

Komunikace v síti M-Bus (Gen1)

Abstrakt

Přenos technologických dat mezi řídicími systémy firmy AMiT a měřiči energií prostřednictvím protokolu M-Bus.

Autor: Zbyněk Říha, Libor Urbačka
Dokument: ap0010_ap_cz_005.pdf

Příloha

Obsah souboru: ap0010_ap_cz_005.zip

abb_p1_cz_01.dso	Příklad komunikace s měřičem F2 od firmy ABB.
comac_p1_cz_02.dso	Příklad komunikace s měřičem CALOR40 firmy COMAC CAL.
danfos_p1_cz_02.dso	Příklad komunikace s měřičem INFOCAL 5 firmy DANFOS.
enbra_p1_cz_01.dso	Příklad komunikace s měřičem SONTEX firmy ENBRA.
landisgyr_p1_cz_03.dso	Příklad komunikace s měřičem 2WR5 firmy SIEMENS (Landis Gyr).
sensus_p1_cz_01.dso	Příklad komunikace s vodoměrem firmy SENSUS.
univerz_p1_cz_02.dso	Univerzální komunikační jádro pro M-Bus (bez dekódování dat).
uni_dek_p1_cz_01.dso	Univerzální komunikační jádro pro M-Bus s dekódováním dat.
modbus_p1_cz_100.dso	Univerzální komunikační jádro protokolem MODBUS pro M-Bus (bez dekódování dat)

Obsah

Obsah	2
Historie revizí	3
Související dokumentace	3
Určení aplikační poznámky	3
1 M-Bus	4
1.1 Základní charakteristika	4
1.2 Fyzická vrstva	4
1.3 Formát rámce	5
1.3.1 Význam polí rámce	5
1.3.2 Využití rámců	6
2 Komunikace řídicích systémů pomocí M-Bus.....	7
2.1 HW parametrizace	7
2.2 SW parametrizace.....	8
2.2.1 ComInit.....	9
2.2.2 ComWrite	9
2.2.3 ComRead	9
2.2.4 Celková koncepce uživatelské komunikace.....	10
2.2.5 MODBUS komunikace	12
2.2.6 Celková koncepce MODBUS komunikace	12
3 Ukázkové aplikace	14
4 Technická podpora	15
5 Upozornění	16

Historie revizí

Verze	Datum	Autor změny	Změny
001	21. 07. 2008	Říha Z., Urbačka L.	Nový dokument.
002	17. 12. 2009	Říha Z.	Oprava překlepu, úprava aplikací, doplnění nové aplikace.
003	30. 03. 2015	Říha Z.	Doplnění nové aplikace.
004	06. 01. 2022	Kupčík M.	Větší úpravy textů, nové kapitoly 2.2.5 a 2.2.6.
005	29. 03. 2023	Novotný M.	Úprava aplikace pro měřič 2WR5.

Související dokumentace

1. Náповěda k části PseDet vývojového prostředí DetStudio
soubor: PseDet_cz.chm
2. **AM-MBUS** – komunikační modul – Katalogový list
soubor: am-mbus_d_cz_xxx.pdf
3. **DM-MBUS64** – komunikační převodník s protokolem M-Bus – Návod na obsluhu
soubor: dm-mbus64_g_cz_xxx.pdf
4. Aplikační poznámka AP0008 – Komunikace v síti MODBUS RTU (Gen1)
soubor: ap0008_ap_cz_xxx.pdf
5. Aplikační poznámka AP0059 – Komunikace v síti MODBUS TCP (Gen1)
soubor: ap0059_ap_cz_xxx.pdf
6. EN 13757 – Sada norem specifikující protokol M-Bus

Určení aplikační poznámky

Aplikační poznámka je určena řídicím systémům Generace 1 – řídicím systémům s operačním systémem NOS programovaných v DetStudiu v editoru PseDet. Více informací o generacích systémů lze nalézt na webu amitautomation.cz.

1 M-Bus

Standard M-Bus (z anglického Meter-Bus) je určen pro sběr dat z měřičů odběru nejrůznějších médií (například pitné a užitkové vody, plynu, tepla, elektrické energie). Byl vyvinut ve spolupráci University Paderborn a firmy Texas Instruments.

1.1 Základní charakteristika

Vzhledem k relativně úzké a poměrně specializované aplikační oblasti jsou na M-Bus kladeny specifické požadavky. Musí zajistit propojení velkého množství zařízení na vzdálenost až několika kilometrů. Přenos dat musí být kvalitně zabezpečen proti chybám. Na druhé straně je typickou vlastností aplikace nepříliš časté odečítání naměřených hodnot s nízkými nároky na odezvy v reálném čase.

Charakteristické rysy standardní konfigurace lze shrnout do následujících bodů:

- ♦ speciální implementace fyzické vrstvy,
- ♦ galvanicky oddělené rozhraní,
- ♦ možnost napájení připojených zařízení po sběrnici,
- ♦ dvoudrátové vedení s délkou až několik kilometrů,
- ♦ řízení komunikace na principu Master – Slave,
- ♦ bez implementace síťové vrstvy maximálně 250 účastníků,
- ♦ asynchronní přenos znaků, 8 bitů dat, sudá parita,
- ♦ přenosová rychlost 300 bps až 38400 bps,
- ♦ zabezpečení datového bloku pomocí kontrolního součtu.

