

Prostředky průmyslové automatizace

Garant předmětu:

Doc. Ing. František Zezulka, CSc.

Autoři textu:

Doc. Ing. František Zezulka, CSc.

Ing. Petr Fiedler

Ing. Zdenek Bradáč

Obsah

1. Úvod	8
2. Přehled automatizačních prostředků	8
2.1 Procesní instrumentace	8
2.2 Řídící členy	9
2.3 Komunikační podsystemy	9
2.4 Nadřazené řízení	9
3. Procesní instrumentace	9
3.1 Čidla, snímače, převodníky	9
3.1.1 Přehled čidel, převodníků a přístrojů pro sběr informace z procesu.	11
3.1.2 Tabulka čidel teploty	12
3.1.3 Šumová imunita	13
3.2 Akční členy, aktory	14
3.2.1 Elektrické pohony	15
3.3 Průmyslové regulátory	17
3.4 Čítače, časovače	18
4. Programovatelné automaty	18
4.1 Charakteristiky PLC	20
4.1.1 HW PLC	20
5. Průmyslová PC v řízení	22
5.1 Příklad použití IPC	23
6. Distribuované systémy pro řízení (DCS/PCS)	24
6.1 Úroveň procesní instrumentace:	26
6.2 Úroveň bezprostředního řízení:	26
6.3 Operátorská úroveň:	27
6.4 Nejvyšší úroveň řízení:	30
6.5 DCS nové generace	30
7. Systémy reálného času v průmyslové automatizaci	35
7.1 Úvod	35
7.2 Systémy průmyslové automatizace	36
7.3 Historický přehled	39

7.4	Zaklady koncepce systému realneho casu.....	40
7.5	Inzenyrsky navrh „tvrдых“ systému realneho casu (hard real-time systems)	41
7.6	Operacni systemy realneho casu RTOS	42
7.6.1	Úvod	42
7.6.2	Pozadavky na vcasnost	46
7.6.3	Pozadavky na soucasnost	47
7.6.4	Typicke vlastnosti operacnich systemu realneho casu (RTOS)	47
7.6.5	Sprava pameti	48
7.6.6	Sprava pristroju a zarizeni	48
7.6.7	Principy organizace zpracovani uloh.....	49
7.6.8	Multitasking.....	51
7.6.9	Synchronizace a komunikace	55
7.6.10	Problematika synchronizace.....	57
8.	Komunikační systémy pro účely automatizace	58
8.1	Úvod do komunikačních systémů pro účely automatizace	58
8.2	Problematika digitální komunikace.....	62
8.3	Vrstvy ISO/OSI modelu	64
8.3.1	Fyzická vrstva.....	64
8.3.2	Linková vrstva	64
8.3.3	Síťová vrstva.....	66
8.3.4	Transportní vrstva.....	66
8.3.5	Relační vrstva	67
8.3.6	Presentační vrstva.....	67
8.3.7	Aplikační vrstva.....	67
9.	Perspektivy stávajících standardů fieldbusů a průmyslového Ethernetu	68
9.1	Ethernet	68
9.2	Průmyslové sítě a Ethernet.....	68
9.2.1	Komunikační protokoly průmyslového Ethernetu	70
9.2.2	Technologie známé z Internetu	70
9.2.3	MODBUS® TCP.....	70
9.2.4	EtherNet/IP	71
9.2.5	FOUNDATION Fieldbus	72
9.2.6	PROFINET	72
9.2.7	Další firemní standardy	73

9.3 Kombinovaná struktura	73
9.4 Ethernet z hardwarového hlediska	74
9.4.1 Nejobvyklejší varianty Ethernetu	74
9.4.2 Hardware realizující ethernetové rozhraní 10/100Base-TX	75
9.4.3 Volba procesoru	75
9.4.4 Zhodnocení perspektiv	75
10. Průmyslové komunikační sítě	76
10.1 Fyzický přenos	76
10.1.1 Výměna dat mezi jednotlivými účastníky	78
10.1.1.1 Synchronní a asynchronní přenos dat	79
10.1.2 Přístupové metody v lokálních sítích	80
10.2 Protokoly vyšších vrstev	85
11. Přehled průmyslových komunikačních sběrnic	87
11.1 Foundation Fieldbus	88
11.2 Úvod do dalších průmyslových komunikačních systémů	117
11.2.1 Profibus	117
11.2.2 FIP	123
11.2.3 P-Net	124
11.2.4 Protokol CAN (Controler Area Network) [14]	126
11.2.4.1 Úvod	126
11.2.4.2 Základní vlastnosti	126
11.2.4.3 Fyzické médium a fyzická vrstva	126
11.2.4.4 Linková vrstva CAN	128
11.2.4.5 Signalizace chyb	129
11.2.4.6 Základní typy zpráv	130
11.2.4.7 Nabídka elektronických součástek s protokolem CAN [14]	133
11.2.5 Protokol DeviceNet [28], [29]	134
11.2.5.1 Objektově orientovaný model	135
11.2.5.2 Aplikační vrstva protokolu DeviceNet	136
11.2.5.3 Detekce duplicity	137
11.2.5.4 Mechanismus komunikace v síti DeviceNet	137
11.2.5.5 Navázání spojení v síti DeviceNet	137
11.2.5.6 Podpora tvorby hierarchických komunikačních struktur	138
11.2.5.7 Profily v síti DeviceNet	138

11.2.5.8 Fyzická vrstva v síti DeviceNet.....	139
12. Totálně distribuované automatizační systémy	140
12.1 Technologie LonWorks.....	141
12.2 Základní vlastnosti protokolu LonTalk.....	141
13. Průmyslové sítě na nejnižší úrovni sběru dat	146
13.1 Síť Actuator Sensor Interface (AS-Interface, AS-I)	146
13.1.1 Hlavní rysy původní specifikace AS-i:.....	147
13.1.2 Nové vlastnosti zavedené specifikací AS-Interface verze 2.1.....	147
13.1.3 Slave sítě AS-Interface	148
13.1.4 Master sítě AS-Interface.....	148
13.1.5 Napájecí zdroj sítě AS-Interface	149
13.1.6 Kabeláž sítě AS-Interface.....	150
13.1.7 Spolehlivost sítě AS-Interface.....	151
13.1.8 Kdy AS-I není vhodná síť	152
13.2 HART (Highway Addressable Remote Transducer)	152
13.2.1 Jak HART pracuje	152
13.2.2 Komunikace s inteligentními čidly.....	152
13.2.3 Startovací sekvence	153
13.2.4 Adresace zařízení.....	153
13.2.4.1 Adresa mastera.....	154
13.2.4.2 Krátká adresa	154
13.2.4.3 Dlouhá adresa	154
13.2.5 Příkaz.....	154
13.2.6 Počet byte	154
13.2.7 Status	155
13.2.8 Data.....	155
13.2.9 Kontrolní součet	155
13.2.10 Burst mód	155
13.2.11 Fyzická vrstva protokolu HART	155
13.2.11.1 Kabeláž	155
13.2.11.2 Zařízení	156
13.2.12 Rychlost protokolu HART	156
13.2.13 Možné topologie HART sběrnice.....	156
14. Systémy Soft control a perspektivy automatizační techniky	158

14.1 Trendy vývoje automatizační techniky	159
15. Literatura	163

Seznam obrázků

Seznam tabulek

1. ÚVOD

Kurs Automatizační prostředky je určen všem studentům oboru KAM v 9. semestru jejich inženýrského studia na fakultě elektrotechniky a informatiky VUT v Brně. Kurs pod tímto názvem byl poprvé vypsán před 8 roky jako výběrový předmět v 9. semestru. Během doby doznal změn jak v obsahu, tak ve svém určení. V podobě výběrového předmětu se zabýval pouze partikulárními otázkami sériových komunikačních sběrnic pro účely automatizace. Právě pro své současné určení všem studentům oboru KAM musel jeho obsah doznat významných změn proto, že je vlastně poslední příležitostí Ústavu automatizace a měřicí techniky FEI VUT v Brně profilovat odborný názor absolventů oboru. Proto bylo nutné zahrnout do něj celé téma automatizačních prostředků, nejen významnou a moderní problematiku sériových sběrnic pro automatizaci. Třebaže autor skript vychází ze studijního plánu oboru, ze kterého plyne, že značná část studentů, kteří se musí podrobit studiu předmětu AUP již absolvovala kurzy, které se zabývaly čidly a akčními členy, mikropočítači a v určité míře navštěvovala i volitelný předmět Programovatelné automaty, je přesvědčen o tom, že každý absolvent oboru musí mít základy programovatelných automatů a představu úplného spektra automatizačních prostředků a problémů, které při realizaci projektu automatizačního systému souvisejí s pojmem automatizační prostředek. Nově tedy je jedna kapitola a laboratorní cvičení věnováno programovatelným automatům. Je samozřejmé, že tato skripta ve shodě s novým obsahem kursu AUP kladou důraz i na další moderní prostředky řízení a sběru dat jako jsou velké i PC orientované distribuované systémy pro řízení procesů DCS i totálně distribuované mikrořadičové systémy, propojené výkonnými komunikačními systémy. Skripta nemohou ignorovat ani nejnovější vývoj v tomto oboru a třebaže praxe zatím učinila jen první kroky ve směru využití nejrozšířenější sériové sběrnice Ethernet směrem k procesní automatizaci, je i tato problematika ve skriptech diskutována. Nezbytná pozornost je nově věnována i systémům nadřazeného řízení a vizualizace, které představují systémy SCADA, na které je kladen důraz především v laboratorních cvičeních tohoto kursu. Na druhé straně ve skriptech není pro jejich omezený rozsah zahrnuta příprava a návody do laboratorních cvičení z programovatelných automatů, ani PC orientovaných DCS systémů ani systémů SCADA.

2. PŘEHLED AUTOMATIZAČNÍCH PROSTŘEDKŮ

V následujícím stručném přehledu uvedme spektrum automatizačních prostředků, jak jsou chápány v tomto kursu.

2.1 Procesní instrumentace

- čidla, senzory, snímače
- akční členy

2.2 Řídicí členy

- průmyslové regulátory
- mikrořadiče
- PLC
- logika (Xillings, zákaznická logika, a další)
- velké systémy pro řízení procesů DCS/PCS
- totálně distribuované řízení
- IPC a soft control

2.3 Komunikační podsystemy

- paralelní sběrnice mikropočítačů
- paralelní sběrnice SCSI
- sériové sběrnice LAN
- fieldbusy
- malé sítě

2.4 Nadřazené řízení

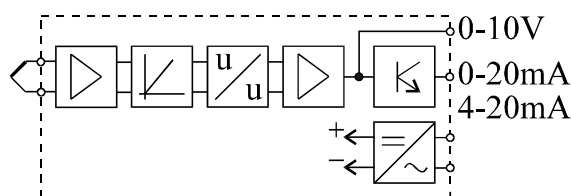
- operátorská pracoviště
- vizualizace a SCADA
- prostředky konfigurace a projektování
- MMI

3. PROCESNÍ INSTRUMENTACE

3.1 Čidla, snímače, převodníky

Procesní instrumentace představuje značnou část problematiky průmyslové automatizace a úspěch projektu z velké části závisí na výběru a umístění čidel, na jejich kvalitě a životnosti. Podobně výběr, kvalita a funkceschopnost akčních členů (aktorů) zaručuje do značné míry úspěch projektu řízení. Procesní instrumentace rovněž představuje významný finanční podíl v celém automatizačním projektu. V poslední době dochází k prudkému vývoji v oblasti senzorů

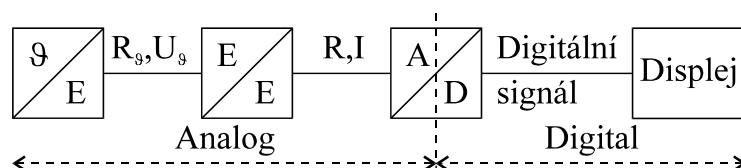
(čidel) pro měření neelektrických veličin. Tento vývoj je poznamenán vývojem mikroelektroniky. Stále více funkcí, souvisejících se zpracováním signálu (signal processing), se uskutečňuje přímo v čidlech, přesněji ve snímačích/ převodnících (transducers). Podobně akční členy (aktory) jsou stále více vybavovány elektronikou. Pro tyto snímače nové konstrukce, vybavené mikroelektronickými obvody se někdy používá názvu inteligentní čidla (smart sensors). Tento pojem není zatím normován a v důsledku toho se používá různě. Zjednodušeně řečeno, inteligentní sensor by měl integrovat kromě systému pro získání a zpracování měřenosné informace i elektroniku, umožňující komunikaci přes standardizované napěťové rozhraní a standardizovaným komunikačním protokolem po sériové dvou nebo 4 vodičové sběrnici s dalšími inteligentními čidly a nadřazeným řídicím členem. Kromě toho elektronika čidla musí umožnit i dálkové nastavení parametrů snímače, jeho diagnostiku a hlášení o stavu snímače nadřazenému členu. V neposlední řadě se od něj očekává i zpracování naměřené informace nejen ve smyslu následujícího obr.3.1, ale v mnoha případech včetně Fourierovy transformace a dalších procedur " signal processing". Stávající převodníky pro průmyslovou automatizaci a měření lze ve velké většině popsát zjednodušeně schématem z obr. 3.1.



Obr. 3.1: Blokové schéma klasického termoelektrického převodníku

Signál z termočlánku jde do převodníku, kde blok 1 je vstupní zesilovač, blok 2 provádí linearizaci, blok 3 galvanické oddělení, blok 4 zesílení a blok 5 převod napěťového signálu na proudový výstupní signál 0-20mA nebo 4-20mA. Takové čidlo se pak proudovou smyčkou propojí s řídicím nebo měřicím systémem, vybaveným proudovým vstupem. Informace se ve většině případů přenáší analogově s rozlišením 12 bitů na velké vzdálenosti, což zaručuje robustnost proudové smyčky. Přenosová rychlost je malá a pohybuje se do 4,8kbitů/s. Nevýhodou je jednak malá rychlost přenosu, jednak dvoubodový spoj. Každé čidlo v tomto provedení vyžaduje minimálně dva vodiče.

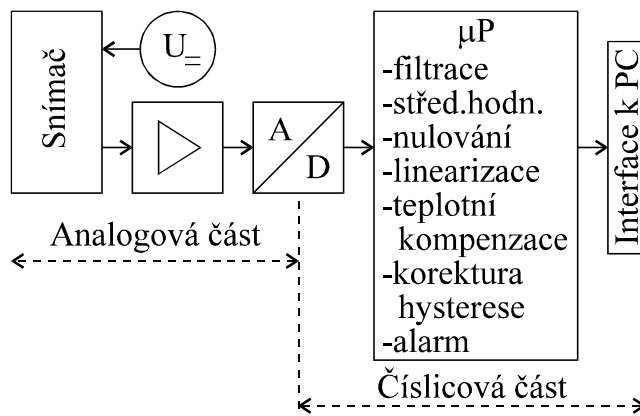
Převodník teploty (čidlo teploty) lze však najít i v digitální podobě. Blokové schéma digitálního čidla je na obr. 3.2.



Obr. 3.2: Blokové schéma digitálního čidla

Signál kupř. z termočlánku se převede na vhodnější elektrický signál v převodníku a v A/D převodníku se transformuje na digitální signál s danou rozlišovací schopností (12bitů kupř.). Tento digitální signál se přivádí na vstupy měřicího přístroje nebo řídicího členu. Od tohoto digitálního čidla je již jen krok k inteligentnímu čidlu. To musí umožňovat tvorbu zprávy v digitální podobě, zabezpečení a přístup čidla k sériovému spoji.

Blokové schéma inteligentního čidla je uvedeno na obr. 3.3. Tento snímač (transmitter) se opět člení na analogovou a digitální část, avšak digitální část je řešena mikrořadičem, který kromě digitalizace a přenosu digitální informace k dalšímu zpracování a vyhodnocení v měřicím přístroji nebo PC, sám provádí celou řadu funkcí vyhodnocení signálu, jak je uvedeno na obr. 3.3.



Obr. 3.3: Blokové schéma inteligentního snímače

Na blokových schématech obr.3.1, 3.2 i 3.3 je vlastní čidlo schematicky znázorněno v zapojení přímo na vstupy zesilovače analogového signálu. V reálných systémech se však v mnoha případech s výhodou používá můstkového zapojení nejrůznějšího provedení [4],[8] a další.

V dalším uvedeme stručný přehled čidel a akčních členů, které tvoří procesní instrumentaci.

3.1.1 Přehled čidel, převodníků a přístrojů pro sběr informace z procesu.

A. Měření neelektrických veličin

- síla
- hmotnost
- moment
- kroutící moment
- tlak, tlaková diference
- rychlost, úhlová rychlost
- zrychlení
- délka a úhel
- tloušťka povrchových vrstev
- vodivost
- množství
- teplota
- průtok

- pH
- redox potenciál
- vlhkost
- analýza plynů

B. Měření elektrických veličin

- napětí (ss, střídavé, vyšší frekvence, špička, minimum)
- proud (ss, střídavý, špička, minimum)
- výkon (okamžitý, průměrný, činný, jalový, jednofázový, třífázový)
- energie

Měření každé z výše uvedených veličin je samostatný problém, charakterizovaný fyzikálním principem, metodou, instrumentací, přesností, linearitou, frekvenčními vlastnostmi, pracovními rozsahy, použitými materiály a dalšími parametry. Všechny tyto fenomény jsou rozpracovány do detailů a jsou předmětem měřicí techniky. Zde je proto neuvádíme. Vzhledem k důležitosti problematiky měření, řízení a regulace teploty, uvedeme zde pouze přehled použití čidel teploty a rozsahy jejich použití.

3.1.2 Tabulka čidel teploty

Termočlánky	Teplota [°C]
J (Fe-CuNi)	-200 ÷ 650
K (NiCr-NiAl)	0 ÷ 1300
S (PtRh10/0)	0 ÷ 1300
B (PtRh30/6)	0 ÷ 1800

Odporové teploměry	Teplota [°C]
Pt	-250 ÷ 850
Termistor	-100 ÷ 200
Posistor	-20 ÷ 200
Křemík	-70 ÷ 140

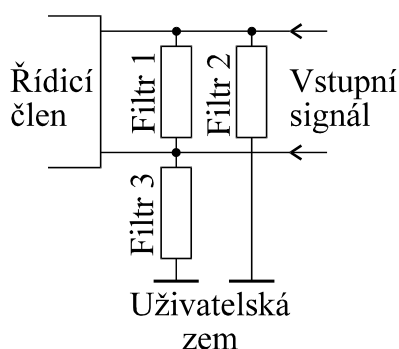
Deformační teploměry	Teplota [°C]
kapalinové-skleněné	-200 ÷ 630
kapalinové-deformační	-35 ÷ 500
bimetalové	-130 ÷ 450
tlakové	-270 ÷ 700

Pyrometrické	Teplota [°C]
spektrální	20 ÷ 3500
pásmové	-100 ÷ 2000
širokopásmové	-100 ÷ 2200

Speciální principy	Teplota [°C]
krystalové	- 80 ÷ 250
tekuté krystaly	do 300
optoelektrické luminiscenční	do 400

3.1.3 Šumová imunita

Důležitým problémem, který úzce souvisí s čidly, jejich připojením k řídicím systémům, úrovně rušení, spolehlivosti a funkceschopnosti je odolnost proti šumu - šumová imunita.



Obr. 3.4: Zvýšení odolnosti vstupu řídicího členu vůči šumu ze snímače (ze vstupu).

Zemění je jedním z nejlevnějších a přitom nejefektivnějších způsobů pro zvýšení odolnosti měřicí a řídicí techniky proti šumu. Je třeba dbát doporučení výrobce PLC a měřicích přístrojů, jak správně zemnit vstupy systému. Kromě toho je praktické zemnit všechny kovové skříně

přístrojů a držáky čidel vodiči s dostatečným průřezem. Nedoporučuje se připojovat na společnou zem s přístroji s nedostatečnou EMC. Další možností zlepšení šumové imunity je snížení vlastní hladiny šumu samotného zdroje šumu. Kupř. použití spínaného napájecího zdroje nebo výkonového zesilovače, který spíná při průchodu střídavého napětí nulou zvyšuje celkovou šumovou odolnost měřicího a řídicího systému.

Na obr.3.4 je způsob provedení filtrů na vstupu přístroje pro zvýšení odolnosti proti vstupnímu šumu. Filtry jsou typu RC,C nebo varistor. Každý filtr mezi signálovými vodiči zlepšuje diferenciální šumovou imunitu (filtr 1), zatímco filtr 2 a filtr 3 (z libovolného signálového vodiče na zem) bude zlepšovat celkovou šumovou imunitu (common-mode noise immunity).

3.2 Akční členy, aktory

Podobně širokou škálu jako čidla tvoří i aktory. V prostředí průmyslové automatizace se používají následující systémy, které lze rozdělit do tří kategorií dle pomocné energie a dále do kategorie přímých aktorů. Jedná se o aktory s pomocnou elektrickou, pneumatickou a hydraulickou energií. Celkový přehled aktorů je uveden v následující tabulce:

- elektrické motory
 - synchronní
 - asynchronní s frekvenčními měniči
 - stejnosměrné
 - krokové
 - lineární
 - elektromagnety
 - lineární posunutí
 - otočné, vibrační
 - termobimetaly
- SMA (shape memory alloy) aktory
- deformační elementy
- elektrochemické aktory
- piezoelektrické aktory
- magnetostrikční aktory
- mikroaktory
- hydraulické motory
 - lineární
 - rotační
- hydraulické ventily

- servoventily
- proporcionální
- pneumatické motory
- membránové aktory
- spínače, přepínače
 - elektromagnetické
 - polovodičové

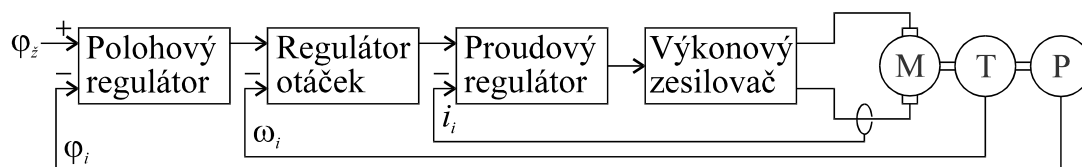
Vzhledem k tomu, že principy čidel a celá problematika snímačů a převodníků se probírá ve volitelných kursech tohoto oboru, ponechávám na studentech, aby si problematiku čidel nastudovali dle potřeby ze studijních materiálů uvedených kursů nebo z doporučené literatury [3], [4], [9].

Na druhé straně se aktorům nevěnuje na oboru zdaleka taková pozornost a proto se budeme v další kapitole uvedenou problematikou zabývat poněkud podrobněji.

3.2.1 Elektrické pohony

Pohon musí být schopen reagovat v daném čase silou nebo výkonem, odpovídajícím žádané hodnotě a pohybovat řízenou soustavou po dané dráze nebo ji uvést do daného bodu. Pohon, který využívá elektrickou energii jako pomocnou energii se označuje jako elektrický pohon. Použití elektrických pohonů v automatizační technice patří k nejběžnějším. Elektrický pohon se skládá z elektrického motoru, řídicího členu, snímačů polohy a rychlosti, převodu. Konstruuje se jako rychlostní nebo polohový. U pohonu pro řízení, nastavování a regulaci polohy se nepočítá s trvalým chodem elektrického motoru. Na druhé straně se na něj kladou zvýšené požadavky na rozsah otáček (1:10 000), velké zrychlení, rychlé brzdění, krátkou časovou

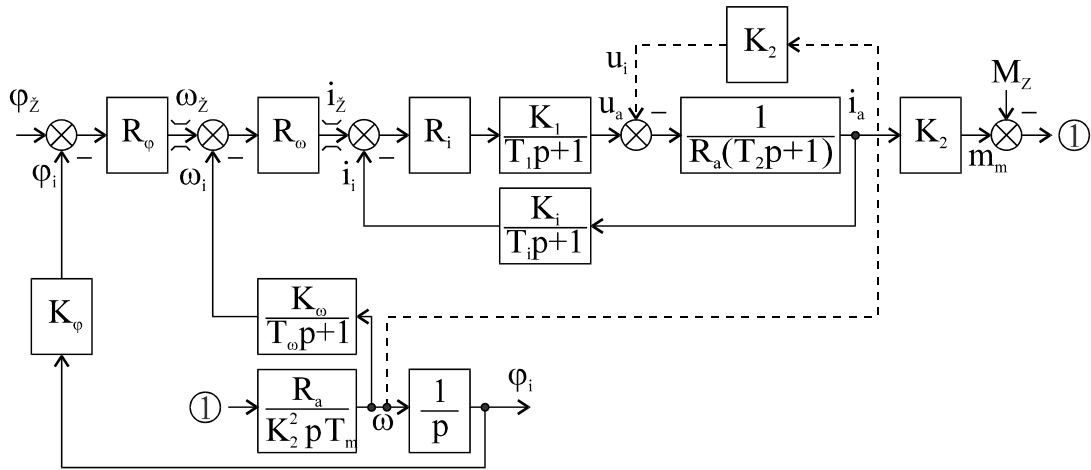
konstantu, velký klidový moment, robustnost, klidný a tichý chod a některé další. Některé z těchto požadavků si protířečí a proto výrobci hledají všeobecně přijatelná řešení.



Obr. 3.5: Instrumentální blokové schéma typického stejnosměrného pohonu.

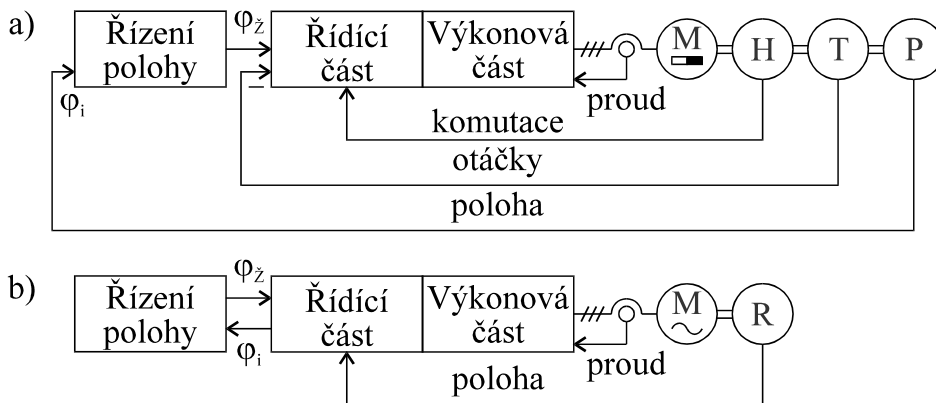
Stejnoseměrné pohony s komutátorovými motory se v automatizaci používají velmi dlouho. Stále jsou nejvíce využívány pro rozsahy výkonů pod 300W a nad 20KW. Jejich hlavní přednost je v nízké ceně a jednoduchém způsobu regulace. Pro své nevýhody, mezi které patří jiskření a opotřebování komutátorů, údržba a omezená dynamika, jsou nahrazovány bezkomutátorovými motory. Až do výkonů kolem 10kW jsou v provedení s permanentními magnety. Na obr. 3.5 je uvedeno instrumentální blokové schéma polohové regulace se stejnosměrným pohonem.

Na dalším obr.3.6 je uvedeno podrobné blokové schéma polohového servomechanismu z obr. 3.5.



Obr. 3.6: Blokové schéma polohového servomechanismu se stejnosměrným pohonem

Schéma představuje typické schéma rozvětvené regulace s dvěma pomocnými regulovanými veličinami (proud kotvou ss motoru a otáčky). Základní regulovanou veličinou je poloha osy motoru (resp. osy převodu). Regulace proudu je zároveň regulací momentu a spolu s regulací otáček slouží ke zvýšení stability a zlepšení dynamických vlastností polohového řízení a regulace.



Obr. 3.7: a) řízení obdélníkovými impulsy
 b) řízení sinusovými impulsy
 kde: H Hallové sondy, T Tachodynamo, P čidlo polohy, R resolver

Vedle těchto ss komutátorových motorů se v automatizaci používají i bezkomutátorové ss (brushless-DC) motory, které se někdy označují jako elektronické (Electronic Commutated) motory. Princip spočívá v tom, že rotor obsahuje magnety a cívky jsou umístěny ve statoru. V poslední době se provedení statoru ustálilo na třífázovém provedení. Na rozdíl od komutátorových ss motorů, jsou EC motory dále vybaveny senzory (Hallové sondy), umístěnými v rotoru. Tyto senzory polohy dávají řídicí elektronice signály, umožňující přepínání statorových vinutí tak, aby pro danou polohu rotoru motor vyvíjel maximální kroutící moment. Řídicí elektronika, polohová čidla a vlastní motor jsou zapojeny do zpětnovazební smyčky, která porovnává požadovanou a skutečnou hodnotu otáček rotoru a změnou frekvence přepínání

eliminuje odchylku v otáčkách. Tento mechanismus stačí jen na vlastní funkci EC motoru, avšak pro realizaci polohového serva s EC motorem je nutné použít ještě přesnější čidlo pro osu motoru, kupř. resolver. Prostřednictvím podřízené proudové smyčky se řídí šířka pulsů šířkové pulsní modulace tak, aby motor vyvíjel požadovaný kroutící moment. Jsou známy dva způsoby řízení EC motorů. Řízení obdélníkovými pulsy a řízení sinusovým průběhem. Provedení EC motorů je odlišné dle použitého principu řízení. Na obr.3.7 je blokové schéma obou způsobů řízení a příslušná instrumentace.

3.3 Průmyslové regulátory

Průmyslové regulátory se používají od samého počátku automatizace strojů, linek a technologických procesů. S příchodem řídicích počítačů a minipočítačů ztratily jen krátce na svém původním dominantním postavení. S příchodem mikroprocesorů se opět staly konkurenci schopnými větším řídicím systémům (PLC a DCS) vzhledem k tomu, že využívají stejně jako alternativní řídicí členy a systémy (PLC, DCS) nejmodernějších mikroelektronických systémů a součástí a nového SW. Mají navíc výhodu v kratším inovačním cyklu a jsou tedy lépe přizpůsobeny zrychlujícímu se tempu vývoje mikroelektroniky. Tyto regulátory lze charakterizovat následujícími vlastnostmi a parametry:

- základ průmyslové automatizace
- vhodné pro měření a řízení teploty, tlaku, průtoku, hladiny, pH, poměrů dvou a více veličin a dalších
- vybaveny několika analogovými vstupy, několika digitálními vstupy a výstupy a 1 - 2 analogovými výstupy
- zobrazují průběhy regulovaných veličin nejen digitálně, ale i názornými sloupcovými grafy (jeden pro ŽH, jeden pro RV)
- mnohafunkční přehledný čelní ovládací panel
- externí vstupy pro modifikaci parametrů regulátoru
- funkční bloky v SW přístroje
- dostatečnou paměť na procesní data
- umožňují realizovat jednu nebo dvě regulační smyčky, včetně rozvětvených regulací
- umožňují změny parametrů bez přerušení funkce
- poskytují ruční i automatické řízení
- obsahují zpravidla self-tuning režim
- jsou vybaveny standardizovaným rozhraním RS 422/485 pro propojení až 32 přístrojů do sítě
- programovatelné z čelního panelu, ručního programovacího přístroje nebo po síti
- mají přímý a invertovaný výstup
- množinu předdefinovaných PID algoritmů

- signalizaci překročení povolených limitů (alarmy)
- jsou vybaveny uživatelským ID kódem
- po zapnutí nabíhá vlastní diagnostika přístroje
- možnost začlenění do SCADA
- a některé další (implementované fuzzy algoritmy ap.)

Třebaže by se dal z takto velmi dobře vybavených výkonných průmyslových regulátorů realizovat i velký automatizační celek, jejich využití je spíše v menších realizacích při řízení a regulaci strojů a malých technologických procesů a to především s ohledem na velký objem práce při konfigurování tohoto heterogenního systému ve srovnání s PLC nebo DCS. V chemických procesech a v elektrárnách se však jako doplněk distribuovaných řídicích systémů nebo jako záloha za účelem zvýšení bezpečnosti provozu používají ve velkém počtu.

3.4 Čítače, časovače

V mnoha procesech a činnostech se i nadále bude vyžadovat nezávislé měření časových intervalů a definování reálného času. Za tím účelem se opět setkáme s elektronickými časovači v kompaktním provedení se širokou škálou funkcí jako časové spínače, intervalové spínače a vypínače, cyklické spínače a další.

Rovněž elektronické čítače nejsou alternativními automatizačními prostředky (PLC) zdaleka vytlačeny. Funkčnost elektronických "stand-alone" čítačů, podobně jako časovačů a nižší cena je rozhodující pro jejich použití.

4. PROGRAMOVATELNÉ AUTOMATY

Programovatelné automaty (Programmable Logic Controlers PLC) se staly nejvýznamnějším řídicím prostředkem pro řízení technologických procesů, výrobních linek a strojů již během první poloviny 80. let. Byly odezvou na vývoj mikroelektronické technologie, který umožnil vytlačit centralizované řízení, reprezentované řídicími počítači a minipočítači distribuovanou řídicí technikou. Tato technika (PLC) sice zůstala na dlouhou dobu pozadu v programátorském komfortu za řídicími počítači a minipočítači, na druhé straně vykazovala nesporné výhody. Mezi ty patří spolehlivost, snazší rozdělení řídicí struktury na samostatné celky s jasně definovatelnými rozhraními, vysoká spolehlivost, nižší náklady na kabeláž. Z toho plyne rychlejší uvedení do chodu, snazší údržba, jednodušší ladění programů, modulární výstavba a tím optimalizace ceny HW, vysoká stabilita jednoduchého operačního systému, nižší nároky na kvalifikaci projekčních a inženýrských pracovníků, celkově nižší náklady na realizaci projektu, uvedení do chodu a závěrečné fáze projektu.

Vzhledem k tomu, že PLC nenahradily jen řídicí počítače a minipočítače, ale i malou automatizaci, reprezentovanou průmyslovými regulátory, bezkontaktní logikou a reléovou logikou, bylo pochopitelné, že jedním z kategoriálních požadavků průmyslu (projektantů, elektroinženýrů a středních odborných pracovníků) byl především jednoduchý programovací jazyk, který by byl velmi podobný jazyku logických schémat, booleovským rovnicím, reléovým schémátům, assembleru. Díky těmto jednoduchým programovacím "jazykům" bylo poměrně

jednoduché klasickou techniku logického řízení nahradit programově orientovanými a tedy nesrovnatelně flexibilnějšími řídicími systémy - programovatelnými automaty. Programovatelný automat umožňuje logické rovnice naprogramovat, zatímco předcházející bezkontaktní nebo reléová logika (nebo v dnešní době programovatelná logická pole) řeší logické rovnice fyzickým propojením logických členů. Jakákoli změna logické struktury se snadno provede změnou programu programovatelného automatu, což je podstatně jednodušší, než přepojení reléového nebo logického schéma. Odhlédneme-li od počáteční nespolehlivosti prvních programovatelných automatů (způsobené především nespolehlivostí elektronických součástek), náhrada relé a bezkontaktní logiky programovatelnými automaty byla jednoduchá a úspěšná.

V případě náhrady řídicího počítače programovatelným automatem nebyla situace pro novou technologii zdaleka tak příznivá. Pokročilejší programovatelné automaty sice vykazovaly již dostatečnou spolehlivost a rovněž organizace projekčních prací a jejich realizace byly výrazným zjednodušením oproti centralizovanému návrhu, sériovém ladění jednotlivých úloh a uvádění složitého systému do chodu, na druhé straně programátorský komfort minipočítačů se programovým prostředím PLC nahradil v plné míře až s příchodem SCADA systémů.

Výhody a nevýhody programovatelných automatů:

A. Výhody :

- rychlé přeprogramování úlohy
- malá varieta náhradních dílů
- možnost vystavění velké hierarchické struktury dle potřeby
- flexibilita (naprojektování na míru)
- modularita (možnost rozšíření)
- hospodárnost (levné velmi malé a malé kompaktní automaty)
- vestavěná diagnostika vlastního PLC
- možnost tvorby diagnostiky vnější
- jednoduché programování
- možnost použití vyšších programovacích jazyků u nových automatů
- jednoduchý a tím spolehlivý OS reálného času
- velká nabídka kvalitních přístrojů různých výrobců

B. Nevýhody:

- nižší programátorský komfort než u minipočítačů
- vyšší cena než IPC ekvivalentního výkonu při nižším programátorském komfortu PLC
- menší flexibilita ve srovnání s IPC
- užití nedostatečně standardizovaných sériových komunikačních sběrnic pro propojení automatů do sítí
- nezbytnost hierarchické architektury při propojování do větších celků

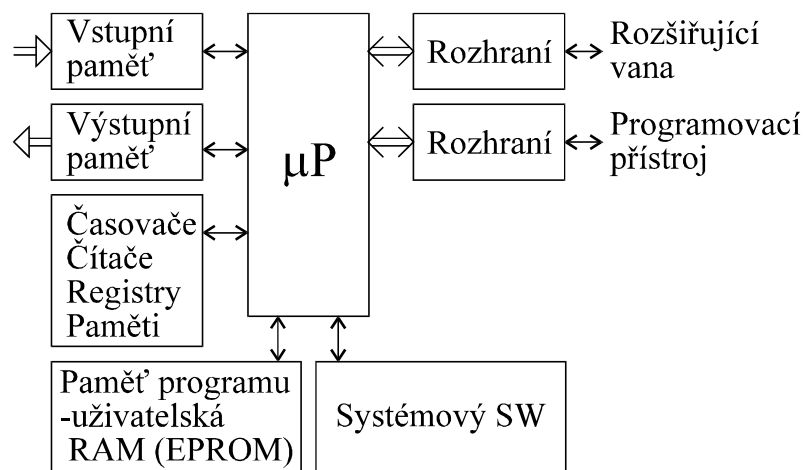
4.1 Charakteristiky PLC

4.1.1 HW PLC

V době svého vzniku (konec 60. let) si programovatelné automaty kladly za úkol nahradit efektivnějším způsobem reléovou a později i bezkontaktní logiku. Proto jejich architektura vycházela z toho, že budou zpracovávat binární informaci. Jako HW jádro používaly bitové procesory. To mělo za následek, že v době velmi pomalých procesorů s 8 nebo 16 bitovým slovem (v průběhu 70. let), se jevíly bitové procesory jako velmi rychlé, kvaziparalelní řešení ve srovnání s 8 a 16bitovými procesory. Proto se na architekturu PLC kladly následující nároky:

- bitově orientovaná CPU
- bitově orientovaná paměť dat
- slovně orientovaná paměť programu
- rozhraní na programovací přístroj
- jednoduchý instrukční soubor na zpracování logických rovnic
- systém speciálních funkcí (časovače, čítače a další)

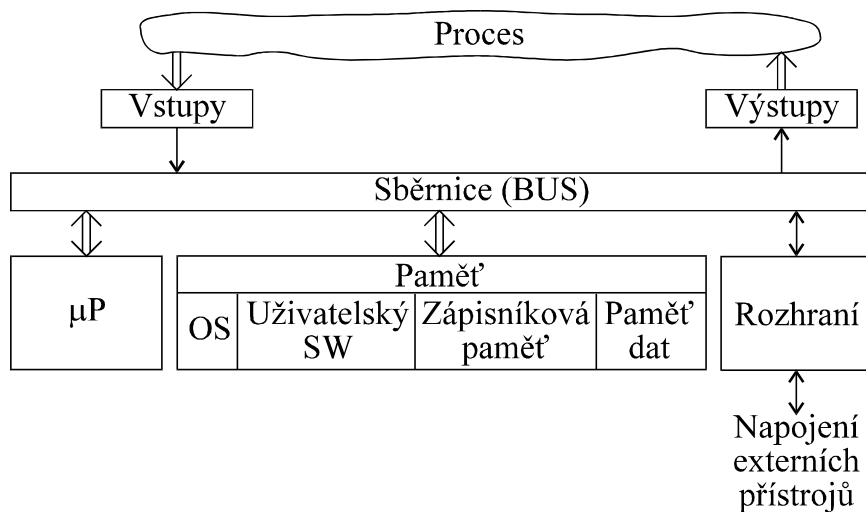
Takto zkonstruovaný PLC se do dnešní doby nezachoval. Rychlost a příznivá cena výkonných mikroprocesorů umožňuje použití slovně orientovaných mikroprocesorů i u velmi malých PLC. Přesto se blokové schéma velmi malých, kompaktních PLC liší od architektury středních a velkých automatů, jak je patrné z obr. 4.1 a obr. 4.2.



Obr. 4.1: Blokové schéma velmi malého PLC

Řízení logické úrovně je nemyslitelné bez toho, aby byly k dispozici v základním vybavení každého PLC časové funkce (časovače) a funkce čítání impulsů (čítače). Proto každý PLC má tyto dvě funkce v základním programovém vybavení.

Je patrné, že blokové schéma standardního modulárního PLC je velmi podobné na architekturu mikropočítače. Základ tvoří vnitřní 16 nebo 32 bitová sběrnice, kolem které je modulárně vytvořen celý PLC.



Obr. 4.2: Blokové schéma standardního modulárního PLC

Zatímco u prvních PLC s bitově orientovanou CPU byla paměť programu oddělena od paměti dat nebo naopak a pro data se používala i jiná (bitová) organizace paměti, dnešní PLC mají jednu operační paměť, ve které jsou vyhrazeny prostory pro vstupní data, výstupní data, vnitřní proměnné a paměťový prostor na vlastní program. Kromě toho jsou v paměti uloženy i funkční bloky a funkce jak systémové, tak vytvořené uživatelem. Operační systém PLC je nadále velmi jednoduchý, umožňuje režim reálného času a hraje významnou roli v konkurenceschopnosti PLC oproti IPC a dalším prostředkům průmyslové automatizace.

Způsob práce, který od počátku charakterizuje PLC a odlišuje je od řídicích mikropočítačů, t.j. cyklický způsob vykonávání programu zůstal základním režimem prakticky všech PLC. Tento základní režim práce PLC je ukázán na obr.4.3

Vedle cyklického režimu mají současné a to již malé až střední PLC i režim přerušování, který může být parametrován, takže časově kritické akce mohou být obslouženy mimo cyklus PLC.

5. PRŮMYSLOVÁ PC V ŘÍZENÍ

Jednou z moderních variant řídicích systémů je osobní počítač PC v průmyslovém provedení. Pro tyto systémy se vžil název IPC (Industry PC). Tato technika se začala na trhu objevovat v začátku 90. let. V té době jen několik evropských firem nabízelo tyto výrobky s nepříliš velkým úspěchem. Myšlenka byla přirozená a lákavá, dodat průmyslovému řízení podobný programátorský komfort, jaký poskytovaly řídicí počítače a minipočítače a jaké ve velké míře poskytují distribuované systémy pro řízení procesů (DCS). To však splňuje každé PC. Avšak pro řízení je nezbytně nutné, aby řídicí systém splňoval ještě celou řadu elektrotechnických norem a doporučení. Kupř. pro výrobce IPC v Německu to znamená splnit následující normy :

- DIN VDE 0113 a DIN VDE 0160 týkající se elektrotechnických výrobků pro průmyslové stroje
- VDI 2880 Blatt 1 bis 5 Programovatelné automaty
- DIN 19239 Programování PLC
- DIN IEC 65B část 1 až 5: Standard programovatelných automatů

V této normě jsou uvedeny požadavky na napájení, na velikosti vstupních signálů, potenciálové oddělení vstupů, velikost vstupního proudu, čas přepnutí a další. Podobně jsou udány parametry výstupních signálů:

- DIN IEC 721 klasifikace požadavků s ohledem na životní prostředí
- DIN IEC 68 požadavky na zkoušky (vibrace, rázy, transport, teplota, síla)
- DIN VDE 0843 díl 1 až 4 elektromagnetická kompatibilita pro měřicí, řídicí a regulační zařízení
- DIN VDE 0839 díl 10 bezpečnost proti poruše z elektrického vedení a výbojem (zde kupř. je jednoznačný požadavek na uvedení všech výstupů do definovaného - bezpečného stavu v případě výskytu poruchy)

Další požadavky jsou na provedení silové části řídicího systému jako na přívodní kabel, konektor a zásuvku a další, na signalizaci vstupů a výstupů na čelním panelu karet I/O.

Vybavit standardní PC i těmito vlastnostmi znamenalo výrazné zvýšení ceny oproti standardnímu (kancelářskému provedení PC), v průměru 2-3x. To bylo vážné omezení. Nejtvrdší podmínku, kterou IPC začátku 90. let a de facto až do rozšíření operačního systému Windows NT byl požadavek na stabilní multitaskový a multiuživatelský operační systém. Systém MS DOS byl pro tyto účely nevhodný, avšak objevovaly se jeho varianty, splňující tyto požadavky (M-DOS a další). Podobně se objevovaly zjednodušené varianty OS UNIX (Flexos 286 a další), které se skutečně v řízení na některých PC používaly.

Od poloviny 90. let, nabývá podíl IPC na trhu automatizace na významu a je jasné, že v jisté době budou vážným konkurentem PLC a další řídicí technice. Na vyšší úrovni řízení vytlačují výkonná PC postupně drahé pracovní stanice (Workstation), avšak i na úrovni bezprostředního řízení jejich význam roste. Již dnes je varianta řídicího systému na bázi IPC levnější, než ekvivalent v PLC provedení. Stále však PLC vykazují vyšší spolehlivost a tím i

bezpečnost, což jsou dva dnes nejsledovanější parametry řízení. Důvodem je stále vyšší spolehlivost jednoduchého, multitaskingového operačního systému reálného času, který mají PLC. Ani Windows NT, které jsou na IPC používány, nemohou být těmto systémům rovnocenným konkurentem ve stabilitě. Vývoj ovšem hovoří pro IPC a další podoby vyšších řídicích systémů a podíl PLC na řízení (zejména ve větších aplikacích, kde nedostatečný programátorský komfort prodlužuje fázi projektu řídicího systému) bude klesat.

V současné době řeší IPC úlohy řízení logické úrovně, přesněji úkoly bezprostředního řízení a regulace s převažujícím logickým řízením.

5.1 Příklad použití IPC

Vzhledem k flexibilitě IPC, nabízí se i možnost kombinování PLC funkcí a CNC funkcí v jednom nebo několika IPC. Takovým způsobem je pak možné projektovat kupř. pružné výrobní buňky (cells). Právě zde je možné hledat uplatnění IPC v největší a nejefektivnější míře.

Na další obr. 5.1c je znázorněno SW vybavení IPC pro účely řízení výrobní buňky. Vzhledem k velkému rušení jsou jednotlivé periferní moduly (umístěné přímo na strojích výrobní buňky) propojeny světlovodičem. Zde použitý [2] LightBus pracuje s rychlostí 2,5 Mbitů/s. Při použití skleněného vlákna je povolena vzdálenost mezi moduly 600m, pro plastický světlovodič to činí jen 45m. Pro takovou aplikaci je nezbytně nutné použít nějaký systém, integrující jak CNC řízení výrobních strojů, tak nezávislý řídicí systém kupř. robota, tak PLC řízení některých spolupracujících strojů (dopravníky, signalizace, řízení vrtání, stavění dopravních cest linky ap.). Průmyslové PC toto může integrovat v jeden systém snáze a s nižšími náklady, než by to bylo možné realizovat z HW nezávislých PLC a CNC. Předpokladem pro to je SW vybavení IPC, které obsahuje jednak:

- real-time kernel pro PLC a CNC programy
- multitasking a multiuser operační systém
- vyšší programovací jazyky pro nadřazené řízení
- programové prostředí pro PLC a CNC programování
- aplikační SW pro PLC a CNC řízení

Řídicí pracoviště dále musí umožnit připojení periférií pro rychlý vstup údajů pro výrobu (scanner) a výstup pro tisk protokolů ap. Rovněž může být vybaveno telefonním modemem pro případ hlášení poruch ap. Propojení s nadřazenou databází standardním Ethernetem je přirozeným vybavením každého PC.

Pro práci řídicího systému (mikropočítačového systému) v reálném čase jsou nezbytné tyto vlastnosti:

- hodiny reálného času
- přerušovací systém

6. DISTRIBUOVANÉ SYSTÉMY PRO ŘÍZENÍ (DCS/PCS)

Další významnou kategorií řídicích systémů jsou velké (nebo též kompaktní) systémy pro řízení procesů (Process Control Systems PCS). Jejich původ je třeba hledat v 60. letech s příchodem prvních řídicích počítačů. Řídicí počítače představovaly centralizované řešení číslicového řízení velkých celků jako elektráren, chemických procesů, železáren a válcoven, cementáren, farmaceutických a dalších provozů. Byly velkým technickým pokrokem v procesu řízení a byly charakteristické velkým výpočetním výkonem a programátorským komfortem, umožňujícím snadnou implementaci složitých řídicích algoritmů. Na druhé straně také nevýhodami, mezi které lze počítat vysokou cenu, nedostatečnou spolehlivost, složitý způsob ladění programů, dlouhou dobou uvádění projektu do provozu, snadným přetížením centrální jednotky, centralizovaným systémem sběru dat a realizace akčních zásahů. Z toho vyplývala složitá a drahá kabeláž z klimatizovaných místností řídicího počítače až k čidlům a aktorům, umístěným ve velkých vzdálenostech od řídicího centra. Od projektantů řídicího systému a odborníků na automatizaci vyžadovaly značné schopnosti programování ve vyšších programovacích jazycích. V průběhu 70. let se objevily první řídicí minipočítače, které sice umožnily částečnou decentralizaci řídicího systému, ale stále ještě představovaly více méně centralizovanou koncepci číslicového PCS. Řídicí počítače a minipočítače se používaly celá 70. a v začátku 80. let. Byla realizována velmi úspěšná řešení řízení složitých celků pomocí této techniky (elektrárny, ocelárny a válcovny, výroba kaučuku a celá chemie, farmaceutické provozy, jaderné elektrárny, atd.).

Distribuované řídicí systémy lze dělit na následující kategorie, dle použití, architektury a dalších vlastností:

1. pro elektrárny
2. pro jaderný program
3. pro ostatní technologické procesy
4. řídicí systémy budov

Některé DCS systémy jsou úzce specializované na určitou výše uvedenou kategorii, některé jsou naopak použitelné ve více oblastech. Výjimku tvoří řídicí systémy pro jaderný a vojenský program a řídicí systémy v exponovaných dopravních prostředcích (pilotované kosmické systémy, letecká civilní doprava, rychlovlaky a další), kde jsou mimořádně vysoké požadavky na bezpečnost a spolehlivost řídicího systému. Vysoce bezpečné a spolehlivé systémy jsou velmi drahé a nejsou proto nasazovány tam, kde toho není nezbytně třeba. Problematika spolehlivosti a bezpečnosti řízení (nejen u velkých DCS systémů, ale u řídicích systémů obecně) se v poslední době stává otázkou číslo jedna.

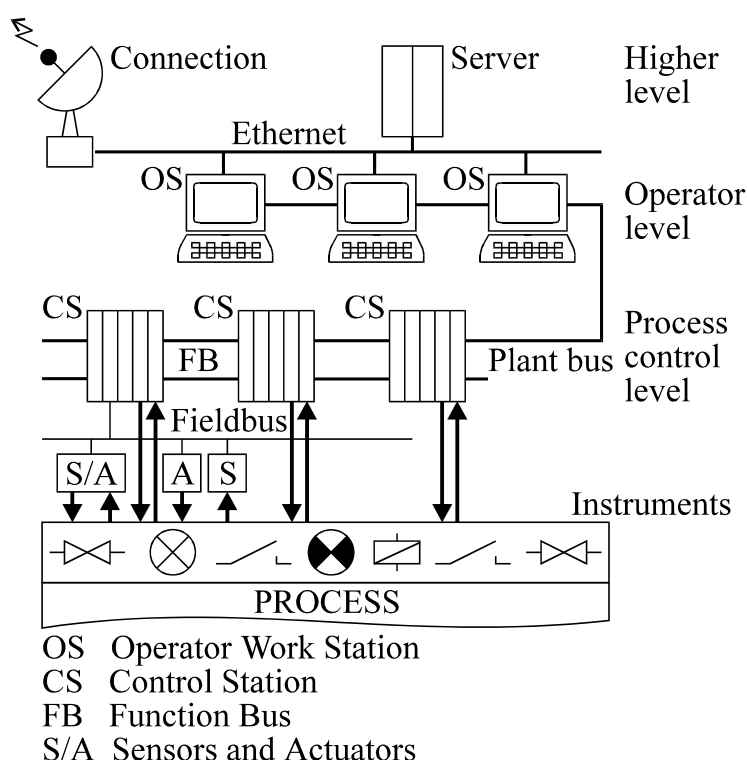
Uvedme nyní bližší specifikaci systémů DCS. Na obr.6.1 je blokové schéma typického distribuovaného systému 2. generace (Eckardt PLS 80E) ze začátku 90.let.

Tento systém je charakterizován důslednou hierarchickou výstavbou se třemi úrovněmi řízení, které jsou zdola nahoru:

- úroveň senzorů, snímačů, aktorů
- úroveň bezprostředního řízení, nazývaná též 1. úroveň řízení (technologické řízení a regulace)

- operátorská úroveň (2. úroveň řízení nebo úroveň řízení procesu)
- nadřazená úroveň (3. úroveň řízení nebo úroveň řízení výroby)

Lze uvažovat i o připojení na další vyšší podnikové a mezipodnikové řídicí úrovně. Přístroje v jedné úrovni a úrovně mezi sebou jsou propojeny komunikačními podsystemy (sériovými sběrnici). U této generace DCS jsou komunikační systémy v drtivé většině vlastním systémem výrobce nebo jsou průmyslovým standardem, který se prosadil nikoli mezinárodním standardizačním procesem (ISO), ale svými vlastnostmi a tím, že firma, která systém vyrobila ho uvolnila pro další použití (open communication system). Otevřenost systému však také může znamenat, že systém je vyvinut tak, aby byl jednoduše propojitelný s dalšími komunikačními systémy, že tedy splňuje obecně přijatá pravidla pro komunikaci. Systémy 2. generace DCS (za 1. generaci DCS můžeme považovat více méně standardizovanou podobu PCS, postavenou na řídicích minipočítačích) nejsou ještě dostatečně otevřené a lze se spolehnout



Obr.6.1 Schéma DCS 2. generace (Eckardt PLS80E)

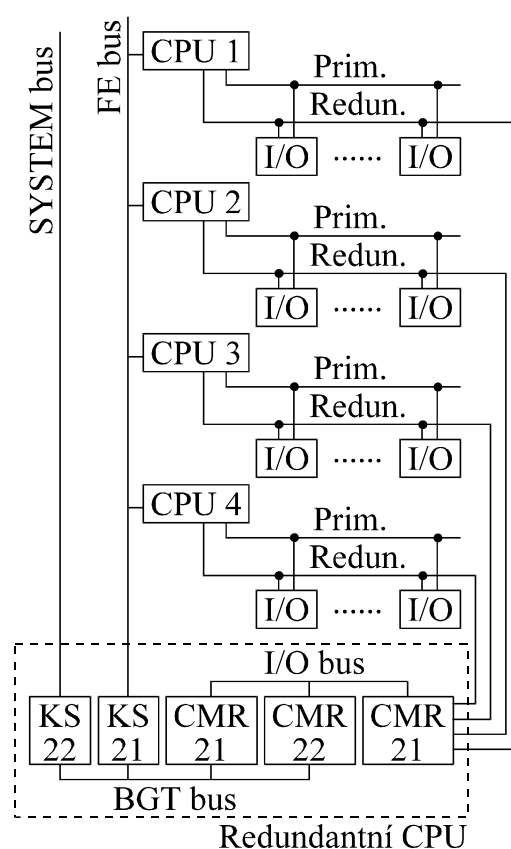
pouze na to, že budou umět komunikovat po Ethernetu od 2. úrovně nahoru. Některé z nich mohou být vybaveny některým z proprietárních standardů fieldbusů (sériových sběrnici pro propojení procesní instrumentace k 1. úrovni řízení) jako kupř. Modbus, Profibus ap.

Instrumentální vybavení jednotlivých úrovní je následující:

6.1 Úroveň procesní instrumentace:

Jde o běžné snímače a akční členy, připojené k řídicím systémům 1. úrovně zpravidla standardizovaným dvoubodovým proudovým spojem (TTY 0-20mA) nebo standardizovaným napěťovým spojem 0 - 10V ap.

Výjimečně jsou použity inteligentní snímače, umožňující předzpracování signálu a přenos po sériové sběrnici do systémů 1. úrovně. Vzhledem k v mnoha případech neúměrně rozsáhlé a drahé kabeláži se pro připojení analogových i digitálních vstupů a výstupů používají multi a demultiplexery signálů, které jsou umístěny v blízkosti technologických skupin (v rozvaděčových skříních nebo provozních skřínkách). Tyto demultiplexery (koncentrátory a rozdělovače signálů) jsou předchůdci mnohem efektivnějšího propojení procesní instrumentace pomocí sériových sběrnic. Používají se však i u vyšší generace DCS, zejména při přechodu z bezpečné do výbušné (EX) zóny a naopak.



