

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

PRAKTICKÁ REALIZACE PŘÍPRAVKU DEMONSTRUJÍCÍ
SYSTÉM RFID

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

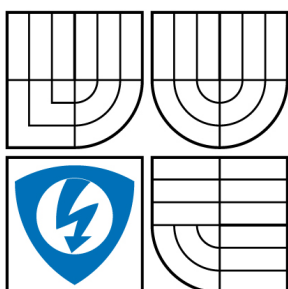
Bc. JOSEF PERNICA

BRNO 2009



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

PRAKTICKÁ REALIZACE PŘÍPRAVKU DEMONSTRUJÍCÍ SYSTÉM RFID

PROTOTYPE REALIZATION FOR RFID SYSTEM DEMONSTRATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

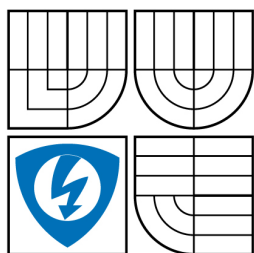
Bc. JOSEF PERNICA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV PFEIFER

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Josef Pernica

ID: 22643

Ročník: 2

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Praktická realizace přípravku demonstrující systém RFID

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s principem činnosti radiofrekvenční (RFID). Navrhněte a realizujte přípravek demonstrující technologii RFID. Ovládání přípravku realizujte pomocí mikroprocesoru Freescale řady HCS08. Ovládací program vytvořte v jazyce C.

DOPORUČENÁ LITERATURA:

- [1] Váňa, V.: Začínáme s mikrokontroléry Motorola HC08 Nitron, BEN, 2003. ISBN 80-7300-124-1.
- [2] Mann, B.: C pro mikrokontroléry, BEN, 2003. ISBN 80-7300-077-6.
- [3] -:125kHz RFID System design Guide: DS 51115, Michrochip Technology Inc.

Termín zadání: 9.2.2009

Termín odevzdání: 26.5.2009

Vedoucí práce: Ing. Václav Pfeifer

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací demonstračního přípravku pro systém RFID. První část se zaměřuje na teorii tohoto systému. Další část se zabývá teorií řešení a použitými součástkami. Dále popisují vlastní návrh přípravku a vytvoření ovládacího programu v jazyce C.

Abstract

This diploma thesis deal with design and realization of demonstration board for RFID system. In the begining is described theory of this system. Next part is about theory of solutions and used components. Next is about practical realization demoboard and control program in C language.

Klíčová slova

RFID, EM4095, mikroprocesor, Freescale, HCS08, anténa, LCD displej, klávesnice, USB, kód Manchester, rezonanční obvod

Key words

RFID, EM4095, microcontroller, Freescale, HCS08, antenna, LCD display, keyboard, USB, Manchester code, resonant circuit

PERNICA, J. *Praktická realizace přípravku demonstrující systém RFID*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 79 s. Vedoucí diplomové práce Ing. Václav Pfeifer.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma “Praktická realizace přípravku demonstrující systém RFID” jsem vypracoval samostatně pod vedením vedoucího diplomové práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Václavu Pfeiferovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

Děkuji také firmě Matron za poskytnutí prostor, přístrojů a programů potřebných k vývoji a fyzické realizaci zařízení, jmenovitě pak Ing. Tronečkovi za jeho cenné odborné rady.

V Brně dne

.....

podpis autora

OBSAH

1. ÚVOD	9
1.1. UPŘESNĚNÍ ZADÁNÍ	9
2. RFID	10
2.1. HISTORIE	10
2.2. RFID TAGY	11
2.2.1. Příklady tagů.....	13
2.3. ČTECÍ ZAŘÍZENÍ.....	16
2.3.1. Výrobci	16
2.3.2. Kód Manchester	18
3. TEORETICKÝ ÚVOD DO PROBLEMATIKY ŘEŠENÍ	19
3.1. POPIS OBVODŮ.....	19
3.1.1. RFID čip EM 4095	19
3.1.2. Mikroprocesor Freescale MC9S08JM32	23
3.1.3. Převodník USB/UART s obvodem FTDI.....	26
3.1.4. LCD displej MC16021E-SYL	30
3.2. PROGRAMOVÁNÍ MIKROPROCESORŮ.....	33
3.2.1. Vývojové prostředí Code Warrior.....	33
3.2.2. Rozhraní BDM.....	33
3.2.3. Programátor	34
4. VYPRACOVÁNÍ	35
4.1. SCHÉMA ZAPOJENÍ	35
4.2. NÁVRH ANTÉNY.....	35
4.3. POPIS ZAPOJENÍ.....	37
4.3.1. RFID.....	37
4.3.2. Řídící část	38
4.3.3. Převodník USB/UART	39
4.3.4. Stabilizace napájení	40
4.3.5. Uživatelská rozhraní	41
4.4. POPIS FIRMWARE.....	44
4.4.1. Popis ovladačů a procedur	45
4.4.2. popis hlavní části programu (main.c).....	54
4.4.3. Popis menu	57
4.5. POPIS SOFTWARE	57
4.6. PŘÍPRAVEK.....	58
4.6.1. Měření průběhů signálů.....	58
5. ZÁVĚR	62
6. POUŽITÁ LITERATURA	63
7. SEZNAM TABULEK, OBRÁZKŮ, POUŽITÉ SYMBOLY	64
7.1. SEZNAM TABULEK	64

7.2.	SEZNAM OBRÁZKŮ.....	64
7.3.	POUŽITÉ SYMBOLY	64
8.	SEZNAM PŘÍLOH.....	66

1. Úvod

1.1. Upřesnění zadání

Tato práce pojednává o základních principech RFID a zabývá se praktickou realizací čtecího zařízení.

Práce bude koncipovaná do několika částí. V první části bude rozebrána teoretická problematika RFID. Seznámení se se základním principem radiofrekvenční komunikace a její využití v praxi. Dále pak s obvody a příslušenstvím nezbytným pro funkčnost RFID komunikace.

V druhé části práce budou probrány možnosti praktické realizace RFID zařízení. Shrnu získané poznatky, jejich výhody a nevýhody pro zadané téma a vyberu nejvhodnější řešení.

Za cíl jsem si dal navrhnout přípravek podle těchto kritérií:

- základní deska s mikroprocesorem HCS08 s uživatelským rozhraním (klávesnice, LCD displej, volné porty pro komunikaci s okolními obvody)
- modul s integrovaným obvodem RFID připojitelný k základní procesorové desce. Modul by měl umožňovat komunikaci s transpondéry, čtení a popř. i zápis
- demonstrace komunikace s různými typy RFID tagů.

2. RFID

Radiofrekvenční Identifikace RFID [1] je automatická identifikační metoda založená na uchování dat a vzdáleném přístupu k datům na zařízení nazvaném RFID tag nebo transpondér.

RFID tag může být umístěn vně nebo uvnitř předmětu, zvířata či osoby za účelem identifikace a sledování použitím rádiových vln. Některé tagy mohou být čteny až na vzdálenost několika metrů a mimo přímý dohled čtečky.

Většina tagů se skládá ze dvou hlavních komponent:

- integrovaný obvod pro uložení a přenos informace, modulátor, demodulátor...
- anténa pro příjem a vysílání signálu

Ve vývoji jsou bezčipové tagy podporující identifikaci bez integrovaného obvodu, což by umožňovalo tisknout tagy přímo na různé objekty levněji než klasické čipy.

RFID je v dnešní hojně využívána ve skladovém hospodářství pro zvýšení efektivity správy a vyhledávání.

2.1. Historie

V roce 1946 vynalezl Léon Theremin špionážní nástroj pro Sovětský Svaz, který přeposílal nahodilé rádiové vlny se zvukovou informací. Zvukové vlny rozechvívaly membránu tvořící rezonátor, který moduloval odražené rádiové frekvence. Přestože toto zařízení bylo pasivní odposlouchávací zařízení, ne identifikační tag, bylo klasifikováno jako předchůdce RFID.

Podobná technologie IFF transpondér byla objevena v Anglii v roce 1939 a běžně používána spojenci ve Druhé Světové Válce k identifikaci letadel. Transpondéry jsou dodnes využívány v moderních letadlech.

Prvním opravdovým předchůdcem RFID byl v roce 1973 Mario Cardullo's, který vynalezl pasivní transpondér s pamětí. Výchozí zařízení bylo pasivní, napájené dotazovacím signálem a demonstrováno v roce 1971 s potenciálním využitím

16bitové paměti pro mýtné zařízení. Jeho základní patent obsahoval využití RF, zvuku a světla jako přenosového média. Původní podnikatelský plán, prezentovaný investorům v roce 1971, předpokládal využití v dopravě (identifikace vozidel, automatické mýtné systémy, elektronické licence, určování historie pohybu vozidel...), bankovníctví (elektronické šekové knížky, elektronické platební karty...), zabezpečovací zařízení (osobní identifikace, automatické brány), medicína (identifikace, karty pacientů).

2.2. RFID tagy

RFID tagy můžeme rozdělit do tří hlavních skupin: pasivní, aktivní a polopasivní (neboli s pomocným bateriovým napájením nebo také poloaktivní). Pasivní tagy nevyžadují napájecí zdroj, a tedy jsou čistě pasivní zařízení (jsou aktivní jen tehdy pokud je v okolí čtečka, která je bezdrátově napájí), zatímco polopasivní a aktivní tagy vyžadují napájecí zdroj, obvykle malou baterii.

Pasivní

Pasivní RFID tagy nemají žádný vnitřní zdroj napájení. Pokud se k tagu přiblíží čtečka, která vysílá signál, naindukuje se na anténě potřebné napájení pro CMOS obvody. Toho lze využít nejen pro vyslání signálu ale také pro uložení dat do paměti EEPROM. Nemusí to být jen číslo (ID čipu) ale jakákoliv data.

Pasivní tagy mají dosah 11 cm až 183 metrů při využití sfázované anténní soustavy. V zásadě čtení a zápis závisí na frekvenci a typu antény. Vzhledem k jednoduchosti návrhu je možné antény tisknout. Využitím toho, že nepotřebujeme napájecí zdroj, získáme velmi malé rozměry zařízení. Komerčně dostupné produkty existují jako zapuštěné ve štítku, nebo implantované pod kůží.

Aktivní

Na rozdíl od pasivních tagů mají vlastní vnitřní zdroj napájení, který napájí integrované obvody, které komunikují se čtečkou. Komunikace se čtečkou je více spolehlivá. Díky vlastnímu zdroji mohou vysílat větší úroveň signálu, což pomáhá při

vysílání ve zhoršených podmínkách, při odrazech od kovů a zajišťuje větší dosah signálu. Životnost tagu může být až roky.

Dnešní tagy mají operační vzdálenost stovky metrů, baterie jim vydrží měsíce až roky a obsahují mnohem více paměti než pasivní tagy. Speciální tagy mohou obsahovat různé speciální senzory, např. teplotní čidla, vibrační čidla, radiační čidla atd.

Polopasivní tagy

Podobně jako aktivní tagy obsahují vlastní napájecí zdroj, který slouží ke zvýšení dosahu snímání. Přídavnou funkcí zdroje je napájení paměti.

Beacon (maják) tagy

Majákové tagy vysílají svoji identitu podle určité předlohy. Mohou vysílat v pravidelných intervalech, v intervalech podle klíče nebo ve spouštěných intervalech.

Typy antén

Antény se dělí dle aplikací a použitých frekvencí.

- nízkofrekvenční (LFID) – 30~300 kHz – pasivní tagy obsahují indukční obvody a protože indukované napětí je úměrné frekvenci musí cívka obsahovat větší počet závitů k zajištění dostatečného napětí k napájení integrovaných obvodů. Kompaktní tagy, např. zapouzdřené ve skle, využívané pro identifikaci zvířat, obsahují 3vstvy po 150 otáčkách na feritovém jádru. Vlastnosti tagů s touto anténou: dosah pod 0.5m, malá rychlost čtení, vysoké výrobní náklady, možnost snímání na kovu a přes kapalinu.

- vysokofrekvenční (HFID) – 3~30 MHz - využívají planární spirály s 5-7 otáčkami velikosti kreditní karty. Tento způsob je méně finančně náročný než předcházející díky využití litografických technik, ale je nezbytné využít 2 metalické desky a izolační vrstvu abychom připojili vnější část spirály k obvodům umístěným

uvnitř spirály. Vlastnosti tagů s touto anténou: dosah do 1 m, dostatečná rychlost čtení, vysoké výrobní náklady, obtížné čtení přes kapalinu.

- mikrovlnné (UHFID) – 300 MHz~6 GHz – tagy pro využití velmi vysokých frekvencí a mikrovln využívají antény vypadající jako klasický dipól. Při realizaci antény je využita pouze jedna kovová vrstva což snižuje výrobní náklady. Tyto antény jsou však velmi rozměrné a pro použití do malých tagů nevhodné, proto se buď skládají nebo stáčíjí do menších rozměrů. Vlastnosti tagů s touto anténou: dosah do 10m, velmi vysoká rychlost, vysoká cena tagů.

Kódování a zabezpečení

Každý tag má výrobcem přednastaven tzv. unikátní 64 -128 bitový kód. Některé mají antikolizní systém, který zabraňuje kolizi vysílaných dat od více tagů současně na jednu přijímací anténu. Zapisovatelné tagy mají pro přenos uživatelských informací kryptovací algoritmy, případně je ochrana dat zabezpečena heslem.

Rozdělení tagů dle tříd

- Class 0 – pouze pro čtení, programováno ve výrobě, 64 nebo 96bit, čtení 1000 tagů za sekundu
- Class 1 – zápis jednou/zápis mnohokrát, programováno při použití, 64 nebo 96bit, čtení 200 tagů za sekundu
- Class 0+ - čtení/zápis, programováno kdykoliv, 256bit, čtení 1000 tagů za sekundu
- Gen 2 – čtení/zápis, programováno kdykoliv, 256bit, čtení 1600 tagů za sekundu

2.2.1. Příklady tagů

Klíčenka “Teardrop” [2] (viz obr. 2.1) – přívěsek na klíče, ve kterém je integrován RFID čip, vyrábí se z ABS vysoce odolného materiálu v různých barvách. Provozní teplota od -25°C do 50 °C, vhodné pro vnitřní i vnější použití, odolné prachu, špíně a vodě.



obrázek 2.1.: klíčenky “Teardrop”.

Klíčenka “Sail” [2] (viz obr. 2.2) – odolnější než předchozí typ, vyrobené z polykarbonátu, provozní teplota od -40°C do 85°C .



obrázek 2.2.: klíčenky “Sail”.

Hodinky [2] (viz obr. 2.3) – přenosný transpondér, který může sloužit také jako elektronická peněženka. Vyrábí se z antialergického materiálu, voděodolné.



obrázek 2.3.: hodinky.

Bezkontaktní karty [2] (viz obr. 2.4) – vyrábí se ve standardním formátu ISO, z PVC materiálu lesklého či matného, lze na ně tisknout, kombinovat s magnetickým proužkem, provozní teplota od -20°C do 50°C .



obrázek 2.4.: bezkontaktní karty.

Samolepící etikety – vyrábí se v různých barvách a velikostech, s různými druhy lepidla...

Teoreticky lze vyrobit tag jakéhokoliv tvaru, velikosti a barvy (viz obr. 2.5)



obrázek 2.5.: různé možnosti vizualizace tagů.

2.3. Čtecí zařízení

RFID čtecí zařízení se používá k aktivaci pasivních tagů bezdrátovou energií a k získávání informací z tagů. Srdcem těchto čtecích zařízení je IO, který zajišťuje přenos informací do/z antény a zpracování signálu.

2.3.1. Výrobci

Atmel: U2270B

- frekvence nosné 100-150kHz
- přenosová rychlost až 5kBd
- možnost napájení z autobaterie (pro aplikace imobilizéru)
- optimalizováno pro imobilizér do automobilu
- rozhraní kompatibilní s mikrokontroléry
- nízký odběr v pohotovostním režimu
- pouzdro SO-16

Texas instrument: RI-TMS3705ADR

- nízký odběr v režimu spánku
- automatické měření frekvence transpondéru a adaptace
- vlastní diagnostika
- digitální FSK demodulátor
- vysokorychlostní přenos dat v synchronním módu
- fázový závěs
- ochrana proti zkratu
- pouzdro SO-16

Melexis: MLX90109

- integrovaný RFID přijímač
- rozsah frekvencí 100 až 150kHz
- koncept paralelní antény pro maximální výkon
- nastavitelná přenosová rychlost
- nevyžaduje externí zdroj hodin
- komunikace po dvou drátech

ID MOS: IM283

- nízká spotřeba
- čtení i zápis
- podporuje většinu 125kHz tagů
- rozsah frekvencí 100-150kHz
- výkonný ovladač antény
- nastavitelná přenosová rychlost
- pouzdro SOIC18

EM Microelectronic: EM 4095

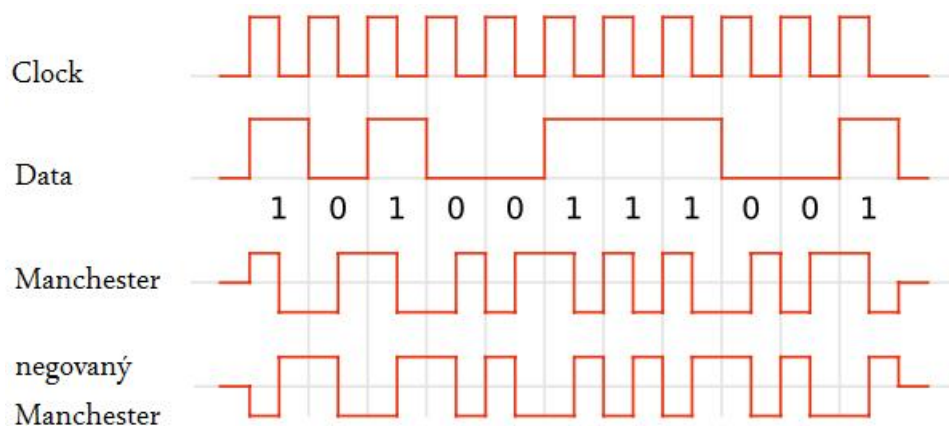
- integrovaný PLL systém k zajištění nastavení rezonanční frekvence antény
- nevyžaduje externí hodiny
- rozpětí nosné 100~150 kHz
- výkonový výstup pro přímé řízení antény
- vysílání dat pomocí OOK (100% AM)
- možnost externího nastavení hloubky modulace AM

- sleep mód $1\mu\text{A}$
- kompatibilní s napájením přes USB
- pracovní teplota -40 až 85°C
- malé pouzdro SO16

V dnešní době je již možno zakoupit procesor se zabudovanými obvody pro RFID, ale pro moji práci, je toto nevhodné, vzhledem k potřebě demonstrace a univerzállosti přípravku.

