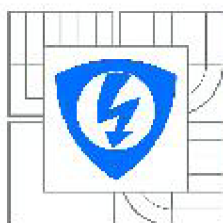




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

VZDÁLENÉ OVLÁDÁNÍ ZAŘÍZENÍ PŘES TCP/IP

REMOTE DEVICE CONTROL OVER TCP/IP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

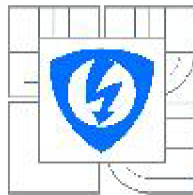
AUTOR PRÁCE
AUTHOR

MAREK SKALNÍK

VEDOUČÍ PRÁCE
SUPERVISOR

Ing. ZBYNĚK FEDRA, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Marek Skalník

ID: 106254

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Vzdálené ovládání zařízení přes TCP/IP

POKYNY PRO VYPRACOVÁNÍ:

Navrhnete elektronické zařízení pro vzdálené ovládání zařízení pomocí protokolu TCP/IP. Zařízení bude mít 10/100Mbps ethernetové rozhraní a 3 relátkové výstupy (12V/4A). Na každém výstupu bude za pomoci LED diody signalizován stav sepnuto/rozepnuto. Napájení zařízení bude z dostupného AC/DC adaptéru jehož vývoj není součástí zadání. Napájecí napětí dle návrhu +5V nebo +12V. Ethernet komunikace bude přes WEB rozhraní chráněna heslem.

Realizujte navržené zařízení. Uživatel bude po přihlášení do administrace vidět stav jednotlivých výstupů. Za pomoci ovládacích tlačítek bude možno každý daný výstup ovládat. Z uživatelského hlediska bude dále možno nastavení IP adresy, masky sítě a přístupového hesla. Zařízení má být schopno obnovit své nastavení po výpadky napájení a resetovat své nastavení.

DOPORUČENÁ LITERATURA:

[1] VAŇA, V. ARM pro začátečníky. Praha: BEN - technická literatura, 2009.

Termín zadání: 7.2.2011

Termín odevzdání: 27.5.2011

Vedoucí práce: Ing. Zbyněk Fedra, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce navazuje na předchozí semestrální projekt. Obsahuje popis jednoho z mikrokontrolérů firmy ATMEL s architekturou ARM a jeho použití v aplikaci s ethernetovým rozhraním. Je zde popsán návrh hardwarové části zařízení a popis softwarové části, která zajišťuje vykonávání činností, tak jak je požaduje obsluha zařízení. Softwarová část je dále implementována do operačního systému reálného času FreeRTOS.

KLÍČOVÁ SLOVA

ARM, RISC, RTOS, mikrokontroler, TCP/IP

ABSTRACT

The bachelor thesis is based on the previous semestral project. The thesis is focused to the design of a remote control switching device. The thesis is divided into two main parts. The first one is a hardware design and the second one is a software design. Main hardware components ATMEL ARM single chip controller, Fast Ethernet Physical Layer Single Chip Transceiver, Flash memory and powering circuit are discussed from the application point of view in the hardware design part. The software desing part is focused to a used real time operating system FreeRTOS, to most important parts of a main application and to a web user interface. Achieved results are discussed in the summary.

KEYWORDS

ARM, RISC, RTOS, microcontroller, TCP/IP

SKALNÍK, M. *VZDÁLENÉ OVLÁDÁNÍ ZAŘÍZENÍ PŘES TCP/IP*. BRNO: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ, 2011. 33 S. VEDOUcí SEMESTRÁLNÍ PRÁCE ING. ZBYNĚK FEDRA, PH.D.

Prohlášení

Prohlašuji, že svůj bakalářský projekt na téma Vzdálené ovládání zařízení přes TCP/IP jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu 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 26. května 2011

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářského projektu Ing. Zbyňku Fedrovi, Ph.D. za účinnou metodickou a pedagogickou pomoc a dále Ing. Martinu Mlaskačovi a Ing. Michalu Kohoutkovi Ph.D. za odbornou pomoc a další cenné rady při zpracování mého bakalářského projektu.

V Brně dne 26. května 2011

.....
podpis autora

OBSAH

<i>Seznam obrázků a tabulek</i>	7
<i>1 Úvod</i>	8
<i>2 Hardwarová část zařízení</i>	8
<i>2.1. Mikrokontrolér AT91SAM7X512</i>	8
2.1.1 Charakteristika procesoru ARM	9
2.2. Integrovaný obvod DM9161A	11
2.3. Dataflash AT45DB041D	12
2.4. Napájení zařízení	13
2.5. Výstupní zařízení	14
<i>3 Softwarová část zařízení</i>	15
3.1 RTOS –operační systém reálného času	15
3.1.1 FreeRTOS	15
<i>4 Nastavení mikrokontroleru</i>	15
4.1. Nastavení sériové komunikace.....	15
4.1.1 Nastavení sériové komunikace – mikrokontroler.....	16
4.1.2 Nastavení sériové komunikace – paměť AT45DB041D	19
<i>5 Ovládání zařízení</i>	21
5.1. Uživatelské webové rozhraní.....	21
5.2. Resetovací tlačítko.....	24
<i>6 Závěr</i>	25
<i>LITERATURA</i>	26
<i>Seznam zkratek</i>	27
<i>PŘÍLOHY</i>	28
Schema	28
Popis činnosti zařízení z hlediska uživatele	30

Seznam obrázků a tabulek

Obr. 1 Typické zapojení napájení mikrokontroléru [2]	9
Obr. 2 ARM z hlediska programování [9].....	11
Obr. 3 Blokové schéma DM9161A [3].....	12
Obr. 4 Blokový diagram paměti AT45DB041D [4].....	13
Obr. 5 Zapojení regulátoru LM317 v mém zařízení a dle datasheetu výrobce [5].....	13
Obr. 7 Příklad zapojení dvou zařízení pomocí SPI [8]	16
Obr. 8 Příklad zapojení 3 zařízení mikrokontroleru AT91SAM7X512 pomocí SPI [2].....	16
Obr. 9 Rozdělení paměti pro periferie AT91SAM7X512 [2]	17
Tab. 1 Identifikace jednotlivých periférií AT91SAM7X512 [2]	18
Tab. 2 Identifikace jednotlivých periférií AT91SAM7X512 [2]	18
Obr. 10 Průběh signálů SPI při nastavení NCPHA=1 [2].....	19
Obr. 11 SPI módy dle datasheetu AT45DB041D [4].....	19
Tab. 3 Seznam OpCodu paměti AT45DB041D [4].....	20
Tab. 4 Instrukce pro čtení paměti AT45DB041D [4]	20
Tab. 5 Instrukce pro zápis do paměti AT45DB041D [4].....	21
Obr. 12. Úvodní stránka pro zadání hesla	22
Obr. 13. Stránka pro uživatelské nastavení	23
Obr. 14. Stránka pro nastavení výstupů.....	23
Obr. 15. Stránka login.shtml po převedení do hexadecimálního kódu	24
Obr. 16. Zapojení resetovacího tlačítka	24
Obr. 17. Úvodní stránka login.shtml po zadání IP adresy zařízení	30
Obr. 18. Zadání přihlašovacího hesla	31
Obr. 19. Uživatelské záložky	31
Obr. 20. Záložka IO	32
Obr. 21. Změna IP adresy	32
Obr. 22. Přihlašovací stránka s novou IP adresou	33

1 Úvod

Tato práce navazuje na semestrální projekt, zabývající se návrhem zařízení pro vzdálené ovládání jiného zařízení (dále IP relé). Jedná se o zařízení, které umožňuje vzdáleně po ethernetovém rozhraní ovládat připojená zařízení přes webové rozhraní. Přístup k ovládání IP relé je chráněno heslem. Přes toto webové rozhraní je uživatel schopen nastavit heslo, IP adresu a masku a dále sledovat stavy jednotlivých výstupů a tyto výstupy také ovládat. Dle zadání má být IP relé napájeno 5V nebo 12V. IP relé lze využít například pro spínání napájecích napětí pro jiná zařízení a tím je zapínat či ovládat nebo třeba spínat resetovací vodič připojeného zařízení a tím ho restartovat. Motivací k návrhu tohoto IP relé byla potřeba vzdáleně restartovat wifi zařízení jednoho z poskytovatelů wifi připojení.

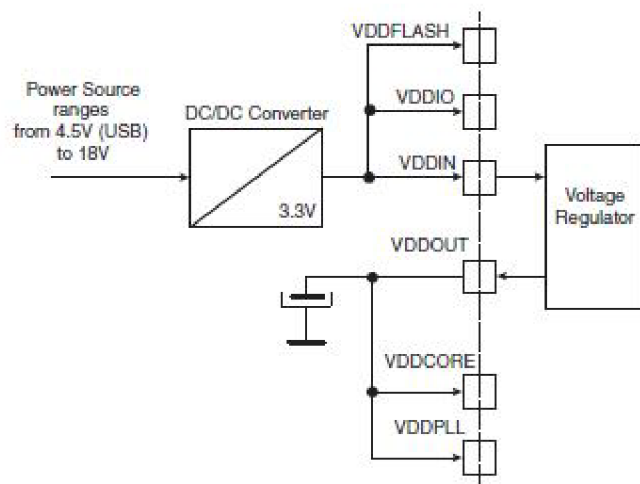
2 Hardwarová část zařízení

2.1. Mikrokontrolér AT91SAM7X512

IP relé je řízeno mikrokontrolérem od firmy ATMEL AT91SAM7X512 s jádrem ARM. Tento mikrokontrolér byl vybrán z důvodu přítomnosti ethernetového řadiče (MAC), dostupnosti informací a vzorových aplikací a také z důvodu možnosti odzkoušet kit s tímto mikrokontrolérem od firmy Kramara. Dále je tento mikrokontrolér připraven pro rozšíření dalších možných funkcí IP relé, vzhledem k dalším rozhraním, kterými disponuje (RS232, RS485, USB, aj.) a také k rozšíření počtu ovládaných výstupů. Jeho základní vlastnosti dle [2] jsou:

- ARM7TDMI ARM Thumb Procesor
- Interní rychlá Flash paměť 512 kB
- Interní rychlá SRAM paměť, přístup v jednom cyklu, až 128 kB
- Memory Controller (MC)- integrovaný Flash kontrolér s detekcí poruchových stavů
- Reset Controller (RSTC) – zajišťující Power-on Reset a výrobcem kalibrováný Low-power Brown-out detektor
- Clock Generator (CKGR) – nízkoodběrový integrovaný RC oscilátor 3-20MHz
- Power Management Controller (PMC), Slow Clock Mode (Down to 500 Hz) and Idle Mode
- Advanced Interrupt Controller (AIC)-rozšířený kontrolér přerušení, individuálně maskovatelné, 8 úrovní priorit, dva externí zdroje přerušení a jeden rychlý zdroj přerušení
- Periodic Interval Timer (PIT), 20bitový programovatelný čítač, 12-bitový intervalový čítač

- Real-time Timer (RTT) – 32 bitový čítač
- Windowed Watchdog (WDT)
- Two Parallel Input/Output Controllers (PIO)- 62 I/O pinů
- Thirteen Peripheral DMA Controller (PDC) Channels
- USB 2.0 Full Speed (12 Mbits per second) Device Port
- Ethernet MAC 10/100 base-T
- One Part 2.0A and Part 2.0B Compliant CAN Controller
- Two Universal Synchronous/Asynchronous Receiver Transmitters (USART)
- Individual Baud Rate Generator, IrDA Infrared
- Podporuje ISO7816 T0/T1 Smart Card, Hardware Handshaking, RS485
- TWI- two wire interface
- Napájení- integrovaný 1,8V regulátor, až 100mA pro jádro a případné externí součásti, 3,3VDDIO napájení I/O vývodů, nezávislé napájení Flash 3,3V VDDFLASH, napájení jádra 1,8V VDDCORE s Brownout detektorem
- Pouzdro LQFP, 100pinů, PbFree



Obr. 1 Typické zapojení napájení mikrokontroléru [2]

2.1.1 Charakteristika procesoru ARM

Architektura ARM (Advanced RISC machines) vytvořená firmou ARM je široce rozšířená v 16/32 bitových embedded RISCových mikroprocesorech Von Neumannovou architekturou založenou na load/store koncepci. Díky výhodné licenční politice firmy ARM, která dovoluje vyrábět jednočipové počítače s jádrem ARM, je použití těchto obvodů velice rozšířené. Jeho další výhodou jsou energeticky úsporné vlastnosti, proto je používány hlavně v mobilním odvětví spotřební elektroniky.

