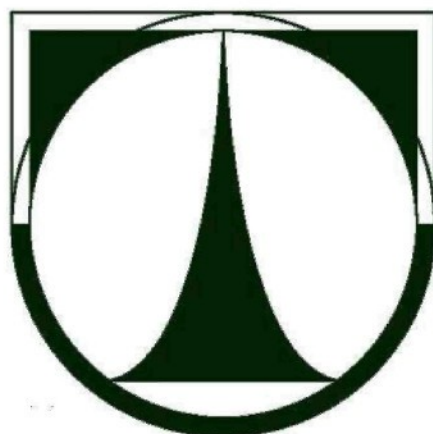


Technická Univerzita v Liberci
Fakulta mechatroniky a mezioborových inženýrství

ÚSTAV INFORMAČNÍCH TECHNOLOGIÍ A ELEKTRONIKY



DIPLOMOVÁ PRÁCE

Elektronické seřizování puškohledů
(Electronic tuning riflescopes)

Autor: Bc. Jiří Laňka

Vedoucí práce: Ing. Jaroslav Buchta

Prohlašuji, že jsem tuto diplomovou práci vypracoval sám, pouze s použitím literatury uvedené v seznamu na konci práce

V Liberci, dne: 21.5.2010

.....
Jiří Laňka

Poděkování

Na tomto místě bych rád co nejupřímněji poděkoval panu Ing. Jaroslavu Buchtovi za odborné vedení této diplomové práce. Dále bych rád poděkoval panu Oldřichu Bělaškovi za dobré rady a náměty pro správný přístup k nastřelování a panu Miloši Slanému za poskytnutí kvalitního puškohledu na testování.

Jiří Laňka

Obsah

Úvod.....	9
1. Motivace	10
2. Hardware	11
2.1 Mikrokontrolér PIC	11
2.1.1 Typy mikrokontrolérů PIC	12
2.1.2 Instrukční soubor	12
2.1.3 Komponenty mikrokontrolérů	12
2.1.4 Programátory	13
2.1.5 Program mikrokontroléru	14
2.2 Krokové motorky	14
2.2.1 Řízení krokových motorků	15
2.2.2 Čtyřtaktní ovládání po jedné fázi	16
2.2.3 Čtyřtaktní buzení po dvou fázích	16
2.2.4 Osmitaktní buzení krokového motorku	16
2.3 Budič krokového motorku	17
2.3.1 Základní vlastnosti	17
2.3.2 Popis činnosti	18
2.3.3 Připojení KM k budiči	19
2.3.4 Rozložení konektorů a součástek	19
2.4 Sériová linka RS-232C	20
2.4.1 Princip sériové komunikace	21
2.4.2 Asynchronní přenos dat	21
2.4.3 Technický popis	22
2.5 Propojení RS232 a krokových motorků	23
2.6 Snímací buňka	26
3. Software.....	29
3.1 Ovládací Aplikace	29
3.1.1 Vzhled aplikace	29
3.1.2 Menu aplikace	30
3.1.3 Funkce aplikace	36
3.1.4 Pořízení snímků nástřelných	37

3.1.5 Vyhodnocení snímků nástřelných	38
3.1.6 Posunutí kříže	39
3.1.7 Kalibrace	39
3.1.8 Paměť aplikace	40
3.2 Komunikační protokol	40
3.3 Algoritmus „Dohánění středu“	41
4. Testování	43
4.1 Testování za pomoci nezkušených uživatelů	43
4.2 Testování za pomoci zkušených uživatelů	44
4.3 Testování za pomoci sestaveného systému	46
4.4 Výsledek testování	46
4.5 Návratnost	48
5. Závěr	49
Seznam použitých zdrojů	50
Seznam obrázků	52
Seznam tabulek	53
Seznam grafů	53
Seznam použitých zkratk	54
Seznam příloh	55

RESUMÉ

Česky

Tato práce seznamuje s problematikou elektronického seřizování puškohledů, která je rozdělena do pěti samostatných bloků. Náplní první části je kratičký úvod o problematice a proč je tato práce realizována. V druhé části je proniknutí do problematiky z hlediska seznámení s teorií jednotlivých komponent a součástí, které jsou použity při navrhování a realizaci funkčního systému, který bude ovládat samostatnou činnost seřizování. V této části je popsán i způsob řešení komunikace mezi jednotlivými hardwarovými celky. Samozřejmostí, která úzce souvisí s hardware je doplnění o řídicí software, pro který je vyhrazena třetí kapitola. Ve čtvrté části jsou popsány dosažené výsledky z průběhu testování. Závěrečná kapitola je poslední částí, kde je popsáno několik průběžných poznatků z celé práce a zda-li se naplnily plánované cíle.

Klíčová slova: Nastřelování, seřizování, puškohled, mikrokontrolér, motor

English

This thesis introduces the problems of electrical tuning scopes, which are divided into five individual blocks. The contents of the first part is short introduction about the problems and why is this job realized. In the second part is the insight into the problem in terms of acquaintance all of the components, which are used in designing and implementing a functional system, which will control the adjustment of individual activity. In this section there is described a way of communication of individual hardware equipment. Obviousness, which is closely related to hardware, is the addition of management software, which is reserved for the third chapter. Fourth section describes the results of the testing. Final section is the last part, where are described some intermediate knowledges of the whole work and if all the targets were realized

Keywords: Driving, adjusting, riflescopes, microcontroller, motor

Vlastní text práce

Vlastní text je rozdělen do pěti kapitol a několika podkapitol.

V práci se vyskytuje několik průvodních obrázků, tabulek a grafů, které jsou systematicky označeny vždy jako (obr.x.y.z), (tab.x.y.z) a (graf.x.y.z), kde „x“ vyjadřuje pořadové číslo kapitoly, „y“ je pořadové číslo podkapitoly a pod označením „z“ se nachází pořadové číslo obrázku, tabulky nebo grafu.

Při uvádění zdrojů ve vlastním textu se odkazují na seznam použité literatury umístěné v závěru práce, a to pořadovými čísly: pro knižní výtisky ve tvaru [i] a pro zdroje z internetu v tom samém tvaru, ale s předponou www [www i].

Kurzívní písmo je vyhrazeno pro doslovné citace.

Úvod

V tomto kratičkém úvodu bude nastylizován průběh celé diplomové práce, od počínajícího navrhování samostatného hardware, přes navrhování řídicího software, až po konečné kompletování a testování celého funkčního systému.

V první části celé diplomové práce, dále jen DP, jsou nastíněny problematiky tohoto oboru, které se především týkají zbraní, střeliva a optik nebo-li puškohledů. Samozřejmě v této části práce jsou i popsány důvody a motivace proč je tato práce realizována.

Druhá část DP se zabývá vysvětlením některých pojmů z oblasti elektroniky a to jednotlivých hardwarových komponent jako jsou například krokové motorky, snímací zařízení terče a jiné další komponenty důležité pro celkovou funkci zařízení.

Třetí kapitola se zabývá softwarovou částí práce, která obsahuje dva vlastní programy. Nejrozsáhlejším programem je řídicí aplikace psaná v jazyce C#, která má za úkol obsluhovat celé zařízení a to například vyhledávání jednotlivých nástřelů, dopočítávání některých důležitých hodnot a posílání hodnot do zařízení, které ovládá jednotlivé motorky. Druhým a méně rozsáhlým programem je program v mikrokontroleru PIC, který má na starosti budič krokových motorků.

Ve čtvrté kapitole je popsáno závěrečné testování a jsou zde i uvedeny jednotlivé výsledky celé diplomové práce včetně návratnosti zařízení.

A v úplně poslední fázi, a to v páté kapitole, je závěrečné shrnutí, zda-li se docílilo stanovených cílů práce a další možnosti o rozšíření.

1. Motivace

Pokud se ohlédneme kolem sebe, uvidíme časté využívání elektronických zařízení, a to dnes a denně. I když se zaměříme na úplně nejobyčejnější věci pro člověka jako je například chůze, zjistíme, že spousta z nás využívá mp3 přehrávače, mobilní telefony, různé krokoměry a mnohé jiné, až po využití elektroniky v nejrůznějších průmyslových odvětvích nebo jiných astronomicky velkých vědách, jako je kupříkladu matematika, lékařství, astronomie a takto by se zde mohl uvést mnohem větší výčet odvětví. Ve zkratce řečeno, dnes bychom se téměř nemohli bez „elektrické energie“ a spotřebičů obejít, natož pomalu vůbec existovat, jelikož nám to velmi usnadňuje a zpříjemňuje život.

I tato práce bude využívat elektrické energie a co je důležitější, je zde vyvíjen spotřebič, respektive elektronické zařízení, které bohužel schází na dnešním trhu pro obyčejné koncové uživatele sportovních střelných zbraní. Samozřejmě se jedná, jak z názvu vyplývá, o zařízení k přesnému seřizování puškohledů. Prvotní myšlenka na vznik této práce nastala při velmi zdlouhavém, v některých případech i finančně dosti náročném, manuálním seřizování puškohledu. Po prvotní myšlence bylo zjištěno, že žádná automatizace nebo zjednodušení neexistuje. Čili ve výsledku existuje výrobce zbraní na straně jedné, existuje výrobce puškohledů na straně druhé, ale neexistuje výrobce zařízení, který by toto kompletoval do jednoho celku a distribuoval. Dalším zajímavým faktorem této práce je propojení dvou dosti vzdálených oborů. A důvod, proč se zabývat touto problematikou a řešit ji pomocí počítačů? Odpověď je prostá: jde o přesnost nastavení. Více bude přiblíženo na následujícím příkladě.

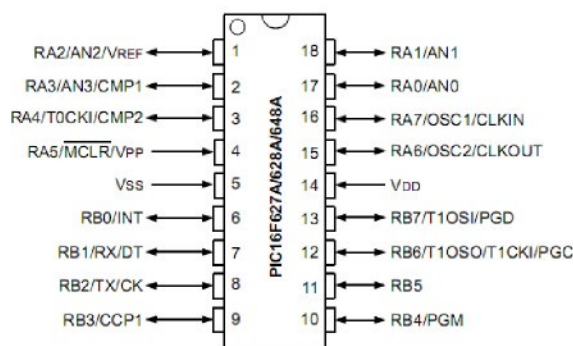
Jelikož jsme se zmínili o sportovních zbraních a puškohledech, představme si například střelecký šampionát se střelbou na delší vzdálenosti, například okolo 200 m. Každý sportovní střelec, který si vezme do ruky zbraň, zalící a zamíří, tak očekává, že kam zaměří osnovu puškohledu a odmáčkne spoušť na zbraň, kulka dopadne přesně na místo zaměření a ne na místo například o dva, tři centimetry vedle. Tyto chyby rozhodují o celkovém umístění. Tato odchylka je ve většině případech při manuálním seřizování dosti velký problém. Bohužel člověk nikdy neudělá nic tak přesně jako počítač, který vše propočítá a nastaví s největší přesností, především když si uvědomíme, že každá zbraň má v přesnosti svoji toleranci, která platí stejně i pro puškohledy a jednotlivé střelivo. S těmito tolerancemi se práce bude zabývat především v řešení algoritmů. Samozřejmě, do přesnosti nastavení se musí počítat i vedlejší vlivy, jako je vítr, déšť a mnohé další.

2. Hardware

Následující blok je věnován teorii jednotlivých součástek, komponent a nebo tzv. „celých setů“ s popsáním komunikační linky RS-232C, která také spadá do hmotného zařízení, jelikož se jedná o drátový přenos dat.

2.1 Mikrokontrolér PIC

Celé mechanické zařízení, které má na starosti řídit pohyby krokových motorků, musí být řízeno nějakým samostatným mikrokontrolérem, umístěném mezi ovládací aplikací, která vyhodnocuje všechna potřebná data k úspěšnému nastavení puškohledu a mezi budičem krokových motorků, který na bázi dat již vykonává konkrétní pohyby motorkami. K tomuto účelu byl zvolen mikrokontrolér PIC16F627A. Na obr.2.1 převzatém z [www 11], lze vidět rozložení jednotlivých pinů.



(Obr. 2.1) – Mikrokontrolér PIC16F627A

Mikrokontrolér PIC je programovatelná jednočipová součástka. Tato součástka je na principu harvardské architektury. To znamená, že mikropočítač má dvě paměti, které jsou navzájem oddělené. Jedna paměť slouží pro data a druhá pro samostatný program. Tyto dvě paměti nemají datová slova stejně dlouhá.

Mikrokontroléry mají především uplatnění v různých průmyslových oborech pro řízení nejrozličnějších systémů. Pro svou velmi malou velikost, pořizovací cenu a především pro svou univerzálnost a nízkou spotřebu elektrické energie, lze tyto součástky v dnešní době nalézt téměř v každé produkované spotřební elektronice, lékařských přístrojích, dopravních zařízeních atd. Více na [www 12].

2.1.1 Typy mikrokontrolérů PIC

Každé zařízení, které využívá těchto součástek má jiné nároky na vlastnosti mikrokontroléru, proto existují v několika provedeních, a to:

- mikrokontroléry, které mohou být naprogramovány pouze jednou. Tyto součástky mají ve svém označení písmeno C
- mikrokontroléry, které mohou být programovány opakovaně a mají paměť EEPROM
- mikrokontroléry, které mohou být programovány opakovaně a mají paměť FLASH. Tyto mikrokontroléry mají ve svém označení písmeno F

Dále si tyto mikrokontroléry můžeme rozdělit do skupin podle šířky datového slova a šířky programového slova. Datová slova jsou délky například 8, 16 a 32 bitů. Oproti tomu programová slova mají zastoupení ve větším spektru, a to například 10, 12, 14, 16 bitů a dalších. Z předchozích informací vycházejí názvy řad mikrokontrolérů. Více viz [www 12].

2.1.2 Instrukční soubor

Tyto mikrokontroléry stojí na architektuře RISC. To znamená, že mají zredukovanou instrukční sadu, která čítá 33 - 58 základních instrukcí (závislé na typu). Všechny tyto instrukce se vykonávají pouze v jednom cyklu (jedinou výjimkou jsou instrukce skoků). Soubor instrukcí se skládá ze tří skupin:

1. pro práci s bajty
2. pro práci s bity
3. pro práci s konstantami

Více viz [www 12].