2.3.2. Kód Manchester

Jedná se o linkový kód, kde pro vyjádření bitu se do poloviny bitového intervalu původního signálu vloží hrana – změna signálu [4]. Pokud signál v této hraně přechází z vysoké úrovně na nízkou úroveň, pak vyjadřuje hrana hodnotu bitu 1. Pokud signál přechází z nízké úrovně na vysokou úroveň, hodnota bitu bude 0. Protože se hrana vždy nachází uprostřed každého bitového intervalu, může snadno sloužit k synchronizaci (viz obr. 2.6).



obrázek 2.6.: kód Manchester

3. Teoretický úvod do problematiky řešení

3.1. popis obvodů

3.1.1. RFID čip EM 4095

Jedná se o CMOS integrovaný přijímací obvod [4] určený pro RFID zařízení k zajištění následujících funkcí:

- řízení antény nosnou frekvencí
- AM modulace signálu pro vysílání
- AM demodulace signálu indukovaného na anténě
- komunikace s mikroprocesorem po jednoduchém rozhraní

Základní vlastnosti:

- integrovaný PLL systém k zajištění nastavení rezonanční frekvence antény
- nevyžaduje externí hodiny
- rozpětí nosné 100~150 kHz
- výkonový výstup pro přímé řízení antény
- vysílání dat pomocí OOK (100% AM)
- možnost externího nastavení hloubky modulace AM
- sleep mód 1 μ A
- kompatibilní s napájením přes USB
- pracovní teplota -40 až 85°C
- malé pouzdro SO16 (viz obr. 3.2)

Tento integrovaný obvod obsahuje ochranu proti statické elektřině. Pracuje správně pouze je-li napájecí napětí v požadovaném rozsahu (viz Tabulka 1). Nevyužité vstupy musí být připojeny na definovanou logickou úroveň.

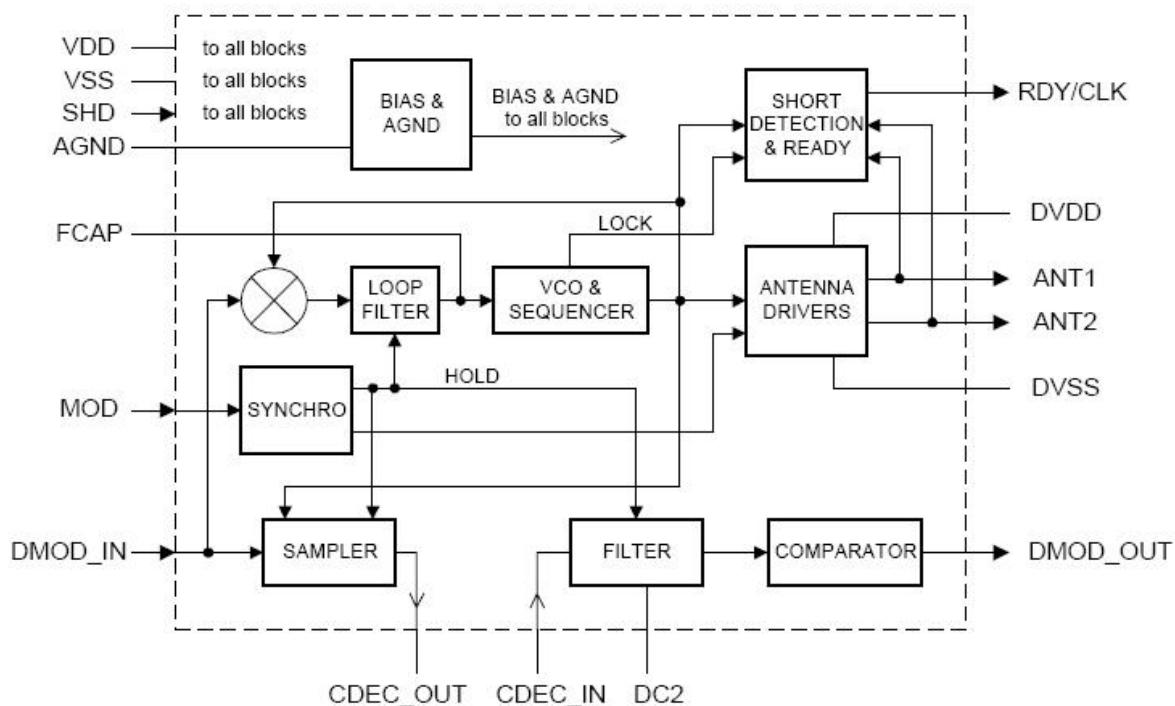
Parametr	symbol	min	typ	max	jednotka
operační bodová teplota	T_J	-40		110	°C
napájecí napětí	V_{DD}	4,1	5	5,5	V
rezonanční frekvence antény	F_{RES}	100	125	150	kHz
špičkový proud anténních vstupů	I_{ANT}			250	mA
C_{FCAP}		*	10	*	nF
C_{DEC}		*	100	*	nF
C_{DC2}		*	6,8	*	nF
C_{AGND}		100		220	nF

* $\pm 10\%$ tolerance kondenzátorů

** uvedené hodnoty C jsou pro doporučené zapojení uvedené výrobcem

Tabulka 1.: provozní podmínky.

Blokové schéma a popis IO



obrázek 3.1.: blokové schéma IO EM 4095.

Popis obvodu

Základní vlastnosti – obvod je určen k použití společně s anténou a mikroprocesorem. Je vyžadováno stabilizované napájení. Funkce obvodu je řízena logickými vstupy SHD a MOD (viz obr. 3.2).

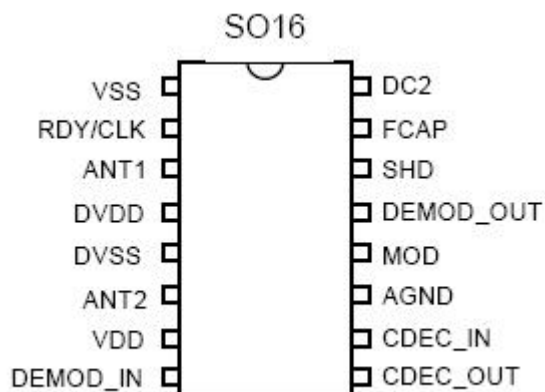
Analogová část – obvody zajišťují dvě funkce: vysílání a příjem. Příjem zahrnuje AM demodulaci přijatého signálu z transpondéru. Vysílání je zajištěno anténou a PLL obvodem.

Výkonový výstup – ovládání antény zajišťuje čtecímu zařízení signál požadované úrovně. Rezonanční frekvence antény je obvykle 125 kHz. Dodávaný proud do antény závisí na kvalitě Q rezonančního obvodu. Je vyžadováno, aby nebyl překročen maximální špičkový proud 250 mA. Dále nesmí být překročena maximální bodová teplota pouzdra (viz. Tabulka 1). Anténa je opatřena ochranou proti zkratu napájecího napětí a při indikaci tohoto stavu se obvod zablokuje.

PLL (Phase Locked Loop) – neboli smyčka fázového závěsu. Využívá se při generaci přesného signálu. Vlastní generování má na starosti napětím laděný oscilátor VCO (viz obr. 3.2), ten je propojen s PLL, který kontroluje frekvenci a podle výsledku doladuje oscilátor. Přesnou referenční frekvenci určuje krystal. Frekvence oscilátoru je digitálními obvody postupně dělena a podobně i frekvence krystalu. Výsledné frekvence se porovnají a produktem tohoto porovnávání je ladící napětí.

Příjem – Modulovaný signál z antény prochází kapacitním děličem na vstup DMOD_IN do vzorkovače (viz obr. 3.2). Kapacitní dělič snižuje napěťovou úroveň signálu z antény na požadovanou hodnotu napětí tj. maximálně VDD-0,5 V.

Popis pouzdra (viz Tabulka 2)



obrázek 3.2.: pouzdro IO EM4095.

pin	název	popis	typ
1	VSS	záporný pól napájení obvodu	GND
2	RDY/CLK	výstup jako vlajkový registr a hodiny	O
3	ANT1	ovládání antény	O
4	DVDD	kladný pól napájení pro anténu	PWR
5	DVSS	záporný pól pro napájení antény	GND
6	ANT2	anténní výstup	O
7	VDD	kladný pól napájení obvodu	PWR
8	DEMOD_IN		ANA
9	CDEC_OUT	pin pro blokový kondenzátor	ANA
10	CDEC_IN	pin pro blokový kondenzátor	ANA
11	AGND	analogová zem	ANA
12	MOD		IPD
13	DEMOD_OUT		O
14	SHD	uvedení do sleep módu	IPU
15	FCAP	filtrační kondenzátor pro PLL	ANA
16	DC2	kondenzátor pro odstranění nosné	ANA

GND: referenční zem PWR: napájení ANA: analogový signál

IPD: vstup s integrovaným pull-up rezistorem O: výstup

IPU: výstup s integrovaným pull-up rezistorem

Tabulka 2.: popis pouzdra.

Typická konfigurace (viz obr. 4.3)

Hlavní obory využití

- logistika a skladování

- řízení skladu
- řízení výroby
- evidence majetku
- retail
- zdravotnictví
- knihovny
- odpadové hospodářství

3.1.2. Mikroprocesor Freescale MC9S08JM32

Jako řídicí mikroprocesor [5], který bude zajišťovat komunikaci s RFID obvodem, PC a ostatními periferiemi jsem vybral mikroprocesor MC9S08JM32 s jádrem rodiny HCS08. Jedná se o verzi v pouzdře LQFP se 64mi piny. Hlavní kritérium výběru mikroprocesoru byl Timer Modul (TPM) obsahující funkci Input Capture. Z rodiny HCS08 tuto funkci obsahuje jen několik mikroprocesorů. Funkce je využita pro dekódování dat v Manchester kódu z RFID obvodu.

Základní údaje o mikroprocesoru MC9S08JM32:

8-Bit HCS08 Centrální procesorová jednotka (CPU)

- HC08 instrukční sada s podporou BGND
- 48 MHz HCS08 (24 MHz frekvence vnitřní sběrnice)
- integrované ladící rozhraní BDM

paměti na čipu (Flash, RAM)

- 32kB paměti Flash s ochranou bloků proti přepisu, mazání a odcizení jejího obsahu
- 2kB paměti RAM pro uživatelské proměnné

interní zdroj hodin (MCG)

- frekvence sběrnice až do 24MHz
- obvod Frequency Locked Loop (FLL) s volitelnými násobiči
- integrovaný generátor hodin s typickou frekvencí 32,7 kHz
- dělička hodin pro sběrnici
- samostatný zdroj hodin pro přerušení v reálném čase (RTI)

A/D převodník (ADC)

- 12-kanálový, 12-bitový A/D převodník

rozhraní sériové komunikace (IIC, SPI, 2x SCI)

- IIC (synchronní), SPI (synchronní), 2xSCI (asynchronní)

rozhraní USB

- plně rychlostní USB 2.0 s přenosovou rychlostí až 12Mb/s. Čip mikroprocesoru obsahuje USB vysílač/přijímač, 3,3 V regulátor napětí, pull-up rezistory a pod.

modul přerušení od klávesnice (KBI)

- 8-pinový modul pro přerušení od klávesnice

časovač/čítač (TPM)

- 2-kanálový a 6-kanálový 16-bitový čítač/pulsně - šířkový modulátor podporující módy činnosti: input capture, output compare, edge-aligned PWM and centre-aligned PWM

vstupy / výstupy (I/O)

- 51-pinů s možností aktivace integrovaných „pull-up“ rezistorů, volby proudové zatížitelnosti a omezení rychlosti změn náběžných a sestupných hran generovaných výstupních signálů

vývojové moduly a nástroje (ICE, BDC)

- Modul In-Circuit Emulation (ICE) je integrován na čipu. Tento ladící modul má výkonné krokovací a trasovací jednotky, které umožní komfortní ladění a monitorování chodu programu.
- Background Debug Controller (BDC) je integrovaný také na čipu a obsahuje jednovodičovou sběrnici, pomocí níž se program v procesoru nahrává, debuguje apod.

řada pouzder

- 64 QFP, 64 LQFP, 48 QFN, 44 LQFP

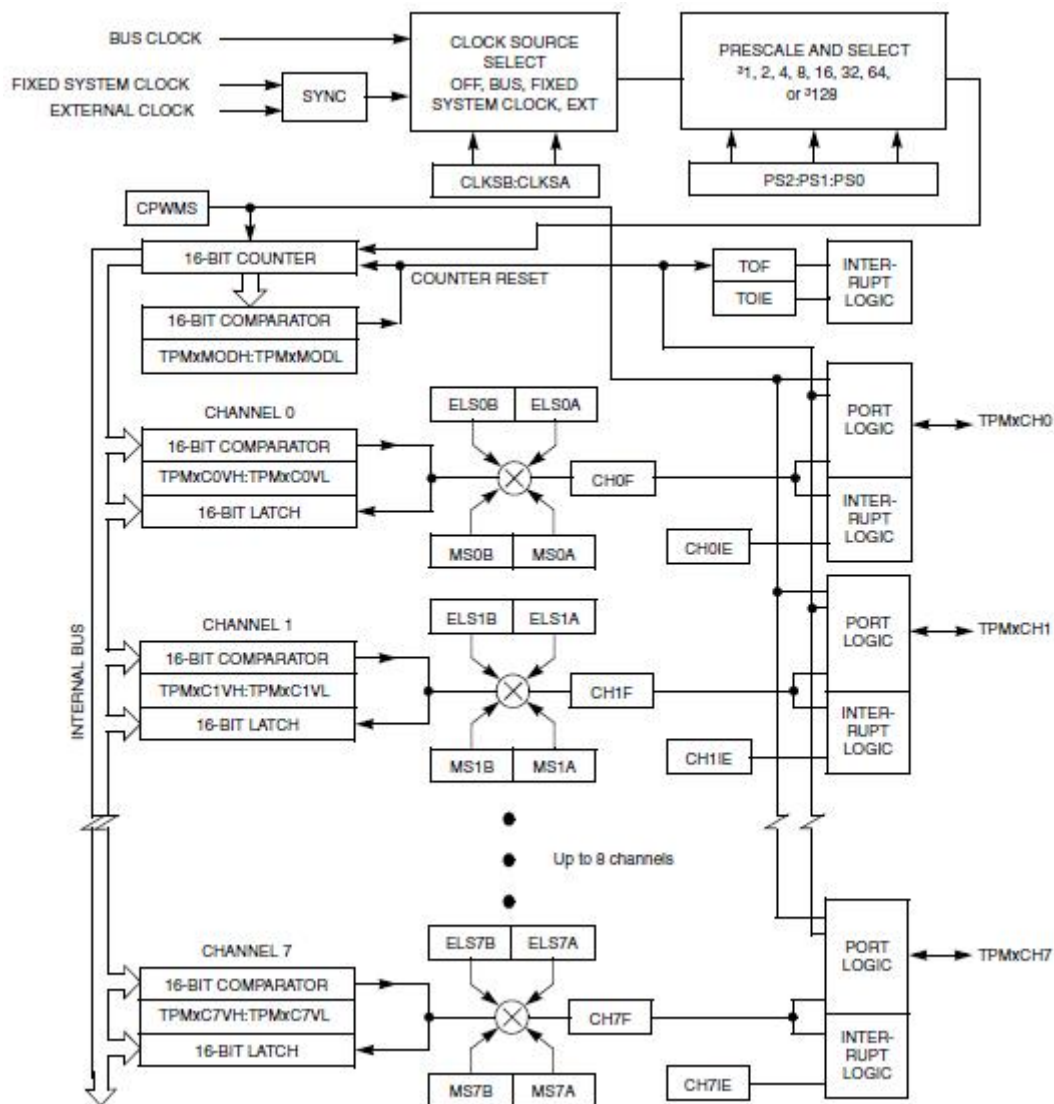
Blokové schéma mikroprocesoru je příloze 4.

Blok Input Capture a Output Compare

Blok volnoběžného 16-bitového čítače využívá externího vstupu TPMCLK (viz obr. 3.9). Při použití externího vstupu čítání je nutné pamatovat na skutečnost, že tento kmitočet je před zpracováním v interním volně běžícím čítači synchronizován s řídicím kmitočtem (bus clock). Jelikož je změna na externím vstupu TPMCLK vzorkována vždy za jednu periodu řídicího kmitočtu, je maximální kmitočet externího signálu nejvýše roven polovině řídicího kmitočtu (bus clock).

Jednotka Input capture je schopna zachytit do šestnáctibitového registru (registry TPMxCnVH a TPMxCnVL) stav čítače při výskytu hrany na řídicím vývodu. Řídicí vývody pro Input Capture funkci lze zvolit mezi TPM1CH0, TPM1CH1, TPM1CH2, TPM1CH3, TPM1CH4, TPM1CH5, TPM2CH0, TPM2CH1. Viz blokové schéma na obrázku 3.3. Přičemž aktivní hranou může být buď přechod z úrovně L do H, z úrovně H do L, nebo se aktivní hranou rozumí každá změna na řídicím vstupu.. Při výskytu aktivní hrany na řídicím vstupu je možné taktéž generovat přerušení.

Další funkce je porovnávání uložené 16-bitové hodnoty s hodnotou volně běžícího čítače. Při dosažení shody hodnot je pak možné příslušný výstupní vývod vynulovat nebo nastavit, nebo je možné invertovat jeho stav. Při dosažení shody je též možné generovat přerušení.



obrázek 3.3.: blokové schéma timeru

Další funkce které jednotka může vykonávat, je generování PWM signálu.

