Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

LCD displej s bezdrátovým rozhraním

Plzeň 2012 Vladimír Mandák

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 7. května 2012 Vladimír Mandák.....

Poděkování

Na tomto místě bych rád poděkoval vedoucímu bakalářské práce Dr. Ing. Karlu Dudáčkovi za připomínky a cenné rady. Dále bych chtěl poděkovat rodině a všem, kteří mě podporovali v mé práci.

Abstract

In this work I am dealing with design of the circuit for receiving data from the wireless Bluetooth interface and their subsequention on the LCD screen equipped with Epson S1D13305 controller. LCD display also features a touch panel. For scanning of coordinates of touch is used ADS7846 circuit connected to the SPI interface. I took advantage of the specialized integrated circuits, which I added components required to implement wireless communications and scanning touch coordinates on the LCD.

Keywords: microcontroller, Bluetooth, SPI, RS-232C, LCD display, instruction set, protocol, touchscreen

Abstrakt

V této práci se zabývám návrhem obvodu pro příjem dat z bezdrátového rozhraní Bluetooth a jejich následným zobrazením na LCD displej vybavený řadičem EPSON S1D13305. LCD displej je navíc vybaven dotykovou deskou. Ke snímání souřadnic dotyku je použit obvod ADS7846 připojený k rozhraní SPI. Využil jsem specializovaných integrovaných obvodů, které jsem doplnil obvody potřebnými pro realizaci bezdrátové komunikace a snímání dotyku souřadnic na LCD displeji.

Klíčová slova: mikrokontrolér, Bluetooth, SPI, RS-232C, LCD displej, instrukční sada, protokol, dotyková deska

Obsah

1 Úvod

Mým úkolem v této práci je navrhnout obvod, který přijímá data pomocí bezdrátového rozhraní Bluetooth nebo ZigBee. Ty pak zobrazuje na vhodně vybraný LCD displej. Řídícím prvkem celého navrženého obvodu je mikrokontrolér. Mikrokontrolér je třeba pečlivě vybrat s ohledem na další obvody, které budou připojeny. Celý obvod je ovládán programem uloženým v paměti mikrokontroléru. Program zpracovává na vstupu příkazy přicházející z bezdrátového modulu a v závislosti na významu příkazu pak program určí, jaké operace mají být s displejem provedeny. Displej je navíc vybaven dotykovou deskou. Ta má vyvedenu čtveřici vodičů, které se připojují ke speciálnímu obvodu, snímající pozici dotyku na desce.

Softwarová část práce má za cíl vytvoření komunikačního rozhraní mezi uživatelem a řadičem LCD displeje. Programové zpracování rozsáhlého instrukčního souboru řadiče displeje má mikrokontrolér uložen v paměti kódu. Není tedy potřeba studovat rozsáhlý soubor příkazů pro řadič displeje ale pouze formát malého množství příkazů. Přijatá data od uživatele obsahují číselný kód příkazu a případný parametr. Přijatá data vyhodnotí mikrokontrolér a provede potřebné operace s řadičem. Dalším úkolem je průběžná detekce dotyku na displeji a zjištění jeho souřadnic. Hodnoty souřadnic pak program posílá sériovou linkou (nebo pomocí Bluetooth) směrem k uživateli. S tím souvisí vytvoření protokolu přenášejícího informace o kontaktu na dotykové desce.

2 Sériový přenos dat

V praxi rozlišujeme dva základní typy sériového přenosu. Synchronní a arytmický. Synchronní režim vyžaduje přítomnost hodinového signálu, který potvrzuje platnost jednotlivých bitů na datových vodičích.
Arytmický režim na rozdíl od synchronního odesílá synchronizační informaci uloženou v datech. Podmínkou správného arytmického přenosu je, že vysílač a přijímač pracují na stejné taktovací frekvenci hodin. Synchronizační informace sestává ze start bitu (začátek dat) a ze stop bitu (konec dat). Součástí přenášených dat může být parita. Mezi jednotlivými odeslanými znaky může byt libovolně dlouhá prodleva. Neexistence jakéhokoliv rytmu objasňuje význam slova arytmický (postrádající rytmus).
Existuje ještě jeden druh přenosu a to asynchronní. Někdy se tak nesprávně označuje arytmický přenos. V praxi se však asynchronní přenos nepoužívá.

2.1 Sériové rozhraní RS-232C

V dnešní době už je toto rozhraní používáno spíše v průmyslových odvětvích. Z osobních počítačů bylo rozhraní vytlačeno univerzální sběrnicí USB. Maximální rychlost rozhraní RS-232C je závislá na délce kabelu a na hodnotách napěťových úrovní. Čím vyšší je použité napětí, tím delší je možná vzdálenost pro přenos, při zachování konstantní přenosové rychlosti.

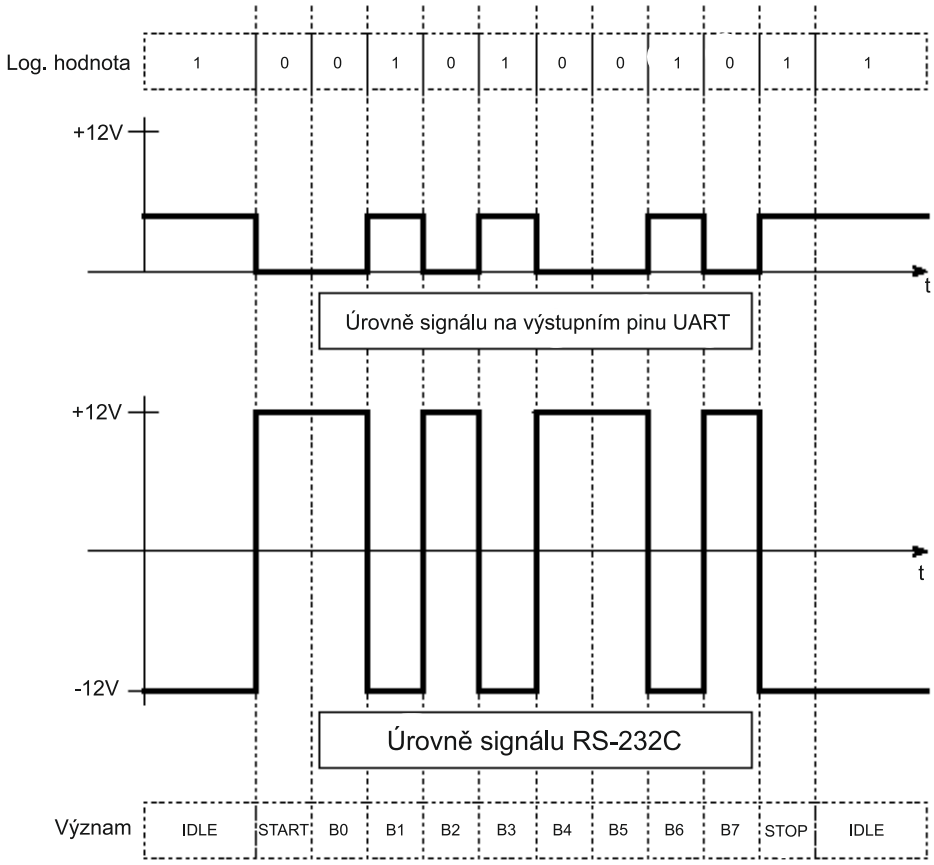
2.1.1 Napěťové úrovně

Co se týče elektrických vlastností, je rozhraní RS-232C odlišné od úrovní TTL. Standard definuje bipolární hodnoty napětí, kde napěťová úroveň v rozsahu -15 V až -3 V je považována za logickou jedničku (stav *MARK*) a napětí v rozsahu +3 V až +15 V za logickou nulu (stav *SPACE*). Tyto úrovně jsou omezeny maximální hodnotou vnějšího napájení.

2.1.2 Přenos dat

RS-232C pracuje pouze arytmicky. Před zahájením samotného přenosu musejí mít vysílač i přijímač nastaveny stejné parametry komunikace (rychlost, počet datových bitů, parita, stop bit a řízení toku). Odesílaným datům předchází startovací bit, který druhou stranu upozorní o přicházející komunikaci. Druhá strana začne vzorkovat datové bity přednastavenou rychlostí. Vysílaný byte má většinou délku osm bitů.

Datové bity mohou být doplněny o paritu. Komunikace končí po přijetí jednoho či dvou stop bitů. Příklad průběhu přenosu dat je na obr. 2.1. Podrobnější informace o rozhraní RS-232C jsou k nalezení v [1] nebo [2].



