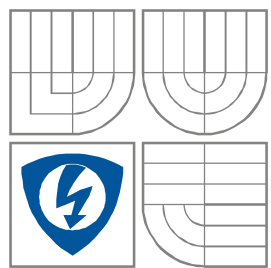


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

NÁVRH A ŘÍZENÍ CNC STROJE

CNC - DESIGN AND CONTROL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VOJTĚCH MATOUŠEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LUKÁŠ KOPEČNÝ, Ph.D.

BRNO 2011

Abstrakt

Mým úkolem bylo sestavit konstrukci menší tříosé CNC frézy. Navrhl jsem elektroniku pro ovládání pohonů. Hlavní částí je realizace řídicí jednotky, která bude zajišťovat řízení stroje. Práce obsahuje její kompletní návrh, DPS, konstrukci a popis programu pro uP ARM. Jednotka může pracovat jak samostatně, tak jako HW interpolátor připojený k PC.

Klíčová slova

CNC, frézka, krokový motor, TB6560, ARM, STM32, FT232, HW interpolátor

Abstract

My task was to build a smaller structure triaxial CNC milling. I designed electronics for motion control. The main part is the implementation of the control unit, which will provide control of the machine. The work includes the complete design, PCB design and program description for the uP ARM. The unit can work independently as well as hardware interpolator connected to the PC.

Keywords

CNC, milling machine, stepper motor, TB6560, ARM, STM32, FT232, hardware interpolator

Bibliografická citace:

MATOUŠEK, V. *Návrh a řízení CNC stroje*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 65s. Vedoucí diplomové práce byl doc. Ing. Lukáš Kopečný, Ph.D.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Návrh a řízení CNC stroje jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **20. května 2011**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Lukáši Kopečnému, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **20. května 2011**

.....
podpis autora

Obsah

1	ÚVOD.....	13
2	KONSTRUKCE	14
2.1	Materiál	14
2.2	Koncepce.....	14
2.3	Vedení OS	16
2.4	Posuvy os	18
2.5	Protizávaží vertikální osy	21
2.6	Vřetenno.....	22
3	ELEKTRONIKA A KROKOVÉ MOTORY.....	23
3.1	Krokový motor.....	23
3.1.1	Popis funkce krokových motorů – obecně	23
3.1.2	Zvolený krokový motor.....	24
3.2	Driver krokového motoru.....	25
3.2.1	TB6560.....	26
3.2.2	Schéma	28
3.2.3	Plošný spoj	29
3.3	Zdroj.....	30
3.4	Koncové spínače a nouzový vypínač	32
4	ŘÍDICÍ JEDNOTKA.....	34
4.1	Koncepce jednotky.....	34
4.2	Vývojový kit STM32 Discovery Value Line	35
4.3	Komunikační modul.....	37
4.3.1	Schéma komunikačního modulu	38
4.3.2	DPS komunikačního modulu	38
4.4	Spodní panel.....	39
4.4.1	Schéma spodního panelu.....	40
4.4.2	DPS spodního modulu.....	41
4.5	Horní panel.....	42
4.5.1	Schéma horního panelu	43
4.5.2	DPS horního panelu	44
5	PROGRAMY PRO ŘÍDICÍ JEDNOTKU	45
5.1	Koncepce programu pro mikrokontroler.....	45
5.2	Program pro mikrokontroler	47

5.2.1	Knihovna inicializace.h.....	47
5.2.2	Knihovna USART_s_prer.h.....	47
5.2.3	Knihovna lcd.h.....	48
5.2.4	Hlavní funkce main.....	48
5.2.5	Lineární interpolace	50
5.2.6	Režim frézování	51
5.3	Program pro PC.....	51
6	TESTOVÁNÍ.....	53
6.1	Frézka.....	53
6.2	Elektronika	54
6.3	Řídicí jednotka	54
7	ZÁVĚR.....	55

Seznam obrázků

Obrázek 1: Frézka	15
Obrázek 2: Montážní otvory	16
Obrázek 3: Lineární vedení Hennlich [2]	16
Obrázek 4: Broušené tyče a linearset [2]	16
Obrázek 5: Vpravo je vedení osy Z, vlevo detail ložisek na ose Y	17
Obrázek 6: Rozložení a upevnění ložisek - schematicky	18
Obrázek 7: Kuličkový šroub s maticí	18
Obrázek 8: Matice a její duralové pouzdro	19
Obrázek 9: Uložení šroubu v ložiskovém pouzdře	19
Obrázek 10: Spojení korkového motoru s trapézovým šroubem a jeho uložení	20
Obrázek 11: Detail matice	20
Obrázek 12: Celkový pohled na posuv	20
Obrázek 13: Protizávaží vertikální osy	21
Obrázek 14: Vřeteno Kress FME 1050	22
Obrázek 15: Vřeteno DWT GS-600 VS upevněné na CNC	22
Obrázek 16: Příklad krokového motoru, stator, rotor [3]	23
Obrázek 17: Vlevo je příklad cívek v unipolárním zapojení, vpravo	24
Obrázek 18: Olivetti 57BYG 060 a Nema23 18.9kg/cm	25
Obrázek 19: Krokové motory připevněné na ose X a Z	25
Obrázek 20: Pouzdro HZIP25 a popis jeho vývodů [4]	26
Obrázek 21: Blokové schéma vnitřního zapojení obvodu TB6560AHQ[4]	28
Obrázek 22: Schéma driveru s TB6560	29
Obrázek 23: DPS – plán osazení a DPS předloha strany spojů	30
Obrázek 24: Hotový driver při testech	30
Obrázek 25: Schéma zdroje	31
Obrázek 26: DPS zdroje a předloha strany spojů	31
Obrázek 27: Hotový zdroj	32
Obrázek 28: Použitý koncový spínač [5]	32
Obrázek 29: Možné zapojení HW koncových spínačů	33
Obrázek 30: Nouzový vypínač	33

Obrázek 31: Plošné spoje jsou frézované	34
Obrázek 32: Řídicí jednotka	35
Obrázek 33: Vývojový kit STM32 Value Line Discovery [6]	36
Obrázek 34: Horní pohled na vývojový kit a jeho popis [6]	37
Obrázek 35: Schéma Komunikačního modulu	38
Obrázek 36: Hotový komunikační modul.....	39
Obrázek 37: Osazovací plán komunikačního modulu	39
Obrázek 38: Spodní panel již s umístěným modulem mikrokotroleru	40
Obrázek 39: Schéma spodního panelu (podrobněji v příloze).....	41
Obrázek 40: Osazovací plán DPS spodního panelu.....	42
Obrázek 41: Horní panel s připevněným displejem.....	43
Obrázek 42: Část schématu horního panelu	43
Obrázek 43: Osazovací plán DPS horního panelu.....	44
Obrázek 44: Okno programu Keil uVision 4.....	45
Obrázek 45: grafické naznačení funkce Bresenhamova algoritmu [11].....	50
Obrázek 46: Okno obslužného programu v PC	52
Obrázek 47: Zkušební nápis „VUT Fekt“.....	53
Obrázek 48: Schéma driveru.....	58
Obrázek 49: Předloha DPS driveru.....	59
Obrázek 50: Osazovací plán driveru.....	59
Obrázek 51: Schéma komunikačního modulu	60
Obrázek 52: DPS komunikačního modulu	60
Obrázek 53: Osazovací plán komunikačního modulu	61
Obrázek 54: Schéma horního panelu	62
Obrázek 55: DPS horního panelu	63
Obrázek 56: Osazovací plán horního panelu	63
Obrázek 57: Schéma spodního panelu.....	64
Obrázek 58: DPS Spodního panelu	65
Obrázek 59: Osazovací plán spodního panelu.....	65

Seznam tabulek

Tabulka 1: Nastavení pinů M1 a M2 [4].....	26
Tabulka 2: Nastavení pinů Dcy1 a Dcy2 – režim nabuzení [4].....	27
Tabulka 3: Tabulka nastavení proudu do cívek motoru. [4].....	27

Obsah CD

Desky plošných spojů
Osazovací plány
Soubory s návrhy pro v programu EAGLE
Program pro mikrokontroler v programu Keil uVision
Program pro PC v programu Visual Studio 2010
Fotografie CNC frézky
Fotografie elektroniky
Diplomová práce v PDF

1 ÚVOD

V této diplomové práci popíši konstrukci tříosé CNC frézky, včetně elektroinstalace a motorů. Dále se budu zabývat elektronikou pro krokové motory (driver) a řídicí jednotkou. Práce bude tedy rozdělena do tří dílčích kapitol. V první ukážu návrh a realizaci konstrukce. Volbu vhodné koncepce a cenově dostupných komponent a materiálů. V druhé části způsob řízení krokových motorů, které zajišťují posuv jednotlivých os. Výběr optimálního driveru, jeho návrh a realizaci. Krátce se zmíním i o napájecím zdroji. Ve třetí části se budu zabývat řídicí jednotkou, která je koncipována jako ovládací panel s displejem. Může pracovat buď ve spojení s PC, nebo samostatně. Důležitou částí bylo nalezení dostatečně výkonného mikrokontroléru, který by zvládl poměrně složité výpočty nutné k řízení CNC stroje pomocí G-kódu. Popíši zvolený procesor, respektive vývojový kit od firmy ST.

V závěru práce se budu zabývat ověřením funkčnosti – testy dílčích zařízení a kompletního stroje.

Frézka je stavěna pro práci se dřevem a hliníkem. Materiál je lepené lamino, což ve spojení s vhodnou konstrukcí, dává frézce zajímavé vlastnosti. Oproti kovové konstrukci nehrozí žádné vibrace a pevnost je srovnatelná.

2 KONSTRUKCE

2.1 Materiál

Ke stavbě hlavních částí jsem vybral jako konstrukční materiál lepenou laminovanou dřevotřískovou desku tloušťky 36mm. A to z důvodu dostupnosti, nízké ceny a velmi snadné opracovatelnosti. Díky lepeným vrstvám se vyznačuje velkou tuhostí a pevností. Díky použití laminovaných desek (včetně řezaných hran) je materiál odolný i proti vzdušné vlhkosti a případnému vsakování chladících emulzí. Nevýhodou je ovšem vyšší hmotnost.

Pro menší a více namáhané plochy (domečky ložisek, ukotvení motorů) jsem volil lepenou překližku 10mm (případně násobky tloušťky – lepená sendvičová konstrukce). Ke spojování byly používány vruty 5x70mm, na rozebíratelné spoje metrické šrouby M8 s pojistnou maticí.

2.2 Koncepce

K volbě vhodné koncepce je důležité vědět, k čemu bude stroj sloužit, z jakého materiálu bude vyroben a v neposlední řadě rozpočet ke stavbě.

Vzhledem k tomu, že jsem chtěl aktivní frézovací plochu alespoň 700x700mm a konstrukční materiál lepenou laminovanou dřevotřísku, rozhodl jsem se pro pohyblivý stůl - v jedné ose (X). Druhá osa (Y) je na dvou nosnících nad stolem a osa Z je tak jediná, která se pohybuje i se svou základnou. Boční pohled na frézku je na Obrázku 1. Tuto koncepci jsem vybral z důvodu poměrně velké váhy stroje. Při zvažování možnosti klasické koncepce s pevným stolem, by byly nároky na tuhost osy X velmi vysoké. Pojezd by nesl váhu všech os a musel by vydržet i veškeré síly působící na všechny osy. To by konstrukci velmi zhmotnilo a také prodražilo.

Zvolená koncepce je vhodná i s důvodu jednoduššího vedení kabeláže, jediné místo kde je třeba řešit vedení pohybu kabelů je k ose Z.

Konstrukce je řešena modulárně, z důvodu případných servisů a transportů stroje. Samotné osy jsou složeny z lepených desek a pevně sešroubovány vruty. Ale veškeré části, které vyžadují jisté nastavování (např. sousost vedení) jsou spojovány metrickými šrouby M8. Jednotlivé osy a části rovněž. Frézku lze rozložit na šest částí: stůl, osa X, osa Y, osa Z a dva boční nosníky.

Jak jsem již zmínil, stroj je navržen tak, aby některé vlastnosti, konkrétně sousost vedení, předpnutí matic a kuličkových ložisek na vedení, bylo možné měnit nastavením utažení šroubů. Ty musím být lehce dostupné. Proto jsou v osách vyvrtány montážní otvory (viz Obrázek 2.) a šrouby k nastavení matic a ložisek jsou volně dostupné, tedy bez odmontovávání jiných částí frézky.



Obrázek 1: Frézka



Obrázek 2: Montážní otvory

2.3 Vedení OS

Vedení zajišťuje každé ose možnost pohybu pouze v daném směru. A v nežádoucích směrech utlumí všechny síly při zachování snadného pohybu. Musí být samozřejmě dimenzováno na práci, kterou má stroj vykonávat. V případě mé frézky je netvrďším materiálem dubové dřevo nebo hliník.

Zvažoval jsem samozřejmě nejpoužívanější lineární vedení. Viz Obrázek 3. Ale cena nových tyčí a hlavně vodících domečků byla mimo mé možnosti.



Obrázek 3: Lineární vedení Hennlich [2]

Další možností jsou hlazené či broušené ocelové tyče s lineárními ložisky. Tyto tyče jsou, ale určeny pro malé síly a hmotnosti samotných os. Upevnění mají pouze na svých koncích. A při rázech, ke kterým při frézování může dojít, mohou zapružit, což vede ke znehodnocení práce.



Obrázek 4: Broušené tyče a linearset [2]

Třetí možností je použití vodících tyčí, ale tzv. podepřených. Tyče jsou přišroubovány k rovné podložce, která slouží jako výztuha, tou je mnohdy samotná konstrukce stroje. Jako posuvy se zde využívají otevřená lineární ložiska.

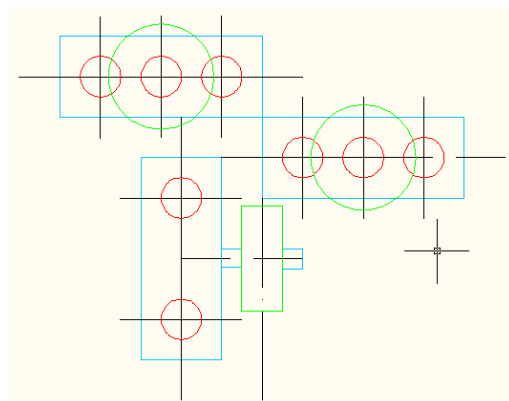
Tato možnost vyhovovala mým požadavkům na tuhost a rozumnou cenu. Místo lineárních ložisek jsem ale navrhl soustavu tří klasických jednořadých ložisek místo každého linearsetu. Viz Obrázek 6. Broušené tyče jsou průměru 26mm. Upevněny jsou přímo k tělu osy, která je poskládána jako kvádr s vnitřními vzpěrami. Připevněny jsou metrickými šrouby M8 skrz laminodesku. V tyčích jsou vytočeny přímo závitové otvory každých 100mm. Vzhledem k tomu že vodící tyče jsou v nutně v páru, tak je nutné nastavit vzájemnou souosost. Jedna osa je upevněna fixně, a druhá mám možnost nastavení v rozmezí až 3mm. Toto nastavení jsem provedl při sestavování. Pokud nedojde k nějaké nehodě s velkým rázem na osu, nebude toto nastavování již nutné.

Pro pojezdy os, jsem zvolil ložiska 6000 2Z od firmy DPI. Jde o ložiska průměru 26mm s otvorem 10mm a šířkou 6mm, krytá. Každé ložisko je nezávisle upevněno přes ocelový kvádr se třemi otvory, na jednom je přišroubováno ložisko a dalšími dvěma je připevněno k laminové konstrukci pojezdu. To umožňuje podle utáhnutí (přes pružnou podložku) nastavení předepnutí vůči vodící tyči.

Kvůli prachu a nečistotám bude nutné posuvy dovybavit prachovou stěrkou. Pro fázi ladění a testování zatím nebyla třeba.



Obrázek 5: Vpravo je vedení osy Z, vlevo detail ložisek na ose Y

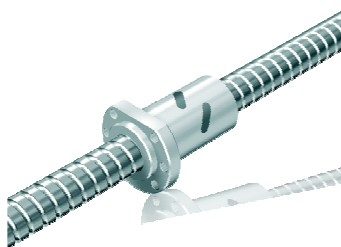


Obrázek 6: Rozložení a upevnění ložisek - schematicky

2.4 Posuvy os

Samotné vedení nám zajišťuje jen v přirovnání koleje pro pohyb, ale samotný pohyb – posuv musíme zajistit jinak. Samozřejmě je důležitá jeho přesnost, jemné rozlišení a jeho řízení.

V praxi se nejčastěji používají velmi přesné kuličkové šrouby, viz Obrázek 7. Ale jejich cena je velmi vysoká a pro dané frézování dřeva či maximálně hliníku je toto řešení zbytečné.



Obrázek 7: Kuličkový šroub s maticí

Proto jsem pro mou konstrukci zvolil jinou možnost - trapézový šroub. Ve spojení s mosaznou maticí.

Šroub jsem vybral průměru 16mm se stoupáním 4mm. U šroubu je důležitá jeho přímost, jen malé prohnutí znemožní funkčnost. Protože způsobuje páčení na matici a tím její opotřebení a taky daleko vyšší potřebu síly k otočení. Z tohoto důvodu jsem musel první šrouby u dvou os demontovat a budou nahrazeny novými – kvalitnějšími.

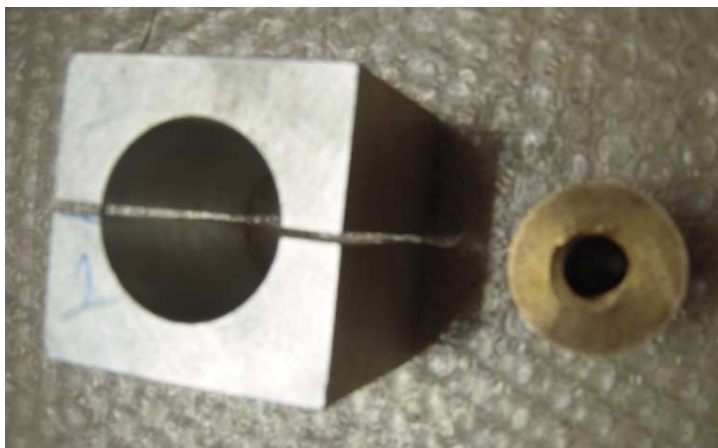
Mosaz byla jako materiál k výrobě matice zvolena díky své samomaznosti. A z důvodu, že je měkčí než ocel. Při neustálém přejíždění, otáčení matice po šroubu by v případě ocel – ocel docházelo i při dostatečném mazání k opotřebení, které by časem způsobilo vůle. Matice má zavít řezaný tak, aby se šroub mohl protáčet naprosto volně, tedy s velkou radiální vůlí. Ale axiální vůle musím být naprosto vymezena. Matice je

uložena v duralovém pouzdře s prizmatickou vložkou. Viz Obrázek 8. Pouzdro s maticí je pevně přišroubováno k pojezdu osy.

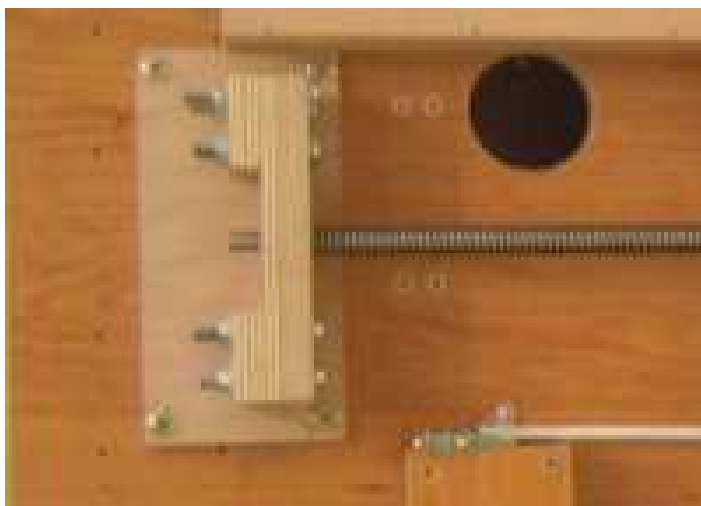
Trapézový šroub je na obou stranách uložen v ložisku, které vymezuje jeho pozici a také axiálně působící síly při posuvu. Konce šroubů jsou k tomuto účelu osoustruženy na 9,99mm a jeden konec šroubu pak následně ještě na 7,99mm – pro připojení pružné spojky Viz Obrázek 10.

