

Zobrazení grafů na MT201

Popis funkcí knihovny pro
zobrazení grafů na displeji
terminálu MT201

3.2008

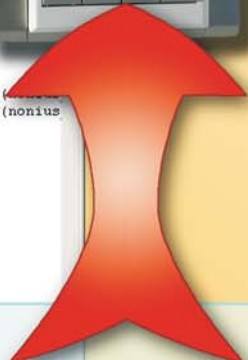
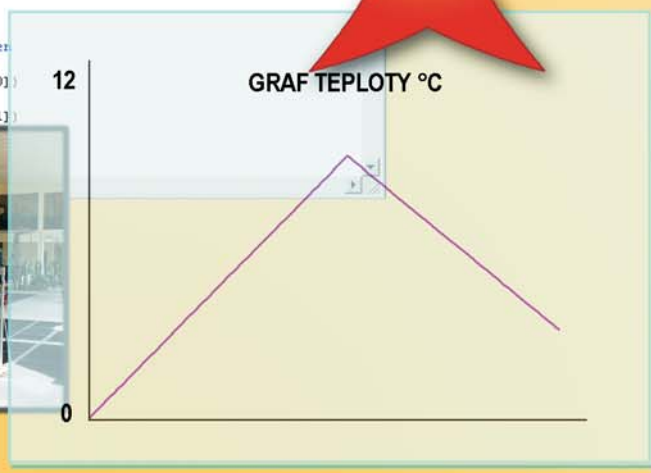
```
----- EDITACNI OBRAZOVKA ZADANA HODNOTA -----
subroutine edit_screen_1(bit start_edit)
var longint nonius_pos

var int tmpm

if (start_edit) then
begin
delta_y = longint
delta_x = longint
if (delta_x = 0)
delta_q = delta
end

increment_nonius_in
nonius_pos = (delta
if (nonius_pos > 10
if (nonius_pos < 10
if (temp_edit <= te
if (temp_edit >= te

if show_preset ? za
begin
position = 0
font = norm_font
display(zad_hodnota_text_detail)
format = 0;
position = 40
font = high_font
if ((T7 & blink_time_msek) = 0) then
begin
display(edit_arrow_text[0])
end else begin
display(edit_arrow_text[1])
end
display
end
```





GRAPHMT201.LIB

Knihovna GraphMT201.lib slouží k zobrazování grafů na terminálu MT201 a lze ji použít samostatně nebo v kombinaci s knihovnou Menulib2_MT201. Knihovna umožňuje vykreslit čárový nebo sloupcový graf z uložených hodnot.

edice 3.2008.

verze 1.0

GraphMT201.lib

© Ing. Jaroslav Kurzweil 2008

MICROPEL s.r.o. 2008

všechna práva vyhrazena

kopírování publikace dovoleno pouze bez změny textu a obsahu

<http://www.micropel.cz>

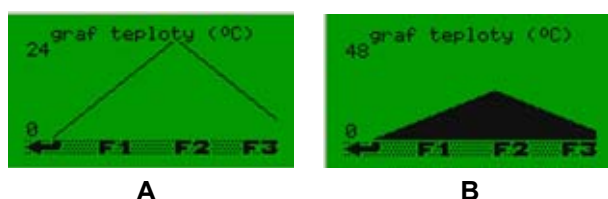
Obsah

I. Základní popis knihovny.....	3
1 Úvod – možnosti knihovny	3
2 Princip a podmínky funkce knihovny	3
3 Popis zobrazovací procedury	5
Nastavení využívané oblasti displeje.....	6
Parametry pro výběr zobrazovaných dat.....	7
4 Příklady použití knihovny.....	7
4.1 Příklad MT201 - začínáme	8
4.2 Příklad EX10	10
Graf teploty	11
Graf napětí	12
Graf kontaktu	12
Graf odporu.....	13
4.3 Příklad „Demo“	14
Typy zobrazení grafů v příkladu	14
5 Závěr	20
II. Pokročilé možnosti použití knihovny	21
1 Úvod	21
2 Příklad „Čtyři grafy“	21
Zajímavosti v kódu	22
3 Příklad „EX10 - monitor“	23
4 Příklad „EX08“	24
5 Závěr	25
Šrafovaný graf	25
Graf s proměnnou šířkou čáry	25
Graf s trojúhelníhovým štítem.....	25
Co dodat ?	25