3.1.3. Převodník USB/UART s obvodem FTDI

Pro připojení zařízení k PC bylo zvoleno rozhraní USB. Bylo jej možné realizovat pomocí již implementovaného rozhraní USB v mikroprocesoru MC9S08JM32 a nebo externího integrovaného obvodu firmy FTDI připojenému k mikroprocesoru. Pro rychlou a úspěšnou realizaci bylo vybráno řešení s externím obvodem. Důvody byly,

dobrá dostupnost a jednoduché použití ovladačů v systému Windows pro program pod Windows.

Obvod FT232R od firmy FTDI je převodník USB-UART, který pro svoji funkci potřebuje minimum externích součástek. K mikroprocesoru se zapojuje pomocí linky standartu UART.

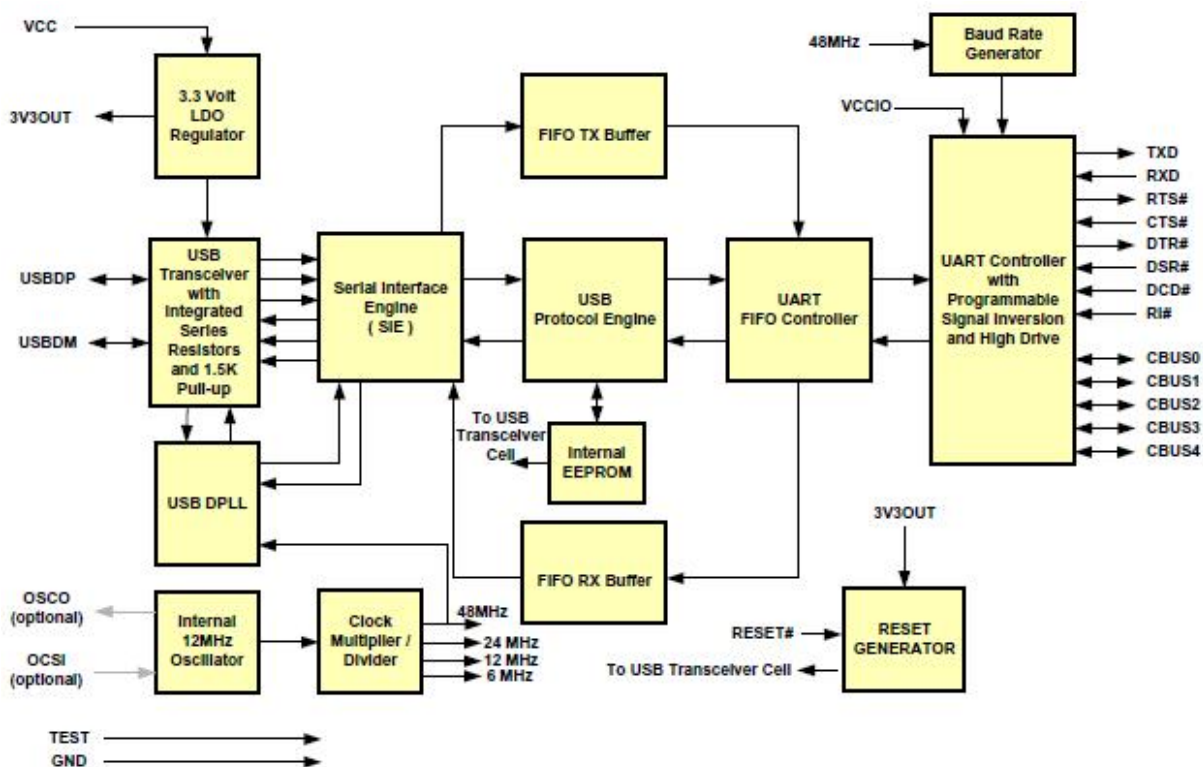
Hlavní vlastnosti obvodu lze shrnout v několika stručných bodech:

- USB 2.0 s přenosovou rychlostí 300Bd až 3MBd (nastavení přenosu 7 nebo 8 datových bitů, 1 nebo 2 stop bity, parita apod.)
- registr FIFO pro příjem a vysílání
- podpora řízení přenosu pomocí HW i SW
- výrobce podporuje všechny běžné operační systémy pomocí vlastních driverů
- možnost synchronního i asynchronního režimu
- podpora úsporných režimů (sleep mód apod.) a probouzení PC
- integrovaná vnitřní EEPROM 1024bit, pro uložení USB VID, PID, sériového čísla, popisu zařízení, a nastavení funkce vstupně/výstupních linek CBUS I/O.
- integrovaný oscilátor na čipu, bez vnějšího krystalu
- podpora logiky 1,8 V, 2,8 V, 3,3 V, 5 V CMOS.
- programovatelná polarita signálu na lince UART
- možnost napájet jak obvod FT232R, tak i připojené zařízení pomocí napájecího napětí 5 V přímo z portu USB.
- pouzdro QFN-32 nebo SSOP28

Výrobce v operačních systémech podporuje obvod pomocí dvou typů driverů.

Drivery VCP (Virtual Com Port), které v systému počítače vytvoří virtuální COM port, na který se potom přistupuje stejně jako ke všem ostatním portům COM přes Windows API. Existují verze s podporou Plug & Play (PNP) a bez ní. Druhou alternativou driverů jsou drivery D2XX. V případě těchto ovladačů se v systému počítače vytvoří přímo další „USB Devices“, na který se přistupuje přímo pomocí dll knihovny. Tato varianta skýtá větší možnosti v ovládání a nastavení USB obvodu než drivery VCP. V našem případě je použito v obslužném programu ve Windows driverů D2XX.

Na obrázku 3.4. Je vidět blokové schéma obvodu FT232R.



obrázek 3.7.: blokové schéma FTDI

Stručný popis jednotlivých bloků:

3.3V LDO Regulator

Integrovaný 3,3 V stabilizátor pro napájení výstupních bufferů v logice 3,3 V.

USB Transceiver

Přijímač / vysílač pro přímé spojení s kabelem USB. Obsahuje sériové rezistory na datových vodičích USB.

USB DPLL

Rozděluje příchozí data typu NRZI z USB na hodinový signál a data pro zpracování v bloku SIE.

Internal 12MHz Oscillator

Generuje 12MHz hodinový signál pro bloky SIE, USB Protocol Engine a UART FIFO controllers blocks. Umožňuje také připojení externího krystalu.

Clock Multiplier / Divider

Dělička / násobička generuje z referenční frekvence 12MHz, frekvence 6MHz, 12MHz, 24MHz, 48MHz určených pro bloky USB DPLL a Baud Rate Generator Block.

Serial Interface Engine (SIE)

Konverze sériových dat na paralelní a obráceně. Kontrola parity apod.

USB Protocol Engine

Komunikace na nejnižší úrovni pomocí USB protokolu.

FIFO TX Buffer

Záchytný registr pro data přicházející z USB. Dále jsou zpracovávána blokem UART FIFO controller a vysílána na UART.

FIFO RX Buffer

Záchytný registr pro data přicházející z UART. Dále jsou zpracovávána blokem UART FIFO controller a vysílána na USB.

UART FIFO Controller

Předává data mezi FIFO RX buferem, FIFO TX buferem a přijímacím/ vysílacím registrem UART

UART Controller with Programmable Signal Inversion and High Drive

Vytváří řídicí signály RTS, CTS, DSR , DTR, DCD and RI na UART. Provádí programovatelný režim změny polaroty signálu na lince UART.

Baud Rate Generator

Programovatelný generátor vytvářející baudovou rychlost od 183 Baudů do 3 milionů Baudů. Lze vytvářet i nestandardní baudové rychlosti.

RESET Generator

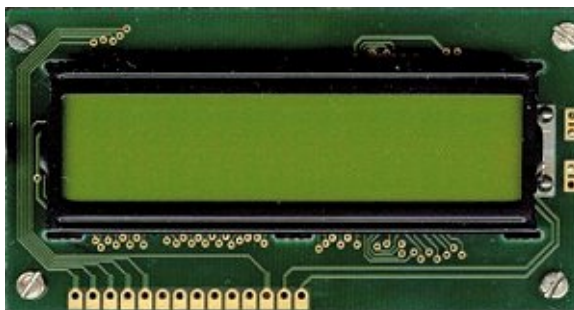
Vstup označen jako RESET# lze zapojit k externímu obvodu vytvářející reset obvodu FTDI.

Internal EEPROM

Slouží k uložení tzv. USB Vendor ID (VID), Product ID (PID), Device Serial Number, Product Description String a další informace pro nastavení obvodu FTDI.

3.1.4. LCD displej MC16021E-SYL

Jedná se o displej od firmy GMe [3] ve variantě 16x2 tzn. 16 znaků ve dvou řádcích (viz obr. 3.5). Obsahuje speciální obvod – řadič, který ovládá celý displej a komunikuje s okolím. Standardem je řadič HD44780 od firmy HITACHI nebo jeho ekvivalent.



obrázek 3.5.: celkový pohled na displej

Základní vlastnosti:

- formát displeje: 16 znaků x 2 řádky
- velikost znaku 2.95(šířka) x 5.55(výška) mm
- zobrazovací oblast: 62.2(šířka) x 17.9(výška) mm
- celkový rozměr: 84(šířka) x 44(výška) x 14.5(tloušťka) mm
- typ LCD: STN žlutý
- typ podsvícení: LED
- barva podsvícení: zelenožlutá
- teplotní rozsah: normální: provozní: 0 - 50°C
skladovací: -20 - 70°C

Zapojení vývodů je uvedeno v tabulce 3 včetně názvu signálů a popisu.

Vývod	název	Popis
1	V _{SS}	napájení GND
2	V _{DD}	napájení +5V
3	V ₀	kontrast 0V .. 5V (typ. 0,8V)
4	RS	Register Select (0=instrukce, 1=data)
5	R/W	Read / Write (0=zápis, 1=čtení)
6	E	Enable (1, hrana z 1 -> do 0)
7	DB0	Data Bus 0
8	DB1	Data Bus 1
9	DB2	Data Bus 2
10	DB3	Data Bus 3
11	DB4	Data Bus 4
12	DB5	Data Bus 5
13	DB6	Data Bus 6
14	DB7	Data Bus 7
15	A	podsvícení - anoda LED
16	K	podsvícení – katoda LED

Tabulka 3.: zapojení vývodů

*DB0 - DB3 (Data Bus), nižší bity datové sběrnice. Nevyužito při 4-bitovém ovládní.

*DB4 - DB7 (Data Bus), vyšší bity datové sběrnice

Komunikace s displejem

Pro komunikaci s displejem je potřeba minimálně 6 vodičů až maximálně 11 vodičů a napájení +5.0 V. Před připojením k procesoru musíme zvolit datovou komunikaci. Komunikace s modulem může být po 8-bitové sběrnici (DB0-DB7) nebo 4-bitové sběrnici (DB4-DB7).

Při 8-bitové komunikaci se vyšle všech 8 bitů na DB0-DB7 a zapíše do displeje signálem Enable. 4-bitová komunikace (DB4-DB7) je vhodná při nedostatku V/V bran procesoru. Komunikace je 2x pomalejší, protože se data vysílají nadvrát. Nejdříve se vyšlou vyšší 4 bity a zapíšou signálem Enable a pak nižší 4 bity a opět zapíšou signálem Enable.

Zobrazení znaků

Každý znak je zobrazován jako matice 5x8 bodů. Definice jednotlivých znaků je uložena napevno ve vnitřní paměti ROM (viz Příloha 1). Dále je možno zobrazit kurzor, nastavit blikání kurzoru, definovat posouvání zobrazených znaků, smazat displej, zobrazit konkrétní pozici apod. Data která se mají zobrazit jsou uložena v paměti DDRAM (Display Data Random Acces Memory). Každý zobrazovaný znak má svoji adresu. Paměť má pro každý řádek 40 pozic ale je zobrazováno pouze 16 aktuálních (viz obr. 3.6).



obrázek 3.8.: paměť DDRAM

Prvních 8 znaků (0-7) v paměti ROM je možno nadefinovat uživatelem, třeba jako české znaky, nebo grafické symboly. Pro češtinu je 8 znaků málo, proto se znaky

můžou měnit v generátoru znaků před jejich zobrazením. Velikost znaku, které zapisujeme do paměti je 8x8 bodů, ale zobrazí se jen 5x8 nebo 5x10 podle zvoleného nastavení displeje při inicializaci.

Tabulka instrukcí (viz příloha 2)

3.2. programování mikroprocesorů

3.2.1. Vývojové prostředí Code Warrior

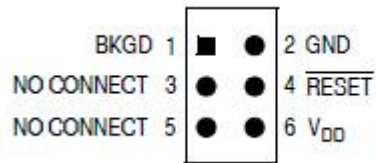
Vývojové prostředí dodávané přímo společností Freescale, které umožňuje vývoj programů jak v assembleru, tak i v jazyce C (C++), obsahuje kompilátor, ladící nástroje a simulátor. Integrovaný nástroj Procesor Expert nejen že dovoluje snadno nastavit systémové registry jádra a jednotlivých periférií, vstupů i výstupů pomocí volby vlastností z nabídek a menu a následně z toho vygenerovat kód v jazyku C, ale spolu s tím i vygenerovat konkrétní funkce v C, jejichž voláním například z hlavní programové smyčky lze nastavit nebo přečíst hodnoty výstupu, resp. vstupu, zahájit převod A/D převodníku, spustit časování, vyslat nebo přijmout bajt ze sériového rozhraní apod.

3.2.2. Rozhraní BDM

Pro spojení mikropočítače s PC obsahují mikroprocesory Freescale řady HCS08, a nejen tato řada, speciální jednovodičové rozhraní pro obousměrnou komunikaci. Pomocí rozhraní nazvaného podle módu do kterého se dostává mikroprocesor během jeho použití je nazýváno BDM (Background Debug Mode). Rozhraní umožňuje čtení a zápis do paměti programu (FLASH), a s pomocným monitorovacím programem nahreným výrobcem v jedné části FLASH paměti i mikroprocesor debugovat v reálném čase.

BDM rozhraní je řízeno interním blokem v mikroprocesoru nazvaném BDC (Background Debug Controller).

Obrázek BDM konektoru 3.7 ukazuje připojení pinu BKGD (obousměrný přenos dat), pinu RESET, a napájecí napětí +5V a GND.



obrázek 3.9.: BDM konektor

3.2.3. Programátor

Na obrázku 3.8 je vyobrazen programátor od firmy PE micro [8], který využívá rozhraní BDM.



obrázek 3.10.: programátor

4. Vypracování

4.1. Schéma zapojení

Viz popis zapojení jednotlivých částí.

4.2. Návrh antény

Nejprve je nutné vypočítat všechny potřebné parametry rezonančního obvodu. Rezonanční frekvence je dána použitým obvodem EM4095 na 125kHz. Pro použití vzorce (1) je nutné znát hodnotu C_0 , což je celková kapacita obvodu, vypočítá se dle

$$f_0 = \frac{1}{2\pi\sqrt{L_A \cdot C_0}} \quad (1)$$

vzorce (2). Dále je již možno ze vzorce (1) odvodit výpočet hodnoty indukčnosti navrhované cívky antény (3).

$$C_0 = C_{RES} + \frac{C_{DV1} \cdot C_{DV2}}{C_{DV1} + C_{DV2}} \quad (2)$$

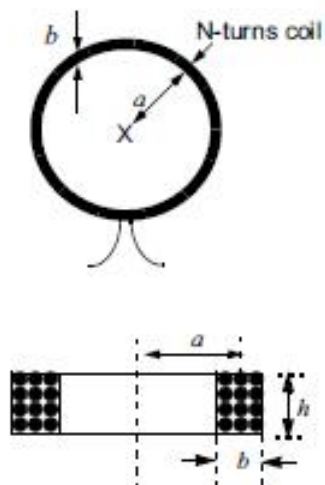
$$f_0 = \frac{1}{2\pi\sqrt{LC}} \Rightarrow \left(\frac{1}{2\pi f_0}\right)^2 = LC \Rightarrow L = \frac{\left(\frac{1}{2\pi f_0}\right)^2}{C_0} \quad (3)$$

Hodnoty kondenzátorů jsou určeny z typického zapojení [4].

Po dosazení do vzorců jsem vypočítal tyto hodnoty: $C_0=10\text{nF}$ a $L=162,1\mu\text{H}$.

Pro návrh antény jsem využil vzorců pro výpočet indukčnosti jednotlivých typů cívek pomocí materiálu Antenna Circuit Design for RFID od firmy Microchip [7]. Na obrázku 4.1 je zobrazen typ cívky, kterou jsem si k návrhu vybral. Rozměr a udává poloměr cívky v centimetrech, b šířku v centimetrech, h výšku v centimetrech a N počet závitů. Dle vzorce (4) jsem dopočítal všechny parametry.

$$L = \frac{0,31(aN)^2}{6a + 9h + 10b} \quad (4)$$



obrázek 4.11.: vícevrstvá kruhová cívka

Po navinutí cívky (viz obr. 4.2) bylo nutné změřit indukčnost a dle potřeby odmotat či přimotat malý počet závitů, abych dosáhl požadované indukčnosti.