Z druhé strany šroubu je přes pružnou spojku připojena hřídel krokového motoru. Díky uložení v ložiskách a pružné spojce na motor nepůsobí kromě žádaného otáčení žádné síly. Jako pružná spojka je použit hliníkový váleček, který je střídavě prořezávaný. Obrázek 10.

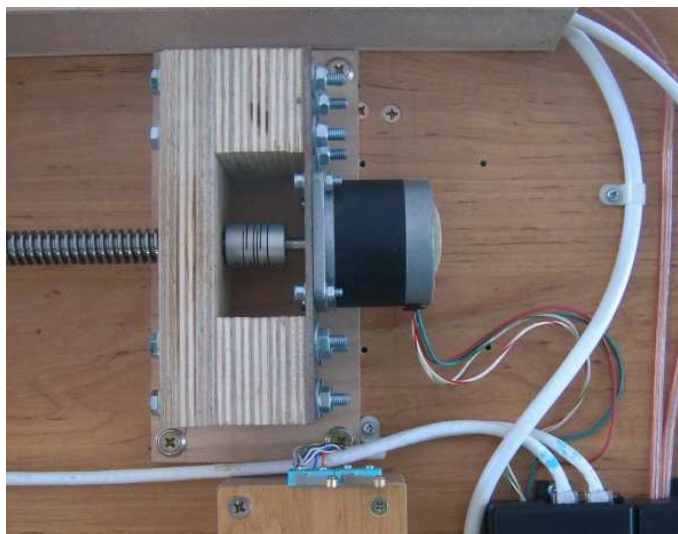
Celkově je posuv proveden tak, že šroubem lze snadno otáčet je rukou a pojezd osy je pevný.



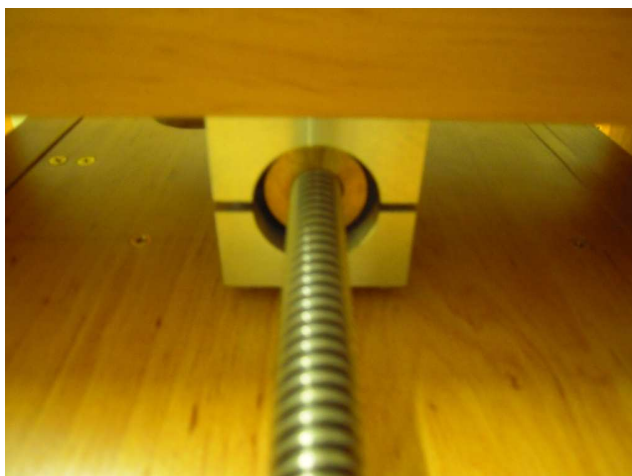
Obrázek 8: Matice a její duralové pouzdro



Obrázek 9: Uložení šroubu v ložiskovém pouzdře



Obrázek 10: Spojení korkového motoru s trapézovým šroubem a jeho uložení



Obrázek 11: Detail matice



Obrázek 12: Celkový pohled na posuv

2.5 Protizávaží vertikální osy

Po prvních testech vertikální osy se potvrdilo, že vozík osy je příliš těžký. Na vodící matici byl takový axiální tlak, že původní motor Olivetti 57BYG 060 (o motorech více v kapitole 3.1.2) se svým krouticím momentem 0,84Nm nedosahoval potřebných otáček bez ztráty kroku. Rozhodl jsem se jej nahradit za silnější motor s krouticím momentem 1,9Nm (18.9kg/cm). Tento motor již s váhou vozíku poradil.

Obával jsem se ale velkého namáhání matice a jejího rychlého opotřebení a z něho vyplývající ztrátě přesnosti. Proto jsem se rozhodl dovybavit osu Z protizávažím. Bylo nutné navrhnout a vyrobit další vedení pro samotné závaží, které se bude pohybovat přesně opačným směrem než vozík osy Z. Na vedení protizávaží už nejsou žádné nároky na přesnost či tuhost. Proto jsem zvolil možnost vedené po dvou hliníkových tyčích tloušťky 10mm. A jako kluzné vedení postačili jen otvory v laminu vyložené filcovými kroužky napuštěnými olejem o vysoké viskozitě. Samotný domek se závažím je celý z lamina. Viz obrázek 13. Domek se závažím je k vozíku osy Z připojen ocelovým lankem 3mm přes tři kladky.

Jako závaží jsem použil olověné válečky o celkové hmotnosti 6,3Kg. Váhu jsem určil následujícím postupem. Na ose Z jsem uvolnil matici posuvu, tím byl vozík osy Z volný. Spojil jsem ho s protizávažím a dodával olověné válečky do domku, dokud nebyly vozík s domkem v rovnováze. Poté jsem ubral asi 400g, aby byl vozík osy tlačén přirozeně dolů.

Sekundárním efektem protizávaží je díky umístění na opačné stran osy Y, než je umístěna osa Z i ulehčení páčivého tlaku na vedení osy Y.



Obrázek 13: Protizávaží vertikální osy

2.6 Vřeteno

Pro testování jsem nepoužíval žádné vřeteno, postačila obyčejná tužka nebo fix. Při testování mohlo dojít k nepředvídané kolizi. Z důvodu bezpečnosti jsem vřeteno namontoval až po několika testech

Rozhodoval jsem se mezi vřetenem KRESS FME 1050-1 (Obrázek 14). Jedná se o vysokootáčkovou frézu s regulací otáček v rozsahu 5000 – 25000 ot/min a s výkonem 1050W. Je dobře chráněno proti prachu a má ochranu proti přetížení. Disponuje standardními kleštinami 6 nebo 8mm.

Druhou možností, kterou jsem i nakonec zvolil je přímá bruska DWT GS-600 VS. Tato frézka má regulaci otáček v rozsahu 12000 – 27000 ot/min a výkon 600W. Oproti vřetenu KRESS není tak dobře krytováno, je nutné častější čištění. V základu je vybaveno pouze kleštinou 6mm, ale ostatní lze dokoupit. N obrázku 15 je vidět po připevnění k CNC. Upevnění vřetena k vozíku osy Z je řešeno opět z překližky.

Vřeteno je spínáno přes relé, není tedy možné elektronicky – programově měnit otáčky. Je nutné je nastavit manuálně.



Obrázek 14: Vřeteno Kress FME 1050



Obrázek 15: Vřeteno DWT GS-600 VS upevněné na CNC

3 ELEKTRONIKA A KROKOVÉ MOTORY

3.1 Krokový motor

3.1.1 Popis funkce krokových motorů – obecně

Občas se v praxi vyskytne potřeba pohonu, který umí přesně nastavit svoji polohu a tuto polohu i přes působící síly udržet (např. souřadnicové zapisovače nebo počítačem řízené obráběcí stroje). Právě to jsou aplikace jako šité na míru pro krokové motory. V robotice používáme krokové motory z důvodu jejich snadné obsluhy. Pro precizní řízení rychlosti nepotřebujeme naprogramovat komplexní PID kontrolér, pokud motory nepřetěžujeme, lze se obejít bez zpětné vazby o změně natočení – stačí počítat kroky. [3]

Nutno zmínit i nevýhodu a to trvalý odběr proudu i v klidové poloze. To způsobuje zbytečné zahřívání motoru. Je možno klidový stav vyhodnocovat a snižovat proud do cívek v době stání.

Na obrázcích je krokový motor s 200 kroky na otáčku (1.8 stupně na krok). Stator krokového motoru je tvořen sadou cívek. Pólové nástavce statoru jsou vroubkovány se stejnou roztečí jako rozteč magnetů na rotoru. Toto je jedna z částí zvyšující přesnost motoru při stejném počtu cívek. Rotor je tvořen hřídelí usazenou na kuličkových ložiskách a prstencem permanentních magnetů. [3]



Obrázek 16: Příklad krokového motoru, stator, rotor [3]

Základní princip krokového motoru je úplně jednoduchý. Proud procházející cívkou statoru vytvoří magnetické pole, které přitáhne opačný pól magnetu rotoru. Vhodným zapojováním cívek dosáhneme vytvoření rotujícího magnetického pole, které otáčí rotorem.

Podle požadovaného krouticího momentu, přesnosti nastavení polohy a přípustného odběru volíme některou z variant řízení. Kvůli přechodovým magnetickým jevům je omezena rychlost otáčení motoru a to na několik stovek kroků za sekundu (závisí na typu motoru a zatížení). Při překročení této maximální rychlosti (nebo při příliš velké zátěži) motory začínají ztrácet kroky. [3]

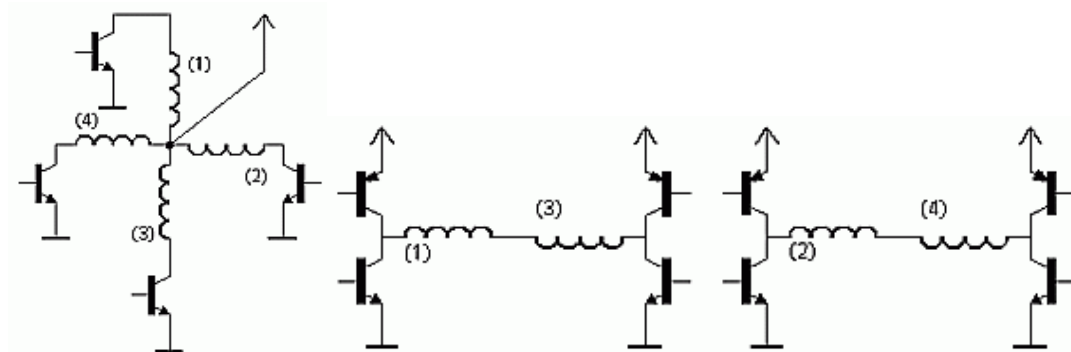
Metody řízení závisí na zapojení cívek statoru. Možnosti jsou dvě. Při unipolárním řízení jsou cívky zapojeny tak aby proud procházel vždy jen jednou cívkou. Viz obrázek 17. Unipolární řízení se vyznačuje polovičním proudem oproti následujícímu typu řízení ale bohužel také polovičním krouticím momentem. Výhodou je jednoduchost řídicí elektroniky.

Při bipolárním řízení prochází proud vždy dvěma protilehlými cívkami. Ty jsou zapojené tak, že mají navzájem opačně orientované magnetické pole. Motor v tomto režimu poskytuje větší krouticí moment, ovšem za cenu vyšší spotřeby. Pro řízení jsou zapotřebí 2 H-můstky, pro každou větev jeden. [3] Viz obrázek 17.

Další možností jak měnit vlastnosti řízení je volba mezi jednofázovým a dvoufázovým řízením.

Jednofázové řízení znamená, že magnetické pole generuje pouze jedna cívka (případně dvojice cívek při bipolárním buzení).[3]

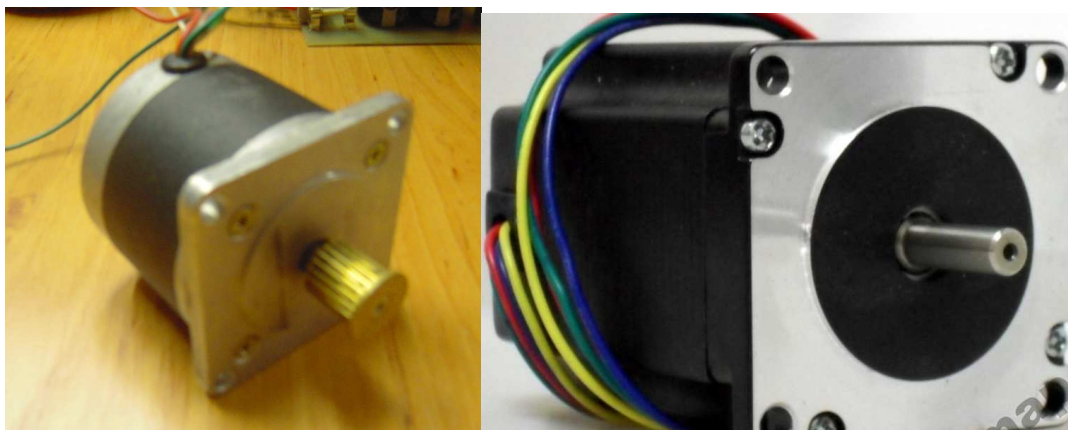
Při dvoufázovém řízení generují shodně orientované magnetické pole vždy dvě sousední cívky. Daní za vyšší krouticí moment je dvojnásobná spotřeba oproti řízení jednofázovému. [3]



Obrázek 17: Vlevo je příklad cívek v unipolárním zapojení, vpravo v bipolárním [3]

3.1.2 Zvolený krokový motor

Obecně platí, že pokud je frézka postavena správně, není i při velikosti a hmotnosti jako je u této prezentované, potřeba zvlášť silný krokový motor. Pro osu X a Y jsem zvolil motor Olivetti 57BYG 060. Mají standardních 200kroků/ot, a krouticí moment 0,84Nm při proudu 2A na fázi. Osazení je Nema23. Jedinou možností řízení je v bipolárním zapojení. Na ose Z jsem použil silnější motor opět Nema23, ale s krouticím momentem 1,9Nm



Obrázek 18: Olivetti 57BYG 060 a Nema23 18.9kg/cm



Obrázek 19: Krokové motory připevněné na ose X a Z

3.2 Driver krokového motoru

Pro řízení krokového motoru je, jak bylo zmíněno v předchozí kapitole, nutné použít takovou elektroniku, která bude spínat jednotlivé cívky a tím motor roztočí. V případě bipolárního řízení je stavba takového driveru s diskrétních součástek zbytečně složitá a tím i náchylnější k poruchám. Proto jsem se rozhodl použít novou generaci integrovaných obvodů Toshiba TB6560AHQ. K tomuto driveru je nutné připojit jen několik externích součástek a umožňuje nastavování proudu do cívek motoru a hlavně mikrokrokování. To umožňuje ještě dále zvýšit rozlišení motoru a to až na 1600kroků/ot. To už není ani tak důležité pro rozlišení frézky, ale pro jemné přechody z jednoho kroku do druhého.

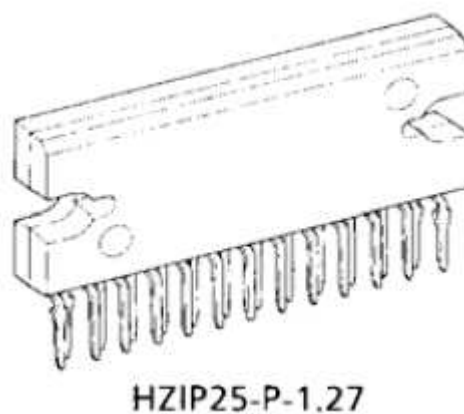
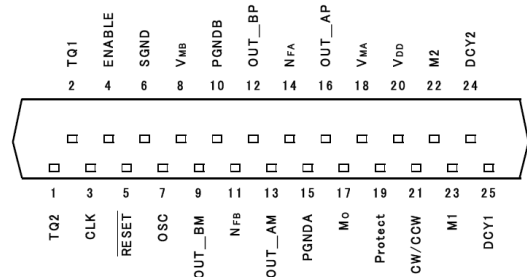
Řídící vstupy obvodu odpovídají standartu řízení os CNC strojů. Jedním vodičem se určuje krok (STEP), driver tedy reaguje na každou nástupnou hranu a překloupí do dalšího stavu – otočí o jeden krok motorem. A druhý vodič (DIR) určuje směr. Pro globální povolení chodu je zde vodič ENABLE. Podrobněji v samostatné kapitole.

K integrovanému obvodu je třeba připojit jen několik odporů a kondenzátorů. Důležité jsou mikropřepínače na nastavování funkcí obvodu. Signály STEP, DIR a ENABLE jsou přivedeny přes optočleny. Z důvodu vzájemné ochrany řídicí jednotky a driveru samotného. Driver je doplněn ještě o ochranu před zvýšením napětí.

3.2.1 TB6560

Integrovaný obvod TB6560AHQ je dvojitý H-můstek doplněný o řídicí logiku s množstvím nastavování. Obvod umožňuje řízení motoru opravdu pouze impulzy na každý krok a určením směru otáčení. Umožňuje mikrokrokování modifikovanou sinusovkou s nastavením počtů kroků na otáčku (200, 400, 800 a 1600). Maximální výstupní proud je 3,5A a napájecí napětí až 40V. To platí pro pouzdro HZIP25 na obrázku 17. Součástka je vybaveny i tepelnou ochranou, která při teplotě nad 85°C obvod odpojí. Součástku je nutné hodně chladit, jinak i při běžném provozu dochází k odpojování. Cena se v době psaní této práce pohyboval okolo 200Kč za kus.

TB6560AHQ (Top View)



Obrázek 20: Pouzdro HZIP25 a popis jeho vývodů [4]

Nastavování mikrokrokování probíhá pomocí pinů M1 a M2, kombinací jejich uzemňování je možno nastavit dělení kroku. Nastavení změny tvaru modifikované sinusoidy. Viz tabulka 1.

Input		Mode (Excitation)
M2	M1	
L	L	2-phase
L	H	1-2-phase
H	L	4W1-2-phase
H	H	2W1-2-phase

Tabulka 1: Nastavení pinů M1 a M2 [4]

Další možností nastavení je režim rychlosti nebuzení/odbuzení do nominálního proudu. Možnosti jsou čtyři. Bez regulace, s regulací pomocí vybíjení přes ochranné diody H-

můstku, regulace obrácením směru proudu. Poslední možnost je značně energeticky náročná a velmi obvod zahřívá. Proto je čtvrtá možnost kombinací dvou regulací, obvod zkouší přebytný proud vybit přes ochranné diody, a pokud se do určitého času nepovede, přepne na druhý režim.

Dcy2	Dcy1	Current Decay Setting
L	L	Normal 0%
L	H	25% Decay
H	L	50% Decay
H	H	100% Decay

Tabulka 2: Nastavení pinů Dcy1 a Dcy2 – režim nabuzení [4]

K nastavení proud cívkami je nutno dbát na maximální omezení 3,5A. Nastavení probíhá připojením adekvátních odporů na piny NfA a NfB. Jejich hodnota se určí výpočtem ze vzorce:

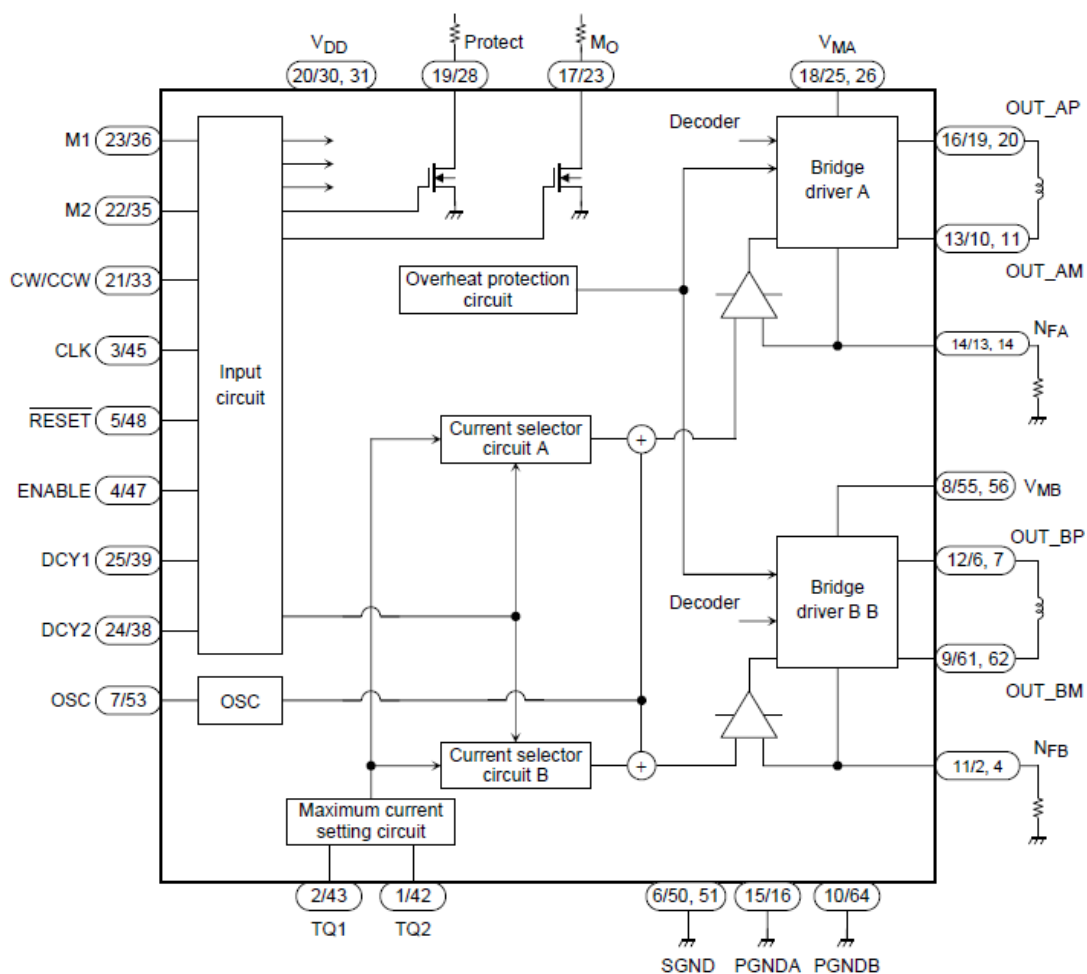
$$R[\Omega] = \frac{0,5}{I[A]}$$

Hodnoty odporu jsou tedy velmi malé. Odpor musí vydržet tepelné ztráty dané procházejícím proudem. Při nejvyšším možném proudu 3,5A je tedy nutné používat rezistory na výkon alespoň 2W.

