

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Využití robotických stavebnic při výuce předmětu Algoritmizace
Tomáš Slabý

Bakalářská práce
2021

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Tomáš Slabý**
Osobní číslo: **I17140**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Téma práce: **Využití robotických stavebnic při výuce předmětu Algoritmizace**
Zadávající katedra: **Katedra informačních technologií**

Zásady pro vypracování

Cílem práce bude navrhnout sadu příkladů za pomoci robotických stavebnic (Ozobot nebo BBC micro:bit – dodá vyučující) pro předmět Algoritmizace. Student provede analýzu stávajícího stavu v oblasti výuky algoritmizace na vysokých školách, zjistí, jaké se používají principy a jaké oblasti algoritmizace se ve výuce probírají. Dále navrhne sadu úloh, která bude využitelná ve výuce předmětu právě za pomoci robotických pomůcek. V práci by se měly objevit úlohy zaměřené na zpracování údajů z robotických senzorů, využití podmínek a cyklů při rozhodování robotů. Student si může připravit vlastní umělou plochu, ve které se bude robot pohybovat. Student navrhne řešení úloh v prostředí, které je vhodné pro daného robota.

Rozsah pracovní zprávy: **min. 30 stran**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

BAUER, Jiří, 2016. Využití robota Lego Mindstorms – návrh a realizace speciálních projektu. ČVUT FEL v Praze. Bakalářská práce.

BĚLÍK, Tomáš, 2010. Využití robota Lego Mindstorms – Návrh a realizace speciálních úloh. ČVUT FEL v Praze. Bakalářská práce.

Vedoucí bakalářské práce: **Ing. Jan Panuš, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. listopadu 2019**

Termín odevzdání bakalářské práce: **7. května 2020**



Ing. Zdeněk Němec, Ph.D.
děkan

Ing. Lukáš Čegan, Ph.D.
pověřený vedením katedry

V Pardubicích dne 17. prosince 2019

Prohlašuji:

Práci s názvem Využití robotických stavebnic při výuce předmětu Algoritmizace jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 5. 5. 2021

Tomáš Slabý

PODĚKOVÁNÍ

Tímto bych rád poděkoval svému vedoucímu práce Ing. Janu Panušovi, Ph.D., za cenné rady a za poskytnutí vybavení, které mi pomohlo při zpracování bakalářské práce. Dále děkuji Domče za podporu.

ANOTACE

Práce se zabývá návrhem sady úloh pro předmět algoritmizace pomocí robotických stavebnic a počítače BBC micro:bit. Teoretická část obsahuje analýzu stávajícího stavu v oblasti výuky algoritmizace na vybraných vysokých školách, popis robota Kitronik :MOVE mini MK2 a dalších pomůcek potřebných k daným úlohám. Dále práce seznamuje s počítačem BBC micro:bit a jeho možnostmi programování. V praktické části jsou realizovány úlohy se zaměřením na zpracování údajů z robotických senzorů.

KLÍČOVÁ SLOVA

algoritmizace, robot, BBC micro:bit, MakeCode, MicroPython

TITLE

Use of robotic kits in teaching the subject Algorithmization

ANNOTATION

The work deals with the design of a set of tasks for the subject of algorithmization using robotic kits and a BBC micro: bit computer. The theoretical part contains an analysis of the current state in the field of teaching algorithms at selected universities, a description of the robot Kitronik: MOVE mini MK2 and other tools needed for the tasks. The work also introduces the BBC micro:bit computer and its programming options. In the practical part, tasks are realized with a focus on data processing from robotic sensors.

KEYWORDS

algorithmization, robot, BBC micro:bit, MakeCode, MicroPython

OBSAH

Seznam obrázků	9
Seznam tabulek	10
Seznam zkratek	11
Úvod	13
1 Analýza výuky algoritmizace na vysokých školách	14
1.1 Popis analýzy	14
1.2 ČVUT FIT.....	14
1.3 VUT FIT	14
1.4 UPCE FEI	15
1.5 OSU PřF.....	15
1.6 Závěr analýzy.....	16
2 BBC micro:bit	17
2.1 Úvod.....	17
2.2 BBC Micro:bit V1 a V2.....	18
2.3 Hardwarové vybavení BBC micro:bit	19
2.3.1 Procesor	19
2.3.2 Displej	20
2.3.3 Tlačítka	20
2.3.4 Anténa.....	20
2.3.5 Akcelerometr	20
2.3.6 Kompas	20
2.3.7 Vstupně-výstupní piny	21
2.3.8 Micro-USB port	22
2.3.9 Konektor pro baterii	22
2.3.10 Resetovací tlačítko	23
2.4 Softwarové vybavení BBC micro:bit.....	23
2.4.1 Formát souboru HEX.....	24
2.4.2 DAL/CODAL	24
2.4.3 Arm mbed	25
2.4.4 DAPLink.....	25
2.4.5 Radio	26
2.4.6 Bluetooth.....	27
3 Programování BBC Micro:bit	28
3.1 Microsoft MakeCode	28
3.1.1 Blokový editor	28
3.1.2 Jazykový editor	29
3.1.3 Hello World	29
3.1.4 Nahrávání programu na BBC micro:bit.....	29
3.1.5 Mobilní aplikace Micro:bit	30
3.2 MicroPython	30
3.2.1 Python editor.....	30
3.2.2 MU editor.....	31

3.2.3	Hello World v MicroPythonu	32
3.2.4	Použité moduly v úlohách.....	32
4	Komponenty použité v úlohách	33
4.1	Kitronik :MOVE mini MK2	33
4.1.1	Kitronik :MOVE Sensor Interface Board	34
4.1.2	Kitronik :MOVE Line Following board	34
4.2	Grove Inventor Kit for micro:bit	35
4.2.1	Ultrazvukový měřič vzdálenosti	35
4.3	Waveshare Joystick pro micro:bit.....	35
5	Úlohy	36
5.1	Nakreslení čtverce.....	36
5.2	Zastavení robota před překážkou	37
5.3	Vyhýbání se překážkám.....	39
5.4	Jízda podél zdi	40
5.5	Parkovací senzor	41
5.6	Grafické zobrazení vzdálenosti.....	43
5.7	Ovládání robota pomocí gamepadu	46
5.8	Sledování čáry řízené semaforem	49
5.9	Sumo zápas	52
5.10	Přesouvání nákladu	53
	Závěr	56
	Použitá literatura	57
	Přílohy.....	60

SEZNAM OBRÁZKŮ

Obrázek 1: Přední a zadní strana Micro:bit. Zdroj: [8].....	19
Obrázek 2: Popis pinů na micro:bit. Zdroj:[5].....	21
Obrázek 3: Vrstvy na BBC micro:bit. Zdroj:[14].....	24
Obrázek 4: Schéma připojení počítače s cílovým MCU. Zdroj: [16].....	26
Obrázek 5: Hello World v prostředí MakeCode. Zdroj: Vlastní	29
Obrázek 6: Editor MU. Zdroj: Vlastní.....	31
Obrázek 7: Hello world v MicroPythonu. Zdroj: Vlastní.....	32
Obrázek 8: Robot Kitronik :MOVE mini MK2. Zdroj: Vlastní	34
Obrázek 9: Úloha 1. MakeCode. Zdroj: Vlastní.....	36
Obrázek 10: Úloha 2. Řešení v MakeCode. Zdroj: Vlastní	38
Obrázek 11: Úlohu 3. Řešení v Makecode. Zdroj: Vlastní.....	39
Obrázek 12: Úloha 4. Řešení v MakeCode. Zdroj: Vlastní	41
Obrázek 13: Úloha 5. Řešení v MakeCode pro robota. Zdroj: Vlastní	42
Obrázek 14: Úloha 5. Řešení v MakeCode pro reproduktor. Zdroj: Vlastní.....	42
Obrázek 15: Úloha 6. Řešení v MakeCode pro zobrazení grafu. Zdroj: Vlastní.....	44
Obrázek 16: Úloha 6. Řešení v MakeCode pro movemini. Zdroj: Vlastní	45
Obrázek 17: Úloha 7. Řešení v MakeCode pro gamepad. Zdroj: Vlastní	47
Obrázek 18: Úloha 7. Řešení v MakeCode pro robota. Zdroj: Vlastní	47
Obrázek 19: Úlohu 8. Řešení v MakeCode pro semafor. Zdroj: Vlastní.....	49
Obrázek 20: Úlohu 8. Řešení v MakeCode pro movemini. Část 1. Zdroj: Vlastní	50
Obrázek 21: Úlohu 8. Řešení v MakeCode pro movemini. Část 2. Zdroj: Vlastní	50
Obrázek 22: Úloha 9. Řešení v MakeCode. Zdroj: Vlastní	52
Obrázek 23: Úloha 10. Řešení v MakeCode. Část 1. Zdroj: Vlastní.....	54
Obrázek 24: Úloha 10. Řešení v MakeCode. Část 2. Zdroj: Vlastní	55

SEZNAM TABULEK

Tabulka 1: Vybrané předměty na vysokých školách.	14
Tabulka 2: Srovnání z hlediska funkcí micro:bit V1 a micro:bit V2.....	18
Tabulka 3: Srovnání specifikace micro:bit V1 a micro:bit V2.....	18
Tabulka 4: Popis funkce malých pinů na micro:bit.	22

SEZNAM ZKRATEK

ČVUT	České vysoké učení technické v Praze
VUT	Vysoké učení technické v Brně
UPCE	Univerzita Pardubice
OSU	Ostravská univerzita
FEI	Fakulta informatiky a elektrotechniky
FIT	Fakulta informačních technologií
PřF	Přírodovědecká fakulta
BBC	British Broadcasting Corporation
NVM	Non-Volatile Memory
PWM	Pulse Width Modulation
LED	Light-Emitting Diode
NC	No Connection
DAL	Device Abstraction Layer
HAL	Hardware Abstraction Layer
ARM	Acorn Risc Machine
I2C	Inter-Integrated-Circuit
USB	Universal Serial Bus
MCU	MicroController Unit
BLE	Bluetooth Low Energy
ISM	industrial, scientific and medical
DFU	Device Firmware Update
PTX	Programming Experience Toolkit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver-Transmitter

IDE	Integrated Development Environment
REPL	Read Eval Print Loop
CMSIS	Cortex Microcontroller Software Interface Standard
DAP	Debug Access Port
JST	Japan Solderless Terminal
GPIO	General-purpose input/output

ÚVOD

S rostoucím vývojem informačních technologií je důležité vzdělávat populaci v oblasti algoritmizace. Jedním ze způsobů vzdělávání může jít i hravou formou, a to s použitím robota a počítače BBC micro:bit.

Cílem bakalářské práce je proto navrhnout sadu příkladů pro předmět algoritmizace s využitím robotických stavebnic. Ta by měla sloužit jako výukový materiál pro studenty, kteří začínají s programováním a algoritmy.

V úvodu práce je provedena analýza stávajícího stavu v oblasti výuky algoritmizace na vysokých školách, kde je popsáno, jaké se používají principy a jaké oblasti algoritmizace se ve výuce probírají.

Ve druhé kapitole je teoretický popis hardwarového a softwarového vybavení počítače BBC micro:bit. V kapitole třetí se rozebírá problematika programování BBC micro:bit a používání různých vývojových prostředí. V následující kapitole je popsáno další hardwarové vybavení použité v úlohách, včetně robota Kitronik :MOVE mini MK2.

V poslední kapitole jsou realizovány úlohy, které jsou řešeny v jazyce MicroPython a ve vizuálním jazyce editoru Microsoft MakeCode. Z hlediska programování jsou v úlohách použity základní konstrukce, jako jsou podmínky a cykly. Úlohy jsou zaměřené na zpracování údajů z robotických senzorů a následnou reakci pohyblivého robota Kitronik :MOVE mini MK2.

Pro toto téma jsem se rozhodl z důvodu toho, že mě baví roboti a robotika obecně. Myslím si totiž, že je potřeba vzdělávat studenty v oblasti programování více a pokud možno zábavnou formou.

1 ANALÝZA VÝUKY ALGORITMIZACE NA VYSOKÝCH ŠKOLÁCH

1.1 Popis analýzy

Cílem této kapitoly je zjistit, jaký je stav výuky základů algoritmizace na vybraných vysokých školách. Vybral jsem si čtyři bakalářské studijní programy z různých vysokých škol se zaměřením na informační technologie. Jmenovitě se jedná o následující: Informatika na FIT ČVUT, Informační technologie na FIT VUT, Informační technologie na FEI UPCE, aplikovaná informatika na PřF OSU. Ke každému studijnímu programu jsem vybral předměty, kde se základy algoritmizace vyučují a následně analyzuji náplň předmětu podle sylabu. Všechny předměty jsou z prvního semestru a jsou vypsaný níže v Tabulka 1.

Tabulka 1: Vybrané předměty na vysokých školách.

Vysoká škola	Fakulta	Studijní program	Název předmětu
ČVUT	FIT	Informatika	Programování a algoritmizace 1
VUT	FIT	Informační technologie (IT-BC-3)	Základy programování
UPCE	FEI	Informační technologie (B0688A140009)	Základy algoritmizace Praktikum z algoritmizace a programování
OSU	PřF	Aplikovaná infomatika	Základy algoritmizace

Zdroj: [22],[23],[24],[25]

1.2 ČVUT FIT

Studenti si osvojí řešení základních problémů s využitím algoritmů a naučí se je programovat v jazyce C. Budou chápat datové typy, výrazy, funkce a příkazy. Naučí se základní algoritmy pro řazení, vyhledávání a složitost algoritmů. V tomto předmětu se z algoritmizace probírají tato témata: „*Algoritmizace, proměnné, přiřazovací příkaz, výrazy, řídicí struktury, větvení a cykly. Složitost algoritmů, algoritmy řazení a vyhledávání. Rozklad problému na podproblémy.*”[22]

1.3 VUT FIT

Výstupem předmětu Základy programování je naučit studenty analyzovat a vytvářet triviální algoritmy. Dále se seznámí s principy vyšších programovacích jazyků a zvolený jazyk budou moci využít pro implementace algoritmů. Porozumí rozvinuté Backus-Naurové formě pro popis programovacího jazyka. Pochopí základní algoritmy, které se poté naučí používat

v reálných problémech. Tyto problémy budou řešit v projektech, které se naučí zdokumentovat a obhájit. Algoritmy a pojmy jsou ilustrovány pomocí jazyka C.[23]

Témata probíraná v tomto předmětu z algoritmizace: „*Vlastnosti algoritmů, notace a zápis algoritmů. Techniky řešení problémů. Návrh a implementace algoritmů. Výrazy a příkazy. Řídící struktury. Bloková struktura programu, rozsah platnosti identifikátorů.*“ [23]

1.4 UPCE FEI

Základy algoritmizace seznámí studenty s algoritmickým způsobem myšlení, se základy algoritmizace a dále připraví na navazující předmět Základy programování. [36] Nevyužívá se programovacího jazyka, ale vývojového prostředí Raptor založeného na vývojových diagramech. Je speciálně navržen tak, aby studentům pomáhal vizualizovat algoritmy.

Témata probíraná v předmětu Základy algoritmizace: „*Pojem algoritmus, základní vlastnosti algoritmu. Principy dělení problému na podproblémy. Způsoby zápisu algoritmů. Vývojový diagram – symbolický algoritmický jazyk pro zobrazení algoritmu zpracování informací. Tvorba a zápis algoritmů s využitím specializovaného SW - krokování algoritmů, verifikace algoritmů a složitost algoritmu. Sekvenční algoritmy. Větvení algoritmů. Cyklické algoritmy. Třídící algoritmy. Výroková logika - logické spojky, symboly jazyka výrokové logiky, syntaktická pravidla, interpretační pravidla.*“[36]

Druhým předmětem je Praktikum z algoritmizace a programování, který je primárně určen pro studenty, kteří nemají dostatečné praktické zkušenosti s algoritmizací a programováním. V předmětu se probírají praktické příklady z učiva předmětu Základy algoritmizace. [24]

1.5 OSU PŘF

Cílem předmětu je naučit zapsat jednoduché programy v programovacím jazyce Java. Dále studenti pochopí: „*problematiku objektově orientovaného programování, referenčních a primitivních datových typů, dovolených operátorů, s vyhodnocování aritmetických a logických výrazů.*“ Z hlediska základních programovacích struktur se seznámí s posloupností, větvením a cykly, metodami a jejich následnými parametry či přetížením.[25]

Témata probíraná v tomto předmětu z algoritmizace: „*Algoritmus. Proměnná a základní datové typy, práce s proměnnou. Operátory relační a logické, operátor přiřazení a vyhodnocování logických výrazů. Základní programové konstrukce. Rozhodovací příkazy jazyka Java a úplné, neúplné, vnořené větvení, vícenásobné větvení. Příkaz s podmínkou na začátku, příkaz cyklu s podmínkou na konci, příkaz cyklu s řídící proměnnou.*“[25]

1.6 Závěr analýzy

Ve všech předmětech se nachází i úvod do programování, který jsem se snažil záměrně vynechat. Soustředil jsem se pouze na základy algoritmizace. Jednotlivé přístupy škol se primárně liší programovacím jazykem, ve kterém představují algoritmizaci. Dalším rozdílem je na vysoké škole v Pardubicích, kde je výuka rozdělena do dvou předmětů, na rozdíl od ostatních škol, kde mají na základy algoritmizace pouze jeden předmět. Probíraná jednotlivá témata se na vysokých školách liší, jádro však zůstává stejné.