Obr. 6.2 Způsob zálohování na 1. úrovni řízení

6.2 Úroveň bezprostředního řízení:

U těchto DCS je úroveň bezprostředního řízení vybavena procesními stanicemi, postavenými na technologii Intel nebo Motorola (32 bitové procesory řady I 80486 a vyšší nebo Motorola HC 68000 a vyšší) s operačními systémy reálného času (kupř. OS2, OS9, CDOS,

RTOS a další). Vnitřní sběrnice procesních stanic jsou Multibus II nebo VME/VMS bus. Stanice obsahují karty AI, AO, DI, DO, komunikační procesory a výjimečně speciální karty pro řízení pohonů, ventilů a pod. Na karty jsou kladeny vysoké nároky na spolehlivost, jednoduché provedení, poměrně malou integraci počtu vstupů a výstupů na jednotlivých kartách, indikaci DI a DO LED diodami na čelním panelu ap. Dále se požaduje, aby na kartách nebyly (s výjimkou procesorového modulu a paměťového modulu) žádné záložní akumulátory, žádné přepínače ap. Moduly I/O musí při výpadku stanice uchovávat poslední žádanou hodnotu, musí být dle potřeby k dispozici v EX provedení. Systém musí umožňovat výměnu karet za chodu procesu a její automatické parametrování a některé další funkce. V případě angažovaných procesů jako chemie, elektrárny a další procesy (a pro ty je kupř. výše uvedený systém PLS 80E určen), je na úrovni procesních stanic realizováno zálohování procesorů způsobem 1 ze 4. (jeden záložní procesor na 4 procesní stanice), zálohující jednak výpadek libovolného ze 4 procesorů a zároveň zálohující způsobem 1 ze 2 i I/O podsystém (viz obr. 6.2).

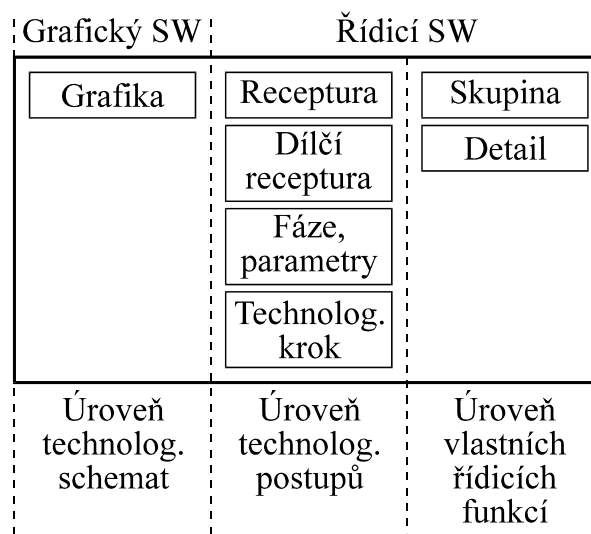
Vzájemné propojení procesních stanic je uskutečněno redundantní systémovou sběrnici. Tato sběrnice je většinou firemním systémovým real-time busem, umožňujícím práci v reálném čase. Vzhledem k velké důležitosti komunikace mezi 1. a 2. úrovní řízení, nemohli výrobci DCS ponechat vývoj systémové sběrnice na mezinárodních standardizačních aktivitách a používají v drtivé míře vlastní systémy. V případě PLS 80E to je redundantní (zdvojená) systémová sběrnice typu Ethernet s modifikovanou přístupovou metodou, umožňující definovanou dobu přístupu ke sběrnici. Přenosovým médiem je většinou koaxiální kabel. Redundantní sběrnice umožňuje peramanentní paralelní chod obou sběrnic s identickými aktuálními daty, takže v případě výpadku komunikace na jednom busu je možné s minimálním zpožděním využívat aktuální informaci z druhého podsystému. Na úrovni procesu je možné kromě zdvojeného systémového busu ještě využívat pomocný FE bus.

6.3 Operátorská úroveň:

Operátorská úroveň (2. úroveň řízení) má pro řízení technologického procesu rovněž zásadní význam. Z této úrovně je možné systém projektovat (strukturalizace systému měření a regulace, konfigurace systému, parametrizace a vlastní programování), diagnostikovat, dokumentovat, sledovat a řídit (včetně ošetření poruchových hlášení). V případě výše uvedeného systému PLS 80E je tato úroveň vybavena v maximální výstavbě až 24 operátorskými stanicemi LS. Tyto stanice tvoří standardní výkonné stanice Intel 130, určené pro nepřetržitý provoz, dále operátorská konzola, myš, resp. trackball, velkoplošná obrazovka. Systém v úrovni řízení procesu je postaven na sběrnici Multibus II jako modulární, extrémně výkonný systém. Stanice LS jsou navzájem mezi sebou a dále se všemi stanicemi CS funkční úrovně propojeny sériovou systémovou sběrnici, popsanou výše. Operátorská úroveň musí umožnit nadřazené řízení procesu. Operátoři procesu musejí mít on-line přístup k žádaným hodnotám regulátorů, realizovaných na 1. úrovni řízení a měnit tak stav systému a najíždět, tlumit provoz a přecházet do libovolného pracovního bodu v reálném čase.

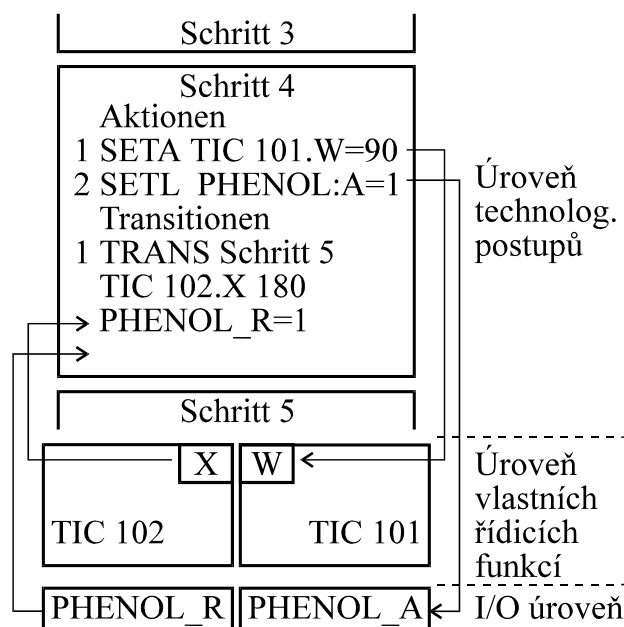
Typickou vlastností celého systému této generace a oblasti použití (chemie, tepelné procesy, farmacie plynárny, a další dávkové a pomalejší procesy), je pevně nastavená doba cyklu, která činí cca 75 ms (procesor CM22). To předurčuje systém PLS80E pro řízení v reálném čase jen v těch technologických procesech, kde obecně nejsou vysoké nároky na rychlost.

Programové vybavení DCS této generace lze charakterizovat opět tím, že se jedná o SW, vyvinutý výrobcem DCS pouze pro tento systém. Na obr. 6.3 je znázorněn strukturovaný způsob, na kterém je uživatelský software vytvořen.



Obr. 6.3: Strukturovaný SW

Software se dělí na grafický a na řídicí software. Grafická část software umožňuje tvorbu technologických schémat a to dle použité CS stanice v omezené nebo plné grafice. Z této grafické úrovně popisu procesu lze sledovat daný výřez technologického procesu a zobrazit i detailnější popis procesních dat. Z této úrovně je rovněž možné proces obsluhovat a řídit. Programy jsou vytvořeny v grafickém prostředí pro tvorbu řídicích receptur pro technologické postupy. Strukturalizace je patrná z obr.6.3.



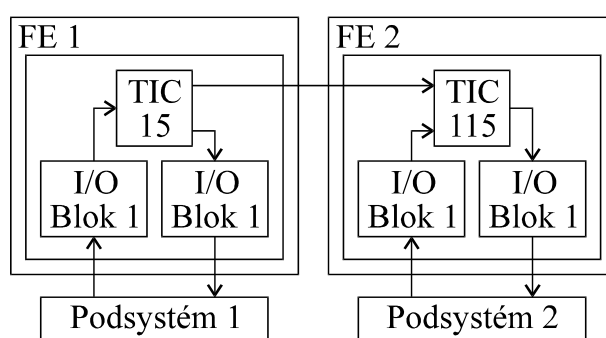
Obr. 6.4: Příklad tvorby programu

Spočívá v postupném řazení Receptury, Dílčí receptury, Fáze (se zadáním parametrů) a Technologického kroku. Toto slouží k popisu momentálního stavu řízení. Receptura spočívá v

grafickém vyjádření sekvenčního a paralelního časového sledu jednotlivých dílčích receptur. Podobně dílčí receptura je vytvořena jako serio- paralelní schéma jednotlivých fází technologického postupu. Fáze technologického postupu je pak vyjádřena sekvencí jednotlivých kroků. Teprve při tvorbě jednotlivých kroků je použit velmi jednoduchý vlastní programovací jazyk interpret EPL, pomocí kterého jsou popsány technologické kroky.

Každý technologický krok se skládá ze dvou částí. První je Aktion (akce, popisující, co se bude konat uvnitř daného kroku) a Transition (stanovující okrajové podmínky pro přechod z daného kroku na krok následující). Příklad tvorby programu pod tímto systémem ukazuje obr.6.4. Na tomto obrázku je zároveň dobře vidět tato vzájemná souhra obou programovacích úrovní řídicího SW z obr.6.3 (úrovně popisu řídicích procedur a úrovně popisu vlastních funkcí) s "fyzickou" I/O úrovní. Z transparentnosti programovacího jazyka EPL je snadné rekonstruovat funkci kroku 4, ve kterém žádaná hodnota regulátoru TIC101 bude nastavena na 90 °C a binární výstup Phenol_A v Bloku I/O (t.j. přímý zásah do úrovně I/O z úrovně popisu řídicích procedur) bude nastaven na log.1. Přechod na krok 5 se uskuteční tehdy, jestliže regulovaná veličina v obvodu regulátoru TIC102 je větší než 180 °C a binární vstup signálu Phenol_R je na hodnotě log.1.

Grafické programovací prostředí rovněž umožňuje grafické zobrazení proměnných (výstupních a řídicích veličin, žádaných hodnot stavu registrů, stavu čítačů ap. a to formou až 8 sloupcových grafů v jedné obrazovce. Programování cyklických procesů (regulační obvody, úroveň binárního řízení), které nezávisí na programovém řízení pomocí receptur se programuje v prostředí EPL (Eckardt Process Language) s využitím knihovny automatizačních bloků, která obsahuje 23 funkcí jako jsou funkce pro zpracování booleovských rovnic, aritmetické operace, čítače, časovače, masky, registry, funkce pro tvoření propojení, funkce jedno- i obousměrného řízení otáček motoru, spojitý regulátor, standardní regulátor, třípolohový regulátor, Schmidtův prediktor, a funkce adaptivního regulátoru. Součástí tohoto procesu programování logických, aritmetických a regulačních funkcí je konfigurování použitých bloků. Spočívá v přiřazení I/O bloku (resp. proměnné) k fyzickému I/O zařízení (resp. signálu). Po tomto kroku je možné daný signál připojit k libovolnému funkčnímu bloku libovolné funkční jednotky. Příklad tohoto softwarového ranžirování a propojování jednotlivých bloků (jde o virtuální spoje) je ukázán na obr. 6.5



Obr.6.5: Programové ranžirování

Vlastní operátorská činnost (sledování procesu, kvitování chybových hlášení, přestavování parametrů, editace řídicího programu a další operátorská činnost je podporována menu-orientovaným prostředím s pomocí předprogramovaných kláves (hotkeys) a oken. Jde především o:

- procesní a systémová hlášení
- systémové údaje
- volně definovaná návěští
- obslužné funkce

Zakázané stavy systému, havarijní a kritické stavy jsou hlášeny odpovídajícím způsobem obsluze a vyžadují její odpovídající reakci. Na tyto funkce bezprostředně navazuje důkladný systém protokolování průběhu procesu. Kupř. každá činnost operátora je přesně zaprotokolována. Systém chybových a procesních hlášení, protokol zásahů a činnosti obsluhy je ukládán a v seříděné formě archivován na disku. Ke každé probíhající receptuře je automaticky zhotoven protokol (charge- protokol), který obsahuje všechna data (parametry, zadané hodnoty, časové údaje o jednotlivých fázích). Systém umožňuje na jakékoli ze stanic úrovně operátora zobrazovat průběh procesních proměnných. U každého projektu řídicího systému je velký důraz kladen na zhotovení dokumentace ke konfiguraci, parametrizaci a programování řídicího systému. Trend vede k systémům pro automatické zhotovení dokumentace. Systém PLS80E automaticky umožňuje zhotovit pouze dopřednou dokumentaci a reagovat na případné změny jen z jedné operátorské konzoly. Změny, které byly zhotoveny z jiné stanice LS není systém schopen zaznamenat (t.zv. dopředná forma tvorby dokumentace).

6.4 Nejvyšší úroveň řízení:

Na nejvyšší úrovni řízení, která je určena pro řízení výroby, jsou uživatelům nabídnuty komfortní podmínky pro zpracování a vizualizaci vlastní produkce. Jedná se o systém EMIS (Eckardt Management Information System), který umožňuje volně konfigurovatelnou tvorbu výrobních diagramů, tabulek a grafů při zachování principu on-line konfigurování tohoto nadřazeného systému. EMIS umožňuje rovněž

- výpočet a zobrazení výrobních proměnných prostřednictvím dat z procesu,
- tisk výrobních dat na obrazovku nebo tiskárnu ve formě textu, tabulek, diagramů, přenos těchto dat do systému datové podpory, přenos povelů až do úrovně bezprostředního řízení procesu a další.

6.5 DCS nové generace

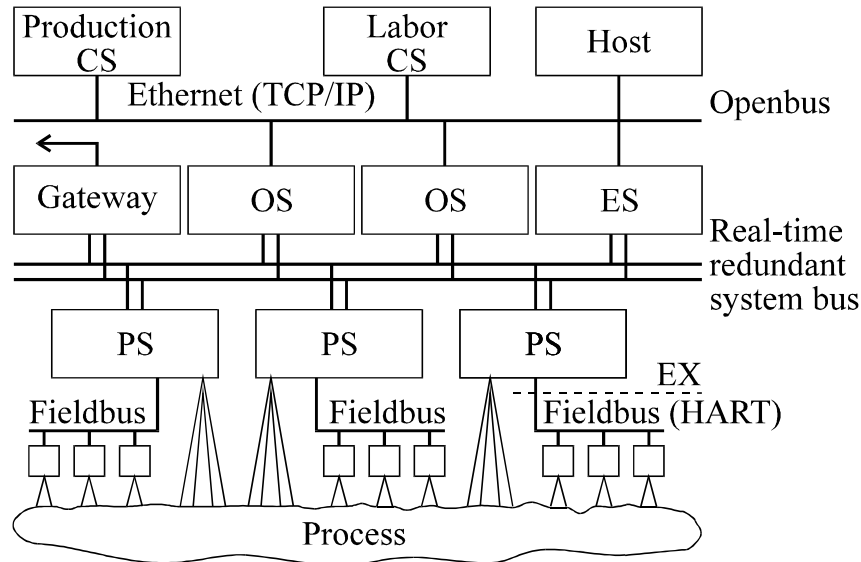
S vývojem mikroelektroniky přišla na trh nová generace DCS (dle klasifikace, uvedené v těchto skriptech, jde o 3.generaci DCS), poznamenaná vývojem, zkušenostmi a požadavky uživatelů těchto systémů. Je charakterizována přechodem na standardy, zejména v oblasti SW (prostředí Microsoft Windows) a komunikačního pod systému (fieldbusy na úrovni procesní instrumentace).

Požadavky, které vedly ke vzniku nové generace DCS se dají shrnout do následujícího stručného přehledu [23]:

- schopnost práce v reálném čase
- vysoká funkčnost v důsledku masivní redundance

- otevřenost a interoperabilita
- průchodnost

Strukturu DCS nové generace je možné zobecnit schématem uvedeným na obr. 6.6.

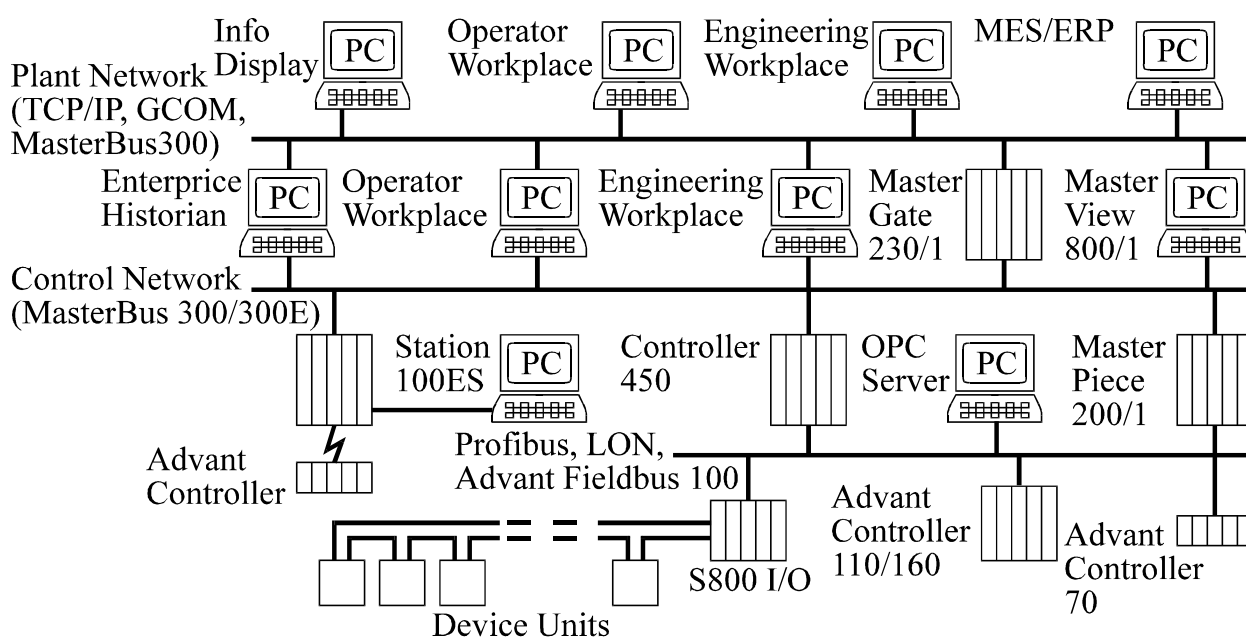


Obr. 6.6: Blokové schéma velkého DCS nové generace

V maximálním provedení se skládá z:

- úroveň procesní instrumentace (pasivní i inteligentní instrumentace včetně standardizovaných fieldbusů HART, FF, Profibus, FIP, InterBus a dalších)
- úroveň řízení technologie (osazenou procesními stanicemi PS, vybavenými výkonnými mikropočítači Pentium a vyšší s multitasking, real-time operačním systémem)
- systémové redundantní sériové sběrnice (real-time system bus)
- úroveň řízení procesu osazené operátorskými stanicemi (OS) pro řízení a sledování procesu, dalšími prostředky MMI (zobrazovací panely, velkoplošné obrazovky, provozní klávesnice, track-ball a další) a jednou inženýrskou stanicí (ES), umožňující konfigurování, parametrování, programování projektu řídicího systému (rovněž gateway umožňující připojení dalších počítačů může být připojen k této systémové sběrnici, která je zpravidla systémově specifická)
- nadřazené úroveň řízení výroby propojující daný DCS s jinými DCS (DCS pro laboratoře a předvýrobní etapy), serverem nebo výkonnou stanicí řízení výroby a podniku a řídicím systémem řízení podniku

Jedním z představitelů této generace může být systém Advant OCS firmy ABB. Blokové schéma systému je uvedeno na obr. 6.7

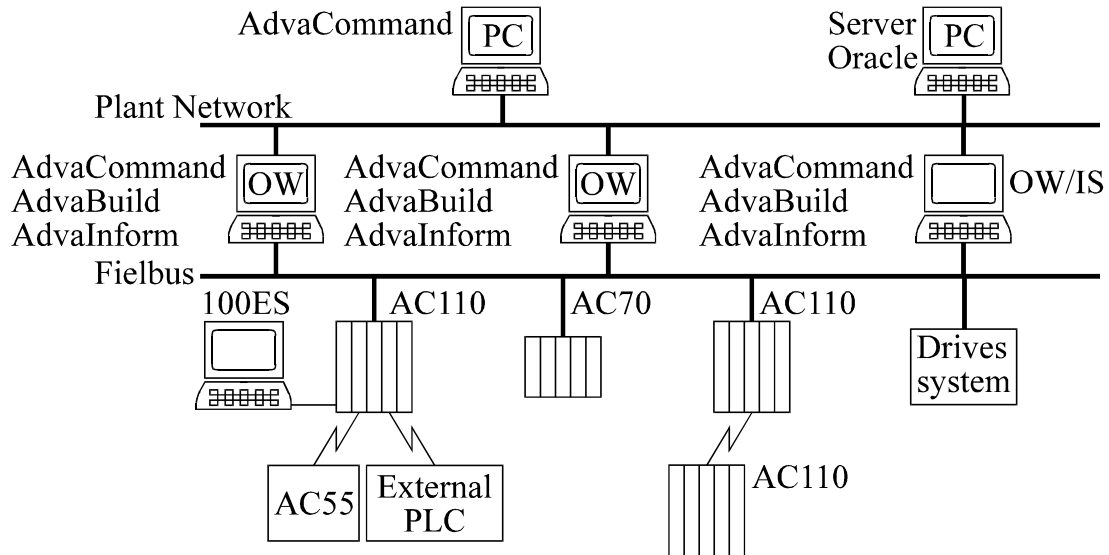


Obr. 6.7: Advant OCS

Systém vychází z úspěšného systému Master této firmy (rovněž DCS 2. generace). Systém Advant OCS (Open Control System) vychází kontinuálně z tohoto vývoje a implementuje na obou úrovních kromě nových stanic Advant i stanice master Piece a Master View systému Master. Na rozdíl od některých jiných DCS 3. generace využívá téměř výhradně firemních komunikačních sběrnic. Na nejnižší komunikační úrovni mezi jednoduššími procesními stanicemi Advant Controller 70, Advant Controller 110/160 a inteligentními přístroji a odloučenými I/O jednotkami S800I/O používá jak firemní fieldbus Advant Fieldbus 100, tak LonWorks nebo Profibus. Interbus S se používá pro propojení S800I/O s jednotkami řízení pohonů. Na vyšších úrovních jsou použity již jen firemní sériové sběrnicové protokoly. Na úrovni systémové sběrnicové z obr. 6.7 to je MasterBus 300/300E, umožňující provoz v reálném čase. Operátorská úroveň je s úrovní řízení výroby propojena Ethernetem s TCP/IP protokolem nebo MasterBusem300 nebo GCOMem). Propojení jednotlivých komunikačních sběrnic je uskutečněno buď řídicím systémem (Advant Controller 450) nebo mezisběrnicevým spojením MasterGate 230/1 (Ethernet a systémová sběrnice). Na schématu je zobrazena celá nabídka systému Advant jak v oblasti procesních stanic:

- Advant Controller 55
- Advant Controller 70,
- Advant Controller 110/160,
- Advant Controller 410,
- Advant Controller 450,

tak v oblasti operátorských a inženýrské stanice MasterView 800/1, Advant Operator Workplace, Advant Engineering Workplace i mezisběrníkovým spojem (MasterGate 230/1). Na obrázku jsou rovněž vidět jednotky odloučených I/O (S800/1), které jsou jedním systémem z kolekce S400 I/O, S600 I/O, S100 I/O. Ze systému Advant je možné realizovat jednodušší a levnější konfigurace dle potřeby projektu, kupř. jen ze stanic Advant Controller 110 a Advant Controller 70 a jednotek odloučených I/O dle obr.6.8.



Obr. 6.8: Řídicí systém ADVANT, postavený ze stanic AC 110 a AC 70

Systém Advant má několik provedení, která se liší především aplikačním SW. Tak kupř. systém Advant Power je modifikací ADVANT OCS pro elektrárny, ve kterém jsou posíleny funkce řízení a regulace, potřebné pro dané aplikace.

V oblasti SW pracují systémy Advant s operačním systémem UNIX a XWindow Systemem, ale rovněž s prostředím Microsoft Windows, Motif, DDE a SQL. Umožňují proto pohodlnou konfiguraci systému řízení v grafickém prostředí s knihovnou předdefinovaných funkčních bloků (AMPL Control Configuration).

Vzhledem k velké konkurenci systémů DCS navzájem mezi sebou i ze strany PLC a PC orientovaných "soft control" systémů, a v důsledku celosvětového trendu ke koncentraci výrobních a finančních zdrojů, došlo v posledních letech ke slučování výrobců systémů DCS. V některých případech skončila výroba jednoho ze systémů DCS, v jiných případech koncern používá nadále více DCS systémů, dokonce s masivní inovací. Kupř. koncern ABB má ve své nabídce jednak výše uvedený systém Advant OCS (firma ABB), dále systém Freelance 2000 (innovovaný systém z firmy Hartmann&Braun) a systém Symphony (firma Elsas Bailey). Tato politika umožňuje koncernu nabídnout celou šíři DCS pro nejrůznější oblasti a velikosti aplikace.

Největším konkurentem pro klasické DCS jsou PC orientované DCS, vyznačující se nižší cenou, větší flexibilitou, v dnešní době již srovnatelným výpočetním výkonem i komfortem v programování, konfigurování, parametrování i vizualizaci procesu, než výše uvedené kompaktní DCS, postavené z velké části na sice výkonných, ale drahých UNIX pracovních stanicích. V každém případě však velké DCS představují stále jediný řídicí prostředek, zaručují potřebnou vysokou míru spolehlivosti v oblastech, kde je potřeba ošetřit velmi vysoký počet vstupů a

výstupů nejrůznějšího charakteru a kde spolehlivost a bezpečnost je naprosto kategoričným požadavkem. Jejich předností je rovněž kompaktnost celého systému.

V následujícím přehledu jsou uvedeny kompaktní DCS 3. generace, které byly v r.1997 na trhu automatizace [23]:

Firma	Produkt	Typ aplikace
ABB	Advant OCS/Master-SW	střední
	Advant OCS/MOD300-SW	velký
	Freelance 2000 (Digimatic Hartmann-Braun)	malý
	Symphony (Contronic S Hartmann-Braun)	střední až velký
Fisher & Rosemount	Delta V	malý
	PROVOX	střední až velký
	RS3	střední až velký
Honeywell	TPS Total Plant Solution	střední
Foxboro Eckardt	PLS 80E	velký
	I/A Series	střední
Moore Products	APACS	střední
M-Tec	A/S Open	střední
PCC	Apron NTi	malý až střední
Siemens	Simatic PCS 7	střední až velký
	Simatic PCS	malý až střední
Valmet	Damatic XDi	malý až střední
Yokogawa	Centum CS	střední
	Centum 1000	malý
a některé další

7. SYSTÉMY REÁLNÉHO ČASU V PRŮMYSLOVÉ AUTOMATIZACI

7.1 Úvod

Režim reálného času počítačového řídicího systému (řídicího počítače, mini a mikropočítače) je takový režim, ve kterém aplikační programy počítačového řídicího členu musí být stále schopny zpracovávat nové události, přicházející z řízeného procesu, přičemž zpracování událostí a odpovídající reakce musí nastat do určitého předem definovaného (krátkého) časového limitu. Data mohou přicházet na výpočetní systém náhodně nebo v předem definovaných časových okamžicích.

Počítačový řídicí člen může být realizován jako řídicí počítač nebo řídicí mini a mikropočítač nebo vestavěný mikropočítač, který je součástí kompletní aplikace.

Systémy reálného času nejsou ničím novým v řídicí technice ani v běžném životě. Právě z běžného života můžeme jmenovat následující příklady systému, pracujících v reálném čase jako :

- Řidič auta, které jede v hustém městském provozu musí stále sledovat jak dopravní světla, tak další auta, chodce, mnohdy i stav vlastního auta a posádky a musí být neustále připraven reagovat za účelem zabránění nehodě a dosažení cíle.
- Hospodyně při svých každodenních domácích pracích musí dbát o vaření, úklid domácnosti, chování dětí, ovládání domácích přístrojů tak, aby ve stanoveném čase bylo vše hotovo. Přitom musí reagovat na měnící se události a přizpůsobovat jak organizaci své činnosti, tak rychlost reakci.
- Řídicí systém elektrárny stále dohlíží na všechny technologické podsystemy a reaguje změnou parametru řídicích clenu na měnící se situaci rozsáhlého celku. Doba odezva je striktně určena a řídicí systém ji musí dodržet, aby nedošlo k poruše nebo katastrofě.

Cítíme, že poslední případ je z uvedených případů hoden nejvyšší pozornosti, třebaže základní vlastnost všech uvedených případů, tj. včasná reakce na měnící se události a kvaziparalální zpracování přicházejících podnětů z okolí je jevem, který je všechny charakterizuje. Je zde ve všech případech nutná včasná reakce s náležitou přesností. Na dalších příkladech je to opět ještě zřetelněji vidět:

- Pro rezervační systém letecké společnosti existuje okolí, sestávající z cestujících, kteří chtějí využít služeb letecké společnosti. Počítačový systém, zajišťující rezervaci letenek musí reagovat na přání zákazníků a vyhledat a rezervovat spoj v čase, který je akceptovatelný pro zákazníky, jinak hrozí nebezpečí ztráty klientů.
- Bankovní informační systém, informující klienty o provedených operacích je plně distribuovaný systém realizovaný na množství terminálů a počítačů. Pokud nepracuje dostatečně rychle, mohou být operace klientů neadekvátní, čímž může docházet k vyšším nákladům v důsledku nepřesně provedených operací a tím opět k možné ztrátě části klientů.

Všechny výše uvedené případy z každodenní praxe jsou si podobné v tom, že jde o výpočetní operace, které jsou vázány na čas. Na požadavek z venčí musí tyto požadavky

akceptovat, klasifikovat a vydat odpovídající reakci. Čas odezvy je určen požadavky okolí a nikoli vlastnostmi výpočetního systému.

Obr. 7.1 Ztráty při pozdní reakci ŘP na událost.

Obr. 7.2 Oblast působnosti RTS

Přes výše uvedenou podobnost všech systémů, je třeba rozlišovat ještě v důsledku jedné důležité okolnosti. Zatímco v prvních třech případech požadavky na řídicí člen (řidiče, hospodyně, řídicí systém elektrárny) jsou značně striktní a nepřesná nebo pozdní reakce může způsobit velké škody, nebo smrt či dokonce katastrofu, v dalších dvou případech je v případě nesprávné nebo pomalé funkce řídicích systémů rezervace a informačního systému peněžního ústavu mnohem měkčí. Proto se hovoří o dvou typech systémů reálného času. První tři případy spadají do kategorie *tvrdých systémů reálného času (hard real-time systems)* a dva zbývající do kategorie měkkých systémů reálného času (*weak real-time systems*). Na obr. 1.1 vidíme graf ztrát při špatné funkci tvrdých a měkkých systémů reálného času. V Tab.1.1 jsou pak uvedeny některé příklady obou kategorií systémů reálného času.

Požadavky na RTS	hard RTS	soft RTS
Následky pozdní reakce	škody na lidech, okolí, velké ekonomické ztráty	ekonomické ztráty rostou s časem zpoždění
Příklady	řidič auta, hospodyně, řízení elektrárny	rezervace letenek, bankovní operace, plánování dopravy
Kriterium optima	maximální výkon v nejkritičtějších stavech	optimalizace průměrného výkonu

Tab 7.1 Příklady systémů RT

Zatím jsme v žádném z uvedených případů nehovořili o době odezvy v jednotkách času. Jak již bylo řečeno, správná reakce systému reálného času je vázána na reakce řízeného systému a závisí proto na časových konstantách nebo též povolené časové odezvě řídicího systému. V každém případě systém reálného času musí reagovat *včasně* vzhledem k parametrům procesu.

V tomto kursu jsou diskutovány pouze systémy reálného času s tvrdými podmínkami (*hard real-time systems*).

7.2 Systémy průmyslové automatizace

Technický proces, který především bude předmětem řídicích systémů průmyslové automatizace je proces, jehož fyzikální veličiny jsou měřeny technickými prostředky a

technickými prostředky je tento systém rovněž ovlivňován. Technické procesy se (dle DIN 66201 Teil 1, Nr. 1.3) dělí na výrobní, distribuční a shromažďovací procesy.

Technické procesy dále dělíme dle několika kritérií na :

- spojitě a nespojitě
- homogenní a nehomogenní
- procesy transformující nebo přetvářející materiály, energii nebo informaci
- na technologické - spojitě, výrobní – produkční, distribuční, měřicí a testovací
- dle povahy procesních veličin na kontinuální, dávkové, kusové procesy

Procesy jsou charakterizovány proměnnými fyzikální a chemické povahy. Úkolem řídicího systému je tyto procesní proměnné měřit s dostatečnou přesností a rychlostí a svými výstupy působit na vstupy systému tak, aby bylo dosaženo požadovaného stavu včetně časového průběhu vývoje procesních proměnných. Dle toho, jakým způsobem je řídicí člen (řídicí počítač ŘP) k technickému procesu připojen, rozlišujeme 4 způsoby připojení ŘP k procesu.:

Způsob připojení	Propojení ŘP – proces	Propojení proces – ŘP
Nepřímé	obsluha	obsluha
přímé, otevřené	přímé spoje	obsluha
prime, otevřené	obsluha	prime spojení
prime, uzavřené	prime spojení	prime spojení

Tab.7.2 Způsoby připojení ŘP k procesu

První dva způsoby z Tab. 7.2 nelze považovat za případy řízení v reálném čase (RT) vzhledem k tomu, že čtení procesních dat obsluhou a jejich ruční zadávání do RP je nepřesné, subjektivní a pomalé.

V praxi se nejvíce vyskytují případy 3 a 4 z Tab. 7.1 a jsou proto předmětem následujícího rozboru. Příklad 3 definuje vlastní systémy automatického sběru dat (Data Acquisition Systems) a případ 4 popisuje přímé řízení počítačem v uzavřené řídicí struktuře (též označované jako DDC).

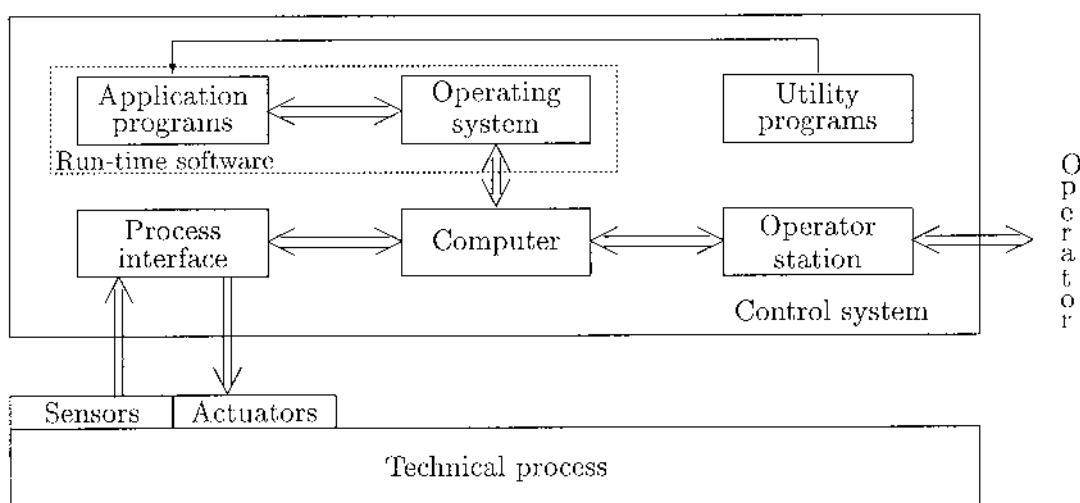
Na obr. 7.3 je schéma řídicího systému reálného času a jeho připojení na proces. Je zřejmé, že je vhodné tuto strukturu dělit na:

1. HW prostředky tvořené:

-rozhraními jako:

- AI
- AO
- DI
- DO

- Citací
- Časovací
- Přerušovacími obvody
- řídicím členem (RP nebo mikropočítačem)
- ovládacími a zobrazovacími přístroji pro obsluhu systému



Obr. 7.3 Schéma řídicího systému RT

1. SW prostředky tvořené :

- operačním systémem
- uživatelskými programy
- vývojovými a dalšími pomocnými programy

Poslední z uvedených programových prostředků se od dalšího SW podstatně liší v tom, že současně nemusí být součástí systému reálného času. Co se týče použití architektury z obr. 7.3 můžeme jmenovat následující aplikace:

1. automaticky sběr dat z procesu (informační systém)
2. dohled a protokolování technického procesu
3. řízení a regulace
4. programové řízení
5. řízení a on-line optimalizace parametru
6. řízení a stabilizace velkých celků

Poznamenejme, že první dva případy patří do kategorie 2 a 3 Tab.7.2, zatímco ostatní jsou z kategorie 4.

7.3 Historický přehled

O prvních systémech reálného času pro řízení lze hovořit až kolem roku 1960, kdy byly počítače nasazeny do průmyslu, kosmických systému, letecké dopravy. Jedny z prvních velkých systému tohoto typu byl American Airlines SABRE pro rezervování leteckých spojů, dále počítačový informační systém burzy v New Yorku a některé další terminálové bankovní systémy. Řídicími členy byly výlučně centralizované řídicí počítače. Nespolehlivý a velmi drahý HW řídicích počítačů (ŘP), velmi drahá kabeláž od skříní jednotek I/O, umístěných zpravidla blízko centrální počítačové jednotky (mnohdy i v klimatizovaných místnostech) zabraňovaly většímu rozšíření počítačového řízení.

V průběhu 70. let se situace značně zlepšila s příchodem levnějšího HW řídicí minipočítačů, s celkovým zvětšením spolehlivosti HW a s částečnou decentralizací ŘP na řadu řídicích minipočítačů. Zde jmenujme celou radu minipočítačů PDP 8 a zejména PDP 11 firmy DEC (Digital Equipment Corporation). Tak mohl být ŘP nasazen do řízení nejrůznějších procesů včetně atomových reaktorů, řízení laboratoří a vojenských mobilních systému.

S příchodem mikropočítačů (1971 – Intel 4004) a zejména s jejich rychlým a úspěšným vývojem (1975 – I8080 atd.) došlo k další decentralizaci HW řídicích počítačů, k dalšímu snížení nakladů na kabeláž, zjednodušení ladění programu atd. Tento vývoj především implikoval rozvoj programovatelných automatů (PLC) jako prvního vážného masového nástupu počítačů do řízení nejrůznějších procesů (od strojírenské výroby, systému řízení a distribuce tepla až po nejrůznější další i velké technologické procesy, jako jsou cementárny, válcovny a další).

Pokrok v HW řídicích členu byl se zpožděním a pomaleji sledován i vývojem programových prostředků. Nejdrive byly vyvinuty operační systémy (OS) a programovací jazyky. První aplikace byly implementovány bez OS, t.j. napsány v assembleru a implementovány přímo. Jiz s příchodem minipocitacu byly k dispozici multitaskingove operační systémy reálného času (RTOS) a začaly uživatelským programům umožňovat virtuální paralelismus s přidělováním procesoru. Od poloviny 60. let se začalo pracovat na vyšších programovacích jazycích pro programování systému reálného času (RT). Přitom se sledovaly dvě cesty. Jednak se pomocí rozšíření pro RT operace začaly používat jazyky pro vědeckotechnické výpočty jako Basic nebo pomocí balíků podprogramů i jazyky jako Fortran (Industry Fortran). Současne byly vyvinuty nové jazyky pro RT systémy (ADA, LTR, PEARL), specialne určené pro průmyslové řízení. Přes úspěšnou mezinárodní standardizaci nedosáhly významného rozšíření.

S příchodem počítačových sítí i v průmyslových řídicích systémech došlo k vývoji distribuovaných RTOS jako QNX i odpovídajících programovacích jazyků (Mehrechner PEARL) pro realizaci fyzického paralelního zpracování výpočtu. Přesto je tento vývoj nedostatečný i v dnešní době.

Jiz od 60. let sleduje vývoj SW principy strukturovaného programování a na těchto principech jsou programové moduly pro RT systémy také vyvinuty. Na tomto základe pak jsou programátory RT systému k dispozici inteligentní prostředky pro vývoj SW zvané jako CASE (Computer Aided Software Tools) na principech strukturovaného návrhu. Posledně jmenované systémy vyvolaly v průběhu 70. a 80. let vydání komplexních a přesných doporučení, jak musí být provedeno formulování požadavků na tvorbu řídicího SW (Requirements Engineering), aby systémy CASE byly použitelné na širokou škálu úloh obecně.

7.4 Zaklady koncepce systému realneho casu

Systém realneho casu (RT systém) se odlišuje od ostatních forem zpracování dat právě pro svůj explicitní vztah ke kategorii čas. Z matematického hlediska posuzujeme čas jako jednodimenzionální euklidovský prostor reálných čísel. Čas vystupuje ve dvou základních požadavcích uživateli systému realneho casu, které RT systémy musí i v nejtežsích situacích splňovat a to je :

- včasnost
- soucasnost

Soucasnost je korelované zpracování dat od více než jednoho vstupu ve stejném časovém horizontu. Od řídicího systému letadla se kupř. požaduje, aby současně koreloval hodnoty o poloze letadla a jeho rychlosti a přitom udržoval teplotu a vlhkost v kabině pro cestující. Což jsou dva kvalitativně odlišné požadavky. Současně zpracování paralelních procesů v jednom řídicím systému a zejména s jedním procesorem se da zajistit jen přibližně. Soucasnost se proto v tomto případě da zajistit jen tehdy, kdy časové konstanty řízeného procesu a rychlost RP se navzájem významně liší. Pro realizaci obou požadavků na systémy realneho casu slouží mechanismy prerušení a paralelní výpočetní prostředky (task), které se v jiných počítačových koncepcích vyskytují jen v úrovni operačních systémů. Prostřednictvím prerušovacích signálů (prerušení) může technicky proces prerušit a odsunout právě probíhající výpočet. Tímto mechanismem mohou jiné úlohy využít kapacity CPU a paměti pro ošetření stavu technického procesu, který prerušení způsobil. V počítači jsou adekvátně modelovány výpočetní procesy, reagující na situace technického procesu. Tato koncepce není závislá na použité struktuře RP a nazývá se programování systému realneho casu. V RT systémech s tvrdými časovými podmínkami musí být požadavek na soucasnost splněn i v nejnepříznivějších případech (vypadky, chyby, přetížení), což vede k dalším specifickým požadavkům :

- předvídatelnost časových událostí
- spolehlivost

Vzhledem k tomu, že data přicházejí z procesu náhodně, vedlo to k dojmu, že systémy realneho casu nemohou a nesmí pracovat deterministicky. Tento závěr je nesprávný. Ve skutečnosti může být technicky proces tak komplexní, že jeho chování se jeví jako náhodné. Přesto musí být snaha navrhnout reakce RP co nejpresněji s plnou předvídatelností těchto reakcí. To platí zejména pro případ současného výskytu více událostí při působení poruch, přetížení a nejrušnějších výpadků. V těchto případech očekává uživatel, že RP povolná sniží kontrolovatelným a transparentním způsobem výkon zpracování požadovaných operací. Jen plně deterministické systémové chování RP může zaručit bezpečnou funkci programovatelných přístrojů pro vysoce bezpečné (a vysoce funkční) aplikace. Vyše uvedené pojmy předvídatelnosti a determinismu lze ukázat na příkladu protipožárního zásahu. Není jasné, kdy začne hořet, ale za normálních okolností očekáváme příjezd hasičů do určitého časového horizontu po oznámení požáru. Jak vidíme, předvídatelnost časových reakcí RS patří mezi základní požadavky, na něj kladene. Předvídatelnost časových reakcí má vliv i na včasnou reakci RT systému.

Pres sebe lepší naplánování činnosti RP zůstává stále možnost, že v případě výskytu kritických situací (vypadky, poruchy a další) budou některé uzly RP přetíženy. V oblasti výpočetní techniky jsou vyvinuty metody, umožňující sdílení a prerozdělování zdroje (CPU, a paměti a dalších) za účelem vyrovnání nerovnoměrného zatížení těchto zdrojů v kritických

situacích tak, že výpočetní procesy mohou běžet na méně využitých uzlech distribuovaných výpočetních systémů. Tyto metody nejsou v automatizačních systémech ještě plně využity a implementovány m.j. i kvůli deterministickému HW připojení I/O podsystemu RP k procesu.

Definice systému reálného času (RT systém) implikuje požadavek spolehlivosti, kdy tyto systémy, které jsou v permanentním nasazení musí splnit své funkce i v případě výskytu poruchy. Není přípustný výpadek RP vzhledem k požadavku vysoké funkčnosti (fail tolerant) řídicího počítače a to jak v oblasti HW tak SW. Očekávání vysoké spolehlivosti nemusí znamenat nesplnitelný požadavek absolutní spolehlivosti, neboť je jasné, že žádný systém nemůže být absolutně spolehlivý. Je třeba však definovat rozsah povolených škod a odpovídající reakce, jak dosáhnout toho, aby pravděpodobnost vzniku škod byla minimální.

Je rovněž dobře prolomit některé falešné představy o RT systémech. Tak kupř. Systémy mnohauzivatelské a mnohafunkční ještě nutně nemusí být RT systémy. Mnohem podstatnějším rysem než rychlost je včasné zpracování dané úlohy. Místo požadavku na spravedlivé přidělení zdroje jednotlivým úlohám a minimalizace průměrné délky odezvy výpočetního systému, vystupuje u RT systému mnohem více do popředí požadavek ošetření kritických stavů, chování v nejméně příznivém případě, pevná doba přístupu k procesoru, a definování a dodržení maximální povolené doby odezvy.

Požadavek na maximální využití procesoru je ve většině případů kritériem klasické informatiky. Pro řídicí techniku je zcela irrelevantní kritérium, zda využití nasazení počítače je optimální nebo ne, protože náklady jsou posuzovány především z hlediska výkonu technického procesu a z toho plynoucích ztrát či nákladů při současném splnění požadavku na bezpečnost. Když posoudíme, co lze získat na jedné straně a na druhé straně ztratit tím, že optimalizujeme využití procesoru, je jasné, že požadavek vysokého průměrného zatížení zdroje v RT nelze preferovat.

7.5 Inženýrský návrh „tvrdých“ systému reálného času (hard real-time systems)

Při návrhu RT řídicích systémů je snaha konstruovat takové systémy, které budou garantovat dostatečnou kvalitu funkce i při aplikacích, které jsou z hlediska bezpečnosti kritické.

RT systémy jsou určeny pro řízení procesu, ve kterých se vyskytují jednak periodické, především známé operace, jednak neperiodické, náhodné události. Zátěž RS je proto v čase proměnná. Některým událostem jsou přiřazeny časové horizonty, ve kterých RT systém musí zajistit reakci, odpovídající dané události. Aby se to dalo zajistit, je třeba provést analýzu procesu a naplánování časového průběhu funkcí RP. Avšak ani ta nejlepší analýza nepřinese zadání výsledky, jestliže nepočítá s nejhorsími případy zatížení a spíčkového přetížení zdroje. Proto je již před návrhem RT řídicího systému třeba definovat největší zatížení, aby se dalo nalézt řešení. S rostoucím zatížením klesá výkon RS. Tam, kde se protínají křivky zatížení a výkonu, tam systém přestává plnit svou funkci. Přesto je potřeba i na takové případy, které vždy mohou nastat, najít řešení. To spočívá ve snížení přesnosti výpočtu nebo odsunutí některých časově nekritických úloh. Každá metodika návrhu PR musí mít tyto mechanismy.

Okolí SW systému je tvořeno HW řídicího počítače a vlastním technickým procesem. Systémy reálného času musí vyhovovat požadavkům, které byly uvedeny v předeslém, i v tom případě, že prostředí vlastního SW systému reálného času vykazuje odchylky od předpokládaného nebo definovaného chování. Tato vlastnost SW systému reálného času se

nazyva robustnost. Z tohoto rozboru dale plyne, ze specifikace pozadavku na systém musi byt uplna. Specifikace tedy musi obsahovat vsechny mozne scenare a musi definovat reakce na nejruznejsi mozne vypadky včetne nahradnich situaci, kdy je vypadek takove povahy, ze se neda dosahnout predepsaneho cile rizeni.

V zasade jde o dva nahradni stavy systému, jako reakce na vypadek dle cile, ktereho ma systém dosahnout. Jsou to stavy :

- Nouzovy stav; U nekterych procesu se vyskytuje bezpecny klidovy stav (roboty, dopravniky, linky).
- Odolnost vuci poruse; U tech procesu, u kterych neni bezpecny stav definovan, musi byt RTS navrzen jako fail tolerant, prestoze tim roste slozitosť a cena (v dusledku zalohovani a redundance). Prikladem techto procesu jsou RP letadel, vytahu, kosmicke techniky a dalsi.

U mnoha řidicích systému RT jsou tyto metody v urcite mire pouzity a vice ci mene vhodne se vzajemne doplňuji. Proto pri nasazeni nejakeho konvencniho RTS je treba posoudit jeho moznosti v kazdem z vyse uvedenych aspektu a pri znalosti konkretniho procesu rozhodnout o vyberu a implementaci nejvhodnejsiho systému realneho casu.

Systémy realneho casu (RTS) predpokladaji soucasne zpracovani nekolika ulok (tasks). Pri navrhu RTS je proto nutne rozhodnout, zda ulohy budou prirazeny striktně HW nezavislym procesorum nebo zda to bude ukol multitaskingoveho RTS. V soucasne dobe u multiprocessorovych systému je situace jasnejsi, presto je treba cely systém rozdelit na mensi, vice ci mene nezavisle funkčni celky a presne definovat synchronizaci a komunikaci techto mensich celku. Ve fazi implementace je nutne vyuzit moznosti OS a programovaciho jazyka k paralelnimu a fail tolerant vyuziti systémovych zdroju. Jakmile je RTS navrzen a realizovan, musi byt testovan. Testy jsou komplexni a nelze postupovat heuristicky, nybrz systémové a komplexne. Testovani predstavuje ½ ceny vyvoje RTS.

Zaverem lze shrnout pozadavky, ktere by mel splnovat navrh RTS:

- Integrovaný pohled na systémový a SW navrh s uplnou specifikaci vsech moznych stavu RS a chovani okoli (prostredi) systému realneho casu.
- Odhad zatizeni RTS
- Specifikaci a metody navrhu testovací procedury RTS
- Metody a prostredky pro implementaci RTS s cilem splneni i pozadavku fail tolerant
- Paralelni HW architektury s OS a programovacimi jazyky, podporujicimi paralelni zpracovani uloh, omezeni zdroju a casove horizonty zpracovani uloh.

7.6 Operacni systemy realneho casu RTOS

7.6.1 Úvod

Pod pojmem řidici pocitac (RP) budeme rozumet jak řidici pocitac nebo minipocitac, tak řidici mikropocitace distribuovaneho řidiciho systemu nebo i vestaveny mikropocitac v dane aplikaci. Vsechny tyto RP potrebují ke sve cinnosti programové vybaveni. Toto vybaveni lze rozdelit na operacni system RP a uzivatelske programy.

Operacní systém je soubor programu, které spolu s dalšími vlastnosti výpočetního systému tvoří základ pro pracovní režimy výpočetního systému. Zejména vykonávají řízení a dohled nad základními programy. (DIN 4430).

Důležité k pochopení operačního systému je pojem zdroj. Zdroje jsou objekty, které využívají programy při svém běhu. Programy musejí na tyto prostředky (zdroje) čekat, pokud jsou zdroje obsazeny.

Ke zdrojům patří především procesory, paměti, periferní jednotky, databáze, ale i systémové programy a procedury. Operační systém se dá definovat také jako řídicí program, který přiděluje konkurujícím programům zdroje výpočetního systému. Jedná se tedy o systematicky vybudovaný soubor řídicích a pomocných programů. Uspodňuje uživateli práci při tvorbě a běhu programu. Přebírá na sebe kupř. ochranu přístupu ke zdrojům a komunikaci mezi programy.

Zvláštní požadavky jsou kladeny na operační systémy reálného času (RTOS), které jsou základním systémovým vybavením RP v automatizaci, kde stojí v popředí jako nejdůležitější požadavek současné provádění většího počtu úloh, jako odezvy na události v řízeném technickém procesu. Za této situace se stává základní požadavek funkce OS pro obecné účely, totiž plně efektivní využití výpočetních zdrojů, zcela sekundárním. Dalším důležitým požadavkem na RTOS je to, že musí reagovat na vnitřní i vnější události ve stanoveném čase. Událost (event) je podmínka, která nastala uvnitř výpočetního procesu nebo mimo něj a vyžaduje speciální reakci.

V automatizačních úlohách je často třeba provádět programy cyklicky nebo se startem v určitém čas. okamžiku.

Základním pojmem v oblasti RTOS je pojem úloha (task). Task zpravidla obsahuje reakce systému na jednu nebo více události (events). Dalším, dost důležitým pojmem je pojem paralelního běhu. Na obr. 7.4 je znázorněn vztah mezi programy a úlohami (tasks) a událostmi, které přicházejí z procesu a spouštějí výpočetní procesy.

P je program, který může být zpracováván ze 2 úlohami (tasks), A1, B1 jsou data, E1 a E2 jsou události, přicházející z technického procesu, TA a TB jsou úlohy (tasks) a A2 a B2 jsou výstupní data. Současný, kvaziparalelní běh více úloh, které využívají stejné nebo různé programy se nazývá multitasking.

Výpočetní proces je časový průběh zpracování sekvencí programu na jednom procesoru.

Obr. 7.4 Programy, udalosti, ulohy (tasky) v multitaskingovém systému

Program je statický objekt, který sestává z množství příkazů, které jsou uloženy v paměti. Na druhé straně je třeba odlišit dynamické provádění programu, které popisuje výpočetní proces. Toto odlišení je důležité zejména pro ty systémy, které obsahují programové komponenty, které jsou schopné znovu vstupovat do výpočetního procesu. Jeden exemplář programového kódu pak může v stejném čase provádět výpočet ve více úlohách (tasks).

Je také třeba rozlišovat mezi vyvoláním výpočetního procesu a vyvoláním podprogramu. Při vyvolání podprogramu je pozastaven hlavní program až do té doby, dokud neskončí běh podprogramu. Naproti tomu při vyvolání výpočetního procesu, pokračuje jak vyvolaný proces tak základní program vedle sebe „současně“, přičemž tento běh je řízen operačním systémem.

Výpočetní proces neexistuje jen během provádění příkazů, ale i během plánované nebo vynucené čekací doby. Začíná požadavkem na seznamu úloh v organizační části programu a končí zrušením v tomto seznamu úloh. Výpočetní proces může být tedy také definován jako operačním systémem RT řízený (průběh) zpracování sekvencního programu. V automatizaci se vyskytuje často případ, že určitý program se vykonává cyklicky nebo se startuje k určitému časovému okamžiku. V těchto případech je třeba naplánovat příslušný výpočetní proces odpovídajícími příkazy pro takové chování s časovou závislostí.

Výpočetní proces se nazývá plánovaným, jestliže jsou pro něj pomocí odpovídajících příkazů přirazena taková opatření, že se v závislosti na splnění určité podmínky nebo uběhnutí jistého času aktivuje.

Nejdůležitějším požadavkem, kladeným na RTOS je to, že musí reagovat jak na vnitřní, tak na vnější udalosti ve stanoveném časovém intervalu. Udalost (event) je podmínka, která nastala uvnitř výpočetního procesu nebo mimo něj a vyžaduje speciální reakci prostřednictvím nějakého programu RP. Je dobrou programátorskou praxí strukturovat programy RT systému, tak, že se sdružují související udalosti do jednotlivých skupin udalostí. Task (úloha) implementuje zpravidla ty reakce RP, které jsou odezvou na jednu udalost nebo skupinu udalostí.

Dochází často ke konfliktům, když různé výpočetní procesy usilují o jeden výpočetní zdroj. Pak nastupuje další funkce RTOS organizovat přidělení zdroje s cílem včasného ošetření udalosti a paralelního běhu různých zdrojů RP.

Úkoly operačního systému reálného času (RTOS) jsou rozsáhlejší, než úlohy OS pro obecné užití a dají se specifikovat následujícím přehledem:

1. Ošetření prerusení

2. Sprava vypocetniho procesu
3. Komunikace mezi vypocetnimi procesy
4. Synchronizace procesu
5. Sprava casu
6. Provedeni vstupnich a vystupnich operaci.
7. Man-machine-komunikace
8. Sprava dat a databazi
9. Sprava pameti
10. Osetreni chyb
11. Osetreni nestandardnich stavu

Krome toho by mel RTOS rovnez podporovat programovani uzivatelskych programu a jejich testovani, tedy pripravu behu uzivatelskych programu.

Aby RTOS byly schopny zajistit tak komplexni ulohy, jsou tyto deleny na menci celky, takze schema RTOSu muze vypadat dle obr. 7.5.

Obr. 7.5 Model architektury RTOS.

Porovnejme tento model se dvema modely OS pro obecne uziti. Obr. 7.6 a 7.7.

Obr. 7.6 a 7.7 Z Benetta

RTOS je podobne jako OS pro obecne vyuziti vybudovan na striktnim dodrzovani vysokeho stupne abstrakce, ktery rika, ze vnejsi vrstva vyuziva sluzby jen nejblize nizsi (vnitrnejsi) vrstvy modelu, aniz by znala konkretni mechanismy implementovani a ukladani do pameti. To umoznuje postupne testovani vrstev od nejnizsi smerem nahoru a udrzuje tak v pripustnych mezich funkce OS (Wartung und Pflege).

7.6.2 Pozadavky na včasnost

Pozadavek na včasnost (vcasnou reakci) znamena, ze vstupni data museji byt ziskana v casovem limitu musi byt proveden vypocet a vystupni data musi byt dana na vystup.

