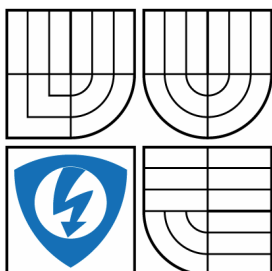


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODULÁRNÍ RGB LED DISPLEJ

MODULAR RGB LED DISPLAY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MIROSLAV COUFAL

VEDOUCÍ PRÁCE

SUPERVISOR

doc.Ing. ZDENĚK BRADÁČ, Ph.D.

BRNO 2013



VYSOKÉ UCENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Miroslav Coufal

ID: 74527

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TĚMATU:

Modulární RGB LED displej

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte modulární panel RGB LED, který umožní zobrazovat data zasilané z centra přes rozhraní Ethernet. Přístroj vybavte mikrokontrolérem, nebytnými budiči RGB LED, zobrazovací LED maticí, rozhraním Ethernet a příslušným programovým vybavením.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6
Dle pokynů vedoucího práce.

Termín zadání: 11.2.2013

Termín odevzdání: 20.5.2013

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Cílem této diplomové práce byl návrh RGB LED displeje s rozhraním ethernet. Vytvořil jsem zobrazovací modul, řízený mikrokontrolérem Atmega 2560-16AU. Tyto moduly lze propojovat pomocí sériového standardu RS485. Připojení k ethernetu je realizováno pomocí zásuvného rozhraní, které využívá možností programovatelného modulu Rabbit RCM 3200. Návrh jsem zdokumentoval a navržené zařízení jsem otestoval.

Klíčová slova

Atmega, Rabbit, RGB LED, displej, ethernet, RS485.

Abstract

The aim of this master's thesis was the design RGB LED display with Ethernet interface. I created a display module, controlled by a microcontroller ATmega 2560-16AU. These modules can be connected via a serial RS485 standard. Ethernet connection is made via plug-in interface that uses the programmable module Rabbit RCM 3200. I documented a proposal. I tested designed device.

Keywords

Atmega, Rabbit, RGB LED, display, ethernet, RS485.

Bibliografická citace:

COUFAL, M. MODULÁRNÍ RGB LED displej. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 57s. Vedoucí diplomové práce byl doc. Ing. Zdeněk Bradáč, CSc.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Modulární RGB LED displej jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **20. května 2013**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Zdeňku Bradáčovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Děkuji také Radomilu Filipovi za cenné rady a připomínky k praktické konstrukci a všem mým kolegům v zaměstnání, kteří na sebe vzali část mých povinností a umožnili mi tak sepsání této práce.

V Brně dne: **20. května 2013**

.....
podpis autora

Obsah

1	Úvod.....	9
2	Reprezentace barev ve výpočetní technice.....	10
3	RGB LED panely	12
3.1	Vlastnosti RGB LED panelů	12
3.2	Řízení RGB LED.....	12
3.3	Výrobci RGB LED panelů.....	14
4	Sběrnice RS 485 [17]	15
5	Ethernet	16
5.1	Ethernet a standard IEEE 802.3.....	16
5.2	ISO OSI model a protokoly TCP/IP	17
6	Výběr mikrokontroléru.....	19
6.1	Přehled.....	19
6.2	Programovatelný modul Rabbit RCM3200	22
6.2.1	Prostředky modulu RCM3200.....	22
6.2.2	Dynamic C.....	24
6.2.3	SBC BL2600	25
6.3	Mikroprocesor ATMEGA 2560-16AU	26
7	Návrh modulu RGB LED displeje	28
7.1	Koncepce zobrazovače a hardware navrženého modulu	28
7.1.1	Napájení	30
7.1.2	RGB LED matice a její buzení.....	31
7.1.3	Řízení LED (Atmega 2560-16AU)	34
7.1.4	Společné řízení modulů a komunikace s PC (Rabbit RCM 3200)	34
7.2	Softwarové vybavení	34
7.2.1	Program pro mikrokontrolér atmega 2560	35
7.2.1.1	PWM řízení LED.....	35
7.2.1.2	Komunikace pomocí RS485	37
7.2.1.3	Funkce main	39
7.2.2	Program pro modul Rabbit RCM3200	41
7.2.2.1	Funkce main	41
7.2.2.2	Obsluha rozesílání dat zobrazovacím modulům.....	42
7.2.2.3	Obsluha ethernetové komunikace.....	43
7.2.3	Program pro zasílání obrazu z PC	45

7.2.3.1	Funkce programu a jeho ovládání.....	45
7.2.3.2	Načtení barev obrazu	46
7.2.3.3	Komunikace po síti ethernet	48
8	Závěr.....	50

1 ÚVOD

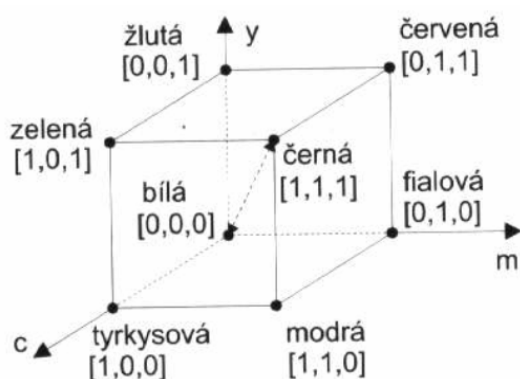
Ať chceme nebo ne, naše společnost je stále více zaměřena na ekonomické výsledky a měřítkem jsou často peníze. Silnou zbraní v konkurenčním boji může být i reklama. Jedním z účinných reklamních prostředků jsou i LED zobrazovače a informační tabule.

Cílem této práce je navrhnout barevný displej, který za pomoci RGB LED bude schopen zobrazit zadaná data. Podmínkou je modularita tohoto displeje a jeho připojitelnost na ethernet. Důležitým kritériem jsou i nízké náklady. V následujících kapitolách popisují návrh a sestavení modulu, který může být jedním z řešení těchto úkolů. Zabývám se krátce reprezentací barev ve výpočetní technice a možnostmi řízení RGB LED. Stručně představuji použité komunikační standardy RS485 a ethernet. Zejména se ale věnuji samotnému návrhu praktického zapojení a sestavení software zobrazovacího modulu a interface pro připojení k ethernetu.

2 REPREZENTACE BAREV VE VÝPOČETNÍ TECHNICE

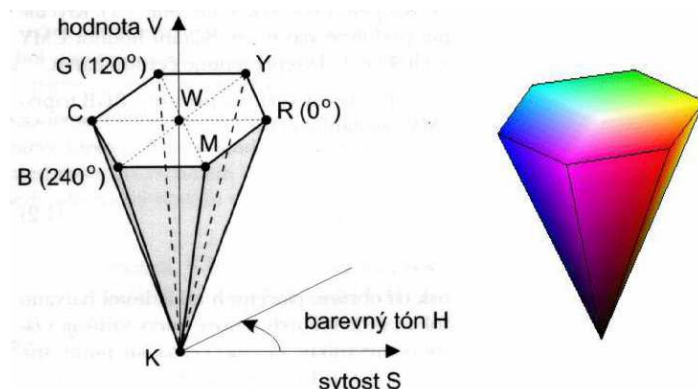
Pro popis a vytváření barev existuje mnoho barevných modelů více či méně vyhovujících danému odvětví technického oboru či grafiky. Nejznámější z nich jsou RGB, CMYK, HSV, HLS.

CMY (obr.1) je subtraktivní model. To znamená, že složením základních složek (C-azurová, M-purpurová, Y-žlutá) by v ideálním případě vznikla černá barva. V praxi je ale nutné černou barvu použít zvlášť, protože uměle vytvořená barviva zcela černou barvu nevytvoří. Používá se tedy spíše model CMYK. Tento model je používán zejména při tisku.



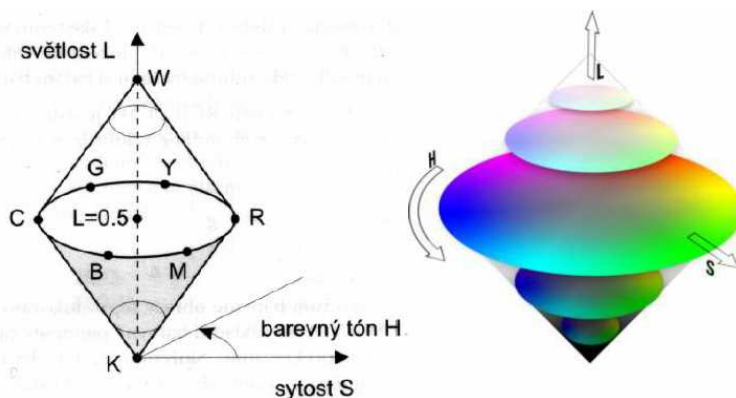
Obr. 1 – Odečet barev v modelu CMY (převzato z [7])

Model HSV je jedním z vyjádření barev, které používá zejména počítačová grafika. Barva je popsána barevným tónem (Hue), sytostí (Saturation) a jasnem (Value). Rozložení barev v tomto modelu je často zobrazeno jako barevný kužel (obr. 2).



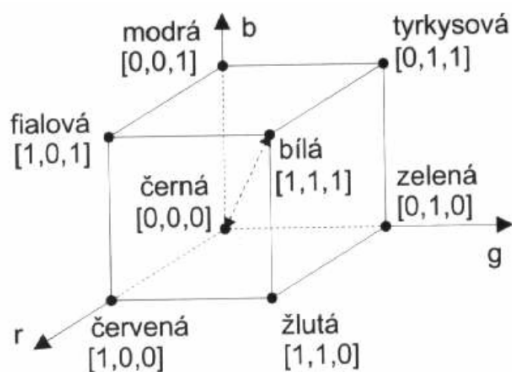
Obr. 2 – Reprezentace barev v modelu HSV (převzato z [7])

Barvy v modelu HLS jsou reprezentovány barevným tónem (Hue), světlostí (Lightness) a sytostí (Saturation), jak naznačuje obr.3.



Obr. 3 – Reprezentace barev v modelu HSV (převzato z [7])

Pro vyjádření barevných odstínů zejména u monitorů a zobrazovacích zařízení je využíván model RGB. Jde o aditivní model, kde jsou jednotlivé barvy vytvářeny skládáním základních složek R- rudá, G-zelená, B-modrá. Každá z těchto složek je ve výsledné barvě zastoupena 0-100 procenty své maximální hodnoty. Na obr.4, kde je znázorněno skládání složek v RGB modelu, odpovídá 100% hodnota 1. V počítačové grafice se někdy používá rozšířený model RGBA, kde složka $A(\alpha)$ odpovídá průhlednosti objektu. Pro uložení hodnot barev v RGB vyjádření je velmi často využíván 24 bitový prostor, kde pro každou ze základních složek je vyhrazeno 8 bitů. To je 256 odstínů jedné složky a 256^3 , tedy 16 777 216 možných barevných odstínů. Toto uspořádání je někdy označováno jako „true color“. Dalšími často používanými velikostmi paměťového prostoru pro ukládání barvy jsou 16, 32 nebo 48 bitů. 48-bitový mód je užíván při profesionální úpravě obrazu a pro každou ze základních složek je používáno 16 bitů. 32 bitové uspořádání je pouze doplněním „true color“ o 8 bitů pro vyjádření průhlednosti. 16 bitů může být mezi složky rozděleno rovnoměrně po pěti bitech, někdy je ale také vyjádřena zelená složka, na kterou lidské oko reaguje nejcitlivěji, pomocí šesti bitů. Vyjádření pomocí pěti bitů jsem z technických důvodů zvolil pro manipulaci s RGB složkami navrhovaného panelu (viz. kapitola č. 7). Pro vyjádření barev ve videotechnice a televizní technice jsou používány například modely YUV (přenos signálu v normě PAL), YIQ (NTSC), $Y_C B_C R_C$ (SECAM, JPEG, MPEG) a další [7].



Obr. 4 – Skládání barevných složek v modelu RGB (převzato z [7])

3 RGB LED PANELY

RGB LED panely se používají především jako reklamní zařízení jako jsou billboardy, reklamní návěští, zobrazovací kostky nebo mantinely na zimních stadionech, velkoplošné obrazovky při velkých společenských událostech apod.. Lze je použít i v interiéru. Jejich vysoká cena však je předurčuje k použití tam, kde klasická obrazovka nedosahuje žádaných rozměrů. Výhody RGB LED panelů jsou především ve velké svítivosti a tedy možnosti sledovat obraz na velkou vzdálenost i za denního světla, malé spotřebě elektrické energie a velmi dlouhé životnosti.

3.1 Vlastnosti RGB LED panelů

Výrobci nabízejí RGB LED panely většinou v modulárním provedení. Jednotlivé moduly lze pak propojovat a montovat do rámu tak, aby vytvořily obrazovku požadované velikosti. Počet LED připadající na modul je různý u různých výrobců i typů (například 16x32 nebo 32x32 ale i větší). Rovněž vzdálenost bodů (rozteč led) je různá. Závisí především na pohledové vzdálenosti. Vyrábějí se panely s roztečí pixelů 10, 12, 16, 18 mm, ale také s většími vzdálenostmi. Rozteč 10mm by přibližně odpovídala minimální vzdálenosti pozorovatele 10 m. Jednotlivé pixely jsou reprezentovány buď RGB LED diodami, velmi často v smd provedení nebo kombinací červené, zelené a modré LED, které na větší pohledovou vzdálenost opticky splývají v jeden barevný bod. Sada barevných LED bývá složena ze třech, čtyř nebo šesti prvků. Vhodné rozmístění LED může při správném řízení opticky zvětšit počet bodů [14]. Napájení je většinou provedeno 230VAC nebo 120VAC kvůli jednoduchému připojení a univerzálnosti modulu. Důležitým kritériem je, to jestli je obrazovka určena do interiéru nebo do venkovních prostor. Tomu musí být uzpůsoben materiál i krytí zařízení. Interiérové panely mají samozřejmě menší vzdálenost pixelů. Barevná hloubka pixelu u RGB panelů většinou bývá 24bitů (true color). Nicméně co se týká rozměrů, počtu pixelů nebo barevné hloubky není zaveden žádný závazný standard.

3.2 Řízení RGB LED

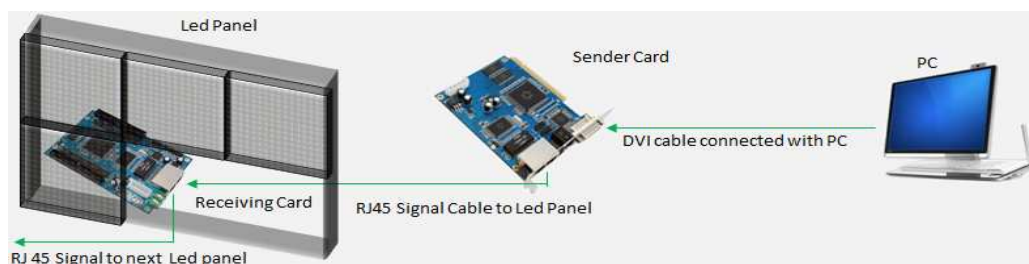
Jak jsem uvedl LED panely bývají sestaveny z diskrétních monochromatických LED nebo sestávají z RGB LED. RGB LED jsou vyráběny se společnou katodou či anodou nebo s vyvedenými všemi šesti vývody v široké škále provedení a vlastností. Obsahují v podstatě na jednom čipu tři diody o základních barvách Red, Green a Blue RGB modelu. Velikosti bývají stejné jako u jednobarevných LED. V provedení THT jsou většinou vyvedeny čtyři vývody, kdy jeden je společná katoda (anoda) a tři další jsou anody (katody) jednotlivých barevných světlo emitujících diod. V provedení smd jsou

většinou používána pouzdra PLCC4 se čtyřmi vývody nebo PLCC6 s vyvedením všech katod i anod.

Řízení barvy emitovaného světla je možné buď v určitém velmi malém rozmezí proudem R, G a B diodami nebo pomocí PWM (pulzně-šířková modulace) modulovaného signálu o dostatečné frekvenci přivedeného na tyto diody. Obnovovací frekvence celého modulu s RGB LED liodami by měla být minimálně 50 Hz, aby obraz rušivě neblikal. Základní barvy (respektive R,G,B diody) lze spínat buď tak, že všechny barvy svítí část periody odpovídající procentuelně zastoupení složky v požadované barvě nebo je možné periodu ještě rozdělit na tři části v poměru barevných složek a v každé z těchto částí sepnout jednu základní barvu. V navrženém zapojení lze provést obě varianty, pokud by ale mohla svítit pouze jedna dioda v rámci RGB LED v jednom okamžiku (např. při rozšíření výstupů pomocí demultiplexeru), museli bychom použít druhou zmiňovanou variantu. Tím by se ovšem zvýšily nároky na spínací frekvenci.

Řízení modulů, ze kterých je panel složen, spočívá v příjmu dat, obsahujících obraz a v zajištění PWM modulace proudu procházejícího jednotlivými barevnými LED. Distribuce obrazového signálu je zajištěna kontrolérem obsluhujícím více zobrazovacích modulů a zároveň přijímajícím obrazová data od nadřazeného systému. Pro příjem obrazu z nadřazeného zařízení (většinou PC) se v praxi používají dvě možné varianty[15].

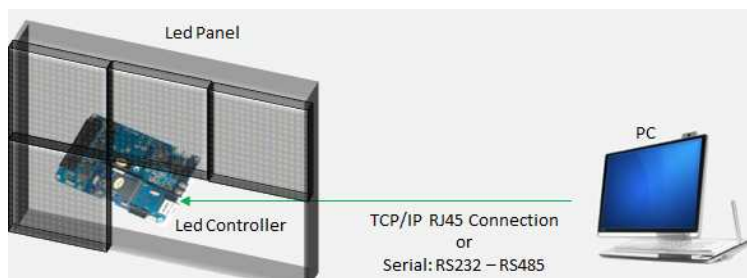
Synchronní řízení probíhá tak, že řídicí karta panelu je připojena k PC pomocí standardů jako jsou VGA (video graphics array), DVI (digital visual interface) nebo HDMI (high-definition multimedia interface) a zobrazuje obraz vysílaný grafickou kartou PC. Panel pak v podstatě funguje jako monitor. Přenos signálu může být ještě rozdělen na vysílač, který převádí video signál z PC grafické karty na formát, vhodný pro některý síťový standard (jako je např. ethernet) a na přijímač, který síťový signál zpracovává a předává řízení zobrazovacích modulů. Tento princip je naznačen na obr.5.



