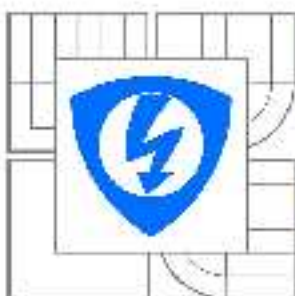


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

REALIZACE MINIATURNÍCH PC MONITORŮ

MINIATURE PC MONITOR REALIZATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAROSLAV NEČESANÝ

VEDOUcí PRÁCE
SUPERVISOR

doc. Ing. TOMÁŠ FRÝZA, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Jaroslav Nečesaný

ID: 72791

Ročník: 2

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Realizace miniaturních PC monitorů

POKYNY PRO VYPRACOVÁNÍ:

Rozeberte problematiku realizace 2 miniaturních PC monitorů ve verzích: (a) USB monitor, (b) VGA(HDMI) monitor. Zvažte možnosti doplnění monitoru o zařízení HID (human interface device). Navrhněte blokové schéma a schéma zapojení pro oba monitory.

Navrhněte DPS miniaturního USB PC monitoru ve vhodné konstrukční třídě a s ohledem na konstrukční možnosti. Desku plošných spojů osadte a oživte a vsadte do krabičky s ohledem na finální vzhled zařízení. Vytvořte program pro obvod CPLD, který upravuje formát dat pro TFT displej.

Vytvořte PC software v jazyce Microsoft Visual C#, který bude do monitoru odesílat obrazová data a umožní řízení periferií monitoru. V souladu s PC softwarem vytvořte program pro řídicí procesor monitoru a jejich vzájemnou spolupráci odlaďte. Proveďte testování monitoru a posuďte jeho technické i uživatelské parametry.

DOPORUČENÁ LITERATURA:

[1] FT2232H. FTDI - Future Technology Devices International Ltd., USB devices [online]. 2010 - [cit. 17. prosince 2010]. Dostupné na [www: www.ftdichip.com/Products/ICs/FT2232H.htm](http://www.ftdichip.com/Products/ICs/FT2232H.htm).

[2] AD9398. Analog devices - HDMITM Display Interface [online]. 2010 - [cit. 17. prosince 2010].

Dostupné na [www:](http://www.analog.com/en/audiovideo-products/analoghdmidvi-interfaces/ad9398/products/product.html)

<http://www.analog.com/en/audiovideo-products/analoghdmidvi-interfaces/ad9398/products/product.html>.

Termín zadání: 6.2.2012

Termín odevzdání: 18.5.2012

Vedoucí práce: doc. Ing. Tomáš Frýza, Ph.D.

Konzultanti diplomové práce:

prof. Dr. Ing. Zbyněk Raida

Předseda oborové rady

Výzkum realizovaný v rámci této diplomové práce byl finančně podpořen projektem
CZ.1.07/2.3.00/20.0007 **Wireless Communication Teams**
operačního programu **Vzdělávání pro konkurenceschopnost**.



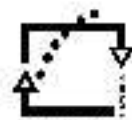
evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Finanční podpora byla poskytnuta Evropským sociálním fondem
a státním rozpočtem České republiky.

Tento příspěvek vzniknul za podpory projektu CZ.1.07/2.3.00/20.0007 WICOMT,
financovaného z operačního programu **Vzdělávání pro konkurenceschopnost**



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

ABSTRAKT

Úlohou Diplomovej práce bolo navrhnuť dve riešenia miniatúrneho PC monitoru. Jeden monitor využíva ako zdroj vstupných dát signál VGA, druhý je pripojený pomocou USB. K praktickej realizácii bol vybraný USB monitor. Zariadenie obsahuje okrem zobrazovacej jednotky LCD čítačku SD kariet a audio prevodník s reproduktorom. Programové vybavenie umožňuje sledovanie obsahu pracovnej plochy osobného počítača v nastaviteľnej polohe a veľkosti. Monitor je možné obsluhovať pomocou touchscreen. Súčasťou riadiaceho programu je jednoduchý prehrávač videa.

KLÚČOVÉ SLOVÁ

Monitor, USB, VGA, USB hub, čítačka kariet, audio adaptér, LCD.

ABSTRACT

Objective of the Master Thesis was to bring in two solutions of miniature PC display. One of them uses the VGA signal as the source of the data. The second uses USB for communication. Display using USB was chosen to be produced. Device contains LCD screen, SD card reader and audio DAC with mini speaker. Software running on the PC allows user to watch part of the screen of the computer. Position and size of the chosen part may be varied. The display may be controlled by touch screen. One part of the software running on PC is a simple video player.

KEYWORDS

Monitor, USB, VGA, USB hub, card reader, audio adapter, LCD.

NEČESANÝ, J. Realizace miniaturních PC monitorů. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 45 s, 13s příloh. Vedoucí diplomové práce: doc. ing. Tomáš Frýza, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Realizace miniaturních PC monitorů 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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce doc. Ing. Tomášovi Frýzovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne

.....

(podpis autora)

OBSAH

Obsah	vii
Úvod	1
1 USB monitor	2
1.1 Prehľad kľúčových komponentov a bloková schéma.....	2
1.2 Schéma zapojenia USB monitoru	3
2 VGA monitor	5
2.1 Prehľad kľúčových komponentov a bloková schéma.....	5
2.2 Schéma zapojenia VGA monitoru	7
3 Návrh DPS USB monitoru	9
4 Osadenie a oživenie monitoru	12
5 Softwarové vybavenie monitoru	16
5.1 Program hradlového poľa XC9536	16
5.2 Program riadiaceho mikrokontroléru HCS08.....	17
5.3 Komunikácia mikrokontroléru s riadiacim programom	22
5.4 Nastavenie a ovládanie prevodníku FTDI	26
5.5 Riadiaci program – príprava a odosielanie dát	30
5.6 Užívateľské rozhranie riadiaceho programu	33
5.7 Nadstavba riadiaceho programu – video prehrávač	38
6 Záver	41
Literatúra	43
Zoznam skratiek	44
Zoznam príloh	45

ÚVOD

Úlohou diplomovej práce je navrhnuť dva miniatúrne PC monitory. V prvom prípade pôjde o monitor zobrazujúci dáta z VGA výstupu počítača. Druhý monitor bude mať ako zdroj obrazových dát USB zbernicu. Súčasťou návrhov budú doplnkové zariadenia ako USB audio prevodník, touchscreen controller displeja, akcelerometer pre rotáciu obrazu podľa aktuálneho otočenia displeja a podobne. V prípade USB monitora bude zariadenie doplnené o čítačku SD kariet. Realizácia USB vyžaduje vytvorenie PC programu. Preto bol k realizácii vybraný práve tento monitor. Text je rozdelený do šiestich kapitol. V úvode bude rozobratá problematika a úskalia oboch návrhov. Následne bude navrhnutá bloková schéma a schéma zapojenia oboch zariadení. V tretej časti textu bude popísaná realizácia USB monitora. Text bude obsahovať postup pri návrhu dosky plošných spojov a následného osadzovania vyhotovenej dosky. V predposlednej kapitole je popísané softwarové vybavenie USB monitora. Text obsahuje podrobný popis programu riadiaceho mikrokontroléru HCS08, popis PC programu a ich vzájomnú komunikáciu. V závere práce sú zhrnuté dosiahnuté výsledky a základné technické parametre USB monitoru.

1 USB MONITOR

V tejto kapitole je popísaná prvá časť projektu, ktorá sa zaoberá návrhom USB monitoru a prípravou riešenia projektu.

1.1 Prehľad kľúčových komponentov a bloková schéma

Kľúčovým komponentom monitora je zobrazovacia jednotka - displej. Do USB monitora bol vybraný LCD modul s kontrolérom HX8347-A predovšetkým pre jeho dobrú dostupnosť v internetových obchodoch a prijateľnú cenu. Ide o LCD s rozlíšením 320 x 240 pri 16 bitovej farebnej hĺbke, osadený na DPS o rozmeroch 94x64mm.

Objem dát prúdiacich do tohto displeja pri obnovovacej frekvencii 30Hz vysoko prekračuje možnosti USB - full speed. Dátový tok obrazového signálu je 13,2MB/s. Preto bolo nutné siahnuť po USB obvodoch podporujúcich rozhranie USB - high speed. Nové prevodníky od firmy FTDI Ltd. túto požiadavku spĺňajú. Vybraný bol obvod FT2232H[1], ktorý v sebe integruje 2 na sebe nezávislé USB - high speed prevodníky.

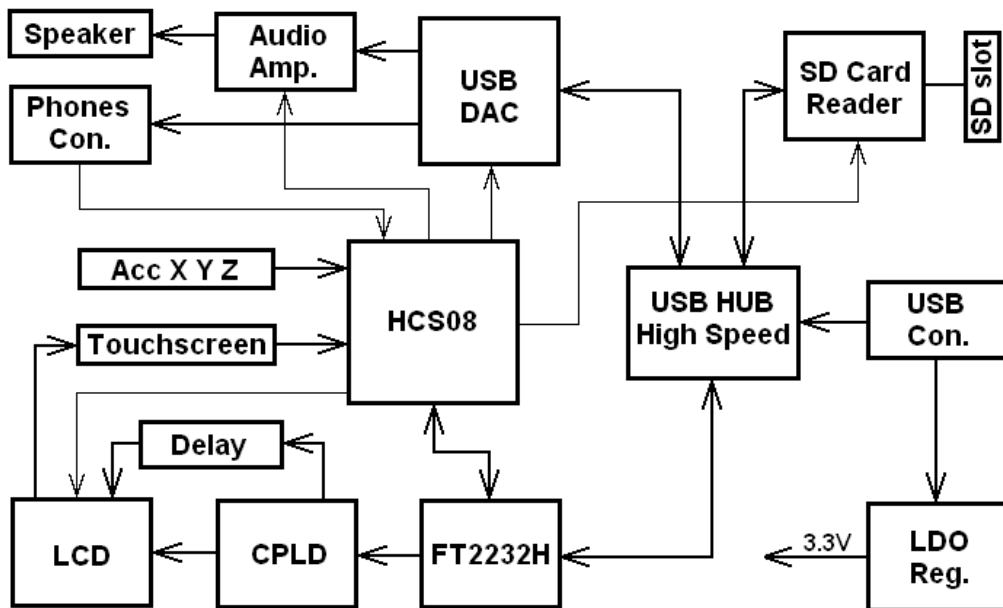
Pre realizáciu zvukového výstupu monitora bol vybraný USB Audio DAC od firmy Texas Instruments s označením PCM2705[3]. Dôvodom výberu boli predovšetkým dobré skúsenosti s týmto obvodom z predchádzajúcich konštrukcií. Tento obvod je kompletným riešením USB audio prevodníkom, čo značne zjednodušuje vývoj celého zariadenia.

V prípade, že používame viac ako jedno USB zariadenie a nechceme užívateľa obťažovať viac ako jedným USB káblom, je nutné zaradiť do zapojenia USB hub. Ten musí byť, rovnako ako prevodník, štandardu USB - high speed. Hub s označením ISP1520[4] bol vybraný najmä kvôli dobrej dostupnosti v elektronických obchodoch.

Ďalším doplnkovým zariadením v rámci miniatúrneho USB monitoru je čítačka SD kariet. Jej realizácia vychádza z referenčného návrhu Freescale semiconductor pre

obvod HCS08JM60[6]. V monitore je použitá konštrukcia bez väčších zmien, nie je pre to predmetom práce a bude v texte spomenutá len okrajovo.

Bloková schéma na obrázku 1 naznačuje prepojenie jednotlivých kľúčových komponentov zariadenia.



Obrázok 1. Bloková schéma USB monitoru

1.2 Schéma zapojenia USB monitoru

Kompletnú schému zapojenia je možné nájsť v prílohe A1 a A2. Celé zariadenie je napájané priamo z PC cez USB konektor. Počítačom dodané napätie 5V je znížené na 3,3V lineárnym stabilizátorom REG104-3.3 od firmy Texas Instruments. Ten je schopný dodať do záťaže prúd maximálne 1A, čo pre potreby monitora postačuje. Celkový prúdový odber zariadenia je približne 200 mA a závisí na intenzite podsvietenia. Limit pre používanie USB pre napájanie zariadenia je pri tom 500 mA. Nežiaduce rušenie je potlačované filtračnými kondenzátormi na vstupe aj výstupe stabilizátora.

Rozbočovač USB označený ako ISP1520BD[4] je pripojený k napájacímu napätiu 3,3V, vnútorné obvody referenčného napätia sú napájané napätím 5V. K pinom Upstream USB portu je pripojený konektor USB mini a na 3 down-stream porty sú pripojené 3 zariadenia USB monitoru. Posledný port ostáva deaktivovaný. Zdrojom hodinového signálu pre USB hub je kryštál s frekvenciou 12 Mhz. Pre každý port je v hube vypnutá "over current detection" a porty sú nastavené ako "removable". Na pin 31-RESET je pripojený obvod, ktorý vždy po pripojení napájacieho napätia obvod resetuje. K USB hubu je cez linku IIC pripojená pamäť EEPROM. V nej môžu byť nahrané vendor ID, product ID a string descriptor.

K down-stream portu 2 USB hubu je pripojený prevodník FT2232H[1]. Ide o prevodník z USB na niekoľko typov rozhraní vrátane napr. SPI alebo JTAG. Je napájaný napätím 3,3V, pričom vnútorný regulátor vytvára napätie 1,8V, ktorým je napájané jadro obvodu. Ku každému napájacímu pinu je pripojený filtračný kondenzátor, ktorý eliminuje rušenie šíriace sa po vodičoch napájania. Zdrojom hodinového signálu prevodníku je kryštál s frekvenciou 12MHz. K prevodníku je zároveň pripojená EEPROM, v ktorej si prevodník uchováva nastavenie o použití ovládača D2XX, vendor ID, product ID a podobne. Výstup z prevodníka pozostáva z 8 dátových bitov, signálu Write Strobe, signálu RESET, signálu RS a signálov RX a TX pripojených k riadiacemu mikrokontroléru. Signály RESET a RS sú pripojené priamo na LCD a vychádzajú zo signálov RTS a DTR. Ide o signály, ktoré sú pôvodne určené k riadeniu toku dát.

Dátové signály sú privedené do CPLD Xilinx XC9536[5]. Pomocou signálu Write Strobe sú 8 bitové dátové slová spájané do 16 bitových a v tejto podobe sú privedené na výstup CPLD odkiaľ sú privedené na vstupy LCD. Schéma logickej funkcie, ktorá je v CPLD vykonávaná je uvedená na obrázku 9 v kapitole 5.1. Výstupný signál Write Strobe upravený pre 16 bitové dáta je privedený do LCD cez oneskorovací obvod TPS1100. Ten zabezpečí oneskorenie Write Strobe impulzu o 10ns, čím kompenzuje oneskorenie toku dát na hradlách CPLD.

Tretí port USB hubu je pripojený k obvodu PCM2705[3], čo je USB DAC od firmy Texas Instruments. Napájacie napätie 5V je do obvodu vedené cez spínací tranzistor, ktorý je riadený z riadiaceho mikrokontroléru. Obvod využíva vlastný regulátor na napätie 3,3V, čím je výrazne zredukované rušenie od ostatných častí USB monitora. Zem audio výstupu je oddelená od zeme digitálnych častí obvodu, čím sa rovnako rušenie výrazne znižuje. Zároveň je pri každom napájanom pine filtračný kondenzátor. Na audio výstup je okrem konektora pre slúchadlá pripojený vstup audio zosilňovača LM386. Ten je tak isto ako DAC napájaný z 5V vetvy cez tranzistor riadený z riadiaceho mikrokontroléru. Na výstup zosilňovača je pripojený miniatúrny reproduktor.

Obsluhu všetkých častí monitoru zabezpečuje mikrokontrolér HCS08, konkrétne MC9S08JM16. Tento je pripojený k druhej časti USB prevodníku FTDI signálmi RXD a TXD, ktorými komunikuje s riadiacim programom v PC. Mikrokontrolér zabezpečuje vyčítavanie dát z akcelerometra MAA3761 a touchscreen controlleru ADS7843. Ďalej ovláda audio obvody, čítačku SD kariet a podsvietenie LCD.

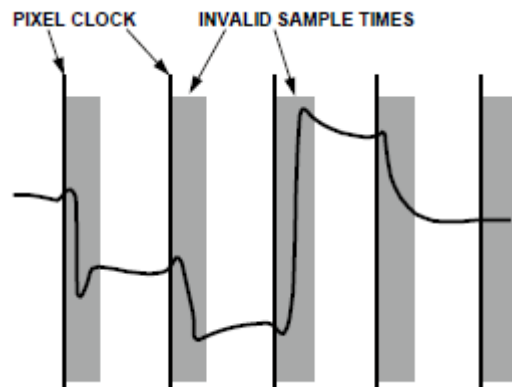
2 VGA MONITOR

Kapitola rozoberá problematiku návrhu blokovej schémy a schémy zapojenia miniatúrneho VGA monitoru.

2.1 Prehľad kľúčových komponentov a bloková schéma

Najkritickejším miestom návrhu VGA monitoru je AD prevodník, ktorý bude prevádzať spojitý analógový video signál na digitálne dáta. Frekvencia video signálu pri rozlíšení 640 x 480 pri obnovovacej frekvencii 60Hz je 25,175MHz. Z toho vyplýva, že použitý AD prevodník musí byť veľmi rýchly. Ideálnym riešením problému je použitie prevodníka AD9985A[2] od firmy Analog devices. Ide o AD prevodník určený priamo na prevod video signálu s maximálnou frekvenciou 140MHz, teda pre veľmi vysoké rozlíšenia. Navyše prevodník obsahuje vlastný, veľmi presný zdroj hodinového signálu, čím odpadá nutnosť návrhu presného oscilátoru. Frekvencia vzorkovania je nastavená

automaticky, podľa vertikálnych a horizontálnych synchronizačných impulzov VGA signálu. Okrem toho, že obvod vzorkuje obrazový signál pri presnej frekvencii, upravuje vzorkovanie tak, aby bola hodnota zosnímaná až po ustálení napätia daného pixlu tak, ako je to znázornené na obrázku 3. Pri takomto snímaní nemusí byť dodržaný vzorkovací teorém a značne sa tým zjednodušuje ďalšie spracovávanie digitálnych dát.