Vcasnost se da posuzovat dvema ruznymi casovymi podminkami:

- absolutni casovou podminkou , kupr. V case 10:45 hod. musi byt vydan signal k odjezdu vlaku
- realtivni casovou podminkou , kupr. Signal k prepnuti rychlosti dopravniku musi byt dan do 10 sec. po dosazeni mezni hodnoty v nasypce ap.

V nasledujici Tab. 7.3 jsou uvedeny priklady ruznych typu casovych podminek v automatizacnich ulohach.

Vykonani funkce	Absolutni cas. podm.	Relativni cas. podm.
k pevnemu okamziku	sejmuti hodnoty na zkusebnim zarizeni	chemicka analyza
v casove toleranci	sejmuti regulovanych velicin	mereni hodnot s klouzavymi limity
ke konci casoveho pasma	prijeti datovych zprav	rozpoznani caroveho kodu zbozi
na zacatku casoveho pasma	kaskadni rizeni davkovych procesu	prijeti signalu ze svetelnych zavor

tab. 7.3.

Z teto tabulky se daji vysledovat 4 pripady casovych podminek:

1. Funkce (kupr. Mereni vstupni veliciny nebo vyslani vystupni veliciny) se musi provest presne v casovych okamzicich t_1, t_2, \dots
2. Funkce se musi provest v danem casovem intervalu.
3. Funkce se musi provest nejpozdeji do urcitého casoveho okamziku.
4. Funkce se smi provest nejdrive po urcitem casivem okamziku.

V casovem diagramu na obr. 4.3 nabyva poradnice hodnot 0 nebo 1 dle toho zda funkce je nebo neni splnena. Jako priklady pripadu 1. az 4. se daji uvest nasledujici priklady z automatizacni praxe:

Ad1) Kupr. Funkce testovaci stolice pri zkouseni motoru predpoklada, ze v urcitych casovych okamzicich se uskutecni jiste funkce. Zda pujde o absolutni casovou podminku nebo relativni casovou

podmínku závisí na tom, zda se toto určení odvozuje od vnitřních hodin (absolutní časová podmínka) nebo od vnější události.

Ad2) Nejčastější případ technické praxe. Jde o periodické snímání dat s pevnou vzorkovací frekvencí. Její toleranční pásmo musí být nastaveno tak, aby vzorkování uvnitř tohoto pásma ještě postáčila k řízení procesu z hlediska stability a předepsané kvality.

Ad3) Vyskytuje se zejména u kusových procesů, u kterých se jisté funkce musí vykonat dle pevného časového harmonogramu nebo podle objektu, které se vyskytnou v bodech měření situace na výrobní lince.

Ad4) V sekvencních procesech, kde se čeká na splnění podmínky přechodu do dalšího stavu.

7.6.3 Pozadavky na současnost

Pozadavek na současný běh výpočetních procesů je odvozen z toho, že RT systém musí reagovat na podněty z okolí a tyto podněty mohou probíhat současně. Proto je nutné v RP současně zpracovávat několik výpočetních procesů. Řešení paralelního běhu výpočetních procesů je v zásadě dvojího druhu. Jednak skutečný paralelní běh, kdy výpočetní procesy probíhají každý na jiném HW, druhá možnost je kvaziparalelní běh více výpočetních procesů na jednom procesoru. Tato druhá možnost má úzkou souvislost s rychlostí řízeného technického procesu, neboť současnost musí být chápána právě ve smyslu současného běhu výpočetních procesů vzhledem k řízenému procesu. V tom druhém případě se hovoří o simultánním běhu a zpracování procesů.

7.6.4 Typické vlastnosti operačních systémů reálného času (RTOS)

Porovnejme navzájem některé vlastnosti OS pro obecné použití a RTOS

Správa procesoru :

a) OS pro obecné účely:

- spravedlivé přidělování zdrojů jednotlivým úlohám (tasks)
- OS dbá o nejefektivnější využití zdrojů

- OS dbá o vykonání (proběhnutí) všech úloh

b) RTOS:

- dosažení paralelního běhu pokud možno všech zdrojů
- podpora úlohy s vysokou urgencí
- vysoké průměrné zatížení zdrojů nemá prioritu, prioritou je naopak včasné ošetření události
- vzájemná ochrana úloh různých uživatelů ztrácí na významu, neboť RP je zpravidla určen pro použití jednoho uživatele
- jsou pozadovány mechanismy synchronizace a komunikace mezi úlohami (tasks)

7.6.5 Sprava pameti

Jsou dva různé aspekty spravy pameti ve výpočetním systému. Ten základní je ve společném využití hlavní pameti různými úlohami. Ten druhý, důležitější, spočívá ve výměně programového kódu mezi hlavní pameti a pomocnými pamětmi tak, aby se vytvořilo virtuální prostředí, které poskytuje v podstatě více prostoru v hlavní pameti, než jí fyzicky skutečně je. Tato pamet se nazývá virtuální pamet

Virtuální pamet je na jedné straně praktická věc, protože se uživatel nemusí starat o velikost pametového prostoru, na druhé straně se za to platí časovým zpožděním při přetahování dat a programu mezi různými pametovými úrovněmi. V případě multitaskového provozu toto časové zpoždění vzhledem k současnému provádění několika úloh nemusí mít velký význam.

Pohled na spravu pameti se výrazně změní, pohlédneme-li na to z hlediska jednoho úkolu. Čas potřebný pro přesun dat a programu z různých pametových úrovní prodlužuje provedení úlohy (task) a zpomaluje tak dobu reakce úlohy. Kromě toho vede na neprůhledný tok dat k neodhadnutelným časovým odezvám, což v případě tvrdých systémů reálného času je velmi nepříjemné. Upřednostňovaná strategie přidělování pameti v real-time systémech je proto spíše statické přidělování pametového prostoru úkolům, kombinovaná s možností přemístování na základě explicitního požadavku příslušného úkolu.

Existuje velký počet vestavených systémů průmyslové automatizace, které používají převážně ROM pameti a využití RAM je určeno staticky. V takových případech není žádná sprava pameti potřeba. V každém případě je sprava pameti poměrně jednoduchá a druhoradá záležitost.

7.6.6 Sprava přístroje a zařízení

Nejdůležitějšími periferními přístroji u RT systému jsou rozhraní. Jsou zpravidla přiřazena jednotlivým uživatelským procesům a těmi jsou také přímo ovládána. Standardní periférie jsou standardním způsobem obsluhované operačním systémem. Mnoho vestavených systémů jako on-board computer, CNC a další nepotřebují vůbec žádné standardní periférie. Protože u vestavených systémů nejsou většinou k dispozici klávesnice a hard disky, ztěžuje se tím významně způsob kódování a oprava chyb v uživatelských programech. To se řeší nasazením dvou různých počítačových systémů ve dvou fázích vývoje systému řízení:

- vývojový systém bezící na počítači běžného typu (PC), který poskytuje vývojové prostředí pro vývoj a ladění uživatelských programů
- cílový mikropočítačový systém, do kterého se nahraje odladený program a ten pak řídí aplikaci.

Sprava programu (File management)

Sprava programu RT systému se příliš neliší od stejné úlohy u OS pro obecné použití.

Předcházející diskuse ukázala, že převážně rozdíly pro vývoj a užití RTOS je v oblasti spravy procesoru, t.j. v manipulování s časem a událostmi, přidělování procesoru, synchronizaci a komunikaci. Další problémy jsou buď druhorade povahy nebo se významně neliší od způsobu řešení s operačními systémy pro obecné použití. Proto se tento oddíl bude zabývat především otázkami spravy procesoru. Situace se dále komplikuje v případě, že RP není chapán jako jeden RP, ale distribuovaný systém reálného času.

7.6.7 Principy organizace zpracování úloh

Principy současnosti a včasnosti jako základní vlastnosti RTOS platí samozřejmě také pro programy. Proto musí tvorby programu vyhovovat následujícím požadavkům:

1. Řídící programy technického procesu musí být provedeny v určitém definovaném čase. Tyto časové intervaly mohou být buď předem dány nebo ne.
2. Jestliže RP musí zpracovávat současně různé úlohy, měl by to vykonávat na několika procesorech. Jestliže je RP vybaven jen jedním nebo jen několika málo procesory a nepřichází tedy v úvahu rozdělení úloh tak, že každá úloha je vykonávána na jednom procesoru, musí RP pracovat kvasisimultánním způsobem.

Pro splnění požadavku na včasnost a současnost aplikacních programů jsou známy dva způsoby:

1. Synchronní zpracování úloh (synchronní nebo sériové programování)
2. Organizování časového průběhu během provádění programu. To se nazývá asynchronní nebo paralelní programování.

První způsob je starší, ale stále se ještě používá v některých časově kritických aplikacích jako kupříkladu v radarovém sledování leticích cílů a v letecké dopravě. Metoda vychází z toho, že jednotlivé úlohy (tasks) v řízení procesu jsou navzájem nezávislé. Mnohé výpočetní procesy navíc jsou periodické povahy, zatímco jiné jsou odezvou na náhodné události z procesu. Jestliže je možné v takových aplikacích nalézt nebo určit horní hranici doby provedení výpočetního procesu, je možné staticky plánovat časový průběh provádění výpočtu prostřednictvím explicitního přidělení času. Časový průběh je synchronizován časovými značkami, které jsou realizovány prerušovacími signály od časovače. Jestliže všechny úlohy mají stejnou periodu zpracování, bude se každá úloha nacházet v každém časovém rámci, definovaném daným počtem taktu (prerušovacích signálů od časovače). Ve svém časovém intervalu proběhne celý úkol až do konce a má exkluzivní přístup ke globálním datům. Je vytvořeno pořadí úloh (úkolů) a toto pořadí úkolů je použito k synchronizaci dat, které se předávají z jednoho do druhého. Prostřednictvím prerušovacích signálů se mohou na konci časového rámce a před začátkem dalšího rámce aktivovat výpočetní procesy vázané na události. Události jsou identifikovány buď mechanismem prerušení nebo dotazovacím režimem (pooling). V posledním případě se kontrolní program po ukončení periodických programů uvnitř časového rámce dotazuje na statut jednotlivých periférií. Vyše popsaný režim činnosti může být popsán obrázkem obr. 7.8, který zobrazuje časový průběh 3 regulačních úloh, které jsou řešeny na principu synchronizovaného programování.

Obr. 7.8 Casový průběh 3 regulačních úloh, které jsou řešeny na principu synchronizovaného programování

Řídící program je pak aktivován hodinami a může vypadat následovně:

Viz program v EPOSU na str. 130-131 PZA I (Halang).

Nevýhodou tohoto postupu je to, že při pevné periodě ramce se někdy provádí jen několik, jindy všechny úkoly. To se dá odstranit víceúrovňovou strukturou časových ramců. Jednotlivý ramec v horní úrovni sestává z pevného počtu ramců nejbližší nižší úrovně. Na obr. 7.9 je znázorněna 2 úrovněová struktura úkolu, obsahující hlavní úkol jako horní úroveň a podúkol jako spodní úroveň. Pokud se celý úkol, patří do hlavní úrovně (s nižší frekvencí opakování) nedá provést v časovém ramci, který odpovídá frekvenci dolní úrovně (podúkol), je rozdělen na více částí a vykoná se v průběhu více ramců dolní úrovně (úkol C se provádí ve dvou cyklech podúkol a úkol D ve 3 cyklech dolní úrovně vzhledem k tomu, že ve druhém cyklu bylo nutné vykonat úkol E reagující na náhodnou událost). Protože přidělení úkolu hlavní úrovně (hlavní úroveň), je poměrně složité, provádí tuto synchronizaci řídicí program centrálně. Tyto programy určují dle seznamu úkolu, který úkol se bude provádět ve kterém časovém ramci. Základní struktura cyklického dvouúrovňového řídicího programu je na obr. 7.9

Obr. 7.9. *synchronní průběh periodických a prerušovacích výpočetních procesů*

Základní struktura cyklického řídicího programu pro dvě úrovně s automatickým dělením úloh je na obr. 7.10

Obr. 7.10 ze str. 132 Halang

Úkoly, které jsou vykonávány ve frekvenci ramců minor, jsou nazývány minoritní procesy, úkoly, prováděné s frekvencí hlavní pak major. Seznam úkolu obou typů je pak označován jako major list nebo minor list. Zkratkou PSW se pak označuje stavové slovo programu a má vztah k obsahu registru procesoru (PC, A a další), které se účastní každého výpočetního procesu.

Aby délka ramce odpovídala potřebám řízení daného procesu, musí ramce splňovat následující požadavky:

- délka dílců ramců (minor) musí být větší než je maximum součtu času všech úloh nižší úrovně, aby tam zbyl ještě čas na provedení případného ošetření události (event)
- délka hlavních ramců (major) musí být větší, než je suma času pro provedení reakce na event a suma všech úloh, které se mají provádět s frekvencí cyklu major
- velikost intervalu, rezervovaná pro úkoly typu event a dále pořadí úkolu na seznamu úkolu musí zaručit, že budou dodrženy požadované lhůty na provedení reakce na event (událost)

Synchronní způsob je efektivní metoda k organizování sekvence, pokud procesy nemusí čekat na nic jiného než na výpočetní čas a jsou stále připraveny. Jestliže úkol musí v některém rámci čekat na přidělení zdroje nebo výskyt nějaké události, měl by dát svůj čas k dispozici procesoru a až v dalším rámci tento čas využít. Jestliže jsou podmínky čekání úkolu v rámci ještě komplikovanější, musí být všechny tyto možné kombinace řešeny v programovém kódu úkolu. To vede k příliš velké komplikovanosti. Navíc musejí být tyto podmínky prozkoumány v každém minoru, což vede k neproduktivnosti práce procesoru.

V případě, že synchronní způsob programování více úloh pro řízení technického systému (toto synchronní provádění úloh se používá tam, kde úlohy nemusejí čekat na přidělení zdroje, jako u PLC, kde jde o RT systém s cyklickým vykonáváním úloh a program je uložen v ROM a podobně je tomu u průmyslových regulátorů, video-rekorderů ap.), když tedy tento cyklický způsob programování selhává, je nutné, aby přidělování zdroje bylo prováděno na asynchronním principu. Jinak totiž se ztratí mnoho času, když úloha, která se provádí musí čekat na přidělení zdroje. Kromě toho by se při používání synchronního způsobu zpracování úloh musela měnit programová struktura při každé změně v některé z úloh (tasks).

Proto v aplikacích s komplikovanými souvislostmi mezi časem, událostmi, výpočetním procesem, zdroji musí operační systém (OS) dynamicky a asynchronně organizovat plánování a provádění úloh. Základem metody je pak současný běh více úloh, řízený víceúlohovým operačním systémem. Jestliže úkol musí čekat na přidělení zdroje nebo výskyt události, je jeho provádění pozastaveno (suspend) a procesor začne vykonávat jiný úkol. Provádění suspendovaného úkolu je automaticky obnoveno, jakmile jsou splněny podmínky, kvůli kterým byl úkol pozastaven (suspend).

7.6.8 Multitasking

Současné provádění výpočetních procesů může být implementováno dvojitým způsobem. Při použití víceprocesorového systému se jednotlivé úlohy mohou provádět na jednotlivých procesorech. V opačném případě, jak je to ukázáno kupř. na obr. 7.11 musejí se jednotlivé úkoly ucházet o přidělení výpočetního prostředku a času. Protože zpravidla je vždy větší počet úkolů, než procesoru i ve víceprocesorovém systému, obr. 7.11 má obecnou platnost. Životní cyklus úlohy (task) se da nejlépe popsat stavovým diagramem úkolu. Trebaze operační systémy reálného času a programovací jazyky reálného času často používají komplikovanější stavové diagramy úkolu, dají se stavy úkolu popsat diagramem z obr. 7.12

Obr. 7.12 Stavový diagram ulohy (task) (R – ready, B- wait, S- suspend, L-

Stav T - terminated V tomto stavu se nalezá task po provedení a před provedením. Je veden na seznamu uloh (tasks), který je spravována OS. Není splněna plánovaná podmínka pro jeho provedení.

Stav R - ready Čeká již jen na přidělení procesoru. Podmínka pro jeho provedení je splněna.

Stav E - running Procesor provádí programový kód ulohy (task).

Stav S – suspended Je pozastaveno provádění ulohy z důvodu nesplnění nějaké podmínky nebo proto, že mu není přidělen zdroj.

Je důležité podívat se na základní rozdíly mezi stavy S (Suspend), T (terminated) a R (ready) na jedné straně a stavem E (running) na straně druhé. Ty první 3 stavy reflektují vztah mezi ulohou a okolím tasku, stejně jako stupeň připravenosti tasku pro stav E. Task ve stavu T nemůže běžet, musí čekat, protože není splněna nějaká podmínka. Task ve stavu R (ready) by mohl běžet, to již nezáleží na jeho stavu, ale na tom, zda procesor bude volný.

V obr. 7.12 jsou uvedeny operace (tasking operations), které způsobují přechod tasku z jednoho do druhého stavu. Operace Start převádí čekající task do stavu ready. Operace Run způsobuje aktivaci tasku do stavu E (running). Inverzní operaci k operaci Run je operace Preempt, která převádí běžící task do stavu R (ready). Operace suspend převádí task ze stavu E do stavu S, odkud se do stavu R může dostat operací resume. Konečné operace Stop a abort převádějí task ze stavu E nebo S a R do stavu T (terminated).

Při použití metod asynchronního programování se snažíme o splnění podmínek včasnosti a současnosti běhu programu bez plánování „off line“. Tasky budou volány resp. aktivovány v libovolném časovém okamžiku. Protože tasky nebudou nijak předem synchronizovány, nebude také zřejmé, kdy se který task bude provádět. Bude to záviset na časovém okamžiku příchodu prerušovacích signálů a tedy ne staticky ale dynamicky.

V obecném případě bude vždy docházet k situacím, kdy je více tasků ve stavu Ready a dochází tudíž ke konfliktu. To, který task se bude vykonávat závisí na zvolené strategii. Nejjednodušší a v praxi nejčastější strategie spočívá v tom, že každý task má podle své důležitosti přiřazenou prioritu (číselná hodnota), podle které se řadí pořadí tasků v případě konfliktu. Tedy tasky s vyšší prioritou jsou vykonávány před tasky s prioritou nižší. Tato strategie bude ještě v dalším ukázaná. Zejména nás bude zajímat způsob, který, pokud je to vůbec možné, tak řadí tasky (v případě konfliktu), že všechny jsou provedeny včas. Ideálním kritériem pro tento způsob plyne z důkladné analýzy časové náročnosti výpočtu jednotlivých tasků. Nejjednodušším způsobem je to, že se dříve provádí ten task, který má nejkratší termín.

„Současnost“ provádění tasku je dosažena tím, že operační systém prerušuje běh tasku a že celá rada tasků je v daném okamžiku v různém stupni rozpracovanosti. Aby se zabránilo chybám při operaci s objekty, které jsou společně mnoha taskům, není možné tento způsob aplikovat bez omezení. Kromě toho řízené technické procesy si samy vyžadují jisté pořadí vykonávání tasků. Velkou roli v „současnosti“ provádění tasků hraje synchronizace.

Metoda asynchronního programování má následující vlastnosti:

- Pozadavky na včasnost bude splněna jen přibližně. Časové podmínky, které mají souvislost s prioritou budou splněny tím lépe, čím vyšší je priorita daného tasku.
- Skutečný časový průběh se může vůči požadovanému časovému průběhu tak posunout, že se tasky navzájem mohou předbíhat.
- Fronta tasků není deterministická, nýbrž se organizuje dynamicky, t.j. podle příchodu prerušovacích signálů se sestavuje dané pořadí tasků. Při programování se proto nedá předem přesně udát, který task bude ke kterému okamžiku vykonán.

Zmeny stavu se uskutečňují prostřednictvím událostí, které se vyskytnou uvnitř nebo mimo výpočetní proces, tedy někde v automatizovaném prostředí. Události (event) se dají rozdělit do 4 tříd:

Externí události vznikají v okolí řídicího počítače. Jsou RP poskytovány zpravidla prostřednictvím vstupního podsystemu. Mohou být ale generovány programem (kupř. programem, který testuje stav čidla).

Casové události odpovídají průběhu specifických časových intervalů (např. Zpoždění). Tyto události jsou poskytovány správou časových událostí operačního systému, který je periodicky generuje na základě přerušeni od časovace.

Vnitřní události odpovídají chybám a odchylkám, které se vyskytnou v průběhu provádění programu. Tyto události mohou být indikovány přerušovacími signály, generovanými HW řídicího počítače (hlášení o dělení nulou od matematického koprocesoru) nebo jsou generovány programem (hlášení o dělení nulou, generované aritmetickou rutinou v pohyblivé radové čarce ap.).

Programové události odpovídají speciálním podmínkám, které se vyskytnou v úloze (tisku) během jejího provádění. Tyto události jsou generovány operačním systémem jako reakce na specifické požadavky.

Multitaskovací operační systém sestává z systémového jádra (kernel), který provádí všechny operace správy úlohy a události a z množství systémových procesů (shell), které poskytují množství služeb operačního systému. Na obr. 7.13 běží systémové procesy (shell) stejným způsobem jako uživatelské procesy. Rozdělení funkcí mezi kernel a shell se liší u jednotlivých operačních systémů (OS), každopádně základní operace s úlohou realizuje kernel OS. Dale se v kernelu provádí správa paměti prakticky u všech OS.

Logická struktura kernelu je na dalším obr. 7.14. Kernel je vlastně exekutiva OS. Na obrázku 7.14 je tedy model jedné možné exekutivy reálného času.

Z obr. 7.14 je zřejmé, že exekutiva je spuštěna buď přerušovacím signálem nebo vnitřní událostí. Blok monitoru (rozpoznání) rozpozná povahu signálu a vyvolá odpovídající rutinu pro zpracování události. Reakcí na událost představuje zpravidla jeden nebo více přechodů z jednoho do druhého stavu, např. Zadržení chybné úlohy nebo aktivování úlohy, které jsou svázány s příslušnou událostí. V posledním kroku úloha scheduler (rutina přidělování úlohy) volá výpočetní proces, který provede výpočet a předává mu řízení.

Spodní blok kernelu sestává z množiny systémových tabulek, které popisují jednotlivé úlohy. Dale obsahuje množinu rutin, které provádějí přechod úlohy z jednoho do druhého stavu. Uvnitř exekutivy (kernelu) je každý úloha popsán tabulkou stavů TST (task state table), která obsahuje údaje o úloze jako stav, priorita, zdroje, které úloha vyžaduje. Těto tabulky se někdy říká také řídicí blok úlohy.

Rutiny přechodu úlohy z jednoho stavu do druhého implementují operace Start, Abort, Suspend a resume, stejně jako rutinu Select, která vybírá úlohu s nejvyšší prioritou. Operace Stop je jen zvláštním případem operace Abort. Operace Run a Preempt jsou implementovány blokem task scheduler (plánovač).

Operace přechodu úlohy z jednoho stavu do druhého jsou popsány prostřednictvím tabulky stavů jednotlivých úloh. Pro urychlení sledu operací jsou tyto tabulky organizovány jako seznamy. Základní seznamy jsou:

Task directory (TD) – adresar (seznam) uloh, který všechny ulohy, registrované v systému a umožňuje identifikátoru úlohy být přeložen jako adresa v TST.

Ready list (RL) – obsahující všechny úlohy, které jsou ready.

Další seznamy pak obsahují úlohy, které čekají z nějakého důvodu (suspended). Jsou vytvořeny i seznamy podle druhu příčiny čekání úlohy (čekání na zdroj, čekání na uplynutí časového intervalu ap.).

Nyní si všimneme tabulky stavu úlohy (TST) a tabulky přechodových subrutin (STS) teď dále. Minimální množina údajů, které musí být uloženy v TST jsou tyto:

State – kód stavu úlohy

Priority – priorita úlohy

PSWo – počáteční stav stavového slova programu (Programm Status Word)

PSW – aktuální stav PSW

Algoritmy, popisující podprogramy Start, Abort, Suspend a Resume a rutinu plánovače (Scheduler) jsou na obr. 7.15 (7.7). Inicialy RT použité v obr. 7.15 pro běh úlohy se vztahují k systémové proměnné, která uchovává identifikátor právě probíhajícího úlohy. Stavové kódy T,S,R odpovídají značení v obr. 7.12. Není žádný speciální kód pro běh úlohy, který je identifikován pomocí proměnné RT.

Je třeba říci, že výše uvedený příklad je silně zjednodušený. V praktických případech je množství údajů v TST mnohem větší a algoritmy jsou komplikovanější. Dodatečné požadavky popisují potřebu jednotlivých úloh na čas a zdroje. Tato data jsou spravována prostřednictvím tabulky a pomocí vztahu mezi úlohami a zdroji. Kupř. operace Abort musí uvolnit všechny zdroje, které úloha, která je podrobena operaci Abort dosud využívala a dát tyto zdroje k dispozici frontám úloh, čekajících na tyto zdroje.

Jestliže úloha má být dynamicky vytvářena v prostředí diskového systému, musí operace Start vytvořit TST (task state table), rezervovat prostor hlavní paměti a nasmontovat programový kód úlohy ze spodní části paměti do rezervovaného prostoru hlavní paměti.

Dalším problémem, který má vztah k implementování systémového jádra je problém, jak chránit kernel proti poškození ať již náhodnému nebo úmyslnému. Problém může být radikálně řešen tak, že omezen striktně na svou část paměti a nemá možnost sahát na jiné segmenty paměti. To se dá řešit HW úrovní prostřednictvím obvodu správy paměti, který řídí a kontroluje přístup procesoru a omezuje je jen na určitý úsek paměti. Při pokusu o zápis do jiné části paměti dojde k přerušování od obvodu „ochrany paměti“.

Procesor může pracovat ve dvou režimech:

- uživatelském
- operátorském

Příkazy, které se týkají obvodu správy paměti jsou privilegované a mohou být tedy vyvolány jen při práci procesoru v operátorském režimu. Je-li procesor v uživatelském režimu, může provádět uživatelské programy. Přepínání režimu procesoru z jednoho do druhého režimu se obecně děje přerušovacími signály. V uživatelském režimu se dá přepnout zpět ale také programově. Další privilegované příkazy slouží k řízení periférií a přerušovacího systému.

Mechanismus ochrany paměti zabranuje úlohám (tasks) sice přístup k systémové datové struktuře, neresí však plně vylučný (exclusive) přístup k systémovým datům. Podívejme se proto

na prerusovací signal, který přišel, když kernel OS pracuje. Jestliže bylo povoleno prerusení, mohla být exekutiva prerusena a mohla být spuštěna jiná úloha. To je jev zvaný „zdvojený kernel“, kdy se pracuje současně se stejnými daty. V jednoprocessorovém systému se tento stav řeší jednoduše: prepínání pomocí prerusovacích signalů se musí zabránit v době, kdy kernel pracuje. V mnohprocessorovém systému je situace složitější, neboť ve stejném okamžiku dva nebo více procesorů chce používat programový kód exekutivy. K tomu jsou nutné mechanismy rezervování exkluzivního přístupu určitého procesoru ke společné paměti.

V nejjednodušším případě může být exkluzivní přístup ke společné datové struktuře realizován HW mechanismy pro rezervování systémové sběrnice (Bus lock a Bus unlock). Je to však velmi neefektivní způsob, neboť tím jsou omezeny další procesory v práci se společnými částmi systému a nejenom k této části datové struktury, která se musí rezervovat. Lepší způsob vytváří koncepcí binárního semaforu, který může chránit jev vybraný přístroj nebo vybranou datovou strukturu.

Binární semafor je proměnná se dvěma stavy, která leží ve společné oblasti paměti. Ukazuje, zda k semaforu přiřazený zdroj (prostředek) je volný nebo obsazený. Jednoduché algoritmy pro semaforové operace jsou na obr. 7.16 (Fig.7.8)

Obr. 7.16

Operace Reserve (B) zkoumá semafor B a suspenduje procesor, když zdroj (prostředek) je obsazen (B=0). Operace Release (B) neboli uvolnění, označuje, že prostředek je volný (B=1). Algoritmus pro rezervovací operace používá aktivní formu čekání ve frontě a využívá HW mechanismus rezervování systémové sběrnice. Systémová sběrnice (system bus) je tak rezervována jen po co nejkratší čas. Problémy synchronizace paralelních procesů budou detailně vysvětleny v další kapitole.

7.6.9 Synchronizace a komunikace

V multitaskingovém systému běží uživatelské programy vedle sebe (současně). Skutečný stav sledu provádění úloh závisí na událostech, které startují ten který task a tento sled je tudíž nepředvídatelný. To však nesmí vést k nepředvídatelným výsledkům. K tomu slouží mechanismy synchronizace za účelem:

- Realizace odpovídajících vztahů mezi tasky
- Realizace logického sledu úloh
- Zabránění chybných operací se společnými objekty

Rozlišujeme 2 formy synchronizačních problémů v paralelních procesech:

- Asymetrické : komunikace mezi úlohami (kupř. Producer-Consumer Problem při sberu a zpracování měřených veličin)
- Symetrické : problém vzájemného vylučování (mutual exclusion problem) při exkluzivním přístupu více tasků ke společnému zdroji

Při vysvětlování metod asynchronního programování bylo zdůrazněno, že jde vlastně o paralelní programování. Je to tím, že vše má souvislost s programováním paralelních „procesů“.

Definujme:

Dvě akce ve výpočetním procesu se nazývají „paralelní“, když mohou probíhat současně.

Dvě akce se nazývají sekvencní, když jsou řazeny do určité následnosti.

Dále se rozlišuje mezi vnější paralelitou (vedle sebe běžící procesy) a vnitřní paralelitou (simultánní běh). Vnější paralelita je způsobena paralelně běžícími technickými procesy, zatímco vnitřní paralelita (simultánní běh) povede k zvýšení výpočetního výkonu.

Definujme dále:

Dvě akce ze dvou různých výpočetních procesů se nazývají paralelně běžícími, když mohou probíhat současně.

Dvě akce jednoho výpočetního procesu se nazývají simultánními, když mohou být prováděny současně.

Prostřednictvím požadavku na provedení výpočetních operací k danému časovému okamžiku v určitých časových intervalech nebo při výskytu určitých událostí jako např. Alarmová hlášení z technických procesů, má být zajištěno, že výpočetní procesy běžící „synchronně“ t.j. v odpovídajícím časovém sledu a minimálním skluzu s událostmi v technickém procesu. Při řízení „současného“ provedení výpočetních procesů prostřednictvím operací. Systému metodou asynchronního programování nedá se jistému časovému skluzu oproti požadovaným okamžikům zabránit. Tím se může stát, že poradi provedení úkolu nemusí odpovídat v jistých okamžicích sekvenci chodu technického procesu. Tím by mohlo dojít k nesprávnému řízení technického procesu. To především v tom případě že průběh výpočetního procesu má bezprostřední vliv na funkci technického procesu.

Prozatím byla diskutována jen logická závislost výpočetních procesů, která je odvozována z průběhu technického procesu.. Vedle toho jsou také závislosti, které vyvstávají v souvislosti s využíváním společných zdrojů.

Jako uzavřená logická smyčka může být posuzován následující případ, kdy se navzájem blokují dva výpočetní procesy, protože oba čekají na uvolnění jednoho přístroje. Nebezpečí logické smyčky musí být odstraněno prostřednictvím časové koordinace obou procesů. Takové časové koordinování paralelní, volně běžících procesů se nazývá synchronizace. Synchronizace výpočetních procesů má stejný význam jako koordinace výstupu těchto procesů.. Existují dvě hlavní formy synchronizace:

Logická synchronizace, t.j. synchronizace orientovaná na synchronizaci úloh nebo procesů

Jde o napasování průběhu výpočtu na průběh postupu v technickém procesu. Tak se dostane definována časová posloupnost akcí, t.j. synchronizace mezi technickým a výpočetním procesem. Tato synchronizace znamená jak splnění požadavku co se týče časové posloupnosti akcí, ale také zohlednění předem zadáných časových okamžiků nebo časových intervalů a reakcí na přerušeni z technického procesu.

Synchronizace, orientovaná na výpočetní zdroje.

Jde o dodržení podmínek co se týče využití společných zdrojů. Přitom se da rozlišovat mezi HW a SW prostředky (zdroje), mezi zdroji, které se dají použít vícekrát a zdroji na jedno použití, mezi zdroji s exkluzivní a obecným použitím ap.

Pro realizaci obou forem je k dispozici množina různých metod jako semafore, kritické oblasti, nebo Rendezvous, které mohou být vytvořeny uživateli (programátory uživatelských programů). Všem těmto metodám je společné, že výpočetní proces musí čekat, dokud nepříjde nějaký signál resp. nenastane nějaká událost. Tedy synchronizace různých výpočetních procesů je dosaženo zavedením čekacích podmínek na odpovídajících místech.

Základní prostředky pro synchronizaci úloh (task) jsou vytvořeny v kernelu OS. Jemnější operace mohou být implementovány systémovými tasky a knihovnami vyšších programovacích jazyků.

V tomto odstavci bude podán přehled základních pojmů synchronizace tasku, aniž by se probíraly metody jejich implementace.

7.6.10 Problematika synchronizace

Na příkladu dvou úloh (task), které běží v automatizačním systému ukážeme nyní princip jejich synchronizace. První task sleduje stav proměnné, porovnává její aktuální hodnotu s předem definovanou hodnotou a produkuje chybové hlášení, když je ta hodnota překročena. Druhý task čeká na chybové hlášení a předává ho po pomalé seriové lince na ovládací panel. Obe funkce musejí být vykonávány dvěma různými tasky, neboť průměrný čas pro provedení jednoho hlášení je mnohem delší, než požadovaná perioda sledování procesní veličiny.

Jeden pokus, jak řešit tento úkol je na obr. 7.16.

Viz obr. Fig. 7.9 str. 139 kopie kniha Halang - Saha

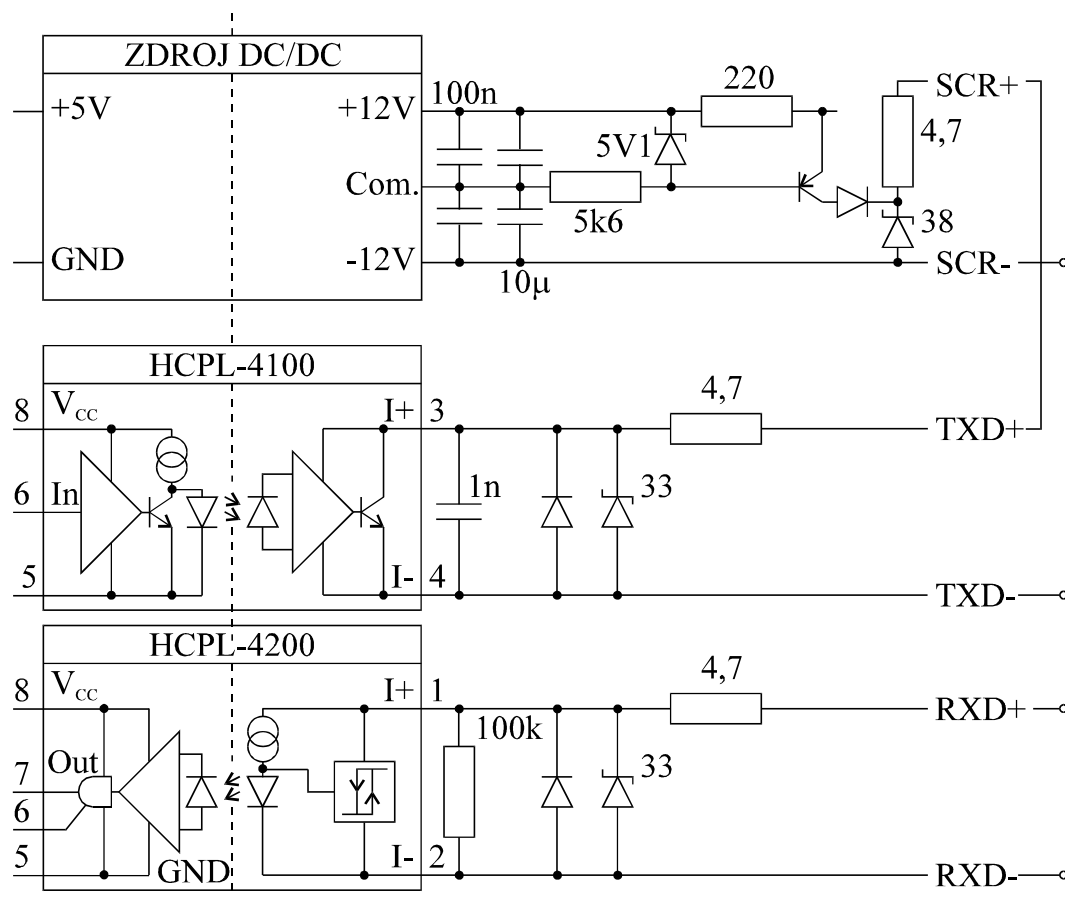
Obr. 7.16. Chybné provedení synchronizace tasku v jednoduše automatizační úloze

8. KOMUNIKAČNÍ SYSTÉMY PRO ÚČELY AUTOMATIZACE

8.1 Úvod do komunikačních systémů pro účely automatizace

V oblasti průmyslové automatizace se v posledních 10 až 15 letech výrazně změnila architektura řídicího systému. Vývoj se posunul od centralizované architektury, reprezentované řídicími počítači a minipočítači k distribuovaným systémům. Pro ty je typické, že výpočetní výkon řídicích členů je dostatečně mohutný i na nejnižší úrovni řízení. Dále platí i to, že inteligence řídicího systému proniká až přímo do procesu. Prostředkem k tomu jsou jednak inteligentní čidla a akční členy, ale i distribuované inteligentní svorkovnice a odloučené karty vstupů a výstupů. Bez nároků na všeobecnou platnost tohoto pojmu, budeme pod inteligentní procesní instrumentací rozumět akční členy a čidla, která mohou komunikovat s nadřazeným řízením (PLC, PC, mikropočítači, zobrazovacími terminály ap.) pomocí sériového komunikačního kanálu nebo jsou ještě navíc vybaveny mikrořadičem pro zpracování signálu (signal processing) nebo realizaci řídicích a regulačních algoritmů (viz kap.3, obr.3.3).

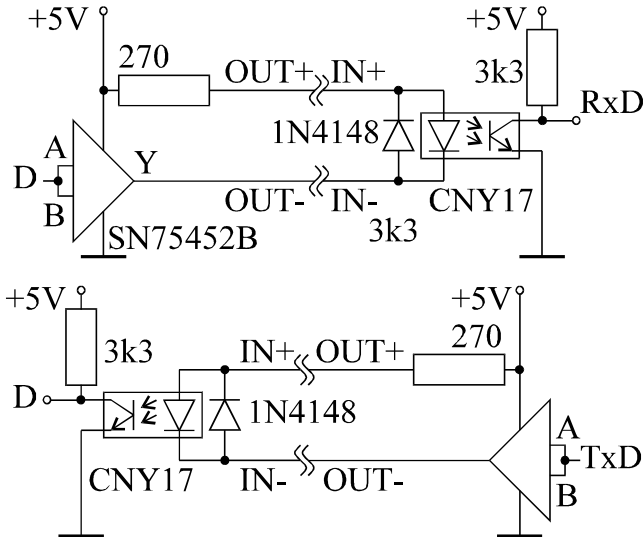
Důvodem pro tento vývoj byla především snaha zlevnit kabeláž mezi procesní



Obr. 8.1a: Proudová smyčka 0-20mA s galvanický oddělením a napájením ze strany vysílače

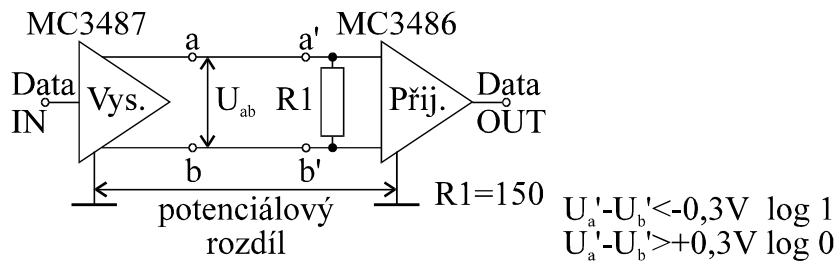
instrumentací a řídicími členy, zkrátit čas a zjednodušit projekt kabeláže pro její instalaci a kontrolu. Sériová komunikace rovněž umožňuje dálkové ovládání této instrumentace (změnu parametrů čidel, kontrolu a kalibraci ap.) a celkově lépe přizpůsobuje procesní instrumentaci současnému stavu mikroelektroniky. Je skutečností, že s ohledem na příznivý vývoj mikroelektroniky, není důvodu nevybavit instrumentaci mikrořadiči (cena mikroelektroniky v inteligentní instrumentaci nehraje dominantní roli), umožňujícími předzpracování procesních dat (filtrace a další "signal processing"). Pro takové řešení je pak realizace sériového komunikačního kanálu přirozeným doplněním jeho funkce.

Převažujícím způsobem propojení na úrovni procesu byla dosud proudová smyčka 0 až



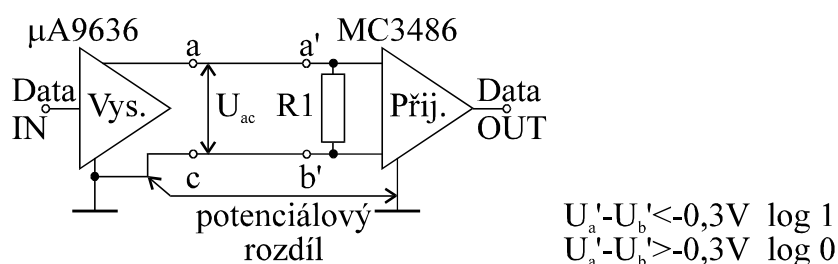
Obr.8.1b: Jiné zapojení proudové smyčky 0-20mA

20mA, schematicky zobrazená na Obr.8.1.

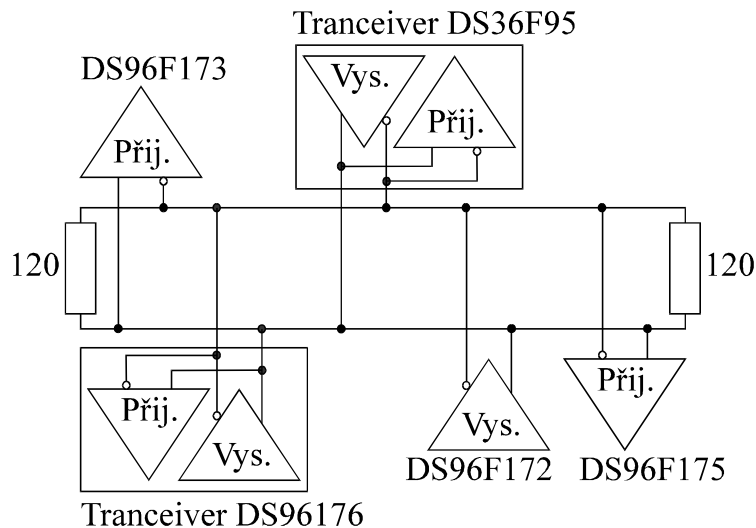


Obr. 8.2: Rozhraní RS422

Její největší předností byla odolnost proti rušení při malých rychlostech a vzdálenostech až stovky metrů. Nevýhodou je jen dvoubodové propojení (tedy ke každému čidlu a akčnímu členu je nutné instalovat dvoudrátový spoj) a malá rychlost přenosu signálu, která vyhovovala sice pro dvoubodový spoj, nebyla by však dostatečná pro sériový mnohobodový spoj. Dodnes se však u některých aplikacích používá pod označením TTY. Ve většině případů však byla z výše uvedených důvodů vytlačena napětovými rozhraními, která se obecně vyznačují vyššími přenosovými rychlostmi a jednodušším zapojením při tvorbě mnohobodového spoje a jak potvrzuje praxe i vyšší odolností proti rušení. Nejprve šlo opět o dvoubodový sériový interface RS 232C a krátce na to o mnohonásobně výkonnější RS 422 (symetrický spoj, plný duplex) a RS 423 (asymetrický duplex). Jako další úspěšné řešení se prosadilo rozhraní RS 485, umožňující dvou vodičový nebo 4 vodičový plný duplex a vysokou rychlost přenosu. Na obr. 8.2 je uvedeno zapojení rozhraní RS 422, na obr.8.3 rozhraní RS 423 a na obr.8.4 zjednodušené zapojení RS 485. Na obr.8.5 pak jsou uvedeny grafy závislosti přenosové rychlosti jednotlivých řešení na

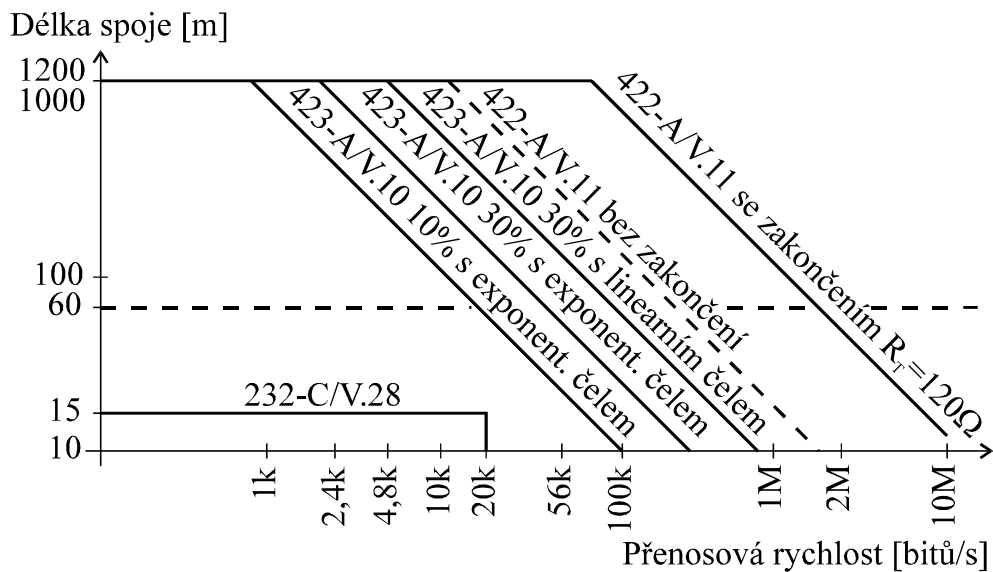


Obr. 8.3: Zapojení RS423



Obr. 8.4: Zapojení RS485

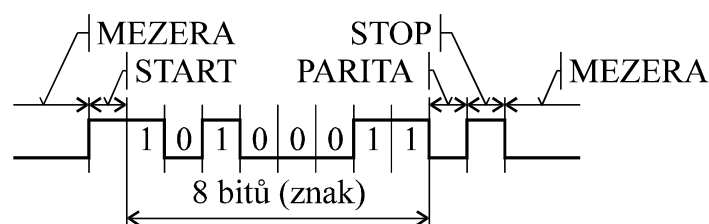
délce spoje [10], [5].



Obr. 8.5: Srovnání komunikačních standardů

Přenosová rychlost pro rozhraní RS422 je až 10MBitů/sec.a doporučuje se pro propojení až 10 účastníků. Používá se hlavně pro propojení počítače (řídícího členu) s více perifériemi jako obrazovky, tiskárny, ale i PLC a průmyslové regulátory. Rozhraní RS 423 je asymetrické s nižší přenosovou rychlostí do 100kHz. Přijímač a vysílač musejí být již výrobcem definovány jako pár. Rozhraní RS 485 je z principu definováno jako multi-user interface. Umožňuje propojit až 32 účastníků. Na rozdíl od RS 422 a RS423 používá třístavových tranceiverů.

Zatímco napětěová rozhraní se používají především pro číslicový přenos (přenos dvouhodnotového napětěového signálu), proudová smyčka sloužila jak pro přenos analogového signálu 0 - 20 mA nebo 4 - 20 mA, tak pro přenos digitálního signálu (dálnopisná smyčka). Přenos dálnopisem může být příkladem nejen pro fyzický přenos signálu, ale i pro jeho kódování. V souvislosti s dálnopisným přenosem byl stanoven jak způsob zakódování, tak způsob sestavení dálnopisné zprávy a její zabezpečení. To představuje vlastnosti komunikačního kanálu, které budou v dalším vysvětleny ve své obecnosti na složitějších způsobech přenosu v souvislosti s ISO/OSI modelem. Na obr. 8.6 je zobrazena zpráva v dálnopisném kódu.



Obr.8.6: Dálnopisný kód

Jde o t.zv. znakově orientovaný asynchronní přenos, kdy jeden rámeček je tvořen 7 nebo 8 významovými bity, vyjadřujícími kupř. hodnotu měřené veličiny, vyjádřené v bitech nebo 8 bitech nějaké zprávy. Dále se zpráva skládá ze START bitu, STOP bitu a paritního (zabezpečovacího) bitu. Přijímací strana musí vědět, že zpráva přichází. Přijímač pak odfiltruje první bit s log. hodnotou 1 a přijme 7 nebo 8 významových bitů zprávy. Dle charakteru parity kontroluje

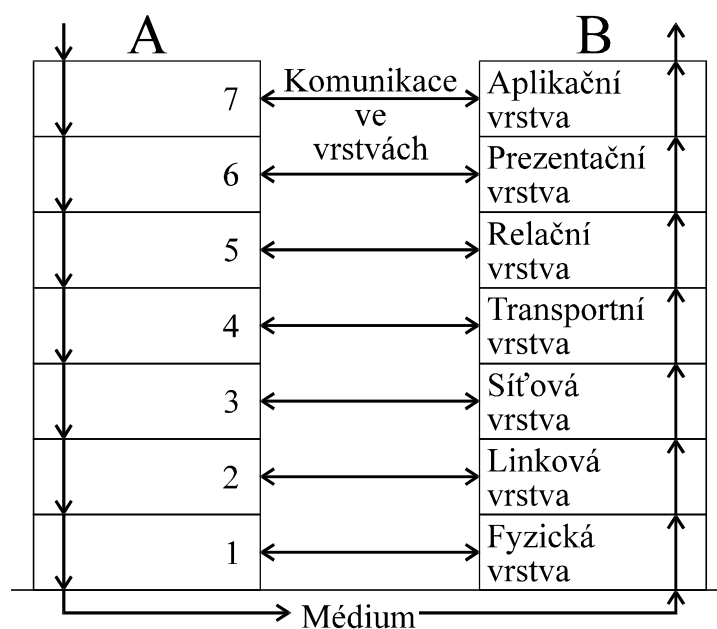
korektnost došlé zprávy (sestavající kupř. z těch 8 bitů, tedy kupř. jedna hodnota analogového signálu, změřená 8 bitovým A/D převodníkem). Po příchodu koncového bitu s hodnotou 1, zpráva končí. Pokud přijímací strana vyhodnotí správně paritu, zpráva je korektní. Přijímač čeká na další začátek dalšího rámce. Vzhledem k tomu, že u dálkopisu šlo vždy o dvoubodový spoj, není třeba vysílat adresu ani přijímací ani vysílací strany. Pokud však nejsme spokojeni se zabezpečením zprávy paritou (při nevhodné kombinaci chyb kontrola paritou selhává), musíme s přijímací stranou dohodnout způsob kontroly přijímané zprávy. Tato dohoda (protokol) je soubor podmínek, které souvisejí s přenosem zprávy a jejím přijímáním. Tak kupř. součástí protokolu může být prosté opakování zprávy ještě jednou. Je-li stejná, je správná. Není-li stejná, musí být vysílací strana informována, že musí vyslat zprávu ještě jednou ap. Jsou mnohem pokročilejší způsoby (protokoly), jak zajistit korektnost zprávy a budou zmíněny na jiném místě.

8.2 Problematika digitální komunikace

Po tomto úvodu do problematiky digitální komunikace můžeme přistoupit k vysvětlení dalších pojmů [1], [2]:

- paket - část přenášeného datového rámce, který lze přenést najednou [2]
- rámec - paket, doplněný o záhlaví, adresaci, zabezpečení, zakončovací posloupnost, příp. další informační znaky

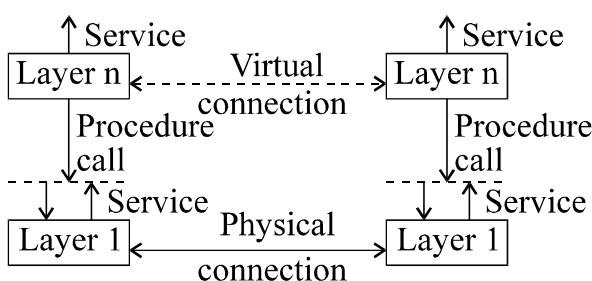
Komplexně se problematikou digitální komunikace zabývá OSI ISO model, neboli model otevřených komunikujících systémů, definovaný mezinárodní standardizační organizací ISO již v r. 1984 jako norma ISO 7498. Tento model definuje podmínky, při jejichž dodržení mohou různí účastníci přenosu spolehlivě komunikovat navzájem mezi sebou. Protože se jedná o model



Obr.8.7: ISO/OSI model

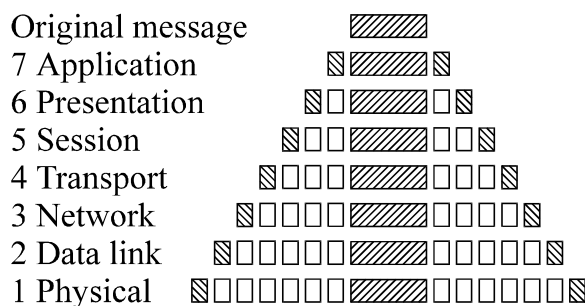
obecného přenosu zpráv, není sám o sobě jednoduchý. Jeho struktura je patrná z obr.8.7.

Obrázek znázorňuje, jak se tvoří zpráva, kterou odesílá účastník A účastníkovi B a zároveň, jak účastník B tuto zprávu přijímá. Vlastní přenos se uskutečňuje prostřednictvím fyzického spoje mezi dvěma nebo více účastníky přenosu. Důležité je, že oba účastníci (peers) přenosu mezi sebou tvoří na každé úrovni modelu virtuální spoje, zatímco reálný přenos dat je samozřejmě pouze v 1. fyzické vrstvě, která obsahuje m.j. rozhraní mezi účastníkem a přenosovým médiem. Virtuální spoj znamená, že účastníci, komunikující spolu na úrovni 7. vrstvy, t.j. kupř. elektronickou poštou, nemají zdání o funkcích vrstev 1 až 6. Každá vrstva má definovány dvě základní funkce. První jsou služby té vrstvy a druhá funkce je protokol vrstvy. Protokol je soubor pravidel, kterými se komunikace mezi účastníky přenosu řídí, t.j. jak lze zahájit přenos, jak ho provést a jak ho ukončit. Dosah každé vrstvy modelu je minimalizován na jednu vrstvu nejbližší vyšší a nejbližší nižší vlastní entity (účastníka) a na hierarchicky stejnou vrstvu jiného nebo jiných účastníků přenosu. Z dalšího obr.8.8 je zřejmé, jak probíhá přenos zpráv mezi jednotlivými vrstvami.



Obr. 8.8: Přenos zprávy mezi vrstvami OSI modelu

Při vysílání zprávy, vyšší vrstva volá procedurou "call" vrstvu nejbližší nižší a naopak směrem nahoru poskytuje nižší vrstva svoje služby vrstvě vyšší. Uvnitř vrstvy se pak uskutečňuje funkce vrstvy a vlastní protokol té vrstvy. Při průchodu zprávy od 7. vrstvy až po vrstvu fyzickou, dochází k nabalování jednotlivých dat vlastního protokolu na originální zprávu. Tento proces je zobrazen na obr.8.9



Obr.8.9: Fyzická tvorba paketu

Je to maximalistická možnost, která nebývá využívána v uvedené míře. Ve většině případů není nutné využívat funkci všech vrstev modelu, takže dochází k redukci (SW) kupř. vrstev 3 a 4 a 6 a 7. Obecně se dá říci, že vrstvy 1 až 3 jsou vrstvy svázané s vlastním komunikačním procesem, zatímco vrstvy 5 až 7 úzce souvisejí s aplikačním SW. Transportní 4. vrstva tvoří přechod mezi těmito podsystemy.

8.3 Vrstvy ISO/OSI modelu

8.3.1 Fyzická vrstva

Z hlediska komunikačního modelu ISO/OSI fyzický přenos dat zajišťují fyzické vrstvy komunikačních účastníků sítě. Fyzická vrstva, uskutečňuje vlastní přenos zprávy formou elektrického (optického, radiového) signálu. Fyzická vrstva rovněž představuje rozhraní mezi fyzickým spojem a linkovou vrstvou a mj. zajišťuje kódování zprávy do formy změn napěťových (nebo proudových) impulsů, dekódování, případně modulování a demodulování a synchronizace takto binárně kódované zprávy. Nedílnou součástí fyzické vrstvy je definice mechanických parametrů fyzického propojení jako např. definice rozměrů konektorů, průřezy vodičů, atd.

Specifikace fyzické vrstvy tedy určuje mechanické a elektrické vlastnosti komunikačního rozhraní. Součástí specifikace fyzické vrstvy tedy jsou zejména:

1. definice rozměrů a vlastností konektorů,
2. popis kabeláže a topologie (materiál, provedení kabelu, kapacita, průřez vodičů, stínění, ...),
3. popis způsobu kódování bitů (RZ, NRZ, Manchester, ...)
4. napěťové (proudové, frekvenční, ...) určení logických úrovní.

