

Komunikace AMREG v síti MODBUS TCP

Abstrakt

Parametrizace regulátorů AMREG komunikujících v síti MODBUS TCP jako master/slave.

Autor: Petr Latina, Zbyněk Říha
Dokument: ap0057_cz_01.pdf

Příloha

Obsah souboru: ap0057_cz_01.zip

master_tcp_p1_cz_01.dso	Příklad parametrizace AMREG – master.
slave_tcp_p2_cz_01.dso	Příklad parametrizace AMREG – slave.

Obsah

	Obsah	2
	Historie revizí	3
	Související dokumentace.....	3
1	Definice použitých pojmů	4
2	AMREG – master	5
2.1	SW parametrizace	5
2.1.1	Komunikační objekty	5
2.1.2	Definice datových bodů	7
2.2	Ukázková aplikace pro AMREG – master.....	9
2.2.1	Význam / použití nadefinovaných datových bodů	10
2.3	Stav komunikace	14
3	AMREG – slave.....	15
3.1	SW parametrizace	15
3.1.1	Komunikační objekt.....	15
3.1.2	Definice datových bodů	15
3.2	Ukázková aplikace pro AMREG – slave	18
3.2.1	Význam / použití nadefinovaných datových bodů	18
3.3	Stav komunikace	22
4	Technická podpora	23
5	Upozornění	24

Historie revizí

Verze	Datum	Autor změny	Změny
100	29. 08. 2016	Latina Petr Říha Zbyněk	Nový dokument

Související dokumentace

1. Návod k vývojovému prostředí DetStudio
soubor: Ovladani_cs.chm
2. Návod k obrazovkám vývojového prostředí DetStudio
soubor: Tridet_cs.chm
3. Návod k části EsiDet vývojového prostředí DetStudio
soubor: Esidet_cs.chm
4. Aplikační poznámka AP0037 – Zásady používání sítě Ethernet
soubor: ap0037_cz_xx.pdf
5. www.modbus.org – protokol MODBUS TCP

1 Definice použitých pojmů

Master

Zařízení, které aktivně zasílá dotazy / pokyny jednotlivým perifériím v síti. V roli mastera je typicky nadřazený systém.

Slave

Zařízení, která do sítě aktivně nezasílají žádné dotazy ani pokyny. Na přijaté dotazy / pokyny pouze odpovídají (v případě, že jsou mu adresovány). V roli slave jsou ovládaná / sledovaná zařízení.

Datový bod

Hodnota přenášená prostřednictvím protokolu MODBUS TCP. V rámci této aplikační poznámky se pojmem datový bod rozumí uchovávací (holding) registry, vstupní (input) registry, cívky (coils) a diskrétní vstupy (discrete inputs). Každý druh datových bodů má vlastní, nezávislou řadu adres začínajících od 0 (viz popis protokolu MODBUS TCP).

Registr

16 bitová hodnota, přenášená prostřednictvím sítě MODBUS TCP.

2 AMREG – master

Jako master v síti MODBUS TCP lze definovat všechny regulátory AMREG, které disponují ethernetovým rozhraním.

V této aplikační poznámce bude jako master v síti MODBUS TCP použit regulátor **AMR-OP84**.

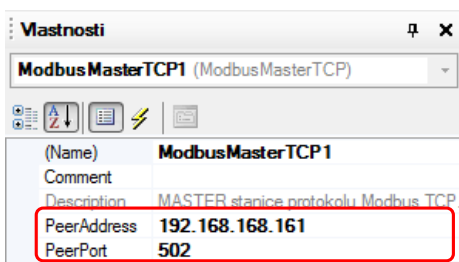
2.1 SW parametrizace

2.1.1 Komunikační objekty

Pro parametrizaci regulátoru AMREG komunikujícího jako Master v síti MODBUS TCP je nutné vložit do projektu objekty:

- ♦ **ModbusMasterTCP** (definuje v regulátoru AMREG komunikaci s jedním slave zařízením v síti MODBUS TCP),
- ♦ **ModbusDeviceTCP** (definuje jedno obecné slave zařízení, se kterými má regulátor AMREG komunikovat).

V panelu „Vlastnosti“, objektu **ModbusMasterTCP**, se nastaví položky, označené na obrázku Obr. 1.



Obr. 1 – Panel „Vlastnosti“ objektu **ModbusMasterTCP**

Význam položek, označených na Obr. 1:

PeerAddress

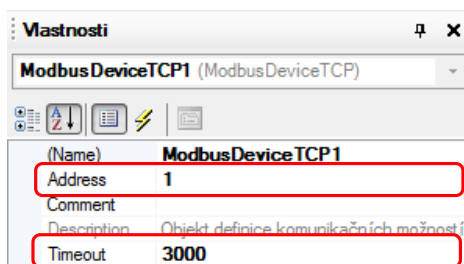
IP adresa slave zařízení, se kterým má regulátor AMREG komunikovat.

PeerPort

TCP port, na kterém slave zařízení komunikuje prostřednictvím MODBUS TCP.

Pro každé zařízení slave (které má v síti MODBUS TCP svou vlastní IP adresu), se kterým má regulátor AMREG komunikovat, je nutné do projektu vložit jeden objekt **ModbusMasterTCP** a jeden objekt **ModbusDeviceTCP**.

V panelu „Vlastnosti“ každého objektu **ModbusDeviceTCP**, se nastaví položky, označené na obrázku Obr. 2.



Obr. 2 – Panel vlastnosti objektu **ModbusDeviceTCP**

Význam položky označené na Obr. 2:

Address

Adresa slave zařízení se kterým bude regulátor AMREG komunikovat.

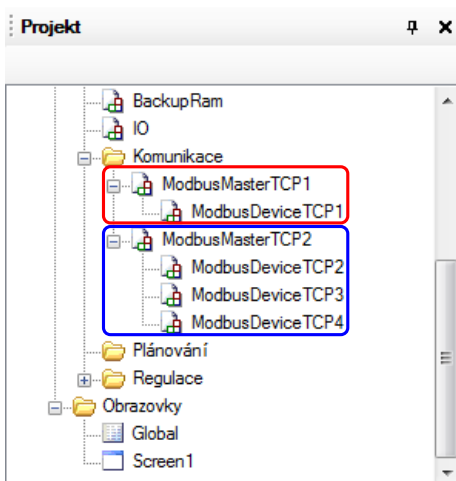
Timeout

Čas [ms], jak dlouho se čeká na komunikační rámec, než se zvýší počítadlo **ErrMessages** nadřazeného objektu **ModbusMasterTCP** (viz nápověda k části EsiDet prostředí DetStudio).