Standardní konfigurace M-Busu zahrnuje jedinou řídicí stanici (Master) a maximálně 250 účastnických stanic. Typický způsob zapojení M-Bus měřičů je do linie. Délka kabelového segmentu v této konfiguraci nesmí překročit 1000 m (350 m pro 9600 bps). Pro rozsáhlejší systémy je nezbytné přejít k složitějším konfiguracím, kdy je celý systém rozdělen na tzv. zóny. Jednotlivé zóny se skládají ze segmentů a jsou řízeny tzv. řadiči zóny.

Kompletní detailní popis M-Bus protokolu poskytuje EN 13757.

1.2 Fyzická vrstva

Standard M-Bus používá speciální implementace fyzické vrstvy protokolu, které jsou popsány v EN 13757-2 a EN 13757-4. Typicky se jedná o dvoudrátovou sběrnici na bázi běžného telefonního kabelu s poloduplexním přenosem dat a řízením přístupu Master – Slave. Aby bylo možné napájení účastnických stanic po tomtéž vedení, používá M-Bus pro přenos od řídicí stanice k účastnickým stanicím změny napěťových úrovní, ve směru opačném změny v odběru proudu. Ve směru od řídicí stanice k účastnickým stanicím odpovídá logické jedničce napětí +36 V na výstupu budiče řídicí stanice, logické nule pak napětí o 12 V nižší, tj. +24 V. Ve směru od účastnické stanice k řídicí stanici je logická jednička reprezentována proudovým odběrem 1,5 mA, logická nula odběrem o 11 mA až 20 mA vyšším. Proud odebíraný ve stavu log. 1 může být využit k napájení galvanicky odděleného rozhraní a případně i vlastního měřiče.

V klidovém stavu je na sběrnici úroveň log. 1 (+36 V) a odběr z budiče řídicí stanice odpovídá počtu účastnických stanic násobeným odběrem účastnické stanice ve stavu log. 1, tj. 1,5 mA. Je zřejmé, že zejména při vyšším počtu účastnických stanic klade standard na budič řídicí stanice poměrně vysoké nároky. Navíc vzhledem k proměnnému počtu účastnických stanic a k požadavku možnosti jejich přidávání a odebírání za provozu nemůže řídicí stanice detekovat absolutní odebíraný proud, ale pouze výše uvedenou změnu. Stejně tak napětí v různých bodech sběrnice, které je monitorováno účastnickými stanicemi, je proměnné a závisí na odporu vedení a na počtu účastnických stanic k němu připojených. To znamená, že také účastnické stanice musí reagovat na změny napětí sběrnice, nikoliv na jejich absolutní hodnoty. Vzhledem k poměrně velkým změnám jak napětí (12 V) tak proudu (11 mA až 20 mA) vykazuje fyzická vrstva standardu vysokou odolnost vůči vnějšímu rušení.

Druhá varianta je použití bezdrátového přenosu dat. Vzhledem k nemožnosti komunikace řídicích systémů AMiT v daných frekvenčních pásmech není tato varianta v této aplikační poznámce více řešena.

1.3 Formát rámce

M-Bus využívá čtyř formátů rámce:

- ♦ jednotlivý znak (Single Character),
- ♦ krátký rámec (Short Frame),
- ♦ řídicí rámec (Control Frame),
- ♦ dlouhý rámec (Long Frame).

Jednotlivý znak	Krátký rámec	Řídicí rámec	Dlouhý rámec
0xE5	Start 0x10	Start 0x68	Start 0x68
	C pole	L pole = 3	L pole
	A pole	L pole = 3	L pole
	Kontrolní součet	Start 0x68	Start 0x68
	Stop 0x16	C pole	C pole
		A pole	A pole
		CI pole	CI pole
		Kontrolní součet	Uživatelská data (0 až 252) bajtů
		Stop 0x16	Kontrolní součet
			Stop 0x16

Jednotlivý znak

Tento rámec se skládá z jediného znaku, jmenovitě 0xE5. Používá se k potvrzení o přijetí jiného vyslaného rámce.

Krátký rámec

Tento rámec má pevnou délku. Začíná úvodním znakem 0x10, následuje pole C, pole A, kontrolní součet a koncový znak 0x16. Kontrolní součet se počítá pouze z polí C a A.

Řídicí rámec

Obsah řídicího rámce odpovídá dlouhému rámci, neobsahuje však položku „uživatelská data“. Kontrolní součet je počítán z polí C, A a CI.

Dlouhý rámec

Tento rámec začíná znakem 0x68, pokračuje dvakrát opakovaným polem L, obsahujícím délku dat a opět úvodním znakem 0x68. Poté následuje pole C, pole A, pole CI, uživatelská data o délce (0 až 252) bajtů, kontrolní součet a ukončovací znak 0x16. Pole L obsahuje počet bajtů uživatelských dat zvýšený o tři (tj. o délku polí C, A a CI). Kontrolní součet je počítán z polí C, A, CI a z uživatelských dat.

1.3.1 Význam polí rámce

Všechna pole mají délku jednoho znaku, což odpovídá osmi bitům.

Pole C (Control Field, Function Field)

Toto pole obsahuje řídicí parametry rámce. Mimo jiné určuje směr toku dat. V popise níže je označován jako hlavní bajt rámce.

Pole A (Address Field)

Pole A slouží k adresaci či určení účastnické stanice při vysílání, resp. příjmu rámce.

