



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**ŘÍZENÍ LEGO POWERED-UP HUB Z PROSTŘEDÍ EV3**

CONTROLLING LEGO POWERED-UP HUB FROM EV3

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**JAKUB POTOČEK**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. RADEK HRANICKÝ, Ph.D.**

BRNO 2023

## Zadání bakalářské práce



147225

Ústav: Ústav informačních systémů (UIFS)  
Student: **Potoček Jakub**  
Program: Informační technologie  
Specializace: Informační technologie  
Název: **Řízení LEGO Powered-up hub z prostředí EV3**  
Kategorie: Vestavěné systémy  
Akademický rok: 2022/23

### Zadání:

1. Seznamte se s prostředími pro motorizaci a řízení modelů LEGO. Zaměřte se na systémy Control+/Powered-up a Mindstorms EV3.
2. Nastudujte specifikaci hardware, software a použitých komunikačních protokolů. Nastudujte principy prostředí EV3 a PyBricks MicroPython.
3. Analyzujte možnosti pro komunikaci prvků EV3 Intelligent Brick a Powered up Hub: sensorové rozhraní, Bluetooth apod.
4. Po konzultaci s vedoucím navrhnete řešení pro ovládání Powered-up Hub z prostředí EV3, např. ovladač pro periferní zařízení a knihovnu pro MicroPython.
5. Navržené řešení implementujte.
6. Sestavte model LEGO zahrnující EV3 Intelligent Brick a Powered-up hub s alespoň jedním připojeným motorem. Vytvořte program pro EV3, který bude model ovládat. Na modelu demonstруйте použitelnost vašeho řešení.
7. Diskutujte dosažené výsledky a navrhnete možná vylepšení.

### Literatura:

- Crnokić, Boris, et al. STEM Classroom: Creating a Python Application for an EV3 Brick Robotic System Used to Transport 3D Printed Boxes. *Annals of DAAAM & Proceedings* 7.1. 2020.
- Li, Yixiao, et al. A platform for LEGO mindstorms EV3 based on an RTOS with MMU support. *OSPERT*: 51. 2014.
- LEGO MindStorms EV3 Communication Developer Kit, Lego Group. 2013.
- Getting started with LEGO Mindstorms Education EV3 MicroPython. Lego Group. Version 2.0.0. 2020.
- PyBricks Documentation. The Pybricks Authors. 2020.

Při obhajobě semestrální části projektu je požadováno:

Body 1 až 4

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hranický Radek, Ing., Ph.D.**  
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.  
Datum zadání: 1.11.2022  
Termín pro odevzdání: 10.5.2023  
Datum schválení: 31.10.2022

## Abstrakt

Táto práca sa zaoberá komunikáciou medzi dvoma nekompatibilnými programovateľnými LEGO kockami LEGO Mindstorms EV3 a LEGO Control+ Technic Hub. Úlohou tejto práce bolo vytvoriť komunikačné rozhranie umožňujúce prenos dát medzi týmito dvoma kockami a umožniť tak kocke EV3 posielat a prípadne aj prijímať dáta z kocky Technic hub. Počas tvorby tejto práce som navrhol a skonštruoval dva prototypy využívajúce optické rozhranie pre prenos dát. Druhý prototyp umožňuje obojsmernú komunikáciu. Tým umožňuje prenášať dáta zo senzorov a návratové hodnoty funkcií do EV3. Aj napriek tomu, že výsledné riešenie neumožňuje dostatočne rýchly prenos dát pre technické limitácie kocky Technic hub je toto riešenie vhodné na použitie pre vlastníkov týchto LEGO stavebníc hľadajúcich možnosti ako využiť obe stavebnice pri tvorbe modelov.

## Abstract

This work deals with the communication between two incompatible programmable LEGO bricks, the LEGO Mindstorms EV3 and the LEGO Control+ Technic Hub. The task of this work was to create a communication interface enabling the transfer of data between these two bricks and thus enable the EV3 brick to send and possibly also receive data from the Technic hub brick. During the creation of this work, I designed and constructed two prototypes using an optical interface for data transmission. The second prototype enables two-way communication. This enables data from sensors and function return values to be transferred to EV3. Despite the fact that the resulting solution does not allow fast enough data transfer due to the technical limitations of the Technic hub brick, this solution is suitable for use by owners of these LEGO building blocks looking for ways to use both bricks when creating models.

## Klíčové slová

vstavané systémy, LEGO Mindstorms EV3, LEGO Technic, LEGO PoweredUp, LEGO Control+, prenos dát, optické rozhranie, oprotlen, komunikačný protokol, UART

## Keywords

embedded systems, LEGO Mindstorms EV3, LEGO Technic, LEGO PoweredUp, LEGO Control+, data transfer, optical interface, oprotcopuler, communication protocol, UART

## Citácia

POTOČEK, Jakub. *Řízení LEGO Powered-up hub z prostředí EV3*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Radek Hranický, Ph.D.

# Řízení LEGO Powered-up hub z prostředí EV3

## Prehlásenie

Vyhlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Radka Hranického, PhD. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Jakub Potoček

7. mája 2024

## Podakovanie

Rád by som poďakoval svojmu vedúcemu práce, Ing. Radkovi Hranickému, za možnosť pracovať na tejto práci a za jeho pomoc pri jej tvorbe. Taktiež by som chel poďakovať mojím rodičom a Ing. Hurtoňovi ktorý ma podporovali počas celého štúdia.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Stručná história programovateľných stavebníc LEGO</b>	<b>7</b>
2.1	LEGO/Logo . . . . .	7
2.2	Červená kocka . . . . .	8
2.3	LEGO Technic Interface A Control Board . . . . .	8
2.4	LEGO Technic Control Center , Control Center 2.0 a LEGO Dacta Technic Control Center . . . . .	9
2.5	LEGO Dacta Serial Interface Box . . . . .	10
2.6	LEGO Barcode Reader/Code Pilot . . . . .	10
2.7	LEGO Mindstorms RCX . . . . .	11
2.8	LEGO Cybermaster . . . . .	12
2.9	LEGO Scout . . . . .	12
2.10	LEGO Micro Scout . . . . .	13
2.11	LEGO Spybotics . . . . .	13
2.12	LEGO Mindstorms NXT a NXT 2.0 . . . . .	14
2.13	LEGO WeDo 1.0 a WeDo 2.0 . . . . .	15
2.14	LEGO Mindstorms EV3 . . . . .	16
2.15	Lego Powered Up/Control+ . . . . .	16
2.16	LEGO Mindstorms Robot Inventor . . . . .	18
<b>3</b>	<b>Komunikačné protokoly</b>	<b>19</b>
3.1	Vlastnosti a delenie komunikačných protokolov využívaných vo vstavaných systémoch . . . . .	19
3.2	Protokol UART . . . . .	20
3.3	Protokol I2C . . . . .	21
<b>4</b>	<b>Technické parametre LEGO Mindstorms EV3 a LEGO Technic Hub, a ich periférie</b>	<b>24</b>
4.1	Technické parametre kocky LEGO Mindstorms EV3 . . . . .	24
4.2	Technické parametre kocky LEGO Technic Hub . . . . .	25
4.3	Konektor a port EV3, a Powered Up . . . . .	26
4.3.1	Porty EV3 kocky . . . . .	26
4.3.2	Powered Up porty . . . . .	27
4.4	Identifikácia periférií . . . . .	27
4.5	Možnosti prenosu dát medzi kockami . . . . .	28
4.5.1	Využitie LEGO periférií . . . . .	29
4.5.2	LEGO komunikačný protokol . . . . .	29

<b>5</b>	<b>Návrh a tvorba prototypu s využitím optického rozhrania</b>	<b>31</b>
5.1	Prenos bitu . . . . .	31
5.2	Prenos bajtu . . . . .	31
5.2.1	Časovanie prenosu . . . . .	32
5.2.2	Proces načítania bajtu . . . . .	32
5.3	Užívateľsky prívetivé odosielanie viac bajtových práv . . . . .	33
5.3.1	Prenos hodnôt . . . . .	34
5.3.2	Inštrukcie . . . . .	34
5.4	Galvanické oddelenie obvodov s rôznym zdrojom napätia . . . . .	35
5.4.1	Optočlen . . . . .	35
5.4.2	RF . . . . .	36
5.4.3	Magnetické oddelenie . . . . .	36
5.5	Prenos dát pomocou LEGO periférií . . . . .	36
5.5.1	Vysielač . . . . .	36
5.5.2	Tvorba vysielača . . . . .	37
5.5.3	Prijímač . . . . .	37
5.5.4	Prenos dát . . . . .	37
5.5.5	Konštrukcia prototypu a prezentovanie jeho funkcionality na jednom programe . . . . .	38
5.5.6	Návrh vylepšení pre ďalší vývoj . . . . .	38
<b>6</b>	<b>Návrh a realizácia obojsmerného komunikačného protokolu</b>	<b>40</b>
6.1	Identifikácia rozhrania pomocou imitácie Arduina ako oficiálnej LEGO periférie	40
6.2	Rozpoznávanie signálov indikujúcich logickú 1 a logickú 0 pri prijímaní a vysiellaní. . . . .	40
6.3	Galvanické oddelenie . . . . .	41
6.4	Konštrukcia a implementácia . . . . .	42
6.4.1	Tvorba komunikačného rozhrania . . . . .	42
6.4.2	Tvorba programu . . . . .	44
6.4.3	Konštrukcia LEGO modelu a zhodnotenie výsledného riešenia . . . .	45
<b>7</b>	<b>Záver</b>	<b>47</b>
	<b>Literatúra</b>	<b>49</b>

# Zoznam obrázkov

2.1	LEGO/Logo kocka <sup>1</sup>	7
2.2	Červená kocka <sup>2</sup>	8
2.3	LEGO Technic Interface A <sup>3</sup>	8
2.4	LEGO Technic Control Center <sup>4</sup>	9
2.5	LEGO Technic Control Center 2.0 <sup>5</sup>	9
2.6	LEGO Dacta Serial Interface Box <sup>6</sup>	10
2.7	LEGO Code Pilot <sup>7</sup>	11
2.8	LEGO RCX Hub <sup>8</sup>	12
2.9	LEGO Cybermaster Hub <sup>9</sup>	12
2.10	LEGO Scout Hub <sup>10</sup>	13
2.11	LEGO Micro Scout Hub <sup>11</sup>	13
2.12	LEGO Spybotics Hub <sup>12</sup>	14
2.13	LEGO NXT Hub <sup>13</sup>	15
2.14	LEGO WeDo Hub <sup>14</sup>	15
2.15	LEGO WeDo 2.0 Hub <sup>15</sup>	16
2.16	LEGO EV3 Hub <sup>16</sup>	16
2.17	LEGO Move Hub, City Hub, Technic Hub <sup>17 18 19</sup>	18
2.18	Spike Prime Hub, Spike Essemcial Hub <sup>20 21</sup>	18
2.19	LEGO Robot Inventor Hub <sup>22</sup>	18
3.1	Schéma zapojenia UART	21
3.2	Grafické znázornenie jedného rámca v protokole UART.	21
3.3	Schéma zapojenia zariadení na I2C zbernicu	22
3.4	Grafické znázornenie jedného rámca v I2C protokole	23
4.1	Nastavenie pinov implementované na výstupných portoch pre aktuátory na kocke LEGO MINDSTORMS EV3 <sup>23</sup>	26
4.2	Nastavenie pinov implementované na vstupne-výstupných portoch pre senzory na kocke LEGO MINDSTORMS EV3 <sup>23</sup>	27
4.3	Nastavenie pinov implementované na vstupne-výstupných portoch pre periférie na kocke LEGO Control+ Technic Hub [12]	27
5.1	Načítanie bitu	32
5.2	Odosielanie bajtu	32
5.3	Načítanie bajtu	33
5.4	Prenos Dát	34
5.5	Schéma napojenia LED na EV3 konektor	37
5.6	Prototyp jednosmernej komunikácie využívajúci LEGO senzor farby a vzdialenosti	39

6.1	Schéma jedno kanálového optočlenu 6N137 . . . . .	41
6.2	Schéma prepojenia fyzického rozhrania kocky Technic Hub a EV3 s Arduinom . . . . .	42
6.3	Komunikačný kanál využívajúci optočleny. . . . .	43
6.4	Model využívajúci moje komunikačné rozhranie a ďalšie LEGO prvky. . . . .	46



# Kapitola 1

## Úvod

LEGO so svojimi plastovými kockami sa stalo jednou z najlepších platforiem pre rozvíjanie detskej kreativity. Samotné možnosti LEGO kociek sú takmer nekonečné. Čo sa však stane, keď sa pridajú ďalšie prvky ako motory? Prvá motorizovaná LEGO súprava vznikla už v roku 1966 a umožnila nadšencom do vláčikov vytvárať vlastné modely či dráhy. V roku 1977 bola predstavená rada LEGO Technic, ktorá rozšírila LEGO o nové kocky pre vytváranie komplikovanejších áut [7]. Novo pridané ozubené kolieska, osky a nový typ blokov umožnil autám zatáčať, alebo ovládať korbu auta. V 80tých rokoch, keď počítače začali byť prístupnejšie, začali výskumníci z MIT pracovať na výukovej stavebnici pre deti, ktorá by rozšírila ich schopnosti s prácou s počítačmi a ako platformu si zvolili stavebnice LEGO [26]. Najskôr sa stavebnica využívala v školách, neskôršie modely sa však už dali kúpiť aj individuálne. Aj keď boli stavebnice mierené prevažne na deti, našla sa však aj komunita nadšencov ktorí si obľúbili tieto stavebnice. Tieto nové možnosti im umožnili vytvárať komplexnejšie stroje schopné ovládania motorov, získavania informácií o okolí pomocou senzorov, neskôr aj možnosť diaľkového ovládania. Schopnosť programovania rozšírila možnosti LEGO sád na takmer nekonečné. Používateľ sa už nemusel riadiť žiadnym návodom či manuálom a mohol vytvárať vlastné stroje.

Keďže v priebehu rokov vyšli rôzne modely programovateľných kociek, množstvo z nich nie je kompatibilné. Aj kocky ktoré vyšli v jednu dobu neboli kompatibilné. Napríklad kocka RCX a kocka Cybermaster používali každá iné rozhranie. RCX využívalo infračervený vysielač pre nahranie kódu na kocku, Cybermaster zas rádiový vysielač. Ani kocky v jednej vývojovej rade nemusia byť kompatibilné. Prvá generácia LEGO Mindstorms s nasledujúcimi generáciami nebola kompatibilná, keďže konektory sa líšili. Druhá generácia s tretou už mali rovnaké konektory a motory novšej EV3 kocky boli kompatibilné so staršou NXT kockou, rovnako tak motory NXT boli kompatibilné s EV3. Zatiaľ čo NXT senzory boli kompatibilné s EV3, EV3 senzory už s NXT kompatibilné neboli. Taktiež žiadna zo zmienených kociek nebola kompatibilná s LEGO PoweredUp/Control+ modelmi. Až nasledujúci model LEGO Mindstorm robot inventor, ktorý mal konektory kompatibilné s PoweredUp/Control+ modelmi. Tým pádom nebol kompatibilný s predchádzajúcimi verziami LEGO Mindstorms.

Existujú možné riešenia pre vybrané kocky. V minulosti existovalo naraz niekoľko podobných modelov. Tie bolo možné prepojiť pomocou VLL (Virtual leaser lines) protokolu. Bolo tak možné prepojiť kocku Scout alebo RCX s kockou Micro Scout [24]. Taktiež existuje spôsob, ako prepojiť kocku EV3 s kockou Spike Prime alebo Large Techic Hub (známy tiež ako Robot Inventor Hub) pomocou Bluetooth [20]. Momentálne však nie je spôsob, ako prepojiť kocky PoweredUp/Control+ s EV3. Pre prepojenie nekompatibilných perifé-

rií s kockami existujú rôzne riešenia. Vznikol spôsob, ako prepojiť nekompatibilné motory LEGO Power Functions s kockou EV3 [1]. Existujú rôzne nadstavce pre prepojenie periférie a programovateľných kociek. Nejedná sa však o oficiálne riešenie.

Na možné riešenie prepojenia dvoch nekompatibilných kociek sa zamerala moja práca. Zadanie navrhol vedúci mojej práce ktorý sa vo voľnom čase zaujíma o robotické stavebnice LEGO Mindstorms EV3 a Powered Up/Control+. Keďže tieto dve stavebnice nie sú navzájom kompatibilné, navrhol som spôsob, ako by mohli navzájom komunikovať. To by malo pomôcť nielen jemu, ale aj hocikomu, kto vlastní nekompatibilné programovateľné LEGO kocky.

