

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Zařízení pro sledování docházky.

Petr Janošík

Bakalářská práce
2012

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr Janošík**
Osobní číslo: **I08228**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**
Název tématu: **Zařízení pro sledování docházky**
Zadávající katedra: **Katedra elektrotechniky**

Z á s a d y p r o v y p r a c o v á n í :

Navrhňte zařízení pro sledování docházky. Požadavkem je, aby celý systém tvořilo jediné zařízení připojené do sítě typu ethernet. Součástí zařízení bude čtečka RFID, datové uložení i záložní zdroj energie. Přístup k datům bude přes webové rozhraní.

Teoretická část práce bude věnována technologiím a standardům RFID.

Praktická část bude zaměřena na návrh koncepce zařízení a vytvoření jeho softwarového vybavení.

Pro ověření funkčnosti využijte vývojový kit mikrokontroléru s připojenou čtečkou RFID.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Stellaris ARM Cortex-M3 [online]. 2007-2010 [cit. 2011-10-21]. Dostupné z WWW: [url: <http://www.ti.com/tool/eks-lm3s6965>].

Kabelová A., Dostálek L.: Velký průvodce protokoly TCP/IP a systémem DNS, Computer Press, Praha 2008, ISBN: 978-80-251-2236-5

ARM The Architecture For The Digital World [online]. Version 2.0. 2010-09-12 [cit. 2011-07-29]. CMSIS - Cortex Microcontroller Software Interface Standard.

Dostupné z WWW: [url: <http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php>]. lwIP - A Lightweight

TCP/IP stack - Summary Technical Report T2001:20. Dostupné z WWW: [<http://savannah.nongnu.org/projects/lwip/>].

Katalogové listy, aplikační poznámky a další dokumentace k použitým součástkám, dostupné online u výrobců příslušných prvků.

Vedoucí bakalářské práce:

Ing. Zdeněk Němec, Ph.D.

Katedra elektrotechniky

Datum zadání bakalářské práce: **12. prosince 2011**

Termín odevzdání bakalářské práce: **11. května 2012**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Zdeněk Němec, Ph.D.
vedoucí katedry

V Pardubicích dne 30. března 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 3. 9. 2012

Petr Janošík

Poděkování

Tímto bych rád poděkoval vedoucímu práce panu Ing. Zdeňku Němcovi, Ph.D. za odborné vedení v průběhu práce.

Dále bych rád poděkoval panu Ing. Martinu Lauterbachovi, který mě v průběhu celé práce poskytoval technickou pomoc a odborné rady.

Anotace

Cílem této práce je návrh zařízení pro evidování docházky pomocí RFID karet s čipem EM4102. Navrhované zařízení představuje kompletní systém obsahující čtečku RFID, mikrokontroler a uložení dat. Uložená data jsou přístupná pomocí sítě ethernet a webového rozhraní.

Klíčová slova

RFID, tag, reader, ethernet, AM modulace, RTOS

Title

RFID reader.

Annotation

The aim of this work is to design equipment for recording attendance with RFID card with chip EM4102. The proposed facility is a complete system including RFID, microcontroller and data storage. Stored data can be accessed via Ethernet and the Web interface.

Keywords

RFID, tag, reader, ethernet, ASK modulation, RTOS

Obsah

Seznam zkratk	8
Seznam obrázků	9
Seznam tabulek	9
1 Úvod	10
2 Systémy RFID	11
2.1 Princip komunikace.....	11
2.2 RFID tag.....	11
2.3 Identifikace tagu.....	12
2.4 Vnitřní paměť tagu.....	12
2.5 Frekvenční pásma.....	13
3 Použitý systém s EM4095	15
3.1 Kódování.....	16
3.2 EM4100.....	16
3.3 Plošný spoj.....	17
4 Použitý vývojový kit	21
4.1 Vývojové prostředí.....	22
4.2 Driverlib.....	22
5 FreeRTOS	23
5.1 Embedded systém.....	23
5.2 Embedded OS.....	24
5.3 Stavba programu v FreeRTOS.....	24
6 LWIP	25
6.1 ISO OSI.....	25
6.2 TCP/IP.....	27
6.2 Ethernet driver.....	27
7 Server HTTP	28
7.1 Funkce HTTP.....	28
7.2 HTTP zpráva.....	28
7.3 SSI a CGI.....	29
7.4 Uložiště stránek.....	29

8	Souborový systém na kartě SD.....	30
8.1	MicroSD	30
8.2	Rozhraní	31
8.3	Knihovna FatFS	31
9	Zpracování signálu z čtečky RFID.....	33
9.1	Časování	33
9.2	Přerušení	33
9.3	Blokový diagram	34
10	Výsledná aplikace	35
	Literatura	37
	Příloha A.....	39

Seznam zkratek

ARM	Advanced RISC Machine
TCP/IP	Transmission Control Protocol / Internet Protocol
CW	Continuous Wave
ASK	Amplitude shift keying
SOF	Start of frame pattern
EOF	End of frame pattern
NRZ	Non-return-to-zero
CMOS	Complementary Metal–Oxide–Semiconductor
LWIP	lightweight IP
RTOS	Real-time operating system
RFID	Radio Frequency Identification
OS	Operating system
SSI	Server Side Includes
CGI	Common Gateway Interface
MCU	Microcontroller

Seznam obrázků

Obrázek 1 – RFID tag.....	11
Obrázek 2 – Princip komunikace EM4095 s EM4100	15
Obrázek 3 – Balík dat ze čtečky EM4095	16
Obrázek 4 – Schéma zapojení EM4095	17
Obrázek 5 – Schéma v Pulsonixu	18
Obrázek 6 – Plošný spoj EM4095	19
Obrázek 7 – EKS-LM3S6965	21
Obrázek 8 – ISO OSI.....	25
Obrázek 9 – Blokové schéma MicroSD	30
Obrázek 10 – FAT File System Modul	31
Obrázek 11– Vývojový diagram zpracování signálu ze čtečky	34
Obrázek 12– Blokové schéma výsledné aplikace.....	35

Seznam tabulek

Tabulka 1 – Frekvenční pásma RFID.....	13
--	----

1 Úvod

Zkratkou RFID se značí Radio-Frekvenční Identifikace. Což je bezkontaktní identifikace sloužící k přenosu a ukládání dat pomocí elektromagnetických vln. Tuto technologii lze nasadit ve skoro všech odvětvích průmyslu. Jako příklady lze uvést třeba zabezpečení zboží v obchodních domech proti odcizení nebo třeba rozeznávání jednotlivých produktů ve velkých továrnách, kde už není možná vizuální kontrola. V podstatě lze říci, že nasazení RFID zkvalitňuje a hlavně zrychluje celý proces identifikace věcí, zvířat nebo i osob.

Úkolem této bakalářské práce je vytvoření zařízení pro sledování docházky. Příchozí osoba by tedy měla přiložit identifikační kartu, uvnitř které se nachází čip typu EM4100. ID čísla karet by měl být požadovaný systém schopen poskytovat po ethernetové síti formou HTTP serveru. Zároveň by měl být schopen tyto ID sám uchovávat, proto je nutné, aby zařízení vlastnilo nějaký typ úložiště.

Náš úkol se dělí na dvě části. V první části bude potřeba vytvořit program pro zpracování dat ve směru od čtečky k procesoru. Tyto data by měl dále dekodovat a připravit k požití v druhé části úkolu – vytvoření webového serveru. Používat se bude ethernetový protokol TCP/IP. Dále je už potřeba vytvořit webová stránka pro uživatele.

Bude třeba nalézt ten správný procesor, který by se pro tento ne zrovna jednoduchý úkol hodil. Dále je pak potřeba nalézt tu správnou čtečku, která bude kompatibilní s EM4100 a bude vracet k procesoru co do jednoduchosti nejvýhodnější signál.

2 Systémy RFID

Následující kapitola vychází především z literatury [2] a [11]. První část systému je čip, který je tvořen elektronickým paměťovým obvodem s anténou. Čip je umístěn na nějakém podkladu. Tento podklad je většinou tvořen papírem nebo umělou hmotou. V některých případech může být čip opatřen i baterií. Odborně se tato část nazývá RFID tag nebo RFID transpondér. Druhá část tvoří zařízení, které RFID tag čte. Toto zařízení obsahuje čtecí, vysílací elektronické obvody a anténu. Toto zařízení odborně nazýváme RFID reader neboli RFID transceiver.

2.1 Princip komunikace

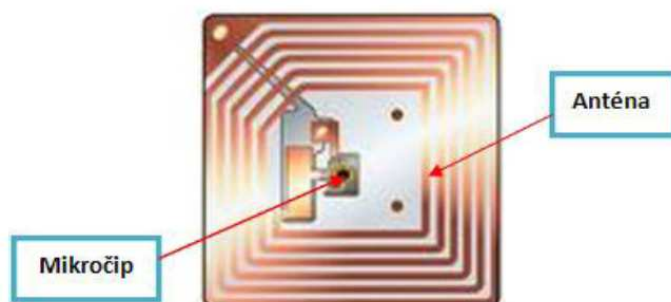
U systému RFID se rozeznávají dva fyzikální způsoby komunikace:

První metoda se nazývá Odrazová. Funguje na podobném principu jako radar. Čtečka vysílá k tagu vysokofrekvenční signál, který tag nejdříve využije k nabití svého čipu, který poté mění parametry antény. Tímto způsobem se v odraženém signálu nachází zakódovaná informace.