Aby fungovala komunikace v síti MODBUS TCP správně, musí mít každá stanice v síti nastaven komunikační port a jedinečnou IP adresu, prostřednictvím které komunikuje. Každý slave musí mít nastavenou jedinečnou adresu v síti MODBUS TCP. IP adresu lze nastavit prostřednictvím displeje. Komunikační port master zařízení v síti MODBUS TCP lze nastavit pomocí skriptu EsiDetu, např. v procesu **ProcessInit**, stejně tak adresu slave zařízení v síti MODBUS TCP (viz nápověda k části EsiDet prostředí DetStudio).

Více informací o nastavení komunikačních parametrů regulátorů AMREG lze nalézt v nápovědě EsiDet.

Pokud je jedno slave zařízení v síti MODBUS TCP ve funkci brány pro více slave zařízení komunikujících v síti MODBUS RTU, je nutné do projektu vložit jeden objekt **ModbusMasterTCP**, který bude definovat slave zařízení ve funkci brány a tolik objektů **ModbusDeviceTCP**, kolik je k bráně připojeno slave zařízení, komunikujících v síti MODBUS RTU (viz Obr. 3).

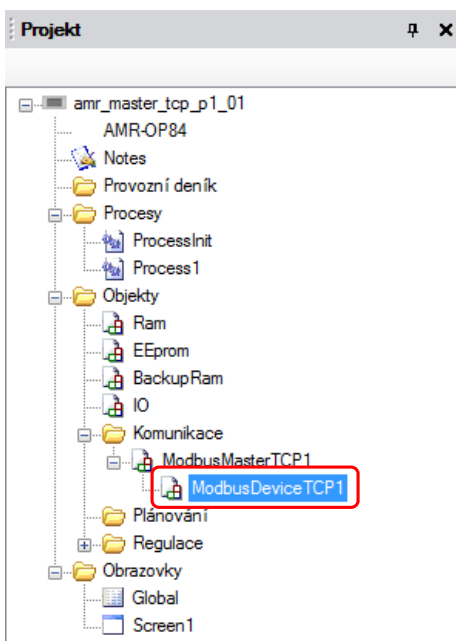


Obr. 3 – Použití objektů **ModbusMasterTCP** v kombinaci s objekty **ModbusDeviceTCP**

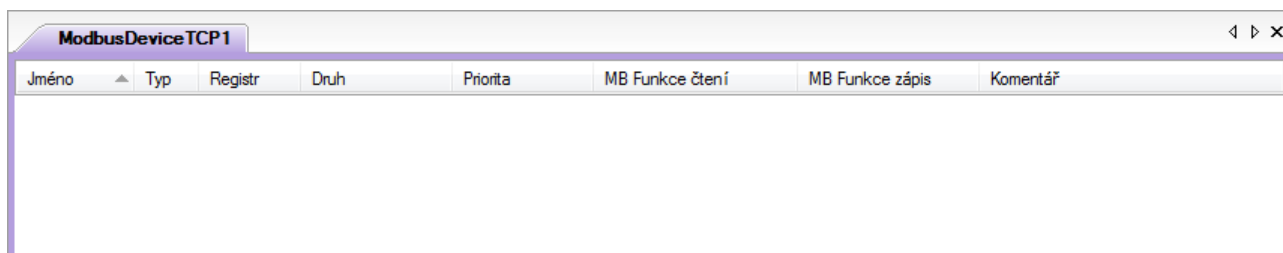
Na výše uvedeném obrázku je červeně zvýrazněna definice jednoho slave zařízení v síti MODBUS TCP, které má svou vlastní IP adresu. Modře je pak zvýrazněna definice jednoho slave zařízení, které je ve funkci brány do sítě MODBUS RTU a 3 zařízení, které komunikují prostřednictvím MODBUS RTU a jsou připojena k bráně.

2.1.2 Definice datových bodů

Datové body, které se mají s jednotlivými slave zařízeními komunikovat se definují v otevřené záložce objektu **ModbusDeviceTCP**. Záložka se otevře dvojklikem myši nad konkrétním objektem **ModbusDeviceTCP** v panelu „Projekt“. Viz Obr. 4.



Obr. 4 – Vybraný objekt **ModbusDeviceTCP1**

Obr. 5 – Otevřená záložka objektu **ModbusDeviceTCP1**

K datu vytvoření této aplikační poznámky lze definovat následující datové body:

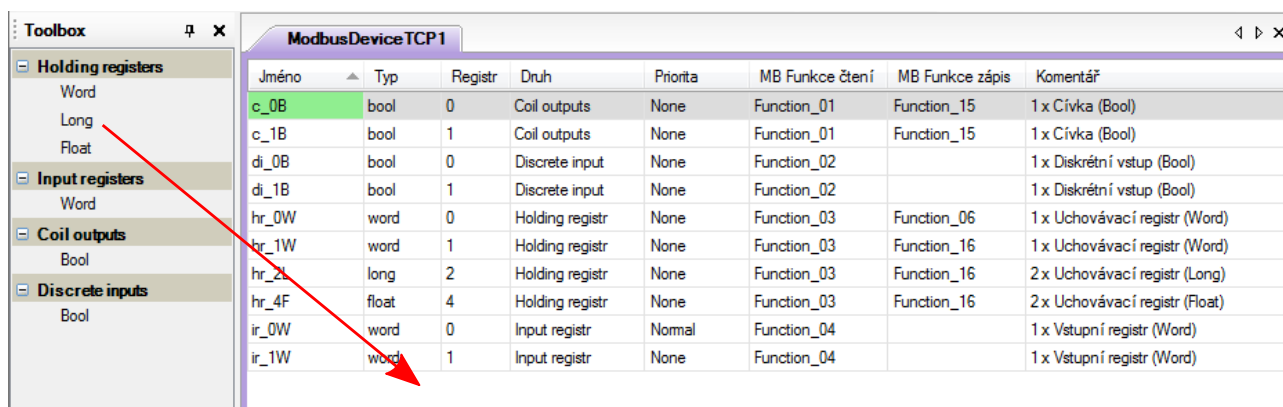
- ♦ Uchovávací registry (holding registers) typu Word, Long, Float *),
- ♦ Vstupní registry (input registers) typu Word,
- ♦ Cívky (coils) typu Bool,
- ♦ Diskrétní vstupy (discrete inputs) typu Bool.

