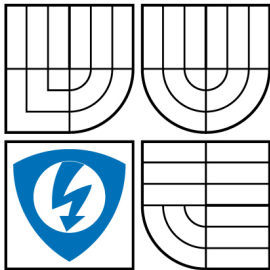


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ČTEČKA ČÁROVÉHO KÓDU A RFID ČIPŮ S AVR MIKROKONTROLEREM

AVR MICROCONTROLLER-BASED READER OF BAR CODES AND RFID CHIPS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

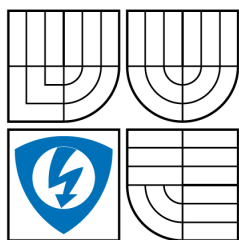
MAREK BARKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL ŠILHAVÝ, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Marek Barka

ID: 83946

Ročník: 3

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Čtečka čárového kódu a RFID čipů s AVR mikrokontrolerem

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte modul čteček čárového kódu a RFID čipů k osobnímu počítači, který bude připojen prostřednictvím PS/2 rozhraní klávesnice. Modul bude obsahovat AVR mikrokontroler, k němuž bude připojen modul čtečky čárového kódu Metrologic IS4125B41 a čtečka RFID čipů realizovaná s pomocí obvodu TI TRF7960.

DOPORUČENÁ LITERATURA:

- [1] Váňa, V.. Mikrokontrolery ATMEL AVR - popis procesorů a instrukčního souboru. BEN Praha 2003. ISBN: 80-7300-083-0.
- [2] Váňa, V.. Mikrokontrolery ATMEL AVR - programování v jazyce C. BEN Praha 2003. ISBN: 80-7300-102-0.
- [3] Šandera, J.. Návrh plošných spojů pro povrchovou montáž - SMT a SMD. BEN Praha 2006. ISBN: 80-7300-181-0.

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Pavel Šilhavý, Ph.D.

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Abstrakt

Práce pojednává o návrhu řešení čtečky čárových kódů a Rfid čipů, jejichž nasnímaná data jsou zpracována mikroprocesorem a odeslána na vstup klávesnice do osobního počítače. Zařazuje čárový kód a Rfid čip do tematické oblasti automatické identifikace. Součástí a cílem práce je realizace celého hardwarového řešení a přenesení aktuálně nasnímaných dat do aplikace v počítači. Zahrnuje v sobě metody komunikace s danými integrovanými obvody, včetně podrobného popisu a testování jejich připojení k použitým rozhraním.

Abstract

This thesis dealt with concept resolution of bar-code scanning and Rfid chip, which their scan data are processing with microprocessor and they are transmitted to input computer keyboard. Filing the bar-code and Rfid chip to the topical area automatic identification. A finish of the work is a realisation hardware and transmission recency data to the computer space. It work includes techniques, which it describes communication with others integrated circuit together detailed definition and test used interfaces.

Klíčová slova

čtečka čárových kódů, Rfid čip, vstup klávesnice, čárový kód, mikrokontrolér, rozhraní.

Keywords

bar code scanner, Rfid chip, input keyboard, bar code, microcontroller, interface.

Bibliografická citace mé práce:

BARKA, M. *Čtečka čárového kódu a RFID čipů s AVR mikrokontrolerem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 47 s. Vedoucí bakalářské práce Ing. Pavel Šilhavý, Ph.D.

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma Čtečka čárového kódu, RFID čipů s AVR mikrokontrolérem jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

podpis autora

1 Obsah

1	Obsah	6
2	Seznam obrázků a tabulek	7
3	Úvod.....	8
4	Tematické zařazení	9
4.1	Čárový kód.....	9
4.1.1	1D kódy - lineární	9
4.1.2	2D kódy – maticové.....	11
4.2	RFID kód (Radio Frequency Identification).....	12
4.2.1	Aktivní čipy	13
4.2.2	Pasivní čipy.....	13
4.3	Snímač	13
4.3.1	Laserový.....	14
4.3.2	CCD kamera	14
4.3.3	RFID snímač (Radio Frequency Identification)	15
5	Navržené řešení.....	16
5.1	Blokové schéma.....	16
5.2	Rozhraní.....	17
5.2.1	SPI.....	17
5.2.2	PS-2.....	19
5.3	Bloky.....	22
5.3.1	Snímač čárového kódu IS4125B-41	22
5.3.2	Převodník SPI/RS232 MAX3110.....	23
5.3.3	RFID snímač TRF 7960.....	24
5.3.4	Mikrokontrolér ATmega8.....	25
5.4	Realizace	26
5.4.1	Software	26
5.4.2	Hardware.....	30
6	Závěr	32
7	Abecední přehled použitých zkratk a symbolů	35
8	Přílohy.....	36

2 Seznam obrázků a tabulek

Obr. 4.1 Čárový kód.	9
Obr. 4.2 Nejpoužívanější lineární čárové kódy.	10
Obr. 4.3 Nejpoužívanější EAN lineární čárové kódy.	11
Obr. 4.4 Nejpoužívanější maticové čárové kódy.	12
Obr. 4.5 Názorné zobrazení RFID kódu (tag, transpondér).....	12
Tab. 4.6 Tabulka frekvenčních pásem a dosah tagů.	13
Obr. 4.7 Zobrazení snímání čárového kódu pomocí laserového snímače	14
Obr. 4.8 Znázornění RFID komunikace.	15
Obr. 5.1 Názorné zapojení bloků čtečky kódů.....	16
Obr. 5.2 Blokové schéma zařízení.	16
Obr. 5.3 Časový průběh na SPI kanálu	17
Obr. 5.4 Časový průběh PS-2 komunikace slave > master.....	19
Obr. 5.5 Časový průběh komunikačního rozhraní USART.....	21
Obr. 5.6 Čelní pohled na snímač čárového kódu IS4125B-41	22
Obr. 5.7 Typické zapojení obvodu MAX3110.	23
Obr. 5.8 Blokové zapojení modulu snímače RFID kódů.....	24
Obr. 5.9 Vývojový diagram čtení čárového kódu.....	27
Obr. 5.11 Vývojový diagram odeslání dat na PS-2.	29

3 Úvod

Čtečky čárových kódů mají za úkol snímat a zpracovat informace převedeny do čar a mezer. Naopak čtečky Rfid čipů pracují s daty uloženými v paměti mikročipu tzv. tagu.

Nejčastěji se s nimi setkáme v obchodních řetězcích, skladech, dopravní a balíkové službě, nebo také ve výrobě. Všude tam, kde najde uplatnění automatická identifikace. Zabezpečí se tím podstatné zrychlení skladových převodů a jednoznačné určení zboží. Uložená informace může reprezentovat například číslo zboží, sériové číslo, šarži, identifikační číslo. Vrcholem zpracování snímaných dat je možnost zjistit přesnou pozici zboží, nebo kdo a kdy s danou položkou manipuloval.

Cílem projektu je navrhnout a realizovat modul pro připojení čtečky čárového kódu a čtečky RFID čipů. Modul bude obsahovat AVR mikrokontrolér, k němuž bude připojen modul čtečky čárového kódu Metrologic IS4125B41 a čtečka RFID s pomocí obvodu TI TRF7960 včetně vestavěné antény pro bezkontaktní přenos informací. Data snímaná pomocí čteček budou zpracována mikrokontrolérem a předána na rozhraní PS-2, které bude možno připojit k PC prostřednictvím konektoru klávesnice.

4 Tematické zařazení

4.1 Čárový kód

Čárový kód je zobrazením numerických, nebo alfanumerických informací, které je možné číst pomocí snímačů emitujících světelný lúč. Je to sestava tmavých čar a světlých mezer. Každá číslice, písmeno, nebo znak má svoje jednoznačné uspořádání, záleží na typu použitého kódu. Může v něm být zakódována jakákoli informace. Např.: číslo skladové karty, cena, hmotnost, jméno osoby...

Spadá do oblasti automatické identifikace, čili čtení dat bez použití klávesnice (ručního zadávání). Výsledkem je pak vysoká spolehlivost a rychlost zpracování [6].

Nejpodstatnějšími parametry čárového kódu jsou hustota a kontrast. Hustota udává počet čar a mezer na jednotku délky. Množství zakódované informace na jednotku délky určuje hustota a druh kódu. Pro různě-barevné čárové kódy platí zásada, že podklad se vybírá ze světlých barev blízké červeného konce a pro čáry barvy tmavé blízké modrého konce světelného spektra [10].

4.1.1 1D kódy - lineární

Lineární kódy se používají pro kódování menšího počtu znaků (nepřesáhne dvě desítky znaků). Hustota zakódovaných informací se dá zvýšit volbou kódu z vyšší hustotou. To však se sebou přináší zranitelnost a citlivost na nečistoty, nebo také dražší snímací techniku.

Na začátku a na konci čárového kódu se vyskytují znaky začátku a konce. Aby bylo možné vyhodnotit, zda byl čárový kód sejmut celý viz obrázek 4.1.



Obr. 4.1 Čárový kód.

Popisek pod čárovým kódem tzv. informační obsah kódu má jen informativní charakter, nemusí být uveden. Nejpoužívanější lineární kódy z této oblasti jsou Code 39 a Code 128:

Code 39

Použití: průmysl

Sada znaků: space \$%*+-. /
0123456789ABCDEFGHIJKL
MNOPQRSTUVWXYZ



12345ABCDE

a)

Code 128

Použití: v logistice

Sada znaků: tři různé
alfanumerické sady (každá
128 ASCII znaků)



1234567890

b)

Obr. 4.2 Nejpoužívanější lineární čárové kódy.

a) Code 39, b) Code 128

Zvláštní postavení mezi čárovými kódy mají kódy **EAN13** a **EAN8**, kterými musí být označeno zboží v obchodních řetězcích. Jejich použití se řídí přísnými pravidly, jejichž koordinací se zabývá národní organizace EAN Czech, která přiděluje kódy výrobcům na základě žádosti [8].

Jsou na nich patrné odlišnosti oproti předešlým (viz obrázek 4.3) a řídí se těmito pravidly:

- znak začátku a konce je řádně odlišen.
- přidán střední kontrolní znak.
- nesmí být zakódován jiný znak než číslice.
- poslední číslice je tzv. kontrolní číslo.
- mají přesně určen počet zakódovaných číslic.

Kontrolní číslo, třinácté (nebo osmé), které je na základě algoritmu vypočteno z předchozích číslic kódu a doplněno do čárového kódu. Eliminují se tím chyby při čtení [6].

S bloku číslic je pak možné vyčíst, v které zemi bylo zboží zabaleno (859) a v které firmě tak bylo učiněno (632), viz obrázek 4.3 a).

EAN 8

Použití: označování zboží
Sada znaků: 0123456789



8596 3245

a)

EAN 13

Použití: označování zboží
Sada znaků: 0123456789



9 770862 937004

b)

Obr. 4.3 Nejpoužívanější EAN lineární čárové kódy.

a) Ean 8, b) Ean 13

V tomto druhu lineárních kódů se setkáme z pojmem tzv. kontrolní číslo, třinácté (nebo osmé), které je na základě algoritmu vypočteno z předchozích číslic kódu a doplněno do čárového kódu. Eliminují se tím chyby při čtení [6].

4.1.2 2D kódy – maticové

Dvourozměrné čárové kódy patří do nové generace čárových kódů, tzv. maticové kódy. Kódují se jimi znakově objemné informace. Vznik prvních kódů se datuje od roku 1998.

Poskytují možnost bezpečného kódování neobyčejně velkého množství znaků s minimální nárokem na velikost plochy. Jednotlivé bajty jsou kódovány po blocích (modulech) do dvou rozměrného obrazce. U těchto kódů se povětšinou na okrajích vyskytují rovné čáry, které napomáhají snímači určit, zda sejmul celý kód. Praktické využití je v oblasti, kde zpracovaná data obsahují několik položek. Může obsahovat instrukce, jak zacházet s objektem, údaje o výrobcích v balení, diagnózu pacienta a pod. Nejvíce používanými kódy je typ kódu Datamatrix a PDF417 [6].

