



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

DEPARTMENT OF COMPUTER SYSTEMS

**PLATFORMA PRO REALIZACI VÝPOČTU  
V KAPALNÉM KRYSTALU**

LIQUID CRYSTAL BASED COMPUTATIONAL PLATFORM

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**JAKUB KLÁZAR**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. MICHAL BIDLO, Ph.D.**

BRNO 2023

## Zadání diplomové práce



140775

Ústav: Ústav počítačových systémů (UPSY)  
Student: **Klášar Jakub, Bc.**  
Program: Informační technologie a umělá inteligence  
Specializace: Vestavěné systémy  
Název: **Platforma pro realizaci výpočtu v kapalném krystalu**  
Kategorie: Umělá inteligence  
Akademický rok: 2022/23

### Zadání:

1. Seznamte se s principy konceptu "Evolution-in-Materio" se zaměřením na realizaci funkcí v kapalných krystalech.
2. Na základě vybraného článku k problematice navrhnete a realizujete experimentální platformu, která umožní vhodným způsobem konfigurovat kapalně krystaly za účelem provádění vybraných elementárních operací.
3. Navrhnete a realizujete experiment, který pomocí vhodného přístupu umožní hledání konfigurací kapalného krystalu za účelem řešení vybrané úlohy. Provedte alespoň dvě sady experimentů s různými nastavením prohlédávacího algoritmu, porovnejte jeho schopnosti ve vztahu k řešení daného problému.
4. Zpracujte podrobnou dokumentaci technického řešení platformy pro možnost realizace vícero exemplářů navržené platformy.
5. Zhodnoťte dosažené výsledky a diskutujte možnosti pokračování projektu.

### Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:  
Splnění bodů 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Bidlo Michal, Ing., Ph.D.**  
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.  
Datum zadání: 1.11.2022  
Termín pro odevzdání: 17.5.2023  
Datum schválení: 31.10.2022

## Abstrakt

Tato práce zkoumá možnosti realizace výpočtu v kapalném krystalu, konkrétně se soustředí na realizaci logických operací. Text se věnuje návrhu, sestavení a používání platformy pro vykonávání experimentů nad výpočtem v kapalném krystalu. Jako platforma vzniklo zařízení, ke kterému je možné připojit LCD displej jako experimentální kapalný krystal a přes počítač skrz tuto platformu řídit experimenty, které využívají mimo jiné evoluční algoritmy. Dále se práce věnuje provádění experimentů, diskuzi a zpracování výsledků a následnému vyvození závěrů.

## Abstract

This work explores the possibilities of computations in a liquid crystal, specifically focusing on the logical operations. The text focuses on the design, build and use of a platform for performing experiments on computation in a liquid crystal. The platform is a device to which an LCD display can be connected as an experimental liquid crystal and experiments can be controlled from a computer through this platform. Experiments use, among other things, evolutionary algorithms. Further, the paper deals with the execution of experiments, discussion and processing of results and then creating conclusions of it.

## Klíčová slova

kapalný krystal, experimentální platforma, logické hradlo, výpočet in-materio, evoluční algoritmus, návrh DPS, ATmega328P, LCD displej

## Keywords

liquid crystal, experimental platform, logic gate, computation in-materio, evolutionary algorithm, PCB design, ATmega328P, LCD display

## Citace

KLÁZAR, Jakub. *Platforma pro realizaci výpočtu v kapalném krystalu*. Brno, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Bidlo, Ph.D.

# Platforma pro realizaci výpočtu v kapalném krystalu

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doktora Michala Bidla. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....  
Jakub Klázar  
4. června 2023

## Poděkování

V této části chci poděkovat především vedoucímu práce doktorovi Michalu Bidlovi za stvoření nápadu na tuto diplomovou práci, vstřícnost, odbornou pomoc a nevyčerpatelné nadšení pro věc. Dále si poděkování zaslouží Ing. Václav Šimek a Ing. Patrik Šimůnek za velmi cenné elektrikářské rady. A také samozřejmě děkuji rodině a všem kamarádům, kteří při mě ještě stále stojí.

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Výpočty v materiálech</b>	<b>4</b>
2.1 Vlastnosti materiálů vhodných pro výpočet . . . . .	4
2.2 Výpočet v DNA . . . . .	5
2.3 Výpočet v emulzi nanočásticí . . . . .	6
2.4 Výpočet v uspořádání nanotrubicí . . . . .	6
2.5 Výpočet v kapalném krystalu . . . . .	6
<b>3 Evolution in materio</b>	<b>7</b>
3.1 Simulovaná evoluce v materiálu . . . . .	7
3.2 Vlastnosti kapalného krystalu . . . . .	7
3.3 Realizace logických hradel v kapalném krystalu . . . . .	8
<b>4 Návrh experimentální platformy</b>	<b>10</b>
4.1 Prvotní prototyp . . . . .	10
4.2 Specifikace požadavků . . . . .	14
4.3 Blokové schéma . . . . .	16
4.4 Výběr elektronických součástek . . . . .	17
4.5 Elektronické schéma . . . . .	20
4.6 Deska tištěných spojů . . . . .	27
4.7 Firmware platformy . . . . .	29
4.8 Skripty experimentů . . . . .	37
<b>5 Experimentální výsledky</b>	<b>38</b>
5.1 Prvotní experimenty s prototypem . . . . .	38
5.2 Hledání hrubou silou . . . . .	39
5.3 Evoluční přístupy . . . . .	40
5.4 Diskuze výsledků . . . . .	40
<b>6 Závěr</b>	<b>41</b>
<b>Literatura</b>	<b>42</b>
<b>A Obsah přiloženého média</b>	<b>44</b>
<b>B Manuál použití platformy</b>	<b>45</b>
<b>C Podrobné schéma systému</b>	<b>46</b>

# Seznam obrázků

3.1	Schéma zapojení vodičů displeje k napěťovým vodičům. Inspirováno schématem z [6]. . . . .	8
3.2	Experimentální platforma pro výpočet v kapalném krystalu pana Hardinga a Millera. Převzato z [7]. . . . .	9
4.1	Blokové schéma zapojení prototypu platformy. . . . .	11
4.2	Schéma zapojení jednoho vodiče displeje v tranzistorovém poli a princip funkce při povolených kombinacích řídicích vodičů. . . . .	12
4.3	Prvotní prototyp platformy. . . . .	13
4.4	Vnitřní zapojení experimentálního LCD displeje. Převzato z [1] . . . . .	14
4.5	Blokové schéma výsledné platformy. . . . .	16
4.6	Část schématu výsledné platformy s MCU. . . . .	20
4.7	Část schématu výsledné platformy s napěťovými zdroji. . . . .	21
4.8	Část schématu výsledné platformy s převodníkem RS232. . . . .	22
4.9	Část schématu výsledné platformy s GPIO expandery. . . . .	23
4.10	Část schématu výsledné platformy s ADC. . . . .	24
4.11	Blokové schéma vnitřního zapojení DAC TLC7524C, převzato z [19]. . . . .	24
4.12	Část schématu výsledné platformy s DAC. . . . .	25
4.13	Zapojení GPIO expanderů, DAC, ADC a sběrnice I2C. . . . .	26
4.14	Pohled shora na výslednou DPS platformy. . . . .	27
4.15	Pohled zespoda na výslednou DPS platformy. . . . .	28
4.16	Detailní záběry na vybrané části DPS platformy. . . . .	28
4.17	Vývojový diagram obsluhy přerušení při přijmutí nového znaku na seriové lince. . . . .	29
4.18	Vývojový diagram hlavní smyčky firmwaru. . . . .	30
4.19	Zapojení zařízení na sběrnici I2C. . . . .	33
4.20	Příklad obvyklé komunikace na sběrnici I2C. . . . .	33
4.21	Příklad komunikace I2C při změně hodnoty vnitřního registru expanderu. . . . .	34
4.22	Příklad komunikace I2C pro nastavení výstupního napětí na individuálním DAC. . . . .	35
4.23	Příklad komunikace I2C pro spuštění měření ADC. . . . .	37
4.24	Příklad komunikace I2C pro čtení výsledku měření z ADC. . . . .	37
5.1	Výsledky prvotních experimentů s prototypem. . . . .	39

# Kapitola 1

## Úvod

Drtivá většina dnešní výpočetní techniky využívá k vykonávání výpočetních operací množství tranzistorů v procesoru, vyrobených z polovodičů, nejčastěji křemíku. Výpočetní technika, od mobilních zařízení přes počítače po servery, je součástí našeho každodenního života. Požadavky na jejich kapacitu a rychlost stále rostou, mimo jiné díky stále se zvyšujícímu množství zpracovávaných informací. Avšak výkon křemíkových procesorů nelze navyšovat do nekonečna, protože tranzistory jsou již v procesoru tak malé, že naráží na hranice fyzikálních zákonů. A zdá se tedy, že takové procesory se již blíží svým limitům a i populární Moorův zákon časem přestane platit [9].

Pod pojmem výpočet si můžeme představit libovolnou sekvenci operací, která předem určeným způsobem transformuje vstupní data na výstupní. Jak již bylo naznačeno, dnešní zařízení provádí tyto operace převážně pomocí křemíkových tranzistorů. Ale existují i jiné přístupy k provádění takových výpočtů, stačí se například podívat na lidské tělo, konkrétně na DNA v hloubce buněk. Totiž i DNA provádí výpočty, a dokonce lze tuto kyselinu deoxyribonukleovou přimět k tomu, aby prováděla výpočet takového zadání, které si určíme. Tuto myšlenku popularizoval Leonard Adleman v jeho experimentu z roku 1994 [2]. Provádění takových výpočtů je zajisté fascinující, ale tyto výpočty trpí vadou, že je lze provádět pouze v laboratoři a žádají si svůj čas. Adleman udává že experiment s DNA trval 7 dní usilovné laboratorní práce.

Lze provádět výpočty i v jiných materiálech, než v DNA? Totiž z článku Setha Lloyd [11] lze zjistit, že výpočetní limity hmoty jsou diametrálně dál, než co dokážeme s aktuální technikou dnes. Pan Lloyd stanovil teoretické limity výpočtu pouze na základě fyzikálních veličin, jako třeba rychlost světla, Planckova konstanta, gravitační konstanta apod. Závěry byly následující: Teoretický 1 kg hmoty může být schopen vykonávat až  $5.4258 \times 10^{50}$  operací za sekundu, nebo uložit přibližně  $10^{31}$  bitů dat. Výsledky práce naznačují myšlenku, že možnosti výpočtu v hmotě/materiálu jsou ohromné. A zdá se, že prozatím stále neznáme, jak tyto možnosti materiálů využít pro náš prospěch.

Cílem této práce je tedy následující: Seznámit čtenáře s přehledem tematiky, což zahrnuje nekonvenční přístupy k provádění výpočtů, jako třeba pomocí DNA, nanotrubicí, mikročástic anebo kapalného krystalu. Následně se text věnuje článkům pana Hardinga a Millera [16][5][7][6], v kterých se mimo jiné věnují výzkumu v oblasti výpočtu v kapalném krystalu. Na těchto člancích staví hlavní cíl práce, kterým je vytvoření snadno použitelné univerzální platformy pro realizaci experimentů nad výpočtem v kapalném krystalu a následné zopakování původních experimentů z uvedených článků. Text popisuje proces návrhu a implementace platformy, provedení experimentů s kapalným krystalem a následnou diskuzi výsledků v porovnání s výsledky z článků pana Hardinga a Millera.

## Kapitola 2

# Výpočty v materiálech

Pevným bodem, od kterého se celá tato práce odráží, je sada vybraných článků od pana Hardinga, pana Millera a jejich týmu [16][5][7][6], kterým se podrobně věnuje další kapitola 3. Jelikož zmíněné články vznikaly v letech 2004 až 2008, znamená to, že nejstarší článek vznikl skoro před 20 lety. Za tuto dobu možnosti techniky a výsledky výzkumu šly kupředu, nicméně je důležité zdůraznit, že některé znalosti zůstávají neměnné.

Proto tato kapitola poskytuje širší náhled do problematiky výpočtu v materiálech. A to díky kombinaci přehledů starších článků, s kterými mohl pracovat již pan Harding, tak i ukázky prací, které vznikly teprve nedávno. Samozřejmě, že článků a prací zpracovávající tuto tematiku je mnoho, proto jsou v této kapitole popsány pouze vybrané články, které jsem dle svého nejlepšího svědomí uznal jako podstatné a zároveň zajímavé.

### 2.1 Vlastnosti materiálů vhodných pro výpočet

Vlastnosti materiálů, v kterých by bylo možné provádět výpočet, je obtížné rigorózně definovat. Je ale důležité, určit si požadavky a očekávání od takových materiálů. Pojmeme výpočet vnímáme transformaci vstupních dat, dle zadaného postupu, na data výstupní. Takový výpočet často využíváme k řešení nějakého problému/úlohy. Z toho plyne, že materiál musí být schopen přijímat vstupní data a generovat výstupní data takovým způsobem, který je pro nás čitelný. A samozřejmě nejdůležitějším požadavkem je schopnost provést samotný výpočet uvnitř materiálu. Bohužel, oba tyto požadavky jsou těžko uchopitelné.

Zkusme si představit hypotetický materiál, který obsahuje mřížku buněk na ploše nebo prostoru. Tyto buňky jsou spolu propojeny, a ne nutně pouze sousední. Každá buňka umí realizovat primitivní funkci a propojení mají schopnost přenášet mezi buňkami informaci. Realizované primitivní funkce a propojení mezi buňkami mohou být předem dané povahou materiálu, nebo konfigurovatelné.

Pro výpočty hledáme právě takové materiály, které obsahují bohaté vnitřní struktury (buňky) a bohaté interakce (propojení), jejichž patřičné nakonfigurování je využitelné k výpočtu. Konfigurací můžeme také uvažovat pouze znalost určující, jakou kombinaci struktur s neměnnou funkcí včetně neměnných propojení využít pro řešení zadaného problému. Situaci mírně komplikují kapaliny/molekuly jako třeba DNA, u kterých může být samotná struktura informací. A pomocí vztahů a reakcí na další molekuly mění svoje vlastní uspořádání a tím provádí transformaci vstupní informace na výstupní, chápanou jako výpočet. Tyto úvahy lze shrnout tak, že: Vhodný materiál pro výpočet by měl obsahovat bohaté vnitřní struktury a bohaté vnitřní interakce, které lze konfigurovat.



Totíž při běžném návrhu libovolného systému se návrháři řídí určitými pravidly. V komplexnějších systémech je pro člověka, jako návrháře, lehké se ztratit. Dle dostupných zdrojů mají dnešní procesory miliardy tranzistorů [18], je nemyslitelné, aby kdokoliv znal design procesoru celý. Přitom každý tranzistor má svojí pevně definovanou funkci. Například je součástí hradla AND, společně s dalšími tvoří sčítačku, poté ALU, poté jádro a poté celý procesor. Návrháři si komplikovaný problém rozloží na jednodušší podproblémy, které pak řeší odděleně. Tato řešení pak opět spojují jako black-box komponenty, do komplikovanějších řešení, aby vznikl komplexní systém s požadovanou funkcionalitou.

V přírodě a světě kolem nás tento sofistikovaně strukturovaný systém překvapivě nevidíme. Zdá se, jako by všechno bylo propojeno se vším a nelze jednoduše určit komponenty a závislosti. I přes to, tyto systémy fungují, aniž by je někdo strukturovaně navrhl. Jako příklad může posloužit lidské tělo, které je nepochybně komplexní systém. Jenom samotný imunitní systém má svoje závislosti po celém těle a pracuje pomocí složitého soustavy chemických reakcí. Ta na první pohled vypadá jako pouhý chaos. Tuto problematiku lidsky přibližuje kniha od Philippa Dettmera [4]. Obecně se příroda zdá, jako záhadně fungující chaos. A nelze ho jednoduše popsat, nakreslit schéma, napsat specifikaci.

Hlavní roli totiž hraje evoluce, která k "návrhu" systémů přistupuje odlišně, než běžní lidští návrháři. Evoluce je populární termín známý široké veřejnosti. I tento pojem je také hojně využit v této práci. Avšak nikoliv evoluce, jakou známe z přírody, ale evoluční algoritmy, které pouze simulují a využívají principy reálné evoluce. Použité evoluční algoritmy pak hledají v materiálech kombinace a konfigurace struktur, na první pohled neviditelné, umožňující požadovaný výpočet. Jak evoluce v přírodě využila komplexnosti světa a materiálů pro stvoření života, tak simulované evoluční algoritmy využívají komplexnosti struktur v materiálech k stvoření výpočetních zařízení.

## 2.2 Výpočet v DNA

Studium výpočtů v DNA započal Leonard Adleman svým článkem [2] v časopisu Science. Článek obsahoval experiment, v kterém použil DNA pro řešení problému Hamiltonovské kružnice v orientovaném grafu. Tento problém je NP-úplný. Následující odstavce shrnují princip experimentu. Detailní popis lze najít v samotném článku a zachází do biochemie, která je mimo obor této práce.

Experiment začínal zakódováním jednotlivých vrcholů grafu unikátními náhodnými sekvencemi 20 nukleových bází T, A, C a G. Poté následovalo vytvoření úseků samostatných jednošroubovic DNA (o délce 20 bází) značící přechody mezi vrcholy grafu, postupně z 2. poloviny sekvence zdrojového vrcholu a 1. poloviny sekvence cílového vrcholu. Dále bylo potřeba vytvořit úseky opačné jednošroubovice obsahující komplementární kód jednotlivých vrcholů. Samotný výpočet se prováděl ve zkumavce, v které se nachází společně jednošroubovice DNA značící přechody mezi vrcholy (složena z polovin sekvencí vrcholů) a tak i komplementy vrcholů (složena z komplementu celé sekvence vrcholu). Jednošroubovice se pak spojují do dvojšroubovic, kde každá značí jednu cestu grafem, protože je to v podstatě sekvence přechodů.

Následovala posloupnost filtrování, čištění a duplikování DNA pomocí bílkovin a dalších chemikálií. Toto odfiltrování zajistí, že ve zkumavce zůstanou pouze takové cesty grafem, které obsahují všechny vrcholy a každý pouze jednou. Pokud ve zkumavce nějaká taková cesta grafem zůstane, pak graf obsahuje Hamiltonovskou kružnici. Za povšimnutí stojí, že tímto způsobem je řešen NP-úplný problém. I náročnější řešení se stále "vejdu" do zkumavky, protože výpočet probíhá na molekulární úrovni.

Bohužel Adleman upozorňuje na chybovost procesu, a to především na skutečnost, že vytváření všech možných cest grafem (dvojšroubovic) probíhá mícháním a to nemusí být dokonalé. Algoritmus je nedeterministický a spoléhá na to, že ve zkumavce vzniknou všechny možné cesty grafem.

Novější článek bližší tématu této diplomové práce pochází z roku 2004 [17]. Japonští vědci přicházejí s konceptem realizace logických hradel pomocí DNA (ne jako první). Využívají opět komplikované principy biochemie a vše s tím související. Nicméně lze zjednodušeně říci, že dokážou zakódovat logické hradlo nebo složitější logickou funkci složenou z logických hradel do jednošroubovice a opačná jednošroubovice obsahuje vstupní data. Poté následuje posloupnost chemických reakcí, které využívají upravené báze a skrz DNA je pro výpočet přesouvána absence molekuly. Výsledkem článku je příklad zakódování plně binární sčítačky v DNA a její funkce.

### 2.3 Výpočet v emulzi nanočásticí

Emulzi nanočásticí popisuje článek z roku 2001 [8]. Článek se nevěnuje ani tolik výpočtu, jako samotné konstrukci mikrovodičů z nanočásticí. Na emulzi je aplikováno elektrické pole a díky dielektroforéze jsou formovány mikrovodiče mezi vodivými plochami. Samotné vodiče žádný výpočet neprovádějí, ale tomuto konceptu je důležité také věnovat pozornost.

Takové materiály, které dokážou samy v sobě konstruovat propojení, jsou pro výpočty v materiálech žádoucí. Emulzi lze použít jako médium, které poskytuje konfigurovatelné propojovací pole mezi dalšími materiály. Ideálním materiálem pro výpočet totiž zajisté nebude pouze jeden materiál, ale kombinace více materiálů, jak lze například vidět v přírodě. A právě taková emulze se schopností propojovat více odlišných materiálů, může být pro výpočty v materiálech příhodná.

### 2.4 Výpočet v uspořádání nanotrubicí

Článek o uspořádání uhlíkových nanotrubicí [10] má, hned po Hardingových člancích, k této práci nejbližší. Autoři pracují se stejnými myšlenkami, a dokonce se sami odkazují na články pana Hardinga a Millera o kapalném krystalu.