*) Typy Long a Float jsou ukládány do dvojice po sobě jdoucích registrů.

Podporované funkce pro komunikaci v síti MODBUS TCP jsou:

Funkce	Význam
1	Read coils – čtení stavu cívek
2	Read discrete inputs – čtení diskretních vstupů
3	Read multiple holding registers – čtení uchovávacích registrů
4	Read input registers – čtení vstupních registrů
5	Write single coil – zápis do jedné cívky
6	Write single holding register – zápis do jednoho uchovávacího registru
15	Write multiple coils – zápis do více cívek
16	Write multiple holding registers – zápis do více uchovávacích registrů

Jednotlivé datové body se definují přetažením požadovaného typu pomocí myši z panelu „Toolbox“ do otevřené záložky objektu **ModbusDeviceTCP1**, viz následující obrázek.

Obr. 6 – Definice datových bodů v záložce objektu **ModbusDeviceTCP1**

Význam jednotlivých sloupců záložky objektu **ModbusDeviceTCP1** je následující:

Jméno

Jméno datového bodu, pod kterým bude využíván v rámci projektu. Lze editovat pomocí klávesy „F2“.

Typ

Typ datového bodu (Word, Long, ...). Nelze editovat – předvyplněn po vložení datového bodu z panelu „Toolbox“.

Registr

Adresa datového bodu na zařízení typu slave. Lze editovat např. pomocí klávesy „F2“. Datové body Bool a Word lze definovat v číselném pořadí např. 100, 101, 102, 103 atd. Datové body typu Long a Float lze definovat jako dvojici uchovávacích registrů.

Druh

Druh datového bodu (uchovávací registr, cívka, ...). Nelze editovat – předvyplněn po vložení datového bodu z panelu „Toolbox“.

Priorita

Priorita komunikace. Jednotlivým datovým bodům lze nastavit (klávesou „F2“) následující úroveň:

- ♦ **None** – v tomto případě se požadavek na komunikaci datového bodu aktivuje na událost, kterou si programově řídí tvůrce kódu ve skriptu procesu, nebo se požadavek na komunikaci aktivuje v případě, kdy daný regulátor má displej, kde je umístěn prvek, který umožňuje zobrazit/editovat hodnotu navázané skalární proměnné/buňky matice. V tomto případě je perioda nastavení požadavku na komunikaci odvozena od vlastnosti **RefreshPeriod** zobrazené obrazovky na displeji (viz text níže).
- ♦ **Low** – automatické vkládání požadavků. Jestliže je v projektu definován proces s periodou větší jak 5000 ms, použije se perioda tohoto procesu. V ostatních případech je použita perioda 5000 ms.
- ♦ **Normal** – automatické vkládání požadavků s periodou 900 ms.
- ♦ **High** – automatické vkládání požadavků. Jestliže je v projektu definován proces s periodou menší než 200 ms, použije se perioda tohoto procesu. V ostatních případech je použita perioda 200 ms.

MB Funkce čtení

Číslo MODBUS funkce, která bude využívána při čtení datového bodu. Nelze editovat – je nastaveno automaticky, v závislosti na druhu datového bodu.

MB Funkce zápis

Číslo MODBUS funkce, která bude využívána pro zápis do datového bodu. Lze editovat pomocí klávesy „F2“.

Komentář

Poznámka k datovému bodu. Lze editovat např. pomocí klávesy „F2“.

2.2 Ukázková aplikace pro AMREG – master

Součástí této aplikační poznámky je ukázková aplikace pro **AMR-OP84**. Jedná se o soubor **master_tcp_p1_cz_xx.dso**. V aplikaci jsou výše uvedeným postupem nadefinovány datové body dle Obr. 7. Tyto registry mapuje do sítě MODBUS TCP slave zařízení – **AMR-OP84** z produkce firmy AMIT.

ModbusDeviceTCP1							
Jméno	Typ	Registr	Druh	Priorita	MB Funkce čtení	MB Funkce zápis	Komentář
c_0B	bool	0	Coil outputs	None	Function_01	Function_15	1 x Cívka (Bool)
c_1B	bool	1	Coil outputs	None	Function_01	Function_15	1 x Cívka (Bool)
di_0B	bool	0	Discrete input	None	Function_02		1 x Diskrétní vstup (Bool)
di_1B	bool	1	Discrete input	None	Function_02		1 x Diskrétní vstup (Bool)
hr_0W	word	0	Holding registr	None	Function_03	Function_06	1 x Uchovávací registr (Word)
hr_1W	word	1	Holding registr	None	Function_03	Function_16	1 x Uchovávací registr (Word)
hr_2L	long	2	Holding registr	None	Function_03	Function_16	2 x Uchovávací registr (Long)
hr_4F	float	4	Holding registr	None	Function_03	Function_16	2 x Uchovávací registr (Float)
ir_0W	word	0	Input registr	Normal	Function_04		1 x Vstupní registr (Word)
ir_1W	word	1	Input registr	None	Function_04		1 x Vstupní registr (Word)

Obr. 7 – Nadefinované datové body

2.2.1 Význam / použití nadefinovaných datových bodů

Cívky c_0B a c_1B

Cívky **c_0B** a **c_1B** mají nastavenou prioritu typu **None**. Cívka **c_0B** je navázána na prvek „BitSwitchUpDn“ obrazovky „Screen1“, která má v panelu „Vlastnosti“ nastavenou vlastnost **RefreshPeriod** na hodnotu 1000 ms. Je-li tato obrazovka aktuálně zobrazena, budou se s periodou 1000 ms vkládat požadavky na její čtení.

