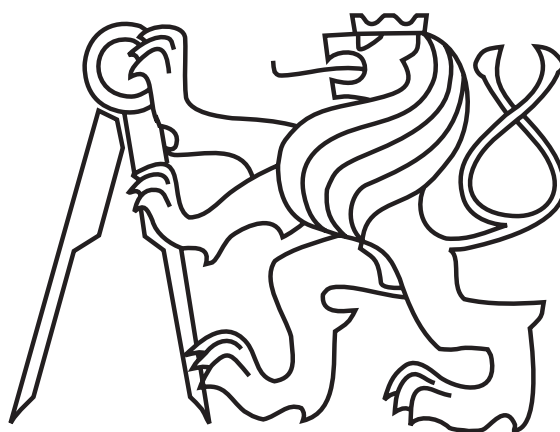


České vysoké učení technické v Praze
Fakulta dopravní

Ústav dopravní telematiky
Obor: Inteligentní dopravní systémy
Studijní program: Technika a technologie v dopravě a spojích



Navigace robota pomocí
dálkoměrných sensorů

BAKALÁŘSKÁ PRÁCE

Vypracoval: Vít Šklebený
Vedoucí práce: Ing. Petr Bureš, Ph.D.
Rok: 2014



K620..... Ústav dopravní telematiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Vít Šklebený

Kód studijního programu a studijní obor studenta:

B 3710 – ITS – Inteligentní dopravní systémy

Název tématu (česky): **Navigace robota pomocí dálkoměrných senzorů a jejich kalibrace**

Název tématu (anglicky): BoE Bot Navigation by Range Sensors and their Calibration

Zásady pro vypracování

Při zpracování bakalářské práce se řiďte osnovou uvedenou v následujících bodech:

- Úvod a obecný popis robota BoE Bot
- Osazení robota dálkoměrnými senzory a návrh a realizace jejich autokalibrace, návrh a vývoj kalibračního přípravku
- Návrh a implementace navigačního algoritmu v jazyce Basic Stamp
- Testy robota v rozsáhlejších prostorech / koridorech a uzpůsobení navigačního a detekčního algoritmu
- Vyhodnocení testů, pohybu v koridorech s / bez kalibrace, před / po úpravách
- Zhodnocení a závěr

Rozsah grafických prací: podle pokynů vedoucího BP

Rozsah průvodní zprávy: minimálně 35 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: PARALLAX. Robotics with the Boe-Bot: Student guide [online]. 3.0. Rocklin, CA 95765, USA, 2013 [cit. 2013-10-23]. Dostupné z: <http://www.parallax.com/sites/default/files/downloads/28125-Robotics-With-The-Boe-Bot-v3.0.pdf>

Vedoucí bakalářské práce:

Ing. Petr Bureš, Ph.D.

Datum zadání bakalářské práce:

19. srpna 2013

(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání bakalářské práce:

30. listopadu 2014

- a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia


doc. Ing. Pavel Hrubeš, Ph.D.

vedoucí
Ústavu dopravní telematiky




prof. Dr. Ing. Miroslav Svítek
děkan fakulty

Potvrzuji převzetí zadání bakalářské práce.


Vít Šklebéný
jméno a podpis studenta

V Praze dne26. srpna 2014

Čestné prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem k tomu pouze zdroje uvedené na konci práce, a to v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací. Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským.

V Praze dne 27.11.2014

.....
Vít Šklebený

Poděkování

Děkuji Ing. Petru Burešovi, Ph.D. za trpělivost a vedení mé bakalářské práce a za podnětné návrhy, které ji obohatily. Dále děkuji svým rodičům a blízkým za podporu během studia i mimo něj.

Vít Šklebený

Název práce:

Navigace robota pomocí dálkoměrných sensorů

Autor: Vít Šklebený

Obor: Inteligentní dopravní systémy

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Petr Bureš, Ph.D.

Ústav dopravní telematiky, Fakulta dopravní, České vysoké učení
technické v Praze

Konzultant: —

Abstrakt: Tato práce popisuje robota Boe-bot a ultrazvukový (PING))) senzor. Pojednává o hledání optimálního kalibračního algoritmu a jeho aplikaci pro vlastní kalibraci senzoru. Dále je zaměřena na vytváření algoritmu pro navigaci, který je testován pohybem robota v bludišti.

Klíčová slova: robot, ultrazvukový senzor, bludiště, navigační algoritmus, kalibrace

Title:

Boe-bot navigation by range sensors and their calibration

Author: Vít Šklebený

Abstract: This thesis provides an overview of the Boe-bot and the ultrasonic (PING))) sensor. The aim of this thesis is to describe the process of developing an algorithm for the calibration of the sensor and to present its application, which is then tested by navigation of the robot in a maze.

Key words: robot, ultrasonic sensor, maze, navigation algorithm, calibration

Obsah

Úvod	9
1 Robot Boe-bot	10
1.1 Programovací jazyk PBasic Stamp	11
1.2 Konstrukční a přídatné části robota Boe-bot	14
1.2.1 Přídatné součásti	16
1.3 PING))) ultrazvukový sensor	16
1.3.1 Zapojení PING))) sensoru	18
1.4 Li-ion power pack 2 cell	18
1.5 Parallax Standard Servo	19
1.6 Části základní desky	20
2 Kalibrace robota Boe-bot s ultrazvukovým PING))) senzorem	22
2.1 Kalibrační přípravek	22
2.2 Postup při kalibraci ultrazvukového PING))) sensoru	23
2.2.1 První fáze kalibrace	24
2.2.2 Druhá fáze kalibrace	28
2.2.3 Třetí fáze kalibrace	31
2.2.4 Čtvrtá fáze kalibrace	36
3 Navigace robota v bludišti	41
3.1 Definice bludiště	41
3.2 Návrh navigačního algoritmu pro pohyb v bludišti	42

3.2.1	Algoritmus pro předem neznámou cestu	43
3.2.2	Algoritmus pro navigaci robota pro předem známou cestu . . .	46
3.2.3	Srovnání algoritmů jednotlivých úloh	48
Závěr		49
Seznam použitých zdrojů		51
Přílohy		52
A Vývojové diagramy a složitější kódy		53

Úvod

Robotika je v dnešní době velmi dynamicky se rozvíjejícím odvětvím spojující několik technických oborů v jeden kompaktní celek. Právě díky tomuto přesahu do ostatních vědních oblastí je robotika tak významná a má velký potenciál pro budoucí rozvoj těchto odvětví. Jedním z těchto odvětví je bezesporu i obor dopravního inženýrství. Využití robotů v tomto oboru je nasnadě v oblasti Inteligentních dopravních systémů pro vývoj a testování efektivnosti navigačních algoritmů.

Tato práce, je orientována na využití robota s ultrazvukovým senzorem pro nalezení optimální cesty a navigaci robota v bludišti. Pro zefektivnění funkce ultrazvukového senzoru byl vyvinut autokalibrační přípravek tohoto senzoru a výsledky kalibrace byly následně porovnány s nenakalibrovanými hodnotami. Pro řešení úkolů byl vybrán produkt americké společnosti Parallax Inc. Boe-bot, který je velmi vhodný pro řešení problematiky navigace v bludištích. Boe-bot je velmi univerzální a umožňuje využívat několik sensorů, které lze i vzájemně kombinovat avšak jeho omezená paměťová kapacita často limituje aplikaci složitějších algoritmů.

Práce je dělena do třech samostatných kapitol. První se zabývá popisem robota a senzoru, rozebírá jeho jednotlivé části a funkce. Součástí první kapitoly je také popis programovacího jazyku BASIC STAMP. Druhá kapitola je věnována vývoji kalibračního přípravku pro ultrazvukový sensor, a vlastnímu procesu kalibrace, shrnuje veškeré naměřené výsledky a závěry z jednotlivých měření. Třetí kapitola popisuje zvolený přístup k navigaci robota a vývoj navigačního algoritmu pro robota pohybujícího se v pravouhlém bludišti a to po předem známé, ale i neznámé cestě.

Kapitola 1

Robot Boe-bot

Boe-Bot je produktem americké firmy Parallax Inc¹., která se specializuje na vývoj a vzdělávání v oblasti robotiky. Tato společnost má dlouholeté zkušenosti s výrobou všech součástí, ale také s vývojem výukových manuálů a postupů pro co možná nejefektivnější práci s jejími produkty.

Boe-bot, který byl použit pro tuto práci je základním robotem společnosti Parallax Inc., jehož hlavní výhodou je jeho univerzálnost. Parallax Inc. sama vyvíjí nejrůznější sensory a součástky, které jsou kompatibilní se základní sestavou Boe-bota a tak umožňují jeho efektivní užití pro specifické úkoly v nejrůznějších oborech.

V mé práci se budu zabývat použitím Boe-bota s nainstalovaným ultrazvukovým (PING))) senzorem pro detekci vzdálenosti, testování navigačního algoritmu v bludišti a kalibrací (PING))) sensoru.

Vlastní robot je složen z několika částí. Všechny tyto součásti robota budou v dalších kapitolách detailněji popsány. Jednotlivé části robota lze rozdělit na konstrukční a přídatné. Konstrukční části jsou takové, bez kterých by robot nemohl fungovat, případně se pohybovat. Jde o základní set Boe-bota od firmy Parallax, Inc., který je nabízen, a v balení s ním, kromě dílů pro složení samotného robota, jsou už jen další základní elektrotechnické součástky, které jsou potřebné pro regulaci obvodů a pro správnou funkci robota. Přídatné jsou pak ty, které jsou jakýmsi doplňkem robota pro řešení konkrétních úloh či situací. Každý tento doplněk je vždy k dispozici v balení s nutnými součástkami pro jeho instalaci na kostru Boe-bota.

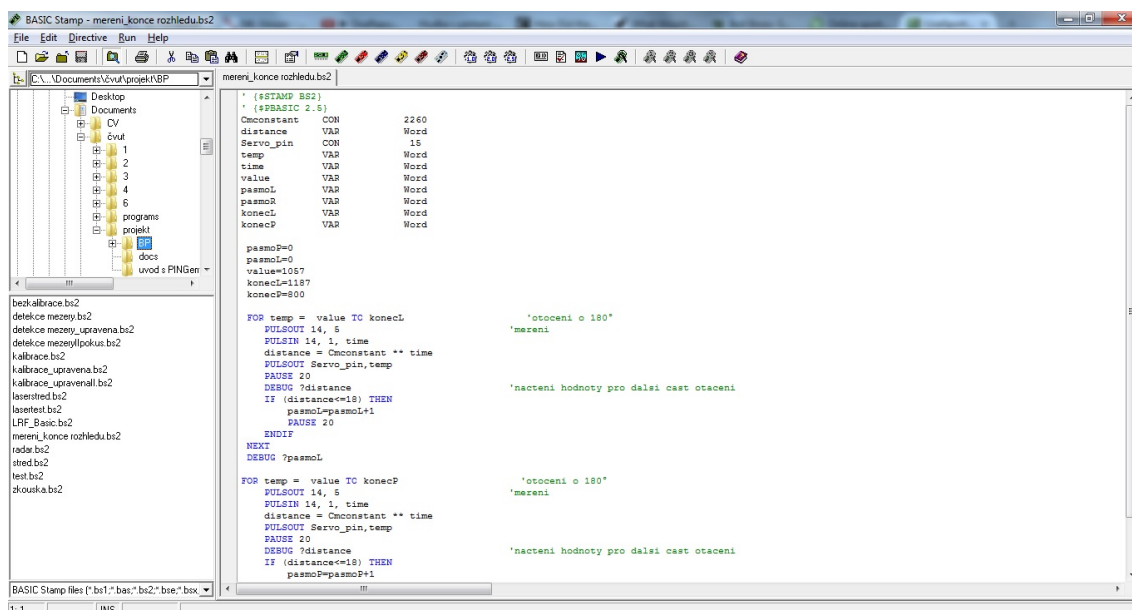
Programovací jazyk vytváří komunikační prostředí mezi uživatelem a mikroprocesorem Boe-bota, je z pohledu funkce specifický, proto nepatří ani do jedné z výše zmíněných kategorií.

¹ Parallax, Inc. <http://www.parallax.com/>

1.1 Programovací jazyk PBasic Stamp

Pro komunikaci s robotem Boe-Bot je využíván programovací jazyk z rodiny Basic ve verzi PBasic 2.5². Na stránkách firmy Parallax, Inc lze získat zdarma Basic Stamp editor, což je vhodný a přehledný software pro programování v jazyce Basic s vlastním debuggerem a možností konzolového výstupu. Basic Stamp editor umožňuje efektivní práci s robotem a snadné získávání hodnot z měření.

Jazyk PBasic 2.5 byl vytvořen především pro práci s Boe-Botem a je tedy uzpůsoben specifikům tohoto robota. Pro vytváření kódů psaných v jazyce PBasic 2.5 byl použit software Basic Stamp Editor v 2.5.3. Screenshot tohoto programu je na obrázku 1.1. V tomto softwaru je kód napsán a následně je přeložen překladačem a odeslán do paměťové jednotky, kde je následně uložen a mikroprocesorem převeden na sérii instrukcí, které robot vykonává. Boe-Bot je vybaven paměťovou jednotkou EEPROM, což je druh paměti, ze které lze data vyvolat i pokud není robot napájen, její nevýhodou je však velmi omezená kapacita dat, kterou je schopna do sebe uložit.



Obrázek 1.1: screenshot ze softwaru PBasic Stamp Editor

Jazyk PBasic stamp je velmi jednoduchý a intuitivní co do použití příkazů a vytváření jejich hierarchie.

Na počátku každého kódu je třeba deklarovat názvy a typy proměnných. V jazyce PBasic stamp 2,5 se používají 4 typy proměnných, všechny tyto typy a jejich specifikace jsou uvedeny v tabulce 1.1. Poslední variantou proměnné, která není v tabulce

²Software Basic Stamp editor a jazyk PBasic stamp jsou ochrannou známkou společnosti Parallax, Inc.

uvedena je typ proměnné CON, neboli konstanta. Příklad takové proměnné je uveden na screenshotu na obrázku 1.1. [1]

Tabulka 1.1: typy proměnných v jazyce PBasic stamp, [2]

typ proměnné	číselné hodnoty, která je proměnná schopna uložit
bit	0-1
nib	0-15
byte	0-255
word	0-65535

Programy psané pro robota Boe-bot v jazyce PBasic stamp využívají několik základních příkazů:

FOR cyklus vytváří opakovanou smyčku příkazů, které jsou prováděny stále dokola, dokud je hodnota proměnné označované jako „step value“ v mezích, které jsou definovány v závorce za příkazem FOR. Příkazy v tomto cyklu jsou tedy prováděny v zapsaném pořadí neustále, dokud je step value v mezích počáteční a koncové hodnoty. K proměnné step value je vždy na konci smyčky přičtena hodnota jedna automaticky, podobně jako je tomu v jazyce MATLAB. Celý cyklus je pak ukončen příkazem NEXT.[1]

Příklad zápisu:

```
FOR (counter = 1:1000)
příkazy obsažené v cyklu
NEXT
```

IF je standardním příkazem, který funguje naprosto stejně, jako je tomu v jiných jazycích jako např. MATLAB nebo C. Příkaz definuje podmínku pro provedení příkazů zapsaných mezi příkazem IF a ENDIF, kterým je příkaz IF ukončen. Pomocí příkazu ELSIF lze v těle příkazu IF definovat další podmínku nebo příkazem ELSE definovat příkazy, které budou provedeny, pokud není vstupní podmínka splněna. Příkaz IF je vždy ukončen příkazem ENDIF. Zápis podmínky musí být následován příkazem THEN, který ukončuje podmínkovou část příkazu IF a iniciuje část s příkazy, které budou provedeny při splnění vstupní podmínky. Spustíme-li program zapsaný níže v příkladu, bude nejdříve vyhodnocena podmínka, zda je proměnná counter větší než nula, pokud ano zobrazí se její hodnota na konzolovém výstupu a vyhodnotí se druhá podmínka, je-li zbytek po dělení dvěma roven nule, pokud ano zobrazí se na konzolovém výstupu zpráva „counter je kladné a sudé číslo“, pokud není druhá podmínka

splněna, zobrazí se na konzolovém výstupu „counter je záporný“.[1]

Příklad zápisu:

```
IF counter > 0 THEN
  DEBUG ?counter
  ELSIF (counter // 2) = 0 THEN
DEBUG CR, \uv{counter je kladné a sudé číslo}
  ELSE  DEBUG CR, \uv{counter je záporný}
ENDIF
```

DO LOOP je typ cyklu, který se opakuje neustále dokola. Bez jakékoliv vstupní podmínky. Obvykle tento cyklus končí splněním IF podmínky, ve které je použit příkaz STOP.[1]

DEBUG je příkaz, který slouží pro zobrazení hodnoty, která je uložena do proměnné, do konzolového výstupu. Příkaz umožňuje více možností zápisu, zde naznačím pouze způsob zápisu, který je v dále v práci používán. Je-li použit zápis použitý níže v příkladu a do proměnné counter je uložena hodnota 100, konzolový výstup zobrazí „counter = 100“.[1]

