

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

**ZAŘÍZENÍ PRO TESTOVÁNÍ VLASTNOSTÍ IR SENZORU
VZDÁLENOSTI**

Ladislav Poledník

Bakalářská práce
2021

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Ladislav Poledník**
Osobní číslo: **I18311**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Řízení procesů**
Téma práce: **Zařízení pro testování vlastností IR senzoru vzdálenosti**
Zadávací katedra: **Katedra řízení procesů**

Zásady pro vypracování

Cílem práce je sestavení zařízení obsahující IR senzor vzdálenosti a servomotor pro nastavení pozice senzoru, které umožní měřit vzdálenost k překážce v poli senzoru na základě úhlu natočení. Zařízení bude schopné komunikovat s nadřazeným počítačem. V prostředí MATLAB, popř. jiném, bude vytvořeno jednoduché uživatelské rozhraní umožňující ovládat zařízení a vhodným způsobem zobrazovat získaná data. Bude využito dostupných komponent a vývojových modulů, jako např. Arduino, a příslušných periférií.

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

Seznam doporučené literatury:

NOVÁK, P. *Mobilní roboty 1 – Pohony, senzory, řízení*. BEN, 2004.
SICILIANO, B., KHATIB, O. (Eds.). *Springer Handbook of Robotics*. Springer-Verlag, 2008.

Vedoucí bakalářské práce: **doc. Ing. Jan Cvejn, Ph.D.**
Katedra řízení procesů

Datum zadání bakalářské práce: **27. listopadu 2020**
Termín odevzdání bakalářské práce: **14. května 2021**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Daniel Honc, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 29. ledna 2021

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

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

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 17. 11. 2021

Ladislav Poledník

Poděkování

Děkuji vedoucímu mé bakalářské práce doc. Ing. Janu Cvejnovi, Ph.D. za cenné připomínky a rady při tvorbě této práce. Dále bych chtěl poděkovat mé přítelkyni a rodině za jejich podporu během celé doby mého studia.

V Pardubicích dne 17. 11. 2021

Ladislav Poledník

ANOTACE

Práce se zabývá návrhem a sestrojením zařízení pro bezkontaktní měření vzdálenosti za pomoci infračerveného senzoru. Sestrojené zařízení se skládá z vývojového modulu Arduino UNO, servomotoru a IR senzoru vzdálenosti. Řídicí program a uživatelské rozhraní je vytvořeno v prostředí MATLAB. Pro komunikaci mezi počítačem a Arduinem je využita externí knihovna MATLAB Support Package for Arduino Hardware.

KLÍČOVÁ SLOVA

bezkontaktní měření vzdálenosti, IR senzor vzdálenosti, Arduino, servomotor, MATLAB.

TITLE

A DEVICE FOR TESTING IR DISTANCE SENSOR

ANNOTATION

The work deals with the design and construction of devices for non-contact distance measurement using an infrared sensor. The constructed device consists of an Arduino UNO development module, a servomotor and an IR distance sensor. The driver and user interface are created in the MATLAB environment. External library MATLAB Support Package for Arduino Hardware is used for communication between the computer and Arduino

KEYWORDS

Non-contact Distance Measurement, IR Distance Sensor, Arduino, Servomotor, MATLAB.

OBSAH

Seznam zkratk a značek	9
Seznam symbolů proměnných veličin a funkcí	10
Seznam ilustrací	11
Seznam tabulek	13
ÚVOD	14
1 IR SENZORY VZDÁLENOSTI	15
1.1 Infračervené záření	15
1.2 Prvky IR senzoru	16
1.2.1 LED	17
1.2.2 Laser	18
1.2.3 Fotocitlivé prvky senzoru	20
1.2.4 CCD snímače	20
1.3 Principy IR senzorů vzdáleností	21
1.3.1 Triangulační metoda	21
1.3.2 Měření doby letu paprsku	22
2 KONSTRUKCE ZAŘÍZENÍ	23
2.1 Senzor GP2Y0A21YK0F	23
2.2 Servomotor SG90	24
2.3 Arduino UNO	25
3 SOFTWAREOVÁ ČÁST ŘEŠENÍ	28
3.1 Program pro měření	28
3.1.1 Parametry měření	29
3.1.2 Volby vykreslování	30
3.2 Propojení s Arduinem	31
3.2.1 Propojení pomocí podpůrného balíčku	31
3.2.2 Propojení pomocí sériové komunikace	32
4 REALIZACE PROGRAMŮ	33
4.1 Realizace programu pro měření v SW MATLAB	33
4.2 Programy pro sériovou komunikaci	37
5 VÝSLEDKY EXPERIMENTŮ	42
5.1 Měření 1	42
5.1.1 Možnosti zobrazení	44

5.1.2	Měření s různými parametry	47
5.2	Měření 2	51
5.3	Měření 3	57
6	ZÁVĚR	61
	POUŽITÁ LITERATURA	62
	PŘÍLOHY	64

SEZNAM ZKRATEK A ZNAČEK

3D	trojrozměrný
CCD	zařízení s vázanými náboji
EEPROM	elektricky vymazatelná paměť pouze pro čtení
FD	fotodioda
FLASH	elektricky programovatelná paměť s libovolným přístupem
GND	uzemnění
GUI	grafické uživatelské rozhraní
I/O	vstupně výstupní
I ² C	multi-masterová sériová sběrnice
IDE	integrované vývojové prostředí
IR	infračervený
IREC	dioda emitující infračervené světlo
ISCP	možnost programování mikrokontroleru přímo v aplikaci sériovou linkou
LED	světlo emitující dioda
LIDAR	detekce a měření světla
PSD	zařízení citlivé na polohu
PWM	pulzně šířková modulace
SPI	sériové periferní rozhraní
SRAM	statická paměť
SW	software
TOF	doba letu
TTL	tranzistorově tranzistorová logika
USB	univerzální sériová sběrnice

SEZNAM SYMBOLŮ VELIČIN A FUNKCÍ

\tilde{L}	aproximovaná délka, m
\bar{L}	průměrná délka, m
c	rychlost světla, ms^{-1}
f	frekvence, Hz
I	elektrický proud, A
L	délka, m
n	proměnná
T	perioda, s
U	elektrické napětí, V
λ	vlnová délka, m
σ	směrodatná odchylka

SEZNAM ILUSTRACÍ

Obrázek 1.1 – Elektromagnetické spektrum (Khan Academy.org, nedatováno)	16
Obrázek 1.2 – Blokové schéma IR senzoru (Alldatasheet.com, nedatováno)	16
Obrázek 1.3 – Modulovaný signál IR detektoru překážek (Novák, 2005)	17
Obrázek 1.4 – Princip funkce LED (ElectroDuino.com, 2020)	18
Obrázek 1.5 – Schéma laseru (Kořán, 2014)	19
Obrázek 1.6 – Řez strukturou CCD snímače (Chlebný, 2009)	20
Obrázek 1.7 – Princip přenosu náboje v CCD snímači (Chlebný, 2009)	21
Obrázek 1.8 – Princip triangulační metody (Makerguides.com, nedatováno)	22
Obrázek 2.1 – Fotografie senzoru GP2Y0A21YK0F	23
Obrázek 2.2 – Měřicí charakteristika senzoru (Alldatasheet.com, nedatováno)	24
Obrázek 2.3 – Fotografie servomotoru SG90	25
Obrázek 2.4 – Fotografie klonu Arduino UNO	26
Obrázek 2.6 – Fotografie reálného zapojení	27
Obrázek 2.5 – Schéma zapojení senzoru a servomotoru do modulu Arduino UNO	27
Obrázek 3.1 – Snímek okna programu	28
Obrázek 4.1 – Snímek tvorby GUI za pomoci nástroje <i>guide</i>	33
Obrázek 4.2 – Výpis inicializační části kódu funkce <i>sberDat</i>	34
Obrázek 4.3 – Výpis části kódu pro první cyklus měření funkce <i>sberDat</i>	35
Obrázek 4.4 – Výpis závěru kódu funkce <i>sberDat</i>	36
Obrázek 4.5 – Výpis inicializační části programu (Arduino)	37
Obrázek 4.7 – Výpis kódu inicializace sériové komunikace (MATLAB)	38
Obrázek 4.6 – Výpis kódu inicializace sériové komunikace (Arduino)	38
Obrázek 4.9 – Výpis části kódu pro přijetí zadaných parametrů (Arduino)	39
Obrázek 4.8 – Výpis části kódu pro zadání parametrů měření (MATLAB)	39
Obrázek 4.11 – Výpis části kódu s cykly pro čtení a ukončení komunikace (MATLAB)	40
Obrázek 4.10 – Výpis části kódu pro cykly měření a ukončení komunikace (Arduino)	40
Obrázek 4.12 – Výpis příkazového řádku v prostředí MATLAB	41
Obrázek 5.1 – Fotografie měřicí plochy 1 (pohled shora)	42
Obrázek 5.2 – Fotografie měřicí plochy 1 (pohled ze strany)	43
Obrázek 5.3 – Snímek naměřené plochy 1 – zobrazení průměrných vzdáleností	44
Obrázek 5.4 – Snímek naměřené plochy 1 – zobrazení aproximovaných vzdáleností	45
Obrázek 5.5 – Snímek naměřené plochy 1 – zobrazení směrodatných odchylek	45

Obrázek 5.6 – Snímek naměřené plochy 1 – zobrazení maximálních a minimálních vzdáleností	46
Obrázek 5.7 – Snímek naměřené plochy 1 – nastavení parametrů 10/10	47
Obrázek 5.8 – Snímek naměřené plochy 1 – nastavení parametrů 45/10	48
Obrázek 5.10 – Snímek naměřené plochy 1 – nastavení parametrů 90/20	49
Obrázek 5.9 – Snímek naměřené plochy 1 – nastavení parametrů 90/1	49
Obrázek 5.10 – Snímek naměřené plochy 1 – nastavení parametrů 90/20	51
Obrázek 5.12 – Fotografie měřicí plochy 2.1 (pohled shora)	52
Obrázek 5.14 – Snímek naměřené plochy 2.1 – nastavení parametrů 90/10	53
Obrázek 5.13 – Fotografie měřicí plochy 2.1 (pohled ze strany)	53
Obrázek 5.15 – Snímek naměřené plochy 2.1 – nastavení parametrů 45/10	54
Obrázek 5.16 – Fotografie měřicí plochy 2.2	55
Obrázek 5.17 – Fotografie měřicí plochy 2.3	55
Obrázek 5.19 – Snímek naměřené plochy 2.3 – nastavení parametrů 45/10	56
Obrázek 5.18 – Snímek naměřené plochy 2.2 – nastavení parametrů 45/10	56
Obrázek 5.20 – Snímek naměřené plochy 3.1 – nastavení parametrů 90/10	57
Obrázek 5.21 – Snímek naměřené plochy 3.2 – nastavení parametrů 90/10	58
Obrázek 5.22 – Snímek naměřené plochy 3.3 – nastavení parametrů 90/10	58
Obrázek 5.23 – Snímek naměřené plochy 3.4 – nastavení parametrů 90/10	59

SEZNAM TABULEK

Tabulka 1.1 – Příměsi ovlivňující vlnovou délku záření LED (Vobecký, 2005)	18
Tabulka 3.1 – Přehled nastavitelných parametrů pro měření	29
Tabulka 3.2 – Doba trvání měření s různými parametry měření	30
Tabulka 5.1 – Předměty využité pro první blok měření	43
Tabulka 5.2 – Předměty využité pro druhý blok měření	52

ÚVOD

Bezdotykové měření je nedílnou součástí měřicí techniky už jistou řadu let, zejména pak měření vzdálenosti a teploty. Tato práce je zaměřena právě na bezkontaktní měření vzdálenosti za pomoci infračerveného záření. Jedná se o způsob měření vzdálenosti běžně užívaný v praxi pro měření malých až středních vzdáleností.

Hlavní motivací pro tuto práci je představa automatického vozítka, nebo robota, který by ke své orientaci v prostoru využíval zařízení obsahující právě infračervený senzor vzdálenosti. Skrze něj by mohl získávat informaci v jaké vzdálenosti se před ním nachází překážka. Pro rozšíření zorného pole bude senzor umístěn na hřídel malého servomotoru, díky čemuž bude možné získávat vzdálenosti překážky přímo před senzorem $\pm 90^\circ$. V této práci budou získaná data využita pro vytvoření rovinné hloubkové mapy. Dalším cílem je prozkoumání samotné komunikace mezi modulem Arduino UNO a osobním počítačem.

Součástí práce bude i měření vzdáleností tímto zařízením a zjištění pro jaké povrchy a prostředí by bylo vhodné tento senzor využít a jestli by bylo jeho použití v praxi vůbec vhodné.

1 IR SENZORY VZDÁLENOSTI

1.1 INFRAČERVENÉ ZÁŘENÍ

Pro měření vzdálenosti je u této práce zvolen infračervený senzor vzdálenosti (dále jen IR senzor). Využívání IR záření, jakožto prostředku pro bezkontaktní měření vzdálenosti, je poměrně efektivní a v praxi běžně využívané. Využívá se více metod, jak se ke konečné vzdálenosti dostat, a všechny tyto metody vyžadují ke svému fungování zdroj IR záření a přijímač IR záření.

IR záření je součástí elektromagnetického spektra, což jsou různé druhy elektromagnetického záření s různou vlnovou délkou. Samotné záření je vlnění, které se skládá ze dvou částí, a to z magnetického pole a elektrického pole. Tyto 2 složky k sobě navzájem kolmo oscilují (Khan Academy.org, nedatováno).

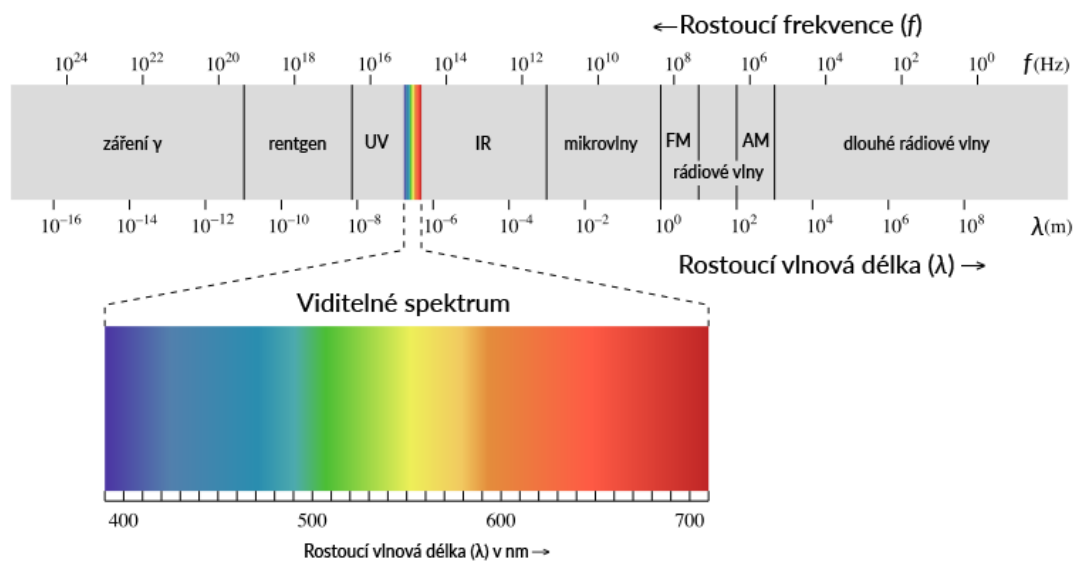
Vlnovou délku záření lze získat ze vztahu

$$\lambda = \frac{c}{f} = c \cdot T, \quad (1.1)$$

kde λ – vlnová délka, m,
 c – rychlost světla, $\text{m} \cdot \text{s}^{-1}$,
 f – frekvence, Hz,
 T – perioda, s.

Vlnové délky IR záření se obvykle pohybují od 760 nm do 1 mm. Na stupnici frekvencí se nachází pod viditelným červeným světlem. V latině slovo „infra“ znamená „pod“, proto infračervené světlo, v angličtině InfraRed = IR.

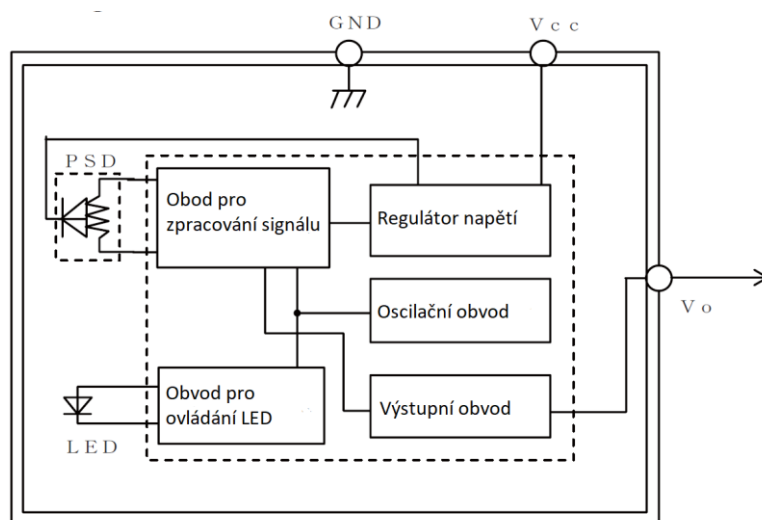
Kromě měření vzdálenosti lze IR záření použít i pro měření teploty, nebo otáček, případně se dá i využít v komunikační technice nebo ve spektroskopii.



Obrázek 1.1 – Elektromagnetické spektrum (Khan Academy.org, nedatováno)

1.2 PRVKY IR SENZORU

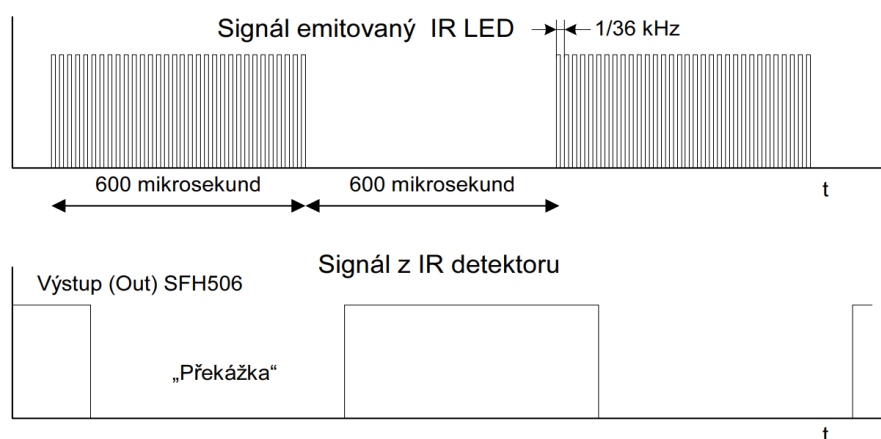
Všechny IR senzory, ať už se jedná o měření vzdálenosti, teploty, nebo i detektoru překážek, vyžadují ke své funkci dvě základní součásti. První je zdroj samotného IR záření. K tomu je možné využít záření infračervené LED, soustředěné pomocí čoček do užšího paprsku. Vhodnější variantou pro delší vzdálenosti je pak využití IR laseru. Druhou nutnou součástí je přijímač záření. Pro ten je využívá IR citlivá fotodioda, fototranzistor nebo CCD snímač. (Novák, 2005)



Obrázek 1.2 – Blokové schéma IR senzoru (Alldatasheet.com, nedatováno)

Je běžné, že součástí senzoru je modulátor a demodulátor signálu. Zdroj světla je buzen modulovaným signálem a přijímač je díky demodulátoru schopen rozpoznat pouze modulované záření ze zdroje o určité vlnové délce. Tohoto principu se v praxi využívá ke zvýšení odolnosti přijímače před okolním světlem. (Novák, 2005)

Modulace je proces, při kterém je měněn určitý parametr signálu, resp. mění se nosný signál za pomoci modulačního signálu. Pro potřeby IR senzorů je měněn kmitočet signálu pomocí modulačního signálu nejčastěji s kmitočtem 36, 38, 48 a 56 kHz. Dále je tento signál modulován nízkým kmitočtem např. 0,6 kHz. (Novák, 2005).