Jednotlivým účastnickým stanicím (slaveům) mohou být přiřazeny adresy 0 až 250. V případě, že bude do adresního pole zadána hodnota 255, bude vyslán tzv. nepotvrzovaný broadcast (nepřijde žádná odpověď). Pokud bude do adresního pole zadána hodnota 254 dojde k vyslání tzv. potvrzovaného broadcastu (očekává se odpověď). Je zřejmé, že při potvrzení broadcastu dojde na sběrnici ke kolizím vždy, je-li připojena více než jedna účastnická stanice. Jeho použití je pouze pro testovací účely. Adresa 253 indikuje tzv. sekundární adresaci. Adresy 251 a 252 jsou rezervovány pro budoucí využití.

V této aplikační poznámce se sekundární adresace neřeší.

Pole CI (Control Information Field)

Pole CI nese informaci, která je již součástí aplikačního protokolu. Jeho úkolem je kromě jiného rozlišení formátu dlouhého a řídicího rámce.

Uživatelská data

V případě, že se jedná o odpověď měřiče s běžnými daty (pole CI má hodnotu 0x72 nebo 0x76 – RSP_UD), mají tyto bajty následující význam:

- ◆ Hlavičkové bajty 8 až 19:
 - ◆ 4 bajty identifikačního čísla měřiče, často odpovídající části sériového čísla nebo sekundární adresy,
 - ◆ 2 bajty kódující tři identifikační písmena výrobce měřiče,
 - ◆ 1 bajt verze měřiče,
 - ◆ 1 bajt kódující měřené médium,
 - ◆ 1 bajt počítadla rámců,
 - ◆ 1 bajt kódující stav měřiče,
 - ◆ 2 bajt „signature“ – příprava pro budoucí specifikaci, hodnoty 0x00 0x00.
- ◆ následující datové bajty:
 - ◆ 1 bajt DIF – definice typu čísla hodnoty; pokud je nastaven nejvyšší bit, bude následovat DIFE,
 - ◆ 0 až 10 bajtů DIFE – rozšiřující identifikátor hodnoty; pokud je nastaven nejvyšší bit, bude následovat další DIFE,
 - ◆ 1 bajt VIF – jednotka a násobitel hodnoty; pokud je nastaven nejvyšší bit, bude následovat VIFE,
 - ◆ 0 až 10 bajtů VIFE – rozšiřující identifikátor jednotky a násobitele hodnoty; pokud je nastaven nejvyšší bit, bude následovat další VIFE,
 - ◆ 0 až xx bajtů hodnota; v případě ASCII jednotky je tato prvně poslána a následně jsou poslány bajty hodnoty,
 - ◆ následuje tolik opakování kombinací DIF (+ n×DIFE) + VIF (+ m×VIFE) + hodnota, kolik veličin měřič v této zprávě zasílá.

1.3.2 Využití rámců

V této aplikační poznámce se využívají rámce:

- ◆ jednotlivý znak – odpověď měřiče na inicializaci,
- ◆ krátký rámec – inicializace měřiče (SND_NKE) – použit hlavní bajt rámce 0x40,
- ◆ krátký rámec – požadavek na běžná data (SND_UD2) – použit hlavní bajt rámce 0x5B nebo 0x7B,
- ◆ dlouhý rámec – odpověď měřiče s běžnými daty (RSP_UD) – přijat hlavní bajt 0x08/0x18/0x28/0x38.

2 Komunikace řídicích systémů pomocí M-Bus

Řídicí systémy firmy AMiT zastávají v síti M-Bus vždy roli **Master** a nelze je tedy v síti M-Bus provozovat v režimu Slave.

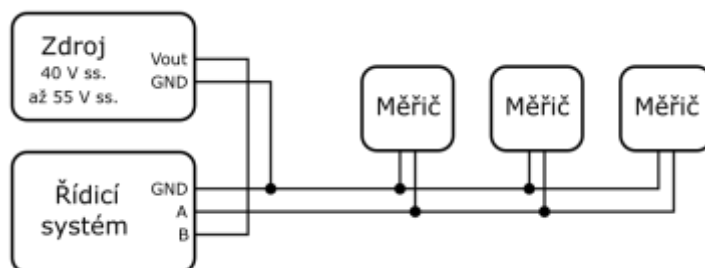
2.1 HW parametrizace

Některé kompaktní řídicí systémy lze do sítě M-Bus připojit pomocí modulu **AM-MBUS** z produkce firmy AMiT. Jedná se o zásuvný modul, který se umístí pod kryt kompaktního řídicího systému. Maximální počet zařízení komunikujících s řídicím systémem pomocí protokolu M-Bus prostřednictvím modulu **AM-MBUS** je dán parametry tohoto modulu (viz příslušný katalogový list).

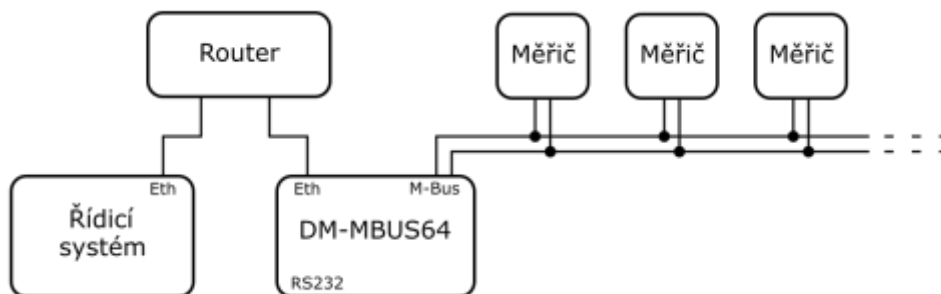
Ostatní řídicí systémy, ve kterých nelze modul **AM-MBUS** použít lze do sítě M-Bus připojit např. prostřednictvím převodníku 232 <-> M-Bus. Firma AMiT nabízí pro komunikaci v síti M-Bus převodník **DM-MBUS64**, který lze k řídicímu systému připojit jak prostřednictvím sériového rozhraní, tak prostřednictvím rozhraní Ethernet. Převodník **neprovádí** žádné přímé dekódování hodnot z měřičů.