Obrázek 2.1: Odeslání znaku „J“ bez parity s jedním STOP bitem. [3]

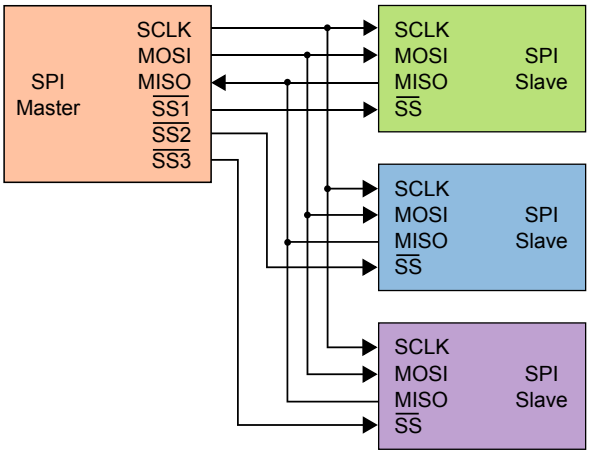
2.2 Rozhraní SPI

SPI je synchronní sériové rozhraní. Pracuje na principu master/slave, kde jediné zařízení typu master funguje jako řadič a ostatní zařízení jsou mu podřízena. Používá se jako komunikační rozhraní mezi řídícími mikrokontroléry nebo paměťovými řadiči.

2.2.1 Zapojení

Zařízení master obsluhuje několik signálových vodičů. Ty se dají rozdělit na řídící a datové. Mezi řídící patří hodinový signál SCK (SPI Clock). Ten je připojen na všechny podřízené obvody a přenáší synchronizační signál. Výběr slave obvodu je možný pomocí signálů $\overline{SS}$ (Slave Select). Každý slave obvod má vyhrazen jeden

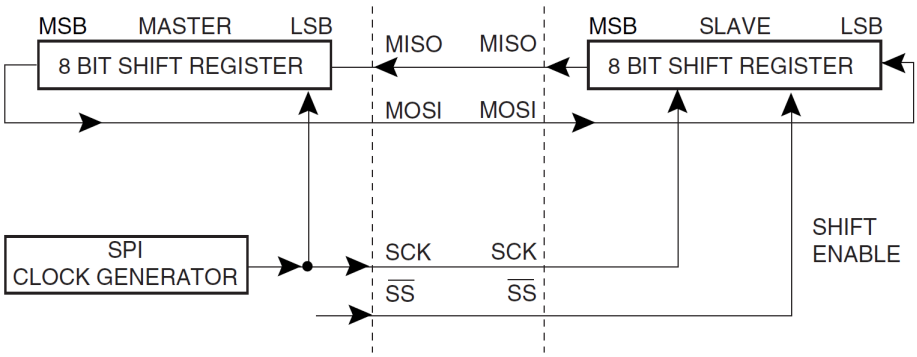
signál $\overline{SS}$. Datové signály jsou dva. Výstup z mikrokontroléru $\overline{MOSI}$ (Master Out Slave In) a vstup $\overline{MISO}$ (Master In Slave Output). Blokové schéma zapojení je na obr. 2.2.



Obrázek 2.2: Zapojení zařízení master a slave. [4]

2.2.2 Přenos dat

Na obrázku 2.3 je blokové schéma komunikace mezi zařízením master a slave. Obě zařízení obsahují 8-bitový posuvný registr. Posuvný registr má k sobě ještě jeden záchytný registr sloužící jako fronta. Zařízení master zahajuje komunikaci aktivací signálu $\overline{SS}$ a začne vysílat hodinový signál SCK. Během komunikace se odesílá z obou zařízení nejvýznamnější bit, který přechází do posuvného registru druhého zařízení jako nejméně významný bit. Rychlost přenosu dat určuje frekvence hodinového signálu nastavená generátorem.



Obrázek 2.3: Princip komunikace SPI. [5]

2.3 Bezdrátová sériová rozhraní

Pro mou práci jsem vybíral ze dvou bezdrátových rozhraní, které lze snadno přidat do návrhu celého obvodu a má rozumné požadavky na konfiguraci a programování. Co se týče spotřeby, mělo by zařízení být schopno pracovat pod napětím +5 V, kterým je celá deska napájena.

2.3.1 ZigBee

Specifikace rozhraní ZigBee (standard IEEE 802.15.4) byla vydána v roce 2004. Jedná se o nové rozhraní, jehož smyslem má být doplnění mezer mezi WiFi a Bluetooth. Na rozdíl od Bluetooth má větší dosah i přes různé překážky. V porovnání s technologií WiFi má ZigBee několikanásobně nižší energetické požadavky. Díky nízké energetické náročnosti se přednostně využívá v zařízeních napájenými bateriemi. Rozhraní je navrženo s orientací na využití v průmyslových odvětvích. Maximální přenosová rychlost je 250 kb/s. Rychlost se může zdát relativně nízká, ovšem pro účely dálkového spínání/vypínání nebo přenos řídících signálů plně postačuje. Oproti rozhraní Bluetooth poskytuje ZigBee tyto zajímavé vlastnosti:

- Větší odolnost vůči rušení v průmyslovém prostředí.
- Využití ad-hoc[1] spojení mezi jednotlivými zařízeními lze propojit dvě zařízení aniž by se koncová zařízení viděla.
- Maximální počet uzlu je 65 000.
- Maximální dosah zařízení je 100m.

Bližší informace o rozhraní ZigBee [6].

2.3.2 Bluetooth

Rozhraní Bluetooth (standard IEEE 802.15.1) je oproti ZigBee daleko známější díky masovému proniknutí do oblasti mobilních zařízení, zejména pak mobilních telefonů a notebooků. Bluetooth bylo vyvinuto jako bezdrátová náhrada za RS-232C. Přenosová rychlost rozhraní se liší podle verze protokolu. Ve verzi 2.0 EDR je propustnost dat 720 kb/s. Nejnovější specifikace rozhraní 4.0 slibuje přenosovou rychlost až 24 Mbit/s při dosahu 100 metrů s podporou šifrování. Komunikaci mezi zařízeními musí předcházet párování. Tento proces zajišťuje vytvoření datového kanálu.

[1] Označení pro dočasné spojení dvou prvků v počítačové síti.

Párování lze zabezpečit pomocí identifikačních čísel, případně textovým klíčem. Podrobnější popis rozhraní viz [1] nebo [2].
Vybral jsem si rozhraní Bluetooth právě z důvodu jeho rozšíření. Návrh mého obvodu může být rozšířen i o rozhraní ZigBee.

3 Výběr mikroprocesoru

Řídícím prvkem celého obvod je mikrokontrolér. Na jeho výběru závisí celková složitost architektury navrženého obvodu. Proto jsem vybíral z vícero procesorů tak, aby byly maximálně využity vlastnosti vybraného mikrokontroléru. Každý z nich má svoje výhody i nevýhody oproti ostatním. Požadavkem byla integrace rozhraní SPI, USART a JTAG. S podporou rozhraní JTAG souvisí i možnosti programování a ladění. Další kritérium byla jednoduchá a srozumitelná práce se vstupně/výstupními linkami. Požadavkům nejvíce vyhovoval mikrokontrolér ATmega32 od firmy Atmel.

3.1 Renesas H8S/2633

Procesor H8S/2633 jsem nevybral z několika důvodů. Prvním důvodem je výpočetní výkon. Vzhledem k relativní jednoduchosti navrženého obvodu i ovládacího programu by velká část výkonu zůstala nevyužita. Dalším důvodem pro mě byla nepříliš dobrá zkušenost s tímto procesorem i s jeho architekturou. Procesor navíc nemá integrovaný řadič rozhraní SPI.

3.2 NEC V850E

Ve školní laboratoři jsem mohl využít i procesor V850 od firmy NEC. Původně byla architektura procesoru vyvíjena pro embeded zařízení. Důvody proč jsem nevybral tento typ procesoru jsou v podstatě shodné jako u H8S. Navíc existuje několik variant přičemž každá má své zvláštnosti a odlišné chování.

3.3 Atmel ATmega32

Pro účely mé práce nejvíce vyhovuje procesor ATmega32. Procesor lze programovat vyšším programovacím jazykem C, což je společně s širokou škálou dokumentačních materiálů velká výhoda. Odpadá tak nutnost programovat vše v jazyce symbolických adres.
Charakteristické vlastnosti procesoru:

- Pracuje na maximálním taktu 16 MHz.
- Pokročilá 8bitová RISC architektura.
- 32 programovatelných vstupně/výstupních linek.
- Vnitřní paměť SRAM o kapacitě 2 kB.
- Paměť EEPROM o kapacitě 1 kB.
- Programová Flash paměť o kapacitě 32 kB.
- Podpora programovacího a debugg rozhraní JTAG.
- Integrovaná řadiče pro podporu rozhraní USART, SPI, A/D převodník, I2C a JTAG.
- Pracovní napětí 4,5 V - 5,5 V.

Jedná se o 8bitový mikroprocesor postavený na architektuře AVR s redukovanou instrukční sadou. Procesor může být programován třemi způsoby:

JTAG: V práci využívám tento způsob kvůli vestavěné podpoře rozhraní přímo v procesoru a dostupnosti programátoru ve školní laboratoři.