2 BBC MICRO:BIT

2.1 Úvod

BBC micro:bit (někdy též zkráceně micro:bit) je jednodeskový počítač, který primárně slouží ke vzdělávání zábavným způsobem. Uživateli může pomoci pochopit principy programování a dozvědět se, jak fungují elektronické součástky. [1]

Celý projekt navazuje na BBC Micro z roku 1982, což byl počítač určený pro zlepšení počítačové gramotnosti.[6] BBC zná téměř každý, jedná se o britskou veřejnoprávní společnost, zajišťující vysílání rozhlasového a televizního vysílání, která tento projekt financuje.[3] V roce 2015 zahájila BBC iniciativu Make It Digital, jejímž cílem mělo být naučit děti programovat a ovládat elektroniku. Důvodem vytvoření této iniciativy byla i skutečnost, že studenti ve Spojeném království maturovali téměř bez znalostí programování, což mělo za následek nedostatečný počet kvalifikovaných pracovníků v informačních technologiích.[2],[18]

Bylo stanoveno několik cílů, jaké by měl počítač splňovat. Jedná se o tyto cíle:

- Vytvořit, co nejlevnější zařízení, aby bylo snadno dostupné a aby šlo lehce nasadit do současné školní IT infrastruktury.
- Vzbudit kreativitu, k čemuž je potřeba nabídnout zajímavý úvod do programování. Projekt se tedy snažil studenty proměnit z pouhých konzumentů na digitální tvůrce.
- Poskytnout jednoduché programování a zapojení počítače, aby to zvládlo i dítě a s narůstajícími znalostmi umožnit uživatelům vzdělávat se ve složitosti aplikace i v programovacím jazyce.
- Výpočetní technologie se stávají více propojenými a všudypřítomnými. Tento trend bude podle tvůrců projektu pokračovat, takže i zařízení by mělo svou povahou zachycovat jeho propojenou a vestavěnou povahu.
- Využitelnost i mimo informační technologie. Projekt by měl podnítit zvědavost v různých oborech od umění až po matematiku.[18]

Do projektu se nakonec zapojilo 29 organizací, které pomohli projekt realizovat. [18] V roce 2016 bylo ve veřejném britském školním systému zdarma rozdáno cca jeden milion micro:bit a to všem studentům 7. ročníků, tedy dětem okolo jedenácti let.[6] Celý projekt v roce 2016 přešel z BBC na neziskovou organizaci Micro:bit Educational Foundation, která šíří povědomí o micro:bit k edukačním účelům. [3]

2.2 BBC Micro:bit V1 a V2

Úspěšnost počítače dokazuje i to, že se dočkal novější verze. Čtyři roky od uvedení první verze byl v říjnu roku 2020 představen micro:bit V2. Ten nabízí všechny funkcionality, kterými disponovala již první verze a navíc obsahuje i nové funkce a vylepšení. Všechny jsou vypsány níže v tabulkách. Mezi hlavní nové funkce V2 patří reproduktor a mikrofon. Vylepšení se obzvlášť dočkalo zvýšení paměti Flash a RAM.[7]

Výhodou je, že si uživatelé při programování nebudou muset vybírat jakou verzi microbit zrovna mají. Díky novému formátu tzv. univerzálnímu hexu může kód běžet na obou verzích. V této bakalářské práci se budu výhradně zabývat micro:bit V1 ve verzi v1.5. [7],[17]

Tabulka 2: Srovnání z hlediska funkcí micro:bit V1 a micro:bit V2

Vlastnosti	micro:bit v1.5	micro:bit v2
LED matice	25 LED	25 LED
Akcelometr	3-osé snímání pohybu	3-osé snímání pohybu
Kompas	Integrovaný magnetometr	Integrovaný magnetometr
Snímání teploty	Integrovaný snímač	Integrovaný snímač
Uživatelská tlačítka	2x programovací tlačítka, A & B	2x programovací tlačítka, A & B
Bluetooth	BLE Bluetooth 4.0	BLE Bluetooth 5.0
Radio	2.4Ghz Micro:bit Radio	2.4Ghz Micro:bit Radio
Resetovací tlačítko	Tlačítko umístěné vzadu	Tlačítko umístěné vzadu
Vypnutí počítače	Není obsažen	Podržím resetovacího tlačítka
Výstup konektoru	25 pinů	25 pinů
Mikrofon	Není obsažen	MEMS mikrofon
Reproduktor	Není obsažen	Integrovaný reproduktor
Dotykové logo	Není obsažen	Ano
Indikátor napájení	Není obsažen	LED indikátor napájení

Zdroj: [7]

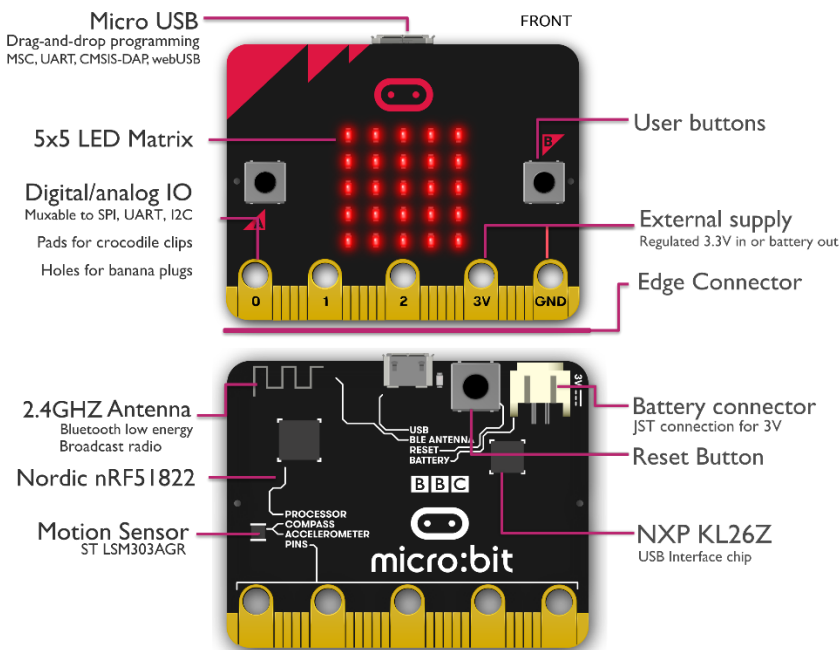
Tabulka 3: Srovnání specifikace micro:bit V1 a micro:bit V2

Vlastnosti	micro:bit v1.5	micro:bit V2
Procesor	Nordic Semiconductor nRF51822	Nordic Semiconductor nRF52833
Flash paměť	256 kB	512 kB
RAM paměť	16 kB	32 kB
Vývojová deska/rozhraní	NXP KL26Z	NXP KL27Z
I2C	Sdílená sběrnice I2C	Vyhrazená sběrnice I2C pro periferní zařízení
Napájení	3V přes konektor nebo baterii, 5V přes micro-USB port	3V přes konektor nebo baterii, 5V přes micro-USB port
Proud na výstupu	90 mA pro příslušenství	200 mA pro příslušenství
Pohybový senzor	ST LSM 303	ST LSM 303
Rozměry	5 cm (šířka) x 4 cm (výška)	5 cm (šířka) x 4 cm (výška)

Zdroj: [7]

2.3 Hardwarové vybavení BBC micro:bit

Micro:bit je malá deska s plošnými spoji o rozměrech 4,5 x 5 cm, která již obsahuje řadu namontovaných komponentů. Na přední straně má BBC micro:bit displej a dvě tlačítka. Zadní strana zahrnuje resetovací tlačítko, micro-USB konektor, radio a další komponenty. [2]



Obrázek 1: Přední a zadní strana Micro:bit. Zdroj: [8]

2.3.1 Procesor

Procesor je umístěný v levém horním rohu na zadní straně BBC micro:bit. Ve standardním stolním počítači je oddělena operační paměť, úložiště a procesor. V micro:bit jsou avšak všechny tyto komponenty zahrnuty do jednoho integrovaného obvodu označovaného jako systém na čipu.[1]

Procesor má architekturu typu ARM, který byl navržen tak, aby při nízké spotřebě energie poskytoval vysoký výkon, takže při používání doporučených baterií by měl počítač vydržet několik měsíců. Typ procesoru je Nordic nRF51822, který obsahuje 16 Mhz jednojádrový Cortex-M0. Paměť RAM má 16 kB a energeticky nezávislou paměť (NVM) pro ukládání programu, ta tvoří 256 kB.[1]

BBC micro:bit má i druhý procesor KL26, který je umístěn vpravo nahoře na desce. Když se připojí pomocí micro-USB k počítači, tak tento druhý procesor řídí komunikaci a dále přenáší nový program do hlavního procesoru. [1]

2.3.2 Displej

Pro program běžící na tomto počítači je displej primární výstupní prostředek pro komunikaci, skládá se z matice 5 x 5 červených led diod. Led dioda tvoří jeden pixel displeje, u kterého lze měnit jas každého pixelu. Celkově má tedy 25 pixelů, což stačí na základní funkce. Umožňuje zobrazit různé vzory, písmena, číslice, prvky jednoduché hry, graf nebo obrázek. Displej lze také naprogramovat tak, aby detekoval, měřil a reagoval na úrovně světla.[12]

2.3.3 Tlačítka

Další velmi viditelnou částí počítače jsou dvě tlačítka označena písmenem A a B, jsou umístěna po stranách displeje a tvoří primární vstupní zařízení pro počítač. Možné jsou tři vstupy: stisknutí tlačítka A, stisknutí tlačítka B a stisknutí obou tlačítek zároveň. Technicky se tlačítka definují jako okamžitý spínač. [1]

2.3.4 Anténa

Micro:bit V1 neobsahuje žádnou externí anténu, místo toho je do desky počítače zabudována měděná dráha. Můžeme ji nalézt vlevo nahoře na zadní straně desky pod označením „BLE ANTENNA“. Anténu využívají dvě důležité funkce počítače: radio a bluetooth. [1]

2.3.5 Akcelerometr

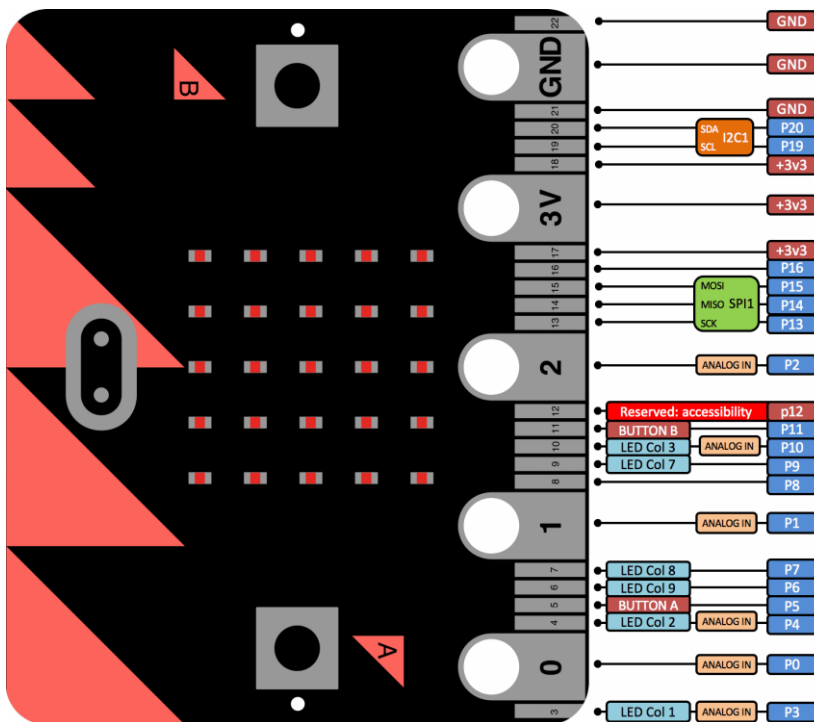
Jeden ze dvou senzorů, které jsou zabudované v tomto počítači je akcelerometr. Senzor měří hodnotu zrychlení a přesně ví, jak je micro:bit situován v prostoru v daném čase. Senzor sleduje úhel počítače na všech třech osách X, Y a Z, to znamená, že detekuje i mírný posun nebo zatřesení. Akcelerometr se využívá například v mobilních telefonech, kde umožňuje automatické obrácení obrazovky při otočení telefonu. [1], [3]

2.3.6 Kompas

Druhým vestavěným senzorem v BBC micro:bit je kompas. Obdobně jako u akcelerometru, jde o nepatrný senzor umístěný na zadní straně počítače. Pracuje jako navigační nástroj stejného jména, který slouží pro nalezení a nasměrování k magnetickému severu. Funguje na principu snímání magnetického pole, tedy podobně jako klasické kompasů na bázi jehly. Oproti klasickému kompasu má jednu zajímavou funkci navíc, kromě magnetického pole Země umí indikovat i jiná magnetická pole. Takže ve výsledku může detekovat kov, což na druhou stranu znamená, že okolní magnetická pole mohou ovlivnit jeho přesnost. [1]

2.3.7 Vstupně-výstupní piny

BBC micro:bit můžeme rozšířit ve formě vstupně-výstupních pinů. Lze je využít buď k odeslání vstupu nebo pro přijímání výstupu. Jsou to proužky mědi na obou stranách BBC micro:bit desky. „Technický termín pin se nevztahuje na spojení samotná, ale na pin procesoru, ke kterému jsou spoje připojena“ [1] „(překlad autora)“. Celkově obsahuje pět velkých pinů a dvacet malých pinů. Velké piny jsou též někdy označovány jako prstény z důvodu pro jejich kruhovitý tvar, který umožňuje připojit hardware pomocí krokosvorky nebo banánových konektorů. Jejich označení je vidět na přední straně desky: 0, 1, 2, 3V a GND. První tři velké piny jsou univerzální pro libovolné účely, zkráceně se nazývají GPIO. GND zajišťuje uzemnění obvodu pro ochranu před chybou. Pin 3V slouží hlavně pro napájení výstupních periférií. Dalším využitím pinu může být přímo napájení micro:bit, čímž se však obejde ochrana proti vysokému napětí a zkratu, což může poškodit počítač. [5], [1]



Obrázek 2: Popis pinů na micro:bit. Zdroj:[5]

Malé piny jsou označeny číslem od 3. do 22. (viz Obrázek 2) a jsou sdíleny s dalšími částmi na desce, například pin 3 je sdílen s LED diodami, pokud se tedy využívá displej micro:bit, tak se nemůže použít pin 3 pro další funkci. [5]

Tabulka 4: Popis funkce malých pinů na micro:bit.

Číslo pinu	Funkce pinu
3	Sdílen se sloupcem 1 na LED matici. Pokud je displej vypnut, tak lze pin použít jako digitální I/O nebo ADC.
4	Sdílen se sloupcem 2 na LED matici. Pokud je displej vypnut, tak lze pin použít jako digitální I/O nebo ADC.
5	Externě detekovat spuštění tlačítka A.
6	Sdílen se sloupcem 9 na LED matici. Pokud je displej vypnut, tak lze pin použít jako digitální I/O.
7	Sdílen se sloupcem 8 na LED matici. Pokud je displej vypnut, tak lze pin použít jako digitální I/O.
8	Digitální I/O.
9	Sdílen se sloupcem 7 na LED matici. Pokud je displej vypnut, tak lze pin použít jako digitální I/O.
10	Sdílen se sloupcem 9 na LED matici. Pokud je displej vypnut, tak lze pin použít jako digitální I/O nebo ADC.
11	Externě detekovat spuštění tlačítka B.
12	Rezervován pro usnadnění přístupu.
13	Pro hodinový signál(SCK) SPI sběrnice.
14	Pro signál Master In Slave Out (MISO) sběrnice SPI.
15	Pro signál Master Out Slave In (MOSI) sběrnice SPI.
16	Pro signál Chip Select sběrnice SPI.
17, 18	Připojeny k napájecímu napětí 3V.
19,20	Implementuje hodinový signál (SCL) a datovou linku (SDA) sběrnice I2C.
21, 22	Připojeny ke GND.

Zdroj: [5]

2.3.8 Micro-USB port

Nalezneme jej uprostřed zadní strany počítače. Jeho prvním účelem portu je poskytování energie pro BBC micro:bit, pokud zrovna nepoužíváme baterii. Druhým účelem portu je připojení BBC micro:bit k počítači, pomocí kterého můžeme komunikovat se zařízením a posílat do něho vlastní program. Počítač nemá žádné napájecí tlačítko, takže při připojení s micro-USB konektorem se BBC micro:bit sám spustí. Počítač vyžaduje k provozu 3,3 voltů, naproti tomu USB dodává napětí kolem pěti voltů. Ke snížení vstupního napětí se používá regulátor na úroveň, kterou může deska používat, když je napájena počítačem. Port by neměl být používán pro vysokorychlostní USB nabíječky, pro speciální USB porty a ani pro vysoce výkonné USB baterie, protože tyto typy nabíjení mohou poškodit počítač. [2]

2.3.9 Konektor pro baterii

Napájení pomocí baterie usnadňuje práci s BBC micro:bit, Pokud počítač používáme pouze na jednom místě, tak pravděpodobně bude stačit jen napájení pomocí micro-USB portu. Když však potřebujeme s počítačem pracovat dál od pracovního stolu, či s ním nějakým způsobem pohybovat, tak bude napájení pomocí baterie vhodnější. Baterie jsou vloženy v ochranném držáku a prostřednictvím kabelu připojeny k BBC micro:bit. Kompatibilní ochranný držák má

konektor JST PH a dva sloty pro baterii typu AAA. Po připojení baterie se BBC micro:bit stejně jako u napájení pomocí USB automaticky spustí. A stejně je tomu i při odpojení JST konektoru, kdy se micro:bit vypne. [1]

2.3.10 Resetovací tlačítko

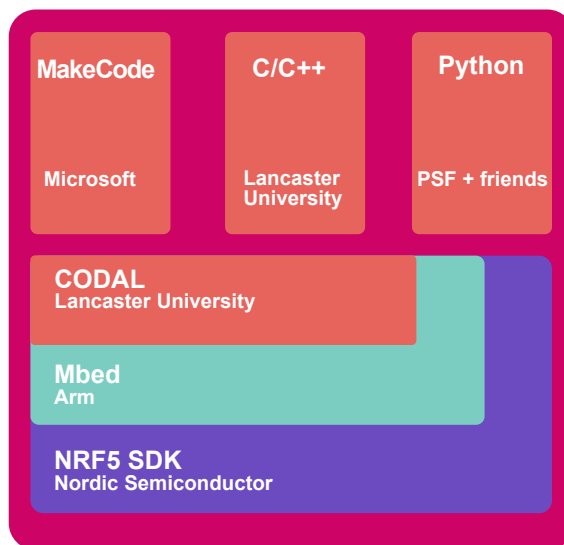
Je umístěno na zadní straně mezi konektorem na baterii a micro-USB portem. Stisknutím tohoto tlačítka dojde k odeslání signálu do procesoru, který na krátký čas přeruší napájení a program se spustí od začátku. Toto tlačítko se nám může hodit z několika důvodů, například některé programy jsou určeny pouze k jednomu spuštění a možným způsobem, jak je spustit znova může být právě pomocí resetovacího tlačítka. Dále lze použít při testování programu a následného hledání chyb, kde se program může spustit opět pomocí tohoto tlačítka.