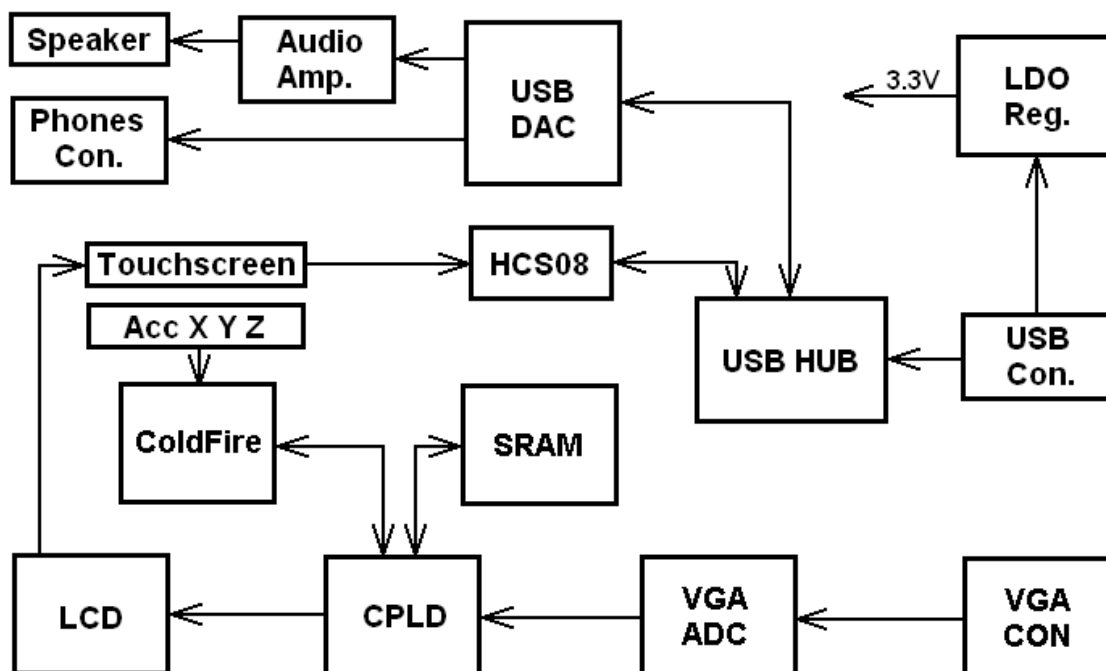


Obrázok 3. – Vzorkovanie video signálu [2].

Najnižšie rozlíšenie podporované počítačovými grafickými kartami je 640 x 480. Vybraný displej má však rozlíšenie 320x240. Obraz je preto nutné zmenšiť a to dvoma možnými spôsobmi. Prvá možnosť je prevzorkovať signál – 4 susedné pixely zpriemerovať, zlúčiť do jedného, a to pre každú farbu zvlášť. Ďalšou možnosťou je urobiť výrez z originálneho obrazu. Pokus o realizáciu oboch týchto metód prebehne v CPLD Xilinx – XC95144XL. Existuje možnosť, že sa zamýšľané úpravy obrazu nepodarí uskutočniť v CPLD z kapacitných dôvodov. V tom prípade bude návrh prepracovaný s použitím FPGA Spartan 3. Dočasné dáta potrebné pre prevzorkovanie musia byť priebežne ukladané, k čomu slúži SRAM IS61LV6416L pripojená k CPLD. K CPLD je pripojený aj 32 bitový mikrokontrolér Cold Fire, ktorý zabezpečuje inicializáciu LCD.

Akcelerometer pripojený k mikrokontroléru sníma údaje o aktuálnom náklone displeja vzhľadom k rovine zeme. Tieto dáta budú využité na prispôsobovanie náklonu obrazu. Ako zobrazovacia jednotka je použitý rovnaký displej ako v USB monitore, a to modul displeja s kontrolérom HX8347-A. Touchscreen kontrolér displeja je pripojený k mikrokontroléru HCS08. Ten je cez USB hub pripojený k PC a bude vybavený softwarom simulujúcim počítačovú myš. K druhému portu USB hubu je pripojený obvod PCM2705, ktorý funguje ako USB DAC. Na jeho výstup je okrem konektoru na

slúchadlá pripojený audio zosilňovač a miniatúrny reproduktor. Bloková schéma VGA monitoru je uvedená na obrázku 4.



Obrázok 4. Bloková schéma VGA monitoru

2.2 Schéma zapojenia VGA monitoru

Kompletná schém zapojenia je uvedená v prílohe B1 a B2. Obrazový signál je privedený na konektor X4. Odtiaľ R,G a B signály vedú 75 ohmové rezistory, na ktorých pretekajúci prúd vytvára napätie snímané AD prevodníkom IC3. Signál je zároveň filtrovaný na tlmivkách. Kvalita tohto obvodu priamo ovplyvňuje výsledný obraz. Okrem video signálov sú k prevodníku privedené aj signály vertikálnej a horizontálnej synchronizácie obrazu HSYNC a VSYNC. K napájacím pinom prevodníka sú pripojené filtračné kondenzátory C49-C58. Video prevodník má na výstupe pre každú farbu k dispozícii 8 bitové dáta. Farebná hĺbka displeja je však len 16 bitov, preto je pre ďalšie spracovanie vybratých len 5 najvýznamnejších bitov z červenej a modrej farby a 6 najvýznamnejších zo zelenej farby. Tieto výstupné piny sú pripojené k CPLD Xilinx, ktorý je v schéme označený ako IC4. K hradlovému poľu je tiež pripojená SRAM (IC5), mikrokontrolér Cold Fire (IC6) a konektor LCD. Keďže

vstupy a výstupy hradlového pola sú ľubovoľne konfigurovateľné, nezáleží na tom, ku ktorým pinom sú spomínané obvody pripojené. Toto pripojenie je možné prispôbiť optimálnejšiemu návrhu DPS. Monitor okrem zobrazovacej časti obsahuje aj dve doplnkové USB zariadenia. Linka USB, ktorá je do monitoru privedená, musí byť rozdelená v USB hube. Pre potreby VGA monitora postačuje použitie USB full - speed hubu TUSB2046 od firmy Texas instruments, ktorý je v schéme označený ako IC1. K up-stream portu hubu je cez rezistory pripojený konektor USB mini. Na prvý využitý down-stream portu je pripojený mikrokontrolér HCS08, ktorý komunikuje s touchscreen controllerom, ktorý je súčasťou modulu LCD. K druhému down-stream portu je pripojený audio adaptér PCM2705 od firmy Texas instruments, ktorý je v schéme označený ako IC11. Zapojenie tohto obvodu je rovnaké ako v predchádzajúcej verzii USB monitoru.

Zariadenie je napájané z USB konektoru cez LDO stabilizátor REG 104, ktorý je v schéme označený ako IC8. Prítomnosť napájacieho napätia 5V a 3,3V signalizujú LED diódy LED3 a LED4.

3 NÁVRH DPS USB MONITORU

Jedným z hlavných požiadavkou na DPS monitoru boli rozmery, spolu s optimálnym rozmiestnením jednotlivých komponentov, najmä konektorov. Ako najlepšie konštrukčné riešenie bolo umiestniť DPS priamo pod modul displeja tak, aby výška modulu displeja a výška DPS bola rovnaká, teda 64mm. Tým celé zariadenie pôsobí ako jednoliaty blok čo prispieva k lepšiemu estetickému dojmu. Šírka DPS presahuje šírku modulu displeja o 7 mm na každej strane, plošný spoj je teda široký 110 mm. Všetky konektory sú umiestnené na pravej strane dosky kvôli lepšej manipulovateľnosti so zariadením. To sa týka hlavne konektoru USB - CON4 a konektoru 3,5 mm JACK - CON1. Umiestnenie konektoru SD karty - SLOT1 na opačnej strane by neprekážalo, nebolo však možné umiestniť ho vľavo. Konektor displeja – CON2 je umiestnený pri ľavom okraji dosky tak, aby bol samotný displej zarovnaný presne do stredu celého zariadenia. Posledným konektorom je programovací konektor – CON3. Zariadenie obsahuje 3 programovateľné obvody. Dva mikrokontroléry Freescale sa programujú cez BDM interface, obvod CPLD od firmy Xilinx sa programuje cez rozhranie JTAG. Štandardné programovacie konektory týchto rozhraní sú príliš veľké a nebolo by možné ich na DPS umiestniť pri zachovaní súčasných rozmerov. Zväčšenie rozmerov monitoru len kvôli programovaciemu interface tiež neprichádzalo do úvahy. Ako vhodné riešenie sa ukázalo vytvoriť vlastný konektor s všetkými programovacími rozhraniami. Tento konektor bol umiestnený k hornému okraju dosky tak, aby bolo možné pripájanie a odpájanie pri nasadenom module displeja. Modul displeja je k DPS pripevnený štyrmi skrutkami M2, montážne otvory sú umiestnené v rohoch modulu, resp. popisovaného plošného spoja s riadiacou elektronikou. Rozmiestnenie súčiastok možno vidieť na osadzovacom výkrese v prílohe C3 – vrstva Top a v prílohe C4 – vrstva Bottom.

Priamo na doske plošných spojov je osadený miniatúrny reproduktor – RE1. Aj keď ide o miniatúrny reproduktor, jeho rozmery sú v pomere k rozmerom celého monitoru značné. Nebolo preto veľa možností kam tento reproduktor umiestniť. Reproduktor je umiestnený v spodnej časti dosky vpravo od konektoru CON2. Kvôli mechanickej pevnosti a odolnosti zariadenia musí byť reproduktor pevne usadený na

DPS. To by samo o sebe malo za následok utlmenie zvuku, preto sú pod membránou vyvítané otvory ktorými sa zvuk voľne šíri. Po rozmiestnení konektorov a reproduktoru ostal na DPS priestor s ktorým bolo nutné nakladať veľmi hospodárne. V duchu úspory priestoru bol lineárny regulátor IC7 umiestnený do vrchnej časti dosky vedľa programovacieho konektoru. Tam je totiž dostatok priestoru pre vytvorenie polygónu pre odvod tepla. Tento polygón sa nachádza na Bottom vrstve plošného spoja a nie je pokrytý ochrannou maskou na rozdiel od zvyšku dosky. Jeho povrch je ochránený cínom pre lepší odvod tepla z jeho povrchu prirodzeným prúdením vzduchu. Pod regulátorom sa nachádzajú prekovy ktorými je teplo z regulátoru odvádzané na chladiaci polygón. Veľmi veľkú pozornosť bolo nutné venovať návrhu audio časti. Tá je totiž najviac náchylná na rušenie šíriace sa z VF častí monitoru, teda USB linky, USB hubu a dátových ciest do displeja.

Audio obvody sú umiestnené v spodnej časti dosky tak, aby vzdialenosť medzi prevodníkom IC11, zosilňovačom IC2 a reproduktorom RE1 bola čo najkratšia. Samozrejmosťou je samostatne vedená vetva GND ku každému z obvodov a rozdelenie zeme na analógovú a digitálnu. V oblasti audio obvodov sú umiestnené tzv. rozliate zeme v Top aj Bottom vrstve spoja. Tie zabezpečujú odtienenie audio obvodov od okolia. Odrušenie je zosilnené ďalšou vetvou GND vedenou okolo celého audio obvodu a tieniacich polygónov. Tým je zabezpečená maximálna ochrana pred rušením. USB hub je umiestnený v prostriedku dosky. Vede doňho USB linka z konektoru CON4. Z neho následne vychádzajú USB linky do ostatných modulov, teda čítačky kariet, audio prevodníka, a prevodníka FTDI. Vedľa obvodu sú umiestnené LED, ktoré signalizujú aktivitu na down-stream portoch hubu. Obvod čítačky SD kariet je umiestnený v pravom hornom rohu priamo pod slotom SLOT1. Obvody FTDI a CPLD sú umiestnené v ľavej hornej časti DPS. Zmyslom tohto rozmiestnenia je úspora priestoru a skrátenie ciest medzi jednotlivými obvodmi na minimálnu mieru. Výkres DPS je uvedený v prílohe C1 – Top vrstva a C2 – Bottom vrstva. Výkresy sú uvedené v merítku 2:1.

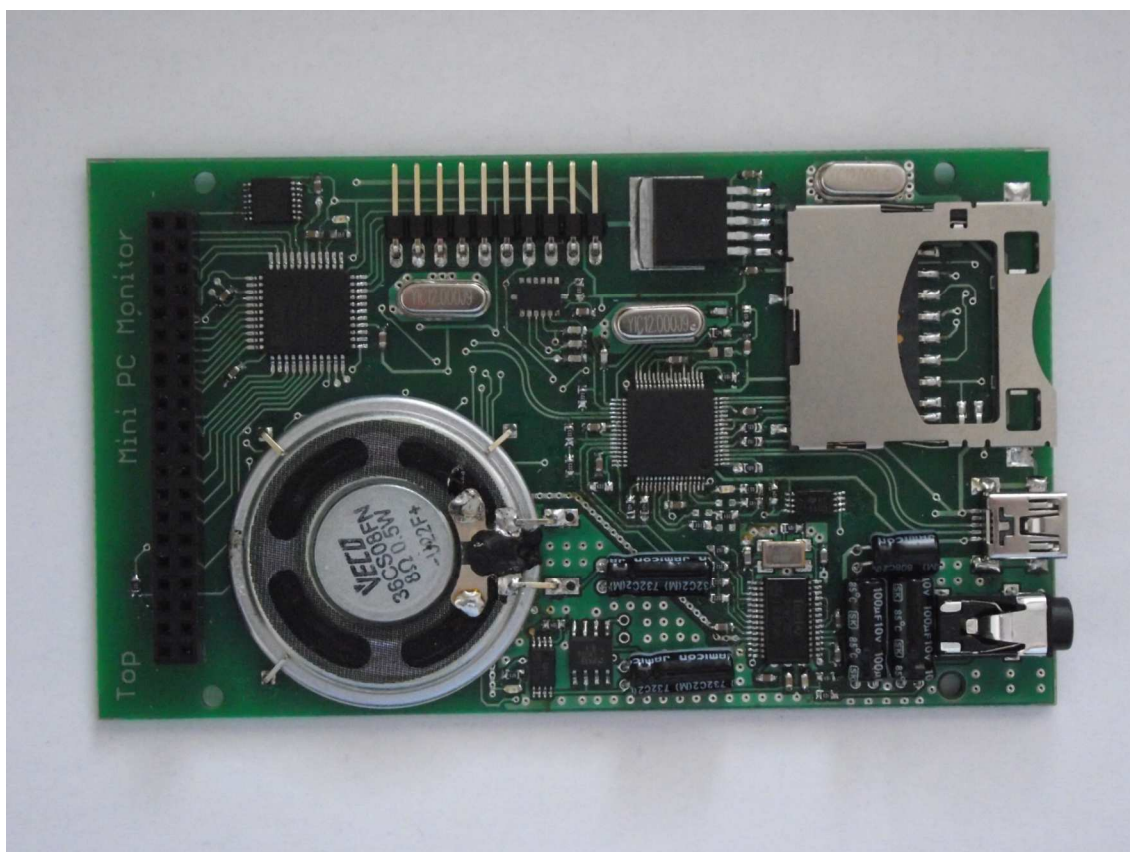
K vytvoreniu návrhu DPS bol použitý program Eagle série 5. To umožnilo využiť nástroje pre kontrolu návrhu a vyvarovať sa tak zbytočného opakovania výroby. Keďže bol plošný spoj navrhnutý v konštrukčnej triede 6, jeho výroba v domácich podmienkach nebola možná. Výroba bola preto zadaná firme MATRON ktorá v rámci

tzv. prototypového programu vyrába dosky plošných spojov v jednotkách kusov.

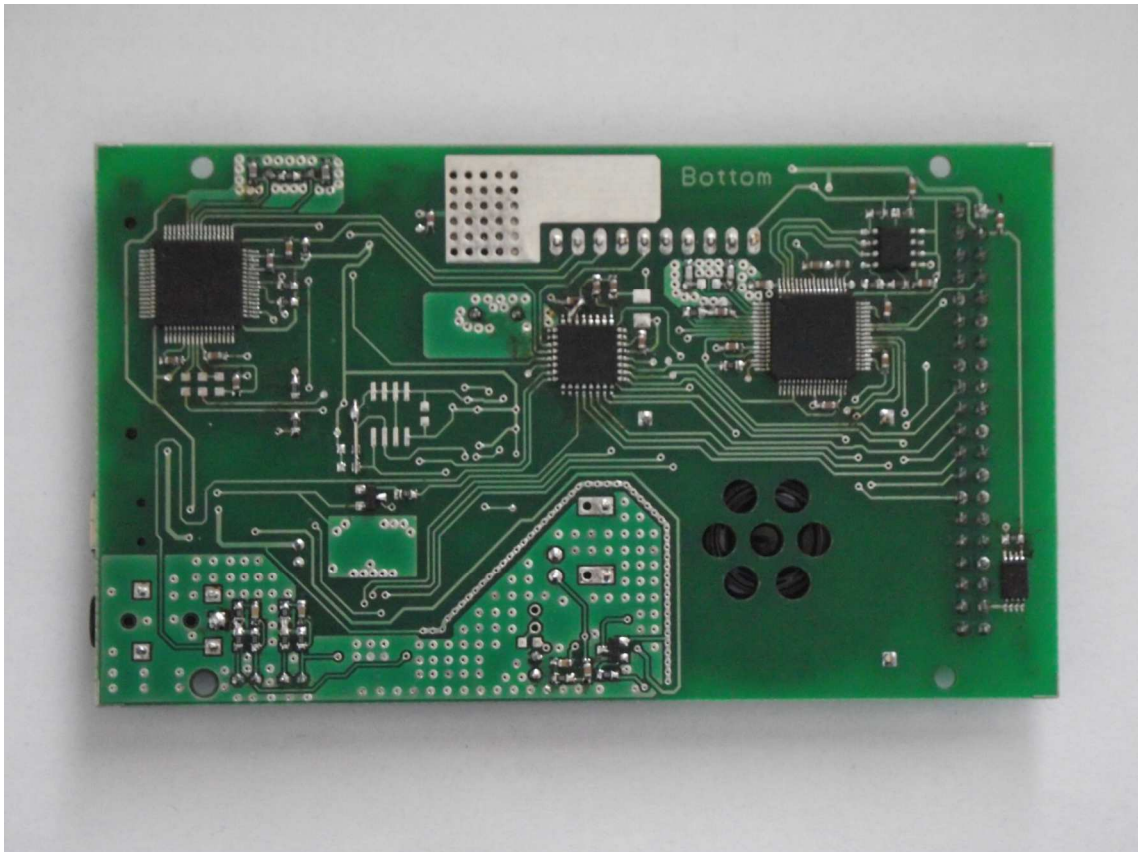
4 OSADENIE A OŽIVENIE MONITORU

Osadzovanie a oživovanie jednotlivých blokov monitoru prebiehalo súčasne. Zabránilo sa tým zbytočnému plytvaniu súčiastkami v prípade, že by na DPS bola nejaká chyba a muselo by byť celé zariadenie vyrobené znova. Zoznam súčiastok je uvedený v prílohe D1. Ako prvý bol osadený lineárny regulátor IC7. Po overení jeho funkčnosti tým bolo zabezpečené napájacie napätie pre ostatné obvody. Ďalším osadeným obvodom bol akcelerometer IC12 a to z toho dôvodu, že puzdro LGA sa v domácich podmienkach osadzuje veľmi komplikovane. Pre prípadnú opravu je teda vhodnejšie, ak na doske ešte ostatné súčiastky nie sú osadené. Akcelerometer sa však podarilo osadiť bez problémov. Ako ďalší obvod bol osadený riadiaci mikrokontrolér HCS08. Po osadení tohto obvodu bola odhalená chyba v návrhu DPS, ktorá pochádzala z chyby v schéme zapojenia. Napájacie napätie a GND obvodu boli vzájomne vymenené a obvod nemohol fungovať. Táto chyba bola odstránená jednoduchým zásahom do zapojenia, bez nutnosti opakovanej výroby plošného spoja. V schéme a v návrhu DPS bola táto chyba taktiež opravená. Po oprave bol do riadiaceho mikrokontroléru IC8 nahraný špeciálny SW ktorý uviedol všetky riadiace signály do požadovaného stavu a umožnil tak ďalšie oživovanie. Ako ďalší bol osadený USB hub so všetkými potrebnými súčiastkami a monitor bol po prvé pripojený k PC. USB hub bol po pripojení bez problémov zdetekovaný, teda plne funkčný. Ďalším osadeným obvodom bola čítačka SD kariet. Po osadení bol do mikrokontroléru IC9 nahraný ostrý SW a funkčnosť čítačky bola následne overená pripojením k PC. Nasledovalo osadenie audio obvodu, teda prevodníka IC11, zosilňovača IC2 a príslušných pasívnych súčiastok. Po osadení prevodníka bola funkčnosť obvodu overená opäť pripojením k PC a výstup na slúchadlá bez problémov fungoval. Po osadení zosilňovača a reproduktoru, bola funkčnosť opäť bezchybná. Posledným krokom bolo osadiť prevodník FTDI a overiť funkčnosť displeja. Po osadení prevodníka IC6 a EEPROM IC4 bol monitor pripojený k PC a muselo prebehnúť nastavenie obvodu. Pomocou programu MPROG bola do EEPROM prevodníku nahraná konfigurácia obvodu pre potreby monitora. Do obvodu CPLD bol následne nahraný SW ktorý bol vytvorený ešte počas pokusov na prvej verzii monitora. Po opätovnom pripojení k PC bol v štúdiu Microsoft Visual C#

