VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MODULÁRNÍ ZOBRAZOVACÍ SYSTÉM NA BÁZI LED DIOD

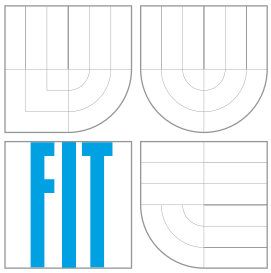
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ ŠARMAN

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MODULÁRNÍ ZOBRAZOVACÍ SYSTÉM NA BÁZI LED DIOD

MODULAR DISPLAY SYSTEM WITH LED DIODES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — ONDŘEJ ŠARMAN
AUTHOR

VEDOUCÍ PRÁCE — Ing. VÁCLAV ŠIMEK
SUPERVISOR

BRNO 2015

Abstrakt

Cílem této práce bylo vytvoření modulárního displeje za pomoci levných technologií. Tento displej měl komunikovat s počítačem za pomoci bezdrátové technologie Bluetooth. V tomto dokumentu lze najít popis komunikačních technologií, rozbor hardwarových technologií a popis implementace.

Abstract

The main goal of this work is to create modular display using cheap technologies. This display schould communicate with PC using Bluetooth. Description of communication technologies, hardware technologies and implementation details can be found in this document.

Klíčová slova

Led, Modulární displej, CAN, Kinetis, HCS08, SPI, Bluetooth, GIF, RGB, PWM

Keywords

Led, Modular display, CAN, Kinetis, HCS08, SPI, Bluetooth, GIF, RGB, PWM

Citace

Ondřej Šarman: Modulární zobrazovací systém na bázi LED diod, bakalářská práce, Brno, FIT VUT v Brně, 2015

Modulární zobrazovací systém na bázi LED diod

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana ing. Václava Šimka

.....
Ondřej Šarman
20. května 2015

Poděkování

Mé největší díky patří Ing. Václavu Šimkovi za poskytnutí užitečných rad ohledně projektu, hardwarového návrhu a celkově nápadů pro cíle projektu. Dále děkuji jemu, a celé škole, za propůjčení nástrojů, ať už softwarových nebo hardwarových a hlavně poskytnutého vzdělání. Nakonec bych rád poděkoval Bc. Jiřímu Chytilovi za rady okolo hardwarových částí práce.

Obsah

Kapitola 1

Úvod

Tento projekt vyjadřuje snahu o vytvoření nízkonákladové verze modulárního displeje, podobného těm na Piccadilly Circus v Londýně nebo na Times Square v New Yorku.

Obrazovky tohoto typu jsou používány například na festivalech, sportovních událostech, automobilových závodech a dopravnách informačních tabulích. Takovéto displeje však nalézají využití i ve vnitřních prostorách jako pozadí pro různé televizní show, natáčené v divadlech; obrazové stěny v klubech, výstavách a autosalonech, ale i domácnostech nejbohatších lidí.

Hlavní nevýhodou těchto high-endových obrazovek je to, že jsou pro běžného člověka v podstatě nepoužitelné, zejména kvůli jejich ceně.

Cílem je tedy malá, levná, uživatelsky přívětivá verze obdobného displeje. Základním rysem by měla být spousta komunikačních kanálů, po kterých lze do displeje posílat data. Mezi tyto kanály patří v první řadě Bluetooth, dále pak Ethernet a USB.

Představa fungování displeje (např. pro Bluetooth):

- V počítači bude jednoduchý software, který se pokusí připojit k displeji.
- Displej automaticky přijme požadavek na párování.
- Poté uživatel posílá soubory (obrázky/videa).
- Displej automaticky zobrazuje příjmaná data.

Sestavení displeje by mělo být intuitivní a potřeba propojování a konfigurování omezena na minimum.

Kapitola 2

Teoretická část

2.1 Inspirace

V praxi jsou již tato zařízení velmi rozšířená a jejich vývojem se zabývá spousta velkých firem. Mezi nejznámější patří *Barco* nebo *Darkronics*.

Displeje těchto výrobců sestávají z poněkud propracovanějších modulů. Jeden takový modul (pro vnitřní použití) váží okolo 10-ti kg a jeho rozměry se pohybují od 40x40cm. Rozlišení modulů se začíná na 48*48px a roste s velikostí modulu. Barevné rozlišení displejů je většinou 16 bitů na barvu, což dává hloubku $281*10^{12}$ barev. Obnovovací frekvence se pohybuje od 1,2KHz do 3,2KHz. K propojení používají SATu nebo optiku[1]. Cenou se tyto firmy nijak zvlášť nechlubí a i oficiální distributoři sdělí cenu až po založení uživatelského účtu.

Praxe je taková, že existují firmy, které disponují těmito obrazovkami. Tato firma je na dobu festivalů a sportovních akcí přivezou, namontují, řeší jejich provoz a po skončení si je zase rozmontuje a odveze. To vše za pouhých několik desítek tisíc Korun na den [18]. V české republice jsou to například firmy Gold Office s.r.o., AVLED s.r.o. nebo SCREEN 21 s.r.o.

2.2 Driver

Podle představy by měl driver dokázat komunikovat po více kanálech, takže cílem je, aby byl schopen používat Bluetooth, Ethernet, USB a možná i SD kartu. Další, velice důležitý požadavek je, že musí implementovat protokol CAN.

Řídící modul musí „přemýšlet“ nejen o tom, která data poslat, kterému modulu, ale i komunikovat po přístupových kanálech a v ideálním případě i zpracovávat různé typy obrazových dat.

Z tohoto důvodu by měl být tvořen výkonným procesorem. Více o driveru v kapitole Driver [3.2].

2.3 Moduly

Každý modul bude mít rozlišení 16x16 RGB LED diod. Ty bude možno libovolně skládat do větších ploch. Dohromady budou propojeny průmyslovou sběrnicí CAN, což má, bohužel, za následek nízké barevné rozlišení nebo pomalou obnovovací frekvenci.

[1] Informace získány ze stránek http://www.barco.com/ [14].

Více o modulech v kapitole Modul [3.3].

2.4 Komunikační kanály displeje

Komunikaci s počítačem řeší výhradně driver. Pro úplnost, některé výhody oddělení:

- Zjednodušení hardwarového řešení.
- Snížení ceny jednoho modulu.
- Při přechodu na jinou technologii komunikace je třeba vyměnit pouze driver.

Výhodou implementace driveru do každého modulu by byla naprostá autonomita jednotlivých modulů. Toto řešení se, nicméně, úplně vymyká cíli této práce.

2.4.1 Bluetooth

Protokol bluetooth je bezdrátová technologie, která byla vyvinuta firmou Ericsson v roce 1994. Původně měla představovat bezdrátovou verzi protokolu RS-232 [15].

Bluetooth pracuje v pásmu ISM (od 2.402 GHz do 2.480 GHz), ve kterém využívá 79 kanálů. Toto pásmo je ve většině zemí pásmo nezatíženo licencí. Dosah se uvádí do 10-ti metrů, přičemž se spotřeba pohybuje okolo 0dBm (1mW). Dosah lze, nicméně, zvýšit na 100 metrů zesílením vysílače na 20dBm [16]. Z toho, v každém případě, plyne, že Bluetooth je primárně určen pro sítě PAN (personal area network), což je propojení zařízení v blízkém okolí uživatele.

Protokol

Jádro systému Bluetooth tvoří 4 základní vrstvy.

- L2CAP vrstva
- LMP Linkový manažer
- Vrstva Baseband
- RF vrstva

Celkový pohled na model protokolu ukazuje obrázek 2.1.

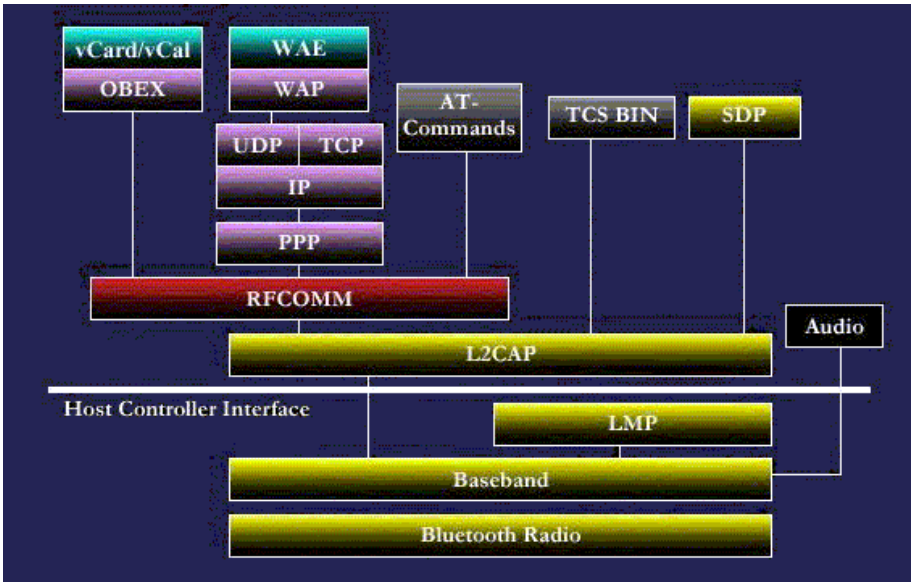
L2CAP vrstva (*Logical Link Control and Adaptation Protocol*) je zodpovědná za zajištění QoS (quality of service), výběr aktuálního z vyšších protokolů, rozdělování a zpětné sestavování paketů.