Příklad zápisu:

```
DEBUG ?counter
```

PULSOUT je příkazem, který slouží k dodání elektrického pulsu z napájení do daného pinu na základní desce, dále lze určit délku tohoto impulsu v mikrosekundách. Je-li zapsán příkaz shodný s příkladem níže bude na pin číslo 10 generován puls o délce 100 μ s.[1]

Příklad zápisu:

```
PULSOUT 10, 50
```

PULSIN je příkaz, sloužící k měření délky pulsu na definovaném pinu a uložení hodnoty do proměnné. Pro správnou funkci příkazu je nutné definovat typ pulsu, stoupající typ tzv. high (do příkazu po definici pinu bude zapsána hodnota 1), nebo klesající typ tzv. low (do příkazu po definici pinu bude zapsána hodnota 0). Užitím zápisu, jako v příkladu zapsaném níže, bude změřena délka pulsu typu high a výsledek bude uložen do proměnné time.[1]

Příkaz zápisu:

```
PULSIN 10, 1, time
```

PAUSE : Příkaz PAUSE definuje počet milisekund, o který bude odloženo další provádění následujících příkazů. Tento příkaz je velmi důležitý pro práci s přídatnými součástkami, jako např. Parallax standard servo, které pro svůj správný chod potřebují určitý časový rozestup mezi dodávanými elektrickými pulsy. Provedení příkazu zapsaného v příkladu odloží další provádění příkazů o 500ms.[1]

Příklad zápisu:

```
PAUSE 500
```

GOSUB je velmi užitečným příkazem, který zjednodušuje čtení kódu a správné převedení algoritmu do sledu příkazů. Příkazem GOSUB lze vyhledat sled příkazů, který byl dříve definován. Nejdříve je tedy nutné si definovat jednotlivé příkazy např. pro otočení robota o 90° a označit je slovem nebo slovy oddělenými podtržítkem a zakončenými dvojtečkou a na konec přidat příkaz RETURN. Toto pojmenování vyvolá příkaz GOSUB a příkazy budou provedeny. Tento příkaz je velmi užitečný pro zápis složitějších algoritmů, kde jednotlivé kroky obsahují více příkazů a orientace v takovém zápisu by byla obtížná. [1]

Příklad zápisu:

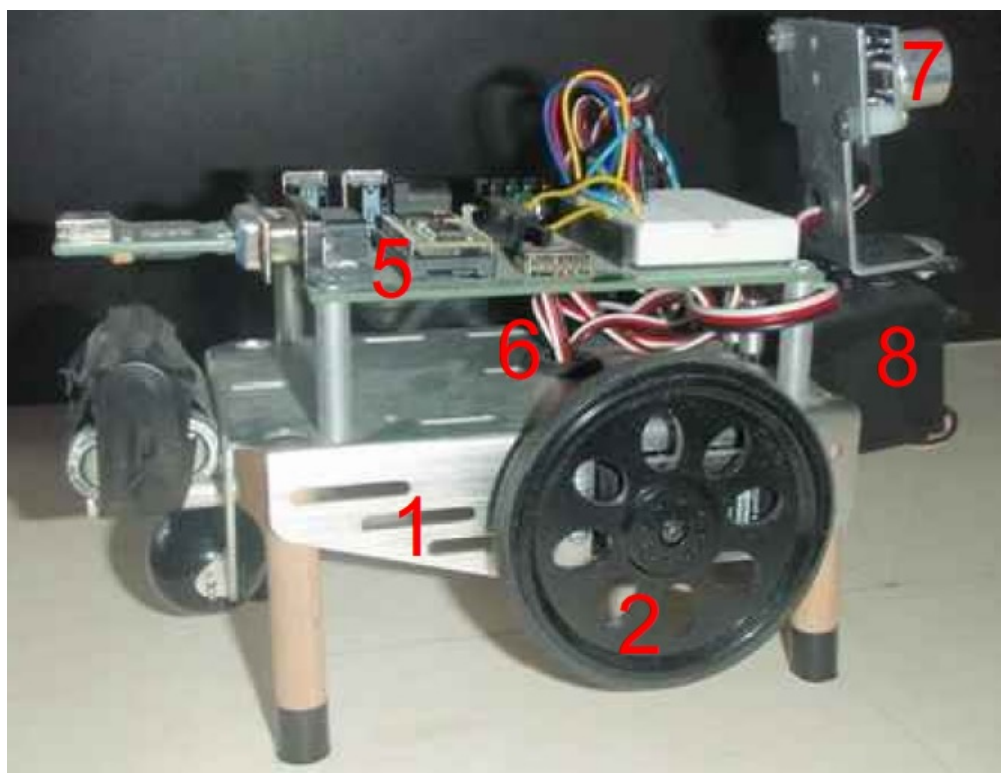
```
vystup_counter:  
    DEBUG? Counter  
    RETURN  
    GOSUB vystup_counter
```

1.2 Konstrukční a přídatné části robota Boe-bot

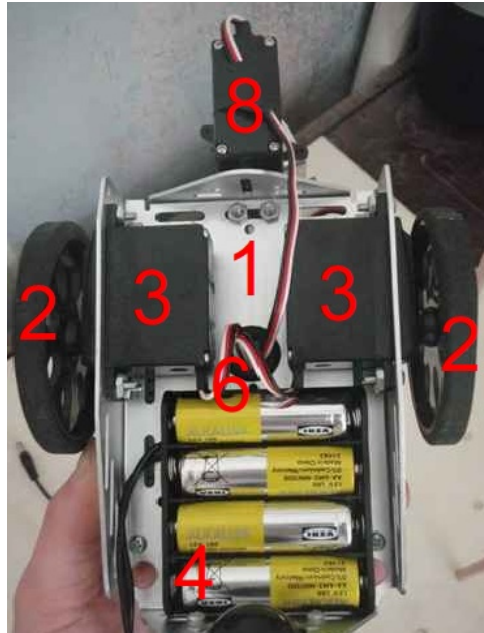
Konstrukční části robota jsou zobrazeny na bočním pohledu na robota na obrázku 1.2 a spodním pohledu na robota na obrázku 1.3 Na obrázku 1.2 i 1.3 jsou dále zobrazeny přídatné součásti Parallax standard servo (č.8) a ultrazvukový PING))) sensor (č.7). Ostatní čísla odpovídají konstrukčním součástkám popsaným níže.

1. **Hliníková konstrukce** – vytváří nosnou kostru robota, ke které jsou později přichyceny všechny ostatní součásti
2. **Pár kol** - kola o průměru 66mm díky nimž je umožněn pohyb robota, každé kolo je vybaveno gumovým proužkem na obvodu pro zamezení prokluzování na hladkých površích.

3. **Pohybová serva** – Parallax servo motors jsou servo motory zajišťující pohyb robota, každé servo je se základní deskou propojeno kabelem, které zajišťují jejich napájení.
4. **Modul s bateriemi** – nosič baterií, který je uchycen zesponu ke kostře robota, modul je uzpůsoben k připojení čtyř AA 1,5V baterií, pomocí kterých je napájena základní deska
5. **Základní deska** – tato součást bude dále popsána detailněji v samostatné podkapitole
6. **Propojovací kabely** – vedení sloužící k převodu elektrických impulsů ze základní desky do součástek případně z modulu s bateriemi do desky
7. **PING))) Ultrazvukový sensor** – tato součást bude popsána v samostatné podkapitole
8. **Parallax Standard servo** – pohybové servo pro ultrazvukový sensor



Obrázek 1.2: boční pohled na robota



Obrázek 1.3: spodní pohled na robota

1.2.1 Přídavné součásti

Přídavné součásti nejsou nezbytnými pro funkci robota, ale jsou jeho doplňky pro řešení dané úlohy.

PING))) ultrazvukový sensor vzdálenosti – tento sensor bude dále popsán v samostatné podkapitole

Standard servo motor – servo motor, který má na rozdíl od pohybových servo motorů rozsah pouze 0° - 180° , slouží k otáčení PING))) senzorem, případně laserovým LRF senzorem. [2]

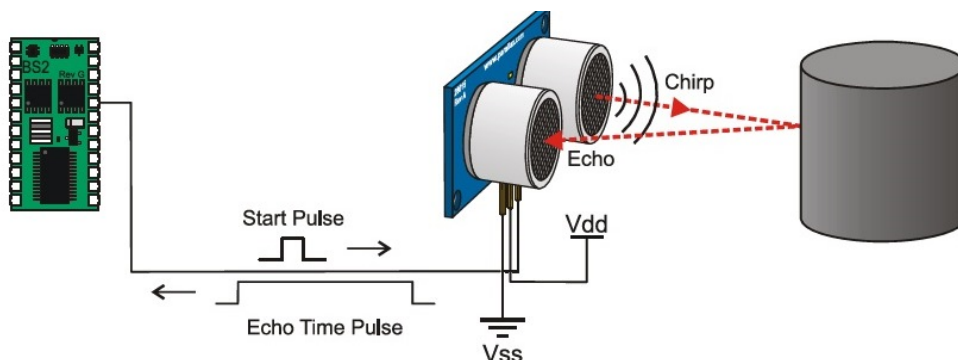
Piezelectrický reproduktor – pro vlastní řešení úlohy není tato součástka nutná, ale je využita jako indikátor počátku nějakého děje v kódu (viz. „dále“). [2]

Li-oin power pack 2cell – vylepšený napájecí modul se dvěma komorami pro dobíjecí baterie o napětí 3,7V. Tato součást nahrazuje původní napájecí modul s AA 1,5V bateriemi. [3]

1.3 PING))) ultrazvukový sensor

Ultrazvukový PING))) sensor byl v práci použit jako primární sensor a měření s tímto senzorem je hlavní náplní této práce. Tento sensor pracuje na principu

měření časové odezvy na signál, který je generován vysílačem sensoru a přijímán přijímačem. Tato odezva je dále zpracována do výpočtu vzdálenosti detekované překážky, kterou vyhodnocuje mikročip na základní desce. Schéma principu činnosti ultrazvukového PING))) sensoru je zobrazen na obrázku 1.4. [4]



Obrázek 1.4: schéma funkce PING))) sensoru, [4]

Na základě znalosti doby odezvy (t) na tento signál je určena vzdálenost (l) sensoru od překážky pomocí vzorce (1.1)

$$l = c_v \cdot t \quad (1.1)$$

kde c_v je rychlost šíření signálu ve vzduchu a je dána pomocí vzorce (1.2)

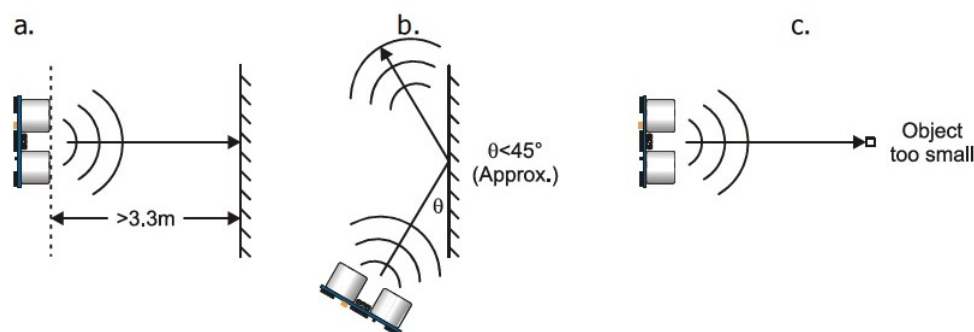
$$c_v = 331,5 + (0,6 \cdot T_c)[ms^{-1}] \quad (1.2)$$

kde T_c je teplota vzduchu ve stupnici Celsia. Tento vzorec je uveden v manuálu k ultrazvukovému sensoru PING))) vydaném na webových stránkách firmy Parallax, Inc. Jak dále uvádí relativní chyba měření senzorem se v rozsahu teplot $0^\circ C - 70^\circ C$ pohybuje mezi 11%-12%. [4]

Výsledky měření také značně ovlivňuje struktura povrchu a tvar detekovaného objektu, ale samozřejmě i úhel, pod kterým je objekt snímán. Jak výrobce uvádí, hraniční hodnota úhlu, od které již objekt nemusí být detekován, je 45° mezi kolmicí od PING))) sensoru a povrchu detekovaného objektu. [4]

Jak je patrné z obrázku 1.5 PING))) sensor je schopen detekovat překážku od vzdálenosti 3cm před senzorem do vzdálenosti 3,3m a není schopen zaznamenat příliš malé objekty. Významnou vlastností PING))) sensoru je, že nevysílá zvukové vlny pouze kolmo před sebe, ale do širokého pole. Pokud tedy pomocí serva otáčíme senzorem, sensor překážku detekuje po delší dobu, než se vlny přestanou odrážet od překážky směrem zpět k sensoru. Tato vlastnost ultrazvukového PING))) sensoru je předmětem několika měření, které jsou detailněji popsány v následující kapitole.

Parametry PING))) sensoru jsou shrnuty v tabulce 1.2. [4]



Obrázek 1.5: specifikace PING))) sensoru, [4]

Tabulka 1.2: 2parametry PING))) sensoru, [4]

veličina	hodnota	jednotka
napětí stejnosměrné	5	V
proud	30-35	mA
doba odezvy	min 115, max 18,5	$\mu s, ms$

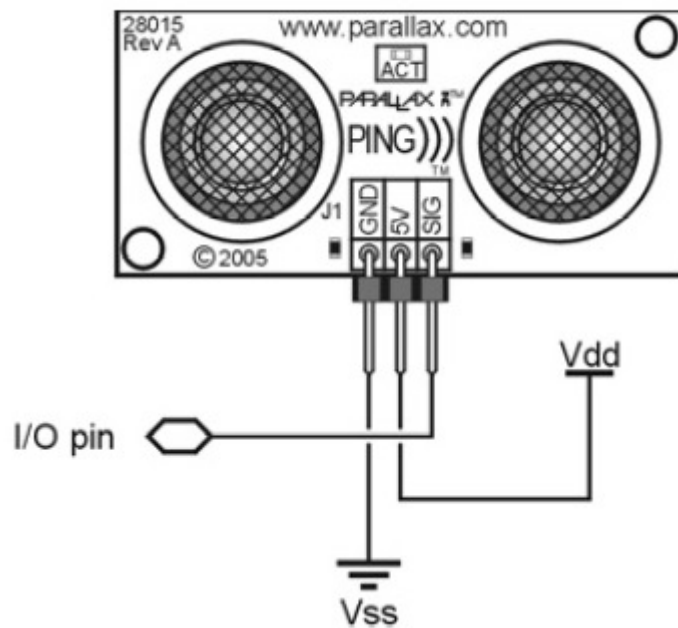
1.3.1 Zapojení PING))) sensoru

Jak již bylo uvedeno v předchozí kapitole PING))) sensor je do základní desky robota připojen trojkabelem, do portu pro trojkabely na základní desce. Schéma zapojení je patrné z obrázku 1.6. Konec označený Vdd přivádí potřebné napětí 5V pro správnou funkci PING))) sensoru z napájecího modulu, konec označený I/O pin slouží ke komunikaci s čipem na základní desce a přenáší tak informaci získanou senzorem, konec označený Vss pak slouží k uzemnění celého obvodu. Celá tato soustava je přichycena na Parallax Standard Servo, které umožňuje otáčení v rozsahu 0° - 180° . Součinnost těchto dvou prvků bude předmětem kalibrace robota na kalibračním přípravku, která bude podrobněji probrána v dalších kapitolách práce. [4]

1.4 Li-ion power pack 2 cell

Li-ion power pack 2cell je napájecím modulem, který má oproti původnímu modulu pro 4 AA 1,5V baterie výhodu, že jeho baterie je možné průběžně nabíjet a tak zaručit, že pro každé nové měření bude shodný zdroj napájení.

Tento fakt se v průběhu práce ukázal jako klíčový pro úspěšné nakalibrování sensoru, jelikož kalibrace je velmi citlivý proces, kde i menší pokles napětí ve zdroji způsoboval značné výkyvy v naměřených hodnotách.



Obrázek 1.6: schema zapojení PING))) sensoru, [4]

Tento disponuje dvěma komorami pro baterie s napětím 3,7V. Má dvě možnosti výstupu a to buď regulovaný výstup, nebo neregulovaný. Neregulovaný výstup přivádí napětí shodné s tím, které nabíjí modul. Regulované potom přivádí v úplně nabitém stavu 7,4V do robota a méně v závislosti na stavu baterií. [3]

1.5 Parallax Standard Servo

Jak bylo uvedeno Parallax Standard servo je součástí, na které je připojen PING))) sensor, a která umožňuje jeho otáčení a tím pádem i detekci objektů v rozsahu 0°-180°. Toto Servo na rozdíl od serv, která jsou použita pro otáčení kol robota Boe-bot, neumožňuje několika násobné otočení a má významně jemnější chod, tedy vyšší poměr počtu pulsů na úhel natočení. Pro kalibraci je tento fakt velmi významný, jelikož během kalibrace půjde právě o nalezení závislostí mezi počtem pulsů na úhel natočení při průchodu celým rozsahem Standard serva.