Počas tvorby tejto práce som skonštruoval dva prototypy. Prvý využíval LEGO periférie pre prenos dát a pomocou svetelných signálov prenášal dáta jednosmerne z EV3 do kocky Technic hub. Prenosová rýchlosť však nebola dostatočná. Navrhol som preto aj druhý prototyp využívajúce vysoko rýchlostné optočleny v snahe urýchliť prenos dát a umožniť obojsmernú komunikáciu. Architektúra kocky Technic hub a firmvér Pybricks pre PoweredUp zariadenia však nie sú na takéto využitie usposobené a keďže firmware nieje otvorený softvér, nemohol som ho upraviť. Druhý prototyp síce umožňuje obojsmernú komunikáciu, prenosová rýchlosť je však aj naďalej nevyhovujúca. Aj napriek nedostatkom výsledného riešenia je táto metóda použiteľná pri stavbe vlastných LEGO modelov ako demonštrujem v záverečnej kapitole.

Táto práca pozostáva zo siedmich kapitol. Kapitola 2 popisuje stručnú históriu programovateľných LEGO setov a LEGO kociek. V kapitole 3 sa venujem teórii komunikačných protokolov pre vstavané systémy a detailnejšie popisujem UART a I2C, ktoré sa využívajú v programovateľných kockách LEGO. Kapitola 4 detailne popisuje vlastnosti kociek LEGO Mindstorms EV3 a LEGO Technic Hub Control +, popisuje akým spôsobom prebieha komunikácia s perifériami a predstavuje možnosti ich prepojenia. V kapitole 5 popisujem návrh komunikačného protokolu a tvorbu prototypu. Kapitulu 6 venujem vylepšeniu prototypu s využitím pokročilejších komponent a hlbším porozumením fungovania LEGO kociek a ich protokolov a tvorbe LEGO modelu pre demonštrovanie výsledného riešenia.

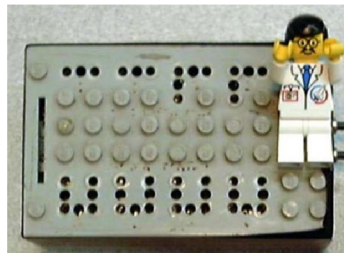
## Kapitola 2

# Stručná história programovateľných stavebníc LEGO

V tejto kapitole sa budem venovať predstaveniu jednotlivých programovacích kociek, ich histórii, vlastností, schopností, možnosti programovania a uvediem čísla LEGO setov, v ktorých sa dané kocky nachádzali.

### 2.1 LEGO/Logo

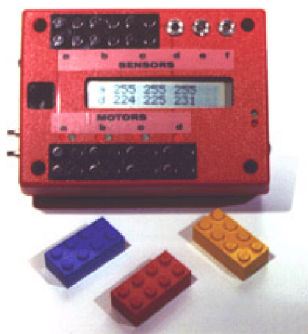
V polovici 80tých rokov sa výskumníci z MIT snažili vytvoriť programovateľné zariadenie, ktoré by umožnilo deťom vytvárať vlastné automatizované výtvyry a rozvíjali tak svoj záujem o počítače a svoje schopnosti riešenia problémov [26]. Výskumníci si preto vybrali LEGO ako platformu, na ktorej budú budovať svoje zariadenie. Ako programovací jazyk vytvorili upravenú verziu jazyka Logo. Ten bol už pre mnohé deti známy, vďaka programovateľným „korytnačkám“, či už na obrazovkách stolných počítačov, alebo reálnych robotických korytnačiek. Na LEGO/Logo kocke (obr. 2.1) sa však neovládalo žiadne už vyrobené zariadenie. Pomocou motorov, senzorov, prijímačov či vysielačov a samozrejme kociek LEGO sa dalo skonštruovať množstvo robotických alebo motorizovaných zariadení, kocka podporovala paralelizmus, takže nebol problém obsluhovať niekoľko procesov naraz, taktiež mohli jednotlivé kocky spolu komunikovať. Nejednalo sa však o vstavaný mikropočítač či mikroprocesor. LEGO/Logo kocka musela byť pripojená k počítaču a to značne limitovalo tvorbu pohyblivých výtvyrov. Keďže však v tej dobe neboli domáce počítače ešte natoľko rozšírené, hlavné uplatnenie našla LEGO/Logo kocka v školách.



Obr. 2.1: LEGO/Logo kocka <sup>1</sup>

## 2.2 Červená kocka

Ďalším modelom bola takzvaná Červená kocka (obr. 2.2) (preložené z Red Brick). Tá nebola konceptom príliš vzdialená od LEGO/Logo kocky, bola však robustnejšia. Predovšetkým však bola rozšírená o LCD obrazovku a mikroprocesor, vďaka ktorému nemusela byť neustále pripojená ku stolnému počítaču [26].



Obr. 2.2: Červená kocka <sup>2</sup>

## 2.3 LEGO Technic Interface A Control Board

Na rozdiel od Červenej kocky, tu sa nejednalo o prenosný mikroprocesor. Jednalo sa skôr o kontrolnú stanicu, ktorá umožňovala ovládať motory a senzory pomocou PC [11]. Táto kontrolná stanica (obr. 2.3) bola vydaná v roku 1986. Jednalo sa o edukačnú verziu určenú pre školy, a teda nebola voľne dostupná. Podporovala 3 4,5V motory schopné obojsmerného otáčania alebo 6 motorov, pokiaľ sa otáčali len jednosmerne a 2 senzory, senzor dotyku alebo svetelný senzor.

Táto LEGO kocka sa vyskytla v sadách:

- 1093 - Interface A (Dacta)
- 9750 - Interface A



Obr. 2.3: LEGO Technic Interface A <sup>3</sup>

<sup>1</sup><https://www.brothers-brick.com/2020/01/31/a-history-of-lego-education-part-2-path-to-mindstorms-feature/lego-education-part-2-evolution-of-mindstorms-brick/>

<sup>2</sup><https://el.media.mit.edu/logo-foundation/resources/logoupdate/v7n1/v7n1-pbrick.html>

<sup>3</sup><https://lgauge.com/technic/articles/LEGOInterfaceA/9750right.jpg>

## 2.4 LEGO Technic Control Center , Control Center 2.0 a LEGO Dacta Technic Control Center

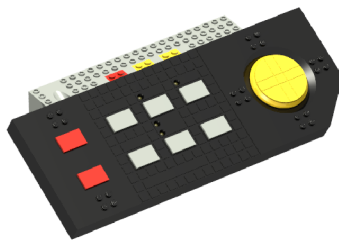
Prvá sada Control Center (obr. 2.4) vyšla v roku 1990, Dacta Control Center určený pre školy v roku 1993 a Control Center 2.0 (obr. 2.5) v roku 1995[4, 25]. Podporoval ovládanie 3 výstupných zariadení ako motory, lampy a zvukové elementy, a ukladal 2 programy. Primárnou funkciou Control Center bolo ovládanie výstupných zariadení, ktoré sa ovládali pomocou zabudovaných tlačidiel. Control Center podporovalo programovanie pomocou funkcie Record. Tá nasnímala pohyby, ktoré po stlačení tlačidla Play prehrala. Maximum zrefazených pohybov bolo 51, maximum pre jeden program 41 [3].

Tieto LEGO kocky sa vyskytli v sadách:

- 8094 - Control Center
- 8485 - Control Center II
- 9753 - Technic Control Center I (Dacta)



Obr. 2.4: LEGO Technic Control Center <sup>4</sup>



Obr. 2.5: LEGO Technic Control Center 2.0 <sup>5</sup>

---

<sup>4</sup>[https://brickscout.com/en/products/detail/4057296513314/part\\_\\_2840c01/black-technic-control-center-i](https://brickscout.com/en/products/detail/4057296513314/part__2840c01/black-technic-control-center-i)

<sup>5</sup><https://www.steinpalast.eu/1-x-lego-technic-elektric-control-center-ii-schwarz-geprueft-2-rote-6-alt-hell-graue-steuer-programmier-knoepfe-fuer-set-8485-9752-2840c02>

## 2.5 LEGO Dacta Serial Interface Box

Sada vyšla v roku 1993, je pokračovaním Interface A. Taktiež sa jedná o edukačný model (obr. 2.6) [2]. Podporoval však už 9V motory. Stanica obsahovala 8 portov pre motory a iné výstupné zariadenia a 8 portov pre senzory, taktiež jeden „Always On“ port pre testovanie. Programovanie prebiehalo v programe Control Lab jazykom Logo. Program umožňoval graficky zobrazovať hodnoty, grafy, vytvárať procedúry a tie priradovať tlačidlám.

Táto LEGO kocka sa vyskytla v sadách:

- 9751 - Control Lab Serial Interface & Adapter (Dacta)



Obr. 2.6: LEGO Dacta Serial Interface Box <sup>6</sup>

## 2.6 LEGO Barcode Reader/Code Pilot

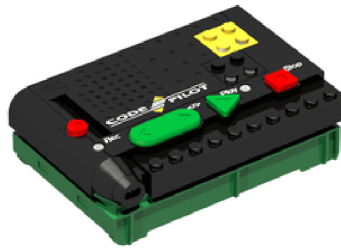
Sada obsahujúca kocku Pilot, (obr. 2.7) bola predstavená v roku 1997. Predstavovala zaujímavý koncept programovateľnej kocky tzv. Pilota. Kód sa nahrával do kocky pomocou skenera čiarových kódov [4]. V sete boli papiere s 44 vytlačenými čiarovými kódmi. Po ich naskenovaní sa nahral buď hotový program, alebo sa dal program vytvoriť tak, že sa jednotlivé inštrukcie skladali za seba. Počas behu programu mohlo byť pred senzor nainštalované časovacie koleso, vďaka ktorému mohol systém počítať počet otáčok a vypočítať tak rýchlosť. Kocka podporovala 2 motory a 2 dotykové senzory ktoré sa na ňu mohli pripojiť. Konektory boli obdobné so setom LEGO Mindstoms. Pilot bol teda programovateľný bez použitia počítača, a teda dnes by sa používal rovnako, ako v dobe keď vyšiel, čo sa nedá povedať o väčšine ostatných kociek.

Táto LEGO kocka sa vyskytla v sadách:

- 8479 - Barcode Multi-Set

---

<sup>6</sup><http://www.computinghistory.org.uk/det/6998/Lego-Dacta-Interface-70909/>



Obr. 2.7: LEGO Code Pilot <sup>7</sup>

## 2.7 LEGO Mindstorms RCX

Skutočná revolúcia nastala v roku 1998 keď Lego predstavilo LEGO Mindstorms Robotic Invention Set (obr. 2.8). Sada v hodnote 200 dolárov obsahoval 717 LEGO súčiastok, motory, kolieska, senzory a predovšetkým novú RCX kocku [15]. S takýmto setom mohol každý nadšenec vytvárať svoje projekty. Kocka bola napájaná 9V a bola kompatibilná s už existujúcimi 9V perifériami z iných LEGO setov. Kód sa písal v grafickom editore a do kocky sa nahrával pomocou špecializovaného infračerveného rozhrania. RCX kocka našla úspech nielen u detí, ale aj v komunite hackerov, ktorí vytvárali množstvo softvéru. Taktiež vznikali alternatívne programovacie prostredia napríklad NQC (Not Quite C) [22]. Okolo programovateľných LEGO kociek tak vznikla široká komunita ľudí, ktorú Lego podporilo webovými stránkami a fórami, pre zdieľanie výtvorov a návodov, a taktiež vznikom súťaží pre školákov napríklad FLL (First LEGO League) .

Táto LEGO kocka sa vyskytla v sadách:

- 3804 - Robotics Invention System, Version 2.0
- 9709 - RCX Programmable Brick
- 9719 - Robotics Invention System, Version 1.0
- 9747 - Robotics Invention System, Version 1.5
- 9785 - ROBO Technology Set (Serial)
- 9786 - ROBO Technology Set (USB)

---

<sup>7</sup><https://rebrickable.com/blog/373/classic-review-8479-1-barcode-multi-set/>



Obr. 2.8: LEGO RCX Hub <sup>8</sup>

## 2.8 LEGO Cybermaster

V roku 1998 vyšla zároveň s LEGO Mindstorms kocka LEGO Cybermaster (obr. 2.9). Miesto infračerveného rozhrania sa využívalo rádiové rozhranie, to umožnilo väčší dosah [21]. Taktiež obsahovala dva vstavané motory, vstavaný tachometer a rýchlomer. Ako senzory sa mohli použiť len pasívne senzory. Inak bola kocka výrazne limitovaná a nemala taký úspech ako RCX. Keďže každá kocka využívala iné komunikačné rozhranie, nebolo možné ich priamo prepojiť. Mohli však spolu komunikovať cez počítač s dvoma sériovými portmi.

Táto LEGO kocka sa vyskytla v sadách:

- 8482 - CyberMaster
- 8483 - CyberMaster with Storage Case



Obr. 2.9: LEGO Cybermaster Hub <sup>9</sup>

## 2.9 LEGO Scout

Cenovo dostupnejšou verziou k sade LEGO Mindstorms Robotic Invention Set bol LEGO Robotics Discovery Set, ktorý vyšiel v roku 1999. Ten obsahoval jednoduchšiu kocku Scout (obr. 2.10). Obsahovala dva porty pre motory a dva porty pre senzory, ale bez PC rozhrania [25]. Programovanie prebiehalo kombinovaním vstavaných inštrukcií, čo značne limitovalo jej použitie. Rovnako ako pri Cybermaster mohli byť použité len pasívne senzory. Aj keď LEGO plánovalo vytvoriť rozšírenie pre kocku Scout, náročnosť sa ukázala ako príliš vysoká a projekt tak bol zrušený. Scout však môže byť prepojený s RCX pomocou Send IR Message

<sup>8</sup><https://el.media.mit.edu/logo-foundation/resources/logoupdate/v7n1/v7n1-pbrick.html>

<sup>9</sup><https://www.satakore.com/lego-toy-store,,1,,5559,,8483-Technic-CyberMaster.html>



[16].

Táto LEGO kocka sa vyskytla v sadách:

- 9735 - Robotics Discovery Set



Obr. 2.10: LEGO Scout Hub <sup>10</sup>

## 2.10 LEGO Micro Scout

Kocka Micro scout, (obr. 2.11) je ešte jednoduchšia ako kocka Scout. Taktiež vyšla v roku 1999, nie je však programovateľná, môže len spúšťať jeden zo siedmich pred pripravených programov. Môže byť však ovládaná pomocou optického spojenia [28].

Táto LEGO kocka sa vyskytla v sadách:

- 9748 - Droid Developer Kit
- 9754 - Dark Side Developer Kit



Obr. 2.11: LEGO Micro Scout Hub <sup>11</sup>

## 2.11 LEGO Spybotics

Jedná sa o radu vychádzajúcu z LEGO Mindstorms. Vyšla v roku 2002 vo forme štyroch rôznych vozidiel, elektronická kocka je znázornená na obrázku 2.12. Vozidlá bolo možné ovládať infračerveným komunikačným rozhraním [6]. Sada taktiež obsahovala CD, ktoré

<sup>10</sup><https://www.bricklink.com/v2/catalog/catalogitem.page?P=32104c01#T=C&C=7>

<sup>11</sup><https://www.bricklink.com/v2/catalog/catalogitem.page?P=32344c01#T=C>

obsahovalo 10 misií, tie mohol používateľ splniť so svojím robotom, neskôr bolo možné vytvárať aj vlastné misie.

Táto LEGO kocka sa vyskytla v sadách:

- 3806 - Gigamesh G60
- 3807 - Snaptrax S45
- 3808 - Shadowstrike S70
- 3809 - Technojaw T55



Obr. 2.12: LEGO Spybotics Hub <sup>12</sup>

## 2.12 LEGO Mindstorms NXT a NXT 2.0

V roku 2006 prišiel nástupca RCX kocky. Kocka NXT (obr. 2.13) má nový typ konektorov, a tak staré motory nie sú priamo kompatibilné [17]. Ostáva však 9V napájanie. NXT je rozšírená o Bluetooth komunikačné rozhranie a nové senzory. V roku 2009 vyšla druhá verzia NXT 2.0. Sada predstavila nové funkcie, napríklad RFID, NXT kocka sa však nezmenila. NXT kocka sa dá programovať aj v sebe samotnej, pre zložitejšie programy je však nutné využiť prostredie LabVIEW. Jedná sa o grafické blokovo orientované rozhranie podobné pôvodnému prostrediu pre kocku RCX.

Táto LEGO kocka sa vyskytla v sadách:

- 8527 - Mindstorms NXT
- 8547 - Mindstorms NXT 2.0
- 9797 - Mindstorms Education NXT Base Set (Dacta)
- 9841 - NXT Intelligent Brick
- 10287 - Intelligent NXT Brick (Black)

---

<sup>12</sup><https://www.brickowl.com/catalog/lego-spybotics-receiver-assembly>



Obr. 2.13: LEGO NXT Hub <sup>13</sup>

## 2.13 LEGO WeDo 1.0 a WeDo 2.0

Jedná sa o edukačné sady mierené pre dolný stupeň základných škôl. WeDo 1.0 (obr. 2.14) vyšla v roku 2009 [27]. Sada obsahovala základné motory, senzory a WeDo USB Hub. Hub sa programoval v grafickom prostredí WeDo Software. Konektory motorov sú totožné s PoweredUp konektormi. Sada WeDo 2.0 (obr. 2.15) vyšla v roku 2016 [19]. Predstavila novú kocku podobnú kocke City Hub.

