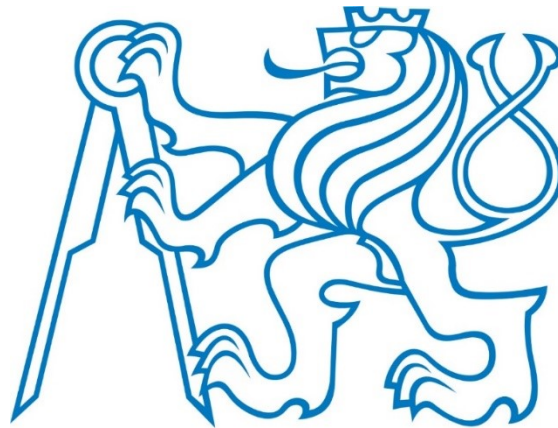


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ

OBOR GEODÉZIE, KARTOGRAFIE A GEOINFORMATIKA

PROGRAM GEODÉZIE A KARTOGRAFIE



BAKALÁŘSKÁ PRÁCE

Mobilní aplikace řízení systému pro kontrolu jeřábové dráhy

Mobile Application of System Control for Crane Way Testing

Autor: Svetlana Usik

Vedoucí práce: Ing. Zdeněk Vyskočil, Ph.D.

Praha, 2019



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Usik Jméno: Svetlana Osobní číslo: 423591

Zadávací katedra: K155 - Katedra geomatiky

Studijní program: (B3645) Geodézie a kartografie

Studijní obor: (3646R011) Geodézie, kartografie a geoinformatika

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce: Mobilní aplikace řízení systému pro kontrolu jeřabové dráhy

Název bakalářské práce anglicky: Mobile application of system control for crane way testing

Pokyny pro vypracování:

Návrh struktury řízení systému (vozíku pro proměrování jeřabové dráhy)

Vytvoření řídicí aplikace pro systém Android

Testování systému na reálné jeřabové dráze


Seznam doporučené literatury:

Jméno vedoucího bakalářské práce: Ing. Zdeněk Vyskočil, Ph.D.

Datum zadání bakalářské práce: 22.2.2019

Termín odevzdání bakalářské práce: 26.5.2019

Údaj uveďte v souladu s datem v časovém plánu příslušného ak. roku


Podpis vedoucího práce


Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v bakalářské práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

22.02.2019
Datum převzetí zadání


Podpis studenta(ky)

ABSTRAKT

Tato bakalářská práce se zabývá tvorbou mobilní aplikace, která umožňuje řízení systému pro kontrolu jeřábové dráhy v online vývojovém prostředí MIT App Inventor 2. Zařízení se zakládá na jednodeskovém počítači Arduino, ke kterému byla připojena veškerá periferie. Dále zde je popsán postup, jakým je možné externí součástky napájet a jak je lze ovládat.

KLÍČOVÁ SLOVA

Kontrola jeřábové dráhy, Arduino, Arduino IDE, App Inventor, Android aplikace

ABSTRACT

This bachelor thesis deals with creating an application for mobile phones, which enables managing a system for testing crane ways in online development environment MIT App Inventor 2. Device is based on a single-board computer Arduino, to which all the periphery is connected. There is a description of a method, which allows to connect different parts, and how to manage them.

KEYWORDS

Crane way test, Arduino, Arduino IDE, App Inventor, Android software

PROHLÁŠENÍ

Já, Svetlana Usik, prohlašuji, že tuto bakalářskou práci na téma "Mobilní aplikace řízení systému pro kontrolu jeřábové dráhy" jsem vypracovala a napsala samostatně pod vedením Ing. Zdeňka Vyskočila, Ph.D., za použití zdrojů uvedených v referenci.

V Praze dne

Svetlana Usik

PODĚKOVÁNÍ

Ráda bych zde poděkovala vedoucímu bakalářské práce Ing. Zdeňku Vyskočilovi, Ph.D. za jeho odborné rady, ochotu a čas, který mi věnoval při torbě této práce. Také děkuji Jiřímu Emlerovi, který pomohl s technickou realizací projektu.

OBSAH

ÚVOD.....	8
I KOMPONENTY.....	10
1 Arduino	10
1.1 Vývojová deska.....	10
2 Periferie	11
2.1 Bluetooth modul HC-05.....	12
2.2 Krokový motor	14
2.3 Ovladač krokového motoru.....	15
II ARDUINO IDE	17
3 Představení vývojového prostředí	17
3.1 Sériová komunikace.....	18
3.2 Vstupy a výstupy	18
3.3 Autorská funkce.....	18
3.4 Práce s datovým typem char	18
3.5 Příkaz break.....	19
4 Ovládaní Arduino desky.....	19
III APP INVENTOR	20
5 Blokový jazyk.....	20
6 Popis prostředí.....	20
7 Sestavení vlastní aplikace.....	20
7.1 Popis jednotlivých bloků.....	20
IV UŽIVATELSKÁ DOKUMENTACE.....	23
8 Způsoby nahrání aplikace do zařízení	23
8.1 Spustitelný soubor	23
8.2 QR kód.....	23
9 Spouštění aplikace.....	23
10 Uživatelské rozhraní	24
10.1 Párování zařízení.....	24

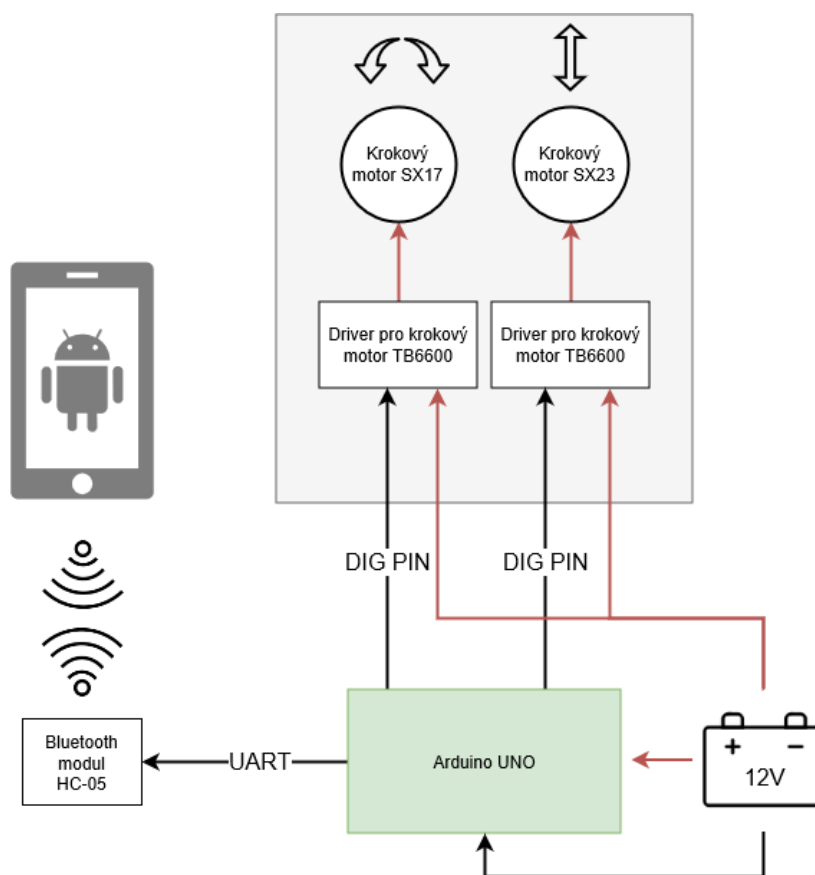
10.2	Otáčení hranolem	25
10.3	Pohyb zařízení.....	26
10.4	Nouzové zastavení	26
REFERENCE		28
SEZNAM OBRAZKŮ		30
SEZNAM POUŽITÝCH PROGRAMU		31
SEZNAM PŘÍLOH		31
ELEKTRONICKÁ PŘÍLOHA.....		32

ÚVOD

Cílem této bakalářské práce je vytvořit mobilní aplikaci, která by umožnila řízení systému pro kontrolu jeřábové dráhy. Navzdory globální modernizaci přístrojového vybavení, stále hraje hlavní úlohu při měření lidský faktor. Pohyb měřického a technického personálu po nadzemní jeřábové dráze představuje bezpečnostní riziko spojené s prací ve výškách [1].