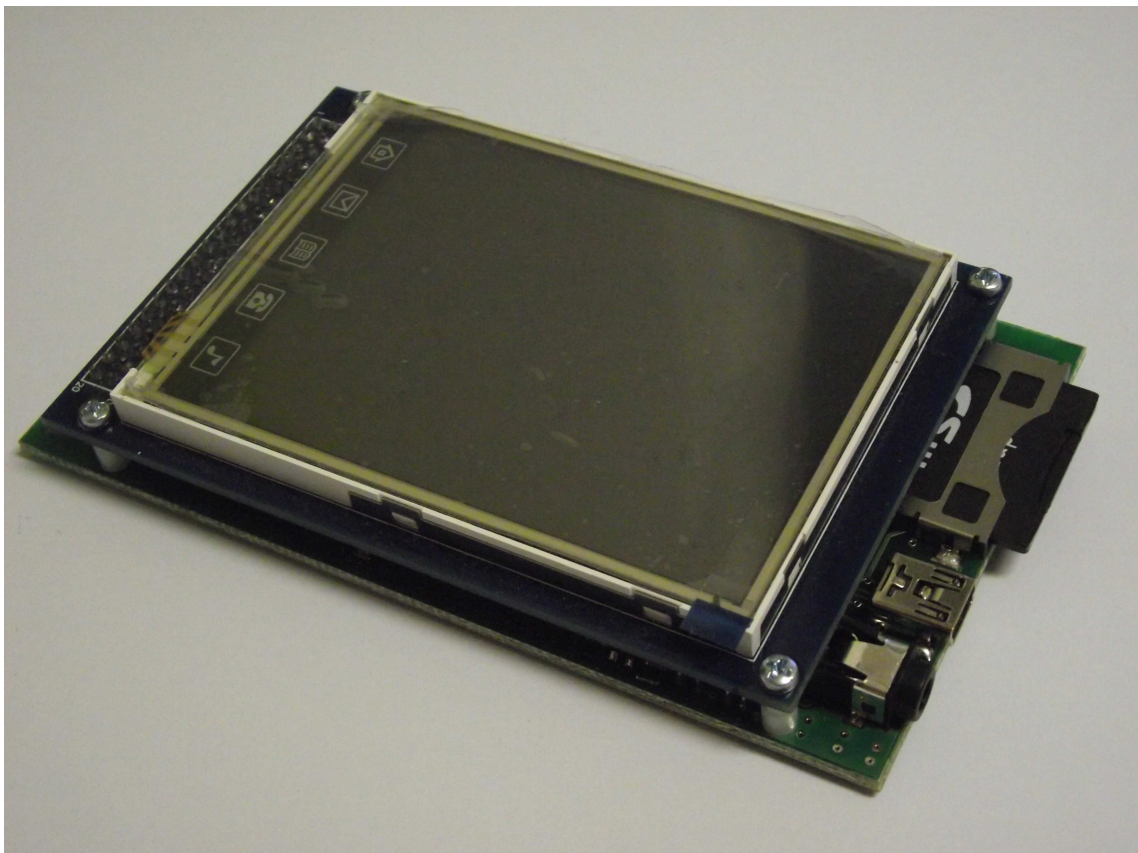
vytvorený testovací program ktorý do prevodníku FTDI odosiela testovacie bajty ktoré boli následne kontrolované na výstupe obvodu CPLD, teda na konektore CON2. Po overení funkčnosti bol k riadiacemu modulu monitora pripojený modul displeja. Po opätovnom pripojení k PC bola spustená vývojová verzia programu ktorý odosiela obrazové dáta na displej monitora. V tomto kroku boli všetky časti zariadenia úspešne oživené a monitor mohol byť považovaný za funkčný. Na obrázkoch 5, 6, 7 a 8 sú fotografie osadeného a oživeného monitoru.



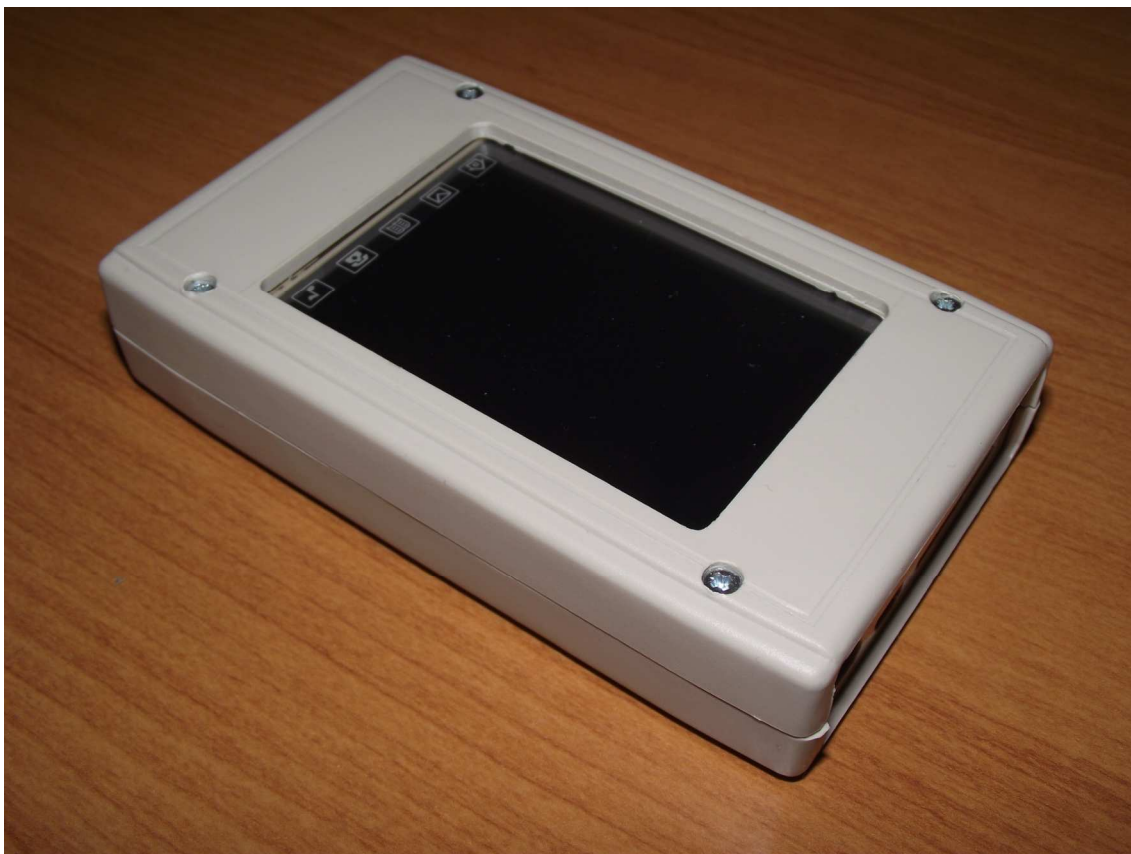
Obrázok 5. Fotografia osadeného DPS zo strany Top



Obrázok 6. Fotografia osadeného DPS zo strany Bottom



Obrázok 7. Fotografia osadeného DPS s displejom



Obrázok 8. Fotografia monitora v krabičke

5.2 Program riadiaceho mikrokontroléru HCS08

V zariadení je použitý mikrokontrolér HCS08JM16 ktorý komunikuje s riadiacim programom a na základe jeho pokynov ovláda požadované periférie. Komunikácia s riadiacim programom bude popísaná v nasledujúcej kapitole. Program pre mikrokontrolér bol vytvorený v štúdiu CodeWarrior 6.3. Na začiatku programu je spustená inicializačná sekvencia ktorá nastaví oscilátor a vnútorné periférie mikrokontroléru. K tomu slúži knižnica `MCUinit.c`. Táto je vygenerovaná pomocou štúdia CodeWarrior 6.3 na základe nastavenia v rámci utility Device Initialization. Nástroj umožňuje nastaviť periférie pomocou grafického rozhrania a tým výrazne zjednodušuje prvotné nastavenie periférií. Knižnica v prvom kroku nastaví tzv. `write once registers`. Ide o registre `SOPT1`, `SOPT2`, `SPMSC1` a `SPMSC2`. Tieto upravujú použitie Watchdogu a niektorých prerušení. Nasleduje nastavenie registrov `MCGC1`, `MCGC2` a `MCGC3` ktoré určujú nastavenie oscilátoru. V našom prípade je ako zdroj hodinového signálu použitý výstup z mikrokontroléru riadiaceho čítačku SD kariet. Tým bola odbúraná nutnosť použitia ďalšieho kryštálu prípadne oscilátoru. Nasleduje nastavenie pinov GPIO, inicializácia AD prevodníku, sériového kanálu UART, prerušenia od KBI a nastavenie kanálu TPM pre riadenie podsvietenia displeja. Knižnica `MCUinit.c` v závere obsahuje inicializáciu vektorov prerušení.

Súbor `main.c` obsahuje hlavný program a kódy obsluhy prerušení. V hlavnom programe je ako prvá zavolaná funkcia `MCU_init` z knižnice `MCUinit.c` ktorá je popísaná vyššie. Po inicializácii periférií sú nastavené piny portu PTE. Na týchto pinoch je emulovaná linka SPI pripojená k Touchscreen controlleru a audio prevodníku. V tomto projekte sa však možnosť komunikácie s audio prevodníkom nevyužíva. Piny ktorých označenie začína symbolom “_” určujú smer toku dát daného pinu. Konfigurácia je nasledovná.

```
_DOUT = 0;
_DIN = 1;
_DCLK = 1;
_CS = 1;
_DBUSY = 0;
```

Pin nastavený na hodnotu 0 je nastavený ako vstup a pin nastavený na hodnotu 1 je nastavený ako výstup. Smer komunikácie je určený z pohľadu touchscreen controlleru, preto je pin DOUT nastavený ako vstup a pin DIN nastavený ako výstup. Pin DBUSY je príznak zaneprázdnenosti ktorým controller povoľuje komunikáciu s mikrokontrolérom.

Bezprostredne po inicializácii je zavolaná funkcia AD7843. Táto funkcia vyčítava dáta z touchscreen controlleru. V úvode funkcie je čakacia slučka `while(DBUSY==1);`. Program čaká na povolenie komunikácie s controllerom. Po tom čo je obvod AD7843 pripravený, program aktivuje signál CS(v negatívnej logike) čím povolí komunikáciu s touchscreen controllerom. Potom odošle do obvodu znak 0x9C. Tým nastaví obvod k premeraniu dotyku na súradnici Y. Po uvoľnení príznaku DBUSY dôjde k prečítaniu hodnoty Y-ovej súradnice. Následne sa postup opakuje s odoslaním príkazu 0xDC pre vyčítanie hodnoty na X-ovej súradnici. Dáta sú uložené do premenných TP_X a TP_Y. Na konci funkcie AD7843 je signál CS nastavený na 1 a tým je komunikácia s controllerom vypnutá.

Po prvom vyčítaní dát z Touchscreen controlleru program pokračuje do nekonečného cyklu. V tomto cykle je kontrolovaný pin PTD2. K tomuto pinu je privedený signál z touchscreen controlleru „Pen Interrupt“. Ten je nastavený vždy keď dôjde k aktivite na dotykovej fólii displeja a k vynulovaniu dochádza po prečítaní dát. Ak je signál aktívny, dôjde k volaniu funkcie AD7843 ktorá prečíta dáta z controlleru a uloží ich do vyššie uvedených premenných. V prípade že premenné neobsahujú hodnotu 0, do premennej `active` je zapísaná hodnota 1 a obsah premennej TP_X je skopírovaný do premennej `touchx`. To isté sa udeje s premennou TP_Y ktorá je skopírovaná do premennej `touchy`. Tento postup je potrebný kvôli tomu, aby za platné dáta neboli považované nulové súradnice ale súradnice posledného miesta dotyku. Príznak `Pen Interrupt` sa totiž nastavuje aj v prípade uvoľnenia displeja a následne vyčítané dáta sú potom rovné 0. Premenná `active` určuje, či sa užívateľ v danom momente dotýka alebo nedotýka displeja. Obsah premennej je odosielaný do PC ako súčasť správy popísanej v kapitole 5.3. Po prekopírovaní dát ešte dochádza aktualizácií premennej `zoom`. Ide o premennú do ktorej je uložená hodnota Y-ovej súradnice dotyku v prípade, že X-ová hodnota je väčšia ako 220, ide teda o dotyk v krajnej časti displeja. Hodnota premennej `zoom` je súčasťou správy popísanej v kapitole 5.3. je možné ju v PC

programe použiť k zmene veľkosti obrazu.

Paralelne s hlavným programom prebieha obsluha prerušenia ADC. Prerušenie je vyvolané vždy po ukončení A/D prevodu na niektorom z kanálov. Ide o vyhodnocovanie dát z akcelerometra ktorý je k spomínanému A/D prevodníku pripojený. Výstup merania preťaženia na súradnici X je pripojený ku kanálu 5, výstup súradnice Y ku kanálu 8 a súradnica Z je pripojená ku kanálu 9. Obsluha prerušenia začína výberom meraného kanálu. Na to slúži podmienka `switch` ktorá vyberá z registru `ADCSC1_ADCH`. Podľa hodnoty, ktorá odpovedá číslu práve zmeraného kanálu, je podmienkou `case` realizované spracovanie príslušných dát. V prípade že je hodnota `ADCSC1_ADCH` rovná 5, zmeraný bol práve 5. kanál. To znamená že v registri `ADCR` je hodnota merania súradnice X. Táto hodnota je v rámci obsluhy podmienky `case` uložená do premennej `x`. Keďže pri analógovom prenose, rovnako ako pri vlastnom A/D prevode dochádza k pridaniu šumu k užitočnej zložke signálu, je ho nutné odfiltrovať. Ako postačujúca metóda pre potreby daného zariadenia sa ukázalo exponenciálne zabúdanie. Nameraná hodnota v premennej `x` je vynásobená 10krát a pripočítaná k 90 násobku premennej `xfilt`. Výsledok je uložený do premennej `xfilt` a vydelený 100. Tým má vždy aktuálne meranie váhu iba 0,1 v porovnaní s predchádzajúcimi meraniami. Premenná `xfilt` je dátového typu `long`, preto ju ešte pred ďalším spracovaním skopírujeme do premennej typu `CHAR` s názvom `xres`. Na konci obsluhy podmienky `case` je do registru `ADCSC1_ADCH` uložená hodnota 8. To inicializuje ďalší A/D prevod, tento krát na kanáli 8. Spracovanie v rámci obsluhy prerušenia prebieha obdobne ako pri spracovaní dát z X-ovej súradnice. Po skončení obsluhy merania Y-ovej súradnice je opakovaný obdobný postup pre súradnicu Z. Výsledne dáta sú teda uložené v premenných `xres`, `yres` a `zres`.