Tieto LEGO kocky sa vyskytli v sadách:

- 9580 - WeDo Robotics Construction Set (Dacta)
- 45300 - WeDo 2.0 Core Set (Dacta)
- 45301 - WeDo 2.0 Smart Hub (Dacta)



Obr. 2.14: LEGO WeDo Hub <sup>14</sup>

---

<sup>13</sup><https://www.mamevsechno.cz/Lego-Mindstorms-NXT-Inteligentni-kostka-9841-NPI53788-d304.htm>

<sup>14</sup><https://stavebnice-lego.heureka.sk/lego-9581-wedo-usb-rozhrani-hub/#prehlad/>



Obr. 2.15: LEGO WeDo 2.0 Hub <sup>15</sup>

## 2.14 LEGO Mindstorms EV3

Tretia generácia LEGO Mindstorms (obr. 2.16) vyšla v roku 2013 a predstavila novú EV3 kocku a nové senzory [29]. Kocka je po novom rozšírená o slot pre SD kartu a možnosť pripojenia na internet. Na EV3 kocke je nainštalovaný operačný systém ev3dev, jedná sa o distribúciu OS Linux založenú na DebianOS. Kód sa programuje v jazyku Scratch.

Táto LEGO kocka sa vyskytla v sadách:

- 31313 - Mindstorms EV3
- 45500 - EV3 Intelligent Brick
- 45544 - EV3 Core Set (Dacta)



Obr. 2.16: LEGO EV3 Hub <sup>16</sup>

## 2.15 Lego Powered Up/Control+

Jedná sa o pokračovateľa LEGO Power Functions z roku 2007, to bola sada motorov bez možnosti programovania. Spočiatku sa jednalo o kocky Move hub a City hub, tie vyšli v roku 2017 [5]. Move hub sa programoval cez mobilnú aplikáciu LEGO BOOST. City hub sa programoval cez mobilnú aplikáciu LEGO Powered Up. V roku 2020 pribudla kocka, Technic Hub a s ňou aj mobilná aplikácia Control+, v ktorej sa dala kocka programovať. Programovanie v aplikáciách sa podobalo jazyku Scratch. Kocky sú zobrazené na obrázku

<sup>15</sup><https://www.eduxe.sk/p/301/45301-smart-hub-2-io>

<sup>16</sup>[https://www.detsky-raj.sk/lego-mindstorms-ev3-45544-kocka-pocitac-hub-novy#googtrans\(sk\)](https://www.detsky-raj.sk/lego-mindstorms-ev3-45544-kocka-pocitac-hub-novy#googtrans(sk))

2.17. V roku 2020 bola taktiež predstavená verzia Spike Prime s viacerými portmi pre motory a senzory. Jedná sa však len o edukačnú verziu ktorá nie je voľne dostupná. V roku 2021 vyšla aj jej menšia verzia Spike Essential Hub. Kocky sú zobrazené na obrázku 2.18.

Tieto LEGO kocky sa vyskytli v sadách:

- 17101 - BOOST Creative Toolbox
- 21323 - Grand Piano
- 42099 - 4x4 X-treme Off-Roader
- 42100 - Liebherr R 9800
- 42109 - App-Controlled Top Gear Rally Car
- 42114 - 6x6 Volvo Articulated Hauler
- 42124 - Off-Road Buggy
- 42129 - 4x4 Mercedes-Benz Zetros Trial Truck
- 42131 - Cat D11 Bulldozer
- 45345 - SPIKE Essential Set (Dacta)
- 45601 - Large Hub for SPIKE Prime (Dacta)
- 45609 - Small Hub for SPIKE Prime (Dacta)
- 45678 - SPIKE Prime Set (Dacta)
- 60197 - Passenger Train
- 60198 - Cargo Train
- 60336 - Freight Train
- 60337 - Express Passenger Train
- 71044 - Disney Train and Station
- 75253 - Droid Commander
- 76112 - App-Controlled Batmobile
- 88006 - Move Hub
- 88009 - Hub
- 88012 - Technic Hub

---

<sup>17</sup><https://www.lego.com/en-sk/product/move-hub-88006>

<sup>18</sup><https://www.lego.com/en-us/product/hub-88009>

<sup>19</sup><https://www.lego.com/sk-sk/product/technic-hub-88012>



Obr. 2.17: LEGO Move Hub, City Hub, Techic Hub <sup>17 18 19</sup>



Obr. 2.18: Spike Prime Hub, Spike Essemcial Hub <sup>20 21</sup>

## 2.16 LEGO Mindstorms Robot Inventor

Posledná programovateľná kocka rady LEGO Mindstorms (obr. 2.19). Kocka nie je kompatibilná s predchádzajúcim modelom EV3 ani NXT, ale je kompatibilná s motormi Powered Up/Control+ [5]. V podstate sa jedná o komerčne dostupnú verziu LEGO Spike Prime. Aj napriek tomu, že kocka bola predstavená len v roku 2021, jej podpora už bola ukončená.

Táto LEGO kocka sa vyskytla v sadách:

- 51515 - Robot Inventor
- 88016 - Large Hub



Obr. 2.19: LEGO Robot Inventor Hub <sup>22</sup>

<sup>20</sup><https://www.eduxe.sk/p/355/45601-lego-spike-prime-programovatelny-hub>

<sup>21</sup><https://www.stavebniceprochytredeti.cz/sk/lego-education-45609-spike-prime-maly-hub-924.html>

<sup>22</sup><https://www.amazon.com/LEGO-Technic-Axel-Hubs-Accessory/dp/B00B42IRBC>

## Kapitola 3

# Komunikačné protokoly

Aby mohla kocka LEGO MINDSTORMS EV3 posielat dáta vo forme signálov a aby mohla kocka LEGO Technic Hub spoľahlivo prijať tieto signály spojiť do dát a následne ich interpretovať ako informácie na základe ktorých bude vykonávať ďalšie funkcie, musí byť jasne definované, ako prenos dát prebieha. V tejto kapitole sa preto zameriam na teóriu komunikačných protokolov pre prenos signálov, ich delenie, využitie výhody a nevýhody. Popíšem už existujúce protokoly využívané v kockách LEGO EV3 a LEGO Powered Up a to UART a I2C. V ďalšej kapitole potom využijem získané znalosti pri návrhu vlastného prototypu.

### 3.1 Vlastnosti a delenie komunikačných protokolov využívaných vo vstavaných systémoch

Komunikačný protokol je systém pravidiel ktorí umožňujú prenos signálov a výmenu dát medzi dvoma alebo viacerým zariadeniami prostredníctvom nejakej fyzikálnej veličiny (napätie, prúd, elektromagnetické vlny...). Komunikačný protokol popisuje spôsob ako sa:

- **Reprezentujú dáta** Ako sa zakódujú dáta pre prenos a akým spôsobom sa dekodujú.
- **Synchronizuje odosielanie a príjem dát** Ako sa zariadenia nastaví aby mohli odosielať/prijímať dáta rovnakou rýchlosťou, v rovnaký čas.
- **Detekujú a opravujú chyby** Ako sa implementujú systémy pre detekciu chýb a ich potenciálne ošetrenie.
- **Riadi tok dát** Ako sa spracovávajú dáta, ak sa odosiela viac dát než sa stíha prijať.
- **Adresujú zariadenia a smerujú správy** Ako sa určuje cesta dát po zbernici a ako sa jednotlivé zariadenia na zbernici identifikujú.
- **Dáta zaobalujú do paketov a rámcov** Ako sa posielajú jednotlivé pakety/rámce dát a ako prebieha ich spracovanie [9].

Vo vstavaných systémoch sú komunikačné protokoly niekedy usposobené na mieru daným zariadeniam a ich limitáciám ako výpočtový výkon, odber energie, pamäť. Existuje však rada štandardizovaných protokolov. Rôzni výrobcovia elektronických súčiastok tak nemusia vyvíjať vlastné protokoly a ich zariadenia budú kompatibilné so zariadeniami iných výrobcov, ktorí sa rozhodli využiť rovnaký protokol. Jednotlivé komunikačné protokoly majú rôzne vlastnosti, výhody a nevýhody. Teda sú vhodné na rôzne použitia. Môžeme ich porovnávať podľa niekoľkých základných vlastností:

- **Vonkajšie a vnútorné** Vonkajšie protokoly slúžia pre komunikáciu medzi dvoma oddelenými systémami (UART, USB). Vnútorné protokoly slúžia pre komunikáciu prvkov na jednej obvodovej doske (I2C,SPI,CAN).
- **Synchrónne a asynchrónne** Synchrónne protokoly zabezpečujú synchrónny hodinový signál na všetkých pripojených zariadeniach. Vyžaduje náročnejšiu konfiguráciu ale typycky umožňuje vyššiu prenosovú rýchlosť (I2C, SPI). Asynchrónne protokoly využívajú štart a stop bity na dátovom vodiči pre indikovanie začiatku a konca odosielaného paketu. To sa hodí v prípadoch kde je synchronizácia nemožná alebo je treba šetriť počet využitých pinov (UART).
- **Simplexné, polovične duplexné a plne duplexné** Duplexné protokoly umožňujú posielanie dát oboma smermi zároveň (USART, SPI). Polovične duplexné môžu naraz poselať dáta len jedným smerom (I2C,1-Wire). Simplexné umožňujú prenos dát len jednosmerne (UART využívajúci len jeden vodič).
- **Počet potrebných vodičov** Jednotlivé protokoly vyžadujú rôzny počet dátových vodičov. Protokol 1-Wire vyžaduje len jeden dátový vodič a zem. I2C a UART vyžadujú dva dátové vodiče. SPI vyžaduje 3 až 4 vodiče. Paralelne rozhranie vyžaduje počet vodičov podľa toho, koľko bitová je architektúra zariadenia plus ďalší hodinový vodič.
- **Možný počet master / slave zariadení** Master je zariadenie ktoré posela príkazy a má kontrolu nad zbernicou. Môže byť vybrané z viacerých takýchto zariadení. Slave je zariadenie ktoré vykonáva príkazy a odpovedá na dotazy, zvyčajne má pridelené ID pomocou ktorého sa identifikuje. UART Je priama komunikácia medzi dvoma rovnocennými zariadeniami. SPI podporuje jedno master zariadenie a viac slave zariadení. Protokol CAN nemá pevne stanovený maximálny počet periférií, všetky sú však rovnocenné a záleží na prioritě správy ktorú posielajú.
- **Rýchlosť** Rýchlosť protokolov UART a I2C môže byť v rámci jednotiek Mbps. SPI môže dosahovať niekoľko desiatok Mbps a USB protokol je schopný dosahovať rýchlosť až 20Gbps.

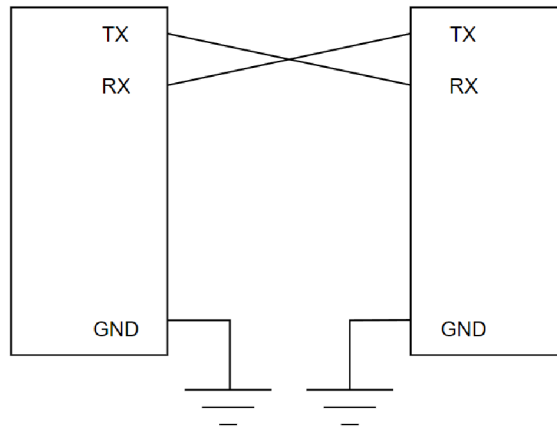
## 3.2 Protokol UART

Univerzálny asynchrónny prijímač vysielateľ (schéma na Obr. 3.1). Je protokol umožňujúci prenos dát medzi dvoma zariadeniami. Jedná sa o jeden z najjednoduchších protokolov pre komunikáciu s perifériami. Pre obojsmernú komunikáciu vyžaduje 2 dátové vodiče a spoločnú zem. Komunikácia pomocou UART môže byť:

- **simplexná** - využíva sa len jeden dátový vodič TX-RX,
- **polovične duplexná** Využívajú sa dva dátové vodiče TX-RX a RX-TX ale dáta sa môžu poselať v jeden čas len jedným smerom,
- **plne duplexná** Dáta sa môžu poselať naraz oboma smermi.

Keďže sa jedná o asynchrónny protokol, nie je nutné aby mali zariadenia spoločný hodinový signál. Obe zariadenia však musia prijímať/vysielateľ dáta na dopredu stanovenej rýchlosti (typické prenosové rýchlosti v baudoch(Bd) sú 4800, 9600, 19.2K, 57.6K alebo





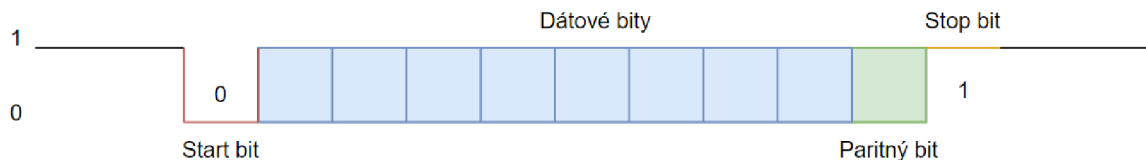
Obr. 3.1: Schéma zapojenia UART

115.2K). Výhodou UART je jeho jednoduchá implementácia a nízka cena. Nevýhodou je nižšia prenosová rýchlosť rýchlosť.

Dáta sa posielajú zabalené v rámcoch [10]. Rámec pozostáva zo Štart bitu 5-9 Dátových bitov nasleduje 1 voliteľný paritný bit a 1-2 Stop bity.

#### Význam jednotlivých bitov (obr.3.2)

- **Štart a Stop bity** Slúžia na detekciu začiatku a konca prenosu jedného rámca. Hodnota Štart bitu je opačná ako pokojová hodnota na dátovom vodiči, hodnota Stop bitu je rovnaká ako pokojová hodnota na dátovom vodiči.
- **Dátové bity** Sú užitočné bity štandardne posielené od najmenej významného po najviac významný.
- **Paritný bit** Je voliteľný a slúži pre detekciu chýb v rámci. Existujú dva typy parity, a to párna parita, a nepárna. Pri párnej parite je hodnota paritného bitu taká aby súčet všetkých jednotiek v Dátových bitoch spolu s Paritným bol párný. Nepárna parita funguje opačne, súčet jednotiek v dátových bitoch spolu s paritným bitom musí byť nepárny. Ak by pri kontrolnom súčte bolo toto pravidlo porušené, indikovalo by to chybu v prenose.



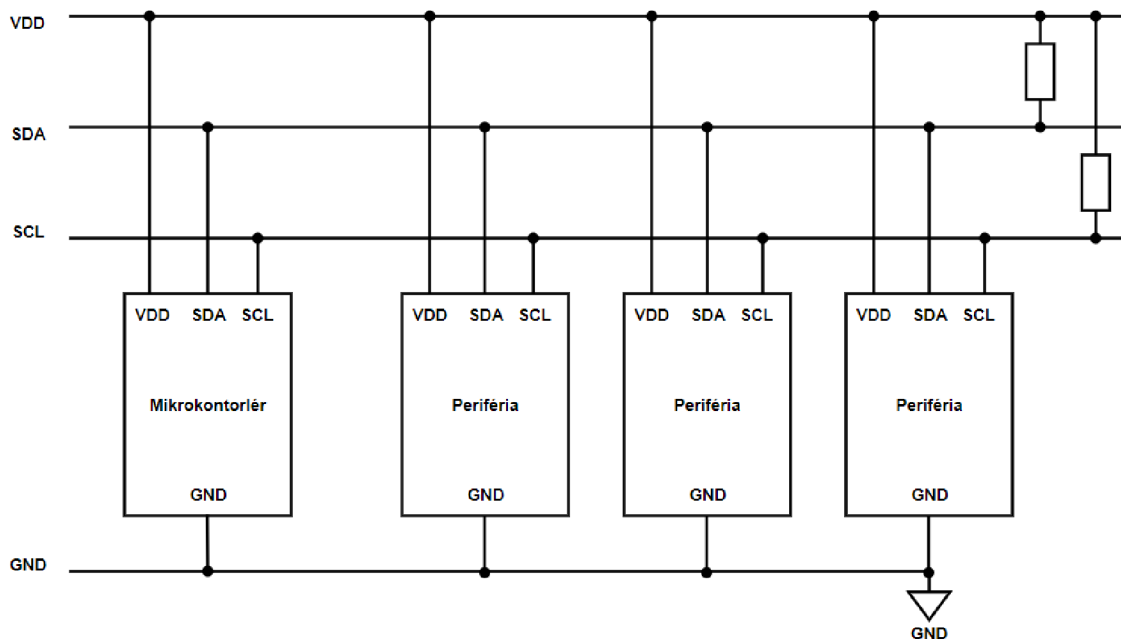
Obr. 3.2: Grafické znázornenie jedného rámca v protokole UART.