Technicky je možné spustit program znova i odpojením a připojením napájení, avšak toto řešení zbytečně zatěžuje napájecí konektory, které může poškodit. BBC micro:bit lze vypnout či resetovat bez toho, aniž by se poškodil, protože na rozdíl od klasického počítače neprochází procesem vypínání. To samé platí pro odpojení micro-USB kabelu či baterie. K poruše může dojít pouze při nahrávání programu, který ale nezpůsobí trvalé poškození. Pokud dojde k tomuto problému, stačí připojit BBC micro:bit k počítači a program znovu nahrát.[1]

2.4 Softwarové vybavení BBC micro:bit

Při programování pro micro:bit se na aplikaci spojí více softwaru nižší úrovně, které nakonec vytvoří finální hex soubor. Mezi vrstvy nižší úrovně patří DAL/CODAL, Arm mbed, Nordic nRF5 SDK a DAL, některé z nich byly vytvořeny speciálně pro micro:bit, jako například DAL. Tyto softwarové vrstvy umožňují snadné používání programovacích jazyků, jako je JavaScript či Python. [30]

Nadace Micro:bit Educational Foundation zastává open source politiku. To znamená, že zdrojové kódy, které nadace publikuje pod licencí open source mohou být kontrolovány, upravovány a vylepšovány absolutně kýmkoliv.[9]



Obrázek 3: Vrstvy na BBC micro:bit. Zdroj:[14]

2.4.1 Formát souboru HEX

Každý editor pro micro:bit dokáže vygenerovat hex soubor, ten obsahuje kód programu napsaný ve formátu, kterému micro:bit rozumí. Hex je ve formátu Intel hex, kde jeden řádek reprezentuje jeden záznam. Každý řádek začíná znakem dvojtečky a končí kontrolním součtem, kterým lze ověřit integritu dat. Každý hexadecimální kód zahrnuje binární objekt, který umožňuje používat radio a Bluetooth.[29]

Po vydání micro:bit ve verzi V2 byl uveden univerzální hex kód, což je nadmnožina formátu Intel hex. Univerzální hex byl vyvinut za účelem lehkého přechodu mezi variantami a umožňuje kompatibilitu kódu na starší V1 i novější verzi V2. Kompilovaný soubor univerzálního hexu má větší velikost, cca v řádech megabajtů, zatímco starý hex zabírá prostor v řádech stovek kilobajtů.[29]

2.4.2 DAL/CODAL

Je software též známý jako runtime, který slouží pro podporu většiny programovacích jazyků pro micro:bit. Umožňuje se ponořit hlouběji do problematiky, pokud chceme například naprogramovat vylepšení pro micro:bit nebo hardwarová rozšíření. DAL je základní sada mechanismů a ovladačů, která byla vytvořena pomocí ARM mbed.[14]

DAL má sadu funkcí pro zpracování jazyků vyšší úrovně a usnadňuje programování v C nebo C++. Poskytuje jednoduché API pro C/C++ programy, dokonce i mnoho bloků v Microsoft MakeCode používá funkce, které jsou v DAL. Runtime vyvinula univerzita v Lancasteru s cílem podpory informatiky na školách.[14], [30]

Klíčové komponenty DAL jsou:

- Plánovač, pomocí kterého může micro:bit vykonávat více úkolů paralelně.
- Systém událostí, který umožňuje psát reaktivní kód. Když se stane nějaká událost, tak program může informovat uživatele.
- Před složitostí správy paměti, ochrání programy spravované typy, které byly původně implementovány na vyšší programovací jazyky, ale nakonec našly uplatnění pro programátory v C nebo C++.
- Ovladače zařízení představující hlavní hardwarové bloky na micro:bit včetně displeje, senzorů atd.[14]

2.4.3 Arm mbed

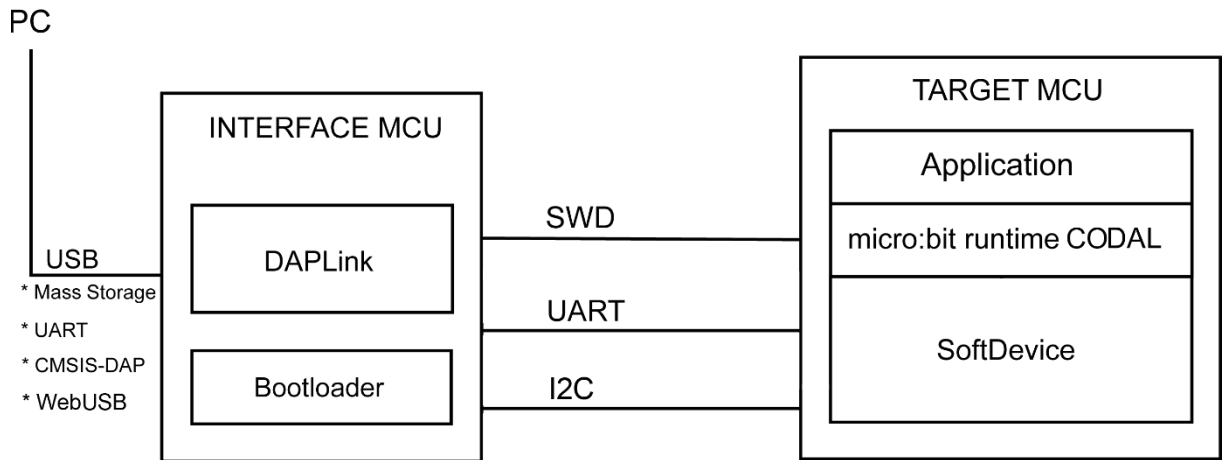
Poskytuje standardizované ovladače pro periferní zařízení a abstrahuje nízko-úrovňový hardware. ARM mbed má jednotnou abstraktní vrstvu HAL pro čipy, které využívají procesor ARM Cortex. Toho můžou využít vývojáři při psaní kódu, protože program, který byl vyvinut na ARM mbed platformě je spustitelný i na micro:bit, jsou-li k dispozici dané periférie a dostatek paměti. To znamená, že vývojáři mohou využít obrovské množství knihoven. Ve vyšších úrovních prostředí umožňuje přístup k perifériím jako jsou I2C, SPI nebo sériovému portu.[30][14]

2.4.4 DAPLink

DAPLink je open source projekt, který umožňuje ladit a programovat aplikace běžící na procesorech Arm Cortex. BBC micro:bit je snadno použitelný právě díky DAPLink, protože umožňuje ladění typu drag and drop bez nutnosti instalace ovladačů.[16]

Aby se dala plně pochopit problematika DAPLink, tak je potřeba vysvětlit pojem MCU rozhraní. Jednou z vlastností micro:bit je, že se po připojení pomocí USB prezentuje jako USB disk bez nutnosti jakýchkoliv ovladačů, což značně ulehčuje nahrávání programů na micro:bit. Tato vlastnost je možná díky rozhraní MCU, jež je vyhrazeno pro připojení USB, programování a pro ladění. Takovým rozhraním (čipem) je na micro:bit Freescale KL26Z. Čip, na kterém běží program a jsou k němu připojeny periférie se nazývá „cílový MCU“, což

je v případě BBC micro:bit procesor Nordic nRF51822. Cíl a rozhraní MCU jsou spolu propojeny viz Obrázek 4. DAPLink je součástí Arm Mbed.[16]



Obrázek 4: Schéma připojení počítače s cílovým MCU. Zdroj: [16]

Software DAPLink poskytuje vývojářům čtyři koncové body USB:

- Programování USB Drag-and-drop, díky kterému lze programy v hexadecimálním nebo binárním formátu přesunout na micro:bit, který se prezentuje jako USB disk. Z USB disku se po přesunutí nahraje program do paměti cílového MCU.
- Zajišťuje sériový průchod z cílového MCU do počítače. Takhle například vývojář dostane zprávu z kódu, který napíše do počítače.
- Ladění na základě CMSIS-DAP. Poskytuje kanál, který běží přes ladící protokol CMSIS-DAP, což umožňuje použít pokročilé debuggery.
- Prostřednictvím webového prohlížeče podporující WebUSB lze komunikovat se zařízením.[16]

2.4.5 Radio

Radio modul je součástí hlavního procesu a slouží pro dvě funkce. První z nich je bezdrátové seskupení více BBC micro:bit bez nutnosti kabelu. Druhá funkce má na starost komunikaci s ostatními zařízeními pomocí technologie Bluetooth Low Energy.[1]

Radio je jednou z nejzajímavějších funkcí tohoto počítače, umožňuje komunikovat s ostatními BBC micro:bit nebo s jinými počítači, který používají stejný protokol. Na rozdíl od BLE radio používá jen jednoduchý protokol, který se nestará o bezpečnostní protokoly ani o vztahy

důvěryhodnosti, takže si pouze přeposílá data s jinými zařízeními, které používá stejný komunikační protokol. Bluetooth a radio nelze používat současně.[4]

Při programování radia v MakeCode je třeba nastavit skupinu pro komunikaci mezi počítači. V MakeCode lze odesílat řetězec i číslo pomocí bloků send. Dále obsahuje bloky onReceived, které po přijetí dat vykonají akci uvnitř svého bloku. V Micropythonu se aktivuje radio příkazem on. Na rozdíl od MakeCode se nemusí nastavovat skupina, protože je defaultně nastavena na 0. Posílat a přijímat lze pouze řetězec pomocí příkazů send a receive.[38]

2.4.6 Bluetooth

V DAL je definovaný Bluetooth profil, který se skládá z různých služeb a charakteristik pro snadný přístup k hardwaru počítače. BBC micro:bit V1 obsahuje verzi Bluetooth 4.1 s technologií BLE označující nízkoenergetickou verzi bezdrátového standardu Bluetooth. Pracuje v pásmu ISM od 2,4 GHz až do 2,41 GHz. Rychlost přenosu dat záleží na tom, jak aplikace využívají funkci Bluetooth, ale v nejlepším případě to bude v řádech stovek kilobajtů za sekundu. Tato technologie umožňuje komunikovat s jinými zařízeními podporující BLE, například smartphony. [15]

Mnoho vývojových prostředí nabízí omezené nebo vůbec žádné připojení Bluetooth, na MicroPythonu není Bluetooth vůbec povoleno z důvodu náročnosti na paměť. Na webových stránkách tech.microbit.org je k dispozici předkompilovaný hexadecimální soubor umožňující službu Bluetooth. Tento hex z důvodu paměti neobsahuje, buď službu magnetometru nebo službu DFU. [15]

3 PROGRAMOVÁNÍ BBC MICRO:BIT

Kolem micro:bit je velká komunita lidí, kteří pracují na vylepšení a rozšíření micro:bit na jiné platformy. Uživatelé si tak mohou vybrat ze spousty oficiálních či neoficiálních editorů a programovacích jazyků. [10]

Pro začátečníky v programování nebo menší děti jsou vhodné vizuální programovací jazyky, které jsou znázorněny v grafickém zobrazení. Mezi vizuální programovací jazyk pro micro:bit patří nejvíce podporovaný Microsoft MakeCode, ale můžeme využít například i Scratch. Pro klasické textové programování slouží Python (popřípadě MicroPython) nebo JavaScript. Existují i další programovací jazyky, které nám umožní komunikovat s micro:bit, mezi ně patří kupříkladu Rust, Forth, Pascal, Ada, Sniff, Ulisp, C#/F# atd. [10]

3.1 Microsoft MakeCode

Je to online webový editor vycházející z frameworku PTX, který umožňuje vytvořit program pomocí bloků.[11] MakeCode se obvykle nazývá editor, ale poskytuje mnohem více funkcí: kombinuje editor programovacího jazyka, editor bloků, simulátor cílového zařízení a generátor cílového kódu. Umožňuje vytvářet a spouštět uživatelské programy na skutečném nebo simulovaném počítači. [13]

MakeCode využívá pro uživatele přístupný model programování pomocí bloků, které se přímo přepisují do řádků kódu v programovacím jazyce, uživatel tak může postupně přejít na psaní komplexnějších programů v daném jazyce.[13]

3.1.1 Blokový editor

Blokový editor funguje intuitivně a velice jednoduše, uživatel si bloky přetáhne tažením z panelu nástrojů a umísťuje je do pracovního prostoru. Bloky představují programovací struktury, které se dají zapsat v klasickém programovacím jazyce. Ty se nacházejí v jednotlivých kategoriích panelu nástrojů. Některé konstrukce jako cykly, podmíněné příkazy a události mají bloky, do kterých se vnořují jiné bloky. Takové bloky zapadající do ostatních bloků se nazývají bloky funkce a přiřazení. MakeCode ve výchozím nastavení obsahuje jen omezené množství bloků. Lze si ovšem stáhnout různá rozšíření z internetu, které obsahuje různé speciální bloky. Možné je i vytvářet vlastní bloky, které si pak uživatel do projektu načte pomocí rozšíření. [13]

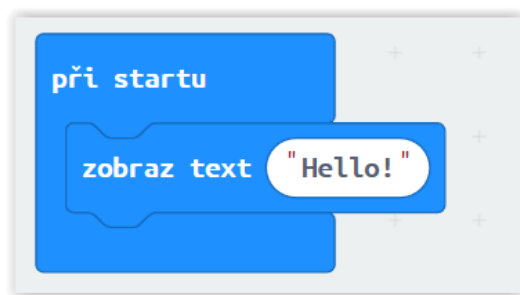
3.1.2 Jazykový editor

Doplňuje editor bloků, v němž dané bloky odpovídají kódu v jazykovém editoru. Uživatel si může jednoduše překlíknout mezi bloky a programovacím kódem, což pomůže uživateli postupně přejít na textové programování. V jazykovém editoru může uživatel napsat komplexnější kód, než jaký se dá vytvořit z bloků. Na výběr je v editoru ze dvou programovacích jazyků JavaScript a Python.[13]

3.1.3 Hello World

Na vytvoření programu se nemusí nic instalovat, stačí jít na webovou stránku makecode.microbit.org. Na této stránce vytvoříme a pojmenujeme projekt. Celý projekt se ukládá do lokální mezipaměti prohlížeče, takže se projekt po vymazání této paměti smaže. Po otevření projektu se načte pracovní plocha, na které jsou na začátku vždy dva bloky s názvem: „při startu“ a „opakuj stále“. Část uvnitř bloku „při startu“ se spustí vždy při zapnutí micro:bit nebo když se restartuje, tedy stisknutím restartovacího tlačítka. [11]

Blok „opakuj stále“ poběží dokola, dokud se program neukončí. Blok „opakuj stále“ není v tomto programu potřeba, můžeme ho proto odstranit stisknutím klávesy delete. V základní kategorii si vybereme blok „zobraz text“ a přetáhneme jej dovnitř bloku „při startu“. Nakonec si uživatel může finální program nahrát do micro:bit nebo odzkoušet na simulátoru stisknutím tlačítka zapnout simulátor.



Obrázek 5: Hello World v prostředí MakeCode. Zdroj: Vlastní

3.1.4 Nahrávání programu na BBC micro:bit

Finální program lze přesunout, buď USB kabelem nebo bezdrátově pomocí Bluetooth. V MakeCode se po kliknutí na tlačítko „stáhnout“ převede kód z editoru do nativního formátu cíle a vygeneruje HEX soubor. Po připojení micro:bit pomocí USB kabelu k počítači se

micro:bit díky DAPLinku tváří jako USB jednotka s názvem MICROBIT. Po přetáhnutí hex souboru z našeho počítače na jednotku MICROBIT se programový zavadač již sám načte HEX soubor. Micro:bit zabliká a spustí nový program. [11]

3.1.5 Mobilní aplikace Micro:bit

Alternativnější možnost je programovat počítač micro:bit pomocí mobilní aplikace Micro:bit. Tuto aplikaci lze stáhnout na přes Google play, nebo Appstore. V aplikaci můžeme vyvíjet program úplně stejně jako v Microsoft MakeCode, buď pomocí bloků nebo jazyků JavaScript či Python. Komunikace mezi mobilem a micro:bit probíhá přes technologii Bluetooth. [35]

Na Google play má aplikace k roku 2021 velmi špatné hodnocení a spoustu uživatelů si stěžuje na nefunkčnost celé aplikace. Z mé osobní zkušenosti však aplikace funguje dobře, bez problémů.