Snahou této bakalářské práce je navrhnout jednu z možností, jak zefektivnit měřický proces a zvýšit bezpečnost měřického personálu. Je třeba vzít v úvahu, že tato práce nepojednává o charakteristikách přesností sestaveného zařízení, nýbrž se zajímá o způsob, kterým se dá zařízení ovládat.

V osnově celého projektu leží vývojová deska Arduino Uno, která řídí dva krokové motory pomocí driverů. Jeden krokový motor zajišťuje pohyb zařízení, druhý krokový motor zaručuje otáčení odrazného hranolu. K Arduino je připojen Bluetooth modul díky čemuž zařízení dokáže komunikovat s mobilním telefonem (viz obr. 1).



Obrázek 1: Základní připojovací schéma

Ovládání zařízení je realizováno ve dvou částech. Za prvé bylo třeba napsat program tzv. skicu, která v podstatě je řídicím programem celého systému. Tento krok je nedílnou součástí při práci s Arduino a periferií. Za druhé v online vývojovém prostředí MIT App Inventor 2 byla vytvořena mobilní aplikace, díky které ovládání zařízení je uživatelsky přístupné. Toto vývojové prostředí jsem vybrala, protože program lze sestavovat z přeprogramovaných bloků, což ušetří čas. Další výhodou tohoto prostředí je možnost okamžitého náhledu na aplikaci v průběhu tvorby.

Tato práce je rozdělená do čtyř částí. V první části se seznámíme s komponenty, kterými je zařízení tvořeno. Následně se seznámíme s Arduino IDE a projdeme základními kroky kódu. Ve třetí části si povíme, jak byla tvořena aplikace ve vývojovém prostředí MIT App Inventor a nakonec čtenář bude obeznámený s dokumentací aplikace.

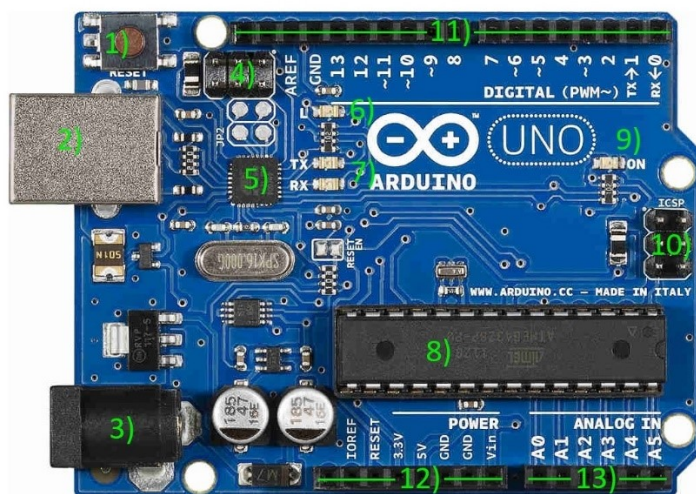
I KOMPONENTY

1 Arduino

Již léta je Arduino fenoménem na trhu. Samo o sobě nepřináší nic nového, přesto je mnohem populárnější než řešení, která existují desítky let. Arduino je tedy výjimečný tím, že se člověk dokáže prostřednictvím jednoduchého a srozumitelného vývojového prostředí dostat k programování mikroprocesoru bez nutnosti studovat složitou architekturu a logiku programování celého systému [2].

1.1 Vývojová deska

V dnešní době se na trhu objevuje celá řada více či méně vhodných vývojových desek. Pro tento projekt byla zvolena vývojová deska Arduino Uno (viz obr. 2), která současně je nejčastěji používaným typem. Je plně dostačující pro jednoduché zapojení a zkoušení nenáročných modulů. Mezi její výhody patří příznivá cena, přítomnost napájecího konektoru, USB port, dostačující počet pinů (14 digitálních vstupně výstupních a 6 analogových vstupních, které lze použít taky jako digitální) a dostačující velikost flash paměti (32 kB). Arduino Uno pracuje na procesoru ATmega328 od firmy Atmel [3].



Obrázek 2: Vývojová deska Arduino Uno

Nyní detailně prozkoumáme tuto desku [2]:

- 1) Resetovací tlačítko – použijeme ho, pokud chceme program spustit od začátku.
- 2) USB konektor – slouží k napájení, pro tvorbu sériové komunikace nebo pro nahrání kódu na desku.
- 3) Napájecí konektor – využijeme ho v případě, že Arduino nechceme napájet přes USB.
- 4) ICSP hlavice – slouží pro externí programování USB – sériového převodníku.
- 5) USB – sériový převodník – se stará o komunikaci mezi hlavním čipem a PC.
- 6) Indikační LED dioda L – je připojena k výstupu číslo 13 a proto se s ní dá vyzkoušet blikání bez připojené externí LED diody.
- 7) Indikační LED diody Rx a Tx – blikají, pokud probíhá komunikace přes sériovou linku.
- 8) Hlavní čip desky.
- 9) Indikační LED dioda ON – svítí, když je připojeno napájení.
- 10) ICSP hlavice – slouží pro externí programování hlavního čipu.
- 11) Digitální piny – slouží k připojení různých periférií.
- 12) Pevně napájecí výstupy Arduina.
- 13) Analogové vstupy – složí k připojení vodičů nebo je lze využít jako digitální vstupy a výstupy.

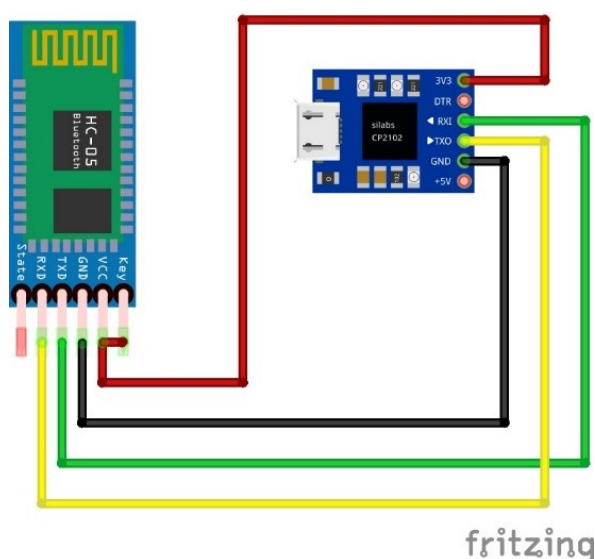
2 Periferie

Je zřejmé, že mikrokontrolér sám o sobě nemá praktický přínos. Až po jeho propojení s externími perifériemi začne vystupovat jako inteligentní zařízení. Mikrokontrolér načítá informace získané z externích modulů, které následně zpracuje podle definované logiky a přepoše na požadovaný výstup. Další interpretace výsledků je záležitostí uživatele [1].

