

Bakalářská práce

2010

František Kroupa

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra fyziky

Bakalářská práce

Tester periferních zařízení s mikropočítačem
ATMEL AVR

Vedoucí práce: ing. Michal Šerý

Autor práce: František Kroupa

Studijní obor: MVT-K

Ročník: 3

Anotace (česky)

V této bakalářské práci je popsána realizace testeru rozhraní I2C, SPI a 1WIRE. V teoretické části je uveden jak popis těchto rozhraní, tak protokoly. V praktické části je pak uveden návrh zařízení a popis jeho funkce. Zařízení se skládá ze dvou hlavních částí, a to „master zařízení“ (PC vizualizace ovládání testeru) a „slave zařízení“ (mikrořadič ATmega-8 a FTDI 232). Funkčnost tohoto zařízení byla prakticky ověřena.

Abstract

The objective of this Bachelor's thesis was to design a tester of I2C, SPI and 1WIRE interfaces. In the theoretical part of the thesis, these interfaces as well as protocols are described. In the practical part, a design of the device and description of its functions are presented. The device is composed of two main components, a “master” (PC visualization of tester controlling) and “slave” (microcontroller ATmega-8 and FTDI 232). Function of the device was verified in practice.

Prohlašuji, že jsem předloženou práci vypracoval samostatně, pouze s použitím uvedené citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění, souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

27. prosince 2010

Touto formou děkuji svému konzultantovi Ing. Michalu Šerému za cenné rady a připomínky při zpracování mé práce.

OBSAH

1.	ÚVOD	6
2.	PŘÍSTROJOVÉ SBĚRNICE	6
2.1	Sběrnice I2C	7
2.1.1	HW zapojení I2C	7
2.1.2	Popis protokolu I2C	7
2.2	Sběrnice SPI/MicroWire	11
2.2.1	HW zapojení SPI/MicroWire	11
2.2.2	Popis protokolu SPI	11
2.2.3	Popis protokolu MicroWire	12
2.3	Sběrnice 1WIRE	14
2.3.1	HW zapojení 1WIRE	14
2.3.2	Popis protokolu 1WIRE	14
3.	MIKROKONTROLÉRY AVR	17
3.1	Architektura AVR	17
3.2	Mikrokontrolér ATmega8	20
3.3	Sériová komunikace UART	21
4.	NÁVRH ŘEŠENÍ	23
4.1	Popis zařízení TESTER	23
4.2	Popis programu	26
4.2.1	Zařízení slave	26
4.2.2	Zařízení master	35
5.	ZÁVĚR	44
6.	SEZNAM LITERATURY	45
7.	PŘÍLOHY (na CD)	46
7.1	Zdrojový kód aplikace master a slave	46
7.2	Schéma a Layout DPS	46

1. ÚVOD

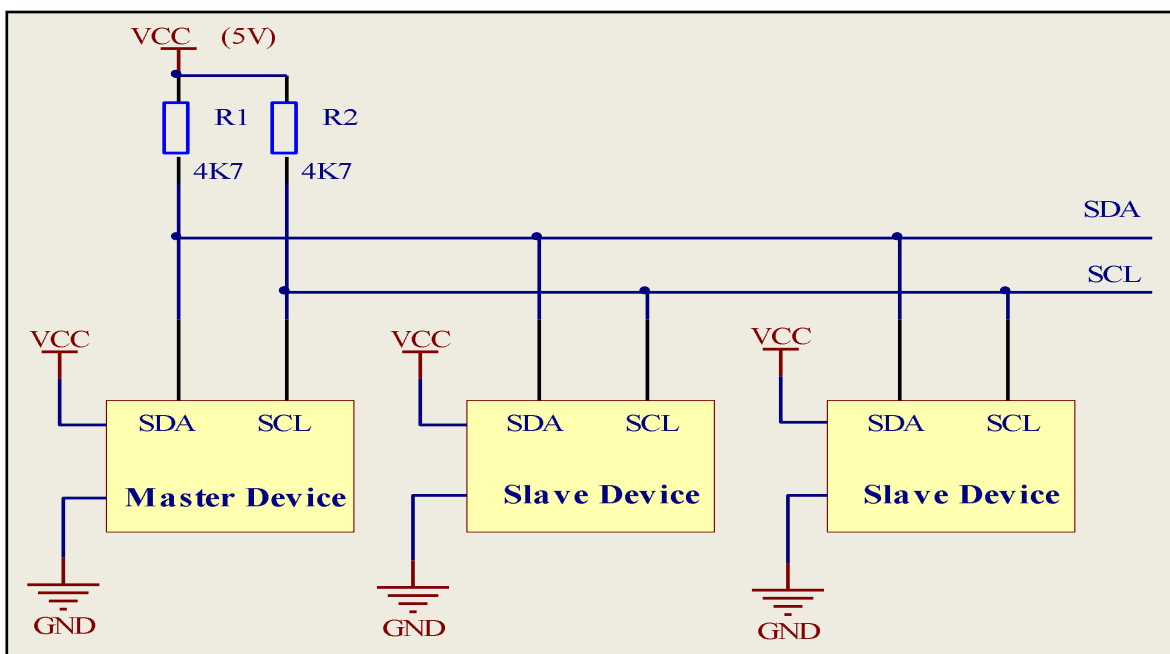
Cílem předkládané práce bylo sestrojít zařízení, které by umožňovalo otestovat periferní obvody, které disponují rozhraními (sběrnicemi) I2C, SPI a 1WIRE. Mohlo by také sloužit jako učební pomůcka, která by umožňovala zájemcům, aby si osvojili práci s výše uvedenými rozhraními. Zařízení se skládá ze dvou hlavních částí, a to „master zařízení“ (PC vizualizace ovládání testeru) a „slave zařízení“ (mikrořadič ATmega-8 a FTDI 232).

2. PŘÍSTROJOVÉ SBĚRNICE

Přístrojové sběrnice jako 1WIRE, SPI a I2C jsou v dnešní době nejčastějšími rozhraními, které se používají ke komunikaci mezi integrovanými obvody, případně ke komunikaci mezi mikropočítačovými moduly, na krátkou vzdálenost nepřesahující několik metrů. Hlavním důvodem použití je zmenšení počtu vývodů pouzder integrovaných obvodů pro samotnou komunikaci a díky jednoduchosti těchto protokolů možnost programové realizace na vývodech vhodného portu jakéhokoliv mikrokontroléru.

2.1 Sběrnice I2C

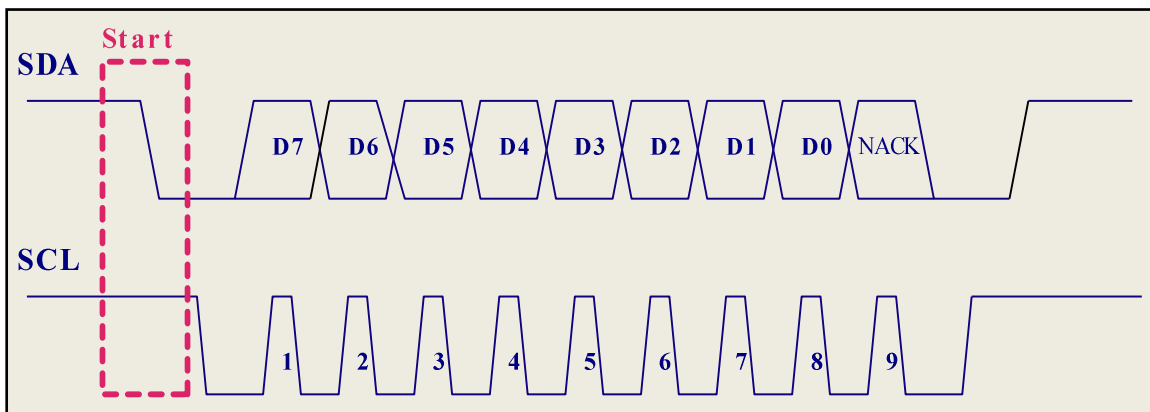
2.1.1 HW zapojení I2C



Obr. č. 1. Zapojení obvodů na sběrnici I2C.

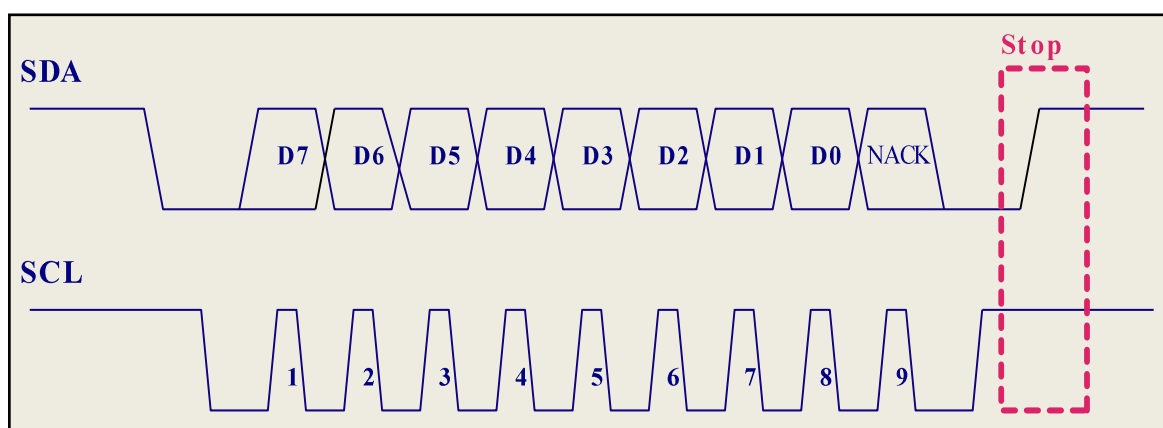
2.1.2 Popis protokolu I2C

Sběrnice I2C (Inter-Integrated-Circuit Bus) byla vyvinuta firmou Philips a dnes je součástí nejen spotřební elektroniky, pro kterou byla původně navržena, ale také průmyslové elektroniky. Řadiče této sběrnice naleznete nejen u procesorů firmy Philips (80C552,80C528), ale i u produktů jiných výrobců, např. Atmel (např. 24C64) nebo Motorola (např. HC05T7). Sběrnice I2C je tvořena dvojicí vodičů, na kterých je v klidovém stavu díky Pull-up rezistorům stav logické 1. Vodič označený jako SCL generuje hodinový synchronizační signál pouze na straně MASTERA a vodič označený jako SDA slouží k obousměrnému synchronizovanému přenosu dat. Jednotlivé fyzické vývody (SCL, SDA) těchto obvodů jsou vnitřně vybaveny budiči s otevřeným kolektorem, které umožňují připojení více obvodů na tuto sběrnici, aniž by došlo k hardwarovým kolizím (HW). Dále umožňují kterémukoliv obvodu kontrolovat bitovou synchronizaci a přizpůsobit tak rychlost přenosu dat. Přenos na sběrnici začíná, když MASTER „stáhne“ SCL z logické úrovně 1 na 0 při log. úrovni SDA = 1 [1], viz obrázek č. 2.



Obr. č. 2. Start sekvence na sběrnici I2C

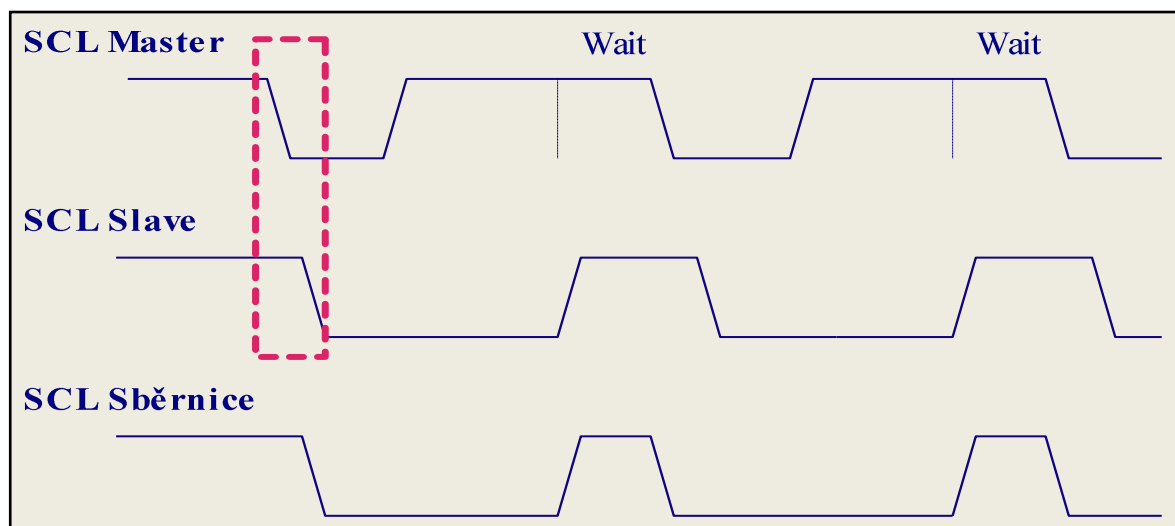
MASTEREM je postupně vysílána osmibitová hodnota začínající bitem s největší vahou. Každý přenesený bit musí zachovat neměnnou hodnotu po dobu trvání log. úrovně = 1 na SCL a pak je vzorkován při sestupné hraně. Vyslaná osmibitová hodnota je potvrzena přijímačem tak, že se vývod SDA přepne na logickou úroveň 0 v době devátého hodinového impulsu (platí pro první byte). Tento stav se značí jako signál ACK. Přenos je ukončen vzestupnou hranou na vodiči SDA při log. úrovni SCL = 1 [1], viz obrázek č. 3.



Obr. č. 3. Stop sekvence na sběrnici I2C

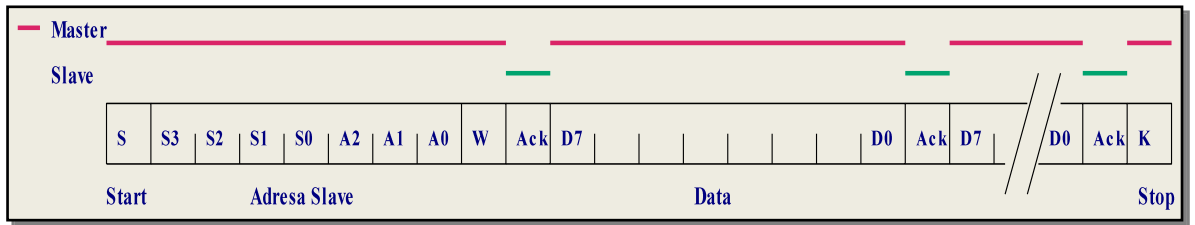
Každý obvod si kontroluje dobu svého signálu SCL v log. úrovni 0, aby byla zajištěna synchronizace přenosu u pomalejších obvodů připojených ke sběrnici I2C. To znamená, že obvod nastaví svůj SCL do log. úrovně 1, je-li připraven k příjmu dalšího bitu. V opačném případě jej podrží v log. úrovni 0. K synchronizaci obvodu dochází po startu přechodem SCL ze stavu 1 na 0. Pokud součástka s nejdelší dobou odezvy nebude připravena, bude SCL ve stavu 0. Součástky s kratší dobou odezvy svůj signál SCL nastaví do log. úrovně 1 a přejdou do stavu čekání. Tím dochází

k synchronizaci signálu SCL, jehož doba trvání log. úrovně 0 je dána součástkou s nejdelší odezvou. Po přechodu signálu SCL ze stavu log. úrovně 0 do 1 může být zahájen přenos dalšího bitu [1], viz obr. č. 4.