Architektura ARM

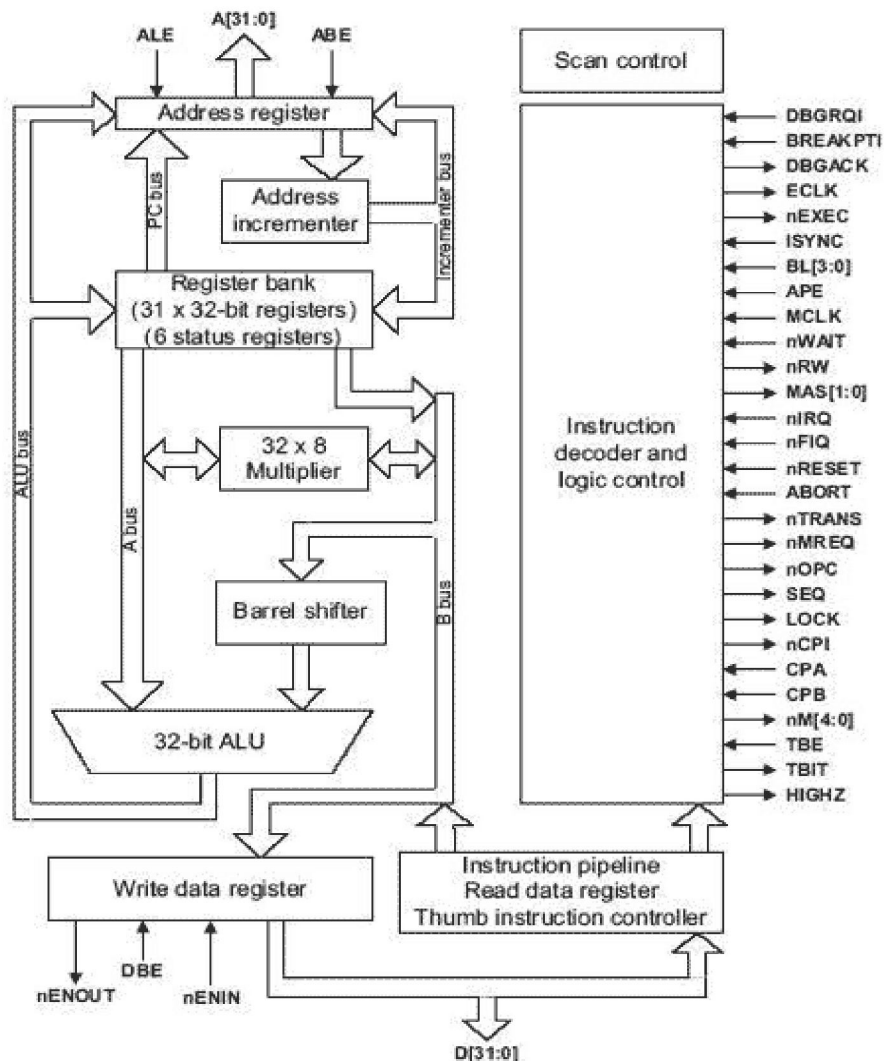
Je založena na redukovaném instrukčním souboru RISC (ARM obsahuje 44 základních instrukcí s jednotnou šířkou 32 bitů), jenž mají vysokou instrukční výkonnost a rychlou reálnou odezvu na přerušení. Další předností je využití také 16bitové Thumb podmnožiny, která dokáže ušetřit 35-40% paměti ve srovnání s 32bitovým instrukčním souborem a to bez toho, aby se výrazně snížil výkon.

ARM má šest operačních módů:

- **uživatelský**, neprivilegovaný režim, ve kterém běží většina úloh
- **FIQ**, používaný při vyvolání přerušení s vysokou prioritou (fast)
- **IRQ**, používaný při vyvolání přerušení normální prioritou
- **Supervisorský**, používaný po resetu a při provádění instrukce softwarového přerušení
- **Ukončující (abort)**, používané při řešení problému s přístupem do paměti
- **Nedefinovaný (unref)**, používané v případě nedefinovaných instrukcí

V procesoru je obsaženo 25 částečně se překrývajících 32bitových registrů (15 registrů je univerzálních a zbývajících 10 má speciální funkce), z toho 16 registrů je v každém režimu činnosti programově přístupných. Další vlastností procesorů s jádrem ARM je jejich tzv.

Load/Store architektura. Ta spočívá v tom, že všechny operace jsou prováděny pouze mezi registry. Výhodou této architektury je velmi rychlé CPU. Nevýhodou je často nutnost použití více instrukcí při zpracování dat uložených v paměti, než u jiných architektur [1].

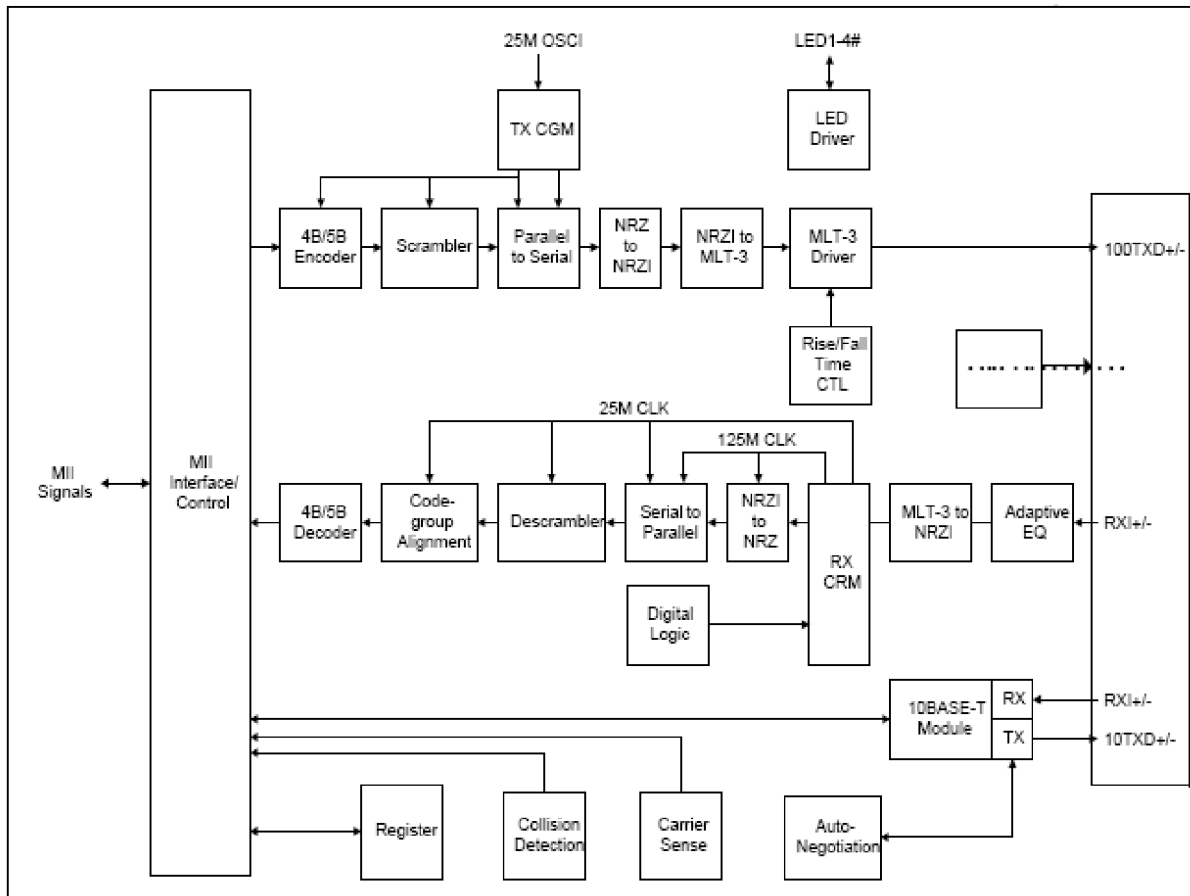


Obr. 2 ARM z hlediska programování [9]

2.2. PHY rozhraní DM9161A

Dalším důležitým obvodem v IP relé je integrovaný obvod firmy Davicom. Jeho blokové schéma je na obr. 3. Tento integrovaný obvod DM9161A je 10/100Mbps Fast Ethernet Physical Layer Single Chip Transceiver s pouzdrem LQFP48. Napájecí napětí pro tento obvod je 3,3V [3].

Tento obvod komunikuje s mikrokontrolérem standardizovaným rozhraním MII a tvoří poslední, fyzickou vrstvu a tím zprostředkovává komunikaci po ethernetu.



Obr. 3 Blokové schéma DM9161A [3]

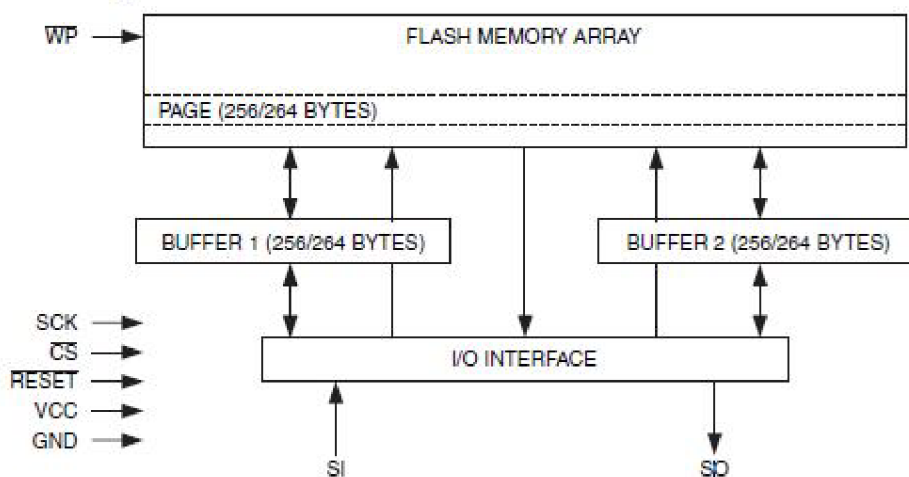
2.3. Dataflash AT45DB041D

Pro ukládání různých uživatelských nastavení byla vybrána sériová dataflash paměť od firmy Atmel AT45DB041D. Důvod výběru této paměti byla její nízká cena a dostatečná kapacita s výhledem na možné širší využití IP relé. Jedná se o 4MBit paměť, napájena může být 2,7V-3.6V. Její maximální hodinová frekvence pro RapidS rozhraní je až 66MHz. Dále pracuje v SPI modech 0 nebo 3 do kmitočtu 20MHz. Výrobce garantuje [4] minimálně 100 000 zápisů.

Paměť obsahuje 2 SRAM data bufery velikosti 256/264 bytů. Tyto bufery jsou schopny přijímat data v době, kdy probíhá zapisování dat do hlavní paměti. Hlavní paměť je rozdělena na 2048 stránek (pages) po 256/264 bytech. Součástka je v pouzdru SO8 a obsahuje piny:

- SI - vstupní data
- SO - výstupní data
- SCK - hodinový signál
- CS - chip select, aktivní v úrovni logické 0
- WP- write protect, aktivní v úrovni logické 0
- RESET - pro resetování obvodu, aktivní v úrovni logické 0
- Vcc - napájecí napětí
- GND – napájecí napětí

Block Diagram

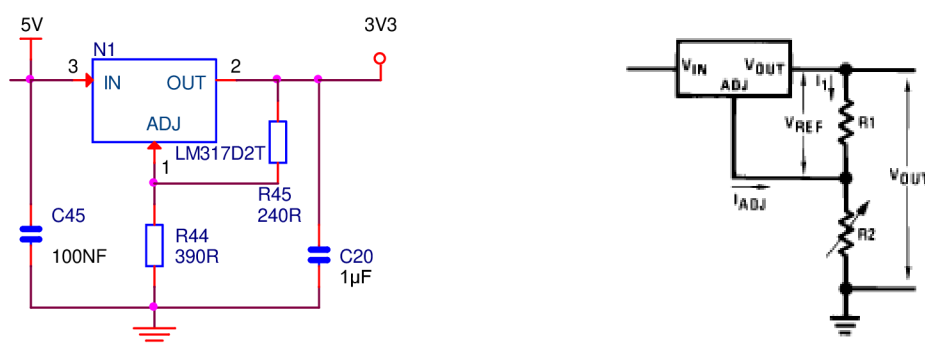


Obr. 4 Blokový diagram paměti AT45DB041D [4]

2.4. Napájení zařízení

Napájení IP relé je řešeno dle požadavku vyplývajícího ze zadání z AC/DC adaptéru 230V/+5V a dále je možné zařízení napájet z USB rozhraní +5V, přičemž v případě připojení obou zdrojů, bude mít vyšší prioritu +5V adaptér. V zařízení je regulátor LM317 firmy National Semiconductor. Slouží k transformaci napětí +5V na +3.3V pro napájení digitálních obvodů.