Takto nastavený proud se dá snižovat pomocí pinů TQ1 a TQ2. Viz tabulka 3.

TQ2	TQ1	Current Ratio
L	L	100%
L	H	75%
H	L	50%
H	H	20% (weak excitation)

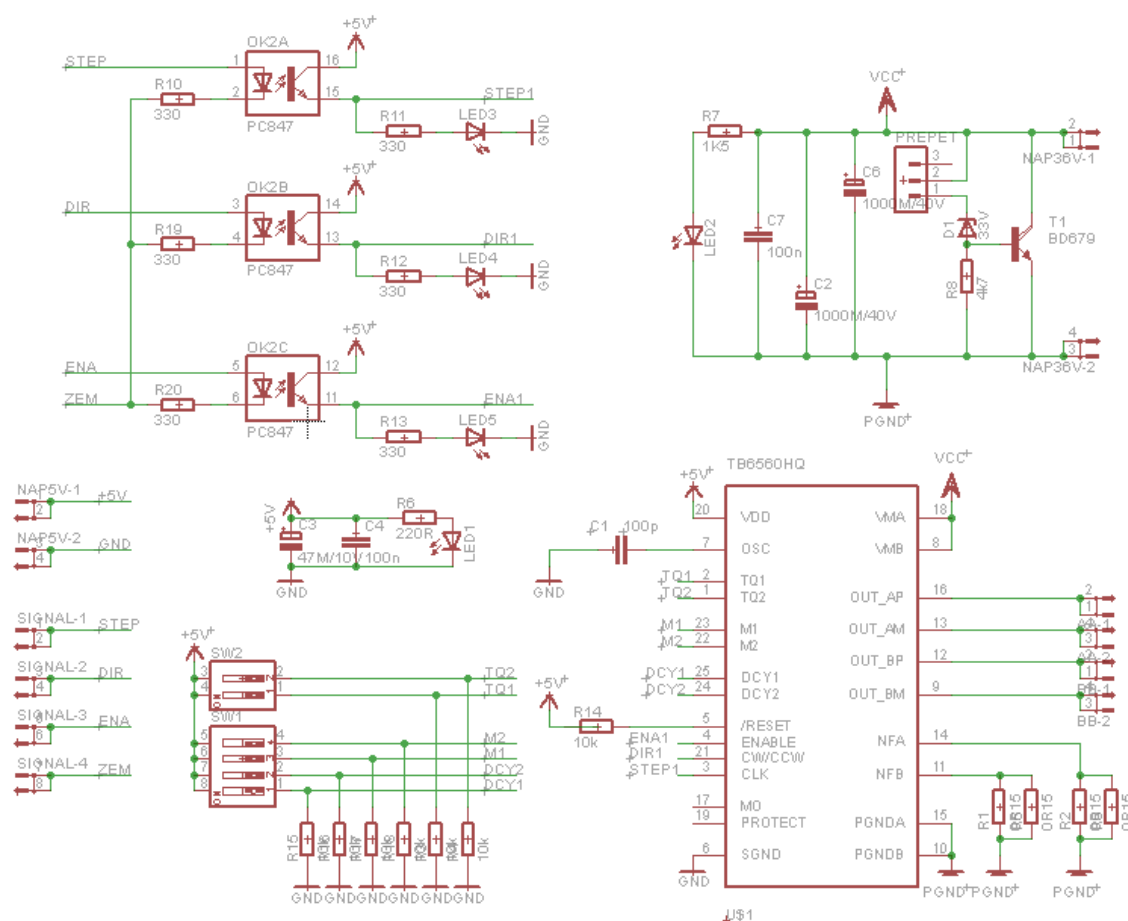
Tabulka 3: Tabulka nastavení proudu do cívek motoru. [4]



Obrázek 21: Blokové schéma vnitřního zapojení obvodu TB6560AHQ[4]

3.2.2 Schéma

Na schématu (obrázek 22) je vidět optočleny, přes které jsou přiváděny řídicí signály, každý signál má i indikační LED. Indikována jsou i obě napájecí napětí, pro logiku a pro výkonovou část. Je nutné zajistit dostatečnou filtraci napájecích napětí pomocí kondenzátorů C2, C6 a C7 pro výkonovou část a C3 a C4 pro 5V. Dále je zde zapojení pro omezení vyššího napětí, to určuje zenerova dioda, v tomto případě 33V. Vyšší napětí se může objevit při brzdění motoru. Ochrana je připojitelné přes jumper. Na schématu jsou dvě sady mikrosplínačů pro ovládání vlastností obvodu TB6560 a samotný obvod.



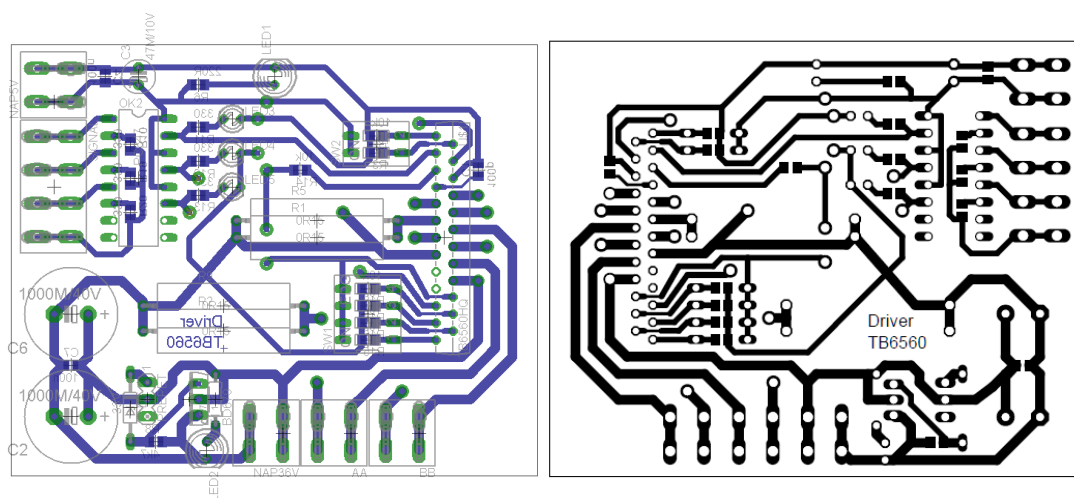
Obrázek 22: Schéma driveru s TB6560

3.2.3 Plošný spoj

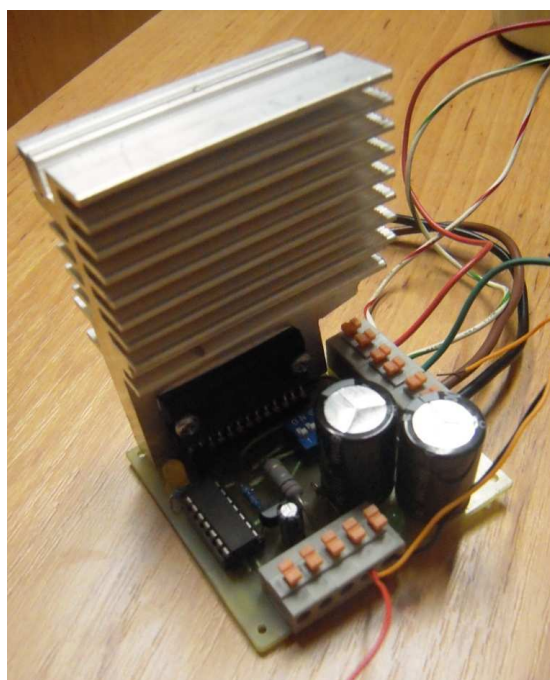
Návrh jednostranné desky plošných spojů byl problematický. Obvod TB6560 má v daném pouzdře rozložení pinů navrženo spíše pro oboustrannou desku kde je obvod připojen na hranu desky. Tedy jedna řada pinů je na vrchní straně a druhá na spodní. S výrobou těchto oboustranných desek jsem měl ale malou úspěšnost, proto jsem navrhl jednostrannou s drátovými propojkami, ty na obrázku 20 nejsou, kvůli přehlednosti. U první verze DPS jsem nedbal na rozvedení zemnicích ploch, což vedlo ke špatné funkci. Návrh jsem upravil, logická zem je spojena se silovou v jednom bodě. Výkonová část má zesílené vodiče. Optočlen je umístěn v patici. Po zvážení, že obvod TB6560 je vyroben s technologií CMOS, jsem i jej umístil do patice.

Obvod TB6560 je umístěn ke kraji plošného spoje, kvůli připojení chladiče. Ten je hliníkový s oboustranným žebrováním. Při provozu se nezahřál na více než 32°C. Aktivní chlazení tedy zatím není potřeba, je zde dostatečná rezerva i pro provoz v zátěži. Ventilátor bude třeba zvážit po uložení do nějaké krabičky.

Vývody jsou rozděleny na dvě skupiny. První výkonová, obsahuje čtyři zdířky pro připojení motoru a dvě pro napájené výkonové části. Druhá skupina je pro logické vstupy STEP, DIR a ENABLE a napájení 5V.



Obrázek 23: DPS – plán osazení a DPS předloha strany spojů



Obrázek 24: Hotový driver při testech

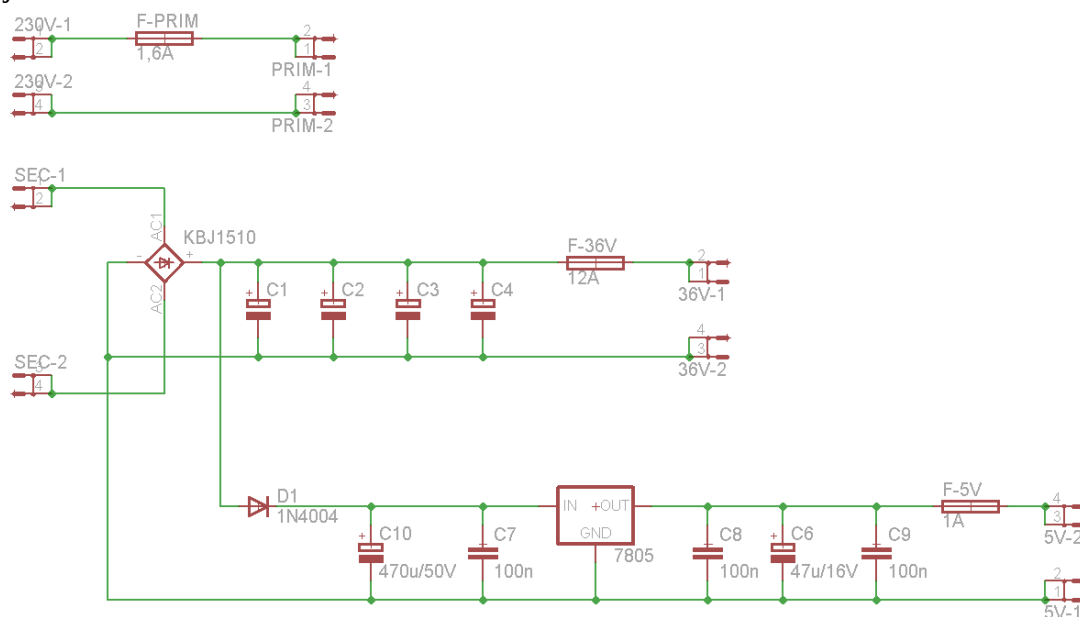
3.3 Zdroj

Pro napájení driveru je potřeba napětí mezi 12V – 40V. Platí, čím vyšší, tím vyšších otáček motor může dosáhnout. Vyšší napětí totiž rychleji protlačí proud cívkami. Zároveň je ale pro TB6560 nebezpečné napětí nad 40V. k nárůstu může dojít při brzdění. Požadovaný proud je sice nestálý (dochází ke spínání), ale doporučuje se mít jej dispozici. Pro tři drivery potřebuji ideálně 12A. Dále je potřeba napětí 5V pro logiku.

Podle uvedených kritérií jsem použil toroidní transformátor 230/22,5V pouze 10A, ale krátkodobě zvládne až 15A. Po usměrnění v můstku dostávám tedy stejnosměrné napětí 32V. To je třeba vzhledem k velkým proudům filtrovat velkou kapacitou. Použil jsem čtyři kondenzátory 10mF/50V. Stabilizace napětí není vzhledem k proudovému spínání třeba.

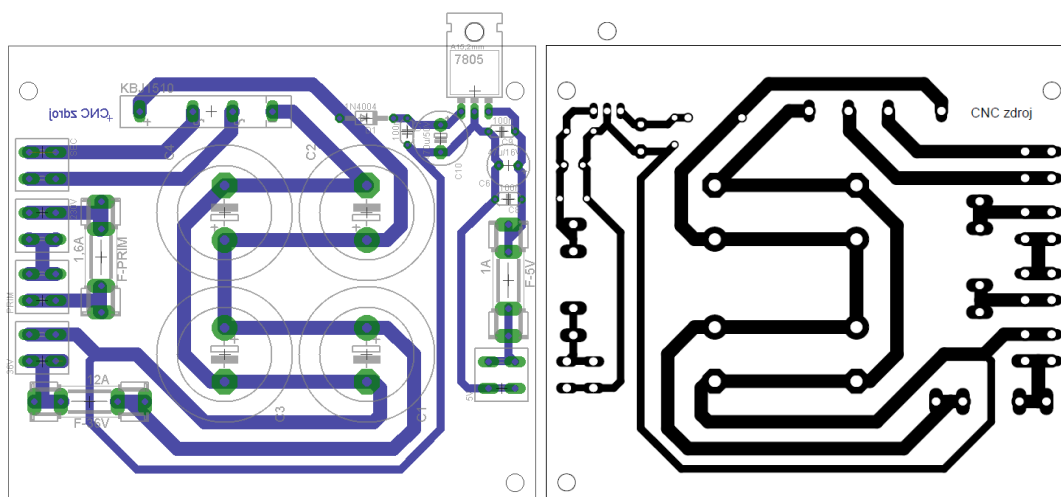
Pro logiku je stabilizováno 5V, maximálně pro odběr 1A. Viz schéma na obrázku 22.

Primární okruh je vybaven pojistkou 1,6A, sekundární 12A a výstup nízkého napětí je jištěn na 1A.

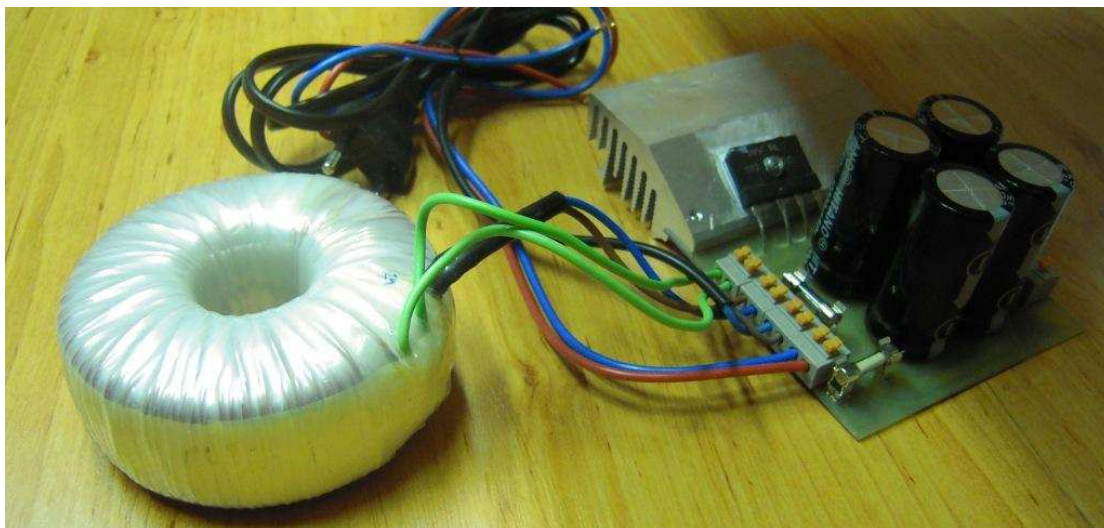


Obrázek 25: Schéma zdroje

Plošný spoj jen navržen jako jednostranný bez propojek. Usměrňovací můstek a stabilizátor jsou připojeny k chladiči na jedné straně DPS.



Obrázek 26: DPS zdroje a předloha strany spojů



Obrázek 27: Hotový zdroj

3.4 Koncové spínače a nouzový vypínač

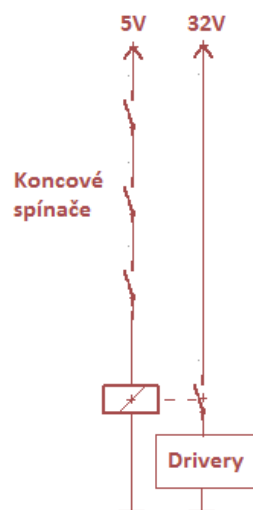
Koncové spínače os, zajišťují ochranu frézky před poškozením. A díky nim je možné stroj kalibrovat. Jsou umístěny na obou koncích os. Jsou dva typy koncových spínačů. První typ je softwarový, ten informuje řídicí jednotku o najetí na počátek osy, kalibraci. Řídicí jednotka má informace o délce os a počítá kroky oběma směry, nemělo by tedy nikdy dojít k najetí na koncový spínač mimo kalibraci.

Pokud ale řídicí jednotka nebo koncový spínač selže, a stroj se nezastaví při sepnutí spínače, je v druhé řadě umístěn spínač hardwarový. Ten u menších strojů jako je tento rovnou odpojí napájení pro drivery motorů a tím stroj zastaví. Zapojení je schematicky na obrázku 29.

V obou případech jde o rozpínací kontakty, tím se částečně předejde vadnému spínači. Nouzový vypínač je klasický central stop viz obrázek 30. Ten to vypínač drží kontakty relé, přes, které jde napájení do celého silového systému. Pokud je stisknuto, odpojí se napájení pro drivery (podobně jako v případě HW koncových spínačů) a navíc se vyšle signál do řídicí jednotky, aby měla informaci o problému.



Obrázek 28: Použitý koncový spínač [5]



Obrázek 29: Možné zapojení HW koncových spínačů



Obrázek 30: Nouzový vypínač

4 ŘÍDICÍ JEDNOTKA

Řídicí jednotka umožňuje generování potřebných impulsů STEP a DIR pro drivery krokových motorů, podle ručního zadávání nebo dle ISO kódu. Slouží tedy jako HW interpolátor.

Program – ISO kód dokáže přijmout z PC po emulované sériové lince. Uloží ho do své paměti a může jej vykonávat. Zajišťuje také referování, manuální posuvy, detekce chyby, což může být problém v programu, či najetí na koncový spínač.