Obr. č. 4. Synchronizace na sběrnici I2C

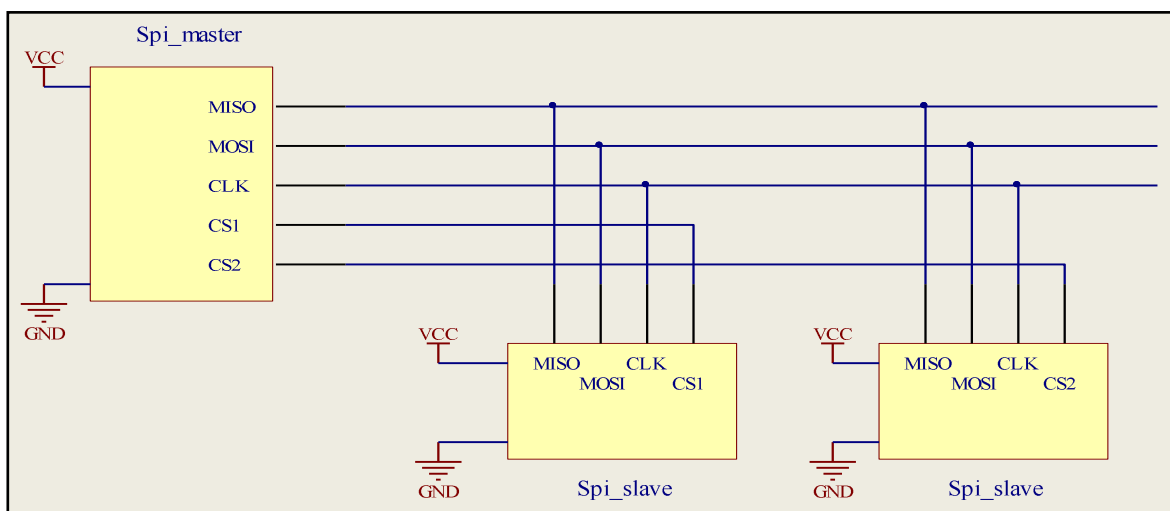
Sběrnice I2C se dá rozdělit do třech datových oblastí podle typu přenášených dat. První datová oblast se vysílá vždy po zahájení sekvence Start na sběrnici. Tato oblast má zpravidla délku 8 bitů, kde prvních 7 bitů znamená adresu zařízení na sběrnici a osmý bit znamená buď čtení (= 0) nebo zápis (= 1). Prvních 7 bitů se dále standardně rozděluje na 4 bity, které jsou dané a identifikují konkrétní obvod (S3-S0) a zbývající 3 bity, pomocí kterých lze, pokud to konkrétní obvody umožňují, nastavit jedinečnou adresu obvodového zapojení a tím dosáhnout připojení identických obvodů na téže sběrnici (A2-A0). Na sběrnici lze proto připojit maximálně 8 identických obvodů. Tato oblast dat je zakončena 1 bitovým signálem ACK ze strany adresovaného přijímače. Druhá datová oblast má zpravidla délku 8/16 bitů a znamená adresaci vnitřního paměťového místa nebo registru přijímače, který má být čten nebo zapisován, viz první datová oblast. Třetí datová oblast má zpravidla délku 8 a jen výjimečně 16 bitů a reprezentuje samotná čtená nebo zapisovaná data [1]. Na obrázku č. 5 je znázorněna bitová sekvence komunikace Master-Slave.



Obr. č. 5. Komunikace Master-Slave

2.2 Sběrnice SPI

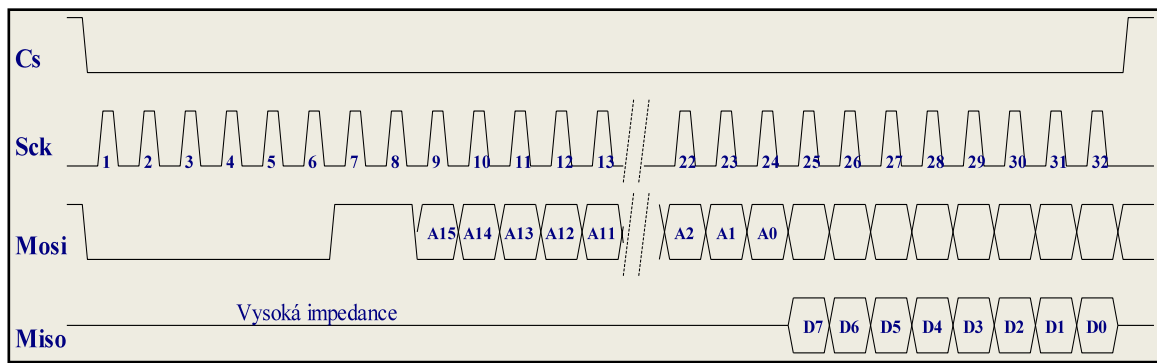
2.2.1 HW zapojení SPI/MicroWire



Obr. č. 6. Zapojení obvodů na sběrnici SPI

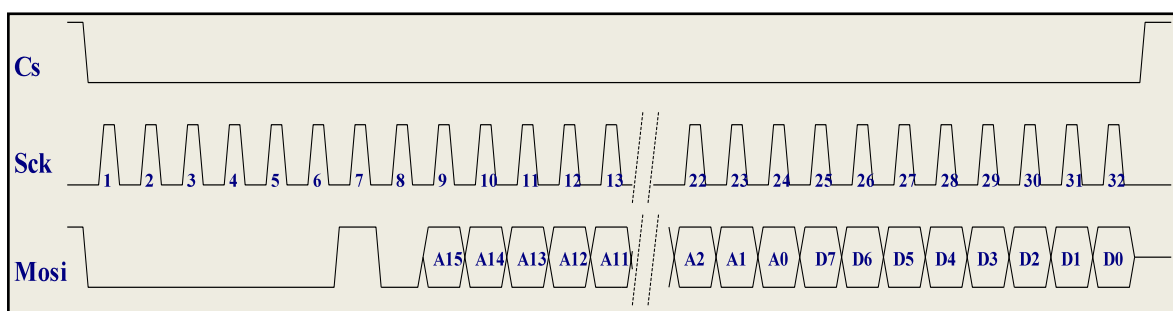
2.2.2 Popis protokolu SPI

Sběrnice SPI (Serial Peripheral Interface), byla navržena firmou Motorola, je v současné době asi nejperspektivnější sběrnici pro připojení periferních obvodů k mikropočítači. Řada výrobců procesorů nabízí obvody s tímto rozhraním (Atmel 89C52, ATmega xx, Analog Devices ADUC812, ADUC824, Microchip PICxxxx). Zmiňované obvody mají integrované řadiče rozhraní SPI na čipu procesoru, což umožňuje realizovat snazší a hlavně mnohem rychlejší komunikaci s periferním obvodem. Pokud mikroprocesor není touto sběrnici vybaven, musíme její funkci realizovat pomocí vstupně výstupních vývodů a vhodného programu. Přenos dat po sběrnici SPI je povinně obousměrný a sběrnici tvoří tři vodiče, viz obr. č. 6. Synchronizaci zajišťuje signál SCK generovaný nadřízenou jednotkou (mikrokontrolérem). Data jsou přenášena signály MISO (Master In Slave Out - přenos z periférie do mikroprocesoru) a MOSI (Master Out Slave In - přenos z mikroprocesoru do periférie) mezi osmibitovými posuvnými registry řadiče a podřízeného obvodu. Podřízený obvod, se kterým má probíhat komunikace, je aktivován signálem, obvykle označeným SS (Slave Select) nebo CS (Chip Select), viz obr. č. 7 [1].



Obr. č. 7. Operace čtení na sběrnici SPI

Napětíové úrovně jednotlivých signálů rozhraní SPI jsou dané použitou technologií. Obvody se standardně vyrábějí v provedení napájení 5 a 3,3 V. Maximální hodinová frekvence signálu se pohybuje až do 2 MHz. Pro snadné připojení obvodů, které nejsou pro sběrnici SPI standardní, je možné programovat polaritu hodinového signálu a okamžik vzorkování datových signálů pomocí bitů CPOL a CPHA. Většina radičů sběrnice SPI je navíc vybavena možností změny synchronizačního hodinového signálu pro připojení obvodů s různou rychlostí. Dokončení přenosu řízeného radičem SPI je signalizováno příznakovým bitem, pomocí kterého je možno vyvolat přerušení v mikroprocesoru. Délka přenášeného slova na sběrnici SPI je pevná, osmibitová. Komunikace na sběrnici se zpravidla zahajuje vysláním tzv. osmibitové instrukce, která určuje, jestli se bude jednat o zápis nebo čtení z periferních obvodů a jakou posloupnost dat může periférie očekávat. Na obr. 8 je zobrazen průběh signálů na sběrnici [1].

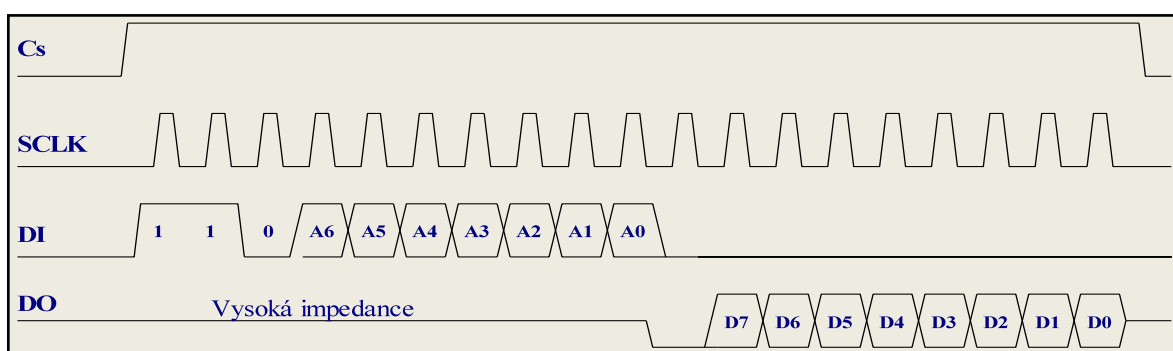


Obr. č. 8. Operace zápis na sběrnici SPI

2.2.3 Popis protokolu MicroWire

Obvodovou sběrnici, jako je SPI, zavedla jako svůj standard firma National Semiconductor pod označením MicroWire. Obvodově je sběrnice tvořena třemi vodiči

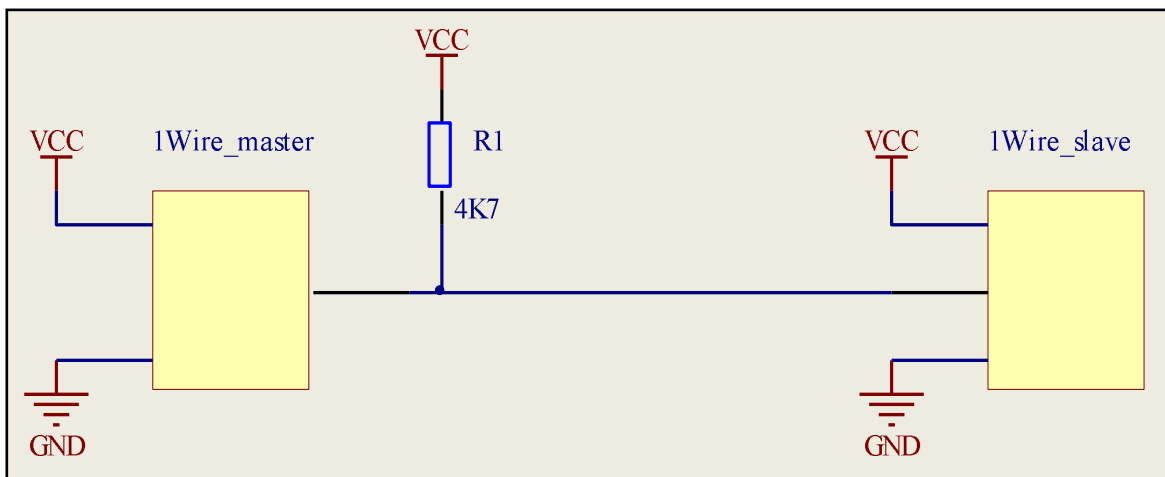
jako SPI a to SCLK, SO/DI a DO/SI, ale její použití je univerzálnější. Náběžnou hranou hodinového signálu SCLK se řídí přenos po zbývajících dvou datových vodičích. První propojuje výstup řadiče SO (Seriál Out) se vstupy DI (Data In) periferních obvodů, druhý připojuje výstupy DO (Data Out) periferních obvodů na vstup SI (Serial In) řadiče. Délka přenášeného slova na sběrnici MicroWire není pevně definována, ale obvykle bývá násobkem čtyřbitové slabiky. Přenos může být pouze jednosměrný. Protože je komunikace na této sběrnici pro jednotlivé periferní obvody různá, nebývá řadič této sběrnice integrován přímo do mikroprocesorů a emuluje se programově na zvolených vývodech. Přenos po sběrnici začíná přenosem log. úrovně 1 do periferního obvodu následovaným určitým počtem bitů s jednoduchým definovaným protokolem. Po přijetí posledního bitu posloupnosti je informace zapsána do periferního obvodu. Tento obvod, pokud se nejedná o jednosměrnou komunikaci, vyšle data do mikrokontroléru v závislosti na zapsaném příkazu. Na obr. č. 9 je zobrazen průběh signálů na sběrnici [1].



