



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

**ŘÍDÍCÍ MODUL DVOUOSÉHO KARTÉZSKÉHO
MANIPULÁTORU PRO PLC S7-1200**

CONTROL MODUL OF THE BIAxIAL CARTESIAN MANUPULATOR FOR PLC
S7-1200

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Miroslav Hofman

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Michal Pavlík, Ph.D.

BRNO 2016

ABSTRAKT

Práce pojednává o návrhu modulu pro řízení dvouosého kartézského manipulátoru. Řídící modul přijímá signály STEP a DIR z PLC, které mohou mít frekvenci až 100 kHz. Tyto signály modul zpracovává a posílá na vstupy driverů, kterými jsou řízeny motory dvouosého manipulátoru. Obě osy manipulátoru jsou řízené současně, což PLC bez tohoto modulu nedokáže. Nastavené parametry se zobrazují na LCD displeji. Řídící modul je odolný proti rušení. Všechny signály jsou odděleny rychlými optočleny, které zajišťují minimální zkreslení signálů. Data jsou zpracovávána mikroprocesorem ATXMEGA. Program mikroprocesoru je napsán v jazyce C. Modul je možné přepnout do NPN nebo PNP logiky a lze zvolit výstupní napětí 5 nebo 24 V podle typu a nastavení použitého driveru pro řízení motorů.

KLÍČOVÁ SLOVA

Řídící modul, kartézský manipulátor, PLC, driver, LCD, ATXMEGA

ABSTRACT

This thesis deals with the design of module for control of biaxial cartesian manipulator. Control module receives signals STEP and DIR from PLC - these signals can have frequency up to 100 kHz. They are processed by the module and send to inputs of drivers, which control motors of biaxial manipulator. Both axis of manipulator are driven simultaneously. PLC is not able to manage this process without this module. Adjusted parameters are displayed on LCD display. Control module is disturbance-resistant. All signals are isolated by fast optocouplers, which guarantee minimum distortion of signals. Data are processed by microprocessor ATXMEGA. Program of microprocessor is written in the C language. It is possible to switch module to NPN or PNP logic and it is also possible to select output voltage 5 or 24 V according to the type and adjustment of used driver for motor control.

KEYWORDS

Control module, Cartesian manipulator, PLC, driver, LCD, ATXMEGA

HOFMAN, M. *Řídící modul dvouosého kartézského manipulátoru pro PLC S7-1200*.
Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních
technologií. Ústav mikroelektroniky, 2015. 55 s., 15 s. příloh. Diplomová práce.
Vedoucí práce: Ing. Michal Pavlík, Ph.D.

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma „Řídící modul dvouosého kartézského manipulátoru pro PLC S7-1200“ jsem vypracoval samostatně pod vedením vedoucího sdíplomové 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 tohoto projektu 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 § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 24. května 2016

.....
podpis autora

Poděkování

Děkuji vedoucímu semestrální práce Ing. Michalu Pavlíkovi, Ph.D. a vedoucímu výroby ve firmě Cutter Systems s.r.o. Ing. Ondřeji Tomáškoví za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování projektu.

V Brně dne 24. května 2016

.....
podpis autora

Obsah

Obsah	v
Seznam obrázků	vi
Seznam tabulek	viii
Úvod	1
1 Programovatelný automat Simatic S7-1200	2
1.1 Totally integrated Automation portal	3
1.2 Programovací jazyk structured control language.....	3
1.3 Motion control funkce	4
2 Manipulátory	7
2.1 Kartézský manipulátor	8
2.2 Driver ES-D508	10
3 Návrh modulu pro řízení dvouosého kartézského manipulátoru	12
3.1 Řídící mikrokontrolér ATXMEGA 128A1.....	13
3.2 Zdroj napájení	20
3.3 Galvanické oddělení vstupních signálů	21
3.4 Galvanické oddělení výstupních signálů	23
3.5 Zobrazovací jednotka.....	24
4 Firmware	25
4.1 Verze firmwaru 1.0.....	26
4.2 Verze firmwaru 1.1	29
4.3 Verze firmwaru 2.0.....	34
4.4 Verze firmwaru 2.1	37
5 Testování	44
5.1 Nastavení parametrů pro testování	47
5.2 Kontrola přesnosti pohybu manipulátoru	48
5.3 Změřené závislosti aktuálních pozic na čase	49
6 Závěr	55
Seznam použité literatury	57
Seznam příloh	58

Seznam obrázků

Obr. 1: PLC Simatic S7-1200	2
Obr. 2: Režim „Pulse A and direction B“	4
Obr. 3: Režim „Count up A and count down B“	4
Obr. 4: Režim „A/B phase shifted“	5
Obr. 5: Režim „A/B phase shifted – fourfold“	5
Obr. 6: Nastavení parametrů motoru	6
Obr. 7: Nastavení pohybu	6
Obr. 8: Základní kinematické struktury manipulátorů [2].....	7
Obr. 9: Uvažovaný kartézský manipulátor	8
Obr. 10: Princip uvažovaného kartézského manipulátoru	9
Obr. 11: Driver Leadshine ES-D508 [4].....	10
Obr. 12: Prodlevy mezi jednotlivými signály driveru ES-D508 [4].....	11
Obr. 13: Blokové schéma zapojení řízení manipulátoru.....	12
Obr. 14: ATXMEGA 128 A1 v pouzdře TQFP100.....	13
Obr. 15: Blokové zapojení ATXMEGA 128 A1 [5]	14
Obr. 16: Označení pinů mikroprocesoru ATXMEGA 128A1 [5]	16
Obr. 17: Blokové schéma čítače/časovače [5]	17
Obr. 18: Generování Jednoduchý sklon PWM signálu [5].....	18
Obr. 19: Generování frekvenčního (FRQ) signálu [5].....	18
Obr. 20: Schéma zapojení zdroje	20
Obr. 21: Vnitřní zapojení HCPL0630 [7]	21
Obr. 22: Schéma zapojení vstupu	22
Obr. 23: Schéma zapojení výstupu	23
Obr. 24: Schéma zapojení přepínače a DC/DC převodníků	23
Obr. 25: Schéma zapojení displeje	24
Obr. 26: Nastavení signálů pro vynášení do grafu.....	44
Obr. 27: Nastavení vzorkování	44
Obr. 28: Nastavení začátku měření.....	45
Obr. 29: Obrazovka pro ovládání manipulátoru v automatickém režimu na ovládacím panelu.....	45
Obr. 30: Obrazovka pro ovládání manipulátoru v manuálním režimu na ovládacím panelu.....	46
Obr. 31: Testovací zapojení řídicího modulu v rozvaděči.....	46
Obr. 32: 3D- Sensor	48

Obr. 33: Testovací pracoviště s manipulátorem	48
Obr. 34: Závislosti polohy manipulátoru na čase pro frekvenci 10 kHz (1_10k – jednoduchý pohyb, 2_10k – sdružený pohyb)	49
Obr. 35: Závislosti polohy manipulátoru na čase pro frekvenci 30 kHz (1_30k – jednoduchý pohyb, 2_30k – sdružený pohyb)	50
Obr. 36: Závislosti polohy manipulátoru na čase pro frekvenci 50 kHz	51
Obr. 37: Závislosti polohy manipulátoru na čase pro frekvenci 70 kHz (1_70k – jednoduchý pohyb, 2_70k – sdružený pohyb)	52
Obr. 38: Závislosti polohy manipulátoru na čase pro frekvenci 10 kHz (1_70k – jednoduchý pohyb, 2_70k – sdružený pohyb)	53

Seznam tabulek

Tab. 1: Vybrané parametry PLC Simatic S7-1214C DC/DC/DC.....	2
Tab. 2: Prodlevy a parametry driveru ES-D508 [4].....	11
Tab. 3: Pohyby manipulátoru.....	42
Tab. 4: Tabulka korekcí kroků mezi jednoduchými pohyby	42
Tab. 5: Tabulka korekcí kroků mezi sdruženými pohyby	43
Tab. 6: Nastavené parametry motoru pro testování	47
Tab. 7: Nastavené rychlosti pro testování.....	47

Úvod

Cílem diplomové práce bylo seznámit se s možnostmi konstrukce modulu pro řízení dvouosého kartézského manipulátoru pomocí řídicího systému PLC S7-1200.

Po seznámení se s možnostmi konstrukce bylo toto elektronické zařízení navrženo a otestováno. V současné době se na trhu podobné zařízení nevyskytuje. Kartézský manipulátor uvažovaný v této práci má atypické řízení os. Samotné PLC není schopné řídit oba motory dvouosého kartézského manipulátoru současně a zároveň spolehlivě, proto musí udělat pohyb nejdříve po jedné ose a až po dokončení tohoto pohybu začne manipulovat s druhou osou. Modul pro řízení má za úkol řídit oba motory současně, tím dosáhneme koncové polohy manipulátoru za kratší čas. Automatizované linky, které jsou na tomto čase závislé, mohou tak produkovat mnohem více výrobků.

Modul je řízen mikroprocesorem. Vstupní signály STEP a DIR z PLC jsou obdélníkové s maximální frekvencí 100 kHz. Tyto signály jsou v PLC řízeny motion control funkcemi. Frekvence vstupních signálů určuje rychlost, jakou se manipulátor pohybuje. Modul bude umístěn v rozvodných skříních na halách, kde bývá mnoho rušivých signálů (svařování, rušení motorů...) proto bylo potřeba tento modul odrušit. Výstupní signál řídí drivery motorů dvouosého kartézského manipulátoru. Na LCD displeji se zobrazují aktuální parametry. LED diody dávají uživateli informaci o funkci. Přepínači je možné nastavit výstupní napětí 5 nebo 24 V a PNP nebo NPN logiku podle typu použitého driveru pro motor.

1 Programovatelný automat Simatic S7-1200

Simatic S7-1200 je označení pro řadu programovatelných logických automatů (Programmable Logic Controller – PLC), které se hodí především pro aplikace menšího rozsahu. Simatic PLC S7-1200 jsou kompaktní, modulární a poměrně levné PLC. Kompaktní jsou pro jejich malé rozměry a modulární pro jejich snadnou rozšiřitelnost pomocí množství přídatných modulů, což je dělá vhodné pro použití v širokém spektru aplikací.

PLC Simatic S7-1200 disponují především:

- integrovaným rozhraním Profinet pro programování, připojení HMI (Human– Machine Interface) či jiných zařízení;
- integrovanými technologickými funkcemi jako jsou čítače, časovače, PID řízení...
- integrovanými binárními a analogovými vstupy a výstupy.



Obr. 1: PLC Simatic S7-1200

Tab. 1: Vybrané parametry PLC Simatic S7-1214C DC/DC/DC

Parametr	Hodnota
Integrované binární I/O	14 vstupů 24 VDC / 10 výstupů 24 VDC
Integrované analogové I/O	2 vstupy 0-10 VDC
Integrovaná pracovní paměť	50 kB
Integrovaná pevná paměť	2 MB
Rychlost	0,1 μ s / instrukci
Napájecí napětí	24 DC

1.1 Totally integrated Automation portal

Totally Integrated Automation Portal (TIA Portal) je vývojové prostředí společnosti Siemens pro programování kompletních automatizačních systémů. Svým názvem dobře vystihuje hlavní myšlenku tohoto nového prostředí, to jest integrace všech dílčích součástí projektování automatizačních systémů do jednoho softwarového nástroje s jednotným ovládáním. V současné době (verze 13) je dokončena integrace systému pro vývoj aplikací z řídicí techniky a decentralizovaných periférií (současná Step 7 V13), dále pro vývoj vizualizací v blízkosti stroje (ovládací panely). S pomocí TIA portálu lze vytvořit program pro CPU v PLC, testovat a nastavovat motory motion kontrol funkcemi...

1.2 Programovací jazyk structured control language

K programování PLC v prostředí TIA Portal STEP 7 je možné použít různé programovací jazyky. PLC řady S7-1200 lze programovat v jazyce LAD (Ladder Logic), FBD (Function Block Diagram) a SCL (Structured Control Language). SCL je z těchto jazyků nejpokročilejším programovacím jazykem, a proto byl také zvolen pro programování všech bloků v této práci. Na rozdíl od grafických programovacích jazyků LAD a FBD je to jazyk textový, založený na jazyce PASCAL. Dle standardu IEC 61131-3, který definuje syntaxi a strukturu programovacích jazyků pro PLC, spadá do kategorie programovacího jazyka Structured Text (ST) a je tedy plně standardizován. Výhodou jazyka SCL je zejména podpora typu instrukcí známých z vyšších programovacích jazyků, jako jsou např. konstrukce IF..THEN..ELSE, FOR cykly, WHILE cykly a další. Samozřejmě podporuje i blokový koncept STEP 7 a lze jej tak libovolně kombinovat s bloky naprogramovanými v ostatních podporovaných jazycích, jinak řečeno, uvnitř bloku psaném v SCL lze volat bloky psané v ostatních jazycích a naopak. Nevýhodou programování PLC S7-1200 je absence tzv. debuggeru, který umožňuje zastavení běhu programu v požadovaném místě pomocí breakpointů. Pokračování programu lze potom ručně krokovat po jednotlivých řádcích a detailně tak sledovat tak průběh programu s hodnotami veškerých proměnných. To je obrovská výhoda u rozsáhlejších aplikací, neboť lze snadno odhalit logické chyby v kódu. Naproti tomu bez debuggeru je hledání chyb složité a hlavně velmi zdlouhavé. Je nutné vytvořit řadu pomocných proměnných a kódu navíc, který nám umožní sledovat chod programu a odhalit tak chyby. [1]

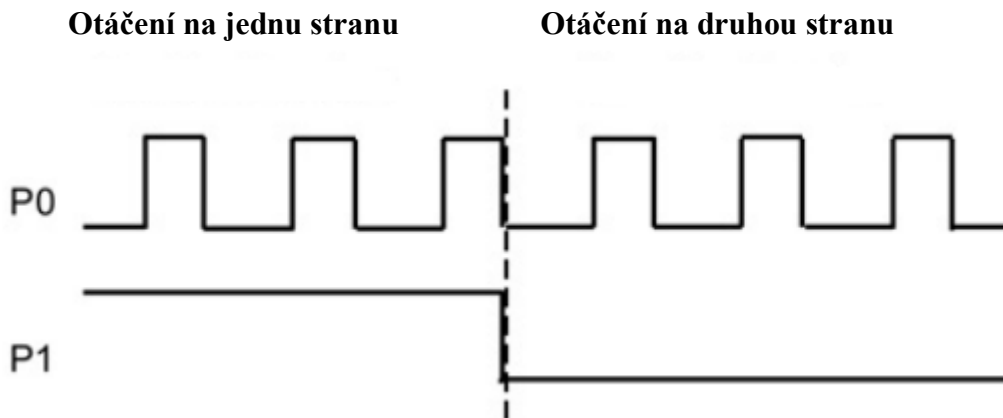
1.3 Motion control funkce

Tyto funkce slouží k nastavení řízení motorů a nalezneme je v TIA portálu. PLC Simatic S7-1200 umí jak PWM (Pulse Width Modulation), tak PTO (Pulse Train Output Modulation) . V této práci bude využívána pouze PTO. PTO znamená obdélníkový signál se střídou 50%. Změnou frekvence tohoto signálu se reguluje rychlost otáčení motoru. Použitý procesor v S7-1200 má 4 PTO výstupy. Z každého z nich můžeme přivést STEP signál na jeden motor.

Pro řízení motorů pomocí PTO existuje 5 následujících režimů:

1) „Pulse A and direction B“

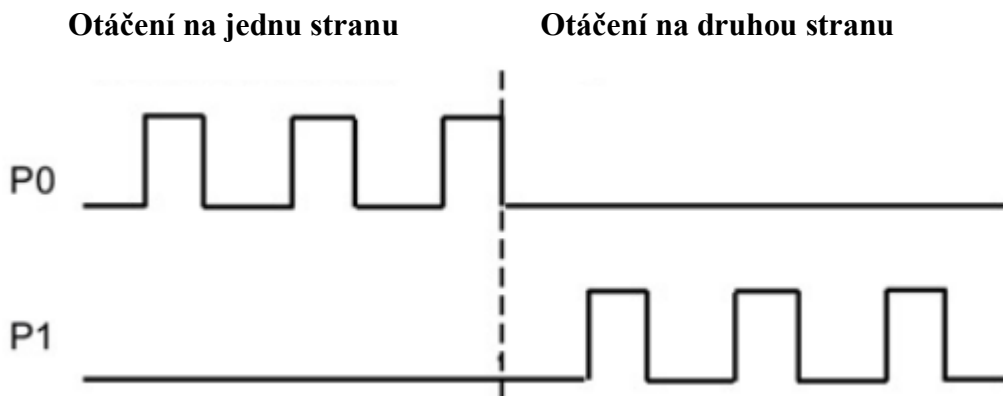
Pulzní signál STEP na P0 řídí rychlost motoru a signál DIR na P1 řídí směr otáčení motoru. Funkce je znázorněna na obrázku č. 2.



Obr. 2: Režim „Pulse A and direction B“

2) „Count up A and count down B“

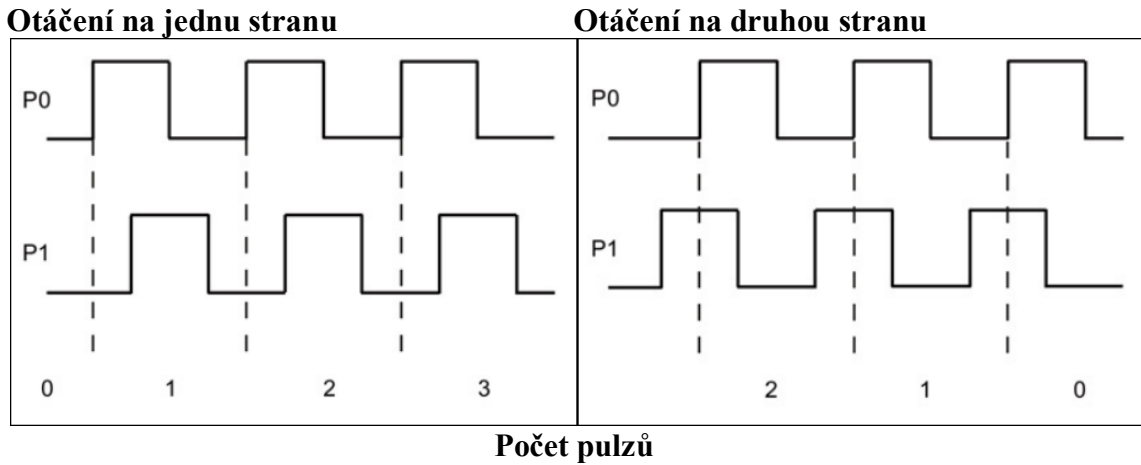
U tohoto režimu se pulzní signály střídají. Pokud jsou otáčky nastavené v jednom směru, jsou pulzy pouze na výstupu P0. Naopak pokud jsou otáčky nastavené v opačném směru, jsou pulzy pouze na výstupu P1.



Obr. 3: Režim „Count up A and count down B“

3) „A/B phase shifted“

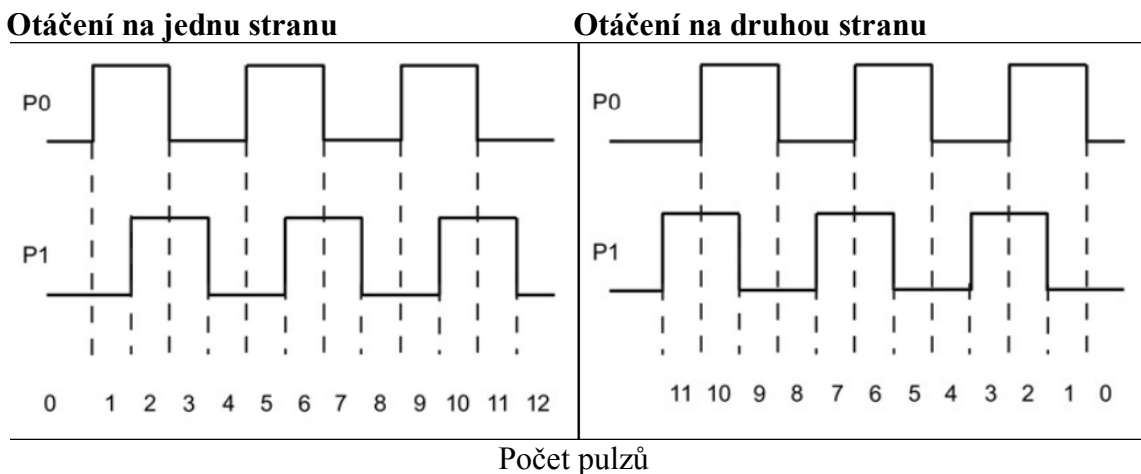
U tohoto režimu mají oba pulzní výstupy stejnou frekvenci, ale jsou vzájemně posunuté o 90° . Pokud jsou otáčky nastavené v jednom směru, signál P0 řídí signál P1. Naopak pokud jsou otáčky nastavené v opačném směru, signál P1 řídí signál P0. Jako jeden pulz se počítá jako perioda signálu.



Obr. 4: Režim „A/B phase shifted“

4) „A/B phase shifted - fourfold“

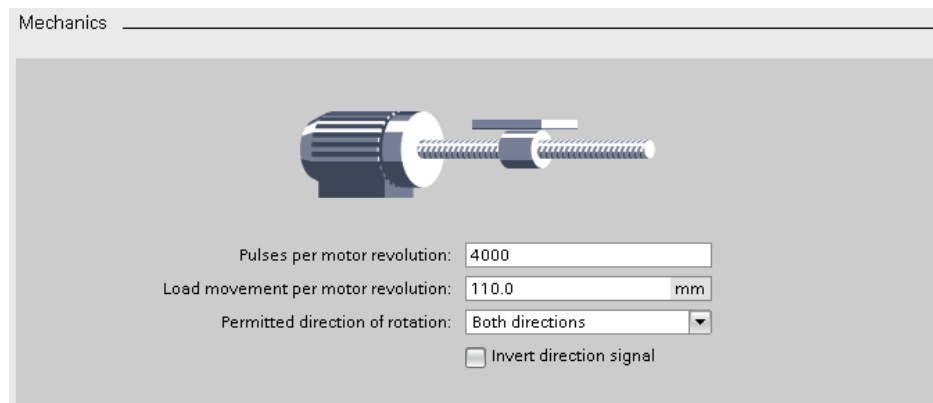
U tohoto režimu mají oba pulzní výstupy stejnou frekvenci, ale jsou vzájemně posunuté o 90° . Pokud jsou otáčky nastavené v jednom směru, signál P0 řídí signál P1. Naopak pokud jsou otáčky nastavené v opačném směru, signál P1 řídí signál P0. Jako jeden pulz se počítá jako $\frac{1}{4}$ perioda signálu. Oproti obyčejnému „A/B phase shifted“ je v tomto režimu možné nastavit počet přechodů(max.4).



Obr. 5: Režim „A/B phase shifted – fourfold“

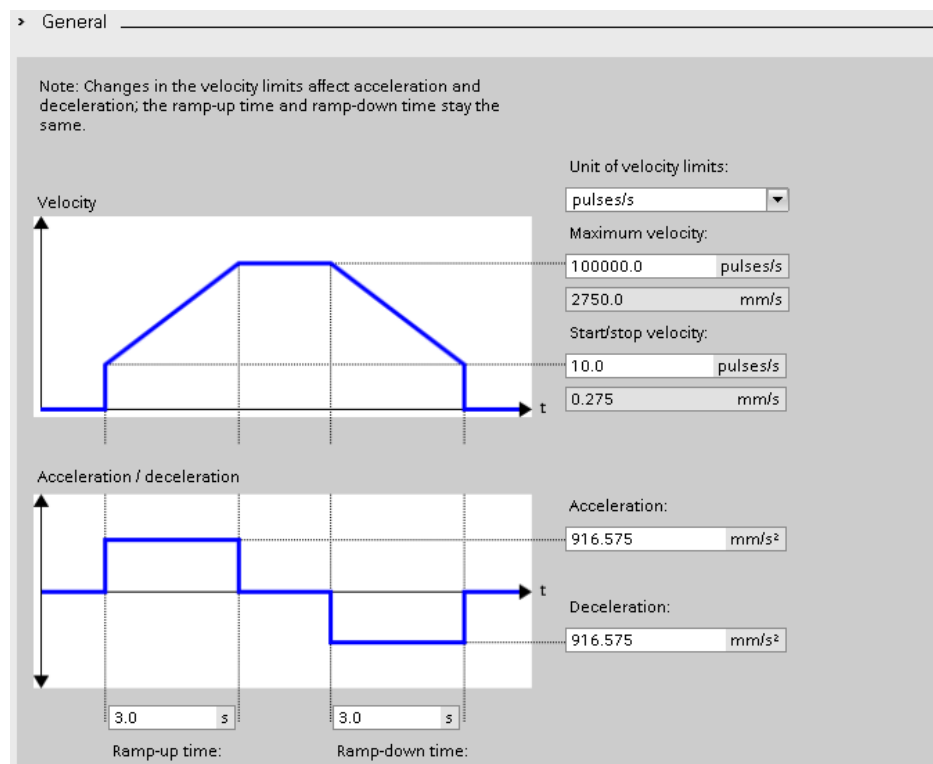
Nastavení parametrů motoru

U krokových motorů se v TIA portálu nastavuje řada parametrů, mezi které patří i počet pulzů na otáčku motoru (Pulses per motor revolution). Hodnota tohoto parametru se pohybuje od stovek po tisíce podle typu motoru. Dalším parametrem je posuv na otáčku (Load movement per motor revolution). Hodnoty tohoto parametru se pohybují od jednotek po stovky mm. Povolené řízení otáček (Permitted direction of rotation) udává v kolika směrech bude motor řízen. U dvouosých manipulátorů je řízení obousměrné (Both direction). Toto nastavení je znázorněno na obrázku č. 6.



Obr. 6: Nastavení parametrů motoru

Dále se u motoru nastavuje rychlost rozjetí a zpomalení (Start/stop velocity), Maximální rychlost (Maximum velocity), zrychlení (Acceleration) a zpomalení (Deceleration). Tyto parametry je třeba správně určit, aby dvouosý manipulátor pracoval správně. Nastavení těchto parametrů je znázorněno na obrázku č. 7.

