

Jihočeská universita v Českých Budějovicích

Pedagogická fakulta

Katedra fyziky

Připojení měřících sběrnic k PC

bakalářská práce

Autor: **Miroslav Kubeš**

Vedoucí bakalářské práce: **Ing. Michal Šerý**

České Budějovice 2008

Anotace

Tato bakalářská práce se zabývá popisem struktury a vysvětlením komunikace sběrnic I2C a 1-Wire, jejich možné připojení k uživatelskému PC přes konektor RS232 a následnému vyvinutí komunikačních softwarů mezi uživatelským PC a sběrnicemi v programovacím jazyku Visual Basic 6.0, které budou přidány k bakalářské práci na kompaktním disku.

Synopsis

This thesis concerns description of structure and explanation of data bus I2C and 1-Wire communication, their connection to user's PC through RS232 connector and sequential software development among user's PC and data buses in program language Visual Basic 6.0. These will be added to this thesis on compact disk.

Tímto bych chtěl poděkovat Panu Ing. Michalu Šerýmu za vedení mé bakalářské práce, za jeho trpělivost, cenné rady a připomínky, ve kterých jsem hledal inspiraci nejen v nejsložitějších chvílích mé práce.

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v úpravě vzniklé vypuštěním vyznačených částí archivovaných pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

1 Úvod	7
<u>Periferie</u>	
2 USB (universal serial bus)	8
2.1 Struktura USB	8
2.2 Linky a komunikace	8
2.3 Schémata zapojení pro jednotlivé rychlosti	9
2.4 Napět'ové úrovně TTL logiky	10
2.5 Napájení	11
2.6 Přenos dat	11
2.7 Zajímavé obvody pro USB.....	11
3 RS 232 rozhraní	12
3.1 Napět'ové úrovně.....	13
3.2 Propojování dvou pc a problém s napájením.	14
3.3 Monitorování sériové komunikace.....	14
3.4 Vzájemné propojení konektorů s plným řízením toku.	15
3.5 Ovlivnění rychlosti přenosu a délky vedení.....	15
3.6 Pojem parita	16
3.7 Asynchronní a synchronní přenos dat	17
<u>Sběrnice</u>	
4 I2C (Philips Semiconductors corporation)	19
4.1 Produkt Tree.....	20
4.2 Napájecí napětí.....	21
4.3 I2C – Bus protokol	23
4.4 I2C – Bus Hardwarové složení.....	25
4.5 I2C – bus s více MCU.....	27
4.6 Jak můžete vědět jestli nějaké další zařízení nevysíla po sběrnici ?	28
4.7 Jaké je riziko poškození dat ?.....	28
4.8 Synchronizace hodin	29
4.9 Zvláštní adresy a výjimky	30
4.10 I2C Fast mode	32
4.11 I2C (High speed mode)	33
4.12 Přenos vysokých rychlostí.....	33
4.13 Rozšířené adresování (10 – Bitů).....	35
4.14 I2C START a STOP podmínka.....	35

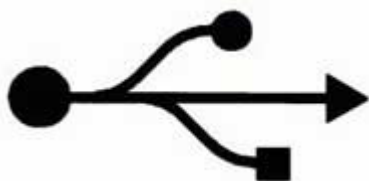
4.15 Vysílání Bitu k Slave zařízení.....	36
4.16 Přijímání Bitu od Slave zařízení.....	36
4.17 Přijmutí Acknowledge signálu od slave zařízení.....	37
4.18 Poslání Acknowledge signálu do Slave zařízení.....	38
4.19 No Acknowledge signál.....	39
4.20 Jednoduchý teploměr pro PC (I2C).....	40
4.20.1 Možnosti DS 1621.....	40
4.20.2 Schéma zapojení a požadované součástky.....	41
5 1-Wire (Dallas Semiconductor Corporation).....	43
5.1 Produkt Tree.....	44
5.2 Temperature sensor.....	44
5.3 Rychlost přenosu dat po 1-wire.....	45
5.4 TTL signál a napájení.....	46
5.5 Přenosová media a topologie.....	46
5.6 Realizace 1-wire systému na personálním pc.....	46
5.7 Počet zařízení na 1-wire sběrnici.....	47
5.8 Příkazy Search a Alarm Search.....	47
5.9 Algoritmus hledání.....	49
5.10 Grafické Znázornění práce algoritmu.....	51
5.11 Reset puls.....	51
5.12 Vysílání dat.....	52
5.13 Zápis Log „0“ a Log „1“.....	53
5.14 Čtení log „1“ a Log „0“.....	54
5.15 Jednoduchý Teploměr pro PC (1-wire).....	55
5.15.1 Schéma zapojení teploměru.....	57
5.15.2 Software.....	59
5.15.3 USB redukce a PC karty.....	61
5.15.4 Vlastní realizace.....	62
<u>Programovací jazyk</u>	
6 Stručný popis programovacího jazyka Visual Basic 6.0.....	63
7 Závěr.....	65
8 Použité prameny a literatura.....	66

1. Úvod

V dnešní době je moderní sběr informací velice důležitý a dá se říci, že bezproblémový přenos informace je jedno z hlavních kritérií komunikace vůbec. Pokud chceme komunikovat se zařízením, nebo potřebujeme zařízení řídit potřebujeme k tomu nějaké vedení, po kterém nám bude plynout tok informací tím způsobem, aby nedocházelo k jejich ztrátě, nebo poškození či zdeformování tak, že by nebyla informace čitelná. V těchto případech se používají moderní komunikační sběrnice jako je I2C, 1-Wire a jiná, jejichž škála použití je široká a neustále se vyvíjí nová zařízení, která jsou schopna po těchto sběrnících komunikovat. K mnohým zařízením můžeme směřovat datové kanály popřípadě adresy. V položkách hromadné výroby jako jsou například Tv sety, různá zvuková zařízení a jiné je snaha o neustále zdokonalování, zlepšování a upravování. Tím roste také počet použitých součástek, je zde velký nárůst datových vodičů. Tomu se snaží výrobce vyvarovat, každou ušetřenou součástkou zvyšuje svůj zisk a tím pádem množství produktů pro koncového zákazníka. Navíc velké množství součástek zvyšuje procento poruchovosti. Zařízení jako celek je více náchylný k poruchám elektromagnetickou interferencí, nebo elektrostatickým výbojem. Na velké množství datových vodičů tažených u sebe na větší vzdálenosti může působit elektromagnetická indukce. Kvalitní moderní sběrnice by měla být konstruována tak, aby těmto věcem předcházela. Z toho důvodu se dá na dnešní sběrnice připojit mnoho zařízení, jsou navrženy tak aby komunikovali pokud možno, po co nejmenším možném počtu vodičů, měly vyhovující rychlosti datových přenosů a komunikace se dala provádět i na větší vzdálenosti. Samozřejmě každá sběrnice ať už jakákoliv má své výhody a nevýhody, které musejí být zváženy na základě parametrů a požadavků, jaké od dané sběrnice očekáváme.

2. USB (universal serial bus)

USB rozhraní se stalo za poslední dobu běžnou součástí každého počítače a dokonale vytlačilo klasický sériový port RS 232 a někde dokonce i paralelní port. Koupit notebook nebo stolní počítač se sériovou nebo paralelní linkou je v dnešní době problém a proto se výrobci musejí stále více přizpůsobovat USB rozhraní.

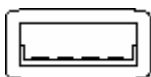


Základní parametry rozhraní USB jsou následující. USB může komunikovat rychlostí od 1,5 Mbit/s až do 480 Mbit/s. Komunikační vzdálenost je pět metrů. Je tu možnost připojení více zařízení. Lze připojit až 127 zařízení pomocí jednoho konektoru.

Programové vybavení USB konektoru plně odpovídá standardu plug & play, který je od Windows 95 obsažen v každém operačním systému.

1.1 Struktura USB

Typy konektorů pro USB rozhraní jsou následující.



Typ „a“ který každý dobře zná, je umístěn na každém novém PC minimálně ve dvou konektorech. Pokud nekoupíme dražší desku, která má zabudovaný USB hub, který obsahuje až osm USB portů.



Druhý konektor typ „b“ je určen pro připojování periferních zařízení a je tady definován standard připojovacího kabelu.

1.2 Linky a komunikace

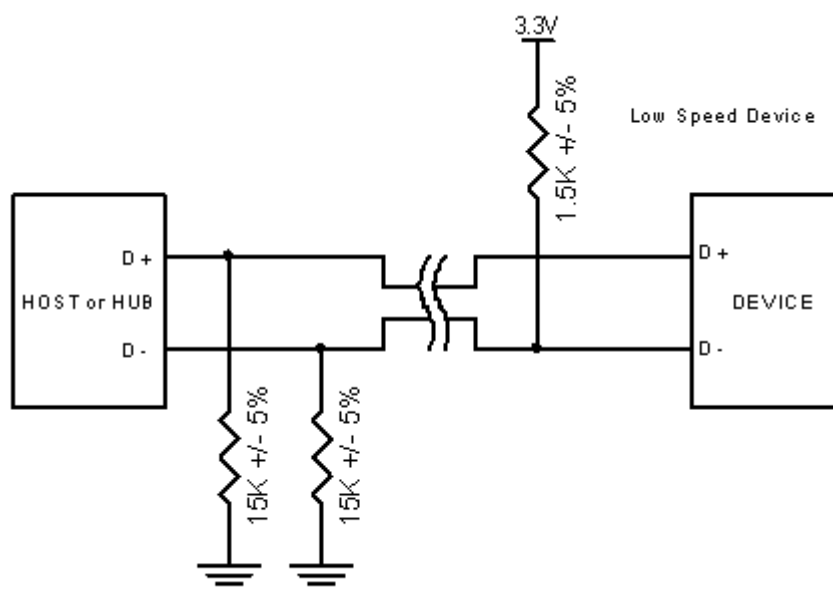
Stíněný	AWG	Max. Délka
Data: 28 AWG twisted	28	0,81
Power: 28 AWG non - twisted	26	1,31
Nestíněný		
Data: 28 AWG non - twisted	24	2,08
Power: 28 AWG non - twisted	22	3,33

Jsou definovány dvě verze fyzické vrstvy USB 1.1 a verze USB 2.0, kde byla doplněna nejrychlejší vrstva. Všechny verze mohou být provozovány současně pro připojení různých typů periférií k jednomu počítači

Při rychlostech „Full speed“ (12 Mbit/s) a „High speed“ (480 Mbit/s) je logická „1“ přenášena diferenciálně. D+ se uvede přes 2,8V s připojeným 15 K rezistorem na GND a D- je pod 0,3V se stejným rezistorem.

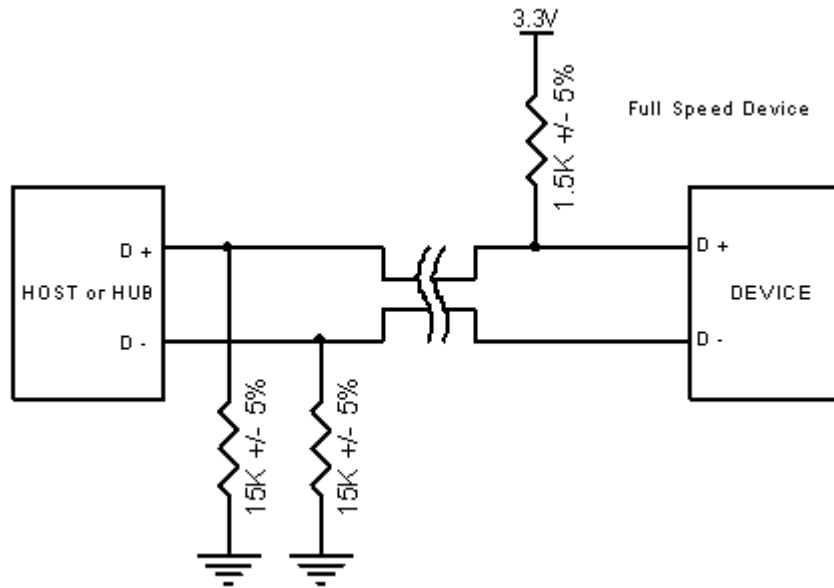
1.3 Schémata zapojení pro jednotlivé rychlosti

Pro správnou funkčnost zařízení je určen speciální nastavovací režim, kdy je D+ nebo D- připojeno ke 3,3V. Full speed zařízení připojí rezistor u D+ a low speed připojí rezistor u D-.



Obrázek č.1 - Low Speed Device (pull up rezistor na D-)

Zařízení podporující High speed je připojují stejně jako zařízení podporující Full speed. Změna rychlosti se řeší softwarově. Zařízení podle standardu nemusí podporovat Full speed, protože není u všech zařízeních potřebná tak velká komunikační rychlost.

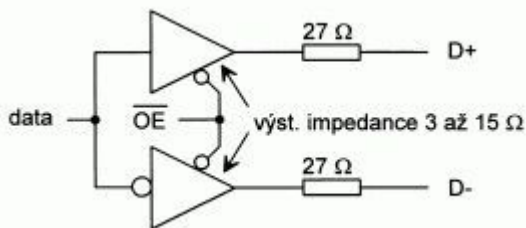


Obrázek č.2 - Full Speed device (pull up rezistor na D+)

1.4 Napět'ové úrovně TTL logiky

Logická nula je 0,3V se zatížením do 1,5K proti napájení. Logická jednička je přibližně 2,8V se zatížením do 15K proti zemi. Pro využití maximální rychlosti 12 Mbps může být kabel dlouhý maximálně 5 metrů a musí být stíněný a kroucený. Nestíněný kabel může mít maximální délku 3 metry.

Pro přenosové rychlosti 1,5 Mb/s nemusí být kabel kroucený ani stíněný. Maximální délka takového kabelu je 3 metry. Pro obě varianty je použito diferenční zapojení vysílačů.



Obrázek č.3 – znázornění zakončovacích R

Maximální vstupní napětí přijímače je 200 mV při souhlasném rušivém napětí 2,6V. V Rozhraní USB jsou dále doplněna zakončovacími odpory.

1.5 Napájení

Napájení je buď přímo z USB sběrnice, pokud odběr elektrického proudu překročí 500 mA (pro jedno zařízení na celé USB sběrnici) Pokud mají USB zařízení vlastní zdroj například počítač, který vlastní systém pro distribuci napájení nezávislý na USB), je řízen USB sběrnici. Každý segment USB umožňuje omezený přenos výkonu pro napájení USB zařízení, přičemž zařízení může být současně napájeno z vlastního zdroje.

1.6 Přenos dat

Pro komunikaci jsou k dispozici tři typy paketů. Každá komunikace začíná vysláním Token paketu, obsahující popis typu a směru dat, adresu USB zařízení a číslo koncové jednotky (Endpoint numer). Poté zařízení vysílá data. Vyšle datový paket, nebo odpovídá, že žádná data nejsou k dispozici. Příjemci strana nakonec vyšle Handshake packet, kterým informuje, zda přenos proběhl úspěšně.

Existují zde dva typy přenosového modelu :

Tok dat (stream) využívající izochronní přenos dat v reálném čase. Nemá přesně definovanou strukturu.

Zpráva (message) využívající asynchronní přenos.

1.7 Zajímavé obvody pro USB

Pomocí integrovaných obvodů firmy FTDI lze skrz USB snadno připojit široké spektrum aplikací. Základní použití obvodu se ale před konstruktérem skrývá. Z hlediska PC se pak jeví jako standardní (ovšem velmi rychlý) sériový port, z hlediska zařízení se buď jeví jako standardní RS 232, nebo obousměrná osmibitová sběrnice 245. FT8U232AM – konvertor USB => UART, FT8U245AM – konvertor USB=> FIFO, FT8U100AX – multifunkční USB hub mikrokontrolér porty RS232, PS/2, IR a uživatelsky definovanými I/O piny (input/output).

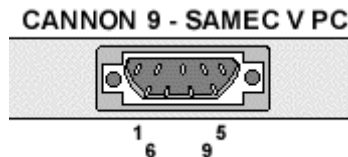
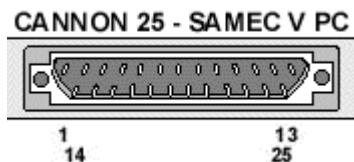
3. RS 232 rozhraní

RS 232 je zařízení, které mělo původně zajišťovat komunikaci dvou zařízení na vzdálenost maximálně dvacet metrů. Z důvodu odolnosti proti rušení je informace přenášena větším napětím než je standardních 5V. Přenosová rychlost je pevně nastavena a přenos probíhá asynchronně sestupnou hranou startovacího impulsu.

PIN	NÁZEV	POPIS
1	SHIELD	Shield ground
2	TXD	Transmit data
3	RXD	Recieve data
4	RTS	Reguest to send
5	CTS	Data to send
6	DSR	Data set ready
7	GND	Systém ground
8	CD	Carrier detect
9 až 19	N/C	-
20	DTR	Data terminal
21	N/C	-
22	RI	Ring indicator
23 až 25	N/C	-

PIN	NÁZEV	POPIS
1	C	Carrier detect
2	RXD	Recieve data
3	TXD	Transmit data
4	DTR	Data Terminal
5	GND	Systém ground
6	DSR	Data set ready
7	RTS	Reguest to send
8	CTS	Clear to send
9	RI	Ring indicator

Obrázky č.4 a č.5 – konektory Canon a popis jejich jednotlivých pinů



- **(DCD)Data carrier detect (CD)** – Používá se na detekci nosné vlny. Zařízení oznamuje mikrokontroléru, že na lince detekoval nosný kmitočet.
- **(RDX) Recieve data** – Tok dat ze zařízení do mikrokontroléru.
- **(TXD) Transit data** – Tok dat z mikrokontroléru do zařízení.
- **(DTR) Data terminal ready** – Mikrokontrolér oznamuje zařízení, že je připraven komunikovat.
- **(SGND) Signal ground** – Signálová zem.