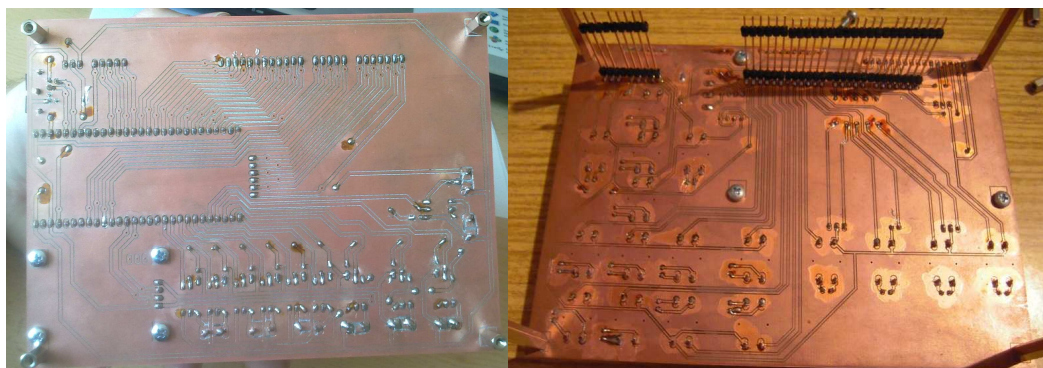
Jedná se o pouze o tzv. 2,5 osý interpolátor. Umožňuje gravírování/frézování v osách X a Y, u osy Z lze nastavovat její pozici, ale nelze vykonávat interpolaci v kombinaci s jinou osou. To ale na většinu požadovaných výrobků stačí.

4.1 Koncepce jednotky

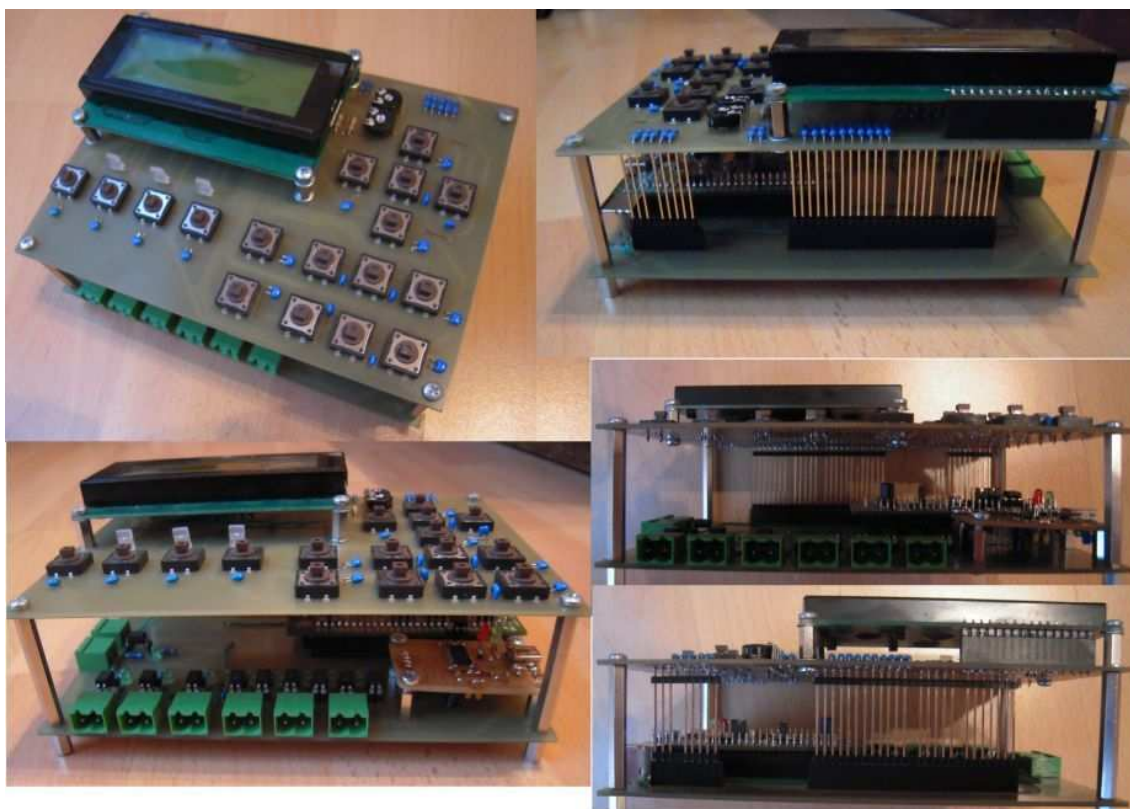
Jádrem interpolátoru je 32 bitový mikrokontroler ARM. Pracuje na frekvenci 24MHz. Další důležitou částí je zajištění komunikace s PC. To jsem zajistil pomocí obvodu od firmy FTDI. Jedná se o převodník FT232RL. Důležité je i galvanické oddělení od driverů, ve kterých se používá napětí desítek voltů, a to by bylo pro logické obvody v případě poruchy zničující. To je zajištěno pomocí optočlenů. Opticky odděleny jsou i vstupní vodiče, koncové spínače a ESTOP (bezpečnostní tlačítko). Dalšími součástmi je rozměrný LCD displej (4x20 znaků) a ovládací tlačítka.

Konstrukce je řešena modulárně, jedná se o sendvičovou konstrukci dvou desek plošných spojů nad sebou, propojených pevnými piny, viz obrázek 32. Jednotlivé moduly jsou: DPS s mikrokontrolerem – celý vývojový kit, komunikační modul, display. Optočleny jsou z důvodu snadné výměny umístěny do patič.

Obě desky plošných spojů jsou frézované na popisované CNC frézce. Byla použita gravírovací jehla 3,15mm s úhlem 30°. Výsledné plošné spoje na obrázku 31.



Obrázek 31: Plošné spoje jsou frézované



Obrázek 32: Řídicí jednotka

4.2 Vývojový kit STM32 Discovery Value Line

Jak je zmíněno v minulé kapitole, pro řídicí jednotku jsem se rozhodl použít 32bitový ARM procesor. Kvůli ceně a velmi dostupnému vývojovému kitu jsem se přiklonil k verzi od firmy ST, procesory řady STM32F100. Procesor je vybaven všemi standardními periferiemi a vlastnostmi. Zajímavé jsou [6]:

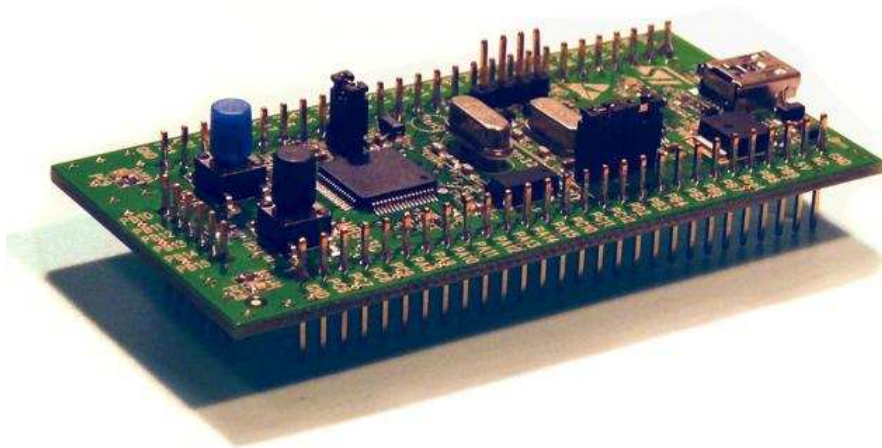
- Maximálně 7 PWM 16-bit timerů, včetně rozšířeného ovládání, pro celkem max. 26 kanálů PWM
- Tři nezávislé PWM 16-bit timery s komplementárním výstupem a generací dead-time
- 12-bit dual channel DAC s podporou DMA
- 400 kHz I²C, do 12 Mbit/s master and slave SPI, do 3 Mbit/s USART
- 7-kanálové DMA
- CRC (cyclic redundancy check) s podporou DMA
- Zabudováno POR (power-on reset) , PDR (power-down reset) , LVD a watchdog

- z výroby nastavené RC oscilátory: 8 MHz RC pro mcu a 40 kHz pro RTC a watchdog
-

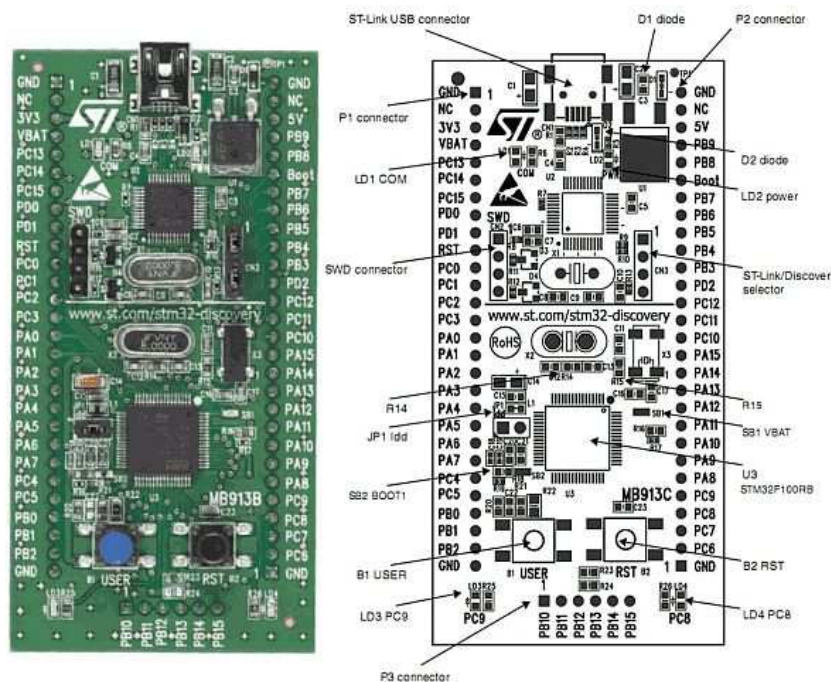
Firma ST vyrábí i velmi zajímavý vývojový kit s tímto procesorem. STM32 Value Line Discovery je vybaven jedním z levnějších a méně výkonných zástupců rodiny procesorů Value Line, ale pro vývoj by měl být dostačující. Je v 64pinovém pouzdře, tudíž nabízí dostatek V/V pinů. Součástí kitu je programátor, který se připojuje k PC přes miniUSB konektor, ten zajišťuje i napájení celého kitu. Tento integrovaný programátor lze pomocí jumperů přepnout na jiné piny a na ně připojit i externí procesor, což je pro budoucí konstrukci řídicí jednotky velmi výhodné.

Na kitu jsou integrovány i dvě tlačítka, jedno slouží k resetování a druhé je uživatelské, je možno na něm zkusit první programy. Ze stejného účelu jsou zde i dvě SMD led diody. Všechny piny procesoru jsou vyvedeny po stranách vývojového kitu. Zde je zajímavé, že konektory jsou vyvedeny oboustranně. Nevýhodou je šestice pinů na spodní straně kitu, tímto nešikovným umístěním je nemožné využít je v nepájivém kontaktním poli.

Pro konstrukci řídicí jednotky jsme se rozhodl použít celý vývojový kit jako hotový modul. A to vzhledem k jeho nízké ceně a složitosti a obtížím pájení mikrokontroleru v malém SMD pouzdře.



Obrázek 33: Vývojový kit STM32 Value Line Discovery [6]



Obrázek 34: Horní pohled na vývojový kit a jeho popis [6]

4.3 Komunikační modul

Pro spojení s PC jsem vybral ověřenou možnost využít obvod FT232RL od firmy FTDI. Tento obvod vytváří v operačním systému po nainstalování příslušných ovladačů virtuální sériový port. Sériový port je velmi snadný na obsluhu, co se týče programování. Já využil možnost Visual Studio 2010, kde je připravená komponenta „serialPort“.

Komunikace tedy mezi PC a obvodem FT232 fyzicky probíhá po sběrnici USB. Obvod slouží jako převodník mezi USB a USART. Vývody TxD a RxD USARTu v FT232 jsou připojeny na patřičné vývody mikrokontroleru ARM.

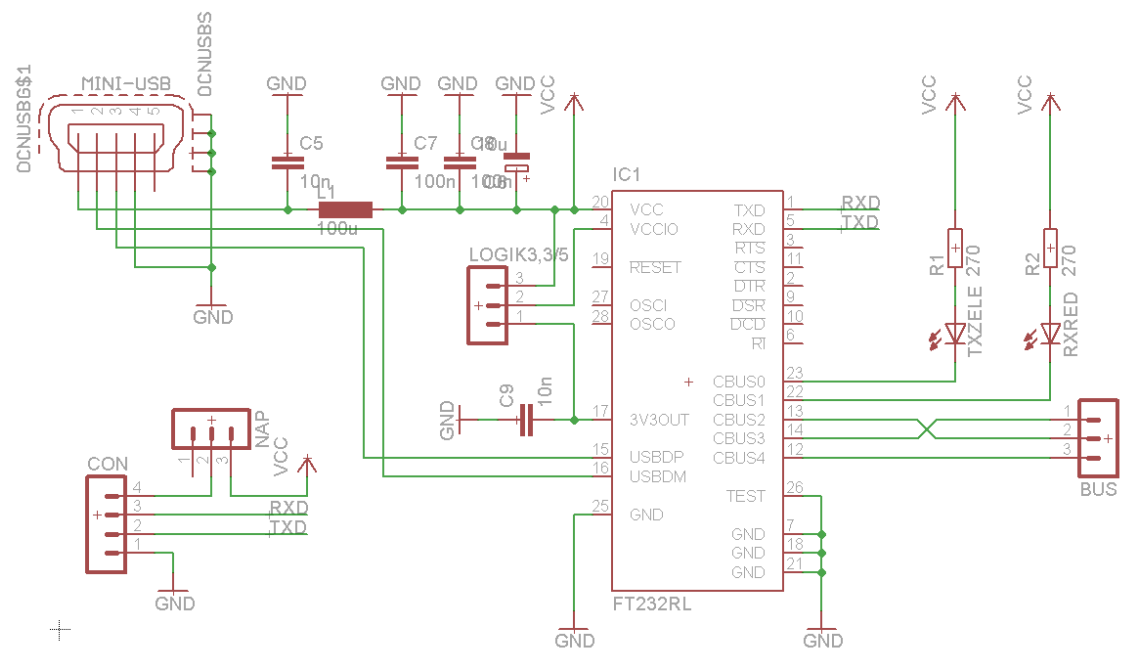
FT232Rl umožňuje komunikovat po sběrnících USB1.1 i USB2.0, Rychlost přenosu se může pohybovat v rozmezí 300 baudů – 3 Mbaudů.

Obvod má defaultně z výroby uloženy v paměti některé parametry. Ty je možné měnit. Dá se nastavit požadovaný proud z USB, identifikátory (VID a PID), název obvodu, výrobce, sériové číslo. Je možné využít některý s vývodů BUS jako zdroj hodinového signálu, jehož frekvence se dá nastavit. Ke konfiguraci slouží program MProg přímo od výrobce.

Napájení obvodu je 5V, to je se nabízí přímo na sběrnici USB. Dají se měnit i napěťové úrovně signálu (v rozsahu 1,8 – 5V), dle trojaké napětí se přivede na pin VCCIO (psáno podle [9]).

4.3.1 Schéma komunikačního modulu

Základem jak již bylo řečeno je obvod FT232RL, tato verze obvodu vyžaduje jen minimum externích součástek. Napájen je s USB v PC, ve schématu je jen filtrace napájení, dle datasheetu[9]. Modul jsem navrhl univerzálně, proto je zde možnost přepínání mezi 5voltovou a 3,3voltovou logikou pomocí jumperu na konektoru „LOGIK“. Pro použití v řídicí jednotce bude nastaven na 3,3V. Použitý konektor k připojení k PC je USB-MINI (stejný jako na vývojovém kitu ST). K propojení s řídicí jednotkou slouží konektor „CON“ a případně „BUS“. Na „CON“ jsou vyvedeny komunikační bity RxD a TxD, dále GND a případně napájení 5V, to se dá aktivovat opět pomocí jumperu na konektoru „NAP“. Na pinech BUS0 a BUS1 jsou připojeny kontrolky RxD a TxD.



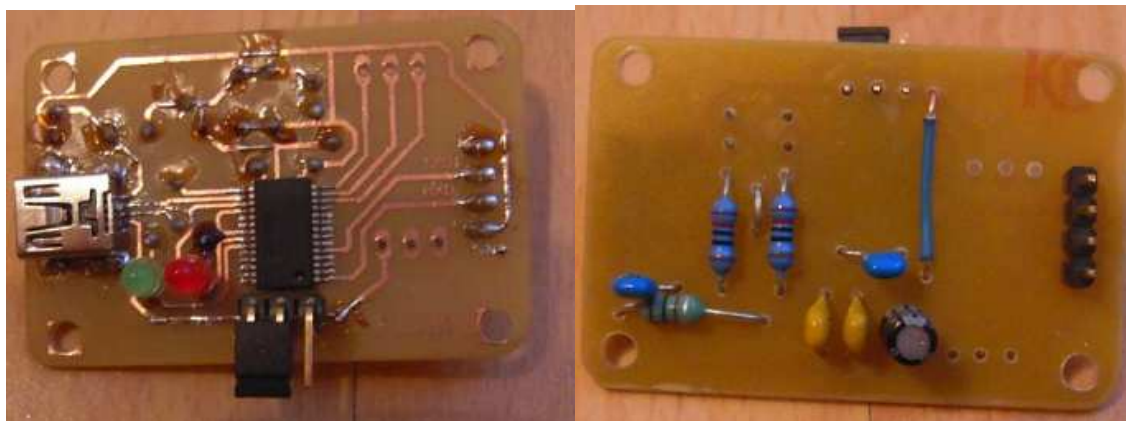
Obrázek 35: Schéma Komunikačního modulu

4.3.2 DPS komunikačního modulu

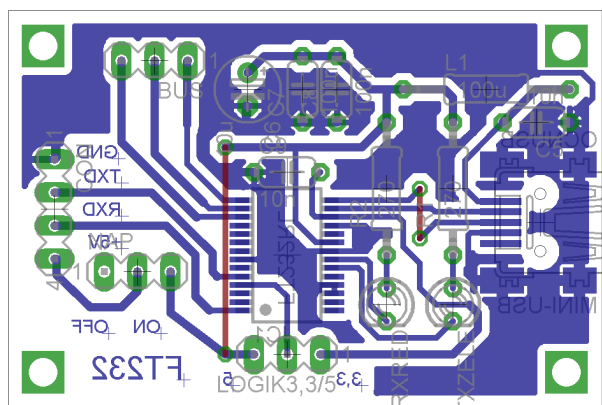
Plošný spoj komunikačního modulu je navržen jako jednostrany s dvěma propojkami. Vzhledem k záměru použít stejný konektor jako je na modulu s mikrokontrolerem jsem se rozhodl plošný spoj orientovat nestandardně stranou spojů vzhůru. Ve spodní straně jsou tedy umístěny součástky jako, kondenzátory, cívka a odpory. Dále je zde konektor pro spojení s řídicí jednotkou („CON“). Druhý konektor „BUS“ není vůbec osazen, protože v této aplikaci nemají signály BUS využití.

Na vrchní straně (straně spojů) je osazen samotný obvod FT232RL a SMD MINI USB konektor. Kvůli orientaci plošného spoje byly vyvedeny i LED diody 3mm na této straně. Dále je zde třtinový konektor s jumperem k volbě mezi 3,3voltovu a 5voltovou logikou.

V návrhu bylo počítáno s „rozlitím mědi“. Na předlohu pro fotovrstvu jsem ji ale nakonec neudělal, kvůli snadnějšímu pájení obvodu FT232.