Obr.5 – princip synchronního řízení LED panelu (převzato z [15])

Asynchronní řízení (obr.6) obsahuje přijímací kartu, která je připojena přímo k PC pomocí ethernetu nebo sériové linky. Při tomto způsobu je nutný příslušný software, který převádí obraz na formát vhodný pro přenos do zobrazovacího panelu. Přijímací karta panelu často může

obsahovat také paměť pro uchování zadaných dat a panel tak může fungovat autonomně. Například po odpojení PC může přehrávat zadané reklamní spoty.



Obr.6 – princip asynchronního řízení LED panelu (převzato z [15])

3.3 Výrobci RGB LED panelů

LED technologie jsou v současné době trendem ve zobrazovací a osvětlovací technice a také výrobců a distributorů LED panelů je stále více. Z tuzemských výrobců zabývajících se výrobou Velkoplošných LED panelů uvedu například LEDbow, GEMA s.r.o., RTG-Tengler nebo VIDEOBILLBOARD s.r.o. . Kromě klasických pevných LED panelů jsou vyráběny i například ohebné LED závěsy či sítě nebo polopropustné panely založené na technologii LED, jež lze kupříkladu instalovat na prosklené budovy. Ke zobrazovačům nabízejí výrobci samozřejmě i obslužný software.

Nabídka firem zabývajících se LED reklamní technikou je velmi široká a nebylo by příliš smysluplné rozvádět příliš průzkum trhu. Jelikož se vzhledem k omezeným zdrojům zaměřím spíše na vývoj menšího levného modulu, uvedu pouze pro příklad panel z nabídky firmy Elvac a.s. . outdoor full color LED display - p10 (obr. 7, [16]). Tento modul je určen pro venkovní použití . Má napájení 220V+-15% a maximální odběr je 19,2W. LED jsou uspořádány ve trojicích 1R,1G a 1B v rastru 10mm. Je obsaženo 16x16 bodů, rozmery modulu jsou tedy 160x160 mm. Řízení led je provedeno tzv. ¼ scanováním (multiplexováním). Displej může zobrazit 16,7 mil. barev. Obnovovací frekvence je $\geq 180\text{Hz}$, obrazová frekvence je $\geq 60\text{Hz}$.

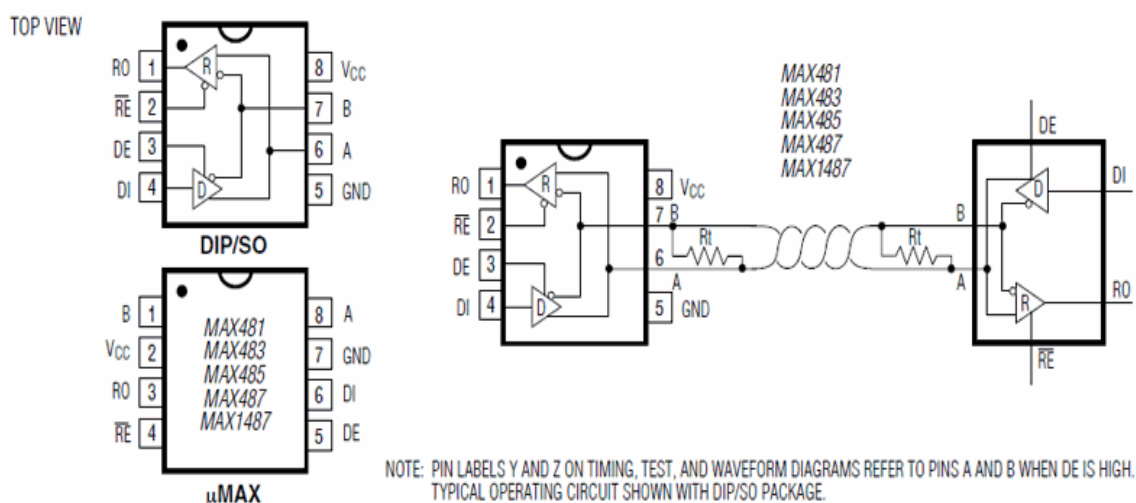


Obr.7 - Full color LED display - p10 z nabídky Elvac a.s. (převzato z [16])

4 SBĚRNICE RS 485 [17]

V této práci je využívána komunikace pomocí sběrnice RS 485. Sběrnice RS 485 pracuje na rozdíl od RS 232, ve které jsou signály reprezentovány napětím na signálovém vodiči proti zemi, s diferencí napětí na signálových vodičích A a B. Pokud je A zápornější a B kladnější je signál neaktivní, pokud je naopak A kladnější a B zápornější je signál aktivní. Takovéto zapojení je odolnější proti rušení a umožňuje mnohem větší délku propojení než RS 232 a vyšší komunikační rychlosti.

Signál je veden většinou krouceným párem vodičů a to buď jedním párem, kdy pro vysílání i příjem je používáno jedno vedení a je pak nutné použít třístavové budiče nebo dvěma páry, kdy druhý pár je použit ke komunikaci směrem k masteru. Délka vedení může být až 1200m a typické komunikační rychlosti jsou kolem 2,5MB/S. Na jedno vedení lze připojit až 32 vysílačů. Především při vyšších rychlostech a délkách vodičů je třeba zakončit vedení terminačním odporem, který zamezuje odrazům vlny na koncích vodičů.



Obr. 8 - Zapojení integrovaného obvodu MAX485

Při použití RS 485 u zařízení, která původně komunikovala po RS 232 je nutné přizpůsobit napěťové úrovně a pokud je použit 1 kroucený pár (většina případů), zajistit přepínání směru, respektive vysílání/příjem. Pro účely přizpůsobení jsou k dispozici integrované převodníky jako například max 485 (obr.8), max487 apod. . Přepínání zajišťuje řídicí software pomocí jednoho signálu nebo pokud není řídicí signál k dispozici, je třeba použít převodník s automatickým přepínáním.

5 ETHERNET

5.1 Ethernet a standard IEEE 802.3

Dalším použitým síťovým standardem je ethernet ve spojení s rodinou protokolů TCP/IP. V současné době stále více používaný pojem ethernet je původní označení pro sériovou komunikační sběrnici vyvinutou na konci sedmdesátých let firmou Xerox a později rozvinutou ve spolupráci s Intel a Digital Equipment Corporation. Název ethernet se později vžil i pro označování normy IEEE 802.3 CSMA/CD, která je spravovaná IEEE (Institute of Electrical and Electronics Engineers). Tato norma vznikla na základě zmíněné sítě ethernet a specifikuje například několik fyzických médií. Ethernet původně definoval jako fyzické médium pouze koaxiální kabel, rychlost byla 10Mb/s a topologie sběrnice. Dnes se velmi často užívá jako přenosové médium kabel UTP nebo STP obsahující kroucené páry nebo optické vlákno. Možný je samozřejmě již zmíněný koaxiální kabel, IrDa přenos nebo bezdrátové připojení pomocí wifi či bluetooth. V současné době jsou známy tyto typy ethernetu (převzato a upraveno z [19]):

- **10Base5** Původní Ethernet na koaxiálním kabelu o rychlosti 10 Mbit/s.
- **10Base2** Ethernet na tenkém koaxiálním kabelu o rychlosti 10 Mbit/s.
- **10Base-T** Jako přenosové médium používá kroucenou dvojlinku s rychlostí 10 Mbit/s. Využívá dva páry strukturované kabeláže ze čtyř.
- **10Base-F** Varianta s optickými vlákny o rychlosti 10 Mbit/s.
- **100Base-TX** Varianta s přenosovou rychlostí 100 Mbit/s, které se říká **Fast Ethernet**, používá dva páry UTP nebo STP kabelu kategorie 5.
- **100Base-FX** Fast Ethernet používající dvě optická vlákna.
- **1000Base-T** Ethernet s rychlostí 1000 Mbit/s, nazývaný **Gigabit Ethernet**. Využívá 4 páry UTP kabeláže kategorie 5e, je definován do vzdálenosti 100 metrů.
- **1000Base-SX** Gigabit Ethernet používající mnohavidové optické vlákno.
- **1000Base-LX** Gigabit Ethernet používající jednovidové optické vlákno.
- **10GBase-T** Ethernet s rychlostí 10 Gbit/s, nazývaný **Ten Gigabit Ethernet**.
- **40GBASE** a **100GBASE** s rychlostí 40 a 100 Gbps by měl používat optická vlákna; měděné kabely jen do délky několika desítek metrů.

5.2 ISO OSI model a protokoly TCP/IP

V modelu ISO (internacional standard organization) OSI (open system interconnection) by ethernet spadal do úrovně fyzické a linkové vrstvy [20]. Velmi často se ethernet používá se souborem protokolů TCP/IP. Tento velmi rozšířený standard předpokládá využití čtyřech vrstev. Jsou to:

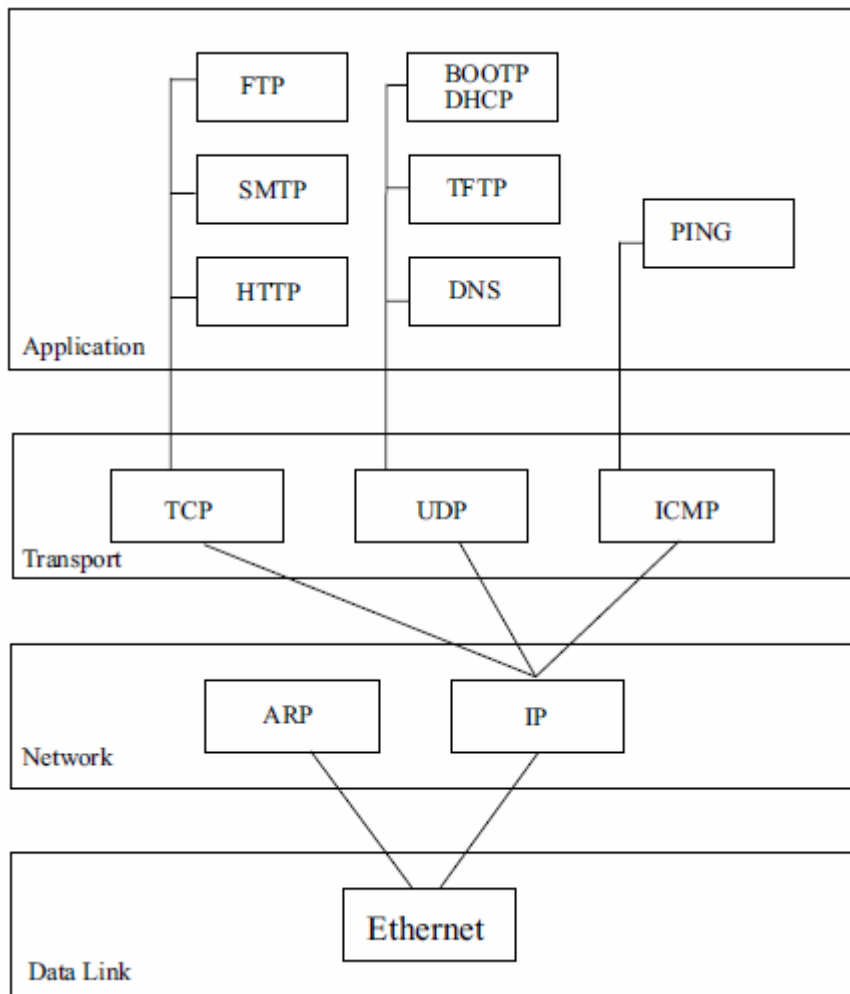
1. vrstva síťového rozhraní (odpovídá fyzické a linkové vrstvě ISO OSI)
2. síťová vrstva
3. transportní vrstva
4. aplikační vrstva (odpovídá vrstvám 5-7 ISO OSI)

Rozložení protokolů v jednotlivých vrstvách je částečně naznačeno na obr. 9. Pro adresování v síti je důležitý protokol IP (internet protokol), který na základě IP adresy směruje paket nebo datagram ze zdrojového zařízení do cílového zařízení (host). Tento protokol používá další protokoly síťové vrstvy jako ARP (adres resolution protocol). Původně se užíval pouze protokol IPv4. Masové rozšíření sítě internet ale přineslo potřebu adresovat víc zařízení, proto vznikl protokol IP verze 6 s rozšířeným adresováním (IPv6).

Za navázání spojení a doručení paketu nebo datagramu jsou zodpovědné protokoly transportní vrstvy. Jsou to protokoly TCP a UDP. UDP je nespojovanou službou, zajišťuje pouze odeslání datagramu a negarantuje jeho doručení. TCP je bezpečnější, ale pomalejší službou transportní vrstvy. Při využití této služby je navázáno spojení s hostem a je garantováno doručení odesílaných dat nebo oznámení o nedoručitelnosti. Protokol rovněž zabezpečuje znovuodeslání dat při zjištěném nedoručení. Důležitým protokolem transportní vrstvy je také ICMP (internet control message protocol). Tento protokol se používá k odesílání chybových zpráv.

Zvláštní postavení mají protokoly SLIP a PPP, které využívají IP adresace na úrovni síťové vrstvy, ale jsou využívány pro připojení pomocí sériové linky. ([<http://www.earchiv.cz/a96/a632k150.php3>])

Ve vyšších vrstvách existuje poměrně hodně protokolů, které nabízejí využití pro přenos souborů (FTP, TFTP), sdílení souborů (NFS), umožňují elektronickou poštu (SMTP, POP, IMAP), služby jako WWW (http, HTML..) apod..



Obr.9 – Rozložení protokolů rodiny TCP/IP ve vrstvách (převzato z [5])

6 VÝBĚR MIKROKONTROLÉRU

6.1 Přehled

Pro připojení aplikace k ethernetu je možné zvolit buď cestu, kdy zvolený mikrokontrolér komunikuje standardními způsoby s ethernetovým řadičem nebo přímo použít mikrokontrolér, který ethernetový řadič obsahuje. Ve druhém případě se většinou jedná pouze o integrovanou MAC vrstvu a aplikaci je třeba doplnit ještě o fyzickou vrstvu (PHY). Rozhraní mezi těmito dvěma vrstvami je pak definováno protokoly MII (media independent interface) nebo RMII (reduced media independent interface). Integrovaný ethernetový řadič mají například rodiny mikroprocesorů PIC32MX6xx nebo PIC32MX7xx od firmy Microchip, některé typy mikroprocesorů v řadách STM32F1, STM32F2 a STM32F4 firmy ST-Microelectronics atd. .

Další zmíněnou možností připojení k ethernetu je využití některého ethernetového řadiče. Tyto řadiče jsou k mikrokontroléru připojeny buď paralelně nebo pomocí UART, SPI či některého dalšího sériového rozhraní. Je možné využít například tyto obvody (převzato z [13]):

Realtek RTL8019 – paralelní, 10Base-T

Crystal LAN CS8900A – paralelní (ISA), 10Base-T,

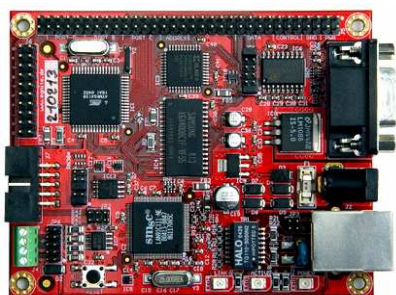
SMCS LAN91C111 – paralelní, 10Base-T/100Base-TX

Microchip ENC28J60 – SPI, 10Base-T,

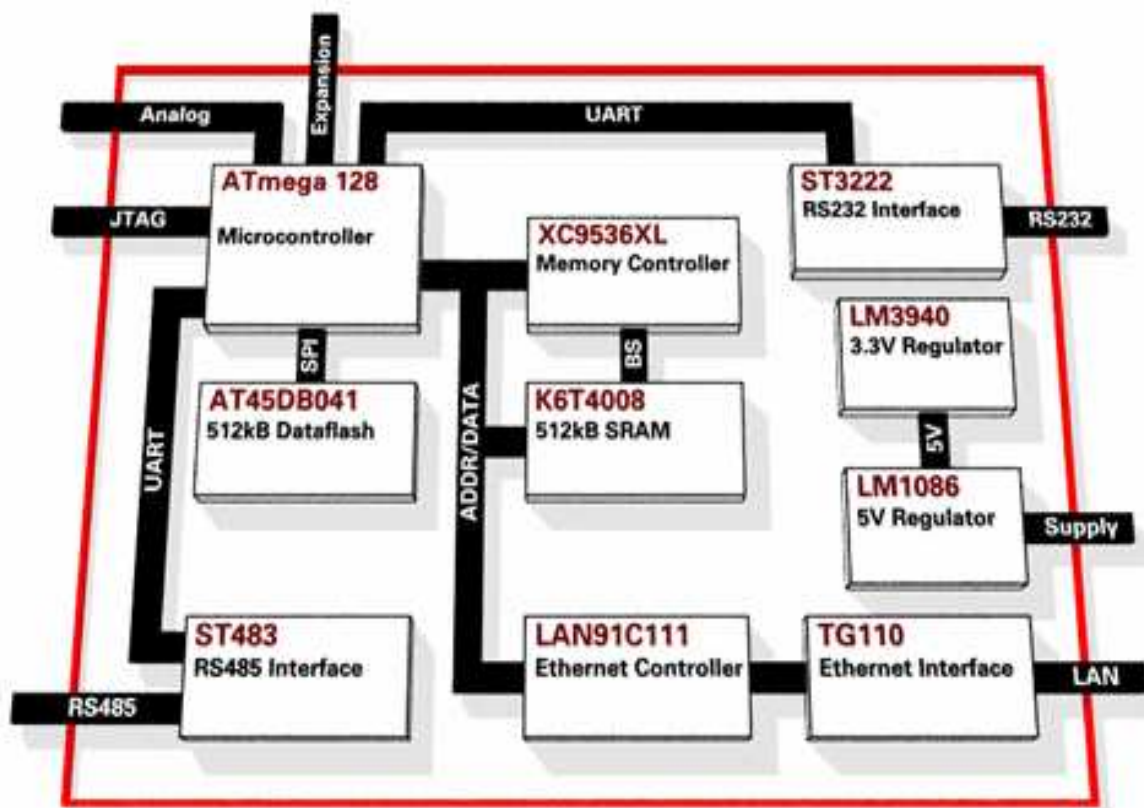
Connect One Nano SocketLAN – UART/SPI/USB, 10Base-T/100Base-TX.

Výsledné zařízení musí samozřejmě ještě obsahovat konektor RJ45 s oddělovacími transformátory a rezistorem, chránícím obvod proti vysokému napětí. Je také možné použít tzv. magjack, který již tyto součástky obsahuje.