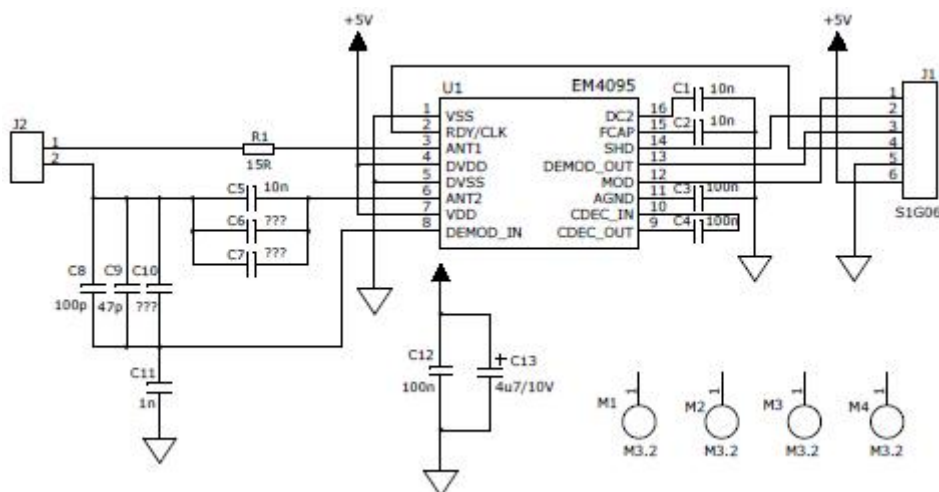
obrázek 4.12.: vyrobené cívky

Tato cívka má velice špatné vyřazovací schopnosti, pro dosažení čtení čipu bylo nutné tento téměř přiložit na danou cívku, proto jsem vytvořil cívku na plošném spoji. Vzhledem k tomu, že jsem neměl dostupný software na návrh antény na DPS navrhl jsem více typů cívek, odlišných počtem závitů a po změření indukčnosti se rozhodl která je nejvhodnější.

4.3. Popis zapojení

4.3.1. RFID

Hlavní obvodem RFID komunikace (viz obr. 4.3) je integrovaný obvod IO1 EM4095 v pouzdru SO20 firmy EM Microelectronic. Obvod EM4095 ke své činnosti potřebuje napájení + 5V. Napájecí napětí obvodu EM4095 je odrušeno tantalovým kondenzátorem C13 a keramickým kondenzátorem C12. Keramické kondenzátory C8, C9, C10 s kondenzátorem C11 tvoří kapacitní dělič napětí pro vstup DEMOD_IN obvodu EM4095. Kapacitní dělič chrání vstup DEMOD_IN proti velkému napětí vznikajícího na cívce rezonančního obvodu při jejím buzení.



obrázek 4.13.: RFID část schématu.

Externí cívka (anténa) s rezonančními keramickými kondenzátory C5, C6 a C7 tvoří sériový rezonanční obvod. Rezistor R1 upravuje činitel jakosti antény. Keramický kondenzátor C4 odstraňuje stejnosměrnou složku přijímaného signálu na vstupu DEMOD_IN. Keramický kondenzátor C1 odstraňuje zbývající nosnou AM

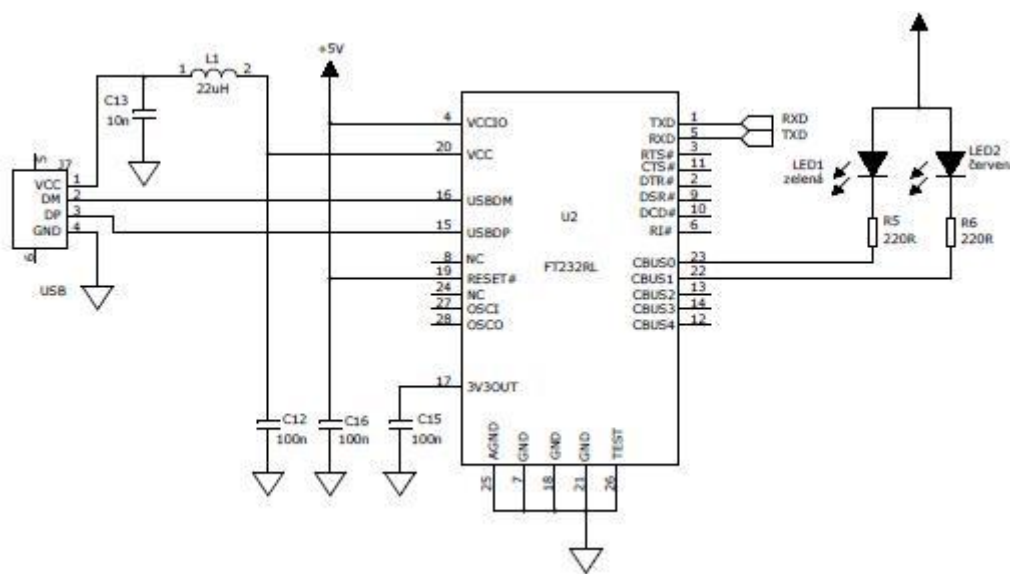
signálu a případný šum. V obvodu fázového závěsu PLL je zapojen keramický kondenzátor C2 tvořící filtr typu dolní propust. Keramický kondenzátor C3 slouží ke správné funkci obvodu EM4095 nastavením vnitřního referenčního bodu.

4.3.2. Řídící část

Hlavní řídicí obvod je mikroprocesor IO2 MC9S08JM32 v pouzdru LQFP64 firmy Freescale (viz obr. 4.4). Napájecí napětí mikroprocesoru je + 5 V. K mikroprocesoru jsou připojeny blokovací keramické kondenzátory C1 a C2 zapojené k vývodům V_{DD} , V_{REFH} a V_{DDAD} . Na vývody EXTAL, XTAL je možné zapojit externí krystalový oscilátor X1. Ke krystalovému oscilátoru X1 jsou podle doporučení výrobce zapojeny keramické kondenzátory C4, C5 (22 pF) a rezistor R2 s hodnotou 10 M Ω . Na vstup RESET je připojen pull-up rezistor R3 5,6 k Ω s blokovacím keramickým kondenzátorem C13 100 nF tvořící RC filtr k zajištění lepší elektromagnetické kompatibility. Piny mikroprocesoru PTE2, PTE3, PTF4 a IRQ jsou využívány ke komunikaci s obvodem EM4095. Pomocí konektoru J4 je propojena řídicí DPS s DPS s obvodem EM4095. Port PTF (PTF5 - PTF7) je využit pro řídicí signály displeje LCD1. Datové signály pro displej LCD1 jsou zapojeny na port Port PTF (PTF0 - PTF3). Obsluha klávesnice je řešena na portech PTA (PTA0 – PTG4). Na port PTD jsou zapojeny LED diody LED1 až LED8 (8 kusů), které mohou být využity pro indikaci různých stavů naprogramovaných v mikroprocesoru. Na port PTB jsou zapojeny tranzistory T1 až T8 (8 kusů), které mohou být využity pro spínání zátěže proti zemi (GND).

K naprogramování mikroprocesoru a případnému ladění programu v aplikaci je k mikroprocesoru zapojen debugovací BDM konektor J6. Do konektoru J6 jsou zapojeny signály RESET a BKGDM pro komunikaci s mikroprocesorem, napájecí napětí + 5 V a zem GND.

CBUS2 jsou zapojeny LED diody LED1 a LED2 spolu s rezistory R5, R6 s hodnotami 220Ω. LED diody indikují přítomnost dat procházejících přes obvod FTDI, tzv. příjem Rx a vysílání Tx na portu UART. Výstupy TXD a RXD z obvodu FT232RL jsou zapojeny přímo na piny mikroprocesoru.

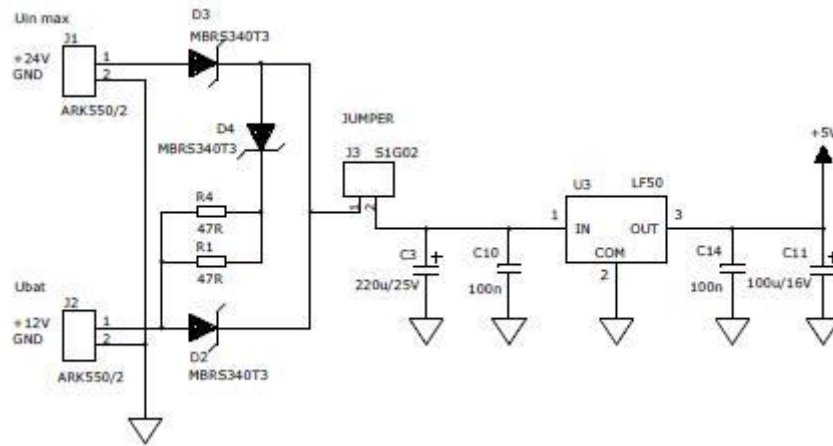


obrázek 4.15.: zapojení převodníku USB/UART

4.3.4. Stabilizace napájení

Napájecí část tvoří stabilizátor U3 LF50 v pouzdru D-PAK (viz obr. 4.6). Vstupní externí stejnosměrné napětí z konektoru J1 jde přes ochranou diodu D3 MBRS340T3, přes propojku (jumper) J3 na filtrační elektrolytický kondenzátor C3 220uF/25V. Ochranná dioda D3 zamezuje poškození stabilizátoru přepólováním vstupního napětí. Keramické kondenzátory C10, C14 (100nF) jsou zapojeny podle doporučení výrobce stabilizátoru U3. Slouží k zamezení rozkmitání stabilizátoru a k filtraci napětí. Elektrolytický kondenzátor C11 100μF/16V zlepšuje filtraci stabilizovaného napájecího napětí + 5 V. Konektor J2 umožňuje připojení i externího zdroje - akumulátoru s hodnotou napětí 6-12V současně i s připojeným zdrojem napětí na konektoru J1. Akumulátor může sloužit jako záložní zdroj. Pro přizpůsobení obou zdrojů napětí k stabilizátoru U3 a proti vzájemnému ovlivňování obou zdrojů je zapojena dioda D2 MBRS340T3. Jednoduchý nabíjecí obvod pro dobíjení

akumulátoru je složen z diody D4 MBR5340T3 a rezistorů R1 a R4 s hodnotami 47Ω. Rezistory omezují maximální nabíjecí proud do akumulátoru.

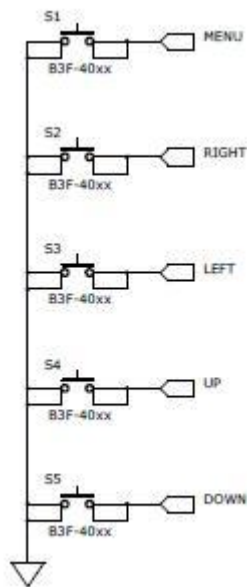


obrázek 4.16.: stabilizace napájení.

4.3.5. Uživatelská rozhraní

Klávesnice

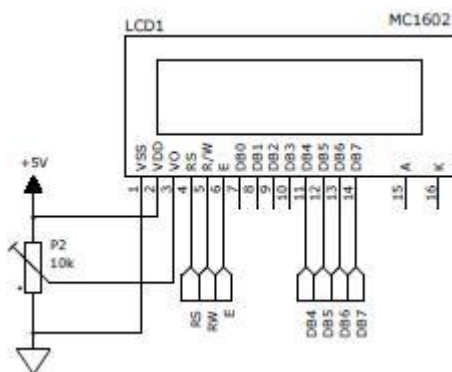
Klávesnici tvoří 5 spínacích tlačítek S1 – S5 (viz obr. 4.7) zapojených na porty mikroprocesoru. Tlačítka jsou zapojena na porty, které mají interně v mikroprocesoru připojeny tzv. pull-up rezistory zajišťující v klidu logickou úroveň H tedy + 5 V. Po stisku tlačítka je na portu mikroprocesoru logická úroveň L tedy 0 V (GND).



obrázek 4.17.: klávesnice

Displej

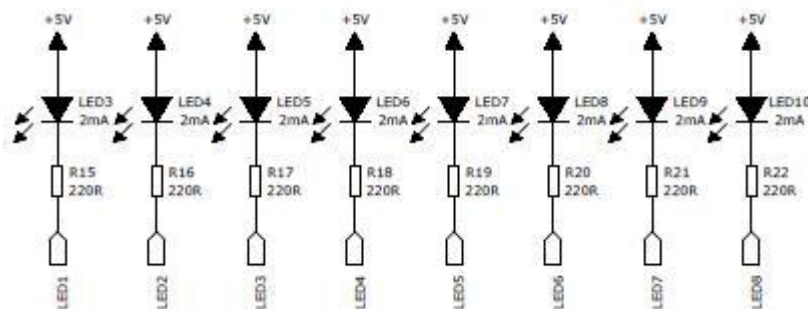
Pro komunikaci s obsluhou je použit dvouřádkový LCD displej 16x2 znaků LCD1 (viz obr. 4.8). Datové a řídicí piny LCD displeje jsou zapojeny k mikroprocesoru. Napájecí napětí logiky LCD displeje a LED podsvětlení je napájeno + 5 V. Trimr P2 10 k Ω slouží k regulaci kontrastu znaků zobrazovaných na displeji LCD1.



obrázek 4.18.: LCD displej.

Výstupní pole informačních LED diod

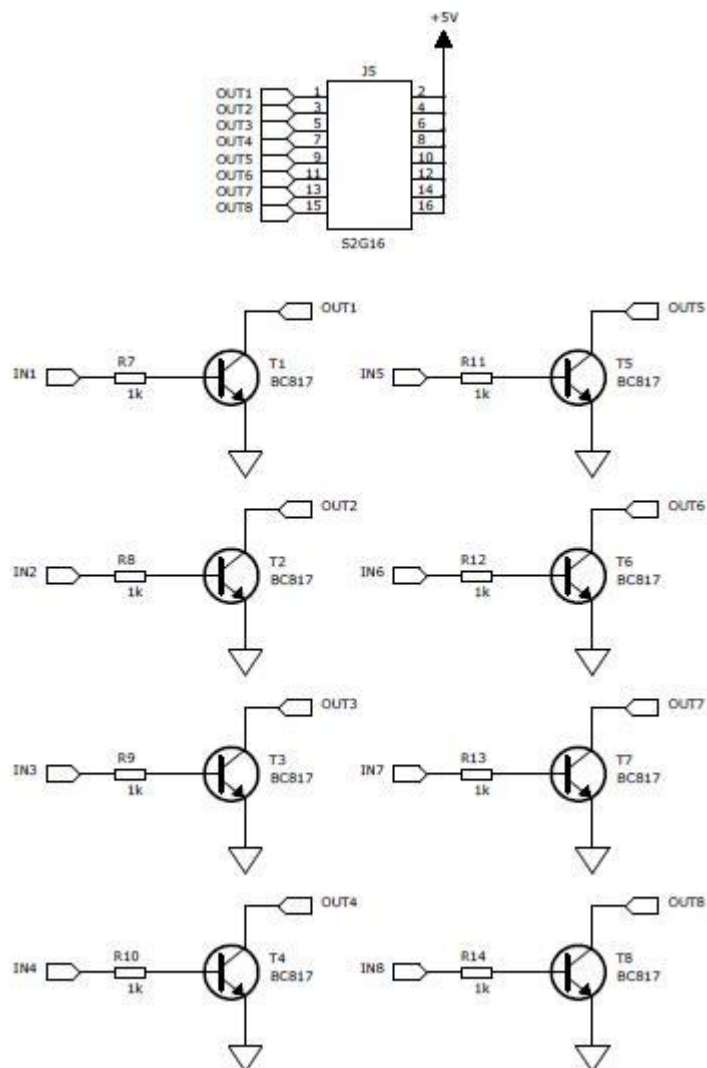
Na volné porty mikroprocesoru byly přidány LED diody LED3 až LED10 pro indikaci o stavech naprogramovaných funkcí a režimů např. pro práci s RFID. Jedná se pouze o indikační pole jak je vidět na obr. 4.9. Rezistory R15 až R22 s hodnotou 220Ω omezují proud LED diodami na cca 12mA.



obrázek 4.19.: pole LED diod

Výstupní pole tranzistorů

Na volné porty mikroprocesoru byly přidány tranzistory T1 až T8 BC817 v pouzdru SOT-23, umožňující spínání zátěže do 0,5 A proti zemi (GND). I zde je možné si tyto výstupy uživatelsky naprogramovat a používat ve spojení s funkcemi RFID. Jak je vidět na obr. 4.10 rezistory R7 až R14 s hodnotou $1k\Omega$ omezují proud do bází tranzistorů z portu mikroprocesoru. Kolektory tranzistorů jsou vyvedeny na jednu řadu pinů konektoru J5. Druhá řada pinů konektoru je připojena na napájecí napětí + 5 V.



obrázek 4.20.: pole tranzistorů

4.4. Popis FirmWare

Program je koncipován jako použití více ovladačů v hlavním programu. Vytvořil jsem ovladače pro různé části hardware, jako LCD displej, RFID část, USB modul a ovladačů pro funkce procesoru jako např. zpoždění, a podprogramy přerušení např. od časovače nebo sériového rozhraní. Dále uvedu pouze stručný popis jednotlivých ovladačů, jejich přesný zdrojový kód je uveden v příloze na DVD.

Ovladač je realizován dvěma typy souborů, tzv. hlavičkový soubor, který obsahuje definice proměnných a definici procedur, a soubor s příponou *.c, který obsahuje zdrojové kódy procedur nebo funkcí.