Vrstva LMP (*Link Manager Protocol*) zajišťuje vytvoření spojení, což zahrnuje bezpečnostní funkce, ověřování a šifrování. Dále zajišťuje vyjednávání vlastností spojení.

Vrstva Baseband zajišťuje veškeré propojení s fyzickou vrstvou (bezdrátovým vysílačem). Jejím hlavním úkolem je zajištění přístupu ke všem modulům, ke kterým byl dohodnut přístup při vyjednávání spojení. Dále je zodpovědná za kódování a dekódování RF paketů. Tato vrstva také řídí tzv. frekvenční hopy, což znamená přenášení paketů v definovaných časových intervalech na definovaných frekvencích.

RF (*rádiová*) vrstva transformuje datový tok a baseband pakety do požadovaného formátu.

Spodní 3 vrstvy jsou často spojovány v podsystém *Bluetooth controller*. Ty jsou s vyššími vrstvami propojeny rozhraním *Host Controller Interface*.



Obrázek 2.1: Bluetooth protokol[16]

AT-Command set

Sada příkazů, které umožňují nastavení parametrů sériové linky. V této práci byl použit modul LM780, který implementuje základní sadu AT-příkazů.

Příkazy jsou posílány ve formě `AT+...<cr>`. Odpovědí mohou být `<cr,lf>OK<cr,lf>` nebo `<cr,lf>ERROR<cr,lf>`.

Tabulka 2.1 ukazuje výčet základních příkazů, implementovaných v modulu LM780, které jsou použity v implementaci.

Bližší informace v AT Command Manuálu [6]

Výhody použití bluetooth

- Je implementován v téměř každém, i ne tak „chytrém“, zařízení.
- Softwarová implementace není příliš náročná.
- Od verze 2 je již velmi zjednodušeno párování.
- Od verze 3 je rychlost 24 Mbit/s.
- Žádné vodiče.

2.4.2 Ethernet

Ethernet je součástí modelu počítačových sítí. Jedná se o drátovou formu propojení fyzických vrstev síťového modelu.

Síťové protokoly se dělí na dva nejzákladnější.

- ISO/OSI
- TCP/IP

Tabulka 2.1: Příkazy AT, implementované v LM780

Příkaz	**Význam**
`AT`	Testovací příkaz. Pokud modul funguje správně, tak vrací OK
`AT+ENQ`	Aktuální nastavení modulu
`AT+RESET`	Vrací modul do továrního nastavení
`AT+BAUD`	Nastavení rychlosti sériové linky
`AT+STOP`	Počet stop bitů
`AT+PAR`	Počet paritních bitů
`AT+FLOW`	Nastavení flow-control
`AT+MODEM`	Nastavení modemových signálů
`AT+ROLE`	Nastavení role master/slave
`AT+NAME`	Nastavení jména modulu
`AT+BOND`	Párování
`AT+SLEEP`	Šetření energií
`AT+IOTYPE`	Konfigurace možností zařízení
`AT+MITM`	Zabezpečení
`AT+PASSCFM`	Potvrzení správnosti párovacího hesla

ISO/OSI

ISO/OSI je referenční model, který byl vytvořen a následně v roce 1987 vydán standardizační organizací OSI. Tento model je dnes považován za primární architekturu pro komunikaci mezi počítači.

ISO/OSI model tvoří 7 vrstev. V rámci jednoho zařízení komunikuje vrstva vždy jen s vrstvou nad a pod. V rámci sítě spolu komunikují pouze odpovídající-si vrstvy.

Jednotlivé vrstvy:

- *Aplikační vrstva* definuje aplikace, komunikující po síti.
- *Prezentační vrstva* zajišťuje převod dat z formátu, který využíva aplikace (binární/ASCII) do jednotného formátu (podle standardu ISO 8824 je to formát ASN.1)
- *Relační vrstva* zajišťuje vytvoření a udržování spojení mezi komunikujícími aplikacemi.
- *Transportní vrstva* garantuje spolehlivý přenos. Definuje několik typů spolehlivosti služeb.
- *Síťová vrstva* zajišťuje adresování a směrování dat.
- *Linková vrstva* popisuje přenos dat na konkrétní lince a adresování na linkové vrstvě.
- *Fyzická vrstva* definuje fyzické vlastnosti linky.

TCP/IP

Tento model vznikl již v 60. letech 20. století. Síť měla být róbustní, decentralizovaná a snadno implementovatelná.

Architektura TCP/IP modelu je oproti ISO/OSI modelu pouze 4-vrstvá.

- *Aplikační vrstva* je tvořena veškerými procesy, komunikující po síti a řeší i reprezentaci dat.
- *Transportní vrstva* vytváří logické spojení mezi aplikacemi, komunikujícími po síti.
- *Internetová (IP) vrstva* vytváří datagramy, adresuje je a snaží se je s co *největším úsilím* doručit.
- *Vrstva fyzického rozhraní* popisuje standardy pro fyzická média a elektrické signály.

Vrstva fyzického rozhraní bývá většinou implementována na síťové kartě, ovládání ostatních vrstev musí zajišťovat operační systém (v tomto projektu firmware driveru).

Veškeré informace o síťvých modelech byly čerpány z opory předmětu ISA na FIT/VUT[5].

2.4.3 USB

USB je komunikační protokol, který umožňuje připojit až 127 zařízení k jednomu kořenovému rozbočovači (za použití až 5 vrstev rozbočovačů). Je tvořen 4-mi vodiči:

- napájecím,
- zemnícím,
- datovým D+,
- a datovým D-.

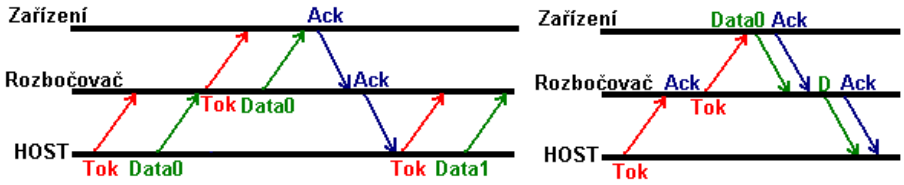
Každý přenos dat musí být iniciován kořenovým rozbočovačem. Komunikace je definována jako transakce obvykle 3 paketů (*polling*).

- *ToK* - Token packet: ID, adresa zařízení; iniciováno hostitelským zařízením.
- *Data/nic* - Data přenáší hostitel nebo zařízení.
- *ACK* - Handshake packet - potvrzení.

Ukázka komunikace je na obrázku 2.2.

Každý paket obsahuje CRC kód, který je schopen detekovat chybu.

Informace čerpány z laboratoří z bředmětu IPZ na FIT/VUT[19].



Obrázek 2.2: USB protokol[19]

2.5 Komunikace uvnitř displeje

Základem modulárních zařízení je bezpochyby komunikace jednotlivých prvků systému.

K tomu, aby mohl displej zobrazovat celistvý obraz, je třeba použít sběrnici, která podporuje režim *Master-Slave*, zvládne přenášet potřebné množství dat a její řadič je implementován na jednoduchých mikrokontrolérech.

Nabízí se spousta sběrnic, které jsou více, či méně vhodné.

2.5.1 SPI

SPI je jedna s nejrozšířenějších sběrnic v elektronickém průmyslu a to díky jednoduché hardwarové implementaci. Pracuje v režimu Master-Slave a je implementována jako posuvné registry, které si vzájemně mění data.

Jako výhody lze uvést neomezená rychlost, dostupnost v téměř každém mikrokontroléru. Další výhodou, je její plná duplexita.

Nevýhodami této sběrnice jsou vysoká náchylnost k rušení a hlavně absence adresace. Adresovaný prvek je vybírán pomocí separátního vodiče „Slave Select“, tudíž s každým novým modulem by přibyl jeden nový vodič a potřeba prvku, který by tento vodič aktivoval (pin na řídícím mikrokontroléru nebo nějaký dekodér) [4].

2.5.2 I^2C

Master-slave sběrnice, která klade důraz na jednoduchost a nízké náklady na implementaci. Komunikovat může obousměrně, nicméně vždy vysílá maximálně jeden uzel.

I^2C je tvořena 2-ma vodiči. Jeden (*SDA*) pro data a druhý (*SCL*) pro synchronizaci. Umožňuje adresovat až 127 zařízení. Po odeslání adresy (vybrání uzlu) je možné poslat neomezené množství dat (ukončení přenosu dat se děje speciální kombinací hran na vodičích) [4].

Nevýhoda této sběrnice je opět náchylnost k rušení kvůli synchronizaci a její rychlost je velmi nevyhovující (běžné 8-bitové mikrokontroléry podporují maximálně 100kbps *viz.: http://cz.farnell.com/*).

2.5.3 CAN

Jedná se o protokol firmy Bosch z 80.let 20. století [9]. Tato sběrnice se hojně využívá v automobilovém a leteckém průmyslu.

Její největší předností je velká odolnost vůči rušení a poměrně vysoká rychlost (1Mbit/s).

Typ sběrnice je multi-master, nicméně díky ostatním výhodám tohoto protokolu nebude velký problém ji použít. Problémem mohou být poměrně velká režie, tvořící okolo 50% velikosti paketu, nemožnost větvení a lehce drahá implementace v mikrokontrolérech.

Stejně, jako většina protokolů, je i CAN rozdělen do vrstev. Skládá se ze 3 vrstev.