2.1 Bluetooth modul HC-05

Pro navázání bezdrátové komunikace mezi mobilním telefonem a Arduino deskou byl použit Bluetooth modul HC-05. Tento modul obsahuje Bluetooth verze 2.0 a komunikuje s Arduinem pomocí sériové linky výchozí rychlostí 9600 baudů. Dosah, vzhledem k malému rozměru antény, je omezen na 10 metrů. Bluetooth modul vyžaduje napájecí napětí v rozsahu 3,3 až 6V [3].

Modul HC-05 funguje v režimech Master (z angl. nadřízený) a Slave (z angl. podřízený). Proto, aby se dalo propojit Bluetooth modul s mobilním telefonem, musí modul vystupovat jako podřízený. Toto lze zajistit nastavením AT příkazů. Jelikož modul se dá konfigurovat pouze z UART rozhraní, byl použit USB/TTL převodník, připojený k BT modulu přes datové linky TX (z angl. Transmit = odchozí data) a RX (z angl. Receive = příchozí data). Tyto linky je nutné překřížit. Způsob zapojení je vidět na následujícím obrázku (viz obr.3).



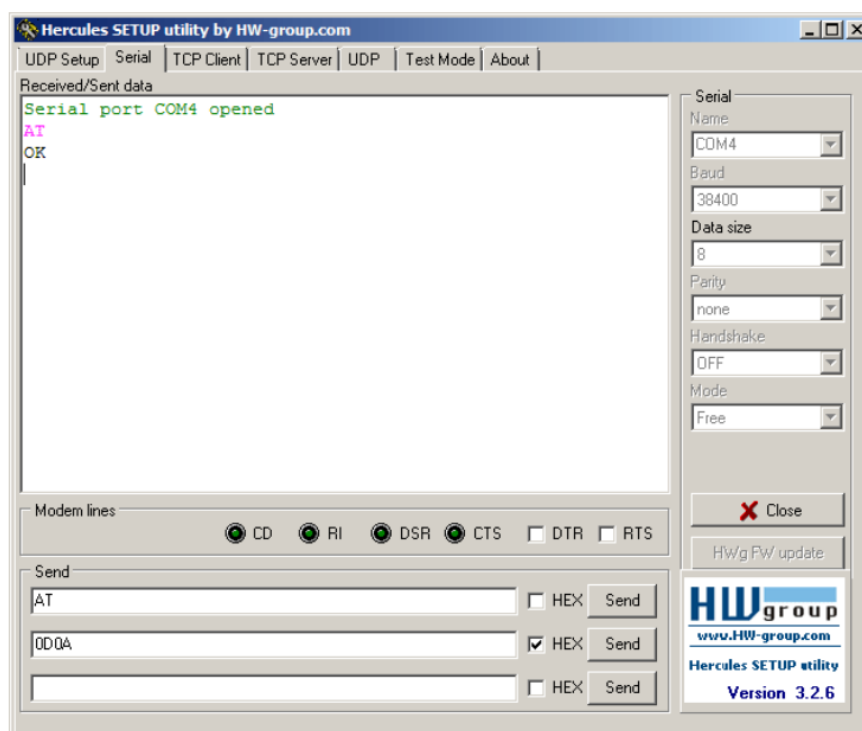
Obrázek 3: Ukázka zapojení Bluetooth modulu HC-05 a USB/TTL převodníku

Po propojení BT modulu a USB/TTL převodníku, propojíme převodník s PC. K zapisování AT příkazů byl využit sériový terminál Hercules, který je dostupný pro Windows z této adresy <https://www.hw-group.com/software/hercules-setup-utility>. V programu zvolíme záložku pro sériový port, nastavíme číslo (COM) portu, komunikační rychlost 9600,

přenos 8 bitů, bez parity a handshake. Číslo portu lze zjistit ve správci zařízení v záložce Porty (COM a LPT). V tuto chvíli můžeme začít posílat AT příkazy [4], [5].

Pro ověření, že máme s BT modulem spojení odešleme pouze samotné AT. Tento příkaz je bez parametru, a tedy nic nemění. Následně Bluetooth modul nám odpoví hlášením OK. Pošleme příkaz AT+ROLE = 0, tím nastavíme to, aby modul pracoval v režimu SLAVE (0) a dostaneme potvrzení OK. Můžeme toto nastavení zkontrolovat tak, že pošleme příkaz AT+ROLE bez parametru. Dostaneme odpověď +ROLE:0.

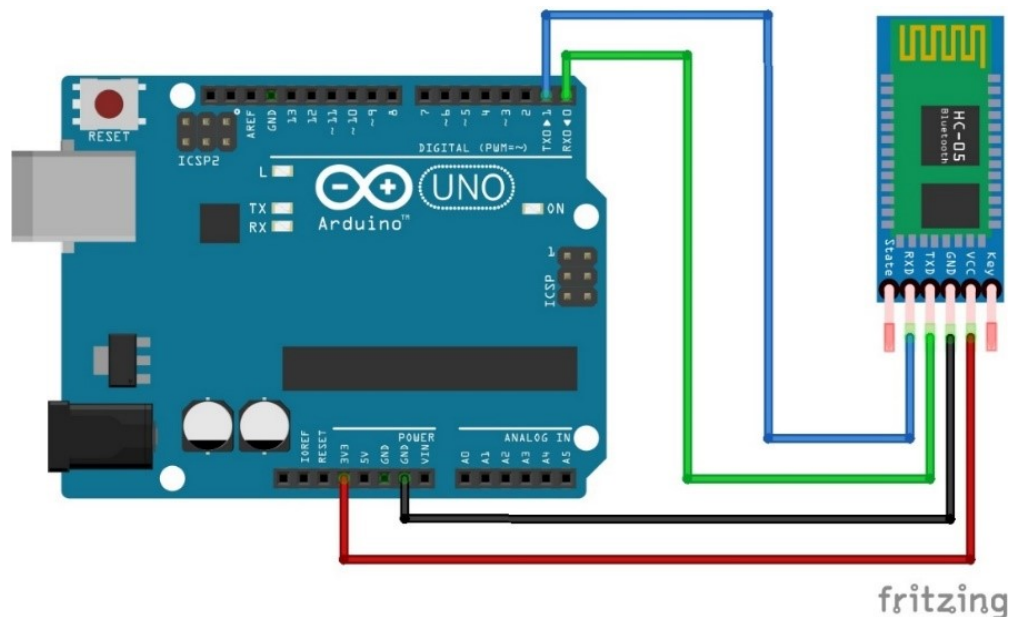
Protože prostřednictvím Bluetooth modulu bude ovládané celé zařízení, je také vhodné změnit jméno modulu pro snadnou identifikaci mezi ostatními zařízeními. Toto provedeme odesláním příkazu AT+NAME=<požadovaný název zařízení>. Stejným způsobem jako v předchozím kroku dostaneme potvrzující hlášku a můžeme zkontrolovat nastavení jména pomocí samotného příkazu bez parametru (viz obr. 4).