Pulsový rozsah serva je 1000 pulsů, během kterých se servo otočí o celých 180°. Tento rozsah je dán rozmezím 200-1200. V kódech, které budou dále představeny, jsou nakalibrované hodnoty vyšší o počáteční hodnotu 200 oproti hodnotám uvedeným v tabulkách, které shrnují výsledky jednotlivých fází kalibrace.

Zapojení Standard serva k základní desce je obdobně jako u PING))) sensoru pro-

vedeno pomocí trojkabelu zavedeného do portu pro trojkabely na základní desce.

Parametry serva jsou shrnuty v tabulce 1.3. [5]

Tabulka 1.3: parametry Standard serva, [5]

veličina	maximální hodnota	jednotka
napětí	4-6	V
proud	+140/-50	mA
délka pulsu/interval pulsu	0.75-2,25 / 20	ms
minimální mezidobí mezi pulsy	20	ms

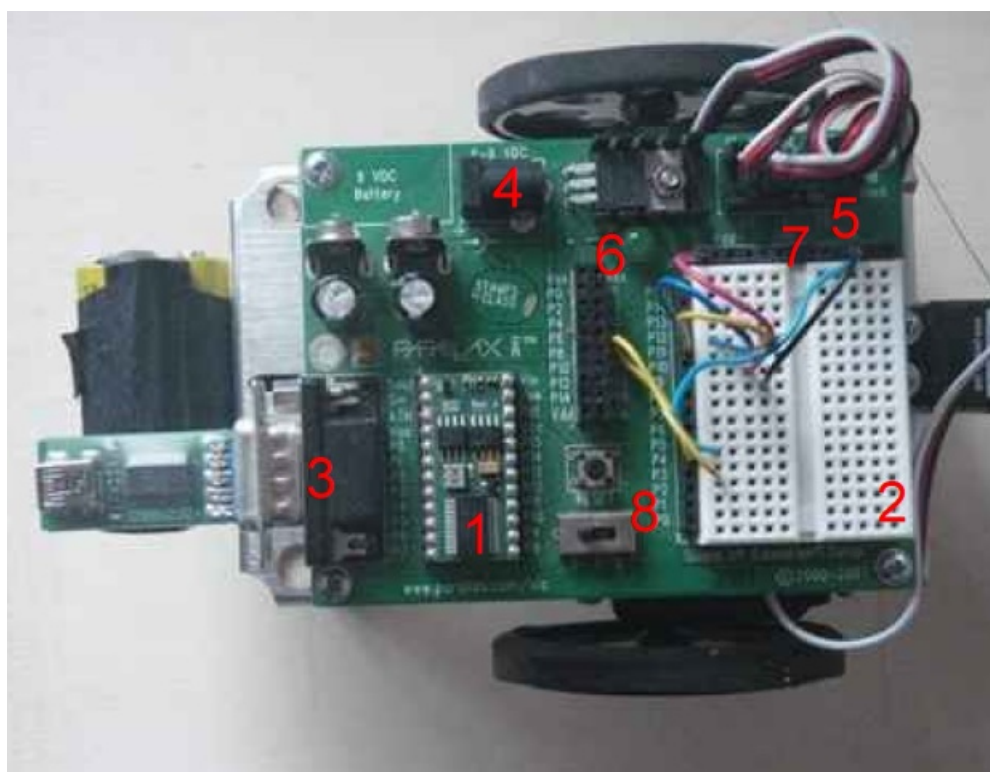
1.6 Části základní desky

Základní deska s mikročipem (jednočipovým počítačem) je naprosto nezbytnou součástí Boe-Bota, bez které by jeho fungování nebylo možné. Deska je pomocí distančních šroubů přichycena ke kostře robota a kabely propojena s ostatními součástmi robota.

Vlastní deska je systémem několika součástek: Celá deska je na obrázku 1.7, kde jsou očíslovány jednotlivé součásti desky, které jsou dále popsány. [2]

1. **Mikročip (jednočipový počítač)** – jednotka vybavená překladačem jazyka PBASIC (viz. „dále“), ve které se vyhodnocují veškeré rozhodovací procesy a probíhá překlad kódu. [2]
2. **Breadboard sockets** – část desky určená k připojení jednoduchých elektro-technických součástek (rezistorů, diod, kondenzátorů či jednoduchých propojovacích kabelů) potřebných k vytvoření obvodů a správnému přenosu elektrických impulsů, celkem má tato část desky 16pinů pro připojení součástek. [2]
3. **VGA/USB port** – slouží pro propojení desky s počítačem a přenos kódu programu a výstupů do/z mikročipu pomocí propojovacího kabelu. [2]
4. **Port pro kabel s napájením** – zajišťuje napájení základní desky z modulu s bateriemi. [2]
5. **Porty pro trojkabely³** – pomocí těchto portů jsou připojeny k desce kabely od jednotlivých serv a od PING))) sensoru, celkem je možné na desku připojit

³Trojčabel se skládá ze tří částí: 1. napájecí, 2. k uzemnění obvodu, 3. k propojení s daným pinem



Obrázek 1.7: základní deska robota s popsanými částmi

až 4 tyto trojkabely, jeli nutné připojit ještě další součásti, lze využít volné piny na Breadboard sockets. [2]

6. **LED dioda** – dioda indikující zapnutí/vypnutí napájení základní desky. [2]
7. **Přepínač pro použitý typ napájení** – tato součást základní desky, má dvě polohy pro dva rozlišení možných typů napájení desky. Vin poloha slouží pro napájení pomocí základního modulu dodávaného v setu s Boe-botem se 6V, Vdd poloha pak slouží pro použití zdrojů s vyšší hodnotou napětí. [2]
8. **Ovládací tlačítka** – RESET tlačítka – umožňující resetování programu
 Polohovací spínač – tři polohy
 - 0 – robot vypnut
 - 1 – robot zapnut, porty s trojkabely odpojeny
 - 2 – robot zapnut s napájením do všech částí

Kapitola 2

Kalibrace robota Boe-bot s ultrazvukovým PING))) sensorem

Kalibrace robota Boe-Bot, je jedním z hlavních úkolů celé práce. Cílem vlastní kalibrace je definování poměru počtu pulsů nutných k natočení serva o jeden stupeň. Vzhledem k tomu, že se servo v celém svém rozsahu chová značně nelineárně, nestačí pouze rozdělit jeho pulsový a úhlový rozsah na jeden poměr a tuto hodnotu následně násobit žádaným úhlem. Přestože zde nastiňuji, že tyto hodnoty nejsou žádoucí a nejsou přesné, budou ve vlastní kalibraci brány, jako výchozí hodnoty, které se budou kalibrací opravovat a korigovat do co možná nejpřesnějších hodnot, které budou odpovídat danému úhlu. Tyto Výsledné nakalibrované hodnoty budou srovnány s původními a bude stanoven rozdíl mezi nimi. Tento rozdíl mimo jiné bude sloužit jako důkaz nelineárního chodu serva během průchodu celým jeho úhlovým rozsahem. Veškeré postupy během kalibrace jsou popsány v následujících kapitolách.

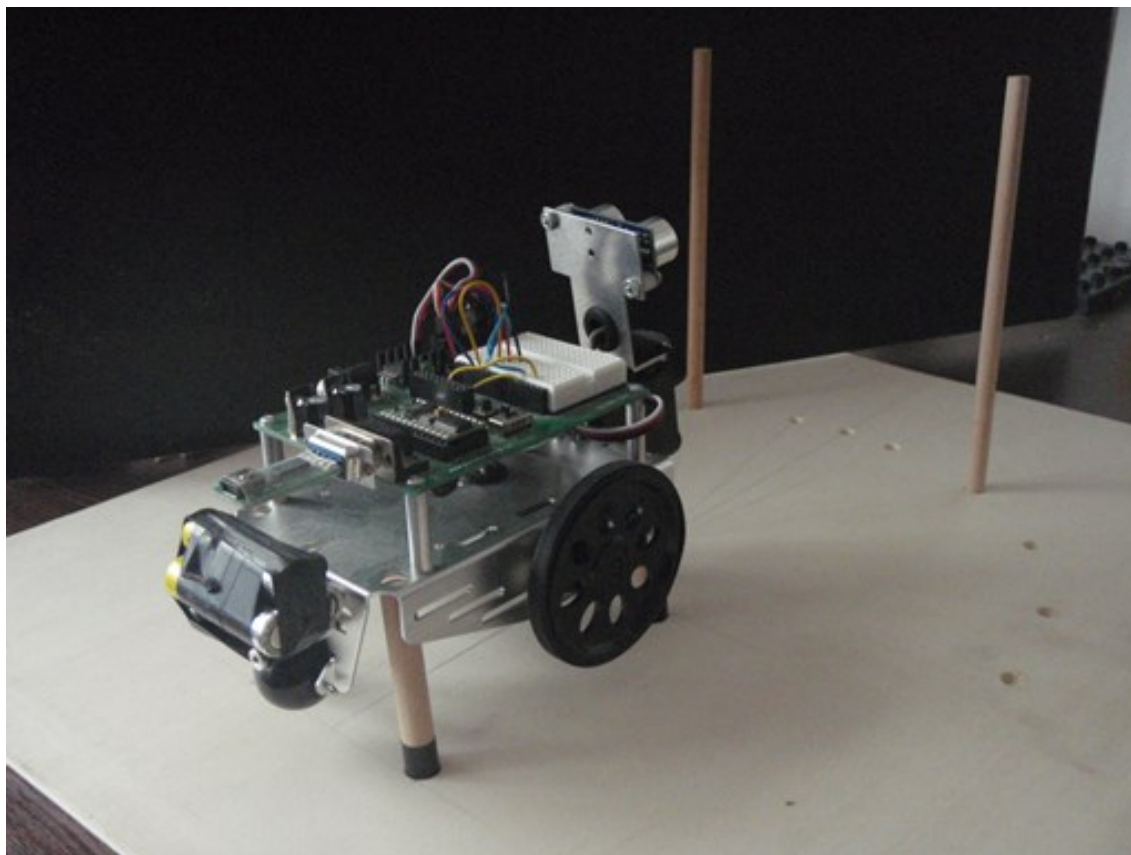
2.1 Kalibrační přípravek

Kalibrační přípravek bylo nutné vytvořit tak, aby zajistil shodné podmínky pro všechna měření, a jejich výsledky bylo možné mezi sebou porovnávat, a také takovým způsobem aby byla zajištěna bezproblémová práce robota během měření a byl zajištěn bezpečný průchod celým rozsahem serva robota.

Kalibrační přípravek byl vytvořen pomocí 5mm tlusté modelářské překližkové desky, do které byly navrtány otvory pro zasazení dřevěných tyček o průměru 8 mm, které byly rozmístěny na desce v půlkruhu o poloměru 18 cm se středem ve středu otáčení

serva s ultrazvukovým PING))) senzorem. Tyto dřevěné tyčky byly rozmístěny v definovaných úhlech 0° , 20° , 40° , 60° , 80° , 90° , 100° , 120° , 140° , 160° a 180° . Schéma kalibrační desky je zobrazeno na obrázku 2.2.

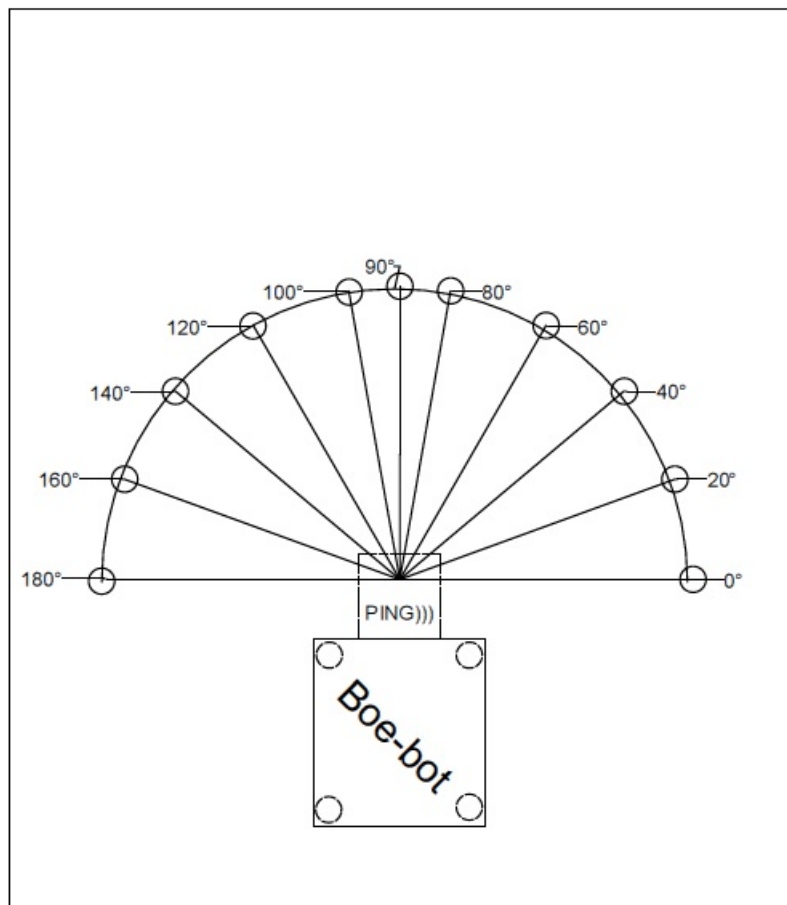
Dále byly do desky navrtány 4 otvory pro tyčky o průměru 9mm, které sloužily k upevnění robota ve shodné poloze pro každé měření. Tyto kolíky také držely robota několik milimetrů nad povrchem desky, aby byl eliminován vliv úvodního natočení kol po spuštění napájení do robota na jeho polohu na kalibračním přípravku. Robot zasazený v desce je zobrazen na obrázku 2.1.



Obrázek 2.1: robot v měřící poloze na kalibračním přípravku

2.2 Postup při kalibraci ultrazvukového PING))) sensoru

Proces kalibrace procházel vývojem po celou dobu jejího provádění, vzhledem ke komplikacím, které během plnění tohoto úkolu nastávaly. Celá kalibrace je rozdělena do několika fází podle charakteristik daných měření, které byly v rámci fází prováděny.



Obrázek 2.2: schéma kalibrační desky

2.2.1 První fáze kalibrace

V první fázi kalibrace probíhalo měření dle následujícího algoritmu. Servo je uvedeno do krajní polohy svého pulsového i úhlového rozsahu, tak aby se z této polohy mohlo pohybovat pouze jedním směrem. (pro všechny měření byl zvolen pohyb proti směru hodinových ručiček, tak aby byl v cyklech v kódech kladný nárůst hodnot počtu opakování cyklu, přestože robot je schopen vykonávat i pohyb opačný). Otáčení servem a zároveň načítání počtu pulsů do proměnné. Pokud detekuje sensor překážku, do další proměnné načte počet pulsů, během kterých tuto překážku detekuje. Vývojový diagram tohoto algoritmu je znázorněn na obrázku A.1.

Tento algoritmus byl spuštěn pro vyhledání překážky v každém definovaném úhlu na desce zvlášť, jelikož při prvním spuštění algoritmu bylo zjištěno, že při rozmístění kolíků po 20° ve vzdálenosti 18 cm od středu otáčení serva s PING))) senzorem, sensor nebyl schopen detekovat překážky jednotlivě, ale zaznamenával jednotlivé kolíky jako celistvou překážku v rozsahu 0°-180°.

Kód pro první fázi kalibrace

Vývojový diagram pro první fázi kalibrace je na obrázku A.1. Kód pro první fázi kalibrace je zobrazen na obrázku 2.3.

```
counter=0
value=200

FOR temp = value TO 600
  PULSOUT 14, 5
  PULSIN 14, 1, time
  distance = Cmconstant ** time
  PULSOUT Servo_pin,temp
  PAUSE 20
  DEBUG ?distance
  IF (distance<=18) THEN
    pasmo=pasmo+1
  ELSE counter=counter+1
  ENDIF
NEXT
DEBUG ?counter
DEBUG ?pasmo
STOP
```

Obrázek 2.3: kód pro první fázi kalibrace

Pro správný chod výše popsaného algoritmu bylo nutné zavést do kódu tři proměnné. Do proměnné distance je načítána vzdálenost zaznamenaná senzorem a je vypočtena dle vzorce, který byl doporučen výrobcem v manuálu pro ovládání PING))) sensoru (viz. Kapitola 1.1.3 PING))) ultrazvukový sensor). Tato proměnná bude zavedena ve všech kódech zmíněných v této práci, proto nebude již dále popisována. Další proměnnou je counter, která zaznamenává počet pulsů, který je třeba dodat servu k natočení do blízkosti překážky, aby ji sensor zaznamenal. Poslední proměnnou je pasmo, kam je načten počet pulsů, po které sensor zaznamenává vzdálenost překážky. Kód obsahuje jeden FOR cyklus, během kterého je provedeno kontrolní měření vzdálenosti a servu je dodán puls pro otočení do další polohy, v cyklu je obsažena IF podmínka, která kontroluje, zda je hodnota vzdálenosti změřená senzorem menší nebo rovna 18 cm. Pokud ano, znamená to, že sensor zaznamenává překážku a do proměnné pasmo je načten jeden puls.

