

BACnet

Protokol BACnet

[Podporované typy a verzie zariadení](#)
[Konfigurácia komunikačnej linky](#)
[Konfigurácia komunikačnej stanice](#)
[Konfigurácia meraných bodov](#)
[Scheduler v zariadeniach Siemens Desigo](#)
[Scheduler v zariadeniach Delta Controls](#)
[Informácie o eventoch](#)
[Informácie o alarmoch](#)
[Poznámka k cachovaniu adres](#)
[Poznámka k zariadeniam Delta Controls](#)
[Poznámka k zariadeniam E-DDC3.1](#)
[Poznámka k zariadeniam Klimasoft MBG-MSTP](#)
[Poznámka k zariadeniam Siemens Desigo](#)
[Poznámka k iLON 10 Ethernet adaptéru](#)
[Poznámka k implementácii BACnet MS/TP](#)
[Poznámka k podpore BBMD \(BACnet Broadcast Management Devices\)](#)
[Tell príkazy](#)
[Literatúra](#)
[Zmeny a úpravy](#)
[Revízie dokumentu](#)

Podporované typy a verzie zariadení

Protokol BACnet (Building Automation and Control Networks) je implementácia štandardu ANSI/ASHRAE 135-2001. Implementácia bola testovaná na nasledovných zariadeniach:

- Siemens
 - Desigo PXM20 (ovládacia jednotka, LON interface, BACnet over LON)
 - Desigo PXC22 (riadiaca stanica, LON interface, BACnet over LON)
 - Desigo PXC22-E.D (riadiaca stanica, Ethernet interface, BACnet/IP)
 - Desigo PXG80-N (BACnet router, Ethernet interface, LON interface, BACnet/IP, BACnet over LON)
 - Desigo PXC36-E.D (riadiaca stanica, Ethernet interface, BACnet/IP) - testovaná bola [asová synchronizácia](#)
- Delta Controls
 - [DSC-1212E](#) (System controller, Ethernet interface, BACnet/IP)
 - [DAC-633](#) (Application controller, MS/TP interface pripojené k DSC-1212E, ktorý funguje ako BACnet router)
 - [DAC-633](#) (Application controller, MS/TP interface pripojené k Moxa 5250 serial/ethernet prevodníku a nakomunikované priamo ako BACnet MS/TP zariadenie v UDP móde)
 - DAC-1146 - (Application controller, MS/TP interface pripojené k DSC-1212E, ktorý funguje ako BACnet router)
- Sauter
 - EYK220F001 (automatizovaná stanica, Ethernet interface, BACnet/IP)
 - EYR203F001 (univerzálny regulátor, pripojený k EYK220F001)
 - EYR207F001 (univerzálny regulátor, pripojený k EYK220F001)
- York
 - BACnet MS/TP MicroGateway (komunikačná karta pre chladie York, RS485 rozhranie, BACnet MS/TP)
- SE-Elektronik GmbH:
 - [E-DDC3.1](#) (DDC automatizovaná stanica, Ethernet interface, BACnet/IP)
- Klimasoft
 - [BACnet/Mbus prevodník MBG-MSTP](#): prevodník z BACnet na Mbus protokol(RS485 interface, BACnet MS/TP)

Vlastnosti aktuálnej verzie sú:

- komunikácia v sieach Ethernet (BACnet/IP) a LONTalk
- obmedzená podpora MS/TP siete (master-slave token-passing na RS-485): bez automatického vyhľadávania Master staníc
- podpora BACnet routra (prepojenie medzi BACnet/IP a LONTalk sieami)
- ítanie a zápis jednoduchých hodnôt (binárne, celoíselné, reálne, reakce, dátum, as at.) aj ubovoných ASN sekvencií
- podpora pollingového spôsobu ítania dát (správy ReadProperty-Request a ReadPropertyMultiple-Request)
- podpora zmenového spôsobu ítania dát (voliteľná registrácia pomocou správ SubscribeCOV-Request, resp. SubscribeCOVProperty-Request a následné spracovanie ConfirmedCOVNotification-Request a UnconfirmedCOVNotification-Request)
- zápis hodnôt správou WriteProperty-Request
- dynamická zmena adresy meraného bodu pomocou TELL príkazu [SETPTADDR](#) (na naítanie hodnôt objektov typu schedule)
- práca s objektmi typu schedule (asové plány)

Protokol BACnet sa pozerá na všetkých účastníkoch komunikácie ako na sievové zariadenia (network devices). Každé sievové zariadenie obsahuje aspoň jeden (väšinou práve jeden) objekt typu Device (jeho Object Identifier musí byť unikátny v celej sieti). Tento objekt obsahuje ďalšie objekty zadefinovaných typov (Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Calendar, Command, Event, Group, File at.) Zisovanie objektov - viď popis Request type [Who-Is](#) a [Who-Has](#).

Jednotlivé objekty majú vlastnosti (properties), pričom norma definuje pre jednotlivé typy objektov povinné a voliteľné vlastnosti a navyše každý výrobca BACnet zariadení môže implementovať ďalšie rozširujúce vlastnosti podľa potreby.

Správy protokolu BACnet sa týkajú manipulácie s objektmi a ich vlastnosťami. Správy sú definované pomocou ASN.1 (Abstract Syntax Notation version 1) a zakódované pomocou zjednodušenej verzie BER (Basic Encoding Rules - základné kódovanie ASN.1 správ).

Definícia správ obsahuje okrem pevne definovaných položiek aj položky typu 'Abstract Syntax & Notation', o znamená, že na danom mieste správy môže byť ubovoná sekvencia (resp. 'strom'), ktorej význam uruje implementátor. Použitie BER umožňuje parsovanie aj takejto správy s vopred neznámymi položkami. BER definuje dva základné typy položiek (tagov): aplikované a kontextové. Aplikované tagy sú preddefinované:

- **Null** - prázdna hodnota
- **Boolean** - hodnota typu áno/nie
- **Unsigned** - kladné celé číslo
- **Signed** - celé číslo
- **Real** - 4-bajtové reálne číslo
- **Double** - 8-bajtové reálne číslo
- **Octet String** - postupnosť znakov
- **Character String** - znaková sada + textový reazec
- **Bit String** - postupnosť bitov
- **Enumerated Value** - vymenovaný typ
- **Date** - dátum
- **Time** - as
- **Object Identifier** - identifikátor objektu (32-bitové číslo, skladá sa z 10-bitového čísla typu-Object Type a 22-bitového čísla inštalácie-Instance)

Kontextové tagy sú závislé na kontexte (na mieste v správe). Bez znalosti kontextu (popisu správy, ktorá sa parsuje) je možné zistiť, že na konkrétnom mieste sa nachádza kontextový tag, 5 s dĺžkou 4 bajty, ale je potrebná dodatočná informácia, či je to Unsigned, Signed, Real, Bitstring alebo iný typ hodnoty. Okrem jednoduchých aplikovaných a kontextových tagov môžu byť vlastnosti aj komplexné:

- **Sequence** - postupnosť, ktorá sa skladá z ďalších vlastností (jednoduchých aj komplexných), tieto sú buď povinné alebo voliteľné. Príklad:
- **Sequence of** - postupnosť N-tíc vlastností
- **Choice** - jedna z N možností

Príklad - výpis z trasy súboru KOM procesu so zapnutým debugom:

```
=== ASN Body beg ===
objectIdentifier (tag 0) OBJID 0 analog-input,10
listOfResults (tag 1) SEQUENCE {
  propertyIdentifier (tag 2) ENUM 85 present-value
  propertyValue (tag 4) SEQUENCE {
    ENUM 1
  }
}
=== ASN Body end ===
```

Interpretácia:

ide o sekvenciu (Sequence) dvoch tagov: tag objectIdentifier je kontextový s číslom 0, typu Object Identifier. Jeho hodnota je Object type=0 (analogový vstup), Instance=10. Tag listOfResults má kontextový tag 1 a je to sekvencia (Sequence of) dvoch tagov. Prvý tag je propertyIdentifier, je to kontextový tag . 2 typu Enum s hodnotou 85, ktorá zodpovedá vlastnosti 'present-value'. Druhý tag je kontextový s číslom 4 a je to sekvencia obsahujúca jeden tag typu Enumerated Value s hodnotou 1 (je to aplikovaný a nie kontextový tag).

Na parsovanie tejto správy musí poznať KOM proces ASN.1 definíciu správy, pretože bez nej by zistil, že sa v správe nachádza kontextový tag 2 (s hodnotou o dĺžke 1 bajt), ale nevedel by, že ide o Enumerated Value a teda nedokázal by bajt interpretovať (mohlo by ísť o Enumerated Value, Unsigned alebo Signed číslo), nevedel by, že tento kontextový tag má názov propertyIdentifier a hodnota 85 zodpovedá vlastnosti 'present-value'.

Vlastností (properties) objektov sú v konfigurácii D2000 mapované na merané body. Kvôli existencii kontextových tagov je na meranom bode možné špecifikovať [Application tag](#), ktorý hovorí, ako treba interpretovať daný kontextový tag. Aby bolo možné získať hodnoty z implementátorom definovanej sekvencie, obsahuje meraný bod položku [Complex address](#), ktorá udáva 'cestu' v parsovanom 'strome'.

Konfigurácia komunikanej linky

- Kategória komunikanej linky: [TCP/IP-UDP](#), [LonWorks](#), [Serial](#), [SerialOverUDP Device Redundant](#).
- Parametre linky TCP/IP-UDP:
 - Host: IP adresa sievového rozhrania, ktoré KOM proces používa na komunikáciu. Je možné zadať aj symbolické meno, ktoré sa dá previesť na IP adresu.
Pozn: Je možné zadať aj adresu **ALL** - v tom prípade sa používajú všetky dostupné rozhrania.
 - Port: číslo UDP portu, ktorý KOM proces používa na komunikáciu (podľa normy 0xBAC0, t.j. 47808).
Pozn: parametre záložného servera (Host a Port) nie sú v protokole použité

Parametre protokolu linky

Kúové slovo	Plný názov	Popis	Jednotka	Náhradná hodnota
DBGI	Debug Input	Rozšírené debug informácie o vstupných dátach. Význam jednotlivých bitov: <ul style="list-style-type: none"> 1. bit - debugovanie parsovania ASN správy 2. bit - debugovanie názvov meraných bodov, ktorým prišla nová hodnota ostatné bity - zatiaľ nepoužité 	-	0
DTQ	Debug Timeout Queue	Rozšírené debug informácie o správach v asovej fronte.	-	False
DI	Device Instance	Nenulová hodnota spôsobí, že KOM proces odpovedá na požiadavku Who-Is správou I-Am, v ktorej uvádza zadané Device Instance. Nulová hodnota spôsobí, že Who-Is požiadavky budú ignorované.	-	0
RB	Receive Buffer	(iba pre TCP/IP-UDP linku) Vekos prijímacieho buffra nastavovaná na UDP sockete. Hodnota 0 znamená, že sa vekos buffra nemení. Štandardná vekos na Windows XP je 8192 bajtov, pri vašom pote stanic, resp. intenzívnejšej komunikácii je vhodné buffer zväši.	bytes	0
RO	Receive Only	Ak je hodnota True, žiadnej stanici na linke sa neposielajú žiadne správy. Parameter je použitý napr. pri odposluchu komunikácie LonTalk: Na linke sa nakonfiguruje adresa zhodná s adresou existujúceho LonTalk zariadenia a nakonfiguruje sa stanica s adresou zariadenia, ktorého komunikáciu potrebujeme odpúva. V logovacom súbore linky sa bude nachádza zaznamenaná komunikácia medzi zariadeniami. RO=True zabezpečí, že KOM proces neovplyvní komunikáciu vlastnými príkazmi a odpoveami.	-	False
SC	Send Count	(iba pre LonWorks linku) počet opakovania jedného paketu - prednastavená hodnota je 1, ale v určitých situáciách pri použití iLON(tm)10 Ethernet Adapter-a prvá správa akoby neprešla a komunikácia zaala korektné fungova, ke sa nastavil SC=2. Poznámka: neskôr bolo zistené, že na vine bolo neukonenie Free topology zbernice terminátorom, ale parameter bol už implementovaný..	-	1
SD	Send Delay	(iba pre LonWorks linku) doplnok parametra SC, ktorý udáva oneskorenie (v ms) po každom poslaní paketu.	ms	0
VI	Vendor ID	Parameter Vendor ID správy I-Am (vi parameter Device Instance).	-	1