Zde uvádím schema zapojení regulátoru s výpočtem hodnot odporů dle [5]



Obr. 5 Zapojení regulátoru LM317 v mém zařízení a dle datasheetu výrobce [5]

$$V_{OUT} = V_{REF} \left(1 + \frac{R2}{R1} \right) + I_{ADJ} R2 \quad [V] \quad 1.1$$

$$V_{OUT} = V_{REF} + \frac{V_{REF} R2}{R1} + I_{ADJ} R2$$

$$V_{OUT} - V_{REF} = R2 \left(\frac{V_{REF}}{R1} + I_{ADJ} \right)$$

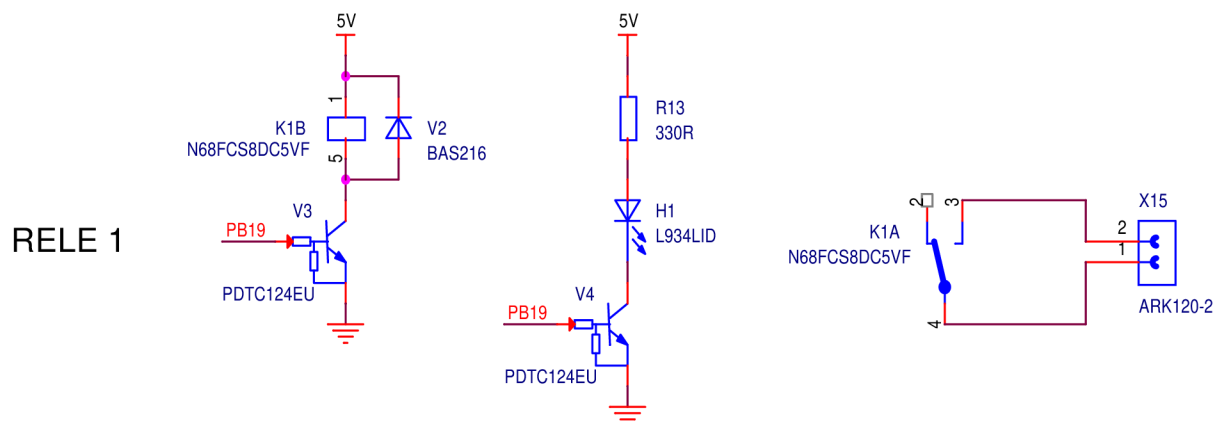
$$R2 = \frac{V_{OUT} - V_{REF}}{\frac{V_{REF}}{R1} + I_{ADJ}} \quad [\Omega] \quad 1.2$$

Po úpravě referencí součástí vzhledem k referencím v mém schématu je tedy $R2 = R44$ a $R1 = R45$. $R45$ bylo zvoleno dle doporučení výrobce 240ohmů. Hodnota referenčního napětí označeného ve vzorci 1.1 a 1.2 jako V_{REF} je dle datasheetu 1,25V a hodnota proudu I_{ADJ} je dle datasheetu výrobce 50uA. Po úpravě vzorce 1.1 získáme vzorec 1.2 pro výpočet rezistoru $R2$, respektive $R44$. Po dosazení všech hodnot tedy získáme velikost rezistoru $R44$ 390R.

2.5. Výstupní zařízení

Na výstupu zařízení jsou osazena relátka firmy Forward Relays N68FCS8DC5VF. Tyto relátka byla vybrána, protože jejich spínací kontakty umožňují spínat 8A/230VAC a nebo 8A/30VDC což přesahuje hodnoty dané zadáním a navíc rozšiřuje možné uplatnění IP relé. Dále jejich spínací cívka je ovládána napětím 5V, což je napájecí napětí IP relé, a proto již není nutné dále toto napětí upravovat.

Ovládání cívek relétek z jednotlivých výstupů je řízeno pomocí tranzistorů PDTC124EU od firmy NXP Semiconductors s vestavěným děličem 1:1 (22kohmů/22kohmů). Stejně je ovládána i signalizace pomocí LED diod pro jednotlivé výstupy. Zde uvádím schéma zapojení výstupního relé a jeho signalizace stavu sepnutí.



Obr. 6 Zapojení relé na výstupu a jeho signalizace

3 Softwarová část zařízení

3.1 RTOS –operační systém reálného času

Využití systému pro řízení v reálném čase RTOS (Real Time Operating System) je velice časté v oblasti řízení technologických procesů, kde se jedná často o kritické úlohy reálného času (např. antiblokovací systém u vozidel) nebo nekritické systémy reálného času (např. rezervace místenek v dopravě). Velké využití má RTOS ovšem i v oblasti vestavěných systémů (Embedded zařízení). Aplikační oblasti použití vestavěných řídicích systémů s mikropočítači se neustále rozšiřuje a v současné době je lze nalézt ve všech oblastech lidské činnosti.

3.1.1 FreeRTOS

Na mnou navrhovaném IP relé je implementován volně dostupný operační systém reálného času FreeRTOS[7]. Důvodem výběru tohoto RTOS byl hotový web server s uIP taskem, do kterého je možné přidávat další funkce. Mezi jeho hlavní výhody patří:

- má minimální nároky na využití paměti
- je velmi jednoduchý, jeho jádro tvoří pouze 3 .c soubory.
- je možné ho použít i v komerčních zařízeních
- je vhodným řešením pro různé typy architektury mikrokontrolerů včetně architektury ARM7
- je pevně zavedeným systémem a se stále rostoucí uživatelskou základnou

Z webových stránek projektu FreeRTOS[7] byl stáhnut balíček s aktuální verzí FreeRTOS V6.1.1.

Velice užitečné informace obsahuje adresář Demo, ve kterém je příklad implementace tohoto operačního systému pro mikrokontrolér AT91SAM7X, z něhož bylo čerpáno.

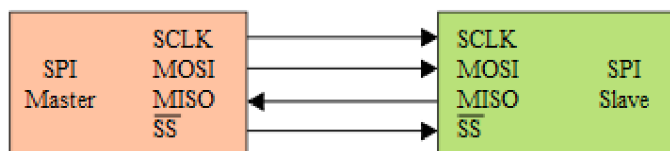
4 Nastavení mikrokontroleru

V navrhovaném IP relé je mikrokontrolér použit pro ovládání výstupů (relátek) a také pro komunikaci pomocí sériového rozhraní SPI s připojenou sériovou pamětí, do které jsou ukládány nastavení zadávané uživatelem.

4.1. Nastavení sériové komunikace

SPI (*Serial Peripheral Interface*) je sériové periferní rozhraní [8]. Používá se pro komunikaci mezi řídicími mikroprocesory a ostatními integrovanými obvody (EEPROM, A/D převodníky, displeje...). Komunikace je realizována pomocí společné sběrnice. Vždy je jedno zařízení Master a druhé Slave. Na sběrnici může být připojeno více zařízení. Adresace mezi nimi se provádí pomocí zvláštních vodičů, které při logické nule aktivují příjem a vysílání zvoleného

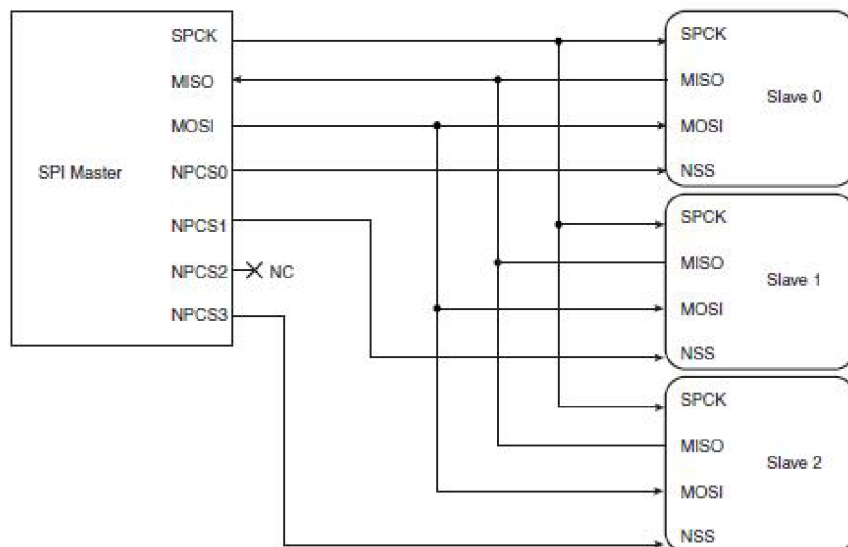
zařízení (piny SS nebo CS). Pin SCLK slouží pro hodinové pulsy, pin MOSI (Master Output Slave Input) slouží k odesílání dat od zařízení Master k zařízení Slave a pin MISO (Master Input Slave Output) je vstupním pinem řídicího obvodu Master a výstupním obvodem zařízení Slave. Na obrázku níže je šipkami naznačen směr signálů mezi komunikujícími zařízení na SPI sběrnici.



Obr. 7 Příklad zapojení dvou zařízení pomocí SPI [8]

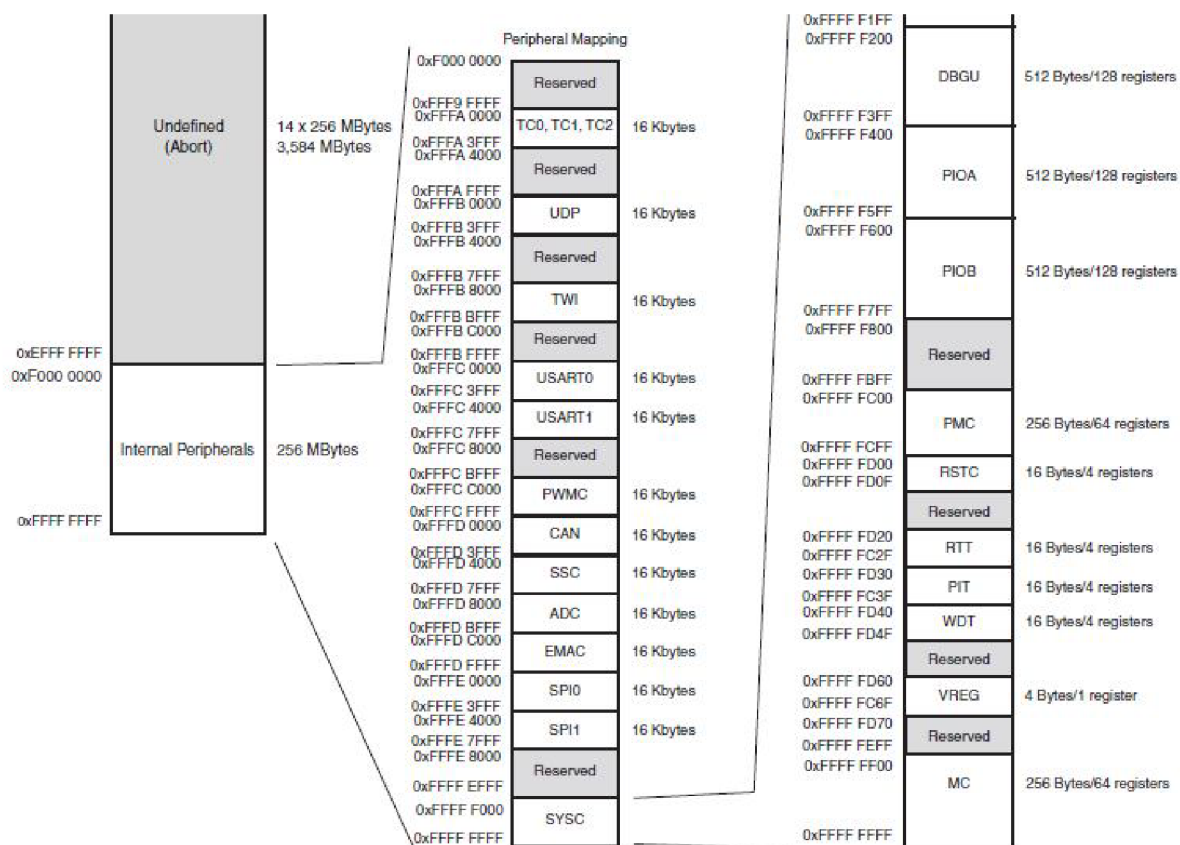
4.1.1 Nastavení sériové komunikace – mikrokontroler

V IP relé je mikrokontroler nastaven jako Master a podřízeným zařízením Slave je paměť AT45DB040D. Mikrokontroler disponuje dvěma periferiemi pro seriovou komunikaci SPI0 a SPI1. V tomto zařízení je použita periferie SPI0. Má čtyři piny NPCS(0-3) pro adresaci až čtyřech různých slave zařízení.



