 26 x GPIO HDMI a MIPI-DSI Ethernet Wi-Fi 802.11n Bluetooth 4.0 Arduino Leonardo SD karta	1x USB 2.0 client 1x USB 2.0 host Wi-Fi 802.11bgn Bluetooth 4.1 9-os. Akcelerometr Barometr 11 x LED 2 x tlačítka Li-Po akumulátor	2 x USB 3.0 host 1 x USB 2.0 host 42 x GPIO Ethernet HDMI RTC SD karta
Kompatibilní software	Debian, Ubuntu, Windows 10	Windows 10	Debian, ROS, Ardupilot	Ubuntu, Android
Rozměry	85 × 56 mm	88 x 70 mm	86 x 53 mm	83 x 58 mm
Hmotnost	45 g	55 g	40 g	38 g (bez chlad.)

Tabulka 1: Přehled aktuálně dostupných Embedded PC systémů

2.1 ARDUINO

Arduino vzniklo v Italském Interaction Design Intitute ve městě Ivrea a vyvíjí se od roku 2005. Cílem projektu bylo vytvořit levnou a snadno dostupnou vývojovou platformu pro studenty. Projekt byl natolik úspěšný, že se tvůrci rozhodli poskytnout ho celému světu s otevřenou licencí [7]. Arduino se skládá z vývojového prostředí (IDE) a hardware, kterým jsou jednodeskové počítače, z počátku založené na 8 bitových procesorech ATmega. Desky Arduino jsou rozšiřitelné o další moduly (shieldy) stejného formátu. Shieldy budou podrobněji popsány v kapitole 2.1.4.

Programovací jazyk Arduino je vysokoúrovňový a lehce srozumitelným, proto je dobrou volbou pro začínající uživatele. Při programování navíc není potřeba konfigurovat hardware mikroprocesoru, což zjednodušuje práci. Arduino se těší velké oblibě mezi amatéry i profesionály, kteří neustále rozšiřují Arduino o nové knihovny k periferiím a poskytují rady začátečníkům. Hlavními výhodami Arduino platformy je jednoduchost použití, obrovské množství kompatibilního hardware a nízká cena. Je možné setkat se i s označením „Genuino“, které je určené pro Arduino produkty na Evropském trhu. Na obrázku 1. Je vývojová deska Arduino UNO. Detailnímu popisu této desky bude věnována kapitola 2.1.2.



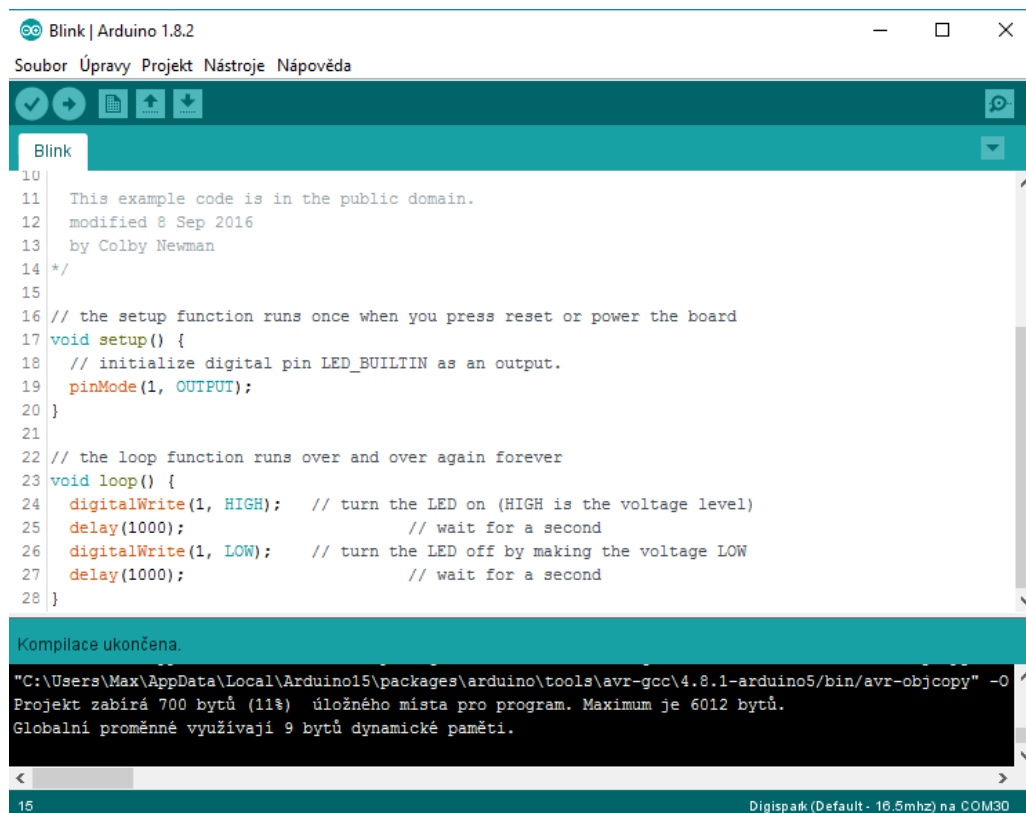
Obrázek 1: Vývojová deska Arduino UNO [8]

2.1.1 Vývojové prostředí Arduino

Programová část IDE vychází z prostředí Processing, což je programovací jazyk s vlastním editorem určený k výuce programování. Arduino je možné programovat v jazyce C/C++, ale nejvíce se používá programovací jazyk Wiring, který je někdy označován jako jazyk Arduino [7]. IDE je poměrně strohé a nenabízí uživateli příliš velký komfort, pro daný účel je však zcela postačující. Kromě barevného označení syntaxe neposkytuje software žádnou nápovědu při programování.

V horní liště najdeme základní tlačítka pro ověření kódu, nahrání kódu do Arduino zařízení a pro práci se soubory. V pravém horním rohu je ikona pro otevření sériového terminálu, který slouží jako základní nástroj při odladňování. Pokud chceme začít programovat, musíme nejdříve zvolit vybranou vývojovou desku v záložce nástroje. Poté už stačí jen vybrat správný komunikační port a nastavit frekvenci procesoru. Když máme program hotový, klikneme na tlačítko nahrát. Spustí se kompilátor a pokud nenalezne žádné chyby, dojde k nahrání programu do zařízení. Chybové hlášky a výstup z kompilátoru jsou vypisovány ve spodní části okna.

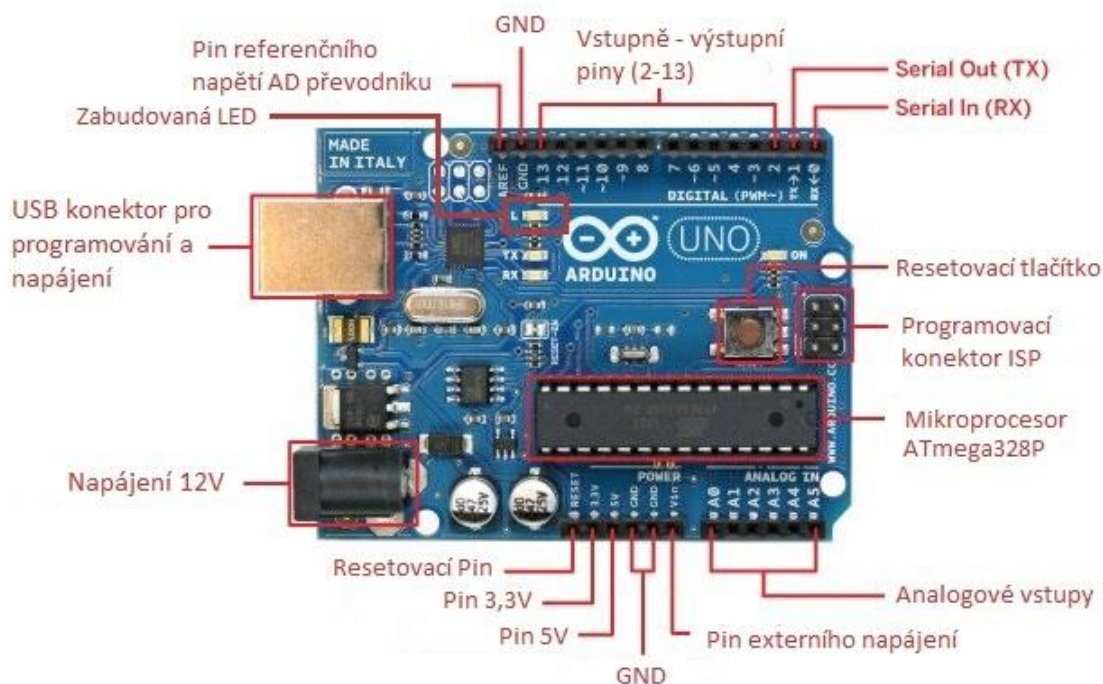
Arduino IDE má zabudovaný nástroj Boards Manager, kterým je možné přidávat další definice hardware a programovat i neoficiální vývojové desky. Potřebné informace bývají většinou dostupné na stránkách výrobce. V záložce „Soubor“ se dají nalézt ukázkové programy (Examples), které jsou přímo součástí aplikace. Na obrázku 2 je zachyceno okno IDE Arduino s ukázkovým programem „blikání LED diodou“.



Obrázek 2: Vývojové prostředí Arduino

2.1.2 Arduino UNO

Arduino UNO je jedna z nejjednodušších a mezi uživateli nejoblíbenějších vývojových desek. Technické parametry jsou uvedeny v tabulce 2. Deska se skládá z těchto hlavních částí: Mikroprocesor ATmega328P, USB-USART převodník, obvod umožňující nepřetržité napájení z více zdrojů, stabilizátor napětí, vstupně výstupní konektory a resetovací tlačítko. Mikroprocesor ATmega328P bude podrobněji popsán v kapitole 2.1.2. Popis desky Arduino UNO je na obrázku 3.



Obrázek 3: Popis desky Arduino UNO

Parametr	Hodnota
Mikroprocesor	ATmega328P
Napětí mikroprocesoru	5 V
Napájecí napětí (doporučené)	7-12 V
Napájecí napětí (limitní)	6-20 V
Digitální I/O Piny	20
PWM I/O Piny	6
Analogové vstupy	6
Max. povolený proud na I/O Pin	20 mA
Max. povolený proud na 3.3 V Pin	50 mA
Rozměry	68,6 x 58,4 mm
Flash Paměť	32 KB (ATmega328P), 0.5 kB zabírá bootloader

Tabulka 2: Technické parametry Arduino UNO [9]

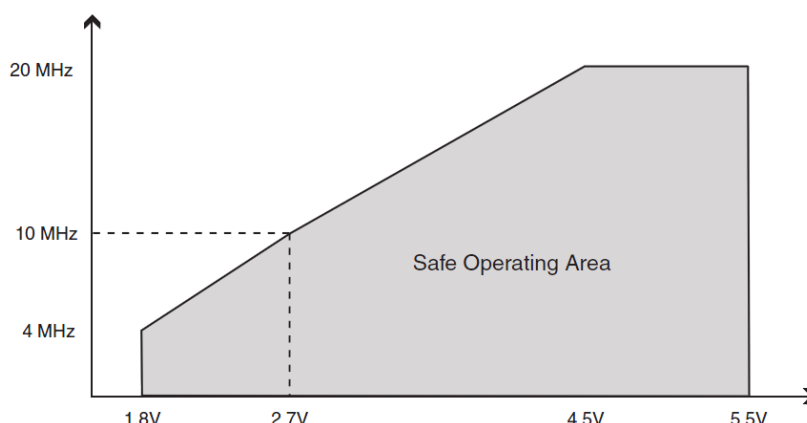
2.1.3 Mikroprocesor ATmega328P

Mikroprocesor ATmega328P je vyráběn firmou Atmel a spadá do rodiny ATmega, což jsou 8 bitové jednočipové mikroprocesory s architekturou AVR a redukovanou instrukční sadou (RISC). Důležité parametry mikroprocesoru jsou uvedeny v tabulce 3. Mikroprocesor dále obsahuje programovatelný Watchdog čítač, programovatelný interní oscilátor a 5 nastavitelných úsporných režimů. Čip dosahuje přibližného výkonu 1 MIPS/MHz. [10] Graf závislosti maximální taktovací frekvence na operačním napětí je na obrázku 4. Blokové schéma vnitřních periférií mikroprocesoru je na obrázku 5.