Obrázek 4: Terminál Hercules pro COM port

Nyní můžeme propojit Bluetooth modul HC-05 s Arduino deskou. Stačí zapojit čtyři vodiče. Pin RXD připojíme k signálu s datovým výstupem. Na

Arduino desce je tento pin označený jako TX. Pin TXD připojíme na datový vstup. Na Arduino desce je tento pin označený jako RX. Pro datové piny RXD a TXD lze využít také ostatní digitální piny, ale bylo by nutno tuto volbu definovat v projektu. Připojíme GND k pinu GND na desce. Výstup VCC připojíme na pin s napájecím napětím 3,3V. Zbylé piny EN a STATE necháme nezapojené [2]. Schéma zapojení lze vidět na obrázku dole (viz obr.5).

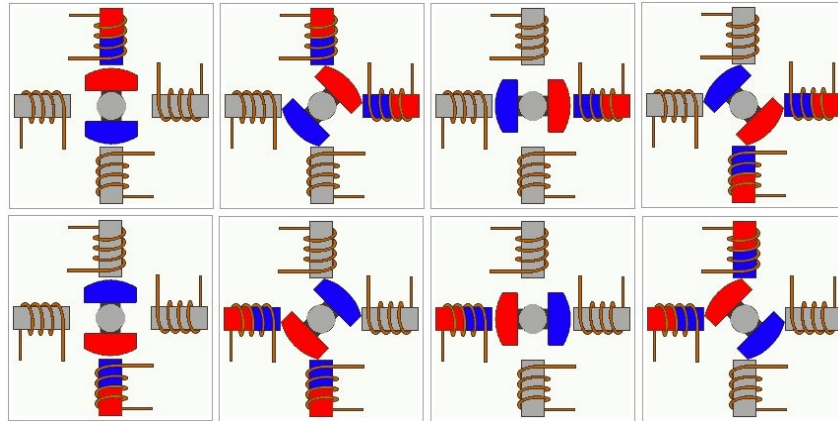


Obrázek 5: Ukázka zapojení Bluetooth modulu HC-05 k Arduino desce

2.2 Krokový motor

Zde se stručně seznámíme s popisem a principem krokového motoru. Jedná se o synchronní točivý stroj, většinou napájený impulsy stejnosměrného proudu. Krokový motor se skládá ze statoru (statická součást elektromotoru) a rotoru (pohyblivá součást). Stator krokového motoru se skládá z několika dvojic cívek. Rotor je tvořen hřídelí usazenou na kuličkových ložiskách a prstencem permanentních magnetů. Proud procházející cívkou statoru vytvoří magnetické pole, které přitáhne opačný pól magnetu rotoru (viz obr. 6). Vhodným zapojováním cívek dosáhneme vytvoření rotujícího magnetického pole, které otáčí rotorem. K pohybu krokového motoru je vždy třeba řídicí elektronika – ovladač. Výhodou tohoto

motoru je velká mechanická odolnost, dlouhodobá životnost a provoz téměř bez údržby [6], [7].



Obrázek 6: Princip práce krokového motoru – zdroj [7]

V tomto projektu pohyb zařízení dopředu nebo dozadu je realizován pomocí bipolárního dvoufázového krokového motoru řady SX23-2520 od výrobce Microcon. Rotaci hranolu zajistí bipolární dvoufázový krokový motor řady SX17-1003. Při bipolárním řízení prochází proud vždy dvěma protilehlými cívkami s navzájem opačně orientovanými magnetickými poli. Výhodou je větší krouticí moment, větší stabilita kroku, nevýhodou – vyšší spotřeba [7]. Zapojení vinutí motoru je řešeno podle dokumentace dodávané výrobcem [8], [9].

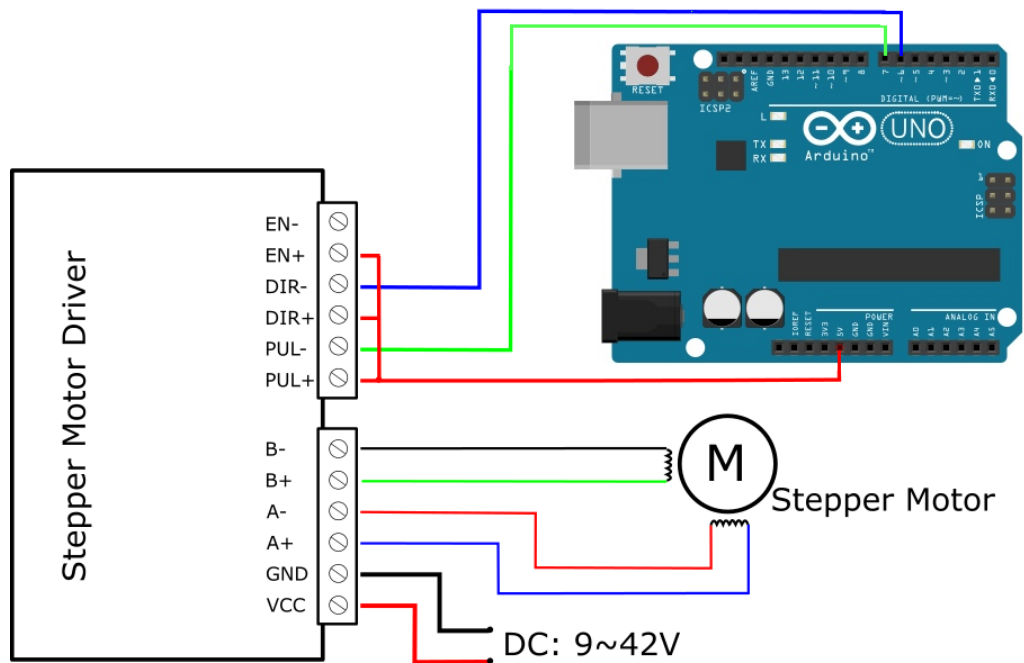
Je třeba vzít na vědomí, že oba motory mají úhlový krok $1,8^\circ$, což znamená, že pro otočení motoru o 360° je třeba učinit 200 kroků.

2.3 Ovladač krokového motoru

K ovládání krokových motorů byla použita dvojice ovladačů TB6600. Je to snadno ovladatelný driver krokového motoru, který je kompatibilní s Arduino a dalšími mikrokontroléry, které mohou produkovat 5V digitální pulzní signál. Modul je určen pro řízení jednoho bipolárního krokového motoru pomocí signálu PUL a DIR. Obsahuje obvody pro mikrokrokování a zmenšení pracovního proudu při nečinnosti, které omezí přehřívání motoru. Řídící vstupy jsou opticky odděleny, na vstupu STEP je použit vysokorychlostní obvod 6N137. Tento driver má rozsah napájecího napětí 9-42V a je schopen vyprodukovat 4A špičkový proud, který je dostačující pro

většinu krokových motorů. Krokový ovladač podporuje řízení rychlosti a směru. Pomocí DIPpřepínače lze nastavit jeho mikrokrok a výstupní proud. K dispozici je 7 druhů mikrokroků a 8 druhů řízení proudu [10].

