

GYMNASIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6



Univerzální sériový programátor

Maturitní práce

Autor: Petr Šícho

Třída: R8.A

Školní rok: 2021/2022

Předmět: Informatika

Vedoucí práce: Emil Miler

Praha, 25.3.2022



GYMNASIUM JANA KEPLERA
Kabinet informatiky

ZADÁNÍ MATURITNÍ PRÁCE

Student: Petr Šícho
Třída: R8.A
Školní rok: 2021/2022
Platnost zadání: 30. 9. 2022
Vedoucí práce: Emil Miller

Název práce: Univerzální sériový programátor

Pokyny pro vypracování:

Cílem práce je vymyslet a vytvořit univerzální programátor minimálně pro nejznámější čipy AVR, tedy řady Attiny a Atmega. Univerzální programátor má ulehčit nahrávání do zmíněných čipů, bez nutnosti zapojovat cokoli na nepáživém poli. Programátor by měl detekovat jaký je do něj vložený čip a umožnit do něj nahrávat z počítače. Za dodatečné, vedlejší, funkce by se dalo například považovat možnost nahrávat do čipů přes pin header, tedy bez vyndávání čipu z postaveného obvodu. Výsledkem projektu by minimálně měl být funkční prototyp programátoru, který je použitelný pro specifikovaný účel.

Doporučená literatura:

Zde bude *vedoucím práce* doplněna literatura, ze které je doporučeno při práci na tématu vycházet.

URL repozitáře:

<https://github.com/Xmen097/USP>

student

vedoucí práce

V Praze dne 19. 10. 2021

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 16. března 2022

Petr Šícho

Poděkování

Děkuji Emilovi Milerovi za pomoc při vedení této maturitní práce. Dále bych chtěl poděkovat open hardware komunitě, specificky pak Arduino komunitě, na jejíž fórem jsem našel značné množství informací.

Abstrakt

Tato maturitní práce si klade za cíl zjednodušit proces programování a nahrávání do mikrokontrolérů v pozdře DIP. Univerzální sériový programátor (USP) je obvod, který je schopný, po vložení podporovaného čipu do ZIF patice, automaticky detekovat rozložení vývodů vloženého čipu. Na základě toho rozpozná o který model, či rodinu modelů, se jedná a uzpůsobí pro něj nahrávací obvod. Teoreticky by měl USP umět programovat všechny čipy pracující na 5 voltech a všechny dostatečně pomalé bit-banging protokoly, včetně těch, které vyžadují vysoké napětí (12V) pro resetování.

Klíčová slova

programátor, čip, AVR, nahrávač, USP

Abstract

Abstract.

Keywords

programmer, chip, AVR, uploader, USP

Obsah

1	Teoretická část	3
1.1	Detekce	3
1.1.1	Detekce velikosti a polohy	4
1.1.2	Detekce speciálních pinů	4
1.1.3	Rozpoznání VCC od GND	5
1.2	Nahrávání	5
1.3	Shrnutí požadovaných funkcí	6
2	Implementace	7
2.1	Hardware	7
2.1.1	Vkládání čipu	7
2.1.2	Řídící mikroprocesor	7
2.1.3	Funkcionalita pinů	8
2.1.4	Dodatečný hardware	11
2.1.5	Plošný spoj	11
2.2	Software	12
2.2.1	Nízkoúrovňové funkce	12
2.2.2	Čtení kapacity	12
3	Technická dokumentace	13
	Závěr	15
	Seznam použité literatury	17
	Seznam obrázků	19
	Seznam tabulek	20

1. Teoretická část

Stejně jako existuje nepřeberné množství různých čipů, je také značné množství možných rozložení vývodů, které není jednotné ani u konkrétní velikosti a jednoho výrobce. Pro každé rozložení je nutné mít nebo si postavit externí nahrávací obvod. Cílem USP je sjednotit tyto nahrávací obvody do jednoho univerzálního, který se přizpůsobí vloženému čipu. Programátor má tedy dvě principiální části - detekci a nahrávání.

Chceme-li vytvořit opravdu univerzální programátor, není možné použít žádný dedikovaný hardware, který by zajišťoval nahrávání. Naopak se nám hodí použít univerzálních vstupních / výstupních pinů neboli GPIO pinů ¹.

Jako minimální požadovanou funkcionalitu jsem bral podporu čipů AVR od výrobce Atmel, konkrétně pak řady ATmegy a ATtiny, které zahrnují čipy, jež pohání známé desky Arduino. Jde o čipy, které jsou oblíbené pro svou jednoduchost a i já je nejčastěji používám. Přestože se snažím o vytvoření univerzálního programátoru, přišlo mi vhodné vzít si určitou rodinu čipů jako referenční bod, neboť nikdy stejně nedocílíme podpory 100% všech čipů. Na těchto čipech zároveň budu testovat funkce programátoru. Přesto doufám, ikdyž to nemůžu otestovat, že vlastnosti těchto čipů, na kterých je detekce a nahrávání postaveno budou do velké míry přenositelné i na jiné rodiny či výrobce čipů.

1.1 Detekce

Základem detekce je poznat jak velký čip je - tedy kolik má vývodů. Jen díky tomu se nám radikálně sníží počet možných kandidátů. Při zahrnutí jen několika modelů, jednoho pro každou velikost, bychom tak měli prakticky hotovo. Bylo by však nutné arbitrárně stanovit tyto podporované čipy, přičemž při vložení jiného, ať třeba ze stejné rodiny, který má prohozené piny GND a VCC, by došlo pravděpodobně k jeho zničení. Naší ambicí tedy bude zjistit o čipu co nejvíce informací, díky nimž budeme schopni rozeznávat mezi velkým počtem různých čipů.

