



Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Diplomová práce**

# **Polohovací jednotka pro mikroskop**

# Zadání Diplomové práce

## **Prohlášení**

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne.....

.....

**Bc. Tomáš Pohanka**

## **Poděkování**

Rád bych tímto způsobem poděkoval všem, kteří mi jakýmkoliv způsobem pomohli při tvorbě diplomové práce. Zvláště bych chtěl poděkovat vedoucímu práce Dr. Ing. Karlu Dudáčkovi za odborné vedení, vstřícnost, ochotu, cenné rady a podněty.

# **Abstrakt**

## **Positioning unit for microscope**

The aim of this Diploma Thesis is to design and create hardware and software for positioning unit for microscope. This unit will be fixed to manual locating mechanism and will control motors of automatic drive. Theoretic selection of actuators, microcontroller and other integrated circuits are described in this thesis. The practical part includes proposal of communication between microcontroller and personal computer, proposal of PCB design, firmware for microcontroller and user interface for personal computer. For the subject of investigation a test plate which served for testing correct functioning of the whole unit was created.

## **Key words**

Positioning unit. Microscope, microcontroller, PCB design, actuating devices

## Obsah

<b>1. Úvod</b> .....	<b>11</b>
<b>2. Rozbor řešení elektromechanických polohovacích mechanismů</b> .....	<b>12</b>
2.1. Krokové motory .....	13
2.2. Stejnoseměrný motor .....	13
<b>3. Volba vhodné koncepce polohovací jednotky, určené pro automatické sekvenční snímání mikroskopických vzorků</b> .....	<b>13</b>
3.1. Požadované funkce .....	14
3.2. Výběr vhodných integrovaných obvodů .....	14
3.2.1. Mikrokontrolér NEC V850E/IG3 .....	15
3.2.1.1. Čítače a časovače .....	16
3.2.1.2. A/D převodník .....	17
3.2.1.3. Debug Interface .....	17
3.2.1.4. UART .....	17
3.2.1.5. DMA .....	18
3.2.1.6. Přerušení .....	18
3.2.1.7. Obvody pro řízení motorů .....	19
3.2.2. IO obvody pro komunikace .....	21
3.2.3. Motory, převodovka, enkodér .....	22
3.2.3.1. Ideální kombinace pro realizaci .....	22
3.2.3.2. Cenově dostupná varianta .....	25
3.2.3.3. Testovací varianta .....	26
3.2.4. Světelné závory .....	27
3.2.5. Přizpůsobovací členy .....	28
<b>4. Návrh řídicí elektroniky včetně firmware</b> .....	<b>29</b>
4.1. Návrh komunikace .....	29
4.2. Návrh schématu a plošného spoje .....	30
4.3. Návrh firmware pro mikrokontrolér NEC V850E/IG3 .....	35
4.4. Návrh uživatelského rozhraní pro osobní počítač .....	45
4.5. Komunikace osobní počítač ← → mikrokontrolér .....	48
<b>5. Realizace a ověření praktické použitelnosti</b> .....	<b>50</b>
<b>6. Závěr</b> .....	<b>53</b>
Literatura .....	54

## Seznam obrázků

Obr. 2.1: Křížový stolek pro mikroskop [1] .....	12
Obr. 2.2: Blokové schéma polohovací jednotky .....	14
Obr. 3.1: Blokové schéma NEC V850E/IG3 [3] .....	15
Obr. 3.2: Patice L293 [4] .....	19
Obr. 3.3: Výstup integrovaného obvodu L293D [4].....	20
Obr. 3.4: Připojení motoru k 1/2 L293D [4].....	20
Obr. 3.5: Zapojení patice MAX232 [5] .....	22
Obr. 3.6: Motor Maxon RE 16 [6] .....	23
Obr. 3.7: Čelní a planetární převodovka Maxon GP 16 A [7].....	23
Obr. 3.8: Enkodér Maxon MR a popsání pinů konektoru [8].....	23
Obr. 3.9: Časový průběh výstupu enkodéru Maxon MR .....	24
Obr. 3.10: Celková délka kompletu .....	24
Obr. 3.11: Motor F-GM12-N20VA110 [9] .....	26
Obr. 3.12: Motor F-GM12-N20VA110 s enkodérem [10] .....	26
Obr. 3.13: Motor s převodovkou GHM-04 [11] .....	26
Obr. 3.14: Celý testovací mechanismus.....	27
Obr. 3.15: Umístění světelných závor na pracovní stolek .....	27
Obr. 3.16: Vnitřní zapojení světelné závoře [12] .....	28
Obr. 3.17: Vnitřní zapojení 74HC14N [13].....	28
Obr. 3.18: Blokové schéma polohovací jednotky s označením bloků.....	29
Obr. 4.1: Navržená komunikace .....	30
Obr. 4.2: Krabička U-KM 35 [17] .....	32
Obr. 4.3: Rozvod napájení a GND na desce .....	34
Obr. 4.4: Blokové schéma časovače TABn [1] .....	36
Obr. 4.5: Blokové schéma časovače TMTn [1].....	39
Obr. 4.6: Uživatelské rozhraní.....	46
Obr. 4.7: Posuvy při sekvenčním snímání .....	47
Obr. 4.8: Ukázka komunikace MCU a uživatelského rozhraní .....	50
Obr. 5.1: Deska osazená MCU NEC V850E/IG3.....	50
Obr. 5.2: Testovací deska .....	51



## **Seznam tabulek**

Tab. 3.1: Funkce pinů .....	19
Tab. 3.2: Úrovně vstupních signálů pro L293D [4].....	21
Tab. 3.3: Ceny položek Maxon.....	25
Tab. 4.1: Rozdělení registrů.....	38
Tab. 4.2: Nastavení vhodného čítání pulzů v závislosti na vstupech [1].....	40

## Seznam zkratk

PCB	- Printed circuit board
LVI	- Low voltage input
LED	- Light-emitting diode
MTX	- Multiplexor
MCU	- Mikrokontrolér
USB	- Universal Serial Bus
A/D	- Analogovo-digitální
RISC	- Mikroprocesor s redukovanou instrukční sadou
LQFP	- Low-profile quad flat package
PWM	- Pulse-width modulation (pulzní šířková modulace)
CCR	- Capture/ Compare register
DCU	- Debug control unit
IO	- Integrovaný obvod
IČ	- Inkrementální čidlo
V/V	- Vstupně / výstupní
SVZ1(2)	- Světelná závora 1 (2)
MSB	- Most significant bit (byte)
LSB	- Least significant bit (byte)
bps	- Bitů za sekundu

## 1. Úvod

V současné době existuje mnoho variant a způsobů, jak pohybovat se zkoumaným vzorkem po pracovním stolku mikroskopu. Používají se manuální mechanismy, kde je k pohybu vzorku potřeba činnosti člověka, nebo automatické mechanismy, kde pohyb obstarávají akční členy ovládané počítačem.

Úkolem diplomové práce bylo navrhnout automatické ovládání manuálního mechanismu pohybu, dále pak vybrat hlavní řídicí mikrokontrolér, vhodné integrované obvody, akční členy, vytvořit software pro polohovací jednotku a uživatelské rozhraní pro osobní počítač.

Podle zadání má polohovací jednotka posouvat vzorek v ose X i v ose Y, určovat jeho polohu a zajišťovat, aby nedošlo k vysunutí vzorku mimo pracovní plochu. Ovládání a kontrola činnosti jednotky se má, provádět pomocí osobního počítače.

Práce je rozdělena do čtyř částí, které popisují celkový postup práce. První část se věnuje teoretickému popisu zadaného úkolu a výběru vhodných akčních členů určených k pohybu vzorku, včetně jejich výhod i nevýhod.

Druhá část obsahuje teoretický popis, výběr jednotlivých integrovaných obvodů a mikrokontroléru na základě požadovaných funkcí, kterými má jednotka disponovat. U vybraného mikrokontroléru NEC V850E/IG3 jsou následně popsány jednotlivé jeho části a jejich funkce. Dále jsou v této kapitole popsány tři varianty akčních členů, které by byly vhodné k realizaci.

Ve třetí části je popsán návrh a realizace schématu a plošného spoje. Dále také software pro polohovací jednotku, kde je znázorněno nastavení registrů pro požadovanou funkci včetně ukázky konkrétních hodnot. U uživatelského rozhraní pro osobní počítač je popsána funkce jednotlivých částí i konkrétních tlačítek. Část této kapitoly je věnována popisu konkrétní komunikace mezi osobním počítačem a mikrokontrolérem.

V poslední části se věnují testování navrženého softwaru a hardwaru včetně komunikace mezi zařízeními.

V přílohách je navržené schéma a plošný spoj, dále zdrojový kód nahrávaný do mikrokontroléru a samozřejmě i skript uživatelského rozhraní.

## 2. Rozbor řešení elektromechanických polohovacích mechanismů

Při prvním seznámení s mikroskopem, pro který je úloha vytvořena, jsem si prohlédl, jak se provádí pohyb po pracovním stolku. Pro pohyb a odměřování v osách X a Y je použit křížový stolek (viz obrázek 2.1), který je připevněn k pracovnímu stolku.



Obr. 2.1: Křížový stolek pro mikroskop [1]

Mechanické nastavení pozice se provádí manuálně pomocí otočného knoflíku. K přechodu mezi koncovými pozicemi v ose X je potřeba 18 a v ose Y 1,5 otočení knoflíkem. Pro automatický posuv nebo sekvenční snímání bylo nutné zjistit krouticí moment, který je potřeba pro otočení knoflíku a následně vybrat vhodný elektromechanický polohovací systém. Na obou otočných knoflících je však potřeba rozdílný krouticí moment, proto budu pracovat s vyšším momentem, aby byly pro oba směry posuvu použity stejné motory a zařízení k nim připojené. Po zjištění všech potřebných informací bylo možné vybrat mezi krokovými motory a stejnosměrnými motory s víceúrovňovou převodovkou a inkrementálním čidlem. Motory musely také vyhovovat svými rozměry a svou hmotností, protože je nutné, aby byly umístěny na křížovém stolku a pohybovaly se současně s celým mechanismem pracovního stolku.

## **2.1. Krokové motory**

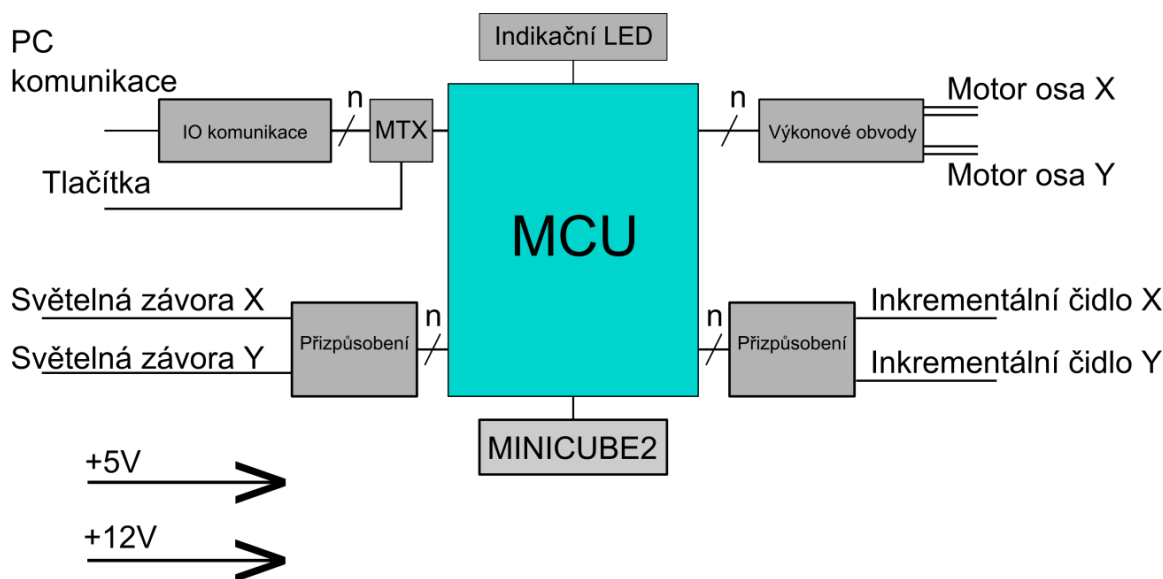
Tyto motory je vhodnější použít při realizaci polohovací jednotky pro mikroskop z důvodu jednoduchosti řízení a zjišťování polohy. Je několik variant řízení. Buď s plným, nebo polovičním krokem, různou rychlostí a směrem otáčení. Jednotlivé varianty řízení se volí podle velikosti požadovaného krouticího momentu, přesnosti nastavení polohy a přípustného odběru [2]. Následné zjištění polohy by se vypočítalo z počtu a velikosti kroků již programově. Podle zjištěné velikosti krouticího momentu by krokový motor pro potřebný krouticí moment měl příliš velké rozměry a umístění na křížový stolek nebylo možné z důvodu vyšší hmotnosti. Proto bylo vhodnější použít již stejnosměrný motor.

## **2.2. Stejnosměrný motor**

Rozměry stejnosměrných motorů s víceúhňovou převodovkou pro stejný krouticí moment jsou menší a tím i přijatelnější pro praktickou realizaci. Tento druh motorů je buzen stejnosměrným napětím, které je přiváděno na napájecí svorky motoru a jehož velikostí se nastavuje rychlost otáčení motoru. Směr otáčení motorů je možno řídit pomocí výkonových obvodů. Řízení rychlosti motorů je složitější, protože je nutné, aby zdroj mohl vysílat analogové napětí s různou velikostí, nebo aby výstupem byl PWM signál. Poté je možno řídit rychlost otáčení. Díky víceúhňové převodovce, která je součástí motoru, není nutné, aby samotný motor zvládal tak velký krouticí moment. Inkrementální čidlo vysílá pulzy, pomocí nichž je možné odečítat pozici na pracovním stolku.

## **3. Volba vhodné koncepce polohovací jednotky, určené pro automatické sekvenční snímání mikroskopických vzorků**

Po konzultaci se zadavatelem práce a zadáním přesným funkcí jednotky, jsem navrhl základní blokové schéma, viz obrázek 2.2. Na tomto obrázku je vidět uspořádání integrovaných obvodů nutných pro správnou funkčnost jednotky.



Obr. 2.2: Blokové schéma polohovací jednotky

### 3.1. Požadované funkce

- ovládání jednotlivých kroků automatického posuvu, zadání hodnot pro sekvenční snímání nebo nastavení rychlosti otáčení motorů pomocí osobního počítače přes sériovou linku
- kontrola počáteční pozice a zamezení pohybu stolečku do pozice v záporných hodnotách na obou osách
- pohyb s křížovým pracovním stolečkem v osách X a Y pomocí stejnosměrných motorů s víceúhňovou převodovkou
- snímání aktuální pozice z impulzů inkrementálního čidla připojeného k ovládanému motoru

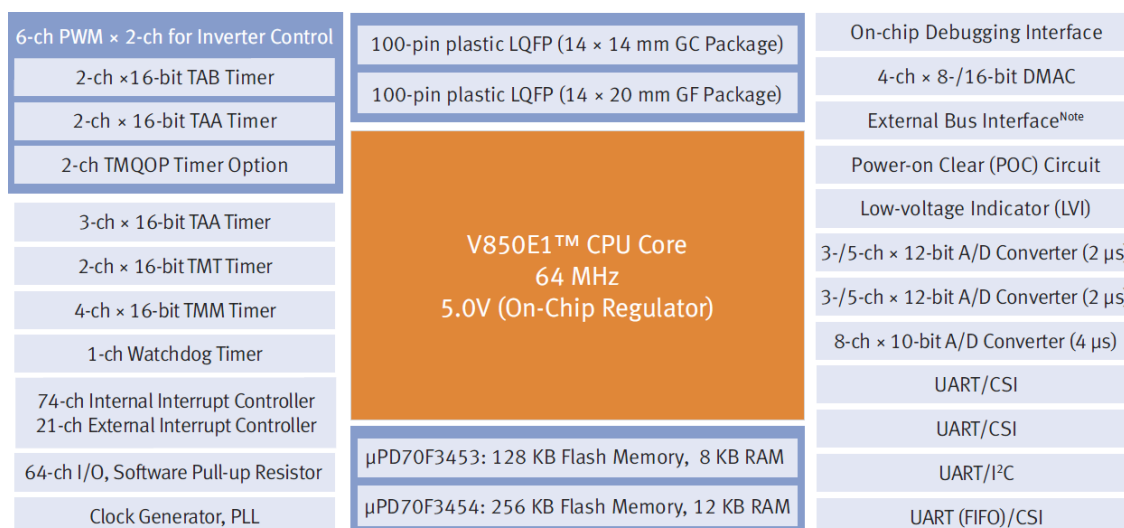
### 3.2. Výběr vhodných integrovaných obvodů

K požadovaným funkcím bylo nutné vybrat vhodné integrované obvody, které by zaručily správnou funkčnost jednotky. V následujících kapitolách jsou popsány integrované obvody vybrané pro výslednou realizaci. Nejdříve bylo nutné zvolit hlavní MCU (mikrokontrolér). Rozhodoval jsem mezi MSP430 od firmy Texas Instruments a NEC V850E od firmy Renesas. Oba tyto mikrokontroléry jsou vhodné pro řešení zadaného projektu. Konečnou volbou byl NEC V850E/IG3. Hlavním faktorem při výběru mikrokontroléru bylo provedení, ve kterém je mikrokontrolér dodáván.

NEC V850E/IG3 je již osazen na desce. To vede k usnadnění práce a odpadne tím dosti složitá část návrhu plošného spoje. Vývojové prostředí IAR Embedded Workbench používané k tvorbě vnitřního programu je dodáváno k mikrokontroléru a je možné se s ním lehce seznámit. Velikost kódu nahrávaného do paměti je omezena na velikost 64 kB. Tato velikost je pro zadaný úkol dostatečná. Mikrokontrolér má dostatečný výpočetní výkon a varianta IG3 je přímo určena pro řízení motorů.

### 3.2.1. Mikrokontrolér NEC V850E/IG3

Mikrokontrolér NEC V850E/IG3 je 32bitový typ RISC [1]. Využívá se pro řízení vysoce výkonných systémů. Detailní popis pinů na čipu se nachází v příloze I. Blokové schéma tohoto mikrokontroléru je zobrazeno na obrázku 3.1.



Obr. 3.1: Blokové schéma NEC V850E/IG3 [3]

Na jednom čipu je umístěn vysokorychlostní procesor o frekvenci 64 MHz, flash paměť o velikosti 265 kB, RAM paměť o velikosti 12 kB, multifunkční čítače/časovače, A/D převodníky s 10bitovým nebo 12bitovým rozlišením, vysokorychlostní sériová rozhraní, indikátor nízkého napětí (LVI) a dva čítače pro generování PWM signálu. Zařízení pracuje při napájecím napětí +5V a vyrábí se v 100pinovém provedení s nízkým profilem čtvercové patice (LQFP). Umožňuje při použití vhodného programátoru a vývojového prostředí ladění programu přímo na mikrokontroléru [3].

### 3.2.1.1. Čítače a časovače

Mikrokontrolér obsahuje pět různých 16bitových čítačů/časovačů událostí, které se liší svými vlastnostmi a použitím, dvoukanálový časovač, uzpůsobený pro šestikanálový PWM signál k řízení motorů a časovač Watchdog.

**6-ch PWM pro řízení motorů x 2-ch** je tvořený dvěma kanály časovačů: TAB, TAA TMQOP. V závislosti na nastavené funkci a hodnotách časovačů může řídit šestikanálový PWM výstup, čítat impulzy atd.

**Timer AA (TAA<sub>n</sub> n= 0 to 4)** disponuje funkcemi capture, compare, čítač vnějších událostí, jednorázový impulsní výstup, PWM výstup, volně běžící časovač, měření šířky pulzu atd. Vstupní signál je možné dělit mocninou dvou pro časovače TAA0 a TAA1 nebo 2, 4, 8, 16, 64, 256, 1024, 2048 pro časovače TAA2 až TAA4. Funkce jednotlivých kanálů časovačů TAA<sub>n</sub> se mohou lišit.

**Timer AB (TAB<sub>n</sub> n= 0 to 1)** má stejné funkce jako časovač TAA, avšak umožňuje třífázový PWM výstup. Dělí vstupní signál osmi dělícími poměry. Vhodným nastavením TAB<sub>n</sub> a TMQOP<sub>n</sub> registrů je možné řídit funkci motorů.

**Timer T (TMT<sub>n</sub> n = 0 to 1)** má stejné funkce jako časovač TAA. Když se však nachází v režimu vnitřního časování, nemá funkci počítání externích událostí. Čítač při nastavení vhodné funkce porovnává impulzy na vstupech TENC<sub>n</sub>0, TENC<sub>n</sub>1. V závislosti na hodnotách těchto vstupů inkrementuje interní 16bitový čítač nahoru nebo dolů. Tuto funkci lze použít např.: při čítání impulzů pro funkci porovnání s CCR<sub>n</sub> buffer registrem nebo odečítání pozice.

**Timer M (TMM<sub>n</sub> n = 0 to 3)** má pouze omezený počet funkcí. Vstupní signál je možný dělit mocninou dvou v osmi dělících poměrech. Obsahuje jeden 16bitový čítač, ze kterého nelze číst, a porovnávací registr, do kterého nelze zapisovat během provozu časovače a také přerušení, které se vygeneruje po shodě s porovnávacím registrem.