Existují i hotová nebo předzpracovaná řešení. Příkladem je projekt ethernut, který v GNU GPL licenci nabízí využití kooperativního operačního systému nut/OS, rozhraní nut/net, ale třeba i hotovou vývojovou desku (např. Ethernut 2.1 – obsahuje ATmega128, 100 MBit Eth, 512 kb SRAM, 512 kb SPI FLASH – obr.10 a 11). Kompatibilní s ethernut je například projekt charon.



Obr. 10 – modul Ethernut 2.1 (převzato z [21])

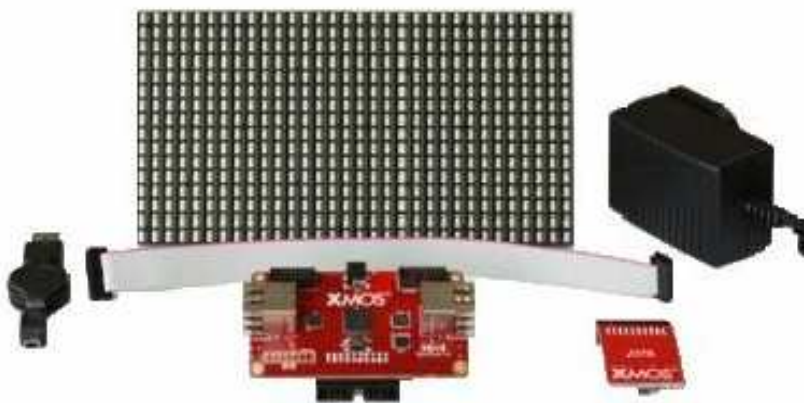


Obr. 11 – Blokové schéma modulu Ethernet 2.1 (převzato z [21])

Potřebný hardware, ale také softwarovou podporu v podobě knihoven pro programovací jazyky, jako jsou C nebo Pascal, většinou nabízejí vývojové kity a vývojové desky které nabízejí především sami výrobci mikroprocesorů. Při vývoji nových aplikací je použití těchto výrobků nejjednodušší a nejrychlejší cestou, proto se dále zaměřím především na ně.

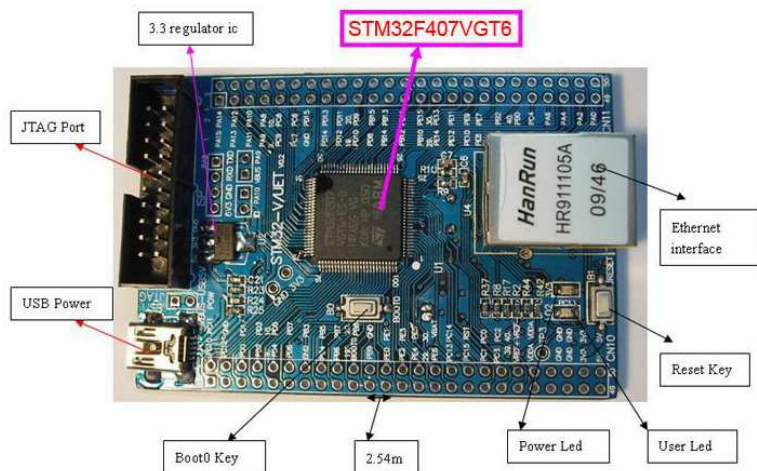
Pro vývoj displejů a panelů s RGB LED je například přímo určen vývojový kit XC-3 od firmy XMOS (obr. 12). Tento kit obsahuje základní desku XC-3, modul s 16x32 polem LED RGB diod, 16-ti pinový propojovací kabel a XTAG konektor s USB kabelem. Tento výrobek u nás distribuovala firma MES Praha. Bohužel však již není v jejím sortimentu.

Cenově velmi přijatelná je vývojová deska od firmy ST-Microelectronic STM32F4 Discovery, která je založena na výkonném mikrokontroléru STM32F407GVT6 s jádrem cortex M4. Tento mikrokontrolér by měl mimo jiné obsahovat i podporu průmyslového ethernet standardu IEEE 1588v2. Cena tohoto modulu je v současné době zhruba 300Kč.



Obr.12 - Vývojový kit XC-3 firmy XMOS (převzato z [8])

Podobný produkt založený také na mikroprocesoru STM32F407GVT6 je UET STM32F407GVT6 100-pin core board. Tento modul je napájen přes USB, obsahuje JTAG připojení pro ladění a také konektor RJ45 pro připojení ethernetu. Cena tohoto modulu je v současné době cca 73 USD. Vyobrazení této desky s popisky je na obr. 13.



Obr. 13 – UET STM32F407GVT6 100-pin core board – (převzato z [9])

Pro použití v této práci byl vybrán především kvůli dostupnosti a velmi obsáhlé podpoře a kvalitní dokumentaci programovatelný modul firmy Digi internacional RCM 3200. Tento vývojový modul je původně produktem firmy Rabbit Semiconductors, kterou však v roce 2006 odkoupila společnost Digi Internacional Inc.. Tato společnost pokračuje v podpoře projektů Rabbit a jejich dalším vývoji . Možnosti modulu jsou popsány v další kapitole.

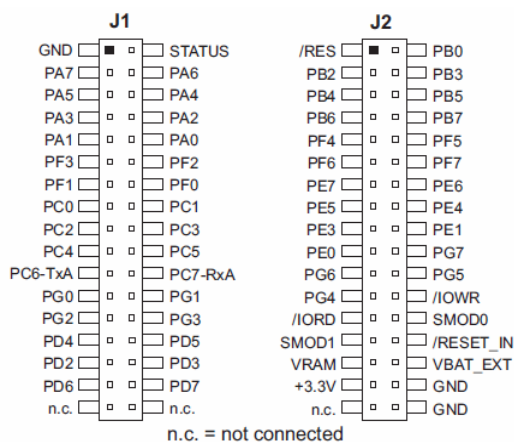


Obr.14 – vývojový modul RCM3200 (převzato z [1])

6.2 Programovatelný modul Rabbit RCM3200

6.2.1 Prostředky modulu RCM3200

Modul RCM 3200 (obr. 14) je umístěn na desce o rozměrech 47 mm x 69 mm a jeho výška je 22mm. Propojení s hardware vyvíjené aplikace je možné pomocí dvou konektorů o 2x17 pinech s poněkud nestandardní roztečí 2 mm. Popis zapojení konektorů je znázorněn na obr.15.

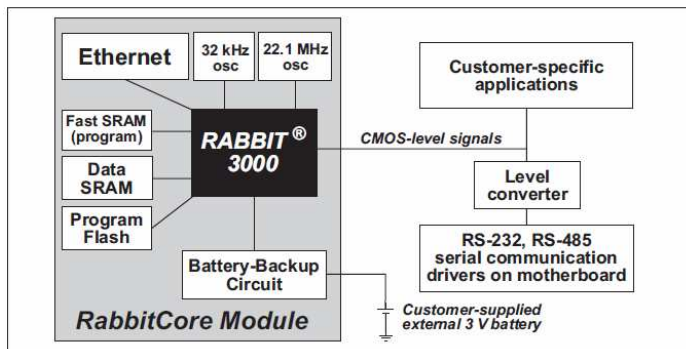


Note: These pinouts are as seen on the Bottom Side of the module.

Obr.15 – schéma konektorů pro připojení programovatelného modulu RCM3200 (převzato z [1])

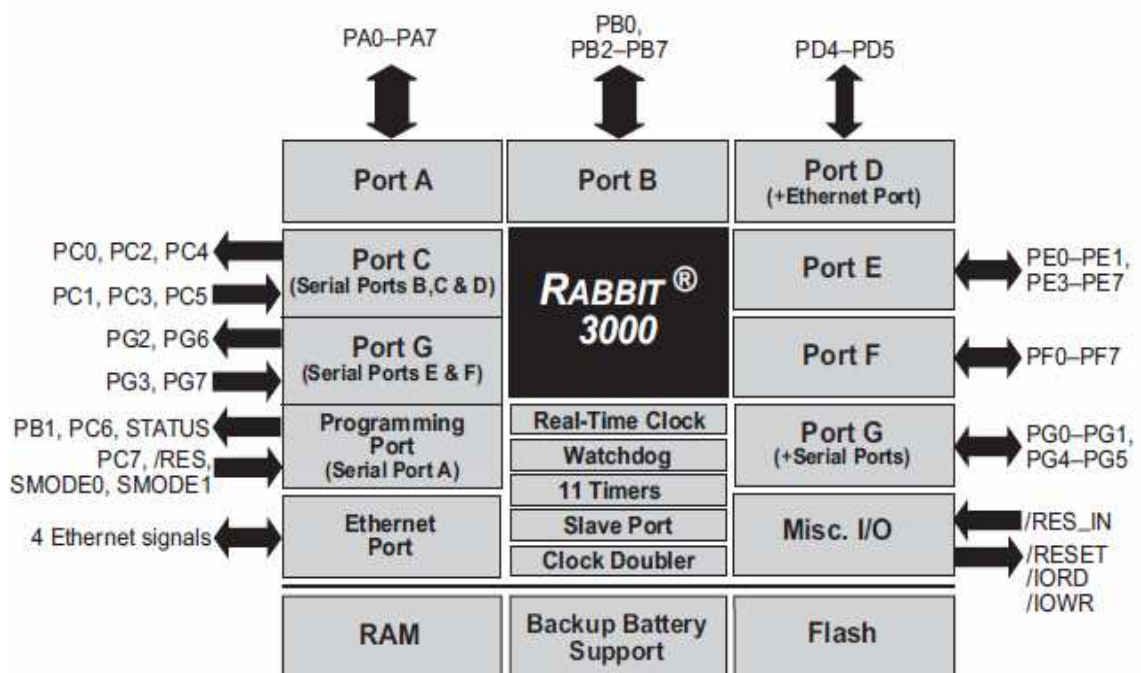
RCM3200 obsahuje mikroprocesor Rabbit 3000 pracující na frekvenci 44,2 MHz. Na desce je osazen ethernetový řadič Realtek 8019AS, který umožňuje připojení k síti ethernet. Činnost řadiče je indikována třemi LED (LNK – indikuje ethernetové připojení, ACT – indikuje aktivitu na lince a SPD – indikuje 100base-T síť). Dále je tu

512 KB flash paměti, 512KB paměti programu typu SRAM a 256 KB SRAM datové paměti. Blokové schéma desky RCM 3200 je znázorněno na obr.16.



Obr. 16 – hlavní bloky desky Rabbit RCM 3200 (převzato z [1])

Je možné využít 52 vstupně/ výstupních linek na sedmi portech. 44 z nich je konfigurovatelných jako vstup nebo výstup, 4 jsou pevně vstupy a 4 pouze výstupy. Připojení zařízení s CMOS technologií je bezproblémové a vstupy jsou tolerantní i vůči připojení 5V logiky, přestože napájení mikroprocesoru je 3,3V. Možné využití portů je naznačeno na obr. 5. Procesor je vybaven externím resetovacím vstupem. Samozřejmostí je využití některých I/O linek pro sériovou komunikaci. Čtyři porty jsou konfigurovatelné pro SPI sériovou komunikaci, dva pro SDLC/HDLC. Na desce nejsou bohužel přímo implementovány transceivery pro RS232 a RS485, ale jsou například na prototypové desce, která je součástí vývojového kitu . RCM 3200 obsahuje také hodiny reálného času, zálohované externí baterií.



Obr. 17 – využití I/O portů mikrokontroléru Rabbit 3000 – Převzato z [1]

Při programování Rabbit 3000 můžeme využít deset osmibitových časovačů a jeden desetibitový časovač, desetibitový PWM čítač se čtyřmi registry, dvoukanálový dekodér pro kvadrurní vstupy. RCM3200 také podporuje 1,5 Mb/s IrDA transceiver.

6.2.2 Dynamic C

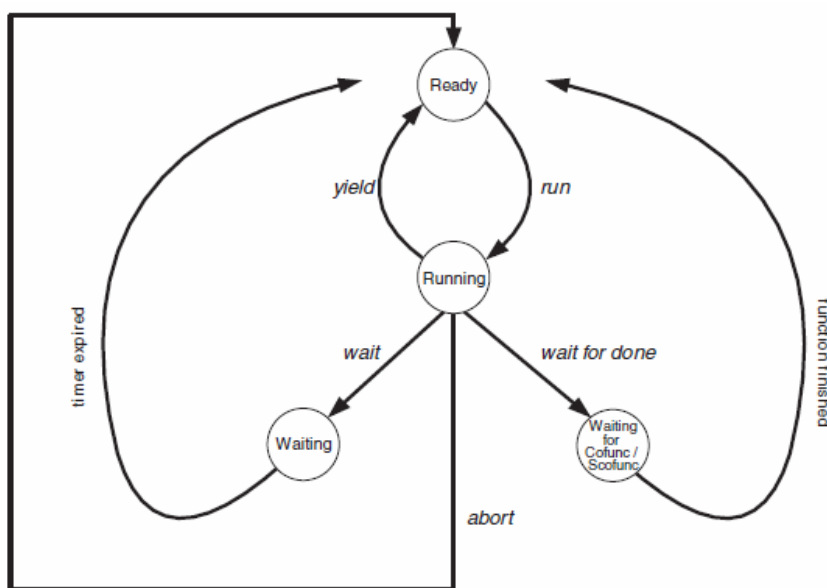
Ke programování mikroprocesorů Rabbit řad 2000 - 6000 a ladění aplikací je určen softwarový nástroj Dynamic C. Tento software je zdarma možné stáhnout ze stránek www.digi.com. Digi Internacional poskytuje v rámci tohoto programu také příklady aplikací a knihovny pro podporu vývojových desek. Velmi dobrá je i několikadílná podrobná dokumentace vysvětlující například požití TCP/IP stacku pro embeded systémy apod. .

Jednou z velkých předností dynamic C je podpora multitaskingu. Za tímto účelem je možné použít tři konstrukce, které umožňují buď kooperativní nebo preemptivní multitasking.

Kooperativní multitasking se vyznačuje tím, že právě probíhající aplikace spolupracuje s operačním systémem a po dokončení úlohy předává cpu zpět operačnímu systému. Tento způsob je méně náročný na hardwarové prostředky, pokud ovšem při provádění úkolu dojde k selhání aktivní aplikace hrozí pád celého systému. V dynamic C j e kooperativní multitasking umožněn použitím syntaxe (převzato z [6]):

```
costate [ name [state] ] { [ statement | yield; | abort; | waitfor( expression ); ] . . . }
```

Klíčové slovo *costate* před blokem uzavřeným ve složených závorkách označuje task. Ten může nabývat stavy naznačené na obr.18 . Pomocí příkazů *yield*, *abort*, *waitfor*, nebo *waitfordone* lze přecházet mezi jednotlivými stavy a je tak možné předávat kontrolu CPU ostatním taskům.



Obr.18 – Možné stavy tasku při použití kooperativního multitaskingu (převzato z [6])

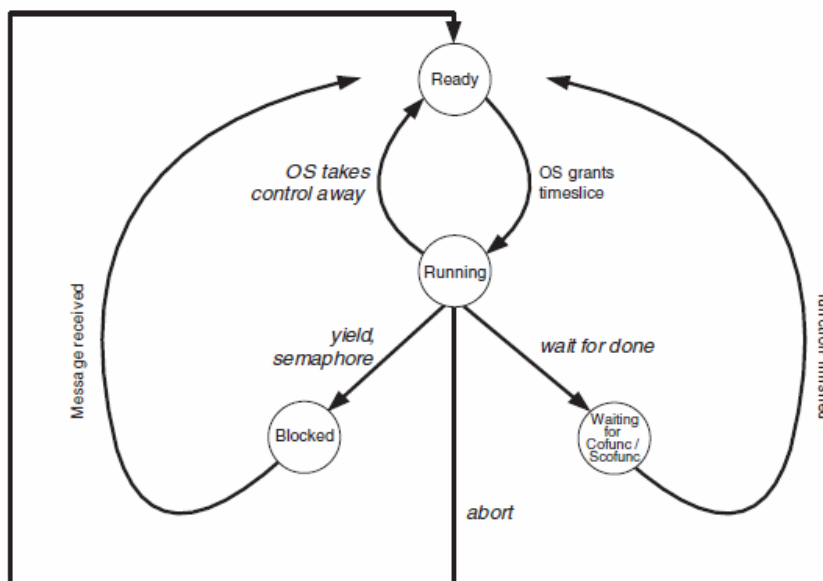
Preemptivní multitasking je technicky náročnějším, avšak bezpečnějším způsobem řízení paralelně prováděných úloh. Při tomto uspořádání řídí přidělování cpu operační systém. Tasky využívají svůj paměťový prostor o zadané velikosti. Stavy tasků při použití preemptivního multitaskingu a přechody mezi nimi jsou naznačeny na obr.19. Dynamic C umožňuje použití dvou metod pro implementaci preemptivního multitaskingu:

1. konstrukci slice
2. systém reálného času $\mu\text{C}/\text{OS-II}$.

První ze zmiňovaných způsobů je použitím podobný uspořádání costate a využívá syntaxe (převzato z [6]):

```
slice ([context_buffer,] context_buffer_size, time_slice)  
[name] {[statement|yield;|abort;|waitfor(expression);]}
```

Použití operačního systému $\mu\text{C}/\text{OS-II}$ je překladači deklarováno připojením potřebné knihovny ucos2.lib. Vytvoření tasků, semaforů, řízení přístupu ke zdrojům a cpu apod. je pak prováděno pomocí maker a funkcí této knihovny.



Obr.19 – možné stavy tasku při použití preemptivního multitaskingu (převzato z [6])

6.2.3 SBC BL2600

Pro vývoj aplikací s RCM 3200 lze využít základní desku SBC BL2600. Ta je i součástí vývojového kitu s RCM 3200 a je osazena tímto modulem. Nabízí rozšíření o dva porty RJ45 (Rabbit Net pro rozšiřující desky), A/D a D/A převodníky, RS485/RS232 porty apod.. Některé programové příklady v adresáři samples

vývojového prostředku Dynamic C jsou přímo určeny pro použití s touto deskou a nabízejí tak možnost velmi jednoduchého praktického odzkoušení při výuce systému.