Ďalším prerušením ktoré je v programe využívané je prerušenie od príjmu na kanáli `SCI`. Po prijatí dát z druhého kanálu prevodníka `FTDI` je vyvolané prerušenie. Na začiatku je prečítaná hodnota z registru `SCI1S1`. Dôjde tým k uvoľneniu príznaku prerušenia. Následne sú do premennej `data[]` skopírované platné prijaté dáta. Premenná `data[10]` je pole typu `char` o dĺžke 10 bytov. Adresa pola je vždy daná aktuálnym ukazateľom na pole ktorý je po každom prijíme dát inkrementovaný. V prípade že je aktuálne prijatá hodnota `0x0D` a ukazateľ na pole má hodnotu 8 znamená

to, že bola prijatá kompletná správa z riadiaceho program v PC. V tomto prípade je ukazateľ na pole vynulovaný a prijatá správa prichádza k ďalšiemu spracovaniu. Ak je splnená podmienka že hodnota premennej `data[0]` je rovná identifikátoru "B", správa je považovaná za platnú. V tomto momente je v premennej `data` uložená správa ktorej formát je popísaný v kapitole 5.3. Z prvej pozície je hodnota intenzity podsvietenia skopírovaná priamo do registru `TPM1C1V` ktorý riadi duty cycle PWM modulácie. Z pozície 2 je skopírovaná hodnota do premennej `sound_en` dátového typu `char`. Z tretej pozície je podobne skopírovaná hodnota do premennej `sd_en` rovnakého dátového typu. Keďže spracovanie prijatých dát obnáša iba niekoľko príkazov, je realizované v rámci obsluhy prerušenia ihneď po prijatí dát. Pripojenie slúchadiel je detekované na pine `PTD1`. Tento je makrom nazvaný ako pin `PHONES`. Pri spracovaní prijatej správy je hodnota premennej `sound_en` logicky sčítaná s negovanou hodnotou pinu `PHONES`. Výsledok je následne skopírovaný na pin `PTF4` označovaný ako `SNDON`. V ďalšom kroku je obsah premennej `sound_en` skopírovaný priamo na pin `PTB4` označovaný ako `AMPON`. Tým dosiahneme požadovaný efekt zapnutia USB DAC ihneď po pripojení slúchadiel, nedôjde však k pripojeniu NF zosilňovača. V prípade že dorazí pokyn zapnúť zvukový výstup z riadiaceho programu v PC, dôjde k zapnutiu NF zosilňovača aj USB DAC bez ohľadu na to či sú slúchadlá pripojené alebo nie. V prípade že sú pripojené, obsah správy nemá na aktivitu DAC vplyv. V ďalšom kroku je hodnota premennej `sd_en` skopírovaná priamo na pin `PTG4` ktorý je označovaný ako `SDON` a ovláda aktivitu čítačky SD kariet.

Pri oživovaní zariadenia došlo k zisteniu, že USB DAC nemôže byť odpájaný iba pomocou riadenia napájacieho napätia. Audio prevodník zdieľa dátovú linku spolu s Touchscreen controllerom a ten je aktívny počas celej doby prevádzky. Napätie na dátovej linke sa cez ochranné diódy vo vnútri obvodu dostávalo na odpojenú napájaciu vetvu a tým pádom cez resistor `R49` na D+ linky USB. To malo za následok falošné detekcie prítomnosti neidentifikovaného zariadenia na linke USB. Túto situáciu bolo nutné vyriešiť úpravou zapojenia. Jedna strana rezistoru `R49` bola odpojená z hladiny riadeného napájania a bola pripojená k pinu procesoru. Ten riadi napájanie DAC a zároveň po pripojení napájania aktivuje pin `PTE7` označený ako `DACPULLUP` ktorým k jednej strane rezistoru `R49` pripojí napätie. K aktivácii dochádza v obsluhu prerušenia od príjmu dát a na pin `DACPULLUP` je skopírovaná aktuálna hodnota pinu `SNDON`.

Po spracovaní prijatých dát je odoslaná správa z mikrokontroléru do riadiaceho program v PC. Jednotlivé dáta sú odosielané po sériovej linke tak, ako je to popísané v kapitole 5.3. Ako prvá je odoslaná hodnota premennej `xres`, nasleduje premenná `yres`, a po nej `yres`. Ďalej je odoslaný symbol nesúci informáciu o 5 bitoch. Formát odoslania je nasledovný.

```
Send((unsigned
char)((active*1)+(PHONES*2)+(SD_ON*4)+(SNDON*8)+(AMPON*16))
);
```

Na pozícii LSB sa nachádza hodnota premennej `active`, nasleduje negovaná hodnota pinu `PHONES`, potom pin `SD_ON`, `SNDON` a nakoniec pin `AMPON`.

Nasleduje odoslanie hodnôt premenných `xtouch`, `ytouch` a `zoom`. Posledný odoslaný znak má hodnotu `0x0D` a slúži k identifikácii ukončenia správy.

Mikrokontrolér vykonávajúci tento program pracuje pri frekvencii 24MHz. Beh programu nie je sledovaný pomocou Watchdogu. Pri testovaní sa však nevyskytli žiadne problémy, kolízie alebo poruchy.

5.3 Komunikácia mikrokontroléru s riadiacim programom

Vzájomná komunikácia PC a mikrokontroléru HCS08 prebieha pomocou prevodníku FTDI. Tento prevodník má k dispozícii 2 na sebe nezávislé dátové kanály. K prenosu videa sú využité oba kanály. Druhý kanál pracuje v móde rozhrania RS232 a to umožňuje využiť signály RXD, TXD, RTS a DTR. Nastavenie portov prevodníku je bližšie popísané v kapitole 5.4. U druhého kanálu ide iba o využitie signálov riadenia toku RTS (ready to send) a DTR (data transmit ready). Tieto však nie sú využívané vo svojom obvyklom význame a slúžia ako pomocné signály. Ich fungovanie je popísané v kapitole 5.5. Signály RXD a TXD ostávajú nevyužité k prenosu alebo riadeniu obrazových dát a preto je ich možné využiť ku komunikácií riadiaceho programu s mikrokontrolérom.

Prenos dát prebieha v asynchrónnom rámci o dĺžke 8 bitov s použitím 1 stop bitu, bez použitia kontroly prenosu paritou. Rýchlosť komunikácie je nastavená na 1120000Bd.

Komunikácia oboma smermi prebieha v správach dlhých 8 bytov z ktorých je každá ukončená znakom 0x0D ktorý v ASCII reprezentácii odpovedá znaku CR (carrier return). Správa z riadiaceho programu v PC je znázornená na obrázku 10.

ID	BL	SND	SD	X	X	X	<CR>
'B'	0 - 255	0/1	0/1				13

Obrázok 10 – formát správy medzi PC a MCU

Znak ID na pozícii 0 nesie vždy hodnotu „B“. Slúži k identifikácii platnej správy a k potvrdeniu správnej pozície ukazateľa na prijímací buffer.

Nasledujúci znak BL na pozícii 1 nesie hodnotu od 0 do 255. Táto určuje intenzitu podsvietenia pričom hodnota 255 znamená najväčšiu intenzitu podsvietenia a hodnota 0 vypnuté podsvietenie.

Znak SND nesie hodnotu 0 alebo 1. Ak by obsahoval inú hodnotu, táto bude ignorovaná. Hodnota 1 zapína USB DAC, hodnota 0 tento prevodník vypne pokiaľ nie sú v rovnakom momente pripojené slúchadlá. V tom prípade DAC ostáva zapnutý.

Znak SD môže nadobúdať hodnôt, rovnako ako SND, 0 alebo 1. V prípade že SD je rovný, 1 dochádza k zapnutiu čítačky SD kariet. Po vyhodnotení znaku 0 je čítačka SD kariet vypnutá.

Nasledujúce 3 byty nenesú nijakú informáciu a boli ponechané ako rezervné. Posledný byte je ukončovacia znak CR.

Prijatie správy od riadiaceho programu v PC zároveň inicializuje odoslanie správy opačným smerom. Formát správy je uvedený na obrázku 11.

X	Y	Z	STAT	TOX	TOY	ZOOM	<CR>
0 - 255	0 - 255	0 - 255	PH 0/1 SD 0/1 SND 0/1 AMP 0/1	0 - 255	0 - 255	0 - 255	13

Obrázok 11 – formát správy medzi MCU a PC

Byty X, Y a Z nesú informáciu o aktuálnom náklone monitoru vo všetkých 3 osiach.

Nultý bit A znaku STAT nesie informáciu o tom či sa v danom momente užívateľ dotýka touchscreen senzoru alebo nie. V prípade že áno, bit je nastavený na 1, v prípade že nie, byt je vynulovaný.

Bit PH znaku STAT obsahuje informáciu o pripojení slúchadiel. V prípade že sú slúchadlá pripojené, bit je nastavený. V opačnom prípade je vynulovaný.

Bit SD obsahuje informáciu o aktivácii čítačky SD kariet. Ak je čítačka aktívna, bit je nastavený. V opačnom prípade je bit vynulovaný. Táto informácia je totožná s informáciou zo správy odoslanej z PC do mikrokontroléru. Môže byť preto použitá pre kontrolu správneho prijatia dát.

Bit SND obsahuje správu o aktivácii DAC a rovnako ako bit SD je kópiou prijatej správy. V prípade že by prijatá hodnota SND alebo SD nesúhlasila s odoslanou, znamená to chybu v prenose dát. Existuje teoretická možnosť skratu na pine PTF4 (pre prípad

nesúhlasu v hodnotách SND) alebo PTG4 (pre prípad nesúhlasu v hodnotách SD). V prípade už oživeného HW umiestneného v pevnej krabičke však môžeme túto možnosť vylúčiť.

Bit AMP oznamuje, či je zapnutý alebo vypnutý audio zosilňovač. V prípade že je bit AMP nastavený, audio zosilňovač je aktívny. V opačnom prípade je zosilňovač vypnutý.

Byte TOX nesie informáciu o hodnote vyčítanej z touchscreen controlleru. Ide o X-ovú súradnicu dotyku monitoru. Môže nadobúdať hodnôt od 0 do 255.

Byte TOY nesie rovnako ako TOX informáciu o hodnote vyčítanej z touchscreen controlleru. V tomto prípade ide o Y-ovú súradnicu dotyku plochy monitoru. Rovnako nadobúda hodnôt od 0 do 255.

Byte ZOOM nesie informáciu o poslednom dotyku v Y-ovej súradnici za hranicu $X > 220$. Ide o oblasť pravého okraja displeja kde poloha ukazateľa môže slúžiť k určeniu veľkosti výrezu obrazovky. Rovnako ako TOY nadobúda hodnôt 0-255.

Posledný znak správy je ukončovacia znak CR.

Komunikáciu vyvoláva vždy výlučne riadiaci program v PC. K tomu slúži funkcia v riadiacom programe nazvaná `McuSet(UInt16 pwm)`. Parameter `pwm` určuje triedu PWM cyklu podsvietenia LCD. Funkcia je volaná po prijatí kompletnej a korektnej správy alebo ako reakcia na podnet užívateľa. Môže ísť napríklad o žiadosť zmeny intenzity podsvietenia. Mikrokontrolér v prerušení od príjmu UART vyhodnotí dáta a okamžite odosiela odpoveď. Odpoveď je spracovaná v riadiacom programe v PC v timeri nazvanom `readtimer` ktorý je vyvolávaný s periódou 100ms. Na začiatku obsluhy timeru je pomocou nasledujúceho príkazu overené či boli prijaté dáta.

```
ftStatus = FtdiDeviceB.GetRxBytesAvailable(ref fronta);
```

Ak je hodnota premennej `fronta` rovná hodnote 8, dáta boli prijaté. Po prijatí je skontrolovaná správnosť dát kontrolou prítomnosti ukončovacieho znaku `0x0D` na pozícii 7. V prípade že áno, dáta z príjmového bufferu sú skopírované do premenných k ďalšiemu spracovaniu. Ak správa nebola prijatá korektne, obsah príjmového bufferu je ignorovaný a neprečítané dáta v prevodníku zmazané. Nasleduje nové vyvolanie komunikácie, volaním funkcie `McuSet`. Pri spracovaní prijatých dát dochádza k

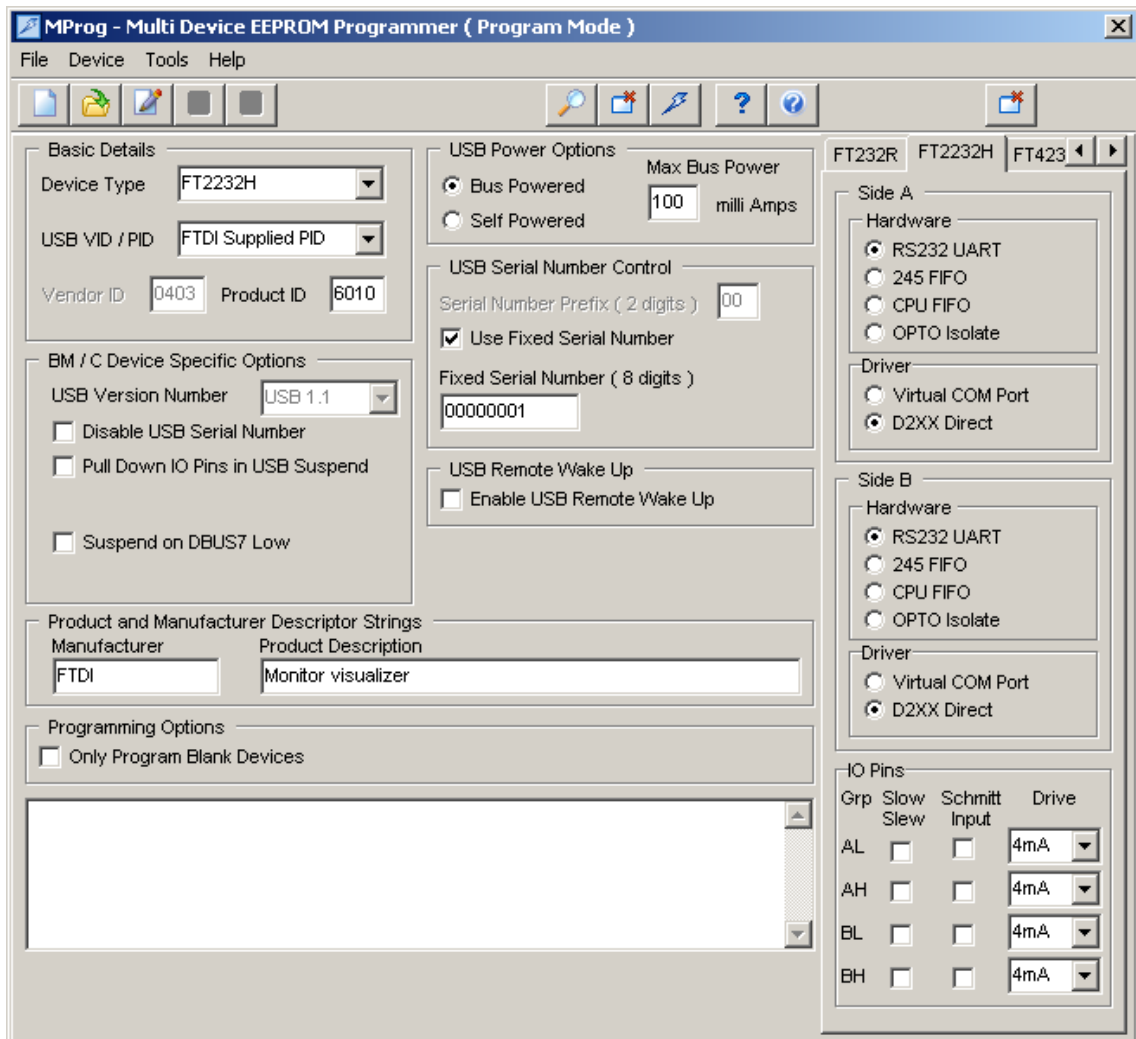
vyhodnoteniu prijatých dát o aktuálnom zrýchlení.

```
if ((xa > 80) & (xa < 160) & (ya < 100)) rotation = 0;
if ((ya > 80) & (ya < 170) & (xa < 80) & (player==0))
rotation = 3;
if ((xa > 80) & (xa < 160) & (ya > 170)) rotation = 2;
if ((za > 60) & (za < 160) & (xa > 160) & (player==0))
rotation = 1;
```

Využitie premennej `rotation` je popísané v kapitole 5.5. Súčasť podmienky predpokladajúca hodnotu 0 v premennej `player` rozlišuje či je riadiaci program v režime video prehrávača alebo nie. V tom prípade totiž nedochádza k rotácii obrazu na pomer strán 3 x 4 ale vždy len 4 x 3. Režim video prehrávača je podrobne popísaný v kapitole 5.7. V prípade, že nie je zamknutý touchscreen, teda hodnota premennej `touchlock` je rovná 0, dochádza k spracovaniu dát z touchscreen controlleru. V prípade že bol vykonaný dotyk krajnej časti displeja, teda hodnota premennej `touchx` je väčšia ako 220, je do premennej `cutsiz` nakopírovaná hodnota premennej `touchy`. Jej význam je podrobnejšie popísaný v kapitole 5.6. V prípade že je vykonaný dotyk medzi súradnicami 8 a 18 na X-ovej ose a medzi súradnicami 28 a 50 na Y-ovej ose, dôjde k aktivácii alebo deaktivácii ovládania touchscreenom. Ďalej je podľa premennej `rotation` upravené spracovanie prijatých dát z touchscreen. Pre správny posun obrazu presúvaním kurzoru po povrchu displeja je nutné vykonať prepočet spracovaných dát vzhľadom na aktuálne natočenie monitoru. V poslednom kroku je z posuvu medzi dvomi periódami počas držania kurzoru na displeji monitora spočítaný posuv výrezu obrazu a pripočítaný k aktuálnej hodnote.

5.4 Nastavenie a ovládanie prevodníku FTDI

Obvod FT2232H je prevodník USB rozhrania od firmy Future Technology Devices Ltd. Obsahuje 2 na sebe nezávislé prevodníky USB rozhrania na niekoľko rôznych typov rozhraní pohodlne použiteľných v mikroprocesorovej technike. Okrem zrejmych rozhraní ako sú napríklad RS232, FIFO, RS 245, umožňuje komunikáciu na rozhraniach IIC, SPI alebo JTAG. Tieto však v našom prípade nebudú využité. Informácie o aktuálnej konfigurácii prevodníku sú uložené v externej pamäti EEPROM typu 93LC46. Z nej si obvod vždy po reštarte požadovanú konfiguráciu načíta a použije k internému nastaveniu. K nakonfigurovaniu pamäte EEPROM je možné využiť program M-Prog dodávaný firmou FTDI. Screenshot z programu M-Prog je uvedený na obrázku 12.



Obrázok 12 – screenshot programu M-Prog

Konfiguráciu je možné vykonať aj z užívateľského programu. V sériovej výrobe je výhodné použiť vopred naprogramovanú EEPROM na danú konfiguráciu obvodu. Pre účely vývoja je najvýhodnejšie použiť program M-Prog. Aktuálne nastavenie obvodu je vždy možné jednoducho vyčítať z EEPROM a pri prípadnej úprave nie je nutné konfigurovať celý obvod nanovo. Taktiež je možné nastavenie uložiť do súboru s príponou .ept a použiť ho vždy pri novej konfigurácii. Program M-Prog využíva ku konfigurácii grafické rozhranie, takže nastavovanie je veľmi pohodlné.

Firma FTDI dodáva k obvodom sadu knižníc ktoré je možné využiť pri vývoji aplikácií v rôznych vývojových prostrediach. Riadiaci program Miniaturneho PC monitoru je vytvorený v štúdiu Microsoft Visual C#.