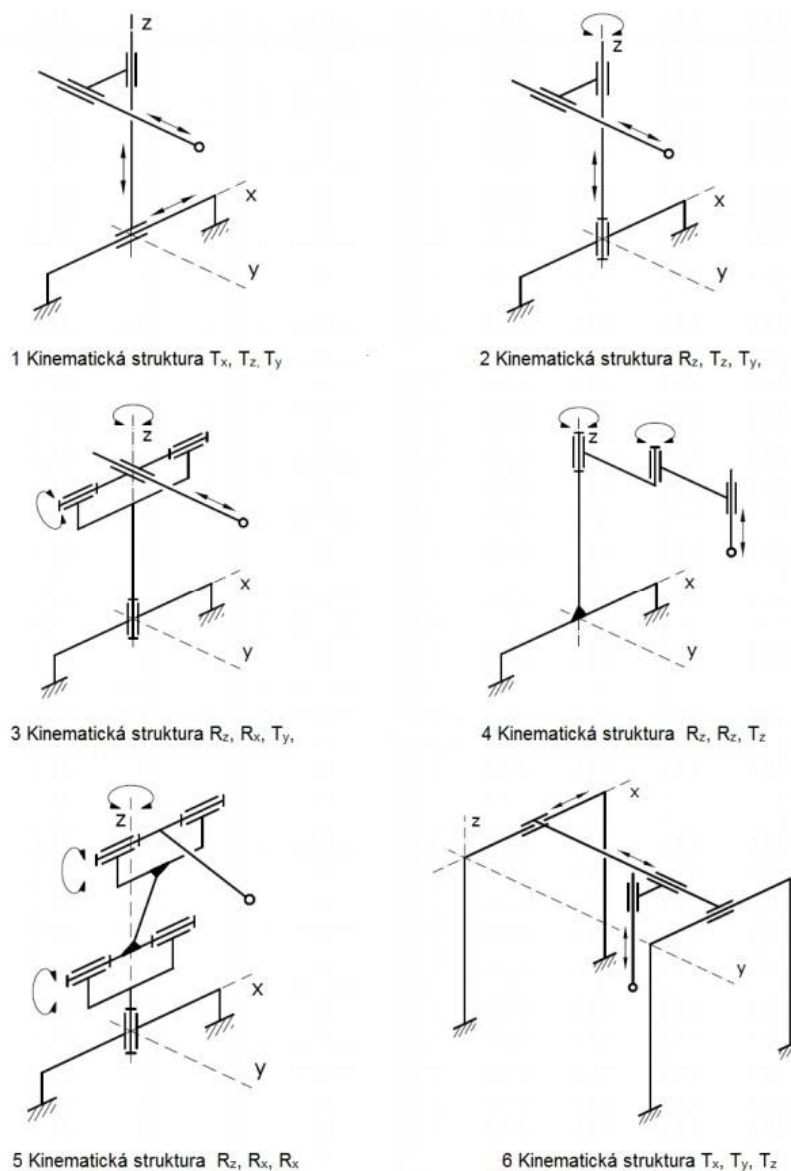


Obr. 7: Nastavení pohybu

2 Manipulátory

Mechanismy manipulátorů jsou reprezentovány kinematickými dvojicemi a jejich vazbami v prostoru, které tvoří kinematický řetězec. V analýze akčního členu systému manipulátoru má struktura složení kinematického řetězce významné místo podle toho do jaké míry ovlivňuje morfologii, vzhled a stavbu jednotlivých manipulátorů. Tento kinematický řetězec reprezentuje polohovací zařízení manipulátoru, které slouží k požadovanému ustavení referenčního bodu. V praxi se nejvíce rozlišují manipulátory základních kinematických struktur: [2]

- tři translační kinematické dvojice TTT
- jedna rotační a dvě translační dvojice RTT
- dvě rotační a jedna translační dvojice RRT
- tři rotační dvojice RRR



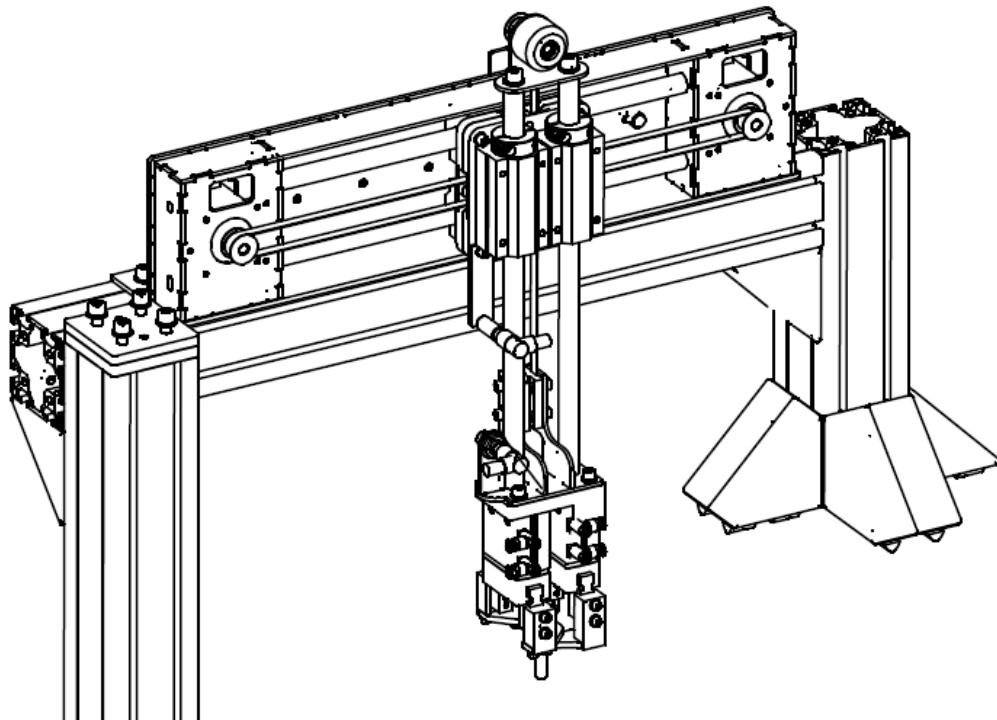
Obr. 8: Základní kinematické struktury manipulátorů [2]

2.1 Kartézský manipulátor

Kartézský manipulátor využívá pouze lineárních vazeb. Tato vlastnost definuje jeho konstrukci, kinematiku a možnosti využití. Z kinematického pohledu se jedná o sériový manipulátor. Sériový (otevřený) manipulátor je takový řetěz, mezi jehož vazbami existuje vždy maximálně jedna cesta. Pohyb v jednotlivé vazbě není vázán na pohyby v dalších vazbách, jako u manipulátorů paralelních. Maximální i operační prostor tříosého kartézského manipulátoru je tvaru hranolu a je dán třemi lineárními vazbami. Každá lineární vazba odebírá po jednom stupni volnosti (rotace kolem os x , y , z), proto má kartézský manipulátor tři stupně volnosti. U některých aplikací kartézského manipulátoru jsou doplněny navíc rotace kolem os nebo má pouze dvě osy a proto jen dvě osy volnosti. Přidáním tří rotačních vazeb pro rotaci kolem každé z os lze dosáhnout konfigurace univerzálního manipulátoru s šesti stupni volnosti. Této konfigurace využívají CNC obráběcí stroje. Lineárních pohybů kartézského manipulátoru lze docílit elektrickými, hydraulickými nebo pneumatickými pohony.

Jednotlivé pohony se mohou mezi sebou různě kombinovat a vytvářet tak hybridní systém těžící z výhod jednotlivých řešení. Výběr pohonu definuje přesnost, plynulost a rychlost polohování manipulátoru. Z těchto důvodů na pohon vyplývají požadavky na velký výkon a minimální moment setrvačnosti. Z montážního hlediska je podstatná hmotnost a tvarové rozložení. Realizace přímočarého pohybu elektrickým motorem využívá točivých strojů nebo lineárních motorů. Pro převod točivého pohybu na lineární lze využít valivých šroubů, ozubených řemenů nebo hřebenu. Lineární elektrický motor je elektromotor, který vznikne rozložením statoru a rotoru do roviny.

Další informace jsou uvedeny v pramenu [3].



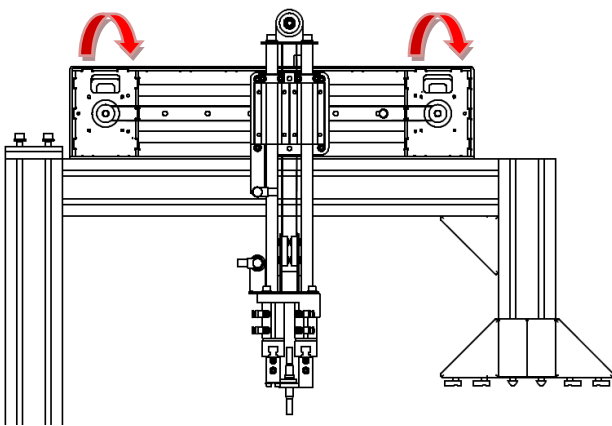
Obr. 9: Uvažovaný kartézský manipulátor

Aplikace kartézských manipulátorů:

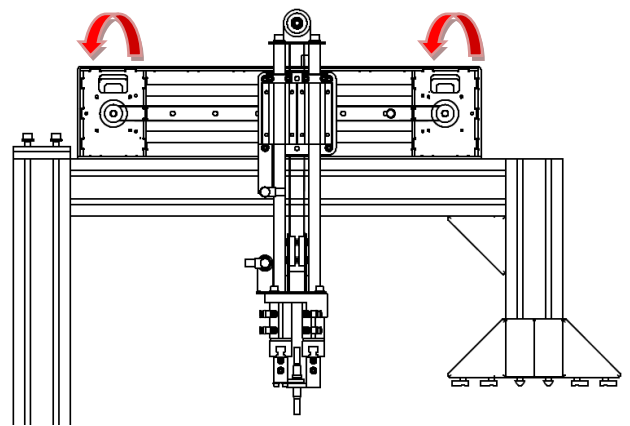
Kartézské manipulátory mají široké spektrum využití. Lze se s nimi setkat u úloh s vysokými požadavky na přesnost a rychlost, nebo v úlohách pro manipulaci s těžkými břemeny. Kartézský manipulátor je možné pro speciální případy vyrobit v portálovém provedení, což zvedá jeho mechanickou pevnost a nosnost. Mezi aplikace kartézského manipulátoru patří např.:

- SMT osazovací automaty
- Manipulační úlohy
- Montážní úlohy
- CNC obráběcí stroje

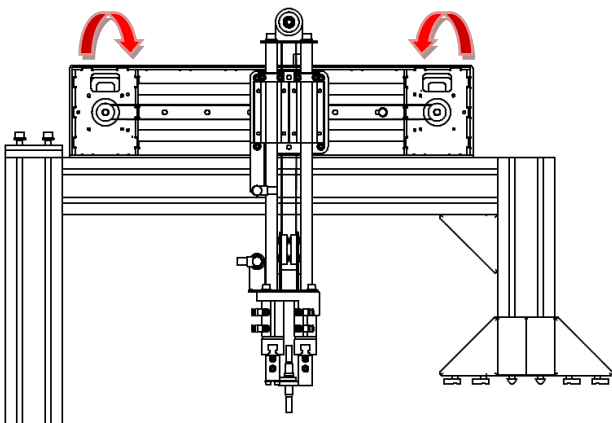
Princip uvažovaného dvousosého kartézského manipulátoru, který bude řízen modulem :



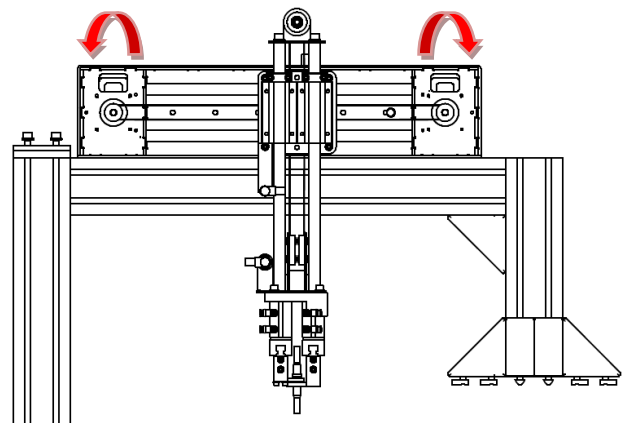
Pohyb dopravníku doleva



Pohyb dopravníku doprava



Pohyb dopravníku nahoru



Pohyb dopravníku dolů

Obr. 10: Princip uvažovaného kartézského manipulátoru

2.2 Driver ES-D508

ES-D508 je modelem z řady Leadshine ES-D, která byla vyvinuta pro snadné řízení krokových motorů. Pracovní napětí tohoto driveru je 20 – 50 V DC a výstupní proud může nabývat hodnot 0.5 – 8 A. Doporučené vstupní napětí je 5, 12 nebo 24 V a vstupní frekvence obdélníkového signálu STEP(PUL) až 200 kHz. Výstupní frekvence driveru je 51,2 kHz. Je schopen řídit krokové motory s enkodéry NEMA 17, 23, 24, a 34. Driver je založen na nejnovější DSP technologii a je řízen pokročilým algoritmem.

Tento driver dostává při řízení krokových motorů zpětnou vazbu z enkodéru stejně jako tomu je při řízení tradičních bezkartáčových servopohonů. To umožňuje uzavření polohové smyčky a vylučuje se tak možná ztráta synchronizace pohybu. Proto je u driverů ES-D508 zcela eliminována ztráta kroků, která je nežádoucí pro většinu aplikací. Velikost výstupního proudu je upravována podle aktuálního točivého momentu. Pokud je požadavek na malý točivý moment motoru, je snížen výstupní proud. Naopak pokud je požadavek na velký točivý moment, tak se proud adekvátně zvětší. Proudové přizpůsobení točivému momentu je velice rychlé, čímž je zajištěna malá odezva systému. Driver ES-D508 také snižuje ohřev motorů až o 50% oproti klasickému řízení krokových motorů. [4]

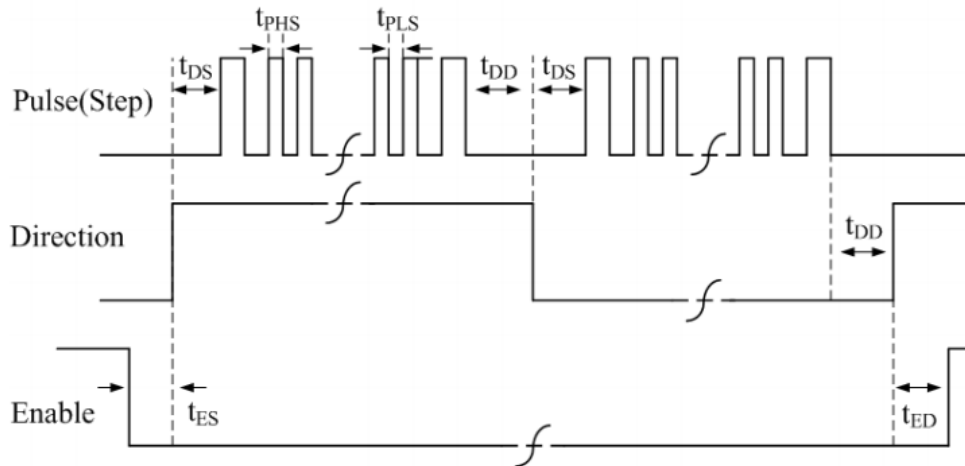


Obr. 11: Driver Leadshine ES-D508 [4]

Popis vstupů:

- PUL** - vstup PUL nebo také STEP je určen pro připojení pulzního signálu. Driver reaguje buď na náběžnou, nebo na sestupnou hranu tohoto signálu podle toho, jak je softwarově nakonfigurován. Při vstupním napětí 12 nebo 24 V je potřeba připojit vstupní signál přes rezistory pro omezení vstupního proudu. To platí i pro vstupy DIR a ENA.
- DIR** - požadovaný směr je dán signálem (tj. úrovní napětí TTL, 0-1) na vstupu DIR, na základě toho driver změni pořadí a smysl (polaritu) připojení vinutí motorů ke zdroji.
- ENA** - tento vstup slouží pro připojení signálu, kterým se zapíná a vypíná driver. Může být zapínán i vypínán log. 0 i log.1, to záleží na softwarovém nastavení, které je možné měnit.

Prodlevy mezi jednotlivými signály:



Obr. 12: Prodlevy mezi jednotlivými signály driveru ES-D508 [4]

Kde:

t_{DS} – čas mezi hranou signálu DIR a prvním pulzem STEP potřebný pro nastavení směru otáčení

t_{PHS} – čas půlperiody vstupního signálu v log.1

t_{PLS} – čas půlperiody vstupního signálu v log.0

t_{DD} - čas mezi pulzem STEP a hranou signálu DIR potřebný pro nastavení nového směru otáčení

t_{ES} – čas po zapnutí driveru signálem ENABLE potřebný pro inicializaci

t_{ED} – čas mezi posledním vstupním pulzem a signálem ENABLE před vypnutím driveru

Tab. 2: Prodlevy a parametry driveru ES-D508 [4]

Vstupní frekvence [kHz]	Výstupní frekvence [kHz]	t_{DS} [us]	t_{PHS}/t_{PLS} [us]	t_{DD} [us]	t_{ES} [ms]	t_{ED} [ms]
200	51,2	>5	>2,5	>5	>50	>50

3 Návrh modulu pro řízení dvouosého kartézského manipulátoru

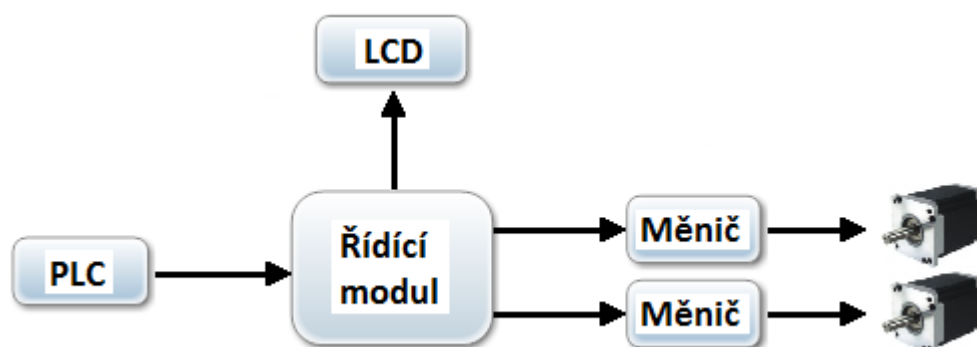
Při volbě součástek, ze kterých bude modul vytvořen, je nutné volit takové parametry, které zajistí:

- odolnost proti rušení
- bezporuchovou funkci pro velký rozsah pracovních teplot
- jednoduché nastavování výstupních parametrů pro obsluhu

Obsluha nemusí vždy umět modul správně používat, proto je nutné zabránit:

- možnosti přepólování napájecího, vstupního a výstupního napětí,
- přivedení obou výstupních napětí na výstup současně (5 V, 24 V),
- přivedení vyššího napětí na svorky modulu, než je požadováno,
- poškození DPS řídicího modulu.

Modul musí být řízen rychlým procesorem, aby dokázal dostatečně rychle reagovat a přenášet změny vstupního signálu na výstup. Zde se jako nejvhodnější jeví ATXMEGA, který má dostatečný počet pinů s asynchronním přerušením a dostatečný počet časovačů pro generování výstupních PWM signálů. Vstupní i výstupní signály jsou oddělené optočleny. Výstupní napětí je volitelné 5 nebo 24 V a galvanicky oddělené převodníky DC/DC napětí. Dá se měnit přepínačem. Modul obsahuje 4 DIP přepínače, kterými je možné nastavit například výstupní PNP nebo NPN logiku. Řídicí modul má celkem 3 galvanicky oddělené části. Vstup, vlastní napájení a výstup. Každé zařízení v rozvodové skříni má galvanicky oddělené napájení. PLC má galvanicky oddělené vstupy, výstupy a napájení CPU. To všechno kvůli maximálnímu omezení rušivých signálů, které mohou přicházet po vodičích spolu se signály očekávanými.



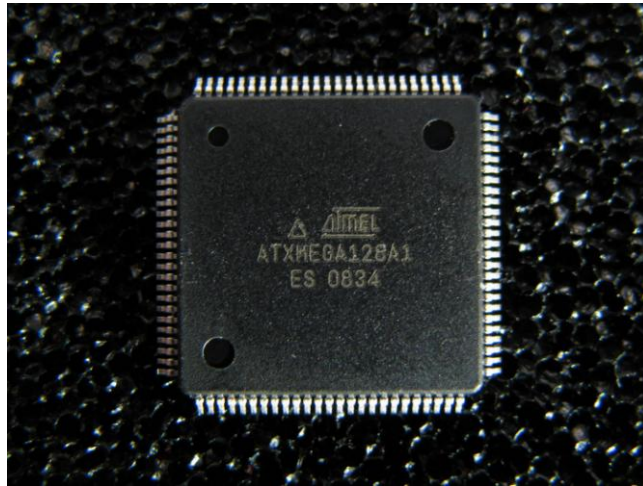
Obr. 13: Blokové schéma zapojení řízení manipulátoru

Řídicí modul bude realizován ve dvou verzích. Jedna verze má LCD displej, druhá verze ho mít nebude. Modul s displejem je široký 70 mm včetně krabičky na DIN lištu.

Jako LCD displej bude použitý typ EADOG128W-6, což je grafický displej o velikosti 64 x 36 mm, na kterém budou zobrazené nastavené parametry. LED diody dávají obsluze informace o aktuálně nastavených signálech a správném napájení.

3.1 Řídící mikrokontrolér ATXMEGA 128A1

ATxmega128A1 je jeden z mikroprocesorů patřící do rodiny XMEGA. Tato rodina mikroprocesorů je založená na architektuře AVR se zvětšenou RISC sadou (138 instrukcí). Mezi hlavní klady této rodiny patří velká výbava periferiemi, vysoký výkon díky maximálnímu taktu až 32MHz při současně nízké spotřebě. Ta dosahuje v aktivním módu maximálně 72mW, přičemž nesmíme zapomenout na to, že vykonání jedné instrukce provede v jednom hodinovém cyklu. [5]

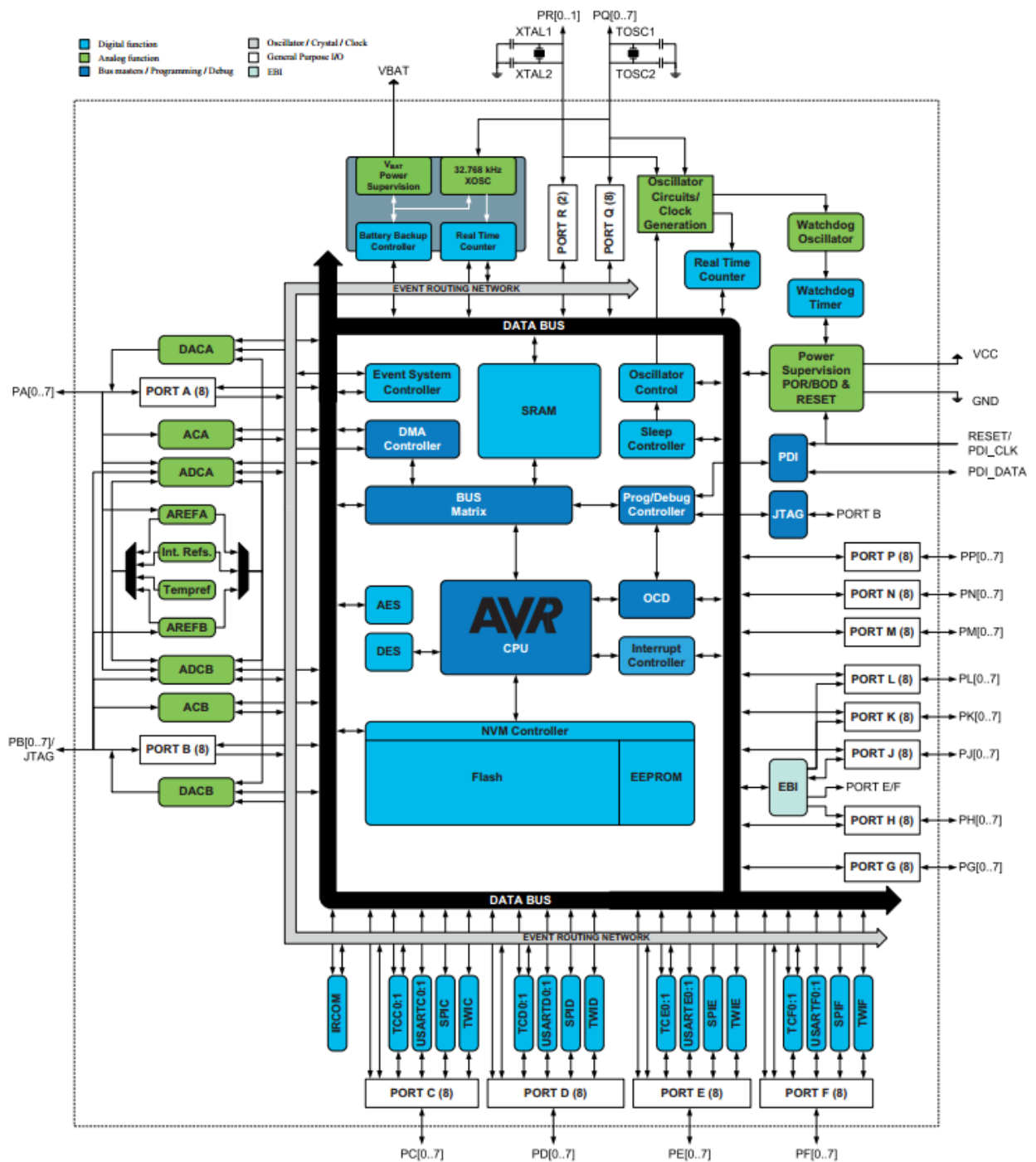


Obr. 14: ATXMEGA 128 A1 v pouzdře TQFP100

Tento mikroprocesor disponuje řadou periferií a rozšíření, mezi které patří například:

- 128 Kbyte paměti flash,
- 2 Kbyte EEPROM,
- 8 Kbyte SRAM,
- rozhraní JTAG a PDI,
- DMA pro přímý přístup do paměti s možností externí komunikace,
- osm kanálů Event System pro předávání události mezi periferiemi,
- osm 16-ti bitových časovačů,
- 16-ti bitový čítač reálného času s odděleným oscilátorem,
- modul kryptovacích mechanismů AES a DES,
- dva osmikanálové analogově digitální převodníky s rozlišením 12 bitů,
- dva dvoukanálové digitálně analogové převodníky s rozlišením 12 bitů,
- interně i externě nastavitelné hodiny za pomoci PLL a předděliček,
- víceúrovňový systém pro obsluhu přerušování.

Výčet výhod není samozřejmě konečný a další možnosti je možno dohledat v [5].



Obr. 15: Blokové zapojení ATXMEGA 128 A1 [5]

Vstupně-výstupní porty

Vstupně-výstupní porty se liší podle provedení mikrokontroléru. U AVR ATXMEGA128A1 je devět portů po osmi pinech, jeden port se čtyřmi piny a jeden s dvěma piny. Celkem tudíž 78 vstupně / výstupních pinů, 20 napájecích pinů, jeden pin pro PDI a jeden pin pro RESET. Tyto porty zajišťují komunikaci s okolím. Ke každému portu jsou přiřazeny 4 registry, které nám usnadňují práci s porty. [5]

DIR – Tento registr určuje směr toku dat pro jednotlivé piny portu. Je-li bit nastaven na log.1, bude pin konfigurován jako výstupní, když bude zapsána log.0, bude pin vstupní.

DIRSET – Registr může být použit namísto registru DIR. Při čtení tohoto registru bude vrácena hodnota DIR.

DIRCLR – registr lze použít k vynulování bitů. Zápisem budou smazány příslušné bity v DIR. Při čtení registru bude vrácena hodnota v DIR.

DIRTGL – registr lze použít k nastavení bitů. Zápisem budou nastaveny příslušné bity v DIR. Při čtení bude vrácena hodnota v DIR.

OUT – hodnota registru bude nastavena na pinech portu, když je port prostřednictvím registru DIR nastaven jako výstupní. K registru OUT přísluší virtuální registry OUTSET, OUTCLR, OUTTGL.

IN – registr pro čtení dat z portu.