8.3.2 Linková vrstva

Linková vrstva zajišťuje zejména následující služby:

- Přístup k médiu
- Přenos ucelených rámců
- Základní zabezpečení proti chybám při přenosu
- Detekci chyb
- Opakované vyslání poškozených rámců
- Potvrzování správně přijatých rámců

Součástí specifikace linkové vrstvy bývá definice minimálních a maximálních prodlev mezi rámci a další časové parametry definující komunikaci. Jednou z hlavních funkcí linkové vrstvy je stanovení tzv. přístupové metody. Je-li na síti více účastníků, je nutné nějakým způsobem stanovit, kdo má kdy vysílat. Dojde-li k situaci, že se několik stanic pokouší vysílat současně, tak se obvykle žádné stanici nepodaří zprávu vyslat, protože současné vysílání

několika stanic způsobí neplatnost údajů na sběrnici. Přístupová metoda je tedy definice pravidel vedoucích k získání oprávnění vysílat zprávu.

V závislosti na přístupové metodě závisí i způsob přenosu informací mezi stanicemi a používaná terminologie. Podle způsobu adresování se přenos dat provádí buď stylem zdroj/cíl, kde je přesně znám odesílatel i příjemce, nebo producent/konzument, kde významný je pouze obsah zprávy a nikoliv odesílatel či adresát.

Zdroj/cíl (Source/Destination)

Přenos informací typu zdroj/cíl je ve své podstatě přenos dat mezi dvěma zařízeními. Informace přenášení ve formě paketů mají jednoznačně určený zdroj (adresa odesílatele) a jedinečný cíl (unikátní adresa příjemce). V okamžiku, kdy je nutné předat stejná data více příjemcům, musí být pro každého příjemce vyslána zvláštní zpráva, což zbytečně zatěžuje přenosové médium. Navíc každý příjemce obdrží tuto zprávu v jiný okamžik, což znesnadňuje synchronizaci mezi stanicemi.

Při přenosu dat způsobem zdroj/cíl může být vztah mezi zařízeními buď typu Master/Slave, nebo Peer-to-Peer.

Master/Slave

Komunikace Master/Slave vychází z principu, že Master je oprávněn vyslat zprávu vždy, když to uzná za vhodné, zatímco Slave smí vyslat zprávu pouze tehdy, když je k tomu vyzván Masterem. Přenos dat ze zařízení Slave do zařízení Master probíhá tak, že Master pošle výzvu a Slave po obdržení této vý-zvy posílá odpověď.

Přímý přenos dat mezi zařízeními typu Slave není obvykle možný. Přenos dat mezi zařízeními Slave probíhá tak, že Master si vyžádá data o jednoho zařízení (Slave A) a vzápětí tyto data pošle druhému zařízení (Slave B).

Na síti je obvykle jedno zařízení Master a několik zařízení typu Slave. Některé sítě a protokoly umožňují existenci více zařízení typu Master na jedné síti (tzv. Multi-master konfigurace).

Peer-to-Peer

Komunikace mezi zařízeními typu Peer-to-Peer vychází z principu, že všechna zařízení jsou si rovna. Tedy všechna zařízení na síti mohou vysílat tehdy, když to uznají za vhodné. Na síti neexistuje pevně vyhrazené zařízení typu Master. Stanice si předávají tzv. Token, což je oprávnění k vysílání. Stanice která vlastní Token je oprávněna vysílat. Ostatní stanice jsou na příjmu. Aby se zabránilo zahlcení sítě jedi-nou neustále vysílající stanicí, je obvykle oprávnění k vysílání omezeno na určitou dobu a toto oprávnění je cyklicky předáváno od jedné stanice k druhé stanici (tzv. token passing).

Producent/konzument (Producer/Consumer)

Při přenosu dat tímto způsobem není významné, které zařízení zprávu vyslalo, ale podstatný je obsah zprávy. Adresa odesílatele je tak nahrazena identifikátorem obsahu zprávy

(tzv. identifikátor zprávy). Všechny stanice na síti "slyší" všechny přenášené zprávy a mají možnost tuto zprávu přijmout. Na základě identifikátoru zprávy se každé zařízení rozhoduje, zda má o data obsažené v dané zprávě zájem a tedy zda zprávu přijme a zpracuje, nebo zda bude tuto zprávu ignorovat. Odesílatel zprávy obecně nemá informaci o tom, kolik je příjemců zprávy. Na odeslání stejných dat více zařízením je tak nutné vyslání pouze jediné zprávy.

Na sítích s přenosem dat typu Producent/konzument lze samozřejmě realizovat jak přenos dat typu Master/Slave, tak přenos dat typu Peer-to-Peer. Z hlediska přístupu k přenosovému médiu jsou si zařízení rovna (není-li protokolem stanoveno jinak) a k vysílání zpráv dochází na základě nějakého podnětu. Podněty k vysílání jsou následující:

- a) Obdržení žádosti o data od jiné stanice;
- b) Nutnost získat data od jiné stanice (stanice tedy odesílá žádost o data);
- c) Došlo ke změně nějaké veličiny a stanice má povinnost každou změnu této veličiny hlásit ostatním stanicím;
- d) Stanice má povinnost periodicky informovat okolní stanice o hodnotě dané veličiny, a od předchozího odeslání informace již uplynul vymezený čas.

Typy zpráv

Přenášené zprávy lze obecně rozdělit podle počtu příjemců na zprávy typu:

Unicast - zpráva má jediného příjemce;

Multicast - zpráva je určena více příjemcům

Broadcast - zpráva je určena všem stanicím na síti.

Vztah Klient/Server

Mnohdy se při přenosu dat používá pojem klient a server. Pojmy klient a server jsou rozlišovány stanice z hlediska poskytování služeb a informací.

Klient - stanice, která vyžaduje službu nebo informace

Server - stanice, která poskytuje službu nebo informace.

8.3.3 Síťová vrstva

Síťová vrstva zajišťuje adresování, specifikuje tedy formát adres a způsob adresování. V některých sítích lze jednu stanici adresovat několika způsoby, je možné definovat skupiny adres (zprávy typu multi-část), nebo společnou adresu pro všechny účastníky sítě (zprávy typu broadcast). V průmyslových sítích, kde topologie jsou poměrně jednoduché, bývá tato vrstva vynechána a její služby poskytuje aplikační vrstva.

8.3.4 Transportní vrstva

Transportní vrstva zajišťuje spolehlivý a bezchybný přenos dat (transport), fragmentaci a defragmentaci dlouhých zpráv, detekci chyb a znovuvyžádání poškozených rámců. Vyšším vrstvám hlásí pouze neopravitelné chyby. V průmyslových sítích, kde jsou protokoly a přenášené zprávy poměrně jednoduché, bývá tato vrstva vynechána a její služby poskytuje aplikační vrstva.

8.3.5 Relační vrstva

Relační vrstva navazuje, vyjednává, udržuje a ukončuje spojení mezi stanicemi. Využívá transport-ní vrstvu pro bezpečný přenos dat, která jsou jí předávána Prezentační vrstvou. V průmyslových sítích, kde jsou přenášeny zprávy poměrně jednoduché a pro přenos každého rámce je navázáno samostatné spojení, bývá tato vrstva vynechána a její služby poskytuje aplikační vrstva.

8.3.6 Presentační vrstva

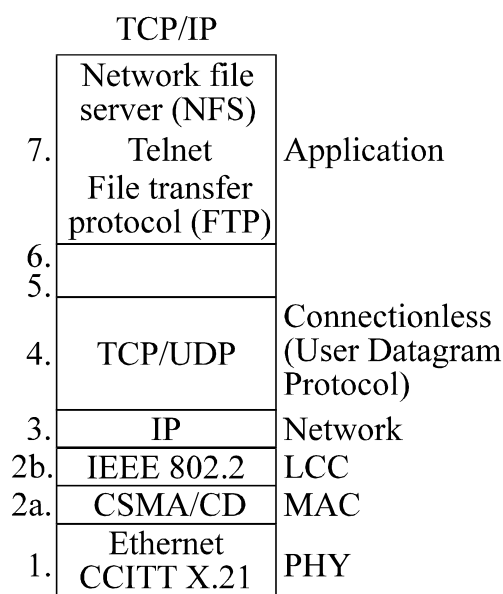
Presentační vrstva převádí přenášená data do tvaru vhodného pro zpracování Aplikační vrstvou. Převádí data do vhodného formátu (např. převod mezi big-endian/little-endian), rovněž může zajišťovat kryptografické služby (šifrování a dešifrování) při zabezpečené komunikaci. V průmyslových sítích bývá tato vrstva vynechána a její služby poskytuje aplikační vrstva.

8.3.7 Aplikační vrstva

Aplikační vrstva zajišťuje poskytování komunikačního rozhraní aplikaci. Pro aplikaci tedy zajišťuje přenos dat mezi stanicemi, definuje význam přenášených zpráv a přenášených dat, definuje datové typy a specifikuje reprezentaci reálných fyzikálních veličin definovanými datovými typy. Aplikační vrstva rovněž zajišťuje služby těch nižších vrstev, které nebyly protokolem implementovány.

ISO/OSI model není jediným možným přístupem k účinnému komunikačnímu systému, je však značně rozšířen jak v oblasti rozlehlých komunikačních systémů (pro které byl vyvinut), tak v oblasti LANů, včetně LANů pro výrobu (systémových busů, fieldbusů a nižších průmyslových sběrnic SAN).

Jako příklad jiného, velmi rozšířeného komunikačního protokolu lokálních sítí LAN je protokol TCP/IP, jehož model [1] je uveden na obr. 8.9a.



Obr. 8.9a: Model protokolu TCP/IP

Je vhodný pro heterogenní sítě LAN a je svázán s LANem typu Ethernet. Protokol TCP (Transmission Control Protocol) odpovídá protokolu 4. vrstvy ISO/OSI modelu, zatímco IP (Internet Protocol) odpovídá protokolu 3. vrstvy modelu ISO/OSI. Protokol TCP/IP pak nevyužívá vrstev 5. a 6. modelu ISO/OSI a ve vrstvě aplikační používá výše uvedené protokoly (Telnet, FTP).

9. PERSPEKTIVY STÁVAJÍCÍCH STANDARDŮ FIELDBUSŮ A PRŮMYSLOVÉHO ETHERNETU

9.1 Ethernet

Ethernet je sériová sběrnice vyvinutá na konci 70. let firmou Xerox. Vychází ze specifikace IEEE 802.3 pro fyzickou vrstvu a doplňuje ji o specifikaci linkové vrstvy, zejména horní podvrstvy definující LLC (Logical Link Control - řízení logického spoje). Dolní podvrstva linkové vrstvy, MAC (Medium Access Control), definující způsob přístupu k přenosovému médiumu, je charakterizována jak u Ethernetu, tak u IEEE 802.3 nedeterministickou přístupovou metodou CSMA/CD (Carrier Sense Multiple Access/Collision Detection).

Zatímco IEEE 802.3 specifikuje několik variant fyzické vrstvy, Ethernet využívá původně jen jedné specifikace s přenosovou rychlostí 10 Mb/s, přenos v základním pásmu, délka segmentu je do 500 m, topologie sběrnice a padesátiohmový tlustý koaxiální kabel. K této původní variantě Ethernetu se již delší dobu využívají i další varianty fyzické vrstvy, přizpůsobené tenkému koaxiálnímu kabelu a kroucené dvoulince. V poslední době se dále objevuje rychlý (Fast) Ethernet, u něhož řešení fyzické vrstvy umožňuje rychlost 100 Mb/s. Teprve tyto vysoké rychlosti a úpravy topologie sítí Ethernet předurčují tento systém také pro průmyslovou komunikaci.

Lokální sítě LAN, pro které je Ethernet původně určen, vesměs nevyužívají pro otevřenou komunikační strukturu referenční model RM ISO/OSI, nýbrž jednodušší síťový model TCP/IP s příslušnými protokoly definujícími způsob přenosu dat. V ekvivalentu RM ISO/OSI by tyto protokoly představovaly třetí síťovou (IP) a čtvrtou transportní (TCP) vrstvu.

Ethernet dále nespecifikuje také interpretaci obsahu datových souborů. Kupř. u průmyslových sítí je součástí specifikace 7. vrstvy mj. definice datových typů. Podle této definice je např. analogová veličina vždy reprezentována datovým typem s pohyblivou řádovou čárkou podle normy IEEE 754, což každý účastník dané sítě správně interpretuje. Další rozdíl spočívá v tom, že zatímco průmyslové sítě jsou navrženy především pro nasazení v průmyslovém prostředí jako sítě menšího rozsahu, Ethernet je využíván v sítích rozsáhlejších a umožňuje bezproblémové napojení na Intranet/Internet.

9.2 Průmyslové sítě a Ethernet

Při veškerém nadšení pro Ethernet je nutné vidět to, že průmyslové sítě na jedné straně a Ethernet TCP/IP na straně druhé jsou dvě značně odlišné věci. Z toho důvodu nelze v nejbližších

létech očekávat vytlačení průmyslových sítí Ethernetem. Ve srovnání se zavedenými průmyslovými sítěmi může Ethernet vykazovat tyto výhody:

1. dlouhodobě ověřené technologie (Ethernet, TCP/IP, http, ftp, ...)
2. kompatibilita s dalšími lokálními sítěmi (LAN) a s Intranetem a Internetem
3. vyšší přenosová rychlost ve srovnání s průmyslovými sítěmi (Profibus, Device Net, ...)
4. při použití přepínaného Ethernetu možnost duplexního režimu, defacto zdvojnásobení přenosové rychlosti
5. jednoduché a levné připojení na PC, Internet\Intranet
6. masová výroba síťových komponent se odráží v nízké ceně, velká podpora různých médií
7. vývojoví pracovníci většinou mají s technologií TCP/IP již značné zkušenosti

Mezi nevýhody Ethernetu ve srovnání s průmyslovými sítěmi je možné uvést:

1. nedeterministický přístup k médiu. To je způsobeno použitím přístupové metody CSMA/CD. Pokud zařízení při přístupu na sběrnici detekuje kolizi, čeká náhodný časový interval než se pokusí o přístup znovu.
2. délka datového pole není přizpůsobena potřebám průmyslové komunikace
3. použití aktivních prvků (switche, routery, ...) v síťové topologii. Oproti jiným sítím využívajících pasivní sběrnice je toto řešení dražší a z principu náchylnější na poruchu.
4. neukončený vývoj (protokolů), a tím další náklady na vývoj
5. nutnost vyvinout průmyslové verze konektorů, kabelů a dalších síťových prvků. Je nutné mít k dispozici prvky vhodné pro vyšší rozsahy teplot, větší úroveň rušení a zajišťující vysokou spolehlivost provozu (redundance).
6. není vhodný pro připojení jednoduchých (levných) sensorů a aktorů. Cena připojení jednoho čidla k ethernetu je v současné době vysoká v porovnání s jinými sběrnici. Také se ještě nevyrábějí dostatečně integrované prvky umožňující splnit prostorové omezení.

Mechanismus CSMA/CD je plně postačující v případě informačních technologiím kde příliš nezáleží na pravidelnosti přenosu informace. Řídící aplikace mají ovšem vyšší nároky na determinismus, vyžadují přesně definovanou odezvu systému (například PID regulátor vyžaduje pravidelnou periodu vzorkování). Tento problém lze řešit tím, že se celá síť Ethernet rozdělí pomocí mostů na logické segmenty. Zprávy se pak omezují jen na daný segment, a tudíž se vznikající kolize plynoucí z podstaty náhodného přístupu k médiu v síti Ethernet omezí na daný segment. Pravděpodobnost kolize se výrazně zmenšuje i při značném zatížení sítě. Přepínače (směrovače nebo mosty) přepínají zprávu jen do toho segmentu, kde je adresát zprávy. Zprávy tedy zůstávají jen v nejmenším možném počtu segmentů.

Přepínaná struktura sítě Ethernet řeší tedy do značné míry průchodnost sítě a při extrémně vysokých rychlostech (Fast Ethernet 100 Mb/s) může zaručit režim blízký režimu v reálném čase. Ethernet se tak stává velmi atraktivní variantou k průmyslovým sítím i pro propojení

řídících členů, přístrojů a jisté třídy inteligentní instrumentace (inteligentní čidla a inteligentní akční členy).

9.2.1 Komunikační protokoly průmyslového Ethernetu

Norma IEEE 802.3 definuje pouze dvě nejnižší vrstvy obecného ISO/OSI modelu, fyzickou a linkovou, které nezaručují doručení zprávy (ať už v rámci jedné sítě nebo mezi sítěmi). Pro praktické použití je třeba definovat další vrstvy. V současnosti nejrozšířenější je použití protokolu TCP/IP.

IP (Internet Protocol) – zajišťuje přenos dat mezi jednotlivými sítěmi, tvoří síťovou vrstvu ISO/OSI modelu

TCP (Transmission Control Protocol) – stará se o doručení zpráv, tvoří transportní vrstvu ISO/OSI modelu

Tyto čtyři vrstvy zajišťují bezpečný přenos dat mezi jednotlivými zařízeními v síti. Služeb těchto vrstev využívá nejvyšší vrstva ISO/OSI modelu, aplikační vrstva. Na rozdíl od nižších vrstev, situace se standardizací aplikační vrstvy není jednoduchá. Lze využít služeb používaných v Internetu, jako je HTTP, FTP, SMTP a další. Existuje již několik průmyslových standardů využívajících TCP/IP, další se vyvíjejí. Vedle těchto existují i protokoly využívající pouze Ethernetu, ale již ne TCP/IP.

9.2.2 Technologie známé z Internetu

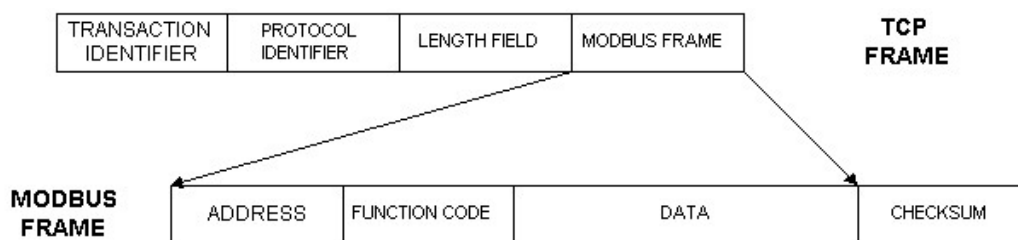
Významnou výhodou Ethernetu je, že lze použít již existujících služeb a aplikací.

- www – lze použít při instalaci, konfiguraci a údržbě zařízení
- ftp – použitelné pro přenos dat v podobě souborů
- e-mail – pro přenos různých hlášení a alarmů

Výhodou je rovněž i možnost přímého připojení na Internet. Lze použít mnoha existujících prostředků, jako jsou webovské prohlížeče a další. Průmyslové komponenty se zabudovaným web serverem dnes již nabízí celá řada firem.

9.2.3 MODBUS® TCP

Protokol MODBUS®, původně vyvinutý firmou Modicon, je dnes široce používán. Definuje strukturu zpráv, která umožňuje navázat spojení typu master/slave mezi inteligentními zařízeními. Protože MODBUS® definuje pouze zprávy, je z principu nezávislý na použité fyzické vrstvě. Tradičně je používán na RS232/422/485. Nejnovější specifikace definuje přenos komunikačních zpráv za použití prostředků TCP/IP protokolu prostým vložením Modbusové zprávy do TCP paketu.



Obr. 9.1 Princip vložení Modbusové zprávy do TCP paketu

Výhodou Modbusu je jeho jednoduchost a otevřenost. Jeho definice je volně přístupná na Internetu. Protože MODBUS® TCP vychází ze starší definice, je velmi jednoduché vyrobit bridge mezi Ethernetem a jiným médiem. Tím se zpřístupňuje celý svět starších zařízení podporujících Modbus. Nevýhodou tohoto protokolu je jeho zastaralost. Modbus byl vytvořen roku 1978 a od té doby prakticky nedoznal změn. Na rozdíl od modernějších protokolů je zaměřen pouze na přenos dat, nijak nedefinuje jejich formát a význam, žádné profily zařízení. Proto firma Schneider připravila rozšíření protokolu (Object Messaging Specification for the MODBUS/TCP Protocol), které definuje obecný datový typ, jeho vlastnosti, metody a strukturu zpráv pracujících s těmito objekty. Definice konkrétních objektů je ponechána na uživateli.

Jak je zřejmé z obr 9.1, Modbus TCP přejímá formát zprávy ze starších variant protokolu. Vynechává se část kontrolního součtu, protože o bezchybné doručení je postaráno prostředky protokolu TCP, navíc je úvodní sekvence šesti bytů. Modbus TCP také s malými výjimkami kompletně přebírá datový model a množinu funkcí definovaných na tomto datovém modelu. Tyto funkce jsou rozděleny do třech tříd. Třída 0 jsou univerzální a zaručeně kompatibilní funkce, třída 2 obsahuje užitečné, ale ne plně kompatibilní funkce. budoucí rozšíření množiny funkcí není vyloučeno. Vždy bude možno určit, zda dané zařízení podporuje určitou funkci prostým požadavkem na vykonání této funkce a otestováním chybového hlášení. Takto bude vždy zajištěna interoperabilita zařízení.

9.2.4 EtherNet/IP

Zkratka IP ve jménu tohoto protokolu znamená Industrial Protocol. EtherNet/IP je vyvíjen společným úsilím organizací Control Net Int. a ODVA. EtherNet/IP je sběrnice založená na klasické technologii Ethernet/TCP/IP, využívající aplikační vrstvu shodnou s protokoly DeviceNet a ControlNet. To zaručuje vysokou úroveň interoperability mezi těmito třemi sítěmi.

EtherNet/IP definuje explicitní i implicitní zprávy. Explicitní zprávy (například konfigurace, informace, off-line data) jsou implementovány pomocí prostředků TCP. Implicitní zprávy (přenos real-time dat) používají běžný protokol UDP (User Datagram Protocol).

Nad touto aplikační vrstvou je ještě definována uživatelská vrstva. Součástí této uživatelské vrstvy jsou definice slovníku objektů. Definují se objekty typu analogový vstup, PID regulátor, skupina objektů a mnoho dalších (a stále přibývají). Spolu se standardními profily zařízení je tak zaručena vysoká interoperabilita zařízení.

První oznámení tohoto protokolu se konalo v březnu 2000. Během léta roku 2000 byly volně (bez poplatku) publikovány specifikace (protokoly, slovníky objektů, ...) spolu s příklady zdrojových kódů. Vzhledem k široké komunitě členů sdružení ODVA a ControlNet Int., kteří

mají bohaté zkušenosti s použitou aplikační vrstvou je velmi pravděpodobné úspěšné rozšíření tohoto protokolu.

9.2.5 FOUNDATION Fieldbus

V roce 1985 začal za účasti ISA a později i IEC vývoj standardní mezinárodní průmyslové řídicí sběrnice. Na tyto práce navázala organizace Fieldbus Foundation (založena roku 1994), která s využitím částí protokolů Profibus a WorldFIP dokončila specifikaci sběrnice Foundation Fieldbus H1 (fyzická vrstva je součástí normy IEC 1158, snaha o zahrnutí linkové a aplikační vrstvy pokračuje).

V březnu 1998 organizace FF ohlásila záměr použít v následující verzi protokolu jako fyzickou a linkovou vrstvu standardní Ethernet s přenosovou rychlostí 100 Mb/s včetně vrstev TCP/IP. Finální verze specifikace Foundation Fieldbus H2 (také HSE – High Speed Ethernet) byla uvolněna 29. března 2000. Verze HSE podporuje všechny funkce předchozí verze, včetně funkčních bloků a DDL (Device Description Language). Tím je umožněno snadné propojení obou technologií. HSE může sloužit jako velmi výkonná páteřní síť, ke které jsou pomocí H1 připojeny koncová zařízení. Koncová zařízení velmi náročná na rychlost lze připojit přímo použitím HSE. Podpora standardních funkčních bloků (jako Analog I/O, PID, ...) byla rozšířena tzv. FFB (Flexible Function Blocks). FFB jsou určeny hlavně pro diskrétní řízení, programovací jazyk je definován standardem IEC 61131.

9.2.6 PROFINET

Organizace PROFIBUS International v říjnu tohoto roku oznámila podrobnosti o chystané technologii PROFINET. Podle slov organizace je PROFINET otevřené, transparentní řešení propojující standardní průmyslové sběrnice Profibus s Ethernetem TCP/IP, které zajistí komunikaci mezi softwarovými aplikacemi na všech úrovních řízení.

Dosažení tohoto cíle má proběhnout ve třech krocích:

Mapování cyklických a acyklických služeb Profibusu na TCP/IP, umožnit přístup k proměnným a konfiguračním a diagnostickým datům, definovat odpovídající softwarový interface pro OPC. Tím se uživatelům umožní vzdáleně monitorovat Profibus zařízení přes Internet/Intranet, data budou dostupná až do úrovně kancelářských aplikací.

Přímé směřování TCP/IP na Profibus, umožnit použití technologií Internetu a softwaru firmy Microsoft přímo v průmyslových zařízeních. Umožnit komplexnějším zařízením přímo využít běžných služeb technologie DCOM.

Přímé propojení Profibusu a Ethernetu/TCP/IP. Komplexní průmyslová zařízení budou přímo integrovat příslušné služby. Jednodušší zařízení budou integrována za použitím OPC proxy-serveru.

ProfiNET bude používat metod COM/DCOM, komunikace bude složena na TCP/IP, OPC, ActiveX a XML. ProfiNET bude také obsahovat editor komponent, který umožní definovat objekty a poskytne prostředky pro komunikaci mezi inteligentními zařízeními. Navíc, použitím proxy služeb bude možno spojit Ethernet a Profibus s mnoha různými zařízeními. ProfiNET bude poskytován bezplatně všem zájemcům. Uvedení prvních produktů používajících ProfiNET se očekává v září roku 2001.

9.2.7 Další firemní standardy

Vedle otevřených protokolů existuje ještě řada proprietárních firemních sběrnic využívajících Ethernet, ať už jen jako fyzickou a linkovou vrstvu nebo spolu s TCP/IP:

- Siemens – systém používaný firmou Siemens nese název Industrial Ethernet. Ve své podstatě se jedná o dříve používaný protokol SINEC H1, přenesený s využitím TCP/IP na Ethernet.
- Allen-Bradley – používá vlastní aplikační vrstvu CSP/PCCC pro komunikaci mezi automaty PLC-5 a SLC-500.
- GE Fanuc – používá TCP/IP s vlastní aplikační vrstvou SRTP (Service Request Transfer Protocol) pro komunikaci mezi automaty GE Fanuc Series 90.
- Mitsubishi – použit protokol TCP/IP s vlastní aplikační vrstvou MELSEC-A nebo MELSEC-Q. Použito pro komunikaci mezi Mitsubishi PLC Series A/Series Q.
- B+R (Bernecker + Reiner) - Obchází TCP/IP a zavádí vlastní protokol Powerlink, který může koexistovat s TCP/IP. Umožňuje cyklus sítě <1 ms pro 30 připojených stanic.
- další proprietární standardy.

9.3 Kombinovaná struktura

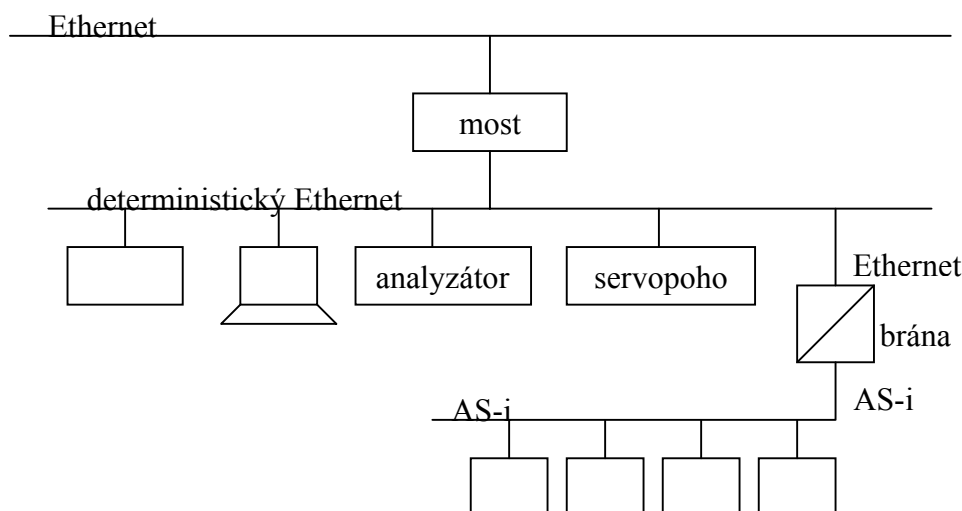
Za velmi přijatelný kompromis v diskusi o průniku Ethernetu do průmyslové komunikace lze považovat kombinaci stávajících jednodušších průmyslových sítí (Sensorbus a Devicebus) s průmyslovým Ethernetem.

Přes všechny své pozitivní rysy nemá v současné době totiž technologie Ethernet šanci se cenově prosadit na úrovni jednoduchých (binárních) čidel a akčních členů. Tuto instrumentaci, která ve velké třídě řízených strojů, linek a procesů má převažující podíl, nelze z cenových ani prostorových důvodů vybavit obvody pro přímé připojení na síť Ethernet (elektronika pro připojení čidla k síti musí stát pod 3 DEM). Pro tyto případy by bylo nutné připojovat na Ethernet multiplexery binárních V/V. Z cenových důvodů však nelze u multiplexeru uvažovat o menším počtu V/V než 16. To však představuje krok zpět v decentralizaci procesní instrumentace a prodražuje dvoubodovou kabeláž od multiplexerů k jednotlivým čidlům a akčním členům. Proto se nabízí vhodná kombinace Ethernetu a jednoduchého sensorbusu, navzájem propojených mezisběrníkovým spojem - bránou. Na obr. 9.2 je jedno z možných řešení s použitím sítě AS-interface.

Avšak ani kombinace stávajících fieldbusů (vyšších průmyslových sítí) s průmyslovým Ethernetem není vyloučena. Sdružení uživatelů a výrobců standardů (Profibus, Interbus, další) reagují následovně. Technologii Ethernet navrhuji pro projení větších zařízení (přístroje a řídicí členy, panely apod.) s požadavky na velký objem přenášených dat , jejichž přenos není časově kritický. Standardní průmyslové sítě vyřeší rychlou výměnu malého objemu dat z akčních členů a senzorů. Nižší sériová sběrnice se musí umět přizpůsobit protokolům TCP/IP. V tom okamžiku má otevřený přístup do prostředí Internetu.

Nástupu Ethernetu do oblastí řídicích systémů se přizpůsobuje i sběrnice Interbus, a to vývojem mezisběrníkového spoje (brána), kdy protokoly TCP/IP budou využívány pro služby

orientované na Internet, zatímco Interbus bude pracovat jako podřízená síť na úrovni akčních členů a senzorů. Podobně reaguje na tento vývoj i sdružení výrobců jednoduché sériové sběrnice typu sensorbus - sběrnice AS-interface, jak je patrné z obr. 9.2.



Obr. 9.2 Architektura s AS-Interface a deterministickým Ethernetem

Brána Ethernet/AS-i může být tvořena kartou v PLC, kartou v PC nebo samostatným mezisběrnicovým spojem.

9.4 Ethernet z hardwarového hlediska

9.4.1 Nejobvyklejší varianty Ethernetu

Množinu nepoužívanějších fyzických médií přehledně uvádí Tab..

10Base-5	10 Mb/s, základní médium je "tlustý" koaxiální kabel 50 ohm, připojení přes AUI konektor, topologie sběrnice, přenos dat - poloviční duplex
10Base-2	10 Mb/s, médium je "tenký" koaxiální kabel 50 ohm, připojení přes BNC nebo AUI, topologie "sběrnice" (daisy chain) nebo hvězda (point-to-point), přenos dat - poloviční duplex
10Base-T	10 Mb/s, médium je nestíněný dvojitý kroucený pár (UTP kabel min. kategorie 3), připojení přes RJ-45 (8 kontaktů z toho 4 jsou využity), topologie hvězda (point-to-point), přenos dat může být plný duplex
100Base-TX	100 Mb/s, médium je nestíněný dvojitý kroucený pár (UTP kabel min. kategorie 5), ostatní jako 10Base-T. V případě potřeby zvýšit odolnost proti rušení (Industrial Ethernet) lze použít stíněný kabel (STP) ve spojení s 9-ti pinovými konektory "D-shell".

1000Base-T	1 Gb/s, médium je čtveřice nestíněných kroucených párů (kabel kategorie 5)
-------------------	--

Tab. 9.1 - Varianty fyzického média

Existuje i Ethernet pracující na optických vláknech (100Base-FX, 1000Base-FX). Ethernetové zařízení může současně podporovat 10Base-5, 10Base-2 a 10Base-T. Uzly podporující 100Base-TX jsou obvykle zpětně kompatibilní s 10Base-T bez zvýšení nákladů na uzel. Uzly podporující 1000Base-T mohou být zpětně kompatibilní s 100Base-TX.

9.4.2 Hardware realizující ethernetové rozhraní 10/100Base-TX

1. Procesor
2. Paměti
3. Ethernetový řadič (Ethernet Controller) - tvorba Ethernetového rámce - linková vrstva ISO/OSI
4. Ethernet Transciever (Ethernet PHY) - fyzická vrstva ISO/OSI - modulace a demodulace signálu
5. Signálový transformátor
6. EEPROM - uložení konfigurace a MAC adresy

9.4.3 Volba procesoru

Procesor je klíčový z hlediska výpočetního výkonu (doba odezvy, práce v reálném čase) a schopnosti adresovat dostatečné množství paměti pro realizaci potokolů. Možnosti jsou 8-bitový procesor, 16-bitový procesor, 32-bitový procesor.

8-bitový procesor je schopen obsloužit dostatečné množství paměti pro realizaci základních protokolů. Odezeva bude relativně pomalá, neboť režie na zpracování protokolů je značná. Lze použít Ethernet s komunikační rychlostí 10 Mb/s i 100 Mb/s, avšak použití 100 Mb/s barianty nemá valný smysl, neboť

8-bitový procesor je schopen generovat nepřetržitý datový tok o velikosti řádově jednotky Mb/s.

16-bitový procesor určitě je schopen obsloužit dostatečné množství paměti a určitě má i dostatečný výkon k zajištění relativě rychlých odezev. V případě střídmych nároků na rychlost lze zajistit práci v reálném čase.

32-bitový procesor poskytne práci v reálném čase, dostatečné množství paměti pro základní i doplňkové protokoly a funkce, zpracování měřených dat v reálném čase pomocí pokročilých algoritmů a metod, které jsou výpočetně náročné.

9.4.4 Zhodnocení perspektiv

V současné době se stále častěji objevují na trhu automatizačních prostředků moduly s přímým připojením na síť Ethernet. To vyvolává bouřlivou diskusi mezi výrobcí, systémovými integrátory a konečnými uživateli na téma, zda a v jakém časovém horizontu se Ethernet TCP/IP stane standardem komunikace i v průmyslovém prostředí. V dohledné době rozhodně nedojde k vytlačení průmyslových sítí a Ethernet TCP/IP se nestane jediným celosvětovým standardem,

neboť přední výrobci automatizační techniky již začali prosazovat své vlastní protokoly využívající Ethernet. Historie se tedy opakuje a v blízké době bude trh opět zaplaven vzájemně nekompatibilními protokoly.

10. PRŮMYSLOVÉ KOMUNIKAČNÍ SÍTĚ

10.1 Fyzický přenos

Při sériovém přenosu dat [21] se informace přenáší bit po bitu. Pokud jsou okamžiky přechodu od jednoho přenášeného bitu k dalšímu od sebe vždy stejně vzdáleny, jedná se o synchronní přenos. Jestliže toto není splněno, jedná se o asynchronní přenos. Při synchronním přenosu je často kromě datového signálu přenášen rovněž synchronizační signál, na jehož základě dochází k synchronizaci mezi vysílací stanicí a jednou nebo více přijímacími stanicemi. Zvláštním případem synchronního přenosu jsou kódy s vnitřní synchronizací, kdy synchronizace je zajištěna signálem, přenášeným data. To se dá zajistit dostatečným počtem přechodů mezi úrovněmi log. "0" a log. "1" během přenosu jednoho znaku. Této podmínce vyhovují některé způsoby kódování signálu (kupř. Manchester) nebo některé přenosové protokoly, které využívají vkládání bitů (t.zv. bit stuffing), jak je tomu kupř. u sběrnice CAN. Při přenosu dat asynchronním způsobem je předávání dat mezi vysílací stanicí a jednou nebo více přijímacími stanicemi obvykle zajištěno externími signály, které např. potvrzují, že data vyslaná na linku jsou připravena ke čtením přijímacími stanicemi nebo potvrzují, že přijímací stanice již data převzala. Zvláštním případem asynchronního přenosu je t.zv. arytmičtý přenos dat, kdy se jednotlivé znaky přenáší asynchronně, ale jednotlivé bity uvnitř znaku jsou přenášeny synchronně. Při arytmičtém přenosu dat se synchronizace pro přenos bitu uvnitř jednoho znaku zajišťuje tak, že každý přenášený znak je doplněn t.zv. rozběhovým prvkem. Za start bitem následují informační bity (datové bity přenášeného znaku) a po nich následuje t.zv. závěrný prvek, nazývaný též stop bit. Ten někdy ještě předchází zabezpečovací - paritní bit. Klidová úroveň přenosové linky mívá při asynchronním přenosu obvykle hodnotu log. "1". Jako start bit se používá přechod z klidového stavu log. "1" do log. "0" (na dobu bitového intervalu). Stop bit bývá log. "1" po dobu jednoho nebo dvou bitových intervalů.

Vlastní fyzický přenos dat se děje prostřednictvím přenosového média, což je v prostředí průmyslové automatizace zpravidla koaxiální kabel, kroucená dvoulinka nebo světlovodič, výjimečně bezdrátový přenos. Nejrozšířenějším médiem bude kroucený dvou vodič. Koaxiální kabel je velmi rozšířený v LANech, umožňuje komunikaci rychlostí nad 10 Mbitů/sec, avšak používá se dost i v průmyslových aplikacích. Třebaže kroucená dvoulinka nevykazuje dostatečnou odolnost proti rušení a rychlost přenosu je nižší, je populární pro svoji jednoduchost a nízkou cenu. Stíněná kroucená dvoulinka je předepsána u některých velmi rozšířených sériových sběrnic (Profibus). Optický kabel je přenosové médium, které se prosazuje a je často kombinováno s ostatními přenosovými médii (viz kupř. řešení sítě SINEC L2OP v systému SIMATIC S5 ap.). Používá se zejména v prostředí silného elektromagnetického rušení.

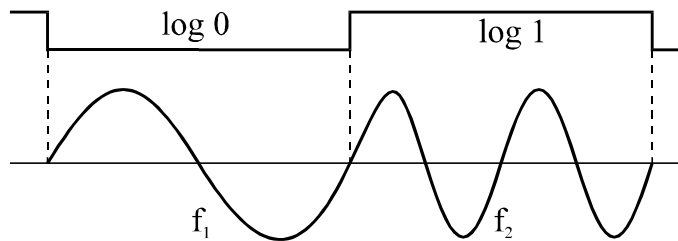
Na fyzickém přenosu dat se dále podílejí fyzické vrstvy komunikujících účastníků sítě. Fyzická vrstva, jak již řečeno uskutečňuje vlastní přenos zprávy formou elektrického signálu a proto musí zaručit rozhraní mezi fyzickým spojem a linkovou vrstvou, kódování zprávy do formy změn napěťových (nebo proudových) impulsů, dekódování, případně modulování a demodulování takto binárně kódované zprávy.

Pro vícenásobnou komunikaci (více než dva účastníci přenosu) se pro přenosové médium dvoudrát používá nejčastěji fyzické rozhraní RS 485, umožňující vysokou rychlost přenosu. (a dvou vodičový poloduplexní nebo čtyřdrátový duplexní přenos dat.)

Zpráva ve formě bitově kódované sekvence impulsů se přenáší následujícím způsobem:

- přímo v základním pásmu
- frekvenčně v základním pásmu
- namodulovaná na nosný vf signál, t.j. širokopásmově

Frekvenční přenos v základním pásmu může být řešen kupř. [1] způsobem z obr. 10.1a.

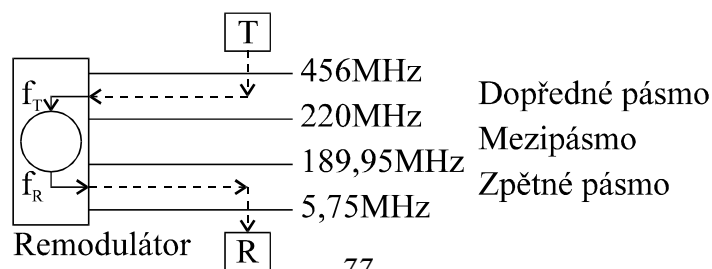


Obr. 10.1a: Přenos v základním pásmu

Úroveň log0 je reprezentována frekvencí f_1 a úroveň log1 kupř. frekvencí f_2 . V plném duplexu se na obou stranách spoje nachází modem (modulator/demodulátor), který provádí modulaci a demodulaci zprávy kupř. takto:

dotaz:	log1	1.270 Hz	odpověď	log1	2.225 Hz
			:		
	log0	1.070 Hz		log0	2.025 Hz

Při širokopásmovém přenosu, instaluje se na jedno přenosové médium více přenosových kanálů, včetně videa, dat, řeči ap. Přenos probíhá nepřímou, t.j. vysílací stanice vysílá dopředně směrem k remodulátoru v kanálu se šířkou pásma 6Mhz. Tam je zpráva transformována do zpětného frekvenčního pásma a poslána k přijímači. Dopředné pásmo obsahuje 30 kanálů, zpětné

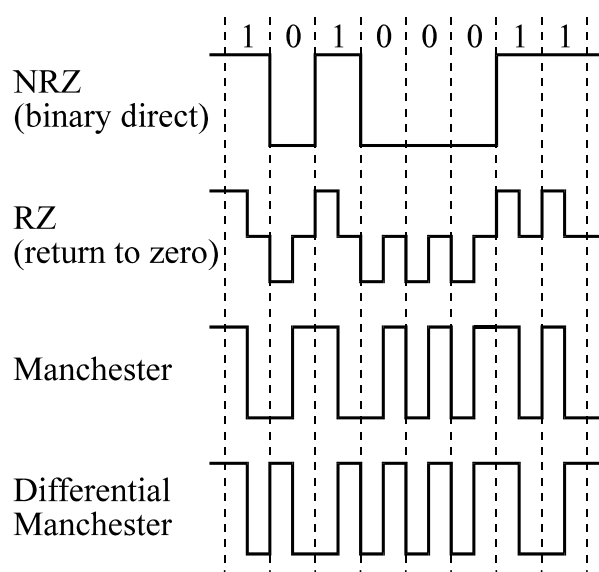


Obr. 10.1b: Princip frekvenčních pásem

pásmo 39 kanálů. Tedy 30 kanálů může být využíváno obousměrně ve stejném okamžiku.

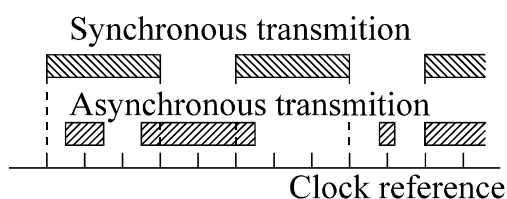
Co se týká kódování proudu bitů, v prostředí sítí se používá kódování NRZ, RZ, Manchester II, Diferencial Manchester, obr.10.2.

Pokud je takto kódovaný signál přenášen v základním pásmu, šířka přenosového pásma je určena rychlostí přenosu. Avšak i v průmyslových sítích se může použít širokopásmový přenos, který předpokládá namodulování digitálně kódovaného signálu, některým z výše uvedených způsobů, na nosný vlnový signál. Používají se především fázová a frekvenční modulace.



Obr.10.2: Kódování signálu

Pro zabezpečení přenosu kódovaného signálu je nutná jeho synchronizace. Druhá možnost je asynchronní přenos zpráv, obr.10.3.

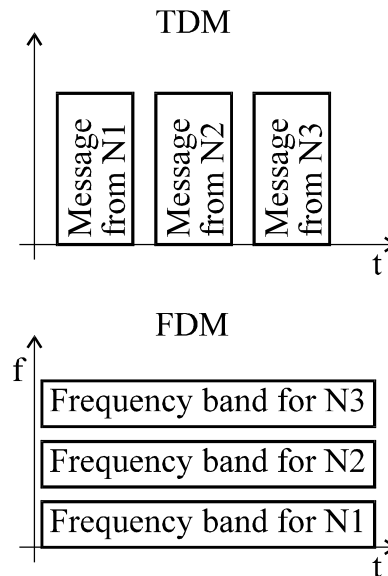


Obr. 10.3: Synchronní a asynchronní přenos zpráv

10.1.1 Výměna dat mezi jednotlivými účastníky

Přístup účastníků sítě ke společnému přenosovému médiumu je na jedné straně určen fyzicky - rozhraním (kupř. RS 485), zaručujícím tento spoj fyzicky - jednak funkcí dolní podvrstvy 2.

vrstvy modelu ISO/OSI. Přístup více účastníků k médiu se nazývá multiplexování a všeobecně se používají dva druhy multiplexingu. Časový (TDM) a frekvenční (FDM), obr. 10.4.



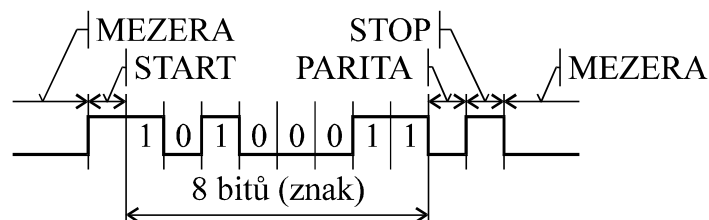
Obr.10.4: TDM a FDM

Časový multiplexing předpokládá synchronizaci přenosu zpráv. Každý z účastníků má přidělený určitý časový interval uvnitř přenosového cyklu. V době tohoto intervalu má právo vysílat svoji zprávu do média. Čas jednotlivých účastníků musí být synchronizován hodinovými impulsy. Frekvenční multiplex přiděluje každému účastníku jeho frekvenční pásmo po celou dobu cyklu. Frekvenční multiplex není jediným asynchronním způsobem přenosu zpráv, naopak časový a frekvenční multiplex jsou méně používané přístupové metody, než přístupové metody, vyvinuté zejména pro potřeby lokálních počítačových sítí LANů.

10.1.1.1 Synchronní a asynchronní přenos dat

Klasické asynchronní přenosové protokoly lze rozdělit na

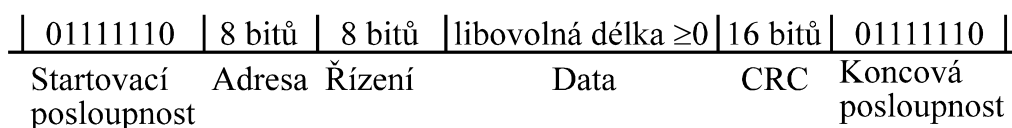
- znakově orientované (obr.10.5)
- bitově orientované
- blokově orientované



Obr. 10.5a: Znakově orientovaný přenos

Znakově orientované metody jsou velmi jednoduché. Jejich společným rysem je, že se snaží přenášet každý byte nezávisle. Každý datový byte zprávy je pak zahájen start bitem a ukončen stop bitem. Tento způsob se používá v telegrafii, v telefonních modemech ap. Je jednoduchý, avšak pomalý a méně efektivní (start, stop a paritní bit ke každému byte).

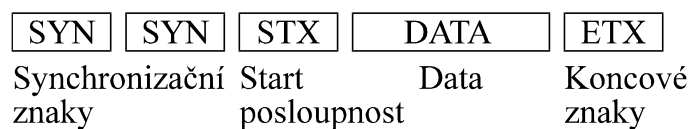
Bitově orientované přenosové metody vycházejí z toho, že se přenáší různě dlouhý řetězec bitů zprávy najednou. Z klasických metod tohoto typu se jedná o metodu SDLC (synchronous data link control) firmy IBM a jí podobné metody používané jinými firmami. V automatizaci více používaná je metoda HDLC (high-data link control). Tento protokol je standardizován organizací ISO a slouží jako srovnávací protokol pro jiné typy protokolů linkové vrstvy. Formát rámce HDLC je ukázán na obr.10.5b.



Obr.10.5b: Rámec HDLC

Rámec je zahájen startovací posloupností 6 jedničkami mezi dvěma nulami a stejnou posloupností je i ukončen. Vzhledem k tomu, že ve vlastní zprávě často dochází k podobné posloupnosti a přijímací stanice by ji mohla vyhodnotit jako ukončení přenosu rámce HDLC, používá se pro kódování zprávy metoda vkládání nuly, t.j. v datové posloupnosti, ve které se vyskytne 5 jedniček za sebou se za pátou jedničku vkládá nula a teprve pak se pokračuje další jedničkou. Dekodér nadbytečnou nulu vyřadí ze zprávy. Adresové pole se využívá, má-li více stanic zprávu přijímat. Kontrolní posloupnost indikuje typ zprávy. Informační část rámce nese vlastní data. Supervizory část slouží pro odpověď přijímacího uzlu o ztrátě přijímané informace. CRC je zabezpečovací část rámce, vypočítávaná z polynomu CRC.

Blokově nebo též bytově orientované přenosové metody vycházejí z toho, že jednotlivé byte jsou soustředěny do bloků a každý blok je ohraničen speciálními kontrolními znaky jako SOH (start of header), SYN (synchronization), STX (start of text), ETX (end of text) a EOT (end of transmission). Každý přenos začíná SOH a STX a končí ETX a EOT. Několik synchronizačních znaků SYN na začátku přenosu slouží někdy k synchronizaci přijímacího uzlu. Jeden příklad možného blokového přenosu je na obr.10.6.



Obr. 10.6: Blokově orientovaný přenos

10.1.2 Přístupové metody v lokálních sítích

Kromě výše uvedených synchronních i asynchronních protokolů linkové vrstvy se v LANech používají ve velké míře přístupové metody, vyvinuté především pro lokální sítě. Lokální síť (včetně lokálních sítí ve výrobě) je zpravidla omezena svým rozsahem, vyžaduje se u ní vysoký stupeň zabezpečení dat a je určena pro velký počet účastníků přenosu. Pro

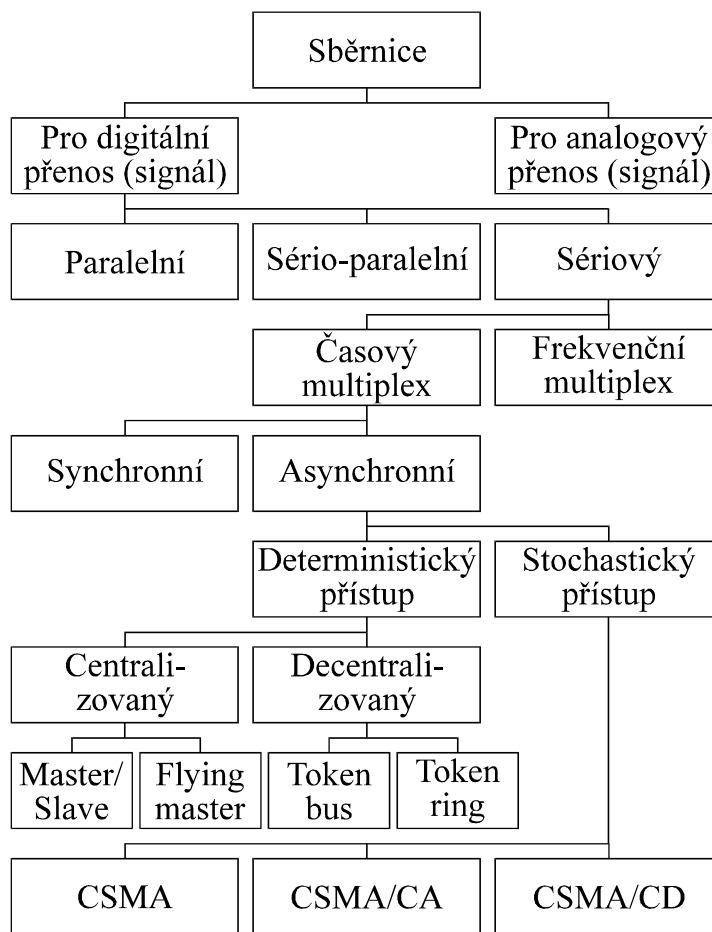
zabezpečení přístupu jednotlivých účastníků k jedinému přenosovému kanálu se používají v zásadě dvě skupiny metod:

- metody náhodného přístupu
- metody s definovaným přístupem

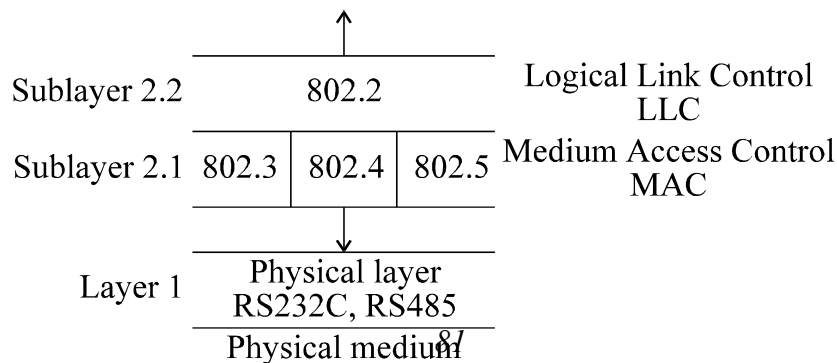
Tyto metody se dále ještě dělí na synchronní a asynchronní.

Rozdělení sériových komunikačních sítí podle způsobu přístupu k přenosovému médiumu je uvedeno na zjednodušeném obr. 10.7.

Na obr.10.8a je ukázáno přiřazení norem oběma podvrstvám linkové vrstvy modelu OSI, jak toto přiřazení provedly organizace IEEE a ISO.

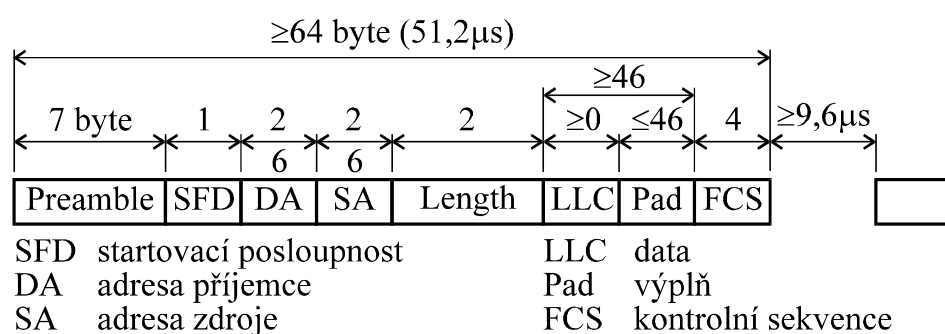


Obr.10.7: Přehled sériových komunikací



Obr. 10.8a: IEEE a ISO normy, použité v OSI modelu

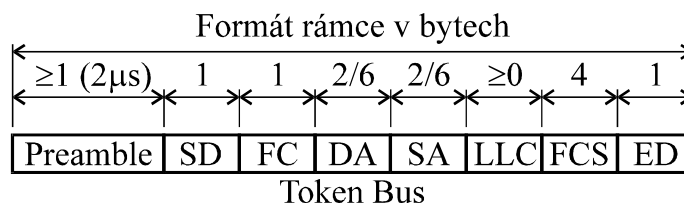
Jak je tedy patrné z obr.10.8a, přístupové metody jsou definovány na nižší podvrstvě linkové vrstvy. V současnosti nejvíce používaná přístupová metoda u sítí typu LAN je metoda CSMA/CD (Carrier Sence Multipliing Access with Collission detection), která patří do skupiny metod s náhodným (libovolným) přístupem. Používá se především u nejvíce rozšířených LANů typu Ethernet. Mezinárodní standardizační organizace ISO ji zahrnula jako normu ISO 8802-3. Ethernet se vyznačuje relativní jednoduchostí, nízkou pořizovací cenou, relativně jednoduchou instalací, několikerým provedením, vysokou přenosovou rychlostí. Jeho popularita neklesá. Je používán od r. 1980. Provedení je dvojí s tvrdým a měkkým koaxiálním kabelem. Používá se pro propojení osobních počítačů, pracovních stanic a to i na vyšších úrovních řízení procesu. Vzhledem k tomu, že metoda CSMA/CD nezaručuje definovaný přístup k médiu, není vhodný (v této původní podobě) k přímému řízení v reálném čase. Tvar paketu Ethernet je uveden na obr.10.8b. Je zřejmé, že vychází z tvaru rámce HDLC.



Obr. 10.8b: Tvar rámce Ethernet

Token bus (logický kruh) je další velmi rozšířená přístupová metoda. Vzhledem k tomu, že zaručuje definovaný přístup ke sběrnici, používá se v průmyslu v široké míře. Podporuje velmi rozšířenou topologii typu sběrnice (bus), vyniká vysokou rychlostí a jako přenosového média se používá koaxiální kabel i kroucená dvojlinka. Přístup ke sběrnici má pouze ten účastník přenosu, který vlastní pověření (token). Token je zvláštní paket, který nenesé vlastní data, pouze toto pověření a předává se v "logickém kruhu" od jednoho účastníka ke druhému.

