



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**EMULACE ZNAKOVÉHO LCD DISPLEJE NA BÁZI
E-INK TECHNOLOGIE**

EMULATION OF CHARACTER-BASED LCD DISPLAY USING E-INK TECHNOLOGY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ RUSÍN

VEDOUcí PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2022

Zadání bakalářské práce



Student: **Rusín Tomáš**
Program: Informační technologie
Název: **Emulace znakového LCD displeje na bázi e-ink technologie**
Emulation of Character-Based LCD Display Using E-Ink Technology
Kategorie: Vestavěné systémy

Zadání:

1. Seznamte se s problematikou řadičů znakových LCD displejů. Zaměřte se na 2 různé typy řadičů, jejichž funkcionalitu podrobně prostudujete.
2. Detailně nastudujte problematiku grafických displejů na bázi e-ink technologie. Zaměřte se především na technické aspekty činnosti jejich řadičů.
3. Zvolte vhodný vývojový kit, který umožní prakticky si vyzkoušet práci s LCD i e-ink displeji.
4. Pro displej na bázi e-ink technologie navrhnete a na obvodové úrovni zrealizujete vhodný typ propojovací desky (na bázi MCU nebo FPGA), která umožní připojení e-ink displej namísto původního LCD displeje.
5. K propojovací desce z bodu 4) zadání vytvořte modulární obslužnou knihovnu včetně uživatelského rozhraní (API), která bude emulovat vzhled a funkcionalitu původního LCD displeje za použití e-ink technologie
6. Hardwarovou realizaci, knihovnu i API pečlivě zdokumentujte a vhodným způsobem demonstруйте funkčnost navrženého řešení.
7. Zhodnoťte dosažené výsledky a diskutujte možnosti dalšího rozšíření.

Literatura:

- Dle pokynů vedoucích.

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

Vedoucí práce: **Šimek Václav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 29. října 2021

Abstrakt

Cílem této bakalářské práce je navrhnout a vytvořit jednoduchou propojovací desku plošných spojů. Deska bude osazena mikrokontrolérem, který bude přijímat komunikační signály původně určené pro znakové LCD displeje. Komunikace může být paralelní nebo sériová, v podobě rozhraní typu SPI a I²C. Tyto signály přijme a převede na signály pro displej typu E-Ink. Na něm se zobrazí informace ve stejném formátu, jako byly určeny pro LCD displeje. E-Ink displej nahrazuje displeje typu LCD bez nutnosti změny původní knihovny nebo původního komunikačního rozhraní. Tento způsob nahrazení je vhodný v případě zničení znakového LCD displeje, kdy za něj není možnost sehnat odpovídající náhradu. Případně pokud je osazení E-Ink displeje vhodnější.

Abstract

The aim of this bachelor thesis is to design a simple printed circuit board interconnect. The board is planned to be equipped with a microcontroller that will receive communication signals originally intended for LCD character displays. The communication may be parallel or serial, in the form of SPI and I²C type interfaces. These signals are planned to be received and converted to E-Ink display signals. That display will show information in the same format as it was designed for LCD displays. The E-Ink display replaces LCD displays without the need of changing the original library or the original communication interface. This method of replacement is suitable when a LCD character display is destroyed and a suitable replacement cannot be found, or when the E-Ink display is more suitable.

Klíčová slova

E- Ink displeje, znakové LCD displeje, emulace vzhledu, komunikace, SPI, I²C, Arduino

Keywords

E- Ink display, character LCD display, emulation of the look, communication, SPI, I²C, Arduino

Citace

RUSÍN, Tomáš. *Emulace znakového LCD displeje na bázi e-ink technologie*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Emulace znakového LCD displeje na bázi e-ink technologie

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Tomáš Rusín
9. května 2022

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Václavu Šimkovi za odbornou pomoc při řešení této práce.

Obsah

1	Úvod	4
2	Znakové LCD a E-Ink displeje	6
2.1	Technologie znakových LCD displejů	6
2.1.1	Výběr jednotlivých LCD displejů	7
2.2	Technologie displejů E-Ink	8
2.2.1	Použití E-Ink displejů v praxi	10
2.2.2	Výběr jednotlivých E-Ink displejů	11
2.2.3	Funkce řadiče E-Ink displeje	12
3	Druhy komunikace s periferními zařízeními	13
3.1	Paralelní komunikace	13
3.1.1	Funkce řadiče LCD displeje	13
3.2	Sériová komunikace	14
3.2.1	Komunikace typu SPI - Serial Peripheral Interface	15
3.2.2	Převodník SPI komunikace	15
3.2.3	Komunikace typu I ² C - Inter-Integrated Circuit	16
3.2.4	Převodník I ² C komunikace	17
4	Vytvoření propojovací desky	19
4.1	Výběr vývojového kitu	19
4.2	Testování v simulátoru	20
4.3	Vytvoření návrhu propojovací desky	20
4.4	Realizace desky a výběr jednotlivých součástek	22
4.5	Postup nahrání bootloaderu a zdrojového kódu	23
5	Implementace obslužné knihovny	26
5.1	Obslužná knihovna	26
5.1.1	Přijímání paralelní komunikace	27
5.1.2	Přijímání I ² C komunikace	27
5.1.3	Přijímání SPI komunikace	27
5.1.4	Výpis na E-Inkový displej	28
5.1.5	Podpora funkcí displeje	29
5.2	Vytvoření nových fontů	29
5.3	Instalace	30
5.4	Testování	30
5.5	Vhodné další rozšíření	32

6 Závěr a zhodnocení	33
Literatura	34
A Obsah přiložené SD karty	36
B Schémata desky	38

Seznam obrázků

2.1	Schéma technologie znakového LCD displeje	7
2.2	Použití znakového LCD displeje v zabezpečovacím systému	8
2.3	Schéma technologie E-Ink displeje	9
2.4	E-Ink displej použitý jako cenovka v obchodě	10
2.5	Porovnání desek od výrobce Waveshare (nahore) a Pervasive displays (dole)	11
2.6	Ukázka instrukce pro E-Ink displej	12
3.1	Ukázka instrukce pro LCD displej	14
3.2	Schéma zapojení SPI převodníku	16
3.3	Schéma zapojení I ² C převodníku	17
4.1	Ukázka testovacího zapojení pro paralelní komunikaci	21
4.2	Vlevo je první prototyp desky s úpravami, vpravo druhý prototyp desky, který správně nefunguje	24
4.3	Schéma zapojení pro nahrání bootloaderu	25
5.1	Testovací výpis znaků na SPI komunikaci	31
5.2	Vlevo je font displeje DEM16216SYH, vpravo pro displej RC1602B	31
B.1	PCB návrh propojovací desky	38
B.2	Schéma propojovací desky první strana	39
B.3	Schéma propojovací desky druhá strana	40

Kapitola 1

Úvod

Znakové LCD displeje jsou poměrně stará a časem vyzkoušená technologie. Dlouhodobě se využívají v komerční sféře pro zobrazování krátkých, formátově jednodušších informací. Znakové LCD displeje, dále jen LCD displeje, se používají například jako displeje kalkulaček, různých měřících přístrojů, digitálních hodinek, tiskáren, zabezpečovacích systémů a celkově v mnoha odvětvích spotřební elektroniky. Jejich využití však může být limitováno. V dnešní době se kladou čím dál větší nároky na energetickou úspornost zobrazovacích panelů. Zároveň je snaha minimalizovat fyzické rozměry displeje a zařízení, do kterých se montují. Fyzické rozměry zařízení mohou hrát velkou roli při rozhodování, zda se pro implementaci v daném problému dají použít či nikoliv. Dlouhodobá technická podpora a údržba LCD displejů může být problémová. Některá zařízení, která tyto displeje využívají nemusí být s inovovanými verzemi LCD displejů kompatibilní. Zařízení by tak v případě poruchy displeje nemohla dále fungovat. V jiných případech může být inovace zobrazovací technologie vhodná pro celkové zlepšení funkčnosti zařízení. Bylo by dobré mít možnost tyto LCD displeje snadno vyměnit za novější v podobě E-Ink displejů.

Cílem bakalářské práce je tedy nahradit nefunkční, nedostupné nebo technologicky zastaralé LCD displeje pomocí displejů typu E-Ink. Toto nahrazení by se mělo odehrávat v podobě připojení nové propojovací desky, která na sobě obsahuje E-Ink displej, na původní místo. Nová deska bude přizpůsobena tak, aby ji bylo možné připojit na stávající komunikační rozhraní. Tato deska bude obstarávat překlad komunikačních signálů přicházejících pro starý LCD displej. Signály se upraví na příkazy pro E-Ink displej, na kterém se vypíšu instrukce a data tak, aby co nejvíce odpovídaly původnímu vzhledu na LCD displeji. V této podobě by tak nebylo nutné zasahovat do původního komunikačního rozhraní ani knihoven LCD displeje. E-Ink displej bude věrohodně zobrazovat původní formát dat a podporovat základní funkce původního displeje.

Téma mě zaujalo hned z několika důvodů. Při úvodní konzultaci s vedoucím bakalářské práce jsme se bavili o možnostech využití E-Inkových displejů. Technologie E-Ink displejů je poměrně nová a přináší řadu inovací v oblasti zobrazovacích panelů. Zároveň má však i svoje limity, které omezují možnosti jejího využití. Právě díky těmto výhodám a limitům mi přišlo jako dobrý nápad použít tento typ displejů pro nahrazování za LCD displeje. Nevyhazovat starou, rozbitou techniku z toho důvodu, že LCD displeje, které na ní zobrazují informace nelze opravit nebo nahradit. Místo toho mít možnost nahradit původní LCD displeje novými E-Inkovými displeji a nadále tak tato zařízení používat. Zároveň mě zaujala možnost vytvořit vlastní desku plošných spojů a pracovat s různými hardwarovými zařízeními, tedy něco mimo rámec klasického studijního programu.

Nejprve jsem v teoretické části popsal, na jakém principu LCD displeje a E-Ink displeje fungují [2](#), jakým způsobem mohou komunikovat se zdrojovým zařízením a jak je tato komunikace realizována [3](#). V následující praktické části je pak podrobně popsáno, jakým způsobem jsem vybral a pracoval s testovacími zařízeními, jak jsem vytvořil desku plošných spojů [4](#), zprovoznil mikrokontrolér osazený na desce a vytvořil knihovnu s fonty [5](#) pro překlad a výpis na E-Ink displeji. Následuje část o testování a zhodnocení dosažených výsledků [6](#).

Kapitola 2

Znakové LCD a E-Ink displeje

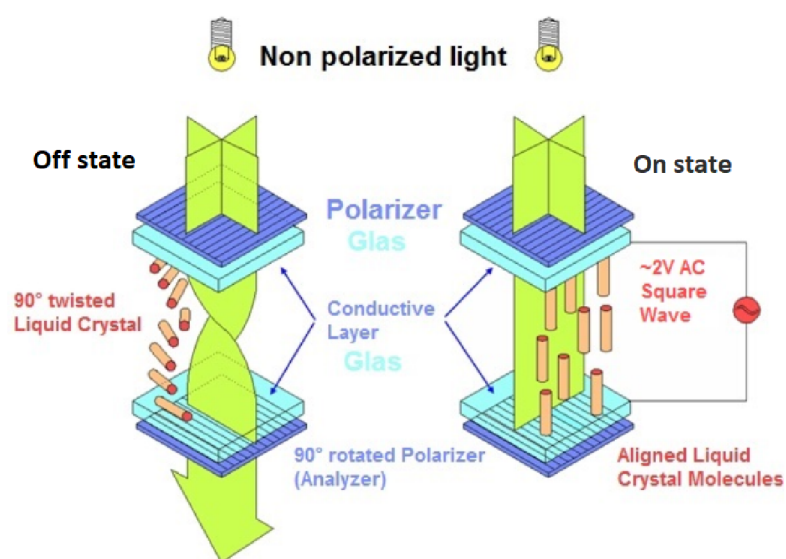
2.1 Technologie znakových LCD displejů

Technologie LCD displejů je založena na principu několika vrstev materiálů a filtrů, které za vhodných podmínek propouštějí světlo nebo ho naopak zadržují. Struktura LCD displeje se skládá ze zadní světelné desky (nebo odrazové desky), která vysílá zdroj světla k uživateli. Světlo, tedy druh elektromagnetického záření kmitá ve všech směrech. Pro jeho blokování, tedy zobrazení znaků, je potřeba do struktury displeje umístit polarizační filtry. Když polarizačním filtrem prochází světlo, v podobě elektromagnetických vln, filtr propouští pouze určitý směr těchto vln a ostatní absorbuje. V struktuře displeje jsou dva polarizační filtry. Jeden je umístěn vzadu před světelnou deskou tak, že propouští vlny v horizontálním směru. Před přední skupinou elektrod v struktuře displeje je umístěn druhý polarizační filtr tentokrát tak, že propouští vlny ve vertikálním směru. Takto vytvořená křížová struktura zajišťuje, že světlo ze zadní podsvícené desky nepronikne k uživateli. Bez vnějšího vlivu nemají světelné vlny možnost se protočit skrze tyto polarizační filtry. Proto, aby se světlo dostalo k uživateli, je potřeba přidat do struktury displeje soustavu elektrod, krycích skel a tekutých krystalů[17].

Nejprve soustava krycích skel spolu s průhlednými elektrodami. Dvě elektrody se nacházejí u každého jednotlivého pixelu. V multiplexovaném displeji jsou elektrody spojeny do skupin. Jedna skupina elektrod je se zdrojem napětí a druhá skupina s odběrem napětí. Skupiny jsou seskupeny tak, aby bylo možné ovládat každý jednotlivý pixel s pomocí unikátní adresy. Děje se tak kvůli finanční úspoře, není nutné ke každému pixelu jednotlivě přivádět vodiče. Ve struktuře následuje zobrazovací filtr. Ten se nachází před přední skupinou elektrod a vytváří šablonu tvarů. Tyto tvary se shodují s tvary, které se budou uživateli zobrazovat v případě, že bude displej zapnutý. Například tvary číslic na hodinkách. Samotná přední elektroda má na sobě drážky ve vertikální poloze. Zadní elektroda má pak drážky v horizontální poloze. Tekuté krystaly jsou umístěny mezi elektrodami a kopírují linie obou drážek v těchto elektrodách. Díky tomuto rozmístění vzniká matice ve tvaru „spirály“. Takto stočené krystaly umožňují průchod světla ze zadní podsvícené desky, skrz polarizační filtry, směrem k uživateli. Tekuté krystaly umožňují ohyb světla mezi filtry 2.1. To značí i pojmenování této technologie, TN tedy „twisted nematic“. Celá tato soustava elektrod a filtrů je schopna ovlivnit každý jednotlivý pixel. Proto pokud je potřeba vypsát nějaký znak, jednotlivé pixely jsou světlu průchodné nebo ne. Při vytvoření elektrického pole na elektrodách se vodivé tekuté krystaly přesunou ze svých drážek do pozic podle směru toku proudu. Rozbije se tím struktura matice a světlo se nebude moci otáčet ve

spirále, nepronikne k uživateli. Pixel bude černý a na displeji se vypíše text. Když se napětí přeruší, krystaly se vrátí do svých pozic a pixely se nezobrazí [6].