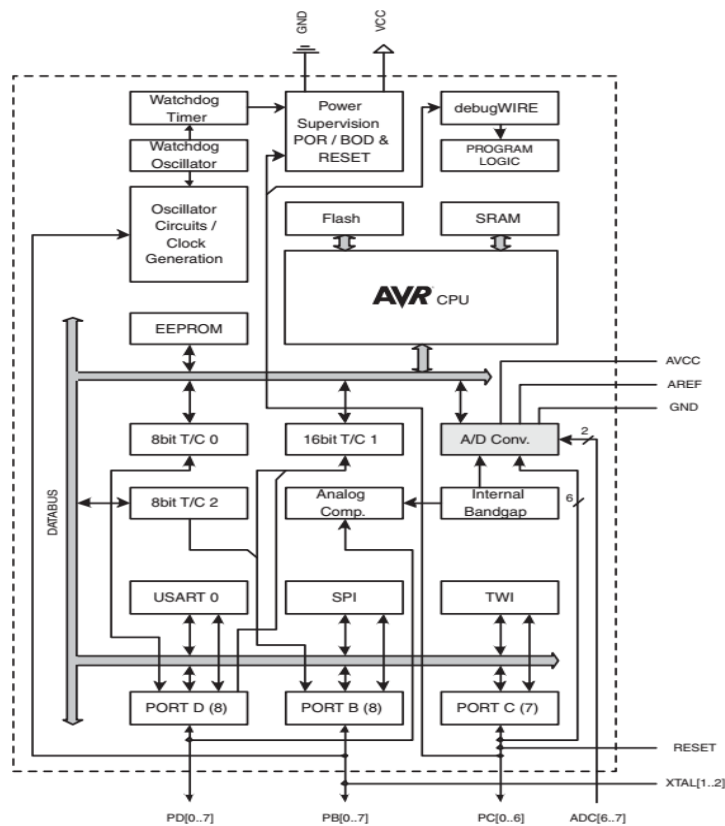
Parametr	Hodnota
Architektura CPU	8 bit AVR
Max. výkon procesoru	20 MIPS při 20 MHz
FLASH	32 KB
SRAM	2 KB
EEPROM	1 KB
Počet pinů, Pouzdro	28 pin PDIP, MLF, 32 pin TQFP, MLF
Maximální frekvence procesoru	20 MHz
Operační napětí	1,8-5,5 V
Počet I/O pinů	23
A/D převodník	6ch 10 bit A/D Converter
Čítače, Časovače	2x 8 bit, 1x 16 bit Timer/Counter
PWM kanály	6
Komunikační rozhraní	USART, SPI, I2C

Tabulka 3: Parametry mikroprocesoru ATmega328p [10]

Figure 29-1. Maximum Frequency vs. V_{CC}



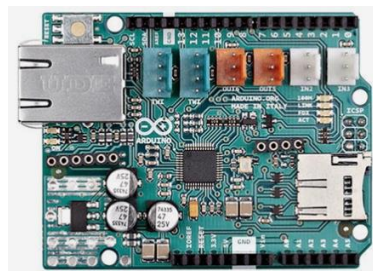
Obrázek 4: Graf závislosti taktovací frekvence na napájecím napětí [11]



Obrázek 5: Blokové schéma mikroprocesoru ATmega328p [11]

2.1.4 Arduino Shieldy

Shield je rozšiřující karta, která se vkládá se shora do patič vývojové desky. Shieldů existuje velké množství a mohou obsahovat nejrůznější periferie, komunikační převodníky, displeje, tlačítka, senzory nebo výkonové spínače. Užitečným příkladem je Ethernet shield, který umožňuje připojení Arduina k síti internet a je vyobrazen na obrázku 6. Shieldy je možné skládat na sebe a vytvořit z nich sendvič. Je však třeba dbát na to, aby nedocházelo ke konfliktům kvůli vícenásobnému použití stejných pinů. Stejné rozmístění pinů jako desky Arduino používají i ostatní výrobci jednodeskových počítačů. Je možné použít Arduino shieldy i s jinými vývojovými deskami např. NUCLEO.

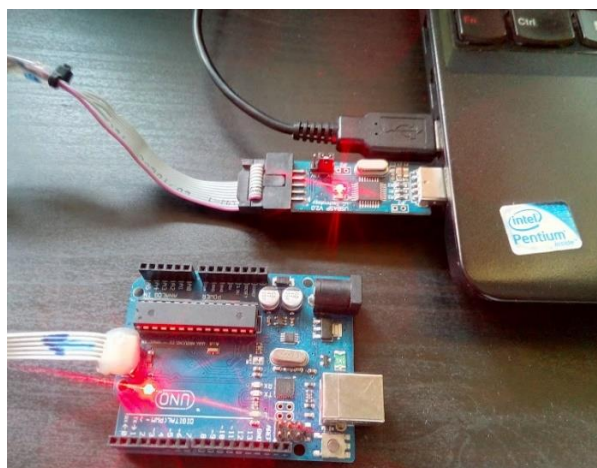


Obrázek 6: Arduino Ethernet shield [9]

2.1.5 Programování vývojových desek ARDUINO

Arduino využívá bootladeru v paměti mikroprocesoru, při programování přes rozhraní USART. Bootlader je program nahraný většinou na konci nebo na začátku paměti flash a řídí její přepsání a verifikaci ve specifický okamžik po zapnutí mikroprocesoru. Pokud se počítači podaří spojit s bootladerem, pak může počítač přepsat paměť flash.

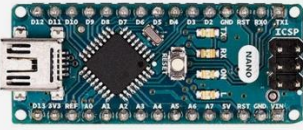






Bootlader však není pouze jediný způsob, jak mikroprocesor programovat. Abychom mohli bootlader využívat, musíme ho nejprve do paměti zapsat. K tomu slouží rozhraní ISP. Na propojení mezi PC a Arduino použijeme zařízení zvané programátor. Programátor může být například typu USBasp, který je Open Source a je založen mikroprocesoru ATmega8 a pouze pár dalších pasivních součástkách [15]. Tímto programátorem můžeme opravit poškozený bootlader Arduina nebo programovat Arduino bez bootladeru a ušetřit místo v paměti. Cheme-li použít programátor s Arduino IDE, je potřeba vybrat v záložce „Projekt“ volbu: „Nahrát pomocí programátoru“. Programátor se dá použít k programování většiny mikroprocesorů ATmega/ATtiny jak v prostředí Arduino tak v prostředí Atmel Studio. Fotografie připojení Arduino UNO k PC pomocí programátoru USBasp je na obrázku 7.



Obrázek 7: Připojení Arduino UNO k programátoru USBasp

2.1.6 Přehled vývojových desek Arduino

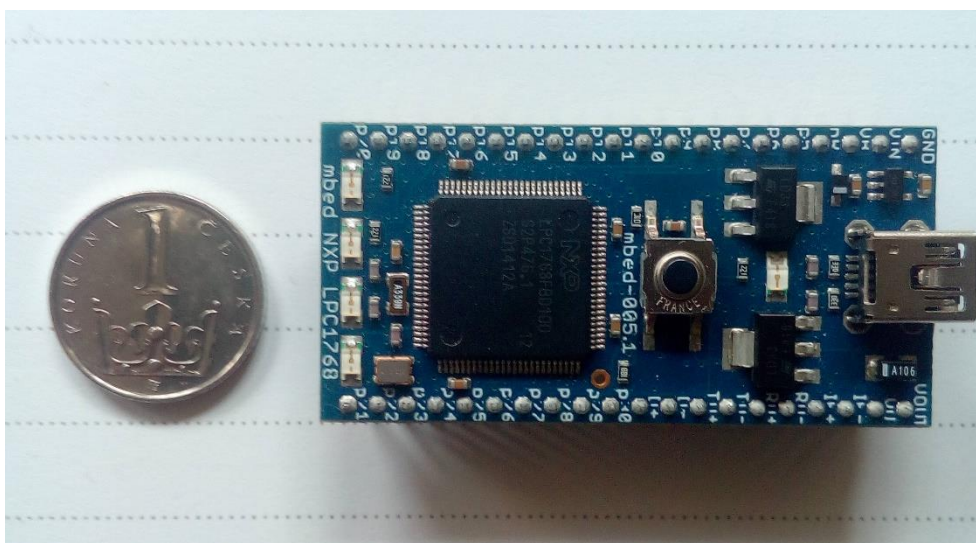
Vývojové desky Arduino bývají přizpůsobené k různým účelům. Některé jsou přímo určené pro internet věcí, jako třeba „Arduino Yun“ nebo „Arduino MKR1000“. Desky s formátem terčíku jsou určené pro nositelnou elektroniku a jsou uzpůsobené tak, aby je bylo možné jednoduše přišít na oděv nebo připevnit pomocí patentů. Mezi tyto desky patří Arduino „GEMMA“ nebo „LILYPAD. Záměrně zde nejsou uvedeny všechny produkty, ale jen výběr z některých. V tabulce 4. je uveden přehled oficiálních jednodeskových počítačů Arduino několika kategorií. Společně s oficiální řadou existuje ještě spousta neoficiálních výrobků (klonů), které se dají pořídit za zlomkovou cenu. Díky volně dostupným podkladům je možné vyrobit si Arduino desku i vlastními prostředky.

Obrázek	Název	Procesor	Poznámka
	NANO	ATmega328p (8 bit, 5 V, 16 MHz, 32 kB flash)	Zmenšená verze Arduino UNO, univerzální použití
	LEONARDO	ATmega32u4 (8 bit, 5 V, 16 MHz, 32 kB flash)	Mikroprocesor s USB rozhraním, Virtuální COM port, podpora HID. (Human Interface Device)
	GEMMA	ATTiny85 (8 bit, 3V3, 8 MHz, 8 kB flash)	Nositelná elektronika, pouze 4 I/O piny, USB programování, bateriové napájení
	101	Intel Curie (32 bit, 3V3, 32 MHz, 194 kB flash)	5 V kompatibilní I/O 6-os. akcelerometr, Gyroskop, Bluetooth
	DUE	SAM3X8E ARM Cortex-M3 (32 bit, 3V3, 84 MHz, 512 kB flash)	Velké množství I/O, Výkonný procesor, USB debugging, D/A převodník
	MKR1000 MKRFOX1200	SAMD21 ARM Cortex-M0+ (32 bit, 3V3, 48 MHz, 256 kB flash)	Internet věcí, Spojení výkonného mikroprocesoru a WIFI nebo SIGFOX modulu
	YUN	Atheros AR9331, (32bit, 3V3, 400MHz, 16 MB flash) ATmega32u4 (viz. LEONARDO)	Internet věcí, propojení Arduina a Operačního systému linux, WIFI, USB, ETHERNET

Tabulka 4: Srovnání Arduino desek [9]

2.2 MBED

ARM mbed je vývojová platforma a také operační systém určený pro zařízení jednoduše připojitelná k internetu. Projekt je vyvíjen společností ARM ve spolupráci s dalšími partnery. Vývojové desky mbed jsou založené na 32bit ARM Cortex-M mikroprocesorech a lze vybírat ze široké škály různých druhů. V této kapitole bude popsána vývojová deska LPC1768, která je na obrázku 8. a také práce s online vývojovým prostředím mbed. V kapitole 2.1.6. bude uveden přehled vybraných mikropočítačů této platformy.

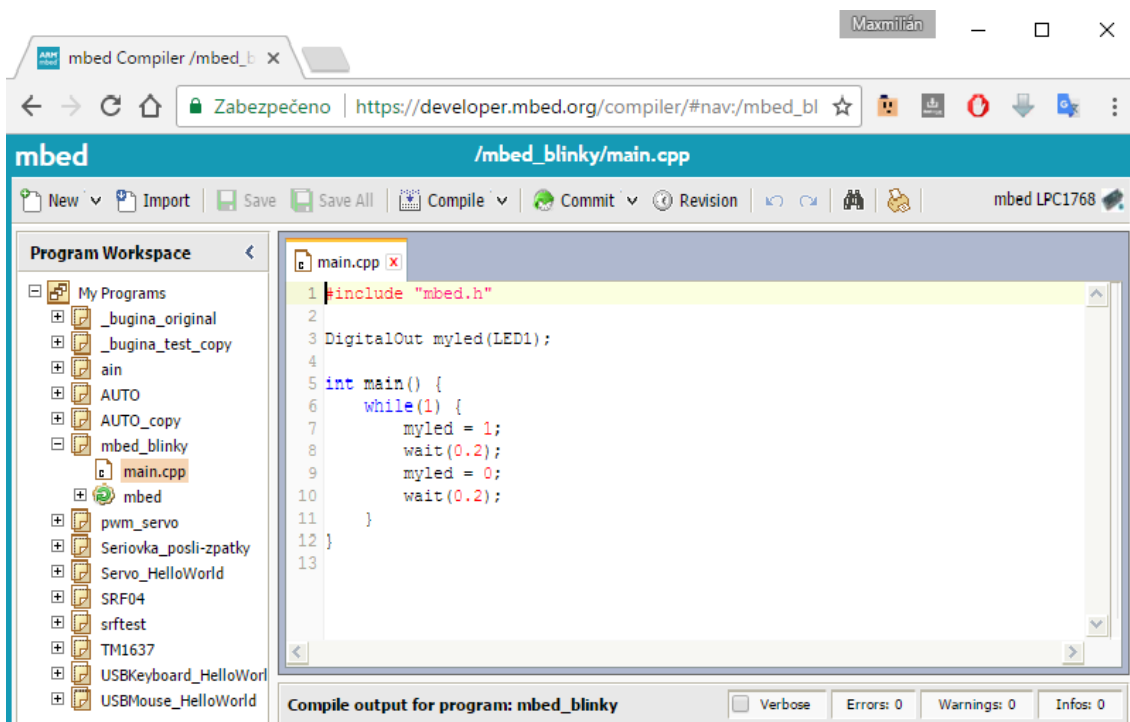


Obrázek 8: Vývojová deska mbed NXP LPC1768

2.2.1 Online vývojové prostředí MBED

Aplikace pro mikropočítače mbed lze programovat v online vývojovém prostředí na stránkách www.mbed.com. K editování a zkompilování kódu nám tak stačí, aby na počítači byl nainstalován pouze webový prohlížeč. Kompilování probíhá na cloudovém serveru v kompilátoru ARMCC C/C++. Abychom získali přístup do IDE, je potřeba založit si uživatelský účet. Tím také získáme soukromý úložný prostor, kam si vývojáři mohou ukládat své rozpracované projekty nebo mohou sdílet hotový kód s dalšími uživateli. Okno IDE mbed můžeme vidět na obrázku 9.