INTCTRL – registr ovládá přerušení. Jsou využity bity 0 a 1 pro INT0. Bity 2 a 3 ovládají přerušení INT1

INT0MASK – Bity registru jsou použity pro maskování, které piny budou použity jako zdroje přerušení INT0. Pro každý pin rozhoduje PINCTRL registr.

INT1MASK – Bity registru jsou použity pro maskování, které piny budou použity jako zdroje přerušení INT1. Pro každý pin rozhoduje PINCTRL registr.

PINnCTRL -Registr ovládá vlastnosti jednoho pinu v portu. Takže ke každému portu je přiřazeno osm těchto registrů. Zápis např. PORTD_PIN1CTRL = 0x01;

Bit 7 – SRLen: nastavením bitu omezíme rychlost přeběhu na pin n.

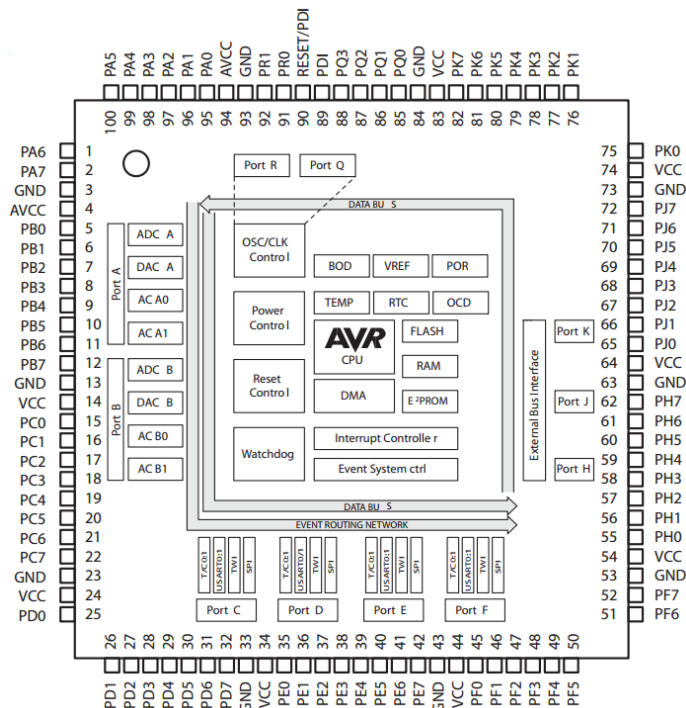
Bit 6 – INVEN: invertuje vstup a výstup dat na pin n.

Bit 5 : 3 – OPC: Output/pull konfigurace pinu n.

Bit 2 : 0 – ISC: Nastavuje vstup a konfiguraci pinu podle tabulky.

Konfigurace rozhoduje o tom, jakým způsobem může být vyvoláno přerušení.

Ke každému portu se rovněž váže alternativní funkce (analogově-digitální převod, přerušení, pulzně-šifkovaná modulaci, čítač,...) Další informace jsou uvedeny v pramenu [5].



Obr. 16: Označení pinů mikroprocesoru ATxMEGA 128A1 [5]

Zdroj hodinového signálu

Zdroj hodinového signálu u procesorů rodiny ATxmega není ovládán pomocí bitů pojistek, jak je zvykem u ATmega. Hodinový kmitočet u procesorů xmega je řízen programově, prostřednictvím několika registrů. Po resetu je vždy připojen interní RC oscilátor 2MHz. Jiný zdroj signálu můžeme připojit nastavením registrů. Je možnost připojit interní 32MHz oscilátor, 32,768kHz, nebo externí krystal. Maximální kmitočet procesoru je 32MHz, některé periferie mohou běžet na vyšším kmitočtu. Obvod procesoru sleduje chod oscilátorů a zobrazuje jej na registru OSC_STATUS. Takže před připojením oscilátoru je možné a zároveň vhodné zjistit, jestli oscilátor funguje. Chod procesoru nelze zablokovat špatným naprogramováním pojistek, jako u ATmega.

// spustí 32 MHz oscilátor

```

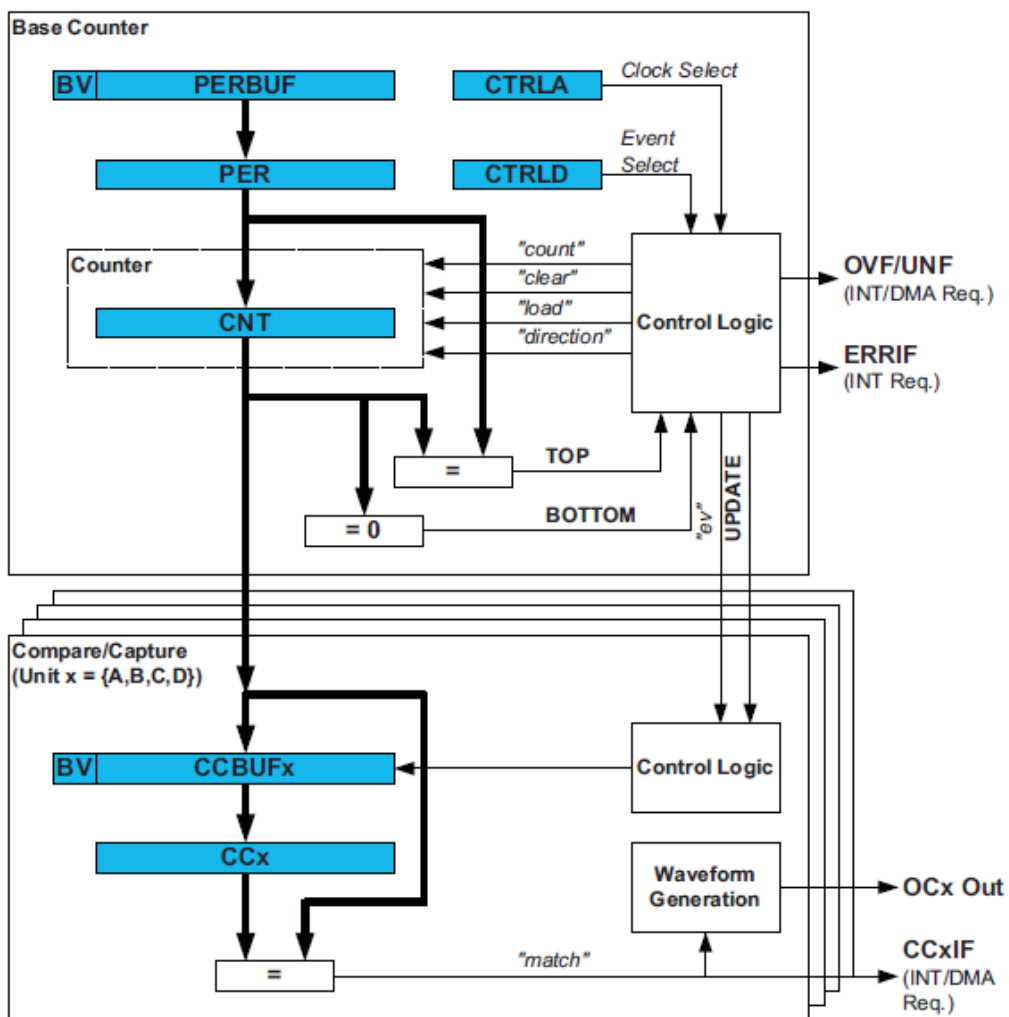
OSC_CTRL |= OSC_RC32MEN_bm;
// ceka, az bude oscilator aktivni
while(!(OSC_STATUS & OSC_RC32MRDY_bm));
// zápis hodnoty 0xD8 do registru, který umožní další zápis
CCP = CCP_IOREG_gc;
// připojení 32MHz oscilatoru
CLK_CTRL = CLK_SCLKSEL_RC32M_gc;

```

Registr OSC_CTRL spouští oscilátory, registr OCS_STATUS zobrazuje jejich správnou funkci. Registr CLK_CTRL připojí oscilátor k procesoru. Zápis do tohoto registru je chráněn tím, že je nutné napřed zapsat hodnotu 0xD8 do CCP. Potom je zápis do chráněných registrů umožněn po dobu čtyř cyklů. Všechny přerušení jsou během této doby zakázány. Nevyřízené přerušení bude zpracováno po ukončení této doby. [6]

Čítač/časovač

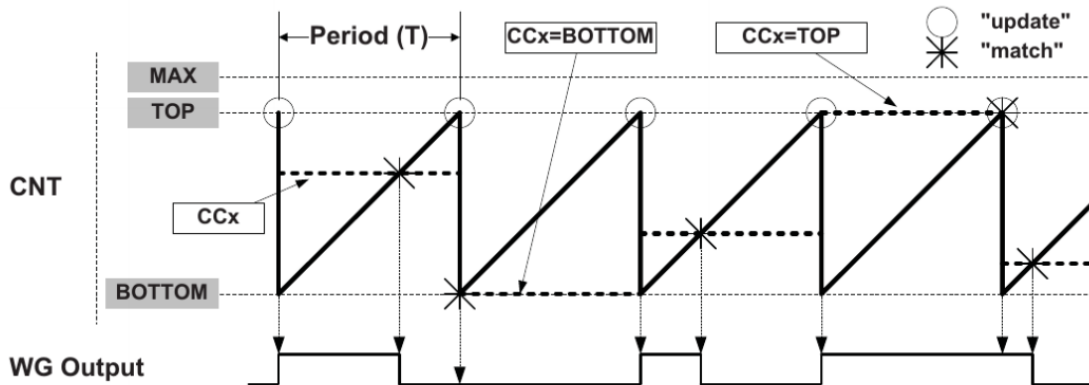
Mikrokontrolér ATXMEGA128A1 obsahuje periférie umožňující generovat časový úsek, (časovač) případně načítat vstupní pulsy (čítač). Díky tomu můžeme vytvořit požadovaný časový úsek, změřit krátký časový interval, vytvořit jednoduchý čítač, atd. K tomu lze využít některého z nabízených časovačů (čítačů). ATXMEGA128A1 obsahuje osm 16-ti bitových časovačů. K čítání a časování se využívají registry CNT(Counter register), PER,PERBUF(Period register, with buffer), CC,CCBUF(Capture and Compare register, with buffer), CTRLn (Control register n). Další informace je možné dohledat v [5].



Obr. 17: Blokové schéma čítače/časovače [5]

Generování PWM signálu

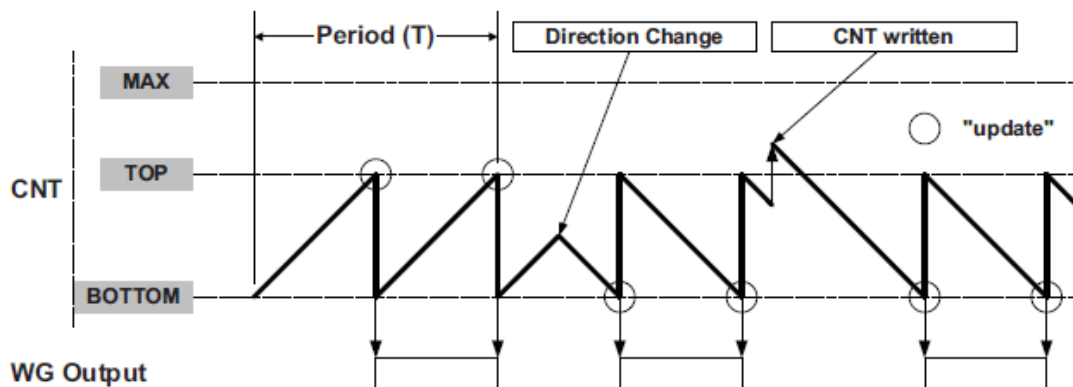
Generování PWM (Pulse Width Modulation) signálu s periodou T je řízeno registrem PER. U průběhu Jednoduchého sklonu PWM můžeme komparační hladinu nastavit v rozmezí hodnot BOTTOM až TOP. V průběhu generace PWM můžeme měnit komparační úroveň a tím docílíme různé délky periody signálu. Průběh generace PWM signálu pro čítač je na obrázku č. 17. Při generování PWM signálu můžeme dosáhnout maximální frekvence okolo 65 kHz. [5]



Obr. 18: Generování Jednoduchý sklon PWM signálu [5]

Frekvenční generování

Generování FRQ signálu (Frequency Waveform Generation) s periodou T je řízeno CCA registrem. Výstup frekvenční modulace je zajištěn při každém porovnávacím zápisu mezi CNT a CNA registry, jak je znázorněno na obrázku č. 18. Výhodou tohoto generování je vysoká frekvence signálu, která může dosahovat až poloviny frekvence externího oscilátoru. [5]



Obr. 19: Generování frekvenčního (FRQ) signálu [5]

Externí přerušení

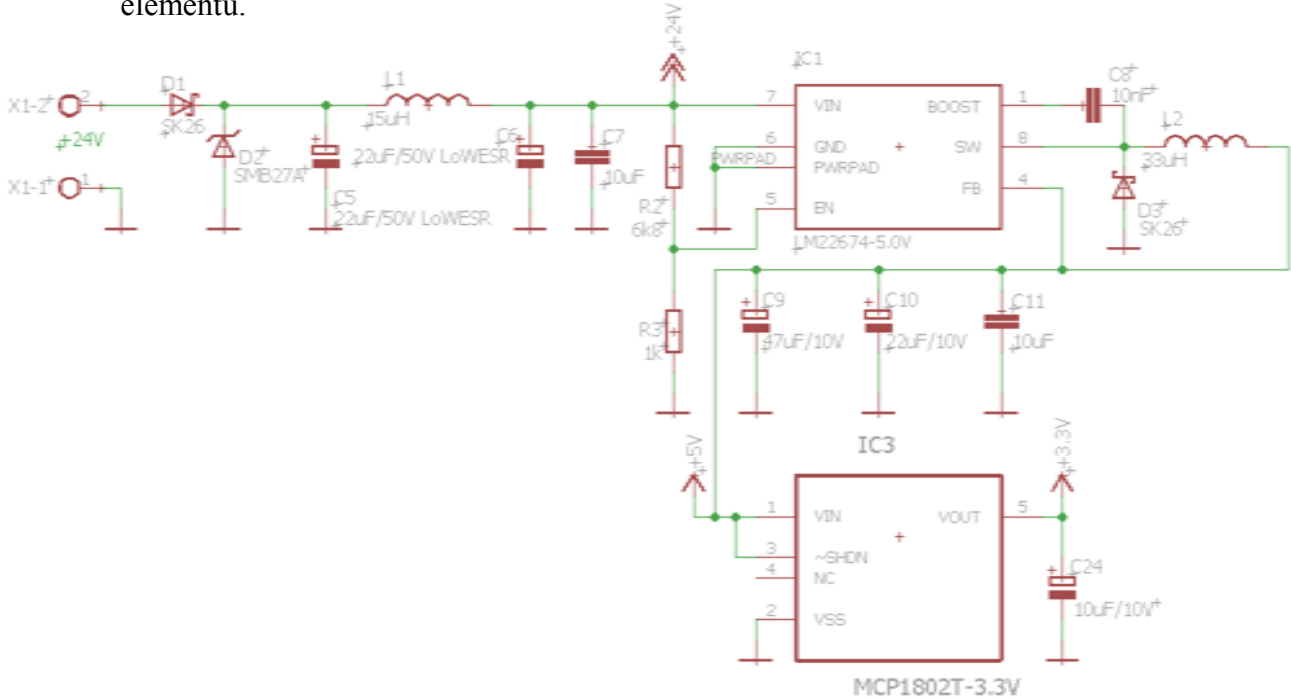
Programovatelný, víceúrovňový ovladač přerušení PMIC, nastavuje priority požadavků na přerušení. Stejně jako u jiných procesorů by měl být nastaven před globálním povolením přerušení, pomocí bitu SREG.

Je-li povoleno přerušení, PMIC přijímá požadavek na přerušení a ten buď vykoná, nebo čeká, na základě nastavených priorit. Po vykonání požadavku na přerušení, se PMIC vrátí na původní úroveň, nebo do stavu před voláním přerušení. To znamená, že přerušení s vyšší prioritou, přeruší vykonávání programu přerušení s nižší prioritou. Po zpracování přerušení s vyšší prioritou, bude zpracováno další přerušení. V základním nastavení mají vektory s nejnižší adresou nejvyšší prioritu. Například, vektor přerušení Reset má nejvyšší prioritu a nejnižší adresu. V Xmega existují dva typy přerušení, to jsou statické a dynamické. Statické přerušení jsou ty, které mají pevnou adresu vektoru a zároveň nízkou prioritu přerušení.

Dynamické chování je dosaženo Round-robin Scheduling (RRS) technikou. RRS zajišťuje, že přerušení nízké úrovně není vynecháno. Použije-li se tato technika, pak poslední vyvolané přerušení bude mít nejnižší prioritu. V nízké úrovni přerušení existuje mechanismus, který zajistí, že všechny přerušení jsou zavolány v určitém časovém horizontu. [6]

3.2 Zdroj napájení

Na řídicí modul je přivedeno napětí 24 V DC z externího zdroje umístěného v rozvodové skříni (vstupy X1-1 a X1-2). Dioda SK26 slouží jako ochrana proti přepólování. Transil SMB 27A ochrání obvod při přivedení napětí vyššího než 27 V. Tyto ochrany jsou nutné, aby se zamezilo zbytečnému poškození obvodů. Pracovník, který bude zapojovat modul do rozvodové skříně, může dráty prohodit. Kondenzátory C5, C6 a cívka L1 tvoří odrušovací filtr, který zabraňuje vniknutí nežádoucích rušivých elementů.



Obr. 20: Schéma zapojení zdroje

Spínaný zdroj LM22674

Tento stabilizátor se vyrábí v nastavitelném nebo nenastavitelném provedení. Hodnota výstupního napětí se tedy buď dá měnit a nebo je nastavena pevně. V tomto případě je výstupní napětí 5 V bez možnosti nastavení. Vstupní napětí může být 8 až 42 V. Ke stabilizaci je nutné použít kondenzátory s nízkou hodnotou ESR kvůli omezení zvlnění napětí. Spínací frekvence stabilizátoru je 500 kHz. Teplotní rozsah pro použití je -40 až +125 °C. Odporový dělič tvořený rezistory R2 a R3 slouží k zapnutí stabilizátoru. Na pin EN (enable) musí být přivedeno napětí alespoň 1,3 V, aby obvod začal stabilizovat. Kondenzátor C7 slouží k omezení zvlnění vstupního napětí. Kondenzátor C8 pomáhá spínat mosfet, který zajišťuje maximální výstupní proud stabilizátoru 500 mA. Cívka L2 slouží ke stabilizaci režimu spínaného zdroje a její hodnota je určena hlavně podle očekávaného výstupního zvlnění proudu. Schottkyho dioda D3 je důležitá kvůli své rychlosti a proudu, který ji může protékat a musí být minimálně roven maximálnímu výstupnímu proudu stabilizátoru. Kondenzátory C9 – C11 slouží ke stabilizaci a filtraci výstupního napětí.

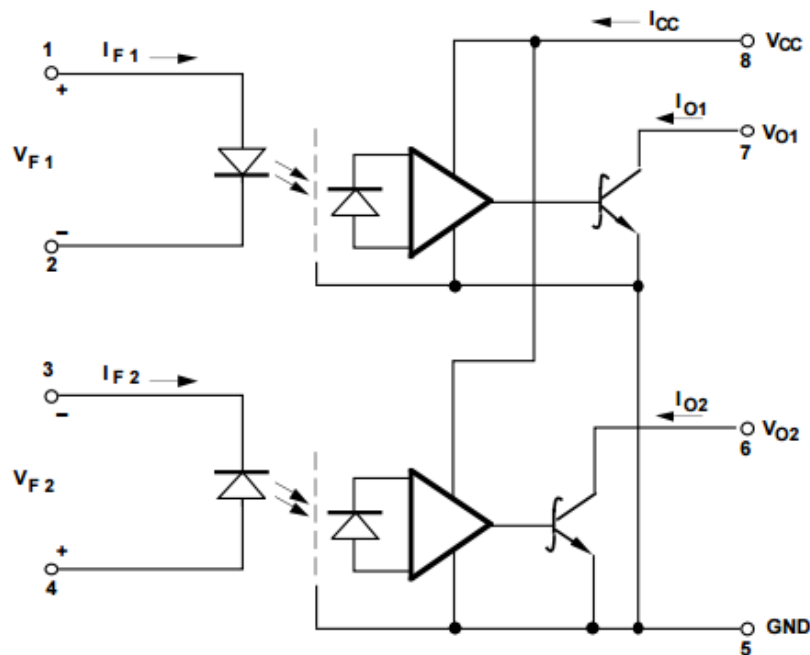
Nenastavitelný stabilizátor MCP1802T zajišťuje napětí 3,3 V pro napájení většiny komponent v řídicím modulu. Jeho maximální výstupní proud je 300 mA, což je dostačující. Shutdown pin má trvale připojen na vstupní napájení, tím je zajištěna trvalá stabilizace. Výhodou je ochrana proti zkratu.

3.3 Galvanické oddělení vstupních signálů

Pro oddělení vstupních signálů jsou použity optočleny HCPL0630. Tyto optočleny jsou dvoukanálové a mají opticky propojená logická hradla, která kombinují GaAsP emitující diody a integrované vysoce citlivé fotodetektory. Fotony z fotodiody jsou shromažďovány ve fotodetektoru a proud je zesilován vysoce ziskovým lineárním zesilovačem. Optočlen je napěťově, proudově a teplotně kompenzovaný. Optočlen má izolační napětí 5 kV/us. Unikátní design poskytuje maximální izolaci střídavého i stejnosměrného napětí a je kompatibilní s TTL a LSTTL. Hodnoty parametrů tohoto obvodu jsou definované pro pracovní teploty od $-40\text{ }^{\circ}\text{C}$ do $+85\text{ }^{\circ}\text{C}$. Tento teplotní rozsah vyhovuje. Hradlové optočleny HCPL jsou vyvinuty pro vysokorychlostní logické signály o frekvenci až 10 MHz. [7]

Frekvence kterou optočlen bude přenášet je max. 100 kHz. HCPL signál s takovou frekvencí velice rychle převede z vstupu na výstup a tím vznikne pouze minimální zkreslení, které je ještě únosné. Dvouosý manipulátor musí dojíždět na definovanou pozici s maximální přesností. Pokud bude modul řídit automatizovaný stroj, který bude v provozu non-stop a vyrobí miliony výrobků ročně, nemůže se jakákoliv poloha, na kterou bude dvouosý manipulátor jezdit, měnit ani o desetinu milimetru.

Hlavním důvodem k použití tohoto optočlenu je galvanické oddělení napájení výstupů PLC od napájení modulu řízení dvouosého kartézského manipulátoru. PLC má tři galvanicky oddělené napájecí větve – napájení procesoru, vstupů a výstupů. Tímto oddělením je možné zamezit vzniku rušení, které přichází z každého výkonnějšího stroje umístěného na hale, kde je stroj s dvouosým manipulátorem umístěn.

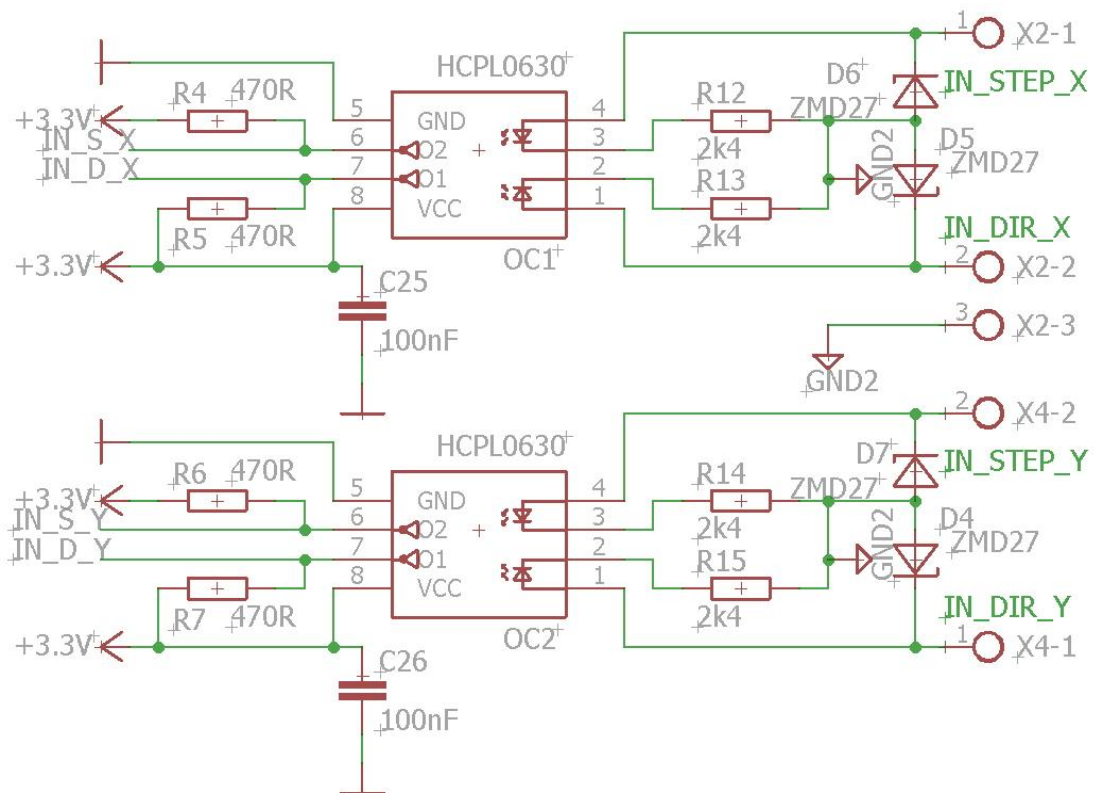


Obr. 21: Vnitřní zapojení HCPL0630 [7]

Typické využití optočlenu HCPL0630

- Oddělování vysokorychlostních logických signálů
- Systémová rozhraní mikroprocesorů
- Oddělení signálů z přijímačů
- Rozhraní pro počítačové periferie
- Oddělení signálů v A/D a D/A převodech
- Oddělení výkonových tranzistorů pro řízení motorů

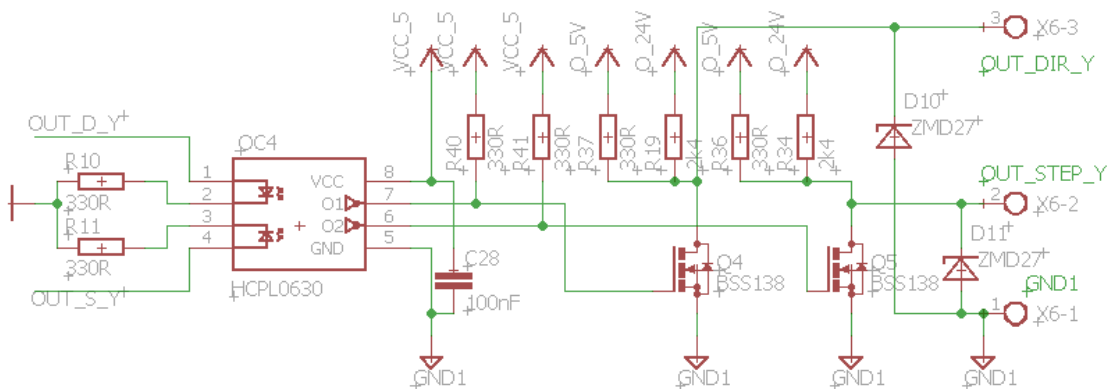
Na obrázku č. 22 je znázorněno schéma zapojení optočlenů v modulu pro řízení dvouosého manipulátoru. Vstupní proud optočlenu by se měl pohybovat v rozmezí 5 – 15 mA. Volím tedy hodnotu 10 mA, která je nastavena SMD rezistory s hodnotou 2k4 při napětí 24 V z PLC. SMD rezistory mají pouzdra 2010 kvůli výkonu. Na výstupech jsou ochranné rezistory 470 Ω omezující proud do vstupů mikrokontroléru. Kondenzátor C25 je blokovací. Jeden optočlen odděluje jak STEP, tak i DIR signál. Na vstupech jsou Zenerovy diody jako ochrana proti přepólování nebo přivedení vyššího napětí než 27 V. Zenerova dioda plní funkci ochrany, tedy součástky, která se při překročení závěrného napětí nedestruktivně prorazí (tunelový průraz). Veškerý proud (max. 20 mA) poteče přes diodu a nedojde tak k destrukci obvodu za diodou. Transil zde nemohl být použit kvůli vysoké parazitní kapacitě, která by způsobovala zkreslení signálu.