Parametre protokolu linky špecifické pre BACnet MS/TP

Kúové slovo	Plný názov	Popis	Jednotka	Náhradná hodnota
BR	MS/TP baud rate	Rýchlos linky. Tento parameter slúži na prepoet niektorých timeoutov, ktoré sú v parametroch protokolu linky zadávané kompatibilne s normou v bitových asoch , tj. v násobkoch doby, ktorú si pri konkrétnej nastavenej prenosovej rýchlosti vyžiada prenos 1 bitu.	bits/sec	9600
MIF	MS/TP N max_info_frames	Maximálne množstvo informaných rámcov, ktoré môže KOM proces vysla pred tým, ako musí odovzda token. Norma nešpecifikuje konkrétnu hodnotu, iba hovorí, že pokia táto hodnota v zariadení nie je konfigurovaná, musí by 1. ím väčšia hodnota je nastavená, tým menšie množstvo asu ostane pre ostatných Mastrov, ale na druhej strane sa znižuje množstvo rámcov bez informaného obsahu.	-	5
MO	MS/TP N min_octets	Minimálne množstvo dát (bajtov) prijatých na linke, ktoré musí prija KOM pred vyhlásením linky za "aktívnu".	-	4
MY	MS/TP my address	Adresa KOM procesu na linke RS-485. Platná hodnota je z intervalu 0 - 127. Adresa sa musí líši od adries ostatných zariadení na linke (ich adresy budú uvedené v konfiguráciách stanic).	-	1
TFA	T _{frame_abort}	Minimálny as (zadávaný v dčke vysielania bitov, t.j. závislý od parametra MS/TP baud rate), po ktorého vypršaní, bez prijatia ďalšieho znaku počas prijímania rámca, sa celý rámec zahodí. Podľa normy môžu implementácie používa aj väčšie hodnoty, ktoré neprekroia v absolútnom ase hodnotu 100 ms.	bits	60
TNT	T _{no_token}	as (zadávaný v ms) po ktorého vypršaní bez prijatia dát bude vyhlásená strata tokenu.	ms	500
TR	T _{reply_timeout}	Minimálny as (zadávaný v ms), ktorý musí KOM aka, kým stanica zane odpoveda na požiadavku.	ms	255
TS	T _{slot}	as (zadávaný v ms), počas ktorého môže stanica vygenerova token.	ms	10
TU	T _{usage_timeout}	Minimálny as (zadávaný ms), ktorý musí KOM aka, kým partner zane používa token alebo odpovie na <i>Poll for master</i> rámec. Štandardná hodnota je 20 ms, podľa normy môžu implementácie používa aj väčšie hodnoty, ktoré neprekroia 100 ms.	ms	20

Konfigurácia komunikanej stanice

Komunikaná stanica zodpovedá zariadeniu na BACnet sieti, s ktorým KOM proces komunikuje.

- **Typ stanice:** Stanica nakonfigurovaná na linke [TCP/IP-UDP](#) musí ma typ BACnet/IP, stanica nakonfigurovaná na linke [LonWorks](#) musí ma typ LonWorks. Stanica nakonfigurovaná na linkách [SerialOverUDP Device Redundant](#) alebo [Serial](#) musí ma typ MS/TP.

- **Adresa:**
 - Stanica BACnet/IP: IP adresa stanice (v tvare A.B.C.D, napr. 172.16.0.99)
 - Stanica LonWorks : adresa LON subsiete a LON uzla (v tvare subnet.node, kde subnet je 8-bitové číslo a node je 7-bitové číslo)
 - Stanica MS/TP: číslo nodu na linke (0-254, adresa 255 je broadcast)
- **Port:** (iba pre BACnet/IP): číslo UDP portu stanice (poda normy 0xBAC0, tj. 47808)
- **Doména:** (iba pre LonWorks): 0 alebo 1, súvisí s konfiguráciou linky. Na linke [LonWorks](#) je možné nakonfigurovať príslušnosť k jednej alebo dvom doménam, na stanici BACnet sa výberom domény udáva, do ktorej domény zariadenie patrí (výber ovplyvňuje 'domain' bit v LON adrese)
- **Source network:** číslo zdrojovej siete (tj. siete, do ktorej patrí KOM proces). Pre linku [LonWorks](#) sa štandardne nenastavuje, pre linku TCP/IP-UDP je to 16-bitové číslo (alebo sa nenastavuje, vi nižšie [Poznámka 2](#)).
- **Destination network:** 16-bitové číslo cieovej siete (tj. siete, do ktorej patrí zariadenie, s ktorým KOM proces komunikuje). Pre linku [LonWorks](#) sa nastavuje v prípade, že KOM proces komunikuje so zariadením, ktoré sa nachádza za BACnet routrom. V takom prípade [Adresa](#) stanice je adresa BACnet routra a [Destination address](#) je adresa cieového zariadenia. Pre linku [TCP/IP-UDP](#) sa parameter **Destination network** podobne použije iba v prípade komunikácie medzi rôznymi sieťami BACnet.

Poznámka 1: Táto konfigurácia bola otestovaná nasledovne:

- Linka: [TCP/IP-UDP](#)
- Typ stanice: [BACnet/IP](#)
- Adresa: [172.16.99.1](#) (adresa BACnet routra PXC80-N)
- Destination network: [1](#)
- Destination address: [1.1](#) (adresa PXC22 na LON sieti za BACnet routrom)

KOM proces komunikoval so zariadením PXC22 pripojeným k LON sieti prostredníctvom BACnet routra PXC80-N. Komunikácia medzi KOM procesom a BACnet routrom je po sieti Ethernet, preto je linka typu [TCP/IP-UDP](#). Komunikácia medzi BACnet routrom a stanicou PXC22 prebiehala po sieti LON.

Poznámka 2: Riešili sme podobnú konfiguráciu, kde bol použitý Delta Controls DSM-RTR (pripojený po Ethernet sieti) a za ním cez MS/TP rozhranie pripojené zariadenie Klimasoft MBG (gateway na M-Bus). V skúšanej konfigurácii sa komunikácia rozbehla, pokiaľ *nebola* nakonfigurovaná *Source network*, ale iba *Destination network* (konkr. hodnota 50020) a *Destination address* (konkr. 96). Prítom v inom prípade v podobnej konfigurácii komunikácia fungovala aj s parametrom *Source network*, takže treba vyskúšať a poexperimentovať, ktoré nastavenie sieťových parametrov, ktorému zariadeniu vyhovuje.

- **Destination address:** Adresa cieového zariadenia, pokiaľ s ním KOM komunikuje cez BACnet router. Pri zadaní tohto parametra je možné (ale nie nutné, vi poznámku o [E-DDC3.1](#)) zadať aj parameter [Destination network](#). Parameter **Destination address** je zadávaný v tvare subnet.node (ak cieové zariadenie je na LON sieti) alebo v tvare A.B.C.D (ak cieové zariadenie je na BACnet/IP sieti).

Poznámka 1: Na stanici typu BACnet/IP je možné nakonfigurovať **Destination address** v tvare subnet.node (napr. 1.31). Táto konfigurácia zodpovedá BACnet routru, ktorý s KOM procesom komunikuje cez BACnet/IP a k cieovému zariadeniu je pripojený LONTalk sieťou.

Poznámka 2: Na stanici typu BACnet/IP je možné nakonfigurovať **Destination address** ako číslo z intervalu 1-255. Táto konfigurácia zodpovedá BACnet routru, ktorý s KOM procesom komunikuje cez BACnet/IP a k cieovému zariadeniu je pripojený MS/TP zbernicou ([DA C-633](#)).

Poznámka 3: Na stanici typu BACnet/IP je možné nakonfigurovať **Destination address** ako väčšie číslo (napr. 2001), o fungovalo pre [E-DDC3.1](#).

- **Resubscribe interval:** čas v sekundách, po uplynutí ktorého sa znovu posielajú stanici žiadosti o posielanie meraných bodov. Tento parameter sa týka meraných bodov s [Request type](#) rovným [SubscribeCOV](#) alebo [SubscribeCOVProperty](#).
- **Max APDU:** Maximálna veľkosť správy (APDU=application protocol data unit), ktorú KOM proces posielajú. Prednastavená hodnota je:
 - 1467 oktétov pre linku [TCP/IP-UDP](#)
 - 487 oktétov pre linku [SerialOverUDP Device Redundant](#) alebo [Serial](#) (BACnet MS/TP)
 - 55 oktétov pre linku [LonWorks](#) ([LON 10 Ethernet adaptéra](#)) (obmedzenia sú dané veľkosťami paketov, ktoré sú schopné protokoly siete Ethernet a LonWorks prepravovať, v prípade LonWorks je maximálna hodnota 206 a hodnota 55 je kvôli obmedzeniam)

Menenie prednastavenej hodnoty má zmysel kvôli testovaniu a na prispôbenie sa stanicam, ktoré sú schopné spracovať iba menšie správy. V súčasnosti zmenšenie parametra *Max APDU* má vplyv iba na veľkosť a množstvo správ [ReadPropertyMultiple-Request](#). Tieto správy slúžia na periodické čítanie hodnoty meraného bodu (vi konfigurácia meraného bodu).

Poznámka: Nastavenie Max APDU nemá vplyv na veľkosť parametra max-APDU-length-accepted v APDU BACnet-Confirmed-Request-PDU, ktorým KOM proces oznamuje partnerovi, akú najväčšiu správu je schopný spracovať. Tento parameter je konfigurovaný pomocou parametra protokolu stanice [Segment-Response](#).

- **Priorita:** priorita správy v BACnet protokole. Existujú 4 priority, prednastavená je Normal, vyššie sú Urgent, CriticalEquipment a LifeSafety.
- **Rpt_timer & reply:** (iba pre LonWorks): parametre Repeat timer a Retry protokolu LonTalk. Prednastavené hodnoty sú 1 a 1.
- **Tx_timer:** (iba pre LonWorks): parameter Tx_timer protokolu LonTalk. Prednastavená hodnota je 3.
- **Timeout a retry:** timeout v milisekundách na potvrdenie správy. Prednastavená hodnota je podľa protokolu BACnet 3000 ms. Po vypršaní timeoutu sa správa posielajú opäť a to až **retry**-krát. Ak nie je prijaté žiadne potvrdenie, zvýši sa počítadlo chýb na stanici.

Poznámka: Pri testovaní zariadenia Siemens PXC64-U (komunikácia cez LonTalk) bolo potrebné nastaviť Retry=8, Timeout=300 (viac opakovaní s kratším timeoutom), v dôsledku toho bolo treba zvýšiť aj hodnoty COM_ERR=10, HARD_ERR=20, aby pri opakovaní posielania správy neprechádzala stanica do chybového stavu.

- **COM_ERR**: hodnota počítadla chýb na stanici, pri ktorej prechádza stanica do stavu COM_ERR. Za chybu sa považuje, ke stanica neodpovie na výzvu ítania alebo zápisu hodnoty. Chybou nie je negatívne potvrdenie príkazu (odmietnutie zápisu). Prednastavená hodnota je 5. Vi popis parametrov *Timeout a retry*.
- **HARD_ERR**: hodnota počítadla chýb na stanici, pri ktorej prechádza stanica do stavu HARD_ERR. Prednastavená hodnota je 10. Vi popis parametrov *Timeout a retry*.
- **Register-Foreign-Device, R-F-D Time to live**: majme stanice nachádzajúce sa na LONTalk sieti za BACnet routrom, ktorý komunikuje s KOM procesom v sieti Ethernet (napr. Desigo PXG80-N). BACnet router preposiela broadcasty zo siete LONTalk na Ethernet sie ako UDP broadcasty. Pokia nie je povolené šírenie UDP broadcastov alebo sa nachádza KOM proces na inom segmente siete ako BACnet router (takže UDP broadcasty sa k nemu nedostanú), je vhodné zaškrtnú na stanici vobu *Register-Foreign-Device*. Táto spôsobí, že po štarte pošle KOM proces routru BVLC (BACnet Virtual Link Control) správu Register-Foreign-Device. Správa žiada o registráciu do FDT tabuky routra (Foreign Device Table). Zariadeniam, ktoré má router registrované v FDT tabuke, preposiela broadcasty vo forme UDP unicastov (ktorých šírenie nie je obmedzené na jeden segment). Parametrom správy Register-Foreign-Device je TTL - as v sekundách (1-65535), po ktorom registrácia vyprší a UDP unicasty sa prestanú posiela. KOM proces preto musí pred vypršaním TTL znovu požiada BACNet router o registráciu. Pokia sa nachádza za BACnet routrom niekoľko staníc, staí zaškrtnú vobu *Register-Foreign-Device* na jedinej stanici.