Sériově: Skrze rozhraní SPI.

Paralelně: Připojením napětí 12 V na RESET má za následek připojení adresních a datových vodičů vnitřní paměti na piny vstupně/výstupních portů.

ATmega32 pracuje se čtyřmi osmibitovými I/O porty. Každý z nich má navíc možnost speciálního využití:

PORT A: Všech osm pinů slouží jako vstup A/D převodníku.

PORT B: Integrovaný řadič sběrnice SPI.

PORT C: Piny PC0 a PC1 jsou připojeny k řadiči rozhraní TWI. Bity PC4 až PC7 mají funkci rozhraní JTAG.

PORT D: Piny PD0 a PD1 vedou datové signály RxD a TxD modulu UART. Ostatní piny mají funkci vnějších přerušení, případně interních čítačů a časovačů.

Pro práci s porty je určena trojice registrů, kde x je písmeno portu (A, B, C, D):

PORTx (Data Register) Datový registr ukládající data k odeslání.

DDRx (Data Direction Register) Nastavuje pin jako vstupní nebo výstupní.

PINx (Port Input Pis) Pouze ke čtení hodnot pinů na portu. Není nijak závislý na předchozích dvou registrech.

Každý bit registru DDRx nastavuje pin jako výstupní pokud je na daný bit zapsána log. 1. V případě, že je na daném bitu registru log. 0, je pin nastaven jako vstupní. Registr PORTx má pak využití, při nastavování logických hodnot na konkrétním výstupním pinu portu. Pokud je bit registru PORTx nastaven na log. 1 a zároveň je odpovídající DDRx bit nastaven na log. 1, bude na výstupním pinu hodnota log. 1. Registr PINx uchovává logické hodnoty na pinech přicházející z připojených periferních obvodů. Příklad pracovních registrů portu C je na obr. 3.1.

Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0	
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Obrázek 3.1: Pracovní registry portu C. [5]

4 Bezdrátový modul CB-OEMSPA-311i

OEMSPA311 je kompaktní bezdrátový Bluetooth přijímač. Deska přijímače je osazena interní SMD anténou se ziskem +0,5 dB, oscilátorem generující hodinový signál o frekvenci 13 MHz a dalšími obvody. Hlavní součástkou je řídící obvod s integrovanou SRAM a Flash pamětí. Modul se konfiguruje pomocí AT příkazů. Režim komunikace lze přepínat do 16 různých komunikačních profilů. Jedním z nich je profil sériové linky, ve kterém se modul chová jako rozhraní RS-232C. UART bezdrátového přijímače má vyvedeny všechny signály na pinech vlastní desky. Díky tomu je možné připojit bezdrátový modul k mikrokontroléru ATmega32. V práci využívám nejjednoduššího možného zapojení pouze se signály RxD a TxD.
Zapojení modulu jsem doplnil kontrolními diodami signalizujícími činnost přijímače. Schématické zapojení diod je v aplikačních poznámkách manuálu [7]. LED diody signalizují, v jakém stavu se bezdrátový modul právě nachází. Tyto stavy popisuje tab. 1. LED dioda je aktivní (svítí) ve stavu log. 0.

Mód	Stav	Zelená	Modrá	Červená
Datový	Nečinný	0	1	1
AT	Nečinný	0	1	0
Datový, AT	Připojování	1	0	0
Datový, AT	Připojen	1	0	1

Tabulka 1: Význam signalizace stavů LED diodami.

5 Výběr displeje

Řadič jsem vybíral s ohledem na zobrazovací schopnosti připojeného displeje. Dále jsem zvážil zda řadič umí pracovat v textovém i grafickém režimu. Přednost dostal displej PG320240-D vybavený řadičem EPSON S1D13305.

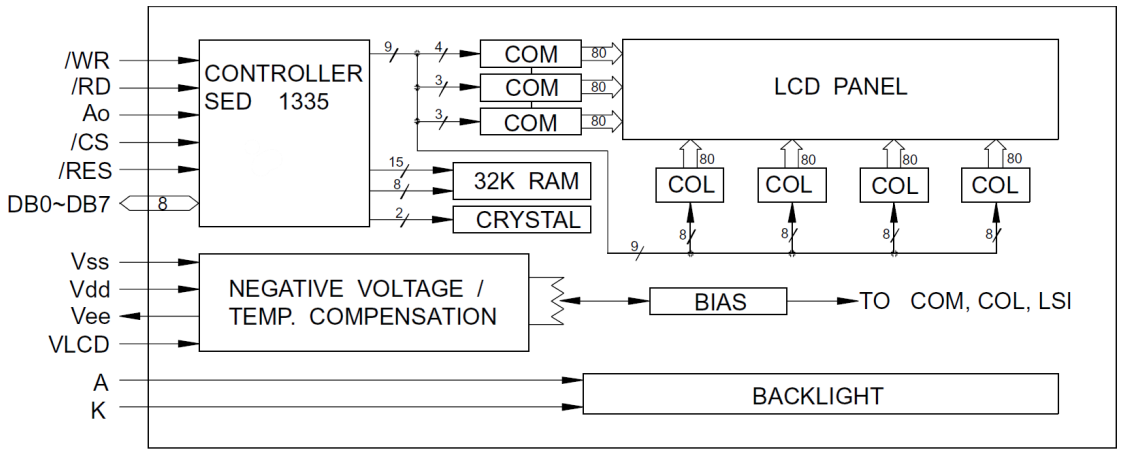
5.1 Displej Batron BT42005

Tento displej je osazen řadičem od firmy SHARP. Umí zobrazovat alfanumerické znaky a různé další symboly. Existuje několik variant, které se liší počtem řádků a stejně tak se mohou lišit počtem zobrazitelných znaků v jedné řádce. Připojuje se zpravidla ke 4bitovým nebo 8bitovým řadičům. Řadič displeje přijímá jednotlivé znaky v ASCII kódu a ukládá je do vlastní RAM paměti. Vzory všech zobrazitelných znaků jsou uloženy v ROM paměti. Tyto vzory se zobrazují do matic o rozměru 5×7

pixelů. Programátor může využít základní funkce podporované řadičem displeje jako jsou např. smazání displeje, pohyb kurzoru, posun znaků doleva i doprava nebo rozsvícení a zhasnutí displeje. Pro více informací viz [8].

5.2 Displej PG320240-D

Oproti předchozímu typu je tento displej poněkud vyspělejší. Podporuje navíc grafický režim s rozlišením 320×240 bodů. O řízení činnosti displeje se stará řadič S1D13305 (starší označení SED1335) k němuž je připojena paměť RAM o kapacitě 32kB, budiče displeje a další obvody. Schéma celého LCD panelu včetně řadiče je na obrázku 5.1.



Obrázek 5.1: Blokové schéma displeje PG320240-D. [9]

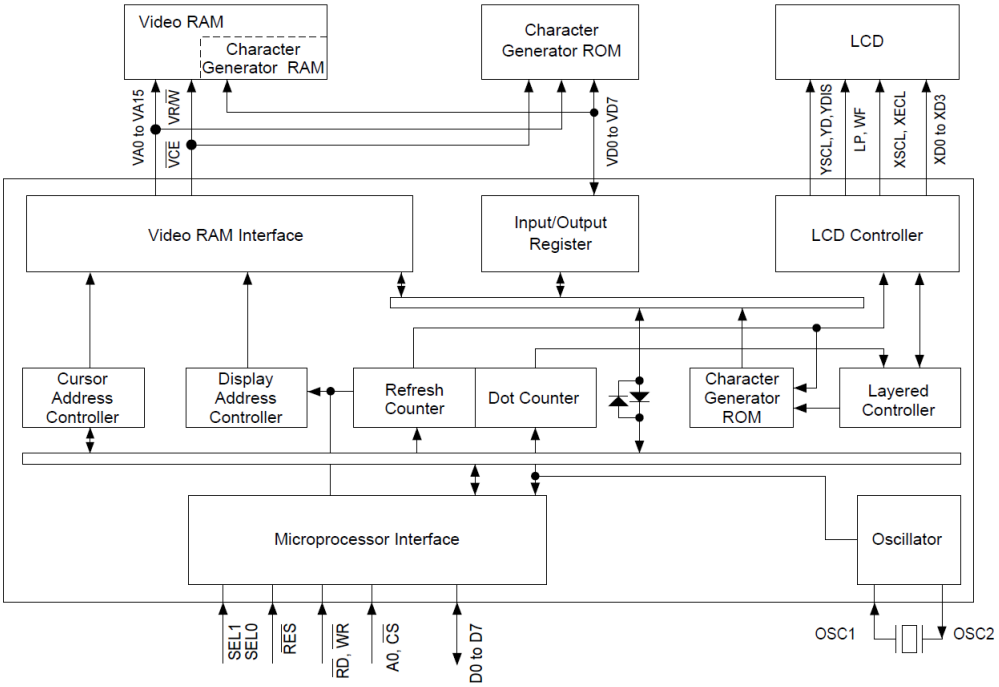
5.2.1 EPSON S1D13305

Řdič EPSON S1D13305 (starší označení SED1330) je navržen k zobrazování textu a grafiky. Veškerá zobrazovaná data ukládá řadič do frame-buffer paměti. Umí pracovat se třemi nezávislými vrstvami přičemž každá má oddělený paměťový prostor. Pokud řadič pracuje s vrstvou v textovém režimu, využívá šablony znaků uložených v paměti ROM. Tyto znaky zobrazuje na displej podle ASCII hodnot znaků uložených ve frame-buffer paměti. V grafickém režimu jsou pixely adresovány bitově. Zvýšení kontrastu zajišťuje podsvícení z LED diod bílé barvy.
Základní charakteristiky řadiče:

- Textový, grafický a kombinovaný režim.
- Tři překrývající se vrstvy v grafickém módu.