Vybrané LCD displeje pro testování mají mřížky tekutých krystalů vytvořené pomocí technologie STN, tedy „Super twisted nematic“. Tyto displeje mají více vrstev tekutých krystalů umístěných v drážkách. Díky tomu jsou schopné „pročít“ světelný tok až o 270° stupňů oproti standardním 90°. Zároveň je možné do multiplexovaných skupin přidat vícero řádků a sloupců. Těmito vylepšeními lze dosáhnout lepších pozorovacích úhlů a větší ostroty obrazu. Displeje také mohou být tzv. STN Positive nebo STN Negative. STN Positive označuje, že znaky jsou tmavé na světlém pozadí, zatímco u STN Negative jsou znaky světlé na tmavém pozadí. Toho je docíleno obrácením ploch, které světlo blokují nebo propouští. Výhodou může být lepší pozorovací schopnost v závislosti na okolním prostředí. Výhodou STN Positive displejů je menší spotřeba elektrické energie, protože ke svému fungování nepotřebují podsvícení. Vybrané displeje jsou STN positive.



Obrázek 2.1: Schéma technologie znakového LCD displeje (zdroj: [1])

2.1.1 Výběr jednotlivých LCD displejů

Při hledání vhodného LCD znakového displeje jsem zjistil, že výběr v této oblasti je poměrně široký. Displeje se odlišují v několika základních parametrech. Nejprve to jsou fyzické rozměry displeje a s tím spojený formát výpisu znaků. Je možné mít displej o jednom, dvou nebo čtyřech řádcích, kdy každý řádek může mít osm, šestnáct nebo dvacet sloupců. Po testování a přečtení různých článků [2] jsem se rozhodl pro displeje, které jsou o velikosti šestnácti sloupců a dvou řádků. Rozměr a formát těchto displejů je nejrozšířenější a nejčastěji se objevoval v různých implementacích. Zařízení jako jsou kalkulačky, měřicí přístroje nebo zabezpečovací systémy 2.2 mají displeje v podobném poměru stran. Přesně tyto typy zařízení jsou vhodná pro možnost instalace náhradní desky s E-Inkovým displejem.

Dle zadání jsem se měl zabývat alespoň dvěma typy řadičů. V nabídce společnosti TME je k dispozici přibližně 20 typů řadičů znakových displejů. Je logické pokrýt co nejvíce zařízení, které mají znakové LCD displeje. Zaměřil jsem se tedy na ty řadiče, které mají nejvíce

variant displejů, tedy alespoň 10 různých druhů displejů s totožným řadičem. Takových řadičů je v nabídce přibližně 7. Po tomto zjištění jsem vybral řadiče typu ST7066U [20] a HD44780 [13]. Řadič HD44780 je neznámější a nejčastěji se vyskytující řadič u znakových LCD displejů vyvinutý firmou Hitachi už v roce 1980. ST7066U se několikrát objevoval použitý v projektech, zároveň displeje s tímto řadičem byly druhé nejprodávanější. Pro řadič typu HD44780 jsem vybral displej RC1602B-GHY¹ od výrobce Raystar. Pro řadič ST7066U je displej DEM16216SYH-LY² od výrobce Display Elektronik.

Po prostudování jejich datasheetů jsem zjistil, že řadiče LCD displejů jsou téměř totožné. Jejich zapojení, průběh komunikace a použité funkce jsou stejné. Odlišují se zejména v použité znakové sadě. Zapojení, popis komunikace a funkce řadiče LCD displeje je více popsáno v kapitole 3.1.1. Při procházení materiálů jsem dále zjistil, že se LCD displeje v projektech a zařízeních nezapojují pouze do paralelního připojení, ale i do sériového. K vybraným displejům jsem tedy pořídil adaptéry komunikací. Ty se připojují mezi vývojovou desku a displej. Deska tak může komunikovat s displejem sériově, i když ten má pouze paralelní rozhraní. Rozhodl jsem se proto rozšířit zadání o podporu pro komunikaci I²C a SPI s tím, že propojovací deska bude schopná přijímat tyto komunikace.



Obrázek 2.2: Použití znakového LCD displeje v zabezpečovacím systému

2.2 Technologie displejů E-Ink

E-Paper neboli E-Ink displej je typ zobrazovací technologie připomínající klasický list papíru. Je tvořen několika vrstvami průhledného materiálu, mezi kterým se pohybují inkoustové částice. Inkoustové částice jsou uzavřeny v mikrokapslích naplněných průhlednou tekutinou. Jeden pixel displeje je vždy tvořen hned několika mikrokapslemi. Mikrokapsle jsou umístěny mezi průhledné elektrody, nad kterými jsou ještě průhledné krycí desky. Na elektrodách se generuje napětí. Kladné nebo záporné, v závislosti na požadovaném zobrazení.

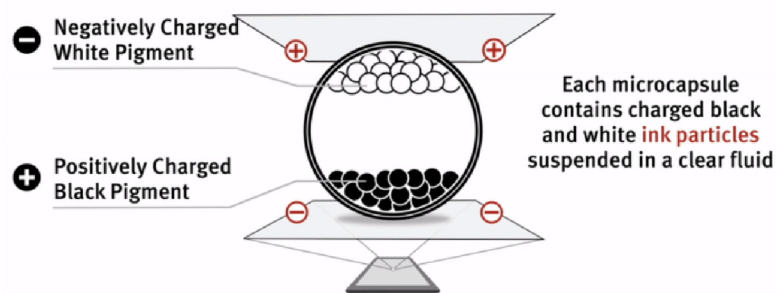
¹ <https://www.tme.eu/cz/details/rc1602b-ghy-csxd/>

² <https://www.tme.eu/cz/details/dem16216syh-ly/>

Existují dva typy displejů. První typ má v kapslích uzavřen pouze jeden druh inkoustových částic, které jsou obarveny na černo. Když se změní napětí inkoust se seskupí do středu mikrokapsle a na displeji se zobrazí černá. Při použití opačného napětí se rozestoupí do stran mikrokapsle a umožní tak prohlédnout na bílé pozadí. Druhý typ displeje má v sobě více druhů inkoustu v závislosti na barevnosti. Černá je v kapslích vždy. Další barvy se rozlišují podle toho, zda je displej stále černobílý nebo barevný. Pokud je displej černobílý má druhý inkoust barvu bílou pro lepší kontrast displeje. V případě dvou barev inkoustových částic v jedné kapsli, jsou barvy nabity opačným nábojem tak, aby bylo možné střídat, která z nich je vidět a která ustoupí do pozadí 2.3. Pokud je displej barevný, barvy jsou nejčastěji žlutá nebo červená, doplněny o bílou a černou. V případě tří a více barev inkoustu, například černá, červená a bílá záleží na tom, jak moc jsou jednotlivé částice nabité. Nabití jednotlivých částic jde regulovat, a tak některé budou přitahovány více než jiné. Zároveň je možné částice odlišit i velikostmi. Menší částice se budou pohybovat rychleji než větší, proto budou na popředí dříve než větší částice. Díky těmto vlastnostem lze barvy střídat tak, aby vznikaly různé barevné útvary. Změny barev lze dosáhnout i přidáním barevných filtrů. Od tohoto přístupu se ale odstupuje, jelikož přidání filtrů způsobovalo špatnou čitelnost [14].

První nápad na E-Inkový displej se zrodil ve firmě Xerox PARC už v roce 1970, ale nikdy nebyl realizován. Prototyp E-Inkového displeje byl vytvořen až v roce 1995 na univerzitě MIT, kdy jeden ze studentů dostal nápad na vytvoření elektronické knihy, která by byla schopna změnit svůj obsah při stisknutí tlačítka. V roce 1997 pak založil s kolegy firmu E Ink Corpartion, která si tyto displeje patentovala [29]. Firma je momentálně hlavní vývojářskou firmou v této oblasti a určuje budoucnost E-Ink displejů. Jejich poslední novinkou je displej založený na technologii míchání barev systémem CMYK. Inkoustové částice v mikrokapslích mají barvu Cyan, Magenta, Yellow a místo Key, černé barvy, je zde bílá. Kombinací těchto čtyř barev inkoustu lze vytvořit plný barevný gamut obsahující i 8 základních barev. Díky tomu lze E-Inkové displeje rozšířit do dalších oblastí spotřební elektroniky [14].

Oproti technologii LCD displejů jsou E-Ink displeje výrazně jednodušší v konstrukci a tím i snazší na výrobu. Velkou výhodou oproti LCD displejům je, že E-Ink displeje jsou bistabilní. Spotřebovávají energii pouze při změně zobrazení. Ve chvíli, kdy je obraz vykreslen může být E-Ink displej vyjmut z obvodového zapojení. Má dobré pozorovací úhly a dobře se čte i na přímém slunci. Limitem je obnovovací frekvence displeje. Ta je závislá na tom, jak rychle se dokáže inkoust v mikrokapslích přesunout.



Obrázek 2.3: Schéma technologie E-Ink displeje (zdroj: [14])

2.2.1 Použití E-Ink displejů v praxi

V současnosti se E-Ink displeje vzhledem ke svým vlastnostem používají převážně k zobrazování statických nebo méně často měnících se informací. Lze je najít v zařízeních, kde je důraz kladen na nízkou spotřebu energie. Sledování E-Ink displeje není tak náročné pro oči, jelikož nevyzařuje žádné nechtěné světelné záření. Můžeme se s nimi setkat v jednoduchých mobilních telefonech nebo hodinkách. Další, asi nejznámější použití pro tyto displeje, což jsou čtečky knih, časopisů a textů. E-Ink displeje se dají použít i pro různé digitální zápisníky, díky jejich schopnostem dobře napodobit papír. Struktura vrchní vrstvy displeje neomezuje zobrazovací možnosti, a tak může být hrubší, tedy klást odpor při psaní, podobně jako reálný papír. Magnetické pero, používané v těchto zápisnících, dokáže inkoust přitáhnout nebo odpudit a zanechat za sebou stopu. Vzniká tak jednoduchý způsob digitalizace vlastnoručně psaných poznámek. Vyrábějí se i flexibilní E-Ink displeje, kdy je jejich podobnost k reálnému papíru ještě věrohodnější.

Existují i více extravagantní využití E-Ink displejů. Dobrým příkladem je použití E-Inkových displejů jako cenovek v obchodech. Zobrazování ceny je pouze statické, moduly jsou napájeny z baterie, díky tomu vydrží pracovat dlouhou dobu. Pokud je potřeba některé ceny změnit, není obsluha supermarketu nucená každou jednotlivou cenovku vyměňovat. Displeje jsou připojeny na WiFi a personál dokáže hodnoty cenovek vzdáleně přepsat. Tento typ cenovek se používá například v brněnských obchodech řetězce Billa nebo Albert 2.4. Nedávno se povedlo úspěšně implementovat E-Inkové displeje do karoserie prototypu elektromobilu³. Díky tomu mohou jednotlivé panely karoserie měnit barvu dle přání zákazníka, avšak prozatím pouze v monochromatických odstínech. V podobném duchu se nese i využití v oděvním průmyslu, kdy se podařilo E-Ink displeje implementovat do obuvi. Díky tomu může mít zákazník každý den boty s jiným designem.



Obrázek 2.4: E-Ink displej použitý jako cenovka v obchodě (zdroj: [23])

Z vypsaných příkladů použití je zřejmé, že největším omezením E-Ink displejů je obnovovací frekvence. Proto se na tento problém, spolu s rozšířením barevné palety, momentálně ubírá velká část vývoje. Důraz se klade na nové materiály a technologii přeskupení inkoustu tak, aby se byl schopen rychleji pohybovat v kapsli. Bohužel vzhledem k fyzikálním zákonům

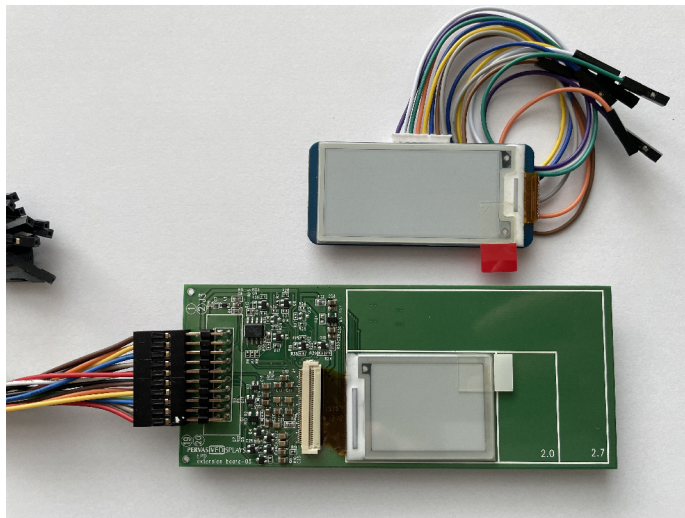
³ <https://www.bmw.com/en/events/ces2022/ixflow.html>

se pravděpodobně nikdy nedokáže frekvence displejů snížit natolik, aby dokázaly nahradit ostatní typy displejů.

2.2.2 Výběr jednotlivých E-Ink displejů

Při výběru výrobce E-Inkových displejů jsem narazil nejčastěji na tři výrobce. Firmy Pervasive displays, Waveshare Electronics a Midas Displays. Všechny mají širokou nabídku E-Ink displejů od barevných (červené, žluté, modré) po černobílé, různých velikostí.

Firma Midas displays nabízí kromě E-Inkových displejů i jiné, například TFT, LCD nebo OLED displeje. Pro některé z nich má v nabídce testovací desky, z nichž některé jsou však dlouhodobě nedostupné. Pro E-Ink displeje se mi nepodařilo najít žádnou testovací desku. Zároveň nenabízí jakékoliv základní zdrojové kódy pro obsluhu jejich displejů. Podpora pro vývoj na displejích od firmy Pervasive je značně lepší. V nabídce jsou dvě testovací desky EXT2 a EXT3. Verze EXT3 je nejnovější verzí z roku 2021/2022, kterou jsem k dispozici neměl. Na testování jsem měl zapůjčenou verzi EXT2 v první generaci [8]. Tato deska má pouze jednu velikost a je k ní možné připojit vícero typů displejů. Tedy i pro malý displej je potřeba poměrně velká deska. Rozměry desky by mohly být omezující, pokud by se používala v pokročilých fázích vývoje nějakého zařízení 2.5. Propojit testovací desku s vývojovou deskou je poměrně komplikované. Jelikož je možné na desku připojovat E-Ink displeje s vícero typy řadičů, je potřeba jejich hodnoty nastavovat. Návodů a příložených zdrojových kódů pro obsluhu nejsou přehledné. Pokoušel jsem se zprovoznit tento typ desky na obou vývojových platformách, které jsem vybral, ale nepovedlo se mi na displej cokoliv vypsát.



Obrázek 2.5: Porovnání desek od výrobce Waveshare (nahore) a Pervasive displays (dole)

Firma Waveshare oproti ostatním vyniká ve velice dobré podpoře koncového uživatele. Má poměrně rozsáhlou a přehlednou Wiki stránku pro jednotlivé typy E-Ink displejů a jejich testovacích desek. Ke každému typu displeje je dodávána testovací deska, která rozměry kopíruje velikost E-Ink displeje. Toto řešení mi přijde pro testování a vývoj nového zařízení jako lepší. Vývojová deska má na sobě otvory pro uchycení, proto bych si dovedl představit prototypy zařízení používající přímo tuto desku 2.5. Ke každému displeji je k dispozici základní obslužná knihovna [25]. Ta je dostupná v několika modifikacích pro všechny zá-

kladní výrobce vývojových desek jako je Arduino, Espressif systems, ST Microelectronics nebo Raspberry Pi. Díky široké podpoře je tak možné ovládat testovací desky i vzdáleně pomocí Bluetooth nebo WiFi. Některé desky jsou přímo osazeny mikrokontroléry, které tyto funkce podporují, jako je ESP8266/ESP32. K nim jsou pak dodávány speciální propojovací flex kabely na napojení na E-Ink displejů. Jiné desky nabízí univerzální připojení na více rozměrů E-Inkových displejů. Nabídka je široká a je tak poměrně jednoduché s pomocí testovací a vývojové desky vyzkoušet základní výpis na displeji. Výhodou firmy Waveshare je také její rozšířenost.