6.3 Mikroprocesor ATMEGA 2560-16AU

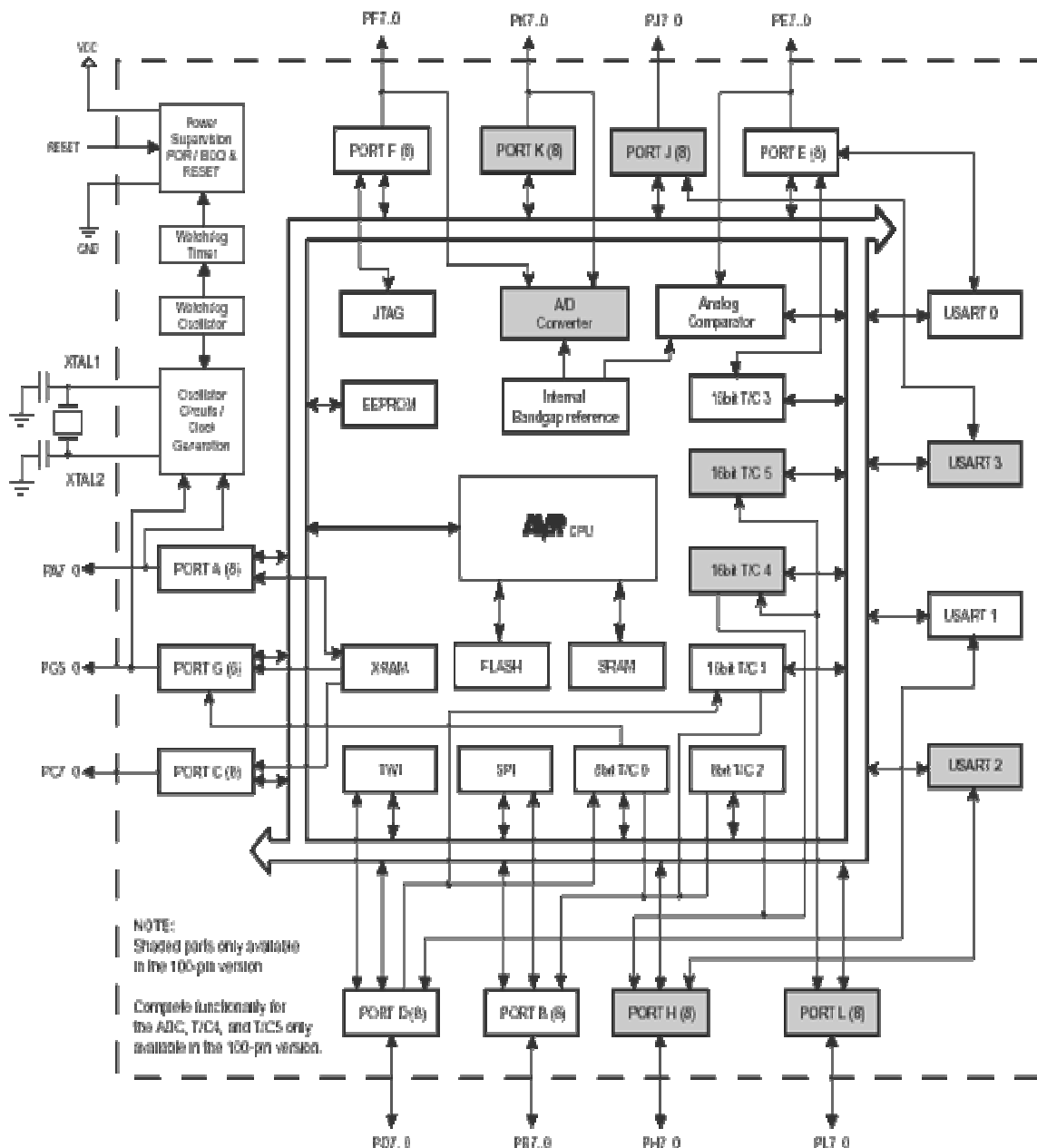
Pro řízení LED matice v této aplikaci jsem, především kvůli počtu vstupů a velikosti napájecího napětí (až 5.5V) vybrán mikroprocesor ATMEGA 2560-16AU. Velikosti paměti flash a RAM jsou také dostatečné.

Tento osmibitový mikroprocesor s jádrem AVR firmy Atmel je postaven na rozšířené RISC architektuře. Obsahuje 256 kB programové paměti typu flash s životností až 10000 zápisů, 8 kB SRAM a 4 kB EEPROM s životností až 100000 zápisů. Programování flash je možné buď pomocí SPI nebo JTAG rozhraní a to přímo v aplikaci. Relativně velké množství vstupně výstupních portů je rozloženo v pouzdře TQFP nebo CBGA se 100 vývody. Jako vstupy nebo výstupy pro aplikaci lze využít až 86 vývodů. Všechny I/O piny lze nastavit buď jako vstupy nebo jako výstupy. Při nastavení jako digitální vstup lze u každého z pinů také programově sepnout propojení na pull-up rezistor. 16 vstupních pinů můžeme využít jako analogově – digitální převodníky s rozlišením 10 bitů. Lze využít 12 PWM kanálů s volitelným rozlišením 2-16 bitů. K dispozici jsou 2 časovače s osmibitovým rozlišením a 4 šestnáctibitové časovače. Časovače lze provozovat v jednom z šestnácti módů. Lze také spínat některé výstupy přímo časovačem, respektive v závislosti na stavu časovače, nezávisle na vykonávaném programu. Pro komunikaci je možné využít například 4 USART zařízení nebo TWI zařízení, což je v podstatě úprava standardu I2C. Přehledové schéma prostředků je uvedeno na obr.6. Pro úsporu energie může mikroprocesor pracovat v 6 šetřících režimech. Samozřejmostí je watchdog, uzamykací bity nebo nastavení zdroje hodinového signálu. Mikroprocesor pracuje v základním nastavení na frekvenci 8MHz generovaného vnitřním oscilátorem a děleného osmi. Lze však použít externí oscilátor s frekvencí až 16MHz. Při této hodinové frekvenci uvádí výrobce výkon až 16MIPS.

Rozsah napájení Mikroprocesoru je od 1,8V do 5,5V, přičemž pro rozsah napájení 1,8V-5,5V lze použít hodinovou frekvenci 0-2MHz, pro rozsah 2,7V-5,5V může být frekvence 0-8MHz a při napájecím napětí v rozsahu 4,5V-5,5V je možné pro taktování mikroprocesoru použít frekvence 0-16MHz. Výstupní piny mohou dodávat do obvodu proud až 40mA. Proud vývody Vcc a Gnd může být však maximálně 200mA.

Program pro tento mikroprocesor lze vyvíjet pomocí mnoha nástrojů. Lze programovat pomocí jazyka symbolických instrukcí nebo i pomocí vyšších jazyků jako jsou C, C++ nebo pascal. Jako příklad programovacího nástroje uvedu Atmelstudio 6.0, které jsem také pro vytvoření projektu v diplomové práci použil. Tento program je nástupce předchozích vývojových prostředků firmy Atmel AVR studio 4 a 5. Je určen k programování mikroprocesorů s jádrem AVR a ARM vyráběných firmou Atmel pomocí jazyka symbolických instrukcí, jazyka C nebo C++. Tento nástroj integruje

ARM GCC, AVR GCC, AVR assembler, Atmel gallery, Atmel software framework, visual assist X for Atmel studio. Kromě nástrojů pro vývoj programu lze pomocí Atmel studia programovat flash mikroprocesorů za použití standardních programátorů pro SPI nebo JTAG. Program lze také odladit pomocí simulátoru daného mikroprocesoru, který je součástí Atmel studia.



Obr.20 – přehledové schéma Atmega 2560 (převzato z [11])

7 NÁVRH MODULU RGB LED DISPLEJE

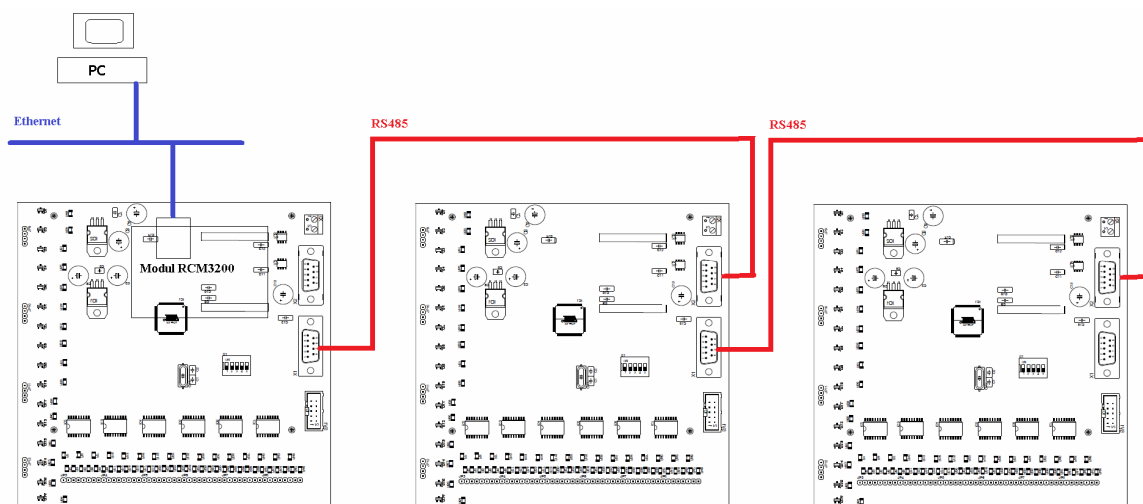
V další části práce se zaměřím na popis vlastního návrhu zobrazovacího modulu RGB LED displeje. Uvedu stručně popis základních součástí modulu a součástky použité v návrhu. Popíšu také software vytvořený v rámci práce pro řízení zobrazovacího modulu, ethernet interface a pro testování komunikace s PC.

7.1 Koncepce zobrazovače a hardware navrženého modulu

Při výběru koncepce zobrazovače byla důležitá zvláště dvě kritéria, cena a dodržení modularity systému. Pro návrh jsme tedy zvolili uspořádání, kdy zobrazovací moduly jsou řízeny levnějším, avšak dostatečně výkonným mikrokontrolérem atmega 2560-16AU a jsou připojeny ke sběrnici RS485 jako slave stanice. Rozesílání obrazových dat zobrazovacím modulům pak zabezpečuje programovatelný modul Rabbit RCM3200, který zároveň disponuje integrovaným rozhraním ethernet, pomocí něhož přijímá data z nadřazeného PC. Standardně používané převodníky pro RS485 (a také mnou použitý obvod max485) omezují počet připojených stanic na 32. Pro každých 31 zobrazovacích modulů je tedy nutné použít nový master Rabbit RCM3200.

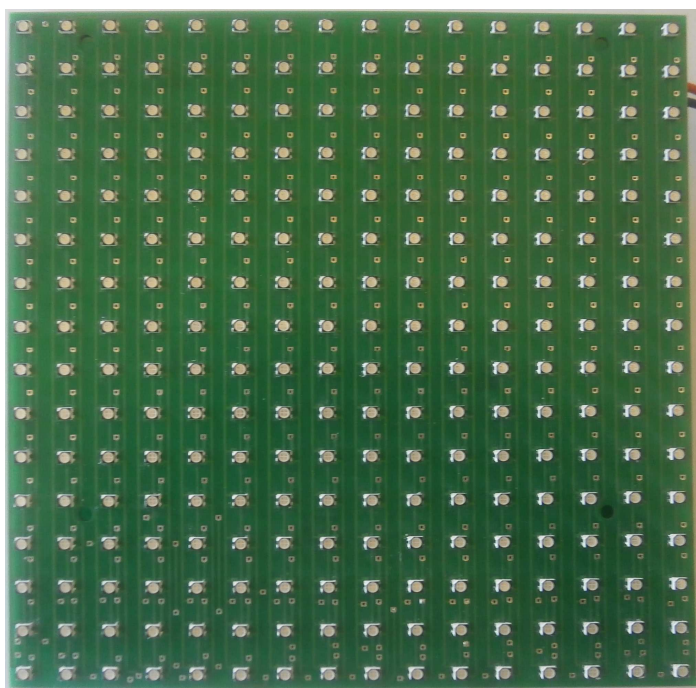
Jako převodník Ethernet-RS485 je samozřejmě možné při dodržení komunikačního protokolu použít jakékoliv jiné zařízení. Toto zařízení může být vloženo stejně jako Rabbit RCM3200 do konektorů na desce modulu zobrazovače nebo připojeno pomocí volného konektoru d-sub X1 či X2.

Návrh zobrazovacího modulu obsahuje zobrazovací matici o velikosti 16x16 RGB LED. Modul je sestaven ze dvou desek plošných spojů. Vrchní deska obsahuje LED matici a hřebínkové lišty pro připojení signálů z řídicí desky. Spodní deska plošných spojů je osazena řídicím mikroprocesorem atmega 2560-16AU, obvody pro úpravu napájení, tranzistory pro buzení řádků, obvody ULN 2803 pro buzení sloupců a rezistory, upravujícími pracovní bod RGB LED. Dále jsou tu dva konektory 2x17 pinů pro vsazení modulu RCM 3200, který slouží jako rozhraní ethernet/RS485 a distribuuje požadované jasové hodnoty jednotlivým modulům připojeným na RS485. Převod signálových úrovní procesor/RS485 je proveden integrovanými obvody max485 (obr. 8). K propojení modulů budou sloužit devítipinové konektory d-sub, kde na pin 2 je přiveden signál A a na pin 8 signál B sběrnice RS 485. Dalším konektorem je desetipinový idc konektor sloužící pro programování mikroprocesoru atmega pomocí SPI. Propojení jednotlivých modulů je naznačeno na obr.x. Napájení je na řídicí desku přivedeno pomocí svorkovnice do desek plošných spojů. Napájecí zdroj pro uzpůsobení síťového napájení potřebám panelu není součástí návrhu a při testování prototypu byl použit univerzální zdroj 230VAC/7,5VDC.

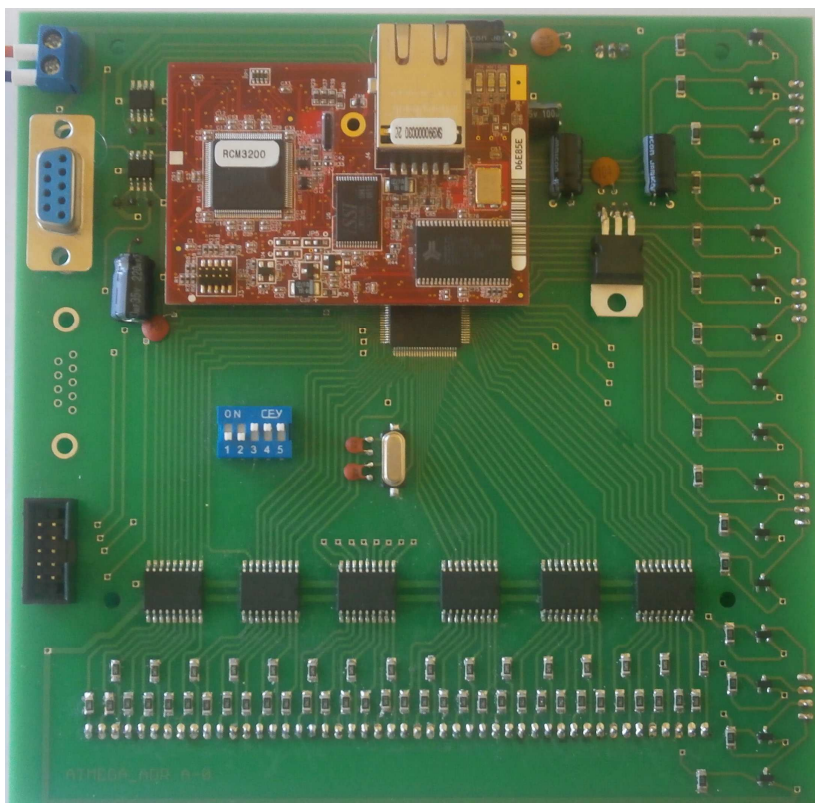


Obr.21 – Propojení RGB LED modulů

Schéma zapojení a návrh desek plošných spojů jsem vytvořil pomocí programu Eagle a jsou součástí práce jako přílohy 1 a 2. Jak již bylo uvedeno návrh sestává ze dvou oboustraných desek plošných spojů o velikosti 158x158mm, které budou spojeny pomocí hřebíkové lišty. Předpokládá se, že elektronika modulu bude vestavěna do boxu s okrajem jeden milimetr a celkový rozměr modulu bude 160x160mm, tak aby rozteče LED mezi moduly sestavenými do zobrazovacího panelu byla 10 mm. Pro pevné spojení jsou v deskách čtyři vrtané otvory o velikosti 3mm. Osazená deska s LED maticí je zobrazena na obr.22 a deska řízení s osazeným modulem RCM3200 je na obr.23. Hlavní části zapojení jsou rozebrány v následujících kapitolách.



Obr.22 – osazená deska plošných spojů RGB LED matice



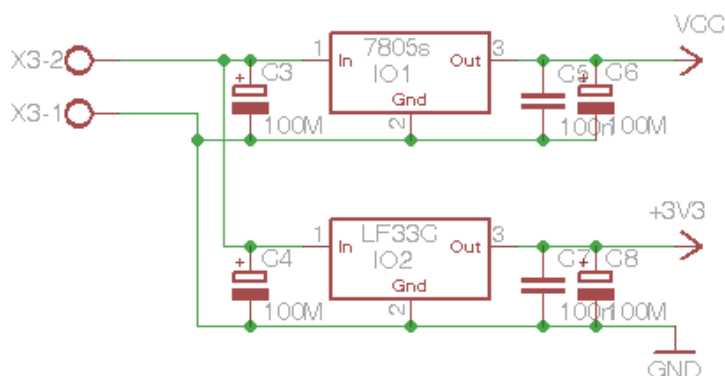
Obr.23 – osazená řídicí deska zobrazovacího modulu

7.1.1 Napájení

Napájení je přivedeno na řídicí desku pomocí svorkovnice X3. Toto napětí je dále upraveno integrovanými stabilizátory 78S05 (IO1) a LF33CV (IO2) na hodnoty 5V a 3,3V. Schéma zapojení zdrojové části modulu je na obr. 24.

Maximální rozsah napájecího napětí pro obvod 78S05 je 35V. Pro obvod LF33CV je maximální vstupní napětí, kdy ještě tento obvod reguluje 18V. Napájecí napětí RGB modulu je tedy omezeno na rozsah 5-18V. Při stavbě prototypu jsem pro úpravu napětí nouzově použil obvod LT1086CT-3.3, jako náhrada LF33CV. Pro tento obvod je přípustné vstupní napětí 20V, rozsah napájení modulu je tedy zachován.