2.1.3 Komponenty mikrokontrolérů

Každý mikrokontrolér PIC vlastní nějaké implementované periferie i komponenty pro jeho komunikaci s okolím a jeho běžnou funkci. V následující části je jen malý, avšak důležitý výčet z nich:

- oscilátory
- A/D převodníky

- I/O vstupy a výstupy binárního charakteru
- USART
- Časovače
- Rozhraní pro RS232, CAN, USB

Více viz [www 12].

2.1.4 Programátory

Mikroprocesory, které nemají hned po výrobě svou upotřebitelnost v nějakém elektronickém zařízení a slouží jako polotovary na prodej, jsou bez nainstalovaného softwaru. Pokud je ho třeba pro vývoj nového produktu, je zapotřebí mu software nejprve implementovat. Úprava obvodu (mikrokontroléru PIC) pro příslušný produkt se provádí „nehmotným“ procesem, a to naprogramováním. Výhodou tohoto postupu je snadnost modifikovat a upravovat funkci výrobku a především urychlení vývojové fáze s velmi nízkými náklady. Ve výsledku se to projeví na konkurenceschopnosti finálního produktu. K tomuto procesu (naprogramování) je třeba hardwarového programátoru, který nahraje příslušný software do mikrokontroléru. Dnešní trh nabízí několik typů kvalitních programátorů [www 9] [www10]:

- | | |
|---|--|
| <ul style="list-style-type: none"> • Presto Asix <ul style="list-style-type: none"> - Vysoká rychlost naprogramování - Řízení a napájení skrze USB - Programování ICSP - Diagnostické LED [www 9] • MPLAB REAL ICE <ul style="list-style-type: none"> ○ Plná rychlost emulace ○ Rychlé ladění a programování ○ Malé rozměry [www 10] | <ul style="list-style-type: none"> • PICKit 3 <ul style="list-style-type: none"> - Komunikace s PC skrze USB - Podpora nízkého napětí 2,0 V - Čtení a zápis programu a datové paměti [www 8] |
|---|--|

Další programátory a mnohé rozšiřující informace o mikrokontroléru jsou uvedeny [www 12]



(Obr. 2.1.4) – Programátor mikrokontrolérů PIC – převzato z [www 9]

2.1.5 Program mikrokontroléru

Pro správnou funkci a komunikaci mezi budičem a řídicí aplikací bylo třeba napsat mimo jiné i program pro mikrokontrolér. Tento program ovládá jednotlivé výstupy mikrokontroléru na bázi přijatých dat z aplikace po sériové lince RS-232C. V prvním kroku se nejdříve musel nainkludovat hlavičkový soubor `pic.h`, který má v sobě implementovány vlastnosti některých vybraných typů mikrokontrolérů PIC. Pro tuto práci je důležitý použitý mikrokontrolér PIC16F627A, který soubor `pic.h` má vlastnosti mikrokontroléru již v sobě implementován. Dále bylo třeba definovat výstupní piny pro ovládání obou použitých krokových motorků (podle obr. 2.1 to jsou piny 6, 9, 10 a 11) a komunikaci po sériové lince.

Po těchto několika nastaveních lze již přistoupit k samotnému výkonovému kódu, který na základě dat přijatých od řídicí aplikace aktivuje jednotlivé výstupní piny mikrokontroléru PIC16F627A a tím je vyvolán pohyb krokového motorku. V kódu je implementováno rozhodování, zda-li se poslala data, která se mají akceptovat, či nikoli. Tato data mají svůj předdefinovaný formát. Program je k nahlédnutí v příloze na CD.

2.2 Krokové motorky

Existuje motor, který je řízen impulsním napájením. Jeho pohyb je nespojitý, jelikož je přerušován jednotlivými pulzy, kde každé vyvolání pulzu pootočí o nějaký úhel α rotor motorku. Pulzy se vyvolávají pomocí budiče nebo ovladače krokového motorku, který systematicky řídí jednotlivé fáze vinutí. Budiče se dají snadno sestavit nebo si je lze zakoupit (příkladem může být budič SMCU 05-P). Pro vlastní pochopení textu je třeba znát několik základních pojmů krokových motorků:

- **Krok** je jednotkový mechanický pohyb rotoru na řídicí impuls z jedné polohy do druhé, ale té nejbližší
- **Velikost kroku** je o jaký úhel α se pootočí rotor motorku na jednotkový impuls
- **Statický úhel zátěže** je o jaký úhel β se pootočí rotor motorku při zátěžném stavu na jednotkový impuls

- **Otáčky rotoru** jsou kolik rotor vytvoří otoček za jednotku času, tyto otáčky se dají spočítat jak je uvedeno v literatuře [www 5]

$$n = (60 * f_k * \alpha) / 360$$

kde: n je počet otáček za minutu

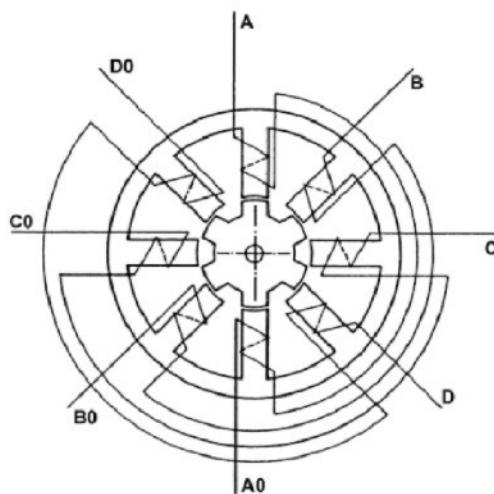
f_k je kmitočet kroků v Hz

α je velikost kroku ve stupních

[citováno: www 5]

2.2.1 Řízení krokových motorků

Stator krokového motorku je v podstatě tvořen několika navinutými cívkami na každém z pólů. Na převzatém obr.2.2.1 z literatury [www 5] je pro demonstraci vytvořen imaginární řez čtyřfázového motorku s pasivním rotorem. Na obrázku si lze povšimnout, že na statoru motorku je 8 pólů a každý z nich má na sobě navinuty cívky. Protější póly spolu uzavírají dvojici a tím tvoří jednu větev, nebo-li jednu fázi. Fáze jsou označeny jednotlivě jako A,B,C a D. Oproti tomu, rotor je tvořen pouze ze šesti pólů a nemá žádné vinutí. Pokud výstupy A, B, C a D připojíme k nějakému budiči, nic nebrání k jeho buzení a řízení.



(Obr. 2.2.1) – Řez čtyřfázovým krokovým motorkem

2.2.2 Čtyřtaktní ovládání po jedné fázi

Jak z názvu vyplývá, jedná se o řízení jednotlivých fází ve čtyřech taktech. Hned na začátku je nutné definování stavů. Pro jednoduchost : uvedené hodnoty v závorkách budou znázorňovat, zda-li se na dané větvi vyskytuje napětí nebo ne. Čili v první fázi buzení nastane (1,0,0,0), tedy větví A protéká proud, ale skrze ostatní nikoli. Rotor se natočí tak, jak bude mít nejmenší magnetický odpor. To má pouze v jedné poloze, jelikož zbytek pólů jsou neaktivní, takže si nejbližší póly rotoru cívky přitáhnou do souhlasné polohy. Pokud budič rozhodne o aktivaci dalšího řídicího impulsu, změní se kombinace buzení a to na (0,1,0,0). Tímto se nabudí větev B a zbytek je neaktivní. Rotor se analogicky potočí jako v předešlém případě a vše se opakuje, dokud bude aktivní budič. Čili v dalších krocích nastanou situace (0,0,1,0) a (0,0,0,1).

Pohyb krokového motorku je $\frac{1}{4}$ zubové rozteče na rotorových pólech. V tomto případě je rozteč rotorových pólů 60° a $\frac{1}{4}$ čili 15° je velikost jednoho kroku

2.2.3 Čtyřtaktní buzení po dvou fázích

Ovládání tímto způsobem je obdobné jako předchozí, ale liší se tím, že jsou spínány vždy dvě sousední fáze. Čili spínání fází bude v tomto pořadí (1,1,0,0), (0,1,1,0), (0,0,1,1) a (1,0,0,1). Velikost kroku zůstává stejná jako v předchozím řízení (15°). Výsledná poloha rotoru nebude souhlasně s póly na statoru, ale bude souhlasně mezi dvěma sousedními póly. Síla motorku bude téměř dvojnásobná, ovšem s větším odběrem proudu.

2.2.4 Osmitaktní buzení krokového motorku

Osmitaktní řízení není nic jiného, než kombinace předešlých dvou způsobů. To znamená, že vložíme kombinace buzení dvoufázové mezi ty jednofázové. Pro ilustraci spínání bude vypadat nějak takto: (1,0,0,0), (1,1,0,0), (0,1,0,0), (0,1,1,0), (0,0,1,0), (0,0,1,1), (0,0,0,1) a (1,0,0,1). Tyto kombinace se opakují stále dokola, ať v jednom nebo druhém směru. Tímto řízením se lehce dosáhlo polovičního kroku motorku. Původně to bylo 15° a tímto řízením je to pouhých $7,5^\circ$. Obvyklá velikost kroku u takovýchto krokových motorek bývá mezi $1 - 5^\circ$. Díky své jednoduché konstrukci lze očekávat i nízkou cenu.

2.3 Budič krokového motorku

Jen za pomoci napsané aplikace je zcela nemožné řídit jednotlivé fáze krokového motorku. Proto byla zakoupena hotová stavebnice unipolárního budiče krokových motorků pod názvem SMCU05-P od firmy HOBBYROBOT. Tento budič pracuje v osmitaktím řízení motorku. Díky unipolárnímu řízení je zajištěn velký krouticí moment. Budič lze vidět na obr.2.3.



(Obr. 2.3) – Budič krokového motorku

2.3.1 Základní vlastnosti

- *Motor je řízen signály STEP a DIR v napěťových úrovních TTL*
- *Maximální frekvence řídicího signálu STEP: 40kHz*
- *Maximální proud jedné fáze: 250mA*
- *Maximální napájecí napětí: +24V ss*
- *Osmitaktí indexer je realizován na bázi mikroprocesoru PIC (microchip)*

[citováno: www.13]

2.3.2 Popis činnosti

Tento modul SMCU05-P je určen pro unipolární dvojfázové krokové motorky (dále jen KM) se šesti vývody. Alternativním použitím krokového motorku se šesti vývody může být ještě s pěti a osmi vývody. Pokud by to byl KM jiný (například se čtyřmi vývody), tak je použití tohoto budiče zcela nevhodné.

Hlavní aktivní součástí, která ovládá celý budič a řídí fáze jednotlivých větví KM, směr otáčení, ale i rychlost v osmitaktním režimu, je integrovaný obvod SB3101 (mikrokontrolér PIC 12F629), který již byl naprogramován z výroby.

Dalším integrovaným obvodem na tomto budiči je ULN2803, který je připojen na výstupy předchozího obvodu. Tento obvod tvoří osm výkonových tranzistorů v darlingtonově zapojení a navíc obsahuje i integrované ochranné diody. To je vše osazeno v pouzdře DIL18. Pokud by z nějakého důvodu byla výkonová ztráta tohoto obvodu v podobě tepla moc velká, pak je umožněno přilepení přídavného žebrového chladiče na viditelnou plochu pouzdra.

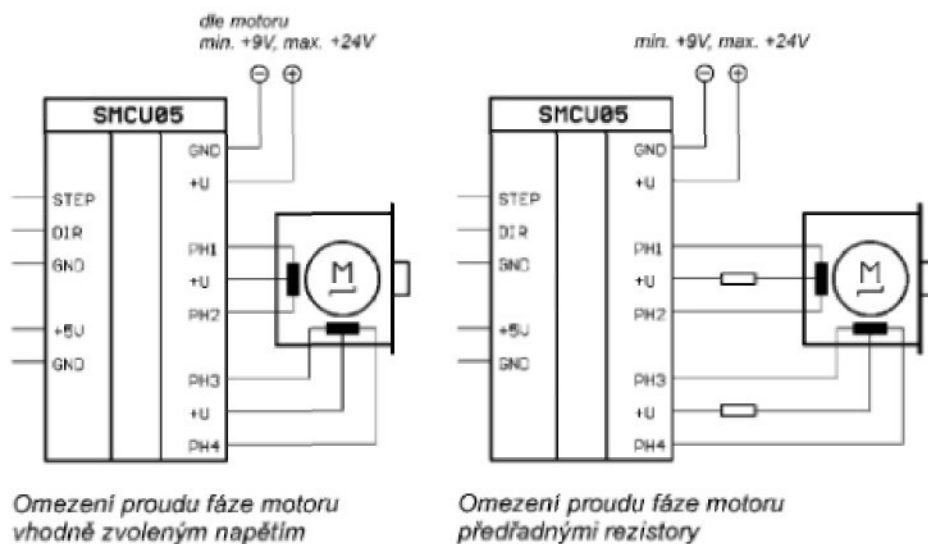
Aby obvod SB3101 měl správnou funkci bez chyb, nesmí mít žádné výkyvy napětí. Proto je potřebné zajištění přívodu stabilizovaného napětí 5V. Již zmíněných stabilizovaných 5V obstarává monolitický stabilizátor napětí pod označením 7805 v pouzdře TO220. Toto stabilizované napětí je také vyvedeno i na druhý pin konektoru XC1, ze kterého je možné napájet různá externí zařízení (pokud je třeba). Tento pin je využit pro napájení napěťového převodníku na TTL úroveň MAX232. Stabilizátor 7805 je na destičce upevněn distančním sloupkem a tím je i vhodně zajištěno i dobré chlazení stabilizátoru.