**Watchdog** čítač/časovač slouží k ochraně programu proti skoku do programu, který není implementovaný tzv. zabloudění programu a následné uvíznutí. Má vlastní vstup pro hodinový signál, kde frekvence f<sub>xx</sub> je dělena 2<sup>10</sup>. Spouští se automaticky po spuštění programu nebo po resetu. Podle hodnoty příslušného registru se může nastavit maximální čas, do kterého časovač čítá a po jeho přetečení se vyvolá jedna ze dvou funkcí:

- reset operací po přetečení časovače watchdog a vygenerování signálu WDTRES
- vyvolání nemaskovatelného přerušení operací a vygenerování signálu INTWDT



### 3.2.1.2. A/D převodník

Součástí obvodu jsou dva analogově-digitální převodníky s 12bitovým rozlišením, které mohou načítat hodnoty ze dvou obvodů současně. Pro analogový vstup bez použití převodníku je možné použít maximálně 10 kanálů ze dvou obvodů: A/D převodník 0 i 1 používá 5 kanálů. Při použití převodníku lze použít maximálně 6/8 kanálů ze dvou obvodů: A/D převodník 0 i 1 používá 3 (4) kanálů. Mikrokontrolér dále obsahuje převodníky s 10bitovým rozlišením.

### 3.2.1.3. Debug Interface

Existují dva způsoby, jak odladit program přímo na čipu. Mezi těmito způsoby nelze příliš vybírat. Způsob ladění je závislý na použitém programátoru.

- použitý programátor MINICUBE

Tento způsob používá k ladění jednotku DCU. Programátor pracuje s piny rozhraní  $\overline{DRST}$ , DCK, DMS, DDI, DDO, pomocí kterého lze ladit program v průběhu vývoje přímo na mikrokontroléru. Laděný program se uloží do paměti Flash. MINICUBE je připojen přes rozhraní USB 2.0., jeho hlavní výhodou je, že neblokuje žádné rozhraní mikrokontroléru. Hlavní nevýhodou je cena programátoru, která je přibližně o třetinu větší než u MINICUBE2

- použitý programátor MINICUBE2

K ladění programu se nepoužívá jednotka DCU. Rozhraní pro ladění programu na mikrokontroléru si vybírá vývojář, který navrhuje a osazuje čip mikrokontroléru na desku. V mém případě je deska již navržena a ladící rozhraní přiřazené. Používá asynchronní sériové rozhraní UARTA0, u kterého jsou ke komunikaci využity piny RXDA0, TXDA0. Při použití synchronního rozhraní CSIB0 jsou využity piny SIB0, SOB0,  $\overline{SCKB0}$ , HS (P43). MINICUBE2 může napájet programovatelný obvod +5V, +3V nebo je možné napájení odpojit. Pro ladění programu se používá metoda monitor, která má některé příkazy uložené ve flash paměti a zbytek je posílán po příslušném rozhraní. Nevýhodou je zablokování jednoho rozhraní mikrokontroléru a ne příliš přesné časování. Hlavní výhodou je cena programátoru, která je \$ 177,66.

### 3.2.1.4. UART

Mikrokontrolér je vybaven čtyřmi sériovými rozhraními, která se mohou přepínat mezi synchronními a asynchronními režimy. Rozhraní UART0 (CSIB0) jsou rezervována pro debug, proto není možné s nimi počítat při realizaci. Komunikace tedy

může probíhat přes asynchronní sériová rozhraní UART1, UART2 nebo UARTB, popřípadě přes alternativní synchronní rozhraní I<sup>2</sup>C, CSIB1 nebo CSIB2, vždy však pouze přes jedno rozhraní z dané dvojice.

### 3.2.1.5. DMA

MCU má 4 nezávislé DMA řadiče s maximální velikost přenášeného bloku je 65 536, které používá pro přímý přístup do paměti a kontroluje data přenášená při DMA přenosu. DMAC řídí datové přenosy mezi vnitřní pamětí a vstupními/výstupními (V/V) perifériemi nebo mezi více V/V perifériemi na základě žádostí o přerušení od periférií (sériové rozhraní, čítače/časovač, A/D převodníky) nebo žádostí od DMA přenosu.

### 3.2.1.6. Přerušení

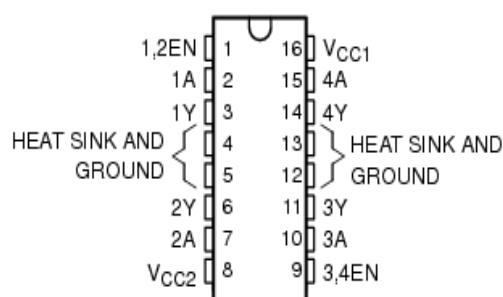
Pro servis a zpracování žádosti o přerušení je vyhrazen interrupt kontrolér (INTC), který může provádět celkem 89 žádostí o přerušení z možných 96. Přerušení je událost, která nastane nezávisle na provádění programu. Výjimkou je událost, která je závislá na prováděném programu. Žádost o přerušení může nastat od periférií na čipu nebo od externích zdrojů. Nemaskovatelné přerušení má pouze jeden interní zdroj, počet maskovatelných přerušení se mění podle druhu zařízení. NEC V850E/IG3 má 95 zdrojů přerušení, z toho je 21 zdrojů vnějších a 74 zdrojů vnitřních. Je možné vybrat si z 8 úrovní priorit a tím ovlivnit i pořadí, ve kterém se budou přerušení vykonávat. Hardwarové řešení přerušení využívá přerušovacího vektoru, který obsahuje, na jakém místě v paměti jsou umístěny skoky na obslužné procedury. Výhodou takového řešení je v podstatě neomezená délka obsluhy. Toto namapování na konkrétní proceduru obstarává vývojové prostředí přes direktivu např.: Direktiva odesílání dat na UART.

```
#pragma vector=INTUAIT_vector  
__flat __interrupt void UAIT_isr (void);
```

Pokud bychom tuto funkci nenamapovali, při přerušení se vykoná příkaz return a vrátí se zpět do hlavního programu. Toto může mít zásadní nevýhodu, ta vyplývá z toho, že obsluha některých přerušení neshazuje tzv. vlajku přerušení, což vede k zacyklení na obsluhovací proceduře. Typickým příkladem je příjem znaku na UART, kdy shození vlajky je podmíněno čtením datového registru.

### 3.2.1.7. Obvody pro řízení motorů

Jednotka pohybuje se vzorkem upevněným v křížovém pracovním stolku v osách X a Y. Motory není možné ovládat přímo z výstupů mikrokontroléru, které pro tuto funkci nejsou přizpůsobeny, proto je nutné připojit výkonové obvody pro jejich řízení. Řídit stejnosměrné motory je možné několika způsoby a za použití různých integrovaných obvodů. K realizaci byly použity vhodné můstkové budiče. Existuje mnoho takovýchto integrovaných obvodů, např.: L298 nebo LMD18245, oba v několika variantách, které obsahují plnohodnotné H můstky, jimiž jsou motory ovládány. Vybral jsem 4kanálový můstkový budič stejnosměrných motorů L293. Na obrázku 3.2 je zobrazen tento integrovaný obvod i s popsányými piny v tabulce 3.1.



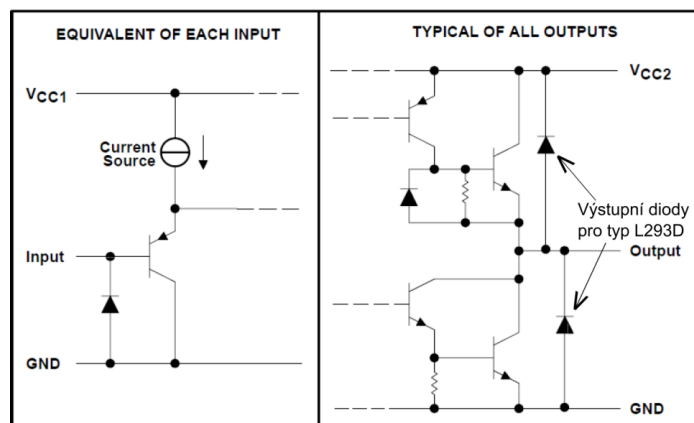
Obr. 3.2: Patice L293 [4]

Tab. 3.1: Funkce pinů

Číslo pinu	Funkce
1, 9	Povolení pro vstupy 1,2 nebo 3,4
2, 7, 10, 15	Vstupy 1, 2, 3, 4
3, 6, 11, 14	Výstupy 1, 2, 3, 4
4, 5, 12, 13	GND
16	Napájení +5V pro integrovaný obvod
8	Napájení pro motor 4,5V až 36V

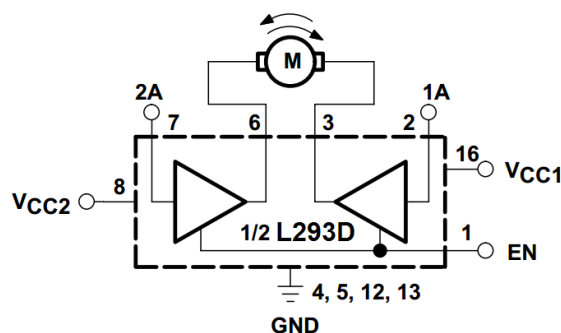
Všechny vstupy jsou kompatibilní s technologií TTL. Povolení se vždy provádí pro dvojici vstupů. Pokud je povolení v logické „1“, potom je možno pracovat s výstupními hodnotami. Je-li v logické „0“, potom je řízení motorů zakázáno. Obvod je navržen tak, aby mohl ovládat dva motory v obou směrech nebo čtyři motory pouze v jednom směru. Maximální výstupní proud je 1A a napájecí napětí v rozsahu 4,5V až

36V. Existují i další varianty tohoto obvodu, které nesou označení L293D, E, B. Pro realizaci diplomové práce byla vybrána varianta L293D. Ta pracuje stejně jako L293, pouze stálý výstupní proud na jeden kanál je 600mA a špičkový výstupní proud může být maximálně 1,2A. Výstupy u typu L293D mají na výstupních svorkách umístěny diody. Ty slouží k ochraně koncových tranzistorů H můstku a není tedy nutné dodávat je z vnějšku obvodu. Na obrázku 3.3 je vidět, jak jsou diody připojeny k výstupu. Výstup u obvodu L293D a L293 je stejný. Pouze u L293 nejsou na výstupu označené diody.



Obr. 3.3: Výstup integrovaného obvodu L293D [4]

Konkrétní připojení jednoho stejnosměrného motoru pro ovládání oběma směry je ukázáno na obrázku 3.4. K jednomu IO L293D je možné připojit dva stejnosměrné motory. V tabulce 3.2 je ukázáno, jaké úrovně signálu je nutné nastavit na jednotlivé vstupy, aby se motor točil v příslušném směru. Tyto obvody řídí pouze směr otáčení, rychlost otáčení je nutné řídit jiným způsobem.



Obr. 3.4: Připojení motoru k 1/2 L293D [4]

Tab. 3.2: Úrovně vstupních signálů pro L293D [4]

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

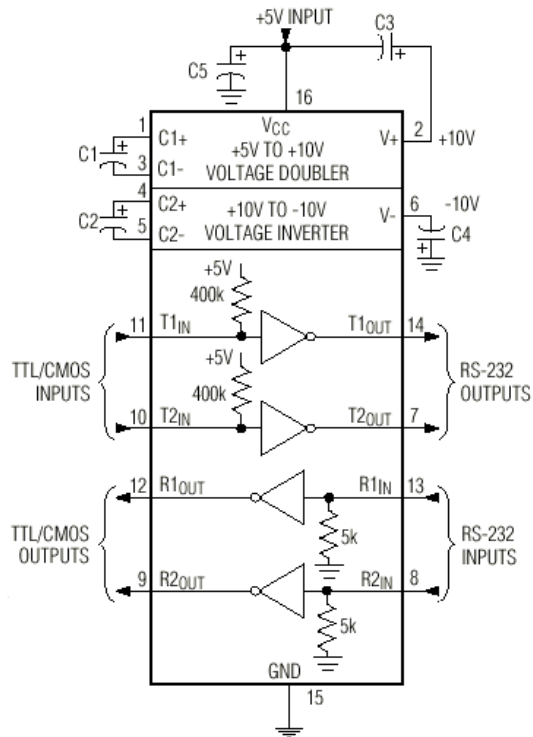
L = low, H = high, X = don't care

### 3.2.2. IO obvody pro komunikace

První ovládání jednotky bylo prováděno pomocí mechanických tlačítek. K tlačítkům byly připojeny pasivní součástky (odpory, kondenzátor), které odstranily zákmity vzniklé po stisku a následném uvolnění tlačítka a také bylo nutné nastavit vhodně piny mikrokontrolér. Ačkoliv tato funkce nebyla požadována v zadání, je implementována v programu a vstupy pro tlačítka jsou připraveny i na navrženém plošném spoji.

Ovšem požadavkem bylo ovládání jednotky pomocí osobního počítače. Komunikace mezi počítačem a mikrokonrolérem bude probíhat mezi asynchronním rozhraním UART1 na straně mikrokontroléru a sériovým rozhraním RS232 na straně osobního počítače.

Na pinech mikrokontroléru TXDA1 a RXDA1, které jsou uzpůsobeny pro přenos informace pomocí UART1, jsou logické úrovně TTL. Tyto úrovně však nejsou kompatibilní s úrovněmi na sériové lince RS232, proto je nutné použít integrovaný obvod MAX232, který slouží pro přizpůsobení úrovní a následnou komunikaci. Na obrázku 3.5 je patice tohoto obvodu včetně zapojení příslušných kondenzátorů a popsanych vstupů a výstupů. Z obrázku 3.5 je také vidět vnitřní zapojení a schopnost obvodů upravit až dvojici signálů. Já použiji pouze jeden kanál, proto je potřeba připojit i nezapojené vstupy, aby nedošlo k nevhodnému chování obvodu.



Obr. 3.5: Zapojení patice MAX232 [5]

### 3.2.3. Motory, převodovka, enkodér

Samostatné motory vhodné pro posun křížového stolku po pracovním stolku mikroskopu byly opět příliš velké, proto bylo nutné vybrat menší motory s malým krouticím momentem a osadit je planetární převodovkou, která by zvýšila celkový krouticí moment celku. Na základě toho se hlavními kritérii výběru staly hmotnost a rozměry.

#### 3.2.3.1. Ideální kombinace pro realizaci

První a ideální varianta kompletu pro pohon křížového stolku se skládá z komponent od firmy Maxon. Motor s označením RE16 a typovým číslem 320177, který je zobrazen na obrázku 3.6, má průměr těla 16 mm, kartáčky jsou z drahých kovů a maximální napájecí napětí je +12V. Při těchto podmínkách má motor výkon 2W, krouticí moment 2,3mNm a otáčky při zatížení 4340 ot/min .

Další součástí kompletu je přesná čelní nebo planetární převodovka, která je zobrazena na obrázku 3.7. Převodovka je určena pro motory Maxon, ozubená kola jsou výhodně uzpůsobena pro konkrétní typy motorů a ozubené kolečko prvního stupně převodovky je pevně připevněno k hřídeli motoru. Katalogové označení vybrané

převodovky je GP 16 A a typovým číslem 134781. Její průměr je 16 mm, má 4 stupně a převodový poměr 561:1. Výrobce ji doporučuje ke kombinaci s motorem RE 16, protože není problém s kompatibilitou.

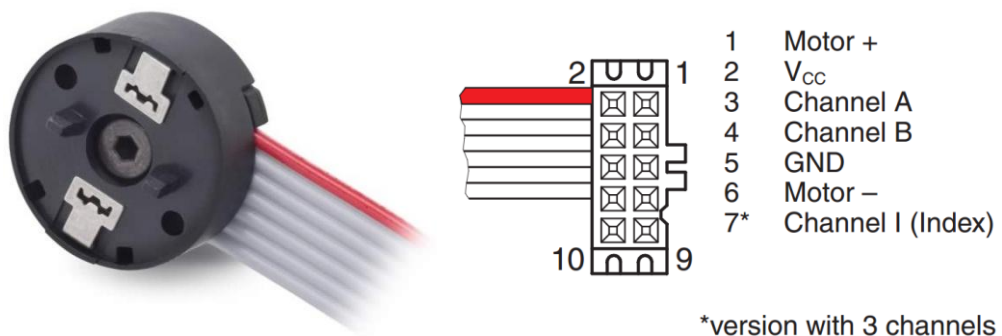


Obr. 3.6: Motor Maxon RE 16 [6]

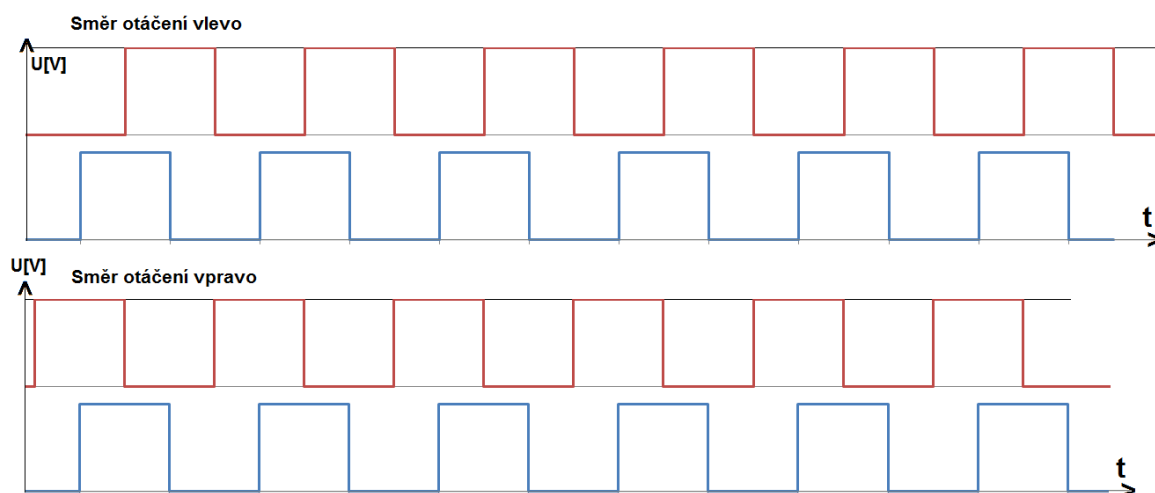


Obr. 3.7: Čelní nebo planetární převodovka Maxon GP 16 A [7]

Poslední částí celého kompletu je velmi přesný dvoukanálový enkodér s katalogovým označením MR a typu M zobrazený na obrázku 3.8 včetně umístění pinů na konektoru. Typovým číslem enkodéru je 201935. Napájecí napětí ( $V_{CC}$ ) celého enkodéru je v rozsahu 2,7 – 5,5V. Počet pulzů, které vygeneruje za jednu otáčku, je 32, maximální snímaná rychlost je 15000 ot/min. Fázový posun mezi oběma kanály je 45°. Na obrázku 3.9 je vidět, jak vypadají výstupní průběhy signálů enkodéru jednotlivých kanálů v závislosti na směru otáčení motoru.

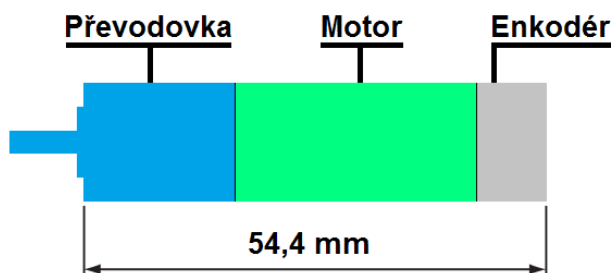


Obr. 3.8: Enkodér Maxon MR a popsání pinů konektoru [8]



Obr. 3.9: Časový průběh výstupu enkodéru Maxon MR

Výsledná délka celého kompletu bude 54,4 mm (viz obrázek 3.10) a hmotnost bude přibližně 60 g (motor = 21 g, převodovka = 31 g, enkodér = 8 g). Otáčky vypočtené podle rovnice (3.1) budou 0,129 ot/s. To znamená, že jedna otáčka osy motoru bude trvat 7.75 sekundy.



Obr. 3.10: Celková délka kompletu

$$(Rovnice 3.1) \quad \text{Výsledné otáčky} = \frac{\frac{\text{otáčky motoru}}{60}}{\text{převodový poměr}} = \frac{\frac{4340}{60}}{561} = \mathbf{0,129 \text{ ot/min}}$$

Výsledný počet pulzů, které enkodér vygeneruje za jednu otáčku, bude podle rovnice (3.2) 17952 pulzů/otáčku.

$$(Rovnice 3.2) \quad \text{Výsledný počet pulzů} = \text{Počet}_{\text{pulzů}} * \text{Převodový poměr} = 32 * 561 = \mathbf{17952 \text{ pulzů/otáčku}}$$



Celý komplet od firmy Maxon je ideální pro použití v mém projektu, ale jeho hlavní nevýhodou je jeho cena. Ceny jednotlivých komponent jsou v tabulce 3.3 uvedeny ve Švýcarských francích, což při kurzu 21,10 Kč za jeden Švýcarský frank znamená cenu 4469 Kč. Tato cena je v závislosti na použití příliš vysoká, proto bylo nutné vyhledat přijatelnější variantu.

