

Komunikační protokol CANopen

Základní informace a zadání úloh

Katedra řídicí techniky, skupina řídicích systémů

Obsah

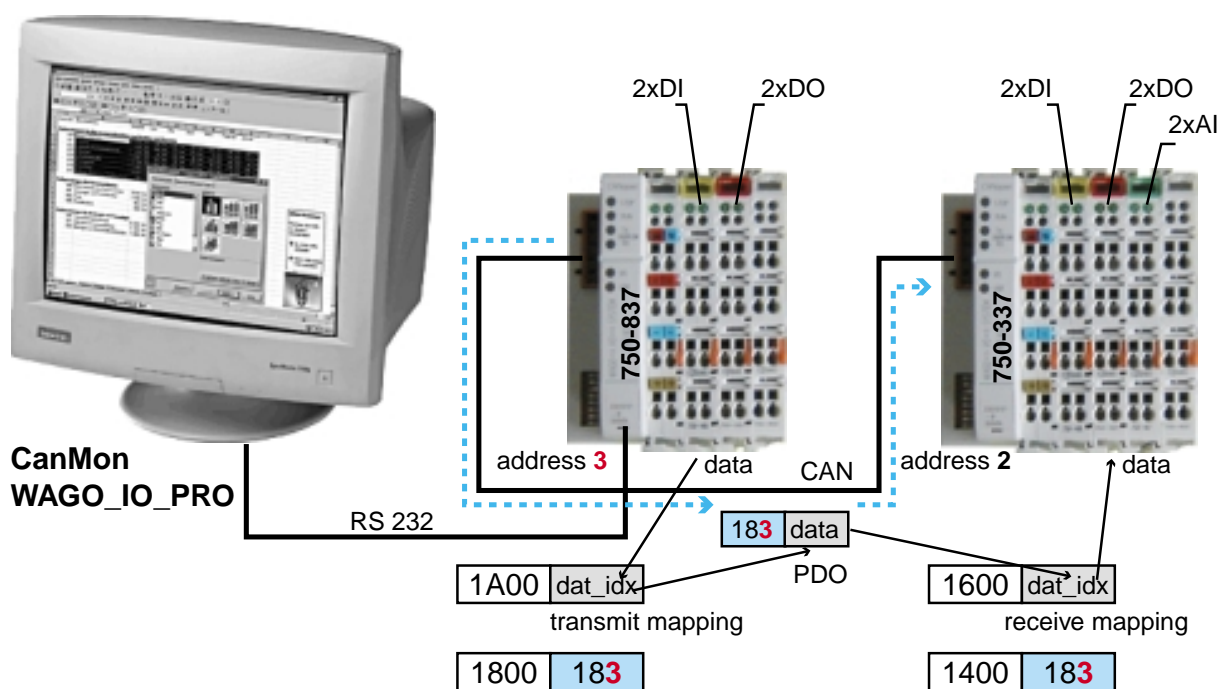
1	Popis a zadání úlohy	2
1.1	Opravdu důležitý úvod	2
1.2	Zadání úlohy	4
1.2.1	Přečtení parametrů všech zařízení	4
1.2.2	Kopírování vstupu stanice 3 na stanici 2	6
1.2.3	Konfigurace pomocí WAGO PLC	6
2	Základní informace o CANopen	9
2.1	Komunikační model CANopen	10
2.1.1	Mapování aplikačních objektů na PDO	12
2.1.2	Přenos dat SDO	12
2.2	Adresář objektů CANopen	13
2.3	Komunikační profil	14
2.4	Profily zařízení	16

Kapitola 1

Popis a zadání úlohy

1.1 Opravdu důležitý úvod

Model pro práci s komunikační sběrnici CANopen je znázorněn na obrázku 1.1 a skládá se ze dvou zařízení WAGO, a to 750-837 a 750-337. První z nich je programovatelný logický automat, do nějž lze programy nahrávat z prostředí WAGO_IO_PRO po sériové lince RS 232. Na obrázku vidíme kromě použitých zařízení