Displeje od této společnosti jsem viděl použité v mnoha projektech a diskusních fórech. Z těchto důvodů jsem se rozhodl vybrat displeje právě od této společnosti. Pro základní testování jsem zvolil displeje o velikosti 2,13“ a 2,9“ palců spolu se základní testovací deskou. Tyto velikosti totiž nejvíce odpovídají velikosti zvolených LCD displejů.

2.2.3 Funkce řadiče E-Ink displeje

Existují dvě základní varianty řadiče, a to externí a interní zapojení. Značení tohoto zapojení se může u různých výrobců lišit, například firma Pervasive používá eTC a iTC, Waveshare epd. Displej s interním řadičem(iTC/epd) má přímo ve své konstrukci zapojenou soustavu časovačů a posuvných registrů, které pouze přijímají příkazy a bitmapové soubory z hlavního mikrokontroléru, který je osazený na vývojové desce. Displej s externím řadičem(eTC) přijímá řídicí signály přímo od mikrokontroléru, který se používá na celkový vývoj programu. Na displeji nejsou žádné logické konstrukce, které by přijímali vyšší komunikace. V tomto případě je nutné obsloužit chování displeje, jeho řídicí fáze, rámec, ve kterém se data vypisují a hodinový signál. Displeje od firmy Waveshare fungují na principu interního řadiče [25].

Komunikace probíhá přes rozhraní SPI. VCC, GND jsou vodiče pro připojení periferního zařízení napětí a uzemnění. CS vodič určuje, kterému zařízení jsou data posílána. DC vodič nastavuje, zda se zapisují data nebo instrukce. CLK označuje vodič s hodinovým signálem. DIN je obdoba vodiče MOSI, tedy mikrokontrolér přes něj posílá data na E-Ink displej. Vodiče BUSY a RST se odlišují od klasické SPI komunikace. Vodič BUSY signalizuje, kdy je na E-Ink displej zapisováno a práce jeho řadiče by neměla být přerušována jinými příkazy. RST resetuje funkcionalitu na displeji a uvede jej do počátečního stavu. Datasheet obsahuje jednotlivé instrukce, s kterými E-Ink displej pracuje. Pro popis instrukcí je vybrána funkce na přechod E-Ink displeje do úsporného režimu 2.6. Instrukce mají osmibitový formát, protože komunikace SPI taktéž pracuje v osmibitovém formátu. Zobrazuje nastavení jednotlivých pinů pro vykonání instrukce. Ovládání instrukcí je dostupné pomocí obslužné knihovny.

R/W#	D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description	
0	0	10	0	0	0	1	0	0	0	0	Deep Sleep mode	Deep Sleep mode Control	
0	1		0	0	0	0	0	0	0	A ₀			A[0] :
												0	Normal Mode [POR]
												1	Enter Deep Sleep Mode

Obrázek 2.6: Ukázka instrukce pro E-Ink displej (zdroj: [24])

Kapitola 3

Druhy komunikace s periferními zařízeními

Při komunikaci nějakého mikrokontroléru s periferním zařízením je možnost přenášet data fyzickou cestou nebo bezdrátově. Bezdrátový přenos je realizován nejčastěji pomocí technologií založených na principu Bluetooth nebo WiFi. Pro fyzický přenos dat se pak používají nejčastěji dva typy komunikace a to paralelní nebo sériová. Komunikace sériová je použita u E-Ink displeje s využitím rozhraní SPI. LCD displeje nejčastěji používají paralelní komunikace, ale zároveň je lze připojit i přes sériové rozhraní SPI nebo I²C.

3.1 Paralelní komunikace

Princip paralelní komunikace je založen na přenášení každé jednotlivé hodnoty bitu, pomocí separátního vodiče. Pokud je potřeba přenést informace o velikosti osm bitů, tak je nutné k zařízení připojit minimálně osm vodičů a další vodiče potřebné pro signalizaci. Z toho lze jednoduše odvodit, že pro větší datovou náročnost by bylo potřeba ještě více vodičů. To je situace, které se snažíme spíše vyhnout. Proto se v praxi častěji používá komunikace sériová. Avšak velkou výhodou paralelní komunikace je její rychlost. Díky separátním vodičům jsme schopni přenášet všechna data nebo instrukce v jeden časový okamžik.

3.1.1 Funkce řadiče LCD displeje

Mnoho znakových LCD displejů, které jsou dostupné, využívá paralelní komunikaci. Nejčastější formou této komunikace je šestnáctipinová varianta. Vodiče jsou VCC, GND, V0, RS, R/W, E, D0 až D7, LED+ a LED-.

První dva vodiče VCC a GND jsou pro připojení displeje na napětí a uzemnění. V0 je vodič nastavující kontrast displeje, měl by být připojen na potenciometr, aby šlo jeho hodnoty regulovat. Vodič RS rozlišuje, zda se zapisují do registru data nebo instrukce. Vodič R/W určuje, zda se bude na displej zapisovat nebo se z něj bude číst. Nastavení logických hodnot těchto vodičů je uvedeno v datasheetech příslušných displejů. Nejdůležitějším vodičem pro komunikaci je Enable, pomocí kterého se signalizuje přicházející komunikace. Datové vodiče D7 až D0 přenáší instrukce a hodnoty znaků pro výpis na displeji. Hodnoty znaků jsou v binárních ASCII hodnotách. Tedy například písmeno A je zapsáno jako 01000001 (MS-LSB), jeden vodič pro jednu hodnotu. Vodiče LED+ a LED- je připojení pro podsvícení displeje. Katoda má značení LED-, anoda je LED+. Komunikace probíhá v časovém rámci, který je zobrazen na přiloženém obrázku. Nejprve jsou nastaveny hodnoty RS a R/W. Poté je vodič

Enable přesunut z logické 0 do 1 čímž signalizuje, že je možno nastavit hodnoty na vodiče D7 až D0. Data se nastaví a vodič Enable přechází z logické hodnoty 1 do hodnoty 0. V tu chvíli se začínají číst hodnoty na datových vodičích [9].

Displeje jsou schopny pracovat ve čtyřbitovém nebo osmibitovém módu. Osmibitový mód je rychlejší, protože jsou všechna data přenesena v jednom časovém úseku. Pokud se komunikuje ve čtyřbitovém módu jsou potřeba dva cykly přenosu. Bity se přenášejí postupně ve skupinách. Nejvýznamnějším bitem je D7, tedy v prvním cyklu se přenesou část D7 až D4, v druhém cyklu pak D3 až D0.

Řadič HD44780 se ve všech zmíněných částech shoduje s řadičem ST7066U. Drobné odlišnosti jsou v časování některých přechodů vodičů. Na používání displejů to ale nemá žádný větší význam. Další odlišnosti se nachází v použité znakové sadě, kde některé hodnoty ASCII zobrazují jiné symboly, než je standardem. Tuto odlišnost jsem zahrnul do implementace a vytvořil dvě varianty fontů. Dále se v datasheetu nachází tabulka s instrukcemi k ovládání displeje. Na obrázku 3.1 lze vidět instrukci pro odstranění obsahu displeje. Zobrazuje hodnoty jednotlivých vodičů, které je potřeba nastavit pro provedení dané instrukce. Tabulka instrukcí je opět u obou řadičů shodná. Samotné fungování řadičů je tvořeno skupinou posuvných registrů, klopných obvodů a čítačů a časovačů. Jejich konkrétní zapojení lze zjistit v příloženém datasheetu. Pro ovládání displejů jsem použil knihovnu LiquidCrystal.h dostupnou ve vývojovém prostředí Arduino IDE. Tato knihovna fungovala pro obě varianty řadičů. Návod na propojení s vývojovou deskou je popsán v komentářích zdrojového kódu.

Instruction	Instruction Code										Description	Execution time (fosc=270Khz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "00H" to DDRAM and set DDRAM address to "00H" from AC	1.53ms

Obrázek 3.1: Ukázka instrukce pro LCD displej (zdroj: [13])

3.2 Sériová komunikace

Princip sériové komunikace je založen na přenesení všech potřebných vícebitových informací, pomocí jednoho vodiče. Sériová komunikace se dělí na dva základní typy. Komunikace synchronní a asynchronní. Při synchronní komunikaci se zároveň s daty přenáší na jiném vodiči i hodinový signál. Ten udává časový údaj, kdy se na datovém vodiči objeví další bit a zároveň kdy byl celý bit přenesen a je možné přečíst jeho hodnotu. Asynchronní komunikace nepoužívá další vodič s hodinovým signálem. Přijímací zařízení si generuje hodinový signál samo. Pro správnou funkčnost je důležité, aby přijímací zařízení bylo synchronizováno s vysílacím zařízením. Toho se docílí pomocí stejné frekvence a stejné fáze komunikace. Tyto hodnoty se nejčastěji dohodnou před začátkem komunikace. Pokud se synchronizace provádí dostatečně často, nedochází k chybám v přenosu. V případě, kdy by mohlo k chybám v přenosu dojít, dá se jim zabránit například použitím paritních bitů. Příkladem asynchronní komunikace je UART příkladem synchronních komunikací je SPI a I²C [5].

3.2.1 Komunikace typu SPI - Serial Peripheral Interface

SPI je druh sériové komunikace, používající se k přenosu dat po sériové lince a připojování periferních zařízení. Zařízení připojená na sběrnici SPI jsou dvojího typu, Master nebo Slave. Zařízení typu Master jsou nejčastěji mikrokontroléry, které získávají data z periferních zařízení. Master generuje hodinový signál, začíná s komunikací a řídí její proces. Na jedné sběrnici SPI je pouze jedno zařízení Master. Zařízení typu Slave jsou nejčastěji periferní zařízení, s kterými Master komunikuje. Přijímají instrukce, případně posílají data zpět na Master.

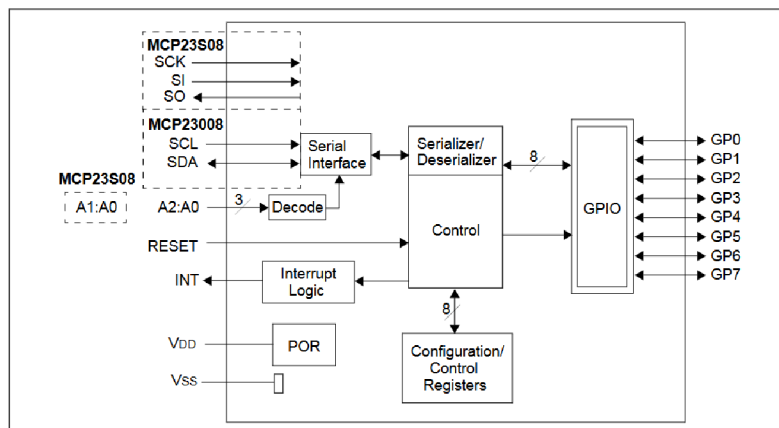
Sběrnice se skládá z šesti vodičů. VCC, GND jsou vodiče pro připojení periferního zařízení na napětí a uzemnění. Komunikace probíhá na dvou datových vodičích MISO a MOSI. SPI má specifikaci full duplex, což znamená, že v každém okamžiku probíhá komunikace oběma směry. Na každém zařízení jsou vždy dva posuvné registry, kdy z jednoho se ve stejný čas data posílají pryč a do druhého se přijímají. Vodič označený MISO znamená Master In Slave Out. Tedy na tomto vodiči vycházejí data z periferního zařízení a přichází na mikrokontrolér. Vodič označený MOSI znamená Master Out Slave In. Na tomto vodiči vycházejí data z mikrokontroléru a přichází na periferní zařízení. Na vodiči s označením SCK se generuje hodinový signál, který řídí celou komunikaci. Šestý vodič má značení SS, ten určuje, se kterým zařízením typu Slave bude probíhat komunikace. Komunikace na sběrnici probíhá vždy pouze s jedním zařízením typu Slave. Jeho hodnota se nastavuje jako první, signálizací do hodnoty logické 0. Ostatní zařízení na sběrnici mají hodnotu tohoto pinu v logické jedničce. Poté Master začne na vodiči SCLK generovat hodinový signál. V té chvíli začíná proudit komunikace na vodičích MISO a MOSI. Když se data přenáší, velikost v jednom cyklu je 8 bitů. Po jejich přenesení je možné data přečíst a nastavit nové hodnoty pro další přenos. Komunikace je ukončena přerušáním hodinového signálu a nastavením vodiče SS na logickou hodnotu jedna [5].

Vztah mezi hodinovým signálem a daty se určuje dvěma konfiguračními bity, které se označují jako CPOL a CPHA. CPOL udává polaritu hodin. Pokud je hodnota CPOL 0, pak jsou hodiny v klidovém stavu také v hodnotě 0. V opačném případě jsou hodiny označeny jako invertované a mají klidový stav v hodnotě 1. CPHA udává, při které hraně se čtou informace. V logické hodnotě 0 se čte informace při změně z neaktivního stavu do aktivního. Při hodnotě logické 1 se čte v opačném směru. Těmito dvěma bity vznikají čtyři varianty, kdy je možné číst data z datových pinů. Nejčastěji se používá kombinace CPOL = 0 a CPHA = 0. Pokud by se hodnoty lišily, budou zapsány v datasheetu použitého zařízení. Některá periferní zařízení neposílají žádná data nazpět, proto u nich není nutné připojovat vodič MISO [18].

3.2.2 Převodník SPI komunikace

Pro zapojení LCD displeje do SPI komunikace jsem použil převodník signálů od firmy Adafruit¹. Tento převodník je schopen konvertovat signály SPI nebo I²C na signály paralelní komunikace. Mezi typem komunikace se vybírá propojením dvou vývodů pinů v obvodu desky. Ne desce je osazen I/O expandér, který rozlišuje mezi komunikací I²C a SPI, data jsou z něj předána na posuvný registr. Posuvný registr funguje na principu postupného převodu signálů. Vždy s náběžnou hranou hodinového signálu se generují data na vodičích pro paralelní komunikaci. Hodnota těchto dat se řídí hodnotou, která se nachází v daný okamžik na sériovém vodiči. Data se postupně přidávají do posuvného registru, když je

¹ <https://www.adafruit.com/product/292>



Obrázek 3.2: Schéma zapojení SPI převodníku (zdroj: [19])

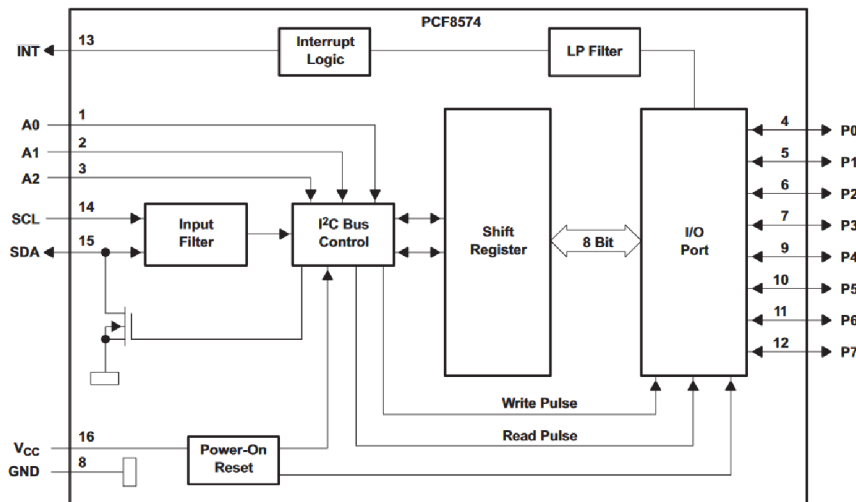
komunikace dokončena jsou jednotlivé hodnoty vypsané na výstupní registr a z něj předány na konkrétní piny paralelní komunikace 3.2.