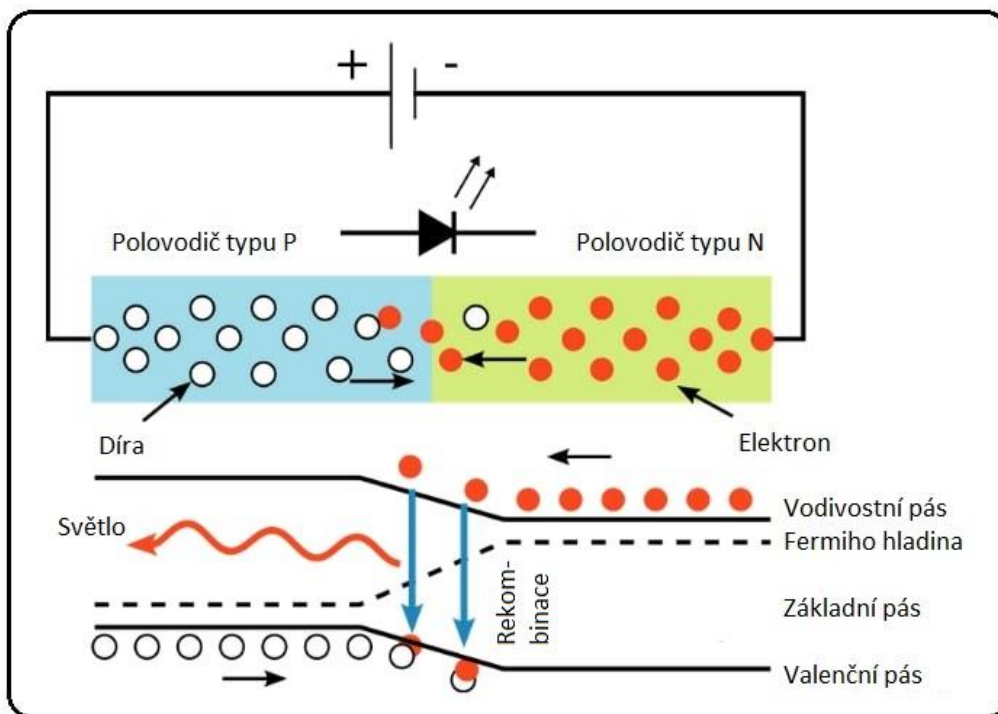


Obrázek 1.3 – Modulovaný signál IR detektoru překážek (Novák, 2005)

1.2.1 LED

LED, (zkratka z anglického „Light-Emitting Diode“ = „elektroluminiscenční dioda“), je dnes jedním z nejrozšířenějších světelných zdrojů. Jejimi přednostmi jsou především spolehlivost a nízká energetická náročnost (Vobecký, 2005).

Její funkce vychází z běžné diody, kdy při přivedení napětí v propustném směru dojde k přivedení elektronů z polovodiče N do P a děr z P do N. Na vzdálenosti difúzní délky většina elektronů a děr rekombinuje. Při tomto procesu se uvolní energie, která se vyzáří ve formě fotonu. Vlnovou délku (barvu světla) lze ovlivnit různými příměsemi v polovodiči (tabulka 1.1), (Vobecký, 2005).



Obrázek 1.4 – Princip funkce LED (ElectroDuino.com, 2020)

Tabulka 1.1 – Příměsi ovlivňující vlnovou délku záření LED (Vobecký, 2005)

Materiál	Barva světla/záření	Vlnová délka, nm	Úbytek napětí, V
SiC, GaN	Modrá	450	3,6
GaP	Zelená	565	2,2
GaAs _{0,15} P _{0,85} :N	Žlutá	585	2,1
GaAs _{0,35} P _{0,65} :N, GaAs _{0,6} P _{0,4} , GaP:Zn-O	Červená	635	2,0
SiC/GaN + luminofor na povrchu	Bílá	450-650	3,6
GaAs:Si	IR záření	820-950	1,5

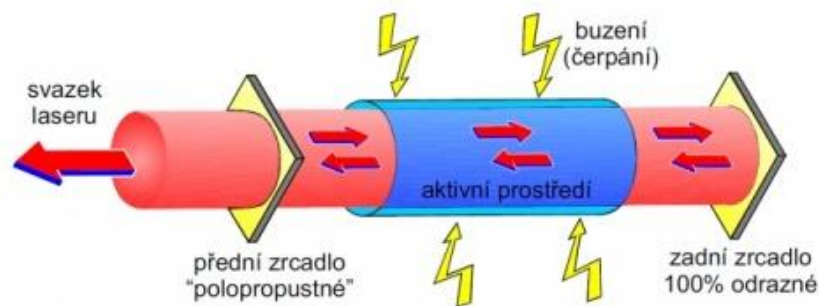
1.2.2 Laser

Laser je akronym z anglického „Light Amplification by Stimulated Emission of Radiation“ = „zesilování světla stimulovanou emisí záření“. Na rozdíl od záření z LED je záření z laseru koherentní, tj. elektromagnetické vlnění, jehož vlny jsou ve stejném směru, frekvenci i fázi (Kořán, 2014).

Laser pro své fungování vyžaduje:

- 1) Aktivní prostředí – Obsahuje oddělené kvantové energetické hladiny elektronů. Může se jednat o pevnou, kapalnou i plynnou látku.
- 2) Rezonátor – Skládá se ze dvou rovnoběžně umístěných zrcadel, z nichž je jedno nepropustné a druhé polopropustné. Zrcadla jsou umístěna mezi aktivním prostředím a kolmo k paprsku.
- 3) Zdroj záření – Slouží k dodání energie elektronům v aktivním prostředí např. za pomoci výbojky.

Základní schéma laseru



Obrázek 1.5 – Schéma laseru (Kořán, 2014)

Celý proces pak funguje následovně. Při zažehnutí výbojky přeskóčí většina elektronů v aktivním prostředí na vyšší energetickou hladinu. Elektrony se následně snaží dostat zpět do své původní energetické hladiny a dochází ke spontánní emisi fotonů. Vyzářené fotony následně putují náhodným směrem, přičemž mohou narazit na elektron ve vyšší energetické hladině a způsobit tzv. stimulovanou emisi, která vytvoří nový foton. Tento nový foton pak putuje stejným směrem jako ten původní. V případě, že fotony směřují mimo zrcadla, jsou absorbovány, ale pokud dopadnou na jedno ze zrcadel, dojde k řetězové reakci a začne vznikat velké množství fotonů pohybující ve stejném směru mezi zrcadly. Část fotonů je následně uvolněna polopropustným zrcadlem a vznikne tak laserový paprsek (Kořán, 2014).

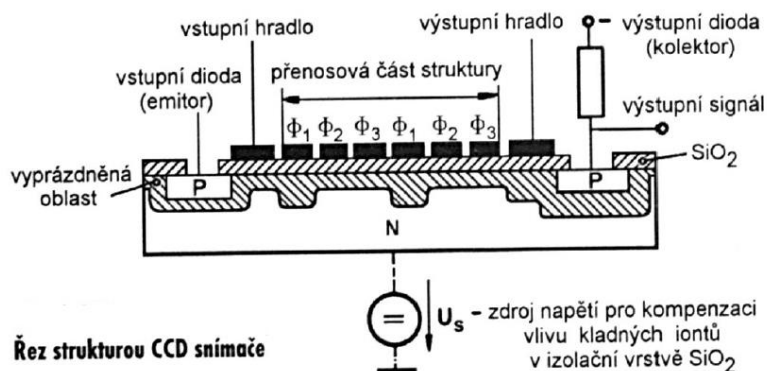
1.2.3 Fotocitlivé prvky senzoru

Nejběžnějším fotocitlivým prvkem je fotocitlivá dioda, která funguje na principu běžné diody a fotoelektrického jevu. V případě, kdy záření dopadá na speciálně upravený PN přechod, vytvoří se páry elektron-díra, které generuje dodatečný proud, resp. snižují elektrický odpor polovodiče. Pokud na FD světlo nedopadá, chová se jako běžná dioda (Vobecký, 2005).

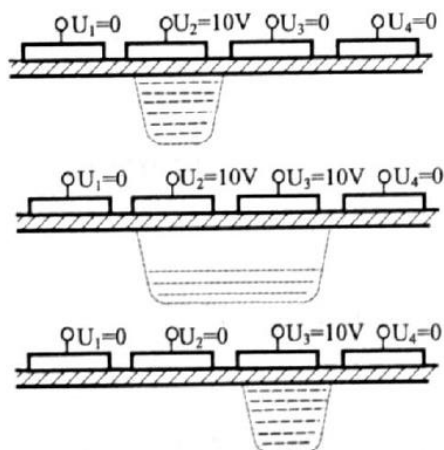
Podobně jako fotodioda má fototranzistor upraven přechod báze-kolektor, aby na něj mohlo dopadat záření. Kolektorový proud je zde řízen intenzitou záření dopadající na oblast emitorového přechodu (Vobecký, 2005).

1.2.4 CCD snímače

CCD (zkratka z anglického „Charge Coupled Devices“ = zařízení s vázanými náboji) snímače jsou určeny ke snímání obrazové informace. Podobně jako předchozí součástky využívá ke své funkci fotoelektrický jev. Senzory se skládají z vodivé vrstvy N (křemík) a nevodivé vrstvy SiO₂ (oxid křemičitý), na které se nachází řada vzájemně izolovaných transparentních elektrod tvořící síť pixelů (obrázek 1.6). Při osvětlení pixelu dochází ke vzniku elektrického náboje, který je úměrný ozáření a je přitahován kladně napájenou elektrodou. Postupným připojováním a odpojováním napětí na jednotlivých elektrodách lze dovést vzniklé náboje na výstup snímače (obrázek 1.7), kde dojde k převedení náboje na napětí (Chlebný, 2009).



Obrázek 1.6 – Řez strukturou CCD snímače (Chlebný, 2009)



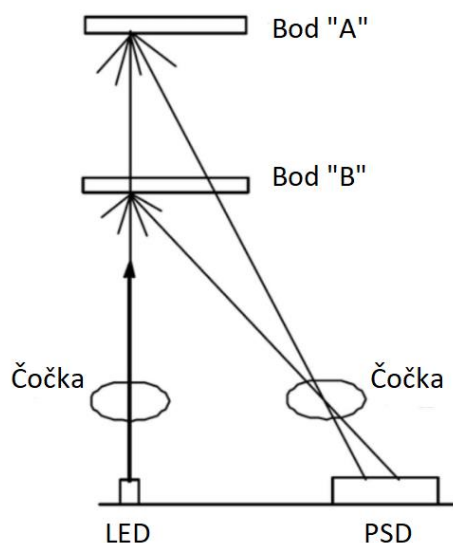
Obrázek 1.7 – Princip přenosu náboje v CCD snímači (Chlebný, 2009).

1.3 PRINCIPY IR SENZORŮ VZDÁLENOSTI

Nejprve je třeba rozlišit měření konkrétních vzdáleností a měření vzdálenosti ve smyslu detekce překážky od určité vzdálenosti. Druhá možnost je principiálně jednodušší. Detekci překážky lze provést například pomocí odrazu paprsku od překážky, kdy k odrazu paprsku dojde pouze v určité oblasti před senzorem, ne nutně na konkrétním místě. Lze tedy jen říci, že máme např. senzor s dosahem 1 m a ve vzdálenosti 0,5 m před něj postavíme překážku. V tuto chvíli senzor pouze ví, že někde mezi jeho minimální a maximální vzdálenosti, kterou je schopen rozpoznat, je překážka. Tento druh senzorů se nazývá detektor překážky. Pro získání konkrétní vzdálenosti překážky od senzoru se využívá několik sofistikovanějších principů. (Novák, 2005; Siciliano, 2008).

1.3.1 Triangulační metoda

Měření vzdálenosti IR senzorem s triangulační metodou se podobá předchozí popisované detekci překážky. Spočívá ve vysílání paprsku infračerveného světla a zachycení jeho následného odrazu od překážky. Pro vysílání IR paprsku je využívána LED, jejíž záření je soustředěno pomocí čočky. Odraz paprsku od překážky dále dopadá skrze čočku přijímače na CCD snímač. V tomto případě záleží, na jaké místo snímače odraz dopadne, jelikož podle nejvíce ozářeného pixelu snímače je vypočítána hodnota napětí, reprezentující změřenou vzdálenost. Na intenzitě odrazu zde nezáleží (Novák, 2005; Siciliano, 2008).



Obrázek 1.8 – Princip triangulační metody (Makerguides.com, nedatováno)

1.3.2 Měření doby letu paprsku

Sensory využívající princip doby letu paprsku se přezdívá LIDAR (Light Detection And Ranging), nebo ToF (Time of Flight) senzory. Funkcí se podobají radarům. Je měřen čas, za který světelný paprsek po vyslání ze světelného zdroje narazí na překážku a odrazí se na přijímač. Omezení těchto snímačů je minimální vzdálenost, respektive minimální čas, který jsme schopni zaznamenat. Další nevýhodou je i vyšší cena než u ostatních IR senzorů. Výhodou je možnost měření pohybujících se předmětů (Siciliano, 2008).

Pro měření času je buď využito pulzního laseru, kdy je přímo měřen čas od vyslání a přijetí paprsku. Tato metoda vyžaduje elektroniku schopnou rozlišovat pikosekundy, navíc zařízení využívající tuto metodu bývají velmi drahá a dosti nepřesná. (Siciliano, 2008).

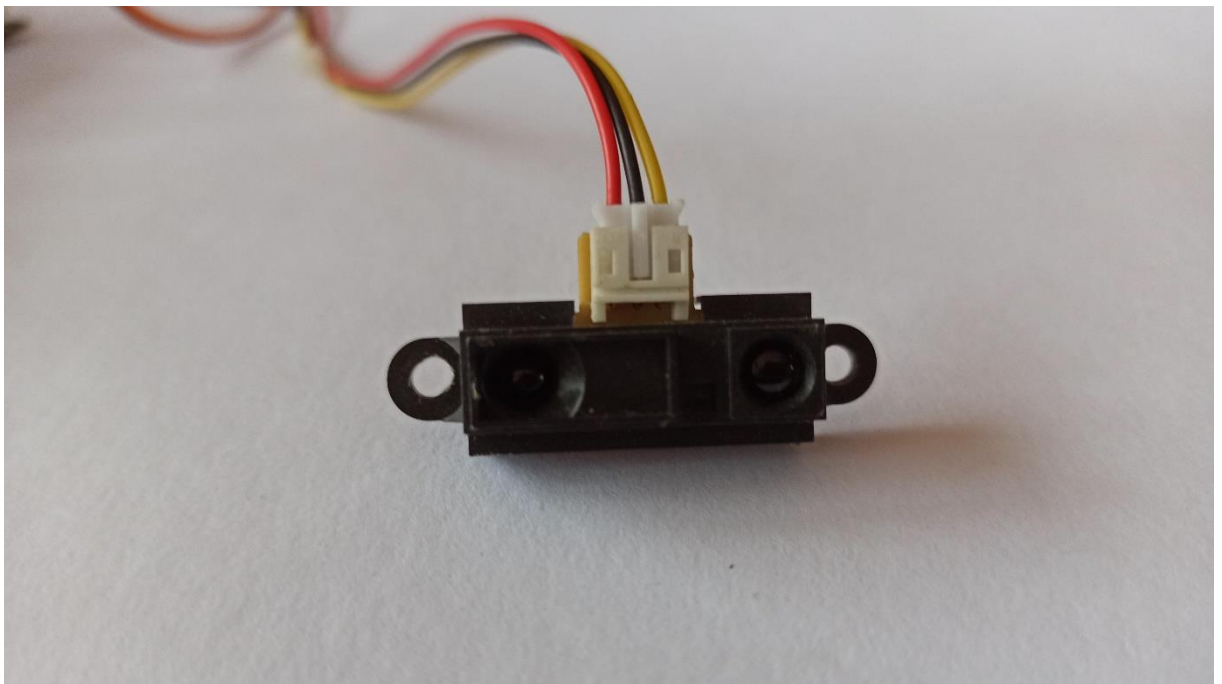
Druhou možností je měření fázového posunu. V tomto případě je porovnáván vyslaný signál se signálem odraženým. Ve chvíli odrazu vyslaného paprsku dojde k jeho fázovému posuvu a při porovnání původního a odraženého signálu získáme jejich rozdíl, který je úměrný době letu paprsku. Touto metodou lze dosáhnout velmi přesných výsledků (Siciliano, 2008).

2 KONSTRUKCE ZAŘÍZENÍ

Zařízení se skládá z infračerveného senzoru vzdálenosti GP2Y0A21YK0F a servomotoru SG90. Obě periferie jsou ovládány pomocí klonu modulu Arduino UNO, který je pomocí USB spojen s nadřazeným počítačem.

2.1 SENZOR GP2Y0A21YK0F

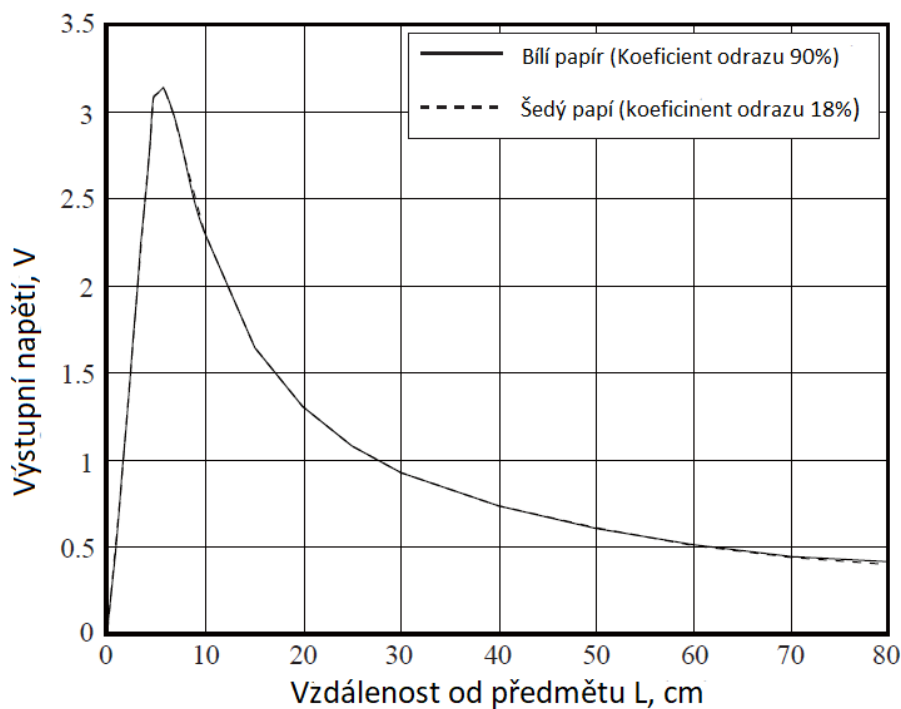
Pro praktickou část byl jako senzor zvolen infračervený senzor Sharp GP2Y0A21YK0F. Pracuje na základě triangulační metody a skládá se z detektoru místa osvit, IRED a obvodů pro zpracování signálu. Obsahuje 3 vývody, dvojici napájecích kabelů s označením V_{CC} (červený vodič) a GND (černý vodič) a analogový výstup s označením V_0 (žlutý vodič) (Dratek.cz, nedatováno).



Obrázek 2.1 – Fotografie senzoru GP2Y0A21YK0F

Jeho měřicí rozsah je od 10 do 80 cm s předpokládanou odchylkou $\pm 0,3$ mm. Tomuto rozsahu odpovídá výstupní napětí $V_0 = 2,3$ až $0,4$ V. Pro jeho napájení je vyžadováno napětí $V_{CC} = ,5$ až $5,5$ V, při proudu $I_{CC} = 30$ mA. Celkové rozměry senzoru jsou $29,5 \times 13 \times 13,5$ mm (Dratek.cz, nedatováno).