Druhá metoda se nazývá induktivní. Jak už název napovídá, pracuje na principu vzájemné indukci dvou cívek (anténa tagu a anténa transceiveru).

2.2 RFID tag

Čip, který se nachází na tagu už v dnešní době může mít vskutku malé rozměry v řádech mm nebo i dokonce stovek um. Problematickou věcí z hlediska velikosti tagu je tedy anténa, která udává vzhled tagu. Většinou ovšem platí, že čím je vyšší použitá frekvence, tím může být anténa tagu menší.



Obrázek 1 – RFID tag[6]

RFID tag dělíme na aktivní a pasivní.

2.2.1 Pasivní tag

Tyto tagy nemají žádný vlastní zdroj energie, díky čemuž jsou mnohem jednodušší na údržbu, dají se lépe ochránit proti nepříznivým podmínkám jako např. vlhkost, vysoká teplota nebo mráz. Ovšem mají i své nevýhody v podobě několikanásobně kratší čtecí vzdálenosti oproti aktivním tagům.

2.2.2 Aktivní tag

Za předpokladu, že je potřeba dosáhnout většího dosahu a to pomocí stejné frekvence se používají u RFID komunikace aktivní tagy. Tyto tagy mají vlastní zdroj napájení, který využijí k napájení čipu nebo zesílení signálu, který zrovna odesílají transceiveru. Odeslaný signál, tak může mít až mnohonásobně vyšší dosah.

2.3 Identifikace tagu

Ona nejdůležitější vlastnost každého tagu, je ta, že by měl být jednoznačně identifikovatelný. Proto je v každém tagu uloženo číslo, které je odesíláno v případě, že k tomu dá nějaká čtečka podmět. Toto číslo má podobu 64 bitů, u kterých je ale použitelných pouhých 40. Ostatní bity zajišťují bezchybný přenos. Po finálním dekodování má toto číslo podobu 10 hexadecimálních znaků.

2.4 Vnitřní paměť tagu

Tag může mít v některých případech nejen uložené ID číslo, ale může obsahovat i paměť. Nebo nemusí být naprogramovaný vůbec. A dotyčný si ho může naprogramovat pro danou aplikaci sám. Tagy, které obsahují paměť (která se pohybuje většinou okolo 40 až 512 bitů) se přitom dělí do třech základních kategorií:

2.4.1 Tagy RO (Read-Only)

Tyto tagy jsou specifické tím, že mají v paměti uloženo číslo, které už by nemělo být možno změnit. Tyto tagy jsou obdobou čárkových kódů ale je diskutabilní jestli se jim je někdy podaří nahradit (ekonomické důvody). Tyto tagy jsou rovnou naprogramovány v továrně.

2.4.2 Tagy WORM (Write Once Read Many)

Jsou tagy pouze pro čtení. Výrobce je ale nenaprogramuje přímo v továrně, ale dotyčný si je naprogramuje podle potřeb a poté už je tag možné jen číst ale nikoliv na něj zapisovat. Na trhu se lze setkat s těmito tagy, u kterých výrobce tvrdí, že opakované přepsání paměti je možné, avšak na to nedává záruku. Realizovány jsou většinou pomocí paměti **EEPROM**.

2.4.3 Tagy RW (Read Write)

Tyto tagy je možné jak číst, tak také na ně zapisovat. Paměť je adresovatelná a snadno se přepisuje. Mívají i mnohem větší paměť než ostatní typy, dosahující až 8 Kbit.

2.5 Frekvenční pásma

Frekvence je u systému RFID jedna ze stěžejních veličin. Při jejím výběru se musí postupovat obezřetně, jelikož frekvencí se dá ovlivnit mnoho fyzikálních omezení, jako jsou například rychlost přenosu dat, vzdálenost pro čtení/zápis, omezení vyzářené energie a jiné. U frekvence ale také platí, že čím větší je, tím větší může být vzdálenost mezi RFID čtečkou a tagem. Ovšem radiové vlny na vyšších frekvencích jsou mnohem více náchylné na různé nedokonalosti prostředí a problémové materiály jako jsou kapaliny, uhlík a samozřejmě kovy. Záleží tedy na vlnové délce - $\lambda = \frac{c}{f}$.

Pásma se dají rozdělit na dvě části. První by byla do desítek MHz (v této oblasti tagy fungují na induktivním principu) a druhá metoda od desítek MHz výše (zde fungují tagy na principu odrazovém).

Ovšem běžné dělení frekvenčního spektra vypadá takto:

Tabulka 2 – Frekvenční pásma[6]

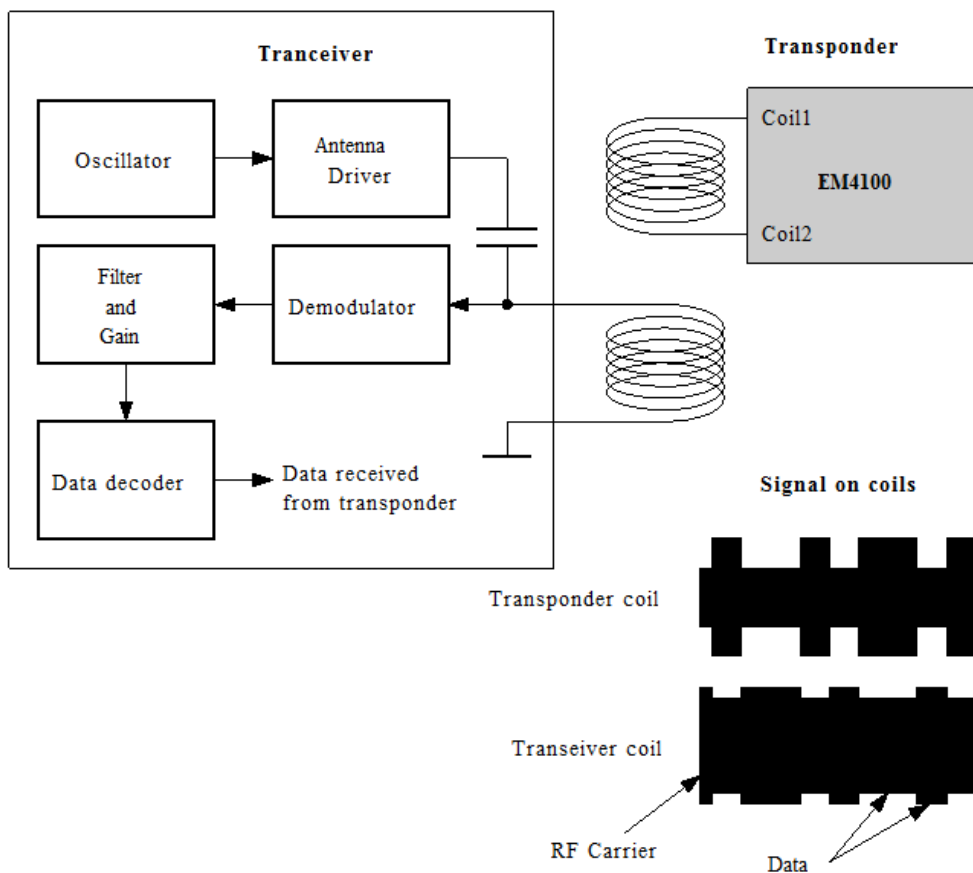
Frekvenční pásmo	Anténa	Rychlost a množství dat	Čtecí vzdálenost	Využití	+ Výhody / - Nevýhody
<u>Nízkofrekvenční (LF)</u> <u>125 a 135 KHz</u>	Indukční cívka na feritovém jádře	- malá rychlost čtení - malé množství dat	Krátká až střední do 0,5 m	- kontrola přístupu - identifikace a sledování zvířat - imobilizéry automobilů - inventura - identifikace kovových produktů (např. pivních kegů)	+ větší odolnost proti rušení + možnost upevnění v blízkosti vody (tekutiny), vlhkých prostředích + možnost upevnění na kovové podložce - malá komunikační rychlost - velká anténa
<u>Vysokofrekvenční (HF)</u> <u>13.56 MHz</u>	Indukční cívka rovná (3-5 návinů)	- střední rychlost čtení - malé až střední množství dat	Krátká: do 1 m	- chytré karty (Smart Cards) - bezkontaktní placení - chytré etikety (Smart Labels)	+ menší rozměry antény + větší komunikační rychlost než LF + větší čtecí dosah než LF - kovové podložky a voda již významně snižují čtecí dosah a ruší komunikaci