I. Základní popis knihovny

1 Úvod – možnosti knihovny

Knihovna GraphMT201.lib slouží k zobrazování grafů na terminálu MT201 a lze použít samostatně nebo v kombinaci s knihovnou menu „Menulib2_MT201“. Knihovna umožňuje vykreslit graf čárový (bodový) tj. složený z bodů reprezentující hodnoty grafu - Obr. 1A nebo sloupcový, kde je zobrazená hodnota reprezentována vyplněným sloupcem - Obr. 1B. Knihovnu je možné využít v aplikaci vícenásobně např. v různých částech menu a je možno zobrazit průběh více veličin v jednom grafu. Pro optimální zobrazení lze měnit měřítka v ose x i y, pomocí parametrů posouvat zobrazovanou oblast horizontálně i vertikálně přes celou oblast zobrazovaných dat. Je možné nastavit velikost a umístění oblasti tisku grafu v rámci plochy grafického displeje MT201. Nastavení všech výše uvedených vlastností se provádí pomocí osmi parametrů.



Obr. 1 Podporované typy grafu

Použití knihovny je dokumentováno sadou příkladů, obsažených v instalaci společně s knihovnou, s jejichž pomocí jsou ukázány některé možnosti a chování knihovny. Zdrojové kódy jsou elektronickou přílohou tohoto manuálu.

Zdrojový text knihovny je vytvořen velmi úsporně, neboť knihovna při plném využití jejich funkcí zabere v aplikaci méně než 3kB a tudíž nijak výrazně neomezuje další možnosti programování MT201.

2 Princip a podmínky funkce knihovny

Knihovna GraphMT201 tiskne graf na displej terminálu MT201 v maximálním rozlišení, přičemž k tisku využívá maximálně šesti znakových řádek. Na jednom řádku lze umístit 21 znaků a znaky jsou tištěny v matici 8x6 bodů. Největší dosažitelné rozlišení grafu vyplývající z uvedeného počtu řádků vychází na 48x126 grafických bodů. Pro realizaci zobrazení využívá knihovna celkem 126 znaků, které předefinuje z celkových 256 dostupných. Využívá k tomu tzv. „netisknutelné“ znaky, které mají následujících ASCII kódy:

0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 39, 91, 92, 93, 94, 95, 96,
123, 124, 125, 127, 129, 130, 131, 132, 133, 134, 135, 136, 137, 140,
144, 152, 153, 156, 159, 161, 162, 163, 165, 166, 167, 168, 169, 170,
171, 172, 173, 174, 175, 178, 179, 180, 181, 182, 183, 184, 185, 186,
187, 188, 189, 190, 191, 192, 194, 195, 196, 197, 198, 199, 202, 203,
206, 208, 209, 212, 213, 214, 215, 219, 220, 222, 223, 224, 226, 227,
228, 229, 230, 231, 234, 235, 238, 240, 241, 244, 245, 246, 247, 251,
252, 254, 255

Znaky zmíněných kódů jsou průběžně „překreslovány“ na sadu grafických prvků potřebných pro zobrazení grafů a tudíž nelze původní znaky využít pro tisk textu. Aktuálně dostupnou kompletní znakovou sadu lze prohlížet přímo v simulátoru v příloženém příkladu „Demo“.