Poznámka 1: Ak router nepodporuje BBMD funkcionalitu (BACnet/IP Broadcast Management Device), odpovie na správu Register-Foreign-Device chybovým kódom a nebude preposiela LonTalk broadcasty KOM procesu vo forme UDP unicastov. V tom prípade je nutné použiť iné riešenie (komunikácia cez iLon Ethernet Adapter, umiestnenie KOM procesu na rovnaký segment siete ako BACnet router at).

Poznámka 2: Router Desigo PXG80-N funkcionalitu podporuje (odskúšané), riadiaca stanica Desigo PXC22-E.D ju údajne podporuje (zatia neodskúšané).

Poznámka 3: V prípade zariadení Desigo, ak je D2000 KOM proces na inom sieovom segmente ako Desigo zariadenie, tento parameter musí by na stanici zaškrtnutý. Bez toho nebudú fungova dotazy *Who-Is* a *Who-Has* (a teda ani adresácia menom objektu), keže odpovede na tieto dotazy sú posielané ako UDP broadcasty, ktoré neprejdú cez router.

- **Master:** (iba pre MS/TP): stanica je typu Master. KOM proces bude odovzdáva token stanici typu Master, ktorá má najbližšiu vašiu adresu ako je adresa KOM procesu (parameter linky *MS/TP address*). Ak všetky stanice typu Master majú nižšie adresy ako KOM proces, token bude odovzdaný stanici typu Master s najnižšou adresou. Ak nie je nakonfigurovaná žiadna stanica typu Master, KOM proces predpokladá, že je jediný master, a token neodovzdáva. Informáciu o tom, i je stanica typu Master, je potrebné získa od výrobcu alebo z dokumentácie zariadenia.

Poznámka: Aktuálna verzia protokolu neobsahuje implementáciu automatického vyhľadávania Master stanice. Viac informácií je možné nájs v [Poznámke k implementácii BACnet MS/TP](#).

- **Poznámka:** je možné zapnú [asovú synchronizáciu](#) BACnet stanice s komunikaným počítaom.

Príklad konfigurácie stanice na linke [TCP/IP-UDP](#):

- **Typ stanice:** BACnet/IP
- **Adresa:** 10.0.0.1
- **Port:** 47808
- **Source network:** 1

Príklad konfigurácie stanice na linke [LonWorks](#):

- **Typ stanice:** LonWorks
- **Adresa:** 1.15
- **Doména:** 0

Parametre protokolu stanice

Kúové slovo	Plný názov	Popis	Jednotka	Náhradná hodnota
RAS	Read After Subscribe	Parameter ovplyvuje merané body s Request type = SubscribeCOV alebo SubscribeCOVProperty. Nastavenie hodnoty na True spôsobí, že po výzve SubscribeCOV resp. SubscribeCOVProperty pošle KOM proces následne aj požiadavku na ítanie (ReadProperty-Request). Pozn: Štandardne parameter netreba nastavova, pretože ako odpove na SubscribeCOV resp. SubscribeCOVProperty vášina zariadení pošle okrem potvrdenia aj aktuálnu hodnotu. Parameter bol implementovaný kvôli komunikácii so Sapphire Communication Controller rev.077-0210, ktorý poslal iba potvrdenie, ale nie aktuálnu hodnotu.	-	False
RSD	Receive-send Delay	Oneskorenie medzi prijatím odpovede stanice a poslaním ďalšieho paketu.	ms	0
SR	Segment-Response	Bajt obsahujúci Max Segs a Max Resp parametre (vi špecifikáciu protokolu BACnet). Povolené sú iba niektoré hodnoty z intervalu 0-127, ktoré špecifikuje norma BACnet. Hodnotu 128 interpretuje KOM proces ako default: <ul style="list-style-type: none"> • LonWorks linka: hodnota sa nastaví na 0x70 (akceptovaných je viac ako 64 segmentov, max. dška správy 50 bajtov) • TCP/IP-UDP linka: hodnota sa nastaví na 0x75 (akceptovaných je viac ako 64 segmentov, max. dška správy 1476 bajtov) • Serial a SerialOverUDP Device Redundant linka: hodnota sa nastaví na 0x73 (akceptovaných je viac ako 64 segmentov, max. dška správy 480 bajtov) 	-	128

TSU	Time-Synchronizácia UTC	<p>Parameter má význam, iba ak je povolená synchronizácia v konfigurácii stanice na záložke "asové parametre".</p> <p>Ak je hodnota parametra True (default), asová synchronizácia je vykonávaná pomocou správy UTCTimeSynchronization-Request (synchronizácia v UTC ase). Ak je hodnota parametra False, asová synchronizácia je vykonávaná pomocou správy TimeSynchronization-Request (synchronizácia v lokálnom ase).</p> <p>Poznámky:</p> <ul style="list-style-type: none"> • Odporúčame synchronizáciu v UTC ase, pokiaľ ju zariadenie podporuje - je tak možné vyhnúť sa problémom s prechodom asov. • Požiadavky na asovú synchronizáciu sú nepotvrďované správou - t.j. od zariadenia nepríde odpoveď ani v prípade, ak podporuje asovú synchronizáciu, ani ke ju nepodporuje. • Asová synchronizácia bola otestovaná na zariadení Siemens PXC36-E.D (HW=V3.02). Toto zariadenie podporuje synchronizáciu cez UTC aj lokálny as. Je možné vyíť aktuálny as a dátum ako property local-date(56) a local-time(57) objektu typu Device(8). <p>Z tohto objektu je možné vyíť aj property utc-offset(119) udávajúci offset lokálneho asu od UTC (v minútach, t.j. -60 je stredo európsky pásmový as) ako aj property daylight-savings-status(24) udávajúci, či pracuje zariadenie na letnom ase (v septembri 2012 bola na testovanom zariadení hodnota True).</p> <p>Po asovej synchronizácii sa hodnoty local-date(56) a local-time(57) príslušne zmenili.</p>	-	True
-----	-------------------------	---	---	------

Konfigurácia meraných bodov

Možné typy hodnôt bodov: **Ai,Ci,Di,TiA,TiR,TxtI,Ao,Co,Dout,ToA,ToR,TxtO**.

Meraný bod zodpovedá vlastnosti objektu (object property).

- **Request type:** ítanie a zápis vlastností objektov je možný niekoľkými spôsobmi:
 - **ReadProperty** - periodické ítanie vlastností objektu systémom výzva-odpoveď, periódu pollingu je nastavená na stanici na záložke *asové parametre*. Na výzvu sa použije správa ReadProperty-Request, zariadenie ako odpoveď posiela správu ReadProperty-Ack. Periodické ítanie zaažuje sa, je neefektívne, preto pokiaľ zariadenie podporuje posielanie zmenových dát, odporúčame použiť SubscribeCOV alebo SubscribeCOVProperty. Správa ReadProperty-Request je posielaná iba vtedy, ak je zaškrtnutý checkbox [Subscribe/read](#).
 - **ReadPropertyMultiple** - podobné ako predchádzajúce, na rozdiel od ReadProperty sa v jednej výzve aj odpovedi posiela niekoľko vlastností, takže komunikácia je efektívnejšia. Na výzvu sa použije správa ReadPropertyMultiple-Request, zariadenie ako odpoveď posiela správu ReadPropertyMultiple-Ack. Správa ReadPropertyMultiple-Request je posielaná iba vtedy, ak je zaškrtnutý checkbox [Subscribe/read](#).
 - **WriteProperty** - zapisovanie hodnôt správou WriteProperty-Request. Je nutné špecifikovať aj parameter [Application tag](#). Ak je zaškrtnutý [Subscribe/read](#), po zápise sa spätne íta zapísaná hodnota správou ReadProperty-Request.
 - **SubscribeCOV** - ítanie hodnoty objektu zmenovým spôsobom. Ak je zaškrtnutý checkbox [Subscribe/read](#), po štarte KOM pošle správu SubscribeCOV-Request, ktorou oznámi zariadeniu, že chce byť informovaný o zmenách hodnoty objektu. Je možné špecifikovať, či zariadenie má posielať potvrďované (správa ConfirmedCOVNotification-Request) alebo nepotvrďované (UnconfirmedCOVNotification-Request) notifikácie. Potvrďovaná notifikácia je správa, ktorá vyžaduje explicitné potvrdenie KOM-u správou BACnet-SimpleACK-PDU, takže zaažuje sa potvrdeniami, ale pravdepodobnosť, že sa notifikácia stratí, je podstatne menšia ako u nepotvrďovanej (ak zariadenie nedostane potvrdenie, správu opakuje).

Poznámka 1: Okrem dynamickej registrácie správou SubscribeCOV-Request môžu niektoré zariadenia podporovať statickú registráciu (uloženú v konfigurácii), takže nie je potrebné sa registrovať a checkbox [Subscribe/read](#) môže byť odškrtnutý.

Poznámka 2: Registrácia môže byť posielaná v pravidelných intervaloch (napr. kvôli možnému výpadku napájania zariadenia). Interval posielania registrácie sa nastavuje na stanici ako parameter [Resubscribe interval](#).

- **SubscribeCOVProperty** - podobné ako *SubscribeCOV*, navyše je možné špecifikovať aj *Property identifier* (takže je možné sledovať aj zmeny iných vlastností objektov ako iba hodnoty) a prípadne Increment - vekos prírastku, ktorý spôsobí poslanie zmeny (t.j. pásmo necitlivosti). Posielaná správa je SubscribeCOVProperty-Request.

Poznámka: Testované zariadenie Siemens PXC64-U nepodporovalo parameter Increment.

- **Whols**-identifikovaná správa Who-Is-Request na zistenie, aký Device Object zariadenie obsahuje. Odpoveďou je správa I-Am-Request (obsahuje polia iAmDeviceIdentifier, maxAPDULengthAccepted, segmentationSupported, vendorID). Ak je meraný bod typu TxtI, všetky tieto informácie sú v textovej podobe extrahované do hodnoty meraného bodu. Ke takto identifikujem Device Object, môžem si nakonfigurovať meraný bod na načítanie vlastnosti *object-list* identifikovaného Device Object-u a ak zariadenie túto vlastnosť implementuje, vráti zoznam identifikátorov všetkých objektov, ktoré obsahuje. Následne je možné získať vlastnosti týchto objektov (object-name, location, description, present-value ..)

Poznámka: Pre zariadenie Siemens PXC64-U je nutné ítať vlastnosť *object-lists* nastaveným Array index, pričom Array index=0 udáva počet prvkov poa, prístup k jednotlivým prvkom poa je cez Array index=1 až N.

- **WhoHas**-identifikovaná správa Who-Has-Request na zistenie mena objektu z [identifikátora objektu](#) alebo naopak. Odpoveďou je správa I-Have-Request (obsahuje polia deviceIdentifier, objectIdentifier a objectName). Správa Who-Has-Request sa posiela iba raz pri inicializácii meraného bodu (resp. po TELL príkaze [SETPTADDR](#)) a slúži na prevod medzi menami a identifikátormi objektov. Správa Who-Has-Request bude obsahovať buď meno alebo identifikátor objektu podľa toho, či je na meranom bode nakonfigurovaný [Address type](#) ako *Name* alebo *Object type+Instance*. Podľa toho, či je zaškrtnutý [Subscribe/read](#), môže sa použiť informácia z [cache](#), o je podstatne rýchlejšie ako získavanie z komunikácie.
- **ReadWriteScheduler**: na výzvu sa použije správa ReadProperty-Request, na zápis WriteProperty-Request, pričom pri zápise sa zapisuje N dvojíc as-hodnota. Konfigurácia sa používa na ítanie a zápis objektov typu *schedule*, podrobnejší popis vi [Scheduler v zariadeniach Siemens Desigo](#).
- **GetEventInformation**: zistenie zoznamu objektov, ktoré sú v alarmovom alebo chybovom stave alebo potrebujú kvitovanie, podrobnejší popis vi [Informácie o eventoch](#).
- **AcknowledgeAlarm**: kvitovanie alarmov, ktorých zoznam bol načítaný requestom GetEventInformation. Podrobnejší popis vi [Informácie o alarmoch](#). Meraný bod musí byť textový výstup (TxtO).
- **Address type**: Každý objekt v protokole BACnet je adresovaný cez [Identifikátor objektu](#). Pri návrhu aplikácie v systéme Desigo sú objekty reprezentované názvami, pričom adresa objektu nie je prístupná a môže sa meniť v dôsledku zmien aplikácie. Zariadenia Delta Controls majú naopak objekty, ktorých adresy zadáva tvorca aplikácie. Z tohto dôvodu sú možné dva spôsoby zadávania adresy meraného bodu zodpovedajúce dvom *Address type*.