Neobvyklou součástí v tomto budiči je varistor. Varistor je bezpřechodová symetrická součástka, která funguje v podstatě jako napěťově proměnný rezistor. Při vzrůstajícím napětí se odpor snižuje. Použitý Varistor má skoro nekonečný odpor až do maximálního možného napájení budiče a to do 24V. V tomto obvodu má varistor význam chránit budič před napěťovými špičkami, které mají často vliv na chybnou funkci, dokonce v jednom z nejhorších případů, i trvalého poškození mikrokontroléru PIC a dalších obvodů či součástek.

Výstup z budiče je vyveden do šroubových svorek, kde svorky 3 až 8 slouží pro řízení fází KM a svorky 1 a 2 slouží k napájení KM. V případě této práce je napájecí napětí rovno 12V.

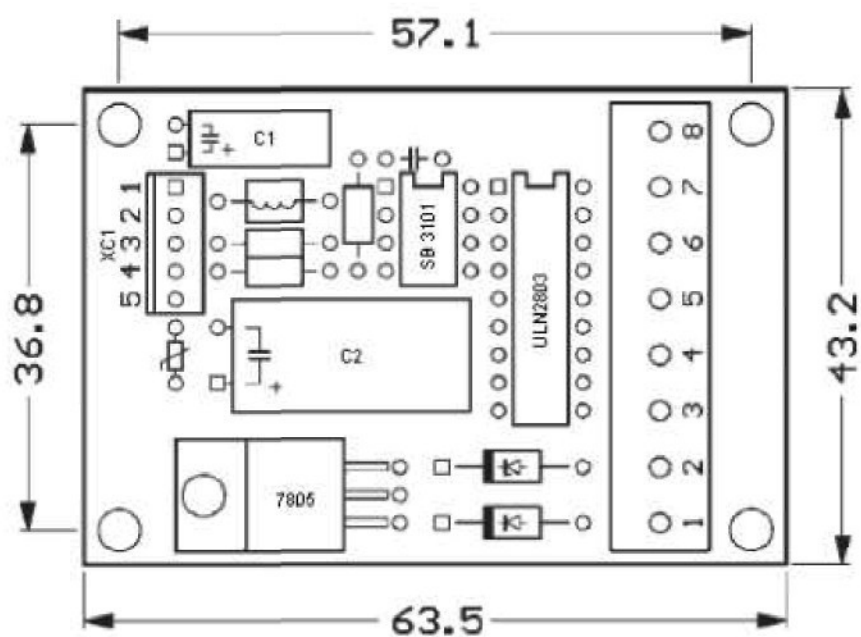
[www 13]

2.3.3 Připojení KM k budiči



(Obr. 2.3.3) – Připojení krokového motorku k budiči

2.3.4 Rozložení konektorů a součástek



(Obr. 2.3.4) – Rozložení konektorů a součástek na budiči KM

Rozložení pinů na konektoru XC1 je : 1 – GND

2 – +Ur (5V / 0.1A)

3 – STEP (krok) – úroveň TTL

4 – DIR (směr otáčení motoru) – úroveň TTL

5 – GND

[citováno: www 13]

Rozložení pinů svorkovnice pro krokové motorky:

1. GND

5. PH1

2. +9V až 24V (12V)

6. PH2

3. +9V až 24V (12V)

7. PH3

4. +9V až 24V (12V)

8. PH4

Svorky 2, 3 a 4 se mohou navzájem zaměnit. Hodnoty napětí uvedené u těchto pinů jsou v rozmezí z důvodu proměnlivosti napájení krokových motorků. Hodnota uvedená v závorce je konkrétní nastavená hodnota na zdroji napájení pro použité motorky. Piny 5,6 a 6,7 nelze mezi sebou prohodit. Tyto dvojice pinů jsou uzavřené okruhy cívek v krokových motorkách. Pokud by se vzájemně prohodily, nastalo by špatné spínání fází a uvnitř motorku by došlo k chaosu. Navenek by se to projevovalo různorodým škubáním hřídelky na jednu i druhou stranu.

Uvedené obrázky (Obr.2.3.3 a Obr.2.3.4) jsou převzaty z literatury [www 13]

2.4 Sériová linka RS-232C

Linka RS-232C, občas nazývaná také jako UART (v podstatě se ale jedná o nepřesné označení), byla v minulosti navržena pro připojení k modemu s textovými terminály. Později se začala sériová linka pro svoje vlastnosti propojení dále rozšiřovat mezi ostatní komponenty. Začala se k ní připojovat nejrůznější zařízení jako jsou počítačové myši, plottery, tiskárny, počítače a mnohá další. Dnes je tento port především využíván například k programování regulátorů motorů, ke správě některých síťových prvků, serverů, jelikož ještě mnohé servery podporují již zmíněné textové terminály, pro komunikaci s mikroprocesory PIC atd. Více viz [www 1] a [www 2].

2.4.1 Princip sériové komunikace

Základním účelem je posílání dat po jednotlivých bitech v simplexním nebo poloduplexním módu. Kromě přeposílaných dat (znaků, informací o stavu nějakých tlačítek, atd.) se tento proud informací rozšiřují ještě o další bity, které zajišťují synchronizaci. Tyto bity tam musí být zakomponovány, aby přijímací strana od sebe jednotlivé bity rozeznala. Pro synchronizaci je možné využití ještě jednoho vodiče s hodinovým signálem.

Na začátku bylo řečeno, že se port RS-232C občas nazývá jako UART. Přesněji RS-232C jenom spadá do rodiny zařízení UART. Ve významu to znamená, že se jedná o zařízení, které má vlastnosti jako universální asynchronní přijímač a vysílač. Nejvýznamnější je slovo asynchronní, které definuje, jak se proud bitů bude přenášet. Asynchronní přenos obvykle nepoužívá žádného vedlejšího synchronizačního vodiče s hodinovým signálem a ani v přenášených datech se neobjevují informace o synchronizaci. Po uvedení předchozích informací by to znamenalo, že jak vysílací strana, tak i přijímací strana, by musela být vybavena přesným zdrojem hodinových signálů, které by řešily detekci začátku a konce bloku přenášených bitů. V praxi tohle řešení je ale zcela nemožné, jelikož jakýkoli dostupný krystal, který je byť považován za opravdu velmi přesný, tak by během pár sekund vykázal tak výraznou odchylku, že by to bylo neúnosné. Přesto se nějaká synchronizace u zařízení UART používá, ale ne na této úrovni jednotlivých bitů. Více bude přiblíženo v následující podkapitole.

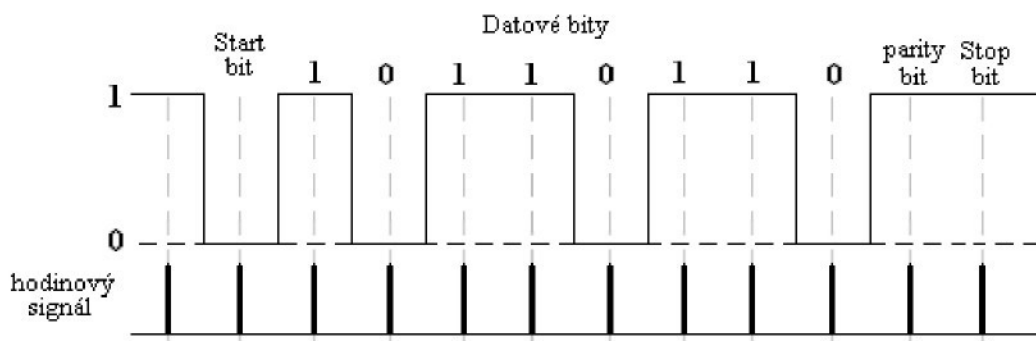
[www 1]

2.4.2 Asynchronní přenos dat

Tento typ přenosu využívá tzv. synchronizačních značek, které se vyskytují na začátku a na konci bloku bitů. Nejčastěji se jedná o celý byte. Nejprve, před veškerou komunikací, je třeba vhodně nakonfigurovat obě zařízení tak, aby přijímací prvek předem „věděl“, jak bude definován formát posílaných dat a jaká bude přenosová rychlost uváděná v bitech za sekundu neboli „bitrate“. Nadále se musí na obou zařízeních nastavit počet datových bitů v jednom bloku, jaká se má použít parita (sudá, lichá, žádná, mezera a značka), která slouží jako nejjednodušší ochrana před chybou posílání a poslední nastavení tzv. počtu stopbitů.

Důležité je si uvědomit, že nastavení těchto parametrů musí být jak na vysílací straně, tak na přijímací straně naprosto shodné, což je především důvod k tomu, aby byly ve většině případech UART zařízení konfigurovatelné, aby spolu mohly vzájemně komunikovat bez nějakých výraznějších chyb.

Před přenosem je datový vodič neaktivní, čili má na vodiči nastavenou logickou 0 na úrovni +12V. Před samotným začátkem přenosu datových bitů (jejich počet přímo závisí na konfiguraci sériové linky, zda-li se pošle oněch 5 až 8 bitů), je v první řadě poslán „start bit“, který je nastaven na 0, což zaručí změnu stavu linky. Na přijímací straně je velmi důležité nastavení citlivého rozeznání změny stavu linky, aby byl rozeznán začátek „start bitu“. V této chvíli je přijímací straně známo, že má přijmout předem definovaný počet bitů s rychlostí, která byla na začátku nakonfigurována pro vzájemnou komunikaci. Na konci toku bloku dat se může vyskytovat již zmíněný paritní bit a bezprostředně po něm následuje „stop bit“. Ten je nastaven na logickou 1, jak je vidět na Obr.2.4.2



(Obr. 2.4.2) – Přenos dat po RS-232C

[www 1] [www 3]

2.4.3 Technický popis

Klasický sériový port RS-232C má 9pinový konektor k připojení k jinému hardwaru. U přenosu dat je třeba znát napěťové úrovně logických stavů 0 a 1. Napěťové úrovně jsou bipolární, ale jejich velikosti nejsou jednoznačné. Hodnoty mohou u těchto úrovní nabývat hodnot $\pm 5V$, $\pm 10V$, $\pm 12V$, ale i dokonce $\pm 15V$. Hodnoty závisí na použitém zařízení. Nejčastěji užívanou úrovní je však $\pm 12V$, kde $-12V$ je logická 1 a $+12V$ je logická 0.

Na obr.2.3.4 lze vidět 9pinový konektor DB9 a rozložení jednotlivých vývodů, které jsou pro přenos dat nezbytné. Základní piny jsou pro příjem (RxD), vysílání (TxD) a zem. Zbytek nemusí být zapojen, neboť to jsou jen rozšiřující funkce.



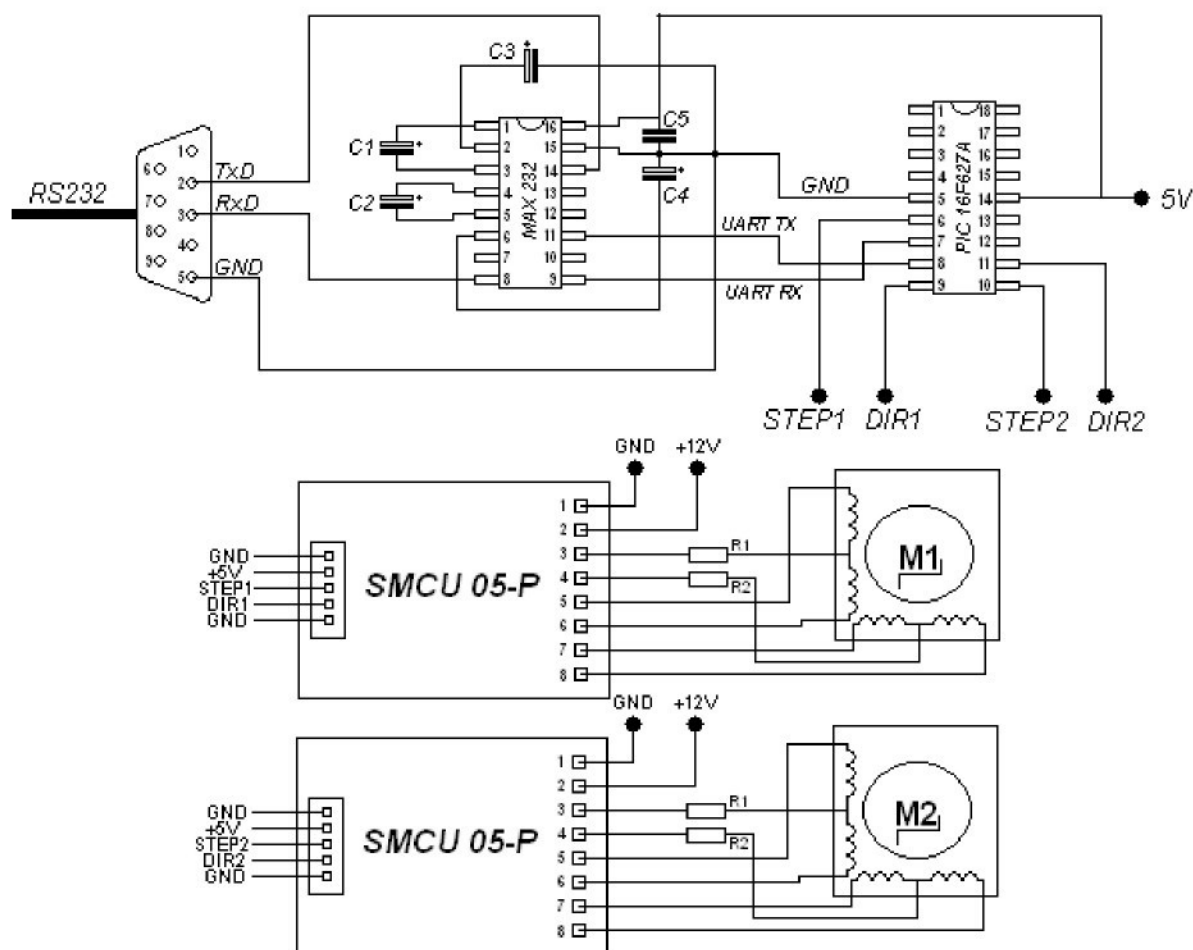
Pin	zkratka	význam
1	DCD	Data Carrier Detect
2	RxD	Received Data
3	TxD	Transmitted Data
4	DTR	Data Terminal Ready
5	SG	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CS	Clear to Send
9	RI	Ring Indikator

(Obr. 2.4.3) – konektor RS-232C DB9

2.5 Propojení RS232 a krokových motorků

Zde nastává problém již na samém počátku. A to pokud PC nebo notebook, na kterém je spuštěna ovládací aplikace pro seřizování puškohledu, není vybaven sériovým portem RS232. V tomto případě se musí přejít k alternativě, a to za pomoci portu USB. Jedním z řešení je si zakoupit nebo si postavit vlastní převodník z USB na RS232. Z konektoru RS232 je zapotřebí pro propojení s budičem pouze tři pinů, po kterých bude vedena celá komunikace. Tyto piny jsou označeny jako RxD, TxD, a GND.