- **(DSR) Data set ready** – Zařízení tímto signálem říká mikrokontroléru, že je připraveno komunikovat
- **(RTS) Request to send** – Mikrokontrolér oznamuje pomocí tohoto signálu, že cesta pro komunikaci je volná.
- **(CTS) Clar to send** – Zařízení tímto signálem oznamuje mikrokontroléru, že komunikační cesta je volná.
- **(RI) Ring Indocator** – Indikátor tvoření, u modemů oznamuje modem terminálu, že na telefonní lince detekoval signál zvonění.

Dříve byly tyto „handshakové“ signály používány pro řízení poloduplexní komunikace s modemy. V duplexní komunikaci některé ze signálů ztrácejí svůj původní význam. Programy je využívají poněkud odlišně, než byl jejich původní význam. Například způsobem, že DTE hlásí, jestli je zařízení připraveno, nebo nepřipraveno komunikovat.

Programátoři k tomu využívají hojně signálů DTR nebo RST, se kterými si většinou mohou v různých programových jazycích libovolně manipulovat. Některé programy umožňují zvolit, které signály se pro řízení dat použijí.

1.8 Napět'ové úrovně

RS 232 používá už standardně dvě napět'ové úrovně. Logickou „1“ a logickou „0“. Log „1“ je někdy označovaná jako marking state, nebo taky klidový stav a Log „0“ je nazývána jako space state.

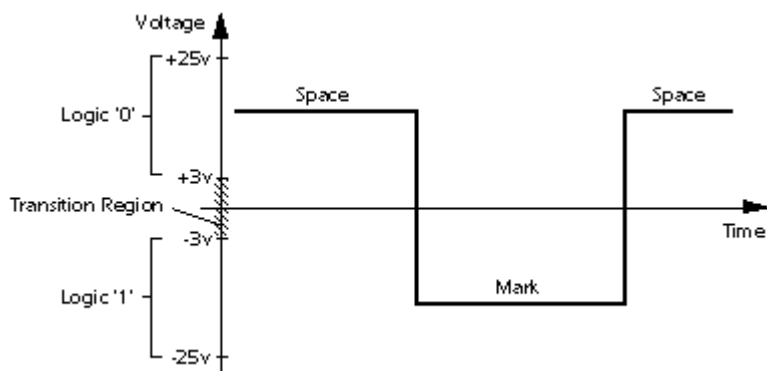
Logická „1“ je indikována zápornou úrovní napětí, zatím co Logická „0“ je indikována kladnou úrovní.

Nejběžněji se pro generování napětí používá napět'ový zdvojovač z 5V a investor. Logické úrovně jsou pak přenášeny napětím +10V pro Log „0“ a -10V pro Log „1“.

	Datové Signály	
Úroveň	Vysílač	Přijímač
Log. L	5V to 15V	3V to 25V
Log. H	5V to -15V	3V to -25V
Nedefinovaný	3V to	3V

	Řídící signály	
Signál	Driver	Terminátor
"Off"	-5V to -15V	-3V to -25V
"On"	-5V to +15V	+3V to +25V

Obrázek č.6 – datové a řídicí signály.

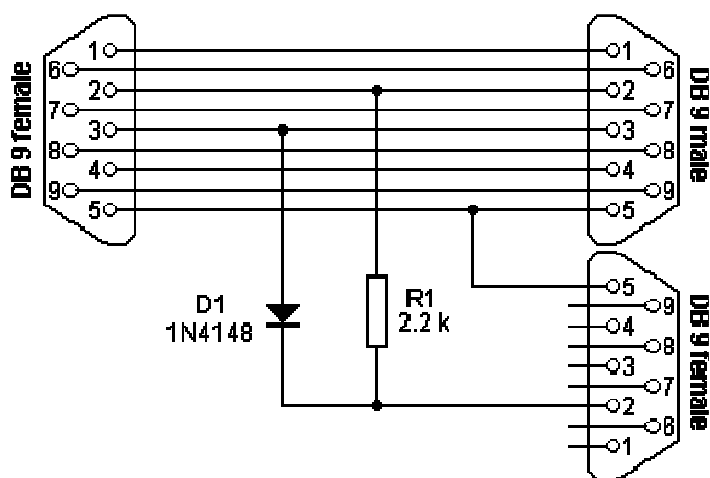


Obrázek č.7 - grafické znázornění přenosové charakteristiky logických úrovní a úrovně napětí v čase

1.9 Propojování dvou pc a problém s napájením

Pokud se přes RS 232 pokoušíme propojit dva počítače a každý z nich je napojen na jinou zásuvku, je dobré změřit napětí mezi jednotlivými zeměmi RS 232 před jejich propojením. Pokud je každý počítač připojen na jinou větev i fázi, může být vlivem různých spotřebičů na každé větvi rozdílové napětí i cca 100V. Takle hodnota bez problémů zničí jakýkoliv RS 232 port.

1.10 Monitorování sériové komunikace



Obrázek č.8 – napojení sledovacího konektoru

Pokud chceme monitorovat komunikaci mezi dvěma zařízeními propojených s PC, lze využít následující zapojení propojovacího kabelu. Dva konektory jsou spojeny přímým kabelem a kontrolní PC je připojen ke třetímu konektoru. Tento monitorovací kabel sleduje komunikaci z obou zdrojů na jediném RS 232 portu. Software si můžeme napsat, nebo je volně ke stažení na internetu.

1.11 Vzájemné propojení konektorů s plným řízením toku

9pin D-sub	to		9pin D-sub
DB9 - 1			DB9 - 1
Recieve data	2	3	Transmit data
Transmit data	3	2	Recieve data
Data terminal	4	6+1	Data set ready + carrier detect
Systém ground	5	5	Systém ground
Data set ready + carrier detect	6+1	4	Data terminal
Request to send	7	8	Clear to send
Clear to send	8	7	Request to send

25pin D-sub	to		25pin D-sub
DB25 - 1			DB25 - 1
Transmit data	3	2	Recieve data
Recieve data	2	3	Transmit data
Data terminal	20	6+8	Data set ready + carrier detect
Systém ground	7	7	Systém ground
Data set ready + carrier detect	6+8	20	Data terminal
Request to send	4	5	Clear to send
Clear to send	5	4	Request to send

9pin Dsub	to		25pin D-sub
DB9			DB25
Recieve data	2	2	Transmit data
Transmit data	3	3	Recieve data
Data terminal	4	6+8	Data set ready + carrier detect
Systém ground	5	7	Systém ground
Data set ready + carrier detect	6+1	20	Data terminal
Request to send	7	5	Clear to send
Clear to send	8	4	Request to send

1.12 Ovlivnění rychlosti přenosu a délky vedení

Maximální možná délka jednoho vodiče se podle standardu RS 232 uvádí kolem patnácti metrů, nebo jako délka vodiče o kapacitě 2500pF. Z toho vyplývá, že pokud jsou použity kvalitní vodiče, tak se můžeme s délkou vedení dostat na zhruba padesát metrů za dodržení standardu.

Kabel je možno používat i za snížené přenosové rychlosti, protože bude navíc odolnější vůči rušení a velké kapacitě vedení. Uvedené parametry počítají s přenosovou rychlostí 19200 Bd.

Texas instrument jako výsledek jejich pokusných měření uvádí tyto délky vodičů a k nim odpovídající přenosovou rychlost. Nicméně, těchto podmínek bylo dosaženo v laboratoři, v praxi můžeme očekávat rušení.

Baud Rate [Bd]	Max Length [ft]	Max length [m]
19200	50	15
9600	500	150
4800	1000	300
2400	3000	900

Tabulka č.9 – přenos dat na určité vzdálenosti

Baud je jednotka, která se používá pro měření přenosu dat na vzdálenosti. Je definována jako přenos dat z jednoho datového média na jiné datové médium. Baud rate udává počet změn signálu za sekundu. Počet jednotlivých změn je udáván v Baudech. V počítačových zařízeních se bere jako základní jednotka jeden Bit (dosahuje hodnot „1“ nebo „0“) Do každé signálové změny lze zakódovat i více než jeden bit. Proto nemůžeme slučovat bps (bit per sekund) s pojmem Baud.

1.13 Pojem parita

Parita nám bez nároku na výpočetní výkon zabezpečuje přenos dat. Ve vysílání po sběrnici se sečtou jednotlivé Bity a doplní se paritním Bitem tak, aby byl zachován buď sudý nebo lichý počet jedničkových Bitů.

- **Sudá parita** – Počet jedničkových bitů + paritní Bit = Sudé číslo
- **Lichá parita** - Počet jedničkových Bitů + paritní Bit = Liché číslo
- **Space parita** – Nulová parita, paritní Bit je vždy na Log „0“
- **Mark parita** – Paritní byt je vždy v Log „1“

1.14 Asynchronní a synchronní přenos dat

Synchronní přenos informací – ve vodičích je nastavena nějaká určitá úroveň, která přenáší informaci, validita (schválení) informace se potvrdí impulsem, nebo změnou úrovně synchronizačního signálu. Synchronizačním signálem jsou informace kvantovány.

Vlastnosti:

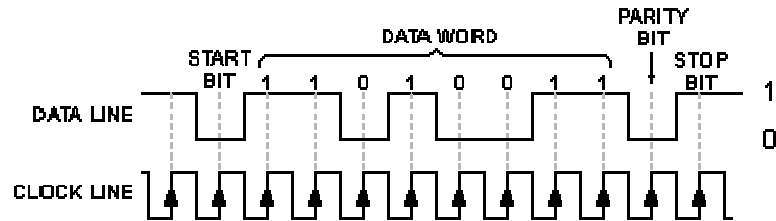
- Výhodné pro velké objemy dat přenášených po více vodičích.
- Nutnost určit, kdo vysílá synchronizační impulsy.
- Nutnost synchronizačního vodiče „navíc“, který nepřenáší v podstatě žádnou informaci.
- Zařízení nemusí být nijak složité.

Asynchronní přenos informací – Data jsou přenášeny v sekvencích. Rychlost přenosu je daná a je neměnná, uvozená startovací sekvencí, na kterou se synchronizují všechna zařízení. Všechny strany musejí obsahovat svůj vlastní přesný oscilátor, na kterém odečítají data v definovaných intervalech. Po ukončení sekvence je další příjem opět synchronizován startovací sekvencí.

Vlastnosti:

- Nevýhodné pro velké objemy dat, ale vhodné pro dlouhá vedení, na nichž by synchronizační vodič činil nezanedbatelné náklady.
- Lze použít téměř všude, ale musí se ošetřit aby počítač neměnil hardwarově přenosovou rychlost.
- Celkem složitá a drahá elektronika z důvodů použití oscilátorů s krystalem.
- Až o 20% menší komunikační rychlost vzhledem k nutnosti použití startovacích Bitů.

RS 232 používá asynchronní přenos informací, Každý přenesený Bite je proto nutno synchronizovat. Pro synchronizaci používáme sestupnou hranu „Start Bitu“ za níž následují posílaná data.



Obrázek č.10 – synchronizace dat

Kompletní přenosová skupina obsahuje přenášená data (7/8 bitová) doplněná o Start Bit, Stop Bit, a Paritu. Přenosový rámec je tedy minimální objem dat který lze přenést. Pokud je potřeba přenosový rámec zastavit je vygenerován nepřetržitý impuls v Log. „0“ po dobu 100 – 600 ms.

4. I2C (Philips Semiconductors corporation)

Sběrnice I2C byla vyvinuta na počátku osmdesátých let (Rok 1981) firmou Philips Semiconductors. Její prvotní účel byl poskytovat snadný způsob spojení CPU a periferních čipů a televizních stanic. Periferní zařízení ve složených systémech jsou často připojena k MCU, jako mapovače pamětí, vstupních a výstupních zařízení a sběrnice dat popřípadě mikrokontroléry paralelních adres. Můžeme připojit vedení k mnohým zařízením a směřovat k nim adresy a datové kanály.

V položkách hromadné výroby, jako jsou TV sety, VCR a zvukové zařízení je zapotřebí, aby se s součástkami šetřilo. Každá ušetřená součástka znamená pro výrobce snížení nákladů a zvýšenou rentabilitu a tím pádem více produktů pro koncového zákazníka. Mimo to, více takovýchto zařízení implementovaných ve výrobku je více náchylných k poruchám elektromagnetickou interferencí (EMI) a elektrostatickému výboji (ESD).

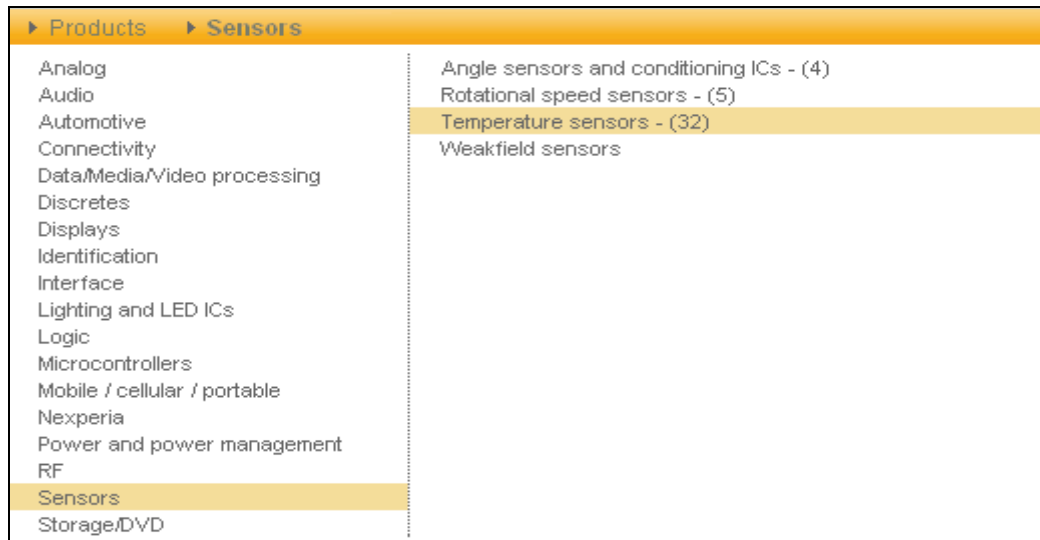
Výzkum provedený v laboratořích firmy Philips v Eindhovenu (Nizozemí) se snažil najít řešení daného problému. Tyto problémy měly za následek vyvinutí dvoudrátové přenosové sběrnice, která byla nazvána I2C – bus. Její jméno doslova vysvětluje její účel. Poskytovat komunikační spojení mezi integrovanými obvody.

Dnes se sběrnice I2C používá i v jiných oblastech, než je právě zvukové a video vybavení. Sběrnice je dá se říct ve skutečnosti používána a brána v průmyslu jako standart. I2C – bus byl adoptovaný několika vedoucími výrobci čipů, jako je Xicor, ST Microelectronics, Infineon Technologies, Intel, Texas Instruments, Maxim, Atmel a také různá analogová zařízení.

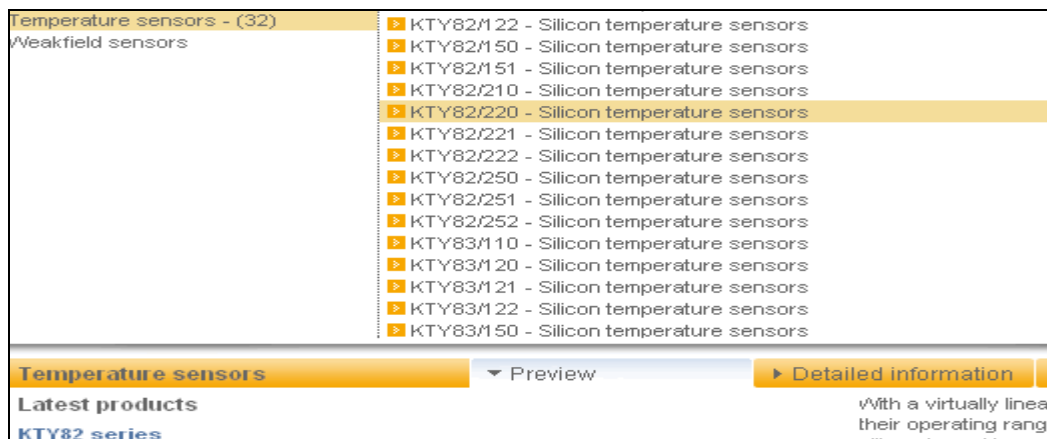
Všechny společnosti samozřejmě uvádět nebudu, vybral jsem ze seznamu ty nejhlavnější. Pravda je ta, že I2C sběrnice je implementovaná v tisícovkách různých součástek, jako jsou teplotní čidla, analogové časovače, operační zesilovače, audio zesilovače, audio signal procesor, mikrokontroléry a mnohá další. I2C – bus licensovaně používá více než padesát různých světových společností

1.15 Produkt Tree

Ukázka stromu produktů které společnost Philips corporation nabízí. Součástek je opravdu hodně. Spotřebitel si zaručeně vybere co potřebuje.



Samozřejmostí je, že ke každé součástce která se ve stromě produktů vyskytuje se dá stáhnout odpovídající obsáhlý datasheed ve kterém jsou všechny potřebné informace které by měli být uváděny výrobcem. Dobrá věc co mě upoutala je, že v každé kategorii produktů je následně poukázáno na novinky a nejnovější výrobky firmy Philips Corporation.