### 3.3 Protokol I2C

Vnútroň Integrovaný Obvod (Inter Integrated Circuit) je synchronný komunikačný protokol schopný prenášať dáta po zbernici, na ktorú je pripojené jedno Master zariadenie

a umožňuje viac ako jedno Slave zariadení [30]. Jedná sa o polovične duplexný protokol využívajúce dva dátové vodiče, jeden pre prenos samotných dát (SDA), druhý pre prenos hodinového signálu (SCL). Zariadenia musia mať taktiež spoločný zdroj napätia a zem (schéma na obr.3.3).

- **Master** Je mikroprocesor riadiaci komunikáciu pomocou hodinového signálu.
- **Slave** Je periféria pripojená na zbernicu, má unikátne zvyčajne 7 alebo 10 bitové ID.
- **Zbernica** Sú vodiče SDA a SCL na ktoré sú pripojené všetky zariadenia. Vodiče sú pripojené na Zdroj napätia cez Pull Up rezistor.

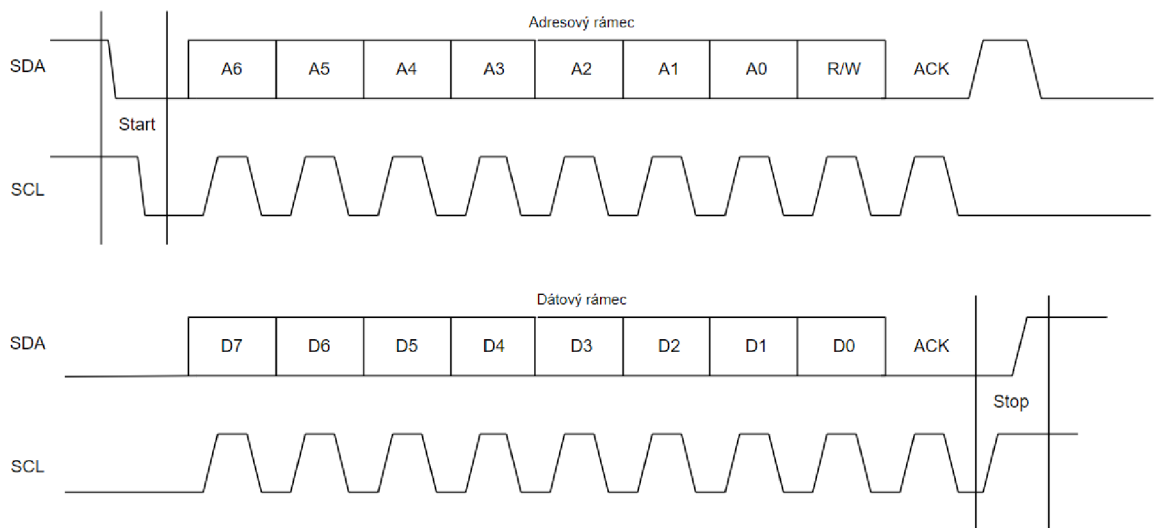


Obr. 3.3: Schéma zapojenia zariadení na I2C zbernicu

#### Posielanie rámca (obr. 3.4)

- **Start** Začiatok dátového prenosu sa signalizuje tak, že SDA sa nastavený na 0 a následne SCL sa prepne na 0. Akonáhle sú obe hodnoty 0 začnú sa vysielat a prijímat dáta.
- **Adresový rámec** Obsahuje 7-bitovú adresu periférie s ktorou chce Master komunikovať. Bity sa posielajú od najvýznamnejšieho po najmenej významný. Ôsmy bit slúži na nastavenie módu, ake má hodnotu 1 chce Master z periférie čítať, ak 0 chce zapisovať. Nasleduje deviaty bit indikujúci, či prenos prebehol úspešne. SDA sa nastaví na 0 pokiaľ je SCL 1 ak je prenos úspešný (ACK), ostane 1 ak je neúspešný (NACK). To indikuje, či bola adresa prijatá.
- **Dátový rámec** Po úspešnom odoslaní adresového rámca nasleduje jeden alebo viac dátových rámcov. Tie obsahujú 8 dátových bajtov a deviaty kontrolný, značiac či boli dáta prijaté, či už Mastrom alebo perifériou podľa nastaveného módu.

- **Stop** Po odvysielaní všetkých dát sa komunikácia ukončí nastavením SCL na 1 a následne SDA na 1. Akonáhle sú obe hodnoty 1, prenos je ukončený.



Obr. 3.4: Grafické znázornenie jedného rámca v I2C protokole

## Kapitola 4

# Technické parametre LEGO Mindstorms EV3 a LEGO Technic Hub, a ich periférie

V tejto kapitole sa zameriavam na LEGO Programovateľné kocky, s ktorými budem pracovať. Popíšem ich HV vlastnosti a možnosti programovania.

### 4.1 Technické parametre kocky LEGO Mindstorms EV3

Kocka LEGO Mindstorms EV3 je treťou generáciou robotického setu LEGO Mindstorms. Vydaná bola v roku 2013 a nahradila druhú generáciu LEGA Mindstorms kocku NXT. Rovnako ako NXT napájaná šiestimi 1,5V AA batériami. Hlavným rozdielom od starej NXT kocky je novší procesor [8]. Zatiaľ čo NXT využíva ARM7 mikroprocesor, EV3 má silnejší ARM9, na ktorom beží operačný systém Linux, konkrétne ev3dev, operačný systém založený na Debiane. EV3 taktiež obsahuje USB konektor a Micro SD slot až do 32GB. EV3 kocka má taktiež možnosť Wi-Fi pripojenia a ako bolo už v NXT generácii, Bluetooth pripojenie. EV3 má oproti predchádzajúcemu modelu aj kvalitnejší display, z 100 x 64 pixlov Monochromatického LCD na 178 x 128 pixlov Monochromatického LCD. Pamäť EV3 je rozšírená z 64 KB RAM a 256 KB Flash na 64 MB RAM a 16 MB Flash. Hlavný procesor je TI Sitara AM1808 @300 MHz. EV3 má 8 portov, 4 pre motory a 4 pre senzory. Konektory ostávajú rovnaké ako v generácii NXT a tak sú aj staré periférie kompatibilné, EV3 senzory však nefungujú na NXT.

Oficiálne periférie vydané pre EV3 :

- Dotykový senzor
- Svetelný a farebný senzor
- Ultrazvukový senzor vzdialenosti
- Gyroskopický senzor
- Infračervený senzor
- Infračervený vysielateľ
- Veľký motor

- Stredný motor

Staršie kompatibilné NXT periférie:

- Interaktívny servo motor
- Zvukový senzor
- Svetelný senzor
- Dotykový senzor
- Ultrazvukový senzor
- Lampa

EV3 nepodporuje programovanie v NXT programovacím softvéri ale programuje sa v prostredí LabVIEW v grafickom jazyku G. Taktiež bola vyvinutá možnosť programovať EV3 v jazyku MicroPython v prostredí Visual Studio Code [29]. Pre možnosť programovať v jazyku MicroPython je potrebné nahráť na SD kartu špeciálny firmvér EV3 MycroPython Image. SD karta potom nabojuje firmvér pre programovanie vo VS Code. Taktiež je potrebné stiahnuť rozšírenie pre VS Code LEGO Mindstorms EV3 MicroPython.

## 4.2 Technické parametre kocky LEGO Technic Hub

Kocka vyšla v roku 2020 v setoch Control+ a PoweredUp. Kocka je napájaná šiestimi 1,5 V AA batériami. Má štyri vstupno-výstupné porty. Tlačidlo pre spustenie. Mikroprocesor STM32L431RCT6 s jadrom ARM CORTEX-M4 a 256 kB ROM [13]. Pre Bluetooth spojenie využíva mikroprocesor CC2640 32-bit ARM CORTEX-M3. Technik hub sa primárne používal pre ovládanie cez Bluetooth s mobilnou aplikáciou Control+. Tá podporovala aj grafický programovací jazyk ako väčšina aplikácií pre podobné LEGO kocky. Pre programovanie na PC sa využíva open-source webové prostredie PyBricks. To podporuje programovanie v jazyku Micro Python a má rozsiahlu dokumentáciu. Pre prenos kódu do kocky taktiež využíva Bluetooth rozhranie.

Kocka Technic hub podporuje všetky periférie s Powered Up konektorom:

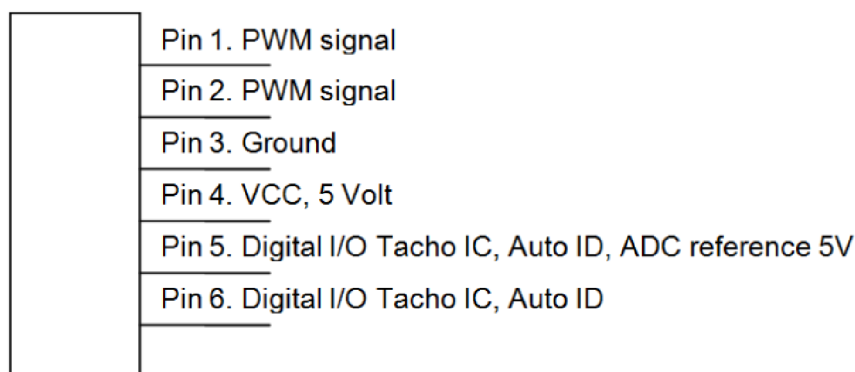
- Motory bez senzoru natočenia
- Motory so senzorom natočenia
- Senzor náklonu
- Infračervený senzor
- Senzor farby, osvetlenia a vzdialenosti so vstavanou LED
- Senzor farby
- Ultrazvukový senzor
- Senzor sily
- 3x3 Maticový displej
- Svetlá
- Diaľkové ovládanie [8]

## 4.3 Konektor a port EV3, a Powered Up

Aj keď obe kocky pracujú s podobnými 6 žilovými káblami, konektory sa líšia. Tu sa pozriem na oba konektory a účel jednotlivých vodičov rozhrania portov.

### 4.3.1 Porty EV3 kocky

Kocka EV3 má porty totožné s predchádzajúcou generáciou LEGO Mindstorms NXT a NXT 2.0. A teda podporuje aj ich staršie periférie. Kocka EV3 má 4 výstupné porty určené pre ovládanie aktuátorov (schéma na obr 4.1). 6 žilové rozhranie navyše umožňuje aby aktuátory posielali dáta naspäť do EV3 kocky bez toho aby museli použiť ďalší prijímací port.



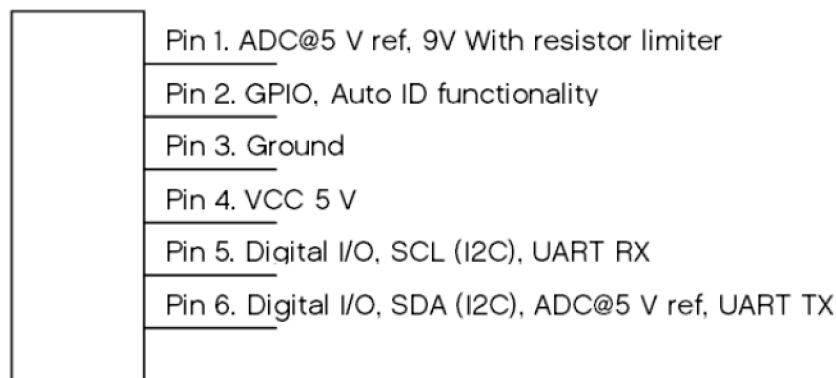
Obr. 4.1: Nastavenie pinov implementované na výstupných portoch pre aktuátory na kocke LEGO MINDSTORMS EV3<sup>23</sup>

Na pinoch Pin1 a Pin2 je posielaný PWM signál pre ovládanie aktuátorov. Tieto signály sú ovládané interným ovládačom motora ktorý dokáže nepretržite napájať každý port 700mA až 1A. Vďaka výstupnému PWM signálu môžu byť motory brzdené alebo uvedené do volnobehu. Porty sú ochránené voči skratovaniu a prehriatiu, ak je zistený príliš veľký dober z kocky, ovládač motora automaticky reguluje výstupný prúd. Pin3 slúži pre uzemnenie kocky. Pin4 slúži na napájanie kocky napätím 5V. Pin5 a Pin6 slúžia ako vstupné piny. Slúžia na auto detekciu pripojeného aktuátora a taktiež ako vstup pre dáta ako napríklad otáčky motora, natočenie osy a iných údajoch ktoré môže motor posielat naspäť do hubu. Pin5 je taktiež pripojený k 10-bitovému AD prevodníku čo umožňuje načítavať analógové hodnoty ak treba.

Ďalej má kocka EV3 ešte 4 vstupné porty. Ich hlavná funkcionality je umožniť systémom reagovať na prostredie prostredníctvom odozvy zo senzorov (schéma na obr 4.2. 6 káblové rozhranie je implementované tak, aby umožnilo externým zariadeniam posielat dáta naspäť do EV3 kocky rôznymi spôsobmi. Systém umožňuje posielat dáta vo forme analógových hodnôt, I2C komunikácie alebo UART komunikácie.

Ako sa dá vyčítať na obrázku vyššie, piny sa dajú využiť rôznymi spôsobmi podľa požadovanej funkcionality alebo kontextu. Pin1 podporuje načítavanie analógových hodnôt alebo podporuje senzory ktoré vyžadujú vyššie napätie. Pin2 slúži na auto detekciu. Po rozpoznaní pripojenej periférie sa automaticky nastaví piny Pin5 a Pin6 na požadovanú funkcionality.

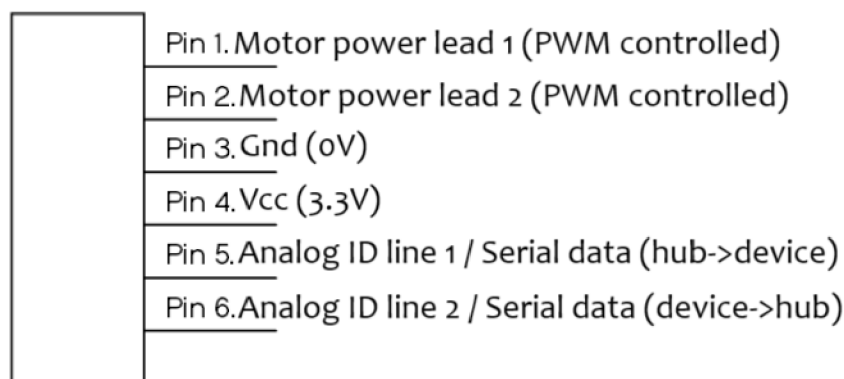
<sup>23</sup><https://www.amazon.com/LEGO-Technic-Axel-Hubs-Accessory/dp/B00B42IRBC>



Obr. 4.2: Nastavenie pinov implementované na vstupne-výstupných portoch pre senzory na kočke LEGO MINDSTORMS EV3 <sup>23</sup>

### 4.3.2 Powered Up porty

Pod pojmom Powered Up sa rozumejú všetky zariadenia využívajúce konektor aký je znázornený na obrázku (obr). Narozdiel od EV3, modernejšie Powered Up zariadenia nerozlišujú medzi vstupnými a výstupnými portmi (schéma na obr. 4.3). Využívajú jeden typ 6 káblového rozhrania pre aktuátory aj pre senzory.



Obr. 4.3: Nastavenie pinov implementované na vstupne-výstupných portoch pre periférie na kočke LEGO Control+ Technic Hub [12]

Podobne ako pri EV3 kočke, piny Pin1 a Pin2 slúžia na ovládanie motorov pomocou PWM signálov. Pin3 slúži pre uzemnenie periférie. Pin4 slúži pre napájanie periférie (3.3V alebo 5V) Piny Pin5 a Pin6 slúžia pre analógovú identifikáciu alebo ako sériová linka. Pin5 slúži pre prenos dát z hubu do periférie (TX), Pin6 prenáša dáta z periférie do hubu (RX). Pri prenose sa využíva UART protokol.

## 4.4 Identifikácia periférií

Každá existujúca periféria pre obe LEGO kocky sa musí najskôr identifikovať aby sa mohla používať. Na identifikáciu periférie slúžia piny Pin5 a Pin6 pri oboch kockách. Pre niektoré jednoduchšie periférie stačí nastaviť istú hodnotu napätia na dané porty (tabuľka 4.1). Pre

komplikovanejšie alebo modernejšie periférie sa následne využíva sériová linka, protokol UART alebo I2C (I2C len pri EV3).