Použití funkce pro tisk grafu je velmi jednoduché. Jedinou uživatelem používanou procedurou je procedura GraphMT201Draw definovaná:

**subroutine GraphMT201Draw (var GraphMT201DataType GraphMT201Data,
var GraphMT201ParType GraphMT201Pars)**

Parametry procedury představuje dvojice odkazů. První parametr je odkaz na strukturu dat zobrazovaného grafu, druhý ukazuje na strukturu ovlivňující způsob zobrazení grafu. Strukturu dat zobrazovaného grafu tvoří jednoduché pole wordů obsahující data určená k zobrazení. Velikost této struktury je daná konstantou GraphMT201DataTypeLenght. Tato konstanta je definovaná uživatelem a proto jsou pro použití knihovny GraphMT201 nutné následující dva úkony.

- Vložit knihovnu GraphMT201 mezi použité knihovny do projektu a to tak, aby se vložená knihovna týkala jen příslušného automatu MT201.
- Určit délku všech používaných datových struktur definicí.

Definici délky datových struktur zapíšeme ve tvaru:

const GraphMT201DataTypeLenght=xxx ,

kde xxx představuje v zásadě libovolné číslo z rozsahu 1-65535. Smysl má však pouze taková hodnota, která si vyžádá alokaci datové paměti menší než je k dispozici u MT201 pro uživatelské proměnné Simple 4. Maximální rozsah paměti, který můžeme pro uložení dat použít, je tedy asi 15kB, za předpokladu, že ponecháme nějakou rezervu pro další uživatelské proměnné. Maximální počet položek, které můžeme definovat konstantou GraphMT201DataTypeLenght je tedy dán:

GraphMT201DataTypeLenght = 15360 / 2 / počet_grafů

To v případě jednoho grafu dává pro maximální počet měřených bodů hodnotu 7680. Při zobrazení více než jednoho grafu, musíme počítat s tím, že pro všechny datové struktury záznamů vychází stejný počet položek, který je dán právě popisovanou konstantou.

!!! POZOR !!! Bez splnění druhého bodu projekt nelze přeložit. Překladač hlásí chybu „nedefinovaný identifikátor GraphMT201DataTypeLenght“.

Další krok, který musí programátor učinit, pokud chce kromě úspěšného překladu vytisknout i nějaký graf, spočívá v definování potřebné datové struktury a struktury parametrů. Datová struktura bude obsahovat hodnoty grafu a musí být typu „GraphMT201DataType“. Proměnných daného typu můžeme definovat v případě potřeby zobrazení většího počtu grafů i více (každou z nich pro jeden graf) a to až do omezení dané velikostí paměti MT201, jak bylo zmíněno výše. Pro definici proměnné použijeme zápis ve tvaru:

var GraphMT201DataType „jméno“ ,

kde na pozici „jméno“ použijeme jméno proměnné, které si zvolíme. Podle příkladu „Demo“ popsaneho v kapitole 4.3. použijeme například zápis deklarace ve tvaru:

var GraphMT201DataType Graph1

Typ GraphMT201DataType je definován knihovnou GraphMT201 jako pole wordů délky GraphMT201DataTypeLenght. Výhodou této konstrukce je právě to, že uživatel může určit délku i počet datových struktur optimálně podle řešené aplikace.

Pro uchování parametrů grafu definuje knihovna datový typ GraphMT201ParType. Tento typ představuje datovou strukturu, která je tvořena následujícími položkami:

```
type struct
word Yoffset,
word XBegin,
word XScale,
word YScale,
byte Mode
end GraphMT201ParType
```

Význam těchto jednotlivých položek je popsán v následující kapitole „3. Popis zobrazovací procedury“ v odstavci „Parametry pro výběr zobrazovaných dat“. Pro použití knihovny je potřeba deklarovat strukturu parametrů typu GraphMT201ParType konstrukcí:

```
var GraphMT201ParType Paramethers,
```

kde Paramethers je název – identifikátor deklarované struktury parametrů (viz. kapitola „Příklad „Demo““).

Shrnutí:

Z uvedeného popisu vyplývají pro použití knihovny tři základní úkony, které je třeba bezpodmínečně dodržet. V kroku prvním je nutné správně vložit knihovnu do projektu a v kroku druhém pak definovat hodnotu konstanty GraphMT201DataTypeLenght. Nakonec je nutné deklarovat strukturu dat a strukturu parametrů.