Obr. č. 9. Operace čtení na sběrnici MicroWire

2.3 Sběrnice 1WIRE

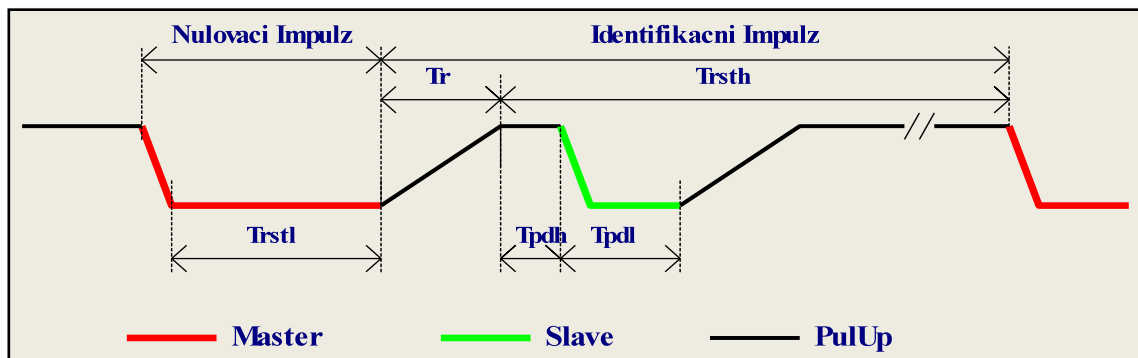
2.3.1 HW zapojení 1WIRE



Obr. č. 10. Zapojení obvodů na sběrnici 1Wire

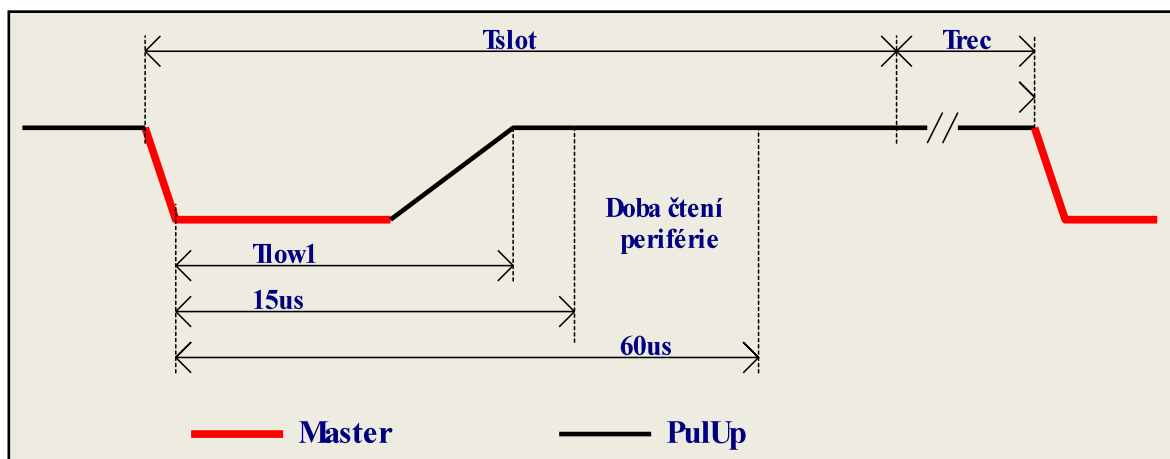
2.3.2 Popis protokolu 1WIRE

Sběrnice 1WIRE neboli MicroLAN byla navržena firmou Dallas původně pro její „dotykové paměti“ (Touch Memory). Zařízení s touto implementovanou sběrnicí obsahuje pouze dva vývody. Jeden vodič zajišťuje obousměrnou komunikaci včetně napájení a druhý je zem. Pro velký úspěch firma rozšířila řadu obvodů o teploměry, A/D převodníky, expandery sběrnic atd., které jsou vybaveny jednodrátovou sběrnicí, jak již název napovídá. K tomu, aby mohl řídicí obvod Master provádět na jednom vodiči obousměrnou komunikaci, je potřeba jeho výstup vybavit obvodem s otevřeným kolektorem. Pokud je spotřeba obvodů připojených na tuto sběrnici velmi malá, může být obvod přímo napájen z tohoto vodiče. V opačném případě jsou obvody vybaveny ještě třetím napájecím vodičem. Na sběrnici jsou definovány čtyři základní operace, a to a) nulování a identifikace obvodu, b) zápis log. úrovně 0, c) zápis log. úrovně 1 a d) čtení jednoho bitu. Na obr. č. 11 je zobrazen počáteční nulovací impuls pro obvody připojené na sběrnici a dále poloha identifikačního impulsu, kterým obvod oznamuje svojí přítomnost na sběrnici [2, 3].



Obr. č. 11. Nulovací a identifikační impuls

Proces nulování je definován jako osamocený impuls s log. úrovní 0 o minimální délce 480 μ s následovaný impulzem o log. úrovni 1 a stejné délce. Impuls s log. úrovní 1 je nutný nejen pro napájení připojených zařízení, ale i jako časový prostor k realizaci identifikačního impulsu zařízení. Ten začíná v rozmezí 15 až 60 μ s po log. úrovni 0 a jeho délka se pohybuje od 60 do 240 μ s. Řídicí obvod tak může zjistit, zda jsou připojeny na jeho sběrnici obvody s tímto rozhraním. Je-li na sběrnici připojeno více obvodů, pak řídicí obvod čte logický součin všech identifikačních impulsů od jednotlivých obvodů [2, 3]. Časové průběhy při zápisu bitu do obvodu po sběrnici jsou zobrazeny na obr. č. 12,13.



Obr. č. 12. Časový průběh pro zápis log. úrovně 1 do periférie.

3. MIKROKONTROLÉRY AVR

3.1 Architektura AVR

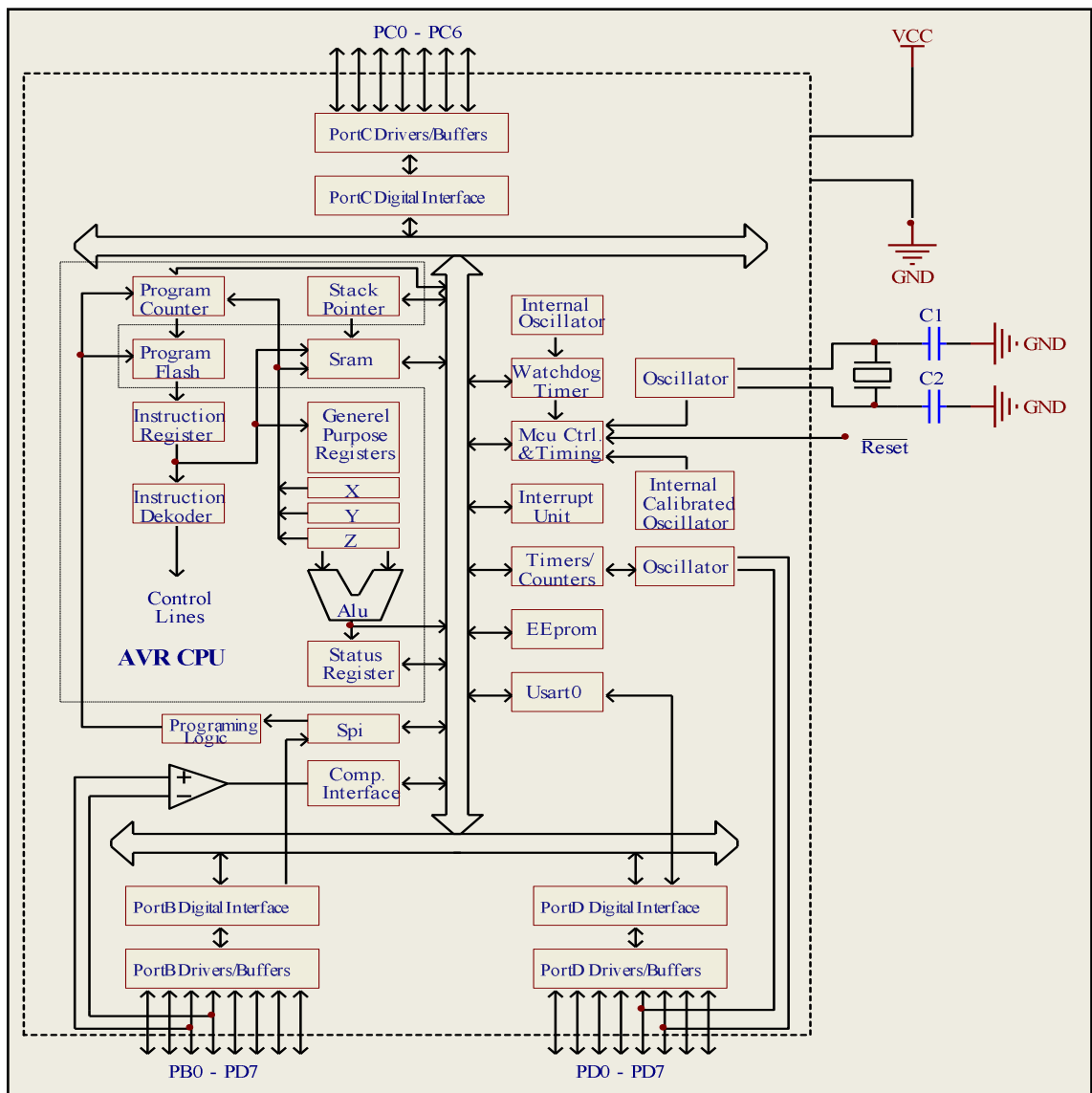
AVR procesory jsou RISC procesory postavené na Harvardské architektuře (mají oddělenou paměť pro program a data). Skládají se z třiceti dvou stejných 8bitových registrů, které mohou obsahovat jak data, tak i adresy. Jsou přímo propojeny s aritmeticko-logickou jednotkou, proto je ALU schopna provést jednu operaci v jednom hodinovém cyklu. Posledních 6 registrů je možno také po párech využít jako ukazatele pro nepřímé adresování do paměti [4].

Architektura AVR vychází z koncepce rychle přístupného registrového pole, které obsahuje 32 obecně použitelných registrů délky 8 bitů. Přístup do registrového pole je proveden v jediném strojovém cyklu. To znamená, že během jednoho strojového cyklu lze vykonat jednu aritmeticko-logickou operaci. Oba operandy aritmeticko-logické instrukce jsou načteny z registrového pole, operace je provedena a výsledek je uložen zpět do registrového pole. To vše v jediném strojovém cyklu [4].

Hlavní vlastností RISC procesorů jsou:

- redukovaná sada instrukcí
- obsahuje převážně jednoduché instrukce
- délka provádění jedné instrukce je vždy jeden cyklus
- délka (počet bitů) všech instrukcí je stejná
- mikroinstrukce jsou hardwarově implementovány na procesoru
- využívá se zde techniky řetězení instrukcí

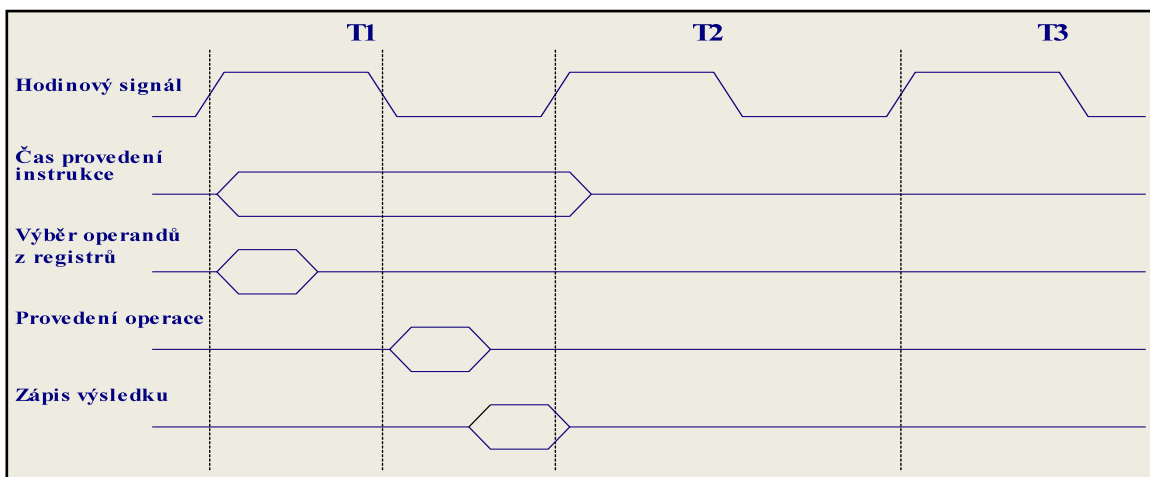
Většina instrukcí z instrukčního souboru má délku jednoho slova (16 bitů), pouze 4 instrukce jsou dvě dlouhá slova (32 bitů). To umožňuje rychlejší vykonávání instrukcí a také dovoluje jednodušší strukturu mikroprocesorového řadiče. Většina instrukcí umožňuje přímý jednocyklový přístup do všech registrů [4]. Na obrázku č. 14 je uvedeno zjednodušené blokové schéma mikrokontroléru AVR.



Obr. č. 14. Blokové schéma mikrokontroléru AVR

Strojový cyklus mikrokontrolérů AVR odpovídá hodinovému cyklu. Nedochozí k žádnému dělení hodinových cyklů jako u jiných mikrokontrolérů (např. řada 8051 potřebuje pro vykonání jednoho strojového cyklu 12 hodinových cyklů). Mikrokontroléry AVR používají velmi primitivní, ale účinný, předvýběr instrukce (prefetch). Tím je implementován jednofázový pipelining (zřetězené provádění instrukcí). Z programové paměti je nejdříve načtena první instrukce. Současně s tím, jak se v následujícím hodinovém cyklu tato instrukce provádí, se předvybírání druhá instrukce. Ve třetí fázi hodin se provádí druhá instrukce a zároveň se dopředu z instrukční fronty (pipeline) načítá třetí instrukce atd., viz obr. č. 15. Posledním prvkem, který výrazně zvyšuje výpočetní výkon mikrokontrolérů AVR, je jednotná délka instrukce. Všechny instrukce mají délku 16 bitů (výjimkou jsou instrukce LDS a STS, které mají délku 32 bitů). V těchto 16 bitech je uložen nejen operační znak

instrukce (určuje její typ), ale i informace o operandech. Tento prvek zjednodušuje realizaci dekodéru instrukcí a jednoduchost přináší zvýšení rychlosti [4].



Obr. č. 15. Zřetězené provádění instrukcí

Vyrábějí se 3 základní řady procesorů - AVR ATtiny, AVR ATmega, AVR ATXmega [4], viz obr. č. 16.