Obr. 22: Schéma zapojení vstupu

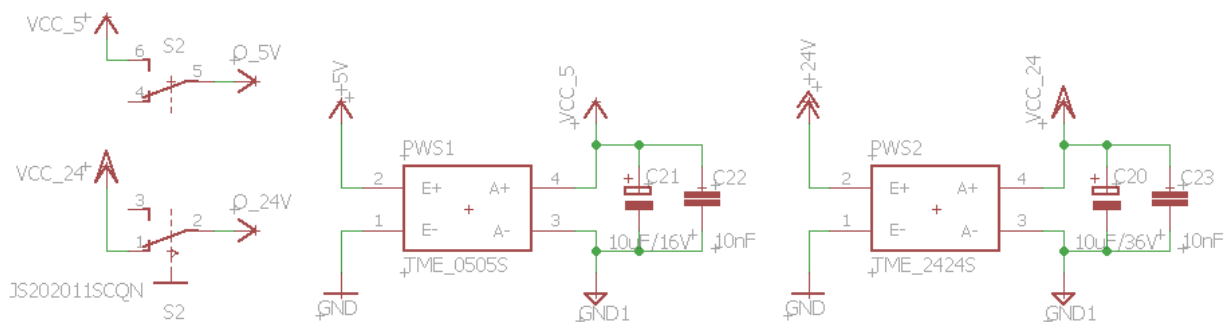
3.4 Galvanické oddělení výstupních signálů

Stejně jako u vstupních signálů jsou k oddělení použité hradlové optočleny HCPL 0630. Nevýhodou tohoto optočlenu je výstupní napětí pouhých 5 V. Výstupní napětí je třeba přepínat mezi 5 a 24 V podle typu použitého driveru. Nabízela se možnost použití tranzistorového optočlenu. Některé typy tranzistorových optočlenů mají výstupní napětí až 70 V. Pro tento případ by tranzistorový optočlen byl ideální, kdyby nezkresloval přenášený signál. Doba náběžných a sestupných hran se u hradlového optočlenu pohybuje řádově v desítkách nanosekund, kdežto u tranzistorového optočlenu řádově v mikrosekundách. Čím menší změna obdélníkového signálu, tím přesnější budou dojezdy manipulátoru, obzvláště pokud bude zapnutý 24 hodin denně po celý rok. Proto bylo zachováno řešení s HCPL 0630, kde se výstupních 5 V používá pouze ke spínání tranzistorů NMOS. Přes tyto tranzistory je přivedeno buď 5 V, nebo 24 V. Pokud zvolíme PNP logiku, driver bude reagovat na sestupnou hranu obdélníkového signálu. U NPN logiky bude driver reagovat na náběžnou hranu obdélníkového signálu. Log.0 bude na výstupu při sepnutém tranzistoru a log.1 pokud tranzistor není sepnutý. Jakékoliv řešení pomocí PMOS tranzistorů není možné, protože by byl PMOS trvale sepnutý (napětí na bázi vždy menší jak na kolektoru). Doba náběžné a sestupné hrany obdélníkového signálu po průchodu hradlovým optočlenem a tranzistorem se pohybuje okolo 60 ns, což je pořád mnohonásobně méně než při použití tranzistorového optočlenu.



Obr. 23: Schéma zapojení výstupu

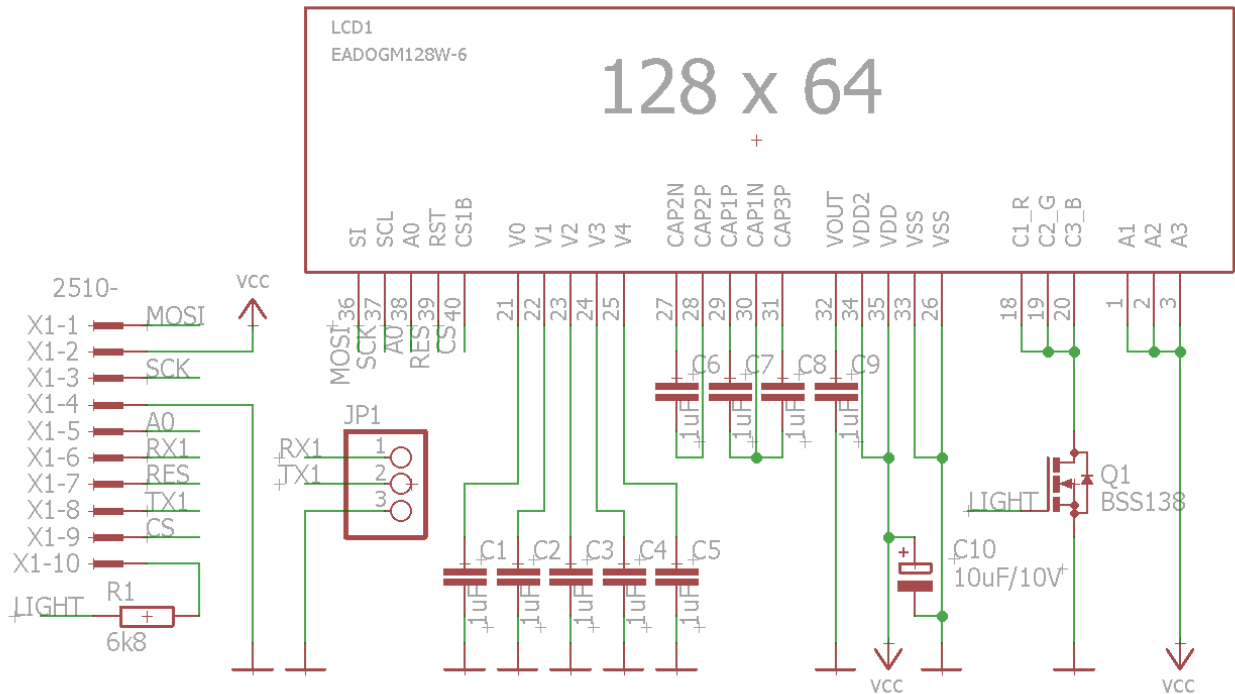
Pro galvanické oddělení výstupních signálů jsou zde použité převodníky napětí 5 V_5V a 24 V_24 V. Výstupní napětí je voleno přepínačem S2.



Obr. 24: Schéma zapojení přepínače a DC/DC převodníků

3.5 Zobrazovací jednotka

pro přehledné zobrazování informací byl vybrán monochromatický grafický LCD displej s rozlišením 128x64 pixelů s označením EAD0GL128W-6. Displej je vybaven možností připojení na sériovou směrnicí SPI, přes kterou jsou přenášena data z mikrokontroléru na displej. Samotné zobrazování zajišťuje řadič integrovaný v displeji. Při zapojení Low Power specifikované v dokumentaci dostačuje displeji napájení 3V.



Obr. 25: Schéma zapojení displeje

Na obrázku č. 25 je k vidění zapojení displeje. Kondenzátory C1 – C9 jsou připojeny k integrované nábojové pumpě, která zajišťuje vyšší napětí potřebné pro samotnou funkci displeje. Kondenzátor C10 slouží jako blokovací a je umístěn co nejbližší napájecích svorek displeje. Samotný LCD neobsahuje podsvícení, proto je nutné jej dokoupit samostatně. Podsvícení se skládá ze tří větví LED. Anody LED jsou připojeny přímo na baterii. Rezistory pro omezení proudu zde nejsou zapotřebí, protože při maximálním jasu teče větve proud 180mA, na který je dimenzován stabilizátor. Katody LED jsou připojeny na potenciál GND přes tranzistor Q1. Hodnota rezistoru R1 je dostačující pro ochranu báze tranzistoru Q1. [8]

4 Firmware

Program pro procesor ATXMEGA128A1 je psán v jazyce C. K realizaci programu bylo využito Atmel studio verze 6.2. Diplomová práce byla testována se třemi verzemi firmwaru. Pro správný provoz manipulátoru musí být každý krok přijatý z PLC generován řídicí jednotkou na vstup driveru. V řídicí jednotce se tedy nesmí žádné kroky ztrácet, ani přibývat. Z tohoto důvodu bylo při psaní firmwaru pokaždé využito přerušení procesoru. V rámci vývoje řídicí jednotky pro kartézský manipulátor byly vytvořeny dvě verze řídicí jednotky. Firmware se dá použít pro obě verze.

Význam použitých proměnných :

IN_D_X – signál na pinu 2 portu H
- Vstup signálu DIR pro osu X

IN_D_Y – signál na pinu 2 portu J
- Vstup signálu DIR pro osu Y

OUT_D_X – signál na pinu 5 portu K
- Výstup signálu DIR pro osu X

OUT_D_Y – signál na pinu 3 portu K
- Výstup signálu DIR pro osu Y

IN_S_X – signál na pinu 3 portu E
- Vstup signálu STEP pro osu X

IN_S_Y – signál na pinu 3 portu F
- Vstup signálu STEP pro osu Y

OUT_S_X – signál na pinu 3 portu C
- Výstup signálu STEP pro osu X

OUT_S_Y – signál na pinu 3 portu D
Výstup signálu STEP pro osu Y

INTER_STEP_X a INTER_STEP_Y

- pokud je v této proměnných zapsáno číslo 0, manipulátor se bude pohybovat vodorovně nebo svisle
- pokud je v této proměnných zapsáno číslo 1, manipulátor se bude pohybovat v obou osách zároveň

4.1 Verze firmwaru 1.0

V této verzi se na každé přerušení na vstupech STEP vytvoří příznak přerušení a zařadí se do fronty příznaků v paměti procesoru. Takto vytvořený příznak je vyvolán a zpracován hned nebo až na něj dojde řada, podle jeho priority.

Hlavní program „main()“ je prázdný. Veškerý zdrojový kód je napsán v těle přerušení na signál STEP osy X a na signál STEP osy Y.

Hlavní myšlenkou je generování pulzu na výstupu řídicí jednotky pro směr DIR, se kterým přišel na vstup. V těle přerušení se generuje jeden krok v ose, pro kterou bylo přerušení vyvoláno. Výsledný sdružený pohyb manipulátoru je schodovitý.

Nastavení přerušení (osa Y se liší jiným portem) :

```
void Interrupt_IN_S_X ()
// Indikace přerušení (sestupné hrany signálu STEP osa X - PE3)
{
    PORTE.PIN3CTRL = PORT_ISC_FALLING_gc;
                                     // reaguje na sestupnou hranu
    PORTE.INT0MASK = 0x8;
                                     // aktivní přerušení na pinu 3, port E
    PORTE.INTCTRL = PORT_INT0LVL_HI_gc;
                                     // nastavení přerušení INT0
    PMIC.CTRL |= PMIC_HILVLEN_bm;
                                     // nejvyšší priorita přerušení
    //PMIC.STATUS |= PMIC_HILVLEX_bm;
    sei();
                                     // aktivuje globální přerušení
}
```

Část zdrojového kódu pro jednoduché pohyby manipulátoru vodorovně :

```
//Přerušeni po přivedení sestupné hrany na IN_S_X(PE3)
ISR (PORTE_INT0_vect)
{
    PORTE.INTFLAGS = 0x01; // mazání příznaku přerušeni

    INTER_STEP_X = 1;

    if (INTER_STEP_Y == 0 && IN_D_X == 0)
        // pohyb doprava
        {
            _delay_us(6);
            OUT_D_X = 0;
            OUT_D_Y = 0;
            _delay_us(6);
            OUT_S_X = 1;
            OUT_S_Y = 1;
            _delay_us(3);
            OUT_S_X = 0;
            OUT_S_Y = 0;
        }
    else if (INTER_STEP_Y == 0 && IN_D_X == 1)
        // pohyb doleva
        {
            _delay_us(6);
            OUT_D_X = 1;
            OUT_D_Y = 1;
            _delay_us(6);
            OUT_S_X = 1;
            OUT_S_Y = 1;
            _delay_us(3);
            OUT_S_X = 0;
            OUT_S_Y = 0;
        }
    ...
}
```

Princip generování výstupních signálů STEP a DIR je stejný jak pro vodorovné, tak pro svislé pohyby manipulátoru. Z tohoto důvodu je zde uveden pouze zdrojový kód pro jeden z nich.

Na každý příznak přerušeni se generuje jeden pulz STEP a podle vstupního signálu DIR dané osy se nastaví výstupní signály DIR pro obě osy manipulátoru. Část kódu pro jednoduché pohyby je volána pouze pokud je splněna podmínka `INTER_STEP_Y == 0`. `INTER_STEP_Y` má hodnotu 1 pokud mikroprocesor přijímá signály STEP z PLC pro obě osy zároveň (sdružený pohyb – příznaky přerušeni pro obě osy signálu step aktivní).

Část zdrojového kódu pro sdružené pohyby manipulátoru :

```

if (INTER_STEP_X == 1 && INTER_STEP_Y == 1)
{
    if (IN_D_X == 0 && IN_D_Y == 0)
    // pohyb doprava - nahoru - DOPRAVA
    {
        _delay_us(6);
        OUT_D_X = 0;
        OUT_D_Y = 0;
        _delay_us(6);
        OUT_S_X = 1;
        OUT_S_Y = 1;
        _delay_us(3);
        OUT_S_X = 0;
        OUT_S_Y = 0;
    }
    else if (IN_D_X == 0 && IN_D_Y == 1)
    // pohyb doprava - dolů - DOPRAVA
    {
        _delay_us(6);
        OUT_D_X = 0;
        OUT_D_Y = 0;
        _delay_us(6);
        OUT_S_X = 1;
        OUT_S_Y = 1;
        _delay_us(3);
        OUT_S_X = 0;
        OUT_S_Y = 0;
    }
    ...
}

```

Princip generování výstupních signálů STEP a DIR je stejný jak pro osu X, tak pro osu Y u sdružených pohybů manipulátoru. Z tohoto důvodu je zde uveden pouze zdrojový kód pro jeden z nich. V přerušení pro osu X se generuje krok pro pohyb ve vodorovném směru. V přerušení pro osu Y se generuje krok pro pohyb ve svislém směru.

Na každý příznak přerušení se generuje jeden pulz STEP a podle vstupního signálu DIR dané osy se nastaví výstupní signály DIR pro obě osy manipulátoru.

4.2 Verze firmwaru 1.1

V této verzi se na každé přerušení na vstupech STEP vytvoří příznak přerušení a zařadí se do fronty příznaků v paměti procesoru. Takto vytvořený příznak je vyvolán a zpracován hned nebo až na něj dojde řada, podle jeho priority.

Převážná část zdrojového kódu je napsána v těle hlavní funkce „main()“. Pomocí přerušení se pouze přičítá nebo odečítá počet kroků pro danou osu a směr pohybu manipulátoru.

Hlavní myšlenkou je vytvoření zásobníku s kroky a jejich postupné generování na výstupu v časových úsecích. U této verze je zajištěno méně časté přepínání směrů DIR v driveru.

Nastavení přerušení je stejné jako ve verzi 1.0.

Význam použitých proměnných pro verzi 1.1:

COUNT_TIME

- proměnná, do které se zapisuje čas v mikrosekundách
- slouží k zajištění generování pouze určitého počtu kroků během jedné změny směru

COUNT_STEP_X

- proměnná, do které se přičítá nebo odečítá krok v rámci jednoho přerušení

COUNT_STEP_Y

- proměnná, do které se přičítá nebo odečítá krok v rámci jednoho přerušení

COUNT_STEP_MAIN

- pomocná proměnná, která slouží k určení směru, kterým se bude manipulátor pohybovat

MOVEMENT

- může nabývat hodnot 1 – 4, tato čísla představují směry, kterými se může manipulátor pohybovat

REF_TIME

- referenční časový úsek, po kterou je zachován jeden směr pohybu

ONLY_FIRST

- slouží k zajištění pouze jednoho přepnutí směru pohybu za referenční časový úsek

Část zdrojového kódu pro přerušení :

```
ISR (PORTE_INT0_vect)
{
    if (IN_D_X == 0)                // pohyb doprava ++
    {
        COUNT_STEP_X++;
    }
    else if (IN_D_X == 1)           // pohyb doleva --
    {
        COUNT_STEP_X--;
    }

    PORTE.INTFLAGS = 0x01; // maže příznak přerušení
}
...

ISR (PORTF_INT0_vect)
{
    if (IN_D_Y == 0)                // pohyb nahoru ++
    {
        COUNT_STEP_Y++;
    }
    else if (IN_D_Y == 1)           // pohyb dolů --
    {
        COUNT_STEP_Y--;
    }

    PORTF.INTFLAGS = 0x01; // maže příznak přerušení
}
...
```

V této části kódu dochází k vyhodnocování kroků. PLC vyšle určitý počet kroků definující vzdálenost, kterou má manipulátor ujet buď jednoduchým nebo sdruženým pohybem. Tyto kroky se podle směru definovaného z PLC buď odečítají nebo přičítají do dané proměnné.

Část zdrojového kódu hlavní funkce „main()“ pro určení směru pohybu :

```
if (COUNT_TIME == 0)
{
    if (COUNT_STEP_X > 0)
    {
        COUNT_STEP_MAIN = COUNT_STEP_X;
        MOVEMENT = 1;
    }
    else if (COUNT_STEP_X < 0)
    {
        COUNT_STEP_MAIN = labs(COUNT_STEP_X);
        MOVEMENT = 3;
    }

    if ((COUNT_STEP_MAIN < COUNT_STEP_Y) && (COUNT_STEP_Y > 0))
    {
        COUNT_STEP_MAIN = COUNT_STEP_Y;
        MOVEMENT = 2;
    }
    else if ((COUNT_STEP_MAIN < labs(COUNT_STEP_Y)) &&
(MOVEMENT != 2) && (COUNT_STEP_Y < 0))
    {
        MOVEMENT = 4;
    }
}
```

V této části kódu dochází k porovnávání proměnných s kroky. V případě záporných hodnot v těchto proměnných je porovnávána absolutní hodnota počtu kroků.

Podle proměnné s největším počtem kroků se nastavuje směr pohybu manipulátoru.

Podmínka `COUNT_TIME == 0` je splněna při určování směru pro první příchozí kroky z PLC a po uplynutí doby `REF_TIME` – dokončení generování určitého počtu pulzů pro jeden směr a nastavení nového směru pohybu.

Část zdrojového kódu hlavní funkce „main()“ pro generování kroků v jednom směru :

```
if((COUNT_STEP_X > 0) && (MOVEMENT == 1) &&
(COUNT_TIME < REF_TIME))
    {
        if( ONLY_FIRST != 1)
            {
                ONLY_FIRST = 1;
                _delay_us(6);
                OUT_D_X = 1;
                OUT_D_Y = 1;
                _delay_us(6);
            }
        _delay_us(3);
        OUT_S_X = 1;
        OUT_S_Y = 1;
        _delay_us(3);
        OUT_S_X = 0;
        OUT_S_Y = 0;
        COUNT_STEP_X--;
        COUNT_TIME = COUNT_TIME + 6;
    }
```

Kroky se na výstupu řídicí jednotky generují, pokud je referenční čas větší jako hodnota času v proměnné COUNT_TIME (COUNT_TIME < REF_TIME). Dále je nutné mít nastavený směr pohybu a počet kroků pro daný směr musí být větší nebo menší než 0. Směr DIR se nastavuje pouze před generováním prvního kroku. Pro další kroky zůstává stejný.

Část zdrojového kódu hlavní funkce „main()“ pro ukončení pohybu:

```
if((COUNT_STEP_X <= 0) && (MOVEMENT == 1) &&
(COUNT_TIME < REF_TIME))
{
    COUNT_TIME = REF_TIME + 1;
}

if((COUNT_STEP_Y <= 0) && (MOVEMENT == 2) &&
(COUNT_TIME < REF_TIME))
{
    COUNT_TIME = REF_TIME + 1;
}

if((COUNT_STEP_X >= 0) && (MOVEMENT == 3) &&
(COUNT_TIME < REF_TIME))
{
    COUNT_TIME = REF_TIME + 1;
}

if((COUNT_STEP_Y >= 0) && (MOVEMENT == 4) &&
(COUNT_TIME < REF_TIME))
{
    COUNT_TIME = REF_TIME + 1;
}
```

Pokud nastane případ, že se v rámci jednoho pohybu vyprázdní zásobník kroků (COUNT_STEP_X nebo COUNT_STEP_Y) a COUNT_TIME je stále menší než REF_TIME, tak dojde k okamžitému ukončení pohybu.

```
if(COUNT_TIME > (REF_TIME - 1))
{
    COUNT_TIME = 0;
    MOVEMENT = 0;
    ONLY_FIRST = 0;
    COUNT_STEP_MAIN = 0;
}
```

Při překročení referenční doby trvání pohybu v jednom směru se vynulují všechny proměnné a opakuje se cyklus pro nastavení dalšího pohybu.

4.3 Verze firmwaru 2.0

V této verzi se na každé přerušení na vstupech STEP vytvoří příznak přerušení a zařadí se do fronty příznaků v paměti procesoru. Takto vytvořený příznak je vyvolán a zpracován hned nebo až na něj dojde řada, podle jeho priority.

Hlavní program „main()“ je prázdný. Veškerý zdrojový kód je napsán v těle přerušení na signál STEP osy X a na signál STEP osy Y.

Hlavní myšlenkou je pohyb manipulátoru s jedním zapnutým a druhým vypnutým motorem pro sdružené pohyby. Zapnutý motor se vždy pohybuje dvojnásobnou frekvencí a tím dochází k pohybu v obou osách zároveň.

Nastavení přerušení je stejné jako ve verzích 1.0 a 1.1.

Část zdrojového kódu pro jednoduché pohyby manipulátoru vodorovně :

```
//Přerušení po přivedení sestupné hrany na IN_S_X(PE3)
ISR (PORTE_INT0_vect)
{
    INTER_STEP_X = 1;

    if(((INTER_STEP_Y == 0) && (IN_D_X == 0) && (OUT_D_X == 0) && ...)
    // pohyb doprava
    {
        _delay_us(6);
        OUT_D_X = 1;
        OUT_D_Y = 1;
        _delay_us(3);
    }
    else if(((INTER_STEP_Y == 0) && (IN_D_X == 1) && (OUT_D_X == 1)
    && ...)
    // pohyb doleva
    {
        _delay_us(6);
        OUT_D_X = 0;
        OUT_D_Y = 0;
        _delay_us(3);
    }

    if(((INTER_STEP_Y == 0) && (IN_D_X == 0)) // pohyb doprava
    {
        _delay_us(3);
        OUT_S_X = 1;
        OUT_S_Y = 1;
        _delay_us(3);
        OUT_S_X = 0;
        OUT_S_Y = 0;
    }
}
```

```

else if ((INTER_STEP_Y == 0) && (IN_D_X == 1)) // pohyb doleva
{
    _delay_us(3);
    OUT_S_X = 1;
    OUT_S_Y = 1;
    _delay_us(3);
    OUT_S_X = 0;
    OUT_S_Y = 0;
}

```

Nejprve se nastaví směr pohybu motoru pouze pod podmínkou, že není nastaven stejně z předešlého přerušení. Směr se přepíná pouze jednou za celý pohyb manipulátoru, čímž je kladena nižší zátěž na driver.

Část zdrojového kódu pro nastavení směrů sdružených pohybů manipulátoru :

```

if ((INTER_STEP_X == 1) && (INTER_STEP_Y == 1))
{
    if (((IN_D_X == 0) && (IN_D_Y == 0) && (OUT_D_X == 1) && ...))
    // pohyb doprava-nahoru
    {
        _delay_us(6);
        OUT_D_X = 1;
        OUT_D_Y = 1;
        _delay_us(3);
    }
    else if (((IN_D_X == 1) && (IN_D_Y == 1) && (OUT_D_X == 1) && ...))
    // pohyb doleva - nahoru
    {
        _delay_us(6);
        OUT_D_X = 0;
        OUT_D_Y = 0;
        _delay_us(3);
    }
    ...
}

```

Signály DIR se nastaví stejně, aby se manipulátor mohl pohybovat pouze vpravo nebo vlevo.

Část zdrojového kódu pro generování kroků u sdružených pohybů manipulátoru :

```
if ((IN_D_X == 1) && (IN_D_Y == 1))           // pohyb doleva - dolů
    {
        _delay_us(3);
        OUT_S_X = 1;
        _delay_us(3);
        OUT_S_X = 0;
        _delay_us(3);
        OUT_S_X = 1;
        _delay_us(3);
        OUT_S_X = 0;
    }
else if ((IN_D_X == 0) && (IN_D_Y == 0))      // pohyb doprava - nahoru
    {
        _delay_us(3);
        OUT_S_X = 1;
        _delay_us(3);
        OUT_S_X = 0;
        _delay_us(3);
        OUT_S_X = 1;
        _delay_us(3);
        OUT_S_X = 0;
    }
...

```

Podle nastaveného směru pohybu se generují kroky pouze pro jeden motor, druhý motor se nehýbe. Generováním dvojnásobného počtu kroků na jeden motor je dosaženo šikmého pohybu nahoru nebo dolů. Pokud jsou tedy aktivní přerušení v obou osách zároveň, manipulátor vykoná pohyb pod úhlem 45 °.

4.4 Verze firmwaru 2.1

U verze 2.0 byly při testování zjištěny nedostatky v podobě ztráty nebo přidání kroků při přechodu mezi jednotlivými typy pohybů. Ve verzi 2.1 jsou tyto nedostatky odstraněny.

Hlavní program „main()“ obsahuje převážně zdrojový kód pro vyhodnocení ukončení sdruženého pohybu. Zbytek zdrojového kódu je napsán v těle přerušení na signál STEP osy X a na signál STEP osy Y.

Hlavní myšlenkou je pohyb manipulátoru s jedním zapnutým a druhým vypnutým motorem pro sdružené pohyby. Zapnutý motor se vždy pohybuje dvojnásobnou frekvencí a tím dochází k pohybu v obou osách zároveň. Jednotlivé typy pohybů se zapisují do pole, odkud je možné následně tyto pohyby vyčíst a určit tak zda při přechodu mezi nimi nedošlo ke ztrátě nebo přidání kroku. Podle toho se potom přidá nebo ubere krok, aby nedocházelo k najíždění manipulátoru do nežádoucích poloh. Všechny přechody mezi pohyby manipulátoru se ztrátou kroku jsou uvedeny v tabulce č.5 a 6.