Výsledky první fáze kalibrace

Po provedení kalibrace pro každý úhel zvlášť byly pro každý úhel získány hodnoty počtu pulsů k začátku pole, kde sensor detekuje překážku, a velikosti tohoto pole opět v počtu pulsů. Předpokládáme, že sensor detekuje překážku o stejný počet pulsů dříve, jako později po průchodu kolem překážky. Tedy hodnotu velikosti pole, kdy sensor překážku zaznamenal, dělíme dvěma a přičteme ji k hodnotě počtu pulsů k poli detekce. Takto získáme nakalibrovanou hodnotu definovaného úhlu. Výsledky úvodního měření první fáze kalibrace jsou shrnuty v tabulce 2.1:

Tabulka 2.1: nakalibrované hodnoty z úvodního měření první fáze kalibrace

úhel[°]	nakalibrovaná hodnota [počet pulsů]	úhel[°]	nakalibrovaná hodnota [počet pulsů]
20	127	100	510
40	231	120	588
60	332	140	686
80	411	160	762
90	452	180	859

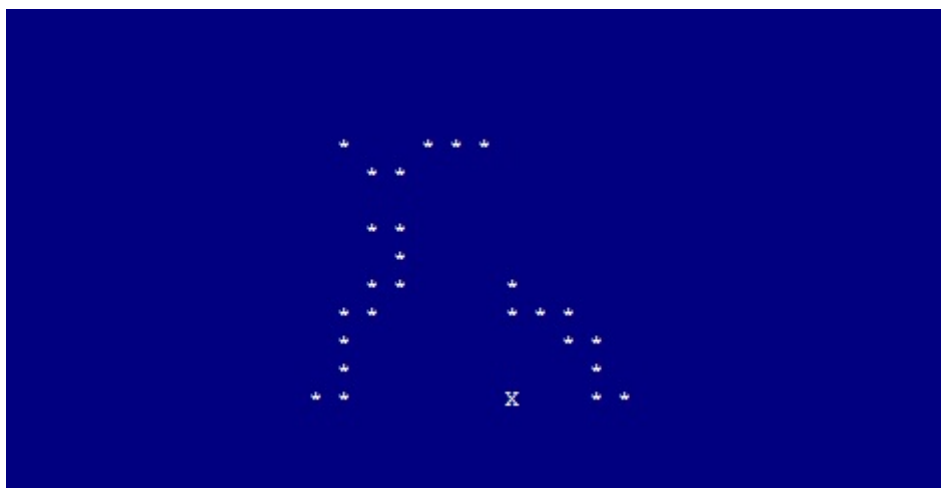
Závěry z první fáze kalibrace

Během této fáze kalibrace byly stanoveny základní hodnoty pro pozdější propracovanější kalibrační algoritmus. Dále bylo zjištěno, že servo se chová značně nelineárně ve svém celém rozsahu, ale především na jeho konci v okolí 180° jeho rozsahu. Pro hodnotu 0 pulsů bylo servo skutečně v počátku svého úhlového rozsahu, tedy 0°, ale úhlu 180° odpovídala hodnota 859 pulsů (nikoliv 1000 pulsů jak je popsáno v manuálu výrobce), a oproti informacím udávaným výrobcem nebyl rozsah 0°-180°, ale 0°- cca 190°.

Pokud bylo servo spuštěno ve svém celém rozsahu 1000 pulsů, přibližně posledních 20 pulsů nemělo vliv na otočení, ale servem pouze škubalo. Byl tedy dosažen mechanický limit serva. Z tohoto důvodu byla hodnota pro úhel 180° pro všechny ostatní měření stanovena na výsledek z první fáze kalibrace, tedy 859 pulsů.

Původní předpoklad byl, že ultrazvukový (PING)) sensor bude schopen zaznamenat každý kolík na kalibrační desce zvlášť a rozestupy po 20° budou dostačující, aby nedocházelo k „rušení“ signály odraženými se od sousedních kolíků. Avšak po prvním spuštění kalibračního algoritmu s kolíky umístěnými na všech možných pozicích, bylo dle výsledků zřejmé, že k rušení dochází, jelikož sensor zaznamenal kolíky jako

souvislou překážku ve vzdálenosti 18 cm po celou dobu průchodu rozsahem 0° - 180° . Z tohoto důvodu byl použit kód pro radarový konzolový výstup, aby byla celá situace jasnější. Příklad konzolového výstupu kódu s radarem je uveden na obrázku 2.4, kde velký křížek znázorňuje polohu robota v síti a menší křížky označují detekované překážky.



Obrázek 2.4: konzolový výstup kódu s radarem

Tento obrázek znázorňuje detekci dvou kolíků ve 20° a 80° stupních na kalibrační desce, které vytvářejí shluk křížků v pravé části obrázku. V levé části obrázku je detekována stěna na hranici stolu, kde byla deska položena. V kódu s radarem dochází k většímu otočení mezi jednotlivými měřeními, než je tomu v kódu pro jednotlivé fáze kalibrace.

První fáze kalibrace dále ukázala nelineární nárůst počtu pulsů na natočení serva o jeden stupeň úhlu. Tento fakt, fakt bylo třeba dále prostudovat a identifikovat rozdíly během průběhu celým úhlovým rozsahem serva. Pro názornost jsou podrobnější výsledky první fáze kalibrace shrnuty v tabulce 2.2. Kde první sloupec udává úhel, ve kterém se nachází detekovaná překážka, druhý sloupec udává nenakalibrovanou hodnotu pro tento úhel pro pulsový rozsah serva udávaný výrobcem (0-1000 pulsů), třetí sloupec udává nenakalibrovanou hodnotu se změněným pulsovým rozsahem (0-859 pulsů), čtvrtý sloupec udává nakalibrovanou hodnotu z první fáze kalibrace, v pátém sloupci je rozdíl nakalibrované a nenakalibrované hodnoty se změněným pulsovým rozsahem, šestý sloupec udává počet pulsů k následující překážce při průchodu serva od 0° do 180° jeho úhlového rozsahu a v posledním sedmém sloupci tabulky se nachází hodnota poměru jednoho stupně úhlu, na puls dodaný servu.

Z tabulky je jasně patrné, že vliv elektrického pulsu, který je servu dodán ze zdroje, na natočení serva se liší v průběhu jeho otáčení. Dále je zřejmé, že nelze jednoznačně identifikovat pravidelnost v chování serva v závislosti na úhlu jeho natočení.

Tabulka 2.2: výsledky druhého měření první fáze kalibrace

1	2	3	4	5	6	7
úhel kolíku [°]	nenakalibrované hodnoty, bez změny rozsahu [počet pulsů]	nenakalibrované hodnoty, se změnou rozsahu [počet pulsů]	nakalibrované hodnoty [počet pulsů]	rozdíl nakalibrované hodnoty a nenakalibrované hodnoty [počet pulsů]	počet pulsů k další překážce [počet pulsů]	poměr [° úhlu/puls]
0	0	0	0		127	0,157
20	111,1	95,4	127	31,6	104	0,192
40	222,2	190,9	231	40,1	101	0,198
60	333,3	286,3	332	45,7	79	0,253
80	444,4	381,8	411	29,2	41	0,244
90	500,0	429,5	452	22,5	58	0,172
100	555,6	477,2	510	32,8	78	0,256
120	666,7	572,7	588	15,3	98	0,204
140	777,8	668,1	686	17,9	76	0,263
160	888,9	763,6	762	-1,6	97	0,206
180	1000	859,0	859	0,0		

Výsledkem první fáze kalibrace je seznámení se se specifiky PING))) sensoru a Parallax Standard serva a nutnost vytvoření dalšího propracovanějšího kalibračního algoritmu.

2.2.2 Druhá fáze kalibrace

Cílem druhé fáze kalibrace bylo zpřesnit výsledky z fáze první a objasnit některé nesrovnalosti, které byly po první fázi evidentní. Těmito nesrovnalostmi je myšlena především různá délka pásma, kde sensor detekoval překážku, během průchodu rozsahem 0°-180°.

Tento problém byl řešen metodou měření počtu pulsů, kde sensor zaznamenává

překážku, od kolíku směrem na obě dvě strany, kde jako hodnota pozice středu kolíku byla stanovena nakalibrovaná hodnota z první fáze kalibrace. Tato metoda byla použita pro všechny definované pozice na desce, kromě prvního kolíku v 0°, pro který je tato metoda neúčinná, jelikož není možné se pohybovat na obě dvě strany kolíku. Vývojový diagram druhého kalibračního algoritmu je znázorněn na obrázku A.2, kód je na obrázku 2.5.

```
pasmoP=0
pasmoL=0
value=1057
konecL=1187
konecP=800

FOR temp = value TO konecL
  PULSOUT 14, 5
  PULSIN 14, 1, time
  distance = Cmconstant ** time
  PULSOUT Servo_pin,temp
  PAUSE 20
  DEBUG ?distance
  IF (distance<=18) THEN
    pasmoL=pasmoL+1
    PAUSE 20
  ENDIF
NEXT
DEBUG ?pasmoL

FOR temp = value TO konecP
  PULSOUT 14, 5
  PULSIN 14, 1, time
  distance = Cmconstant ** time
  PULSOUT Servo_pin,temp
  PAUSE 20
  DEBUG ?distance
  IF (distance<=18) THEN
    pasmoP=pasmoP+1
    PAUSE 20
  ENDIF
NEXT
DEBUG ?pasmoP
STOP
```

Obrázek 2.5: kód pro druhou fázi kalibrace

V kódu je na počátku deklarováno pět proměnných. Uvedený příklad je kódem pro měření posledního kolíku na kalibrační desce, pro který je nakalibrovaná hodnota z první fáze 857 pulsů. Z tohoto důvodu je do proměnné value vložena hodnota 1057, která představuje polohu kolíku. Od této hodnoty se servo otáčí směrem po směru hodinových ručiček až dosáhne konce svého rozsahu, který představuje hodnota

1187 vložená do proměnné konecL. Během tohoto pohybu je do proměnné pasmoL načítán počet pulsů, během kterých byla zaznamenána vzdálenost menší než 18 cm, tedy vzdálenost kolíku. Na konzolový výstup je také zobrazována proměnná distance, která zobrazuje aktuální vypočtenou vzdálenost zaznamenanou PING))) senzorem. Po ukončení natáčení serva po směru hodinových ručiček je příkazem DEBUG zobrazena hodnota proměnné pasmoL a servo se natáčí zpět do polohy kolíku a celý cyklus je proveden znovu tentokrát s natáčením proti směru hodinových ručiček a použity jsou proměnné s písmenem P.

Výsledky druhé fáze kalibrace

Výsledky tohoto měření jsou shrnuty v tabulce 2.3

Tabulka 2.3: naměřené hodnoty druhé fáze kalibrace

počet pulsů ve směru rozsahu serva	pozice kolíku [°]	počet pulsů proti směru rozsahu serva
158	20	114
144	40	120
133	60	121
118	80	98
144	90	102
125	100	81
126	120	93
118	140	98
128	160	114
123	180	122

Cílem této metody bylo vyhlazení a zpřesnění hodnot umístění kolíků z první fáze kalibrace. Z tabulky je zřejmé, že všechny hodnoty z první fáze kalibrace jsou posunuty do směru proti pohybu serva, jelikož při tomto měření jsou všechny hodnoty na straně proti pohybu serva nižší než hodnoty po směru pohybu serva. Z tohoto důvodu jsem provedl přepočty hodnot a určil výslednou nakalibrovanou hodnotu druhé fáze kalibrace jako součet původní hodnoty z první fáze kalibrace a střední hodnoty rozdílu hodnot v protisměru a po směru otáčení serva. Tedy pro příklad kolíku v úhlu 60° je kalibrovaná hodnota z první fáze 332 pulsů, po směru otáčení serva bylo naměřeno 133 pulsů a protisměru otáčení serva 121 pulsů. Rozdíl je tedy 12 pulsů a střední hodnota 6 pulsů. Výsledná hodnota z druhé fáze kalibrace pro kolík v úhlu 60° na kalibrační desce je tedy $332+6=338$ pulsů.

Výsledky druhé fáze kalibrace jsou shrnuty v tabulce 2.4.

Tabulka 2.4: výsledky druhé fáze kalibrace

úhel kolíku [°]	20	40	60	80	90	100	120	140	160	180
nakalibrovaná hodnota [počet pulsů]	149	243	338	421	473	532	604,5	696	769	859,5

Závěry z druhé fáze kalibrace

Při srovnání výsledků prvního měření druhé fáze kalibrace s výsledky z první fáze kalibrace je zřejmé, že došlo ke zpřesnění hodnot, ve smyslu prvního kalibračního algoritmu. Hodnoty, které jsou shrnuty v tabulce 2.4, sice již nejsou naprostým středem pásma, kde byla detekována překážka a na první pohled tedy nesplňují prvotní myšlenku kalibračního algoritmu, která byla stanovit pásmo detekce kolíku a jeho střed prohlásit za nakalibrovanou hodnotu pro daný definovaný úhel na kalibrační desce. Výsledky prvního měření druhé fáze kalibrace jsou však více průkazné jelikož zaručují, že sensor, natočen servem příslušným počtem pulsů, bude detekovat překážku při pohybu od kolíku oběma směry po stejný počet pulsů a lze tak předpokládat, že tato hodnota přesněji určuje definovaný úhel na kalibrační desce než prostý střed pásma detekce kolíku.

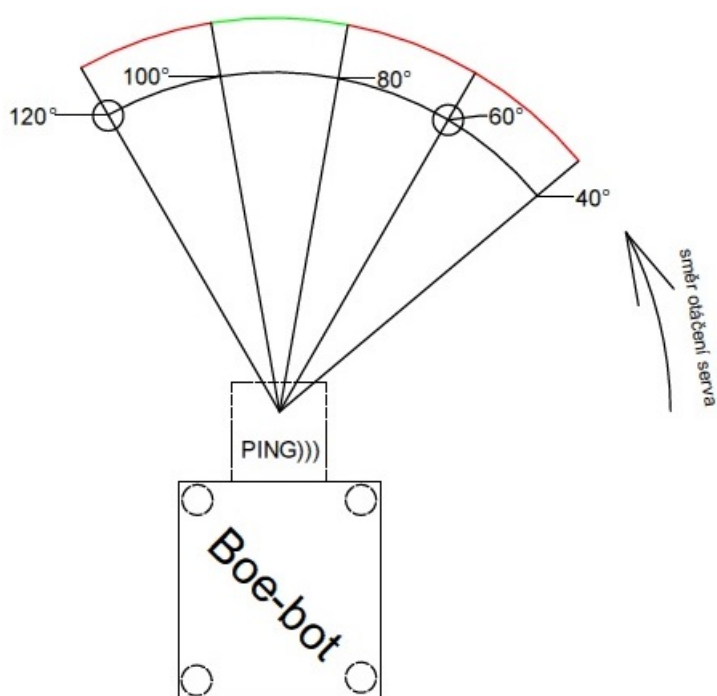
2.2.3 Třetí fáze kalibrace

Třetí fáze kalibrace měla za úkol objasnit chování sensoru při hledání prostoru mezi překážkami. Kalibrační deska byla původně konstruována pro definování počtu pulsů nutných k natočení serva se senzorem do určitého předem definovaného úhlu, ale vzhledem ke konstrukci desky je na ni možné provádět i měření mezery mezi těmito překážkami. Právě výsledky tohoto měření budou velice zásadní pro tvorbu navigačního algoritmu pro bludiště což je druhým úkolem této práce. Z tohoto důvodu bylo nutné klást velký důraz na výsledky tohoto měření.

Jak již bylo nastíněno v předchozích kapitolách (ultrazvukový PING))) sensor neměří vzdálenost bodově jako laserový sensor LRF, ale na základě vysílání ultrazvukových vln do okolí sensoru a měření odezvy na tento signál. Z tohoto důvodu často dochází k rušení mezi odezvami od kolíků, které jsou blízko u sebe. Opakovaným měřením bylo zjištěno, že šířka pásma, ve kterém sensor překážku zaznamenává, je přibližně 40°. Tedy je-li kolík postavený v úhlu 40° na kalibrační desce, bude zaznamenán již při natočení sensoru přibližně na dvacátý stupeň kalibrační desky a sensor bude vracet hodnotu vzdálenosti tohoto kolíku po celou dobu průchodu až do přibližně šedesátého stupně na kalibrační desce. Tato informace je velmi zásadní právě pro

měření mezery mezi kolíky. Z výše uvedeného je zřejmé, že měřená mezera na desce musí být větší než 40° .

