



NAPOJENÍ ZAŘÍZENÍ S KOMUNIKACÍ BELIMO MP-BUS NA SÍŤ AUTOMATŮ MICROPEL

Principy komunikace zařízení na MP-Bus s automatem MICROPEL,
popis prostředků pro zákaznický program

edice 02.2014
verze 1.0

MPC400 – MP-Bus

© MICROPEL s.r.o. 2014
Ing. Tomáš Rázga

všechna práva vyhrazena
kopírování publikace dovoleno pouze bez změny textu a obsahu
<http://www.micropel.cz>

PRINCIP KOMUNIKACE

Pro ovládání zařízení s komunikací MP-Bus automatem MICROPEL lze použít převodník komunikací MP-Bus – Modbus RTU, konkrétně Belimo UK24MOD. Automat pak může pomocí sady Modbus registrů převodníku ovládat a vyčítat stav zařízení připojených k převodníku na lince MP-Bus. UK24MOD umožňuje připojit max. 8 MP-Bus zařízení.

NÁVOD K TVORBĚ PROGRAMU

V automatu je třeba mít spuštěn USART ovladač linky RS485. V programu pak lze využít obecnou Simple4 knihovnu komunikace Modbus RTU master a rozšiřující knihovnu pro komunikaci s převodníkem UK24MOD, převodník funguje jako Modbus RTU slave. Programem můžeme obsluhovat i více převodníků UK24MOD, použití nijak neomezuje možnost přístupu k jiným Modbus RTU slave zařízením na lince. Všechny slave nicméně musí mít nastaveny shodné komunikační parametry.

KNIHOVNA PRO UK24MOD

Každý ovládaný převodník bude v programu reprezentován proměnnou s předdefinovanou strukturou `_UK24MOD`. Po resetu automatu v programu zavoláme inicializační proceduru komunikace s daným převodníkem, mimo to je třeba také nastavit komunikační parametry Modbus master. Poté začne - na základě volání obslužné procedury v každém průchodu programovou smyčkou - automatické vyčítání hodnot řídicích a stavových registrů zařízení MP-Bus z převodníku po lince Modbus. Vyčítané hodnoty budou ukládány do proměnné převodníku a bude možno s nimi v programu snadno pracovat. Také bude možno zadávat povely k zápisu hodnot řídicích registrů do převodníku přes linku Modbus. Každé zařízení MP-Bus může mít navíc zapojen externí senzor, komunikační strojek automaticky zajistí požadované nastavení typu senzoru daného zařízení podle zadání. Zápisy do registrů *Sensor-type* v převodníku se tak uživatel v programu nijak nemusí zabývat.

Nastavení typu senzoru

Hodnoty typů senzorů připojených k MP-Bus zařízením se předají jako jeden z parametrů inicializační procedury. Po spuštění komunikačního strojku dojde nejprve k zápisu těchto hodnot do registrů *Sensor-type* v převodníku, až poté začne proces opakovaného vyčítání registrů podle nastavení. Navíc bude strojek zápisy registru *Sensor-type* neustále opakovat (konkrétně každých 90 sekund), jak je doporučeno výrobcem v manuálu k převodníku UK24MOD.

Práce s vyčtenými hodnotami registrů převodníku

Každému správně naadresovanému zařízení na MP-Bus odpovídá jedna pozice. Každá pozice je v převodníku reprezentována 20 registry, ke kterým lze přistupovat přes linku Modbus. Připojených MP-Bus zařízení může být 8, s adresami v rozmezí 1-8 (resp. pozicemi 0-7). Inicializační proceduře se předá informace, jaké pozice budou na MP-Bus aktivní. Komunikačním strojkem pak budou z převodníku opakovaně vyčítány pouze registry zařízení na těchto pozicích. Proměnná převodníku obsahuje také položku `MPDev`, což je pole 8 struktur typu `_UK24MOD_DevReg`. Definice struktury vypadá takto:

```
type struct
    word Setpoint,
    word OverControl,
    word Command,
```

```

word ActualPosition,
word RelVolFlow,
word ErrorsMalfunctions,
word SensorType,
word SensorValue,
word SeriesNum1,
word SeriesNum2,
word SeriesNum4,
word ActuatorType,
word TimeMonitoring,
word Min,
word Max,
word AbsVolFlow,
word _nu1,
word _nu2,
word _nu3,
word _nu4,
byte readoutErr,
byte stype,
longword stypetime
end _UK24MOD_DevReg

```

Od prvního wordu *Setpoint* až po word *_nu4* struktury budou uloženy vyčítané hodnoty 20 registrů z převodníku. Položka *readoutErr* určuje platnost vyčtených hodnot, zbylé položky jsou pracovní proměnné komunikačního strojeku. Obecně lze říci, že hodnoty prvních 20 položek jsou platné, když je hodnota *readoutErr* rovna 0.