Při všem musíme dbát, abychom čip nepoškodili. Budeme se snažit držet v hodnotách, které povoluje datasheet. Vzhledem k tomu, že to, o co se pokoušíme není standardní zacházení s čipy, nenalezneme v datasheetu konkrétní povolené hodnoty a situace, kterým můžeme čip bezpečně vystavit. Můžeme z něj ale vyčíst ale základní principy, kterých bychom se měli držet, abychom minimalizovali poškození čipu. Konkrétně se jedná o:

- Nevystavovat žádný pin napětí nižšímu než $-0.5V$
- Nevystavovat žádný pin, kromě RESETu, napětí vyššímu než $V_{CC} + 0.5V$
- Nevystavovat RESET napětí vyššímu než $13V$
- Nenechat mezi žádný dvěma piny nezapojeného čipu téct nezanedbatelně velký proud po nezanedbatelně dlouhou dobu

Při splnění těchto podmínek by měl být čip dostatečně chráněn obvody, které běžně slouží k ochraně

¹ z anglického General-purpose input/output, tedy pin, který se umí chovat buď jako digitální vstup, nebo výstup, přičemž mezi těmito módy je možné přepínat.

proti elektrostatickým výbojům (ESD), které jsou zabudovaná do prakticky všech čipů.²

1.1.1 Detekce velikosti a polohy

Pro detekci velikosti čipu můžeme využít například toho, že každá reálná součástka, kromě svých primárních vlastností, vykazuje také elektrickou kapacitu. To bude jistě platit i pro spoje a součástky uvnitř integrovaných obvodů. Elektrická kapacita pinů jednoduchých CMOS³ čipů[5, 6, 9] se pohybuje řádově v hodnotách pikofaradů, což není mnoho. Budeme tedy potřebovat měřit elektrickou kapacitu s dostatečnou přesností. Není mi znám žádný způsob přímého měření elektrické kapacity, budeme tedy muset provést měření nepřímé. Můžeme kapacitor pomalu nabíjet nebo vybíjet a měřit čas, který to zabere. Bude takové měření ale dostatečně přesné? Zkusíme to odhadnout s pomocí rovnice definující energii elektrického kondenzátoru

$$W = \frac{1}{2}CU^2 \quad (1.1)$$

Budeme-li kapacitor nabíjet napětím U a proudem I po dobu t , dodáme mu tím maximálně

$$W_{\max} = U \cdot I \cdot t \quad (1.2)$$

Jedná se o horní odhad vykonané práce, která počítá s nulovým odporem kapacitoru a nulovým elektrickým potenciálem kapacitoru. Úpravou vztahů můžeme vyvodit úměrnost

$$C \simeq t/R \quad (1.3)$$

Kapacita je tedy přímo úměrná času, po který jsme kapacitor nabíjeli a nepřímo úměrná odporu, přes který jsme nabíjeli.⁴ Čas budeme schopni měřit velice přesně, nabíjet bude možné po dobu řádově mikrosekund (10^{-6} s). Řekněme, že budeme chtít zjistit změnu kapacity $10\text{pF} = 10^{-11}\text{F}$. Dosazením do vztahu (1.3) zjistíme, že k tomu budeme potřebovat nabíjecí odpor o velikosti řádově $100\text{k}\Omega$. To vypadá proveditelně.

Změřením kapacity každého z pinů socketu zjistíme nejenom jak velký čip je, ale zároveň s tím určíme kde přesně v socketu je vložen. Není tedy nutné nijak specifikovat kam má být čip pro nahrávání vložen.

1.1.2 Detekce speciálních pinů

Každý mikrokontrolér má kromě vstupně/výstupních pinů i piny, které slouží k nějaké speciální funkci. Například může jít o RESET pin, napájecí piny (GND a VCC) nebo piny hodin (XTAL). Tyto piny budou pravděpodobně vykazovat rozdílné vlastnosti od ostatních pinů. Největší rozdíl bude v napájecích pinech, které jsou vnitřně připojené k všemožným součástem mikrokontroléru. Při dostatečně přesném měření kapacity bychom mohli tyto rozdíly detekovat.

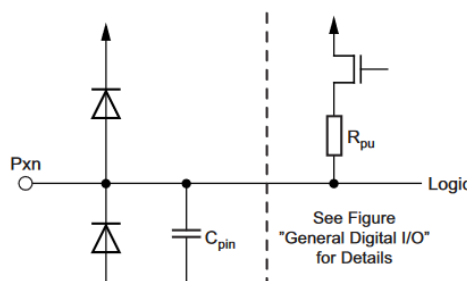
²Pro naše účely budeme považovat za zanedbatelný proud řádu mikroampér a čas mikrosekund.

³Complementary metal-oxide semiconductor; technologie výroby integrovaných obvodů

⁴Ke stejnému výsledku bychom došli i použitím rozměrové analýzy.

1.1.3 Rozpoznání VCC od GND

Při špatném rozpoznání mikrokontroléru může největší poškození způsobit převrácení pinů GND a VCC. Abychom se tomuto problému vyvarovali, bylo by dobré rozpoznávat s jistotou VCC a GND, protože měření kapacity nebude nejspíš dostatečně přesné na jejich rozlišení.



Obrázek 1.1: Schéma vnitřního zapojení IO pinu mikrokontroléru [3, str. 58]

Využijeme ochranných diod, které jsou připojeny ke každému IO pinu mikrokontroléru jak ukazuje obrázek 1.1. Povšimněme si, že dioda spojující pin s GND směřuje opačně než dioda spojující pin s VCC. Díky tomu bychom měli být schopni rozpoznat, zda je určitý pin GND či VCC.

Pin GND můžeme detekovat takto. Budeme znát úbytek napětí na diodě U_d . Přivedeme-li na pin GND napětí U převyšující U_d , měli bychom na všech IO pinech detekovat napětí $U - U_d$. Napětí U by mělo být co nejmenší možné, konkrétně $U < 0.5V$. Napětí přivedeme přes dostatečně velký odpor tak, aby mikrokontrolérem a diodou tekla co nejmenší proud. S malým procházejícím proudem budeme mít také menší úbytek napětí na diodě (menší, než běžně udávaný $\sim 0.7V$).

Obdobným způsobem je možné detekovat pin VCC. Stačí na libovolný (nebo lépe postupně na všechny) IO pin přivést napětí U a na pinu VCC by se nám vždy mělo ukázat napětí $U - U_d$.

1.2 Nahrávání

Po detekování čipu nám už zbývá pouze do něj nahrát. Vzhledem k tomu, že přesně nevíme jaké rozložení pinů bude čip mít, budeme použít tzv. bit-banging⁵. Díky tomu se nemusíme na při stavbě nahrávače na určité čipy, ale bude teoreticky možné nahrávat do všech čipů, které nevyžadují příliš velké komunikační rychlosti.