Tab. 3.3: Ceny položek Maxon

Název položky	Cena položky CHF
Motor RE16	79,60
Převodovka GP 16 A	74,10
Enkodér MR	58,10
Celková	211,80

### 3.2.3.2. Cenově dostupná varianta

Pro druhou cenově dostupnou variantu pohonu křížového stolku je možné použít motor od firmy SGSTCL typ GM12-N20VA110, který je možné objednat z GM electronic. Součástí kompletu je i převodovka, viz obrázek 3.11. Při napájecím napětí +5V a jmenovité zátěži je krouticí moment 770 g/cm, odběr 200mA a otáčky 35 ot/min. Rozměry tohoto motoru jsou 12 mm v průměru, výška 10 mm, délka celku je 25 mm (převodovka 9 mm + motor 16 mm). Délka výstupní hřídele je 4,4 mm. Možný převodový poměr je v rozsahu 10 – 298/1.

K tomuto motoru je nutné ještě připojit obvod pro snímání polohy. Jeden ze způsobů snímání je obarvení prvního kolečka připojeného k motoru dvěma různými barvami (černá a bílá), a poté optické snímání kolečka a počítání přechodů mezi barvami. Protože toto snímání není ideální pro mé použití, hledal jsem jiné možnosti jak snímat polohu. Podařilo se mi najít celkový komplet, který se skládá z motoru GM12-N20VA i enkodéru (viz obrázek 3.12), který lze objednat z [10]. Cena tohoto kompletu je v rozsahu 3,5 – 5,2 US \$/kus. Bohužel je nutné objednat větší množství kusů.



Obr. 3.11: Motor F-GM12-N20VA110  
[9]



Obr. 3.12: Motor F-GM12-N20VA110  
s enkodérem [10]

### 3.2.3.3. Testovací varianta

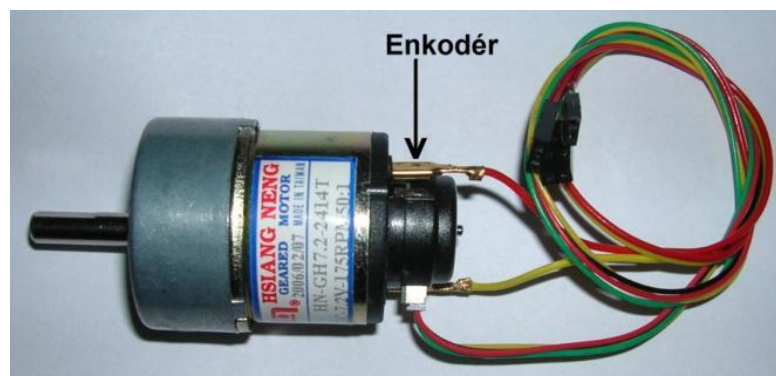
Třetí variantu určenou pro testování pohonu křížového stolku byla sestavena z komponent, které již zakoupila katedra v dřívější době, a nejvíce se podobaly ideální sestavě. Hlavním rozdílem je velikost a váha celého kompletu.

Použitý motor, jehož součástí je již i převodovka, má označení GHM-04, viz obrázek 3.13. Rozměry tohoto motoru jsou 37 mm vnější průměr a 123 g hmotnost. Napájecí napětí tohoto motoru je 7,2V, odběr při jmenovité zátěži je 556mA, otáčky 175 ot/min a převodový poměr 50:1. Mezní krouticí moment je 7,1 kg/cm.



Obr. 3.13: Motor s převodovkou GHM-04 [11]

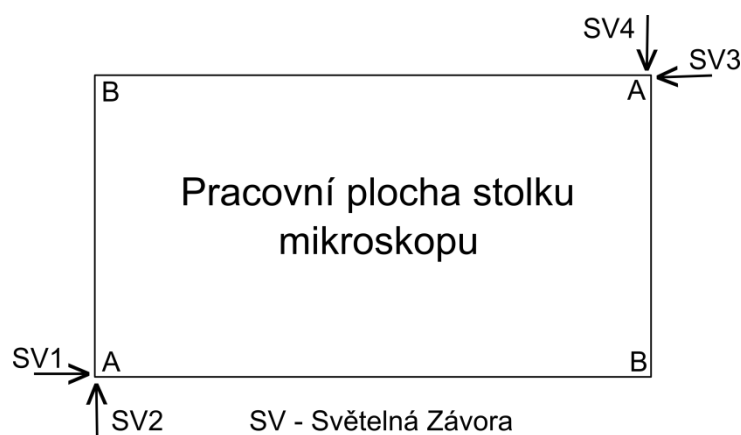
K zadní části motoru byl připojen enkodér s označením QME-01, který má na výstupech podobný signál jako na obrázku 3.9. Tyto signály je však nutné před vstupem do mikrokontroléru upravovat pomocí dalších obvodů. Enkodér je napájen +5V. Za jednu otočku se vytvoří 400 pulzů na výstupu a mezní frekvence snímání je 30kHz. Celý tento mechanismus je vidět na obrázku 3.14.



Obr. 3.14: Celý testovací mechanismus

### 3.2.4. Světelné závory

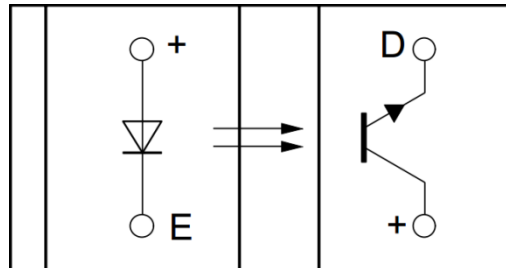
Z důvodu ochrany proti vysunutí pracovního vzorku mimo prostor pracovního stolku, jeho vymezení a zjištění počáteční pozice bylo nutné použít světelné závory. Pro zajištění všech tří podmínek je nutné použít čtyři světelné závory, viz obrázek 3.15.



Obr. 3.15: Umístění světelných závor na pracovní stolek

Existuje několik druhů světelných závor. První, který lze použít, se skládá ze dvou oddělených obvodů. V jednom obvodu je umístěn zdroj světelného signálu a v druhém obvodu je umístěn přijímač (optočlen). Zdroje signálů jsou umístěny na pracovním stolku mikroskopu podle obrázku 3.15 v pozicích označených písmenem A, přijímače v místech označených písmenem B. Po vniknutí snímaného vzorku do prostoru světelné závory se změni hodnota na jejím výstupu na opačnou.

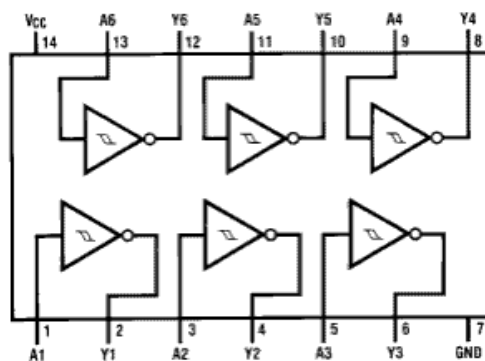
Druhý má přijímač i vysílač umístěn v jednom pouzdře. Takovéto světelné závory se nazývají vidlicové. Na obrázku 3.16 je zobrazeno vnitřní zapojení vidlicové světelné závory. U závory tvořené dvěma obvody je zapojení stejné, pouze jsou dále od sebe.



Obr. 3.16: Vnitřní zapojení světelné závory [12]

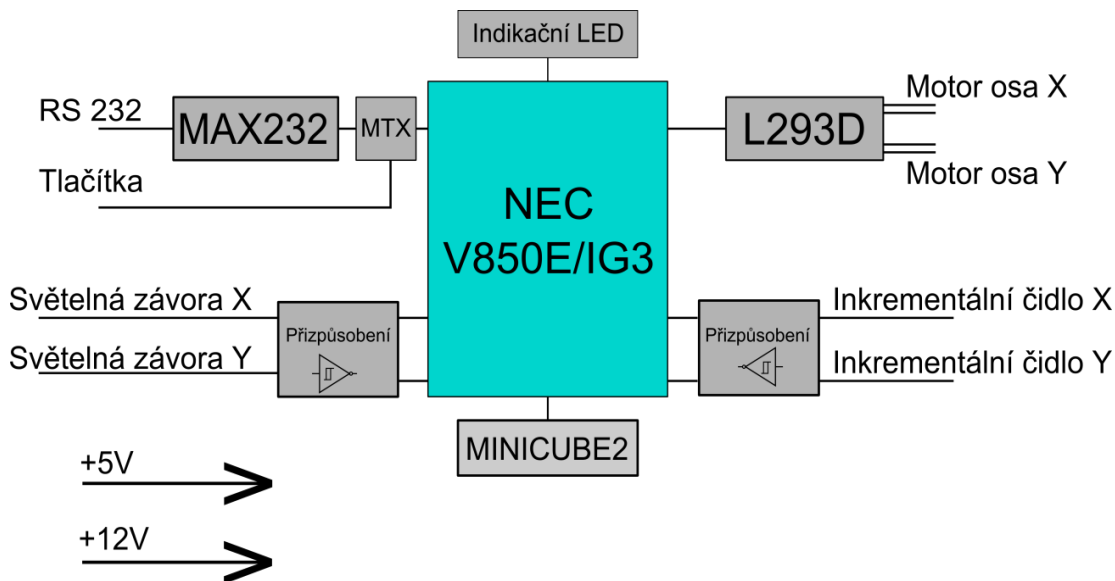
### 3.2.5. Přizpůsobovací členy

Pro připojení enkodérů a světelných závor k mikrokontroléru bylo potřeba použít obvody pro přizpůsobení signálů. Vybral jsem invertující Schmittův klopný obvod, který slouží k úpravě signálů a jeho základní vlastností je hystereze. Výstupní signál z hysterezního členu je závislý na hodnotě vstupu a současně i na hodnotě původního stavu. K praktické realizaci je možné použít integrovaný obvod s označením 74HC14N, který obsahuje šest takovýchto členů. Konkrétní vnitřní zapojení, viz obrázek 3.17.



Obr. 3.17: Vnitřní zapojení 74HC14N [13]

Po výběru všech obvodů nutných pro řízení a komunikaci bylo možno navrhnout schéma zapojení. Výsledné blokové schéma s konkrétním označením jednotlivých bloků je na obrázku 3.18 a v příloze II. je navržené detailní schéma zapojení.

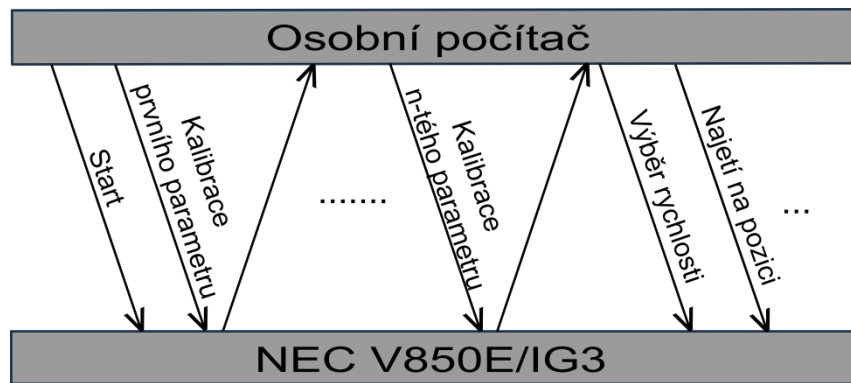


Obr. 3.18: Blokové schéma polohovací jednotky s označením bloků

## 4. Návrh řídicí elektroniky včetně firmware

### 4.1. Návrh komunikace

Na začátku tvorby firmware pro mikrokontrolér a uživatelského rozhraní pro osobní počítač bylo nutné navrhnout komunikační protokol mezi těmito zařízeními. Rozhraní osobního počítače pro komunikaci bylo vybráno RS232. Poté jsem upravil signály pro mikrokontrolér a vybral příslušný komunikační protokol. Přenos dat probíhá sériovým plně duplexním přenosem po dvou hlavních vodičích TxD (vysílaná data) a RxD (přijímaná data) s možností využití dalších řídicích vodičů. Velikost přenášených dat se může lišit s ohledem na velikost jednotlivých sekcí. Datové sekce mají rozsah 4 – 8 bitů, různé typy paritního bitu (lichá, sudá, jedničková, nulová, žádná) a různý počet stop bitů (jeden, jeden a půl nebo dva). Na obrázku 4.1 je zobrazena ukázka komunikace po sériové lince pro polohovací jednotku.



Obr. 4.1: Navržená komunikace

## 4.2. Návrh schématu a plošného spoje

Existuje mnoho návrhových systémů, které se dají použít pro návrh schématu a plošného spoje této diplomové práce. Bylo možné vybrat ze dvou systémů, Eagle Light a FORMICA 4.40, které jsou volně ke stažení, ale jejich využití je omezené velikostí navrhovaného plošného spoje, počtem součástek nebo vývodů atd. Vzhledem k tomu, že Západočeská univerzita v Plzni vlastní síťovou licenci na neomezenou verzi návrhového systému FORMICA 4.40, byla zvolena tato varianta.

Z vybraných součástek a integrovaných obvodů bylo navrženo schéma zapojení a také plošný spoj. K tvorbě schématu slouží editor SCHEMATIC 4.40. Schéma bylo navrhováno na velikost stránky A4, aby vše bylo přehledné. Většina součástek a integrovaných obvodů je součástí knihoven, které se nahrají při instalaci programu. Pouze mikrokontrolér NEC V850E/IG3 nebyl obsažen v knihovnách programu. Proto bylo nutné vytvořit nebo nalézt tuto značku. Symbol byl vybrán v katedrální databázi součástek. Na jednotlivé periferie mikrokontroléru byly připojeny vývody a dále ostatní obvody. Výsledné schéma viz příloha II.

Jednotlivé konektory pro připojení externích vstupů/výstupů jsou označeny Hx, kde x je číslo konektoru.

### H1, H2 – pulzy pro motory, signály od inkrementálního čidla

pin 1 – vstup IČ

pin 2 – vstup IČ

pin 3 – výstup 1 pro motor 1(2)

pin 4 – výstup 2 pro motor 1(2)

**H3 – hlavní napájecí konektor**

pin 1, pin 3, pin 5 – napětí +12V

pin 2, pin 4, pin 6, pin 8 – GND

**H4, H5 – napájení inkrementálního čidla**

pin 1 – GND

pin 2 – napětí +5V

**H6 – komunikační sériová linka RS232**

pin 2 – DSR

pin 3 – RXD

pin 5 – TXD

pin 7 – DTR

pin 9 – GND

**H7, H8 – připojení světelných závor**

pin 1 – VCC

pin 2 – vstup světelné závory 1(2)

pin 3 – GND

pin 4 – GND

**H9 – napájecí konektor NEC V850E/IG3**

pin 1 – VCC

pin 3 – GND

**H10 – tlačítko reset**

pin 1 – GND

pin 2 – T\_RESET

**H11 – tlačítko stop s aretací**

pin 1 – GND

pin 2 – PDL\_0

**H12 – konektor pro připojení tlačítek k manuálnímu ovládní**

pin 1, pin 3, pin 5, pin 7 – připojení tlačítek

pin 2, pin 4, pin 6, pin 8 – GND

Diody D2 – D5 indikují směr otáčení motoru. K diodám je do série zapojený odpor, který omezuje proud protékající diodami. U napájecího konektoru H3 je připojena pojistka s minimální hodnotou 1 A a Zenerova dioda před stabilizátorem napětí, která snižuje jeho vstupní napětí. Stabilizátor vybraný pro tento účel je

7805-STM v provedení TO-220, ke kterému je možno připevnit dodatečné chlazení. Na vstupu i výstupu stabilizátoru jsou připojeny kondenzátory pro filtrování napětí.

Po dokončení schématu byl vygenerován netlist (seznam propojení jednotlivých součástek) a partlist (seznam součástek), ve kterém se nachází označení součástek, patice nebo hodnoty součástek.

Dalším krokem byl návrh plošného spoje. K tomuto návrhu se používá druhá část programu FORMICA 4.40, nazvaná LAYOUT 4.40. Před vlastním návrhem desky bylo nezbytné nastavit prostředí LAYOUT 4.40. Vybrat rastr, ve kterém se součástky budou rozmísťovat, a nastavit vhodnou izolační vzdálenost mezi vodiči. V první verzi návrhu byl použit rastr metrický. Ten však ukázal jako nevhodný, protože většina integrovaných obvodů má patice navržené v palcovém měřítku a desku osazenou mikrokontrolérem nebylo možné vsadit do takto navržené patice. Po nastavení prostředí bylo nutné vybrat rozměry plošného spoje. Ten musel být větší než deska, na které je osazen NEC V850E/IG3 a zároveň dostatečně velký, aby se na něj vešly všechny potřebné obvody a konektory. Rozměry plošného spoje splňující tyto podmínky byly 100x80 milimetrů. Podle rozměrů plošného spoje bylo možné vybrat krabičku. Nejvhodnější byla krabička s označením U-KM 35 (viz obrázek 4.2), její vnější rozměry jsou 110x90x32 mm (d x š x v) a je možné do ní vložit plošný spoj o rozměrech 100x84 mm [17]. Krabičku bylo nutné upravit, protože je vyrobena pro uchycení plošného spoje uprostřed. Tento typ uchycení nemohl být použit, protože mikrokontrolér zabírá více než polovinu plošného spoje. Proto bylo nutné upravit krabičku k uchycení destičky na okrajích.



Obr. 4.2: Krabička U-KM 35 [17]



Při rozmístování součástek a integrovaných obvodů chyběla v knihovných návrhového systému patice k osazení destičky s mikrokontrolérem, proto bylo nutné vytvořit. Tvorba byla složitá, protože rozchod dvojice konektorů osazených na okraji desky, které slouží k propojení s ovládanou deskou, není definován. K zjištění požadovaného rozchodu byla použita dutinková lišta neosazeného konektoru, pomocí které se propojili konektory na okrajích desky, a spočítal počet pinů mezi nimi. Poté bylo možné navrhnout patici i s konektorem pro napájení destičky s mikrokonrolérem. Integrované obvody 74HC14N a MAX232 byly umístěny pod destičku s mikrokonrolérem, protože jejich vyzařované teplo je minimální. Jedinou nevýhodou takto osazených obvodů je špatný přístup při jejich výměně či opravě. Výkonový obvod pro řízení motorů L239D se během provozu ohřívá, proto byl osazen vedle desky s mikrokontrolérem. Na piny konektorů patice mikrokontroléru jsou připojeny následující signály.

**Konektor CN1:**

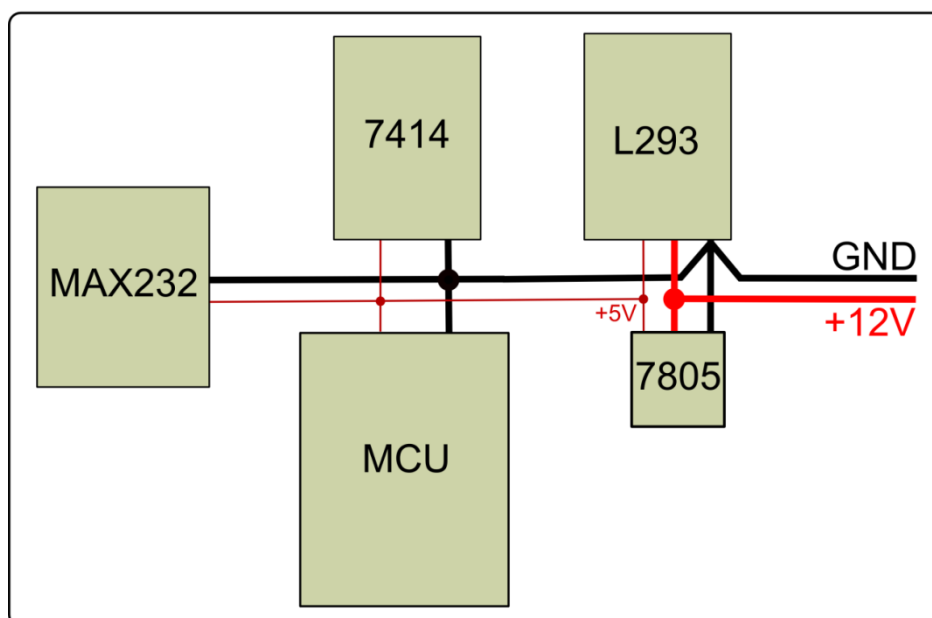
- pin 1,7,12,27,37,42 – GND
- pin 28,29,30,31 – přídavné LED
- pin 40 – RESET

**Konektor CN2:**

- pin 1 – TENC10
- pin 2 – TENC11
- pin 5 – RXDA1
- pin 6 – TXDA1
- pin 16 – hardwarové tlačítko vpravo
- pin 17 – hardwarové tlačítko vlevo
- pin 18 – hardwarové tlačítko nahoru
- pin 19 – hardwarové tlačítko dolů
- pin 22 – světelná závora 2
- pin 23 – světelná závora 1
- pin 24 – ENAB B
- pin 25 – ENAB A
- pin 31 – centrální STOP
- pin 32 – TENC01

pin 33 – TENC00  
pin 44 – R2OUT  
pin 45 – T2IN  
pin 47 – IN2,4 (výstup časovače TOB03)  
pin 49 – IN1,3 (výstup časovače TOB01)  
pin 64, 85 – GND

Ostatní konektory by měly být co nejbližší k okraji desky, aby bylo jednodušší jejich vyvedení vně krabičky. Blokovací kondenzátory byly umístěny co nejbližší k napájecím pinům integrovaných obvodů. Při rozvodu napájení a země (GND) bylo nutné dodržet strukturu propojení zobrazenou na obrázku 4.3.