Dále se musí vyřešit připojení k budiči. Samostatná sériová linka RS232 nelze propojit napřímo s budičem krokových motorků. Mezi těmito dvěma komponenty potřebuje ještě další dvě součástky, nebo-li obvody, a to mikrokontrolér PIC16F627A, který na základě dat, které mu přeposlala ovládací aplikace rozhodne, jakým směrem se má každý motor otáčet a o kolik kroků. Druhou součástí je převodník napěťových úrovní TTL MAX232, který nám zajistí přesných a stabilizovaných 5V. Na obr.2.5.1 lze vidět celé funkční zapojení



(Obr. 2.5.1) – Zapojení řídicího HW

Význam všech uvedených pinů u převodníku napěťových úrovní MAX232 lze dohledat v literatuře [www 14] , u mikrokontroléru na obr.2.1 a u budiče krokových motorků SMCU05-P na obr.2.3.4.

KM je připojen do svorkovnic na budiči způsobem, jak je uvedeno na obr.2.5.1. V tomto zapojení si lze všimnout připojených rezistorů R1 a R2. Jejich velikost je zvolena dle připojených konkrétních motorků z důvodu, že každý druh motorku má více či méně odlišné parametry. Paradoxem celého zapojení je, že celé toto hardwarové zařízení je napájeno zpětně, téměř od výstupu. Vstup napájení je na svorkách +12V na budiči SMCU05-P. Tímto napětím jsou napřímou napájeny krokové motorky (M1 a M2) určené pro vertikální i horizontální pohyby zaměřovací osy u puškohledu. Ve vnitřním obvodu budiče je toto napětí upraveno a stabilizováno na pouhých 5V, které jsou vyvedeny na volně použitelný pin. Jelikož integrované obvody MAX232 a mikrokontrolér PIC jsou standardně napájeny 5V, tak

zmíněný pin je připojen k těmto obvodům. Obrovskou výhodou tohoto zapojení je, že není potřeba použít více napěťových zdrojů, ale stačí pouze jen jeden. Zemní piny všech obvodů v zapojení jsou propojeny.

U obvodu MAX232 jsou připojeny kondenzátory C1 až C5. Kondenzátor C5 je keramický o velikosti 100nF. Oproti tomu zbylé kondenzátory jsou elektrolytické o shodné velikosti 10 μ F dle [www 6] U těchto čtyř kondenzátorů je třeba mít na paměti připojení se správnou orientací nebo by mohlo dojít k explozi elektrolytu a i jinému lehkému, v horším případě těžkému, poškození hardwaru.

Pro napájení celého hardwaru byl sestaven stabilizovaný zdroj, kterému lze plynule měnit výstupní napětí v rozmezí 0-30V. To samé platí pro proudové omezení, které lze nastavit na hodnoty v rozmezí 0-3A. Napětí zdroje je nastaveno na pouhých 12V, jak bylo výše uvedeno a s proudovým omezením 250mA. Toto proudové omezení je nastaveno tak, aby nedošlo k překročení maximálního dovoleného proudu pro jednotlivé fáze motorku, které by mohlo vést k jeho destrukci.

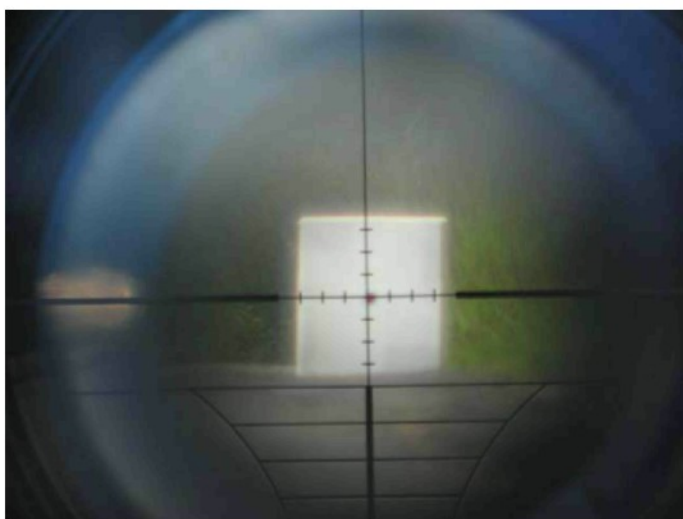


(Obr. 2.5.2) – Testovací zapojení řídicího HW

2.6 Snímací buňka

K této problematice lze přistoupit dvěma způsoby. Jak je obvyklé ve veškerých úlohách, kde se vyskytuje několik řešení a ne jen jedno, každé z řešení má svoje kladné, ale i záporné vlastnosti. Prvním řešením pro tuto problematiku je zhotovení snímací buňky přímo na puškohledu. Druhým řešením je sestavení snímací buňky u samotného terče.

Pokud bychom chtěli sestavit snímací buňku přímo na puškohledu, tak je třeba mít na paměti několik důležitých problémů, které si sebou toto řešení nese. Nejvýznamnějším problémem sestavení takového zařízení, které by bylo schopno snímat obraz skrze puškohled v takové kvalitě, aby se ze snímku dalo vyčíst kam byl proveden zásah a následně ho předat aplikaci k vyhodnocení. Jak je vidět na obr.2.6.1, tak to není skoro reálné. Snímek byl pořízen ze vzdálenosti pouhých 50ti metrů. Předpokladem celé práce je používání především delších vzdáleností, řádově ve stovkách metrů. Ze snímku pořízeném skrze puškohled si lze představit, jak by vypadal snímek pořízený například z dvouset metrové vzdálenosti. Kvalita obrazu, která by nás skutečně zajímala, čili skutečný terč, by byla velmi mizivá. Nemenším problémem by bylo nastavení snímacího zařízení před vstupní čočku puškohledu, jelikož zorný úhel puškohledu je ve velmi malém rozsahu a tím je i složitější zkalibrovat středy čoček obou komponent, puškohledu i kamery, tak, aby byly v přímém zákrytu. Na toto upevnění se samozřejmě musí nahlížet i z hlediska kvality výroby a detailního propracování, protože zpětné nárazy po výstřelu jsou velmi silné a s velikou pravděpodobností by nebylo možné zabránit pohybům kamery, což by ve výsledku ovlivnilo původní synchronizaci komponent a hlavně celý výsledek.



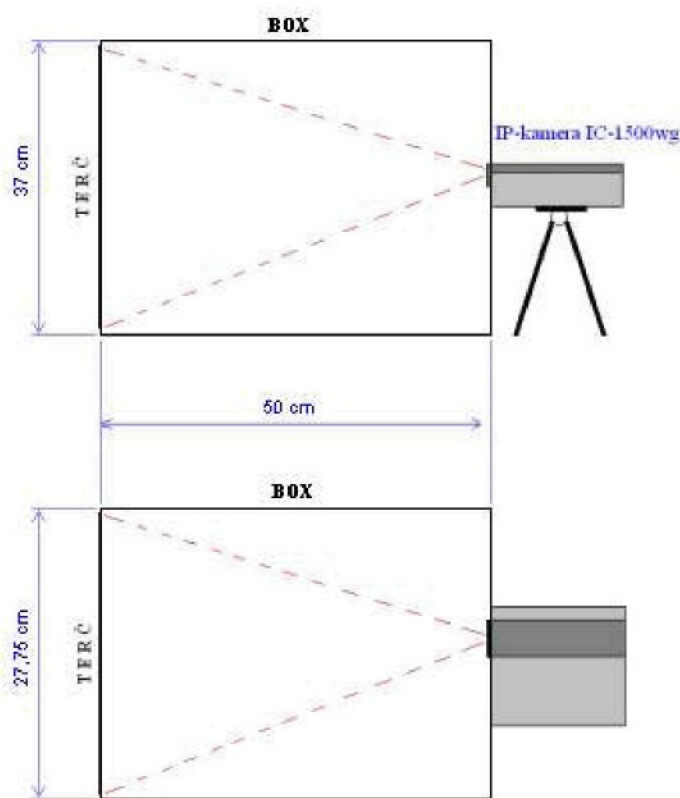
(Obr. 2.6.1) – Ukázka snímaného obrazu skrze puškohled

Pokud bychom ale byli schopni všechny tyto vedlejší vlivy nějakým výraznějším způsobem eliminovat, pak by tento způsob snímání byl na velmi dobré úrovni, jelikož si sebou nese i spousty dobrých vlastností. Kupříkladu by byla možná plná automatizace zaměření osnova kříže na střed terče, čili by nebyl od člověka očekáván žádný zásah do procesu nastřelování od uložení zbraně do nastřelovací stolice až po poslední kontrolní výstřel. Neméně významnou výhodou tohoto řešení je, že se nemusí řešit snímací buňka proti destrukci při nastřelování, jelikož na opačné straně střelecké dráhy je pouze samostatný terč.

Po zvážení předešlých problémů a nevýhod se muselo přejít k alternativě snímací buňky a to umístěním na straně terče. Zde nastal první problém při rozhodování zda bude lepší focení terče na volném prostranství za denního světla nebo zda není lepší uzavřený box, na který by se upevnil terč a kamera by byla umístěna uvnitř boxu ve tmě. Po zhotovení a prozkoumání vlastností fotek, jak z jednoho nebo druhého případu, bylo zjištěno, že realizace algoritmu pro vyhledání zásahu při focení terče za denního světla by byla náročná a velice zdlouhavá. Tyto fotky měly obrovský šum a spoustu vedlejších záznamů, kde se občas nedalo rozhodnout, zda-li se jedná o zásah nebo jen o chybu focení. Oproti tomu focení z boxu je na rozeznávání podstatně jednodušší, jelikož se využívá focení ze tmy do světla. Tím se zabrání jakémukoli šumu nebo chybě při focení, jelikož pozadí je černé, ale pouze vstup kulky do terče je výrazně světlý. Proto bylo zvoleno řešení s boxem.

Prozatím je snímací buňka provedena velmi jednoduchým a prostým způsobem. Jako box posloužila větší kartónová krabice ve tvaru krychle o rozměrech 50 x 50 x 50cm. Kartónová krabice byla vybrána především z důvodu, že netvoří skoro žádný odpor kulce při zásahu a vstup střely zůstává „otevřený“. Jako snímací komponenta byla vybrána wi-fi IP-kamera Edimax IC-1500WG, jelikož má velmi dobré parametry za dosti příznivou cenu. Tato kamera byla umístěna na opačnou stranu boxu než se vyskytuje terč, a to do nejzazší vzdálenosti na zmíněných 50cm. Snímaný obraz z této vzdálenosti byl změřen kvůli zornému úhlu kamery. Pokud známe tento úhel, lze snadno dopočítat, jakou hodnotu vzdálenosti má jeden pixel na snímku ve skutečnosti na terči. Po zkoumání byla tato hodnota stanovena na cca 0,578mm a je především důležitá pro řídicí aplikaci, která na základě těchto dat přistupuje k dalšímu řešení problematiky. Po pořízení snímku terče je tento snímek odeslán řídicí aplikaci skrze wi-fi síť do předem definovaného úložiště na pevném disku PC nebo popřípadě lze snímek odeslat na FTP server a přistupovat k němu tam. Napájením kamery je bateriový

12V zdroj. Napájení skrze bateriový článek je z důvodů, že povětšinou na žádné střelnici v červeném pásmu (střelecké dráze) není standardně přivedena elektrická síť.



(Obr. 2.6.2) – Snímací buňka

Největším a nejzásadnějším problémem snímací buňky pomocí boxu, je eliminování destrukce snímací kamery, jelikož je umístěna bezprostředně za terčem. Každým výstřelem je tato kamera ohrožena. Nabízí se dvě řešení:

- Řešení výsuvných modulů i s kamerou
- Snímání terče pod jeho výškovou úrovní

Výsuvné moduly se zdají být dobrým nápadem, pokud ovšem budou dobře chráněny v zóně, kam se projektil nedostane. Ale díky rozměrům a vlastnostem boxu (podle toho jak je sestrojen), je tento způsob nevhodný, protože by bylo třeba vytvořit onu bezpečnostní zónu, což není zrovna lehký úkol, zvláště u střelných zbraní.

Umístění kamery pod výškovou úrovní by znamenalo zešíkmení snímaného obrazu. Deformace obrazu se dá řešit vhodným algoritmem, ale problém nastává v tom, že pokud není obraz focen kolmo, tak nemusí být vidět zásah do terče. Tato situace se dá řešit tak, že se

zvětší vzdálenost od terče, tím pádem se změní zorný úhel. Toto řešení přináší další vedlejší problém, a to zbytečné snímání rozměrného obrazu, který v podstatě nepotřebujeme a také větší vzdálenost. Řešení šikmého obrazu a z toho dopočítávání hodnot, zvyšuje v dosti velké míře chybu při výpočtech a následně ovlivňuje celý výsledek.

Prozatímním řešením je manuální umístování kamery na stejné místo. Kamera je na pevném stativu o třech nožičkách, které se umísťují do označených míst, což nám zaručí snímání stejného obrazu.

3. Software

Tato část se bude zabývat především řídicí aplikací. Tato aplikace je v podstatě mozkiem celé práce. Veškeré výpočty, nastavení, ale i metodika způsobu nastřelování se provádí právě zde. Pro zdárné seřízení puškohledu ke zbrani, je třeba vědět, co aplikace všechno umožňuje, proto je vytvořen manuál, jak s ní zacházet a na co si dát pozor.