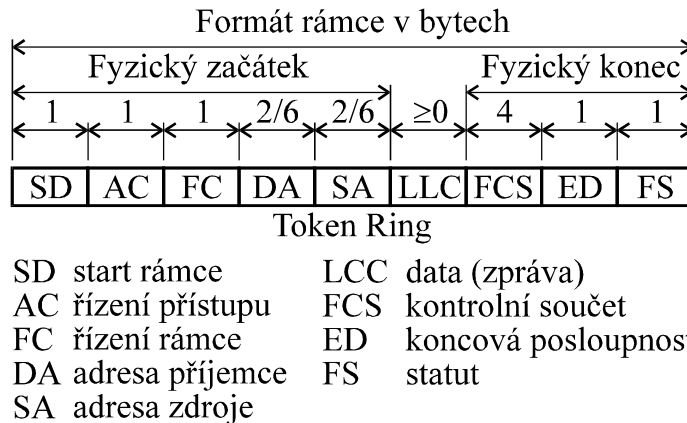
Pokud účastník, který obdržel token, nemá co vysílat, vyšle pouze pověření dalšímu účastníku. Vzhledem k tomu, že sousedství v síti je pouze logické, token musí obsahovat cílovou adresu. Předávání token komplikuje stykové obvody a zpomaluje komunikaci ve srovnání s



Obr.10.9a: Formát rámce token bus

CSMA/CD, avšak zaručuje definovaný okamžik přístupu k síti. Je nákladnější než Ethernet.

Poslední z těchto asynchronních přístupových metod je token ring (fyzický kruh). Princip funkce je podobný jako u Token bus, rozdíl je v tom, že sousedství je definováno fyzickým



Obr. 10.9b: Formát rámce Token Ring

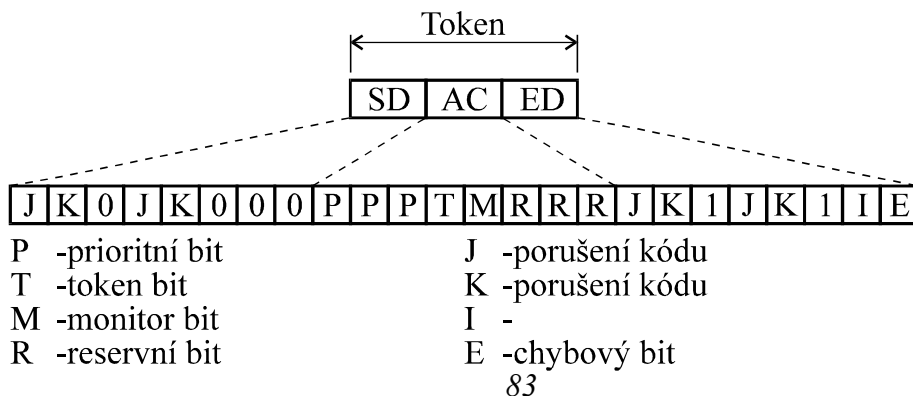
sousedstvím, t.j. kruhovým propojením jednotlivých účastníků. Pověření (token) cirkuluje po fyzickém kruhu k dalšímu účastníku. Jakmile účastník dostane token a má co vysílat, vyšle zprávu. Zpráva cirkuluje, každý z účastníků ji musí zachytit a pokud není určena pro něho, pošle ji dál. Dle typu sítě, může zpráva také dorazit opět až k vysílači s potvrzením, že byla příjemcem zachycena. Token ring je ještě pomalejší (1 - 4 Mbit/sec.), používá kroucenou dvoulinku nebo světlovodič (topologie fyzický kruh je pro světlovodič velmi vhodná na rozdíl od sběrnice). Je podporována firmou IBM pro průmyslové aplikace, příliš rozšířena zatím není.

Kromě metod náhodného přístupu (CSMA, CSMA/CD, CSMA/SD, Aloha, taktovaná Aloha a další) a metod deterministického přístupu (token passing) se v průmyslových sítích užívá hojně metod, vycházejících z TDM. Tyto metody vyžadují arbitra sítě, který dbá o přesné přidělování času každému účastníku k vysílání. Způsob předávání zpráv je pak polling a organizace sítě může být kupř. master-slave. Na dalším obrázku je vysvětlen způsob předávání zpráv poolingem a to při master-slave organizaci sítě a při uspořádání typu multi-master.

Centralizovaný provoz je nejprísnejší formou pollingu, kde výměna informace probíhá jen mezi řídicí stanicí a podřízenými stanicemi (slave). Základní schéma poolingem v centralizovaném provozu je na obr. 10.10a. Řídicí stanice vyzývá dvě podřízené stanice postupně se žádostí o data. První stanice nemá data k dispozici, posílá jen informaci o konci vysílání. Druhá stanice je vyzvána výzvou Poll T2, má připravena data, předává data a hlásí konec vysílání. Pak následuje výzva řídicí stanice další stanici Tn atd.

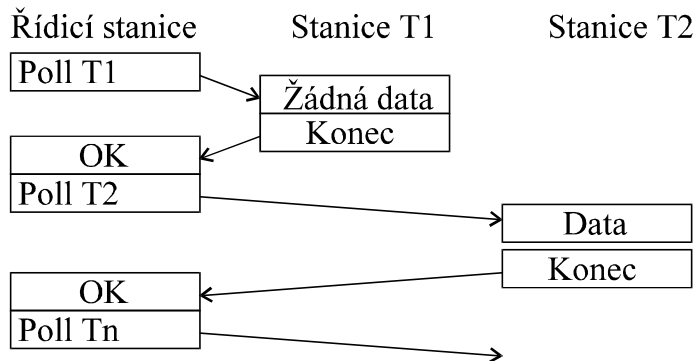
Jiný možný způsob centralizovaného pollingu je příčný provoz, zobrazený na obr. 10.10b.

Umožňuje přímý provoz mezi libovolnými stanicemi, které musí být příslušně

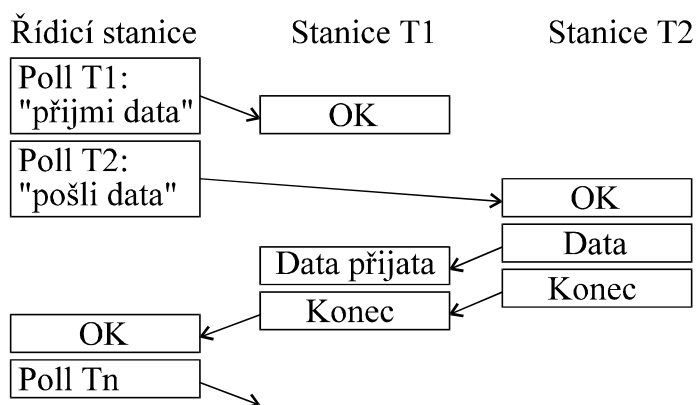


Obr. 7.9c: Formát pověření (Token)

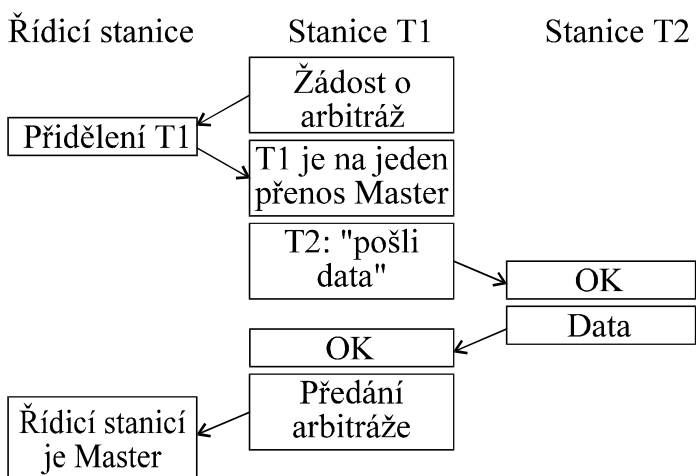
naprogramovány. Obrázek ukazuje předávání práva vysílat/přijímat na dvě stanice. Přitom však zůstává řízení na řídicí stanici, která povoluje jen časově omezený provoz mezi libovolnými stanicemi. Stanice T1 je informována, že dostane data od stanice T2 a stanice T2 je vyzvána předat data stanici T1. Po ukončení přenosu dat do stanice T1 hlásí stanice T1 řídicí stanici ukončení relace. Poté řídicí stanice vyzývá k pollingu další účastníky přenosu.



Obr. 10.10a: Polling- centralizovaný provoz master-slave



Obr. 10.10b: Polling- příčný provoz



Obr. 10.10c: Polling - multi-master provoz

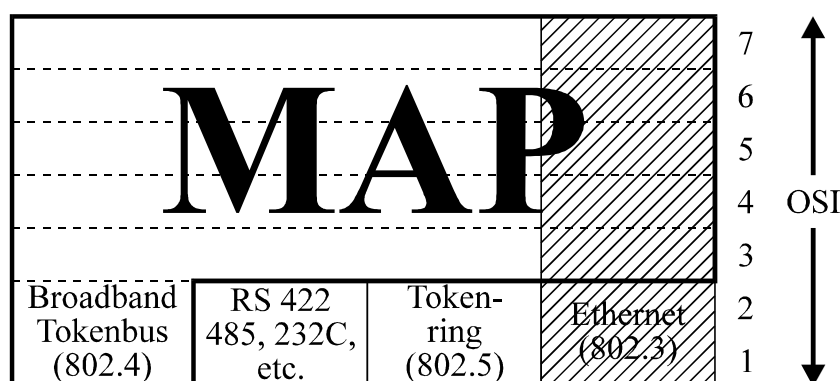
Posledním způsobem centralizovaného pollingu je struktura typu multi-master. Je to taková struktura, ve které minimálně na určitou dobu každý účastník komunikace může přebírat funkci mastera. To, která stanice bude v daném časovém intervalu masterem, je určeno v programu řídicí stanice nebo to vyplyne z aktuálních požadavků. V každém případě musí být vytvořena správa sítě, která slouží jako arbitr. Schéma centralizovaného pollingu ve struktuře multi-master je ukázána na obr. 10.10c.

Jako decentralizovaný (distribuovaný) polling jsou někdy označovány metody token passing.

10.2 Protokoly vyšších vrstev

Jak již bylo řečeno, 7 vrstvý model OSI je velmi obecným modelem, umožňujícím komunikaci v každém prostředí (rozlehlé komunikační sítě WAN, lokální sítě pro bankovní sektor, rezervaci letenek a jízdenek, komunikace mezi databázovými systémy ap.). Z toho důvodu má definovanou celou 7 vrstvou strukturu. V prostředí průmyslové výroby, kam je především zaměřena naše pozornost však mají funkce vrstev 3 až 6 malý význam. Proto pro malé průmyslové sítě je 7 vrstvý model zpravidla redukován jen na vrstvy 1, 2, a 7. Část služeb vrstev 3 až 6 musí převzít vrstva 2 a především vrstva 7, jak uvidíme dále při výkladu standardů fieldbusů. Proto se zaměříme již jen na vrstvu 7 modelu OSI. V této vrstvě se můžeme setkat s protokolem FTAM (File Transfer Access and Management protocol), který tvoří abstraktní notaci pro popis vlastností jednotlivých souborů (file), jako je tvorba dat, řízení přístupu, ochranný klíč, vícenásobný přístup. Jiným protokolem 7. vrstvy je MHS (Message and Handling System), podporující elektronickou poštu nebo VTP (Virtual Terminal protocol), definující hardwarově nezávislé ovládání terminálových výstupů.

Na závěr této části je třeba se zmínit o protokolu MAP (Manufacturing Automation Protocol), který je příkladem koncepce komunikace mezi stovkami účastníků průmyslového procesu jako jsou inteligentní čidla a akční členy, programovatelné automaty, pracovní stanice pro technickou a technologickou přípravu výroby, distribuované řídicí mikropočítače, terminály, plotry, počítače řízení výroby, počítače podnikové sítě, velké databázové systémy a další. Tento soubor pravidel pro komunikaci na všech úrovních modelu OSI umožňuje propojení heterogenních systémů různých výrobců. Systém byl vyvinut pro potřeby koncernu GM pro



Obr. 10.11: Přiřazení ISO a IEEE norem modelu protokolu MAP

automatizovanou výrobu a měl se stát průmyslovým de facto standardem. Definování norem pro jednotlivé vrstvy je uvedeno na obr.10.11.

Jak je patrné, ve fyzické vrstvě podporuje MAP pouze Token Bus, definuje způsob kódování a dekódování signálů (ISO 8824 a ISO 8825) a v nejvyšší vrstvě implementuje jiný standard MMS (Manufacturing Message Specification). I když i jiné kombinace standardů v jednotlivých vrstvách jsou technicky možné (a někdy i výhodné), MAP je nepřipouští (kupř. Ethernet není zahrnut do MAP). V automatizované výrobě jsou zhruba tři řídicí úrovně. Úroveň řízení podniku, úroveň řízení výroby a úroveň řízení technologického procesu. MAP se pohybuje na horních dvou úrovních a v úrovni řízení procesu jen částečně. Nezasahuje do úrovně čidel a akčních členů. Je příliš složitým a výkonným prostředkem, aby mohl být implementován na první úroveň řízení, která musí operovat v reálném čase. Proto ani fieldbusy nejsou zahrnuty do MAP. Na druhé straně je nástrojem pro CIM aplikace.

Ve vrstvě 7 implementuje MAP soubor abstraktních povelů pro dálkové řízení a monitorování jednotlivých prostředků automatizované výroby (MMS- Manufacturing Message Specification). MMS je ISO standardem ISO/IEC 9506. Tento dokument má 6 částí, které popisují :

Část 1 : Definici služeb

Část 2 : Definici protokolů

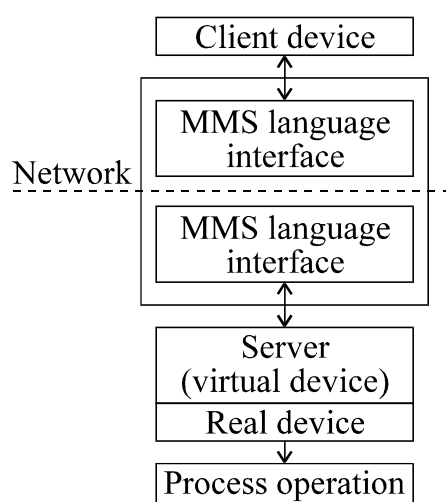
Část 3 : Pokyny pro řízení robotů

Část 4.: Pokyny pro řízení NC strojů

Část 5 : Pokyny pro řízení pomocí programovatelných automatů

Část 6 : Pokyny pro řízení procesů

MMS provádí velký počet různých služeb a volitelných funkcí. V prvních dvou částech MMS je popsáno, jak se provádí kupř. čtení a zápis proměnných na vzdáleném prvku, start a stop programu, přenos souborů mezi dvěma uzly ap. Další části MMS jsou jak patrně orientovány ke konkrétním technickým prostředkům řízení výroby. MMS je založena na objektově orientovaném programování. Základní kolekci příkazů pro jednotlivé komponenty tvoří VDM (Virtual manufacturing device). Služby MMS staví na modelu client - server, obr.10.12.

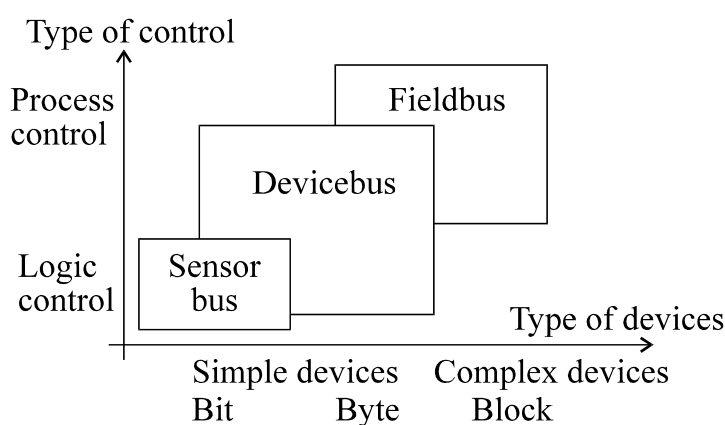


Obr.10.12: Architektura

Třebaže protokol MAP byl definován v druhé polovině 80. let, úspěšně aplikován ve výrobě koncernu General Motors a v dalších aplikacích, nestal se mezinárodně uznávaným standardem, zejména pro vysokou cenu implementace. Fieldbuses však ze specifikace MAP v mnoha ohledech vycházejí. Kupříkladu definování 7. vrstvy Profibusu (a jiných fieldbusů) je dáno FMS (Fieldbus message specification), která je podtřídou MMS z protokolu MAP.

11. PŘEHLED PRŮMYSLOVÝCH KOMUNIKAČNÍCH SBĚRNIC

V následujícím výkladu se zaměříme na nejvýznamnější a nejvíce rozšířené reprezentanty všech tří výše uvedených kategorií průmyslových sběrnic [13]. Jak již bylo řečeno a jak je patrné i z obr.11.1 rozdělení sběrnic do jednotlivých kategorií není striktní.



Obr.11.1: Rozdělení průmyslových sítí

Jednotlivé kategorie se navzájem překrývají a daný standard nebo kvazistandard je možné proto zařadit do obou ze sousedních skupin. Tak kupř. Interbus S je někdy charakterizován jako typický SensorBus, jindy je považován za DeviceBus (přenos po Bytech). Na druhé straně však by AS-interface nebo HART neměly být charakterizovány jako fieldbuses.

V následujícím výkladu uveďme parametry, vlastnosti, tvar protokolu a oblasti použití některých z následujícího přehledu nejpoužívanějších průmyslových sériových sběrnic:

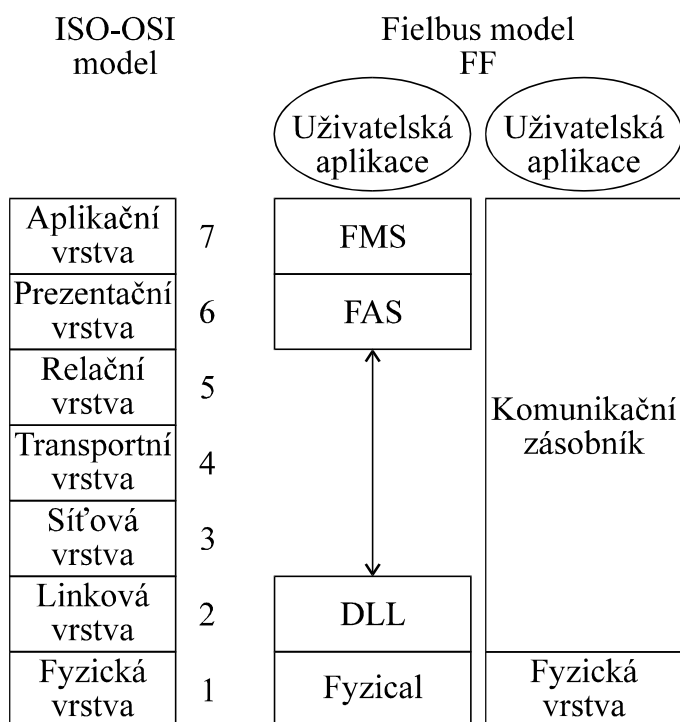
- Foundation Fieldbus
- Profibus
- FIP
- P-Net
- LonTalk
- CAN
- DeviceNet
- CANopen

- ControlNet
- Modbus (J-bus)
- Ethernet
- Interbus S
- HART
- BacNet

11.1 Foundation Fieldbus

Sériová průmyslová sběrnice, která nabývá na významu, zejména díky podpoře severoamerického trhu automatizace. Není sice identická s léta připravovanou normou IEC pro jednotný fieldbus. Vychází důsledně z většiny doporučení ISA SP50, zejména z definování linkové vrstvy dle IEC1158.

Současně však vychází i z projektu ISP (Interoperable Systems Project) a standardu WorldFIP. Spojuje tedy v sobě prvky Profibusu a FIPu, dvou nejpokročilejších evropských fieldbusů. Asi před 15 roky pověřila ISA (Instrument Society of America) pracovní skupinu SP50 návrhem celosvětového standardu fieldbusu. Z mnoha důvodů se práce protahovaly a mezitím vznikla celá řada úspěšných nebo méně úspěšných řešení. Je otázkou, jak se standard FF za této situace prosadí na trhu. Každopádně se jedná o otevřený systém, který je otevřený každému uživateli bez placení licenčních poplatků, jak je tomu u většiny sběrnic, uvedených v této práci.



Obr. 8.1a: Komunikační model FF

Vzhledem k důležitosti tohoto budoucího standardu sběrnice typu fieldbus, popíšme v dalším poměrně podrobně technické řešení a parametry FF.

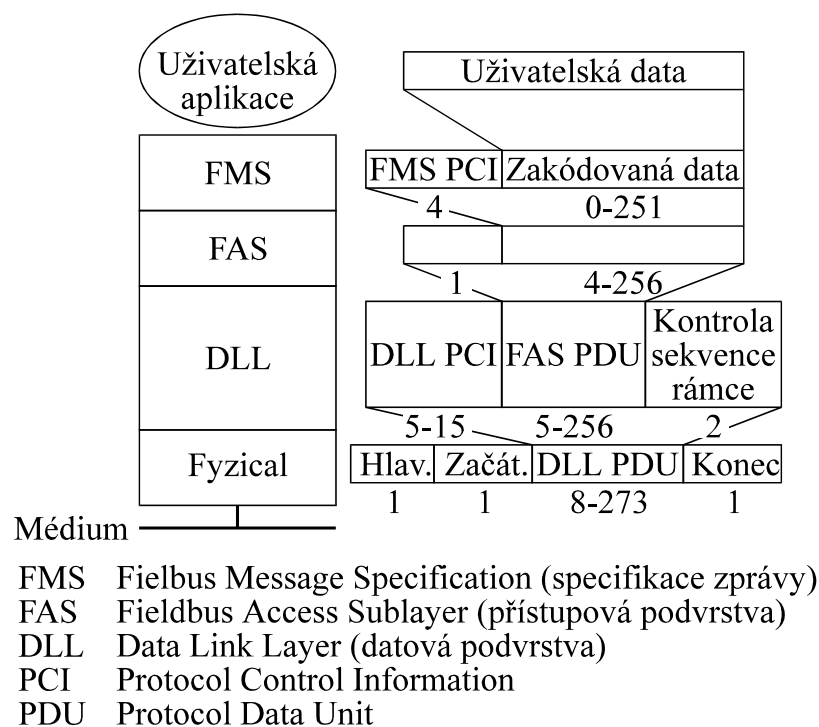
Standard vychází ze 7 vrstevového modelu ISO/OSI, využívá její 3 vrstvy, jak je u fieldbusů obvyklé, avšak netradičně definuje i "osmou vrstvu", nazvanou "User application", kterou budeme překládat jako uživatelskou aplikaci. Další anomálie je ve sloučení linkové a aplikační vrstvy v t.zv. komunikační zásobník (communication stack). V přehledu tedy komunikační model FF obsahuje:

- fyzickou vrstvu
- komunikační zásobník (vrstvy 2 a 7)
- uživatelskou aplikaci

Na obr.8.1a je nakreslen přibližný vztah mezi vrstvami protokolu FF a ISO/OSI modelem:

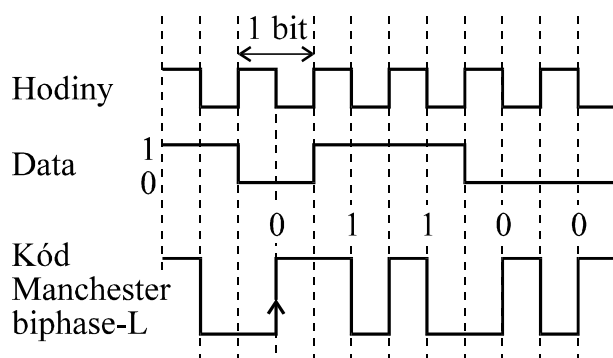
- fyzická vrstva odpovídá fyzické vrstvě ISO/OSI
- linková vrstva FF je podvrstva DLL linkové vrstvy modelu ISO/OSI
- přístupová vrstva FAS tvoří interface mezi 7. vrstvou (FMS) a linkovou vrstvou
- specifikace zprávy (FMS) je 7. vrstvou modelu ISO/OSI

Na obrázku je dále nakreslena uživatelská vrstva, která je pro uživatele nejdůležitější, ale není standardizována v modelu ISO/OSI. V obr. 8.1b je v jednotlivých vrstvách zapsána specifikace její funkce a rozsah v bajtech. Smysl pravé části obrázku bude zřejmý z dalšího výkladu funkce a služeb jednotlivých vrstev modelu FF.



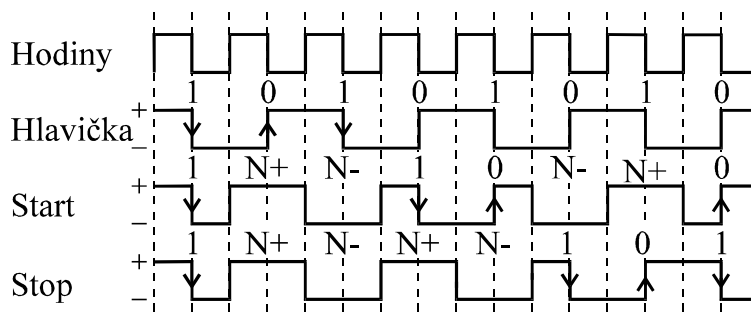
Obr. 8.1b: Funkce a služby vrstev FF

Fyzická vrstva je standardizována jednak americkým standardem ISA a zejména IEC jako IEC1158-2. Vrstva přijímá zprávy z komunikačního zásobníku a transformuje je na elektrický signál. Funkce zahrnuje doplňování a odstraňování záhlaví (preamble) a bitů, které charakterizují záhlaví a konec rámce (start/end delimiter). Dále kóduje signál kódem Manchester II (označovaným jako Manchester Biphase-L). Jak je patrné i z obr.8.2a, přijímač reaguje na kladnou nebo zápornou hranu signálu uprostřed bitového intervalu (zde je bitový interval dvojnásobkem jednotkového intervalu).



Obr. 8.2a: Kódování signálu

Lze hovořit o synchronním způsobu kódování. Sestupná hrana je vyhodnocena jako log. 1, vzestupná jako log.0.

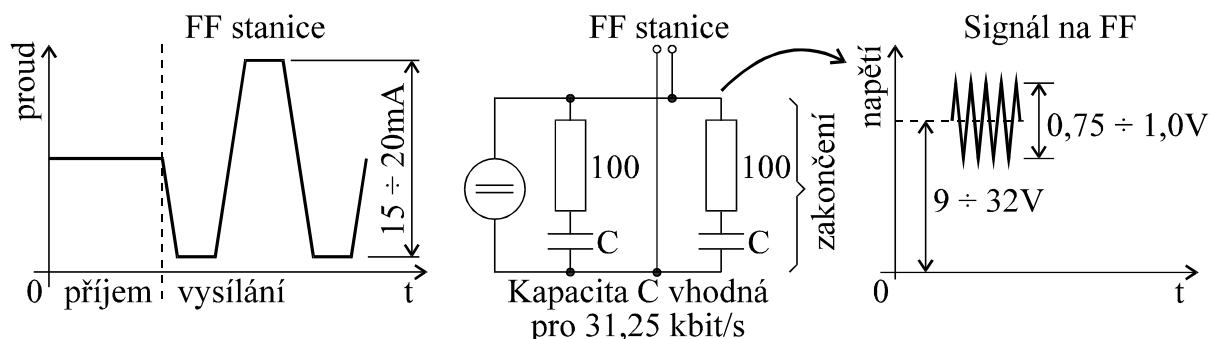


Obr. 8.2b: Tvar záhlaví zprávy

Pro záhlaví (Preamble) a začátek zprávy (Start Delimiter) a konec zprávy (End Delimiter) jsou vyhrazeny v FF zvláštní znaky N+ a N-, jak je patrné z obr. 8.2b

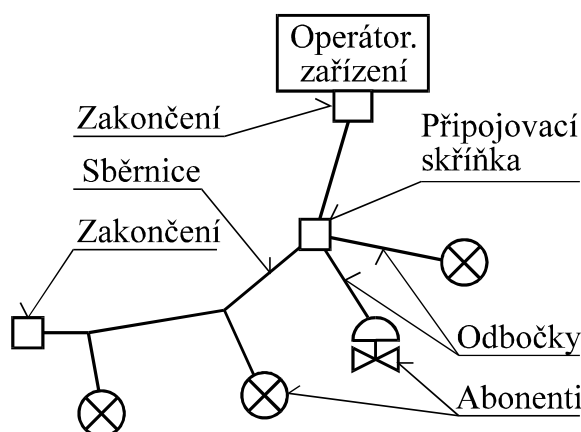
Povšimněme si, že znaky N+ a N- se nikdy nemění uprostřed bitového intervalu, jako je tomu u ostatních informačních znaků log1 a log0. To slouží přijímači k snadnému rozpoznání začátku a konce zprávy. Záhlaví slouží k synchronizaci zdroje taktovacího kmitočtu v přijímači. Přijímač začne přijímat zprávu po příchodu začátku zprávy a přijímá až do příchodu konce zprávy.

Základní přenosová rychlost ve variantě H1 je 31,25 kbitů/s. Tato rychlost je vhodná k přenášení údajů při řízení procesů typu: regulace teploty, hladiny, průtoku a to jak v nevybušných, tak ve výbušných provozech.



Obr. 8.2c: Signály na vedení a připojení přístrojů

Síla FF je m.j. v tom, že od počátku je definován do prostředí s nebezpečím výbuchu (EEx). Je tedy přizpůsoben potřebám chemického průmyslu a dalším výbušným provozům. Musí v ní mít zaručeno napájení přístrojů po sběrnici a maximální celkový proud do 20mA. Přístroje ve výbušné zóně musí mít možnost být napájeny přímo ze sběrnice proudem do 20mA a fungovat po vedení, které bylo v minulosti využíváno pro přenos a řízení proudovou smyčkou 4 - 20mA. U FF vysílač vysílá při 31,25 kbitů/s proudový signál $\pm 10 \text{ mA}$ do zátěže 50 ohm, takže napětíový signál, namodulovaný na nosné ss napětí je 1V. V případě instalace ve výbušné zóně, musí být umístěna I.S. (intrinsically safe) bariéra mezi napájecí zdroj v bezpečné zóně a I.S. přístroj, umístěný v EEx zóně. Napájecí stejnosměrný zdroj může pracovat s napětím 9 až 32V, avšak v případě I.S. aplikací velikost napětí závisí na provedení bariéry. Signálové poměry a připojení přístrojů a napájecího zdroje ve variantě FF H1 je na obr. FF2c.

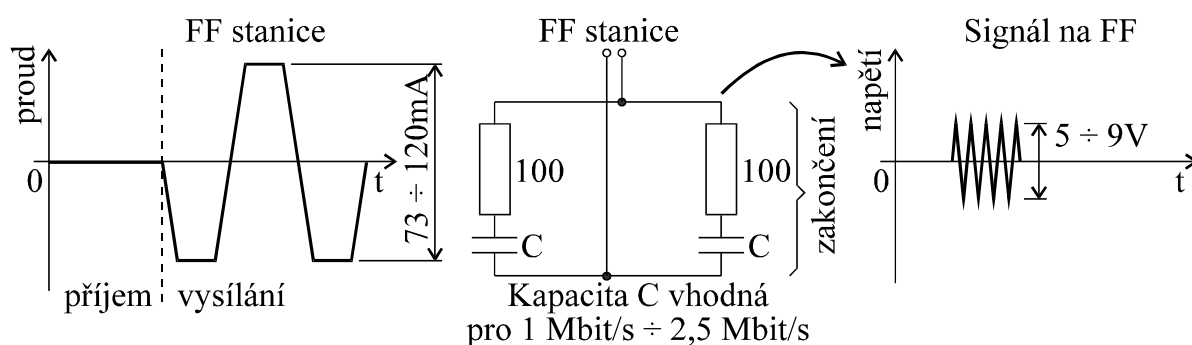


Obr. 8.2d: Příklad topologie

Topologie v případě FF H1 je sběrnice s krátkými odbočkami (do 1m). Delší odbočky musí být zahrnuty do celkové délky vedení, která je ve variantě H1 max. 1900m. Takže kupř. pro max. 12 účastníků mohou být odbočky od segmentu až 12m dlouhé. Typický počet účastníků sítě FF je 32 (při odbočkách do 1m délky). Příklad topologie sítě uveden na obr. 8.2d.

Pro rychlé procesy je definována rychlost přenosu 1 nebo 2,5Mb/s (varianta H2). Varianty H2 (1 nebo 2,5 Mb/s) se používají při řízení rychlých a rozlehlých procesů. Zde se nepřipouští napájení ze sběrnice. Stanice musejí být napájeny po zvláštním dvou vodiči (tedy čtyřvodičový kabel) anebo lokálně. Vysílací proud je $\pm 60\text{mA}$ do zátěže s ekvivalentním odporem 75 ohmů. Vznikne tak modulační signál 9V špička-špička (viz obr.8.3).

Varianta s přenosovou rychlostí 1Mb/s podporuje i zvláštní režim (current mode), umožňující napájení ze sběrnice a určený do výbušné zóny. Zpráva je namodulována na 16 kHz střídavý proud, umožňující vlastní napájení přístrojů. Přístroj je ke sběrnici připojen pomocí speciálních konektorů, které používají induktivní vazbu k oddělení signálu od napájení.



Obr. 8.3: Signálové poměry na síti H2

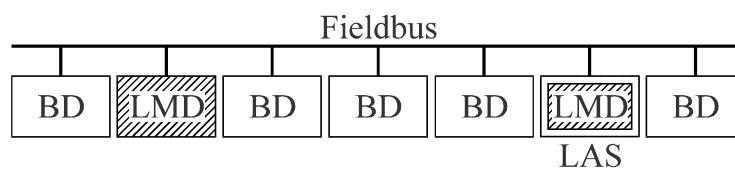
Ve variantě H2 (přenosová rychlost 1 Mbit/s nebo 2,5Mbitů/s) nejsou přípustné odbočky s ohledem na odrazy. Topologie je tedy jen čistá sběrnice s krátkými přípojkami a zakončením na obou stranách.

Komunikační zásobník

Linková vrstva (DLL)

Řídí přístup účastníka sítě ke sběrnici a to pomocí deterministického centralizovaného řadiče sběrnice LAS (Link Active Scheduler). Linková vrstva definuje 3 typy účastníků, viz obr. 8.4.

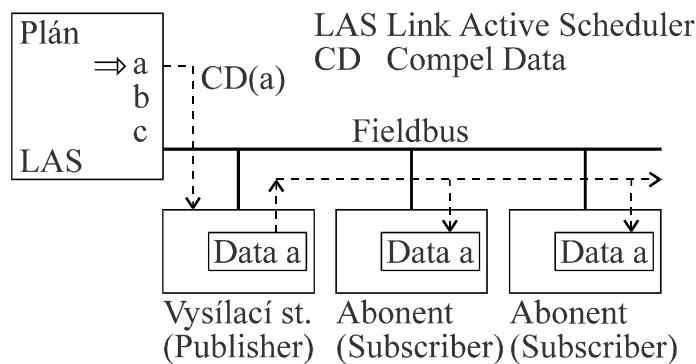
- basic device (BD)- nemohou fungovat jako LAS
- link master (LMD)- obsahují řadič sběrnice a mohou fungovat jako LAS
- bridge - slouží k propojování segmentů sítě, včetně propojování systémů H1 a H2



Obr. 8.4: Účastníci přenosu

Na příkladu linkové vrstvy protokolu FF je vhodné podrobně vysvětlit její funkci v aplikaci na fieldbusu.

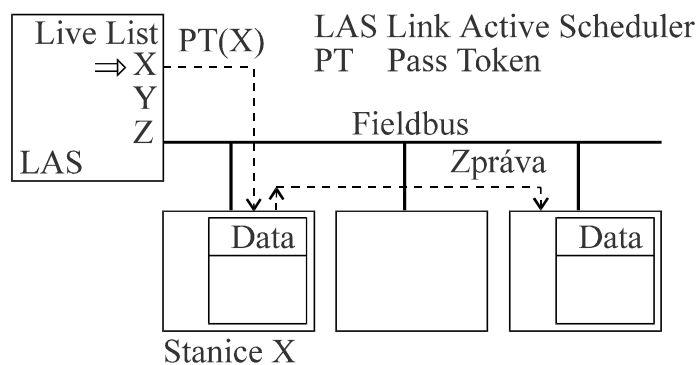
Při konfigurování sběrnice se v řadiči vytvoří seznam všech vysílacích časů pro všechny vyrovnávací paměti (buffer) ve všech zařízeních, ze kterých se mají cyklicky vysílat údaje. V okamžiku, kdy má určité zařízení vyslat data, zašle mu řadič zprávu se žádostí o poskytnutí dat (Compel Data - CD). Jakmile zařízení obdrží CD, vysílá nebo "publikuje" data z bufferu všem zařízením, která jsou nakonfigurována přijímat tato data. Tyto entity jsou nazývány "subscribers". Na obr.8.5 je tato situace znázorněna.



Obr. 8.5: Komunikace na síti

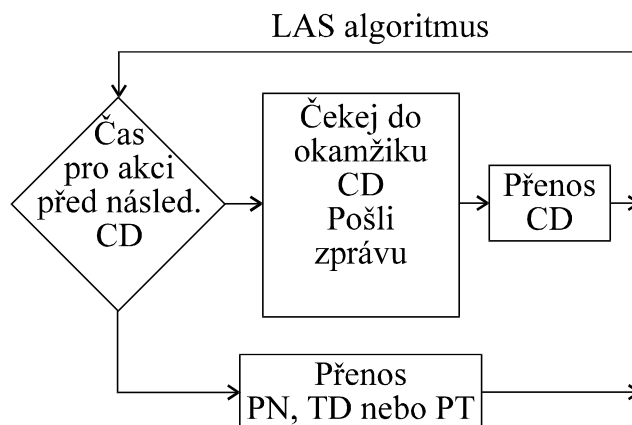
Tyto t.zv. plánované přenosy jsou typické pro pravidelné, cyklické přenosy údajů v řídicích smyčkách, tvořených zařízeními připojenými ke sběrnici. Jejich zajišťování je pro řadič sběrnice aktivitou s největší prioritou.

Všechna zařízení, připojená ke sběrnici, mají možnost v mezičasech plánovaných cyklických přenosů vyslat zprávu mimo plánovaný čas. Obr.8.6.



Obr. 8.6: Vysílání zprávy mimo plánovaný čas

Jde kupř. o běžný případ žádosti o data od operátorů nebo od SCADA systémů nebo v opačném případě o data, vysílaná přístroji, umístěnými v procesu (výstražná hlášení ap.). Při konfiguraci musí být těmto neplánovaným přenosům zajištěna dostatečná doba. Tím, že LAS pošle jednotlivým abonentům PT (Pass Token - pověření k vysílání), je od okamžiku obdržení PT příslušné zařízení oprávněno vysílat tak dlouho, jak potřebuje, pokud není překročena maximální doba držení token. Vlastní PAS (řadič sběrnice) pracuje dle vývojového diagramu z obr.8.7.



Obr. 8.7: Vývojový diagram řadiče sběrnice

Seznam účastníků přenosu, který přesně odpovídá účastníkům, kterým LAS zasílá token, je t.zv. seznam živých adres (live list). V tomto seznamu se uchovávají adresy všech stanic, které správně reagují na zaslání PT. Aby mohlo být kdykoli ke sběrnici připojeno nové zařízení, PAS periodicky zasílá zprávu zkouška uzlu (Probe Node PN) všem stanicím, které správně nereagují a nejsou tudíž na "live list". Poté, co PAS obešle pověření PT všechny živé stanice, musí alespoň jednu neživou osvěžit signálem PN. Adresa zůstává na seznamu živých adres tak dlouho, dokud účastník sítě správně reaguje na zaslání PT. Řadič ji odstraní z "live list" poté, když zařízení buď nevyužije práva vyslat zprávu nebo je vrátí řadiči při třech bezprostředně po sobě následujících pokusech o předání. O každém zařazení nebo vyjmutí adresy ze seznamu uvědomuje řadič sběrnice všechna zařízení, která si tak sama mohou uchovat aktuální kopii "live list". Další signál, který řadič vysílá je nastavení času (Time distribution - TD). Tím se periodicky stará o synchronizaci taktovacích generátorů všech připojených stanic. Synchronizace je důležitá jak pro plánované přenosy, tak pro spouštění funkčních bloků, o čemž bude řeč v následujícím.

Na sběrnici FF může být několik zařízení ve funkci link master. Tito účastníci mají m.j. tu funkci, že v případě chyby řadiče přebírají jeho funkci. Foundation fieldbus je tudíž označován jako "fail operational".

Subvrstva FAS

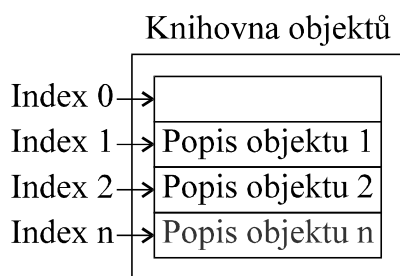
Přístupová podvrstva komunikačního zásobníku protokolu FF (nebo 7. vrstvy modelu ISO/OSI) umožňuje komunikaci mezi 2. vrstvou a podvrstvou FMS aplikační vrstvy (7. vrstva modelu ISO/OSI). Využívá plánovaných i neplánovaných činností v linkové vrstvě k poskytování služeb své nadřazené vrstvě, kterou je specifikace zprávy FMS. Typy služeb,

poskytovaných přístupovou subvrstvou jsou popsány virtuálními komunikačními vztahy (Virtual Communication relationship - VCR). Virtuální komunikační vztahy lze přirovnat k funkci telefonního přístroje s pamětí. I tam stačí navolit příslušné číslo volaného včetně mezinárodního a národního kódu jen jednou a je uloženo pro nové použití. Podobně u VCR, po konfiguraci jsou v paměti přístupové vrstvy virtuální komunikační vztahy a stačí vyvolávat jen jejich kód. VCR jsou tohoto typu :

- klient/server používané pro požadovanou, neplánovanou, uživatelem iniciovanou komunikaci mezi dvěma zařízeními na sběrnici (příkladem může být typicky - komunikace operátora při změnách nastavení, potvrzování výstražných hlášení, vkládání programů do zařízení ap.)
- šíření zprávy (report distribution) používaného pro pořadovou, neplánovanou, uživatelem dat iniciovanou komunikaci jednoho zařízení s několika zařízeními současně (typická komunikace zařízení na sběrnici při zaslání výstrahy k operátorským panelům, hodnot k zobrazení časových průběhů ap.)
- poskytovatel/příjemce (publisher/subscriber) používaného pro přenos právě aktuálních údajů z jednoho do několika zařízení současně (v síti jsou k dispozici pouze aktuální údaje a nové údaje kompletně přepisují staré; tato služba se používá pro cyklické plánované šíření I/O dat v síti)

Specifikace zprávy FMS (nejvyšší vrstva komunikačního modelu FF)

Podvrstva "specifikace zprávy" (Fieldbus Message Specification FMS) kóduje a dekóduje příkazy uživatelské vrstvy (User Layer Commands) a umožňuje tak jednomu uživateli posílat zprávy druhému uživateli při použití množiny standardních formátů zpráv. FMS popisuje

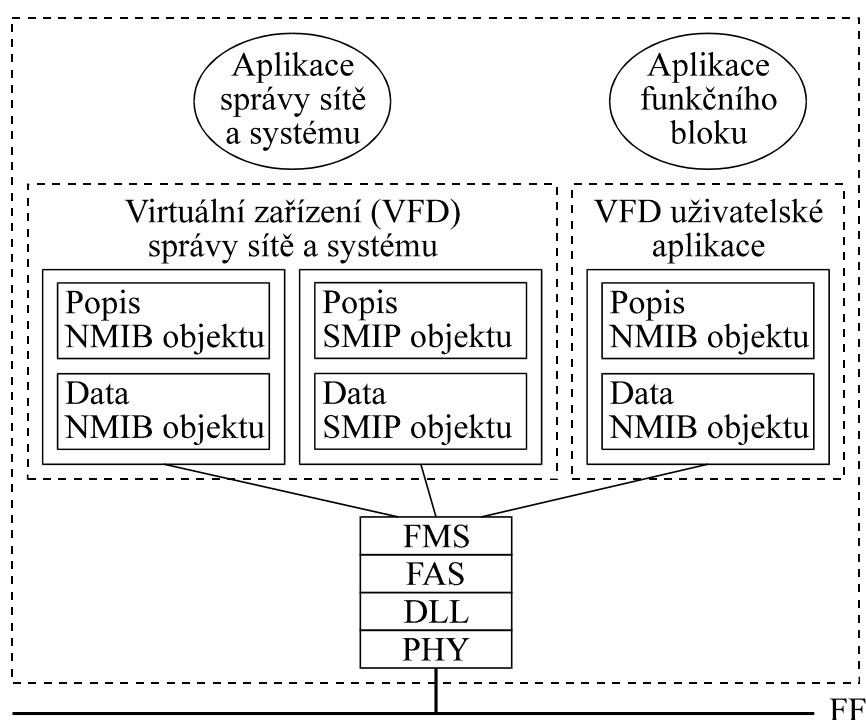


Obr. 8.8: Slovník objektů

komunikační služby, formát zpráv a protokol, potřebný pro tvorbu zprávy v uživatelské vrstvě (User Application). I u protokolu FF je založena na činnosti 7. vrstvy modelu ISO/OSI.

Údaje přenášené po sběrnici jsou přitom popsány prostřednictvím popisu objektů (object description). Tyto popisy objektů jsou uchovány ve struktuře, nazvané slovník objektů (Object Dictionary - OD), ve které je každý popis objektu identifikován svým indexem v OD. Na obr. 8.8 je tato situace znázorněna. Pod indexy menšími, než 256 jsou definovány typy dat jako boolean, integer, float, bitstring a dále struktury dat, používané k sestavování ostatních popisů objektů. Popisy objektů uživatelské aplikace mohou začínat kterýmkoli indexem nad 255.

K tomu, aby se dalo přistupovat po sběrnici k údajům, uloženým v bufferu stanice ve slovníku objektů (OD), slouží mechanismus virtuálního zařízení (Virtual Field Device - VFD). Typický účastník přenosu obsahuje alespoň 2 tato virtuální zařízení, jak je ukázáno na obr. 8.9.



Obr. 8.9: Mechanismus virtuálního zařízení VFD

Jedná se jednak o virtuální zařízení uživatelské aplikace (User Application VFD), jednak o Network and System Management VFD. Tyto VFD umožňují přístup k informační bázi správy sítě (NMIB- Network Management Information Base) a bázi správy systému (SMIB- System Management Information Base). Data v NMIB obsahují také VCR (Virtual Communication Relationship), dynamické proměnné, statistická data a LAS (Link Data Scheduler), v případě, že zařízení je typu Link Master. Na druhé straně data v SMIB obsahují identifikátory přístroje (device tag) a adresu a časový plán pro provádění blokových funkcí.

Řízení systému (System management) je pak ještě popsán v části, popisující uživatelskou aplikaci.

Jak bylo řečeno, FMS obsahuje rovněž popis komunikačních služeb (Communication Services).

A. Komunikační služby

Komunikační služby zajišťují standardizované vybavení pro tvorbu uživatelské aplikace jako funkční bloky pro komunikaci po fieldbusu. Pro každý objekt jsou definovány specifické FMS komunikační služby. Všechny z těchto služeb může využívat jen VCR (Virtual Communication Relationship) typu Client/server.

Služba správy zahájení a ukončení (Context Management Services)
sestavující z následujících služeb:

- Initiate - ustavení komunikace
- Abort - zrušení komunikace
- Reject - odmítnutí nekorektní služby
- Status - čtení statutu zařízení
- Unsolicited Status - posláni nepotvrzené zprávy
- Identify - čti výrobce, typ a verzi zařízení

Služba používání Slovníku objektů (Object Dictionary Services)

Následující služby umožňují uživatelské aplikaci přístup k objektu a provedení jeho změny (OD) ve VFD:

- GetOD - čti slovník objektů (OD)
- InitiatePutOD - start nahrávání OD
- PutOD - nahrávání OD do zařízení
- TerminatePutOD - ukončení OD nahrávání

Služba přístupu k proměnným (Variable Access Services)

Následující FMS služby umožňují uživatelské aplikaci přístup a změnu proměnných přiřazených popisu objektu.

- Read - čti proměnnou
- Write - zapiš proměnnou
- InformationReport - pošli data
- DefineVariableList - definuj seznam proměnných
- DeleteVariableList - vymaž seznam proměnných

Služba správy událostí (Event Services)

Následující služby umožňují uživatelské aplikaci ohlásit výskyt události a provést její ošetření.

- EventNotification - ohlášení události
- AcknowledgeEventNotification - potvrzení události
- AlterEventConditionMonitoring - umožnit/znemožnit událost

Služba Upload/Download

Je občas nutné provést dálkově upload a download dat a programů prostřednictvím fieldbusu, zejména pro složitější zařízení jako jsou např. PLC, propojené pomocí FF. K tomu jsou ve službách FMS vyhrazeny domény "domain". Domény reprezentují paměťový prostor v daném zařízení.

Následující FMS služby umožňují uživatelské aplikaci upload and download domény do vzdáleného zařízení (remote device).

- RequestDomainUpload - žádost o upload
- InitiateUploadSequence - otevřít obsah upload
- UploadSegment - číst data ze zařízení
- TerminateUploadSequence - stop upload
- RequestDomainDownload - žádost o download
- InitiateDownloadSequence - otevření obsahu downloadované zprávy
- DownloadSegment - poslat data do zařízení
- TerminateDownloadSequence - stop download

Služba vyvolání programu (PI)

Tato služba dovoluje řídit exekutivu programu v daném zařízení na dálku (po fieldbusu).

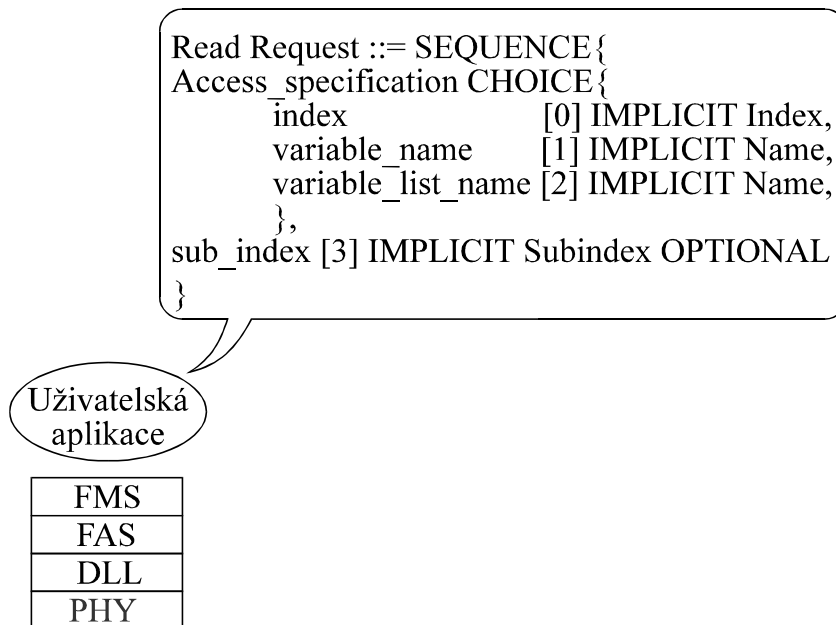
Zařízení může zapsat (download) program do domény jiného zařízení s použitím služby download a pak dálkově zpracovat program při využití služeb PI.

- CreateProgramInvocation - vytvořit programový objekt
- DeleteProgramInvocation - vymazat programový blok
- Start - start programu
- Stop - stop programu
- Resume - znovu spustit provedení programu
- Reset - znovunastavení programu
- Kill - odstranění programu

Služba formátování zprávy

Přesné formátování FMS zpráv je definováno formálním syntaktickým popisným jazykem zvaným Abstract Syntax Notation 1 (ASN.1). Tento jazyk byl vyvinut mezinárodní telegrafní a telefonní konzultační organizací (CCITT) na začátku 80. let jako jedna z činností organizace CCITT sloužící ke standardizaci aktivit pošty. Na obr. 8.10 je ukázka formátování v tomto jazyce. Aplikace se týká služby Read z FMS.

Příklad ukazuje, že položky Access- specification a sub-index se vyskytují v části SEQUENCE zprávy. Položka Access-specification je charakteru CHOICE, t.j. přístup k dané proměnné je možné buď přes index nebo použitím jména proměnné. Položka sub-index je charakteru OPTIONAL. T.j. použije se jen pro výběr jednotlivého elementu nebo pole nebo k zápisu proměnné. Čísla v hranatých závorkách jsou aktuálními dekodovacími čísly, která jsou použita pro

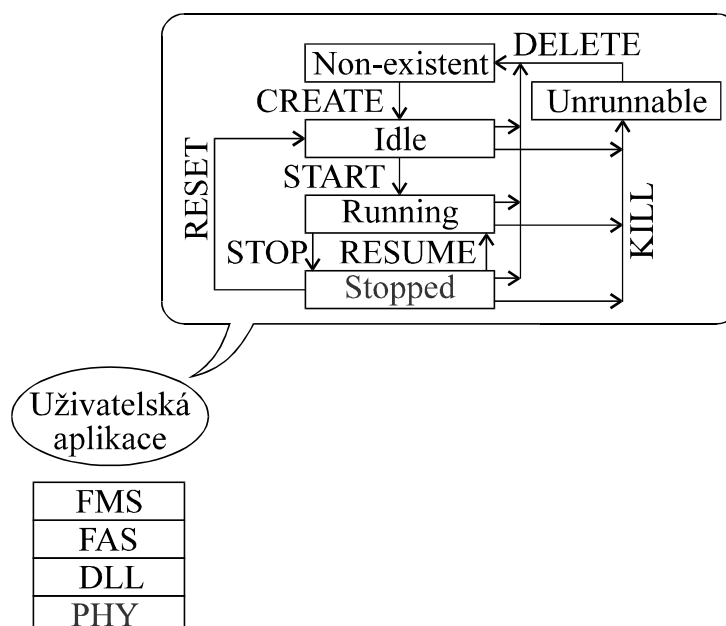


Obr. 8.10: Způsob formátování

identifikaci polí v dekodované zprávě.

Chování protokolu (Protocol Behavior)

Jistý typ objektů má zvláštní pravidla chování ve specifikaci FMS. Např. zjednodušené chování služby PI je na následujícím obr. 8.11. Vzdálené zařízení



Obr. 8.11: Zjednodušené chování služby PI

může řídit stav programu v jiném zařízení na fieldbusu. Např. vzdálené zařízení může využít služby CREATE PROGRAM INVOCATION ke změně stavu programu z Non-existent na Idle.

B. Uživatelská vrstva

Na rozdíl od jiných standardů, je uživatelská vrstva ve standardu FF definována. Uživatelská vrstva se nachází nad vrstvou aplikační (nejvyšší (7.) vrstvou ISO/OSI modelu. Prostřednictvím aplikace, napsané v uživatelské vrstvě se může kupř. přímo ovládat ventil nebo regulátor, připojený ke sběrnici.

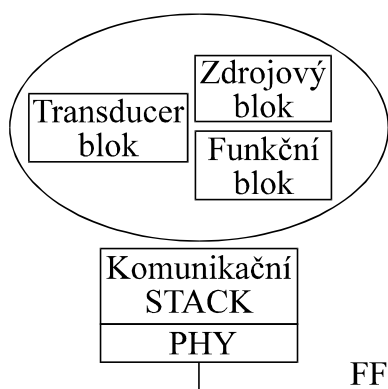
Uživatelská vrstva plní tři samostatné okruhy funkcí :

- řízení sítě
- řízení systému
- tvorbu uživatelské aplikace

Funkční bloky

Vlastní uživatelská aplikace je vystavěna z funkčních bloků (Function Block - FB). Jde o SW moduly, kterými se dají přímo ovládat funkce a činnost jednotlivých připojených účastníků sítě. FB zvyšují universálnost použití a zjednodušují tvorbu aplikace. FB jsou definovány svými vstupy a výstupy, parametry, reakcí na událost a volbou režimu příslušného zařízení. Každá z uvedených položek představuje "objekt", který je spolu s ostatními objekty mapován do "slovníku objektů" a může s nimi být různě seskupován v mnoha různých programových souborech.

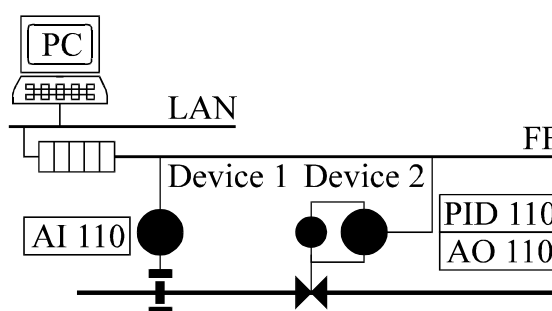
Vstupní a výstupní parametry funkčních bloků se dají předávat po sběrnici. Každý funkční blok je prováděn v přesně stanoveném čase. Jedna aplikace může obsahovat několik funkčních bloků. Použití bloků v aplikaci je patrné z obr.8.12.