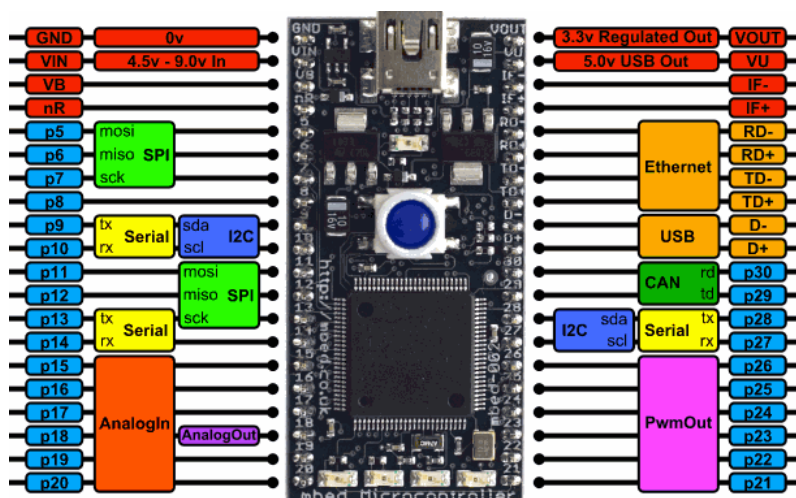
Program napsaný v jazyce C/C++ a přeložený do binárního kódu určeného mikrokontrolerům (s příponou .bin) je následně nabídnut ke stažení. K nahrání firmware do vývojové desky nejsou potřeba žádné speciální nástroje. Firmware se jednoduše uloží na flash disk, který je součástí vývojové desky. Aplikace mohou být taktéž vyvíjeny s použitím vývojových prostředí Keil μ Vision nebo TrueStudio.



Obrázek 9: Okno IDE mbed

2.2.2 mbed NXP LPC1768

První oficiálně vydaná vývojová deska mbed je označována jako mbed NXP LPC1768. Deska obsahuje mikroprocesor LPC1768, napěťové stabilizátory, 4 signalizační diody, resetovací tlačítko a interface mbed, který slouží jako uživatelská paměť. Svou velikostí a umístěním vývodů zespodu je deska vhodná pro použití v kontaktním poli. Na obrázku 10. je popis I/O pinů (pinout). Barevně jsou odlišeny sběrnice mikroprocesoru. Technické parametry této desky jsou uvedeny v tabulce 5.



Obrázek 10: Popis I/O pinů vývojové desky mbed [12]

Parametr	Hodnota
Mikroprocesor	NXP LPC1768
Napětí mikroprocesoru	3,3 V
Napájecí napětí (doporučené)	4,5-9 V
Digitální I/O Piny	26
PWM I/O Piny	6
Analogové vstupy	6
Max. povolený proud na I/O Pin	45 mA
Max. povolený proud na 3.3 V Pin	800 mA
Rozměry	54 x 26 mm

Tabulka 5: Technické parametry desky mbed LPC1768 [12]

2.2.3 Mikroprocesor LPC 1738

LPC1738 je výkonný mikroprocesor architektury ARM CORTEX M3 s nízkou spotřebou. Společnost ARM, která dříve vyráběla mikroprocesory, se nyní věnuje pouze jejich vývoji a licenčně poskytuje ostatním výrobcům jádra mikroprocesorového systému Cortex. Mikroprocesor LPC1738 vyrábí firma NXP (původně Phillips Semiconductors). Firmy které se výrobou mikroprocesorů zabývají (Microchip, Freescale, STM) si jádro ARM přizpůsobí dle vlastní potřeby a doplní mikroprocesor o vnitřní periferie nebo pozmění instrukční sadu.




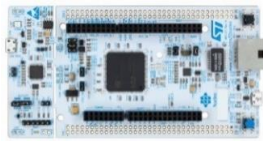
Jádra Cortex-M jsou členěna podle výkonu a spotřeby od nejmenších M0 a M0+ přes středně výkonné a nejvíce univerzální Cortex M3 až po velmi výkonná jádra M7 s taktem 200 MHz, která jsou na hranici mezi realtime procesory a procesory určenými pro mobilní zařízení. Parametry mikroprocesoru LPC 1738 jsou v tabulce 6.

Parametr	Hodnota
Architektura CPU	32 bit ARM Cortex-M3
Maximální frekvence CPU	120 MHz
FLASH	512 kB
SRAM	32-64 kB
Počet pinů, Pouzdro	100 pin, LQFP100
Operační napětí	2,4-3,6 V
Počet I/O pinů	70
A/D převodník	8ch 12 bit Analog/Digital Converter
Čítače, Časovače	4x General Purpose Timers
PWM kanály	6
Vnitřní periferie	4x UART, 2x CAN, 3x SPI, 3x I2C, 10 bit DAC, RTC, Ethernet, USB

Tabulka 6: Parametry mikroprocesoru LPC1768 [14]

2.2.4 Přehled vývojových desek MBED

Vybrané vývojové desky kompatibilní s prostředím MBED jsou uvedeny v tabulce 7. Některé desky (např. Seeeduino) jsou kompatibilní s Arduino shieldy. Záměrně zde nejsou uvedeny všechny druhy, protože jejich výběr je velmi široký.

				
Název	mbed LPC1114	Seeeduino Arch Pro	FRDM-K64F	Nucleo-F746ZG
Procesor	NXP LPC1114	NXP LPC1768	Kinetis MK64FN1M0VLL12	STM32F746ZGT6
Architektura	Cortex-M0	Cortex-M3	Cortex-M4	Cortex-M7
Max. frekvence	48MHz	100MHz	120 MHz	216 MHz
Paměť RAM	8KB	32KB	256 KB	320 KB
Flash paměť	32KB	512KB	1 MB	1 MB
Vlastnosti	USB 2xSPI, I2C UART 6xADC GPIO USB drag 'n' drop FLASH programmer	4xUART 3xI2C 2xSPI USB virtual serial port Ethernet USB Host/Device Debug using CMSIS DAP	3x SPI 3x I2C 5x UART 1x CAN 2x ADC 16bit, mux 12 bit DAC 3 os. akcelerometr magnetometr USB OTG Ethernet 2x tlačítka RGB LED	2x 12 bit DAC 8x USART 4x I2C 6x SPI 2x CAN 3x 24ch 12 bit ADC 2x Advanced-control Timer 1x Low-power Timer Random Generator (HW entropy) USB 2.0 OTG Ethernet
Napájení	4.5-9 V nebo 2,4-3,3V baterie	3V3, 5 V	5 V nebo 4.5-9 V	3V3, 5 V nebo 7-12 V
Rozměry	54 x 26 mm	68 x 58 mm	81 x 53 mm	133 x 70 mm

Tabulka 7: Přehled vybraných vývojových desek mbed [13]

2.3 RASPBERRY PI

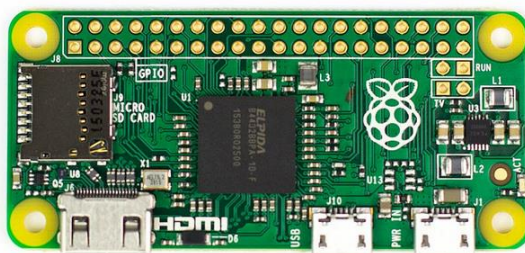
Jednodeskové počítače označené logem maliny vyvíjí britská nadace Raspberry Pi Foundation za účelem podpory výuky informačních technologií ve školách. Miniaturní počítač má základní desku velikosti platební karty a je založen na SoC společnosti Broadcom. Raspberry Pi je určeno pro domácí automatizaci, experimentální elektroniku, ale uživatelé pro něj našli i další uplatnění a používají jej jako multimediální centrum, domácí server či jako desktopový počítač [19].

2.3.1 Verze a provedení počítače Raspberry pi

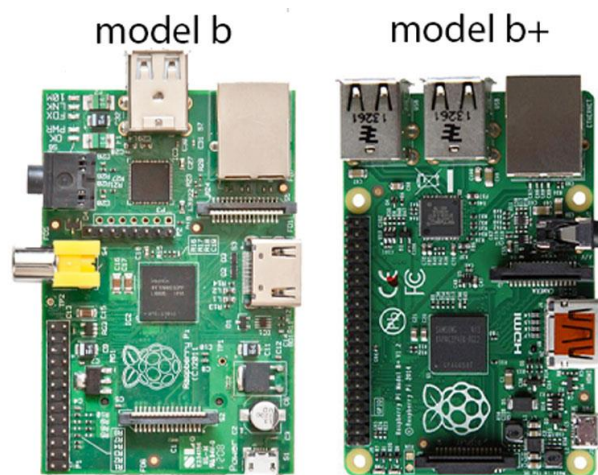
První generace počítače Raspberry Pi se začala prodávat roce 2012 a tyto desky byly označovány jako model A a B. Oproti novějšímu provedení mají jiné rozmístění konektorů, méně GPIO pinů a poněkud nepraktické umístění montážních otvorů. Model A vznikl odlehčením modelu B o síťovou kartu, ethernet port, USB hub a je vybaven pouze 256 MB paměti RAM. Svým nižším výkonem a menší spotřebou je vhodný pro bateriově napájená zařízení.

V roce 2014 byla na trh uvedena druhá generace počítačů Raspberry Pi označovaná jako A+ a B+. Byl zvýšen počet GPIO pinů na 40, přibýly 2 USB porty a byla použita mikro SD karta. Byl také vylepšen napájecí zdroj, kde byl vyměněn lineární stabilizátor napětí za spínaný. Procesory v obou verzích zůstaly stejné. Rozdíly mezi starším a novějším provedením jsou patrné z obrázku 12. Podstatný nárůst výkonu přinesl až model 2 B, který má dvoujádrový procesor. Poslední verze Raspberry Pi model 3 B je zatím nejvýkonnější a dá se použít jako plnohodnotný desktopový počítač. Srovnání výkonnostních parametrů počítačů Raspberry Pi je na obrázku 13.

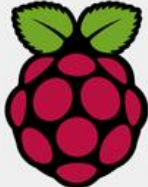
Vlnu kritiky vyvolalo uvedení zatím nejmenšího počítače Raspberry Pi Zero, který je na obrázku 11. Tento miniaturní počítač se měl stát nejlevnějším a nejdostupnějším počítačem na trhu a to díky ceně pouhých 5 \$. Došlo však k enormnímu zájmu o tento počítač a všechny vyrobené kusy byly okamžitě rozprodány. Vznikla paradoxní situace, kdy počítač „pro každého“ je prodáván v druhotném prodeji za cenu 45 \$. Raspberry Pi zero však není jediným počítačem, který se snaží stlačit cenu hardware co nejnižší. Konkurenci k Pi Zero dělá počítač C.H.I.P, který stojí 9 \$ [22].



Obrázek 11: Raspberry Pi Zero [20]



Obrázek 12: Odlišná provedení Raspberry Pi model B [21]

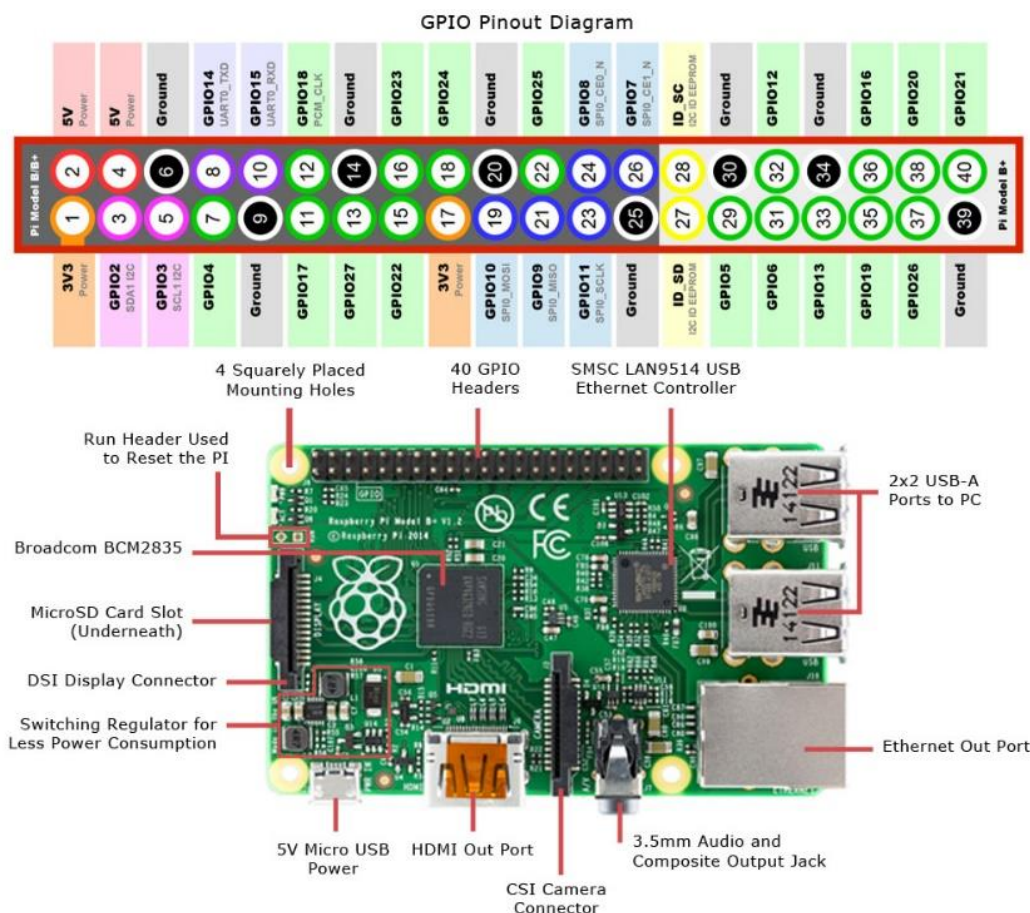
	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	\$35	\$5	\$35	\$35

Obrázek 13: Porovnání parametrů počítačů Raspberry Pi [22].