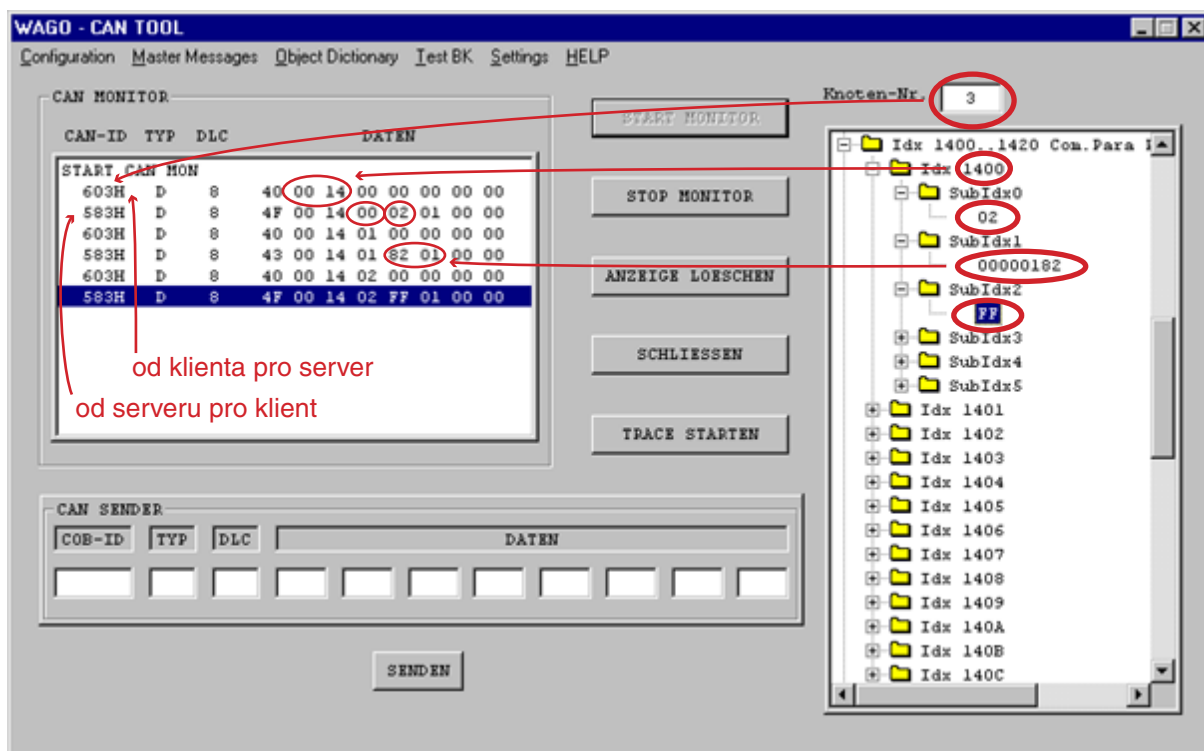
Obrázek 1.1: Schema modelu

také příklad přenosu vstupů ze stanice s adresou 3 na výstupy stanice 2. Tento popis odpovídá zadání uvedenému v odstavci 1.2.2 a jedná se o následující údaje:

- v zařízení s adresou 3 musíme nastavit parametry vysílaných procesních dat (PDO). Podle standardního komunikačního profilu se tyto parametry nacházejí na indexu 0x1800.

- v zařízení s adresou 3 musíme nastavit mapování fyzických vstupů do odpovídajícího rámce procesních dat, který je vysílán na sběrnici (Transmit PDO). Mapování se nastavuje na indexu 0x1A00. Jestliže je na tomto indexu nastaveno umístění vstupních dat v adresáři objektů (viz standardní I/O profil v tabulce 1.5) a na indexu 0x1800 je nastavena hodnota COB_ID na 0x183, znamená to, že příslušná data ze vstupního modulu zařízení 3 budou přenášena v CANovské zprávě s identifikátorem 0x183.
- v zařízení s adresou 2 musíme nastavit mapování fyzických výstupů na indexu 0x1600 tak, aby přijatá data byla umístěna na správné místo v adresáři objektů podle standardního profilu I/O (viz tabulka 1.5).
- v zařízení s adresou 2 musíme nastavit parametry přijímaných procesních dat, především pak COB_ID, který musí být stejný, jako je COB_ID dat, která jsou vysílána zařízením s adresou 3, tj. COB_ID = 0x183.

Na PC, který je připojen k zařízení 750-837, se v adresáři *D:\Canopen\Monitor* nachází aplikace Can-Mon, jejíž ovládání je znázorněno na obrázku 1.2. Obrázek ukazuje, jak se nastavují jednotlivé parametry



Obrázek 1.2: Obrazovka monitoru komunikace CANopen

pro přijímaná procesní data (Receive PDO) v adresáři objektů v pravé části obrazovky a jak tyto údaje odpovídají skutečným datům přenášným v rámcích CAN. Rovněž je vidět, jak jsou vytvořeny příslušné identifikátory rámců CAN, které jsou totožné s COB_ID popsaným pro model klient/server v části 2.3. Uvědomte si znovu, že při tomto typu komunikace se otevře spojení mezi zařízeními klient a server (v našem případě to jsou stanice 2 jako klient a stanice 3 jako server) a v rámci takového spojení lze měnit nebo číst hodnoty libovolné položky v adresáři objektů. *Prostudujte obrázek 1.2, abyste si uvědomili význam přenášných dat.* Rovněž si uvědomte, že v zařízení existuje vždy alespoň jeden objekt SDO

Server, který má nastaven COB_ID v adresáři objektů v položce SDO Server, která je umístěna na indexu 0x1200 a subindexu 0x01 na hodnotu 0x600 + adresa stanice (viz tabulka 2.4).

Jednotlivé kroky, které je třeba provést, abychom aplikaci CanMon dostali do stavu, který je na obrázku 1.2, jsou následující:

- nastavte sériový port na PC na COM2. To se provede v menu v položce *Settings*.
- nastavte adresu stanice v políčku *Knoten-Nr.*
- stiskněte tlačítko *Start Monitor*.
- položky označené červeným kroužkem najdete v tabulce 1.1 a uvědomte si jejich význam i v souvislosti s daty, která jsou posílána na sběrnici.
- v aplikaci CanMon budete řešit body 1 a 2 zadání, které jsou popsány v částech 1.2.1 a 1.2.2.

Při práci s aplikací CanMon je třeba dodržovat následující postup, jestliže potřebujeme změnit hodnotu některé položky v adresáři objektů:

- klikneme myší na požadovanou položku a nastavíme nejvyšší bit na 1, tj. číslice nejvíce vlevo musí být 8. To způsobí při doručení dat do patřičné položky adresáře objektů vymazání staré hodnoty.
- myší klikneme na jinou položku, nestačí pouhý stisk klávesy Enter. Tím došlo k odeslání rámce na sběrnici CAN s příslušnými daty s nejvyšším bitem rovným 1, což můžeme kontrolovat v levé části okna aplikace.
- nyní postup opakujeme s tím, že nejvyšší bit bude nastaven na 0 a nastavíme položku na požadovanou hodnotu.