Po spustení aplikácie je vždy nutné nadviazať komunikáciu s prevodníkom. Jedna z možností je pripojiť sa na virtuálny sériový port COM. Táto možnosť je v mnohých aplikáciách hojne využívaná. Je však užívateľsky nepohodlná pretože užívateľ musí pred každým spustením programu vybrať port na ktorý je zariadenie pripojené. To okrem zdržania vyžaduje od užívateľa určitú technickú zdatnosť pretože musí mať prehľad o aktuálnom namapovaní portov v danom PC. Ďalšou možnosťou ako sa k prevodníku pripojiť je využiť pripojenie podľa poradového čísla. Táto možnosť je užívateľsky značne pohodlnejšia. V tom prípade ale musí byť zabezpečené, že k PC nie sú pripojené iné zariadenia od firmy FTDI. To sa vždy zaručiť nedá a mohlo by dôjsť ku kolíziám. Najpraktickejšia možnosť je využiť pripojenie podľa sériového čísla prevodníka. Každý prevodník od firmy FTDI obsahuje sériové číslo. Pokiaľ toto číslo nie je užívateľom zmenené, je vygenerované náhodne. Ak má teda prevodník PC monitoru sériové číslo stanovené „00000001A“ je možnosť kolízie s náhodným sériovým číslom iného zariadenia zanedbateľná. Jedinou nevýhodou tohto pripojovania je fakt, že nie je možné k jednému PC pripojiť viac monitorov a spustiť viac užívateľských programov súčasne. Zápis kódu pre pripojenie k prevodníku je nasledovný.

```

ftStatus = FtdiDeviceA.GetNumberOfDevices(ref ftdiDeviceCount);
ftStatus = FtdiDeviceA.GetDeviceList(ftdiDeviceList);
if (ftdiDeviceCount > 1)
{
    if ((ftdiDeviceList[0].SerialNumber == "00000001A") &
        ftdiDeviceList[1].SerialNumber == "00000001B"))
    {
        ftStatus =
FtdiDeviceA.OpenBySerialNumber(ftdiDeviceList[0].SerialNumber);

        ftStatus =
FtdiDeviceB.OpenBySerialNumber(ftdiDeviceList[1].SerialNumber);

```

V prvom kroku je vhodné overiť počet pripojených zariadení. Tým zabránime kolízii v prípade, že k PC nie je pripojené nijaké zariadenie. Počet zariadení sa uloží do premennej `ftdiDeviceCount` a informácia o priebehu operácie sa uloží do premennej `ftStatus`. Tú je vhodné sledovať pri vývoji, v aplikácii jej hodnotu môžeme ignorovať. V ďalšom kroku program vyčíta zoznam FTDI zariadení ktoré sú k PC pripojené a uloží ich do štruktúry `ftdiDeviceList`. Po vyhodnotení počtu pripojených zariadení program skontroluje či sa jedná o prevodníky Miniaturného PC monitoru. Kontrola prebieha vyhodnotením premennej `ftdiDeviceList[].SerialNumber`. V prípade že sériové čísla oboch zariadení súhlasia, dôjde k otvoreniu komunikácie na oboch portoch. V ďalšom kóde nebudú zariadenia rozlišované podľa sériového čísla ale budú rozlíšené volaním funkcie `FtdiDeviceA` alebo `FtdiDeviceB`. Port A je pritom využitý k odosielaniu obrazových dát do displeja, port B slúži ku komunikácii s mikrokontrolérom HCS08 po linke UART.

V ďalšom kroku dochádza k nastaveniu komunikačných módov. Port A je nastavený do komunikačného módu Async Bit Bang príkazom.

```

ftStatus = FtdiDeviceA.SetBitMode(0xFF,
FTDI.FT_BIT_MODES.FT_BIT_MODE_ASYNC_BITBANG);

```

Ten tento mód nastaví spodný byte portu A ako paralelný vstupno-výstupný port. Horný byte obsahuje signály riadenia toku dát.

Ďalej je nutné nastaviť rýchlosť komunikácie. Aj keď sa v tomto momente už nejedná o asynchrónnu sériovú komunikáciu, rýchlosť prenosu dát sa odvíja od nastavenia baudrate generátoru. Preto príkaz pre nastavenie rýchlosti komunikácie vypadá nasledovne.

```
ftStatus = FtdiDeviceA.SetBaudRate(1200000);
```

Rýchlosť prenosu dát je nastavená na maximum podporované prevodníkom. Port B sa nastavuje obdobne, nenastavujeme však nijaký špeciálny mód. Komunikácia ostáva v režime sériového asynchrónneho prenosu. Baudrate je nastavený na hodnotu rovnakú ako v mikrokontroléri, tj. 1120000Bd.

5.5 Riadiaci program – príprava a odosielanie dát

Základnou funkciou riadiaceho programu je funkcia `SendInt(UInt16)`. Jej vstupom je 16 bitový integer ktorý funkcia odošle do prevodníka FTDI. Dáta sú odoslané pomocou funkcie `FtdiDeviceA.Write`. Ako prvý je odoslaný spodný byte, po ňom nasleduje horný. Táto funkcia je volaná z funkcií rozdelených podľa toho či sa jedná o odosielanie dát alebo príkazov. Funkcia `main_Write_COM` odosiela do displeja príkaz, `main_Write_DATA` odosiela dáta. Vo vnútri funkcie je výber prevedený stavom na pine RS. Vo funkcii odosielajúcej dáta je pin RS nastavený, pri odosielaní príkazov je RS vynulovaný. Pin RS rovnako ako RST je riadený vo funkciách `RSSet()` a `RSClr()`, resp. `RSTSet()` a `RSTClr()`. Ako riadiace piny sú použité piny druhého kanálu FTDI prevodníka. Ide o signály DTR (RS) a RTS (RST). Príkaz pre nastavenie pinu RS má nasledujúci tvar.

```
FtStatus = FtdiDeviceB.SetDTR(Convert.ToBoolean(0));
```

Piny RS a RST sú nastavované v invertovanej logike. Funkcie `main_Write_COM` a `main_Write_DATA` sú využité vo funkcií `main_W_com_data(UInt16 com, UInt16 dat)`.

S jej pomocou je možné na zadanú adresu v pamäti LCD zapísať požadované dáta. Táto funkcia je využitá napríklad v rámci funkcie `address_set(UInt16 x1, UInt16 y1, UInt16 x2, UInt16 y2)`. Táto funkcia nastaví súradnice ľavého horného a pravého dolného rohu obrazu. Následne odosielané dáta budú zobrazované práve medzi týmito dvomi bodmi. Pri každom štarte programu je do displeja odoslaná pevná sekvencia dát ktorá displej inicializuje. Táto je vykonaná z funkcie `LCD_Init()`. Na začiatku funkcie nastavený, nulovaný a opäť nastavený signál RST. Tým je zabezpečené že dáta budú prijaté správne displejom a zároveň budú správne rozdelené v obvode CPLD. Po reštarte sú do registrov displeja uložené inicializačné dáta nesúce informácie o nastavení vstupných farieb, kontrastu a iných interných vlastnostiach [7]. Takto nastavený displej je pripravený na príjem obrazových dát. Odosielanie dát prebieha v rámci obsluhy časovača s názvom `VideoTimer`. Ten je po aktivácii vykonávaný s periódou 30ms. Na začiatku timeru je vyhodnotená aktuálna poloha kurzora myši. Podľa nej je nastavovaná priehľadnosť okna riadiaceho programu.

Priehľadnosť nemôže byť nastavená na 100% pretože by bolo okno riadiaceho programu snímané a jeho obraz odosielaný do monitora. Priehľadnosť okna je nastavená na 80% v prípade že sa kurzor myši nachádza v priestore okna, na 50% v momente kedy je kurzor myši na inej pozícii. V ďalšom kroku je vyhodnotený údaj o aktuálnom natočení monitora na základe obsahu premennej `rotation` dátového typu `Int16`. V prípade že premenná nesie hodnotu 0 alebo 2, bude z obrazovky PC zosnímaný obraz v pomere strán 4 x 3. V prípade hodnôt 1 alebo 3 bude zosnímaný obraz v pomere 3 x 4.

```
case 0:
case 2:
    trackx = Convert.ToUInt16(cutsizesize * 4);
    tracky = Convert.ToUInt16(cutsizesize * 3);
    xx = posx - trackx / 2;
    yy = posy - tracky / 2;
    if (xx < 0) xx = 0;
    if (yy < 0) yy = 0;
    if (xx > Screen.PrimaryScreen.Bounds.Width - trackx)
        xx = Screen.PrimaryScreen.Bounds.Width - trackx;
    if (yy > Screen.PrimaryScreen.Bounds.Height - tracky)
        yy = Screen.PrimaryScreen.Bounds.Height - tracky;
    Xpos = Convert.ToUInt16(xx);
    Ypos = Convert.ToUInt16(yy);
break;
```

Do premennej `trackx`, resp. `tracky` typu `UInt16` je uložená veľkosť snímaného obrazu v danej ose. Tá závisí na obsahu premennej `cutsizesize` ktorá po spustení programu nesie hodnotu 80. Táto hodnota sa môže počas behu programu meniť. Premenné `posx`, resp. `posy` nesú údaj o aktuálnej polohe kurzora ktorý ukazuje na stred snímaného obrazu. Nasledujúcim výpočtom teda dôjde k stanoveniu snímaného priestoru v okolí súradníc `posx` a `posy`. Poloha je následne skontrolovaná. Ak je

jedna zo súradníc menšia ako 0, prípadne prekračuje veľkosť aktuálnej pracovnej plochy, jej hodnota je obmedzená na daný limit. Obdobne je postupované pre prípad, že premenná `rotation` nesie hodnoty 1 alebo 3. V ďalšom kroku obsluhy `VideoTimer`-u program vyhodnocuje či má snímaný obraz vlastnosti video prehrávača alebo nie. Podrobný popis módu video prehrávača je uvedený v kapitole 5.7. V prípade že program nefunguje ako video prehrávač, je opäť vyhodnocovaná hodnota v premennej `rotation`. Podľa hodnoty je inicializovaná lokálna premenná `memoryImage` s rozlíšením obrázku v pomere 4 x 3 alebo 3 x 4, v definovanej farebnej hĺbke. Farebná hĺbka je nastavená na 16 bitov.

```
gfxScreenshot = Graphics.FromImage(memoryImage);  
gfxScreenshot.CopyFromScreen(Xpos, Ypos, 0, 0,  
memoryImage.Size, CopyPixelOperation.SourceCopy);
```

Uvedený kód kopíruje obrazové dáta z obrazovky monitora do premennej `memoryImage` zo súradníc `Xpos` a `Ypos`. Následne je opäť v závislosti na obsahu premennej `rotation` rozhodované o zmene veľkosti obrazu. Ak je obsahom `rotation` hodnota 0 alebo 2 a premenná `cutsizes` nenesie hodnotu 80 (obraz nemá rozlíšenie 320 x 240), dôjde k zmene veľkosti obrazu. K tej slúži funkcia `ResizeBitmap(Bitmap b, int nWidth, int nHeight)`. Ta zmení veľkosť obrazu na rozlíšenie 320 x 240, alebo 240 x 320. Takto pripravený obraz je zobrazený v okne riadiaceho programu ale len v prípade, že je okno programu maximalizované. Do odosielaného obrazu môžu byť následne pridané informačné ikony ktoré budú popísané v kapitole 5.8. Následne je z obrazu vytvorený dátový typ `MemoryStream` a z neho vytvorené pole `byteArray`. Toto pole nesie obsah súboru `.bmp` s obrazovými dátami. Podľa obsahu premennej `rotation` je nastavený register s adresou `0x0016` v LCD. Tento register určuje vertikálny a horizontálny smer vykresľovania dát na displeji. Týmto nastavením je docielené natočenie obrazu podľa aktuálnej polohy displeja. Pole `byteArray` je následne zbavené BMP hlavičky a odoslané do prevodníka. V prípade že sa jedná o prvý snímok po zapnutí displeja, nasleduje príkaz zapnutia LCD a do riadiaceho mikrokontroléru je odoslaný príkaz k zapnutiu podsvietenia. Tento postup zabráni nepríjemnému prebliknutiu monitora medzi zapnutím a zobrazením prvých platných dát.

5.6 Uživatelské rozhranie riadiaceho programu

Riadiaci program je v prostredí Microsoft Visual C# vytvorený ako WinForm aplikácia. Po spustení programu je ako prvý aktivovaný `Startup` timer. Ten má nastavenú periódu 1ms. V rámci obsluhy timeru je zosnímaný aktuálny obraz pracovnej plochy počítača presne v prostriedku obrazovky. Tento obraz sa uloží ako obrázok pozadia okna `f2`. Okno `f2` sa následne zobrazí. Na súradniciach odkiaľ bol zosnímaný obrázok plochy. To znamená že užívateľ PC nepozoruje nijakú zmenu obrazu na monitore. Následne sú zavolané funkcie `FTDI_Init()` a `LCD_Init()` ktorých činnosť je popísaná v kapitole 5.5. Následne je timer `Startup` deaktivovaný. Program pokračuje vykonávaním obsluhy timeru `F2opacity`. Ten s periódou 10ms zväčšuje nepriehľadnosť okna `f2`. Okno `f2` má ako obrázok pozadia nahraný predchádzajúci stav pracovnej plochy ale zároveň je súčasťou okna `f2` obrázok s fotografiou zariadenia. Fotografia má priehľadné okolie namiesto ktorého je zobrazovaný predchádzajúci stav pracovnej plochy PC. Výsledkom je postupné a pomalé vynáranie sa fotografie zariadenia zo stredu pracovnej plochy. Spomínaný úvodný efekt trvá 1 sekundu. Keď nepriehľadnosť dosiahne 100%, je aktivovaný `VideoTimer`. Súčasťou aktivácie timeru je aj aktualizácia hodnoty premennej `firstst_on` typu `int`. Po prvej aktivácii `VideoTimer-u` je `first_on` rovný 1. V obsluhu video timeru sa táto premenná vyhodnotí. Ak je nastavená na 1, bude po prvom nahraní obrazových dát do LCD aktivované zobrazenie LCD, aktivované podsvietenie. Následne bude premenná `first_on` vynulovaná. Týmto postupom sa zabráni užívateľsky nepríjemnému prebliknutiu neplatných dát pred zobrazením platného obrazu. Po uplynutí 3500ms od spustenia programu je aktivovaný timer nazvaný `StopIntro`. V rámci obsluhy tohto timeru je uzavreté okno `f2`, deaktivovaný timer `F2opacity` a `StopIntro`. V tomto stave je spustený `VideoTimer` a v pravom dolnom rohu obrazovky sa nachádza `Notify Icon` s obrázkom monitora a popisom „USB Monitor“. Dvojklikom na ikonu sa zobrazí okno riadiaceho programu. V prípade že v čase spustenia programu nebol monitor pripojený, program sa nachádza v offline režime. Program v tomto režime je vybavený popisom „USB Monitor (offline)“ a neodosiela v rámci obsluhy `VideoTimer-u` nijaké dáta. Okno programu je v tomto móde vybavené tlačidlom „Try to connect“. Kliknutím

na tlačidlo dôjde k opakovanému pokusu nadviazať spojenie s monitorom, teda FTDI prevodníkom zavolaním funkcie `FTDI_Init()`. V prípade úspešného spojenia s prevodníkom, program opúšťa režim offline a odosiela dáta do LCD. Dvojklikom ľavého tlačidla myši na Notify Icon je možné minimalizovať alebo maximalizovať okno programu. Dvojklikom pravého tlačidla sa zobrazí dialógová roleta v ktorej sú na výber jednotlivé položky z menu. Na výber je zobrazenie okna programu, spustenie video prehrávača, zobrazenie okna „About“ a ukončenie programu. Screenshot programu je na obrázku 13.



Obrázok 13 – screenshot užívateľského programu

Okno programu má v pravom hornom rohu ikony pre minimalizáciu a ukončenie programu. Kliknutím na ikonu „X“ je ako prvé vypnuté podsvietenie displeja, následne je resetovaný port A prevodníka FTDI, uzavretá komunikácia a nakoniec ukončený celý program. Kliknutím na ikonu „_“ dôjde k minimalizácii okna. K tomu je využitá funkcia `this.Hide()` a do premennej `minimize` typu `UInt16` je nahraná hodnota 1. Obsah premennej má význam pri vyhodnocovaní dvojkliku pravým tlačidlom myši na Notify Icon. Program musí poznať či je práve minimalizovaný alebo maximalizovaný. Až potom môže aktuálny stav zmeniť. V strede okna programu je zobrazovaný náhľad na aktuálne obrazové dáta. Obraz je obnovovaný v rámci obsluhy `VideoTimer-u` a

otočenie prislúcha aktuálnemu stavu v premennej `rotation`. Obraz je tak zobrazovaný v pomere 4 x 3 (320 x 240) alebo 3 x 4 (240 x 320). Pozdĺž spodného a pravého okraja okna sú 2 obrázky. Na oboch je zobrazený postupný prechod farieb z bielej do čiernej. Vždy je však viditeľný len jeden obrázok, v závislosti na aktuálnom pomere strán obrazu. V prípade pomeru 4 x 3 je viditeľný obrázok na spodnej hrane okna, v prípade pomeru 3 x 4 je aktívny obrázok na pravej strane okna. Výber aktivity obrázku prebieha v obsluhu `VideoTimer-u` na základe hodnoty v premennej `rotation`. Kliknutím na aktívny obrázok je možné zmeniť aktuálnu hodnotu podsvietenia LCD. V okamihu kliknutia program zosníma aktuálnu polohu kurzora myši. V prípade že bolo kliknuté na vertikálny obrázok, je pre výpočet žiadanej hodnoty podsvietenia použitá Z-ová súradnica polohy kurzora myši. V prípade že bolo kliknuté na horizontálny obrázok, pre nastavenie podsvietenia bola použitá X-ová súradnica kurzora myši. Prepočítaná hodnota na rozsah 0 – 255 je uložená do premennej `backlight` typu `UInt16`. V najbližšom cykle timeru obsluhujúceho sériovú komunikáciu je odoslaná do riadiaceho mikrokontroléru.