Některé programovací protokoly vyžadují použití vyššího napětí. U AVR čipů jde konkrétně o HVSP (high voltage serial programming) a HVPP (high voltage parallel programming), které se hodí při určitém nastavení mikrokontroléru, při kterém není možné standardní programování.⁶ Vysokým napětím je v obou případech myšleno 12 voltů.[7]

⁵Softwarová implementace určitého protokolu, která přímo generuje protokolem specifikovaný signál, bez použití specializovaného hardwaru

⁶Jde např. o situace kdy je vypnuté resetování pomocí RESET pinu, protože ho chceme použít na jinou funkci.

1.3 Shrnutí požadovaných funkcí

V této sekci si shrneme funkce, které budeme potřebovat pro úspěšnou detekci a nahrávání do čipů. Tyto funkce by měl mít každý z pinů socketu, neboť nevíme kde zrovna se čip ocitne. Některé funkce budou komplexní (přečti kapacitu připojenou k pinu), u jiných se bude jednat o nějaký elementární stav (připojení k zemi/5V). Některé funkce nebo stavy, které zde budou vyjmenované, třeba bod 1 a 3, se prolínají nebo jsou velmi podobné. Zmiňuji je separátně proto, že jsou potřeba z jiného důvodu a mohly by být implementovány rozdílně.

1. **5V s proudem >20 mA** je potřeba pro napájení čipu. Proud 20 mA by měl stačit pro napájení malých AVR čipů.[4] Větší čipy mají sice větší spotřebu než 20 mA, ale mají také několik separátních napájecích pinů, mezi které se proud rozloží.[3]
2. **0V (GND) s proudem >20 mA** je potřeba též pro napájení čipu.
3. **5V s libovolně malým proudem** reprezentuje logický stav 1 při komunikaci, odebíraný proud bude zanedbatelný
4. **0V (GND) s libovolně malým proudem** reprezentuje logický stav 0 při komunikaci
5. **Čtení digitální hodnoty** pro čtení odpovědi čipu při nahrávání; standardní funkce GPIO pinů
6. **Přečtení kapacity** pro detekci čipu
7. **Napětí přibližně 0.4-0.5V** je potřeba pro rozpoznání GND a VCC, viz sekce 1.1.3
8. **Přečtení napětí na pinu** je také potřeba pro rozpoznání GND a VCC
9. **12V** bude potřeba pokud budeme chtít high-voltage programování.
10. **Nepřipojeno** k ničemu. Piny, které nevyužíváme k jiné funkci by se měly chovat tak, jako kdyby nebyly k ničemu připojeny.

2. Implementace

Implementace projektu měla dvě fáze. Nejprve bylo potřeba navrhnout hardwarovou část - vymyslet a vyzkoušet jednotlivé části obvodu tak, aby plnily požadované funkce. Poté je bylo potřeba všechny spojit do jednoho obvodu, který se vyrobí jako plošný spoj. Po osazení plošného spoje součástkami a otestování základní funkčnosti přišel čas na software. Nejdříve bylo potřeba implementovat low-level komunikaci s jednotlivými součástkami, pak přišly na řadu high-level funkce. Pojd'me si jednotlivé fáze rozebrat detailněji.

2.1 Hardware

Převážná část práce na hardwaru probíhala s pomocí nepájivého pole, na kterém jsem postupně testoval jednotlivé ideje uvedené v teoretické části práce. V následujících sekcích detailně popíši jaké součástky byly použity a jak se pomocí nich implementují požadované funkce.

2.1.1 Vkládání čipu

Jako patici, do které se bude vkládat čip k programování jsem zvolil ZIF (zero insertion force) socket. Jak již jméno napovídá, pro vložení čipu do patice není na čip nijak tlačit. Stačí čip položit na patici a pomocí páčky na straně patice se pak čip zajistí a vytvoří se spolehlivý kontakt s piny čipu. Nehrozí tedy ohnutí nožiček čipu a jeho vyndání a zandání je velmi rychlé. Nevýhodou ZIF socketu může být o trochu vyšší pořizovací cena a větší velikost, ale i přesto se v našem případě ZIF socket vyplatí. Konkrétně jsem zvolil patici se 40 piny (dvaceti na každé straně), která by měla postačovat pro naprostou většinu DIP¹ pouzder. Patice podporuje vložení jak úzké (0.3"), tak široké (0.6") varianty DIP.

2.1.2 Řídící mikroprocesor

Bylo potřeba zvolit mikroprocesor, který bude ovládat celý nahrávač. Volil jsem mezi mikrokontroléry AVR, s kterými jsem nejvíce obeznámen, umí jednoduše komunikovat s Arduino IDE a také pro ně existují knihovny, které se nám budou dále hodit. Zásadní otázkou, kterou bylo potřeba vyřešit je počet potřebných GPIO pinů na mikrokontroléru. Budeme potřebovat jeden pin pro každý pin ZIF patice, tedy 40. Další desítky pinů budou potřeba pro ovládání dalších potřebných komponent, o kterých se budu detailněji zmiňovat v dalších sekcích. Teoreticky by bylo možné přidat další piny pomocí shift registrerů či IO expanderů. Tím by však utrpěla rychlost komunikace a vzrostla by komplexita desky a tím i prostor pro chyby. Rozhodl jsem se tedy pro jeden co největší čip, který bude dostupný. Tím je mikrokontrolér ATmega2560 v pozdě TQFP100, který má 84 GPIO pinů, což by mělo stačit.[2]

Nejprve jsem plánoval, že bych využil dostupných schémat Arduino desek a v upravené podobě je přímo zakomponoval do desky nahrávače. To by o něco snížilo náklady na jednu desku, umožnilo

¹Dual in-line package; pouzdro s THT piny s roztečí 2.54mm

využít čip bez omezení. Vzhledem k tomu, že nejsem schopný sám osadit plošný spoj SMD součástkami s tak malými vývody, chtěl jsem si nechat plošný spoj osadit spolu s výrobou. Žádný výrobce nedodává méně než 5 kusů desek, což by znamenalo čtyři zbytečné ATmegy. Proto jsem se rozhodl programátor vyvynout jako shield (nasazovací desku) na Arduino mega ve variantě s procesorem Atmega2560. Nevýhodou tohoto postupu je snad jen to, že Arduino mega dává k dispozici pouze 70 z 84 GPIO pinů, ale i to by mělo pro projekt stačit.[1]