1.16 Napájecí napětí

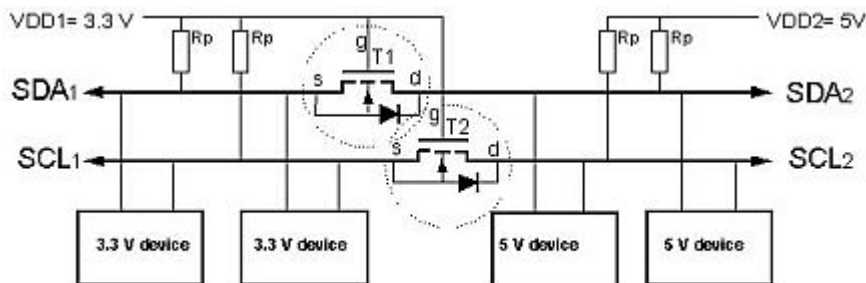
Je tu značná celosvětová snaha řídit napájecí napětí a redukovat ho pod 5V pro nová I2C zařízení. Už nyní je široce užívané napětí 3,3V a dá se počítat s tím, že v budoucnu bude napájecí napětí redukováno ještě více. Nicméně zařízení využívající napájecí napětí 5V jsou ještě dnes běžně používané a stejně tomu bude ještě i nějaký ten čas v budoucnu, kvůli velikému počtu variant zařízení využívajících napájení 5V, která jsou dostupná. Samozřejmě jsou zařízení, které prostě potřebují vyšší voltáž, jako jsou analogové a výkonové napěťové stupně na vstupech a výstupech systémů. Je tu ale potřeba a snaha tyto zařízení vyčlenit, protože by mohla v budoucnu velmi zkomplikovat design nových systémů.

Musel být navrhnout jeden design zařízení pro součástky s různým napájecím napětím. Standardní řešení je použití úrovnového přepínače (Level-shifters). Nicméně v případě I2C sběrnice nemůže být použit standardní úrovnový přepínač, protože signály jsou obousměrné a není na sběrnici žádný přímý indikační signál. V projektování přepínání hladin napětí chtěl Philips zachovat eleganci a jednoduchost, která udělala I2C ve skutečnosti standart jako je dnes.

Obousměrný úrovnový přepínač se skládá z N – kanálů s MOSFETEM, který spojí vzájemně dvě části na I2C – bus systému. Každá část s různým napájecím napětím a různými logickými úrovněmi bude sepnuta. Musejí být zvažovány tři stavy.

1. Žádné zařízení nestahuje vedení „dolu“. Vedení nižšího napětí je zvedáno „nahoru“ zdvihacími odpory k 3,3V. Vedení vyššího napětí je zvedáno až na 5V. MOSFET není ve vodivém stavu
2. Zařízení s napájecím napětím 3,3V strhává vedení k nízké úrovni. MOSFET se stává vodivým a vedení vyšší části je taženo „dolů“.
3. Zařízení s napájecím napětím 5V strhává vedení k nízké úrovni. Nejdříve dioda a poté i MOSFET se stává vodivým a vedení nižšího napětí je taženo „dolů“.

V každém případě, analýza obvodu dokazuje, že logické úrovně jsou přenesené v obou směrech sběrnice systému, nezávisle na řídicí části.



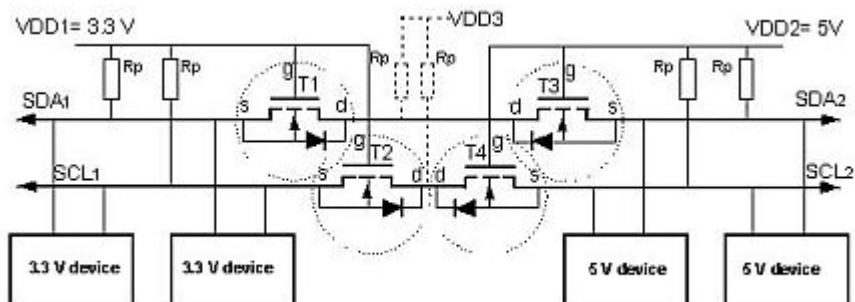
Obrázek č.11 - Obousměrný úrovněový obvod – přepínač spojuje dvě různé napětěové části I2C sběrnice

V závislosti na parametrech MOSFETU, úrovněový přepínač může pobrat široký okruh napájecích napětí. Od nižších než 2V až po vyšší než 5V, pokud bude potřeba. Hodnoty zdvihacích rezistorů (a možné předřadné rezistory mezi vstupem a výstupem vedení) jsou odvíjeny od nejhoršího možného případu komunikace na sběrnici a jejich hodnotách napájecího napětí a logických úrovní, délky a množství vedení a časových požadavků mezi stahováním sběrnice „dolu“ (na Low) a „nahoru“ (na High).

Tento obvod nejen poskytuje přepínání hladin napětí pro I2C – bus pro všechny potřebné voltáže, ale také pro logické úrovně. To nám poskytuje dvě dodatečné funkce. Zprvce nižší sekce napětí může být napětěově odstavena, efektivně je tak odebrána ze systému sběrnice a tím pádem nám bude pracovat vyšší sekce napětí efektivněji. Zadruhé to může být použito jako výborná ochrana nižší napětěové sekce proti výkyvům napětí vyprodukovaných v nižší napětěové části, dokud MOSFET nebude moci sám těmto výkyvům odolat.

Jednoduše, přidáním dalšího MOSFETU na každé vedení dosáhneme toho, že při delších vedeních se nám nebude ztrácet výkon. Můžeme přidat i další sekce s dodatečným napájecím napětím, jaké budeme potřebovat přes dodatečný MOSFET, který zapojíme do běžných svorek kolektoru. (vyšší, nižší, popřípadě stejné) Každá jednotlivá sekce může být izolována od zbytku systému, když odpojíme její napájecí

napětí, avšak všechny ostatní přepínače mezi jednotlivými sekcemi zůstávají operativní. Hodnoty VDD3 a přidaných zdvihacích Rezistorů, které ustanoví definovanou úroveň napětí v zapojení se společným kolektorem tranzistoru řízeného polem jsou volitelné.



Obrázek č.12 - I2C – bus systém ve kterém je vyšší část napětí izolována a opojena od zdroje.

Navíc specifikace I2C sběrnice byla rozšířena tak, aby kryla nižší napájecí napětí. I2C sběrnice může být používána pro veškerá nová zařízení, která potřebují UCC pod 2,7 V. Používáním těchto přepínačů je zaručeno, že I2C sběrnice bude 100% komunikovat i s jinými zařízeními, které nejsou navrženy pro I2C – bus a mají operativní napájecí napětí jiné, než I2C. Toto zajišťuje, že I2C bude dále expandovat i v následujících letech.

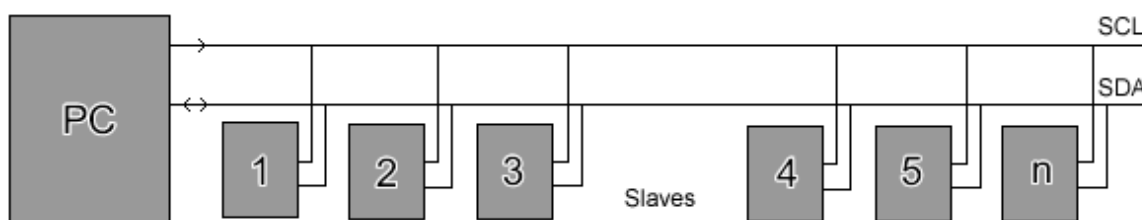
1.17 I2C – Bus protokol

I2C sběrnice se fyzicky skládá z dvou aktivních vedení a zemnicí přípojky. Aktivní vedení nazvané SDA a SCL je obousměrné, jak už bylo uvedeno výše. SDA je sériový kanál dat (Dataline) a SCL je časová linka (Clock).

Každé zařízení připojené na sběrnici, má svou jedinečnou adresu, bez ohledu na jeho typ. (MCU, LDC řadič, paměť, nebo ASIC). Každý z těchto čipů může působit jako vysílač, nebo přijímač v závislosti na jeho funkci. Například LCD řadič je jen přijímač, zatím co paměť, nebo I/O čip (vstupní/výstupní) může být jak vysílač tak i přijímač. I2C sběrnice je multi – master sběrnice. To znamená, že vícero zařízení,

kteří využívají přenos dat, mohou být připojeni k I2C sběrnici. Specifikace I2C protokolu je taková, že IC inicializuje přenos dat na sběrnici, což je nazýváno jako Bus Master. Následkem toho, v ten samý čas jsou všechny další IC zařízení považovány jako Bus Slaves.

Jako Bus Masteri jsou považovány převážně mikrořadiče. (Microcontrollers) Koukněme se na hlavní vnitřní komunikaci po sběrnici I2C.



Obrázek č.13 – blokové znázornění připojení více zařízení na sběrnici

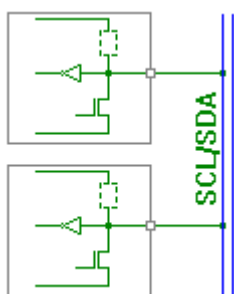
Nejprve musí MCU vyžádat Start signál. Tento signál uvede do pozoru všechna zařízení připojená ke sběrnici. Všechny IC prvky na sběrnici začnou naslouchat přichozím datům po sběrnici. MCU pak pošle adresu zařízení, které chce zpřístupnit. Spolu s indikačním signálem, zda přístup je pro čtení, nebo pro zápis. (zápis v našem případě) Máme přijatou adresu, všechny IC zařízení ji porovnají se svou vlastní. Jestliže adresa neodpovídá jednoduše čekají dál dokud sběrnice není uvolněná Stop signálem. Jestli – že je adresa stejná s daným zařízením, čip bude komunikovat tak zvaným komunikačním signálem (Acknowledge signal). Jakmile MCU přijme komunikační signál, poté může začít vysílání nebo přijímání dat. V našem případě bude MCU vysílat data. Jestliže vysílání proběhne, MCU vysílá Stop signál. To je signál, který nám uvolní sběrnici a zařízení, které jsou na ni připojené očekávají další vysílání, které může kdykoliv začít.

Opět můžeme očekávat několik stavů na sběrnici například: Start signál, Stop signál, Komunikační signál, Adresaci zařízení. Tohle jsou veškeré jedinečné podmínky na sběrnici. Předtím, než si jednotlivé podmínky vysvětlíme, potřebujeme porozumět fyzické struktuře a hardwaru sběrnice.

1.18 I2C – Bus Hardwarové složení

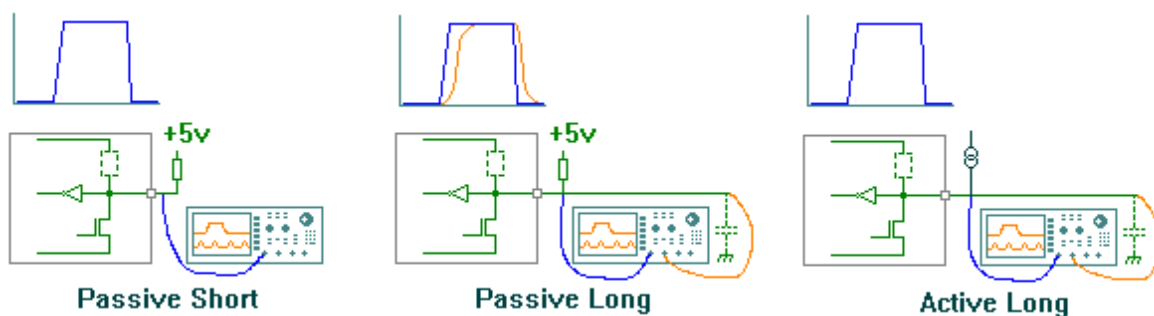
Jak bylo poznamenáno dříve, sběrnice se skládá s SDA vodiče (data) a SCL vodiče (hodiny) a zemnicí přípojky.

Oba vodiče SDA a SCL jsou zpočátku obousměrné. To znamená, že ve specifickém zařízení mohou být tyto linky řízené IC zařízením samotným, nebo jiným externím zařízením. K tomu, aby se dosáhlo téhle funkčnosti, tyto signály používají „otevřený kolektor“, nebo „otevřený emitor“ výstupů. (v závislosti na technologii)



Sběrnicové rozhraní je postaveno na vstupním bufferu a na otevřeném emitoru, nebo otevřeném kolektoru tranzistoru. Když je sběrnice IDLE, sběrná vedení jsou v logické horní úrovni. (všimněte si na obrázku vedle textu, že externí zdvihací rezistory jsou nezbytně důležité pro toto zapojení nicméně jsou snadno zapomenutelné) Pustíme signál na sběrnici, čip řídí jeho tranzistorový výstup, to stahuje sběrnici k nízké úrovni (Low). Zdvihací rezistory jak je vidět ze zapojení, jsou ve skutečnosti malé zdroje proudu. Hezká věc na tomto zařízení je ta, že má zabudované technické řízení sběrnice. Jestli-že je sběrnice obsazena čipem, který posílá log. „0“

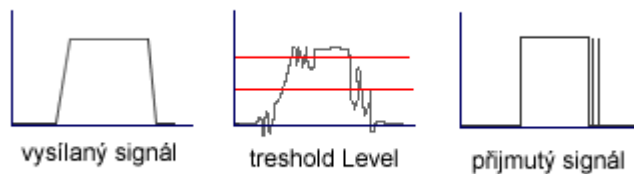
Ovšem i tyto techniky mají své nevýhody. Například metoda s otevřeným kolektorem. Pokud máme dlouhé vedení sběrnice, tak bude mít vážný účinek na jeho rychlost, kterou můžeme získat. Dlouhé linky představují kapacitní zátěž oproti výstupu. Tohle nám bude celkem významně ovlivňovat tvary signálů a IC zařízení nebude schopno zřetelně rozlišit log. „1“ nebo „0“.



Obrázek č.14 – vliv délky vedení na tvaru signálu

Na předchozím obrázku č.14 jsme viděli vliv délky vedení a zakončení sběrnic na křivkách. Další možnost jak se může promítnout dlouhé vedení na signálu je to, že můžeme dostat odrazy ve velké rychlosti, které mohou být tak špatné, že „násobné signály“ ruší náš přenos a mohou zkazit data, které vysíláme. Stačí jenom umístit Schmittův klopný obvod na vstup IC zařízení, který nám bude tyto efekty eliminovat

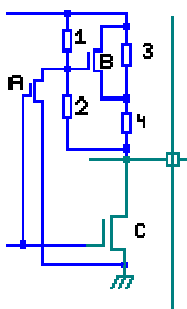
Obrázek č.15 – treshold level



Treshold je úroveň, kdy musí čip sledovat svůj vstup před tím, než detekuje logickou „1“ nebo logickou „0“. Kvůli překonání tohoto problému Philips vyvinul

aktivní „I2C terminátor“. Toto zařízení se skládá z dvojčlenné nábojové pumpy. Dá se to definovat jako dynamický rezistor. V momentu může měnit stavy, to poskytuje velký proud a malý dynamický odpor na sběrnici. Právě proto se je schopen parazitní kondenzátor velmi rychle nabít. Voltáž je zdvižena nad jistou úroveň, kterou jsme schopni detekovat. Výstupní proud je ostře oříznut a snižuje se strměji.

Vezměte si tohle zapojení. Dokud je sběrnice držena na „Low“ (tranzistor C je sepnut) nábojová pumpa je vyřazena, protože báze tranzistoru B je také držena na „Low“ tranzistorem A.

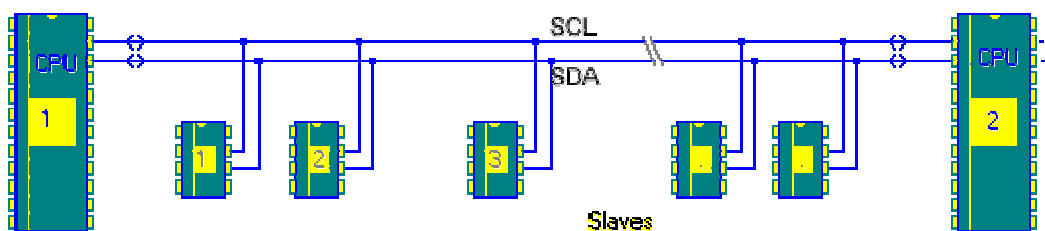


Jakmile čip uvolní sběrnici. (tranzistor A a C jsou vypnuty) Kondenzátor se začne nabíjet, elektrický proud prochází všemi čtyřmi rezistory. (1-4) Snížení napětí přes rezistor (2) způsobí to, že tranzistor (B) sepne a zkratuje nám rezistor (3), který má relativně malou hodnotu. Elektrický proud se začne zvedat. V jistém časovém okamžiku proud mezi bází tranzistoru B a zdrojem nebude dostatečně velký na to, aby nám tento tranzistor udržel ve vodivém stavu. Ten přejde do stavu nevodivého a přestane procházet elektrický proud. V tomhle případě jen externí zvedací rezistor je schopen překonat prosakování elektrického proudu na sběrnici.

Bereme v úvahu, že tohle je zjednodušené vysvětlení. Aktuální zapojení je realizováno mnohem složitějším schématem, kde musí být bráno v úvahu ještě přebíjení jednotlivých čipů, zda nám některý jednotlivý čip nestahuje sběrnici na „Low“ a tak podobně. Toto zařízení se může hodit, pokud potřebujeme I2C vedení, které má být delší než několik metrů.