4.4.1. Popis ovladačů a procedur

LCD displej (lcd_driver.c lcd_driver.h)

Základní definicí v hlavičkovém souboru je definice portu procesoru ke kterému je displej připojen, definice pinů, na které jsou připojeny řídicí signály. Dále jsem nadefinoval konstanty, které odpovídají řídicím signálům LCD displeje (viz. Příloha 2). Uvedu příklad pro vymazání displeje:

```
static unsigned char lcd_inst_clrdisplay = 0x01;
```

Další příklad je pro definici funkce zapsání znaku do paměti DDRAM tzn. zobrazení na displeji. Funkce nemá žádný výstup (void na začátku definice [4]) a vstupní parametr je 1 byt (ASCII kód znaku, který chceme zobrazit):

```
void lcd_send_char (unsigned char data);
```

Dále jsou uvedeny a vysvětleny všechny použité funkce:

```
-void lcd_init_4bit ()
```

Inicializace displeje, nejprve je nutné nastavit 8-bitovou komunikaci, zobrazení jednoho řádku a nastavení zobrazování matice 5x7 bodů. Po tomto kroku je nutné dát displeji čas na zotavení a vyčkat minimálně 4,1ms. Následuje nastavení 4bitové komunikace nejprve pro jednořádkový mód a následně pro dvouřádkový mód. Dalšími kroky je vypnutí a zapnutí displeje, nastavení automatické inkrementace kurzoru po zapsání znaku a vymazání displeje. Mezi tyto kroky je opět nutné vložit zpoždění.

```
-void lcd_putnybble (char n)
```

Jelikož jsem vybral pro svoji práci 4-bitovou komunikaci, je tedy nezbytné vytvořit tuto funkci, která rozdělí 8 bitů (n) na dvě 4-bitové části a vyšle data na port procesoru, který je připojen k displeji. Tato funkce je základem všech dalších funkcí.

```
-void lcd_send_char (unsigned char data)
```

Tato funkce má jako vstup 1 bytové číslo, které reprezentuje ASCII znak a tento vyšle na datový port displeje a tím ihned zobrazí na aktuální pozici displeje. Nezbytnou součástí této funkce je zpoždění po vykonání zápisu do displeje (viz Příloha XY).

```
-void lcd_send_cmd (unsigned char data)
```

Tato funkce vznikla vzhledem k nutnosti rozlišit data, která posíláme do displeje na tisknuté znaky a řídicí příkazy pro displej. Výše uvedená funkce `lcd_send_char` slouží pro vysílání dat a tato funkce pro zasílání řídicích znaků. I tady je nutné po vyslání dat připojit zpoždění, ale již je nutné rozlišovat mezi dvěma druhy příkazů. U funkce smaž displej a nastav kurzor na počáteční adresu je nutno připojit větší zpoždění než u ostatních funkcí.

```
-void lcd_send_string (unsigned char *data)
```

Tato funkce má jako vstup ukazatel do paměti, kde se nacházejí data, to může být obecně jedna položka nebo více. Já předpokládám, že se v paměti nachází řetězec znaků, které chceme zobrazit na displeji. Tato data jsou postupně pomocí funkce `lcd_send_char` posílána do displeje. Data jsou uložena paměti DDRAM a pokud je řetězec delší, než je rozlišení připojeného displeje, je zobrazena pouze část řetězce ale ostatní data nejsou ztracena a lze je posunutím displeje zobrazit.

```
-void lcd_getxy (unsigned char x, unsigned char y)
```

Vstupem této funkce je pozice na displeji kam chceme nastavit kurzor: `x` odpovídá řádku a `y` odpovídá sloupci v maticové interpretaci zobrazovací oblasti displeje. Tato funkce nejprve nastaví kurzor na první pozici displeje a poté posune kurzor o určitý počet míst doprava. Pokud se požaduje nastavit kurzor na prvním řádku, není ve funkci potřeba se posunout na druhý řádek. Pokud je hodnota `x` větší jak 1, tzn. Je třeba posunout kurzor na další řádek, posunuje se kurzor z první pozice nejen o požadovaný počet sloupců, ale o počet zvětšený o maximální velikost paměti DDRAM jednoho řádku (viz obr. 3.6).

Ovladač pro předem definované zpoždění (delay.h, delay.c)

Vytvoření tohoto ovladače vzniklo nejen z nutnosti realizace zpoždění pro ovládání LCD displeje, ale i pro ovládání dalších periférií, jako např. USB modul. Vytvořil jsem funkci na zpoždění cca 1 μ s realizovanou pomocí několika prázdných instrukcí procesoru v assembleru. Využitím této funkce jsem definoval další dvě na volitelné zpoždění v 10kách mikrosekund (`void delay_10uS(unsigned int x10uS)`) a stovkách mikrosekund (`void delay_100uS(unsigned int x100uS)`).

Ovládání klávesnice

Kód pro ovládání klávesnice je vložen v souboru MCUinit.c kde se nachází procedura pro obsluhu přerušení při přetečení timeru 1. Timer má nastaven čítač tak, aby každých 200ms docházelo k přetečení (přerušení). V tomto přerušení se testují vstupy procesoru a podle toho, které tlačítko bylo stisknuto (hodnota 0 na příslušném pinu) se nastaví příznak. S těmito příznaky se pracuje v hlavní části programu.

Procedury pro práci s RFID

Tyto procedury jsou umístěny v hlavní části programu, ale je možno je umístit zvlášť do samostatného souboru, který bude dát využíván jako ovladač. Pro práci se signálem příchozím z RFID části je nutno využít timer procesoru s funkcí input capture, tj. vyvolání přerušení procesoru pokud na vstupu je detekována nástupná nebo sestupná hrana signálu.

```
void em4095_init () {  
    PTDD |= SHD;  
    PTDD |= MOD;  
    PTDD &= ~SHD;  
    PTDD &= ~MOD;  
}
```

Základem pro práci s obvodem EM4095 je jeho správná inicializace. Pro správnou funkce je nutné nastavit vstupy SHD a MOD do nuly. SHD na nízké úrovni značí probuzení zařízení ze slep módu a signál MOD v nule zajistí maximální výstup.


```

void timer_init (char edge2){
    TPM2SC = 0x00;           // stop timer
    TPM2CNTH = 0x00;        // clear counter
    TPM2CNTL = 0x00;        // clear counter
    if (edge2 == 1) TPM2C0SC = 0x44; // set rising edge
    else             TPM2C0SC = 0x48; // set falling edge
}

```

Slouží k nastavení funkce timeru, na kterou hranu (nástupnou či sestupnou) bude reagovat. Vstupním parametrem je hodnota proměnné edge2, která uchovává informaci o požadované funkci. Jednička znamená že chceme nastavit nástupnou hranu a nula odpovídá sestupné hraně. Prvním krokem je zastavení timeru, což provedeme zápisem hodnoty 0x00 do registru TPM2SC, dále je potřeba vynulovat počítadlo pulsů, to se provede vynulováním příslušných registrů. Teď jen stačí v kontrolním registru nastavit příslušné bity (viz popis procesoru).

```

void set_rf_edge (char edge){
    if (edge) {
        timer_init(1);           // call subroutine
        edge_dir = 1;           // rising edge
    }
    else{
        timer_init(0);           // call subroutine
        edge_dir = 0;           // falling edge
    }
}

```

Využívá předchozí procedury k nastavení požadované funkce timeru a nastavení proměnné, která udává funkci timeru.

```

void checksum_h (){
    unsigned char j;
    unsigned char k;
    unsigned char point = 0;
    unsigned char parity = 0;
    for (j=1;j<11;j++){
        parity = 0;

```

```

for (k=1;k<5;k++){
    if (rf_buff[point]==1) parity ^=0x01;
    point++;
}
if (rf_buff[point] != parity) parity_err = 1;
else parity_err = 0;
point++;
}
}

```

Jedná se o výpočet horizontální parity. Pro všech 10 bloků (4 bity) se vypočítá parita a srovnává s následujícím bitem, pokud souhlasí parity pro všechny data vynuluje se příznak `parity_err`, v opačném případě se nastaví na 1. S tímto příznakem se pracuje v dalších částech programu.

```

void get_ID (){
    // 4bits to char
    unsigned char j;
    unsigned char point = 0;
    for (j=0;j<10;j++){
        ID[j] = 0;
        ID[j] += rf_buff[point] * 8;
        point++;
        ID[j] += rf_buff[point] * 4;
        point++;
        ID[j] += rf_buff[point] * 2;
        point++;
        ID[j] += rf_buff[point];
        if (ID[j]>9) ID[j] += 55; // offset for alphabetic
        else ID[j] += 48; // offset for numeric
        point++;
        point++;
    }
    ID[10] = '\0'; // terminating char
}

```

Převéde 4 datové bity na znak (číslo), který bude kompatibilní se zobrazením na displeji. Jednotlivé bity vynásobím příslušnou hodnotou váhy a sečtu, dále podle výsledku rozhodnu, zda se jedná o číslo nebo znak abecedy a přičtu příslušný offset. To se provede 10krát a uloží do paměti do proměnné ID. Posledním znakem musí

být ukončovací znak, aby při čtení nedocházelo ke přístupu do paměti mimo tuto proměnnou.

```
void get_bits (){
    fl = 0;                // flag find long
    ci = 0;                // counter interrupt
    set_rf_edge(0);       // set falling edge
    bit_saved = 1;        // first bit need to be saved
    bit_trans_num = 0;     // number of recieved bits
    old_width = 0;        // old pulse width
    irq = 0;              // inerrupt flag
    h_count = 0;          // onces counter in header
    head = 0;             // header flag
    parity_err = 1;       // parity flag
    rec_err = 1;          // recieved ok flag
    TPM2SC = 0x18;        // start timer RF
    while (rec_err){
        while (irq == 0){} // do nothing while interrupt occurred
        if (head){        // found header
            rf_buff[bit_trans_num] = bit_value; // store bit
            bit_trans_num++; // increment pointer
            if (bit_trans_num == 50) { // recieved 50 bits?
                TPM2SC = 0x00; // stop timer
                checksum_h (); // check horizontal parity
                if (parity_err == 1){ // wrong parity
                    head = 0; // reset head
                    bit_trans_num = 0; // reset pointer
                    TPM2SC = 0x18; // start timer
                }
                else rec_err = 0; // recive ok
            }
        }
        else { // not found header
            if (bit_value) { // recieved bit value 1
                h_count++;
                if (h_count == 9) head = 1; // counter ones in head
            }
        }
    }
}
```

```

        }
        else h_count = 0;
    }
    irq = 0; // clear interupt flag
}
TPM2SC = 0x00; } // stop timer

```

Tato procedura obsahuje všechny potřebné části k příjmu celého bloku dat včetně kontroly obsahu. V první řadě nastaví všechny potřebné příznaky a lokální proměnné, nastaví funkci timeru na reakci na sestupnou hranu signálu a spustí timer. Hlavní smyčka `while (rec_err)` bude probíhat do té doby, než bude v pořádku přijata celá posloupnost dat a zkontrolována. Druhá smyčka `while (irq == 0)` čeká, dokud nebude v přerušení nastaven příznak `irq` na hodnotu 1. Poté začne hledání hlavičky v příchozích toku dat. Hlavička musí obsahovat 9 po sobě jdoucích bitů s hodnotou 1. Dokud nebude nalezena hlavička, nebudou se žádná data ukládat. Po nalezení hlavičky může dojít k samotnému ukládání přijatých bitů. O tom zda uložit jedničku či nulu rozhoduje hodnota příznaku `bit_value`. Po přijetí 50ti datových jednotek musím provést kontrolu parity voláním procedury `checksum_v`, která nastaví příslušný příznak a podle něho buď označím příjem za zdařilý, nebo v opačném případě resetuji všechny příznaky a začnu celý proces znovu. Po každé je na konci druhé smyčky nutno vynulovat příznak přerušení `irq`, aby bylo opět možné čekat na další příchozí přerušení. Po úspěšném příjmu celého datového bloku a jeho kontrole je nutno vynulovat příznak `rec_err` aby bylo možné opustit hlavní smyčku a dokončit proceduru. Posledním krokem je zastavení timeru, aby nedocházelo ke stálému skákání do přerušení. Vývojový diagram uveden v příloze 3.

```

__interrupt void isrVtpm2ch0(void)
{
    unsigned int value;
    unsigned int width;
    TPM2C0SC &= 0x00; // stop timer
    value = TPM2C0V; // get pulse width
}

```

```

width =value - old_width;
old_width = value; // store old value width
if (edge_dir){
    TPM2C0SC &= ~0x04; // reset bitu
    TPM2C0SC |= 0x08; // set bit
    edge_dir = 0;
}
else {
    TPM2C0SC &= ~0x08; // reset bit
    TPM2C0SC |= 0x04; // set bit
    edge_dir = 1;
}
if (!ci) { ci = 1; // ignore first interrupt
    TPM2C0SC |= 0x40; // set falling edge
}
else {
    if (!fl) { // wait fo long pulse
        if (width>40 && width < 80) { // define long pulse
            fl = 1; // flag of long pulse
            bit_value = 1; // value of saved bit
            bit_saved = 0; // actual bit saved
            irq = 1; // interupt flag set
            TPM2C0SC |= 0x40; // set rising edge
        }
        else {
            fl = 0; // clear interrupt flag
            TPM2C0SC |= 0x40; // set rising edge
        }
    }
    else {
        if (width>40){
            bit_value = ~bit_value; // bit revers
            bit_value &= 0x01; // mask first bit
            irq = 1; // interupt flag set
            bit_saved = 0; // next bit will be sotred
        }
    }
}

```

```

    }
    else {
        if (bit_saved){                // will be bit store?
            irq = 1;                    // interupt flag set
            bit_saved = ~bit_saved;    // revers bit_saved
            bit_saved &= 0x01;
        }
        else {
            bit_saved = ~bit_saved;    // revers bit_saved
            bit_saved &= 0x01;
        }
    }
    TPM2COSC |= 0x40;                  // restart timer
}
}
}

```

Výše uvedený kód je obsažen v souboru MCUinit.c, jedná se o soubor s inicializací procesoru a podprogramy od různých přerušení. Tato část je pro obsluhu přerušení od timeru 2 kanál 0, který má aktivovaný mód input capture. Prvním krokem je zastavení timeru, následuje zjištění šířky pulsu, to znamená kolik proběhlo cyklů hodinového signálu od poslední změny hrany. S tímto parametrem budu pracovat dále. Nutností je na začátku každého přerušení nastavit opačnou hranu signálu na kterou bude timer reagovat. To se provede nastavením nebo vynulováním dvojice bitů v kontrolním registru TPM2COSC. Vzhledem k potřebě synchronizovat se na dlouhý puls ignoruji první přerušení. Dále čekám na příchozí dlouhý puls se sestupnou hranu a nastavím potřebné příznaky, se kterými pracuje procedura get_bits. Dále už jen rozhoduji, zda přišel krátký nebo dlouhý impuls. V prvním případě vždy nastavím příznak ukládání bitu a uloží se opačná hodnota bitu než v předchozím přerušení. V případě krátkého pulsu se rozhoduji zda předcházel krátký nebo dlouhý puls. Pokud přecházel krátký, tak se uloží stejná hodnota bitu jako v minulém přerušení a pokud dlouhý, tak se jen nastaví příznaky a nic se neukládá. Vývojový diagram k této proceduře se nachází v příloze 3.

Přijem dat z USB

Obslužný kód je opět implementován v obsluze přerušení od sériové linky mikroprocesoru. V této obsluze se pouze ukládají data, která přichází ve formátu čísla (ASCII znak) do paměti. Po příjmu 16 znaků se nastaví příznak úspěšného příjmu a případná další data jsou ignorována.

4.4.2. popis hlavní části programu (main.c)

Obsahuje v první řadě inicializaci procesoru, LCD displeje obvodu EM4095. Dále následuje nekonečná smyčka, která obsahuje 3 základní bloky.

```
MCU_init();                /* call Device Initialization */
lcd_init_8bit ();          // init LCD display
em4095_init ();           // init IC EM4095
start_win ();             // start_up window
```

V prvním bloku je rozhodovací část, která testuje zda přišlo přerušení od FID části a jeho obsluha. Volá se procedura `get_bits`, která zařídí příjem kompletní datové struktury, dále procedura `get_ID` extrahuje identifikační kód. Tento kód se porovnává s údaji v paměti a pokud je nalezena shoda, je na displeji zobrazeno ID a popis tagu. V opačném případě se zobrazí jeho ID a dotaz na vložení popisu pomocí programu na PC a rozhraní USB. Nastaví se potřebné příznaky k vykonávání programu.

```
if (irq2 ==1){
    get_bits ();           // get bits from reciever
    get_ID ();            // extract ID card
    sum1 = 0;             // clear pointers
    sum2 = 0;
    sum3 = 0;
    for (i=0;i<10;i++){   // find tag in memory
        if (ID[i]==tag1[i]) sum1++;
        if (ID[i]==tag2[i]) sum2++;
        if (ID[i]==tag3[i]) sum3++;
    }
}
```

```

if (sum1 >= 10) {
    print_tag (tag1,popis1);
    TPM2SC = 0x18;
}
else if (sum2 >= 10) {
    print_tag (tag2,popis2);
    TPM2SC = 0x18;
}
else if (sum3 >= 10) {
    print_tag (tag3,popis3);
    TPM2SC = 0x18;
}
else {
    // no match in memory
    lcd_send_cmd (lcd_inst_clrdisplay);
    lcd_send_string ("ID: ");
    lcd_send_string (ID);
    lcd_getxy(2,1);
    lcd_send_string ("<ano popis ? ne");
    flag_popis = 1;
}
rec_err = 1;
irq2 = 0;
}

```

Další blok obsahuje rozhodovací část, zda je vyžadována obsluha klávesnice. Pokud ano, testuji zda byla zmáčknuta klávesa s šipkou vlevo či vpravo, na ostatní klávesy v této fázi program nereaguje. Po stisku klávesy vlevo program čeká na příjem popisu z PC a na displeji se zobrazí nápověda pro zrušení přidání popisu. Pokud je příjem kompletní, dojde k vypsání popisu na displej a nastavení příznaku zápisu do paměti. Pokud během čekání dojde ke stisku klávesy OK, vypíše se na displej zpráva o zrušení přidávání popisu. Po stisku klávesy vpravo dojde k odmítnutí vložení popisu a zobrazí se zpráva nezvolen popis.

```

if (flag_popis == 1){
    if (Flag_Key == Flag_Left) {
        // if pressed Left key

```



```

Flag_Key = 0x00;                // clear flag
lcd_getxy(2,1);                 // display question
lcd_send_string ("   zrusit   OK");
usb = 0;
p = 0;
while (usb == 0){               // wait for data from USB
    if (Flag_Key == Flag_Start){ // except OK key
        Flag_Key = 0x00;
        lcd_getxy(2,1);
        lcd_send_string ("   zruseno   ");
        usb = 1;
        zrus = 1;
        flag_popis = 0;
        TPM2SC = 0x18;          // start RF timer
    }
}
rec_mess[16] = '\0';           // terminating char
}
if (Flag_Key == Flag_Right) { // if pressed Right key
    flag_popis = 0;             // clear flags
    Flag_Key = 0x00;
    lcd_getxy(2,1);
    lcd_send_string (" nezvolen popis ");
    flag_popis = 0;
    TPM2SC = 0x18;             // start RF timer
}
}

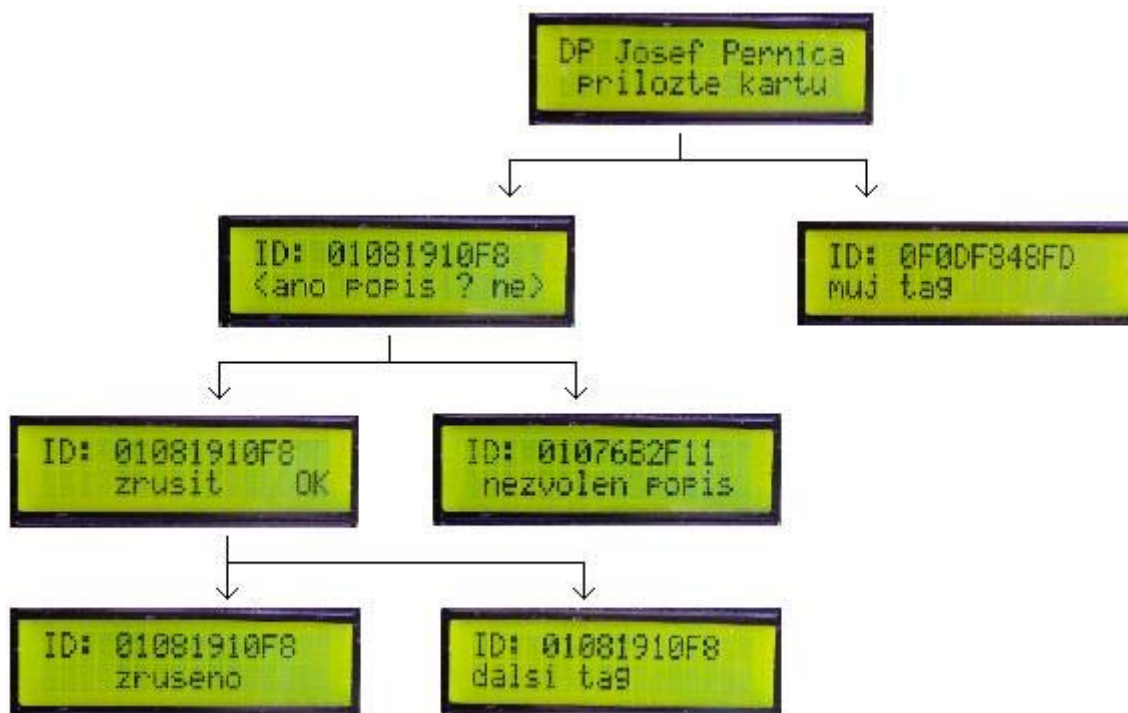
```

V posledním bloku je rozhodovací část, zda je vyžadováno zapsání popisu do paměti mikroprocesoru. Pokud jsou nastaveny potřebné příznaky, dojde k uložení popisu na příslušné paměťové místo a nastavení příznaků.