K převodníku je dodávána obslužná knihovna a testovací programy. Tu je potřeba importovat do vývojového prostředí Arduino IDE. V sekci Sketch > Include Library > Manage Libraries stačí do vyhledávacího pole zadat Adafruit LiquidCrystal a nainportovat knihovnu. Na desce se nachází potenciometr pro nastavení kontrastu displeje. Každý displej může mít hodnoty kontrastu nastaveny jinak a může se tak stát, že po připojení se displej jeví jako nefunkční, i když je pouze potřeba nastavit správné hodnoty. Tento převodník jsem používal pouze pro komunikaci SPI.

3.2.3 Komunikace typu I²C - Inter-Integrated Circuit

Sériová komunikace I²C je synchronní sériové rozhraní. Princip komunikace je založen na čtyřech vodičích VCC, GND, SDA a SCL. VCC, GND jsou vodiče pro připojení periferního zařízení na napětí a uzemnění. SDA neboli Serial Data vodič. Nastavuje Start a Stop podmínky, přenáší se pomocí něj data a potvrzuje se jejich přijetí. SCL neboli Serial Clock je synchronizační vodič, určuje synchronizační signál pro sběrnici. Tato komunikace má specifikaci half duplex. To znamená, že data mohou být přenášena v obou směrech, ale ne ve stejný časový okamžik.

Jednotlivá zařízení se dělí opět do dvou skupin Master a Slave. Oproti komunikaci SPI se odlišuje v tom, že zařízeních typu Master může být na jedné sběrnici vícero. To je možné, protože jednotlivá zařízení na sběrnici jsou rozlišena svými adresami. Adresu periferního zařízení pro použití v I²C komunikaci lze najít v datashetu k němu přiloženému, popřípadě pomocí různých skenovacích skriptů. Adresování na sběrnici může být sedmibitové nebo desetibitové. Zařízení Master generuje startovací hranu SDA. Data jsou přenášena v osmi-bitových rámcích. Prvním bitem prvního rámce je vždy Start bit. Start bit je zaznamenán, pokud je SCL vodič v logické hodnotě jedna a SDA vodič se přesune z logické hodnoty jedna do logické hodnoty nula. Následuje sekvence dalších sedmi bitů, které obsahují adresu zařízení na sběrnici a jeden bit s volbou zápisu nebo čtení na zařízení (R/W je 0 nebo 1). Po nich následuje Acknowledge bit. Tento bit používá zařízení Slave, aby oznámilo, zda úspěšně přijalo předchozí osmibitovou sekvenci. Následuje sekvence osmi bitů obsahujících adresu vnitřního registru. Vnitřní registr nejčastěji obsahuje informace získané z periferního zařízení. Poté přichází opět Acknowledge bit, pro potvrzení přijetí. V poslední osmibitové



Obrázek 3.3: Schéma zapojení I²C převodníku (zdroj: [15])

části jsou pak přenesena samotná data. Těchto osmibitových rámců může být vícero. Za nimi je Acknowledge bit a Stop bit. Stop bit je nastaven, pokud je SCL vodič v logické hodnotě jedna a SDA vodič se přesune z logické nuly do jedničky. Stop bitem je komunikace ukončena [5].

Protože na sběrnici I²C může být připojeno vícero zařízení, nemůžou jednotlivá zařízení uvádět vodiče do logické hodnoty 1. Pokud by totiž některé zařízení na sběrnici přešlo ve stejnou chvíli do logické hodnoty 0, došlo by ke zkratu. Pokud by na vodiči nebyla nastavena logická 0 nebo logická 1, nacházel by se v nedefinovaném stavu. Tyto pull up rezistory jsou připojeny do obvodu mezi komunikační vodiče SDA, SCL a napětí VCC. Na vodičích SDA a SCL je neustále udržována logická hodnota 1. Tato hodnota je tzv. měkká, proud na vodičích je omezen pull up rezistory. Pokud je potřeba začít zapisovat je zařízení Master schopno tuto hodnotu změnit na logickou 0, zkrat nenastane. Po ukončení zápisu se hodnota vrací zpět [18].

3.2.4 Převodník I²C komunikace

Převodník paralelní komunikace na komunikaci typu I²C se skládá z několika základních modulů². Prvním je I²C control bus neboli ovladač sběrnice. Na něj přichází informace z vodičů SDA a SCL spolu s hexadecimální adresou zařízení typu Slave. Ovladač sběrnice při přijmutí dat nejprve nastaví hodnoty zápisu nebo čtení na výstupní registr. Obdržená data jsou z ovladače sběrnice předána posuvnému registru. Posuvný registr je skupina klopných obvodů, které za pomoci hodinového signálu postupně převádí sériovou komunikaci na paralelní. Tento formát dat je následně vypsán na výstupní registr, který předává data na konkrétní piny 3.3.

Převodník se ovládá pomocí vestavěné knihovny Wire.h, která je součástí vývojového prostředí Arduino. Tato knihovna se dá použít nejen pro komunikaci s převodníkem, ale i pro jiná zařízení komunikující přes I²C rozhraní. Pro komunikaci je důležité nastavit hodnoty adresy zařízení. Podobně jako na převodníku Adafruit je i zde potenciometr pro

² <https://www.gme.cz/prevodnik-i2c-display-product-38324>

nastavení kontrastu displeje, jelikož displej nekomunikuje přímo s vodičem V0. Předem definovaná adresa desky je 0x27. Stejnou hodnotu jsem použil pro vytvořenou propojovací desku. Adresa pro komunikaci se dá změnit piny A0 až A2.

Kapitola 4

Vytvoření propojovací desky

4.1 Výběr vývojového kitu

V prvotní fázi práce jsem zvažoval možnost použít pro řešení bakalářské práce školní výukovou platformu FITkit3 neboli Minerva. Tato vývojová deska má tu výhodu, že je osazena jak MCU typu Kinetis-K60, tak FPGA typu Xilinx Spartan-6. Zároveň má dostatečný počet pinů pro připojení všech periférií. Myslel jsem si, že osazení je výhodné, protože by bylo možné použít programovatelné hradlové pole pro přijímání komunikace z LCD displeje. Následně tyto signály převést na signály pro E-Ink a za pomoci mikrokontroléru zapsat požadované funkce nebo vypsát text. Při prvotním testování, zda vůbec budu schopen používat FITkit3 pro provedení funkcí nebo výpis znaků na LCD displeji, jsem zjistil, že podpora pro osazený mikrokontrolér je poměrně malá. Ze školních zdrojů jsem měl k dispozici pouze datasheety a ukázkový „Hello World“ program z předmětu IMP. Z toho jsem pochopil, že pro jakoukoliv další práci bych potřeboval vytvořit vše znovu. Tedy knihovny pro LCD a E-Ink displeje, knihovny pro převodníky komunikace I²C a SPI. K tomu ještě samotnou obslužnou knihovnu pro překlad signálů ze zadání. Vše od registrů, maker, časovačů a globálních funkcí. Řešení ovládnutí všech periferních zařízení na nízké úrovni by bylo časově náročné a zároveň by to značně převyšovalo zadání bakalářské práce, proto jsem tuto vývojovou desku zamítnul. Zvažoval jsem i možnost vyrobit si vlastní platformu obsahující známější FPGA a MCU u kterých by byla podpora knihoven pro periferní zařízení lepší. Během hledání vhodných součástek jsem si však uvědomil, že použití obou logických jednotek je zbytečně složité. Zapojení FPGA pro překlad signálů z LCD displeje by bylo velice výhodné, ale pro zapisování na E-Ink už ne. Jelikož dostatečně silné MCU bude schopno zvládnout obě úlohy, je lepší použít pouze to. Nevznikali by tak větší náklady na výrobu a celý proces bude celkově snazší.

Proto jsem se rozhodl pro všeobecně známé vývojové kity. Kity z rodiny Arduino, kde jsem vybral Arduino Nano¹, a pro kit od firmy Espressif WeMos D1 R32 UNO ESP32², který mi byl zapůjčen z fakulty. Pro testování bylo potřeba vybrat minimálně dvě zařízení tak, abych měl na komunikaci vždy zařízení typu Master a zařízení typu Slave a mohl mezi nimi přeposílat instrukce. Zároveň jsem na obou vývojových deskách vyzkoušel obslužné knihovny pro LCD displeje. Převodníky komunikací SPI/I²C. Testoval jsem výpis na E-Inkové displeje, abych měl jistotu, že jsou desky vzájemně nahraditelné. Testování proběhlo v pořádku. Byl jsem schopen na všech zařízeních provést požadované operace. Díky

¹ <https://store.arduino.cc/products/arduino-nano>

² <https://www.laskakit.cz/wemos-d1-r32-uno-esp32/>

tomu jsem zjistil, že případné použití jednoho z těchto mikrokontrolérů na mnou vytvářené propojovací desce nepřinese nežádoucí problémy.

Při vývoji knihovny jsem narazil na zajímavý poznatek. Když jsem vyvíjel komunikaci typu SPI, bylo potřeba zpomalit rychlost předávání dat z jedné vývojové desky na druhou. Toto omezení je způsobeno rozdílným taktováním obou mikrokontrolérů. Kdy mikrokontrolér na desce Nano pracuje na frekvenci 16 MHz, zatímco na desce D1 R32 pracuje při frekvenci 20 MHz. Tento problém jsem vyřešil přidáním jednoduché funkce `delay()` pro zpoždění odeslání. Tento fakt byl důležitý pro vybrání správného mikrokontroléru na propojovací desku.

4.2 Testování v simulátoru

Když jsem začal testovat, neměl jsem k dispozici ihned všechny potřebné periferie a specifické součástky. Během testování se mi také povedlo poškodit jeden z LCD displejů a jeden USB/UART převodník. Potenciometry nebo některé odpory a kondenzátory, které jsem potřeboval k vytvoření obvodů, jsem také dokupoval až během testování. Abych mohl pokračovat v práci, testoval jsem některé obvody a zdrojové kódy v simulátoru. Nejprve jsem využíval simulátor Wowki³. Je to jednoduché webové rozhraní, které umožňuje vybírat různé druhy vývojových desek a simulovat na nich spuštění zdrojového kódu. K deskám se dají připojovat elektro součástky, různé druhy displejů, servomotory nebo senzory. Bohužel není možné připojit do simulace dvě desky najednou. Zároveň uživatelské rozhraní není ideální, například hodnoty odporů je potřeba měnit v zdrojovém kódu stránky.

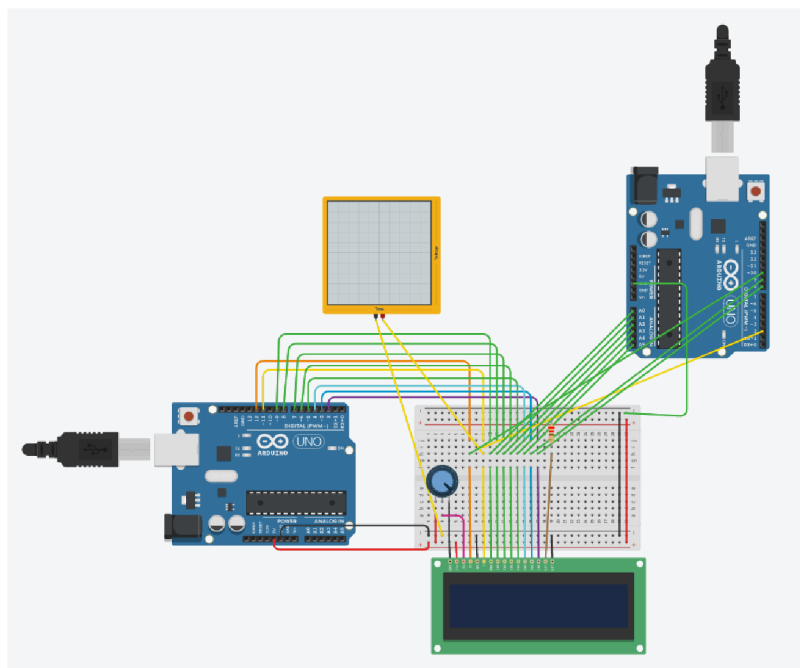
Když jsem potřeboval zapojit vícero desek použil jsem simulátor Tinkercad⁴. V tomto návrhovém prostředí se dají taktéž vytvářet návrhy obvodových zapojení 4.1. Uživatelské rozhraní je kvalitněji zpracováno, než v případě Wowki. Nabídka součástek je poměrně široká a obsahuje vývojové desky, nepájivé pole, LCD displeje a velkou škálu drobných součástek jako jsou kondenzátory, posuvné registry nebo LED diody. Zároveň má vestavěnou podporu pro některé základní knihovny z vývojového prostředí Arduino. Simulátor je po vytvoření uživatelského účtu volně dostupný. SPI komunikace pro vývojové desky nebyla v době vytváření dostupná.

4.3 Vytvoření návrhu propojovací desky

Vytvoření návrhu desky jsem rozdělil do dvou logických částí. První část obsahuje samotný mikrokontrolér a součástky potřebné pro jeho fungování. Výběr mikrokontroléru byl poměrně jednoduchý. Jelikož jsem velkou část testování dělal na vývojové desce Arduino Nano a na vývojové desce WeMos D1 R32, rozhodoval jsem se právě mezi MCU, která jsou osazena na těchto deskách. Deska Nano je osazena mikrokontrolérem Atmega328. Tento mikrokontrolér je založený na architektuře AVR RISC. Na desce Wemos D1 R32 se nachází mikrokontrolér ESP32, architekturou jádra RISC-V. Práce s mikrokontrolérem ESP32 je ve vývojovém prostředí Arduino mírně složitější. Zapůjčený typ desky je specifický a nemá přímou podporu. ESP32 má navíc funkce jako WiFi nebo Bluetooth, které jsem pro realizaci nepotřeboval. Vzhledem k problémům s rozdílnou pracovní frekvencí bylo požadavkem, aby zdrojová a překladová deska pracovala na stejné frekvenci. Atmega 328P při dalším testování splňovala všechny požadavky. Mikrokontroléry z řady Atmega jsou časem prověřené

³ <https://wokwi.com/>

⁴ <https://www.tinkercad.com/>



Obrázek 4.1: Ukázka testovacího zapojení pro paralelní komunikaci

a existuje k nim spousta užitečných tutoriálů a návodů z nichž jsem čerpal. Proto jsem se při prvotním návrhu rozhodl pro tento mikrokontrolér.

Následně bylo potřeba přidat součástky pro mikrokontrolér. Při vytváření návrhu této části jsem se řídil hned několika návody [11], [10]. Součástkami jsou 16 MHz krystal a soustava kondenzátoru propojených tak, aby generovaly hodinový signál pro mikrokontrolér. Zapojení lze vidět na přiloženém datasheetu na jeho druhé straně. Následně jsem přidal součástky pro napojení externích zařízení: tedy dva osmipinové konektory pro paralelní komunikaci (H11, H10), následně jeden čtyřpinový konektor pro komunikaci I²C (H4), jeden šestipinový konektor pro komunikaci SPI (H3), signalizační LED diodu a tlačítko pro resetování mikrokontroléru, vývod na UART rozhraní (H1) pro nahrání kódu na mikrokontrolér a dva jednoduché dvoupinové konektory (H5, H6), které fungují, jako signalizační rozlišení typu komunikace mezi paralelní, SPI nebo I²C. Vytvoření této části návrhu bylo poměrně snadné, jelikož neobsahovala velký počet součástek.