1.19 I2C – bus s více MCU

Doposud jsme si říkali o I2c - bus a jejích operacích s hlediska používání jen jednoho Masteru na sběrnici. (MCU) I2C byl původně vyvinut, jako multi-master bus. To znamená, že na sběrnici může být připojeno více než jedno zařízení, které nám inicializuje přenosy dat. Pokud používáme jen jeden Master, tak defakto není žádné riziko kolize a ztráty dat kromě toho, že Slave nemusí dobře fungovat nebo, že vedení je špatně nastaveno. Tato situace se mění se dvěma MCU.



Obrázek č.16 – blokové schéma připojení I2C zařízení ke dvěma MCU

MCU1 vyšle start signál a pošle adresu, všechny slave zařízení poslouchají. (Včetně MCU2 které je v té době považováno za Slave zařízení také). Jestli se adresa nerovná adrese MCU2, tak musí zařízení setrvat v klidu, dokud se sběrnice neuvolní Stop signálem.

Pokud oba dva MCU kontrolují co se děje na sběrnici a pokud monitorují transakce signálů na sběrnici dokud není vydaný Stop signál, tak není žádný problém.

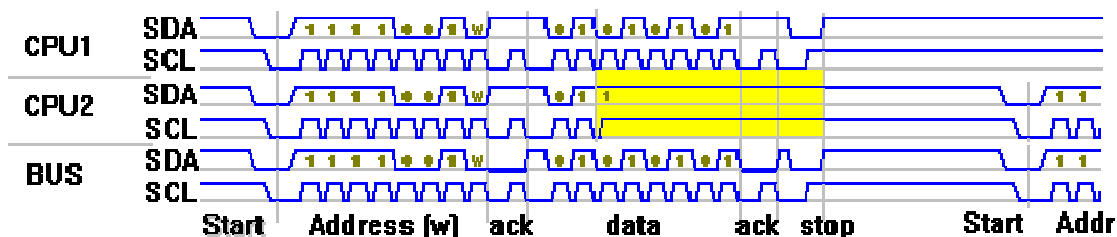
Jestli-že jedno MCU zmešká Start signál a po tomto signálu setrvává v nečinnosti, nebo chce vysílat po sběrnici, když nemá, to je skutečně problém.

1.20 Jak můžete vědět jestli nějaké další zařízení nevysílá po sběrnici ?

Naštěstí nám pomáhá fyzické nastavení sběrnice. Od té doby co byla vyvinuta struktura sběrnice, se používá logický součin AND. (pokud nějaké zařízení stáhne linku na „Low“ zůstává na „Low“) Můžeme testovat, jestli je sběrnice obsazená nebo není. Pokud master změní stav linky na „High“, musí vždycky zkontrolovat, že linka opravdu přešla na „High“. Jestli-že zůstává stažená na „Low“ je to pro něj znamení, že nějaké zařízení linku využívá a stahuje sběrnici na „Low“. Toto je hlavní pravidlo tak zvaného odhadu. Jestli Master nemůže dostat jistou linku na „High“, potřebuje od ní odstoupit a musí čekat dokud Stop signál neuvolní sběrnici před dalším pokusem Mastera, který může opět začít vysílat po sběrnici.

1.21 Jaké je riziko poškození dat ?

Předchozí pravidlo nám říká, že Master ztrácí řízení, když nemůže dostat buď SCL nebo SDA linku na „High“, když potřebuje. Zařízení, které posílá logickou „0“ nám vlastně určuje pravidla na sběrnici. Jeden Master nemůže rušit přenos druhého Mastera, protože pokud nemůže stáhnout linku na „Low“ tak od ní ustoupí a čeká. Stejně se bude chovat i druhý Master. Tento druh nadvlády se bude vyskytovat, jestli-že dvě úrovně přenášené dvěma Mastery nejsou stejné. Proto se podívejme na následující obrázek č.17 , kde dva MCU začnou vysílat zároveň.



Obrázek č.17 – vysílání dvou Masterů současně

Dva MCU nám zpřístupní Slave v zapisovacím módu na adrese „11110011“. Slave se řídí podle tohoto. Doposud si oba Masteři myslí, že mají kontrolu nad Sběrnicí. MCU1 chce přenášet „01010101“ ke Slavu, zatím co MCU2 chce přenést „01100110“ ke Slavu. V tomhle momentě se nám datové bity nerovnejí, protože to co posílají MCU je různé od toho, co máme na sběrnici. Jeden z nich ztrácí kontrolu a ustupuje od sběrnice. Zřejmě to je ten MCU, který nedostal svá data na sběrnici. Dokud nebude na sběrnici poslán Stop signál a MCU už bude mít své data odeslána nebude nadále na sběrnici vyvíjet aktivitu a opouští SDA a SCL linku sám. (Žlutá zóna na obrázku) Dále byl poslán Stop signál, MCU2 se může pokusit o vysílání znovu.

Z výše uvedeného příkladu můžeme usoudit, že Master který nám stahuje linku na Low „dolů“ vždy přebírá kontrolu nad sběrnicí. Master, který chtěl linku zvedat na „High“ v momentě, když je sběrnice tažena na „Low“ jiným Masterem ztrácí kontrolu nad sběrnicí. Téhle podmínce se říká ztráta arbitráže, nebo odstupující podmínka. Když MCU ztratí kontrolu nad sběrnicí musí odstoupit a čekat na vyslaný Stop signál. Poté ví, že předchozí přenos byl dokončený.

1.22 Synchronizace hodin

Všichni Masteři generují na SCL lince jejich vlastní hodiny pomocí kterých převádějí zprávy na I2C sběrnici. Data platí pouze v průběhu trvání „High“ periody na hodinách. Definované hodiny proto musí bit po bitu zaujmout své místo na lince. Hodinová synchronizace je navržena používáním drátového AND spojení I2C rozhraní s SCL linkou.

To znamená že přechod z „High“ na „Low“ na lince SCL způsobí, že zařízení, kterých se to týká začnou počítat „Low“ stavy a jednou za periodu hodin stáhnou SCL linku na „Low“. SCL linka zůstane tak dlouho v tom stavu, dokud na hodinách nepřijde „High“ stav a nezmění ji. Nicméně přechod z „Low“ do „High“ nemusí měnit stav SCL linky pokud další hodiny nejsou stále v jejich „Low“ periodě. Proto SCL linka ví, ve kterém zařízení se koná nejdelší „Low“ doba. Zařízení s kratší „Low“ dobou vstoupí na „High“ a trvá na něm nový čekací stav.

Pro všechny zařízení, kterých se týkalo odpočítávání jejich „Low“ doby => SCL bude uvolněna a půjde na „High“. Poté nebude rozdíl mezi přístrojovými hodinami a SCL linkou a tato zařízení začnou počítat jejich „High“ stavy. První zařízení, aby dokončilo svůj „High“ stav, nám bude stahovat SCL linku na „Low“. Synchronizované SCL hodiny budou generovat s jejich „Low“ stavem na určitém zařízení, které má tuto dobu nejdelší a jeho „High“ perioda je určena nejkratší hodinovou „High“ dobou.

1.23 Zvláštní adresy a výjimky

V I2C mapě adres jsou tak zvané „rezervované adresy“. Tato část obsahuje nějaké další detaily na těchto adresách a co dělají.

Adresa	R/W	Popis
0000-000	0	Hlavní volání adres (Note1)
0000-000	1	Start signál (note2)
0000-001	x	Rezervováno pro (nyní zastaralý) C-Bus formát (note3)
0000-010	x	Rezervováno pro jiný sběrniceový formát (note4)
0000-011	x	Rezervováno pro budoucí účely (note5)
0000-1xx	x	Rezervováno pro budoucí účely (note5)
1111-1xx	x	Rezervováno pro budoucí účely (note5)
1111-0xx	x	10ti bitový Slave adresovací mód (note6)

Tabulka č.18 – přehled některých rezervovaných adres

Note1 : Hlavní volání adres

Tato adresa je užívána pro zpřístupnění všech zařízení na sběrnici, která jsou schopna zacházet s hlavním voláním a mohla by tyto data potřebovat. Zařízení, která jsou schopna s voláním zacházet, ale tyto data nepotřebují tak neodpoví. Všechny bity přijaté po téhle adrese mohou nebo nesmějí být přijaty Slavy, kteří na toto volání odpoví. Jestli žádný Slave neodpoví na přenesený bit, operace je zastavena Stop signálem na sběrnici. Význam hlavní adresy je specifikovaný v prvním přeneseném bitu po tomto „hlavním volání“ tento první bit může obsahovat následující informace.

Jestli-že je LSB nastaveno na „0“

0000-0110

Reset a zápis jsou programovatelné části Slave adresy. Všechny zařízení, které na toto budou reagovat budou resetovány a přijmou programovatelnou část jejich adresy. Když je toto provedeno, jsou znovu přečteny úrovně adres na pinech zařízení. (jestli je nějaké zařízení připojeno) Tento příkaz se také může používat jako reset celého I2C systému.

0000-0100

Stejně jako u případu uvedeného výše, akorát bez resetu. Tohle může být užitečné, když je stav adres na pinech zařízení konfigurovatelný. Tak mohou být adresy změněny.

Jestli-že je LSB nastaveno na „1“

xxxx-xxx1

Toto je hardwarové volání. Jestli-že zařízení potřebuje naléhavou pozornost Mastera, bez informování, kterého Mastera potřebuje, pak může používat tohoto volání. Zařízení si zkrátka zavolá po sběrnici „Kontaktujte mě prosím, potřebuju být opraveno“ Všichni Masteri budou poslouchat. Master ví jak zacházet se zařízením a jeho adresou a vhodně s ním manipulovat.

Note2: Start signál.

Toto může být použito mezi Mastery. Master, který nemá I2c rozhraní v hardwaru, ale v softwaru potřebuje sběrnici monitorovat po celou dobu. To může požadovat mnoho času na pracování (procesing time) Počáteční adresa byla prednastavená. Masteri mohou používat sběrnici v tak zvaném Low-ratu. Jakmile zjistí ze SDA linka je na Low (to se koná za dobu menší než je sedm clock period)

poté může přejít na vyšší vzorkovací kmitočet k tomu, aby zjistil start signál. Tato podmínka není následována Stop signálem ale opakovaným Start signálem.

Note3 a 4

Tyto adresy jsou používány tehdy, když používáme jiná data, než umí I2C vysílat po sběrnici.

Note5

Tyto adresy jsou rezervovány pro budoucí účely, které nám nejsou ještě známy, ale mohly by se v budoucnu objevit.

Note6

Bude detailněji popsáno v sekci o 10bitové adresaci.

1.24 I2C Fast mode

Od uvolnění první I2C specifikace pro veřejnosti (rok 1982) bylo provedeno několik modifikací. V roce 1992 byla uvolněna novější verze I2C parametrů. Tato nová specifikace obsahovala možnost rozšíření I2C sběrnice do Fast mode režimu a také deseti bitovou adresaci.

Ve Fast modu (rychlý režim) nejsou pozměněny fyzické parametry sběrnice. Protokol, sběrnice úroveň, kapacitní zátěž a tak dále zůstalo nezměněno. Nicméně rychlost přenosu byla zvětšena na 400 Kbit/s. K realizaci tohoto kroku bylo ale potřeba upravit časování sběrnice. Všechny CBUS aktivity byly zrušeny, proto nadále sběrnice není kompatibilní s CBUS časováním. Vývoj kompatibility IC zařízení s CBUS byly nadlouho zastaveny. CBUS zařízení prostě nemohou ovládat tyto vyšší taktovací kmitočty.

Na vstup Fast mode zařízení se osazují Schmittovy klopné obvody, které nám potlačují hluk. Výstupní blok nám zahrnuje obvody pro kontrolování sestupné hrany

signálů po vodičích SDA a SCL. Pokud není zařízení připojeno ke zdroji, Fast mode zařízení jsou vypnuté. Jednotlivé piny nám nesmějí svým vysíláním ovlivňovat sběrnici. Zdvihací obvody musejí být přizpůsobeny. Pro kapacity 200 pF nemusejí být zdvihací rezistory upravovány, ale pro kapacitu mezi 200pF a 400pF musí být předefinován zdroj proudu.

1.25 I2C (High speed mode)

Vysokorychlostní mód (H mode) nabízí neuvěřitelný skok v přenosových rychlostech po I2C sběrnici. Hs-mode zařízení komunikují v rychlostech až do 3,4Mbit/s. Navíc zůstávají plně kompatibilní se zařízeními, které komunikují na nižších rychlostech jako jsou Fast nebo Standard – mode (F/S mode) zařízení pro obousměrnou komunikaci na sběrnici. Hodinová synchronizace je vykonána během Hs- mode přenosu. Sběrníkový protokol a formát dat je stejný jako u F/S – mode systému. Nová zařízení mají Fast nebo Hs – mode I2C – bus rozhraní, ačkoliv se upřednostňují Hs – mode zařízení, protože jsou navrženy k většímu množství aplikací.

1.26 Přenos vysokých rychlostí

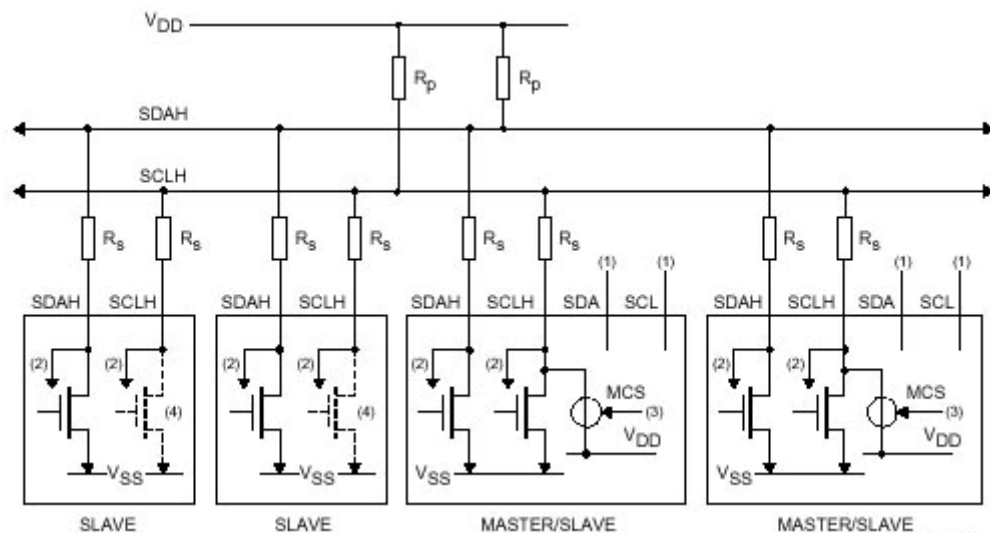
Pro dosažení přenosové rychlosti až do 3,4 Mbit/s musely být provedeny následující zlepšení a zapsány do specifikace I2C sběrnice.

- Hs – mode zařízení mají otevřený Emitter (open-Drain) na výstupním bloku pro SDAH signál. Kombinací strhávání otevřeného Emitoru a zvedání zdroje proudu v obvodu získáváme signál, který je přiveden na SCLH výstup. Zdroj proudu v obvodu zkracuje dobu po kterou nám signál na SCLH roste. V jeden čas je možno přivádět elektrický proud jen do jednoho Mastera, pouze po dobu trvání Hs – modu.
- Žádné rozhodování o Masteru, nebo hodinová synchronizace není vykonávána během Hs – mode přenosu. Procedura o rozhodnutí Mastera

(arbitration procedure) je vždy ukončena po předcházejícím vysílání Mastera ve F/S módu.

- Hs – mode Master zařízení generují sériové hodinové signály z „High“ na „Low“ úroveň s poměrem $\frac{1}{2}$. To snižuje požadavky pro nastavování (set-up) a paměť (hold – time).
- Hs – mode Master zařízení mohou mít implementované můstkové zapojení. Během Hs – mode přenosu, vysokorychlostní data (SDAH) a vysokorychlostní sériové hodiny (SCLH) jsou oddělovány toutle formou můstkového zapojení. U zařízení F/S – mode jdou to linky SDA a SCL. To redukuje kapacitní zátěž SDAH a SCHL linek majících za následek rychlejší vzestup a pád časového signálu.

Následující obrázek č.19 fyzické I2C sběrnice, konfiguruje jen Hs – mode zařízení. Piny SDA a SCL jsou použity jen ve smíšených rychlostních systémech a nejsou spojeny v čistě Hs – mode zařízeních. V takových případech mohou být použity pro jiné funkce.



Obrázek č.19 – I2C sběrnice se zařízeními pracujícími v High speed módu

Popis obrázku č.19. (1) Linky SDA a SCL nejsou připojeny a mohou být využity jiným způsobem. (2) Výstupní filtry (3) Zvyšováním proudu můžeme uvést do

provozu jen jednoho aktivního Mastera. (4) Osazené tranzistory mají otevřený emitorový výstup a mohou strhávat signál sériových hodin SCLH.

1.27 Rozšířené adresování (10 – Bitů)

Kvůli rostoucí popularitě sběrnice I2C byl 7-bitový adresový prostor vyčerpán. To nadneslo problém, který se týkal vyvíjení nových I2C zařízení. Proto I2C standart byl aktualizovaný k tomu, aby realizoval 10-bitový adresační režim.