Obr. 8 Příklad zapojení 3 zařízení mikrokontroleru AT91SAM7X512 pomocí SPI [2]

Vývody pro periferii SPI0 se musí nejdříve nastavit pomocí registrů PIO kontroleru, poněvadž mikrokontroler má některé piny použity pro více funkcí. A to jako piny vstupní/výstupní (I/O), nebo mohou být nastaveny jako vývody periferie A nebo B. Toto platí i pro SPI0. Veškeré periferie mikrokontroleru mají vyhrazenou paměť 256MB a na každou z nich připadá 16kB.



Obr. 9 Rozdělení paměti pro periferie AT91SAM7X512 [2]

Jednotlivé periferie jsou identifikovány podle ID čísel. SPI0 je definována jako číslo 4, periferie A. Jak je uvedeno v Tab. 1. Piny pro SPI komunikaci nastavíme v registru PIO_ASR PIO kontroleru.

```
AT91C_BASE_PIOA->PIO_ASR = ((unsigned int) AT91C_PA16_SPI0_MISO ) |
((unsigned int) AT91C_PA17_SPI0_MOSI ) |
((unsigned int) AT91C_PA18_SPI0_SPCK ) |
((unsigned int) AT91C_PA12_SPI0_NPCS0 );
```

Hodinový signál je řízen Power Management Controllerem PMC, a to konkrétně pomocí zapsání do registru PMC_PCER (Peripheral Clock Enable Register).

```
AT91C_BASE_PMC->PMC_PCER = 1 << AT91C_ID_SPI0;
```

Vlastní povolení SPI komunikace se provádí pomocí zápisu do Control Registru SPI_CR.

```
AT91C_BASE_SPI0->SPI_CR = AT91C_SPI_SPIEN;
```

Definování mikrokontroleru jako Master zařízení umožňuje registr SPI_MR. V tomto registru se také nastavuje počet Slave zařízení. Jelikož v navrhovaném IP relé je pouze jedno zařízení, jsou nastaveny ještě bity Peripheral Chip Select (PCS) pro vývod NPCS0.

```
AT91C_BASE_SPI0->SPI_MR = AT91C_SPI_MSTR | AT91C_SPI_PS_FIXED |
AT91C_SPI_MODFDIS;
```

Peripheral ID	Peripheral Mnemonic	Peripheral Name	External Interrupt
0	AIC	Advanced Interrupt Controller	FIQ
1	SYSC ⁽¹⁾	System Controller	
2	PIOA	Parallel I/O Controller A	
3	PIOB	Parallel I/O Controller B	
4	SPI0	Serial Peripheral Interface 0	
5	SPI1	Serial Peripheral Interface 1	
6	US0	USART 0	
7	US1	USART 1	
8	SSC	Synchronous Serial Controller	
9	TWI	Two-wire Interface	
10	PWMC	Pulse Width Modulation Controller	
11	UDP	USB Device Port	
12	TC0	Timer/Counter 0	
13	TC1	Timer/Counter 1	
14	TC2	Timer/Counter 2	
15	CAN	CAN Controller	
16	EMAC	Ethernet MAC	
17	ADC ⁽¹⁾	Analog-to Digital Converter	
18 - 29	Reserved		
30	AIC	Advanced Interrupt Controller	IRQ0
31	AIC	Advanced Interrupt Controller	IRQ1

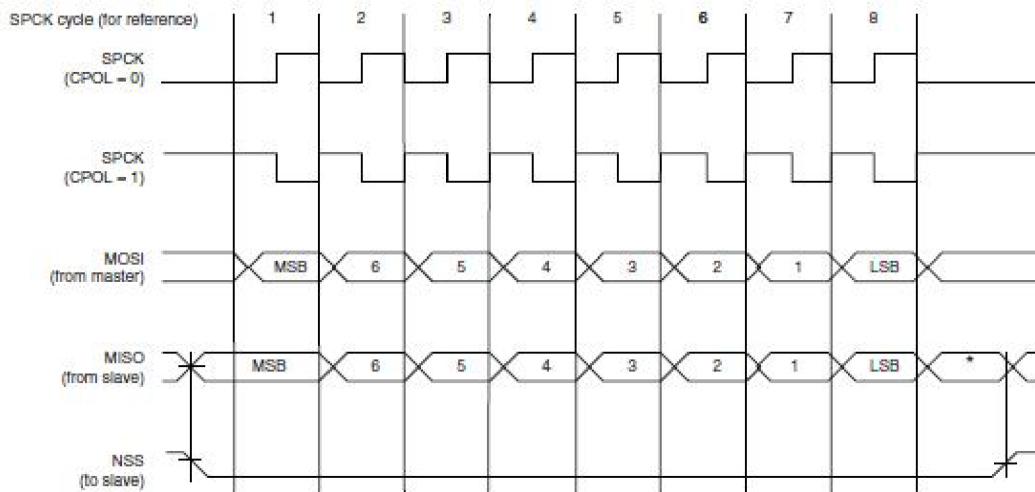
Tab. 1 Identifikace jednotlivých periférií AT91SAM7X512 [2]

Mikrokontroler umožňuje čtyři módy SPI komunikace značené jako módy SPI 0 až SPI 3.

SPI Mode	CPOL	NCPHA
0	0	1
1	0	0
2	1	1
3	1	0

Tab. 2 Identifikace jednotlivých periférií AT91SAM7X512 [2]

V navrhovaném IP relé byl nastaven mód SPI 2. Toto nastavení se provádí zápisem do registru Chip Select Registru SPI_CSR[0]. Zde byly nastaveny bity CPOL a NCPHA na 1.



Obr. 10 Průběh signálů SPI při nastavení NCPHA=1 [2]

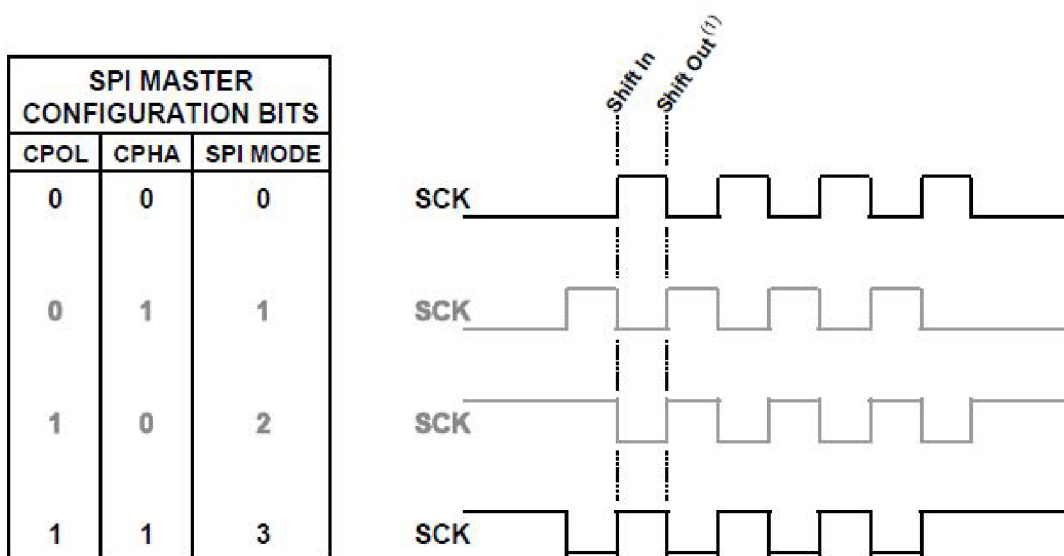
Jak je z Obr. 10 patrné, nastavením CPOL=1 jsme nastavili úroveň logické 1 v neaktivním stavu pro vývod s hodinovým signálem SPCK a nastavením NCPHA=1 byla nastavena fáze hodinového signálu, která určuje, na kterou hranu budou data posouvána a na kterou vzorkována. V tomto případě tedy při nastavení NCPHA a CPOL na 1 jsou data posouvána s padající hranou a vzorkována s hranou náběžnou.

Dále zde ještě byl nastaven počet bitů v poli Bits per Transfer (BITS) na 8 bitů.

```
AT91C_BASE_SPI0->SPI_CSR[0] = AT91C_SPI_CPOL | AT91C_SPI_NCPHA |
AT91C_SPI_CSAAT | AT91C_SPI_BITS_8 | (4 << 8);
```

4.1.2 Nastavení sériové komunikace – paměť AT45DB041D

Sériová paměť AT45DB041D firmy Atmel může s mikrokontrolerem komunikovat po SPI sběrnici v modech SPI 0 nebo SPI 3. Zde je nesrovnalost mezi mody, poněvadž dle datasheetu paměti [4] jsou mody nastaveny tak, jak je vidět na obr. 11, to znamená že mód 3 má CPOL i CPHA =1, což bylo v datasheetu AT91SAM7X512 [4] uvedeno jako SPI mód 2.



Obr. 11 SPI módy dle datasheetu AT45DB041D [4]

Paměť provádí své operace s daty na základě instrukcí z mikrokontroléru. Tyto instrukce jsou zadávány pomocí speciálních příkazů, tzv. OpCodu. Tyto příkazy jsou definovány jako 1 Bytová slova. Celková instrukce se skládá tedy z OpCodu, dále následují 4 rezervované bity (je doporučeno nastavit 0000), a poté jsou posílány 3 adresové byty. Po nich jsou někdy požadovány ještě 4 byty tzv. Don't care bitů nebo jsou již posílány rovnou data.

Opcode	Opcode	Address Byte				Address Byte				Address Byte				Additional Don't Care Bytes Required								
		Reserved	Reserved	Reserved	Reserved	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3		PA2	PA1	PA0	BA3	BA2	BA1	BA0	
50H	0 1 0 1 0 0 0 0	r	r	r	r	P	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	N/A
52H	0 1 0 1 0 0 1 0	r	r	r	r	P	P	P	P	P	P	P	P	B	B	B	B	B	B	B	B	4 Bytes
53H	0 1 0 1 0 0 1 1	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
54H	0 1 0 1 0 1 0 0	x	x	x	x	x	x	x	x	x	x	x	B	B	B	B	B	B	B	B	B	1 Byte
55H	0 1 0 1 0 1 0 1	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
56H	0 1 0 1 0 1 1 0	x	x	x	x	x	x	x	x	x	x	B	B	B	B	B	B	B	B	B	B	1 Byte
57H	0 1 0 1 0 1 1 1	N/A				N/A				N/A				N/A								
58H	0 1 0 1 1 0 0 0	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
59H	0 1 0 1 1 0 0 1	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
60H	0 1 1 0 0 0 0 0	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
61H	0 1 1 0 0 0 0 1	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
68H	0 1 1 0 1 0 0 0	r	r	r	r	P	P	P	P	P	P	P	B	B	B	B	B	B	B	B	B	4 Bytes
81H	1 0 0 0 0 0 0 1	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
82H	1 0 0 0 0 0 1 0	r	r	r	r	P	P	P	P	P	P	P	B	B	B	B	B	B	B	B	B	N/A
83H	1 0 0 0 0 0 1 1	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
84H	1 0 0 0 0 1 0 0	x	x	x	x	x	x	x	x	x	x	B	B	B	B	B	B	B	B	B	B	N/A
85H	1 0 0 0 0 1 0 1	r	r	r	r	P	P	P	P	P	P	P	B	B	B	B	B	B	B	B	B	N/A
86H	1 0 0 0 0 1 1 0	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
87H	1 0 0 0 0 1 1 1	x	x	x	x	x	x	x	x	x	x	B	B	B	B	B	B	B	B	B	B	N/A
88H	1 0 0 0 1 0 0 0	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
89H	1 0 0 0 1 0 0 1	r	r	r	r	P	P	P	P	P	P	P	x	x	x	x	x	x	x	x	x	N/A
D2H	1 1 0 1 0 0 1 0	r	r	r	r	P	P	P	P	P	P	P	B	B	B	B	B	B	B	B	B	4 Bytes
D4H	1 1 0 1 0 1 0 0	x	x	x	x	x	x	x	x	x	x	B	B	B	B	B	B	B	B	B	B	1 Byte
D6H	1 1 0 1 0 1 1 0	x	x	x	x	x	x	x	x	x	B	B	B	B	B	B	B	B	B	B	B	1 Byte
D7H	1 1 0 1 0 1 1 1	N/A				N/A				N/A				N/A								
E8H	1 1 1 0 1 0 0 0	r	r	r	r	P	P	P	P	P	P	P	B	B	B	B	B	B	B	B	B	4 Bytes

Note: r = Reserved Bit
P = Page Address Bit
B = Byte/Buffer Address Bit
x = Don't Care

Tab. 3 Seznam OpCodu paměti AT45DB041D [4]

V navrhovaném zařízení byly využívány dva typy instrukcí. A to pro zápis dat a pro čtení dat.