Základem experimentu je podložka s několika vodivými ploškami plnícími funkci elektrod. Podložka má rozměr přibližně  $300 \mu\text{m}$ . Na tuto podložku s elektrodami je následně náhodně rozmístěna množina uhlíkových nanotrubicí. Zvolí se vstupní elektrody, výstupní elektrody a konfigurační elektrody. Nanotrubičky se poté chovají jako odporová síť, kterou lze simulovaným evolučními algoritmy přimět k tomu, aby vykonávala funkci logického hradla.

Autoři přistupují k hradlům z nanotrubicí v režimu "threshold logic", tudíž hradlo má definovanou hladinu výstupního napětí, kterou když překročí, je výstup chápán jako log. 1 a naopak. Výsledky byly takové, že zvyšující se binární hodnota vstupů monotónně zvyšuje i výstupní napětí. Nicméně práce dospěla k úspěšné realizaci hradel AND nebo OR. Pro hradlo XOR bylo nutné rozšířit "threshold logic" o další spodní práh.

### 2.5 Výpočet v kapalném krystalu

Posledním příkladem materiálu, zmíněným v této kapitole, je kapalný krystal. Tento materiál detailně popisuje následující kapitola 3. Nicméně tento materiál se podobá uspořádání nanotrubicí, s rozdílem, že funkci nanotrubicí zastávají molekuly kapalného krystalu.

## Kapitola 3

# Evolution in materio

Následující kapitola se podrobněji soustředí na přístup k výpočtu v materiálu od pana Hardinga, pana Millera a jejich týmu [16][5][7][6]. Zmíněné články realizují obecné myšlenky vycházející z kapitoly 2 do konkrétních řešení. Přicházejí s platformou pro realizaci výpočtu v kapalném krystalu, nad kterou následně provádějí experimenty, mimo jiné, za účelem realizace logických hradel. Tato kapitola popisuje principy z uvedených článků včetně konstrukce a realizace experimentů s touto platformou, která slouží jako výchozí bod pro následný návrh a konstrukci platformy prezentované v této práci.

### 3.1 Simulovaná evoluce v materiálu

S pojmem "Evolution in materio" přichází jako první článek s názvem "Evolution in materio: Looking Beyond the Silicon Box" [16]. Předchozí kapitola již trochu nastínila, co si můžeme pod takovým pojmem představit. "Evolution in materio" lze stručně definovat jako využití evolučních algoritmů pro zkoumání vlastností vnitřní struktury materiálu, s cílem najít takové konfigurace struktur materiálu, jenž řeší vybrané úlohy. Tyto konfigurace jsou právě to, co se vyvíjí pomocí simulované evoluce. Typ a způsob jakým je konfigurace aplikována na materiál samozřejmě závisí na druhu a povaze materiálu.

Takovou konfiguraci lze vyjádřit v rámci evolučního algoritmu genomem. Každý takový genom lze ohodnotit takzvanou fitness funkcí. Tato funkce hodnotou udává, jak je daný genom úspěšný v řešení zadané úlohy. Evoluční algoritmy zpravidla obsahují populaci kandidátních řešení, z kterých se křížením a mutacemi vytváří množina genomů, z které jsou následně vybrány nejúspěšnější genomy jako další generace a proces celý se opakuje. Velikost populace, počet generací, způsob výběru genomů pro křížení, metody křížení, mutací a samotné reprezentace genomů závisí na zvoleném evolučním algoritmu a materiálu.

### 3.2 Vlastnosti kapalného krystalu

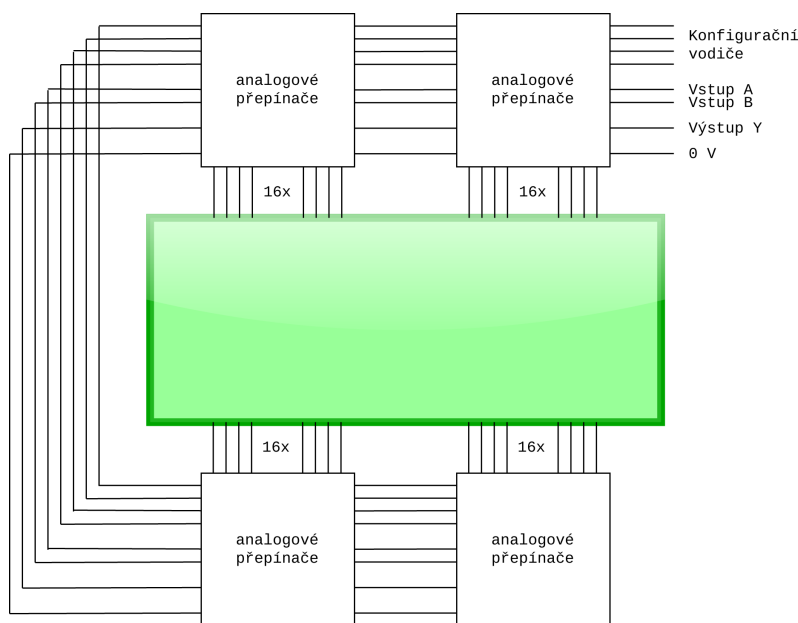
Pan Miller a Downing si ve své práci vybrali jako slibný materiál pro výpočet právě kapalný krystal. Svůj záměr odůvodnili větou: "*Liquid crystal appears to be most promising in this regard as it digitally writeable, reconfigurable and works at a molecular level.*" [16] Kapalný krystal je zajímavý svojí strukturou, protože molekuly mohou být organizovány do mřížky jako krystalické materiály, ale zároveň se může nacházet v stavu kapalném, to znamená, že molekuly jsou v materiálu rozmístěné náhodně. Zároveň lze přechod mezi těmito stavy kontrolovat pomocí elektrického pole a lze tak materiál snadno konfigurovat.

### 3.3 Realizace logických hradel v kapalném krystalu

Tato podkapitola se věnuje konkrétní konstrukci platformy a experimentu s cílem realizace logických hradel v kapalném krystalu z článku "Evolution In Materio: Evolving Logic Gates in Liquid Crystal" [6]. Platforma z uvedeného článku bude nadále v textu nazývána jako původní platforma. Pan Harding v rámci původní platformy pro experimenty použil monochromatický displej s rozlišením 180 na 120 pixelů. Displej obsahoval 64 vývodů, kterými bylo možné přímo ovládat kapalný krystal v displeji. Je klíčové, aby byly vodiče uvnitř displeje přímo připojeny ke kapalnému krystalu a displej tím pádem neobsahoval žádný kontrolér. Displej s kontrolérem je pro tento druh experimentů nepoužitelný.

Všechny vodiče tohoto displeje bylo možné, pomocí sestavené platformy, připojit přes analogové přepínače libovolně k jednomu z osmi napěťových vodičů. Čtyři z těchto napěťových vodičů byly pevně nastaveny na: jeden zemnicí vodič, dva vstupní vodiče a jeden výstupní. Zbylé čtyři vodiče byly ponechány evolučnímu algoritmu, který hledal konfiguraci napětí na těchto vodičích v intervalu od -10 V do +10 V. Schéma zapojení napěťových vodičů přes analogové přepínače k displeji ukazuje obrázek 3.1.

Na vstupní napěťové vodiče bylo, pro log. 0 vstupní kombinace, připojeno napětí 0 V a pro log. 1 napětí +1 V. Před snímáním napětí na výstupním napěťovém vodiči bylo, po připojení napětí na vodiče displeje, vyčkáno 200 ms z důvodu ustálení všech vnitřních struktur displeje. Prahové napětí výstupního napěťového vodiče, podle kterého se rozhodovalo zda je na výstupu log. 1 nebo log. 0, bylo nastaveno na +0.1 V.



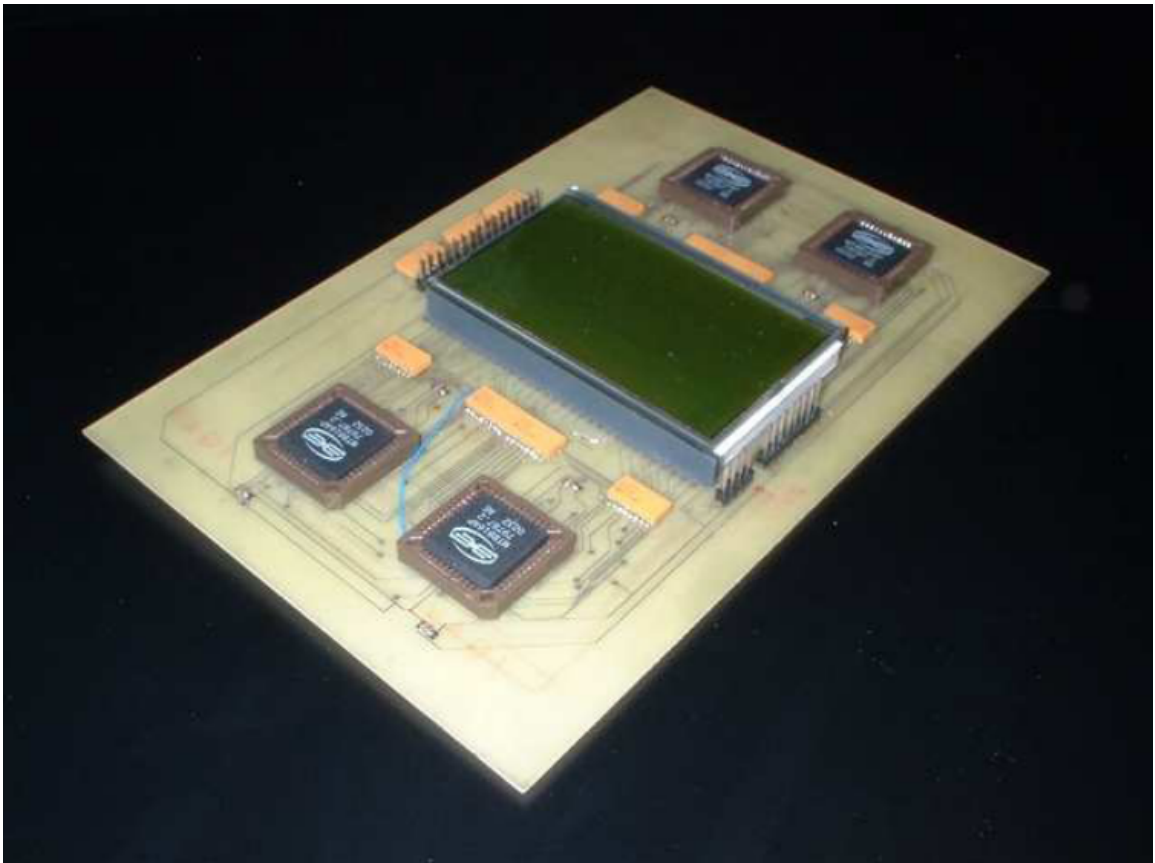
Obrázek 3.1: Schéma zapojení vodičů displeje k napěťovým vodičům. Inspirováno schématem z [6].

Samotný genom v evolučním algoritmu byl rozdělen na dvě části. První část obsahovala 64 čísel v rozsahu 0 - 7. Toto pole čísel reprezentovalo připojení jednotlivých vodičů displeje k jednomu z osmi napěťových vodičů pomocí analogových přepínačů. Napětí na 4 konfigurovatelných napěťových vodičích ovlivňovala druhá část genomu, která byla složena ze čtyř hodnot napětí v rozsahu od -10 V po +10 V, příslušící jednotlivým konfigurovatelným napěťovým vodičům.

Pro evoluční algoritmus je dále zásadní fitness funkce. Hodnota této funkce udává úspěšnost genomu v řešení zadané úlohy. Výpočet této funkce byl následující: Každá platná kombinace na vstupech hradla (00, 01, 10, 11) byla v kapalném krystalu otestována 3x a v náhodném pořadí. Vzniká tak 12 testování. Výsledkem fitness funkce byl součet správných výstupů kapalného krystalu v závislosti na vyvíjenou logickou funkci. Hodnota fitness funkce tedy mohla nabývat celočíselných hodnot v intervalu od 0 do 12.

V rámci experimentu z uvedeného článku byl použit evoluční algoritmus za cílem realizace logických hradel pomocí popsaných principů a s následujícím nastavením: Velikost populace byla stanovena na 40 jedinců. Maximální počet generací byl omezen na 100 generací. Metoda křížení nebyla použita. Metoda mutací byla omezena na maximální počet pěti mutací jednotlivých položek v genomu. Každá mutace spočívala v náhodném výběru jedné položky genomu a její změnou na jinou náhodnou hodnotu. Do nové generace přecházelo vždy 5 genomů, vybraných pomocí turnajového výběru.

Sestrojená původní platforma z roku 2004 pro experimenty s kapalným krystalem je na obrázku 3.2. Výsledky z článku ukazují, že se podařilo v kapalném krystalu naleznout všechna libovolná hradla od funkce OR, přes NAND, po XOR. Avšak za funkční hradla byla považována i taková, jejichž hodnota fitness funkce byla alespoň větší nebo rovna 10. Vyvíjená hradla tedy nemusela být vždy absolutně korektní pro všechny vstupní kombinace. Nicméně autoři také uvádějí, že nejobtížnějším vyvíjeným hradlem byla funkce XOR.



Obrázek 3.2: Experimentální platforma pro výpočet v kapalném krystalu pana Hardinga a Millera. Převzato z [7].

## Kapitola 4

# Návrh experimentální platformy

Tato kapitola popisuje proces návrhu vlastní experimentální platformy vycházející z původní platformy pana Hardinga popisované v předchozí podkapitole 3.3. Vlastní návrh experimentální platformy má za cíl vyrovnat se možnostem původní platformy, a navíc přinést inovace s využitím technologií ze současné doby. Jakým způsobem toho je dosaženo, ukážou následující podkapitoly. Pozornost je věnována všem důležitým aspektům, které takový proces návrhu zahrnuje. První podkapitola se věnuje návrhu a konstrukci prvotního prototypu platformy, následně se již kapitola věnuje návrhu výsledné platformy, který sestává z výběru a návrhu zapojení elektronických součástek, návrhu desky tištěných spojů, její realizace a následné implementace řídicího firmwaru.

### 4.1 Prvotní prototyp

Prvotní prototyp platformy sloužil k otestování, zda je vůbec výpočet v kapalném krystalu realizovatelný a má smysl pokračovat v této práci. Prototyp je zapojen v nepájivém poli a obsahuje pouze omezenou funkcionalitu původní platformy. Nicméně funkcionalitu takovou, která postačuje na otestování základních principů realizace logických hradel v kapalném krystalu.

#### Specifikace požadavků

Hlavním požadavkem na prototyp bylo vytvořit co nejkompaktnější obvod, na kterém by bylo možné ověřit principy experimentů nad výpočtem v kapalném krystalu a tím i význam budoucí práce. Prototyp proto musí být jednoduchý a lehce a rychle sestavitelný. Na takové aplikaci je příhodné použít nepájivé pole, v kterém je i samotný prototyp zapojený.

Prototyp musí umožňovat připojit LCD displej pomocí alespoň osmi vodičů. Tento počet je sice pouze osminou počtu vodičů oproti původní platformě, ale zároveň dostatečně netriviální pro potřeby experimentů. Navíc je počet 8 vhodný z pohledu elektronického návrhu i softwarové implementace.

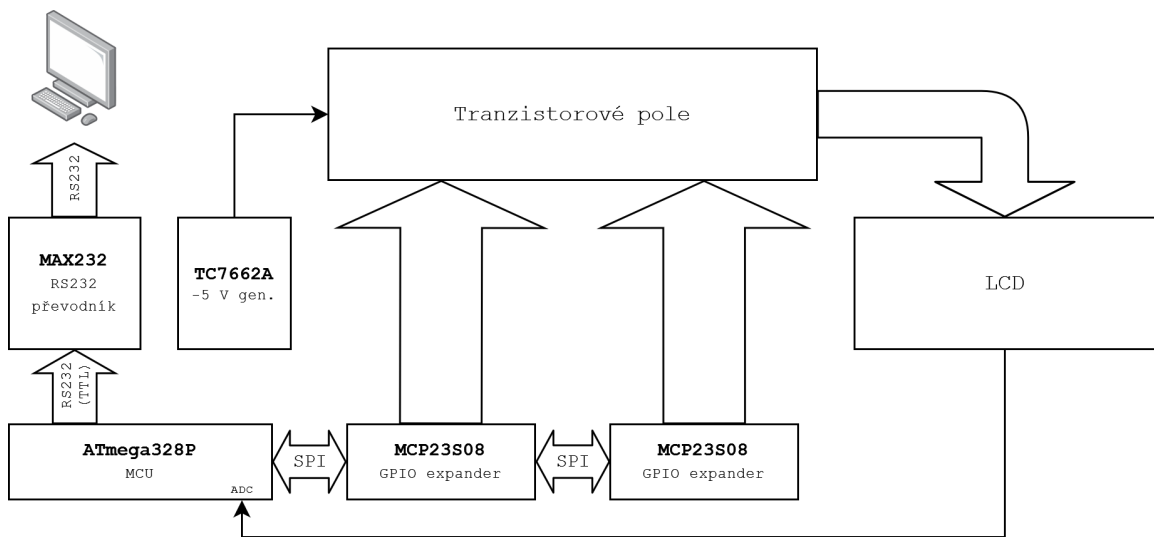
Po vzoru původní platformy musí být prototyp schopen na těchto vodičích displeje řídit napětí mezi alespoň jedním kladným, záporným a nulovým napětím. Tyto tři napětí stačí na ověření předpokladu, že displej reaguje na různá napětí na vodičích displeje. Dále musí být prototyp schopen číst napětí na vybraném výstupním vodiči displeje. Posledním zásadním požadavkem je pak možnost připojit platformu k počítači, z kterého je možné číst výsledky experimentů a ověřit tak možnosti kapalného krystalu.

Shrnutí rozdílů prototypu oproti původní platformě:

- Prototyp využívá pouze 8 vodičů displeje, namísto 64.
- Prototyp využívá pouze 3 napěťové vodiče, namísto 4.
- Napěťové vodiče mají fixní hodnoty: +5 V, -5 V a 0 V, oproti libovolnému napětí v intervalu od -10 V do +10 V.
- Prototyp neobsahuje vstupní vodiče, oproti původní platformě, a přivedení logické kombinace je realizováno změnou propojení vybraných vodičů displeje s napěťovými vodiči.

## Zapojení prototypu

Celý prototyp je zapojený v jednom samostatném nepájivém poli. Nepájivé pole je ideální právě pro tento druh prototypů a je to jednoduchý a rychlý způsob, jak sestavit jednodušší druhy elektronických obvodů. Bohužel nepájivé pole trpí na občasné rozpojování kontaktů, které dokonce nemusí být viditelné. Navíc vzniká omezení na součástky, které musí být pouze typu THT. Proto výsledná platforma vyžaduje vlastní DPS. Blokové schéma zapojení prototypu je na obrázku 4.1. Tato podkapitola popisuje pouze obecný přehled zapojených součástek a vynechává podrobnosti, protože tento prototyp není hlavním výsledkem práce.



Obrázek 4.1: Blokové schéma zapojení prototypu platformy.

Řídící jednotkou celé platformy je MCU ATmega328P. Tento mikrokontrolér bohatě dostačuje pro řízení celé platformy a zároveň je pohodlně dostupný a použitelný pro podobné elektrotechnické projekty. Komunikace s počítačem je zajištěna pomocí sériového portu. V rámci prototypu pouze ve směru z platformy do počítače, jelikož platforma pouze odesílá výsledky experimentů, které jsou naprogramovány přímo v platformě. Ovšem pro uskutečnění takové komunikace je potřeba převést logické úrovně TTL do logických úrovní sériové linky RS232, k tomu slouží připojený RS232 převodník.