3.2 MicroPython

Je to reimplementace programovacího jazyka Python 3 určeného pro menší zařízení jako micro:bit. Nabízí klasické funkce jako u ostatních programovacích jazyků: základní datové typy, třídy, výjimky, generátory, seznamy, slovníky atd. Navíc má spoustu zajímavých funkcí: syntetizátor řeči, hudební programovací jazyk a nabízí různé způsoby připojení k zařízením, jakými jsou I2C, SPI a UART a další. Jak již bylo popsáno v kapitole 2.4.6, tak MicroPython nepodporuje Bluetooth a používá jednoduchý radio modul. spuštěním programu používá MicroPython kombinaci kompilátoru a interpretu. Nejdříve analyzuje script, jehož výsledkem je abstraktní syntaktický strom. AST se zkompiluje do Python bytecode, který se poté předá do virtuálního stroje, nikoliv do skutečného počítače a spustí se program. Pro MicroPython existuje mnoho editorů, mezi nejpoužívanější patří online prohlížečový Python editor nebo offline MU.[31]

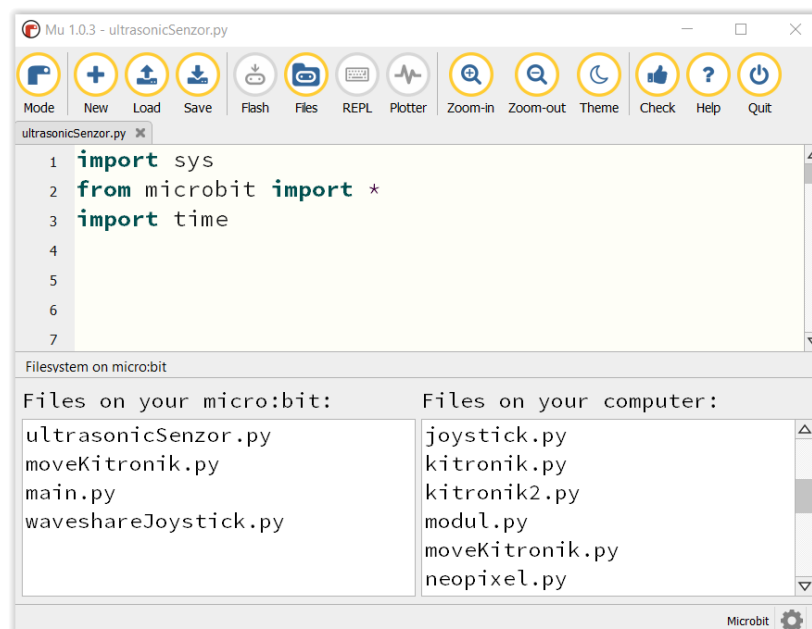
3.2.1 Python editor

Jedná se o online webový editor, který byl primárně navržen pro studenty a učitele, protože při promítání na tabuli, nebo velké obrazovce lze snadno zvětšit velikost. V editoru se můžou nahrávat nebo stahovat soubory s příponou hex nebo py. Mezi další funkce patří zvýraznění syntaxe, skládání kódu a automatické odsazení. Editor je dostupný na této webové adrese: python.microbit.org/v/2. [32]

3.2.2 MU editor

Na rozdíl od předchozího webového editoru se vývojové prostředí MU instaluje do počítače. Existuje mnoho důvodů, proč je lepší instalovat IDE na počítač. Za prvé je zde možnost ukládat si kód do lokální paměti. Další výhodou je funkce doplňování kódu, když si programátor není jistý, jakou funkci má zrovna použít nebo jaké parametry jsou ve funkci použity. Například Microsoft MakeCode sice též obsahuje tuto funkci, ale jeho slovník je velmi omezený. MU editor podporuje pouze programovací jazyk Python a lze nainstalovat na operační systémy Windows, Mac a Linux.[2]

Uživatelské rozhraní je velmi přívětivé, význam většiny ikon je zřejmý z názvu. Ikony new, load a save slouží pro uložení programu do počítače. Ikona Flash načte aktuální program do micro:bit, kód však načte bez předchozí kontroly. V případě chyby sice micro:bit vypíše chybu na LED displej, ale číst chybu na tomto displeji je poměrně komplikované, proto je lepší v editoru kód předem pořádně zkontrolovat. REPL poskytuje tzv. živé programování. Na kód, který napíšeme dostaneme okamžitou zpětnou vazbu z micro:bit. Ikona Files nám umožní vidět soubory, které jsou na počítači micro:bit. Po kliknutí na Files se v dolní části editoru zobrazí dva panely vedle sebe viz Obrázek 6. Levý panel ukazuje soubory, které jsou na micro:bit a pravý panel zobrazuje soubory nacházející se v adresáři mu_code. Pro přidání souboru na micro:bit stačí přetáhnout daný soubor myši z pravého panelu do levého.[2]



Obrázek 6: Editor MU. Zdroj: Vlastní

3.2.3 Hello World v MicroPythonu

Každý program psaný pro micro:bit musí obsahovat příkaz `import microbit`, který načte modul `microbit` ovládající počítač. Znak hvězdička znamená, že se importují všechny funkce, třídy a proměnné z tohoto modulu, což je jednodušší v praxi i pro začátečníka. Pokud uživatel chce z modulu načíst pouze jednu položku, tak stačí hvězdičku nahradit danou položkou, například v tomto případě `display`. Funkce `display.scroll` na třetím řádku byla importována právě díky prvnímu řádku a umožňuje zobrazit text na displeji micro:bit. [33]

```
1 from microbit import *
2
3 display.scroll('Hello World')
```

Obrázek 7: Hello world v MicroPythonu. Zdroj: Vlastní

3.2.4 Použité moduly v úlohách

Modul je klasický python soubor, takže stačí zkopírovat kód z přílohy daného modulu do python souboru, který poté přesuneme na micro:bit pomocí ikony File v editoru MU. V programu pak daný modul lze použít pomocí klíčového slova `import`, za kterým bude následovat jméno daného modulu. V úlohách jsou celkem tři speciální moduly, které jsem použil v praktických úlohách. Tyto moduly se nacházejí v této příloze bakalářské práce.

Modul `waveshareJoystick` ovládá gamepad od firmy Waveshare. Na stránkách výrobce je demo kód, ze kterého jsem daný modul převzal. [19] V modulu je třída `JOYSTICK`, ta obsahuje metodu `Listen_key` pro detekci stisknutého tlačítka na gamepadu.

Dalším modulem je `moveKitronik`, který slouží k pohybu robota `movemini`. Obsahuje třídu `servo` a `MK2`. Pro řízení serva slouží třída `servo`, kterou jsem převzal z githubu [20], obsahuje metodu `write_angle`, která je ekvivalentní k bloku `servo write pin` v MakeCode. Třídu `MK2` slouží pro ovládání směru robota, která je podobná jako rozšíření `Servo:lite` v Makecode. Obsahuje metody `driveBackward`, `driveForward`, `stop`, `leftTurn` a `rightTurn`. Pro otočení robota doleva nebo doprava o daný počet stupňů řídí metody `turnLeftDegrees` a `turnRightDegrees`.

Posledním modulem je `ultrasonicSensor` pro detekování vzdálenosti objektu před ultrazvukovým senzorem. Ze stránek výrobce senzoru jsem upravil kód, který byl původně pro Raspberry Pi. [21] Tento modul obsahuje třídu `GroveUltrasonicRanger`, který má metodu `get_distance` pro zjištění vzdálenosti od ultrazvukového měřiče.

4 KOMPONENTY POUŽITÉ V ÚLOHÁCH

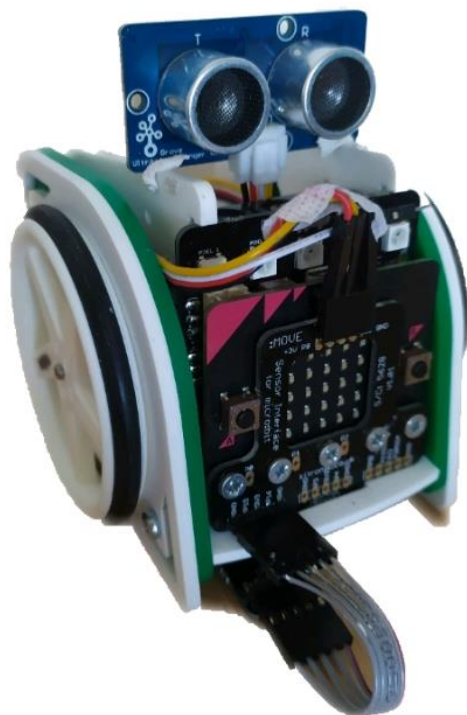
Věci popsané v této kapitole jsem použil v úlohách bakalářské práce, avšak nejsou součástí micro:bit, je tedy nutné je dokoupit. Všechny tyto komponenty jsou vhodné použít výhradně s počítačem micro:bit.

4.1 Kitronik :MOVE mini MK2

Někdy též movemini je dvoukolový robot, který ve spojení s počítačem micro:bit slouží pro zábavný úvod do robotiky. Movemini je vhodný pro autonomní provoz i pro projekty bezdrátového dálkového ovládání pomocí aplikace na mobilním zařízení, nebo s použitím druhého micro:bit, který využívá technologii Bluetooth nebo radio. Pohyb robota umožňují dva servomotory s kontinuální rotací, jejichž rychlost se dá ovládat změnou signálu pulzně šířkové modulace (PWM). Ta lze nastavit v editoru MakeCodu díky bloku Servo Write Pin. Kitronik vytvořil i vlastní bloky pro rychlé a jednoduché řízení robota. Součástí movemini je i pět led diod, které mohou plnit funkci indikátorů či couvacích světel. V konstrukci robota je mezera na fixu, pomocí které lze malovat tvar podle toho, kam movemini jede. Nejdůležitější součástí robota je deska Servo:lite, do které se namontuje pomocí pěti šroubů počítač micro:bit. Deska umožňuje připojit až tři serva, jež pracují při napětí 3,3 voltů. Napájení desky obstarává slot pro tři baterie AAA. Pro zapnutí či vypnutí desky slouží tlačítko vypínače. [26]

Před sestavením robota je potřeba zkalibrovat serva pomocí trimu na jejich zadní straně. K zprovoznění třetího serva je třeba na zadní straně desky servo:lite přeříznout cestu mezi dvěma žlutými čtverečky a spájet cestu k třetímu servu podle návodu. Tato akce zprovozní funkci třetího serva, ale vyřadí LED diody. Pro obnovení funkce LED diod je potřeba spájet cestu zpět mezi čtverečky. [26]

Nevýhodou robota je z mé zkušenosti to, že nedokáže jet přesně rovně, kvůli levným servo motorům. Další slabinou je, že se úlohy nedokážou snadno stejně zopakovat, protože výkon robota záleží na nabití baterií a jeho zatížení. Tato vlastnost například způsobuje, že se robot po vykonání bloku turnLeftDegrees zatočí různě podle stavu baterií.



Obrázek 8: Robot Kitronik :MOVE mini MK2. Zdroj: Vlastní

4.1.1 Kitronik :MOVE Sensor Interface Board

Nezbytnou součástí počítače pro přidání dalších senzorů jako například ultrazvukový měřič vzdálenosti nebo sledování čáry je deska :MOVE Sensor Interface Board. Deska se přišroubuje k počítači micro:bit a Servo: Lite pomocí pěti šroubů, napájena je ze Servo:Lite. [28]

4.1.2 Kitronik :MOVE Line Following board

Pro sledování černé čáry na bílém pozadí či pro bílé čáry na pozadí černém je vhodné použít Line Following desku. Na desce jsou dva potenciometry, jeden pro každý senzor. Po detekování se na desce rozsvítí LED indikátory. Pokud má deska detekovat tmavší povrch, je třeba potenciometr otočit po směru hodinových ručiček a pro světlejší povrch otočit proti směru hodinových ručiček. Tato deska se připojuje k desce :MOVE Sensor Interface. Nejlepšího výsledku se dosáhne, pokud vzdálenost mezi spodní částí desky a plochou po které robot jede je přibližně 5 mm. [27] Pro detekování černé je potřeba na podkladě mít sytě černou barvu. Jízda po čáře může být nepřesná z důvodu konstrukce robota, jehož přední část, kde se nachází i tato deska se při rozjezdu nadzvedne. Tento problém lze vyřešit zatížením přední části robota pomocí nějakého předmětu.

4.2 Grove Inventor Kit for micro:bit

Tato sada rozšiřuje možnosti micro:bit, obsahuje celkem osm modulů včetně senzorů, akčního členu a displeje. S moduly lze komunikovat jednoduše pomocí desky Grove Shield, do které se zasune micro:bit, takže není potřeba pájet ani zapojovat pomocí propojovacích vodiče. Pro MakeCode existuje rozšíření Grove, kde jsou bloky komunikující s touto sadou. V úlohách jsem z této sady využil ultrazvukový měřič vzdálenosti, červenou LED diodu a reproduktor. [37]

4.2.1 Ultrazvukový měřič vzdálenosti

K desce :MOVE Sensor Interface je oficiálně potřeba speciální ultrazvukový měřič vzdálenosti s napětím 3,3 voltu. Takový senzor s tímto napětím je ale těžké na českém trhu sehnat. Proto jsem jako alternativu použil Grove senzor, který též potřebuje napětí 3,3 voltů. Avšak je potřeba ho k desce :MOVE Sensor Interface Board připojit pomocí propojovacího kabelu s konektory Grove a DuPont. Inspiroval jsem se tak na blogu zonepi [34], kde byl použit takto použit. Měřič vzdálenost od firmy Grove má jeden pin typu NC, což znamená, že neplní funkci spoje, to umožňuje nechat na desce volný pin 10 a zachovat funkci displeje. Grove senzor pracuje na 40kHz a funguje tak, že vysílá 8 úrovní cyklů a detekuje ozvěnu. Vzdálenost se poté vypočítá podle vzorce, ve kterém je zahrnut čas ozvěny a rychlost zvuku. V MakeCodu lze získat vzdálenost ze senzoru pomocí rozšíření Grove, kde jsou bloky, které ji umožňují detekovat v centimetrech. [21]

Tento senzor jsem k robotu movemini připevnil pomocí plastické lepící hmoty, nechtěl jsem jej připevnit úplně napevno, protože jsem chtěl měnit místo senzoru na robotu. Například kvůli úloze 5.4, kde robot jezdí podél zdi.

4.3 Waveshare Joystick pro micro:bit

Je to rozšiřující modul gamepadu, do kterého se zasune micro:bit. Obsahuje integrovaný joystick a tlačítka. Mezi další funkce patří napájení energie z baterie a přehrávání hudby. Pomocí gamepadu lze vytvořit jednoduché arkádové hry, které mohou být zobrazovány na displeji micro:bit nebo na externím displeji. Na stránkách výrobce je odkaz na github pro rozšíření v MakeCodu, které ulehčuje programování gamepadu.[19]

5 ÚLOHY

Jsou koncipovány jako jednoduché úlohy pro začátečníky v programování a algoritmizaci. Úlohy dále demonstrují možnosti a funkce robota Kitronik :MOVE mini MK2 a počítače BBC micro:bit. Každá úloha obsahuje popis, potřebné komponenty k úloze, a nakonec řešení dané úlohy v Microsoft MakeCode a MicroPythonu. V MicroPythonu jsou použity speciální moduly, které se musí vložit do micro:bit, ty jsou popsány v podkapitole 3.2.4. Implementace modulů se nachází v příloze bakalářské práce. Pro přesné opakování úloh je třeba mít plně nabitě baterie v robotu movemini, takže některé parametry v úlohách je potřeba změnit a přizpůsobit, podle stavu robota.

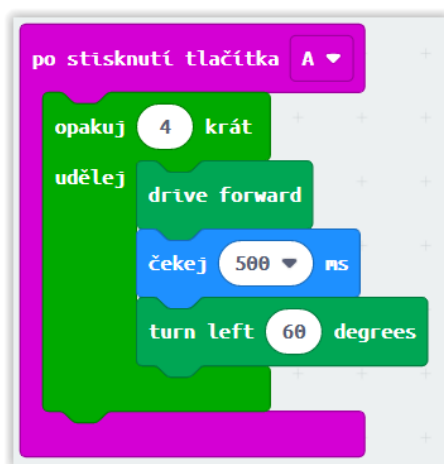
5.1 Nakreslení čtverce

Cílem je nakreslit čtverec s využitím for cyklu. Uvnitř robota movemini je otvor pro vložení například fixy, pomocí níž můžeme napsat obraz na papír, podle toho, jakým směrem se robot pohybuje. Úloha se spustí stisknutím tlačítka na počítači micro:bit. Čtverec nebude úplně přesný z důvodu nepřesnosti samotného robota.

K úloze je potřeba: BBC micro:bit, Kitronik :MOVE mini MK2 a fix.

MakeCode

Pro zjednodušení jsem použil rozšíření Servo:Lite, které je nutné přidat do MakeCode. Ulehčuje řízení robotů od firmy Kitronik. Alternativně lze místo Servo:Lite použít blok Servo Write Pin. Zpoždění určuje, jak dlouho pojede robot rovně, tedy délku strany čtverce. V bloku turn left by správně měla být zapsána hodnota 90, značící pravý úhel, ale kvůli nepřesnosti robota je přesné zatočení doleva po vložení nových baterií na hodnotě 60.