Tato verze je doplněna o přepínání PNP a NPN logiky pomocí přepínače DIP. Nastavení logiky signálů se mění při inicializaci řídicí jednotky. Pro opětovné nastavení je nutný restart řídicí jednotky.

Nastavení přerušení je stejné jako v předchozích verzích.

Význam použitých proměnných pro verzi 2.1:

MOVEMENT

- proměnná, do které se zaznamenává poslední provedený směr pohybu
- pro všechny směry

MOV

- proměnná, do které se zaznamenává poslední provedený směr pohybu
- pro směry ve sdružených pohybech

ONLY_FIRST

- proměnná, která zajišťuje přidání nebo odebrání kroku pouze jednou v rámci jednoho sdruženého pohybu (pro přechod mezi libovolným a sdruženým pohybem)

GET_S_Y_UP

- pokud je hodnota této proměnné větší než 0, odebírá se krok navíc při pohybu v Y ose směr nahoru

GET_S_Y_DOWN

- pokud je hodnota této proměnné větší než 0, odebírá se krok navíc při pohybu v Y ose směr dolů

PUT_S_Y_UP

- pokud je hodnota této proměnné větší než 0, přidá se krok navíc při pohybu v Y ose směr nahoru

PUT_S_Y_DOWN

- pokud je hodnota této proměnné větší než 0, přidá se krok navíc při pohybu v Y ose směr dolů

ARRAY[3]

- pole, do kterého se ukládají tři poslední po sobě jdoucí pohyby

LL_S[2]

- pole, do kterého se ukládá nastavení NPN nebo PNP logiky signálů STEP při inicializaci řídicí jednotky
- podle uložených hodnot v tomto poli se nastavuje logická úroveň výstupních signálů

LL_D[2]

- pole, do kterého se ukládá nastavení NPN nebo PNP logiky signálů STEP při inicializaci řídicí jednotky
- podle uložených hodnot v tomto poli se nastavuje logická úroveň výstupních signálů

COUNT_TIME_X

- doba bez sdruženého pohybu v ose X
- přičítá se čas po cca 10 us
- pokud je hodnota větší jak 90000 us, došlo k ukončení sdruženého pohybu

COUNT_TIME_Y

- doba bez sdruženého pohybu v ose Y
- přičítá se čas po cca 10 us
- pokud je hodnota větší jak 90000 us, došlo k ukončení sdruženého pohybu

COUNT_X

- při přerušení v ose X se do proměnné zapíše 1
- v hlavním programu dojde k vynulování času **COUNT_TIME_X**

COUNT_Y

- při přerušení v ose Y se do proměnné zapíše 1
- v hlavním programu dojde k vynulování času **COUNT_TIME_Y**

Přepínání NPN a PNP logiky ve funkci volané při inicializaci řídicí jednotky:

```
void logic_level_init(void)
{
    if (PSW1 == 0)
        // přepnutí na PNP logiku po zapnutí/restartu
        {
            LL_S[0] = 1;
            LL_S[1] = 0;
            LL_D[0] = 1;
            LL_D[1] = 0;
        }
    else if (PSW1 == 1)
        // přepnutí na NPN logiku po zapnutí/restartu
        {
            LL_S[0] = 0;
            LL_S[1] = 1;
            LL_D[0] = 0;
            LL_D[1] = 1;
        }
}
```

Hodnoty z polí LL_S a LL_D jsou čteny v přerušeních pro signály STEP v obou osách.
Log.1 na výstupu při generování signálů STEP a DIR pro driver znamená NPN.
Log.0 na výstupu při generování signálů STEP a DIR pro driver znamená PNP.

Příklad zdrojového kódu pro DIR:

```
...
_delay_us(6);
OUT_D_X = LL_D[1];
OUT_D_Y = LL_D[0];
_delay_us(3);
...
```

Příklad zdrojového kódu pro STEP:

```
...
_delay_us(3);
OUT_S_X = LL_S[1];
OUT_S_Y = LL_S[1];
_delay_us(3);
OUT_S_X = LL_S[0];
OUT_S_Y = LL_S[0];
...
```

Příklad zdrojového kódu pro jeden pohyb:

```
if (INTER_STEP_X == 0 && IN_D_Y == 0)           // pohyb dolů
    {
        MOVEMENT = 4;
        if (GET_S_Y_DOWN == 0)
            {
                _delay_us(3);
                OUT_S_X = LL_S[1];
                OUT_S_Y = LL_S[1];
                _delay_us(3);
                OUT_S_X = LL_S[0];
                OUT_S_Y = LL_S[0];
                if (PUT_S_Y_DOWN > 0)
                    {
                        _delay_us(3);
                        OUT_S_X = LL_S[1];
                        OUT_S_Y = LL_S[1];
                        _delay_us(3);
                        OUT_S_X = LL_S[0];
                        OUT_S_Y = LL_S[0];
                        PUT_S_Y_DOWN--;
                    }
            }
        else if (GET_S_Y_DOWN > 0)
            {
                GET_S_Y_DOWN--;
            }
    }
```

Pokud je `INTER_STEP_X = 0`, jedná se o jednoduchý pohyb a vykoná se při log.0 na vstupu řídicí jednotky pro signál DIR v ose Y. V proměnné MOVEMENT dojde k zaznamenání aktuálního pohybu, který se následně v další části programu zapisuje do pole:

```
...
ARRAY[0] = ARRAY[1];
ARRAY[1] = ARRAY[2];
ARRAY[2] = MOVEMENT;
...
```

Při větší hodnotě jak 0 v proměnné GET_S_Y_DOWN je krok odebrán.

Při větší hodnotě jak 0 v proměnné PUT_S_Y_DOWN je krok přidán.

Po přidání nebo odebrání kroku dojde k přičtení nebo odečtení čísla 1 z příslušné proměnné, aby se korekce pro daný krok neprovedla znovu.

U generování kroků pro sdružené pohyby nedochází ke korekci kroků a index pohybu se kromě proměnné MOVEMENT zapisuje také do proměnné MOV.

Příklad zdrojového kódu pro vyhodnocení ztráty kroku v přechodech mezi pohyby:

```
if((ARRAY[0] == 1) && (ARRAY[1] == 0) && (ARRAY[2] == 4))
{
    PUT_S_Y_DOWN++;
}
```

Příklad zdrojového kódu v hlavní funkci programu „main()“:









```
_delay_us(10);
if (MOV != 0)
// pokud je aktivní sdružený pohyb, začne detekovat zastavení pohybu v jedné ose na víc
jak 90 ms
{
    if ( COUNT_X == 0)
    // pokud není pohyb v ose X
    {
        COUNT_TIME_X = COUNT_TIME_X + 10;
    // přičte 10 us - čas bez pohybu v rámci jednoho cyklu hlavního programu
    }
    else
    // pokud je aktivní pohyb v ose X
    {
        COUNT_TIME_X = 0;
    // vynuluje čas bez pohybu v ose X
        COUNT_X = 0;
    // vynuluje proměnnou pro indikaci pohybu v ose X
    }

    if (COUNT_X > 90000)
    // pokud je čas bez pohybu v ose X větší jak 90 ms, byl ukončen sdružený pohyb
    {
        MOV = 0;
    // vynuluje se proměnná s indexem sdruženého pohybu
        ONLY_FIRST = 0;
    // vynuluje se proměnná pro zajištění správné korekce kroků ve sdružených
    pohybech
    }









}
```

V hlavním programu se detekuje ukončení sdruženého pohybu. Sdružený pohyb je aktivní pouze při dvou po sobě jdoucích přerušeních. Jedno v ose Y a druhé v ose X. Po ukončení sdruženého pohybu je aktivní buď jedno, nebo žádné přerušení. Po cca 90 ms + případně po dobu trvání všech přerušení se vynulují patřičné proměnné pro nastavení nového sdruženého pohybu. Tím je ošetřeno, že dochází ke korekci kroků pouze jednou za sdružený pohyb při přechodu z libovolného předchozího pohybu na sdružený.

Tab. 3: Pohyby manipulátoru

Znázornění	Vstupy		Výstupy		MOVEMENT	Pohyb
	DIR_X	DIR_Y	DIR_X	DIR_Y		
	1	X	0	0	1	doprava
	0	X	1	1	2	doleva
	X	1	0	1	3	nahoru
	X	0	1	0	4	dolů
	1	1	0	0	5	vlevo-nahoru
	1	0	0	0	6	vlevo-dolů
	0	1	1	1	7	vpravo-nahoru
	0	0	1	1	8	vpravo-dolů

Tab. 4: Tabulka korekcí kroků mezi jednoduchými pohyby









Přechod	ARRAY[0]	ARRAY[1]	ARRAY[2]	Korekce
	1	0	4	Přidat krok v ose Y ↓
	1	0	3	Přidat krok v ose Y ↑
	2	0	4	Přidat krok v ose Y ↓
	2	0	3	Přidat krok v ose Y ↑
	4	8	1	Odebrat krok v ose Y ↓
	3	7	1	Odebrat krok v ose Y ↑
	3	5	2	Odebrat krok v ose Y ↓
	4	6	2	Odebrat krok v ose Y ↑

U prvních čtyř přechodů v tabulce č.4 dochází ke ztrátě kroku v ose Y. Ztráta je způsobena přechodem mezi přerušením pro osu X a následně přerušením pro osu Y. Tím je nastaven sdružený pohyb, který se nevykoná v rámci přerušení pro osu Y. Takový nevykonaný pohyb je zapsaný pod indexem 0.

Všechny sdružené pohyby se vykonávají v přerušení pro osu X.

U následujících čtyř pohybů v tabulce č.4 dochází k přidání kroku v ose Y. Přidání kroku je způsobeno přechodem mezi přerušením pro osu Y a následně přerušením pro osu X. Nežádoucí sdružený pohyb se vykoná v těle přerušení pro osu X.

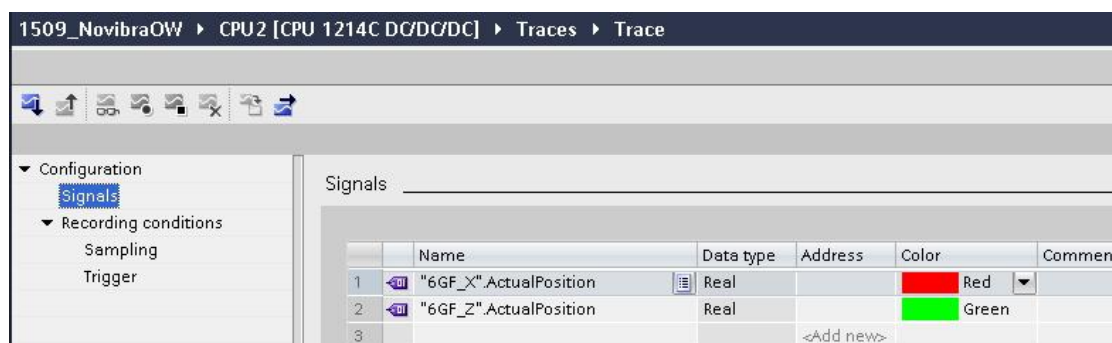
Tab. 5: Tabulka korekcí kroků mezi sdruženými pohyby

Přechod	ARRAY[0]	ARRAY[1]	ARRAY[2]	Popis
	1/2	0	7	Přidat krok v ose Y ↑
	1/2	0	8	Přidat krok v ose Y ↓
	1/2	0	5	Přidat krok v ose Y ↑
	1/2	0	6	Přidat krok v ose Y ↓
	3/4	7	7	Odebrat krok v ose Y ↑
	3/4	8	8	Odebrat krok v ose Y ↓
	3/4	5	5	Odebrat krok v ose Y ↑
	3/4	6	6	Odebrat krok v ose Y ↓

V tabulce č.5 jsou znázorněny přechody pro sdružené pohyby manipulátoru. Vznik chyby kroku je založen na stejném principu jako u jednoduchých pohybů v tabulce č.4.

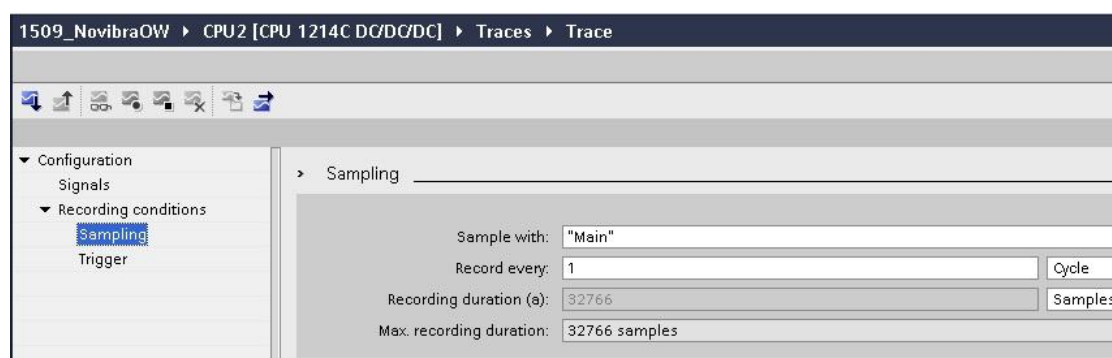
5 Testování

Jedná se o testování řídicí jednotky na kartézském dvouosém manipulátoru. Při testování bylo použito PLC S7-1200 s procesorem 1214C. Z PLC byly generovány signály STEP pro oba krokové motory s hodnotami 10 kHz, 30 kHz, 50 kHz, 70 kHz a 100 kHz. Řídicí modul signály zpracovával a posílal na vstupy driverů krokových motorů. Správná funkce řídicího modulu byla ověřována pomocí funkce „Traces“ v TIA portálu.



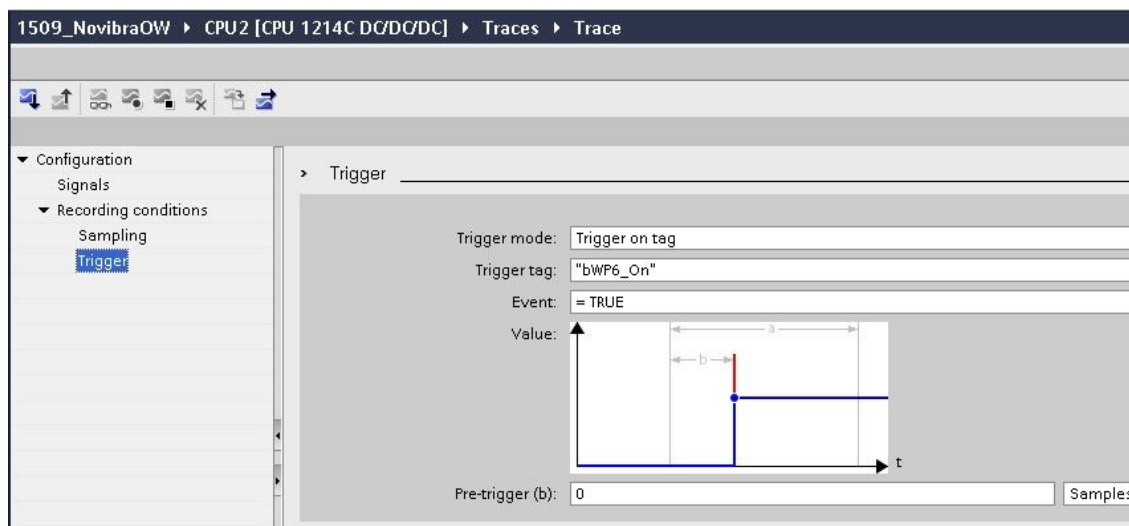
Obr. 26: Nastavení signálů pro vynášení do grafu

V první řadě bylo potřeba definovat signály, které funkce „Traces“ v TIA portálu vynese do grafu. Signály „6GF_X“. ActualPosition a „6GF_Z“. ActualPosition mají za úkol informovat o aktuální pozici manipulátoru. Každý signál představuje pohyb jednoho krokového motoru – jedné osy.



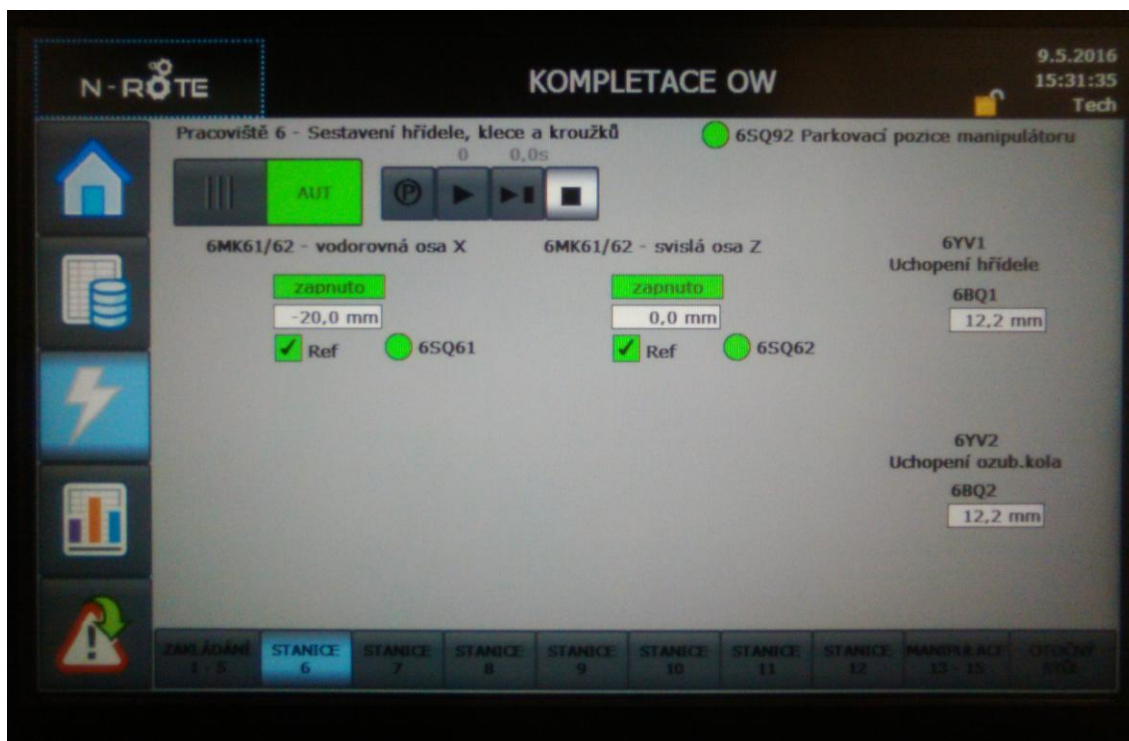
Obr. 27: Nastavení vzorkování

Následně bylo potřeba nastavit, který funkční blok na procesoru bude použit pro vzorkování. V bloku „Main“ je volána sekvence s programem pro manipulátor, na kterém byla tato diplomová práce testována. Dále bylo nastaveno čtení hodnot sledovaných signálů jedenkrát za cyklus procesoru a maximální počet vzorků 32766 pro jeden graf.



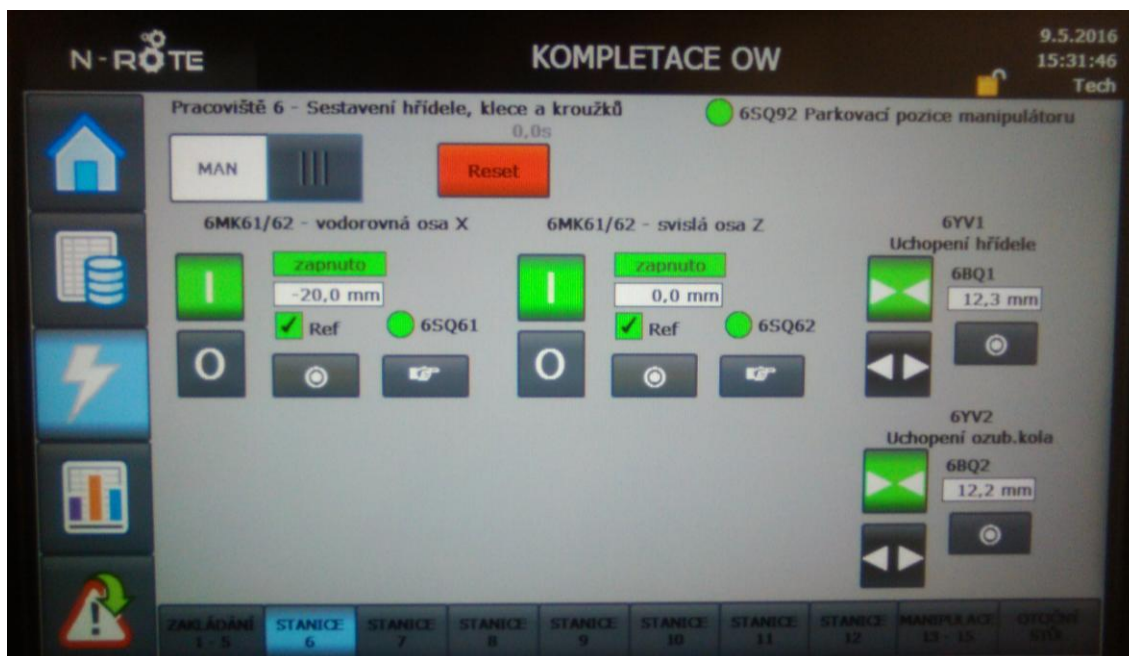
Obr. 28: Nastavení začátku měření

Všechny závislosti byly vykreslovány po nastavení proměnné „bWP6_On“ do log.1 (TRUE) z obrazovky uživatelského panelu pro ovládání pracoviště s kartézským manipulátorem.



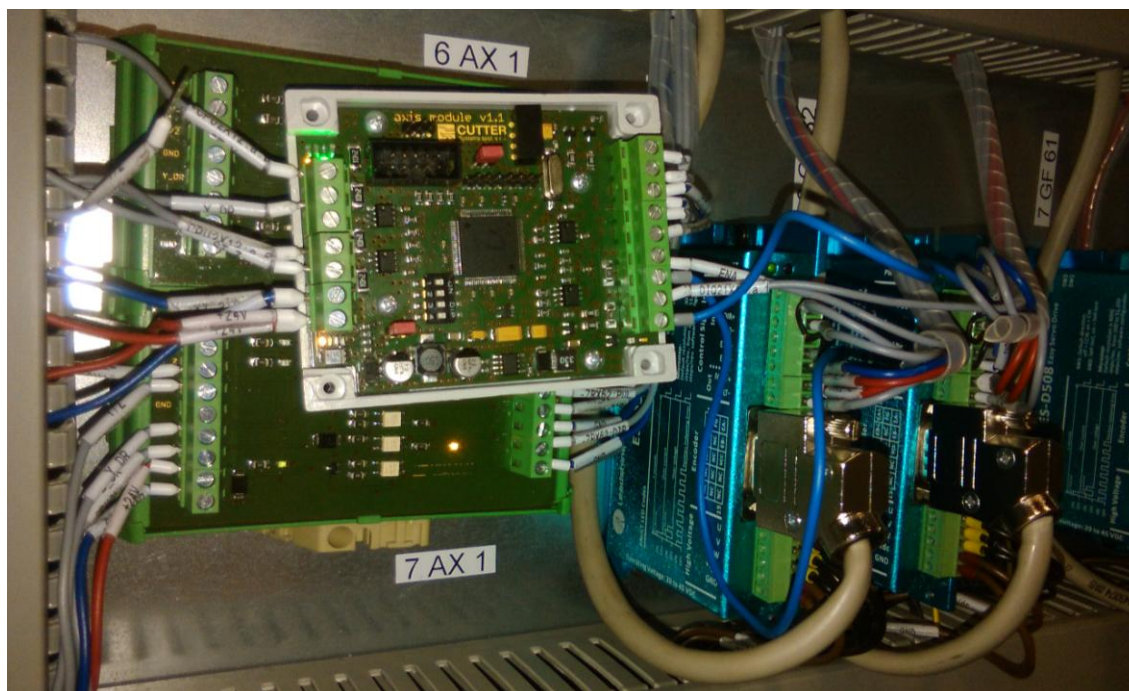
Obr. 29: Obrazovka pro ovládání manipulátoru v automatickém režimu na ovládacím panelu

Na obrázku č.29 můžeme vidět tlačítka pro ovládání manipulátoru v automatickém režimu: \textcircled{P} - návrat manipulátoru do parkovací polohy, \blacktriangleright - spuštění sekvence s manipulátorem, $\blacktriangleright\blacksquare$ - krokování sekvence, \blacksquare - zastavení cyklu s manipulátorem.



Obr. 30: Obrazovka pro ovládání manipulátoru v manuálním režimu na ovládacím panelu

Na obrázku č.30 můžeme vidět tlačítka pro ovládání manipulátoru v manuálním režimu: © – slouží k vyproštění manipulátoru z jakékoliv pozice při havárii, ☞ - zobrazí tabulku, kde je možné zadat souřadnice na které má manipulátor dojet, I – zapnutí motoru, O – vypnutí motoru.



Obr. 31: Testovací zapojení řídicího modulu v rozvaděči

5.1 Nastavení parametrů pro testování

Testování probíhalo pro frekvence 10 – 100 kHz řídicích signálů STEP z PLC. Pro tyto frekvence byly nastaveny správné hodnoty všech parametrů motoru. Tyto hodnoty pro nastavení motion control funkcí jsou uvedeny v tabulkách č.3 a č.4. Nastavování těchto parametrů je blíže popsáno v kapitole 1.3.

Tab. 6: Nastavené parametry motoru pro testování

Počet pulzů na otáčku motoru	4000 kroků
Posuv na otáčku motoru	59,879 mm

Z počtu pulzů a posuvu na otáčku motoru byly vypočítány rychlosti pro jednotlivé frekvence.