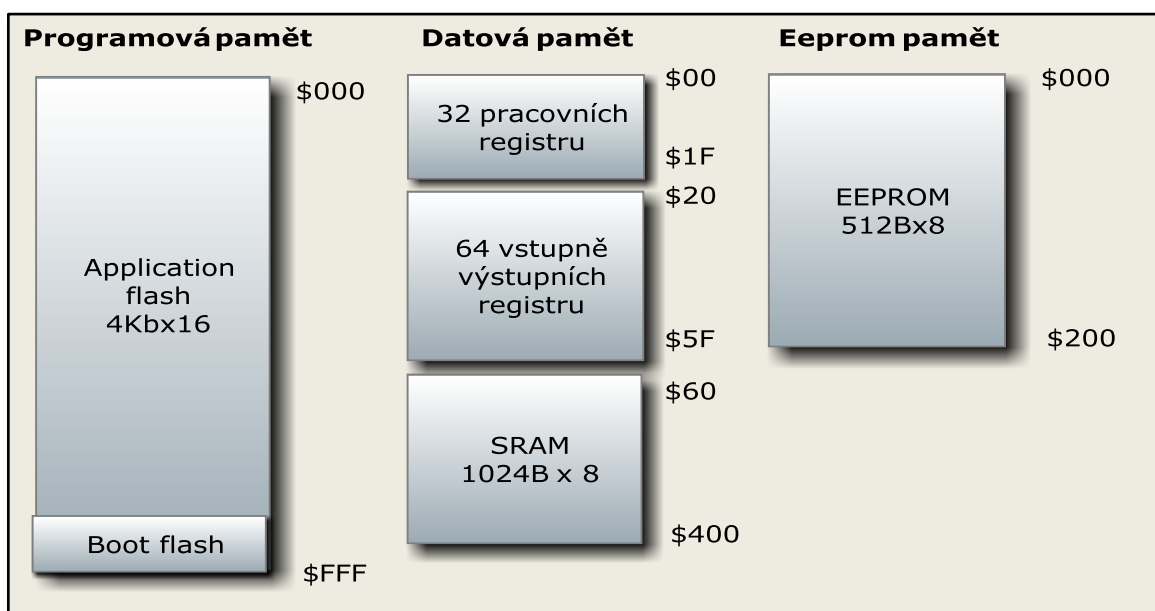
Parametry	AVR ATtiny	AVR ATmega	AVR ATXmega
Flash Pamět	1 - 8 KB	4 - 256 KB	16 - 256 KB
Sram Pamět	64 - 512 B	512 B - 8 KB	4 - 16 KB
Eeprom Pamět	64 - 512 B	265 B - 4 KB	1 - 4 KB

Obr. č. 16. Tabulka řady procesorů

3.2 Mikrokontrolér ATmega8

ATMEL ATMEGA 8 je nízko-příkonový 8bitový mikrokontrolér založený na AVR RISC architektuře. Tím, že provádí výkonné instrukce v jediném hodinovém cyklu, dosahuje tento mikrokontrolér výpočetního výkonu 1 MIPS na 1 MHz. Instrukční soubor tohoto mikrokontroléru obsahuje 130 instrukcí a má k dispozici 32 registrů délky 8 bitů. Paměť programu je tvořena zabudovanou FLASH, jejíž kapacita je 8 KB. FLASH je programována buď klasickým paralelním programátorem nebo přímo v systému pomocí rozhraní SPI (počet přeprogramování je cca 1000 cyklů). Dále je zde možnost nahrát program pomocí zavaděče takzvaného bootloaderu. Jedná se o aplikaci, která se mapuje obecně do horní části Flash paměti a umožňuje přepsání Flash pomocí implementovaných periférií jako je sériová linka UART, I2C nebo jakýmkoli jiným námi vytvořeným rozhraním. V případě ATmegy-8 je zavaděč možno umístit ve čtyřech modech velikosti. Funguje to následujícím způsobem. Po zapnutí nebo resetu zařízení se začnou jako první vykonávat instrukce, které jsou umístěné na adrese 0xC00, kde začíná kód zavaděče (bootloaderu) [4].

Datová paměť je typu RAM o velikosti 1KB. Dále je zde implementována paměť EEPROM o velikosti 512B, která slouží pro uchování získaných dat i po vypnutí zařízení. Na obrázku č. 17 je přehled organizování paměti v ATmega8 [4].



Obr. č. 17. Organizace paměti v ATmega8

Mikrokontrolér má zabudované dva 8bitové čítače/časovače a jeden 16bitový čítač/časovač. Tyto čítače lze nastavit do různých režimů (output compare, input

compare, pwm, prescaler). Dále je vybaven obvodem WDT (watchdog timer, hlídač správného běhu programu). Je zde k dispozici SPI kanál, který mimo jiné slouží k sériovému downloadu (programování FLASH nebo EEPROM). Nechybí zde periférie jako programovatelný USART, I2C, analogový komparátor a 10bitový A/D převodník. Zajímavostí je zabudovaný kalibrovatelný RC oscilátor jako zdroj hodinového kmitočtu pro samotný mikrokontrolér. Dostupné jsou čtyři základní frekvence (1, 2, 4, 8MHz). Volba tohoto zdroje hodinového signálu se provádí modifikací pojistky CKSEL [4].

3.3 Sériová komunikace UART

Mikrokontrolér ATmega8 obsahuje jeden plně duplexní asynchronní přijímač a vysílač (umožňuje současný příjem a vysílání dat). Přijímač se skládá z generátoru přenosových rychlostí s velkým počtem nastavení). Dále pak z 8 bitového buferu a posuvného registru. Řízení je prováděno pomocí 4 vstupně/výstupních registrů[4].

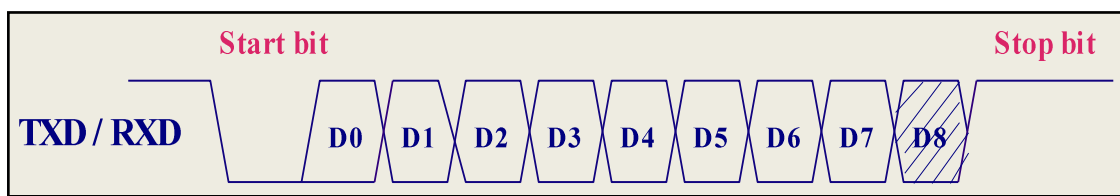
UDR – datový registr (obsahuje přijatou nebo vysílanou hodnotu)

USR – stavový registr (příznak stavu přenosu)

UCR – řídicí registr (povoluje R/W, obsahuje masky přerušení)

UBRR – registr přenosové rychlosti (volí přenosovou rychlost)

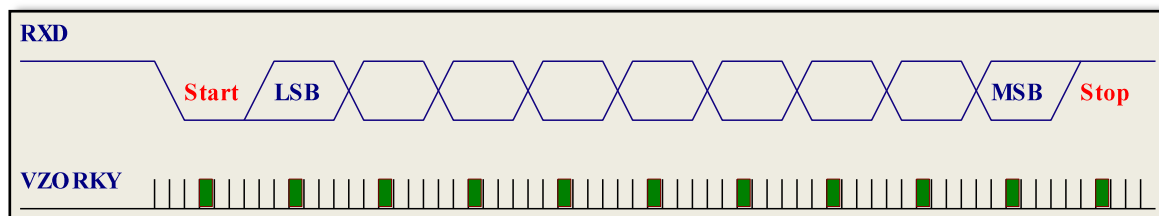
Standardní přenosový rámec se uskuteční v případě vysílání asi takto. Nejprve se na TXD vyšle start bit – log. úroveň 0. Pak se postupně vysílá 8 nebo 9 bitů podle nastavení v pořadí od bitu s nejnižší váhou. Přenos se ukončí vysláním stop bitu – log. úroveň 1, viz obr. č. 18.



Obr. č. 18. Sekvence bitů na sériové sběrnici UART

U příjmu přijímač vzorkuje signál na vývodu RXD kmitočtem 16 x vyšším než je přenosová rychlost (na jeden bit připadá 16 vzorků). Linka je nečinná v log. 1, každý následující vzorek v log.0 je interpretován jako začátek start bitu, a to jen v případě, že vzorky 8, 9, 10 nejsou v log. úrovni 1. Je-li detekován platný start bit, provádí se vzorkování datových bitů následujících za start bitem. Důležité jsou opět

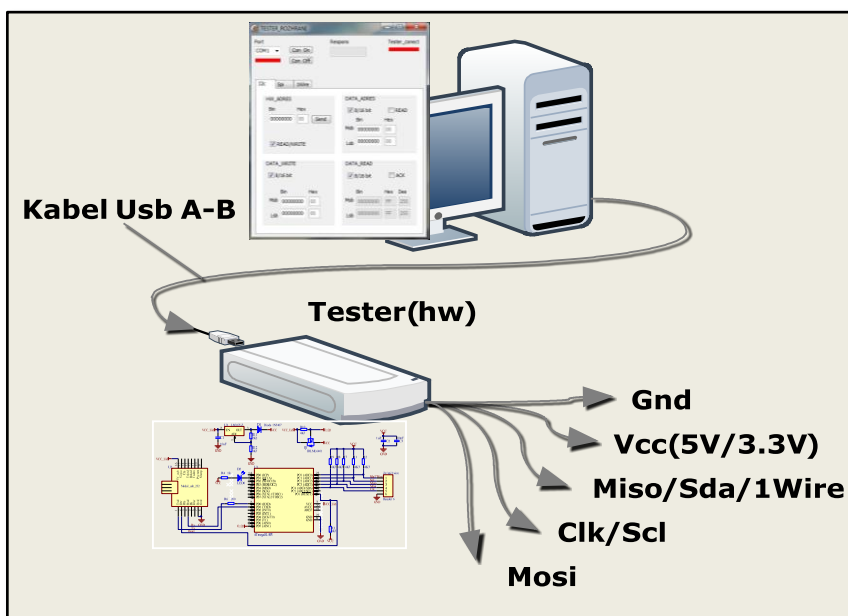
vzorky 8, 9, 10. Logická hodnota zjištěná minimálně ve dvou vzorcích z této trojice je brána jako hodnota bitu, viz obr. č. 19 [4].



Obr. č. 19. Vzorkování signálu na vývodu RXD

4. NÁVRH ŘEŠENÍ

4.1 Popis zařízení TESTER

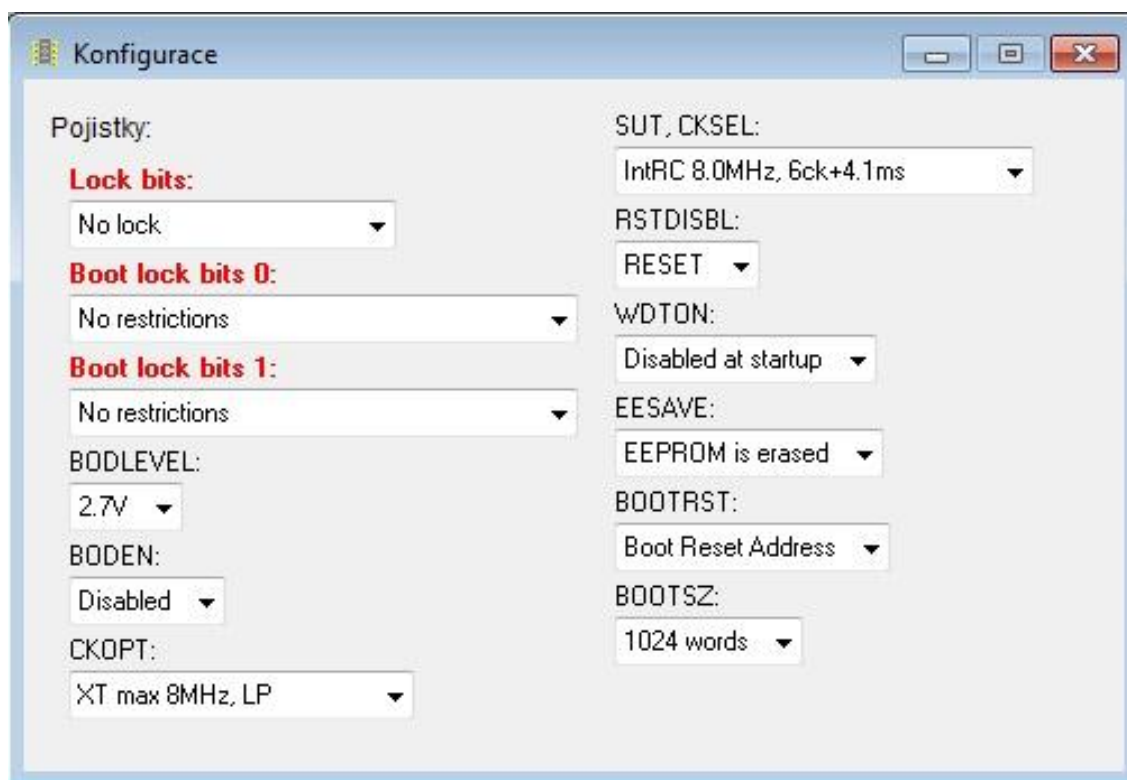


Obr. č. 20. Blokové schéma zařízení TESTER

Zařízení, které jsem nazval TESTER, je rozděleno na dva různé hardwarové celky, které mezi sebou komunikují způsobem Master – Slave. Zařízení Master je PC, kde běží GUI aplikace napsaná v jazyce vizual Pascal Delphi. Tato aplikace komunikuje (dotazuje) se zařízením Slave pomocí virtualizovaného sériového portu nad rozhraním USB, které je dnes již rozšířeným standardem. Blokové schéma je na obrázku č. 20.

Zařízení Slave vykonává nejen funkci převodníku úrovní pro námi zvolené rozhraní, ale také implementuje kompletní protokoly, které již byly zmíněny. Jádrem zařízení je mikrokontrolér ATmega-8, kde běží aplikace napsaná v prostředí BASCOM-AVR z pohledu jiných programovacích softwarů se jedná o vysokoúrovňovou implementaci jazyku Basic. Z důvodu minimalizace externích součástek jsem zvolil jako zdroj hodinového signálu pro mikrořadič interní kalibrovatelný RC oscilátor. Zkompilovaná aplikace se nahrává bootloaderem do Flash eeprom pomocí vizualizované sériové linky s nastaveným bitrate 34600kb/s. Tento kód programu je umístěn v horních 2048 bytech flash. Výhodou je, že je součástí vývojového prostředí BASCOM-AVR, takže odpadá jakákoliv manipulace s externím programátorem v případě dalšího ladění samotné aplikace. Tato základní nastavení, tj. volba zdroje hod. signálu, boot reset address atd., se provedou pomocí

tzv. konfiguračních bitů (pojistik), a to pouze jednou, jakýmkoliv externím programátorem. V tomto případě jsem použil sériový SPI programátor od firmy Asix Presto pro nahrání aplikace bootloaderu. Screenshot nastavení pojistik je na obrázku č. 21.