Obrázek 9: Úloha 1. MakeCode. Zdroj: Vlastní

MicroPython

While cyklus dokola kontroluje, jestli nedošlo ke stisknutí tlačítka. Modul moveKitronik slouží pro pohyb robota movemini. Modul microbit je popsán v podkapitole 3.2.3. Objekt robot představuje robota movemini.

```
from microbit import *  
import moveKitronik  
  
robot = moveKitronik.KitronikMK2()  
  
while True:  
    if button_a.is_pressed():  
        for i in range(4):  
            robot.driveForward()  
            sleep(500)  
            robot.turnLeftDegrees(60)
```

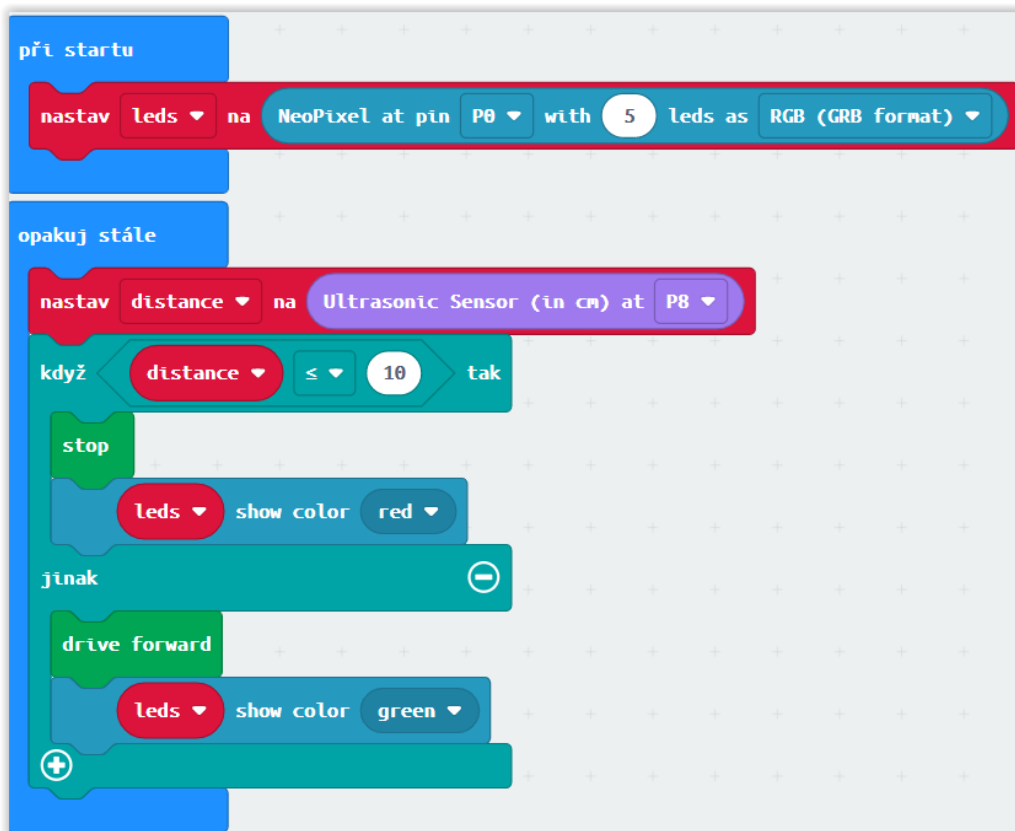
5.2 Zastavení robota před překážkou

Robot movemini pojedje rovně a zastaví deset centimetrů před překážkou, kterou detekuje díky ultrazvukovému senzoru. Jako bonus jsou ještě v úloze využity LED diody na přední straně robota. Pokud se robot pohybuje, diody se zbarví do zelena, v opačném případě se pak rozsvítí červeně. Ultrazvukový senzor je k desce Sensor Interface připojen přes pin 8.

K úloze je potřeba: Grove ultrazvukový senzor, Kitronik :MOVE mini MK2, BBC micro:bit, propojka pro senzor vzdálenosti, Move Sensor Interface.

MakeCode

V mém řešení jsem použil rozšíření Grove pro ovládání senzoru, Servo:Lite a Neopixel. V bloku „při startu“ se nastaví pět LED diod na pinu 0 do proměnné leds. V bloku „opakuj stále“ se do proměnné distance uloží vzdálenost objektů před movemini, na základě čehož se poté vyhodnotí, jestli robot zastaví nebo pojedje dál.



Obrázek 10: Úloha 2. Řešení v MakeCode. Zdroj: Vlastní

MicroPython

V MicroPythonu nelze tak snadno nastavit barvu diod, proto jsem vytvořil funkci `setColor`, která pomocí for cyklu změní barvu všech pět LED diod. Modul `ultrasonicSensor` je v příloze bakalářské práci, která slouží pro měření vzdálenosti. Modulem `neopixel` se ovládají LED diody. V třídě `NeoPixel` v prvním parametru definuji, který pin ovládá LED diody. V druhém parametru určím počet diod, kterých je pět. Proměnné `red` a `green` definují barvy. Objekt `senzor` představuje ultrasonický měřič vzdálenosti.

```

from microbit import *
import ultrasonicSensor, moveKitronik, neopixel

def setColor(color):
    for i in range(5):
        leds[i]=color
        leds.show()

leds = neopixel.NeoPixel(pin0,5)
senzor = ultrasonicSensor.GroveUltrasonicRanger(pin8)
robot = moveKitronik.KitronikMK2()
green = (0, 128, 0)
red = (255, 0, 0)

```

```

while True:
    distanceValue = sensor.get_distance()
    if distanceValue <= 10:
        robot.stop()
        setColor(red)
    else:
        setColor(green)
        robot.driveForward()

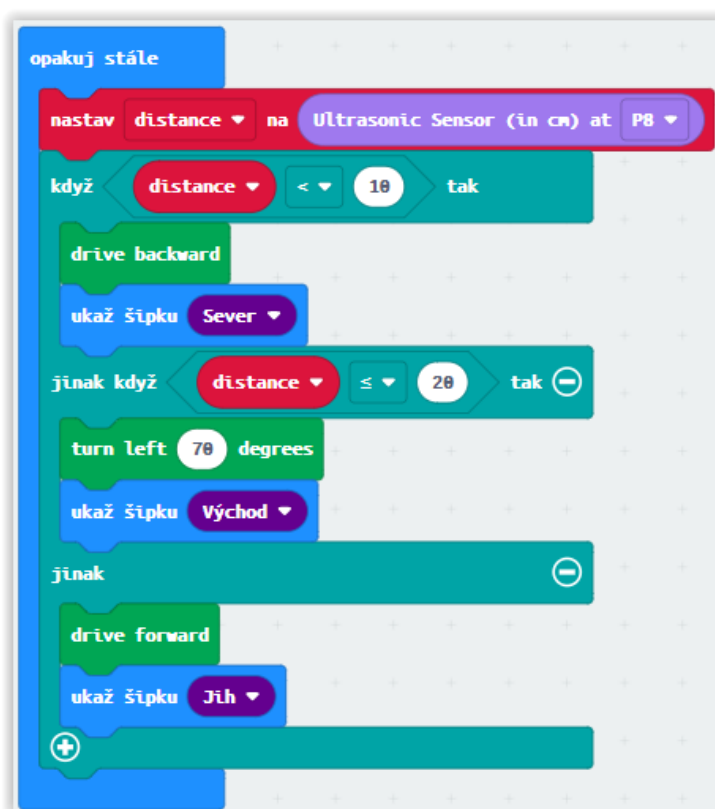
```

5.3 Vyhýbání se překážkám

Robot movemini se bude stále pohybovat směrem dopředu, pokud však před sebou detekuje překážku pomocí ultrazvukového senzoru, tak zatočí doleva. V mém zapojení robota funguje ultrazvukový senzor i displej micro:bitu zároveň, což budu demonstrovat na této úloze. Na displeji se zobrazí šipka ukazující směr pohybu.

K úloze je potřeba: Groove ultrazvukový senzor, propojka pro ultrazvukový senzor, Kitronik :MOVE mini MK2, BBC micro:bit a Move Sensor Interface.

MakeCode



Obrázek 11: Úlohu 3. Řešení v Makecode. Zdroj: Vlastní

Robot pojedě dopředu, avšak zaznamená-li však objekt, který se nachází méně jak deset centimetrů před ním, tak pojedě dozadu. Pokud objekt bude mezi deseti a dvaceti centimetry, tak se otočí doleva o devadesát stupňů. Z důvodu nepřesnosti robota je nastaveno na 70 stupňů.

MicroPython

Třída Image již obsahuje řadu obrázků pro zobrazení na displej micro:bit včetně šipek.

```
from microbit import *
import ultrasonicSensor
import moveKitronik

senzor = ultrasonicSensor.GroveUltrasonicRanger(pin8)
robot = moveKitronik.KitronikMK2()

while True:
    distanceValue = senzor.get_distance()
    if distanceValue<=10:
        robot.driveBackward();
        display.show(Image.ARROW_N)
    elif distanceValue<=20:
        robot.turnLeftDegrees(90)
        display.show(Image.ARROW_E)
    else:
        robot.driveForward()
        display.show(Image.ARROW_S)
```

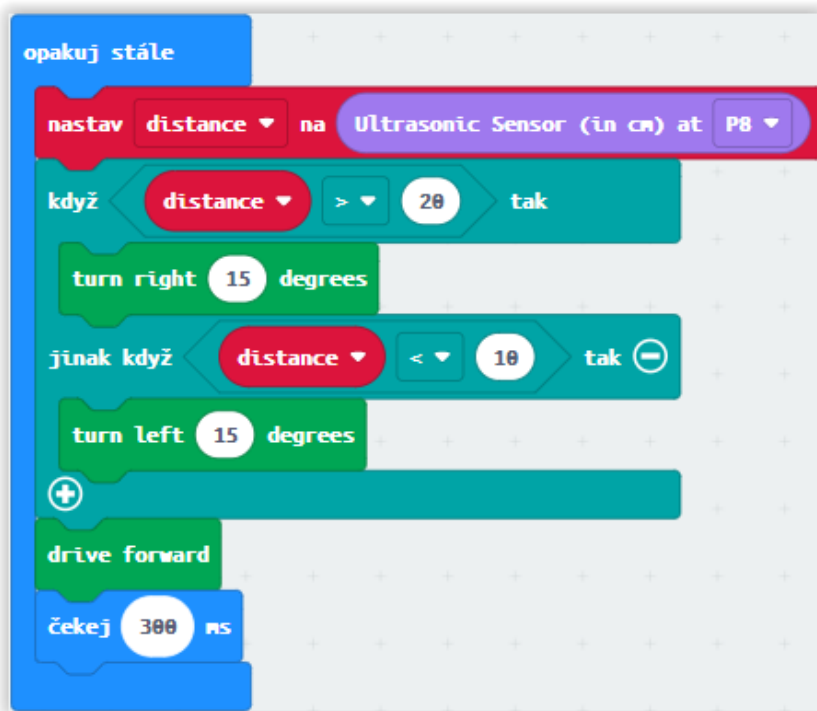
5.4 Jízda podél zdi

V úloze robot movemini pojedě podél zdi a bude se držet deset až dvacet centimetrů od zdi pomocí ultrazvukového senzoru Grove. V mém řešení lze senzorem lze senzorem volně pohybovat, připevnil jsem ho na bok robota pro snímání zdi.

K úloze je potřeba: Grove ultrazvukový senzor, propojka pro ultrazvukový senzor, Kitronik :MOVE mini MK2, BBC micro:bit a Move Sensor Interface.

MakeCode

Movemini jede stále rovně. Pokud se díky ultrazvukového senzoru zjistí, že vyjel ze svého směru, tak movemini zahne cca o 15 stupňů zpátky. U nových baterií stačí 15 stupňů, avšak pokud jsou již baterie použité, tak bude potřeba stupně navýšit. Každých 300 milisekund se bude kontrolovat, jestli nevyjel ze svého směru.



Obrázek 12: Úloha 4. Řešení v MakeCode. Zdroj: Vlastní

MicroPython

```

from microbit import *
import ultrasonicSensor, moveKitronik

sensor = ultrasonicSensor.GroveUltrasonicRanger(pin8)
robot = moveKitronik.KitronikMK2()

while True:
    distanceValue = sensor.get_distance()
    if distanceValue > 20:
        robot.turnRightDegrees(15)
    elif distanceValue < 10:
        robot.turnLeftDegrees(15)

    robot.driveForward()
    sleep(300)

```

5.5 Parkovací senzor

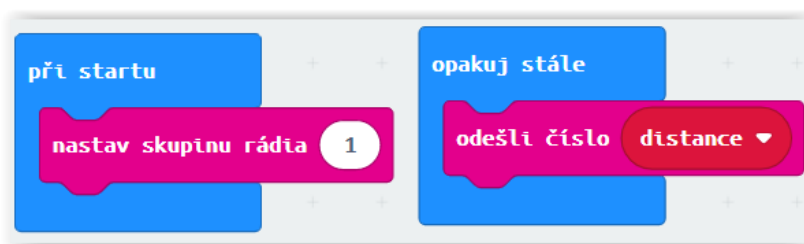
Tato úloha je rozšířením úloh 5.2 a 5.3. Čím blíže se robot bude přibližovat k překážce, tím rychleji bude reproduktor na druhém micro:bit pípat. Jak jsem již popsal v teoretické části, tak BBC micro:bit V1 nemá reproduktor, tento problém řeším druhým počítačem micro:bit, ke kterému je připojen reproduktor pomocí desky Grove Shield. Počítače spolu komunikují

bezdrátově pomocí radia, tento modul je více popsán v podkapitole 2.4.5. V řešení bude pouze rozšíření, které lze přidat k předchozím úlohám.

K úloze je potřeba: Grove ultrazvukový senzor, propojka pro ultrazvukový senzor, Kitronik :MOVE mini MK2, 2x BBC micro:bit, Move Sensor Interface, Grove Shield a Grove speaker.

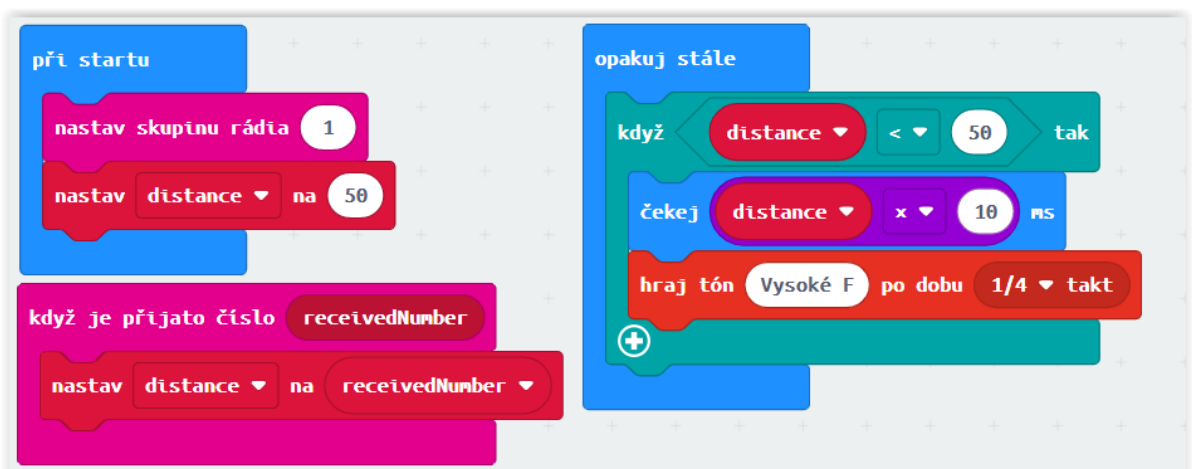
MakeCode

V micro:bit u robota je třeba nejdříve nastavit stejnou skupinu pro komunikaci radia mezi počítači. Dále do bloku „opakuje stále“ jsem přidal blok „odešli číslo“, který bude posílat vzdálenost z ultrazvukového senzoru na druhý micro:bit.



Obrázek 13: Úloha 5. Řešení v MakeCode pro robota. Zdroj: Vlastní

Růžový blok „když je přijato“ se vykoná vždy, když se přes radio přijme číslo. V tomto bloku tak aktualizují hodnotu vzdálenosti. V bloku „opakuje stále“ se již řeší pravidelné pípání, které se vykoná pouze je-li vzdálenost menší než 50 centimetrů. Pro zvuk parkovacího senzoru jsem využil tón F z kategorie Hudba. Vzdálenost je při startu nastavena na 50, aby při startu reproduktor nezačal pípat.



Obrázek 14: Úloha 5. Řešení v MakeCode pro reproduktor. Zdroj: Vlastní

MicroPython

V MicroPythonu se nemusí nastavovat skupina pro komunikaci radia. Radio se aktivuje příkazem `radio.on`. Posílání a přijímání obstarávají příkazy `send` a `receive`. V MicroPythonu existuje modul `music`, který obsahuje tóny. Metoda `set_tempo` určuje rytmus přehrávání.

Kód pro micro:bit s reproduktorem

```
from microbit import *
import radio, music
music.set_tempo(ticks=16)
tune = "F"
radio.on()
distance=50

while True:
    incoming = radio.receive()
    if incoming is not None:
        distance=float(incoming)
        if distance < 50:
            delay=10*distance
            sleep(delay)
            music.play(tune)
```

Radio přijímá a posílá řetězec, takže je nutné přetypovat číslo na řetězec pomocí `str`.