Frekvenční pásmo	Anténa	Rychlost a množství dat	Čtecí vzdálenost	Využití	+ Výhody / - Nevýhody
Ultrafrekvenční (UHF) 860 až 960 MHz (Evropa 868 MHz, USA a Canada 915 MHz)	Samostatná nebo duální dipólová anténa	<ul style="list-style-type: none"> - vysoká rychlost čtení - malé až střední množství dat 	Střední: do cca 3 m	<ul style="list-style-type: none"> - sledování palet při přepravě a ve skladech - současná identifikace více zabalených produktů - elektronické mýtné - parkovací karty - sledování toku vratných obalů - protokoly: ISO 18000-6A/B, EPC Class 0/ 	<ul style="list-style-type: none"> + možnost i vzdáleného čtení = identifikace průjezdem brány + velká přenosová rychlost = možná větší kapacita paměti RFID tagu + dipólová anténa + levná výroba + Anti-kolize 1500 tagů/sekundu - nečitelnost přes kapaliny - obtížné čtení na kovových podložkách - celosvětově nejednotná frekvence
Mikrovlnná 2.45 a 5.8 GHz	Single dipólová anténa	<ul style="list-style-type: none"> - vysoká rychlost čtení - střední množství dat 	Střední: do 2 m	<ul style="list-style-type: none"> elektronické mýtné - identifikace zavazadel při letecké přepravě - bezdrátový záznam a přenos dat v reálném čase 	<ul style="list-style-type: none"> + vysoká přenosová rychlost až 2 Mb/s + malé rozměry dipólové antény = malé tagy + Anti-kolize 50tagů/sekundu - drahá a složitá konstrukce - menší dosah než UHF RFID - velký vliv rušení (kovu, kapalin apod.)

3 Použitý systém s EM4095

Tato kapitola většinou čerpá z [15]. EM4095 je čtecí čip typu CMOS, který se používá v 125 kHz RFID aplikacích. Obsahuje kompatibilitu přenosového protokolu s mnoha tagy včetně EM4102, který je požadován pro naši aplikaci. Čip má rozsah pracovní frekvence od 100MHz do 150MHz. Přenos dat je zde zajištěn pomocí amplitudové modulace.

Ke správné funkci této čtečky je potřeba jen mikroprocesor, anténa a ještě pár elektronických součástek ke stabilizaci a úpravě průběhů na určitých pinech.

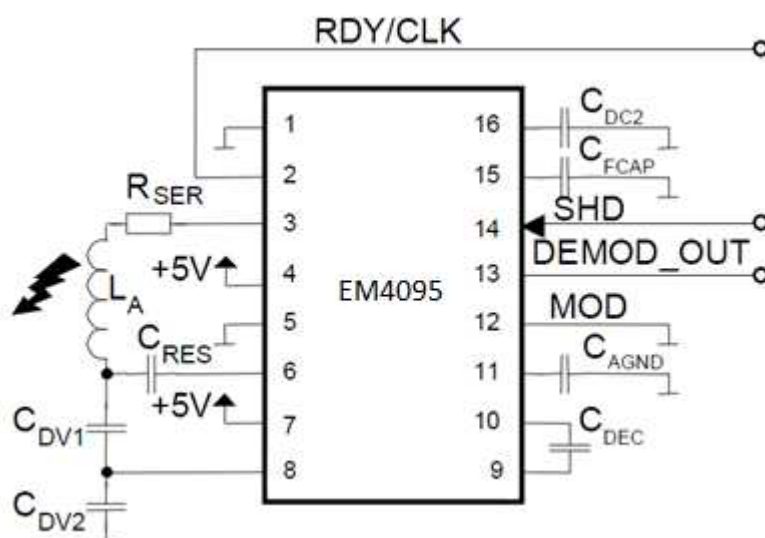


Obrázek 2 – Princip komunikace EM4095 s EM4100[15]

3.3 Plošný spoj

Pro tento návrh byl použit program PULSONIX.

Pro dobrou stabilizaci napájecího napětí je zapotřebí dát na svorky napájecího napětí kondenzátor s velkou kapacitou. Pro splnění této funkce se zdál kondenzátor 470uF/16V plně dostačující. Zapojení ostatních součástek je zobrazeno na schématu zapojení čtečky EM4095 (obrázek 4).



Obrázek 4 – Schéma zapojení EM4095[15]

Kde hodnoty součástek jsou následující: CDC2 a CFCAP mají shodnou velikost 10nF. CAGND a CDEC mají velikost 100 nF. Hodnotám ostatních součástek se budu věnovat v další části textu, jelikož úzce souvisí s anténou.

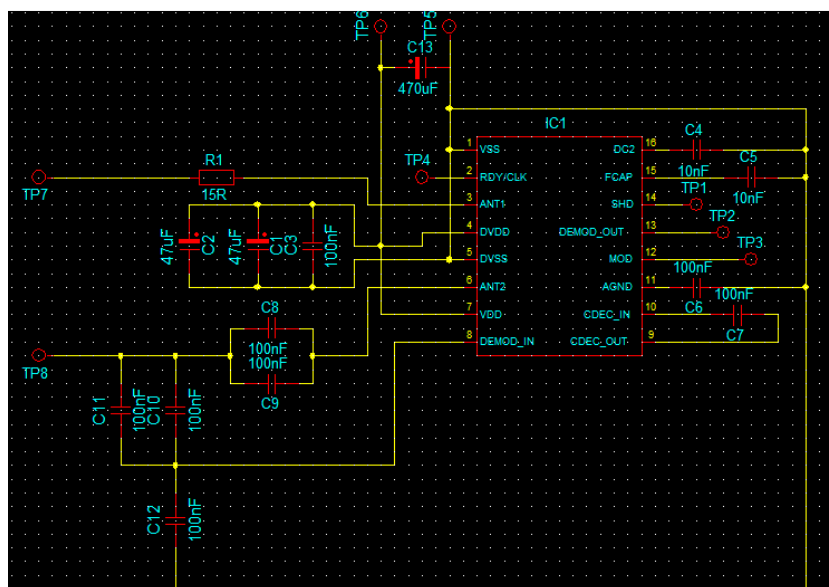
Z hlediska vstupů a výstupů jsou pro nás zajímavé piny RDY/CLK, SHD, DEMOD_OUT a svorky antény.

- RDY/CLK - Hodinový signál z důvodů synchronizace čtečky a procesoru.
- SHD - Při přivedení napájecího napětí na tento pin se zařízení uvede do režimu spánku, kde odběr čtečky karet se přibližuje jen 1uA.
- DEMOD_OUT - Výstup kde vychází demodulovaný signál zakódovaný v Manchesteru.

V mojí aplikaci jsem se rozhodl použít jen pin DEMOD_OUT. Synchronizační signál by mělo být možné programově obejít a uvádění čtečky do režimu spánku jsem se rozhodl nepoužít.

Nyní je potřeba navrhnout plošný spoj v nějakém programu pro návrh plošných spojů. Zde je jako příklad použit program PULSONIX.

Takto jsou spojeny jednotlivé součástky v Pulsonixu:

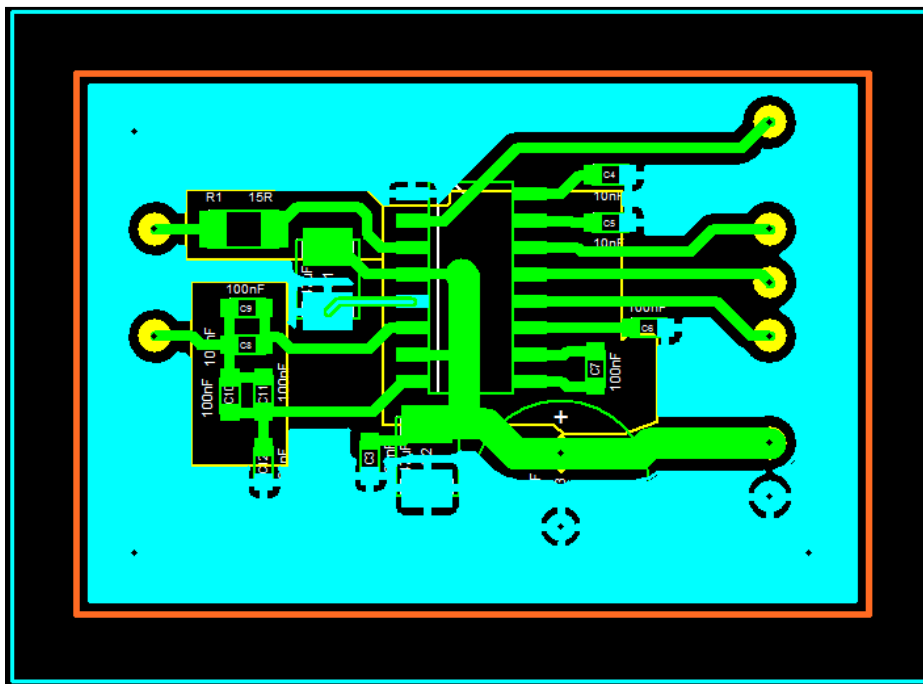


Obrázek 5 – Schéma v Pulsonixu

Z obrázku 5 je vidět že zapojení je velice podobné jako to, které je na obrázku 4.