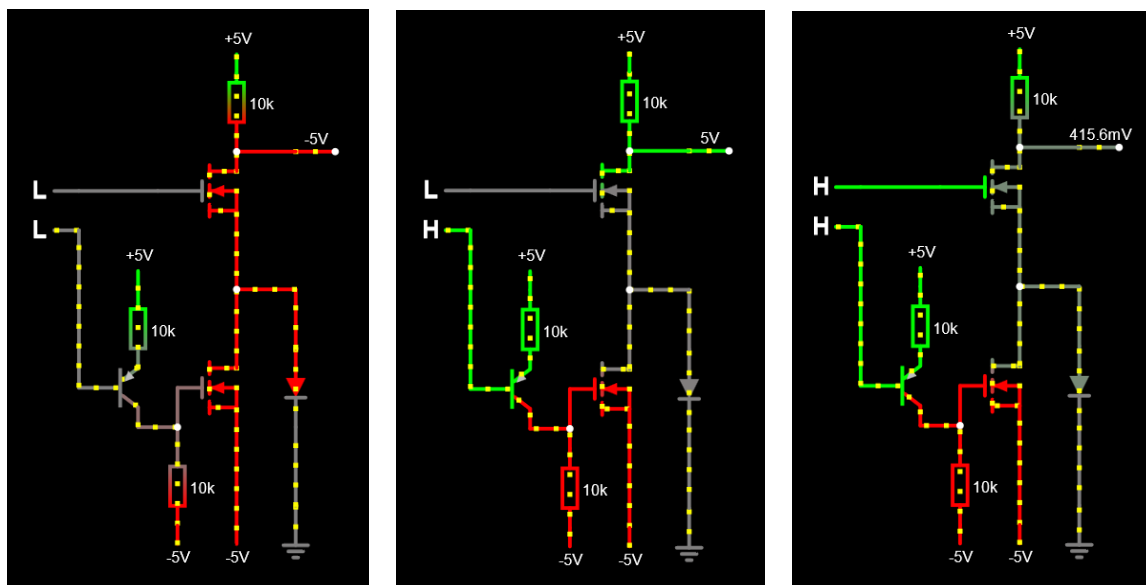
K MCU jsou dále připojeny GPIO expandery pomocí sběrnice SPI. Jelikož MCU nemá dostatek GPIO pro obsluhu požadovaného množství vodičů displeje (konkrétně tranzistorového pole), je toto řešení nutné. Na každý jeden vodič displeje z osmi připadají 2 řídicí vodiče (3 výstupní napětí), proto jsou využity dva 8-bit GPIO expandery.

Expandery ovládají tranzistorové pole, které využívá napájecích +5 V, 0 V a generovaných -5 V. Pro generování -5 V slouží připojený invertor napětí ze vstupních 5 V. Dle datového listu se tento invertor chová jako napěťový zdroj s vnitřním odporem přibližně 50  $\Omega$ , ale nepředpokládá se, že displejem budou téct dostatečně velké proudy na to, aby generované napětí výrazně stoupl.

Tranzistorové pole obsahuje zapojení z obrázku 4.2 pro každý jednotlivý vodič displeje. Toto tranzistorové pole dokáže každý vodič displeje přepínat, na základě hodnoty na 2 řídicích vodičích, mezi napěťovými vodiči +5 V, 0 V a -5 V.

Zapojení tranzistorů využívá dvou N-MOSFETů, kde první spíná výstup na +5 V a druhý na -5 V. Jelikož MOSFET spínající -5 V potřebuje na svém gate vstupu pro uzavření také -5 V, je na jeho gate připojen PNP tranzistor, který převádí vstupních 0 V na 0 V a 5 V na -5 V a zajišťuje tak korektní uzavření MOSTET tranzistoru. Dále je mezi tranzistory dioda, která umožňuje výstup připojit také na 0 V. Všechny povolené konfigurace řídicích vodičů a následné výstupní napětí na vodiči displeje jsou na obrázku 4.2.

Vodiče z tranzistorového pole jsou následně připojeny k LCD displeji. A z displeje je vyveden jeden fixní vodič propojující vybrané výstupní vodiče displeje a je veden do AD převodníku v MCU, který provádí následné měření výstupního napětí při experimentech.



Obrázek 4.2: Schéma zapojení jednoho vodiče displeje v tranzistorovém poli a princip funkce při povolených kombinacích řídicích vodičů.

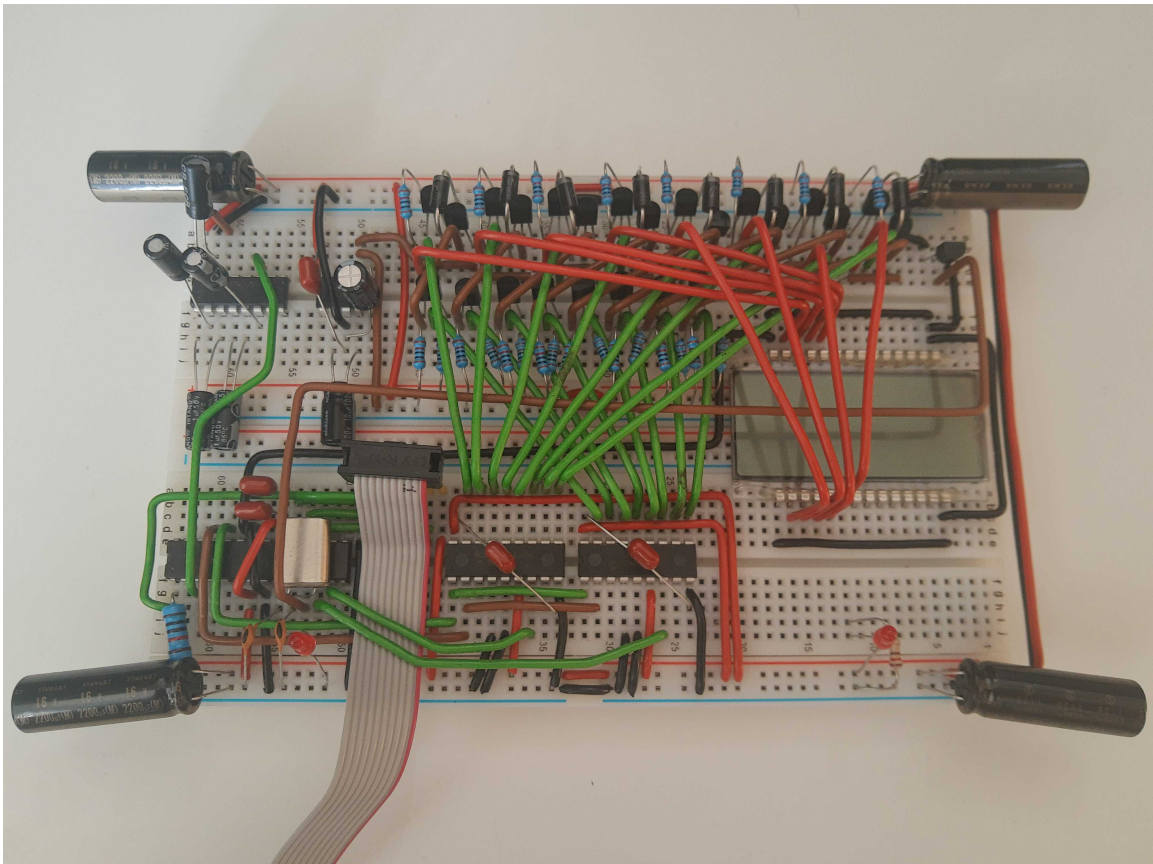


## Realizace prototypu

Zkonstruovaný prototyp je na obrázku 4.3. Prototypu dominují čtyři kondenzátory stabilizující napětí v okrajových částech nepájivého pole. Rozložení součástek přibližně odpovídá rozmístění v blokovém schématu. Je jasné vidět, že ovládání více vodičů displeje by vyžadovalo další nepájivé pole.

Ostatní součástky, které jsou zapojeny v prototypu, ale nejsou popsány v blokovém schématu, jsou potřebné pasivní součástky pro funkci hlavních komponent. Jsou to například stabilizační a externí kondenzátory pro RS232 převodník, invertor napětí apod.

Konkrétní zapojení všech součástek, a tak i výsledné zapojení není pro tuto kapitolu podstatné a ani tak pro celou práci. Prototyp byl stvořen pouze pro otestování základních principů z článků pana Hardinga. Pro relevantní experimenty je tento prototyp nepoužitelný, především svojí omezenou funkcionalitou, ale také poruchovostí, která znemožňuje déle běžící experimenty. Nicméně prototyp ověřil pomocí prvotních experimentů, které popisuje podkapitola 5.1, že displej reaguje na napětí na jeho vodičích a na výstupním vodiči lze měřit různá napětí při odlišných kombinacích napětí na vstupních vodičích displeje.



Obrázek 4.3: Prvotní prototyp platformy.

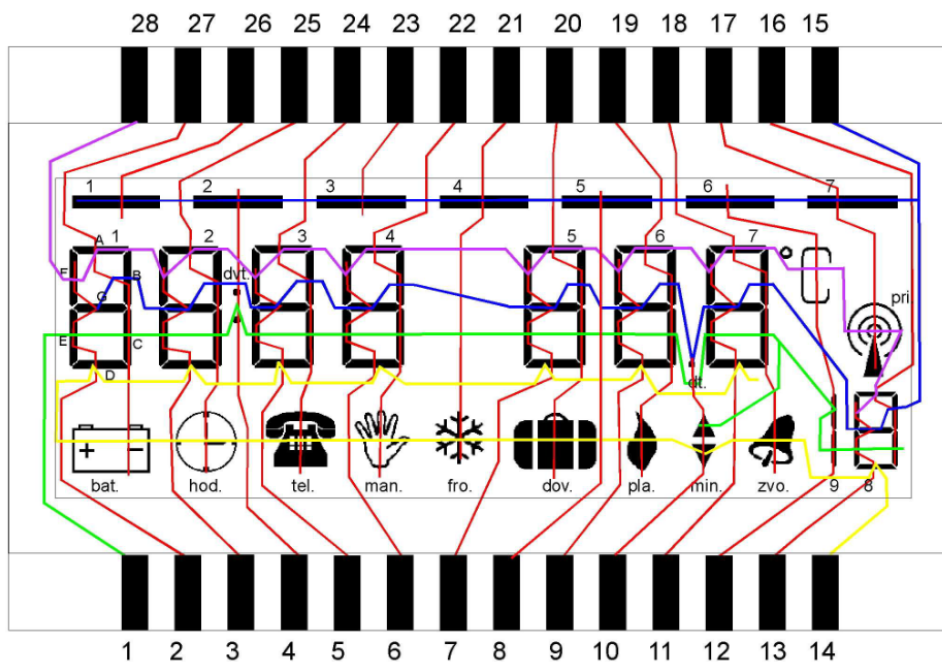
## 4.2 Specifikace požadavků

Tak jako byly definovány požadavky u prvotního prototypu, je nutné si taktéž specifikovat požadavky na funkcionalitu výsledné platformy. Prvotní prototyp dosahoval pouze zlomku možností původní platformy z článků pana Hardinga. U výsledné platformy je nutné, aby funkcionalita rozumnou formou dosahovala rozsahu původní platformy.

První požadavek na počet konfigurovatelných vodičů displeje velmi úzce souvisí s výběrem experimentálního displeje, nad kterým budou prováděny samotné experimenty. Jelikož na trhu v době návrhu platformy nebylo mnoho dostupných LCD displejů bez řadiče, byl jsem nucen zvolit LCD displej s označením LCD3924. Jedná se o obyčejný sedmisegmentový displej se sedmi číslicemi a několika dalšími piktogramy. Vnitřní zapojení displeje ukazuje obrázek 4.4.

Displej obsahuje 24 řídicích vodičů připojených zpravidla k 4 segmentům kapalného krystalu najednou. Tyto řídicí vodiče doplňují 4 zemnicí vodiče (1, 14, 15, 28), pomocí kterých lze aktivovat jeden nebo více ze čtyř segmentů připojených k řídicímu vodiči. Zemnicí vodiče vybízejí k funkci měřících vodičů, protože procházejí celým displejem a mohou měřit napětí skrz celým displejem. 24 řídicích vodičů vybízí k funkci konfigurovatelných vodičů displeje.

Aby bylo možné připojit a ovládat všechny řídicí vodiče vybraného displeje, vzniká tak požadavek na 24 konfigurovatelných vodičů displeje. Tento počet vodičů nedosahuje počtu 64 v původní platformě, avšak větší počet by v této práci nebylo možné rozumně využít. Nadále vzniká požadavek na 4 měřící vodiče (AD převodníky), které dokáží nezávisle měřit napětí na všech definovaných výstupních vodičích displeje.



Obrázek 4.4: Vnitřní zapojení experimentálního LCD displeje. Převzato z [1]

Původní platforma podporovala rozsah napětí na jednotlivých napěťových vodičích od -10 V do +10 V. Navrhovaná platforma v této práci se drží identického rozsahu a to především, protože větší napětí má již tendence displeje ničit. Experimentální displej má dle datového listu pracovní napětí v rozsahu 2 V až 4 V, avšak platforma chce zkoumat vlastnosti displejů na hranici pracovních podmínek. Tudíž rozsah napětí -10 V až +10 V dosahuje schopnostem původního zařízení a zároveň dimenzuje platformu pro experimenty s displeji v podmínkách hraničících s zničením. Větší rozsah by velmi pravděpodobně nepřinášel nic dalšího přínosného, ale pouze možnost efektivně ničit displeje.

Velmi důležitou součástí původní platformy byly analogové přepínače spínající napěťové vodiče k vodičům displeje. V tomto návrhu platformy tento přístup není uvažován. Namísto fixního počtu napěťových vodičů spínaných na větší počet vodičů displeje, je každý vodič displeje připojen přímo na samostatný konfigurovatelný napěťový vodič. Vzniká větší volnost a variabilita, protože na každý jednotlivý vodič displeje může být přiloženo libovolné napětí z rozsahu -10 V až +10 V. Nikoliv pouze jedno z omezené množiny napětí na napěťových vodičích, tak jako tomu bylo u původní platformy. Toto je velký přínos nově navrhované platformy, protože jí lze použít v různých dalších aplikacích, kde potřebujeme ovládat množinu napěťových vodičů jednotlivě. Nadále tento přístup vypouští potřebu vstupních vodičů.

Dalším, ale zásadním požadavkem je samotné použití platformy. Jelikož je platforma experimentální, musí být ze své povahy dostatečně univerzální. Tak, aby bylo možné pohodlně měnit zkoumané displeje případně jiné materiály/zařízení. Vzniká tak další požadavek, aby displej či jiné zkoumané zařízení bylo možné do platformy vložit, připojit libovolným způsobem napěťové a měřící vodiče a následně opět vyjmout a vyměnit za jiný displej/zařízení.

Platforma jako taková by neměla provádět žádné předprogramované experimenty, ale pouze poskytovat rozhraní k řízení napěťových a měřících vodičů. Samotné experimenty je v rámci udržení univerzálnosti vhodnější řídit z počítače příhodným vyšším programovacím jazykem, pomocí vhodného komunikačního spojení mezi PC a platformou. Při splnění těchto požadavků vznikne platforma univerzální, jak pro testování různých displejů/zařízení, tak i pro libovolnou implementaci experimentů, jako například evolučních algoritmů, genetického programování atd.

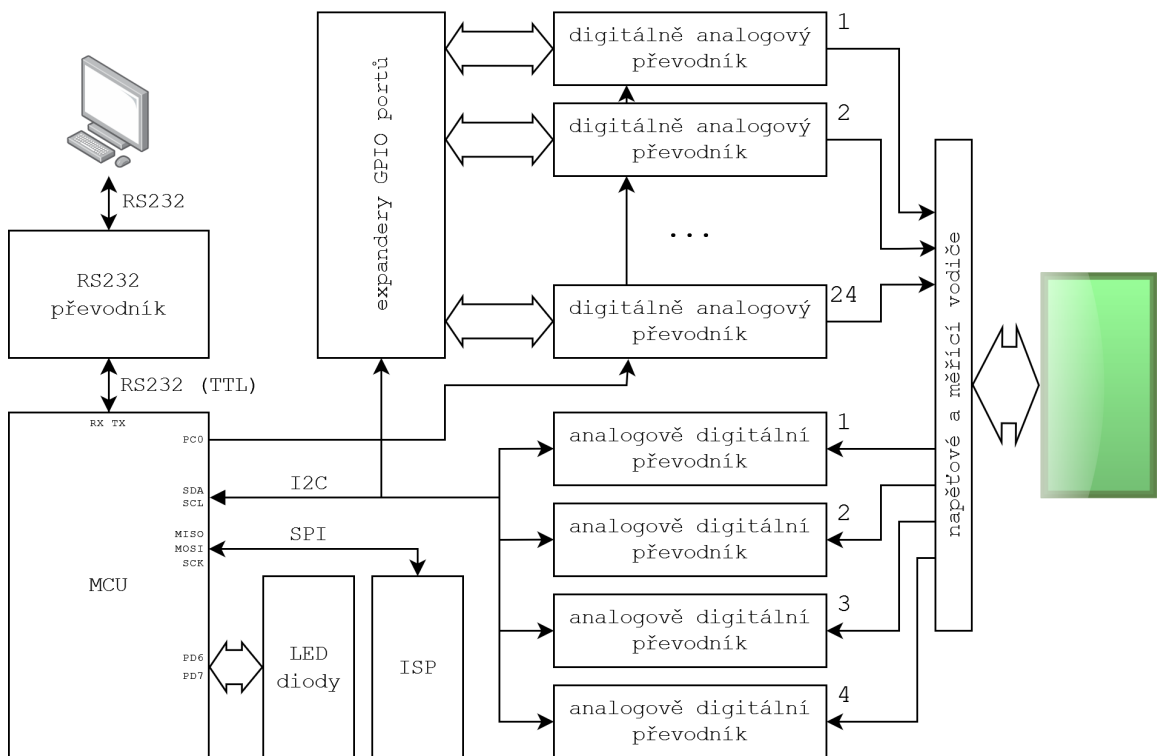
Shrnutí požadavků na výslednou experimentální platformu je následující:

- Platforma poskytuje 24 libovolně konfigurovatelných napěťových vodičů.
- Napěťové vodiče poskytují napěťový rozsah od -10 V do +10 V.
- Platforma poskytuje 4 měřící vodiče pro nezávislé měření výstupů.
- Experimentální displeje/zařízení je možné pohodlně v platformě vyměňovat a případně přepojovat napěťové a měřící vodiče.
- Experimenty je možné řídit pohodlně z PC pomocí příhodného vyššího programovacího jazyka.

### 4.3 Blokové schéma

Pomocí definování požadavků na funkcionalitu výsledné platformy je možné pokračovat v návrhu podrobněji. Vhodným dalším krokem je vytvořit blokové schéma celého systému jako celku. Blokové schéma musí obsahovat všechny klíčové logické celky a jejich vzájemné propojení tak, aby systém splňoval všechny definované požadavky. Proces návrhu blokového schéma se překrývá s výběrem elektronických součástek, nicméně v této práci jsou tyto 2 paralelní činnosti oddělené do dvou následujících podkapitol 4.3 a 4.4.

Na obrázku 4.5 je blokové schéma navrhované platformy. Jádrem celého systému je mikrokontrolér v bloku MCU, který je obvyklým řídicím elementem v podobných typech aplikací. Aby bylo možné s tímto mikrokontrolérem komunikovat a řídit experimenty z PC, je k mikrokontroléru připojeno rozhraní RS232, opět pomocí RS232 převodníku jako u prototypu platformy, ale nyní již obousměrně. Dále je k MCU připojeno rozhraní ISP, tak aby bylo možné MCU snadno přeprogramovat přímo na desce. Tato funkčnost usnadňuje vývoj firmware a případně umožňuje snadno změnit firmware platformy, například při nalezení chyby. Je dobrým zvykem připojit k MCU alespoň jednu programovatelnou LED diodu, například pro signalizaci vnitřní chyby nebo definovaného stavu MCU. Tuto funkci zajišťuje blok LED diody.



Obrázek 4.5: Blokové schéma výsledné platformy.

K experimentálnímu LCD jsou skrz konektor napěťových a měřících vodičů připojeny digitálně analogové převodníky (DAC) a analogově digitální převodníky (ADC). 24x DAC pro napěťové vodiče a 4x ADC pro měřící vodiče. ADC mohou být z povahy součástky připojeny přímo pomocí sběrnice I2C k MCU. Naopak DAC musí být připojeny k expanderům GPIO portů, a to z důvodu absence I2C rozhraní v použitých DAC. Tato skutečnost je diskutována později v podkapitole 4.4.

Expandery jsou následně připojeny k MCU pomocí sběrnice I2C a tím zprostředkovávají komunikaci mezi MCU a DAC. Za pozornost stojí vodič z PC0 vedoucí z MCU přímo do všech DAC, tento vodič spouští přenos dat z expanderů do DAC a jeho podrobná funkce a ovládání je také diskutována v podkapitole 4.4. Jelikož DAC, dle požadavků na platformu, musí podporovat rozsah napětí od -10 V do +10 V, ve výsledné platformě se musí nacházet generátory těchto napětí. Nicméně v blokovém schématu nejsou uvažovány, jelikož jde o otázku napájení, a nikoliv logického zapojení.