Je nutné vzít v úvahu poměrně vysoký proud odebíraný z 5V stabilizátoru, který by podle předpokladů (viz kapitola 7.1.2) mohl být špičkově až 1,68 A. Stabilizátor 78S05 je podle katalogového listu schopen dodávat do obvodu maximální proud 2A. Na napájení řídicího mikroprocesoru a ostatních obvodu zbývá tedy cca 0.32A. To by podle autora práce mělo být dostačující. Prototyp byl zatím zkoušen bez přídatného chlazení stabilizátorů, bude však vhodné pro oba stabilizátory použít pasivní chladič.



Obr.24 – zdrojová část modulu

7.1.2 RGB LED matice a její buzení

Matice RGB LED je sestavena z smd LED s typovým označením PLCC4RGBCT-CA. Původně jsem uvažoval o použití LED s pouzdem PLCC6, kde jsou vyvedeny jak katody, tak anody, rozhodující byla ale nižší cena použitých LED. Některé typické a maximální hodnoty pro tento typ diody jsou uvedeny v tab.1.

Maximální stálý proud v propustném směru je 25mA, pro účely rychlého spínání s malou střídou je ale nutné zvýšit intenzitu emitovaného světla vyšším proudem. Podle katalogového listu je pro navrženou LED při frekvenci spínání 1kHz a střídě 1:10 maximální proud až 100mA. Rezistory pro omezení proudu na 35mA byly navrženy experimentálně. Bylo použito napájecí napětí 5V a rezistory vypočtené pro nastavení pracovního bodu 25mA při předpokládaném úbytku 0,7V na Darlingtonově dvojici:

$$R_R = \frac{5 - 0,7 - 2}{0,025} = 92\Omega \cong 100\Omega ; \quad (7.1)$$

$$R_G = R_B = \frac{5 - 0,7 - 3,2}{0,025} = 44\Omega \cong 47\Omega . \quad (7.2)$$

Pomocí osciloskopu byl zjištěn skutečný úbytek na ULN2803 a vypočteny rezistory pro proud 35mA diodami R, G a B : $R_R = 56\Omega$; $R_G = R_B = 27\Omega$.

Matice LED je uspořádána tak, že v řádcích jsou spojeny společné anody LED, ve sloupcích jsou propojeny katody. Spínání sloupců, tedy spojení katod se záporným pólem napájecího napětí je provedeno pomocí šesti integrovaných obvodů ULN2803. Tyto integrované obvody obsahují osm dvojic bipolárních tranzistorů, uspořádaných

v Darlingtonově zapojení. Kolektorový proud výstupního tranzistoru je až 500mA, napětí U_{CE} až 50V. Pro předpokládaný impulzní proud LED 35mA je tedy vyhovující. Výhody použití integrovaného obvodu ULN2803 jsou především ve velkém zesílení a tedy spolehlivém sepnutí už při malém bázevém proudu a v malých rozměrech. Pro 5V logiku není nutné také použít vstupní omezovací prvky. Na Darlingtonově zapojení však vzniká poměrně velký napěťový úbytek a pro spínání řádků je tedy při zachování 5V napájecího napětí nutné použít prvek s malým vnitřním odporem, tak aby napětí mezi anodou a katodou RGB LED bylo dostatečné (viz tab. 1).

Tab.1 – Některé typické a maximální hodnoty RGB LED PLCC4RGBCT-CA (převzato z [10])

Parametr	Color	Test condition	Min	Typ	Max	Unit
Forward voltage	Red	IF=20mA		2.0	2.4	V
	Green			3.2	3.5	
	Blue			3.2	3.6	
Forward current	Red				25	mA
	Green					
	Blue					
Pulse current	Red	Duty=0,1; 1kHz			100	mA
	Green					
	Blue					
Luminous intensity	Red	IF=20mA	102	280		mcd
	Green		280	770		
	Blue		145	280		
Viewing Angle	Red	IF=20mA				Deg.
	Green			120		
	Blue					
Dominate wavelenght	Red	IF=20mA		625		nm
	Green			525		
	Blue			470		

Dalším omezujícím kritériem pro výběr spínacích prvků pro řádek je poměrně vysoký proud, který bude tímto prvkem protékat. Při předpokládaném impulzním proudu jednou LED 35mA je maximální proud celým řádkem $I_r = 3 \cdot 16 \cdot 0,035 = 1,68A$.

Na základě výše uvedených požadavků jsem pro spínání řádků vybral unipolární tranzistor MOSFET s kanálem typu P IRLM 9301 v provedení smd. Zapojení tranzistoru pro spínání řádku je na obr.25. Důležité mezní hodnoty pro tento tranzistor jsou:

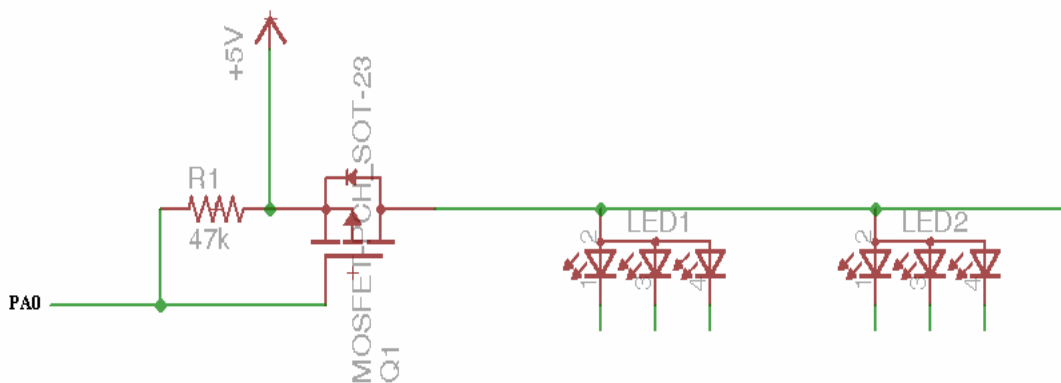
proud drainem: $I_D = -3,6A$,

maximální napětí drain-sorce: $V_{DS} = -30V$,

odpor mezi drain a source při sepnutém stavu, při $U_{GS} = -4,5V$: $R_{DS} = 103m\Omega$,

gate treshold napětí: $-1,3V - -2,4V$,

maximální ztrátový výkon: 1,3W při 25°C a 0,8W při 70°C.



Obr.25 – Schéma spínání řádku RGB matice

Pro spolehlivé sepnutí tranzistoru je důležitou hodnotou údaj gate treshold, který udává rozsah hodnot řídicího napětí, kdy může dojít u daného typu tranzistoru k plnému sepnutí (rozepnutí). Skutečné prahové napětí se může lišit kus od kusu, ale vždy by mělo být v tomto rozsahu. V použitém zapojení je otevření tranzistoru závislé na rozdílu výstupního napětí portu mikroprocesoru a napájecího napětí. Jsou nutné tyto hodnoty výstupního napětí portu:

pro otevření $< 2,6V$

pro zavření $> 3,7V$.

Výrobce Atmel uvádí pro mikroprocesor atmega 2560 při napájení 5V hodnoty minimálně 4,2V pro stav logická jedna a maximálně 0,9V pro stav logická nula. Spínací podmínky by tedy měly být splněny.

Ztrátový výkon na tranzistoru při proudu $I_D=1,68A$ a odporu drain-source $R_{DS} = 103m\Omega$ bude:

$$I_D^2 \cdot R_{DS} = 1,68^2 \cdot 0,103 = 0,29W . \quad (7.3)$$

To je méně než maximální hodnota i při teplotě $70^\circ C$.

7.1.3 Řízení LED (Atmega 2560-16AU)

Pro řízení modulu RGB LED jsem vybral osmibitový mikroprocesor atmega 2560-16AU od firmy atmel. Rozsah napájení tohoto zařízení respektive velikost výstupního napětí je dostačující pro spínání řádku pomocí unipolárního tranzistoru (viz kapitola 7.1.2). Mikroprocesor je vybaven dostatečným počtem vstupně/výstupních vývodů. K řízení LED matice, respektive pro spínání řádků a sloupců, je použito 64 výstupů, pro komunikaci pomocí usart zařízení a povolení zápisu/čtení jsou využity 3 I/O piny, pro připojení přepínače nastavujícího binárně adresu modulu na RS485 je nutných 5 pinů a 3 piny portů jsou rezervovány k programování mikroprocesoru atmega pomocí SPI. Moduly řízení LED na základě přijatých jasových hodnot jednotlivých led spínají proud diodami s takovou frekvencí, aby spínání nebylo pro lidský zrak pozorovatelné. Frekvence spínání, kdy už oko spolehlivě nerozlišuje blikání světelného zdroje je 50Hz. Obnovovací frekvence celého displeje by tedy měla být nejméně než 50Hz.

7.1.4 Společné řízení modulů a komunikace s PC (Rabbit RCM 3200)

Jako hlavní řídicí část modulárního displeje je využit vývojový modul RCM3200 od firmy Digi internacional. Tento modul bude vsunut do jedné z desek řízení LED pomocí dvou konektorů 2x17 pinů. Na těchto konektorech je vyvedeno napájení RCM3200 (tedy stabilizované napětí 3.3V) a je zde také propojení sériové komunikace modulu na jeden z převodníků MAX485.

Modul Rabbit RCM3200 přijímá data z PC pomocí ethernetového připojení, zpracovává je a dále je rozesílá zobrazovacím modulům připojeným na RS 485.

7.2 Softwarové vybavení

V rámci vypracování diplomové práce jsem sestavil tři programy. Program pro řízení led a komunikaci po sériové lince RS485 v mikrokontroléru atmega, program pro příjem dat z nadřazeného PC a rozesílání těchto dat zobrazovacím modulům v mikrokontroléru Rabbit a program pro testování navrženého zapojení pomocí PC.

Všechny tři programy jsou součástí příloženého DVD a jejich základní části a algoritmy jsou popsány v následujících podkapitolách.

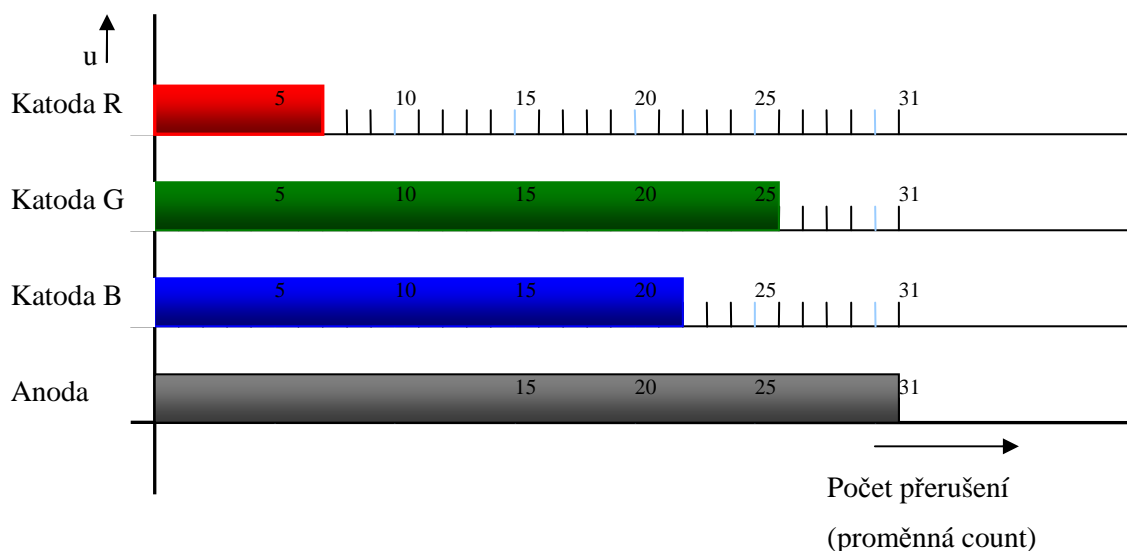
7.2.1 Program pro mikrokontrolér atmega 2560

7.2.1.1 PWM řízení LED

Základním úkolem mikrokontroléru atmega 2560 je řízení svitu RGB LED. Uvažoval jsem původně o využití true color schématu, kdy by pro jednu barvu bylo vyhrazeno 8 bitů a celkový počet zobrazovaných barev by byl 16 777 216. Při sestavování programu pro mikrokontrolér atmega jsem však kvůli rychlosti překreslování celého obrazu zvolil maximální rozlišení na jednu základní barvu 31 odstínů (respektive 32 úrovní včetně zhasnuté LED = černá). Panel je tedy schopen zobrazit 32768 barev.

Mikrokontrolér nabízí sice několik PWM kanálů, ty jsou ale vázány na určité výstupy. Použití PWM kanálů by tak muselo být zahrnuto už do návrhu desky plošných spojů. Návrh by ale pak byl zbytečně složitý a nepřehledný. Spínání výstupů řídicích LED je tedy řízeno v přerušení vyvolaném časovačem 1.

Princip řízení barev je naznačen na Obr.26. Přerušení od časovače 1 je vyvoláno po $40,6 \mu s$. Čas po který může maximálně LED svítit (nebo také čas vyhrazený jednomu řádku) je 31 přerušení, tedy 1,3 ms. Pro zobrazení celého obrazu je postupně rozsvíceno a zhasnuto 16 řádků. Čas nutný pro obnovení obrazu je tedy 20ms. To odpovídá obrazové frekvenci 50Hz. Časové intervaly jsem takto zvolil s ohledem na to aby ještě obraz rušivě neblíkal, ale zároveň zbylo co nejvíce času na ostatní úkoly, jako je především načtení zprávy ze sériového portu a příprava dalšího obrazu.



Obr.26 – Intervaly sepnutí základních LED a řádku pro barvu R=7,G=26,B=22

Pro uložení barev v obraze jsem použil dvourozměrné pole struktur jas. Struktura jas sjednocuje a pojmenovává pouze tři proměnné typu uchar ve kterých jsou uloženy jasové složky barev red, green a blue. Je definována takto:

```
typedef struct {unsigned char R; unsigned char G; unsigned char B;} jas;
```

Rozsah typu unsigned char by dovoľoval uložení 256 hodnot. Jak jsem ale uvedl na začátku této kapitoly, z technických důvodů je použito pouze pět bitů. Uložení jedné barvy by sice bylo možné pomocí patnácti bitů (respektive 16), pro další práci s uloženými složkami barvy a pro adresaci těchto složek je toto uspořádání výhodnější. Původní porovnání zadané hodnoty barvy s pořadím právě probíhajícího cyklu a následné sepnutí nebo rozepnutí výstupu pro ovládání LED v přerušení se ukázalo být příliš dlouhé a při zpracování všech LED překročila doba zpracování instrukcí dobu přerušení. Z tohoto důvodu jsem zvolil postup kdy v hlavním programu připravím hodnoty celých portů použitých jako výstupy pro led a uloží je do proměnných odpovídajících stavu portu v právě probíhajícím cyklu. Za tímto účelem jsem vytvořil Osum statických polí odpovídajících využitým portům B,C,D,E,G,H,K,L. Pro určení kde se nachází výstup příslušný dané LED jsem deklaroval pole Radr, Gadr a Badr. Prvky tohoto pole jsou struktura mapadr, která obsahuje jednak ukazatel na zástupce portu a jednak číslo bitu v rámci daného portu. Nastavení hodnot výstupních portů v časech 0-31 a jejich uložení do příslušné proměnné pak probíhá pomocí nepřímé adresace a následujícího kódu:

```
for(tmpradek=0; tmpradek<=RADEKMAX; tmpradek++)
{
    for (tmpbarva=0; tmpbarva<=BARVAMAX; tmpbarva++)
    {
        tmpB[tmpbarva]=0;
        tmpC[tmpbarva]=0;
        tmpD[tmpbarva]=0;
        tmpE[tmpbarva]=0;
        tmpG[tmpbarva]=0;
        tmpH[tmpbarva]=0;
        tmpK[tmpbarva]=0;
        tmpL[tmpbarva]=0;
    }

    for (tmpsloupec=0; tmpsloupec<=SLOUPECMAX; tmpsloupec++)
    {
        for (m1=0; m1<LED[tmpradek][tmpsloupec].R; m1++)
            (*(Radr[tmpsloupec].port+m1)|=Radr[tmpsloupec].pin);}
        for (m2=0; m2<LED[tmpradek][tmpsloupec].G; m2++)
            (*(Gadr[tmpsloupec].port+m2)|=Gadr[tmpsloupec].pin);}
        for (m3=0; m3<LED[tmpradek][tmpsloupec].B; m3++)
            (*(Badr[tmpsloupec].port+m3)|=Badr[tmpsloupec].pin);}
    }
```

```

for (tmpbarva=0; tmpbarva<=BARVAMAX; tmpbarva++)
{
    outB[tmppradek][tmpbarva]=tmpB[tmpbarva];
    outC[tmppradek][tmpbarva]=tmpC[tmpbarva];
    outD[tmppradek][tmpbarva]=tmpD[tmpbarva];
    outE[tmppradek][tmpbarva]=tmpE[tmpbarva];
    outG[tmppradek][tmpbarva]=tmpG[tmpbarva];
    outH[tmppradek][tmpbarva]=tmpH[tmpbarva];
    outK[tmppradek][tmpbarva]=tmpK[tmpbarva];
    outL[tmppradek][tmpbarva]=tmpL[tmpbarva];
}
}

```

Nastavení probíhá po řádcích. Pro zpracování je použita dočasná proměnná, aby došlo vždy k rychlému přepsání celého řádku. Nejdříve je celý řádek (prvky pro jas 0-31) v „zástupci“ portu vynulován. V cyklech for jsou pak procházeny jednotlivé prvky matice LED a načten jas. Odpovídající bit v „zástupci“ portu je nastaven do 1 pro všechny prvky s indexem menším než hodnota jasu. Následně jsou hodnoty pro příslušný řádek uloženy pro použití v přerušení. Uvedený způsob je náročný na paměťový prostor, dovoluje však velmi krátké zpracování výstupů v přerušení, kde je pak pouze zapsána na výstup hodnota „zástupce“ portu.

7.2.1.2 Komunikace pomocí RS485

Dalším úkolem programu je obsluha komunikace pomocí RS 485. K tomuto účelu je využito jedno ze čtyř zařízení USART (asynchronní / synchronní sériový přijímač a vysílač), které jsou v mikrokontroléru k dispozici. V programu jsem vytvořil funkce pro příjem a odeslání zprávy na RS485, použito je ale pouze přijímání obrazových dat. Parametry komunikace, které jsou nastaveny ve funkci USART0_Init jsou:

Počet datových bitů:	8
Počet stopbitů:	1
Parita:	žádná
Rychlost:	76800 bit/s.