Druhá logická část se skládá ze součástek potřebných pro správné fungování E-Ink displeje. Protože jsem pro realizaci bakalářské práce rozhodl použít E-Ink displeje od firmy Waveshare Electronics, při návrhu jsem se řídil zejména jimi zveřejněnými datasheety [24] a schémata zapojení [26]. Jako vzor jsem si vybral dodávanou testovací desku k displeji. Oproti ní je návrh mírně jednodušší, protože neobsahuje signální IDC konektor pro připojení na vývojovou desku. Základním prvkem návrhu je dvacetičtyřpinový konektor FPC pro připojení displeje. Ten bylo poměrně složité najít. Pravděpodobně je konstruován speciálně pro výrobce Waveshare. Stejný typ se nedá sehnat a musel jsem víceméně odhadnout a vybrat alespoň podobný. Poté bylo potřeba navrhnout obvodové zapojení pro RESET displeje, aby v případě nereagování displeje nebo nutnosti synchronizace bylo možno nastavit displej na základní stav. Bylo potřeba vybrat správný převaděč napětí, protože pracovní napětí pro mikrokontrolér je 5 V, zatímco displej pracuje při napětí 3.3 V. Stejně tak bylo zapotřebí přidat převodník napětí pro komunikační piny přenášející data. Zde je důležité

zkontrolovat připojení na převaděč. Na vodič VCCB musí jít větší napětí než na vodič VCCA, jinak by mohlo dojít k poškození převaděče. Ještě je potřeba obvodové zapojení pro dočasné zvýšení napětí, které se používá pro smazání obsahu na displeji. Při vytvoření svojí desky jsem se snažil dodržet hodnoty v zapojení obvodu totožné, jako je uvedeno v datasheetech a schématech od výrobce. Hledal jsem stejné součástky, popřípadě takové, které jim nejméně odpovídají. Do rohů desky jsem umístil otvory pro uchycení, aby se s deskou dalo snadněji manipulovat, případně ji uchytnit. Zobrazení bloků obvodů a jednotlivých součástek je přehledně zdokumentované na první stránce v datasheetu.

4.4 Realizace desky a výběr jednotlivých součástek

Pro návrh desky plošných spojů bylo potřeba vybrat vývojové prostředí. S tímto jsem neměl prozatím žádné zkušenosti. Proto jsem se poradil s vedoucím bakalářské práce. Následně jsem zhlédl několik tutoriálů na výrobu desek plošných spojů. Z nich nejlépe vycházelo vývojové prostředí EasyEDA⁵. Výhodou tohoto vývojového prostředí je jeho přehlednost a snadná práce s ním. Mnou vytvořená propojovací deska je jedna z jednodušších, a proto mi toto vývojové prostředí postačovalo. Pokud bych ale vytvářel desku složitější, s vícero návrhy, pravděpodobně bych zvolil jiné prostředí, například Altium Designer. Další velkou výhodou prostředí EasyEDA je možnost realizace návrhu desky. EasyEDA je totiž napojena na knihovnu součástek čínského výrobce JLCPCB. Ten nabízí možnost nechat si vyrobit samotnou desku nebo celou desku i se zvolenými součástkami. Bohužel vzhledem k současné situaci s nedostatečnými dodávkami součástek jsem se dostal do problému s nedostupností hned několika mnou vybraných součástek. Nebyl jsem schopen zaručit, že součástky nabízené v knihovně EasyEDA jsou opravdu skladem. Zároveň by hrozilo, že i kdyby se desku podařilo vytvořit a osadit součástkami, nemusela by přijít dostatečně brzo.

Požádal jsem proto mého vedoucího bakalářské práce, zda by mi s realizací propojovací desky pomohl. Nabídl mi možnost realizovat výrobu desky ve společnosti Gatema, s níž měl dobré zkušenosti. Následně jsem musel upravit výběr jednotlivých součástek, protože v lokální distribuční síti taktéž nebyly dostupné. Změna součástek si vyžádala změnu návrhu zapojení. Ta nejvíce ovlivnila výběr mikrokontroléru. Jelikož základní Atmega 328P nebyla dostupná, musel jsem vybrat z dostupných jiný podobný mikrokontrolér. Volba padla na Atmega 1284. Stejně jako předchozí typ je od výrobce Atmel, na architektuře AVR RISC. Jako jedna z mála byla v současné chvíli dostupná. Zároveň se prodává s označením typu PU tedy v DIP pouzdře. Pouzdro typu DIP značí, že mikrokontrolér je uzavřen v plastovém pouzdře s dvěma řadami pinů. Nepájí se tedy přímo na desku, ale pouze se upevní do vhodné velikosti patice, která je na desce připájena. Výhodou je, že pokud se mikrokontrolér poškodí, je možno jej jednoduše vyměnit. Další výhodou mikrokontroléru je, že má více pinů, tedy 40, což bylo výhodné, jelikož jsem se rozhodl podporovat vícero typů komunikace a na starším mikrokontroléru mi komunikační piny docházely. Je k ní dostupný, i když neoficiální firmware, o kterém píšu později v samostatné kapitole 4.5. Má 128 Kbytu flash paměti. Již při testování na Atmega 328P, která má 32 Kbytu paměti, jsem se při používání E-Ink displeje dostával přibližně k 60 % využití paměti. Proto jsem se rozhodl, že pokud je nutnost vzít jiný dostupný mikrokontrolér, bude lepší vybrat takový, který má větší integrovanou paměť. Zároveň by tak byla možnost rozšířit vývoj a použít pro vypisování na E-Ink displeji tuto extra paměť a tím vykreslování zrychlit.

⁵ <https://easyeda.com>

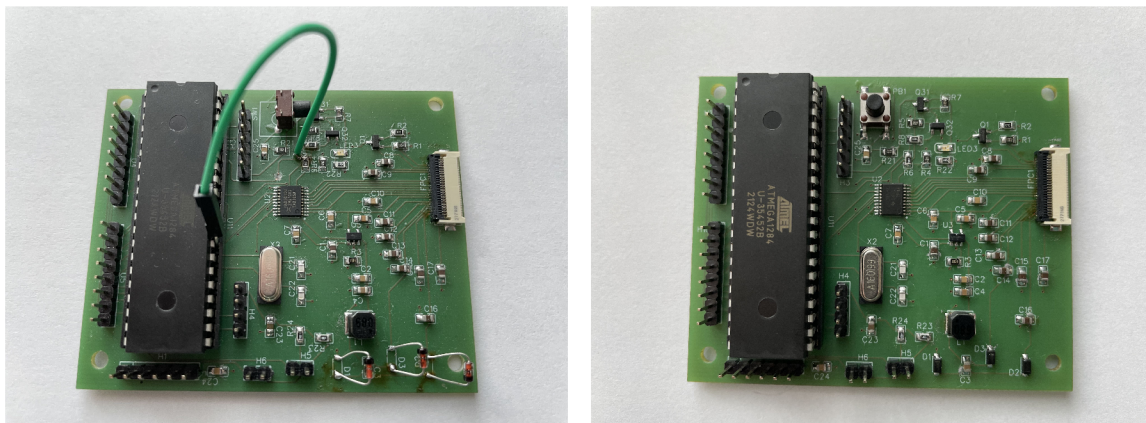
Bylo také potřeba změnit všechny tranzistory, FPC konektor pro připojení E-Ink displeje a převodníky napětí. Některé datasheety neobsahovali všechny potřebné informace, proto bylo hledání některých součástek vcelku komplikované. Objednávky součástek byly realizovány s pomocí vedoucího práce v obchodech Farnell a Mouser. Když byla deska plošných spojů vyrobena, osazení jednotlivých komponentů proběhlo ve školních laboratořích 4.2. Při implementaci součástek jsem objevil několik chyb. Nejprve špatné zapojení pro resetovací tlačítko. Tlačítka zakoupená mají připojení pro čtyři konektory, zatímco návrh desky počítal pouze se dvěma. Tato chyba se opravila mírnou modifikací zapojení tlačítka. Zjistil jsem, že mikroprocesory Atmel umožňují na výstupní piny namapovat pouze pull up rezistory. Bohužel na dvojici propojovacích pinů mám přivedeny také pull up rezistory. Proto nejsem na těchto pinech schopný docílit jiné hodnoty než logické 1. V tomto stavu jsem schopen rozlišovat komunikaci pomocí přivedení uzemnění na jeden z dvojice pinů. Pokud by bylo zapojení správné stačilo by piny pouze propojit. Z důvodu nedostupnosti byly na původní desce nahrazeny usměrňovací diody jinou verzí s menšími hodnotami proudu. I přesto deska po připojení k napětí nevykazovala žádné problémy. Poté jsem otestoval základní funkčnost desky. Všechny typy komunikace fungovaly. Byl jsem schopen připojit do FPC konektoru E-Ink displej a úspěšně na něm vypsát testovací program.

Bohužel při předělávání návrhu desky plošných spojů jsem v části s mikrokontrolérem špatně připojil RESET pin E-Inkového displeje. Chyba v návrhu vyústila v nutnost opravit první prototyp desky. Plošný spoj jsem přerušil a na původní místo napájel Dupont propojovací kabel, jehož druhý konec jsem připojil ke správnému pinu. Díky tomu jsem mohl na desce pokračovat ve vývoji. Rozhodl jsem se vytvořit druhý prototyp propojovací desky 4.2. Návrh plošného spoje na desce jsem opravil, zároveň s ním jsem změnil zapojení resetovacího tlačítka. Na připomínku od vedoucího bakalářské práce se upravili technické popisky na desce. Desku se podařilo včas vyrobít a znovu napájet všechny součástky. Bohužel po osazení mikrokontrolérem a základním testování jsem byl schopen pouze přijímat komunikační signály. Nebyl jsem schopen vypsát cokoliv na připojeném E-Inkovém displeji. Z toho vyplývá, že chyba bude v obvodech pro E-Ink displej. Na tuto verzi desky již byly osazeny originální usměrňovací diody, tedy problém by mohl být v nich. Druhou, pravděpodobnější možností, kde se chyba nachází je převodník napětí pro komunikační vodiče. Ten je na nové desce v jiné variantě, protože v první objednávce nepřišlo požadované množství. Chyba by šla odhalit ve školních laboratořích, za pomocí multimetru a osciloskopu. Na realizaci odhalení chyby jsem už neměl dostatek času.

Celý vývoj knihovny byl proveden na první variantě desky s hrubou úpravou. Oproti jiným deskám pro E-Ink displej je tato deska výhodná ve třech faktorech. Prvním je, že mikrokontrolér, který E-Ink displej ovládá se nachází přímo na desce a není tak nutnost desku znovu propojovat s jinou vývojovou deskou. Dále se na desce nacházejí dedikované komunikační piny pro paralelní, SPI i I²C komunikaci. Třetí čistě praktickou výhodou je, že všechny součástky se na desce nacházejí na jedné straně, tedy na druhou stranu by bylo možné přidělat E-Ink displej a usnadnit tak celkovou montáž do nějakého většího zařízení.

4.5 Postup nahrání bootloaderu a zdrojového kódu

Mikrokontroléry se běžně dodávají bez nainstalovaného obslužného firmwaru neboli bootloaderu proto, aby si na ně firmy nebo zákazníci mohli nahrát vlastní bootloader. Mikrokontrolér lze používat i bez něj, což má výhodu v možnosti využívat větší část integrované paměti. Bootloader slouží k snazšímu ovládní mikrokontroléru. Nabízí základní funkce pro práci s mikrokontrolérem tak, aby uživatel nemusel nutně přistupovat k jednotlivým regis-



Obrázek 4.2: Vlevo je první prototyp desky s úpravami, vpravo druhý prototyp desky, který správně nefunguje

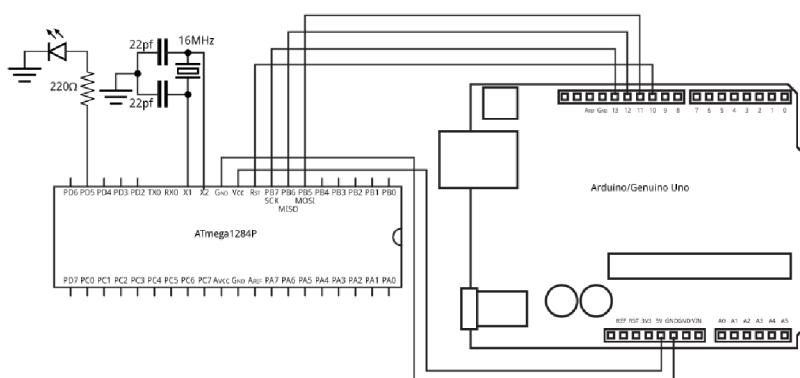
trům, časovačům a dalším modulům. Zároveň umožňuje snazší nahrávání zdrojového kódu. Jelikož jsem celý vývoj kódu knihovny dělal ve vývojovém prostředí Arduino a spoléhal se na základní funkce tohoto vývojového prostředí, jako je například `delay()`, `millis()`, `digitalWrite()` apod., musel jsem bootloader nahrát. Bootloader pro Atmegu 1284 však není společností Arduino podporován, jelikož se zatím nenachází na žádné jejich vývojové desce. Zároveň jsem neměl možnost použít pro nahrání bootloaderu USB/ISP programátor, jelikož Armege 1284 opět není programátorem podporována.

Proto jsem hledal jiné možnosti. Narazil jsem na návod určený přímo pro Atmegu 1284 [22]. V tomto návodu jsou konkrétně popsány způsoby, jak lze s mikrokontrolérem pracovat a jakým způsobem na něj lze nahrát bootloader. Součástí návodu je i odkaz na GitHub repositář, který obsahuje neoficiální bootloader [12]. GitHub repositář obsahuje návod, jak nainstalovat bootloader pro nepodporované desky do rozhraní Arduino IDE. Zároveň obsahuje i nastavení vstupních parametrů pro nahrání. Do Arduino IDE sekce File > Preferences je potřeba přidat odkaz na propojovací JSON strukturu. Ta umožní nahrát ovladače přímo z repositáře. Toto nahrání se provede v sekci Tools > Board > Boards Manager. V něm je potřeba najít desky s názvem MightyCore a nainstalovat je. Poté je na desku možné nahrávat. Jelikož jsem nemohl použít výše zmíněný USB/ISP programátor, bootloader jsem nahrál pomocí desky Arduino Nano v režimu ISP programátoru. Ten se na desku nahraje v sekci File > Examples > ArduinoISP. Pro samotné nahrání jsem připojil Atmegu 1284 do obvodového zapojení dle schématu na stránce. 4.3 Místo desky Arduino Uno jsem použil desku Arduino Nano, přičemž zapojení pro SPI komunikaci, tedy piny MISO, MOSI, SS a SCK, zůstaly totožné. Vývojové prostředí Arduino IDE v základu nabízí funkci nahrání bootloaderu na podporované desky. Přidáním podpory z Github repositáře jsem tuto funkci mohl využít. V sekci Tools stačilo nastavit připojenou desku Nano jako ISP a následně nahrání potvrdit stiskem Burn Bootloader. Bootloader se podařilo úspěšně nahrát, což potvrdil i výpis v terminálu vývojového prostředí. Mohl jsem začít nahrávat zdrojový kód.