2.3.2 Raspberry Pi model 3

Raspberry Pi model 3 vybaven 64bitovým CPU o taktu 1,2 GHz. Jde o čtyřjádrový procesor ARM Cortex-A53 a dle výrobce je o 50 % rychlejší než Raspberry Pi 2. Stejně jako Model 2 je vybaven 1 GB operační pamětí. Novinkou jsou kromě 64bitového procesoru i integrované Wi-Fi a Bluetooth moduly. Rozmístění konektorů se oproti předchozímu modelu nemění. Raspberry dále disponuje HDMI výstupem, síťovým rozhraním, USB porty a především množstvím GPIO pinů. Na 40 pinovém konektoru jsou vyvedeny komunikační sběrnice USART, SPI, I2C. Raspberry Pi neobsahuje A/D

převodník a tuto činnost za něj musí obstarat jiný mikroprocesor nebo převodník, který je možné připojit přes jednu z uvedených sběrnic. Popis počítače a rozšiřujících GPIO pinů je na obrázku 14.



Obrázek 14: popis počítače Raspberry Pi model 3 a pinů GPIO [23]

2.3.3 Operační systém

Návrh desek Raspberry Pi je uvolněn pod svobodnou licencí, stejně jako software pro počítač určený. Primárním operačním systémem je Raspbian, klon linuxové distribuce Debian, který je přeložen pro procesory ARM. Vznikla však řada dalších systémů včetně OpenWRT, NetBSD, FreeBSD, Gentoo a dalších. [19] Na Raspberry Pi model 2 a 3 je možné spustit i operační systém Windows 10 IoT.

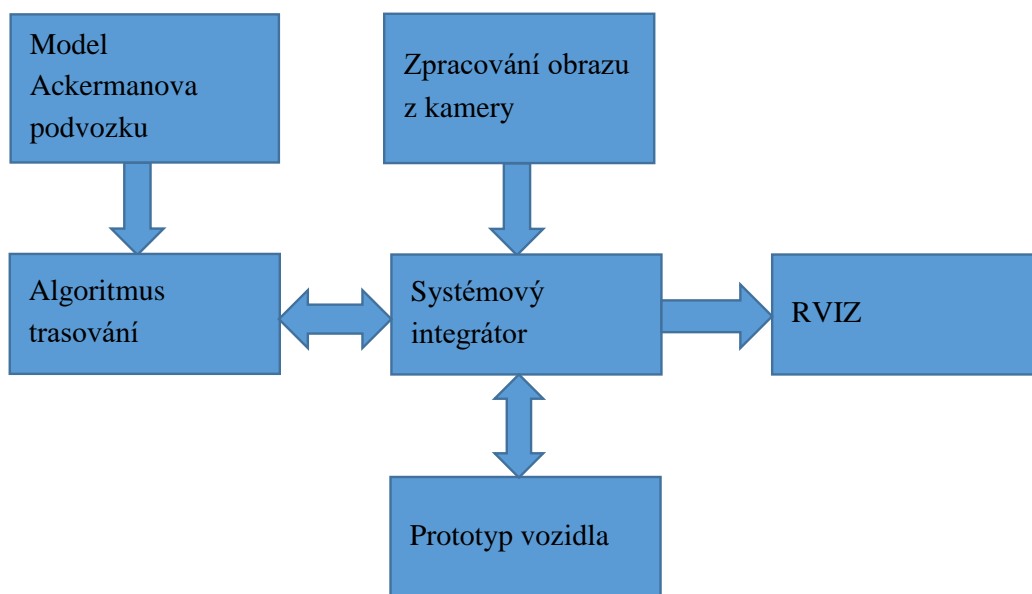
Operační systém je u počítačů Raspberry Pi umístěn na kartě SD. Toto provedení má svá pozitiva i negativa. SD karta je relativně pomalé paměťové zařízení. Oproti ostatním jednodeskovým počítačům, (viz. tab. 1.) které používají jako trvalé úložiště paměť typu eMMC je to nevýhoda a prodlužuje se tím doba spouštění programů a start systému. Na druhou stranu je SD karta výhodná, protože je levná a velmi rozšířená. Zjednodušuje se tak práce s instalováním systému na kartu, kterou v prostředí Windows obstará utilita Win32DiskImager.

3 NÁVRH ŘÍDÍCIHO SYSTÉMU

Tato bakalářská práce je součástí projektu konvoje autonomních vozidel. Projekt se skládá z několika modulů, které dohromady tvoří celek autonomního vozidla. Jednotlivé moduly jsou zpracovány v samostatných bakalářských pracích. Blokové schéma projektu můžeme spatřit na obrázku 15. Řídicí systém se bude vyhodnocovat obraz z webkamery a data budou předávána do operačního systému ROS.

Modul trasování bude používat algoritmus na jehož základě bude vozidlo předvídat trajektorii pohybu. K tomu bude používat model Ackermanova podvozku. RVIZ bude prostřednictvím systému ROS komunikovat se vzdáleným počítačem a vizualizovat data v reálném čase. Systémový integrátor se stará o propojení všech modulů. Integrátor nakonec předá informace o žádané rychlosti a směru pohybu řídicímu systému vozidla. Řídicí systém tyto data převezme a nastaví chod aktuátorů.

V této práci bude navrhnout řídicí systém a realizován prototyp vozidla. Těchto vozidel může být teoreticky v konvoji mnoho, pro vyzkoušení však bude realizováno pouze jedno vozidlo řídicí a jedno vozidlo autonomní.



Obrázek 15: Blokové schéma projektu

3.1 Robotic Operating System

Robotický Operační Systém (ROS) není operační systém v pravém slova smyslu. Je to knihovna programů a nástrojů, která ke svému běhu potřebuje plnohodnotný operační systém. ROS je otevřený projekt se snahou poskytnout prostředí s aktuálně nejvyspělejšími nástroji pro usnadnění vývoje robotických úloh na celé řadě platform. Bylo nám doporučeno abychom pro realizaci vozidla použili distribuci Ubuntu.

Vytvoření skutečné univerzálního robota je obtížný úkol. Z pohledu robota, mohou být pro člověka triviální úkoly tvrdý oříšek. Řešení těchto úkolů je natolik obtížné, že žádná osoba, laboratoř ani instituce nemohou doufat, že je zvládnou samy. Na světě může existovat několik laboratoří a každá může vynikat v řešení jiného problému robotiky (např. orientace robota v prostoru nebo rozpoznávání objektů). ROS vznikl, aby podpořil rozvoj robotického softwaru tím, že umožní vývojářům spolupracovat a stavět na sobě navzájem [17].

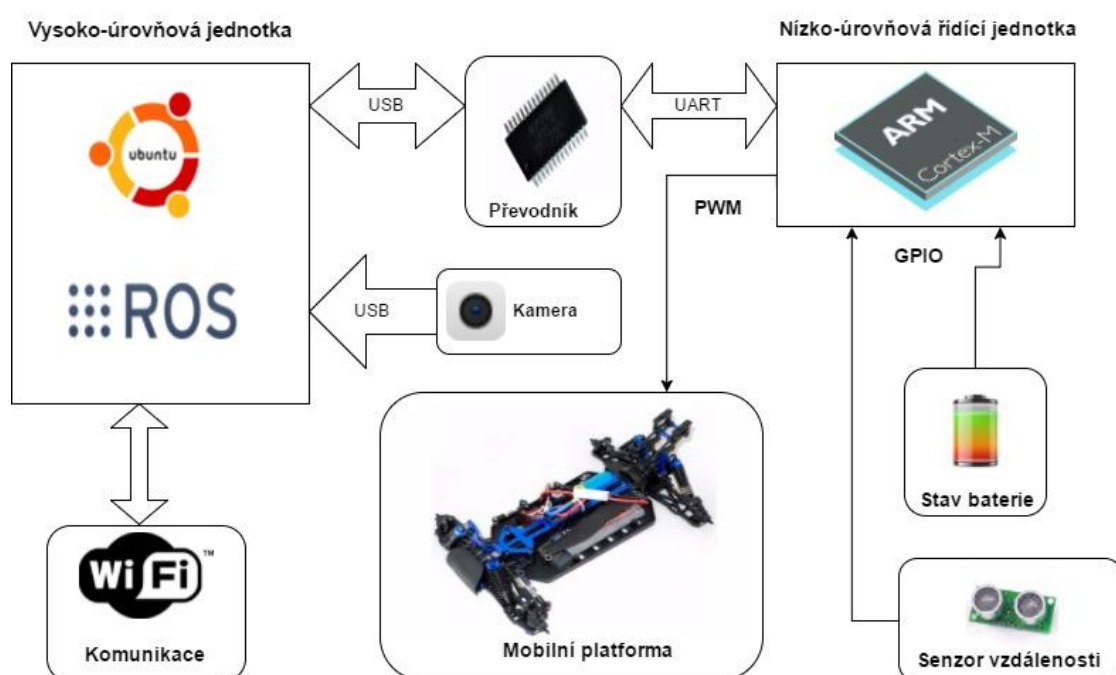
V rámci bakalářské práce budeme využívat systém ROS na nejnižší úrovni, jako nástroj pro zjednodušení komunikace mezi vozidly. Systém nabízí rozhraní pro obousměrný přenos zpráv, které zprostředkovává komunikaci mezi procesy a je obecně označováno jako middleware. Systém zasílání zpráv mezi komunikačními uzly funguje prostřednictvím anonymního mechanismu Publisher / Subscriber. Komunikace po síti systému ROS je nezabezpečená a doporučuje se provozovat ji v rámci VPN.

3.2 Požadavky na řídicí systém

Autonomní vozidlo konvoje bude sledovat značku na vozidle před sebou pomocí kamery. Algoritmus rozpoznání obrazu bude snímky z kamery zpracovávat a vyhodnocovat v reálném čase s frekvencí 25x za sekundu. Tato data budou dále zpracována trasovacím algoritmem. Vozila si budou předávat informace o vzájemném pohybu v rámci komunikační sítě systému ROS. Řídicí počítač musí zpracovávat všechny tyto činnosti najednou, proto budou kladeny relativně vysoké nároky na výkon počítače. Aby se zjednodušila práce počítače, bude obsahovat ještě podřízený nízko-úrovňový počítač, který bude pouze přijímat příkazy z nadřazeného počítače a bude je vykonávat. Náplní práce nízko-úrovňového systému bude ovládat aktuátory vozidla a odesílat naměřená data ze senzorů.

3.3 Návrh struktury řídicího systému

Řídicí systém se bude skládat ze 2 embedded počítačů. Volba platforem bude popsána kapitolách 3.3.1 a 3.3.2. Schéma propojení všech hardwarových modulů je na obrázku 16. Mobilní robot bude mít 2 zdroje energie. Primárním zdrojem je modelářský Li-Po akumulátor, který je součástí mobilního platformy a z něj bude napájen pohon a nízko-úrovňová jednotka. Sekundárním zdrojem energie bude powerbanka, běžně používaná k nabíjení mobilních telefonů. Tento zdroj bude napájet vysokoúrovňovou výpočetní jednotku. Poweranka se jeví jako vhodná, protože má kapacitu v řádech jednotek Ah, integruje v sobě obvod pro nabíjení a ukazatel stavu baterie. Navíc dokáže dodat vysoký proud a napětí 5 V nutné k chodu vysokoúrovňového řídicího systému.