Obr. 4.3: Rozvod napájení a GND na desce

Obvody jsou zapojeny tímto způsobem, aby zemní proudy motorů neprotékaly po stejných větvích jako proudy signálových vodičů. Z důvodu nežádoucího odporu vedení by docházelo k poklesu napěťových úrovní na vstupech integrovaných obvodů připojených na stejnou větev jako motory a tím k chybám při užívání. Výsledný návrh plošného obvodu v jednotlivých vrstvách je v příloze III. a osazovací schéma se nachází v příloze IV.

### 4.3. Návrh firmware pro mikrokontrolér NEC V850E/IG3

Pro návrh firmware jsem použil vývojové prostředí KickStart edition of IAR Embedded Workbench for V850 s omezením velikosti vyvíjeného kódu na 64kB.

Všechny informace o nastavení jednotlivých registrů byly nalezeny v Hardware User's Manual [1], který je možno volně stáhnout ze zdroje na internetu. Díky vzorovým příkladům umístěným na stránkách výrobce bylo jednodušší pochopit a vyřešit několik problémů, které při tvorbě firmware vznikly.

Na začátku tvorby firmware byl použit příklad pro blikání LED osazených na desce s mikrokontrolérem. Tento příklad je umístěn na stránkách výrobce a jeho součástí je nejen hlavní program, ale také soubory, které slouží k nastavení a inicializaci mikrokontroléru [18]. Jednotlivé moduly mají následující funkce.

globals.c – definice globálních dat

globals.h – deklarace globálních dat

interrupt.c – rutiny pro obsluhu přerušeni

low\_level\_initialization.c – základní nastavení mikrokontroléru

low\_level\_initialization.h – základní nastavení maker mikrokontroléru

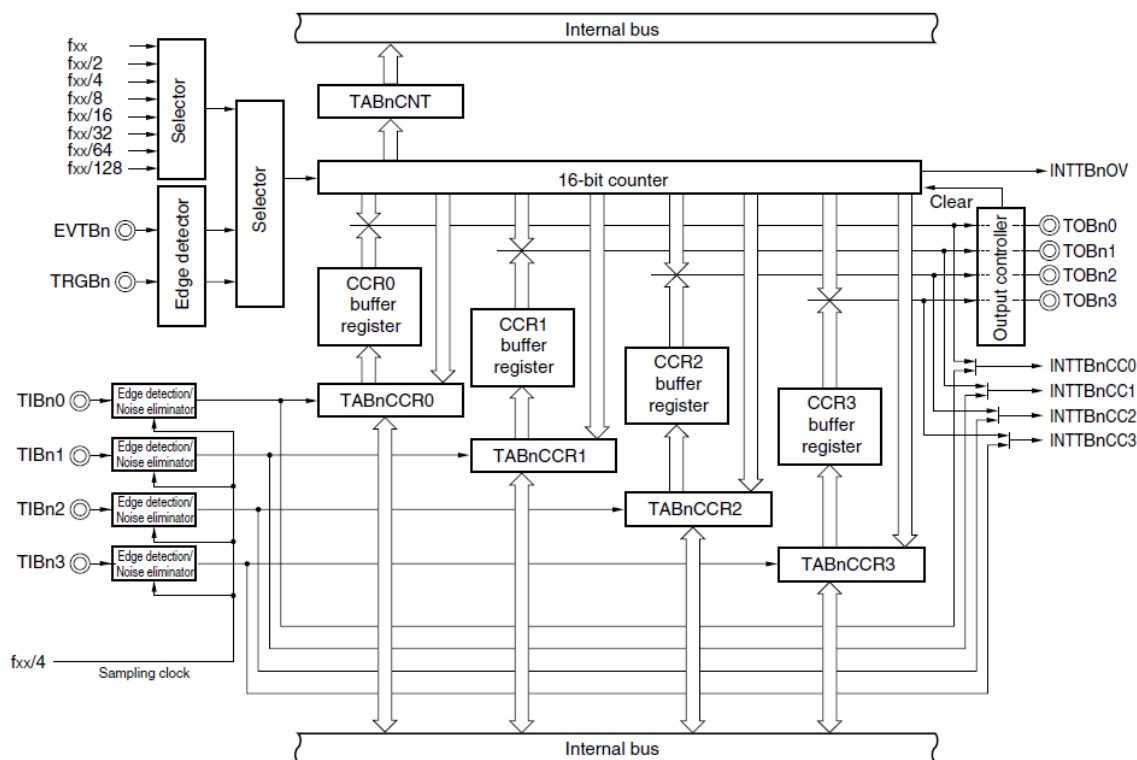
macros.h – definice datových typů

main.c – hlavní vykonávaný program

types.h – definice datových typů

Výsledný firmware pro polohovací jednotku jsem tvořil v main.c. Celý program je součástí dokumentace uložené na přiloženém CD. Při tvorbě programu bylo postupováno následujícím způsobem. Nejdříve bylo třeba nastavit časovače a portu k získání PWM výstupu, který je nutný pro řízení rychlosti stejnosměrných motorů. K řízení je nejvhodnější použít časovač TAB.

Na obrázku 4.4 je znázorněno jeho blokové schéma. Časovač obsahuje zařízení pro výběr vstupního časování, vnitřní 16bitový čítač, s jehož obsahem se porovnává obsah registrů CCRn a při shodě se vygeneruje buď impuls pro výstupní časovače TOBn0 – TOBn3, nebo přerušeni INTTBnCCn. Toto přerušeni se také vygeneruje při příchodu pulzů na vstupní časovače TIBn0-TIBn3. Vstupy časovačů TIBnm jsou osazeny detektory hran/omezovačem šumu, kde jsou nastaveny hodiny pro vzorkování na  $f_{xx}/4$ .



Obr. 4.4: Blokové schéma časovače TABn [1]

Nastavením následujících registrů docílíme požadované funkce registru TAB.

**TAB0 control register 0 (TAB0CTL0)** – nastavení vstupního časovače obsahuje bit TAB0CE, jehož nastavením se povolí či zakáže operace vykonávané časovačem

**TAB0 control register 1 (TAB0CTL1)** – nastavení režimu, ve kterém časovač pracuje

**TAB0 option register 0 (TAB0OPT0)** – nastavením zajistíme, že CCR registr bude nastaven jako porovnávací

TAB obsahuje ještě další registry např.: umožňuje generování vstupně/výstupních událostí, určuje hranu, na kterou bude časovač reagovat atd.

V následující ukázce programu je konkrétní nastavení časovače TAB pro polohovací jednotku.

```
TAB0CE = 0;           /* vypnutí časovače pro nastavení */
TAB0CTL0 = 0x03;     /* frekvence vstupních hodin fxx/8 */
TAB0CTL1 = 0x04;     /* nastavený režim PWM */
TAB0OPT0 = 0x00;     /* nastavení registru CCR0 a CCR1 jako porovnávací */
```

```

TAB0IOC0 = 0x00;      /* piny časovačů TOB00-TOB03 vynulovány */
TB0CCMK0 = 1;        /* povolení přerušení po shodě s registrem CCR0 */
TB0CCMK1 = 1;        /* povolení přerušení po shodě s registrem CCR1 */
TB0CCMK2 = 1;        /* povolení přerušení po shodě s registrem CCR2 */
TB0CCMK3 = 1;        /* povolení přerušení po shodě s registrem CCR3 */
TB0OVMK = 1;         /* povolení přerušení časovače TAB0 */
TAB0CE = 1;          /* zapnutí časovače po nastavení */

```

Po ukončení nastavení časovače TAB0 je možné zadat hodnoty do registrů CCR0 – CCR3. Ty nastavují střihu výstupního signálu časovačů TOB01 a TOB03, jejichž výstup je stejný jako PWM signál. V následující ukázce programu je nastavení registrů pro nejpomalejší rychlost otáčení 1 a nejrychlejší 10. Jednotlivé hodnoty pro registry TAB0CCR1 a TAB0CCR3 byly získány vydělením maximální hodnoty 61 440. Cílem bylo vytvořit 10 rychlostí posuvu, proto bylo nutno dělit maximální hodnotu jedenácti, protože rychlost při hodnotách všech registrů 61 440 je nulová.

```

/* rychlost otáčení 1 */
TAB0CCR0 = 61440;
TAB0CCR1 = 55855;
TAB0CCR2 = 61440;
TAB0CCR3 = 55855;

/* rychlost otáčení 10 */
TAB0CCR0 = 61440;
TAB0CCR1 = 5590;
TAB0CCR2 = 61440;
TAB0CCR3 = 5590;

```

Dalším krokem bylo nastavení příslušného portu a jednotlivých pinů podle potřeby. K nastavení portů P0 – P7 slouží následující registry o velikosti 8 bitů.

**Port n mode register (PMn)** – nastavení pinů portu na vstup nebo výstup

**Port n register (Pn)** – nastavení výstupu do logické „1“ či logické „0“

**Port n mode control register (PMCn)** – nastavení alternativní funkce portu nebo nastavení jako V/V port

**Port n function control register (PFCn)** – specifikuje alternativní funkci pinu na portu n, každý bit registru odpovídá jednomu pinu portu

**Port n function control expansion register (PFCEn)** – specifikuje konkrétní funkci jednotlivých pinů, protože každý pin má rozdílné funkce. Používá se v kombinaci s PFCn

**Pull-up resistor option register (PUn)** – nastavením tohoto registru je možné k některým portům připojit interní pull-up odpor, který mění jeho hodnotu na logickou „1“, používá se například u tlačítek vyvolávající přerušení

V následující ukázce programu je port 1 nastaven pro výstup časovačů a také pro vstup a výstup pomocných signálů potřebných ke komunikaci po sériové lince.

```

/*nastavení pinu P15 pro výstup T2IN a P14 pro vstup R2OUT do nuly*/
PM1  = 0x20;    /* pin 15 výstupní, ostatní vstupní */
P1    = 0x00;    /* na všech pinech bude na počátku nastavena „0“ */
PMC1 = 0x87;    // pin P17 nastaven jako TOB00 output/INTP09 input/A7 output
                          //pin P12 nastaven jako TOB0T2 output/TIB03 input/TOB03 output/A2 output
                          // pin P11 nastaven jako TOB0B1 output/TIB02 input/TOB02 output/A1 output
                          // pin P10 nastaven jako TOB0T1 output/TIB01 input/TOB01output/A0 output
                          // ostatní nastaveny jako V/V

/* nastavení konkrétní funkce */
PFC1  = 0x00;    /* TOB00 = OUT, TOB03 = OUT, TOB02 = OUT, TOB01 = OUT */
PFCE1 = 0x07;    /* TOB00 = OUT, TOB03 = OUT, TOB02 = OUT, TOB01 = OUT */

```

Po nastavení portu a časovače pro řízení motorů bylo nutné nastavit port DL pro signály ENAB\_A, ENAB\_B, SVZ1, SVZ2 a možné připojení tlačítek. Port DL je možné nastavit jako V/V nebo ADn vstup/výstup. K nastavení portu pro příslušnou funkci nebo jako V/V slouží registry (viz tabulka 4.1), které mají stejné funkce jako registry pro porty P0 - P7, ale jejich velikost je 16 bitů.

Nastavení je však možné provádět po 8 bitech, protože každý z těchto registrů je rozdělen na dvě části. Část označená písmenem **L** slouží k nastavení pinů portu PDL0 – PDL7 a část označená písmenem **H** k nastavení pinů portu PDL8 – PDL15. Přesné označení jednotlivých částí viz tabulka 4.1.

Tab. 4.1: Rozdělení registrů

Název registru	Označení L části	Označení H části
<b>Port DL register (PDL)</b>	PDLL	PDLH
<b>Port DL mode register (PMDL)</b>	PMDLL	PMDLH
<b>Port DL mode control register (PMCDL)</b>	PMCDLL	PMCDLH
<b>Pull-up resistor option register DL (PUDL)</b>	PUDLL	PUDLH

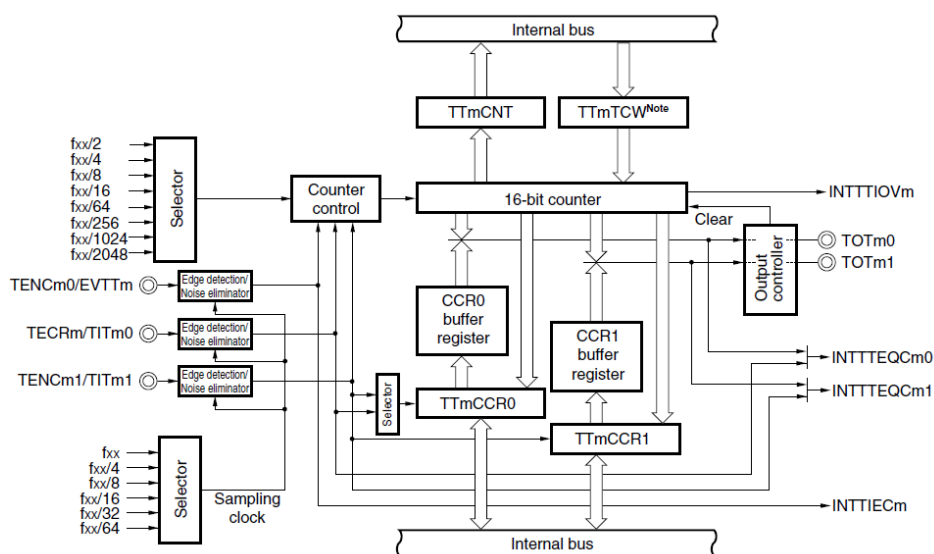
V následující ukázce programu je konkrétní nastavení portu DL pro příslušné signály a vstupy.

```

/*nastavení registru PMCDL*/
PMCDLH = 0x00;      /* PDL0 – PDL7 V/V */
PMCDLL = 0x00;      /* PDL8 – PDL18 V/V */
/*nastavení registru PMDL*/
PMDLH = 0xF3;       /*VSTUP: PDL8, PDL9, PDL12 – PDL15, VÝSTUP: PDL10, PDL11*/
PMDLL = 0x00;       /* VÝSTUP: PDL0 – PDL7 */
PUDL = 0xF000;      /*kontrola, zda jsou na pinech PDL12-PDL15 připojena tlačítka*/
/* počáteční nastavení výstupu pinů PDL6 (ENAB_A), PDL7 (ENAB_B) do logické „1“ */
PDL = 0xC0;

```

Když bylo možné ovládat rychlost a směr otáčení motorů, bylo nutné zabývat se zjišťováním pozice, ve které se snímáný vzorek nachází. V uživatelské příručce je uvedeno, že časovač TMT, jehož blokové schéma je uvedeno na obrázku 4.5, umožňuje při vhodném výběru funkce čítat pulzy v závislosti na signálech připojených na vstupní piny mikrokontroléru TENCn0, TENCn1. Z důvodu snímání hodnot ze dvou motorů jsem použil tyto vstupy od dvou čítačů/časovačů TMT. Každá dvojice těchto pinů je umístěna na jiném portu, TENC00 a TENC01 na portu P0 a TENC10, TENC11 na portu P4. K nastavení těchto portů se používají stejné registry jako při nastavení portu P1 u časovače TAB.



Obr. 4.5: Blokové schéma časovače TMTn [1]

Nastavení čítače/časovač TMT se provádí pomocí následujících registrů.

**TMTn control register (TTnCTL0)** – slouží k nastavení vstupní frekvence hodin a obsahuje bit TTnCE, pomocí kterého se povoluje/zakazuje provádění operací časovače TMT, tento bit je nutné na začátku konfigurace nastavit na logickou „0“ a po dokončení nastavení na logickou „1“

**TMTn control register 1 (TTnCTL1)** – nastavuje režim, ve kterém časovač pracuje. Je možné vybrat z následujících možností:

- vnitřní časovač
- čítač vnějších událostí
- vnější spouštěč výstupních pulzů
- jednorázový impulzní výstup
- PWM výstup
- volně běžící časovač
- režim měření délky pulzu
- trojúhelníkový PWM výstup
- režim zachytávání a porovnání

**TMTm control register 2 (TTmCTL2)** – řídí funkce v režimu porovnávání snímání a je tedy možné ho použít pouze v tomto režimu, nastavením bitů TTmUDS0 a TTmUDS1 se inkrementuje nebo dekrementuje vnitřní 16bitový čítač, v tabulce 4.2 jsou uvedeny kombinace, které je možno nastavit a tím vybrat potřebný způsob čítání pulzů

Tab. 4.2: Nastavení vhodného čítání pulzů v závislosti na vstupech [1]

TTmUDS1	TTmUDS0	Up/down count selection
0	0	When valid edge of TENCm0 input is detected Counts down when TENCm1 = high level. Counts up when TENCm1 = low level.
0	1	Counts up when valid edge of TENCm0 input is detected. Counts down when valid edge of TENCm1 input is detected.
1	0	Counts down when rising edge of TENCm0 input is detected. Counts up when falling edge of TENCm0 input is detected. However, count operation is performed only when TENCm1 = low level.
1	1	Both rising and falling edges of TENCm0 and TENCm1 are detected. Count operation is automatically identified by combination of edge detection and level detection.



**TMTm I/O control registers 0 to 3 (TTmIOC0 to TTmIOC3)** – slouží k nastavení vstupních a výstupních časovačů TMT, pomocí registru TTmIOC3 je možné nastavit mazání čítače po příchodu příslušné úrovně signálů na vstupy TENCn0, TENCn1, TECRn nebo nastavení platné hrany (náběžné, doběžné, obou), na kterou mikrokontrolér reaguje

**TMTn option register 0 (TTnOPT0)** – nastavením příslušných bitů se vybere, zda registry CCR0, CCR1 budou nastaveny jako porovnávací nebo zachytávací, bit TTnOVF detekuje přetečení 16bitového čítače

**TMTm option register 1 (TTmOPT1)** – z tohoto registru je možno vyčíst, zda došlo k přetečení nebo podtečení 16bitového registru, a také směr čítání pulzů (nahoru/dolů), tedy i směr otáčení motoru

V následující ukázce kódu je vidět, nastavení čítač/časovač TMT0 pro motor 1. Nastavení pro motor 2 je totožné, pouze používá čítač/časovač TMT1.

```
TT0CE = 0;           /* vypnutí časovače pro nastavení */
TT0CTL0 = 0x00;     /* frekvence vstupních hodin fxx/2 */
TT0CTL1 = 0x08;     /* režim zachytávání a porovnání */
TT0CTL2 = 0x00;     /* TTmUDS1, TTmUDS0 = 00*/
TT0IOC3 = 0x01;     /* platná hrana nastavená snímači ze vstupů TENCm0, */
                    /* TENCm1 bude náběžná */
TT0CE = 1;         /* spuštění časovače po nastavení */
```

Posledním krokem při návrhu firmware bylo nastavení rozhraní pro komunikaci s osobním počítačem. K ladění programu na mikrokontroléru byl použit programátor MINICUBE2, proto není možné použít asynchronní sériové rozhraní UART0.

Využívá se tedy další rozhraní v pořadí UART1, při výběru tohoto rozhraní se však zamezí použití jako alternativního synchronního rozhraní I<sup>2</sup>C. UART1 používá plně obousměrnou komunikaci. Nastavení portu tři, který využívá toto rozhraní se liší od klasického nastavení portu. Ke konfiguraci se používají tři nejnižší bity registrů PMC3, PFC3, PFCE3. Vhodnou kombinací bitů v jednotlivých registrech se nastaví specifická funkce pro piny mikrokontroléru P31 a P30.

Po nastavení portu jsem nastavil pomocí příslušných registrů rozhraní UART1.