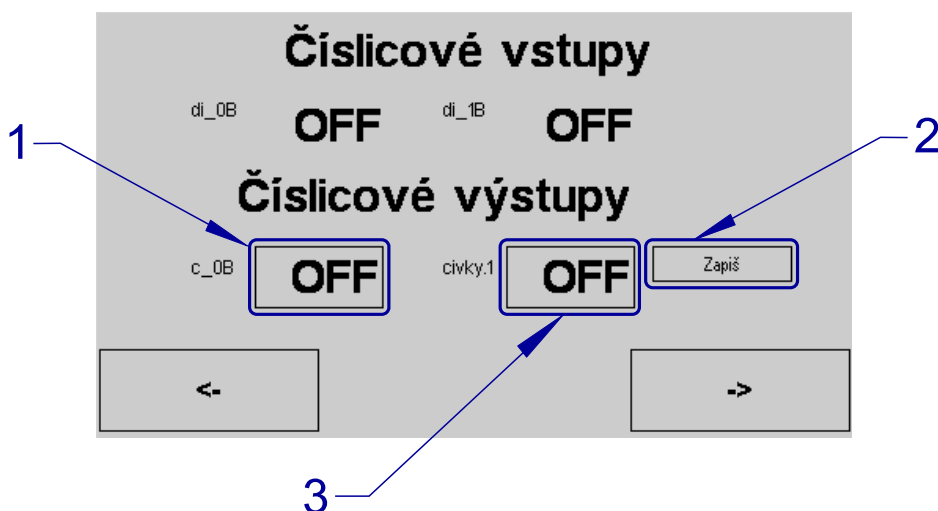
Obr. 8 – Hodnota vlastnosti **RefreshPeriod** obrazovky „Screen1“

Cívka **c_1B** je použita pro událostní zápis, který se vykoná na nastavení bitu č. 0 lokální proměnné **Ram.zapis**. Jakmile bude nastaven bit č. 0 proměnné **Ram.zapis**, bude vykonán ve skriptu procesus kód, který vyvolá požadavek na zápis stavu bitu č. 1 proměnné **Ram.civky** do cívky **c_1B**.

```

If Ram.zapis.0 then
    Ram.zapis.0 = false;
    ModbusDeviceTCP1.c_1B = Ram.civky.1;
endif;

```

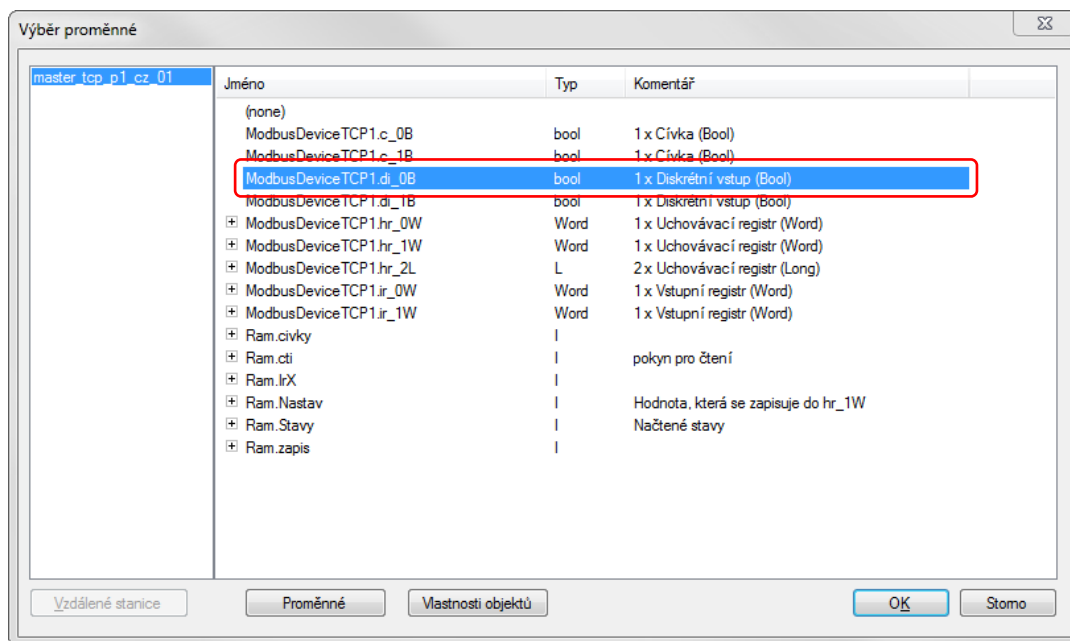


Obr. 9 – Popis položek obrazovky „Screen1“

Legenda

Číslo	Význam
1	Stav cívky c_0B
2	Tlačítko pro zápis stavu bitu č. 1 proměnné Ram.civky
3	Stav bitu č. 1 proměnné Ram.civky

Navázání datových bodů/proměnných na obrazovkové prvky pro zobrazení/editaci hodnoty lze provést např. tak, že v návrhu obrazovky se dvojklikem myši nad vybraným prvkem otevře okno „Výběr proměnné“, ve kterém se vybere požadovaný datový bod (proměnná) a následně se potvrdí tlačítkem „OK“, viz následující obrázek.



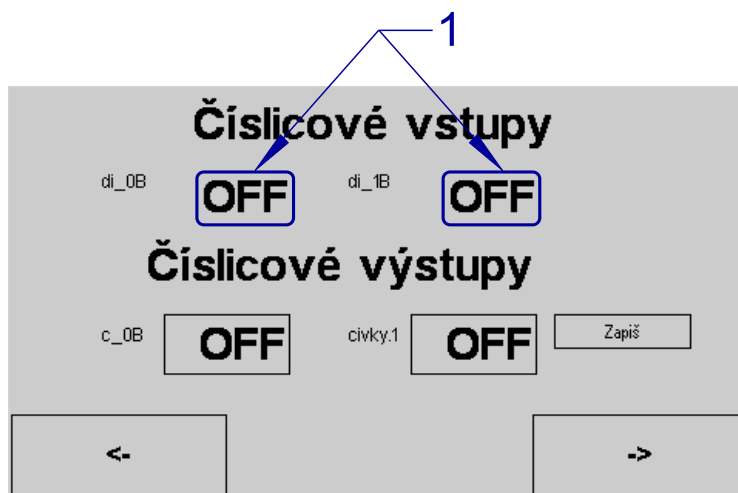
Obr. 10 – Okno „Výběr proměnné“

Poznámka

Pokud jsou datové body navázány pouze na obrazovkové prvky typu „BitSwitchUpDn“ a nejsou využívány pro regulační algoritmy v některém z periodických procesů, doporučujeme v takovém případě nastavit vždy prioritu **None**.