## 4.4 Výběr elektronických součástek

Blokové schéma nám dává přibližnou představu o tom, jaké součástky bude nutné pro sestavení platformy vybrat a následně zakoupit. V tuto chvíli je důležité podotknout, že v roce realizace návrhu této platformy (2023), trh s elektronickými součástkami zažívá výpadky, které se dotkly i mě při realizaci této platformy. Několikrát jsem musel měnit vybrané součástky a tím i navržený systém, protože původní vybrané součástky s kterými jsem uvažoval další návrh platformy, mizely z obchodů se součástkami ze dne na den. Také se mi stalo, že než jsem stihl dokončit celý návrh a nakoupit součástky, tak jednu nebo více součástek někdo ze skladu vykoupil a nová dodávka do skladů byla v desítkách týdnů. Proto se některé vybrané součástky mohou zdát jako neideální, ale často jsem musel činit rozhodnutí mezi neideální součástkou nebo žádnou součástkou.

K výběru součástek jsem využíval především své osobní zkušenost v kombinaci s internetovými obchody, které nabízí užitečné filtry. Tyto filtry usnadňují výběr za předpokladu, že nakupující ví, jaké parametry od součástky požaduje. Například pracovní napětí, počet výstupů apod. Obvykle filtrováním zbyde množina součástek se stejnými parametry od různých výrobců a rozdílnou cenou. Nejjednodušší je vybrat takové součástky, které jsou nejlevnější a zároveň splňující potřebné parametry.

V návrhu jsem si stanovil určité pevné body jako třeba MCU nebo DAC od kterých jsem následně odvíjel další výběr. Protože výběr jedné další součástky často způsobí potřebu dalších a vzniká vcelku komplexní síť s množstvím závislostí. Tyto závislosti pak může lehce zbořit například zmiňované vyskladnění součástky. Je dobré při návrhu postupovat systematicky, nejlépe právě po logických celcích z blokového schématu.

## Mikrokontrolér

MCU je samotným jádrem platformy a jeho výběru je nutné věnovat dostatek pozornosti. Již při výběru musíme znát co konkrétně bude mikrokontrolér řídit a jakým způsobem. Je potřeba znát jaká bude povaha vykonávaného kódu, z které plynou požadavky na paměť a rychlost MCU. Nadále musíme stanovit jaké MCU potřebuje periferie, například časovače či podporu sběrnic. A samozřejmě minimální počet GPIO pinů, které bude MCU potřebovat k činnosti.

Mikrokontrolérů je na trhu celá řada, jako například PIC nebo nově mikrokontroléry s jádrem ARM Cortex-M0+. Mimo tyto existuje také řada AVR osmi bitových mikrokontrolérů. Jelikož mám jako autor a AVR mikrokontroléry dobré zkušenosti, uchýlil jsem se právě k této rodině mikrokontrolérů. Konkrétně v MCU s označením ATmega328P [15].

Z pohledu programu na MCU nebudou kladeny velké nároky, kód bude hrát roli prostředníka mezi PC a připojenými DAC a ADC. Kód nebude obsahovat nijak paměťové náročné konstrukce, proto stačí 32 kB paměti flash pro program a 2 kB SRAM pro data, které toto MCU nabízí. ATmega328P dále obsahuje periferie pro UART, SPI a I2C, které MCU musí obsahovat dle blokového schéma. Vzhledem k počtu GPIO jsou požadavky: 2 (UART) + 2 (I2C) + 2 (LED) + 1 (DAC ovládací vodič) + 3 (ISP). To je 10 potřebných volných GPIO, avšak ATmega328P má až 20 volných GPIO, proto tento mikrokontrolér tuto podmínku také splňuje. Toto MCU tak bylo optimální volbou, protože splňuje všechny požadavky na paměť výkon i periferie, zároveň není zbytečně předimenzován a především bylo ho možné zakoupit v reálním čase.

## Digitálně analogové převodníky

Digitálně analogové převodníky, aneb DAC jsou druhou klíčovou částí celé platformy. Vykonávají funkci generování požadovaného analogového napětí na jednotlivých vodičích displeje. Je zřejmé, že DAC musí být v platformě celkem 24 a každý musí být možné řídit nezávisle. Zároveň musí být DAC schopný generovat napětí od -10 V do +10 V. Tento požadavek velmi snížil počet DAC, které je možné zakoupit, na jednotky.

Po podrobném zkoumání různých DAC jsem zvolil jako vhodný TLC7524C [19]. Bohužel tento 8-bit DAC musí být řízen vlastní sběrnicí, tedy pomocí osmi datových vodičů. Tudíž není možné s jednotlivými DAC komunikovat například po sběrnici I2C nebo SPI. Krom osmi datových vodičů, které reprezentují požadovanou digitální hodnotu na výstupu, je DAC řízen dalšími dvěma řídicími vodiči. Pokud jsou oba v log. 0, pak DAC převede přiloženou digitální hodnotu z osmi datových vodičů na analogové výstupní napětí.

Tyto skutečnosti právě způsobují potřebu GPIO expanderů, jak pro datové vodiče, tak i jeden z řídicích vodičů. Vzniká tak potřeba volných 8 + 24 tedy 32 GPIO, kterou jinak než expandery nelze uspokojit. Větší MCU by bylo zbytečným plýtváním materiálem. Jako expandery jsem vybral MCP23018 [13], kde každý obsahuje 16 GPIO pinů, a tak vystačí pro celou platformu pouze 2. Poměr ceny ku parametrům u těchto expanderů se zdál jako nejlepší.

DAC je dle svého datového listu nutné zapojit v kombinaci s dvěma operačními zesilovači. Toto zapojení přímo podporuje duální napěťový rozsah, tedy od -10 V do +10 V. Jelikož nejsou uvažovány velké proudy do LCD displeje, nejsou na operační zesilovače kladeny velké výkonové nároky, nicméně musí zvládnout požadovaný rozsah napětí. Jako výhodné operační zesilovače jsem nakonec vybral TL072C [21].



## Napěťové zdroje

Platforma je napájena 5 V, což vyplývá ze specifikace MCU, které větší napětí nepodporuje. Navíc, toto napětí je hojně používáno v logických obvodech a je to také například napětí používané v USB. Aby platforma mohla být snadno použitelná, nebylo by vhodné, aby musela být napájena specifickým napětím, které může generovat pouze laboratorní zdroj, nebo je jinak těžko dostupné. Platforma si proto generuje vlastní napětí 10 V pro DAC z napájecích 5 V. Dále si generuje 14 V z 5 V a z těchto 14 V následně také -14 V. 14 V a -14 V jsou nutné pro napájení operačních zesilovačů, protože dle datového listu musí být pro výstupní rozsah napětí od -10 V do +10 V napájení OZ minimálně od -14 V do +14 V.

Pro generování 10 V a 14 V jsem použil dva step-up regulátory LM2577T-ADJ [20]. Tyto napěťové zdroje jsou dimenzovány tak, aby měla celá platforma stabilní napájení s co nejmenším šumem a zároveň nedocházelo k nechtěným výkyvům napětí. Pro experimenty je žádoucí, aby bylo napětí co nejstabilnější.

Pro generování -14 V jsem zvolil invertor napětí TC7662B [12]. Každé jednotlivé DAC zapojení má svůj invertor napětí, ale větev -14 V je propojena navzájem mezi všemi DAC a tím se kompenzuje jinak vcelku velký vnitřní odpor tohoto invertoru napětí.

## Předovnik RS232 úrovní

Jak již bylo zmíněno u prototypu platformy, standard sériové linky RS232 definuje logické úrovně jinak než TTL. Proto je nutné TTL úrovně z MCU převést na úrovně RS232 při odesílání znaků do počítače a také v opačném směru při přijímání. Potřeba realizace této funkcionality je v různých aplikacích častá, proto je větší množství integrovaných obvodů, které tuto funkci umožňují. V návrhu jsem použil integrovaný obvod MAX232 [3], který je kvalitní a zároveň za dobrou cenu.

## Analogově digitální převodník

Analogově digitální převodník, aneb ADC provádí měření napětí na měřících vodičích displeje. MCU sice obsahuje jeden ADC, ale specifikace platformy požaduje 4 nezávislé ADC. Navíc, ADC v MCU může měřit maximálně do 5 V, což nepokrývá interval napětí od -10 V do +10 V, které je generováno na napěťových vodičích, a tak se může objevit na měřících vodičích.

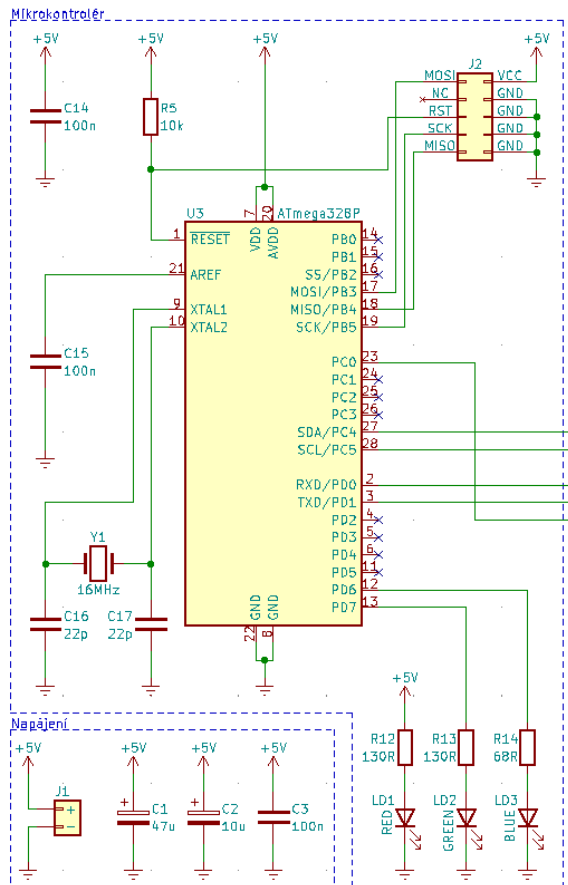
Původní záměr byl najít takový ADC, který zvládá měřit minimálně na celém generovaném rozsahu napětí. Bohužel v době návrhu takový ADC nebylo možné zakoupit či jinak získat, nicméně z experimentů s prototypem jsem měl již zkušenost, že na měřících vodičích se nikdy neobjevuje napětí větší než 1 V. A hodnoty jsou nejčastěji v desítkách mV, zřídka stovkách mV.

Následným zvažováním jsem vybral ADC MCP3424 [14], kterému sice hrozí poškození při přiložení napětí větší než 5 V na měřící vodič, ale zase dokáže měřit v rozsahu od 0 V do 2 V s přesností až 18 bitů. Vybraný ADC obsahuje 4 zabudované kanály, pomocí kterých lze měřit na 4 nezávislých měřících vodičích. Riziko zničení ADC je bohužel možné, nicméně datový list udává, že napětí mimo rozsah 0 V až 5 V je zkratováno ochrannou ESD diodou. Tudíž krátkodobý výkyv napětí nemusí ihned ADC zničit, ale pouze vrátit neplatnou hodnotu měření.

## 4.5 Elektronické schéma

Tato podkapitola se podrobně věnuje elektronickému zapojení jednotlivých částí platformy. Po částech popisuje jednotlivé logické celky schématu. Pro návrh elektronického schématu jsem použil program KiCAD, který je zdarma a zároveň skvěle funguje pro práci na projektech, jako je například tato platforma. Celé elektronické schéma se nachází v příloze C.

### Mikrokontrolér, napájení a LED diody



Obrázek 4.6: Část schématu výsledné platformy s MCU.

K pinům PD6 a PD7 jsou připojeny programovatelné LED diody LD2 a LD3 v kombinaci s ochrannými odpory R13 a R14. Tyto LED diody doplňuje navíc LED dioda LD1 s ochranným odporem R12, která je připojena přímo na napájecí napětí a na výsledné DPS platformy signalizuje aktivní napájecí napětí. Tato část schématu také obsahuje napájecí konektor J1 a stabilizačními kondenzátory napájecího napětí C1, C2 a C3.

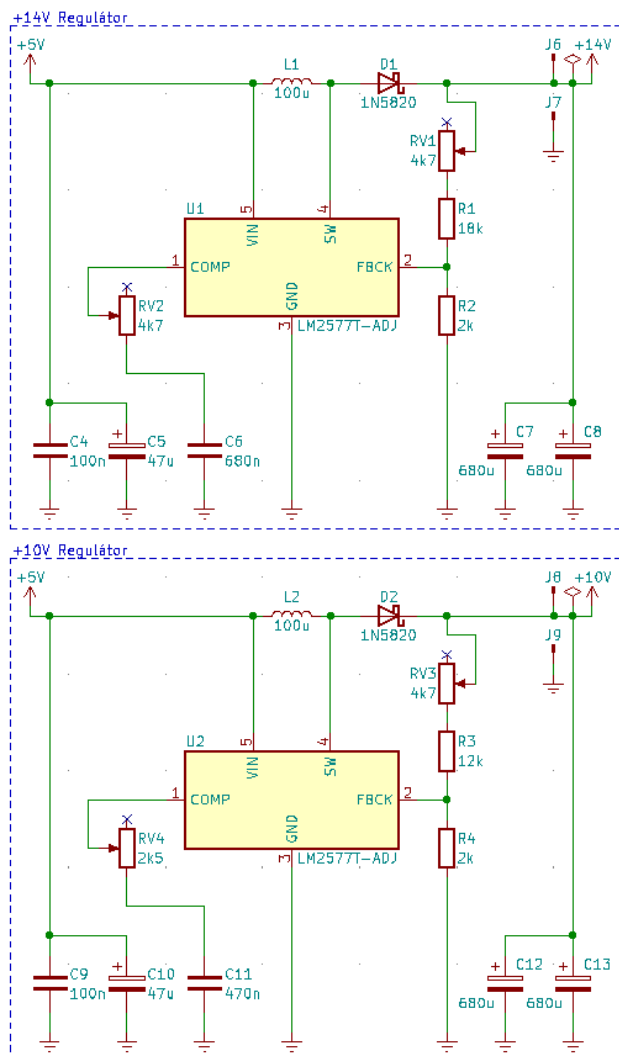
Na obrázku 4.6 je část schématu obsahující zapojení MCU, napájení a LED diod. Hlavní komponentou této části je MCU U3 ATmega328P. Pomocí tří vodičů sběrnice SPI je k MCU připojen ISP konektor J2, konkrétně v pinům PB3, PB4 a PB5. ISP musí být dále připojeno k pinu  $\overline{\text{RESET}}$ . Připojení doplňuje externí pull-up rezistor R5 s hodnotou  $10\text{k}\Omega$ . Hodnota může být dle datového listu MCU i vyšší, ale i takto je proud odporem při aktivovaném resetu zanedbatelný.

MCU doplňuje stabilizační kondenzátor C14 s hodnotou  $100\text{nF}$ , která je běžně používaná pro tyto účely, protože odfiltruje většinu nežádoucího šumu, nebo výkyvů napětí. K pinu AREF je dobrým zvykem, a datový list to také doporučuje, připojit také stabilizační kondenzátor C15 opět s hodnotou  $100\text{nF}$ .

K pinům XTAL1 a XTAL2 je připojen externí krystal Y1 v kombinaci s kondenzátory C16 a C17. Zapojení včetně hodnot kondenzátorů specifikuje datový list MCU. Pomocí externího krystalu může MCU pracovat na frekvenci 16 MHz a nemusí používat nedostatečně kalibrovaný a pomalý vnitřní oscilátor. ATmega328P podporuje frekvenci až 20 MHz, a tak je frekvence 16 MHz kompromis mezi výkonem a zároveň spolehlivostí.



## Napěťové zdroje



Obrázek 4.7: Část schématu výsledné platformy s napěťovými zdroji.

mohlo být mírně odchyleno od +14 V resp. +10 V, z důvodů odchylených hodnot rezistorů. Pomocí potenciometru RV1 resp. RV3 lze tuto odchylku výstupního napětí vykompenzovat a nastavit výstupní napětí na přesných +14 V resp. +10 V. Nastavitelné napětí má další výhodu případného vykompenzování odchylky výstupního napětí, z důvodu stárnutí součástek, či jiných okolních vlivů. Přepočtené intervaly nastavitelných napětí ukazuje tabulka 4.1.

Výst. napětí	Min. napětí	Max. napětí	Max - Min
+14 V	12.3 V	15.2 V	2.9 V
+10 V	8.6 V	11.5 V	2.9 V

Tabulka 4.1: Variabilita generovaných napětí pomocí napěťových zdrojů ve výsledné platformě.

Zapojení napěťových zdrojů je na obrázku 4.7. Část schématu obsahuje oba zdroje, jak pro +10 V, tak i +14 V. Zapojení obou zdrojů je identické, ale liší se v hodnotách některých součástek. První zdroj s výstupním napětím +14 V je dimenzován na maximální výstupní proud 600 mA, druhý zdroj pro +10 V je dimenzován pro maximální výstupní proud 800 mA.

Hlavní součástí zdroje je regulátor U1 resp. U2. K jeho pinu FBCK je připojen napěťový dělič z rezistorů R1 a R2 resp. R3 a R4 a potenciometru RV1 resp. RV3 definující výstupní napětí. Potenciometr RV1 resp. RV3 doplňující rezistor R1 resp. R3 upravuje jeho efektivní hodnotu a umožňuje tak nastavitelnost výstupního napětí.

Datový list regulátoru definuje výstupní napětí jako funkci hodnot rezistorů R1 a R2 resp. R3 a R4, za předpokladu že hodnota R1 resp. R3 je uvažována jako součet R1 a RV1 resp. R3 a RV3. Pak je vzorec pro výpočet výstupního napětí následující:

$$V_{OUT} = 1.23 * (1 + R1/R2)$$

Nastavitelnost výstupního napětí je nutná, protože každá reálná elektronická součástka má svoje efektivní parametry pouze v určitém intervalu tolerance a nikoliv jako přesnou hodnotu. A proto by výsledné napětí zdroje

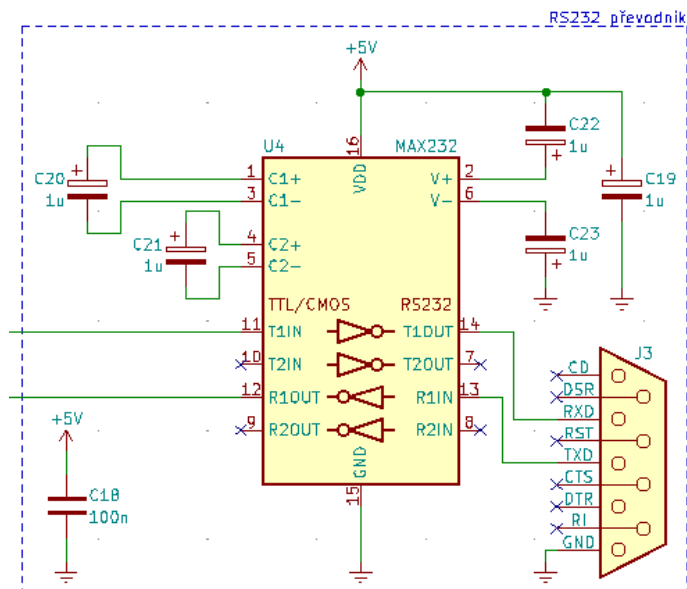
Další potenciometr RV2 resp. RV4 zapojený s kondenzátorem C6 resp. C11 k pinu COMP vytváří takzvanou kompenzační síť. Jejich hodnoty definuje datový list regulátoru a tato síť ovlivňuje zvlnění výstupního napětí při skokovém odpojení/připojení zátěže ke zdroji. Datový list doporučuje tuto síť ponechat nastavitelnou a posléze zkonstruovaný zdroj nastavit na ideální odezvu na změnu zátěže, přímo na desce.

Kondenzátory C4, C5, C7 a C8 resp. C9, C10, C12 a C13 jsou vstupní a výstupní stabilizační kondenzátory, jejichž hodnoty udává výpočet uvedený v datového listu regulátoru. Výstupní kondenzátory jsou zapojeny ve dvojici paralelně se stejnou hodnotou z důvodu snížení výstupního ESR, jelikož datový list definuje maximální ESR pro správnou funkci regulátoru. Celý napěťový zdroj doplňuje cívka L1 resp. L2 a Schottkyho dioda D1 resp. D2, zapojeny dle datového listu tak, jak tomu obvykle bývá z step-up regulátorů.

Jelikož bude po výrobě desky nutné nastavit potenciometry při ladění výstupního napětí, zdroje obsahují navíc konektory J6 a J7 resp. J8 a J9 pro pohodlného připojení voltmetru či osciloskopu pro naladění výstupního napětí a odezvy při změně zátěže.