- *Objektová vrstva* zjišťuje, které zprávy je třeba v kterou chvíli poslat; které zprávy z transportní vrstvy se mají přijmout. Tato vrstva tvoří rozhraní pro aplikace, využívající tento protokol.
- *Transportní vrstva* představuje jádro protokolu CAN. Provádí zapouzdření a rozbalení paketů, kontrolu chyb a hlášení chyb přímo na sběrnici. Přidává speciální časovací bit do paketu. Při arbitraci provádí vysílání adresy a kontrolu skutečného stavu sběrnice

(více o arbitraci[2.5.3]) a kontroluje, jestli je sběrnice volná. Funkčnost transportní vrstvy je dána a nelze ji měnit.

- *Fyzická vrstva* vykonává samotné vysílání a čtení stavu sběrnice. Dbá na aktuální elektrické vlastnosti sběrnice - nastavení fyzické vrstvy musí tedy ctít všechny uzly na sběrnici.

Elektrické vlastnosti

Sběrnice je tvořena dvěma vodiči (*CAN_L* a *CAN_H*). Mohou nastat dvě situace:

- Na obou vodičích je stejné napětí (přibližně 1/2 referenčního napětí) – tento stav se nazývá *recesivní stav* a je považován za logickou 1.
- Na vodiči CAN_H je referenční napětí a na CAN_L je nízké napětí, blížící se logické 0 – tzv. *dominantní stav*, který reprezentuje logickou 0.

Sběrnici není možné větvit a na obou koncích musí být zakončena typicky 120Ω odporem.

Časování bitu

Tato sběrnice je asynchronní a synchronizace probíhá vždy na hraně signálu. Za účelem synchronizace jsou také vkládány synchronizační bity (pokud bylo odesláno 5 bitů se stejnou hodnotou, je automaticky vložen synchronizační bit s hodnotou opačnou). Jeden bit je rozdělen do více segmentů:

- *Synchronizační segment* – v tomto okamžiku je očekávána hrana a dochází k synchronizaci uzlů.
- *Propagační segment* – čas, který kompenzuje fyzické zpoždění sběrnice, vstupních komparátorů a výstupních řadičů uzlů.
- *Fázovací segment 1 a 2* – segmenty, které kompenzují synchronizační chyby.
- *Vzorkovací bod* – moment mezi fázovacími segmenty resp. na konci fázovacího segmentu 1.
- *Čas pro zpracování informace* – počíná fázovacím segmentem 2.

Délky časových úseků jsou násobky ČASOVÉHO KVANTA. Jeho délka je odvozena od lokálního časovače. Typicky je dána vzorcem:

$$\text{ČASOVÉ KVANTUM} = \text{m} * \text{MINIMÁLNÍ ČASOVÉ KVANTUM}$$

kde m je hodnota předděličky.
Synchronizační segment je dlouhý 1 ČASOVÉ KVANTUM.
Propagační segment je dlouhý 1,2,...,8 ČASOVÉ KVANTUM.
Fázovací segment 1 je dlouhý 1,2,...,8 ČASOVÉ KVANTUM.
Fázovací segment 2 je dlouhý maximálně tak, jako fázovací segment 1.
Čas pro zpracování informace nůže být maximálně 2 ČASOVÉ KVANTA dlouhý.

Typy zpráv

Pro CAN jsou typické 4 typy zpráv.

- *Datový rámec* obsahuje data od jednoho uzlu pro, potencionálně, všechny uzly.
- *Žádost o data* – rámec, ve kterém uzel žádá data od jiného uzlu.
- *Chybový rámec.*
- *Rámec přetečení* vyslán především tehdy, když uzel potřebuje delší čas pro zpracovaní dat.

Arbitrace

Začátek rámce je vždy jen v jeden časový okamžik a v ten může začít vysílat více uzlů současně. Proto existuje systém, který jednoznačně určí, který uzel bude moci vysílat. Tento systém se nazývá *arbitrace* a vítězem je vždy uzel s nejnižším ID.

Uzly začínají vysílat adresu od nejvýznamějšího bitu. Pokud se jeden uzel snaží odeslat log. 1 a druhý log. 0, na sběrnici se objeví dominantní stav, který reprezentuje logickou 0. Z toho plyne, že čím nižší má uzel adresu (ID), tím má větší šanci vyhrát arbitraci. Uzel, který vyslal logickou 1 a přečetl 0 přestává vysílat.

Datový rámec

Datové rámce mohou být ve standardním formátu (11-bitová adresa) nebo v rozšířeném formátu (29-bitová adresa).

Zpráva je rozdělena do několika polí:

- *identifikátor* 11-bitový, nebo nejvýznamějších 11 bitů pro rozšířený formát. Obsahuje adresu odesílajícího uzlu.
- *RTR* (Remote Transmission Request). Jeden bit, udávající, jestli se jedná o datový rámec nebo žádost o data. V rozšířeném formátu je tento bit vždy recesivní a RTR bit je přesunut za zbytek identifikátoru. V Datovém rámci je tento bit vždy dominantní (log. 0).
- *IDE* jeden bit, který identifikuje, jestli se jedná o standardní, nebo rozšířený formát. Pokud je tento bit recesivní, rámec je rozšířený a následuje zbylých 18 bitů adresy. Poté je odvysílán bit RTR a jeden rezervovaný bit.
- *jeden rezervovaný bit*
- *DLC* – 4-bitové pole, udávající délku dat (v bytech).
- *Data* – počet bytů dat musí korespondovat s počtem, uvedeným v DLC.
- *CRC* – spodních 15 bitů cyklické redundantní kontroly, počítané ze všech předchozích polí. Algoritmus je uveden zde (1).
- *ERC* (End Of CRC) – 1 bit.
- *ACK* – recesivní bit, kdy se očekává odpověď od přijímacího uzlu.
- *EOF* (End of frame) – 1 recesivní bit.
- *INTERFRAME* – 7 bitů klid na sběrnici.

Algoritmus 1: Algoritmus CRC ve sběrnici CAN

```
1: CRC_RG = 0 ; // vynulování posuvného registru
2: repeat
3:     CRCNXT = NXTBIT xor CRC_RG(14);
4:     CRC_RG(14:1) = CRC_RG(13:0) ; // posunutí vlevo
5:     CRC_RG(0) = 0;
6:     if CRCNXT then
7:         CRC_RG(14:0) = CRC_RG(14:0) xor (0x4599);
8:     end if
9: until dosaženo CRC SEQUENCE or nastal ERROR;
```

Žádost o data

Rámec Žádost o data má podobnou strukturu, jako datový rámec. Jediný rozdíl je, že bit RTR je recesivní (log. 1) a data neposílá odesilatel, nýbrž uzel, od kterého data potenciálně žádá.

DLC udává uzel, žádající data a udává kolik bytů dat je třeba poslat.

Chybový rámec

Chybový rámec je spíše reakcí sběrnice na chybu. Při chybě, což znamená 6 a více bitů stejné hodnoty. Při zjištění tohoto stavu přestanou všechny uzly vysílat a na sběrnici musí být recesivní stav alespoň 3 bity.

Implementace je taková, že uzel „vysílá“ recesivní bity a čte dominantní, dokud nepřečte recesivní bit. Od té doby čeká po dobu 7 bitů po které může začít odesílání nového rámce.

Rámec přetečení

V podstatě jde o chybový rámec, s tím rozdílem, že pokud je odvysílán v době klidu mezi datovými rámci (INTERFRAME), tak musí následovat 8 recesivních bitů.

Informace byly čerpány z BCANPSV2.0/D[7] a handout_canbus2[9].

2.6 Mikrokontroléry

Mikrokontroléry jsou integrované obvody, které v sobě obsahují procesor a periferie. Mezi tyto periferie patří např. sériové sběrnice, AD/DA převodníky nebo RAM. Na trhu existuje spousta firem, které se zabývají jejich výrobou. Mezi nejznámější patří Atmel, Texas Instruments nebo Freescale.

Mikrokontroléry jsou určeny pro aplikaci do vestavěných systémů, ať už k jejich řízení nebo ovládání nějaké jejich části. Existuje spousta různých mikrokontrolérů a ty se liší jádrem procesoru a typem a množstvím periférií. Různí výrobci si také udávají, jak se budou jejich mikrokontroléry programovat, popřípadě debugovat.

Mikrokontroléry lze rozdělit do dvou skupin podle jejich architektury.

2.6.1 CISC

Complex Instructions Set Computing

Architektura CISC implementuje velké množství instrukcí proměnné délky. Velké množství těchto instrukcí také pracuje s pamětí. Jádro také disponuje pouze několika málo registry, takže je třeba často přistupovat do paměti. To má za následek celkově malou rychlost procesoru a složité optimalizace programu.

Mikrokontroléry s touto architekturou lze ještě dále rozdělit do dvou skupin. První využívá Von-Neumannovskou architekturu, což znamená, že pro data i program je použita jedna fyzická paměť. Naproti tomu existuje architektura Harvardovská, která má fyzicky oddělenu paměť programu od paměti pro data.

Většina dnešních mikrokontrolérů využívá Von-Neumannovskou architekturu, protože je implementačně jednodušší. Problém sdílené paměti je sdílená sběrnice, kterou je třeba posílat data i instrukce [2].

Jelikož je tato architektura starší, je i trh s těmito mikrokontroléry rozvinutější a mikrokontroléry jsou rozmanitější ohledně periférií. Výsledkem toho je, že si vývojář může vybrat z již existujících procesorů jeden, který vyhovuje přesně jeho aplikaci.

CISC procesory jsou také levnější díky tomu, že náklady na výrobu a vývoj RISC procesoru jsou o mnoho vyšší. Tyto procesory se také jednoduše programují a paměť vyžífají velmi efektivně [11].