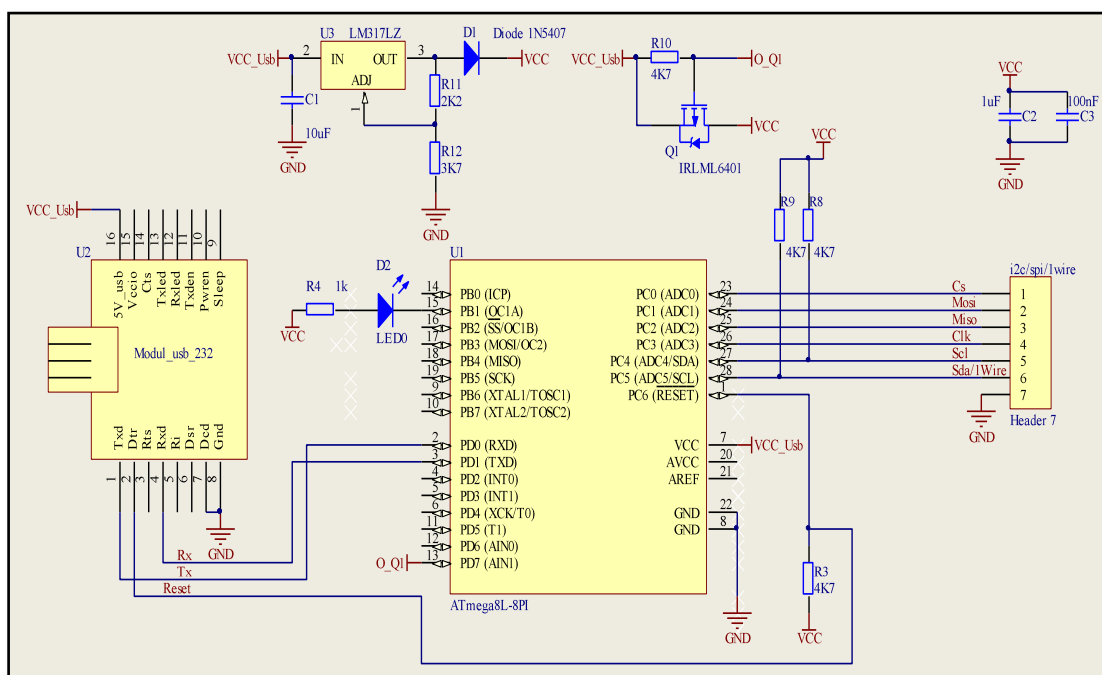


Obr. č. 21. Nastavení pojistik

Zařízení Slave dále obsahuje protokolový převodník USB/RS232-TTL. Jedná se již o hotový modul od firmy PEKDesign v provedení SODIL 16, postaveném na obvodu FT232RL, který obsahuje kromě signálových vývodů RX TX, které zajišťují samotnou komunikaci, ještě další signálový vývod DTR, který využívá GUI aplikace bootloaderu pro reset a následný download samotné funkční aplikace. Předpokládá se, že testované obvody budou umístěny z důvodu snazší manipulace na nepájivém poli a propojí se lankovými vodiči vždy zakončenými KONPC-SPK-1 konektory, které jsou součástí TESTERU. Napájení zařízení Tester Slave bude napájen přímo z USB portu (5V), jelikož se nepředpokládá (i s testovanou součástí) odběr větší než 100mA. Tato hodnota je asi pětikrát nižší než maximální hodnota proudu, kterou je podle normy možné odebírat z portu USB.

Poslední součástí, která stojí za pozornost, je lineární stabilizátor LM317 v zapojení s výstupním napětí 3.5V a P-Mosfet tranzistor, pomocí kterého lze přepínat

napájení (5V/3,3V) testovaného obvodu. Napěťová rezerva 200mV je volena záměrně z důvodu úbytku napětí na diodě D1. Schéma zařízení Slave je na obrázku č. 22.



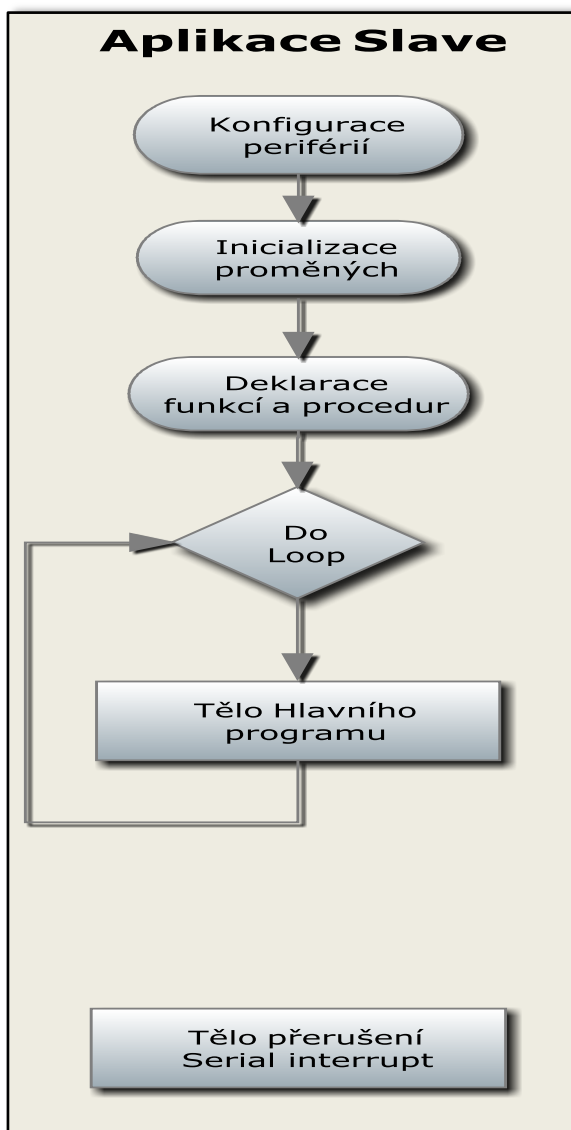
Obr. č. 22. Schéma zařízení Slave

Zařízení master je, jak již bylo zmíněno, PC, kde běží GUI aplikace napsaná v jazyce Delphi. Základem aplikace je komponenta TForm, na které jsou umístěny ovládací prvky jako Combobox pro výběr komunikačního portu, výstupního napětí, tlačítka zahajující komunikaci, panelboxy signalizující stavy komunikace a hlavně tři TabSheety s pojmenováním komunikačních protokolů, které obsahují další EditBoxy pro setování adres. Dále pak CheckBoxy pro nastavení velikosti datových slov a tlačítka řídicí čtení nebo zápis do příslušného testovaného obvodu. Podrobnější popis a funkce těchto komponent se nachází v kapitole 4.2 Popis programu (4.2.1 Zařízení master).

4.2 Popis programu

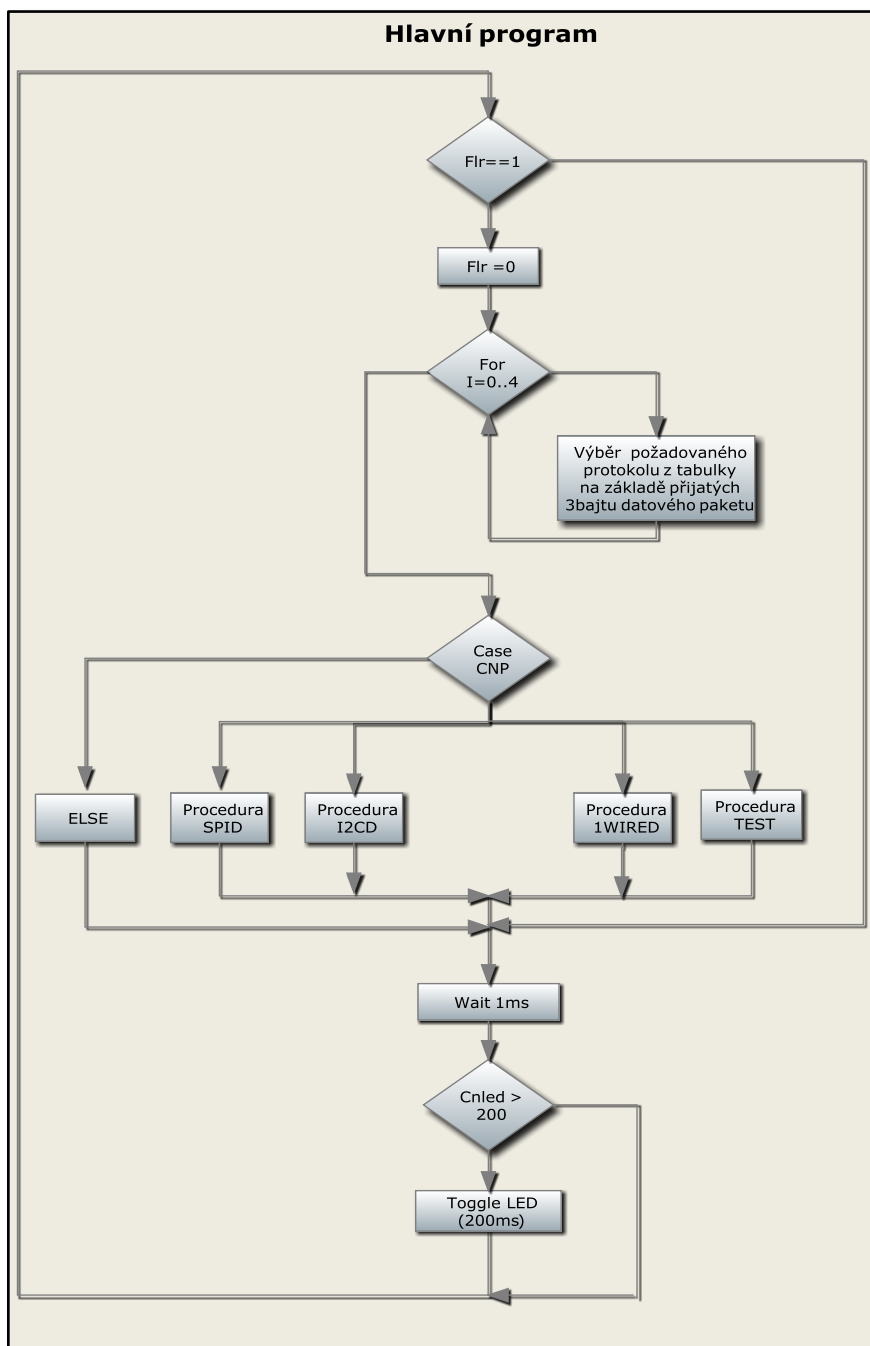
4.2.1 Zařízení slave

Jak již bylo řečeno, aplikace je napsána ve vývojovém prostředí BascomAVR. Jedná se o jakýsi derivát jazyku Basic pro osmibitové mikrořadiče ATmega firmy Atmel. Toto IDE poskytuje velice komfortní prostředí a nepřehledné množství již hotových procedur/funkcí pro práci s různými periferiemi, které jsou již zkompileované v tomto prostředí. Aplikace zařízení slave je vyobrazena na základním vývojovém diagramu na obrázku č. 23.



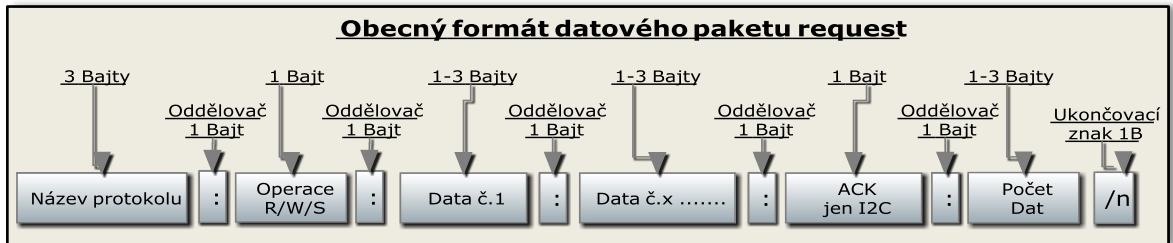
Obr. č. 23. Základní vývojový diagram

První položkou na vývojovém diagramu je ‘Konfigurace periférií’. Jedná se o část programu, kde se nastavují hodnoty registrů pro práci se sériovým kanálem, RC oscilátoru a pinů, na kterých budou vykonávány komunikace námi zvolených rozhraní. Položku ‘Inicializace proměnných’ a ‘Deklarace funkcí a procedur’ není jistě třeba více komentovat. V přiloženém zdrojovém programu jsou tyto položky podrobně okomentované. Další položkou je tělo hlavního programu, které je umístěno v nekonečné smyčce a bližší popis je uveden na následujícím vývojovém diagramu. Dále se zde nachází „Tělo přerušení (Serial interrupt)“, které je také popsáno ve zvláštním vývojovém diagramu, viz níže.



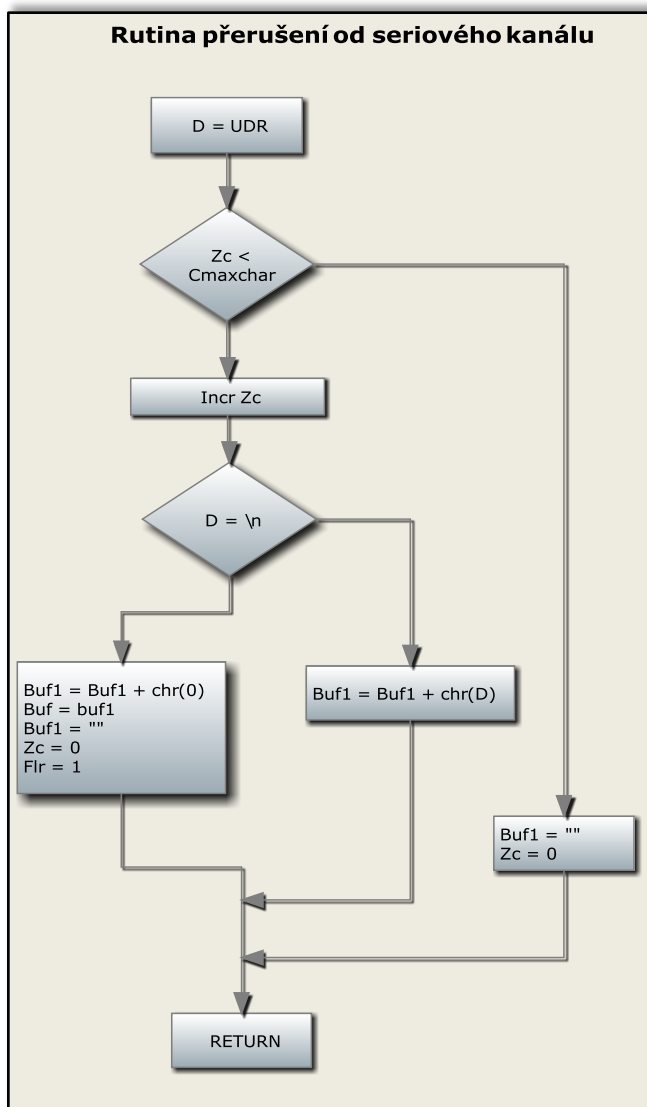
Obr. č. 24. Vývojový digram Hlavní program

Na vývojovém diagramu Hlavního programu obr. č. 24 je vidět, že se každou milisekundu provádí test příznakového bytu Flr, který nastavuje procedura přerušení od sériového kanálu. Tento byte signalizuje přijmutí datového paketu. Datový paket je uložen v proměnné Buf, viz zdrojový kód. Komunikace na sériové lince probíhá ve znakovém režimu z důvodu jednodušší manipulovatelnosti s příchozími daty. Obecná struktura datového paketu ze strany Master-Slave je vidět na obrázku č. 25.