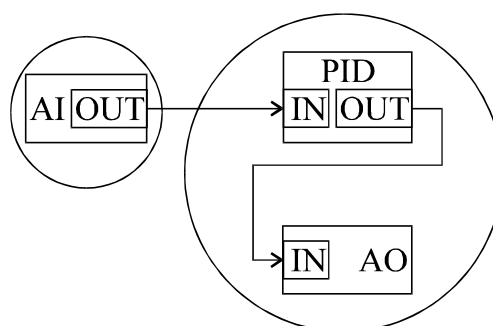
Obr. 8.12: Použití bloků v aplikaci

Na obr.8.13 je ukázán způsob realizace regulační smyčky pomocí funkčních bloků, umístěných v příslušných účastnících sítě. Zde je to jednak měřící člen (senzor tlaku nebo teploty s nakonfigurovaným funkčním blokem AI) a ventil, v jehož aplikaci (uživatelském programu) je použit jednak FB typu AO, jednak FB typu regulátor PID. Je to typická ukázka jednoduchého "soft control" v prostředí "totálně distribuovaného systému".

Na obr. 8.14 je pak znázorněn proces konfigurování regulační smyčky z obr. 8.13, kde AO je funkční blok analogového výstupu, AI je funkční blok analogového vstupu a PID je funkční blok regulátoru.



Obr. 8.13: Regulační smyčka tlaku



Obr. 8.14: Konfigurování regulační

Standard FF definuje těchto 10 funkčních bloků :

- analogový vstup (AI)
- analogový výstup (AO)
- odchylku (Bias - B)
- výběr řízení (Control Selector - CS)
- diskrétní vstup (DI)
- diskrétní výstup (DO)
- ruční vložení (Manual Loader - ML)

- PD regulátor (PD)
- PID regulátor (PID)
- poměrový regulátor (Ratio - RA)

Základní vybavení funkčními bloky umožňuje jak sledování (monitor only), tak i nadřazené řízení (Override) a distribuované řízení (Split Range Control). Pro vyšší funkce řízení je ve standardu definováno dalších 19 funkčních bloků, zahrnujících generátory složitějších vstupních signálů (rampa), až po kupř. dynamický kompenzátor.

Funkční bloky lze instalovat do zařízení na sběrnici tak, aby jednotlivá zařízení prováděla požadované funkce. Tak např. jednoduchý vysílač tlaku může obsahovat funkční blok analogového vstupu (AI). Regulační ventil může kromě nezbytného FB typu analogového výstupu (AO) obsahovat i blok (PID). Tak je možné (totálně distribuovaným systémem) realizovat regulační funkce, které by bylo, bez této sběrnice architektury a takto definované uživatelské vrstvy, nutné realizovat kupř. řídicím mikropočítačem nebo PLC. Tento příklad je uveden na obr.8.26 na konci této kapitoly.

Jak je ukázáno na obr.8.12, kromě funkčních bloků, obsahuje aplikace i zdrojový blok (Resource Block) a přenosový blok (Transducer Block).

Resource Block

Tento blok popisuje charakteristiky zařízení, připojeného k fieldbusu, jako je jméno zařízení, výrobce, číslo série. Každá aplikace obsahuje jen jeden tento Resource Block.

Transducer Block

Tento blok odděluje FB od požadovaných lokálních I/O informací. Obsahuje informace jako kalibrační data a typ sensoru. Uživatelská aplikace zpravidla musí mít ke každému vstupnímu a výstupnímu funkčnímu bloku příslušný Transducer Block.

Další objekty, definované v uživatelské aplikaci:

Link Objects

- definují spojení mezi vstupy a výstupy funkčních bloků uvnitř zařízení i napříč sítí

Trend Objects

- umožňují nadřazeným stanicím nebo dalším účastníkům sledování lokálních trendů jednotlivých parametrů funkčního bloku

Alert Objects

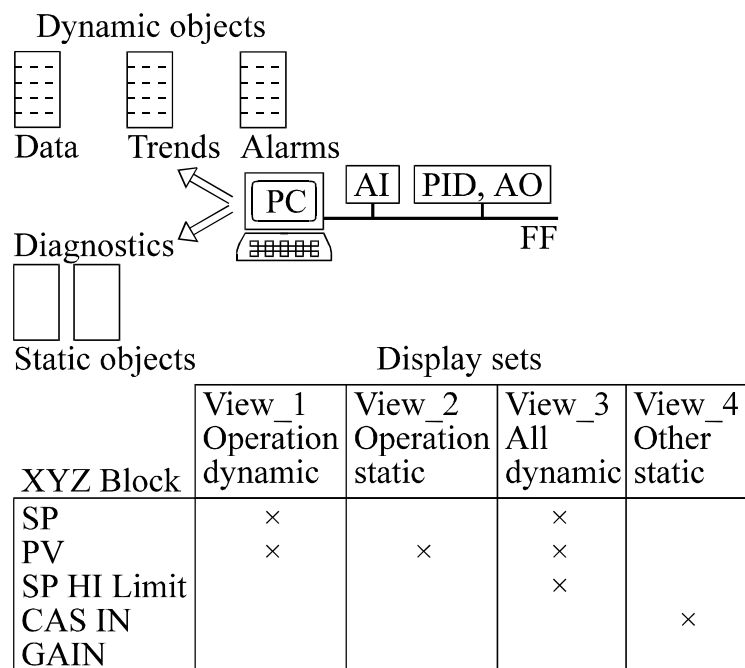
- umožňují zobrazení alarmů a událostí na fieldbusu

View Objects

- jsou předdefinované skupiny parametrů bloku, které mohou být použity v MMI. Specifikace definuje 4 atributy (view) pro každý typ bloku

Na obr. 8.15 je ukázáno využití těchto View Objects:

- View 1 - Operation Dynamic - informace, požadovaná operátorem pro běh procesu
- View 2 - Operation Static - informace která může být jednou přečtena a pak zobrazena na pozadí dynamických dat
- View 3 - All Dynamic - informace, která se mění a může být zobrazena na detailním displeji
- View 4 - Other Static - konfigurační a opravné informace



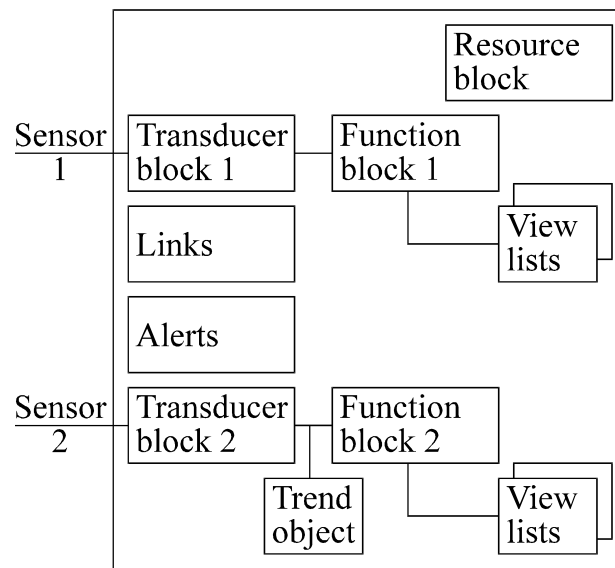
Obr. 8.15: Využití View objects

Definování zařízení (Fieldbus Device Definition)

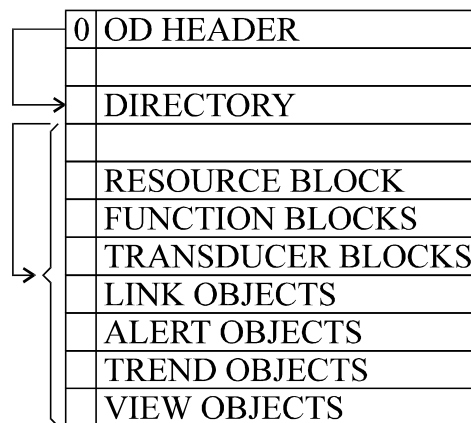
Funkce zařízení je určena konfigurací funkčních bloků a jejich vzájemným propojením (viz obr.8.16).

Prostřednictvím VFD (Virtual Field Device) je toto uspořádání viditelné. Uspořádání uživatelské aplikace je patrné z obr. 8.17

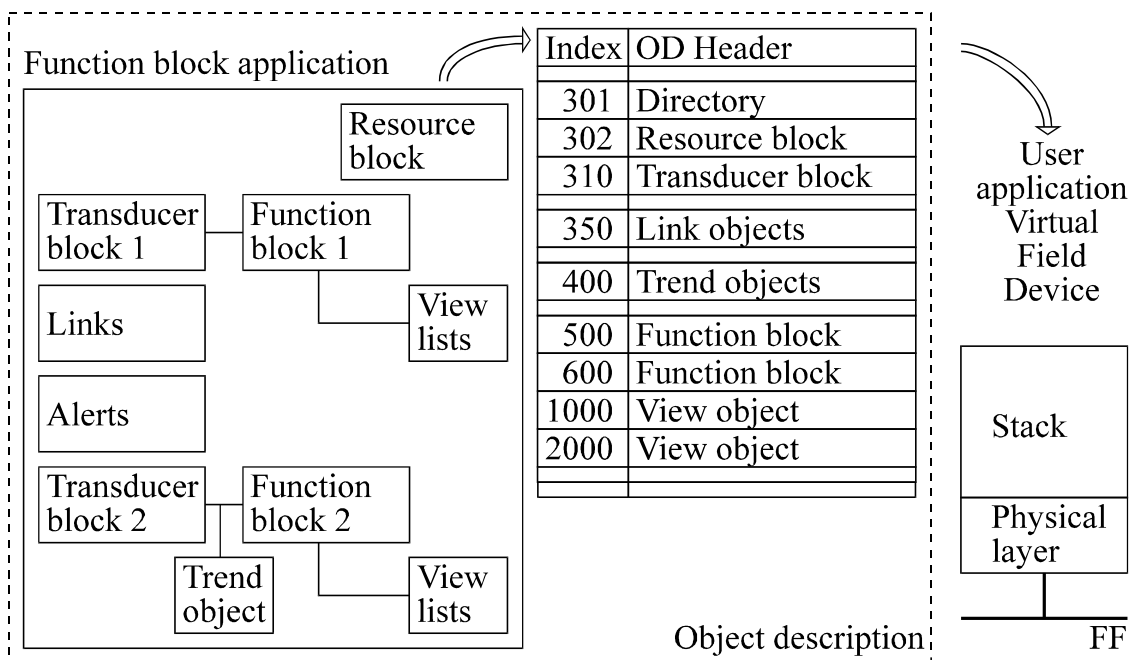
Začíná vždy hlavičkou, a následuje adresářem, ve kterém jsou uvedeny všechny další potřebné startovací indexy všech funkčních bloků. Pomocí virtuálních komunikačních vztahů VCR (Virtual Communication Relationship) je obsah VFD přístupný ze sběrnice, jak je patrné z obr. 8.18.



Obr.8.16: Propojování funkčních bloků a jejich konfigurace



Obr. 8.17: Uživatelská aplikace



Obr.8.18: Použití VFD v aplikaci

C. Správa sítě (Network management)

Jednotlivé funkce "správy sítě" zabezpečují konfigurování řadiče sběrnice (LAS) a zajišťují dále sledování činnosti, vyhledávání závad a konfigurování ostatních funkcí komunikačního zásobníku.

D. Správa systému (System management)

Funkce "správa systému" především řídí provádění (synchronizaci) funkčních bloků. Současně v systému zajišťují další důležité činnosti :

- všem zařízením předávají aktuální čas, včetně případného přepnutí na redundantní zdroj časového údaje
- automaticky předávají jednotlivým zařízením jejich adresy
- vyhledávají na sběrnici parametry podle jejich názvů (tag)
- a další.

Funkční bloky musí začít fungovat v přesně stanoveném okamžiku a v přesném vzájemném sledu. Toto je úkolem Správy sítě.

System Management synchronizuje provádění funkčních bloků a komunikaci parametrů funkčních bloků po síti.

Správa sítě také spravuje další důležité vlastnosti FF jako je sdělování údaje o čase všem účastníkům, včetně automatického přepnutí na redundantní časový

zdroj, dále automatické přidělování adres účastníkům a vyhledávání názvů parametrů nebo "tags" na síti.

Všechny potřebné informace pro funkci správy sítě jsou popsány jako objekty ve VFD každého zařízení. Tento VFD uskutečňuje přístup k informační bázi správy systému (System Management Information Base - SMIB) a také k informační bázi správy sítě (Network Management Information Base - NMIB).

Plánování (Function Block Scheduling)

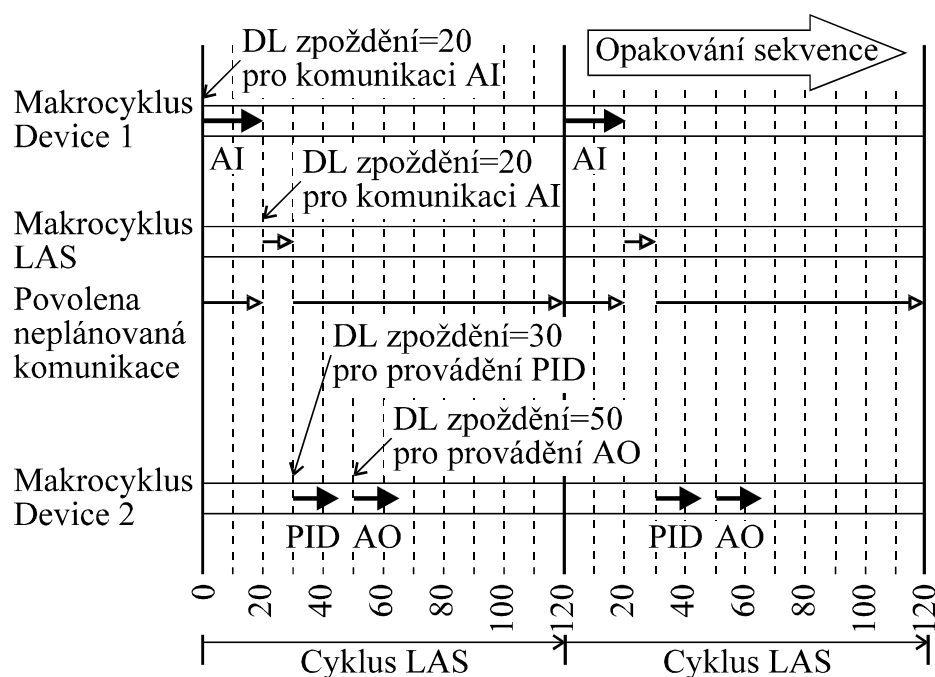
Prostředky tvorby plánování se používají ke generování funkčních bloků a plánování chodu sítě pro LAS (Link Active Scheduler).

Předpokládejme, že prostředky tvorby plánování vytvořily následující plán obr. 8.19 pro smyčku, které již byla popsána na obr. 8.14

Posun od absolutního počátku plánovaného přenosu	
Plánované rozšíření funkčního bloku AI	0
Plánovaná komunikace bloku AI	20
Plánované provádění bloku PID	30
Plánované provádění bloku AO	50

Obr. 8.19: Function Block Scheduling

Makrocyklus je jednoduchá instance plánu uvnitř daného zařízení (účastníka sítě). Následující obrázek ukazuje vztah mezi absolutním, časováním LASu, makrocyklem LASu, makrocyklem zařízení a zpožděním odstartování (start time offset).



Obr. 8.20: Správa systému

Na obr.8.20 je ukázáno, jak Správa systému (System Management) způsobí, aby se provedla funkce bloku AI a to se zpožděním 0. Dále zajistí, aby se zpožděním 20 arbir sítě (LAS) přenesl data (Compel Data) do bufferu funkčního bloku AI, umístěného v transmitteru a data v bufferu byla zveřejněna na fieldbusu.

Se zpožděním 30 způsobí Správa systému ve ventilu provedení funkce PID funkčního bloku a dále provedení funkce bloku AO se zpožděním 50. Tato časová posloupnost se stále opakuje. Její délka má název LAS Schedule duration (obr.8.20).

Poznamenejme, že během provádění exekutivy funkčního bloku, posílá LAS pověření (Pass Token) k vysílání nepotvrzených zpráv všem zařízením, takže mohou vysílat své neplánované zprávy jako třeba alarmy, nastavení žádaných hodnot operátorem ap.

Jediný čas, kdy se neplánované zprávy nedají na fieldbus vysílat je čas mezi zpožděním 20 a 30. Ten je výhradně vyhrazen publikování dat z AI na sběrnici.

Přidělení času aplikaci

FF podporuje funkci přidělování času aplikaci. Čas aplikace je zpravidla roven dennímu lokálnímu času nebo universálně koordinovanému času.

Správa systému má distributor času, který periodicky posílá zprávu o čase všem zařízením na fieldbusu. Plánovaný čas je vzorkován a posílán se zprávou o času aplikace, takže přijímací stanice mohou nastavit svůj lokální čas aplikace. Mezi synchronizačními zprávami, je aplikační čas nezávisle zpracován v každém zařízení dle jejího lokálního času.

Synchronizace aplikačního času umožňuje jednotlivému účastníku označovat data časovým údajem po celém fieldbusu. Jestliže je na fieldbusu umístěn náhradní application clock publisher, stává se aktivním LASem, jakmile základní LAS vypadne.

Přidělování adresy jednotlivým zařízením

Každé zařízení musí mít svoji jednoznačnou adresu a své fyzické označení (tag), aby byla zaručena jeho správná funkce. Aby nebylo nutné toto adresování dělat fyzicky pomocí přepínačů na jednotlivých zařízeních, může být činnost přidělování adres zařízením na sběrnici provedena automaticky Správou systému.

Sekvence přidělení adresy novému zařízení na síti se provádí následovně:

- a) Prostřednictvím konfiguratoru (fyzické zařízení na úrovni PC) je novému účastníku přiděleno fyzické označení (tag). To se provede buď " off line" na workbench nebo " on line" po sběrnici prostřednictvím speciálního „default network addresses".
- b) Prostřednictvím default network addresses požádá správa systému fyzické zařízení o jeho značku (tag). Toho využije k tomu, aby vyhledala novou adresu v konfigurační tabulce. Správce systému

pak pošle speciální zprávu "set adres" k danému zařízení, čímž ho naadresuje.

- c) Tato sekvence je opakována pro každé fyzické zařízení s default adresou.

Vyhledávací služba

Pro pohodlí nadřazeného systému, umožňuje Správa systému službu vyhledání stanice nebo proměnné prostřednictvím vyhledávání dle "tags". Dotaz vyhledání značky je vyslán všem zařízením. Jakmile je zařízení zachytí, každé prohledává své VFD pro odpověď o značce a vrací zpět kompletní údaj (pokud je "tag" nalezen) včetně síťové adresy, čísla VFD, indexu virtuálních komunikačních vztahů (VCR) a indexu slovníku objektů OD. Jakmile je jednou tato cesta nalezena, má nadřazené zařízení (host) přístup k datům pro "tag".

E. Popis zařízení (Device Description)

Kritickou věcí, kterou po komunikujícím zařízení požadujeme je jeho interoperabilita. K dosažení interoperability je využívána technika popisu zařízení (Device Description DD) spolu s definováním parametrů a chování standardních funkčních bloků.

DD provádí rozšířený popis každého objektu ve Virtual Field Device (VFD).

Device Description DD opatřuje informaci, potřebnou pro řídicí člen, k porozumění významu dat ve VFD, včetně operátorského rozhraní pro kalibraci a diagnostiku. Takže DD může být chápáno jako driver příslušného zařízení.

DD jsou podobné k driverům osobních počítačů pro různé tiskárny ap. Každý řídicí systém může s příslušným účastníkem sítě operovat, pokud vlastní jeho DD.

1. Device description Tokenizer

Popis zařízení je proveden ve standardizovaném jazyku, nazvaném Device Description Language (DDL). Tokenizer, představovaný PC, přepisuje zdrojové vstupní soubory, popsané v DD do výstupních DD souborů při přepisování klíčových slov a standardních stringů do zdrojového souboru s fixovnými "token". Jednoduchý příklad tohoto programu je patrný z obr. 8.21.

Standard FF je vybaven pro transformování všech standardních funkčních bloků a bloků typu Transducer.

Prostředí DD je dostupné na disketě u organizace FF.

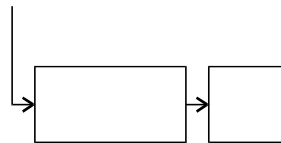
2. Device Description Service (DDS)

Popis zařízení (DD) je možné ze strany host číst pomocí knihovny funkcí, nazvané DDS. Tak se dá číst jen popis účastníka sítě, nikoli operační hodnoty. Ty se čtou z fieldbusu pomocí komunikačních služeb FMS.

DDL Source File

```
VARIABLE ProcessVariable
{LABEL "MEASURED_VALUE";
TYPE FLOAT
{DISPLAY_FORMAT "3.1f";
MAX_VALUE 110.0;
MIN_VALUE 0.0;}
```

```
}
```



Tokenizer
Tool

DD Output File

```
009 101
002 "MEASURED_VALUE"
001 010
061 "3.1f"
021 066 220 000 000
020 000 000 000 000
```

Obr. 8.21: Příklad použití DDL

3. Interoperabilita

Každý výrobce bude zasílat organizaci FF zprávu o testu interoperability svých zařízení. Zpráva se týká těchto částí:

- Obecné údaje (Universal)
- Funkční bloky (Function Blocks)
- Bloky přenosu (Tranducer Blocks)
- Specifické parametry výrobce (Manufacturer Specific Parameters)

4. Konfigurace systému (System Configuration)

Sestává ze dvou částí :

- Systémový návrh
- Konfigurace zařízení

Systémový návrh

Systémový návrh je velmi podobný současným DCS s následujícími odchylkami.

První odchylka je ve fyzickém spoji. Z proudové smyčky 4 - 20mA se přechází na napět'ové sériové rozhraní v topologii sběrnice.

Každý účastník přenosu musí mít svoji identifikační značku "tag" a odpovídající síťovou adresu.

Další odlišnost je v možnosti distribuovat po síti některé vstupní/výstupní funkce z řídicího členu jednotlivým účastníkům. To může redukovat složitost topologie procesní instrumentace.

Konfigurace zařízení

Poté, co byl systém navržen a jsou pro něj vybrána zařízení, zkonfiguruje se vzájemným propojením vstupů a výstupů funkčních bloků tak, jak to vyžaduje struktura řízení (např. dle obr.8.14).

Po zadání všech propojení bloků a ostatních konfiguračních položek (názvů zařízení, názvů smyček, periody taktování smyček ap.) vygeneruje konfigurační zařízení informaci pro každé zařízení na sběrnici. Je-li mezi zařízeními na sběrnici přístroj, schopný fungovat jako řídicí člen sběrnice (link master), mohou být jednotlivé smyčky generovány i samostatně. Taková smyčka může pracovat i bez konfiguračního HW zařízení nebo centrální ovládací konzoly. Systém se stane provozuschopným poté, co zařízení na sběrnici jsou nakonfigurována.

Testovací konfigurace FF

Zdá se, že výhody řízení procesu pomocí fieldbusů mohou být vidět z následujícího příkladu použití fieldbusu na testování procesní instrumentace.

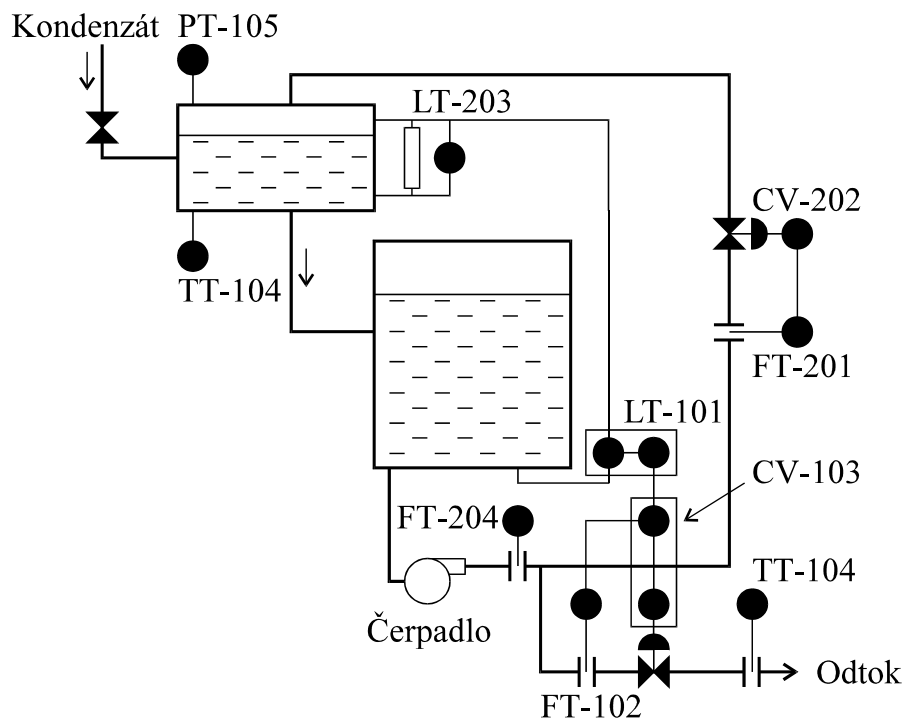
Fieldbus byl instalován na utilizační kondenzační stanici. Zkondenzovaná technologická pára se využívá k ohřevu teplé vody. Proces se skládá ze dvou tanků. První tank (flash tank) je namontován přímo na kondenzační tank. Zpětná větev vrací kondenzát zpět do flash tanku. Tam se v důsledku nižšího tlaku může stát, že kondenzát se přemění v páru. Tekutý kondenzát přetéká do nižšího kondenzačního tanku, ze kterého je pumpován do boileru (obr. 8.22).

Instrumentace pro testování systému sestává z měření:

- výšky hladiny v každém tanku a v systému tanků jako celku
- tlaku ve flash tanku
- průtoku v okruhu boileru
- celkového průtoku čerpadlem
- průtoku zpětným potrubím

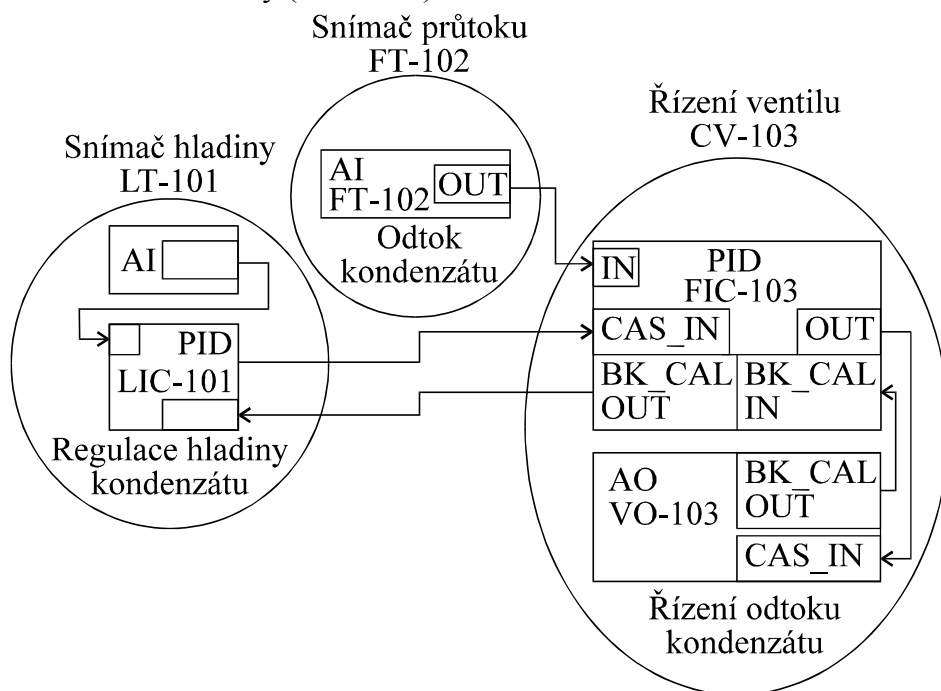
Řídicí ventily byly vybaveny číslicovými servy pro jejich ovládání tak, aby se daly řídit průtoky výstupním i zpětným potrubím. Instrumenty byly připojeny k jednomu ze dvou segmentů fieldbusu, připojených k DCS systému. Propojení bylo provedeno jednak existujícími

kabely, jednak zvláštními kabely pro tento účel. Třebaže to proces nevyžaduje, bezpečná (intrinsic) bariera byla na systému také demonstrována. Chod fieldbusu byl monitorován analyzátozem sběrnice. Propojení bylo realizováno tak, aby propojovalo procesní instrumentaci s jedním ze dvou panelů v rozvaděčové skříni (junction box), umístěné přímo v procesu (u přístrojů). Dvě linky (dva dvou vodiče) - od každého panelu jeden - byly zavedeny do řídicího centra.



Obr. 8.22: Technologické schéma testovací konfigurace FF

Celková hladina kondenzátu je řízena žádanou hodnotou PID obvodu (LIC-101), který je lokalizován do vysílače výšky hladiny. Regulační obvod výšky hladiny je brán jako základní regulační obvod kaskádního zapojení spolu s obvodem řízení průtoku (FIC-103), umístěném ve ventilu v obvodu uživatkové vody (feedwater).



Obr.8.23: Implementace systému řízení

Recirkulace kondenzátu z kondenzačního do flash tanku je řízena dalším PID obvodem (FIC-103), lokalizovaném ve ventilu.

Všechny řídicí algoritmy pro tuto kaskádní regulaci jsou plně implementovány do procesní instrumentace (bez potřeby centrálního řízení pomocí mikrokontroleru nebo PLC). Na obr.8.23 je ukázána implementace systému řízení.

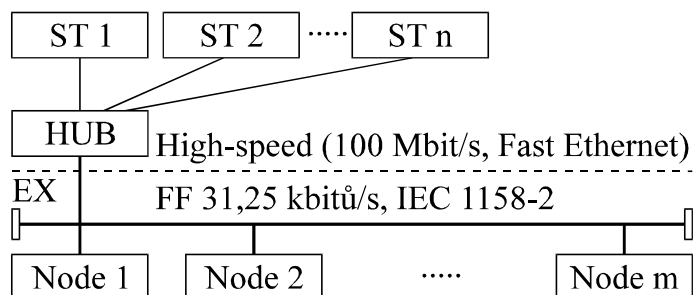
Vzhledem k poměrně velkým vzdálenostem mezi jednotlivými přístroji, dochází při instalaci totálně distribuovaného systému ke značné úspoře při kabeláži ve srovnání s původní instalací dvoubodových spojů proudovou smyčkou. Značnou úsporu je možné vyčíslit i v počtu I/O interfejsních karet. Rovněž při kontrole funkce jednotlivých přístrojů dojde s instalací fieldbusu k úspoře alespoň jednoho člověka.

Jakmile se zařízení odpojilo od busu, neovlivnilo to funkci žádného dalšího přístroje. Jestliže bylo zase zpět připojeno, systém fungoval dál beznázarově. Dále byly sníženy nároky na výkon DCS, neboť tento dílčí proces (kaskádová regulace) fungoval bez nároků na výpočetní čas DCS systému.

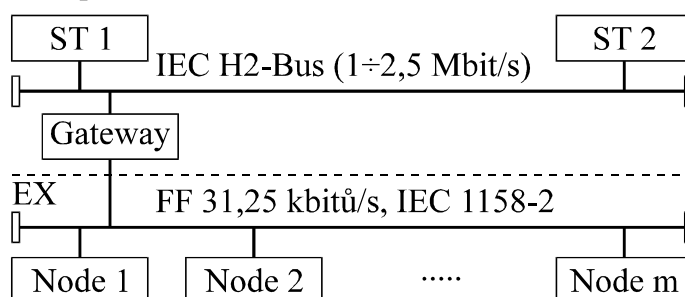
Aktuální doplněk

Organizace Foundation Fieldbus reagovala pružně na nejnovější vývoj technologie lokálních sítí, který se dá charakterizovat pronikáním technologie Ethernet i do řízení v reálném čase. Na rozdíl od IEC, která v návrhu světové normy fieldbusu na to dosud nereagovala, publikovala FF v r. 1998 strukturu, uvedenou na obr. 8.24. Staví tak rychlou variantu S.I. technologie s využitím Ethernetu pro připojení účastníků mimo S.I. oblast při zachování S.I. v provedení FF H1.

a) FF specification



b) IEC specification



Obr. 8.24: Napojení FF na průmyslový Ethernet

11.2 Úvod do dalších průmyslových komunikačních systémů

Jak již bylo řečeno v předešlém, v současné době existuje velké množství průmyslových komunikačních systémů (fieldbusů a nižších průmyslových sériových sběrnic a interface). Důvody tohoto vývoje jsou známy a jsou uvedeny v literatuře [17] a další.

Je nad rámec tohoto učebního textu probrat do detailů všechny průmyslově používané komunikační systémy. Zaměříme se proto jen na evropské standardy. Ani tam nelze podat vyčerpávající výklad.

Evropské i zámořské firmy v oblasti automatizace začaly vyvíjet a velmi brzo na to i vyrábět a používat fieldbusy v první polovině osmdesátých let. K dnešnímu dni jsou standardizovány (norma DIN, ISO, IEEE) tyto fieldbusy. Interbus-S firmy Phoenix, Profibus (Siemens a FZI Karlsruhe), FIP (Francie), CAN (Bosh-Intel), Bitbus (Intel), P-net, DeviceNet (Allen-Bradley), LonTalk (Echelon-Toshiba-Motorola), AS-interface, BacNet (ASHRE-USA). Kromě těchto de-facto standardů existuje ještě několik čistě firemních fieldbusů, které pro jejich vynikající vlastnosti (vysoký stupeň integrace, cena, výkon, vazba na velkou firmu) používají ve svých výrobcích další firmy v oblasti průmyslové automatizace. Je to především Modbus (s variantou J-bus) a Modbus Plus firmy Modicon. Kromě toho existuje ještě mnoho jen firemních fieldbusů, které používá jen daná firma pro svoje úspěšné produkty, Data Highway plus (Allen-Bradley), Sinec L1 (Siemens) a další. Je však i celá řada fieldbusů, které byly vyvinuty, avšak neuplatnily se v silné konkurenci nebo teprve na svoji budoucnost čekají.

Faktem zůstává, že prvním velkým nasazením fieldbusu byl Bitbus firmy Intel a krátce nato Interbus-S firmy Phoenix.

Nejvýznamnější evropskou iniciativou v této oblasti v druhé polovině 80.let byl však vývoj evropského fieldbusu. Vývoj byl od počátku poznamenán národním soupeřením Německa a Francie a tak začátkem 90. let vznikly dva "národní standardy". Německý Profibus (Process Fieldbus) a francouzský FIP (Factory Industrial Protocol). Zatímco Profibus byl vyvíjen v Německu s podporou Ministerium für Forschung und Entwicklung, FIP byl vyvinut francouzskými firmami s podporou Francie. Oba standardy byly dotazeny do průmyslového nasazení a rozšíření jak v zemi svého původu, tak v dalších zemích. V Dánsku vyvíjený P-Net našel uplatnění rovněž v mnoha průmyslových aplikacích. Všechny tři návrhy v nepozměněné podobě byly v průběhu 90. let prohlášeny za evropské standardy a normovány normou EN 50170 Díl1 (P-Net), Díl2 (Profibus FMS,DP) a Díl3 (FIP). Na americkém kontinentu dosáhly oba evropské standardy Profibus a FIP značného rozšíření a komerční úspěch. Postupem času a pod tlakem IEC bylo vytvořeno několik iniciativ (PNO, ISP, WordFIP, Foundation Fieldbus), které se snažily přiblížit se požadavkům IEC na jednotný světový fieldbus. Největší implementaci požadavků IEC dosáhl FF (Foundation Fieldbus). Slouží zároveň jako příklad kompromisního řešení, sdružujícího přínos Profibusu i FIPu k vývoji světového fieldbusu. Promě toho může podrobnější výklad principů FF ukázat všechny důležité poznatky o průmyslovém komunikačním standardu.

11.2.1 Profibus

Profibus (Process Fieldbus) byl vyvinut s podporou několika významných německých firem pod koordinací DBFT (Deutsches Bundesministerium für Forschungs und Technik). Stal se německou normou DIN 19245. Jak již bylo řečeno, pro potřebu sítí typu fieldbus i

PROFIBUS implementuje pouze vrstvy 1,2 a 7 modelu OSI. I když je podporován integrovanými obvody, umožňuje i svoji programovou implementaci na společných CPU.

Fyzickým médiem je stíněná kroucená dvoulinka s rozhraním RS 485 s max. délkou 1200m (do 4800m s opakovači) nebo optické vlákno. Pro variantu PA pak proudová smyčka. Přenosová rychlost je normována na 9,6, 19,2, 187 a 500kbitů/sec. Později byla rychlost upravena na širokou škálu rychlostí 1,5 M, 3M,6M, 9M a 12Mbitů/s a více (Profibus DP). Profibus PA má nastavenou jedinou rychlost dle normy IEC 1158-2 tedy 31,25kbitů/s.

Jednotlivé vrstvy zjednodušeného OSI modelu definují:

1. Vrstva fyzická : fyzické spojení mezi účastníky, mechanické, elektrické vlastnosti, topologii sítě, rozhraní. Pro přenos po sběrnici (kroucená stíněná dvoulinka) využívá RS 485 s nestandardně definovanými parametry [18]. Dále podporuje přenos světlovodičem a ve výbušné zóně pro variantu Profibus PA pak standard IEC 1158-2.

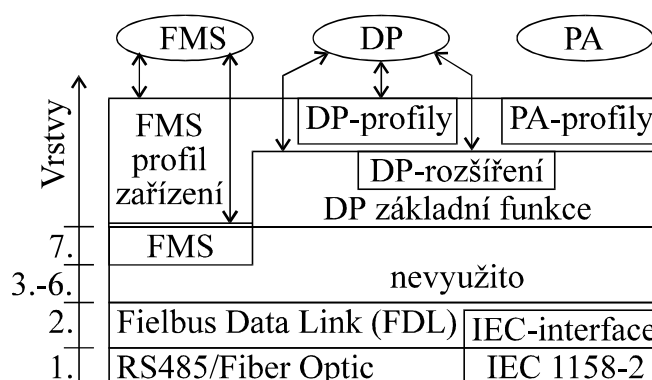
2. Linková vrstva je vrstva přístupu k síti a zabezpečení tvorby zprávy, tvořené tokem bitů. Profibus zde využívá hybridní verze IEEE 802.4 Token bus s přístupem Master Slave.

7. Aplikační vrstva poskytuje nezbytné služby pro komunikaci. Dělí se na dvě podvrstvy, označené shora jako FMS (Fieldbus Message Specification) a LLI (Lower Layer Interface).

Nad 7. vrstvou je definováno komunikační rozhraní ALI (Application Layer Interface), kterým jednotlivá komunikující zařízení přistupují ke komunikačnímu modelu (k 7. vrstvě).

Existují tři varianty Profibusu. Profibus DP, Profibus FMS a Profibus PA.

Nejjednodušší a nejrozšířenější variantou je Profibus DP, určený pro rychlou komunikaci typu Master-slave. Je vhodný zejména pro rychlý přenos signálů z procesu pomocí decentralizovaných periférií a odloučených I/O jednotek. Komunikace je po RS 485 nebo světlovodičem rychlostmi do 12 Mbitů v široké nabídce možností. Přenos se uskutečňuje prostřednictvím jednoduchého souboru bloků a funkcí.

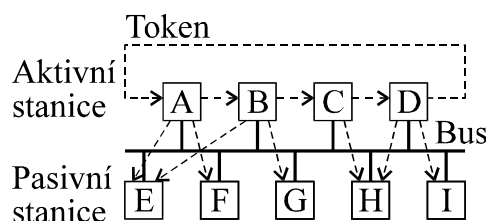


Obr. 8.25: Architektura protokolů Profibus

Pro komunikaci v heterogenním prostředí na vyšší úrovni je vhodný Profibus FMS s širokou nabídkou služeb pro práci s daty, programy, alarmy. Přenosové médium je shodné s variantou DP. Rychlost je nižší.

Profibus PA zatím existuje jen jako norma DIN 19245. Je určen pro řízení procesů, včetně prostředí s nebezpečím výbuchu. Do fyzické vrstvy proto implementuje standard IEC 1158-2 (proudová smyčka). Obr. 8.25 porovnává všechny tři varianty Profibusu.

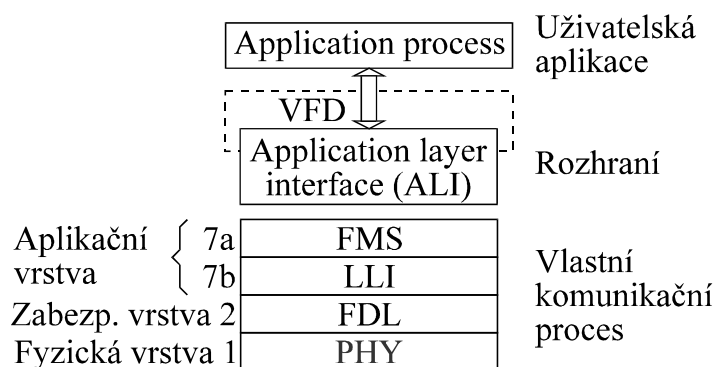
Je-li třeba připojit více účastníků přenosu sítí Profibus, než je 32 stanic, povolených standardem RS 485, propojují se jednotlivé segmenty mezi sebou opakovači až do maximálně 10 segmentů při celkovém počtu stanic max. 127. Každý segment vyžaduje zakončení aktivními terminátory.



Obr. 8.26 Princip přístupu k síti u sítě Profibus

Stanice, připojené k Profibusu se dělí na pasivní a aktivní stanice. Aktivní stanice si předávají token a mohou vysílat, když Token vlastní. Pasivní stanice nemohou vlastnit token a musejí být adresovány některou aktivní stanicí, aby mohly předat data. Způsob propojení stanic a předávání pověření a zpráv je patrné z obr. 8.26 Stanice A, B, C, D jsou aktivní stanice (Master) a jen mezi nimi může cirkulovat pověření (token). Stanice E, F, G, H, I jsou stanice pasivní a jsou vyzývány příslušným Masterem k předání dat.

Každá aktivní stanice zná adresy svých aktivních sousedů. T.j. ví od koho může dostat token a komu token předat. Doba, po kterou aktivní stanice vlastní token je parametrovatelná a celková doba cyklu sítě může být nastavena.



Na obr. 8.27 je znázorněn komunikační model PROFIBUSu

PROFIBUS definuje 4 komunikační služby pro přenos dat. Je to:

- SDN Send Data with No Acknowledge
- SDA Send Data with Acknowledge
- SRD Send and Request Data
- CSRD Cyclic Send and Request Data

Důvodem pro tak širokou nabídku komunikačních služeb je použití fieldbusu jednak pro rychlý cyklický přenos malého objemu dat (datová informace z čidel a na akční členy) a současně fieldbus musí být vhodným prostředkem pro přenos velkého objemu dat při parametrování a při výměně velkého objemu dat ve vyšších hierarchických úrovních řídicí architektury.

Pro přenos zpráv jsou u PROFIBUSu využity obvykle osvědčené způsoby kódování a přenosu zpráv. Vlastní kódování proudu bitů je typu NRZ. Pro přenos znaku je použito 11-ti bitové kódování znaků typu UART s jedním start bitem, jedním stop bitem, 8 informačními bity

Formát s pevným informačním polem bez datového pole



Formát s pevným informačním polem a datovým polem 8 byte



8 Byte

Formát s proměnnou délkou informačního pole



max.246 Byte

Obr.8.28: Formáty telegramu Profibusu

a jedním paritním bitem (sudá parita) dle normy DIN 19244. Formáty telegramu PROFIBUS jsou ukázány na obr.8.28.

Zde:

SD1 - SD3 Start Delimeter (variantu 1 až 3 pro rozlišení Formátu)

LE/LE_R Length/Length with repetiton, udává počet Byte v informačním poli u telegramu s proměnnou délkou. Z důvodu zabezpečení informace je tento údaj opakován.

DA Destination adress

SA Source adress

FC Frame control (udává typ telegramu jako výzva, potvrzení, řídicí informace ap.)

FCS Frame Check Sequence

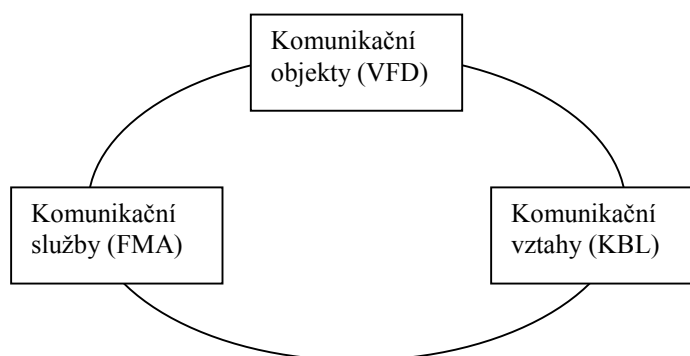
ED End Delimiter

Jak již bylo řečeno, u PROFIBUSu nejsou vrstvy 3 - 6 ISO/OSI modelu explicitně vyjádřeny. Aplikační vrstva 7 má tedy přístup přímo na vrstvu 2. Toto připojení je uskutečněno pomocí LLI (Lower Layer Interface), které je součástí 7. vrstvy, ve které dochází k zobrazení služeb vrstvy 7 do vrstvy 2, protože LLI plní i nezbytné funkce vrstev 3 - 6. Při tvorbě komunikačního modelu PROFIBUSu byly zohledněny následující aspekty. Na jedné straně omezený rozsah funkcí automatizačních prostředků v nejnižší úrovni řízení (čidla, akční členy) a dále nutnost propojení PROFIBUSU směrem k hierarchicky vyšším vrstvám řídicí struktury (nadřazené operátorské stanice ap.). Oba tyto protichůdné aspekty řeší FMS (Fieldbus Message Specification), která tvoří horní podvrstvu aplikační vrstvy PROFIBUSu. FMS je definována jako podmnožina MMS (Manufacturing Message Specification) standardu MAP (Manufacturing Automation Protocol). Cílem komunikačního modelu PROFIBUSu je sjednotit distribuované aplikační procesy do jednoho procesu. FMS je objektově orientovaný systém. Reálné objekty jako kupř. stav sepnuto proximity senzoru nebo hodnota teploty na čidle teploty jsou přeměněny na komunikační objekty. FMS popisuje tvorbu, způsob popisu těchto objektů a služby, definované nad těmito objekty.

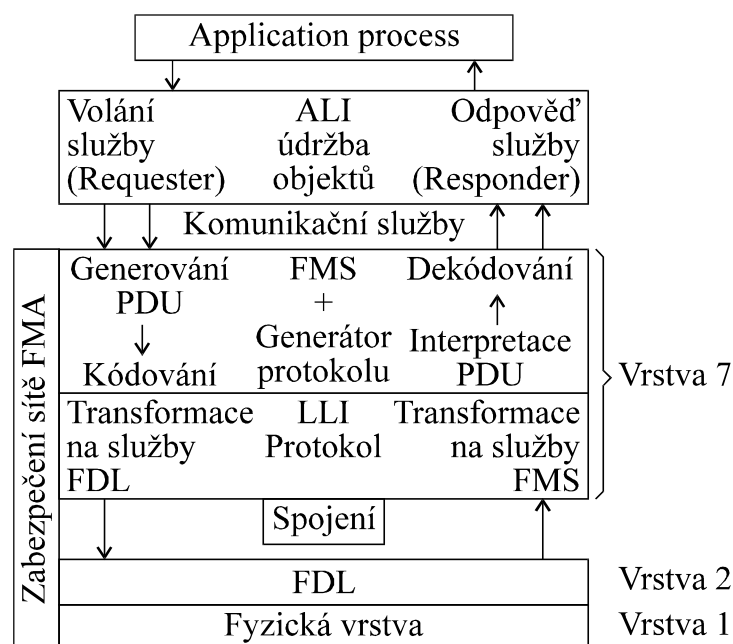
Funkce aplikační vrstvy jsou vyjádřeny třemi komunikačními elementy. Jsou to:

- komunikační objekty
- komunikační služby
- komunikační vztahy

Komunikačními objekty jsou kupř. data z procesu, data pro akční členy, podprogramy, parametry. Tyto objekty mohou využívat účastníci komunikace pomocí komunikačních služeb. Komunikační vztahy jsou pak logické spoje mezi účastníky komunikace. U PROFIBUSu jsou tyto vztahy definovány tabulkou komunikačních vztahů KBL. Komunikační služby jsou operace na objektech definované třídy. Určité třídě objektů jsou povoleny určité služby. Kupř. změřená veličina z čidla je objekt (ze třídy proměnných), na které je povolena operace (služba) čtení a zápis.



Obr. 8. 29x Vzájemné vazby komunikčních komponent



Obr.8.29: Členění komunikačních služeb

Všechny služby 7. vrstvy jsou uživateli k dispozici pomocí aplikace služeb - FMA (Fieldbus Management Application). Členění komunikačních služeb PROFIBUSu je ukázáno na obr.8.29.

Rozhraní mezi aplikačním procesem a aplikační vrstvou komunikačního modelu ISO/OSI tvoří Application Layer Interface ALI. Aby systém byl dostatečně otevřený, je v tomto rozhraní definován otevřený seznam abstraktních objektů. Tuto část rozhraní nazývá PROFIBUS Virtual Field Device (virtuální zařízení VFD). Kromě virtuálních zařízení obsahuje ALI ještě správu služeb a správu objektů.

Až na malé výjimky (broadcasting a multicasting SDN) probíhá komunikace po Fieldbusu mezi dvěma účastníky přenosu. Přitom vždy jeden z účastníků požaduje služby (client) od druhého účastníka (server). Mezi Client a Server probíhá abstraktní logický kanál. Každý účastník může současně navazovat tyto logické spoje s více účastníky. Každý tento komunikační spoj je definován souborem parametru. K těmto parametrům patří kromě adresy partnerské stanice a bodem přístupu ke službě ještě seznam služeb, které toto spojení může poskytovat a dále délka posílané zprávy a její odpověď.

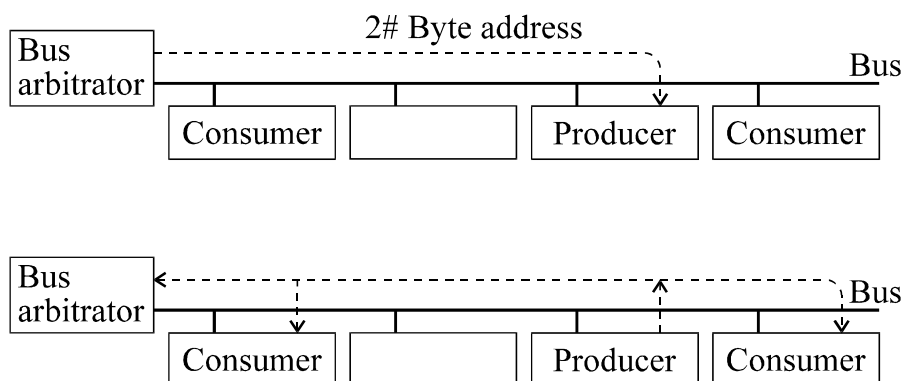
Další použití PROFIBUSU

V oblasti průmyslové výroby, zejména při sběru dat z výrobních linek a ovládání jednoduchých pohonů a dalších akčních členů se využívá ve zvýšené míře PROFIBUS DP, který na rozdíl od normy 19245 (PROFIBUS FMS) připouští pouze přístupovou metodu master-slave, avšak vykazuje zvýšenou rychlost přenosu na 12 Mbitu/sec, což je v daných aplikacích

nezbytné. Významnou vlastností PROFIBUSu DP je příprava přenášených dat ve formě cyklického datového obrazce, který je zpracováván objektově orientovaným rozhraním FMS. Na místo 7. vrstvy pak vystupuje uživatelský interface, který popisuje standardizované využití a dále DDLM (Direct Data Link Mapper), který realizuje rozhraní na 2. vrstvu. V oblasti řízení technologických procesů, včetně výbušné zóny je připraven PROFIBUS PA. Fyzická vrstva a použité médium odpovídá normě IEC 1158-2.

11.2.2 FIP

Skupina francouzských, německých a italských firem vyvinula fieldbus, který se stal standardem především ve Francii. Na severamerické kontinentu se uplatnil díky organizaci WorldFIP, která později spolu s PNO (Profibus Nutzer Organisation) definovala ISP



Obr. 8.30: FIP (Factory Instrumentation Protocol).

(Interoperability System Project), který slučuje oba komunikační modely v jeden celek a měla se maximálně přibližovat požadavkům ISO na světový standard fieldbusu. FIP používá kroucenou dvoulinku a přenosovou rychlost 31, 25kbitů/sec., 1 Mbit/sec. a 2,5Mbitů/sec. Při rychlosti 1Mbit/sec. je povolena délka segmentu 500m. Jednotlivé segmenty mohou být propojeny navzájem pomocí opakovačů. Každý segment je určen maximálně pro 32 účastníků. Celkem k FIPu může být připojeno 256 účastníků. Interface je integrován v obvodu FULLFIP firmy CEGELEC. Tento obvod má interface na PC a na druhé straně na FIB bus. FIP používá t.zv. zdrojově orientovanou pooling adresovací metodu. Řízením sběrnice je pověřen arbitř, který cyklicky pověřuje ostatní stanice, které chtějí data buď přijímat nebo vysílat. Arbitř vlastní seznam všech proměnných, které jsou používány ve všech připojených stanicích a každá proměnná je identifikována vlastní 16ti bitovou adresou (tedy celkem 65536 identifikátorů). Když arbitř vyšle adresu na sběrnici, zařízení, které ji rozpozná odpoví a pošle zpět obsah příslušné proměnné. V tom stejném okamžiku přijímač, který rovněž rozezná adresu této proměnné si zapamatuje její hodnotu, kterou čte z busu (viz obr.8.30).

Několik stanic (účastníků) může ve stejném okamžiku sloužit jako přijímací stanice a dostávat identickou informaci. Arbitř průběžně šíří (pools) všechny proměnné ze všech připojených stanic. Efektem je distribuovaná databáze, ve které každá proměnná může být buď přístupná prostřednictvím svého identifikátoru nebo se její hodnota periodicky objevuje na sběrnici. Výhodou tohoto způsobu je i to, že tím, že arbitř updatuje data, zvyšuje jejich konzistenci. Operace dále mohou být definovány tak, že fixní (známý) čas je použit pro "variable

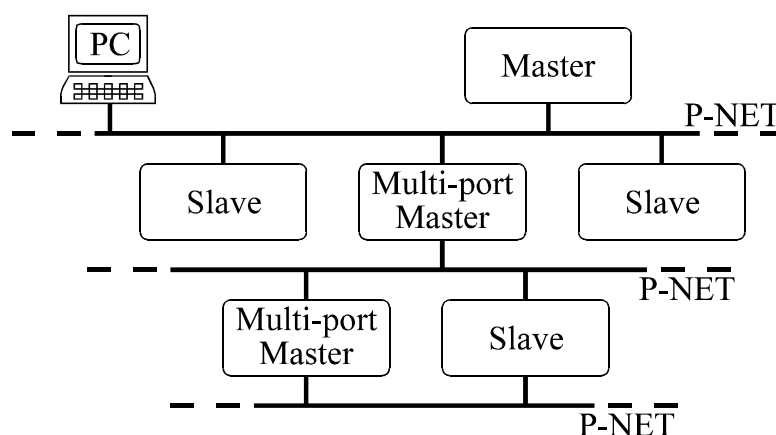
pooling", zatímco asynchronní operace jsou proměnné délky. Tím je zaručeno, že sbíraná data jsou vždy ve stejné fázi (není tam žádný fázový posun).

Zprávy ve FIPu obsahují adresu nebo data a identifikační kód datového typu. Tvar rámce je podobný rámci HDLC formátu s úvodní posloupností (preamble), checksum (kontrolním součtem) a koncovou křídlovou značkou (flag). V základním operačním módu není potřeba potvrzování zpráv, protože data v reálném čase jsou periodicky updatována ve velmi krátkém čase. FIP má rovněž funkce pro síťové operace, t.j. start a stop odloučených úloh (remote tasks).

11.2.3 P-Net

Dánský standard fieldbusu, je zahrnutý do evropské normy EN 50170 jako Díl1. P-Net je určen především k automatizaci procesů. Umožňuje vytváření velkých propojených sítí. P-net dělí účastníky přenosu na zařízení podřízená a na zařízení, označovaná jako řídící. Podřízená zařízení mají vlastní inteligenci pro zpracování vlastního uživatelského programu, který je psaný ve speciálním assembleru a nahraje se z vývojového prostředí do zařízení a tam se může dále editovat a krokovat (v některých zařízeních). Zařízení dále pracují se vstupními a výstupními údaji jako s objekty. Údaje si ponechávají fyzikální charakter (napětí proud, délka ap.).

Řídící stanice (zařízení) vykonávají vlastní funkci řízení procesu a dále mohou fungovat jako směrovače zpráv a propojovat jednotlivé segmenty sítě. Topologie sítě P-net je ukázána na obr. 8.31



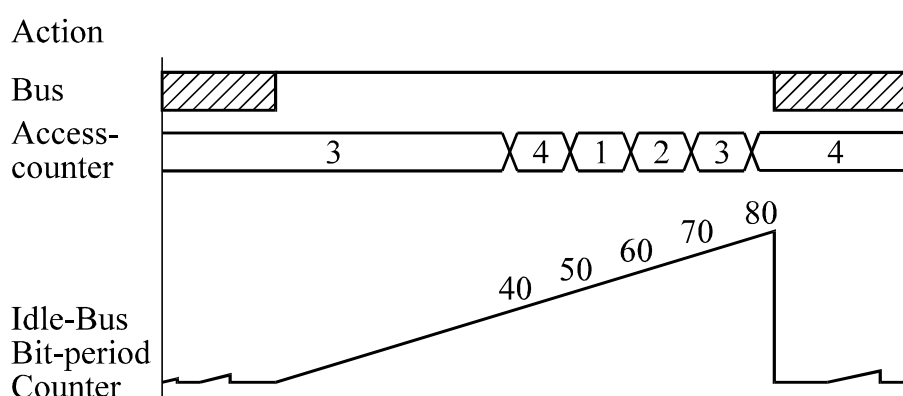
Obr. 8.31: Topologie sítě P-net

Sběrnice P-net je typu master -slave, ale umožňuje rovněž režim multimaster. Tato zařízení se programují v jazyku Process Pascal, což je Pascal, rozšířený o prostředky pro řízení procesů (spouštění několika procesů, přístup k proměnným ostatních účastníků ap.). P-net je podporován SW VIGO pro prostředí Windows. Může fungovat jako OLE server a sloužit tedy jako SW rozhraní k standardním produktům jako Excel, Visual Basic, Visual C++ ap. Vzhledem k tomu, že při definování P-netu byl kladen důraz zejména na segmentování a propojování většího počtu segmentů do jedné sítě, podporuje protokol P-net 1. až 4. a 7.vrstvu OSI modelu.