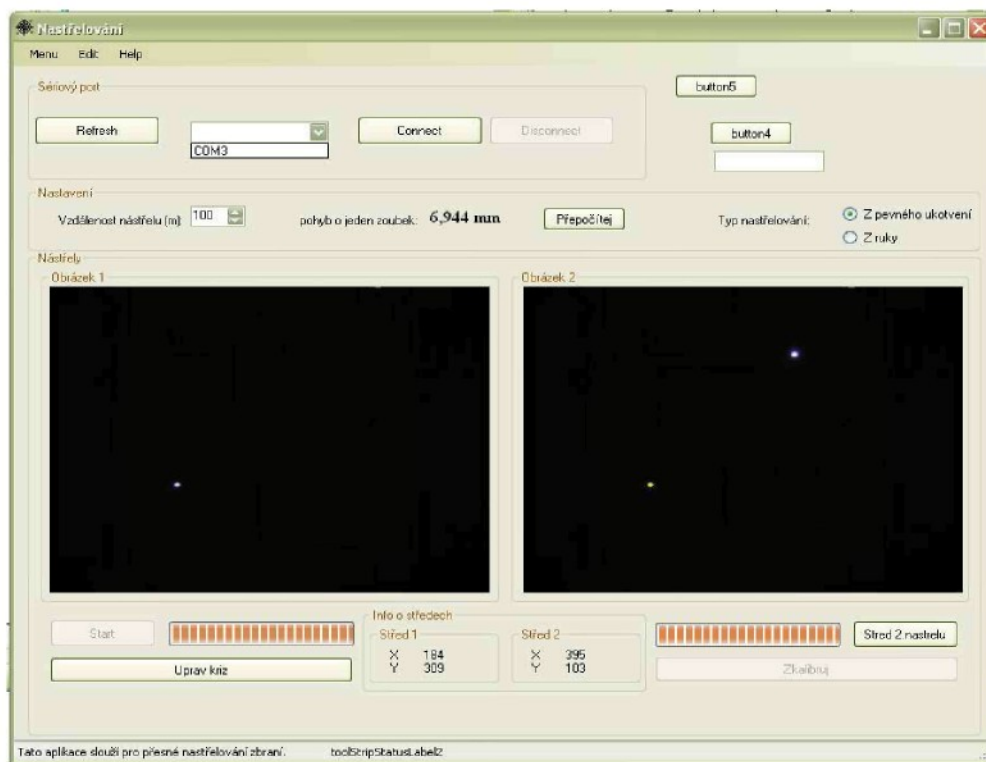
3.1 Ovládací Aplikace

Stěžejním předmětem této diplomové práce je vytvoření řídicí aplikace, která se stará o kompletní obsluhu a ovládání hardwarového systému. Pod tento systém patří nejen ovládání a seřizování puškohledu, ale i snímací buňka, ze které se posílají data, respektive fotky terče, na vyhodnocení souřadnic jednotlivých nástřelů (tyto obrázky jsou ukládány ve formátu .JPG a jsou zasílány do příslušného adresáře v PC popřípadě mohou být posílány na FTP). Tato aplikace je vyvíjena pro co nejširší oblast uživatelů, jelikož každý uživatel má jiné nároky na seřízení, a proto obsahuje nejrůznější drobná nastavení. Program je psán v programovacím jazyce C#, jelikož toto prostředí je uživatelsky přívětivé a srozumitelné.

3.1.1 Vzhled aplikace

Při prvotním stanovení cílů byl především kladen důraz na jednoduchost ovládání a ulehčení práce pro koncové uživatele. Proto je program vytvořen jako klasická okenní

aplikace, kterou dokáže ovládat i obyčejný laik. Na následujícím obr.3.1.1 je možno vidět hlavní okno programu.



(Obr. 3.1.1) – Vzhled aplikace

Po zapnutí programu se objeví okno s informacemi o aktuální verzi programu a datem jejího zhotovení, kontakty na autora, samozřejmostí je i umístění loga a uvítání s poděkováním za užívání této aplikace.

3.1.2 Menu aplikace

Po zapnutí aplikace a jejím zobrazení, lze vidět samostatné menu, které nabízí základní tři položky (Menu, Edit, Help), ve kterých se skrývá víceméně celé nastavování programu, ukončení nebo jen aktuální informace o aplikaci.

Menu

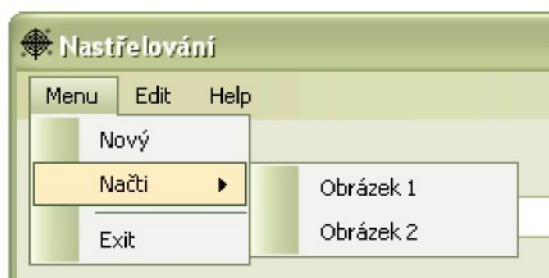
První záložkou v této liště je, jak název napovídá, „Menu“. Pod touto záložkou se nacházejí tři podkategorie dalších záložek. Konkrétně se jedná o tři a to:

- „Nový“
- „Načti“
- „Exit“

Pokud se jako první zvolí „Nový“, tak se nejdříve vymaže celá paměť a to i s načtenými obrázky, pokud tedy již byly užívány. Proto se tento způsob hodí pro opakované použití programu při jednom spuštění. Následně, hned poté, se objeví dialogové okno, které nabízí načtení prvního obrázku, nebo-li obrázku s první nástřelnou ranou v terči, který nám zaznamenala snímací buňka.

Další její záložkou je položka „Načti“, která opět obsahuje další podkategorie záložek, jak je vidět na obr.3.1.2.1 pro načtení prvního a druhého obrázku se zásahem terče, čili poskytuje načtení první a druhé nástřelné rány. Rozdílem oproti předchozímu je, že při použití těchto jednotlivých záložek nezasahujeme do celkové paměti programu, ale maže se jen ta část paměti, které již nebude v budoucnu potřeba (například pokud by bylo potřeba použít jeden nebo dva kontrolní nástřely navíc, tak aby se nesmazala data z předchozího nástřelu, která jsou důležitá pro následující seřízení, ale aby smazala pouze ta data, která byla použita ze začátku a nemají už na nic vliv).

Poslední záložkou v tomto menu je „Exit“. Jak již vyplývá z pojmenování záložky, jedná se o spolehlivé ukončení běhu programu. Jejím užitím se navíc bezprostředně uzavře (pokud již není uzavřena), komunikace po seriové lince RS-232 resp. otevřený COM port, přes který jsou posílána data pro mikrokontrolér PIC16F527A.



(Obr. 3.1.2.1) – Menu

Edit

Další součástí menu je záložka „Edit“, která obsahuje další tři podkategorie položek pro různé editace nebo nastavení běhu programu, dle přání každého uživatele a řešitele problematiky nastřelování. Těmito záložkami se rozumí:

- „Start“
- „Smazat“
- „Nastavení“

Při použití první záložky „Start“ se spustí samostatná činnost programu, pokud tedy bude již první obrázek nástřelu předán do aplikace a bude připraven k jeho vyhodnocení. Po jeho prohledání jsou vráceny hodnoty $[X,Y]$, s kterými lze nadále pracovat. Jestliže není tento obrázek připraven v aplikaci, nelze tuto funkce aktivovat a zavolá se chybové hlášení.



Info o středech	
Střed 1	
X	184
Y	309
Střed 2	
X	395
Y	103

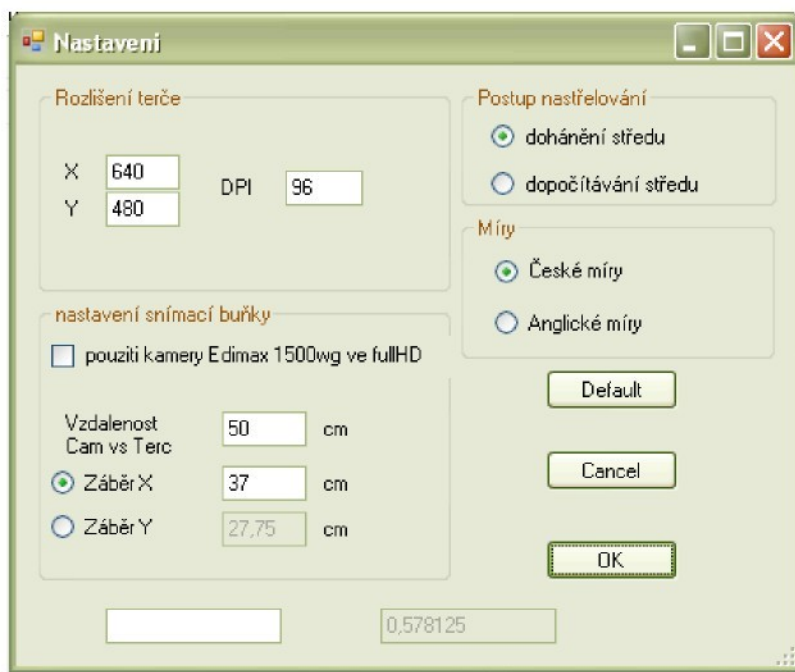
(Obr. 3.1.2.2) – Informace o středech

Pokud zvolíme v pořadí druhou záložku, kterou je „Smazat“, tak se zobrazí nabídka ke smazání prvního nebo druhého obrázku. Použití této funkce je především výhodné za situace, pokud se rozhodneme vymazat nebo změnit konkrétní nástřelný obrázek nebo dojde k častějšímu jevu, že nastala nějaká chyba při focení terče. Proto lze smazat jednotlivý obrázek z aplikace i s příslušnou částí paměti a pak lze znovu přistoupit k načtení nového obrázku.



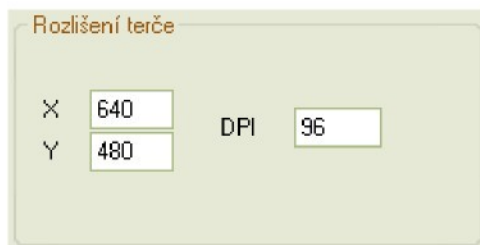
(Obr. 3.1.2.3) – Smazat

Zvolením třetí záložky v „Edit“ je jak je výše uvedeno na obr.3.1.2.3 „Nastavení“. Po kliknutí na tuto záložku se otevře nová okenní forma, která obsahuje kompletní nastavení kvality focení resp. rozlišení fotky, snímací buňky a nastavení zda-li se mají používat české či anglické jednotky. Samozřejmostí nastavení je i vybrání způsobu, jakým algoritmem se má celý systém řídit.



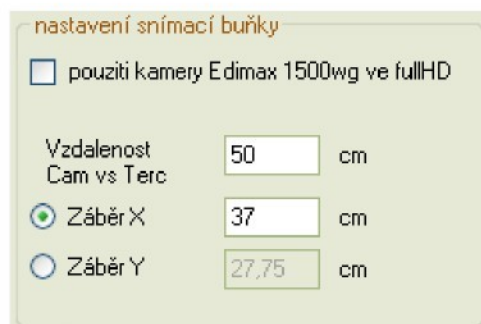
(Obr. 3.1.2.4) – Nastavení

Rozlišení focení terče je klasicky uváděno počtem pixelů na souřadnicích X (horizontálně - řádky) a souřadnicích Y (vertikálně - sloupce). Defaultní nastavení je na kvalitě VGA čili 640x480px o poměru stran 4:3 a rozložení 307200 pixelů na terč. Toto rozlišení je pro snímací buňku maximum, jelikož kamera není schopna obrázky o větším rozlišení snímat. Pro tuto kameru je popřípadě ještě vhodné požití méně kvalitního rozlišení QVGA 320x240 což odpovídá na obraz o 76800 pixelů. Samozřejmě platí pravidlo, čím větší rozlišení, tím přesnější souřadnice středu nástřelu program vyhodnotí. Jediné, co pravděpodobně bude zůstat stejné (ve většině případů) je , že i za použití lepší kamery bude nejčastější poměr stran, a to 4:3.



(Obr. 3.1.2.5) – Rozlišení snímku terče

V nastavení pro snímací buňku se lze rozhodnout pro standardní použití kamery Edimax 1500wg, kde se jen zadá vzdálenost kamery od terče pro vypočítání jejího záběru. Prozatím tato práce podporuje jen tuto kameru, ale není žádný problém práci rozšířit o další typy. Pokud ale nebude kamera z nějakých důvodů vyhovovat, může se použít jiná alternativa s rozdílem, že nebude předdefinovaná, ale zadá se záběr kamery manuálně.



nastavení snímací buňky

☐ použití kamery Edimax 1500wg ve fullHD

Vzdálenost Cam vs Terc cm

☒ Záběr X cm

☐ Záběr Y cm

(Obr. 3.1.2.6) – Nastavení snímací buňky

Důležitým nastavením je určení algoritmu, podle kterého bude řešeno vyhodnocování jednotlivých nástřelů. Tyto algoritmy se od sebe liší tím, že jeden využívá všech uvedených hodnot od výrobce puškohledů. A druhý algoritmus tyto hodnoty ignoruje a sám si je dopočítává. Tím má obrovskou výhodu oproti předchozímu, že hodnoty uvedené od výrobce mohou být špatně zadány, mohou být nepřesné nebo nemusí být uvedeny vůbec a přesto program bude schopen pracovat a vše zkalibrovat.



Postup nastřelování

☒ dohánění středu

☐ dopočítávání středu

(Obr. 3.1.2.7) – Postup řešení

Posledním nastavením je určení českých nebo anglických jednotek. Tyto jednotky se dají nezávisle měnit a to hned z několika prostých důvodů:

- Při koupi nového puškohledu jsou povětšinou uváděny jednotky v anglických mírách (například ¼"/100YD), proto to lze nastavit na jednotky které jsou na puškohledu uvedeny, aby se omezilo chybám při ladění a převádění mezi nimi.
- Při opačném pohledu je pro českého uživatele nepřírozené užívat cizí jednotky, jelikož s nimi nedokáže pracovat. Proto je možnost zaškrtnutí jednotek, které

jsou pro něj přirozené a uživatelsky lepší, protože s nimi dokáže rychle pracovat a odhadnout, jaká to vlastně vzdálenost je.