V každom rohu okna programu je umiestnená jedna ikona. Tieto ikony sú v offline móde skryté. V on-line móde však umožňujú ovládať pomocné zariadenia monitoru. V ľavom hornom rohu sa nachádza ikona známa z OS Windows znázorňujúca vymeniteľný disk. Kliknutím na ikonu sa nastaví premenná `SD_en` na hodnotu 1. Tá je v najbližšej správe odoslaná do mikrokontroléru a spôsobí aktiváciu čítačky SD kariet. Následne je deaktivovaný a zneviditeľnený obrázok ikony vymeniteľného disku. Namiesto toho je aktivovaná ikona preškrtnutého disku. Kliknutím na preškrtnutú ikonu sa premenná `SD_en` nastaví na hodnotu 0 čím sa čítačka SD kariet vypne. Následne sa opäť deaktivuje obrázok s ikonou preškrtnutého vymeniteľného disku a objaví sa pôvodná ikona. Pri každom kliknutí na niektorú z ikon je do premennej `sdtimer` typu `UInt16` uložená hodnota 0. Premenná `sdtimer` je inkrementovaná v rámci obsluhy timeru `icontimer` s periódou 1000ms. V obsluhu `VideoTimer-u` je sledovaná hodnota premennej `sdtimer`. Ak je menšia ako 5, to znamená 5 sekúnd po každej zmene, je do odosielaného obrázku vložená informačná ikona. Zobrazuje či bola čítačka kariet aktivovaná alebo deaktivovaná. K vloženiu ikony do obrazu slúži funkcia `Bitmapicon(Bitmap b, UInt16 index, UInt16 ipos)` ktorá do obrazu `b` vloží ikonu s poradovým číslom `index` na pozíciu `ipos`. Pozícia `ipos` určuje polohu ikony ,

teda jej ofset na Y-ovej súradnici obrazu od počiatku súradníc. Ikona je pred pridaním do obrazu otočená podľa aktuálnej hodnoty v premennej `rotation`.

V pravom hornom rohu sa nachádza ikona pre ovládanie zvukového výstupu monitora a v ľavom dolnom rohu ikona zámku touchscreen. Obe tlačidlá pracujú podobným spôsobom ako tlačidlo ovládajúce čítačku SD kariet. V pravom dolnom rohu je ikona spúšťajúca video prehrávač.

V programe je s periódou 100ms vykonávaný timer s názvom `Mouse`. V ňom je sledovaná aktuálna poloha kurzora myši. Ak dôjde k stlačeniu ľavého tlačidla myši v oblasti náhľadu obrazu, je do premenných `mouse1x` a `mouse1y` ktoré sú dátovými typmi `UInt16` uložené aktuálne súradnice polohy kurzora. Tie sú brané ako východiskové počas doby po ktorú tlačidlo myši ostáva stlačené. Pri stlačení tlačidla sa počas každej periódy timeru spočíta rozdiel medzi aktuálnou polohou a súradnicami uloženými v premenných `mouse1x` a `mouse1y`. Tento rozdiel je uložený do premenných `deltax` a `deltay`. Ďalšie spracovanie závisí od stavu premennej `rotation`. V prípade že je jej obsahom hodnota 0 alebo 2 ďalšie spracovanie vypadá nasledovne.

```
if (posx + deltax - delta2x > (cutsizex * 4) / 2)
posx = Convert.ToUInt16( posx + deltax - delta2x);
else
posx = Convert.ToUInt16( (cutsizex * 4) / 2);
if (posx > Screen.PrimaryScreen.Bounds.Width - (cutsizex * 4)
/ 2)
posx = Convert.ToUInt16(Screen.PrimaryScreen.Bounds.Width -
(cutsizex * 4) / 2);

if (posy + deltay - delta2y > (cutsizex * 3) / 2)
posy = Convert.ToUInt16(posy + deltay - delta2y);
else
posy = Convert.ToUInt16((cutsizex * 3) / 2);
```

```
if (posy > Screen.PrimaryScreen.Bounds.Height - (cutsize *  
3) / 2)  
  
posy = Convert.ToUInt16(Screen.PrimaryScreen.Bounds.Height -  
(cutsize * 3) / 2);
```

Premenná `delta2x` resp. `delta2y` obsahuje hodnotu `deltax` resp. `deltay` z predchádzajúceho vyhodnocovania. Program pre každú os vyhodnotí či sa snímaný obraz nebude nachádzať za hranicou pracovnej plochy. Ak áno, okraj snímaného obrazu bude zarovnaný k hranici pracovnej plochy PC. Pre prípad že premenná `rotation` obsahuje hodnoty 1 alebo 3, postup je obdobný len s rozdielom, že sú otočené pomery dĺžok strán obrazu.

5.7 Nadstavba riadiaceho programu – video prehrávač

Video prehrávač môže byť spustený z menu Notify Icon alebo z riadiaceho programu kliknutím na ikonu prehrávača. V oboch prípadoch dôjde k minimalizácii hlavného okna. Následne je otvorené nové okno prehrávača. Okno je pevne umiestnené v ľavom hornom rohu monitora. Na pozadí okna je umiestnené logo prehrávača. V programe je spustený timer nazvaný `TimerIcon` ktorého perióda je nastavená na 100ms. Počas jeho obsluhy je sledovaný kurzor myši. Pokiaľ sa kurzor nachádza na súradniciach okna prehrávača, je splnená podmienka. V tom prípade sú všetky ovládacie a informačné prvky nastavené ako viditeľné. Ide o tlačidlá „Play“, „Pause“, „Stop“, „Open“, „Forward“, „Backward“, tlačidlo „Close“, label s koncovým a aktuálnym časom a `TimeBar` pre posuv videa. V prípade že kurzor opustí plochu okna prehrávača, všetky spomínané prvky sa stávajú neviditeľnými. V timeri je zároveň aktualizovaný čas prehrávania videa. Screenshot video prehrávača je na obrázku 14.

```
if ((m_objMediaPosition !=
null)&((m_CurrentStatus==MediaStatus.Running)|
(m_CurrentStatus==MediaStatus.Paused)))
{
    int s = (int)m_objMediaPosition.Duration;
    int h = s / 3600;
    int m = (s - (h * 3600)) / 60;
    s = s - (h * 3600 + m * 60);
    labelDuration.Text = String.Format("{0:D2}:{1:D2}:{2:D2}",
h, m, s);
    s = (int)m_objMediaPosition.CurrentPosition;
    h = s / 3600;
    m = (s - (h * 3600)) / 60;
    s = s - (h * 3600 + m * 60);
    labelPosition.Text = String.Format("{0:D2}:{1:D2}:{2:D2}",
h, m, s);}
```

Ak je splnená podmienka že pozícia nie je rovná null, a status videa je Running alebo Paused, sú aktualizované dáta na labeloch labelDuration a labelPosition. V prípade že podmienka splnená nie je, čas na oboch labeloch je rovný „00:00:00“. Ak je načítané video, v závere obsluhy Timeru je aktualizovaná hodnota TimeBaru. Kliknutím na tlačidlo „Open“ je otvorený dialóg výberu súboru. Filter dialógového okna je nastavený na výber podporovaných multimediálnych formátov, umožňuje však výber aj ľubovoľného súboru. Po výbere súboru je cesta k súboru použitá pre otvorenie videa.

```
m_objFilterGraph = new FilgraphManager();
m_objFilterGraph.RenderFile(openFileDialog.FileName);
m_objBasicAudio = m_objFilterGraph as IBasicAudio;
try
{
    m_objVideoWindow = m_objFilterGraph as IVideoWindow;
    m_objVideoWindow.Owner = (int)panel2.Handle;
    m_objVideoWindow.WindowStyle = WS_CHILD | WS_CLIPCHILDREN;
    m_objVideoWindow.SetWindowPosition(panel2.ClientRectangle.Left,
    panel2.ClientRectangle.Top,
    panel2.ClientRectangle.Width,
    panel2.ClientRectangle.Height);
}
catch (Exception)
{
    m_objVideoWindow = null;
}
```

Po načítaní videa je príkazom `m_objMediaControl.Run()`; spustené prehrávanie. Ako maximálna hodnota `TimeBaru` je nastavená dĺžka videa a `TimeBar` je aktivovaný. Do premennej `loaded` je nakopírovaná hodnota 1. Kliknutím na tlačidlo „pause“ je možné prehrávanie videa pozastaviť. Na začiatku kódu je skontrolovaná hodnota premennej `loaded`. Ak je obsahom hodnota 1, je vykonaný príkaz k zastaveniu prehrávania pomocou funkcie `m_objMediaControl.Pause()`; Podobne fungujú aj tlačidlá „stop“ a „pause“. Tlačidlo „forward“ posúva video o 5 sekúnd dopredu. Ak je status prehrávania `running` a aktuálny čas prehrávania videa po pripočítaní 5 sekúnd neprekročí celkovú dĺžku videa, je vykonaný posun videa dopredu o 5 sekúnd. Podobným spôsobom funguje tlačidlo „backwards“, to posunie video o 5 sekúnd dozadu v prípade, že to aktuálny čas videa dovolí. Posunom `TimeBaru` je možné posunúť video na žiadanú pozíciu. Po kliknutí na tlačidlo „X“ je prehrávanie videa zastavené a okno prehrávača skryté pomocou funkcie `this.Hide()`.



Obrázok 14 – screenshot programu video prehrávača

6 ZÁVER

Diplomová práca vychádza z úspešného zvládnutia zadaní Semestrálneho projektu 1 a Semestrálneho projektu 2. Úlohou bolo navrhnuť riešenie dvoch miniatúrnych PC monitorov. Prvý návrh počítal s využitím signál VGA ako zdroja obrazových dát. Návrh obsahuje USB audio prevodník s reproduktorom a výstupom na slúchadlá, touchscreen controller pre pohodlné ovládanie zariadenia a akcelerometer pre úpravu natočenia obrazu. Druhý návrh monitora počíta s použitím zbernice USB pre pripojenie k PC. Tento monitor v sebe integruje USB hub, ktorý umožňuje pripojenie viacerých USB zariadení. Toho využívajú prídavné zariadenia monitoru ako USB audio výstup a čítačka SD kariet. Pre oba návrhy bola vytvorená bloková schéma a schéma zapojenia. K realizácii bol vybraný monitor využívajúci zbernicu USB. V rámci Semestrálneho projektu 2 bola navrhnutá doska plošných spojov, zariadenie bolo fyzicky vyhotovené vložené do plastovej krabičky a bola overená funkčnosť jednotlivých blokov zariadenia. Posledná časť diplomovej práce mala za úlohu vytvoriť softwarové vybavenie monitoru. Vo vývojovom štúdiu CodeWarrior 6.3 bol vytvorený program pre riadiaci mikrokontrolér HCS08 umiestnený v monitore. V prostredí Microsoft Visual C# bol vytvorený PC program riadiaci činnosť zariadenia. Program vyhodnocuje dáta spracované mikrokontrolérom a požiadavky od užívateľa. Pripravené obrazové dáta sú odosielané do prevodníka FTDI a následne zobrazované na displeji. Program umožňuje riadenie prídavných obvodov. Súčasťou programu je zároveň jednoduchý prehrávač ktorý na umožňuje prehrávanie video záznamov vo formáte prispôbenom formátu monitoru. Ďalšou možnosťou rozšírenia programu je vytvorenie virtuálnej pracovnej plochy počítača. Pomocou programu by bolo možné taktiež ovládať kurzor myši a prehrávanie videa. Hmotnosť monitora je približne 150g. Rozlíšenie displeja je 320 x 240 bodov pri farebnej hĺbke 16 bitov. Obnovovacia frekvencia obrazu je 33,3Hz. Prúdový odber pri vypnutej čítačke SD kariet, so zapnutým audio prevodníkom a intenzite podsvietenia 75% je 200mA. Na obrázku 15 možno vidieť funkčné zariadenie v prevádzke.

Projekt realizovaný v rámci Diplomovej práce bol prihlásený do súťaže v rámci konferencie EEICT 2011 usporiadanej na VUT v Brne kde obsadil 1. miesto v kategórii „Elektronika a komunikácie“.



Obrázok 15 – Miniatúrny PC monitor

LITERATÚRA

- [1] FT2232H. FTDI – Future Technology Devices International Ltd., USB devices [online]. 2011 – [cit. 14. prosince 2011]. Dostupné na [www: www.ftdichip.com/Products/ICs/FT2232H.htm](http://www.ftdichip.com/Products/ICs/FT2232H.htm).
- [2] AD9985A. Analog devices – VGA Display Interface [online]. 2011 – [cit. 14. prosince 2011]. Dostupné na [www: http://www.analog.com/en/audiovideo-products/analoghdmi-interfaces/ad9985/products/product.html](http://www.analog.com/en/audiovideo-products/analoghdmi-interfaces/ad9985/products/product.html).
- [3] PCM2705. Texas instruments – STEREO AUDIO DAC WITH USB INTERFACE, SINGLE-ENDED HEADPHONE OUTPUT AND S/PDIF OUTPUT [online]. 2011 – [cit. 17. prosince 2011]. Dostupné na [www: http://www.ti.com/lit/ds/symlink/pcm2705.pdf](http://www.ti.com/lit/ds/symlink/pcm2705.pdf).
- [4] ISP1520BD. Philips semiconductor – Hi-Speed USB hub controller [online]. 2011 – [cit. 14. prosince 2011]. Dostupné na [www: http://www.datasheetcatalog.org/datasheet/philips/ISP1520BD.pdf](http://www.datasheetcatalog.org/datasheet/philips/ISP1520BD.pdf).
- [5] XC9536. Xilinx – In-System Programmable CPLD [online]. 2011 – [cit. 14. prosince 2011]. Dostupné na [www: http://cnx.org/content/m11678/latest/XC9536.pdf](http://cnx.org/content/m11678/latest/XC9536.pdf).
- [6] DRM104. Freescale semiconductor - SD Card Reader Using the M9S08JM60 Series [online]. 2011 – [cit. 14. prosince 2011]. Dostupné na [www: http://cache.freescale.com/files/microcontrollers/doc/ref_manual/DRM104.pdf](http://cache.freescale.com/files/microcontrollers/doc/ref_manual/DRM104.pdf).
- [7] HX8347-A. Himax Technologies, Inc. - 240RGB x 320 dot, 262K color, with internal GRAM, TFT Mobile Single Chip Driver [online]. 2007 – [cit. 05. Kětna 2012]. Dostupné na [www: www.allshore.com/pdf/himaxHX8347A.pdf](http://www.allshore.com/pdf/himaxHX8347A.pdf).

ZOZNAM SKRATIEK

LCD	Liquid Crystal Display
VGA	Video Graphics Array
USB	Universal Serial Bus
SRAM	Static Random-access Memory
CPLD	Complex Programmable Logic Device
SD	Secure Digital
LDO	Low-drop Output
LED	Light-emitting Diode
DAC	Digital to Analog Converter
DPS	Doska plošných spojov

ZOZNAM PRÍLOH

A Schéma zapojenia USB monitoru

A.1	List 1	46
A.2	List 2	47

B Schéma zapojenia VGA monitoru

B.1	List 1	48
B.2	List 2	49

C DPS USB monitoru

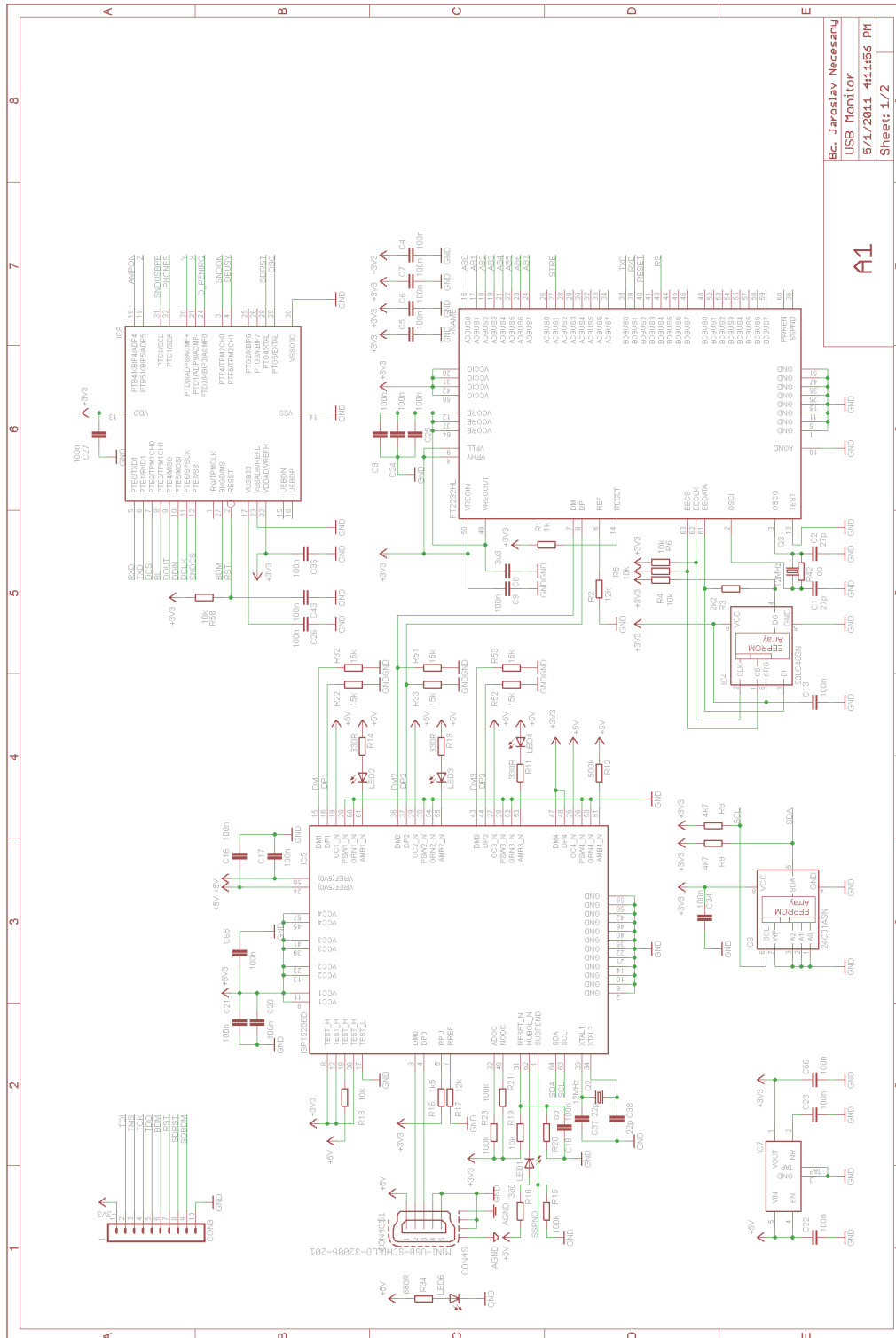
C.1	Top vrstva DPS	50
C.2	Bottom vrstva DPS.....	51
C.3	Osadzovací výkres Top DPS	52
C.4	Osadzovací výkres Bottom DPS	53

D Zoznam súčiastok USB monitoru

D.1	Zoznam súčiastok USB monitoru	54
-----	-------------------------------------	----