- Maximální rozlišení připojeného displeje 640×256 pixelů.
- Externí frame-buffer RAM o velikosti 64 kB.
- Generátor 160 znaků o rozměru 5×7 pixelů.

Blokové schéma řadiče SED1330F je na obr. 5.2.



Obrázek 5.2: Blokové schéma řadiče SED1330F. [10]

5.2.2 Ovládání řadiče

Řadič je připojen k procesoru ATmega32 pomocí pěti řídících signálů a 8bitovou datovou sběrnicí.

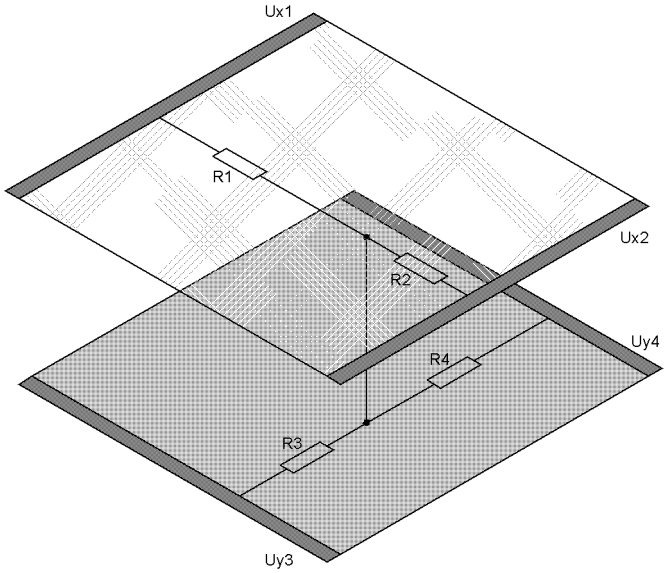
- $D0$ až $D7$ Signály datové sběrnice.
- $\overline{RD}$ Čtení dat z řadiče.
- $\overline{WR}$ Zápis dat do registru řadiče. Data jsou zapisována v okamžiku náběžné hrany signálu.
- $\overline{CS}$ Signál aktivující výběr řadiče (chip select).
- $\overline{RES}$ Hardwarový reset řadiče.

- *A*0 Určuje, zda jsou po datové sběrnici posílány příkazy (A0 je v log. 0) nebo data (A0 je v log. 1).

Aby řadič mohl s displejem provádět nějakou činnost, je nutné nejdřív poslat řadiči příkaz. Řadič na základě signálu $A0 = 0$ rozezná, že se jedná o příkaz. Případné parametry příkazu jsou odesílána jako běžná data. Tedy signál $A0 = 1$. Pokud budou řadiči odeslána samotná data bez předchozího příkazu, nebude řadič vědět, co má s daty provést. Například příkaz zobrazení znaku bude řadič po přijetí příkazu očekávat hodnotu souřadnice X, hodnotu souřadnice Y a ASCII hodnotu znaku.

5.3 Dotyková deska

Na LCD panelu PG320240-D jsou položeny odporové fólie dotykové obrazovky. Pracují na čtyřvodičovém principu, viz obr. 5.3. Obě desky jsou vyrobeny z elektricky vodivého materiálu. Ve chvíli, kdy dojde ke styku obou desek v libovolném místě, chovají se desky jako napěťový dělič. Ke snímání pozice dotyku jsem využil specializovaný obvod ADS7846.



Obrázek 5.3: Princip čtyřvodičového zapojení. [12]

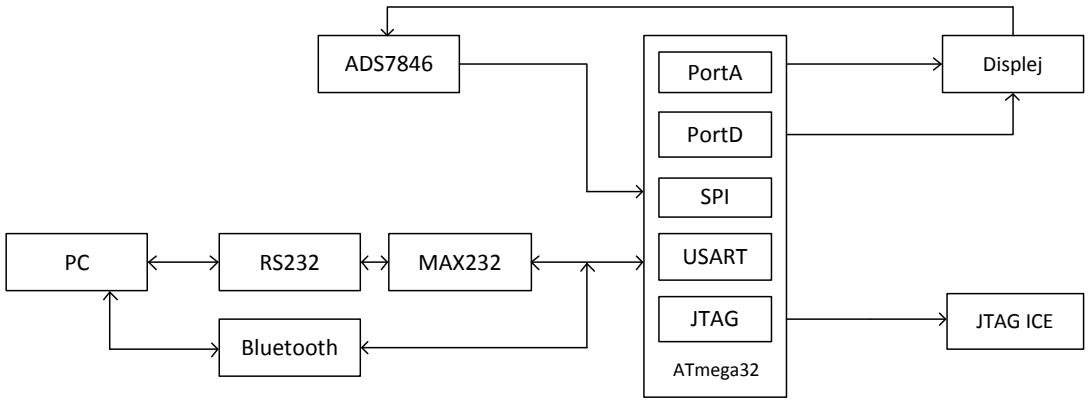
6 Obvod ADS7846

ADS7846 je obvod pracující s dotykovými deskami založenými na čtyřvodičovém principu. Obvod navíc umí měřit sílu přítlaku na dotykové desky. Ve svém principu

funguje jako postupný aproximační registr přijímající data z analogově digitálního převodníku. Komunikace s mikrokontrolérem probíhá po datových vodičích (MISO, MOSI) a dvou signálových (CS, SCK) sběrnice SPI. Obvodu je nutno nejdříve zaslat řídící byte. ADS7846 pak vstoupí do vzorkovacího módu a následně vrátí binární podobu vzorkované analogové hodnoty. Bližší popis možností komunikace s ADS7846 je v manuálu [11].

7 Navržený obvod

Celý obvod je napájen napětím 5V přivedeným na svorkovnici P1. Součástka, která řídí celý obvod je mikrokontrolér ATmega32. Využity jsou všechny porty procesoru. Blokové schéma celého obvodu je na obr. 7.1.



Obrázek 7.1: Blokové schéma navrženého obvodu

Mikrokontrolér ATmega32 má na pinech XTAL1 a XTAL2 připojen krystal generující hodinový signál o frekvenci 7,373 MHz. Z portu A vedou datové vodiče na konektor řadiče displeje H3 společně s řídícími signály vedoucími z portu D. Port B pomocí rozhraní SPI pracuje s obvodem ADS7846 zapojeným do patice U4. Obvod ADS7846 má piny vstupních signálů přivedeny na konektor H6, ke kterému jsou připojeny vodiče z dotykové desky na LCD displeji. Port C má vyvedeny signály ladícího rozhraní JTAG na desetipinový konektor H2 a další dva vodiče pro konektor H1 sběrnice I2C. Port D má integrovanou podporu pro USART. Mezi konektor sériové linky H5 a piny portu D je zapojen obvod MAX232. Ten konvertuje odlišné hodnoty napětí sériové linky do TTL logiky a naopak.
K připojení Bluetooth modulu do patice U3 bylo nutné uvažovat nad smyslem toku elektrických signálů. Pin USART modulu TxD je přímo připojen na pin T1IN obvodu MAX232 a na pin RxD Bluetooth modulu. V tomto případě není nutno nijak

rozdělovat směr toku signálu, protože data vycházející z procesoru mohou být odesílány jak přes sběrnici RS232, tak i rozhraním Bluetooth. Jiná situace nastává v případe pinu RxD na modulu USART u mikrokontroléru. Pokud by došlo k situaci, že z rozhraní Bluetooth a RS-232C budou současně přijímána data, došlo by ke kolizi a následnému rušení signálů. Proto je pin RxD z USARTu vyveden na header J1, který slouží jako přepínač mezi MAX232 a Bluetooth modulem. Použití rozhraní Bluetooth nebo RS-232C je tedy nutné zvolit manuálně propojením dvou pinů zkratovací propojkou. Propojením pinů 1 a 2 budou data přijímána ze sériové linky. Propojením pinů 2 a 3 budou přijímána data z Bluetooth modulu. Činnost bezdrátového modulu je signalizována připojenými diodami.
Reset modulu je možný stisknutím tlačítka SW2. Kompletní návrh obvodu programem Formica je v příloze.