- **Name:** zadá sa meno objektu. Typ objektu a číslo inštancie sa zistia dynamicky z komunikácie. Kvôli nezahľovaniu komunikovaných liniek pri štarte KOM procesu sa využíva ukladanie údajov do [BACnet cache](#).
- **Object type + Instance:** zadá sa typ objektu a číslo inštancie. Vhodné pre BACnet objekty s konštantnými adresami.
- **Object type:** typ objektu, ktorého vlastnosti chcem íta/zapísať. Je možné použiť preddefinované typy alebo zapísať číslo nového typu objektu, ktoré si zadefinoval výrobca zariadenia. Typ objektu je 10-bitové číslo.
- **Instance:** poradové číslo objektu v rámci typu objektu. Každý objekt má v rámci zariadenia unikátny **Object Identifier**, o je dvojica [Object type, Instance]
- **Object Name:** názov objektu, keď Address type= Name, t.j. keď adresa meraného bodu sa zisťuje dynamicky z komunikácie. Object Name musí byť zadaný presne, t.j. netolerujú sa zbytočné medzery na začiatku ani na konci a aj malé a veľké písmená sa musia zhodovať s názvom objektu, ktorý je uložený v zariadení, s ktorým sa komunikuje.
- **Property type:** typ vlastnosti - pre *Simple* sa zadáva iba *PropertyIdentifier*, pre *Complex* treba zadať aj *Complex address*. Komplexný typ vlastnosti je potrebný pri parsovaní implementátorom definovaných rozšírení štandardných správ (Abstract Syntax & Notation). Pri posielaní správ ReadProperty-Request, ReadPropertyMultiple-Request, SubscribeCOV-Request, SubscribeCOVProperty-Request je *Complex address* ignorovaná.
- **Property identifier:** identifikátor vlastnosti objektu. Je možné použiť preddefinované vlastnosti alebo zapísať číslo novej vlastnosti, ktoré si zadefinoval výrobca zariadenia. Identifikátor vlastnosti je typu Enumerated Value, vlastnosti 0-511 sú rezervované pre BACnet, čísla 512-4194303 sú použité pre výrobcov zariadení.
- **Array index:** niektoré vlastnosti môžu byť definované ako polia hodnôt, v tom prípade je možné íta alebo zapísať konkrétnu položku poa.
- **Application tag:** nutne špecifikovať pri zápise hodnoty (Request type=WriteProperty) a prípadne aj pre iné typy žiadostí, pokiaľ parsovaná odpoveď obsahuje kontextové tagy, ktorých aplikovaný typ nie je známy, pretože ide o implementátorom definované rozšírenie správ. Výnimkou je výstupný textový bod, ktorý sa pri nešpecifikovanom aplikovanom tagu chápe ako 'AnyTree' a slúži na zápis ubovenej užívateľom zadanej ASN sekvencie.

Poznámka: Pokiaľ je zapisovaná hodnota *Invalid*, nezapíše sa ako definovaný *Application tag*, ale ako *Application tag "Null"*.

- **Complex address:** adresa tagu v 'strome' v prípade implementátorom definovaných rozšírení správ.

Príklad adresy: *[1],[3].2.1*

Popis:

[1]- kontextový tag .1 (predpokladá sa, že je to sekvencia),
[3]- v rámci tejto sekvencie kontextový tag .2 (opäť musí byť sekvencia),
 2- v rámci tejto sekvencie druhý tag v poradí (opäť sekvencia),
 1- v rámci tejto sekvencie prvý tag v poradí.

Adresa v 'strome' sa začína od úrovne propertyValue (vi príklady správ nižšie).

Najjednoduchší spôsob zobrazenia parsovanej správy je zapnutie debugu na linke a sledovanie výpisov na konzole alebo v logu linky.

Príklad 1: Majme zariadenie, ktoré obsahuje objekt typu 2 (Analog Value) s číslom inštancie 1 a predpokladajme, že zariadenie posiela ako hodnotu objektu trojicu čísel, z ktorých prvá je aktuálna hodnota, druhá minútový priemer a tretia desaminútový priemer. Výpis parsovanej správy môže byť nasledovný:

```

=== ASN Body beg ===
objectIdentifier (tag 0) OBJID 2 analog-value,1
listOfResults (tag 1) SEQUENCE {
  propertyIdentifier (tag 2) ENUM 85 present-value
  propertyValue (tag 4) SEQUENCE {
    REAL 1.40000E+00
    REAL 1.10000E+00
    REAL 1.30000E+00
  }
}
=== ASN Body end ===

```

Ak máme záujem o všetky tri hodnoty, nakonfigurujem 3 merané body, (Object type=analog_value, Instance=1, Property-identifier=present-value, Property-type=complex), ktoré sa budú líši komplexnou adresou (pre prvý bod 1, pre druhý 2, pre tretí 3). Iba jeden meraný bod by mal mať zaškrtnutý checkbox [Subscribe/read](#), pretože odpoveďou na jednu žiadosť je správa s tromi hodnotami. Pri posielaní správ ReadProperty-Request, ReadPropertyMultiple-Request, SubscribeCOV-Request, SubscribeCOVProperty-Request ani WriteProperty-Request sa komplexná adresa nepoužíva.

Poznámka: Ak by bol nakonfigurovaný meraný bod s Property-type=simple, jeho hodnota by sa po parsovaní správy nastavila na prvú nájdenú hodnotu (v príklade 1.40000E+00).

Príklad 2: Zariadenie Siemens Desigo PXC64-U má meraný bod (Object type=schedule, Instance=6, zaškrtnutý Subscribe-read, Property-identifier=weekly-schedule, Property-type=complex, Array index=1, Complex address=1). Na linke je zapnutý debug. Po uložení meraného bodu KOM proces pošle požiadavku a vypíše odpoveď:

```

=== ASN Body beg ===
objectIdentifier (tag 0) OBJID 17 schedule,6
propertyIdentifier (tag 1) ENUM 123 weekly-schedule
propertyArrayIndex (tag 2) UNSIGNED 1
propertyValue (tag 3) SEQUENCE {
  SEQUENCE {
    TIME 0:0:0.0
    UNSIGNED 2
    TIME 4:0:0.0
    UNSIGNED 3
    TIME 22:0:0.0
    UNSIGNED 1
  }
}
=== ASN Body end ===

```

V propertyValue sa nachádza sekvencia 6 hodnôt (striedavo as a kladné číslo). Ak chceme pristupovať k prvému asu, treba nastaviť Complex address=1.1, ak k prvému kladnému číslu, treba nastaviť Complex address=1.2, t.j. prvý element - sekvencia - a v rámci neho druhý element v poradí (UNSIGNED 2). Ak potrebujeme pristupovať k viacerým asom a/alebo hodnotám naraz, stačí nakonfigurovať niekoľko meraných bodov, z ktorých iba jeden bude mať zaškrtnuté Subscribe/read.

Poznámka 1: Hodnota meraného bodu po vytvorení a uložení s komplexnou adresou 1 zostane Unknown, pretože nakonfigurovaná komplexná adresa 1 zodpovedá 1. prvku v rámci propertyValue, o je sekvencia a nie jednoduchý zobrazený typ.

Poznámka 2: Meraný bod typu *Text* je schopný obsiahnuť nielen jednoduchú hodnotu ale aj ubovenú ASN sekvenciu. Jednotlivé hodnoty budú zapísané podľa pravidiel pre **zápis ASN sekvencie**. Ak v predchádzajúcom príklade nastavíme Complex address=1 a meraný bod zmeníme na textový vstup alebo výstup, jeho hodnota bude reazec " T0:0:0.0; u2; T4:0:0.0; u3; T22:0:0.0; u1; ". Ak bude Property-type=complex, ale Complex address nebude zadaná, výsledok bude " 0{ T0:0:0.0; u2; T4:0:0.0; u3; T22:0:0.0; u1; }".

- **Increment:** prírastok zmeny hodnoty vlastností objektu, ktorý spôsobí reportovanie zmeny (vi popis [SubscribeCOVProperty](#)).
- **Confirmed:** ak je checkbox zaškrtnutý, pre nakonfigurované Request typu [SubscribeCOV](#) a [SubscribeCOVProperty](#) špecifikuje, i zariadenie má posielať potvrdzované (správa ConfirmedCOVNotification-Request) alebo nepotvrdzované (UnconfirmedCOVNotification-Request) notifikácie.
- **Subscribe/read:** ak je checkbox zaškrtnutý, sú pre nakonfigurované Request typy posielané príslušné správy na ítanie/registráciu zmien hodnôt:
 ReadProperty: správa ReadProperty-Request
 ReadPropertyMultiple: správa ReadPropertyMultiple-Request
 SubscribeCOV: správa SubscribeCOV-Request
 SubscribeCOVProperty: správa SubscribeCOVProperty-Request
 ReadWriteScheduler: správa ReadProperty-Request

- **Period:** ak je zadaná nenulová hodnota a zároveň je zaškrtnutý checkbox [Subscribe/read](#), správy Subscribe/read sa nebudú posielať v intervale *Perioda pollingu* nakonfigurovanom na stanici, ale so zadanou periódou (v sekundách). Takto je možné nakonfigurovať na jednej stanici rôzne periódy polling pre rôzne objekty alebo pomocou jedného meraného bodu s Request type [ReadProperty](#) a s malou Period zísť, i stanica komunikuje.
- **Local time:** ak je checkbox zaškrtnutý, zapisované resp. ítané hodnoty asov a dátumov sa považujú za hodnoty v lokálnom ase, v opanom prípade ide o monotónny as UTC.
- **Flags-to-flags:** ak je checkbox zaškrtnutý, pre nakonfigurované Request typy [SubscribeCOV](#) a [SubscribeCOVProperty](#) špecifikuje, že okrem hodnoty meraného bodu sa nastavujú aj jeho užívateľské príznaky - flagy FA,FB,FC,FD, do ktorých sa mapuje hodnota status-flags (typu BACnetStatusFlags), pokiaľ sú posielané. BACnetStatusFlags je štvorica bitov (in-alarm, fault, overridden, out-of-service) poskytujúca dodatočné informácie o hodnote objektu.
- **Write priority:** pre zápis vlastností, ktoré norma definuje ako 'commandable' je možné špecifikovať prioritu 1-16, pričom priorita 1 je najvyššia a priorita 16 najnižšia. Nakonfigurovaná priorita 0 znamená, že sa žiadna priorita pri zápise nepoužije.

Prehľadovanie adresného priestoru

Po stlačení tlačidla "**Browse**" v záložke **Adresa** meraného bodu sa zobrazí dialóg **Bacnet Item Browser**. Dialógové okno umožňuje prehľadanie adresného priestoru zariadenia a vkladanie jeho prvkov do adresného dialógu meraného bodu.

Prehľadovací dialóg umožňuje filtrovanie prvkov podľa 4 základných kritérií:

- číslo inštancie
- typ
- meno objektu
- popis objektu

Poznámka: *Typ, meno a popis objektu* nie je potrebné zadať celé. Postavuje nasledujúci zápis ****FILTROVANÝ VÝRAZ****, kde hviezdiky reprezentujú ubovený text pred začiatkom a koncom výrazu.

Poznámka: Pomocou Ctrl+C je možné skopírovať obsah dialógu **Bacnet Item Browser** do clipboardu. Pokiaľ je vyznačený konkrétny riadok, skopíruje sa iba ten.

Poznámka: Vo verziách z 20.12.2018 a novších bolo implementované recyklovanie prehľadacieho dialógu. Pokiaľ je dialóg zavretý tlačidlom Cancel alebo po výbere objektu, v skutočnosti je iba skrytý a je k dispozícii pre browsovanie iného meraného bodu v rámci tej istej stanice, takže sa zachová štruktúra prehliadaných objektov. Kliknutie na krížik vpravo hore spôsobí skutočné zavretie dialógu.