Tab. 7: Nastavené rychlosti pro testování

Frekvence [Hz]	Rychlost [mm/s]	Zrychlení [mm/s ²]
10000	150	200
30000	449	600
50000	748	1000
70000	1048	1400
100000	1497	2000

Část sekvence programu z PLC pro měření závislosti aktuální pozice na čase:

Nastavení rychlosti :

```
1: // Nastavení rychlostí krokových motorů  
"6MK_X_DB".rSpeed := 150.0;  
"6MK_Z_DB".rSpeed := 150.0;  
#bNext := true;
```

Pohyby ve vodorovné a svislé ose:

```
13: // Pohyb po ose X  
"r6MK_X_SP" := 180.0;  
"b6MK_X_Change" := true;  
#bNext := NOT "b6MK_X_Moving";  
14: // Pohyb po ose Y  
"r6MK_Z_SP" := -90.0;  
"b6MK_Z_Change" := true;  
#bNext := NOT "b6MK_Z_Moving";
```

Sdružený pohyb v obou osách:

```
13: // Sdružený pohyb  
"r6MK_X_SP" := 180.0;  
"r6MK_Z_SP" := -90.0;  
"b6MK_X_Change" := true;  
"b6MK_Z_Change" := true;  
#bNext := NOT "b6MK_X_Moving" AND NOT "b6MK_Z_Moving";
```

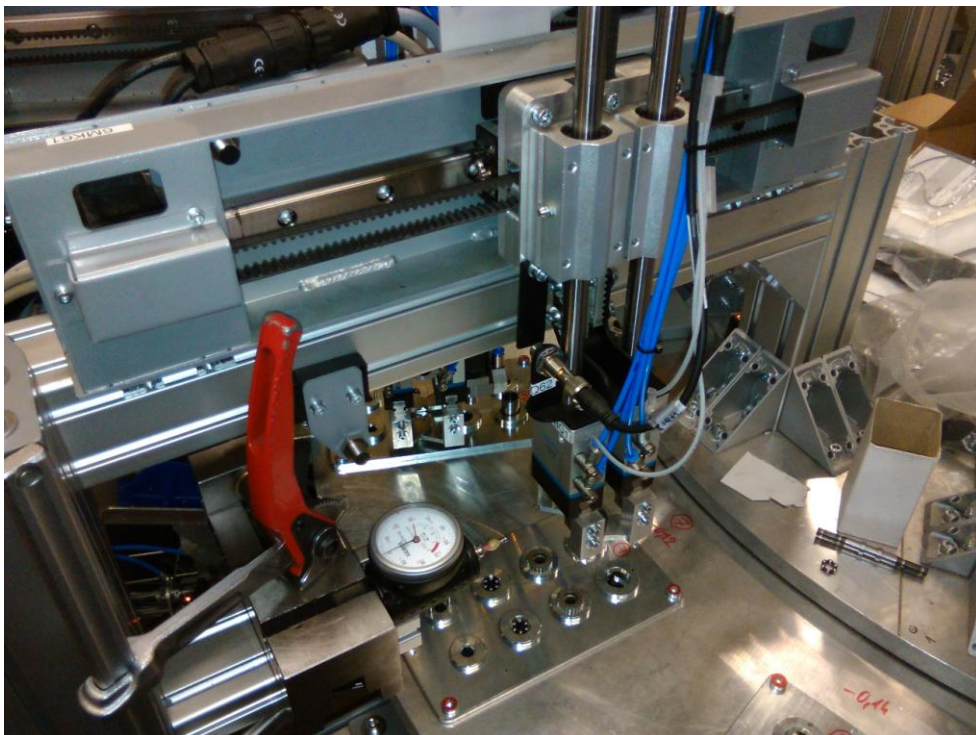
5.2 Kontrola přesnosti pohybu manipulátoru

3D-Sensor je velmi přesný a univerzální měřicí přístroj pro frézování a EDM stroje (izolované sondy). 3D-Sensor je upnut do svěráku a přichycen pevně ke konstrukci manipulátoru. Jeden dílek stupnice znázorňuje setinu milimetru. Může se přibližovat v libovolném směru (X-, Y-, Z-osa - odtud název "3D-Sensor"). Měřicí přístroj byl umístěn tak, aby při doteku manipulátoru stupnice ukazovala 25 setin milimetru. [9]

Testování probíhalo pro různé kombinace pohybů vždy několik desítek cyklů po sobě, aby bylo ověřeno, že manipulátor dojíždí na správnou pozici po ukončení cyklu programu v PLC. Manipulátor byl testován s verzí firmwaru 2.1.

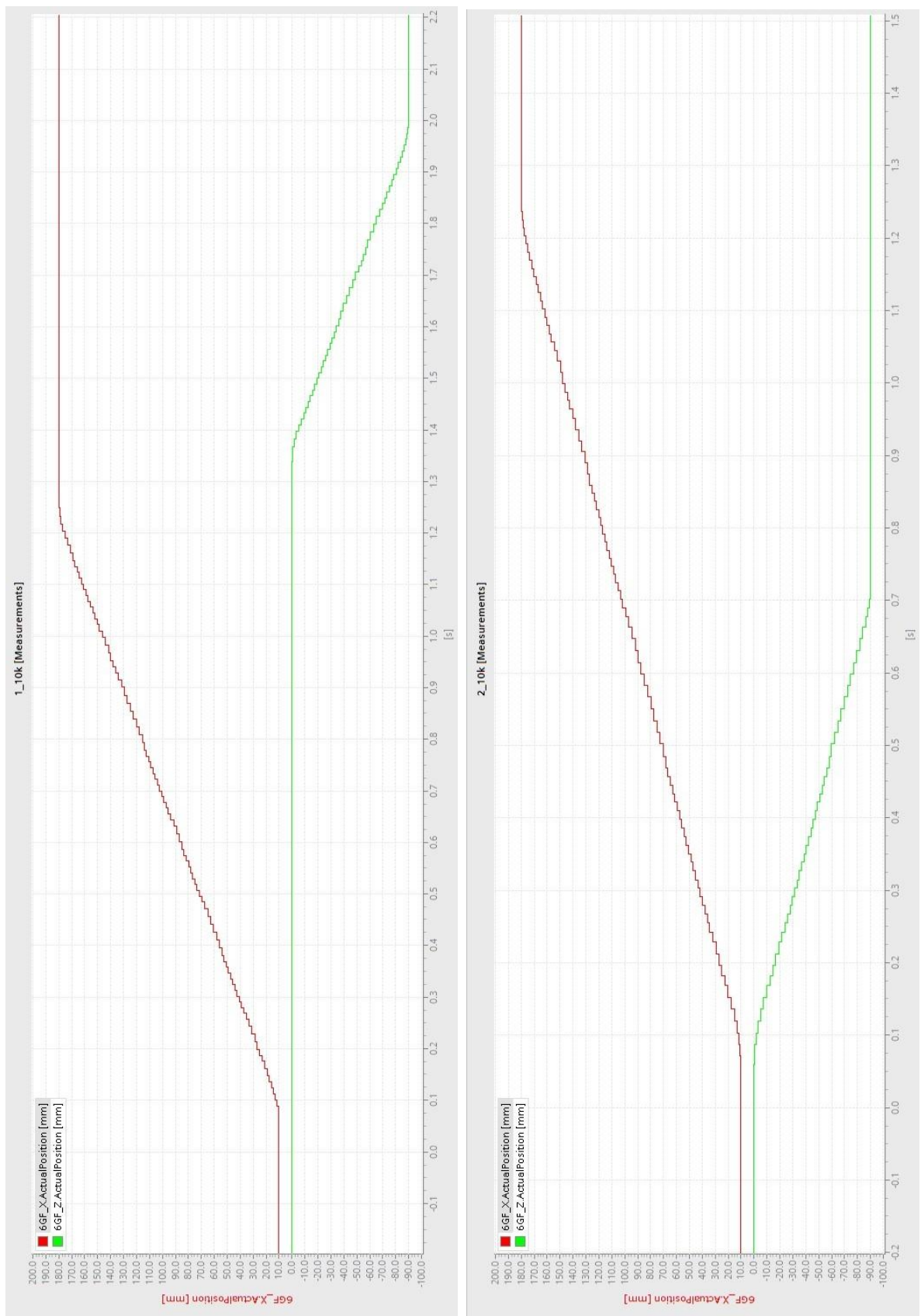


Obr. 32: 3D- Sensor

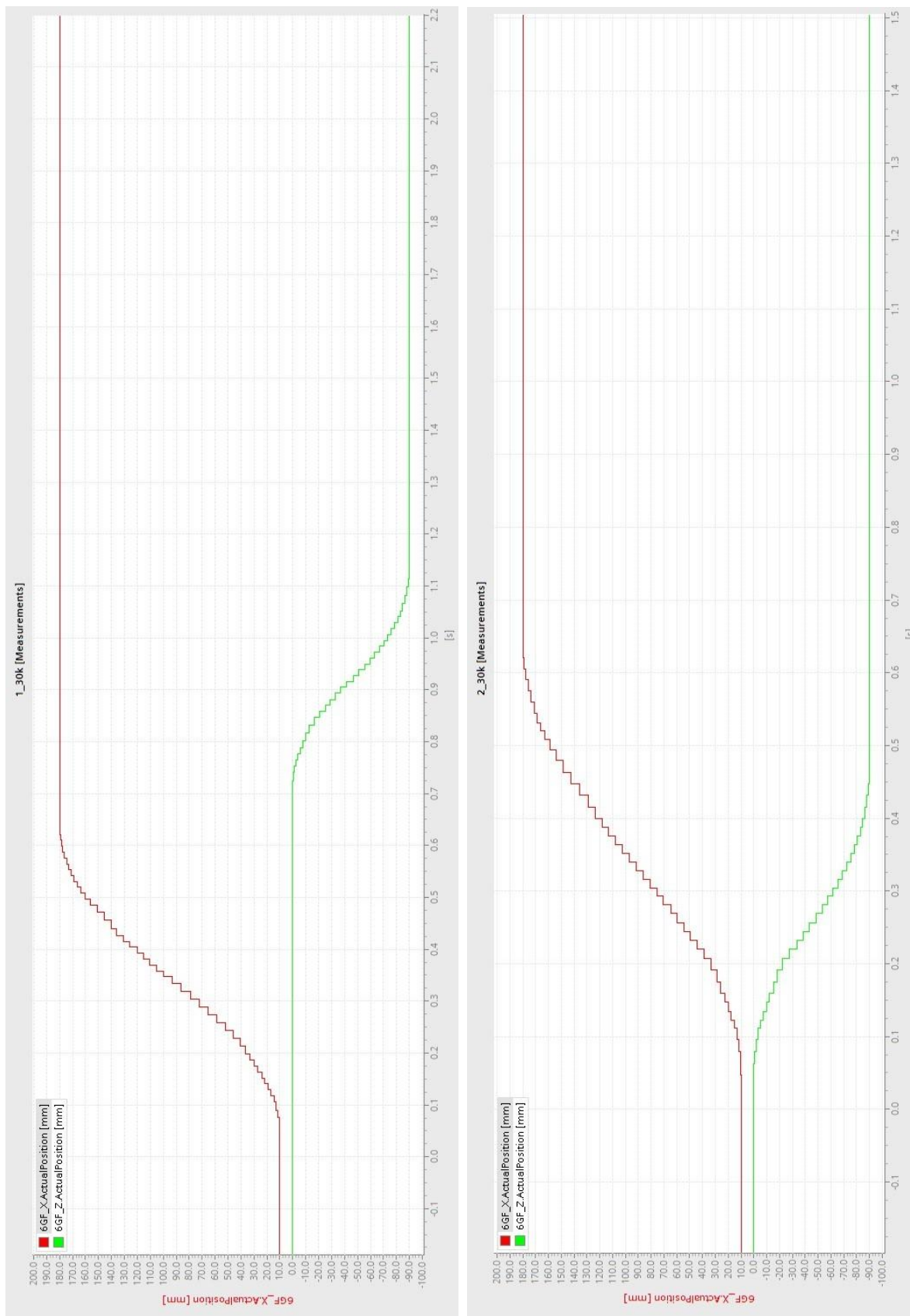


Obr. 33: Testovací pracoviště s manipulátorem

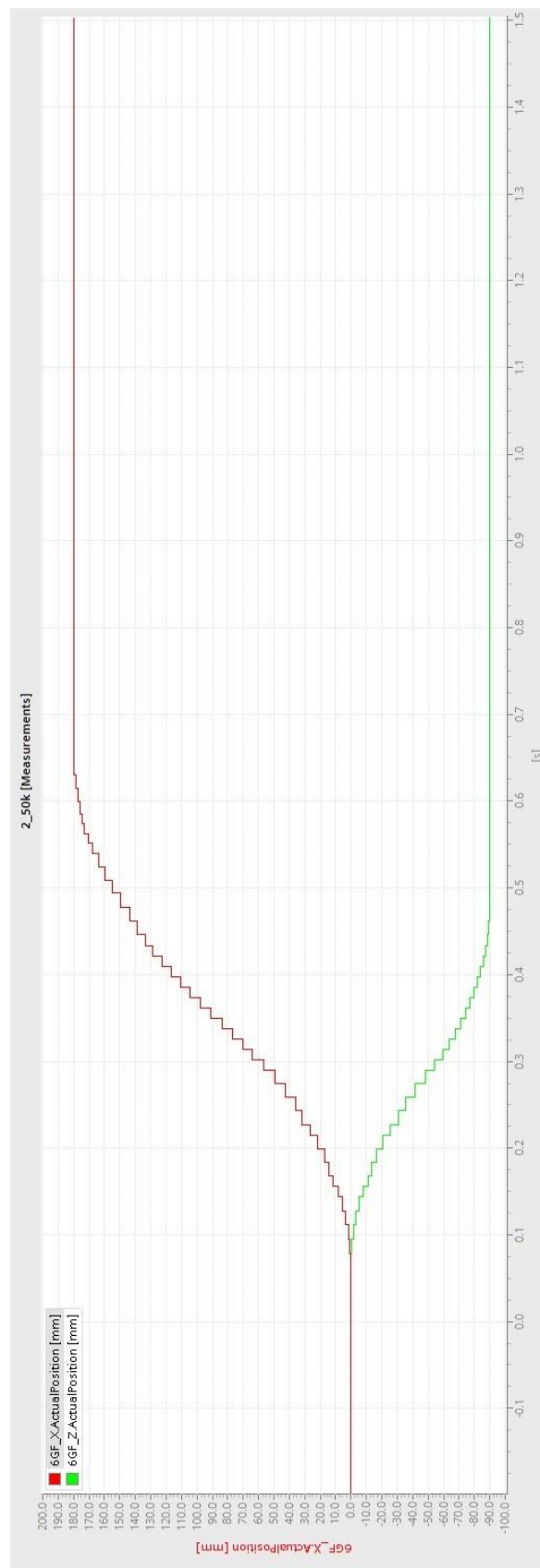
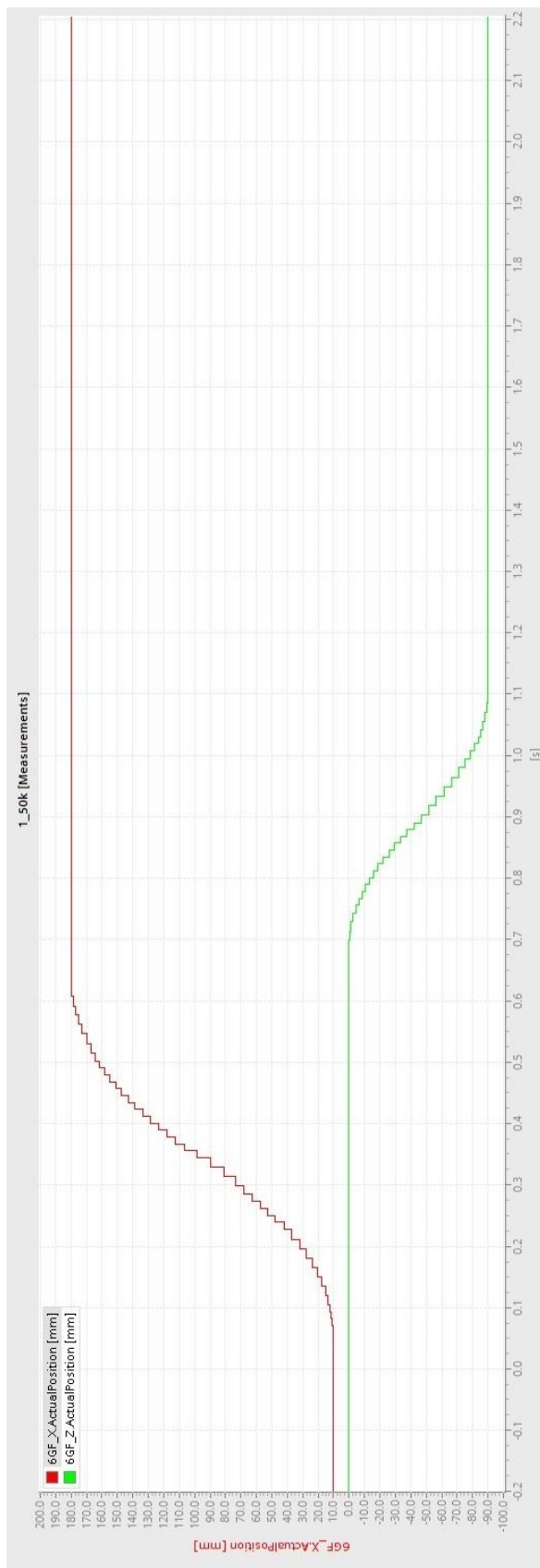
5.3 Změřené závislosti aktuálních pozic na čase



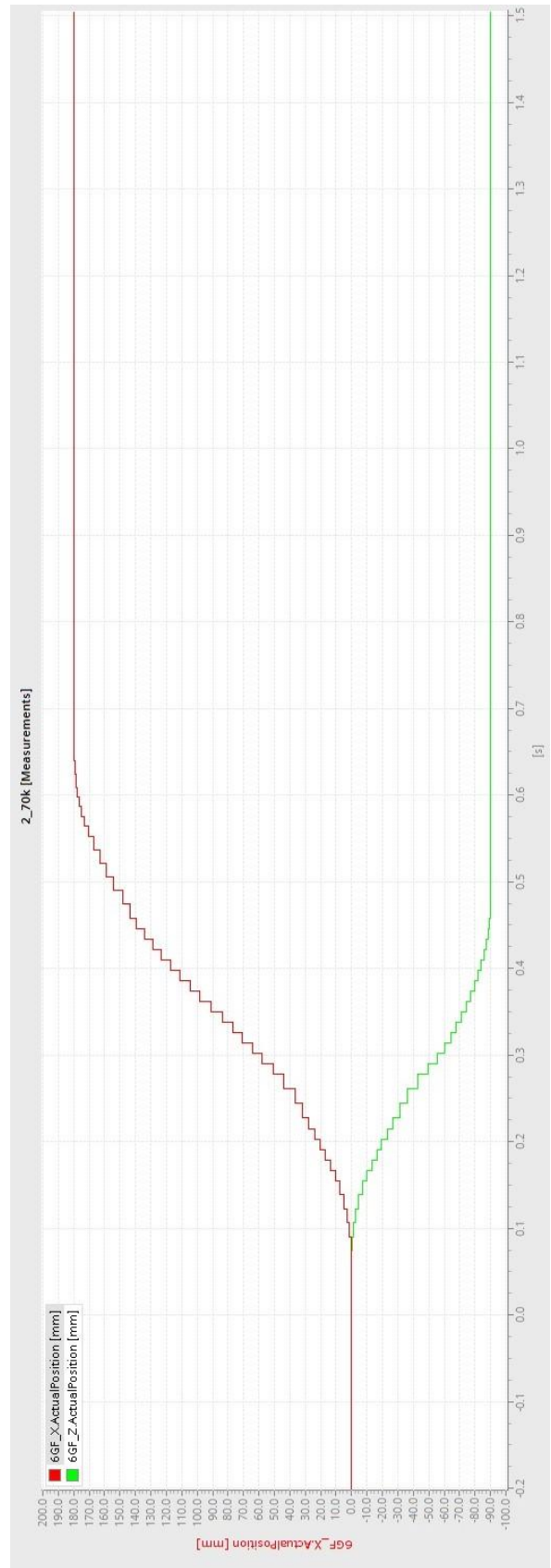
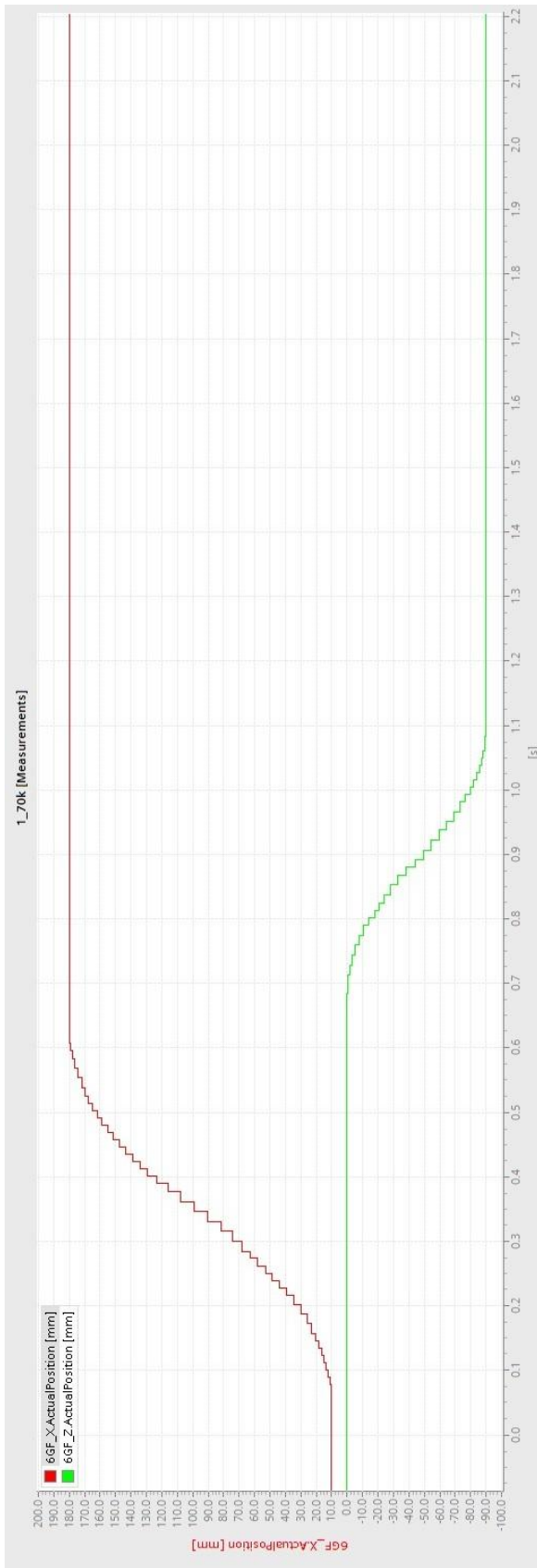
Obr. 34: Závislosti polohy manipulátoru na čase pro frekvenci 10 kHz (1_10k – jednoduchý pohyb, 2_10k – sdrúžený pohyb)



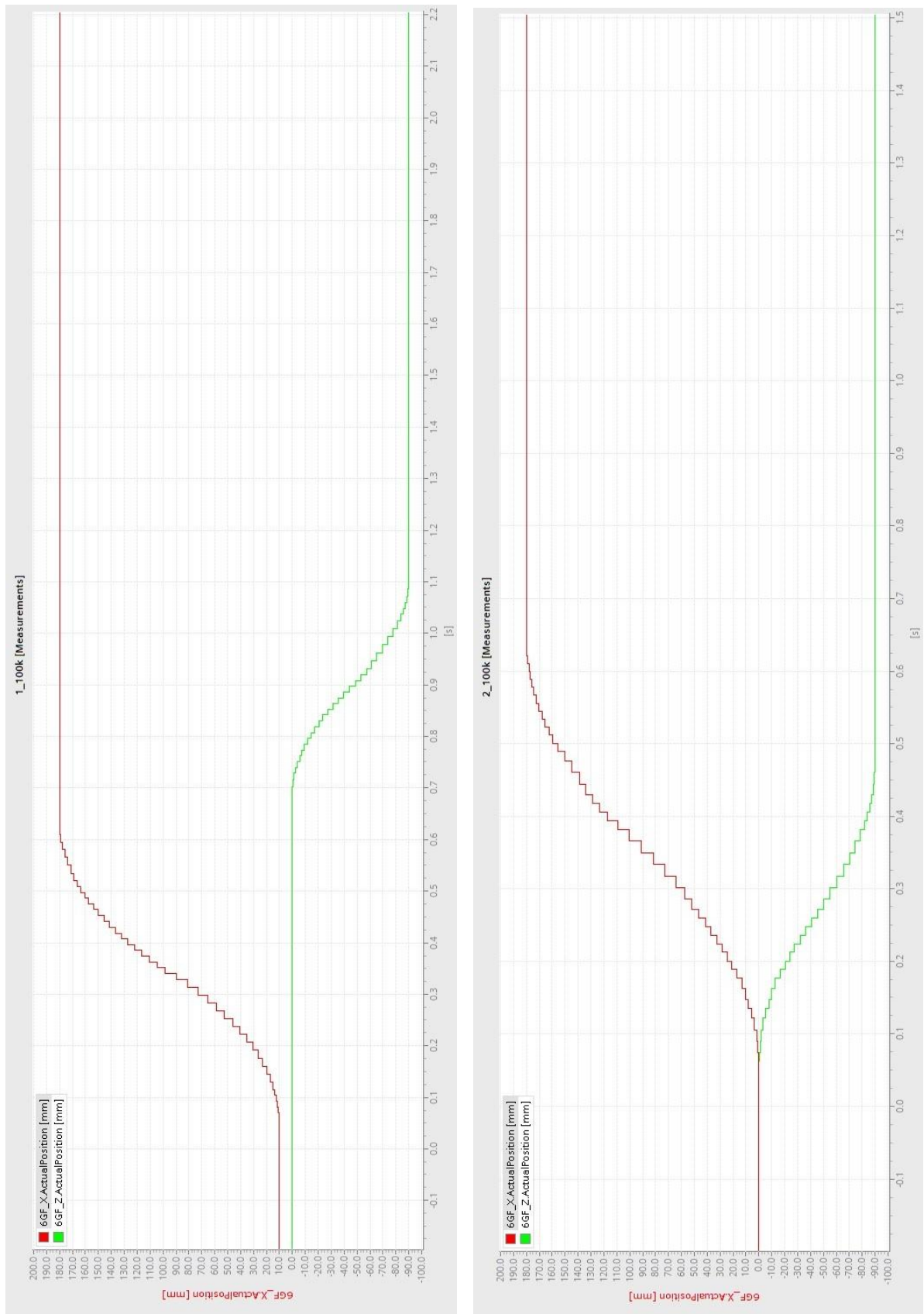
Obr. 35: Závislosti polohy manipulátoru na čase pro frekvenci 30 kHz (1_30k – jednoduchý pohyb, 2_30k – sdrúžený pohyb)



Obr. 36: Závislosti polohy manipulátoru na čase pro frekvenci 50 kHz
 (1_50k – jednoduchý pohyb, 2_50k – sdrúžený pohyb)



Obr. 37: Závislosti polohy manipulátoru na čase pro frekvenci 70 kHz (1_70k – jednoduchý pohyb, 2_70k – sružený pohyb)



Obr. 38: Závislosti polohy manipulátoru na čase pro frekvenci 10 kHz (1_70k – jednoduchý pohyb, 2_70k – sdružený pohyb)

Výše uvedené závislosti polohy manipulátoru na čase byly měřeny s verzí firmwaru 2.1 a řídicí jednotkou axis_module verze 1.1. Výstupní signály byly přivedeny na drivery krokových motorů ES-D508.

Měření bylo aktivní od okamžiku stisknutí tlačítka ► na ovládacím panelu, kterým se nastavila proměnná „bWP6_ON“ na hodnotu TRUE (log.1) a došlo tak ke spuštění sekvence pracoviště s manipulátorem.

Měření začíná v čase 0 s. Všechny naměřené hodnoty byly exportovány z TIA portálu a jsou na příloženém CD.

Ze závislostí je patrné, že při sdruženém pohybu manipulátoru dochází k úspoře času. Doba pohybu manipulátoru při sdruženém pohybu je rovna pouze době pohybu manipulátoru v ose, ve které ujede delší vzdálenost.

Drivery jsou schopny přijímat vstupní STEP signály s maximální frekvencí 200 kHz a zpracovávat je tak, aby na krokové motory posílaly signály s maximální frekvencí 51,2 kHz. Od frekvence kroků 51,2 kHz posílaných na vstupy driverů jsou proto závislosti poloh manipulátoru na čase stejné.

Nicméně ze závislostí vyplývá, že byly splněny cíle diplomové práce. Pokud by měl driver větší výstupní frekvenci, časy pohybů by se zkracovaly i pro vyšší frekvence signálů STEP.