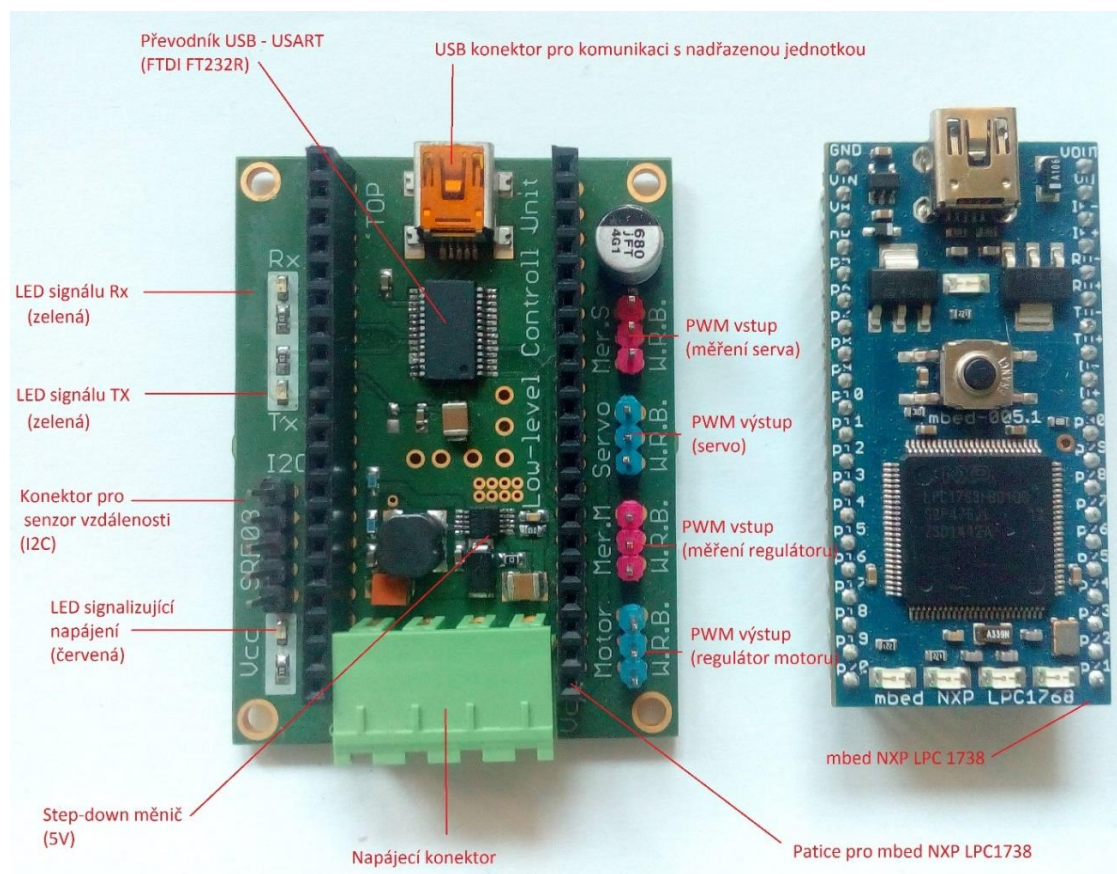
Obrázek 16: Návrh struktury řídicího systému

3.3.1 Vysokoúrovňová řídicí jednotka

Pro realizaci vysoko-úrovňové řídicí jednotky byl vybrán počítač Raspberry Pi 3. Byl zvolen, protože bylo zakoupeno několik kusů do laboratoře programovatelných automatů. Tento systém je výkonem plně srovnatelný s konkurenčními embedded počítači a je přímo navržen pro operační systém Linux Ubuntu, což bylo vyžadováno kvůli náplni ostatních bakalářských prací. Raspberry Pi 3 má integrovaný 4 portový USB hub. Připojíme-li k němu klávesnici, myš a HDMI monitor, můžeme na něm pracovat jako na běžném stolním počítači.

3.3.2 Nízkoúrovňová řídicí jednotka

Nízko-úrovňový řídicí systém bude založen na desce LLCU (Low-level Control Unit), kterou můžeme vidět na obrázku 17. Problematika řízení vozidla autonomního konvoje byla už několikrát řešena na Fakultě strojního inženýrství v rámci bakalářských prací. V roce 2014 byla Vojtěchem Kolomazníkem navržena jednotka LLCU, která bude sloužit jako základ řídicího systému [24]. LLCU funguje ve spojení se mikropočítačem mbed, který oproti platformě Arduino nabízí větší možnosti.



Obrázek 17: Low-level control Unit

Na DPS jednotky LLCU nalezneme spínaný regulátor napětí (step-down měnič), který snižuje napětí Li-Po akumulátoru ze 7,4V na 5 V a napájí všechny periferie včetně mikropočítače mbed. Měnič začíná pracovat od napětí 6 V, tak nízké napětí však baterie neposkytuje, protože při vybití na tuto úroveň by utrpěla trvalé poškození. Deska dále obsahuje obvod pro měření baterie, který bude popsán v kapitole (4.3.3) a komunikační převodník USB – USART, od výrobce FTDI

3.3.3 USART

Sběrnice USART se často používá ke komunikaci mezi mikroprocesorovými systémy. Propojení systémů může být v našem případě provedeno více způsoby. Buďto můžeme využít virtuální komunikační port mbedu nebo na něm máme dispozici několik hardwarový portů. Hardwarový port můžeme připojit buďto na GPIO piny Raspberry Pi nebo využít převodník USB – USART. Připojení na sběrnici USART na GPIO pinech Raspberry Pi 3 je však komplikované, protože oproti starším verzím počítače je na původní sběrnici připojen Bluetooth modul a na GPIO je vyvedena náhradní sběrnice. Muselo by se řešit její konfigurování. Použití převodníku FTDI tak bylo nejjednodušší řešení. Připojením přes převodník můžeme programovat mbed a odlaďovat zároveň, protože máme volný programovací USB konektor. Převodník se po připojení k Raspberry Pi objeví jako device usbttty0. Jeho ovladače jsou součástí systému Linux a můžeme k němu hned přistupovat.

4 REALIZACE EXPERIMENTÁLNÍHO VOZIDLA

4.1 EXPERIMENTÁLNÍ MOBILNÍ PLATFORMA

Jako základ experimentálního vozidla byl ústavem informatiky poskytnut model offroad vozidla 1:10, který je vidět na obrázku 18. Podvozek je vybaven nezávislým odpružením všech kol a všechna kola vozidla jsou hnaná. Zdrojem kroutícího momentu je DC elektromotor. Rozvod výkonu je uskutečněn centrální hřídelí a každá náprava má vlastní diferenciál. Přední náprava je říditelná a splňuje podmínku Ackermanova podvozku. Podvozek bude popsán v kapitole 4.1.1. O natočení přední nápravy se stará modelářský servomotor, který bude rozebrán v kapitole 4.1.2. Elektromotor a způsob řízení motoru bude popsán v kapitolách 4.1.3 a 4.1.4. Detailní popis a seznam všech součástí podvozku je uveden v příloze.

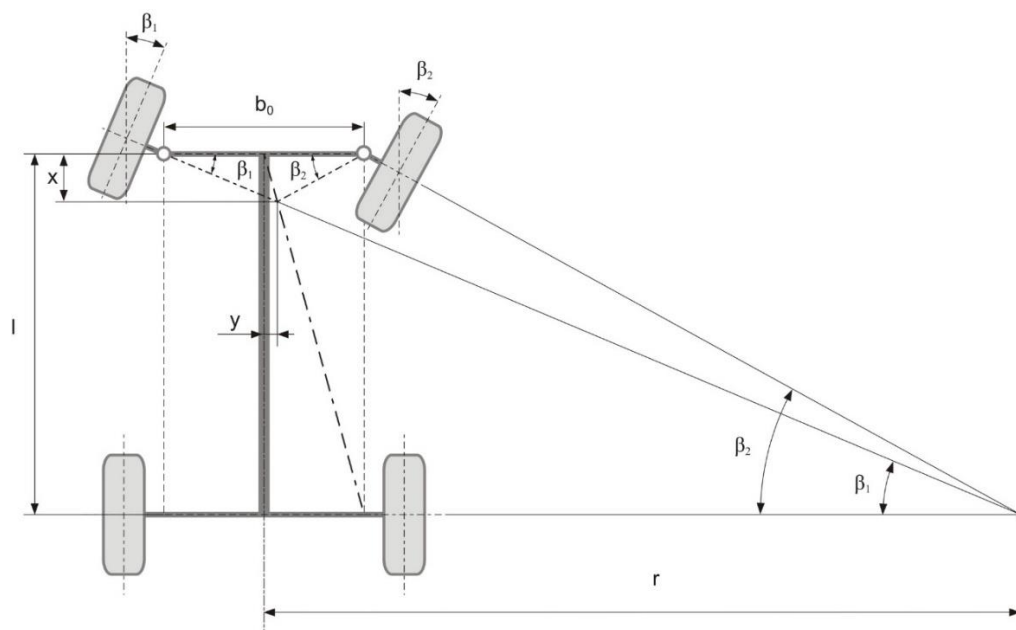


Obrázek 18: Model autonomního vozidla 1:10 [25]

4.1.1 Ackermanův podvozek

Ackermanův podvozek je typ automobilového podvozku s říditelnou přední nápravou. Když jede automobil do zatáčky, jeho vnitřní kola opisují kratší trajektorii než kola vnější. Aby nedocházelo ke smyku a rychlému opotřebování pneumatik je nutné, aby byla kola natočena pod různými úhly. Kolo vnitřní bude natočeno pod větším úhlem než kolo vnější. Vzhledem k tomu, že jsou trajektorie kol různé, budou se lišit i rychlosti otáčení kol na vnitřní a vnější straně vozidla. Vztah mezi úhly natočení kol popisuje rovnice (1). Úhly natočení kol označíme β_1 a β_2 , rozchod kol označíme b_0 a rozvor náprav je označen symbolem l . Podmínku Ackermanova podvozku ilustruje o obrázek 19. Vzdálenost r je poloměr zatáčení.

$$\cot g(\beta_1) - \cot g(\beta_2) = \frac{b_0}{l} \quad (1)$$



Obrázek 19: Ackermanův podvozek [26]

4.1.2 Aktuátor natočení přední nápravy

Servomotor neboli servo řídí natočení přední nápravy. Servomotor je kompaktní celek tvořený stejnosměrným elektromotorem s převodovkou a řídicí elektronikou. Řídicí elektronikou je zpětnovazební regulátor a jako snímání polohy bývá v servech běžně použit potenciometr. Vstupem serva je PWM signál, kterým můžeme nastavit polohu servomotoru v rozsahu 0-180°. Takto velký rozsah však při řízení podvozku nevyužijeme. Krouticí moment je od servomotoru k nápravě přenášen pomocí systému táhel. Servomotor uvedený na obrázku 20. je běžné modelářské servo, které dosahuje krouticího momentu 0,32 N·m a provozní rychlosti 60°/0,23s při napětí 5 V [27].



Obrázek 20: Modelářský servomotor [27]

4.1.3 Stejnosměrný motor

Nejvýkonnějším prvkem vozidla je jeho motor. Jedná se o stejnosměrný motor s permanentními magnety a komutátorem. Na obrázku 21. je DC elektromotor Himoto typ 540, který je použit na bugině. Komutátorové motory mají tu nevýhodu, že přenos el. proudu na rotor je proveden pomocí uhlíků nebo kartáčů. Při chodu motoru dochází na kontaktech k jiskření a vzniklé napěťové špičky mohou rušit chod řídících systémů. Motor je řízen pomocí modelářského regulátoru ESC (Electronic Speed Controller).

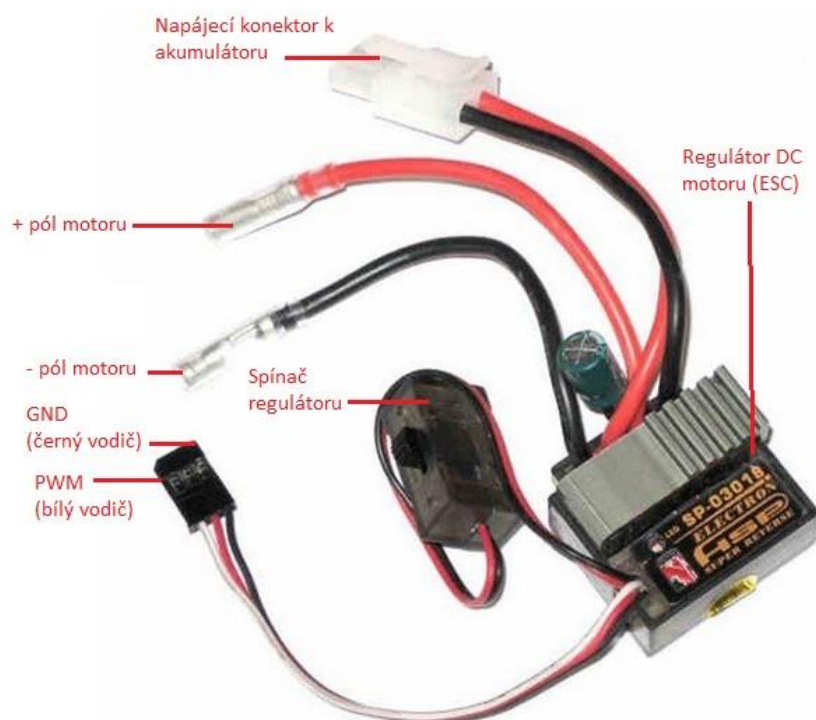


Obrázek 21: Modelářský elektromotor Himoto

4.1.4 Regulátor Stejnosměrného motoru

Regulátor se skládá ze spínače, napájecích konektorů, řídicí a výkonové části. Popis regulátoru je na obrázku 22. Řídicí část obsahuje mikroprocesor a výkonová část je složena z tranzistorů mosfet. Regulátor umožňuje plynule měnit rychlost a směr otáčení motoru. Směr otáčení motoru se mění změnou polarity na jeho pólech, z toho důvodu jsou výkonové tranzistory uspořádány do h-můstku.