3 Popis zobrazovací procedury

Jak již bylo zmíněno výše, je použití knihovny pro kreslení grafu extrémně jednoduché. Pokud chce programátor vykreslit graf stačí, aby:

- definoval strukturu dat a parametrů (např. Graph1 a Paramethers)
- nastavil parametry a pořídil platná data
- zavolal proceduru kreslení grafu příkazem volání procedury ve tvaru:

```
GraphMT201Draw (Graph1,Paramethers)
```

Datová struktura Graph1, obsahující všechny pořízené hodnoty k zobrazení grafu, může obsahovat daleko více položek, než je zobrazitelné maximum 126, které je dáno rozlišením displeje terminálu MT201.

To je dovoleno proto, aby bylo možné:

- využít archivovaná data i k jiným účelům
- posouvat zobrazovanou oblast v rámci grafu a měnit měřítko zobrazení

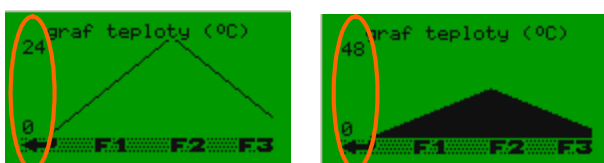
Příkladem použití vyššího počtu položek než 126 může být úloha záznamu teploty za posledních 24 hodin v minutovém časovém rastru, kdy požadujeme zobrazení přehledového grafu za celých uplynulých 24 hodin a současně prohlížení detailů záznamu ve zvoleném časovém úseku. Pro zaznamenání potřebných dat je vhodná struktura se 1440 záznamy vycházející z počtu minut za 24 hodin tj. součinu $24 \cdot 60$. Tisk dat ze struktury pak budeme řídit pomocí parametrů tisku grafu v kontextu ke zvolenému požadavku typu přehled nebo detail.

Parametry pro tisk grafu se dají rozdělit do dvou skupin. První skupinu tvoří tři parametry, které určují oblast tisku grafu na displeji MT201. Ty jsou globálně definované knihovnou a proto je můžeme kdekoli v programu libovolně použít resp. nastavit. Druhou skupinu tvoří pět parametrů, které jsou obsaženy v proměnné s datové strukturou parametrů, kterou v aplikaci deklaroval uživatel. Tyto parametry určují vlastnosti výběru dat z datové struktury a způsob jejich tisku. Protože knihovna umožňuje tisk více grafů současně, máme možnost přiřadit každému grafu samostatnou strukturu parametrů. Naproti tomu, parametry první skupiny musí být společně všem aktuálně tištěným grafům a proto jsou definovány globálně.

Nastavení využívané oblasti displeje

Pro nastavení oblasti displeje využívané k tisku grafu jsou určeny následující knihovnou globálně definované proměnné:

- **var byte GraphMT201DispXSize** – určuje počet tiskových bodů grafu. Hodnota proměnné může být maximálně 126. Pokud chceme zjistit počet znaků v řádku využitých pro tisk grafu vyjdeme z faktu, že šest bodů grafu tvoří vždy jeden tištěný sloupec znaků. Počet tištěných sloupců je tedy dán jako počet bodů děleno šesti a zaokrouhleno nahoru, tj. při počtu 126 bodů se využívá všech 21 znaků. Při počtu 119 bodů grafu bude využito pouze 20 znaků a jeden znak tak v každém řádku zůstane k dalšímu využití. Pokud se tiskne méně než 21 znaků, je graf zarovnán na displeji vpravo. Prázdné nevyužité sloupce se nacházejí vlevo od plochy grafu a volné místo, které představují, může sloužit např. pro dotisk popisu svislé číselné osy (viz. Obr. 2).