Fyzická vrstva P-netu používá rozhraní RS 485, avšak v uspořádání fyzický kruh. Tak je možné propojit rozhraním RS 485 i větší počet účastníků, než 32 na jednom segmentu (128 vzhledem k délce adresy 7 bitů). Další parametry segmentu sítě jsou tato:

- médium: kroucená stíněná dvoulinka
- délka segmentu: 1 200m
- počet zařízení : 125
- přenosová rychlost: 76,8kbitů/s pevně nastavená
- kódování: NRZ

Princip přístupu k médiu je unikátní. Každé z 32 řídicích zařízení, která mohou být na segmentu definována, má svoji adresu 1 až 32. Má svůj čítač hodinové frekvence (bitová frekvence) a čítač přístupu k síti. Čítač přístupů přičítá jedničku, dosáhne-li čítač bitové frekvence hodnot 40, 50, 60, atd. Porovnává také svoji adresu s čítačem přístupů. Jsou-li oba údaje shodné, může přistoupit k řízení sítě. Jakmile čítač přístupu načítá hodnotu větší, než je



Obr.8.32: Mechanismus přístupu ke sběrnici

počet řídicích stanic, připojených k síti, vynuluje se (obr. 8.32)

Podřízené účastníci mohou přistupovat k síti v čase mezi 11. a 30. bitovým intervalem po přijetí požadavku od nadřízené stanice (řídicí stanice). Toto časové okno představuje jen $30 \times 13\mu\text{s} = 390\mu\text{s}$ (při přenosové rychlosti 76 800 bitů/sec je bitový interval $13\mu\text{s}$). Podřízené stanice proto musejí odpovídat velmi rychle. Tím se zvětšuje celková propustnost sítě i při zdánlivě nízké rychlosti přenosu ve srovnání s jinými fieldbusey.

Data se po síti přenášejí v rámcích s adresou zdroje i příjemce, řídicím a chybovým bytem a s kontrolním polem. Tvar rámce je na obr.8.36

Adresa	Řízení/ Status	Délka inf.rámce	Inform. rámec	Zabezpečení
2÷24 byte	1 byte	1 byte	0÷63 byte	1÷2 byte

Obr. 8.33: Tvar rámce protokolu P-net

Byte jsou přenášeny asynchronně s jedním start bitem, osmi významovými bity, jedním bitem, určujícím, zda jde o adresu nebo data a jedním stop bitem. Mezi Byte rámce není mezera.

Rámce jsou odděleny mezerou minimálně 11 bitových intervalů. V poli adresy má jeden bit hodnotu 1, což umožňuje orientaci nově připojenému zařízení nebo zařízení, které vypadlo ze synchronizace, opět se připojit k přenosu. Jestliže jsou zařízení na jednom segmentu, obsahuje rámec dat jen jednu adresu zdroje a jednu adresu cíle zprávy. V případě složitějších topologií se počet adres zvětšuje. Adresní pole má až 24 byte. Bity 0 - 6 každého byte adresového pole udávají vlastní adresu a bit 7 rozlišuje adresu cíle a zdroje.

Ve složitější topologii, pro kterou byl P-net vyvinut se uplatní i funkce síťové i transportní. Aplikační vrstva má obvyklé určení jako u jiných fieldbusů.

Výhodou sítě P-net je její snadná konfigurovatelnost a tedy i nízké náklady a kratší čas při uvádění do chodu. Vzhledem k pevně nastavené rychlosti se konfiguruje jen struktura sítě. Předpokladem je, že zařízení jsou dostatečně inteligentní, aby se po síti předávaly jen povely a měřené a zpracované hodnoty (zpracované ve vhodném tvaru a jednotkách - proto je zachována fyzikální podstata přenášených dat). Nevýhoda nižší přenosové rychlosti je do jisté míry kompenzována výše uvedeným mechanismem přístupu k síti.

11.2.4 Protokol CAN (Controler Area Network) [14]

11.2.4.1 Úvod

Velmi populární a rozšířený sériový komunikační prostředek vhodný pro průmyslová použití. Vyvinut firmou Bosch na technologii Intel původně pro potřeby automobilového průmyslu. Pro svoji otevřenost (protokol první a druhé vrstvy je k dispozici každému výrobcí) a pro relativně vysokou rychlost a podporu mnohými výrobci mikroelektronických součástek, kteří implementují protokol do svých mikrokontrolerů, je velmi rozšířen. Dvě první vrstvy jsou definovány standardem ISO 11898. Nad těmito definovanými vrstvami postupem času bylo vytvořeno několik standardů aplikační vrstvy (vrstva 7 modelu ISO/OSI). Tak vznikly standardy dalších průmyslových sítí jako DeviceNet, CANopen, SDS ap. Sama norma CAN má dvě varianty a to variantu CAN 2.0A a CAN 2.0B s rozšířeným formátem zprávy, vzhledem k tomu, že uživatelé měli výhody k relativně kratšímu datovému rámci v některých průmyslových aplikacích.

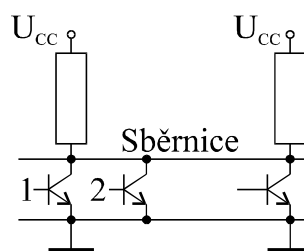
11.2.4.2 Základní vlastnosti

System umožňuje časově nedeterministické řízení v reálném čase. Tomu je přizpůsobena jak rychlost (do 1 Mbitu/s), tak přístupová metoda (multimaster). Znamená to, že každý účastník sítě může být na určitou dobu řídicím členem. Tato metoda je kombinována s náhodným přístupem k síti, která řeší kolize na základě prioritního rozhodování. Komunikace na síti probíhá prostřednictvím předávání zpráv (datová zpráva a žádost o data). Řízení sítě (jako signalizace chyb, pozastavení komunikace) je pak zajištěno pomocí dvou speciálních zpráv (chybové zprávy a zprávy o přetížení).

Zpráva, vysílaná po sběrnici neobsahuje adresu příjemce a je přijímána všemi ostatními zařízeními. Každá zpráva je uvozena identifikátorem nesoucím informaci o významu zprávy a její prioritě. Nejvyšší prioritu má zpráva s identifikátorem 0. CAN protokol zajišťuje, aby zpráva s nejvyšší prioritou byla v případě kolize přenesena přednostně. Pomocí identifikátoru je možné zajistit, aby daná stanice přijímala jen ty zprávy, které se jí týkají (acceptance filtering).

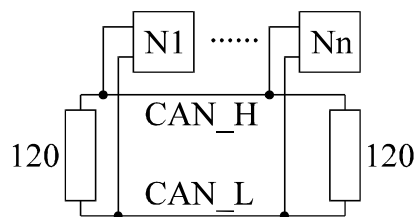
11.2.4.3 Fyzické médium a fyzická vrstva

Protokol CAN definuje vlastní rozhraní k fyzickému přenosovému médiu a v tomto smyslu se odchyluje od modelu ISO/OSI. Na druhé straně jsou vlastnosti fyzické vrstvy velkou předností protokolu CAN. Požadavkem pro realizaci je, aby ve fyzické vrstvě byl realizován logický součin. Za účelem zvýšení rychlosti a odolnosti vůči rušení je účelné, aby spoj byl symetrický. CAN proto definuje dvě vzájemně komplementární hodnoty bitů na sběrnici dominant a recessive. Jedná se o plovoucí stejnosměrné úroveň, jejichž stejnosměrná hodnota je dána použitým médiem a vůbec konkrétní realizací fyzické vrstvy (nemusí jít ani o napěťový signál, jak je tomu v případě světlovodiče jako přenosového média). Funkce sběrnice musí být následující: Jestliže všechna připojená zařízení vysílají bit recessive, na sběrnici musí být úroveň recessive. Jestliže alespoň jedno zařízení vysílá bit dominant, musí být na sběrnici úroveň dominant. Tato funkce se dá ukázat např. na světlovodiči, kde úroveň recessive je nesvítlí a úroveň dominant je svítí. Tedy dominant je vždy, když alespoň jeden zdroj má bit dominant, tedy svítí.



Obr. 8.34: Metoda přístupu ke sběrnici

Jestliže žádný ze vstupů tranzistorů (jako výstupních členů interfejsů jednotlivých účastníků k médiu) nemá log.1, sběrnice má vysokou úroveň. Jestliže však libovolný ze vstupů tranzistorů má úroveň log.1, celý výstup (sběrnice) má úroveň log.0.



Obr. 8.35: Fyzické uspořádání sítě CAN

Protocol CAN je ve fyzické vrstvě definován mimo jiné standardem ISO 11898. Fyzické uspořádání sítě CAN je pak na obr. 8.35

Úroveň recessive a dominant jsou definovány rozdílovým napětím mezi vodiči CAN_H a CAN_L. Pro úroveň recessive to je 0V rozdílového napětí, pro úroveň dominant to jsou 2V rozdílového napětí. Sběrnice musí být přizpůsobena na obou koncích zakončovacími odpory 120 ohmů.

Ke sběrnici může být teoreticky připojen libovolný počet účastníků, prakticky s ohledem na zatížení sběrnice, je počet účastníků podstatně nižší a uvádí se kolem 64 na segment. Rovněž přenosová rychlost 1 Mbit/s je dosažitelná jen na krátké vzdálenosti do 40m a se vzdáleností prudce klesá, takže na 1,2km činí asi 70 kbitů/s. Plyne to z původního poslání sběrnice CAN, která byla určena pro malé vzdálenosti v instalaci automobilů.

11.2.4.4 Linková vrstva CAN

Tak jako v modelu ISO/OSI i v protoku CAN je linková vrstva rozdělena na podvrstvu LLC a MAC:

- MAC (medium access control) zabezpečuje kódování dat a přístup účastníka k médiu, dále vkládání a odstraňování doplňkových bitů (stuffing/destuffing), detekci chyb, hlášení chyb a potvrzování správně přijatých zpráv
- LLC (logical link layer) je podvrstva řízení datového spoje, což zde znamená filtrování přijatých zpráv (acceptance filtering) a hlášení o přetížení (overload notification)

Řízení přístupu k médiu a řešení kolizí

Vzhledem k tomu, že jde o síť typu multimaster, každý z účastníků může zahájit vysílání, jakmile je připraven a na síti je klid (bus free). Kdo přijde první, ten vysílá. Ostatní mohou vysílat až poté, co je zpráva odvysílána. Výjimku tvoří chybové rámce, které se dají vysílat okamžitě po identifikaci chyby kterýmukoli účastníkem.

Pokud začne současně vysílat několik účastníků, získá přístup k médiu ten, který má vyšší prioritu, t.j. nižší identifikátor, uváděný na začátku zprávy. Vysílač každého účastníka porovnává hodnotu právě vysílaného bitu s hodnotou na sběrnici a zjistí-li, že na sběrnici je jiná hodnota než vysílá (to se dá poznat jedinečně tehdy, kdy vysílač vysílá recessive bit a na sběrnici se objeví dominant bit). V tom okamžiku přerušuje vysílání. Tak je zajištěno, že zpráva s vyšší prioritou je odvysílána jako první a nedojde k jejímu přerušení, což by znamenalo nové vysílání a tím prodloužení doby přenosu zprávy. Zařízení, které se nedostalo k odeslání zprávy může začít vysílat, jakmile je bus free.

Zabezpečení

Protokol CAN je silný v mechanismu zabezpečení přenosu. Současně působí tyto mechanismy :

- monitoring
- CRC kód
- vkládání bitu
- kontrola zprávy
- potvrzení přijaté zprávy

Monitoring:

Monitoring znamená, že vysílač porovnává vysílanou hodnotu bitu s úrovní na sběrnici. Jsou-li hodnoty stejné, vysílá další bit. Jakmile je rozdíl mezi těmito dvěma hodnotami a probíhá-li právě fáze řízení přístupu ke sběrnici, přeruší se vysílání a přístup získá stanice s vyšší prioritou. Jestliže je neshoda indikována v jiném okamžiku, než při řízení přístupu k médiu nebo ACK, hlásí se chyba bitu.

CRC kód:

CRC kód (Cyclic Redundancy Check) o délce 15 bitů tvoří poslední pole vysílané zprávy. Proto se může generovat ze všech do té doby odvysílaných bitů zprávy dle vzorce

Jestliže libovolné zařízení detekuje chybu CRC, je generováno hlášení o chybě bitu.

Vkládání bitu (bit stuffing):

Jedná se o vložení jednoho bitu opačné úrovně, když se ve zprávě objevuje po sobě 5 bitů stejné úrovně. Tento způsob upravuje i správné časování přijímačů jednotlivých uzlů sítě. Objeví-li se chyba, je generována chyba při vkládání bitu.

Kontrola zprávy (message frame check):

Provádí se porovnáním zprávy s pevným formátem zprávy, uvedeným ve specifikaci. Pokud je na některém z bitů zjištěna nepovolená hodnota, je vygenerováno hlášení chyby rámce.

Potvrzení přijetí zprávy (acknowledge):

Každé zařízení, připojené ke sběrnici musí správně přijatou zprávu potvrdit. Činí tak změnou bitu v poli ACK (1 bit) z recessive - vysílané vysílačem na dominant. To platí i pro ta zařízení, která mají zapnuto filtrování a tedy zprávu nepřijímají.

11.2.4.5 Signalizace chyb

Každá stanice má zabudovány dva čítače. Jejich obsah je určen počtem chyb při vysílání a při příjmu zprávy. Pokud zařízení generuje příliš velký počet chyb, je automaticky odepnuto. V závislosti na stavu čítače chyb je zařízení identifikováno jako: aktivní (error active), pasivní (error passive) nebo je odpojené (bus-off).

Error active se aktivně podílejí na sběrnici. Pokud detekují chybu, hlásí její výskyt vysíláním aktivního příznaku chyby (active error flag), který je tvořen 6 bity dominant za sebou. Tím je porušeno pravidlo o vkládání bitu a dojde k porušení zprávy.

Error passive se také podílejí na sběrnici, ale v případě detekce chyby vysílají pouze pasivní příznak chyby, což je 6 bitů recessive po sobě. Tím zpráva není porušena.

Bus-off zařízení nemají žádný vliv na chování na sběrnici.

11.2.4.6 Základní typy zpráv

CAN rozlišuje celkem 4 druhy zpráv:

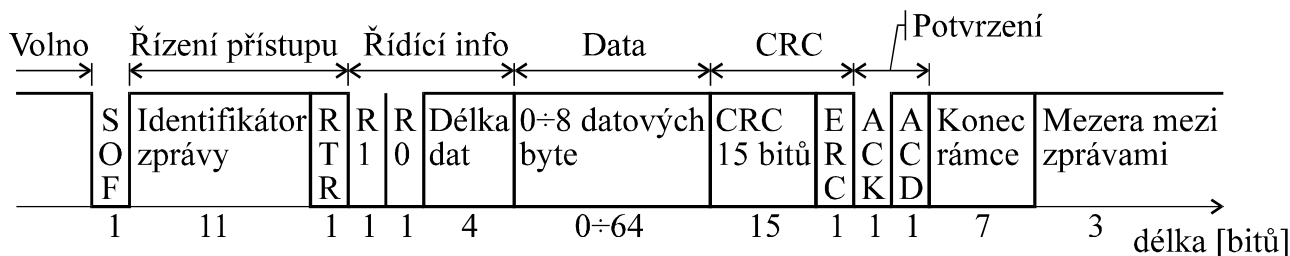
- datová zpráva
- žádost o data
- zpráva o chybě
- zpráva o přetížení

Datová zpráva a žádost o data se týkají přenosu dat. Datová zpráva tvoří základ komunikace, umožňuje zařízení vyslat zprávu dlouhou až 8Byte. Naopak při jednoduchých typech datových zpráv, jako povel vypni /zapni ap. není třeba posílat žádná data, tyto binární příkazy mohou být obsaženy v identifikátoru zpráv. Tím se zvyšuje rychlost přenosu v protokolu CAN. Zařízení může rovněž požadovat data vysláním zprávy žádost o data. Zařízení, které tato data vlastní je vyšle na sběrnici.

Další dva typy zpráv slouží k řízení sběrnice a to k signalizaci chyby a eliminaci chybných zpráv a k signalizaci o přetížení, tedy vyžádání prodlevy v komunikaci.

Datová zpráva (data frame)

Specifikace 2.0A (standard frame) definuje standardní, t.j. 11bitový identifikátor zprávy, zatímco rozšířený formát (Specifikace 2.0B) má 29bitový identifikátor. Na obr.8.36 je struktura datové zprávy.



Obr. 8.36: Struktura datové zprávy

Volná sběrnice je charakterizována recesivní úrovní. V tom okamžiku je možné vyslat zprávu, která bude mít následující formát :

- začátek zprávy (1b) SOF - Start of Frame
- řízení přístupu ke sběrnici a identifikátor zprávy (Arbitration Field), (11b), určuje prioritu zprávy a význam přenášené zprávy
- RTR (Remote Request) - 1b, slouží k rozlišení zprávy zda jde o datovou zprávu (dominant) nebo žádost o přístup ke sběrnici (recessive)
- řídicí pole (Control Field), R0 a R1 celkem dva bity, rezervováno
- délka datové zprávy (4b)
- datová oblast (Data Field) - max. 8Byte dat

- CRC - zabezpečovací kód (15b)
- ERC - 1b dominant, CRC oddělovač
- potvrzení ACK - (2b), z toho 1b (ACK), 1b oddělovač (ACK)- recessive
- konec zprávy (End of Frame) - 7b recessive
- mezera mezi zprávami (Interframe Space) - 3bity recessive

Formát 2.0A má v řídicím bitu R1 úroveň dominant, formát 2.0B úroveň recessive.

Formát 2.0B viz kupř. lit. [14], [27].

Na následujícím obr. 8.37 je na příkladu protokolu DeviceNet (pojednáno dále) ukázán význam pole Identifikátor zprávy u standardního formátu CAN 2.0A. Podrobnější rozbor této

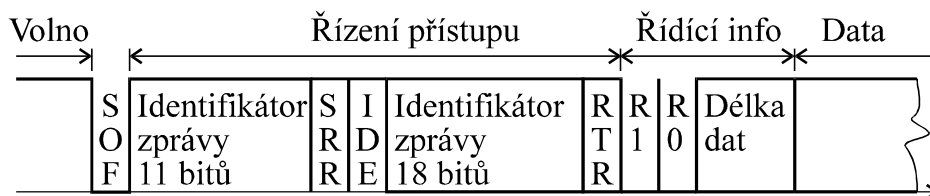
Identifikátor 11 bitů											Rozsah	Skup.
10	9	8	7	6	5	4	3	2	1	0		
0	Msg. ID				MAC ID zdroje						000-3FF	1
1	0	MAC ID					Msg. ID				400-5FF	2
1	1	Msg. ID			MAC ID zdroje					600-7BF	3	
1	1	1	1	1	Msg. ID					7C0-7EF	4	
1	1	1	1	1	1	1	×	×	×	×	7F0-7FF	NE!

Obr. 8.37: Význam identifikátoru zprávy

tabulky bude podán v kapitole, pojednávající o protokolu DeviceNet.

Z této tabulky je také vidět, že pomocí pole identifikátorů zprávy lze určit povahu zprávy (Msg. ID), adresu zdroje zprávy (MAC ID zdroje) a adresu příjemce zprávy (MAC ID).

Na dalším obr. 8.38 je rozšířený formát zprávy, používající identifikátor o celkové délce 29 bitů. Uspořádání obou formátů zaručuje, že při kolizi standardního a rozšířeného formátu má



Obr.8.38: Rozšířený formát zprávy

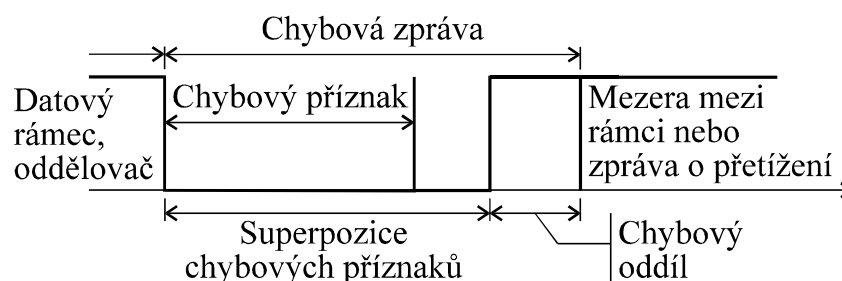
přednost standardní formát zprávy.

Žádost o data

Chybí zde přirozeně datové pole a RTR bit je recessive. Pokud zařízení požaduje nějakou zprávu, nastaví svůj identifikátor na identifikátor požadované zprávy. Vzhledem k tomu, že Žádost o data má nastavený RTR recessive, má nižší prioritu s případnou kolidující zprávou na sběrnici a nemůže ji ohrozit.

Zpráva o chybě

Tato zpráva slouží k signalizaci chyb na sběrnici. Jestliže libovolné zařízení, připojené ke sběrnici, zjistí v přenášené zprávě chybu (chybu bitu, CRC, chybu vkládání bitu nebo chybu rámce), vygeneruje zprávu o chybě. Podle stavu stanice (error active/ error passive), generující zprávu o chybě, jde buď o generování aktivního (šest bitů dominant) nebo pasivního (šest bitů recessive) příznaku chyby. Je zřejmé, že při generování aktivního příznaku je zpráva porušena (pravidlo o vkládání bitu) a zprávy o chybě tedy musejí začít vysílat i ostatní stanice. Hlášení chyb je pak indikováno superpozicí všech chybových příznaků vysílaných jednotlivými zařízeními. Délka tohoto úseku může být mezi 6 a 12 bity. Zpráva o chybě je na obr.8.39.



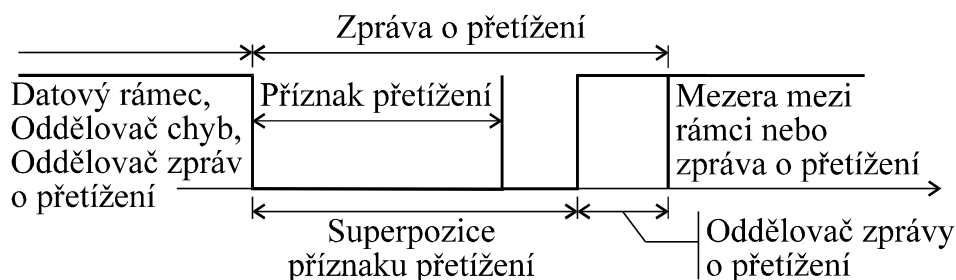
Obr. 8.39: Zpráva o chybě protokolu CAN

Po vyslání chybového příznaku, vysílá každé zařízení bity recessive. Současně detekuje stav na sběrnici a je-li recessive, vyšle dalších 7 bitů recessive, které fungují jako oddělovač - t.j. ukončení zprávy o chybě.

Zpráva o přetížení

Tuto zprávu vysílají zařízení, která kvůli vlastnímu přetížení nejsou schopna vysílat nebo přijímat další zprávy.

Struktura této zprávy je podobná zprávě o hlášení chyby, ale nesmí být vysílána okamžitě, ale až po konci zprávy (end of frame), nebo po oddělovači chyby nebo po oddělovači předcházející zprávy o přetížení, obr.8.40.



Obr. 8.40: Zpráva o přetížení

Zpráva o přetížení se skládá z příznaků přetížení (šest bitů dominant) a případné superpozice všech příznaků přetížení, pokud jsou generovány několika zařízeními současně. Za příznaky přetížení následuje dalších sedm bitů recessive tvořících oddělovač zprávy o přetížení.

11.2.4.7 Nabídka elektronických součástek s protokolem CAN [14]

- budiče sběrnice (tvoří fyzický interface z TTL na linky CAN_L a CAN_H)
 - CAN Philips 80C250
 - Philips 80C251,
 - Temic B10011S
 - Untirode UC5350
- samostatné řadiče
 - Philips 82C200
 - Philips SJA 1000
 - Intel 82527
 - Siemens 81C90/91
 - NEC APD72005
- osmibitové řadiče s integrovaným rozhraním CAN
 - Philips 80592
 - Philips 80598
 - Motorola MC68HC05X32
 - Motorola MC68HC08AZXX
 - Siemens C515C
 - Temic TSC8051A11
 - TSC80251A3
- 16/32 bitové řadiče s integrovaným rozhraním CAN
 - Motorola MC 6876
 - Intel 87C196CA/CB
 - Siemens C167CR
 - Philips XA-3

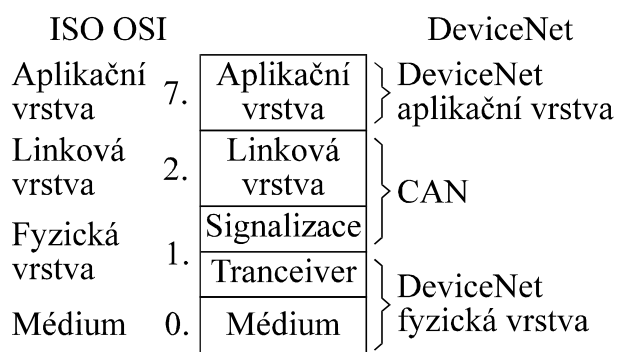
11.2.5 Protokol DeviceNet [28], [29]

Protokol DeviceNet definuje sériový přenosový kanál mezi prvky a systémy průmyslové automatizace. Je jednou z variant 7. vrstvy modelu OSI/ISO, která je definována nad protokolem CAN (1. a 2. vrstva modelu). Vyznačuje se těmito základními vlastnostmi [28], [29]:

- podpora až 64 účastníků sítě
- připojení/odpojení bez přerušení funkce sítě
- přenosová rychlost 125/250/500kBd
- společný rozvod datových signálů i napájení pro účastníky
- SW prostředky pro komunikaci typu žádost/odpověď
- prostředky pro efektivní přenos I/O dat
- fragmentace dlouhých bloků dat
- detekce duplicitních adres zařízení
- vzájemná zaměnitelnost zařízení od různých výrobců (specifikace obsahuje standardy pro typy konektorů, diagnostiku a profily různých typů zařízení)

Na obr.8.41 je zobrazen komunikační model protokolu DeviceNet.

Obrázek rovněž provádí srovnání specifikace CAN a modelu ISO/OSI. Jak je zřejmé již z popisu protokolu CAN, CAN definuje pouze fyzickou a linkovou vrstvu modelu ISO/OSI. Neobsahuje ani specifikaci fyzického média a interfejsu k médiu (MAU - medium attachment unit) ani specifikaci aplikační t.j. 7. vrstvy modelu. Specifikace DeviceNet nejen že (podobně jako SDS a CANopen) definuje aplikační vrstvu, ale obsahuje i specifikaci fyzického interfejsu.

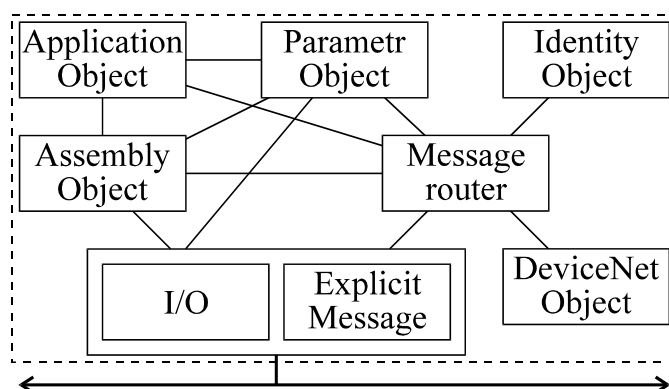


Obr. 8.41: Model DeviceNet

11.2.5.1 Objektově orientovaný model

Další důležitou vlastností DeviceNet je, že jde o objektový model přenosu. Zařízení, připojené k síti je popsáno jako množina objektů, z nichž každý je abstraktní reprezentací některé jeho části. Pojem třída definuje množinu objektů stejných nebo podobných vlastností. Každá třída definuje množinu atributů, množinu služeb a chování určitého typu objektu. Pojem instance se pojmenovává konkrétní realizace objektu určité třídy.

Adresace účastníků pak má čtyři složky - vlastní adresu zařízení (MAC ID), identifikátor třídy, číslo instance a číslo atributu nebo služby. Objektový model každého zařízení v síti DeviceNet je ukázán na obr. 8.42.



Obr. 8.42: Objektový model zařízení, komunikujícího v síti DeviceNet

Jak je z obr. 8.42 patrné, obsahuje typické zařízení tyto objekty:

- Identity object (obsahuje identifikaci výrobce, typ zařízení, číslo série ap.), každé zařízení vlastní jednu instanci tohoto objektu
- Message router object - zajišťuje předávání zpráv mezi objekty, vždy existuje jedna instance
- DeviceNet object obsahuje např.adresu zařízení (MAC ID), údaj o přenosové rychlosti apod., opět existuje jedna instance tohoto objektu
- Assembly object - jedna nebo několik instancí s úkolem spojovat různé atributy různých aplikačních objektů do jednoho atributu přenášeného po síti v rámci jednoho komunikačního objektu linky
- Connection object - nejméně dvě instance, reprezentuje vždy jeden koncový bod virtuálního komunikačního kanálu mezi dvěma účastníky sítě

- Parametr object - je použit v zařízeních s konfigurovatelnými parametry, pro každý parametr existuje jedna instance, která představuje standardní cestu pro jeho nastavování
- Application object - zajišťuje konkrétní aplikační činnost zařízení (pro různé typy zařízení existuje řada standardně definovaných objektů)

Povinné jsou objekty : Identity Object, DeviceNet Object, Connection Object a Message router object. Ostatní mohou, ale nemusí být u daného zařízení realizovány.

11.2.5.2 Aplikační vrstva protokolu DeviceNet

Aplikační vrstva definuje komunikační model (DeviceNet Messaging), profily zařízení a knihovnu aplikačních a pomocných objektů. Komunikační model zahrnuje způsob využití CAN zpráv, význam datového pole a rozdělení CAN identifikátorů, fragmentaci zpráv delších než 8 bytů, potvrzování přijetí, atd. Profily zařízení a knihovny objektů umožňují dosazení vzájemné zaměnitelnosti (interchangeability) a propojitelnosti (interoperability) produktů různých výrobců.

DeviceNet definuje pro předávání zpráv účinnější způsob, než je adresace typu zdroj/cíl, používaný u jiných způsobů sériové komunikace. Tento způsob se nazývá poskytovatel/příjemce. Identifikační pole neobsahuje adresu přijímací stanice, ale typ dat přenášených daným datovým rámcem. Tento způsob umožňuje využít větší počet úrovní priorit při povolování přístupu k přenosovému médiu, efektivní přenos I/O dat a komunikaci mezi jedním vysílacím objektem a několika objekty přijímajícími.

Přenášené zprávy jsou rozděleny do čtyř skupin, lišící se strukturou pole identifikátorů v protokolu CAN. Obsahují v různých kombinacích jednu nebo dvě z následujících komponent:

- identifikátor zprávy (Msg. ID) : identifikuje zprávy uvnitř dané skupiny zpráv a umožňuje tak navázat několik propojení pomocí zpráv dané třídy z jednoho zařízení. Po navázání spojení je s použitím odpovídajícího identifikátoru zprávy a adresy zařízení vytvořen identifikátor příslušný danému spojení, sloužící dále jako identifikátor rámce CAN
- adresu zdroje zprávy (MAC ID zdroje)
- adresu zdroje nebo příjemce zprávy (MAC ID)

Určování priority zpráv skupiny 1 a 3 je distribuováno mezi všechna zařízení v síti. Žádá-li o přístup ke sběrnici současně několik zpráv skupiny 1 nebo 3, rozhoduje nejprve identifikátor zprávy (Msg.ID). Při shodě rozhoduje jako druhé kritérium adresa vysílacího zařízení (MAC ID zdroje).

Protože zprávy skupiny 1 jsou jediné, které mají nejvýznamnější bit dominantní, mají nejvyšší prioritu. Zprávy skupiny 2 mají jako celek vyšší prioritu než zprávy skupin 3 a 4, ale uvnitř skupiny 2 rozhoduje o prioritě nejprve adresa zařízení a až jako druhé kritérium se uplatní obsah zprávy ve formě jejího identifikátoru (adresová položka - MAC ID zde může obsahovat adresu zařízení přijímajícího nebo vysílajícího). Část zpráv skupiny 4 má speciální význam, zbytek je vyhrazen pro pozdější použití a zatím není definován (viz obr. 8.37).

Podle toho, které dvě kombinace identifikátorů jsou použity, rozdělujeme zprávy do 4 skupin.

11.2.5.3 Detekce duplicity

Pro zaručení jedinečnosti své adresy na síti DeviceNet musí každé zařízení před započetím komunikace testovat duplicitu adresy. Za tím účelem vysílá zprávu Duplicate MAC ID Check Request. Tato zpráva je zachycena všemi zařízeními na sběrnici a zařízení porovnají svoji a vyslanou adresu. V případě shody vyšle příslušné zařízení zprávu Duplicate MAC ID Check Response. Pokud testovaná stanice nedostane tuto zprávu od žádného zařízení ani při opakování Duplicate MAC ID Check Request, je její adresa označena jako jedinečná a může pokračovat v činnosti související se sběrnici. V opačném případě nemůže komunikovat a jeho adresa musí být změněna.

11.2.5.4 Mechanismus komunikace v síti DeviceNet

DeviceNet definuje komunikaci pomocí komunikační linky tzv. Connection. Existují 2 typy Connection/Propojení, tzv. I/O Connection a Explicit Connection, přičemž pro každou komunikační linku je definován jiný typ zpráv (I/O zprávy, Explicitní zprávy).

Vstupní/výstupní zprávy

Vstupní/výstupní (I/O) zprávy jsou určeny pro přenos časově kritických aplikačně orientovaných dat a mají tudíž identifikátory s vysokou prioritou. V datové části rámce pak tyto zprávy nepoužívají žádný protokol s výjimkou fragmentačního protokolu, kterým se rozdělují větší bloky dat než 8 Byte, které je schopné přenést 2. vrstva CANu. Jeden Byte je pak využíván fragmentačním protokolem.

Jak již bylo řečeno (zde, ale i u protokolu CAN), význam zprávy je určen identifikátorem. Předtím, než je I/O zpráva odeslána, musí být komunikující zařízení nakonfigurována. Spočívá to v tom, že je definován identifikátor (Connection ID), zdrojový objekt, cílové objekty a jejich atributy. To tedy znamená, že zdroj a cíl (producent/konzument) jsou určeny při počáteční konfiguraci a při vlastním přenosu dat se již neuvádějí.

Explicitní zprávy

Přenos explicitních zpráv naopak probíhá zásadně mezi dvěma zařízeními. Proto se používá mechanismu typu žádost/odpověď. Používají se identifikátory s nízkými prioritami. Používají se hlavně pro konfigurování zařízení. Proto má jejich datová část přesnou strukturu obsahující specifikaci úlohy, která se má provést, adresu objektu a hodnotu jeho identifikátoru.

11.2.5.5 Navázání spojení v síti DeviceNet

Spojení se navazuje pomocí zpráv Open Explicit Message Connection Request a Close Connection Request. Ty jsou definovány ve 3. skupině ve formě tzv. explicitního propojení, kterým lze přenášet explicitní zprávy.

Na druhé straně I/O propojení jsou dynamicky navazována následujícím postupem :

- naváže se explicitní spojení s daným koncovým zařízením
- odesláním zprávy Create Request se v tomto zařízení vytvoří nová instance objektu I/O Connection
- objekt I/O Connection se zkonfiguruje

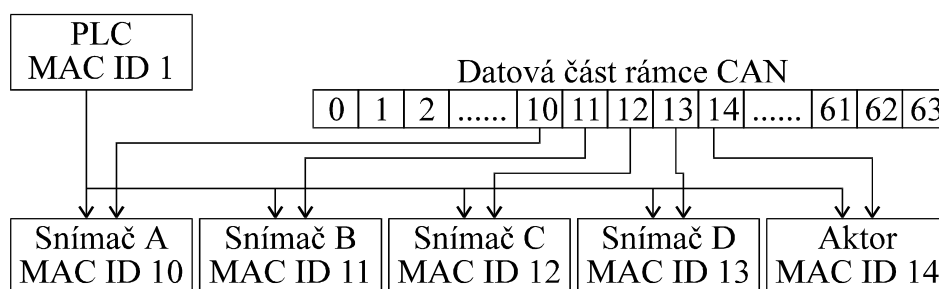
Propojení I/O může být typu point-to-point nebo multicast. V případě multicast několik účastníků poslouchá zprávu a potvrzuje její přijetí nebo odešle odpověď zpět vysílajícímu účastníku přenosu.

11.2.5.6 Podpora tvorby hierarchických komunikačních struktur

Kromě výše uvedeného manuálního způsobu navazování komunikace v síti DeviceNet, podporuje tento protokol i jednodušší způsob navazování spojení pomocí předdefinovaného souboru Master/Slave Connection set. Ten obsahuje následující definice typů přenosů :

- I/O Bit-Strobe
- I/O Poll
- Change of State and Cyclic Transmition

První z nich I/O Bit-Strobe je zprávou typu multicast. Několik podřízených stanic reaguje na vysílání mastera. Master odvysílá jeden bit dat každému zařízení, které má ve svém seznamu (scan-list). Povel obsahuje pole 8Byte. Každý bit z tohoto pole odpovídá jedné adrese MAC ID v síti (viz obr. 8.43).



Obr.8.43: Tvorba komunikační struktury

Celé pole 64 bitů je odesláno vždy, bez ohledu na to, kolik slave stanic má master na svém seznamu. Odezva na tuto výzvu ze strany slave může být až 8Byte I/O dat nebo informace o stavu dotazovaného zařízení.

Zpráva I/O Poll se uskutečňuje vždy mezi dvěma účastníky. Master musí odeslat povel každému slave samostatně. Mezi masterem a slave může být pak přenášen libovolně dlouhý objem dat, která jsou v případě potřeby fragmentována.

Zpráva Change of State and Cyclic transmition je zprávou, kterou může odeslat jak master, tak slave. Zpráva je určena vždy jen jednomu zařízení a může nebo nemusí být potvrzována. Lze opět přenášet libovolný objem dat (s fragmentací). Inicializace přenosu se uskutečňuje buď změnou stavu atributu určitého objektu (Change of state) nebo aktivací časovače (Cyclic).

11.2.5.7 Profily v síti DeviceNet

Každé zařízení má mít definovány třídy objektů. Některé z nich jsou povinné, některé nepovinné. Jak již bylo řečeno, k povinným objektům patří definování tříd objektů Connection

Object, DeviceNet object, Identity Object a Message Router Object. To je pro komunikaci v síti DeviceNet nezbytné. Ostatní objekty se týkají vlatní funkce daného zařízení.

Pro zajištění vzájemné interoperability zařízení různých výrobců, ale stejného typu, musí být dána doporučení pro výše uvedené nepovinné objekty. Takový standard musí popisovat chování daného typu zařízení a zajistit tak stejné vnější chování zařízení daného typu. Takovému standardu se říká profil (Device Profile).

Každý profil musí definovat objektový model daného zařízení, formát I/O dat, konfigurační parametry a rozhraní pro jejich zpřístupnění uživateli. V současné době existují profily pro analogové I/O, DI/DO, snímače, komunikační rozhraní, řídicí jednotky motorů a další.

11.2.5.8 Fyzická vrstva v síti DeviceNet

Fyzická vrstva se skládá z tranciveru, oddělovacích obvodů a ochranných obvodů. Optické oddělení není povinné pro zařízení napájené ze sběrnice. Přenosové médium se využívá i pro napájení stanic napětím 24V =. Proudové omezení celé sběrnice je omezeno na 8A. Proud jednotlivými přípojnými vedeními na 3A.

Protokol DeviceNet není pouze proprietárním komunikačním prostředkem systémů Allen-Bradley, ale pro své výhodné vlastnosti (staví na oblíbené, otevřené a levné technologii CAN a přitom definuje aplikační vrstvu) proniká do výrobků celé řady výrobců průmyslové automatizace. Výhodou je plná otevřenost protokolu.

12. TOTÁLNĚ DISTRIBUOVANÉ AUTOMATIZAČNÍ SYSTÉMY

V této krátké kapitole pojednáme o jednom významném moderním proudu prostředků průmyslové automatizace. Jedná se o systémy, které jsou charakterizovány velkou samostatností a kompetencí, takže nevyžadují žádný nadřazený řídicí člen. Páteř tohoto systému musí tvořit výkonný komunikační systém (fieldbus nebo průmyslový Ethernet), každý automatizační člen (aktor, sensor, automat, CNC jednotka, atd.) musí být vybavena mikrořadičem, který je jednak schopen zajistit komunikaci s libovolným připojeným účastníkem (node), jednak vykonávat funkci interface k automatům, CNC, průmyslovým regulátorům a jiným řídicím členům, jednak plnit funkci zatím poměrně jednoduchého řízení a sběru dat. Komunikační protokol musí umožnit samostatnou komunikaci každého s každým (peer to peer) a zaručení definovaného okamžiku přístupu k síti v případech, kdy to řízení procesu vyžaduje. Tyto systémy jsou velmi vhodné v automatizaci budov, kde je třeba, aby kupř. aktor uzavření ventilu teplovodního vytápění reagoval na signál z několika čidel, signalizujících otevření některého z oken (nebo všech kombinací) v dané místnosti. Dále nacházejí tyto systémy uplatnění i ve spínacích stanicích vysokého napětí a v dalších průmyslových aplikacích. Tento trend bude sílit s vývojem výkonnějších mikrokontrolerů, kterými jsou node vybavovány.

Jako příklad tohoto trendu a jako jeden z příkladů totálně distribuovaného systému druhé poloviny 90. let probereme stručně technologii LonWorks.

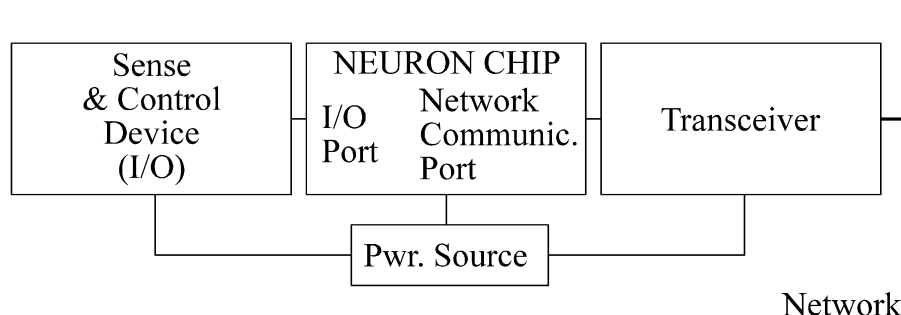
12.1 Technologie LonWorks

Technologie LonWorks, která staví na výše uvedených vlastnostech totálně distribuovaných systémů, vyvinula v začátku 90. let americká firma Echelon. Základ technologie - mikrořadič Neuron - začaly vyrábět dvě přední firmy Motorola a Toshiba. V současné době vyrábí Neuron jen firma Toshiba. Vlastnosti technologie LonWorks a komunikačního protokolu LonTalk (který lze řadit do kategorie fieldbusů ve smyslu kap. 8), vycházejí z požadavků, které si tvůrci tohoto systému zadali:

- protokol musí být implementovaný do velmi levného čipu
- musí podporovat přenos nejběžnějšími médii od kroucené dvoulinky, radiového přenosu, telefonní linky, silových rozvodů, infračerveného přenosu až po koaxiální kabel a světlovodič
- musí umožnit připojení až desítek tisíc účastníků sítě
- musí zaručovat velmi bezpečný provoz, neboť jeho základní poslání je v řízení procesu a ne pouze v přenosu dat
- doba odezvy musí být nezávislá na rozlehlosti sítě
- musí umožnit peer-to-peer komunikaci, t.j. každý účastník musí mít možnost bez arbitra komunikovat s jiným libovolným účastníkem (což má úzkou souvislost s bezpečným provozem sítě, která tak není vystavena zhroutilí při zhroutilí arbitra)
- musí umožnit zabudovat arbitráž jednoho účastníka nad druhým. Řešení musí být softwarové a zcela nezávislé na výrobci čidel a akčních členů (musí to být možné provést v protokolu)
- na LonTalk musí být připojitelné výrobky různých výrobců bez jakýchkoli dohod a konzultací prostě jen tím, že jsou pomocí čipu připojitelné do sítě LonWorks
- interface musí být pro komunikující prvek zcela průhledný

12.2 Základní vlastnosti protokolu LonTalk

Na obr. 9.1 je blokové schéma typického uzlu (node). Jádro tvoří mikrořadič NEURON MC143120 nebo MC143150 (nebo jejich varianty, vyráběné Toshiba). Dále každý node obsahuje stykový obvod (transceiver), který odpovídá zvolenému přenosovému médii. V případě



Obr.9.1: Blokové schéma LonWorks node

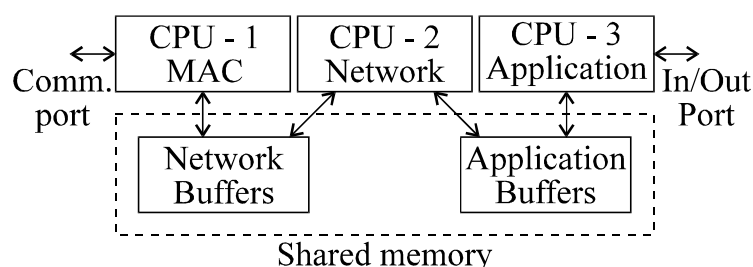
čipu MC143150 node obsahuje ještě vnější paměť.

Protokol LonTalk je implementován do obvodu NEURON. Obr. 9.2. Čip obsahuje tři 8 bitové procesory (CPU1, CPU2, CPU3), které jsou využity následovně:

- MAC (Medium Access Control) vrstva 1-2
- Network Processor vrstva 3-6
- Application Processor vrstva 7

NEURON dále obsahuje minimálně 512 byte EEPROM pro konfiguraci "node", paměť RAM pro procesní proměnné a program, řídicí blok a I/O rozhraní, kde jde každou z 11 linek programově nastavit jako vstup nebo výstup.

V provedení Neuron MC143120 je v čipu 10kB paměť ROM s uloženým protokolem LonTalk a uživatelská paměť pro menší aplikace. Externí paměť nelze připojit. Model MC143150 má vyvedenou sběrnici pro připojení externí paměti až 42 kB, kam však je též třeba uložit kromě uživatelského programu i protokol LonTalk.

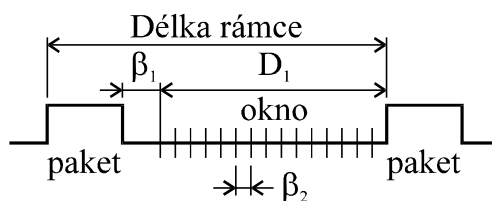


Obr. 9.2: Principiální schéma NEURONu

Node (uzel, zařízení ap.) se programuje buď v assembleru nebo v jazyku Neuron C, což je upravený ANSI C jazyk pro práci v reálném čase.

V poslední době se objevily i nové čipy NEURON s implementovanými nižšími vrstvami protokolu a 32 bitovým výkonným procesorem pro uživatelský program a vyšší vrstvy protokolu, což umožní připojovat k síti LonWorks i výkonná zařízení.

LonTalk protokol je peer-to-peer predikativní *p*-naléhající CSMA protokol, vyvinutý firmou Echelon. Tvůrci tvrdí, že tak dosahují v nejhorším případě stejné doby odezvy jako u token passing způsobu, a realizace je výrazně levnější a jednodušší (žádné starosti se ztrátou token nebo jeho zdvojením ap.).

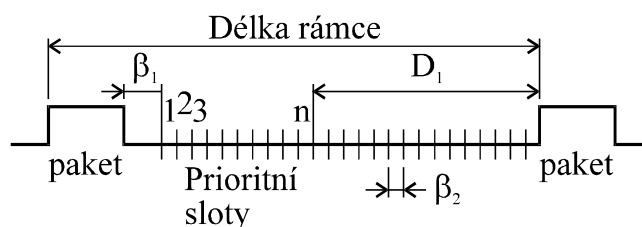


Obr.9.3: Rámec protokolu LonTalk bez využití priorit

Zařízení, které hodlá vysílat, čeká před vysláním rámce náhodnou dobu D_1 , počítanou v násobcích doby β_2 . Maximum čekací doby se mění dle zatížení sítě. Doba β_1 představuje čas, potřebný k rozpoznání konce rámce, obr.9.3.

Není-li médium obsazeno, pokusí se uzel zprávu vyslat s pravděpodobností p . Během vysílání se testuje kolize, podporuje-li použité médium detekci kolizí. Dojde-li ke kolizi, upraví se parametry vysílání a vysílání se opakuje.

Pokud je povolena priorita, vkládají se před vysílaný rámec prioritní časové sloty (obr.



Obr.9.4: Rámec protokolu LonTalk s využitím priorit

9.4.) Čím vyšší je priorita, tím menší je počet prioritních slotů

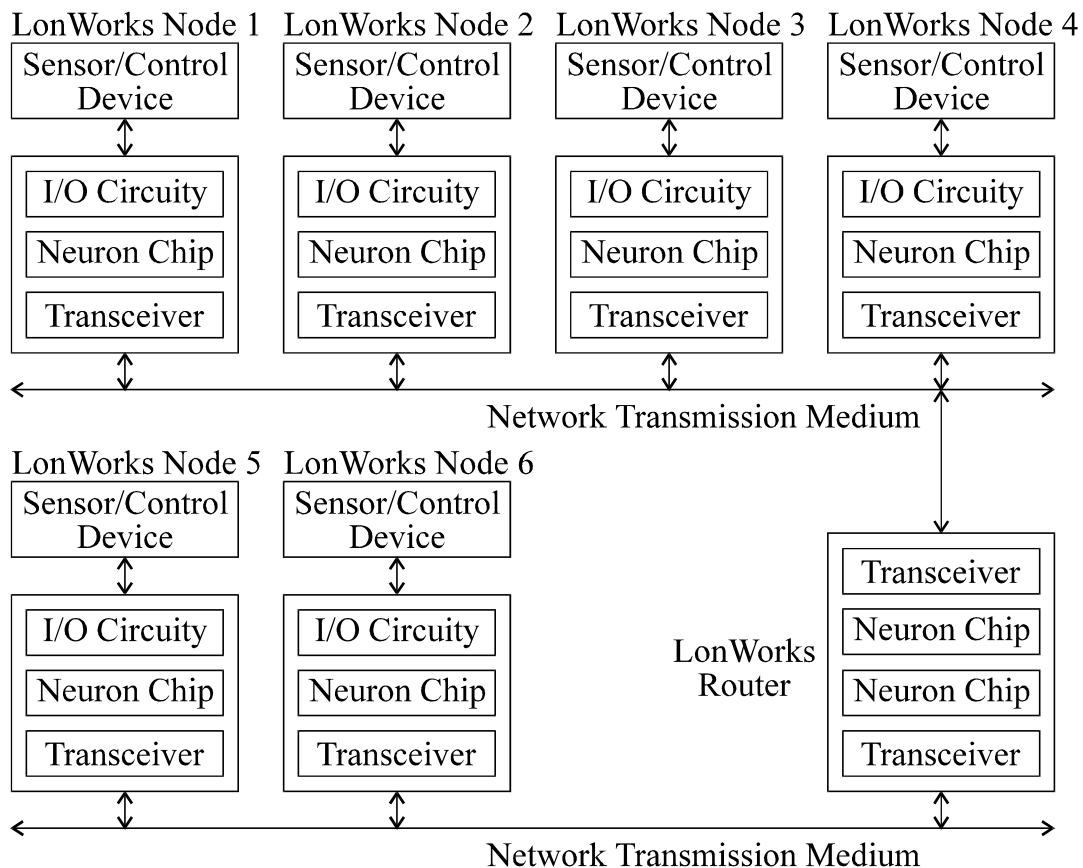
Rychlost přenosu dat závisí na použitém rozhraní. U rozhraní a síť RS-485 je to 4.8kbitů/s až 1.25Mbitů/s a v současné době i 2,5 Mbitů/s. Při použití galvanického oddělení přes tranceivery je rychlost 78 kbitů/s až 1.25 Mbitů/sec. Maximální vzdálenost přenosu u RS-485 je 1.2km. Pomocí opakovačů se dá zvýšit. Počet účastníků je prakticky neomezen (32 000 node).

Signál je kódován kódem ManchesterII. Každý přenášený rámec je zabezpečen 16 bitovým CRC cyklickým redundantním kódem. Rámce mohou mít velikost max. 255 byte, bývají však podstatně kratší 10 až 16 byte. Vzhledem k možné rozsáhlosti sítě, poskytuje protokol LonTalk bohaté možnosti adresování. Na nejnižší úrovni (jednotlivá zařízení) se používá Neuron ID, což je 48 bitové číslo, jednoznačně přiřazené danému "node". Dále se síť může dělit na domény. Domény se dělí na podsítě a fyzicky na kanály. Node může být členem max. 2 domén. Doména má délku 0, 1, 3 nebo 6 byte. Kanál je část domény s jedním fyzickým médiem. Kanály se navzájem spojují pomocí opakovačů (repeater) nebo mostů (bridge). Spojené se chovají jako podsítě. Jednotlivé podsítě se propojují směrovači (router). Router se dá realizovat pomocí "node" se dvěma Neurony. Na obr. 9.5 je znázorněna jednoduchá síť se dvěma podsítěmi, spojená routerem.

Domény se mezi sebou propojují pomocí bran (gateway).

Transportní vrstva protokolu zajišťuje nepotvrzované a potvrzované spojení. Zařízení (node) může komunikovat s jedním zařízením (unicast) nebo s několika (multicast) nebo se všemi (broadcast). Jsou poskytovány tyto typy služeb:

- ACKD - Potvrzovaná zpráva, je posílána jednomu nebo skupině uzlů a je očekáváno potvrzení individuálně od každého příjemce zvlášť. Jestliže potvrzení nepříjde do určitého času, je zpráva opakovaně vyslána. Počet znovu vysílání a časy jsou konfigurovatelné. Potvrzení jsou generována automaticky síťovým CPU bez zásahu aplikace.
- REQUEST - Žádost/odpověď, zpráva je poslána uzlu nebo skupině a je očekáváno potvrzení individuálně od každého příjemce zvlášť. Příchozí zpráva je zpracována aplikací a na jejím základě je poslána odpověď, která již může obsahovat data. Znovuvysílání funguje jako u potvrzovaných zpráv (ACKD).
- UNACKD-RPT – Nepotvrzovaná zpráva s opakováním posílána jednomu nebo skupině uzlů.
- UNACKD - Nepotvrzovaná služba, používána pro přenos rozsáhlých zpráv (nekritických).



Obr. 9.5: Jednoduchá síť LonWorks

Aplikační program může deklarovat speciální třídu statického objektu, zvanou síťová proměnná. Síťová proměnná může být typu input nebo output. Přiřazení hodnoty výstupní proměnné způsobí přenesení hodnoty do všech uzlů deklarujících tuto proměnnou jako vstupní.

Druhou možností aplikačního programu jsou explicitní zprávy. Jsou využívány tam, kde není vhodné použití síťových proměnných, t.j. kupř. při použití datových typů nestandardních v jazyku Neuron C. Každá explicitní zpráva nese kód, mající uživatelem definovaný význam a dále může obsahovat data o délce max. 228 byte. Síťové proměnné i explicitní zprávy mohou být vysílány všemi druhy zpráv.

Jak již bylo řečeno, technologie LonWorks podporuje širokou škálu přenosových médií. To jí zaručuje možnost průniku do mnoha aplikací s využitím stávajících instalací komunikačních i jiných přenosových cest včetně bezdrátového přenosu. S posilováním samostatnosti a výkonu mikrokontroleru NEURON, bude vhodná i pro novou důsledně distribuovanou architekturu řídicích systémů procesů.

13. PRŮMYSLOVÉ SÍTĚ NA NEJNIŽŠÍ ÚROVNI SBĚRU DAT

Typickými představiteli sběrnic typu sensor/actor bus je poměrně nová evropská sběrnice AS-interface. Jak již sám název napovídá, nejedná se o průmyslovou sběrnici, která má ambice konkurovat hrahým a komplikovaným evropským standardům. Přesto se prosadila na nejnižší úrovni řízení a její budoucnost je zajištěna i v případě masivního nástupu internetových komunikačních technologií v oblasti řízení a sledování výroby. Proto je jí věnována v celém kursu i v laboratorním cvičení zvýšená pozornost. Další typickou sběrnici typu sensor/actuator bus je protokol HART, používaný především v měřicích systémech. Poslední ze sběrnic této kategorie je jedna z nejstarších průmyslových sběrnic, která je velmi rozšířená v německém průmyslu. Jedná se o sběrnici Interbus S, která tvoří spojovací článek mezi kategorií sensor/actuator bus a devicebus.