Kalibrační deska umožňuje rozestavit kolíky po 20° (jen mezi úhly 80° - 100° po 10°), proto jsem pro měření mezery určil mezeru, která je na kalibrační desce široká 60° a je ve stálé vzdálenosti 18 cm od robota. Během měření mezery byl hledanou veličinou počet pulsů, během kterých sensor zaznamenává objekt ve vzdálenosti větší než 18 cm. Hodnota 18 cm byla stanovena, jako hraniční, jelikož sensor při natáčení serva směrem k překážce zaznamenává vzdálenosti větší a dále „nabíhá“ po určitý malý počet pulsů na vzdálenost překážky (tento proces trvá do deseti pulsů) dále po otočení serva se senzorem od překážky opět vzdálenost po podobný počet pulsů nabíhá do vzdálenosti větší než je překážka. Celá situace je znázorněna na obrázku 2.6.



Obrázek 2.6: schéma třetí fáze kalibrace

Jak bylo řečeno v předchozím odstavci, měření ukázala, že pole, po které sensor zaznamenává překážku, bylo široké přibližně 40° . Toto pole je na obrázku 2.6 znázorněno červenou linií v okolí kolíku umístěného v 60° na kalibrační desce. Zelená linie znázorňuje pole, po které sensor zaznamenává vzdálenost větší než je vzdálenost překážky (v tomto případě stanovená hodnota 18 cm), právě šířka tohoto pole je hledanou veličinou ve třetí fázi kalibrace. Následná druhá červená linie znázorňuje pole, po které již sensor zaznamenává kolík umístěný ve 120° na kalibrační desce,

který je druhou překážkou ohraničující měřenou mezeru.

Kód pro třetí fázi kalibrace Vývojový diagram kalibračního algoritmu třetí fáze kalibrace je znázorněn na obrázku A.3, kód pro program je zobrazen na obrázku 2.7.

```
do_18=0
pasmo=0
value=200
zacatek:
FOR temp = value TC 1200
  PULSOUT 14, 5
  PULSIN 14, 1, time
  distance = Cmconstant ** time
  PULSOUT Servo_pin,temp
  PAUSE 20
  do_18=do_18+1
  IF (distance<=18) THEN
    PAUSE 200
    DEBUG ?distance
    value = temp +1
    FOR temp = value TC 1200
      PULSOUT 14, 5
      PULSIN 14, 1, time
      distance = Cmconstant ** time
      PULSOUT Servo_pin,temp
      PAUSE 20
      do_18=do_18+1
    IF (distance>18)THEN
      DEBUG ?do_18
      value=temp +1
      FOR temp = value TC 1200
        PULSOUT 14, 5
        PULSIN 14, 1, time
        distance = Cmconstant ** time
        PULSOUT Servo_pin,temp
        PAUSE 20
        pasmo=pasmo+1
        IF (distance<=18) THEN
          DEBUG ?pasmo
          STOP
        ENDIF
      NEXT
    ENDIF
  NEXT
ENDIF
NEXT
PAUSE 50
NEXT
```

Obrázek 2.7: kód pro třetí fázi kalibrace

Pro provedení detekce mezery ve smyslu popisu z předchozí kapitoly, bylo nutné zavést dvě proměnné. A to proměnné value a pasmo. Proměnná do_18 je pouze kontrolní proměnnou a pro vlastní výsledek měření nemá žádný význam, použita však byla, jako kontrolní. Do této proměnné jsou načítány pulsy, které bylo nutno

dodat servu, aby se natočilo do místa, kde již sensor detekuje vzdálenost větší než 18 cm a tedy počátek měřené mezery. Proměnná `value` slouží k předání hodnot mezi **FOR** cykly tak, aby byl zajištěn hladký průběh otáčení servem. Nejdůležitější proměnnou tohoto kódu je `pasmo`, která obsahuje cílovou informaci tohoto měření a to počet pulsů, během kterých sensor zaznamenává vzdálenost větší než 18 cm.

První část kódu je **FOR** cyklus, během kterého je servu dodán jeden puls, dochází k natáčení směrem k prvnímu hraničnímu kolíku a zároveň je prováděno kontrolní měření vzdálenosti. První hodnota vzdálenosti menší nebo rovna 18 cm je vstup do druhého **FOR** cyklu kdy se děje to samé, ale v tomto cyklu je již senzorem zaznamenávána vzdálenost prvního hraničního kolíku. Další **IF** podmínka je vstupem do třetího **FOR** cyklu, kdy je opět dodán jeden puls servu k dalšímu otáčení, je prováděno kontrolní měření, ale již nabývá na hodnotě i proměnná `pasmo`. V okamžiku, kdy je kontrolním měřením dosažena hodnota opět menší než 18 cm, je program zastaven a příkazem **DEBUG?** `pasmo` je na konzolovém výstupu zobrazena výsledná hodnota proměnné `pasmo`.

Výsledky třetí fáze kalibrace

V rámci tohoto měření byly získány hodnoty pro sedm mezer v celém úhlovém rozsahu serva, kdy se mezera pro každé další měření posunula o 20°. Celá třetí fáze byla provedena ve dvou měřeních, celkové výsledky měření třetí fáze kalibrace shrnuje tabulka 2.5. V tabulce 2.5 je v prvním sloupci uveden rozsah měřené mezery ve stupních úhlu kolíků ohraničujících měřenou mezeru, ve druhém sloupci jsou hodnoty vyjadřující velikost měřené mezery v počtu pulsů, tyto hodnoty byly získány rozdílem z příslušných hodnot z výsledků druhé fáze kalibrace následujícím způsobem. Ve druhé fázi kalibrace byla výsledná hodnota pro překážku ve 20° na kalibrační desce 149 pulsů, pro kolík v 80° na kalibrační desce byla tato hodnota 421 pulsů, mezera mezi 20°-80° je tedy široká 272 pulsů. Třetí a čtvrtý sloupec obsahuje počet pulsů, po které sensor zaznamenal vzdálenost větší než 19 cm. V pátém a šestém sloupci se nachází poměr počtu pulsů detekované mezery a počtu pulsů šířky mezery, tedy poměr mezi hodnotami ve třetím a druhém sloupci. V sedmém a osmém sloupci je šířka detekované mezery vyjádřená v centimetrech. Tato hodnota byla získána následujícím způsobem. Kolíky jsou na kalibrační desce umístěny ve vzdálenosti 18 cm od středu otáčení serva. Nachází se tedy na obvodu kružnice o poloměru 18 cm. Všechny měřené mezery jsou široké 60° tedy jednu třetinu celkového obvodu kružnice. Vynásobíme-li třetinu obvodu kružnice poměrem ze sloupců číslo pět respektive číslo šest, dostaneme šířku detekované mezery v centimetrech.

Závěry z třetí fáze kalibrace

Tato poslední třetí fáze kalibrace byla klíčová pro druhý úkol mé bakalářské práce, kterým je navigace robota v bludišti. Výsledky této fáze jsou rozhodující pro tvorbu bludiště a nastavení jeho parametrů. Data ze třetí fáze totiž napovídají, jak velké musí být mezery pro průjezd robota, a jaké jsou výkyvy v měření sensoru během jeho otáčení na servu. Jelikož jsou výsledky této fáze kalibrace tak zásadní bylo toto měření provedeno vícekrát. Jak je zřejmé z tabulky 2.5 výsledky dvou na sobě nezávislých měření jsou značně odlišné a nelze mezi nimi najít žádnou spojitost. Přestože tato měření byla provedena ve stejných podmínkách, jsou jejich výsledky od sebe značně rozdílné a to místy i o desítky pulsů. Tento fakt byl rozhodujícím pro hledání příčiny výkyvů v naměřených hodnotách, které byly patrné již v některých předchozích měřeních.

Tabulka 2.5: výsledky třetí fáze kalibrace

1	2	3	4	5	6	7	8
rozsah měřené mezery [°]	velikost mezery [počet pulsů]	velikost detekované mezery v prvním měření [počet pulsů]	velikost detekované mezery ve druhém měření [počet pulsů]	poměr velikosti mezery a detekované mezery v prvním měření	poměr velikosti mezery a detekované mezery ve druhém měření	velikost detekované mezery v prvním měření	velikost detekované mezery ve druhém měření [cm]
0-60	338	22	50	0,065	0,148	1,227	2,788
20-80	272	42	56	0,154	0,206	2,911	3,881
40-100	289	58	63	0,201	0,218	3,783	4,109
60-120	267	81	52	0,304	0,195	5,729	3,678
80-140	275	91	76	0,331	0,276	6,237	5,209
100-160	237	78	50	0,329	0,211	6,204	3,977
120-180	255	50	53	0,196	0,208	3,696	3,918

Z výše uvedených důvodů byl robot doplněn o nový napájecí modul Li-ion power pack 2 cell (viz. Kapitola 1.4), který nahradil původní napájecí modul dodávaný v základním balení Boe-bota firmou Parallax, Inc. Vzhledem k výhodám tohoto modulu a vlastní změně napájení bylo třeba provést všechny fáze kalibrací popsanych v kapitolách 2.2.1 - 2.2.3 znovu.

2.2.4 Čtvrtá fáze kalibrace

Čtvrtá fáze kalibrace je zopakováním všech měření, které byly popsány v kapitolách 2.2.1 - 2.2.3. Jelikož bylo nutné všechny hodnoty naměřit znovu s vyměněným napájecím modulem, který byl před každým měřením plně nabit tak, aby byly zajištěny stejné podmínky pro každé měření.

Postup a kódy jednotlivých měření byl naprosto totožný s postupy popsány v kapitolách 2.2.1 - 2.2.3. Z tohoto důvodu nebudou v kapitole 2.2.4 znovu popisovány, ale budou představeny pouze výsledky těchto měření a komentáře k nim.

První fáze kalibrace s napájením z modulu Li-ion power pack 2 cell

Výsledky znovu naměřené první fáze kalibrace jsou shrnuty v tabulce 2.6

Tabulka 2.6: výsledky úvodního měření první fáze kalibrace s napájením pomocí Li-ion power pack 2 cell a jejich porovnání s původními hodnotami

Úhel [°]	Nakalibr. hodnota [počet pulsů]	Původní naka- libr. hodnota [počet pulsů]	Úhel [°]	Nakalibr. hodnota [počet pulsů]	Původní naka- libr. hodnota [počet pulsů]
20	150,5	127	100	503	510
40	239	231	120	586,5	588
60	335	332	140	679	686
80	405,5	411	160	763	762
90	446	452	180	856,5	859

Tabulka 2.6 shrnuje výsledky úvodního měření první fáze kalibrace s novým napájecím modulem. Z tabulky je patrné, že hodnoty z úvodního měření při napájení Li-ion power pack 2 cell jsou ve všech měřených úhlech rozdílné proti hodnotám z tabulky 2.1, avšak pouze nepatrně. Dále je patrné, že do 80° jsou nové hodnoty vyšší a od 90°, s výjimkou 160°, kde je rozdíl pouze jeden puls, nižší.

Tato fáze kalibrace je stejně jako v měření s původním napájecím modulem vstupním měřením a slouží pro získání základních hodnot středů kolíků v definovaných úhlech na základní desce pro druhé měření první fáze kalibrace.

Výsledky druhého měření první fáze kalibrace jsou shrnuty v tabulce 2.7. Stejně jako v kapitole 2.2.1 o první fázi kalibrace s původním napájecím modulem, jsou v tabulce ve druhém a třetím sloupci předpokládané hodnoty pro dané úhly bez kalibrace, beze změny rozsahu a s jeho změnou na základě prvního měření.

Tabulka 2.7: výsledky druhého měření první fáze kalibrace s napájením pomocí
Li-ion power pack 2 cell

1	2	3	4	5	6	7
úhel kolíku [°]	nenakalibrované hodnoty, bez změny rozsahu [počet pulsů]	nenakalibrované hodnoty, se změnou rozsahu [počet pulsů]	nakalibrované hodnoty [počet pulsů]	rozdíl nakalibrované hodnoty a nenakalibrované hodnoty [počet pulsů]	počet pulsů k další překážce [počet pulsů]	poměr [° úhlu/puls]
0	0	0	0		150,5	0,133
20	111,1	95,2	150,5	55,3	88,5	0,226
40	222,2	190,3	239	48,7	96,5	0,207
60	333,3	285,5	335	50,0	70	0,286
80	444,4	380,7	405,5	24,8	40,5	0,247
90	500,0	428,3	446	17,7	57	0,175
100	555,6	475,8	503	27,2	83,5	0,240
120	666,7	571,0	586,5	15,3	92,5	0,216
140	777,8	666,2	679	12,8	84	0,238
160	888,9	761,3	763	1,7	93,5	0,214
180	1000	856,5	856,5	0,0		

Závěry z první fáze kalibrace s napájecím modulem Li-ion power pack 2 cell

Podobně, jako tomu bylo po první fázi kalibrace s původním napájecím modulem, nelze jednoznačně identifikovat závislost mezi natočením serva a dodanými pulsy. Porovnáním tabulky 2.2 s výsledky tohoto měření s původním napájecím modulem a tabulky 2.7 lze sledovat podobný charakter vývoje poměru vyjádřeného v posledním sloupci. Tento ukazatel vyjadřuje citlivost serva na dodaný puls v rozmezí mezi dvěma kolíky. Na základě tohoto měření však nelze jednoznačně určit, že by změna měla jasný vliv na citlivost serva na puls, jelikož hodnoty nejsou jednoznačně menší nebo větší než hodnoty v tabulce 2.2. Srovnáním výsledků v předposledním sloupci, který vyjadřuje počet pulsů k sousednímu kolíku ve směru otáčení serva, lze sledovat vyšší hodnoty než s původním napájecím modulem, avšak jen do mezery mezi

kolíky ve 100° a 120° . Rozdíly mezi těmito hodnotami v obou tabulkách jsou řádově v jednotkách pulsů, s rozdílem prvních dvou mezer, kde rozdíl dosahuje více než dvaceti pulsů.

Závěr z této fáze kalibrace je tedy obdobný jako v kapitole 2.1.1. Byly identifikované obdobné nesrovnalosti, jako s původním modulem a je nutná další kalibrace pro získání přesnějších hodnot. Ani srovnáním tabulek s výsledky stejných měření s různými napájecími moduly nelze nalézt jednoznačnou závislost, která by napomohla vysvětlit nerovnoměrnost výsledků měření v první fázi kalibrace.

Druhá fáze kalibrace s napájením z modulu Li-ion power pack 2 cell

Výsledky druhé fáze kalibrace s napájecím modulem Li-ion power pack 2 cell jsou shrnuty v tabulce 2.8.

Tabulka 2.8: výsledky měření druhé fáze kalibrace s napájením
Li-ion power pack 2 cell

počet pulsů ve směru serva	pozice kolíku [$^\circ$]	počet pulsů proti směru rozsahu serva
135	20	133
145	40	124
143	60	107
133	80	76
120	90	56
147	100	114
136	120	101
136	140	102
121	160	111
116	180	107

Závěry ze druhé fáze kalibrace s napájecím modulem Li-ion power pack 2 cell

Stejně jako v druhé fázi kalibrace s původním napájecím modulem jsou naměřené hodnoty po směru pohybu serva vyšší než hodnoty naměřené pohybem proti obvyklému směru serva. Pro přepočítání a korekci nakalibrovaných hodnot z první fáze kalibrace tedy mohl být použit stejný vzorec, jako v kapitole 2.2.2. Srovnáním tabulky 2.8s novými výsledky s tabulkou 2.3, ve které jsou výsledky původního měření, lze sledovat vyšší hodnoty s novým napájecím modulem až do kolíku v úhlu 120° , kdy se hodnoty k sobě téměř na jednotky pulsů přiblíží.

Touto fází kalibrace byla ukončena měření, pomocí jejichž výsledků jsem hledal závislost mezi dodaným pulsem servu a jeho natočením. Tímto měřením byly nalezeny absolutní středy pásem, během kterých sensor detekuje kolík v definovaném úhlu. Přestože na základě výsledků znovu naměřených prvních dvou fází kalibrace, nelze jednoznačně určit, který z napájecích modulů je vhodnější, bude pro všechna další měření a testy použit modul Li-ion power pack 2 cell, který má možnost dobíjení před každým novým měřením a je tedy jistějším napájecím zdrojem.

Tabulka 2.9: výsledky druhé fáze kalibrace s napájením Li-ion power pack 2 cell

úhel kolíku [°]	20	40	60	80	90	100	120	140	160	180
naklibrovaná hodnota [počet pulsů]	151,5	249	353,5	434	478	519,5	604	696	768	861

Třetí fáze kalibrace s napájením z modulu Li-ion power pack 2 cell

Výsledky třetí fáze kalibrace s napájecím modulem Li-ion power pack 2 cell jsou shrnuty v tabulce 2.10.