Na výstupu senzoru je analogový signál, který reprezentuje naměřenou vzdálenost. Na obrázku 2.2, lze vidět, jak s rostoucí vzdáleností postupně klesá napětí. Dále je možné si všimnout, že průběh obsahuje i hodnoty pro vzdálenost menší než 10 cm. Tuto část však není možné využít, jelikož hodnoty napětí se v této části překrývají s hodnotami pro vyšší vzdálenosti.



Obrázek 2.2 – Měřicí charakteristika senzoru (*Alldatasheet.com*, nedatováno)

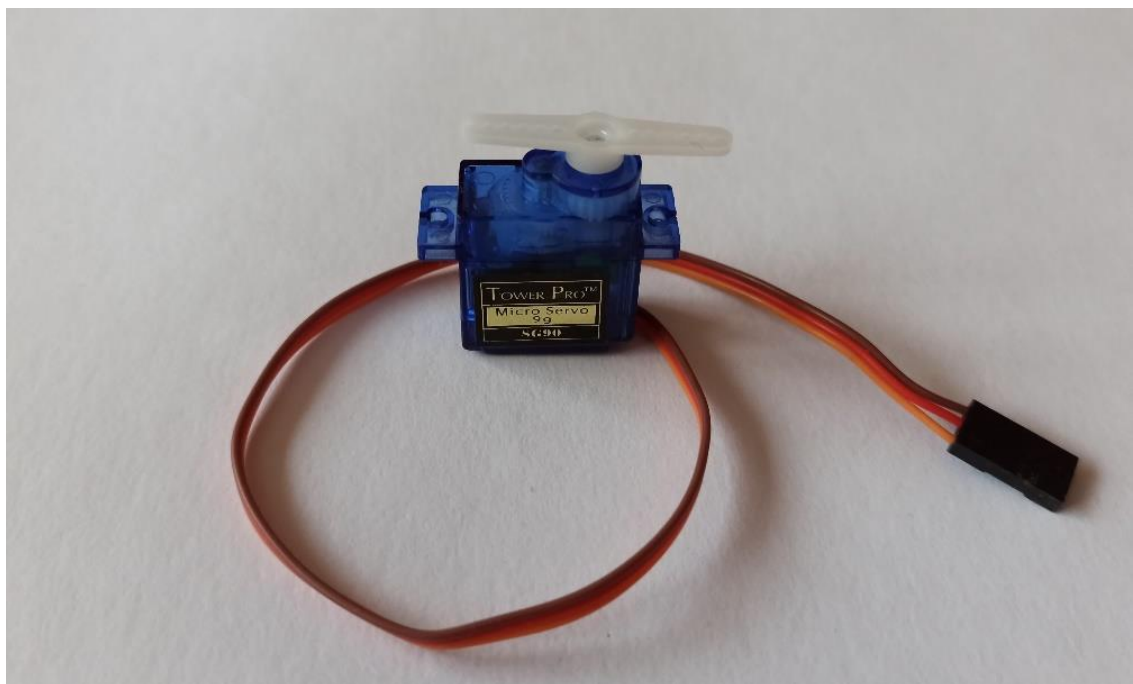
Výrobce uvádí, že senzor je určen pro využití jako bezdotykový přepínač, např. pro ovládání osvětlení, senzor pro úsporu energie a orientaci robotů. Dále je řečeno, že výsledky měření nejsou díky triangulační metodě zásadně ovlivněny reflexivitou objektu, okolní teplotou ani dobou provozu (*Alldatasheet.com*, nedatováno).

2.2 SERVOMOTOR SG90

Pro měření vzdáleností na ploše je senzor umístěn na hřídeli modelářského servomotoru. Tento druh motorů je velice často využíván při stavbě tohoto druhu zařízení, kvůli jejich spolehlivosti, funkčnosti, a především malé pořizovací ceně (Novák, 2005).

Obecně se servomotory skládají z drobného stejnosměrného motoru, drobného převodu, řídicího obvodu a výstupní hřídele, na kterou je připojen potenciometr pro sledování úhlu natočení. Hodnota úhlu natočení je obvykle šířkově modulovaný signál na úrovni TTL s periodou 20 ms a šířkou pulzu od 1 ms (pro maximální levé natočení) do 2 ms (maximální pravé natočení), střední hodnota je pak 1,5 ms. Většina modelářských servomotorů má krouticí moment o hodnotě 30 Ncm/4,8 V až 40 Ncm/ 6 V. Rozsah úhlu natočení bývá $\pm 90^\circ$, ale můžou být i jiné (Novák, 2005; Chlebný, 2009)

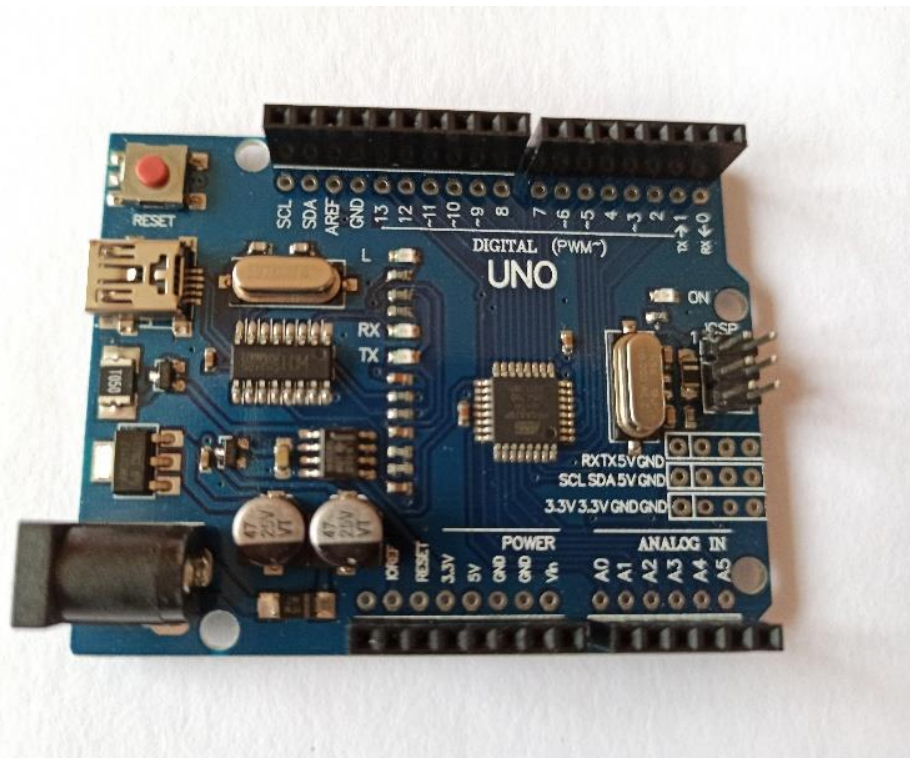
Pro účely této práce byl zvolen servomotor Servo SG90 9g micro motor. Jedná se o jednoduchý a cenově dostupný prostředek pro natáčení našeho senzoru na ose 180° . K Arduinu je připojen třemi vodiči. Oranžový vodič s označením MWM slouží k ovládání servomotoru, červený s označením VCC je napájení a hnědý GND je zem (Drátek.cz, nedatováno).



Obrázek 2.3 – Fotografie servomotoru SG90

2.3 ARDUINO UNO

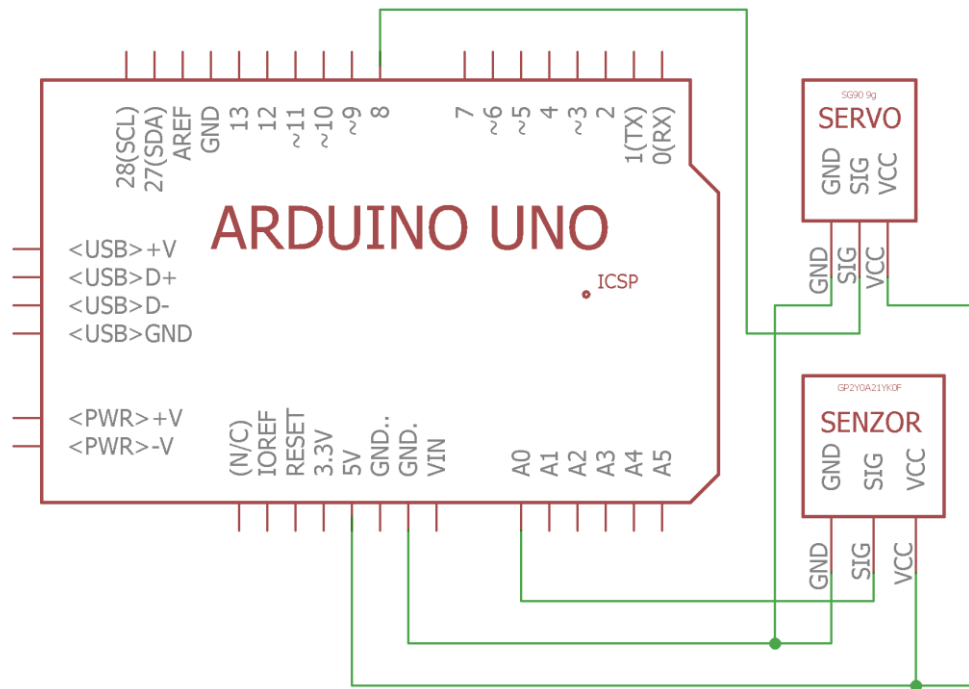
Pro spojení uživatelského počítače se senzorem a servomotorem je využit mikrokontroler Arduino UNO, respektive jeho klon Arduino UNO R3. Klon je dle prodejce plně kompatibilní s originálem a pro účely této práce je zcela postačující (Drátek.cz, nedatováno).



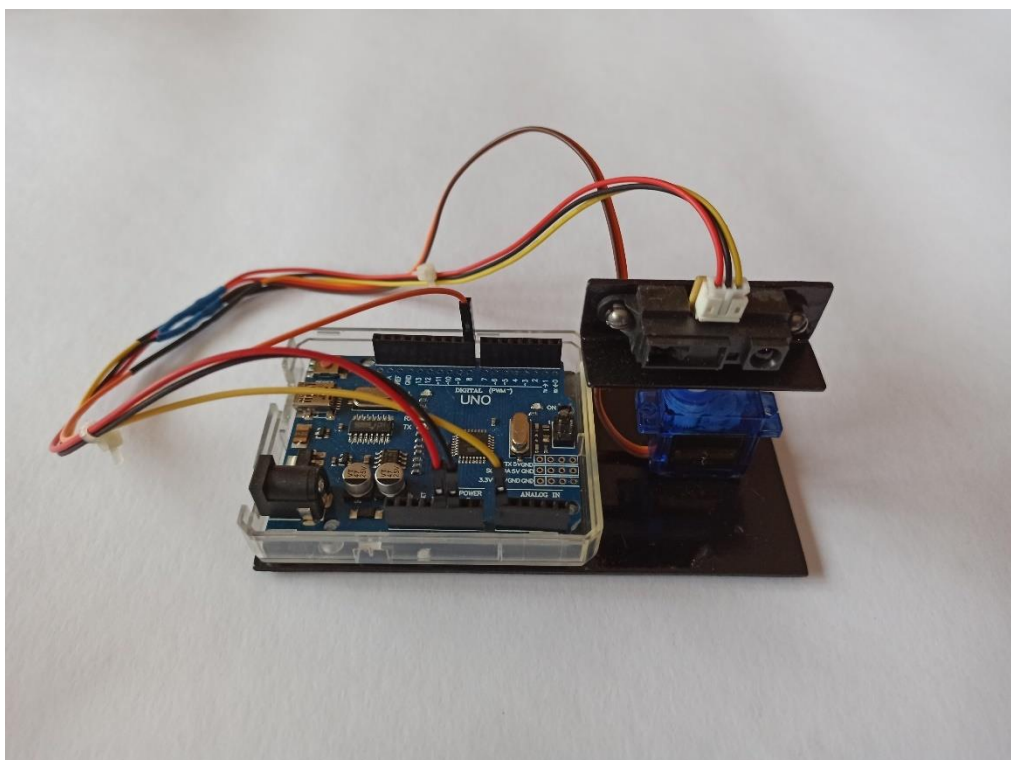
Obrázek 2.4 – Fotografie klonu Arduino UNO

Modul obsahuje procesor ATmega328, který využívá 3 různé paměti, Flash = 32 kB (z toho 0,5 kB je využito na bootloader), SRAM = 2 kB a EEPROM = 1 kB. Dále deska obsahuje 14 digitálních I/O pinů (6 z nich lze použít jako výstupy PWM), 6 analogový vstupů, 16 MHz krystal, mini USB konektor, napájecí konektor ISCP rozhraní a tlačítko pro reset (Drátek.cz, nedatováno).

Pro účely této práce je využit konektor mini USB pro napájení desky a ostatních komponent a pro sériovou komunikaci. Dále digitální pin 8 pro servomotor a analogový pin A0 pro senzor a samozřejmě napájecí konektory 5 V a GND, pro obě komponenty.



Obrázek 2.6 – Schéma zapojení senzoru a servomotoru do modulu



Obrázek 2.5 – Fotografie reálného zapojení

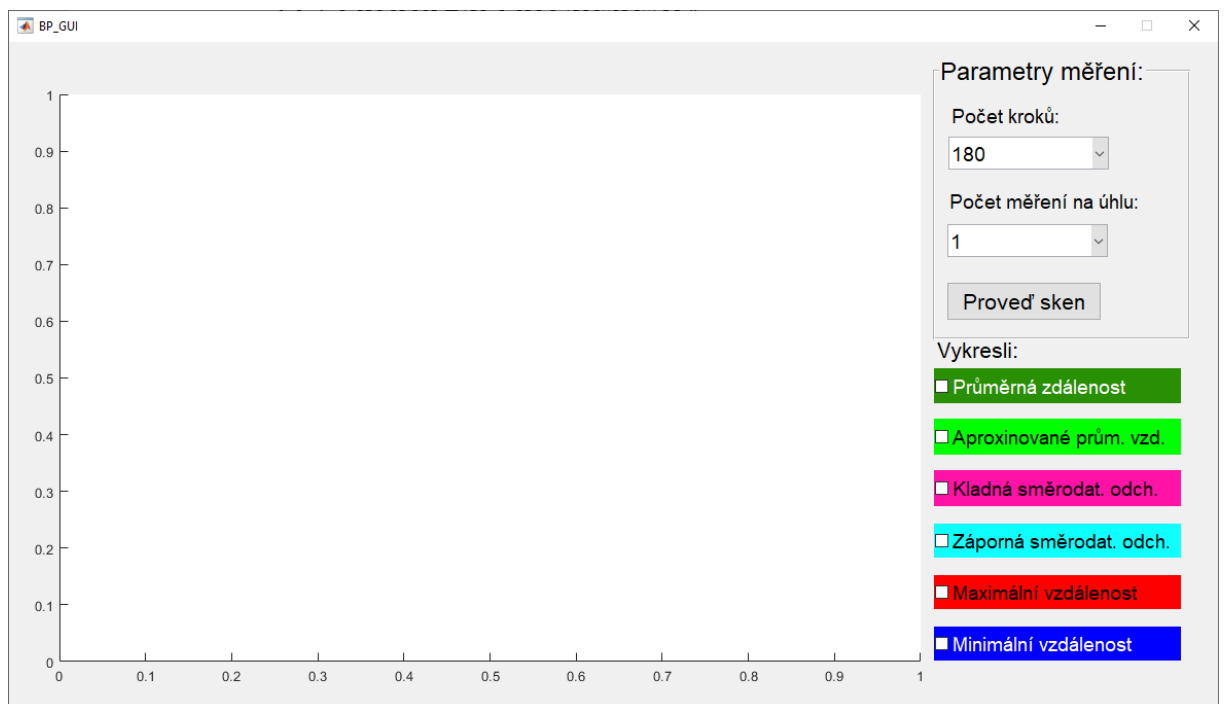
3 SOFTWAREOVÁ ČÁST ŘEŠENÍ

Hlavní softwarové řešení je kompletně zpracováno v SW MATLAB. Při tomto řešení modul nevyžaduje nahrání žádného programu a veškerá práce s modulem a připojenými perifériemi probíhá v prostředí MATLAB, za pomoci podpůrného balíčku.

Dále je popsána možnost řešení za pomoci sériové komunikace, kdy je v modulu Arduino nahrán program, který je připraven pro sériovou komunikaci. Druhým zařízením pro komunikaci bude opět PC s aplikací v SW MATLAB. Pro obě řešení budou vysvětleny veškeré funkce programů.

3.1 PROGRAM PRO MĚŘENÍ

Při spuštění hlavního programu BP_GUI, je vykresleno grafické uživatelské rozhraní obsahující plochu pro vykreslení grafu (hloubkové rovinné mapy), dvojici rolet s dostupnými parametry měření (počet kroků a počet měření na úhlu) a 6 zaškrtačkových políček pro vykreslování hloubkové rovinné mapy.



Obrázek 3.1 – Snímek okna programu

Po zvolení parametrů měření a stisknutí tlačítka „Proved' sken“ je započat proces měření vzdáleností a nastavování pozice servomotoru. Senzor provede počet skenů podle zvoleného druhého parametru a následně servomotor otočí hřídelí doleva o jeden krok. Počet kroků, které vykoná je určen prvním parametrem. Po změření vzdáleností na posledním kroku se celý proces opakuje v opačném směru. Po dokončení měření jsou připravena veškerá potřebná data pro vykreslování grafu. O stavu měření informuje program uživatele na příkazovém řádku. Po dokončení měření může uživatel kliknutím na jednotlivá zaškrťovací políčka vykreslit graf. Barvy pozadí jednotlivých možností vykreslení odpovídají průběhům v grafu. Program dovoluje vykreslení více průběhů najednou.

3.1.1 Parametry měření

Parametry zvolí uživatel před započítím nového měření. V první kolonce vybere požadovaný počet kroků prováděné servomotorem. Samotné měření začíná již od nulového úhlu, proto je výsledný počet získaných vzdáleností roven počtu kroků + 1. Druhý parametr určí kolikrát je změřena vzdálenost na každém úhlu a následně je z těchto hodnot vypočítán průměr, směrodatná odchylka, minimum a maximum.

Tabulka 3.1 – Přehled nastavitelných parametrů pro měření

Počet kroků	Počet měření na úhlu
180	1
90	2
45	5
18	10
10	20

Při výběru počtu kroků je třeba brát v potaz velikost objektů, které senzor dokáže zachytit. Bude-li počet kroků nastaven na hodnotu 18 a bude-li umístěn předmět o šířce 2 cm do vzdálenosti 50 cm před senzor, je pravděpodobné, že senzor předmět nezachytí. V opačném případě, když bude počet kroků nastaven na 180, tak zmíněný předmět sice senzor určitě zachytí, ale zobrazení bude obsahovat větší počet nerovností a doba měření bude mnohem vyšší.

U měření na úhlu je obecně lepší volit hodnotu 5 a více, jelikož pokud bude zvolena menší hodnota, budou dodatečné možnosti zobrazení, kromě aproximace, neplatné. Při vyšším počtu měření zase vzniká více chybných měření.

Dále je třeba zmínit dobu trvání měření s různými parametry. Je jasné, že s vyššími čísly nastavovaných parametrů vzrůstá i doba trvání měření. Pro některé zvolené parametry jsou doby trvání měření uvedeny v tabulce 3.2.

Tabulka 3.2 – Doba trvání měření s různými parametry měření

Počet kroků	Počet měření na úhlu	Doba měření, s
180	20	243,58
180	5	71,97
90	5	37,99
45	10	35,40
10	20	17,94
10	1	4,80

3.1.2 Volby vykreslování

Po dokončení měření je možno vybrat z několika možností zobrazení. První a nejdůležitější je zobrazení průměrných vzdáleností. Jedná se o primární výsledek měření a jednotlivé hodnoty jsou vypočítány jakožto aritmetický průměr z naměřených hodnot zvoleným parametrem „Počet měření na úhlu“ a hodnot z obou směrů měření. V grafu je vykreslena tmavě zelenou barvou.