2.1.3 Funkcionalita pinů

Nyní přichází na řadu implementace požadovaných funkcí popsaných v sekci 1.3. Ke každému pinu ZIF socketu bude připojený jeden GPIO pin. Jenom tímto samo o sobě zajistíme funkce 3, 4, 5, které GPIO pin zvládá nativně. Funkce 1 a 2, které vyžadují minimální proud 20 mA také s trochou opatrnosti jednoduše zvládneme. Digitální piny ATmegy2560 dokáží dodávat maximální proud 40 mA [2, str. 355]. Musíme tedy jen dát pozor na to, aby další součástky, které budou zapojeny mezi GPIO pinem a pinem ZIF socketu neměly příliš velký odpor nebo nevyžadovaly více omezený protékající proud. S dalšími funkcemi to již nebude tak jednochuché.

Desátý požadavek vyžaduje trochu zamyšlení. Pin určitě nebude možné fyzicky odpojit, to bychom museli provést manuálně. Budeme tedy alespoň chtít maximalizovat odpor čili minimalizovat proud, který může skrz tento 'odpojený' pin téct. Když nastavíme pin jako vstupní, bude ve stavu s tzv. vysokou impedancí, tedy budou jím téct velmi malý proud. Datasheet specifikuje maximální proud jako $1\mu\text{A}$, typický proud ale očekávám výrazně menší. Odpojení si navíc můžeme pojistit použitím analogového přepínače, u kterého je udáván maximální proud 100nA a typický 0.05nA . [9, str. 3]

Zapojení multiplexerů

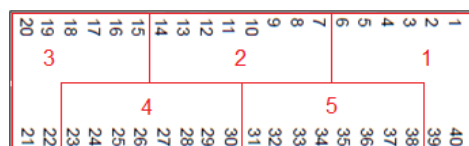
Implementaci funkcí 6, 7, 8 a 9 si zjednodušíme pozorováním, že v jednu chvíli bude daná funkce potřebná pouze na jednom pinu. Nemusíme ji tedy implementovat 40x, ale můžeme použít multiplexery k spojení určitého pinu s požadovaným stavem nebo obvodem zajišťujícím funkci. Napětí na pinu budem číst pomocí ADC² převodníku. Atmega2560 disponuje jedním ADC převodníkem, který je ale vnitřně multiplexován na 16 různých pinů. My dále pomocí multiplexerů rozšíříme tuto funkcionalitu na všech 40 pinů.

Pro zajištění správného přečtení napětí nemůžeme nechat ADC převodník tzv. floating, tedy nepřipojený k ničemu, kdy vrací víceméně náhodné hodnoty. ADC pin jsem tedy připojil rezistorem $1\text{M}\Omega$ k zemi. Takto vysokou hodnotu rezistoru jsem volil, abych nezkreslil měření kapacity.

Multiplexery se standardně vyrábí s 2^n kanály, které se ovládají pomocí n pinů. Zvolil jsem 8 kanálové multiplexery[5], tedy s 3 ovládacími piny. Pro obsluhu 40 pinů jich bude potřeba celkem 5. Multiplexery jsou připojeny k pinům ZIF socketu způsobem znázorněným na obrázku 2.1

Díky tomuto způsobu zapojení budeme schopni odhalit závadu na multiplexeru při rozpoznávání čipu, neboť i nejmenší čip, který má pouzdro DIP4 se 4 piny na každé straně bude vždycky mít alespoň dva piny v různých multiplexerech. Nestane se tedy, že by kvůli závadě na jednom mul-

²Analogově digitální převodník; zařízení, které převede analogový signál (napětí) v určitém rozsahu na digitální tvar s určitým počtem bitů, v našem případě se jedná o převodník 10-bitový

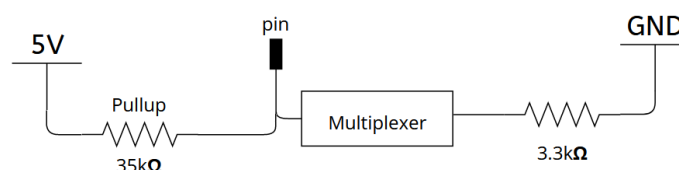


Obrázek 2.1: Schéma zapojení multiplexerů k pinům ZIF patice

tiplexeru (ta může být způsobená i třeba dotekem uživatele, čímž se zvýší kapacita) byl čip chybně rozpoznán.

Napěťový dělič

Napětí 0.4-0.5V, jak si žádá 7. požadavek, vytvoříme pomocí napěťového děliče. Zde se nám budou hodit již výše zmíněné multiplexery. Dělič napětí totiž vytvoříme právě mezi GPIO pinem a multiplexovaným pinem. Na každém GPIO pinu je možné zapnout tzv. pullup rezistor, který pin připojí přes odpor s přibližnou velikostí 30 – 40kΩ³ k 5 voltům. Chtěné napětí přibližně půl voltu vytvoříme dalším rezistorem s velikostí 3300Ω⁴ k zemi. Zapojení děliče napětí na pinu je vidět na obrázku 2.2



Obrázek 2.2: Schéma děliče napětí na pinu

Měření kapacity

Kapacita lze měřit, jak jsme vyvodili v sekci 1.1.1, pomocí pomalého nabíjení - tedy nabíjení přes dostatečně velký rezistor. Vypočítali jsme sice, že by se nám hodil rezistor s odporem 100kΩ. Připojovat takový odpor k pinům by si vyžádalo dalších multiplexerů nebo switchů. Můžeme ale využít stejných pullup rezistorů, které využíváme pro napěťový dělič. Ty jsou o něco menší a mohlo by se stát, že kapacitor nabijeme příliš rychle. Můžeme ale chytře využít ADC převodníku, který máme multiplexovaný k pinům. Původně jsme chtěli měřit čas, za který nabijeme plně kapacitor, což bychom zjišťovali pomocí digitálního vstupu GPIO pinů. Můžeme to ale provést i opačně. Kapacitor budeme nabíjet nějaký konstantní čas, a poté pomocí ADC převodníku zjistíme napětí na kapacitoru. Čím vyšší napětí bude, tím více jsme kapacitor nabili a tím je tedy kapacitor menší. Další možností by bylo kapacitor plně nabít a přestat ho nabíjet, čímž se začne postupně vybíjet. Počkáme konstantní čas a opět pomocí ADC převodníku zjistit jak moc se vybil.