Pro komunikaci mezi nadřazeným zařízením (rabbit RCM3200) a zobrazovacím modulem jsem zvolil krátké zprávy o délce sedm bytů. Výhodnější z hlediska rychlosti přenosu by byly delší zprávy, kde by byla obsažena celá obrazová informace, při příjmu by však mohlo dojít ke ztrátě informace při vykonávání časového přerušení pro obsluhu zobrazení nebo naopak k výpadku zobrazení vlivem odskoku do přerušení od USART. Složení zprávy je znázorněno na obr.27. Komunikační protokol je postaven na faktu, že pro zadání barvy nebo adresy v rámci sítě je třeba jen 5 bitů v bytu (bity 0-4) a pokud je některý další bit obsazen jde o klíčový byte. Bit 5 označuje stop byte. Bity 6 a 7 označují start byte a mohou být využity pro rozlišení typu zprávy (command 1-3). Pro nastavení barvy LED na modulu jsou oba bity v log. 1. Zpráva pro nastavení LED má tedy následující složení:

byte1 = start (command + 0 + adresa příjemce),
 byte2 = souřadnice řádku (0-15),
 byte3 = souřadnice sloupce (0-15),
 byte4 = hodnota složky R (0-31),
 byte5 = hodnota složky G (0-31),
 byte6 = hodnota složky B (0-31),
 byte7 = stop (001 + adresa odesílatele).

V programu je příjem dat z RS485 ošetřen pomocí hardwarového přerušení vyvolaného při přijatém bytu na USART0. V obsluze přerušení je tento byte zapsán do bufferu a otestován na přítomnost startbyte nebo stopbyte. Pokud je v buferu načtena celá sedmibytová zpráva, je inkrementována proměnná pro čítač čekajících zpráv. Kvůli pomalému zpracování hlavním programem používám pro uložení zpráv pro buffer dvourozměrné pole proměnných char. První rozměr udává index právě aktuálního bufferu a druhý je pozice bytu ve zprávě. Zpracování přijatých zpráv probíhá nezávisle na přijímání v hlavním programu pomocí následujícího algoritmu:

```

RXenable();

    if(RXtgmOK>0)
    {

        if(RXbufInd<RXtgmOK)
            RXrdInd=(RXBUFINDMAX+1+RXbufInd-RXtgmOK);
        else
            RXrdInd=(RXbufInd-RXtgmOK);

        RXtgmOK--;

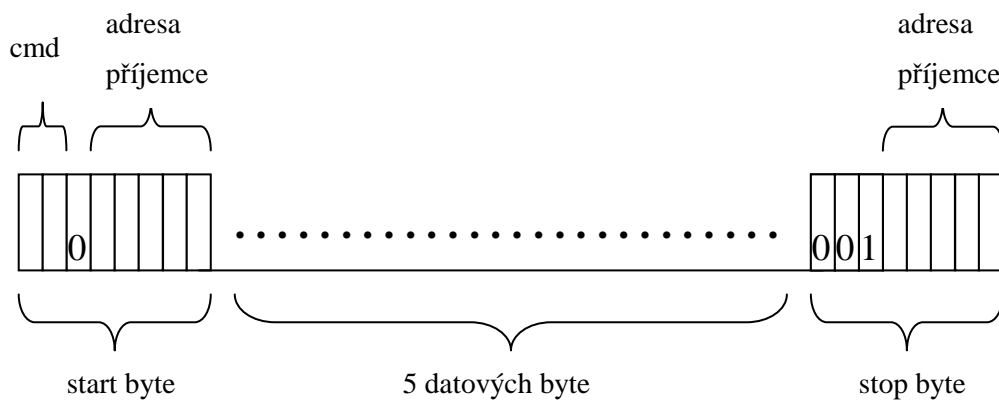
        if(RXbufRead(RXbuf[RXrdInd], RXTGMMAX, RXnew) == adr485)
        {
            RXsetLED(RXnew, LED);
        }
    }
  
```

Funkce RXbufRead načte data pro zpracování a vrátí adresu příjemce. Jestliže je tato adresa shodná s adresou modulu, je použita funkce RXsetLED, která uloží jasové hodnoty do matice LED na pozice řádku a sloupce. Adresa zobrazovacího modulu je zadávána pomocí DIP přepínače na zadní straně řídicí desky modulu a načítá se v hlavní smyčce ještě před zpracováním zprávy, takže ji lze měnit za běhu programu. Přepínání příjmu a odesílání je možné pomocí funkcí

```

inline void RXenable(void);
a
inline void TXenable(void); ,
  
```

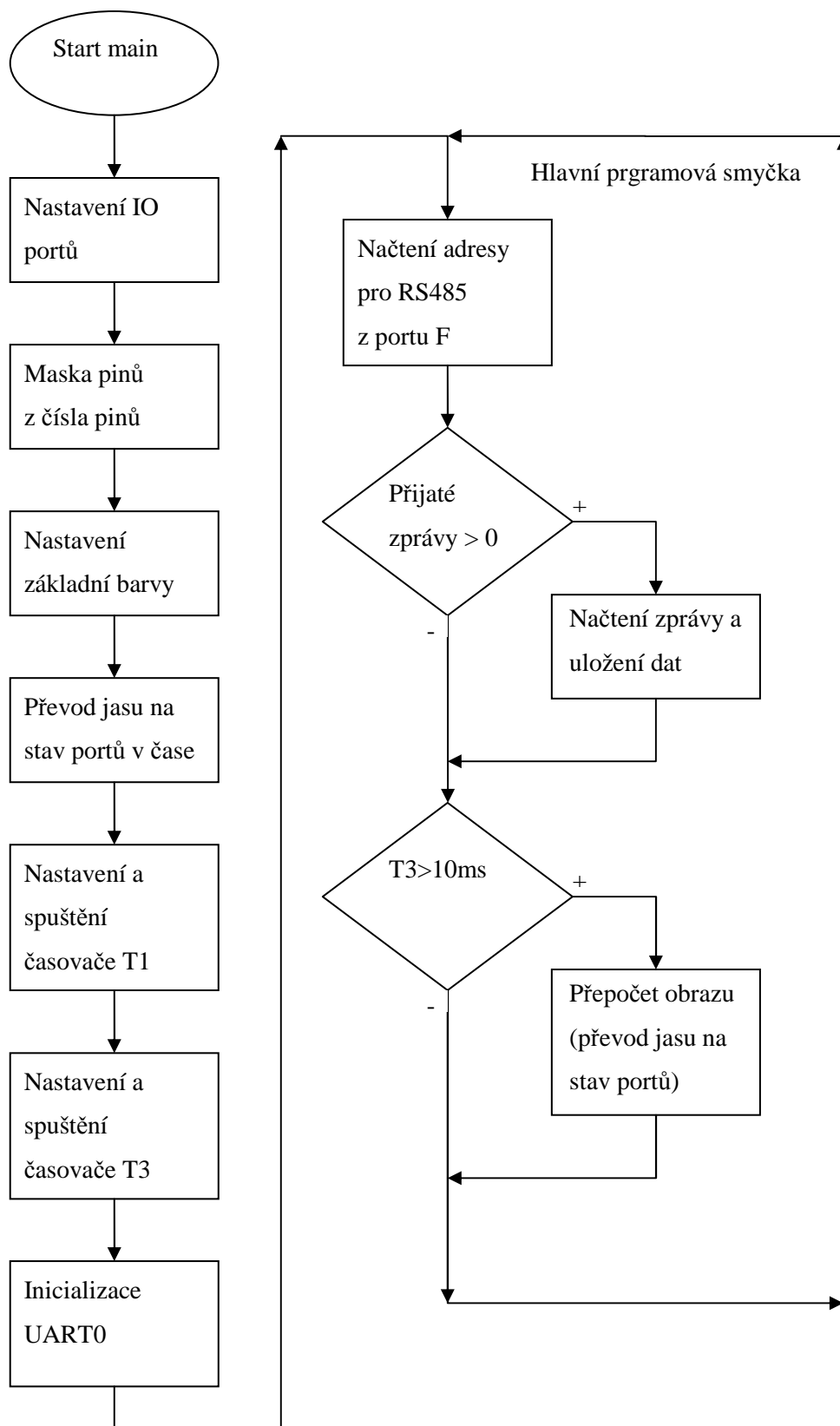
které řídí vstup obvodu MAX485 a zároveň povolují, či zakazují přerušení od načtení byte na USART0.



Obr.27 – Formát zprávy přijímané zařízením USART mikrokontroléru Atmega 2560

7.2.1.3 Funkce main

Příprava portů, časovačů i USART a také úlohy, jejichž pomalejší zpracování není tak kritické jako obsluha zobrazení a sériová komunikace jsou umístěny v hlavní funkci. Vývojový diagram této funkce je znázorněn na obr.28.



Obr.28 – vývojový diagram základní části programu v atmega2560-16AU

7.2.2 Program pro modul Rabbit RCM3200

Program pro mikrokontrolér rabbit 3000 jsem vypracoval pomocí vývojového prostředí Dynamic C, které lze bezplatně získat na stránkách společnosti Digi Internacional.

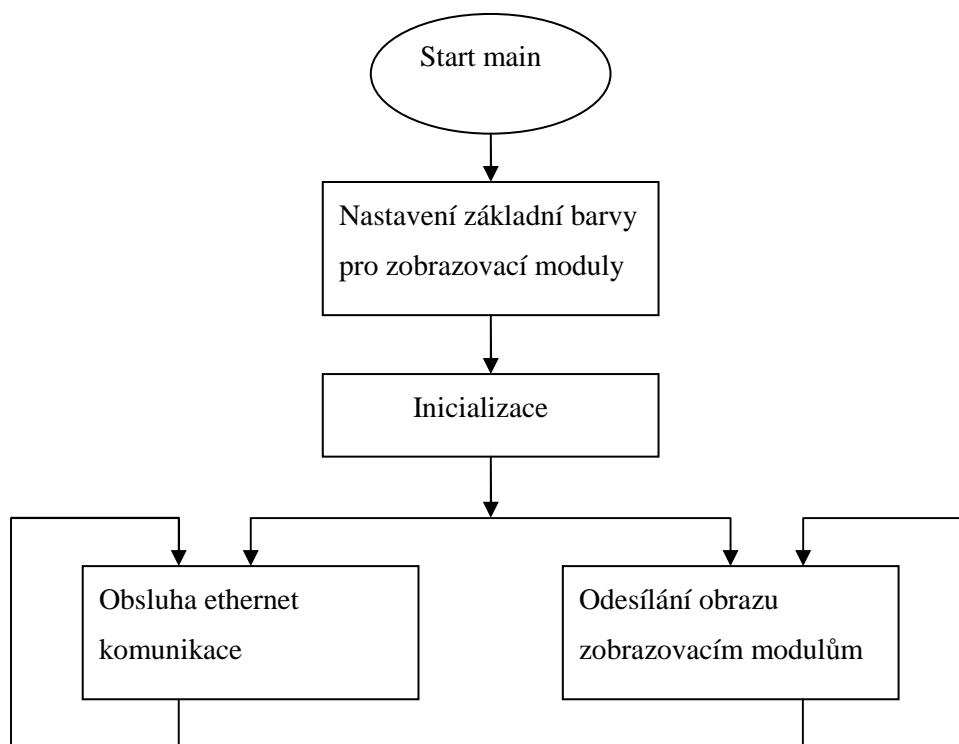
Programovatelný modul Rabbit RCM 3200 je použit v podstatě jako interface mezi nadřazeným PC a zobrazovacím modulem. Jako takový zabezpečuje především dvě funkce: komunikace s PC pomocí sítě ethernet a rozeslání obrazových dat pomocí RS485 do jednotlivých zobrazovacích modulů. Přijatá data jsou zde ale i ukládána a pokud by byla k zapojení připojena i baterie pro zálohování programu v paměti na desce Rabbit, mohl by být obraz zachován i po odpojení PC a vypnutí zařízení. Tato možnost však není využita. Pro uložení jasových hodnot slouží podobně jako v programu pro atmega pole struktur jas. Toto pole je ještě rozšířené o identifikaci zobrazovacího modulu adresou a je ve formátu LED [adresa panelu][řádek][sloupec].

7.2.2.1 Funkce main

Vývojový diagram hlavní funkce programu v mikrokontroléru Rabbit 3000 je naznačen na obr.29 . Po startu je nejdříve v cyklech for nastavena základní hodnota pro všechny připojené zobrazovací moduly, respektive tyto hodnoty jsou uloženy do pole vyhrazeného pro obrazy odesílané do těchto modulů.

V další části programu jsou provedeny základní nastavení. Pomocí funkce brdInit je provedeno nastavení vstupně výstupních pinů desky RCM3200. Mnou vytvořená funkce UARTinit nastavuje a spouští provoz na UART zařízení D. Funkce sock_init zabezpečuje přípravu socketu pro komunikaci pomocí ethernet. Do tohoto bloku spadá ještě základní nastavení pomocných proměnných ethState a ethTimeout.

V hlavní programové smyčce je použita konstrukce costate pro kooperativní multitasking. Tento nástroj umožňuje zdánlivý současný běh dvou tasků které zabezpečují jednak obsluhu rozesílání dat zobrazovacím modulům a jednak obsluhu ethernetové komunikace.



Obr.29 – Vývojový diagram programu v modulu Rabbit RCM3200

7.2.2.2 Obsluha rozesílání dat zobrazovacím modulům

Odeslání dat uložených v poli LED je provedeno vždy když je nastavena proměná obnovPanel[adresa panelu] na hodnotu 1 (byla přijata nová data z PC). V cyklech for jsou postupně odesílány hodnoty LED[adresa panelu][řádek][sloupec] pomocí funkce TXsetLEDadr. Parametry této funkce jsou řádek, sloupec, R složka, G složka, Bsložka, adresa příjemce, adresa odesilatele. Tato funkce zašle příjemci příkaz na změnu jasu příslušné LED. Využívá k tomu další funkci, vytvořenou pro setavení a odeslání zprávy na RS 485. Tato funkce je definována takto:

```

cofunc int TXbufSend(unsigned char buf[],int maxbuf, unsigned char cmd, unsigned
char outadr, unsigned char myadr)
{
    int byte;
    unsigned char startbyte, stopbyte;

    if(cmd>3) cmd=3;
    if(myadr>31) myadr=31;
    if(outadr>31) outadr=31;
    startbyte=(outadr | (cmd<<6));
  
```

```

stopbyte=(myadr | 0x20);

costate
{
    TXenable();
    wfd cof_serDputc(startbyte);
    for(byte=0; byte<=maxbuf; byte++)
    {
        wfd cof_serDputc ( buf[byte]);
    }
    wfd cof_serDputc(stopbyte);
    while(serDwrUsed() || BitRdPortI(SDSR, 2));
}
return(1);
}

```

Funkce přidá k zadaným datům start byte s adresou uživatele a typem zprávy (cmd), přiřadí zároveň stopbyte s adresou odesílatele a zabezpečí odeslání všech dat do bufferu zařízení UART D. Odeslání samotné je provedeno funkcí implementovanou v knihovně rabbit cof_serDputc. Při vykonávání této funkce je po naplnění a uzamčení bufferu UART předán automaticky čas procesoru ostatním blokům costate (ostatním „paralelním“ úlohám). Čekací smyčka `while(serDwrUsed() || BitRdPortI(SDSR, 2));` je použita pro jisté odeslání dat. Funkce `serDwrUsed()` kontroluje místo v bufferu sériového portu D a volání funkce `BitRdPortI(SDSR, 2)` zjišťuje stav systémového bitu, který indikuje zaneprázdnění portu D.

7.2.2.3 Obsluha ethernetové komunikace

Pro komunikaci mezi modulem Rabbit a PC jsem zvolil model, kdy RCM3200 funguje jako TCP server a očekává na portu 50486 připojení TCP klienta, spuštěného na PC. Po připojení jsou klientem odesílána data v časových intervalech a server na každou přijatou zprávu odpoví odesláním stavu.

Během inicializace ve funkci main je volána funkce `sock_init()` ze systémové knihovny `net.lib`. Tato funkce inicializuje řízení paketů a deklaruje použití knihovny `DCRTCP` a v podstatě tak připraví překladač na volání funkcí, které jsou spojené s použitím ethernet komunikace. Všechny další příkazy a funkce vztahující se ke komunikaci po ethernetu jsem umístil do funkce `EthManag();`. Tato funkce je vykonávána cyklicky v rámci druhého tasku a obsahuje sekvenci úkonů, které jsou nutné k navázání komunikace, příjmu dat z PC a odeslání dat do PC.

Stav ve kterém se funkce EthManag nachází je uložen v globální proměnné ethState. Po startu a inicializaci DCPRT je stav nastaven na 0. Pomocí proměnné ethState jsou definovány stavy které jsou popsány v následujícím textu.

ethState = 0 – vytvoření spojení

Jsou volány především funkce

```
sock_mode(&sock, TCP_MODE_BINARY); a  
tcp_listen( &sock, LOCAL_PORT, 0, 0, NULL, 0); .
```

První z těchto funkcí nastaví daný socket do binárního módu. Druhá funkce připraví systém na to, že socket sock bude čekat na portu LOCAL_PORT (nastaveno 50486) na příchozí spojení. Hodnota nula ve třetím a čtvrtém parametru znamená, že není preferována vzdálená adresa ani vzdálený port a budou akceptována spojení od jakékoliv vzdáleného zařízení.

ethState = 1 – čekání na spojení

Dotaz `if (!(sock_established(&sock)))` je použit pro zjištění zda je již uzavřeno spojení a jestli je možné přejít do stavu přijetí dat. Funkce `(sock_established(&sock))` uzavírá očekávané spojení. Pokud je přijat požadavek na připojení od vzdáleného zařízení provede tato funkce tzv. „handshake“ (výměnu dat pro navázání spojení) a vrátí hodnotu 1 po navázání spojení.

ethState = 2 – přijetí dat

Přijetí dat je provedeno voláním následujícího kódu:

```
prijato=sock_fastread(&sock, bufin, buflenin);
```

Funkce `sock_fastread` z knihovny `dcprt.lib` zabezpečí odeslání dat v parametru `bufin` o délce `buflenin` bytů a pak vrací počet odeslaných bytů nebo vrací hodnotu -1.

ethState = 3 – zpracování přijatých dat

V tomto bloku je proveden přenos přijatého bloku dat do příslušného pole LED podle načtené adresy. Zároveň je nastavena proměnná `obnovPanel[]`, aby při provádění tasku s obsluhou sériové komunikace došlo k v nejbližší možné době k odeslání přijatých dat do zobrazovacího modulu.

ethState = 4 – načtení dat pro odeslání

V bloku 4 obsluhy ethernet komunikace je umístěna pouze příprava bufferu pro odeslání dat a zjištění délky tohoto pole pro pozdější použití ve funkci `sock_write`.

ethState = 5 – odeslání dat

Je provedeno volání funkce `sock_write(&sock, bufout, buflenout)`. Tato provede odeslání dat připravených v poli `bufout`. Funkce vrací počet odeslaných bytů nebo -1 v případě že při odesílání došlo k chybě.