Nyní propojíme Driver TB6600 s Arduino deskou tak, že piny s názvem PUL- a DIR- připojíme na digitální piny Arduino. Pomocí pinu DIR (z angl. direction) lze řídit směr otáčení krokového motoru. Pin PUL (z angl. pulse) zajišťuje signál impulsu motoru. Jakmile driver dostane puls, motor pohne o jeden krok. Pin EN- necháme nezapojený. Výstupy PUL+, DIR+ a EN+ připojíme k napájení řídicího systému +5V. Piny GND a VCC se připojí na zdroj napájení. Schéma zapojení lze vidět na obrázku (viz obr. 7).



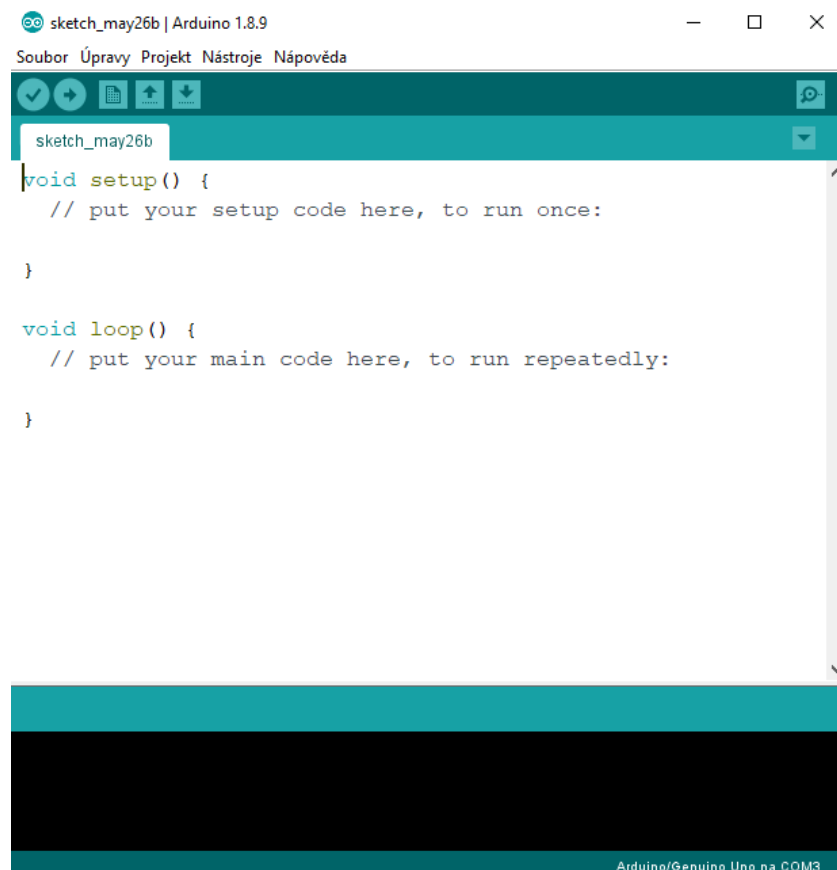
Obrázek 7: Ukázka zapojení krokového motoru přes driver TB6600 k Arduino desce – zdroj [10]

II ARDUINO IDE

3 Představení vývojového prostředí

Aby Arduino deska mohla plnit nějaké úkoly, musíme do ní nahrát Arduino kód tzv. sketch (česky skica). Takový sketch se vytváří ve vývojovém prostředí zvaném Arduino IDE. Je to open-source vývojové prostředí napsané v jazyce Java, které umožňuje psát a nahrávat kód do paměti Arduina. Arduino IDE lze nainstalovat na všechny nejrozšířenější operační systémy – Windows, Linux i Mac OS X. Na stránkách Arduino.cc je podrobný návod na instalaci.

Pro zjednodušení pro programování Arduino desky se používá knihovna Wiring programovacího jazyku C++.



Obrázek 8: Vývojové prostředí Arduino IDE

Podstatné je to, že editor je rozdělený do dvou bloků (viz obr. 8). V horní části se nachází blok `void setup()`, kde se mezi složené závorky píše

kód, což se provede jen jednou na začátku programu. Druhým blokem je *void loop()*, do jehož složených závorek se zapíše kód, který se bude opakovat stále dokola až do odpojení napájení. Tyto dvě části musí obsahovat každý program [2].

3.1 Sériová komunikace

Ke čtení informace, kterou posílá Arduino deska na PC, se používá tzv. *Serial.monitor*. Je důležité před spuštěním nastavit správný COM port, na který je připojená Arduino deska a po spuštění sériového monitoru nastavit rychlost komunikace (v našem případě 9600 baudů). K zahájení sériové komunikace mezi Arduino deskou slouží funkce *Serial.begin()*. K odeslání hodnoty z Arduina do PC se používá funkce *Serial.print()* nebo *Serial.println()*.

3.2 Vstupy a výstupy

Jak jsme již zmínili dříve, desky Arduino mají analogové a digitální piny, které lze podle potřeby nastavit jako vstupní nebo výstupní. Toto lze provést pomocí funkce *pinMode()*. Když máme digitální pin nastavený jako OUTPUT, můžeme pomocí funkce *digitalWrite()* nastavit pin na požadovanou úroveň logickou 1 (HIGH) nebo logickou 0 (LOW).

3.3 Autorská funkce

Hlavními prvky našeho zařízení jsou dva krokové motory se stejným způsobem ovládaní. Proto mnou byla vytvořena funkce *move_motor()* pro pohyb motoru, která obsahuje proměnné typu *integer* a *float*.

3.4 Práce s datovým typem char

Aby Arduino dokázal odlišit příkazy, které uživatel posílá pro ovládní motoru, byla použita funkce *Serial.readString*, která dokáže přečíst textový řetězec. Dále získáváme z textového řetězce první symbol, který má index 0. Tento okamžik je rozhodující. Jestli prvním symbolem je písmeno 's' pak se provede pohyb malým motorem, který zajišťuje rotaci hranolu. Naopak, jestli Arduino přijímá písmeno 'b', pak začne pohyb velkým motorem, který je zodpovědný za posun zařízení dopředu nebo dozadu.

Pomocí funkce *substring.toInt* zbytek textu se převede na datový typ *integer* a jako číslo je vstupní hodnotou pro krokový motor.

3.5 Příkaz *break*

K opuštění smyčky *for* se použije příkaz *break*, který dokáže zastavit rotaci motoru v případě, že na zařízení byl poslán chybný a příliš velký počet kroků.

4 Ovládaní Arduino desky

S kompletním kódem se lze seznámit v příloze (viz příloha 1). Jednotlivé části skici jsou vysvětleny pomocí komentářů.

III APP INVENTOR

5 Blokový jazyk

App Inventor je vizuálně zpracovaný přesunovací nástroj, který slouží k tvorbě mobilních aplikací na platformě Android. Je to online nástroj, a proto není vázaný na konkrétní počítačové zařízení, stačí zapnout internetový prohlížeč. Pomocí GUI nástroje lze sestavit uživatelské rozhraní (jak bude aplikace vypadat) a následně pomocí „bloků“, které se skládají dohromady, lze definovat, jak se bude aplikace chovat. Výhodou bloků je to, že zabraňují chybám preventivně, protože ne všechny bloky do sebe zapadají. Pokud například funkční blok očekává na vstupu číslo, nelze do něj zapojit text [11].