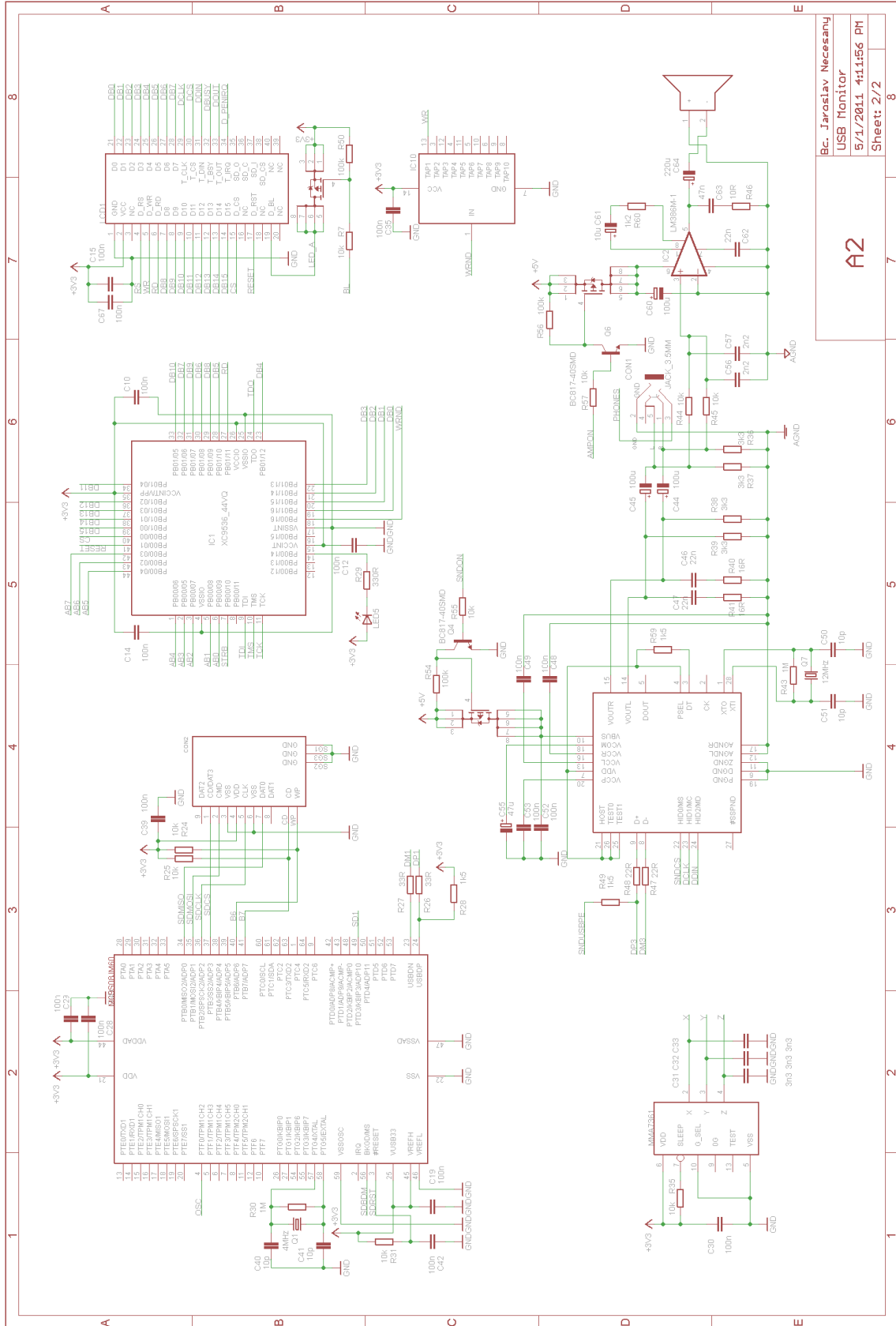
A - SCHÉMA ZAPOJENIA USB MONITORU

List 1



List 1 schéma USB monitoru

List 2



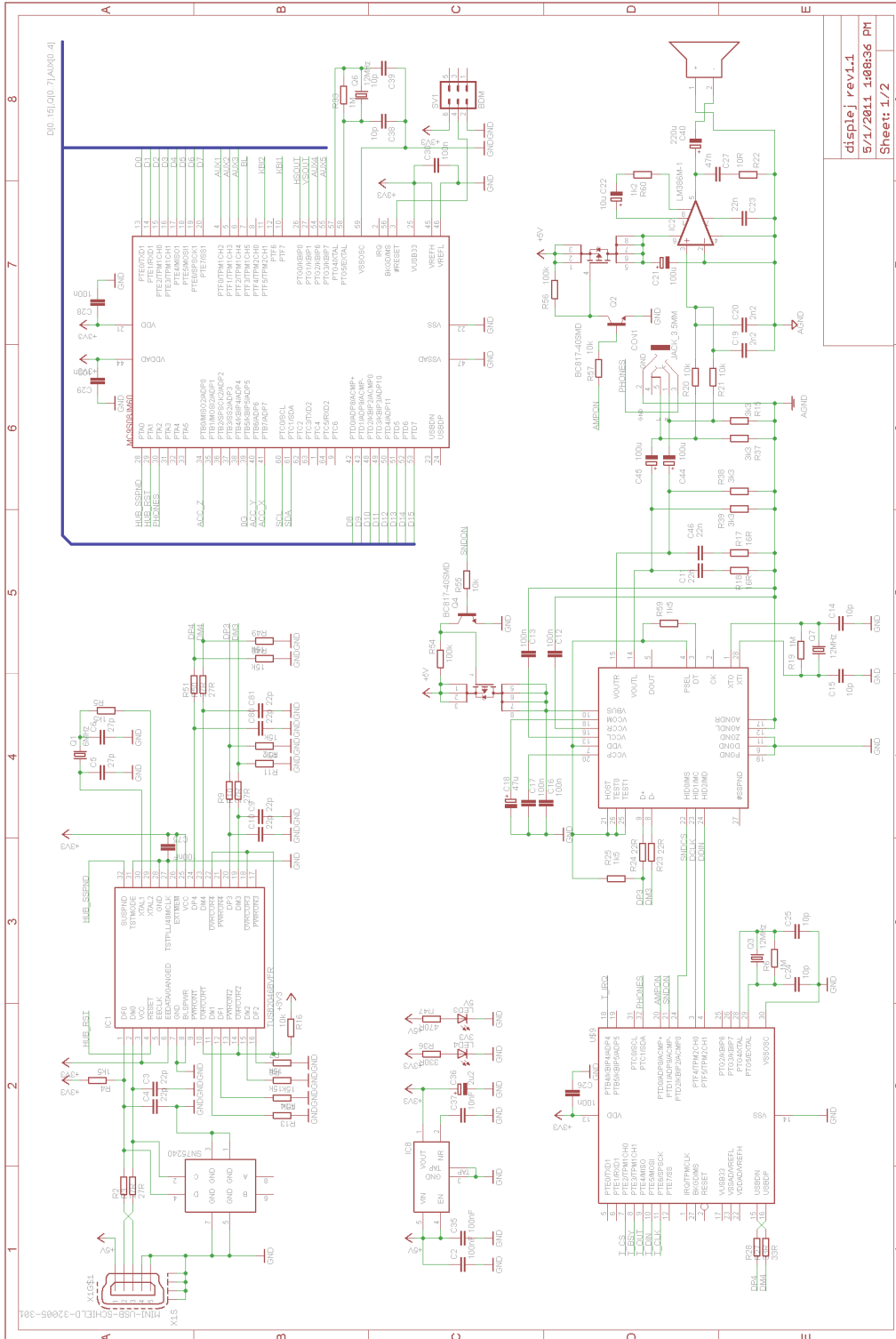
Bc. Jaroslav Necessary
 USB Monitor
 5/1/2011 11:56 PM
 Sheet 2/2

A2

List 2 schéma USB monitoru

B - SCHÉMA ZAPOJENIA VGA MONITORU

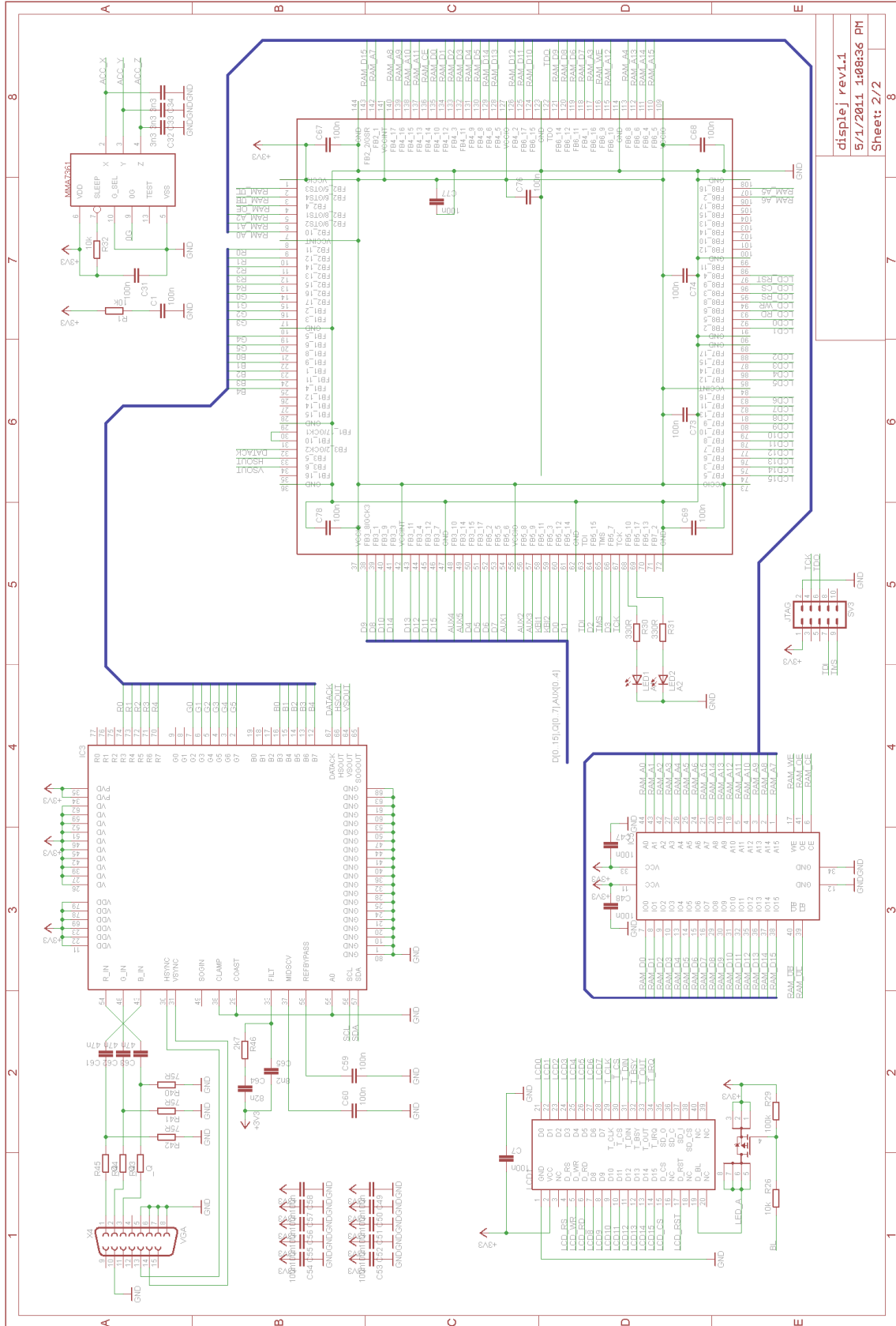
List 1



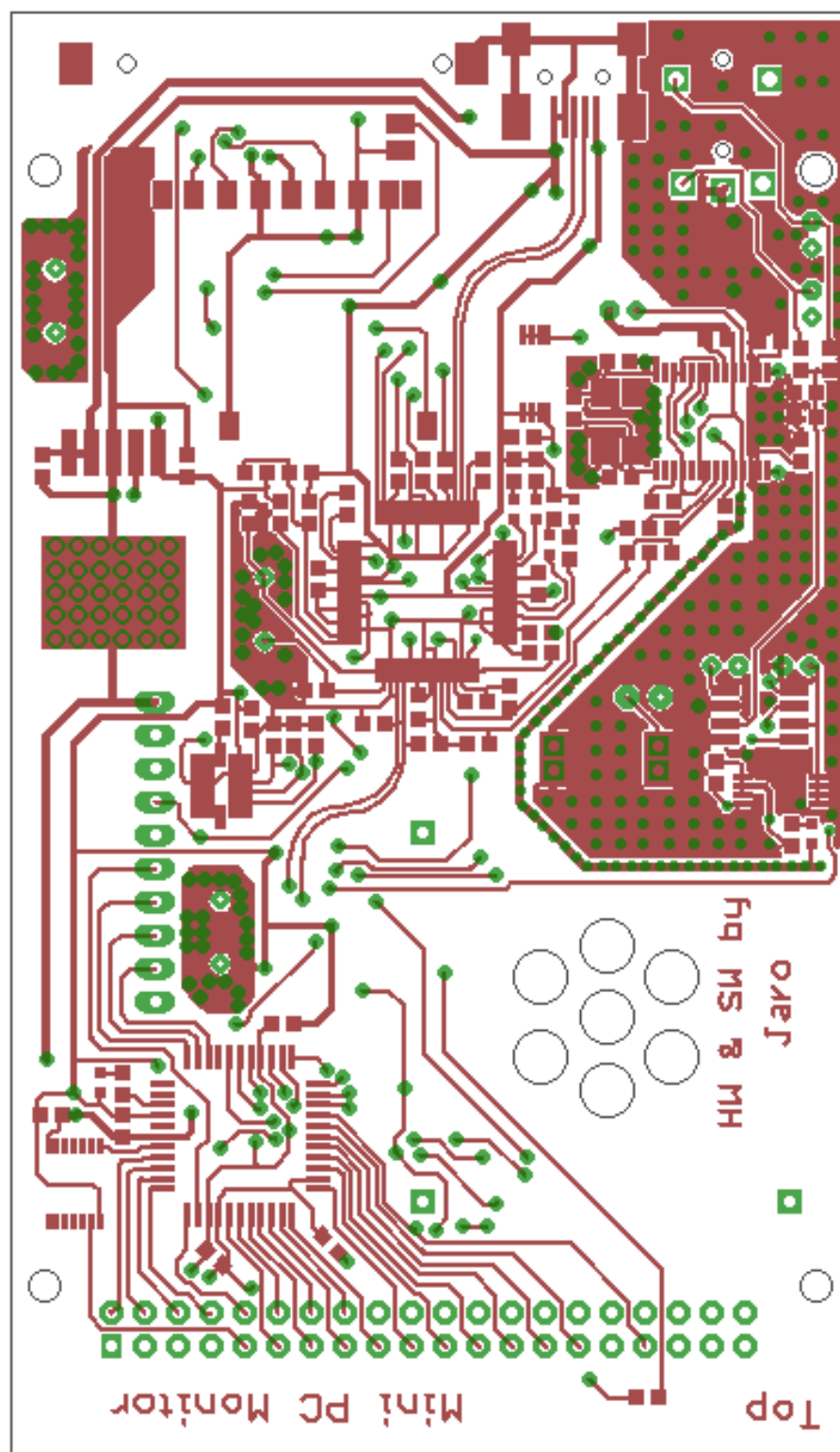
display rev1.1
5/1/2011 1.08636 PH
Sheet: 1/2