Datamatrix

Použití: označování malých
předmětů

Kapacita: 2335 znaků
nebo 3113 čísel

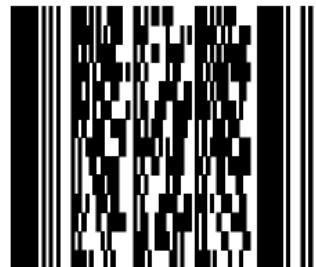


a)

PDF 417

Použití: všeobecné

Kapacita: 1850 znaků
nebo 2710 čísel



b)

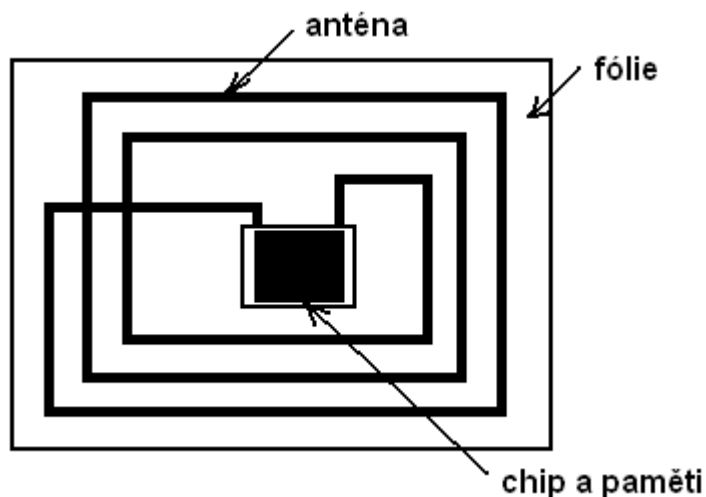
Obr. 4.4 Nejpoužívanější maticové čárové kódy.

a) Datamatrix, b) Pdf 417

4.2 RFID kód (Radio Frequency Identification)

Rádio-frekvenční systém identifikace je moderní technologie identifikace objektů pomocí radiových vln.

Informace jsou elektronicky uloženy v malých paměťových čípech se spirálovou anténou tzv. tagů, ze kterých je lze následně načítat pomocí radiových vln. Podobně jako čárové kódy se tag připevňuje na sledované objekty.



Obr. 4.5 Názorné zobrazení RFID kódu (tag, transpondér).

Vyhotovení tagu, může mít podobu folie, plastové podložky jako vidíme na obrázku 4.5. Nebo podobu etikety, na kterou je možno potisknout grafiku, nebo čárový kód (RFID smart label). V poslední době se objevují náramky (RFID wristband) a plastové karty (RFID card).

Dělení podle zdroje:

4.2.1 Aktivní čipy

Vysílají samy své údaje do okolí (TTF tag talks first), to umožňuje vlastní miniaturní baterie umístěna v čipu, která vydrží cca 1-5 let. Tyto čipy však mají kvůli baterii menší odolnost na teplotu a je nutné baterii měnit. Používají se pro sledování osob, sledování zvířat, vozů a tam kde lze čip opětovně použít). Vzdálenost čtení je až 100 m, velikost paměti na čipu může dosahovat až 100 Kb.

4.2.2 Pasivní čipy

Pomocí vln vyzářených z čtecího zařízení dojde k nabití čipu a následně se informace uložená v čipu bezdrátově přeneše zpět do čtecího zařízení (RTF reader talk first). Jsou cenově výhodnější. Akční vzdálenost čtení je od 0,5m do 10m, velikost paměti 64 - 256 bits. V této skupině se setkáváme ještě s pojmem semiaktivní čipy, které jsou vybaveny baterií, ale používají ji jen k zvětšení komunikační vzdálenosti od čtecího zařízení.

Frekvence	Dosah	Popis
125 – 134 kHz (LF) nízká frekvence	max. 0,5 m	možnost snímání v blízkosti kovu a přes vodu; nízká rychlost snímání
13,56 MHz (HF) vysoká frekvence	max. 1 m	obtížné snímání přes vodu; rychlost snímání / zápis cca. 10x rychlejší než LF (20 kB/s)
865 – 869 MHz (UHF) velmi vysoká frekvence	max. 3 m	nelze snímat přes kapalinu a obtížně přes kov
2,45 GHz ; 5,8 GHz (MW) mikrovlnná frekvence	max. 10 m	možnost čtení při velmi vysokých rychlostech; vysoká cena čipů

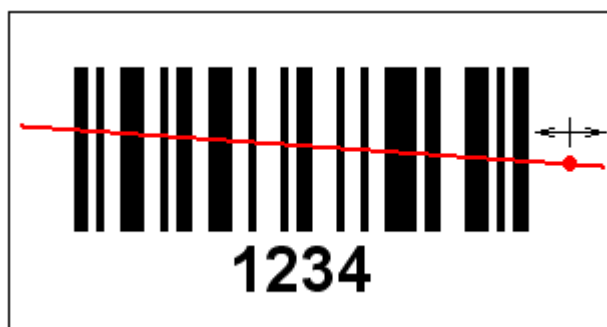
Tab. 4.6 Tabulka frekvenčních pásem a dosah tagů.

4.3 Snímač

Je to v podstatě převodník, který vhodnou metodou načte kód a převede jej na elektrickou informaci.

4.3.1 Laserový

Laserové snímače jsou nejstarší. Lze s nimi snímat lineární čárové kódy. Nejčastěji se používá červený, dobře viditelný, paprsek o vlnové délce zpravidla 650 nm. Tento paprsek je vhodně rozkmitán. Proto se u snímačů setkáváme s výrazem snímací rychlost. Je to počet kmitání za jednotku času. Světlo je pohlcováno tmavými proužky a světlými proužky je odraženo. Tím dostáváme elektrický signál, který se pak dekóduje.



Obr. 4.7 Zobrazení snímání čárového kódu pomocí laserového snímače

Běžně je možno snímat čárový kód ze vzdálenosti 10 - 50 cm. Použitím vyšší třídy laseru (třída II) je možné tuto vzdálenost zvětšit až na 13 metrů [9]. Nesmíme zapomenout, že mezi snímačem a čárovým kódem musí být přímá viditelnost.

Pro zvláštní aplikace se používají snímače pracující v neviditelném infračerveném spektru. Pro čtení v infračerveném pásmu jsou kódy tištěné na termocitlivém papíru [3].

4.3.2 CCD kamera

Tyto snímače byly vyvinuty především pro čtení maticového 2D kódu. Snímačem je tedy CCD kamera, která v dnešní době pořídí snímek v rozlišení 1280x1024 pixel. Obraz je na základě algoritmu zpracován a vhodnou analýzou se dekóduje zachycený čárový kód. Díky tomu je možno těmito snímači snímat i lineární čárové kódy.

Také se tady setkáme s pojmem snímací rychlost, která odpovídá počtu snímků za jednotku času.

Při samotném snímání někteří výrobci snímáný objekt osvětlí aby se eliminovala chyba čtení při nevhodném osvětlení. Také zde je potřeba dodržet přímou viditelnost mezi snímaným kódem a snímačem.

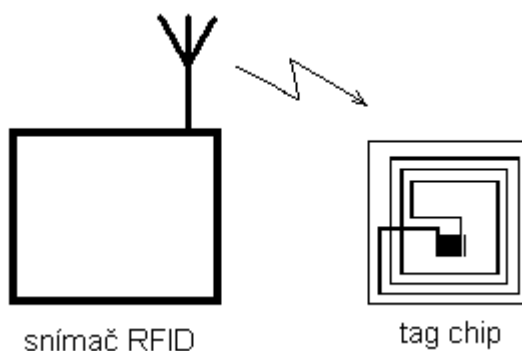
Čárový kód se těmito snímači snímá v rozmezí 10 – 30 cm. Je to zřejmé, protože z větší vzdálenosti není možné s tímto rozlišením korektně rozlišit tmavé a světlé oblasti čárového kódu [3].

4.3.3 RFID snímač (Radio Frequency Identification)

Bezdrátová komunikace, čili komunikace pomocí elektromagnetických vln, mezi terminálem a kódem, je v této době nejmodernější technologií. Tento systém lze úspěšně nasadit v mnoha odvětvích a oblastech, kde je kladen důraz na co nejrychlejší a přesné zpracování informací a okamžitý přenos těchto načtených dat k následnému zpracování.

Mezi snímačem a kódem nemusí být přímá viditelnost. Problémy se projeví jen při snímání přes kapalinu, nebo kov.

Ze snímače je vysílán RF signál směrem k rádiovému transponderu (tj. nosič dat, čip s anténou - tag), který po zachycení signálu vyšle zpět specifický kód, EPC kód (electronic product code), který je jedinečný pro každý tag. Je to vlastně sériové číslo tagu. Pak je možno načíst informaci z příslušného tagu, nebo zapsat data, pokud to vyhotovení dovoluje.

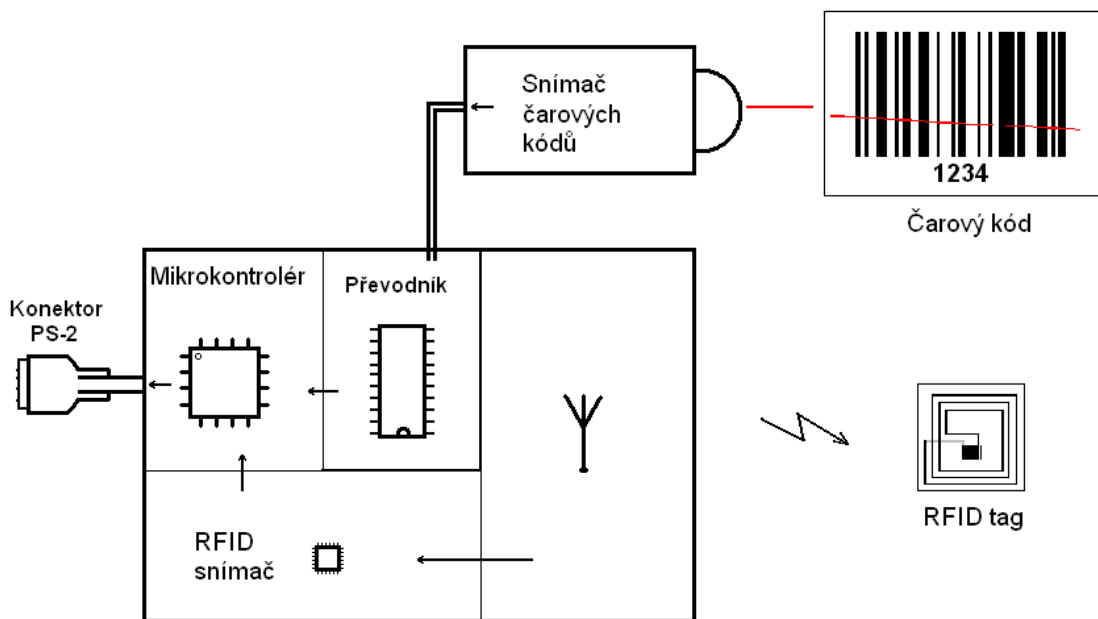


Obr. 4.8 Znárodnění RFID komunikace.

Zpracování se však neděje jednotlivě jako u čárových kódů, ale hromadně. Současná čtecí zařízení dokáží najednou načíst až několik set tagů za minutu [1], [2].

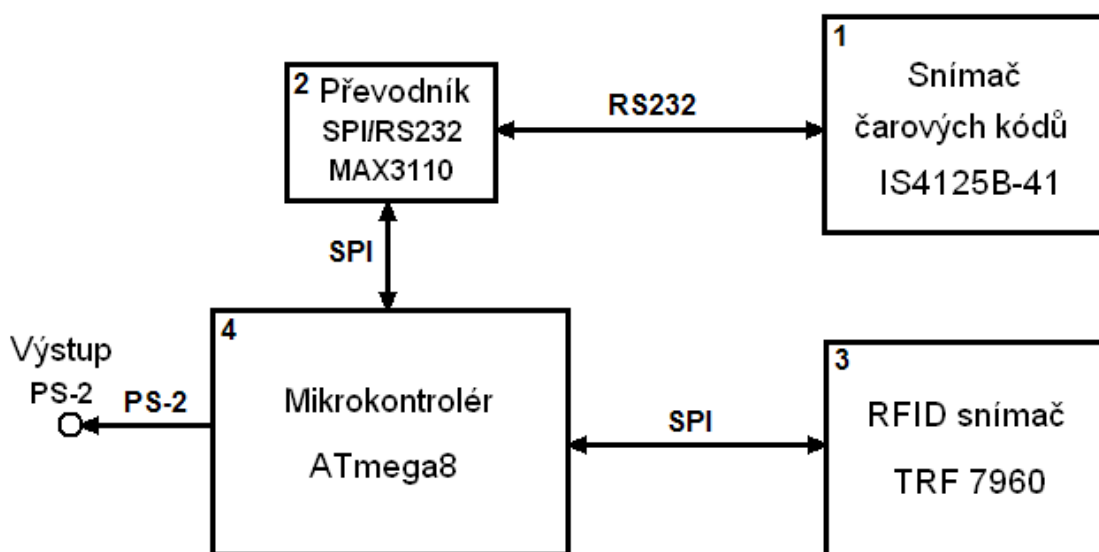
5 Navržené řešení

Tato čtečka by měla načítat RFID tagy pomocí radiových vln, nebo čárové lineární kódy pomocí laserového snímače. Ne ale současně, mikrokontrolér komunikuje vždy s jedním zařízením. Dostupné data se pošlou na PS-2 výstup. Celé zařízení je implementováno do jednoho celku, tak jako je znázorněno na obrázku 5.1.