Diskrétní vstupy di_0B a di_1B

Diskrétní vstupy di_0B a di_1B0 mají nastavenou prioritu typu **None**. Jsou navázány na prvky „BitSwitchView“ obrazovky „Screen1“, která má v panelu „Vlastnosti“ nastavenou vlastnost **RefreshPeriod** na hodnotu 1000 ms. Je-li tato obrazovka aktuálně zobrazena, budou se s periodou 1000 ms vkládat požadavky na jejich čtení.



Obr. 11 – Obrazovka „Screen1“

Legenda

Číslo	Význam
1	Stavy diskretních vstupů di_xB

Poznámka

Pokud jsou datové body navázány pouze na obrazovkové prvky typu „BitSwitchView“ a nejsou využívány pro regulační algoritmy v některém z periodických procesů, doporučujeme v takovém případě nastavit vždy prioritu **None**.

Vstupní registry ir_0W a ir_1W

Registry ir_0W má nastavenou prioritu typu **Normal**. Automaticky se tedy bude komunikovat s periodou 900 ms. Je použit ve skriptu procesu **Process1**. Následující příklad zápisu ukazuje přiřazení hodnoty registru do lokální proměnné vytvořené v paměťovém prostoru **Ram**.

```
Ram.Stavy = ModbusDeviceTCP1.ir_0W;
```

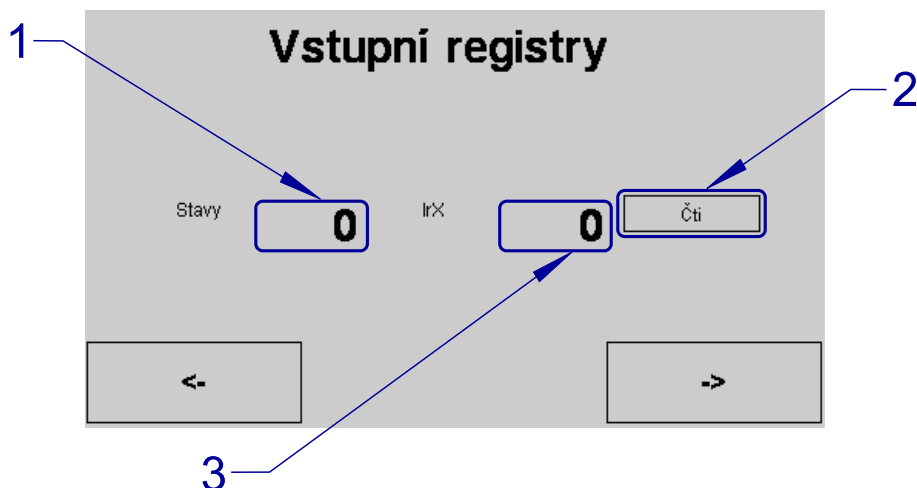
Registr ir_1W je v ukázce využit k událostnímu čtení (má nastavenou prioritu typu **None**), které se vykoná po nastavení bitu č. 0 lokální proměnné **Ram.cti**. Jakmile bude nastaven bit č. 0 proměnné **Ram.cti**, bude vykonán ve skriptu procesu kód, který vyvolá požadavek na čtení registru. V následujícím běhu procesu bude do lokální proměnné **Ram.IrX** uložena hodnota načteného registru.

```

if Ram.cti.0 then
    Ram.cti.0 = false;
    ModbusDeviceTCP1.ir_1W.Refresh();
else
    Ram.Reg_IrX = ModbusDeviceTCP1.ir_1W;
endif;

```

Hodnoty registrů jsou zobrazeny na obrazovce „Screen2“.



Obr. 12 – Obrazovka „Screen2“

Legenda

Číslo	Význam
1	Hodnota proměnné Ram. Stavy
2	Tlačítko pro nastavení bitu č. 0 proměnné Ram.cti
3	Hodnota proměnné Ram. IrX

Poznámka

*Pokud jsou datové body navázány pouze na obrazkové prvky typu „NumericView“ a nejsou využívány pro regulační algoritmy v některém z periodických procesů, doporučujeme v takovém případě nastavit vždy prioritu **None**.*

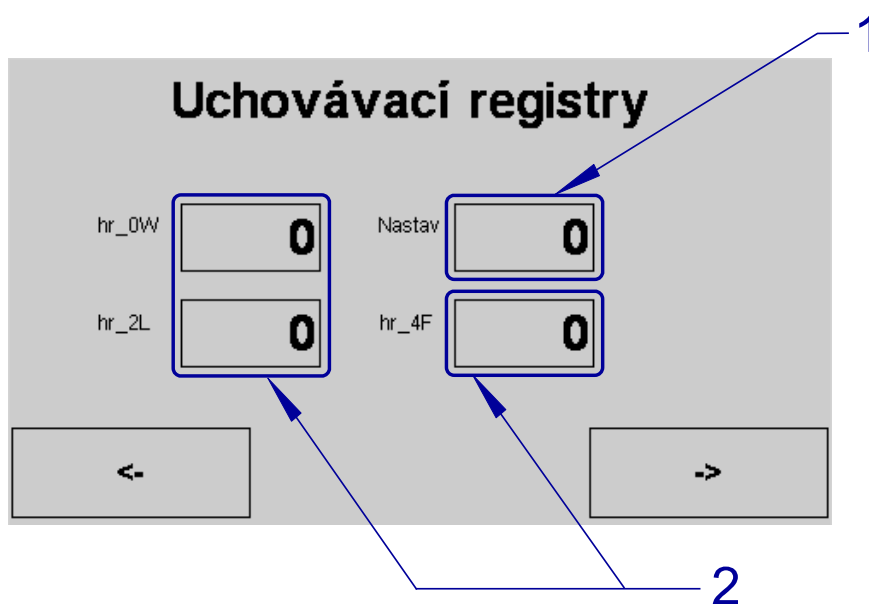
Uchovávací registry hr_0W, hr_1W, hr_2L a hr_4F

Uchovávací registry **hr_0W**, **hr_1W**, **hr_2L** a **hr_4F** mají nastavenou prioritu typu **None**. Registry **hr_0W**, **hr_2L** a **hr_4F** jsou navázány na prvky „NumericEdit“ obrazovky „Screen3“, která má v panelu „Vlastnosti“ nastavenou vlastnost **RefreshPeriod** na hodnotu 1000 ms. Je-li tato obrazovka aktuálně zobrazena, budou se s periodou 1000 ms vkládat požadavky na její čtení.