Jelikož jsem na vývojovou desku neumístil USB port, nahrávání zdrojového kódu mohlo být provedeno dvěma způsoby. Nejprve jsem vyzkoušel metodu nahrávání pomocí desky Arduino Nano. Tu jsem připojil komunikačními piny RX pro příjem a TX pro odesílání. Propojil jsem piny pro napájení. Následně jsem propojil RESET pin a zemnicí pin desky Arduino Nano. Čímž jsem docílil toho, že USB/UART převodník FT232R na desce Nano

nebude posílat zdrojový kód na vlastní Atmegu 328P, ale pošle ho přes komunikační UART rozhraní na Atmegu 1284. Jelikož jsem RESET pin připojil na uzemnění, musel jsem mačkat RESET tlačítko na propojovací desce tak, aby byla Atmega 1284 schopná přijmout nový kód. Orientoval jsem se podle terminálu vývojového prostředí. Ve chvíli, kdy byl zdrojový kód přeložen a začal se nahrávat, jsem stisknul tlačítko. Testovací „Hello world“ kód se mi tímto způsobem podařilo nahrát.

Protože jsem pro další vývoj potřeboval používat desku Arduino Nano jako vysílač signálů, pro nahrávání zdrojových kódů jsem pořídil USB/UART převodník CP2102. Ten funguje na totožném principu zapojení. Nejprve piny 5V a GND pro napětí a uzemnění. Následně piny RX a TX, které se v tomto případě zapojují do kříže, tedy RX na TX a TX na RX. Opět ve chvíli, kdy začne nahrávání na desku je potřeba zmáčknout její tlačítko RESET. Tímto způsobem jsem nahrával zdrojové kódy, když jsem vyvíjel komunikační knihovnu.



Obrázek 4.3: Schéma zapojení pro nahrání bootloaderu (zdroj: [22])

Kapitola 5

Implementace obslužné knihovny

5.1 Obslužná knihovna

Obslužná knihovna je rozdělena do dvou logických částí. První část se stará o příjem všech typů komunikace a jejich převedení. Druhá část pak slouží pro výpis znaků na E-Ink displeji. K výpisu znaků na displeji je použita dodávaná obslužná knihovna pro E-Ink displeje [28]. K tomuto kroku jsem se rozhodl ze dvou důvodů. Za prvé, pokud bych knihovnu vytvářel vlastní, výsledný zdrojový kód by se mohl odlišovat převážně jinak použitými programovými konstrukcemi. Ovládání displeje na nižší úrovni musí být totožné jako dodávaná knihovna, jinak by displej nefungoval korektně. Proto mi přišlo zbytečné vytvářet znovu víceméně totožný kód. Navíc pokud bude knihovna pro E-Ink displej v budoucnu aktualizována o další funkce, či jinak upravena, nebude potřeba moji knihovnu upravovat, ale pouze napojit na novou verzi E-Ink knihovny. Druhým důvodem je modulárnost mojí knihovny pro příjem komunikace. E-Inkové displeje od Waveshare mají různé velikosti. Sám jsem testoval i 2,13" variantu. Knihovny různých velikostí se od sebe odlišují, ať už v názvech jednotlivých funkcí, nebo v práci s E-Inkovým displejem. Proto pokud bych chtěl podporovat další rozměry E-Ink displejů, je snazší pouze upravit moji část knihovny, která pracuje s vykreslovacími funkcemi a pozicováním na displeji. Také by bylo potřeba upravit napojení na knihovnu `SoftSPI.h`, kterou jsem použil. Popis je možné najít v kapitole 5.1.4. Popis jednotlivých zdrojových souborů je v souboru `README.txt`, který je u nich přiložen.

První verze knihovny byla vytvořena tak, aby četla hodnoty postupně z každého portu zvlášť. K tomu jsem se pokusil použít funkci `digitalread()`, ta má ale velkou časovou režii. Pro její použití by bylo potřeba upravit vstupní zdrojový kód s pomocí `delay()` funkce tak, aby mezi vstupními znaky a instrukcemi bylo větší zpoždění. To by nebylo v reálném použití vhodné, proto jsem zvolil hromadné čtení z vybraných registrů. Díky úpravě pro čtení v registrech jsem schopen přečíst jednou instrukcí všech osm bitů v jeden časový okamžik. První verzi knihovny jsem v kódu ponechal v případě, kdyby nebylo možné namapovat všechny datové vodiče na jeden druh registru. Druhá verze knihovny je vytvořena v souboru `LCD_reader.cpp` je logicky rozdělena do tří částí podle typu komunikace. Základní prvek knihovny je konstruktor `LCD_READER`. V něm jsou nejprve nastaveny hodnoty pinů na vstup nebo výstup. Následně jsou nastaveny piny s pull up rezistory a piny pro vytvoření přerušení. Typy komunikace se rozlišují podle počtu parametrů konstruktoru třídy `LCD_READER`.

5.1.1 Přijímání paralelní komunikace

Pokud jsou konstruktoru `LCD_READER` předány dva parametry, je komunikace paralelní. Pro paralelní komunikaci jsou nastaveny piny následovně. Hodnota PIN A je vstupní datový registr portů ze skupiny PA0 až PA7, na ně byly při návrhu napojeny datové piny pro paralelní komunikaci. Zde se ukazuje jedna z výhod mikrokontroléru Atmega 1284. Pro čtení šlo použít pouze jeden typ registru, jelikož je na něm až osm pinů. Oproti Atmega 328P, kde jich je pouze šest a bylo by potřeba použít více registrů. Hodnota RS pinu je napojena na PIN PD3 z registrů sekce D. Tato hodnota je používána pro rozlišení dat od instrukcí. Kdyby se přidávala ještě funkce, která umožňuje čtení z E-Ink displeje, bylo by potřeba přidat na některý pin z D sekce připojení R/W. Hodnota R/W značí, zda se čte anebo zapisuje. Hodnota pinu Enable je připojena na PIN PD2, na který je nastaven pull up rezistor. Generuje se na něm přerušení, které volá funkci `recieveEvent_paraler()`. Ta začíná číst příchozí data.

Pokud je na pinu RS logická hodnota 1, přišla data, která je potřeba zapsat do bufferu a inkrementovat index. Pokud je logická 0, přichází instrukce. Na zpracování příchozích instrukcí se používá funkce `parserinstructions()`, které jsou předány hodnoty přicházejících osmi bitů. U této funkce je přidán atribut `always_inline` [16]. Tento atribut omezuje režii spouštěním funkce tak, aby doba přerušení, kdy se čtou instrukce, byla co nejkratší. Tělo funkce je tvořeno jednoduchou `if()` konstrukcí, kde se porovnávají příchozí bitové otisky a podle nich se nastavují signalizační pravdivostní proměnné, případně se upravuje index pozice znaků na displeji. Pokud bude požadavek na přidání zatím nepodporované instrukce, stačí ji přidat pouze do této funkce a bude možné ji využívat ve všech typech komunikace. Pokud přichází data, odpovídají formátu znaků v datové struktuře `char`. Data se ukládají do znakového pole `displaybuff []`. Pozici pro ukládání určuje proměnná `index`.

5.1.2 Přijímání I²C komunikace

Pokud je konstruktoru zadán jeden parametr, je to hodnota adresy I²C. Následně se začne pomocí knihovny `Wire.h` poslouchat komunikace na zadané adrese. Pokud přijdou na datovém vodiči nová data, je vyvolána funkce `receiveEvent()` v podobě interrupt přerušení. Tato funkce má jeden parametr pro počet bytů. Vždy přichází rámeček jednoho byte. D3 až D0 jsou čtyři spodní bity v podobě Enable, RS, R/W, V0. Vrchní bity D7 až D4 obsahují data v podobě instrukcí nebo textu. Pokud by nastalo přerušení nebo chyba v přenosu a nepřenесли by se jednotlivé celky, ale pouze části dat, je výhodné mít oddělené signalizace pro přenos instrukcí a dat. Signalizace je tedy vytvořena pro každou část zvlášť a při chybě je možné data zpětně sesynchronizovat. Funkce `recieveEvent()` v případě přicházejících dat zavolá funkci `parsertext()`. Ta přijme první čtyři bity, které uloží do mezipaměti. Poznámá, že jsou uloženy. Při druhém průchodu jsou již příšlé bity spojeny s novými v pořadí D7 až D0. Podle nastavené hodnoty indexu jsou výsledné znaky postupně vypsané do globálního pole. Instrukce v komunikaci I²C jsou zpracovány totožně jako při paralelní komunikaci.

5.1.3 Přijímání SPI komunikace

Pokud jsou konstruktoru zadány čtyři parametry MISO, MOSI, SCLK, SS, převádí se komunikace SPI. Porty jsou nastaveny na vstup a výstup následovně. MISO výstup, MOSI vstup, SS vstup, SCLK vstup. Dále se nastaví kontrol registr `SPCR`, aby poslouchal příchozí data a vyvolal přerušovací funkci `ISR`. Ta značí, že přenos dat byl ukončen a je možné z

registru `SPCR` data číst. Na začátku funkce je vymaskován registr `SPCR`, protože jeho velikost je větší než potřebných osm bitů. Následně obdobně jako u `I2C` komunikace jsou vymaskovány datové hodnoty na pozicích pinů `D6(1)` až `D3(1)`. Další hodnoty jsou pouze signalizační. Příchod prvních datových hodnot je poznačen. V dalším rámci komunikace přichází druhá polovina datových hodnot opět na pozici `D6(2)` až `D3(2)`. Jsou spojeny dohromady nejprve první polovina `D6(1)` až `D3(1)`, k ní je přidána `D6(2)` až `D3(2)`. Na rozdíl od `I2C` komunikace jsou bity v opačném pořadí a po jejich propojení je potřeba je porovnat bitovým posunem. Algoritmus pro třídění je variací na algoritmus Merge sort. Posloupnost všech bitů se rozdělí na dvě poloviny, které promění svoji pozici. V dalším kroku se tyto dvě poloviny znovu rozdělí a opět prohodí. Prohazování končí při rozdělení na samostatné bity. V tu chvíli jsou seřazeny od nejnižšího po nejvyšší, případně opačně v závislosti na vstupu [4]. Časová složitost je \sqrt{n} a prostorová složitost je $2n$. Potom je možné přidat data do bufferu znaků a začít z něj tisknout. V případě instrukcí jsou hodnoty opět převedeny přes již popsanou `if()` konstrukci.

5.1.4 Výpis na E-Inkový displej

Spustitelným souborem je `epd2in.ino`. Pokud je soubor nahrán, nejprve se inicializuje E-Ink displej v podobě vyčištění předchozího obsahu. Následně se vypíše jednoduchý počáteční výpis, aby bylo uživateli oznámeno, že je deska připravena přijímat příkazy. V souboru se také nachází konstrukce pro rozlišení příchozí komunikace. Podle logických hodnot připojených na rozlišovacích pinech na desce (`H5`, `H6`) se vybere hodnota, která bude předána jako parametr pro rozlišování komunikace `LCD_READER`. V cyklu je volána funkce `auto_print()`. Ta, pokud bylo dosaženo čekání alespoň tří vteřin, začne vypisovat případná nově příchozí data. Tato funkce volá původní funkce pro vykreslování z originální knihovny pro E-Ink displej. Zároveň pozicuje jednotlivé znaky do řádků a sloupců. Vykreslování na E-Ink displeji se drží hodnot vytvořených tak, aby napodobovaly LCD displeje. Je tak možné zobrazit 16 sloupců a 2 řádky. Celkově mimo viditelnou zónu je možnost zapsat do bufferu až 80 symbolů, tedy 40 na každý řádek.

Kvůli pomalé obnovovací frekvenci E-Ink technologie je potřeba větší časová prodleva mezi výpisy. `Delay()` funkce se nepoužívá, protože k přidanému času funkce by se připočítala i délka přerušení při zpracování příchozích dat. Výsledkem by byla nepravidelná aktualizace obrazovky. Nejprve se uloží hodinový čas mikrokontroléru a v dalším kroku se počítá časový rozdíl. Pokud je větší než požadovaný interval, tak se na displej vypíše. Výpis na displeji je omezen pravdivostní hodnotou, ta se mění pouze v případě, že přijdou nová data pro výpis. Při výpisu je nejdříve nahrán podkladní rámeček, následně se v cyklech vypisují dva řádky znaků. Do paměti mikrokontroléru je nahrán obrázek znaku v podobě hexadecimálního pole. Z paměti se pomocí funkce `SetFrameMemory()` vloží na příslušnou pozici. Poté, když se všechna písmena nahrají, zobrazí se na displeji najednou. První pozice na displeji je pevně stanovena hodnotami `startX` a `startY`. Další pozice se počítají podle šířky znaku a šířky mezery mezi nimi postupným vypisováním podle počtu příchozích znaků. Znaky jsou uloženy v pomocném poli `displaybuff()`. Hodnota dalšího řádku je opět pevně stanovena o posun na vertikální poloze.

Při vývoji jsem také použil knihovnu `SoftSPI.h` [21]. Ta se využívá, protože mikrokontrolér má pouze jednu skupinu pinů pro SPI komunikaci. Tato skupina už je použita pro přijímání signálů z LCD displeje. Protože E-Ink displej používá SPI komunikaci také, bylo potřeba přemapovat jiné piny tak, aby mohly odesílat na E-Ink displej. Na LCD displej jsou použity původní SPI piny mikrokontroléru, protože v případě příchozího přerušení jsou

data už nahrána v registrech. Pokud by se `SoftSPI.h` knihovna zvolila pro příjem, trvalo by přerušeni podstatně delší dobu, musely by se přečíst všechny hodnoty bitů. Knihovna byla přidána do souboru `epd.cpp` a `epd.h`. Nejprve se nainicializují porty, které nebyly původně určeny pro SPI komunikaci. Následně se vymění třídy `SoftSPI` a `SPI`. Další průběh je totožný s původní knihovnou.

5.1.5 Podpora funkcí displeje

LCD displeje mají hned několik funkcí pro práci se znaky [3]. Z jejich nabídky jsem vybral základní set z mého pohledu nejdůležitějších instrukcí.

- `clear()` - odstraní obsah na displeji
- `home()` - nastaví kurzor na levý horní roh displeje
- `write()` - vypíše znak na displej
- `print()` - vypíše řetězec na LCD displej
- `scrollDisplayLeft()` - posouvá text displeje o jedno pole doleva
- `scrollDisplayRight()` - posouvá text displeje o jedno pole doprava
- `autoscroll()` - zapíná funkci automatického posunu znaků na displeji
- `noAutoscroll()` - vypíná funkci automatického posunu znaků na displeji
- `cursor()` - zobrazuje na aktivní pozici symbol kurzoru
- `noCursor()` - vypíná zobrazování kurzoru

5.2 Vytvoření nových fontů

Pro co nejméně emulaci znakového LCD displeje jsem se rozhodl vytvořit na E-Ink displej vlastní fonty. Tyto fonty vzhledově kopírují jednotlivé rozložení pixelů na obou vybraných displejích, tedy 8 pixelů na výšku a 5 pixelů na šířku. Jejich reálné rozměry jsem vypočítal na 31x14 pixelů. Pokud bych chtěl znaky větší při dodržení obdélníkového formátu, nemohlo by se jich na displej vypsát 16x2. Při této hodnotě se mi jejich velikost zdála nejvíce věrohodná. Jejich vzhled jsem podřídil vzhledu znaků v datasheetech přiložených k displejům. Fonty písma se rozlišují u dvou použitých řadičů. Proto jsem vytvořil také dva fonty tak, abych podporoval oba řadiče. Rozsah znaků, který můj font podporuje, je v decimálních hodnotách ASCII od 33 do 127. Displeje podporují vícero znaků, tedy i od hodnot 160 do 255. Tyto znaky se však z velké části skládají ze symbolů čínské abecedy, proto jsem se rozhodl, že je z důvodu nevyužitelnosti nezahrnu do návrhu.