Jelikož se všechny tyto části nacházejí v nekonečné smyčce, dochází k jejich neustálému vykonávání a program je tedy řízen nastavováním příznaků.

4.4.3. Popis menu

Po zapnutí zařízení se zobrazí úvodní obrazovka a výzva k přiložení karty. Pokud je přiložena již známá karta, zobrazí se na displeji její identifikační číslo a popis. Pokud je karta neznámá, zobrazí se dotaz, zda chceme přidat popis karty. Pokud ano, zařízení čeká na data, která jsou posílána z PC přes rozhraní USB. Toto čekání lze nerušit stisknutím klávesy OK. Po úspěšném přenosu dat se zobrazí popis, který byl přijat a uloží se do paměti. Pokud odmítneme přidání popisu, či jeho přidávání, zobrazí se na displeji zpráva o tomto stavu. Během čekání na stisk klávesy nebo během příjmu dat z PC není možné číst další kartu (tag). Schéma menu je zobrazeno na obrázku 4.11.

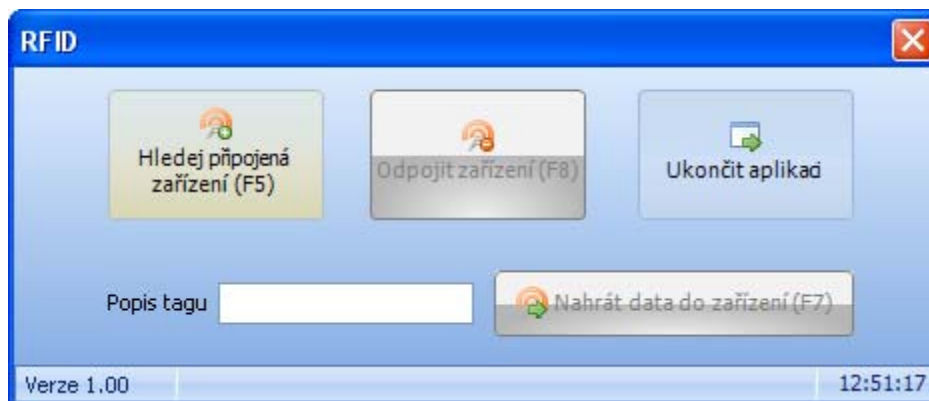


obrázek 4.21.: menu zařízení

4.5. Popis Software

Pro zadání popisu tagu jsem využil rozhraní USB. Program pro zaslání popisu RFID.exe jsem zapůjčil od firmy Matron. Součástí zadání nebylo vytvoření takového

programu, ale pro lepší demonstraci funkcí přípravku jsem zvolil toto řešení. Vzhled programu je na obrázku 4.12.



obrázek 4.22.: program RFID.exe

Program detekuje připojené zařízení na USB portu a provede jeho inicializaci. Přenosová rychlost je nastavena na 9600Bd. Tato pevná rychlost je i nastavena při inicializaci procesoru. Je možno vložit 16 znaků popisu a odeslat na port USB. O obsluhu dat se stará program v mikroprocesoru.

4.6. Přípravek

Ke každému dílčímu plošnému spoji přípravku je v přílohách uveden motiv plošného spoje strana A a B a osazovací plán strany B.

V další samostatné příloze je foto osazených DPS.

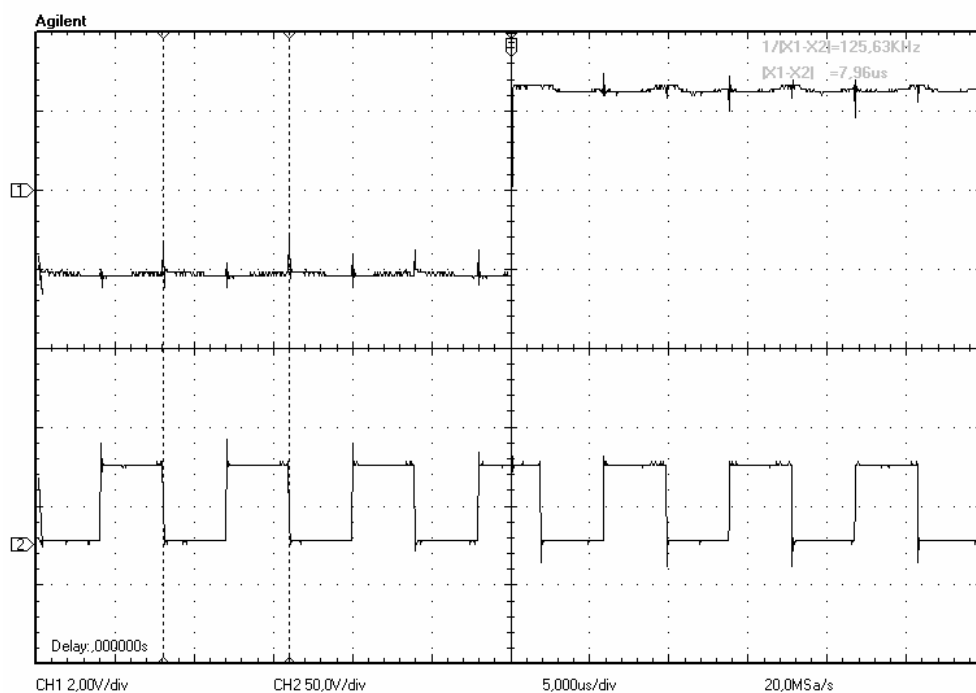
Poslední příloha obsahuje seznam součástek.

4.6.1. Měření průběhů signálů

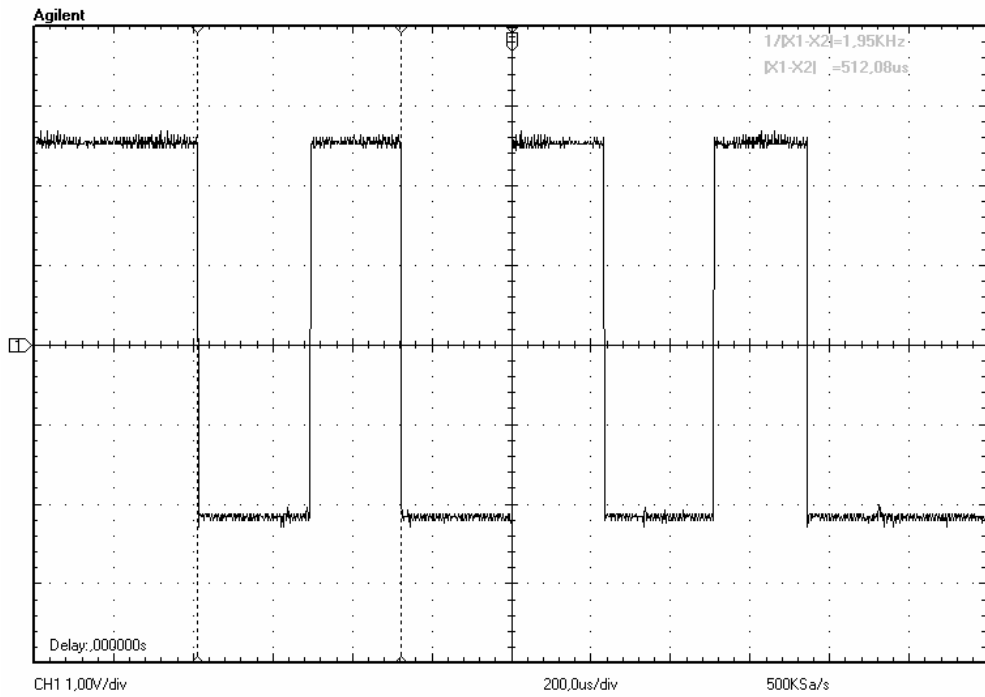
Datová komunikace probíhá na dvou vodičích, na výstupu RDY/CLK je hodinový signál (obr. 4.13) a na vstupu DEMOD_OUT je demodulovaný signál v kódu Manchester (obr 4.14).

Na obrázku 4.15 jsou zobrazeny oba signály současně a je z nich patrné, že jednomu datovému bitu odpovídá 32 period hodinového signálu.

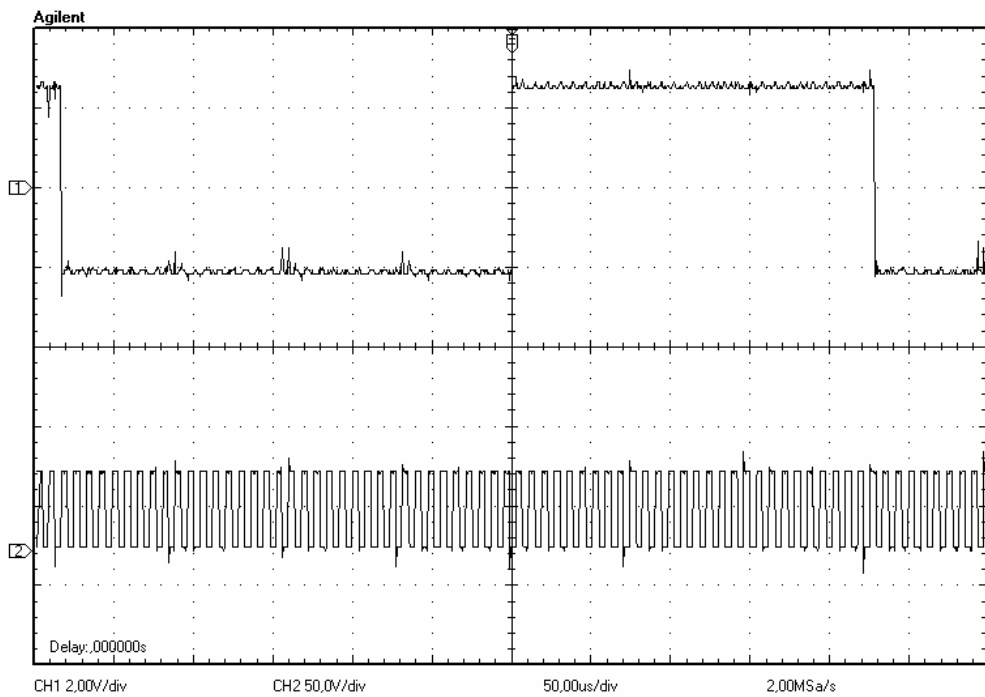
Na obrázku 4.16 je zobrazena perioda nosného signálu pro AM modulaci. Frekvence je přibližně 120kHz což koresponduje se základním principem RFID. Při měření byla použita starší verze mírně rozladěného rezonančního obvodu. Ve finální verzi přípravku je rezonanční kmitočet přesně 125kHz.



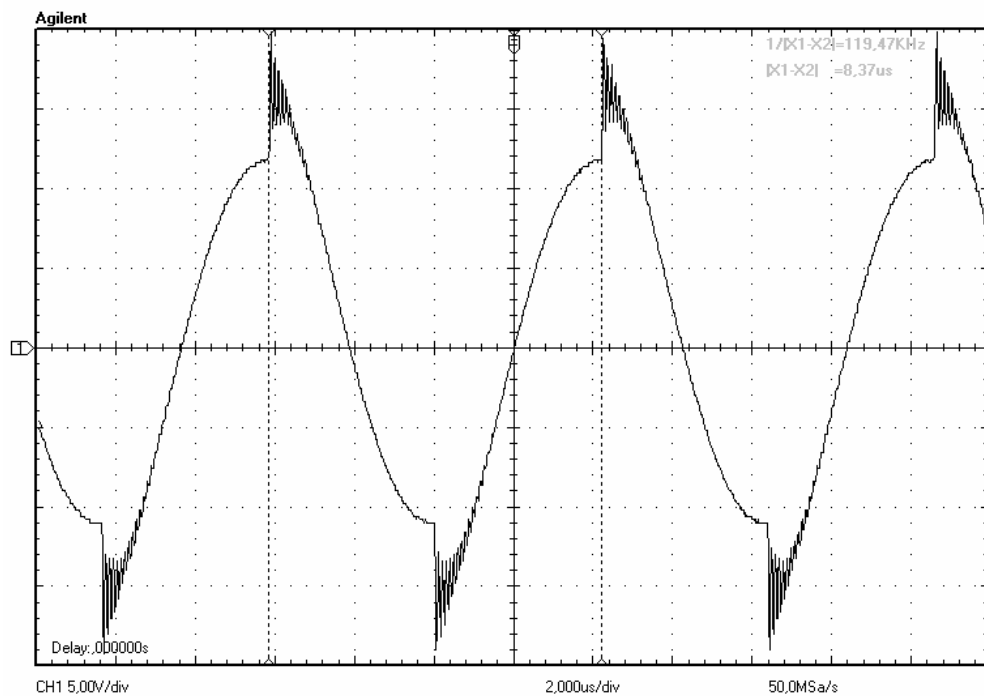
obrázek 4.23.: průběh hodinového signálu



obrázek 4.24.: průběh dat v kódu Manchester



obrázek 4.25.: společný průběh signálů



obrázek 4.26.: průběh nosné pro AM

Pro měření průběhu jsem využil digitální osciloskop firmy Agilent DSO 3102A, měření jsem prováděl v projektové laboratoři na Ústavu telekomunikací.

5. Závěr

Diplomová práce se zabývá problematikou a realizací přípravku demonstrující systém RFID. V úvodních kapitolách se nachází historie a vývoj tagů s popisem čtecích zařízení od různých výrobců. Dále následuje podrobný teoretický popis použitých obvodů v následné praktické realizaci přípravku. Jsou popsány obvody EM4095, MC9S08JM32, FT232R, LCD displej 16x2 znaků.

Jako řídicí prvek byl vybrán mikroprocesor MC9S08JM32, který splňuje požadavky dané na přípravek a to je dostatečný počet výstupních portů, timer s možností externího vstupu hodinového signálu a sériové rozhraní.

Pro komunikaci uživatele s RFID kartami byl mikroprocesor doplněn o klávesnici a LCD displej 16x2 znaků, který je postačující pro zobrazení menu a uživatelského rozhraní.

RFID obvod EM4095 byl vybrán vzhledem k dobré dostupnosti a poměru cena/výkon. Je hojně využíván v průmyslových aplikacích.

Byl vytvořen ovládací firmware pro mikroprocesor, který tak umožní ovládání RFID obvodu EM4095 (čtení tagů), komunikaci s uživatelem a komunikaci s PC.

Výsledný přípravek je koncipován tak, aby bylo možno připojit jiné obvody pro komunikaci s RFID tagy a různé typy antén. Tím se stává přípravek univerzální a popřípadě i vhodný pro studijní účely.