**UARTAn control register 0 (UAnCTL0)** – kontrolní registr, který slouží k určení operací, výběru prvního přenášeného bitu (MSB, LSB), výběru parity, nastavení délky přenášených dat a počtu stop bitů. Na začátku nastavení je nutné nastavit bity UA1PWR, UA1TXE, UA1RXE na hodnotu logické „0“, aby bylo možné rozhraní nastavit

**UARTAn control register 1 (UAnCTL1)** – používá se k nastavení vstupních básových hodin, nastavení se provádí pomocí čtyř spodních bitů registru

**UARTAn control register 2 (UAnCTL2)** – pomocí tohoto registru se nastavuje přenosová rychlost sériového rozhraní UARTn, podle rovnice 4.1 je možné spočítat přenosovou rychlost rozhraní z nastavených hodnot registrů UAnCTL2 a UAnCTL1

(Rovnice 4. 1) 
$$\text{Přenosová rychlost} = \frac{f_{UCLK}}{2 * k} \text{ [bps]}$$

$f_{UCLK}$  = frekvence hodin na vstupu

$k$  = hodnota nastavená registrem UAnCTL2

Konkrétní výpočet přenosové rychlosti podle rovnice 4.1 pro NEC V850E/IG3:

$f_{xx} = 8 \text{ MHz}$

$f_{UCLK} = 250 \text{ kHz}$

$k = 13$

$$\text{Přenosová rychlost} = \frac{250000}{2 * 13} = 9615 \text{ bps}$$

**UARTAn option control register 0 (UAnOPT0)** – určuje, zda budou data přijímána stejně, jako byla vysílána (normálně), nebo opačně (invertovaně)

**UARTAn status register (UAnSTR)** – zobrazuje stav přenosu a obsah přijímaných chyb

**UARTAn receive data register (UAnRX)** – data přijímaná UARTn se uloží do posuvného registru (UARTAn receive shift register) a po příjmu jednoho bajtu se uloží do UAnRX registru a na konci tohoto přenosu se vygeneruje přerušení INTUAnR

**UARTAn transmit data register (UAnTX)** – při odesílání se data nejdříve přenesou z registru UAnTX do posuvného registru (UARTAn transmit shift register), odkud se odesílají, po skončení přenosu dat z UAnTX do posuvného registru se vyvolá přerušení INTUAnT

U následující ukázky programu je zobrazeno konkrétní nastavení pinů P30, P31 pro rozhraní UART1 a také vlastní nastavení rozhraní UART1.

```

/* PMC31 = 1, PFC31 = 0, PFC31 = 0 TXDA1 output */
/* PMC30 = 1, PFC30 = 0, PFC30 = 0 RXDA1 input */
PMC3 = 0x03;
PFC3 = 0x00;
PFCE3 = 0x00;
UA1CTL0 = 0x12;      /* přenášení dat od bitu LSB */
                      /* zakázání operací UA2PWR=0, UA2TXE=0, UA2RXE=0 */
                      /* STOP bit délka 1, délka přenášených dat 8 bitů */
                      /* parita při příjmu i odeslání žádná */
UA1CTL1 = 0x04;      /* vstupní frekvence hodin fuclk = fxx/32 */
UA1CTL2 = 0x0D;      /* přenosová rychlost k = 13 */
UA1OPT0 = 0x14;      /* normální příjem i odeslání dat */
/* nastavení priorit přerušeni */
UA1REIC = 0x00;      /* přerušeni při chybě příjmu */
UA1RIC = 0x01;       /* přerušeni při dokončení příjmu */
UA1TIC = 0x02;       /* přerušeni při povolení odeslání */
UA1PWR = 1;          /* povolení operací UARTn */
UA1TXE = 1;          /* povolení operace odeslání */
UA1RXE = 1;          /* povolení operace příjmu */

```

Při vyvolání přerušeni příjmu nebo odeslání se skočí do obslužné procedury ve zdrojovém kódu interrupt.c. Obslužná procedura pro čtení dat vypadá takto:

```

__flat __interrupt void UA1R_isr (void) {      /* označení přerušeni pro příjem dat */
    if (count_rx == RX_SIZE) {                 /* kontrola počtu přijatých znaků, zda buffer není plný */
        count_rx = 0;                           /* nuluje se index v poli a data se zapisují od počátku */
    }
    buf_rx[count_rx] = UA1RX;                   /* ukládání znaků do pole k dalšímu zpracování */
    count_rx++;
}

```

Obslužná procedura odeslání, která se vyvolá po zapsání dat do UA1TX:

```

__flat __interrupt void UA1T_isr (void) {      /* označení přerušeni pro odesílání dat */

```

```

int i = 0;                               /* pomocná proměnná */
if(count_tx == TX_SIZE){                 /* kontrola počtu odeslaných znaků zda buffer není plný */
    count_tx = 0;                         /* nuluje se index v poli a data se zapisují od počátku */
    for (i; i < 65535; i++){__NOP();}     /* zpoždění z důvodu dokončení přenosu */
    UA1PWR = 1;                           /* povolení operací UARTn */
    UA1TXE = 0;                           /* zakázání operace odeslání */
    UA1RXE = 1;                           /* povolení operace příjmu */
}
}

```

Poté co jsou všechna rozhraní, čítače/časovače a porty nastaveny, program skočí do nekonečné smyčky. V této smyčce se vykonávají příchozí příkazy a volají se jednotlivé funkce.

**void rychlost(void)** – v této funkci se nastavuje požadovaná rychlost otáčení motorů změnou registrů TAB0CCR1, TAB0CCR3

**void kontrola\_zavor(void)** – kontrola, zda snímaný prostor světelné závory je prázdný, pokud ano, zablokuje jeden nebo oba povolovací piny výkonových obvodů a motory se zastaví

**void vyber\_smeru(void)** - vybere směr otáčení povolením vhodného výstupu časovače TAB a příslušného ENAB\_1A, ENAB\_1B, při startu programu je vše vypnuté

**void najeti(void)** – slouží k najetí na počátek souřadnic, tedy na [0,0]

**void nacteni\_hodnoty(void)** – odešle počet pulzů, které se načítají po najetí na vzdálenost jednoho centimetru v ose X

**void zadana\_hodnota(void)** – načte hodnotu, která určuje počet pulzů potřebných na jeden milimetr, z této hodnoty následně vychází najetí na pozici

**void najeti\_na\_pozici(void)** – najede na souřadnice zadané v uživatelském rozhraní, pamatuje si předchozí zadanou hodnotu a při zadání nové hodnoty je porovná a podle výsledku určuje směr otáčení a tím i hodnotu pro čítání pulzů

**void sekvence(void)** – tato funkce na základě nastavených hodnot posouvá vzorek po krocích. Podle konkrétní pozice je několik možností pohybu stolku se vzorkem. Pokud je na poslední pozici v ose X, spustí se návrat na nulovou pozici. Po dojetí na tuto pozici se posune v ose Y, pokud je i na této ose v poslední pozici, vrátí se nejdříve osa X do nulové pozice a následně tu samou činnost vykoná v ose Y.

Kompletní zdrojový kód firmwaru je součástí přiloženého CD.

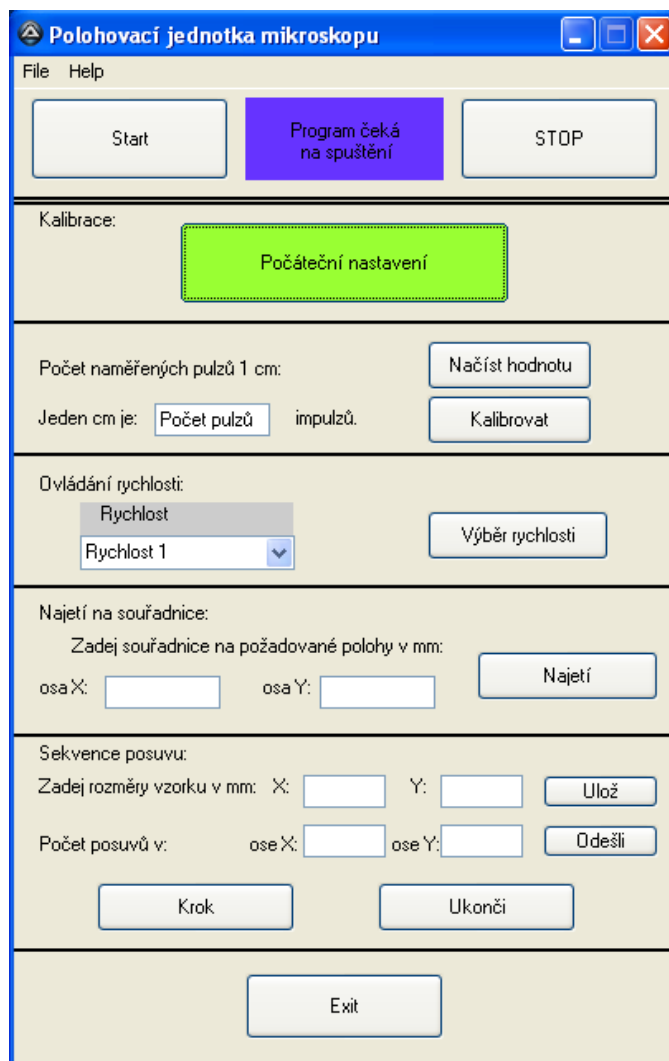
#### 4.4. Návrh uživatelského rozhraní pro osobní počítač

Uživatelské rozhraní bylo vytvořeno ve volně šiřitelném programu AutoIt [19], který je navržen k tvorbě uživatelských rozhraní a jiných skriptů pro Windows. Ve verzi 3.3.8.1 programu AutoIt byl upraven skriptovací jazyk, aby se co nejvíce podobal jazykům ze skupiny BASIC. Po dokončení celého skriptu je možné vytvořit spustitelný soubor. Není tedy nutné mít AutoIt nainstalovaný přímo v počítači. Tento jazyk byl vybrán, protože je schopen používat mnoho doplňků a modulů pro konkrétní aplikace. Může lehce odkazovat na knihovny uložené v systému Windows. AutoIt má nejen velmi dobrou on-line podporu [20] od výrobce, ale také je to dosti rozšířený způsob skriptování mezi uživateli.

Výstupní port komunikace je nastavený na COM1. V uživatelském rozhraní nelze tento port měnit. Jeho změna je možná pouze ve zdrojovém kódu skriptu. Funkce nastavující port, která bude použita, se nazývá Dynamické volání funkcí v DLL a její formát je „*DllCall ("dll", "návrst typ", "funkce" [, Type1, param1 [, typ N, param n]])*“. Volaná knihovna je Kernel.dll. Pátým parametrem funkce je komunikační port, v tomto případě \\.\COM1. Po změně portu ve zdrojovém kódu je nutné provést kompilaci a nově vytvořit spouštěcí soubor.

Při vlastním návrhu uživatelského rozhraní jsem několikrát měnil jeho vzhled a uspořádání v závislosti na funkcích, kterými polohovací jednotka disponovala. Snažil jsem se, aby rozhraní bylo přívětivé k uživateli, proto jsem všechny funkce umístil do jednoho společného okna. V jednotlivých krocích kalibrace a řízení informuje uživatele dodatečnými okny o průběhu činnosti nebo se ujišťuje, zda jsou zadávané hodnoty správné. V případě zadání nevyhovujících hodnot poradí uživateli, které hodnoty jsou správné.

Navržené uživatelské rozhraní zobrazené na obrázku 4.4 je rozdělené do sedmi částí, které na sebe postupně navazují. První část obsahuje tlačítko „Start“, informační pole a tlačítko „STOP“. Po stisku tlačítka „Start“ se odešle příkaz a nastaví se hodnoty mikrokontroléru tak, aby bylo možné začít s kalibrací a následným posuvem vzorku. Informační pole se mění v závislosti na běhu programu a radí, co je nutné provést v dalším kroku používání. Tlačítko „Stop“ slouží k zastavení programu.



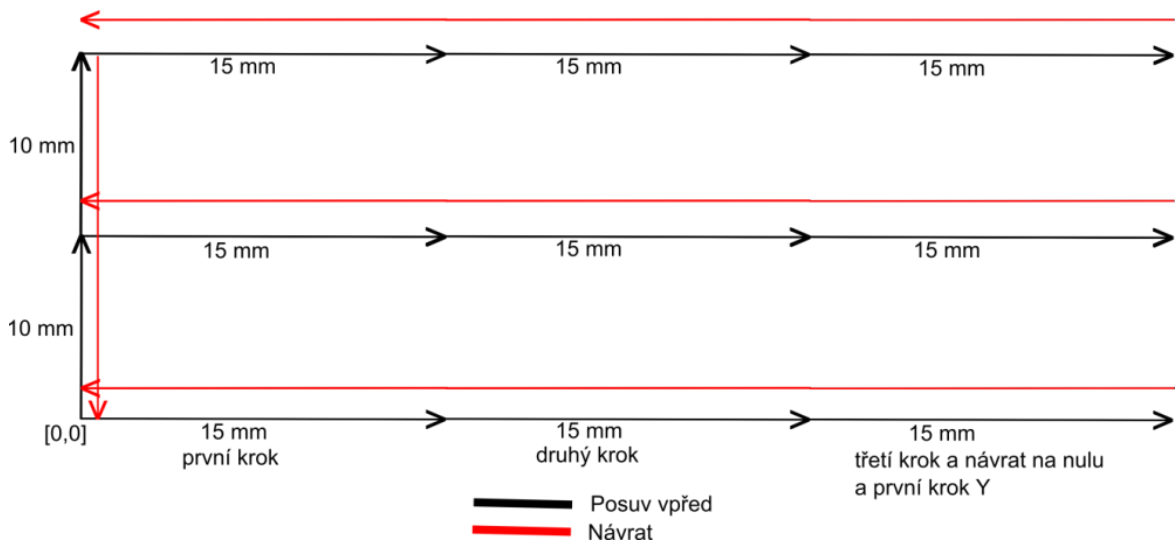
Obr. 4.6: Uživatelské rozhraní

Druhá a třetí část uživatelského rozhraní slouží ke kalibraci programu s křížovým stolek. Nejdříve je nutno najet do počáteční pozice, protože mikrokontrolér po odpojení napájení ztratí informace o poslední pozici vzorku. K tomuto účelu slouží tlačítko „Počáteční nastavení“. Po jeho stisku se nejdříve křížový stolek najede do pozice nula v ose X a poté v ose Y. Z počátečních souřadnic [0,0] je možné pokračovat v kalibraci. Dalším krokem je odměření 10 milimetrů v ose X. Uživatel najede na tuto vzdálenost, stiskne tlačítko „Načíst hodnotu“ a zobrazí se mu počet pulzů, které se vygenerovaly v mikrokontroléru. Toto číslo se запиše do políčka „Počet pulzů“ a poté uživatel stiskne tlačítko „Kalibrovat“, tím se odešlou data zpět k mikrokontroléru. Kalibrace je ukončena a je možno pracovat s jednotkou. Postup kalibrace je celek, který je závislý na vnitřních hodnotách mikrokontroléru, z tohoto důvodu je nutné přesně dodržet postup kalibrace.

V části čtyři se nastavuje rychlost otáčení motoru. Je možné vybrat si z deseti rychlostí otáčení motorů. Po zvolení vhodné rychlosti je nutné potvrdit ji tlačítkem „Výběr rychlosti“. Tuto hodnotu lze v průběhu činnosti měnit. Po výběru rychlosti je možné pohybovat vzorkem pomocí šipek na klávesnici osobního počítače.

Část pět slouží k najetí na souřadnice zadané uživatelem. Hodnoty jsou v milimetrech. Maximální hodnoty jsou omezeny velikostí pracovního stolku mikroskopu, v ose X je to 100 mm a v ose Y 50 mm. Nastavení v částech čtyři a pět je možné libovolně kombinovat.

Část šest slouží k sekvenčnímu snímání vzorků. Postupně se uloží hodnoty potřebné k tomuto pohybu. Nutné jsou rozměry vzorku, ze kterých se určuje vzdálenost, o kterou se bude vzorek posouvat v ose X a Y. Po zadání těchto hodnot se stiskne tlačítko „Ulož“ a hodnoty se odešlou do mikrokontroléru. Další dvě políčka slouží k zadání počtu posuvů v obou osách a opět je nutné hodnoty odeslat mikrokontroléru. Po zadání všech potřebných hodnot a stisku tlačítka „Krok“ se vždy vykoná jeden krok. Po dokončení kroků v ose X se automaticky vrátí na počátek, posune se o jeden krok v ose Y a prochází znovu řádku X. Po dojetí na konec sekvence se automaticky vrátí do počátku soustavy souřadnic. Na obrázku 4.7 je znázorněna sekvence posuvu pro vzorek o rozměrech 15x10 mm a s počtem posuvů v ose X = 3 a v ose Y = 2.



Obr. 4.7: Posuvy při sekvenčním snímání

V poslední sedmé části je umístěno tlačítko „Exit“, pomocí kterého se celé uživatelské rozhraní ukončuje.

Podrobný popis uživatelského rozhraní, včetně ukázek všech funkcí, viz Příloha V. - Uživatelský manuál. Kompletní skript uživatelského rozhraní i se spouštěcím souborem je uložen ve složce *Uživatelské\_rozhraní* na přiloženém CD.

#### 4.5. Komunikace osobní počítač ← → mikrokontrolér

Komunikace probíhá zasláním jednoduchých sekvencí znaků mezi těmito zařízeními. Maximální délku přenášené zprávy jsem nastavil na hodnotu šest na straně mikrokontroléru, což dostačuje mému použití.

Uživatelské rozhraní používá k odeslání znaků funkci *writeString()*, která využívá ke své funkci *writeChar()* a odesílá postupně ve smyčce jednotlivé znaky příkazu. Z důvodu různé délky přenášených příkazů bylo nutné doplnit některé příkazy pomocnými znaky pro jejich další použití. K příjmu znaků se využívá funkce *readLineBlock()*, která opět volá funkci *readCharBlock()* a v cyklu čte jednotlivé znaky přijímané od mikrokontroléru.

Mikrokontrolér přijímá data přes UART1 do registru UA1RX a následně je ukládá do *buf\_rx[count\_rx]*, se kterým následně pracuje program uvnitř mikrokontroléru. Po přijetí těchto znaků kontroluje obsah *buf\_rx* a pokud se přijaté znaky shodují s jednou z nadefinovaných sekvencí znaků, vykoná se daná část programu.

Konkrétní příkazy, které vysílá uživatelské rozhraní mikrokontroléru, jsou následující:

- Po stisku tlačítka **Start** se odešle příkaz „**goo###**“
- Po stisku tlačítka **Stop** se odešle příkaz „**sto###**“
- Po stisku tlačítka **Počáteční nastavení** se odešle příkaz „**poc###**“
- Po stisku tlačítka **Načíst hodnotu** se odešle příkaz „**kal###**“
- Po stisku tlačítka **Kalibrovat** se odešle příkaz „**hodnota1mmi**“
  - o Např.: 1234i
- Po stisku tlačítka **Výběr rychlosti** se odešle příkaz „**rx#####**“
  - o rx označuje konkrétní rychlost vybranou z nabídky deseti možných, hodnota rychlosti se zadává v HEX formátu
  - o poté Rychlost 1 je „**r1#####**“, Rychlost 2 je „**r2#####**“ až Rychlost 10 je „**ra#####**“

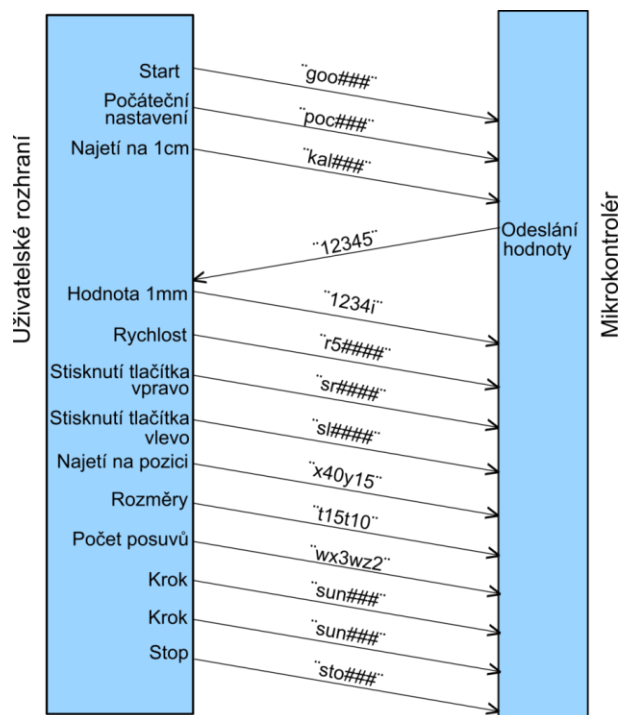


- Po stisku tlačítka **Najetí** se odešle příkaz “xAAyBB“
  - o souřadnice pro najetí jsou zadávány v milimetrech a formát zadávaných hodnot je dekadický, maximální hodnota je tedy 99x99mm
  - o AA označuje zadanou hodnotu do políčka osa X:
  - o BB označuje zadanou hodnotu do políčka osa Y:
- Po stisku **šipky nahoru** se odešle příkaz “su#####“
- Po stisku **šipky dolů** se odešle příkaz “sd#####“
- Po stisku **šipky vpravo** se odešle příkaz “sr#####“
- Po stisku **šipky vlevo** se odešle příkaz “sl#####“
- Po uvolnění **jakékoliv šipky** se odešle příkaz “sq#####“
- Po stisku tlačítka **Ulož** se odešle příkaz “tAAAtBB“
  - o rozměry vzorku jsou zadávány v milimetrech a formát zadávaných hodnot je dekadický, maximální hodnota je tedy 99x99 mm
  - o AA jsou rozměry v ose X
  - o BB jsou rozměry v ose Y
- Po stisku tlačítka **Odešli** se odešle příkaz “vxAvyB“
  - o počet posuvů je bezrozměrná veličina a formát zadávaných hodnot je dekadický, maximální hodnota je tedy 9 posuvů v ose X i v ose Y
  - o A počet posuvů v ose X
  - o B počet posuvů v ose Y
- Po stisku tlačítka **Krok** se odešle příkaz “sun###“
- Po stisku tlačítka **Ukončit** se odešle příkaz “ukonci“

V příloze V. je tabulka příkazů, které se odesílají po stisku jednotlivých tlačítek.

Když mikrokontrolér přijme nějaký z těchto příkazů, vykoná činnost, která je pro něho naprogramovaná. Potvrzování dokončeného úkonu není implementováno, správnou činnost je možné vizuálně ověřit v průběhu činnosti polohovací jednotky. Pouze při přijetí příkazu “kal###“, který požaduje kalibrační hodnotu jednoho centimetru, odešle mikrokontrolér data obsahující počet pulzů vygenerovaných po najetí na pozici jednoho centimetru v ose X.

Na obrázku 4.8 je znázornění ukázky komunikace.