Registr **hr_1W** je ve skriptu procesu **Process1** plněn hodnotou proměnné **Ram.Nastav** následovně.

```
ModbusDeviceTCP1.hr_1W = Ram.Nastav;
```

Na obrazovce „Screen3“ se tedy nastavuje hodnota proměnné **Ram.Nastav**.



Obr. 13 – Obrazovka „Screen3“

Legenda

Číslo	Význam
1	Hodnota proměnné Ram.Nastav
2	Hodnoty registrů hr_0 , hr_2L a hr_4F

Poznámka

*Pokud jsou datové body navázány pouze na obrazovkové prvky typu „NumericEdit“ a nejsou využívány pro regulační algoritmy v některém z periodických procesů, doporučujeme v takovém případě nastavit vždy prioritu **None**.*

2.3 Stav komunikace

Pro vyhodnocení stavu komunikace lze v regulátoru AMREG využít několik vlastností objektu **ModbusMasterTCP**, které lze navázat např. na obrazovkové prvky nebo je vyhodnocovat v některém z periodických procesů.

Jedná se o následující vlastnosti:

- ♦ **ErrMessages**,
- ♦ **FailConnections**,
- ♦ **OKConnections**,
- ♦ **OKMessages**,
- ♦ **TCPStatus**.

Více informací lze nalézt v nápovědě EsiDet u objektu **ModbusMasterTCP**.

3 AMREG – slave

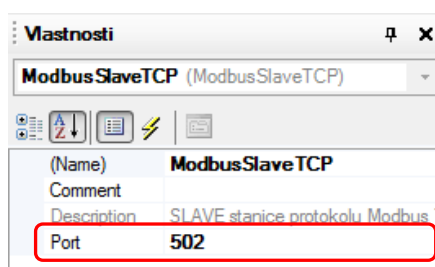
Jako slave v síti MODBUS TCP lze definovat všechny regulátory AMREG, které disponují ethernetovým rozhraním.

V této aplikační poznámce bude jako slave v síti MODBUS TCP použit regulátor **AMR-OP84**.

3.1 SW parametrizace

3.1.1 Komunikační objekt

Pro parametrizaci regulátoru AMREG komunikujícího v síti MODBUS TCP jako slave je nutné do projektu vložit objekt **ModbusSlaveTCP**. V panelu „Vlastnosti“ tohoto objektu se nastaví pouze položka **Port**, označená na Obr. 14.



Obr. 14 – Panel „Vlastnosti“ objektu **ModbusSlaveTCP**

Význam položky, označené na Obr. 14:

Port

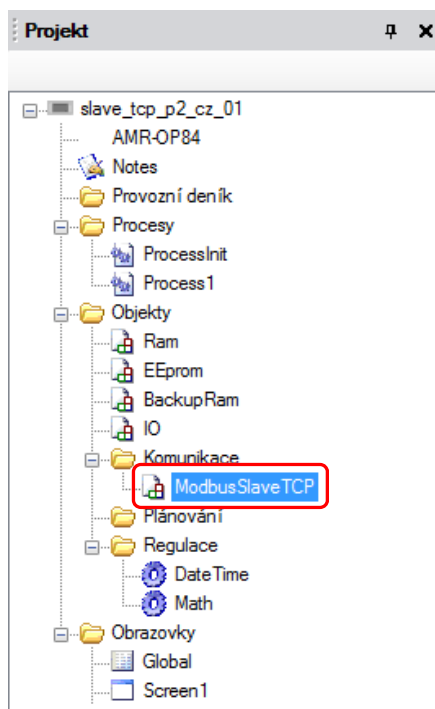
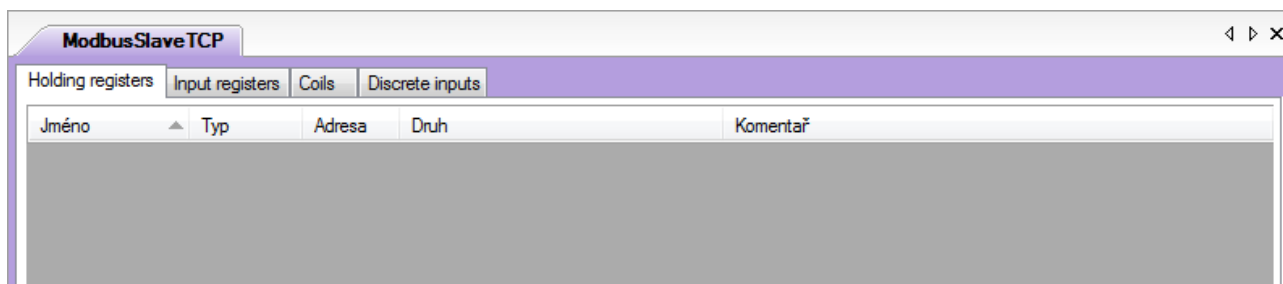
TCP port, na kterém regulátor AMREG komunikuje prostřednictvím MODBUS TCP.

Aby fungovala komunikace v síti MODBUS TCP správně, musí mít každá stanice v síti nastaven komunikační port a jedinečnou IP adresu, prostřednictvím které komunikuje. Každý slave musí mít nastavenou jedinečnou adresu v síti MODBUS TCP. IP adresu lze nastavit prostřednictvím displeje. Komunikační port master zařízení v síti MODBUS TCP lze nastavit pomocí skriptu EsiDetu, např. v procesu **ProcessInit**, stejně tak adresu slave zařízení v síti MODBUS TCP (viz nápověda k části EsiDet prostředí DetStudio).

Více informací o nastavení komunikačních parametrů regulátorů AMREG lze nalézt v nápovědě EsiDet.

3.1.2 Definice datových bodů

Datové body se definují v otevřené záložce objektu **ModbusSlaveTCP**. Záložka se otevře dvojklikem levým tlačítkem myši nad objektem **ModbusSlaveTCP** v panelu „Projekt“. Viz následující obrázek.