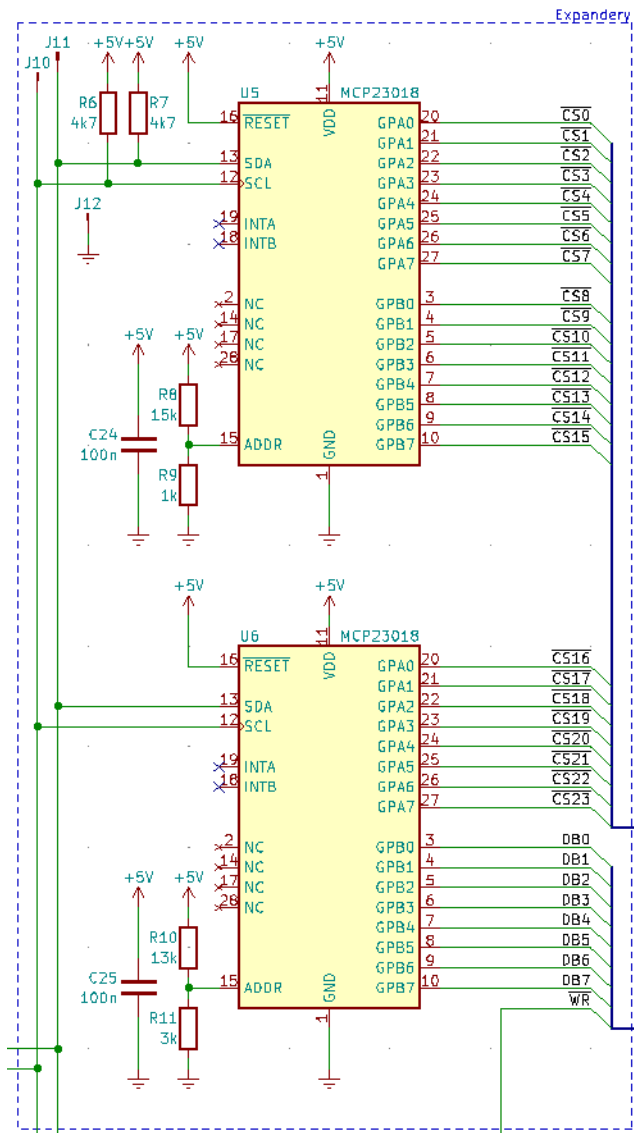
## Převodník RS232 úrovní

Část schématu s převodníkem úrovní RS232 je na obrázku 4.8. Hlavní komponentou je převodník U4 MAX232. Kondenzátory C19 až 23 jsou externí kondenzátory, jejichž zapojení a hodnoty přesně specifikuje datový list převodníku. Kondenzátor C18 plní funkci stabilizačního kondenzátoru pro integrovaný obvod s obvyklou hodnotou 100nF. Převodník poskytuje 2 přijímače a 2 vysílače. V návrhu je využit pouze jeden přijímač a jeden vysílač, zbylé jsou ponechány nezapojené. TTL strana vysílače a přijímače je připojena dál k MCU. RS232 strana vysílače a přijímače je připojena ke konektoru DB9 J3. Z konektoru jsou využity pouze piny RX, TX a GND, ostatní jsou pro tuto aplikaci nepodstatné.



Obrázek 4.8: Část schématu výsledné platformy s převodníkem RS232.

## GPIO expandy



Obrázek 4.9: Část schématu výsledné platformy s GPIO expandy.

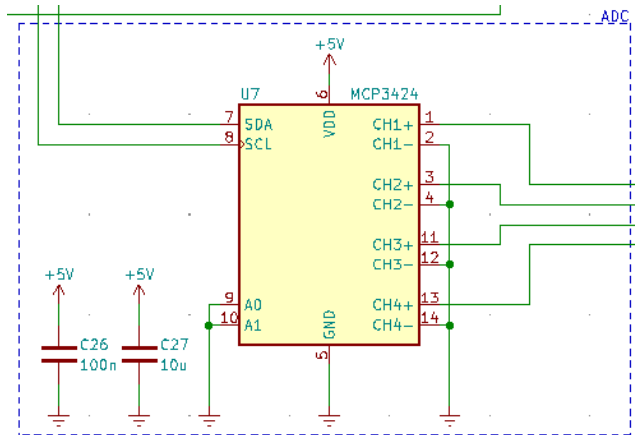
Jelikož expandy komunikují pomocí sběrnice I2C jako slave zařízení, vyžadují nastavení části své I2C adresy. Expander umožňuje nastavení jedné z 8 adres pomocí pouze jednoho pinu ADDR. Mimochodem, lze tedy na jedné I2C sběrnici připojit až 8 těchto expanderů. Adresa zařízení je nastavena podle napětí na pinu ADDR, proto je k tomuto pinu připojen napěťový dělič z rezistorů R8 a R9 resp. R10 a R11. Poměr hodnot těchto rezistorů definuje datový list pro všech 8 adres. Kombinace hodnot rezistorů 15 k $\Omega$  a 1 k $\Omega$  určuje první adresu, kombinace 13 k $\Omega$  a 3 k $\Omega$  definuje druhou adresu atd.

Část schématu na obrázku 4.9 obsahuje zapojení GPIO expanderů. Hlavní částí jsou dva expandy U5 a U6 MCP23018. Expandy jsou 16-bitové a dohromady poskytují 32 volných GPIO. 24 z těchto GPIO jsou využity pro  $\overline{CS}$  signály pro DAC. Zbýlých 8 posléze pro samotnou digitální hodnotu, která je po výběru DAC pomocí  $\overline{CS}$  signálního vodiče a aktivací pomocí signálního vodiče  $\overline{WR}$  nahrána do DAC. Signální vodič  $\overline{WR}$  je vyveden přímo z MCU a doplňuje sběrnici v které je všech těchto 33 signálních vodičů sdruženo.

$\overline{RESET}$  pin obou expanderů je připojen přímo na 5 V, protože resetování expanderů není uvažováno. Expandy jsou k MCU připojeny pomocí sběrnice I2C, která vyžaduje pull-up rezistory R6 a R7. Jejich hodnota je doporučená dle standardu I2C. Pull-up rezistory jsou nutné externí, interní pull-up rezistory v MCU jsou příliš velké a není možné je použít pro standardní rychlosti I2C, jako třeba 100 kHz nebo 400 kHz. V rámci vývoje jsou k sběrnici připojeny konektory J10, J11 a J12. Jejich účel je pohodlnější připojení sond osciloskopu ke sběrnici v rámci vývoje firmwaru.

Piny expanderů INTA a INTB umožňují generování signálu při změně stavů GPIO. Tato funkce není využita a tyto piny jsou ponechány nezapojené. Kondenzátory C24 a C25 jsou stabilizační kondenzátory pro integrovaný obvod se standardní hodnotou.

## Analogově digitální převodník

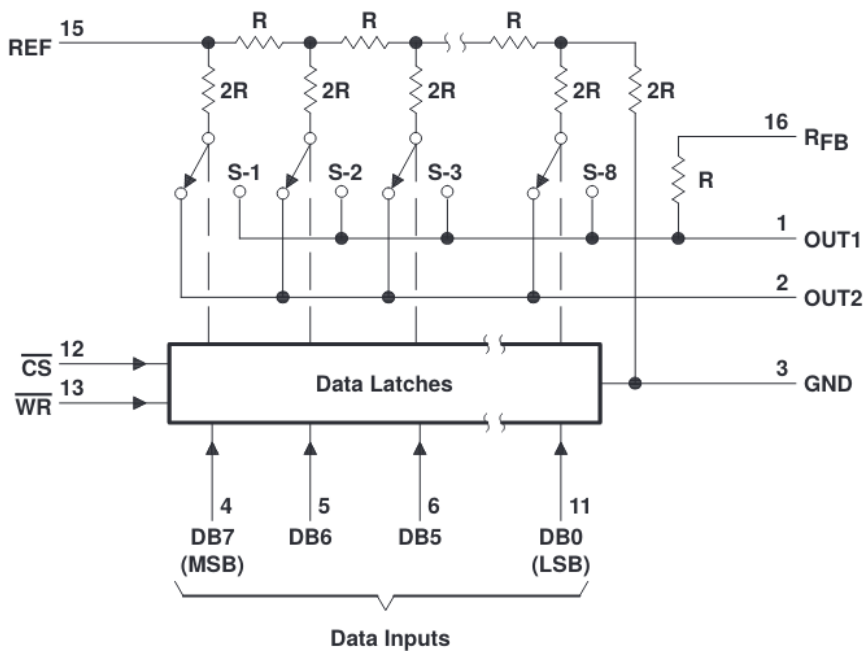


Obrázek 4.10: Část schématu výsledné platformy s ADC.

Část schématu s ADC je na obrázku 4.10. Obsahuje pouze samotný ADC U7 MCP3424 a dva stabilizační kondenzátory C26 a C27, C26 s obvyklou hodnotou. A Kondenzátor C27 doplňuje hodnotou  $10\mu\text{F}$  obvyklý C26, kvůli doporučení z datového listu ADC pro kvalitnější měření analogové hodnoty.

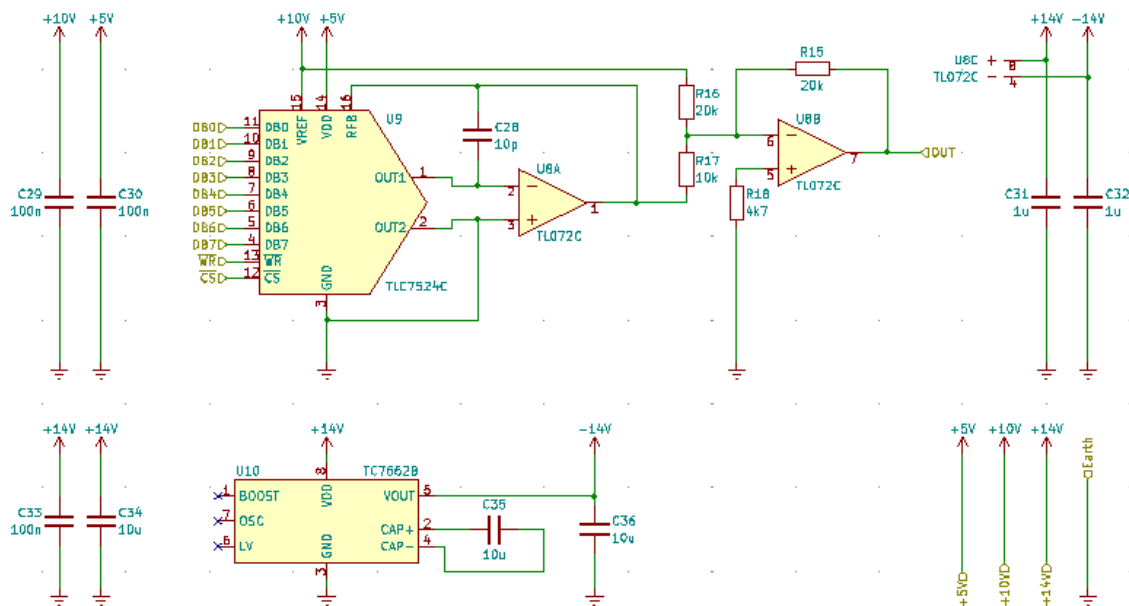
ADC je k MCU připojen pomocí sběrnice I2C a konfigurovatelná část adresy I2C je nastavena pomocí pinů A0 a A1 na hodnotu první adresy. K pozitivním pinům kanálů ADC CH1+ až CH4+ jsou následně připojeny měřící vodiče a negativní piny kanálů ADC CH1- až CH4- jsou uzemněny, protože měření probíhá relativně vůči 0 V.

## Digitálně analogové převodníky



Obrázek 4.11: Blokové schéma vnitřního zapojení DAC TLC7524C, převzato z [19].

Schéma zapojení s jedním DAC je na obrázku 4.12. Toto schéma se opakuje v návrhu platformy 24x a každé je připojeno ke sběrnici vedoucí z GPIO expanderů. Jádrem celého zapojení je samotný DAC U9 TLC7524C. Vnitřní zapojení DAC je často používaný R-R2 rezistorový žebřík. Detailnější schéma vnitřního zapojení z datového listu DAC je na obrázku 4.11.



Obrázek 4.12: Část schématu výsledné platformy s DAC.

DAC je napájený 5 V, avšak na pin VREF je připojeno 10 V jako maximální referenční analogové napětí. Datové vodiče DB0 až DB7 a signální vodiče  $\overline{WR}$  a  $\overline{CS}$  jsou k DAC připojeny ze sběrnice vedoucí od expanderů.

Datový list DAC jasně definuje požadované zapojení DAC v kombinaci s operačními zesilovači pro získání výstupního napětového rozsahu od  $-VREF$  po  $VREF$ . První OZ U8A zajišťuje zpětnou vazbu do DAC a vytváří výstupní napětí v intervalu od 0 V po  $-VREF$ . Kondenzátor C28 vychází z doporučení datového listu pro filtraci šumu. Druhý připojený OZ U8B s připojenými rezistory R15 až R18 převádí výstupní napětí z prvního OZ na požadovaný rozsah výstupního napětí od  $-VREF$  po  $VREF$ . Schématická značka U8C je pouhým připojením napájení operačních zesilovačů na +14 V a -14 V.

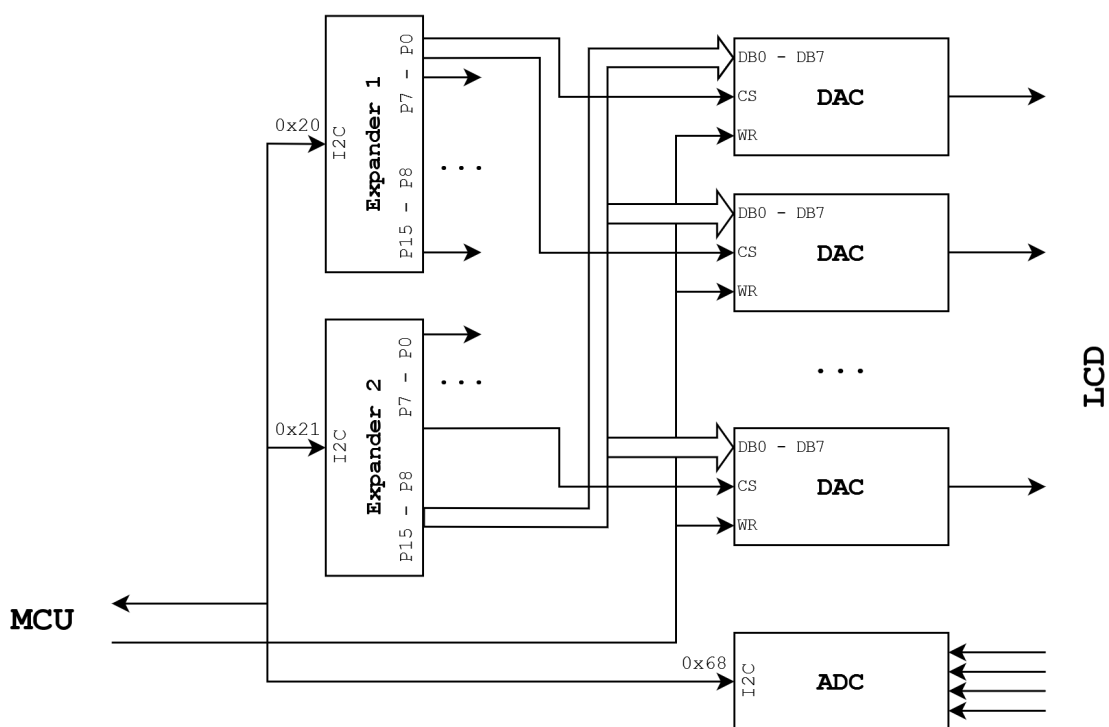
-14 V je generováno v celé platformě 24x u každého DAC zvlášť pomocí invertoru napětí U10 TC7662B. Napájecí větev -14 V propojuje výstupy všech invertorů napětí, čímž je docíleno snížení vnitřního odporu invertorů pomocí paralelního zapojení. Invertor vyžaduje k správné funkci dva externí kondenzátory C35 a C36, jejichž hodnotu definuje datový list.

Ostatní zbylé kondenzátory stabilizují napětí v rámci celého zapojení DAC s kombinací OZ. C33 a C34 stabilizují vstupních +14 V před invertorem napětí. C31 a C32 stabilizují napájecí napětí operačních zesilovačů, tedy +14 V a -14 V. A C29 a C30 stabilizují napájecí a referenční napětí pro DAC.

## Zapojení GPIO expanderů, DAC, ADC a sběrnice I2C

Schéma na obrázku 4.13 shrnuje popisované zapojení sběrnice I2C včetně připojených DAC skrz expanderu. Signální vodič  $\overline{WR}$  je připojen přímo z MCU a je sdílený všemi DAC. Datové vodiče připojené z horním 8 bitů druhého expanderu jsou také sdíleny všemi DAC. Všech 16 bitů prvního expanderu a spodních 8 bitů druhého expanderu tvoří množinu signálních vodičů  $\overline{CS}$ , kde každý je připojen k přidělenému DAC jednotlivě.

DAC reflektuje hodnotu na datových vodičích pouze pokud jsou oba signální vodiče  $\overline{WR}$  a  $\overline{CS}$  v log. 0. Je očividné, že při nahrávání rozdílných hodnot do více DAC je nutné postupovat jednotlivě. Nicméně zapojení umožňuje nahrávat stejnou hodnotu do více DAC najednou.



Obrázek 4.13: Zapojení GPIO expanderů, DAC, ADC a sběrnice I2C.

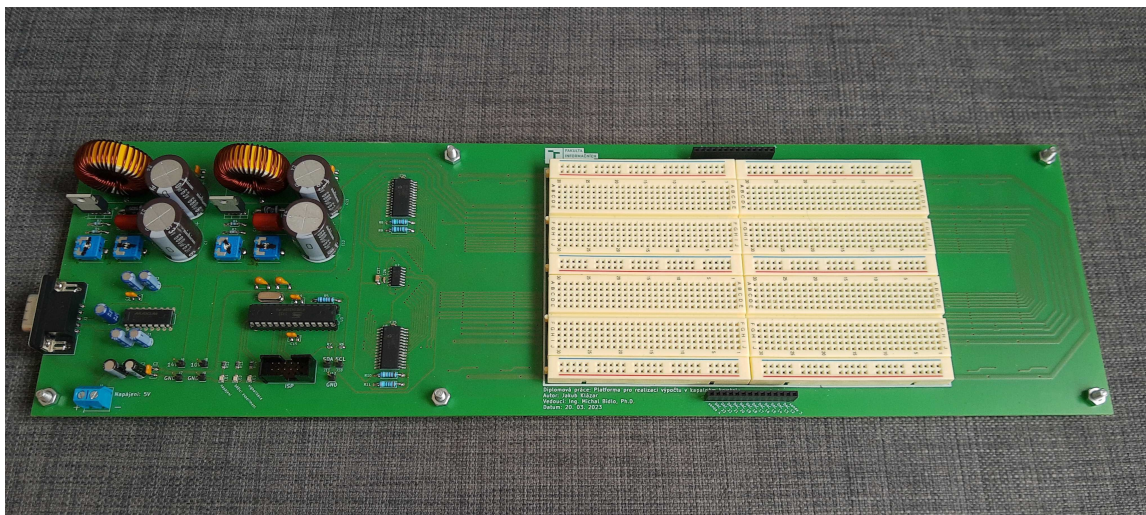


## 4.6 Deska tištěných spojů

Po finalizaci elektronického schéma bylo možné přistoupit k samotnému návrhu DPS. Tato kapitola obsahuje zajímavé pasáže fyzické realizace výsledné platformy. Samotný návrh DPS je převážně záležitostí designu, proto se kapitola procesu samotného kreslení DPS nevěnuje nijak do hloubky.

Jelikož jsem k návrhu schématu použil KiCAD, pokračoval jsem v tomto programu i nadále v návrhu samotné DPS. Cílem úspěšného návrhu DPS je samozřejmě umístění všech součástek na desku takovým způsobem, aby se minimalizovala velikost desky, ale také délka a křížení propojů. Pečlivé promyšlení rozmístění součástek je dobrým předpokladem kvalitního návrhu DPS.