2.6.2 RISC

Reduced Instructions Set Computing

Architektura, která je dnes využívána v téměř všech chytrých telefonech a tabletech. Postupně se dostává i do jednodužších vestavěných systémů.

Rozdíl od architektury CISC(2.6.1) je v tom, že využívá instrukce pevné délky. Tím pádem je možné zvýšit frekvenci, na které procesor pracuje. Díky tomu se zvýší celkový výkon procesoru a snížit jeho spotřebu. I přesto, že na některé operace je třeba více instrukcí, než v architektuře CISC je výsledná doba trvání programu kraší díky vyšší frekvenci. Další změna oproti CISC je v použití pouze 2 instrukcí pro práci s pamětí (LOAD a STORE).

Největší změnu přineslo zavedení registrových polí, kdy jeden procesor obsahuje několik desítek registrů pro obecné využití, čímž se omezí potřeba práce s pamětí.

Dnes jsou nejpoužívanější jádra společnosti ARM ze série Cortex-M. Ta se dělí na M0-M7.

- *Cortex-M0* – nejjednodužší procesory s ARM jádrem, srovnatelné s dražšími jednoduchými 8-bitovými procesory.
- *Cortex-M0+* – jednoduché procesory s nízkou spotřebou.
- *Cortex-M1* – speciálně navrženy pro implementaci do FPGA.
- *Cortex-M3* – nejrozšířenější procesor pro různé aplikace.
- *Cortex-M4* – specializace na zpracování signálů.
- *Cortex-M7* – nejsofistikovanější procesory s nejvyšším výkonem.

Tyto procesory jsou pak základem mikrokontrolérů firem Atmel, Freescale, TI nebo NXP.

Informace čerpány z datasheetu ARM7DI[1], stránek společnosti ARM[13] a e-shopu Farnell[17].

2.7 Obrazové formáty

Cílem aplikace je podporovat co nejvíce obrazových formátů, nicméně jako demonstrační byl použit formát GIF.

2.7.1 GIF

Formát GIF byl zvolen zejména kvůli tomu, že jeho parsování je relativně jednoduché, paměťově nepříliš náročné (pokud je používáno správně) a licence kompresního a dekpomresního algoritmu je dnes již zdarma. Specifikace GIF dokonce umožňuje seřazení barev v paletě podle důležitosti, což je vítaná vlastnost zejména kvůli omezení rychlosti sběrnice CAN.

Formát gif je organizován proudově, narozdíl od bitmapových formátů. Skládá se z bloků a hlaviček. Bloky se mohou skládat ze subbloků, které začínají bytem „počítadlo“. Ten indikuje počet datových bytů bloku. Subbloky jsou ukončeny počítadlem s hodnotou 0.

Maximální barevná hloubka je 8 bitů (pro všechny složky). Pro zobrazení RGB dat je použita paleta barev. Obrazová data pak udávají index do této palety. Tato data jsou vždy komprimována metodou LZW. Obvyklá komprese této metody u obrázků GIF je 40%.

Kompresní metoda LZW

Algoritmus, který vymysleli pánové Abraham Lempel a Jakob Ziv v roce 1977.

Tento algoritmus je založen na slovníkovém kódování. Při kompresi i dekompresi je budován slovník, který obsahuje vzorky (podřetězce) původního toku dat.

Pokud se vzorek nevyskytuje ve slovníku, je podle obsahu informace vytvořena kódová fráze, která je vložena do slovníku. Tato fráze je pak zapsána do výstupního toku.

Pokud se fráze objeví znovu, je rovnou zapsána na výstup.

Algoritmus LZW nepotřebuje do souboru ukládat slovník, protože jeho počáteční stav je vždy dán 256 znaky ASCII tabulky. Z těchto hodnot je pak dynamicky budován zbytek slovníku v momentě komprese i dekomprese.

Patent na používání algoritmu LZW vypršel 20.6.2003 pro USA a pro ostatní země, kde se datum podepsání patentu lišilo již vypršel také[3].

Organizace souboru

Existují 2 verze specifikace GIF: 87a a 89a. Verze 89a přidává možnost vkládat text a grafiku do jednoho souboru.

Formát GIF rozděluje soubor do několika logických celků.

- *Hlavička* obsahuje informace o souboru (že se jedná o GIF a jeho verzi), šířku a výšku plátna v pixelech, informaci o barvách a poměru stran.
- *Globální tabulka barev* je přítomna, pokud je tak uvedeno v hlavičce. Je složena ze sekvence 3-bytových čísel, udávající hodnotu červené, zelené a modré barvy.
- *Lokální popisovač obrázku* je blok, obsahující lokální hlavičku, může obsahovat lokální tabulku barev a obrazová data. Tento typ bloku je vždy uveden 1-bytovou hodnotou `0x2C`.
- *Trailer* – jediný byte, který ukončuje soubor GIF. Jeho hodnota je `0x3B`.

- *Rozšíření* obsahuje informace jako například zpoždění. Je uveden bytem `0x21`. Rozšíření se dále může větvit do různých typů bloků, ale to už je nad rámec tohoto projektu.

Informace čerpány z knihy Encyklopedie grafických formátů [8].

Kapitola 3

Řešení

Celý projekt je složen ze 3 základních částí.

- Uživatelská aplikace pro počítač.
- Driver displeje.
- Soustava modulů displeje.

3.1 Uživatelská aplikace

V tuto chvíli jde pouze o konzolovou aplikaci, která se pokusí přes Bluetooth vyhledat displej a s tím se spárovat. Pro úspěšné spárování je třeba potvrdit požadavek párování tlačítkem na Driveru.

Po spárování by měl operační systém automaticky vyhledat ovladače a umožnit otevření virtuální sériové linky.

Implicitní nastavení sériové linky v Bluetooth modulu, který byl použit 3.2.1:

Přenosová rychlost	19 200 Bd
Data	8 bitů
Stop bit	1
Parita	NE
Flow control	NE

Po úspěšném otevření sériové linky požádá aplikace o vložení cesty k obrázku, který se má zobrazit na displeji. Program umožňuje parsování gifu, který dekomprimuje obrazová data a ta odešle i s paletou a hlavičkou po sériové lince.

Z důvodu inspirace bylo použito kódu GraphApp [1] [10]. V aplikaci je použita pouze dekódovací část.

Po úspěšném odeslání dat se program vrací do stavu, kdy čeká na zadání cesty k obrázku.

3.2 Driver

Pro driver celého displeje bylo rozhodnuto použít vývojový kit *FRDM-K64F* (3.1) společnosti Freescale.

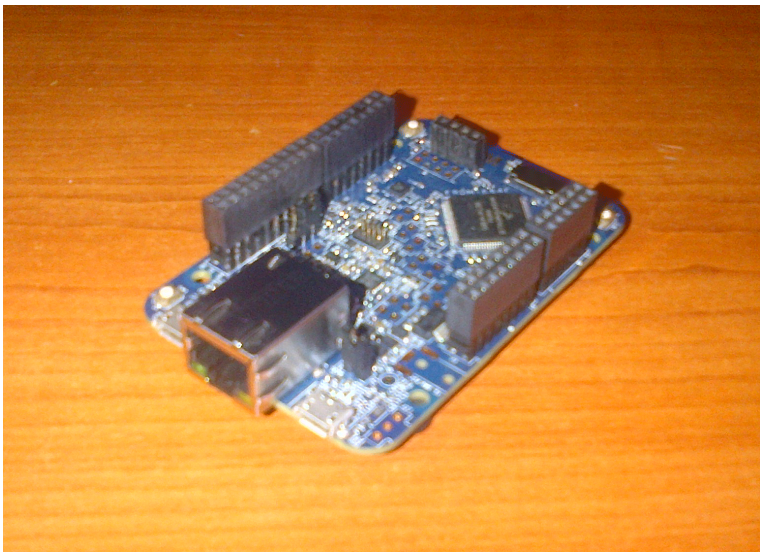
[1] Zdrojové kódy zaštiťuje licence *App Software License*, umožňující jejich volné používání v originální i zmodifikované podobě: http://enchantia.com/graphapp/faq/license.htm

Tento kit obsahuje mikrokontrolér Kinetis K64F (MK64FN1M0VLL12). Jádro tohoto mikrokontroléru tvoří procesor Cortex-M4 s FPU a frekvencí 120MHz.

Rysy mikrokontroléru:

- *1MB programové paměti*
- *256KB RAM*
- *16-ti kanálový řadič DMA* – důležitý při rozesílání dat do modulů
- *CAN, ETHERNET, 6xUART, USB, SDHC*

I přes vyšší pořizovací cenu je tento mikrokontrolér velice výhodný díky svému vysokému výkonu a podpory celé řady periférií. V pozdější době by měl obsáhnout i program pro dekódování obrázků.



Obrázek 3.1: Modul driveru

Ke kitu je připojen shield, obsahující Bluetooth modul LM780, převaděč spojitosti pro CAN sběrnici a napájecí jack. Shield také obsahuje 2 jumpery, kterými lze připojit signály CTS (Clear To Send) a RTS (Ready To Send). Tyto signály jsou pozůstatkem z dob minulých, nicméně modul LM780 implicitně vyžaduje jejich použití.