Tabulka 2.10: výsledky třetí fáze kalibrace s napájecím modulem Li-ion power pack 2 cell

1	2	3	4	5	6	7	8
rozsah měřené mezery [°]	velikost mezery [počet pulsů]	velikost detekované mezery v prvním měření [počet pulsů]	velikost detekované mezery ve druhém měření [počet pulsů]	poměr velikosti mezery a detekované mezery v prvním měření	poměr velikosti mezery a detekované mezery ve druhém měření	velikost detekované mezery v prvním měření	velikost detekované mezery ve druhém měření [cm]
0-60	335,5	60	73	0,179	0,218	3,371	4,101
20-80	255	74	90	0,290	0,353	5,470	6,653
40-100	264	106	108	0,402	0,409	7,568	7,711
60-120	251	92	103	0,367	0,410	6,909	7,735
80-140	273,5	105	106	0,384	0,388	7,237	7,305
100-160	260	63	76	0,242	0,292	4,567	5,510
120-180	270	49	56	0,181	0,207	3,421	3,910

Závěry ze třetí fáze kalibrace s napájecím modulem Li-ion power pack 2 cell

Tato poslední fáze kalibrace, jak již bylo Během třetí fáze kalibrace s napájecím modulem Li-ion power pack byly stejně jako s původním měřením provedeny dvě měření

za stejných podmínek a se shodným kódem. V tabulce č.14 jsou shrnuty výsledky tohoto měření. Srovnáním rozdílů z tabulky 2.5 s původním měřením z tabulky 2.10 s novým měřením je zřejmé že s novým napájecím modulem bylo dosaženo menších rozdílů. Zatímco v původním měření dosahoval rozdíl mezi naměřenými hodnotami až 29 pulsů v jednom a 28 pulsů ve dvou případech, v opakovaném měření v rámci čtvrté fáze kalibrace byl nejvyšší rozdíl 16 pulsů. Lze tedy předpokládat, že použitím Li-ion power pack 2 cell bylo dosaženo menších výkyvů hodnot v rámci stejných měření.

Tato fáze kalibrace, jak již bylo zmíněno v kapitole 2.2.3, je rozhodující pro stanovení parametrů bludiště pro druhý úkol této práce. Díky tomuto měření je nyní jasnější, jakým způsobem je robot pomocí sensoru schopen zaznamenat mezeru mezi souvislými překážkami, což bude velmi platné při tvorbě navigačního algoritmu pro bludiště.

Kapitola 3

Navigace robota v bludišti

Kapitola 2 se zabývala kalibrací ultrazvukového sensoru a hledáním závislosti počtu dodaných pulsů servu se senzorem na jeho natočení. V této kapitole budou tyto poznatky využity v praxi a to především při definování a tvorbě bludiště a následném vývoji navigačního algoritmu pro robota.

3.1 Definice bludiště

Prvním krokem pro navigaci robota v bludišti je definování bludiště, aby bylo možné zajistit pohyb robota v bludišti, je nutné, aby byly navrženy odpovídající rozměry částí bludiště vůči rozměrům robota. Rozměry robota jsou shrnuty v tabulce 3.1.

Tabulka 3.1: rozměry robota Boe-bot s PING))) senzorem

Rozměr	Hodnota [cm]
Délka	20
Výška	12
Šířka	11,5

Bludiště, které měl robot projet, bylo navrženo jako labyrint se stěnami vzájemně svírajícími pravé úhly. Ve stěnách byly výstupky do prostoru labyrintu, s pravoúhlým přechodem od stěny, a mezery různých velikostí průjezdné i slepé.

3.2 Návrh navigačního algoritmu pro pohyb v bludišti

Pro průjezd robota bludištěm splňujícího podmínky z kapitoly 3.1 byla zvolena metoda následování stěny (pro sepsání algoritmu do kódu byla zvolena levá stěna ve směru jízdy robota). Vzhledem k faktu, že během všech fází kalibrace byla zvolena vzdálenost robota od překážek na 18 cm, byla stanovena vzdálenost robota od stěny při jejím následování také na 18 cm. Celý navigační algoritmus je shrnut do vývojového diagramu na obrázku A.4. kód je na obrázku 3.1.

```
' {$STAMP BS2}
' {$PBASIC 2.5}

Cmconstant      CON 2260
time            VAR Word
distance        VAR Word
distanceL       VAR Word
distanceR       VAR Word
distancel       VAR Word
servoPin        CON 15
leftwheel       CON 13
rightwheel      CON 12
limit           CON 15
PINGpin         CON 14
piezoPin        CON 4
counter         VAR Word
temp            VAR Word
mezera          VAR Word

FOR temp = 660 TO 678
    PULSOUT servoPin,temp
    PAUSE 20
NEXT

Main:
DO
    GOSUB dopredu
    counter=counter+1
    IF (counter//5=0) THEN
        PAUSE 50
        GOSUB mereni
        IF (distance <=limit) THEN
            PAUSE 50
            GOSUB vyrovnani
            GOSUB podel_steny
        ENDIF
    ENDIF
LOOP
```

Obrázek 3.1: základní kód pro pohyb robota

Na obrázku A.4 je shrnut celý navigační algoritmus do jednoho vývojového di-

agramu. Jednotlivé kroky v tomto diagramu však představují více činností (tzv. subrutin), kód pro základní subrutiny je na obrázku 3.2. Tyto kroky jsou v kódu pro navigační program sepsány do samostatných částí viz. Kapitola 1.1 bod s příkazem **GOSUB**. V následujících podkapitolách tedy budou představeny tak, jak jsou označeny v kódu.

```
mereni:
    PULSOUT PINGpin , 5
    PULSIN PINGpin , 1, time
    distance = Cmconstant **
RETURN

otocka_90_right:
FOR counter = 1 TO 10
    PULSOUT 13, 1150
    PULSOUT 12, 1150
    PAUSE 50
NEXT
RETURN

otocka_90_left:
FOR counter = 1 TO 18
    PULSOUT 13, 350
    PULSOUT 12, 350
    PAUSE 50
NEXT
RETURN

dopredu:
PULSOUT leftwheel, 850
PULSOUT rightwheel, 650
RETURN

dopredu_pomalou:
PULSOUT leftwheel, 800
PULSOUT rightwheel, 700
RETURN
```

Obrázek 3.2: kód pro základní subrutiny

3.2.1 Algoritmus pro předem neznámou cestu

Celý algoritmus pro předem neznámou cestu v pravoúhlém bludišti se skládá z několika částí, které budou představeny v následujících podkapitolách.

Algoritmus pro příjezd robota ke stěně a vyrovnání se stěnou

Tento algoritmus je úvodní částí kódu a zajišťuje bezpečný příjezd robota z libovolného místa uvnitř bludiště k jeho stěně a následné vyrovnání do rovnoběžného směru se stěnou aby ji mohl následovat. Vývojový diagram této části je zobrazen na obrázku A.5. Kód pro vyrovnání robota se stěnou je na obrázku A.6.

Na počátku tohoto algoritmu je robot postaven do libovolného místa labyrintu a pohybuje se směrem vpřed, v každém cyklu se do kontrolní proměnné counter přičte hodnota jedna, je-li zbytek po dělení proměnné counter pěti roven nule. Provádí se kontrolní měření vzdálenosti od překážky přímo před robotem. Dokud není dosaženo limitu (v kódu nastaven na 15 cm od překážky) je tento cyklus opakován. Při dosažení limitu dochází ke kontrolnímu měření do 40° a do 140° (viz obrázek 2.2), tyto dvě hodnoty jsou porovnány mezi sebou (hodnota vzdálenosti v úhlu 40° je uložena do proměnné **distanceP**, hodnota vzdálenosti v úhlu 140° je uložena do proměnné **distanceL**). Je-li proměnná **distanceP** větší, znamená to, že robot je natočen svým levým bokem ke stěně a robot je otočen o 45° směrem doprava, v opačném případě je robot natočen svým pravým bokem ke stěně a je otočen o 180° doprava. Následuje proces vyrovnávání se stěnou, kdy je servo s PING))) senzorem natočeno zcela do své levé krajní pozice a provádí kontrolní měření. Je-li naměřená vzdálenost menší než 18 cm, následuje krátký pohyb dopředu a kód je posunut opět na část s kontrolním měřením, je-li vzdálenost větší, dojde k natáčení celého robota po krátkých pulsech směrem doleva (ke stěně bludiště). Následuje další měření a podmínka, ve které je vzdálenost robota opět porovnávána s hodnotou 18 cm. Při hodnotě vzdálenosti větší než 18 cm je proces vrácen k natáčení robota ke stěně a měření je provedeno znovu, pro hodnoty menší nebo rovné 18 cm je tato část kódu ukončena a robot je připraven následovat stěnu.

Algoritmus pro následování stěny

Vzhledem ke konci algoritmu popsaného v kapitole 3.2.1 je zřejmé, že na konci tohoto algoritmu bude robot v přibližně vodorovné pozici se stěnou, kterou bude mít po svém levém boku. Algoritmus pro následování trasy má za úkol pohybovat s robotem tak, aby svou jízdou kopíroval stěnu. Vzhledem k faktu, že pohybová serva robota nelze nastavit tak, aby vedly robota naprosto přesně rovným směrem, bylo zavedeno určité toleranční pásmo vzdálenosti robota od stěny na 15-21 cm, ve kterém se může robot pohybovat bez korekce směru pohybu. Vyjede-li z tohoto pásma, jeho směr je korigován zpět do tohoto tolerančního pásma). Vývojový diagram algoritmu pohybu podél stěny je na obrázku A.7, kód pro jízdu podél stěny je na obrázku A.8, Kód pro rozhodnutí při příjezdu ke kolmé překážce je znázorněn na obrázku 3.3.

```

stena_limit:
  FOR temp = 1045 TC 1061
    PULSOUT servoPin, temp
    PAUSE 20
  NEXT
  GOSUB mereni
  distance = distanceL

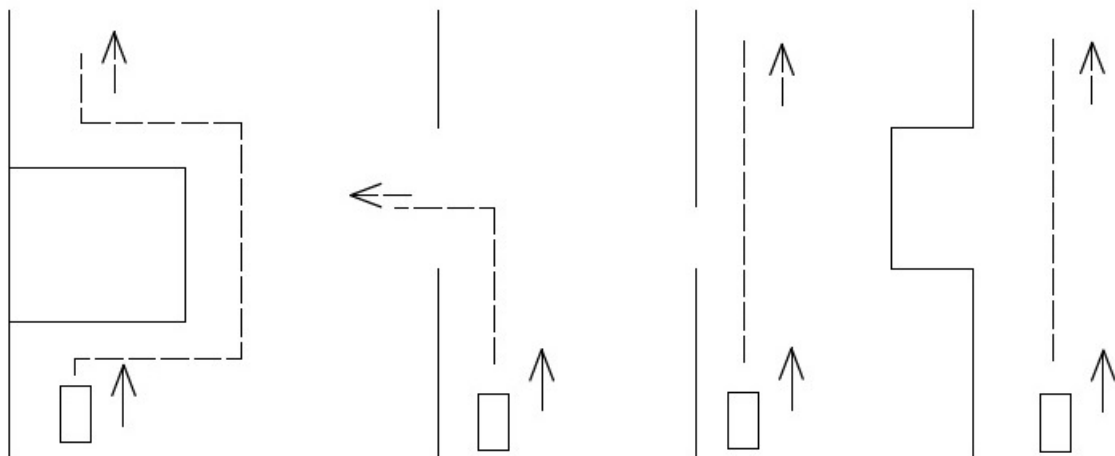
  FOR temp = 210 TC 200
    PULSOUT servoPin, temp
    PAUSE 20
  NEXT
  distance = distanceR

  IF DistanceR>=distanceL THEN
    GOSUB otocka_90_right
    PAUSE 100
    GOTC podel_steny
  ELSEIF distanceL>distanceR THEN
    GOSUB otocka_90_left
    PAUSE 100
    FOR counter = 1 TC 100
      GOSUB dopredu
    NEXT
    GOTC podel_steny
  ENDIF
RETURN

```

Obrázek 3.3: kód pro rozhodnutí při příjezdu ke kolmé překážce

V tomto algoritmu jsou zachyceny identifikace překážek a proces jejich objetí. Překážky, které mohou nastat v bludišti, jsou znázorněny na obrázku 3.4.



Obrázek 3.4: typy překážek v bludišti

Prvním typem překážky je výstupek ze stěny do směru koridoru labyrintu. V tomto případě robot zaznamená jako první kolmou překážku před sebou. V tomto okamžiku se zastaví a provede měření zcela vpravo od sebe a zcela vlevo od sebe. Tyto hodnoty porovná a provede otočku o 90° směrem, kde je větší vzdálenost (tento algoritmus je ve schématu na obrázku č. 20 označen jako `stena_limit`). V případě znázorněném na obrázku č. 21 se robot natočí doprava a bude dále následovat stěnu po svém levém boku. Na hranici překážky měřením vyhodnotí, že po svém levém boku mu vznikla mezera větší než 25 cm v tomto okamžiku se začne pohybovat dopředu pomaleji a dále měří vzdálenost od stěny. Počet pulsů (během kterých se robot posunul dopředu a zaznamenal vzdálenost větší než 35 cm) se postupně zvětšuje, je-li jejich počet roven osmi, robot provede otočku o 90° směrem vlevo a popojede dopředu. Následuje opět návrat k algoritmu pro následování stěny. Stejný proces se opakuje pro průjezd podél celé překážky.

Druhým typem překážky je mezera ve stěně dostatečně velká pro průjezd robota. V tomto případě je postup robota naprosto shodný s úvodem prvního případu až na úvodní cyklus `stena_limit`. Po otočení robota doprava a popojetí dopředu je příkazem **GOTO** kód vrácen na začátek algoritmu pro následování stěny.

Třetí typ překážky je mezera ve stěně, ale na rozdíl od druhého případu není dostatečně široká pro průjezd robota. V tomto případě proměnná mezera nedosáhne hodnoty 8 a kód se vrátí na počátek algoritmu pro následování stěny.

Čtvrtý typ překážky je slepé odbočení, v tomto případě robot nezaznamená dostatečnou vzdálenost koridoru ve stěně a pokračuje v následování stěny.

Všechny tyto typy překážek mohou být rozmístěny v bludišti zcela náhodně. Signálem pro robota, že úspěšně vyjel z bludiště, je hodnota vzdálenosti před sebou větší než 1 m a zároveň alespoň jedna ze vzdáleností napravo nebo nalevo od robota větší než 1 m.

3.2.2 Algoritmus pro navigaci robota pro předem známou cestu

Kapitoly 3.2.1 a 3.2.2 se zabývaly problematikou navigace robota po neznámé cestě v bludišti s pravouhlými stěnami. Tato kapitola shrne postup pro navigaci robota pro cestu, jejíž charakteristiky jsou dopředu známy, cílem je následování vytyčené cesty. Jelikož robot (v setu, který mi byl k dispozici pro plnění této práce) není vhodný pro pohyb ve venkovním prostředí, byla pro tento úkol vybrána cesta mezi třemi pokoji v bytě. Schéma trasy robota je znázorněné na obrázku 3.5 čárkovanou čarou.

3.2.3 Srovnání algoritmů jednotlivých úloh

Algoritmus popsáný v kapitole 3.2 je „tvrdým“ řešením úlohy, která byla zadána. Vzhledem k výbavě robota nebylo bohužel možné použít sofistikovanější algoritmy, které by braly v úvahu již projatá místa v bludišti a případně je zaznamenávala do paměti. V každé situaci kdy probíhá měření je tak řešeno zároveň řešeno několik podmínek najednou a to celkově zpomaluje plynulý pohyb robota v labyrintu. Další nevýhodou pro hladké provedení algoritmu je princip, na kterém pracuje samotný ultrazvukový sensor (popsáno v kapitole 1.1.3 o PING)) sensoru), a problematika měření překážek (popsaná v kapitole 2.2.1). Tyto dva fakty současně ovlivňovaly testování algoritmu a lazení (především hledání odpovídající rychlosti robota v bludišti) kódu nejvíce. Výsledkem první úlohy bylo tedy řešení, které bylo následně využito a aplikováno na konkrétní situaci, kterou řešila druhá úloha se známou cestou.

Závěr

Cílem této práce bylo seznámení se s ultrazvukovým (PING))) senzorem a robotem Boe-bot., vytvoření kalibračního přípravku a algoritmu kalibrace senzoru a následné navržení algoritmu pro pohyb robota v bludišti pomocí dálkoměrného ultrazvukového senzoru.