Obr. 15 Vybraný objekt **ModbusSlaveTCP**Obr. 16 Otevřená záložka objektu **ModbusSlaveTCP**

K datu vytvoření této aplikační poznámky lze definovat následující datové body:

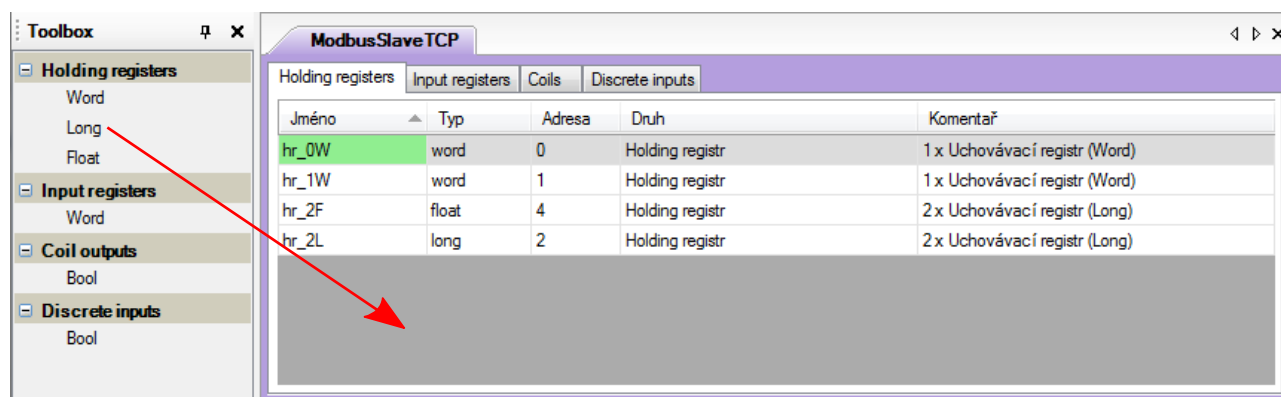
- ♦ Uchovávací registry (holding registers) typu Word, Long, Float *),
- ♦ Vstupní registry (input registers) typu Word,
- ♦ Cívky (coils) typu Bool,
- ♦ Diskrétní vstupy (discrete inputs) typu Bool.

*) Typy Long a Float jsou ukládány do dvojice po sobě jdoucích registrů.

Podporované funkce pro komunikaci v síti MODBUS TCP jsou:

Funkce	Význam
1	Read coils – čtení stavu cívek
2	Read discrete inputs – čtení diskretních vstupů
3	Read multiple holding registers – čtení uchovávacích registrů
4	Read input registers – čtení vstupních registrů
5	Write single coil – zápis do jedné cívky
6	Write single holding register – zápis do jednoho uchovávacího registru
15	Write multiple coils – zápis do více cívek
16	Write multiple holding registers – zápis do více uchovávacích registrů

Jednotlivé datové body se definují přetažením požadovaného typu pomocí myši z panelu „Toolbox“ do otevřené záložky objektu **ModbusSlave**, viz následující obrázek.



Obr. 17 – Definice datových bodů v záložce objektu **ModbusSlaveTCP**

Význam jednotlivých sloupců záložky objektu **ModbusSlave** je následující:

Jméno

Jméno datového bodu, pod kterým bude využíván v rámci projektu. Lze editovat pomocí klávesy „F2“.

Typ

Typ datového bodu (Word, Long, ...). Nelze editovat – předvyplněn po vložení datového bodu z panelu „Toolbox“.

Adresa

Adresa datového bodu na zařízení typu slave. Lze editovat např. pomocí klávesy „F2“. Datové body Bool a Word lze definovat v číselném pořadí např. 100, 101, 102, 103 atd. Datové body typu Long a Float lze definovat jako dvojici uchovávacích registrů.

Druh

Druh datového bodu (uchovávací registr, cívka, ...). Nelze editovat – předvyplněn po vložení datového bodu z panelu „Toolbox“.

Komentář

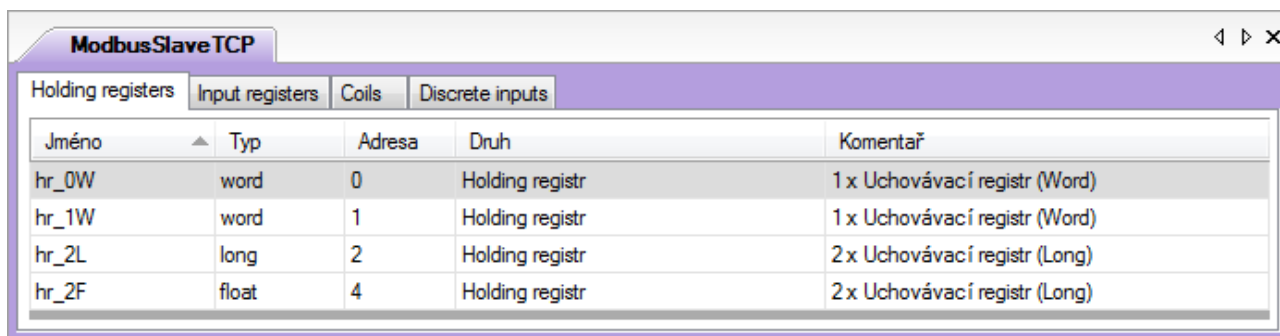
Poznámka k datovému bodu. Lze editovat např. pomocí klávesy „F2“.

3.2 Ukázková aplikace pro AMREG – slave

Součástí této aplikační poznámky je ukázková aplikace pro regulátor AMREG jako slave. Jedná se o soubor **slave_tcp_p2_cz_xx.dso**. V aplikaci jsou výše uvedeným postupem definovány různé datové body.

3.2.1 Význam / použití nadefinovaných datových bodů

Uchovávací registry



ModbusSlaveTCP				
Holding registers				
Jméno	Typ	Adresa	Druh	Komentář
hr_0W	word	0	Holding registr	1 x Uchovávací registr (Word)
hr_1W	word	1	Holding registr	1 x Uchovávací registr (Word)
hr_2L	long	2	Holding registr	2 x Uchovávací registr (Long)
hr_2F	float	4	Holding registr	2 x Uchovávací registr (Long)

Obr. 18 – Uchovávací registry

Hodnoty uchovávacích registrů jsou nastavovány masterem v síti MODBUS TCP. Zároveň je lze nastavovat také z displeje regulátoru.