Instance number	Type	Object name	Object description
55669	multi-state-output(14)	DHW	<undefined>
125203	binary-input(3)	DHW.DhwOverheat	<undefined>
124249	binary-input(3)	DHW.SensorInputs.DhwSensor.DhwSensorShift	<undefined>
97527	analog-input(0)	DHW.SensorInputs.DhwSensor.DhwSensor	<undefined>
87366	binary-input(3)	DHW.SensorInputs.DhwRtnSensor.DhwRtnSensorShift	<undefined>
87505	analog-input(0)	DHW.SensorInputs.DhwRtnSensor.DhwRtnSensor	<undefined>
113858	analog-input(0)	DHW.SensorInputs.DhwPressSensor	<undefined>
70525	analog-value(2)	DHW.DhwStptCalc.LegionelTemp	<undefined>
90211	analog-value(2)	DHW.DhwStptCalc.ComfTemp	<undefined>
123956	analog-value(2)	DHW.DhwStptCalc.EcoTemp	<undefined>
71431	analog-value(2)	DHW.DHW MaxTemp	<undefined>
104915	analog-value(2)	DHW.DHWPressXsEco	<undefined>
129386	analog-value(2)	DHW.DHWPressXsComf	<undefined>
66370	binary-input(3)	DHW.CircPumpError	<undefined>
98898	multi-state-value(19)	DHW.CircPump.CircPmpProgram	<undefined>
111900	multi-state-output(14)	DHW.CircPump.CirculPump	<undefined>
71431	binary-input(3)	DHW.DHW MaxTemp	<undefined>

99 available tag(s)

Refresh Cancel

Zápis ubovonej ASN sekvencie

Pomocou nakonfigurovaného meraného bodu typu textový výstup (TxtO) s nenastaveným aplikaným tagom je možné zapísať ubovonú ASN sekvenciu. Pravidlá zápisu sú nasledovné:

- prvok sa skladá z voliteľného ísla kontextového tagu, písmena urujúceho aplikovaný tag a hodnoty
- jednotlivé aplikované tagy sa zapisujú nasledovne (príklady ukazujú použitie bez a s kontextovým tagom):
 - Null: *[tag] n*, príklad: "n", "3n",
 - Boolean: *[tag] b [0/1/n/y/N/Y]*, príklad: "b0", "3b1"
 - Unsigned: *[tag] u hodnota*, príklad: "u 123", "10 u123"
 - Signed: *[tag] s hodnota*, príklad: "s-123", "10s 5"
 - Real: *[tag] r hodnota*, príklad: "r 1.23", "10r-3.14"
 - Double: *[tag] d hodnota*, príklad: "d 1.23", "10 d -3.14"
 - Octet string: *[tag] O string*, každý bajt stringu je zapísaný ako hexa číslo (bajtu 1 zodpovedá 01, bajtu 26 zodpovedá 1A), príklad: "O 1A33f0", "10 Obb004E"
 - Character string *[tag] C 'reazec'*, príklad: "C 'hello world'", "10C 'apostrofov' v reazci" reazec musí byť uzavretý v apostrofoch, ak sa vnútri reazca vyskytuje apostrof, musí byť zdvojený ako v druhom príklade. Prázdny reazec je možné zadať nasledovne: "C";
 - Bit string: *[tag] B bity*, príklad: "B 100101", "23B00101"
 - Enumerated value: *[tag] E hodnota*, príklad: "E 123", "10 E123"
 - Date: *[tag] D day.month.year[.day_of_week]*, príklad: "D 1.10.2005", "D3.4.2004.5" (pondelok=1 .. nedeľa=7)
 - Time: *[tag] T hour:minute:sec[.ms]*, príklad: "T 5:12:33.133", "T10:00:00"
 - Object identifier: *[tag] o typ.instance* alebo *[tag] o objid*, kde typ je 10-bitové číslo, instance je 22-bitové číslo, objid je 32-bitové číslo. Príklady ukazujú dvojzložkový a jednozložkový zápis toho aplikovaného tagu s type=3 (binary-input) a instance=2 "o 3:2", "3o3:2", "3o 12582914"
- sekvencia sa skladá z jednotlivých prvkov oddelených medzerami a/alebo bodkami, napr. "1b0 2u13 ; 3 B 1001;4E14"
- sekvencia môže obsahovať vnorené sekvencie
- vnorená sekvencia sa začína voliteľným číslom kontextového tagu a znakom '{'. Ak nie je zadaný kontextový tag, použije sa hodnota 0
- vnorená sekvencia sa končí znakom '}'
- príklad vnorenej sekvencie: "1u2 2{ 1s-1; 2E0 }", dve úrovne vnorenia: "{ 1{ u23 s34 } 2E56 3r7.89 }"

Poznámka 1: Ak je pri ítaní meraného bodu z komunikácie výsledkom ASN sekvencia a meraný bod je typu Txt, táto ASN sekvencia je zapísaná (poda hore uvedených pravidiel) do meraného bodu, pričom platí:

- pred každým prvkom nasleduje medzera a za prvkom bodkočiarka (napr. "1E4; 2B111; 3u1;"),
- medzi číslom kontextového tagu a písmenom urujúcim aplikovaný tag nie je medzera,
- medzi písmenom urujúcim aplikovaný tag a hodnotou nie je medzera,
- pred každou vnorenou sekvenciou je kontextový tag (napr. "0{ 0o1:2; 1E4; }"),
- formát času je hh:mi:ss.mss, napr. "T11:01:02.000; 1T12:00:00.000; ",
- formát dátumu je dd.mm.yyyy, napr. "D25.01.2005; 3D01.01.2005;".

Poznámka 2: Zápis prázdnej sekvencie je možný reazcom " ", reazec s dĺžkou 0 (t.j. "") je pri zápise ignorovaný.

Príklady konfigurácií meraných bodov (zariadenie Siemens Desigo PXC64-U):

Analogový vstup:

- Request type: SubscribeCOV
- Object type: analog-input(0)
- Instance: 1
- Confirmed: Y
- Subscribe/read: Y
- Property type: Simple
- Property identifier: present-value(85)

Binárny vstup:

- Request type: SubscribeCOV
- Object type: binary-input(3)
- Instance: 1
- Confirmed: Y
- Subscribe/read: Y
- Property type: Simple
- Property identifier: present-value(85)

Binárna hodnota:

- Request type: ReadProperty
- Object type: binary-value(5)
- Instance: 8
- Subscribe/read: Y
- Property type: Simple
- Property identifier: present-value(85)
- Application tag: Enum

Poznámka: Ak sa nenastaví správny Application tag (pre binárnu hodnotu je to Enum), stanica Desigo bude pri pokuse o zápis vracia chybu 'invalid-data-type':

```
error-class ENUM 2 property
error-code ENUM 9 invalid-data-type
```

Pre zariadenie Siemens Desigo PXC64-U sú pri zápise potrebné nasledovné nastavenia Application tag:

- binary-value, binary-output: Enum
- multi-state-value, multi-state-output: Unsigned
- analog-value, analog-output: Real

Poznámka k vstupno-výstupným meraným bodom: je možné nakonfigurovať meraný bod, ktorý bude zároveň vstupný aj výstupný:

Meraný bod typu Ao - Analog výstup:

- Request type: ReadProperty
- Object type: analog-value(2)
- Instance: 45
- Subscribe/read: Y
- Property type: Simple
- Property identifier: present-value(85)
- Application tag: Real (treba kvôli zápisu hodnoty)

Pri zápise hodnoty sa použije správa WriteProperty-Request. Keže je zaškrtnuté Subscribe/read, po zápise sa späťne íta zapísaná hodnota. Keby bol meraný bod nakonfigurovaný s Request type: WriteProperty, jeho správanie by sa líšilo iba absenciou periodických ítaní hodnôt (pri štarte KOM procesu a počas jeho behu, priom perióda je nastavená na stanici na záložke *asové parametre*).

Poznámka k zariadeniam Siemens Desigo: merané body sú v riadiacom systéme Desigo reprezentované textovým názvom. Inštancia meraného bodu sa dá zistiť zo súboru DOTS00.DAT konfigurácie aplikácie - nachádza sa 24 bajtov pred začiatkom názvu.

Scheduler v zariadeniach Siemens Desigo

D2000 podporuje ítanie a zápis schedulerov (asových plánov). Scheduler má implementované BACnet atribúty *weekly-schedule(123)* a *exception-schedule(38)*.

Weekly-schedule je pole 7 položiek (jedna položka pre každý deň týždeňa). Každá položka je postupnosť dvojíc as-hodnota, ktoré určujú zmeny hodnoty schedulera v danom ase. Pri ítaní aj zápise je možné nakonfigurovať aj *Array index*, a prístupovať tak k položkám podľa *weekly-schedule(123)*. Index podľa je 1-7 pre jednotlivé dni (1=pondelok), index 0 obsahuje rozmer podľa (hodnota 7 pre atribút *weekly-schedule(123)*), hodnota 0 až N pre *exception-schedule(38)*. Exception-schedule sa používa, keď je pre špeciálne dni (sviatky, prázdniny) potrebné nakonfigurovať iný režim ako pre bežný týždeň. Exception-schedule je postupnosť 0 až N položiek, pričom jedna položka vždy obsahuje dátum (alebo rozsah dátumov), niekoľko dvojíc as - hodnota u weekly-schedule a priority (najvyššia 1, najnižšia 16). Priority udáva, ktorá z položiek sa uplatní, ak sa v nejakom ase prekrývajú.

Ítanie schedulera (atribút weekly-schedule)

- Hodnotu po hodnote: dynamickou zmenou komplexnej adresy (1.1, 1.2, 1.3 at) v skripte je možné vyíta všetky hodnoty a asy podobne ako pre iné vlastnosti.
- Všetky asy a hodnoty pre jeden de naraz:
 - Typ hodnoty: Textový vstup (ítanie schedulera) alebo Textový výstup (ítanie aj zápis)
 - Request type: ReadProperty (ítanie schedulera) ReadWriteScheduler (ítanie aj zápis)
 - Subscribe/read: Y
 - Object type: schedule(17)
 - Instance: íslo inštancie (napr. 6) zistené z konfigurácie Desigo alebo pomocou Whols requestu.
 - Property type: Complex
 - Property identifier: weekly-schedule(123)
 - Array index: 1 až 7 poda naítavaného da
 - Application tag: ak nie je udaný, použije sa Unsigned (potrebný iba pri zápise hodnoty)
 - Complex address: 1 (adresa sekvencie)

Do textovej hodnoty sa naítajú asy a hodnoty oddelené bodkoiarkou (vi [Poznámka](#)).

Pri zápise hodnoty schedulera je potrebné si uvedomi, že hodnota môže by poslaná s rôznymi apliknými tagmi (Unsigned, Signed), priom zariadenie oakáva konkrétny tag (najjednoduchšie sa dá zisti naítaním hodnoty pri zapnutom debugu na linke). Aplikný tag hodnoty je konfigurane urený položkou Application tag v konfigurácii. Platné hodnoty Application tag sú Boolean, Unsigned, Signed, Real a Double. V prípade, že je nastavený iný typ, sa automaticky posíla Unsigned hodnota. Typ hodnoty je možné dynamicky zmeni - ak je prvý znak zapisovanej textovej hodnoty B,U,S,R alebo D, je chápaný ako (B)olean, (U)nsigned, (S)igned, (R)eal alebo (D)ouble.

Zápis schedulera (atribút weekly-schedule)

- Nutné je nakonfigurova Request type=ReadWriteScheduler a do meraného bodu typu Textový výstup priradi postupnos dvojic as a hodnota oddelených bodkoiarkou, napr. "0:0:0; 1; 2:30:40.5; 2; 5:00:00;1".
- Aby sa po uložení konfigurácie D2000 alebo po štarte KOM procesu nezapisovala do schedulera 'prázdna sekvencia', je textový reazec s džkou 0 ignorovaný. Preto, ak je potrebné vymaza asový plán schedulera pre konkrétny de, do meraného bodu je treba priradi reazec nenulovej džky, ktorý neobsahuje žiadny as ani hodnotu: " ".