Je také zřejmé, že hodnoty mnoha součástek jsem musel dosadit jen hodně přibližně, jelikož se teprve budou dopočítávat.

Na dalším obrázku už je vidět vzhled plošného spoje.



Obrázek 6 – Plošný spoj EM4095

Nyní je zapotřebí zjistit hodnoty zbývajících součástek a namotat anténu. Zjištění součástek se provádí pomocí výpočtu:

$$R_{AD} = 3\Omega \quad V_{SS} - V_{DD} = 5V$$

Operační frekvence čtečky je $125\text{kHz} = f_0$. R_{SER} byla zvolena 22Ω .

Indukčnost cívky antény:

Namotaná anténa má 40 závitů. Indukčnost antény se zjistí pomocí měřiče indukčnosti cívek. Po změření cívky je tedy indukčnost $L_A = 258\text{uH}$.

Po změření odporu antény se zjistí že $R_{ANT} = 5,3\Omega$.

Činitel jakosti se poté dá spočítat podle vzorce: $\frac{2 \cdot \pi \cdot f_0 \cdot L_A}{R_{ANT}} = Q_A = 38,23$

Přes následující výpočet se dostaneme k hodnotě první součástky:

$$C_{RES} = \frac{1}{(2 \cdot \pi \cdot f_0)^2 \cdot L_A} = 6,28\text{nF}$$

Výpočet proudu anténou (ve špičce): $I_{ANT} = \frac{4 \cdot (V_{DD} - V_{SS})}{\pi \cdot (R_{ANT} + R_{SER} + 2 \cdot R_{AD})} = 191 \text{mA}$

Výpočet napětí na anténě (ve špičce): $V_{ANT} = \frac{I_{ANT}}{2 \cdot \pi \cdot f_0 \cdot C_{RES}} = 38,7 \text{V}$

Výpočet reálné rezonanční frekvence:

Nejprve je potřeba sečíst kapacita což je C_{DV1} sériově s C_{DV2} a tyto dva kondenzátory poté

paralelně s C_{RES} takže: $C_0 = \frac{C_{DV1} \cdot C_{DV2}}{C_{DV1} + C_{DV2}} + C_{RES}$

Celková kapacita může být potom dosazena do vzorce pro rezonanční frekvenci:

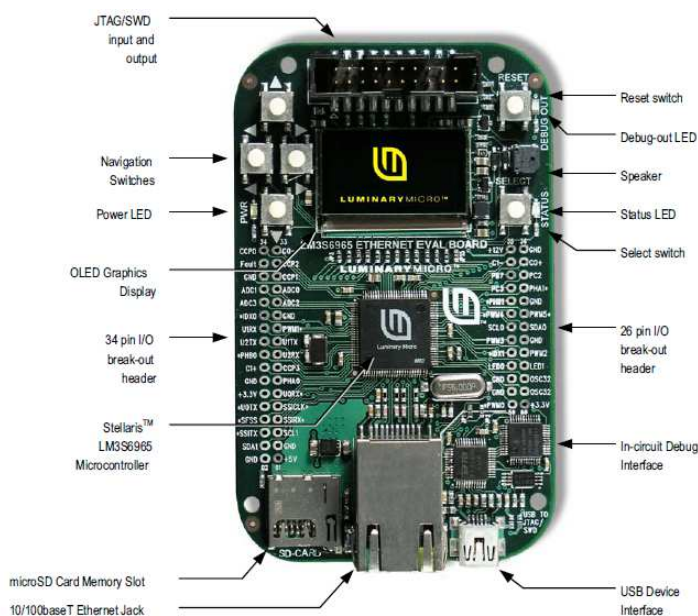
$f_0 = \frac{1}{2 \cdot \pi \cdot \sqrt{L_A \cdot C_0}}$ Za dodržení těchto podmínek by měla čtečka pracovat na rezonanční frekvenci.

4 Použitý vývojový kit

Tato kapitola většinou čerpá z [1]. Jedná se o vývojový kit EKS-LM3S6965, který má velmi široké spektrum uplatnění v různých aplikacích.

Uprostřed kitu se nachází procesor LM3S6965

Jedná se o 32-bitový RICS procesor postavený na architektuře ARM® Cortex™-M3 v7M který má 256 KB paměti typu Flash a 64 KB paměti typu SRAM. Maximální operační frekvence tohoto procesoru je 50 MHz. K dispozici tu jsou čtyři 32-bitové časovače, vestavěná MAC, tři programovatelné UARTy typu 16C550, dva nezávislé analogové komparátory, dva I2C moduly, tři PWM bloky s 16-bitovým čítačem a dvěma analogovými komparátory schopnými generovat dva nezávislé PWM signály, 42 vstupů a výstupů (v závislosti na naší konfiguraci).



Obrázek 7 – EKS-LM3S6965[1]

Funkce, které vývojový kit má:

- Integrovaný grafický OLED display s rozlišením 128 x 96.
- Integrovaný 10/100 ethernetový kontroler.
- Tlačítka a diody.
- USB rozhraní pro odladění projektů a napájení procesoru.
- Volné výstupní nebo vstupní piny propouští ve vlastní aplikaci.
- Slot pro paměťovou kartu typu microSD.
- Magnetický reproduktor.

4.1 Vývojové prostředí

Společně s vývojovým kitem EKS-LM3S6965 je dodáváno také vývojové prostředí Code Composer Studio, jehož jádro je postaveno na vývojovém prostředí Eclipse, které je poskytováno zdarma. Toto vývojové studio je přiloženo na DVD ve verzi 4. Je však možné ho v současné době aktualizovat na verzi 5.

4.2 Driverlib

[1] Stellaris Peripheral Driver Library je sada ovladačů pro přístup k perifériím které se nachází na mikroprocesorech Stellaris ARM® Cortex™-M3.

Ovladače mají následující vlastnosti:

- Jsou snadno pochopitelné.
- Snaží se být přiměřeně efektivní, pokud jde o paměť a využití procesoru.
- Jsou psány výhradně v C s výjimečnými případy, kdy to není možné.
- Poskytují mechanismus, který umožňuje snadné použití daného zařízení na určité periférii.
- Ovladače jsou soběstačné, jak je to jen možné.

Pro mnohé aplikace, mohou být ponechány ovladače tak, jak jsou. Ale v některých případech, musejí být ovladače upraveny nebo přepsány tak, aby mohly plnit svou funkci. V takovém případě je možné použít stávající ovladač jako návod, jak ovládat periférie.

Nyní následuje popis jednotlivých položek v Driverlibu:

- | | |
|---------------|--|
| ewarm/- | Tento adresář obsahuje zdrojové soubory, které jsou specifické pro Embedded IAR nástroje. |
| gcc/ - | Tento adresář obsahuje zdrojové soubory, které jsou specifické pro nástroje GNU. |
| hw_*. H - | Hlavičkové soubory, na každou periférii, které popisují všechny registry a bitové pole v rámci uvedených registrů pro danou periférii. |
| src/ - | Tento adresář obsahuje zdrojové kódy pro ovladače. |
| third_party/- | Tento adresář obsahuje third party software, který byl portován na Stellaris mikrokontroléry. |

5 FreeRTOS

Informace z následující kapitoly pocházejí většinou z [9] a [12]. Operační systémy reálného času musí garantovat jisté mezní časy, které by se neměly překročit. Těmto časům se říká deadlines. Na rozdíl od operačních systémů, které se používají na osobních počítačích (např. Windows) musí RTOS dosáhnutí těchto časů zaručit.

- Real-time systém je systém, který reaguje předvídatelným způsobem na nepředvídatelné externí události.
- Pokud lze dokázat, že realtime systém splní svá ultimáta (deadlines) a to za použití chování systému v nejhorším možném případě, nikoliv analýzou průměrného chování systému, potom můžeme říci, že chování systému je předvídatelné.

Operační systémy reálného času se dále dělí na HRTOS a SRTOS.

HRTOS - Hard real time OS (tvrdý real time OS)

- Operační systém používaný v kritických aplikacích, ve kterých je neodpuštělné vytvořit jakoukoliv drobnou odchylku v deadlines. Například airbag v osobním autě nebo palubní počítač v letadle.

SRTOS - Hard real time OS (tvrdý real time OS)

- Na rozdíl od předchozího HRTOS si SRTOS může dovolit nějaké minimální (předem stanovené) odchylky v reakcích na dané události.

5.1 Embedded systém

Dle [12] se embedded systémy dělí do dvou kategorií:

Do první kategorie spadají embedded systémy které se využívají v průmyslových aplikacích. Zde se kladou většinou obrovské nároky na přesnost. Také podmínky za jakých musí zařízení správně pracovat (vlhkost, prašnost, nízké a vysoké teploty) nemusí být vždy zrovna optimální. Selhání takových zařízení má většinou katastrofální následky v podobě obrovských finančních škod nebo i ohrožení lidských životů.

Do druhé kategorie patří výrobky spotřební elektroniky. V této kategorii se už neklade takový důraz na přesnost a zároveň už přibývají nároky na nízké výrobní náklady.