Parametrizace převodníku se provádí typicky prostřednictvím webových stránek, když je převodník v tzv. servisním režimu. Více informací o parametrizaci převodníku jsou uvedeny v návodu na obsluhu převodníku.

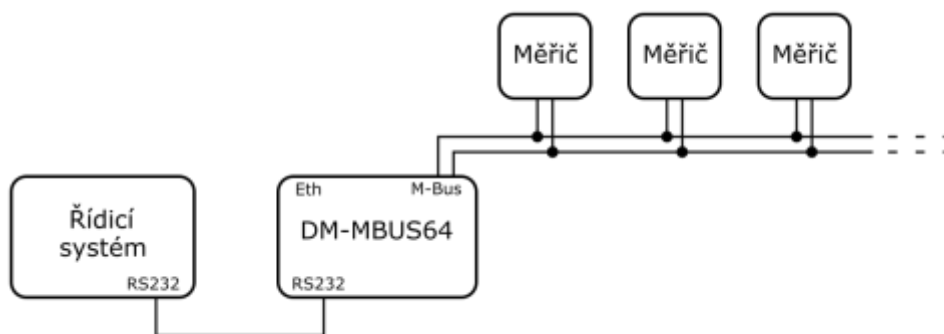
Převodník umožňuje komunikaci prostřednictvím transparentního přenosu dat (režim Direct), zapouzdření M-Bus dat do protokolu APE a prostřednictvím protokolů MODBUS.



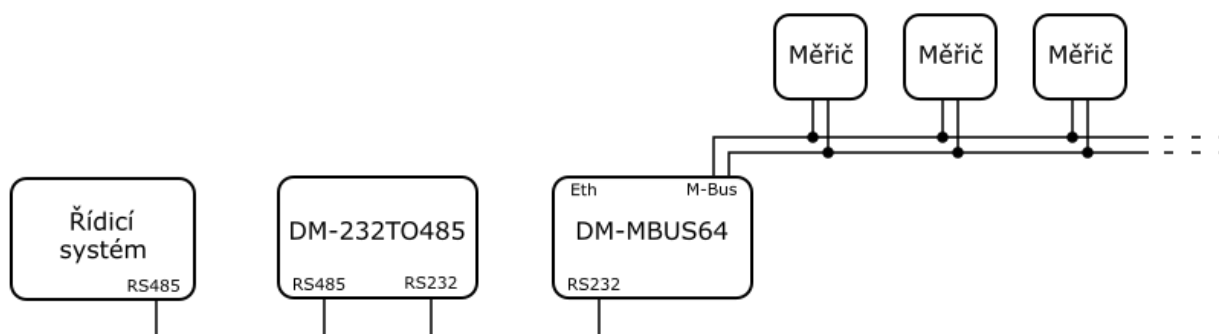
Obr. 1 – Připojení řídicího systému přes komunikační modul **AM-MBUS**



Obr. 2 – Připojení řídicího systému k převodníku **DM-MBUS64** přes rozhraní Ethernet



Obr. 3 – Připojení řídicího systému k převodníku **DM-MBUS64** přes rozhraní RS232



Obr. 4 – Připojení řídicího systému k převodníku **DM-MBUS64** přes RS485 a převodník **DM-232TO485**

Algoritmy uvedené níže lze použít i v případě použití převodníku jiného výrobce. Převodník jiného výrobce však musí umožňovat transparentní přenos dat (co přijme prostřednictvím sériového rozhraní nebo rozhraní Ethernet, to musí poslat na rozhraní M-Bus bez jakékoliv změny [po stránce SW] a obráceně).

Pozor

Na jednom sériovém rozhraní řídicího systému nebo terminálu, lze provozovat pouze jeden komunikační protokol. V případě využití komunikace s převodníkem po sériovém rozhraní musí být zvolen stejný komunikační protokol jako s případnými jinými zařízeními na tomto sériovém rozhraní.

2.2 SW parametrizace

Pro komunikaci s měřičem se typicky využívá následujícího postupu:

1. Vyslání inicializačního krátkého rámce,
2. čekání na jednoznakovou odpověď 0xE5,
3. a) v případě, že odpověď nepřijde, opakovat vyslání inicializačního krátkého rámce,
b) v případě, že odpověď přijde, vyslat krátký rámec s požadavkem na běžná data,
4. a) v případě, že odpověď s běžnými daty nepřijde, opakovat vyslání krátkého rámce s požadavkem na běžná data,
b) v případě, že odpověď s běžnými daty přijde, provést jejich dekodování,

V závislosti na použitém HW je možno výše uvedený komunikační postup realizovat buď pomocí bloků tzv. uživatelské komunikace **ComInit**, **ComWrite** a **ComRead** nebo definice MODBUS komunikace.