Obr. 5.1 Názorné zapojení bloků čtečky kódů.

5.1 Blokové schéma



Obr. 5.2 Blokové schéma zařízení.

5.2 Rozhraní

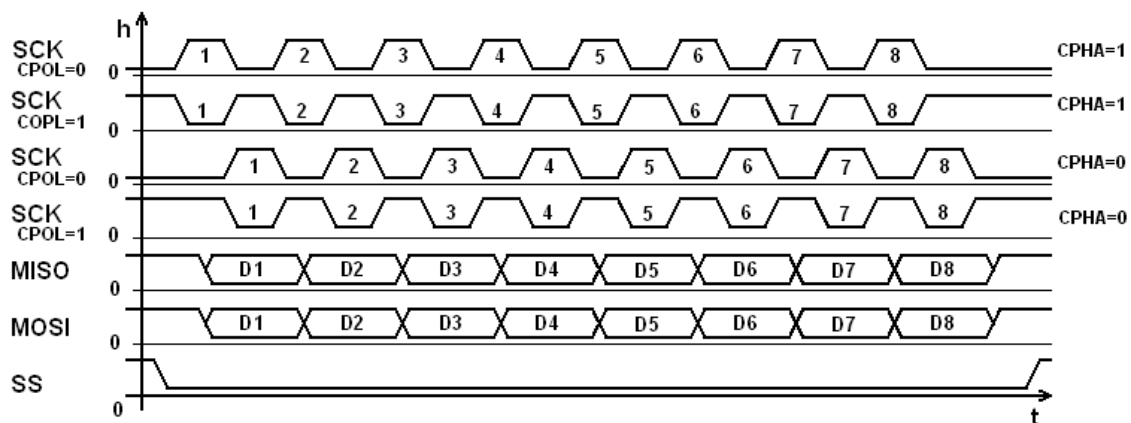
5.2.1 SPI

Zajišťuje vysokorychlostní přenos dat na synchronním sériovém periferním rozhraní. Může současně přijímat i vysílat data. Rozlišujeme:

- Master zařízení.
- Slave zařízení.

Synchronizace je zajištěna pomocí hodinového signálu na kanálu SCK, který generuje master zařízení. Pomocí kanálu MOSI jsou data vysílána ze zařízení master a na kanálu MISO vysílá data slave zařízení. Posledním komunikačním kanálem je SS, nebo-li slave select, který určuje se kterým zařízením komunikuje master [12].

Když chceme zahájit komunikaci pomocí mastru se slave zařízením, nejdřív musíme SS kanál shodit z výchozí úrovně logické 1 na pracovní úroveň logické 0. Tím se nám začne generovat hodinový signál na kanálu SCK. Pak se na základě hodin, vždy přenese 1 bit tam po MOSI a 1 bit zpět po MISO. Po přenose 8 bitů se kanál SS nastaví do klidového stavu logické 1, nebo se může pokračovat ve výměně dalších 8 bitů. Nejlépe to popisuje časový průběh na obrázku 5.3.



Obr. 5.3 Časový průběh na SPI kanálu

Kanály u SPI jsou ovládány pomocí 3 registrů:

- SPCR – registr řízení
 - bit 7: povoluje přerušení SPI.
 - bit 6: aktivuje SPI kanál.
 - bit 5: určuje pořadí bitů (LSB - nejméně významný bit na začátku = 1).
 - bit 4: konfiguruje mikrokontrolér jako master.

bit 3: volí polaritu hodin (CPOL).

bit 2: určuje fázi hodin (CPHA).

bit 1: kmitočet hodin.

bit 0: kmitočet hodin.

- SPSR – stavový registr

bit 7: příznak konce přenosu.

bit 6: příznak kolize zápisu.

- SPDR – datový registr

Ukázka odeslání dat přes SPI rozhraní:

Použitý obvod MAX3110 je připojen k mikrokontroléru ATmega8, který je tímto komunikačním rozhraním vybaven. Pokud by bylo více zařízení slave připojí se kanály SCK, MOSI a MISO na stejné piny a kanál SS musíme připojit na kterýkoli volný výstupní pin mikrokontroléru ATmega8. V mém případě je připojen k 1. bitu portu D. Nejdřív nastavím klidový stav na SS kanálu na úroveň logické 1, pak doporučuji doplnit timeout a následně shodím SS kanál do pracovní úrovně logické 0:

```
PORTB=0b00000010;
```

```
PORTB=0b00000000;
```

Nastavím řídicí registr na požadované hodnoty, dle dokumentace k MAX31100:

```
SPCR=0b01010011;
```

Aktivuji SPI kanál, první se přenáší nejvyšší bit MSB, konfiguruji mikrokontrolér jako master, polarita hodin je 0, fáze hodin je 0 a nakonec rychlost, která se určuje z kmitočtu mikrokontroléra: $1\text{Mhz}/128 = 7812,5\text{ b/s}$.

Přenos dat uskutečním funkcí, která se spustí příkazem umístěným kdekoli v těle programu:

```
SPI(0b11000100);
```

Pak se zavolá funkce SPI:

```
char SPI(char a) {                                     // odeslání registru přes SPI
    SPDR = a;                                           // naplní registr daty z proměnné a
    while((SPSR & 0x80) == 0); //čeká na příznak správného odeslání dat
    return (a);                                         //vrací hodnotu a
}
```

Ted už jsou přijata data v registru SPDR a stačí je uložit např. do proměnné c:

```
c= SPDR;
```

Na závěr ukončím SPI komunikaci zvednutím SS kanálu do klidového stavu logické 0: `PORTB=0b00000010;`

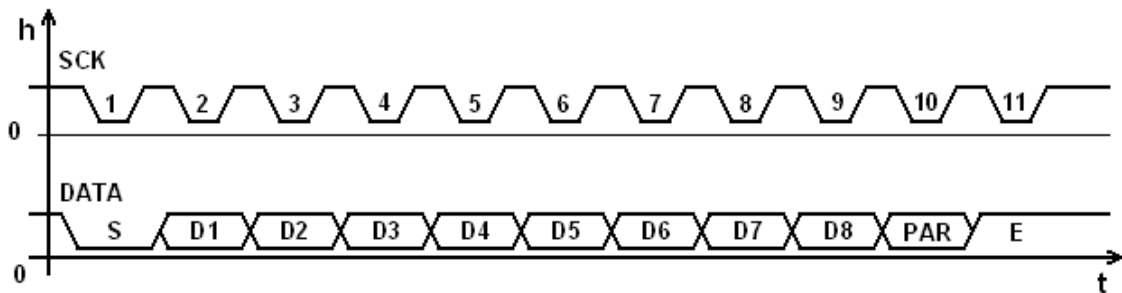
Poznámka: Logické úrovně na výstupech SPI je možné sledovat pomocí programu pro debugování (simulování) AVR studio.

5.2.2 PS-2

Jedná se o synchronní obousměrný sériový přenos informací, který dovoluje připojit jen jediné zařízení slave k zařízení master. Hodinový signál zaručí synchronizaci a vždy jej generuje slave.

Používá jen 2 komunikační kanály: hodinový signál - SCK a data – DATA. Informace se předávají s frekvencí 10-16 kHz.

Popíšu přenos ze zařízení slave (v mém případě mikrokontrolér Atmega8) a master (PC). Opačný přenos v projektu nepotřebuji. Klidový stav musí být zajištěn v obou kanálech na úrovni logické 1 po dobu minimálně 50 μ s. Po tomto timeoutu může slave samovolně začít přenášet bajt. Celkově je přenášeno 11 bitů. První bit označen jako start bit (S) na úrovni 0, pak následuje 8 datových bitů řazených od LSB (nejméně významný bit na začátku), dále pak paritní bit - parita lichá (pro sudý počet jedniček = 1) a nakonec stop bit (E) na úrovni 1. Data které jsou na kanálu DATA se načítají přesně uprostřed úrovně 0 hodinového signálu SCK. Časový průběh komunikace je zachycen na obrázku 5-4.



Obr. 5.4 Časový průběh PS-2 komunikace slave > master.

Tímto rozhraním ale není vybaven mikrokontrolér Atmega8, proto jsem se rozhodl tuto komunikaci naprogramovat pomocí časovače 1, kterým je tento obvod vybaven. Vycházel jsem ze skutečnosti, že pokud časovač dosáhne určitou hodnotu, mohou nastat změny na výstupu mikrokontroléru. Celkově jsem použil 4 registry pro správnou funkci časovače [11]:

- TIMSK – registr masek přerušení
bit 4: povolení přerušení od OCR1.

- TCCR1 – registr hodnoty časovač 1
- OCR1 – registr požadované hodnoty časovače 1.
- TIFR – registr příznaků časovače - stavový
bit 4: časovač přetekl nad požadovanou hodnotou.

Po nastavení parametrů časovače 1 (řádek 1-5, přílohy A) a zajištění logické 1 na oba kanály po dobu alespoň 50 μ s se testuje, zda časovač 1 dosáhl požadovanou hodnotu (řádek 8, přílohy A). Pokud ano, změní se hodinový kanál SCK (Clock) na opačnou logickou hodnotu. Tím jsem zajistil střídání logické 1 a logické 0 na výstupu T0.

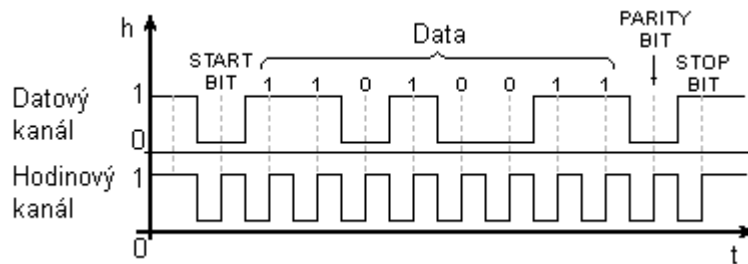
Pak v dalším podmínkovém příkazu testuji, zda je hodinový signál v logické 0, abych mohl posílat data přes PS-2 rozhraní (řádek 20, přílohy A). V něm dodržuji odesílání dat dle protokolu PS-2 tzn. start bit, 8x data bit, parita a stop bit (řádky 22-47, přílohy A). Paritní bit je počítán pomocí příkazu $>c = d[a] \wedge c;$ přičemž proměnná c je ve výchozí hodnotě 1.

Ukončení komunikace na PS-2 rozhraní je zajištěno počítáním změn hodinového signálu v proměnné b, která v podmínkové funkci vypne časovač 1 (řádek 14-18, příloha A).

Ukázalo se ale, že toto řešení při frekvenci 1MHz hodinového signálu mikrokontroléru Atmega8 nestačí přepínat logickou hodnotu kanálu SCK. Optimalizoval jsem proto program prefixem „#“. Ale ani to nestačilo a logická 1 se změní za dobu nejméně 50 μ s a logická 0 za nejméně 127 μ s. Rozdíl je jasný, protože ve stavu logické 0 se spouští funkce vyslání dat přes datový kanál. Je to dáno počtem instrukcí, které mikrokontrolér vykoná. Já ale potřebuji aby oba stavy trvaly 50 μ s. To odpovídá frekvenci hodinového signálu 10kHz. A navíc, aby se datový kanál nastavil za 25 μ s od nastavení hodinového signálu do logické úrovně 0.

Řešení proto bylo nastavit interní hodiny mikrokontroléru na 4 MHz, čím by se tato skutečnost eliminovala. Mikrokontrolér tuto frekvenci podporuje, ale zjistil jsem, že mnou použitý programátor STK500, komunikuje jen na 2 MHz a na vyšší frekvenci nenalezne připojený mikrokontrolér.