Dalším zobrazením je aproximace průměrných hodnot. Toto zobrazení slouží k uhlazení průměrných hodnot v grafu a získáním tak lepší představy o měřené ploše. Aproximaci získáme jakožto průměr ze tří hodnot. Toto lze provádět od třetí hodnoty, první 2 hodnoty aproximovány nejsou. V grafu jsou aproximace vykresleny světle zelenou barvou.

Zbývající čtyři volby jsou pro vykreslení směrodatných odchylek (přičtených a odečtených od průměru) a maximální a minimální hodnoty. Všechny tyto hodnoty jsou vypočítány nebo získány z hodnot získaných při měření na jednom úhlu a následným zprůměrováním obou otočení. Přičtená směrodatná odchylka má v grafu růžovou barvu, odečtená tyrkysovou, maximální červenou a minimální modrou.

3.2 PROPOJENÍ S ARDUINEM

Pro hlavní řešení bylo propojení uskutečněno za pomoci podpůrného balíčku MATLAB Support Package for Arduino Hardware. Tento způsob je ovšem jednoúčelový a pro jeho fungování je zapotřebí využívat software MATAB s nainstalovaným podpůrným balíčkem.

Druhou, víceúčelovou volbou, je využití sériové komunikace. V rámci této práce byl vytvořen návrh programů propojující PC s modulem skrze sériovou komunikaci. Toto řešení neobsahuje veškeré potřebné funkce a slouží pouze k demonstraci možného řešení.

3.2.1 Propojení pomocí podpůrného balíčku

Jak bylo již zmíněno, hlavní řešení je vytvořeno v softwaru MATLAB, a to bez potřeby programování samotného modulu Arduino. Toto je možné díky podpůrnému balíčku MATLAB Support Package for Arduino Hardware, která dovoluje interagovat a komunikovat s moduly Arduino a jejich perifériemi.

Balíček umožňuje komunikaci s modulem, čtení dat ze senzorů, práci s motory a servomotory, práce s registry modulu, podporu SPI a I²C a podporu řady specifických senzorů. Pro potřeby této práce je využito vytvoření objektu *arduino*, jehož prostřednictvím je možné přistupovat k perifériím modulu. Pro servomotor je ještě třeba vytvořit objekt *servomotor*, který obsahuje objekt *arduino* a číslo portu, kde je servomotor na modulu připojen. Následně je možné příkazem *writePosition* nastavovat pozici hřídele. Pro čtení napětí ze senzoru stačí využít příkaz *readVoltage* s objektem *arduino* a číslem portu, kde je senzor připojen. V práci jsou tedy využity následující funkce balíčku:

- vytvoření objektu Arduino, příkaz: $a = \text{arduino}(\text{COM3}, \text{UNO})$,
- vytvoření objektu servomotoru příkaz: $\text{servomotor} = \text{servo}(a, \text{D8})$,
- nastavení pozice servomotoru, příkaz: $\text{writePosition}(\text{servomotor}, \text{pozice})$,
- získání hodnoty napětí na senzoru, příkaz: $V = \text{readVoltage}(a, \text{A0})$.

Výhodou tohoto řešení je možnost využití funkcí softwaru MATLAB přímo v průběhu měření, bez potřeby přenášení pouhých výsledků. Práce s daty je tedy mnohem jednodušší

a přehlednější. Jasnou nevýhodou je podmínka instalace sw MATLAB s příslušným balíčkem, bez těchto prostředků bude program nepoužitelný.

3.2.2 Propojení pomocí sériové komunikace

Programovou část práce je možné řešit i pomocí sériové komunikace, kdy již naprogramovaný modul komunikuje s nadřazeným počítačem, skrze prostředí MATLAB, pomocí sériové linky. Toto řešení se dá označit jako obecnější, jelikož k práci s modulem je možno využít i jiné prostředí nebo nástroje než právě MATLAB. Oproti hlavnímu programovému řešení má řešení se sériovou komunikací tu výhodu, že k fungování nevyžaduje žádné dodatečné nástroje.

Základním kamenem sériové komunikace je posílání dat (bitů) za sebou po jednom vodiči. Pro komunikaci mezi dvěma zařízeními na jedné sériové lince je třeba použít příslušný protokol, který určí kdy, jaké zařízení vysílá a jaké přijímá. Přenos na sériové lince pak může být synchronní, asynchronní nebo arytmičkový (Peterka, 1996; Chlebný, 2009).

Synchronní přenos spočívá v synchronizaci vysílače a přijímače pomocí hodinového signálu. Bity jsou vysílány v konstantním rytmu v intervalech o stejné délce a přijímač je ve stejném rytmu přijímá. Jednotlivé intervaly pak obsahují bity tvořící jednu značku. Pro správné fungování je na straně odesílatele generátor synchronizačních pulzů (Peterka, 1996; Chlebný, 2009).

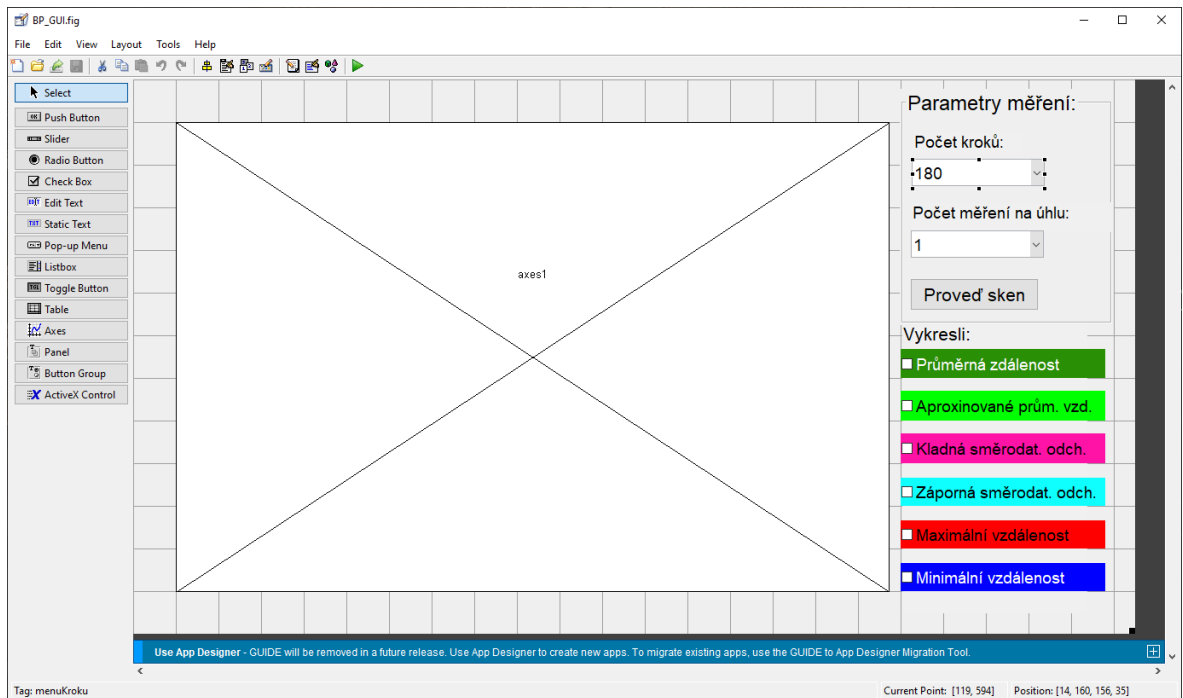
Další způsob je asynchronní přenos, při kterém nejsou intervaly odděleny synchronizačními pulzy, ale odesílatel informuje příjemce, kdy interval začíná a kdy končí. Docílí toho tím, že jednotlivé intervaly rozdělí speciálními bity, vyjadřující začátek a konec intervalu. V tomto případě mohou být intervaly různě dlouhé (Peterka, 1996; Chlebný, 2009).

Poslední možností je arytmičkový přenos, který kombinuje oba předchozí způsoby přenosu. Při přenosu jsou přenášeny pevně stanovené bloky bitů ohraničené start a stop bitem. Start bit spouští časovou základnu přijímače a stop bit ji zase ukončuje. Přenos samotného bloku bytů je však zcela asynchronní, přičemž dochází k desynchronizaci vysílače s přijímačem a nemůže tak obsahovat velké množství znaků (Peterka, 1996; Chlebný, 2009).

4 REALIZACE PROGRAMŮ

4.1 REALIZACE PROGRAMU PRO MĚŘENÍ V SW MATLAB

Program vychází z grafického rozhraní vytvořeného pomocí nástroje *guide*, kde je v první řadě vytvořen soubor s příponou *.fig*, obsahující vzhled okna programu. Na samotné ploše okna se pak nachází rolety s možnostmi volby počtu kroků a počtu měření na jednom úhlu, 6 zaškrťovacích políček pro volbu zobrazení a plochu pro vykreslení grafu.



Obrázek 4.1 – Snímek tvorby GUI za pomoci nástroje *guide*

Po vytvoření souboru obsahující GUI je zároveň automaticky vytvořen soubor s příponou *.m*, který obsahuje program s úvodní inicializační funkcí pro spuštění okna a komentáře s popisem funkcí. Dále soubor obsahuje funkce *callback*, které vykonávají naprogramovanou činnost, pokud dojde k události spojené s daným prvkem, např. zmáčknutí tlačítka nebo zaškrtnutí políčka.

Celý proces funguje následovně. Po spuštění programu *BP_GUI.fig* se zobrazí okno programu. Uživatel může vybrat z předdefinovaných možností měření (počet kroků a počet měření na úhlu) a stiskne tlačítko „Proveď sken“. Na příkazovém řádku jsou vypsány zvolené parametry měření a je volána funkce *sberDat*.

Funkce *sberDat* nejdříve na příkazovém řádku napíše „Probíhá měření...“. Dále vytvoří objekty *a* pro Arduino a *servomotor* pro servomotor, za pomoci balíčku MATLAB Support Package for Arduino Hardware. Na konci inicializační části funkce jsou pak definovány potřebné proměnné a matice obsahující parametry zvolené uživatelem.

```

55 %Funkce pro provedení měření vzdáleností.
56 function sberDat(kroky,opakovani)
57 disp('Probíhá měření...');
58 %Inicializace funkcí knihovny MATLAB Support Package for Arduino Hardware.
59 global a;
60 a = arduino('COM3','UNO');
61 servomotor = servo(a,'D8');
62 %-----
63 %Iniciakizace pomocných proměnných a matic.
64 i = 1;
65 j = 1;
66 rozestupy = zeros(1,kroky+1);
67 maxVz = zeros(2,kroky+1);
68 minVz = zeros(2,kroky+1);
69 prumer = zeros(2,kroky+1);
70 aproxVzd = zeros(1,kroky+1);
71 mereniVzd = zeros(1,opakovani);
72 A = zeros(2,opakovani);
73 smerDatOdch = zeros(2,kroky);

```

Obrázek 4.2 – Výpis inicializační části kódu funkce *sberDat*

Dále probíhá již samotné nastavování pozice hřídele servomotoru a měření vzdáleností. Jedná se o 2 vnější cykly pro nastavení pozice hřídele, jeden pro pohyb doprava a druhý doleva a 2 vnitřní cykly pro měření vzdáleností. Počet opakování vnějších cyklů odpovídá počtu kroků zvolený uživatelem a počet opakování vnitřních cyklů pak zvolenému počtu měření.

Vnější cyklus vždy začne nastavením pozice hřídele servomotoru (začíná na 0). Dále nastává cyklus měření, kdy je pomocí příkazu *readVoltage* přečtena hodnota napětí na senzoru a proběhne vypočítání vzdálenosti, která je zapsána do matice *mereniVzd*, na pozici odpovídající pořadí měření. Pro výpočet vzdálenosti je využit vztah

$$L = 29,988 \cdot U^{-1,173}, \quad (3.1)$$

kde L – délka, m,
 U – napětí, V.

Následně je hodnota ohraničena podmínkou, aby naměřená hodnota nemohla být větší než 81 a menší než 9. Po dokončení vnitřního cyklu je z naměřených hodnot do matice *maxVz* uložena nejvyšší naměřená hodnota, do *vzMin* minimální naměřená hodnota a do *prumer* aritmetický průměr z naměřených hodnot na tomto kroku. Následně je vypočítána směrodatná odchylka z naměřených vzdáleností, která je uložena do matice *smérDatOdch*. Její výpočet je podle vztahu

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (L_i - \bar{L})^2}, \quad (3.2)$$

kde σ – směrodatná odchylka,
 n – počet vzdáleností,
 L_i – délka, m,
 \bar{L} – průměrná délka, m.

```

75 % Cykly pro provedení měření.
76 for theta = 0:1/kroky:1
77     writePosition(servomotor,theta);
78     for k=1:1:opakovani
79         U = readVoltage(a,'A0');
80         mereniVzd(k) = 29.988 * U^(-1.173);
81         % Omezení minimální a maximální hodnoty vzdálenosti.
82         if mereniVzd(k) >= 81
83             mereniVzd(k) = 81;
84         elseif mereniVzd(k) <= 9
85             mereniVzd(k) = 9;
86         end
87     end
88     maxVz(1,i) = max(mereniVzd);
89     minVz(1,i) = min(mereniVzd);
90     prumer(1,i) = sum(mereniVzd)/opakovani;
91     % Výpočet směrodatné odchylky.
92     for k=1:1:opakovani
93         A(1,k) = ((mereniVzd(k)-prumer(1,i))^2);
94     end
95     smerDatOdch(1,i) = (sum(A(1,:))/(opakovani-1))^(1/2);
96     rozestupy(1,i) = (i-1);
97     i = i+1;
98 end
99
100 for theta = 1:-1/kroky:0
101     writePosition(servomotor,theta);
102     for k=1:1:opakovani

```

Obrázek 4.3 – Výpis části kódu pro první cyklus měření funkce

Na závěr vnitřního cyklu je přidána pozice pro zobrazení na grafu do matice *rozestupy* a inkrementace pomocné proměnné *i*. Cykly pro opačný směr probíhají obdobně, pouze jsou hodnoty zapisovány v opačném směru a je inkrementována pomocná proměnná *j*, která je odečítána od *i*. Po dokončení obou cyklů jsou všechny získané hodnoty zprůměrovány a na příkazovém řádku je vypsána hláška „Měření dokončeno.“. Po dokončení měření proběhne výpočet aproximovaných vzdáleností pomocí vztahu

$$\tilde{L} = \frac{1}{3} \sum_{i=3}^n (L_{i-2} + L_{i-1} + L_i), \quad (3.3)$$

kde \tilde{L} – aproximovaná vzdálenost.

Na konci funkce jsou veškerá potřebná data pro vykreslování grafu uložena, aby k nim měla zaškrťovací políčka přístup. Zároveň jsou matice obsahující vykreslované vzdálenosti uloženy do workspace pro možnost zobrazení konkrétních hodnot.

```

122 - vzdalenosti(1,:) = prumer(1,:);
123 - vzdalenosti(2,:) = prumer(2,:);
124
125 % Součet a zprůměrování vzdáleností z obou směrů
126 - maxVz = (maxVz(1,:) + maxVz(2,:))/2; %Matice maximálních vzdáleností.
127 - minVz = (minVz(1,:) + minVz(2,:))/2; %Matice minimálních vzdáleností.
128 - prumer = (prumer(1,:) + prumer(2,:))/2; %Matice průměrných vzdáleností.
129 - smerDatOdch = (smerDatOdch(1,:) + smerDatOdch(2,:))/2;
130 - vzdalenosti(3,:) = prumer;
131 - disp('Měření dokončeno.');
```

```

132 %-----
133 % Vytvoření aproximací vzdáleností.
134 - for l=1:l:kroky+1
135 -     if l~=1 && l~=2 && l~=kroky
136 -         aproxVzd(l) = (prumer(l-2)+prumer(l-1)+prumer(l))/3;
137 -     else
138 -         aproxVzd(l) = prumer(l);
139 -     end
140 - end
141 % Vzdálenosti se směrodatnými odchylkami
142 - vzSDO1 = prumer + smerDatOdch;
143 - vzSDO2 = prumer - smerDatOdch;
144 %-----
145 % Uložení získaných dat, pro vykreslení grafu.
146 - setappdata(0, 'rozestupy', rozestupy);
147 - setappdata(0, 'kroky', kroky);
148 - setappdata(0, 'prumer', prumer);
149 - setappdata(0, 'aproxVzd', aproxVzd);
```

Obrázek 4.4 – Výpis závěru kódu funkce *sberDat*

4.2 PROGRAMY PRO SÉRIOVOU KOMUNIKACI

V této podkapitole je popsán sériový způsob komunikace mezi modulem Arduino a počítačem jako alternativa ke způsobu popsanému.

Toto řešení obsahuje dva programy. První je vytvořen v prostředí Arduino IDE a je nahrán přímo na modul. Druhý je vytvořen v prostředí MATLAB a běží na osobním počítači. Cílem programů je získání různého počtu vzdáleností z různého počtu úhlů, přičemž parametry měření jsou zadány ze strany počítače.

Aby mohl modul získávat vzdálenosti ze senzoru, je třeba do programu přidat knihovnu *SharpIR.h* a pro nastavování pozice servomotoru je třeba přidat knihovnu *Servo.h*. Pro jejich použití je třeba obě knihovny vložit do adresáře knihoven v programu Arduino IDE a v inicializační části programu je připojit pomocí příkazu `#include <>`. Na obrázku 4.5. je výpis inicializační části programu v modulu a funkce pro čtení výstupu senzoru.

```
//Připojení knihoven senzoru a serva.
#include <SharpIR.h>
#include <Servo.h>
// Inicializace senzoru a serva.
SharpIR sensor(SharpIR::GP2Y0A21YK0F, A0);
Servo servo;
//Inicializace pomocných proměnných.
int pos = 0;
float Vzdz;
String data;
char parametr1;
char parametr2;
char konec;
int opak;
int krok;|
int j=0;
int ziskejVzd()
{
    Vzdz = sensor.getDistance();
    return Vzdz;
}
```

Obrázek 4.5 – Výpis inicializační části programu (Arduino)

Komunikace programů funguje následovně. Na obou stranách musí být předem nastavená stejná přenosová rychlost, v našem případě $9\,600\text{ bit}\cdot\text{s}^{-1}$ a název sériového portu. Zároveň je ještě třeba zvolit parametry měření, to je provedeno přepsáním písmen řetězce *parametry* v programu v prostředí MATLAB. V dalším kroku se obě strany ujistí, že je komunikace dostupná tím, že si navzájem pošlou písmeno „a“. Je důležité zmínit, že veškeré

zapisování a čtení přenášených dat probíhá na sériovém bufferu modulu a po přečtení jsou data automaticky z bufferu odstraňována. Chceme-li k přečteným datům opětovně přistupovat, musíme je při čtení uložit do proměnné. Na obrázku 4.6 je kód pro inicializaci komunikace v Arduino a na obrázku 4.7 je kód pro inicializaci komunikace v SW MATLAB (instructables.com, nedatováno).

```
void setup() {
  servo.attach(8);
  servo.write(0);
  //Nastavení sériové komunikace.
  Serial.begin(9600);
  Serial.println('a');
  char a = 'b';
  while (a != 'a') {
    a = Serial.read();
  }
  while (Serial.available() == 0)
  {}
}
```

Obrázek 4.7 – Výpis kódu inicializace sériové komunikace

```
80 % Funkce pro nastavení sériové komunikace.
81 □ function[linka,flag] = nastaveniLinky(comPort)
82   flag = 1;
83   % Inicializace seriového objektu "linka"
84   linka = serial(comPort);
85   set(linka, 'DataBits', 8);
86   set(linka, 'StopBits', 1);
87   set(linka, 'BaudRate', 9600);
88   set(linka, 'Parity', 'none');
89   fopen(linka);
90   % Navázání sériové komunikace.
91   a = 'b';
92   □ while (a~='a')
93     disp('Příprava linky...');
94     a=fread(linka,1,'uchar');
95   end
96   fprintf(linka, '%c', 'a');
97   % Informační okénko s finformací, že je seriová komunikace připravena
98   % Tlačítkem "Ok" je spuštěna komunikace
99   mbox = msgbox('Sériová komunikace nastavena.');
```