2.2.1 ComInit

Jedná se o hlavní modul uživatelské komunikace, ve kterém definujeme např. komunikační port, prostřednictvím kterého chceme vysílat a přijímat data, tvar znaku (délka, parita, počet stopbitů), komunikační rychlost atd.

Tento modul se umísťuje do procesu INIT.

Popis parametrů tohoto modulu je součástí nápovědy k návrhovému prostředí DetStudio.

V případě, že je použit převodník **DM-MBUS64**, je nutné v převodníku nastavit mód komunikace Direct na rozhraní RS232 a mít na tomto rozhraní nastavené identické komunikační parametry (tvar znaku a komunikační rychlost), jako se nastavují v modulu **ComInit**.

2.2.2 ComWrite

Modul slouží pro zápis vysílaných znaků do vysílacího bufferu. Tomuto modulu se zadá odkaz na buffer (databázovou proměnnou), ve kterém je připraven vysílaný rámeček. Modul provede to, že znaky ze svého bufferu přesune do interního vysílacího bufferu modulu **ComInit** (parametr **TxBuf**). Modul **ComInit** začne znaky okamžitě vysílat. O vysílání jednotlivých znaků se aplikace dále nemusí starat. Je-li to z nějakého důvodu potřeba, lze vyplnit parametr **TxLtr** modulu **ComInit** a dostávat tak přerušování od vyslaného znaku i do aplikace. Modul **ComInit** pak vyvolává příslušný podprogram bezprostředně po odeslání každého znaku. Parametrem **LastIntr** modulu **ComInit** lze určit, zda se má vyvolávat podprogram přerušování po každém odvysílaném znaku nebo pouze po posledním. Poslední znakem se myslí situace, kdy modul **ComInit** při vysílání vybere celý buffer a nemá již co vysílat. Prakticky to znamená, že když modulem **ComWrite** vložíme celý rámeček, dostaneme přerušování po odvysílání posledního znaku tohoto rámce. Pokud bychom ovšem ještě předtím než se stačí rámeček odvysílat, vložili další vysílané znaky opětovným voláním modulu **ComWrite**, dostaneme přerušování až po odvysílání všech těchto vložených znaků.

Popis parametrů tohoto modulu je součástí nápovědy k návrhovému prostředí DetStudio.

2.2.3 ComRead

Modul slouží pro načtení přijatých znaků z přijímacího bufferu. Z hlediska zpracování přijatých znaků umožňuje modul dva přístupy.

Dotazovací (pooling)

Modul ukládá přijaté znaky do interního bufferu (parametr **RxBuf**). Aplikace se periodicky dotazuje, zda něco přišlo. Provádí to modulem **ComRead**, který znaky z interního bufferu přesouvá do svého vlastního bufferu. Tento vlastní buffer je potom přístupný pro další zpracování v aplikaci.

Při dotazovacím přístupu se parametr **RxLtr** modulu **ComInit** nezadává (ponechává se hodnota **NONE**).

Přerušovací (interrupt)

Modul vyvolá okamžitě po přijetí znaku uživatelský podprogram na zpracování přijatého znaku. V uživatelském podprogramu se znak vybere opět pomocí modulu **ComRead**. Jelikož doba provádění tohoto podprogramu může být v některých případech relativně dlouhá, může dojít k tomu, že další znak přijde dřív, než se stačí podprogram dokončit. Aby tato situace nezpůsobila ztrátu přijatého znaku, ukládá modul **ComInit** přijatý znak do interního bufferu (parametr **RxBuf**) a speciálním mechanismem vyvolává uživatelský podprogram tak, aby jej vyvolal vždy až po dokončení předchozího vyvolání. Tak je zajištěno, že se přijaté znaky neztrácí. Vedlejším účinkem tohoto mechanismu je to, že na jedno vyvolání podprogramu může modul **ComRead** načíst i více znaků než jeden. Počet vyvolání podprogramu tak může být menší než počet přijatých znaků.

Při přerušovacím přístupu se parametr `Rxltr` modulu `ComInit` musí zadat.

Doporučujeme používat raději přerušovací přístup, protože méně zatěžuje systém. Přijaté znaky může aplikace zpracovávat přímo v podprogramu přerušování. Reakce na přijatý rámec tak může být rychlejší než v případě dotazovacího přístupu.

Popis parametrů tohoto modulu je součástí nápovědy k návrhovému prostředí `DetStudio`.

2.2.4 Celková koncepce uživatelské komunikace

Popíšeme jednotlivé části, ze kterých se komunikace obecně skládá. Dále pak popíšeme, jak jednotlivé části realizovat – které moduly se mají použít a jakým způsobem.

Definice parametrů linky

Parametry se definují v hlavním modulu `ComInit`.

Sestavení vysílaného rámce

Sestavením rámce se myslí zformátování hodnot nějakých databázových proměnných do formátu vysílaného rámce. Slouží k tomu modul `StrFormat`. Modul umí zformátovat jednu hodnotu do několika buněk výstupní databázové proměnné typu `MI`. Skládá-li se rámec z více hodnot, řeší se to vícenásobným vložením modulu s ukládáním na různé indexy buněk výstupní matice. Každá jedna buňka této maticové proměnné odpovídá jednomu znaku rámce.

Zabezpečení vysílaného rámce

Pro tvorbu zabezpečení vysílaného rámce se používá modul `ChkCreate` viz “Kontrola zabezpečení přijatého rámce” níže.

Vyslání celého rámce

Sestavený rámec vyšleme najednou pomocí modulu `ComWrite`.