6 Závěr

Cílem diplomové práce bylo seznámit se s problematikou řízení dvouosých manipulátorů pomocí řídicího systému PLC S7-1200. Na základě poznatků o řízení krokových motorů pomocí systému PLC byl navržen vlastní modul pro řízení dvouosého kartézského manipulátoru s frekvencemi řídicích signálů do 100 kHz, který využívá drivery krokových motorů řízené signálem STEP-DIR.

V první kapitole práce je popsán řídicí systém PLC S7-1200 spolu s vývojovým prostředím TIA portal od firmy Siemens. Následně je popsána tvorba programu v jazyce SCL a řízení krokových motorů pomocí motion control funkcí. Tuto kapitolu bylo nutné napsat kvůli vysvětlení řízení krokových motorů pomocí PLC. Tyto poznatky byly využity při návrhu řídicího modulu a při testování.

Druhá kapitola se zabývá srovnáním dostupných manipulátorů a vysvětlením funkce kartézského manipulátoru. Dále se zabývá drivery, kterými jsou řízeny krokové motory manipulátoru.

Třetí kapitola pojednává o návrhu konstrukce řídicího modulu a způsobu řízení. Modul bude využíván k řízení manipulátorů na výrobních linkách používaných v průmyslových halách. Z tohoto důvodu bylo nutné modul odrušit. Předpokládá se, že moduly budou kupovat i zákazníci z jiných firem, proto bylo potřeba zajistit jednoduché nastavování parametrů, odolnost proti přepólování napájení nebo signálů a odolnost proti přivedení vyššího napětí.

Galvanické oddělení vstupních a výstupních signálů je realizováno pomocí rychlých hradlových optočlenů a převodníků napětí DC/DC. Na desce se kvůli oddělení obvodů vyskytují tři galvanicky oddělené části – napájení modulu, vstupní a výstupní část. Jednoduché nastavování parametrů bylo realizováno přepínači DIP. Odolnost proti přepólování a přivedení vyššího napětí je řešena transily a zenerovými diodami. Testováním bylo zjištěno, že při přepólování všech vstupních i výstupních napětí nedojde k destruktivním průrazům. Modul je řízen mikroprocesorem Atmel ATXMEGA128, který byl zvolen kvůli velké frekvenci oscilátorů (až 32 MHz), dostatečnému počtu přerušení a časovačů.

Mezi další požadavky patří univerzální využití řídicího modulu pro různé drivery k motorům. Proto je možné volit výstupní napětí 5 nebo 24 V a PNP nebo NPN logiku podle použitého driveru. Rovněž bylo požadováno, aby obsluha mohla sledovat činnost modulu. Z tohoto důvodu jsou na modulu umístěné LED diody a v případě zájmu zákazníka i LCD displej.

Ve čtvrté kapitole se zabývám testováním různých verzí firmwaru. Všechny verze firmwaru využívají přerušení, aby nedocházelo ke ztrátě příchozích kroků z PLC.

Ve verzi 1.0 je vytvářen zásobník kroků, které se postupně v určitém počtu pro daný směr generují na výstupu. Manipulátor se pohybuje určitou dobu jedním směrem. Poté zůstane ve stejném směru nebo se přepne do jiného směru a pohybuje se určitou dobu dál. Tato verze nebyla dokonalá. Manipulátor nedojížděl na pozice.

U verze 1.1 byl pro každé přerušení generován jeden krok na výstupu s daným směrem. Pro jednoduché pohyby byla tato verze spolehlivá. U sdružených pohybů driver nebyl schopen dostatečně rychle přepínat směry a manipulátor již po dvaceti minutách testování dojížděl do stanovených pozic s chybou pět setin milimetru. Tato chyba je pro některé aplikace nepřijatelná.

Ve verzi 2.0 se přepíná směr pohybu pouze jednou za dobu trvání pohybu. Pro jednoduché pohyby se každý krok převádí ze vstupu rovnou na výstup se stejným směrem. U sdružených pohybů vždy jeden motor stojí a druhý se hýbe dvojnásobnou rychlostí. Tím je dosaženo pohybu manipulátoru pod úhlem 45° pro dva po sobě jdoucí kroky v různých osách. Při přechodech mezi jednotlivými pohyby se ztrácí nebo přibývají kroky.

Verze 2.1 má stejný základ jako 2.0 s tím rozdílem že je zde provedena korekce přidaného nebo odebraného kroku po přechodu mezi jednotlivými pohyby. Všechny přechody s nutností korekce kroku mezi jednotlivými pohyby jsou znázorněny v tabulkách č. 4 a 5. Jednotlivé indexy po sobě jdoucí pohyby jsou ukládány do pole v programu. Toto pole slouží k vyhodnocení a případné korekci kroku pro danou kombinaci pohybů. U této verze byl problém s určením konce sdruženého pohybu. Tento problém byl odstraněn. Indexy sdružených pohybů se tak mohou správně zapisovat do pole pro vyhodnocování potřeby korekce kroku. Verze 2.1 je finální.

V páté kapitole je popis výsledků testování s verzí firmwaru 2.1 na kartézském dvouosém manipulátoru s driverem krokových motorů ES-D508. Dojezd manipulátoru byl testován úchylkoměrem (3D-Sensor). Testování proběhlo pro frekvence vstupních signálů STEP z PLC o frekvencích 10, 30, 50, 70 a 100 kHz. Závislosti polohy manipulátoru na čase jsou znázorněny na obrázcích č. 34 - 38. Naměřené hodnoty jsou uvedeny na příloženém CD. Do příloh je nebylo vhodné vložit, protože by zabíraly cca 30 stran formátu A4. Z uvedených závislostí je patrné že se čas pohybu manipulátoru pro sdružené pohyby zkracuje o čas pohybu v ose s kratší ujetou vzdáleností.

Výstupní frekvence driveru je 51,2 kHz a na vstup může být přivedeno až 200 kHz. Pokud tedy modul posílá na driver signály STEP s frekvencí nad 51,2 kHz, tak je driver zpracuje a postupně posílá na krokové motory s frekvencí 51,2 kHz. Z tohoto důvodu jsou naměřené závislosti pro frekvence nad 50 kHz u jednoduchých pohybů stejné.

Při sdružených pohybech je na výstupu modulu frekvence dvojnásobná než na vstupu, proto jsou závislosti stejné již od 25 kHz. I přes toto omezení dochází k výrazné úspoře času pohybu manipulátoru.

U verze modulu 1.1 bylo z ekonomických důvodů odstraněno přepínání výstupního napětí. Dále byly přidány konektory pro připojení enable signálů pro driver a k nim byly přidány rezistory 2k2 pro omezení proudu enable signálů. Kvůli způsobu používaného zapojení rozvaděčů ve firmě, pro kterou byl tento modul vyvíjen byly přidány konektory pro propojky, kterými je možné spojit země v případě, že vstupní a výstupní signál není potřeba galvanicky oddělovat.

Na další verzi desky plošných spojů budou zasouvací konektory a nižší řada mikrokontroléru s potřebnou frekvencí oscilátoru 32 MHz. Použití ATXMEGA128A1 se jeví prozatím jako zbytečné. Byl zvolen kvůli širokým možnostem využití, ale ukázalo se, že ho zatím není možné naplno uplatnit. Do budoucna je plánováno na desce plošných spojů provést EMC testy a na další verzi provést případné úpravy.

Všechny cíle diplomové práce byly splněny. Navržený modul byl otestován a uveden do výroby. V současné době je již používán v rozvaděčích pro řízení dvouosých kartézských manipulátorů.

Seznam použité literatury

- [1] KŘÍŽ, P. Vývoj generátoru kódu PLC pro bloky faceplate pro S7-1200, S7-1500, S7-300/400. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 52 s.
- [2] PRUŠA, L. Konstrukční návrh manipulátoru obrobků. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 84 s.
- [3] DRHA, Š. Řízení tříosého kartézského manipulátoru. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 39 s.
- [4] Leadshine. ES-D508 Datasheet.[online], [cit 10-5-2016]
URL: http://www.cncshop.cz/PDF/ES/ES-D508d_V1.2.pdf
- [5] Atmel Corporation. ATXMEGA128A1 Datasheet. [online], [cit 10-5-2016]
URL: <http://www.atmel.com/Images/doc8077.pdf>
- [6] Procesor ATxmega128A1. In: ok2jnj.ok2kld [online], [cit. 10-5-2016]
URL: <http://ok2jnj.ok2kld.cz/ok2jnj/2016/04/procesor-atxmega128a1u/>
- [7] Agilent Technologies. HCPL0630 Datasheet. [online], [cit. 10-5-2016]
URL: http://pdf.datasheetcatalog.com/datasheets/166/83213_DS.pdf
- [8] KRÁLÍČEK, M. Bezdrátový termostat pro řízení kotle: Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 64s., 12s příloh.
- [9] 3D-Sensor New Generation. In: ok2jnj.ok2kld [online], [cit. 10-5-2016]
URL: <http://www.haimer.biz/products/measuring-instruments/sensors/3d-sensor/3d-sensor-new-generation/3d-sensor-new-generation.html>

Seznam příloh

A Návrh zařízení

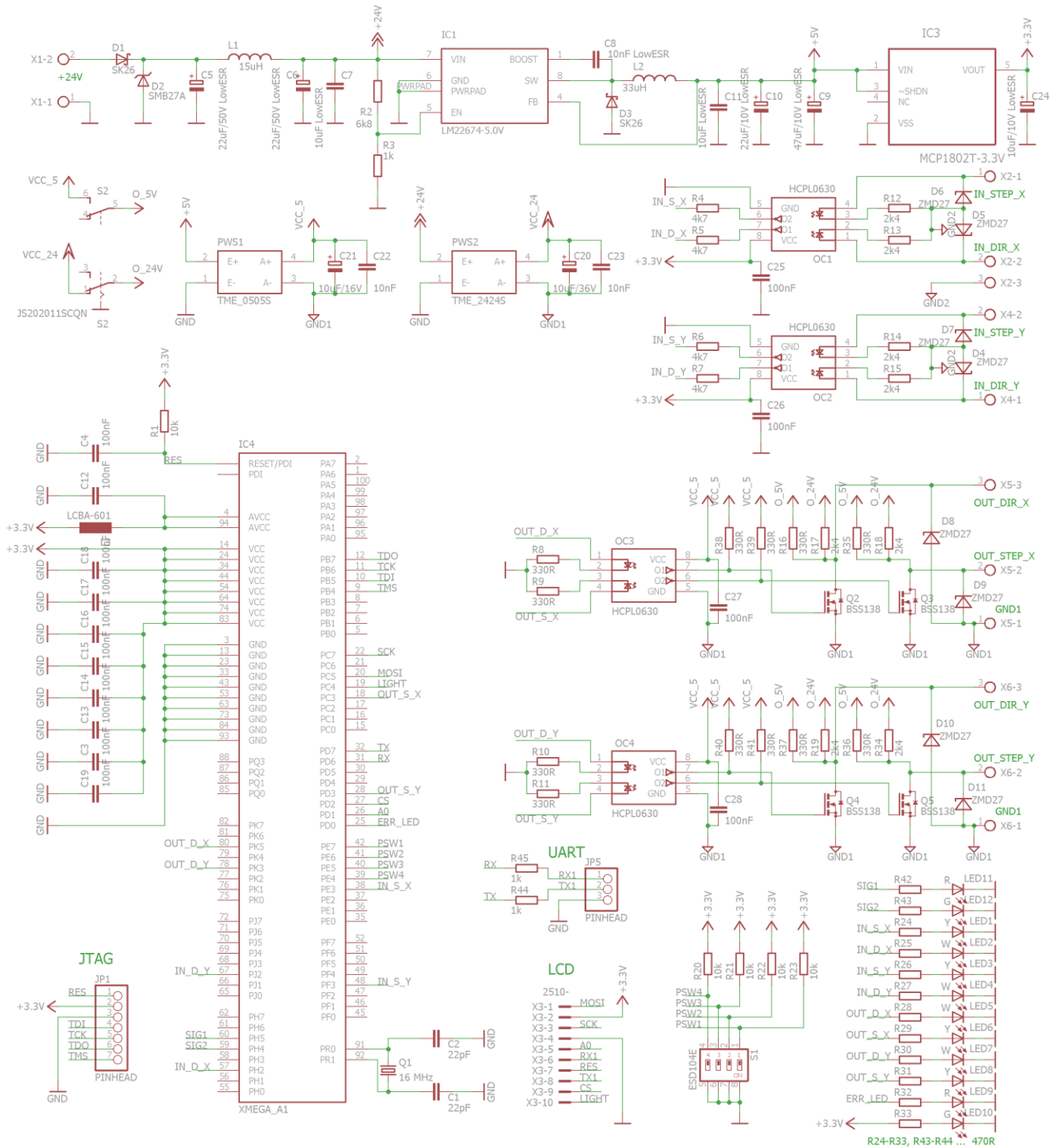
A.1	Schéma zapojení modulu pro řízení manipulátoru v 1.0.....	59
A.2	Deska plošného spoje modulu v 1.0 TOP	60
A.3	Deska plošného spoje modulu v 1.0 BOTTOM	61
A.4	Schéma zapojení modulu pro řízení manipulátoru v 1.1.....	62
A.5	Deska plošného spoje modulu v 1.1 TOP	63
A.6	Deska plošného spoje modulu v 1.1 BOTTOM	64
A.7	Schéma zapojení displeje.....	65
A.8	Deska plošného spoje displeje TOP	66
A.9	Deska plošného spoje displeje BOTTOM	67

B Seznam osazovacích plánů

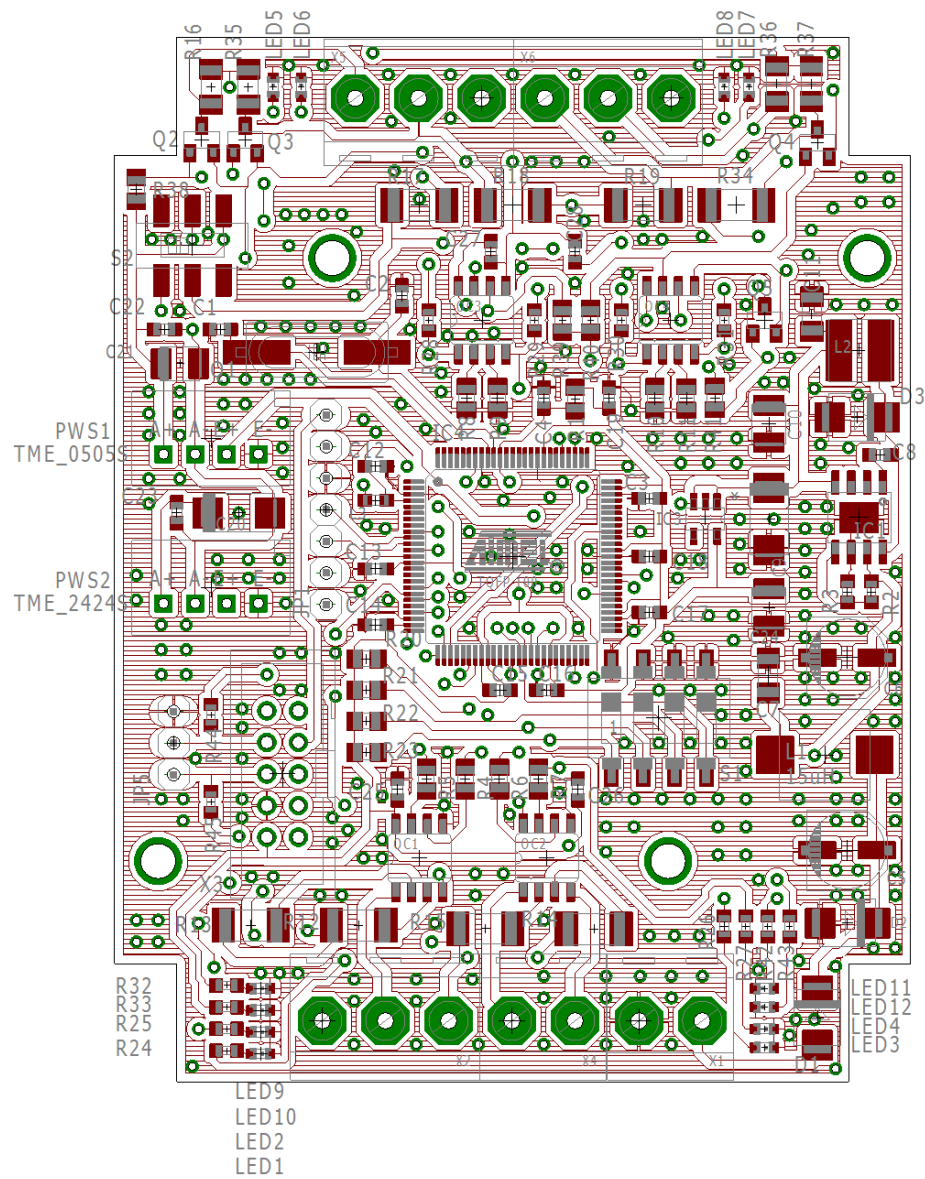
B.1	Osazovací plány pro desku plošného spoje modulu v 1.0	68
B.2	Osazovací plány pro desku plošného spoje modulu v 1.1.....	70
B.3	Osazovací plány pro desku plošného spoje displeje	72

C Obsah CD

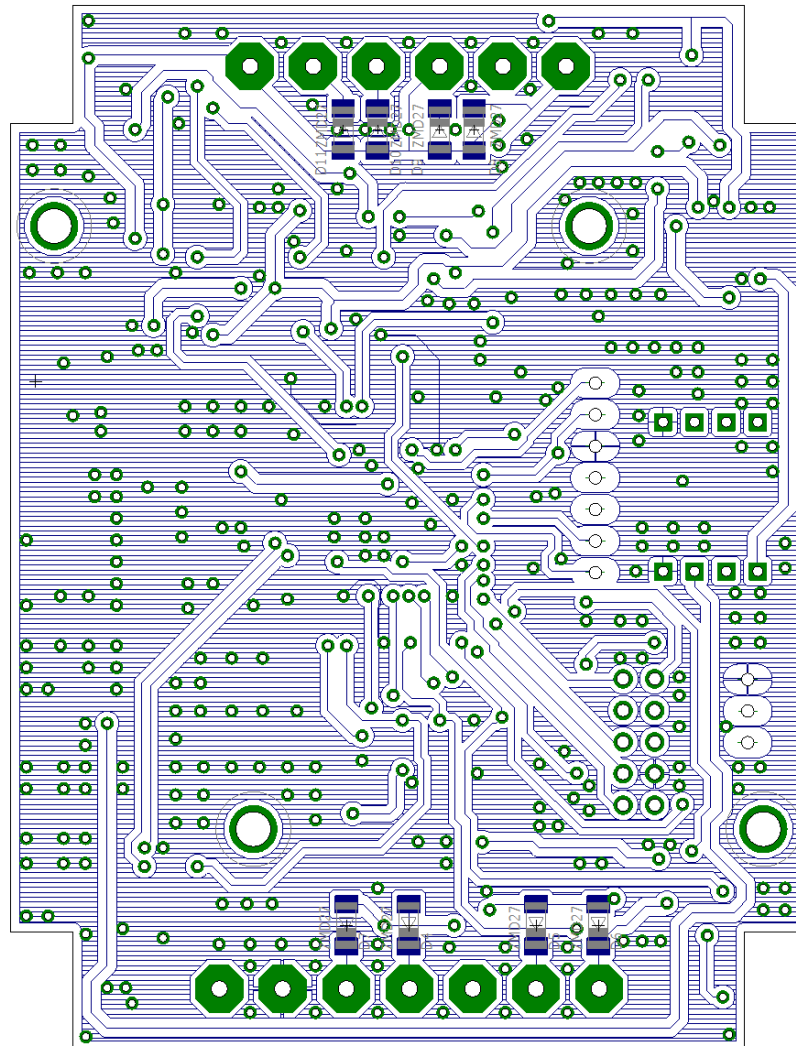
A.1 Schéma zapojení modulu pro řízení manipulátoru v 1.0



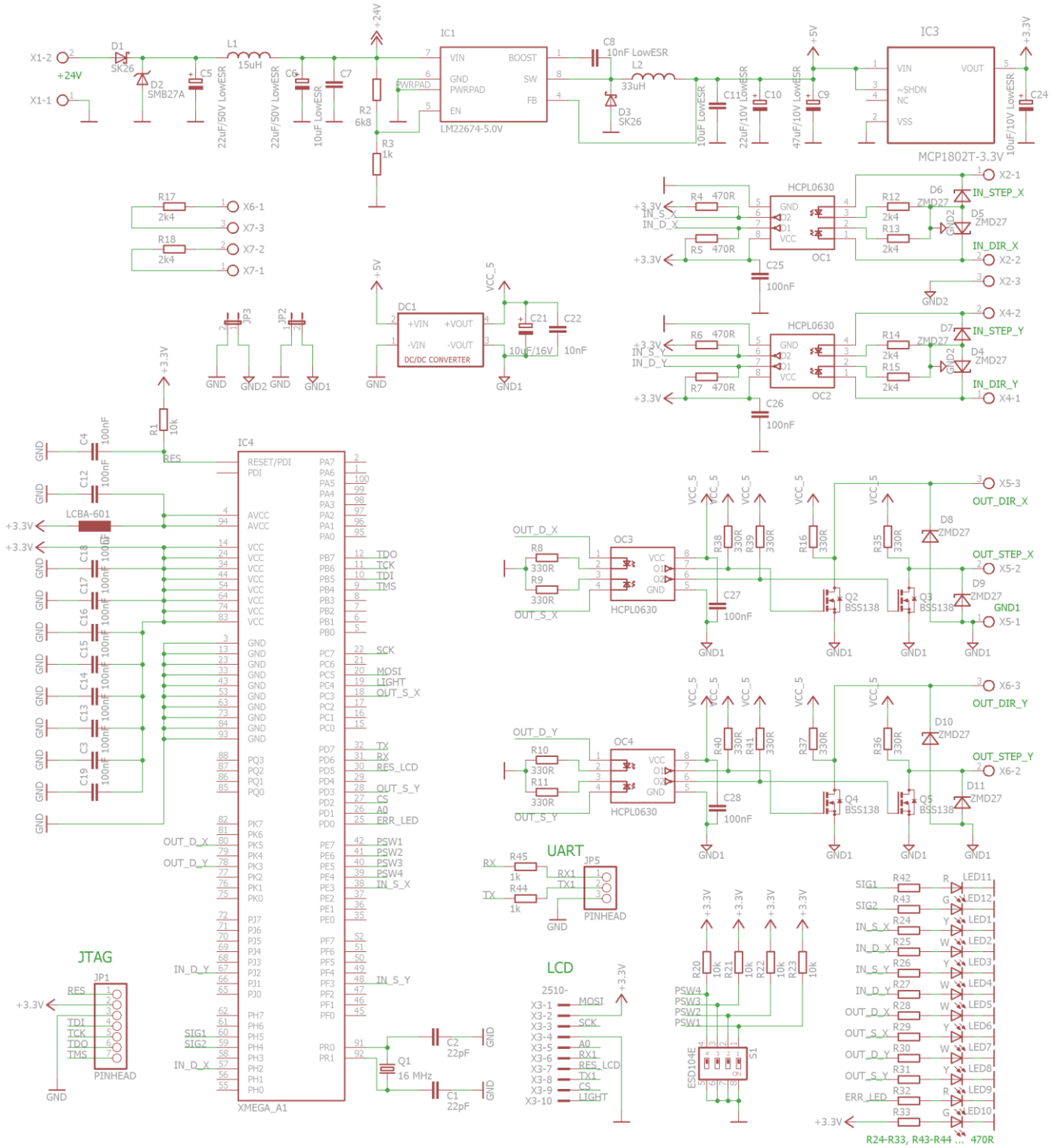
A.2 Deska plošného spoje modulu v 1.0 TOP



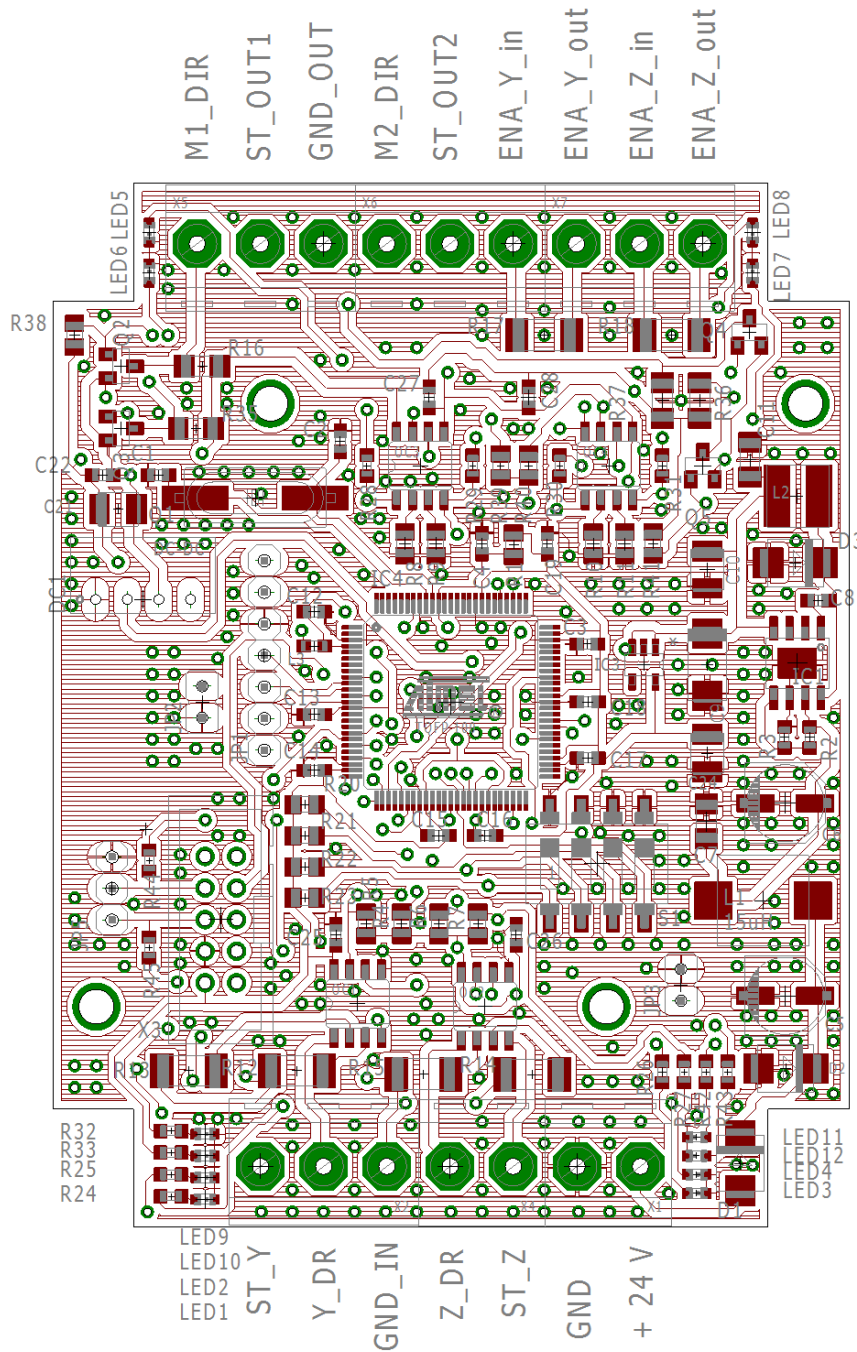
A.3 Deska plošného spoje modulu v 1.0 BOTTOM