Obrázek 4.6 – Výpis kódu inicializace sériové komunikace

Proběhne-li krátká výměna znaku v pořádku, SW MATLAB zobrazí okénko se zprávou o úspěšném nastavení sériové komunikace a tlačítkem „OK“. Stisknutím tlačítka jsou z počítače odeslány parametry měření.

```

15 % Zvolení parametrů.
16 % První písmeno: A = 1 měření na úhel   Druhé písmeno: a = 3 kroky
17 %           B = 2 měření na úhel       b = 5 kroků
18 %           C = 5 měření na úhel       c = 10 kroků
19 parametry = 'Cb';
20
21 switch parametry(1)
22     case 'A'
23         pocet = 1;
24     case 'B'
25         pocet = 3;
26     case 'C'
27         pocet = 5;
28 end
29 switch parametry(2)
30     case 'a'
31         krok = 3;
32     case 'b'
33         krok = 5;
34     case 'c'
35         krok = 10;
36 end
37 % Odeslání parametrů a získání odpovědi pomocí funkce "komunikace"
38 vz = komunikace(linka,parametry);
39 % Pokud se vrátí parametr "E" nebo "e", dojde k upozornění, jsou
40 % jsou vypsány dostupné argumenty a sériová komunikace je ukončena.
41 if vz(1) == 'E'
42     disp('Špatný první vstupní argument.');
```

Obrázek 4.9 – Výpis části kódu pro zadání parametrů měření

```

if (Serial.available() > 0) {
    //Pokud je nastavena sériová komunikace, čeká program na zaslání parametrů měření.
    //Pokud parametry odpovídají dostupným možnostem, odpoví "R".
    //Pokud neodpovídají odpoví "E" nebo "e", podle chybného parametru.
    data = Serial.readString();
    parametr1 = data.charAt(0);
    parametr2 = data.charAt(1);
    if (parametr1 == 'A') {
        opak = 1;
    }
    else if (parametr1 == 'B') {
        opak = 3;
    }
    else if (parametr1 == 'C') {
        opak = 5;
    }
    else {
        Serial.println('E');
    }

    if (parametr2 == 'a') {
        krok = 3;
    }
    else if (parametr2 == 'b') {
        krok = 5;
    }
    else if (parametr2 == 'c') {
        krok = 10;
    }
    else {
        Serial.println('e');
    }
}

```

Obrázek 4.8 – Výpis části kódu pro přijetí zadaných parametrů

Pokud nastavení parametrů měření proběhlo v pořádku, jsou na straně modulu připraveny cykly pro měření a odesílání hodnot (obrázek 4.10). Veškeré nastavování a měření probíhá v části programu *setup()*. Na straně počítače jsou pak připraveny cykly pro čtení (obrázek 4.11). Pokud jeden z parametrů neodpovídá dostupným možnostem, dojde k ukončení komunikace a v příkazovém řádku prostředí MATLAB je vypsána hláška o špatném formátu zadaných parametrů a nápověda.

```

Serial.println('K');
int uhel = 180/(krok-1);

//Pokud jsou parametry vpořádku, začíná měření vzdáleností a nastavování serva.
for (int i = 0; i < krok; i++) {
  j=0;
  servo.write(uhel*i);
  for (j; j < opak; j++) {
    delay(200);
    ziskejVzd();
    Serial.println(Vzd);
  }
}
//Po dokončení měření čeká na odpověď s parametrem "X", po jeho získání odpoví parametrem "Y".
data = Serial.readString();
konec = data.charAt(0);
if (konec == 'X') {
  Serial.println('Y');
}
}

```

Obrázek 4.11 – Výpis části kódu pro cykly měření a ukončení

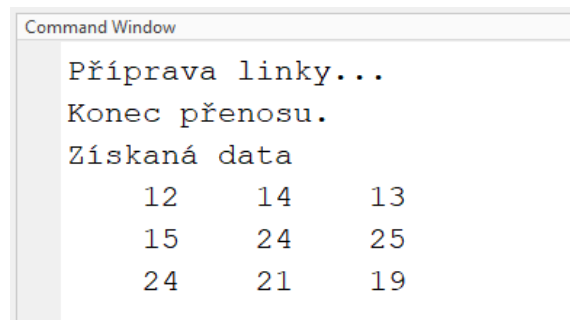
```

49 % Pokud jsou argumenty vpořádku, je zaslán argument "K"
50 % a nastavá proces získávání dat (vzdálenosti).
51 if vz(1) == 'K'
52     matice = zeros(krok,pocet);
53     for k=1:1:krok
54         for p=1:1:pocet
55             vz = fscanf(linka);
56             VZ = str2double(vz);
57             matice(k,p) = VZ;
58         end
59     end
60 end
61 % Po přenosu dat je zaslán argument "X", pokud je vrácen
62 % argument "Y", komunikace je ukončena a jsou vypsána
63 % získaná data (vzdálenosti) ve formě matice.
64 vz = komunikace(linka, 'X');
65 if vz(1) == 'Y'
66     disp('Konec přenosu. ');
67     disp('Získaná data');
68     disp(matice);
69 end
70 fclose(linka);

```

Obrázek 4.10 – Výpis části kódu s cykly pro čtení a ukončení

Po provedení cyklů SW MATLAB odešle písmeno „X“. Je-li odpověď „Y“, je vypsána hláška o dokončení přenosu společně s maticí obsahující získané vzdálenosti a komunikace je ukončena.



```
Command Window
Připrava linky...
Konec přenosu.
Získaná data
    12    14    13
    15    24    25
    24    21    19
```

Obrázek 4.12 – Výpis příkazového řádku v prostředí MATLAB

5 VÝSLEDKY EXPERIMENTŮ

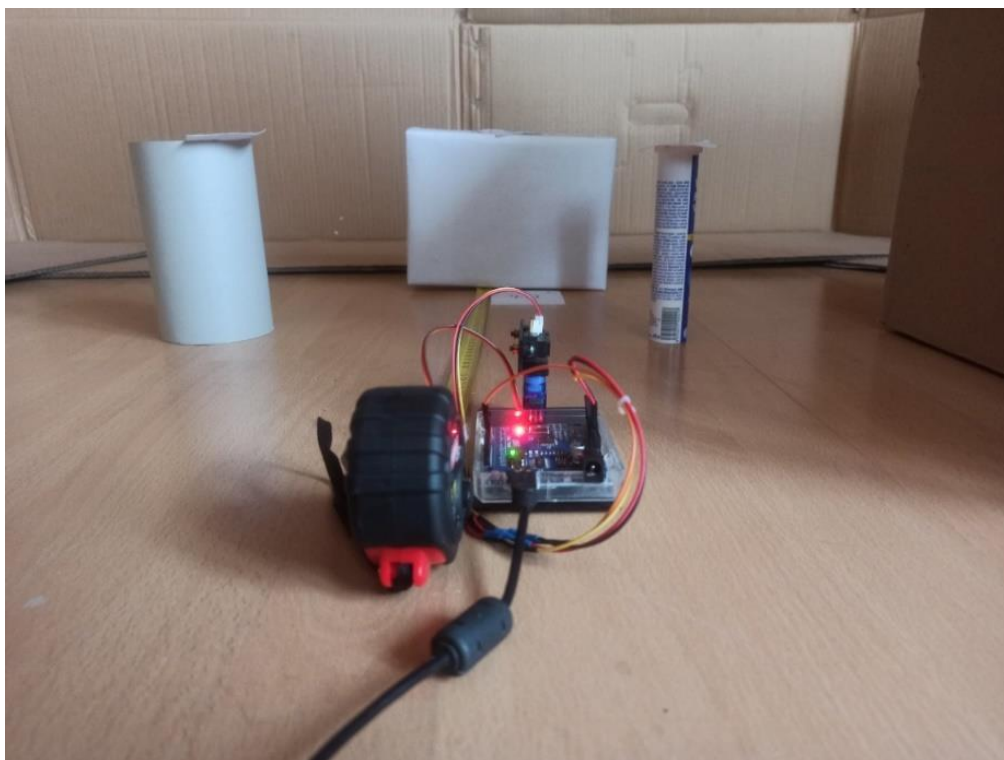
Tato kapitola se věnuje experimentálním měřením zhotoveného zařízení a obsahuje 3 bloky. První blok je věnován možnostem zobrazení rovinných hloubkových map a testování vlivů nastavitelných parametrů. Cílem bude určení optimálního nastavení parametrů. Druhý blok bude věnován testování senzoru pro různé povrchy. Poslední blok bude zaměřen na dosah senzoru.

5.1 MĚŘENÍ 1

V tomto bloku měření budou demonstrovány možnosti programu v softwaru MATLAB. Budou testovány možnosti zobrazení a dostupné možnosti nastavení parametrů měření. Měření



Obrázek 5.1 – Fotografie měřicí plochy 1 (pohled shora)



Obrázek 5.2 – Fotografie měřicí plochy 1 (pohled ze strany)

vzdáleností budou probíhat na ploše obsahující 5 objektů různých velikostí, povrchů, barev a vzdáleností vůči zařízení.

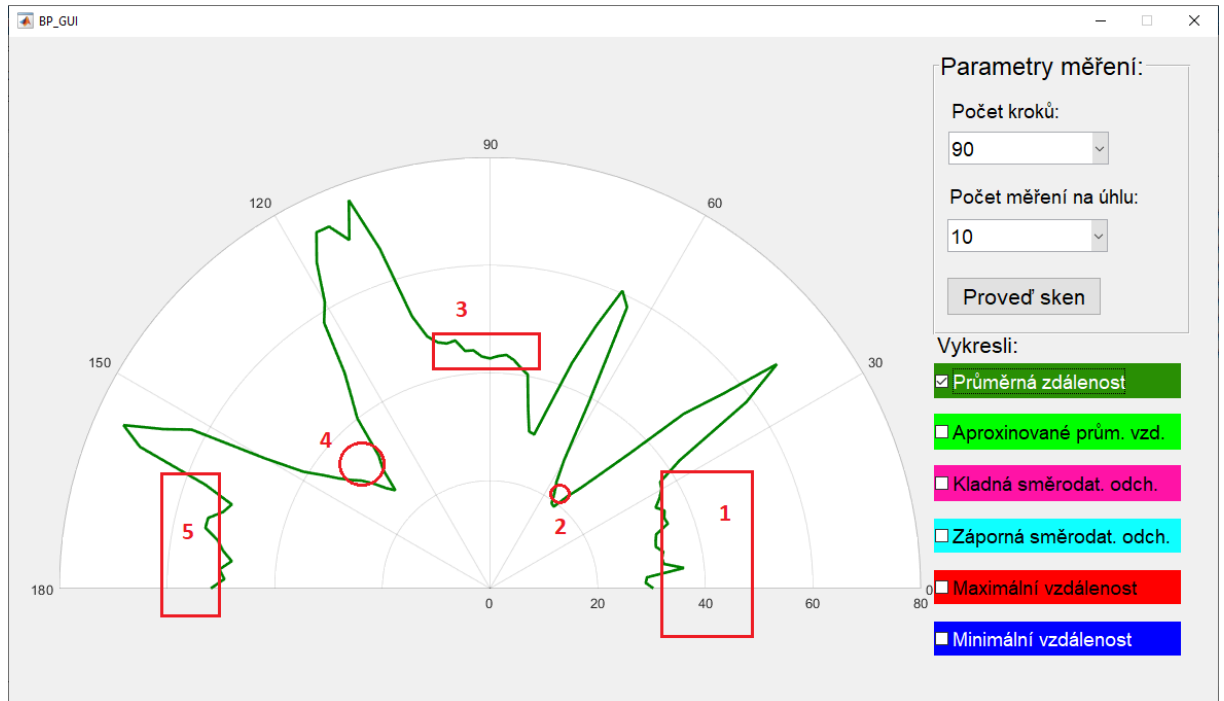
Na obrázcích 5.1 a 2 se nachází rozložení objektů na měřicí ploše. Všechny objekty jsou označeny číslem, kterým na ně bude odkazováno jak v textu, tak v grafu. Pro přehlednost jsou předměty uvedeny v tabulce 5.1.

Tabulka 5.1 – Předměty využité pro první blok měření

Číslo	Předmět	Vzdálenost, cm	Šířka, cm	Barva
1	Krabice	30	40	Hnědá
2	Tubus	20	2,8	Bílá/modrá
3	Krabice	40	20	Bílá
4	Sklenice	30	8,5	Šedá
5	Krabice	50	38	Hnědá

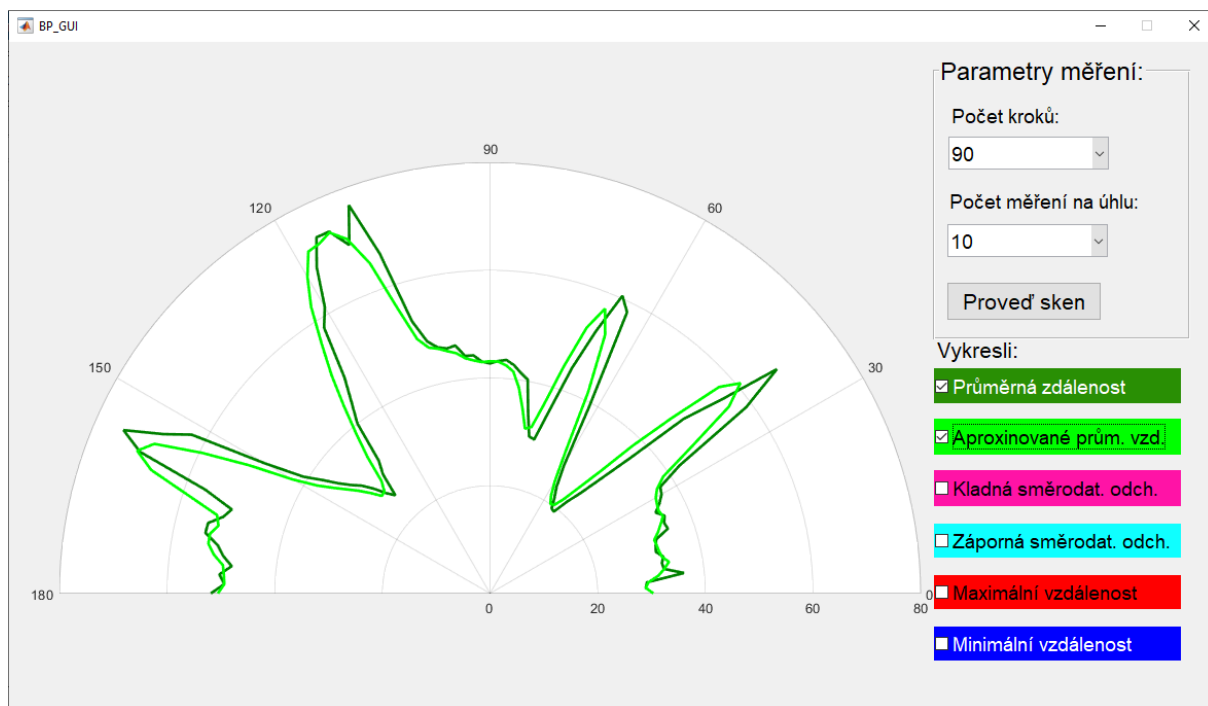
5.1.1 Možnosti zobrazení

První měření bylo změřeno s počtem kroků 90 a počtem měření 10 pro každý krok. Toto nastavení parametrů je optimální pro většinu měření, která nevyžadují měření objektů s šířkou pod 1 cm a pro ukázanou plochu je vyhovující. Na obrázku 5.3 jsou zobrazeny výsledky měření, konkrétně průměrné vzdálenosti s naznačenou polohou ležících předmětů.



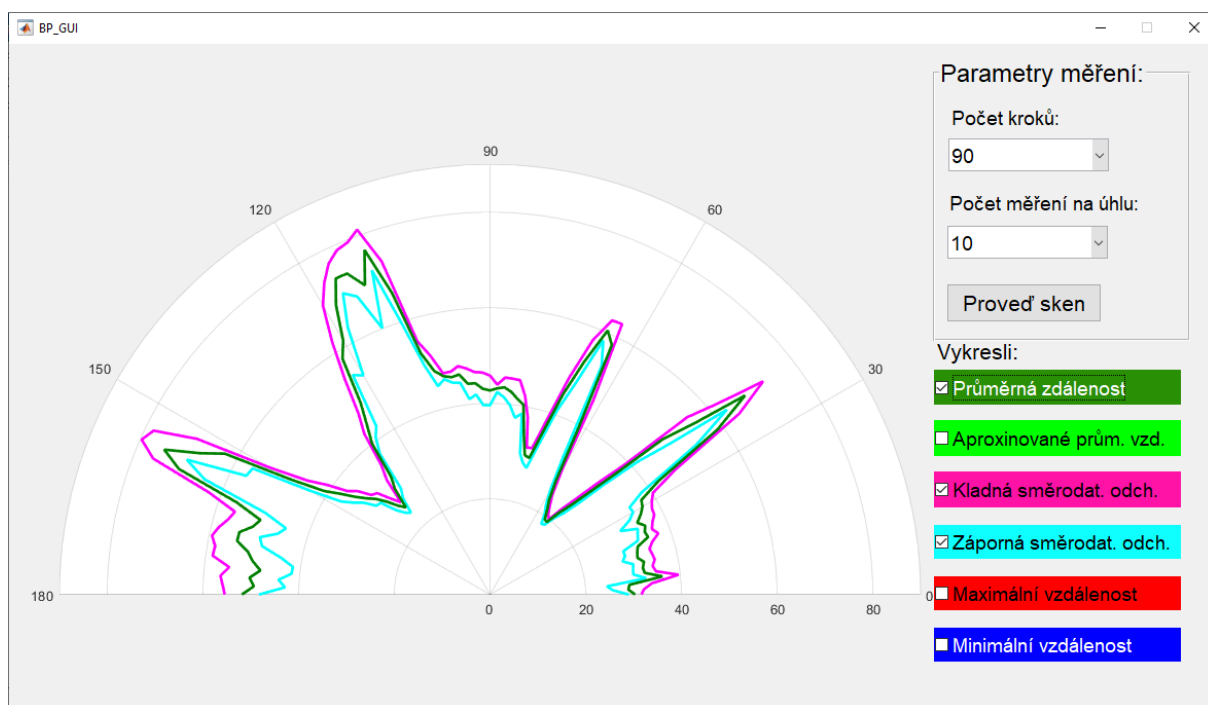
Obrázek 5.3 – Snímek naměřené plochy 1 – zobrazení průměrných vzdáleností

Na obrázku 5.3 lze rozpoznat přibližné rozpoložení předmětů. Hned si je možné všimnout velkého počtu nerovností, zejména u rovných povrchů, u kulatých se zase tvoří špice. Pro zlepšení zmiňovaných nerovností obsahuje program možnost zobrazení aproximací (obrázek 5.4).