Vysílání jednotlivých znaků vysílaného rámce

O vysílání jednotlivých znaků se aplikace nestará, je zabezpečeno interně modulem `ComInit`.

Kontrola maximálního času čekání na odpověď

Na kontrolu časových prodlev se používají moduly `Tmo`, `TmoStart`, `TmoStop` a `TmoCheck`. Viz “Kontrola maximální časové prodlevy (timeoutu) mezi přijímanými znaky rámce” níže.

Příjem znaků přijímaného rámce

Na příjem znaků se používá modul `ComRead`. Vkládá se buď do podprogramu přerušování od přijatého znaku (přerušovací přístup) nebo do periodického procesu (dotazovací přístup).

Kontrola maximální časové prodlevy (timeoutu) mezi přijímanými znaky rámce

Na kontrolu časových prodlev se používají moduly `Tmo`, `TmoStart`, `TmoStop` a `TmoCheck`. Modul `Tmo` definuje jeden kanál timeoutu. Modulem `TmoStart` se startuje timeout, modulem `TmoStop` se timeout zastavuje. Modul `TmoCheck` slouží ke kontrole stavu timeoutu (zda již nastal nebo ne). Typicky se moduly používají v přerušovacím režimu. Modulu `Tmo` se zadá odkaz na podprogram, který bude modulem spouštěn jako přerušování bezprostředně po uplynutí timeoutu. Do tohoto podprogramu se umísťuje kód na reakci na timeout.

Modul `TmoCheck` slouží naopak pro dotazovací režim. Lze jej volat periodicky a zjišťovat, zda již timeout nastal. V přerušovacím režimu se modul nepoužívá, protože vypršení timeoutu je jednoznačně identifikováno vyvoláním podprogramu přerušování.

Rozpoznání začátku a konce rámce

Jsou v zásadě tři možnosti:

♦ očekávání rámce

Jedná se o nejjednodušší (nezabezpečený) způsob rozpoznávání rámců. Lze jej provádět u protokolů typu bod-bod master-slave, kdy se komunikuje způsobem dotaz-odpověď. Master zasílá dotazy nebo příkazy a na každý takový rámec následuje odpověď od slavea. Tehdy se dá předpokládat, že první znak, který přijde po odeslání rámce stanicí, je začátek rámce protistrany. Rámce mohou mít pevnou strukturu, ze které jednoznačně vyplývá délka rámce. Při příjmu (obvykle v podprogramu přerušení od přijatého znaku) se kontroluje délka rámce a dle ní se rozpoznává konec rámce.

♦ speciální znaky začátku a konce rámce

Začátek a konec rámce je označen speciálními znaky začátku a konce, které se na jiném místě v rámci již nesmí vyskytnout. Pokud se při sestavování uvnitř rámce objeví znak stejný jako znak začátku nebo konce rámce, musí být tento znak nějak nahrazen. Obvykle se provádí náhrada dvěma znaky, přičemž první znak má význam přesmykače, a druhý znak je definovaným způsobem překódovaný původní znak. Když se potom rámec na protistanici rozebírá, tak přesmykač oznamuje, že následný znak je překódován a je jej tedy nutno překódovat zpět. Na překódování speciálních znaků uvnitř rámce tam i zpět se používá modul `StrSubst`. Vlastní rozpoznávání začátku a konce rámce se provádí obvykle v podprogramu přerušení od přijatého znaku. Je ovšem nutno pamatovat na to, že z výše uvedených důvodů může modul `ComRead` na jedno vyvolání podprogramu načíst i více než jeden znak. Z toho důvodu nelze jednoduše kontrolovat poslední přijatý znak a porovnávat jej se znakem začátku rámce. V tomto případě se musí projít v cyklu všechny znaky, které modul v každém běhu načte.

Kontrola znaku konce rámce je jednodušší, protože za posledním znakem rámce bývá obvykle delší časová mezera, kdy protistrana čeká na odpověď. V takovém případě se lze spolehnout na kontrolu pouze posledního přijatého znaku.

Poznámka

Problémům s načtením více znaků na jedno vyvolání podprogramu se lze vyhnout tím, že modulu `ComRead` se zadá za parametr `Buffer` místo matice jednoduchá proměnná. Modul pak na jedno vyvolání načte pouze jeden znak. Aplikace pak ovšem musí zajistit postupné ukládání celého přijímaného rámce do nějaké matice ve své režii.

♦ časová mezera mezi rámci

Konec rámce je identifikován časovou mezerou mezi znaky. Je-li tedy mezera mezi přijímanými znaky větší než stanovená ukončovací mezera, jedná se o konec rámce. Takto zadané rámce se rozpoznávají tak, že po příjmu každého znaku se nastartuje timeout s hodnotou stanovené ukončovací mezery. Nastane-li timeout, vyvolá se podprogram přerušení od timeoutu a v tomto podprogramu se zpracuje přijatý rámec.

Kontrola zabezpečení přijatého rámce

Aby byl rámec zabezpečen proti zkreslení při přenosu, provede se před vysláním kontrolní součet jednotlivých znaků dat rámce a takto vzniklé zabezpečení se přidá do vysílaného rámce. Protistrana po přijetí rámce provede znovu kontrolní součet znaků dat a porovná jej s přijatým zabezpečením. Pokud se zabezpečení liší, vyhodnotí se to jako chyba při přenosu rámce. Pojem "kontrolní součet" zde uvedený znamená obecně nějaký typ zabezpečení.