Obrázek 36: Hotový komunikační modul

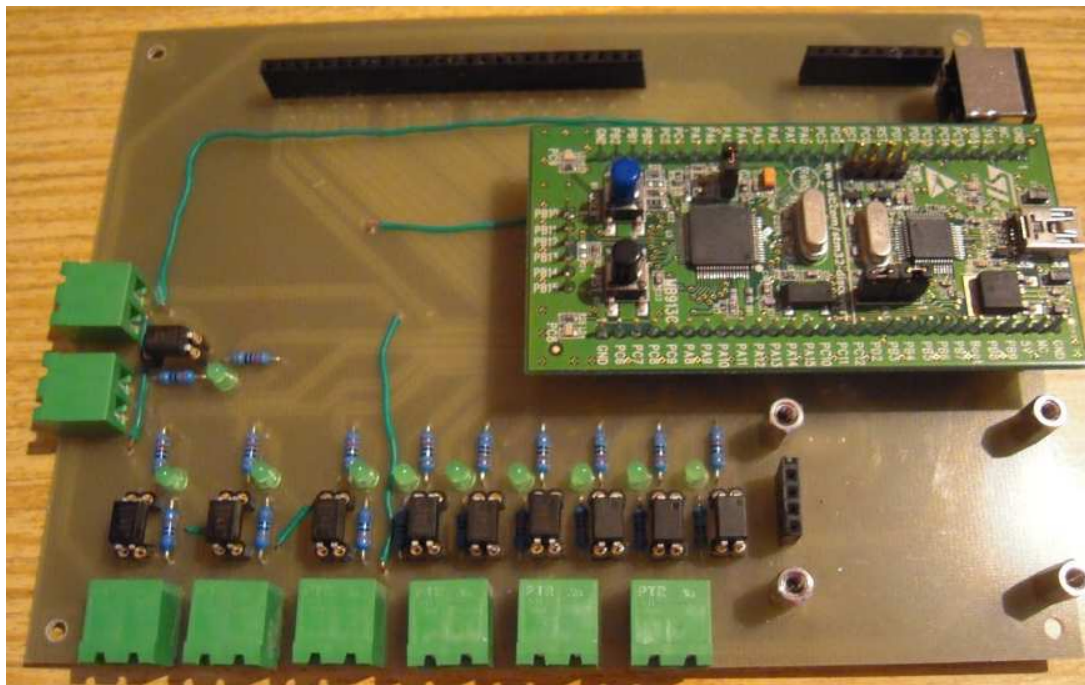


Obrázek 37: Osazovací plán komunikačního modulu

4.4 Spodní panel

Spodní panel je v podstatě jakási „spojka“ jednotlivých modulů s horním panelem a optickým oddělením. Obsahuje optické oddělení pro šest výstupních pinů a pro čtyři vstupy. Svorkovnice pro připojení jsou úhlové, kabely se do nich upevňují šroubem. Svorkovnice je dvoudílná, kabely od panelu lze snadno oddělit bez šroubování. Moduly mikrokontroleru a komunikačního modulu jsou umístěny do patič a v případě modulu

s FT232 ještě upevněny pomocí distančních sloupků. V horní části desky je řada patič pro spojení s horním panelem, jde o signály k displeji, tlačítkům a kontrolkám. Na panelu je osazen i konektor PS/2, ale není využit.

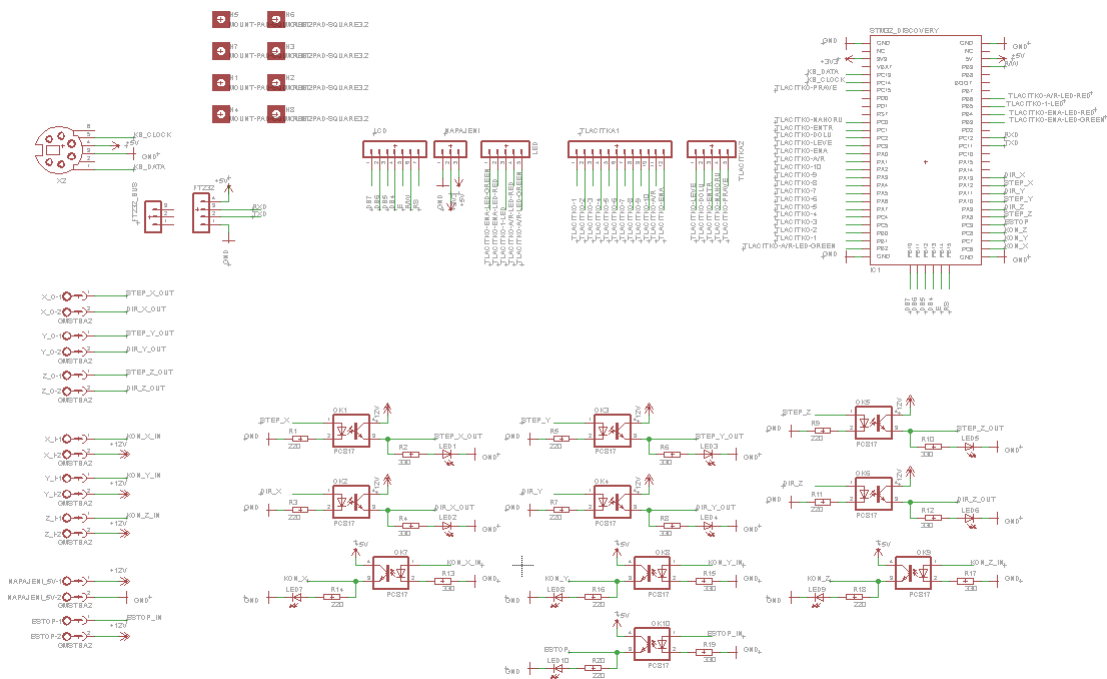


Obrázek 38: Spodní panel již s umístěným modulem mikrokotroleru

4.4.1 Schéma spodního panelu

Pro modul s mikrokontrolerem – vývojový kit, je k dispozici knihovna do návrhového programu Eagle. Je ke stažení zde [10]. Dále jsem vložil konektor pro spojení s komunikačním modulem. Jsou zde konektorové lišty pro propojení s horní částí panelu.

Panel obsahuje deset optočlenů, čtyři oddělovací vstupy a šest oddělovací výstupy. Každý optočlen je na výstup vybaven kontrolkou v podobě třímilimetrové LED diody. Zároveň optočlen zastává funkci převodu napětí. Vnější napětí signálů je 5V a vnitřní 3,3V. Pět voltů pro vnější signály musí být přivedeno externě, například z nějakého adaptéru.



Obrázek 39: Schéma spodního panelu (podrobněji v příloze)

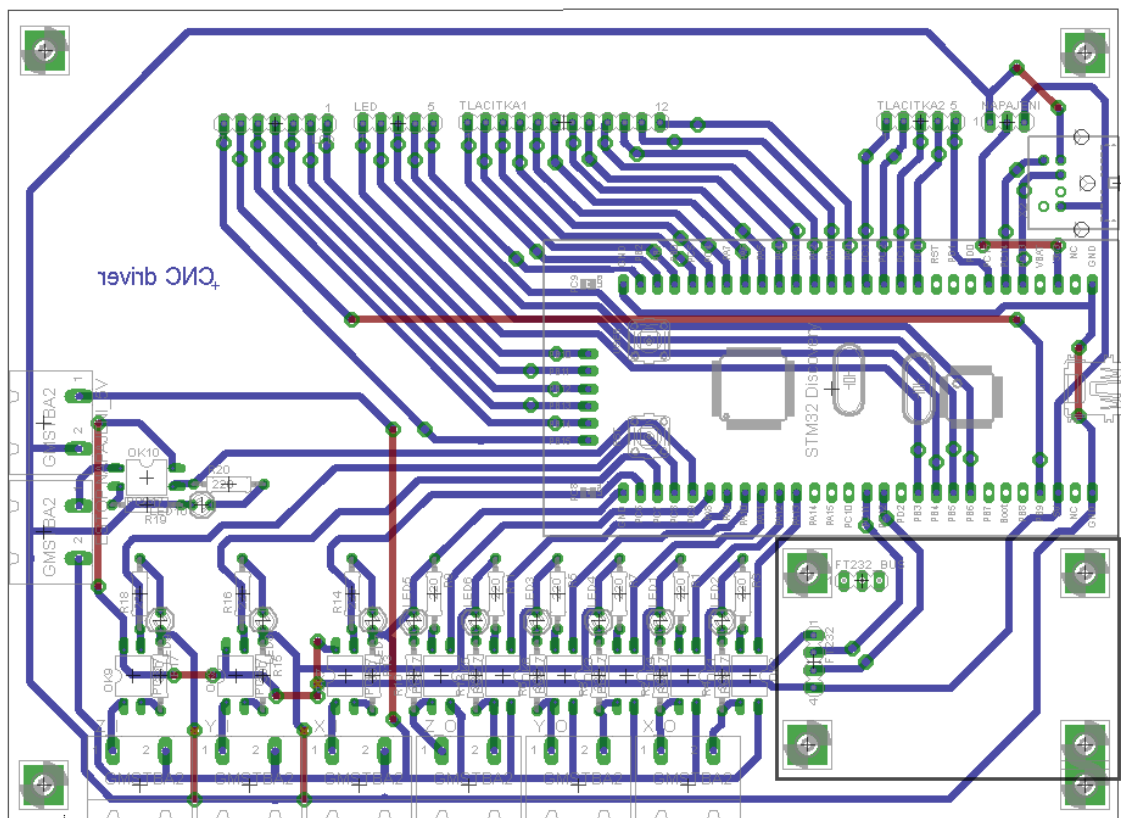
4.4.2 DPS spodního modulu

Plošný spoj jsem navrhoval tak, aby jej bylo možné vyrobit na CNC frézce. Cesty jsou proto široké 1mm a mezi nimi jsou dostatečné mezery. To bohužel mělo za následek větší množství propojek, protože nebylo možné vést cesty mezi nožičkami IO obvodů nebo konektorů.

V pravé části (vzhledem k obrázku 40) jsou moduly mikrokontroleru a komunikační. Ve spodní části je řada svorkovnic pro koncové spínače a ovládání motorů. U každého vstupu/výstupu jsou patřičné optočleny s kontrolkou. V levé části je pouze svorkovnice pro napájení signálů a vstup pro ESTOP signál.

V horní části je konektor pro pojení s horním panelem, šest signálů pro display, devatenáct pro tlačítka a pět pro kontrolky. Poslední tři piny vpravo jsou 3,3V, 5V a GND pro horní panel. V rozích jsou otvory pro připevnění distančních sloupků, další pak pro připojení komunikačního modulu.

Vedle modulu mikrokontroleru a vrchní řady konektorů jsou ještě přidány piny pro případné změny při odrušování a ladění.



Obrázek 40: Osazovací plán DPS spodního panelu

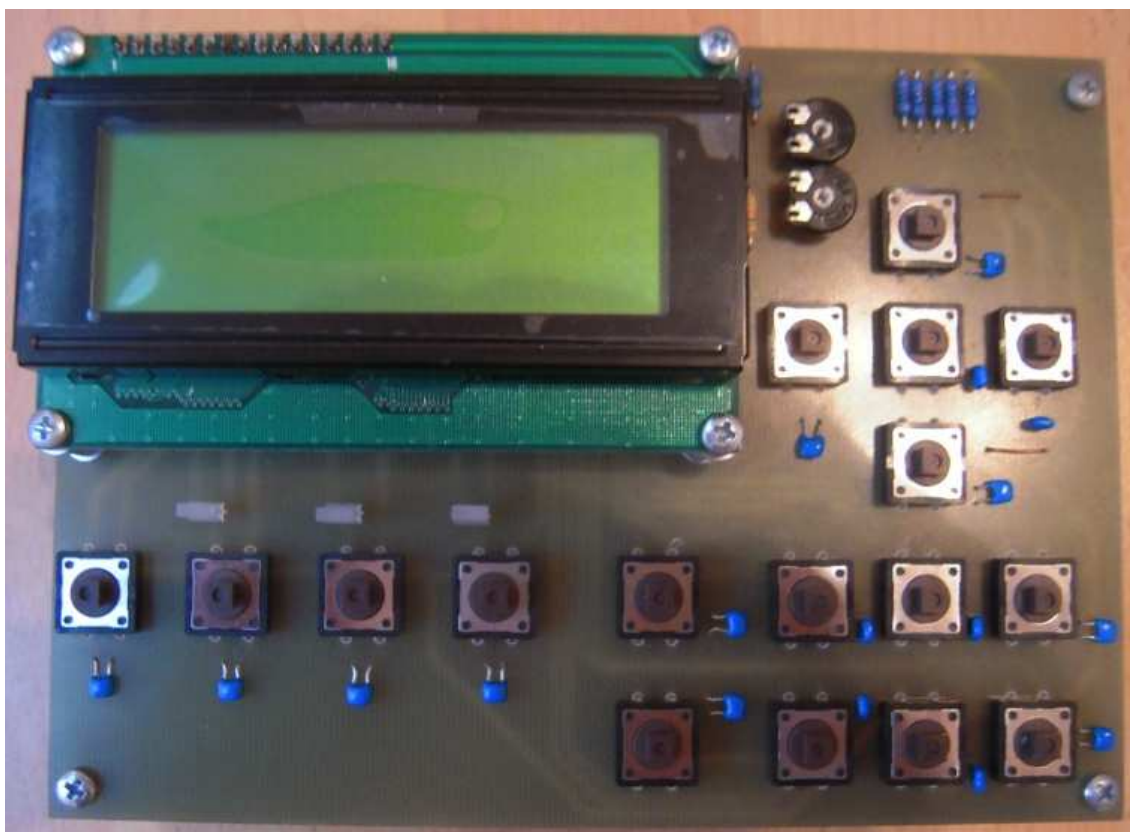
4.5 Horní panel

Vrchní panel, je ovládací částí řídicí jednotky. Dominantní je velký display 4x20 znaků s řadičem HD44780. Vpravo jsou dva trimery pro ovládání jeho kontrastu a jasu podsvícení. V pravé části jsou směrová a potvrzovací tlačítka.

Pod displejem jsou čtyři tlačítka pro volbu režimů jako reset, manuální mód, kalibrace. U nich jsou kontrolky pro přehlednost spuštěných režimů. Vpravo dole jsou dvojice tlačítek pro manuální posuv os X, Y a Z a uprostřed pro změnu rychlosti posuvu.

V rozích plošného spoje jsou umístěny otvory pro připojení distančních sloupků. Jeden je společný i pro distanční sloupek displeje, zbylé tři jsou samostatně.

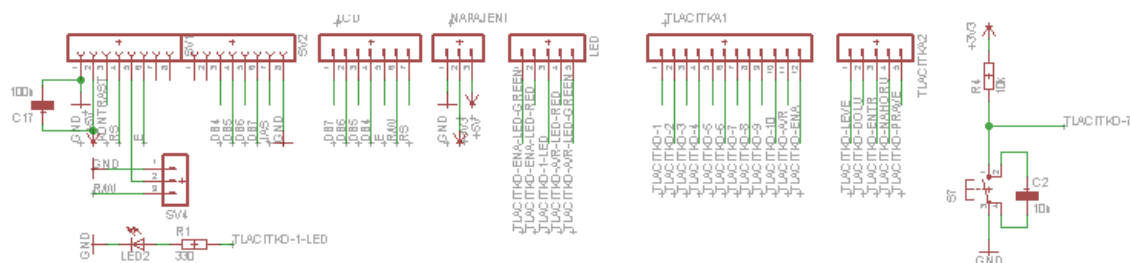
Pod displejem je navíc umístěn přepínatelný jumper pro nastavení bitu W/R displeje. Vzhledem k faktu, že z displeje data nenačítám, ale jen zapisuji, je bit trvale uzemněn.



Obrázek 41: Horní panel s připevněným displejem

4.5.1 Schéma horního panelu

Pod textem je zobrazena jen část schématu, kompletní je v příloze. Schéma zapojení tlačítka a kontrolky se stále opakuje. Zajímavé je pouze zapojení konektoru, který spojuje horní panel se spodním. U displeje je zajímavá již zmíněná možnost trvalého uzemnění bitu W/R.

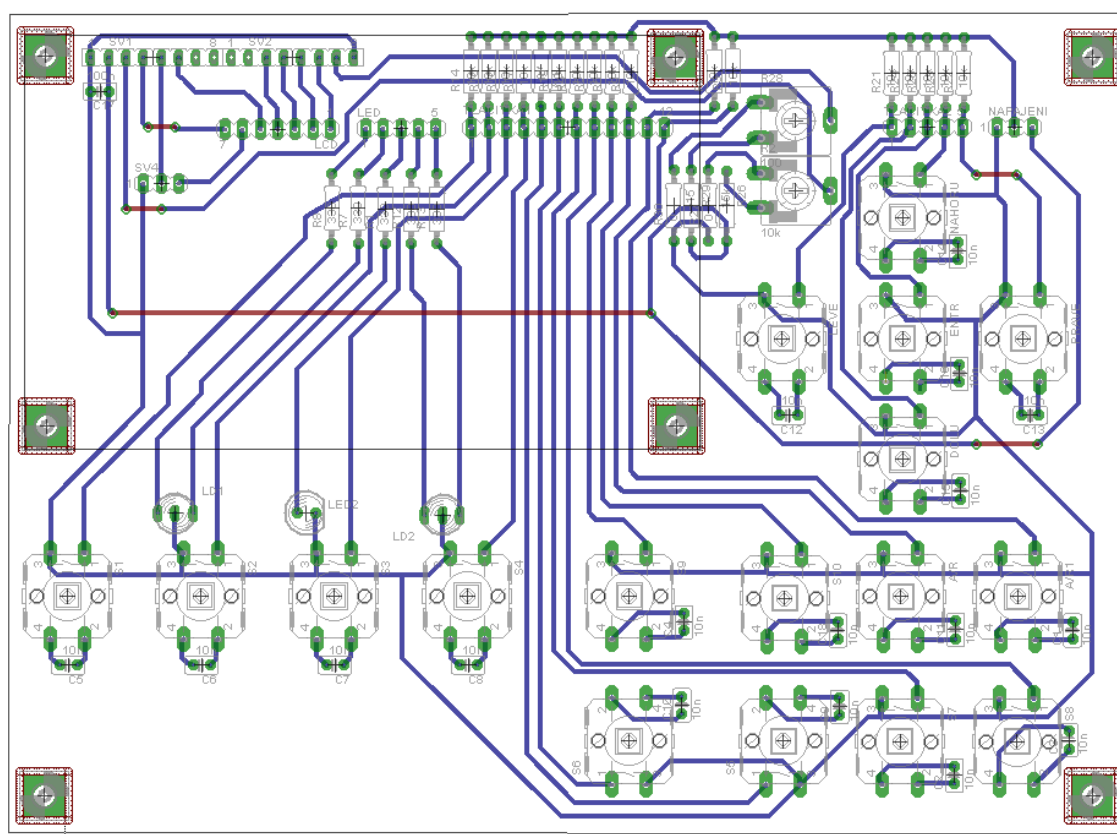


Obrázek 42: Část schématu horního panelu

4.5.2 DPS horního panelu

Tento plošný spoj byl první, který jsem frézoval. Není navržen optimálně jako DPS spodního panelu. Cesty nejsou navrženy dostatečně široké, po frézování jsou tedy výrazně slabší. Některé mezery byli příliš malé a málem došlo k úplnému odfrézování potřebných ploch. Ale po drobných úpravách je DPS stoprocentně funkční.

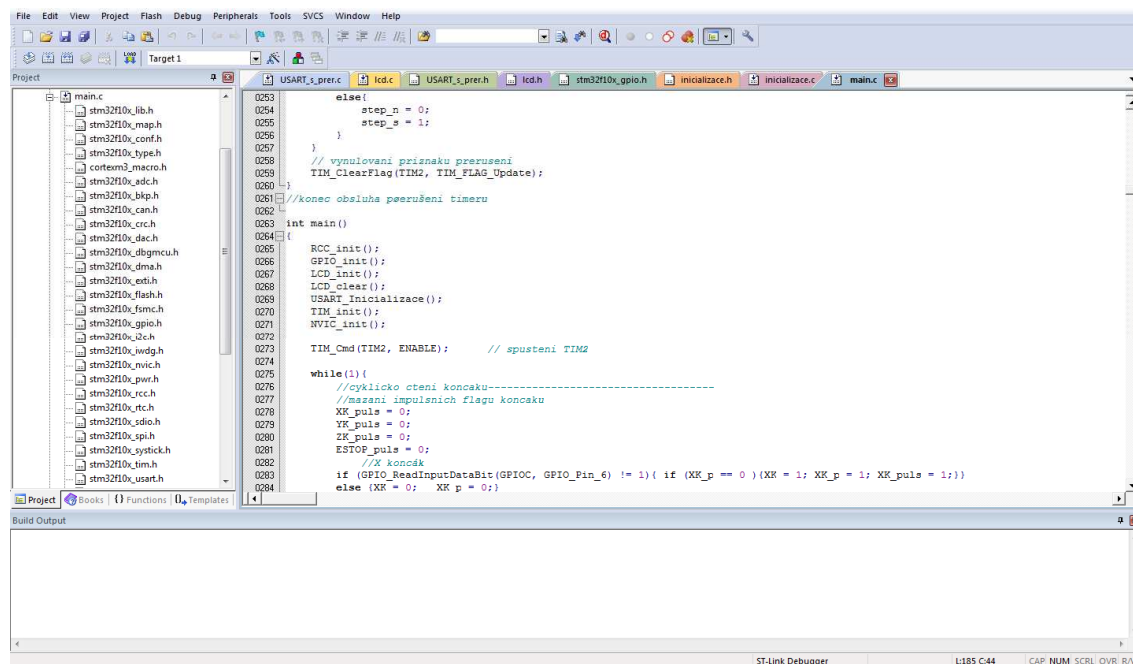
Při osazování je nutné zapojit čtyři krátké a jednu dlouhou propojku – viz osazovací plán.