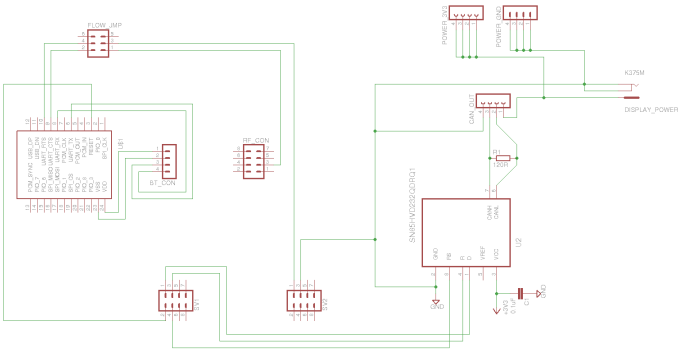
Schéma zapojení shieldu viz. 3.2.

3.2.1 Bluetooth modul LM780

Z nabídky Bluetooth modulů byl vybrán model LM780-0223 od společnosti LM Technologies. Tento typ již podporuje verzi Bluetooth 2.1 + EDR. Dále implementuje základní sadu AT-Command setu pro spárování a otevření sériové linky.

Vlastnosti modulu:

- Dosah integrované antény je až 25m v otevřeném prostoru.
- Podpora zabezpečeného párování.



Obrázek 3.2: Schéma komunikačního shieldu pro driver

- Plná podpora EDR (až 3Mbps)

Modul bohužel neumožňuje použití protokolu FTP (protokol pro přenos souborů), takže není možné posílání dat přes Bluetooth, ale je nutné posílat je byte po bytu po sériové lince.

3.2.2 Firmware driveru

Firmware řídícího modulu se skládá z několika částí.

- *Počáteční inicializace* – V této fázi dochází k nastavování konfigurace samotného řídícího mikrokontroléru. To obnáší povolení různých přerušení, nastavení UART pro komunikaci s Bluetooth modulem, inicializaci řadiče CAN sběrnice, popřípadě nastavení dalších periférií. V této fázi dochází také ke kontrole správné funkčnosti Bluetooth modulu.
- *Rozpoznání tvaru displeje* – Po inicializaci je spuštěna komunikace po CAN sběrnici, kde driver zjišťuje kolik modulů a v jakém tvaru jsou poskládány.
- *Chybový stav* – Pokud při počátečních fázích vznikl neřešitelný problém, tak program skončí v nekonečné smyčce, kde rozbliká červenou LED diodu a zastaví veškeré provádění.
- *Cyklus displeje* – Po úspěšné inicializaci driver čeká na data a ta pak rozesílá do správných modulů.

Počáteční inicializace

Při inicializaci mikrokontroléru je provedeno i základní otestování komunikace s Bluetooth modulem. Pokud na základní příkaz „AT“ odpoví „AT<cr><lf>OK<cr><lf>“, pak je vše v pořádku. Pokud odpoví „AT<cr><lf>ERROR<cr><lf>“, pak je problém pravděpodobně řešitelný.

Modul je také potřeba nastavit do požadované konfigurace (3.1).

Rozpoznání tvaru displeje

Po inicializaci periférií odešle driver, po CAN sběrnici, broadcastovou zprávou požadavek na nejnižší ID. Na tuto zprávu se pokusí odpovědět všechny moduly a díky arbitračnímu

Tabulka 3.1: Počáteční konfigurace Bluetooth modulu

Operace	Příkaz
Nastavení jména	AT+NAME=Modular Display
Nastavení možnosti zařízení („Display with Yes or No“)	AT+IOTYPE1
Vypnutí Flow Control	AT+FLOW-
Vypnutí funkce Modem	AT+MODEM-
Zapnutí zabezpečení párování	AT+MITM+

mechanizmu dostane jedinou zprávu od modulu s nejnižším ID. Další postup je popsán algoritmem 2.

Algoritmus 2: Načtení adres a pozic do lineárního seznamu

```
1:  Ziskej modul s nejnizsim ID
2:  Vloz do seznamu s pozici [0,0]
3:  Ukazatel active nastav na tento zaznam
4:  while active != NULL do
5:      ziskej adresu modulu nad active
6:      if existuje and neni v seznamu then
7:          vloz na konec seznamu se souradnici active->Y - 1
8:      end if
9:      ziskej adresu modulu vpravo active
10:     if existuje and neni v seznamu then
11:         vloz na konec seznamu se souradnici active->X + 1
12:     end if
13:     ziskej adresu modulu pod active
14:     if existuje and neni v seznamu then
15:         vloz na konec seznamu se souradnici active->Y + 1
16:     end if
17:     ziskej adresu modulu vlevo active
18:     if existuje and neni v seznamu then
19:         vloz na konec seznamu se souradnici active->X - 1
20:     end if
21:     active = active->next
22: end while
```

Pozice modulů jsou do seznamu vkládány relativně k pozici modulu s nejnižším ID (0,0). Získání adres sousedních modulů probíhá následujícím způsobem:

- Driver odešle zprávu, obsahující adresu aktivního prvku a pozicí, ze které chce získat adresu.
- Aktivní modul zareaguje nastavením logické 1 na pinu, vedoucím na příslušnou stranu.
- Sousední modul (pokud je přítomen) musí odpovědět do 5ms nastavením log 1 na odpovídajícím pinu.
- Pokud se komunikace povede, odpovídá aktivovaný modul zasláním zprávy „OK“. Jinak odpovídá aktivní modul zasláním zprávy „ERROR“.

Po načtení všech adres je potřeba seznam seřadit podle pozice zleva doprava a shora dolů. Relativní pozice modul přepočítá tak, aby se souřadnice `[0,0]` nacházely v levém horním rohu. Toto umístění počátku je výhodné zejména kvůli tomu, že jsou tímto způsobem ukládány informace v obrázcích. Datová informace pro konkrétní modul řadič získá pomoci vzorce (3.1).

Listing 3.1: Vzorec pro získání dat modulu podle pozice.

```
#define MODULE_SIZE 16

int column = module[addressed]->pos_x * MODULE_SIZE + wanted_pixel_x;
int row = module[addressed]->pos_y * MODULE_SIZE + wanted_pixel_y;
Color *pixel = image[row][column];
```

Cyklus displeje

Po zinicializování displeje je program řídícího modulu „uvězněn“ do nekonečné smyčky, která končí vypnutím displeje nebo jeho resetem.

V této smyčce řeší dva základní úkoly:

- Příjem dat po některém z komunikačních kanálů.
- Rozeslání obrazových dat do jednotlivých modulů.

Kromě těchto úkolů však musí ošetřovat i jiné situace, jako například propálení diody nebo vypadnutí celého modulu.

Příjem dat po Bluetooth

Driver je ve slave módu, takže čeká na dvě možné události. První možností je požadavek na párování a tou druhou je otevření spojení od nějakého, již spárovaného přístroje.

Požadavek na párování přichází ve formě řetězce `PASSKEY CFM."XXXX-XX-XXXXXX",dddddd`, kde XXXX-XX-XXXXXX je mac adresa Bluetooth zařízení, požadující párování a dddddd heslo, které se musí shodovat na obou zařízeních.

Na tento požadavek je třeba odpovědět řetězcem `AT+PASSCFM=XXXXXXXXXXXX,c`, přičemž XXXXXXXXXXXX je mac adresa Bluetooth zařízení, požadující párování a `c` je buďto potvrzení (`Y`), nebo odmítnutí (`N`) požadavku.

Po úspěšném spárování může přijít požadavek na **otevření spojení** zasláním příkazu `CONNECT "XXXX-XX-XXXXXX"`. Pokud je zařízení v command modu, tak přejde do datového režimu a je schopno přijímat data.

Pro odpojení odešle master zařízení příkaz `DISCONNECT "XXXX-XX-XXXXXX"`. Tím Bluetooth přechází zpět do příkazového režimu a očekává připojení nebo požadavek na párování.

Rozeslání dat modulům

Rozesílání dat modulům se děje pouze prostřednictvím CAN sběrnice, proto tento fakt již nebude uváděn.

Po přijetí dat po některém z komunikačních kanálů nejdříve vytvoří paletu barev [1].

Tuto paletu poté odešle broadcastovou zprávou všem modulům. Následně odesílá data postupně všm modulům na sběrnici. To se děje následujícím postupem:

[1] Formát Gif používá paletu barev, u ostatních formátů může vzniknout nutnost tuto paletu vytvořit.

1. Driver odesílá data modulům v pořadí ze seznamu adres a pozic modulů vytvořeného při inicializaci (2). Adresu odešle broadcastovou zprávou na níž by měl adresovaný modul zareagovat.

2. Adresovaný modul si postupně vyžádá všechna data prostřednictvím rámce *žádost o data*. Data jsou ve formě indexů do palety barev.

3. Po úspěšném odeslání všech dat do modulu se algoritmus vrací na bod 1

Počet rámců *žádost o data* se vypočítá podle vzorce 22.

$$X = \frac{log_2 p * 64 * 4}{64} = 4 * log_2 p$$

Kde `X` je počet rámců a `p` počet záznamů v paletě barev.
Výsledek je zaokrouhlen nahoru.

3.3 Modul

Zobrazovací moduly lze považovat za nejdůležitější součást displeje. Základem modulu je DPS, která tvoří i nosnou základnu pro samotné zobrazovací prvky.

Jednotlivé moduly lze spojovat dohromady pomocí kolíkových konektorů.

3.3.1 Hardware modulu

Modul je složen z 8-bitového mikrokontroléru a PWM modulů. Schéma zapojení a seznam součástek viz příloha (C.3 a C.1).

Požadavky na mikrokontrolér byly nízká cena a hardwarová podpora CAN sběrnice.