Obr. 4.8: Ukázka komunikace MCU a uživatelského rozhraní

## 5. Realizace a ověření praktické použitelnosti

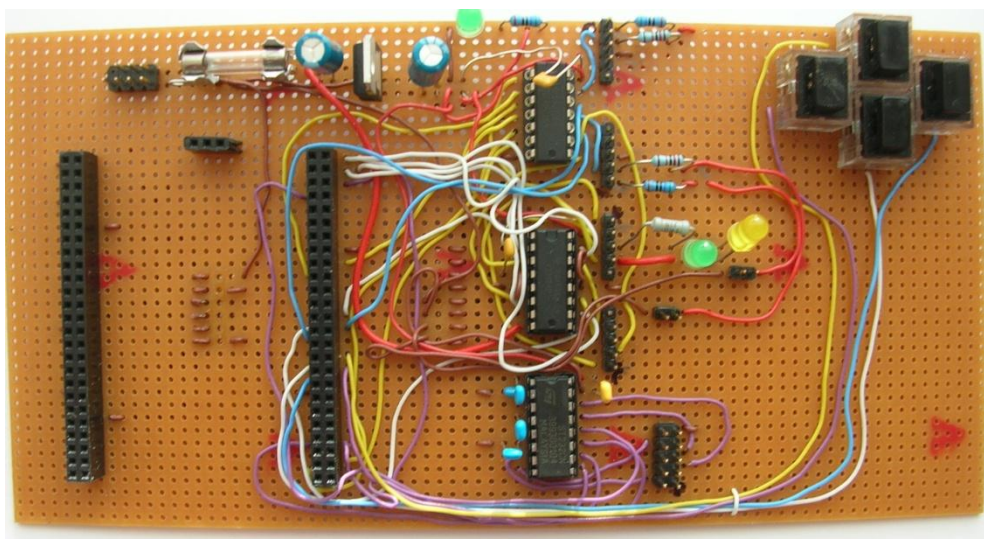
V počátcích tvorby firmware, nastavování časovačů, portů a kontrole výstupů polohovací jednotky ke kontrole výstupů a testování jednotky stačila deska osazená mikrokontrolérem (viz obrázek 5.1), vývojové prostředí s možností ladění a osciloskop, nebo logický analyzátor. Při prvním testování a kontrolách výstupů na osciloskopu byl zobrazovaný signál zašuměný, to bylo způsobeno připojením na nulový potenciál analogové části mikrokontroléru. Po připojení na napájecí nulový potenciál destičky mikrokontroléru šum zmizel.



Obr. 5.1: Deska osazená MCU NEC V850E/IG3

Po nastavení všech registrů a kontrole správnosti výstupů na pinech mikrokontroléru s teorií bylo nutné vytvořit desku s ovládacími tlačítky, pomocnými obvody řízení a obvody, na které budou propojeny další periferie.

Po výběru všech potřebných obvodů bylo navrženo schéma zapojení. Návrh plošného spoje, jeho kontrola a výroba je časově dosti náročná. Souběžně s návrhem plošného spoje běžela také tvorba testovací desky na univerzálním plošném spoji, který měl motiv trojice propojených otvorů. Z tohoto důvodu bylo nutné univerzální plošný spoj v jedné části upravit, aby bylo možné osadit patice pro mikrokontrolér. S osazením ostatních integrovaných obvodů a součástek již nebyl problém. Na desce je též několik LED, které indikují chod testovací desky. Na obrázku 5.2 je výsledná testovací deska bez osazeného mikrokontroléru. V levé části je umístěna patice pro mikrokontrolér, nad ní je napájecí konektor s filtračními a blokovacími kondenzátory, stabilizátor a LED indikující napájení desky. Napravo od patice mikrokontroléru se nachází trojice použitých integrovaných obvodů a konektory k připojení motorů, enkodérů, světelných závor a komunikačního rozhraní RS232. Dvojice diod (žlutá, zelená) indikuje směr otáčení. Na této testovací desce jsou osazeny pouze u jednoho motoru, ale ve výsledném návrhu jsou použity u obou motorů. Na pravém okraji jsou tlačítka, která byla použita k testování. Pasivní součástky k nim nejsou připojeny. Ve finálním návrhu je deska osazena konektorem pro připojení desky s tlačítky, která již pasivními součástkami bude osazena.



Obr. 5.2: Testovací deska

Poté co byla dokončena realizace testovací desky, byly připojeny testovací motory a začalo se s jejich řízením. Ovládání bylo v počátku řízeno hardwarovými tlačítky umístěnými na desce. Při správném chodu motorů bylo možné připojit enkodéry a testovat dekodování jejich signálů. V průběhu tvorby firmware pro polohovací jednotku byly čítač/časovače experimentálně nastaveny. V průběhu testování se ukázalo, že zvolené nastavení není vhodné, proto bylo nutné opětovné nastavení.

Při správném chodu jednotky bylo vytvořeno uživatelské rozhraní v osobním počítači, připojeno komunikační rozhraní k mikrokontroléru a mohlo se začít s testováním a kontrolou správného nastavení komunikace. Největším problémem bylo nastavení přenosové rychlosti dat na sériové lince, které je 9600 bps. Přijímané znaky byly buď nulové, nebo nesmyslné. Potom byly pomocí rovnice 4.1 vypočítány hodnoty, které je nutné nastavit, a komunikace již fungovala správně.

Když jednotlivé části správně fungovaly, byla jednotka testována jako celek. Bohužel testování neproběhlo přímo na křížovém stolku, protože nebyla navržena mechanická součástka, která by motory připevnila k osám křížového stolku.

## 6. Závěr

Cílem diplomové práce bylo navrhnout a realizovat plošný spoj a firmware polohovací jednotky pro mikroskop, která je řízena po sériové lince uživatelským rozhraním spuštěným v osobním počítači. Výsledná realizace by neměla mít příliš velké rozměry z důvodu uživatelského použití.

V teoretické části návrhu jsem pročítal katalogové listy a uživatelské manuály různých obvodů k zjištění jejich vlastností. Prostudoval jsem uživatelský manuál zvoleného mikrokontroléru NEC V850E/IG3, který disponuje vhodnými vstupy, výstupy a funkcemi k řízení motorů. Na základě požadavků a informací získaných z dokumentace jsem vybral integrované obvody nutné pro správnou funkci celé polohovací jednotky, tři varianty akčních členů vhodných k použití, které se však liší svými vlastnostmi a parametry.

V praktické části bylo vytvořeno schéma zapojení obvodu a návrh plošného spoje polohovací jednotky a návrh komunikace po sériové lince mezi polohovací jednotkou a uživatelským rozhraním. Na základě navržené komunikace byl vytvořen firmware pro mikrokontrolér a uživatelské rozhraní pro osobní počítač.

Všechny body zadání se podařilo splnit, ovšem z časových důvodů nebylo možné vyrobit a osadit výsledný plošný spoj. Výsledkem práce tedy je firmware běžící uvnitř mikrokontroléru, uživatelské rozhraní pro osobní počítač a testovací plošný spoj, který je osazen na univerzálním plošném spoji, na kterém jsem ověřil funkčnost celé jednotky, navrženého firmware a komunikace s rozhraním v osobním počítači.

## Literatura

- [1] Renesas Electronics, Manuál NEC V850E/IG3 [online]. 2012, [cit. 15. 2. 2012].  
Dostupné z URL: < [http://www.renesas.eu/req/product\\_document\\_lineup\\_child.do?REGION\\_KEY=4&LAYER\\_KEY=32306&PDF\\_URL=http://documentation.renesas.com/doc/DocumentServer/U18279EJ3V0UD00.pdf&TKUPDATE=true&APNO TE=true](http://www.renesas.eu/req/product_document_lineup_child.do?REGION_KEY=4&LAYER_KEY=32306&PDF_URL=http://documentation.renesas.com/doc/DocumentServer/U18279EJ3V0UD00.pdf&TKUPDATE=true&APNO TE=true) >
- [2] Dalekohledy a mikroskopy, Křížový stolek [online]. 2002,[cit. 12. 5. 2012].  
Dostupné z URL: < <http://www.dalekohledy.com/JAPE/goods-5942600-28-krizovy-stolek-pro-mikroskop.html>>
- [3] Robotika.cz, Krokové motory [online]. 28. 10. 2002, [cit. 15. 5. 2012]. Dostupné z URL: <<http://robotika.cz/articles/steppers/en>>
- [4] NEC Electronics America, NEC V850 32-Bit Microcontrollers [online]. 2007, [cit. 14. 5. 2012]. Dostupné z URL: <<http://documentation.renesas.com/doc/DocumentServer/U18589EJ1V0PFU1.pdf>>
- [5] Texas Instruments, Katalogový list L293 [online]. 6. 2002,[cit. 15. 5. 2012].  
Dostupné z URL: < <http://www.datasheetcatalog.org/datasheet/texasinstruments/l293d.pdf>>
- [6] MAXIM, Katalogový list MAX232 [online]. 1. 2006,[cit. 15. 5. 2012]. Dostupné z URL: < <http://www.datasheetdir.com/MAXIM-MAX232+download>>
- [7] Maxon motor, Maxon motor RE16 [online]. 2012,[cit. 16. 6. 2012]. Dostupné z URL: < <http://www.maxonmotor.com/maxon/view/product/motor/dcmotor/re/re16/320177> >
- [8] Maxon motor, Maxon převodovka GP 16 [online]. 6. 2002,[cit. 16. 6. 2012].  
Dostupné z URL: < <http://www.maxonmotor.com/maxon/view/product/gear/planetary/gp16/134781> >
- [9] Maxon motor, Maxon enkodér MR, typ M [online]. 6. 2002,[cit. 16. 6. 2012].  
Dostupné z URL: < <http://www.maxonmotor.com/maxon/view/product/sensor/encoder/imp32/201935> >
- [10] SGTCL, Motor F-GM12-N20VA110 [online]. 2003,[cit. 17. 6. 2012]. Dostupné z URL: < <http://www.sgstcl.com/html/eng/prod/pop.asp?id=32> >
- [11] Alibaba.com, Motor F-GM12-N20VA110 s enkodérem [online]. 2012, [cit. 17. 6. 2012]. Dostupné z URL: < [http://www.alibaba.com/product-gs/430698780/12mm\\_12v\\_dc\\_gear\\_motor\\_Can.html](http://www.alibaba.com/product-gs/430698780/12mm_12v_dc_gear_motor_Can.html) >

- [12] Lynxmotion, Gear Head Motor GHM-04 [online]. 2010, [cit. 17. 6. 2012].  
Dostupné z URL: < <http://www.lynxmotion.com/p-96-gear-head-motor-72vdc-501-175rpm-6mm-shaft.aspx> >
- [13] Lynxmotion, Quadrature Motor Encoder [online]. 2010, [cit. 17. 6. 2012].  
Dostupné z URL: < <http://www.lynxmotion.com/p-448-quadrature-motor-encoder-wcable.aspx> >
- [14] Vishay Semiconductor, Světelná závora TCST, [online]. 8. 6. 1999,  
[cit. 18. 6. 2012] Dostupné z URL: < <http://www.gme.cz/dokumentace/521/521-043/dsh.521-043.1.pdf> >
- [15] National Semiconductor, Hysterezní člen 74HC14, [online]. 1995,  
[cit. 18. 6. 2012]. Dostupné z URL: < <http://www.bkbelectronics.com/pdf%20files/74HC14.pdf> >
- [16] MCU.cz, Komunikace RS232 [online]. 13. 10. 2002, [cit. 18. 6. 2012]. Dostupné  
z URL: < <http://mcu.cz/print.php?news.231> >
- [17] GME, Plastový Box [online]. 2012, [cit. 17. 6. 2012]. Dostupné  
z URL: < <http://www.gme.cz/plastove-krabicky-pristrojove/u-km35-p622-109/> >
- [18] Renesas Electronics, Examples [online]. 2012, [cit. 20. 6. 2012]. Dostupné z URL:  
< [http://am.renesas.com/products/mpumcu/v850/V850eIx/v850eif3\\_ig3/app\\_notes.jsp](http://am.renesas.com/products/mpumcu/v850/V850eIx/v850eif3_ig3/app_notes.jsp) >
- [19] Jonathan Bennett, AutoIt [online]. 2012, [cit. 19. 6. 2012] Dostupné z URL:  
< <http://www.autoitscript.com/site/autoit/> >
- [20] Jonathan Bennett, AutoIt function [online]. 2012, [cit. 19. 6. 2012] Dostupné  
z URL: < <http://www.autoitscript.com/autoit3/docs/functions.htm> >

## **Přílohy**

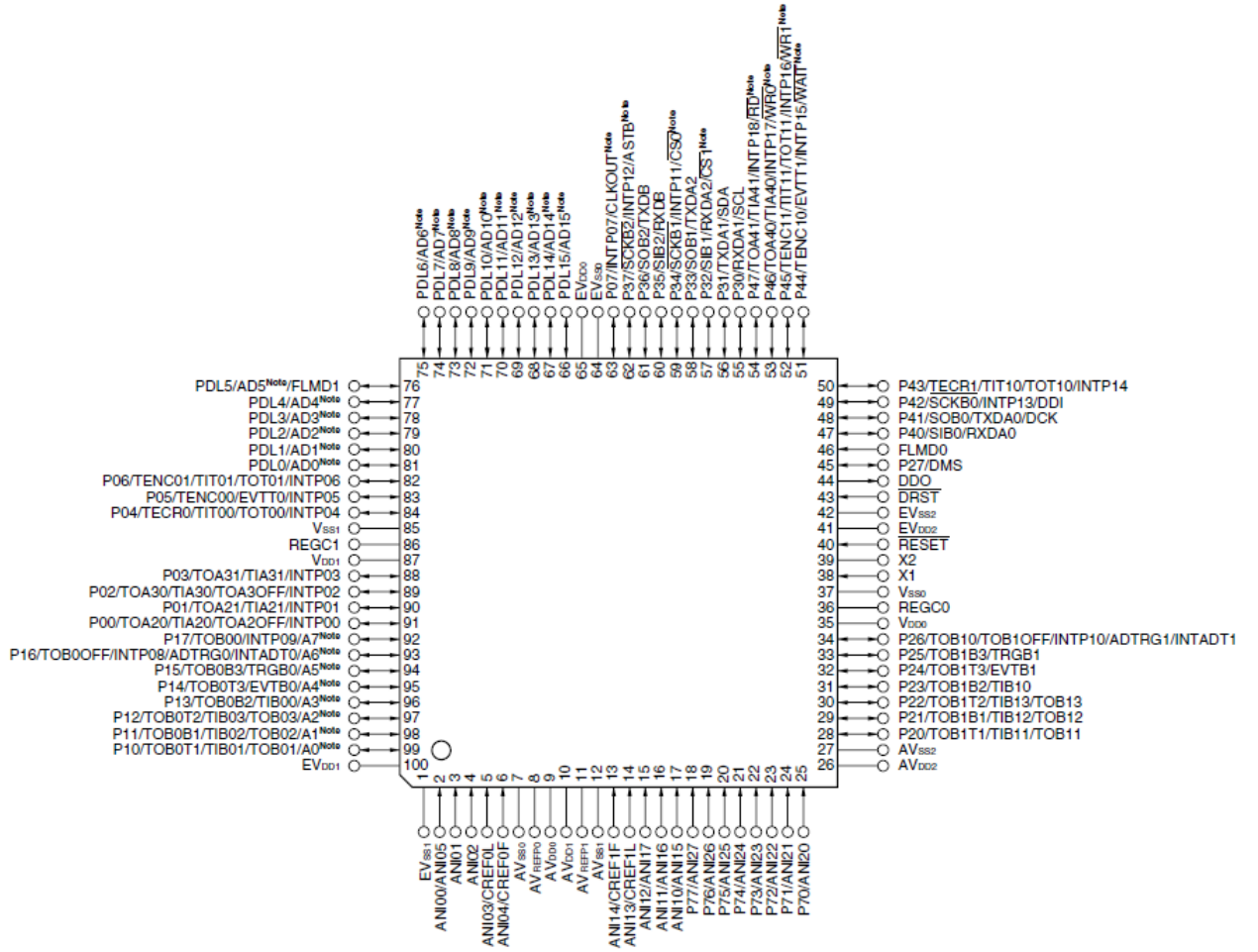
### **Seznam příloh:**

- I. – Popis pinů mikrokontroléru NEC V850E/IG3
- II. – Schéma zapojení
- III. – Plošný spoj
  - a) vrstva A
  - b) vrstva B
  - c) vrstva s popisy
- IV. – Osazovací schéma
- V. – Tabulka příkazů odesílaných po stisku tlačítek
- VI. – Uživatelský manuál
- VII. – CD