6 Popis prostředí

Programovací prostředí se dělí na designer komponent a editor bloku. V režimu s názvem Designer vidíme to, co po spuštění hotové aplikace uvidíme na displeji mobilu. Při tvorbě lze využít dva typy objektů, a to vnitřní, který bude umístěný na obrazovce mobilního zařízení (např. CheckBox, Label, Button, ListPicker atd.) nebo vnější, který bude svoji činnost vykonávat mimo obrazovku (např. BluetoothClient, Clock, AccelerometerSensor atd.).

V režimu bloků vytváříme vlastní program pomocí tzv. bloků, které dělíme na vestavené a specifické. Vestavené bloky jsou standartní bloky z nabídky, když sezemické bloky jsou vázané na komponenty, které jsme umístili na obrazovku aplikace v režimu Designer [12].

7 Sestavení vlastní aplikace

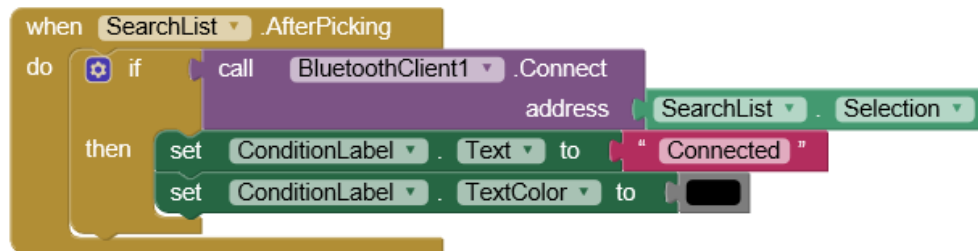
7.1 Popis jednotlivých bloků

Pro začátek chci, aby aplikace dokázala zobrazit seznam dostupných Bluetooth zařízení. K tomu jsem v designeru použila objekt ListPicker

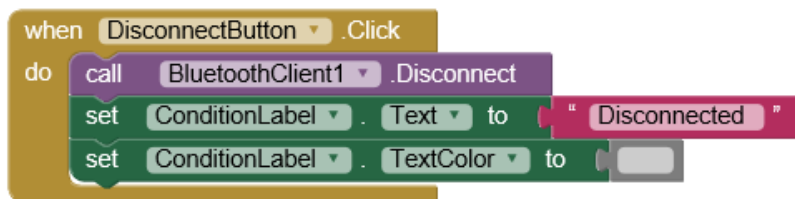
a do vývojáře jsem přidala specifický blok, který umožňuje vykonat děj, když objekt je vybrán. Poté co zařízení bude vybráno, provede se připojení.



Aby uživatel poznal, zdali připojení proběhlo úspěšně, vytvořím jednoduchou podmínku. Díky této podmínce v případě navázání komunikace mezi zařízeními na displeji mobilního telefonu se objeví nadpis Connected. Jelikož v původním stavu objekt Label je nastavený na šedou barvu, je taky třeba změnit barvu na jinou. Zvolila jsem černou.



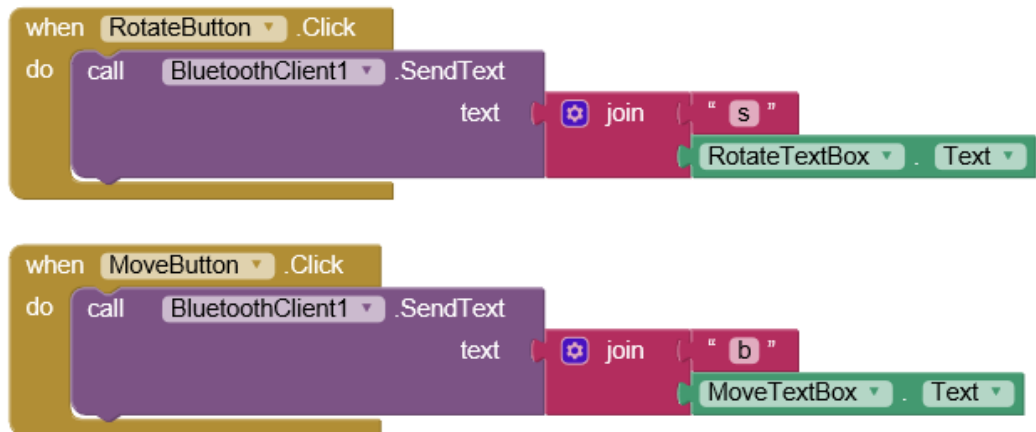
Předpokládá se, že po ukončení práce uživatel zařízení odpojí. Proto dále byl použit objekt Button. Po zmačknutí tohoto tlačítka Bluetooth komunikace bude přerušena. Objekt Label změní svůj text na Disconnected a příslušnou barvu. Za neaktivní jsem zvolila šedou.



Teď se pustíme do ovládacích tlačítek. Přidáme do okna designeru TextBox, do kterého uživatel bude moci vepsat číselnou hodnotu. Pro odeslání této hodnoty na zařízení použijeme objekt Button. Stejným způsobem přetáhneme další objekty TextBox a Button a tak vytvoříme přístup k ovládání pohybu zařízení.

Rozlišujeme pohyb hranolem pomocí písmena "s" a pohyb celým zařízením pomocí písmena "b". Tato písmena uživatel nevidí a ani je nemusí

vkładat. Navíc textové okno má omezení a lze do něj vepsat pouze číslice a znaménko mínus, které ovlivňuje směr otáčení hranolu, resp. směr pohybu zařízení.



Při práci s jakoukoliv aplikací je samozřejmé, že uživatel má soustředit svoji pozornost na dané konkrétní činnosti. Avšak budeme počítat s lidským faktorem a nabídneme uživateli možnost zařízení či hranol kdykoliv zastavit. Vytvoříme tlačítko, pomocí kterého lze odeslat zařízení hodnotu 0.

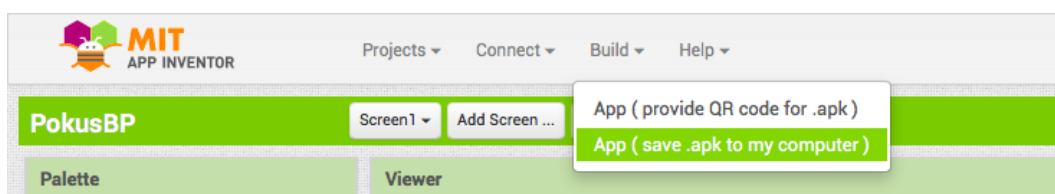
Chci podotknout, že na dané téma existuje spousta příspěvků a video manuálů na internetu, ovšem osobně bych doporučila oficiální stránky, kde v záložce Resources lze najít dostačující množství doporučených knih, tutoriálů a hlavně dokumentaci.

IV UŽIVATELSKÁ DOKUMENTACE

8 Způsoby nahrání aplikace do zařízení

8.1 Spustitelný soubor

Aplikace je dodávána ve formátu .apk a jde ji spustit na mobilním zařízení nebo tabletu s OS Android (viz obr. 9). Soubor s názvem **CWtest.apk** je nahraný na CD (viz elektronická příloha).