6. Použitá literatura

- [1] RFID [online] url: <http://www.wikipedia.org/wiki/RFID>
- [2] RFID tagy: <http://www.lux-ident.com/cs/produkty>
- [3] Technická specifikace LCD displeje: <http://www.ges.cz/sheets/c/cm1624.pdf>
- [4] Technická specifikace obvodu EM4095
http://www.emmicroelectronic.com/webfiles/Product/RFID/DS/EM4095_DS.pdf
- [5] Mann Burkhard, C pro mikrokontroléry. 1. vydání, nakladatelství BEN,
ISBN 80-7300-077-6, 2003
- [6] Antenna Circuit Design for RFID:
<http://ww1.microchip.com/downloads/en/AppNotes/00710c.pdf>
- [7] Technická specifikace obvodu MC9S08JM60:
http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08JM60.pdf
- [8] Technická specifikace programátoru:
http://www.freescale.com/files/soft_dev_tools/doc/fact_sheet/USBMULTILINKBDMFS.pdf?srch=1

7. Seznam tabulek, obrázků, použité symboly

7.1. Seznam tabulek

Tabulka 1.: provozní podmínky	20
Tabulka 2.: popis pouzdra	22
Tabulka 3.: zapojení vývodů	31

7.2. Seznam obrázků

obrázek 2.1.: klíčenky “Teardrop”	14
obrázek 2.2.: klíčenky “Sail”	14
obrázek 2.3.: hodinky	14
obrázek 2.4.: bezkontaktní karty	15
obrázek 2.5.: různé možnosti vizualizace tagů	15
obrázek 2.6.: kód Manchester	18
obrázek 3.4.: blokové schéma FTDI	28
obrázek 3.6.: paměť DDRAM	32
obrázek 3.7.: BDM konektor	34
obrázek 3.8.: programátor	34
obrázek 4.1.: vícevrstvá kruhová cívka	36
obrázek 4.2.: vyrobené cívky	36
obrázek 4.3.: RFID část schématu	37
obrázek 4.4.: řídicí část zapojení	39
obrázek 4.5.: zapojení převodníku USB/UART	40
obrázek 4.6.: stabilizace napájení	41
obrázek 4.7.: klávesnice	42
obrázek 4.8.: LCD displej	42
obrázek 4.9.: pole LED diod	43
obrázek 4.10.: pole tranzistorů	44
obrázek 4.11.: menu zařízení	57
obrázek 4.12.: program RFID.exe	58
obrázek 4.13.: průběh hodinového signálu	59
obrázek 4.14.: průběh dat v kódu Manchester	60
obrázek 4.15.: společný průběh signálů	60
obrázek 4.16.: průběh nosné pro AM	61

7.3. Použité symboly

RFID	Radio Frequency Identification
IO	Integrovaný Obvod
PC	Personal Computer
CPU	Central Processing Unit
USB	Universal Serial Bus
RAM	Random-Access Memory
ROM	Read-Only Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
ASCII	American Standard Code for Information Interchange
FIFO	First In, First Out
IIC	Inter-Integrated Circuit
SPI	Serial Peripheral Interface

UART	Universal Asynchronous Receiver Transmitter
QFP	Quad Flat Pack
QFN	Quad Flat Network
SOIC	Small-Outline Integrated Circuit
AM	Amplitude Modulation
FSK	Frequency-Shift Keying
OOK	100% AM
PLL	Phase Locked Loop
FTDI	Future Technology Devices International
BDM	Background Debug Mode
BDC	Background Debug Controller
UID	User Identifier
PID	Personal Identifier
GND	Ground

8. Seznam příloh

- Příloha 1: Znaková tabulka pro LCD displej
- Příloha 2: Tabulka instrukcí pro LCD displej
- Příloha 3: Vývojové diagramy
- Příloha 4: Blokové schéma MC9S08JM32
- Příloha 5: Deska plošných spojů řídicí části (motiv, osazovací plán)
- Příloha 6: Deska plošných spojů RFID části (motiv, osazovací plán)
- Příloha 7: Deska plošných spojů antény (motiv, osazovací plán)
- Příloha 8: Foto osazených DPS
- Příloha 9: Seznam součástek

Příloha 1 – znaková tabulka LCD displeje

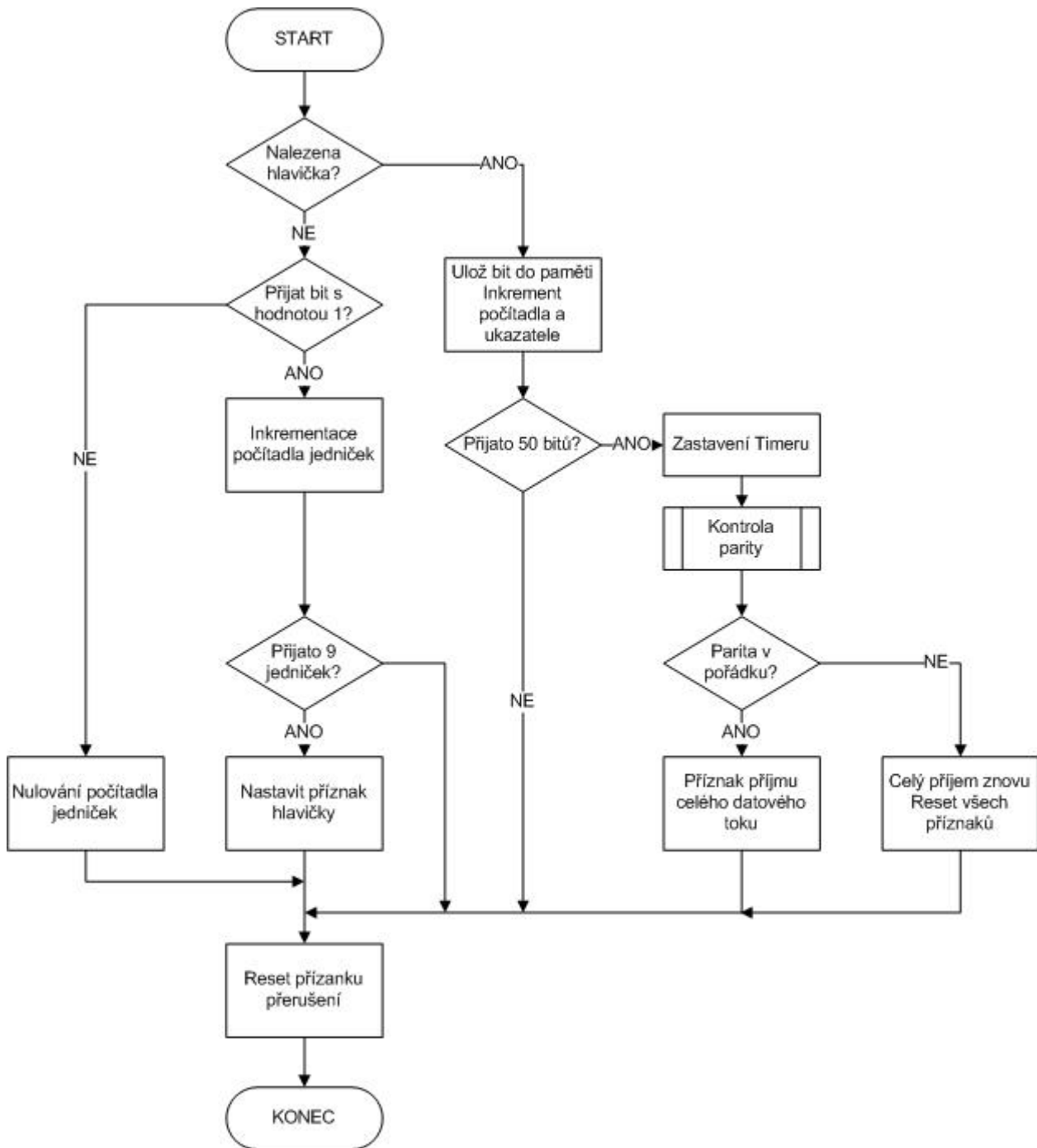
Upper bits LCM160100	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CC RAM (*)			0	1	P	`	P			-	夕	ミ	α	ρ	
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	△	ä	g
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	×	ρ	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		\$	4	D	T	d	t			、	エ	ト	カ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	ε	ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			フ	キ	ヌ	ラ	g	π
xxxx1000	(1)		(8	H	X	h	x			ィ	ク	ネ	リ	フ	×
xxxx1001	(2))	9	I	Y	i	y			ウ	ケ	ル		'	y
xxxx1010	(3)		*	:	J	Z	j	z			エ	コ	ン	レ	j	キ
xxxx1011	(4)		+	;	K	L	k	l			オ	サ	ヒ	ロ	*	斤
xxxx1100	(5)		,	<	L	¥	l	l			カ	シ	フ	ワ	φ	円
xxxx1101	(6)		-	=	M	l	m	}			ユ	ス	ヘ	ン	モ	÷
xxxx1110	(7)		.	>	N	^	n	→			ヨ	セ	ホ	°	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ツ	ソ	マ	°	ö	■

Příloha 2 – Tabulka instrukcí pro LCD displej

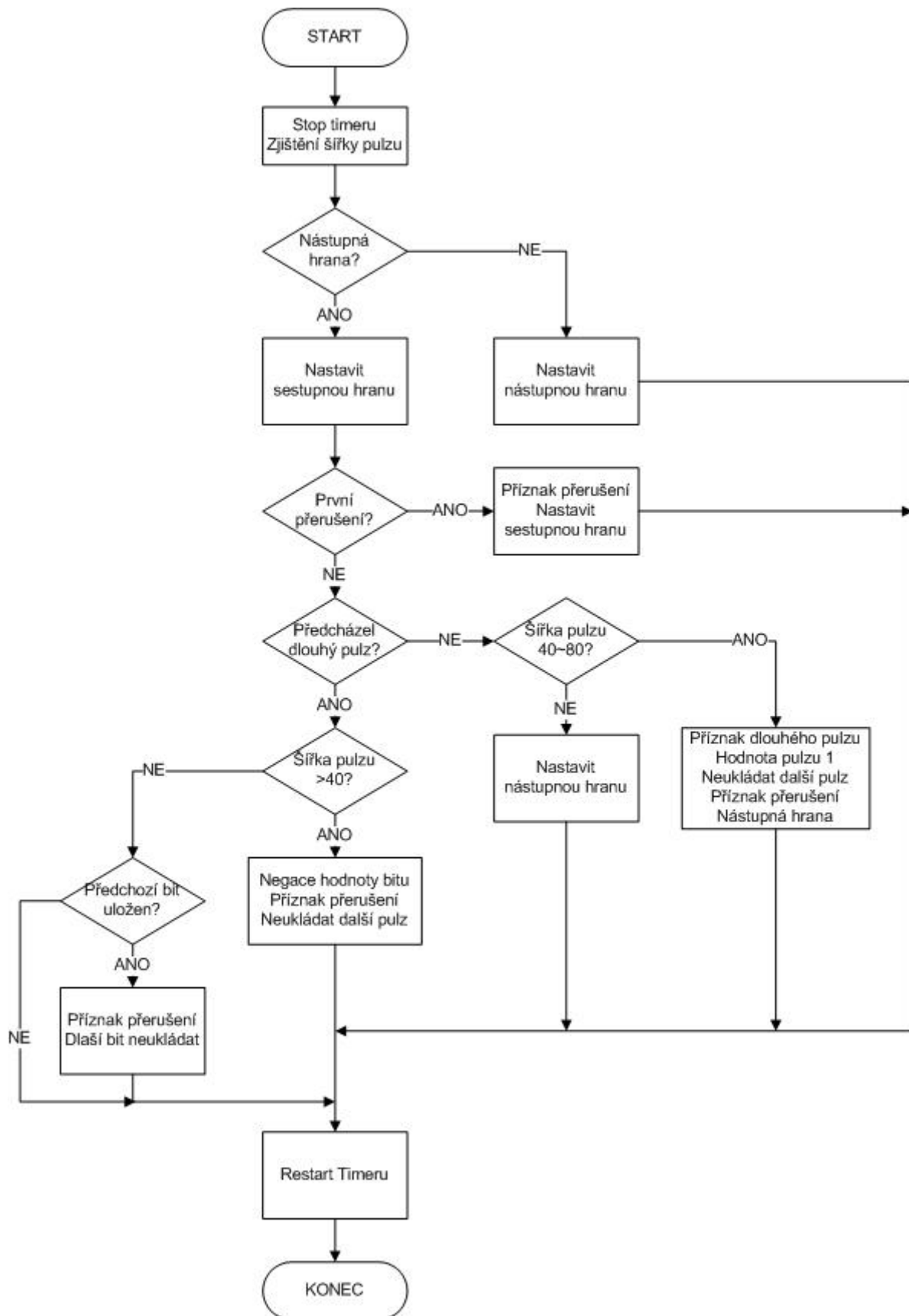
Instrukce	Signál										Popis	čas **	
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0			
Smazat displej	0	0	0	0	0	0	0	0	0	0	1	Smaže displej a nastaví adresu DDRAM na 0.	1.64ms
Návrat na pozici 0	0	0	0	0	0	0	0	0	0	1	*	Nastaví adresu DDRAM na 0. Vráti kurzor na pozici 0 a posune displej zpět do původní pozice. Obsah DDRAM zůstane zachován.	1.64ms
Nastavení módu	0	0	0	0	0	0	0	0	1	I/D	S	Nastaví směr pohybu kurzoru, a specifikuje posun displeje. Operace jsou prováděny během zápisu a čtení dat.	40us
Kontrola displeje zap/vyp	0	0	0	0	0	0	0	1	D	C	B	Zapne/vypne displej (D), zapne/vypne kurzor (C) a blikání kurzoru (B).	40us
Posun kurzoru, displeje	0	0	0	0	0	0	1	S/C	R/L	*	*	Pohyb kurzoru a posun displeje bez změny obsahu DDRAM.	40us
Nastavení funkce	0	0	0	0	1	DL	N	F	*	*	*	Délka rozhraní (DL), počet řádek displeje (N) a velikost fontu (F).	40us
Nastavení adresy CGRAM	0	0	0	1	CGRAM adresa						Nastaví adresu CGRAM. Data jsou přenesena po tomto nastavení.	40us	
Nastavení adresy DDRAM	0	0	1	DDRAM adresa						Nastaví adresu DDRAM. Data jsou přenesena	40us		

					po tomto nastavení.	
Čtení příznaku Busy Flag a adresy	0	1	BF	CGRAM / DDRAM adresa	Čte příznak (BF), který indikuje provádění vnitřních operací a čte adresu CGRAM nebo DDRAM (v závislosti na předchozí instrukci).	0us
Zápis dat do CGRAM nebo DDRAM	1	0		zápis dat	Zapíše data do CGRAM nebo DDRAM.	40us
Čtení dat z CGRAM nebo DDRAM	1	1		čtení dat	Přečte data z CGRAM nebo DDRAM.	40us