(Obr. 3.1.2.8) – Míry

Help

Poslední záložkou v menu je „Help“, která nabízí základní a aktuální informace o tomto programu.

Hlavní funkcí v této záložce je „Nápověda“, kde se koncový uživatel dozví, jak zacházet s aplikací, kdy a proč je třeba načtení prvního a kdy druhého obrázku s nástřelnou ranou., kdy je možné měnit jednotky, aby nenastaly nějaké neočekávané problémy při chodu programu. Zde se i dozvíme, které tlačítko k čemu slouží a mnohé další užitečné informace.

Úplně poslední nabídkou v „Help“ je klasická informativní záložka „O aplikaci“. Zde se nachází konkrétní informace o programu, jako je jeho verze, autor aplikace a celého systému, kontakt resp. kontakty na autora a reklamní logo.



(Obr. 3.1.2.9) – Help

3.1.3 Funkce aplikace

Realizace programu je uvedena ve dvou funkčních celcích. První celek vyhodnocuje informativní hodnoty a výsledky, druhý celek je výkonového charakteru, který už řeší jednotlivé problémy nastřelování puškohledu se zbraní.

Velmi důležitým parametrem prvního celku programu, je nastavování vzdáleností a to ať se jedná o vzdálenost zbraň-terč, kde se tyto hodnoty pohybují řádově v desítkách metrů resp. yardů nebo o vertikální a horizontální pohyb kříže v puškohledu (jedná se o změnu na terči a ne v tubusu optiky) v milimetrech resp. v palcích na přeskočení o jeden zoubek na šroubovici nastavovacích šroubů puškohledu v závislosti na vzdálenosti terče od místa výstřelu.

The image shows a software interface titled "Nastavení" (Settings). It contains a text input field for "Vzdálenost nástřelu (m):" with the value "100" and a small icon to its right. To the right of this is the text "pohyb o jeden zoubek: 6,944 mm". Further right is a button labeled "Přepočítej". On the far right, under the heading "Typ nastřelování:", there are two radio button options: "Z pevného ukotvení" (which is selected) and "Z ruky".

(Obr. 3.1.3) – Nastavení vzdáleností

Hodnota vzdálenosti je jednoduše zjistitelná měřením a minimálně se dá určit s přesností řádově na centimetry ba dokonce i na milimetry, podle použité metody. Jelikož se nastřelování musí provádět na střelnicích nebo místech k tomu určených, tak nejjednodušší způsob zjištění vzdálenosti (ale ne nejpřesnější) je, že ve řadě případů mají tyto střelecké dráhy orientační značky, které vyjadřují hrubou vzdálenost. Ideální by bylo využití laserového dálkoměru, ale bohužel toto zařízení je dosti náročné v nákladech na pořízení. Dálkoměry, které jsou schopné měřit vzdálenosti řádově stovky metrů, stojí na dnešním trhu kolem 10 000 Kč.

Oproti předešlé vzdálenosti střelby, hodnota uvedená jako „jeden zoubek na X metrů“ je spíše jen orientační, než-li přesná. Zde se musí vzít v úvahu dalších několik parametrů a záleží velmi na tom, jaký se projeví více či méně. Tyto parametry jsou především mechanického charakteru:

- Nepřesnost zbraně
- Nepřesnost projektilu (kulky)
- Nepřesnosti na uvedených hodnotách puškohledu
- Zákluz zbraně a pohyb nastřelovací stolice po výstřelu

Pokud budeme předpokládat tyto parametry jako ideální, pak hodnota dopočítaná v aplikaci bude naprosto absolutní. Nejhorší případ, nebo-li lépe - největší odchylka od této předpokládané hodnoty, nelze zde uvádět, jelikož ne vždy, když se sejdou všechny tyto faktory, které toto ovlivňují, musí zásadně změnit „ideální“ hodnotu. Pro ilustraci nastavíme předpoklad, že se sejdou tři parametry s hrubou chybou:

1. nepřesnost zbraně
 - 1cm / 100m – směr vpravo dolů
2. nepřesnost kulky
 - 2cm / 100m – směr vlevo
3. pohyb zbraně a NS při výstřelu
 - 1cm / 100m – směr nahoru vpravo

V tomto případě by se dosáhlo opět stejné a téměř přesné hodnoty. Pro určení odchylky nebo výstižněji - přesnost uváděnou v procentech, lze uvádět jen a pouze tehdy, pokud bude provedeno testování tohoto zařízení na několika různých zbraních, s různým střelivem, typů nastřelovacích stolic a samozřejmě i puškohledů.

Druhý funkční celek se zabývá aktivní částí aplikace. Tím je myšleno samostatné vyhodnocování dat a předávání výsledků na výstup. Výstupem je miněna sériová linka RS232. Mezi začátkem vyhodnocování a podání příslušných dat na výstup se dá program rozvětvit do pěti nebo sedmi bloků, podle použitého algoritmu:

1. pořízení snímku první nástřelné a následné načtení do aplikace
2. vyhodnocení zásahu
3. posunutí kříže
4. pořízení snímku druhé nástřelné a následné načtení do aplikace
5. vyhodnocení druhého zásahu
6. kalibrace (jen u jednoho algoritmu)
7. vyhodnocení třetího zásahu (jen u jednoho algoritmu)

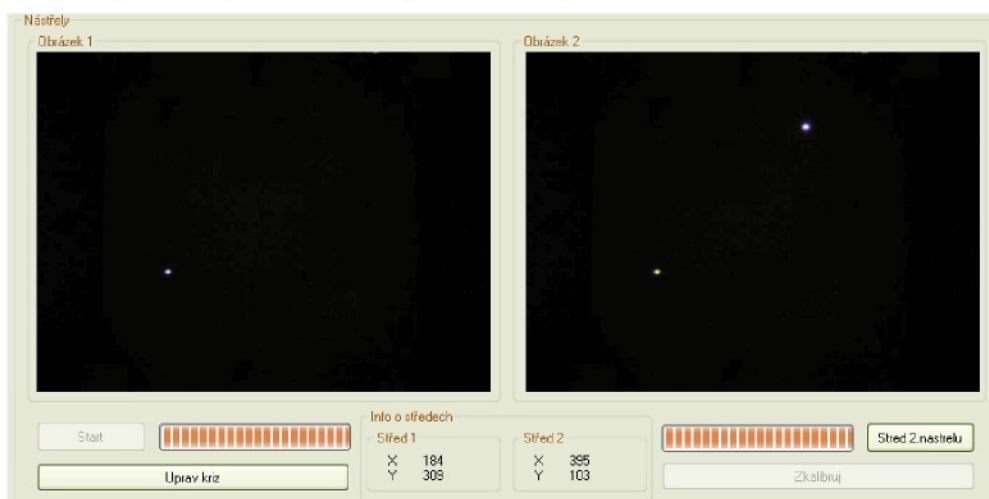
3.1.4 Pořízení snímků nástřelných

Tuto funkci obsluhuje především snímací buňka. Jako tato buňka slouží bezdrátová IP-kamera Edimax IC-1500WG, která je umístěna bezprostředně za terčem ve vzdálenosti

50ti centimetrů ve světlo neprodyšném prostoru. Tato vzdálenost zaručuje snímání celého použitého terče. Po každém výstřelu je tato kamera aktivována, poté zaznamená obraz terče, který následně pošle po přenosném mediu (v tomto případě práce skrze wi-fi síť) do příslušného adresáře v PC, do kterého má aplikace přístup a odkud lze tento pořízený snímek načíst do aplikace.

3.1.5 Vyhodnocení snímků nástřelných

Jelikož tato aplikace a vůbec celá práce slouží k přesnému nastavení kříže puškohledu, je třeba rozeznat na již pořízených snímcích vstup kulky do terče. Toto vyhodnocování se provádí pomocí aplikování dvou „for-cyklů“ na příslušný snímek. Tato část kódu zajistí, že o každém pixelu na snímku je rozhodne, zda-li se zařadí mezi body vstupu kulky (tyto body se vyhodnocují podle barvy a do paměti se ukládají jen souřadnice X,Y), nebo se tento pixel bude ignorovat a takto se to celé opakuje až do posledního pixelu ve snímku. Po vyhodnocení posledního pixelu si aplikace opět přečte v paměti uložené hodnoty souřadnic X, sečte je a vydělí je celkovým počtem uložených souřadnic X. To samé platí pro souřadnice Y. Tímto způsobem jsou získány pouze jedny souřadnice X,Y, o kterých lze prohlásit, že se jedná o střed vstupu kulky do terče. Zde se nabízí otázka: „co když by nastal při focení nějaký šum ve snímku a zobrazily by se tam i jiné pixely, o kterých by se rozhodlo, že tam byl vstup kulky do terče?“ Odpověď je velmi jednoduchá: pravděpodobnost tohoto jevu je téměř zanedbatelná, a to díky řešení snímací buňky, protože terč je focen ze tmy do světla, tím pádem se eliminují ostatní světelné body, které nemohou být jinde, než kde prošla kulka skrze terč. Tyto snímky z terče jsou zobrazeny na následujícím obr.3.1.5



(Obr. 3.1.5) – Vyhodnocování nástřelu

3.1.6 Posunutí kříže

V této části se již přistupuje k samostatnému hardwaru. Proto je nejprve nutné zablokovat nebo-li otevřít pro program důležitý COM port a nastavit jeho parametry pro komunikaci s HW. Prvotním předpokladem je, že toto zařízení bude mobilní nebo spíše řečeno toto zařízení „je“ mobilní, takže nejpravděpodobněji bude tato aplikace spouštěna na noteboocích, které bohužel ve většině případech postrádají příslušný sériový port. Proto je tento problém řešen skrze redukci z USB na RS-232C, kde je využíváno tři pinů a to GND, TxD a RxD, které jsou připojeny k převodníku na TTL úrovni MAX232.



(Obr. 3.1.6) – Redukce USB - RS-232C

Následně je načtena hodnota vzdálenosti zbraň-terč a je přepočítána hodnota u pohybu kříže o jeden zoubek na nastavovacím šroubu puškohledu.

Posunutí kříže lze provádět pouze tehdy, pokud byly v předchozím kroku (vyhodnocení snímku nástřelných) získány souřadnice středu zásahu X,Y. Pokud ano, program si určí v jakém kvadrantu se souřadnice nachází a rozhodne se, podle jakého algoritmu se má pokračovat. To už ovlivní obsluha programu. O jaké algoritmy se jedná, bude popsáno v další kapitole této práce

3.1.7 Kalibrace

Kalibrace je poslední krok v programu. Za předpokladu že je již znám první střed nástřelu (vstupu kulky v terči), souřadnice středu terče nebo první posun vertikálně i horizontálně a střed druhého nástřelu (záleží opět na typu algoritmu), pak si program vyhodnotí z těchto hodnot posun poslední a to takový, že při dalším provedení výstřelu, by měl být zásah proveden přímo do středu terče nebo by se měl minimálně provést do středu s drobnou tolerancí.

3.1.8 Paměť aplikace

Při vývoji aplikace nastával problém při rozhodování, kolik paměti v počítači se má pro program resp. pro algoritmus, který vyhodnocuje jednotlivé nástřely, vyhradit. Nejdříve byla paměť staticky nadefinována, a to polem o 10000 prvcích na každou souřadnici. Po ladění programu a různorodosti provedených nástřelů bylo zjištěno, že vyhrazená paměť pro ukládání dat byla zbytečně moc velká, anebo že nevystačila. Zjišťování optimální velikosti by bylo zbytečné, když nám programovací jazyk C# umožňuje užití dynamické alokace paměti. Této vlastnosti bylo využito za pomoci funkce `List<>`. Tato funkce si nezabírá předem žádnou paměť, ale alokuje ji podle toho, kolik je třeba uložit prvků – v tomto případě počet souřadnic.

3.2 Komunikační protokol

Řízení jednotlivých krokových motorků probíhá skrze budič SMCU05-P. Tento budič zajišťuje pohyb krokových motorků v levém či pravém směru. Jelikož tato práce využívá dvou krokových motorků pro vertikální a horizontální posuny, je třeba zajistit, zda-li se má pohnout jedním, druhým nebo oběma motorky najednou, a to ve stejných směrech nebo v různých směrech. Proto bylo třeba navrhnout nějaký jednoduchý a přesto spolehlivý komunikační protokol.

Pro charakter této práce nebyl kladen žádný důraz na šifrování či kódování jednotlivých bytů, aby nebyly zneužity třetí stranou – za prvé je zcela nemožné zneužití, jelikož se jedná o přenos skrze drátové medium a za druhé, i kdyby se připustila možnost nějakého odposlechu, tak by stejně byla data dotyčným k ničemu.

Z řídicí aplikace je posíláno po sériové lince vždy po dvou bytech. Každý byte obsluhuje jeden budič krokových motorků. Tyto byty mají, jak vyplývá již z výše zmíněného protokolu, svůj standard. Tím je myšleno, že každý bit v tomto bytu má svůj nezbytně důležitý úkol. Složení a význam bitů lze vidět u obr.3.2 kde:

- M - jaký motor se má použít
 - „0“ – horizontální motorek
 - „1“ – vertikální motorek

- S - směr otáčení
 - „0“ – směr doleva resp. směr nahoru
 - „1“ – směr doprava resp. směr dolů
- X - binární číslo, které udává počet kroků každého motorku
 - $10011_b = 19_{dec} \Rightarrow$ pohyb 19 kroků
- K – kontrolní paritní bit



(Obr. 3.2) – Byte přenosy

Na znázorněném obr.3.2 jsou vidět oba dva byty. Tyto byty musí být rozdílné u bitu označeném jako M. Toho se již využívá při rychlé kontrole, zda-li se zdařil příjem obou bytů v mikrokontroléru PIC. Při kontrole správného přenosu je to velmi slabé zabezpečení, a proto byl v každém bytu zaveden kontrolní paritní bit označený jako K. Tento paritní bit se v řídicí aplikaci nastavuje tak, aby byl počet logických jedniček vždy sudý, neboli sudá parita. Čili toto je další kontrolou v mikrokontroléru PIC, zda-li byl přenos úspěšný a že v každém bytu je po přenosu sudý počet jedniček. Pokud by po párování bytů v mikrokontroléru PIC došlo k vyhodnocení, že je počet logických jedniček lichý, tak by tento byte nemohl být akceptován a nastala by žádost o znovu-poslání bytů a opětovných kontrol zdařilého přenosu.