Obrázek 43: Osazovací plán DPS horního panelu

5 PROGRAMY PRO ŘÍDICÍ JEDNOTKU

Důležitou částí, bez které by hardware nemohl fungovat, je program pro mikrokontroler a PC. Pro program pro mikrokontroler STM32F100 jsem použil studio Keil uVision 4. Jedná se o program zdarma (do 32KB programu) a měl jsem s ním již zkušenosti s prací na osmibitových procesorech.



Obrázek 44: Okno programu Keil uVision 4

Pro ovládací program pro PC jsem zvolil produkt Microsoftu – Visual Studio 2010. Program jsem psal v jazyce C#.

Komunikace mezi PC a řídicí jednotkou je programově řešena sériovou linkou. Visual Studio nabízí komponentu serialPort a v mikrokontroleru je ovládání USARTu také velmi snadné.

5.1 Koncepte programu pro mikrokontroler

Program pro mikrokontroler je koncipován tak, aby byla jednotka funkční i bez připojeného PC.

Celý program je navržen jako stavový automat. Jednotlivé stavy se přepínají na základě stisku tlačítek nebo přijatých dat. Celkem je k dispozici šest stavů: Init, Pripraveno, Nahravani programu, Frezovani, Referovani a Chyba. Stavy se vypisují na poslední řádek displeje. Stav Frézování má ještě tři podstavy – módy. Ty prezentují jednotlivé příkazy pro frézování: interpolace os X a Y, interpolace osy Z a konec programu.

Po zapnutí napájení se jednotka přivede do stavu „Init“, kde vyčkává, dokud se netiskne tlačítko reset, či nepřijde zpráva od PC, o jeho připojení. Po té přejde do stavu „Připraveno“.

V tomto stavu setrvává do dalšího požadavku – stisk tlačítka, příchozí pokyn přes USART. Tento stav je výchozí pro všechny ostatní stavy a také se do něj ze všech stavů vrací.

Při stisku tlačítka pro referování dojde k přepnutí do stavu „Referovani“ a okamžitě se začne generovat obdélníkový signál na všechny výstupy STEP. Všechny osy se tedy začnou pohybovat směrem ke koncovým spínačům, které teď poslouží jako referenční. Signál pro jednotlivé osy je generován, dokud nedojde k najetí na referenční spínač, tedy dokud nepřijde signál od koncového spínače dané osy. Po té se signál na dané ose generuje dál ale jen určitý – nastavený počet kroků (představující ujetou vzdálenost), ale nastaví se výstup DIR, čímž se změní směr pohybu. Osy tedy sjedou ze svých koncových spínačů a jsou zreferovány. Až se tak stane u všech os, přepne se opět do stavu „Připraveno“.

Pokud chceme nahrát program, musí být připojeno PC, jinak do stavu „Nahravani programu“ nelze přejít. Pokud tedy je připojeno PC, v ovládacím programu zvolíme textový soubor, který se má odeslat do řídicí jednotky. Ten se po řádcích odesílá jako znakový řetězec. Mikrokontroler program přijímá dokud nepřijde ukončovací příkaz „M2“. Jinak procesor přijatá data v tomto stavu nekontroluje. Po nahrání přejde procesor opět do stavu „Připraveno“.

Frézování aktivujeme buď stiskem tlačítka, nebo příkazem z PC. Po té přejde do stavu „Frezovani“ a ihned začne vykonávat frézovací program. Vypíše jej na displej, u aktuálně prováděného řádku je zobrazena šipka. Program vyhodnotí, o jaký příkaz se jedná, převede patřičné znaky reprezentující parametry příkazu na čísla a přejde do příslušného módu. Pokud jsou příkazy nesmyslné nebo nepodporované přejde mikrokontroler do stavu „Chyba“. Jinak vykonává řádky kódu, dokud nenarazí na příkaz „M2“, který program ukončuje. Po té se přejde do stavu „Připraveno“.

Když chceme ručně posouvat CNC, musíme přejít do stavu „manual“. A to buď tlačítkem manual nebo příkazem s PC. V tomto stavu se po stisku příslušného tlačítka pro posuv generuje signál STEP a případně dle požadovaného směru signál DIR. V případě že dojde k najetí na koncový spínač, přejde jednotka do stavu „Chyba“, lze pak znovu přejít do režimu manuálního posunu a to z důvodu sjetí s koncového spínače, pak už je ochrana na obsluhu. Vypnout manuální režim lze pak stejným tlačítkem jako při zapínání. Nebo příkazem z PC.

Celková rychlost frézování či manuálního posuvu lze kdykoli měnit pomocí dvojice tlačítek pro změnu rychlost.

Chybový stav, pokud nastane, vynuluje běh programu a také příznak kalibrace. Po chybě je tedy nutné znovu referovat.

5.2 Program pro mikrokontroler

Program se skládá z hlavní funkce main a třech vytvořených knihoven. Knihovna inicializace.h slouží pro nastavování V/V, časovačů, systémových hodin a priorit. Knihovna USART_s_prer.h definuje nastavení USARTu. A knihovna lcd.h slouží k obsluze displeje.

Komunikace probíhá po řetězcích ukončených znakem „0x0d“. Pokud na konci řetězce nepřijde tento znak, nejsou data platná a jsou zahozena. Každá platná přijatá data jsou potvrzovaná vysláním řetězce „OK“.

5.2.1 Knihovna inicializace.h

Knihovna obsahuje tyto funkce:

```
void RCC_init(void);  
void GPIO_init(void);  
void TIM_init(void);  
void NVIC_init(void);
```

RCC_init nastavuje hodinový systém a aktivuje pouštění hodinových signálů do periférií (bran, časovačů, usartu). Defaultně je vše zakázáno, z důvodu úspory energie.

GPIO_init, zde jsou definovány, které piny jsou vstupní, jaký mají odpor a chování (například pro tlačítka jsou v režimu pull up). Velmi podobné je to pro výstupní piny. Každou bránu je nutné definovat zvlášť.

TIM_init slouží k nastavení časovače, je nastaven na čítání nahoru s autoreloadem. Tato funkce se volá pokaždé změně rychlosti, protože je nutné změnit hodnotu děličky vstupní frekvence.

NVIC_init definuje priority přerušení. V programu jsou využity dvě přerušení: časovač a USART, vyšší priorita je nastavena USARTu, aby nedocházelo ke ztrátě dat.

5.2.2 Knihovna USART_s_prer.h

Knihovna obsahuje pouze funkci:

```
void USART_Inicializace(void);
```

Ta zajišťuje nastavení USARTu. Rychlost 115200 Baudů, 8 datových bitů, jeden stop bit, žádná parita a žádná kontrola toku dat.

5.2.3 Knihovna lcd.h

Definuje port na kterém LCD bude připojeno a významy pinů, displej je obsluhován pomocí čtyř bitové komunikace. A řídicí bit R/W je trvale uzemněn – tedy přepnut na zápis.

```
#define LCD_port GPIOB
#define LCD_RCC RCC_APB2Periph_GPIOB
#define RS GPIO_Pin_15
#define E GPIO_Pin_14
#define DB7 GPIO_Pin_10
#define DB6 GPIO_Pin_11
#define DB5 GPIO_Pin_12
#define DB4 GPIO_Pin_13
```

Dále obsahuje tyto funkce:

```
void LCD_init(void);
void LCD_clear(void);
void LCD_char(char znak);
void LCD_string(char *text);
void LCD_yx(uint8_t y, uint8_t x);
```

LCD_init pošle sekvenci příkazů k inicializaci displeje, například nastaví 4bitovou komunikaci.

LCD_clear smaže vše na displeji.

LCD_char odešle jeden znak na aktuální pozici.

LCD_string pošle řetězec znaků na aktuální pozici.

LCD_yx přemístí kurzor na zadanou pozici.

5.2.4 Hlavní funkce main

Na začátku je definováno velké množství proměnných. Protože není podporován datový typ bool, je místo něj využívána nejmenší možný typ unsigned integer a to 8 bitový (uint8_t). pro proměnné na ukládání hodnot posuvu a počtu impulsů se využívají datové typy (int16_t a int32_t).

Pod deklarací proměnných jsou tyto funkce:

```
void Strcpy(uint8_t *d1, const uint8_t *s1)
uint8_t Strcmp (const uint8_t * s1, const uint8_t * s2)
int16_t ABSS(int16_t cislo2)
uint16_t prevod_na_cislo()
```

```
uint8_t Vypis (uint8_t poz, uint8_t radek)
void USART1_IRQHandler(void)
void TIM2_IRQHandler(void)
```

Funkce Strcpy slouží pro kopírování obsahů řetězců.

Funkce Strcmp porovná dva řetězce a pokud jsou shodné vrátí hodnotu 1.

ABSS je funkce pro výpočet absolutní hodnoty.

Funkce převod na číslo převede zadaný řetězec na číslo.

Funkce Vypis se stará o zobrazení programu na display.

USART1_IRQHandler je volán vždy při přerušení od usartu tedy při příjmu dat. Zde se vykonává zápis do řetězce a pokud přijde znak 0x0d (zvolený znak na ukončení posílání dat). Po té se nastaví příznak příjmu dat.

TIM2_IRQHandler je funkce volaná po přerušení od časovače. Slouží jen ke generaci obdélníkového signálu pro výstupy STEP.

Pak už následuje samotná funkce main. Zde se zavolají všechny výše popsané inicializační funkce a funkce pro smazání displeje. Zapne se časovač a pak už následuje nekonečný cyklus while.

Na začátku cyklu se provede scan všech vstup s tlačítky, koncovými spínači a estop. Pro všechny je funkce stejná, proto popíši jen jednu. Každý vstup reprezentují dva flagy, jeden je nastaven po celou dobu stisku tlačítka a druhý se nastaví jen po zmáčknutí pro jeden cyklus a pak se resetuje. Lze ho opět nastavit jen dalším stiskem.

Následuje kontrola a úprava rychlosti posuvu. Pokud se rychlost změní ať programově nebo manuálně tak zde se přepočítá i změna hodnoty děličky časovače, změní se a časovač se znovu ziniculuje.

Po programu věnovanému rychlosti následuje hlídání vstupu ESTOP, pokud přijde signál je ihned nastaven stav „chyba“.

Následuje kód pro stav „Init“. Zde se jen vypíše úvodní text na displej a čeká se na stisk tlačítka nebo příjem dat od PC.

Pokud je nastaven stav nahrávání programu probíhá kód, který přijímá data ze sériové linky a ukládá je do dvourozměrného pole po řádcích. Kontroluje se jen konec programu, a pokud přijde (příkaz M2), načítání se zastaví.

Po kódu pro nahrávání programu následuje část pro realizaci frézování, respektive posuvu os dle programu. Této části kódu je věnována následující kapitoly 5.5.5. a 5.2.6.

Následný kód pro manuální posuv je jednoduchý, pokud je stisknuto určité tlačítko pro posuv, generují se příslušné signály STEP a DIR. Hlídá se jen stav koncových spínačů.

Část programu věnované referování také není složitá. Jak bylo popsáno v minulé kapitole, jen se opět generuje signál a hlídají koncové spínače. Po najetí na ně se program postará o sjetí z nich /samozřejmě na opačnou stranu) a nastaví flag informující ostatní stavy o provedené kalibraci.

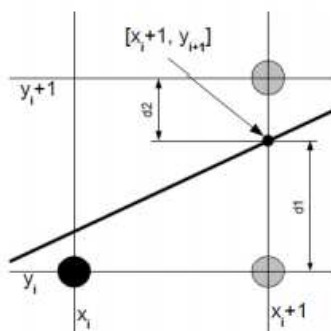
Kód pro stav „Připraveno“ je pouhou cyklickou kontrolou, zda není požadavek na přejítí do jiného stavu – režimu.

Následný kód je pro režim chyby. Ten vypne všechny výstupy a vypíše chybové hlášení na display.

5.2.5 Lineární interpolace

Pro lineární interpolaci jsem využil Bresenhamův algoritmus pro vykreslení úsečky. Je to nejpoužívanější a nejefektivnější algoritmus pro rasterizaci úsečky. Používá celočíselnou aritmetiku – sčítání, posuny, porovnání. Je velmi efektivní – ideální pro HW implementaci. [11].

Vykresluje se po pixelu od $[x_1, y_1]$ do $[x_2, y_2]$. V ose X s přírůstkem jedna. A o posunu v ose Y rozhoduje znaménko predátoru. [11]



$$d_2 > d_1 \Rightarrow P < 0 \Rightarrow y_{i+1} = y_i$$

$$d_2 \leq d_1 \Rightarrow P \geq 0 \Rightarrow y_{i+1} = y_i + 1$$

Obrázek 45: grafické naznačení funkce Bresenhamova algoritmu [11]

5.2.6 Režim frézování

V tomto stavu se provádí program uložený v paměti procesoru. Podle příkazu „G1“ a prvního názvu osy pro kterou se má provádět se určí následný mód (viz výše). Ty jsou dva, pro interpolaci os X a Y, a druhý pro interpolaci os Z. Než se do těchto módů přejde, jsou prvně převedeny hodnoty pro příslušné posuvy a vypočteny počty potřebných kroků – impulsů (STEP). Pokud je nový řádek programu „M2“ program je dokončen.

V módu 1 se provádí interpolace dvou os naráz, proto je zde využit Bresenhamův algoritmus, který nevyžaduje výpočty s desetinnými místy. Ten je popsán v předchozí kapitole. Pro potřeby řídicí jednotky je ale nutné algoritmus mírně upravit. Algoritmus je navržen jen pro kladné hodnoty a podmínku že rozdíl vzdáleností x je větší než rozdíl y . Aby bylo možné interpolovat je nutné před začátkem provést pomocné výpočty. Bohužel Bresenhamův algoritmus využívaný pro vykreslování v grafických aplikacích je sice potřebně upraven, ale je zjednodušen tak že v případě rozdílu x a y vykresluje úsečku obráceně. Což není pro frézování použitelné. Bylo nutné provést potřebné úpravy, viz část kódu v Příloze 5.

V módu 2 je prováděna jen interpolace jedné osy. Nejsou tedy třeba žádné složité výpočty. Jen se vygeneruje a spočítá potřebný počet kroků k ujetí požadované vzdálenosti.

5.3 Program pro PC

Ovládací program pro PC je jen velmi jednoduchý. Díky komponentám Visual Studio a jazyku C# stačilo napsat jen obsluhující kód pro několik tlačítek s využitím komponent OpenFileDialog a SerialPort.

V akci tlačítka „Připojit“ se provede pokus o připojení na nastavený COM port, pokud se podaří, aktivují se všechna tlačítka, která mohou být použita. Pokud připojování selže, vypíše se selhání do statusu. Při použití tlačítka „Odpojit“ se postupuje opačně.

Tlačítkem „Reset“ je po sériové lince odeslán znak „S“ tím se řídicí jednotka inicializuje do stavu „Připraveno“.

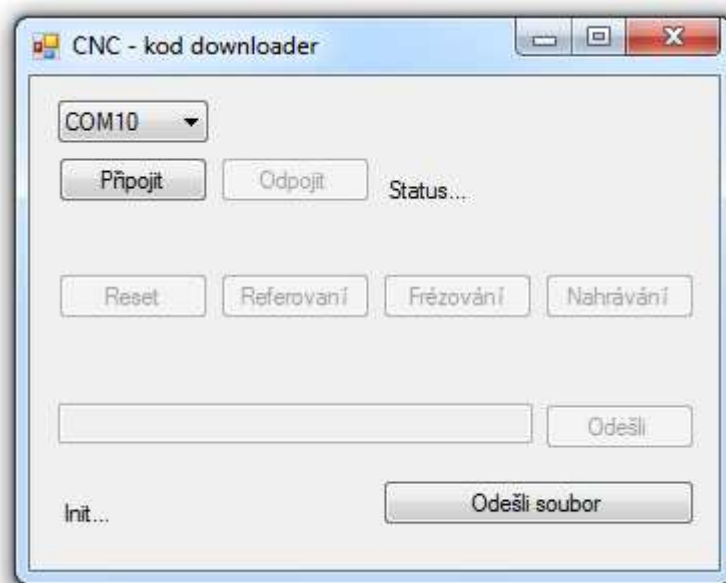
Tlačítko „Referování“ odešle znak „K“ a tím spustí referování.

Tlačítko „Nahrávání“ odešle znak „W“ a aktivuje textový řádek a tlačítko „Odešli“. Pak je možné posílat vlastní textové řetězce programu.

Tlačítko „Frézování“ je aktivní, až když je odeslán nějaký program. Odešle znak „A“ a tím spustí frézování.

Tlačítko „Odešli soubor“ otevře okno, kde vybereme cestu a soubor k nahrání. Soubor se po té odešle.

Ve spodním statusu je zobrazován status řídicí jednotky.



Obrázek 46: Okno obslužného programu v PC

6 TESTOVÁNÍ

6.1 Frézka

Při testování pevnosti se konstrukce stroje osvědčila. Při zkouškách posuvů os ale docházelo u osy X a Z k zadrhávání šroubu. Po bližším zkoumání se ukázalo, že trapézové šrouby jsou mírně prohnuté. Na celé délce osy to ale způsobovalo velké tlaky na matici – docházelo k nechtěnému radiálnímu tření. To bylo ale neúnosné, proto jsem musel posuvy těchto dvou os demontovat a šrouby nahradím novými kvalitnějšími. U osy Y (Obrázek 12) k tomu naštěstí nedošlo. K žádným jiným problémům nedošlo.

Jako první frézování jsem zkusil gravírování nápisu „VUT Fekt“. K vygenerování programu jsem použil CAM procesor CamBam. Je volně dostupný na [8]. U tohoto CAM procesoru je možné nastavit generování ISO kódu pouze za pomoci lineární interpolace (příkaz G1). S výsledného textového souboru jsem pouze odstranil nepodporované příkazy (nastavování otáček vřetene, příkazy pro odsávání apod.). Jako nástroj jsem použil drážkovací frézku 3mm. Výsledný nápis je na obrázku 47.



Obrázek 47: Zkušební nápis „VUT Fekt“

Při testech jsem prováděl i měření maximálních posuvů a zrychlení jednotlivých os. Osa X a Y mají stejné parametry což je, vzhledem ke stejným motorům a podobnému zatížení, samozřejmé. Rychlost posuvu os v zátěži je 25mm/s. Rychloposuv (bez zátěže) je pak 35mm/s. Osa Z má vzhledem k silnějšímu motoru a kompenzaci váhy vozíku protizávažím rychlost větší, ale zrovna u této osy nebude využita. Rychlost posuvu v zátěži je 40mm/s. Rychloposuv (bez zátěže) je pak 60mm/s. Zrychlení osy X a Y je při rychloposuvu až 50mm/s^2 , v zátěži 35mm/s^2 a u osy Z (motor s krouticím momentem 1,9Nm) je při rychloposuvu až 60mm/s^2 , v zátěži 40mm/s^2 . Hodnoty zrychlení u osy Z nejsou o tolik vyšší, jak by se dalo čekat, vzhledem k síle motoru. To je způsobeno velikou váhou samotného vozíku osy a protizávaží, které musím motor udělit pohyb. Dynamika osy Z je tedy řádově nižší. S použitím protizávaží tedy vyplývá, že zvyšuje statickou rychlost, ale výrazně snižuje dynamiku, tedy dle předpokladu. Jeho použití bylo ale nevyhnutelné.

Měření parametrů jsem prováděl za pomoci počítače, který generoval přesné impulzy motorům, a pomocí digitálního měřidla jsem měřil skutečnou ujetou vzdálenost. Ve chvíli kdy neodpovídala programovému posuvu, bylo jasné, že motor už ztrácel kroky. Do statistiky byla tedy zapsána předchozí. Měření jsem prováděl po pěti mm/s a u zrychlení po pěti mm/s².