Místo pro dotisk popisu osy

Obr. 2 Zarovnání grafu na displeji a popis osy

- **var byte GraphMT201DispYSize** – určuje, kolik znaků bude pro graf použito ve svislém směru. Pro displej terminálu MT201 mají smysl hodnoty z rozsahu 1-6.
- **var byte GraphMT201DispYBegin** – určuje počet nepoužitých řádků nad oblastí grafu. Jedná se o počet vynechaných řádků od vrchního okraje displeje. Logicky tedy musí platit, že:

$$\text{GraphMT201DispYSize} + \text{GraphMT201DispYBegin} \leq 8.$$

Parametry pro výběr zobrazovaných dat

Jsou součástí uživatelem deklarované proměnné s definovanou datovou strukturou parametrů. Při použití různých měřítek zobrazení grafu je nutné správně volit hodnoty jednotlivých parametrů struktury. Pro nastavení parametrů tisku grafu slouží následující položky datové struktury:

- **var word XScale** – musí být větší než nula a určuje měřítko horizontální osy (osa X). Pokud je hodnota proměnné 1, pak budou z datové struktury zobrazovány hodnoty po sobě v souvislé řadě. Při hodnotě 2 bude zobrazena každá druhá hodnota z datové struktury atd. Z uvedeného principu zpracování měřítka plyne, že zobrazení grafu v maximálním detailu je to, kdy jsou jednotlivé hodnoty zobrazovány právě sousedními grafickými body displeje.
- **var word YScale** – určuje měřítko vertikální osy. Pokud má být např. zobrazena teplota s rozlišením jednoho °C, musíme, při standardním záznamu teploty s rozlišením jedné desetiny kelvina [0.1K], nastavit parametr na hodnotu 10. Pro přepočítání stupňů K na stupně C použijeme ještě následující proměnnou **YOffset** pro posun v ose Y. Pro zpracování měřítka v ose Y platí stejný princip jako v ose X.
- **var word YOffset** – určuje posun grafu ve vertikálním směru. Pokud bude zaznamenána teplota větší než 0°C, použijeme hodnotu 2732 pro přizpůsobení rozsahu hodnot na rozsah zobrazovaný včetně přepočtu jednotek K na °C. Ta bude při zobrazování grafu odečtena od hodnoty zaznamenané a tak bude dosaženo požadovaného přizpůsobení a posunu. Pokud budeme zobrazovat hodnoty od -10°C použijeme k přizpůsobení hodnotu 2632. Z uvedeného plyne, že hodnoty, které chceme zobrazovat, musíme nastavením měřítek a posunu přepočítat do rozsahu od 0 do **GraphMT201DispYSize * 8**.
- **var word XBegin** – určuje pozici první zobrazené hodnoty grafu v datové struktuře. Pokud bude mít struktura velikost například 200 záznamů a hodnota tohoto parametru bude 199, pak první zobrazený záznam bude představovat poslední položkou s indexem 199 a dál graf naváže hodnotami ze začátku struktury, tj. od indexu 0. Této vlastnosti lze s výhodou využít při postupném ukládání dat a též při požadavku na průběžné zobrazování dat například v typickém požadavku „za poslední hodinu“. Potom paralelně s indexem, který určuje pozici zápisu do struktury, posouváme tento parametr tak, aby byly zobrazeny požadované údaje. Graf se pak v čase automaticky posouvá doleva (viz příklad „Demo“ položka Graph1-a). Výhoda spočívá v tom, že není potřeba přesunovat hodnoty datové struktury.
- **var byte Mode** - představuje parametr, který určuje mód tisku. Je-li 0, tiskne se sloupcový graf, je-li 1 tiskne se čárový graf, přičemž hodnota parametru určuje šířku kreslené čáry grafu (smysl mají hodnoty 1..48). Nejvyšší bit parametru má speciální význam při současném tisku více grafů. Jednička zamezí tiskové proceduře přechod na tisk dalšího bodu grafu a tím umožní tisk více grafů „přes sebe“. Použití této funkce je podrobně ukázáno v řešených.