Jednou z možností bylo použití Mikrokontroléru s jádrem ARM Cortex-M0. Bohužel i přestože by tyto mikrokontroléry měly nahrazovat 8-bitové CISC procesory, jsou jeho výkon a cena příliš vysoké pro tuto aplikaci.

Nakonec tedy bylo rozhodnuto použít mikrokontrolér *MC9S08DZ32* s jádrem HCS08. Architektura procesoru je CISC a může pracovat na frekvenci až 40MHz. Disponuje pamětí 32KB, řadičem CAN sběrnice (použitelný s knihovnou MSCAN) a řadičem SPI sběrnice, která je důležitá pro komunikaci s PWM moduly.

Jeden modul se skládá ze 4 maticových RGB displejů. Rozměry displeje jsou 60x60mm a obsahuje 8x8 RGB LED diod, které mají společnou anodu. Displeje fungují na principu časové multiplexe, což znamená, že v jednom časovém okamžiku svítí pouze 1 řádek tohoto displeje. V tento časový okamžik je třeba na každou diodu přivést PWM signál, aby bylo docíleno správného smíchání barev.

Za pomoci experimentu na platformě FitKit bylo zjištěno, že k dosažení pěkného obrazu, který rušivě nebliká je třeba přepínat řádky s frekvencí alespoň 200Hz. Z toho plyne, že PWM signál nebude možné vytvářet softwarově, ale bude potřeba speciální hardwarový prvek, který v ideálním případě umožní načítat data při zobrazování. Zároveň bude třeba, aby byl tento hardware schopen pracovat dostatečně rychle.

PWM moduly

Po průzkumu nabízených modulů bylo rozhodnuto použít výtvor společnosti Texas Instruments *TLC5946PWP*. Ten obsahuje 16 PWM výstupů s přesností 12 bitů. Výsledkem je možnost namixování 68.7 miliard barev. Zbytečnost takovéto barevné hloubky bylo rozhodnuto přehlížet s ohledem na nabídku jiných PWM modulů.

Podle datasheetu potřebuje tento modul externí zdroj hodinového signálu pro PWM (`GSCLK`). Je třeba přivést hodinový signál, jehož frekvence odpovídá požadavku na přepínání řádků s frekvencí 200Hz. Jedna perioda PWM potřebuje 2^{12} taktů a bylo by vhodné, aby bylo provedeno alespoň 5 period. Frekvence oscilátoru je tedy dána vztahem:

$$f = 200 * 2^{12} * 5 = 4\,096\,000\,\mathrm{Hz}$$

Jeden modul je složen ze 4 displejů 8x8, přičemž v jednom časovém okamžiku svítí jen jeden řádek každého displeje. Z toho plyne, že v jeden okamžik svítí 32 RGB LED. Kařdá z těchto diod ovšem potřebuje dedikovaný kanál PWM na každou barvu, takže celkem je potřeba 96 kanálů PWM.

Vznikla tedy potřeba 6-ti PWM modulů pro jeden modul displeje. Problém je, že ke každému z těchto modulů je nutno vést 4MHz hodinový signál.

Takto vysokou frekvenci rozvádět po celé desce lze považovat za velmi nevhodné a naprosto nemyslitelné by bylo za zdroj použít mikrokontrolér. Proto bylo navrženo řešení, kdy jsou vždy 3 PWM moduly seskupeny kolem 4MHz oscilátoru, ze kterého je signál veden do invertoru. Ten pak distribuuje hodinový signál, nejkratší cestou a bez ostrých zlomů, přímo k PWM modulům.

Největší *nedostatek PWM modulů* je signál `BLANK`, který provádí nulování čítače PWM. Pokud je signál v logické 1, čítač je nulován. Problém je, že tento čítač nefunguje kontinuálně, ale musí být externě nulován tímto signálem. Nabízí se dvě různá řešení tohoto problému.

Buďto přidání externího čítače (další speciální hardware), který bude připojen na stejný hodinový signál, jako PWM moduly. Tento čítač by po přetečení vyslal signál `BLANK`.

Duhá možnost je signál rozvést z mikrokontroléru a čítač nulovat při přetečení čítače s odlišným oscilátorem.

Po krátkých úvahách bylo jednohlasně rozhodnuto přiklonit se k variantě rozvedení signálu z mikrokontroléru.

Vzhledem k většímu množství PWM modulů může vzniknout problém příliš vysokého nároku na *rychlost SPI sběrnice*.

$f_{SPI} = \mathrm{barevna_h loubka} * \mathrm{pocet_k analu} * \mathrm{pocet_m odulu} * \mathrm{f_p repinani_r adku} f_{SPI} = 12 * 16 * 6 * 200 = 230\,400\,\mathrm{Hz}$

Výsledek výpočtu dává potřebnou frekvenci 230,4 KHz, což je bez problémů dosažitelná frekvence.

Maximální výstupní proud kždého kanálu se nastavuje referenčním odporem R_{IREF}. Jeho hodnota je dána vztahem

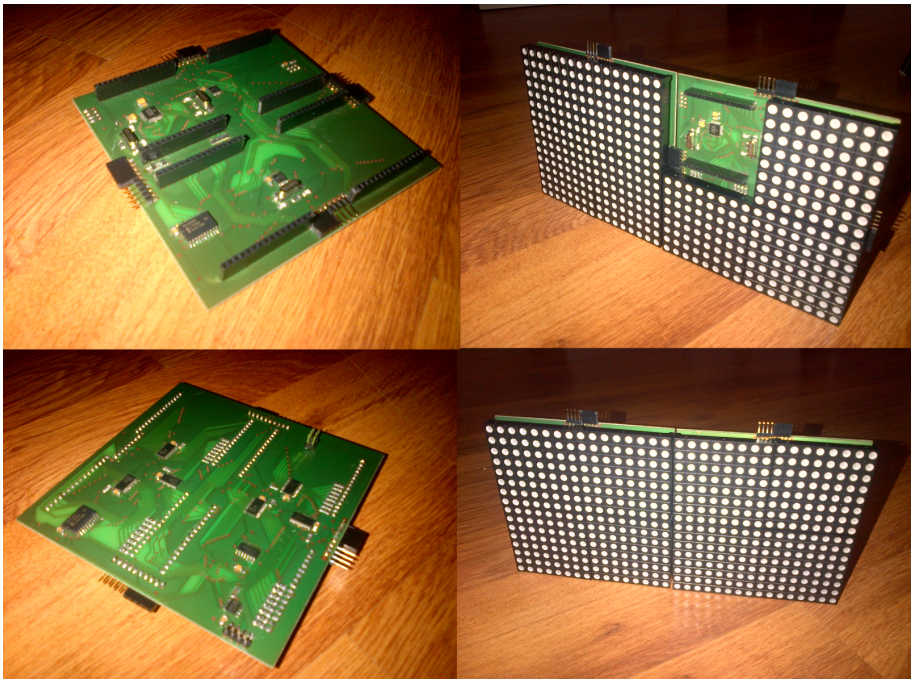
$$R_{IREF} = 42,5 * \frac{V_{IREF}}{I_{OLCMax}}$$

Kde $V_{IREF} = 1.20V$.

Výsledkem tedy je, že referenční odpor by měl mít hodnotu 2,55KΩ, k čemuž nejbližší hodnota, která je dostupná, je 2,61KΩ.

Informace získány z datasheetu TLC5946 [12]

Výsledný vzhled modulů



Obrázek 3.3: Výsledný vzhled modulů

3.3.2 Firmware modulu

Po zapnutí má modul 2 úkoly:

- Nainicializovat PWM moduly.
- Očekávat broadcastovou zprávu na zjištění tvaru displeje.

Inicializace PWM modulů

Obnáší nastavení DC (Dot Correction). Tato hodnota upravuje maximální proud každého kanálu. Může nabývat hodnot od 0 (0%) do 63 (100%). Výstupní proud je dán vztahem

$$I_{OUTn} = I_{OLCMax} * \frac{DCn}{63}$$

kde I_{OLCMax} je maximální výstupní proud kanálu.

Data do DC registrů jsou zasílána po stejné lince (`SIN`) jako data pro hodnoty PWM. Výběr registrů je dán signálem `MODE`. Pokud je v log. 1, jsou zpřístupněny registry DC, jinak jsou data zapisována do PWM registru.

Data začínají být validní na vzestupnou hranu signálu `XLAT`. To platí při zápisu do DC i PWM registru.

Přijetí požadavku na zjištění tvaru displeje

Na tuto zprávu se snaží modul odpovědět zprávou se svou adresou. Tato adresa je generována při nahrávání firmwaru a každý modul by měl mít své vlastní, unikátní ID. Arbitraci vyhrává modul s nejnižším ID.

- Pokud arbitraci vyhraje, čeká na zprávu od driveru s požadavkem na zaktivování některého sousedního modulu. Po přijetí této zprávy nastaví příslušný GPIO na logickou 1 a očekává odpověď na příslušném pinu. Pokud odpověď nedostane do 5ms, odešle po CAN sběrnici zprávu „ERROR“.
- Pokud arbitraci nevyhraje, čeká na zaktivování. Na zaktivování reaguje nastavením logické 1 příslušného GPIO a na CAN sběrnici odešle zprávu „OK“. Zároveň čeká, jestli po sběrnici nepřijde zpráva adresovaná jemu s požadavkem na zaktivování některého ze svých sousedních modulů. V tom případě postupuje podle bodu 3.3.2.

Přijímání a zobrazování dat

Při samotném zobrazování dat jsou na sběrnici CAN (dále jen sběrnice) důležité 2 zprávy.