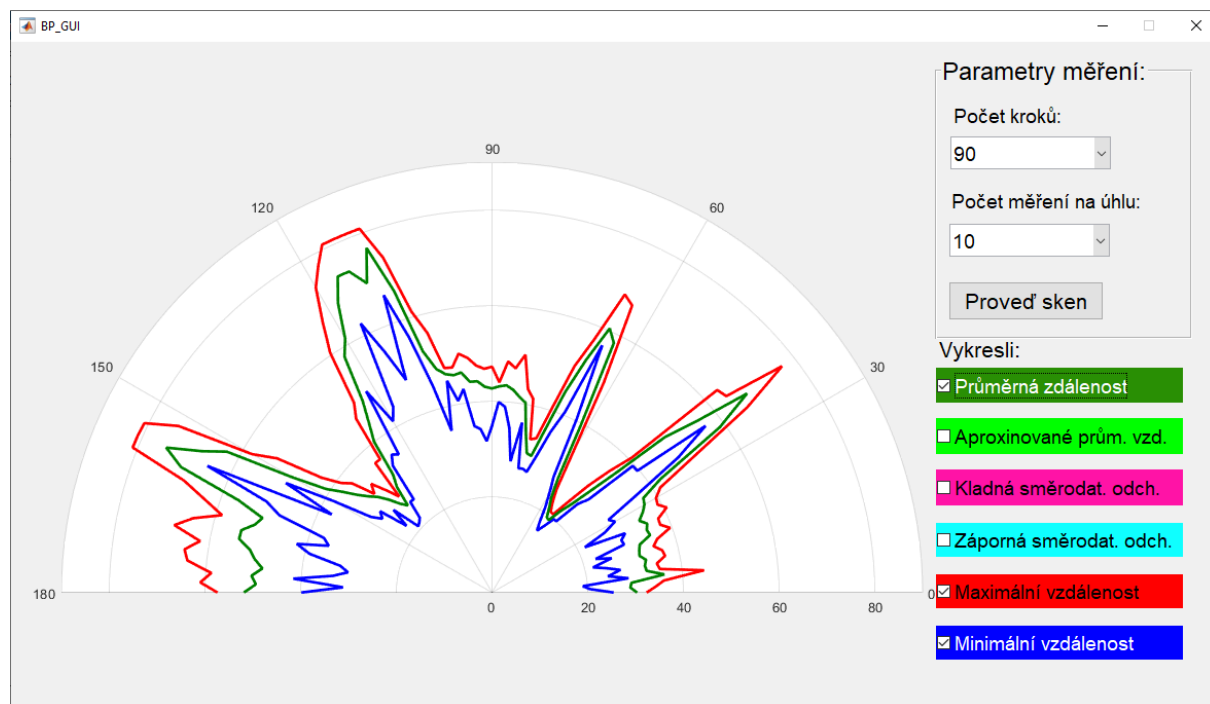
Obrázek 5.4 – Snímek naměřené plochy 1 – zobrazení aproximovaných vzdáleností

Aproximace pomohla lehce uhladit nerovnosti, ale stále nelze říci, že rovné povrchy jsou opravdu rovné. Dále následuje zobrazení s vyznačením směrodatných odchylek (obrázek 5.5).



Obrázek 5.5 – Snímek naměřené plochy 1 – zobrazení směrodatných odchylek

Směrodatné odchylky nám ukazují, kde dochází k největšímu rozdílu v naměřených hodnotách. K největšímu rozptylu dochází u rovných povrchů a v místech, kde by senzor již neměl mít dosah, tedy 80 cm a více.



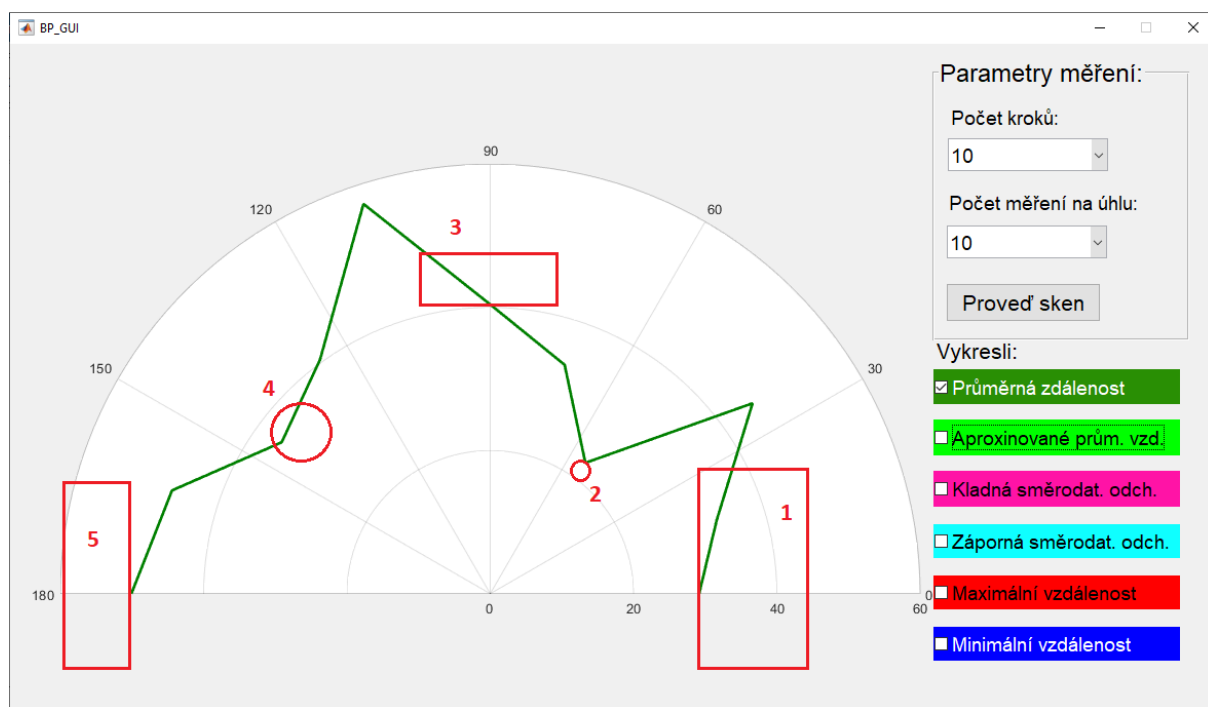
Obrázek 5.6 – Snímek naměřené plochy 1 – zobrazení maximálních a minimálních vzdáleností

Při zobrazení na obrázku 5.6, kde jsou znázorněna naměřená minima a maxima, se již ukazují hranice dosahu senzoru, které by se měly ukazovat i v případě zobrazení průměrných hodnot v místech, kde není žádná překážka. Poslední blok měření se bude této tématice věnovat a bude se v něm i zjišťovat maximální efektivní dosah senzoru.

Další důležitý poznatek je velká nerovnost na pravé straně plochy objektu 3, která je značně větší než zbytek nerovností. Jak lze vidět na obrázku 5.1, tak se jedná o rovný povrch a takto velká nerovnost by vzniknout neměla.

5.1.2 Měření s různými parametry

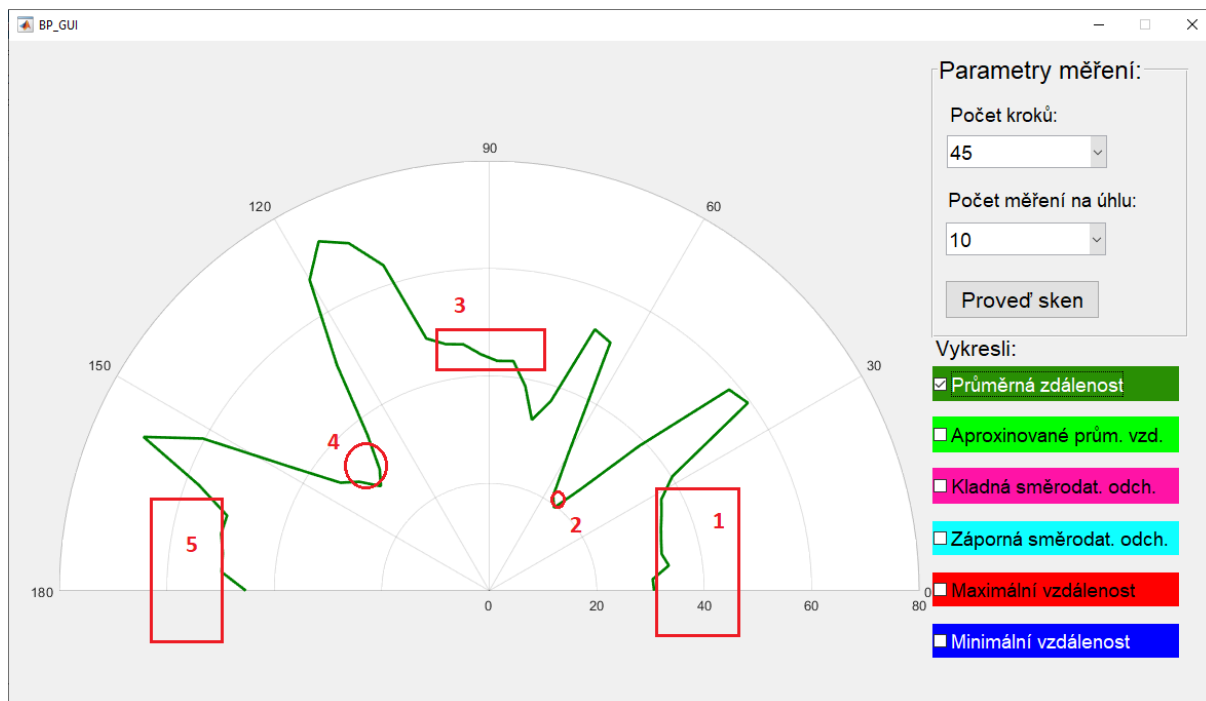
Následuje ukázka nastavení parametrů měření při stejném rozložení překážek, jako při ukázce zobrazení. Pro vykreslení hloubkové rovinné mapy bude využito pouze zobrazování průměrných vzdáleností. Nastavení parametrů půjde od nejnižších hodnot po ty nejvyšší. Jako první je nastavení s nejnižším možným počtem kroků 10. Počet měření vzdáleností na úhlu zůstane na hodnotě 10.



Obrázek 5.7 – Snímek naměřené plochy 1 – nastavení parametrů 10/10

Na první pohled je zřejmé, že díky malému počtu kroků obsahuje mapa menší počet nerovností, což ve výsledku působí větší přehledností. Při porovnání s obrázkem 5.1, lze i zde rozpoznat přibližné umístění všech překážek, ovšem při porovnání s tabulkou 5.1 lze vidět, že většina získaných vzdáleností neodpovídá skutečným vzdálenostem objektů. Objekty 1 a 5 jsou zhruba v pořádku, 2 a 4 jsou vychýleny a 4 je takřka nerozpoznatelný. Dále jsou téměř nerozpoznatelné některé kroky. Nejlépe je to vidět u 6. kroku, kdy prakticky splývá s kroky 5 a 7 a ztrácí se v přímce, ačkoliv se nachází ve správné vzdálenosti. Další potenciální problém by mohl nastat, pokud by se malý předmět, jako je na této ploše objekt 2, nacházel o 3 cm vedle. V tomto případě by ho zařízení vůbec nemuselo zachytit. Za zmínku jistě stojí doba měření, která trvala pouhých 11 vteřin.

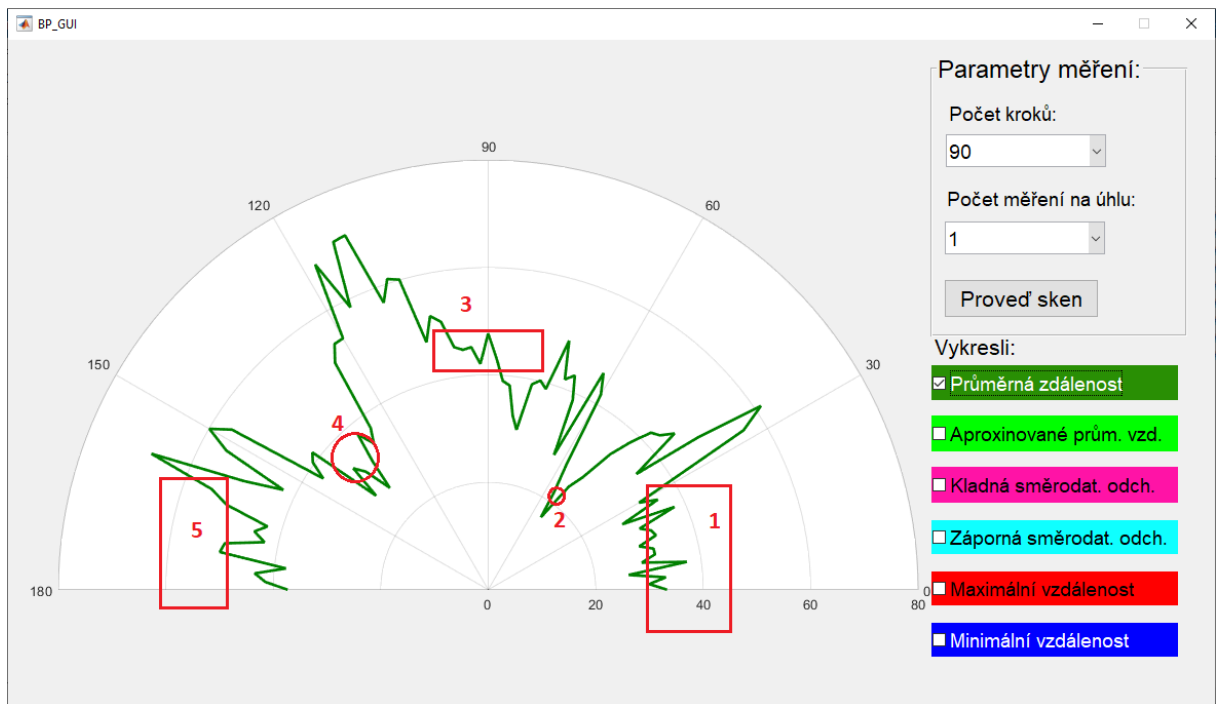
Nízký počet kroků zejména ukázal rychlé a jednoduché zobrazení hloubkové rovinné mapy za cenu zhoršení rozpoznání jednotlivých kroků a risk nezaznamenání drobných objektů. Pro další měření je navýšen počtu kroků na 45.



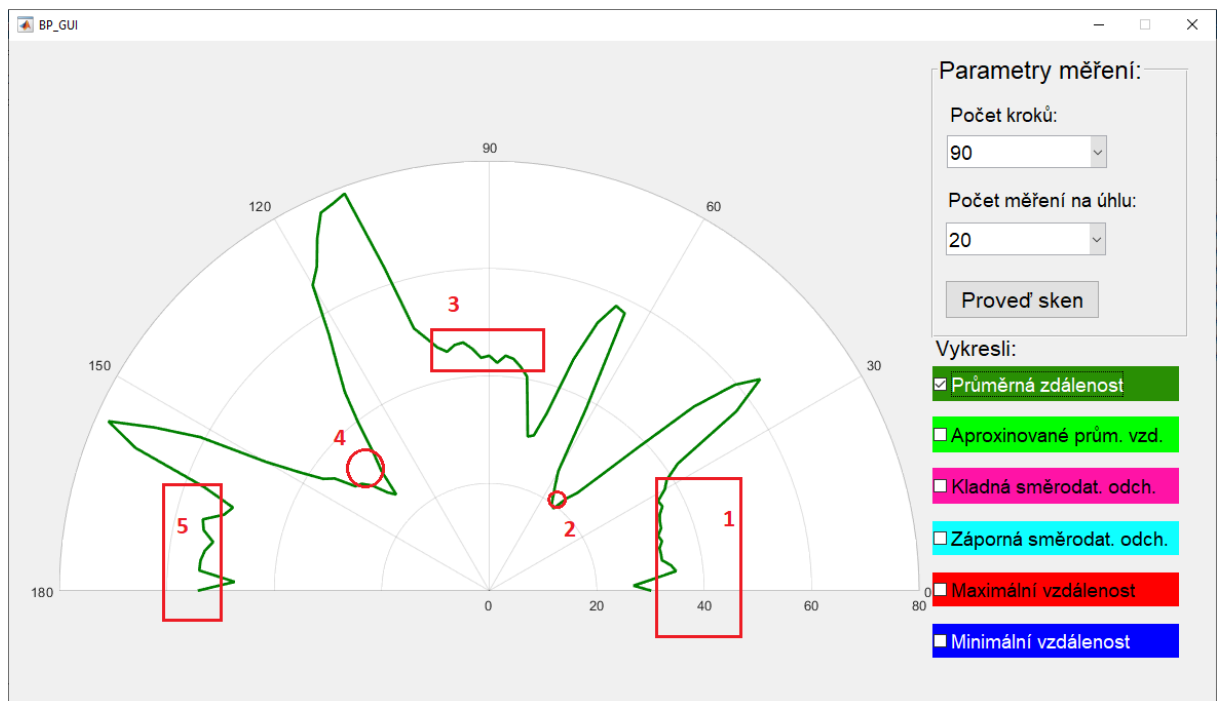
Obrázek 5.8 – Snímek naměřené plochy 1 – nastavení parametrů 45/10

Zvýšení počtu kroků rozhodně pomohlo k rozpoznání jednotlivých kroků a zvýšila se rozpoznatelnost některých objektů. Za cenu tohoto zlepšení však opět přibyly nerovnosti, zejména u rovných povrchů. Při zanedbání některých nerovností lze říci, že většina vzdáleností zhruba odpovídá až na překážku číslo 3, kde se opět ukazuje velká nerovnost na pravé straně. Tato nerovnost byla i u předchozího nastavení, ale nebyla tak rozpoznatelná jako zde. Více než trojnásobek kroků zapříčinil i odpovídající navýšení doby měření na 35 vteřin.

U předchozích dvou měření byl pouze zvyšován počet kroků, ale nyní bude nastavován i počet měření na jednom úhlu. Proběhnou 2 měření s nastavením počtu měření na úhel 1 a 20 a s počtem kroků 90. Při těchto měřeních lze provést porovnání s měřením na obrázku 5.3, kde byly předváděny možnosti zobrazení s parametry 90/10.



Obrázek 5.10 – Snímek naměřené plochy 1 – nastavení parametrů 90/1



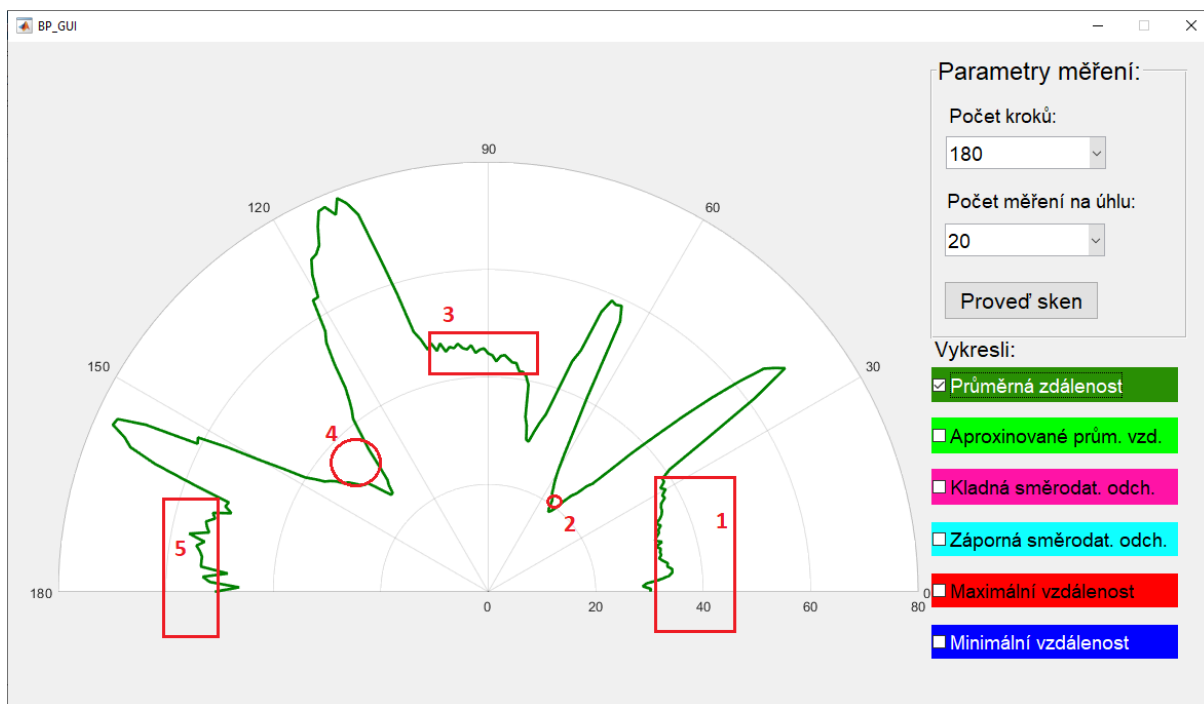
Obrázek 5.9 – Snímek naměřené plochy 1 – nastavení parametrů 90/20

Nejdříve k době trvání jednotlivých měření se stejným počtem kroků a různým počtem měření na úhlu. U měření pro ukázkou zobrazení na obrázku 5.3 byla doba měření 66 vteřin, měření na obrázku 5.9 14 vteřin na obrázku 5.10 124 vteřin. Z tohoto vyplývá, že počet měření na úhlu ovlivňuje dobu trvání měření obdobným způsobem, jako počet kroků.