1.2 Zadání úlohy

1.2.1 Přečtení parametrů všech zařízení

Přečtěte parametry všech zařízení CANopen připojených na sběrnici CAN. Jedná se především o parametry uvedené v tabulkách 1.1 až 1.4, které se týkají komunikačních parametrů pro vysílání a přijímání dat, a dále potom v tabulce 1.5, která popisuje základní parametry standardního profilu DS401.¹

Index	Subindex	Popis
Idx1400	SubIdx0	komunikační parametry Rx (receive): první byte výstupů
	SubIdx1	celkový počet sub indexů v daném indexu
	SubIdx2	COB ID, přes který bude probíhat výměna dat, tj. identifikátor zprávy, která bude obsahovat výstupní data. Když se na sběrnici objeví zpráva s tímto COB ID, data v ní obsažená se zapíšou na výstup tj. identifikátor zprávy, kterou bude toto zařízení přijímat
Idx1401		typ komunikace (0xFF ... state_of_change) dtto, ale druhý byte výstupů (kdyby byl připojen) atd.

Tabulka 1.1: Komunikační parametry Rx

¹Změna hodnot - zápisem 1 do nejvyššího bitu - např. Idx1400, SubIdx 1. Zařízení Weidmueller - nefunguje změna COB ID na Idx1400, SubIdx 1 - zapíšeme 183, přečteme 183, ale zařízení reaguje na předcházející COB ID 205.

Index	Subindex	Popis
Idx1600		mapování Rx PDO - je zde řečeno, jak se budou přenášená data interpretovat, tj. kam se budou přijaté hodnoty výstupů fyzicky mapovat
	SubIdx0	počet fyzických bytů
	SubIdx1	kam jsou data mapována na fyzické výstupy, tj. na příslušné aplikační objekty (xxxxyyzz: xxxx - Idx, yy - pořadí bytu, zz - počet bitů). Ve standardním profilu DS301 jsou tyto údaje uvedeny.

Tabulka 1.2: Mapování Rx PDO

Index	Subindex	Popis
Idx1800		komunikační parametry Tx (transmit): první byte vstupů
	SubIdx0	
	⋮	jako u parametrů Rx
	SubIdx2	

Tabulka 1.3: Komunikační parametry Tx

Index	Subindex	Popis
Idx1A00		mapování Tx PDO - viz Idx1600
⋮	⋮	⋮

Tabulka 1.4: Mapování Tx PDO

Index	Popis
Idx6000	STDP Digital Input
Idx6005	globální povolení digitálních vstupů (0 - zakázáno, 1 - povoleno, zařízení může vysílat na sběrnici zprávy obsahující hodnoty vstupů)
Idx6006	maskování vstupů
Idx6007	maskování hran na vstupech
Idx6008	
Idx6200	STDP digital output
⋮	⋮
Idx6700	...

Tabulka 1.5: Standardní profil zařízení

1.2.2 Kopírování vstupu stanice 3 na stanici 2

Nastavením Tx parametrů stanice 3 a Rx parametrů stanice 2 na stejnou hodnotu COB ID docílíme toho, že vstupy (Tx parametry) stanice 3 se budou přímo zobrazovat na výstupech (Rx parametry) stanice 2. Proved'te.

1.2.3 Konfigurace pomocí WAGO PLC

WAGO PLC vystupuje na sběrnici jako NMT master, který dokáže konfigurovat ostatní zařízení. Zapsáním programu do PLC pomocí aplikace WAGO_IO_PRO proved'te kopírování vstupů ze stanice 2 na výstupy stanice 3.

- nastavte použití funkce SDO_WRITE4.
- nastavte nastavení parametrů SDO zpráv pro server a klient
- doplňte ukázkový program (část bloku CASE pro hodnoty 4 až 7), aby odpovídaly zadání

Celý program pro WAGO_IO_PRO je vysvětlen v následujícím textu. Jeho kostrou je konstrukce *case ... end_case*, viditelná ve výpisu 1.1, který popisuje jakýsi stavový automat, který postupně prochází stavy 0 ... 8, přičemž stav 8 je koncový stav (proto je v tomto stavu nastavena proměnná *Status = 18*, což znamená, že v žádném dalším cyklu PLC se již nebude konstrukce *case ... end_case* vykonávat.

Výpis 1.1: Hlavní struktura programu

CASE Status OF

```

0: (*Get local Node ID*)
  (*adresa stanice, kde bezi program - PLC*)
  Status:=1;

1: (*Enter Node 2: Index 0x1280, COB ID Client-->Server (tx)*)
  (*server SDO = 0x600 + ID_stanice v nasem pripade: 0x0602*)
  Status:=2;

2: (*Enter Node 2: Index 0x1280, COB ID server-->client (rx)*)
  (*client SDO = 0x580 + ID_stanice, v nasem pripade 0x582 *)
  Status:=3;

3: (*Enter Node 2: Index 0x1280, subidx 2 = Module ID 02*)
  (*je treba vlozit ID stanice, se kterou komunikujeme *)
  Status:=4;

4: (*make invalid the cob ID of the 1st tx PDO, node ID 3 to remap it*)
  Status:=5;

5: (*set 1st tx PDO of node 3 to COB ID 16#200*)
  Status:=6;

6: (*make invalid the COB ID of the 1st rx PDO, node ID 2 to remap it*)
  Status:=7;

7: (*set 1st tx PDO of node 2 to COB ID 16#200*)

```

```

    Status:=8;

    8:(*set nodes operational*)
      (*vyslání ramce, který spustí vzdalene stanice *)
      Status:=18;
END_CASE;

```

V programu, který je dostupný na webu DRS, jsou stavy 0 ... 3 a stav 8 hotové, což dává určitou představu, jakým způsobem se má program psát. Stav 4 ... 7 je třeba doplnit na základě zkušeností získaných z předchozích úkolů při práci s aplikací CanMon podle následujícího popisu:

4. V tomto stavu se musí vymazat stará hodnota COB_ID prvního PDO pro vysílání zápisem jedničky do nejvyššího bitu v odpovídající položce parametrů pro vysílání ve stanici s adresou 2, viz zadání této části (1.2.3).
5. Následně se musí zapsat nová hodnota odpovídající stanici, v našem případě bude přijímající stanice uzal s adresou 3. Jinými slovy, stanice 2 má vysílat svá vstupní data na sběrnici tak, aby je stanice s adresou 3 mohla přijmout a zapsat na své výstupy.
6. Stanice s adresou 3 je přijímající stanice, a tak je třeba nastavit její přijímací parametry na COB_ID, které vysílá stanice 2. Nejprve musíme opět smazat starou hodnotu zápisem jedničky do nejvyššího bitu.
7. V tomto stavu můžeme zapsat hodnotu novou.