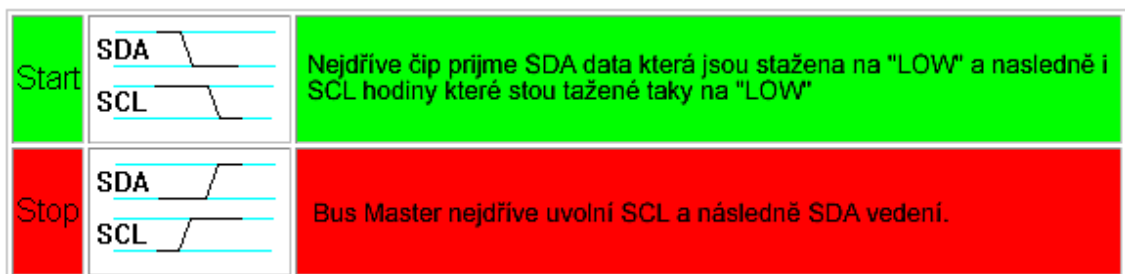
Čip, který se přizpůsobil novému standardu, přijímá dva adresační bity. První se skládá z rozšířeného adresování rezervované adresy zahrnující dvě MSB adresy zařízení a Read/Write bit. Druhý bit obsahuje osm LSB adresačních bitů.

Toto schéma nám zajistí ,že 10-bitový adresační režim zůstává plně transparentní pro další zařízení na sběrnici. Všechny nové designy zařízení by měli mít implementováno toto nové adresační schéma.

1.28 I2C START a STOP podmínka

Před každou transakcí na sběrnici musí být vyslaná po sběrnici START podmínka. START podmínka je signál, kterému naslouchají všechna IC zařízení a udává jim, že se něco na sběrnici začíná dít. Všechny čipy poslouchají.

Když je START podmínka dokončena, je poslána STOP podmínka po sběrnici. Toto je také signál pro všechna zařízení na sběrnici. Jestli-že čip byl zpřístupněný svou adresou a má přijatá data během poslední transakce, musí je následně zpracovat. (pokud už je nemá zpracované během přijmutí zprávy.)

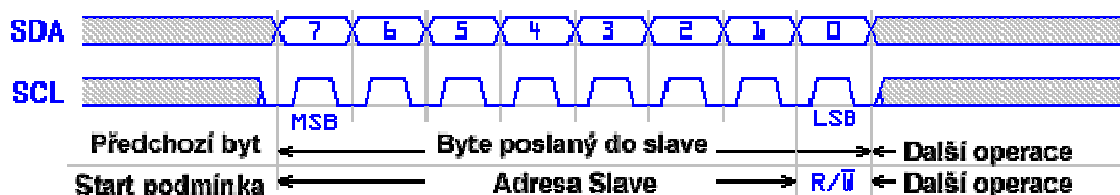


V I2C sběrnici je běžné, že je vyslán vícenásobný START signál. Tak zvaný „opakovaný začátek“. STOP podmínka vždycky značí konec přenosu. I když je

vyslána uprostřed transakce, nebo uprostřed bitu. Čip v tomto případě nedbá na rozposlanou informaci a zahazuje ji a pokračuje v poslouchání čekající na nový začátek.

1.29 Vysílání Bitu k Slave zařízení

START podmínka byla poslána, byt může být poslán od Mastera ke Slave. Tento první bit bude identifikovat Slave zařízení na sběrnici (adresa) a vybere mu mód operací. Význam všech následujících bitů závisí na Slave zařízení.



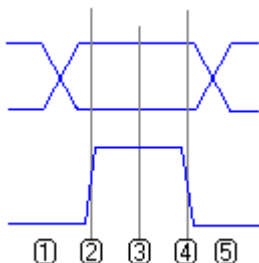
Obrázek č.20 – grafické znázornění přenosů bytů ke Slave zařízení

Řada adres byla rezervována pro zvláštní účely. Jedna z těchto adres je rezervována pro rozšířený adresační mód. Jsou-li Slave zařízení na sběrnici, která umějí pracovat s prodlouženým 10-bitovým adresačním režimem, budou všichni reagovat na cyklus, který poslal Master. Druhý bit, který přijmou od Mastera, bude pak posuzován proti jejich adrese. Dokonce i v 10-bitovém rozšířeném adresačním módu, Bit 0, první bit po START podmínce, uvádí Slave zařízení do jednotlivých módů. („1“ = Read ; „0“ = Write).

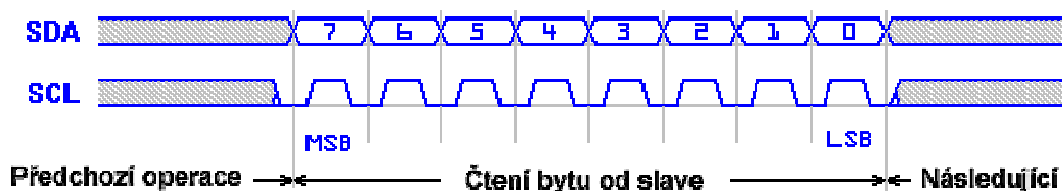
1.30 Přijímání Bitu od Slave zařízení

Jestliže byl Slave adresovaný a potvrdil to, pak od něj může být byte přijatý, pokud je R/W bit nastaven na Read (nastaven na „1“). Protokolová syntaxe je stejná jako v případě vysílání bytu ke slave zařízení, až na to, že se nyní Masteru nebude dovoleno pracovat s SDA linkou. Před posláním osmi hodinových impulsů potřebných k příchodu bytu na SCL vedení, Master uvolní SDA linku. Slave nyní bude tuto linku ovládat sám. Linka se bude zvedat na „High“ pokud bude Slave chtít

přenášet Logickou „1“ nebo zůstane na „Low“ pokud bude chtít přenášet logickou „0“.



K obrázku vedle textu. Všechno co musí Master udělat je to, že vygeneruje rostoucí signál na SCL lince (2) číst úroveň SDA (3) a generovat padající signál na SCL lince (4). Slave nebude měnit data během času, kdy je SCL linka na „High“. Během (1) a (5) může Slave měnit stav SDA linky. Pro shrnutí této sekvenci musí předcházet osm hodinových signálů, kvůli dokončení datového bitu. Nejprve jsou vždycky přenášeny bity MSB.

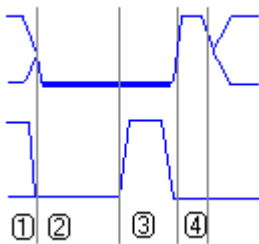


Obrázek č.21 – grafické znázornění přenosu bytu od Slave zařízení

Význam všech bitů pro čtení závisí na Slave zařízení. Není tu nic podobného jako stavový registr. Musí se každý sám podívat do datasheetu, co každý jednotlivý bit který je přenesený znamená.

1.31 Přijmutí Acknowledge signálu od slave zařízení

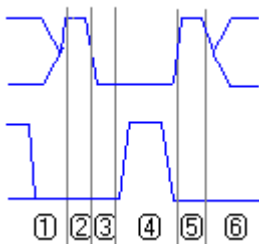
Když adresa, nebo datový bit byl přenesen na sběrnici, musí následovat potvrzení Slave zařízení (Acknowledge signál). V případě adresy: jestli se shoduje adresa Slave zařízení s adresou obsaženou v potvrzovacím signálu, bude Slave zařízení reagovat. Slave, který přijmul potvrzovací signál, stáhne SDA linku na „Low“ ihned po přijmutí osmi bitů. V praktických případech tohle nebude rozeznatelné.



K obrázku vedle textu. To znamená, že jakmile Master stáhne SCL linku na „Low“ dokončuje se zrovna přenos bitu (1). SDA bude staženo na „Low“ Slave zařízením. Master nyní vydá hodinový impuls na SCL lince (3). Master uvolní pro Slave zařízení linku SDA na dokončení tohoto hodinového impulsu (4). Na sběrnici je nyní možnost znova posílat data k Masteru, nebo generovat Stop signál. V případě, že data měla být zapsána do Slave zařízení, musí se tento cyklus dokončit před Stop signálem. Slave zařízení bude SDA linku držet na „low“ dokud Master nevygeneruje hodinový impuls na lince SCL.

1.32 Poslání Acknowledge signálu do Slave zařízení

Pro přijetí bitu od slave zařízení, Master musí uznat, že Slave zařízení smí vysílat. Master je v plné kontrole nad SDA a SCL linkou.



K obrázku vedle textu. Po přenosu posledního bitu Masteru (1) slave zařízení uvolní SDA linku. SDA linka by měla jít na „High“ a Master si jí opět stáhne na „Low“ (3). Další master vyšle hodinový impuls na SCL linku. Po dokončení tohoto hodinového impulsu Master opět uvolní SDA linku. Slave zařízení nyní znovu získá kontrolu nad SDA linkou.

Uznat bit, který Slave zařízení vysílá je vždy nezbytné s výjimkou posledního přijatého bitu. Jestli-že Master chce zastavit přijímání dat ze Slave zařízení, potom musí být schopný poslat Stop signál. Poté získá od Slave zařízení opět kontrolu nad SDA linkou po acknowledge signálu vyslaném Masterem. To by mohlo způsobovat problémy.

Pojďme přijmout další bit, který byl poslán Masteru a který je Logická „0“. SDA linka je tažena na „Low“ ihned Slave zařízením po procesu, Kdy Master stáhne SCL linku také na „Low“. Master se nyní pokusí vygenerovat Stop signál pro zastavení přenosu na sběrnici. To uvolňuje SCL linku jako první a potom zkouší uvolnit SDA linku, která je tažena na „Low“ Slave zařízením. To znamená, že žádný Stop signál

nebyl generován na Sběrnici. Tato podmínka je nazvaná NACK: „NOT Acknowledge“ signál. Nespleťte si to s „No Acknowledge signalem“.

- **Not Acknowledge signal (NACK)** – Může být použit jen po tom, když Master přečetl bit ze Slave zařízení.
- **No Acknowledge signal** – Může být použit jen po tom, když Master zapsal bit do Slave zařízení.

1.33 No Acknowledge signál

Tohle není přímo definované jako podmínka na sběrnici. Je to pouze stav toku dat mezi Masterem a Slavem.

Jestli po přenosu osmi Bitů z Mastera ke Slave zařízení nebude stažena SDA linka na „Low“, potom je to tento stav považován za No acknowledge signál.

To znamená že jeden nebo druhý:

- Slave není nalezen (co se adresy týče).
- Slave zmeškal puls a dostal se ze synchronizace se SCL linkou Mastera.
- Sběrnice je „přilepena“ jedna z linek je permanentně tažena na „Low“.

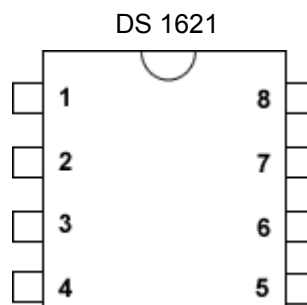
V každém případě Master by se měl pokusit přerušit spojení Stop signálem na sběrnici a provést test na „přilepení sběrnice“. Stop signál může být cyklováný.

1.34 Jednoduchý teploměr pro PC (I2C)

DS 1621 je digitální teploměr a termostat, který poskytuje 9-bitové čtení teploty, která indikuje teplotu zařízení. Teplotní výstup (Termostat), Tout je aktivní, když teplota zařízení TH překročí uživatelsky nastavenou teplotu TH. Výstup setrvává ve svém stavu, dokud teplota neklesne pod uživatelsky nastavenou teplotu TL, toto chování umožňuje nastavení hystereze.

Uživatelská nastavení jsou uchovávány v paměti DS 1621 dokud nebude čidlo odpojeno od napětí, potom se paměť vymaže. Schopnost uchovávání v paměti poskytuje tomuto čipu možnost programování v reálném čase a přímo v technologii. Teplotní nastavení a čtení teploty jsou zprostředkovány přes sériové 2-vodičové vedení. Díky možné adresaci DS 1621 je možno připojit až 8 těchto zařízení paralelně.

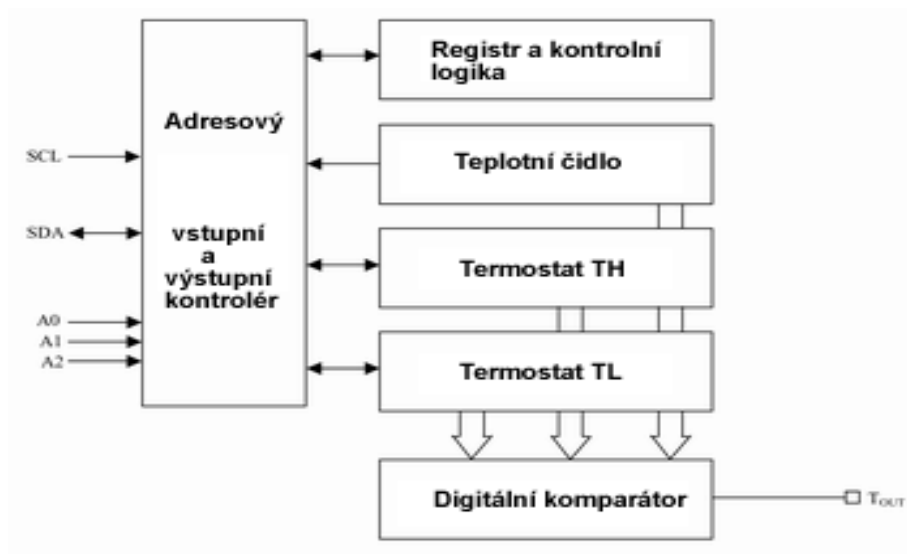
PIN	OZNAČENÍ	FUNKCE
1	SDA	Datový vstup / výstup
2	SCL	Hodinový vstup / výstup
3	Tout	Výstup termostatu
4	GND	Zemní pin
5	A2	Adresový vstup A2
6	A1	Adresový vstup A1
7	A0	Adresový vstup A0
8	Vdd	Napájecí pin



Obrázek č.22 – popis čidla SD1621 a jeho jednotlivých vývodů

1.34.1 Možnosti DS 1621

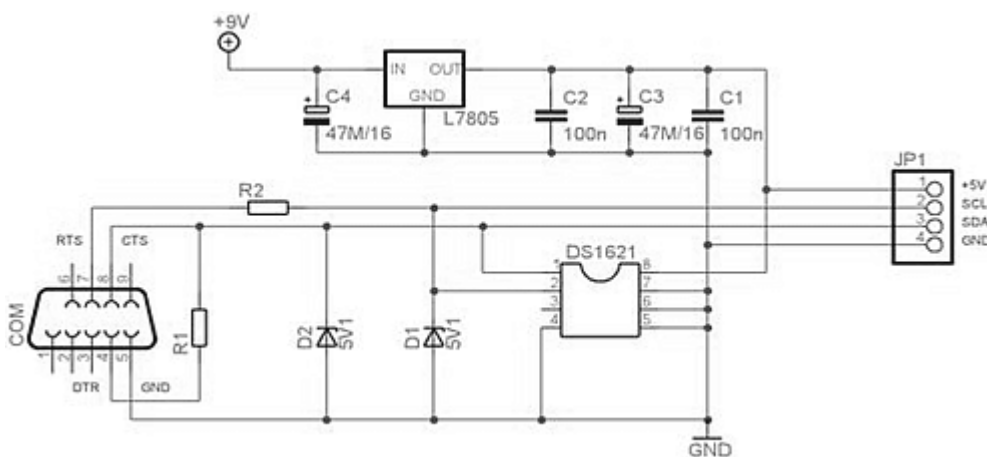
Měření teploty nevyžaduje žádná externí čidla, rozlišení teploměru je od -55°C až 125°C po 0,5°C, Teplota je dána jako 9-bitové slovo (2-bytový přenos), široký rozsah napájecího napětí 2,7V až 5,5V, převod teploty do digitální podoby do 1s, pouzdro DIP8, nastavení termostatu jsou uživatelsky definované a proměnné, data jsou přenášena přes dvou vodičové sériové vedení, spotřeba 1.25mA.



Obrázek č.23 - Blokové schéma DS 1621

1.34.2 Schéma zapojení a požadované součástky

Schéma zapojení je jednoduché a požadovaných součástek není mnoho. Vystačíme si s několika diodami, rezistory, kondenzátory a v neposlední řadě budeme potřebovat konektory na připojení k PC.



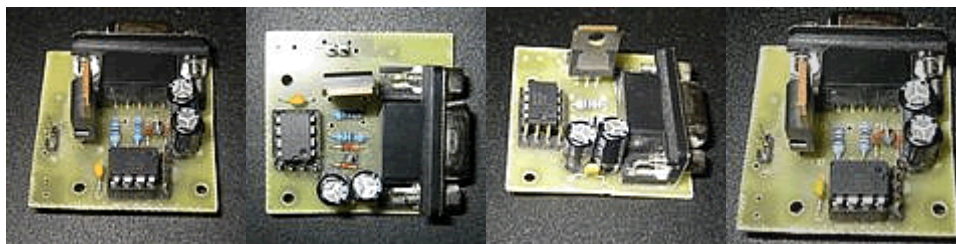
Obrázek č.24 – elektrické schéma zapojení teploměru

Sběrnice -[I2C Teploměr]-

Počet kusů	Součástka	velikost
4	Krytka konektromi CANON9	
2	Konektor samice CANON9	
1	Konektor samec CANON9	
1	Čtyř žilový kabel	
2	Elektrolitický kondenzátor	4,7KΩ/16V
2	Rezistor	4,7KΩ
2	Zenerova dioda	5,1V
1	IO DS 1621	
2	Kondenzátor	100nF
1	pinová řada	3 piny
1	Baterie 9V	9 - 12V
1	Patice DIP 8	

Tabulka č. 25 – Rozpis jednotlivých součástek

Tyto součástky jsou pouze pro hlavní desku s jedním teploměrem. Mohli by jsme připojit ještě dalších osm dodatečných čidel, paralelně k našemu hlavnímu čidlu.