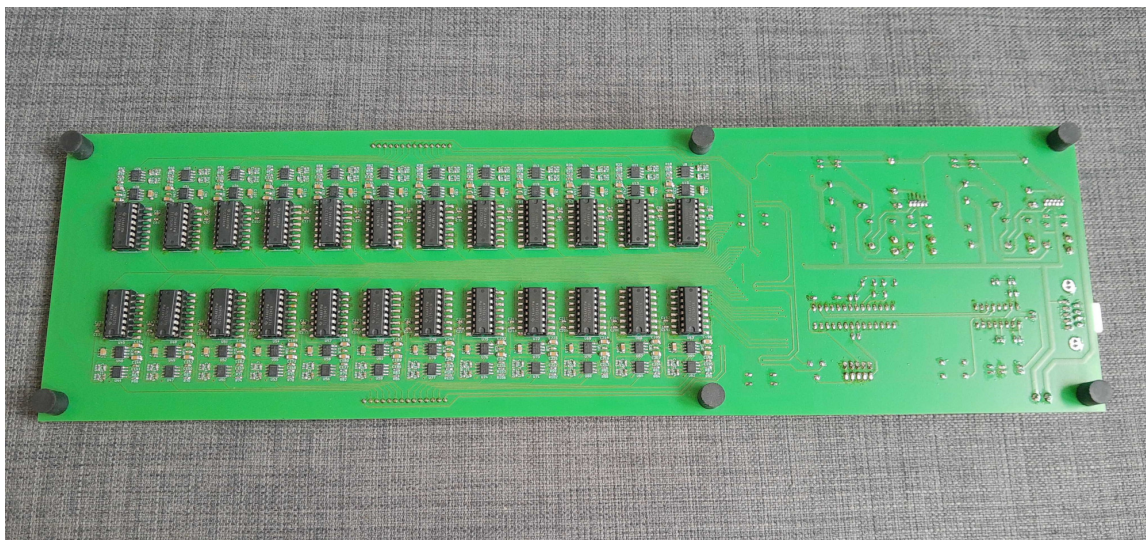
Při návrhu rozmístění jsem vycházel z členění schématu na logické celky, z kterých lze přepokládat, které součástky je vhodné umístit blíže u sebe. Zvláštností DPS této platformy je nepájivé pole, které plní požadavek na vyměnitelnost experimentálního displeje/zařízení. Nepájivé pole je také největší součástkou na samotné DPS platformy. Strana DPS přímo pod nepájivým polem nelze použít pro žádné umístění součástek, protože je touto plochou přilepené nepájivé pole k samotné desce.



Obrázek 4.14: Pohled shora na výslednou DPS platformy.

Zkonstruovaná platforma je na obrázcích 4.14 a 4.15. V levé horní části DPS jsou oba napěťové zdroje pro 10 V a 14 V. Hned pod zdroji je MCU s převodníkem pro RS232. V levé spodní části platformy jsou také programovatelné LED diody, vyvedené konektory pro měření a ladění napěťových zdrojů a I2C, konektor pro ISP a na kraji se nachází konektor seriové linky DB9.

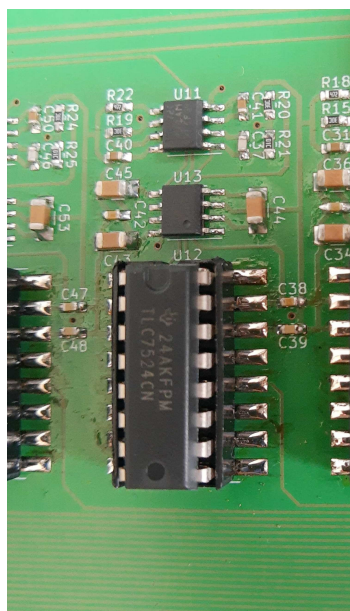
Přibližně v půlce desky mají své místo expander a ADC. Vodiče z expanderů dále postupují do pravé části desky která náleží 24 DAC a nepájivému poli. Nepájivé pole je na horní straně a DAC jsou na spodní straně DPS. Nad a pod nepájivým polem jsou kolektory pro připojení napěťových a měřících vodičů k nepájivému poli, do kterého je možné vložit experimentální displej.



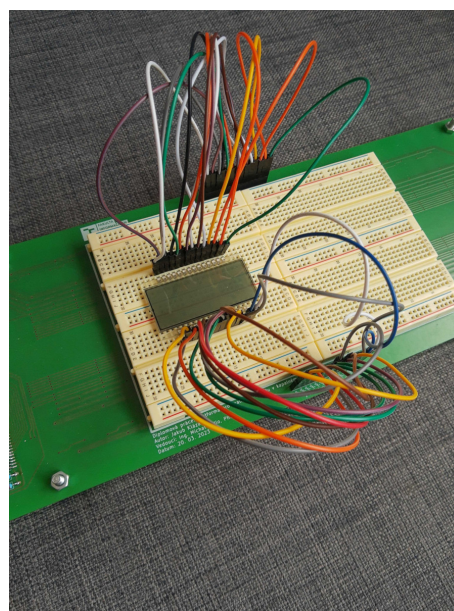
Obrázek 4.15: Pohled zespoda na výslednou DPS platformy.

Na obrázku 4.16a je detail individuálního DAC zapojený s operačními zesilovači a invertorem napětí. I přesto, že jsem se snažil prostorové nároky na DPS minimalizovat, tak DAC zabírají výraznou část celé DPS.

Na obrázku 4.16b je zapojený experimentální displej pomocí propojovacích drátů ke konektoru napěťových a měřicích vodičů. Návrh DPS s nepájivým polem dává platformě volnost v připojování a přepojování libovolných napěťových a měřicích vodičů k experimentálnímu displeji/zařízení.



(a) Individuální DAC na DPS platformy.



(b) Zapojený experimentální displej v platformě.

Obrázek 4.16: Detailní záběry na vybrané části DPS platformy.



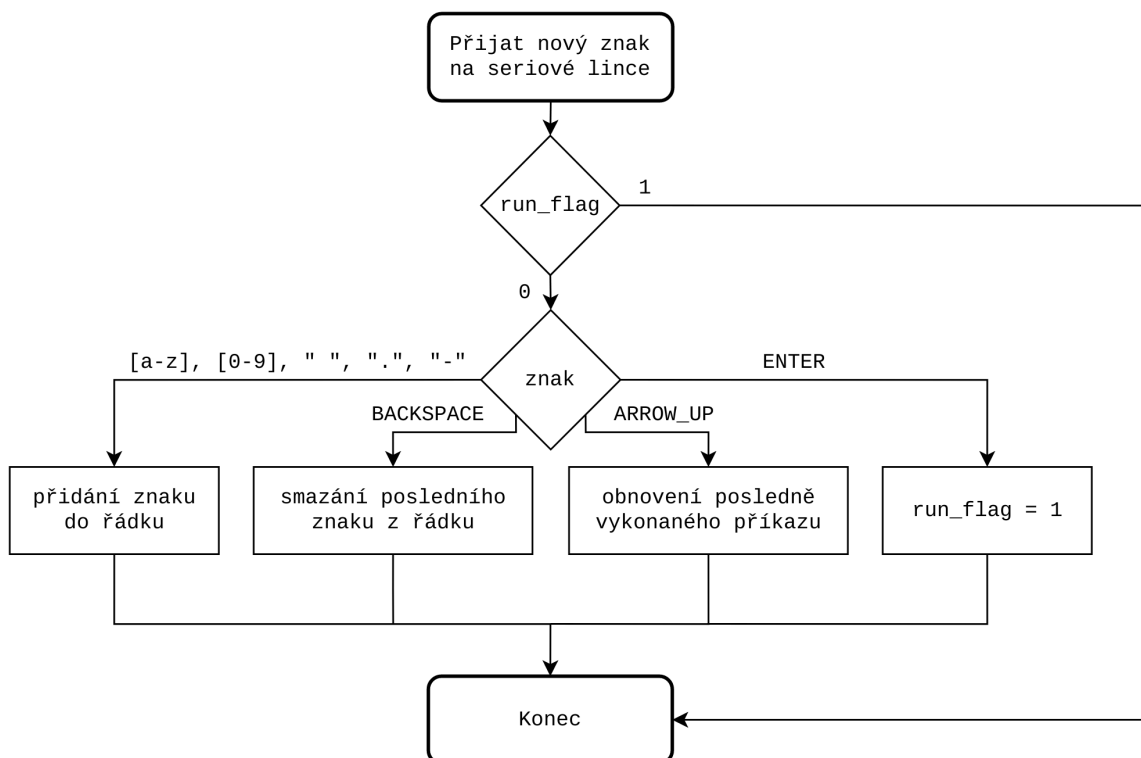
## 4.7 Firmware platformy

Tato podkapitola se věnuje vybraným částem implementace firmwaru platformy, tedy programu běžícím přímo v MCU na desce platformy. Program je napsaný v jazyce C s využitím knihovny `avr-libc` a je kompilován pomocí odnože kompilátoru `gcc`, tedy `avr-gcc`. Cílem této kapitoly není diskuze problematiky programování mikrokontrolérů, ale ukázka vybraných částí implementace, které jsou relevantní k procesu návrhu podobných aplikací.

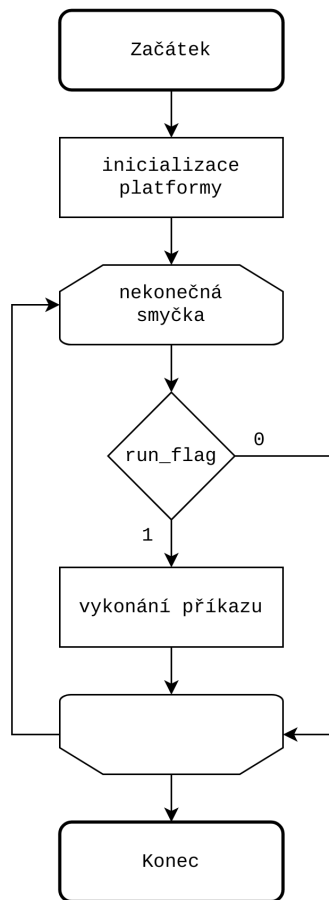
Hlavní a zároveň jedinou funkcí firmwaru, vycházející ze specifikace požadavků, je abstrahovat ovládání DAC a ADC na desce platformy a vytvořit tak rozhraní pro pohodlné ovládání experimentů nad připojeným zařízením v platformě. Na tomto předpokladu staví koncept celého firmwaru platformy, který pracuje jako minimalistický terminál. Naslouchá na sériové lince a přijímá příkazy zpravidla o nastavení zadaného napětí na patřičném DAC, nebo vrácení měřeného napětí z vybraného ADC.

Po připojení k platformě sériovou linkou pomocí programu, který umožňuje přijímat a odesílat znaky, lze pak s platformou pohodlně komunikovat. Pro vykonávání skriptovaných experimentů umožňuje platforma přepnutí do tichého módu bez vypisování promptu terminálu platformy. Tento mód usnadňuje načítání vrácených výsledků z platformy a automatizaci celého procesu experimentu.

### Základní struktura firmwaru



Obrázek 4.17: Vývojový diagram obsluhy přerušení při přijmutí nového znaku na seriové lince.



Obrázek 4.18: Vývojový diagram hlavní smyčky firmwaru.

vytváření interaktivní dojem. Do vnitřního bufferu pro aktuálně zadávaný příkaz jsou přidávány nebo odmazávány znaky tak, jak činí uživatel pomocí zadávání tisknutelných znaků, nebo klávesy BACKSPACE.

Pokud uživatel stiskne klávesu ENTER a tím odešle znak CR, pak obsluha přerušení nastaví příznak `run_flag` na hodnotu 1. Tento příznak dočasně zablokuje přijímání nových znaků a spustí se vykonávání příkazu v hlavní smyčce, jejíž vývojový diagram je na obrázku 4.18. Dle řetězce tisknutelných znaků v bufferu příkazu se rozhodne, zda jde o platný příkaz včetně argumentů a následně se vykoná, případně se vypíše chybová hláška. Po skončení vykonávání příkazu se nastaví příznak zpět na hodnotu 0, tím se opět aktivuje zpracovávání přijatých znaků a hlavní smyčka začne opět čekat na potvrzení příkazu v nekonečné smyčce. Tento proces se stále opakuje.

Vývojový diagram zjednodušuje některé aspekty implementace. Totiž znak `ARROW_UP` je reprezentován jako escape sekvence tří znaků. Proces zpracování právě tohoto znaku obsahuje malý stavový automat pro korektní zpracování escape sekvencí, konkrétně `ARROW_UP`. Pro funkci obnovení posledního vykonaného příkazu obsahuje bufferu příkazu příznak `valid_flag`. Tento příznak určuje, zda je příkaz v bufferu platný či nikoliv. Při zapisování příkazu do bufferu je tento příkaz nastaven na hodnotu 1, po vykonání příkazu je příkaz v bufferu ponechán, ale příznak nastaven na hodnotu 0. Pokud je následující přijatý znak terminálem `ARROW_UP`, pak je příznak opět nastaven na hodnotu 1 a nadále je s bufferem příkazu nakládáno jako obvykle. Naopak pokud je přijat jakýkoliv jiný znak, pak je příznak také nastaven na hodnotu 1, ale buffer je přepsán novým tisknutelným znakem anebo pouze vyprázdněn (`ENTER`, `BACKSPACE`). Bohužel na větší historii posledních vykonaných příkazů nepostačuje SRAM mikrokontroléru.

Jelikož se platforma chová jako terminál, vychází z tohoto celý způsob návrhu implementace. Hlavní schopností platformy je implementace minimalistického interaktivního terminálu se základní sadou příkazů, umožňující pohodlné ovládání experimentů s platformou. Terminál platformy tak nabízí základní uživatelské prostředí podobné například běžným unixovým shellům.

Firmware využívá periferie `USART`, kterou nabízí MCU pro přijímání a odesílání znaků po sériové lince. `USART` umožňuje aktivaci přerušení při přijetí nového znaku odeslaného od uživatele. Zjednodušený vývojový diagram obsluhy tohoto přerušení je na obrázku 4.17, kde dochází k zpracování nově přijatých znaků. Jiné znaky mimo uvedené v tomto diagramu jsou ignorovány.

Terminál platformy si ukládá nově přijaté tisknutelné znaky do vnitřního bufferu aktuálního příkazu. Zároveň při přijetí tisknutelného znaku platforma tento znak opět odesílá zpět uživateli, čímž terminál vytváří interaktivní dojem. Do vnitřního bufferu pro aktuálně zadávaný příkaz jsou přidávány nebo odmazávány znaky tak, jak činí uživatel pomocí zadávání tisknutelných znaků, nebo klávesy BACKSPACE.

Pokud uživatel stiskne klávesu ENTER a tím odešle znak CR, pak obsluha přerušení nastaví příznak `run_flag` na hodnotu 1. Tento příznak dočasně zablokuje přijímání nových znaků a spustí se vykonávání příkazu v hlavní smyčce, jejíž vývojový diagram je na obrázku 4.18. Dle řetězce tisknutelných znaků v bufferu příkazu se rozhodne, zda jde o platný příkaz včetně argumentů a následně se vykoná, případně se vypíše chybová hláška. Po skončení vykonávání příkazu se nastaví příznak zpět na hodnotu 0, tím se opět aktivuje zpracovávání přijatých znaků a hlavní smyčka začne opět čekat na potvrzení příkazu v nekonečné smyčce. Tento proces se stále opakuje.

## Podporované příkazy terminálu

Tato část textu shrnuje formát a funkci všech podporovaných příkazů, které platforma přijímá a následně dokáže vykonat. Formát příkazů je následující: Každý příkaz začíná identifikačním slovem, které následuje jedna nebo více mezer a posléze případné parametry. Jednotlivé parametry jsou opět odděleny pomocí jedné nebo více mezer. Mezery vyskytující se na začátku příkazu, tak i na konci, jsou ignorovány. Parametry příkazů jsou typu: slovo upravující funkci příkazu, celé číslo nebo desetinné číslo. Terminál striktně kontroluje typ a počet argumentů. Podporované příkazy jsou následující:

### **help**

Příkaz nepřijímá žádné parametry a vypíše stručnou nápovědu se seznamem příkazů, jejich parametry a funkcí.

### **clear**

Příkaz nepřijímá žádné parametry a vypíše na terminál escape sekvenci pro vyčištění obrazovky terminálu.

### **verb <verbosity>**

Příkaz mění mód terminálu mezi interaktivním a tichým. Přepnutím terminálu do tichého módu je doporučeno použít při spouštění skriptovaných experimentů, protože terminál ihned po spuštění příkazu **verb 0** přestane vypisovat prompt. Přestane vracet přijaté znaky a odesílá uživateli jen a pouze výsledky příkazů nebo případně chybová hlášení. Terminál lze zpět přepnout terminál do interaktivního módu pomocí zadání příkazu **verb 1**.

<verbosity> - Číslo 0 přepíná terminál do tichého módu, číslo 1 přepíná terminál do interaktivního módu.

### **volt set <pin> <voltage>**

Příkaz nastaví stanovené napětí na zadaný DAC vodič. První parametr **set** pouze odlišuje druh funkce příkazu **volt** od příkazu **volt clear**.

<pin> - Celé číslo v intervalu od 0 do 23 stanovující na jaký DAC vodič bude napětí nastaveno.

<voltage> - Desetinné číslo v intervalu od -10.0 do 10.0 určující jaké napětí nastavit (ve voltech). Terminál podporuje zápis napětí na maximálně čtyři desetinná místa.

### **volt clear**

Příkaz nastaví na všech DAC vodičích napětí 0 V. Jediným parametrem příkazu je slovo **clear**, které odlišuje funkci příkazu **volt** od příkazu **volt set**.

### **scan <pin>**

Příkaz získá hodnotu ze specifikovaného ADC a vrátí změřené napětí jako celé číslo v mikrovoltech.

<pin> - Celé číslo v intervalu od 0 do 3 stanovující, z jakého ADC vodiče bude napětí měřeno.

### **exp <voltages> <hold\_time>**

Příkaz spustí experiment, který spočívá v nastavení všech DAC vodičů na požadovaná napětí, poté vyčkání stanoveného času a následně změření napětových hodnot na všech ADC vodičích. Hodnoty z ADC jsou v mikrovoltech a při vypsání jsou odděleny čárkou.

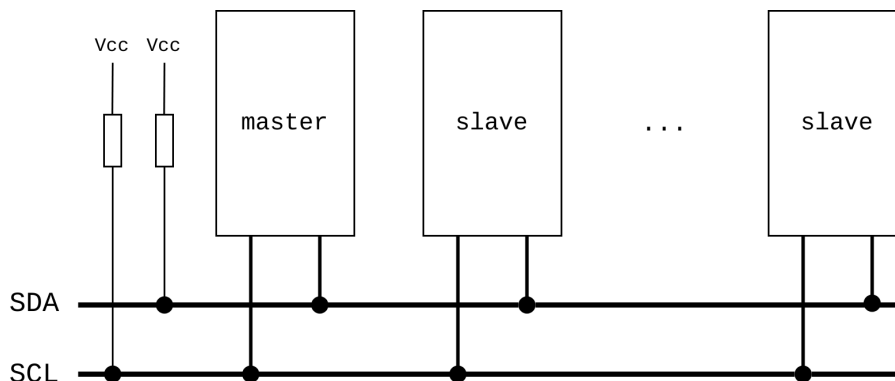
<voltages> - Seznam 24 desetinných čísel stanovující napětí na všech DAC vodičích, hodnoty musí být v intervalu od -10.0 do 10.0 a odděleny jednou nebo více mezer.

<hold\_time> - Celé číslo v intervalu od 0 do 10000 (v mikro sekundách), definující časový interval mezi nastavením napětí na DAC vodičích a měřením napětí na ADC vodičích.

## Sběrnice I2C

Další části této podkapitoly se věnují samotné komunikaci MCU s DAC a ADC, která je nutná pro vykonání příkazů, jako třeba `volt set` nebo `scan`. Jelikož komunikace probíhá po sběrnici I2C, následujících pár odstavců se věnuje základním pojmům a principům této sběrnice.

Zařízení jsou na této sběrnici propojeny pomocí dvou vodičů SDA a SCL, tak jak ukazuje obrázek 4.19. Vodič SDA slouží pro přenos dat a vodič SCL pro přenos synchronizačního hodinového signálu. Zařízení spínají vodiče pomocí open-drain zapojení, proto je nutné pro každou I2C sběrnici připojit jednotlivý pull-up rezistor pro oba vodiče sběrnice.

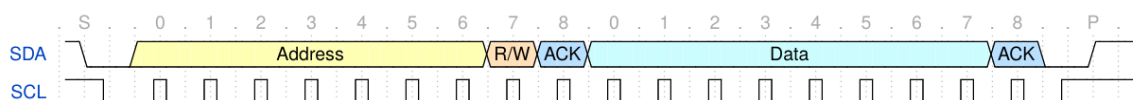


Obrázek 4.19: Zapojení zařízení na sběrnici I2C.

I2C rozlišuje dva druhy zařízení `master` a `slave`. Zařízení, které je `master` iniciuje/ukončuje komunikaci a generuje hodinový signál. Pokud chce `master` zahájit komunikaci se `slave` zařízením, musí specifikovat adresu daného `slave` zařízení. Standard I2C umožňuje připojení více zařízení `master`, nicméně v této práci je tato možnost nepodstatná, tak jí nadále není věnována pozornost a je uvažován vždy pouze jeden `master` na sběrnici, konkrétně MCU.