Za zmínku stojí ještě použití funkce SDO_Write4, které je znázorněno na obrázku 1.3 a ve výpisu 1.2.

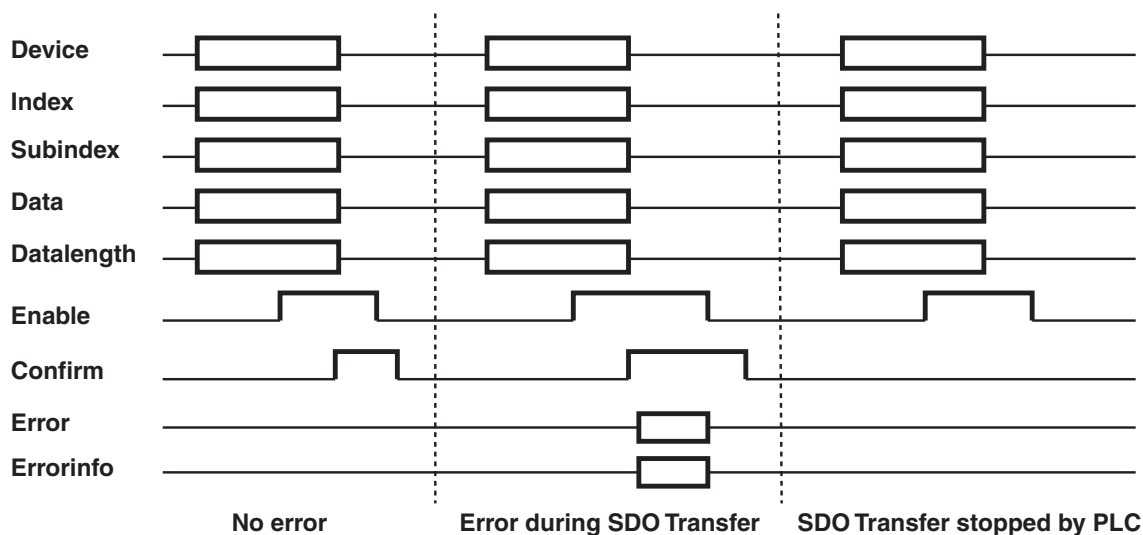
Výpis 1.2: SDO_Write4

```

1 CASE Status OF
2   :
3   1:(*Enter Node 2: Index 0x1280, COB ID Client-->Server (tx), 0x602*)
4     Data[1]:=16#02;
5     Data[2]:=16#06;
6     Data[3]:=16#00;
7     Data[4]:=16#00;
8
9     (* nabezna hrana na parametru ENABLE spusti funkcní blok SDO_WRITE4*)
10    SDO_WRITE4(DEVICE:=OwnNodeID, INDEX:=16#1280, SUBINDEX:=1,
11      ENABLE:=TRUE, DATA:=Data, DATALENGTH:=4 );
12
13    IF SDO_WRITE4.CONFIRM=TRUE THEN
14      errorcheck [status]:=SDO_WRITE4.error;
15      Status:=2;
16
17      (* nuluje se parametr ENABLE po uspesnem potvrzeni funkce SDO_WRITE4,
18        abychom se pripravili na dalsi nabeznu hranu *)
19      SDO_WRITE4(DEVICE:=OwnNodeID, INDEX:=16#1280, SUBINDEX:=1,
20        ENABLE:=FALSE, DATA:=Data, DATALENGTH:=4 );
21    END_IF;
22   :
23 END_CASE;

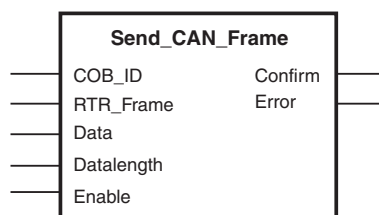
```


Jak vyplývá z obrázku 1.3, je třeba na vstupu ENABLE zajistit náběžnou hranu. To je provedeno zavoláním funkce s parametrem `ENABLE = TRUE` a následným čekáním na pozitivní návratovou hodnotu `CONFIRM`. Jestliže je `CONFIRM = TRUE`, potom opět podle obrázku byla funkce dokončena (případný chybový kód je v návratovém parametru `ERROR`), a tak můžeme parametr `ENABLE` vrátit na nulu. To nám rovněž zajistí to, že při dalším volání funkce s `ENABLE = TRUE` bude náběžná hrana na tomto parametru opět přítomna. Dále je třeba si všimnout, že data, která jsou funkci předávána, se nacházejí v poli `DATA` a že pořadí bytů ve wordu je obrácené, tj. pro předání hodnoty `0x602` je pořadí jednotlivých bytů `0x02`, `0x06`, jak je vidět na řádcích 4 a 5 ve výpisu 1.2.



Obrázek 1.3: Způsoby vykonávání funkce SDO_Write

Poslední poznámka patří ke stavu 8 stavového automatu, který provede spuštění všech stanic CANopen připojených ke sběrnici. Provede se to vysláním zprávy s `COB.ID = 0` a s daty `0x01`, `0x00` pomocí funkce `Send_CAN_Frame`, jejíž rozhraní je vidět na obrázku 1.4.