Obrázek 9: Ukázka záložky pro generaci spustitelného souboru

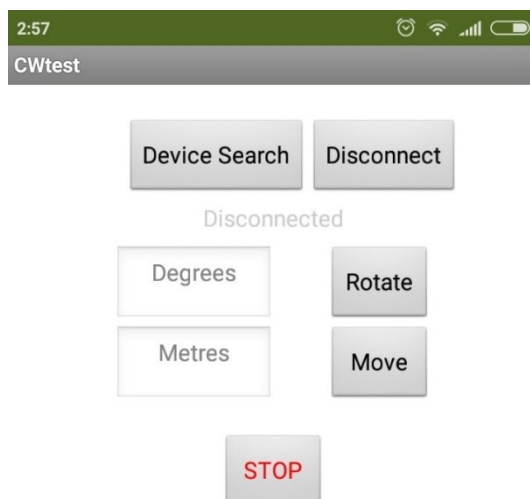
8.2 QR kód

Chce-li uživatel načíst QR kód a nainstalovat aplikaci do mobilního zařízení, musí nainstalovat aplikaci MIT AI2 Companion z Google Play. Dále už stačí načíst QR kód pomocí aplikace a potvrdit instalaci. Avšak QR kód je platný pouze dvě hodiny od vygenerování, proto tento způsob opustíme.

Při instalaci aplikací ve formátu .apk do mobilního zařízení musíte povolit instalaci aplikací z neznámých zdrojů.

9 Spouštění aplikace

Aplikace se spustí kliknutím myši na ikonu. Vzápětí se objeví hlavní okno (viz obr. 10).



Obrázek 10: Rozložení hlavního okna aplikace

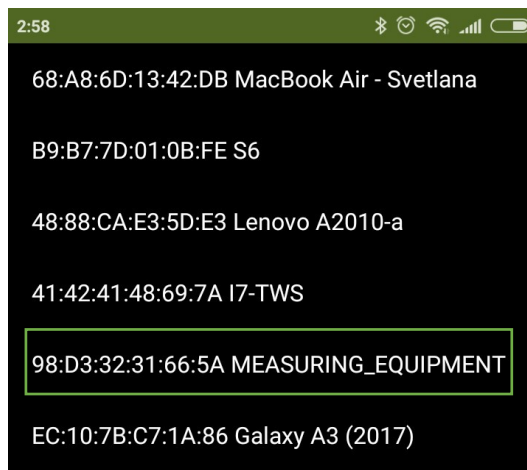
10 Uživatelské rozhraní

10.1 Párování zařízení

Jak již bylo řečeno dříve, aplikace pomocí Bluetooth umožňuje uživateli připojit se k Arduino desce.

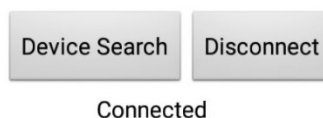
Je důležité si pamatovat, že aplikace při vyhledávání rozeznává pouze zařízení, ke kterým byl již mobilní telefon jednou připojen. Proto je nutné před prvním spouštěním aplikace připojit se k zařízení přes nastavení mobilního telefonu. Pro další navázání komunikace není nutné tento krok provádět, dokud zařízení bude v paměti Android.

Stisknutím tlačítka **Device Search** se zobrazí nabídka všech dostupných Bluetooth zařízení. Pro navázání komunikace je třeba vybrat zařízení s názvem `MEASURING_EQUIPMENT` a kliknutím je nutné výběr potvrdit (viz obr. 11).



Obrázek 11: Ukázka seznamu Bluetooth zařízení

V případě, že párování proběhlo v pořádku, indikační nadpis pod tlačítkem se změní z *Disconnected* na *Connected* (viz obr. 12).



Obrázek 12: Rozložení tlačítek odpovídajících za připojení/odpojení Bluetooth

Pro ukončení komunikace stačí jednou krátce stisknout tlačítko **Disconnect** vpravo nahoře. Jestli se odpojení zdaří, stavový nadpis *Connected* se změní zpátky na *Disconnected*. Nelze se připojit k zařízení, se kterým již komunikace probíhá. Výsledkem bude chybová hláška.

10.2 Otáčení hranolem

Arduino zařízení se snadno ovládá pomocí řídicích tlačítek a textových polí.

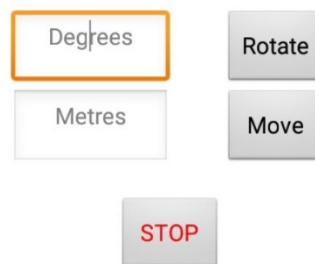
Do horního textového pole se zapisuje hodnota ve stupních. Hodnota se odešle stisknutím tlačítka **Rotate**. Zadáním kladné hodnoty bude realizován posun po směru hodinových ručiček. Zadáním záporné hodnoty bude realizován posun v opačném směru. Zorientovat se pomáhá text nápovědy v textovém poli.

10.3 Pohyb zařízení

Pomocí tlačítka **Move** a příslušného textového pole lze se zařízením hýbat. Kladná hodnota realizuje pohyb dopředu, záporná dozadu. Hodnota se zadává v metrech.

10.4 Nouzové zastavení

V případě že uživatel zadá omylem příliš velkou hodnotu pro otáčení nebo posun, nebo z jiného důvodu bude potřebovat zařízení zastavit, pak k tomuto slouží tlačítko **STOP**.



Obrázek 13: Rozložení řídicích tlačítek

ZÁVĚR

V rámci své bakalářské práce jsem vytvořila aplikaci, která ovládá systém pro kontrolu jeřábové dráhy. Řešení dané problematiky bylo rozděleno do tří částí a čtvrtou část tvoří dokumentace.

Za prvé bylo nutné navrhnout samotné zařízení. Základem jsou dva krokové motory s příslušnými ovládacími drivery. K řízení těchto motorů byla použita vývojová deska Arduino Uno. Komunikace mezi deskou a mobilním telefonem byla vyřešena prostřednictvím Bluetooth modulu HC-05.

Dále byl vytvořen sketch ve vývojovém prostředí Arduino IDE, který byl nahrán z počítače na desku přes USB kabel. Pro optimalizaci práce byla vytvořena funkce na ovládání krokových motorů. Taky byl ošetřen nezávislý přístup k jednotlivým motorům s možností náhlého zastavení. Aby práce s aplikací byla uživatelsky nenáročná, jednotkou pro řízení motoru, který bude otáčet hranolem, jsou stupně. Obdobně tak pro pohyb bylo vhodné, aby hodnota byla zadávána v metrech. Tato záležitost byla vyřešena pomocí přepočtu kroků na stupně a metry.

Nakonec byla vytvořena mobilní aplikace v prostředí MIT App Inventor 2, která umožňuje uživateli ovládání zařízení prostřednictvím textových polí a tlačítek. Tuto mobilní aplikaci lze jednoduše nainstalovat na Android zařízení. Aplikace byla testována na jeřábové dráze, která se nachází na střeše FSv ČVUT.

V této práci jsem se převážně zaměřila na základní ovládání mechanického zařízení prostřednictvím řídicího kódu a tvorbu mobilní aplikace. Během práce jsem se seznámila s open source elektronickou platformou Arduino a jejím vývojovým prostředím, kde se programuje v příslušném programovacím jazyce. Taky jsem získala základy práce s blokovým programovacím nástrojem.