Obr. č. 25. Obecný datový paket Master-Slave

Na základě přijatých třech bytů se proměnné I přiřadí příslušná hodnota s tabulky hodnot, která v přepínači Case-select vybere příslušnou proceduru k vykonání. Dále program pokračuje a vykonává test na čítači cyklů, který v pravidelných intervalech 200 ms zapíná/vypíná led diodu. Tato dioda signalizuje opakované provádění těla hlavního programu. Rutina přerušení od sériového kanálu se stará o zpracování asynchronních dotazů ze strany Master. V okamžiku příjmu znaku na sériové lince se vykoná přerušení a začne se vykonávat rutina zobrazená na vývojovém diagramu obrázku č. 26.

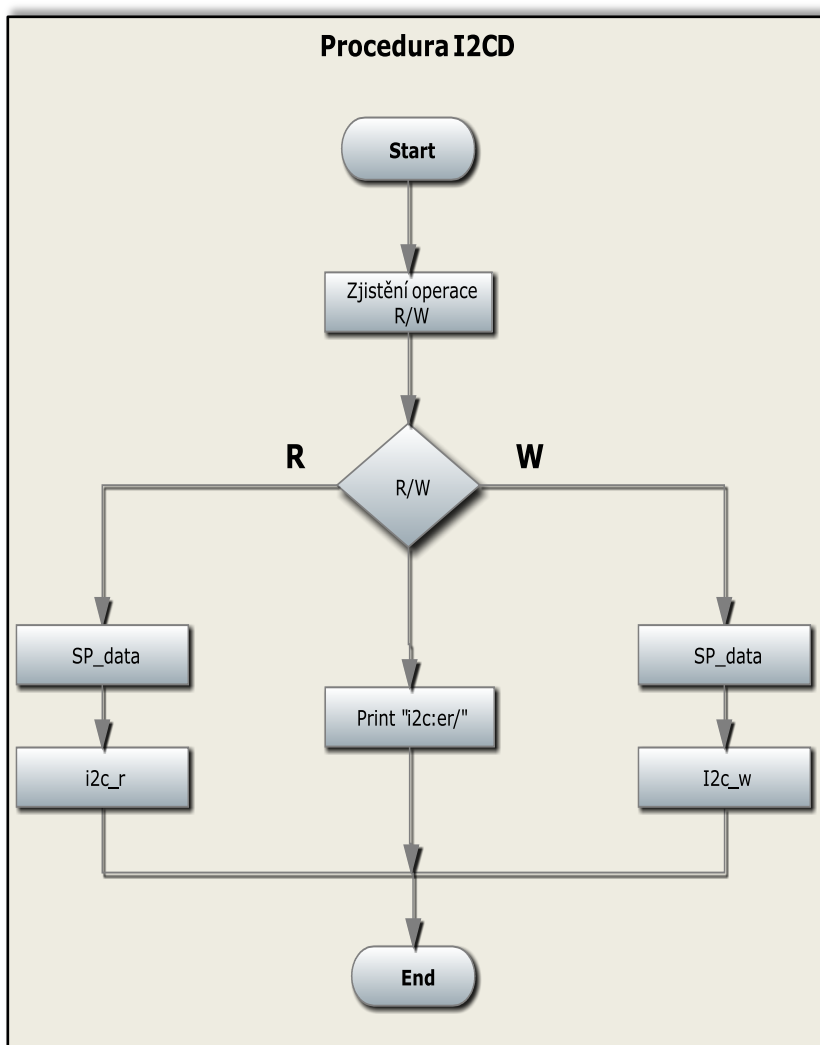


Obr. č. 26. Přerušení od sériového kanálu

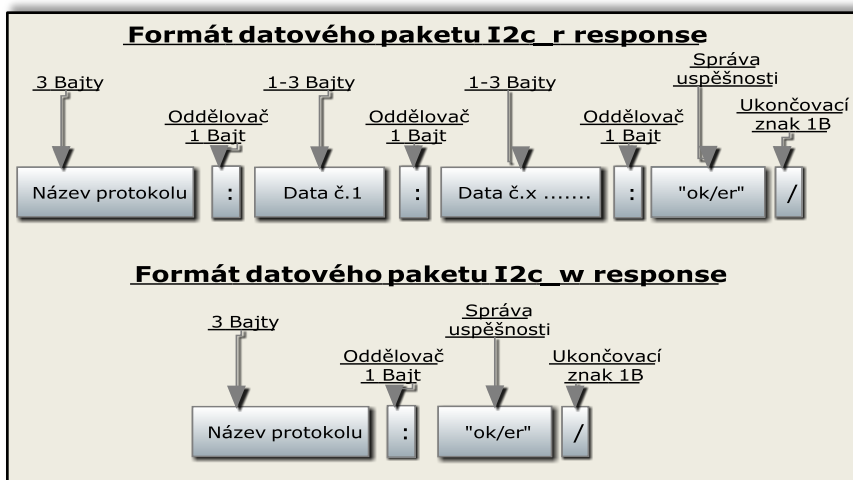
V prvním kroku se zakáže Přerušení od sériového kanálu a proměnné D se přiřadí hodnota z UDR registru, kde je uchován přijatý znak. Tímto přiřazením se automaticky vynuluje příznak přerušení. Dále program pokračuje testem na maximální počet přijatých znaků s čítačem znaků, který ošetřuje případnou kolizi nepřijetí znaku ukončující konec datového paketu. Každý přijatý znak se testuje se znakem '/n' (10) a v případě rovnosti se na konec buferu Buf1 zapíše ukončující znak. Hodnoty z vyrovnávacího buferu Buf1 se přiřadí do buferu Buf a nastaví se flag Flr na 1, což znamená přijetí celého datového paketu. Příkazem Return se běh programu vrací na místo, kde byl původně přerušen.

Na dalším obrázku č. 27 je vývojový diagram popisující výkonnou proceduru I2CD, která je součástí hlavního programu a vykonává požadavky na sběrnici I2C. Prvním krokem zajišťující úspěšnost provedení této procedury je zjištění operace,

kteřá se má provést nad touto sběrnici. V datovém paketu se testují znaky R/W a v případě neúspěšnosti se zařízení master pošle error zpráva zakončená znakem zpětného lomítka, kteřá znamená konec datového paketu ze strany Slave-Master. Jinak program pokračuje podle testu znaků R/W voláním procedury SP_data. Tato procedura analyzuje datový paket uložený v poli znaků Buf a na základě oddělovačů převede znaky na číselné hodnoty, které vrací jako parametr v poli bytů Ar s počtem vrácených hodnot v parametru Cnb. S těmito hodnotami, které se předají procedurám I2c_r nebo I2c_w se provede komunikace na této sběrnici. Úspěšnost provedení procedur je signalizována tzv. ERR bitem a na základě jeho hodnoty se pošle zařízení Master zpráva. Formát zpráv datového paketu posílaný zařízením Master je vidět na obr. č. 28.

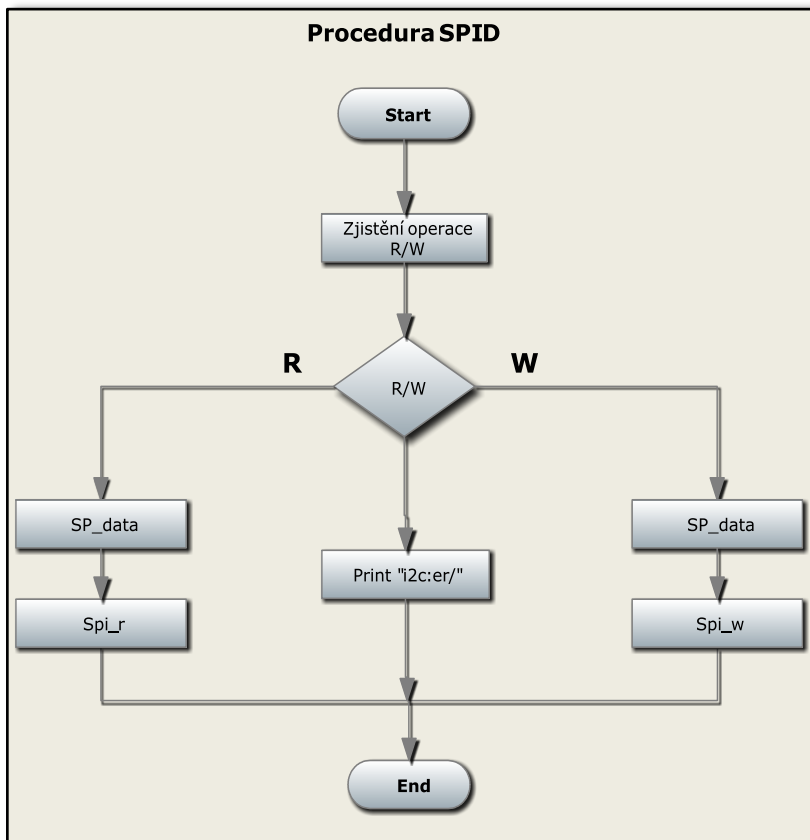


Obr. č. 27. Procedura I2CD

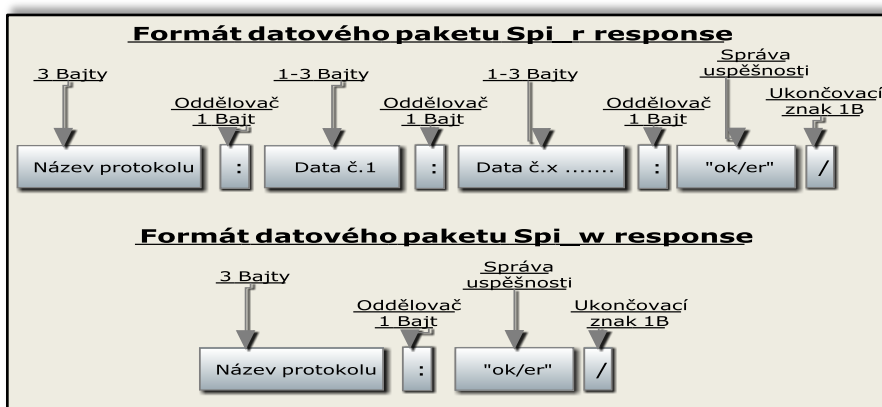


Obr. č. 28. Datový paket I2C Slave-Master

Další vývojový diagram na obr. č. 29 popisuje proceduru SPID. Tato procedura, jak již napovídá název, provádí komunikaci na rozhraní SPI. Tento vývojový diagram zde nebudu popisovat, jelikož se jedná o stejnou koncepci jako u předešlého diagramu. Uvedu zde pouze formát datových paketů posílaných ze strany Slave-Master na obr. č. 30.



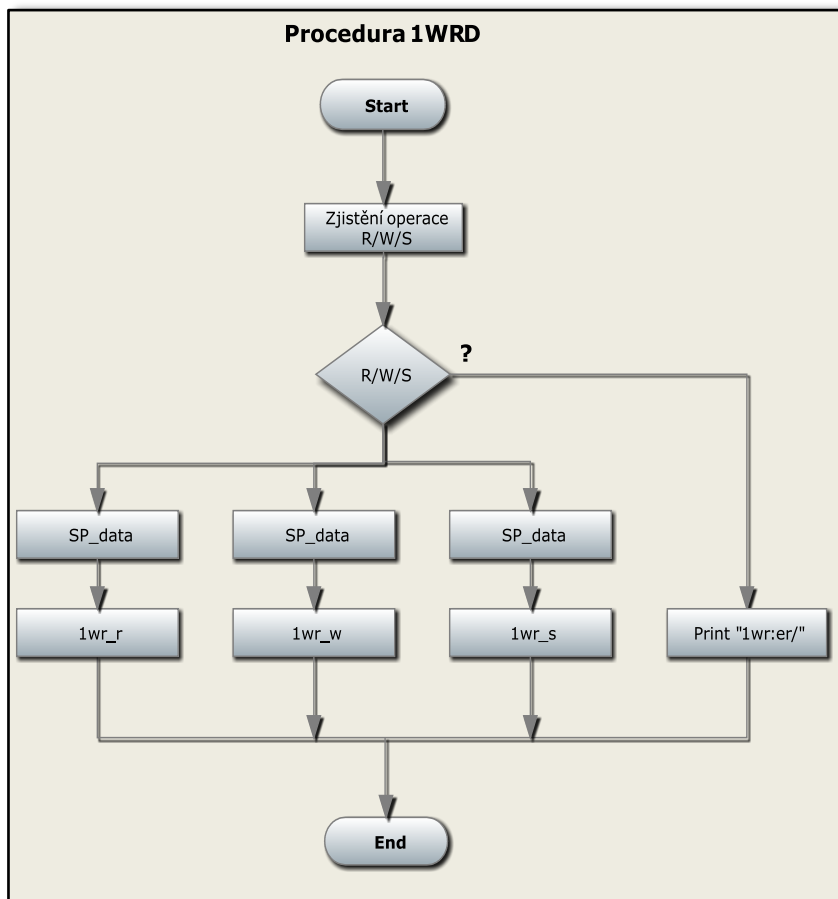
Obr. č. 29. Procedura SPID



Obr. č. 30. Datový paket SPI Slave-Master

Vývojový diagram na obr. č. 31 popisuje proceduru 1WRD, která provádí komunikaci nad rozhraním 1Wire. Na této sběrnici se provádějí tři základní operace, a to R - čtení, W - zápis a S - search. Struktura provedení procedury 1WRD je obdobná jako provedení procedur I2CD a SPID pouze doplněná o operaci S - search. Operace search čte z testovaného obvodu tzv. ROMcode. Jedná se o jedinečné 64bitové číslo, které obecně identifikuje 1wire zařízení na sběrnici. Pro adresaci

operace R/W se Romkod v této aplikaci nepoužívá z důvodu toho, že se předpokládá, že je na sběrnici připojen pouze jeden testovaný obvod. Průběh identifikace operace se opět provádí testem Buf na odpovídající znaky (R, W, S). Program pak dále pokračuje voláním procedury SP_data a předáním parametrů odpovídající proceduře provádějící požadovanou operaci na tomto rozhraní. Výsledky operací jsou posílány zařízení Master v datovém formátu, viz obr. č. 32.