5.2 Embedded OS

OS pro embedded patří mezi systémy tzv. víceúlohové. To znamená, že OS je schopný spouštět více úloh současně. OS potom rozděluje úlohy ke zpracování procesoru.

Zpracovávání více procesů najednou vyžaduje určitou koordinaci. Taková schopnost koordinace při provádění několika procesů současně se nazývá multitasking.

Multitasking se dělí na Kooperativní a Preemptivní:

Multitasking Kooperativní

Funkce vypadá tak, že operační systém cyklicky přiděluje jednotlivým úlohám čas procesoru (časové kvantum), každá úloha je přitom povinna pravidelně předávat řízení zpět operačnímu systému, aby mohl spustit jinou úlohu. Pokud ovšem nějaká úloha neodevzdá řízení zpět operačnímu systému, dojde k zaseknutí celého systému.

Multitasking Preemptivní

Preemptivní multitasking je modernější forma multitaskingu. O přidělování a odebírání času procesoru zde plně rozhoduje jádro operačního systému (kernel). Ten vždy v pravidelných intervalech (například 10 ms) přeruší danou úlohu a zhodnotí, jaké jiné úlohy čekají ve frontě a jakou mají prioritu. Poté zadá vykonat buď jinou úlohu s vyšší prioritou, nebo nechá dokončit přerušenu úlohu. I v preemptivním multitaskingu se úloha sama může vzdát přiděleného času procesoru (např. v situaci, kdy na něco čeká – např. na načtení vstupních dat).

5.3 Stavba programu v FreeRTOS

Nejdříve je potřeba vytvořit úlohy, které chceme, aby se vykonávaly. V těchto úlohách je potřebné, aby se jim udělil název, priorita a vymeziła velikost zásobníku.

Příklad vytvoření úlohy:

```
xTaskCreate(    vTask1,    // pointer na tento task
               "Task 1",  // název tohoto tasku
               240,      // velikost v zásobníku
               NULL,     // pointer který je použit jako parametr ve
                       // vzniklém tasku
               1,        // priorita tohoto tasku
               NULL );   // tento parametr nepoužijeme
```

Samotná úloha bude vypadat takto:

```
void vTask1 (void *pvParameters)
{
    for(;;)
    {
    }
}
```

Poté už bude následovat příkaz, kterým se spustí plánovač (Scheduler):

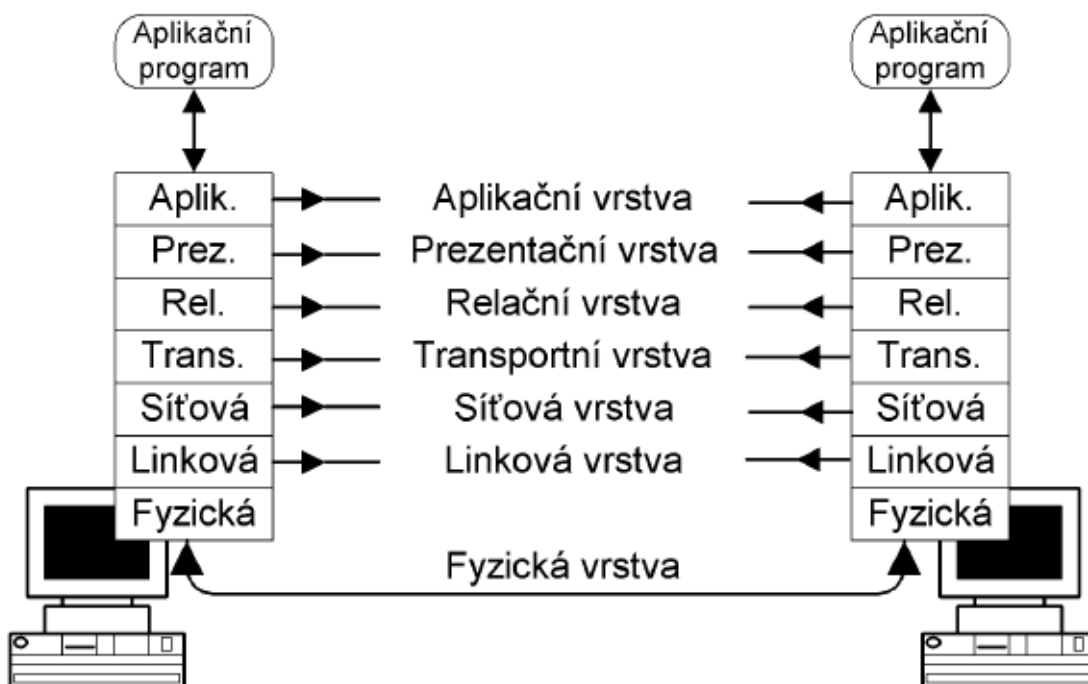
```
vTaskStartScheduler();
```

6 LWIP

V této kapitole budou informace pocházet hlavně z [2] a [4]. lwIP je implementace TCP / IP protokolu, která byla vyvinuta Adamem Dunkelsem pro počítače a síť architektury (CNA) ve Švédském Ústavu Výpočetní Techniky. LwIP si klade za úkol snížit využívání zdrojů, zatímco používá plný rozsah TCP. Z toho vyplývá, že je lwIP vhodné pro použití v oblasti vestavěných systémů s desítkami kilobajtů volné paměti RAM a prostorem pro kolem 40 kilobajtů kódu ROM.

6.1 ISO OSI

Je soustava protokolů, která byla normalizována Mezinárodním normalizačním úřadem (ISO). ISO OSI využívá sedm komunikačních vrstev:



Obrázek 8 – ISO OSI[2]

Fyzická vrstva

Popisuje elektrické či optické signály používané při komunikaci mezi počítači. Na fyzické vrstvě je vytvořen tzv. fyzický okruh. Na fyzický okruh mezi dva počítače bývají často vkládána další zařízení, např. modemy, které modulují signál na telefonní vedení atp.

Linková vrstva

Základní jednotkou pro přenos dat je na linkové vrstvě datový rámeček. Ten se skládá ze záhlaví přenášených dat a zápatí. V záhlaví nese linkovou adresu příjemce a odesílatele. V zápatí obvykle nese kontrolní součet přenášených dat. Jelikož se pakety vyšších vrstev vkládají do přenášených dat nižších vrstev, obsahují přenášená data paket síťové vrstvy.

Síťová vrstva

Síťová vrstva zabezpečuje přenos dat mezi vzdálenými počítači WAN. Základní jednotkou přenosu je síťový paket. Síťový paket se také skládá ze záhlaví a datového pole.

V rozsáhlých sítích (WAN) mezi počítači leží zpravidla jeden nebo více směrovačů. Mezi sousedními směrovači je na linkové vrstvě vždy přímé spojení. Směrovač vybalí síťový paket z datového rámce a před odesláním do jiné linky jej opět zabalí do jiného datového rámce.

Transportní vrstva

Transportní vrstvě se jeví jakoby žádné modemy, opakovače, mosty či směrovače na cestě nebyly. Transportní vrstva se zcela spoléhá na služby nižších vrstev. Také předpokládá, že spojení mezi počítači je zajištěno, proto se bez zbytečných starostí může věnovat spojení mezi aplikacemi na vzdálených počítačích. Jednotkou přenosu je transportní paket, ten se skládá ze záhlaví a datové části. Transportní paket se přenáší opět v datové části síťového paketu.

Relační vrstva

Relační vrstva zabezpečuje výměnu dat mezi aplikacemi, tj. provádí tzv. checkpoint, synchronizaci transakcí (*commit*), korektní uzavírání souborů atd. Základní jednotkou je relační paket, který se opět vkládá do transportního paketu.

Prezentační vrstva

Prezentační vrstva je zodpovědná za reprezentaci a zabezpečení dat. Reprezentace dat může být na různých počítačích různá. Např. se jedná o problém, zdali je nejvyšší bit v bajtu zcela vlevo nebo vpravo. Zabezpečením se rozumí šifrování, zabezpečení integrity dat, digitální podepisování.

Aplikační vrstva

Aplikační vrstva předepisuje v jakém formátu a jak mají být data přebírána/předávána od aplikačních programů. Např. protokol Virtuální terminál popisuje, jak mají být data formátována, ale i dialog mezi oběma konci spojení.

6.2 TCP/IP

Rodina protokolů TCP/IP se nezabývá na rozdíl od ISO OSI fyzickou a linkovou vrstvou. V praxi se i v Internetu používají pro fyzickou a linkovou vrstvu často protokoly vyhovující normám ISO OSI. Každá skupina má vlastní definici svých vrstev i protokolů jednotlivých vrstev. V praxi však je třeba využívat komunikační zařízení vyhovující ISO OSI pro přenos IP-paketů nebo např. naopak realizovat služby podle ISO OSI přes Internet.