Obrázek č.26 – osazený tištěný spoj

Zde je možno vidět náhled už hotového teploměru i s vývody pro další externí teplotní čidlo, který komunikuje po I2C sběrnici přes rozhraní RS232.

5. 1-Wire (Dallas Semiconductor Corporation)

1-Wire® síť je levná síť, ve které počítače komunikují digitálně, přes kabely například z kroucené dvoulinky s užitím 1-wire komponentů. (teplotní čidlo)
Rozhraní bylo vyvinuto zhruba v 90. letech firmou Dallas Semikonduktor a bylo navrženo pro použití ve třech základních oblastech. Jedná se o zařízení s jedním datovým vodičem.

- Přístroje ve speciálních pouzdrech MC pro řešení problémů identifikace, přenosu dat nebo jejich zpracování.
- Programování vnitřních pamětí obvodů s rozhraním 1-wire
- Systémy automatizace (1-wire sítě).

První oblast pro identifikační účely je již známa a dlouhodobě používána. Druhá oblast programování vnitřních pamětí obvodů s rozhraním 1-wire zajišťuje snadné změny funkce jednotlivých součástek a obvodů, které vyrábí firma Dallas-Maxim
A třetí oblast ještě není dostatečně oceněná z důvodu jejího dalšího rozvoje.

Z počátku nebylo společností dodáváno příliš mnoho součástek a použití bylo velmi omezené, nicméně tato situace na trhu se velmi rychle mění. V poslední době se objevují mnohá nová zařízení, součástky a použití sběrnic 1-wire v různých oblastech. Stále více vývojářů se zaměřuje na tuto technologii, což je především umožněno stále rostoucí nabídkou součástek pro 1-wire síť.

Na stránkách www.maxim-ic.com je možnost si objednat libovolný druh a počet součástek, jak na firmu tak i jako fyzická osoba. Ceny jsou uváděny v dolarech a ke každé součástce je poskytován obsáhlý manuál s parametry, ve kterém si může každý člověk najít přesně to, co potřebuje. Datasheet je samozřejmě v anglickém jazyce. Na této stránce je k dispozici všechno důležité, od různých kontaktů, přes informace o společnosti, až po záložku Whats new, kde je možné se dočíst všechny novinky o součástkách, s odkazem přímo na ní a s platným datumem.

1.35 Produkt Tree

Ukázka stromu produktů, které společnost Dallas Corporation nabízí. Strom se skládá z 1-wire produktů a Ibutton zařízení.



1.36 Temperature sensor

Jednou z položek jsou také tepelná čidla, které jsou nejčastějším zařízením 1-wire sítě.

Part Number & Description <input type="button" value="Compare"/> <input type="button" value="Clear All"/>	Accuracy (±°C)	Parasite Power	Temperature Threshold	Temperature Resolution (Bits)	Multi-Droppable	Min. Foot-print (mm ²)	RoHS Available	Price*
<input type="checkbox"/> DS28EA00 1-Wire Digital Thermometer with Sequence Detect and PIO	0.5	Yes	Programmable (NV)	9 10 11 12	Yes	16	Yes	\$2.25 @ 1k
<input type="checkbox"/> DS1825 Programmable 1-Wire Digital Thermometer with 4-bit Address	0.5	Yes	Programmable (NV)	9 10 11 12	-	16	Yes	\$1.70 @ 1k

Výhodnost tohoto síťového standardu můžeme vidět v tom, že v něm – li v úvahu analýzy, které vypovídají, že více jak 60% objektů vyžadují automatizaci a když nebudou potřebovat větší přenosovou rychlost jak 16 kbit/s. Pak můžeme využívat následující standardy 1-wire jako:

- Jednoduché řešení adresovatelnosti účastníků
- Jednoduchý protokol
- Jednoduchá struktura spojení
- Malé množství potřebných komponent
- Snadná změna konfigurace sítě
- Značný dosah sítě
- Nízká cena technologie

Uvnitř každé součástky, nebo zařízení opatřeného síťovou verzí protokolu 1-wire, je uložena jeho unikátní neměnná adresa (To, že se daná adresa nebude opakovat v žádném jiném zařízení zaručuje firma Dallas-Maxim.) má taková síť prakticky neohraničený adresový prostor. Tedy my můžeme připojit na 1-wire takřka libovolný počet zařízení. Přitom je každá součástka 1-wire ihned připravena v síti 1-wire a to bez podpůrných programových modifikací. Přenos dat v síti 1-wire je asynchronní a poloduplexní. Veškeré informace obíhající v síti jsou přijímány jako příkazy, nebo jako data. Příkazy jsou generovány řídicím členem (Master) sítě, provádějí se různé varianty hledání a adresace známých zařízení, určují adresaci na síti a řídí přenos dat v síti.

1.37 Rychlost přenosu dat po 1-wire

Standardní rychlost práce 1-wire sítě je 16 kbit/s, byla vybrána jednak pro zajištění maximální spolehlivosti přenosu dat na velké vzdálenosti a z důvodu přizpůsobení rychlosti k nejrozšířenějšímu typu mikropočítačů, které se musí bezpodmínečně používat pro komunikaci se zařízením 1-wire sběrnice. Rychlost 16 kbit/s se ale může zmenšovat nebo zvětšovat, z důvodu zpomalení jednotlivých bitů dat. Je také možné po 1-wire síti komunikovat rychlostí 128 kbit/s tak zvaný

overdrive, který je možno připustit na nevelké vzdálenosti a s kvalitními komponenty linky, která není nikterak přetěžována.

1.38 TTL signál a napájení

Ve 1-wire síti se používá klasická úroveň TTL signálu. Napájení komponent až už teplotních čidel, nebo různých převodníků může být z vnějšího zdroje s pracovním napětím 2,8 – 6 V nebo jako tak zvaná parazitní vazba, která využívá signálu z výstupu konektoru PC, který je samozřejmě nutno oříznout zenerovými diodami z úrovně +12V na požadovanou úroveň.

1.39 Přenosová media a topologie

Základem 1-wire sítě jsou 3 vodiče. Data - datový vodič, Ret – zpětný, nebo zemnicí vodič, Ext Power – externí napájení nejen řízených slave obvodů, ale i dalších senzorů a systémů řízení.

Použité kabely, délka linky a počet zařízení spolu ovlivňují dosažitelné parametry sítě. Při utváření složitých 1-wire sítí musíme brát v úvahu možné snížení dosahu sítě. Tyto vlivy musíme omezit hlavně vhodnou volbou topologie. Je možno větvení sítě do různých úrovní buď adresovatelným obvodem DS2406, který zabezpečuje větvení sítě komutováním zpětného vodiče, nebo specializovaným obvodem DS2409 pro přímou komutaci sběrnice dat. Tato varianta se zdá být vhodnější, protože komponenty na odepnuté větvi zůstávají v aktivním stavu. To zlepšuje a umožňuje prodloužit délku společné sběrnice a zvětšit počet zařízení na ní.

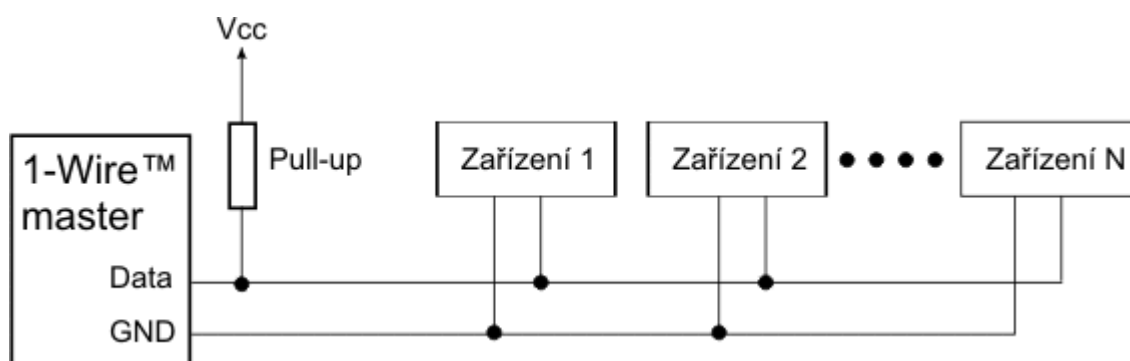
1.40 Realizace 1-wire systému na osobním pc

Pokud by docházelo ke komplikacím, je doporučováno použití inteligentního adaptéru pro sériový port typu Link. Ten používá mikro počítač. Poté je možné pokračovat v práci s obvodem i za specifického rušení. Link nám taky zlepšuje „aktivní přidržení na lince“, což nám umožňuje udržet kvalitní signál přenosu i při

délkách kabelu nad 300 metrů a také zvýšit počet zařízení na síti až na 100. Používá algoritmus pro číslicovou filtraci, tím výrazně snižuje poruchovost 1-wire sběrnice.

1.41 Počet zařízení na 1-wire sběrnici

Každé zařízení, u kterého to dává smysl (převodníky, teploměry...) má v sobě paměť ROM, která si pomatuje unikátní 64 bitové číslo, podle kterého je možno jednotlivá zařízení od sebe na síti rozlišit. Toto číslo je složeno z prvních 8 bitů, které jsou uvedeny v manuálu, sériového čísla 48 bitů a z CRC kódu nejvyšších 8 bitů. Tímto číslem je zařízení jednoznačně identifikováno. Při komunikaci s jednotlivým zařízením je třeba vyslat RESET puls, poté vyslat příkaz Match ROM a pak 64 bitový kód zařízení, se kterým se má komunikovat.



Obrázek č.27 – blokové schéma znázorňující připojení více 1-wire prvků na sběrnici

1.42 Příkazy Search a Alarm Search

Vešina 1-wire zařízení implementuje příkaz SEARCH (kód 0xF0). Zařízení která mohou být i v poplachovém stavu implementují příkaz ALARM SEARCH (kód 0xEC) na tento příkaz reagují jen ty zařízení, která se nachází v poplachovém stavu. (například teplotní čidlo, u něhož sme nastavili hranici teploty programu)

Jde jen o to zjistit které zařízení má být rychleji obslouženo.

První zařízení, které přijme příkaz SEARCH odpoví tím, že odešle první nejnižší bit svého 64 bitového kódu. Pokud je na sběrnici více takových zařízení odpoví všechny najednou. Podle obrázku č.27 zařízení jsou připojena paralelně ke

společnému vodiči s otevřeným kolektorem. Výsledkem je logický součin and všech přijatých bitů. Po vyslání těchto bitů požádá Master o vyslání dalšího pořadového bitu. Zařízení odpoví negací prve vyslaného bitu. Z těchto dvou přijatých bitů můžou na sběrnici logicky nastat čtyři různé možnosti.

Bit 1	Bit 2	Situace
0	0	Na sběrnici je více zařízení. Tato zařízení mají na tomto místě ve svých kódech různé hodnoty. Došlo k neshodě.
0	1	Na sběrnici je jedno či více zařízení. Tato zařízení mají na tomto místě ve svých kódech bit s hodnotou 0.
1	0	Na sběrnici je jedno či více zařízení. Tato zařízení mají na tomto místě ve svých kódech bit s hodnotou 1.
1	1	Na sběrnici není žádné zařízení, které by reagovalo na příkaz SEARCH.

Tabulka č.28 – jednotlivé možnosti stavů na sběrnici

Master nyní vyšle jeden potvrzovací bit. Dále se budou vyhledávání účastnit jen ty zařízení, které mají na prvním místě stejnou hodnotu jakou vyslal master. Pokud mají všechna zařízení na dané pozici stejnou hodnotu bitu, vyšle master tuto hodnotu. Nastala první možnost, oba bity byly nulové. Master si musí poznamenat na které pozici došlo k neshodě a vyšle buď „0“ nebo „1“, což určí na základě předchozích hledání. Dále vysílá nejprve nulu a po dalším průchodu vysílá jedničku.

Tento postup se opakuje dokud není načteno všech 64 Bitu identifikace. Pokud při načítání došlo k neshodě, posílá se na dané pozici bit opačné hodnoty. Bylo detekováno více zařízení a musí se projít druhá větev.

1.43 Algoritmus hledání

Algoritmus je poněkud složitější, než popis příkazů Search a Alarm Search. Proto se o něm zmíním jen stručně. Podrobně je popsán v Application Note 187: 1-Wire search algorithm. Procházení sběrnice se skládá z opakovaných hledání jednotlivých zařízení. V každém cyklu je nalezeno jedno zařízení. V algoritmu je to trochu jiné. Je potřeba si udržovat informaci o kódu posledního nalezeného zařízení a o pozici neshody (jestliže nějaká neshoda nastala, jestliže ne je zařízení na síti samo).

Algoritmus pracuje s následujícími proměnnými.

- **Id_bit_number** – Pořadové číslo bitu, který je právě prohledáván (1-64).
- **Id_bit** – Bit přečtený jako první. Jeho hodnota je logický součin všech přijatých bitů na pozici id_bit_number.
- **Cmo_id_bit** – Bit přečtený jako druhý doplněk id_bit.
- **LastDiscrepancy** – Hodnota udávající pořadové číslo bitu, u kterého došlo k neshodě.
- **LastDeviceFlag** – Poslední nalezené zařízení.
- **Last_zero** – Pořadové číslo bitu, u kterého došlo k neshodě, které vrátilo odpověď „0“.
- **Rom_no** – Buffer o velikosti 64 bitů, který obsahuje aktuálně načtené identifikační číslo zařízení.
- **Search_direction** – Bitová proměnná, udává jak bude probíhat další hledání zařízení, které má hodnotu na pozici Id_bit_number. Tatáž hodnota se bude účastnit dalšího vyhledávání, ostatní se účastnit nebudou.

Před prvním hledáním (funkce **First**) je proměnná LastDiscrepancy vynulovaná a po prvním hledací rutině je do Rom_no načten první bit, pokud se nic nenačte je hledání ukončeno s chybou.

Hledání dalšího zařízení (funkce **Next**) využívá hodnoty z prvního hledání, která je načtená do funkce Rom_no a LastDiscrepancy. Poté bude vráceno v Rom_no číslo dalšího zařízení, nebo jestli že byl předcházející byt poslední je hledání ukončeno.

Hledání jednotlivých zařízení musí začínat tím že master vyšle RESET puls a čeká na potvrzení PRESENCE pulsem. Pokud je tohle splněno, může hledání pokračovat a nastaví se hodnoty do proměnné Id_bit_number na hodnotu „1“ a last zero na hodnotu „0“. Následuje příkaz SEARCH nebo ALARM SEARCH. Následující smyčka pro hodnotu Rom_no se opakuje 64krát, dokud není načten každý jednotlivý bit zařízení.

Ve smyčce jsou přečteny jako první dva bity (hodnota bitu a její doplněk) pokud jsou od sebe oba přijaté bity různé, je hodnota prvního načtena do Rom_no na hodnotu danou id_bit_number a je také zapsána do search_direction. Pokud jsou obě hodnoty „1“ znamená to, že žádné zařízení neodpovídá na hledání, funkce končí. Pokud jsou oba bity „0“ je třeba se rozhodnout, jestli bude vyslána „0“ nebo „1“ Rozhodování probíhá asi takto.

LastDiscrepancy a id_bit_number	Reakce
Id_bit_number = Last Discrepancy	Last_zero = id_bit_number Search_direction = 0
Id_bit_number > Last Discrepancy	Search_direction = 1
Id_bit_number < Last Discrepancy	Search_direction je nastavena na hodnotu bitu na stejné pozici v Rom_no.

Tabulka č.29 – rozhodování na základě proměnných LastDiscrepancy a Id_bit_number

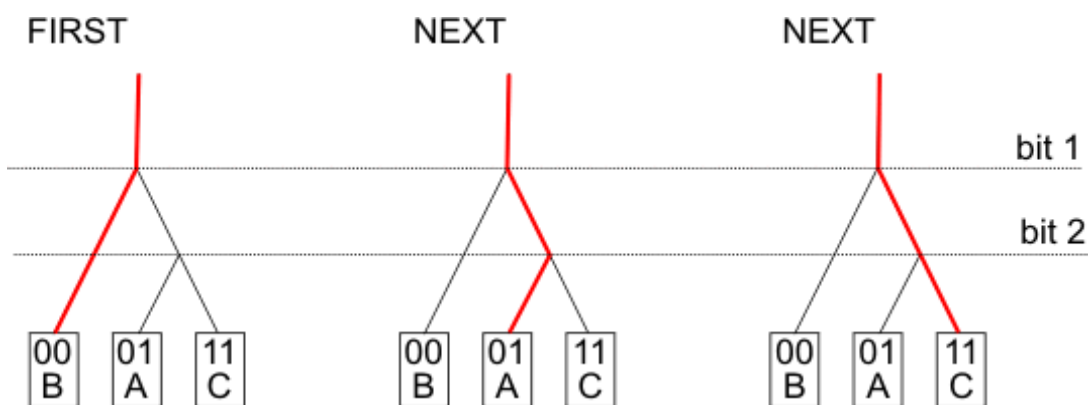
Hned poté je vyslán příkaz Search_direction a jeho hodnota je zapsána na další pozici do bufferu Rom_no. Počítadlo id_bit_nubmer je navýšeno o bit. Smyčka se stále opakuje.

Po zjištění všech bitů je do proměnné LAsDipcrepancy zapsána hodnota Last_zero. V momentě kdy bude hodnota nulová je načtené zařízení poslední.

1.44 Grafické Znázornění práce algoritmu

Znázornění hledání třech zařízení s dvoubitovými kódy.