Kód pro micro:bit uvnitř robota

```
from microbit import *
import ultrasonicSensor, moveKitronik, radio

senzor = ultrasonicSensor.GroveUltrasonicRanger(pin8)
radio.on()

while True:
    distanceValue = senzor.get_distance()
    radio.send(str(distanceValue))
```

5.6 Grafické zobrazení vzdálenosti

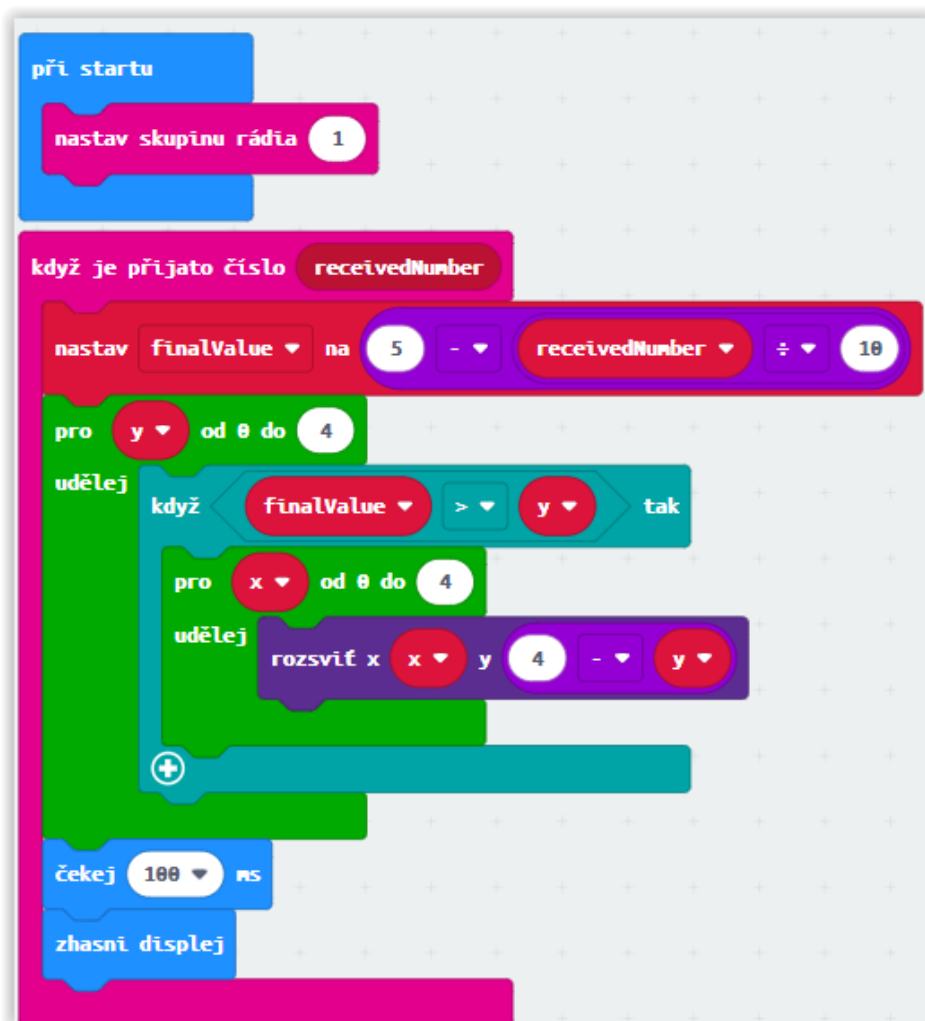
Cílem je grafem zobrazit vzdálenost robota od překážky na displeji `micro:bit`. Tato úloha navazuje na předešlou, ale místo zvukového doprovodu se zobrazuje graficky vzdálenost

překážky na displeji druhého počítači. Pro komunikaci se opět využívá radio. S displejem na micro:bit se musí pracovat jako s maticí.

K úloze je potřeba: Groove ultrazvukový senzor, propojka pro ultrazvukový senzor, Kitronik :MOVE mini MK2, 2x BBC micro:bit, Move Sensor Interface, Grove Shield a Grove speaker.

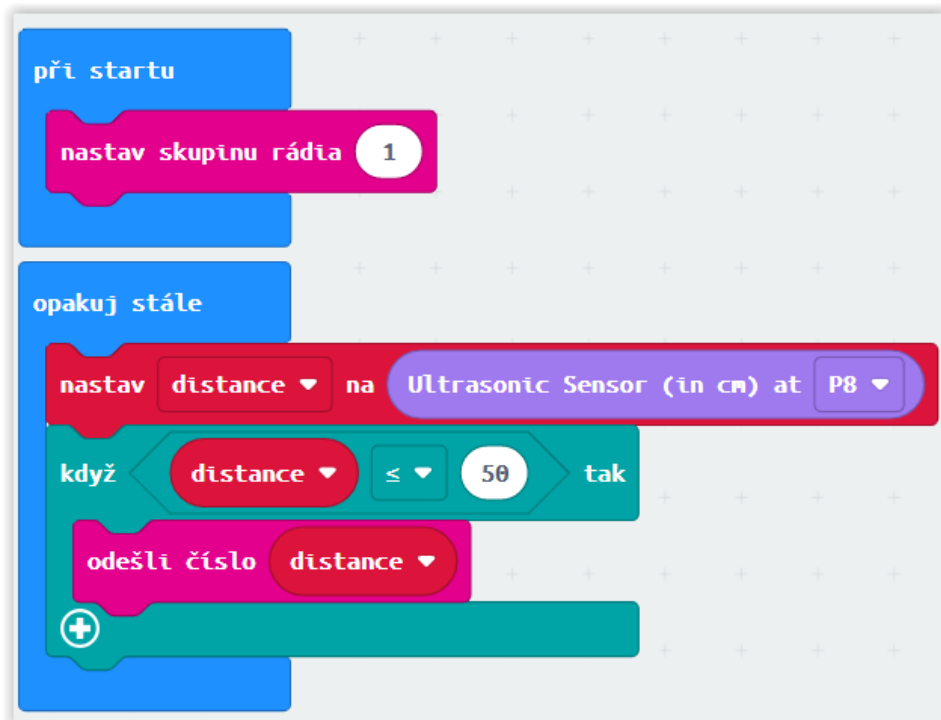
MakeCode

Přijatá hodnota vzdálenosti z druhého počítače je třeba upravit tak, aby se dala použít na displej. Poté se na displeji zobrazí graf na 300 milisekund a displej zhasne, aby se mohl aktualizovat na novější hodnotu vzdálenosti.



Obrázek 15: Úloha 6. Řešení v MakeCode pro zobrazení grafu. Zdroj: Vlastní

Tyto bloky je třeba přidat k předešlým úlohám do micro:bit U radia je třeba nastavit stejnou skupinu. Vzdálenost je odeslána, když je menší než padesát centimetrů.



Obrázek 16: Úloha 6. Řešení v MakeCode pro movemini. Zdroj: Vlastní

MicroPython

Kód pro micro:bit uvnitř robota

```
from microbit import *  
import ultrasonicSensor, moveKitronik, radio  
  
sonar = ultrasonicSensor.GroveUltrasonicRanger(pin8)  
radio.on()  
  
while True:  
    distanceValue = sonar.get_distance()  
    if distanceValue <= 50:  
        radio.send(str(distanceValue))
```

Obsahuje metodu showDistance pro zobrazení grafu na displeji. Nejdříve se kontroluje, jestli přišla zpráva z radia. Pokud nepřišla, celý displej zhasne. Avšak jestli zpráva přišla, tak se řetězec musí převést na číslo a vložit do parametru funkce.

Kód pro zobrazení grafu

```
from microbit import *
import radio

def showDistance(distance):
    display.clear()
    for y in range(0,5):
        if distance > y:
            for x in range(0,5):
                display.set_pixel(x,4-y,9)
radio.on()
while True:
    incoming = radio.receive()
    if incoming is not None:
        finalValue = 5 - int(float(incoming))/10
        showDistance(finalValue)
    else:
        display.clear()
```

5.7 Ovládání robota pomocí gamepadu

Cílem je ovládat robota Kitronik movemini pomocí gamepadu od firmy Waveshare. V této úloze jsou použity dva počítače micro:bit, které mezi sebou komunikují pomocí radia. Jeden počítač bude zapojen do gamepadu a druhý bude ovládat robot movemini. Na gamepadu si lze naprogramovat joystick nebo různá tlačítka.

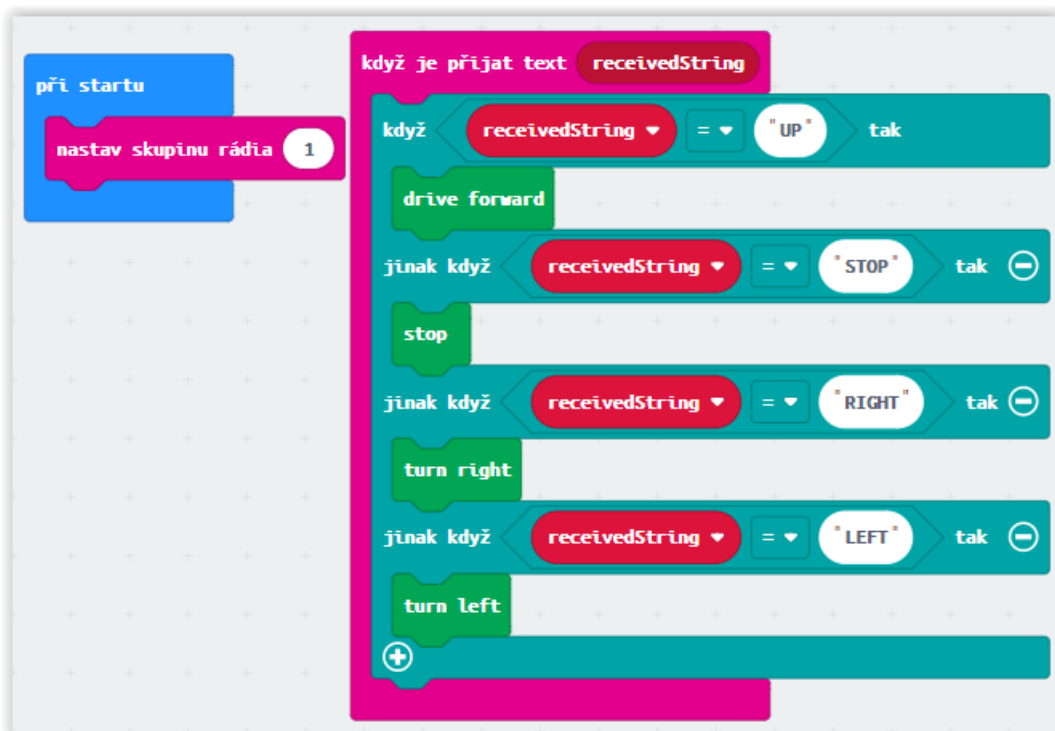
K úloze je potřeba: 2x BBC micro:bit, Waveshare Joystick a Kitronik :MOVE mini MK2.

MakeCode

Pro ovládání joysticku je potřeba v MakeCodu přidat rozšíření pomocí URL: github.com/waveshare/JoyStick. [19] V bloku „při startu“ se nejdříve JoyStickInit nastaví gamepad pro komunikaci s micro:bit a pro komunikaci radia se mezi počítači musí zvolit stejná skupina. Poté se v bloku keyPress detekuje směr stisknutím joysticku a pomocí radia pošle řetězec obsahující směr pohybu na druhý micro:bit, který přijme signál a vykoná danou akci.



Obrázek 17: Úloha 7. Řešení v MakeCode pro gamepad. Zdroj: Vlastní



Obrázek 18: Úloha 7. Řešení v MakeCode pro robota. Zdroj: Vlastní

MicroPython

K této úloze jsou potřeba moduly `moveKitronik` a `waveshareJoystick`, který je nutné importovat pro ovládání gamepadu. Tento modul se nachází v příloze bakalářské práce a více ho popisují v podkapitole 3.2.4. Objekt `JoyStick` představuje gamepad. Proměnnou `KEY` z modulu `waveshareJoystick` definujeme, jaké tlačítko nebo směr joysticku chceme použít.

Kód pro micro:bit v gamepadu

```
from microbit import *
import radio, waveshareJoystick

JoyStick = waveshareJoystick.JOYSTICK()
KEY = waveshareJoystick.KEY
radio.on()

while True:
    while JoyStick.Listen_Key(KEY['E']):
        radio.send('FORWARD')
    while JoyStick.Listen_Key(KEY['C']):
        radio.send('BACKWARD')
    while JoyStick.Listen_Key(KEY['F']):
        radio.send('LEFT')
    while JoyStick.Listen_Key(KEY['D']):
        radio.send('RIGHT')
    while JoyStick.Listen_Key(KEY['P']):
        radio.send('STOP')
```

Kód pro robot

```
from microbit import *
import radio
import moveKitronik

robot = moveKitronik.KitronikMK2()
radio.on()

while True:
    incoming = radio.receive()
    if incoming == 'FORWARD':
        robot.driveForward()
    elif incoming == 'BACKWARD':
        robot.driveBackward()
    elif incoming == 'STOP':
        robot.stop()
    elif incoming == 'LEFT':
        robot.turnLeft()
    elif incoming == 'RIGHT':
        robot.turnRight()
```


5.8 Sledování čáry řízené semaforem

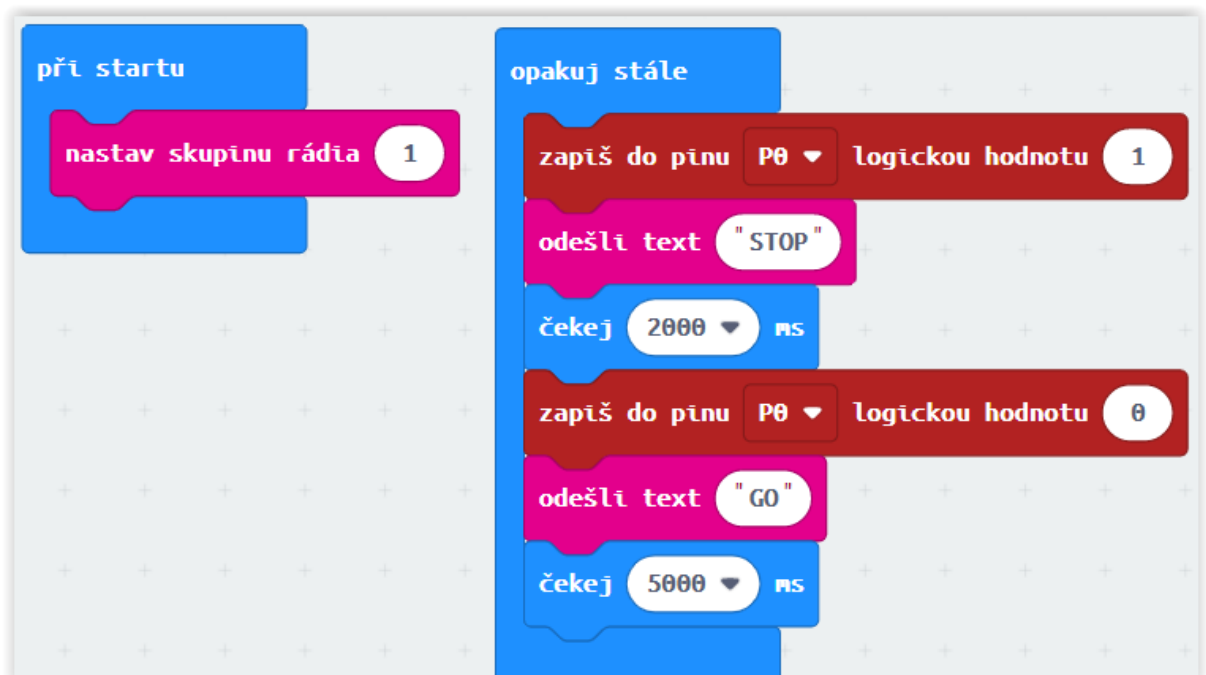
V této úloze budou použity dva počítače micro:bit, které spolu budou komunikovat pomocí radia. Jeden počítač bude řídit semafor, což bude představovat červená LED dioda a druhý bude ovládat robot movemini, který bude jezdit dokola po čáře. Semafor v pravidelných intervalech bude zapínat červenou diodu, čímž se robot zastaví. Když dioda zhasne, tak se robot opět rozjede.

Robot se bude pohybovat dokola po čáře pomocí desky Kitronik Line following, na které je senzor. K detektorům na desce Kitrolink Line Following je micro:bit připojen pomocí pinu 15 a 16. U pinů je třeba nastavit ráz pinu nahoru tzv. pull ups.

K úloze je potřeba: 2x BBC micro:bit, Kitronik Move mini MK2, Grove Red LED, Line following board a Grove shield.

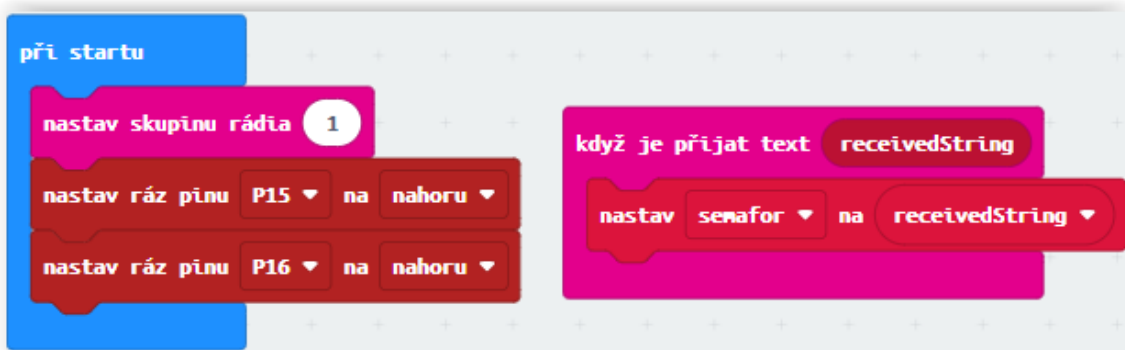
MakeCode

Dioda se zapíná pomocí zapsání logické hodnoty.