Command	SCK Mode	Opcode
Continuous Array Read	Inactive Clock Polarity Low or High	68H
	SPI Mode 0 or 3	E8H
Main Memory Page Read	Inactive Clock Polarity Low or High	52H
	SPI Mode 0 or 3	D2H
Buffer 1 Read	Inactive Clock Polarity Low or High	54H
	SPI Mode 0 or 3	D4H
Buffer 2 Read	Inactive Clock Polarity Low or High	56H
	SPI Mode 0 or 3	D6H
Status Register Read	Inactive Clock Polarity Low or High	57H
	SPI Mode 0 or 3	D7H

Tab. 4 Instrukce pro čtení paměti AT45DB041D [4]

Command	SCK Mode	Opcode
Buffer 1 Write	Any	84H
Buffer 2 Write	Any	87H
Buffer 1 to Main Memory Page Program with Built-in Erase	Any	83H
Buffer 2 to Main Memory Page Program with Built-in Erase	Any	86H
Buffer 1 to Main Memory Page Program without Built-in Erase	Any	88H
Buffer 2 to Main Memory Page Program without Built-in Erase	Any	89H
Page Erase	Any	81H
Block Erase	Any	50H
Main Memory Page Program through Buffer 1	Any	82H
Main Memory Page Program through Buffer 2	Any	85H

Tab. 5 Instrukce pro zápis do paměti AT45DB041D [4]

Z uvedených instrukcí pro čtení z a zapisování do paměti byla vybrána instrukce Main Memory Page Read pro čtení dat a Main Memory Page Program through Buffer 1 pro zápis dat do paměti.

Při zadávání instrukce pro čtení dat byl potřeba dodržet přesný sled bytů posílaných z mikrokontroléru, tak jak je uvedeno v Tab. 3. To znamená nejprve je poslán 1 byte s opcodem D2H (0xD2), po něm následují 4 rezervované bity (0000), tyto jsou následovány 20 adresovými bity, z nichž 11 bitů je adresa stránky v paměti, ze které budeme číst a 9 bitů je adresa prvního bytu na této stránce, od kterého budeme číst. Za těmito adresovými byty se posílají ještě 4 Don't care byty a poté již směřují data z paměti směrem k mikrokontroleru.

Při zadávání instrukce pro zápis dat je nutné nejprve poslat 1 byt s OpCodem 82H (0x82) následovanými 3 adresovými byty, z nichž opět první 4 bity jsou rezervovány a je doporučeno zadat 0000, dále následuje 11 adresových bitů definující adresu stránky, do které bude zapisováno, a 9 bitů určující adresu prvního bytu v bufferu. Poté je možné posílat z mikrokontroleru data , které chceme uložit do paměti. Tato instrukce před vlastním zápisem provede nejdříve vymazání předchozích dat uložených na stejné adrese. Veškeré tyto funkce jsou obsaženy v souboru memory.c

5 Ovládání zařízení

Ovládání zařízení je řešeno dle zadání přes webové rozhraní a resetovacího tlačítka, které umožní uživateli nastavit zařízení do defaultního stavu, s defaultním heslem, IP adresou a nastavenou maskou.

5.1. Uživatelské webové rozhraní

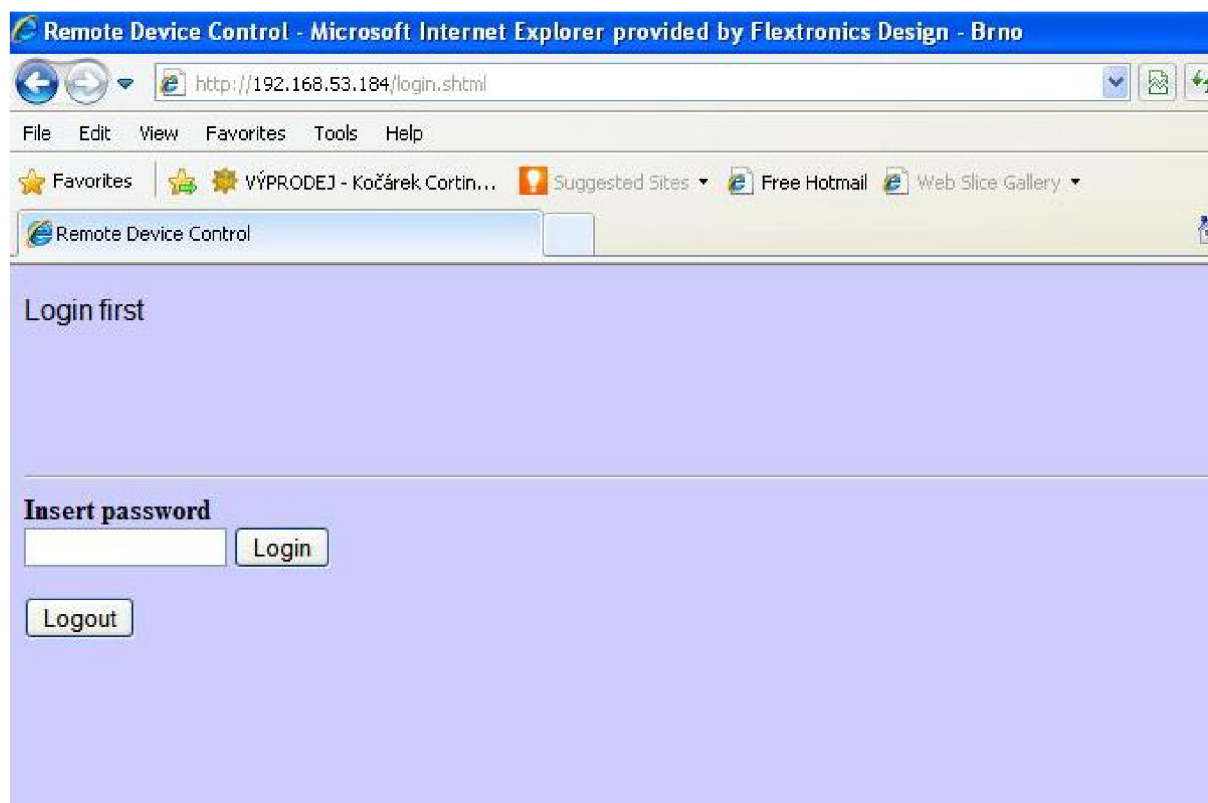
Uživatel bude s IP relém komunikovat pomocí webového prohlížeče. IP adresa je zadána v sériové paměti a po každém zapnutí zařízení, je nejprve tato adresa vyčtena z paměti a nastavena.

Po zadání IP adresy zařízení bude uživatel vyzván k zadání hesla. Po zadání hesla proběhne kontrola s nastaveným heslem, a to tak, že v odeslaném řetězci bude hledán řetězec Password a z něj se následující text za řetězcem Password= porovná s heslem nastaveným, které je uloženo v sériové paměti. Zde je uvedena část zdrojového kódu:

```

if( strstr( pcInput, "Password=" ) != NULL )
{
    at45db_read( 0x00, 0x10, 0x01, chng_pass, 11); //vyčtení dat z paměti
    pass = strstr( pcInput, "Password=" ) + 9; // zapsání hesla do proměnné pass
    if (strncmp(pass, change_pass, 10)) // vyhodnocení, zda je heslo správné
    {
        logged = 0;
    }
    else
    {
        logged = 1;
    }
}
}

```

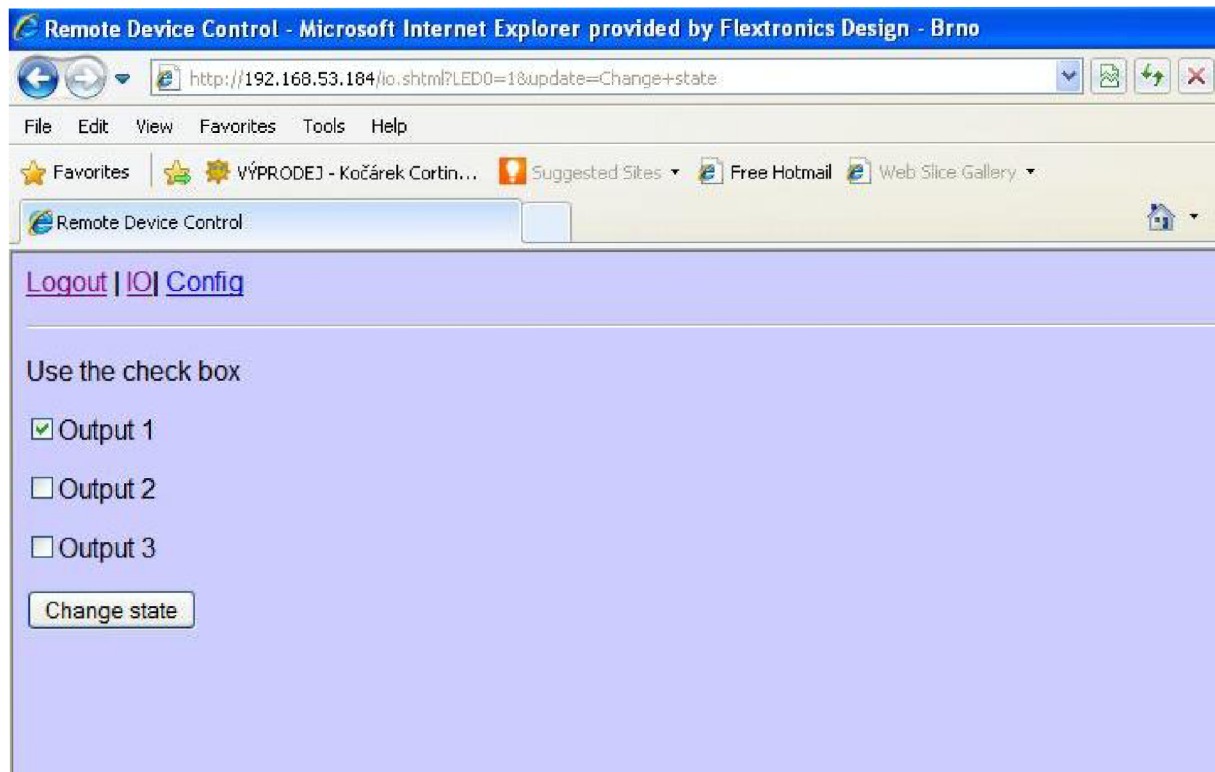


Obr. 12. Úvodní stránka pro zadání hesla

Po správném zadání hesla se objeví záložky, kde je možné provést konfiguraci zařízení, jako je změna hesla, IP adresy a nastavení masky viz Obr.13. Další záložka slouží pro nastavování stavů jednotlivých výstupů. Zároveň je zde zobrazený aktuální stav výstupů. Uživatel vždy zatrhne, které relé chce sepnout a stiskne softwarové tlačítko Change state. Poté dojde k sepnutí relé na výstupu a zároveň k rozsvícení signalizační LED diody daného výstupu viz Obr.14.