Celou práci provázal problém s nepřesností senzoru a složité identifikace hran detekovaných objektů. Tento problém je způsoben principem měření vzdálenosti pomocí měření odezvy na ultrazvukovou vlnu. Vznikají tak situace, kdy robot již přímo před senzorem překážku nemá, ale vrací vzdálenost překážky, která se může nacházet až 20° od přímého směru před senzorem na obě dvě strany zorného pole. Tyto situace jsou velmi problematické pro samotný navigační algoritmus, který je dělen vyhodnocováním podmínek a porovnáváním naměřených vzdáleností. Tento fakt byl objasněn třemi fázemi kalibrace senzoru, kdy každá fáze sledovala funkci senzoru za poněkud odlišných podmínek, ale po všech třech fázích si již bylo možné vytvořit představu, jaké vzdálenosti bude robot v jednotlivých situacích v bludišti měřit a vracet. V rámci kalibrace je nutné dodat, že by bylo vhodné doplnit poznatky z této práce o výsledky podobného senzoru vzdálenosti, který by měřil vzdálenost bodově. Avšak bylo by nutné vytvořit nový kalibrační přípravek, případně využít sensor s podobnými rozměry, jako (PING))) sensor aby byla zajištěna shodná pozice měřícího bodu v kalibračním přípravku.

Závěrečná kapitola práce se zabývala problematikou navigace robota v bludišti s pravoúhlými stěnami a to po předem známé a neznámé cestě. Použitím více popsanych algoritmů byl robot schopen objet všechny překážky a dorazit na místo předem určené jako cíl cesty. I v této části práce bylo však nutné se potýkat s problémy, kdy robot nebyl schopen zcela jasně identifikovat hranu překážky a naměřené vzdálenost, což je jediná veličina, dle které se robot s (PING))) senzorem řídí, na hranách překážek nabývala na hodnotě postupně a nikoliv skokově, jak by to bylo pro efektivnost algoritmu ideální.

Jako velmi slibné rozšíření této práce vidím možnost využití GPS modulu, pomocí kterého by byl robot schopný zaznamenávat a ukládat svou polohu a tímto způsobem

mapovat bludiště jako celek. Využití modulu pro bezdrátovou komunikaci s počítačem pomocí rozhraní Bluetooth, by jistě zjednodušilo práci s robotem a analýzu naměřených výsledků. Nápomocným by mohl být i další dálkoměrný sensor, který by měřil vzdálenost druhé stěny a robota zároveň s prvním, otázkou však je, zda je využití dvou sensorů najednou možné vzhledem k omezené kapacitě paměti robota.

Seznam použitých zdrojů

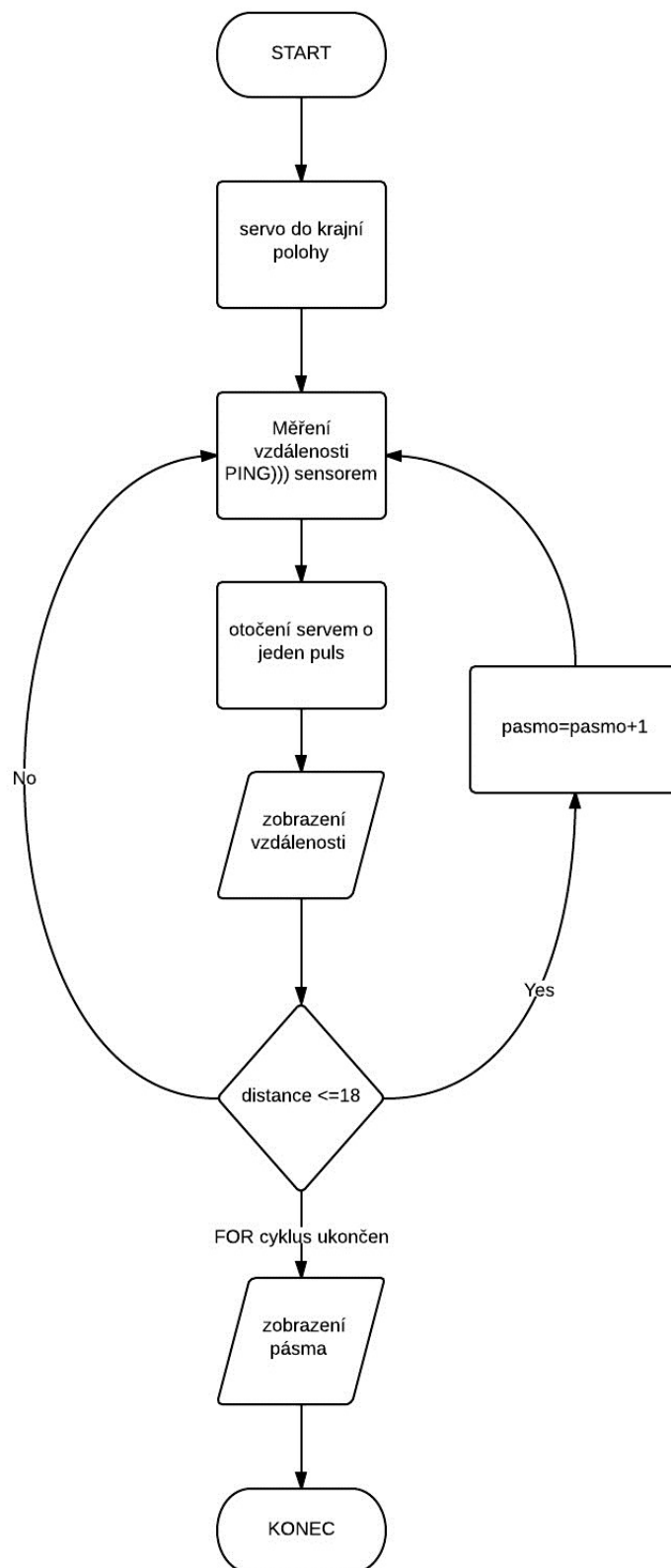
- [1] MARTIN, Jeff, a kol. *BASIC Stamp Syntax and Reference Manual 2.2* [USA]: Parallax, Inc. [2005] [cit. 2014-06-28]. ISBN 1-928982-32-8. Dostupné z: <<http://www.parallax.com/sites/default/files/downloads/27218-Web-BASICStampManual-v2.2.pdf>>.
- [2] LINDSAY, Andy. *Robotics with Boe-bot: Student Guide 3.0* [USA]: Parallax, Inc., [2005] [cit. 2014-06-28]. ISBN 9781928982531. Dostupné z: <<http://www.parallax.com/sites/default/files/downloads/28125-Robotics-With-The-Boe-Bot-v3.0.pdf>>.
- [3] PARALLAX, Inc. *Li-ion power pack 2 cell v1.0 (28986)* [USA]: Parallax, Inc. [2011] [cit. 2014-07-17]. Dostupné z: <<http://www.parallax.com/sites/default/files/downloads/28986-Li-ion-Power-Charger2cell-v1.0.pdf>>.
- [4] PARALLAX, Inc. *PING))) ultrasonic distance sensor (28015)*[USA]: Parallax, Inc. [2013] [cit. 2014-07-02]. Dostupné z: <<http://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf>>.
- [5] PARALLAX, Inc. *Parallax Standard Servo (900-00005) v2.2*[USA]: Parallax, Inc. [2011] [cit. 2014-07-05]. Dostupné z: <<http://www.parallax.com/sites/default/files/downloads/900-00005-Standard-Servo-Product-Documentation-v2.2.pdf>>.

Přílohy

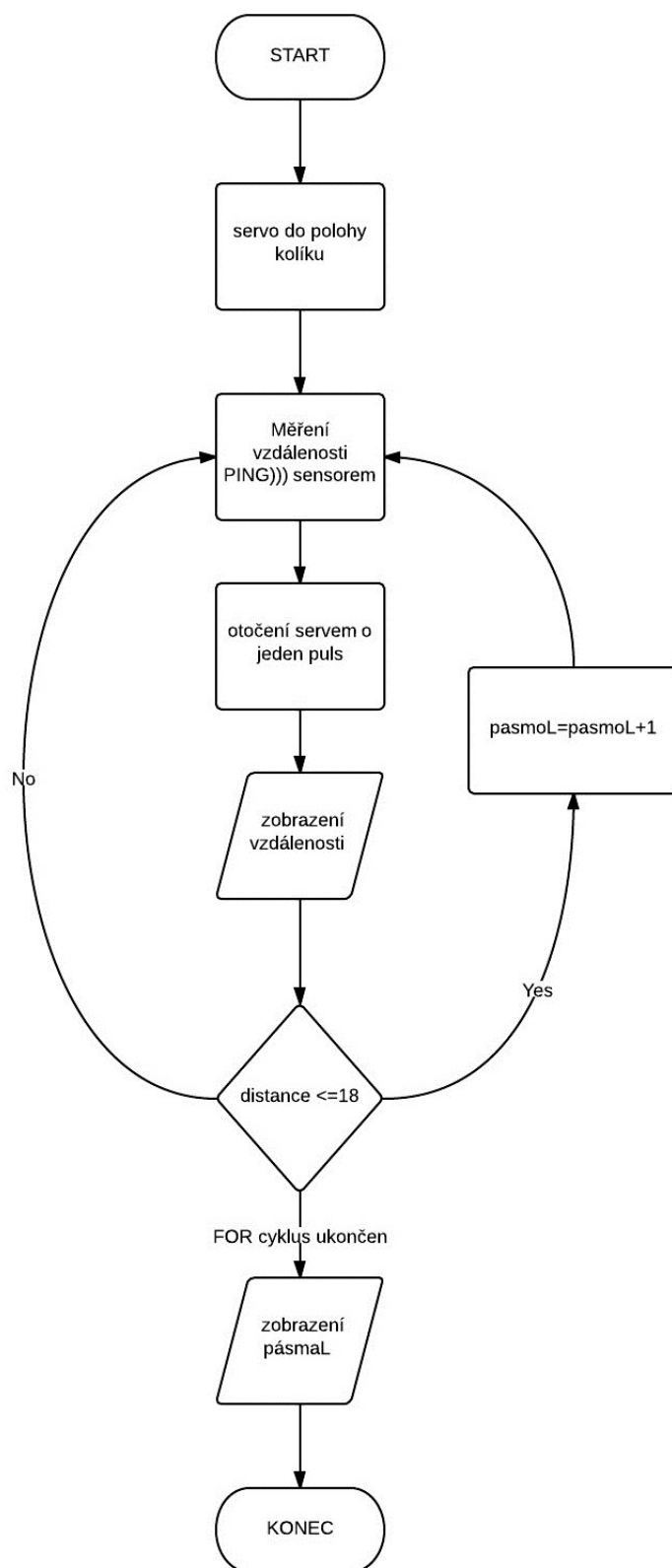
Příloha A

Vývojové diagramy a složitější kódy

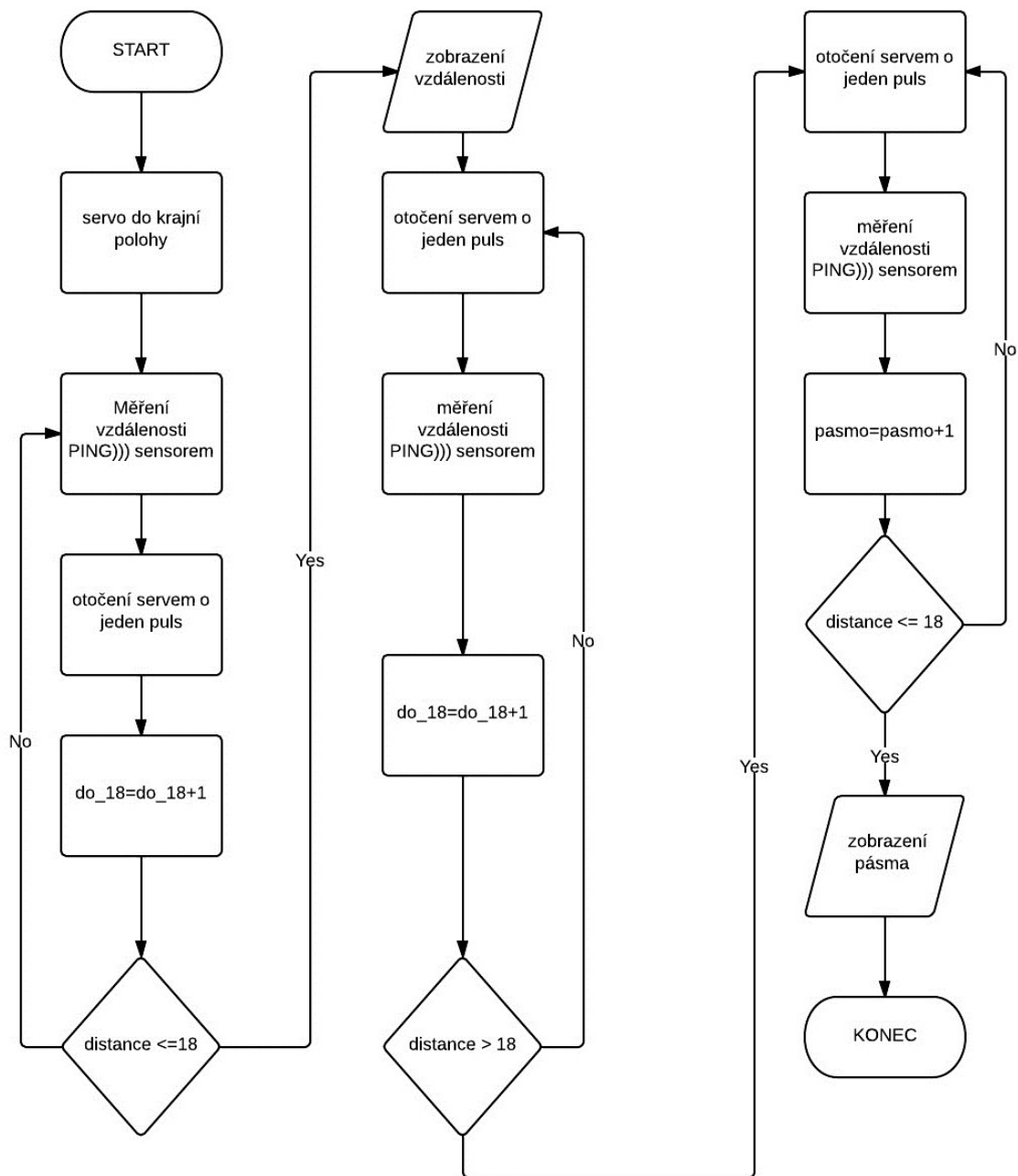
V této části budou postupně uvedeny obrázky, na které nebyl prostor v práci samotné. Jedná se o vývojové algoritmy jednotlivých fází kalibrace a části pohybu robota v bludišti a složitější kódy pro pohyb robota v bludišti.



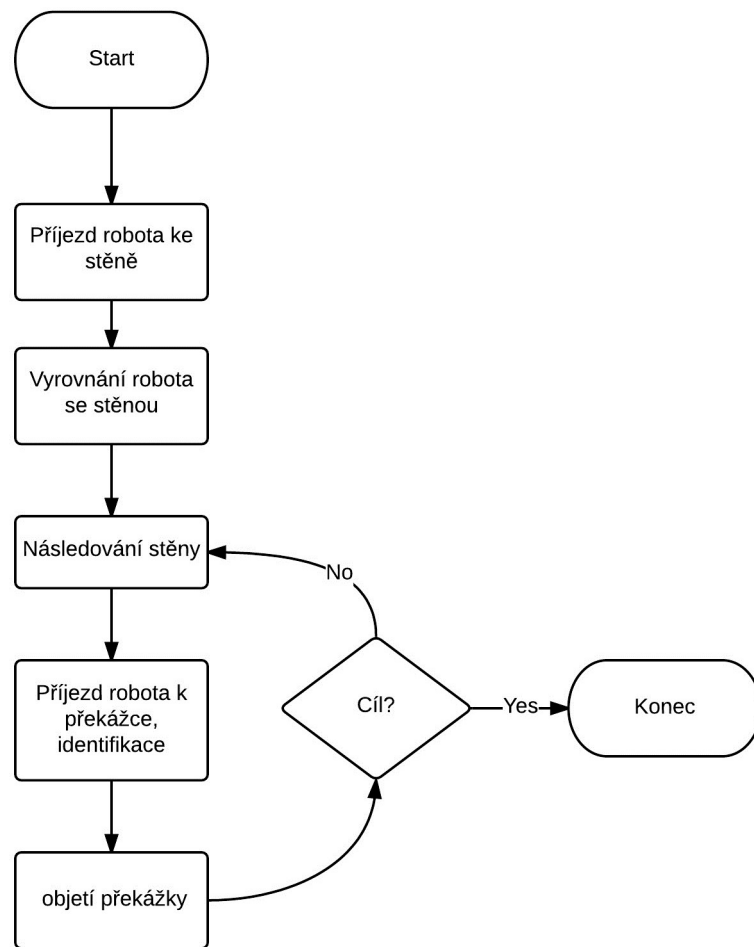
Obrázek A.1: vývojový diagram prvního kalibračního algoritmu