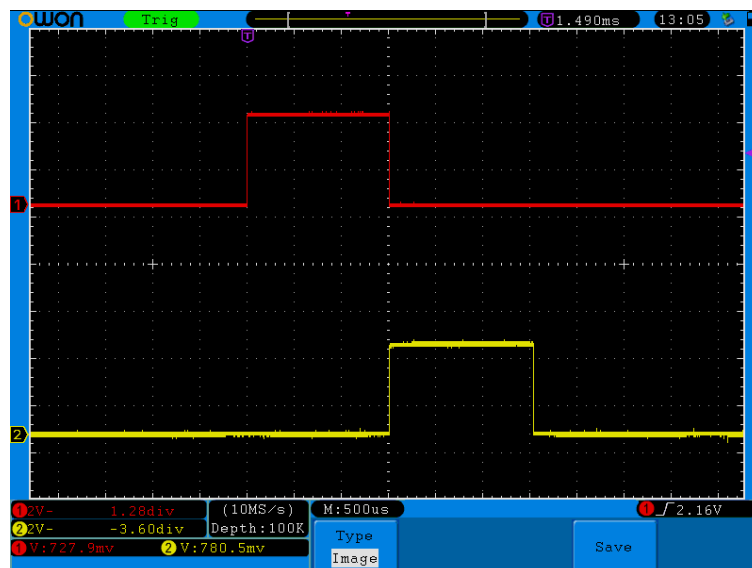
Požadovaná veličina je regulátoru předávána jako PWM signál. Regulátor však nereguluje v pravém slova smyslu, protože nemá od motoru vyvedenu žádnou zpětnou vazbu, pouze řídí výkon a otáčky motoru úměrně ke vstupnímu signálu. Zvětšováním nebo zkracováním délky pulzu pak regulujeme směr a rychlost otáčení motoru. Regulátor se chová tak, že po zapnutí změří PWM signál a od něj pak odvodí nulovou polohu, tzn. stav ve kterém se motor netočí. Je nutné dbát na to, aby při zapnutí regulátoru byl PWM vstup nastaven do výchozího stavu a regulátor se mohl správně inicializovat.



Obrázek 22: Popis regulátoru elektromotoru

4.1.5 Pulzně šířková modulace

Pulzně šířková modulace je způsob přenosu informace, kde jsou data zakódována jako šířka obdélníkového pulzu. Poměru mezi délkou stavu trvání logické jedničky a logické nuly se nazývá střída. Střída se může měnit od 0 do 100% z celkové doby trvání jedné periody. Je tedy zřejmé, že střída 0% odpovídá trvalé logické nule a střída 100% je pak trvalá logická 1. Mikroprocesory jsou většinou vybaveny několika PWM výstupy a ke generování PWM signálu jim slouží periferie časovač neboli Timer. K vytvoření PWM signálu tedy musíme znát periodu budící frekvence, která je pro modelářské účely standardně 50 Hz.



Obrázek 23: Měření PWM signálu digitálním osciloskopem

Abychom získali přehled o změnách PWM signálu, kterým jsou ovládány aktuátory byl provedeno měření výstupu z radiového přijímače modelu. Oba kanály byly měřeny současně digitálním osciloskopem a snímek obrazovky osciloskopu můžeme vidět na obrázku 23. Šířka pulzu na obrázku odpovídá klidovému stavu motoru a natočení kol 0° . Dále byl zjištěn rozsah hodnot PWM signálu pro maximální úhel natočení nápravy a rychlost pohybu vpřed a vzad. Hodnoty jsou uvedeny v tabulkách 8. a 9.

PWM signál Servomotoru		
	Minimum	Maximum
Interval	1220	1980
Mrtvé pásmo	1590	1610
Zatáčení doleva	1220	1590
Zatáčení doprava	1610	1980

Tabulka 8: Rozsah PWM signálu servomotoru

PWM signál regulátoru		
	Minimum	Maximum
Interval	1140	1980
Mrtvé pásmo	1490	1630
Jízda dopředu	1630	1980
Jízda dozadu	1140	1490

Tabulka 9: Rozsah PWM signálu regulátoru

4.2 SENZORY

Senzory se dají přirovnat ke smyslům robota. Díky senzorům robot získává vazbu na okolní prostředí a může reagovat na podněty. Autonomní vozidlo bude používat ke snímání vozidla před sebou WEB kameru a ultrazvukový měřič vzdálenosti.

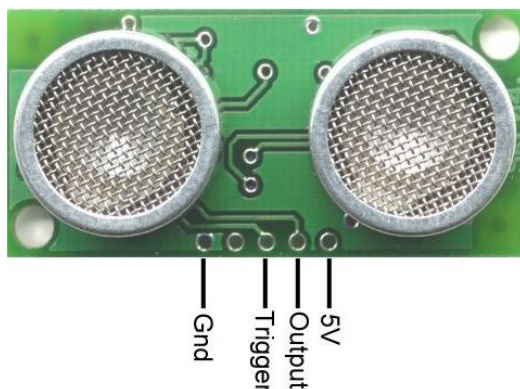
4.2.1 Senzor pro měření vzdálenosti

Řídící jednotka LLCU byla navržena pro použití se senzorem SRF08, který komunikuje po sběrnici I2C. Senzor SRF08 se nám bohužel podařilo poškodit, proto byla zvolena náhradní varianta SRF04, který však komunikuje jiným způsobem. Senzor SRF04 jsme připojili na stejné místo jako SRF08, bylo však nutné odpájet Pull-Up rezistory sběrnice I2C na desce. Ke zprovoznění senzoru SRF04 potřebujeme 4 piny. Popis pinů je na obrázku 24.

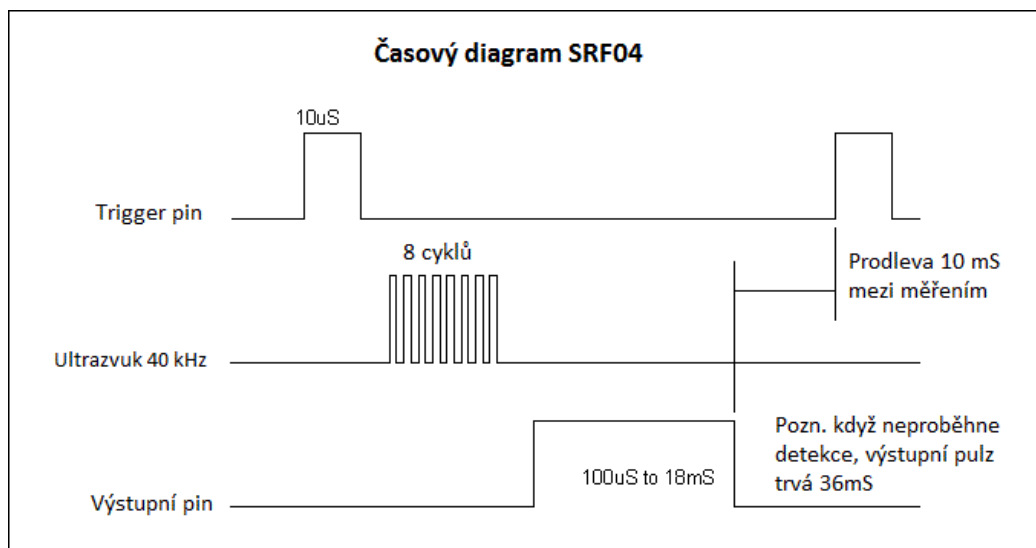
Když vyšleme krátký pulz na pin trigger pin, spustí obvody senzoru sekvenci, která nejprve vybudí ultrazvukový reproduktor. Ultrazvukové vlny o frekvenci 40 kHz se šíří prostorem, a když narazí na překážku, dojde k jejich odražení zpět. Po zachycení odražené vlny ultrazvukovým mikrofonom elektronika senzoru vyhodnotí dobu mezi vysláním a přijmutím zvukové vlny a vyšle na výstupní pin impuls o délce odpovídající vzdálenosti od překážky. Na obrázku 25. můžeme vidět časový průběh na trigger a output pinu během měření.

$$l = \frac{t * c * 10^{-4}}{2} [cm] \quad (2)$$

Vzdálenost překážky od ultrazvukového senzoru je možné vypočítat podle vztahu (2). Symbolem l označíme vzdálenost překážky, délka pulzu v μs je označena t . Rychlost šíření zvuku v suchém vzduchu při teplotě 25 °C je $c = 346,3$ m/s. Uvedený ultrazvukový senzor funguje spolehlivě do vzdálenosti 1,5 m od překážky. Odraz vln může být ovlivněn tvarem překážky a teplotou vzduchu, proto se maximální dosah liší podle prostředí, ve kterém se robot pohybuje. Sonar potřebuje na zotavení mezi měřeními dobu 50 ms, je tedy možné měřit 20x za sekundu. V našem případě budeme měřit pouze 2x za sekundu.



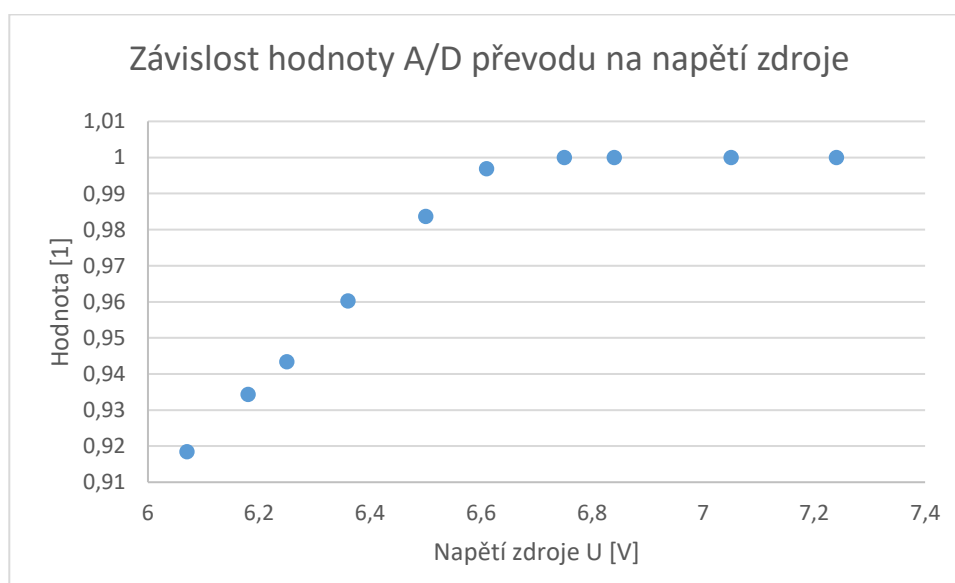
Obrázek 24: Pinout senzoru SRF04 [28]



Obrázek 25: Časový diagram SRF04

4.2.2 Měření stavu akumulátoru

Měření stavu akumulátoru je prováděno A/D převodníkem mikropočítače mbed. Na desce LLCU je obvod složený ze 2 tranzistorů, který umožňuje aktivovat pin p20 pouze po dobu měření. Obvod se skládá z P-channel tranzistoru mosfet, který má v sepnutém stavu velmi malý odpor a vzniká na něm pouze malý úbytek napětí. Tranzistor typu P nelze spínat přímo pinem mbedu, proto je zde navíc NPN tranzistor. Schéma obvodu je uvedeno v příloze. Graf na obrázku 26. ukazuje závislost hodnoty z API AnalogIn na zkušebním napětí laboratorního zdroje.



Obrázek 26: Graf závislosti hodnoty A/D převodu na napětí zdroje

Přes tranzistor mosfet se napájí napěťový dělič, který snižuje napětí akumulátoru na polovinu. Z grafu můžeme vidět, že hodnota nejdříve lineárně roste a při napětí 6,6 V už zůstává konstantní. A/D převodník v mikropočítači mbed má referenční napětí 3,3V, proto není s tímto děličem schopen měřit vyšší napětí než 6,6 V. Akumulátor má v nabitém stavu napětí vyšší než 7,4 V. Z toho důvodu nejsme schopni tímto obvodem napětí měřit, ale pouze indikovat, zda už nekleslo pod úroveň 6,6 V. Abychom mohli měřit vyšší napětí, museli bychom změnit hodnoty rezistorů napěťového děliče, abychom se vešli do měřitelného rozsahu.

4.2.3 Kamera

Kamera je nejvýkonnějším senzorem použitým na vozidle a je stěžejním prvkem při jeho řízení. Na vozidle je použita kamera Microsoft LifeCam HD-3000, kterou můžeme vidět na obrázku 27. Kamery mají oproti dálkoměrům tu výhodu, že poskytují jiný typ dat. Dálkoměry sice dokáží zjistit polohu vozidla v prostoru, ale v reálném provozu je potřeba sledovat i pruhy na vozovce nebo jiné dopravní značení. Řízení vozidla probíhá tak, že kamera sleduje specifickou značku na vozidle před sebou. Snímky z kamery jsou vyhodnocovány algoritmem zpracování obrazu, který používá knihovnu OpenCV.