13.1 Síť Actuator Sensor Interface (AS-Interface, AS-I)

Sběrnice AS-Interface byla počátkem 90-tých let navržena konsorciem 11 výrobců automatizační techniky a při jejím vzniku byly položeny následující požadavky, které nová sběrnice musí splňovat:

1. K jednotnému komunikačnímu rozhraní musí být připojitelné snímače a akční členy různých výrobců;
2. Komunikačním médiem bude nenákladný dvou vodičový kabel, pokud možno nestíněný;
3. Komunikační síť musí umožnit libovolnou topologii sítě (sběrnice, strom, kruh, ...);
4. Data i napájení (ss. 24 V) by se mělo šířit jedním párem vodičů;
5. Vysoká spolehlivost v průmyslovém prostředí
6. Doba odezvy přibližně 5 ms;
7. Počet adresovatelných bitů v každém uzlu: 3 až 4;
8. Jednoduchá realizace elektroniky realizující připojení k síti;
9. Nízké náklady na uzel sítě;
10. Snadná instalace zařízení, snadná lokalizace poruch a snadné odstraňování poruch.

Sběrnice AS-interface tak, jak byla navržena všechny výše uvedené požadavky splňuje. Sběrnice AS-interface je sítí typu master-slave, k síti je připojen jedno zařízení typu master a až 62 zařízení typu slave. V původním návrhu sběrnice AS-interface byla možnost připojit 32 zařízení typu slave, ale v minulém roce byla vytvořena nová specifikace AS-interface verze 2.1, která množství zařízení rozšířila. V současné době jsou na trhu jak zařízení podporující pouze původní variantu, tak zařízení podporující variantu AS-i verze 2.1.

13.1.1 Hlavní rysy původní specifikace AS-i:

- Přenos dat: master-slave s jedním masterem, který cyklicky obvolává zařízení typu slave.
- Adresace: Slave má adresu v rozsahu 0 - 31. Adresu přiřazuje master nebo projektant sítě ručním programátorem. Nový slave má z výroby nastavenou adresu 0. Adresa je uložena v EEPROM, nebo OTP ROM. Ve většině zařízení typu slave je použita EEPROM, kterou lze mnohonásobně přepisovat avšak některá levná zařízení typu slave obsahují OTP ROM a možnost měnit adresu je omezená na několik pokusů!
- Topologie: sběrnice, strom, hvězda, kruh
- Médium: nekroucený nestíněný dvou vodičový profilovaný kabel, který přenáší jak data, tak napájení 24 V ss. Proudový odběr celé sítě je max. 8 A, jedno zařízení typu slave odebírá max. 200 mA.
- Počet slave: 31 (32 v případě, že se využije i adresa 0)
- Počet I/O: Slave má až 4 binární vstupy a až 4 binární výstupy. Celá síť může mít až 124 (128) binárních vstupů a 124 (128) binárních výstupů.
- Adresace: Zpráva vyslaná masterem - výzva - obsahuje adresu jednoho zařízení typu slave. Slave na tuto výzvu okamžitě odpovídá.
- Komunikace: Cyklické zprávy master->slave i slave->master obsahují 4 datové bity. Acyklické parametrizační zprávy obsahují 4 parametrizační bity.
- Rychlost: Přenosová rychlost je 167 kb/s.
- Typy zařízení: Master, slave. Slave je charakterizován svým profilem (ID code) a konfigurací vstupů a výstupů (I/O configuration).
- Profily: Existuje 15 standardních profilů zařízení
- Cyklus sítě: Cyklus sítě závisí na počtu zařízení typu slave. V plné konfiguraci (32 zařízení typu slave) je cyklus sítě <5 ms.
- Délka sítě: Celková délka sítě, včetně odboček, nesmí překročit 100 m, resp. 300 m s opakovači.
- Detekce chyb: Způsob modulace a formát zprávy zajišťuje spolehlivou detekci chyb. V případě poškození zprávy je zajištěno její opakování.
- Master: Zajišťuje inicializaci sítě, identifikaci připojených zařízení, acyklické nastavení parametrů pro všechny slave, detekci chyb při přenosu dat, cyklické obvolávání všech slave, hledání nových připojených zařízení, přidělování adresy zařízením, které byly z důvodu poruchy nahrazeny novým kusem s adresou 0.

13.1.2 Nové vlastnosti zavedené specifikací AS-Interface verze 2.1

- Počet slave: 62, adresy 0-31 sdílí vždy dvě zařízení, tzv. zařízení A a zařízení B. Např. na adrese 24 se nalézají zařízení: 24A a 24B. Celkem lze připojit 248 vstupů a 186 výstupů.
- Přenášená data: Pro rozlišení zařízení A a B se využívá mj. i jeden bit, který ve starší verzi sloužil pro přenos dat. Množství vstupů a výstupů se tak nedosahuje dvojnásobku původní verze. Zařízení jsou zpětně kompatibilní.

Cyklus sítě: Cyklus sítě závisí na počtu zařízení typu slave. V plné konfiguraci (62 zařízení typu slave) je cyklus sítě <10 ms.

Profily: Existuje 225 standardních profilů zařízení

Inovovaná specifikace AS-Interface kromě zvýšení počtu slave přidává i vylepšenou podporu pro analogové uzly typu slave (A/D a D/A převodníky). Podobně jako u starší varianty sítě AS-interface je pro komunikaci s A/D a D/A převodníky, které pracují s více než 4-mi bity nutné využít více cyklů sítě. Např. 12-ti bitový A/D převodník potřebuje k předání naměřených dat 6 cyklů sítě. Vylepšená podpora analogových zařízení v podobě dvou analogových profilů výrazně usnadňuje začlenění analogových I/O do projektu.

Zařízení podporující AS-I verze 2.1 lze připojit na síť se masterem verze nižší, než 2.1 avšak nové schopnosti zařízení nebudou využity. V případě připojení starých slave k masteru verze 2.1 musí toto "staré zařízení" mít svoji nesdílenou adresu.

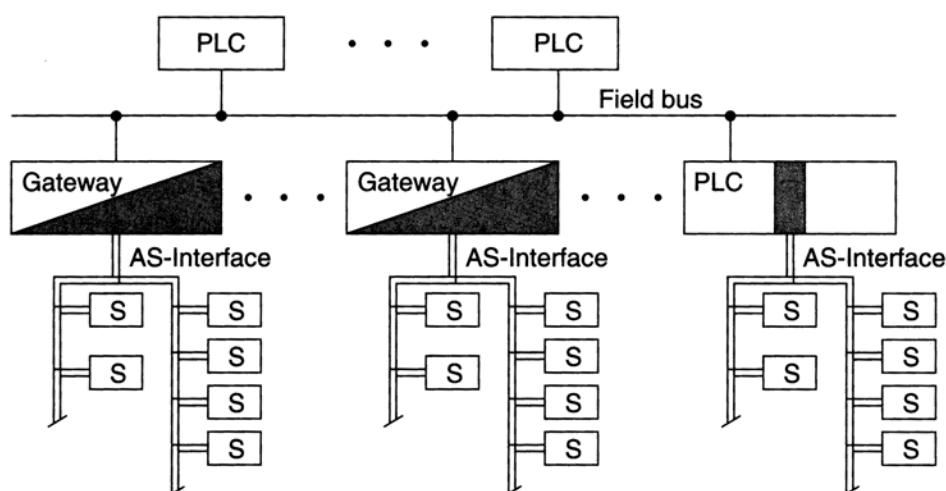
13.1.3 Slave síť AS-Interface

Slave síť AS-Interface je obvykle realizován speciálním AS-I chipem, avšak lze jej softwarově realizovat téměř libovolným mikrokontrolerem. AS-I slave je charakterizován jednak svým profilem, jednak konfigurací vstupů a výstupů. Vlastnosti některých zařízení typu Slave lze ovlivňovat zápisem 4 parametrizačních bitů. AS-I slave smí ze sítě AS-I odebírat max. 200 mA (ve verzi 2.1 je to pouze 100 mA). Vyžaduje-li větší napájecí proudy, je možné použít přídatný napájecí rozvod 24 V ss (černý profilovaný kabel), nebo rozvod 110V/230V stř. (červený kabel), případně může mít slave své vlastní napájení z externího zdroje. Velké množství zařízení typu slave je dostupné v provedení s elektrickým krytím IP 67.

13.1.4 Master síť AS-Interface

Master síť AS-I je obvykle realizován jako:

- a) Master v podobě karty do PLC. Konfigurace sítě AS-I může být zcela automatická, neboť karta v PLC pouze otestuje všechny adresy sítě AS-I, zjistí profily a I/O konfiguraci připojených zařízení a zapíše si ji do paměti. Je možná i manuální konfigurace, kdy je aktuální konfigurace sítě nahrána do paměti pomocí konfiguračního softwaru. Sběrnice AS-I připojená prostřednictvím master-karty do PLC se z hlediska programovatelného automatu jeví jako běžné vstupy a výstupy tohoto automatu. V případě, že je to nutné, po spuštění sítě Master zajistí správné nastavení parametrizačních bitů slave.
- b) Master typu gateway. Master síť AS-I je současně zařízením typu slave jiné, nadřazené sítě (např. Profibus). Tato varianta se používá v případě, kdy v řízeném procesu již existuje nadřazená síť, viz. Obr. 1, a sběrnice AS-Interface reprezentuje nenákladné distribuované vstupy a výstupy nadřazené sítě. Toto řešení je vhodné všude tam, kde množství vstupů a výstupů nebo další požadavky vyžadují použití vyšší komunikační sítě.



Obr. 1 Začlenění sítě AS-Interface do nadřazené sítě

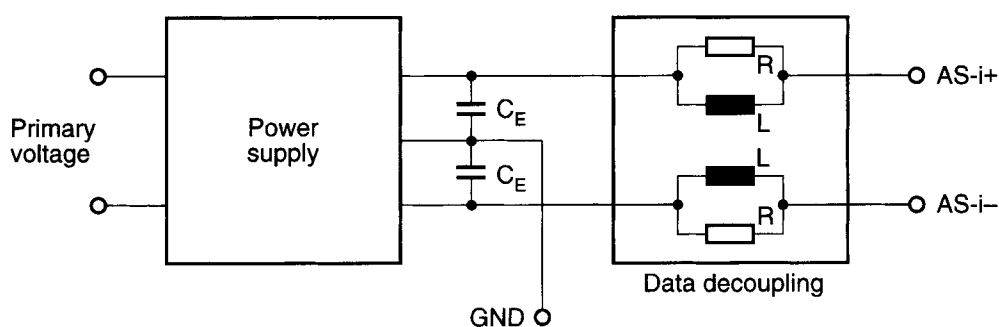
- c) Samostatný master. Samostatný master je v podstatě jednoduchý programovatelný automat, jehož veškeré vstupy a výstupy jsou distribuované prostřednictvím sběrnice AS-Interface. Tato varianta je vhodná pro jednoduché automatizační úlohy, kde nejsou velké požadavky na komplexnost a výpočetní náročnost řídicího algoritmu.

13.1.5 Napájecí zdroj sítě AS-Interface

Napájecí zdroj sběrnice AS-Interface slouží k napájení připojených slave a částečně nebo úplně pro napájení mastera prostřednictvím dvou vodičů žlutého profilovaného AS-I kabelu. AS-I zdroj fakticky zajišťuje generování napěťových pulsů, které slouží ke komunikaci. Principiální konstrukci AS-I zdroje ilustruje obr. 2. Na umístění napájecího zdroje v síti AS-I nejsou kladeny žádné požadavky a proto může být napájecí zdroj zapojen do sítě AS-I na libovolném místě. Vzhledem k tomu, že síť AS-I a zařízení na ni připojená pracují bez uzemnění, napájecí zdroj musí vyhovovat požadavkům na PELV zařízení, neboť zajišťuje bezpečné oddělení sběrnice AS-I od silových napájecích rozvodů.

Specifikace AS-I nepředepisuje maximální proud z napájecího AS-I zdroje, avšak existují následující omezení:

1. úbytek napětí na sběrnici by v případě dlouhého vedení neměl překročit 3 V
2. nesmí dojít k magnetickému nasycení indukčností, které jsou součástí zdroje AS-I.

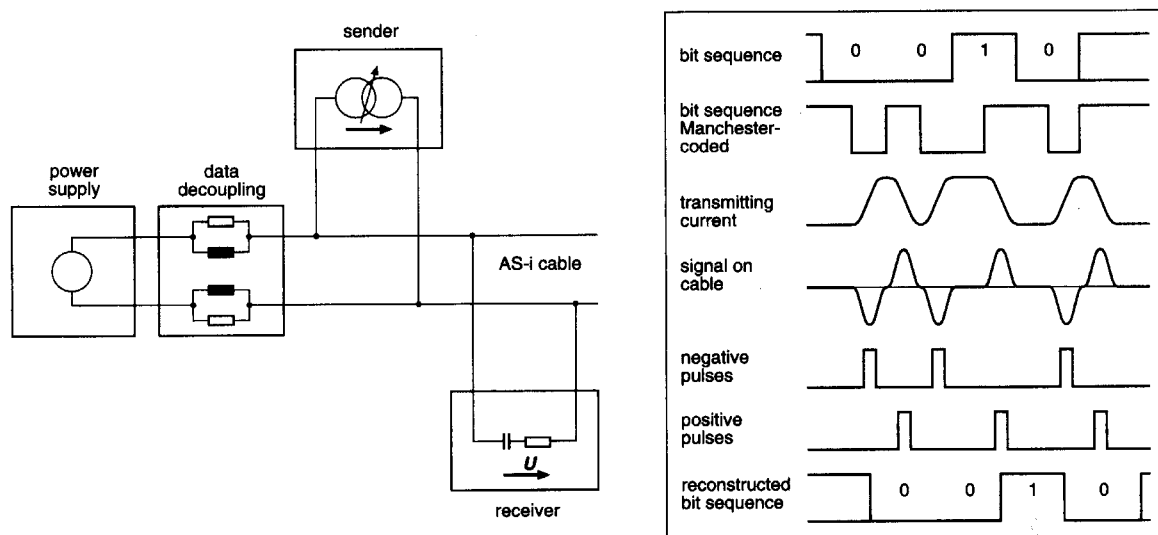


Obr. 2 Konstrukce zdroje AS-Interface

V současné době jsou na trhu zdroje, které umožňují odebírat ze sítě AS-I proudy v rozmezí 2 A až 8 A. Napájecí zdroj "naprázdno" dává napětí 29,5 V až 31,6 V, což zajišťuje že i v nejhorším možném případě bude na konci sběrnice dostatečné napětí. Zdroj AS-I je odolný proti trvalému zkratu a dlouhodobému přetížení a musí být vybaven obvody zajišťujícími omezení zkratového proudu.

Způsob modulace

Síť AS-Interface využívá unikátní způsob modulace nazvaný Alternating Pulse Modulation. Tento způsob modulace umožňuje přenášet data po napájecích vodičích, zajišťuje vysokou pravděpodobnost bezchybného přenesení dat se spolehlivou detekcí chyb při přenosu a rovněž zajišťuje elektromagnetickou kompatibilitu, neboť výsledný signál neobsahuje vysoké frekvence. Způsob modulace a význam speciálního AS-I zdroje dokumentuje Obr. 3.



Obr. 3 Význam zdroje AS-Interface a způsob modulace signálu

13.1.6 Kabeláž sítě AS-Interface

Charakteristickým kabelem pro sběrnici AS-I je profilovaný žlutý dvou vodičový kabel, který neumožňuje chybné založení do vodičích drážek lůžka konektoru. Zařízení se ke kabelu připojují prostřednictvím speciálních lůžek vybavených dvojicemi hrotů, které proříznou izolaci a vytvoří

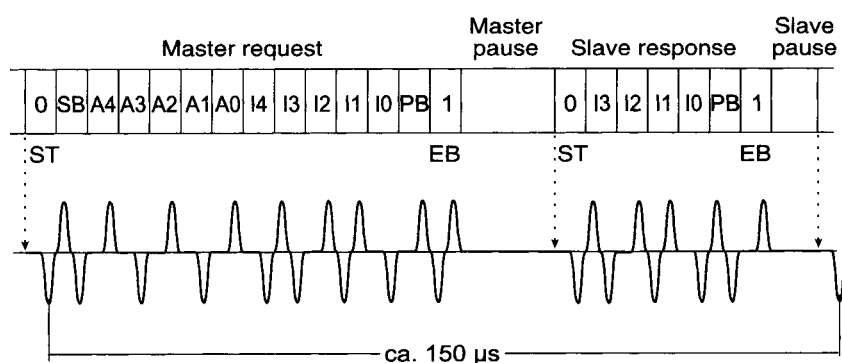
spolehlivý elektrický kontakt s vodiči v kabelu. Pružnost izolačního materiálu použitého na kabel zajišťuje, že v případě odpojení zařízení od kabelu dojde k roztažení izolace tak, že otvory způsobené připojením zařízení jsou spolehlivě zakryty. V případě, že připojená zařízení vyžadují samostatné napájení nebo větší odběr proudu, než je povolený odběr za žlutého kabelu, je k dispozici pomocný rozvod stejnosměrného napájení do 30 V profilovaným černým kabelem. Tento kabel je obvykle napájen ze samostatného zdroje 24 V. Na tento pomocný zdroj 24 V, na rozdíl od zdroje AS-I, nejsou kladeny žádné zvláštní požadavky. V případě že je požadováno dodatečné napájení střídavým napětím 110 V nebo 230 V, používá se profilovaný červený kabel.

V některých aplikacích se také používá standardizovaný kabel kulatého průřezu. V případě použití tohoto kabelu je nutné použít jiné standardizované provedení pouzder a konektorů.

13.1.7 Spolehlivost sítě AS-Interface

Způsob modulace a malá délka zpráv zajišťují velmi dobrou detekci chyb při komunikaci, aniž by bylo nutné používat složité kontrolní součty. Způsob modulace a relativně nízká komunikační rychlost zajišťují spolehlivou komunikaci i v silně zarušeném prostředí. V případě nutnosti opakovat zprávu se doba cyklu sítě prodlužuje o cca. 0.15 ms. Při komunikaci se rozlišují následující chyby:

1. nesprávná polarita start bitu;
2. nesprávná polarita pulsu (musí se střídát);
3. prodleva mezi pulsy je příliš dlouhá;
4. špatný formát zprávy (chybí puls ve druhé polovině bit-intervalu);
5. chyba parity;
6. chybný ukončovací bit (špatná polarita pulsu);
7. chybná délka zprávy.



ST = start bit
 SB = control bit
 A4...A0 = address of slave
 I4...I0 = information part
 PB = parity bit
 EB = end bit

Obr. 4 Příklad Master - Slave komunikace

Simulace i testy v praxi ukazují, že zabezpečení sběrnice AS-Interface je zcela dostatečné a plně srovnatelné s ostatními průmyslovými sběrnicemi, neboť použité zabezpečení je srovnatelné s bezpečnostními kódy s Hammingovou vzdáleností 5. Příklad komunikace mezi Masterem a Slavem ilustruje Obr. 4.

13.1.8 Kdy AS-I není vhodná síť

AS-I nemusí být vhodnou sítí, pokud:

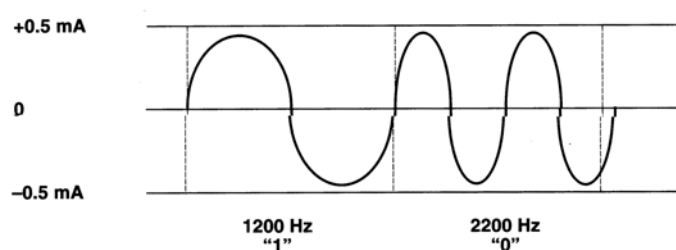
3. aplikace vyžaduje méně, než 15 binárních I/O (cenové důvody);
4. aplikace pracuje v prostředí s nebezpečím výbuchu;
5. aplikace je příliš rozlehlá (max. délka sítě AS-I je 100 m, resp. 300 m s opakovacími);
6. aplikace vyžaduje odezvu v čase <1 ms, přičemž vyžaduje použití mnoha vstupů a výstupů;
7. aplikace vyžaduje odezvu analogových čidel <50 ms.

13.2 HART (Highway Addressable Remote Transducer)

Protokol HART je protokolem master-slave, který je vhodný pro připojení inteligentních (smart) analogových čidel. Přenos dat z čidla do nadřazeného systému probíhá v digitální podobě, což kromě zjištění aktuální hodnoty měřené veličiny mj. umožňuje čidla parametrovat, kalibrovat a zjišťovat diagnostické informace. HART vznikl jako rozšíření tradiční proudové smyčky 4 - 20 mA a zůstává s touto technologií sběru dat kompatibilní.

13.2.1 Jak HART pracuje

HART využívá pro přenos dat využívá standardní proudovou smyčku 4-20 mA, přičemž na analogový proudový signál moduluje digitální data. Způsob modulace odpovídá standardu Bell 202, který využívá FSK modulaci, viz. Obr. 5. Dosažená komunikační rychlost je 1200 Bd.



Obr. 5 Modulace protokolu HART

Vzhledem k tomu, že střední hodnota namodulovaného složky signálu je nulová, nedochází k poškození údaje přenášeného analogovou proudovou smyčkou. Analogovou smyčkou může být přenášen např. signál ze staršího zařízení, které protokol HART nepodporuje.

13.2.2 Komunikace s inteligentními čidly

Každá zpráva, kterou master vysílá, obsahuje příkaz (command) a případně i data. Slave tento příkaz interpretuje, vykoná a výsledek operace indikuje pomocí dvou byte vyhrazených pro

status. Definovaných příkazů je celá řada, avšak povinných příkazů, které musí podporovat každý slave, je v současné verzi HART protokolu (verze 5) pouze třináct. Jedná se o následující příkazy:

Přečti výrobce a typ zařízení	Přečti výrobní číslo
Přečti primární měřenou veličinu a jednotku	Přečti sériové číslo
Přečti proud a procenta z rozsahu	Zapiš popisku, popis a datum
Přečti až 4 předdefinované veličiny	Zapiš zprávu
Přečti popisku, popis a datum	Zapiš výrobní číslo
Přečti zprávu	Zapiš adresu
Přečti rozsah, jednotku a tlumení	

Další příkazy, které již nejsou povinné mj. umožňují kalibraci (nula a zesílení), nastavení fyzikální jednotky, nastavení tlumení, zápis sériového čísla, volbu čtyř měřených fyzikálních veličin, reset čidla, atd. Výše uvedené příkazy ilustrují možnosti, které HART poskytuje.

Protokol HART je typu master-slave, což znamená, že připojená čidla posílají zprávu až po té, co k tomu byla vyzvána masterem. Poněkud netypickým jevem je skutečnost, že HART umožňuje existenci dvou zařízení typu master. Důvodem pro toto řešení je usnadnění konfigurace sítě, neboť se předpokládá, primární master bude řídicí člen sítě sbírající naměřené údaje a sekundární master slouží pro konfiguraci sítě ručním konfigurátorem. Na jednu proudovou smyčku 4-20 mA je možné připojit až 15 zařízení typu slave. Formát zprávy HART protokolu je na Obr. 6.

ÚVODNÍ SEKVENCE	STARTOVACÍ ZNAK	ADRESA	PŘÍKAZ	POČET BYTŮ	NEPOVINNÝ STATUS	NEPOVINNÁ DATA	KONTROLNÍ SOUČET
-----------------	-----------------	--------	--------	------------	------------------	----------------	------------------

Obr. 6 Struktura zprávy protokolu HART

13.2.3 Startovací sekvence

Před každou zprávou je vyslána úvodní sekvence, která obsahuje alespoň tři byte (znaky) s hodnotou FFh (všechny bity mají hodnotu 1). Některá zařízení vyžadují, aby počet úvodních znaků s hodnotou FFh byl vyšší než tři. Tyto úvodní znaky umožňují přijímající straně synchronizovat se na vysílaný signál a jsou proto nezbytné. Po těchto úvodních znacích následuje startovací znak (startovací byte), kterým se rozlišuje typ zprávy. Možné typy zpráv a příslušná hodnota startovacího byte jsou uvedeny v následující tabulce:

	Krátký formát rámce	Dlouhý formát rámce
Master → slave	02h	82h
Slave → master	06h	86h
Burst mód slave → master	01h	81h

13.2.4 Adresace zařízení

HART rozeznává dva druhy adres - krátké adresy (short frame format) a dlouhé adresy (long frame format). Zda se jedná o zprávu v krátkém nebo dlouhém formátu lze rozhodnout na

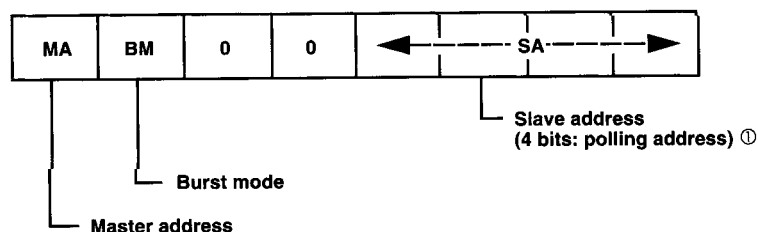
základě startovacího byte. Dlouhý formát adres podporuje až HART ve verzi 5 a novější. Stará zařízení podporují pouze krátký formát zpráv.

13.2.4.1 Adresa mastera

Pro rozlišení mastera se používá nejvýznamnější bit adresového pole. Hodnota tohoto bitu u primárního mastera je 1, u sekundárního mastera je 0.

13.2.4.2 Krátká adresa

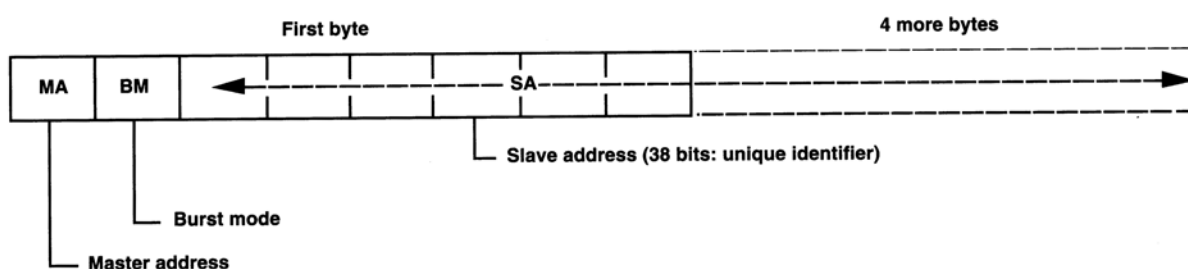
Krátký formát zpráv pro adresu využívá 1 adresový byte. Krátká adresa může nabývat hodnot 0 až 15 a v adresovém byte jsou pro ni vyhrazeny 4 nejméně významné bity, viz Obr. 7.



Obr. 7 Adresa ve zprávě krátkého formátu

13.2.4.3 Dlouhá adresa

Dlouhý formát zpráv pro adresu využívá 5 byte, přičemž slave má pro adresu vyhrazeno 38 nejméně významných bitů, viz Obr. 8. Dlouhá adresa je složena z výrobce zařízení, typu zařízení a identifikačního čísla zařízení.



Obr. 8 Adresa ve zprávě dlouhého formátu

13.2.5 Příkaz

Pro příkaz je ve zprávě jeden byte. Příkazy jsou rozděleny na univerzální a tedy povinné (Universal Commands), obvykle používané (Common-Practice Commands), a specifické pro dané zařízení (Device Specific Commands). Pro všechny standardizované příkazy je definován jejich význam, způsob předání dat v datovém poli a návratová hodnota v poli Status.

13.2.6 Počet byte

V položce počet byte je udáno kolik byte zpráva obsahuje. Přijímající strana tento údaj použije k nalezení byte s kontrolním součtem přijaté zprávy. Do počtu byte se zahrnují všechny informační byte. Tedy všechny byte počínaje startovacím byte, kromě kontrolního součtu.

13.2.7 Status

Status je zahrnut pouze ve zprávách, které odesílá slave a obsahuje dva byte. První byte obsahuje informace o chybách při komunikaci (např. chyba parity, chyba tvaru rámce, chyba v kontrolním součtu, přetečení přijímacího bufferu, ...) a o zpracování přijatého příkazu. Druhý byte informuje o stavu zařízení a výsledku poslední operace. V případě, že nedošlo k žádným chybám, jsou oba dva byte nulové.

13.2.8 Data

Největší povolené množství dat přenesené jednou zprávou je 25 byte. Formát dat ve zprávě je součástí specifikace jednotlivých příkazů. Dovolené typy dat jsou:

- a) celé číslo bez znaménka (8, 16, nebo 24 bitů);
- b) číslo s plovoucí řádovou čárkou podle IEEE 754 (32 bitů);
- c) ASCII řetězec ve formě "packed ASCII", což je speciální formát, ve kterém je pro jeden ASCII znak vyhrazeno pouze 6 bitů. Pro přenesení čtyř ASCII znaků jsou nutné pouze tři byte;
- d) položka z výčtu, která je určena jedním byte (8 bitů). Výčty jsou definovány ve specifikaci protokolu HART a používají se např. pro identifikaci výrobce zařízení, typu zařízení, fyzikálních jednotek, atd.

13.2.9 Kontrolní součet

Pro kontrolní součet je vyhrazen jeden byte. Kontrolní součet je vypočten jako XOR všech informačních byte, tedy všech byte, které se započítávají do počtu byte zprávy. takto koncipovaný kontrolní součet zajišťuje spolehlivou detekci shluku tří po sobě jdoucích chybných bitů a s dobrou pravděpodobností detekuje i další chyby a shluky chyb.

13.2.10 Burst mód

Burst mód je speciální mód, který umožňuje zařízení typu slave vyslat více zpráv najednou. Burst mód nemusí být v zařízení implementován. V případě použití burst módu slave opakovaně posílá zprávy až do doby, než master nařídí ukončit tento mód. O zahájení a ukončení burst módu rozhoduje master. Slave mezi vysílanými zprávami nechává krátkou pauzu, kterou master může využít k vyslání příkazu, jenž burst mód ukončí. Použití burst módu má smysl pouze tehdy, pokud je na daný pár vodičů zapojen pouze jeden slave.

13.2.11 Fyzická vrstva protokolu HART

13.2.11.1 Kabeláž

Doporučuje se použít stíněný kroucený pár. Obsahuje-li kabel více párů, je možné použít kabel se společným stíněním pro všechny páry, avšak v tomto případě je nutné dbát na to, aby ony společně stíněné kabely nepřenášely signály, které by mohly s protokolem HART interferovat. Je přípustné aby v kabelu, který obsahuje kroucené páry se společným stíněním, více párů pracovalo s protokolem HART. Maximální délka HART kabelu je 1500 m, avšak pro použití kabeláže delší než jednotky metrů se doporučují kvalitní kabely, které zajistí, že sběrnice HART bude mít časovou konstantu (RC) menší nebo rovnu 65 μ s. Rovněž s použitím více zařízení na jednom páru vodičů výrazně klesá maximální použitelná délka kabeláže. Např. pro 5 připojených

slave zařízení se maximální délka kabeláže pohybuje v závislosti na parametrech kabelu v rozmezí cca. 25 až 75 metrů.

13.2.11.2 Zařízení

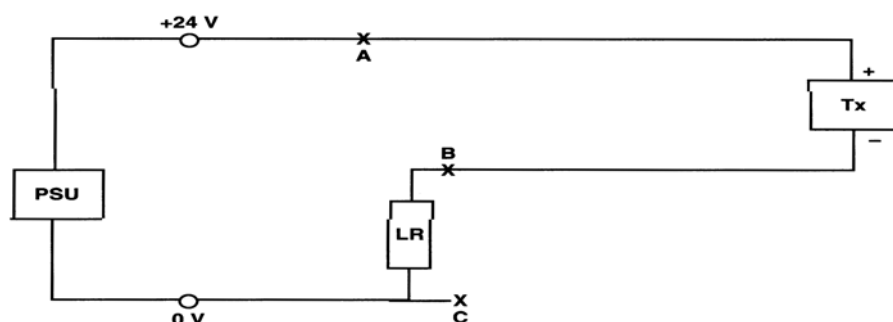
Všechna zařízení (master i slave) mají předepsanou minimální a maximální impedanci. Maximální kapacita slave zařízení je pouze doporučená jako 5000pF. Vzhledem k tomu, že pro stanovení maximální přípustné délky kabeláže je nutné znát mj. skutečnou kapacitu zařízení, je užitečné vědět, že kapacita zařízení HART se někdy udává ve formě Capacitance Number (CN). CN pro dané zařízení se získá jako podíl skutečné kapacity v pF a maximální doporučené kapacity 5000 pF. Výsledek tohoto podílu se zaokrouhlí nahoru. Tedy např. zařízení o kapacitě 22 nF (22 000 pF) má CN hodnotu 5.

13.2.12 Rychlost protokolu HART

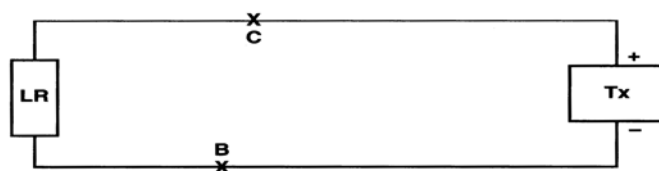
Protokol HART umožňuje každou sekundu provést dvě výměny zpráv. V případě využití burst módu je možné za jednu sekundu přenést nejméně tři zprávy. HART protokol je tedy poměrně pomalý a není vhodný pro rychlé procesy.

13.2.13 Možné topologie HART sběrnice

Zařízení připojitelná na sběrnici HART lze rozlišit na zařízení pasivní (napájená ze sběrnice) a na zařízení aktivní (která sběrnici napájejí). Připojení pasivních zařízení ilustruje Obr. 9, připojení aktivních zařízení ilustruje Obr. 10.



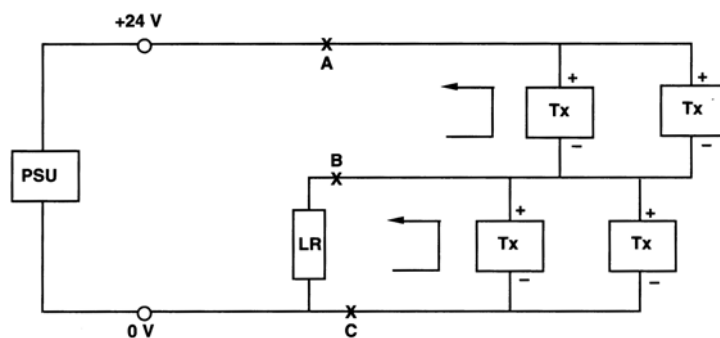
Obr. 9 Zapojení pasivních zařízení



Obr. 10 Zapojení aktivních zařízení

- PSU - napájecí zdroj
- LR - zátěž (rezistor, na kterém protékající proud vytváří napětí)
- Tx - HART zařízení (pasivní nebo aktivní)
- A, B, C - body, k nimž se připojují další HART zařízení

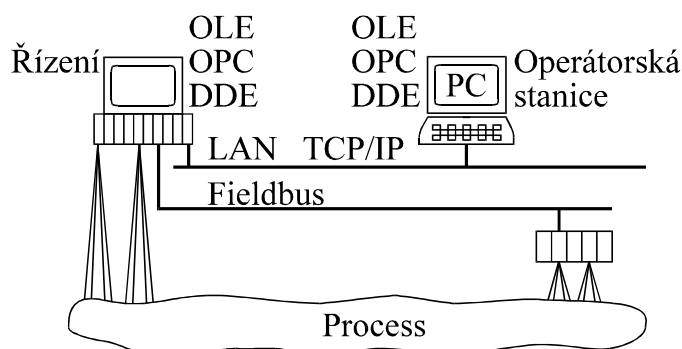
Kombinaci aktivních a pasivních zařízení v jednom segmentu sítě HART ilustruje Obr. 11, přičemž horní větev obsahuje aktivní zařízení, zatímco dolní větev obsahuje pasivní zařízení. Šipky naznačují směr průtoku el. proudu v HART síti.



Obr. 11 Kombinace aktivních a pasivních zařízení

14. SYSTÉMY SOFT CONTROL A PERSPEKTIVY AUTOMATIZAČNÍ TECHNIKY

Kromě kompaktních (velkých) DCS a PLC orientovaných distribuovaných řídicích systémů stále přibývá PC orientovaných DCS systémů, představujících ve větší či menší míře tzv. soft control, tedy řízení procesů, linek a strojů systémy, jejichž základ tvoří výkonná PC bez dalších stanic na procesní (technologické) úrovni řízení. Podobně jako v případě soft PLC i význam soft DCS se bude s postupem času zvětšovat. Pro jejich masivní rozšíření hovoří nižší cena vývoje, větší flexibilita a otevřenost, větší využívání standardů a otevřených SW produktů, vycházející z otevřenosti SW rozhraní. V méně exponovaných technologiích, než jsou elektrárny, chemické a farmaceutické provozy, válcovny, ocelárny, železárny a další těžké provozy, stejně jako nebezpečné provozy, již dnes kompaktní tradiční DCS narážejí na velkou konkurenci ze strany soft control. Rovněž počet předvedených produktů DCS na největší evropské automatizační výstavě Interkama 99 (18.-23.říjen 1999), která ukazuje současný stav a vývoj automatizační techniky na další dva roky, hovoří jasně ve prospěch soft DCS. Blokové schéma systému soft control je uvedeno na obr. 10.1.



Obr. 10.1: Schéma soft control

Problém řízení se transformuje na problém komunikace v jednotném transparentním SW prostředí. Stanice jsou realizovány průmyslovými PC (tzv. IPC) s kartami I/O, umístěnými na vnitřní paralelní sběrnici PC (ISA, PCI). Prostředí Microsoft Windows umožňuje použití OLE, OPC, DDE libovolnou stanicí za účelem výměny dat. SW prostředky pro operátorské řízení a sběr dat (SCADA systémy) využívají tohoto jednotného prostředí a umožňují vizualizaci, nadřazené řízení, ale i realizaci řídicích algoritmů, jejich implementaci, testování simulací i přímé DDC (Factory Suite 2000/ InControl 7.0). Přenosová rychlost a dlouhodobé a velmi široké použití protokolu a sítě Ethernet (TCP/IP) bude význam soft control systémů stále zvětšovat. Soft control se tak v blízké budoucnosti stanou konkurentem všem řídicím prostředkům v průmyslové automatizaci.

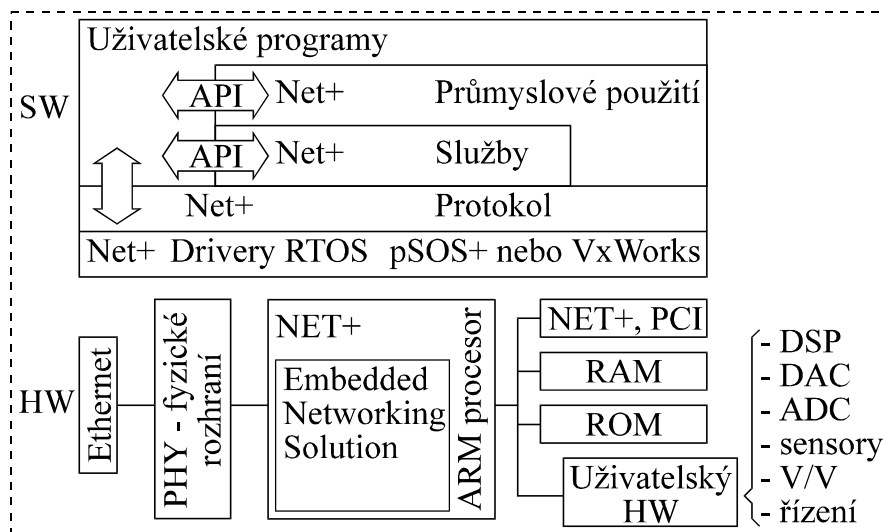
14.1 Trendy vývoje automatizační techniky

Téma této kapitoly je uvozeno již v posledních odstavcích předcházející kapitoly. Zde již jen uvedme aktuální doplnění k vývoji Ethernetu k budoucímu standardu v oblasti průmyslové komunikace.

Na Hannoverském veletrhu 1999 byly poprvé předvedeny produkty několika výrobců, které jako komunikační protokol využívají Ethernet. Firma Beckhoff představila svoji inteligentní svorkovnici (velmi podobnou produktu firmy WAGO nebo Weidmuller) s komunikačním modulem BK9000, který komunikuje protokolem Ethernet a dle provedení využívá rychlosti 10 nebo 100Mbitů/s. K modulu může být připojeno až 64 I/O modulů. Běžně se v této konfiguraci dosud používaly nejrůznější protokoly a sériová rozhraní (CAN, LON, Profibus DP a další). Dalším výrobcem systému bezprostředního řízení propojeného přímo na Ethernet je firma Schneider Electric. K systému distribuovaných vstupů a výstupů, resp. malému PLC typu Momentum rovněž přiřadila Ethernet s přenosovou rychlostí 10Mbitů/s. Momentum má k dispozici 20 různých modulů včetně analogových I/O. Firma OPTO 22 (www.opto22.de) předvedla podobný systém. Spojení přes Ethernet s libovolným nadřazeným systémem je zajištěno využitím protokolů TCT/IP. Rychlost přenosu je do 100Mbitů/s. V systému je integrován WEB browser, tudíž možnost vizualizace po Internetu. Modulární malý řídicí systém SMART2 firmy PEP Modular Computers nabízí rovněž připojení tohoto malého řídicího systému na Ethernet. Tento produkt dokonce v sobě spojuje malý PLC s výkonným systémem vnitřní sběrnice VME a umožňuje tak použití výkonného operačního systému reálného času. Pro sériové spojení s okolím nabízí v jedné variantě Profibus, v nové variantě pak Ethernet. Provedení s Ethernetem je již nyní levnější. Rovněž National Instrument přichází s modulem FP-1600 pro modulární I/O systém FieldPoint. Komunikační rychlost 10 nebo 100 Mbitů/s.

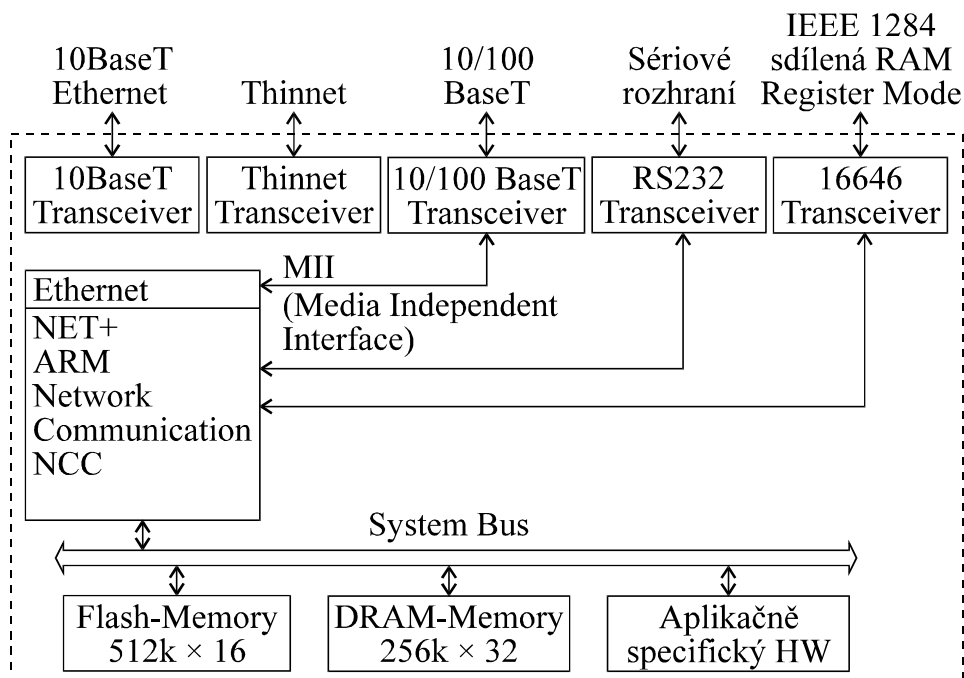
Tato skutečnost vyvolala značnou pozornost médií jak přímo na Hannoverském veletrhu, který ukazuje směr technického vývoje a reflektuje stav techniky v daném roce (duben), tak i v odborném a populárně-odborném tisku. I velcí výrobci řídicí techniky (Siemens, Allen-Bradley), kteří věnovali velké prostředky pro realizaci poměrně nových systémů (Simatic řada S7, ControlLogix ap.), musejí reagovat na tuto situaci, vyvolanou spíše menšími flexibilnějšími producenty, napojenými více na informační a komerční systémy (z principu podporujícími LAN Ethernet). V důsledku jasných signálů o tomto vývoji je zřejmé, že velká varieta fieldbusů tak, jak nyní charakterizuje automatizační systémy, bude v blízké budoucnosti vystřídána koncentrací na výrazně redukovanou množinu systémů. Vzhledem ke stále ještě vyšší ceně nových systémů, ještě stále nedostatečné rychlosti Ethernetu (při zatížení sítě a CSMA/CD přístupové metodě), mají komunikační systémy, zaručující přístup v reálném čase nebo alespoň v definovaném čase určitou šanci. Vývoj mikroelektroniky, tím klesající ceny modulů Ethernet a stále vyšší přenosová rychlost média, však významně hrají pro Ethernet. Kromě toho je zda lákavá jednotnost s LANy, využití Internetu a celého komfortního SW prostředí, které poskytují PC orientované systémy. Rozhodujícím momentem v dalším vývoji však bude stabilita operačních systémů. Prozatím ještě není na trhu spolehlivý levný operační systém, splňující tvrdé požadavky řízení v reálném čase a spolehlivosti při zachování komfortu, který umožňuje Internetové prostředí. Avšak i v tomto směru jsou k posouzení první pokusy.

NetSilicon (Waltham/Massachusetts) vyvinula pro Ethernetové uzly jak HW, s velmi výkonnými procesory, tak SW vývojové prostředí a operační systém reálného času [25].



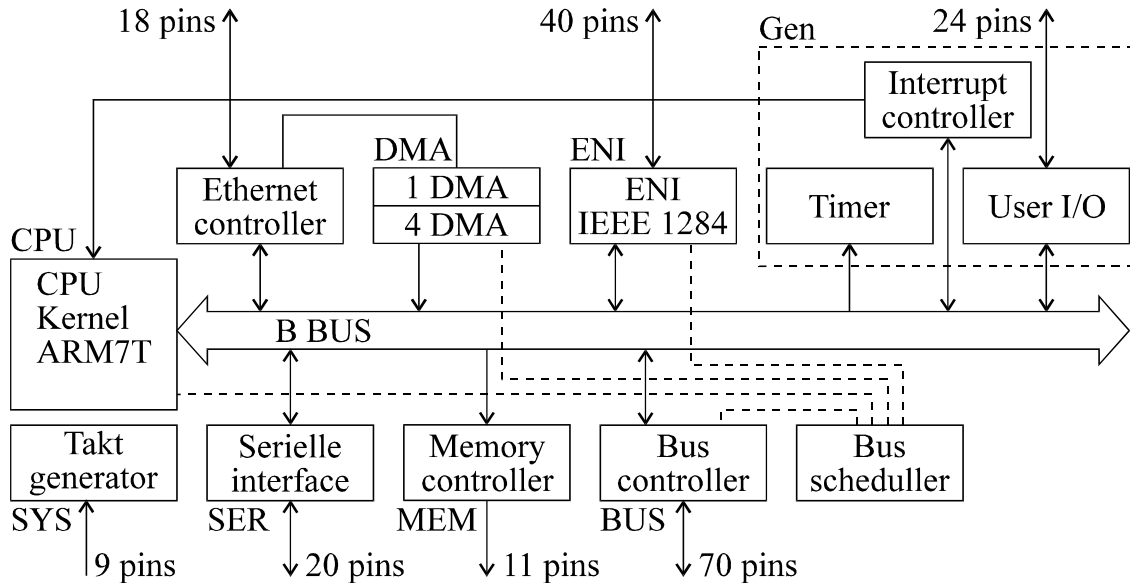
Obr. 10.2: HW a SW Ethernetového uzlu

Na obr. 10.2 je HW a SW řešení karty pro Ethernetový uzel. Jádrem HW tvoří NET+ARM mikroprocesor a je doplněno standardními obvody pro uživatelský interface a pro komunikační interface (zde Ethernet a celá řada dalších rozhraní). Podrobnější blokové schéma uzlu je na obr.10.3



Obr. 10.3: Blokové schéma uzlu

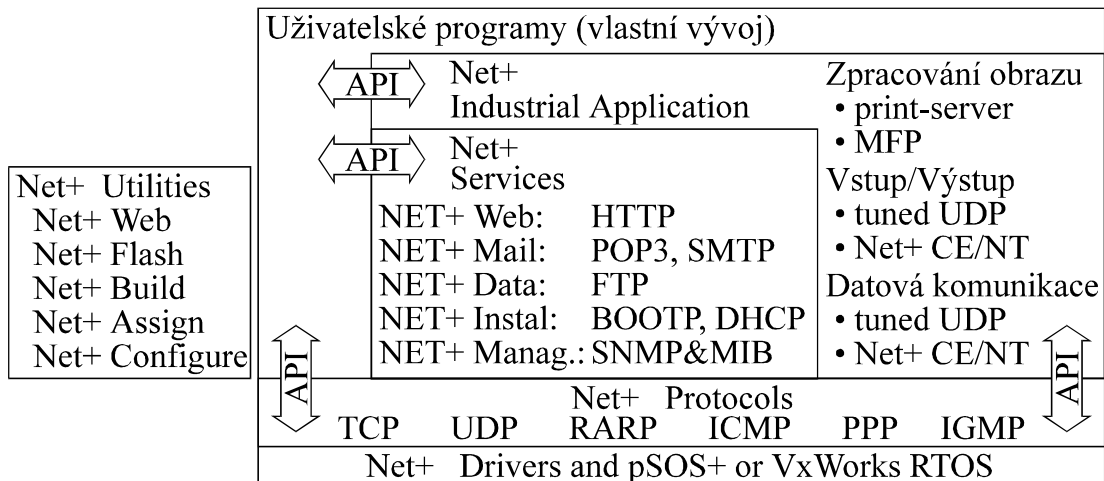
Jádrem je 32 bitový RISC procesor ARM7T s DMA řadičem, timerem, řadičem přerušení a uživatelským rozhraním (24 pinů). K němu patří 10/100Mbitů Ethernet MAC, ke kterému se dá



Obr.10.4: Vnitřní struktura obvodu

ještě připojit externí Ethernet tranceiver 10BaseT/100BaseT. K ARM7T patří ještě 4 portové rozhraní IEEE 1284 pro připojení nadřazeného systému a dále sériové rozhraní RS232C, řadič paměti, řadič vnitřní sběrnice a vnitřní hodiny (viz obr.10.4).

Prostřednictvím paměťového rozhraní se dá k procesoru připojit až 1,5 Gbyte RAM s 8, 16 nebo 32 bitovou délkou slova s rychlostí cyklu do 100MByte/s. Spotřeba čipu činí 330 až 750mW. Pouzdro čipu je vybaveno 208 piny v 0,5mm rastru.



- | | | | |
|------|------------------------------|------|---------------------------------|
| ADC | A/D převodník | MIB | Managed Information Table |
| API | Applic. Programing Interface | TCP | Transmission Control Protocol |
| DAC | D/A převodník | RTOS | Real-Time Operating System |
| HTML | Hyper Text Mark-up Lang. | SNMP | Simple Networ. Manage. Protocol |
| HTTP | Hyper Text Transfer Prot. | SOS | System On Silicon |
| IP | Internet Protocol | UDP | User Datagram Protocol |

Obr. 10.5: Programové vybavení

Programové vybavení tohoto mikrokontroleru sestává z driverů (NET+Drivers), protokolů (NET+Protocolls), služeb (NET+Services) a uživatelských programů, které si zákazník tvoří sám (viz obr.10.5).

Pro vnitřní program a real-time operační systém pSOS je třeba 250 až 360 KByte.

Se systémem je nejvhodnější začít v cenově výhodném provedení t.j. zakoupit Komplet NET+ARM (t.j. Software Toolkit a vývojové prostředí) sestávající z 32-bitového RISC procesoru ARM a NET+WEB softwaru s Internetovými protokoly (HTTP, FTP), Web-browserem, Mail-serverem a dalšími drivery a rozhraními k operačnímu systému reálného času pSOS. Kromě TCT protokolu je k dispozici i UDP protokol, což z hlediska časově kritických přenosů při procesním řízení hraje významně pro tento systém. DDP je podobně jako TCP normován pro použití na Internetu.

Ethernet však nebyl, jak je známo vyvinut pro řízení. To se dá u tohoto systému obejít tak, že akční zásahy (object) jsou přenášeny jako HTML stránka. Výrobce očekává, že bude zájem o tvorbu mezisběrníkových spojů mezi Ethernetem a stávajícími fieldbusy. Počítá proto s převedením dlouho vyvíjených funkčních bloků pro přístup na fieldbusy přímo na HTML stránky a ovládání procesu z Ethernetu.

Programátorský komfort je zaručen dalšími SW produkty, které jsou s uvedeným systémem svázány :

- NET+Web: kompilátor HTML stránek do C
- NET+Build: prostředí pro automatické generování kódu
- NET+Assign: přidělení sériového čísla MAC adresám
- NET+Flash: přenos programového kódu do paměti Flash
- NET+Configure: ukládání nebo čtení konfiguračních parametrů

Pro další rozšíření obdobných systémů a to v takové míře, aby se staly prostředkem, který nahradí PLC v jejich dominantní pozici při řízení je třeba splnit ještě několik předpokladů. Především cena za Ethernetový uzel musí klesnout z dnešních 100 \$ na 10 až 15\$. Cena za NET+ ARM čip musí proto klesnout z dnešních 10 až 30 \$ na 7 až 20\$. Dále průmyslové provedení tohoto systému musí umožnit velmi jednoduchou instalaci (Plug and Play), jak je to dnes u PLC. Kromě toho se musejí tyto nové systémy vyrábět v dostatečně široké paletě provedení, aby odpovídaly široké paletě průmyslových aplikací, které se s nimi budou řešit.

15. LITERATURA

- [1] Schnell G. (Hrsg) : Bussysteme in der Automatisierungstechnik, VieWeg Verl., Braunschweig, 1994, ISBN 3-528-06569-9
- [2] Auer A. : SPS Aufbau und Programmierung, 4. Aufl., Hütting, Heidelberg, 1991
- [3] Considine D.M.: Process / Industrial Instrumentation and Control Handbook, 4th Ed. McGraw Hill, N.Y., 1993, ISBN 0-07-012445-0
- [4] Gevatter H. – J.: Handbuch der Meß- und Automatisierungstechnik, Springer Verl. Berlin, 1999, ISBN-3-540-59135-4
- [5] Reinhardt H.: Automatisierungstechnik, Springer Lehrbuch, Berlin 1996, ISBN-3-540-60626-2
- [6] Kriesel – Heimbold – Teltschow : Bustechnologien für die Automatik, VieWeg Verl., Braunschweig 1998
- [7] Warnock I.G.: Programmable Controllers, Practice Hall, London, 1988
- [8] Wollert J. – Fiedler J.: Automatisieren mit dem PC, 2. Aufl., Springer Verl., Berlin, 1998
- [9] Schnell G. (Hrsh): Sensoren in der Automatisierungstechnik, VieWeg, 2. Auflage, Braunschweig, 1993, ISBN 3-528-13370-8
- [10] Ahlers Horst. (Hrsg) : Multisensorik Praxis, Springer, Berlin, 1996, ISBN 3-540-58997-X
- [11] Blatný J.– Drábek L. : Standardizace číslicových rozhraní, skriptum MON, Praha, 1990
- [12] Olsson G., Piani G.: Computer Systems for Automation and Control, Practice Hall, N.Y., 1992, ISBN 0-13-457581-4
- [13] Suchý K.: Foudation fieldbus, Automatizace, 1998, č.7. str. 412–416
- [14] Kocourek P.: Průmyslové komunikační systémy, Automatizace, 1998, č.7, str. 390–392.
- [15] Spurý K.: CAN, Automatizace 1998, č.7, str. 397–400,
- [16] SPS Magazin, 1999 Nr.1+2, S. 72–79
- [17] SPS Magazin, 1999, Nr.3., S. 58–64
- [18] Zezulka F.: Mezinárodní standardizace v oblasti průmyslových komunikačních sběrnic, Automatizace 1998, č.7, str. 393–396
- [19] Hanzálek Z. –Novák T.–Novotný R. : Profibus, Automatizace 1998, č.7, str. 437–411
- [20] Kostka F. – Katz M. – Bender K.: Profibus na cestě k evropské stand., Automatizace 1993, č.9, str. 270–272.
- [21] Simatic PCS7 : www.ad.siemens.de/simatic-pcs7
- [22] Vrátil Z. – Beneš P. – Chrást J.: Modemy a sériový přenos dat, Gethon, 1992
- [23] Produktübersicht ATP, Sonderheft zur Interkama'99, Oldenbourg, Oktober, 1999
- [24] Früh K.F. (Hrsg.): Handbuch der Prozeßautomatisierung, Oldenbourg Verl. München, 1997, ISBN-3-486-24025-0

- [25] Wollert J.F. – Fiedler J.: Automatisieren mit dem PC, VDI-Verl., Düsseldorf, 1995, ISBN 3-18-401318-9
- [26] Strass H. – Evensen M. : Die totale Industrie Vernetzung, Elektronik, 1999, Nr. 12, S. 46-51