Obyčejný průběh komunikace je na obrázku 4.20. `Master` iniciuje komunikaci pomocí `START` podmínky a následně posílá adresový bajt. Ten se skládá z 7-bitové adresy `slave` zařízení s kterým chce komunikovat a `R/W` bitu, který určuje jestli bude `master` odesílat nebo přijímat data od `slave` zařízení.

Každý bajt v komunikaci následuje jeden `ACK` bit, který se uvažuje jako aktivní v log. 0 (z důvodů open-drain zapojení). `ACK` bit ovládá vždy zařízení, které právě data přijímá, tzn. pokud `master` zapisuje, generuje `ACK` bit `slave` zařízení a pokud `master` přijímá data, tak také generuje `ACK` bit. Význam `ACK` bitu je obvykle potvrzení přijatých dat. Pokud příjemce nedokáže přijmout další data měl by `ACK` bit za posledními přijatými daty ponechat v log. 1. Po adresovém bajtu následuje přenos jednoho nebo víc samotných datových bajtů komunikace. Komunikaci ukončuje `master` zařízením vygenerováním `STOP` podmínky.



Obrázek 4.20: Příklad obyčejné komunikace na sběrnici I2C.

## Komunikace s DAC

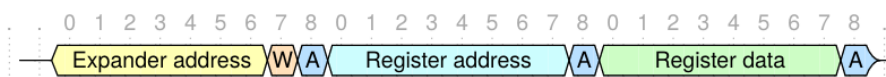
Ke komunikaci s DAC dochází při vykonávání požadavku na změnu napětí na výstupních vodičích pomocí příkazů `volt` nebo `exp`. Se samotnými DAC neprobíhá komunikace přímo po sběrnici I2C, ale skrz expandery, ke kterým jsou pak jednotlivé DAC připojeny.

MCP23018 jakožto expandery GPIO obsahují sadu vnitřních registrů jejichž hodnoty lze měnit pomocí sběrnice I2C. Hodnoty jednotlivých bitů v těchto registrech ovládají patřičnou funkci odpovídajících GPIO pinů expanderů. Registry jsou organizovány po bajtech avšak expandery MCP23018 jsou 16-bitové, proto obsahují všechny ovládací registry v párech, jeden pro horní bajt a druhý pro spodní bajt.

MCP23018 obsahují vnitřní zapojení výstupních pinů typu open-drain, proto expandery umožňují aktivaci vnitřních pull-up rezistorů, které jsou, zřejmě nutné pro správnou funkci. Expandery nabízejí celou řadu funkcí pro piny nastavené jako vstupní, nicméně v této aplikaci jsou expandery využívány pouze jako výstupní GPIO, a tak jsou v rámci implementace zásadní pouze následující registry:

- 0x00 a 0x01 pro nastavení směru GPIO pinů, 0 - výstupní pin, 1 - vstupní pin.
- 0x0C a 0x0D pro nastavení pull-up rezistorů, 0 - pull-up vypnut, 1 - pull-up zapnut.
- 0x14 a 0x15 pro nastavení výstupní hodnoty, 0 - log. 0, 1 - log. 1.

Obrázek 4.21 ukazuje průběh nastavení hodnoty jednoho registru expanderu. Proces spočívá v odeslání adresního bajtu I2C pro daný expander, následně adresy registru uvnitř expanderu, a poté samotnou požadovanou hodnotu registru. Nastavení pull-up rezistorů a směru GPIO pinů probíhá při inicializaci platformy a poté, při obvyklém běhu platformy, jsou již pouze upravovány registry 0x14 a 0x15.



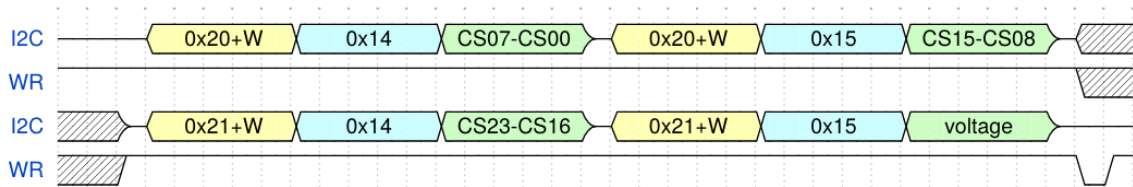
Obrázek 4.21: Příklad komunikace I2C při změně hodnoty vnitřního registru expanderu.

Vstupem DAC je 8 datových vodičů a řídicí signály  $\overline{CS}$  a  $\overline{WR}$ . DAC obsahuje vnitřní registr pro uložení hodnoty na datových vodičích, která reflektuje požadované výstupní napětí. Pokud jsou oba vodiče  $\overline{CS}$  a  $\overline{WR}$  v log. 0, pak je hodnota na datových vodičích reflektována přímo do registru DAC a následně převáděna na analogové napětí. V opačném případě v DAC zůstává uložena předchozí hodnota datových vodičů a výstupní analogové napětí zůstává neměnné. Vztah digitální hodnoty na datových vodičích DAC a výstupní analogové hodnoty ukazuje tabulka 4.2.

Digitální hodnota (DB7-DB0)	Výstupní analogové napětí
0b11111111	10 V * (127/128)
0b10000001	10 V * (1/128)
0b10000000	0 V
0b01111111	-10 V * (1/128)
0b00000001	-10 V * (127/128)
0b00000000	-10 V * (128/128)

Tabulka 4.2: Vztah vstupní digitální hodnoty DAC k výstupnímu analogovému napětí.

Proces nastavení napětí na individuálním DAC je následující: Nastaví se  $\overline{CS}$  signál patřící danému DAC pomocí obou bajtů prvního expanderu a spodního bajtu druhého expanderu. Následně je do horního bajtu druhého expanderu nahrána digitální hodnota požadovaného výstupního napětí. Po přenosu těchto dat následuje pulz log. 0 na sdíleném vodiči  $\overline{WR}$  mezi všemi DAC a tím se nová hodnota zreflektuje do vnitřního registru DAC a tím i na jeho analogový výstup. Celý tento proces ukazuje obrázek 4.22.



Obrázek 4.22: Příklad komunikace I2C pro nastavení výstupního napětí na individuálním DAC.

Příkaz `exp` nastavuje napětí všech 24 DAC jednotlivě. To znamená potřebu  $24 \cdot 4 \cdot 3 \cdot 8 = 2304$  hodinových taktů na sběrnici I2C. Při frekvenci sběrnice 100 kHz je čas potřebný k nastavení všech DAC 23 ms. Ke snížení tohoto času by mohly přispět dvě optimalizace.

První z nich by spočívala v přeskokování nastavování takových DAC, u kterých je nová hodnota identická jako předchozí. Avšak z důvodů povahy předpokládaných experimentů, které obvykle spočívají v častých změnách většiny napěťových pinů, by tato optimalizace nepřispěla zásadním způsobem.

Druhá optimalizace by minimalizovala aktualizaci registrů expanderů ovládající piny  $\overline{CS}$ , které se v příkazu `exp` mění postupně a dochází tak k zbytečnému opakovanému zápisu hodnoty `0xFF`. Jelikož terminál podporuje také příkaz `volt`, musela by se tato optimalizace umět vypořádat i s náhodnou aktualizací DAC. Proto by optimalizace vnášela do kódu další režii náchylnou na chyby a snižující tak robustnost kódu. Žádné optimalizace tedy nejsou ve firmwaru implementovány.

## Komunikace s ADC

Komunikace s ADC, narozdíl od komunikace s DAC, probíhá přímo po sběrnici I2C. Platforma umožňuje měření na 4 ADC, protože MCP3424 obsahuje 4 kanály mezi kterými lze při měření vybírat. Převod lze spustit ve dvou módech, první jednorázový mód provede jeden převod a uloží výsledek do vnitřního výstupního registru. Druhý nepřetržitý mód neustále provádí převody a aktualizuje vnitřní výstupní registr novými výsledky. MCP3424 podporuje nastavení rozlišení převodu na 12, 14, 16 nebo 18 bitů, tím je ale takové ovlivněna rychlost převodu, jak ukazuje tabulka 4.3. U měřeného signálu lze také nastavit zesílení vestavěným zesilovačem a to 1x, 2x, 4x nebo 8x.

Rozlišení převodu	Počet převodů za sekundu	Minimální rozlišení jednoho bitu
12	240	1000 $\mu\text{V}$
14	60	250 $\mu\text{V}$
16	15	62.5 $\mu\text{V}$
18	3.75	15.625 $\mu\text{V}$

Tabulka 4.3: Podporované rozlišení převodu ADC.

MCP3424 obsahuje jeden vnitřní konfigurační registr o velikosti jednoho bajtu, který obsahuje nastavení převodníku. Dále obsahuje výstupní registr, který obsahuje naměřenou hodnotu po dokončení převodu. Význam jednotlivých bitů ukazuje tabulka 4.4

7	6	5	4	3	2	1	0
$\overline{\text{RDY}}$	C1	C0	$\overline{\text{O/C}}$	S1	S0	G1	G0

Tabulka 4.4: Význam bitů konfiguračního registru ADC.

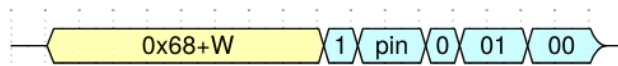
Při čtením konfiguračního registru, bit  $\overline{\text{RDY}}$  popisuje stav dat ve výstupním registru. V případě nepřetržitého módu má tento bit hodnotu log. 0 pouze pokud je ve výstupním registru nová doposud nepřechtená hodnota převodu. Pokud má bit hodnotu log. 1, pak od minulého čtení hodnoty převodu zatím další převod nebyl dokončen a běží na pozadí. V jednorázovém módu tento bit reflektuje běh převodu. Pokud má bit  $\overline{\text{RDY}}$  hodnotu log. 1, pak převod stále běží, naopak pokud má bit hodnotu log. 0, pak byl převod již dokončen a výstupní registr je aktualizovaný hodnotou posledního převodu. U nepřetržitého módu je čtením výstupního registru bit  $\overline{\text{RDY}}$  změněn na log. 1, u jednorázového módu nikoliv. Zápis log. 1 do tohoto bitu způsobí v jednorázovém módu spuštění nového převodu.

Bity C1 a C0 spolu utvářejí binární hodnotu vybraného kanálu na kterém bude MCP3424 provádět měření. Bit  $\overline{\text{O/C}}$  přepíná mezi nepřetržitým a jednorázovým módem, log. 1 pro nepřetržitý, log. 0 pro jednorázový. Bity S1 a S2 definují rozlišení převodu: 00 - 12 bitů, 01 - 14 bitů, 10 - 16 bitů a 11 - 18 bitů. A Bity G1 a G2 definují sílu vstupního zesilovače: 00 - 1x, 01 - 2x, 10 - 4x a 11 - 8x.

Platforma využívá jednorázového módu, protože převod je žádoucí vykonávat pouze při spuštění daného příkazu z terminálu platformy. Vstupní zesilovač je nastaven na hodnotu 1x, tzn. není aplikováno žádné zesílení a ADC tak může pracovat na plném napětovém rozsahu. Rozlišení převodu jsem po testování a ladění zvolil 14-bit, je to ideální kompromis mezi kvalitou přenosu a rychlostí, protože větší rozlišení nepřinášelo žádné lepší výsledky, ale pouze kvalitnější měření nepodstatného šumu.



Převod ADC je spuštěn zápisem konfiguračního registru. Data k zápisu do tohoto registru následují při komunikaci I2C ihned po adresním bajtu. Příklad spuštění převodu ukazuje obrázek 4.23

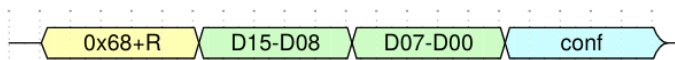


Obrázek 4.23: Příklad komunikace I2C pro spuštění měření ADC.

Proces čtení výstupního registru popisuje obrázek 4.24. ADC je adresován v čtecím módu a poté ADC odesílá MCU výstupní registr následovaný konfiguračním registrem. Výstupní registr má délku 2 bajty pro rozlišení převodu 12, 14 a 16 bitů. Pro 18 bitové rozlišené převodu má výstupní registr délku 3 bajty.

V případech rozlišení nezarovnaného na bajty (12, 14 a 18 bitů) je poslední platný bit rozlišení převodu propagován mezi vyšší bity až do MSB. Hodnota ve výstupním registru je totiž ve formátu dvojkového doplňku a pro získání výsledného napětí musí být tato hodnota násobena minimálním rozlišení jednoho bitu z tabulky 4.3.

Po bajtech výstupního registru ADC odesílá bajt konfiguračního registru, v kterém je nutné dle hodnoty bitu RDY rozhodnout, za jsou načtená data výstupního registru platná či nikoliv. Dalším čtením ADC opakuje odesílání konfiguračního registru.



Obrázek 4.24: Příklad komunikace I2C pro čtení výsledku měření z ADC.

## 4.8 Skripty experimentů

Skripty experimentů mohou být programovány v libovolné jazyce dle potřeb a uvážení experimentátora. Jediným požadavkem na programovací jazyk je schopnost odesílat a přijímat znaky po sériové lince. Já v této práci používám jazyk python pro svojí jednoduchou použitelnost a v rámci experimentů jsem připravil několik skriptů, jako ukázkou práce s platformou pomocí skriptů.

V rámci programování skriptů je silně doporučeno použít jako první příkaz platformy `verb 0` pro přepnutí terminálu do tichého módu. V rámci odeslání všech příkazů musí být řetězec příkazu vždy ukončován znakem `CR`, po odeslání příkazu platforma vrací pouze znak `CR` nebo výsledek příkazu ukončen znakem `CR`. Pokud vybraný jazyk obsahuje funkci pro přijetí řádku po sériové lince, pak je doporučeno po každém odeslaném příkazu načíst řádek, který bude prázdný u příkazů bez výstupu, nebo bude obsahovat výstup provedeného příkazu.

## Kapitola 5

# Experimentální výsledky

Tato kapitola se věnuje provedení samotných experimentů s kapalným krystalem pomocí zkonstruované platformy. Experimenty slouží především k ověření funkčnosti platformy, ale zároveň jsou výsledky porovnány s experimenty s původní platformou pana Hardinga [16][5][7][6].

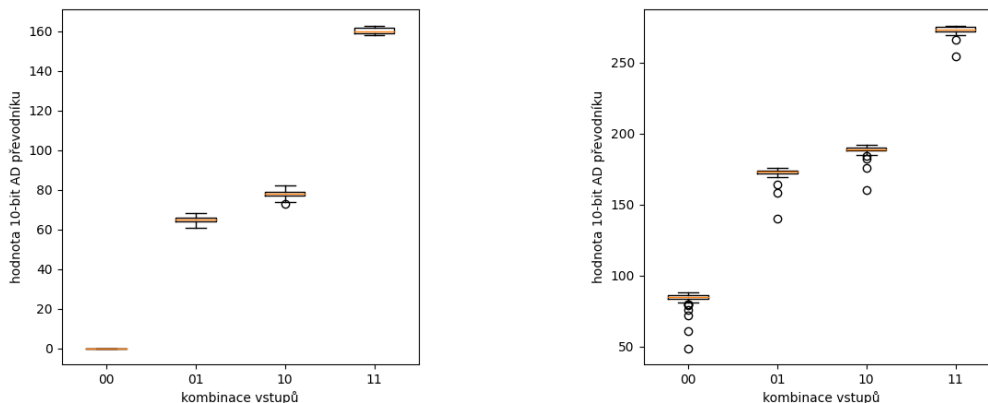
Použité experimenty spojuje jeden cíl, kterým je hledat v kapalném krystalu logické funkce, jako například AND, OR apod. Hledání takových funkcí v kapalném krystalu spočívá v nalezení takové kombinace napětí na vodičích displeje, při které, v kombinaci se zvolenými vstupními vodiči, se kapalný krystal chová jako hledané logické hradlo. Obecně lze k této problematice přistoupit nespočetně způsoby, nicméně ambicí této práce není vést pokročilejší výzkum v této problematice. Proto tato kapitola pouze povrchově shrnuje výsledky na základních experimentech.

### 5.1 Prvotní experimenty s prototypem

V rámci návrhu platformy nejprve vznikl prototyp platformy s omezenou funkcionalitou. Tento prototyp využíval pouze 8 vývodů displeje a 3 napětí, všechny konfigurace napětí displeje tak bylo možné projít hrubou silou. Prvotní experiment procházel všechny kombinace výběru vstupních vodičů a zároveň všechny kombinace napětí na ostatních vodičích pro každý jeden výběr vstupních vodičů. V takové kombinaci následně experiment vyzkoušel 4 platné logické kombinace na vstupních vodičích a zaznamenával měřené napětí na výstupním vodiči.

Naměřené výsledky prototyp odesílal do počítače po sériové lince ve formátu JSON. Jednalo se o 102060 měření a experiment běžel něco přes 6 hodin. Výsledky jsem pak následně zpracoval skriptem, který vyhodnotil, nejlepší konfigurace pro hledaná hradla. Skript hledal takové konfigurace, které měly největší napěťové rozdíly mezi takovými platnými vstupními logickými kombinacemi, jejichž logický výstup má být dle definice logické funkce také rozdílný. Podařilo se naleznout konfigurace umožňující realizaci hradel AND a OR, tím prototyp dokázal, že v kapalném krystalu lze naleznout logická hradla a má smysl pokračovat v práci. Ostatní hradla se prozatím nepodařilo realizovat.

Na obrázcích 5.1a a 5.1b jsou grafy opakovaného měření s vybranou konfigurací, která byla vybrána jako nejlepší výsledek pro zadané logické hradlo v rámci prohledávání hrubou silou.



(a) Krabicový graf 1000 měření nejnadějnější konfigurace pro realizaci hradla AND.

(b) Krabicový graf 1000 měření nejnadějnější konfigurace pro realizaci hradla OR.

Obrázek 5.1: Výsledky prvotních experimentů s prototypem.

## 5.2 Hledání hrubou silou

Tato podkapitola se již věnuje samotnému provádění experimentů s výslednou platformou. Experiment je řízen pomocí jazyka python přes rozhraní platformy, tak jak je k tomu platforma designována. Skript experimentu, tedy odesílá příkaz `exp` a v reálném čase zpracovává měřené výsledky.

Experimentální skript obsahuje několik proměnných, jako třeba počet prohledávaných vodičů, množinu prohledávaných napětí apod. Je důležité zdůraznit, že větší počet zkoumaných vodičů, nebo větší počet prohledávaných napětí, velmi razantně zvětšuje stavový prostor k prohledání a tím i čas experimentu.

Nicméně i tento skript dokáže naléznout logická hradla v kapalném krystalu, byť jen za použití částí všech vodičů displeje. Narozdíl od prototypu, tento experiment vyhodnocuje výsledky za běhu experimentu, nicméně hledá konfigurace podle identického klíče jako u experimentu s prototypem.

Experiment opět spolehlivě našel logická hradla AND a OR. Navíc dokázal nalézt logická hradla NOR a NAND. Bohužel při zopakování měření samostatně pro nalezené konfigurace hradel NOR a NAND, se již kapalný krystal jako tyto hradla nechoval.

### 5.3 Evoluční přístupy

Jelikož hledání hrubou silou nedokáže stačit na větší počet vodičů displeje. Je příhodné použít pro zkoumání rozsáhlejšího stavového prostoru evoluční algoritmy. Evoluční algoritmus začíná s množinou kandidátních řešení, která jsou zpravidla náhodná.

Každé kandidátní řešení sestává z konfigurace napěťových vodičů a dvou indexů vodičů, které jsou uvažovány jako vstupní. Každé takové kandidátní řešení je nazýváno genomem. Nad každým genomem je provedena náhodná mutace, která spočívá v náhodných změnách hodnot konfigurace či indexů vstupních vodičů. V rámci evolučního algoritmu jsou pak jednotlivé genomy ohodnoceny schopností realizovat logické hradlo, podobně jako v předchozích experimentech. Genomy, které jsou v realizaci logické funkce nejlepší, přecházejí do další generace a celý proces se opakuje.

Evoluční algoritmy poskytují variabilitu v přístupu k mutacím, uložení genomu, velikosti populace apod. Bohužel i přes několik pokusů a úpravách algoritmu, se nepodařilo získat výrazně lepší výsledky než pomocí hledání hrubou silou nebo pouze náhodného výběru. Takové chování bude nejspíše způsobené nízkou komplexitou použitého LCD displeje.

### 5.4 Diskuze výsledků

Diskuze výsledků je omezená z důvodu absence provádění komplexnějších experimentů, protože je to velmi časově náročná činnost, která zachází mimo možnosti této práce. Nicméně i s těmito zdánlivě omezenými znalostmi z experimentů, lze definovat vcelku očividné zkušenosti.