Proto jsem hledal jiné řešení komunikace s PS-2 rozhráním. Zjistil jsem, že velmi podobný časový průběh se používá na komunikačním rozhraní USART (viz obrázek 5.5), kterým je mikrokontrolér Atmega vybaven.



Obr. 5.5 Časový průběh komunikačního rozhraní USART.

Z anglického jazyka Universal Synchronous and Asynchronous serial Receiver and Transmitter, či-li sériový univerzální synchronní a asynchronní vysílač a přijímač, podporuje vysokorychlostní přenos dat, 5-9 bitů dlouhých datových slov. Pracuje s 1-2 stop bity a zvládne všechny parity. Master generuje hodinový signál [12]. Proto by nebyla na něm možná obousměrná komunikace pomocí PS-2 rozhraní. Navíc pro vysílání používá TX kanál a pro příjem RX kanál, synchronizace je realizována hodinovým signálem na XCK. Pro moje zapojení ale plně postačuje. Musel jsem ještě připojit výstup TX ke kanálu data, pomocí drátu na plošném spoji. USART využívá tyto registry [12]:

- UCSRA – stavový registr
bit 5: příznak prázdného bufferu.
- UCSRB – registr přerušení a konfigurace
bit 3: povolí vysílání přes TX výstup.
- UCSRC – registr řízení
bit 6: synchronní režim pro logickou 1.
bit 5,4: nastavení liché parity, oba bity na logickou 1.
bit 2,1: určuje počet datových bitů, oba pro 8 bitů nastavit na logickou 1.
bit 0: polarita hodin klesající, čili použijte logickou 1.
- UBRR – registr, jenž nese hodnotu, která dle vztahu 5.1 určuje počet přenesených bitů za sekundu.

$$BAUD = \frac{f_{osc}}{2(UBRR + 1)} \quad (5.1)$$

V mém případě se jedná o hodnotu 100, takže rychlost BAUD bude rovna 4950,5 b/s.

- UDR – datový registr

Ted už stačí konfigurační registry umístit do programu a naplnit datový registr UDR daty:

```
UCSRA=0x00;
```

```
UCSRB=0x08;
```

```
UCSRC=0xF7;
```

```
UBRRL=0x64;
```

```
UDR=0x0F;
```

Pomocí funkce while program počká, než se data přenesou přes rozhraní USART:

```
while ((UCSRA & 0x20) == 0x00);
```

Po úspěšném přenosu dat se uvolní rozhraní a je připraveno na další přenos dat.

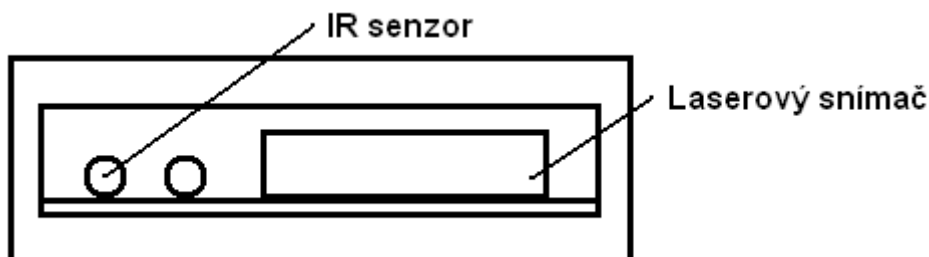
Poznámka: Logické úrovně na výstupu USART se nezobrazili v programu pro debugování (simulování) AVR studio.

5.3 Bloky

5.3.1 Snímač čárového kódu IS4125B-41

Kompaktní zařízení, které používá pro snímání laserový paprsek o vlnové délce 650 nm. Navíc obsahuje pohybový IR senzor. Čelní pohled je zobrazen na obrázku 5.6. Připojen je pomocí 12 žilového plochého kabelu.

Funguje tak, že když senzor zaznamená předmět před čtecí plochou aktivuje laserový paprsek a ten snímá čárový kód. Snímaná data převede na RS232 rozhraní a je připraven na další snímání čárového kódu. Pauza mezi dvěma načteními je min. 1 s.



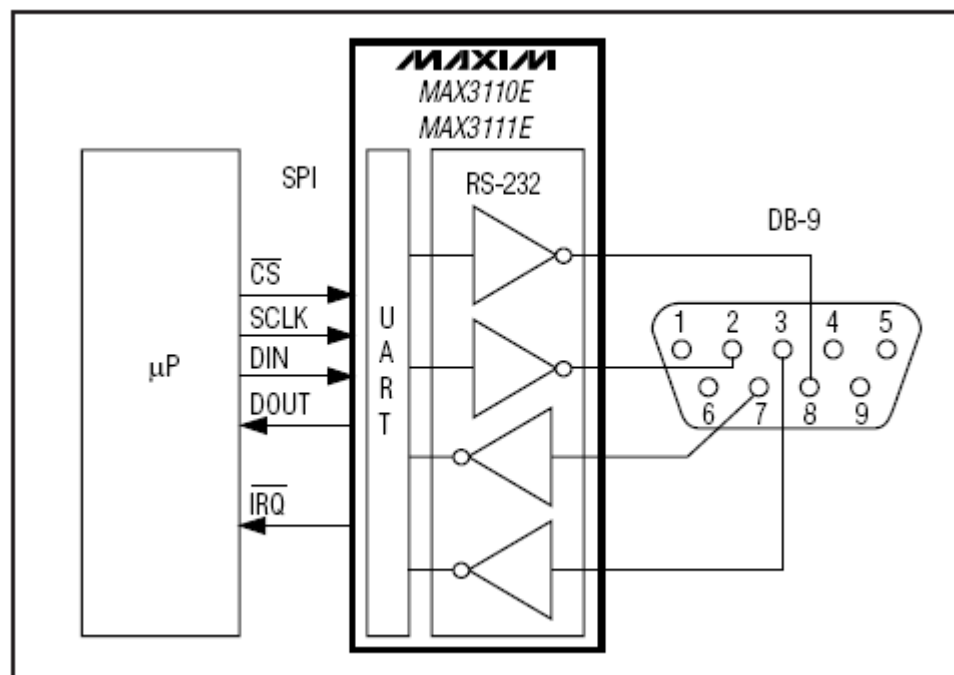
Obr. 5.6 Čelní pohled na snímač čárového kódu IS4125B-41

Doporučená vzdálenost mezi čárovým kódem a snímačem je v rozmezí 12,7 mm – 203.2 mm. Snímací rychlost čtečky je 52 cyklů za sekundu. Napájení je 5 V, při snímání má odběr proudu hodnotu 125 mA.

Výstup RS232 předává data rychlostí 9600 bit za sekundu, parita je mezera, datových bitů je 8, počet stop bitů je 1, řízení toku žádné [4].

5.3.2 Převodník SPI/RS232 MAX3110

Obvod, jehož typické zapojení je znázorněno na obrázku 5.7, popisuje zapojení kdy se vstupy a výstupy z 9 pinového konektoru RS232 připojí na obvod Max3110. Ten je vybaven nábojovou pumpou a mění úrovně $\pm 12V$ RS232 rozhraní na TTL úroveň +5V SPI rozhraní. Komunikace je obousměrná. Plně vyhovuje mému požadavku na převod dat. Je napájen 5V a jeho proudový odběr je $600\mu A$. Garantovaná rychlost přenosu pro SPI kanál je 230 kb/s, data se přenášejí na náběžnou hranu hodin, které jsou na počátku v stave LOW [14].



Obr. 5.7 Typické zapojení obvodu MAX3110.

Komunikace s tímto obvodem není složitá, používá 4 módy, v kterých je možné zapisovat nebo číst a to konfiguraci nebo data. Registry jsou 16 bitové. Na začátku se zapíše konfigurace pro RS232 komunikaci 0b1100010000001010:

- 15, 14 bit = 1: Zápis konfigurace.

- 10 bit = 1: FIFO přerušení je povoleno.
- 3, 1 bit = 1: rychlost komunikace 9600 b/s.

Obvod připraven na přenos dat. Nasnímáním čárového kódu pomocí snímače IS4125B-41 se změní hodnota datového bitu 15 a 14. Tuto skutečnost zjistím odesláním bajtu 0b0000000000000000:

- 15, 14 bit = 0: Čti data.

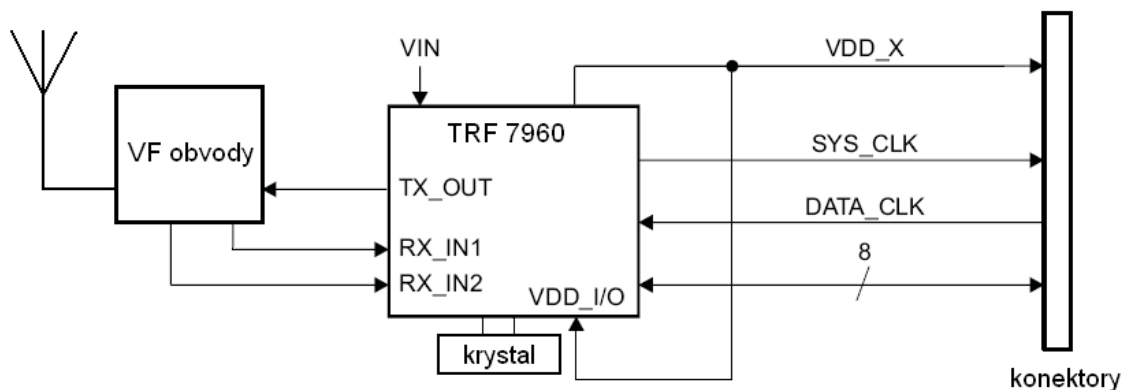
Obdržíme pak datový bajt např. ve tvaru 0b1100000000111000, kde:

- 15, 14 bit = 1: Značí přítomnost uložených dat v obvodě.
- 7 – 0 bit: 1. datový bajt načteného čárového kódu

5.3.3 RFID snímač TRF 7960

Integrovaný obvod TRF7960 komunikuje v pásmu HF 13,56 MHz (High frekvenci). Signál může být modulovaný PM, nebo AM modulací. Vzdálenost tagu od antény nesmí být větší než 5 cm. Největší komunikační rychlost obvodu je 848kb za sekundu. Napájecí napětí obvodu je 5 V při 10 mA oděru proudu [5].

Integrovaný obvod musí být připojen k vysokofrekvenčním obvodům, které pomocí antény přenášejí informace pomocí elektromagnetických vln. Standardní zapojení, je znázorněno na obrázku 5.8.



Obr. 5.8 Blokové zapojení modulu snímače RFID kódů.

Obvod je vybaven kanály pro SPI sériovou komunikaci. Jak jsem ale později zjistil, konfigurace není vůbec jednoduchá, protože výrobce nepřipravil plnohodnotnou technickou dokumentaci. Měl jsem k dispozici dokumentaci ze září 2006 [15] ve které byly strohé informace o komunikaci prostřednictvím SPI rozhraní s SS kanálem. Nastavil jsem ji dle časového průběhu v této dokumentaci na straně 35.

Integrovaný obvod neodpovídal na žádnou komunikaci, kterou jsem do něj posílal. Před nedávnem jsem na webu výrobce našel novou updatovanou dokumentaci [16], která byla vytvořena v únoru toho roku. S potěšením jsem se pustil do práce.

V dokumentu již bylo základní zapojení integrovaného obvodu, rozšířil se popis komunikace přes SPI. Uvádělo se zde, že zápis dat do integrovaného obvodu probíhá při nastavení polarity hodinového signálu $CPOL = 0$ a fáze hodin $CPHA = 1$, naopak čtení dat z integrovaného obvodu při nastavení $CPOL = 0$ a $CPHA = 0$. Časové průběhy však neodpovídali standardu SPI dle [13]. Konfiguraci jsem takto nastavil. Odeslal jsem pro začátek hodnotu `0b01000010` do paměti a pokusil se data načíst zpět do mikrokontroléru. V přijatých datech jsem ale měl hodnotu `0b01100011`.

Domnívám se, že byla nevhodně nastavena fáze hodin i když jsem dodržel technickou dokumentaci výrobce. Přijatá data jakoby protekla do sousedního bitu. Změnil jsem proto nastavení dle své domněnky a to $CPOL = 1$ a $CPHA = 1$. Přijatá data už měla stejnou podobu jako odeslaná. Zkusil jsem proto aktivaci RFID komunikace a načtení dat z mikročipu RFID pomocí příkazu uvedených v dokumentaci. Bohužel se mi nepodařilo dostat do datového registru žádné korektní data a kvůli nedostatku času jsem ponechal slepé testování této komunikace.