Uchovávací registry

hr_0W	<input type="text" value="0"/>	hr_1W	<input type="text" value="0"/>
hr_2L	<input type="text" value="0"/>	hr_4F	<input type="text" value="0"/>

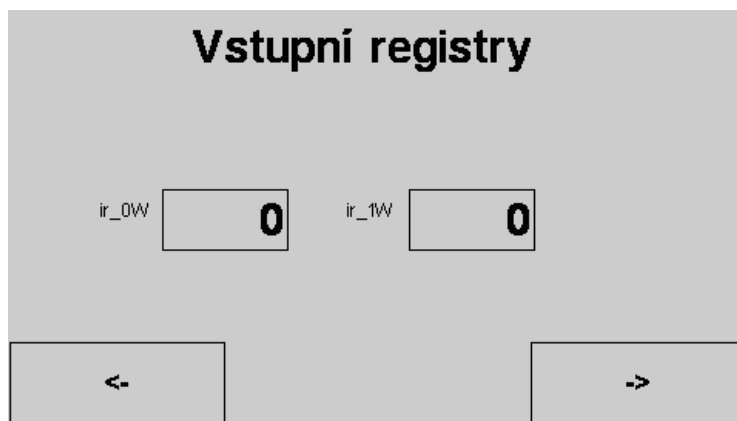
Obr. 19 – Obrazovka „Screen3“

Vstupní registry

ModbusSlaveTCP				
Holding registers	Input registers	Coils	Discrete inputs	
Jméno	Typ	Adresa	Druh	Komentář
ir_0W	word	0	Input registr	1 x Vstupní registr (Word)
ir_1W	word	1	Input registr	1 x Vstupní registr (Word)

Obr. 20 – Vstupní registry

Hodnoty vstupních registrů jsou čteny masterem v síti MODBUS TCP. Zároveň je lze nastavovat z displeje regulátoru.



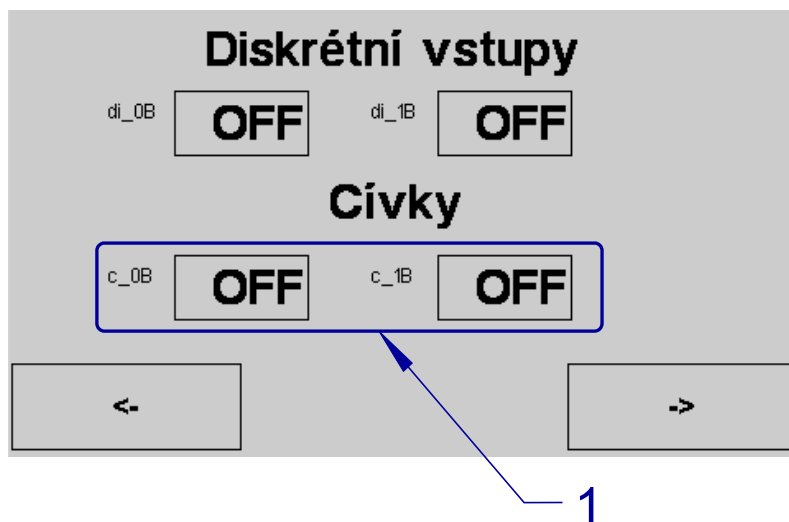
Obr. 21 – Obrazovka „Screen2“

Cívky

ModbusSlaveTCP				
Holding registers Input registers Coils Discrete inputs				
Jméno	Typ	Adresa	Druh	Komentář
c_0B	bool	0	Coil	1 x Cívka (Bool)
c_1B	bool	1	Coil	1 x Cívka (Bool)

Obr. 22 – Cívky

Stavy cívek jsou nastavovány masterem v síti MODBUS TCP. Zároveň je lze nastavovat také z displeje regulátoru.



Obr. 23 – Obrazovka „Screen1“

Legenda

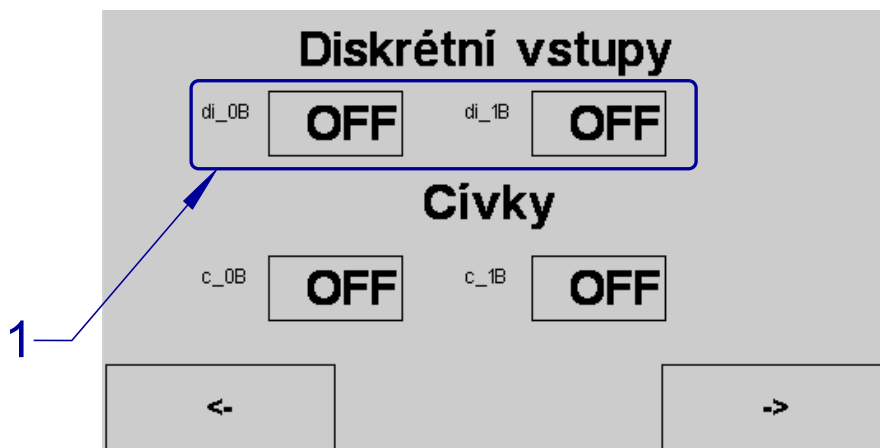
Číslo	Význam
1	Stavy cívek c_0B a c_1B

Diskrétní vstupy

ModbusSlaveTCP				
Holding registers	Input registers	Coils	Discrete inputs	
Jméno	Typ	Adresa	Druh	Komentář
di_0B	bool	0	Discrete input	1 x Diskrétní vstup (Bool)
di_1B	bool	1	Discrete input	1 x Diskrétní vstup (Bool)

Obr. 24 – Diskrétní vstupy

Stavy diskrétních vstupů jsou čteny masterem v síti MODBUS TCP. Zároveň je lze nastavovat z displeje regulátoru.



Obr. 25 – Obrazovka „Screen1“

Legenda

Číslo	Význam
1	Stavy diskrétních vstupů di_0B a di_1B

3.3 Stav komunikace

Pro vyhodnocení stavu komunikace lze v regulátoru AMREG využít několik vlastností objektu **ModbusSlave**, které lze navázat např. na obrazovkové prvky nebo je vyhodnocovat v některém z periodických procesů.

Jedná se o následující vlastnosti:

- ♦ **Connected**,
- ♦ **Incoming**,
- ♦ **Outgoing**.

Více informací lze nalézt v nápovědě EsiDet u objektu **ModbusSlaveTCP**.

4 Technická podpora

Veškeré informace ohledně komunikace regulátorů AMREG s řídicími systémy AMiT, 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.