Kapalný krystal reaguje na napětí na jiných vodičích. Lze tedy měřit napětí na zdánlivě odpojeném vodiči, změnou napětí na jiných vodičích. Převážně se kapalný krystal choval jako napěťová sčítačka, či zvláštní odporová a kondenzátorová síť najednou. Zdá se, že chování kapalného krystalu je závislé na předchozích stavech. Proto občas při procházení všech kombinací napětí na vodičích mohl kapalný krystal vykazovat určité vlastnosti, ale po opakování stejné konfigurace odděleně měl již kapalný krystal vlastnosti jiné.

Je možné, že proces konfigurace musí být iterační, tzn. že kapalný krystal může reagovat na různé odlišné posloupnosti jednotlivých konfigurací. S ohledem na velký stavový prostor, se nabízí myšlenka vytvořit algoritmus, který dokáže v reálném čase oskenovat kapalný krystal a jeho jednotlivé závislosti mezi vodiči. To by znamenalo, že by bylo možné popsat vnitřní strukturu samotného displeje a využít tento popis pro efektivnější hledání logických funkcí. Je očividné, že množství a nápadů na experimenty je nespočetně.

## Kapitola 6

# Závěr

Platformu se podařilo úspěšně navrhnout, zkonstruovat, oživit a ověřit její funkci pomocí realizace základních experimentů. Tímto byl splněn hlavní cíl této práce. Oproti původní platformě pana Hardinga je vytvořená platforma opět o krok dál, a především opět otevírá možnosti dalšímu zkoumání v této oblasti.

Samotná platforma není nijak triviální a beru sestavení takového zařízení jako malý osobní úspěch. Totiž proces konstrukce postupoval od návrhu napěťových zdrojů, až po řešení emulace minimalistického terminálu uvnitř samotné platformy. Návrh tedy procházel všechny aspekty vývoje vestavěných zařízení, včetně všech komplikací a problémů, které taková činnost přináší. Naštěstí se podařilo všechny komplikace vyřešit a dovést práci do zdárného konce.

Výsledkem práce je univerzální platforma, která je snadno použitelná a zároveň poskytuje rozsah možností pro experimenty. Díky rozhraní, které platforma přináší, jsou experimenty pohodlně přenositelné a zároveň nechávají uživateli platformy volnost ve výběru programovacího jazyka, pomocí kterého chce experimenty řídit. Platforma nemusí být použita pouze pro experimenty s výpočtem v kapalném krystalu, ale obecně v aplikacích kde potřebujeme řídit množinu napěťových vodičů v nějakém rozsahu napětí a nezávisle.

Provedené experimenty ukazují, že kapalný krystal reaguje na přikládání napětí a lze ho přimět k napodobování, například logického hradla. Nicméně každý experiment je inspirací na dalších 10 experimentů, které už bohužel nespádají do časových možností této práce. Doufám, že vytvořenou platformu v budoucnu někdo využije k realizaci dalších experimentů a případnému dalšímu zkoumání této oblasti nekonvenčních výpočtů. Anebo třeba poslouží jako výuková pomůcka v rámci studia výčtu v materiálech či evolučních algoritmech.

# Literatura

- [1] *LCD displej* [online]. GM electronic, spol. s r. o. Dostupné z: [https://img.gme.cz/files/eshop\\_data/eshop\\_data/4/513-139/dsh.513-139.1.pdf](https://img.gme.cz/files/eshop_data/eshop_data/4/513-139/dsh.513-139.1.pdf).
- [2] ADLEMAN, L. M. Molecular Computation of Solutions to Combinatorial Problems. *Science*. 1994, sv. 266, č. 5187, s. 1021–1024.
- [3] ANALOG DEVICES INC. . *MAX220-MAX249 +5V-Powered, Multichannel RS-232 Drivers/Receivers* [online]. Analog Devices Inc., 2019. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX220-MAX249.pdf>.
- [4] DETTMER, P. *Immune: A Journey Into the Mysterious System That Keeps You Alive*. Hodder & Stoughton, 2021. ISBN 9781529360684.
- [5] HARDING, S. L. a MILLER, J. F. Evolution in materio: initial experiments with liquid crystal. In: *Proceedings, 2004 NASA/DoD Conference on Evolvable Hardware: 24-26 June 2004, Seattle Washington*. Los Alamitos, CA, USA: IEEE Computer Society, 2004, s. 298–305. ISBN 0-7695-2145-2.
- [6] HARDING, S. L. a MILLER, J. F. Evolution In Materio: Evolving Logic Gates in Liquid Crystal. *Int. J. Unconv. Comput.* Old City Publishing. 2007, sv. 3, č. 4, s. 243–257.
- [7] HARDING, S. L., MILLER, J. F. a RIETMAN, E. A. Evolution in Materio: Exploiting the Physics of Materials for Computation. *Int. J. Unconv. Comput.* Old City Publishing. 2008, sv. 4, č. 2, s. 155–194.
- [8] HERMANSON, K. D., LUMSDON, S. O., WILLIAMS, J. P., KALER, E. W. a VELEV, O. D. Dielectrophoretic Assembly of Electrically Functional Microwires from Nanoparticle Suspensions. *Science*. 2001, sv. 294, č. 5544, s. 1082–1086.
- [9] KHAN, H. N., HOUNSHELL, D. A. a FUCHS, E. R. H. Science and research policy at the end of Moore’s law. *Nature Electronics*. Nature Publishing Group. 2018, sv. 1, č. 1, s. 14–21.
- [10] KOTSIALOS, A., MASSEY, M. K., QAISER, F., ZEZE, D. A., PEARSON, C. et al. Logic Gate and Circuit Training on Randomly Dispersed Carbon Nanotubes. *Int. J. Unconv. Comput.* Old City Publishing. 2014, sv. 10, 5-6, s. 473–497.
- [11] LLOYD, S. Ultimate physical limits to computation. *Nature*. Nature Publishing Group. 2000, sv. 406, č. 6799, s. 1047–1054.

- [12] MICROCHIP TECHNOLOGY INC. . *TC7662B CHARGE PUMP DC-TO-DC VOLTAGE CONVERTER* [online]. Microchip Technology Inc., 2001. Dostupné z: <https://ww1.microchip.com/downloads/aemDocuments/documents/APID/ProductDocuments/DataSheets/21469a.pdf>.
- [13] MICROCHIP TECHNOLOGY INC. . *MCP23018/MCP23S18 16-Bit I/O Expander with Open-Drain Outputs* [online]. Microchip Technology Inc., 2008. Dostupné z: <http://ww1.microchip.com/downloads/en/devicedoc/22103a.pdf>.
- [14] MICROCHIP TECHNOLOGY INC. . *MCP3422/3/4 18-Bit, Multi-Channel  $\Delta\Sigma$  Analog-to-Digital Converter with I2C<sup>TM</sup> Interface and On-Board Reference* [online]. Microchip Technology Inc., 2009. Dostupné z: <https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/22088c.pdf>.
- [15] MICROCHIP TECHNOLOGY INC. . *ATmega164A/PA/324A/PA/644A/PA/1284/P megaAVR® Data Sheet* [online]. Microchip Technology Inc., 2020. Dostupné z: [https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega164A\\_PA-324A\\_PA-644A\\_PA-1284\\_P\\_Data-Sheet-40002070B.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega164A_PA-324A_PA-644A_PA-1284_P_Data-Sheet-40002070B.pdf).
- [16] MILLER, J. F. a DOWNING, K. Evolution in materio: looking beyond the silicon box. In: *Proceedings 2002 NASA/DoD Conference on Evolvable Hardware*. IEEE Computer Society, 2002, s. 167–176. ISBN 0-7695-1718-8.
- [17] OKAMOTO, A., TANAKA, K. a SAITO, I. DNA logic gates. *Journal of the American Chemical Society*. ACS Publications. 2004, sv. 126, č. 30, s. 9458–9463. PMID: 15281839.
- [18] SUN, Y., AGOSTINI, N. B., DONG, S. a KAELI, D. R. Summarizing CPU and GPU Design Trends with Product Data. *CoRR*. 2019, abs/1911.11313.
- [19] TEXAS INSTRUMENTS INC. . *TLC7524C, TLC7524E, TLC7524I 8-bit multiplying digital-to-analog converters* [online]. Texas Instruments Inc., 2007. Dostupné z: <https://www.ti.com/lit/ds/symlink/tlc7524.pdf>.
- [20] TEXAS INSTRUMENTS INC. . *LM1577/LM2577 SIMPLE SWITCHER® Step-Up Voltage Regulator* [online]. Texas Instruments Inc., 2013. Dostupné z: <https://www.ti.com/lit/ds/symlink/lm2577.pdf>.
- [21] TEXAS INSTRUMENTS INC. . *TL07xx Low-Noise FET-Input Operational Amplifiers* [online]. Texas Instruments Inc., 2023. Dostupné z: <https://www.ti.com/lit/ds/symlink/tl072h.pdf>.

# Příloha A

## Obsah přiloženého média

```
/
├── text - Technická zpráva
│   ├── pdf - PDF technické zprávy
│   └── tex - Zdrojové soubory technické zprávy
├── firmware - Řídící firmware platformy
│   ├── source - Zdrojové soubory firmwaru
│   └── binary - Přeložený firmware specificky pro platformu
├── scripts - Python skripty experimentů s platformou
└── board - KiCAD projekt platformy
```



## Příloha B

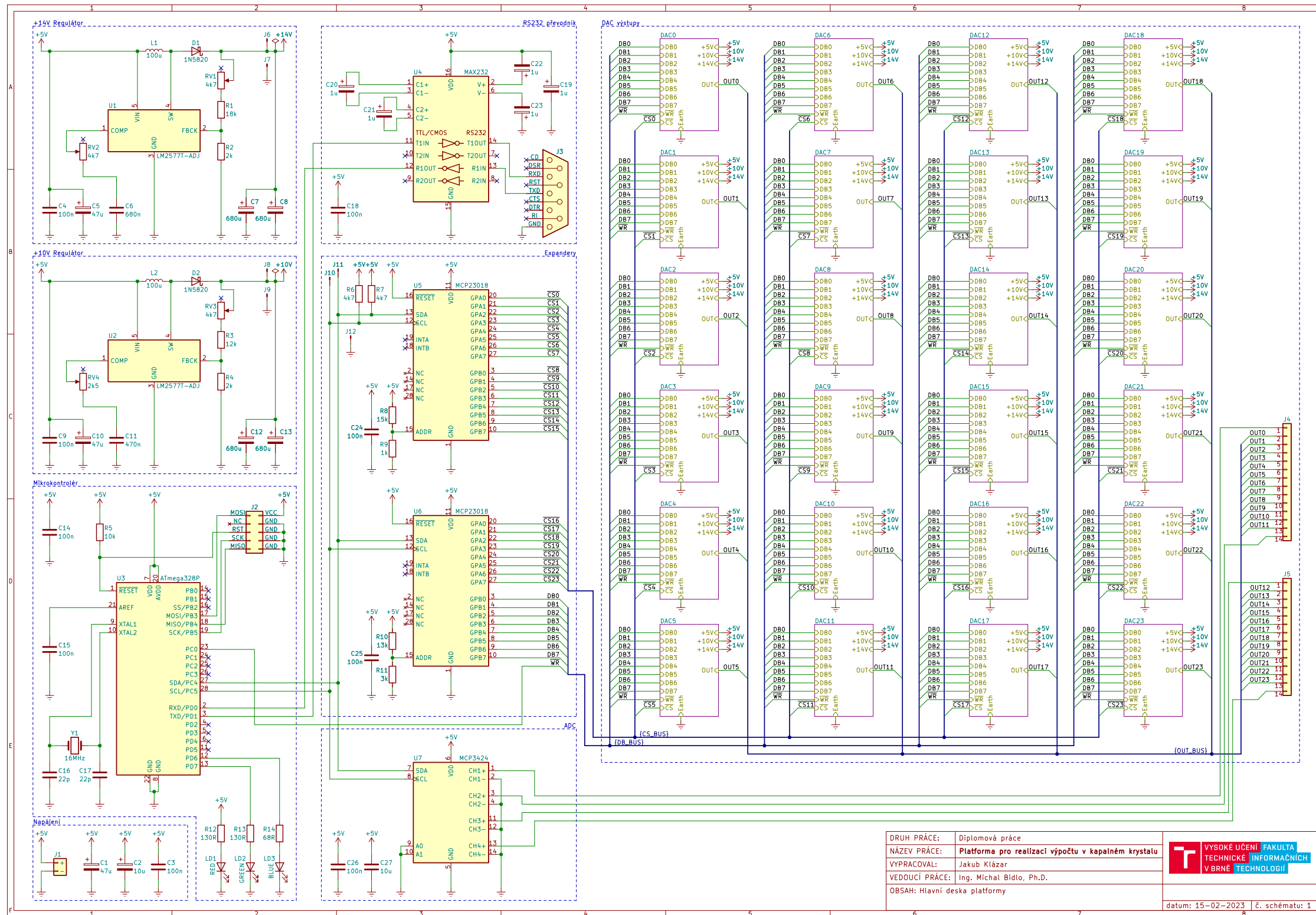
# Manuál použití platformy

Platforma se snaží být co nejvíc intuitivní a jednoduchá na použití. Pro zprovoznění platformy je nutné pouze připojit platformu k napájecímu napětí 5 V a poté se k platformě připojit pomocí sériové linky. Následně je již možné začít používat platformu k vlastním experimentům. Samozřejmě v rámci experimentů je nutné vložit do nepájivého pole platformy experimentální displej a připojit ho patřičně k napěťovým a měřícím vodičům. Tato práce obsahuje předpřipravené python skripty realizující základní experimenty. Před spuštěním skriptu je nutné upravit port, případně emulátor sériové linky, přes který má python skript komunikovat s platformou.

**Příloha C**

**Podrobné schéma systému**

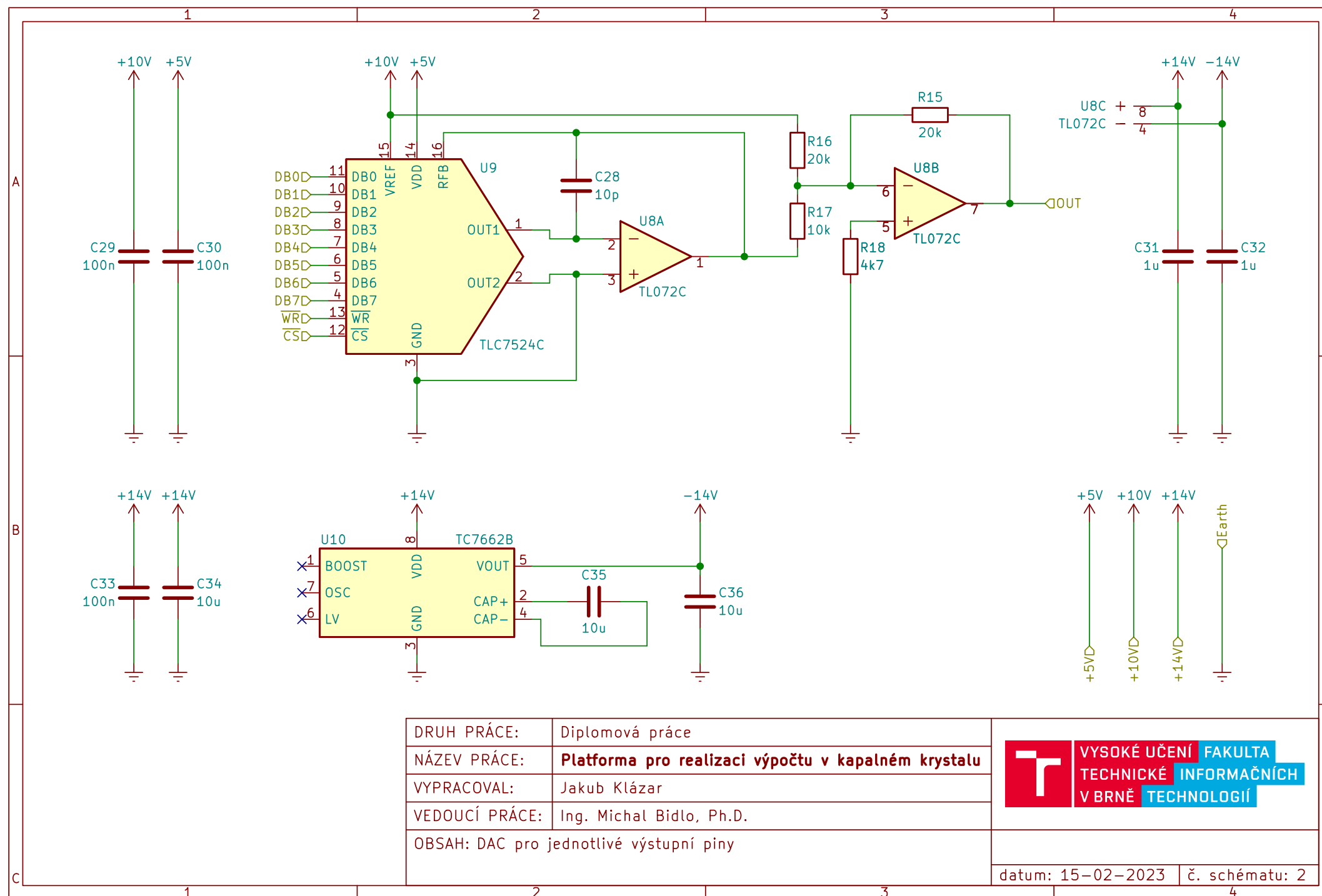
# Kompletní Elektronické schéma platformy



DRUH PRÁCE:	Diplomová práce
NÁZEV PRÁCE:	Platforma pro realizaci výpočtu v kapalném krystalu
VYPRACOVAL:	Jakub Klázar
VEDOUČÍ PRÁCE:	Ing. Michal Bído, Ph.D.
OBSAH:	Hlavní deska platformy



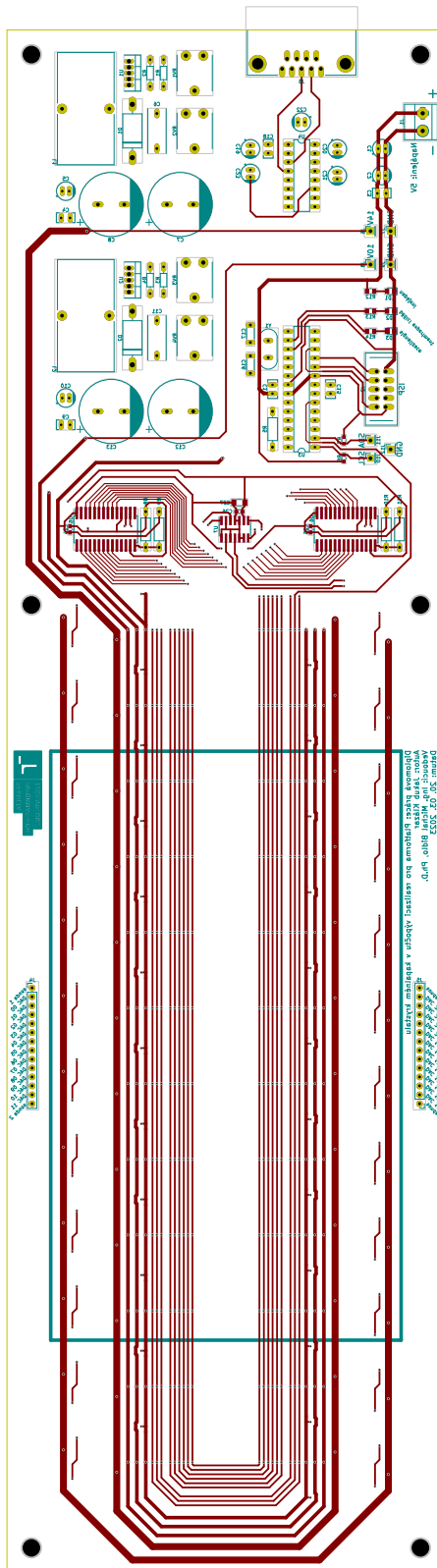
# Elektronické schéma zapojení DAC



DRUH PRÁCE:	Diplomová práce
NÁZEV PRÁCE:	<b>Platforma pro realizaci výpočtu v kapalném krystalu</b>
VYPRACOVAL:	Jakub Klázar
VEDOUCÍ PRÁCE:	Ing. Michal Bidlo, Ph.D.
OBSAH:	DAC pro jednotlivé výstupní piny

	VYSOKÉ UČENÍ	FAKULTA
	TECHNICKÉ	INFORMAČNÍCH
	V BRNĚ	TECHNOLOGIÍ
datum: 15-02-2023		č. schématu: 2

# Přední strana desky tištěných spojů platformy



## Zadní strana desky tištěných spojů platformy