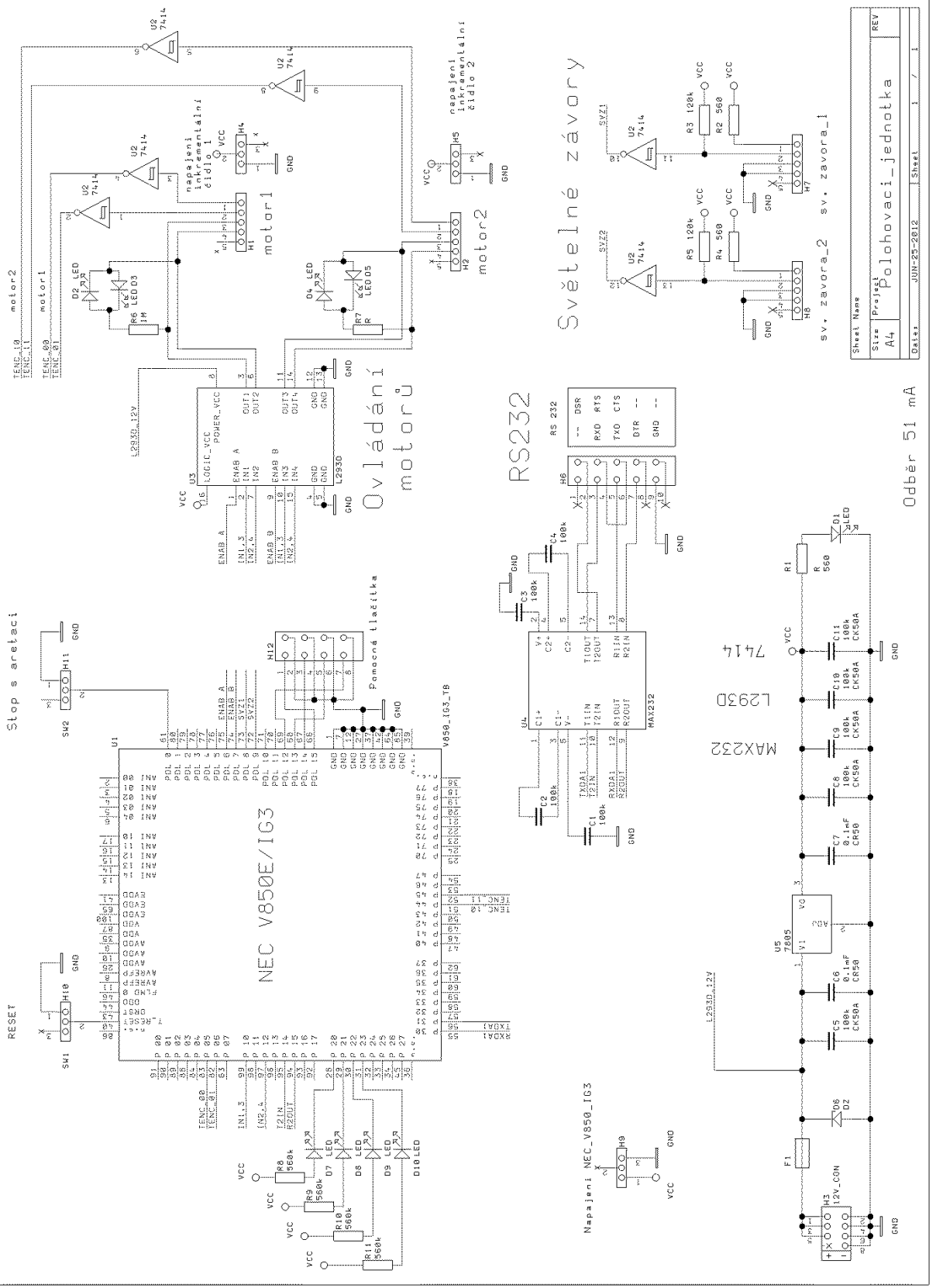


## **Příloha I. – Popis pinů mikrokontroléru NEC V850E/IG3**

Top view



## **Příloha II. – Schéma zapojení**



Odber 51 mA

Sheet Name	Polohovaci_jednotka
Size	A4
Project	REV
Date	JUN-23-2012
Sheet	1 / 1

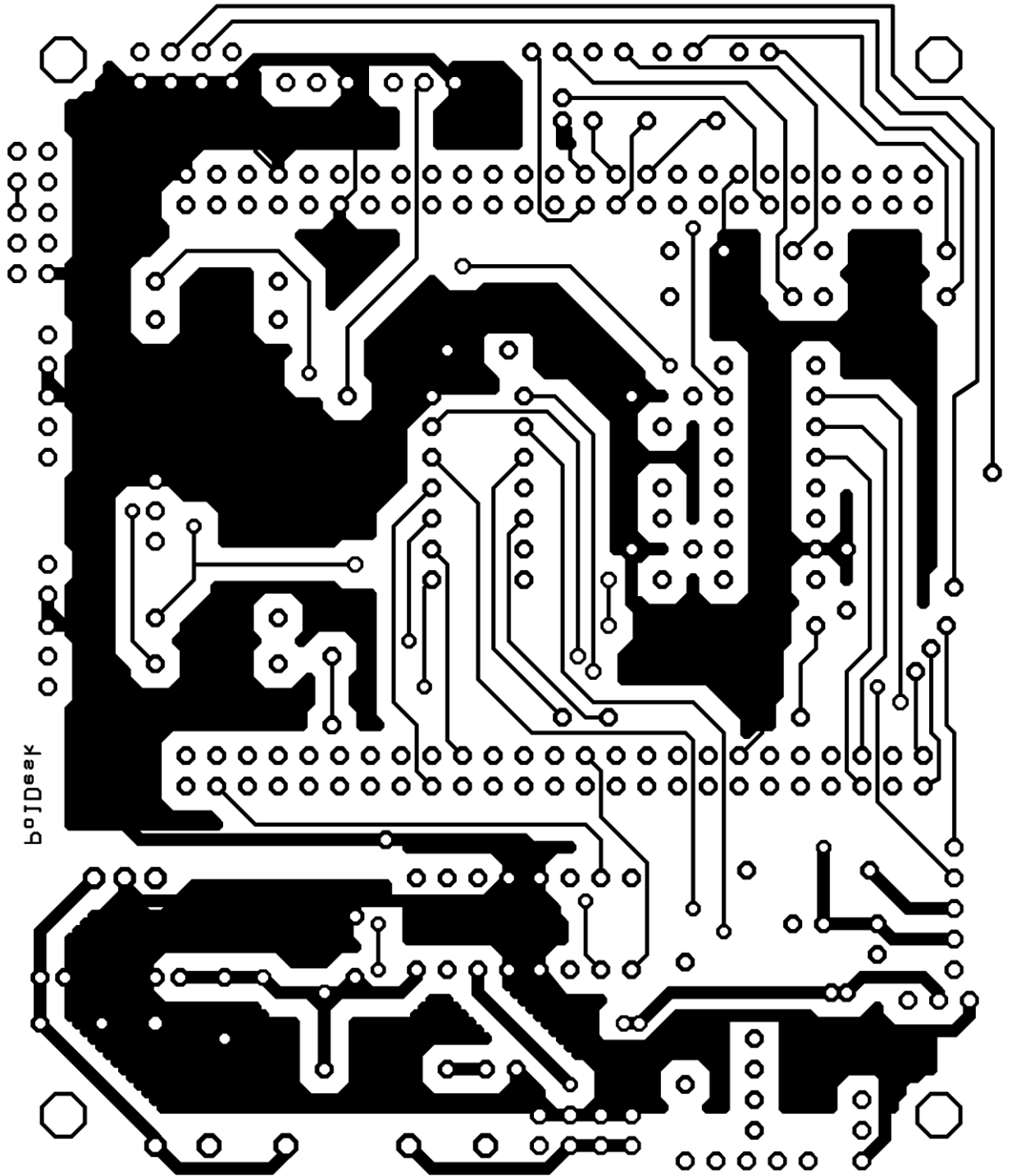
### **Příloha III. – Plošný spoj**

**a) Vrstva A**

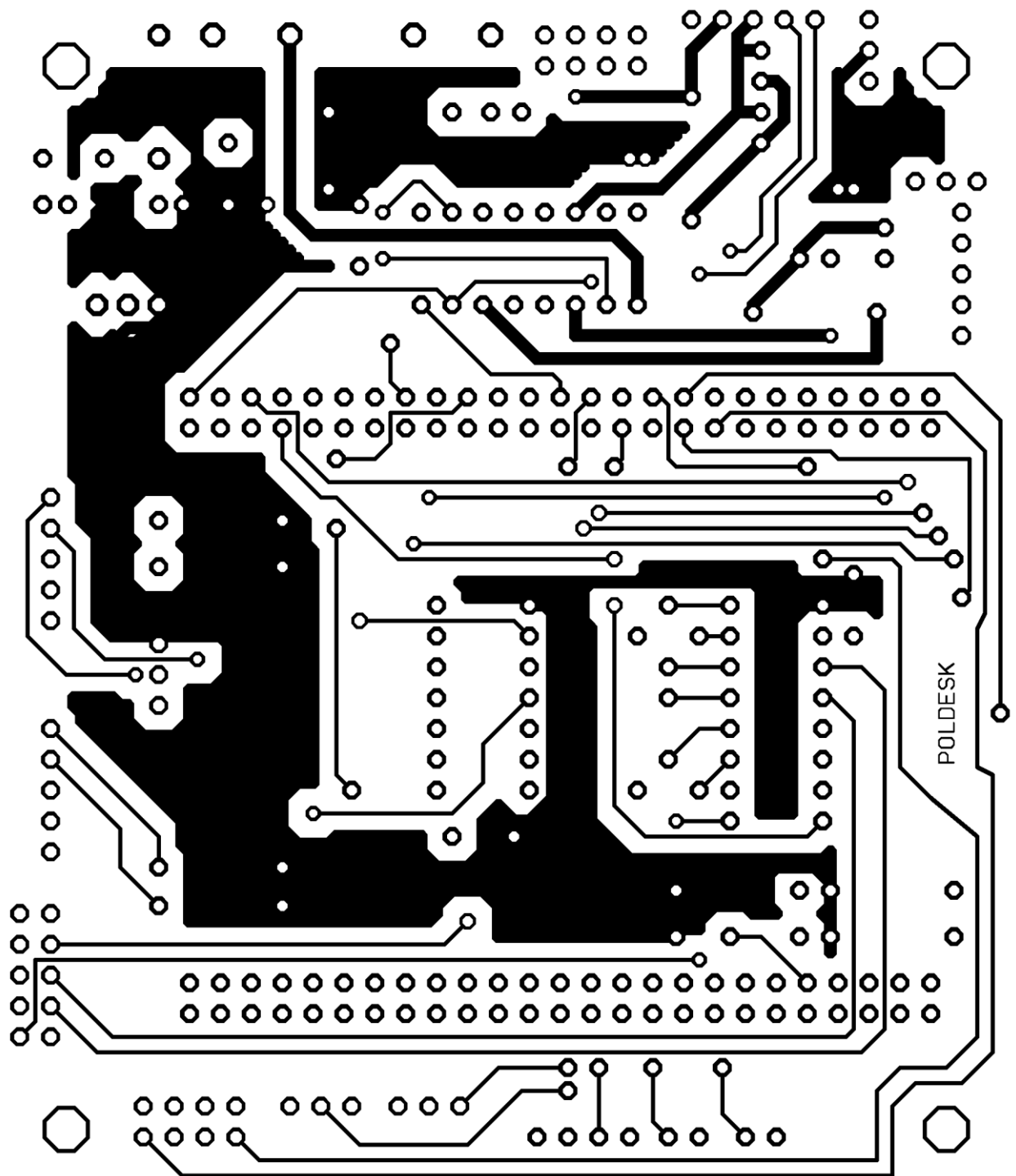
**b) Vrstva B**

**c) Vrstva s popisem a obrysem**

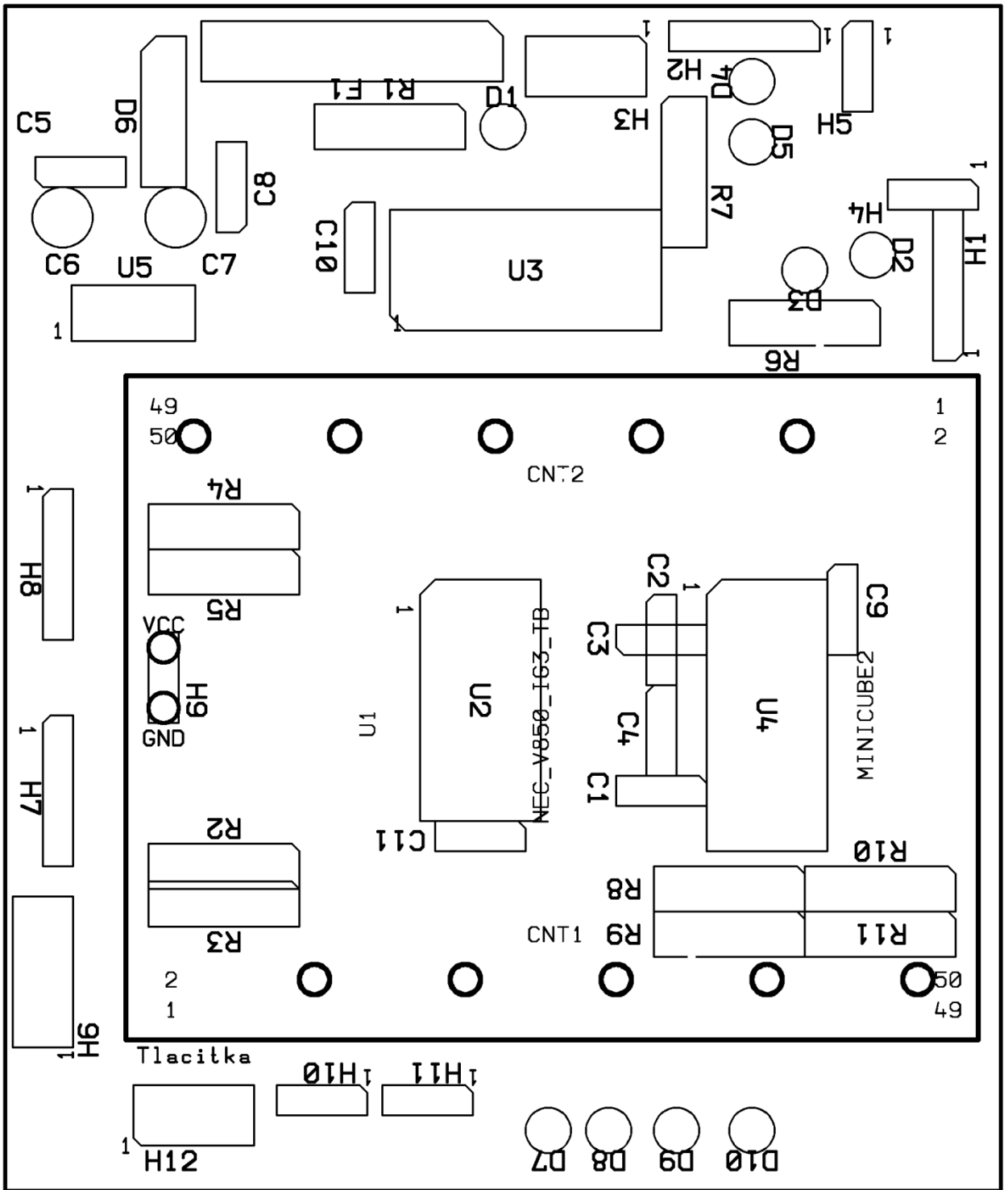
a) Vrstva A v poměru 2:1



b) Vrstva B v poměru 2:1

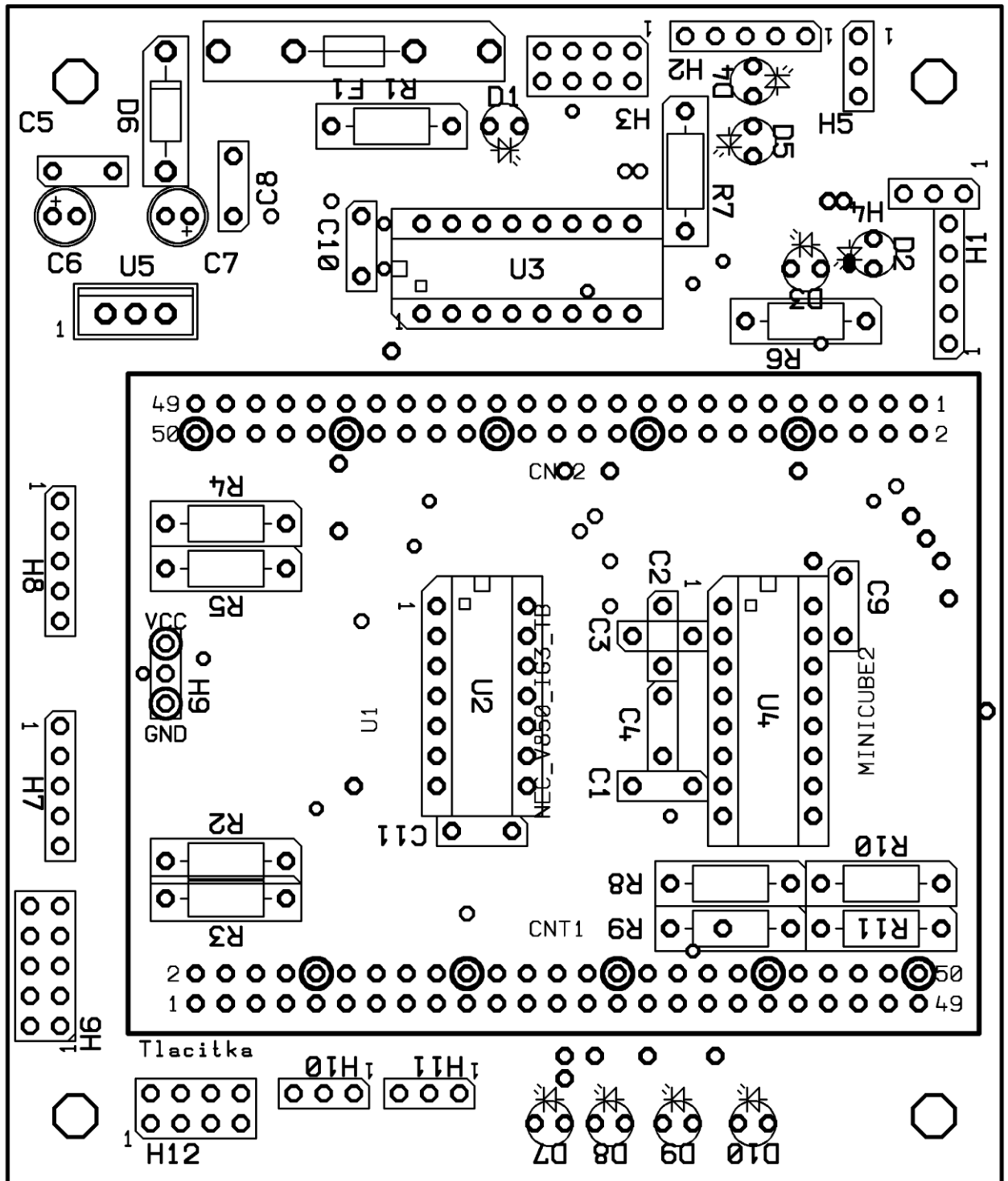


a) Vrstva s popisky a obrysem B v poměru 2:1





Příloha IV. – Osazovací schéma B v poměru 2:1



## **Příloha V. – Tabulka příkazů odesílaných po stisku tlačítek**

<b>tlačítko</b>	<b>příkaz</b>	<b>příklad</b>
Start	"goo###"	
Stop	"sto###"	
Počáteční nastavení	"poc###"	
Načíst hodnotu	"kal###"	
Kalibrovat	"hodnota lmmi"	"1234i"
Výběr rychlosti	"rx#####"	Rychlost1 "r1#####"
Najetí	"xAAYBB"	"x15y10"
Šipka nahoru	"su#####"	
Šipka dolů	"sd#####"	
Šipka vpravo	"sr#####"	
Šipka vlevo	"su#####"	
Uvolnění šipky	"sq#####"	
Ulož	"tAAAtBB"	"t20t12"
Odešli	"vxAvyB"	"vx4vy2"
Krok	"sun###"	
Ukončit	"ukonci"	

## **Příloha VI. – Uživatelský manuál**

- **Uživatelské rozhraní polohovací jednotky pro osobní počítač**

# Uživatelský manuál

Uživatelské prostředí ovládací jednotky pro mikroskop.

28.6.2012

Západočeská Univerzita Plzeň

Pohanka Tomáš

# Úvod

Aplikace je navržena jako součást diplomové práce zadané katedrou Informatiky a Výpočetní techniky Západočeské univerzity v Plzni s názvem Polohovací jednotka pro mikroskop.

Slouží k automatizovanému ovládní polohy zkoumaného vzorku umístěného v křížovém pracovním stolku. Skript je vytvořen pomocí programu AutoIt, použité vývojové prostředí je SciTE4AutoIt3. Celý program, ve kterém je aplikace vytvořena, si uživatel může volně stáhnout ze stránek výrobce [19].

V manuálu se dočtete jak nastavit komunikační kanál, kalibrovat jednotku s konkrétním křížovým stolkem a jak postupovat při vlastním řízení posuvů vzorku po pracovním stolku mikroskopu.

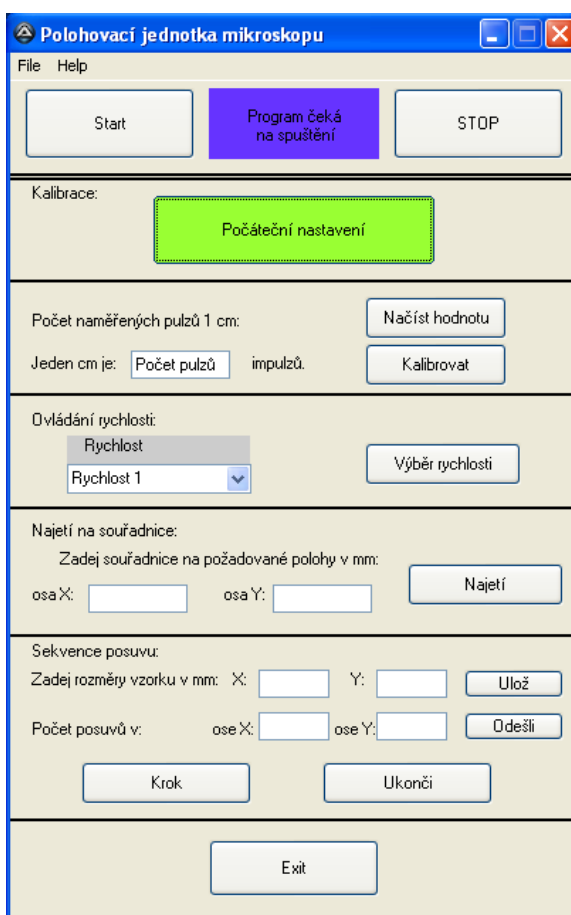
## **Obsah**

- 1. Nastavení komunikačního portu**
- 2. Kalibrace**
- 3. Najetí na pozici**
- 4. Sekvence snímaných vzorků**
- 5. Menu programu**

## 1. Nastavení komunikačního portu

Polohovací jednotka pro mikroskop komunikuje s osobním počítačem, na kterém je spuštěna aplikace uživatelského rozhraní standardně po sériovém portu RS232, konkrétně COM1. Změnu portu je možné provést pouze v zdrojovém skriptu uživatelského rozhraní nazvaném *Polohovací\_jednotka\_pro\_mikroskop.au3*, který je umístěn ve složce *uživatelské\_rozhraní* na příloženém CD.

Konkrétní změna portu se provádí následujícím způsobem. Otevřete si skript uživatelského rozhraní. Pokud se vám po dvojkliku levým tlačítkem myši otevře základní okno pro konfiguraci a ovládání (viz Příloha VI-1), je nutné ho zavřít a stisknout pravé tlačítko myši na tento skript a vybrat Edit Script.



Příloha VI-1: Základní okno

Po otevření zdrojového kódu skriptu se upraví část příkazu z řádky osm, která je označena červeně.



```
$h = DllCall("Kernel32.dll", "hwnd", "CreateFile", "str", "\\.\COM1", "int",  
BitOR($GENERIC_READ,$GENERIC_WRITE), "int", 0, "ptr", 0, "int",  
$OPEN_EXISTING, "int", $FILE_ATTRIBUTE_NORMAL, "int", 0)
```

Standardní komunikace probíhá na portu COM1, proto je v příkazu **\\.\COM1**. Pokud chcete používat např.: COM5 bude v příkazu na označeném místě **\\.\COM5**.

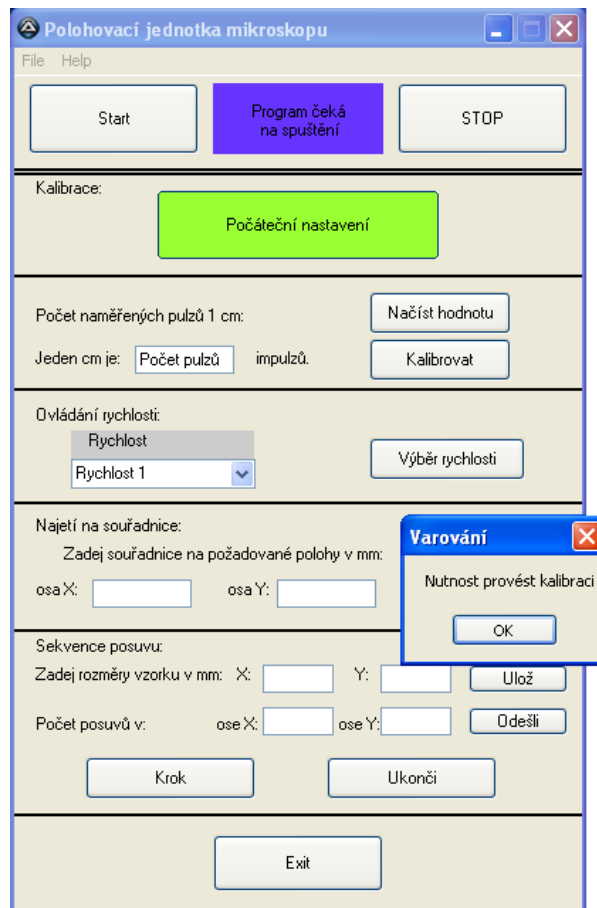
Po provedení změny skriptu je nutné ho uložit. V dalším kroku se provede kompilace souboru a vytvoření spouštěcího souboru se stejným názvem jako skript. Je nutné stisknout pravé tlačítko na skript a vybrat možnost Compile Script. Tím je uživatelské rozhraní připravené k užívání.