5.3.4 Mikrokontrolér ATmega8

Je to velmi levný integrovaný obvod, který používá architekturu RISC. Napájecí napětí obvodu je 5 V. Paměť využívá 512b SRAM EEPROM, znamená to, že je možné do ní zapisovat, nebo ji mazat. A dále využívá dočasnou Flash paměť 8 Kb. Najdeme na něm 20 vstupně-výstupných vývodů. Pracuje na frekvenci 1 MHz prostřednictvím interního rezonátoru [7].

Je vybaven SPI rozhraním pro sériový přenos dat, který využiji pro komunikaci s jinými integrovanými obvody. Dále je zde USART rozhraní, kterým budu vysílat data na PS-2 vstup klávesnici počítače.

5.4 Realizace

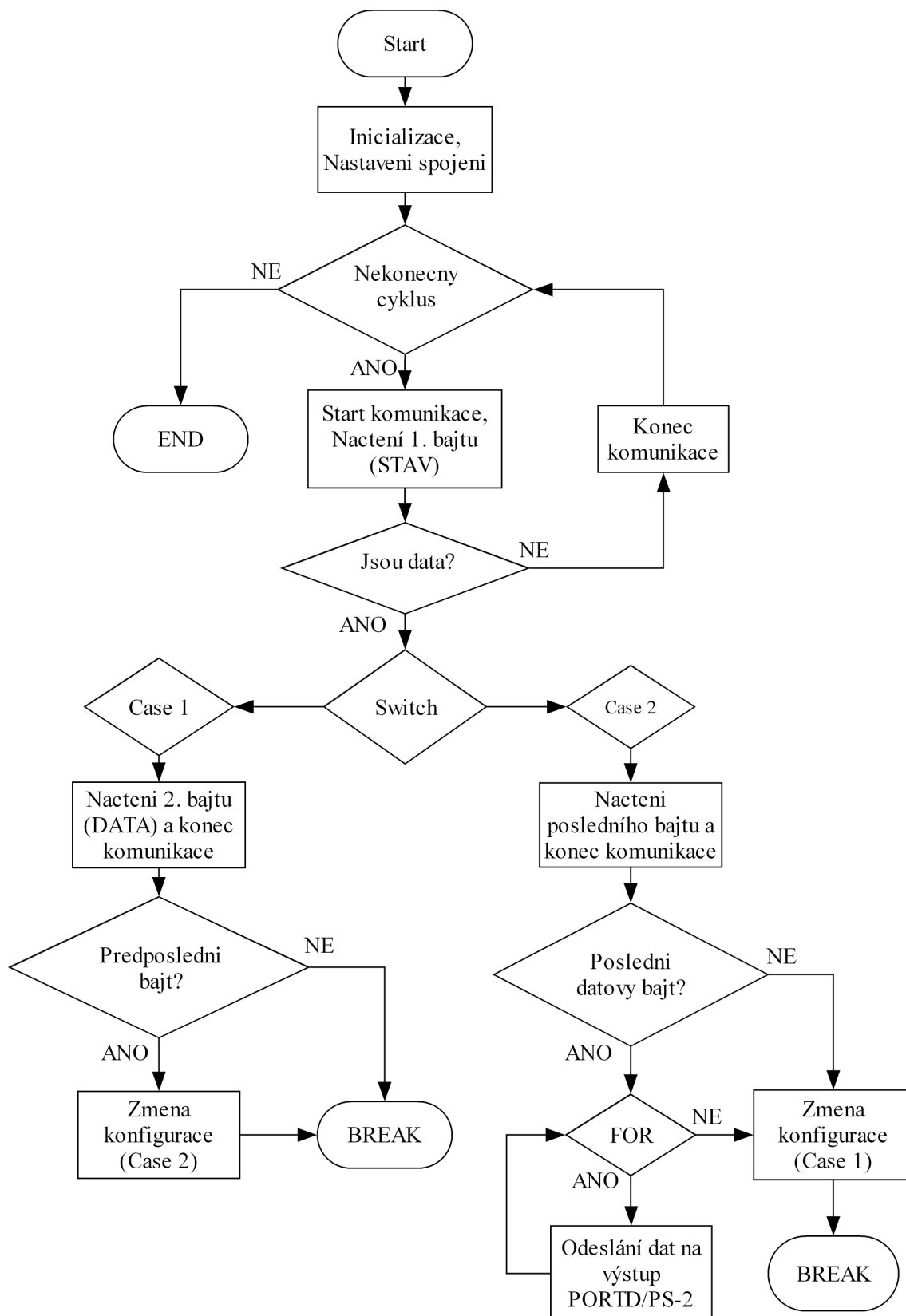
5.4.1 Software

Zdrojový kód pro mikrokontrolér Atmega8 jsem psal v prostředí CodeVisionAVR verze 1.25.2 od firmy HP InfoTech a simulace jsem prováděl v programu AVR Studio 4 verze 4.13 od firmy Tenison Design Automation.

Pro nahrávání programu jsem měl k dispozici programátor STK 500 od firmy Atmel, který byl připojen k počítači pomocí RS232 rozhraní a k zhotovenému zařízení byl připojen přes 6 žilový kabel ISP (In-System Programming), který používá SPI (Serial Peripheral Interface) rozhraní pro přenos dat.

Snímání čárového kódu do mikrokontroléru:

Pro lepší přehlednost jsem vytvořil vývojový diagram snímání čárového kódu obrázek 5.9. Na začátku inicializuji všechny použité registry a nastavuji je do potřebných úrovní. Následně je spuštěn nekonečný cyklus. SPI mikrokontrolér používá jen 8 bitové registry pro komunikaci, využil jsem proto této skutečnosti a pro zjištění zda jsou přítomny data na zpracování vysílám jen 1. bajt hodnoty 0b00000000 přes SPI rozhraní (řádek 12, příloha B). Objeví-li se v přijatém bajtu pořadí znaků 0b11000000 (řádek 14, příloha B) přejde se do stavu Case 1 a pokračuje přenos dat 2. bajtu přes SPI rozhraní. Po ukončení komunikace SPI spouštím proces pro ukládání dat do pole d[e] (řádek 20, příloha B). Datový registr obsahuje přenášená data velikosti 8 bit, které značí ASCII kód, čili znak, dle kódování Windows-1250.



Obr. 5.9 Vývojový diagram čtení čárového kódu.

Data se přenáší dokud se neobjeví předposlední hodnota 0b00001101 (řádek 22, příloha B), měl by to být předposlední znak, který znamená konec dat nasnímaného čárového kódu. Ten pak změní konfiguraci na Case 2 (řádek 24, příloha B) a je očekáván poslední datový bajt 0b00001010 (řádek 22, příloha B). Pokud by nebyl poslední bajt dle předlohy, program se vrací zpět do Case 1 a ukládá přijatá data čárového kódu do registru d[e]. Pravděpodobnost je ale velmi nízká, protože datový znak dle ASCII předposledního bajtu je „ ␣ “ a tento znak se vyskytuje jen v specifických kódováních čárového kódu. Poslední bajt je taky specifický, dle ASCII „ ␣ “. Pokud se tedy tento znak objeví v datech je ukončena komunikace s SPI a program pokračuje do odeslání dat (řádek 35, příloha B) na registr D mikrokontroléru ATmega8. Toto řešení jsem použil proto, abych mohl data zobrazit na připojené LED diody. Používal jsem to při testování, protože je to asi jediný způsob, jak ověřit a zjistit, zda se nasnímaná data přenesly korektně do mikrokontroléru. Ve finální verzi je toto nahrazeno funkcí, která přenáší data přímo na PS-2 rozhraní a popis je uveden v další sekci. Program se následně nastaví do výchozího stavu a čeká na nasnímáním dalšího čárového kódu. Hlavní cyklus programu je zachycen v příloze B.

Celkově je software nastaven na přenos 1-16 znaků, nejčastěji se setkáme s délkou kódu 13 a 8 znaků EAN kódování.

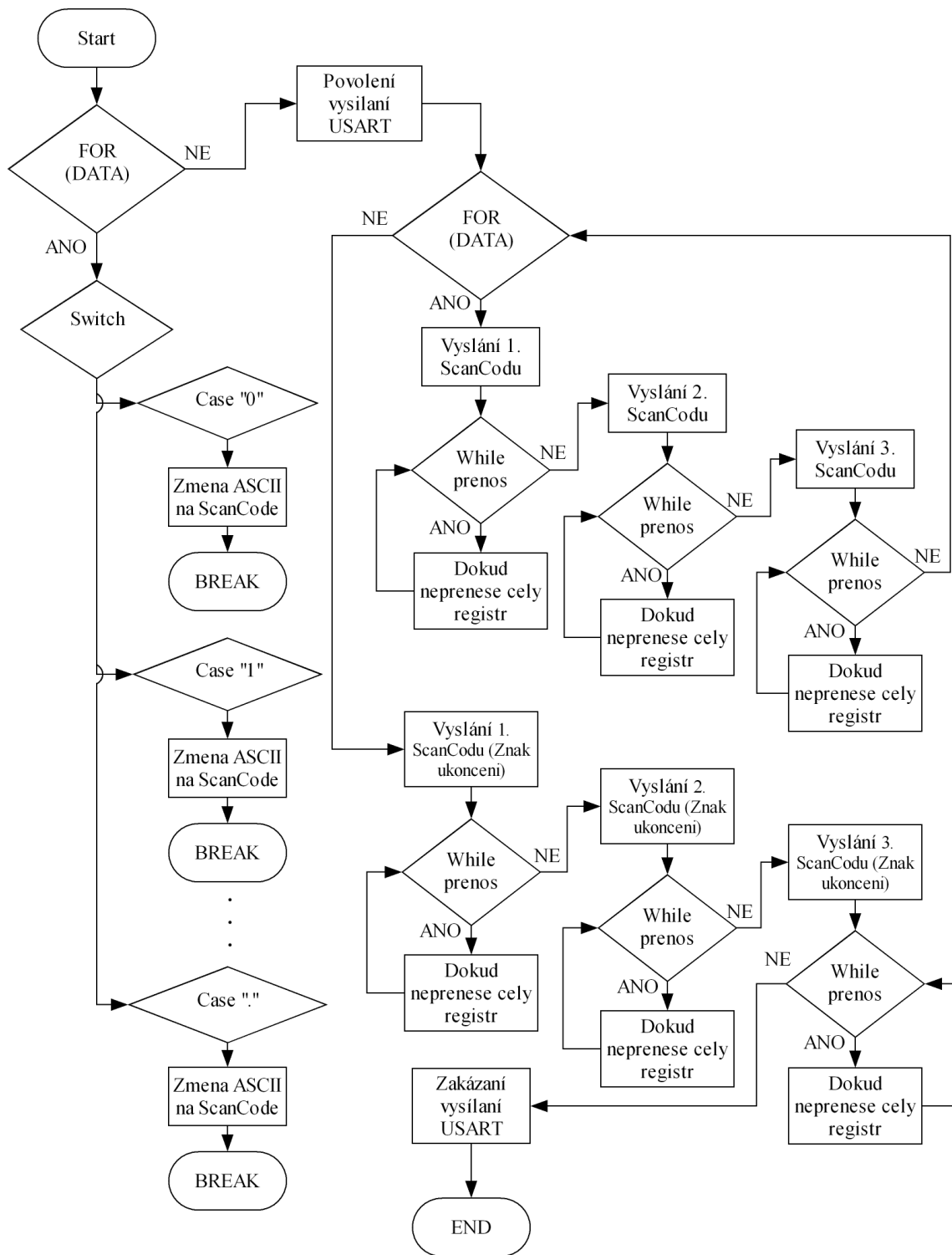
Data uložena v poli d[e] mohou mít následující podobu:

00111000, 00110101, 00111001, 00110101, 00110000, 00110001, 00110011,
00110110, 00110001, 00110010, 00110011, 00110000, 00111001.

Odeslání uložených dat z mikrokontroléru:

Data uložena v mikrokontroléru v proměnné d[e] jsou uložena ve formě kódování ASCII. Ale PS-2 tedy i klávesnice používá kódování tzv. Scancode. Proto jsem musel provést konverzi kódování (řádek 3-44, příloha C). Pro lepší přehlednost uvádím vývojový diagram na obrázku 5.10.