6.2 Elektronika

Při zkouškách driveru se ukázal problém, v provedení návrhu plošného spoje. V první verzi jsem nerespektoval oddělené vedení logické a výkonové země a spojení pouze v jednom bodě. To způsobovalo společně s nedostatečnou filtrací napětí zdroje, takové rušení, že docházelo k nechtěným krokům motoru.

V druhé verzi (prezentované výše) jsem návrh upravil. Po doplnění filtračních kondenzátorů ke zdroji, už k nechtěným krokům nedochází.

Při testech jsem dosáhl s použitými motory na prázdno 120t/s, což odpovídá frekvenci 2400Hz. To vzhledem k použitým motorům a stoupání šroubu odpovídá posuvu 48mm/s. V zátěži je ale hodnota nižší, maximální posuv bez ztráty kroku je 25mm/s. Na běžné gravírování je to plně vyhovující. Pro zamýšlený materiál na obrábění – hliník a dřevo je to dostatečné. Pro plasty a jiné materiály s nízkou teplotou tavení by to bylo nedostatečné. Plasty je nutné obrábět velmi rychle, aby se co největší část tepla odvedla v ubíraném materiálu.

6.3 Řídicí jednotka

Ověřením funkce řídicí jednotky byl výše zmíněný nápis. Díky velmi výkonnému AMR procesoru, by jednotka zvládla řídit posuvy i o výstupní frekvenci 100kHz, což jsem ověřil pomocí osciloskopu. Takové drivery pro ovládání motorů, které by zvládli ovládat motor takto rychle, jsou už ale vybaveny sběrnici než mnou použitou metodou signálů STEP a DIR.

Na plošném spoji jsem z důvodu ladění programu udělal pomocí drátových propojek několik změn. Protože JTAG je vyveden i na některé mnou použité piny. Ve výše zmíněných schématech jsou již tyto změny zahrnuty. Zpočátku vývoj programu také zbytečně brzdila vadná série optočlenů. Musel jsem je vyměnit za nové, stejného typu. Při testech také bohužel došlo k nechtěnému zkratu 3,3 voltového okruhu a byl zničen stabilizátor na modulu s mikrokontrolerem. Po výměně je vše v pořádku.

7 ZÁVĚR

V rámci diplomové práce se mi podařilo dokončit konstrukci frézky. Hotový stroj má aktivní prostor 700x700x300mm, což je teoretická maximální možná velikost obrobku. Celková velikost je pak 1100x1300x1400mm. Je vyroben z lepených laminovaných dřevotřískových desek a překližky. Na posuvy a namáhané části je samozřejmě použit kov. Pevnostně je stroj dimenzován na tvrdé dřevo a hliník. Frézka disponuje vřetenem o výkonu 600W a s rozmezím otáček 12000 – 27000 ot/min. Rychlost posuvu os X a Y v zátěži je 25mm/s. Rychloposuv (bez zátěže) je pak 35mm/s. U osy Z jsou hodnoty posuvu 40mm/s a 60mm/s. Zrychlení osy X a Y (motory Olivetti s krouticím momentem 0,84Nm) je při rychloposuvu až 50mm/s^2 , v zátěži 35mm/s^2 a u osy Z (motor s krouticím momentem 1,9Nm) je při rychloposuvu až 75mm/s^2 , v zátěži 40mm/s^2 . Na pracovní stůl je možné dát obrobek o hmotnosti až 160kg – podle dimenzování ložisek. Teoretické rozlišení frézky je 0,0025mm. Je dáno stoupáním pohybového šroubu a počtem kroků motoru na jednu otáčku. Hodnota rozlišení v praxi, změřená pomocí úchylkoměru, je 0,06mm. Rozdíl od vypočtené hodnoty je dán člověkem nepostřehnutelnými nepřesnostmi a vůlemi ve vedení a styku pohybového šroubu s maticí.

V druhé části práce dále popisuji driver pro řízení krokového motoru. Pro použité motory naprosto vyhovuje řízení za pomoci obvodu Toshiba TB6560AHQ. Tento typ obvodu umožňuje nastavování proudu, mikrokrokování modifikovanou sinusovkou s nastavením počtů kroků na otáčku (200, 400, 800 a 1600) a propracované režimy rychlosti nabuzení/odbuzení do/od nominálního proudu. Driver má opticky oddělené vstupy. Obvod je pouze nutné důkladně chladit. S tímto driverem jsem docílil 12 ot/s krokového motoru, to v ideálním případě zajišťuje posuv os 48mm/s. Součástí návrhu je potřebný elektrický zdroj.

Třetí částí je návrh, konstrukce a program pro řídicí jednotku – HW interpolátor. Tato jednotka umožňuje ovládat tři drivery (osy frézky), zpětnou vazbou jsou koncové spínače. Jádrem interpolátoru je 32 bitový mikrokontroler ARM od firmy ST. Pracuje na frekvenci 24MHz. Zajišťuje funkce jednotky, kterými jsou: referování, manuální posuvy, detekce chyby (programu, či najetí na koncový spínač). Hlavní funkcí ale je příjem dat - programu z PC, jde o program v ISO kódu. Program se následně vypočítává a realizuje přímo v procesoru a ten nastavuje výstupní signály STEP a DIR pro všechny tři osy. Jedná se o pouze o tzv. 2,5 osý interpolátor. Umožňuje gravírování/frézování v osách X a Y, u osy Z lze nastavovat její pozici, ale nelze vykonávat interpolaci v kombinaci s jinou osou. To ale na většinu požadovaných výrobků stačí. Jednotka je vybavena dostatečně velkým displejem, tudíž umožňuje zobrazovat právě vykonávaný kód. Celkově je řešena modulárně v sendvičové konstrukci. Součástí je i ovládací program pro PC.

Literatura

- [1] CNCShop.cz [online]. 2010 [cit. 2011-01-15]. CNCShop. Dostupné z WWW: <<http://www.cncshop.cz/w-vodici-tyce>>.
- [2] Henlich.cz [online]. 2011 [cit. 2011-01-15]. Hennlich. Dostupné z WWW: <<http://lin-tech.hennlich.cz/produkty/linearni-vedeni-a-kulickove-srouby-linearni-vedeni-valive-konvenncni-vedeni-635/typ-sr.html>>.
- [3] Atmel [online]. 2009 [cit. 2009-01-05]. Dostupný z WWW: <<http://www.atmel.com/>>.
- [4] ŘEZÁČ, Kamil. Krokové motory. Robotika.cz [online]. 2002-10-28, č.1., [cit. 2011-01-16]. Dostupný z WWW: <<http://robotika.cz/articles/steppers/cs>>.
- [5] TB65660AHQ datasheet [online]., 13-11-2007 [cit. 2011-01-16]. Dostupné z WWW: <http://www.gaotaobao.com/template/flash_image/TB6560AHQ_AFG_E_2003_20080215.pdf>.
- [6] GME [online]. 2011 [cit. 2011-01-16]. P-B172C. Dostupné z WWW: <<http://www.gme.cz/cz/p-b172c-p630-156.html>>.
- [7] STM32 Value Line. STM32 Discovery kit [online]. 2010, č. 1, [cit. 2011-01-16]. Dostupný z WWW: <<http://mcu.cz/page.php?23>>.
- [8] CamBam [online]. 2011 [cit. 2011-05-18]. Dostupné z WWW: <<http://www.cambam.info/>>.
- [9] Alldatasheet [online]. 2009 [cit. 2011-05-18]. FT23RL datasheet. Dostupné z WWW: <<http://pdf1.alldatasheet.com/datasheet-pdf/view/144591/FTDI/FT232RL.html>>.
- [10] STM32-VL DISCOVERY - popis. *STM32-VL DISCOVERY* [online]. 13.9.2010, 1, [cit. 2011-05-19]. Dostupný z WWW: <<http://mcu.cz/comment.php?comment.news.2058>>.
- [11] PŘIBIL, Bronislav. Generování základních objektů v rastru. *IZG 2010/11, cvičení* [online]. 2010, 2, [cit. 2011-05-19]. Dostupný z WWW: <<http://www.fit.vutbr.cz/>>.

Seznam příloh

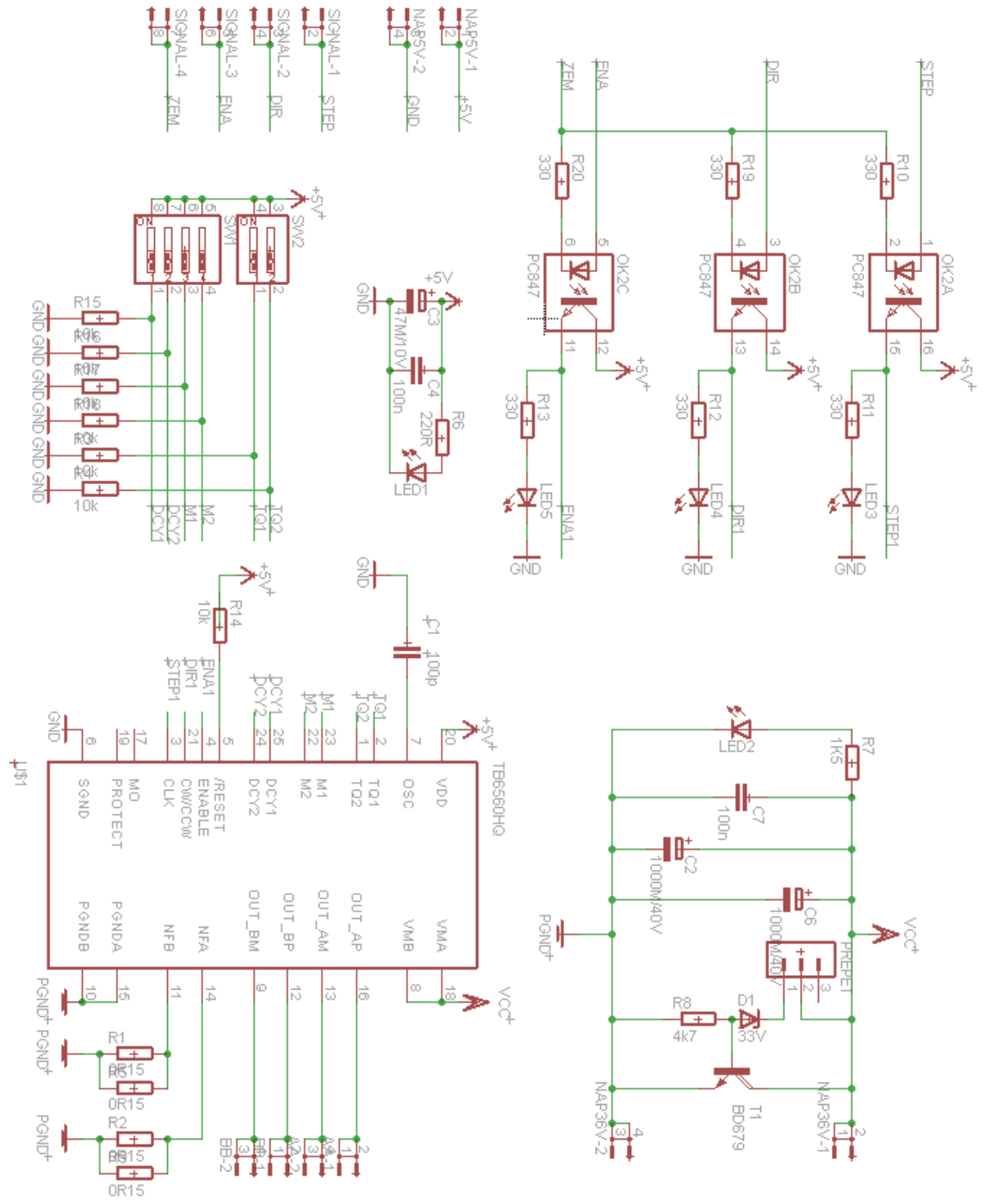
Příloha 1 – Schéma, plošný spoj a osazovací plán driveru

Příloha 2 – Schéma, plošný spoj a osazovací plán komunikačního modulu

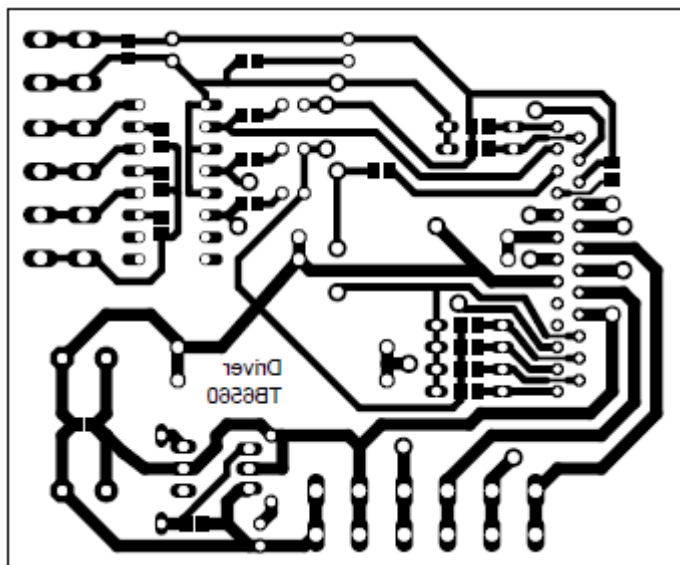
Příloha 3 – Schéma, plošný spoj a osazovací plán horního panelu

Příloha 4 – Schéma, plošný spoj a osazovací plán spodního panelu

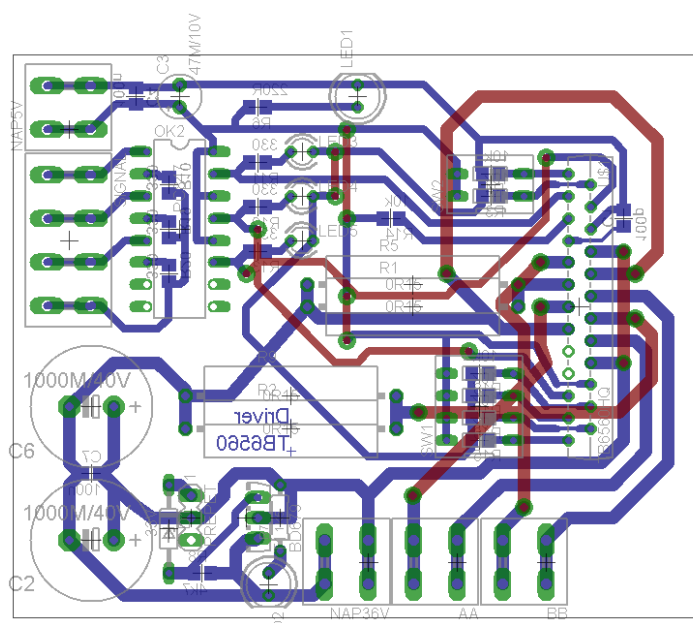
Příloha 5 – Část kódu pro lineární interpolaci os X a Y



Obrázek 48: Schéma driveru

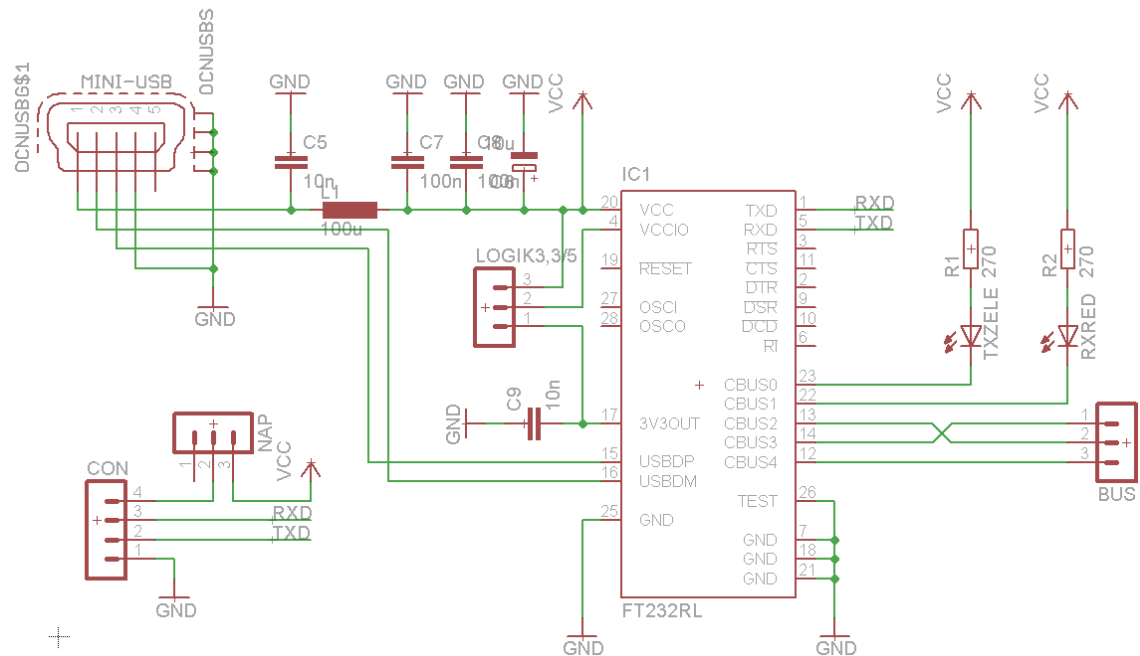


Obrázek 49: Předloha DPS driveru

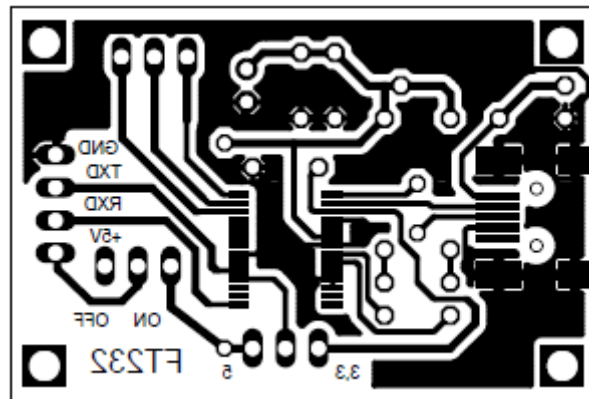


Obrázek 50: Osazovací plán driveru

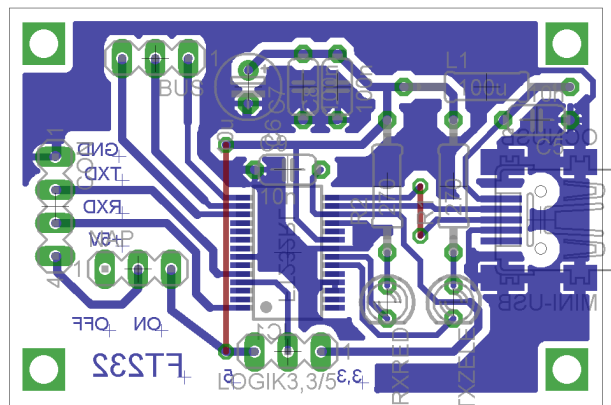
Příloha 2 – Schéma, plošný spoj a osazovací plán komunikačního modulu