Nastavení s nejnižším počtem měření na úhlu je pro většinu měření velmi špatná volba, a to hned z několika důvodů. S tímto nastavením funguje zobrazení pouze průměrných vzdáleností a jejich aproximací, rozdíly mezi vzdálenostmi na jednotlivých krocích jsou značné a tvoří tak velké nerovnosti, a především spoléhat se pouze na jedno měření může být značný risk. Zde je dobré připomenout, že veškeré měření a nastavování kroků probíhá pro oba směry, takže ve skutečnosti se nejedná o 1 sken, ale o 2. Toto samozřejmě platí pro všechna měření. Přesto je riskantní spoléhat se na průměr z pouhých dvou hodnot. Jediné reálné využití takto malého počtu měření je pro rychlou detekci blízkých okolních překážek.

Třetí měření se stejným počtem kroků, ale nejvyšším počtem měření na úhlu na obrázku 5.10 je velice podobné s počtem měření 10 na obrázku 5.3. Rozdíly jsou viditelné pouze u některých nerovností, ale nedá se jasně říct, které nastavení je na tom lépe. Je ovšem dobré zmínit, že s nastavením maximálního počtu měření na úhlu se v grafu místy ukazují i maximální hodnoty senzoru. Ty by se samozřejmě měly ukazovat u všech nastavení v místech, kde není žádná překážka. Toto nastavení je vhodné spíše ve chvíli, kdy je třeba provést opravdu důkladné měření, ale jinak by bylo vhodnější nastavení s nižším počtem měření na úhlu.

Na závěr tohoto bloku měření bude měření s maximálním možným nastavením parametrů (obrázek 5.11). Toto měření by teoreticky mělo nabízet nejvěrnější rovinou hloubkovou mapu měřicí plochy.



Obrázek 5.11 – Snímek naměřené plochy 1 – nastavení parametrů 90/20

Jako první je třeba k obrázku 5.11 říct, že měření trvalo dlouhých 242 vteřin, tedy přes 4 minuty. Při porovnání měření s naším úplně prvním měřením je doba měření čtyřnásobná, což odpovídá dvojnásobku za každý parametr měření. Samotný výsledek je při takto dlouhé době měření vcelku neuspokojivý. Drobné nerovnosti se povětšinou zmenšily, ale zároveň zdvojily. Již zmiňovaná velká nerovnost na pravé straně na objektu 3 i zde stále přetrvává. Nyní je však jasně rozeznatelná i na objektu 1. U objektu 5 je zase spousta středních nerovností. Lze tedy konstatovat, že nastavení maximálních parametrů neposkytne nejlepší výsledky a za cenu obrovské doby měření rozhodně nestojí.

5.2 MĚŘENÍ 2

Tento blok měření se bude věnovat především testování senzoru na různých materiálech a barvách. V první části proběhne měření na pěti objektech stejné velikosti a stejných vzdálenostech od zařízení, ale s různými povrchy. Jejich přehled je v tabulce 5.2. Provedena budou 2 měření, první s počtem kroků 45 a počtem měření na úhlu 10. Druhé s krokem 90 a počtem měření na úhlu 10. Všechny překážky budou umístěny v půlkruhu, zhruba 30 cm před zařízením.

Tabulka 5.2 – Předměty využité pro druhý blok měření

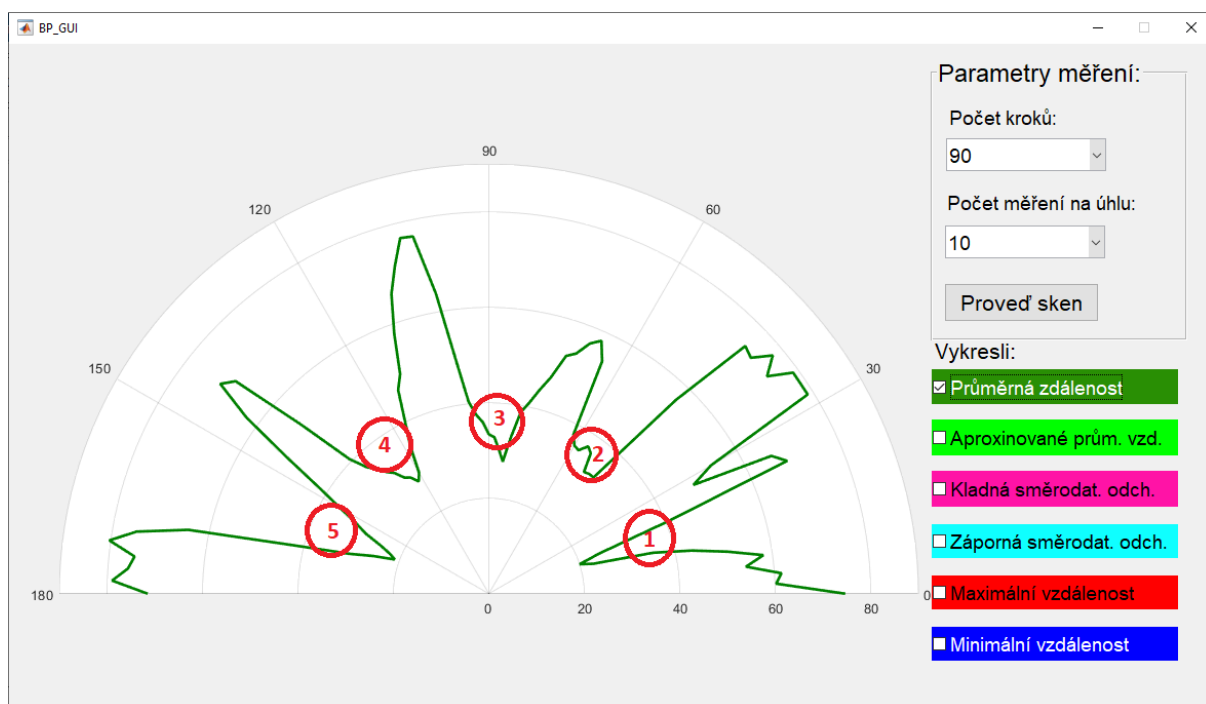
Číslo	Povrch	Barva
1	Sklo	Čirá
2	Papír	Černá
3	Papír	Šedá
4	Papír	Bílá
5	Alobal	Lesklá stříbrná



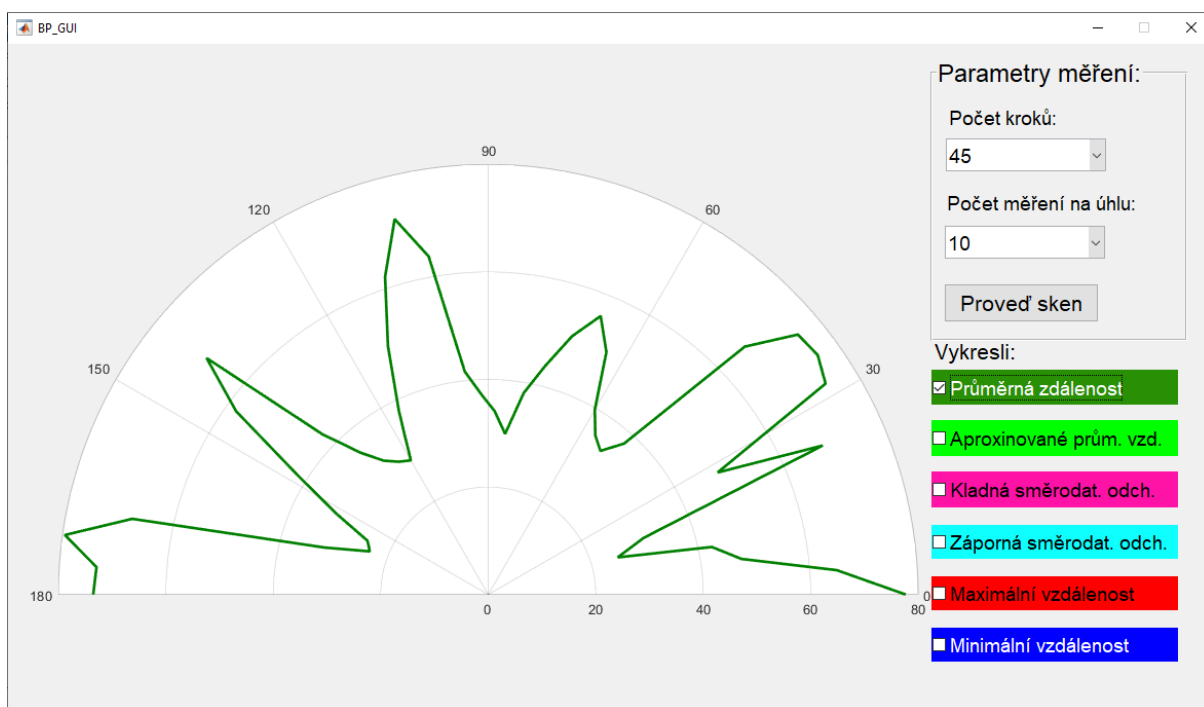
Obrázek 5.12 – Fotografie měřicí plochy 2.1 (pohled shora)



Obrázek 5.14 – Fotografie měřicí plochy 2.1 (pohled ze strany)



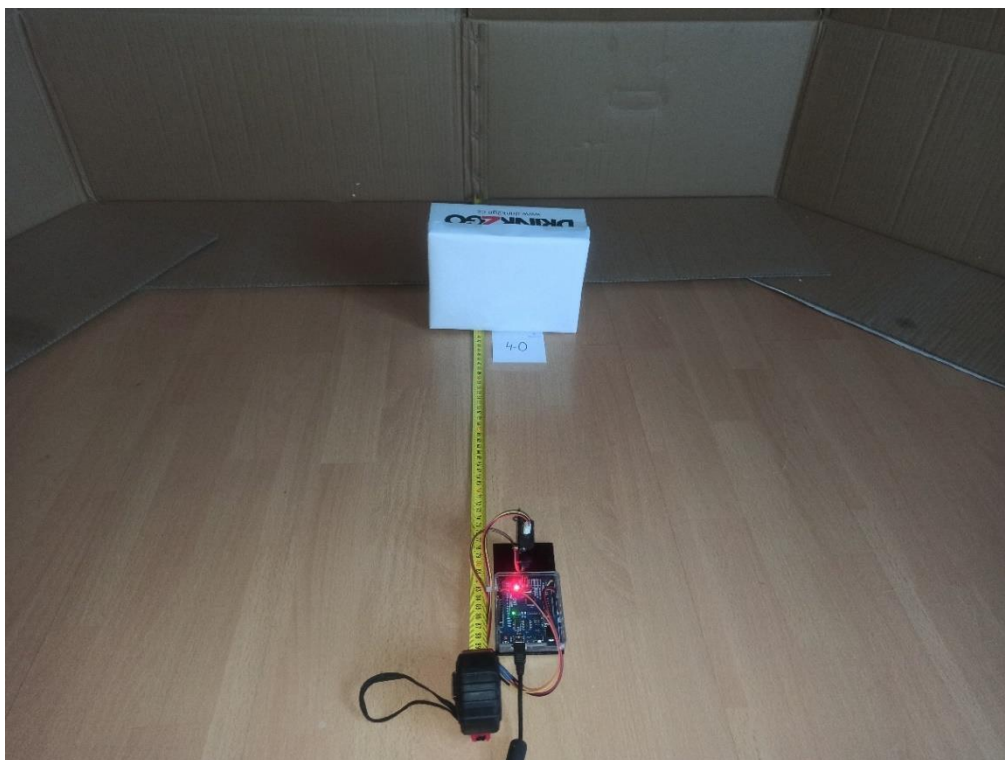
Obrázek 5.13 – Snímek naměřené plochy 2.1 – nastavení parametrů 90/10



Obrázek 5.15 – Snímek naměřené plochy 2.1 – nastavení parametrů 45/10

Na obrázku 5.12 je opět naznačené přibližné rozložení objektů. Přibližná pozice všech předmětů je vcelku rozpoznatelná, ale jejich vykreslená vzdálenost od senzoru se přesnou nazývat nedá. Nejhůře na tom jsou objekty 1 (sklo) a 5 (alobal), u kterých se vytvořila špice. Dále se v okolí jediné čiré sklenice vyskytují nerovnosti, což je nejspíše způsobeno průchodem některých paprsků skrz sklo. Obecně lepší výsledky zde nabízí měření s nižším počtem opakování na úhlu, konkrétně nejlepší výsledky jsou u objektů s černým a bílým povrchem. Proto budou tyto barvy využity pro další měření.

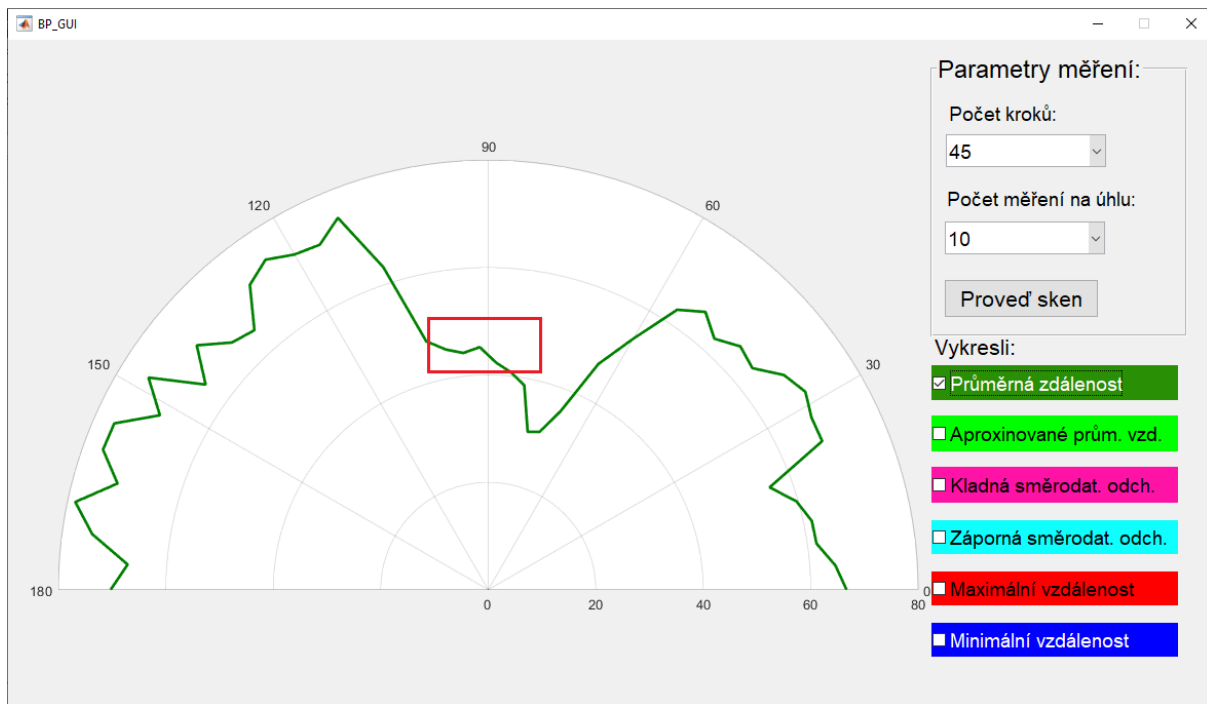
U dalších dvou měření budou použity objekty s rovným povrchem, stejnými rozměry, v identickém umístění na měřicí ploše, ale s rozdílnými barvami povrchu (černá a bílá). Barva s lepšími výsledky bude následně použita v závěrečném bloku měření, věnované zjištění maximální efektivní vzdálenosti senzoru.



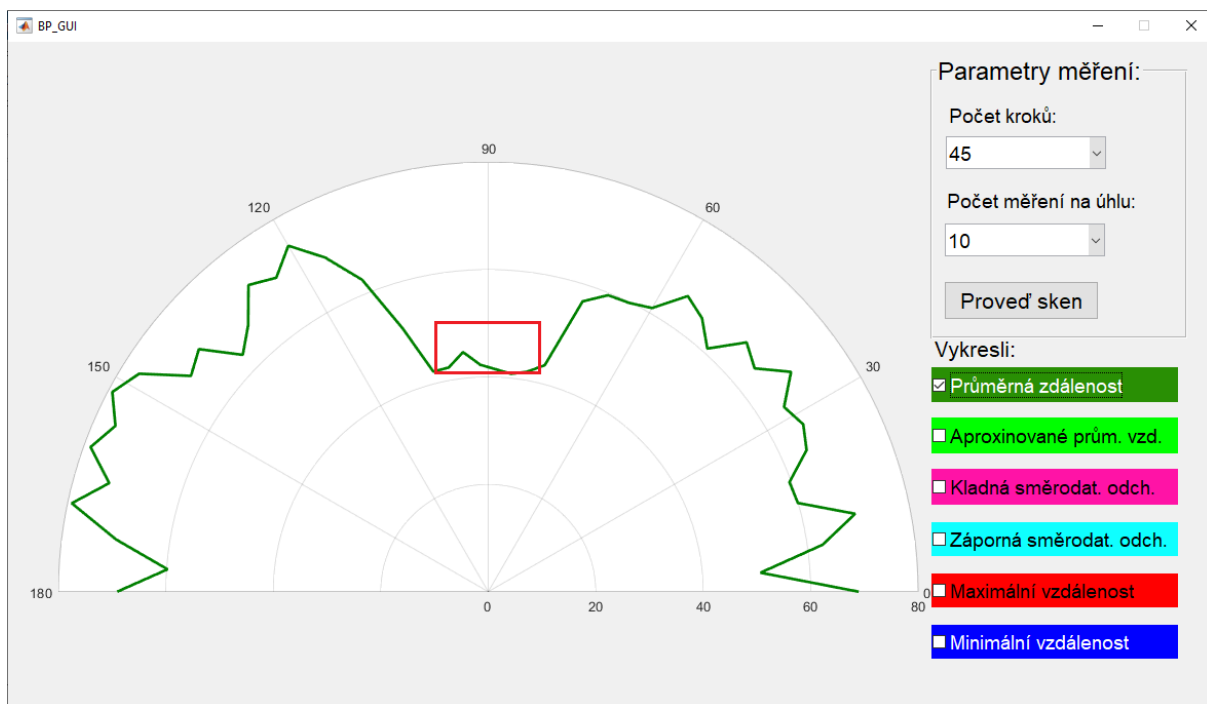
Obrázek 5.16 – Fotografie měřicí plochy 2.2