Pro vytváření zabezpečení vysílaného rámce se používá modul `ChkCreate`. Pro kontrolu zabezpečení přijatého rámce se používá modul `ChkCheck`. Těmto modulům se zadává konkrétní typ zabezpečení (CRC, suma, XOR apod.).

Rozbor rámce

Rozborem rámce se myslí vypreparování dat z přijatého rámce do databázových proměnných. Slouží k tomu modul `StrParse`. Modul umí vypreparovat jednu hodnotu.

Skládá-li se rámec z více hodnot, řeší se to vícenásobným vložením modulu.

2.2.5 MODBUS komunikace

V případě využití převodníku **DM-MBUS64** lze využívat komunikační protokol MODBUS. Konkrétně MODBUS RTU při komunikaci přes sériové rozhraní a MODBUS TCP při komunikaci prostřednictvím rozhraní Ethernet. Druhý zmíněný způsob lze využívat pouze na systémech s písmenem „W“ v názvu (tzv. dvouprocesorové řídicí systémy).

V převodníku je nutné nastavit mód komunikace MODBUS na rozhraní, prostřednictvím kterého se bude komunikovat protokolem MODBUS. Při komunikaci prostřednictvím MODBUS TCP musí být dále vhodně nastaven komunikační TCP port, při komunikaci prostřednictvím MODBUS RTU musí být dále vhodně nastaveny komunikační parametry (tvar znaku a komunikační rychlost).

Pro komunikaci s převodníkem se používá holding registrů na adresách:

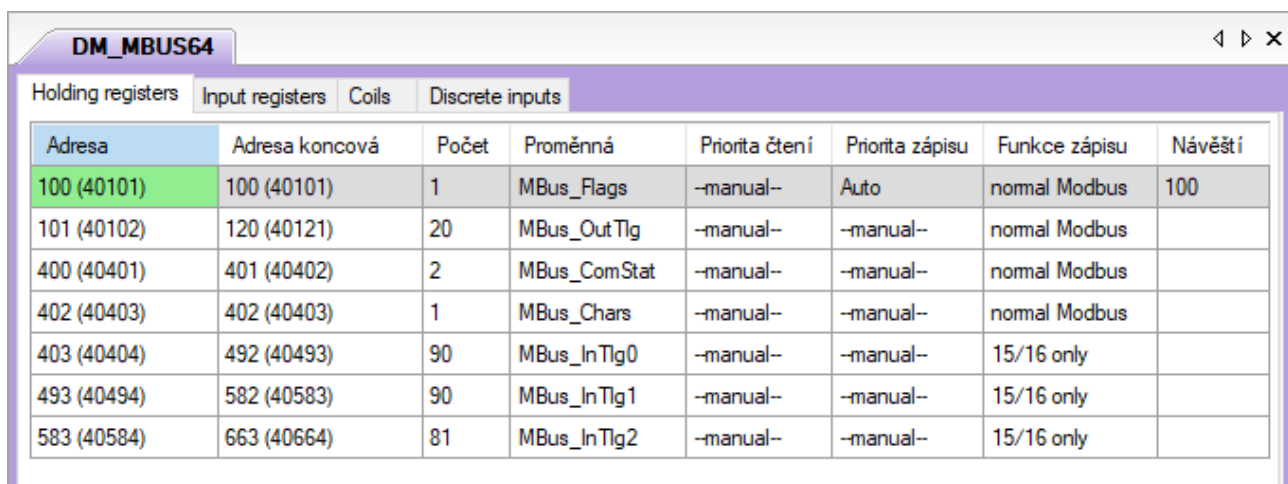
- ◆ 100 – nastavení akce převodníku,
- ◆ 101 až 361 – odchozí rámec do sítě M-Bus,
- ◆ 400 – zjišťování stavu příchozího rámce,
- ◆ 401 – kód poslední chyby,
- ◆ 402 – počet příchozích znaků ze sítě M-Bus,
- ◆ 403 až 663 – příchozí rámec ze sítě M-Bus.

Přesný popis akcí převodníku a komunikačních registrů je uveden v návodu na obsluhu převodníku **DM-MBUS64**.

2.2.6 Celková koncepce MODBUS komunikace

Pro inicializaci měřiče nebo požadavku na běžná data se používá krátkého rámce s délkou 5 bajtů. Jedna z funkcí převodníku **DM-MBUS64** je automatické počítání hodnoty kontrolního součtu a připojování koncového znaku 0x16. Do převodníku tedy stačí zapisovat pouze 3 bajty odchozího rámce a komunikovat registry 101 až 103. Vzhledem k možným budoucím použitím je v ukázkové aplikaci zvolena definice komunikace s 20 registry v rozsahu 101 až 120.

Protokol MODBUS podporuje vyčtení maximálně 125 holding registrů. Příchozí rámec ze sítě M-Bus však může mít až 261 znaků. Z tohoto důvodu se musí registry 403 až 663 rozdělit do tří komunikovaných skupin. V ukázkové aplikaci je zvoleno dělení na 90, 90 a 81 registrů. Toto platí pro komunikaci protokolem MODBUS RTU i protokolem MODBUS TCP.



