

COMMUNICATION PROTOCOL IEC 870-5-101

Jan MIKULKA, Bachelor Degree Programme (3)
Dept. of Control and Instrumentation, FEEC, VUT
E-mail: xmikul16@stud.feec.vutbr.cz

Supervised by: Dr. Miloslav Čejka

ABSTRACT

The subject of this work is summary information about accessible communications standards used in automation and measurement. These protocols differ from the others especially in exact definition of transported data types (e.g. integer values, real values, single-point values etc.). The Biggest part is devoted to detailed description of the standard IEC 870-5-101 which is used to control and regulation of electricity supply network. Practical contribution is dynamic link library (DLL) programmed in C++ language. IEC DLL is application interface between two communicating devices.

1 ÚVOD

Komunikační standard (protokol) je souhrn pravidel, kterými se řídí přenos informací mezi zařízeními přes dané rozhraní. V tomto případě se jedná o sériové rozhraní RS-232 pro dvoubodové spojení (s použitím nulového modemu) popř. propojení více zařízení pomocí modemu. Podle normy [1] spolu mohou dvě stanice komunikovat v širokém rozsahu datových toků, od nejmenších 100 bit/s až po 9600 bit/s, což v mnohých aplikacích, kde se vyskytují tzv. spontánní události, jistě stačí. Všechna data jsou zasílána v rámcích (*frames*), které obsahují jak uvozovací a ukončovací sekvence, tak i symboly pro zabezpečení dat.

Standard podle normy IEC 870-5-101 byl vyvinut pro dálkové řízení, dálkovou ochranu energetické soustavy a s ní spojené další komunikace. Přenos dat podle tohoto standardu dnes často používají přední firmy zabývající se řídicí, resp. měřicí technikou, jelikož je to robustní nástroj pro řízení a formátování přenášených telemetrických hodnot.

Praktickým přínosem této práce je realizace DLL knihovny v jazyce C++ Builder. Knihovna definuje služby druhé (rámce a komunikační procedury) a sedmé (informační elementy a aplikační služby) vrstvy protokolu, čímž vytváří jakékoliv aplikaci rozhraní mezi vzdáleným zařízením a vlastní aplikací formou objektů.

2 POPIS PŘENOSU

Komunikace mezi dvěma nebo více zařízeními může probíhat dvojitým způsobem. Prvním je tzv. *balanced mode*, kdy jsou si obě stanice rovny, co se týče oprávnění přístupu

k médiu. Z toho vyplývá, že řízená stanice (typickým představitelem v energetice je tzv. RTU – *Remote Terminal Unit*) může poslat data k řídicí stanici (SCADA systém, apod.), aniž by k tomu byla předem vyzvána. Druhým režimem je tzv. *unbalanced mode*, ve kterém jsou stanice rozděleny na *master* a *slave*. Podstata režimu spočívá v tom, že *slave* stanice posílá data vždy jen po přijetí žádosti od stanice *master*. Tento režim je nutné použít v případě propojení více stanic komunikujících po společném médiu. Je tím zaručeno, že nezačne vysílat více stanic najednou, čímž by došlo ke ztrátě dat.

Jak již bylo řečeno v úvodu, každý komunikační protokol určený pro automatizační a měřicí techniku musí specifikovat vše, co je úkolem každého protokolu, čili alespoň základní služby nejméně 3 vrstev podle ISO/OSI modelu a dále datové typy a komunikační procedury zabezpečující snadné předávání povelů (příkazů) od operátora k vlastním řídicím prvkům, resp. získávání informací z procesu.