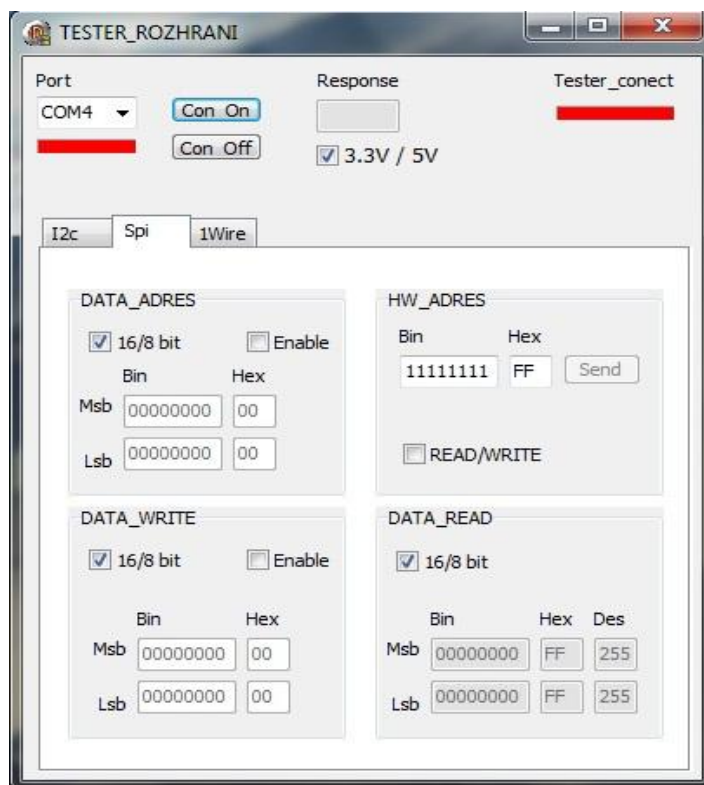
Obr. č. 31. Procedura 1WRD



Obr. č. 32. Datový paket 1WIRE Slave-Master

4.2.2 Zařízení master

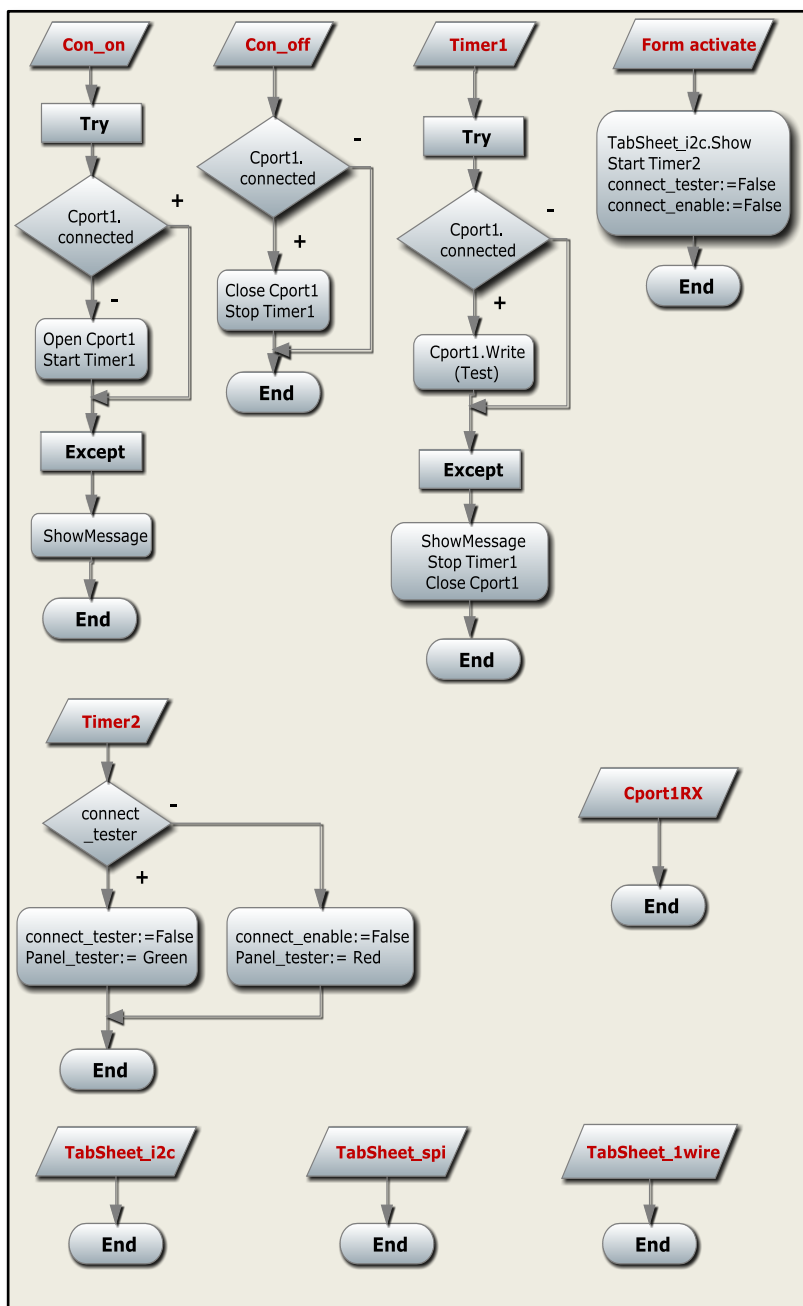
Zařízení je softwarová aplikace napsána v programovacím jazyku Delphi. Jedná se o tzv. pevný Layout (obr. č. 33), kde na formulářové komponentě jsou umístěny ovládací prvky v podobě ComboBoxu pro výběr portu, na kterém bude probíhat komunikace, tlačítka Con_On, Con_Off spouštějí samotnou komunikaci, CheckBox „3,3V/5V“ výběr napětí pro testovaný obvod, Edit „Response“ zobrazující zprávy poslané „Slave“ aplikací, dva led panely indikující stav Testeru a tři propojené TabSheet komponenty prezentující námi zvolené rozhraní na kterých jsou další ovládací prvky (komponenty) Edity pro setování hodnot, Checkbox „16/8 bit“ volba velikosti datových slov, Checkbox „READ/WRITE“ volba operace nad zvoleným rozhraním, tlačítka „Send“ a „Search“ pro zaslání na setovaných hodnot zařízení Slave přes sériovou linku. Pro setování hodnot jsem použil kombinaci binárního a hex zápisu. Jedná se vždy o dva propojené Edity.



Obr. č. 33. Aplikace Master

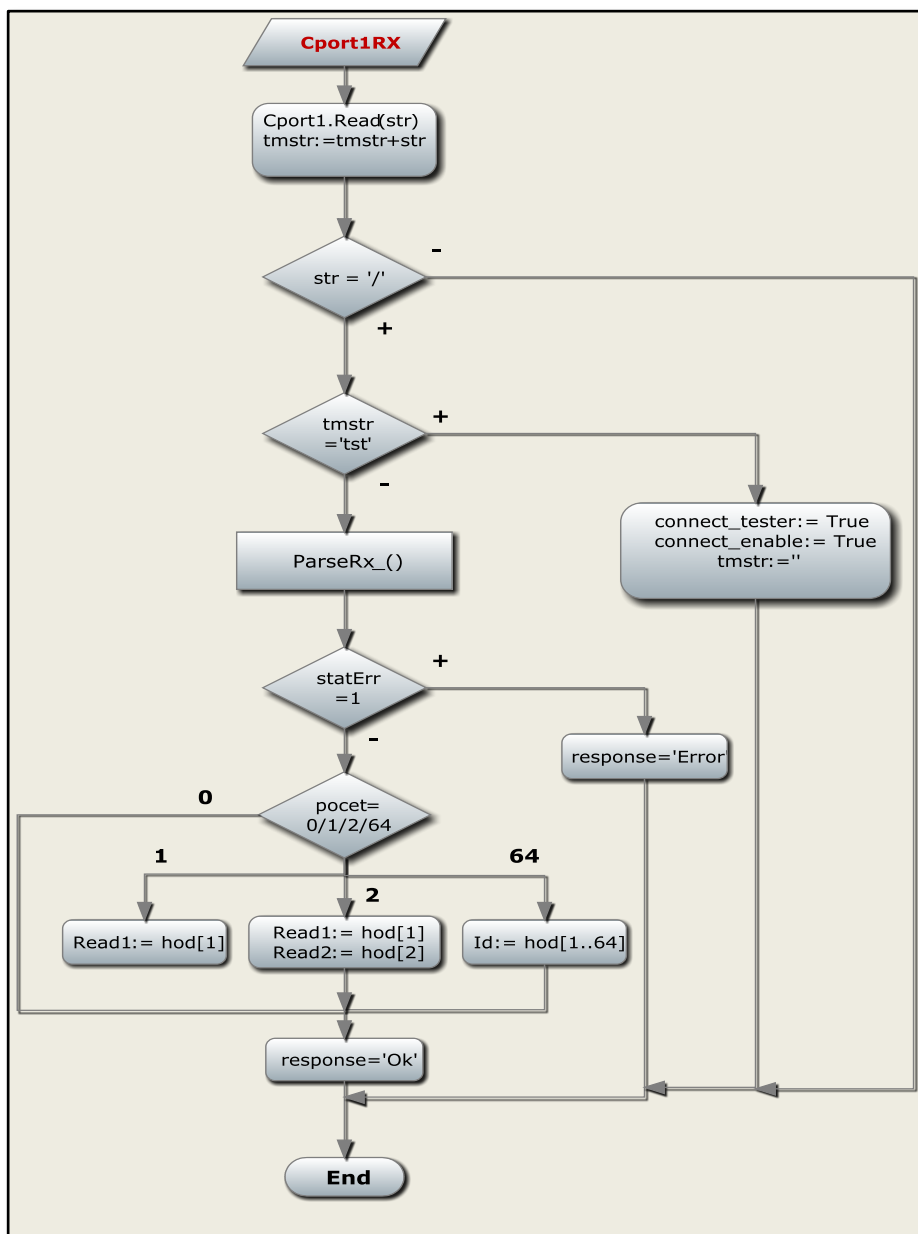
Program aplikace jsem popsal pomocí třech vývojových diagramů. Na vývojový diagram jsem umístil jen ty nejdůležitější asynchronní události, o kterých si myslím, že by měli vysvětlit funkci programu a ty také dále popíšu. První vývojový diagram na obr. č. 34 zjednodušeně zobrazuje celou aplikaci. Jako první při spuštění aplikace se zavolá procedura „Form activate“. V té to proceduře inicializují defaultní hodnoty proměnných, TabSheet_i2c a startují Timer2. V tomto okamžiku je aplikace připravená. Dalším krokem je volba sériového kanálu a aktivování tlačítka „Con_on“. Tím to se vykoná kód, který otestuje otevření sériového portu. V případě, že není otevřen, otevře se a spustí se Timer1. Timer1 je jedna ze tří synchronních událostí, která se v té to aplikaci nachází. Timer1 je procedura, která se volá každých 500 ms. Tato procedura periodicky dotazuje přes sériový port zařízení „Slave“ specifickým formátem: „tst: 0 nebo 1/n“. V tomto přenášeném stringu se posílá požadavek na napájení testovaného obvodu 0 = 3.3V, 1 = 5V.

Úspěšnost provedené operace, která se porovnává v proceduře Cport1RX, podmiňuje používání dalších komponent příznakem „connect_enable“ a ve spojení s procedurou Timer2 příznakem „connect_testr“ i vizualizuje pomocí led panelu „Tester_connect“ stav připravenosti zařízení Slave. Na dalším vývojovém diagramu (obr. č. 35) je znázorněna procedura Cport1RX. Tato procedura se volá pokaždé, pokud dojde k přijetí znaků sériovým portem.



Obr. č. 34. Hlavní vývojový diagram aplikace Master

Procedura Cport1.Read načte do proměnné „str“ znaky přijaté sériovým portem, které dále načítá do proměnné „tmstr“ pro pozdější analýzu. Jako první se

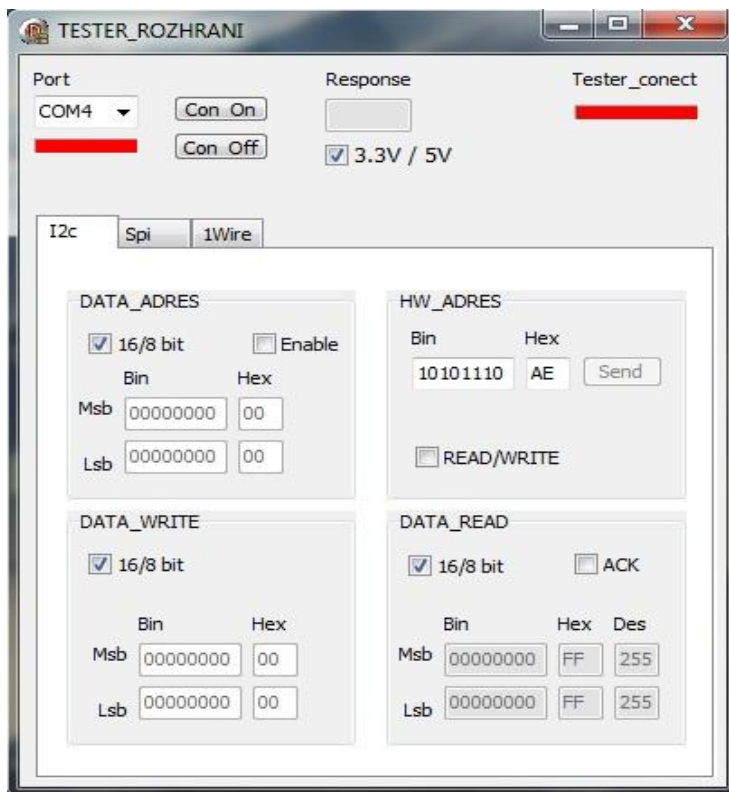


Obr. č. 35. Procedura Cport1Rx

testují příchozí znaky se znakem prezentující konec datového paketu ('/'). V případě neúspěšnosti se procedura ukončí a čeká na přijetí dalších znaků. V okamžiku pozitivního testu program pokračuje vykonáním další podmínky a to testem proměnné „tmstr“ na přítomnost klíčového slova „tst“. Toto klíčové slovo posílá Slave zařízení Master jako odpověď na periodicky se dotazující žádosti zařízení Master ohledně připravenosti TESTERU. V případě negativního testu se předpokládá, že datový paket prezentující proměnná „tmstr“ obsahuje odpovědi ze strany Slave

zařízení na požadavky zařízení Master ohledně námi zvolených protokolu. Tímto se volá procedura „ParseRx“, která analyzuje proměnou „tmstr“ a vrací jako parametry počet přijatých dat, samotná data a příznak „statErr“. Na základě příznaku „statErr“ se rozhoduje, jestli jsou příchozí data platná. Pokud jsou data platná, a nejedná se jen o zprávy potvrzující úspěšnost operace „Write“ na zvolených rozhraních, jsou data na základě velikosti zobrazena v patřičných EditBoxech prezentujících operaci „Read“, jak v binární tak v hexadecimální a desítkové soustavě. Úspěšnost či neúspěšnost operací je vždy zobrazena v EditBoxu „Response“.