Jednotlivé znaky jsem vytvořil v grafickém programu Gimp. Z jednoduchého návrhu skládajícího se z 8x5 černých pixelů jsem postupnou úpravou vytvořil všechny znaky. Ty jsem exportoval ve formátu `.png`. Když jsem měl grafický návrh všech znaků hotový, upravil jsem ho do formátu, který vyžaduje knihovna E-Ink displeje. Formát je pole v jazyku C/C++. Prvky pole jsou souřadnice pixelů v hexadecimální podobě. Využil jsem k tomu online dostupný konvertor [7]. Tento konvertor změní bitmapové soubory typu `.png` na pole hexadecimálních hodnot. Pro výpis je použita knihovnická funkce pro práci s E-Ink displejem,

konkrétně funkce `DrawCharAt`. Výpis jednotlivých znaků se řídí pomocí ASCII hodnot. Tedy například, pokud chci vypsat znak „A“, tak vím, že jeho hexadecimální hodnota je 65. Úplně první znak v sadě fontu je „ “, tedy „whitespace“, s hodnotou 32. Offset pro výpočet pozice prvního pixelu má hodnotu $65 - 32 = 33$. Každý znak má rozměry 31x14 pixelů, přičemž hodnota šířky se dopočítává na velikost dělitelnou osmi. Hodnota musí být osm bitů, protože výsledné obrázky jsou uloženy jako pole charů. Tedy 31x16, což je 496 pixelů. Hodnoty jsou v poli uloženy v hexadecimálním formátu, tedy každý znak má $496/8 = 62$ hodnot. Z toho můžeme snadno zjistit pozici první hodnoty určitého znaku. Pro příklad znak A, offset je stanoven na 33, počet znaků pro písmeno je 62, $33 \times 62 = 2046$. To znamená, že na pozici 2046 je první hodnota pro znak A [27].

Do souboru s fonty byla přidána ještě dvojice speciálních znaků, kterými jsou kurzor a plný znak. Kurzor se zobrazuje při použití funkce `cursor()`, která je více popsána v kapitole 5.1.5. Plný znak se zobrazuje při úvodní obrazovce E-Ink displeje tak, aby displej vizuálně odpovídal inicializaci na LCD displeji. Zároveň jsem přidal bitmapový soubor, který vykresluje na E-Ink displej černý rámeček, ten vizuálně napodobuje vzhled LCD displeje.

5.3 Instalace

Z přiložené SD karty je potřeba stáhnout zdrojové soubory do nového adresáře. Pro nahrání zdrojových kódů na propojovací desku doporučuji vývojové prostředí Arduino IDE. Do něj je potřeba přidat podporu pro neoficiální vývojové desky, která obsahuje mimo jiné i specifikace pro mikrokontrolér Atmega 1284. Vše je popsáno v kapitole 4.5. Pro nahrání na desku je potřeba připojit USB/UART převodník, například CP2102. Přesný postup nahrání kódu je uveden v sekci 4.5.

Po nahrání kódu na desku se na displeji zobrazí úvodní obrazovka. Pokud byla deska odpojena od zdroje napětí a následně připojena je potřeba desku restartovat, aby zobrazila úvodní obrazovku. Pro správné připojení komunikačních pinů desky je potřeba zobrazit si tyto informace v přiloženém PCB návrhu B.1. Pro značení příchozí komunikace je potřeba připojit na konektory H5 a H6 uzemnění. Když je na H6 připojeno uzemnění a na H5 nic nepřipojím komunikace bude typu SPI. V opačném případě je komunikace paralelní. Pokud jsou oba konektory ve stejné logické hodnotě je komunikace typu I²C.

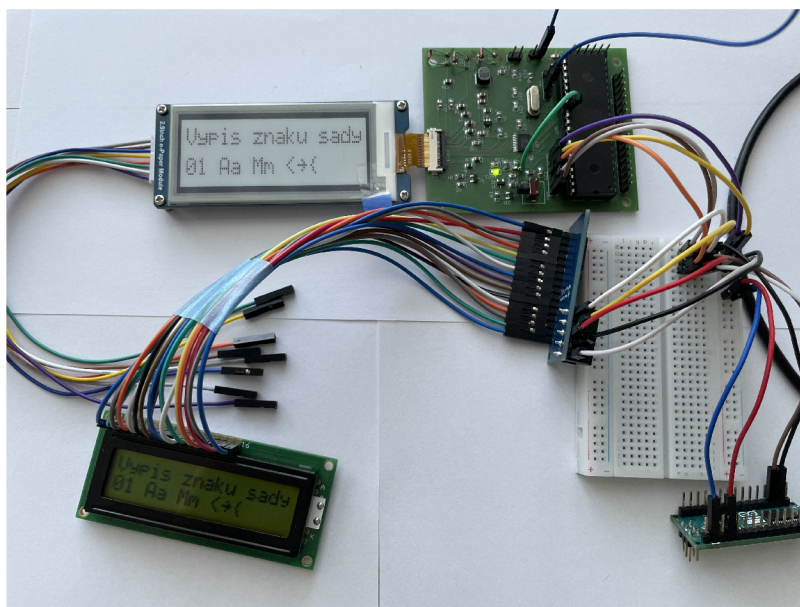
5.4 Testování

Testování funkčnosti bylo provedeno v několika iteracích během průběžného vývoje propojovací desky. Nejprve jsem otestoval, zda je deska schopna rozeznat odlišné typy komunikace pomocí signalizační dvojice pinů (H5, H6). Testování proběhlo úspěšně, když deska byla schopna rozeznat všechny typy komunikace. Následně jsem začal testovat jednoduché výpisy a zobrazení znaků. Otestoval jsem základní písmena a zároveň znaky, které se ve fontech odlišují. Poté jsem na všech typech komunikací začal zkoušet funkce displeje ze seznamu uvedeného v kapitole 5.1.5. Při tomto testování jsem vždy nahrál totožný zdrojový kód na LCD displej a na propojovací desku s připojeným E-Ink displejem. Následně jsem porovnal, zda se oba displeje chovají stejně. Při použití základních testovacích kódů jsem neodhalil žádný problém.

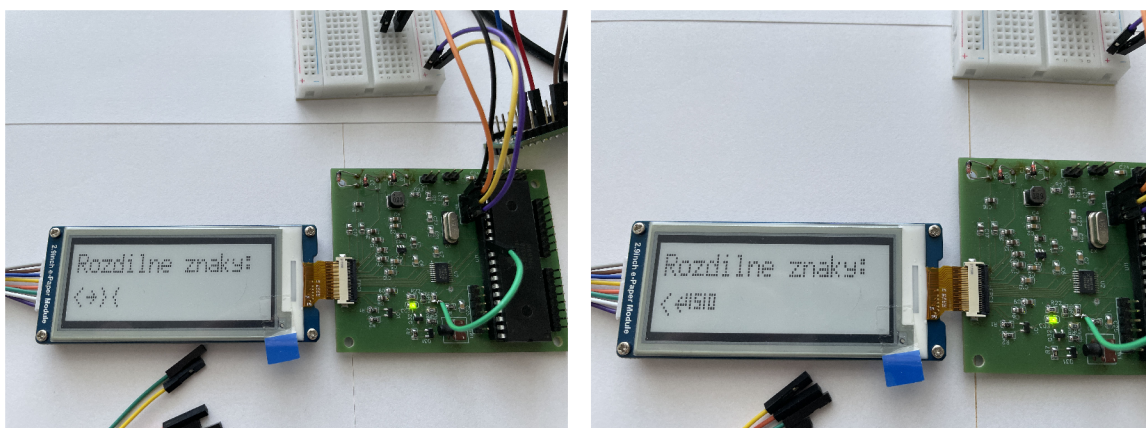
Projevily se limity technologie E-Ink displejů, tedy zejména jejich pomalá obnovovací frekvence. Pokud se vypisovaný text často obnovuje, je potřeba počítat se zpožděním.

Zpoždění lze také zaznamenat při použití funkcí jako je `autoscroll()` nebo `scrollDisplayRight/Left()`. V tomto případě je potřeba do původních zdrojových souborů přidat funkci `delay()` zpoždění alespoň tři vteřin, aby se vyrovnal hendikep. Pokud se tak učiní, je zobrazování na displejích totožné.

Obrázek výpisu znaků při testování je přiložen níže 5.1, zároveň je tu ukázka odlišnosti fontů při tisknutí stejných ASCII znaků 5.2. Zároveň se na odevzdané SD kartě nachází krátké video. Na videu I²C je zdrojová deska Arduino Nano připojena do nepájivého pole. Do něj je zároveň napojen LCD displej s převodní deskou a E-Ink displej s propojovací deskou. Po stisknutí RESET tlačítka na desce Arduino lze vidět výpis na obou displejích. Na videu SPI je stejné zapojení, pouze se vypisuje jiný testovací kód.



Obrázek 5.1: Testovací výpis znaků na SPI komunikaci



Obrázek 5.2: Vlevo je font displeje DEM16216SYH, vpravo pro displej RC1602B

5.5 Vhodné další rozšíření

Prvním rozšířením, které bych se rozhodl udělat, je podpora pro více velikostí obou typů displejů. Tedy pro LCD displeje formátu například osm sloupců a dva řádky by stačil i menší E-Ink displej o velikosti například 2,13". Kód mojí části knihovny je pro to přizpůsoben, jen je potřeba udělat výpočty pro správné pozicování a zobrazení znaků. Následně je nutné napojit originální knihovnu pro E-Ink displej na moji část. Dalším rozšířením by byla možnost využít větší velikost paměti osazeného mikrokontroléru a pokusit se vykreslovat grafiku na E-Ink displeji pomocí něj, místo zabudovaného řadiče v displeji. Řízení komunikace a vykreslování by mohlo být efektivnější.

Dalším možným rozšířením by bylo upravení samotné propojovací desky. Deska je první prototyp, a proto pokud bych ji vytvářel znovu, snažil bych se ji udělat s menšími rozměry tak, aby ji bylo jednodušší implementovat do reálného zařízení. V ideálním případě by měly rozměry desky odpovídat rozměrům E-Ink displeje. Také bych lépe uspořádal jednotlivé komunikační piny tak, aby se snáze připojovali. Když jsem připojoval na desku konektory pro komunikace, uvědomil jsem si, že pokaždé musím nahlížet jak do PCB návrhu, tak do schématu zapojení. Proto jsem ke konektorům v návrhu PCB dodělal technické popisky, tak aby je šlo snadno propojit. Desku s těmito popisky jsem bohužel nestihl realizovat, ale pro další používání bych ji určitě ocenil. Zároveň bych správně propojil signalizační konektory na pull down rezistory.

Kapitola 6

Závěr a zhodnocení

Cílem této bakalářské práce bylo emulovat vzhled a funkcionalitu LCD displeje za pomoci E-Ink displeje. Tento záměr byl splněn. Současný prototyp desky, tedy ten s modifikacemi, je schopen přijímat všechny typy komunikace. Převést text i základní instrukce, které se nejčastěji používají při ovládání LCD displeje a vykreslit je na E-Ink displeji. Myslím si, že při dalším vývoji, který jsem zmínil v rozšířeních, by byla deska schopná reálného nasazení. Vhodné by bylo zaměřit se na jeden konkrétní typ zařízení, například zabezpečovací systémy, které jsou u nás na fakultě.

Osobní přínos vidím v tom, že jsem se blíže seznámil s vytvářením desek plošných spojů. Této oblasti se bakalářské studium na Fakultě informačních technologií věnuje pouze z části a převážně teoreticky. Bylo zajímavé si vyzkoušet realizaci vlastního návrhu schématu zapojení a vytvoření návrhu PCB. Následně i přes komplikace, vybrat potřebné součástky a osadit je na desku. Tu oživit připojením mikrokontroléru. Zvládnul jsem také desku lehce upravit, když se objevily chyby v návrhu tak, aby byla funkční.

Při vývoji knihovny jsem si znovu připomněl probírané učivo různých typů komunikací a měl jsem možnost již naučenou teorii převést do praxe. Zároveň jsem se dozvěděl nové informace o E-Ink displejích. Jakým způsobem komunikují s řídicím mikroprocesorem, jakým způsobem na nich vypisovat informace a jaké jsou jejich výhody a omezení.

Literatura

- [1] AG, D. T. *What actually means “TN“* [online]. 2013. [Online; navštíveno 10.02.2022]. Dostupné z: <https://www.dmbtechnics.com/en/what-does-tn-mean-tn-technology/>.
- [2] ARDUINO. *LCD display projects* [online]. Arduino, 2015. [Online; navštíveno 29.10.2021]. Dostupné z: <https://search.arduino.cc/search?q=lcd%20display&tab=projecthub>.
- [3] ARDUINO. *LiquidCrystal* [online]. 2015. [Online; navštíveno 10.12.2021]. Dostupné z: <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>.
- [4] B., A. *Most Efficient Algorithm* [online]. Wyzant, 2019. [Online; navštíveno 17.03.2022]. Dostupné z: <https://www.wyzant.com/resources/answers/661501/most-efficient-algorithm-for-bit-reversal-from-msb-lsb-to-lsb-msb-in-c>.
- [5] BIDLO, M. *Principy sériové komunikace, sériová komunikační rozhraní* [online]. 2020. [Online; navštíveno 30.09.2021]. Dostupné z: https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Flectures%2F04-IMP-seriova_kom.pdf&cid=13324.
- [6] BUČINA, T. *Jak pracují tekuté krystaly?* [online]. Scienceworld, 2002. [Online; navštíveno 29.04.2022]. Dostupné z: <https://www.scienceworld.cz/technologie/jak-pracuji-tekute-krystaly-3737/>.
- [7] DIGOLE. *Convert picture to C code array* [online]. 2014. [Online; navštíveno 03.03.2022]. Dostupné z: https://www.digole.com/tools/PicturetoC_Hex_converter.php.
- [8] DISPLAYS, P. *EPD Extension Kit Gen 2 (EXT2)* [online]. Pervasive Displays. [Online; navštíveno 24.10.2021]. Dostupné z: <https://www.pervasivedisplays.com/product/epd-extension-kit-gen-2-ext2/>.
- [9] EMBEDDED, P. *How to use an HD44780 based Liquid Crystal Display* [online]. Rocco Marco Guglielmi, 2015. [Online; navštíveno 03.12.2021]. Dostupné z: <https://www.playembedded.org/blog/hd44780-lcdii-and-chibioshal/>.
- [10] HACKSTER.IO. *ATmega328P Standalone Board* [online]. Ahmed Ibrahim Ahmed, 2019. [Online; navštíveno 03.03.2022]. Dostupné z: <https://www.hackster.io/ahmedibrrahim/atmega328p-standalone-board-77044d>.
- [11] HACKSTER.IO. *How to make Arduino Uno clone board* [online]. sagar saini, 2021. [Online; navštíveno 03.03.2022]. Dostupné z: <https://www.hackster.io/sainisagar7294/how-to-make-arduino-uno-clone-board-066b81>.
- [12] HANS. *MightyCore* [online]. 2015. [Online; navštíveno 20.03.2022]. Dostupné z: <https://github.com/MCUdude/MightyCore>.