³Datasheet udává, že odpor bude vždy větší než 20kΩ a menší než 50kΩ. Vzhledem k tomu, že tyto hodnoty jsou pro teplotní rozmezí od -55°C do 125°C můžeme konstatovat, že při pokojové teplotě bude odpor přibližně mezi 30 – 40kΩ.

⁴Chceme dělič napětí přibližně 1:10, hodnotu odporu jsem zvolil spíše o něco nižší, protože i samotný multiplexer bude mít určitý odpor (asi 100 ohmů).

Při testování ukázalo se dokonce ukázalo, že pro měření kapacity se nám velmi hodí i dělič napětí, více o tom však až v sekci 2.2.2

Napětí 12 voltů

Poslední funkcionalitou, kterou nám zbývá implementovat je dostupnost napětí 12 voltů. Ty bychom na pin použili pomocí stejných multiplexerů, které využíváme k připojení ADC převodníku. Stačí pomocí switchu[6] přepínat zdroj multiplexeru mezi 12 volty a ADC převodníkem. Zapojení tohoto obvodu pro multiplexer 1 a 2 je vyobrazeno v příloze č. 2. Od použitého switchu nepožadujeme žádné zvýšené nároky ohledně proudu, pro HVSP i HVPP nám stačí libovolně malý proud. Obecně se ale ukázalo, že podpora 12 voltů nám celou situaci značně zkomplikuje. Náš řídicí mikrokontrolér, ale ani většina dobře dostupných multiplexerů nepodporuje vyšší napětí než 5 voltů. S trochou snahy se dají najít i multiplexery podporující 12 voltů. Řídicí čip jehož piny odolají napětí 12 voltů ale prakticky možné není. Budeme proto muset piny řídicího mikrokontroléru v případě připojení 12 voltů odpojit. K tomu využijeme analogové switchy[9]. Vzhledem k tomu, že přes tyto switchy budeme mimo jiné napájet čip, který budeme chtít programovat, musí podporovat průtok proudu alespoň 20mA.⁵ Každý switch má 4 kanály, které dokáže přepínat a k tomu používá 4 ovládací piny. Není možné ani potřebné ovládat každý z těchto ovládacích pinů individuálně, na to bychom totiž spotřebovali dalších 40 pinů.

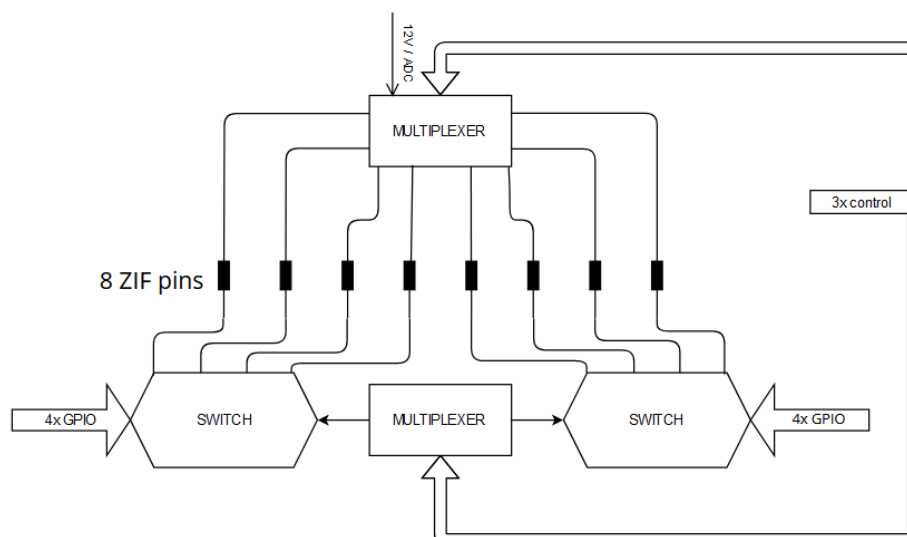
Uvažoval jsem dvě možnosti. Šlo by pomocí tranzistorů a odporů detekovat zvýšené napětí a přepnout ovládací pin switchu. To by však vyžadovalo značné množství externích komponent - na každý pin jeden tranzistor a několik odporů. Lepší řešení je založené na faktu, že 12 voltů bude vždy připojené na pinu, který je multiplexován. Toho využijeme a přidáme další multiplexer, který bude mít stejně zapojené ovládací piny jako první multiplexer. Jeho funkce bude, že vypne switch, který obsluhuje pin, ke kterému je multiplexováno. Tímto zajistíme přepínání s použitím pouze jediného pinu, který bude sloužit k potlačení této funkcionality, která by nám v opačném případě rozbila napěťový dělič a další funkce. Zjednodušené schéma tohoto systému pro 8 pinů je znázorněno na obrázku 2.3. Tento obvod je na USP celkem 5 krát - pro každých 8 pinů, a to podle stejného schématu jako zapojení původních multiplexerů (obrázek 2.1). Kompletní zapojení toho obvodu je dostupné jako příloha č.1

Další komplikací, kterou vyšší napětí přineslo bylo to, že multiplexery a switchy potřebují být ovládané tímto napětím. Nestačí tedy již připojit na ovládací pin multiplexeru/switchu GPIO pin řídicího mikrokontroléru, ale je nutné použít tranzistor a pullup/pulldown⁶ rezistor.

Poslední věcí, kterou musíme vyřešit je kde vůbec získat napětí 12 voltů. Jednou možností by bylo zapojit do zařízení přímo toto napětí z externího zdroje. To by ale výrazně ztížilo použití výsledného zařízení. Rozhodl jsem se proto zabudovat přímo do USP step-up převodník, který nám vytvoří napětí 12V. Obvod step-up převodníku jsem navrhl podle datasheetu regulátoru[8]. Zapojení obvodu převodníku je zobrazeno v příloze č.3

⁵Z tohoto důvodu se použitý switch liší od switchů [6], které používáme u zdroje multiplexeru.