4 Příklady použití knihovny

Cílem následujících příkladů je nejen demonstrovat možnosti použití knihovny pro tisk grafu, ale hlavně ukázat jeho správné začlenění do menu vytvořeného pomocí knihovny Menulib2_MT201. V dalším textu najdete také upozornění na zajímavé detaily, které přiblíží jednotlivé funkce knihovny i některá možná úskalí. Možnosti použití knihovny jsou samozřejmě větší a není možné ukázat všechny varianty, nicméně pro hlubší studium doporučujeme prostudovat oddíl „II. Pokročilé možnosti použití knihovny“.

Všechny níže popsané příklady jsou součástí instalace knihovny GraphMT201.lib a je možné studovat jejich funkci pomocí programovacího prostředí StudioWin a jeho simulátorů.

4.1 Příklad MT201 - začínáme

Tento příklad je nejjednodušší ukázkou použití knihovny GraphMT201. Projekt obsahuje jen terminál MT201. Pro kreslení průběhu teploty, kterou MT201 měří integrovaným čidlem, využívá aplikace 2. až 7. řádek displeje. Po spuštění simulátoru klikněte na tlačítko simulátoru „Teplota“ a zobrazeným jezdcem měňte hodnotu měřené teploty. Časový průběh změn uvidíte „on line“ na displeji MT201. Jemné změny hodnoty je možné provádět pomocí kolečka myši nebo kláves se šipkami „nahoru“ a „dolů“.

MT201 ukládá teplotu do datové struktury GraphTemperature, která obsahuje 120 položek. Záznam hodnot teploty je prováděn v časovém rastru jedné sekundy. Díky nastavení horizontálního rozměru GraphMT201DispXSize na hodnotu 120 a parametru měřítka .XScale na 1, jsou zobrazena všechna data z datové struktury na displeji. Rozsah ukládaných dat a perioda zápisu je zvolena z demonstračních důvodů. Ve skutečnosti se teplota obvykle mění velmi pomalu a perioda vzorkování teploty bývá např. 15 minut nebo i více. Výsledek zobrazení průběhu teploty na simulátoru ukazuje Obr. 3.



Obr. 3 Simulace průběhu teploty z příkladu MT201

Rozsah zobrazovaných teplot je 0°C..48°C (48 bodů v ose Y, .YScale = 10, tj. 1bod = 1°C, .YOffset = 2732). Je ale nutné upozornit na fakt, že pokud měřená teplota bude menší než 1°C, bude počet vykreslených dílků 0. Prakticky použitelný rozsah tak začíná ve skutečnosti právě od 1°C, neboť pro teplotu od 1°C do 1.99°C bude vykreslen jeden dílek grafu. Odlišná situace je v horní části grafu. V případě, že hodnota měřená teploty bude větší nebo rovna 49°C, graf tuto teplotu již nezobrazí. Z uvedených skutečností plyne, že správně zobrazovaný rozsah teplot odpovídá intervalu od 1°C do 48.99°C. Uvedená úvaha platí při použití čárového zobrazení.

V případě sloupcového grafu, při zobrazování hodnoty teploty kolem 1°C, je situace stejná. Významný rozdíl nastane při zobrazování maximální hodnoty rozsahu. Pokud bude teplota vyšší nebo rovna 48°C, dosáhne sloupec maxima a z grafu tak není poznat zda je teplota 48°C nebo třeba 150°C. Proto u sloupcového grafu bude rozsah správně zobrazených teplot o jeden stupeň nižší a to od 1°C do 47.99°C. Čárový graf je výhodnější z hlediska zobrazovaného rozsahu, který je o 1 dílek (1°C) větší oproti sloupcovému zobrazení. To ale zase poskytuje informaci o směru překročení zobrazovaného rozsahu. Je-li teplota pod rozsahem, sloupec není vykreslen vůbec, je-li

vyšší, pak je sloupec vykreslen plně. Čárový graf při překročení mezí jakýmkoli směrem není, jak plyne z logiky věci, vykreslen vůbec a tak nedostaneme o skutečné teplotě žádnou informaci.