Obrázek 5.17 – Fotografie měřicí plochy 2.3



Obrázek 5.19 – Snímek naměřené plochy 2.2 – nastavení parametrů 45/10

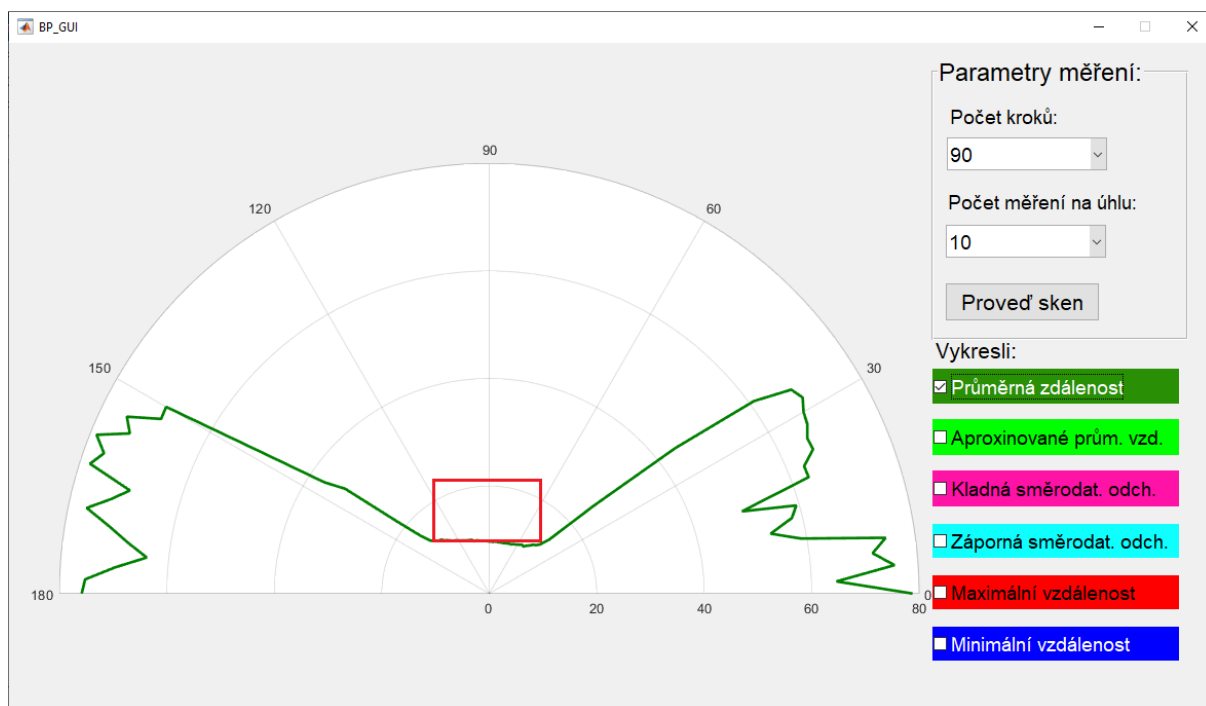


Obrázek 5.18 – Snímek naměřené plochy 2.3 – nastavení parametrů 45/10

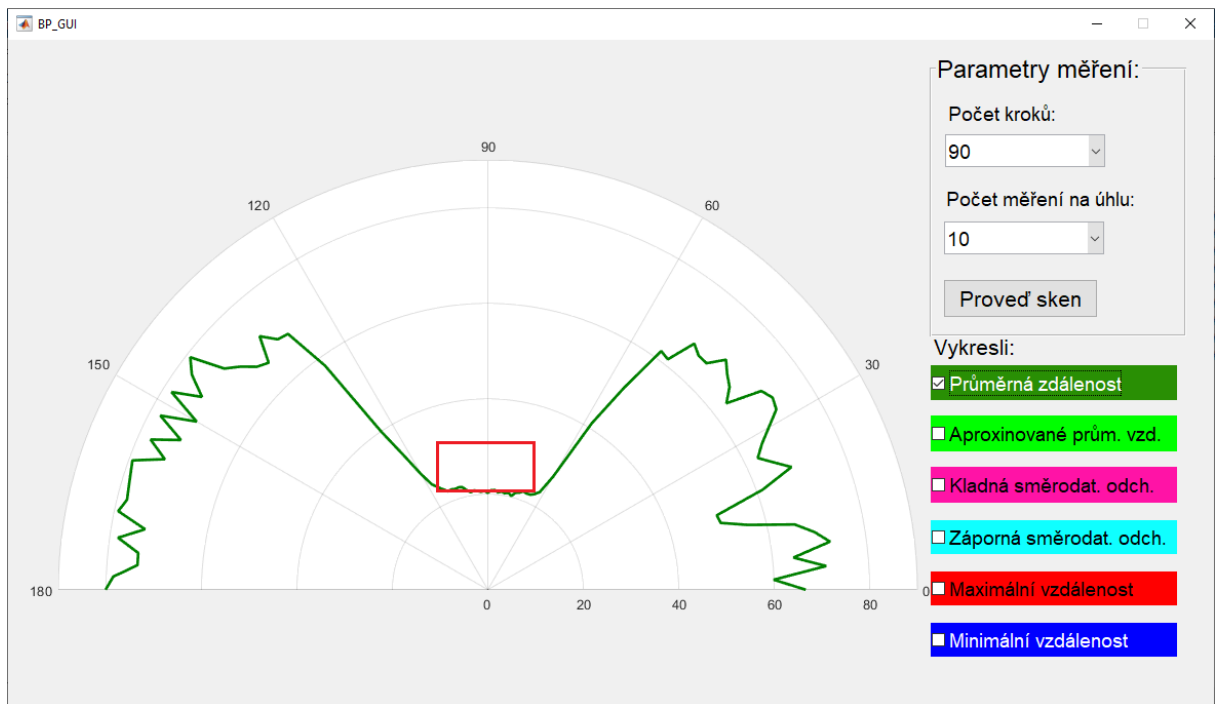
Při porovnání posledních dvou měření je vidět jasný rozdíl. U bílého povrchu se opět ukazuje velká nerovnost na pravé straně objektu, zmiňovaná v prvním bloku měření, která zcela ničí představu o umístění a vzhledu objektu. Druhá část objektu, kde se zmiňovaná nerovnost již nevyskytuje, je navíc posunuta o několik centimetrů dozadu. Druhé měření s černou plochou dopadlo o poznání lépe. Lze rozeznat začátek a konec plochy objektu a z větší části odpovídá vzdálenosti umístěného objektu. Sice plocha obsahuje znatelnou nerovnost směřující od zařízení, ale při porovnání s nerovností u předchozího měření se dá zanedbat. V dalším bloku bude tedy využit objekt na obrázku 5.17.

5.3 MĚŘENÍ 3

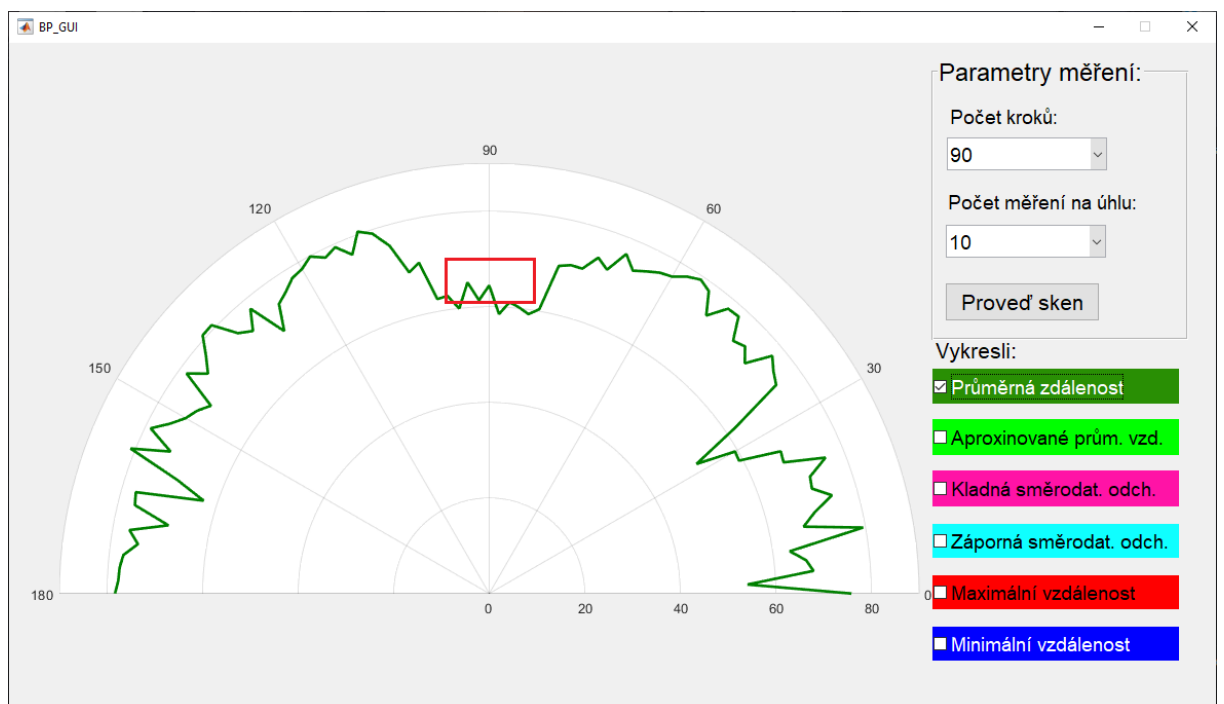
Poslední blok měření je věnován měření jednoho objektu v různých vzdálenostech od zařízení. Závěrem bude určení, pro jakou maximální vzdálenost je dobré zařízení využít. Pro toto měření bude využit objekt na obrázku 5.17, umístěn ve vzdálenostech 10, 20, 60 a 70 cm (plocha 3.1, 3.2, 3.3 a 3.4), s nastaveným počtem kroků 90 a počtem měření na úhlu 10. Výsledky jsou znázorněné na obrázcích 5.20 až 5.23.



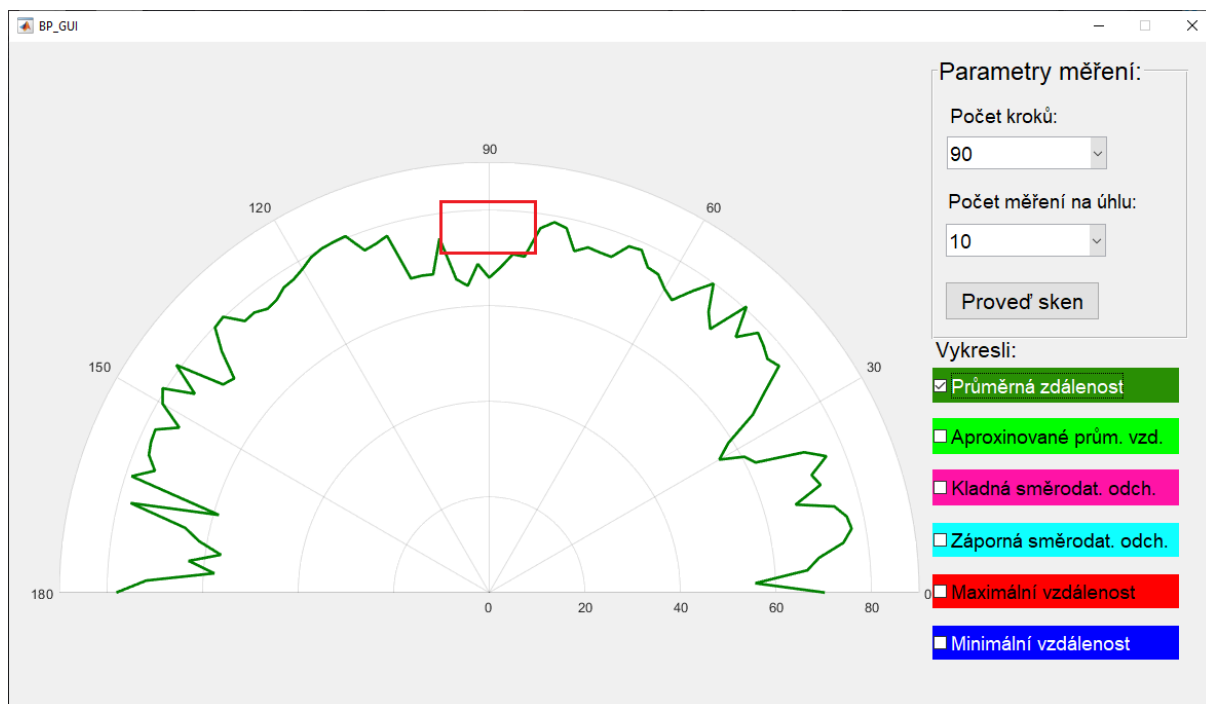
Obrázek 5.20 – Snímek naměřené plochy 3.1 – nastavení parametrů 90/10



Obrázek 5.21 – Snímek naměřené plochy 3.2 – nastavení parametrů 90/10



Obrázek 5.22 – Snímek naměřené plochy 3.3 – nastavení parametrů 90/10



Obrázek 5.23 – Snímek naměřené plochy 3.4 – nastavení parametrů 90/10

První 2 měření, pro vzdálenosti 10 a 20 cm na obrázcích 5.20 a 5.21, lze skvěle vidět umístění překážky a její plocha obsahuje pouze drobné nerovnosti. Lze tedy říci, že měření tohoto druhu objektů je pro zařízení naprosto vyhovující.

Dále lze použít měření z předchozího bloku na obrázku. 5.19, kde je objekt umístěn ve vzdálenosti 40 cm od zařízení. V této vzdálenosti již rovný povrch obsahuje větší nerovnosti, ale stále je dobře rozeznatelný. Zatím lze tedy říci, že senzor měří v první polovině svého dosahu dobře.

Měření na obrázku 5.22 ve vzdálenosti 60 cm od zařízení ukazuje značné zhoršení oproti předchozím vzdálenostem. Objekt je stále rozeznatelný, ale množství nerovností již rozhodně nepůsobí dojmem, že zachycená překážka má rovný povrch. Dále je třeba již zmínit nerovnosti vzniklé v místech, kde není přítomna žádná překážka. V předchozích měřeních by se tyto nerovnosti daly ignorovat, ovšem nyní kdy se zkoumaný předmět nachází na úrovni těchto nerovností, může dojít k jejich záměně, nebo dokonce k dojmu, že se na ploše nachází další překážky.

Poslední měření na obrázku 5.23, se vzdáleností objektu 70 cm od zařízení je již prakticky nepoužitelné. Rovná plocha objektu zde obsahuje tak značné množství velkých nerovností, že ho jde jen stěží odlišit od nerovností vzniklých v prázdném prostoru.

Závěrem tedy je, že zařízení je nejefektivnější pro měření překážek ve vzdálenostech 10 až 40 cm od senzoru a spolehlivé odhalení překážky je maximálně ve vzdálenosti 60 cm od zařízení. Pro překážky ve větší vzdálenosti již hrozí, že splynou s náhodnými nerovnostmi v okolí.

6 ZÁVĚR

Cílem této bakalářské práce bylo vytvoření zařízení skládající se z infračerveného senzoru vzdálenosti umístěného na hřídeli servomotoru. Obě komponenty byly dále spojeny s modulem Arduino UNO, které bylo dále spojeno skrze USB k osobnímu počítači.

Bylo předvedeno kompletní programové řešení, vytvořené v prostředí MATLAB a za pomoci podpůrného balíčku, který obsahuje nástroje pro přímou komunikaci s prvky modulu. Tento program obsahuje veškeré potřebné nástroje pro nastavování parametrů měření a pro vykreslování získaných dat. Dále byla naznačena alternativní možnost programového řešení, a to za pomoci sériové komunikace. Pro tento způsob byl vytvořen jednoduchý komunikační protokol, při kterém dochází k výměně dat mezi počítačem a modulem Arduino.

V experimentální části byla provedena série měření zahrnující ukázkou veškerých nastavitelných parametrů měření a vykreslení získaných výsledků. Zároveň byla měřena doba trvání měření s jednotlivými parametry. Dále byla zkoumána přesnost senzoru pro různé povrchy a barvy. Nakonec bylo zjišťováno, pro které vzdálenosti je senzor nejefektivnější a pro jaké maximální vzdálenosti je senzor vhodné použít. Závěrem je, že senzor zvládá měření i v případě povrchu s nízkou odrazivostí záření. Jeho efektivní použití pro měření vzdáleností je ovšem do dvou třetin maximálního dosahu uváděného výrobcem.

Vytvořené zařízení je plně funkční a je možné jej využít pro potřeby měření vzdáleností k překážkám v oblasti 180° před zařízením. Po úpravě je možné zařízení využít i v dalších aplikacích, např. jako orientační smysl robota, který by na základě získaných vzdáleností sám upravoval směr svého pohybu. Nabízí se také možnost rozšíření samotného zařízení, kdy je teoreticky možné přidat další servomotor, který by hýbal s původním servomotorem ve směru nahoru a dolů. To by umožnilo získání vzdáleností v třírozměrném prostoru a bylo by pak možné sestavit 3D hloubkovou mapu.

POUŽITÁ LITERATURA

- Arduino and Matlab let them talk using serial communication*. 2021. [online]. Instructables Circuits. [cit. 2021-11-20]. Dostupné na: <https://www.instructables.com/Arduino-and-Matlab-let-them-talk-using-serial-comm/>.
- GP2Y0A21YK0F Datasheet (PDF) - Sharp Electronic Components*. 2008. [online]. Alldatasheet. [cit. 2021-11-20]. Dostupné na: <https://www.alldatasheet.com/datasheet-pdf/pdf/412635/SHARP/GP2Y0A21YK0F.html>.
- CHLEBNÝ, J., 2009. *Automatizace a automatizační technika 4.*, aktualiz. vyd., Brno: Computer Press. 296 s. ISBN 978-80-251-2523-6.
- Infračervený senzor vzdálenosti 10-80 cm GP2Y0A21YK0F*. 2021. [online]. Drátek. [cit. 2021-11-20]. Dostupné na: https://dratek.cz/arduino/2095-infracervený-senzor-vzdálenosti-10-80-cm-gp2y0a21yk0f.html?gclid=Cj0KCQiAtqL-BRC0ARIsAF4K3WHd3ii50PpPdOkxiJ9Qe9NzUX8xOtMNoq7miKCRKv7ytfrSZs-VbWYaAsilEALw_wcB.
- IR Distance Sensor Tutorial (SHARP GP2Y0A21YK0F)*. 2021. [online]. Makerguides. [cit. 2021-11-20]. Dostupné na: <https://www.makerguides.com/sharp-gp2y0a21yk0f-ir-distance-sensor-arduino-tutorial/>.
- Klon Arduino UNO R3 ATmega328P CH340 mini USB*. 2021. [online]. Drátek. [cit. 2021-11-20]. Dostupné na: <https://dratek.cz/arduino/1353-klon-arduino-uno-r3-atmega328p-ch340-mini-usb.html>.
- KOŘÁN, P. 2014. *VYUŽITÍ LASERU V PRŮMYSLU – MINULOST A SOUČASNOST*. [online]. LAO – lasery a optika. [cit. 2021-11-20]. Dostupné na: <http://www.lao.cz/lao-info-49/vyuziti-laseru-v-prumyslu--minulost-a-soucasnost-313>.
- LED or Light Emitting Diode – Pin Diagram, Construction, Working Principle*. 2020. [online]. ElectroDuino. [cit. 2021-11-20]. Dostupné na: <https://www.electroduino.com/led-light-emitting-diode/>.
- MATLAB & SIMULINK úvod*. 1991-2021. [online]. Humusoft. [cit. 2021-11-20]. Dostupné na: <https://www.humusoft.cz/matlab/>.
- NOVÁK, P. 2005. *Mobilní roboty: pohony, senzory, řízení*. Praha BEN – technická literatura. 243 ISBN 80-7300-141-1.
- PETERKA, J. 1996. *Synchronní, asynchronní a arytmičtý přenos*. [online]. eArchiv. [cit. 2021-11-20]. Dostupné na: <https://www.earchiv.cz/a96/a650k150.php3>.
- Servo SG90 9g micro motor*. 2021. [online]. Drátek. [cit. 2021-11-20]. Dostupné na: <https://dratek.cz/arduino/897-eses-servo-motor-9g.html>.
- SICILIANO, B., KHATIB, O., (Eds.). 2008. *Springer Handbook of Robotics*. Berlin: Springer-Verlag. 1611 s. ISBN 978-3-540-23957-4.
- Světlo: Elektromagnetické vlnění, spektrum a fotony*. 2021 [online]. Khan Academy. [cit. 2021-11-20]. Dostupné na: <https://cs.khanacademy.org/science/obecna-chemie/xfed2aace53b0e2de:atomy-a-jejich-vlastnosti/xfed2aace53b0e2de:fotoelektronova-spektroskopie/a/light-and-the-electromagnetic-spectrum>.

VOBECKÝ, J. & ZÁHLAVA, V., 2005. *Elektronika: součástky a obvody, principy a příklady* Třetí, rozšířené vydání., Praha: Grada Publishing. 220 s. ISBN 978-80-247-1241-3.

PŘÍLOHY

A – CD

Příloha k diplomové bakalářské práci

Zařízení pro testování vlastností IR senzoru vzdálenosti

Ladislav Poledník

CD

Obsah

- 1 Text bakalářské práce ve formátu PDF.
- 2 Úplný zdrojový kód aplikace hlavního řešení
- 3 Úplný zdrojový kód aplikace pro sériovou komunikaci