Zápisy registrů převodníku

Do převodníku je možno nastavit hodnoty 6 z celkem 20 registrů každého MP-Bus zařízení. Zápis hodnoty registru se kdekoli v programu provede přes zápisovou funkci. Pro čísla těchto 6 registrů, která se uvádí jako jeden z parametrů zápisové funkce, platí předdefinované konstanty: *_MPDEVREG_Setpoint*, *_MPDEVREG_OverControl*, *_MPDEVREG_Command*, *_MPDEVREG_SensorType*, *_MPDEVREG_Min*, *_MPDEVREG_Max*. Nicméně, registr *Sensor-type* bude zapisován komunikačním strojkem automaticky, opakovaně.

Popis knihovních funkcí

Inicializační procedura

```

subroutine UK24MOD_Init(var _UK24MOD ukmod, byte addr, word msk,
word t100, const _UK24MOD_SensorTypes st)

```

Proceduru zavoláme po restartu automatu. Předávanými parametry jsou:

- ukmod* ... proměnná struktury *_UK24MOD*; reprezentuje připojený převodník UK24MOD;
- addr* ... skutečná Modbus adresa převodníku; v rozsahu 1 až 247;
- msk* ... bitová maska aktivních MP-Bus zařízení připojených k převodníku; pouze registry těchto zařízení budou také vyčítány komunikačním strojkem; *msk=0x82* značí aktivní zařízení na pozicích 1 a 7, to odpovídá zařízením MP2 a MP8 na krabici převodníku;
- t100* ... požadovaná doba opakování vyčítání registrů každého aktivního MP-Bus zařízení; jednotka je 100 milisekund; hodnota bude odpovídat požadavkům na aktuálnost vyčtených dat; je zbytečné

vyčítat příliš často, zvlášť jestli bude program obsluhovat více Modbus SL zařízení; pro konkrétní hodnotu *t100* bude vytížení linky Modbus záviset na zadaném počtu aktivních MP-Bus zařízení pomocí *msk*; *st* ... pole 8 bajtů v kódové paměti udávající konfiguraci připojených typů senzorů; každý bajt odpovídá jedné pozici MP-Bus zařízení; u neaktivních zařízení a zařízení bez připojeného senzoru se zadá hodnota 0; viz vzorový příklad;

Pokud dojde k zadání chybné Modbus adresy nebo nulového počtu aktivních zařízení, komunikační strojek pak žádnou komunikaci nezahájí.

Komunikační strojek

```
subroutine UK24MOD_ServiceLoop(var _UK24MOD ukmod, var _modbusMA modM)
```

Proceduru voláme v každém průchodu programovou smyčkou automatu. Strojek zajišťuje vyčítání registrů aktivních MP-Bus zařízení z převodníku UK24MOD. Také obstarává provádění příkazů zápisu registrů zadaných pomocí zápisové funkce. Předávanými parametry jsou:

ukmod ... proměnná struktury *_UK24MOD*; reprezentuje připojený převodník UK24MOD;
modM ... proměnná typu Modbus master reprezentující linku, na níž je převodník UK24MOD připojen;

Zápisová funkce

```
function bit UK24MOD_SetWriteDevReg(var _UK24MOD ukmod, byte devix,  
byte regn, word regval)
```

Funkci voláme kdekoli v programu pro zadání příkazu zápisu registru v převodníku UK24MOD. Předávanými parametry jsou:

ukmod ... proměnná struktury *_UK24MOD*; reprezentuje připojený převodník UK24MOD;
devix ... pozice (0 až 7) MP-Bus zařízení, jehož registr má být zapsán;
regn ... číslo registru (1 až 20) zařízení, který má být zapsán; je vhodné použít jednu z předdefinovaných konstant *_MPDEVREG_Setpoint*, *_MPDEVREG_OverControl*, *_MPDEVREG_Command*, *_MPDEVREG_SensorType*, *_MPDEVREG_Min*, *_MPDEVREG_Max*; ostatní registry slouží jen pro čtení;
regval ... hodnota zapisovaná do registru;

Funkce vrací 0, pokud je v současné době zadán nebo zatím nedokončen jiný příkaz zápisu, pak je třeba požadavek zopakovat později. Po úspěšné zadání příkazu zápisu vrací funkce hodnotu 1, příkaz bude zpracován a je třeba počkat na jeho výsledek. Dokončení zápisu je možno detekovat v programu stavem (*ukmod.cmdSet = 0 and ukmod.cmdResult <> 0*), který bude trvat až do zadání dalšího příkazu. Hodnota *ukmod.cmdResult* udává výsledek posledního zápisu:

1 ... zápis proběhl bez chyby;
13 ... v příkazu zadána neplatná hodnota *devix* nebo *regn*;
jiná ... chyba Modbus; hodnota má přímo význam chybového kódu zpracování Modbus příkazu;