Program aplikace je přímo v kódu okomentován a nepotřebuje mnoho dalších poznámek. Za zmínku stojí časování zápisu do datové struktury odvozené od proměnných Second a Minute reálného času automatu. Takto odvozené časování má výhodu v tom, že stanovení okamžiku zápisu do datové struktury proběhne správně i po výpadku napájecího napětí nebo zastavení automatu. Pro zcela korektní zobrazování by bylo ještě vhodné doplnit kód o vymazání neplatných záznamů od výpadku napájení po jeho náběh. Jinou možností, jak řešit vzorkování a ukládání změřené hodnoty, je využití některého ze systémových časovačů T0 až T7. Tato možnost však není odolná vůči výpadkům napájení. Pro úplnost uveďme ještě výpis zdrojového textu. Části důležité pro nastavení parametrů zobrazení dat a parametrů knihovny jsou zvýrazněny žlutě.

```
const GraphMT201DataTypeLenght = 120      ;Definování délky datové struktury
var GraphMT201DataType GraphTemperature    ;Deklarace datové struktury
var GraphMT201ParType Parameters          ;Deklarace struktury parametrů
```

```
if Reset then begin
```

```
    GraphMT201DispXSize = 120      ;inicializace počtu tištěných sloupců na displeji
    GraphMT201DispYSize = 6        ;inicializace počtu využívaných řádků pro graf
    GraphMT201DispYBegin = 1       ;inicializace počtu prázdných řádků nad grafem
    Parameters.XScale = 1          ;inicializace měřítka v ose X
    Parameters.YScale = 10         ;inicializace měřítka v ose Y (1°C=1 dílek grafu)
    Parameters.YOffset = 2732      ;inicializace posunu ve vertikální ose
    Parameters.Mode = 0x04         ;inicializace módu tisku (bodový graf)
```

```
    Position=2
```

```
    Display( "TEPLOTA 1..48°C" )    ;tisk nadpisu grafu
    Y30=1                          ;rozsvícení displeje
    OpCtrl=0                        ;vypnutí zobrazení lišty popisu tlačítek
```

```
end
```

```
GraphTemperature[second+60*(Minute%2)] = IO ;zapisování integrovaného čidla teploty do datové struktury
```

```
GraphMT201.XBegin = (second+60*(Minute%2) +1) % 120      ;posouvání grafu směrem doleva,
                                                           ;nejnovější zaznamenaná hodnota je zcela vpravo
```

```
GraphMT201Draw(GraphTemperature,Parameters) ;volání tiskové procedury
```

```
Reset = 0
```

```
end
```

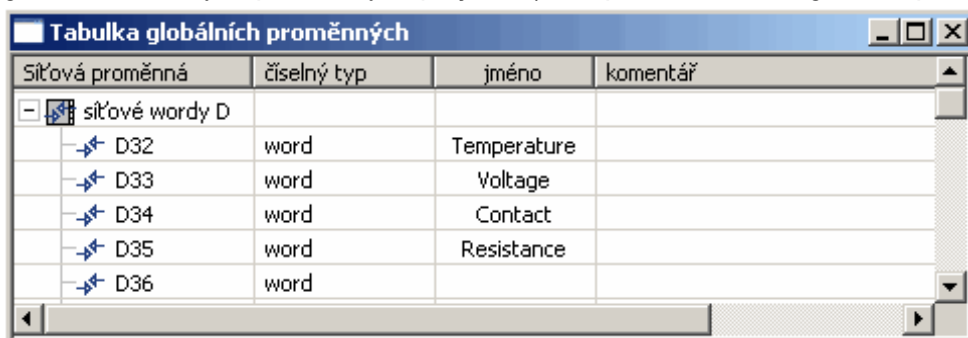
4.2 Příklad EX10

Příklad tvoří a zobrazuje čtyři grafy, přičemž zdrojem dat pro ně je periférie EX10. EX10 má čtyři vstupy a každý z nich je možné nastavit do jiného režimu. Toho je v příkladu využito, vstupy jsou nastaveny podle Tab. 1.