Musím zmínit, že koncept Arduino je velice přizpůsobivý. Jednotlivé komponenty lze tak jednoduše stále doplňovat na základě konkrétních požadavků.

REFERENCE

- [1] SELECKÝ, Matúš. *Arduino: uživatelská příručka* [online]. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.
- [2] Programujeme Arduino | Arduino.cz. *Arduino.cz - Webový magazín o Arduinu a elektronice* [online]. Dostupné z: <https://arduino.cz/programujeme-arduino/>
- [3] Arduino Bluetooth modul HC-05 | Arduino návody. *Webový magazín o ARDUINU | Arduino návody* [online]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/arduino-bluetooth-modul-hc-05.html>
- [4] Sakul.cz - Bluetooth HC-05. *Sakul.cz - Úvod* [online]. Copyright © Resi [cit. 26.05.2019]. Dostupné z: <http://www.sakul.cz/bluetooth-hc-05/n>
- [5] Programování Bluetooth modulů HC-05 (1) RoboDoupě - web nejen o robotice. *RoboDoupě - web nejen o robotice* [online]. Copyright © 2019 [cit. 27.05.2019]. Dostupné z: <http://robodoupe.cz/2015/programovani-bluetooth-modulu-hc-05-a-hc-06/>
- [6] Krokový motor – Wikipedie. [online]. Dostupné z: https://cs.wikipedia.org/wiki/Krokový_motor
- [7] Krokový motor – MediaWiki SPŠ a VOŠ Písek. [online]. Dostupné z: http://wiki.sps-pi.cz/index.php/Krokový_motor
- [8] Microcon. *Návod k instalaci standardní zapojení vinutí. Krokové motory* [online]. Dostupné z: <http://www.microcon.cz/zapojenivinuti2012web/zapojenivinutipdf2012/ZV4lw.pdf>
- [9] Microcon. *Návod k instalaci standardní zapojení vinutí. Krokový motor SX23-2520* [online]. Dostupné z: <http://www.microcon.cz/zapojenivinuti2012web/zapojenivinutipdf2012/SX23-2520.pdf>
- [10] [online]. Dostupné z: https://wiki.dfrobot.com/TB6600_Stepper_Motor_Driver_SKU_DRI0043
- [11] WOLBER, David. *Appinventor*. Brno: ComputerPress, 2014. ISBN 978-80-251-4195-3.

- [12] Naprogramujte si aplikaci pro mobilní telefon. *Metodický portál RVP - Spomocník* [online]. Dostupné z:
<https://spomocnik.rvp.cz/clanek/18203/NAPROGRAMUJTE-SI-APLIKACI-PRO-MOBILNI-TELEFON.html>

SEZNAM OBRÁZKŮ

Obrázek 1: Základní připojovací schéma.....	8
Obrázek 2: Vývojová deska Arduino Uno	10
Obrázek 3: Ukázka zapojení Bluetooth modulu HC-05 a USB/TTL převodníku	12
Obrázek 4: Terminál Hercules pro COM port.....	13
Obrázek 5: Ukázka zapojení Bluetooth modulu HC-05 k Arduino desce.....	14
Obrázek 6: Princip práce krokového motoru – zdroj [7].....	15
Obrázek 7: Ukázka zapojení krokového motoru přes driver TB6600 k Arduino desce – zdroj [10]	16
Obrázek 8: Vývojové prostředí Arduino IDE	17
Obrázek 9: Ukázka záložky pro generaci spustitelného souboru	23
Obrázek 10: Rozložení hlavního okna aplikace	24
Obrázek 11: Ukázka seznamu Bluetooth zařízení	25
Obrázek 12: Rozložení tlačítek odpovídajících za připojení/odpojení Bluetooth.....	25
Obrázek 13: Rozložení řídicích tlačítek	26

SEZNAM POUŽITÝCH PROGRAMU

Arduino IDE

MIT App Inventor 2

Fritzing

Herculese

SEZNAM PŘÍLOH

Příloha 1 19

ELEKTRONICKÁ PŘÍLOHA

CD obsahující soubor řídicího programu ve formátu .ino a soubor mobilní aplikace ve formátu .apk.

Příloha 1 – Arduino sketch

```
1. // deklarace promennych
2. int pul_s = 9;
3. int dir_s = 8;
4. int pul_b = 6;
5. int dir_b = 7;
6.
7. // konstanta pro prevod kroku na stupne
8. float angle_const_s = 200.0/360.0;
9. // konstanta pro prevod kroku na metry
10. float r = 0.5; //cm
11. float pi = 3.14;
12. float distance_const_b = (180.0*1600.0)/(pi*r*360.0*8.0);
13.
14. void setup() {
15. // prikazy v teto funkci se provedou jen poprve
16. // pri spusteni systemu nebo po jeho restartu
17. // nastaveni rychlosti prenosu dat - zahajeni seriove komunikace
18. Serial.begin(9600);
19. // nastaveni vystupnich digitalnich pinu
20. pinMode(pul_s,OUTPUT);
21. pinMode(dir_s,OUTPUT);
22. pinMode(pul_b,OUTPUT);
23. pinMode(dir_b,OUTPUT);
24. // nastaveni digitalnich pinu na logickou 1
25. digitalWrite(dir_s,HIGH);
26. digitalWrite(dir_b,HIGH);
27. }
28. // deklarace funkce pro pohyb motoru
29. // void znamena ze ta funkce nevraci zadnou hodnotu
30. void move_motor(int motor_step, int pul, int dir, float constant) {
31. if(abs(motor_step)){
32. Serial.print("Maly motor (kroky): "); // vypis do konzoloveho okna
33. Serial.println(motor_step);
34. if (motor_step > 0){
35. digitalWrite(dir, HIGH);
36. }else {
37. digitalWrite(dir, LOW);
38. }
39. for(float x = 0; x < abs(motor_step) * constant; x++) {
40. if(Serial.available() > 0){
41. int command = Serial.parseInt();
42. if (command == 0) {
43. break; // pokud prijde 0, tak pomoci break se prerusi for smycka
44. }
45. }
46. digitalWrite(pul, HIGH);
47. delayMicroseconds(500);
48. digitalWrite(pul, LOW);
49. delayMicroseconds(500);
50. }
51. }
52. }
53.
54. void loop() {
55. // prikazy v teto funkci se budou provadet opakovane
56. if(Serial.available() > 0){ // odpovez pouze v pripade ze obdrzis data
57. String command = Serial.readString(); // napr. "s120"
58. char command_motor = command[0]; // ziskame znak 's'
59. // substring udela ze stringu "s120" string "120"
60. // ten chceme prevest na cislo (integer) takze zavolame toInt
61. int command_angle = command.substring(1, command.length()).toInt();
```

```
62.
63.   Serial.println(command);
64.
65.   if (command_motor == 's'){
66.     Serial.println("Using small motor...");
67.     move_motor(command_angle, pul_s, dir_s, angle_const_s);
68.   } else if (command_motor == 'b') {
69.     Serial.println("Using big motor...");
70.     move_motor(command_angle, pul_b, dir_b, distance_const_b); // zavolej funkci s parametry
71.   } else {
72.     Serial.println("Unknown command!");
73.   }
74. }
75. delay(500);
76. }
```