- Zařízení „A“ kód „01“
- Zařízení „B“ kód „00“
- Zařízení „C“ kód „11“



Obrázek č.30 – grafické znázornění vyhledávání třech zařízení s 2 bitovými kódy

1.45 Reset puls

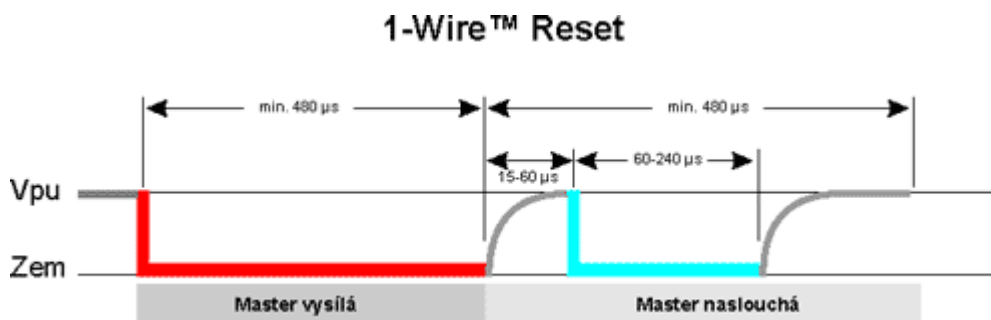
Sběrnice navržená firmou Dallas semiconductor, jak už bylo řečeno umožňuje připojit jedno či více zařízení na síť k řídicí jednotce za předpokladu dvou vodičů.

Tato technologie je využita i u technologie iButon, která je použita v elektronických systémech jako jsou třeba docházkové systémy.

Sběrnice se skládá z jednoho ovládacího obvodu (Master) a aspoň jednoho, ale je možno i více ovládaných obvodů (Slave). Všechny obvody jsou zapojeny na společnou zem a paralelně na datový vodič. Datový vodič je připojen přes nějaký reuistor na Ucc (napájecí napětí) a zdvihá tak sběrnici z „0“ do logické „1“.

Komunikace se zahajuje vždy tak, že master vyšle Reset puls. To znamená, že se musí nejprve stáhnout datový vodič do logické „0“ (uzemní ho) a drží ho na této úrovni alespoň 480 mikrosekund. Poté je potřeba sběrnici uvolnit a naslouchat jí.

Odpor zatím vrátí sběrnici do logické „1“, pokud je samozřejmě na sběrnici připojené nějaké 1-wire zařízení. Detekujeme tuto vzestupnou hranu a po prodlevě (15 – 60 mikrosekund) stáhne sběrnici na (60 – 240 mikrosekund) k logické „0“. Následující text je možno vyjádřit tímto Grafickým zpracováním.



Obrázek č.31 – grafické znázornění Reset pulsu

Reset puls se dá jednodušeji popsat následujícím algoritmem.

- Data out (480 mikrosekund) log „0“
- Data out (60 mikrosekund) log „1“
- Naslouchat
- Data in (250 mikrosekund) cca 50 měření dostaneme sled jedniček a nul (...111100000001111...), důležité pro nás je třeba dejme tomu 15-35 měření, kde dostaneme sérii nul mezi dvěma vzestupnými hranami. 1-wire zařízení je detekováno.

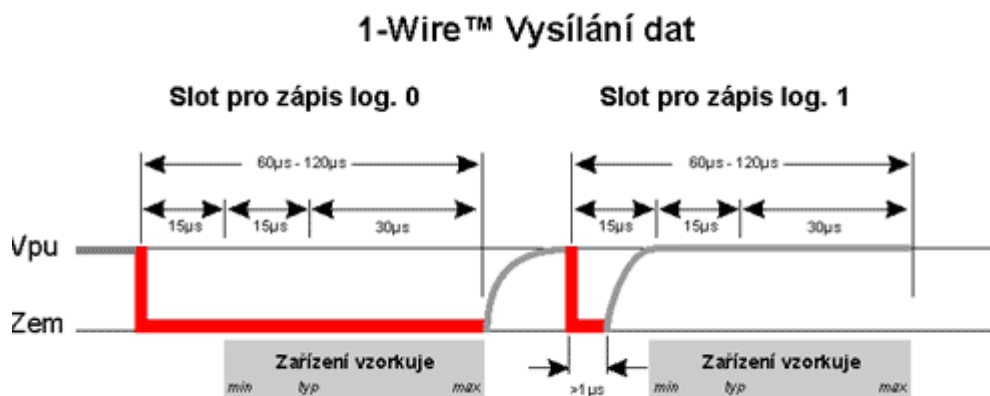
1.46 Vysílání dat

Jestliže správně detekujeme zařízení a správně se nám ohlásí, můžeme začít vysílat a přijímat data. Data jsou vysílána a přijímána v tak zvaných Time slotech, dá se říci v časových intervalech. Časový interval je dlouhý 60-120 mikrosekund a během každého tohoto intervalu je přijat, nebo vyslán jeden bit informace. Mezi sloty musí být nějaká časová mezera (alespoň 1 mikrosekunda), kdy je sběrnice v klidu. Existují 4 druhy vysílání dat:

Zápis log „0“, Zápis log „1“, Čtení log „0“, Čtení log „1“

1.47 Zápis Log „0“ a Log „1“

Vysílání zápisu slouží k tomu, aby master vyslal data do zařízení, které potřebujeme. Zápis log „1“ funguje tak, že master stáhne sběrnici k nule minimálně opět na 1 mikrosekundu a nejpozději do 15 mikrosekund od začátku ji opět uvolní a ponechá uvolněnou. Zdvihací odpor ji opět vytáhne k log „1“. Zápis log „0“ je o dost jednodušší. Jde jenom o to aby master stáhnul sběrnici do log „0“ a nechá ji tam po celou dobu, tedy 60 – 120 mikrosekund. Zařízení naslouchá (vzorkuje po 30 mikrosekundách) po začátku Time slotu.



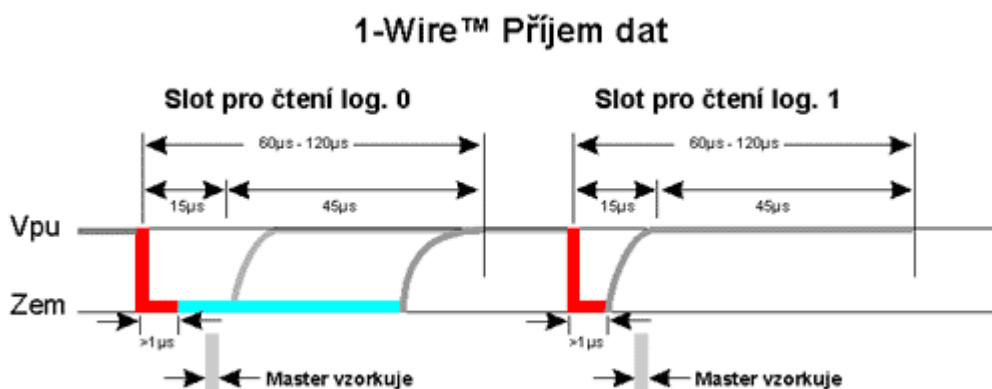
Obrázek č.32 – grafické znázornění vysílání dat

Jednodušší popsání zápisu Log „1“ a Log „0“ následujícím algoritmem.

- Logická „0“
- Data out (120 mikrosekund) log „0“
- Data in (30 mikrosekund) vzorek 4x
- Data out (60 mikrosekund) log „1“
- Logická „1“
- Data out (15 mikrosekund) log „0“
- Data out (120 mikrosekund) log „1“
- Data in (30 mikrosekund) vzorkuje 4x

1.48 Čtení log „1“ a Log „0“

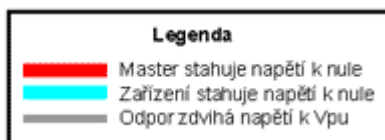
Čtecí sloty inicializuje opět master. Stáhne sběrnici na log „0“ minimálně opět na 1 mikrosekundu a uvolní jí. Po tomhle může zařízení buď vyslat jeden bit buď tím, že ponechá sběrnici v klidu (log „1“), nebo že ji stáhne (log „0“).



Obrázek č.33 – grafické znázornění příjmu dat

Znázornění čtení Log „0“ a Log „1“ následujícím algoritmem.

- Logická „0“
- Data out (15 mikrosekund) log „0“
- Data in (30 mikrosekund) vzorkuje 4x (doba 120 mikrosekund)
- Logická „1“
- Data out (15 mikrosekund) log „0“
- Data out (120 mikrosekund) log „1“
- Data in (30 mikrosekund) vzorkuje 4x



1.49 Jednoduchý Teploměr pro PC (1-wire)

Zkusil jsem vyrobit jednoduchý teploměr, který by měřil okolní teplotu nějakým teplotním čidlem, kterých je na stránkách Dallas Corporation více než dost. Ze začátku jsem nevěděl kde přesně začít. Teplotní čidlo mohlo být připojeno k PC přes LPT1 paralelní port nebo Com1 sériový port popřípadě mohlo být bráno v úvahu USB. Vzhledem k tomu, že mám novější PC tak už paralelní ani sériovou linku na výstupu nemám z toho důvodu se mi nejvíc líbila možnost USB. Po dnu hledání na internetu jsem zjistil že připojit 1-wire k USBčku nebude zrovna snadná věc, nebo jsem aspoň nikde nenašel nějaké kloudné schéma zapojení. Rozhodl jsem se, že to vyřeším jinak. Zjistil jsem, že je běžně k dostání redukce USB => COM i s CD nosičem plným ovladačů a různých utilit, které nám zajišťují a konvertují komunikaci ze sériového portu tak aby tomu USB rozumělo.

Pro samotné měření teploty jsem si vybral čidlo s označením DS 18B20 od výše zmíněné firmy Dallas – Maxim. Patří z hlediska parametrů mezi ty nejlepší a je cenově velmi dostupné pro každého cca 70 Kč. Je velmi přesné, absolutní přesnost je lepší než 0,5 °C. Lze jej připojit pouze dvěma dráty jak už jsme u 1-wiru zvyklí a obsahuje spoustu dalších funkcí, které patrně ani nevyužijeme (např programovatelnou přesnost převodníku 9-12 bitů.)

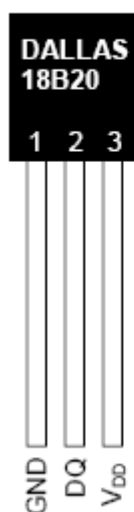
KEY Specifikation			Temperature senzor DS 18B20				
Part number	Interface	Accuracy	Supply voltage range	Parasite power	Operatin temp. Range	Bit Res	Temp. treshold
DS 18B20	1-Wire	0,5 °C	3.0 to 5.5 V	Available	-55 to +155 °C	9-12	1, prog NV

Obrázek č.34 – bližší specifikace dalasovského čidla DS 18B20

Teplotní čidlo je konstruováno do napětí od 3 – 5,5 V což bylo nutné vyřešit protože napěťová úroveň, kterou můžeme dostat ze sériového portu je 12V. Pokud by jsme nechtěli řešit externí napájení, musíme použít tak zvanou parazitní vazbu, kterou nám toto čidlo umožňuje. Docela elegantní mi přišlo řešení dle následujícího schématu, které má následující výhody.

- Moderní konstrukce
- Jednoduchost
- Digitální čidlo, není nutná kalibrace
- Sběrnicové řešení, lze propojit více čidel za sebou
- Ucelený soubor periférií

PIN ASSIGNMENT



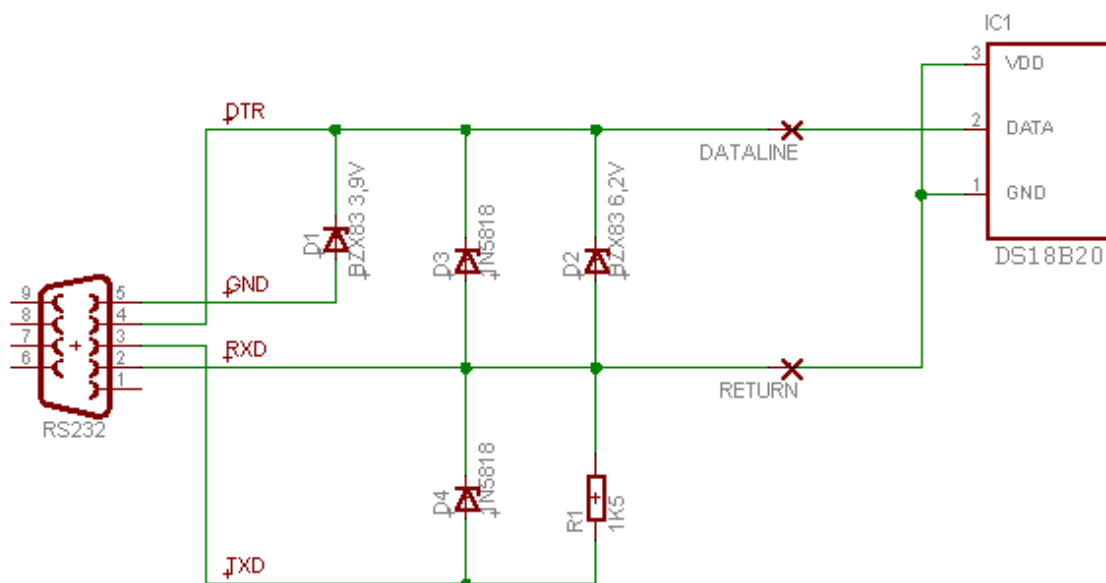
(BOTTOM VIEW)

TO-92
(DS18B20)

PIN DESCRIPTION

GND - Ground
DQ - Data In/Out
V_{DD} - Power Supply Voltage
NC - No Connect

Obrázek č.35 – čidlo DALLAS DS 18B20 a jeho popis vývodů

1.49.1 Schéma zapojení teploměru

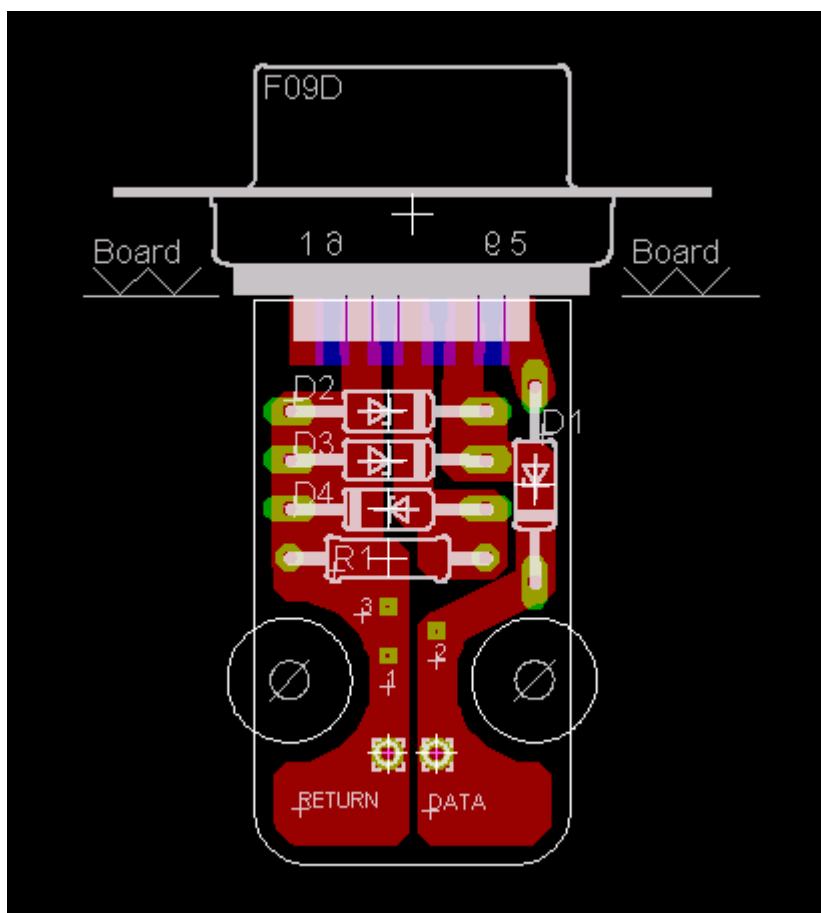
Obrázek č.36 – elektrické schéma zapojení teploměru

Cena celého celku není nikterak veliká. Dá se pořídit do celkových nákladů kolem sta korun. Použité součástky jsou následující.

- Zenerova dioda BZX83 3,9V
- Zeyerova dioda BZX83 6,2V
- Dioda 1N5818 2x
- Rezistor 1K5
- Teplotní čidlo 1-wire DS18B20

Vzhledem k zjednodušení práce jsem použil k zapojení kontaktní pole, do kterého stačí součástky podle daného schématu pouze nasázet, bez potřeby pájení a leptání tištěného spoje.

Na internetu je ke stažení i už předem navržený tištěný obvod který je implementován přímo do konektoru RS232.



Obrázek č.37 – navržený tištěný spoj vsazený přímo do konektoru RS 232

Podle následujícího schématu můžeme součástky napájet přímo do obalu konektoru RS232. Čidlo pak vypadá vskutku originálně neobtěžuje svou velikostí je kompaktní a skladné. Nemusíme se obávat poškození nebo ukroucení kablíků nebo součástek.