Algoritmus hledá značku na snímku z kamery pomocí detekce hran a určuje polohu značky pomocí principu středové projekce. Z měření výšky značky v pixelech je vypočítávána vzdálenost značky. Úhel mezi středem značky a středem kamery se určuje pomocí jednoduché trigonometrie. Algoritmus běží v reálném čase a zpracovává snímky rychlostí 25x za sekundu. Zpracování obrazu však není předmětem této práce. Data která algoritmus zpracuje, jsou potom předána do systému ROS pomocí rozhraní JSON (Java Script Object Notation).



Obrázek 27: Kamera Microsoft LifeCam HD-3000

4.3 FIRMWARE PRO ŘÍDÍCÍ JEDNOTKU MBED

4.3.1 Popis komunikačního protokolu

Firmware pro mbed byl napsán tak, aby bylo ošetřeno přijetí nedefinované zprávy. K tomuto účelu slouží ve zprávě prembly, který je ověřován několika podmínkami. Pokud testem projde, je zpráva dekodována a hodnoty aktuátorů jsou nastaveny. Zpráva se odesílá přes rozhraní USART jako sekvence bajtů. Význam jednotlivých bajtů je uveden v tabulce 10. Hodnoty X a Y jsou jednotlivé číslice předávané hodnoty. Bajt s pořadovým číslem 3 bude odpovídat řádu stovek předávaného čísla.

Čísla mohou nabývat hodnot od 0 do 700, přičemž hodnota 350 je klidová hodnota pro oba aktuátory, při které motor stojí a kola přední nápravy nejsou vychýlena. Čísla nespádající do intervalu <0;700> budou programem ignorována. Rozsah 0-700 byl zvolen, protože odpovídá použitelnému rozsahu natočení nápravy a užitečné rychlosti motoru. Funkce pro výpočet délky PWM bude popsána v kapitole 4.3.4.

Komunikační protokol											
Pořadí bajtu	0	1	2	3	4	5	6	7	8	9	10
ASCII znak	/	/	A	X	X	X	.	Y	Y	Y	&
Význam	Preamble		ID zprávy	Hodnota natočení nápravy		Oddělovací znak		Hodnota rychlosti vozidla		Ukončovací znak	

Tabulka 10: Komunikační protokol

4.3.2 Komunikace s nadřazenou jednotkou

Během psaní firmware pro mbed docházelo často k tomu, že se mikropočítač zasekával. Přejít do tohoto stavu bylo zamezeno několika opatřeními. V první řadě není vhodné při programování v prostředí mbed používat funkci *printf()*. Místo toho je lepší použít při odesílání znaků přes USART klasické funkce *putc()*. Dále není vhodné používat funkci *wait()*, která při svém vykonávání zabírá veškerý strojový čas. Vyvolání přerušení během konání této funkce může mít za následek přechod do neznámého stavu a zaseknutí systému. Kód v ISR (Interrupt Service Routine) by měl být také co nejkratší.

Funkce, která přijímá řetězec ze sériového rozhraní je v tabulce 11. Když je přijat jeden bajt, je vyvoláno přerušení *SerialRecvInterrupt()*. Po načtení celé zprávy do pole *buffer* je celá zpráva dekodována pomocí funkce: *sscanf(buffer, "//A%d.%d", &x, &y)*. Funkce *sscanf()* hledá v poli čísla, na základě předem známého formátování a ukládá je na adresy proměnných x a y. Tečka zde slouží jako oddělovací znak.

SerialRecvInterrupt

```

void SerialRecvInterrupt (void){
    char c;
    c = usart.getc();
    buffer[i] = c;
    i=i+1;
    if (c == '&'){           //když přijde ukončovací znak, nastaví se vlajka
        i = 0;
        flag_serial = 1;
    }
}

```

Tabulka 11: Funkce přerušení SerialRecvInterrupt

4.3.3 Získávání dat ze senzorů

Všechny časované události jsou spouštěny periodicky z API Ticker. Měření Vzdálenosti je prováděno každých 0,5 s a měření baterie každé 3 s. Mbed využívá pro vyvolání přerušení jednoho z univerzálních timerů v mikroprocesoru LPC1738. Ke čtení hodnoty napětí z pinu p20 je použito API AnalogIn. Funkce měření baterie je uvedena v tabulce 12. Měření vzdálenosti je prováděno tak, jak bylo popsáno v kapitole 4.2.1.

Sonar je vybuzen krátkým pulzem na pinu p10. Jakmile přijde vzestupná hrana na pin p9, spustí se timer a začne počítat čas v μ s. Když je vyvoláno přerušení od sestupné hrany, počítání se ukončí a čas je uložen do proměnné *pulse_us*. Funkce která je volána při sestupné hraně, je uvedena v tabulce 13. Je zbytečné přepočítávat vzdálenost z μ s přes vzorec (2), místo toho je lepší dělit délku pulzu v μ s číslem 58. Touto úpravou se zkrátí doba trvání výpočtu.

Měření stavu baterie

```

void BATTERY_MEASURE(){
    led2 = !led2;           //negování stavu led2
    bat_enable = 1;         //aktivuje se obvod pro měření napětí baterie
    value = ain.read();     //čtení hodnoty z AD převodníku
    bat_enable = 0;         //vypnutí obvodu
}

```

Tabulka 12: Funkce měření stavu baterie.

Externí přerušení na pinu p9	
<pre> void fall(){ timer_SONAR.stop(); //zastavím časovač pulse_us = timer_SONAR.read_us(); //získám délku pulzu pin_echo.rise(NULL); //nulování přerušení timer_SONAR.reset(); //Reset resetuji časovač flag_pulse = 1; } </pre>	

Tabulka 13: Funkce externího přerušení

4.3.4 Ovládání aktuátorů

Data o natočení přední nápravy jsou kontrolována ve funkci *Angle()*, která je uvedena v tabulce 14. Hodnota natočení musí být v rozsahu 0-700. Pokud je podmínka splněna, je hodnota převedena na délku pulzu v μs pouhým sečtením s konstantou. Funkce pro kontrolu rychlosti motoru vypadá obdobně. V tabulce 15. je ukázka kódu, která nastaví PWM signál na pinu, který je přiřazen objektu Motor s frekvencí 50 Hz.

Kontrolování dat pro úhel natočení nápravy
<pre> int Angle(int alfa){ int servo_us; if(alfa>=0 && alfa<=700){ if(alfa<350){ servo_us = alfa+1240; //1240+350 = 1590 } else if(alfa>350){ servo_us = alfa+1260; //1260+350 = 1610 } else{ servo_us = SERVO_NULL; //servo null } } return servo_us; } </pre>

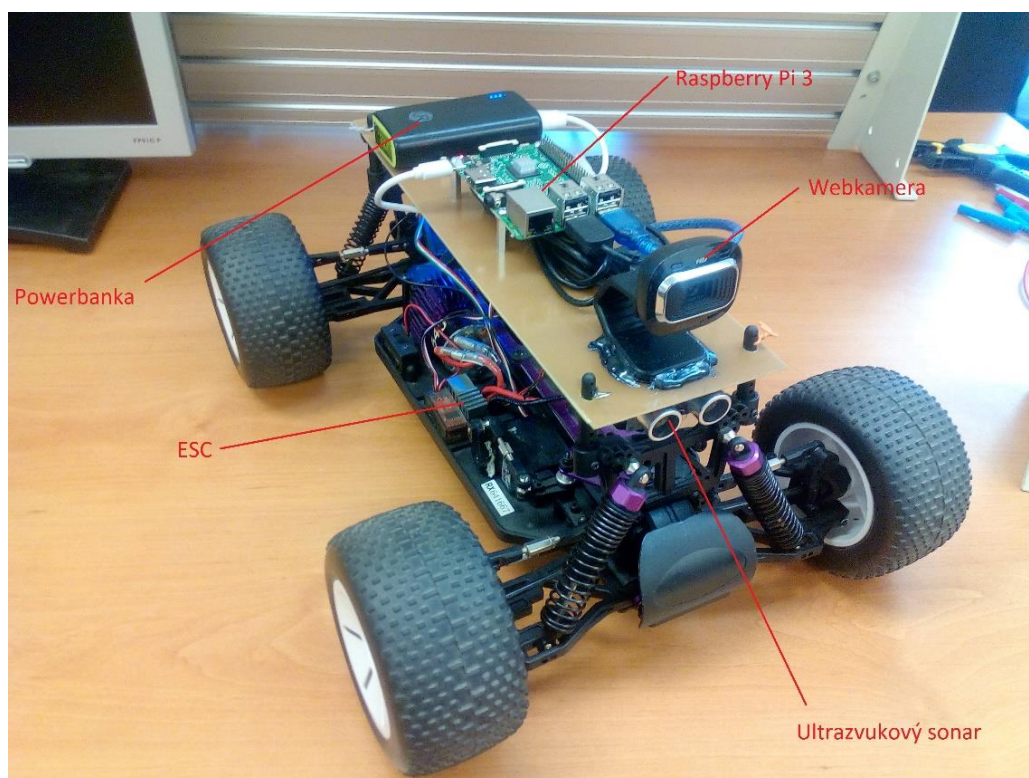
Tabulka 14: Funkce převodu přijaté hodnoty na délku pulzu v μs

Nastavení PWM na výstupním pinu	
Motor.period_ms(20);	//perioda v ms
Motor.pulsewidth_us(SERVO_NULL);	//délka pulzu

Tabulka 15: Nastavení PWM na výstupním pinu

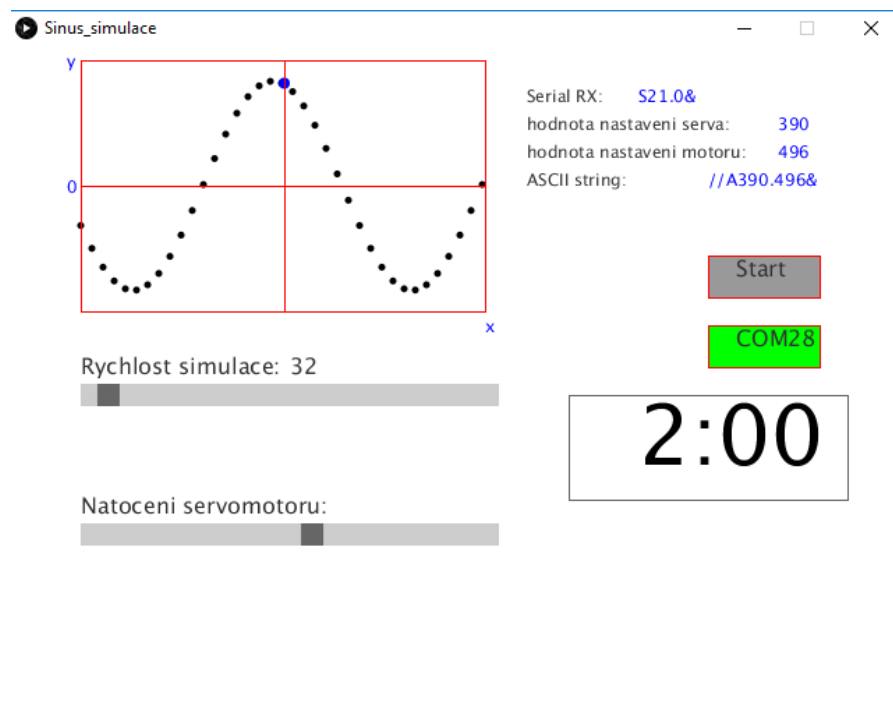
5 TESTOVÁNÍ PROTOTYPU VOZIDLA

Fotografie hotového prototypu vozidla i s popisem jednotlivých částí je na obrázku 28. V rámci testování byla vyvinuta aplikace, která značně ulehčila odladění firmware pro mbed. Aplikace zobrazuje odesílaná i přijímaná data. Po stisku tlačítka start se spouští simulace a začíná se počítat čas. Simulace mění rychlost motoru v závislosti na časovém průběhu funkce sinus. Průběh funkce můžeme vidět na grafu v okně programu na obrázku 29. Hodnota rychlosti odpovídá modrému bodu. Natočení nápravy můžeme měnit pomocí slideru, stejným způsobem se dá změnit i rychlost simulace, která může probíhat nejrychleji 30x za sekundu.



Obrázek 28: Fotografie hotového vozidla.

Aplikace byla napsána v prostředí Processing, které je příbuzné s prostředím Arduino. Programovací jazyk Processing je podobný zjednodušené Javě. V IDE můžeme nalézt spoustu příkladů (examples) k vyzkoušení. Prostředí se nehodí k programování složitějších aplikací, protože není navrženo k vytváření uživatelského rozhraní. Chceme-li v aplikaci použít např. tlačítko, je potřeba si ho naprogramovat. Z tohoto pohledu je mnohem výhodnější použít k vytvoření programu IDE Visual Studio. Silnou stránkou jazyka Processing jsou jeho grafické aplikace. Jednoduše se v něm dají kreslit plošné i prostorové geometrické útvary. Processing také umožňuje vytvořit samostatně spustitelnou Windows aplikaci.



Obrázek 29: Okno testovacího programu

Poté co byl firmware pro mbed připraven, bylo provedeno testování vozidla v rámci autonomního konvoje. V té době jsme ještě neměli trasovací algoritmus, takže řízení autonomního vozidla bylo pouze proporcionální a při zatáčení ztrácela kamera zaměřenou značku. Fotografie z testování konvoje je na obrázku 30.