Poznámka: Okrem špecializovaného requestu ReadWriteScheduler je možné zapísa weekly-schedule aj cez zápis [ASN sekvencie](#), napr. hodnote "0:0:0; 1; 2:30:40.5; 2; 5:00:00;1" z predchádzajúceho príkladu zodpovedá sekvencia "{ T0:0:0 u1; T2:30:40.5 u2; T5:0:0 u1 }" priom konfigurácia meraného bodu sa líši iba v Request type=ReadProperty. Navyše je možné zapísa asový plán na celý týžde, ak sa nenastaví Array index a hodnota obsahuje 7 sekvencií na jednotlivé dni, napr. "{ T0:10:0 u3 T1:3:0 u1; } { T2:0:0 u2 T5:30:10.0; u3; } { T6:0:0 u2 T7:0:0 u3 } { T20:0:0.33; u1 } { T21:0:0.0; u1 } { T22:0:0.0; u2 } { T0:0:0.0; u3; T1:2:0.0; u1; T2:0:0.0; u2; T5:30:10.0; u3}".

Zápis schedulera (atribút exception-schedule)

Zápis exception-schedule je podporený cez zápis ASN sekvencie. Majme meraný bod s nasledovnou konfiguráciou:

- Typ meraného bodu: textový výstup (TxtO)
- Request type: WriteProperty
- Subscribe/read: Y
- Object type: schedule(17)
- Instance: 6
- Property type: exception-schedule
- Application tag: nezadaný (zapiše sa celá [ASN sekvencia](#))

Ak pri zápise použijem reazec "0{ 0D2.10.2005 } 2{ T1:0:0; u1; T12:0:0; u3 } 3u10", tak pomocou neho zapíšem asový plán na 2.10.2005, priom od 1:0:0 bude ma scheduler hodnotu 1, od 12:00:00 hodnotu 3. Priorita exception-schedule je 10. Pri spätnom naítaní hodnoty (je nakonfigurované Subscribe /Read) so zapnutým debugom na linke je vidie nasledovný výpis:

```
=== ASN Body beg ===
objectIdentifier (tag 0) OBJID 17 schedule,6
propertyIdentifier (tag 1) ENUM 38 exception-schedule
propertyValue (tag 3) SEQUENCE 3 {
  readResult (tag 0) SEQUENCE 0 {
    date (tag 0) DATE 2.10.2005 NoDay
  }
  listOfTimeValues (tag 2) SEQUENCE 2 {
    time TIME 1:0:0.0
    value UNSIGNED 1
    time TIME 12:0:0.0
    value UNSIGNED 3
  }
  eventPriority (tag 3) UNSIGNED 10
}
=== ASN Body end ===
```

Ak chceme nastavi exception-schedule pre rozsah dátumov, zapíšeme hodnotu "0{ 1{ D5.10.2005; D8.10.2005} } 2{ T1:0:0; u1; T7:0:0; u3; } 3u15" Táto hodnota pre rozmedzie dátumov 5.10.2005-8.10.2005 nastavuje scheduler od asu 1:00:00 na hodnotu 1 a od asu 7:00:00 na hodnotu 3. Priorita exception-schedule je 15. Pri spätnom naítaní hodnoty (je nakonfigurované Subscribe/Read) so zapnutým debugom na linke je vidie nasledovný výpis:

```

=== ASN Body beg ===
objectIdentifier (tag 0) OBJID 17 schedule,6
propertyIdentifier (tag 1) ENUM 38 exception-schedule
propertyValue (tag 3) SEQUENCE 3 {
  readResult (tag 0) SEQUENCE 0 {
    dateRange (tag 1) SEQUENCE 1 {
      startDate DATE 5.10.2005 NoDay
      endDate DATE 8.10.2005 NoDay
    }
  }
  listOfTimeValues (tag 2) SEQUENCE 2 {
    time TIME 1:0:0.0
    value UNSIGNED 1
    time TIME 7:0:0.0
    value UNSIGNED 3
  }
  eventPriority (tag 3) UNSIGNED 15
}
=== ASN Body end ===

```

Je možné zapísať do exception-schedule niekoľko položiek hore uvedených typov zapísaním reazca, ktorý obsahuje pospájané sekvencie, napr. "0{ 0D2.10.2005 } 2{ T1:0:0; u1; T12:0:0; u3 } 3u10 0{ 0D3.10.2005 } 2{ T0:0:0; u2; T5:0:0; u3; T14:0:0; u1 } 3u11 0{ 1{ D5.10.2005; D8.10.2005}} 2{ T1:0:0; u1; T7:0:0; u3; } 3u15 "

Scheduler v zariadeniach Delta Controls

Podobne ako pre [scheduler v zariadeniach Siemens Desigo](#) bolo otestované ítanie a zápis do schedulera (asového plánu) v zariadení Delta Controls DAC-1146. Otestovaný bol iba BACnet atribút *weekly-schedule(123)*.

Weekly-schedule je pole 7 položiek (jedna položka pre každý deň týždeňa). Každá položka je postupnosť dvojíc as-hodnota, ktoré určujú zmeny hodnoty schedulera v danom ase. Na rozdiel od zariadení Siemens Desigo sú asové plány implementované s Boolean hodnotami True/False, ktoré sú prezentované navonok ako Enum hodnoty 0/1. Príklad asového plánu Delta Controls:

```
"{ T0:10:0 E0 T1:3:0 E1; } {T2:0:0 E1 T5:30:10.0; E0; } { T6:0:0 E0 T7:0:0 E1 } { T20:0:0.33; E0 } { T21:0:0.0; E1 } { T22:0:0.0; E0 } 0{ T0:0:0.0; E0; T1:2:0.0; E1; }"
```

Prejde aj zápis a následné ítanie hodnoty E2, akurát nevieme, ako to DAC-1146 následne interpretuje :-)

Informácie o eventoch

Pomocou requestu [GetEventInformation-Request](#) je možné vyžiadať si od zariadenia zoznam objektov, ktoré sú v stave *Offnormal*/alebo *Fault* alebo ktorých prechod do *Offnormal*, *Fault* alebo *Normal* stavu nebol kvitovaný.

Príklad konfigurácie meraného bodu:

- Typ meraného bodu: textový vstup (Ttxtl)
- Request type: GetEventInformation
- Subscribe/read: Y
- Object type: nezadaný
- Instance: nezadané
- Property type: complex
- Application tag: nezadaný

Odpoveou na GetEventInformation-Request je správa GetEventInformation-Ack, ktorá obsahuje zoznam objektov a pre každý objekt zoznam vlastností:

- objectIdentifier: identifikátor objektu
- event-state: stavu objektu (0=normal, 1=fault, 2=offnormal, 3=high-limit, 4=low-limit, 5=life-safety-alarm)
- acknowledgedTransitions: 3 bity udávajúce, či boli posledné prechody do stavu offnormal, fault a normal kvitované
- eventTimeStamps: asové znaky posledných prechodov do stavu offnormal, fault a normal
- notifyType: udáva, či je to notifikácia alarmu (0) alebo eventu (1)
- eventEnable: 3 bity udávajúce, či sa reportujú eventy to-offnormal, to-fault, to-normal
- eventPriorities: 3 unsigned hodnoty udávajúce priority eventov

Na konci zoznamu sa nachádza atribút moreEvents, ktorý je nenulový, ak zoznam eventov nie je kompletný (prekroenie dŕky maximálnej správy a podobne). Vtedy je nutné prekonfigurovať meraný bod a nastaviť Object type a Instance na posledný objekt v zozname, o spôsobi vygenerovanie novej správy GetEventInformation-Ack, ktorá bude ďalšiu as zoznamu eventov.

Príklad odpovede GetEventInformation-Ack:

```

=== ASN Body beg ===
listOfEventSummaries (tag 0) SEQUENCE 0 {
  objectIdentifier (tag 0) OBJID 0 analog-input,1
  event-state (tag 1) ENUM 3 high-limit
  acknowledgedTransitions (tag 2) BITSTRING 1,1,1 to-offnormal(0),to-fault(1),to-normal(2)
  eventTimeStamps (tag 3) SEQUENCE 3 {
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 9.11.2005 Wed
      time TIME 13:28:49.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 9.11.2005 Wed
      time TIME 13:25:59.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 9.11.2005 Wed
      time TIME 13:25:56.0
    }
  }
  notifyType (tag 4) ENUM 0 alarm
  eventEnable (tag 5) BITSTRING 1,1,1 to-offnormal(0),to-fault(1),to-normal(2)
  eventPriorities (tag 6) SEQUENCE 6 {
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 7
  }
  objectIdentifier (tag 0) OBJID 214,1
  event-state (tag 1) ENUM 2 offnormal
  acknowledgedTransitions (tag 2) BITSTRING 0,1,0 to-offnormal(0),to-fault(1),to-normal(2)
  eventTimeStamps (tag 3) SEQUENCE 3 {
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 18.11.2005 Fri
      time TIME 12:52:11.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 18.11.2005 Fri
      time TIME 11:16:23.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 9.11.2005 Wed
      time TIME 12:23:58.0
    }
  }
  notifyType (tag 4) ENUM 0 alarm
  eventEnable (tag 5) BITSTRING 0,0,0 to-offnormal(0),to-fault(1),to-normal(2)
  eventPriorities (tag 6) SEQUENCE 6 {
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 7
  }
}
moreEvents (tag 1) BOOLEAN FALSE
=== ASN Body end ===

```

Informácie o alarmoch

Možné je kvitovať eventy a alarmy, ktorých zoznam bol načítaný pomocou requestu [GetEventInformation](#). Kvitovací request má podľa BACnet protokolu nasledovný formát (zápis v ASN.1 syntaxi):

```

***** Confirmed Alarm and Event Services *****
AcknowledgeAlarm-Request ::= SEQUENCE {
  acknowledgingProcessIdentifier [0] Unsigned32,
  eventObjectIdentifier [1] BACnetObjectIdentifier,
  eventStateAcknowledged [2] BACnetEventState,
  timeStamp [3] BACnetTimeStamp,
  acknowledgmentSource [4] CharacterString,
  timeOfAcknowledgment [5] BACnetTimeStamp
}

```

Podrobnejší popis jednotlivých parametrov vi [literatúra](#).

Kvitovanie prebieha zápisom hore uvedenej sekvencie do textového výstupného meraného bodu podľa štandardu [zápisu ubovonej ASN sekvencie](#).

Príklad: Pomocou requestu [GetEventInformation](#) načítam nasledovný zoznam alarmov a eventov:

```

=== ASN Body beg ===
listOfEventSummaries (tag 0) SEQUENCE 0 {
  objectIdentifier (tag 0) OBJID 0 analog-input,1
  event-state (tag 1) ENUM 4 low-limit
  acknowledgedTransitions (tag 2) BITSTRING 0,1,1 to-offnormal(0),to-fault(1),to-normal(2)
  eventTimeStamps (tag 3) SEQUENCE 3 {
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 11:54:23.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 10:4:37.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 11:54:23.0
    }
  }
  notifyType (tag 4) ENUM 0 alarm
  eventEnable (tag 5) BITSTRING 1,1,1 to-offnormal(0),to-fault(1),to-normal(2)
  eventPriorities (tag 6) SEQUENCE 6 {
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 7
  }
  objectIdentifier (tag 0) OBJID 0 analog-input,2
  event-state (tag 1) ENUM 3 high-limit
  acknowledgedTransitions (tag 2) BITSTRING 1,1,1 to-offnormal(0),to-fault(1),to-normal(2)
  eventTimeStamps (tag 3) SEQUENCE 3 {
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 10:41:59.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 10:12:20.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 10:12:21.0
    }
  }
  notifyType (tag 4) ENUM 0 alarm
  eventEnable (tag 5) BITSTRING 1,1,1 to-offnormal(0),to-fault(1),to-normal(2)
  eventPriorities (tag 6) SEQUENCE 6 {
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 7
  }
  objectIdentifier (tag 0) OBJID 214,1
  event-state (tag 1) ENUM 2 offnormal
  acknowledgedTransitions (tag 2) BITSTRING 0,1,1 to-offnormal(0),to-fault(1),to-normal(2)
  eventTimeStamps (tag 3) SEQUENCE 3 {
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 11:54:23.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 10:12:20.0
    }
    dateTime (tag 2) SEQUENCE 2 {
      date DATE 25.11.2005 Fri
      time TIME 10:12:21.0
    }
  }
  notifyType (tag 4) ENUM 0 alarm
  eventEnable (tag 5) BITSTRING 0,0,0 to-offnormal(0),to-fault(1),to-normal(2)
  eventPriorities (tag 6) SEQUENCE 6 {
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 3
    eventPriority UNSIGNED 7
  }
}
moreEvents (tag 1) BOOLEAN FALSE
=== ASN Body end ===