3.3 Algoritmus „Dohánění středu“

První myšlenkou při vývoji algoritmu, který dokáže nastavit osnovu kříže do potřebné polohy, bylo využití co nejvíce známých veličin potřebných k tomuto úkonu. Z uvedených hodnot od výrobců, pokud jej tedy každý výrobce všech komponent uvede a pokud se lze na tyto hodnoty spolehnout, tak se dají posuny hrubě odhadnout. Za předpokladu, že uvedené hodnoty budou v realitě přesné a bez nějaké větší odchylky, tak s největší pravděpodobností bude výsledek naprostý.

Jak bylo uvedeno výše, je třeba znát několik veličin či faktorů. Nejzásadnější jsou ale pouze tři, který ovlivní celý proces:

1. vzdálenost na kterou nastřelujeme zbraň
2. pozici nebo-li souřadnice, kam je nastaven zaměřovací kříž před výstřelem
3. vzdálenost o kolik se pohne osnova při přecvaknutí jednoho zoubku na seřizovacích šroubech puškohledu závislém na nastřelovací vzdálenosti.

Uvedeno na obr.3.2



(Obr. 3.3) – Ukázka značení

Z uvedených hodnot je možné hned při prvním nástřelu rozhodnout, jak se má kříž upravit, aby poté zasáhl střed terče.

3.4 Algoritmus „Dopočítávání středu“

Druhotnou myšlenkou seřizování puškohledu, bylo přistoupit k samotnému procesu seřizování opačným způsobem, a to všechny hodnoty, ať jsou nebo nejsou uvedeny, si dopočítávat v průběhu nastřelování ze získaných dat. Tento algoritmus má veškerou výhodu v tom, že nemusí být žádná veličina výrobcem uvedená, dokonce může být uvedena špatně a přesto to nebude mít vliv na přesnost nastavení. Veškerý trik spočívá v inteligentním algoritmu, který sice pro nastavení osnovy do příslušné pozice spotřebuje v průměru o jeden náboj více, ale o to je výsledek přesnější, než u předchozího řešení.

V první fázi algoritmu se vyhodnotí první nástřelná rána. Pokud se nachází v nějakém viditelném kvadrantu, provede se úprava osnovy v puškohledu tak, aby se další nástřelný výstřel objevil v opačném kvadrantu, a to křížem. Čili pokud se první střela nachází v 1. kvadrantu, tak předpoklad je dostat další střelu do kvadrantu č.3. Následně je odvozen z předešlých dvou nástřelů druhý pohyb osnovou, který by měl být již zkalibrován na zásah do středu terče.

4. Testování

Po zvládnutí předešlé teorie z hlediska elektroniky a programování, bylo sestaveno kompletní a plně funkční zařízení, které je schopno zkalibrovat puškohled na střelné zbrani, na víceméně jakoukoli vzdálenost. Poslední otázkou celé práce je, jak toto zařízení pracuje v reálných podmínkách? Proto se tato část bude zabývat jednotlivými testy, které prozatím proběhly na zařízení. Pro přehlednost a lepší pochopení testů je uvedeno několik tabulek s grafy, kde jsou uvedeny jednotlivé výsledky.

4.1 Testování za pomoci nezkušených uživatelů

V první fázi testů bylo třeba udělat menší průzkum, jak si obyčejní uživatelé vedou při seřizování puškohledu bez jakýchkoli pomůcek (kromě nezbytně nutných). Proto bylo vybráno nezávisle na sobě 20 osob, které nemají moc velké zkušenosti s touto procedurou. Po řádném proškolení se zacházení se střelnou zbraní a pod dohledem zkušeného instruktora bylo každému umožněno seřadit si puškohled na zbrani, a to manuálně na místech k tomu určených, čili na střelnici. Jako použitá střelná zbraň, byla použita malorážka „ČZ 451“ s ráží .22LR. Výsledky si lze prohlédnout v tabulce 4.1.

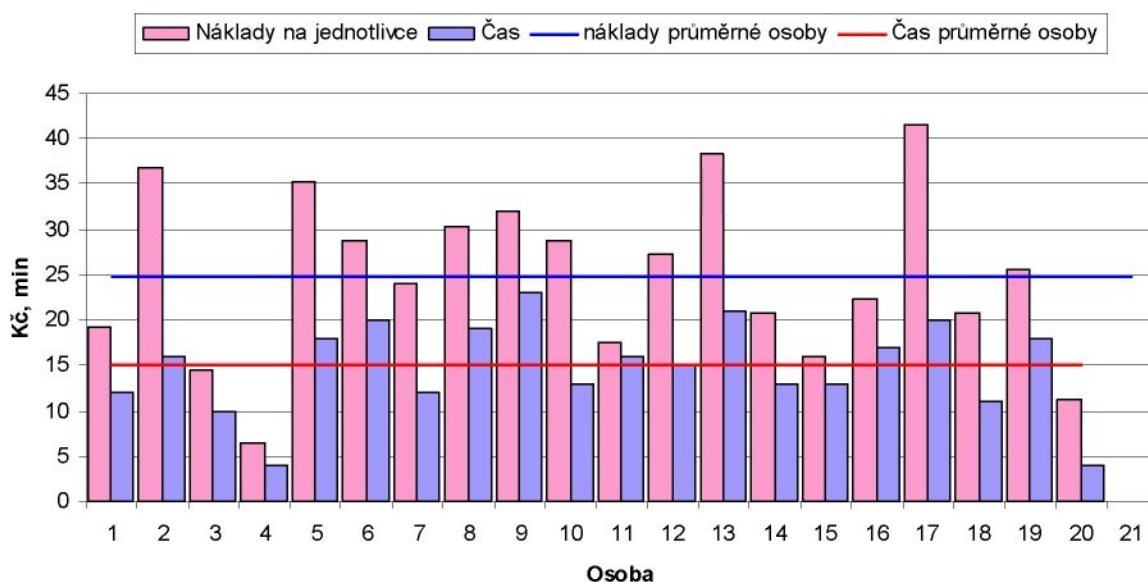
Malorážka ČZ 451 .22LR			
Osoba	Počet nábojů [Ks]	Náklady [Kč]	Čas [min:s]
1	12	19,2	12:44
2	23	36,8	16:12
3	9	14,4	10:57
4	4	6,4	4:02
5	22	35,2	18:26
6	18	28,8	20:55
7	15	24	12:09
8	19	30,4	19:48
9	20	32	23:13
10	18	28,8	13:56
11	11	17,6	16:23
12	17	27,2	15:31
13	24	38,4	21:15
14	13	20,8	13:02
15	10	16	13:24
16	14	22,4	17:21
17	26	41,6	20:03
18	13	20,8	11:51
19	16	25,6	18:32
20	7	11,2	4:40

(Tab. 4.1) – výsledky nezkušených uživatelů

U tohoto testu bylo u každé osoby počítáno, kolik musel provést nástřelných ran, než se dostal na soustřel záměrné osnovy v puškohledu a dopadu střely v terči. Na základě této hodnoty se daly spočítat potřebné náklady pro jednotlivce, když známe cenu náboje (použité střelivo 1,60 Kč / ks). Tento celý proces byl i časově zaznamenáván, takže můžeme z těchto tří hodnot určit průměrnou osobu (z důvodu objektivnosti testu).

Průměrná osoba s minimálními zkušenostmi dosahuje těchto hodnot:

- Počet nábojů = 15,55 Ks
- Náklady = 24,88 Kč
- Čas = cca 15 min



(Graf. 4.1) – výsledky nezkušených uživatelů

Na uvedeném grafu 4.1, jdou vidět všechny dosažené výsledky s touto skupinou lidí včetně obecného výsledku s průměrnou osobou. Za zmínku stojí povšimnutí si 17. osoby s nejhorší variantou nákladů, kde se cena vyšplhala na téměř 42 Kč. Není to nijak zvlášť závratná cena, ale proč je zde uvedena, bude vysvětleno ke konci testu.

4.2 Testování za pomoci zkušených uživatelů

V druhé fázi bylo třeba zjistit, jak se liší tato skupina lidí od těch předchozích. Pro dobré porovnání bylo opět vybráno 20 lidí, kteří již měli nějaké drobné zkušenosti v této problematice. Tato skupina lidí měla naprosto stejné podmínky, totiž že nesměla použít jiných

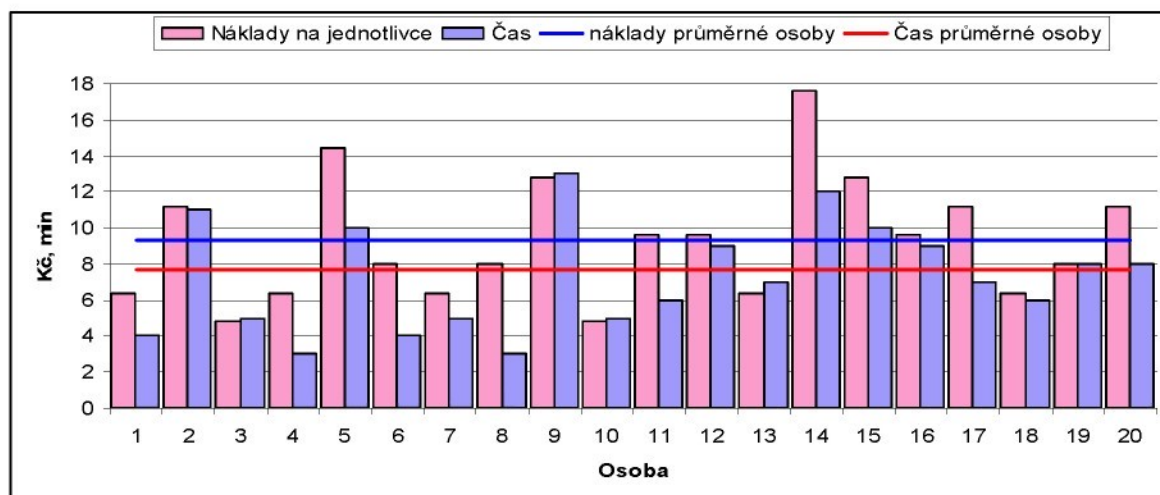
prostředků, než nezbytně nutných. Testování proběhlo na stejné zbrani „ČZ 451“ se střelivem z jedné a té samé série. Výsledky si lze prohlédnout v tabulce 4.2

Malorážka ČZ 451 .22LR			
Osoba	Počet nábojů [Ks]	Náklady [Kč]	Čas [min:s]
1	4	6,4	4:16
2	7	11,2	11:23
3	3	4,8	5:13
4	4	6,4	3:42
5	9	14,4	10:31
6	5	8	4:16
7	4	6,4	5:28
8	5	8	3:06
9	8	12,8	13:12
10	3	4,8	5:47
11	6	9,6	6:26
12	6	9,6	9:00
13	4	6,4	7:36
14	11	17,6	12:11
15	8	12,8	10:34
16	6	9,6	9:23
17	7	11,2	7:08
18	4	6,4	6:52
19	5	8	8:57
20	7	11,2	6:12

(Tab. 4.2) – výsledky zkušených uživatelů

Z těchto výsledků u druhé skupiny lze opět odvodit jedna průměrná osoba, která by měla mít takovéto výsledky:

- Počet nábojů = 5,8 Ks
- Náklady = 9,28 Kč
- Čas = cca 7 min



(Graf 4.2) – výsledky zkušených uživatelů

Na dalším grafu (graf 4.2) je opět celkový pohled na dosažené výsledky, ale té druhé, zkušenější skupiny, včetně průměrné osoby. Opět si všimněme, tentokrát 14. osoby, která má náklady na seřízení téměř 18 Kč.

4.3 Testování za pomoci sestaveného systému

Nejdůležitějším testováním celé diplomové práce bylo otestování sestrojeného a naprogramovaného hardwarového celku. U tohoto zařízení mělo největší vliv, jaký byl použit algoritmus k seřízení.

Při použití algoritmu „dohánění středu“, který byl popsán v kapitole 3.3 byly výsledky občas nepřesné. Tím se potvrdil předpoklad, že buď výrobce sice uvádí důležité hodnoty, které jsou nepatrně zkresleny nebo se jednalo o chybu nepřesností u měřicích prvků. Není vyloučena chybovost obou věcí zároveň a ani kvalita zpracování puškohledu. Z předpokládaného počtu 2 kusů nábojů i s kontrolní střelou byl malinko mylný, jelikož občas bylo za potřeby 3, ale i dokonce 4 kusů ke kalibraci.

Oproti tomu výsledky při použití algoritmu „dohánění středu“ (kapitola 3.4) byly podstatně rozumnější a přesnější. Jelikož si všechny hodnoty nastřelovací systém téměř určuje sám, tak je mnohem flexibilnější k určitým chybám. Předpokládaný počet 3 kusů nábojnic ke kalibraci, (samozřejmě je v tom i započítaná kontrolní střela), se téměř vždy shodoval s realitou. Velmi zřídka objevená chyba, která je ale velmi zanedbatelná, kde bylo zapotřebí 4 kusů nábojnic, je spíše připisována například špatnou náplní střelného prachu, znečištění hlavně atd.

Algoritmus	Průměr počtu nábojnic [Ks]	Náklady [Kč]	Čas [min:s]
Dopočítávání středu	3,2	5,12	1:10
Dohánění středu	3	4,8	0:52

(Tab. 4.3) – výsledky nastřelovacího systému

4.4 Výsledek testování

Na začátku testování bylo řečeno u každé skupiny lidí, že nějaká osoba stojí za povšimnutí. Nyní je třeba se k tomuto tématu vrátit. V následující tabulce 4.4.1 jsou uvedeny

dvě osoby z každé skupiny s nejvyššími náklady na seřízení, obě dvě průměrné osoby a lepší výsledek z testování systému.