Vzorový příklad

Program nastaví parametry a spustí vyčítání hodnot registrů MP-Bus zařízení na pozicích 1 a 3 (počítáno od pozice 0) převodníku UK24MOD. Po nastavení příznaku *event1* resp. *event2* vně tohoto kódu dojde k zápisu hodnoty registru *Setpoint* resp. *Override-control* zařízení na pozici 1. Nastavení *event3* spouští zápis registru *Command* zařízení na pozici 3. Při zadání příkazu zápisu dojde ke shození příznaku.

```
;---- Modbus-master:
var _modbusMA modM
if (RESET) then
    ; Modbus komunikace na lince L2, kom. rychlost 38400 Bd, bez parity,
    ; timeout odpovědi 20 ms, min. doba klidu na lince 5ms
    ModbusMA_Config(modM,1,38400,0,20,5)
ModbusMA_STM(modM)
;---- UK24MOD:
var _UK24MOD ukmod
; MP-Bus zařízení na pozici 1 (počítáno od 0) má připojen pasivní senzor
; 1-20kOhm, zařízení na pozici 3 má připojen aktivní senzor
code _UK24MOD_SensorTypes ukmod_stype = (0,3,0,1,0,0,0,0)
if (RESET) then
    ; převodník UK24MOD s Modbus-slave adresou 7 (komunikační parametry a Modbus
    ; adresa se nastavují prepínači DIP), na rozhraní MP-Bus budou zapojena 2
    ; zařízení na pozicích 1 a 3, pauza mezi dvěma následnými vyčteními registrů
    ; téhož zařízení bude 10*100ms (1 sekunda), jako poslední parametr se předávají
    ; typy senzorů, které jsou připojeny k jednotlivým zařízením MP-Bus
    UK24MOD_Init(ukmod,7,0x0A,10,ukmod_stype)
UK24MOD_ServiceLoop(ukmod,modM) ; obslužná procedura
;---- Obslužný program MP-Bus zařízení:
var byte event1, event2, event3
var word dev1_setpoint
var byte dev1_overval
var byte wr_devpos, wr_regn, wr_pendle
var word wr_regval
if (RESET) then
    begin
        wr_regn=0
        wr_pendle=0
        event1=0
        event2=0
        event3=0
    end
; ** obsluha MP-Bus pozice 1:
if (ukmod.MPDev[1].readoutErr = 0 and ukmod.MPDev[1].ActuatorType <> 0) then
    begin
        ; registry zařízení na pozici 1 vyčteny a zařízení je přítomno
        ; na MP-Bus, mohu pracovat s hodnotami registrů ukmod.MPDev[1]
        ; ...
        if (wr_regn = 0) then ; lze zadat další zápis
            begin
                ; zde se vyhodnocují události, které vedou k zápisu
                if (event1) then
                    begin
                        event1=0
                        wr_devpos=1
                        ; změna polohy ventilu zařízení
                        wr_regn=_MPDEVREG_Setpoint
                        wr_regval=dev1_setpoint
                    end
            end
    end
```

```

else if (event2) then
    begin
        event2=0
        wr_devpos=1
        ; řízení polohy pomocí Override control registru
        wr_regn=_MPDEVREG_OverControl
        wr_regval=dev1_overval
    end
end
end
; ** obsluha MP-Bus pozice 3:
if (ukmod.MPDev[3].readoutErr = 0 and ukmod.MPDev[3].ActuatorType <> 0) then
    begin
        ; registry zařízení na pozici 3 vyčteny a zařízení je přítomno
        ; na MP-Bus, mohou pracovat s hodnotami registrů ukmod.MPDev[3]
        ; ...
        if (wr_regn = 0) then ; lze zadat další zápis
            begin
                ; zde se vyhodnocují události, které vedou k zápisu
                if (event3) then
                    begin
                        event3=0
                        wr_devpos=3
                        ; spuštění Adaptace zařízení
                        wr_regn=_MPDEVREG_Command
                        wr_regval=1 ; příkaz Adaptation
                    end
                end
            end
        end
; ** obsluha požadavků zápisu do převodníku:
if (wr_regn <> 0) then
    begin
        ; řízení zápisu registru
        if (wr_pendle = 0) then
            begin
                ; zde se zadává nový zápis
                if (UK24MOD_SetWriteDevReg(ukmod,wr_devpos,wr_regn,wr_regval)) then
                    wr_pendle=1
                end
            end
        else ; čeká se na dokončení zápisu
            begin
                ; zde se čeká na dokončení zápisu
                if (ukmod.cmdSet = 0 and ukmod.cmdResult <> 0) then
                    begin
                        wr_pendle=0 ; zápis dokončen
                        ; Pokud zápis (komunikace na lince Modbus) proběhl bez chyby,
                        ; případně, pokud byly předány chybné parametry zápisu (wr_devpos
                        ; mimo rozsah 0-7 nebo wr_regn mimo rozsah 1-20), proces zápisu
                        ; ukončíme a bude možno zadat zápis nový. Jinak dojde k opakování.
                        if (ukmod.cmdResult = 1 or ukmod.cmdResult = 13) then
                            wr_regn=0
                        end
                    end
                end
            end
        end
; ... pokračován v programu automatu ...
; --- konec celého programu:
RESET=0
end

```