Obrázek 51: Schéma komunikačního modulu

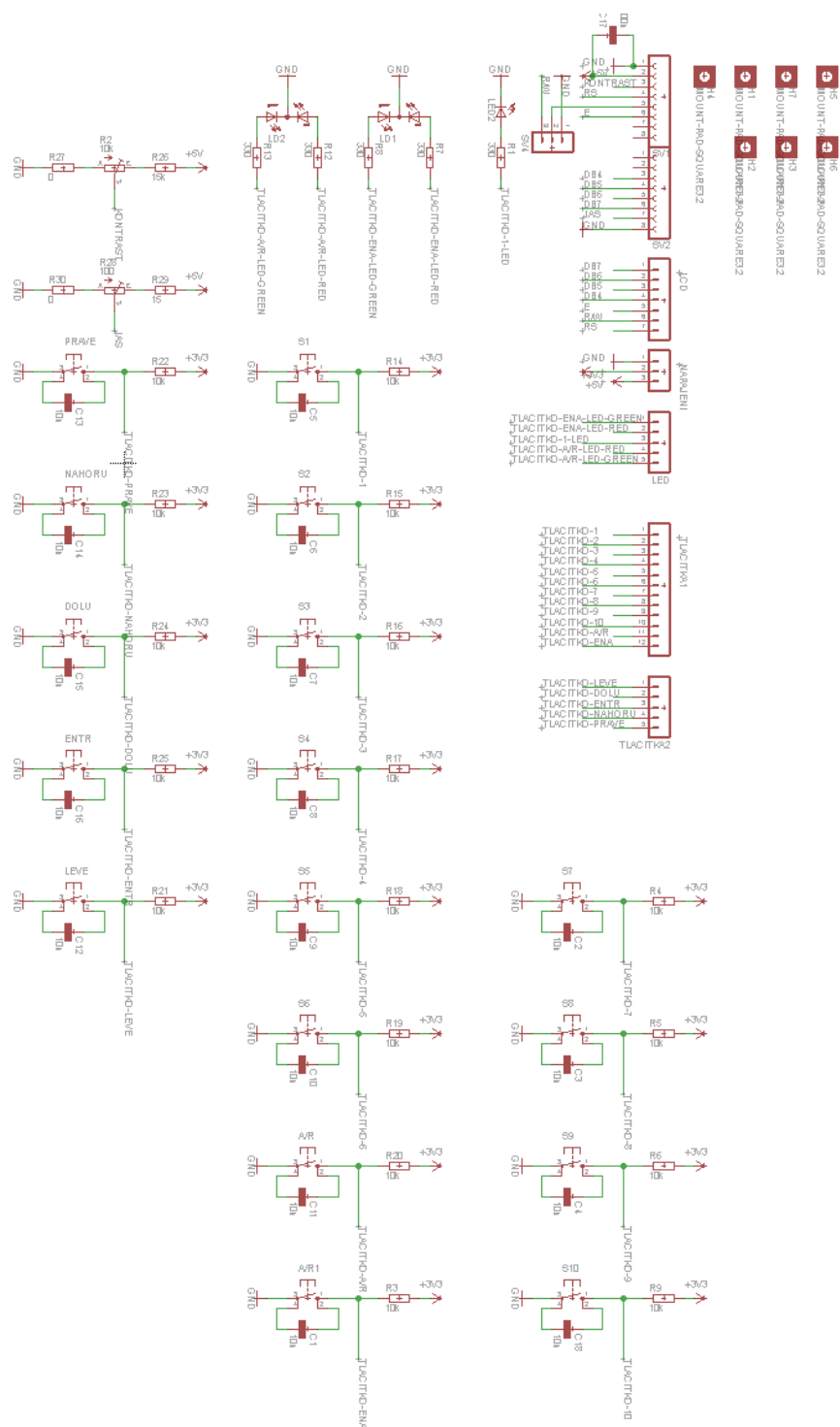


Obrázek 52: DPS komunikačního modulu

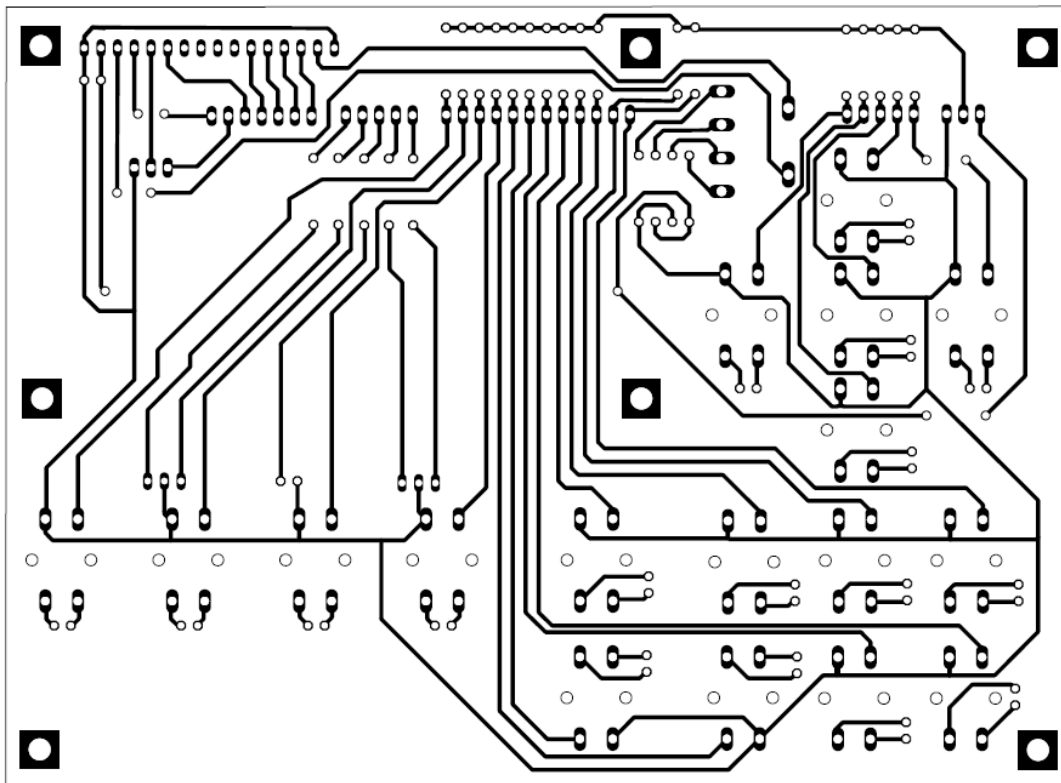


Obrázek 53: Osazovací plán komunikačního modulu

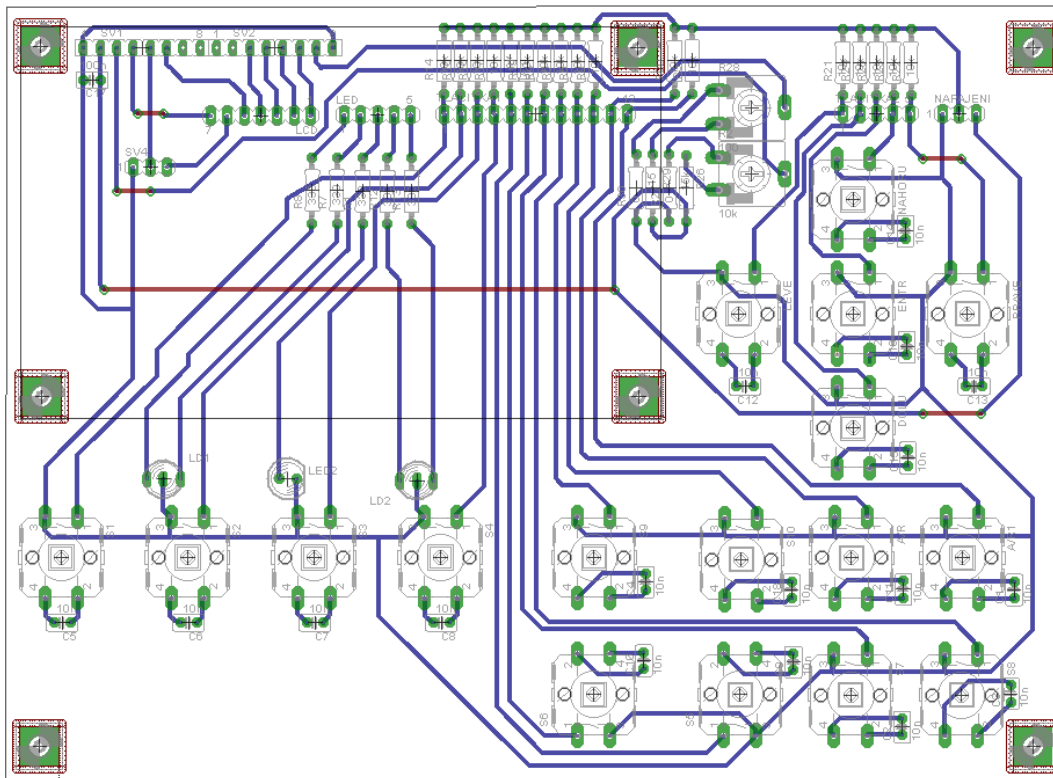
Příloha 3 – Schéma, plošný spoj a osazovací plán horního panelu



Obrázek 54: Schéma horního panelu

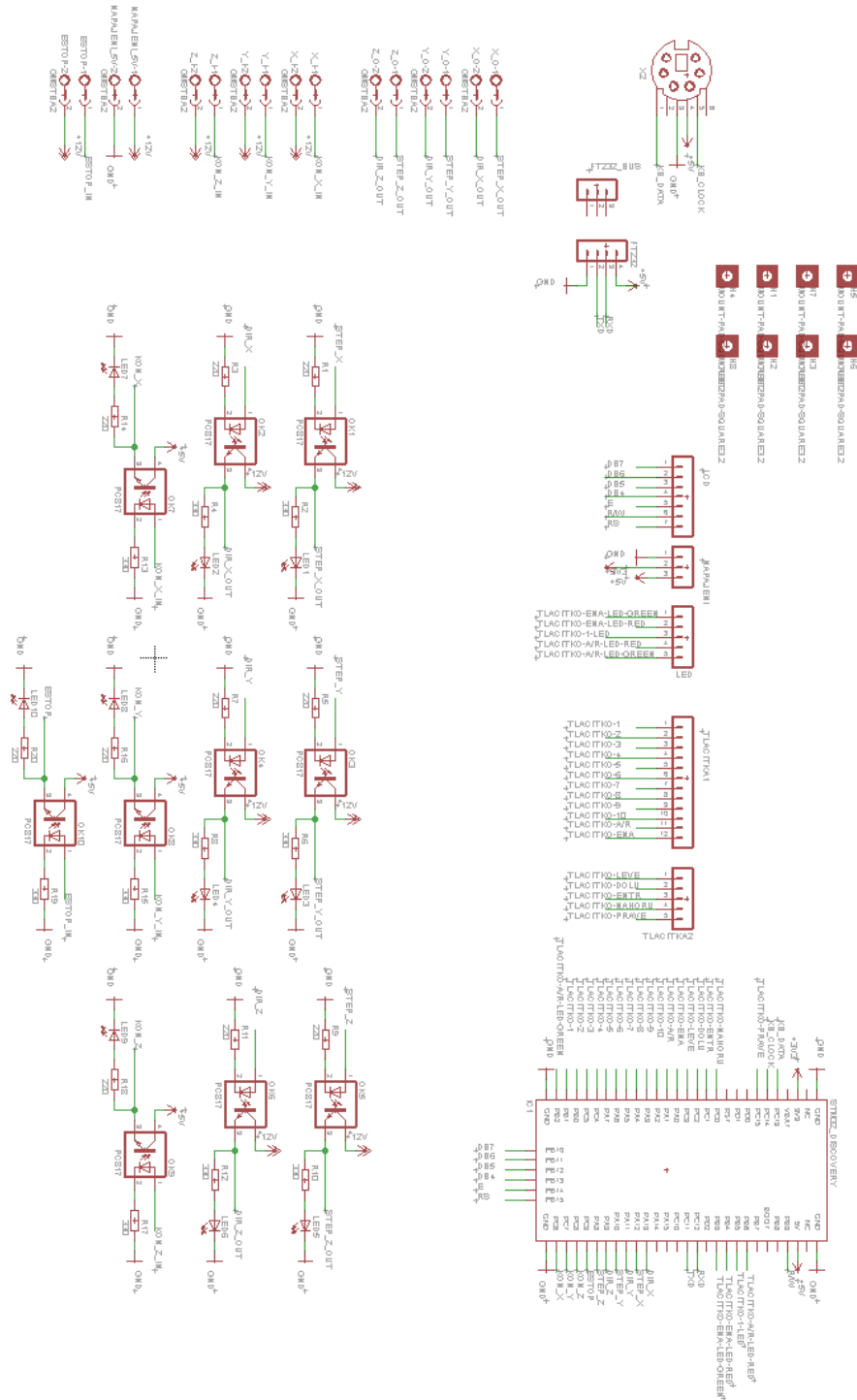


Obrázek 55: DPS horního panelu

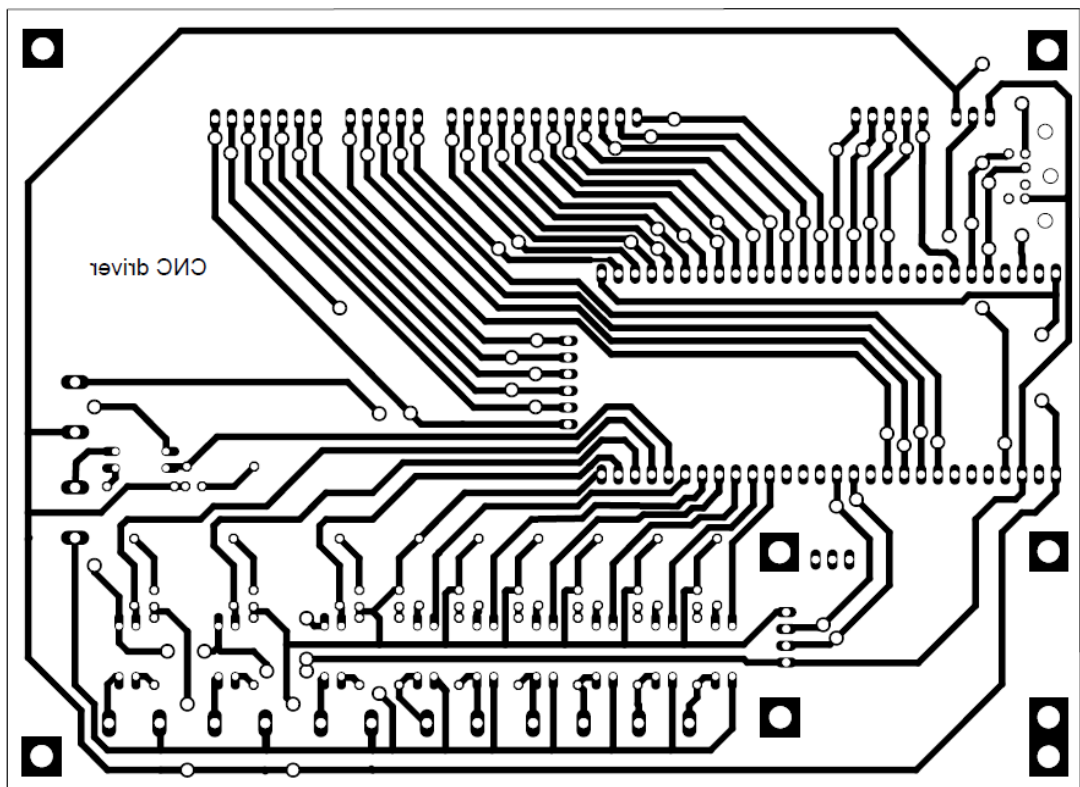


Obrázek 56: Osazovací plán horního panelu

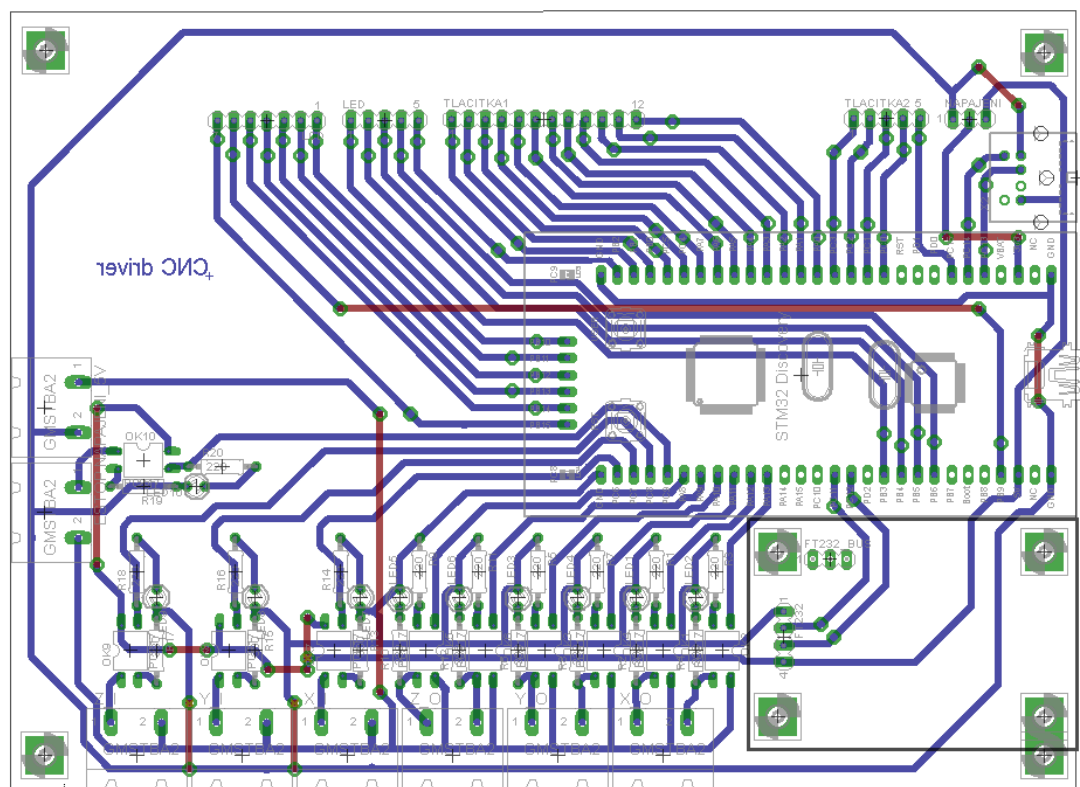
Příloha 4 – Schéma, plošný spoj a osazovací plán spodního panelu



Obrázek 57: Schéma spodního panelu



Obrázek 58: DPS Spodního panelu



Obrázek 59: Osazovací plán spodního panelu

Příloha 5 – Část kódu pro lineární interpolaci os X a Y

```
if (pripravaG1XY == 0){          //provedeni potrebných výpočtu před začátkem interpolace
    if (abs(YI_AKT - YI_NOVA) > abs(XI_AKT - XI_NOVA)) vysoky = 1; //zjištění orientace //usecky
        if (vysoky){            // pokud je orientována na vysoko provede se přepočítání
            pomP = XI_AKT;
            XI_AKT = YI_AKT;
            YI_AKT = pomP;

            pomP = XI_NOVA;
            XI_NOVA = YI_NOVA;
            YI_NOVA = pomP;
        }
        dx = ABSS(XI_NOVA - XI_AKT);    //výpočet rozdílu
        dy = ABSS(YI_NOVA - YI_AKT);
        P = 2*dy - dx;                  //prediktor
        P1 = 2*dy;
        P2 = P1 - 2*dx;
        pripravaG1XY = 1;

        XI_AKTp = XI_AKT;               //ulžení aktuální pozice pro porovnání
        YI_AKTp = YI_AKT;
    }
    if (step_x_up == 0 && step_x_down == 0 && step_y_up == 0 && step_y_down == 0){
        if (vysoky){
            if (XI_AKTp > XI_NOVA){
                step_y_down = 2;
                YI_AKT--;
                if (step_y_flag && (YI_AKTp > YI_NOVA)) step_x_down = 2; XI_AKT--;
                if (step_y_flag && (YI_AKTp < YI_NOVA)) step_x_up = 2; XI_AKT++;
            }
            else{
                step_y_up = 2;
                YI_AKT++;
                if (step_y_flag && (YI_AKTp > YI_NOVA)) step_x_down = 2; XI_AKT--;
                if (step_y_flag && (YI_AKTp < YI_NOVA)) step_x_up = 2; XI_AKT++;
            }
        }
        else{
            if (XI_AKTp > XI_NOVA){
                step_x_down = 2;
                XI_AKT--;
                if (step_y_flag && (YI_AKTp > YI_NOVA)) step_y_down = 2; YI_AKT--;
                if (step_y_flag && (YI_AKTp < YI_NOVA)) step_y_up = 2; YI_AKT++;
            }
            else{
                step_x_up = 2;
            }
        }
    }
}
```

```

        XI_AKT++;
        if (step_y_flag && (YI_AKTp > YI_NOVA)) step_y_down = 2; YI_AKT--;
        if (step_y_flag && (YI_AKTp < YI_NOVA)) step_y_up = 2; YI_AKT++;
    }
}
step_y_flag = 0;
if (P >= 0){                                     //vypocty s prediktorem
    P = P + P2;
    step_y_flag = 1;
}
else P = P + P1;

if (XI_AKT == XI_NOVA || YI_AKT == YI_NOVA){      //test ukončení
    MOD_PR = 0;
    pripravaG1XY = 0;
    vysoky = 0;
    XI_AKT = XI_NOVA;
    YI_AKT = YI_NOVA;
}

```