Periféria	Napätie
LEGO MINDSTORMS EV3 Large motor	Pin6 Vcc a Pin5 približne 2.0V Pin6 Gnd a Pin5 približne 1.6V
LEGO MINDSTORMS EV3 Medium motor	Pin6 Vcc a Pin5 približne 0.45V Pin6 Gnd a Pin5 približne 0.25V
LEGO MINDSTORMS NXT motor	Pin6 Vcc a Pin5 približne 0.45V Pin6 Gnd a Pin5 približne 0.25V

Tabuľka 4.1: Tabuľka napätí na pinoch Pin5 a Pin6 pre EV3 a NXT motory[18]

Periféria	Napätie
LEGO MINDSTORMS I2C zariadenie	Pin2 Gnd a Pin5 Vcc Pin6 Vcc, vyžaduje ďalšiu komunikáciu
LEGO MINDSTORMS NXT svetelný senzor	Pin2 Gnd a Pin5 Gnd
LEGO MINDSTORMS NXT dotykový senzor	Pin2 Gnd a Pin1 menej ako 100mV
LEGO MINDSTORMS NXT dotykový senzor	Pin2 Gnd a Pin1 menej ako medzi 850mV - 950mV
LEGO MINDSTORMS NXT dotykový senzor	Pin2 Gnd a žiadna z vyšších variant
LEGO MINDSTORMS EV3 digitálny senzor	Pin2 Vcc a Pin1 menej ako 100mV
LEGO MINDSTORMS EV3 jednoduchý senzor	Pin2 Vcc a Pin1 medzi 100mV - 3100mV
LEGO MINDSTORMS NXT I2C teplotný senzor	Pin2 Vcc a Pin1 viac ako 4800mV a Pin6 Vcc

Tabuľka 4.2: Tabuľka napätí na pinoch pre EV3 a NXT senzory[18]

Periféria	Zapojenie
Powered Up jednoduchý motor / Wedo 2 motor	ID1 - Gnd cez 2.2kOhm rezistor, ID2 - Gnd
Powered Up vlakový motor	ID1 - Vcc , ID2 - Gnd
LED	ID1 - Gnd cez 2.2kOhm rezistor, ID2 - nezapojený

Tabuľka 4.3: Tabuľka pre identifikáciu vybraných Powered Up prvkov [12]

## 4.5 Možnosti prenosu dát medzi kockami

Tu sa zameriam na jednotlivé možnosti prenosu dát. Popíšem výhody a nevýhody jednotlivých variant.



### 4.5.1 Využitie LEGO periférií

Jednou z možností, ako posielat dáta medzi LEGO kockami je využiť už existujúce LEGO periférie.

#### Prepojenie dvoch motorov

Jednou z prvých možností bol prenos dát pomocou prepojenia dvoch motorov oskou. Aj keď by sa jednalo o veľmi primitívne riešenie, umožnilo by obojsmerný prenos dát. Dáta by sa posielali otáčaním jedného z motorov, a to by otáčalo aj druhý motor, z ktorého by sa čítalo jeho natočenie. Nevýhodou tohto riešenia je však maximálna možná rýchlosť prenosu a presnosť prenesených dát. Taktiež LEGO motory nie sú usposobené na takéto používanie a hrozilo by ich poškodenie.

#### Prepojenie pomocou Bluetooth rozhrania

Keďže obe kocky podporujú komunikačné rozhranie Bluetooth, dáva zmysel využiť ho ku vzájomnej komunikácii. Nezabral by sa žiaden port, a umožnilo by to aj komunikáciu na väčšie vzdialenosti. Tieto kocky však nepodporujú priame spárovanie medzi sebou. Bolo by však možné využiť externé Bluetooth moduly a ovládať ich pomocou Arduina.

#### Prepojenie pomocou optického rozhrania

Ak by sme využili optické rozhranie pre prenos dát, nehrozilo by žiadne riziko poškodenia. Na jednej strane by bola LED napojená na EV3 a na druhej, senzor čítajúci dáta. Nebolo by ťažké vytvoriť funkčný prototyp, keďže existujú LEGO varianty prijímača aj vysieláča.

### 4.5.2 LEGO komunikačný protokol

Pre komunikáciu s pokročilejšími perifériami využívajú kocky EV3 a Technic Hub sériovú linku a protokol UART (EV3 pre niektoré periférie využíva aj protokol I2C) [18]. Komunikácia má dve fázy, úvodné nadviazanie spojenia a následná výmena dát. Na pozorovanie tejto komunikácie som použil logickú sondu a sledoval som piny Pin5 a Pin6 na senzore LEGO BOOST senzor farby a vzdialenosti.

#### Nadviazanie spojenia

Po zapnutí, kocka periodicky posiela signály a čaká až sa pripojí periféria. Po zapojení periférie do konektoru sa nastaví logická hodnota na piny Pin6 na log. 1 a po približne 7ms sa nastaví späť na 0. Po približne 400ms sa odošle do senzora správa s nastavením rýchlosti komunikácie po nadviazaní. Táto správa sa pošle rýchlosťou 115200Bd. Následne sa po 100ms pošlú dáta z periférie do kocky obsahujúce nastavenia komunikácie. Po prenose všetkých dát sa odošle bajt 0x04 do kocky. Ak bolo zariadenie rozpoznané a komunikácia bola nadviazaná, odošle sa správa 0x04 do periférie, rýchlosť prenosu sa prepne na 115200Bd a pošle sa správa o nastavení módu do periférie [31].

#### Po nadviazaní spojenia

Po nadviazaní komunikácie sa rýchlosť prenosu prepne na 115200Bb [31]. Hub posiela správy so periférie a podľa toho očakáva odpoveď. Ak formát odpovede neodpovedá očakávanému formátu, alebo kontrolná suma nesedí, hub to zaznamená ako chybu. Po šiestich chybách

hub ukončí komunikáciu a znova sa prepne do fázy nadväzovania komunikácie. Pomocou logického analyzátora som sledoval aké dáta sa prenášajú po pinoch. Z meraní som zistil nasledujúce informácie:

- Bajt 0x02 sa posiela každých 100ms a očakáva sa odpoveď o stave jednotlivých senzorov podľa nastaveného módu,
- Bajt 0x43 a mód sa posiela ak hub chce vyčítať hodnotu zo senzora. Základný mód ktorý sa nastaví vždy po úspešnom nadviazaní komunikácie je
- 0x06 ten signalizuje, že hub sa pýta na farbu na senzore farby alebo odraz svetla od okolia do daného senzoru.
- 0x04 slúži pre načítanie intenzity svetla zo senzora
- 0x01 vráti vzdialenosť objektu od senzora
- Bajt 0x46, mód a hodnota sa posiela ak hub chce nastaviť hodnotu LED na senzore. Nastaviť sa dajú hodnoty
  - 0x00 LED nesvieti
  - 0x03 modrá
  - 0x05 zelená
  - 0x09 červená

### **Využitie Arduina pre imitáciu periférií**

Ďalšou možnosťou, ako prenášať dáta medzi kockami je využiť komunikačný protokol medzi kockami a ich perifériami. Niektoré periférie ako napríklad motory, umožňujú odosielať aj prijímať dáta. Ak by sa mi podarilo pomocou Arduina imitovať ich chovanie počas komunikácie, mohol by som prenášať dáta obojsmerne na jednom porte, poprípade aj vyššími rýchlosťami ako pri ostatných riešeniach.“

## Kapitola 5

# Návrh a tvorba prototypu s využitím optického rozhrania

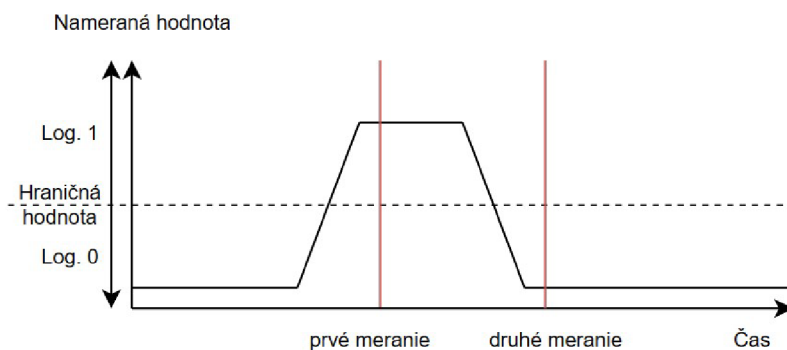
V tejto kapitole popíšem návrh fyzického a softvérového rozhrania pre odosielanie dát a následnú tvorbu prototypu využívajúcu oficiálne periférie alebo ich jednoduché imitácie. Konkrétne použijem LED pre odosielanie dát a LEGO BOOST senzor farby a vzdialenosti pre ich príjem. To mi umožní zamerať sa na programovanie samotného protokolu. Komunikácia bude prebiehať len jednosmerne z EV3 do Technic Hub. Postupne popíšem akým spôsobom sa budú prenášať jednotlivé signály, ako sa dekodujú na bity a bajty, ako bude fungovať signalizácia prenosu dát ako sa tieto dáta dekodujú na jednotlivé inštrukcie.

### 5.1 Prenos bitu

Bit je najmenšia jednotka počítačových dát a môže mať hodnotu 1 alebo 0. Ak chcem poslať jeden bit dát cez optické rozhranie, musím tieto hodnoty reprezentovať fyzickou formou. Teda LED svieti, ak je hodnota bitu 1 a nesvieti, ak je hodnota 0. Aby som odvysielanú hodnotu zachytili, potrebujem svetelný senzor. Ten vie zistiť, či LED svieti alebo nesvieti. Senzor však nevracia hodnotu 1 alebo 0. Posiela rôzne hodnoty podľa osvetlenia priestoru. Aby som vedel určiť, či sa jedná o log. 1 alebo 0, nastavím hraničnú hodnotu. Ak senzor nameria hodnotu vyššiu, ako je táto hraničná hodnota, bude sa jednať o logickú hodnotu 1, a ak bude nameraná hodnota nižšia, bude sa jednať o logickú hodnotu 0 (znázornené na obr. 5.1).

### 5.2 Prenos bajtu

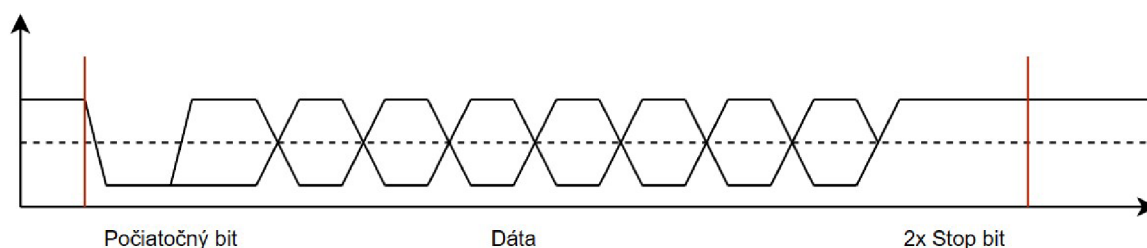
Bajt je základná jednotka počítačových dát a pozostáva z ôsmich bitov. Aby som dokázal preniesť bajt informácií, musím odoslať osem bitov za sebou. Zachytiť takýto reťazec dát však nie je tak jednoduché. Teda, ak chcem preniesť **1111 1111** prejaví sa to, ako neustále svietenie a pokiaľ LED v pokojovom stave svieti, nezaregistruje sa žiadna zmena. To sa dá vyriešiť tak, že pošlem počiatočný bit. Ten bude mať opačnú hodnotu ako pokojový stav. Ak bude pokojový stav 1, tak počiatočný bit bude mať hodnotu 0. Tak nastavím prijímač na čítanie. Po načítaní ôsmich bajtov sa však musí vysielateľ naspäť vrátiť do pokojového stavu. Ak by sa totiž odvysielal bajt končiaci 0 a vysielateľ by sa nevrátil do pokojového stavu, začal by sa okamžite načítať ďalší bajt. Vždy po odvysielaní bajtu informácií sa teda odošle aj dva ukončujúce bity aby sa LED dostala do pokojovej hodnoty pred tým ako začne



Obr. 5.1: Načítanie bitu

Na grafe môžeme pozorovať, že počas prvého merania je nameraná hodnota reprezentovaná ako log. 1 a počas druhého merania, ako log. 0.

odosielanie ďalšieho bajtu. Graf na obrázku 5.2 znázorňuje formát, v ktorom sa odosiela jeden bajt.



Obr. 5.2: Odosielanie bajtu

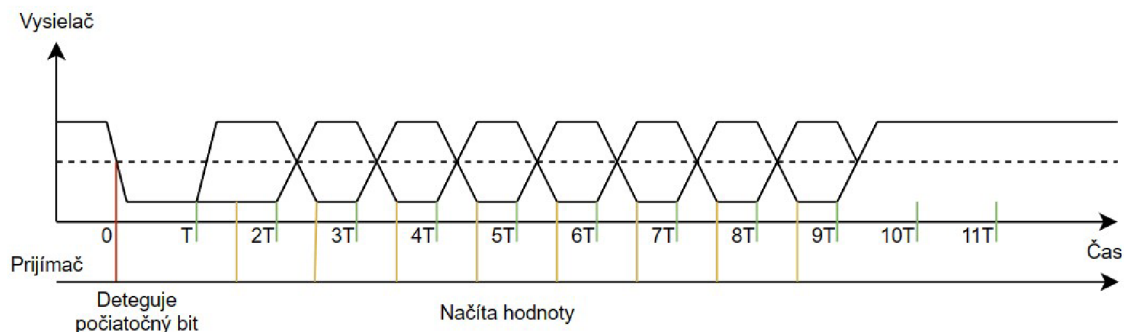
### 5.2.1 Časovanie prenosu

Ak je senzor pripravený načítať dáta, nemôže iba uložiť do bajtu prvých osem hodnôt, ktoré načíta. Je nutné, aby bol synchronizovaný s vysielačom. Musí teda vedieť, s akou frekvenciou pracuje vysielač (LED). Ak vysielač pracuje s frekvenciou 100 Hz, (jeho perióda je  $100^{-1}\text{s}$ ) znamená to, že jeden bit sa bude zobrazovať 10 ms. Počas tohto času môže senzor nasnímať hodnotu tohto bitu. Nie je však bezpečné, aby senzor načítaval hodnotu bitu hneď, ako je zobrazená, a to z niekoľkých dôvodov. Buď môže byť snímač a vysielač mierne nesynchronizovaný a prečítala by sa ešte stará hodnota, alebo by senzor ešte nezobrazoval presnú hodnotu z dôvodu, že senzor nie je dostatočne presný, alebo sa LED pomaly rozsvetuje. Najväčšia presnosť by sa mala dať dosiahnuť ak sa načíta hodnota v strede zobrazovacieho cyklu.

### 5.2.2 Proces načítania bajtu

Povedzme, že vysielač pracuje na frekvencii 100 Hz, teda perióda je 10 ms a chce odoslať hodnotu **1001 0110**, teda binárne reprezentované číslo 150. Celá vysielaná správa však bude rozšírená o operačné bity. Jeden počiatkový bit 0 a dva ukončovúce bity 1. Výsledný tvar správy bude vyzeráť takto: **0 1001 0110 11**. Tým pádom sa spolu odvysliela 11 bitov

v pomere 8:3 dátových ku režijným bitom. Ak sa každý bit prenáša 10 ms, celá správa sa preniesie za 110 ms. Prijímanie prebehne nasledovne (znázorňuje graf na obr. 5.3. Akonáhle senzor načíta logickú hodnotu 0 (teda hodnotu nižšiu ako je mnou nastavená hraničná hodnota), pripraví sa na načítanie správy. Počká 10 ms, až prejde počiatočný bit, a následne počká ďalších 5 ms, aby sa dostal do stredu vysielacej periódy (tam je hodnota na senzore najstabilnejšia). Načíta sa prvý bit prenášanej správy. Prijímač počká 10 ms a opäť sa ocitne v strede periódy. Tak sa načíta všetkých osem dátových bitov prenášanej správy. Potom sa už len počká až sa odvysielajú ukončovacie bity. Tým sa senzor nastaví do pokojového módu a opäť čaká na počiatočný bit ktorý signalizuje vysielanie novej správy.

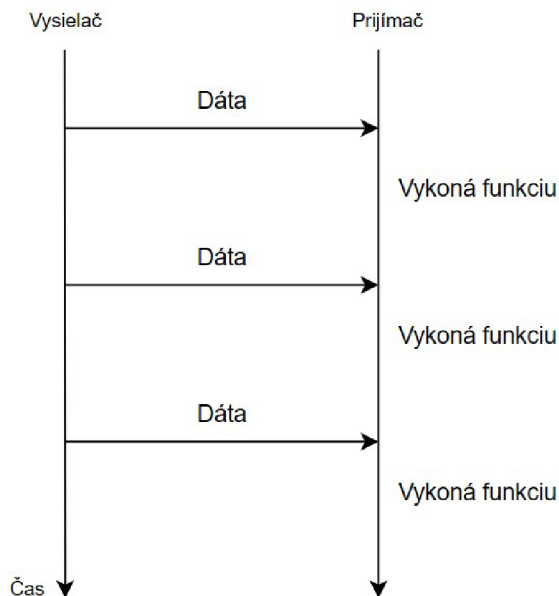


Obr. 5.3: Načítanie bajtu

Graf znázorňuje odvysielanie jedného bajtu a jeho načítanie prijímačom.