Adresa	Adresa koncová	Počet	Proměnná	Priorita čtení	Priorita zápisu	Funkce zápisu	Návěští
100 (40101)	100 (40101)	1	MBus_Flags	-manual-	Auto	normal Modbus	100
101 (40102)	120 (40121)	20	MBus_OutTlg	-manual-	-manual-	normal Modbus	
400 (40401)	401 (40402)	2	MBus_ComStat	-manual-	-manual-	normal Modbus	
402 (40403)	402 (40403)	1	MBus_Chars	-manual-	-manual-	normal Modbus	
403 (40404)	492 (40493)	90	MBus_InTlg0	-manual-	-manual-	15/16 only	
493 (40494)	582 (40583)	90	MBus_InTlg1	-manual-	-manual-	15/16 only	
583 (40584)	663 (40664)	81	MBus_InTlg2	-manual-	-manual-	15/16 only	

Obr. 5 – Definice MODBUS registrů

Pouze u registru 100 se bude využívat automatická priorita zápisu, komunikace ostatních registrů bude volána z algoritmu moduly **MdbmMark**. Registr 100 se rovněž využije pro zjišťování stavu

komunikace s převodníkem pomocí modulu **MdbmReqSt**, proto se tomuto registru nadefinuje návěští.

Více informací o komunikačních definicích **ModbusMaster** a **ModbusDevice** a modulech **MdbmMark** a **MdbmReqSt** lze nalézt v nápovědě vývojového prostředí **DetStudio**.

Jakmile je vytvořena definice MODBUS registrů, lze přistoupit k tvorbě komunikačního algoritmu. Pro komunikaci prostřednictvím komunikačního protokolu MODBUS s převodníkem **DM-MBUS64** lze doporučit následující postup:

1. uložení odchozího rámce do převodníku v závislosti, jestli se jedná o inicializační rámec či požadavek na běžná data,
2. pauza pro provedení komunikace,
3. zapsání do registru akce převodníku hodnotu 1 pro vyslání rámce do sítě M-Bus,
4. pauza pro provedení komunikace,
5. vyčtení hodnoty registru stavu příchozího rámce,
6. pauza pro provedení komunikace,
7. zjištění stavu příchozího rámce; možné varianty:
 - a) příchozí rámec zatím nepřišel, proto opakovat od bodu 4.,
 - b) příchozí rámec přišel, proto provést vyčtení hodnot rámce z převodníku a počtu příchozích bajtů a pokračovat v bodu 8.,
 - c) převodník hlásí chybu, proto se přejde do speciálního bodu, ve kterém se řeší opakování rámce,
8. pauza pro provedení komunikace,
9. zpracování hodnot,
10. zapsání do registru akce převodníku dekadickou hodnotu 203 pro promazání bufferu příchozího rámce ze sítě M-Bus a vynulování stavu a případné chyby komunikace,
11. pauza pro provedení komunikace,
12. na základě přijatých bytů z převodníku:
 - a) pokud se přijal neplatný byte odpovědi na inicializační rámec, přejít do speciálního bodu, ve kterém se řeší opakování rámce,
 - b) pokud se přijal platný byte odpovědi na inicializační rámec, opakovat od bodu 1. se změněným hlavním bytem pro požadavek na běžná data,
 - c) pokud se přijala běžná data nebo není dlouhodobě odpověď na inicializační rámec, pokračovat v kroku 13.,
13. pokud je další měřič ke komunikaci, změnit adresu/index adresy a opakovat od bodu 1, v opačném případě ukončit komunikační algoritmus.

Ve speciálním bodu (řešícím chyby či problémy) provést inkrementaci čítače problémů a pokud je menší než typicky hodnota 3, provede se opětovné poslání rámce do sítě M-Bus (ve výše uvedeném číslování se jedná o bod 3.) Pokud je hodnota čítače problémů větší než typicky 3, je ukončena komunikace s daným měřičem, zaznamenána chyba a provede se vymazání bufferu příchozího rámce (ve výše uvedeném číslování se jedná o bod 10).

3 Ukázkové aplikace

Součástí této aplikační poznámky jsou ukázkové aplikace pro komunikaci s měřiči energií některých výrobců. Dále jsou obsaženy komunikační jádra pro použití při komunikaci pomocí modulů uživatelské komunikace a protokolem MODBUS. Seznam příkladů je uveden v tabulce „Příloha“ na čelní stránce této aplikační poznámky.

Pozor

Jedná se o ukázkové příklady, ve kterých je pouze naznačeno, jakým způsobem lze vyčítat a dekodovat některé hodnoty z měřičů. Nejedná se o příklady, ve kterých je zpracování všech informací, které dané měřiče poskytují.

4 Technická podpora

Veškeré informace ohledně komunikace řídicích systémů firmy AMiT v síti M-Bus, Vám poskytne oddělení technické podpory firmy AMiT. Technickou podporu můžete kontaktovat nejlépe prostřednictvím emailu na adrese **support@amit.cz**.

5 Upozornění

AMiT, spol. s r. o. poskytuje informace v tomto dokumentu, tak jak jsou, nepřijímá žádné záruky, pokud se týče obsahu tohoto dokumentu a vyhrazuje si právo měnit obsah dokumentu bez závazku tyto změny oznámit jakékoli osobě či organizaci.

Tento dokument může být kopírován a rozšiřován za následujících podmínek:

1. Celý text musí být kopírován bez úprav a se zahrnutím všech stránek.
2. Všechny kopie musí obsahovat označení autorského práva společnosti AMiT, spol. s r. o. a veškerá další upozornění v dokumentu uvedená.
3. Tento dokument nesmí být distribuován za účelem dosažení zisku.

V publikaci použité názvy produktů, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.