Obrázek 30: Fotografie z testování konvoje autonomních vozidel

6 ZÁVĚR

Hlavním cílem práce bylo navrhnout strukturu řídicího systému autonomního vozidla a vozidlo experimentálně otestovat. Návrhu struktury předcházela průzkum aktuálně dostupných embedded systémů a výběr vhodných platform. Pro vysoko-úrovňový řídicí systém byl vybrán počítač Raspberry Pi model 3, který je popsán v kapitole 2.3 a pro nízko-úrovňový systém byl vybrán mikropočítač mbed, který je popsán v kapitole 2.2. Dále bylo nutné provést analýzu modelu vozidla a zjistit jeho vlastnosti. Bylo zjištěno, že aktuátory vozidla je možné řídit pomocí PWM signálů.

Pro realizaci řídicího systému nám byla poskytnuta rozšiřující deska Low-Level Control Unit k mikropočítači mbed, která značně usnadnila vývoj řídicího systému, protože už obsahuje napájecí zdroj, komunikační převodník, obvod pro měření baterie a konektory pro připojení periférií. Desku bylo nutné upravit pro použití s měřičem vzdálenosti SRF04, odpájením pull-up rezistorů ze sběrnice I2C. Bylo také zjištěno, že navzdory předpokladu není možné přímo měřit napětí baterie pomocí A/D převodníku, ale pouze indikovat pokles napětí baterie pod pevně stanovenou úroveň. Tento problém by bylo možné řešit dalšími úpravami, ale nebylo k nim přistoupeno.

Ke zprovoznění řídicího systému bylo potřeba vyvinout nový firmware pro mikropočítač mbed. Vývoj firmware je popsán v kapitole 4.3. Při programování firmware byla vyzkoušena práce s on-line vývojovým prostředím mbed. Firmware komunikuje přes rozhraní USART se software v počítači Raspberry Pi, z toho důvodu bylo nutné zvolit komunikační rámec, který budou oba systémy používat. Firmware je dostatečně spolehlivý a robustní, aby vydržel běžet dlouhodobě bez výpadků. K otestování stability firmware a funkčnosti vozidla byla vytvořena aplikace v prostředí Processing. Aplikace umožňuje číst přijímaná data z řídicí jednotky a ovládat rychlost motoru a natočení servomotoru z počítače se systémem Windows. Funkční prototyp vozidla je připraven k testování řídicích algoritmů v rámci autonomního konvoje.

7 SEZNAM POUŽITÉ LITERATURY

- [1] Embedded systémy. *Masaryk University* [online]. Brno: Hana Mršítková, 2005 [cit. 2017-05-09]. Dostupné z: <https://www.fi.muni.cz/usr/jkucera/pv109/2005/xmrstik.htm>
- [2] Big.LITTLE Technology. *ARM* [online]. Cambridge: ARM, 2017 [cit. 2017-05-17]. Dostupné z: <https://www.arm.com/products/processors/technologies/biglittleprocessing.php>
- [3] Raspberry Pi 3 Model B. *Rs-components.com* [online]. Corby (UK): RS components, 2016 [cit. 2017-05-09]. Dostupné z: <http://docs-europe.electrocomponents.com/webdocs/14ba/0900766b814ba5fd.pdf>
- [4] *LattePanda* [online]. Shanghai: DFRobot, 2016 [cit. 2017-05-09]. Dostupné z: <https://www.lattepanda.com/>
- [5] BeagleBone - Blue. *Beagleboard.org* [online]. Michigan: Creative Commons BY 3.0, 2016 [cit. 2017-05-09]. Dostupné z: https://www.electronicsdatasheets.com/datasheet/BeagleBone_Blue.pdf
- [6] ODROID-XU4. *USER MANUAL: ODROID-XU4* [online]. 20170310. Soul: Hardkernel, 2015, s. 3-10 [cit. 2017-05-09]. ISBN není uvedeno. Dostupné z: <https://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf>
- [7] VODA, Zbyšek. *Průvodce světem Arduina*. Bučovice: Martin Stríž, 2015. ISBN 978-80-87106-90-7.
- [8] Snootlab. Arduino UNO. In: *Flickr* [online]. Creative Commons BY 2.0, 2011 [cit. 2017-05-17]. Dostupné z: <https://www.flickr.com/photos/snootlab/6052455554>
- [9] *Arduino Products* [online]. Massachusetts: Creative Commons BY 3.0, c2017 [cit. 2017-05-17]. Dostupné z: <https://www.arduino.cc/en/Main/Products>
licenční podmínky jsou dostupné z: <http://creativecommons.org/licenses/by-sa/3.0/cz/>
- [10] ATmega328p. *Microchip* [online]. United States: Atmel Corporation, c1998-2017 [cit. 2017-05-12]. Dostupné z: <http://www.microchip.com/wwwproducts/en/ATmega328P>
- [11] *ATmega328P: DATASHEET COMPLETE* [online]. Rev. B. United States: Atmel Corporation, c2015 [cit. 2017-05-12]. ISBN Není uvedeno. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
- [12] Mbed LPC1768. *ARM mbed* [online]. Cambridge: ARM, c2017 [cit. 2017-05-17]. Dostupné z: <https://developer.mbed.org/platforms/mbed-LPC1768/>
- [13] *ARM mbed: Boards* [online]. Cambridge: ARM, c2017 [cit. 2017-05-15]. Dostupné z: <https://developer.mbed.org/platforms/>
- [14] *LPC1769/68/67/66/65/64/63: 32-bit ARM Cortex-M3 microcontroller* [online]. Rev. 9.7. Eindhoven: NXP, c2017 [cit. 2017-05-17]. ISBN Není uvedeno. Dostupné z: http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf
- [15] USBasp - USB programmer for Atmel AVR controllers. *USBasp: USB programmer for Atmel AVR controllers* [online]. Schmidtsberg: Thomas Fischl, c1998-2017 [cit. 2017-05-15]. Dostupné z: <http://www.fischl.de/usbasp/>

- [16] THRUN Sebastian, BURGARD Wolfram, and FOX Dieter. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series). Intelligent robotics and autonomous agents. The MIT Press, August 2005.
- [17] ROS.org. ROS.org | Powering the world's robots. [online]. 7.11.2016 [cit. 2016-11-07]. Dostupné z: <http://www.ros.org/>
- [18] *SRF04 - Ultra-Sonic Ranger: Technical Specification* [online]. Norfolk: Robot Electronics, 2003 [cit. 2017-05-15]. Dostupné z: <https://www.robot-electronics.co.uk/htm/srf04tech.htm>
- [19] Raspberry Pi. *Root.cz* [online]. Praha: Petr Krčmář, 2012 [cit. 2017-05-12]. Dostupné z: <https://www.root.cz/n/raspberry-pi/>
- [20] LECOURT, Pierre. Raspberry Pi Zero. In: *Flickr* [online]. Creative Commons BY 2.0, 2011 [cit. 2017-05-17]. Dostupné z: <https://www.flickr.com/photos/13815526@N02/23211671742/>
- [21] GUARISE, Michelangelo. Raspberry Pi Model B+: Upgrades Overview. In: *Volumio* [online]. Itálie: Volumio, 2014 [cit. 2017-05-17]. Dostupné z: <https://volumio.org/raspberry-pi-model-b-plus/>
- [22] *HACKADAY* [online]. WorldPress.com VIP, c2017 [cit. 2017-05-17]. Dostupné z: <http://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/>
- [23] ROBILLARD, Steve. Raspberry Pi GPIO. In: *StackExchange* [online]. East Haven(USA): StackExchange, 2015 [cit. 2017-05-17]. Dostupné z: <https://raspberrypi.stackexchange.com/questions/40318/raspberry-pi-2-can-gpio-pins-29-40-be-used-gpio-gen-input-output-c>
- [24] KOLOMAZNÍK, V. *Návrh a realizace řídicích jednotek pro semiautonomní konvoj*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014.
- [25] *Himoto* [online]. Čína [cit. 2017-05-19]. Dostupné z: http://www.himotoracing.com/product/truck/electric/truck_e001/details.php?information=1
- [26] Ackermanova podmínka. *Autolexicon.net* [online]. není uvedeno: Ing. Jan Sajdl, Ph.D, c2017 [cit. 2017-05-19]. Dostupné z: <http://www.autolexicon.net/cs/articles/ackermannova-podminka/>
- [27] Standardní servo Futaba S3003. *Modelarina.cz* [online]. Jinolice: MDK obchod, c2008-2017 [cit. 2017-05-19]. Dostupné z: <http://www.modelarina.cz/standardni-servo-futaba-s3003-p-107.html>
- [28] BKIT Hardware Club. *Srf04_1*. In: *Flickr* [online]. Creative Commons BY 2.0, 2011 [cit. 2017-05-24]. Dostupné z: <https://www.flickr.com/photos/91448513@N02/8308311023/>

SEZNAM OBRÁZKŮ

Obrázek 1: Vývojová deska Arduino UNO [8]	19
Obrázek 2: Vývojové prostředí Arduino	20
Obrázek 3: Popis desky Arduino UNO	21
Obrázek 4: Graf závislosti taktovací frekvence na napájecím napětí [11]	22
Obrázek 5: Blokové schéma mikroprocesoru ATmega328p [11]	23
Obrázek 6: Arduino Ethernet shield [9]	23
Obrázek 7: Připojení Arduino UNO k programátoru USBasp	24
Obrázek 8: Vývojová deska mbed NXP LPC1768	26
Obrázek 9: Okno IDE mbed	27
Obrázek 10: Popis I/O pinů vývojové desky mbed [12]	27
Obrázek 11: Raspberry Pi Zero [20]	30
Obrázek 12: Odlišná provedení Raspberry Pi model B [21]	31
Obrázek 13: Porovnání parametrů počítačů Raspberry Pi [22].	31
Obrázek 14: popis počítače Raspberry Pi model 3 a pinů GPIO [23]	32
Obrázek 15: Blokové schéma projektu	33
Obrázek 16: Návrh struktury řídicího systému	35
Obrázek 17: Low-level control Unit	36
Obrázek 18: Model autonomního vozidla 1:10 [25]	39
Obrázek 19: Ackermanův podvozek [26]	40
Obrázek 20: Modelářský servomotor [27]	40
Obrázek 21: Modelářský elektromotor Himoto	41
Obrázek 22: Popis regulátoru elektromotoru	42
Obrázek 23: Měření PWM signálu digitálním osciloskopem	43
Obrázek 24: Pinout senzoru SRF04 [28]	44
Obrázek 25: Časový diagram SRF04	45
Obrázek 26: Graf závislosti hodnoty A/D převodu na napětí zdroje	45
Obrázek 27: Kamera Microsoft LifeCam HD-3000	46
Obrázek 28: Fotografie hotového vozidla	51
Obrázek 29: Okno testovacího programu	52
Obrázek 30: Fotografie z testování konvoje autonomních vozidel	52

SEZNAM TABULEK

Tabulka 1: Přehled aktuálně dostupných Embedded PC systémů.....	18
Tabulka 2: Technické parametry Arduino UNO [9]	21
Tabulka 3: Parametry mikroprocesoru ATmega328p [10]	22
Tabulka 4: Srovnání Arduino desek [9]	25
Tabulka 5: Technické parametry desky mbed LPC1768 [12].....	28
Tabulka 6: Parametry mikroprocesoru LPC1768 [14]	28
Tabulka 7: Přehled vybraných vývojových desek mbed [13]	29
Tabulka 8: Rozsah PWM signálu servomotoru.....	43
Tabulka 9: Rozsah PWM signálu regulátoru.....	43
Tabulka 10: Komunikační protokol.....	47
Tabulka 11: Funkce přerušení SerialRecvInterrupt.....	48
Tabulka 12: Funkce měření stavu baterie.....	48
Tabulka 13: Funkce externího přerušení	49
Tabulka 14: Funkce převodu přijaté hodnoty na délku pulzu v μ s	49
Tabulka 15: Nastavení PWM na výstupním pinu.....	50

SEZNAM PŘÍLOH

Přílohy

Příloha 1: Popis všech dílů podvozku RC modelu

Příloha 2: Schéma zapojení obvodu měření baterie

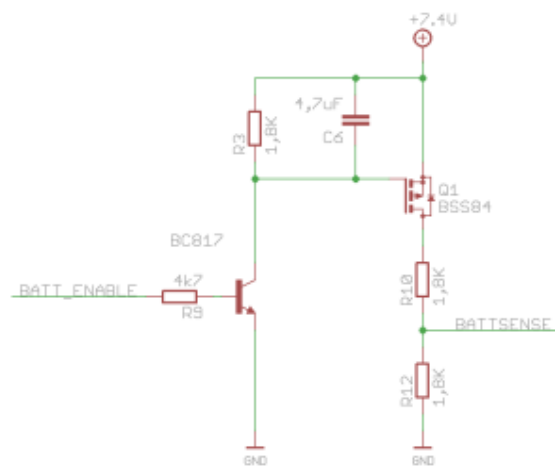
Přílohy na CD

Příloha 1: Elektronická verze práce ve formátu PDF

Příloha 2: Kód firmware pro mbed v textovém souboru

Příloha 3: Zkompilovaný firmware v binárním souboru

Příloha 4: Testovací aplikace



Příloha 2: Schéma zapojení obvodu měření baterie [24]