- *Paleta barev* – zpráva, obsahující paletu barev. Tato zpráva je adresována všem modulům na sběrnici.
- *Adresa modulu*, pro který má driver připravena data. Na tuto zprávu modul reaguje zasíláním rámců *žádost o data*, dokud nedostane všechna obrazová data, která potřebuje.

Literatura

[1] Advanced RISC Machines Ltd.: ARM7DI Datasheet. http://infocenter.arm.com/help/topic/com.arm.doc.ddi0027d/DDI0027D_7di_ds.pdf, 1994-12 [cit. 2015-05-18].

[2] Atmel Corporation: tinyAVR Microcontrollers - White Paper. fy.chalmers.se/~jerzy/_WEBfiler/tinyAVR_whitepaper.pdf, 2008-02 [cit. 2015-05-18].

[3] Dubost, K.: Draft - GIF or PNG. http://www.w3.org/QA/Tips/png-gif, 2003-08-20 [cit. 2015-05-13].

[4] Ing. Michal Bidlo, P.: Synchronní sériová rozhraní: SPI, IIC. https://www.fit.vutbr.cz/study/courses/BMS/private/bms0x7.pdf, 2010-11-10 [cit. 2015-05-16].

[5] Ing. Petr Matoušek, P.: Architektura sítí, adresování, konfigurace TCP/IP. https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ISA-IT/texts/kapitola1.p 2011-10-10 [cit. 2015-05-17].

[6] LM Technologies Ltd.: AT Command Manual. lm-technologies.com/.../AT%20Command%20Manual/AT_Command_ Manual.pdf, 2013 [cit. 2015-05-20].

[7] MOTOROLA LTD.: Bosch Controller Area Network (CAN) Version 2.0. http://www.freescale.com/files/microcontrollers/doc/data_sheet/BCANPSV2.pdf, 1998 [cit. 2015-05-17].

[8] MURRAY, J. D.; Vanryper, W.: *Encyklopedie grafických formátů*. Computer Press, 1995, iSBN-80-85896-18-4.

[9] Natale, M. D.: Understanding and using the Controller Area Network. http://www-inst.eecs.berkeley.edu/~ee249/fa08/Lectures/handout_canbus2.pdf, 2008-10-30 [cit. 2015-05-17].

[10] Patrick, L.: GraphApp. http://enchantia.com/graphapp/index.html, [cit. 2015-05-13].

[11] Schwarz, J.: Vestavné systémy - Úvod. https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IMP-IT/lectures/P1/3-201 2014-9-24 [cit. 2015-05-18].

[12] Texas Instruments Incorporated: TLC5946 - datasheet. http://www.farnell.com/datasheets/1854767.pdf, 2008 [cit. 2015-05-20].

[13] WWW stránky: ARM - The Architecture For The Digital World. http://www.arm.com/products/processors/cortex-m/index.php.

[14] WWW stránky: Barco Indoor displays. http://www.barco.com/en/Products-Solutions/LED-displays/Indoor-LED-displays.

[15] WWW stránky: Bluetooth technology. http://www.bluetooth.com/Pages/Fast-Facts.aspx.

[16] WWW stránky: Bluetooth tutorial. http://www.tutorial-reports.com/wireless/bluetooth/tutorial.php.

[17] WWW stránky: Farnell element14 Ceská republika - Distributor elektronických soucástek. http://cz.farnell.com/.

[18] WWW stránky: Gold Office. http://www.goldoffice.cz/.

[19] Šimek, I. V.: Analysis of the communication over USB. https://wis.fit.vutbr.cz/FIT/st/course-files-st.php /course/IPZ-IT/pclabs/IPZe%20-%20USB.pdf, 2013-05-03 [cit. 2015-05-17].

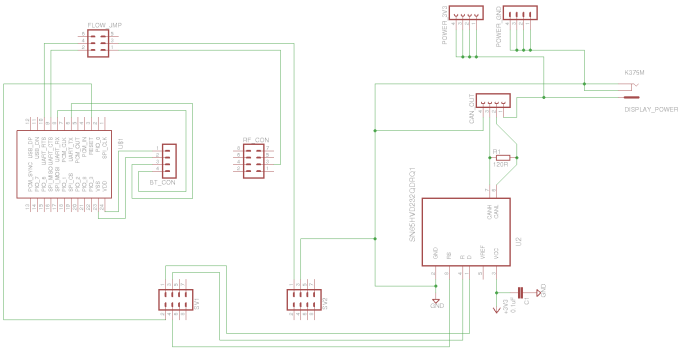
Příloha A

Obsah CD

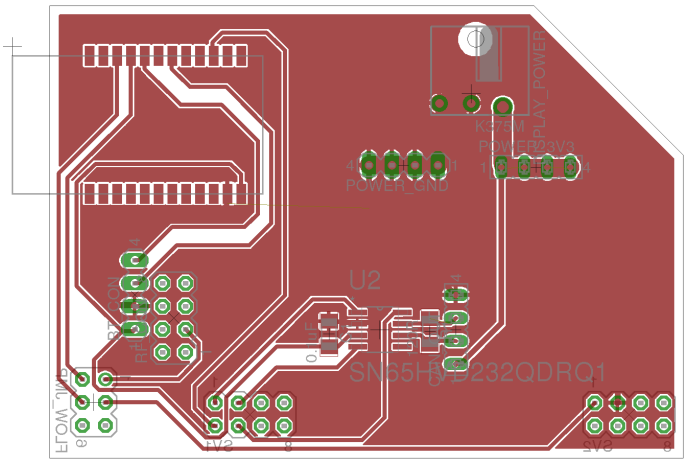
- Zdrojové kódy k uživatelské aplikaci pro počítač.
- Zdrojové kódy firmwaru driveru.
- Zdrojové kódy firmwaru modulu.
- Zdrojový text technické zprávy.
- Schéma zapojení modulu pro program Eagle.
- Schéma zapojení komunikačního shieldu driveru pro program Eagle.

Příloha B

Deska a schéma zapojení shieldu pro driver



Obrázek B.1: Schéma komunikačního shieldu pro driver



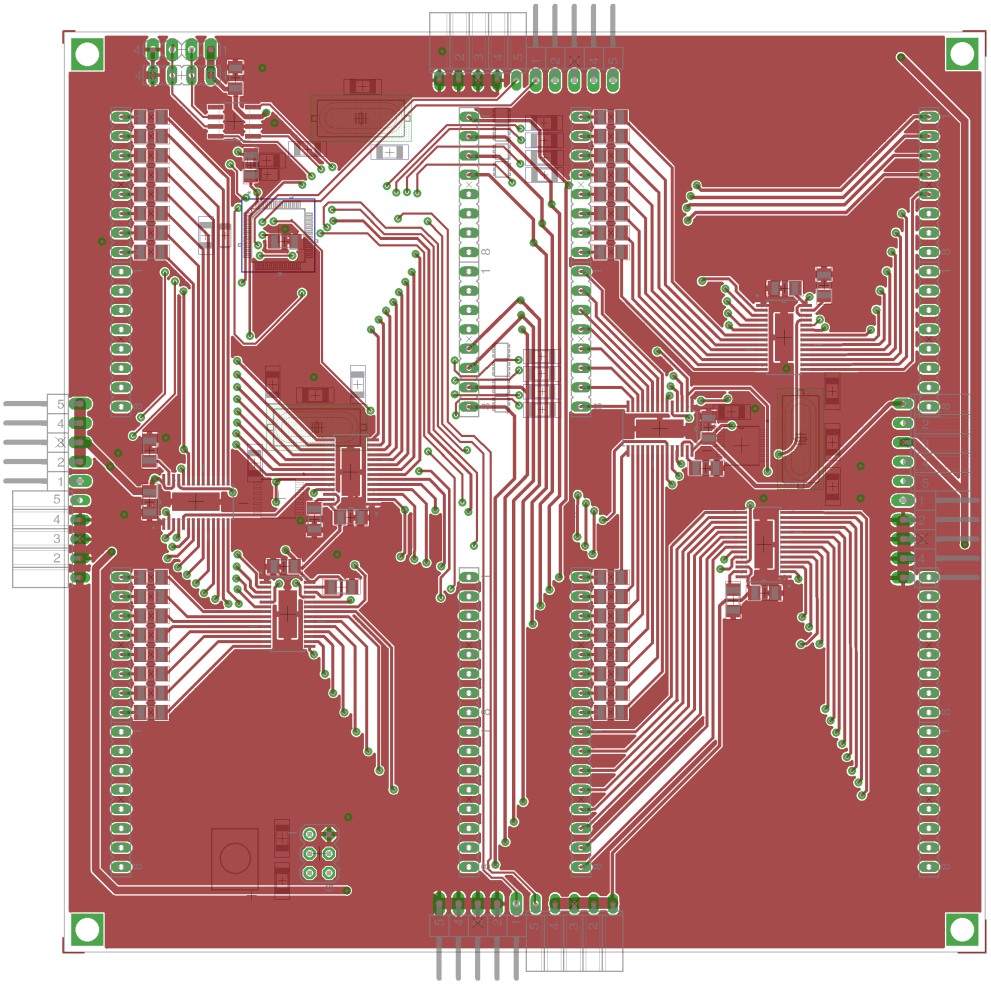
Obrázek B.2: Deska komunikačního shieldu pro driver