Obr. 13. Stránka pro uživatelské nastavení



Obr. 14. Stránka pro nastavení výstupů

Jednotlivé stránky jsou vytvořeny jako HTML stránky. Tyto stránky se po vytvoření převedou pomocí skriptu makefsdata a makestrings, napsaných v programovacím jazyce Perl, do

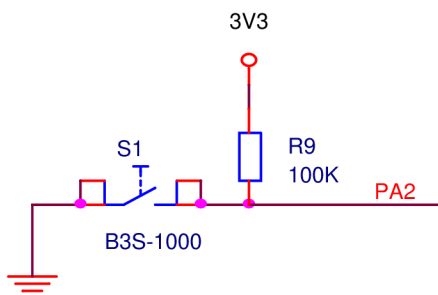
hexadecimálního kódu a tento je uložen v http-fsdata.c . Pro představu na Obr. 15 je část stránky login.shtml

```
static const unsigned char data_login_shtml[] = {
    /* /login.shtml */
    0x2f, 0x6c, 0x6f, 0x67, 0x69, 0x6e, 0x2e, 0x73, 0x68, 0x74, 0x6d, 0x6c, 0,
    0x3c, 0x21, 0x44, 0x4f, 0x43, 0x54, 0x59, 0x50, 0x45, 0x20,
    0x48, 0x54, 0x4d, 0x4c, 0x20, 0x50, 0x55, 0x42, 0x4c, 0x49,
    0x43, 0x20, 0x22, 0x2d, 0x2f, 0x2f, 0x57, 0x33, 0x43, 0x2f,
    0x2f, 0x44, 0x54, 0x44, 0x20, 0x48, 0x54, 0x4d, 0x4c, 0x20,
    0x34, 0x2e, 0x30, 0x31, 0x20, 0x54, 0x72, 0x61, 0x6e, 0x73,
    0x69, 0x74, 0x69, 0x6f, 0x6e, 0x61, 0x6c, 0x2f, 0x2f, 0x45,
    0x4e, 0x22, 0x20, 0x22, 0x68, 0x74, 0x74, 0x70, 0x3a, 0x2f,
    0x2f, 0x77, 0x77, 0x77, 0x2e, 0x77, 0x33, 0x2e, 0x6f, 0x72,
    0x67, 0x2f, 0x54, 0x52, 0x2f, 0x68, 0x74, 0x6d, 0x6c, 0x34,
    0x2f, 0x6c, 0x6f, 0x6f, 0x73, 0x65, 0x2e, 0x64, 0x74, 0x64,
    0x22, 0x3e, 0xa, 0x3c, 0x68, 0x74, 0x6d, 0x6c, 0x3e, 0xa,
    0x20, 0x20, 0x3c, 0x68, 0x65, 0x61, 0x64, 0x3e, 0xa, 0x20,
    0x20, 0x20, 0x20, 0x3c, 0x74, 0x69, 0x74, 0x6c, 0x65, 0x3e,
    0x52, 0x65, 0x6d, 0x6f, 0x74, 0x65, 0x20, 0x44, 0x65, 0x76,
    0x69, 0x63, 0x65, 0x20, 0x43, 0x6f, 0x6e, 0x74, 0x72, 0x6f,
    0x6c, 0x3c, 0x2f, 0x74, 0x69, 0x74, 0x6c, 0x65, 0x3e, 0xa,
    0x20, 0x20, 0x3c, 0x2f, 0x68, 0x65, 0x61, 0x64, 0x3e, 0xa,
    0x20, 0x20, 0x3c, 0x42, 0x4f, 0x44, 0x59, 0x20, 0x62, 0x67,
    0x63, 0x6f, 0x6c, 0x6f, 0x72, 0x3d, 0x22, 0x23, 0x43, 0x43,
    0x43, 0x43, 0x66, 0x66, 0x22, 0x3e, 0xa, 0x3c, 0x66, 0x6f,
```

Obr. 15. Stránka login.shtml po převedení do hexadecimálního kódu

5.2. Resetovací tlačítko

Resetovací tlačítko slouží uživateli k nastavení zařízení do defaultního stavu. Tlačítko je připojeno k vývodu PA2, který je nastaven jako I/O vývod. Zařízení neustále snímá úroveň na jeho vstupu. Tento je připojen přes rezistor na napájecí napětí 3,3V. Po jeho stisknutí se na jeho vstupu objeví logická 0 a dojde k přepsání původních informací o nastavení v sériové paměti. To znamená, že se přepíše IP adresa, maska a přístupové heslo. Zároveň se výstupy nastaví jako nesepté.



Obr. 16. Zapojení resetovacího tlačítka

Poté je nastaven softwarový reset zařízení zapsáním do registru reset kontrol (RSTC) a zařízení se znovu spustí s již změněným nastavením. V defaultním stavu je IP adresa nastavena jako 192.168.53.195, maska 255.255.255.0, heslo = heslo a stavy všech výstupů jsou rozepnuté.

Zde je část zdrojového kódu pro nastavení do defaultního stavu:

```
at45db_write( 0x00, 0x00, 0x00, def, 8);           //Nastavení defaultní IP adresy a masky
at45db_write( 0x00, 0x10, 0x00, def_password, 5); // Nastavení defaultního hesla
at45db_write( 0x00, 0x04, 0x00, def_outstat, 3); // Nastavení defaultních stavů na výstupech
```

```
AT91C_BASE_RSTC->RSTC_RCR = (0xA5 << 24) | AT91C_RSTC_PROCRST;
// SW reset
```

6 Závěr

Cílem této práce bylo navrhnout zařízení s napájecím napětím 5V nebo 12V, které bude schopné ovládat 3 výstupy přes webové rozhraní, zobrazovat jejich stav a uživatel bude mít možnost nastavit si IP adresu, masku a přístupové heslo. Dále také bude možné nastavit zařízení do defaultního stavu a při výpadku napájení zůstanou zachovány nastavené hodnoty. Napájecí napětí bylo vybráno jako 5V a zároveň bylo umožněno IP relé napájet pomocí USB konektoru. Jako další zlepšení do budoucna by bylo možné napájení upravit tak, aby bylo možné IP relé napájet i po ethernetu. Pro ukládání uživatelských nastavení byla využita seriová dataflash paměť, do které se také ukládají stavy výstupů, aby bylo možné obnovení po výpadku napájecího napětí. Její paměť je dostatečná i pro budoucí možné využití. Stejně tak i díky velkému počtu IO pinů mikrokontroléru, lze uvažovat o možnosti spínat více zařízení. IP relé je možné použít všude tam, kde je nutné občas provést restart nějakého zařízení nebo jeho odpojení či zapnutí.

Během návrhu IP relé jsem se podrobněji zabýval mikrokontrolérem firmy Atmel s architekturou ARM a jeho využitím v projektu pracujícím přes ethernetové rozhraní s protokolem TCP/IP. Vycházel jsem z vývojového kitu firmy Kramara a vlastní software jsem implementoval do operačního systému reálného času FreeRTOS.

Z hardwarové části jsem řešil především napájení digitálních obvodů a zapojení výstupních relátek a signalizačních LED diod. Provedl jsem návrh jednoduchého regulátoru napětí pro převod napájecího napětí 5V na 3,3V. U výstupů jsem se snažil o jejich co nejmenší výstupní proudové zatížení, a proto jsem zvolil spínání výstupních relátek a signalizačních LED diod pomocí spínacích tranzistorů. Jelikož se jednalo o jednoduché záležitosti, neobjevil se v této části žádný větší problém.

Softwarová část se skládala z více částí. První část řešila obsluhu mikrokontroléru a jeho komunikaci s ostatními součástkami, především sériovou paměť po SPI sběrnici, ale i ovládání vstupu resetovacího tlačítka a výstupů pro signalizační LED diody a relátka. Jelikož jsem na počátku měl minimální zkušenosti s mikrokontrolery, tak se během této části vyskytovalo řada problémů, jejichž řešením jsem se zároveň učil a získával zkušenosti s prací s mikrokontrolerem. Druhá část řešila implementaci do FreeRTOS. Zde jsem vycházel především ze vzorového příkladu určeného pro mikrokontroler AT91SAM7X512, protože z webových stránek FreeRTOS [7] jsem potřebné informace získával obtížně z důvodu minimálně popsaného API. Díky této části jsem se naučil základům pro práci s tímto operačním systémem reálného času. Třetí částí bylo vlastní vytvoření webových stránek. Zde jsem se snažil hlavně o jednoduchost a přehlednost, aby komunikace se zařízením probíhala co nejjednodušeji.

Celkově tuto práci považuji za velice přínosnou, poněvadž jsem při jejím zpracování narážel na spousty problémů, jejichž vyřešením jsem získal nové a velice užitečné zkušenosti.

LITERATURA

- [1] VÁŇA, V. *ARM pro začátečníky*. Praha: BEN - technická literatura, 2009, ISBN 978-80-7300-246-6
- [2] Atmel. *Datasheet AT91SAM7X512*. 7. April 2011[Online] .[citováno 25.5.2011] Dostupné z WWW http://www.atmel.com/dyn/resources/prod_documents/6120s.pdf
- [3] Davicom. *Datasheet DM9161*. 10. September 2008[Online]. [citováno 25.5.2011] Dostupné z WWW http://www.davicom.com.tw/userfile/24247/DM9161-DS-F05_091008.pdf
- [4] Atmel. *Datasheet AT45DB041D. 09/09* [Online] [citováno 25.5.2011] Dostupné z WWW http://www.atmel.com/dyn/resources/prod_documents/doc3595.pdf
- [5] National Semiconductor. *Datasheet LM317*. 25.February 2011[Online] [citováno 25.5.2011] Dostupné z WWW <http://www.national.com/ds/LM/LM117.pdf>
- [6] NXP Semiconductors. *Datasheet PDTC124EU*.17. August 2004[Online][citováno 25.5.2011] Dostupné z WWW http://www.nxp.com/documents/data_sheet/PDTC124E_SERIES.pdf
- [7] FreeRTOS. [Online]. Real Time Engineers Ltd. 2010-2011, [citováno 25.5.2011] Dostupné z WWW www.freertos.org
- [8] Serial Peripheral Interface, In *Wikipedia: free encyklopedia*, [Online] Wikimedia Česká republika, 2008-, poslední změna 6. 12. 2010 [citováno 25.5.2011], Dostupné z WWW <http://cs.wikipedia.org/wiki/SPI>
- [9] Dvořák Miroslav, *ARM a Linuxu v embedded aplikacích I*. 16. 2.2006[Online] [citováno 25.5.2011] Dostupné z WWW <http://hw.cz/Teorie-a-praxe/Programovani/ART1512-ARM-a-Linuxu-v-embedded-aplikacich-I.html>