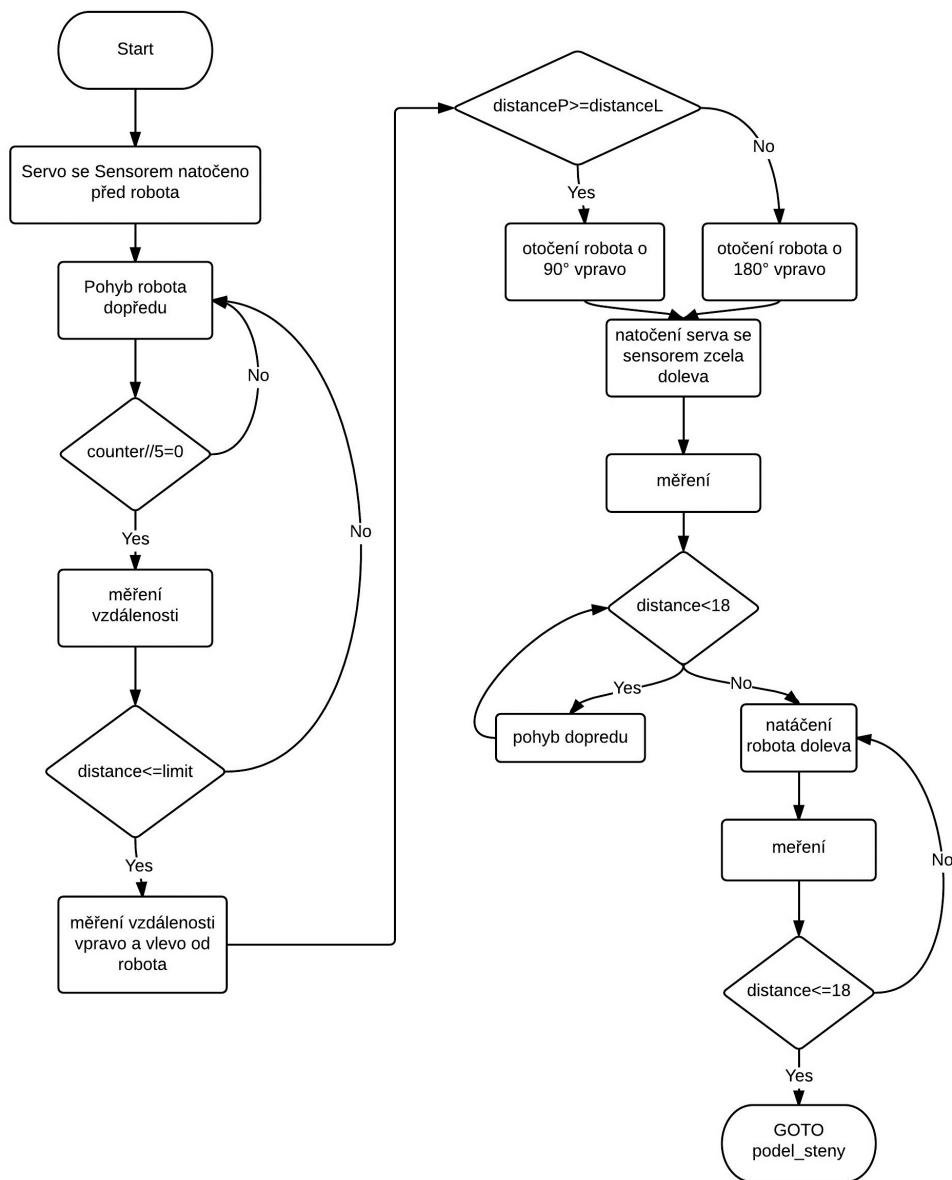
Obrázek A.2: vývojový diagram druhého kalibračního algoritmu



Obrázek A.3: vývojový diagram třetího kalibračního algoritmu



Obrázek A.4: vývojový diagram navigačního algoritmu



Obrázek A.5: vývojový diagram algoritmu pro příjezd robota ke stěně a vyrovnání se stěnou

```

vyrovnani:
FREQOUT piezoPin, 200, 3800
FOR temp = 432 TC 450
  PULSOUT servoPin,temp
  PAUSE 20
NEXT

mereni_stenaR:
GOSUB mereni
IF distance =0 THEN
  GOTC mereni_stenaR
ELSE
  distanceR=distance
  DEBUG? distanceR
ENDIF

FOR temp = 870 TC 896
  PULSOUT servoPin,temp
  PAUSE 20
NEXT

mereni_stenaL:
GOSUB mereni
IF distance = 0 THEN
  GOTC mereni_stenaL
ELSE
  distanceL=distance
  DEBUG? distanceL
ENDIF

IF (distanceR>=distanceL) THEN
  DEBUG "pravo vic"
  FREQOUT piezoPin, 200, 3800
  FREQOUT piezoPin, 200, 3800
  counter=0
  GOSUB ostocka_90_right
  FOR temp = 1050 TC 1061
    PULSOUT servopin, temp
    PAUSE 20
  NEXT
  DO
    GOSUB mereni
    DEBUG ?distance
    IF distance<18 THEN
      GOSUB mereni
      DEBUG ?distance
      IF distance<18 THEN
        FOR counter = 1 TC 2
          FREQOUT piezoPin, 200, 3800
          FREQOUT piezoPin, 500, 3800
          PULSOUT leftwheel, 800
          PULSOUT rightwheel, 700
        NEXT
      ELSE
        DO
          FREQOUT piezoPin, 500, 3800
          FREQOUT piezoPin, 200, 3800
          PULSOUT leftwheel, 730
          PULSOUT rightwheel,730
          GOSUB mereni
          PAUSE 20
          IF distance <= 18 THEN
            RETURN
          ENDIF
        LOOP
      ENDIF
    LOOP
  ENDIF
  RETURN
ELSE
  DO
    FREQOUT piezoPin, 500, 3800
    FREQOUT piezoPin, 200, 3800
    PULSOUT leftwheel, 730
    PULSOUT rightwheel,730
    GOSUB mereni
    DEBUG ?distance
    PAUSE 20
    IF distance <= 18 THEN
      RETURN
    ENDIF
  LOOP
ENDIF
LOOP

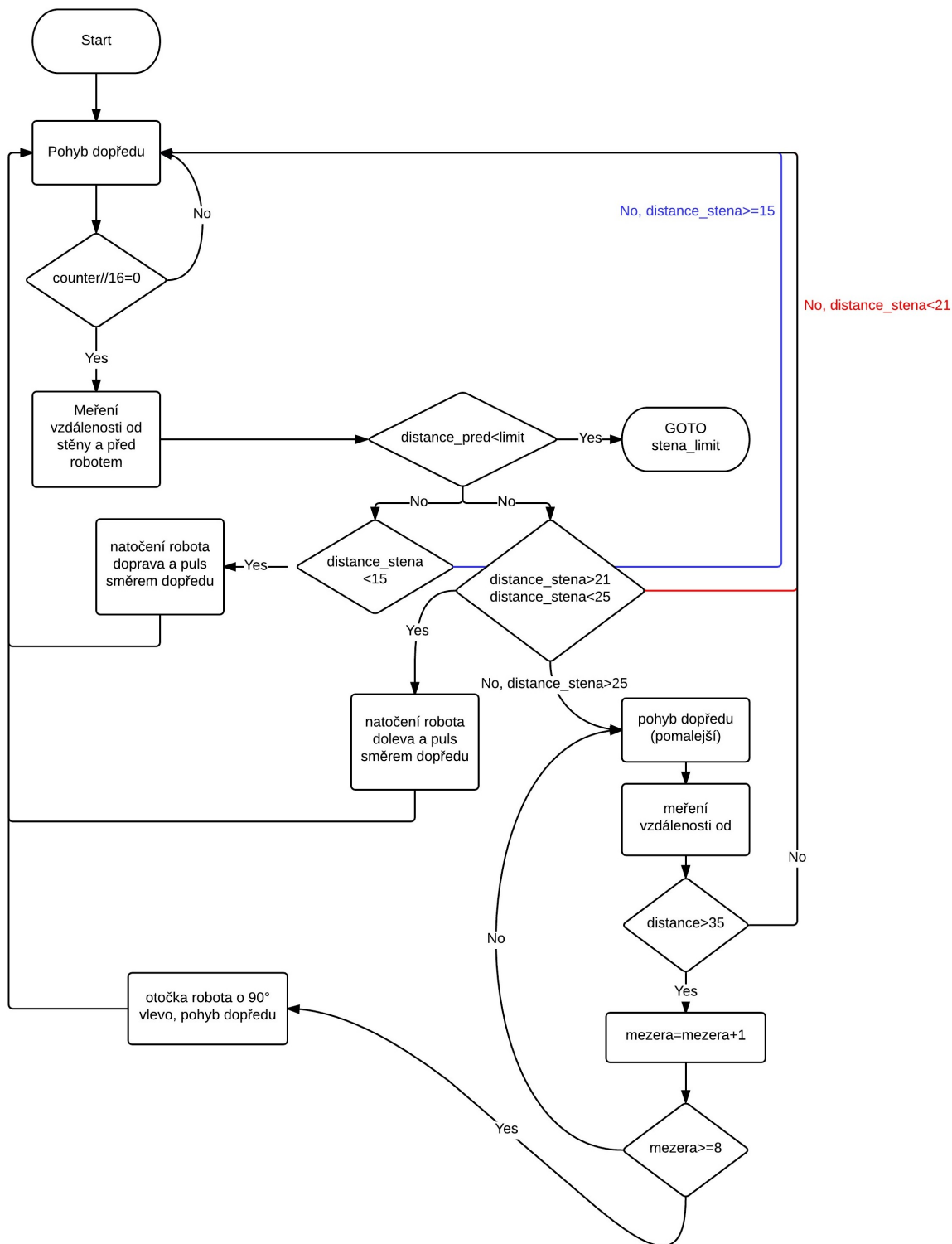
```

```

ELSEIF (distanceL>distanceR) THEN
  DEBUG "levo vic"
  FREQOUT piezoPin, 500, 3800
  FREQOUT piezoPin, 500, 3800
  counter=0
  GOSUB ostocka_90_right
  GOSUB ostocka_90_right
  FOR temp = 1050 TC 1061
    PULSOUT servopin, temp
    PAUSE 20
  NEXT
  DO
    GOSUB mereni
    IF distance<18 THEN
      FOR counter = 1 TC 2
        FREQOUT piezoPin, 200, 3800
        FREQOUT piezoPin, 500, 3800
        FREQOUT piezoPin, 500, 3800
        FREQOUT piezoPin, 500, 3800
        counter=0
        GOSUB ostocka_90_right
        GOSUB ostocka_90_right
        FOR temp = 1050 TC 1061
          PULSOUT servopin, temp
          PAUSE 20
        NEXT
      NEXT
    ELSE
      DO
        GOSUB mereni
        IF distance<18 THEN
          FOR counter = 1 TC 2
            FREQOUT piezoPin, 200, 3800
            FREQOUT piezoPin, 500, 3800
            PULSOUT leftwheel, 800
            PULSOUT rightwheel, 700
          NEXT
        NEXT
      ELSE
        DO
          FREQOUT piezoPin, 500, 3800
          FREQOUT piezoPin, 200, 3800
          PULSOUT leftwheel, 730
          PULSOUT rightwheel,730
          GOSUB mereni
          PAUSE 20
          IF distance <= 18 THEN
            RETURN
          ENDIF
        LOOP
      ENDIF
    LOOP
  ENDIF
  RETURN

```

Obrázek A.6: kód pro vyrovnání se stěnou



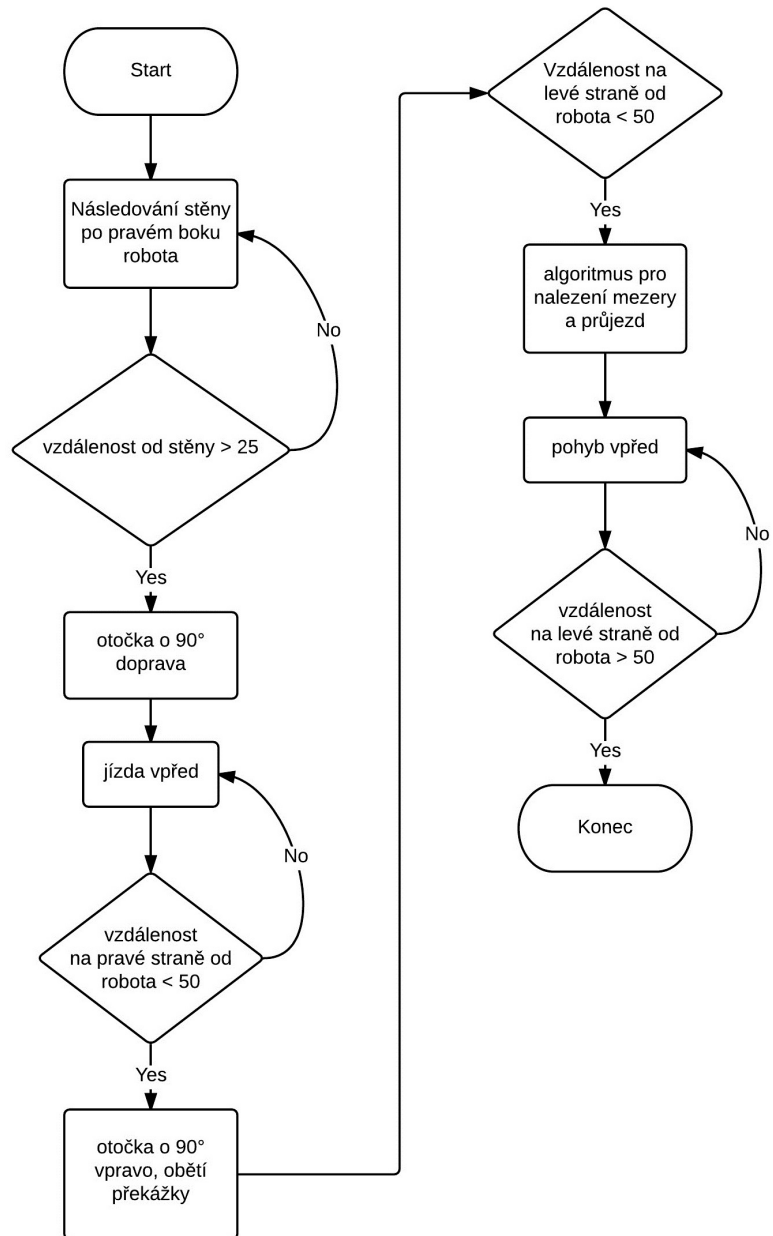
Obrázek A.7: vývojový diagram pohybu podél stěny

```

podel_steny:
mezera = 0
FOR temp = 1045 TO 1061
    PULSOUT servoPin, temp
    PAUSE 20
NEXT
DO
    GOSUB dopredu
    counter = counter +1
    IF (counter//16=0) THEN
        FOR temp = 1045 TO 1061
            PULSOUT servoPin, temp
            PAUSE 20
        NEXT
        GOSUB mereni
        distance1=distance
        DEBUG ?distance1
        IF ( (distance1 >=15) AND (distance1 <=21) ) THEN
            GOSUB dopredu
        ELSEIF ((distance1>=1) AND (distance1<15) ) THEN
            DO
                PULSOUT leftwheel, 770
                PULSOUT rightwheel, 770
                PAUSE 20
                PULSOUT leftwheel, 800
                PULSOUT rightwheel, 700
                PAUSE 20
                GOSUB mereni
                DEBUG ?distance
                IF (distance>= 16) THEN
                    GOTC podel_steny
                ENDIF
            LOOP
        ELSEIF ( (distance1>21) AND (distance1<25) ) THEN
            DO
                PULSOUT leftwheel, 730
                PULSOUT rightwheel, 730
                PAUSE 20
                PULSOUT leftwheel, 800
                PULSOUT rightwheel, 700
                PAUSE 20
                GOSUB mereni
                DEBUG ?distance
                IF (distance<=20) THEN
                    GOTC podel_steny
                ENDIF
            LOOP
        ELSEIF (distance1>25) THEN
            DO
                FOR counter = 1 TO 5
                    GOSUB dopredu_pomalu
                NEXT
                GOSUB mereni
                DEBUG ?distance
                IF (distance > 35) THEN
                    mezera=mezera+1
                ENDIF
                IF mezera=8 THEN
                    DEBUG ?mezera
                    GOSUB ostocka_90_left
                    PAUSE 100
                    FOR counter = 1 TO 150
                        GOSUB dopredu
                    NEXT
                    GOTC podel_steny
                IF distance < 23 THEN
                    GOTC podel_steny
                ENDIF
            ENDIF
        LOOP
    ENDIF
ENDIF
IF (counter//16=0) THEN
    FOR temp = 660 TO 678
        PULSOUT servoPin, temp
        PAUSE 20
    NEXT
    GOSUB mereni
    DEBUG ?distance
    IF (distance<limit) THEN
        GOTC stena_limit
    ENDIF
ENDIF
LOOP
RETURN

```

Obrázek A.8: kód pro pohyb podél stěny



Obrázek A.9: vývojový diagram algoritmu pro navigaci po známé cestě