Tabulka B.1: Seznam součástek pro shield driveru

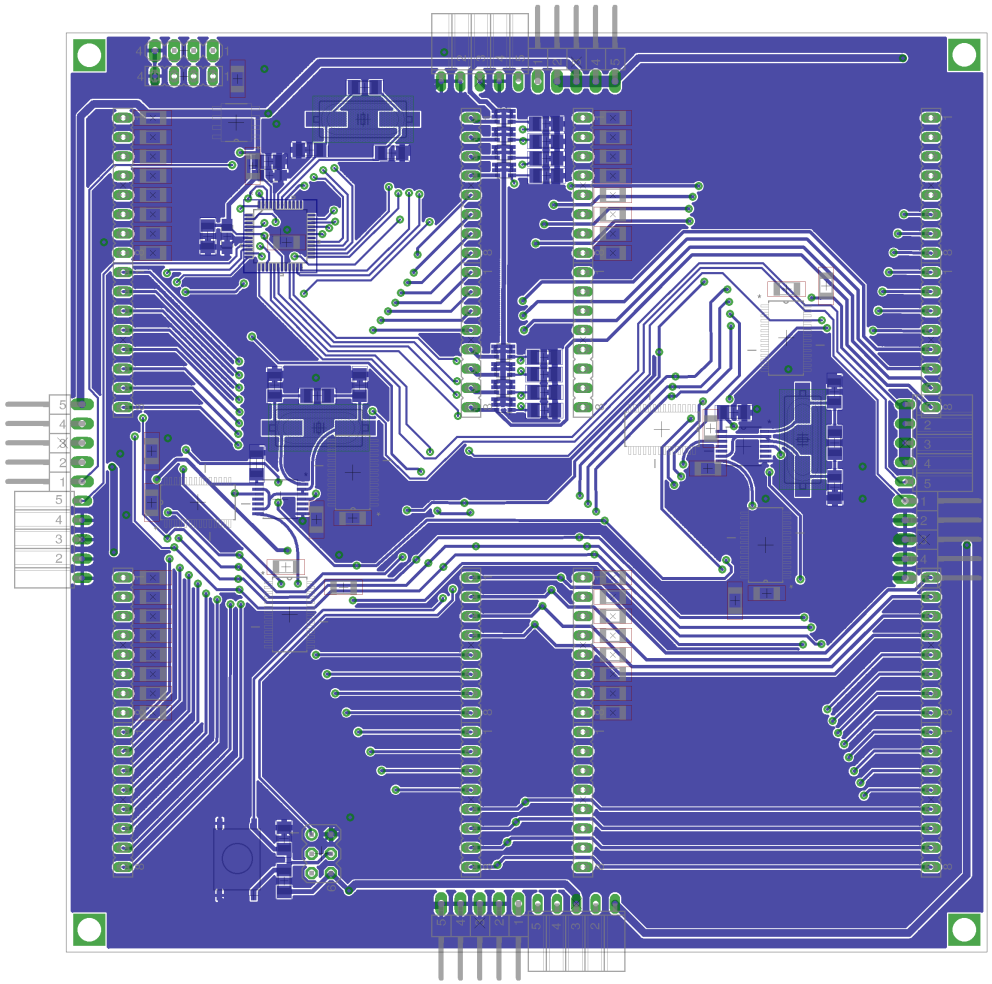
Název	Hodnota	Zařízení	Pouzdro	Počet
Pin Header	4x1	MA04-1	MA04-1	2
Pin Header	4x2	MA04-2	MA04-2	3
Pin Header	3x2	MA03-2	MA03-2	1
Female Header	4x1	FE04-1	FE04-1	2
Resistor	120Ω	R-EU_R1206	R1206	1
Capacitor	0.1μF	C-EUC1206	C1206	1
Power Jack		K375M	K375M	1
Bluetooth Module		LM780-0223	LM780	1
Can Transciever		SN65HVD232QDRQ1	SOIC-8	1

Příloha C

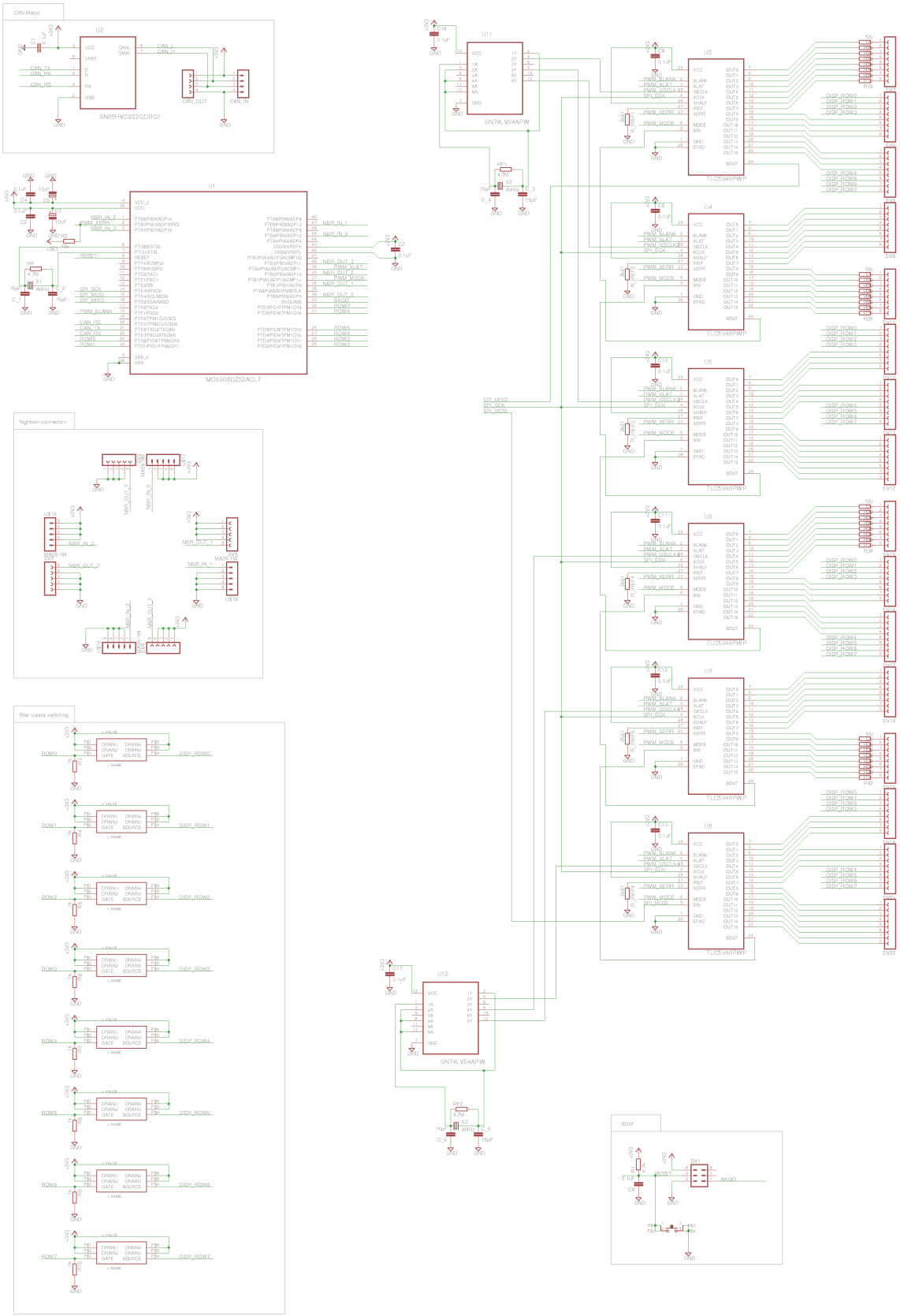
Deska a schéma zapojení modulu



Obrázek C.1: Vrchní strana desky modulu



Obrázek C.2: Spodní strana desky modulu



Obrázek C.3: Schéma modulu

Tabulka C.1: Seznam součástek pro modul

Název	Hodnota	Zařízení	Pouzdro	Počet
Pin Header	4x1	MA04-1	MA04-1	1
Pin Header	3x2	MA03-2	MA03-2	1
Pin Header (90°)	5x1W	MA05-1W	MA05-1W	4
Female Header	4x1	FE4-1	FE04-1	1
Female Header	16x1	FE16-1	FE16	8
Female Header (90°)	5x1W	FE05W	FE05W	4
Microswitch			TD-03XG SMD	1
Tanthal Capacitor	10μF	CPOL-EUSMCA	SMC_A	2
Capacitor	0.1μF	C-EUC1206	C1206	13
Capacitor	15pF	C-EUC1206	C1206	6
Capacitor	0.1μF	C-EUC1206	C1206	1
Resistor	120Ω	R-EU_R1206	R1206	1
Resistor	4.7MΩ	R-EU_R1206	R1206	3
Resistor	10kΩ	R-EU_R1206	R1206	1
Resistor	4.7kΩ	R-EU_R1206	R1206	1
Resistor	2.61kΩ	R-EU_R1206	R1206	6
Resistor	1kΩ	R-EU_R1206	R1206	8
Resistor	55Ω	R-EU_R1206	R1206	32
Crystal	4MHz	CRYSTALHC49UP	HC49UP	3
P-Channel MOSFET		SIA421DJ-T1-GE3	SC-70	8
Microcontroller		MC9S08DZ32ACLF	QFP50P900X900X160-48N	1
Can Transciever		SN65HVD232QDRQ1	SOIC-8	1
PWM Driver		TLC5946PWP	SOP65P640X120-29N	6
Inverter		SN74LV04APW	SOP65P640X120-14N	2