6.2.1 IP (Internet Protokol)

Prakticky odpovídá síťové vrstvě. IP-protokol přenáší tzv. IP datagramy mezi vzdálenými počítači. Každý IP datagram ve svém záhlaví nese adresu příjemce, což je úplná směrovací informace pro dopravu IP datagramu k adresátovi. Takže síť může přenášet každý IP datagram samostatně. IP datagramy tak mohou k adresátovi dorazit v jiném pořadí, než byly odeslány. Každé síťové rozhraní v rozsáhlé síti Internet má svou celosvětově jednoznačnou IP-adresu (jedno síťové rozhraní může mít více IP-adres, avšak jednu IP-adresu nesmí používat více síťových rozhraní). Internet je tvořen jednotlivými sítěmi, které jsou propojeny pomocí směrovačů. Směrovač se anglicky nazývá router, ve starších publikacích se však označuje jako gateway.

6.2.2 TCP (Transmission Control Protocol)

Protokol TCP je tzv. spojovanou službou, tj. příjemce potvrzuje přijímaná data. V případě ztráty dat (ztráty TCP segmentu) si příjemce vyžádá zopakování přenosu. Tento protokol odpovídá transportní vrstvě, takže dopravuje data pomocí TCP segmentů, které jsou adresovány jednotlivým aplikacím. Dále pak zajišťuje spojení mezi aplikacemi běžícími na vzdálených počítačích.

6.3 Ethernet driver

Pro kit EKS-LM3S6965 je k dispozici několik knihoven dostupných z [1], [4], [9], které umožňují jednodušeji využívat ethernet. Pro tuto aplikaci jsou velice důležité funkce, které se nachází v ethernetovém driveru ETHIsr.c. V této knihovně se nachází funkce na ethernetové přerušování, inicializaci a nastavení ethernetového kontroleru, práci s bufferem a jiné.

7 Server HTTP

Tato kapitola čerpá z [13] a [14]. Hypertext Transfer Protocol je protokol, který byl navržen tak, aby byl schopen přenášet Hypertext Markup Language (HTML) dokumenty. HTML je jazyk, který se používá k vytvoření hypertextových dokumentů, které obsahují odkazy na další prvky, jako jsou například grafické obrázky, audio, video klipy a Java applety. V současné době mají HTTP 1.0 a HTTP 1.1 stabilní specifikace, a proto vývoj HTTP hodně opadl.

7.1 Funkce HTTP

HTTP je založen na principu požadavek-odpověď. Klient, který má spuštěnou aplikaci, která se nazývá prohlížeč, vytváří spojení se serverem a odešle požadavek na server. Server odpoví stavem, včetně sdělení o verzi protokolu a informací o úspěchu nebo chybě, následuje zpráva obsahující informace o serveru a obsah. V internetu komunikuje http obecně přes TCP.

http komunikace se dělí na čtyři kroky:

1. Prohlížeč otevře spojení.
2. Prohlížeč odešle serveru požadavek.
3. Server odešle prohlížeči odpověď.
4. Spojení se ukončí.

V dnešní době se požaduje, aby bylo spojení navázáno těsně před každou novým požadavkem od klienta a poté ihned po odpovědi serveru zase uzavřeno. Server i klient si musí být vědomi, že spojení se může kdykoliv z jedné strany nečekaně ukončit. Proto existují určité timeouty po kterých se spojení samo ukončí.

7.2 HTTP zpráva

HTTP zpráva se skládá z následujících částí:

- Typ zprávy: Tento parametr říká, jestli je tato zpráva požadavek od klienta, nebo odpověď serveru.
- Hlavička zprávy: V tomto poli se může nacházet jeden z následujících parametrů: obecná hlavička, hlavička požadavku, hlavička odpovědi, hlavička subjektu.
- Tělo zprávy: Nese tělo zprávy.
- Záhlaví: Možnosti pro použití záhlaví jsou následující: kontrola Cache, připojení, datum, pragma, kódování, upgrade a prostřednictví.

7.3 SSI a CGI

V některých aplikacích nestačí mít stránky statické, z těchto důvodů tu jsou direktivy SSI a CGI, díky kterým mohou být na stránkách i dynamické prvky.

SSI vzniklo dříve a z velké části se využívá pro kompletaci dokumentů, kde jsou například hlavičky shodné. Následná změna se provede jednou a bude viditelná na všech stránkách. Takto se ušetří hodně času a práce.

Direktiva SSI se zapisuje ve tvaru: `<!--#direktiva parametr="hodnota"-->`
Soubory, ve kterých jsou direktivy SSI obsaženy, mají většinou příponu `.shtml`. Výsledek direktivy je uzavřen do poznámky HTML z toho důvodu, aby se při chybě nezměnil vzhled stránky.

Na tvorbu celých dynamických stránek se využije takzvaný CGI skript. Tyto stránky mohou již kompletně měnit svůj obsah, který může být závislý například na tom, z jakého počítače se bude otevírat.

Skript by měl generovat minimálně hlavičku `Content-Type:` kterou následuje prázdný řádek, který značí konec hlavičky. V hlavičce je napsáno, co skript generuje.

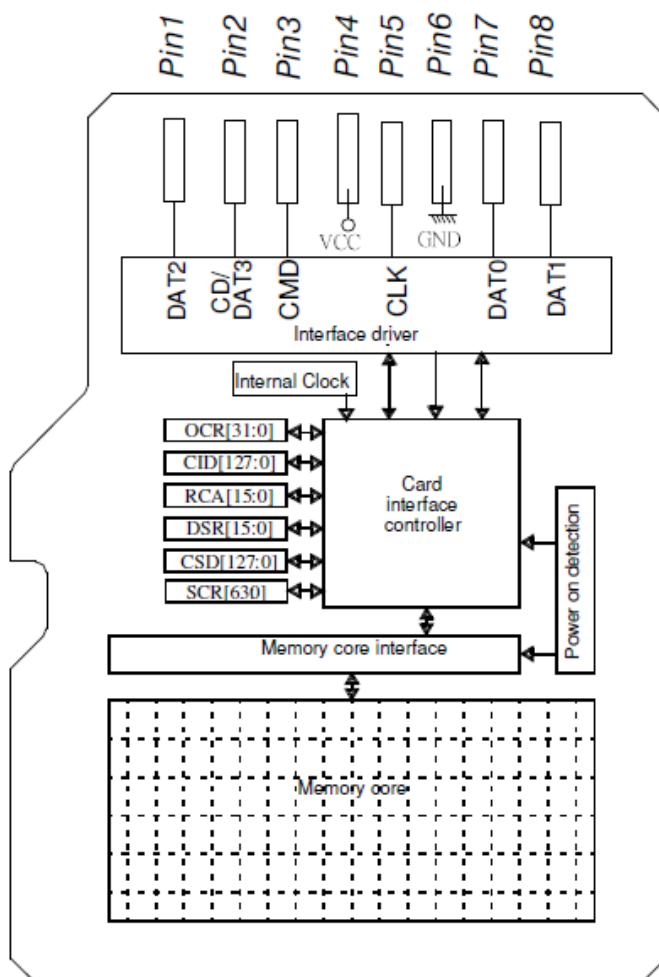
7.4 Uložiště stránek

Jako vnitřní úložiště stránek se používá paměť typu flash, která se nachází v MCU.

8 Souborový systém na kartě SD

8.1 MicroSD

Čerpáno je převážně z [5]. MicroSD karty jsou vysoce integrované flash paměti se sériovou a náhodnou schopností přístupu. Ta je přístupná přes vyhrazené sériové rozhraní optimalizované pro rychlý a spolehlivý přenos dat. Systém je založený na inovacích v polovodičové technice. Toto zařízení je levné, mechanicky stabilní paměťové médium v podobě karty pro multimediální spotřebitelské aplikace. Nízká spotřeba energie a široký rozsah napájecího napětí upřednostňuje mobilní aplikace, jako jsou audio přehrávače, organizátory a elektronické knihy. Použití je velmi vhodné pro systémy využívající kompresi dat, jako je MPEG. Zde bude karta MicroSD dodat dostatek kapacity pro všechny druhy multimediálních dat.



Obrázek 9 – Blokové schéma MicroSD [5]

8.2 Rozhraní

Rozhraní MicroSD karet může pracovat ve dvou různých režimech:

- SD Card režim
- SPI mód

Hostitelský systém může vybrat jeden z režimů. SD Card režim umožňuje 4-bitový vysoce výkonný přenos dat. SPI mód umožňuje snadné rozhraní pro společný SPI kanál. Nevýhodou tohoto způsobu je ztráta výkonu.

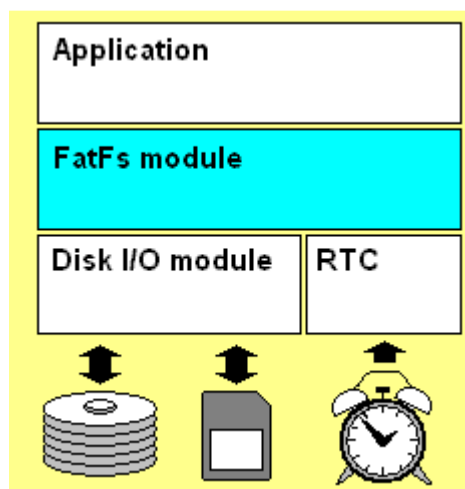
8.3 Knihovna FatFS

FatFs modul je experimentální projekt, kterým lze realizovat souborový systém FAT na malých embedded systémech. Modul FatFs je napsán v souladu s ANSI C, proto je nezávislý na hardwarové architektuře. Proto může být začleněn do většiny 8-bitových mikrokontrolérů, jako je 8051, PIC, AVR, H8, Z80 a atd. ..., bez jakékoli změny.