- [13] HITACHI. *HD44780* [online]. [Online; navštíveno 30.09.2021]. Dostupné z: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>.
- [14] INC., E. I. H. *Two Pigment Ink System* [online]. 2016. [Online; navštíveno 10.10.2021]. Dostupné z: <https://www.eink.com/electronic-ink.html>.
- [15] INSTRUMENTS, T. *PCF8574 Expander* [online]. 2015. [Online; navštíveno 04.11.2021]. Dostupné z: <https://www.ti.com/lit/ds/symlink/pcf8574.pdf>.
- [16] JOURNAL, C. *Arduino Inline Keyword* [online]. Indrek Luuk, 2019. [Online; navštíveno 05.03.2022]. Dostupné z: <https://circuitjournal.com/arduino-force-inline>.
- [17] KONVALINA, J. *LCD a jejich technologie 1 - jak to funguje* [online]. NOTEBOOK.cz, 2013. [Online; navštíveno 29.04.2022]. Dostupné z: <https://notebook.cz/clanky/technologie/2013/LCD-1>.
- [18] MALÝ, M. *Hradla, volty, jednočipy*. 2. vyd. CZ.NIC, 2020. ISBN 978-80-88168-23-2.
- [19] MICROCHIP. *MCP23008 Expander* [online]. 2014. [Online; navštíveno 24.11.2021]. Dostupné z: <https://www.farnell.com/datasheets/3045885.pdf>.
- [20] SITRONIX. *ST7066U* [online]. [Online; navštíveno 30.09.2021]. Dostupné z: https://newhavendisplay.com/content/app_notes/ST7066U.pdf.
- [21] STUDIO, S. *Arduino Software SPI User Guide* [online]. Seed studio, 2016. [Online; navštíveno 29.03.2022]. Dostupné z: <https://wiki.seeedstudio.com/Software-SPI/>.
- [22] TECHNOBLOGY. *Using the ATmega1284 with the Arduino IDE* [online]. 2016. [Online; navštíveno 20.03.2022]. Dostupné z: <http://www.technoblogy.com/show?190V>.
- [23] VÁCLAVÍK, L. *Albert dává papíru sbohem. V prodejnách testuje elektronické cenovky* [online]. 2019. [Online; navštíveno 20.10.2021]. Dostupné z: <https://www.cnews.cz/albert-elektronicke-cenovky>.
- [24] WAVESHARE. *2.9inch e-Paper Datasheet* [online]. Waveshare. [Online; navštíveno 24.10.2021]. Dostupné z: https://www.waveshare.com/wiki/File:2.9inch_e-Paper_Datasheet.pdf.
- [25] WAVESHARE. *2.9inch e-Paper Module* [online]. Waveshare. [Online; navštíveno 24.10.2021]. Dostupné z: https://www.waveshare.com/wiki/2.9inch_e-Paper_Module.
- [26] WAVESHARE. *2.9inch e-Paper Schematic* [online]. Waveshare. [Online; navštíveno 24.10.2021]. Dostupné z: https://www.waveshare.com/w/upload/8/85/2.9inch_e-Paper_Schematic.pdf.
- [27] WAVESHARE. *Make a new larger font for Waveshare SPI e-Paper* [online]. 2018. [Online; navštíveno 10.03.2022]. Dostupné z: <https://wavesharejfs.blogspot.com/2018/08/make-new-larger-font-for-waveshare-spi.html>.
- [28] WAVESHARE. *E-Paper* [online]. Waveshare, 2019. [Online; navštíveno 03.03.2022]. Dostupné z: <https://github.com/waveshare/e-Paper>.
- [29] WIKIPEDIA. *E Ink* [online]. Wikipedia. [Online; navštíveno 29.03.2022]. Dostupné z: https://en.wikipedia.org/wiki/E_Ink.

Příloha A

Obsah přiložené SD karty

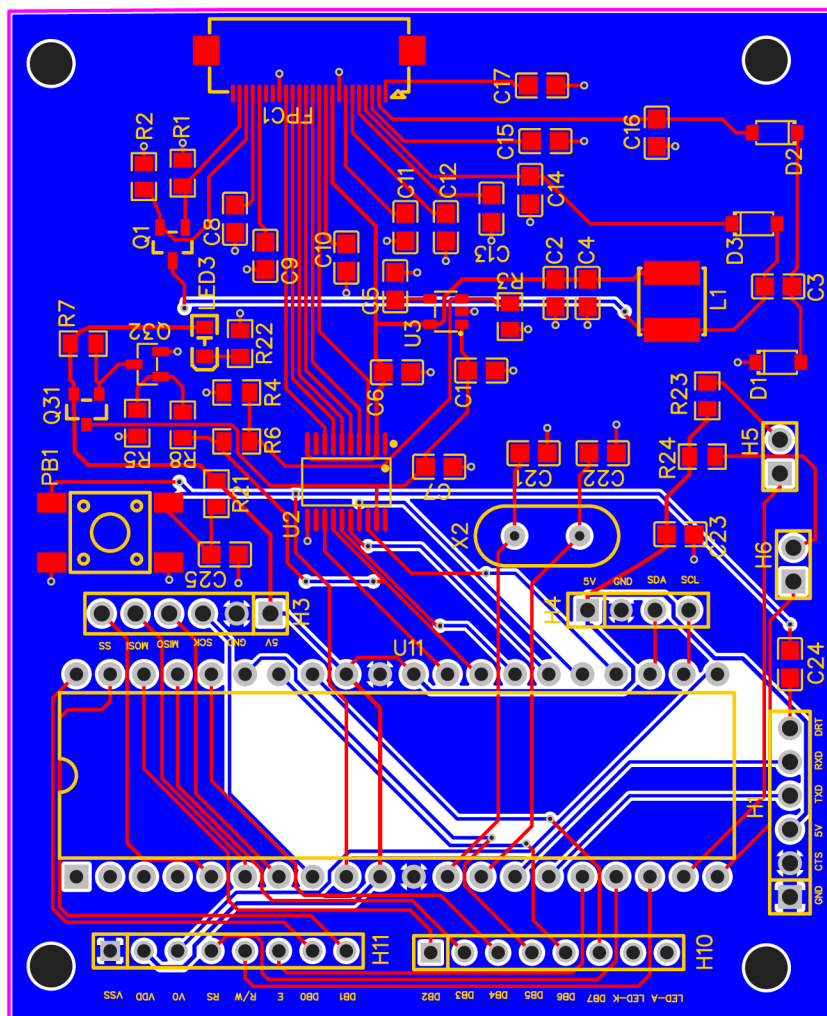

```

└─ xrusin04_lib
  └─ epd2in9.....Zdrojové kódy pro překladovou desku
    ├── LCD_eink.cpp
    ├── LCD_eink.h
    ├── LCD_reader.cpp
    ├── LCD_reader.h
    ├── README.txt
    ├── SoftSPI.cpp
    ├── SoftSPI.h
    ├── epd2in9.cpp
    ├── epd2in9.h
    ├── epd2in9.ino
    ├── epdif.cpp
    ├── epdif.h
    ├── epdpaint.cpp
    ├── epdpaint.h
    ├── fontHD4.cpp
    ├── fontST7.cpp
    └── fonts.h
  └─ navrh_desky .....Schémata překladové desky
    ├── PCB.json
    ├── PCB.pdf
    ├── SHEET.json
    └── Sheet.pdf
  └─ testovani .....Testovací zdrojové kódy
    ├── I2C
    │   ├── sender
    │   │   ├── LiquidCrystal_I2C.cpp
    │   │   ├── LiquidCrystal_I2C.h
    │   │   └── sender.ino
    │   ├── sender2
    │   │   ├── LiquidCrystal_I2C.cpp
    │   │   ├── LiquidCrystal_I2C.h
    │   │   └── sender2.ino
    │   └── sender3
    │       ├── LiquidCrystal_I2C.cpp
    │       ├── LiquidCrystal_I2C.h
    │       └── sender3.ino
    ├── PARA
    │   ├── sender
    │   │   ├── LiquidCrystal.cpp
    │   │   ├── LiquidCrystal.h
    │   │   └── sender.ino
    │   ├── sender2
    │   │   ├── LiquidCrystal.cpp
    │   │   ├── LiquidCrystal.h
    │   │   └── sender2.ino
    │   └── sender3
    │       ├── LiquidCrystal.cpp
    │       ├── LiquidCrystal.h
    │       └── sender3.ino
    ├── SPI
    │   ├── sender
    │   │   ├── Adafruit_LiquidCrystal.cpp
    │   │   ├── Adafruit_LiquidCrystal.h
    │   │   ├── sender.ino
    │   │   └── utility
    │   │       ├── Adafruit_MCP23008.cpp
    │   │       └── Adafruit_MCP23008.h
    │   ├── sender2
    │   │   ├── Adafruit_LiquidCrystal.cpp
    │   │   ├── Adafruit_LiquidCrystal.h
    │   │   ├── sender2.ino
    │   │   └── utility
    │   │       ├── Adafruit_MCP23008.cpp
    │   │       └── Adafruit_MCP23008.h
    │   └── sender3
    │       ├── Adafruit_LiquidCrystal.cpp
    │       ├── Adafruit_LiquidCrystal.h
    │       ├── sender3.ino
    │       └── utility
    │           ├── Adafruit_MCP23008.cpp
    │           └── Adafruit_MCP23008.h
    └─ video.....Ukázka fungování překladové desky
        ├── I2Ctest.mp4
        └── SPITest.mp4
  └─ xrusin04.pdf .....Zdrojové kódy práce v LaTeXu
  └─ zdroj.zip .....Text práce

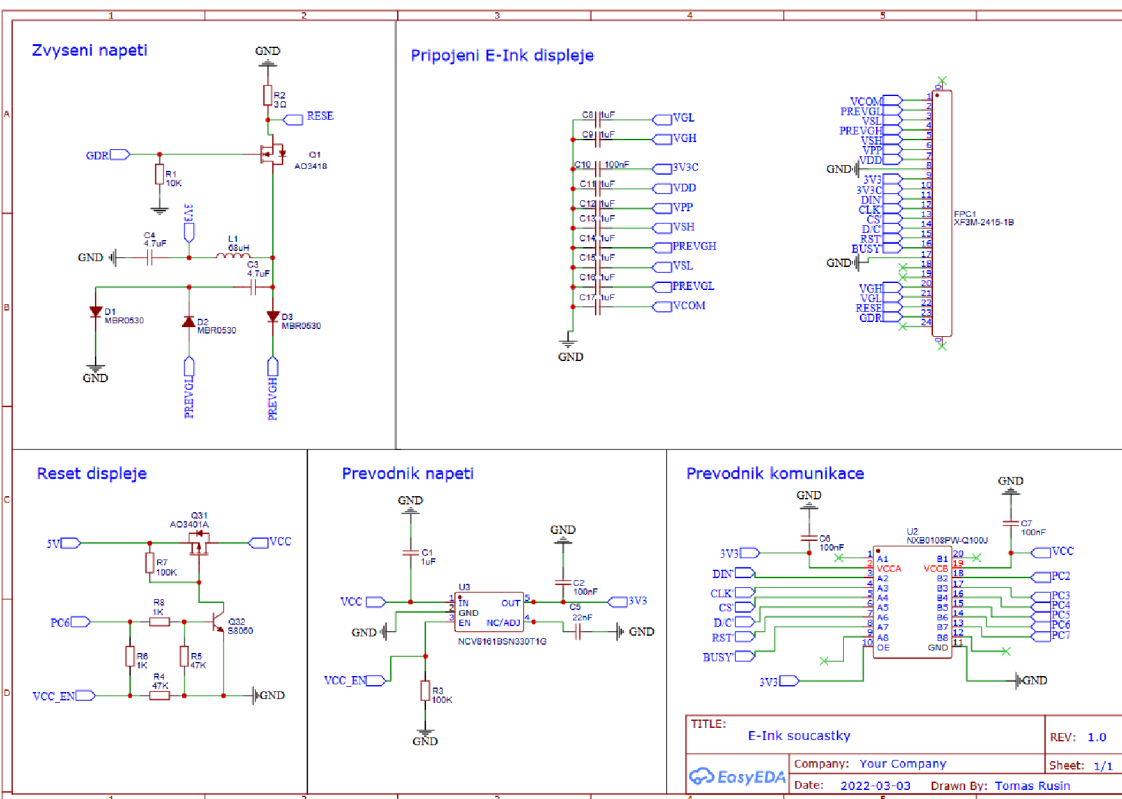
```

Příloha B

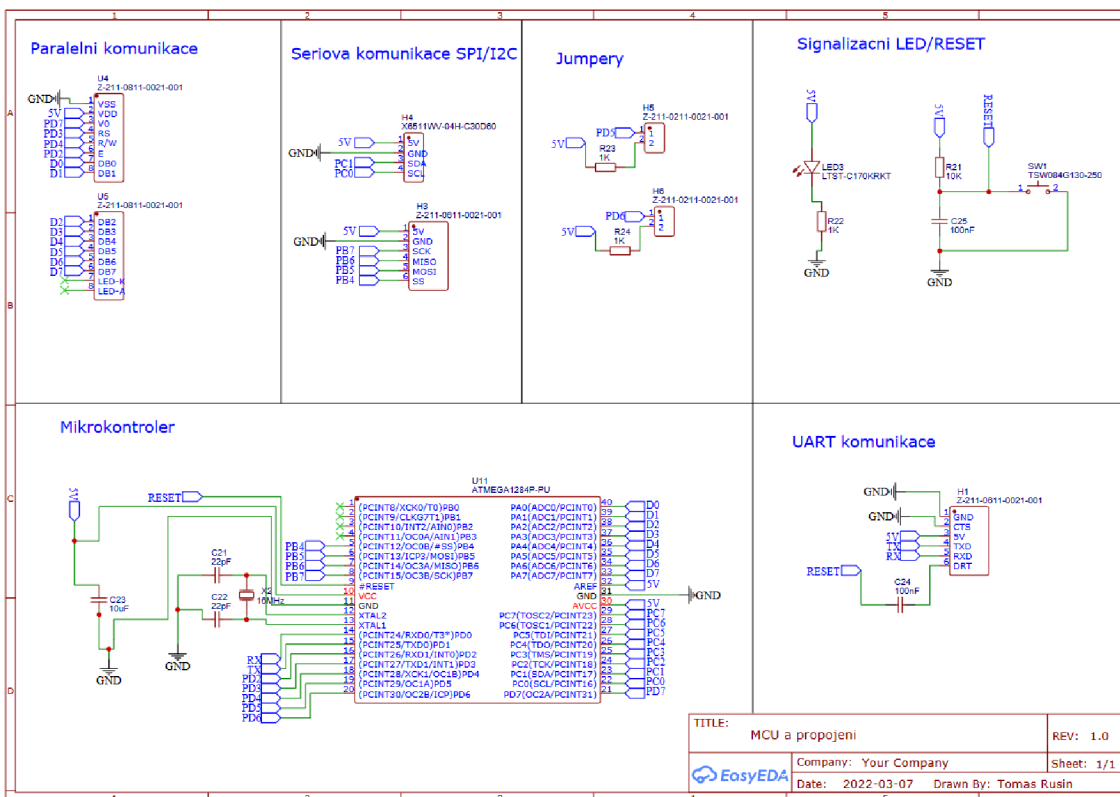
Schémata desky



Obrázek B.1: PCB návrh propojovací desky



Obrázek B.2: Schéma propojovací desky první strana



Obrázek B.3: Schéma propojovací desky druhé strana