Obrázek 1.4: Uživatelské rozhraní funkce Send_CAN_Frame

Kapitola 2

Základní informace o CANopen

Protokol CAN obsahuje popis pouze první a druhé vrstvy podle modelu ISO OSI. Pro reálné aplikace je třeba, aby protokol definoval i vrstvu aplikační, která dodává význam datům přenášeným vrstvou linkovou. Linková vrstva CAN poskytuje dvě základní služby, které nejsou orientované na spojení (connectionless):

- nepotvrzovaný přenos zpráv CAN.¹
- nepotvrzovaná vzdálená žádost o zprávu CAN označená jako *RTR (Remote Transmission Request)*.²

Aby bylo použití CANu v automatizaci usnadněno, vznikly dva protokoly aplikační vrstvy (CAL a CANopen), které umožňují vzájemnou interoperabilitu zařízení od různých výrobců. Předmětem našeho zájmu zde bude CANopen, který rozšiřuje některé služby a protokoly definované vrstvou CAL a navíc definuje množství profilů pro většinu běžných zařízení používaných v průmyslové automatizaci. CANopen vychází z takzvaného „komunikačního profilu“ definovaného v CIA-301 a CIA-302 (CIA = CAN in Automation) a jednotlivé profily zařízení jsou CIA-401 pro digitální a analogové vstupy/výstupy, CIA-402 pro pohony, atd.

Popis funkčnosti zařízení, parametrů apod. se nachází v *adresáři objektů (Object Dictionary OD)*, který obsahuje část s obecnou specifikací zařízení, jako jeho název, výrobce, komunikační parametry atd., a potom část, která obsahuje určitou funkčnost zařízení, parametry a data. Každá položka v adresáři se nazývá objekt a je určena 16-bitovým indexem a 8-bitovým subindexem.

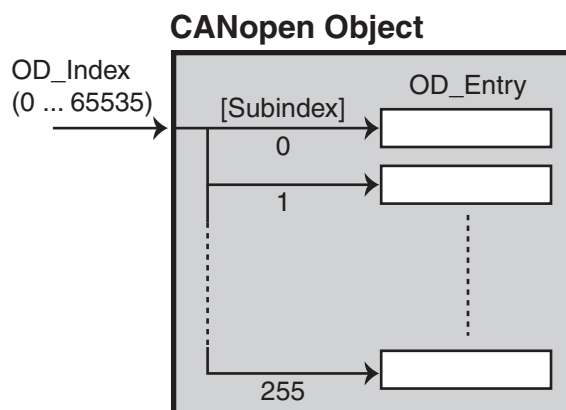
Rozlišují se dva základní mechanismy přenášení zpráv na CANopen. Procesní data určená pro časově kritickou výměnu, nazývaná *Process Data Objects PDO*, a služební zprávy, jejichž doručení není tolik omezeno časem, nazývané *Service Data Objects SDO*. Služební zprávy se používají především pro přenášení parametrů při konfiguraci zařízení nebo pro přenášení delších zpráv. Zpráva na CANu totiž může obsahovat maximálně 8 bytů dat.

Procesní data se přenášejí buď cyklicky, při změně nebo na vyžádání jako broadcastové zprávy bez dodatečné režie. Rozmístění aplikačních objektů (položek z adresáře objektů) do přenášeného objektu PDO je určeno pomocí tzv. mapování PDO – tato informace je uložena v adresáři objektů a může být změněna podle požadavků aplikace (uživatele), jak je ukázáno i v praktické části.

Přenos zpráv SDO odpovídá potvrzovanému přenosu podle modelu *client/server* a je prováděn jako komunikace mezi dvěma účastníky. Odpovídající položka je v adresáři objektů je popsána příslušným

¹Ve skutečnosti jsou zprávy příjemcem potvrzovány nastavením bitu ACK ve vyslaném rámci. Jelikož se však nejedná o explicitní potvrzení speciálním rámcem, jsou takto potvrzované zprávy v dalším textu označovány jako nepotvrzovaná komunikace.

²Zpráva RTR je vlastně potvrzena příslušným rámcem, o který stanice, která RTR vyslala, žádá.



Obrázek 2.1: Adresování objektů CANopen

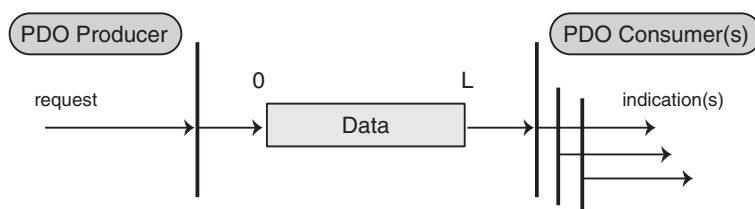
indexem a subindexem, delší zprávy jsou fragmentovány, takže maximální délka vyměňovaných dat není omezena. Tento typ přenosu vyžaduje dodatečnou režii protokolu.

Počáteční nastavení a spuštění celého distribuovaného systému provádí správa sítě – *Network Management NMT*. Zprávy CAN obsahují 11-bitový identifikátor, který mimo jiné určuje také prioritu zprávy (přístup ke sběrnici CAN je založen na rozpoznávání priorit zpráv při případné kolizi a na zrušení zprávy s nižší prioritou). Na CANopen je tento identifikátor označován *COB ID* – *Communication Object ID* a příslušný COB ID je přidělován jednotlivým zprávám PDO nebo SDO v adresáři objektů. Přitom platí, že čím nižší je COB ID, tím vyšší je priorita zprávy.