Obrázek č. 38 – osazené tištěné spoje vsazené do konektoru RS 232

1.49.2 Software

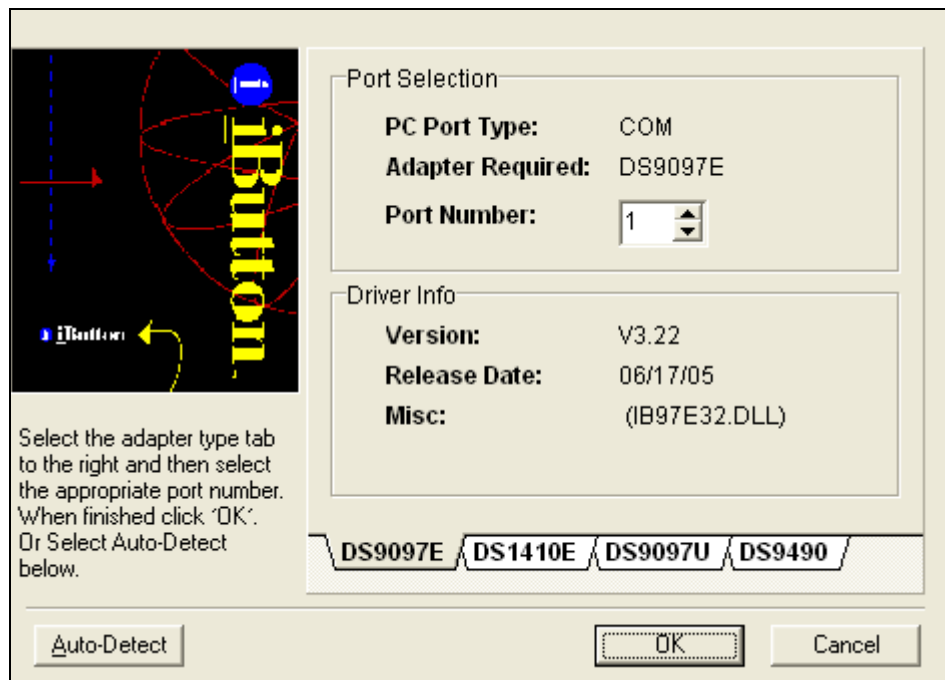
Pro ověření funkčnosti a spolehlivosti čidla jsem se rozhodl použít program LogTemp, který nám bude na první pokusy a ověření funkčnosti stačit. Program funguje pod Windows 95, 98, Windows XP SP2, naopak nefunguje z nějakého mě neznámého důvodu pod OS Windows 2000 SP4. Program je pěkně přehledný a na měření teploty zcela dostačující.

Musím ale dodat, že tenhle program není řešení nějaké složitější realizace řízení. Přepočítává nám sice řetězec ve dvojkové soustavě, který nám posílá čidlo na stupně °C, F a tak dále. Umí kreslit a následně vynášet grafy a spoustu dalších věcí, nicméně pro řízení je nevhodný z důvodu toho, že neumí zapisovat na zařízení 1-wire impulsy, které potřebujeme a převod řetězce na °C, F není zrovna v případě řízení vhodný, protože my potřebujeme s tímto řetězcem pracovat, přeposílat ho dále a manipulovat s ním. Proto zdůrazňuji, že se hodí pouze na měření teploty a ověřování funkčnosti čidel. Pro složitější operace je vhodnější napsat vlastní software.

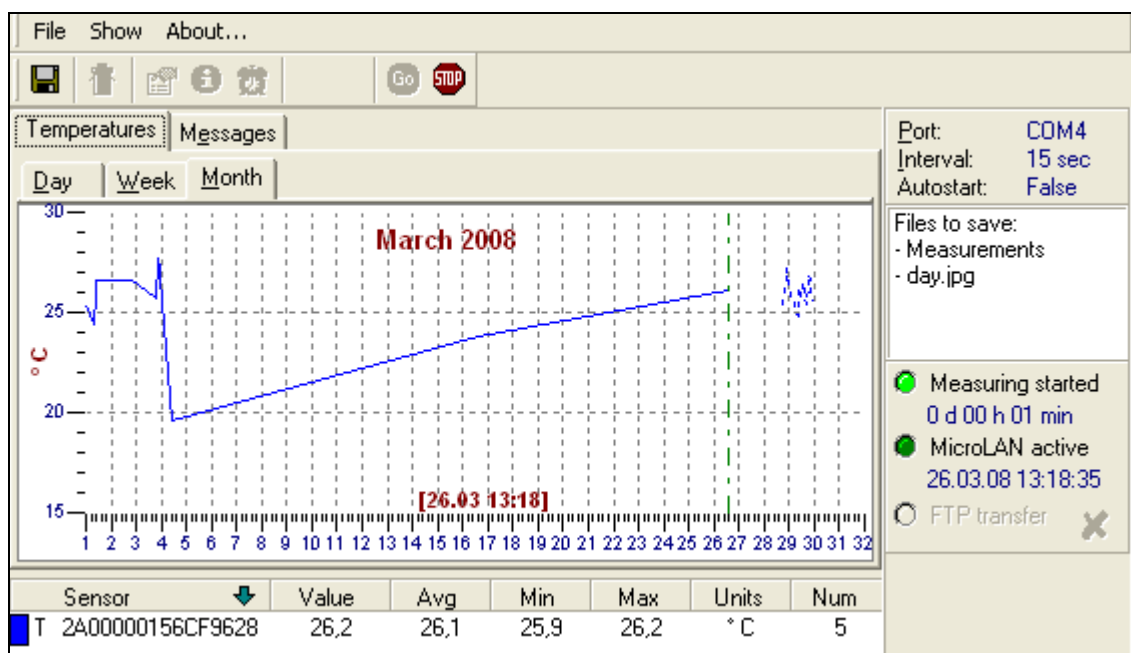
Program také potřebuje mít nainstalovaný program iButton – TMEX and Developers Tool Kit, který nám rozpozná, kam máme dané zařízení připojeno. (USB, LPT, COM) Bez tohoto programu nám LogTemp nebude fungovat. V našem případě se jedná o adaptér DS9097E, COM1. Popřípadě můžete použít tlačítko autodetect, které adaptér samo inicializuje.

Ukázka programu iButton obrázek č.39. Nastavování adaptéru a Autodetekce (COM, USB, LPT) Porty jsou označeny čísla DS909E, DS1410E, DS9097U, DS9490 a ukázka programu LogTemp obrázek č.40. Nastavení portu, intervalu času snímání teploty, vykreslování grafu, a měření průměrné, maximální, minimální teploty.

Sběrnice -[1-Wire Teploměr]-



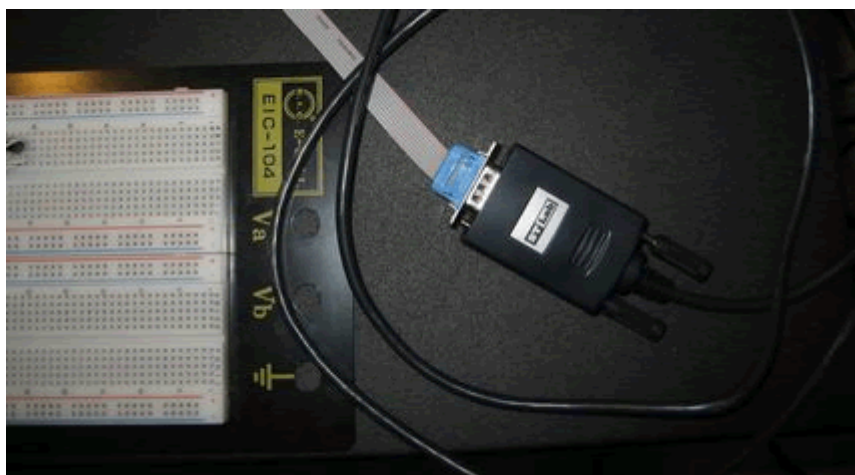
Obrázek č.39 – program iButton



Obrázek č.40 – program LogTemp

1.49.3 USB redukce a PC karty

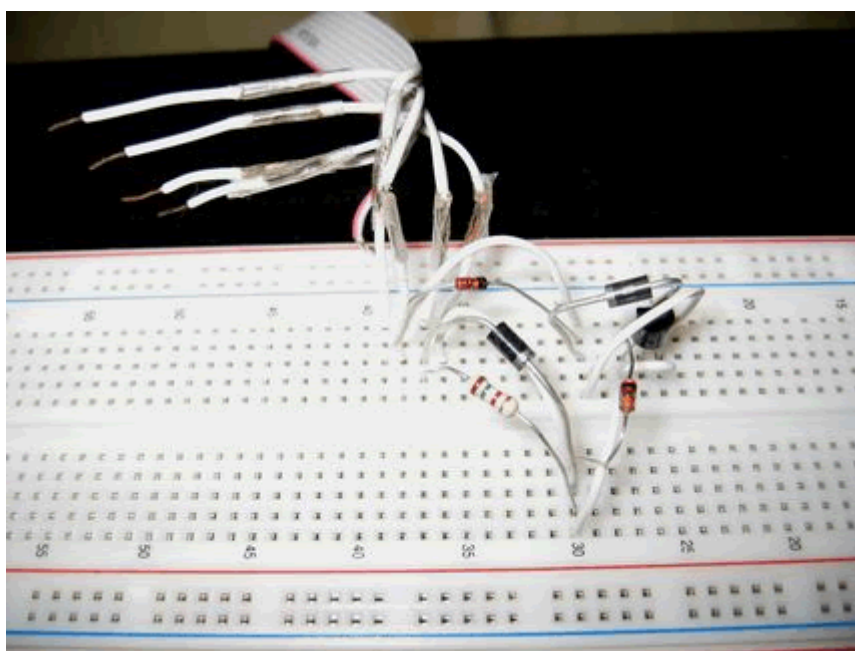
Mezi počítačem a konektorem RS232 používáme redukci USB to RS232 pokud nemáme přímo sériovou linku implementovanou na PC. Přes USB redukci se dá připojit takřka veškeré zařízení od myši až po tiskárny. Je to teda dobrá volba v případě, že potřebujeme řešit nějaký problém přes port, který už v počítači nemáme. Není však dobré se domnívat, že přes redukci bude fungovat úplně všechno. Mohou se objevit nějaká zařízení, nebo software se kterým si redukce nebude umět poradit, proto je to spíše levnější varianta řešení, nicméně ne nejlepší. Dají se sehnat PC karty do isa slotu, které nám plnohodnotně nasimulují COM port se vším všudy a je zajištěna stoprocentní spolehlivost a funkčnost, samozřejmě je to dražší alternativa.



Obrázek č.41 – Redukce USB => RS232

1.49.4 Vlastní realizace

Realizace zapojení byla provedena na dotykovém poli, z důvodu snadného manipulování s jednotlivými součástkami a jednoduché modifikace zapojení. Jediný problém se kterým jsem se setkal, byl že plochý kabel složený z devíti žil který vede do konektoru RS232 neznačí piny od červené žíly postupně za sebou ,ale musíme rozdělit žíly na liché a sudé. Potom liché žíly značí piny 1-5 postupně za sebou od červené a sudé žíly značí piny 6-9 postupně za sebou.



Obrázek č.42 – Osazené dotykové pole součástkami

Teploměr měřil spolehlivě a přesně. Čidlo reaguje na změnu teploty rychle s přesností na 0,5°C což se dá říci že je slušné. Jak je z obrázku vidět používáme jen piny 2-5 (čtyři piny). Vlastní komunikace probíhá však po dvou drátech. Pin číslo 4 Dataline (posílané data) a pin číslo 2 return (čtená data).

6. Stručný popis programovacího jazyka Visual Basic 6.0

Visual Basic a jeho odnože se v posledních letech staly hlavním programovacím prostředkem na platformě produktů firmy Microsoft. Různé mutace tohoto programovacího jazyka se používají nejen pro programování samostatných (i síťových) aplikací, ale také pro tvorbu maker v balíku programů Microsoft Office (Visual Basic for Aplikations) a při programování internetových aplikací, spouštěných na straně serveru (ASP) i klienta (Visual Basic Skript).

K oblibě tohoto programovacího jazyka přispěla hlavně jednoduchá syntaxe, snadná tvorba uživatelského rozhraní a velká variabilita a flexibilita jazyka. Ve Visual Basicu se dají programovat jak kancelářské, tak i složité aplikace. Programování ve Visual Basicu je počítáno mezi objektově orientované a událostmi řízené techniky. V krátkosti řečeno to znamená toto: Programátor může používat velké množství předdefinovaných objektů jako jsou formuláře, textová pole pro zadávání a zobrazování dat, příkazová tlačítka, menu, popisky, a velké množství dalších objektů. Souhrnně tyto objekty nazýváme ovládacími prvky (Controls).

Každý ovládací prvek má definovány své vlastnosti, metody a události. Co jsou a k čemu slouží? Vlastnostmi prvku (Properties) jsou přesně v souladu s názvem vlastnosti daného ovládacího prvku. Tyto vlastnosti udávají vzhled a chování ovládacího prvku v aplikaci. U textového pole a jeho umístění na formuláři, název, pomocí něhož se na ovládací prvek bude odkazovat programový kód a mnoho dalších vlastností. Vlastnostmi mohou být vzhled (Apperenance), chování (Behaviour), vazba na databázová data (Data) a další. Vlastnosti ovládacích prvků však lze rozdělit také podle jejich dostupnosti. Hodnoty některých vlastností je možné nastavit pouze jednou při zakreslování (Design) ovládacího prvku na formulář, ale nelze je již měnit programovým kódem. Příkladem takové vlastnosti je Aligment (zarovnání). Jiné vlastnosti je možné disonovat pouze za běhu aplikace (Run time) programovým kódem, avšak při zakreslování jsou nedostupné. Drtivou většinu hodnot vlastností všech pvládacích prvků je však možné definovat jak při zakreslování a základní definici prvku, programovým kódem.

Metody ovládacích prvků vlastně představují činnost, které daný prvek může vykonávat, případně, které mohou být vykonány na něm. Pokud se podržíme příkladu textového pole, najdeme u něj definované například metody Refresh (obnovit zobrazovaná data) nebo SetFocus (umístění fokusu, to je zaměření na textová pole).

Události definované u všech ovládacích prvků zajišťují, že programování ve Visual Basicu se počítá mezi událostmi řízené programovací techniky. U každého ovládacího prvku najdeme seznam událostí, které mohou při běhu programované aplikace vzniknout v přímém vztahu k tomuto prvku. Klasickými událostmi, které najdeme u většiny ovládacích prvků, jsou Click (Klepnutí myši na ovládací prvek), DblClick (poklepání myši na ovládací prvek), GotFocus a LostFocus (události nabití a pozbytí Focusu, to je zaměření) a mnoho dalších. Události slouží k programování procedur, které se vykonají jako odezva na výskyt určité události. Například procedura, která bude v programovém kódu přiřazena k události Click na příkazové tlačítko, se vykoná vždy, když uživatel na toto tlačítko klikne myší.

Začátečník, který začíná s Visual Basicem by se měl v první řadě seznámit s prostředky a možnostmi, které Visual Basic nabízí. Visual Basic poskytuje celou řadu implementovaných komponent, kterých může, nebo i v některých případech musí uživatel využívat. Komponenty jsou části pomocných programů, které nám v mnoha případech urychlí a usnadní práci. Pro ovládání sériového portu se například používá komponenta MsComm, která nám umožňuje manipulování s RS232 i když s jistými omezeními. Proto je dobré před použitím jednotlivých komponent zjistit, do jaké míry nám daná komponenta pomůže, jak pracuje a co umožňuje.

Uvedené charakteristiky Visual Basicu způsobují, že programování je v mnoha ohledech velice jednoduché a že aplikace vytvořené ve Visual Basicu se zcela shodují s prostředím operačního systému Windows. Tak se každý uživatel, který pracuje ve Windows, velice rychle naučí pracovat s novým programem, jehož ovládání se mu zdá intuitivní a podobné tomu, na co je zvykl

7. Závěr

Jako praktickou část mé bakalářské práce jsem měl za úkol vyvinout komunikační software v programovacím jazyku Visual Basic 6.0, který je vhodný pro programování pod operačním systémem microsoft Windows. Programy mají komunikovat se sběrnici I2C a 1-Wire přes uživatelský počítač po rozhraní RS232. Nicméně jako sekundární produkt mé bakalářské práce bylo vytvoření dvou jednoduchých zařízení, které budou po sběrnici komunikovat s PC, aby bylo možno si ověřit funkčnost programů. V mém případě se jednalo o teplotní čidla DS1621 pro komunikaci po sběrnici I2C a DS18B20 pro komunikaci po sběrnici 1-Wire. Návody na vytvoření teploměrů s popisky elektrickými schématy a potřebnými součástkami jsou popsány v mé bakalářské práci. Vytvořené programy jsou přidány na kompaktní disk k mé bakalářské práci. Při vyvíjení programů jsem se leckdy ubíral špatným směrem, tak mi vznikly v průběhu času programy, které nefungují, nebo jejich funkčnost je omezena, než jsem se dostal ke správnému cíli. Při studii odborné literatury a při tvorbě aplikace jsem si prohloubil znalosti o programování ve Visual Basicu 6.0 a také teoretické znalosti, které se týkají dané problematiky. Práce byla zábavná a poučná, nicméně musím říci, že když jsem se mnohokrát po několikahodinové práci nesešel s úspěchem, tak jsem to mohl nazvat poněkud zneklidujícím.

Použité prameny a literatura

Literatura:

1. MORKES, D. Visual Basic 6.0 Computer Press
ISBN 80-7226-312-9
2. Microsoft Visual Basic 6.0. Compiuter Press
ISBN 80-7226-154-1

Internet:

3. <http://www.pcsvet.cz/vbasic/>
4. <http://www.hw.cz/>
5. <http://www.maxim-ic.com/>
6. <http://www.esacademy.com/>
7. <http://www.nxp.com/>