```

Prvý v zozname je objekt

objectIdentifier (tag 0) OBJID 0 analog-input, 1

ktorý je v stave low-limit

event-state (tag 1) ENUM 4 low-limit

a má nekvitovaný prechod do stavu offnormal (t.j. v terminológii D2000 ide o aktívny alarm *low-limit* vyžadujúci kvitovanie):

acknowledgedTransitions (tag 2) BITSTRING 0,1,1 to-offnormal(0),to-fault(1),to-normal(2)

Tento prechod nastal v ase

dateTime (tag 2) SEQUENCE 2 {

date DATE 25.11.2005 Fri

time TIME 11:54:23.0

}

Alarm budeme kvitovať zápisom textovej hodnoty

0u1; 1o0:1; 2E2; 3{ 2{ D25.11.2005 T11:54:23 } 4C'D2000' 5{ 2{ D25.11.2005 T11:55:23 } }

priom jednotlivé položky sú (vi definíciu AcknowledgeAlarm-Request vyššie):

- *0u1* - tag 0, unsigned hodnota 1 - acknowledgingProcessIdentifier = identifikátor procesu, ktorý kvituje alarm (zrejme môže byť ubovoný)
- *1o0:1* - tag 1, identifikátor objektu typ 0, inštancia 1 - eventObjectIdentifier = identifikátor objektu, na ktorom je alarm
- *2E2* - tag 2, enum hodnota 2 - eventStateAcknowledged = potvrdzovaná udalosť (norma BACnet definuje nasledovné udalosti, ktoré sa dajú kvitovať:
 - normal (0),
 - fault (1),
 - offnormal (2),
 - high-limit (3),
 - low-limit (4),
 - life-safety-alarm (5)v tomto prípade potvrdzujeme prechod do stavu offnormal
- *3{ 2{ D25.11.2005 T11:54:23 } }* - timeStamp = sekvencia dátumu a asu, ktorý potvrdzujeme (musí byť zhodný s dátumom a asom naítaným vyššie)
- *4C'D2000'* - acknowledgmentSource = zdroj potvrdenia alarmu (zrejme ubovoný reazec)
- *5{ 2{ D25.11.2005 T11:55:23 } }* - timeOfAcknowledgment = dátum a as potvrdenia alarmu

Po kvitovaní alarmu a následnom naítaní zoznamu alarmov a eventov pomocou requestu [GetEventInformation](#) bude vidieť, že alarm bol kvitovaný, pretože sa zmení:

acknowledgedTransitions (tag 2) BITSTRING 0,1,1 to-offnormal(0),to-fault(1),to-normal(2)

na

acknowledgedTransitions (tag 2) BITSTRING 1,1,1 to-offnormal(0),to-fault(1),to-normal(2)

Pokiaľ by bol objekt v stave normal, tak sa v zozname alarmov neobjaví vôbec, ale v tomto prípade bol v stave *low-limit*, takže ide o aktívny kvitovaný alarm.

Na naítanie zoznamu alarmov, ich výpis do browsera a kvitovanie alarmu je potrebné napísať obslužné skripty na parsovanie textovej hodnoty requestu [GetEventInformation](#) a skladanie textovej hodnoty pre [AcknowledgeAlarm](#).

Pozn: v konkrétnom prípade pri potvrdzovaní alarmu vyžadovalo zariadenie DESIGO PXC 100E.D, aby v rámci dátumu bol zadaný aj deň v týždni (napr. *D8 .6.2020.1* pre pondelok). Hodnota 255 (unspecified) dáva v týždni, ktorá sa posiela, keď deň nie je zadaný, zariadeniu pri potvrdzovaní alarmu nevyhovovala.

Poznámka k cachovaniu adries

Ak má stanica aspoň jeden meraný bod s [Address Type = Name](#), pri inicializácii takýchto meraných bodov sa zisťuje íselná adresa z [ObjectName](#) pomocou správy Who-Has-Request. Po získaní adresy sa uloží štvorica (ObjectName; Object Type; Instance; DeviceInstance) do cache súboru a je k dispozícii pri ašom spustení KOM procesu.

Cache súbor existuje pre každú stanicu jeden. Nachádza sa v adresári aplikácie v podadresári Cache a jeho názov je *Cache_nazov_stanice.txt*, napr. *Cache_B.StationX1.txt*.

Pri uložení stanice BACnet dôjde k opätovnému naítaniu údajov zo súboru. Pokiaľ súbor nebol nájdený, dôjde k vygenerovaniu Who-Has správy na všetky merané body tejto stanice, ktoré majú [Address Type = Name](#).

Toto chovanie je možné využiť po zmene konfigurácie zariadenia, s ktorým KOM proces komunikuje (ak pri zmene konfigurácie došlo k zmene adries objektov). Stačí vymazať cache súbor tejto stanice a uložiť stanicu - po niekoľkých sekundách cache súbor znovu vznikne a naplní sa získanými údajmi.

Poznámka: Interakcia cache a meraných bodov s Request type = [WhoHas](#).

- Ak takýto bod má zaškrtnuté [Subscribe/read](#), cache sa neprehadáva a do komunikácie sa posiela správa Who-Has-Request. Získaná informácia sa nezapisuje do cache. Má zmysel iba pre dynamicky vznikajúce a zanikajúce objekty, ktoré by inak preplnili cache.
- Ak takýto bod nemá zaškrtnuté [Subscribe/read](#), prehadáva sa cache. V prípade, že sa hadané ObjectName alebo ObjectIdentifier nenašlo, do komunikácie sa posiela správa Who-Has-Request. Ak príde odpoveď, získaná informácia sa zapisuje do cache.

Poznámka k zariadeniam Delta Controls

Testovacia konfigurácia obsahovala zariadenie DSC-1212E pripojené k lokálnej sieti Ethernet a zariadenie DAC-633 pripojené k DSC-1212E cez MS/TP rozhranie (RS-485). D2000 KOM komunikoval priamo s DSC-1212E a cez DSC-1212E aj s DAC-633 (DSC-1212E plnilo funkciu BACnet routra).

Pri testovaní komunikácie bol použitý softvér ORCAview 3.30 Build 1481, z ktorého boli vyítané nasledovné konfigurované informácie:

- **DSC-1212E**

Po prihlásení sa do ORCAview a nájdení zariadenia DSC-1212E sa v pravom paneli okna Navigator nachádza objekt *BACnet Settings 3200* (íslo 3200 je softvérová adresa konkrétneho použitého zariadenia). Po poklikaní na objekt *BACnet Settings 3200* sa otvorí okno, v ktorom sa na prvej záložke (*Setup*) nachádza okrem iných aj port typu UDP/IP. Po kliknutí na sa v spodnej polovici okna ukážu

parametre UDP/IP portu, medzi inými *Network* (pri testovaní konkrétne číslo 40032), *IP Address* a *UDP Port*. Pri komunikácii s DSC-1212E bolo možné v D2000 v konfigurácii stanice nastaviť *Source network* na hodnotu parametra *Network* alebo ho nenastavovať vôbec. Parametre *IP Address* a *UDP Port* je nutné použiť na nastavenie parametrov *Adresa* a *Port* v konfigurácii stanice v D2000. Typ stanice je *BACnet/IP*

Poznámka: Na záložke *Advanced* je v oblasti BACnet Properties nastavené *Local Network Number* ako 10032 (rovnaké ako číslo *Network* portu typu *Ethernet* na záložke *Setup*). Tento port sa používa na komunikáciu BACnet over Ethernet (bez použitia IP protokolu), ktorú D2000 zatiaľ nepodporuje.

• DAC-633

Po prihlásení sa do ORCAView a nájdení zariadenia DAC-633 sa v pravom paneli okna **Navigator** nachádza objekt *BACnet Settings 3202* (číslo 3202 je softvérová adresa konkrétneho použitého zariadenia). Po kliknutí na objekt *BACnet Settings 3202* sa otvorí okno, v ktorom sa na prvej záložke (*Setup*) nachádza okrem iných aj port typu MS/TP s atribútom *Status* s hodnotou *Active* a s atribútom *Status Reference* s hodnotou *BACnet Settings 3202 (NET1)*. Po kliknutí na sa v spodnej polovici okna ukážu parametre MS/TP portu, medzi inými *Network* (pri testovaní konkrétne číslo 50032) a *MAC Address*.

Pri komunikácii s DAC-633 bolo nutné nastaviť nasledovné parametre stanice v D2000:

- Typ stanice: BACnet/IP
- Adresa: položka *IP Address* konfigurácie DSC-1212E
- Port: položka *UDP Port* konfigurácie DSC-1212E
- Source network: položka *Network* konfigurácie DSC-1212E
- Destination network: položka *Network* MS/TP portu konfigurácie DAC-633
- Destination address: položka *MAC Address* MS/TP portu konfigurácie DAC-633

Keže D2000 komunikuje s DAC-633 cez DSC-1212E, zadávajú sa do konfigurácie stanice parametre *Adresa*, *Port* a *Source network*, ktoré zodpovedajú DSC-1212E a parametre *Destination network* a *Destination address*, ktoré zodpovedajú MS/TP sieti medzi DSC-1212E a DAC-633.

Obidve zariadenia podporujú pollingový aj zmenový spôsob ítania dát (ReadProperty aj SubscribeCOV).

Parametre *Object type* a *Instance* v konfigurácii meraného bodu sú zistené z ORCAView z atribútov *Object* a *Description* (napr. objekt 3200.AI12 je Analog Input, Instance 12; objekt 3202.PG3 je Program, Instance 3). Parameter *Address type* je *Object type+Instance* a ak nie je uvedené inak, *Request type* je *SubscribeCOV*.

Nakomunikované typy bodov:

- Analog-input
- Analog-output (Application tag: Real - možnosť zápisu)
- Analog-value (Application tag: Real - možnosť zápisu)
- Binary-input
- Binary-output (Application tag: Enum - možnosť zápisu)
- Binary-value (Application tag: Enum - možnosť zápisu)
- Calendar (Request type: ReadProperty, Property type: Complex, Property identifier: datelist(123), Application tag: Date, Complex address: 1,2,.. at. - postupné vyíťavanie z poa, možnosť zápisu celého kalendára pri nezadanej Complex address pomocou ASN sekvencie, napr. "0D1.9.2006; 0D3.10.2006; 0D8.9.2006")
- Event-enrollment
- Schedule (Request type: ReadProperty alebo WriteProperty - možnosť zápisu; Property identifier: weekly-schedule(123))
- Program (Request type: ReadProperty, Application tag: Boolean - možnosť zápisu - zastavenie a spustenie programu)
- Multi-state-value (Application tag: Unsigned - možnosť zápisu)
- Trend-log (cyklický buffer hodnôt ukladaných s nakonfigurovanou periódou)
 - Property identifier: 1074 - pole trendových časových znaiek v desiatkach milisekúnd od času nulovania dát
 - Property identifier: 1076 - pole bitstringov (atribúty hodnôt?)
 - Property identifier: 1077 - pole trendových hodnôt
 - Property identifier: 1105 - čas nulovania dát

Poznámka k zariadeniam E-DDC3.1

Komunikáciu s testovacou zostavou sa podarilo oživiť v nasledujúcej konfigurácii:

- Linka: *TCP/IP-UDP*
- Typ stanice: *BACnet/IP*
- Adresa: *10.0.6.23* (IP adresa E-DDC3.1)
- Port: *47808*
- Source network: *nezadané*
- Destination network: *nezadané*
- Destination address: *2001* (získané od BACnet OPC servera od dodávateľa, kde bol tento údaj "Device ID")

Bolo vyskúšané ReadProperty, SubscribeCOV, WriteProperty (na objekty typu Binary Output, Binary Value, Analog Value).

Nakomunikované typy bodov (testovacia konfigurácia iné typy neobsahovala):

- Analog-input
- Analog-value (Application tag: Real - možnosť zápisu)
- Binary-input
- Binary-output (Application tag: Enum - možnosť zápisu)
- Binary-value (Application tag: Enum - možnosť zápisu)

S pôvodným firmvérom zariadenie (2.01.05) nereagovalo na [Who-Is](#), po upgrade firmvéru na verziu 2.01.16 sa toto chovanie opravilo a korektne funguje aj WriteProperty.