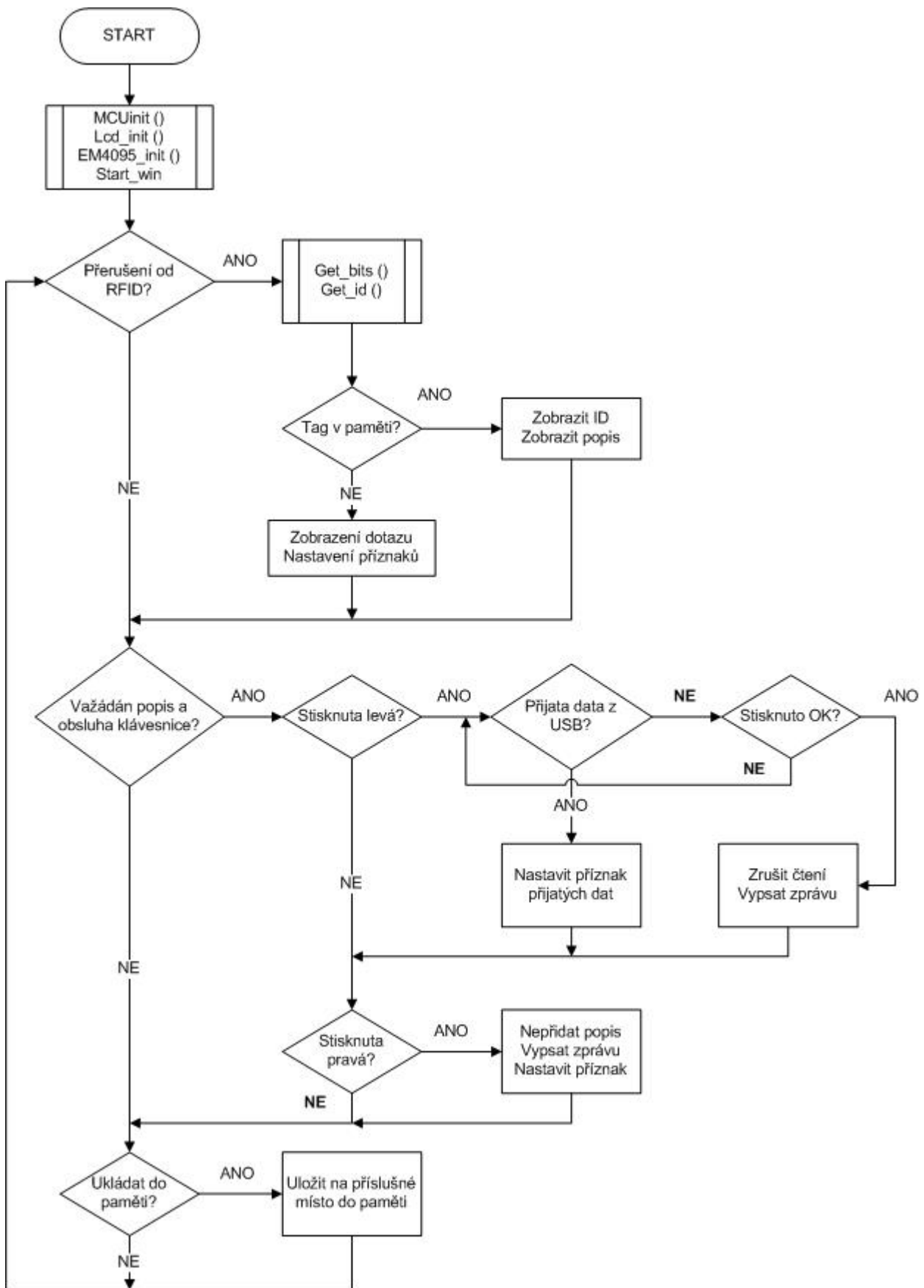
Příloha 3 – vývojové diagramy



Vývojový diagram procedury get_bits

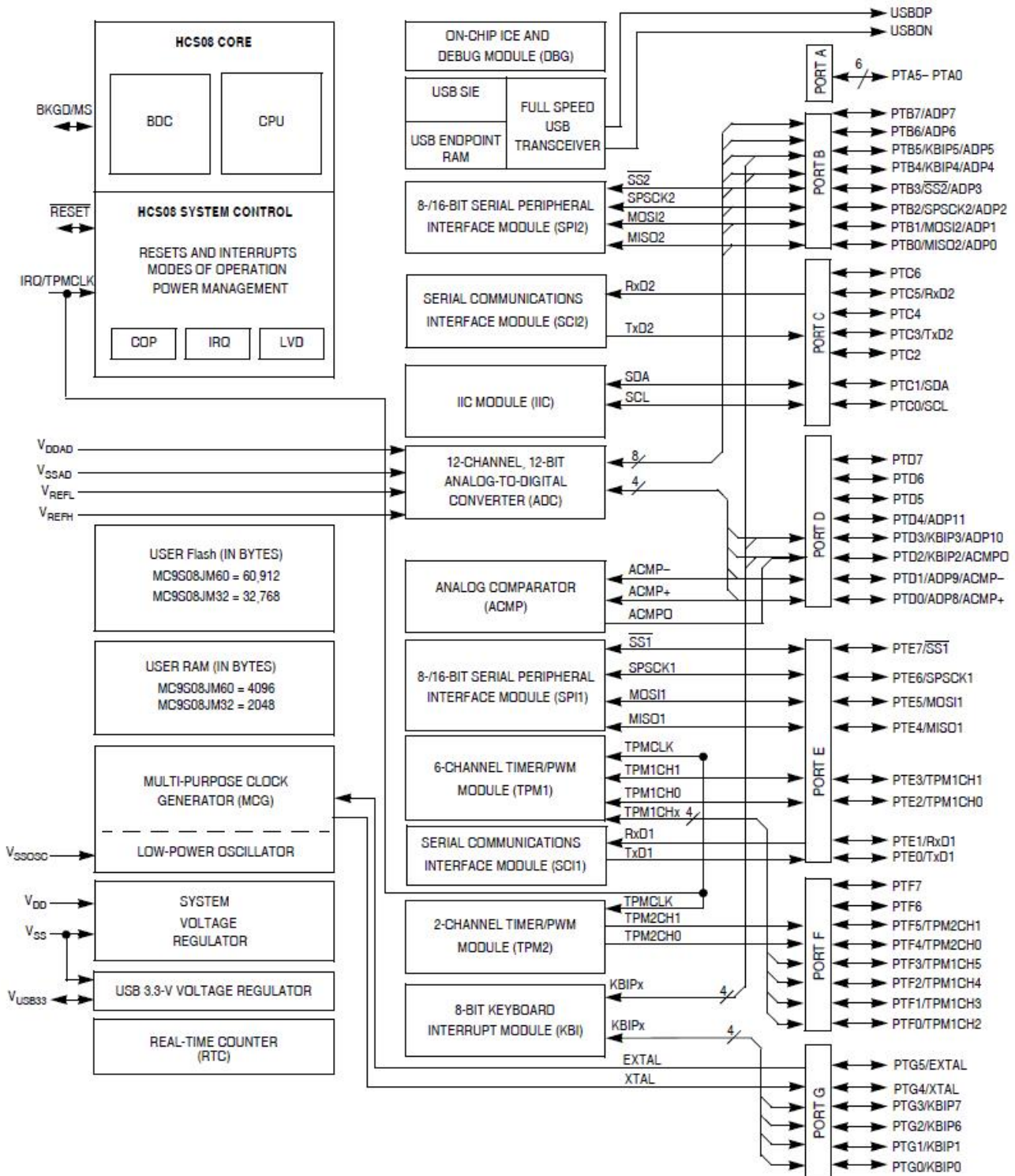


Vývojový diagram podprogramu obsluhy přerušení RFID

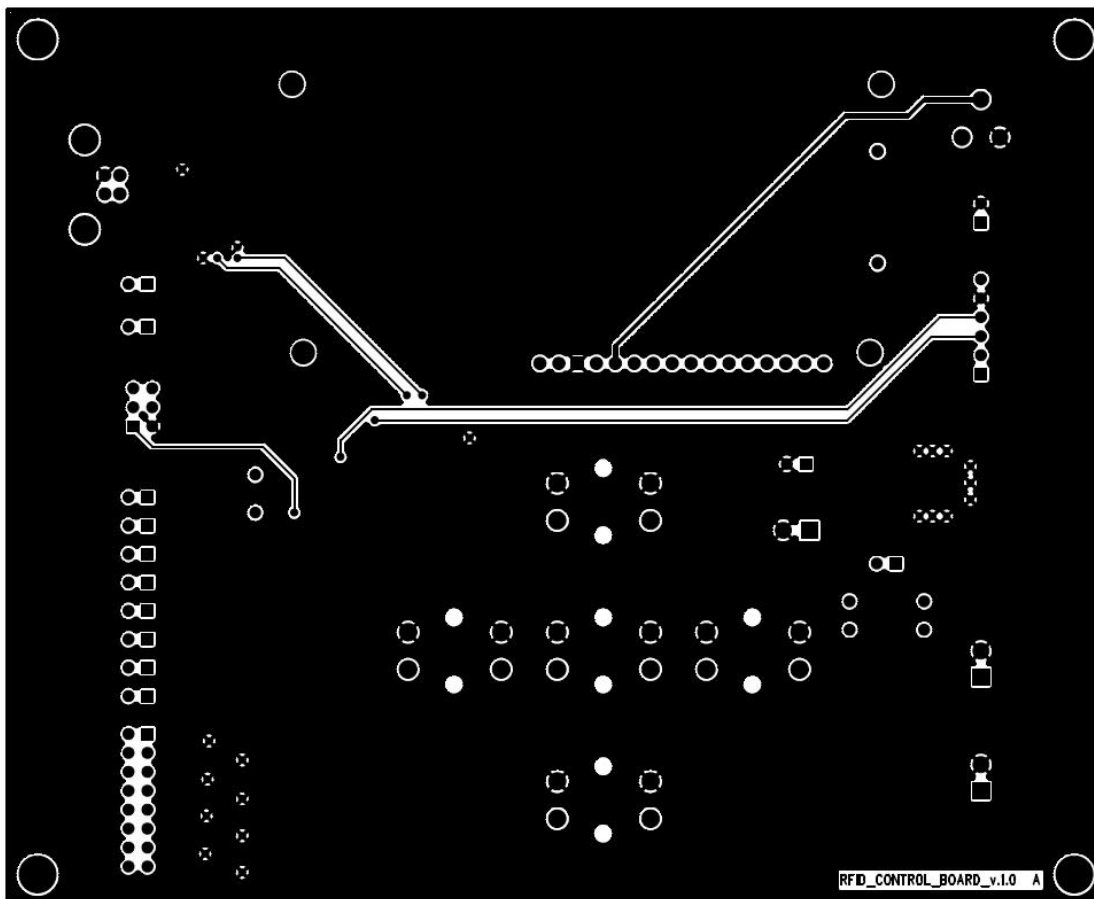


Vývojový diagram hlavní části programu

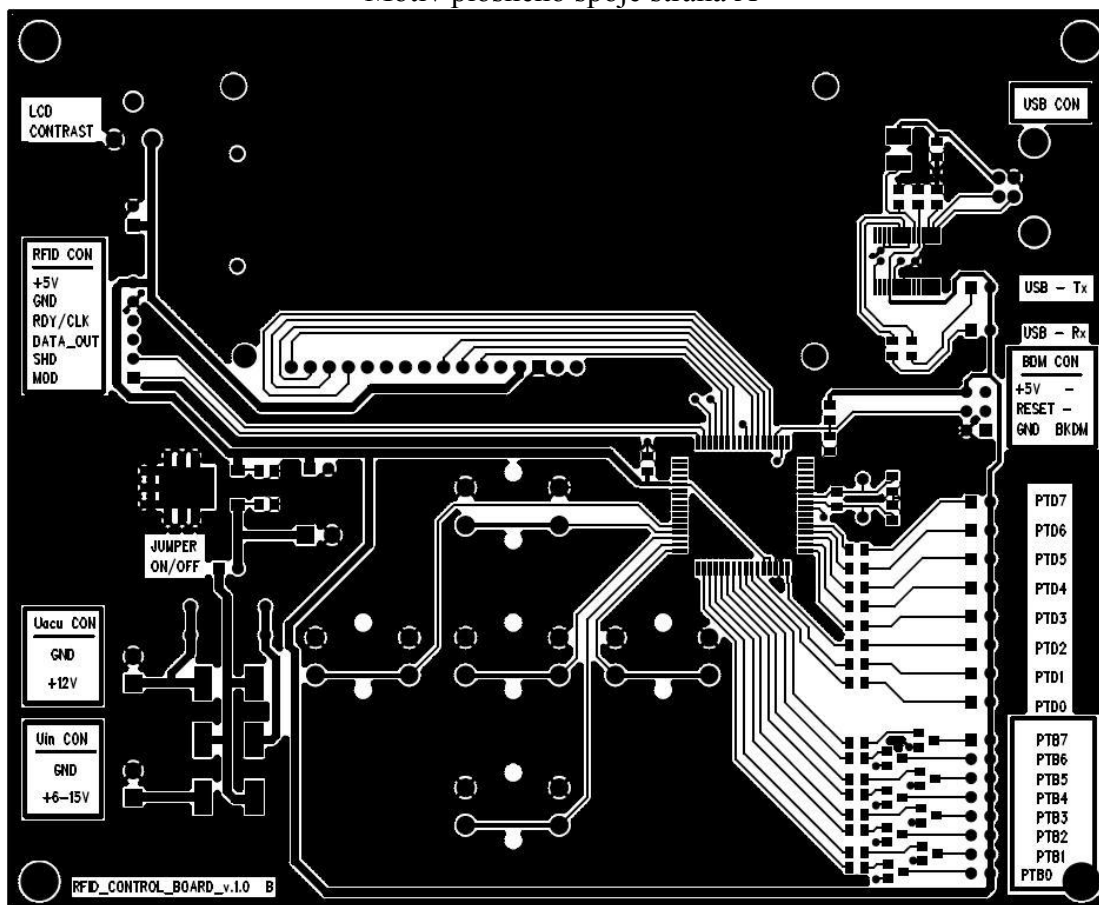
Příloha 4 – blokové schéma mikroprocesoru MC9S08JM32



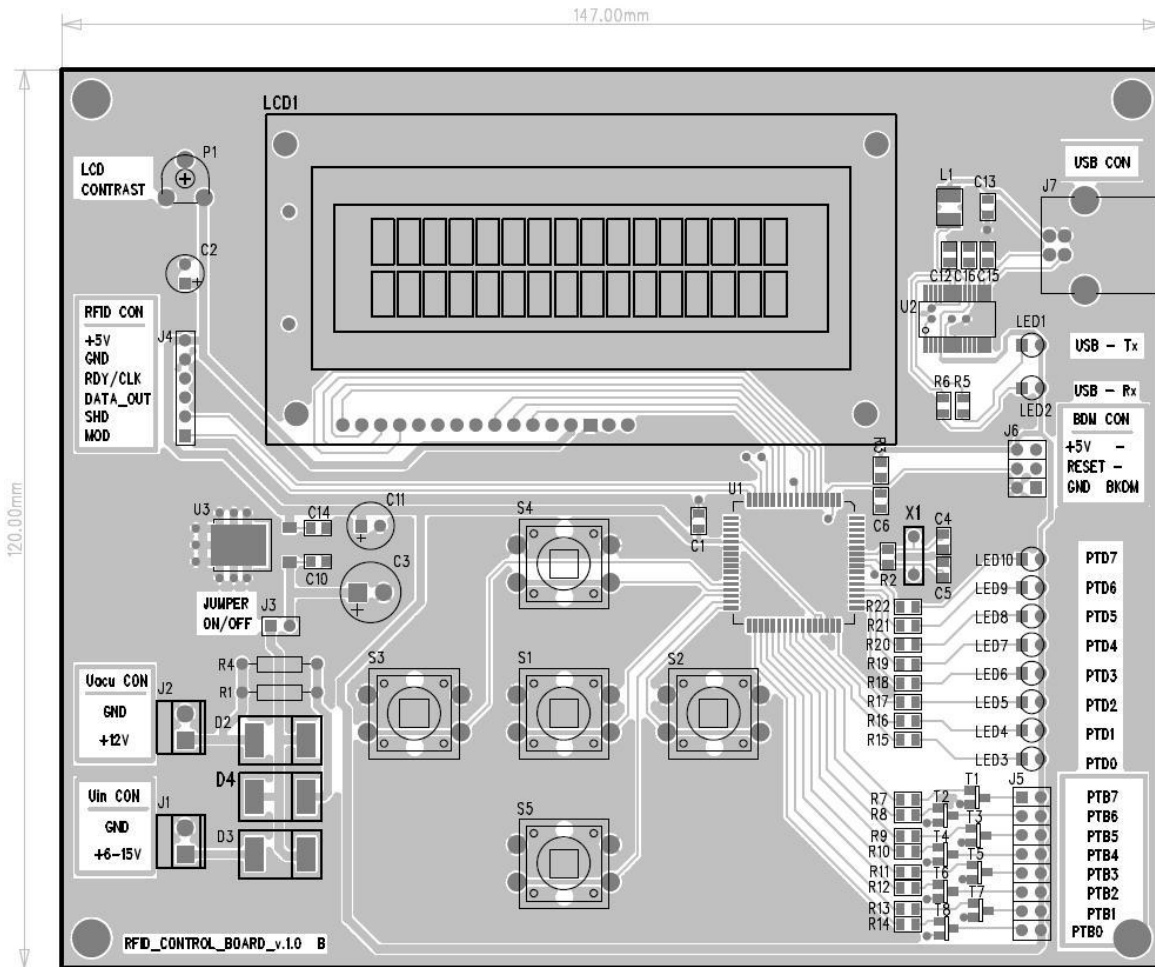
Příloha 5 – DPS řídicí část



Motiv plošného spoje strana A

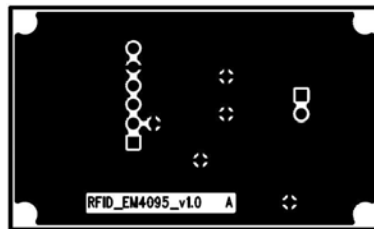


Motiv plošného spoje strana B

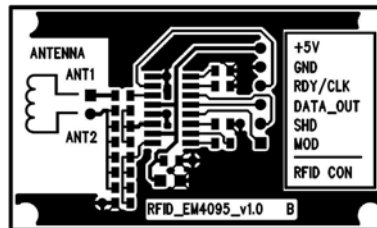


Osazovací plán strana B

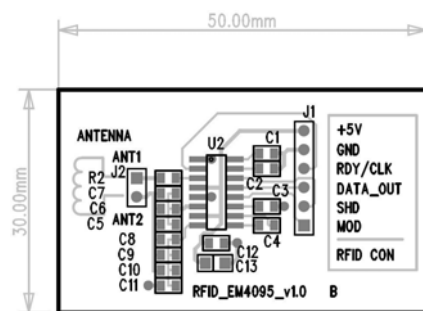
Příloha 6 – DPS RFID část



Motiv plošného spoje strana A

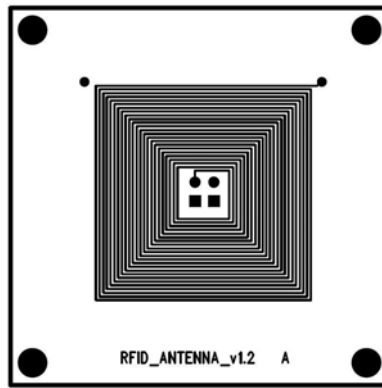


Motiv plošného spoje strana B

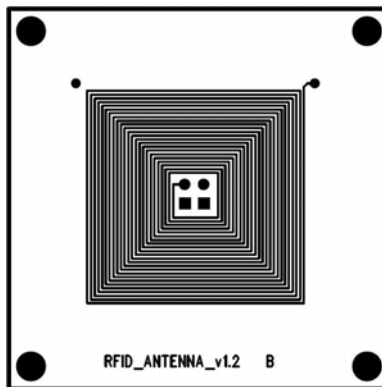


Osazovací plán strana B

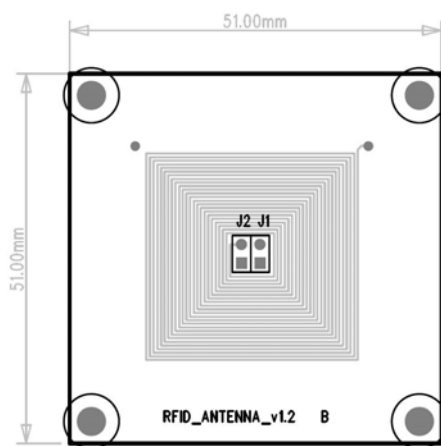
Příloha 7 – DPS anténa



Motiv plošného spoje strana A



Motiv plošného spoje strana B



Osazovací plán strana B

Příloha 8 – foto osazených DPS



Příloha 9 – seznam součástek

Řídící část		RFID část	
J1, J2	ARK/2	J1	S1G06
J3	S1G02	J2	S1G02
J4	S1G06	R1	15Ω
J5	S2G16	C1, C2, C5	10nF
J6	S2G06	C3, C4, C12	100nF
J7	USB	C8	100pF
D2, D3, D4	MBRS340T3	C9	47pF
LED1	zelená	C11	1nF
LED2	červená	C13	4,7μF/10V
LED3, LED4, LED5, LED6, LED7, LED8, LED9, LED10	2mm/2mA		
R1, R4	47Ω		
R2	10MΩ		
R3	4,7kΩ		
R5, R6	220Ω		
R15, R16, R17, R18, R19, R20, R21, R22	220Ω		
R7, R8, R9, R10, R11, R12, R13, R14	1kΩ		
P2	10k		
C1, C6, C10, C12, C14, C15, C16	100nF		
C2	47μF/16V		
C3	220μF/25V		
C4, C5	22pF		
C11	100μF/16V		
C13	10nF		
T1, T2, T3, T4, T5, T6, T7, T8	BC817		
LCD1	MC1602		
U1	MC9S08JM32XFG		
U2	FT232RL		
U3	LF50		
S1, S2, S3, S4, S5	B3F-40xx		
L1	22μH		