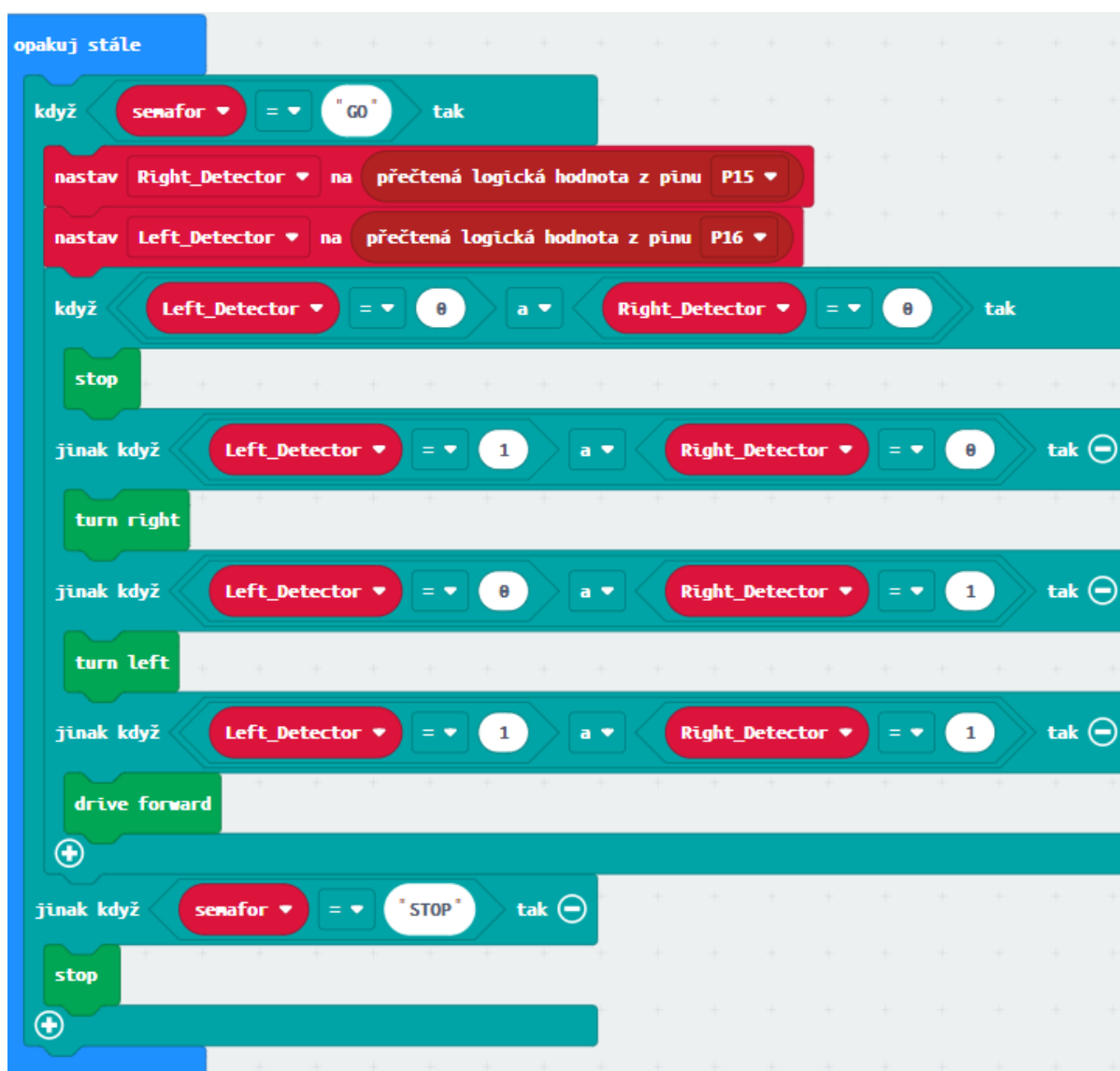


Obrázek 19: Úlohu 8. Řešení v MakeCode pro semafor. Zdroj: Vlastní

Proměnné semafor určuje, jestli má robot jet nebo stát. Proměnná se aktualizuje v radio bloku „když je přijato”. První příkaz if zkontroluje, jestli je semafor zapnutý. Další vnořený if už je pro samotnou jízdu po čáře. Pro detektory na Line Following board slouží proměnné Right_Detector a Left_Detector.



Obrázek 20: Úlohu 8. Řešení v MakeCode pro movemini. Část 1. Zdroj: Vlastní



Obrázek 21: Úlohu 8. Řešení v MakeCode pro movemini. Část 2. Zdroj: Vlastní

MicroPython

Kód pro semafor

```
from microbit import *
import radio

radio.on()
while True:
    pin0.write_digital(1)
    radio.send("STOP")
    sleep(2000)
    pin0.write_digital(0)
    radio.send("GO")
    sleep(5000)
```

Kód pro robota

```
from microbit import *
import radio,moveKitronik

pin15.set_pull(pin15.PULL_UP)
pin16.set_pull(pin15.PULL_UP)
robot = moveKitronik.KitronikMK2()
radio.on()
semafor=None
while True:
    incoming = radio.receive()
    if incoming is not None:
        semafor =incoming
    if semafor == "GO":
        rightDetector =pin15.read_digital();
        leftDetector =pin16.read_digital();

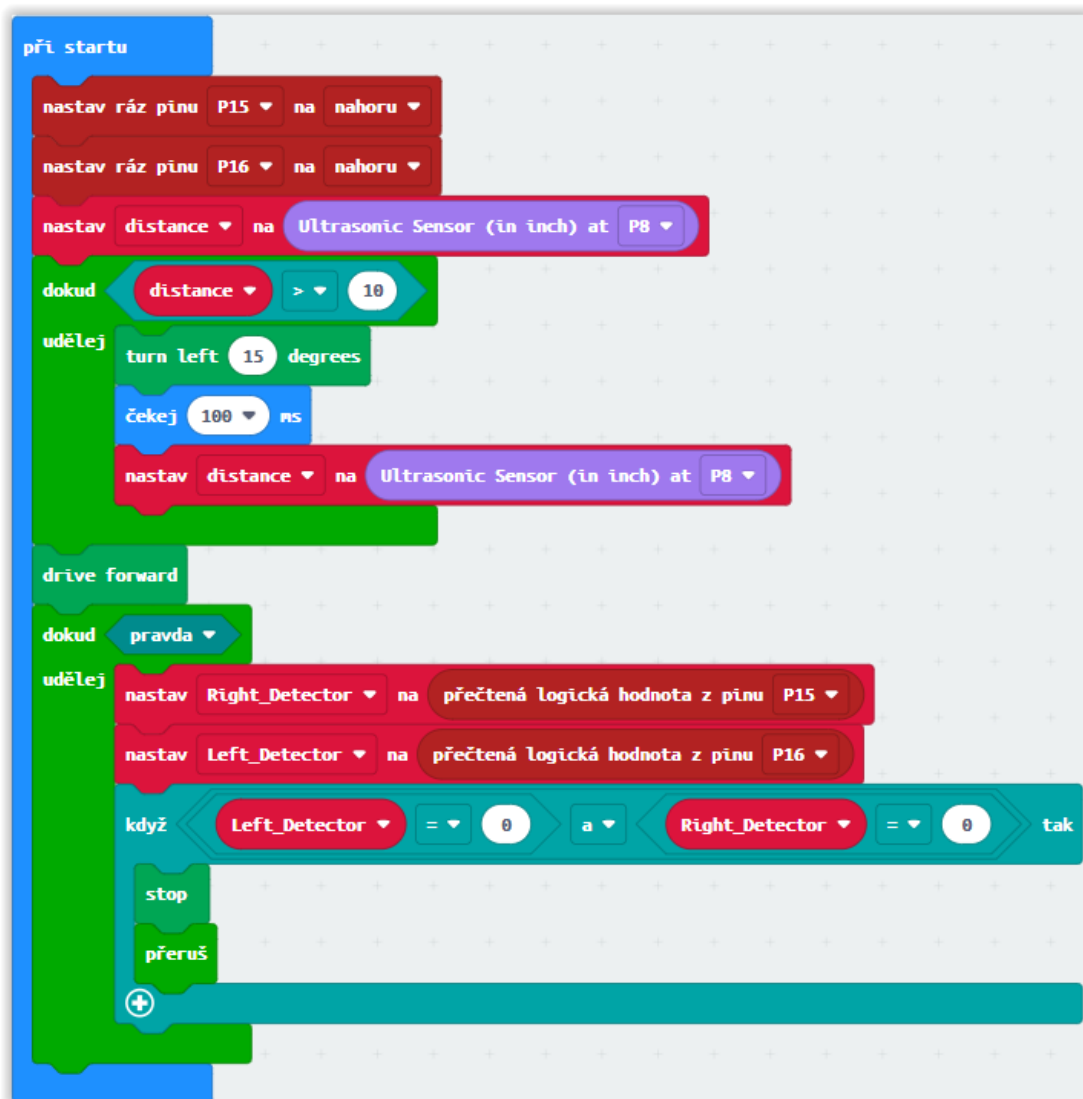
        if leftDetector == 0 and rightDetector == 0:
            robot.stop()
        if leftDetector == 1 and rightDetector == 0:
            robot.turnRight()
        if leftDetector == 0 and rightDetector == 1:
            robot.turnLeft()
        if leftDetector == 1 and rightDetector == 1:
            robot.driveForward()
    elif semafor == "STOP":
        robot.stop()
```

5.9 Sumo zápas

K úloze je potřeba mít vhodný podklad, tedy bílý kruh s černým okrajem. Na začátku vložíme doprostřed kruhu robota a například papírovou krabičku představující protivníka. Robot si otáčením dokolečka sám najde daného soupeře pomocí ultrazvukového senzoru a vytlačí ho mimo kruh, ale sám zůstane uvnitř kruhu díky senzoru, který rozpozná černý okraj z desky Line Following.

K úloze je potřeba: BBC micro:bit, Kitronik Move mini MK2, ultrazvukový měřič vzdálenosti, Line following board a poklad(ring).

MakeCode



Obrázek 22: Úloha 9. Řešení v MakeCode. Zdroj: Vlastní

V úloze jsou použity dva while cykly. V prvním cyklu se po 100 milisekundách otáčí o 15 stupňů. Tento cyklus skončí, když najde protivníka na vzdálenost menší deset centimetrů. Druhý while cyklus kontroluje, jestli robot nenajel na černý okraj, pokud ano, tak se zastaví a přeruší cyklus. V druhém while cyklu je blok přeruš, který funguje jako break, tedy cyklus ukončí cyklus.

MicroPython

```
from microbit import *
import radio,ultrasonicSenzor,moveKitronik

pin15.set_pull(pin15.PULL_UP)
pin16.set_pull(pin15.PULL_UP)
robot = moveKitronik.KitronikMK2()
senzor = ultrasonicSenzor.GroveUltrasonicRanger(pin8)
distance= senzor.get_distance()

while distance>10:
    robot.turnLeftDegrees(20)
    sleep(100)
    distance = senzor.get_distance()

robot.driveForward()

while True:
    rightDetector =pin15.read_digital()
    leftDetector =pin16.read_digital()
    if leftDetector == 0 and rightDetector == 0:
        robot.stop()
        break
```

5.10 Přesouvání nákladu

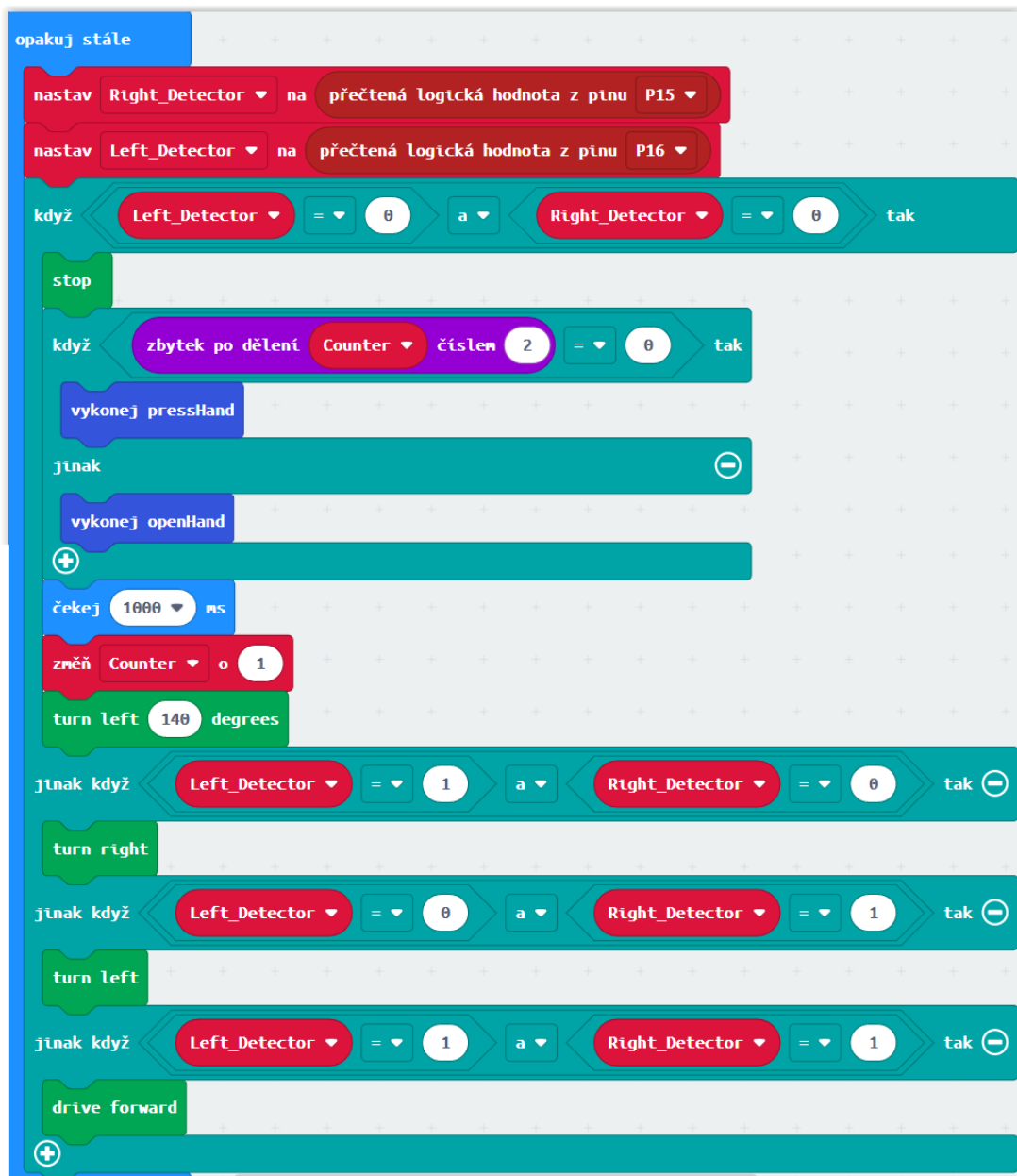
Cílem úlohy je přesouvat předmět po černé čáře z jednoho bodu do druhého pomocí robota movemini a mechanické ruky. V robotu movemini lze použít i třetí servo, což budu demonstrovat v této úloze, kde jej použiji pro ovládání robotické ruky. Firma Kitronik takovou ruku pro robota movemini nabízí, ale na českém trhu není dostupná. Je však možnost si podobně fungující ruku vytisknout na 3D tiskárně a poté ji připevnit na přední část robota. Důležité je si udělat vhodnou dráhu, já si udělal dva černý čtverce spojené černou čarou na bílém podkladu. Na jeden čtverec položím předmět, který robot bude přesouvat. Robot se pohybuje po černé čáře, když detekuje na obou senzorech desky :MOVE Line folowing černou barvu, znamená to, že je na daném bodě. Poté stiskne ruku nebo uvolní ruku podle toho na

jakém bodě je a otočí se zpátky. Robotická ruka se v robotu movemini ovládá pinem 0. V MicroPythonu ovládám ruku pomocí třídy Servo z modulu moveKitronik.

K úloze je potřeba: BBC micro:bit, Kitronik Move mini MK2, ultrazvukový měřič vzdálenosti, Line following board, robotická ruka a podklad.

MakeCode

Proměnná Counter počítá počet zastavení. Každý sudý zastavení vezme robot náklad a každý lichý jej vyloží. Pro přehlednost jsem udělal dvě funkce pro ovládání ruky robota



Obrázek 23:Úloha 10. Řešení v MakeCode. Část 1. Zdroj: Vlastní



Obrázek 24: Úloha 10. Řešení v MakeCode. Část 2. Zdroj: Vlastní

MicroPython

```

from microbit import *
import radio, moveKitronik

pin15.set_pull(pin15.PULL_UP)
pin16.set_pull(pin15.PULL_UP)
robot = moveKitronik.KitronikMK2()
Count_Stops=0
Robot_Hand=moveKitronik.Servo(pin0)

while True:
    Right_Detector =pin15.read_digital();
    Left_Detector =pin16.read_digital();

    if Left_Detector == 0 and Right_Detector == 0:
        robot.stop()
        if Count_Stops%2 ==0:
            Robot_Hand.write_angle(180)
            sleep(300)
            Robot_Hand.write_angle(90)
        else:
            Robot_Hand.write_angle(0)
            sleep(300)
            Robot_Hand.write_angle(90)
        sleep(1000)
        Count_Stops= Count_Stops+1
        robot.turnLeftDegrees(135)

    if Left_Detector == 1 and Right_Detector == 0:
        robot.turnRight()
    if Left_Detector == 0 and Right_Detector == 1:
        robot.turnLeft()
    if Left_Detector == 1 and Right_Detector == 1:
        robot.driveForward()

```

ZÁVĚR

V první části práce jsem zanalyzoval výuku algoritmizace na daných vysokých školách, u kterých jsem vybral předměty, které se základy algoritmizace zaobírají. Tyto předměty a jejich jednotlivé přístupy jsem mezi sebou stručně porovnal a vyjmenoval jejich největší rozdíly.