8 Práce s I/O porty

Každý port poskytuje programátorovi tři registry, pomocí nichž lze rozhraní portu ovládat. Význam jednotlivých registrů je popsán v kapitole 3.3. Nejdříve se nastavuje hodnota registru DDRx (Data Direction Register). Poté již má smysl zapisovat do registru PORTx (Port x Data Register) nebo číst hodnoty na pinech z registru PINxn (Port x Input Pins), kde x je písmeno portu a n je číslo pinu. ATmega32 umí adresovat registry přímo bitově. Použitý kompilátor AVR-GCC však bitovou adresaci nepodporuje. Tento drobný nedostatek lze obejít použitím bitových masek. AVR dovoluje v jazyce C jazykovou konstrukci, uvedení pinu do stavu logické 0 a 1. Příklad nastavení portu, kde nejvyšší 4 bity nastavují piny jako výstupní a nejnižší 4 piny jako vstupní. Registr PORTA uchovává data zapisovaná na výstupní piny portu. Nižší 4 piny jsou ve výchozím stavu nastaveny jako vstupní. Nakonec je na port zapsána hodnota 0xB0.

```
void spi_init() {
    DDRA |= (1 << PB7);     // output pin
    DDRA |= (1 << PB6);
    DDRA |= (1 << PB5);
    DDRA |= (1 << PB4);
    PORTA = 0xB0;           // write value
}
```

8.1 Nastavení SPI

Port B je třeba nejdříve nastavit ve smyslu jednotlivých signálů SCK, SS, MISO a MOSI analogicky s předchozím příkladem.
Chování řadiče SPI nastavují tři registry:

SPCR (SPI Control Register): Řídící registr.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

SPSR (SPI Status Register): Stavový registr.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SPIF	WCOL	-	-	-	-	-	SPI2x

SPDR (SPI Data Register): Datový registr.

Bližší popis významu jednotlivých bitů viz [5]. V mé práci je na sběrnici SPI připojen obvod ADS7846 vyžadující na vstupu $DCLK$ frekvenci hodin 2 MHz. Taktovací frekvence signálu SCK je odvozena od taktu mikrokontroléru. V mém případě je mikrokontrolér taktován na 8 MHz. V tab. 2 lze najít vhodné kombinace bitů SPCR a SPSR, které nastaví frekvenci hodin řadiče SPI.

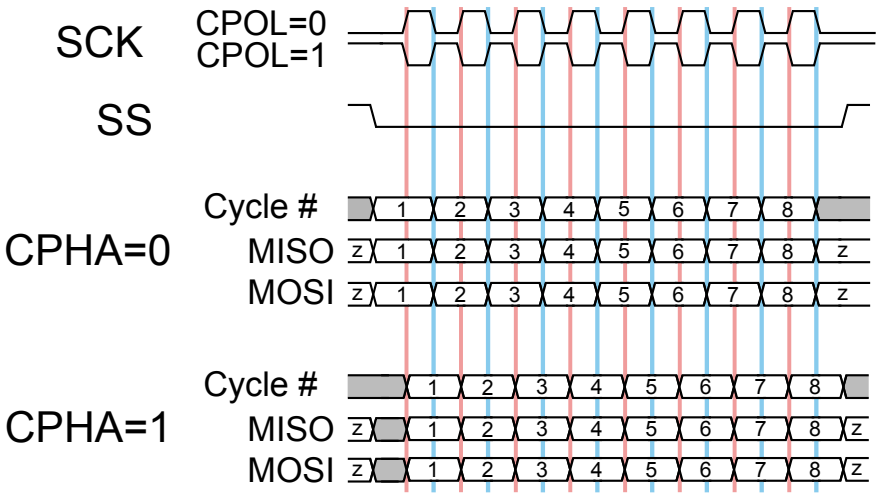
SPI2x	SPR1	SPR0	f_{sck}
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

Tabulka 2: Nastavení hodin SPI.

Ze známého taktu mikrokontroléru $f_{osc} = 8\ MHz$ a požadovaného taktu hodin přivedeného na vstup SCK obvodu ADS7846 $f_{sck} = 2\ MHz$ jsem vypočítal dělící koeficient ze vzorce 1 a nastavil odpovídající bity v řídícím registru SPCR.

$$\frac{f_{osc}}{f_{sck}} = \frac{8\ MHz}{2\ MHz} = 4 \tag{1}$$

Zařízení slave může vyžadovat zvláštní nastavení parametrů komunikace. Prvním parametrem je klidová polarita hodinového signálu a druhým parametrem je fáze hodin určující na jaké hraně hodinového signálu má být vzorkována hodnota na datových vodičích. Možné průběhy signálů jsou znázorněny na obrázku 8.2.



Obrázek 8.2: Časování komunikace. [4]

Oba parametry jsou nastavovány konkrétním bitem v registru SPCR. Binární hodnota bitu CPOL odpovídá klidové hodnotě hodin. Bit CPHA řídí v závislosti na CPOL vzorkování dat na náběžné nebo sestupné hraně hodinového signálu. ADS7846 pracuje správně s klidovou hodnotou hodin v logické nule a vzorkováním dat při vzestupné hraně hodinového signálu.
Nastavení řadiče SPI odpovídající požadavkům obvodu ADS7846:

```
void spi_init() {
    DDRB |= (1 << SPI_MOSI);      // Master Output Slave Input
    DDRB |= (1 << SPI_CS);        // Slave /CS
    DDRB |= (1 << SPI_SCK);       // Slave CLK
    DDRB &= ~(1 << SPI_MISO);     // Master Input Slave Output
    DDRB &= ~(1 << SPI_INT);      // Interrupt is input

    /* SPI Control Register */
    SPCR |= (1 << SPIE);          // SPI interrupt
    SPCR |= (1 << MSTR);          // Master mode
    SPCR |= (1 << SPE);           // SPI enable
    SPCR &= ~(1 << DORD);         // Data ORDer - MSB first
    SPCR &= ~(1 << CPOL);         // Clock POLarity - high when idle
    SPCR &= ~(1 << CPHA);         // Clock PHAse - trailing edge
    SPCR &= ~(1 << SPR1);         // SPI clock Rate - fcpu / 4
```

```
    SPCR &= ~(1 << SPR0);
    SPSR &= ~(1 << SPI2X);

    SPDR = 0x00;                // data register = 0
}
```

8.2 Nastavení USART

Nastavení sériové linky na portu D je analogické nastavení řadiče SPI. Signál RxD je na pinu PD0 nastaven jako vstup a signál TxD na pinu PD1 nastaven jako výstup. Přenosová rychlost USART je odvozena od výchozího nastavení přenosové rychlosti bezdrátového modulu. Ta činí 9600 b/s. Konfigurační registry USART jsou čtyři (UCSRA, UCSRB, UCSRC a UBRR). Registr UBRR (USART Baud Rate Register) nastavuje přenosovou rychlost USART. Jedná se o 16bitový registr, proto musí být hodnota ukládána po 8bitových hodnotách do registrů UBRRH a UBRRL.
Vztah pro výpočet je:

$$UBRR = \frac{f_{osc}}{16 \times baudrate} - 1 \tag{2}$$

Zbývající tři registry obsahují řídící bity nastavující vlastnosti přenostu a příznakové registry zaznamenávající stav USART. Podrobnější popis významu bitů konfiguračních registrů viz [5].

9 Ovládání řadiče displeje

Řadič displeje má přivedeny řídící signály z portu D a datové signály z portu A. Teoretický popis komunikace je popsán v kapitole 5.2.2. Krátká ukázka programu posílající příkaz řadiči:

```
#define LCD_DATA_PORT   PORTA
#define LCD_DATA_DDR    DDRA
#define LCD_DATA_PIN    PINA
#define LCD_RD          PD3
#define LCD_Ao          PD4
#define LCD_WR          PD5
#define LCD_CS          PD6

void cmd_wr(unsigned char command) {
    PORTD |= (1 << LCD_RD);     // nothing to read
    PORTD |= (1 << LCD_Ao);     // sending command => A0 = 1
```

```
    PORTD &= ~(1 << LCD_CS);     // chip select on
    _delay_us(1);
    PORTD &= ~(1 << LCD_WR);     // writing data => write = 0

    LCD_DATA_PORT = command;     // set command values to port pins

    _delay_us(1);
    PORTD |= (1 << LCD_WR);      // command complete => write = 1
    _delay_us(1);
    PORTD |= (1 << LCD_CS);      // chip select off
    _delay_us(1);
}
```

Nejdříve je signál $\overline{RD}$ nastaven do neaktivního stavu log. 1. Poté nastavením signálu $A0$ do log. 1 rozliší řadič displeje, že mu bude poslán řídící příkaz. Signál $\overline{CS}$ aktivuje vstup řadiče a signál $\overline{WR}$ povolí zápis do řadiče. Do datového registru portu je zkopírována hodnota příkazu a odeslána řadiči. Deaktivací signálu $\overline{WR}$ je ukončen zápis a řadič přijatý příkaz zpracuje. Poslání parametrů příkazu řadiči je obdobné. Jediný rozdíl je v práci se signálem $A0$. V případě odesílání příkazu je jeho hodnota v log. 1. V případě odesílání dat log. 0.