### 5.3 Uživateľsky prívetivé odosielanie viac bajtových práv

Ak dokážem odoslať jeden bajt, odoslať viac bajtov za sebou nebude problém. Vysielač ich dokáže odvyselať a prijímač ich dokáže zachytiť. Dostávam sa však na úroveň, na ktorej bude pracovať užívateľ. Navrhmem teda funkcie, ktoré užívateľovi uľahčia prácu s optickým rozhraním. Užívateľ bude schopný zavolať funkciu a priradiť jej hodnoty, ktoré chce odvyselať. Na strane prijímača vytvorím rozhodovací strom, ktorý spracuje prijaté bajty, vyhodnotí ich význam a vykoná požadovanú funkciu. Obr 5.4 znázorňuje prenos dát medzi vysielačom a prijímačom.



Obr. 5.4: **Prenos Dát**

Odosielané dáta obsahujú jedno-bajtový kód funkcie ktorú má prijímač vykonať a jej parametre.

### 5.3.1 Prenos hodnôt

Prenos bajtu mi umožňuje naraz preniesť hodnoty 0 až 255. To stačí na údaje ako kód inštrukcie, ktorú chce užívateľ zavolať, motor na ktorom chce funkciu vykonať, atď. Nestačí to však na prenos hodnôt ako rýchlosť motoru, či nastavenie jeho uhlu natočenia. Hodnoty sa preto prenášajú v dvoch bajtoch. To síce predĺži dobu prenosu správy, zachová to však jednoduchosť návrhu.

### 5.3.2 Inštrukcie

Mojou prvoradou úlohou je umožniť ovládať motory LEGO Technic Hub pomocou LEGO EV3. Navrhmem a naprogramujem inštrukcie, ktoré umožnia užívateľovi vykonávať operácie nad Technic Hub motormi. Keďže však komunikácia je len jednosmerná z EV3 do Technic Hub, nebude možné pracovať s návratovými hodnotami. V tabuľke 5.1 sú vymenované základné inštrukcie pre ovládanie motorov. Ja budem implementovať inštrukcie vysielачa a po tom čo prijímač zachytí hexa kód tejto inštrukcie, vykoná príslušnú Pybricks funkciu na prijímači. Hexa kódy inštrukcií som určil náhodne, snažil som sa ale využiť čo najrôznejšie hodnoty aby som prípadne počas testovania odhalil možné problémy pri rýchlom prepínaní LED. Inštrukcie sú len na prácu s motorom, keďže LEGO Powered Hub nepodporuje iné aktuátory (možné rozšírenie o LEGO senzor farby a vzdialenosti, ktorý má LED s nastaviteľnou RGB farbou).

No.	Inštrukcia vysielača	funkcia v Pybricks	Hexa kód	Popis
1.	technicMotorRun()	run()	0xfe	Motor beží konštantnou rýchlosťou
2.	technicMotorRunTime()	run_time()	0xde	Motor beží konštantnou rýchlosťou po určitý čas
3.	technicMotorRunAngle()	run_angle()	0xc7	Motor beží pri konštantnej rýchlosti pod daným uhlom
4.	technicMotorRunTarget()	run_target()	0x3b	Motor beží konštantnou rýchlosťou smerom k danému cieľovému uhlu
5.	technicMotorStop()	stop()	0x22	Motor sa zastaví a voľne sa točí
6.	technicMotorBrake()	brake()	0x15	Motor sa pasívne zastaví
7.	technicMotorHold()	hold()	0x10	Motor sa zastaví a aktívne sa drží na momentálnom uhle

Tabuľka 5.1: **Tabuľka inštrukcií**

## 5.4 Galvanické oddelenie obvodov s rôznym zdrojom napätia

Keďže sa snažím posielat signály z jednej kocky do druhej. Musím bezpečne prenášať signály medzi dvoma uzavretými obvodmi. Na to sa štandardne využíva galvanické oddelenie. To znamená, že obvody nie sú prepojené priamo vodičom. Pri rôznych hodnotách napätia, prúdu a miestach uzemnenia, hrozí poškodenie súčastí alebo skreslenie prenášaných dát. Využívajú sa preto iné metódy ako bezpečne prenášať signály z jedného obvodu do druhého.

### 5.4.1 Optočlen

Skladá sa z LED, často emitujúcej infračervené žiarenie, na jednej strane izolačnej bariéry. Na druhej strane svetlo citlivý polovodič (fotodióda alebo foto rezistor). LED emituje žiarenie, keď cez ňu prechádza elektrický signál, to je následne detekované polovodičom na druhej strane optočlena. Signál sa teda prenáša pomocou žiarenia bez priameho prepojenia.

- Výhody použitia optočlenov sú:
  - kompaktné a ľahké balenie,
  - odolné voči vonkajšiemu šumu.
- Nevýhody použitia optočlenov sú:
  - LED časom stráca na intenzite,

- obmedzená rýchlosť prepínania medzi stavmi [14].

### 5.4.2 RF

Využitie rádiových frekvencií môže byť využité obdobným spôsobom ako optočlen. Najvýznamnejší rozdiel je vo vlnovej dĺžke prenosového signálu. Signál sa však musí najskôr modulovať do formy v ktorej sa dá odvysielať. Na druhej strane sa následne signál demoduluje.

- Výhody využitia rádiového vysielača a prijímača sú:
  - účinné na väčšie vzdialenosti,
  - vysoká rýchlosť prepínania medzi stavmi.
- Výhody využitia rádiového vysielača a prijímača sú:
  - regulácie spojené s RF emisiami,
  - možné rušenie inými zdrojmi RF [14].

### 5.4.3 Magnetické oddelenie

Aj keď oddelenie transformátormi je bežnejšie pri striedavých prúdoch, existujú techniky ako posielat digitálny signál za pomoci cievok a indukcie. Prijímač sleduje stúpajúce a klesajúce hrany vstupného signálu za pomoci magnetického oddelenia a následne signál dekóduje na normálne logické prechody.

- Výhody využitia transformátorov sú:
  - vhodné aj pre vyššie napätia,
  - vysoká prenosová rýchlosť.
- Nevýhody využitia transformátorov sú:
  - váha a rozmery sú podstatne vyššie ako pri ostatných technológiách,
  - náchylné na rušenie inými zdrojmi elektromagnetických polí (motory) [14].

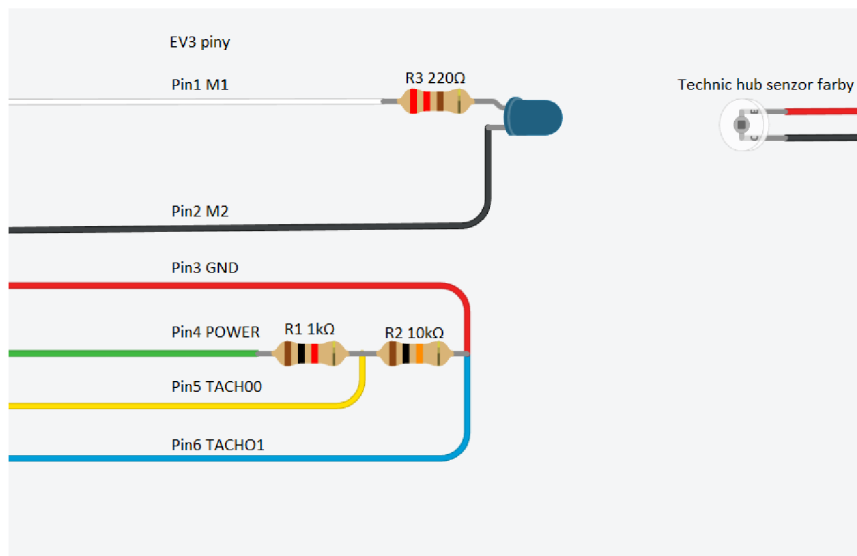
## 5.5 Prenos dát pomocou LEGO periférií

V tejto sekcii popíšem tvorbu prototypu pre odosielanie a prijímanie dát za pomoci prvkov, ktoré mám k dispozícii. Konkrétne použijem LED pre odosielanie dát a LEGO senzor farby a vzdialenosti pre príjem.

### 5.5.1 Vysielač

Aby som mohol odosielať dáta cez LED potrebujem pripojiť LED na EV3. Keďže nemám žiadnu LED z EV3 konektorom, budem musieť napojiť obyčajnú LED na EV3 kábel, aby som dosiahol čo najvyššieho osvetlenia, použijem LED modrej farby, ktorá svieti najjasnejšie spomedzi bežných LED. K tomu mi pomôže schéma z internetu 5.5. Jedná sa o pomerne bežný problém a existuje riešenie ako dosiahnuť požadovaný výsledok. Rezistory R1 a R2 slúžia ako napäťový delič pomocou ktorých sa na Pin5 dostane požadované napätie a EV3 tak rozpozná pripojenú LED ako motor. Rezistor R3 slúži len pre zníženie prúdu prechádzajúceho cez LED aby sa zabránilo jej prípadnému zahrievaniu či zhoreniu.





Obr. 5.5: Schéma napojenia LED na EV3 konektor

### 5.5.2 Tvorba vysieláča

Vedúci práce mi dal EV3 kábel. Ten som rozstrihol, odblankoval jednotlivé káble a skonštruoval obvod podľa schémy. Na biely a čierny pin som dal vodiče s F-F (Female-Female) koncovkami, na ktoré som z druhého konca napojil modrú LED. Keď som konektor pripojil do EV3, kocka ho detekovala ako motor. Potom som pomocou funkcií `run()` a `stop()` mohol blikať LED. Prototyp vysieláča bol hotový.

### 5.5.3 Prijímač

Dáta prijímam pomocou LEGO senzora farby a vzdialenosti. Ten pomocou funkcie `ambient()` vráti hodnotu osvetlenia okolia. Keď som naň však blikať vysieláčom, detekoval nepresné hodnoty. To spôsobovalo osvetlenie okolia, ktoré znemožňovalo presne určiť či LED žiary, alebo ne-žiary. Aby som nemusel senzor pred každým použitím kalibrovať a vyhol sa rušeniu od okolia, uzavrel som senzor aj LED do jednej škatuľky z LEGO kociek. Senzor som naprogramoval tak, aby čítal hodnoty s rovnakou frekvenciou ako LED odosiela.

### 5.5.4 Prenos dát

Aby som mohol dáta prenášať, naprogramoval som zjednodušenú verziu komunikačného protokolu, ktorý som opísal v štvrtej kapitole. Pomocou tlačidla napojeného na EV3 som mohol prvým stlačením spustiť motor a druhým ho zastaviť. Najskôr som pracoval s malou prenosovou rýchlosťou, keď však všetko fungovalo podľa očakávaní, začal som hľadať maximálnu prenosovú rýchlosť.

#### Nájdenie maximálnej prenosovej rýchlosti

Maximálna prenosová rýchlosť určuje, ako rýchlo môžem spoľahlivo prijímať jednotlivé bity. Čím rýchlejšia prenosová rýchlosť, tým viac dát môžem preniesť za jednotku času. Je však nutné, aby prijaté dáta boli rovnaké ako odvysielané dáta, inak je prenos nespoľahlivý. Postupne som teda testoval môj prototyp za účelom nájdenia tejto hodnoty. Ukázalo sa, že

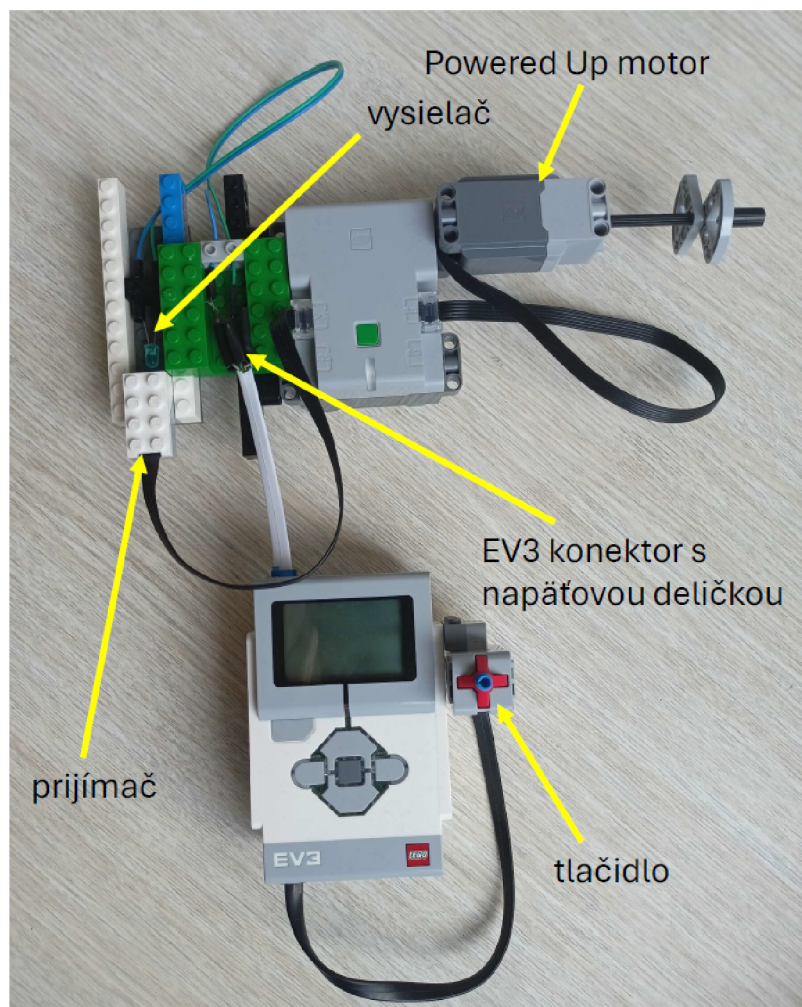
pri frekvencii vyššej ako 12,5 Hz už nie je príjem spoľahlivý. To je pravdepodobne zapríčinené prijímačom aj vysielačom, keďže tieto súčiastky nie sú k takémuto použitiu prispôbené. Táto rýchlosť je pre testovanie dostačujúca, pre praktické účely je však prenos príliš pomalý. Ak by som chcel zrýchliť prenosovú rýchlosť, musel by som využiť súčiastky špeciálne vyrobené pre toto použitie, dostatočne citlivý senzor a vysielač schopný oveľa rýchlejšie meniť svoj stav.

### 5.5.5 Konštrukcia prototypu a prezentovanie jeho funkcionality na jednoduchom programe

Pomocou vytvoreného vysielača a prijímača som vytvoril jednoduchý prototyp. Funkčnosť komunikačného rozhrania a samotného protokolu je znázornená pomocou spínača pripojeného na port **S1** kocky EV3 a motora na porte **B** pripojeného na Technic hub. Pri stlačení tlačidla pokiaľ je motor v klude sa pošle signál, aby motor začal rotovať oskou rýchlosťou 1000 otáčok za minútu, signál je zložený z bajtov **0xfe** reprezentujúci unkciiu `MotorRun()`, bajtu **0xbb** značiaci port B na strane kocky Technic hub a dvoj bajtovú hodnotu rýchlosti. Viac bajtové hodnoty sa posielajú vo forme **Little Endian** (teda prvý sa posielajú najmenej významný bajt). Druhé stlačenie tlačidla následne motor zastaví bajtom **0x22** značiac príkaz `MotorStop()`. Proces je následne možné opakovať. Na obrázkoch 5.6 je možné vidieť primitívne LEGO zariadenie na ktorom beží môj program. Je možné vidieť, že modrá LED je len niekoľko milimetrov vzdialená od senzora a ak chceme aby bolo možné ju spoľahlivo používať, je nutné ju zakryť zo všetkých strán aby svetlo nerušilo prenos signálov.

### 5.5.6 Návrh vylepšení pre ďalší vývoj

Ukázalo sa, že je možné prepojiť kocky EV3 a Technic hub pomocou svetelného rozhrania, prenosová rýchlosť je však nedostatočná. V ďalšej iterácii tohoto prototypu teda využijem špecializované súčiastky určené pre galvanické oddeľovanie. Ukázalo sa, že komunikácia za využitia imitovaného vysielača je možná a ovládanie pomerne jednoduché, otázkou ostáva ako to bude s prijímačom, keďže senzory už využívajú pomerne komplikovanejšie rozhranie ako prosté motory. Taktiež by bolo vhodné, aby mohol Technic Hub odosielať dáta naspäť to EV3. To by umožnilo nielen odosielať návratové hodnoty funkcií, ale aj kontrolovať, či zariadenia správne komunikujú a aj spracovávať dáta zo senzorov.