2.1 Komunikační model CANopen

Jak již bylo uvedeno, CANopen vychází z aplikační vrstvy CAL (CAN Application Layer), definované v dokumentu CIA-200. Existují zde dva typy komunikačních mechanismů:

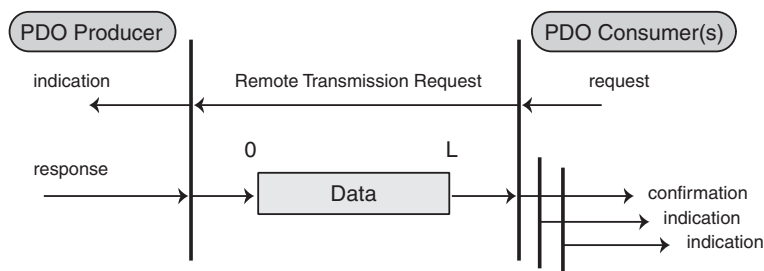
- nepotvrzovaný přenos datových rámců s maximální délkou dat 8 bytů je určen k přenosu procesních dat bez dodatečné režie (*PDO*).
- potvrzovaný přenos dat mezi dvěma stanicemi s přímým přístupem k položkám adresáře objektů (*SDO*). Tento typ přenosu je určen především k výměně dat mezi konfiguračním nástrojem a zařízením CANopen nebo k přenosu většího množství dat mezi stanicemi, přičemž jsou původní data fragmentována tak, aby byla dodržena maximální délka dat v rámci 8 bytů.



Obrázek 2.2: Přenos dat PDO – zápis

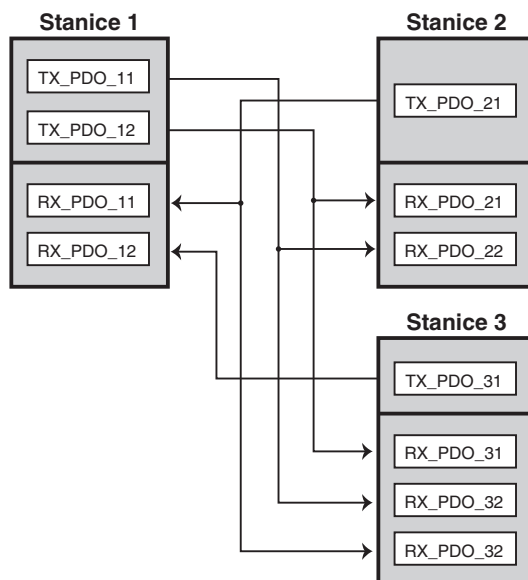
Přenos procesních dat PDO tedy probíhá nepotvrzovanými zprávami podle modelu znázorněného na obrázcích 2.2 a 2.3. Každému PDO je přiřazen jeden jednoznačný identifikátor CAN, který určuje význam

přenášených dat. Navíc je třeba provést tzv. „mapování PDO“ – viz dále. Obrázek 2.2 ukazuje zápis procesních dat, jinými slovy přenos dat od zařízení *PDO-producer* – *producent* k jednomu nebo více zařízením *PDO-consumer* – *konzument*. Producentem dat je typicky zařízení – čidlo, které hodnoty svých vstupů přenáší na sběrnici jako tzv. *Tx-data*. Konzument pak tato data přijme jako tzv. *Rx-data*. Obrázek 2.3 zobrazuje mechanismus vyslání žádosti o data konzumentem prostřednictvím rámce RTR – žádost o data. Producent následně požadovaná data vyšle a číst je potom mohou i ostatní konzumenti, tj. nejenom ten, který žádost o data vyslal. Rovněž je vidět, že producent data vysílá jako broadcast, jak je definováno v protokolu CAN. Na obrázku 2.4 je znázorněn příklad struktury vzájemné komunikace prostřednictvím



Obrázek 2.3: Přenos dat PDO na vyžádání – čtení

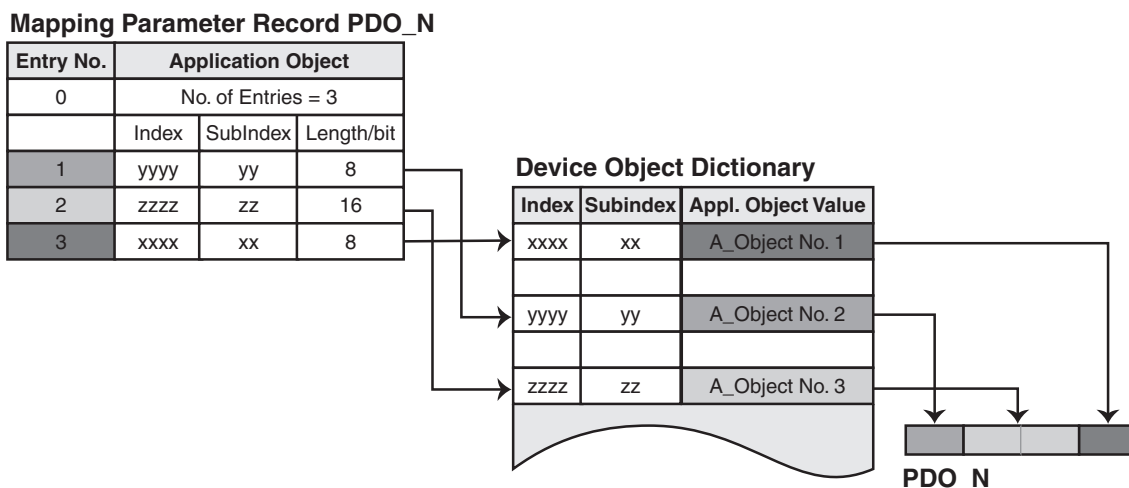
dat PDO. Znovu je třeba si uvědomit, že data označená jako TX_PDO jsou data vstupní, tj. vstupní data daného zařízení vysílaná na sběrnici, a data označená jako RX_PDO jsou data výstupní, tj. výstupní data daného zařízení přijímaná ze sběrnice a zapsaná na výstup.