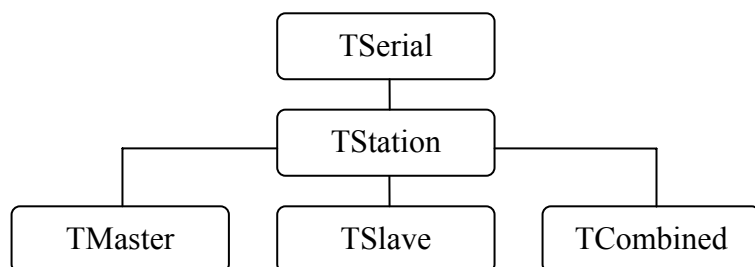
| Název | Information / Command | Význam |
|--|-----------------------|---|
| Single-point | I + C | dvoustavová hodnota (ON / OFF) |
| Double-point | I + C | třístavová hodnota |
| Step position | I | vícetavová hodnota (128 stavů) |
| Bitstring of 32 bit | I + C | bitově orientovaná hodnota (pole 32 bitů) |
| Measured value | I | měřená hodnota v pevné řádové čárce |
| Measured value, short floating point number (FP) | I | měřená hodnota v pohyblivé řádové čárce |
| Integrated totals | I | stav inkrementálního čítače |
| Regulating step | C | pro zvětšení / zmenšení žádané hodnoty |
| Set-point | C | stejně jako Step position |
| End of initialisation | I | informace o příčině vynucené inicializace |
| Interrogation | C | výzva k odeslání aktuálních hodnot |
| Clock synchronization | C | časová synchronizace mezi stanicemi |
| Test | C | konstantní testovací sekvence |
| Reset process | C | pro restart procesu nebo zrušení události |
| Parametr of meas. values | C | pro parametrizaci měření |
| Par. of meas. values, FP | C | pro parametrizaci měření |

Tab. 1: Normou definované datové typy pro řízení a monitorování

Přenášená data tedy můžeme rozdělit na dva základní typy – v řídicím směru (*control direction*) a v monitorovacím směru (*monitor direction*). V řídicím směru jde o tzv. příkazy (*command*) a v monitorovacím směru o informace (*information*). V tab.1 jsou uvedeny všechny definované datové typy, zda se jedná příkaz, resp. informaci nebo obojí a stručně vysvětlen jejich význam (podrobný popis v normě [1]).

3 NÁVRH SOFTWARE

Cílem je navrhnout hierarchii tříd knihovny tak, aby každá vytvořená instance byla schopná komunikace na nejnižším stupni (přes RS-232), ale nedovolila zasahovat do této úrovně uživatelskou vrstvou (aplikaci), čili aby aplikaci poskytovala pouze služby pro vlastní předávání informací, resp. příkazů. Diagram na obr. 1 popisuje takovou strukturu.



Třída TSerial zajišťuje komunikaci na nejnižší úrovni, čili naslouchání a vysílání dat po sériovém kanálu. Z ní odvozená je třída TStation, která těmito elementárními operacemi realizuje přenosy obecných rámců dat jejichž

Obr. 1: Hierarchie tříd knihovny

parametry jsou informační elementy. Podle ISO/OSI modelu je tato třída linkovou vrstvou protokolu. A konečně třídy TMaster a TSlave, které jsou určeny pro aplikace v *unbalanced* režimu a TCombined pro aplikace v *balanced* režimu. Tyto tři třídy jsou vlastně nejvyšší – aplikační vrstvou protokolu, tedy definují informační elementy a poskytují komunikační služby (testování komunikace, časová synchronizace, apod.).

V dané aplikaci se následně vytvoří instance objektu TMaster (jedná-li se o řídicí stanici), TSlave (jedná-li se o řízenou stanici) nebo TCombined (je-li zvolen *balanced* režim), která vždy zpřístupňuje jen nejvyšší komunikační služby (získání měřených hodnot ze vzdálené stanice, řízení procesu, apod.) nikoli však přístup k médiu na nejnižší úrovni (vyslání nebo příjem jednoho znaku apod.).

Jelikož každá procedura výměny dat zabere jistý čas, po který zatěžuje procesor, je nutné veškerou komunikaci provádět pomocí tzv. vláken (*threads*). Veškeré výměny dat mezi stanicemi budou probíhat paralelně ve volném procesorovém čase přiděleném uživatelské aplikaci a právě jen v tomto čase. Tím je zaručeno, že např. při odesílání rámce dat nebude v tomto čase uživatelské rozhraní (okno aplikace, apod.) neovladatelné.

4 ZÁVĚR

Podrobné nastudování normy z hlediska datových typů a komunikačních procedur bylo náplní semestrálního projektu. Tvorba software je věcí bakalářské práce a je tudíž ještě ve fázi vývoje. Prozatím je kompletně hotová a odladěná třída TSerial pro vlastní sériovou komunikaci. Do budoucna se nabízí implementace tohoto protokolu do mikrokontroléru pro možnost zavedení komunikačních služeb až do úrovně procesní instrumentace.

LITERATURA

- [1] IEC 870-5-101 : 2000. Rian, K. Norwegian User Conventions, v. 2.0
Dostupné na internetu URL http://www.statnett.no/Files/Open/IEC-R20_1.pdf