10 Ovládání obvodu ADS7846

Obvod je připojen k mikrokontroléru sběrnicí SPI. Ve chvíli, kdy dojde k dotyku na desce, aktivuje obvod ADS7846 signál $\overline{PENIRQ}$. Po rozpoznání příchozího přerušení mikrokontrolér provede jeho obsluhu. Obsluha spočívá v odeslání řídícího bytu obvodu ADS7846. Odpovědí je binární hodnota, která představuje hodnotu napětí na analogovém vstupu. Programovou kalibrací souřadnic lze určit relativní pozici $[0;0]$ a $[X_{max};Y_{max}]$. Řídící příkaz má velikost jednoho bytu. Významové hodnoty jednotlivých bitů jsou na obr. 10.1.

Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
S	A2	A1	A0	MODE	SER/$\overline{\text{DFR}}$	PD1	PD0

Obrázek 10.1: Řídící příkaz ADS7846. [11]

Popis významu jednotlivých bitů:

S Start bit informující o dalších příchozích bitech. Musí být vždy v log. 1.

A0-A2 Adresní bity pro výběr měřeného kanálu vstupního multiplexeru.

Mode Nastavuje přesnost konverze analogového vstupu na 12bit (log. 0) nebo 8bit (log. 1). Použitá přesnost je 8 bitů kvůli následnému přenosu sériovou linkou.

SER/DFR Nastavení referenčního módu. Pro optimální výsledek měření pozice dotyku [X,Y] a sílu přítlaku je doporučován diferenciální režim.

PD1 - PD0 Napájecí režim obvodu. Vybral jsem režim stálého napájení, kdy obvod po dokončení konverze nepřechází do režimu sníženého odběru (PD1 = 1, PD0 = 1).

Adresy kanálů jsou v tab. 3:

A2	A1	A0	Měřená souřadnice
0	0	1	Y
0	1	1	Z1
1	0	0	Z2
1	0	1	X

Tabulka 3: Adresy kanálů pro měření souřadnic.

Například zjištění místa dotyku na souřadnici X má byte odeslaný ADS7846 hodnotu 0xDB:

S	A2	A1	A0	Mode	SER/DFR	PD1	PD0
1	1	0	1	1	0	1	1

Postupně zjištěné hodnoty jsou ukládány do datové struktury uchovávající hodnoty pozice posledního dotyku:

```
struct TOUCH_COORDS {
    unsigned short  xx;
    unsigned short  yy;
    unsigned short  zz;
    unsigned char   touch;
};
```

11 Komunikační protokol

Návrhu obousměrného komunikačního protokolu mezi PC a deskou je přizpůsoben možnostem sériové linky. Obě strany musejí posílat a přijímat data po osmi bitech. Používané datové typy jsou 16bitové. Proto musejí být před přenosem rozděleny na vyšší a nižší byte.
Protokol řadí data v tomto smyslu:

Počet bytů	Kód příkazu	Argument H	Argument L

Druhá strana je nejdříve informována o počtu následujících bytů tak, aby byla schopná správně interpretovat význam příchozích bytů a nezaměnila hodnotu argumentu za kód příkazu v případě, že by se tyto dvě hodnoty sobě rovnaly.

Příklad příkazu vykreslení obdélníku, kde $[X_1, Y_1]$ jsou souřadnice dolního levého rohu a $[X_2, Y_2]$ jsou souřadnice horního pravého rohu:

0x61	X1	Y1	X2	Y2

Odesílaná data sériovou linkou:

7	0x61	X1 H	X1 L	Y1 H	Y1 L	X2 H	X2 L	Y2 H	Y2 L

Ve směru z desky do PC přenáší protokol hodnoty z dotykového displeje. Řazení dat je obdobné pouze s rozdílem významu jednoho bytu:

Počet bytů	Kód situace	Souřadnice X	Souřadnice Y

První byte opět informuje o počtu následujících bytů. Druhý byte rozlišuje mezi třemi situacemi. Dotyk, pohyb po desce a upuštění dotyku. Další byty přenáší argumenty dotyku, kterými jsou souřadnice $[X, Y]$. Hodnoty souřadnic jsou osmibitové. Není tedy potřeba je posílat rozdělené na vyšší a nižší byte.

12 Příkazy přijímané z desky

Vysílání příkazů do PC závisí na povaze dotyku. První zaznamenaný dotyk odesílá příkaz s kódem dotyku a první zaznamenané souřadnice. Po následném vstupu do

smyčky, kontroluje po 100 ms, zda je pořád aktivní signál PENIRQ. Během aktivního PENIRQ odesílá mikrokontrolér do PC data s kódem posunu po desce a příslušnými argumenty. Pokud obvod ADS7846 vrátí byte se všemi bity v log. 1 (1111 1111), znamená to, že na displeji není žádný dotyk. Poté je do PC odeslán příkaz s kódem situace upuštění dotyku s argumentem poslední známé souřadnice dotyku uložené ve struktuře TOUCH_COORDS.

Dotyk

Kód příkazu: `0xA0`

Argument: Souřadnice prvního zaznamenaného dotyku $[X, Y]$.

Posun po desce

Kód příkazu: `0xB0`

Argument: Nová hodnota souřadnic $[X, Y]$, na které byl zaznamenán dotyk.

Upuštění dotyku

Kód příkazu: `0xC0`

Argument: Poslední zaznamenaná hodnota dotyku $[X, Y]$.

Konečné zpracování přijatých dat závisí na potřebách uživatele. Protokol odesílá jen základní data o dotyku.

13 Příkazy odesílané displeji

Řadič displeje je ovládán sadou příkazů. Příkazy se dají rozdělit do tří kategorií. Řídící příkazy, příkazy textového režimu a příkazy grafického režimu. Zpracovávání všech příkazů přijímaných ze sériové linky nebo z Bluetooth pak pracuje se skutečnými příkazy, které ovládají řadič displeje. Podrobný popis všech příkazů řadiče je uveden v manuálu [10]. Následuje stručný popis funkce používaných příkazů.

13.1 Řídící příkazy

Zapnutí displeje

Kód příkazu: `0x01`

Argument: Bez argumentů.

Popis: Aktivuje výstupy řadiče na displej, zobrazí kurzor a případně i data uložená v paměti.

Vypnutí displeje

Kód příkazu: `0x02`

Argument: Bez argumentů.

Popis: Zhasne displej.

Smazání displeje

Kód příkazu: `0x03`

Argument: Bez argumentů.

Popis: Smaže všechna data uložená v textové a v grafické části paměti.

Smazání textových dat

Kód příkazu: `0x04`

Argument: Bez argumentů.

Popis: Smaže pouze textová data z paměti řadiče.

Smazání grafických dat

Kód příkazu: `0x05`

Argument: Bez argumentů.

Popis: Smaže všechna data uložená v textové a v grafické části paměti.

Horizontální posun

Kód příkazu: `0x06`

Argument: `unsigned char data`

Popis: Posouvá celý obraz displeje o jeden pixel doleva. Počet pixelů je volitelný mezi 0 až 7.

13.2 Příkazy textového režimu

Textový režim má funkce mazání vypsaných znaků. Kvůli popisu principu mazání je uvedena pouze funkce smazání jednoho znaku. Ostatní příkazy mazání jsou obdobou příkazu zápisu s tím rozdílem, že původní znak přepisují prázdným znakem.

Zobrazení znaku

Kód příkazu: `0x10`

Argumenty: `int x`, `int y`, `char c`

Popis: Na souřadnicích $[X, Y]$ vypíše znak c. Zobrazitelné jsou pouze znaky, které jsou uloženy v ROM paměti řadiče viz manuál [10].

Smazání znaku

Kód příkazu: `0x11`

Argumenty: `int x`, `int y`

Popis: Smazání znaku ze souřdnic $[X, Y]$ je formálně nahrazeno zápisem znaku mezery (0x20). Jelikož se jedná o tzv. bílý znak, displej se jeví jakoby na daném místě nebyl zapsán žádný znak.

Zobrazení řetězce

Kód příkazu: `0x20`

Argumenty: `int x`, `int y`, `char[] str`

Popis: Vypíše řetězec (pole znaků) `str` přičemž první znak řetězce vypíše na souřadnici $[X, Y]$.

Zobrazení číslice

Kód příkazu: `0x30`

Argumenty: `int n`, `int x`, `int y`

Popis: Provede konverzi číselné hodnoty do znakové podoby a vypíše předanou hodnotu `int n` jako řetězec.