Definoval jsem stavy pro nepoužívanější znaky v čárových kódech: „0-9“, tečka, pomlčka a lomítko. Použil jsem na to příkaz switch, který ale nedefinuje jiný použitý znak než uvedené znaky. Proto tento znak není konvertován a buď se na PC vůbec nezobrazí, nebo bude zobrazen jako nesmysl.



Obr. 5.10 Vývojový diagram odeslání dat na PS-2.

Testoval jsem tuto konverzi i pomocí příkazu if, v kterém můžu definovat „všechny jiné znaky“ díky instrukci else a nahradit ji např. znakem „X“. Jenomže se to projevilo jako nevhodné, protože cca každý 3. načtený čárový kód v pořadí měl v sobě znak X, i-když se v čárovém kódu vykytovali jen číselné hodnoty. Důvod jsem nezjistil.

Konfigurační registry byly nastaveny při inicializaci v předchozím popisu snímání čárového kódu. Program dále povoluje výstup TX pro přenos dat pomocí USART (řádek 46, příloha C). Vše je připraveno pro přenos dat přes PS-2 rozhraní.

Odesílá se první Scancode, který pro PC značí zmáčknutí klávesy. Vymáčknutí klávesy se provádí odesláním hodnoty 0x0F a následuje Scancode vymáčknuté klávesy (řádek 47-62, příloha C). Proto se pro jeden načtený znak čárového kódu musí odeslat přes PS-2 tři hodnoty Scancode. Po přenosu celé datové proměnné d[e] je ještě zaslán poslední znak „Enter“, který nám v aplikaci např. odřádkuje zobrazený čárový kód. Také je zastoupen třemi Scancode-mi (řádek 63-75, příloha C). Po tomto odeslání dat, je TX kanál deaktivován a program je nastaven do výchozího stavu.

Na monitoru se nám zobrazila hodnota: 8595013612309. Načítaný kód je EAN13, první 3 číslice symbolizují krajinu původu Česká republika a dalších 5 firmu CENTROPEN a.s. [8].

5.4.2 Hardware

Návrh plošného spoje byl realizován v programu Eagle verze 4.15 firmy CadSoft Computer. Zapojení celého zařízení jsem dělal na základě technických dokumentací výrobců integrovaných obvodů [12], [14], [16], schéma celého zapojení je umístěna v příloze D.

Součástky jsou osazeny na oboustranném plošném spoji viz. příloha E. V ní je taky umístěno rozmístění součástek na plošném spoji. Využity jsou v co největší míře SMD pouzdra.

Dominantní částí celého zapojení je vestavěná anténa pro bezkontaktní přenos dat pomocí technologie Rfid. Je velmi důležité aby součástky připojeny k této části měli co nejmenší vzdálenost mezi sebou, aby se co možná nejméně projevil rušení které vzniká mezi cestičkami na plošném spoji. Nachází se zde vysokofrekvenční signál 13.56 MHz.

Na plošném spoji je umístěn zmíněný integrovaný obvod pro Rfid komunikaci TRF7960 (IO1), tento obvod je chlazen přes plošky zesponu součástky a pomocí prokovů je teplo odváděno na měděnou plochu druhé strany plošného spoje. Nad ní je umístěn převodník RS232 komunikace na SPI přenos - MAX3110 (IO2) od firmy Maxim a samotný mikroprocesor ATmega8 (IC1) od firmy Atmel.

Zapojení disponuje konektorem pro připojení čtečky čárového kódu IS4125B41 (X2) firmy Metrologic. Snímač se připojuje pomocí 12-ti žilového plochého kabelu. Vedle konektoru je umístěna signalizační Led dioda režimu čtečky „Aktivní“ (LED2 - Zelená), rozsvítí se ve chvíli kdy je rozkmitán laserový paprsek snímače. Led dioda, která signalizuje načtené znaky čárového kódu (LED1 - Červená) je umístěna na protější straně konektoru. Vlevo jsou pak vyvedeny 3 vypínače mikrokontroléru - Portu C (SV3) a konektor pro odesílání dat k dalšímu zařízení pomocí I2C komunikace (X1).

Na levém okraji plošného spoje jsou vyvedeny všechny bity portu D na Led diody (LED3 – LED10). Pomocí nich se zobrazují data uložené v registru mikrokontroléru. Pod Led diodami se nachází přepínač (SV2), který přepíná 2 bity Portu D na konektor PS-2 (SV4). Bity jsou využity pro přenos dat přes PS-2. Datový kanál je ještě řešen drátem. V blízkosti přepínače je konektor pro programování mikrokontroléru ATmega (SV1). Konektor je uzpůsoben pro programátor STK 500 firmy Atmel. V těsné blízkosti se nachází pin pro volbu, zda má být obvod napájen z konektoru pro programování, nebo nikoli.

V střední části jsou osazeny piny pro shození integrovaných obvodů do režimu „shutdown“ (JP1 – JP3), pokud jsou ve stavu „Open“.

Celé zařízení je napájeno 5 V přes PS-2 rozhraní počítače.

6 Závěr

Cílem této bakalářské práce bylo navrhnout a realizovat čtečku čárových kódů, kterou reprezentuje integrovaný obvod IS4125 a čtečku RFID čipů, kterou reprezentuje integrovaný obvod TRF7960. Obě čtečky obsluhuje jeden mikrokontrolér Atmega8, který přijímá data prostřednictvím SPI rozhraní a posílá je na vstup klávesnice PS-2 do osobního počítače. Slučuje do jednoho zařízení snímání pomocí optického rozhraní a čtení přes elektromagnetické vlny.

Realizace načtení dat ze čtečky RFID čipů se mi nepovedla, nejspíše kvůli špatné technické dokumentaci výrobce. Navzdory všem překážkám, kterým jsem čelil se ostatní částí povedlo dokončit. V této práci jsem se naučil plně pracovat s technickou dokumentací výrobce, i když mi na začátku chyběly znalosti v řešené problematice.

Uvědomil jsem si jak velký je rozdíl rozumět problematice na teoretické úrovni oproti praktickému řešení dané problematiky. Už jen proto, že během testování a programování jsem smazal integrovaný obvod programátora a celkem jsem zničil 5 kusů mikrokontroléru ATmega8.

Díky této práci jsem z části pronikl do problematiky automatické identifikace dat a naučil jsem se pracovat s komunikačním rozhraním SPI a PS-2. Zjistil jsem jak funguje rozhraní USART a obeznámil se s fungováním časovačů a přerušení mikrokontroléru v prostředí jazyka C. V neposlední řadě jsem se naučil využívat debugování vytvořeného programu a zdokonalil si ovládání programu Eagle pro tvorbu plošných spojů.

V práci jsem se snažil srozumitelně vysvětlit všechny použité komunikační rozhraní, aby ji mohli využít čtenáři, kteří s danou problematikou začínají, nebo v ní hledají inspiraci.

Samotný výrobek stačí připojit ke vstupu klávesnice PS-2 počítače, otevřít v počítači např. poznámkový blok. Přiblížit před snímač čárový kód a nasnímaná data se vypíší na řádek do aplikace. Po posledním přeneseném znaku se odřádkují a zařízení je připraveno pro snímání dalšího čárového kódu.

7 Použitá literatura

- [1] Bartech, s.r.o.. bartech : RFID [online]. [2006] [cit. 2007-12-19]. Dostupný z WWW: <<http://www.bartech.cz/rfid/index.htm>>.
- [2] Project Invest, s. r. o.. RFID portál [online]. [2007] [cit. 2007-12-19]. Dostupný z WWW: <http://www.rfidportal.cz/index.php?page=rfid_obecne>.
- [3] IDENTCODE, s.r.o. [online]. 2007 [cit. 2007-12-19]. Dostupný z WWW: <http://www.identcode.sk/index.php?option=com_content&task=view&id=15&Itemid=39>.
- [4] Metrologic Instruments, Inc. [online]. 2003-2007 [cit. 2007-12-19]. Dostupný z WWW: <<http://www.metrologic.com/corporate/products/oem/is4125/>>.
- [5] Texas Instruments Incorporated. [online]. c1995-2007 [cit. 2007-12-19]. Dostupný z WWW: <<http://focus.ti.com/docs/prod/folders/print/trf7960.html>>.
- [6] DAVID, Martin. Ako sa vyznať v čiarových kódoch?. Zive.sk [online]. 2003 [cit. 2007-12-19]. Dostupný z WWW: <<http://www.zive.sk/Autori/Ako-sa-vyznat-v-ciarovych-kodoch/Uvod/sc-44-sr-1-a-256457-ch-164097/default.aspx>>.
- [7] Atmel Corporation. Atmel Corporation [online]. 2007 [cit. 2007-12-19]. Dostupný z WWW: <http://www.atmel.com/dyn/products/product_card.asp?family_id=607&family_name=AVR+8%2DBit+RISC+&part_id=2004>.
- [8] EAN Česká republika. EAN Czech [online]. c2006 [cit. 2009-04-17]. Dostupný z WWW: <<http://www.gs1cz.org/>>.
- [9] Motorola Symbol SE1524ER [online]. c2009 [cit. 2009-04-17]. Dostupný z WWW: <http://www.motorola.com/business/XE-EN/SE1524ER_Loc:XU-EN,XC-EN,XM-EN,XE-EN,+XN-EN,PK-EN,XF-EN.do?vnextoid=69f243d7e346c110VgnVCM1000008406b00aRCRD>.
- [10] Bible automatické identifikace. IDENTCODE [online]. 2003 [cit. 2009-05-30]. Název: B_200302.pdf
- [11] KARAS, Ondřej. Čítače a časovače. *Kvetakov.net* [online]. 2006 [cit. 2009-05-24]. Dostupný z WWW: <<http://www.kvetakov.net/clanky/avr/42-citace-a-casovace.html>>. ISSN 1803-1021.

- [12] ATMEL CORPORATION. Atmega8. *ATmega8* [online]. 2008 [cit. 2009-05-25].
- [13] MATOUŠEK, David. *Práce s mikrokontroléry Atmel ATmega16*. [s.l.] : BEN - technická literatura, 2006. 319 s. ISBN 8073001748.
- [14] MAXIM. MAX3110E Full Data Sheet. *MAX3110E-MAX3111E* [online]. 2000 [cit. 2009-04-24].
- [15] TEXAS INSTRUMENTS. TRF7960/61 Multi-Standard Fully Integrated 13,56-MHz RFID (Rev. A). *Slou168a* [online]. 2006 [cit. 2008-11-04].
- [16] TEXAS INSTRUMENTS. Multi-Standard Fully Integrated 13,56-MHz Radio Frequency Identification (RFID) (Rev. C). *trf7960* [online]. 2009 [cit. 2009-04-05].

8 Abecední přehled použitých zkratk a symbolů

AVR - Automatic Voltage Regulator (automatický regulátor napětí)
BAUD – přenosová rychlost
CCD - Charge-Coupled Device (zařízení s vázanými náboji)
EAN - European Article Number (evropské číselná jednotka zboží)
EPC - electronic product code (elektronický produkt kód)
IR – Infra Red senzor (infra červený senzor)
MISO – Master In Slave Out (Vysílací kanál SPI)
MOSI – Master Out Slave In (přijímací kanál SPI)
PC – Personal Computer (osobní počítač)
PS-2 – konektor pro připojení klávesnice k osobnímu počítači
RISC - Reduced Instruction Set Computer (počítač s redukovanou instrukční sadou)
RF – Reader Frequency (frekvence čtečky)
RFID – Radio-frequency Identification (Rádio-frekvenční identifikace)
RTF - Reader Talk First (čtečka mluví první)
SPI - Serial Peripheral Interface (sériové rozhraní)
SS – Slave Select (povolení komunikace se slave zařízením v SPI)
Tag – Klíčové slovo složené s identifikačních hodnot
TTF - Tag Talks First (tag mluví první)
TTL - třída logiky digitálních integrovaných obvodů
USART - Universal Synchronous Asynchronous Receiver Transmitter
(Synchronní a asynchronní seriové rozhraní)

9 Přílohy

Příloha A: Ukážka odeslání dat na rozhraní PS-2 s použitím časovače 1.

Příloha B: Hlavní cyklus zdrojového kódu pro zpracování dat z čtečky
čárového kódu.

Příloha C: Odeslání dat na rozhraní PS-2 pomocí protokolu USART.

Příloha D: Schéma zapojení.

Příloha E: Plošný spoj.