Poznámka k zariadeniam Siemens Desigo

Poda dokumentu "DESIGO INSIGHT: Installation, setup and communication - Engineering guide" odporúané adresné nastavenia pre komunikáciu cez LonWorks sú:

- DomainID: 0x49
- SubnetID: 1
- NodeID:
 - 1..100 - adresy rezervované pre automatizované stanice (PX) a systémové zariadenia (BACnet route)
 - 101..120 - operátorské zariadenia a management stanice DESIGO INSIGHT
 - 121..127 - doasné operátorské zariadenia (napr. operátorská jednotka PXM20) a nástroje (DTS)

Poznámka k zariadeniam Klimasoft MBG-MSTP

Testovacia konfigurácia obsahovala Moxa NPort 5150 (prevodník z UDP na RS485) pripojený k prevodníku MBG-MSTP, ktorý komunikoval s meraom tepla Siemens UH50-A21R-SK06-G. Vyítavané boli analógové vstupy, prevodník MBG-MSTP podporoval iba ReadProperty. Komunikáciu s testovacou zostavou sa podarilo oživi v nasledujúcej konfigurácii:

- Linka: [SerialOverUDP Device Redundant](#) s nakonfigurovanou IP adresou a portom Moxa NPort 5150
- Parametre protokolu BACnet konfigurované na linke:
 - MS/TP address*: 6 (ubovlná adresa 1-254, ktorá nekoliduje s inými adresami na RS485 zbernici)
 - MS/TP N max_info_frames*: 20. Default hodnota je 5, jej zvýšenie nad počet meraných bodov spôsobí, že všetky merané body sa pri periodickom vyítavaní naítajú v jednom cykle, ktorý nie je prerušený odovzdaním tokenu. Hodnotu parametra neodporúame zvyšovať, ak sú na RS485 zbernici pripojené ďalšie zariadenia, ktorým by vadilo, že dlhšie nedostanú token.
 - MS/TP usage_timeout*: 99. Poda normy musí byť hodnota pod 100 ms. Default hodnota 20 ms spôsobovala problémy v komunikácii (MBG-MSTP nestihol zareagovať do vypršania timeoutu).
- Typ stanice: *MS-TP*
- Adresa: 1 (poda konfigurácie MBG-MSTP)
- Konfigurácia meraných bodov:
 - Request type*: ReadProperty
 - Object type*: analog-input(0)
 - Instance*: poda konfigurácie MBG-MSTP resp. zariadenia, ktoré je za ním

Poznámka k iLON 10 Ethernet adaptéru

Pri použití iLON 10 Ethernet adaptéra na komunikáciu si treba uvedomiť, že komunikovaný procesor tohto zariadenia (Neuron 3120) má prednastavené pomerne krátke buffre (network buffer je 66 bajtov), preto nebude prijímať dlhšie pakety. Prejaví sa to vo webovom rozhraní iLON 10, kde bude v záložke Status narastať číslo Missed Messages pri pokuse íta hodnotu (napr. vždy po uložení meraného bodu, pokiaľ má zaškrtnuté [Subscribe/read](#)). Pri testovaní sa tento problém prejavil, keď asový plán schedulera obsahoval viac ako 4 dvojice as, hodnota. Pri štyroch hodnotách bola celková dĺžka prijatej LON správy 63 bajtov, po pridaní ďalšej dvojice cez terminál PXM20 sa už asový plán naíta nepodarilo.

Pozor! Pomocou utility Nodeutil je možné zväši veľkosť prijímaného paketu zväšením network buffra a aplikovaného buffra, ale rovnako dobre je možné **nastavením nevhodných hodnôt zariadenie iLON 10 zníi** (Neuron 3120 prestal komunikovať s užívateľským procesorom).

Keď sa rozhodnete prestaviť veľkosť buffrov, postaí zmeniť sieťové a aplikované buffre z 66 bajtov na 88 bajtov (a zredukovať veľkosť buffrov), pretože KOM procesor v svojich paketoch oznamuje, že je schopný prijať 50-bajtovú ASDU (+ 3 bajty hlavičky BACnet + 16 bajtov hlavičky LON)

Adaptér PCLTA-10 ISA (postavený na komunikovanom ipe Neuron 3150) mal dostatočne dlhý buffre (255 bajtov).

Po zakúpení nového zariadenia iLON 10 bola úspešná konfigurácia vykonaná s nasledovnými parametrami:

- použitý konfigurovaný program: *Echelon Node Utility Release 1.82*
- nastavené veľkosti a veľkosti buffrov:

```
DEVICE:0> (B)uffer configuration
Node buffer configuration
Type                Count    Size    Bytes
Receive transaction    11      13      143
Transmit transaction    2       28      56
App buffer in          2       82      164
App buffer out         2       82      164
Net buffer in          5       82      410
Net buffer out         2       82      164
App buff out priority  1       82      82
Net buff out priority  1       82      82
==> Total bytes = 1265
```

Poznámka k implementácii BACnet MS/TP

Implementácia BACnet MS/TP v žiadnom prípade nie je úplná. Bola otestovaná iba na jednoduchšej konfigurácii (KOM oproti BACnet MS/TP MicroGateway firmy York). Komunikácia fungovala pri použití linky [Serial](#) (použitý RS485/RS232 prevodník) ako aj linky [SerialOverUDP Device Redundant](#) (použitá Moxa NPort rady 5xxx).

MicroGateway firmy York implementuje komunikáciu typu MS/TP Master (t.j. zisuje prítomnosť ďalších staníc typu Master vysielaním rámca *Poll for master*). KOM sa v súčasnosti spolieha na partnerskú stanicu a nemá implementované vysielanie tohto rámca. Rovnako zatiaľ dostatočne nerieši ani výpadok stanice, ktorej odovzdáva token. Súasná implementácia je vhodná iba na komunikáciu s jedinou stanicou typu Master, môže byť vhodná na komunikáciu s jednou alebo viacerými stanicami typu Slave (netestované). Takisto asové pomery môžu byť v zariadených systémoch problém - KOM nemusí stihnúť odpoveď vo vyžadovanom aso na rámci *Poll for master* alebo *Token*, čím môžu vzniknúť napr. kolízie na linke. asové pomery sa navyše zhoršia zapnutím logovacích výpisov na linke.

Pokiaľ bude záujem rozšíriť implementáciu BACnet MS/TP, bude potrebná explicitná požiadavka a zabezpečenie testovacej konfigurácie.

Poznámka k podpore BBMD (BACnet Broadcast Management Devices)

Do protokolu BACnet bola dorobená (a je k dispozícii vo verziách zverejnených po 17.01.2012) nasledovná vlastnosť týkajúca sa prevodu textových mien na iselné adresy meraných bodov (Siemens zariadenia Desigo) na linke [TCP/IP-UDP](#): ak KOM proces vyšle požiadavku Who-Has a odpoveď I-Have príde z inej IP adresy, hádá sa (v rámci linky) stanica, ktorá vyslala Who-Has request s textovým názvom, ktorý je uvedený v odpovedi. Ak sa takáto stanica nájde, odpoveď sa spája s jej výzvou.

Táto vlastnosť je užitočná, pokiaľ sa komunikuje s viacerými Siemens routermi (za ktorými sú napr. BACnet/LON zariadenia), z ktorých jeden je nakonfigurovaný ako BBMD (Bacnet Broadcast Management Device) a ostatné nie.

Modelová situácia:

KOM proces komunikuje s dvoma BACnet routermi Desigo PXG80-N (Rtr1 a Rtr2). Za každým routerom je LON sieť, pre zjednodušenie s jedinou stanicou Desigo PXM20 (Des1 resp. Des2). Router Rtr2 je nakonfigurovaný tak, aby BACnet broadcasty z LON siete preposielal na Rtr1.

- Rtr1 má IP adresu 10.0.0.1 a má aktivovanú BBMD funkčnosť. Za Rtr1 je LON sieť s adresou siete 11 a na nej zariadenie Desigo PXM20 (Des1) s adresou 1.1
- Rtr2 má IP adresu 10.0.0.2 a má vypnutú BBMD funkčnosť. Za Rtr2 je LON sieť s adresou siete 12 a na nej zariadenie Desigo PXM20 (Des2) s adresou 2.2

Majme jednu komunikačnú linku typu [TCP/IP-UDP](#) na nej dve stanice typu BACnet/IP reprezentujúce Des1 a Des2

- Linka L.Bacnet: typ [TCP/IP-UDP](#)
- Stanica B.Des1:
Typ stanice: BACnet/IP
Adresa: 10.0.0.1 (adresa Rtr1, za ktorým sa nachádza Des1)
Port: 47808 (štandardný BACnet port)
Destination network: 11 (adresa siete za Rtr1)
Destination address: 1.1 (adresa Des1 v LON sieti)
Register-Foreign-Device: zapnutý (aby sa KOM proces registroval ako príjemca broadcastov u Rtr1, ktorý má aktívnu BBMD funkčnosť)
- Stanica B.Des2:
Typ stanice: BACnet/IP
Adresa: 10.0.0.2 (adresa Rtr2, za ktorým sa nachádza Des2)
Port: 47808 (štandardný BACnet port)
Destination network: 12 (adresa siete za Rtr2)
Destination address: 2.2 (adresa Des2 v LON sieti)
Register-Foreign-Device: vypnutý (Rtr2 má neaktívnu BBMD funkčnosť)
- ubovolný bod M.Des2_test na stanici B.Des2 (napr. typu SubscribeCOV) s Address type=Name

KOM proces pošle kvôli prevodu textového názvu nakonfigurovaného v adrese M.Des2_test Who-Has požiadavku na zariadenie Des2. Pri posielaní sa použije IP adresa Rtr2 10.0.0.2.

Rtr2 prepošle požiadavku do LON siete na Des2 (poda parametrov Destination network=12 a Destination address=2.2 uvedených v konfigurácii stanice B.Des2). Z BACnet/LON zariadenia Des2 príde broadcast odpoveď I-Have. Keže BACnet router Rtr2 je bez podpory BBMD, prepošle odpoveď (za predpokladu, že je tak nakonfigurovaný) BACnet routeru Rtr1, ktorý má podporu BBMD.

Rtr1 prepošle I-Have správu ako UDP paket KOM procesu, pretože je nakonfigurovaný parameter stanice B.Des1 Register-Foreign-Device a teda KOM proces sa registroval ako príjemca broadcast správ u Rtr1). Takto sa I-Have správa dostane do KOM procesu s inou IP adresou ako bola pôvodná výzva. Vyššie popisovaná funkčnosť spája takúto odpoveď s výzvou od stanice B.Des2 na základe zhody textového názvu objektu v poslanej výzve.

Z popísaného postupu zároveň vyplýva jedno konfiguračné obmedzenie - všetky popísané inosti sa dejú v kontexte jednej linky, takže je nutné, aby na jednej linke sa nachádzali všetky stanice, ktoré sú v pôsobnosti jedného BBMD zariadenia.

Tell príkazy

Príkaz	Syntax	Popis
STWATCH CH	STWATCH MenoStanice	Tell príkaz pošle na stanicu príkazy ReadProperty, ReadPropertyMultiple a Subscribe podľa konfigurácie jednotlivých meraných bodov.

Literatúra

ANSI/ASHRAE Standard 135-2001: BACnet - A Data Communication Protocol for Building Automation and Control Networks

ASN.1 Complete by Prof. John Larmouth



Blogy

O protokole BACnet si môžete preíta aj blogy:

- [Komunikácia - BACnet protokol](#)
- [Komunikácia - BACnet protokol, as 2](#)
- [Komunikácia - BACnet protokol, as 3](#)

Zmeny a úpravy

-

Revízie dokumentu

- Ver. 1.0 - 30. august 2005 - Vytvorenie dokumentu.
- Ver. 1.1 - 20. október 2005 - Podpora schedulerov, zápis a ítanie ASN sekvencií.
- Ver. 1.2 - 22. november 2005 - Podpora dynamického zisovania adresy meraného bodu z mena objektu, cachovanie adries v súbore
- Ver. 1.3 - 14. jún 2006 - Podpora BACnet routra (otestované s PXG80-N)
- Ver. 1.4 - 02. apríl 2008 - iastoná podpora BACnet MS/TP protokolu (vyskúšané na BACnet MS/TP MICROGATEWAY na chladii od firmy York)
- Ver. 1.5 - 25. jún 2010 - Otestovanie spolupráce s E-DDC3.1 od firmy SE-Elektronik GmbH.



Súvisiace stránky:

[Komunikané protokoly](#)