Pokud dojde v některém ze stavů 2 - 5 k chybě čtení či zápisu nebo pokud dojde k přerušení spojení, je voláním `sock_close(&sock)` uzavřeno spojení a nastaven stav `ethState = 0`.

7.2.3 Program pro zasílání obrazu z PC

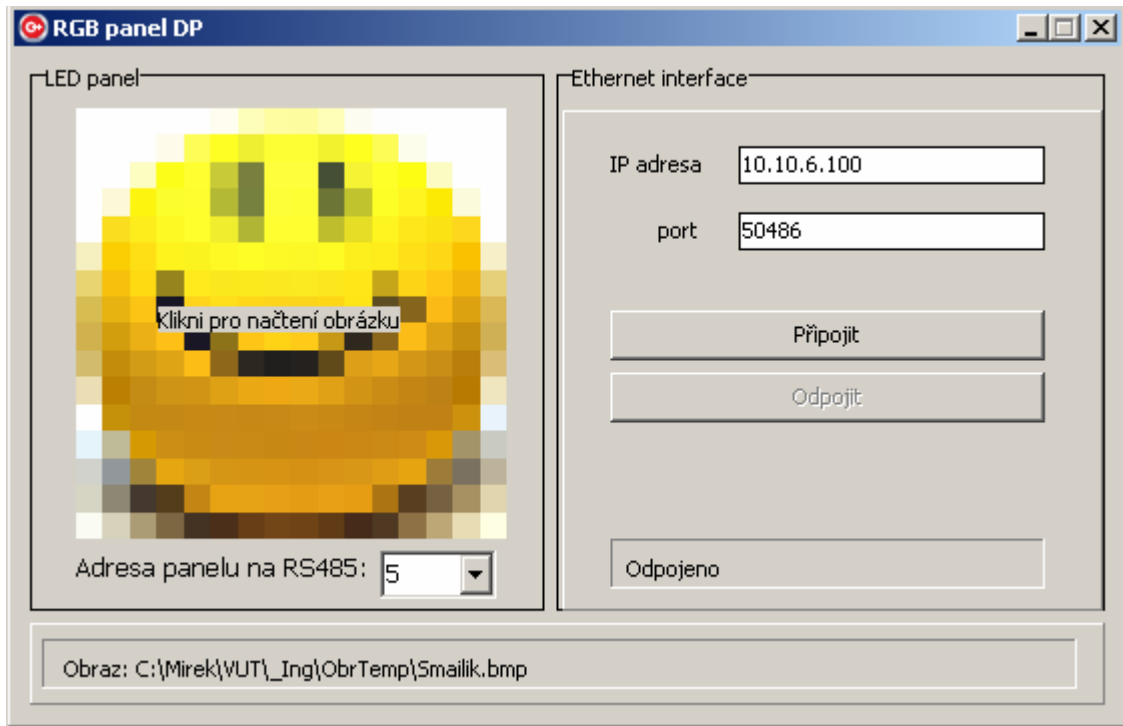
Pro otestování komunikace s PC a nastavování obrazu na navrženém zobrazovacím panelu jsem vytvořil program `RGBpanelDP`. Pro sestavení této aplikace jsem použil vývojový prostředek C++ Builder XE3 firmy Embarcadero Technologies Inc. . Toto vývojové prostředí dovoluje rychlé sestavení programu pomocí vkládání komponent z VCL (visual component library) a následnou úpravy kódu pomocí vlastních knihoven nebo přímé úpravy. C++ Builder vychází z populárního prostředí Delphi studio zaměřeného na objektové programování v pascalu. Také komponenty VCL jsou prakticky totožné jako v Delphi.

K projektu `RGBpanelDP` jsou kromě základní jednotky `Unit1` se soubory `Unit1.cpp`, `Unit1.h` a `Unit1.dfm` přidány jednotky `Panel` a `TCP_client`, které obsahují některé proměnné a funkce pro konverzi dat do formátu vhodného pro přenos do zobrazovacího panelu.

7.2.3.1 Funkce programu a jeho ovládání

Okno výše zmíněného programu pro testování navrženého zapojení je zobrazeno na obr.30. Vlevo je pole pro načtení obrazu o velikosti 16x16 bodů. Při kliknutí na toto pole se otevře menu pro výběr souboru s příponou `bmp`. Výběr je filtrem omezen pouze na tento typ souborů. Vybraný soubor typu `windows bitmap` se zobrazí v poli `LED Panel`. Pokud je vybrán soubor s větším počtem pixelů je načtena pouze jeho část o velikosti 16x16 bodů. V dolní části okna aplikace se zobrazí název souboru a adresářová struktura jeho umístění. Pod polem pro načtení obrazu je ještě umístěn rolovací seznam, který slouží k výběru adresy zobrazovacího modulu, do kterého se má obraz zasílat. V pravé části je panel pro obsluhu komunikace po ethernetu. Adresa a port na kterém očekává spojení interface s `Rabbit RCM 3200` je zadávána do polí `IP adresa` a `port`. Pokud jsou tato pole prázdná jsou ke spojení použity hodnoty `IP adresa 10.10.6.100` a `portu 50486`. V dolní části panelu „Ethernet interface“ je umístěno pole, které zobrazuje stav připojení nebo poruchu. Stiskem klávesy „Připojit“ se aktivuje TCP klient a proběhne „handshake“ mezi PC aplikací a modulem `Rabbit`. Pokud je

spojení navázáno, začne TCP klient v časových intervalech zasílat načtený obraz do modulu RCM3200. Ten ho přijme, uloží a odpoví zpět.



Obr.30 – Program RGBpanelDP pro testování navrženého zapojení

7.2.3.2 Načtení barev obrazu

Obraz, se kterým program pracuje, je načten ze souboru pomocí metod komponenty TImage. Tato komponenta uchovává obraz v objektu TPicture. Pokud je načtený obraz typu bitmap, pak jeho grafiku specifikuje vlastnost Bitmap. Pro načtení obrazového souboru do komponenty TImage, změnu velikosti na 16x16 bodů, zobrazení cesty obrázku v dolní části okna a volání funkce pro načtení barev jsem použil následující kód:

```
if (OpenDialog1->Execute())
{
    Image2->Picture->LoadFromFile(OpenDialog1->FileName);
    Label3->Caption = "Obraz: " + OpenDialog1->FileName;
    Image2->Picture->Bitmap->SetSize(16,16);
    Image2->Repaint() ;
    TImToBufArr(Image2,PanelArr[PanelAdrAct],PanelAdrAct);
}
```

Tato sekvence je volána ve funkci obsluhující událost kliknutí na obrázek respektive na zobrazenou část komponenty TImage.

Funkce TImToBufArr(Image2, PanelArr[PanelAdrAct], PanelAdrAct), která načte barvy z obrazu uchovávaného v TImage je umístěna v souboru Panel.cpp a je definována takto:

```
int TImToBufArr(TImage *aImage, TByteDynArray & aPanel, unsigned char
aPanelAdr)
{
    TColor barva;
    int x,y,xmax,ymax;
    int radek, sloupec;
    int radekmax = 15;
    int sloupecmax = 15;
    int index;
    xmax = aImage->Picture->Bitmap->Width;
    ymax = aImage->Picture->Bitmap->Height;

    aPanel.set_length(773);

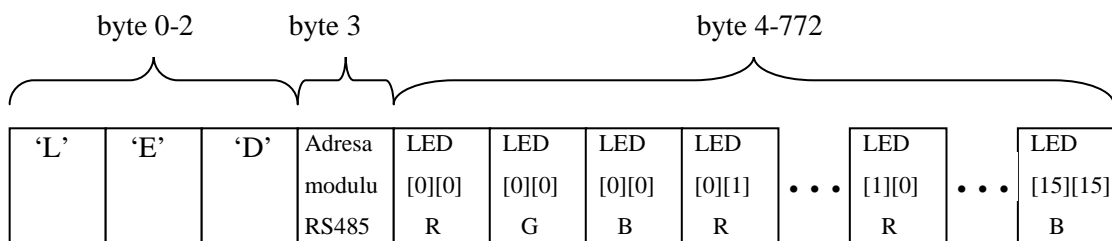
    aPanel[0]='L';
    aPanel[1]='E';
    aPanel[2]='D';
    aPanel[3]=aPanelAdr;
    index=3;

    for(sloupec=0;sloupec<=sloupecmax;sloupec++)
    {
        for(radek=0;radek<=radekmax;radek++)
        {
            barva = aImage->Canvas->Pixels[radek][sloupec];
            index++;
            aPanel[index] = (unsigned char)((barva & 0x0000FF)*31/255);
            index++;
            aPanel[index] = (unsigned char)(((barva & 0x00FF00)>>8)*31/255);
            index++;
            aPanel[index] = (unsigned char)(((barva & 0xFF0000)>>16)*31/255);
        }
    }
    return(1);
}
```

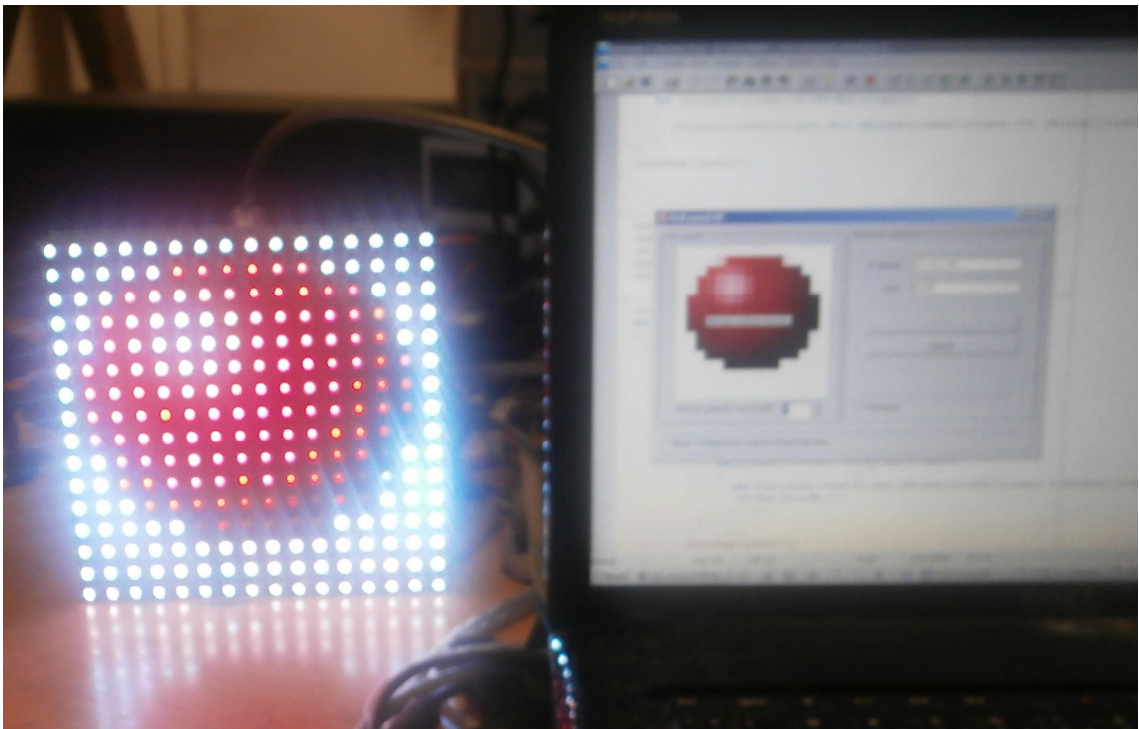
Ke zjištění hodnoty barvy využívá funkce přiřazení
`barva = aImage->Canvas->Pixels[radek][sloupec];`,
která proměnné typu `TColor` přiřadí číslo specifikující barvu pixelu daného souřadnicemi `radek` a `sloupec`. Maximální rozsah proměnné typu `TColor` je `0x00FFFFFF`. Z toho dva nejnižší byty jsou rezervovány pro hodnotu rudé složky barvy, třetí a čtvrtý byte je hodnotou zelené složky a pátý a šestý byte nesou informaci o hodnotě modré složky. Načtení barvy je tedy provedeno maskováním příslušných bitů a rotací výsledného čísla. Vynásobením zjištěné hodnoty číslem `31/255` a přetypováním na `unsigned char` je hodnota převedena na formát vhodný pro zobrazovací modul. Pro uložení a přenos celého obrazu jsem byl nucen použít typ `TByteDynArray`, který využívá metoda objektu `TCP` klient. Typ `TByteDynArray` je proměnná vytvořená v pascalu pro práci s bytovými dynamickými poli.

7.2.3.3 Komunikace po síti ethernet

Pro vytvoření `TCP` klienta jsem použil komponentu `TIdTCPClient`. Tento objekt umožňuje rychle implementovat ethernetovou komunikaci do uživatelské aplikace. V programu jsou využity především jeho funkce `IdTCPClient1->Connect()`, `IdTCPClient1->Disconnect()`. V události vyvolávané časovačem je pak použita funkce `IdTCPClient1->IOHandler->WriteDirect(PanelArr[PanelAdrAct],PanelArr[PanelAdrAct],Length,0)`, která odesílá dynamické bytové pole s daty pro příslušný zobrazovací modul. Tato funkce je předcházena funkcí `IdTCPClient1->IOHandler->WriteBufferClose()`. Ta zajistí vyprázdnění bufferu, pokud by obsahoval nějaká neodeslaná data. Pro zjištění přítomnosti dat odesílaných serverem je využita funkce `IdTCPClient1->IOHandler->CheckForDataOnSource(100)`. Její parametr je časový interval v ms, po který se má čekat na data (timeout). Samotné přijetí dat je zabezpečeno funkcí `IdTCPClient1->IOHandler->ReadBytes(bufInTarr,35,false)`, pomocí níž je přečteno 35 bytů a přečtená hodnota proměnné `bufInTarr`. Složení pole odesílaného zobrazovacímu modulu je naznačeno na obr.31.



Obr.31 - Složení pole pro odeslání obrazových dat



Obr.32 – testování zobrazovacího modulu

8 ZÁVĚR

V této práci jsem navrhnul, vytvořil a odzkoušel RGB LED modul, který je možné využít jako segment většího zobrazovacího panelu. Tento panel může sloužit jako reklamní či informační prostředek.

Navržené moduly budou řízeny z centra (PC). Data pro zobrazování budou zasílána pomocí ethernetu. Velikost jednoho modulu jsem zvolil 16x16 RGB LED s roztečí 10 mm. Komunikace s PC a rozesílání dat pro zobrazení jednotlivým částem displeje bude uskutečněna pomocí desky s mikrokontrolérem. K tomuto účelu byl vybrán vývojový modul RCM 3200, osazený mikroprocesorem Rabbit 3000, ethernetovým řadičem a také konektorem RJ45 včetně oddělovacích transformátorů a ochranného rezistoru. Spínání led zobrazovacího modulu a příjem dat z RCM 3200 bude řídit mikroprocesor atmega 2560-16AU, který je cenově přijatelný, disponuje dostatečnými paměťovými i výkonovými možnostmi a především dostatečným počtem vstupně výstupních portů.

V první části této práce se věnuji teoretickému úvodu týkajícímu se reprezentace barev ve výpočetní technice, skládání barev v aditivním modelu RGB, řízení barev, zobrazovaných RGB LED diodami a možnostem komunikace mikrokontroléru pomocí ethernet a RS485. Také zde uvádím příklad řešení RGB zobrazovacího panelu. Ve druhé části se věnuji návrhu a zprovoznění vlastního modulu. Vysvětluji zde koncepci návrhu a výběr součástek, uvádím stručný popis zapojení. Popisují zde také software, který jsem v průběhu vypracování této diplomové práce vytvořil. Jedná se o program pro řízení zobrazovacího modulu pomocí mikrokontroléru Atmega 2560-16AU, program pro rozesílání dat zobrazovacím modulům a ethernetovou komunikaci s PC pomocí programovatelného modulu Rabbit RCM3200 a také o aplikaci pro PC, kterou je možné odesílat obrazová data načtená ze souboru typu bmp do zobrazovacího panelu.

Funkci navrženého modulu jsem odzkoušel pomocí zmíněné aplikace pro PC. Příjem i zobrazení obrazu je uspokojivé a zařízení fungovalo bez komplikací či závad. Bohužel byl vytvořen pouze prototyp jednoho zobrazovacího modulu, proto nebylo možné vyzkoušet zobrazení většího motivu na rozsáhlejší panelu.

Dalším krokem pro možné využití návrhu v praxi je tvorba software, který by dovozoval komfortní obsluhu celého panelu včetně konfigurace počtu modulů a jejich rozložení. Důležitá je také otázka uzpůsobení napájecího napětí, tedy vytvoření vhodného napájecího zdroje a v neposlední řadě také dořešení krytí a mechanické montáže modulů.