Seznam zkratek

ARM – Advanced RISC Machine

CPU - Central Processing Unit

RISC - Reduced Instruction Set Computer

AC/DC - Alternating Current/Direct Current

RTOS - Real-Time Operating System

SPI - Serial Peripheral Interface

EEPROM - Electrically Erasable Programmable Read-Only Memory

LED - Light-emitting diode

API - Application Programming Interface

IP – Internet Protocol

TCP - Transmission Control Protocol

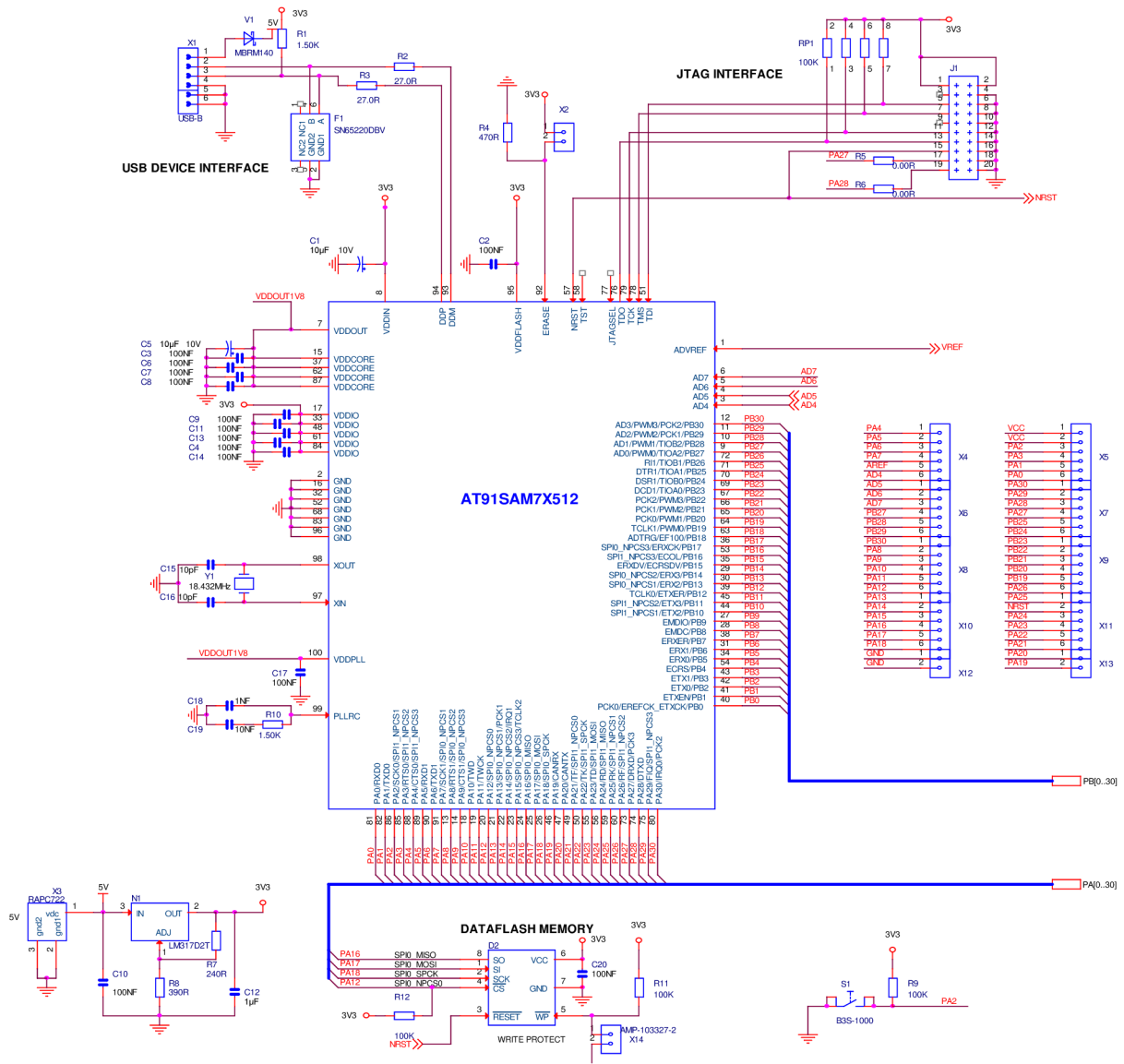
HTML - HyperText Markup Language

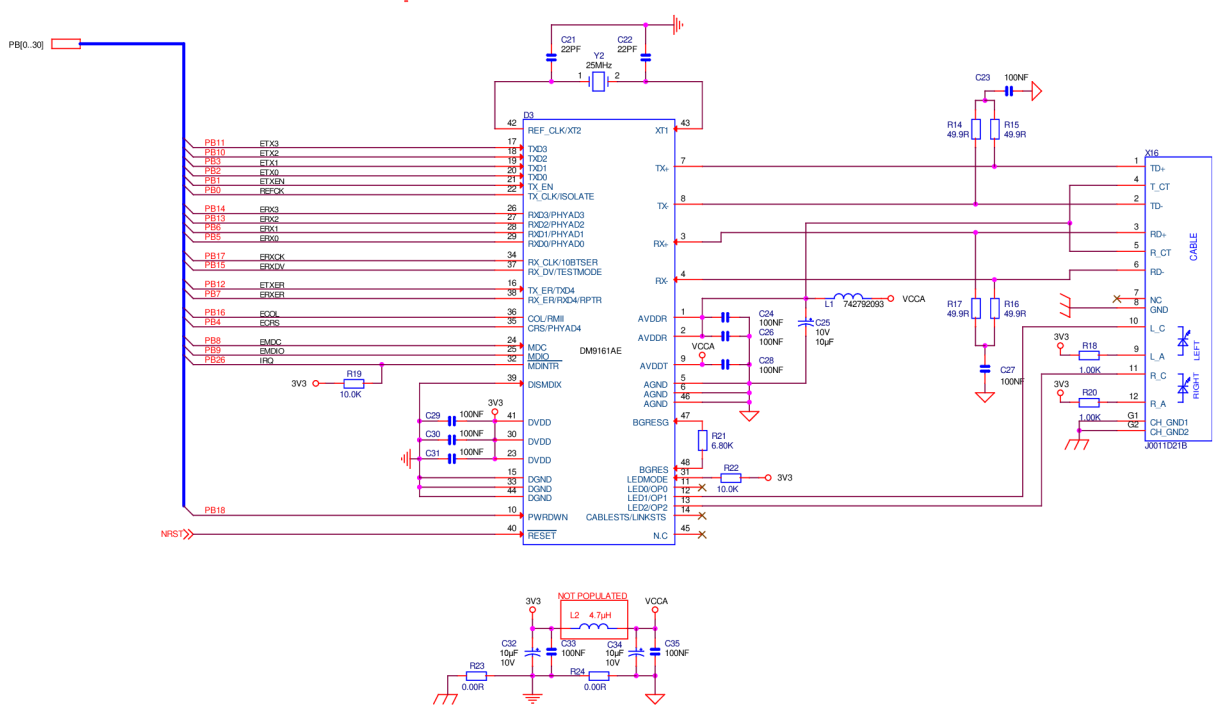
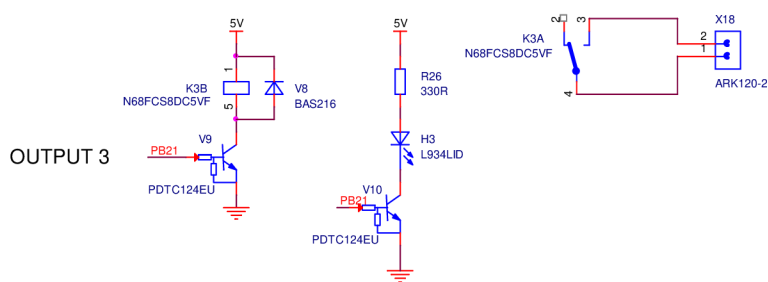
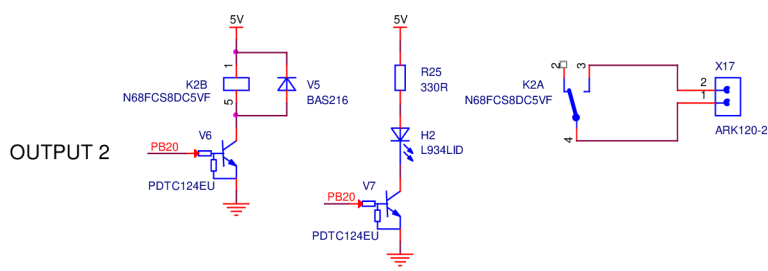
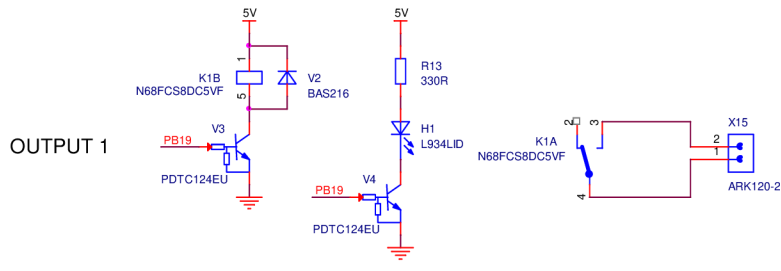
MAC - Media Access Kontrol

PHY - Physical Layer Interface

PŘÍLOHY

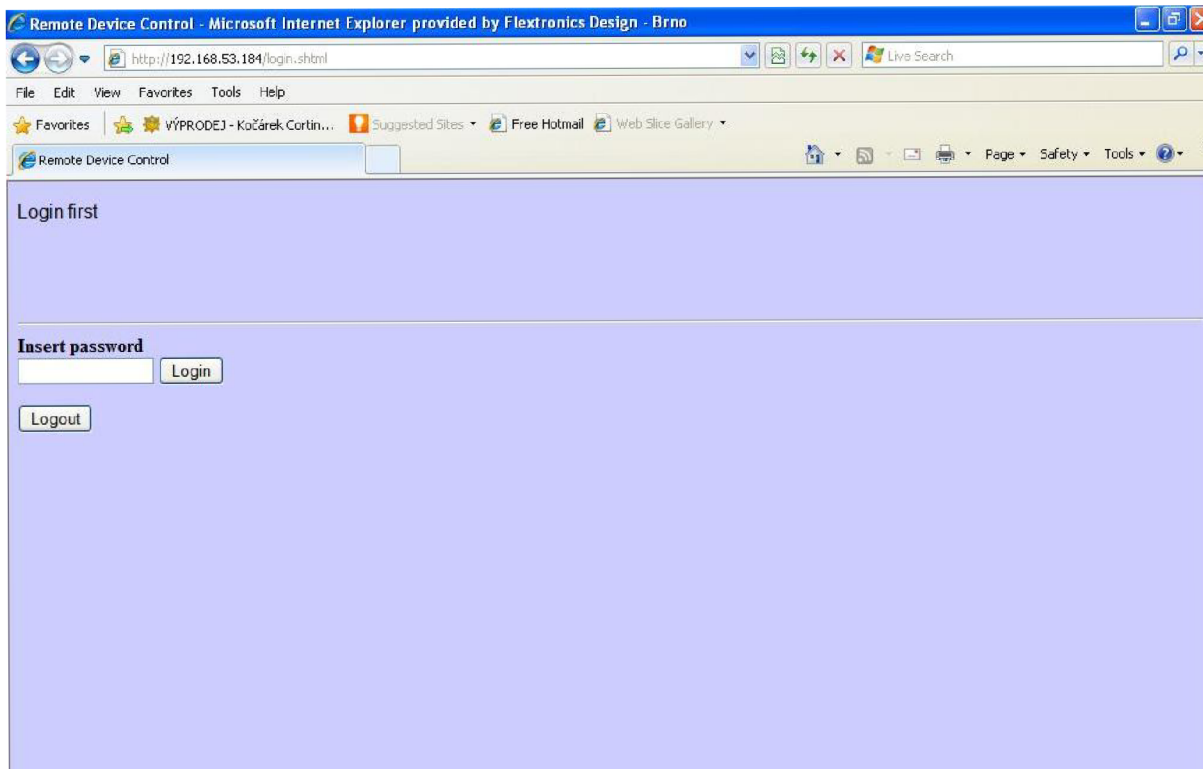
Schema





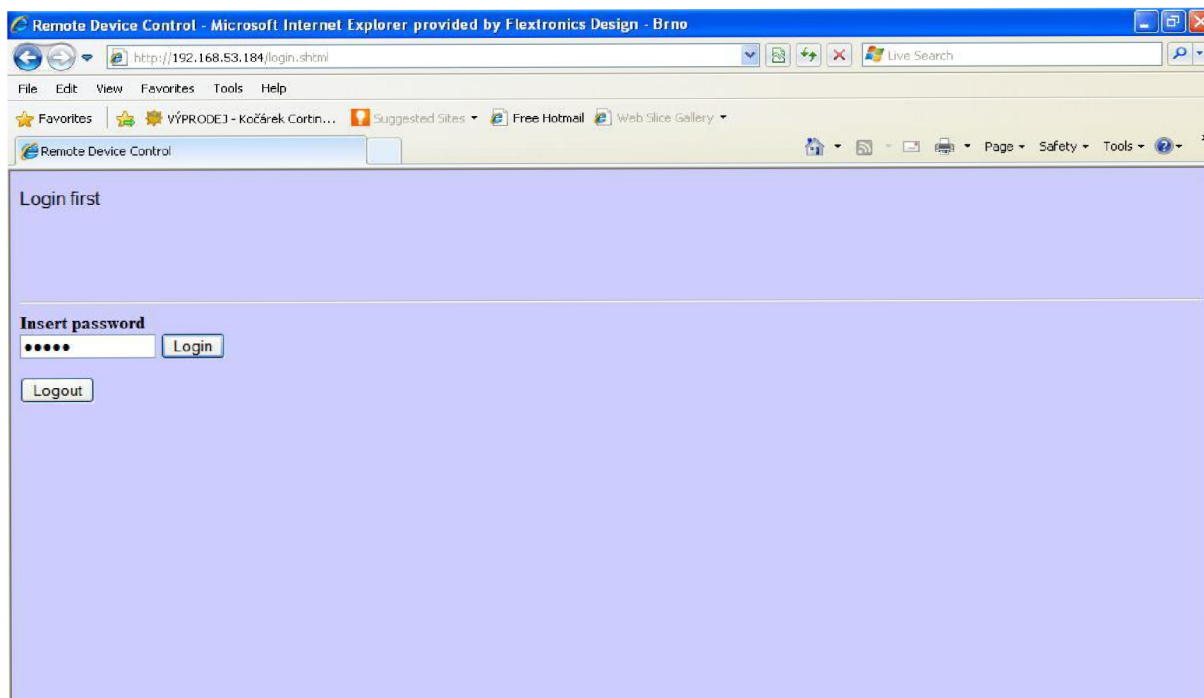
Popis činnosti zařízení z hlediska uživatele