Disk I / O rozhraní:

Vzhledem k tomu, FatFs / Tiny-FatFs modul je zcela oddělen od I / O vrstvy, vyžaduje následující funkce pro nižší vrstvy, které přistupují k čtení / zápisu na fyzický disk.

- disk_initialize - Inicializace disku.
- disk_status - Získání stavu disku.
- disk_read - Přečte sektor (y).
- disk_write – Zapíše do sektoru (y).
- disk_ioctl - Řídící zařízení závislé na funkcích.
- get_fattime - Získá aktuální čas.



Obrázek 10 – FAT File System Modul [5]

FatFs modul poskytuje následující funkce:

- f_mount: Tato funkce pouze inicializuje pracovní místo a zaregistruje jeho adresu do vnitřní tabulky.
- f_open: Funkce f_open vytvoří objekt, který bude použit pro přístup k souboru.
- f_read : Přečte data ze souboru.
- f_write : Zapisuje data do souboru.
- f_lseek : Funkce f_lseek pohybuje pointerem pro čtení / zápis v otevřeném souborovém objektu.
- f_sync: Funkce f_sync vymaže informace o zapsání souboru, které jsou v cache.
- f_opendir: Funkce f_opendir otevře existující adresář.
- f_readdir : Funkce f_readdir čte položky adresáře.
- f_getfree : Funkce f_getfree získá počet volných alokačních jednotek.
- f_stat : Funkce f_stat získá informace o souboru nebo adresáři.
- f_mkdir: Funkce f_mkdir vytvoří nový adresář.
- f_unlink: Funkce f_unlink odstraní soubor nebo adresář.
- f_rename: Funkce f_rename přejmenuje soubor nebo adresář.
- f_mkfs: Funkce f_mkfs vytvoří souborový systém na disku.
- f_chmod : Funkce f_chmod změní atribut souboru nebo adresáře.

9 Zpracování signálu z čtečky RFID

V následující kapitole bude čerpáno převážně z [10]. Jelikož z podstaty kódování Manchester vyplývá, že v jednom přijatém a dekodovaném bitu se nachází oba dva stavy, tedy L i H, lze při zpracování signálu využít délky impulsů. Při přijaté jedničce se v bitovém intervalu mění hodnota signálu z H na L a při nule je to naopak. Dá se tedy předpokládat, že když přicházejí samé jedničky nebo nuly, tak v nedekodovaném stavu přicházejí krátké impulsy, které se střídají v L a H. Hodnota přijatého bitu se mění, jen když přijde dlouhý impuls.

Předchozí uvažování se dá použít pro dekodování dat ze čtečky:

Nejdříve musí úspěšně proběhnout START sekvence, což znamená, že se čeká na jeden dlouhý kladný impuls (nula) poté co přijde, musí následovat 16 krátkých impulsů (9 jedniček za sebou). Po té, co je tato podmínka splněna může přijem samotných dat.

Nejdříve se vytvoří proměnná a přiřadí se jí jednička. Potom můžou přijít čtyři druhy impulsu:

Krátký L impuls (kL)	-	Neděje se nic.
Dlouhý L impuls (dL)	-	Nuluje se proměnná.
Krátký H impuls (kH)	-	Uloží se obsah proměnné jako další bit přijatého kódu.
Dlouhý H impuls (dH)	-	Připíše se 0 jako další bit přijatého kódu a nastaví se proměnná do 1. Zároveň se do dalšího bitu zapíše "1".

Poté je ještě nutné vyřešit kontrolní bity, které slouží jako ochrana proti přijmutí špatného čísla. To znamená, že je potřeba u každé čtveřice přichozích bitů zkontrolovat, jestli souhlasí parita. A nakonec provést kontrolní součet pomocí xor všech deseti čtveřic čísel a kontrolního součtu (poslední čtyři bity přijatého kódu).

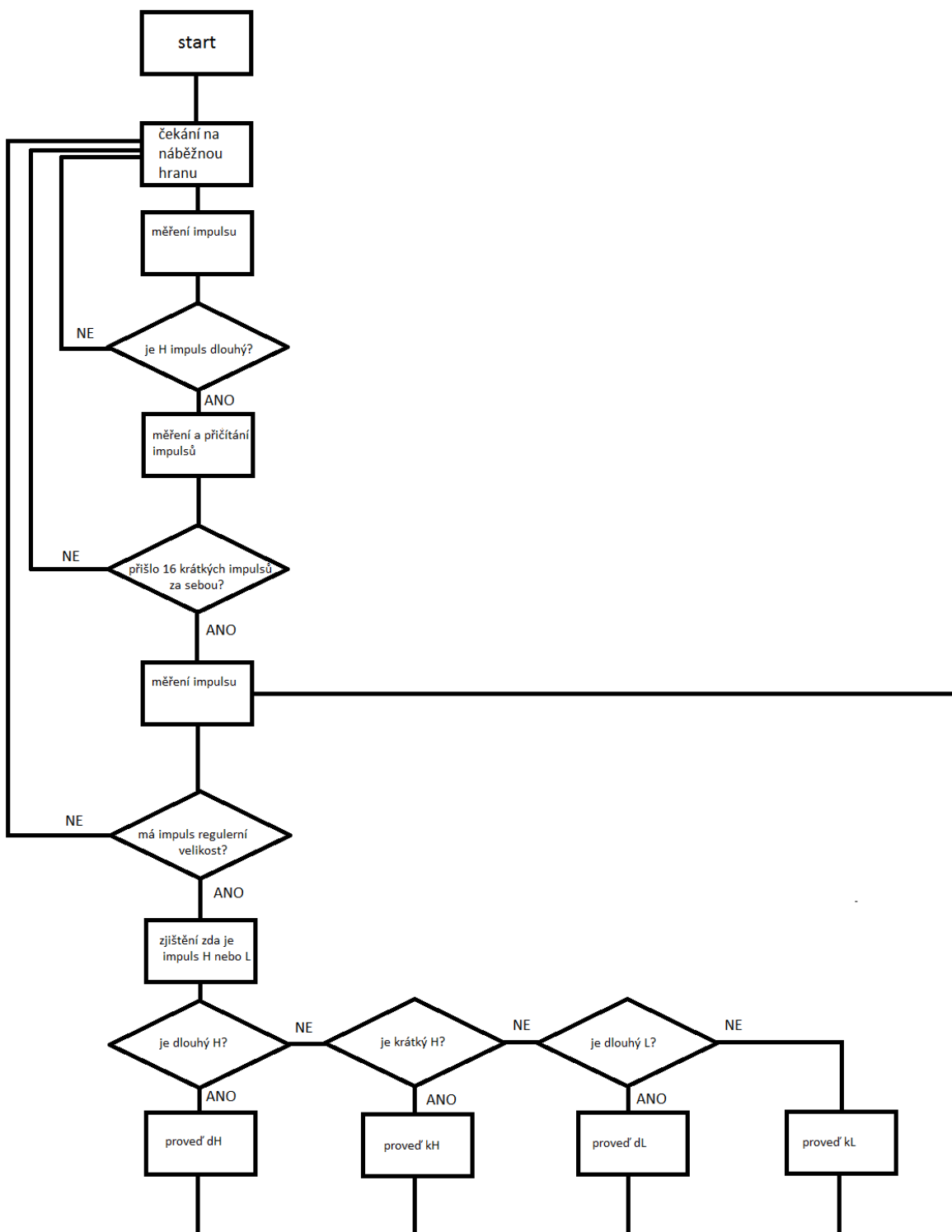
9.1 Časování

Bitový interval má velikost 440 us. Z principu Manchesteru je zřejmé, že hodnota se mění zhruba v polovině, což je asi 220 us. Jak bylo na začátku této kapitoly vysvětleno, může tedy být impuls dlouhý 220 us nebo 440 us.

9.2 Přerušení

Jelikož je tato aplikace náročná na časovou přesnost, je vhodné přidělit přerušení co nejvyšší prioritou. V opačném případě nemusejí být impulsy správně rozeznávány.