Literatura

- [1] Digi International Inc.: *RabbitCore RCM3200 C-programmable modul with ethernet User's manual* [online]. ©2002–2008 [cit.2013-5-18]. Dostupné na URL: <http://ftp1.digi.com/support/documentation/0190118_n.pdf>
- [2] Digi International Inc.: *Schematic diagram RCM3200 series modul* [online]. ©2002–2008 [cit.2013-5-18]. Dostupné na URL: <http://ftp1.digi.com/support/documentation/0900152_k.pdf>
- [3] Digi International Inc.: *Dinamic C Integrated C Development System for Rabbit 2000 and 3000 Microprocessors* [online]. ©2002–2008 [cit.2013-5-18]. Dostupné na URL: <<http://ftp1.digi.com/support/documentation/019-0125K.pdf>>
- [4] Digi International Inc.: *TCP/IP User's Manual Volume 1* [online]. ©2007–2009 [cit.2013-5-18]. Dostupné na URL: <<http://ftp1.digi.com/support/documentation/0190143j.pdf>>
- [5] Digi International Inc.: *An Introduction to TCP/IP For Embedded System Designers RabbitCore* [online]. ©2006 [cit.2013-5-18]. Dostupné na URL: <http://ftp1.digi.com/support/documentation/0190074_j.pdf>
- [6] Hyder, Kamal a Bob Perin. *Embedded Systems Design Using the Rabbit 3000 Microprocessor*. c2005. ISBN 0-7506-7872-0
- [7] Petyovský, P. *Kurz MPOV – přednáška barevné modely*. [cit.2013-5-18]. Dostupné na URL: <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/01c%20-%20Barevne%20modely.pdf>>
- [8] Fischer, Jan. *LED panely s XMOS – snadnější už to nebude* [online]. 22. Červenec 2009 - 0:00 [cit.2012-10-18]. Dostupné na URL: <<http://www.hw.cz/produkty/led-panely-s-xmos-snadnejsi-uz-to-nebude.html>>
- [9] WayEngineer.com, *Informace o vývojovém modulu STM32F4-V/UET core board*. c 2013 [cit.2013-5-18]. Dostupné na URL: <http://www.wayengineer.com/index.php?main_page=product_info&cPath=50_66&products_id=615>
- [10] Hebei I.T. Co., Ltd. . *SMD LED specification*, Part No.: PLCC4RGBCT-CA
- [11] Atmel corp.. *8-bit Atmel Microcontroller with 64k/128k/256k/Bytes in-system programmable flash* [online]. 2549P–AVR–10/2012. [cit.2013-5-18]. Dostupné na URL: www.atmel.com/images/doc2549.pdf
- [12] Atmel corp.. *Manuál AVR libc* [online]. 2012/09/03 08:23:12 [cit.2013-5-18]. Dostupné na URL: <<http://www.nongnu.org/avr-libc/>>

- [13] Janů, Petr a PetrKašpar. *Síťové aplikace s mikrokontroléry Atmel* [online]. 2012/09/03 08:23:12 [cit. 2013-5-18]
Dostupné na URL: <<http://www.urel.feec.vutbr.cz/MIA/2010/Kaspar/index.html> >
- [14] GEMA s.r.o.: *RGB LED tabule.c* Gemas.r.o. 2010 [cit.2013-5-19]. Dostupné na URL: <http://www.wayengineer.com/index.php?main_page=product_info&cPath=50_66&products_id=615>
- [15] Sigmax.com.hk: *LED Screen cntroller*. [cit.2013-5-19]. Dostupné na URL: <<http://www.signax.com.hk/ledcontroller.html>>
- [16] Elvac a.s.: *Outdoor Fullcolor LED display 3LEDs - P10* [cit.2013-5-19]. Dostupné na URL: <<http://www.elvac.eu/Default.aspx?tabid=388&catid=98798>>
- [17] COUFAL M.: *Ruční terminál pro VN tester*. Bakalářská práce. FEKT VUT v Brně, 2009 [cit.2013-5-19]. 45 s.
- [18] Maxim Integrated: *Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers* [online]. Document Ref.: 19-0122 Rev 9; 2009-10-20 This page last modified: 2010-02-11 [cit.2013-5-19]. Dostupné na URL: <<http://www.maximintegrated.com/datasheet/index.mvp/id/1111>>
- [19] Wikipedia.org. *Ethernet* [online]. 17. 3. 2013 v 13:37. [cit. 2013-5-19]. Dostupné z: <<http://cs.wikipedia.org/wiki/Ethernet>>
- [20] Peterka, Jiří. *Báječný svět počítačových sítí Část XX.: Příběh Ethernetu*[online]. © Jiří Peterka, 2011 [cit. 2013-5-19]. Dostupné na URL: <<http://www.earchiv.cz/b06/b1200001.php3>>
- [21] Ethernet Community. *Ethernut 2* [online]. [cit. 2013-5-19]. Dostupné na URL: <<http://www.ethernut.de/en/hardware/enut2/index.html>>

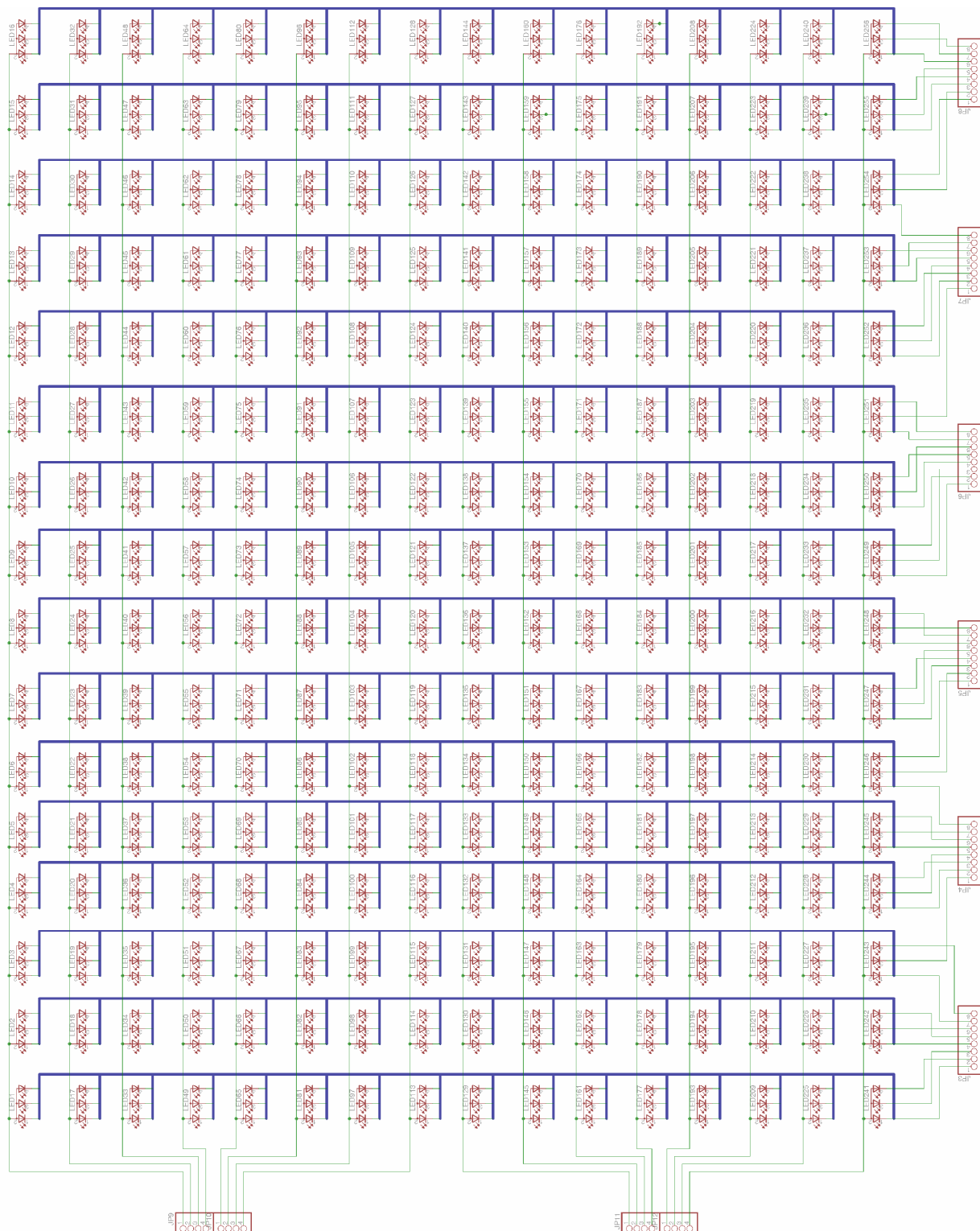
Seznam příloh

Příloha 1. Schéma zapojení

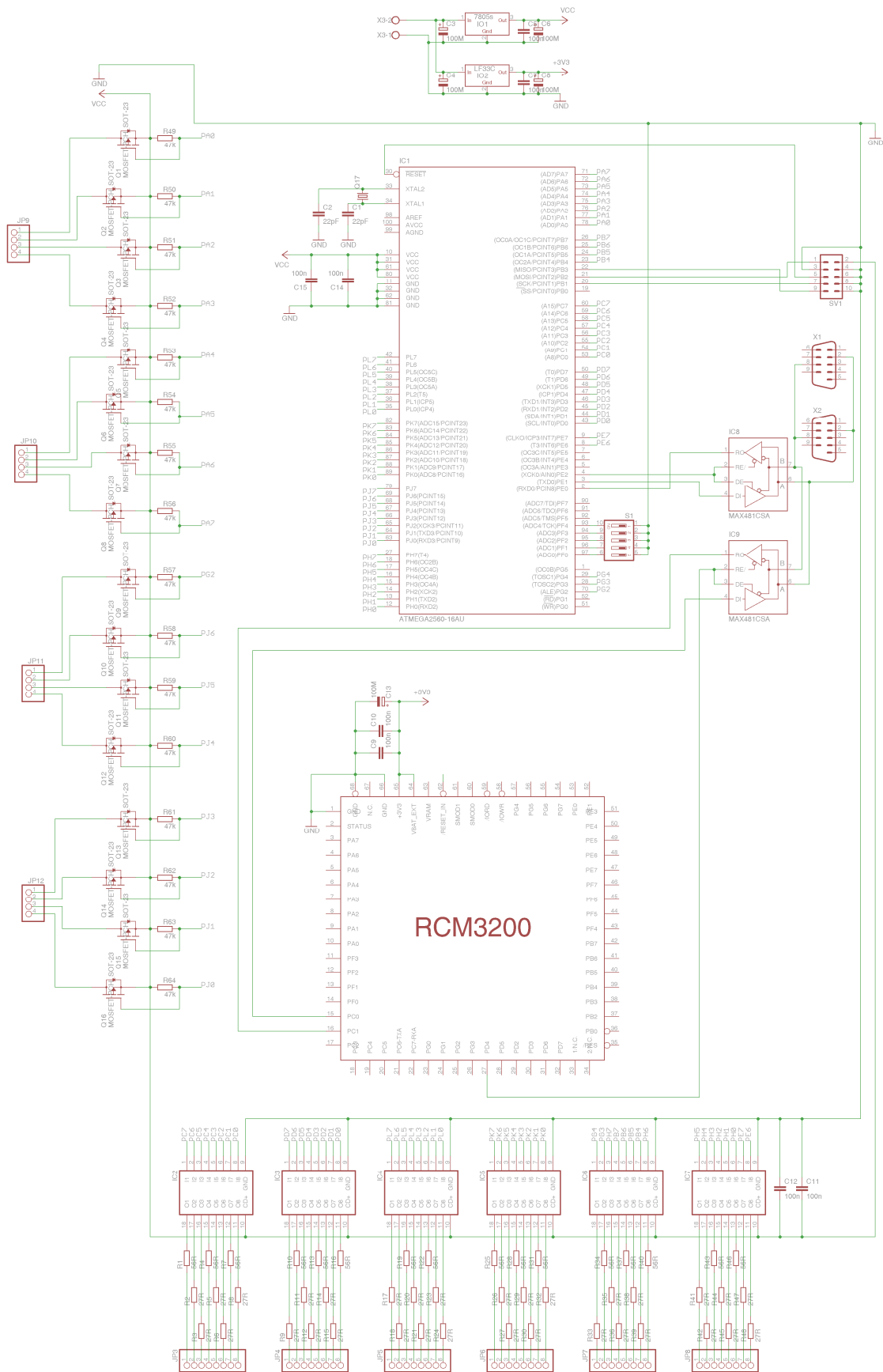
Příloha 2. Návrh desek plošných spojů

Příloha 3. DVD obsahující zdrojové kódy programů, návrh DPS a text diplomové práce

Příloha 1. Schéma zapojení

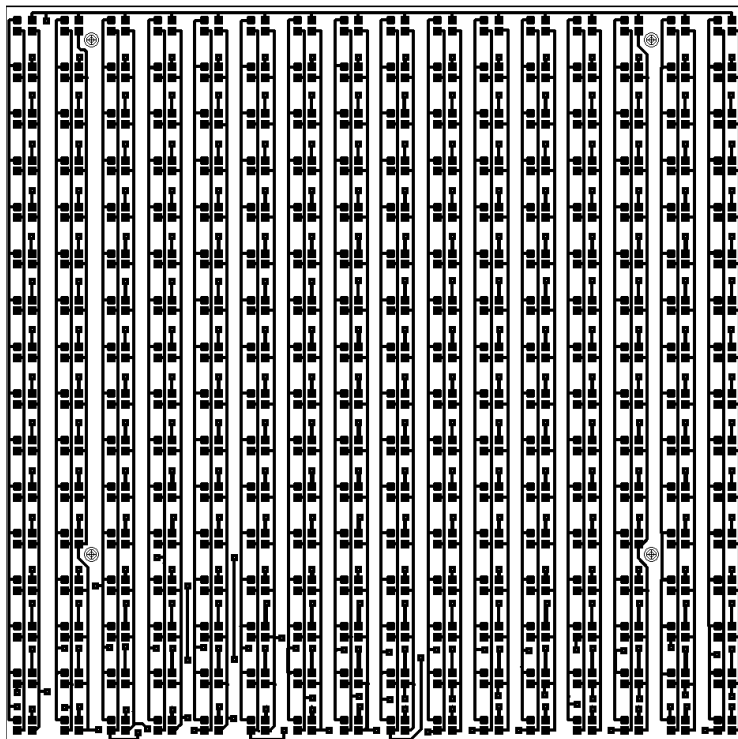


Obr.33 - Schéma zapojení RGB LED matice

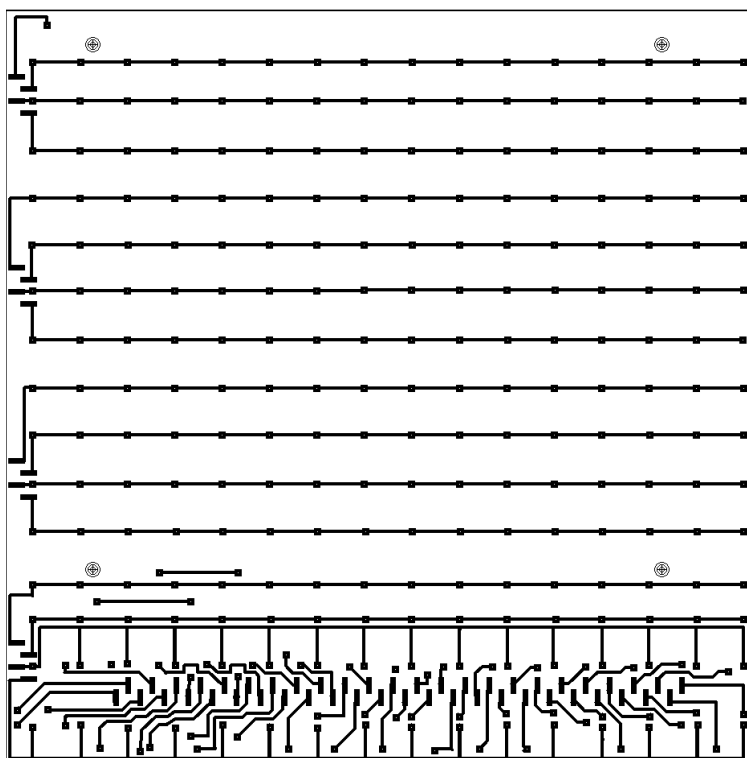


Obr.34 - Schéma zapojení řídicí desky zobrazovacího modulu a připojení Rabbit RCM3200

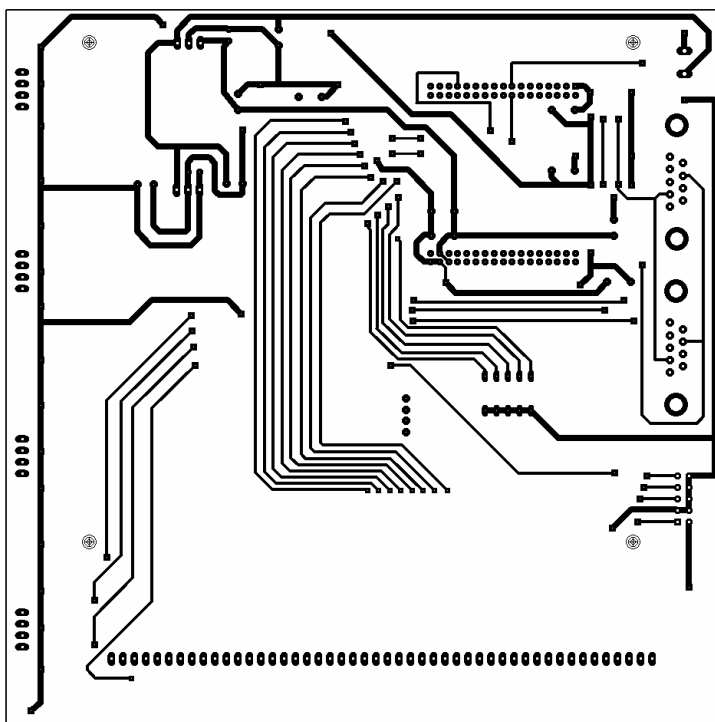
Příloha 2. Návrh desek plošných spojů



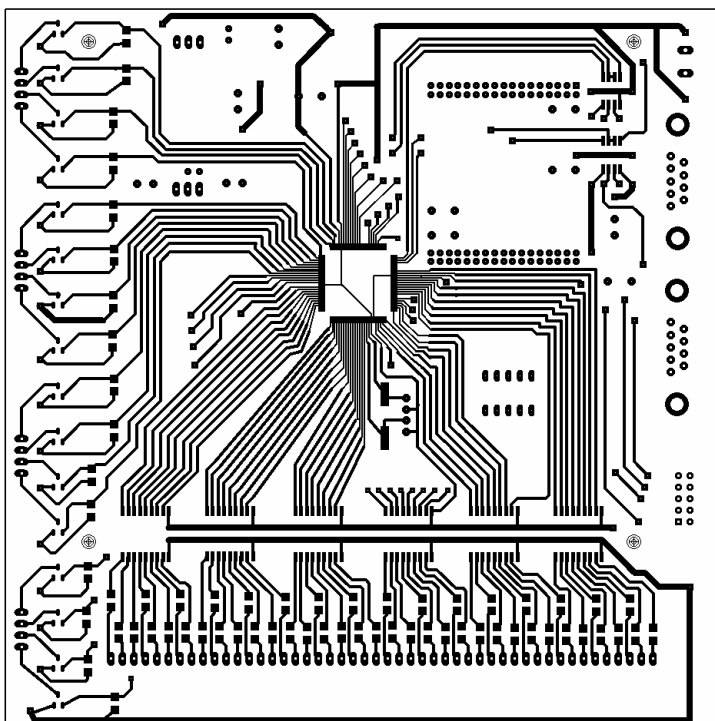
Obr.35 - Deska LED matice – strana top



Obr.36 - Deska LED matice – strana bottom



Obr.37 - Deska řízení LED – strana top



Obr.38 - Deska řízení LED – strana bottom