Obr. 5.6: Prototyp jednosmernej komunikácie využívajúci LEGO senzor farby a vzdialenosti

## Kapitola 6

# Návrh a realizácia obojsmerného komunikačného protokolu

V tejto kapitole sa zameriam na tvorbu rozhrania umožňujúce obojsmerný prenos dát (schéma na obr. 6.2) s využitím Arduina pre prijímanie a vysielanie dát medzi kockami. Cieľom je aby výsledné riešenie čo najmenej ovplyvňovalo fungovanie kociek, bolo jednoduché na zapojenie a použitie. Následne skonštruujem LEGO model využívajúci ďalšie LEGO periférie a experimentálne overím funkčnosť môjho riešenia.

### 6.1 Identifikácia rozhrania pomocou imitácie Arduina ako oficiálnej LEGO periférie

Je optimálne, aby fyzické rozhranie využívalo len jeden port na oboch kockách. Preto je nutné imitovať perifériu s možnosťou prenosu signálov oboma smermi. Pre kocku EV3 to umožňuje trieda `Motor` ale taktiež aj priamo Pybricks trieda `UARTDevice`. Tá mi umožňuje nastaviť potrebnú rýchlosť **115200 Bd** a poskytuje funkcie `waiting()`, `read()` a `write()`. Jednoducho tak vyriešim prijímanie aj odosielanie dát.

Pri kocke Technic hub to umožňuje motor, rovnako ako pri kocke EV3, nie je však možné využiť rovnaké zapojenie, keďže pri novej kocke Technic hub sa všetky “rozumné” periférie identifikujú pomocou sériovej linky a protokolu UART na pinoch Pin5 a Pin6. Technic hub taktiež podporuje senzor farby a vzdialenosti. Tento viac účelový senzor taktiež obsahuje RGB LED ktorá sa dá rozsvietiť na 3 rôzne farby a zhasnúť. Pomocou dát nameraných z logickej sondy popísaných v sekcii 4.5.2 sa dá nadviazať a udržať komunikácia s perifériou a pomocou Pybricks sa dajú posielat a prijímať dáta medzi kockou a Arduinom. Technic hub nemá triedu v knižnici Pybricks pre čistý UART protokol. Budem teda musieť pracovať s imitáciou senzora svetla a vzdialenosti so vstavanou LED. To zvýši réžiu na prenos jedného bajtu a niekoľkonásobne spomalí prenos.

### 6.2 Rozpoznávanie signálov indikujúcich logickú 1 a logickú 0 pri prijímaní a vysielaní.

Aby komunikácia mohla prebiehať, je potrebné rozpoznávať medzi základnými stavmi log. 0 a log 1. Tie skladať do bajtov dát. Odosielať prijímať a následne dekodovať na informáciu ktorú reprezentujú.

EV3 vďaka Pybricks triede `UARTDevice` umožňuje poslať a čítať reťazce bajtov bez toho, aby som musel nejakú modifikovať jej funkcionality. Arduino tak jednoducho dokáže prijímať a poslať dáta.

Technic hub však takéto prosté posielanie dát neumožňuje. Je preto treba pracovať s obmedzeniami a vnútorným rozhraním. Prijímať dáta je možné pomocou Pybricks funkcie `color()` ktorá načíta reťazec znakov obsahujúci RGB dáta zo senzora farby a interpretuje to ako jednu zo šiestich detekovateľných farieb (čierna, červená, modrá, zelená, žltá, biela). Tie sa následne dajú dekodovať na nejakú danú dvoj bajtovú hodnotu. Poslať dáta je možné pomocou Pybricks funkcie `light.on(Color)`. Tá dokáže odoslať štyri rôzne reťazce (čierna, červená, zelená, modrá). Tie môžeme Arduino interpretovať ako jednotlivé dvoj bajtové hodnoty.

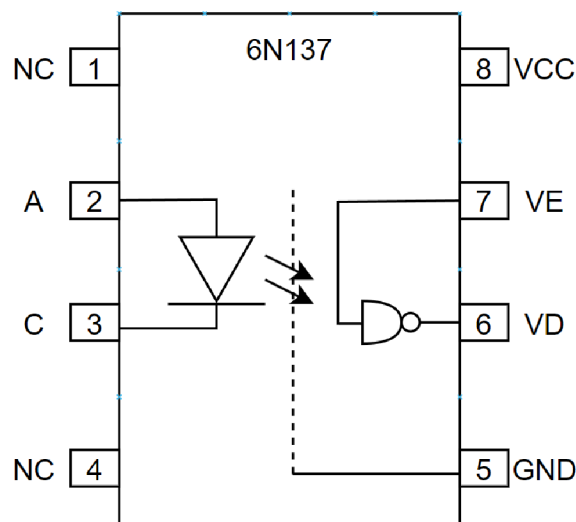
### 6.3 Galvanické oddelenie

Prvý prototyp využíval pre prenos cez optické rozhranie. To sa ukázalo ako najjednoduchší spôsob ako galvanicky oddeliť dva obvody. Nový prototyp teda využije rovnakú techniku, avšak viac uspošobenú na dátový prenos.

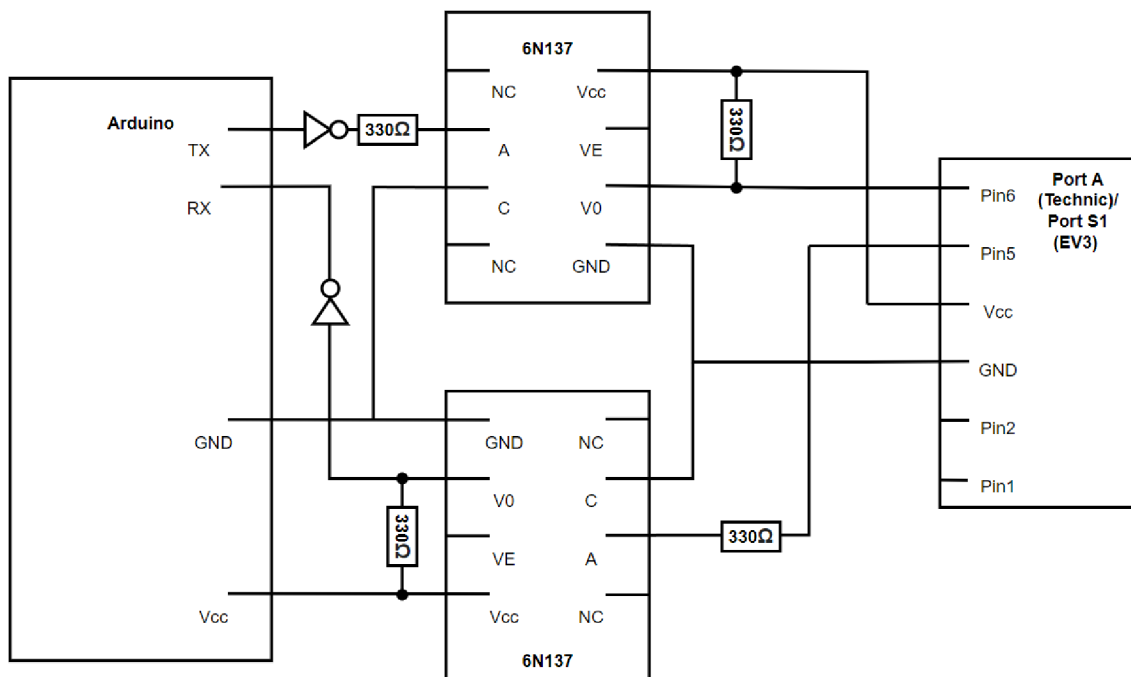
S využitím rýchlych optočlenov sa budú poslať signáli medzi kockami a Arduino. Arduino bude spracovávať prichádzajúce a odchádzajúce dáta a upravovať ich tak aby ich daná kocka vedela prečítať.

EV3 pomocou Pybricks umožňuje využívať piny Pin5 a Pin 6 na sensorových portoch S1 až S4 pre UART komunikáciu pomocou funkcie `UARTDevice` a jej funkcií.

Keďže rýchlosť prenosu na pinoch Pin5 a Pin6 kocky Technic hub je **115200Bd** rovnako ako na EV3, je vyžadovaná vysoká prepínacia rýchlosť pre vysielač aj prijímač. Na to posluží optočlen **6N137**(obr. 6.1). Jedná sa o vysoko rýchlostný jedno kanálový optočlen ktorého stúpajúca hrana signálu trvá typicky **23ns** [23] a klesajúca hrana približne **7ns**.



Obr. 6.1: Schéma jedno kanálového optočlenu 6N137



Obr. 6.2: Schéma prepojenia fyzického rozhrania kocky Technic Hub a EV3 s Arduino

## 6.4 Konštrukcia a implementácia

Pre overenie, že všetko funguje podľa návrhu a následné testovanie použiteľnosti som navrhnuté komunikačné rozhranie postavil, na kocky som napojil ďalšie periférie. Ďalej som zostrojil jednoduchý LEGO model a napísal kód ktorý umožňuje posielanie a prijímanie dát cez komunikačné rozhranie.

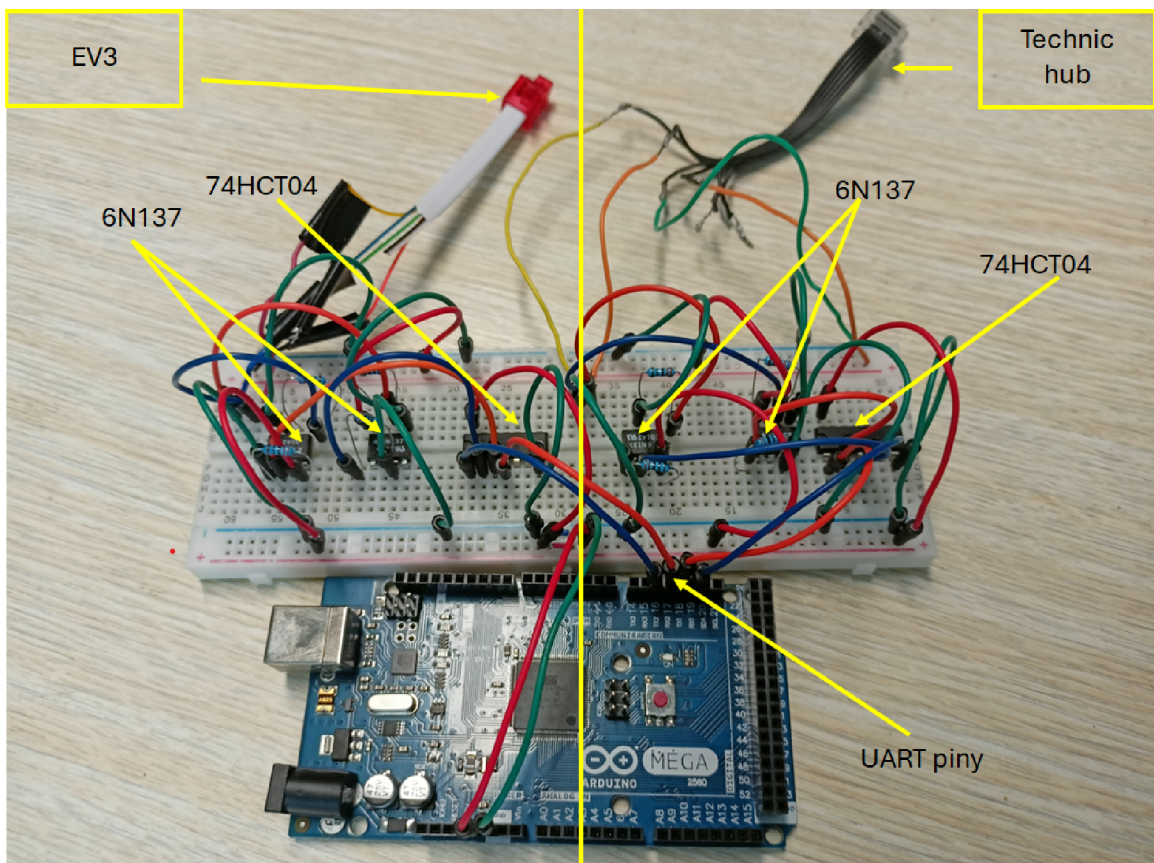
### 6.4.1 Tvorba komunikačného rozhrania

Pre skonštruovanie komunikačného rozhrania je potreba:

- 4 optočleny 6N137,
- vodiče,
- 4 logické hradlá NOT (74HCT04),
- 8  $330\Omega$  rezistorov,
- EV3 port konektor,
- Technic hub port konektor,
- Arduino alebo podobný mikrokontrolér, ideálne s tromi sériovými linkami pre jednoduchšie hľadanie chýb.
- veľký breadboard
- logická sonda na testovanie a hľadanie chýb

## Konštrukcia hardvéru

Najskôr je treba rozstrihnúť EV3 a Technic port konektory tak, aby na jednom konci konektora boli odhalené vodiče. Piny 3 až 6 následne treba odblankovať a spájať na ne pevne jednožilové vodiče aby sa dali napojiť do breadboardu. Následne podľa schémy zapojiť do optočlenov. Dôležité je vždy použiť rezistor pre zapojenie to vysielajúceho optočlena, inak by infračervená LED vo vnútri optočlena zhorela. Na strane Arduina je taktiež treba invertovať signál pomocou 74HCT04, keďže optočlen invertuje vstupný signál. V balení 74HCT04 sú obsiahnuté štyri logické hradlá NOT, a tak stačí jeden kus. Ja som použil dva na sprehľadnenie zapojenia. Nezabudnúť zapojiť aj 74HCT04 na zem a napätie Arduina. Rezistory na prijímacej časti optočlenov slúžia ako oddelenie napätia a výstupného signálu. Dátové vodiče následne zapojiť na sériové porty Arduina. Ja využívam **Arduino MEGA 2500** ktoré má niekoľko sériových portov. Pri LEGO konektoroch, Pin5 slúži ako TX a Pin6 ako RX na strane Arduina teda opačne. Využívam porty 16-19 určené pre Serial1 a Serial2, a linku Serial nechávam rezervovanú pre komunikáciu s PC a hľadanie chýb. Aj keď Arduino má knižnicu pre softvérom implementovaný sériový protokol s názvom `SoftwareSerial`, nedoporučujem ju využívať, keďže v mojom prípade na rýchlosti 115200Bd nefungovala podľa očakávania a prenášané dáta boli poškodené. Výsledné zapojenie by malo vyzeráť približne ako na obr. 6.3



Obr. 6.3: Komunikačný kanál využívajúci optočleny.

## 6.4.2 Tvorba programu

Programovanie tohoto protokolu vyžaduje napísať program pre každú časť, teda EV3, Arduino a Technic hub. Tie sa následne spustia a budú spolu komunikovať.

### Programovanie kocky EV3

Pre Programovanie kocky EV3 využijem knižnicu Pybricks. Tá umožňuje programovať EV3 v prostredí Visual Studio Code s rozšírením LEGO MINDSTORMS EV3 MicroPython. Programuje sa teda v jazyku MicroPython. Celý návod ako spojazdniť Pybricks na EV3 je na ich oficiálnej stránke <sup>24</sup>. Vďaka Pybricks triede `UARTDevice` je táto časť veľmi jednoduchá, stačí si vytvoriť triedu `TechnicHub` ktorej pri inicializácii nastavím názov portu s ktorým sa bude pracovať. A následne funkcie `read()` a `write()` pre obsluhu portu. Táto trieda sa dá jednoducho oddeliť od hlavného kódu a premiestniť ju do vlastného súboru.

### Programovanie kocky Technic hub

Pre programovanie kocky Technic hub využijem taktiež knižnicu Pybricks. Tentokrát je však nutné využiť webové programovacie prostredie Pybricks a podporovaný prehliadač, ja využívam Microsoft Edge. Samotnú implementáciu nie je možné jednoducho oddeliť od zvyšku kódu a aj spustenie kódu je naviazané na Arduino oveľa viac ako pri EV3. V hlavnom cykle prijímam dáta z portu na ktorý je pripojené komunikačné rozhranie pomocou funkcie `color()`. Tá prijíma dáta vo forme farieb. Po načítaní štyroch farieb sa následne dekóduje prijatý bajt. Každá farba má pridelenú dvoj bitovú hodnotu. Podľa hodnoty prijatého bajtu sa následne rozhoduje aká operácia sa vykoná. Ak chcem odoslať dáta do EV3 využijem funkciu `light.on(Color)` ktorá začne posielat dáta na rozsvietenie LED na pôvodnom senzore. Tieto dáta sa tak dostanú po pine Pin 6 do Arduina. Jedna farba má opäť dvoj bitovú hodnotu a odosiela sa **10ms**.