13.3 Příkazy grafického režimu

Základem práce s grafickým režimem je rozsvícení jednoho pixelu. Kreslící algoritmus geometrických obrazců už pak rozsvítí obrazový bod voláním funkce zobrazení pixelu. Smazání objektů vyžaduje shodné argumenty ovšem kód příkazu má na konci 1 místo 0.
Nulové souřadnice $[0, 0]$ jsem ponechal v levém horním rohu displeje.

Zobrazení pixelu

Kód příkazu: `0x40`

Argumenty: `int x`, `int y`

Popis: Rozsvítí obrazový bod na souřadnicích $[X, Y]$.

Zobrazení úsečky

Kód příkazu: `0x50`

Argumenty: `int x1`, `int y1`, `int x2`, `int y2`

Popis: Vykreslí úsečku mezi počátečním bodem $[X_1, Y_1]$ a koncovým bodem $[X_2, Y_2]$.

Zobrazení obdélníku

Kód příkazu: `0x60`

Argument: `int x1`, `int y1`, `int x2`, `int y2`.

Popis: Vykreslí obdélník mezi levým dolním rohem $[X_1, Y_1]$ a pravým horním rohem $[X_2, Y_2]$. Levý horní bod je pak určen jako $[X_1, Y_2]$ a pravý dolní roh pak $[X_2, Y_1]$. Algoritmicky jsou mezi těmito body vykresleny čtyři úsečky.

Zobrazení kružnice

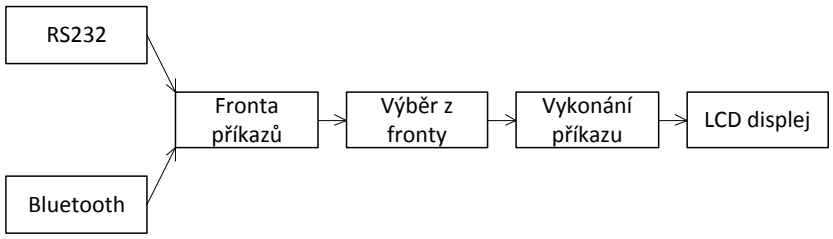
Kód příkazu: `0x70`

Argument: `int x`, `int y`, `int radius`.

Popis: Opíše kružnici o poloměru `radius` se středem v $[X, Y]$.

14 Vykonání příchozích příkazů displeje

Příkazy pro řadič displeje jsou přijímány z rozhraní RS232 a Bluetooth. Příchozí příkazy jsou zařazovány do vstupní fronty příkazů. Ze vstupní fronty jsou vybírány postupně jednotlivé byty a jejich hodnota je porovnána s hodnotou definovaných příkazů. Pokud je ve frontě známý příkaz, pak vyjme z fronty potřebný počet argumentů. Rozeznaný příkaz s argumenty je pak předán funkci, která provádí skutečné operace s řadičem displeje. Blokové schéma zpracování příkazů je na obr. 14.1.



Obrázek 14.1: Zpracování příkazů LCD displeje

15 Používané programy

AVR Studio Vývojové prostředí od firmy Atmel. Podporuje několik programovacích a ladící zařízení včetně mnou používaného rozhraní JTAG.

Win AVR Softwarová sada přidávající podporu AVR-GCC kompilátoru pro C a C++ do AVR Studia. Obsahuje potřebné knihovny pro práci s mikrokontroléry AVR od firmy Atmel.

PonyProg2000 Sériový programátor, který umí pracovat s mikrokontroléry AVR. Užitečnou vlastností je podpora programátoru AVR ISP. Pokud přestane pracovat programovací obvod JTAG ICE, pak je možné znovu obnovit jeho funkčnost pomocí PonyProg nahráním bootloaderu přiloženého na CD. Postup zotavení programátoru viz [13].

16 Programová dokumentace

Základní funkce jednotlivých částí programu:

`main.c`

Po spuštění programu vykoná hlavní funkce inicializaci displeje, sériové linky a fronty. V těle hlavní funkce běží smyčka, která vybírá data z fronty a podle kódu příkazů volá příslušné funkce obsluhující práci s displejem.

`commands.h`

Obsahuje definice příkazů. Každý příkaz má svou číselnou hodnotu.

`queue.h`

Definuje strukturu používaných příkazů. Definuje strukturu používané fronty a její vlastnosti. Obsahuje hlavičky funkcí fronty.

`queue.c`

Inicializace frontu, provádí vkládání a vybírání příkazů z fronty a funkce pro kontrolu její obsazenosti.

`usart.h`

Definuje parametry sériové linky (rychlost přenosu, hodnotu registru UBRR, režim komunikace).

`usart.c`

Inicializuje obsluhu sériové linky. Provádí příjem a odesílání dat sériovou linkou. Obsluhuje přerušení.

`lcd.h`

Pro lepší orientaci v příkazech pro řadič displeje obsahuje definice příkazů jako symbolické řetězce. Hodnoty těchto příkazů jsou uvedeny v manuálu řadiče. Obdobně definuje pojmenování řídících signálů řadiče displeje.

`lcd.c`

Provádí inicializaci řadiče displeje. Nastavuje počáteční adresy jednotlivých adresních prostorů pro textová data a pro data grafiky. Odesílá příkazy k displeji. Pokud příkaz má parametry, pak odesílá i tyto parametry jako data. Dále nastavuje základní parametry displeje jako je zapnutí a vypnutí displeje, smazání dat v paměti, zobrazení kurzoru nebo horizontální posunutí textové vrstvy.

`lcd_graphic.h`

Importuje pouze knihovny `math.h` potřebnou pro algoritmy vykreslující geometrické obrazce a `stdio.h` potřebnou pro konverzi číselných hodnot do řetězcové podoby.

`lcd_graphic.c`

Obstarává zobrazení textových i grafických prvků. Textová data odesílá jako znaky. Základem manipulace s grafickými daty je vykreslení jediného pixelu. Funkci vykreslení jednoho pixelu využívají další funkce pro kreslení základních geometrických obrazců.

`spi.h`

Definuje používané piny portu, nastavuje velikost bufferu sběrnice SPI a připojuje potřebné hlavičkové soubory.

`spi.c`

Inicializuje sběrnici SPI (nastaví směr pinů MISO, MOSI, SCK a SS), přijímá a odesílá data po sběrnici a obsluhuje přerušení.

`ads.h`

Definuje strukturu uchovávající hodnoty souřadnic dotyku a přítlaku na dotykovou desku. Definuje kódy situací pro zpětný přenos dat do PC.

`ads.c`

Odesílá řídící byty do obvodu ADS7846 a současně přijímá surová data. Přijatá data uloží do struktury definované v hlavičkovém souboru. Provádí kalibraci signálů z dotykové desky, tak aby byla data srozumitelně reprezentovatelná. Posílá data o dotyku sériovou linkou uživateli.

17 Závěr

Obecným úkolem práce bylo prostudování vlastností dostupných LCD displejů a mikrokontrolérů tak aby bylo možné vytvořit základní komunikační rozhraní mezi PC a LCD displejem. Celkový návrh zapojení modulu byl postaven na výběru mikrokontroléru. Mou volbu použitého mikrokontroléru ovlivnilo několik faktorů. Použití obvodu ADS7846 vyžadovalo podporu sběrnice SPI, uvažovaný bezdrátový modul vyžadoval připojení k UART, LCD displej ke své činnosti potřebuje 8bitovou datovou sběrnici a 5bitovou řídící sběrnici a také přítomnost programovacího a ladícího rozhraní. Obvodem ADS7846 se mi podařilo získávat hodnoty místa dotyku i velikost přítlaku. Velmi cennou zkušeností mi byla práce s programem pro návrh plošných spojů Formica, ve kterém jsem celý obvod navrhl.

Dále jsem poznal vývojové prostředí pro mikrokontroléry postavené na architektuře AVR a s tím související princip programování a ladění pomocí rozhraní JTAG. Navržený protokol umožňuje přijímat a odesílat data skrze bezdrátové rozhraní. Uživatel má tedy možnost posílat příkazy, které budou pracovat s displejem a zároveň přijímat zpracovaná data z dotykové desky.

Možné rozšíření navrženého obvodu je přidání kompaktního modulu ZigBee, případně vytvoření návrhu s nahrazeným rozhraním Bluetooth.

Zobrazování dat na displeji by mohlo být rozšířeno ve smyslu dotykových tlačítek známých např. z obslužných terminálů v MHD. Toto rozšíření spočívá ve vytvoření programu odesílajícího takové obrazce, ze kterých je patrné ohraničení tlačítka. Při stisku dotykové desky dojde k odeslání souřadnic dotyku, které pak uživatelský program přijme a vyhodnotí jako dotyk na tlačítku.