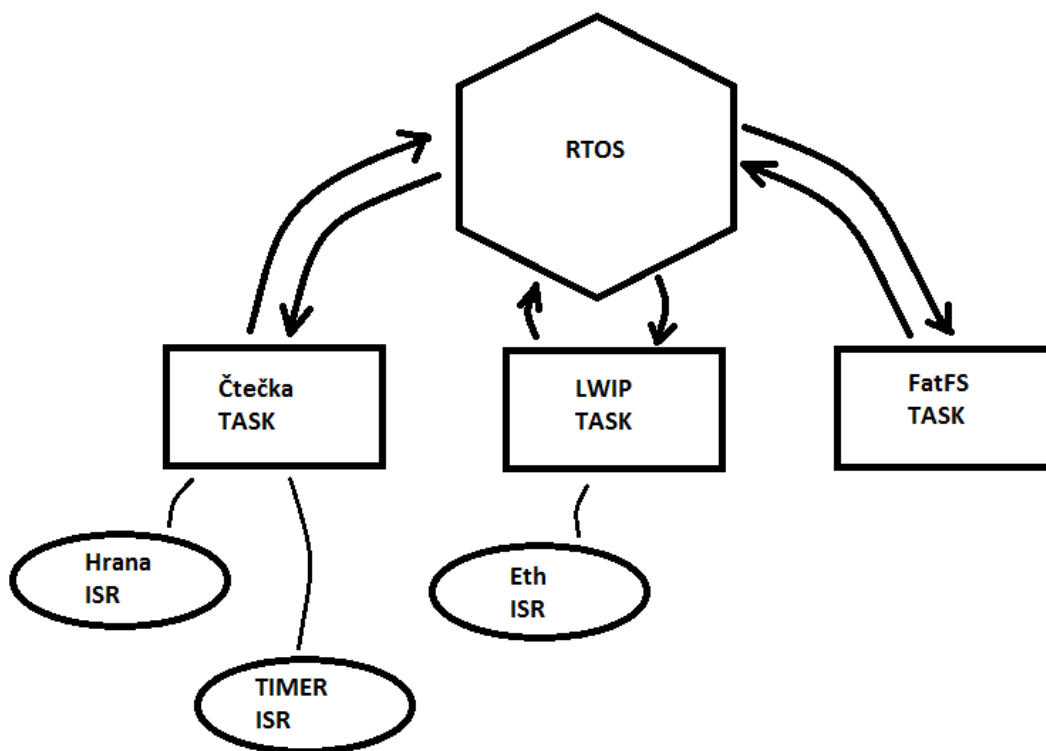
9.3 Blokový diagram



Obrázek 11 – Vývojový diagram zpracování signálu ze čtečky

10 Výsledná aplikace

Výsledná aplikace se skládá z FreeRTOS a tří tasků.



Obrázek 12 – Blokové schéma výsledné aplikace

První task bude použit pro dekódování dat ze čtečky karet. Tento task využívá dvě přerušení, přičemž jedno je vyvolané portem na který máme přiveden signál ze čtečky karet. Toto přerušení je vyvoláno při detekci hrany. Další přerušení je vyvoláváno z důvodů měření délek impulsů.

Další task zprostředkovává http server. Tento task využívá přerušení od ethernetu. Na webových stránkách se dá pomocí dynamických prvků nastavit aktuální čas.

Poslední task se zabývá uložením dat na kartu MicroSD k tomu je využit driver FatFS. Na paměťové médium by se měli postupně ukládat zachycená ID čísla karet.

Závěr

Úkolem této bakalářské práce bylo vytvoření zařízení pro sledování docházky. Příchozí osoba by tedy měla přiložit identifikační kartu, uvnitř které se nachází čip typu EM4100. ID čísla karet by měl být požadovaný systém schopen poskytovat po ethernetové síti formou HTTP serveru. Zároveň by měl být schopen tyto ID sám uchovávat, proto je nutné, aby zařízení vlastnilo nějaký typ úložiště.

Pro tvorbu tohoto systému byl nakonec zvolen procesor LM3S6965 ve formě vývojového kitu, protože splňuje podmínku vlastního úložiště ve formě slotu pro microSD kartu a od výrobce poskytnutým ovladačem pro FatFS. Dále je pak na desce i ethernetový port.

Z důvodů mnoha činností, které měl procesor vykonávat, byl zvolen OS pro embedded systémy. Volba nakonec padla na FreeRTOS. Systém je plně funkční a tasky fungují bez problémů. Ze začátku, když se vyskytly nějaké potíže, tak za ně většinou mohla špatně nastavená priorita u tasků. Na FreeRTOS bylo potom třeba rozchodit LWIP s http serverem ve formě tasku.

Nakonec se nepodařilo vytvořit vlastní zdroj energie, jelikož to tento kit neumožňuje a nelze to provést bez větších zásahů do zapojení.

Výběr čtečky karet EM4095 se povedl. Přes počáteční problémy při nalézání správných kondenzátorů k anténě se nakonec povedlo docílit správné funkce. Čtečka po většinu času posílá správné informace k procesoru, pouze občas nemá impuls tu správnou délku. Tento problém je ale vyřešen v programu.

Dále bych chtěl kladně ohodnotit fóra na stránkách TI E2E Community. Když se vyskytl nějaký problém, tak je tam většinou nějaké podobné téma již řešeno. Je možnost vložit i svůj vlastní dotaz k nějaké problematice a vývojář většinou velmi brzy odpoví.

Nastává zde problém při současném použití SD karty a displaye protože obě zařízení jsou zapojena na stejný port ssi.

Bohužel se nepodařilo aplikaci udělat zatím plně funkční, ale jednotlivé části samostatně fungují. Podařilo se tedy vytvořit funkční http server, který funguje pomocí tasku v RTOS. Dále je funkční načtení ID čísla ze čtečky karet.

Některé věci bylo potřeba probrat jen teoreticky z důvodů jejich dosavadní nefunkčnosti v době psaní této práce, avšak základní princip snad byl vysvětlen dostatečně názorně.

LITERATURA

- [1] Stellaris ARM Cortex-M3 [online]. 2007-2010 [cit. 2011-10-21]. Dostupné z WWW: [url: <http://www.ti.com/tool/eks-lm3s6965>].
- [2] Kabelová A., Dostálek L.: Velký průvodce protokoly TCP/IP a systémem DNS, Computer Press, Praha 2008, ISBN: 978-80-251-2236-5
- [3] ARM The Architecture For The Digital World [online]. Version 2.0. 2010-09-12 [cit. 2011-07-29]. CMSIS - Cortex Microcontroller Software Interface Standard. Dostupné z WWW:[url: <http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php>].
- [4] lwIP - A Lightweight TCP/IP stack - Summary Technical Report T2001:20. Dostupné z WWW: [<http://savannah.nongnu.org/projects/lwip/>].
- [5] MicroSD Card. *MicroSD Card Specification*. 2005. vyd. Dostupné z: http://www.digitalspirit.org/file/index.php/object/download/docs/sd/Kingmax_microsd.pdf
- [6] SOMMEROVÁ, Martina. Základy RFID technologií: Výukový materiál [online]. [cit. 2012-04-14]. Dostupné z: http://rfid.vsb.cz/miranda2/export/sites-root/rfid/cs/okruhy/informace/RFID_pro_Logistickou_akademii.pdf
- [7] Rfid-handbook. [online]. [cit. 2012-04-25]. Dostupné z: <http://www.rfid-handbook.de/rfid/standardization.html>
- [8] ISO/IEC 18000-2. Information technology-Radio frequency identification for item management: Parameters for air interface communications below 135 kHz. 2004-09-15. International Organization for Standardization.
- [9] REAL TIME ENGINEERS LTD. FreeRTOS: The FreeRTOS Project [online]. [cit. 2012-08-20]. Dostupné z: <http://www.freertos.org/>
- [10] CALETKA, Ondřej. Bezkontaktní identifikace prakticky. [online]. [cit. 2012-08-27]. Dostupné z: <http://pluto.pslib.cz/caletka/rfid/index.htm>
- [11] ENDEL, Petr. RFID a jeho aplikace v inteligentních dopravních systémech. 2008. Dostupné z: http://zefram.klfree.net/projekt/soubory/bp_endel_petr.pdf. Bakalářská práce. České vysoké učení technické v Praze. Vedoucí práce doc.ing.Tomáš Zelinek, CSc.
- [12] MYSLÍK, Jan. Srovnávací studie vestavěných operačních systémů. 2008. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/xsmitka_2009bach.pdf. Bakalářská práce. České vysoké učení technické v Praze. Vedoucí práce Ing. Jiří Smítka.

- [13] PARZIALE, Lydia, David T. BRITT, Chuck DAVIS, Jason FORRESTER, Wei LIU, Carolyn MATTHEWS a Nicolas ROSSELOT. IBM. TCP/IP Tutorial and Technical Overview [online]. International Technical Support Organization, 2006 [cit. 2012-09-01]. Dostupné z: <http://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf>

- [14] MASARYKOVA UNIVERZITA. Technické informace: WWW [online]. [cit. 2012-09-01]. Dostupné z: <http://www.fi.muni.cz/tech/index.xhtml.cs>

- [15] EM MICROELECTRONIC-MARIN SA. EM4095: Read/Write analog front end for 125kHz RFID Basestation [online]. 2012 [cit. 2012-08-02]. Dostupné z: http://www.emmicroelectronic.com/webfiles/product/rfid/ds/EM4095_DS.pdf

Příloha A – Popis přílohy

DVD

**Textová část práce ve formátu pdf.
Plné zdrojové kódy aplikace.**