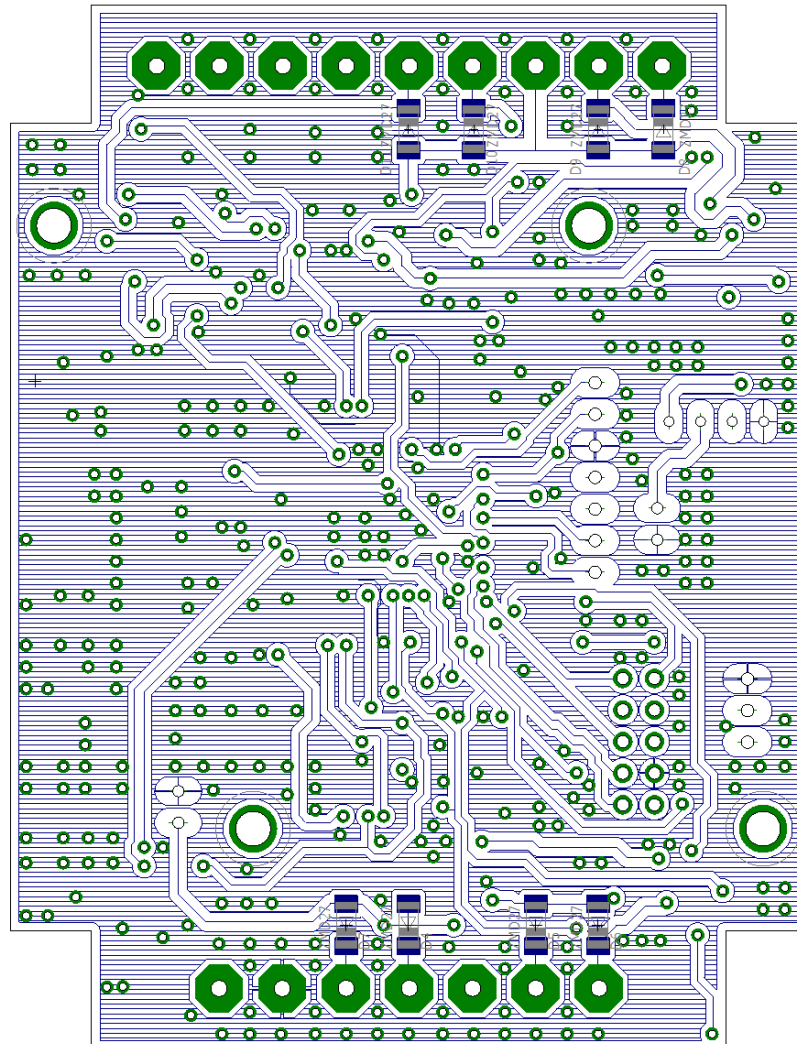
A.4 Schéma zapojení modulu pro řízení manipulátoru v 1.1



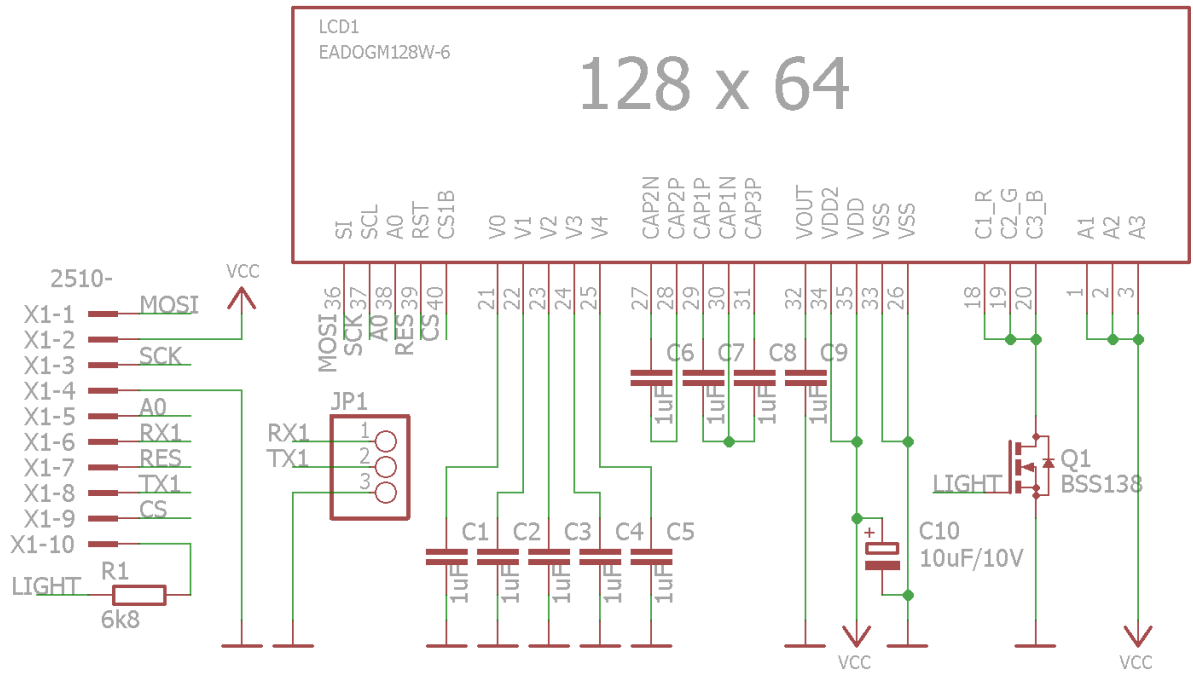
A.5 Deska plošného spoje modulu v 1.1 TOP



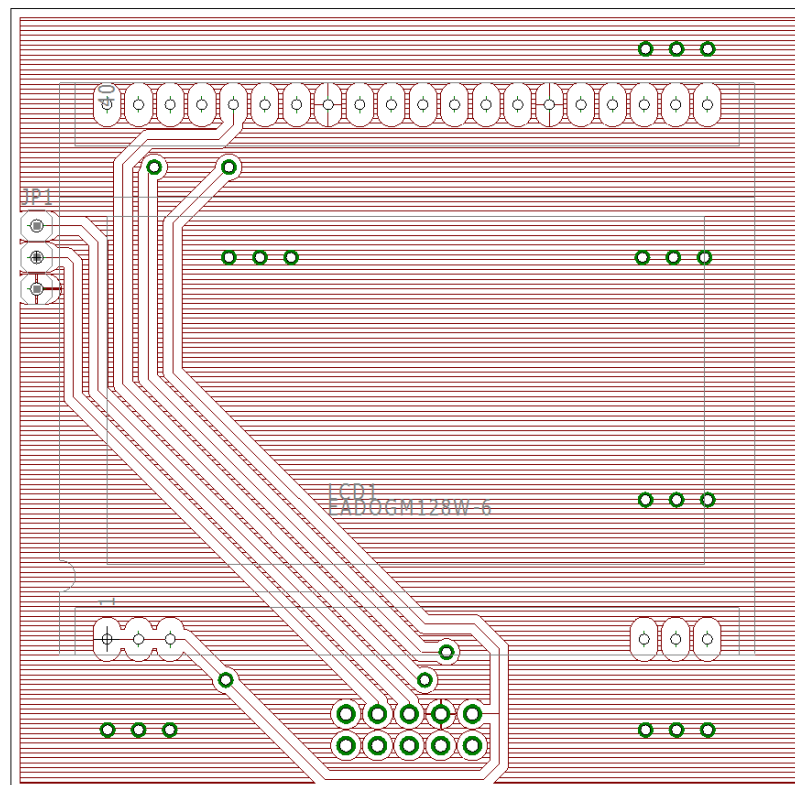
A.6 Deska plošného spoje modulu v 1.1 BOTTOM



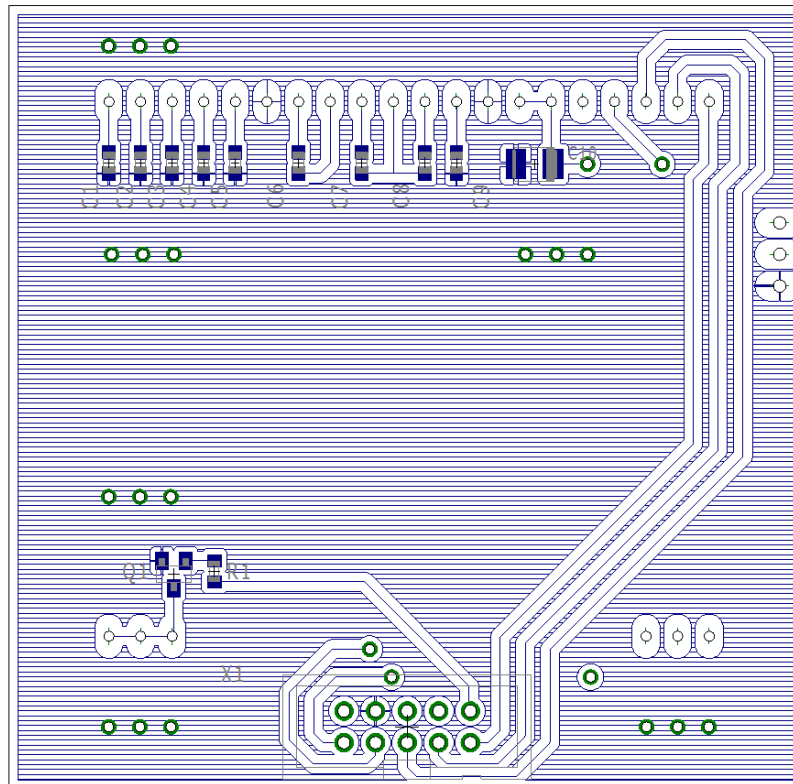
A.7 Schéma zapojení displeje



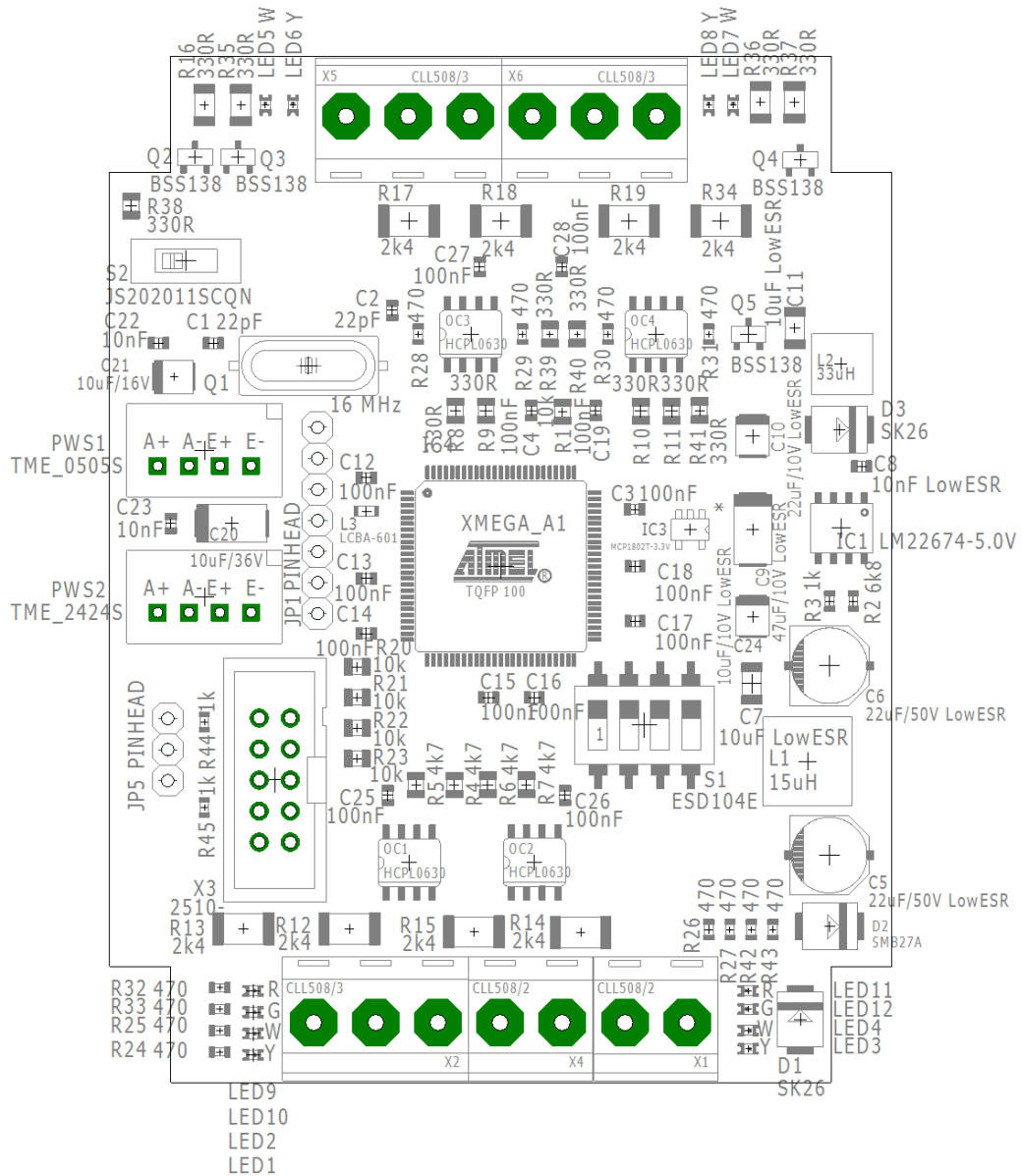
A.8 Deska plošného spoje displeje TOP

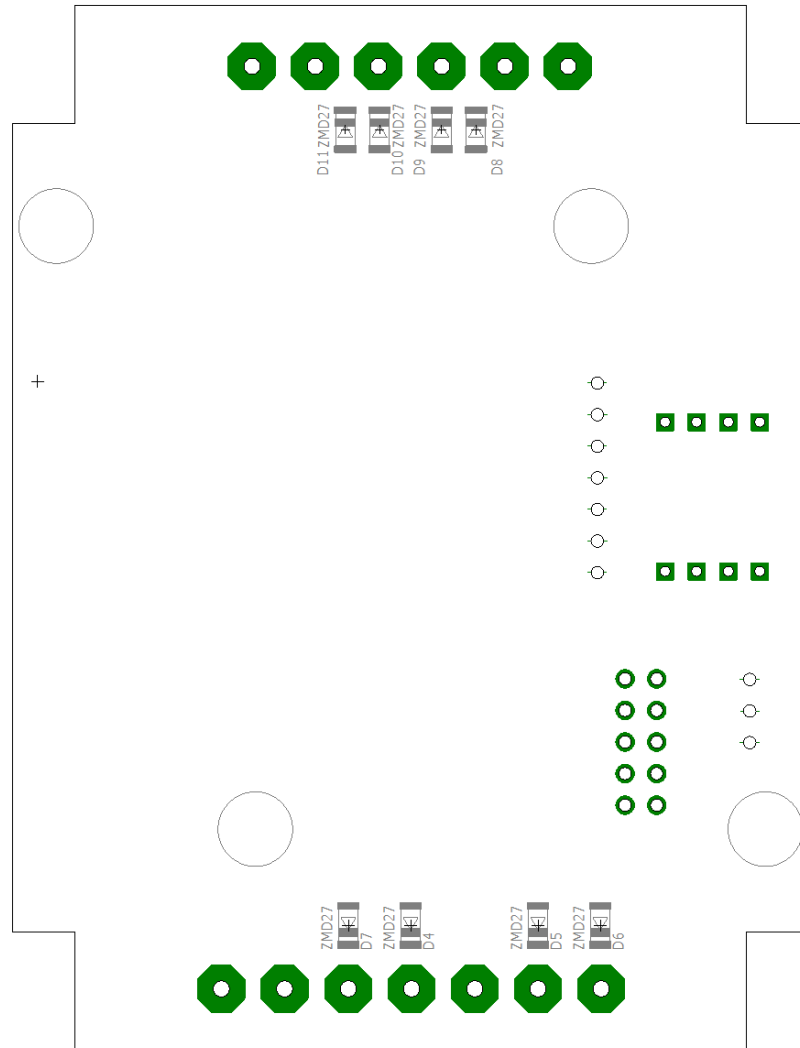


A.9 Deska plošného spoje displeje BOTTOM

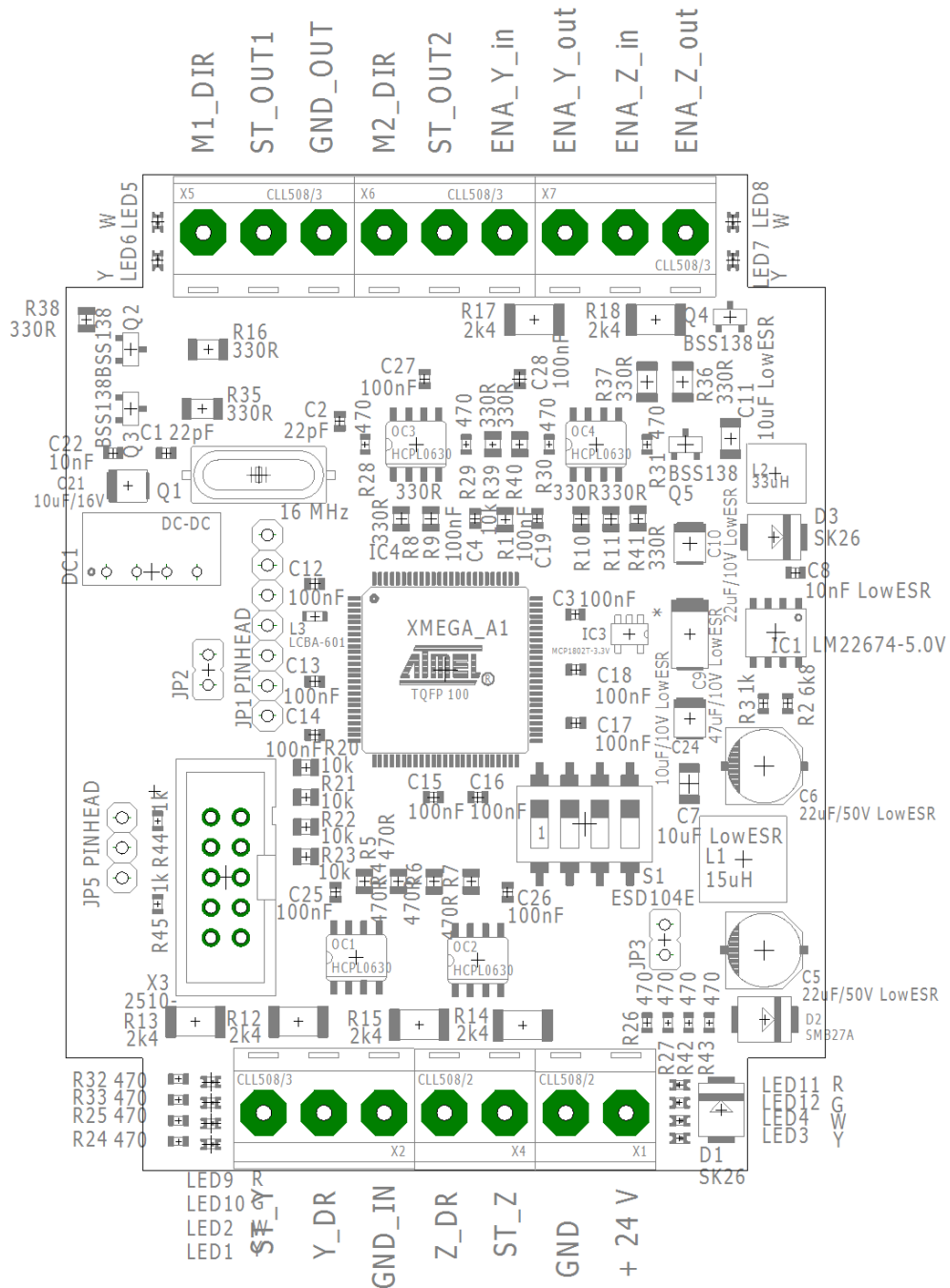


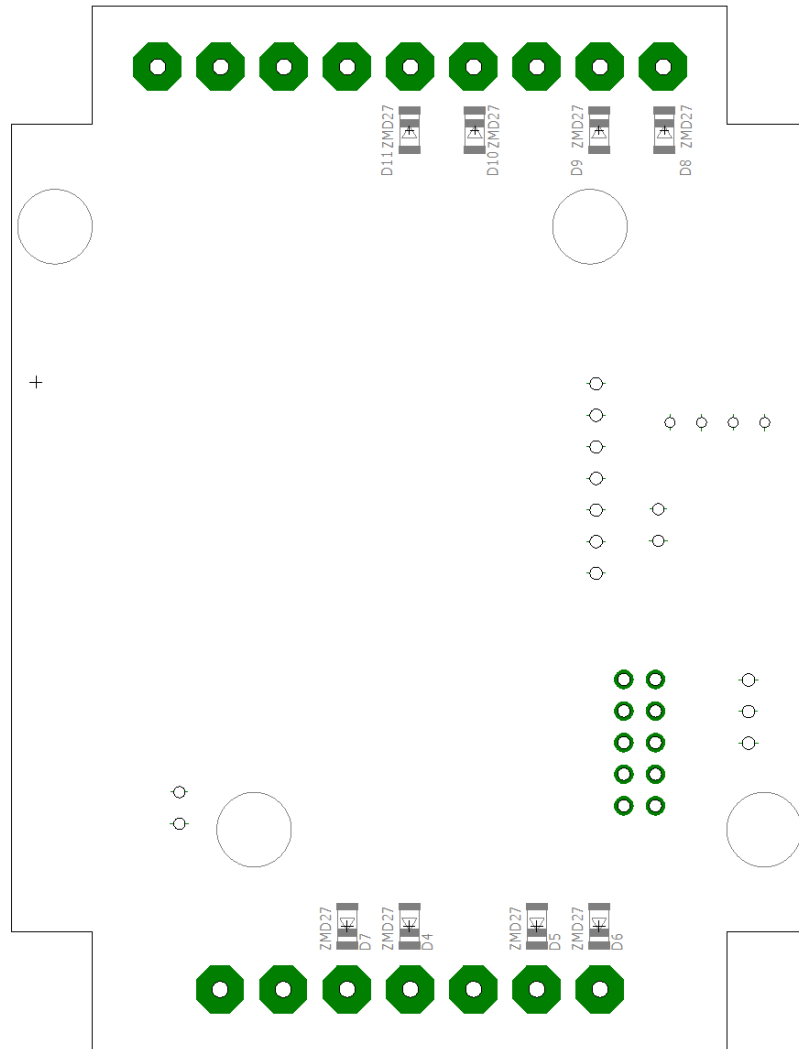
B.1 Osazovací plán pro desku plošného spoje modulu v 1.0



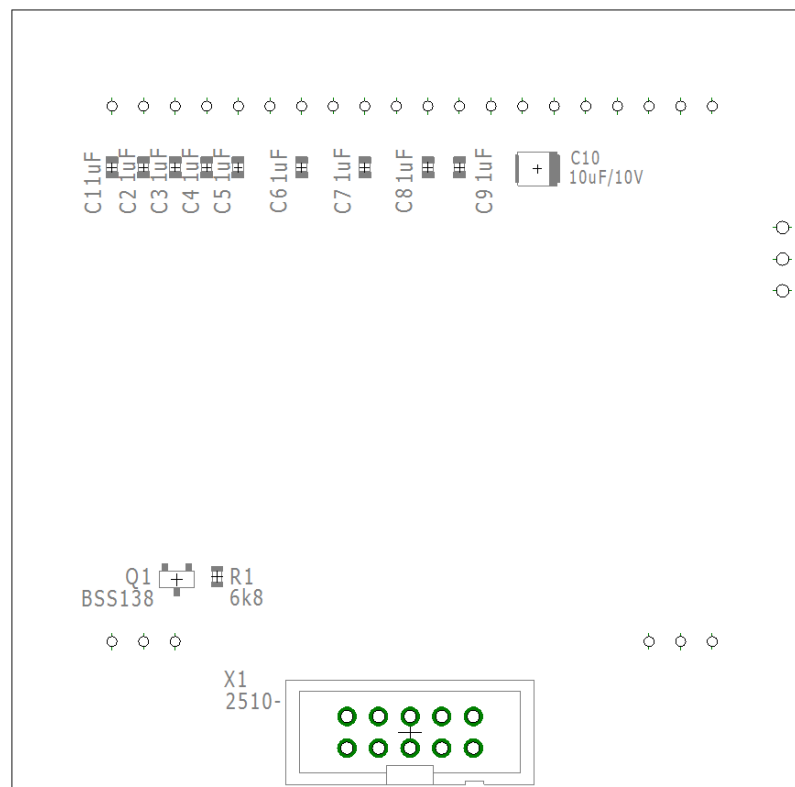
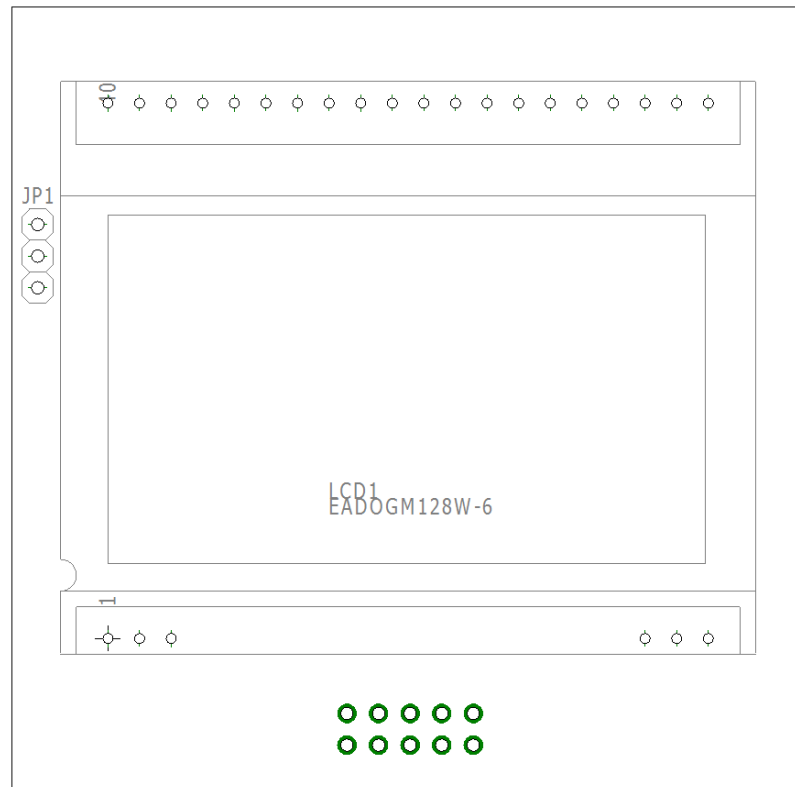


B.2 Osazovací plán pro desku plošného spoje modulu v 1.1





B.3 Osazovací plán pro desku plošného spoje displeje



C Obsah CD

/Diplomova_prace

/Dokument – text práce ve formátu PDF

/Eagle – schémata a DPS

/FW – zdrojový kód

/Katalog – katalogové listy součástek

/Naměřené hodnoty

/Foto – fotografie použité v DP