Ve druhé části popisuji počítač BBC micro:bit po stránce hardwarové a softwarové. Nejdříve jsem uvedl motivy vývoje počítače a porovnal novější verzi micro:bit V2 se starší V1.

V části třetí probírám různé možnosti programování počítače BBC micro:bit. Popisuji zde více do hloubky programování v MicroPythonu a vizuální programování v Microsoft Makecode, který jsem osobně vyhodnotil jako nejlepší editor pro začátečníky v programování.

Ve čtvrté kapitole seznamuji s robotem Kitronik :MOVE mini MK2 a dalším hardwarovým vybavením použitým v úlohách, ke kterým patří gamepad, senzory nebo reproduktor.

Poslední část obsahuje celkem 10 úloh, které jsou zrealizovány pomocí robota Kitronik :MOVE mini MK2. Cílem úloh bylo ukázat různé funkce, kterými robot movemini a počítač BBC micro:bit disponují. V úlohách jsou použity senzory pro detekování vzdálenosti od předmětu a pro zjištění černé nebo bílé plochy, po který robot jede. Dále jsem hojně využíval bezdrátovou komunikaci mezi více počítači micro:bit, a to pomocí radio modulu. Úlohy jsou řazeny podle obtížnosti a některé na sebe navazují, doporučuji je tedy realizovat postupně.

BBC micro:bit ve spojení s robotem Kitronik :MOVE mini MK2 může být vhodným vstupem do světa algoritmů z důvodu nízké ceny a jednoduchého programování, které umožňuje programovací prostředí MakeCode. Na druhou stranu však musím zmínit, že robot je z důvodu jednoduché konstrukce často nepřesný a některé funkce se nechovají vždy stejně.

POUŽITÁ LITERATURA

- [1] HALFACREE, Gareth. The official BBC micro:bit user *guide*. Indianapolis, Indiana: John Wiley, 2018. 312 s. ISBN 9781119386735.
- [2] DONAT, Wolfram. Getting Started with the micro:bit: coding and making with the bbc's open development board. San Francisco: Maker Media, Inc., 2017. 160 s. ISBN 9781680453027.
- [3] MONK, Simon. Programming the BBC Micro:Bit: Getting Started with Micropython. Columbia: Mcgraw Hill Book Co, 2017. 176 s. ISBN 9781260117585.
- [4] Micro:bit Educational Foundation. Wireless micro:bit: Help & Support. Micro:bit Help & Support [online]. 01. 06. 2020 [cit. 2021-01-03]. Dostupné z: <https://support.microbit.org/support/solutions/articles/19000083637-using-the-micro-bit-wirelessly>
- [5] Microsoft. Micro:bit pins. Microsoft Makecode [online]. 2020 [cit. 2021-01-03]. Dostupné z: <https://makecode.microbit.org/device/pins>
- [6] SEBASTIAN, Anthony. BBC Micro:bit—a free single-board PC for every Year 7 kid in the UK. ArsTechnica.com [online]. 7. 8. 2015 [cit. 2021-04-03]. Dostupné z: <https://arstechnica.com/gadgets/2015/07/bbc-microbit-a-free-single-board-pc-for-every-year-7-kid-in-the-uk/>
- [7] DONISON, Mark. The Differences Between micro:bit V1 And microbit V2. Kitronik [online]. 13. 10. 2020 [cit. 2021-03-05]. Dostupné z: <https://kitronik.co.uk/blogs/resources/explore-micro-bit-v1-microbit-v2-differences>
- [8] Micro:bit Educational Foundation. Hardware Details of the 1.5 micro:bit revision, featuring a combined motion sensor. micro:bit [online]. 2021 [cit. 2021-03-05]. Dostupné z: <https://tech.microbit.org/hardware/1-3-revision/>
- [9] Micro:bit Educational Foundation. Open source. micro:bit [online]. [2020] [cit. 2021-03-05]. Dostupné z: <https://microbit.org/get-started/user-guide/open-source/>
- [10] PEREIRA ATENCIO, Carlos. Awesome micro:bit. GitHub [online]. [2020] [cit. 2021-03-05]. Dostupné z: <https://github.com/carlosperate/awesome-microbit#-other-languages>
- [11] SENEVIRATNE, Pradeeka. BBC micro:bit Recipes: Learn Programming with Microsoft MakeCode Blocks. New York: Apress, 2019. ISBN 9781484249123.
- [12] Micro:bit Educational Foundation. User guide: Features in depth. micro:bit [online]. 2020 [cit. 2021-03-05] Dostupné z: <https://www.microbit.org/get-started/user-guide/features-in-depth/>
- [13] Microsoft. About MakeCode. MakeCode [online]. 2021 [cit. 2021-03-04] Dostupné z: <https://makecode.com/about>
- [14] Micro:bit Educational Foundation. The micro:bit runtime DAL/CODAL. micro:bit [online]. 2021 [cit. 2021-03-04]. Dostupné z: <https://tech.microbit.org/software/runtime/>

- [15] Micro:bit Educational Foundation. Bluetooth. [online]. 2021 [cit. 2021-05-02]. Dostupné z: <https://tech.microbit.org/bluetooth/>
- [16] Micro:bit Educational Foundation. DAPLink and the USB interface [online]. 2021 [cit. 2021-03-04]. Dostupné z: <https://tech.microbit.org/software/daplink-interface/>
- [17] Micro:bit Educational Foundation. Hardware: Details of the latest micro:bit hardware revision [online]. 2021 [cit. 2021-03-04]. Dostupné z: <https://tech.microbit.org/hardware/>
- [18] AUSTIN, Jonny, BAKER, Howard, BALL, Thomas, DEVINE, James, FINNERY, Joe, Peli De Halleux, HODGES, Steve, MOSKAL, Michał, STOCKDALE, Gareth. The BBC micro:bit – From the U.K. to the World. Communications of the ACM [online]. březem 2020 [cit. 2021-03-04]. Dostupné z: <https://cacm.acm.org/magazines/2020/3/243028-the-bbc-microbit-from-the-u-k-to-the-world/fulltext>
- [19] Waveshare. Joystick for micro:bit. Waveshare [online]. 12.7.2018 [cit. 2021-03-04]. Dostupné z: https://www.waveshare.com/wiki/Joystick_for_micro:bit
- [20] Microbit Playground. Class for Servo Control in Python on the Microbit. GitHub [online]. [San Francisco]: GitHub, 28.12.2016 [cit. 2021-03-04]. Dostupné z: <https://github.com/microbit-playground/microbit-servo-class>
- [21] Seed Technology. Grove - Ultrasonic Ranger. *Seed Technology* [online]. 2017 [cit. 2021-03-04]. Dostupné z: https://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/
- [22] ČVUT. Programování a algoritmizace 1. [online]. 2015 [cit. 2021-03-04]. Dostupné z: <http://bk.fit.cvut.cz/cz/predmety/00/00/00/00/00/00/01/12/15/p1121506.html>
- [23] VUT. Detail předmětu Základy programování. [online]. 2020 [cit. 2021-03-04]. Dostupné z: <https://www.vutbr.cz/studenti/predmety/detail/224975>
- [24] UPCE. Popis předmětu BPALP. [online]. 2020 [cit. 2021-03-06]. Dostupné ve Stagu: <https://portal.upce.cz/StagPortletsJSR168/CleanUrl?urlid=prohlizeni-predmet-sylabus&predmetZkrPrac=KST&predmetZkrPred=BPALP&predmetRok=2020&predmetSemestr=ZS>
- [25] OSU. Popis předmětu Základy algoritmizace. [online]. 2019 [cit. 2021-03-04]. Dostupné ve Stagu: <https://portal.osu.cz/stag?urlid=prohlizeni-predmet-sylabus&predmetZkrPrac=KIP&predmetZkrPred=ALDS1&predmetRok=2020&predmetSemestr=ZS>
- [26] Kitronik. Kitronik :MOVE mini MK2 buggy kit. Kitronik [online]. 2021 [cit. 2021-04-05]. Dostupné z: <https://kitronik.co.uk/products/5652-move-mini-mk2-buggy-kit-excl-microbit>
- [27] Kitronik. Kitronik :MOVE Line Following Board. Kitronik [online]. 2021 [cit. 2021-04-05]. Dostupné z: <https://kitronik.co.uk/products/5629-move-line-following-board-for-bbc-microbit>

- [28] Kitronik. Kitronik :MOVE Sensor Interface Board for the BBC micro:bit. Kitronik [online]. 2021 [cit. 2021-04-05]. Dostupné z: <https://kitronik.co.uk/products/5628-move-sensor-interface-board-for-the-bbc-microbit>
- [29] Micro:bit Educational Foundation. HEX file format. micro:bit [online]. 2021 [cit. 2021-04-05]. Dostupné z: <https://tech.microbit.org/software/hex-format/>
- [30] Micro:bit Educational Foundation. The micro:bit software ecosystem. micro:bit [online]. 2021 [cit. 2021-04-05]. Dostupné z: <https://tech.microbit.org/software/>
- [31] Micro:bit Educational Foundation. MicroPython. micro:bit [online]. 2021 [cit. 2021-04-05]. Dostupné z: <https://tech.microbit.org/software/micropython/>
- [32] Micro:bit Educational Foundation Python guide. micro:bit [online]. 2021 [cit. 2021-04-05]. Dostupné z: <https://microbit.org/get-started/user-guide/python/>
- [33] Micro:bit Educational Foundation. What is the difference between 'import microbit' and 'from microbit import *'. micro:bit [online]. 2.5. 2019 [cit. 2021-04-05]. Dostupné z: <https://support.microbit.org/support/solutions/articles/19000080153-what-is-the-difference-between-import-microbit-and-from-microbit-import->
- [34] KOVARSKÝ, Josef. Ako sa robot Kitronik :MOVE mini MK2 vyhýba prekážkam. Blog zonepi. [online]. 20.5.2020 [cit. 2021-04-05]. Dostupné z: <https://blog.zonepi.cz/ako-sa-robot-kitronik-move-mini-mk2-vyhyba-prekazkam/>
- [35] Micro:bit Educational Foundation. Guide to mobile & tablet apps. micro:bit [online]. 2021 [cit. 2021-04-05]. Dostupné z: <https://microbit.org/get-started/user-guide/mobile/>
- [36] UPCE. Popis předmětu BZALG. [online]. 2020 [cit. 2021]. Dostupné ve Stagu: <https://portal.upce.cz/StagPortletsJSR168/CleanUrl?urlid=prohlizeni-predmet-sylabus&predmetZkrPrac=KIT&predmetZkrPred=BZALG&predmetRok=2020&predmetSemestr=ZS>
- [37] Seed Technology. Grove Inventor Kit for micro:bit. Seed Technology [online]. 2018 [cit. 2021-03-04]. Dostupné z: <https://www.seeedstudio.com/Grove-Inventor-Kit-for-micro-bit-p-2891.html>
- [38] Radio. BBC micro:bit MicroPython [online]. 2016 [cit. 2021-05-05]. Dostupné z: <https://microbit-micropython.readthedocs.io/en/v1.0.1/index.html#>

PŘÍLOHY

Příloha A – Modul movekitronik.....	61
Příloha B – Modul waweshareJoystick.....	63
Příloha C – Modul ultrasonicsenzor	65

PŘÍLOHA A – MODUL MOVEKITRONIK

Kód třídy Servo jsem převzal z Microbit Playground.[20] Třídu MK2 jsem částečně převzal od výrobce robota.[26]

```
from microbit import *
class Servo:
    def __init__(self, pin, freq=50, min_us=600, max_us=2400, angle=180):
        self.min_us = min_us
        self.max_us = max_us
        self.us = 0
        self.freq = freq
        self.angle = angle
        self.analog_period = 0
        self.pin = pin
        analog_period = round((1/self.freq) * 1000) # hertz to miliseconds
        self.pin.set_analog_period(analog_period)

    def write_us(self, us):
        us = min(self.max_us, max(self.min_us, us))
        duty = round(us * 1024 * self.freq // 1000000)
        self.pin.write_analog(duty)
        # self.pin.write_digital(0) # turn the pin off

    def write_angle(self, degrees=None):
        degrees = degrees % 360
        total_range = self.max_us - self.min_us
        us = self.min_us + total_range * degrees // self.angle
        self.write_us(us)

class KitronikMK2:
    def __init__(self):
        self.MILISEC_IN_A_SEC=1000
        self.DISTANCE_PER_SEC=100
        self.DEGREES_PER_SEC=200
        self.left_servo = Servo(pin2)
        self.right_servo = Servo(pin1)

    def turnLeftDegrees(self,degrees):
        self.stop()
        timeToWait=(degrees*self.MILISEC_IN_A_SEC)/self.DEGREES_PER_SEC;
        self.right_servo.write_angle(45)
        self.left_servo.write_angle(45)
        sleep(timeToWait)
        self.stop()

    def turnRightDegrees(self,degrees):
        self.stop()
        timeToWait=(degrees*self.MILISEC_IN_A_SEC)/self.DEGREES_PER_SEC;
```

```
self.right_servo.write_angle(135)
self.left_servo.write_angle(135)
sleep(timeToWait)
self.stop()

def turnLeft(self):
    self.right_servo.write_angle(45)
    self.left_servo.write_angle(45)

def turnRight(self):
    self.right_servo.write_angle(135)
    self.left_servo.write_angle(135)

def driveForward(self):
    self.right_servo.write_angle(0)
    self.left_servo.write_angle(180)

def driveBackward(self):
    self.right_servo.write_angle(180)
    self.left_servo.write_angle(0)

def stop(self):
    self.right_servo.write_angle(90)
    self.left_servo.write_angle(90)
```

PŘÍLOHA B – MODUL WAWESHAREJOYSTICK

Kód níže jsem převzal ze stránek výrobce gamepadu.[19]

```
from microbit import *

JoyStick_P = pin8
JoyStick_X = pin1
JoyStick_Y = pin2
KEY_A = pin5
KEY_B = pin11
KEY_C = pin15
KEY_D = pin14
KEY_E = pin13
KEY_F = pin12
DIR = {
    'NONE': 0,
    'U': 1,
    'D': 2,
    'L': 3,
    'R': 4,
    'U_L': 5,
    'U_R': 6,
    'D_L': 7,
    'D_R': 8
}
KEY = {
    'NONE': 0,
    'P': 1,
    'A': 2,
    'B': 3,
    'C': 4,
    'D': 5,
    'E': 6,
    'F': 7
}
}
class JOYSTICK():
    def __init__(self):
        self.Read_X = JoyStick_X.read_analog()
        self.Read_Y = JoyStick_Y.read_analog()

    def Listen_Dir(self, Dir):
        Get_Rocker = DIR['NONE']
        New_X = JoyStick_X.read_analog()
        New_Y = JoyStick_Y.read_analog()
        Dx = abs(self.Read_X - New_X)
        Dy = abs(self.Read_Y - New_Y)
        Right = New_X - self.Read_X
        Left = self.Read_X - New_X
        Up = New_Y - self.Read_Y
```

```
Down = self.Read_Y - New_Y
Precision = 150
```

```
if Right > Precision and Dy < Precision:
    Get_Rocker = DIR['R']
elif Left > Precision and Dy < Precision:
    Get_Rocker = DIR['L']
elif Up > Precision and Dx < Precision:
    Get_Rocker = DIR['U']
elif Down > Precision and Dx < Precision:
    Get_Rocker = DIR['D']
elif Right > Precision and Up > Precision:
    Get_Rocker = DIR['U_R']
elif Right > Precision and Down > Precision:
    Get_Rocker = DIR['D_R']
elif Left > Precision and Up > Precision:
    Get_Rocker = DIR['U_L']
elif Left > Precision and Down > Precision:
    Get_Rocker = DIR['D_L']
else:
    Get_Rocker = DIR['NONE']
```

```
if Dir == Get_Rocker:
    return True
else:
    return False
```

```
def Listen_Key(self, Key):
    read_key = KEY['NONE']
    if button_a.is_pressed():
        read_key = KEY['A']
    elif button_b.is_pressed():
        read_key = KEY['B']
    elif KEY_C.read_digital() == 0:
        read_key = KEY['C']
    elif KEY_D.read_digital() == 0:
        read_key = KEY['D']
    elif KEY_E.read_digital() == 0:
        read_key = KEY['E']
    elif KEY_F.read_digital() == 0:
        read_key = KEY['F']
    elif JoyStick_P.read_digital() == 0:
        read_key = KEY['P']
    else:
        read_key = KEY['NONE']
```

```
if Key == read_key:
    return True
else:
    return False
```


PŘÍLOHA C – MODUL ULTRASONICSENZOR

Kód níže jsem převzal ze stránek výrobce senzoru. [21]

```
import sys
from microbit import *
import time

_TIMEOUT1 = 1000
_TIMEOUT2 = 10000

class GroveUltrasonicRanger(object):
    def __init__(self, pin):
        self.dio = pin

    def _get_distance(self):
        self.dio.write_digital(0)
        time.sleep_us(2)
        self.dio.write_digital(1)
        time.sleep_us(10)
        self.dio.write_digital(0)
        t0 = time.ticks_us()
        count = 0
        while count < _TIMEOUT1:
            if self.dio.read_digital():
                break
            count += 1
        if count >= _TIMEOUT1:
            return None

        t1 = time.ticks_us()
        count = 0
        while count < _TIMEOUT2:
            if not self.dio.read_digital():
                break
            count += 1
        if count >= _TIMEOUT2:
            return None

        t2 = time.ticks_us()
        distance = ((t2 - t1) / 29.0 / 2.0)
        return distance

    def get_distance(self):
        while True:
            dist = self._get_distance()

            if dist:
                return dist
```