⁶V závislosti na tom v jakém výchozím stavu chceme, aby byl ovládací pin; ve většine případu se bude jednat o pulldown rezistor



Obrázek 2.3: Zjednodušené schéma obvodu připojeného k 8 pinům ZIF patice

2.1.4 Dodatečný hardware

Krom výše uvedených součástek a obvodů, které slouží k zajištění detekce nebo nahrávání je USP vybaven také 0.91" OLED displayem a 4 tlačítky. To umožní tvorbu uživatelského rozhraní pro jednodušší ovládání programátoru. OLED display byl také skvělým nástrojem pro debugování, protože na něm šlo jednoduše zobrazit v reálném čase informace, bez nutnosti komunikace s počítačem přes sériovou linku (která byla potřeba k jiným účelům). Dále jsou pak na programátoru tři LED - modrá, zelená a červená, které slouží pro grafickou signalizaci stavu programátoru.

2.1.5 Plošný spoj

Po otestování částí některých výše zmíněných obvodů na nepájivém poli jsem se pustil do návrhu plošného spoje. První bylo potřeba vytvořit diagramy všech potřebných obvodů a zapojení, část z nich je možné vidět v přílohách č. 1, 2 a 3. Poté přišlo na řadu návrh plošného spoje, kde, vzhledem k tomu, že autorouter⁷ stále moc nefunguje, bylo potřeba 'natahat' cestičky mezi součástkami. Rozhodl jsem se rovnou pro 4-vrstvý plošný spoj, vzhledem k tomu, že jsem plánoval natěsnat co nejvíce součástek na co nejmenší plochu a vrchní vrstva je tak prakticky celá pokryta ploškami pro kontakty součástek. Prostřední dvě vrstvy jsou udělány jako tzv. plane, kdy většina vrstvy zabírá jeden signál, zde konkrétně země a napájení, a slouží pro snížení rušení na desce. Obrázek hotového zapojení plošného spoje v příloze č.4.

Plošný jsem si nechal vyrobít a osadit SMD součástkami u nejmenované čínské firmy. Hlavní výhodou objednání z Číny je řádově menší cena, než u ostatních výrobců, problémem na druhou stranu může být, že čínané mají často vlastní kopie součástek a jejich datasheety nejsou přeložené do angličtiny. Většinou jsou k nalezení anglické datasheety prakticky totožných součástek jiných výrobců. Po výrobě plošného spoje jsem začal zjišťovat na co vše jsem při návrhu zapomenul a co nefunguje. Naštěstí nic z toho nebylo kritické a bylo možné všechny nedostatky opravit.

⁷Program, který má automaticky nalézt a zapojit potřebné spoje na desce

2.2 Software

2.2.1 Nízkoúrovňové funkce

Jako první bylo na

2.2.2 Čtení kapacity

3. Technická dokumentace

Poslední kapitola obsahuje informace o tom, jak projekt, který v rámci maturitní práce vznikl, nainstalovat, spustit a používat.

Závěr

Závěr obsahuje shrnutí práce a vyjadřuje se k míře splnění jejího zadání. Dále by se zde mělo objevit sebehodnocení studenta a informace o tom, co nového se naučil a jak vnímal svou práci na projektu.

Seznam použité literatury

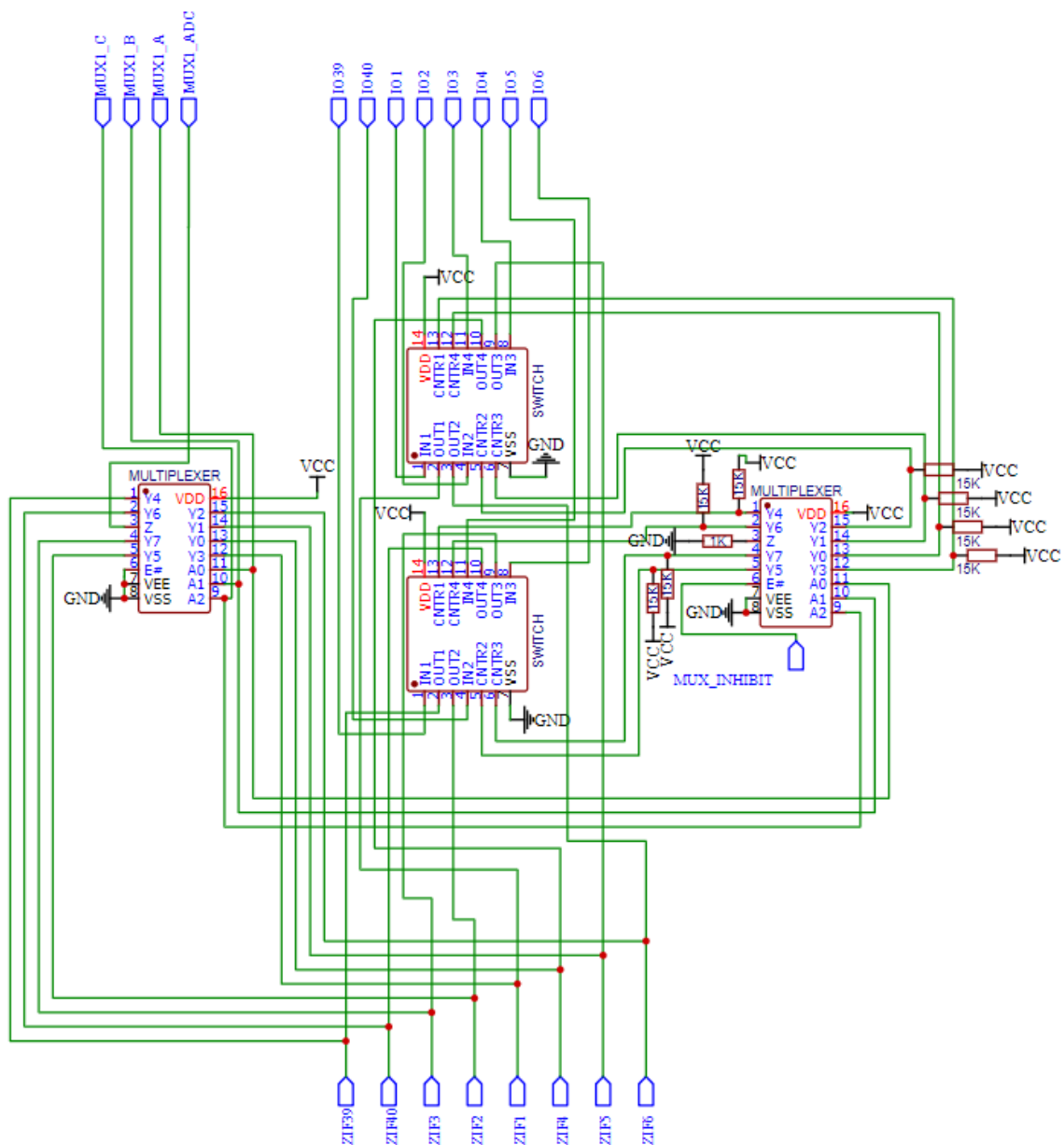
- [1] Arduino. *Arduino Mega*. URL: <https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardMega> (cit. 15.03.2022).
- [2] Atmel. *8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash*. ATmega640/V-1280/V-1281/V-2560/V-2561/V. Datasheet 2549Q-AVR-02/2014.
- [3] Atmel. *8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*. ATmega328P. Datasheet 7810D-AVR-01/15.
- [4] Atmel. *Atmel 8-bit AVR Microcontroller with 2/4/8K Bytes In-System Programmable Flash*. ATtiny25/V / ATtiny45/V / ATtiny85/V. Datasheet 2586Q-AVR-08/2013.
- [5] Texas Instruments. *CD405xB CMOS Single 8-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion*. CD4051B. Datasheet SCHSo47I, Rev. 2017.
- [6] Texas Instruments. *CD4066B CMOS Quad Bilateral Switch*. CD4066B. Datasheet SCHSo51H, Rev. 2020.
- [7] Matthias Neeracher. *ScratchMonkey - Overview*. 2016. URL: <https://microtherion.github.io/ScratchMonkey/Overview.html> (cit. 14.03.2022).
- [8] ON Semiconductor. *1.5 A, Step-Up/Down/ Inverting Switching Regulators*. MC34063A. Datasheet MC34063A/D, Rev. 2010.
- [9] ON Semiconductor. *Quad Analog Switch/Quad Multiplexer*. MC14066B. Datasheet SCHSo47I, Rev. 2017.