List 1 schéma VGA monitoru

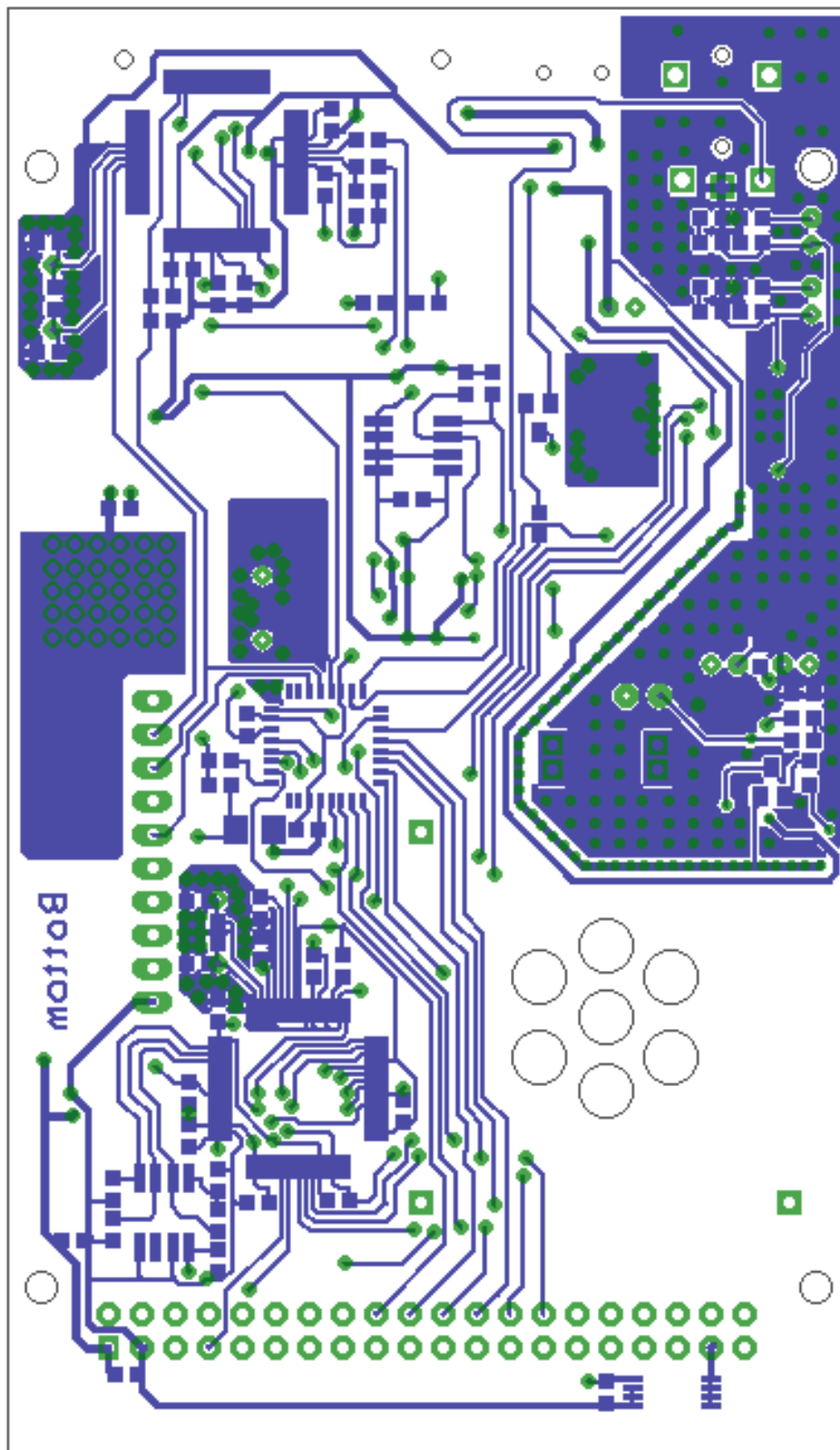
List 2



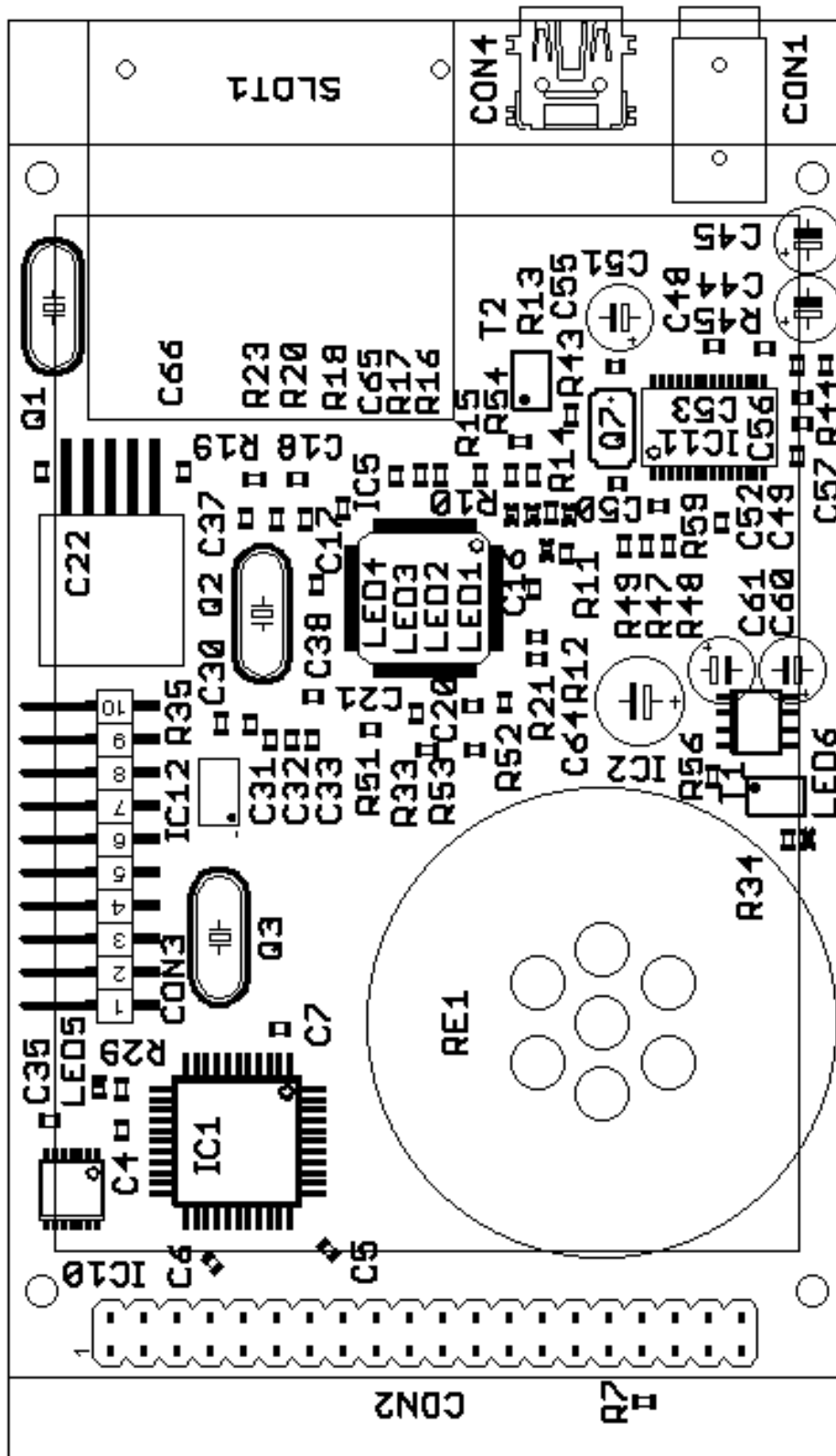
C – DPS USB MONITORU



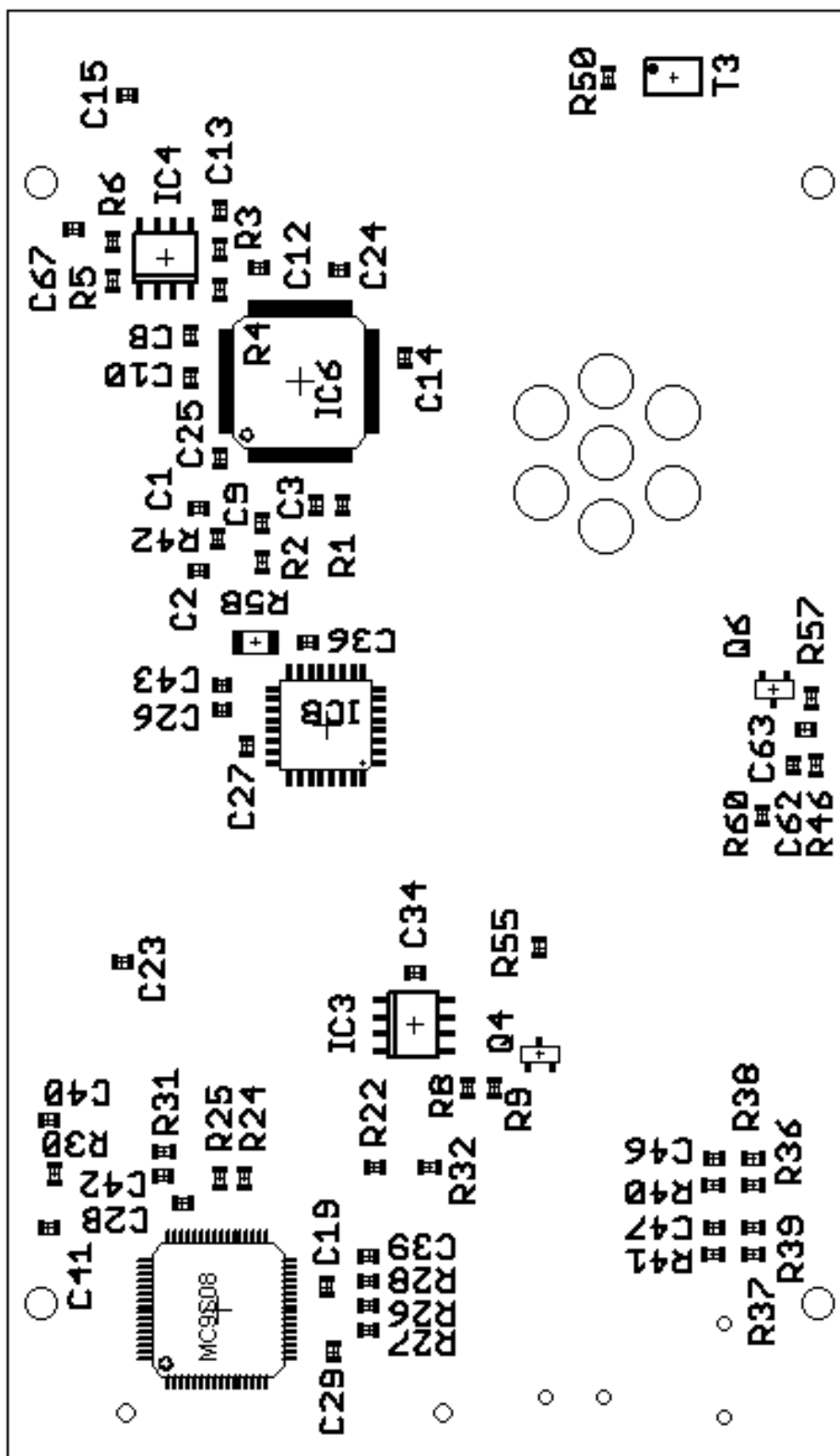
C.1 - Výkres Top vrstvy DPS USB monitoru v merítku 2:1. Rozmer dosky 94x64mm.



C.2 -Výkres Bottom vrstvy DPS USB monitoru v měřítku 2:1. Rozměr dosky 94x64mm.



C.3 - Osadzovací výkres Top vrstvy USB monitoru v merítku 2:1. Rozmer dosky 94x64mm.



C.4 - Osadzovací výkres Bottom vrstvy USB monitoru v měřítku 2:1 (stranovo prevrätaný). Rozmer dosky 94x64mm.

D – ZOZNAM SÚČIASTOK USB MONITORU

Part	Value	Device	Package	Library	Sheet
C1	27p	C-EUC0603	C0603	rc1	1
C2	27p	C-EUC0603	C0603	rc1	1
C3	100n	C-EUC0603	C0603	rc1	1
C4	100n	C-EUC0603	C0603	rc1	1
C5	100n	C-EUC0603	C0603	rc1	1
C6	100n	C-EUC0603	C0603	rc1	1
C7	100n	C-EUC0603	C0603	rc1	1
C8	3u3	C-EUC0603	C0603	rc1	1
C9	100n	C-EUC0603	C0603	rc1	1
C10	100n	C-EUC0603	C0603	rc1	2
C12	100n	C-EUC0603	C0603	rc1	2
C13	100n	C-EUC0603	C0603	rc1	1
C14	100n	C-EUC0603	C0603	rc1	2
C15	100n	C-EUC0603	C0603	rc1	2
C16	100n	C-EUC0603	C0603	rc1	1
C17	100n	C-EUC0603	C0603	rc1	1
C18	100n	C-EUC0603	C0603	rc1	1
C19	100n	C-EUC0603	C0603	rc1	2
C20	100n	C-EUC0603	C0603	rc1	1
C21	100n	C-EUC0603	C0603	rc1	1
C22	100n	C-EUC0603	C0603	rc1	1
C23	100n	C-EUC0603	C0603	rc1	1
C24	100n	C-EUC0603	C0603	rc1	1
C25	100n	C-EUC0603	C0603	rc1	1
C26	100n	C-EUC0603	C0603	rc1	1
C27	100n	C-EUC0603	C0603	rc1	1
C28	100n	C-EUC0603	C0603	rc1	2
C29	100n	C-EUC0603	C0603	rc1	2
C30	100n	C-EUC0603	C0603	rc1	2

C31	3n3	C-EUC0603	C0603	rcl	2
C32	3n3	C-EUC0603	C0603	rcl	2
C33	3n3	C-EUC0603	C0603	rcl	2
C34	100n	C-EUC0603	C0603	rcl	1
C35	100n	C-EUC0603	C0603	rcl	2
C36	100n	C-EUC0603	C0603	rcl	1
C37	22p	C-EUC0603	C0603	rcl	1
C38	22p	C-EUC0603	C0603	rcl	1
C39	100n	C-EUC0603	C0603	rcl	2
C40	10p	C-EUC0603	C0603	rcl	2
C41	10p	C-EUC0603	C0603	rcl	2
C42	100n	C-EUC0603	C0603	rcl	2
C43	100n	C-EUC0603	C0603	rcl	1
C44	100u	CPOL-EUE2-5	E2-5	rcl	2
C45	100u	CPOL-EUE2-5	E2-5	rcl	2
C46	22n	C-EUC0603	C0603	rcl	2
C47	22n	C-EUC0603	C0603	rcl	2
C48	100n	C-EUC0603	C0603	rcl	2
C49	100n	C-EUC0603	C0603	rcl	2
C50	10p	C-EUC0603	C0603	rcl	2
C51	10p	C-EUC0603	C0603	rcl	2
C52	100n	C-EUC0603	C0603	rcl	2
C53	100n	C-EUC0603	C0603	rcl	2
C55	47u	CPOL-EUE2-5	E2-5	rcl	2
C56	2n2	C-EUC0603	C0603	rcl	2
C57	2n2	C-EUC0603	C0603	rcl	2
C60	100u	CPOL-EUE2-5	E2-5	rcl	2
C61	10u	CPOL-EUE2-5	E2-5	rcl	2
C62	22n	C-EUC0603	C0603	rcl	2
C63	47n	C-EUC0603	C0603	rcl	2
C64	220u	CPOL-EUE2.5-7	E2,5-7	rcl	2
C65	100n	C-EUC0603	C0603	rcl	1
C66	100n	C-EUC0603	C0603	rcl	1
C67	100n	C-EUC0603	C0603	rcl	2

CON1	JACK_3.5MM	JACK_3.5MM	CON	audio_con	2
CON2	SD	SD	SD	SD	2
CON3	1	MA10-1W	MA10-1W	con-lstb	1
CON4	MINI-USB	MINI-USB	con-cypressindustries		1
IC1	XC9536_44VQ	XC9536_44VQ	SQFP-S-10X10-44		2
IC2	LM386M-1	LM386M-1	SO08	linear	2
IC3	24C01ASN	24C01ASN	SO-08	microchip	1
IC4	93LC46SN	93LC46SN	SO-08	microchip	1
IC5	ISP1520BD	ISP1520BD	LQFP64	ISP1520BD	1
IC6	FT2232HL	FT2232HL	LQFP64	FT2232HL	1
IC7	REG104	REG104	DDPAK5	REG104	1
IC8	MC9S08JM16	MC9S08JM16	LQFP32	MC9S08JM16	1
IC9	MC9S08JM60	MC9S08JM60	LQFP64	Freescale	2
IC10	DS1110L	DS1110L	TSSOP14	ds1110l	2
IC11	PCM2704	PCM2704	SSOP28	pcm270x	2
IC12	MMA7361	MMA7361	LGA14	mma7361	2
LCD1		LCD	LCD	LCD	2
LED1	LEDCHIPLED_0603	CHIPLED_0603	led		1
LED2	LEDCHIPLED_0603	CHIPLED_0603	led		1
LED3	LEDCHIPLED_0603	CHIPLED_0603	led		1
LED4	LEDCHIPLED_0603	CHIPLED_0603	led		1
LED5	LEDCHIP-LED0603	CHIP-LED0603	led		2
LED6	LEDCHIPLED_0603	CHIPLED_0603	led		1
Q1	4MHz	CRYSTALHC49S	HC49/S	crystal	2
Q2	12MHz	CRYSTALHC49S	HC49/S	crystal	1
Q3	12MHz	CRYSTALHC49S	HC49/S	crystal	1
Q4	BC817-40SMD	BC817-40SMD	SOT23-BEC	transistor-npn	2
Q6	BC817-40SMD	BC817-40SMD	SOT23-BEC	transistor-npn	2
Q7	12MHz	CRYSTALCTS406	CTS406	crystal	2
R1	1k	R-EU_R0603	R0603	rcl	1
R2	12k	R-EU_R0603	R0603	rcl	1
R3	2k2	R-EU_R0603	R0603	rcl	1
R4	10k	R-EU_R0603	R0603	rcl	1
R5	10k	R-EU_R0603	R0603	rcl	1

R6	10k	R-EU_R0603	R0603	rcl	1
R7	10k	R-EU_R0603	R0603	rcl	2
R8	4k7	R-EU_R0603	R0603	rcl	1
R9	4k7	R-EU_R0603	R0603	rcl	1
R10	330	R-EU_R0603	R0603	rcl	1
R11	330R	R-EU_R0603	R0603	rcl	1
R12	500k	R-EU_R0603	R0603	rcl	1
R13	330R	R-EU_R0603	R0603	rcl	1
R14	330R	R-EU_R0603	R0603	rcl	1
R15	100k	R-EU_R0603	R0603	rcl	1
R16	1k5	R-EU_R0603	R0603	rcl	1
R17	12k	R-EU_R0603	R0603	rcl	1
R18	10k	R-EU_R0603	R0603	rcl	1
R19	10k	R-EU_R0603	R0603	rcl	1
R20	oo	R-EU_R0603	R0603	rcl	1
R21	100k	R-EU_R0603	R0603	rcl	1
R22	15k	R-EU_R0603	R0603	rcl	1
R23	100k	R-EU_R0603	R0603	rcl	1
R24	10k	R-EU_R0603	R0603	rcl	2
R25	10k	R-EU_R0603	R0603	rcl	2
R26	33R	R-EU_R0603	R0603	rcl	2
R27	33R	R-EU_R0603	R0603	rcl	2
R28	1k5	R-EU_R0603	R0603	rcl	2
R29	330R	R-EU_R0603	R0603	rcl	2
R30	1M	R-EU_R0603	R0603	rcl	2
R31	10k	R-EU_R0603	R0603	rcl	2
R32	15k	R-EU_R0603	R0603	rcl	1
R33	15k	R-EU_R0603	R0603	rcl	1
R34	680R	R-EU_R0603	R0603	rcl	1
R35	10k	R-EU_R0603	R0603	rcl	2
R36	3k3	R-EU_R0603	R0603	rcl	2
R37	3k3	R-EU_R0603	R0603	rcl	2
R38	3k3	R-EU_R0603	R0603	rcl	2
R39	3k3	R-EU_R0603	R0603	rcl	2

R40	16R	R-EU_R0603	R0603	rcl	2
R41	16R	R-EU_R0603	R0603	rcl	2
R42	oo	R-EU_R0603	R0603	rcl	1
R43	1M	R-EU_R0603	R0603	rcl	2
R44	10k	R-EU_R0603	R0603	rcl	2
R45	10k	R-EU_R0603	R0603	rcl	2
R46	10R	R-EU_R0603	R0603	rcl	2
R47	22R	R-EU_R0603	R0603	rcl	2
R48	22R	R-EU_R0603	R0603	rcl	2
R49	1k5	R-EU_R0603	R0603	rcl	2
R50	100k	R-EU_R0603	R0603	rcl	2
R51	15k	R-EU_R0603	R0603	rcl	1
R52	15k	R-EU_R0603	R0603	rcl	1
R53	15k	R-EU_R0603	R0603	rcl	1
R54	100k	R-EU_R0603	R0603	rcl	2
R55	10k	R-EU_R0603	R0603	rcl	2
R56	100k	R-EU_R0603	R0603	rcl	2
R57	10k	R-EU_R0603	R0603	rcl	2
R58	10k	R-EU_M1206	M1206	rcl	1
R59	1k5	R-EU_R0603	R0603	rcl	2
R60	1k2	R-EU_R0603	R0603	rcl	2
REP1	REPRO	REPRO	REPRO	repro	2
T1	TPS1100	TPS1100	TSSOP8	TPS1100	2
T2	TPS1100	TPS1100	TSSOP8	TPS1100	2
T3	TPS1100	TPS1100	TSSOP8	TPS1100	2