Příloha F: Obsah přiloženého CD

Příloha A: Ukážka odeslání dat na rozhraní PS-2 s použitím časovače 1.

```
1  TCCR1A = 0b00000000;           // vrchní bajt hodnoty časovače 1
2  OCR1A = 0x000A;
3  TCNT1 = 0x0001;                // počáteční hodnota časovače 1
4  TIFR = 0x14;                   // stavový registr
5  TCCR1B = 0b00000001;           // spodní bajt hodnoty časovače 1
6  while(1)
7  {
8      if( (TIFR & 0x10) == 0x10) // jestli časovač dosáhl požadovanou hodnotu
9      {
10         TIFR = 0x14;             // vymaž stavový registr
11         Clock = Clock ^ 1;       // změň polaritu hodinového signálu
12         TCNT1 = 0x0001;         // nastav výchozí hodnotu časovače 1
13         b++;
14         if( b == 22)             // poslední změna hodinového signálu
15         {
16             delay_us(12000);
17             b = 0;
18             TCCR1B = 0b00000000; // ukončí činnost časovače
19         }
20         if(Clock == 0)           // jestli jsou hodiny v logické 0
21         {
22             a++;
23             #if(a == 1)           // start bit
24                 Data = 0;
25             #elif(a == 10)       // paritní bit
26                 {
27                     c = d[a] ^ c; // výpočet paritního bitu - lichá
28                     if(c %2)
29                         Data = 1;
30                     else
31                         Data = 0;
32                     c = 1;
33                 }
34             #elif(a == 11)       // stop bit
35                 {
36                     Data = 1;
37                     a=0;
38                     por=1;
39                     f++;
40                     Test1 = 1;
41                 }
42             #else                 // data bit
43                 {
44                     Data = (d[a] & por);
45                     por++;
46                 }
47             #endif
48     } }
```

Příloha B: Hlavní cyklus zdrojového kódu pro zpracování dat z čtečky čárového kódu.

```
1  PORTB=0b00000100;
2  SPCR=0b01010011;
3  SPI(0b11000100); //Nastavení SPI kanálu
4  SPI(0b00001010); //
5  PORTB=0b00000110;
6  delay_us(50);
7  b=1;
8  e=0;
9  while(1) //Nekonečný cyklus.
10 {
11 PORTB=0b00000100;
12 SPI(0b00000000); //Načti 1. bajt.
13 Test = SPDR;
14 if (Test == 0b11000000) //Jsou data?
15 {
16     switch(prepare)
17     {
18         case 1: e++;
19             SPI(0b00000000);
20             d[e] = SPDR;
21             PORTB=0b00000110;
22             if (d[e] == 0b00001101) //znak Alt + 13 (nota).
23                 {
24                     prep=2;
25                     pocet=e; //proměná v které je uložen
26                 } //počet znaků kodu.
27                 break;
28
29             case 2: e++;
30                 SPI(0b00000000);
31                 d[e] = SPDR;
32                 PORTB=0b00000110;
33                 if (d[e] == 0b00001010) //znak Alt + 11 (kruh ve ctverci).
34                     {
35                         for (e=0; e < pocet; e++)
36                             {
37                                 PORTD=d[e]; //odešle data na výstup PORTD,
38                                 delay_ms(1000); //nahrdí se smičkou pro odeslání
39                                 PORTD=0b00000000; //na PS-2.
40                                 delay_ms(500);
41                             }

```

```
42         pocet=0;
43         e=0;
44     }
45     prep=1;
46     break;
47 }
48 }
49 PORTB=0b00000110;
50 delay_us(100);
51 }
```

Příloha C: Odeslání dat na rozhraní PS-2 pomocí protokolu USART.

```
1  for (e=0; e < pocet; e++) //Konverze ASCII na ScanCode
2      {
3          switch(d[e])
4          {
5              case 0x30:
6                  d[e]=0x70; //Znak "0"
7                  break;
8              case 0x31:
9                  d[e]=0x69; //Znak "1"
10                 break;
11             case 0x32:
12                 d[e]=0x72; //Znak "2"
13                 break;
14             case 0x33:
15                 d[e]=0x7A; //Znak "3"
16                 break;
17             case 0x34:
18                 d[e]=0x6B; //Znak "4"
19                 break;
20             case 0x35:
21                 d[e]=0x73; //Znak "5"
22                 break;
23             case 0x36:
24                 d[e]=0x74; //Znak "6"
25                 break;
26             case 0x37:
27                 d[e]=0x6C; //Znak "7"
28                 break;
29             case 0x38:
30                 d[e]=0x75; //Znak "8"
31                 break;
32             case 0x39:
33                 d[e]=0x7D; //Znak "9"
34                 break;
35             case 0x2D:
36                 d[e]=0x4E; //Znak "-"
37                 break;
38             case 0x2F:
39                 d[e]=0x4A; //Znak "/"
40                 break;
41             case 0x2E:
42                 d[e]=0x49; //Znak "."
```

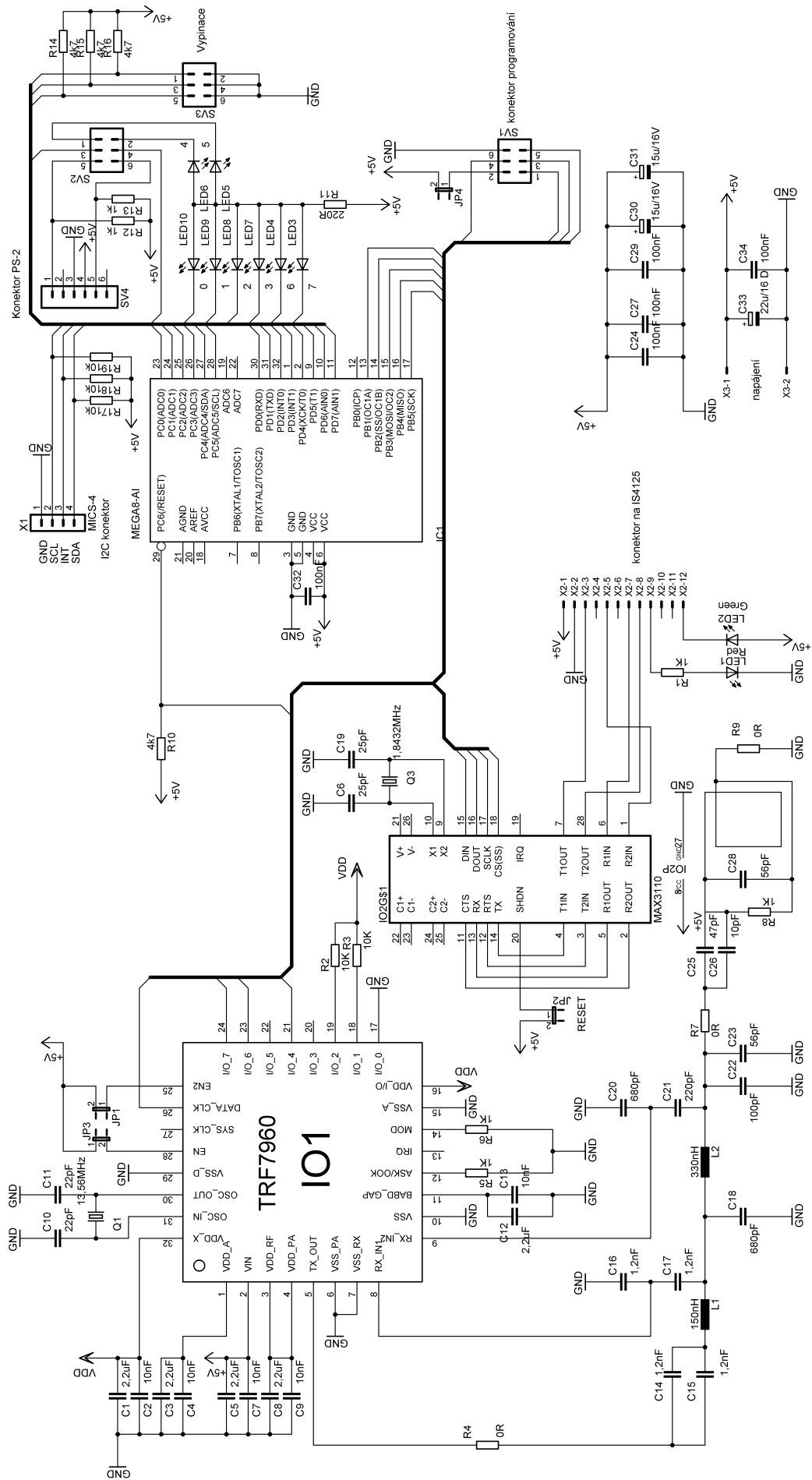


```

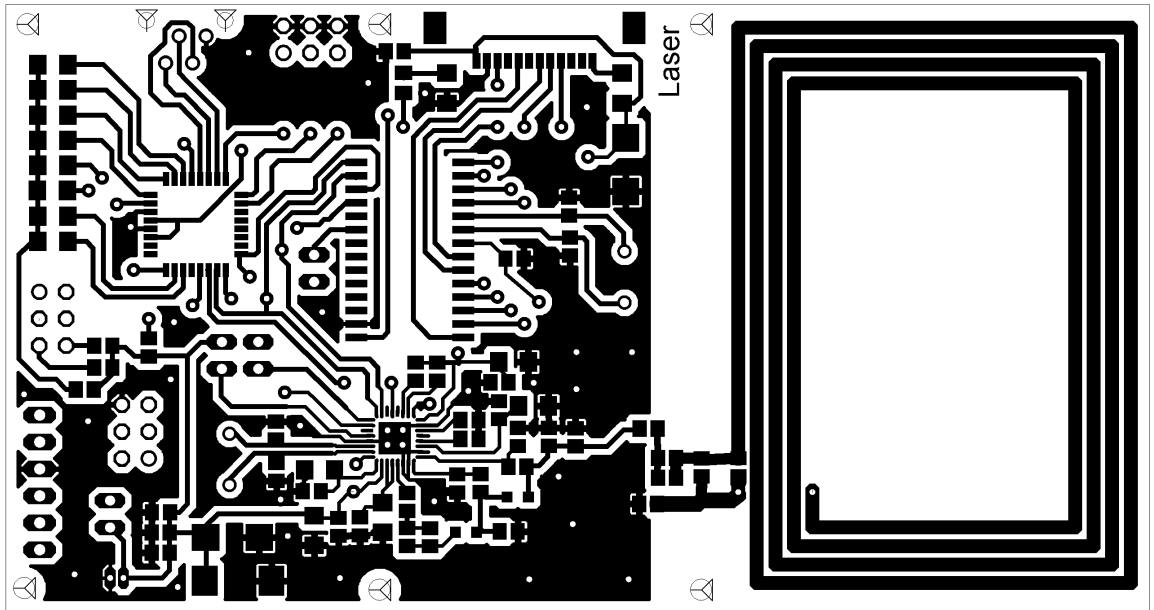
43             break;
44         }
45     }
46     UCSRB=0x08;
47     for (e=0; e < pocet; e++)    //Odeslání dat na PS-2 rozhraní
48     {
49         //if ((UCSRA & 0x20)== 0x20)
50         UDR=d[e];
51         UCSRA=0x00;
52         while ((UCSRA & 0x20)== 0x00);
53         delay_us(1000);
54         UDR=0xF0;                //Znak vypuštění klávesy
55         UCSRA=0x00;
56         while ((UCSRA & 0x20)== 0x00);
57         delay_us(1000);
58         UDR=d[e];
59         UCSRA=0x00;
60         while ((UCSRA & 0x20)== 0x00);
61         delay_us(1000);
62     }
63     UDR=0x5A;                    //Znak klávesy ENTER
64     UCSRA=0x00;
65     while ((UCSRA & 0x20)== 0x00);
66     delay_us(1000);
67     UDR=0xF0;
68     UCSRA=0x00;
69     while ((UCSRA & 0x20)== 0x00);
70     delay_us(1000);
71     UDR=0x5A;
72     UCSRA=0x00;
73     while ((UCSRA & 0x20)== 0x00);
74     delay_us(1000);
75     UCSRB=0x00;