### Programovanie Arduina

Arduino má vlastné programovacie prostredie. Programuje sa v jazyku podobnému C upraveného pre potreby mikrokontroléra. Samotný program spočíva z dvoch štandardných častí. Vo funkcii `Setup()` vykonám počiatkové zahájenie komunikácie s kockou Technic hub, nastavím sériové porty a počkám až kocka začne vykonávať program. Následne vo funkcii `Loop()` kontrolujem, či mi neprichádzajú dáta od kocky EV3. Ak áno, zakódujem správu do dvoj bitových hodnôt a tie uložíam do vyrovnávacej pamäti.

Následne testujem, či mi neprichádzajú dáta z kocky Technic hub. Ak áno, zistím o akú hodnotu sa jedná:

- **0x02** - značí, že Technic hub očakáva kontrolnú odpoveď (áto správa prichádza každých 100ms0,
- **0x3A** - značí, že Technic hub vysiela čiernu farbu (zakódovaná hodnota (0b00),
- **0x39** - značí, že Technic hub vysiela modrú farbu (zakódovaná hodnota (0b01),
- **0x3F** - značí, že Technic hub vysiela zelenú farbu (zakódovaná hodnota (0b10),
- **0x33** - značí, že Technic hub vysiela červenú farbu (zakódovaná hodnota (0b11),

<sup>24</sup><https://pybricks.com/ev3-micropython/>



- **0xBA** - značí, že Technic hub je pripravený prijať jednu dvoj bitovú hodnotu vo forme farby. Prijaté farby sa ako dvoj bitové hodnoty zapisujú do 8-bitovej premennej a po jej naplnení sa správa odošle do EV3. Ak kocka Technic hub chce načítať hodnotu, pošlem prvú hodnotu z vyrovnávacej pamäti. Vyrovnávacia pamäť funguje ako fronta a jej kapacita je 5 bajtov. Arduino takto dokáže prijímať a posielat dáta do oboch kociek.

### Spustenie programu

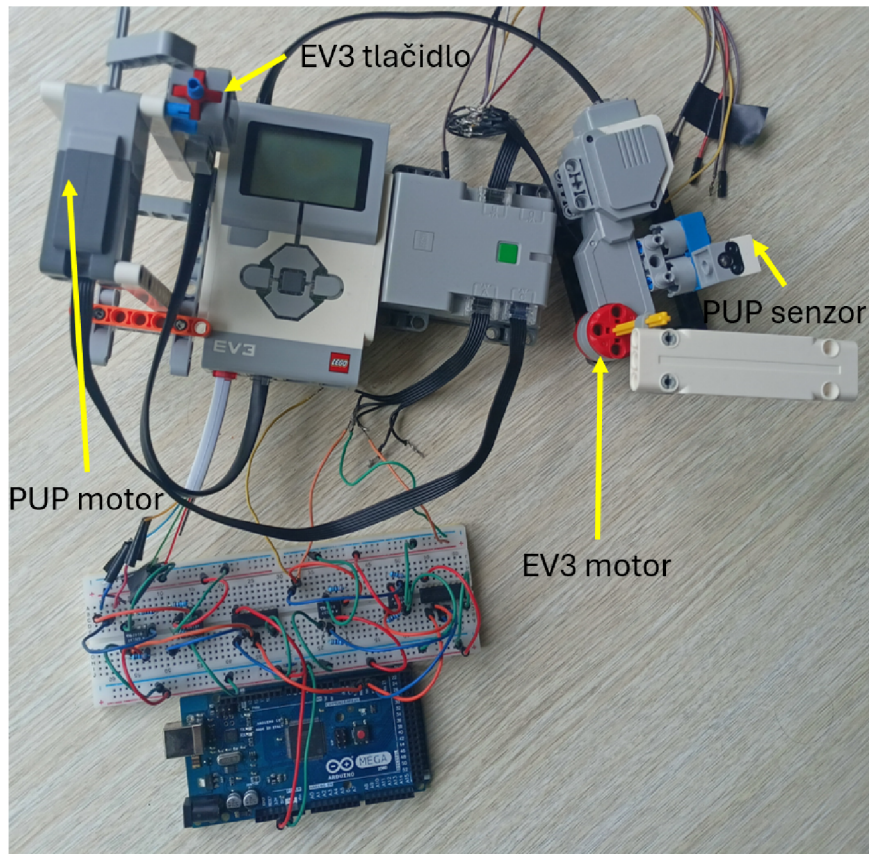
Pre úspešné spustenie programu je treba dodržiavať isté kroky. Prevažne Technic hub je úzko prepojený s Arduino, EV3 môže fungovať samostatne.

Program bežiaci na Ardiune je potrebné reštartovať vždy pred tým, ako je spustený program pre Technic hub. Pri reštartovaní je taktiež nutné rozpojiť vodič TX na strane kocky Technic hub, inak sa program nenahrá správne, a po reštartovaní opäť zapojiť. Následne je možné spustiť program pre Technic hub.

#### 6.4.3 Konštrukcia LEGO modelu a zhodnotenie výsledného riešenia

K zhotovenému komunikačnému rozhraniu som pripojil do voľných portov oboch LEGO kociek motor a senzor. Na strane EV3 tlačidlo a pre Technic hub som využil senzor svetla a vzdialenosti ktorý som v predchádzajúcich kapitolách využíval pri vývoji samotného komunikačného rozhrania (obr. 6.4).

Následne som naprogramoval jednoduchú funkcionálnu pre pozorovanie prenosovej rýchlosti z oboch smerov. Po stlačení tlačidla sa spustil EV3 motor. Ten mal na oske tienidlo. Po tom čo tienidlo prešlo ponad senzor vzdialenosti, odoslala sa správa, aby sa motor zastavil. Následne sa spustil Powered Up motor zapojený na kocku Technic hub. Ten otočil osku o 90 stupňov a na ňu pripojený blok stlačil tlačidlo na EV3 ktoré opäť spustilo EV3 motor a následne sa vrátil na pôvodnú pozíciu.



Obr. 6.4: Model využívajúci moje komunikačné rozhranie a ďalšie LEGO prvky.

### Zhodnotenie

Po spustení programov na oboch kockách a Arduino som stlačením tlačidla uviedol celý model do prevozu. Ukázalo sa, že prenos jedného bajtu z EV3 do kocky Techic hub trvá približne 4 sekundy a opačným smerom menej ako sekundu. To je pravdepodobne spôsobené tým, aký spôsobom sú dáta prijímané. Pri modifikácii kódu a jeho optimalizácii by sa malo dať zvýšiť prenosovú rýchlosť a zabrániť aj chybám v prenose. Pri zanedbaní rýchlosti však model dosahuje uspokojivých výsledkov a druhý prototyp komunikačného protokolu tak hodnotím za využiteľný. Po prevedení celého komunikačného kanála z breadboardu na dosku plošných spojov sta taktiež aj zmenší plocha ktorú model zaberá, čo ešte viac uľahčí prácu s touto perifériou.

# Kapitola 7

## Záver

Cieľom tejto práce bolo navrhnúť a zrealizovať spôsob, ktorým by bolo možné ovládať motory Technic Hub z prostredia EV3. V rámci teoretickej časti som spísal históriu programovateľných LEGO stavebníc od prvých prototypov až po dnešné moderné kocky, ktoré sa využívajú, či už v školách alebo doma. Ďalej som dohľadal existujúce riešenia pre prepojenie dvoch programovateľných kociek, či už pri starších modeloch ako VLL prepojenie Scout a MicroScout, alebo moderných modeloch EV3 a Spike Prime, ktoré sa prepájajú pomocou Bluetooth. Navrhol som možné riešenia prepojenia EV3 a Technic Hubu od jednoduchších, ako prepojenie dvoch motorov jedným hriadeľom po komplexnejšie ako využitie protokolu sériovej komunikácie medzi kockami a ich perifériami. Rozhodol som sa zrealizovať riešenie pomocou optického rozhrania.

Stručne som zhrnul základné vlastnosti protokolov vo vstavaných systémoch a na základe tejto teórie som vytvoril komunikačný protokol a funkčný prototyp. Ten za pomoci LED a svetelného senzoru prenášal dáta z EV3 do kocky Technic Hub a umožňoval tak dvom nekompatibilným súčiastkam spolu komunikovať. Zistil som, že takéto hardvérové zapojenie však nedosahuje dostatočnú rýchlosť a je treba ho upraviť a umožniť tak vyššie prenosové rýchlosti a obojsmernú komunikáciu.

Metódou spätného inžinierstva som zistil som ako prebieha komunikácia medzi perifériou a kockou. Konkrétne ako prebieha nadviazanie komunikácie, udržanie komunikácie pomocou periodických správ, odosielanie a prijímanie dát medzi kockou Technic hub a LEGO sensorom svetla a vzdialenosti. Tieto znalosti som využil a pomocou rýchlo spínacích optočlenov som pripojil na obe LEGO kocky Arduino imitujúci senzor alebo motor. Tým som dosiahol rýchlejšiu a spoľahlivejšiu obojsmernú komunikáciu medzi kockami LEGO MINDSTORMS EV3 a LEGO Control+ Technic Hub.

Táto práca dokázala, že existuje spôsob, ako prepojiť nekompatibilné LEGO programovateľné kocky a komponenty. Odhalila niektoré implementačné detaily interného LEGO komunikačného protokolu, ktorým sa do teraz len málo ľudí zaoberalo. Ukázala niektoré rozdiely medzi kockami EV3 a Technic hub. Prevažne kocka Technic hub nie je na takéto účely navrhnutá a oficiálne ani neoficiálne riešenia neumožňujú programovať túto kocku tak aby bol tento typ komunikácie efektívny. To je prevažne spôsobené firmvérom a tým, že jeho zdrojový kód nie je otvorený softvér a teda nie je možné vytvárať nové triedy pre komponenty na strane softvéru. Môj prístup k riešeniu je však stále jeden z najjednoduchších, ako prepojiť tieto dve kocky a táto práca poskytuje informácie a riešenie pre hocikoho s podobným problémom.

Aj naďalej ostáva na mojej práci množstvo práce. Ešte stále existujú možnosti, ako urýchliť dátový prenos a spoľahlivosť prenosu. Doladiť softvérovú časť komunikačného pro-

tokolu a tým zvýšiť prenosovú rýchlosť. Taktiež, umožniť ovládať EV3 prvky z prostredia Control+ Navrhnuť dosku plošných spojov a zapuzdriť ju do kompaktnej krabičky.

# Literatúra

- [1] BARTNECK, C. *Building Instructions for a cable that connects LEGO Power Functions (PF) with Mindstorms NXT/EV3*. 2015. Dostupné z: <https://www.bartneck.de/2015/06/01/building-instructions-for-a-cable-that-connects-lego-power-functions-pf-with-mindstorms-nxtev3/>.
- [2] BATTERYPOWEREDBRICKS. *Lego Dacta Serial Interface and Control Lab software Retrospective*. 2022. Dostupné z: <https://www.youtube.com/watch?v=16Kz4tRZBBo>.
- [3] BATTERYPOWEREDBRICKS. *Lego Technic Control Center Retrospective*. 2022. Dostupné z: <https://www.youtube.com/watch?v=AJGVnLinfSE>.
- [4] BLACKBIRD. *Blackbirds Technicopedia*. 2007. Dostupné z: <http://technicopedia.com/index.html>.
- [5] BRICK, R. *LEGO® Powered Up summary*. 2021. Dostupné z: <https://racingbrick.com/lego-powered-up-summary/>.
- [6] BRICKIPEDIA. *Spybotics*. 2006. Dostupné z: [https://brickipedia.fandom.com/wiki/LEGO\\_Wiki](https://brickipedia.fandom.com/wiki/LEGO_Wiki).
- [7] BRICKSET. *Brickset*. 1997. Dostupné z: <https://brickset.com/>.
- [8] DAVID LECHNER, I. *PyBricks Documentation*. 2024. Dostupné z: <https://docs.pybricks.com/en/latest/>.
- [9] ELPROCUS. *What are Communication Protocols Their Working*. 2023. Dostupné z: <https://www.elprocus.com/communication-protocols/>.
- [10] ERIC PEÑA, M. G. L. UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter. *Analog Dialogue*. 2020, zv. 54, č. 4. Dostupné z: <https://fgcoca.github.io/Autocaravana-inteligente/datasheet/uart-a-hardware-communication-protocol.pdf>.
- [11] FALCÃO, O. *LEGO Interface A*. 2017. Dostupné z: <https://ofalcao.pt/blog/series/lego-interface-a>.
- [12] HURBAIN, P. *The Powered Up connector*. 2022. Dostupné z: <https://www.philohome.com/wedo2reverse/connect.htm>.
- [13] JNCRATON. *What internal electronics does the Control+ Technic Hub (88012) use*. 2021. Dostupné z: <https://bricks.stackexchange.com/questions/15530/what-internal-electronics-does-the-control-technic-hub-88012-use>.

- [14] KEIM, R. *Beyond the Optocoupler: Understanding Digital Isolator*. 2018. Dostupné z: <https://www.allaboutcircuits.com/technical-articles/beyond-the-optocoupler-understanding-digital-isolators/>.
- [15] LEGO, G. *Robotics Invention System 2.0*. 1999. Dostupné z: <https://www.lego.com/cdn/product-assets/product.bi.core.pdf/4157492.pdf>.
- [16] LEGO, G. *Scout Internals*. 2001. Dostupné z: <https://web.archive.org/web/20010210104326/http://mindstorms.lego.com/products/rds/hackers.asp>.
- [17] LEGO, G. *NXT User Guide*. 2006. Dostupné z: <https://www.generationrobots.com/media/Lego-Mindstorms-NXT-Education-Kit.pdf>.
- [18] LEGO, G. *LEGO MINDSTORMS EV3 Hardware Developer Kit*. 2013. Dostupné z: [https://www.mikrocontroller.net/attachment/338591/hardware\\_developer\\_kit.pdf](https://www.mikrocontroller.net/attachment/338591/hardware_developer_kit.pdf).
- [19] LEGO, G. *LEGO® Education WeDo 2.0*. 2016. Dostupné z: <https://education.lego.com/en-us/product-resources/wedo-2/troubleshooting/faqs>.
- [20] MINDSTORMS, A. *How to Remote Control Robot Inventor with EV3*. 2021. Dostupné z: <https://www.youtube.com/watch?v=H37LTepcvXI>.
- [21] MYBRICKS. *LEGO® CyberMaster*. -. Dostupné z: <https://mybricks.net/set/8482/>.
- [22] NELSON, R. *LEGO® MINDSTORMS(tm) Internals*. 2007. Dostupné z: <http://crynwr.com/lego-robotics/>.
- [23] SEMICONDUCTORS, V. *High Speed Optocoupler, Single and Dual, 10 MBd*. VISHAY INTERTECHNOLOGY, 2024. Dostupné z: <https://www.vishay.com/docs/84732/6n137.pdf>.
- [24] SIEBER, A. *Control a LEGO Code Pilot, Scout, or MicroScout via Swift Playgrounds on iPadOS*. 2020. Dostupné z: <https://adriansieber.com/control-lego-via-swift-on-ipados/>.
- [25] SPOL., D. J. a. *Bricklink*. 2000. Dostupné z: <https://www.bricklink.com/v2/main.page>.
- [26] SPOL., F. L. M. a. *To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines*. 2000. Dostupné z: [https://www.researchgate.net/publication/242467788\\_To\\_Mindstorms\\_and\\_Beyond\\_Evolution\\_of\\_a\\_Construction\\_Kit\\_for\\_Magical\\_Machines](https://www.researchgate.net/publication/242467788_To_Mindstorms_and_Beyond_Evolution_of_a_Construction_Kit_for_Magical_Machines).
- [27] TALENTS, R. G. *HISTORY OF LEGO EDUCATION WEDO*. 2020. Dostupné z: <https://www.robotsgottalents.com/post/history-of-lego-education-wedo>.
- [28] WALTERS, K. *Lego Micro Scout*. 2018. Dostupné z: <https://learn.adafruit.com/bluetooth-remote-for-lego-droid/lego-micro-scout>.
- [29] WIKIPEDIA users. *Lego Mindstorms EV3*. 2019. Dostupné z: [https://en.wikipedia.org/wiki/Lego\\_Mindstorms\\_EV3](https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3).
- [30] WU, J. *A Basic Guide to I2C*. TEXAS INSTRUMENTS, 2022. Dostupné z: [https://www.ti.com/lit/an/sbaa565/sbaa565.pdf?ts=1712807276188&ref\\_url=https%253A%252F%252Fwww.linkedin.com%252F](https://www.ti.com/lit/an/sbaa565/sbaa565.pdf?ts=1712807276188&ref_url=https%253A%252F%252Fwww.linkedin.com%252F).

[31] YSARD. *MyOwnBricks*. 2021. Dostupné z: <https://github.com/ysard/MyOwnBricks>.