**Poznámka:** Pokud nelze ovládat jednotku pomocí uživatelského rozhraní, nepovedlo se vytvořit komunikační kanál mezi zařízeními. Spusťte tedy program putty.exe, zaškrtněte komunikační typ Serial a do políčka s textem COM1 запиšte port, na kterém chcete komunikovat. Pak stiskněte Open, spojení se vytvoří a otevře se okno, to následně zavřete a znovu spusťte aplikaci polohovací jednotky pro mikroskop.

## 2. Kalibrace

Kalibraci je nutno provést po každém odpojení mikrokontroléru od napětí, protože ztratí hodnoty o pozici zkoumaného vzorku a ostatních hodnot nutných k správné funkci. Při kalibraci je nutné, abyste přesně dodrželi následující kroky.

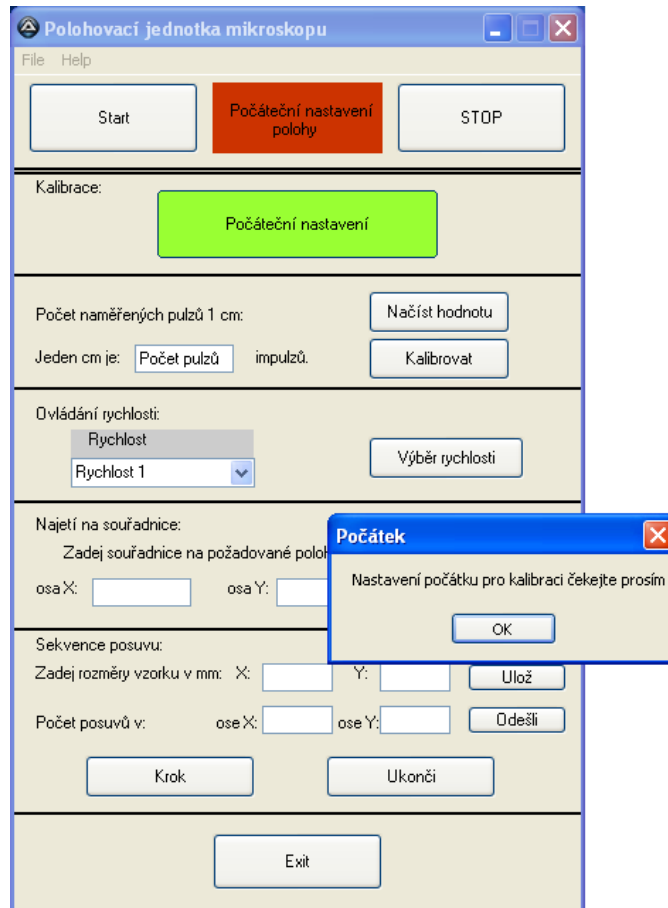
a) Stiskněte tlačítko Start. Zobrazí se okno, které vám oznámí, že je nutné provést kalibraci, viz Příloha VI-2.



Příloha VI-2: Po stisku Start

Stisknutím tlačítka OK se změní informační pole na Počáteční nastavení polohy

b) Stiskem tlačítka Počáteční nastavení se nejdříve pracovní vzorek vrátí do počáteční pozice v ose X a následně v ose Y. Po dojetí na souřadnici  $[0, x]$  se zastaví motor v ose X a na souřadnicích  $[0, 0]$  i motor v ose Y. Zobrazí se pomocné okno, které popisuje tuto činnost, viz Příloha VI-3.

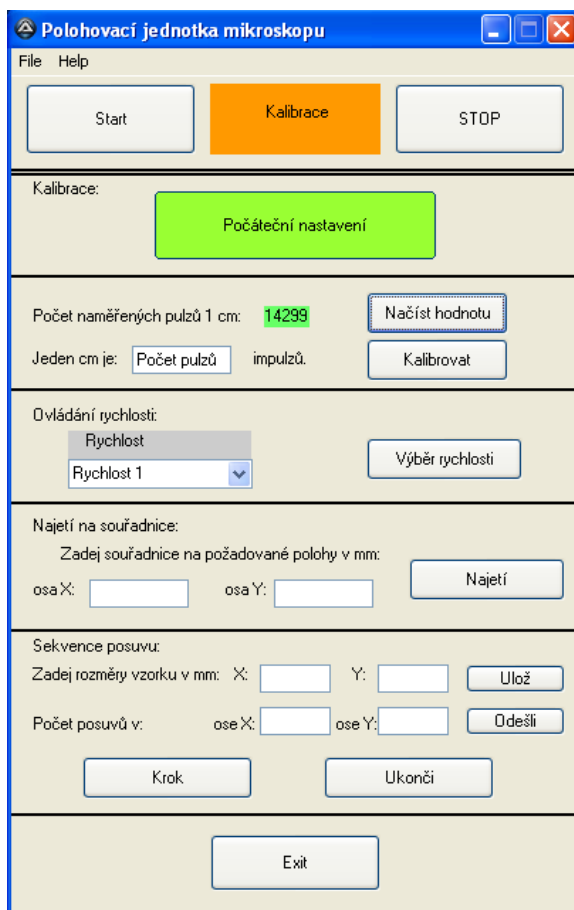


Příloha VI-3: Počáteční nastavení pozice

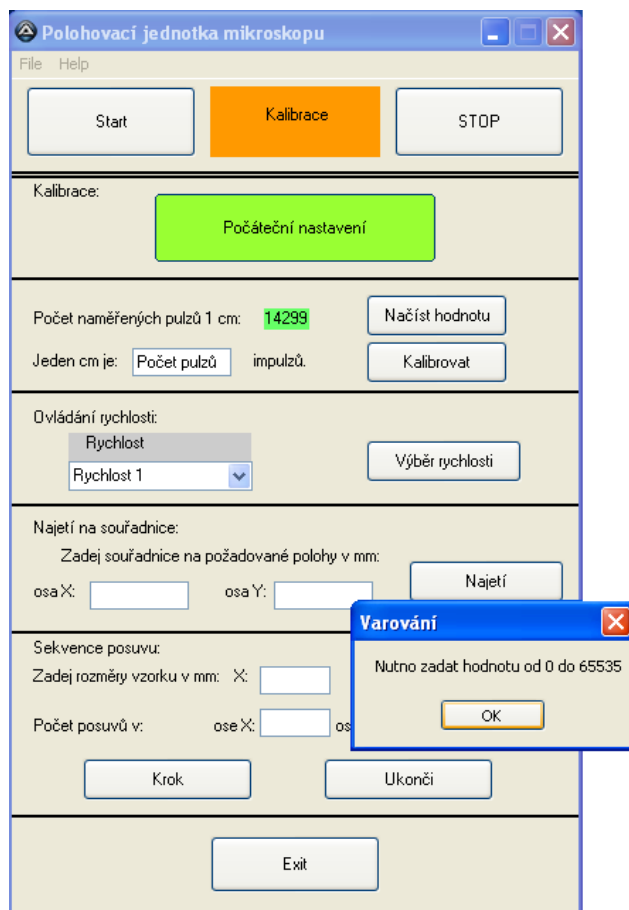
c) Dalším krokem je najetí pracovním stolkem na hodnotu jednoho centimetru v ose X. To se provede stiskem kláves se šipkami. Když je hodnota správně nastavena, stiskněte tlačítko Načíst hodnotu a zobrazí se hodnota s počtem pulzů vygenerovaných po najetí na hodnotu 10 milimetrů, viz Příloha VI-4.

Zde je konkrétní hodnota 14299 pulzů, tu zadejte do políčka označeného Počet pulzů. Zadaná hodnota musí být v rozsahu 0 – 65 535. Při špatně zadané hodnotě, nebo ponechání textu Počet pulzů se zobrazí varování, viz Příloha VI-5.

Když hodnotu zadáte správně, vrátí se stolec do počáteční pozice a kalibrace je úspěšně dokončena.



Příloha VI-4 : Načtení počtu pulzů 10 milimetrů



Příloha VI-5: Špatně zadaná hodnota 10 milimetrů

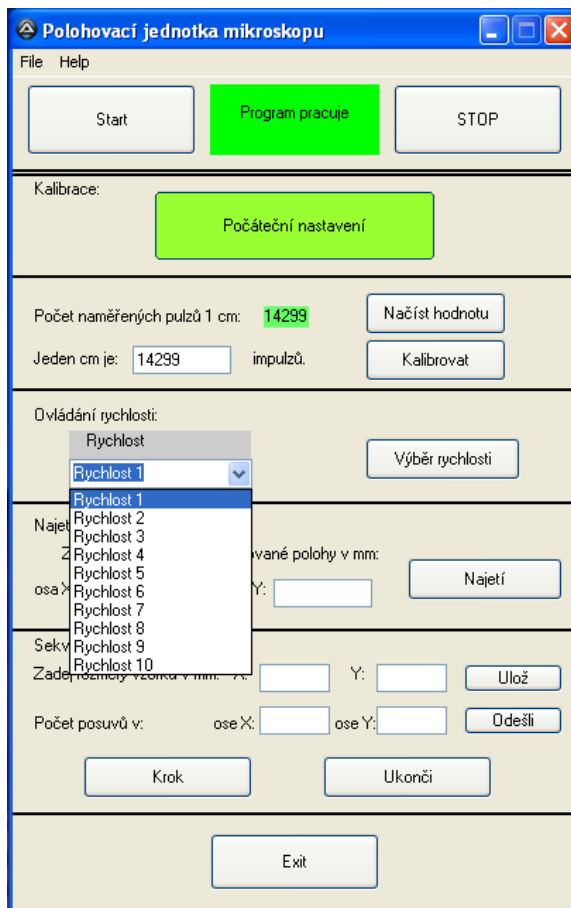
**Poznámka:** Pokud v průběhu kalibrace nastane jakýkoliv problém, nebo provedete kroky ve špatném pořadí, je nutné ji začít znovu od začátku.

Po kalibraci je možné celý systém ovládat šipkami na klávesnici a tím libovolně pohybovat vzorkem.

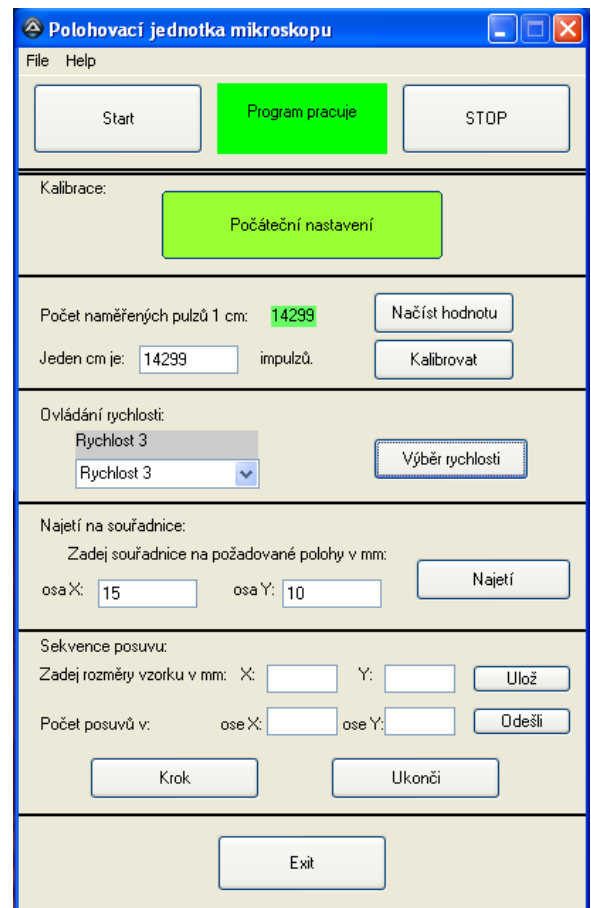
### 3. Najetí na pozici

Po úspěšné kalibraci vyberete rychlost vhodnou pro najetí na zvolenou pozici. Máte možnost výběru z deseti rychlostí, viz Příloha VI-6.

Když vyberete požadovanou rychlost, je nutné ji potvrdit tlačítkem Výběr rychlosti. Vybraná rychlost se zobrazí v poli nad výběrem zadané hodnoty pro najetí na pozici (viz Příloha VI-7), kde je vybrána Rychlost 3 a souřadnice najetí.



Příloha VI-6: Výběr rychlosti



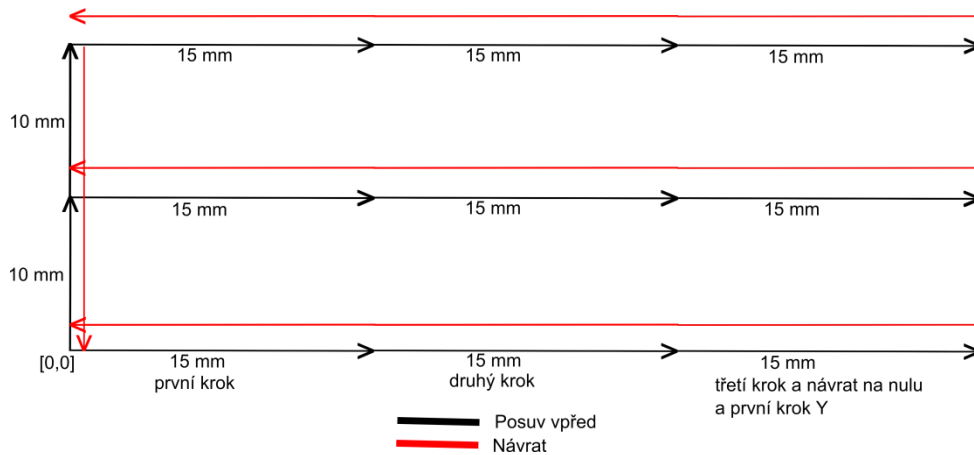
Příloha VI-7: Výběr rychlosti tři a souřadnice pro najetí

Při vhodně zvolené rychlosti zadáte souřadnice požadované polohy. Hodnoty jsou zadávané v milimetrech a maximální rozsah je omezen velikostí pracovního stolku, který je 100 x 50 milimetrů. Po zadání hodnot v požadovaném rozsahu a stisku klávesy Najetí se stolek posune na požadované souřadnice. Posuv se provádí nejdříve v ose X a poté v ose Y. Potvrzení o dojetí se neodesílá zpět k uživatelskému rozhraní prostřednictvím

zprávy, ale je pouze vizuálním pohledem na mikroskop. Hodnoty najetí lze libovolně kombinovat. Můžete nejdříve najet na souřadnice 60 x 30 mm a potom na 30 x 15 mm.

#### 4. Sekvence snímaných vzorků

Sekvence snímání vzorků, pohybuje se zkoumaným vzorkem po pracovním stolku, viz Příloha VI-8.

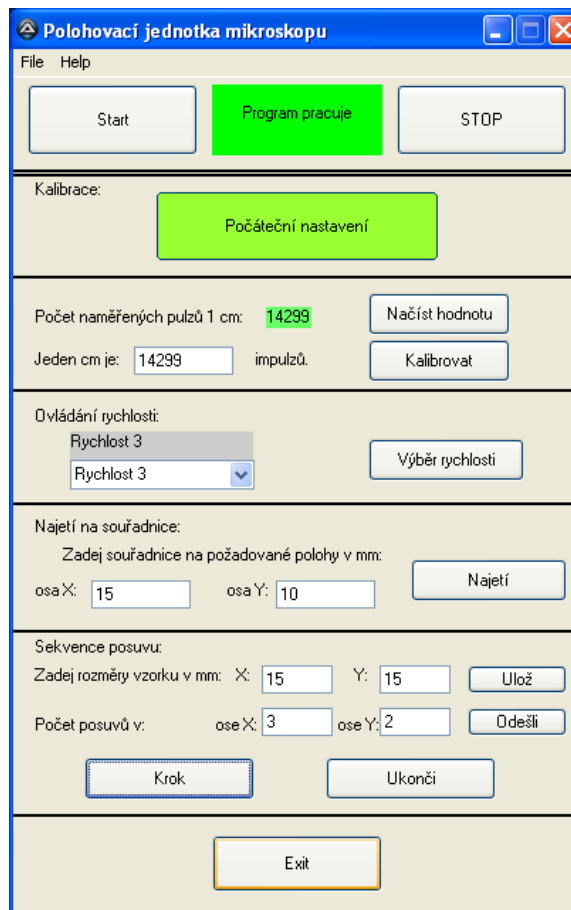


Příloha VI-8: Posuv vzorku při sekvenčním snímání

Hodnoty potřebné k tomuto snímání jsou rozměry zkoumaného vzorku, což je vlastně velikost posuvu v ose X a ose Y a počet posuvů v obou osách.

Minimální rozměr zkoumaného vzorku smí být 10x10 mm a maximální 99x99 mm, maximální počet posuvů v obou osách je 9.

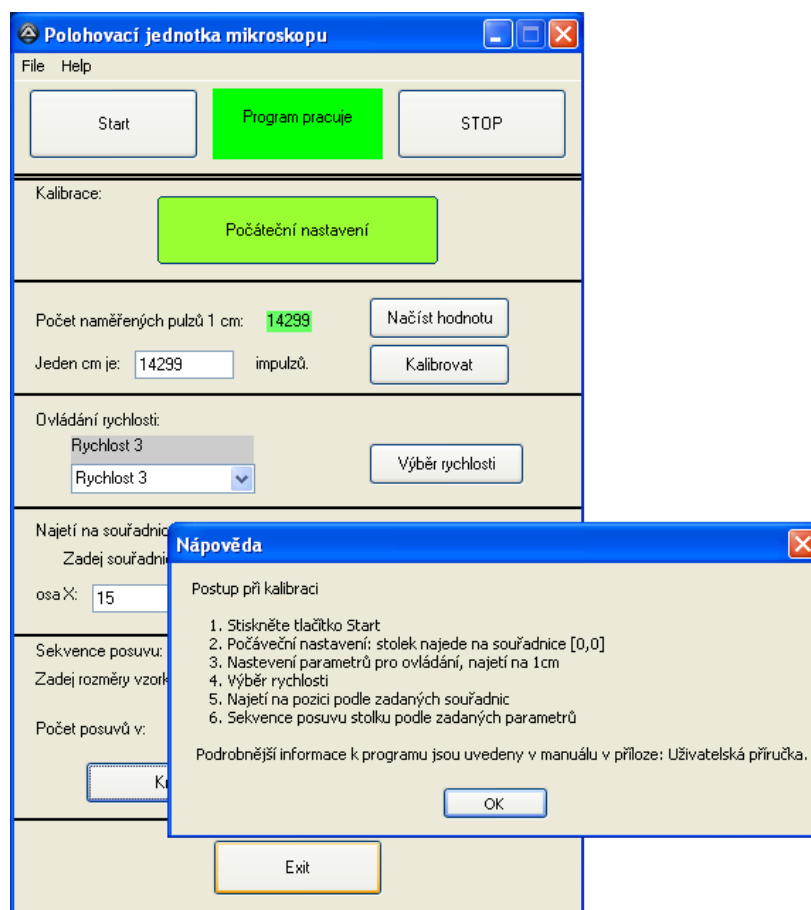
Tyto hodnoty postupně zadávejte do příslušných políček uživatelského rozhraní a vždy dvojici potvrďte, viz obrázek VI-9. Po zadání a potvrzení obou dvojic parametrů, můžete začít se sekvenčním posuvem vzorku. Jednotlivé kroky se vykonají po stisku klávesy „Krok“, po dokončení posledního kroku se pracovní stůl vrátí do počáteční souřadnice  $[0,0]$  a je možno provádět najetí na pozici nebo opět sekvenční snímání vzorku.



Příloha VI-9: Nastavené a potvrzené hodnoty sekvence snímání

## 5. Menu programu

Součástí uživatelského rozhraní je i jednoduché menu, které obsahuje položky File a Help. V položce Help je stručný postup práce s programem (viz Příloha VI-10) a položku About ve které je název a rok vzniku programu.



Příloha V-10: Menu nápověda

V položce File je exit, pomocí kterého je možno program vypnout. Pro tento účel slouží i tlačítko Exit, po jeho stisku se program ujistí, zda ho opravdu chcete ukončit.

Tlačítko Stop zastaví běh programu.

Při tvorbě nového uživatelského rozhraní je nutné alespoň přibližně dodržet nastavenou přenosovou rychlost po sériové lince 9600 bps.

Přeji vám příjemnou a bezproblémovou práci s tímto programem.



## Příloha VII – CD

### Obsah CD:

- /firmware
  - o firmware\_polohovaci\_jednotka
- /uživatelké\_rozhraní
  - o Polohovaci\_jednotka\_pro\_mikroskop.au3
  - o Polohovaci\_jednotka\_pro\_mikroskop.exe
  - o Readme
  - o putty.exe
  - o autoit-v3-setup.exe
- /doc
  - o diplomova\_prace.pdf
- /schema
  - o Polohovaci\_jednotka.sch
  - o Polohovaci\_jednotka.tiff
- /navrh\_desky
  - o Polohova\_jednotka\_vrstvaA.pcb
  - o Polohova\_jednotka\_vrstvaB.pcb
  - o Polohova\_jednotka\_vrstva\_popis.pcb
  - o Polohova\_jednotka\_rozložení.pcb
  - o Polohova\_jednotka\_vysledná.pcb