```

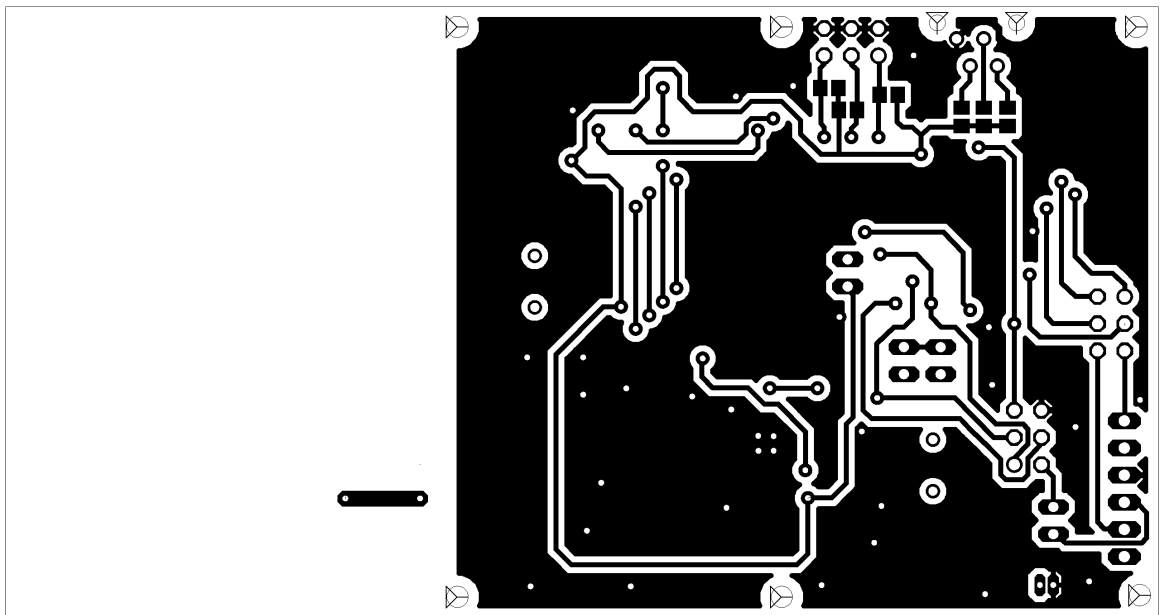
Příloha D: Schéma zapojení.



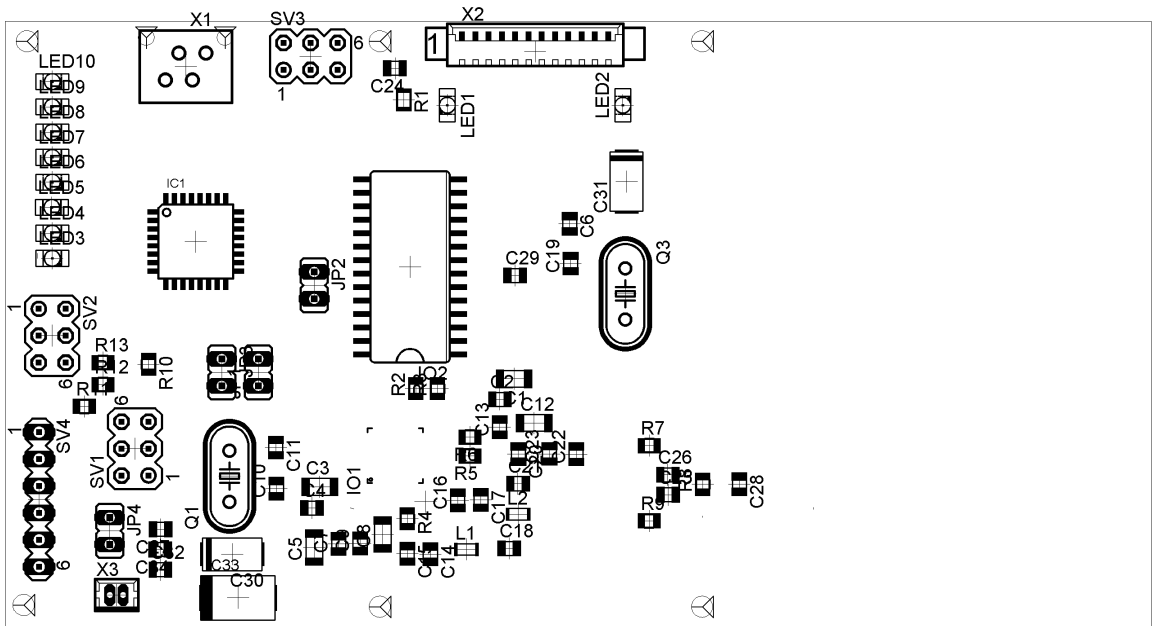
Příloha E: Plošný spoj.



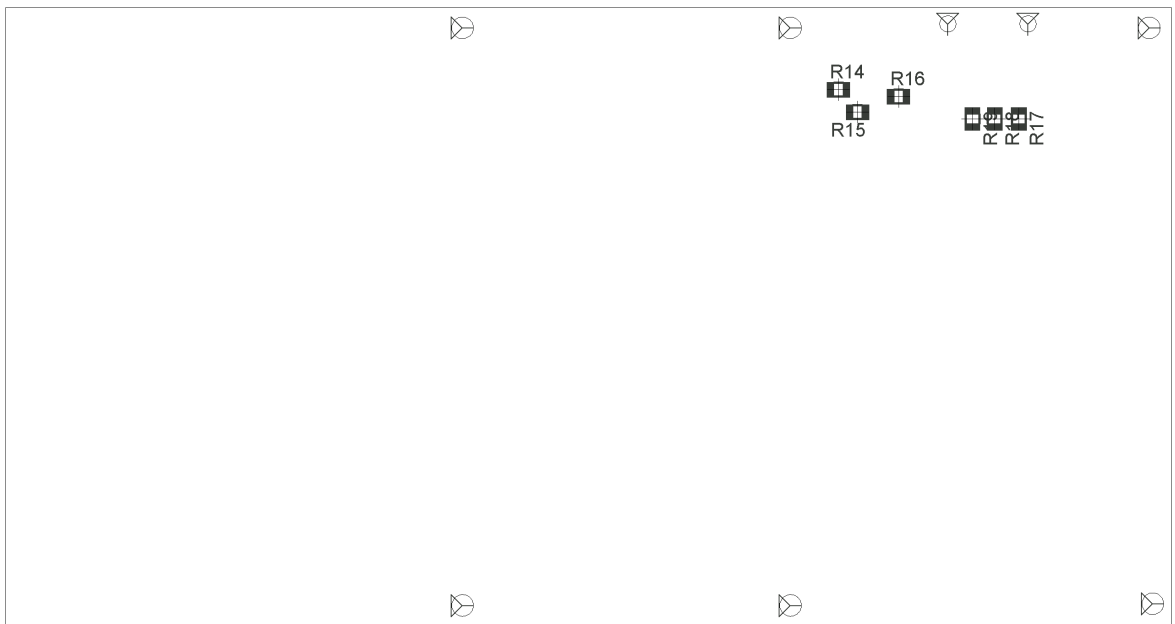
Obr E.1 Plošný spoj – pohled shora.



Obr E.2 Plošný spoj – pohled zdola.



Obr E.3 Rozmístění součástek – pohled shora.



Obr E.4 Rozmístění součástek – pohled zdola.

Příloha F: Seznam součástek.

Part	Value	Device	Package	Library	Sheet
C1	2,2uF	C-EUC1206	C1206	rcl	1
C2	10nF	C-EUC0805	C0805	rcl	1
C3	2,2uF	C-EUC1206	C1206	rcl	1
C4	10nF	C-EUC0805	C0805	rcl	1
C5	2,2uF	C-EUC1206	C1206	rcl	1
C6	25pF	C-EUC0805	C0805	rcl	1
C7	10nF	C-EUC0805	C0805	rcl	1
C8	2,2uF	C-EUC1206	C1206	rcl	1
C9	10nF	C-EUC0805	C0805	rcl	1
C10	22pF	C-EUC0805	C0805	rcl	1
C11	22pF	C-EUC0805	C0805	rcl	1
C12	2,2uF	C-EUC1206	C1206	rcl	1
C13	10nF	C-EUC0805	C0805	rcl	1
C14	1,2nF	C-EUC0805	C0805	rcl	1
C15	1,2nF	C-EUC0805	C0805	rcl	1
C16	1,2nF	C-EUC0805	C0805	rcl	1
C17	1,2nF	C-EUC0805	C0805	rcl	1
C18	680pF	C-EUC0805	C0805	rcl	1
C19	25pF	C-EUC0805	C0805	rcl	1
C20	680pF	C-EUC0805	C0805	rcl	1
C21	220pF	C-EUC0805	C0805	rcl	1
C22	100pF	C-EUC0805	C0805	rcl	1
C23	56pF	C-EUC0805	C0805	rcl	1
C24	100nF	C-EUC0805	C0805	rcl	1
C25	47pF	C-EUC0805	C0805	rcl	1
C26	10pF	C-EUC0805	C0805	rcl	1
C27	100nF	C-EUC0805	C0805	rcl	1
C28	56pF	C-EUC0805	C0805	rcl	1
C29	100nF	C-EUC0805	C0805	rcl	1
C30	15u/16V	CPOL-EUC/6032-28R	C/6032-28R	rcl	1
C31	15u/16V	CPOL-EUC/6032-28R	C/6032-28R	rcl	1
C32	100nF	C-EUC0805	C0805	rcl	1
C33	22u/16 D	CPOL-EUSMCD	SMC_D	rcl	1
C34	100nF	C-EUC0805	C0805	rcl	1
IC1	MEGA8-AI	MEGA8-AI	TQFP32-08	avr	1
IO1	TRF7960	TRF7960	TQFN32	nase3	1
IO2	MAX3110	MAX31102	SO28L	maxim-marek	1
JP1		JP1E	JP1	jumper	1
JP2		JP1E	JP1	jumper	1
JP3		JP1E	JP1	jumper	1
JP4		JP1E	JP1	jumper	1
L1	150nH	L-EUL2012C	L2012C	rcl	1
L2	330nH	L-EUL2012C	L2012C	rcl	1
LED1	Red	LEDSMT1206	1206	led	1
LED2	Green	LEDSMT1206	1206	led	1
LED3	7	LEDSMT1206	1206	led	1
LED4	6	LEDSMT1206	1206	led	1
LED5	5	LEDSMT1206	1206	led	1
LED6	4	LEDSMT1206	1206	led	1
LED7	3	LEDSMT1206	1206	led	1
LED8	2	LEDSMT1206	1206	led	1
LED9	1	LEDSMT1206	1206	led	1
LED10	0	LEDSMT1206	1206	led	1
Q1	13,56MHz	CRYTALHC49U-V	HC49U-V	crystal	1
Q3	1,8432MHz	CRYTALHC49U-V	HC49U-V	crystal	1
R1	1K	R-EU_M0805	M0805	rcl	1
R2	10K	R-EU_R0805	R0805	rcl	1
R3	10K	R-EU_R0805	R0805	rcl	1
R4	0R	R-EU_R0805	R0805	rcl	1

R5	1K	R-EU_R0805	R0805	rcl	1
R6	1K	R-EU_R0805	R0805	rcl	1
R7	0R	R-EU_R0805	R0805	rcl	1
R8	1K	R-EU_R0805	R0805	rcl	1
R9	0R	R-EU_R0805	R0805	rcl	1
R10	4k7	R-EU_R0805	R0805	rcl	1
R11	220R	R-EU_R0805	R0805	rcl	1
R12	1k	R-EU_R0805	R0805	rcl	1
R13	1k	R-EU_R0805	R0805	rcl	1
R14	4k7	R-EU_R0805	R0805	rcl	1
R15	4k7	R-EU_R0805	R0805	rcl	1
R16	4k7	R-EU_R0805	R0805	rcl	1
R17	10k	R-EU_R0805	R0805	rcl	1
R18	10k	R-EU_R0805	R0805	rcl	1
R19	10k	R-EU_R0805	R0805	rcl	1
SV1		MA03-2	MA03-2	con-lstb	1
SV2		MA03-2	MA03-2	con-lstb	1
SV3		MA03-2	MA03-2	con-lstb	1
SV4		MA06-1	MA06-1	con-lstb	1
X1	MICS-4	MICS-4	MICS-4	con-lumberg	1
X2		53261-12	53261-12	con-molex1	1
X3		53047-02	53047-02	con-molex	1

Příloha F: Obsah přiloženého CD

1. Elektronická_podoba_práce
2. Schéma_a_plošný_spoj_v_programu_Eagle
3. Zdrojové_kódy_pro_ATmega8
4. Technické_dokumentace