Obrázek 2.4: Příklad vzájemné komunikace prostřednictvím PDO

2.1.1 Mapování aplikačních objektů na PDO

Skutečná funkčnost poskytovaná zařízením CANopen je definována aplikačními objekty, což mohou být parametry, data a funkce. Například se může jednat o hodnotu digitálního vstupu nebo výstupu, cílová rychlost servopohonu, stav zařízení, příkaz k vypnutí zařízení, atd. Vzhledem k tomu, že lze přenášet maximálně 8 bytů dat v jednom PDO, je třeba definovat umístění aplikačních dat v datovém poli PDO. Tomuto popisu se říká *mapování PDO* a jeho princip je znázorněn na obrázku 2.5. Zde je znázorněn



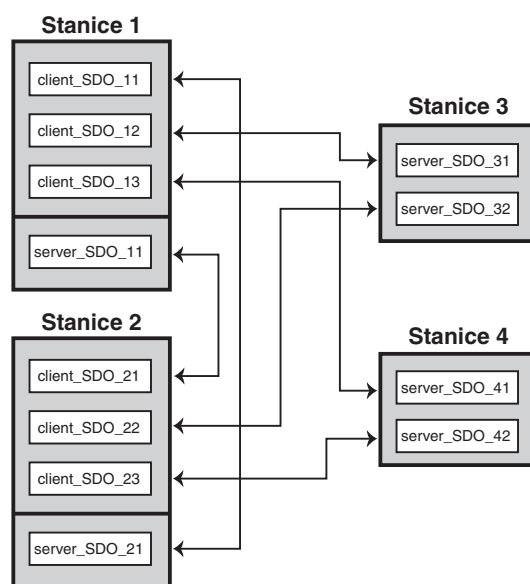
Obrázek 2.5: Princip mapování PDO

seznam aplikačních objektů v adresáři objektů, které mají být mapovány do jednoho rámce PDO. Každý aplikační objekt je jednoznačně určen svým Indexem a Subindexem – v adresáři objektů je na daném místě hodnota aplikačního objektu (např. hodnota 8-bitového digitálního vstupu). V tabulce v levé části obrázku je pak samotné mapování do rámce PDO. Vidíme, že v jednom rámci budou přenášeny hodnoty tří aplikačních objektů, jak je vidět v řádku 0 (Entry No. = 0). Každý řádek pak obsahuje odkaz na příslušný aplikační objekt do adresáře objektů, tj. Index a Subindex daného aplikačního objektu. Navíc je uvedena i délka aplikačního objektu, tj. kolik bytů objekt zabere v rámci PDO. Podle obrázku tedy bude do rámce PDO_N nejprve umístěn aplikační objekt 2, který zabírá 2 byty, potom aplikační objekt 3, který zabírá 2 byty, a nakonec aplikační objekt 1, který zabírá 1 byte.

Příklad konkrétního mapování tak, jak to odpovídá standardnímu profilu vstupů a výstupů implementovanému na zařízení WAGO, bude uveden v části 2.4.

2.1.2 Přenos dat SDO

Zde se jedná o potvrzovaný přenos dat libovolné délky mezi stanicemi prostřednictvím služebních objektů (*Service Data Objects – SDO*) podle modelu *client/server*. Klient, což je stanice, která přenos začíná, tak má přímý přístup k adresáři objektů serveru SDO, díky čemuž může posílat/přijímat data libovolné délky do/z serveru prostřednictvím příslušného Indexu a Subindexu. Pro každou zprávu v příslušném směru (klient → server nebo server → klient) je třeba jeden identifikátor zprávy CAN, a tak pro spojení mezi klientem a serverem jsou třeba celkem dva identifikátory CAN (konkrétní řešení je uvedeno v komunikačním profilu v části 2.3. Takové spojení je také označováno jako *kanál SDO*. Příklad komunikace prostřednictvím kanálů SDO je uveden na obrázku 2.6. Z něj je vidět, že spojení je vytvořeno vždy mezi dvěma účastníky (model klient/server) a že stanice 1 a 2 mají prostřednictvím svých klientů přístup k



Obrázek 2.6: Komunikace prostřednictvím kanálů SDO

adresáři objektů všech ostatních stanic. CANopen umožňuje, aby zařízení spravovalo až 128 kanálů SDO, každé zařízení však musí obsahovat alespoň jeden objekt SDO server (*default SDO*), díky němuž může k dané stanici přistoupit libovolné další zařízení, například konfigurační nástroj. Konfigurace zařízení takovým nástrojem při uvádění systému do provozu je také nejdůležitějším příkladem použití komunikace SDO.

Přenos dat libovolné délky je možný díky fragmentaci původních dat, protože maximální délka dat v jednom rámci CAN je 8 bytů. Data jsou tedy přenášena jako sekvence segmentů. Tomuto přenosu však musí předcházet fáze navázání spojení, ve které je v první zprávě ze strany klienta uvedeno, o jaký typ přenosu se jedná, tj. jaká služba CANopen bude využívána. Například, je-li třeba přenášet pouze 4 byty, může se celý přenos odehrát pouze prostřednictvím jedné datové výměny, kdy spolu s žádostí o vytvoření spojení budou přeneseny požadované 4 byty dat.

2.2 Adresář objektů CANopen

Význam pojmu *objekt* již byl zmíněn dříve a vyplývá z obrázku 2.1. *Adresář objektů (Object Dictionary OD)* pak obsahuje objekty, které jsou nezbytné pro zobrazení všech vlastností a parametrů zařízení, pokud tyto vlastnosti a parametry mají být přístupné přes sběrnici CAN. Jestliže objekt na určitém indexu obsahuje několik dalších položek označených příslušnými subindexy, pak je hodnota nejvyššího využitého subindexu uložena v položce se subindexem 0. Adresář objektů je uveden v tabulce 2.1, ze které se budeme v dalším textu věnovat především oblastem daným indexy 0x1000 až 0x1FFF, což jsou komunikační parametry popsané v komunikačním profilu, a 0x6000 až 0x9FFF, což jsou profily zařízení obsahující parametry, data a funkce dostupné přes CANopen.