Seznam obrázků

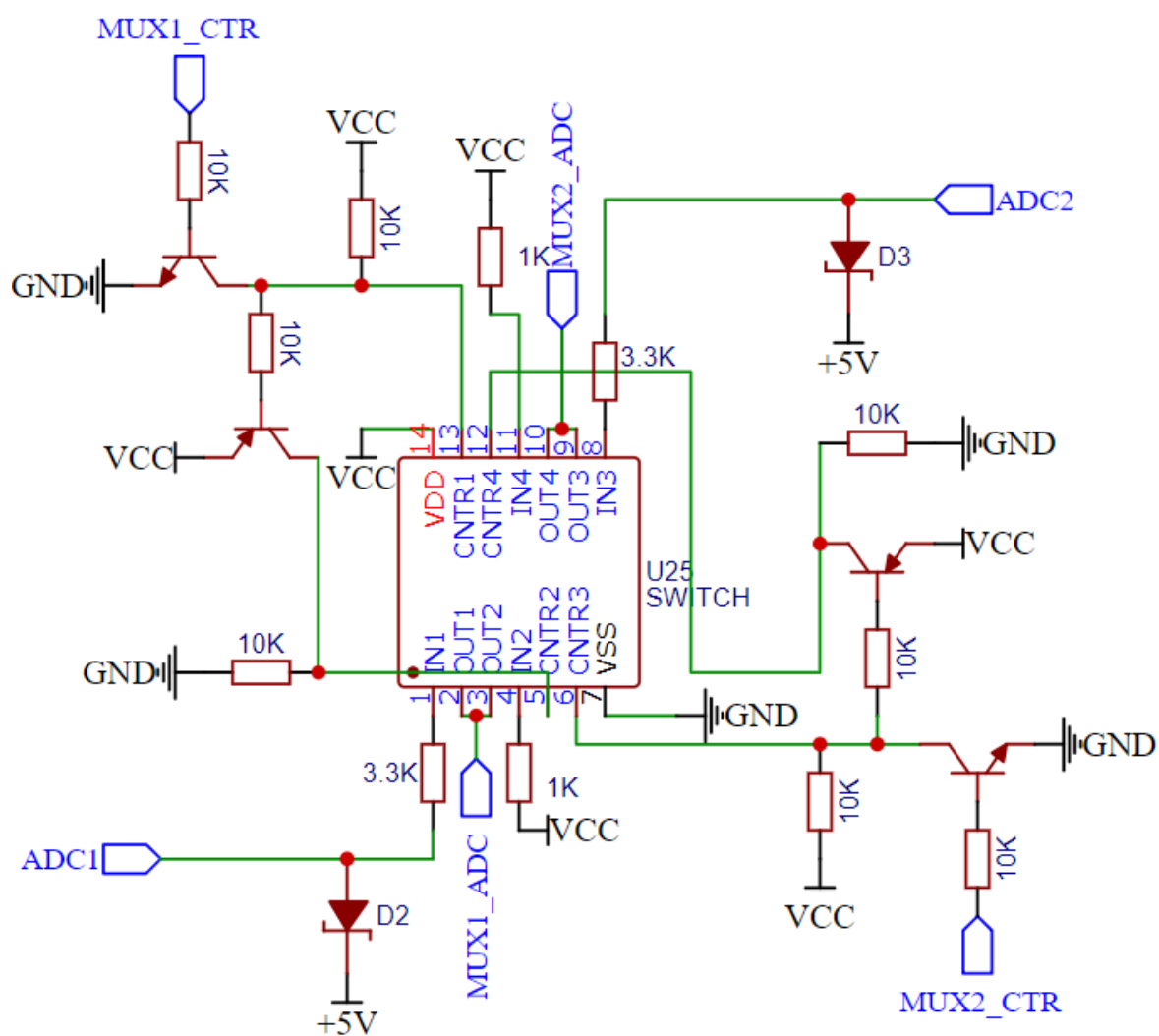
1.1	Schéma vnitřního zapojení IO pinu mikrokontroléru [3, str. 58]	5
2.1	Schéma zapojení multiplexerů k pinům ZIF patice	9
2.2	Schéma děliče napětí na pinu	9
2.3	Zjednodušené schéma obvodu připojeného k 8 pinům ZIF patice	11
1	Úplné schéma obvodu připojeného k 8 pinům ZIF patice	23
2	Úplné schéma zapojení switchu, který přepíná vstup do multiplexerů 1 a 2 mezi ADC pinem a 12V.	24
3	Úplné schéma obvodu step-up převodníku z 5 na 12 voltů. Obvod zahrnuje o ohmové odpory a solder jumper, které umožňují jednoduché odpojení převodníku.	25
4	Zapojení plošného spoje. Žlutou barvou jsou vyznačeny spoje ve vrchní vrstvě, zelenou spoje spodní vrstvy a červenou spoje jedné z vnitřních vrstev. Druhá vnitřní vrstva není na obrázku zachycena.	26