Seznam zkratek

EDR Enhanced Data Rate (Zvýšená rychlost přenosu dat)

I2C Inter-Integrated Circuit (Nízkorychlostní počítačová sběrnice)

ICE In Circuit Emulator (Hardwarový emulátor pro ladění programů)

JTAG Joint Test Action Group (Standard pro programování integrovaných obvodů a jejich testování)

LCD Liquid crystal display (Displej z tekutých krystalů)

MISO Master Input Slave Output (SPI Vstup master výstup slave)

MOSI Master Output Slave Input (SPI Výstup master vstup slave)

RAM Random Access Memory (Paměť s libovolným přístupem)

ROM Read Only Memory (Paměť pouze pro čtení)

SCK SPI Clock (Frekvence hodin sběrnice SPI)

SMD Surface Mount Device (Pájení elektronických součástek přímo na povrch plošného spoje)

SPI Serial Peripheral Interface (Sériové periferní rozhraní)

SRAM Static Random Access Memory (Statická paměť s libovolným přístupem)

TTL Transistor-transistor Logic (Transistorově-transistorová logika)

TWI Two-Wire Interface (Dvouvodičové rozhraní. kompatibilní se sběrnicí I2C)

USART Universal Synchronous/Asynchronous Receiver and Transmitter (Univerzální Synchronní/Asynchronní vysílač a přijímač)

USB Universal Serial Bus (Univerzální sériová sběrnice)

Seznam obrázků

Seznam tabulek

Reference

[1] GOOK, M.: *Hardwarová rozhraní: průvodce programátora*. Brno: Computer Press, první vydání, 2006, ISBN 80-251-1019-2.

[2] MESSMER, H.-P.; DEMBOWSKI, K.: *Velká kniha hardware: [architektura, funkce, programování]*. Brno: CP Books, první vydání, 2005, ISBN 80-251-0416-8.

[3] Best-Microcontroller-Projects.com: How RS232 works [online]. 2005, informační zdroj dostupný z URL: <http://www.best-microcontroller-projects.com/how-rs232-works.html>, <20.2.2012>.

[4] Serial Peripheral Interface Bus: Wikipedia EN [online]. Informační zdroj dostupný z URL: <http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus>, <22.2.2012>.

[5] Atmel Corporation: *ATmega32: 8-bit Microcontroller with 32KBytes In-System Programmable Flash*. 2009, dostupné na přiloženém CD a z: <http://www.atmel.com/Images/doc2503.pdf>, <21.3.2012>.

[6] ZigBee - novinka na poli bezdrátové komunikace. Informační zdroj dostupný z URL: <http://www.hw.cz/Rozhrani/ART1299-ZigBee---novinka-na-poli-bezdratove-komunikace.html>, <25.4.2012>.

[7] connectBlue: *OEM Serial Port Adapter^{TM}cB-0901 Electrical & Mechanical Datasheet*. September 2011, dostupné na přiloženém CD a z: <http://www.connectblue.com/fileadmin/Connectblue/Web2006/Products/Bluetooth/Gen3/M0M1M2/OEM/Common/Documents/E_M_Datasheet_OEMSPA_311_331.pdf>, <3.4.2012>.

[8] SHARP Microelectronics: *Dot-Matrix LCD Units (with built-in controllers) - Display unit user's manual*. 1999, dostupné na přiloženém CD a z: <http://home.zcu.cz/~dudacek/manuals/lm.pdf>, <2.5.2012>.

[9] Powertip: *Powertip PG 320240-D Outline Dimension & Block Diagram*. Dostupné na přiloženém CD a z: <http://home.zcu.cz/~dudacek/manuals/pg320240.pdf>, <22.2.2012>.

[10] S-MOS Systems, Inc, 2460 North First Street, San Jose, CA 95131: *SED1330F/1335F/1336F LCD Controller ICs Technical Manual*. September 1995, dostupné na přiloženém CD a z: <http://home.zcu.cz/~dudacek/manuals/sed_1330.pdf>, <22.2.2012>.

[11] Texas Instruments Inc.: *ADS7846 TOUCH SCREEN CONTROLLER*. September 2003, dostupné na přiloženém CD a z: <http://home.zcu.cz/~dudacek/manuals/ads7846.pdf>, <18.4.2012>.

[12] Dotyková obrazovka: Wikipedia SK [online]. Informační zdroj dostupný z URL: <http://sk.wikipedia.org/wiki/Dotyková_obrazovka>, <22.4.2012>.

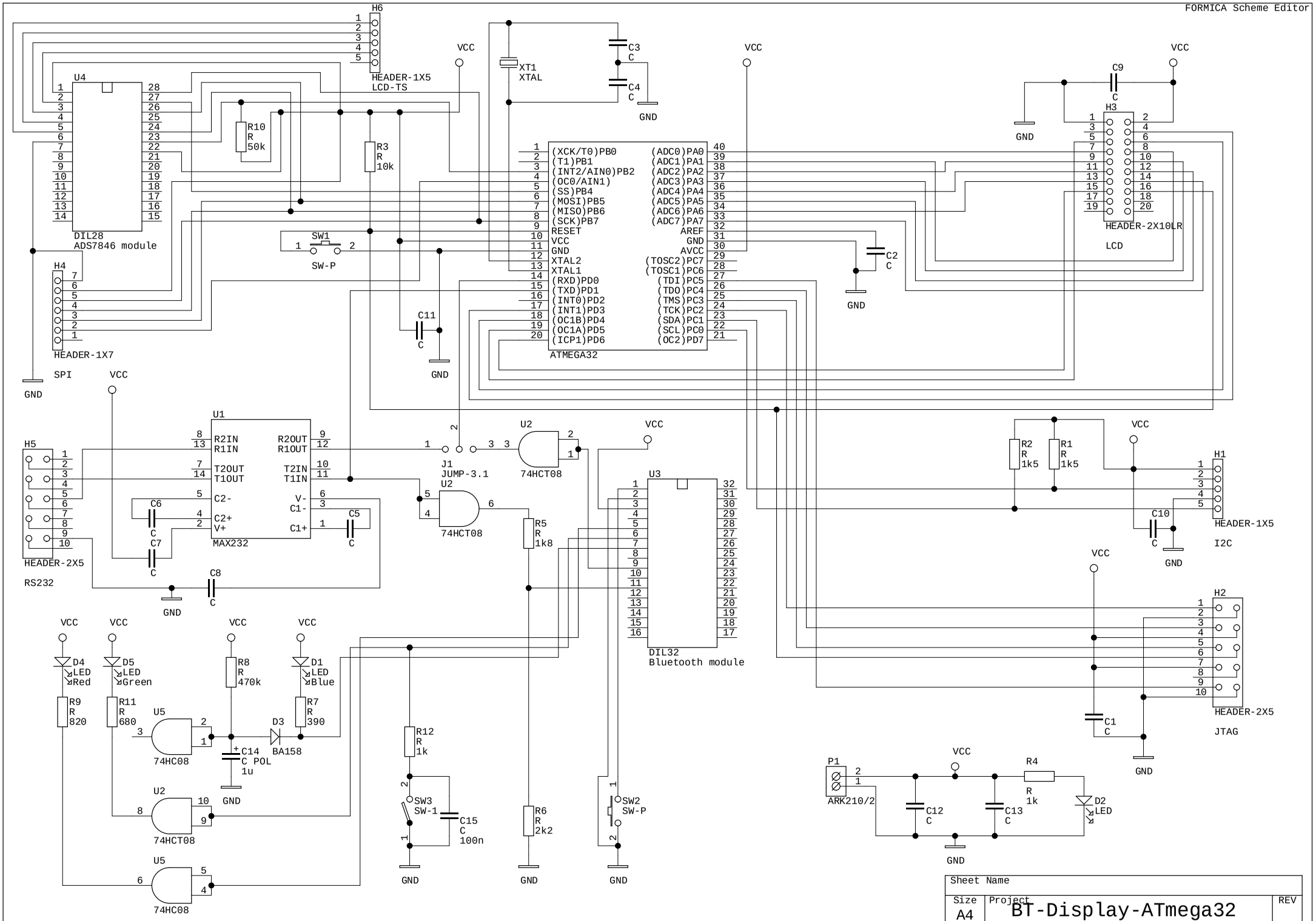
[13] DRÁBEK, T.: *Ladění programů v embedded aplikacích*. Diplomová práce, Západočeská univerzita v Plzni, Plzeň, 2008.

[14] KRAUS, F.: *Rozhraní I2C pro LCD displej*. Bakalářská práce, Západočeská univerzita v Plzni, Plzeň, 2011.

[15] MATOUŠEK, D.: *Práce s mikrokontroléry ATMEL AVR: [měření, řízení a regulace pomocí několika jednoduchých přípravků]*. Praha: BEN - techická literatura, první vydání, 2006, ISBN 80-7300-174-8.

Přílohy

Příloha 1: Schéma zapojení navrženého obvodu



Sheet Name		
Size A4	Project BT-Display-ATmega32	REV
Date: MAY-7-2012	Sheet 1 / 1	