Po zadání IP adresy zařízení do webového prohlížeče se objeví přihlašovací stránka viz Obr. 17



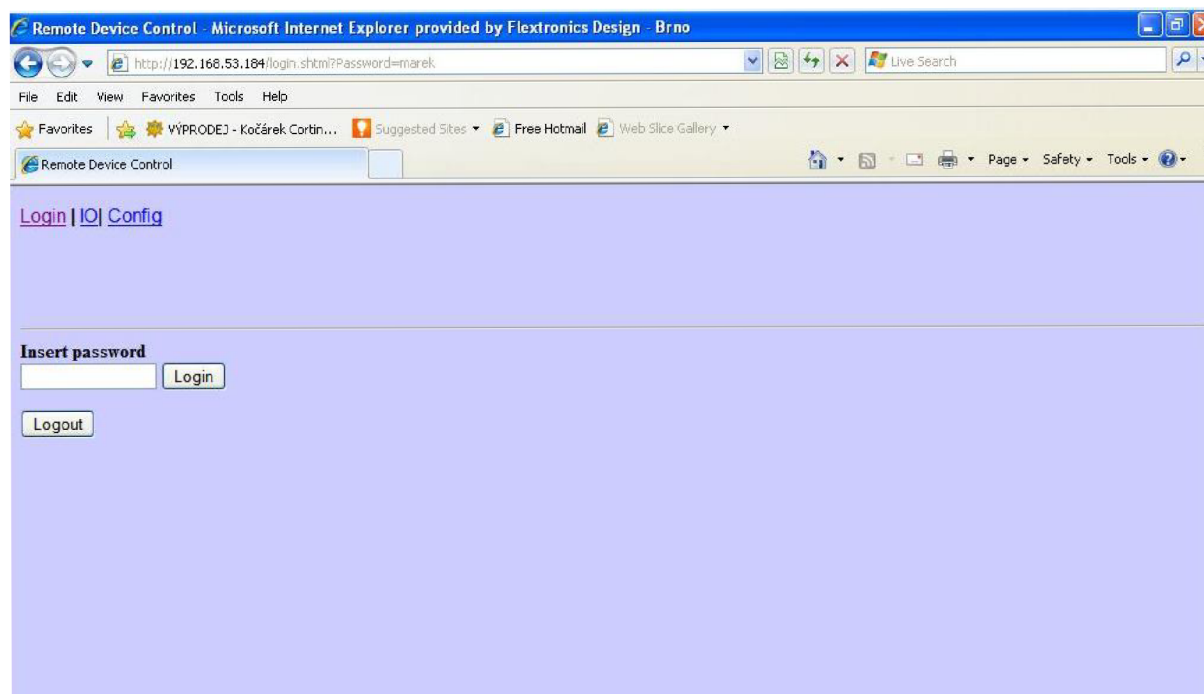
Obr. 17. Úvodní stránka login.shtml po zadání IP adresy zařízení

Poté uživatel zadá přihlašovací heslo a stikne tlačítko Login viz Obr. 18



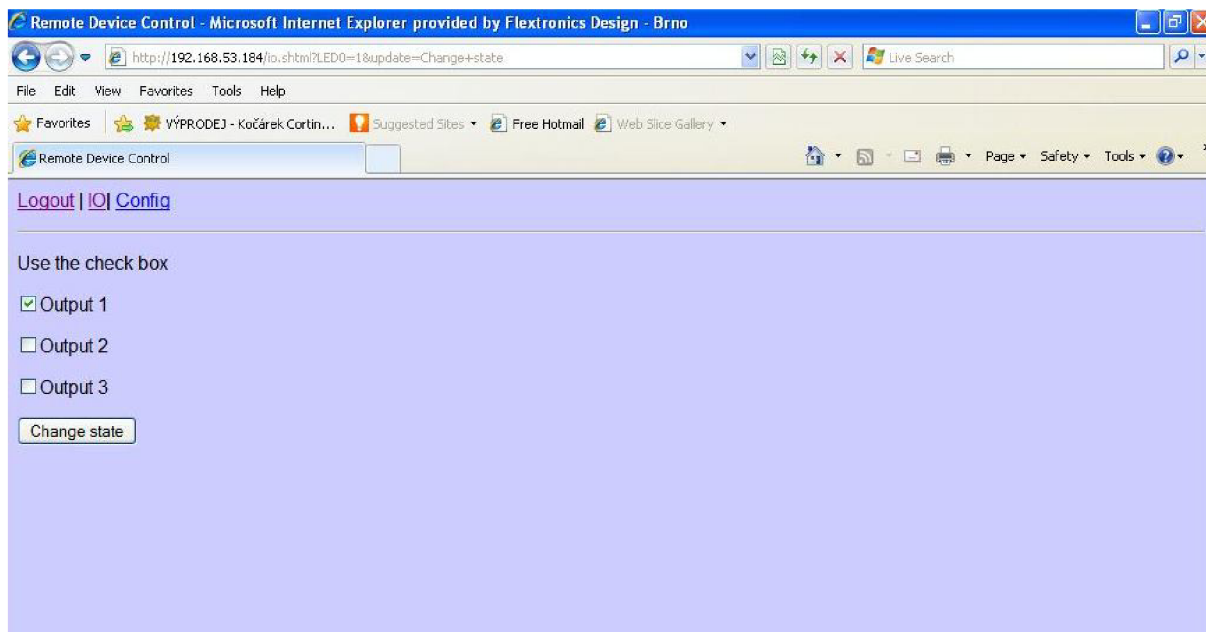
Obr. 18. Zadání přihlašovacieho hesla

Po správném zadání hesla se objeví záložky pro ovládání a nastavování zařízení viz Obr. 19



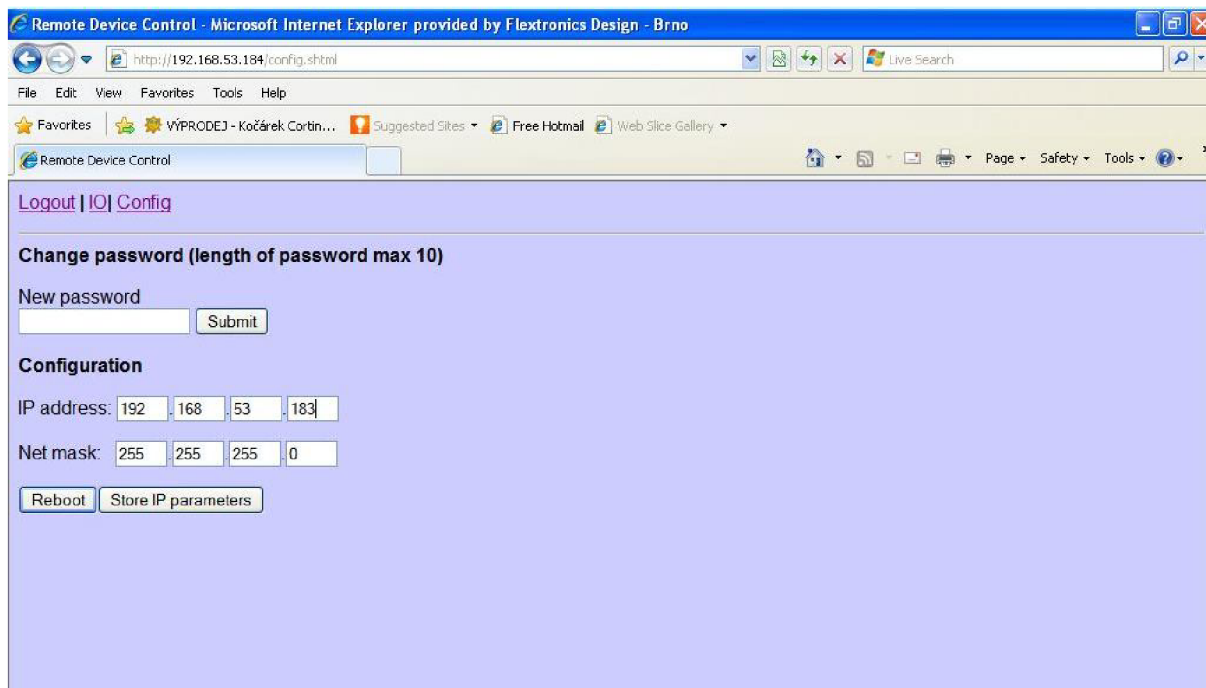
Obr. 19. Uživatelské záložky

V záložce IO uživatel nastavuje stavy jednotlivých výstupů a zároveň vidí, který výstup je zrovna zapnutý. Změnu stavu potvrdí kliknutím na tlačítko Change state viz Obr. 20



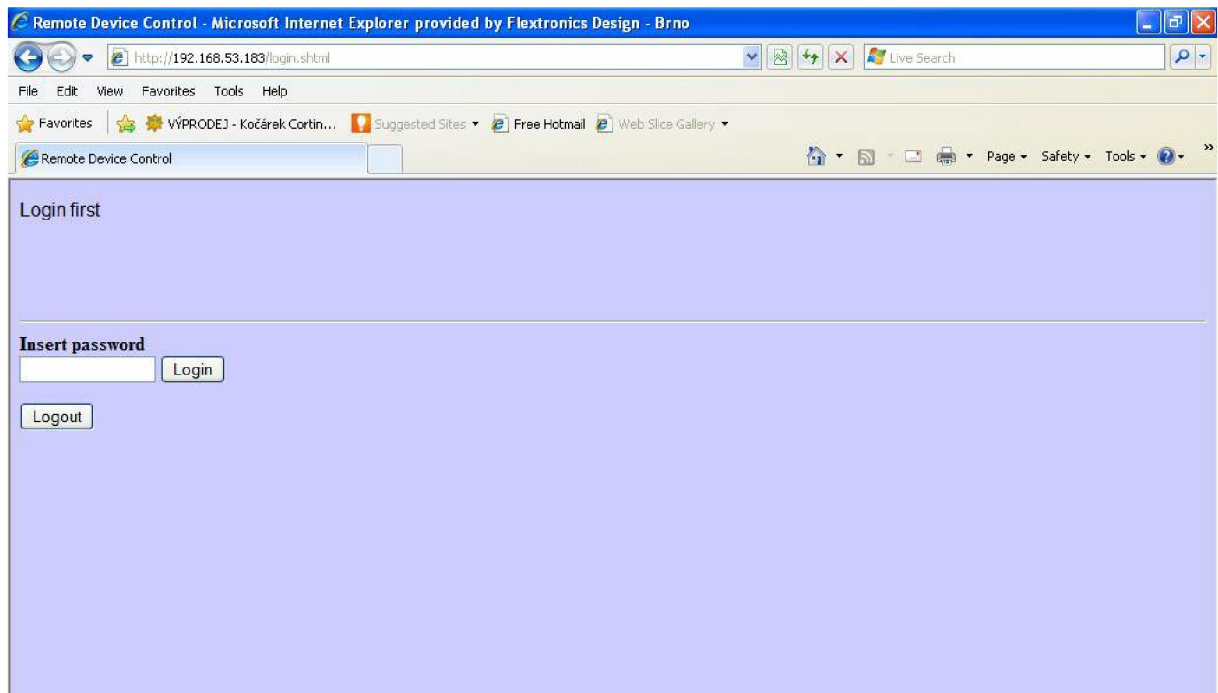
Obr. 20. Záložka IO

V záložce Config uživatel může změnit heslo a potvrdit tlačítkem Submit, dále nastavit IP adresu a masku. Toto se potvrdí tlačítkem Store IP Parameters. Viz Obr. 21



Obr. 21. Změna IP adresy

Po stisknutí tlačítka Reboot dojde k restartování zařízení. Po zapsání nové IP adresy do webového prohlížeče se objeví přihlašovací stránka viz Obr. 22.



Obr. 22. Přihlašovací stránka s novou IP adresou