Seznam tabulek

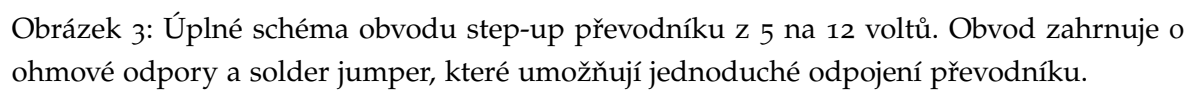
Přílohy

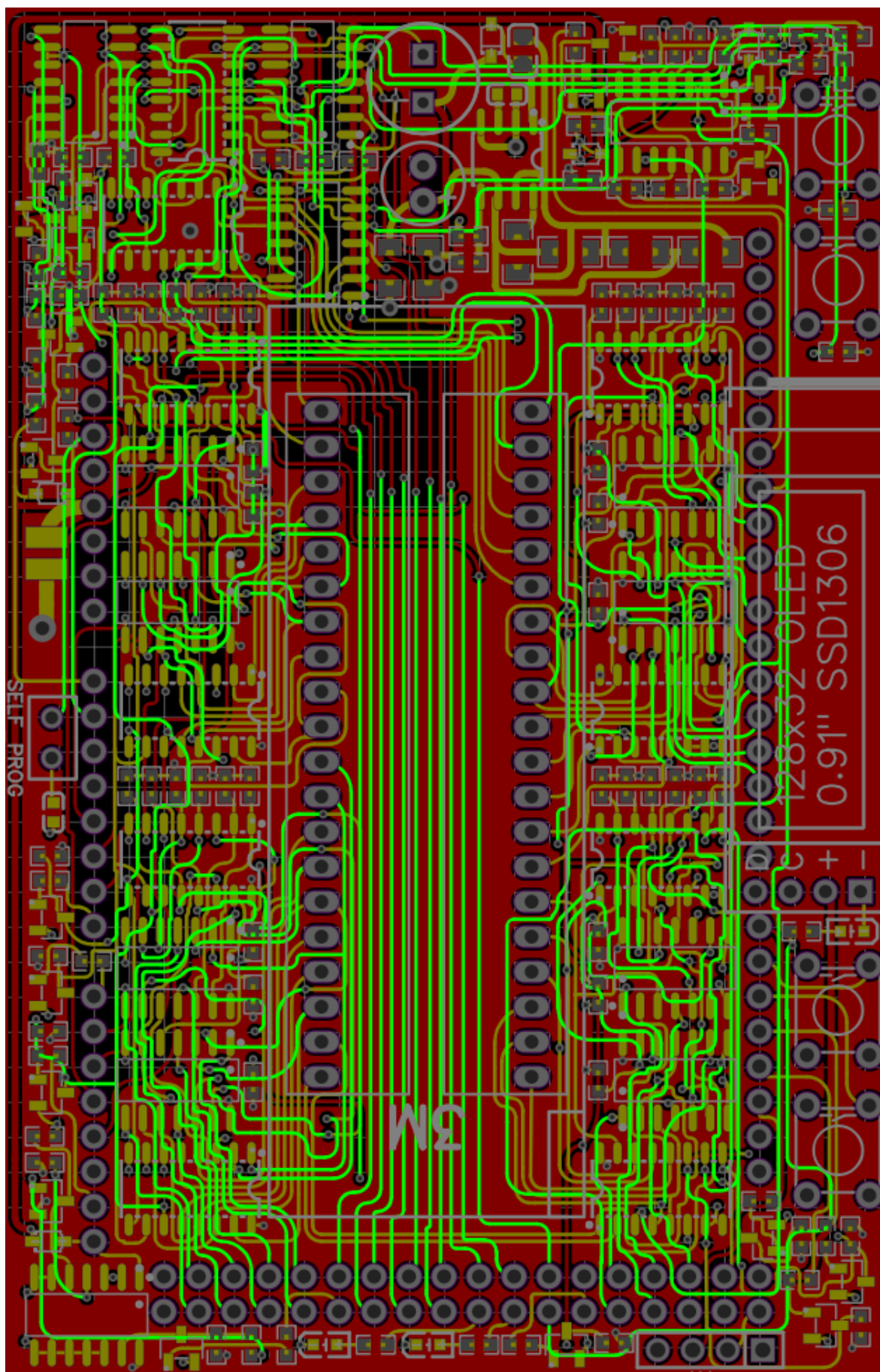


Obrázek 1: Úplné schéma obvodu připojeného k 8 pinům ZIF patice



Obrázek 2: Úplné schéma zapojení switche, který přepíná vstup do multiplexerů 1 a 2 mezi ADC pinem a 12V.





Obrázek 4: Zapojení plošného spoje. Žlutou barvou jsou vyznačeny spoje ve vrchní vrstvě, zelenou spoje spodní vrstvy a červenou spoje jedné z vnitřních vrstev. Druhá vnitřní vrstva není na obrázku zachycena.