Na závěr bych popsal tři TabSheet komponenty s názvem I2C, SPI, 1WIRE, které jsou zobrazeny v hlavním vývojovém diagramu pouze jako těla programu. K popsání těchto komponent použiju screenshoty aplikace z důvodu lepšího porozumění. Jak je vidět na obrázku č. 36, je zde aktivní TabSheet_I2C. Už samotný název napovídá, že jeho volbou budeme požadovat vykonání operace R/W nad zvoleným rozhraním. Na této komponentě jsou umístěny následující komponenty:



Obr. č. 36. TabSheet_I2C

EditBoxy:

- „HW_ADRES“: nastavení hardwarové adresy testovaného obvodu.
- „DATA_ADRES“: nastavení adresy registrů testovaného obvodu z volbou délky datového slova pomocí CheckBoxu „16/8bit“.
- „DATA_WRITE“: nastavení dat, které chceme zapsat na zvolené adresy registru, také s volbou délky datového slova.
- „DATA_READ“: komponenta je neaktivní, zobrazuje pouze výsledky operace „Read“, také s volbou datového slova.

CheckBoxy :

- „READ/WRITE“: volba se nachází v sekci 'HW_ADRES' a nastavuje příslušné operace.
- „READ“: volba se nachází v sekci „DATA_ADRES“ a je aktivní jen v modu „READ“. Tato volba je zde z důvodu zamezení poslání dat „DATA_ADRES“, jelikož to některé testované obvody nevyžadují.
- „16/8bit“: nastavení délky datového slova.

- „ACK“: tuto volbu je možno použít jen v případě operace „Read“, a to z toho důvodu, že některé testované obvody vyžadují v modu „Read“ potvrzení o přijmutí datového slova bitem ACK.

Button:

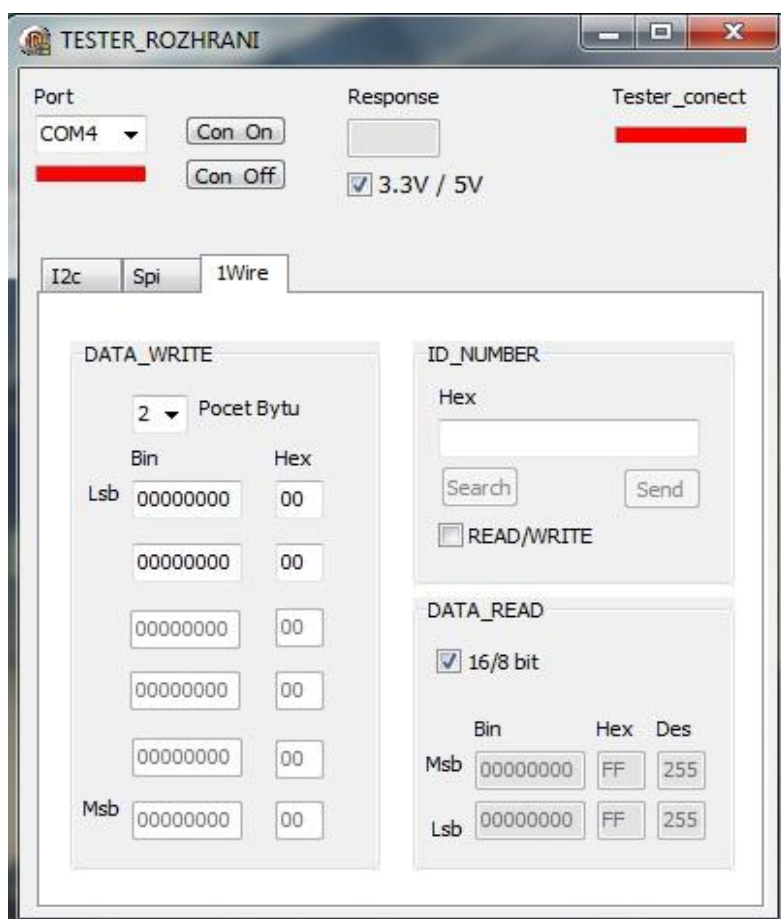
„Send“: Stisknutím tohoto Buttonu se vykoná akce zápisu na sériový port (zaslání požadavku na Slave zařízení), a to v následujícím pořadí:

- proměnná „str“ se zaplní klíčovým slovem „i2c“ identifikujícím protokol a oddělovač,
- 2 - přiřadí se hodnota 0 nebo 1, tato hodnota je podmíněná výběrem operace v CheckBoxu. „READ/WRITE“: 0 = READ, 1 = WRITE plus oddělovač.
- pokud je zvolena operace „Read“ a v sekci „DATA_ADRES“ Checkbox „READ“ je do proměnné „str“ přiřazena hodnota v „DATA_ADRES“ plus oddělovač. V opačném případě se žádná hodnota nepřizuje.
- 4 - přiřadí se hodnota 0 nebo 1. Hodnota je podmíněná výběrem operace v CheckBoxu. „ACK“, 0 = zakázáno, 1 = povoleno.
- 5 - přiřadí se hodnota 1 nebo 2 hodnota je podmíněná výběrem operace v CheckBoxu „16/8bit“ v sekci „DATA_READ“ a znamená kolik bajtů se má vyčíst z testovaného obvodu. Navíc se opět přiřazuje oddělovač a znak konce datového paketu (/n). Tato proměnná je poslána pomocí procedury Cport1.Write na sériový port. Výsledek operace neboli přečtená hodnota je detekována a následně zobrazena procedurou Cport1RX pomocí EditBoxu v sekci „DATA_READ“
- 6 - pokud je zvolena operace „Write“ je do proměnné „str“ přiřazena hodnota v „DATA_ADRES“ plus oddělovač
- 7 - přiřadí se hodnota v „DATA_WRITE“ plus oddělovač a znak konce datového paketu (/n). Tato proměnná je poslána pomocí procedury Cport1.Write na sériový port.

Ovládací prvky umístěné na komponentě „TabSheet_spi“ zde nebudou popisovány, jelikož je jejich struktura a chování podobné těm na komponentě „TabSheet_i2c“ s tím rozdílem, že klíčovým slovem identifikující protokol je „spi“ a nepoužívá se zde CheckBoxu „ACK“.

Poslední komponentou je „TabSheet_1wire“ a jeho zobrazení je vidět na obr. č. 37. Struktura ovládacích prvků je opět podobná jako na ostatních TabSheetech pouze s rozdílem v bodech 1, 2, viz níže.

- **bod 1** - zde je přidáno tlačítko (button) „Search“. Jeho aktivací dochází k čtení tzv. „Romkódu“ testovaného obvodu, tj. dochází k přiřazení klíčového slova do proměnné „str“, oddělovače, znaku „S“ identifikujícího operaci „Search“, oddělovače a znaku konce datového paketu (/n). Tato proměnná je poslána pomocí procedury Cport1.Write na sériový port. Výsledek operace neboli přečtená hodnota „Romkódu“ je detekována a následně zobrazena procedurou Cport1RX pomocí EditBoxu v sekci „ID_NUMBER“.
- **bod 2** - pro adresování a nastavování specifických funkcí testovaných obvodů na 1wire sběrnici je k dispozici pouze sekce „DATA_WRITE“. V této sekci je možno nastavování hodnot variabilnímu počtu přenášených bytů (1 – 6) volených v komponentě ComboBox s názvem „Počet bytů“. Jedná se opět o dva provázané EditBoxy pro binární a hexadecimální zápis požadované hodnoty, kde je počet těchto aktivních EditBoxů prezentován číselnou hodnotou již zmíněném ComboBoxu. Tlačítkem „Send“ jako na ostatních TabSheetech_xx se provede zápis na sériový port a to nejdříve provedením přiřazení klíčového slova „1wr“ do proměnné „str“. Následuje oddělovač, 0/1(R/W), dále se postupně přiřazují hodnoty z EditBoxu od „Lsb“ po „Msb“ v závislosti na jejich aktivním počtu, to vše zakončeno oddělovačem a znakem konce datového paketu (/n). Tato proměnná je poslána pomocí procedury Cport1.Write na sériový port. Výsledek operace je opět detekován a následně zobrazen procedurou Cport1RX pomocí EditBoxu v sekci „DATA_READ“.



Obr. č. 37. TabSheet_1Wire

Závěrem bych chtěl podotknout, že se jedná pouze o stručný popis fungování této aplikace, podrobnosti jsou uvedeny ve zdrojovém kódu aplikace Master, viz příloha.

5. ZÁVĚR

V předkládané práci byl popsán návrh zařízení, které umožňuje otestovat periferní obvody, které disponují rozhraními (sběrnicemi) I2C, SPI a 1WIRE. Zařízení, které bylo nazváno tester, se skládá ze dvou částí, a to „master“ zařízení a „slave“ zařízení. „Master“ je softwarová aplikace běžící pod operačním systémem Windows na běžném osobním počítači. Tato aplikace je napsána v programovacím jazyce Visual Pascal Delphi. Hlavním úkolem této aplikace je ovládat zařízení „slave“ a zobrazovat data jím získaná. Komunikace probíhá přes virtualizovaný sériový port nad rozhraním USB. K realizaci zařízení „slave“ byl použit mikrořadič ATmega8 od firmy Atmel a převodník USB/RS232-TTL od firmy PEKDesign postaveném na obvodu FT232RL, který umožňuje mikrořadiči komunikovat se zařízením Master přes rozhraní USB. Mikrořadič ATmega8 je naprogramován pomocí vývojového prostředí Bascom AVR, kde je implementován překladač jazyku Basic. Mikrořadič pomocí naprogramované aplikace zajišťuje komunikaci se zařízením Master a vykonává jeho příkazy. Na svých vstupně výstupních pinech pak fyzicky realizuje rozhraní I2C, SPI a 1WIRE, pomocí kterých komunikuje s testovanými periferními obvody.

Podle návrhu popsaného v předkládané práci bylo zařízení sestaveno a otestováno. Jeho funkčnost byla ověřena na nejběžnějších periferních obvodech, např. I2C – 24LC128, PCF8574; SPI – 25LC640 a 1WIRE – DS18B20U.

6. SEZNAM LITERATURY

- [1] - DUDÁČEK, KAREL. SÉRIOVÁ ROZHRANÍ SPI, MICROWIRE, I2C A CAN [ONLINE]. PLZEŇ : ZÁPADOČESKÁ UNIVERZITA V PLZNI, 2002 [cit. 2010-12-27].
DOSTUPNÉ Z WWW: <[HTTP://HOME.ZCU.CZ/~DUDACEK/NMS/SERIOVA_ROZHrani.PDF](http://HOME.ZCU.CZ/~DUDACEK/NMS/SERIOVA_ROZHrani.PDF)>.
- [2] - MALÝ, Martin. [Http://hw.cz](http://hw.cz) [online]. 2004 [cit. 2010-12-27]. Sběrnice 1-Wire™.
Dostupné z WWW: <<http://hw.cz/Rozhrani/ART1215-Sbernice-1-Wire.html>>. ISSN 1803-6392.
- [3] - LÁNÍK, Vladimír. [Http://hw.cz](http://hw.cz) [online]. 2005 [cit. 2010-12-27].
MicroLan - A jde to i s jedním vodičem!. Dostupné z WWW:
<<http://hw.cz/Rozhrani/ART1240-MicroLan---A-jde-to-i-s-jednim-vodicem.html>>. ISSN 1803-6392.
- [4] - Atmel Corporation. Data Sheet ATmega8 [online]. U.S.A : Atmel Corporation, 2010 [cit. 2010-12-27]. Dostupné z WWW:
<http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf>.
- [5] Microchip Technology Inc. Data Sheet 24LC128 [online]. U.S.A : Microchip Technology Incorporated, 2010 [cit. 2010-12-27]. Dostupné z WWW:
<<http://ww1.microchip.com/downloads/en/DeviceDoc/21191s.pdf>>. ISBN 978-1-60932-167-3.
- [6] - Philips Semiconductors. Data Sheet PCF8574 [online]. Netherlands : Koninklijke Philips Electronics , 2002 [cit. 2010-12-27]. Dostupné z WWW:
<http://www.nxp.com/documents/data_sheet/PCF8574.pdf>.
- [7] - Microchip Technology Inc. Data Sheet 25LC640 [online]. U.S.A : Microchip Technology Incorporated, 2008 [cit. 2010-12-27]. Dostupné z WWW:
<<http://ww1.microchip.com/downloads/en/DeviceDoc/21223H.pdf>>.
- [9] - Dalas Semiconductor Maxim. Data Sheet DS18B20 [online]. U.S.A : Maxim Integrated Products, 2008 [cit. 2010-12-27]. Dostupné z WWW:
<<http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>>.

7. Přílohy (na CD)

7.1 Zdrojový kód aplikace master a slave

Zdrojový kód aplikace master je uložen v adresáři s názvem aplikace master na přiloženém CD. V tom to adresáři se nachází zdrojové soubory:

- tester.pas
- tester.dfm
- tester.dpr
- tester.exe

Zdrojový kód aplikace slave je uložen v adresáři s názvem aplikace slave na přiloženém CD. V tom to adresáři se nachází zdrojové soubory:

- tester.bas
- tester.bin

7.2 Schéma a Layout DPS

Schéma a Layout Dps je vytvořen v návrhovém systému Altium Designer Summer 09. V adresáři s názvem Dps na přiloženém CD se kromě souborů viz. níže nachází adresář Project Outputs for tester v kterém jsou umístěny Gerber a Ncdrill data potřebné pro výrobu plošného spoje u výrobce.

- tester.SchDoc
- tester.PcbDoc
- tester_sch.pdf
- tester_pcb.pdf
- Project Outputs for tester