		Počet nábojů [Ks]	Náklady [Kč]	Čas [min:s]
1. skupina	Osoba 17	26	41,6	20:03
1. skupina	Průměrná osoba	15,55	24,88	15:00
2. skupina	Osoba 14	11	17,6	12:11
2. skupina	Průměrná osoba	5,8	9,28	7:00
Systém	Dohánění středu	3	4,8	0:52

(Tab. 4.4.1) – Porovnání výsledků 1

U této tabulky 4.4.1, je vidět, jak se náklady na nastřelování snižují v závislosti na zkušenostech u lidí, ale především je vidět výrazný rozdíl nákladů za použití systému. Náklady jsou téměř poloviční, oproti průměrně zkušené osoby. Zde je finanční rozdíl minimální, ale zde se jedná hlavně o otázku, jaký by byl důsledek, kdyby se nepoužila slabá kulovnička ale opravdu výkonná kulovnice s kvalitním střelivem? Pro ilustraci bylo vybráno střelivo „.300 Winchester Matným 180 Swift A-frame“, které se podle dnešních cen pohybuje kolem 102 Kč / ks, jak je uvedeno v [www 7]. Zbytek zachováváme se stejnými hodnotami, jelikož postupy jsou naprosto stejné, jen se změnil typ zbraně a střeliva.

		Počet nábojů [Ks]	Náklady [Kč]	Čas [min:s]
1. skupina	Osoba 17	26	2652	20:03
1. skupina	Průměrná osoba	15,55	1586,1	15:00
2. skupina	Osoba 14	11	1122	12:11
2. skupina	Průměrná osoba	5,8	591,6	7:00
Systém	Dohánění středu	3	306	0:52

(Tab. 4.4.2) – Porovnání výsledků 2

Na předchozí tabulce 4.4.2 jsou finanční rozdíly už velmi podstatné, a to řádově ve stovkách korun. V první skupině lidí to je, v nejhorším případě, ušetření celých 2346 Kč a průměrně 1280,1 Kč. Ve druhé skupině ušetření nákladů bude menší, ale stále nezanedbatelné. V nejhorším případě ušetření nákladů činí 816 Kč a průměrně 285,6 Kč. Co se týče doby strávené seřizováním, tak je z tabulky jasné, že u použití systému je čas, oproti manuálnímu seřízení, výrazně zredukován na minimum.

4.5 Návratnost

Jelikož se jedná o produkt, který šetří náklady, jak bylo uvedeno v předchozí kapitole, tak by bylo dobré uvést, za jak dlouho se navrátí investice do tohoto zařízení.

Celkové náklady:

Zařízení	Cena [Kč]
IP-kamera	1200
Krokové motory	300
Drobná elektronika	200
Zdroj	1200
Budič KM	450
Redukce USB -RS-232	300
Ostatní	500
Suma	4150

(Tab. 4.5) – Náklady na produkt

Celkové náklady na sestavení celého zařízení činily 4150 Kč. V této ceně není zahrnutý čas strávený programováním aplikace a ani vedlejší náklady za věci, které byly při vývoji použity. Celková cena je jen za finální produkt. Jednotlivé ceny za položky jsou zaokrouhleny.

Ve fázi testování jsme docílili jistých výsledků. Pro uzavření otázky z hlediska návratnosti, je třeba mít na paměti, jaké náboje jsou užívány. Jinak platí přímá úměra, čím dražší náboj, tím rychlejší bude návratnost. Pro ilustraci je porovnání s průměrnou osobou 1. skupiny za předpokladu, že náboj stojí oněch 102 Kč, pak návratnost je zhruba při 4. seřazené zbrani.

5. Závěr

V této práci bylo popsáno vše kolem elektronického seřizování puškohledů, od principů každé z komponent, až po realizaci celého zařízení.

Nejdříve, než bylo přistoupeno k samotné problematice této práce, bylo vysvětleno několik důvodů, proč práce tohoto charakteru vlastně vznikla a proč by bylo vhodné v ní i nadále pokračovat.

Druhá část se již zabývá jistými součástkami, komunikací mezi nimi a dalšími tzv. „sety“, jako je například budič krokových motorků, snímací buňka atd., které jsou velmi důležité pro chod celého systému.

Ve třetí části byla popsána řídicí aplikace, která obsluhuje celý nastřelovací systém. Tento program komunikuje s vedlejšími jednotkami, a to se snímací buňkou a mikrokontrolérem PIC, který se už stará o samostatný hardware.

Po složení a oživení celého systému bylo přistoupeno k samostatnému testování na dvou skupinách lidí versus tento systém. Výsledek dopadl dle předpokladů, že člověk nenastaví nic tak rychle a přesně jako počítač. Výsledkem pak bylo snížení nákladů pro koncové uživatele, kteří se zabývají nebo by chtěli zabývat touto problematikou.

V této práci bylo třeba vyvinout dva programy. Hlavní řídicí aplikaci a program, který je nahrán do mikrokontroléru PIC. Tyto programy byly naprogramovány ve dvou různých programovacích jazycích, avšak jsou si velmi podobné (vychází ze stejného standardu).

- Řídicí aplikace – naprogramováno v jazyce C#
- Program v mikrokontroléru PIC – naprogramováno v jazyce C

Seznam použitých zdrojů

- [www 1] Pavel Tišnovský, Sériový port RS-232C [online]. 2008-11-27, Dostupné z www: <<http://www.root.cz/clanky/seriovy-port-rs-232c/>>, [cit. 2010-05-20]
- [www 2] Wikipedie: otevřená encyklopedie, RS-232 [online]. Dostupné z www: <<http://cs.wikipedia.org/wiki/RS-232>>, [cit. 2010-5-20]
- [www 3] Vít Olmr, Sériová linka RS-232 [online]. 2005-12-12, Dostupné z www: <<http://hw.cz/rs-232#parametry>>, [cit. 2010-05-20]
- [www 4] Kamil Řezáč, Krokové motory [online]. 2002-10-28, Dostupné z www: <<http://robotika.cz/articles/steppers/cs>>, [cit. 2010-05-20]
- [www 5] Ing. Pavel Rydlo, Krokové motory a jejich řízení [online]. 2000, Dostupné z www: <www.mti.tul.cz/files/ats/krok2.pdf>, [cit. 2010-05-20]
- [www 6] ASIX, datasheet PVK40 [online]. 2009-11-27, Dostupné z www: <http://www.asix.cz/download/pvk40/pvk40_cz.pdf>, [cit. 2010-05-20]
- [www 7] Gunshop, NORMA střelivo [online]. Dostupné z www: <<http://shop.vmcustom.cz/data/strelivo/norma.pdf>>, [cit. 2010-05-20]
- [www 8] Microchip, PICkit 3 Debug Express [online]. 2009, Dostupné z www: <http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en538340>, [cit. 2010-05-20]
- [www 9] ASIX, PRESTO – rychlý USB programátor [online]. 2010-4-16, Dostupné z www: <http://www.asix.cz/a6_presto.htm>, [cit. 2010-05-20]
- [www 10] Microchip, MPLAB REAL ICE In-Circuit Emulator, 2009, Dostupné z www: <http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en028120>
- [www 11] Microchip, datasheet PIC16F627A, 2009, Dostupné z www: <<http://ww1.microchip.com/downloads/en/DeviceDoc/40044G.pdf>>, [cit. 2010-05-20]
- [www 12] Wikipedie: otevřená encyklopedie, Mikrokontrolér PIC [online]. Dostupné z www: <http://cs.wikipedia.org/wiki/Mikrokontrol%C3%A9r_PIC> [cit. 2010-05-20]
- [www 13] Silicobrain, datasheet SMCU05-P, Dostupné z www: <<http://www.siliconbrain.biz/produkty/pdf/SMCU05-P.pdf>>, [cit. 2010-05-20]
- [www 14] MAX232 [online], 2010, Dostupné z www: <<http://www.d4atasheetcatalog.org/datasheet/texasinstruments/max232.pdf>> [cit. 2010-05-20]
- [15] VACEK, Václav.: Učebnice programování PIC. Praha : BEN, 2002. ISBN 80-86056-87-2

[16] MAŤÁTKO, Jan.: Elektronika. 5. vydání. Praha : IDEA Servis, 2000.
ISBN 80-85970-42-2

[17] BALÁTĚ, Jaroslav.: Automatické řízení., 2. aktualizované vydání, BEN,
ISBN 978-80-7300148-3

[18] VIRIUS, Miroslav.: C# pro zelenáče., Neocortex, ISBN 80-86330-11-7

Seznam obrázků

1. (Obr. 2.1) – Mikrokontrolér PIC16F627A
2. (Obr. 2.1.4) – Programátor
3. (Obr. 2.2.1) – Řez čtyřfázovým krokovým motorkem
4. (Obr. 2.3) – Budič krokového motorku
5. (Obr. 2.3.3) – Připojení krokového motorku k budiči
6. (Obr. 2.3.4) – Rozložení konektorů a součástek na budiči KM
7. (Obr. 2.4.2) – Přenos dat po RS-232C
8. (Obr. 2.4.2) – Přenos dat po RS-232C
9. (Obr. 2.5.1) – Zapojení řídicího HW
10. (Obr. 2.5.2) – Testovací zapojení řídicího HW
11. (Obr. 2.6.1) – Ukázka snímaného obrazu skrze puškohled
12. (Obr. 2.6.2) – Snímací buňka
13. (Obr. 3.1.1) – Vzhled aplikace
14. (Obr. 3.1.2.1) – Menu
15. (Obr. 3.1.2.2) – Informace o středech
16. (Obr. 3.1.2.3) – Smazat
17. (Obr. 3.1.2.4) – Nastavení
18. (Obr. 3.1.2.5) – Rozlišení snímku terče
19. (Obr. 3.1.2.6) – Nastavení snímací buňky
20. (Obr. 3.1.2.7) – Postup řešení
21. (Obr. 3.1.2.8) – Míry
22. (Obr. 3.1.2.9) – Help
23. (Obr. 3.1.3) – Nastavení vzdáleností
24. (Obr. 3.1.5) – Vyhodnocování nástřelu
25. (Obr. 3.1.6) – Redukce USB - RS-232C
26. (Obr. 3.2) – Byte přenosy
27. (Obr. 3.3) – Ukázka značení

Seznam tabulek

1. (Tab. 4.1) – výsledky nezkušených uživatelů
2. (Tab. 4.2) – výsledky zkušených uživatelů
3. (Tab. 4.3) – výsledky nastřelovacího systému
4. (Tab. 4.4.1) – porovnání výsledků 1
5. (Tab. 4.4.2) – porovnání výsledků 2
6. (Tab. 4.5) – náklady na produkt

Seznam grafů

1. (Graf 4.1) – výsledky nezkušených uživatelů
2. (Graf 4.2) – výsledky zkušených uživatelů

Seznam použitých zkratek

1. CAN	Controller Area Network
2. COM	communication port
3. CS	Clear to Send
4. DCD	Data Carrier Detect
5. DSR	Data Set Ready
6. DTR	Data Terminal Ready
7. EEPROM	Electrically Erasable Programmable Read-Only Memory
8. FTP	File Transfer Protokol
9. GND	Ground
10. ICSP	In-Circuit Serial Programming
11. JPG	Joint Photographic Group
12. KM	krokový motor
13. LED	Light-Emitting Diode
14. MAX232	převodník napětových úrovní 0 - 5V
15. PC	Personal Computer
16. PH1 - 4	Phase
17. PIC	Programmable Interface Controller
18. PX	Pixel
19. RI	Ring Indikator
20. RISC	Reduced Instruction Set Computer
21. RS232	sériová linka
22. RTS	Request to Send
23. RxD	Received Data
24. SG	Signal Ground
25. TTL	Transistor-Transistor logic
26. TxD	Transmitted Data
27. UART	Universal Synchronous Asynchronous Receiver Transmitter
28. USART	Addressable Universal Synchr. Asynchr. Receiver Transmitter
29. USB	Universal Serial Bus
30. WI-FI	bezdrátové rozhraní
31. YD	Yard

Seznam příloh

Tato práce obsahuje jako přílohu CD, jehož obsahem je:

- vlastní text práce (ve formátu PDF)
- resumé (ve formátu DOC)
- zdrojový kód mikrokontroléru PIC (jazyk C)
- Zdrojový projekt řídicí aplikace (jazyk C#)

RESUMÉ

Česky

Tato práce seznamuje s problematikou elektronického seřizování puškohledů, která je rozdělena do pěti samostatných bloků. Náplní první části je kratičký úvod o problematice a proč je tato práce realizována. V druhé části je proniknutí do problematiky z hlediska seznámení s teorií jednotlivých komponent a součástí, které jsou použity při navrhování a realizaci funkčního systému, který bude ovládat samostatnou činnost seřizování. V této části je popsán i způsob řešení komunikace mezi jednotlivými hardwarovými celky. Samozřejmostí, která úzce souvisí s hardware je doplnění o řídicí software, pro který je vyhrazena třetí kapitola. Ve čtvrté části jsou popsány dosažené výsledky z průběhu testování. Závěrečná kapitola je poslední částí, kde je popsáno několik průběžných poznatků z celé práce a zda-li se naplnily plánované cíle.

Klíčová slova: Nastřelování, seřizování, puškohled, mikrokontrolér, motor

English

This thesis introduces the problems of electrical tuning scopes, which are divided into five individual blocks. The contents of the first part is short introduction about the problems and why is this job realized. In the second part is the insight into the problem in terms of acquaintance all of the components, which are used in designing and implementing a functional system, which will control the adjustment of individual activity. In this section there is described a way of communication of individual hardware equipment. Obviousness, which is closely related to hardware, is the addition of management software, which is reserved for the third chapter. Fourth section describes the results of the testing. Final section is the last part, where are described some intermediate knowledges of the whole work and if all the targets were realized

Keywords: Driving, adjusting, riflescopes, microcontroller, motor