Index	Object
0001 – 001F	Standard data types
0020 – 003F	Pre-defined complex data types
0040 – 005F	Manufacturer specific complex data types
0060 – 007F	Device profile specific standard data types
0080 – 009F	Device profile specific complex data typex
00A0 – 025F	Device profile specific standard and complex data types for devices with multiple modules
026F – 0FFF	Reseved for extensions
1000 – 1FFF	Communication profile area
2000 – 5FFF	Manufacturer specific area
6000 – 9FFF	Standardized device profile area
A000 – FFFF	Reserved

Tabulka 2.1: Adresář objektů CANopen

2.3 Komunikační profil

Pro naše potřeby jsou důležité především údaje týkající se komunikačních parametrů serverů a klientů SDO, jichž může být maximálně 128, dále pak definice komunikačních parametrů maximálně 512 objektů PDO pro vysílání a příjem (Rx_PDO, Tx_PDO) a mapování aplikačních objektů do objektů PDO. Struktura komunikačního profilu se zdůrazněním uvedených údajů je v tabulce 2.2 a v tabulce 2.3, přičemž oblast indexů přiřazených komunikačnímu profilu je 0x1000 až 0x1FFF, jak již bylo uvedeno výše.

Index	Object Type	Description	Access
Server SDO Parameter			
1200	RECORD	1st Server SDO Parameter	ro
1201	RECORD	2nd Server SDO Parameter	rw
⋮	⋮	⋮	⋮
127F	RECORD	128th Server SDO Parameter	rw
Client SDO Parameter			
1280	RECORD	1st Client SDO Parameter	ro
1281	RECORD	2nd Client SDO Parameter	rw
⋮	⋮	⋮	⋮
12FF	RECORD	128th Client SDO Parameter	rw
1300		reserved	
⋮	⋮	⋮	⋮
1300		reserved	

Tabulka 2.2: Část komunikačního profilu – parametry serverů a klientů

Tabulka 2.2 obsahuje objekty SDO a jak již bylo zmíněno výše, tyto objekty se rozlišují podle SDO serveru a klienta. Klientský SDO je chápán jako iniciátor SDO přenos a má možnost přistupovat k položkám adresáře objektů serveru SDO. Všimněme si, že první parametr serveru SDO je určen pouze ke čtení. To je z toho důvodu, že je povinný, pevně definovaný a není možné jej měnit, protože se používá ke standardní komunikaci s konfiguračním nástrojem. Tento povinný server SDO je umístěn vždy na indexu 0x1200 a jeho struktura je rozepsána v tabulce 2.4.

Index	Object Type	Description	Access
Receive PDO Communication Parameter			
1400	RECORD	1st Receive PDO Parameter	rw
1401	RECORD	2nd Receive PDO Parameter	rw
⋮	⋮	⋮	⋮
15FF	RECORD	512th Receive PDO Parameter	rw
Receive PDO Mapping Parameter			
1600	RECORD	1st Receive PDO Mapping	rw
1601	RECORD	2nd Receive PDO Mapping	rw
⋮	⋮	⋮	⋮
17FF	RECORD	512th Receive PDO Mapping	rw
Transmit PDO Communication Parameter			
1800	RECORD	1st Transmit PDO Parameter	rw
1801	RECORD	2nd Transmit PDO Parameter	rw
⋮	⋮	⋮	⋮
19FF	RECORD	512th Transmit PDO Parameter	rw
Transmit PDO Mapping Parameter			
1A00	RECORD	1st Transmit PDO Mapping	rw
1A01	RECORD	2nd Transmit PDO Mapping	rw
⋮	⋮	⋮	⋮
1BFF	RECORD	512th Transmit PDO Mapping	rw

Tabulka 2.3: Část komunikačního profilu – parametry PDO

Subindex	Definition	Access	Value range	Default value
0	No. of entries	ro	Index 0x1200: 2 Index 0x1201 – 0x127F: 2 – 3	no
1	COB_ID Client → Server (Rx)	Index 0x1200: ro Index 0x1201 – 0x127F: rw	Unsigned 32	Index 0x1200: 0x600 + Remote_Node_ID Index 0x1201 – 0x127F: no
2	COB_ID Server → Client (Tx)	Index 0x1200: ro Index 0x1201 – 0x127F: rw	Unsigned 32	Index 0x1200: 0x580 + Remote_Node_ID Index 0x1201 – 0x127F: no
3	Remote_Node_ID SDO Client	rw	0x1 – 0x7F	no

Tabulka 2.4: Popis objektu SDO serveru

Subindex	Definition	Access	Value range	Default value
0	No. of entries	ro	3	3
1	COB_ID Client → Server (Tx)	rw	Unsigned 32	no
2	COB_ID Server → Client (Rx)	rw	Unsigned 32	no
3	Remote_Node_ID SDO Server	rw	0x1 – 0x7F	no

Tabulka 2.5: Popis objektu SDO klienta

2.4 Profily zařízení

Profily zařízení popisují skutečnou funkčnost pro funkce standardního profilu v rozmezí indexů 0x6000 – 0x9FFF a pro funkce specifikované určitým výrobcem v rozmezí indexů 0x2000 – 0x5fff. Standardní profily jsou vytvořeny pro základní typy zařízení jako například digitální vstupy, digitální výstupy, analogové vstupy, analogové výstupy, které jsou součástí dokumentu CiA-401. Další profily definují parametry a funkce pohonů, inkrementálních snímačů, programovatelných automatů atd. Pro nás je nejvýznamnější profil digitální vstupů a profil digitálních výstupů, které jsou uloženy v oblastech 0x6000 až 0x6128, respektive 0x6200 až 0x6328. Jelikož význam parametrů pro vstupy i výstupy je podobný, je zde uveden pouze popis digitálních vstupů, který odpovídá tabulce 1.5.