Vstup	Funkce	Identifikátor	Síťová proměnná
I0	teplota	Temperature	D32
I1	napětí	Voltage	D33
I2	kontakt	Contact	D34
I3	odpor	Resistance	D35

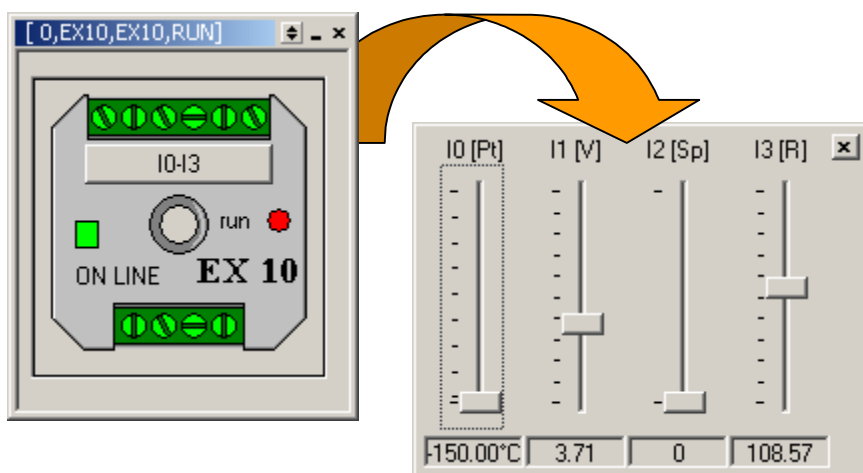
Tab. 1 Propojení vstupů EX10 na síťové proměnné

Uvedené nastavení je obsaženo částečně v nastavení periférie a částečně v nastavení globálních síťových proměnných projektu (viz 4 pracovní okno - globální proměnné).



Obr. 4 Tabulka přiřazení globálních proměnných

Příklad je svoji strukturou hodně podobný jednoduchému příkladu MT201. I zde jsou použity struktury délky 120 a data zaznamenávána po sekundě. Graf zobrazený na displeji představuje všechny pořízené hodnoty a pomocí programové konstrukce, která je stejná s konstrukcí v příkladu 4.1, se posouvá doleva. Stejně, jako v případě příkladu 4.1, lze v simulátoru vše vyzkoušet. K tomu je potřeba spustit v prostředí StudioWin naprogramování celého projektu.



Obr. 5 Simulátor EX10 a jeho ovládání

Po dokončení programování se objeví simulátory MT201 a EX10. Kliknutím na tlačítko „I0..I3“ simulátoru EX10 se rozevře, podle 5, panel se čtyřmi jezdcí. Změny hodnot jsou opět „on line“ vidět na displeji MT201, přičemž se grafy hodnot veličin jednotlivých vstupů periferie EX10 střídají po pěti sekundách.

Pro záznam čtyř grafů potřebujeme samozřejmě čtyři datové struktury. To je vyřešeno elegantně pomocí deklarace proměnné Graphs ve tvaru pole čtyř datových struktur typu MT201DataType. Dále je deklarováno pole čtyř názvů jednotlivých grafů a pole čtyř struktur parametrů pro řízení zobrazení každého z grafů. Tisk grafů jediným příkazem umožňuje to, že jsou všechna potřebná data, která patří k jednotlivým grafům, začleněna do jedné strukturované proměnné. Tisk grafů je prováděn příkazem:

GraphMT201Draw(Graphs[GraphIndex], Paramethers[GraphIndex])

Indexem GraphIndex vybíráme, který ze čtyř grafů bude právě zobrazen. Index je nastavován příkazem **GraphIndex = (Second / 5) % 4** tak, že zobrazování jednotlivých grafů se pravidelně střídá po 5 sekundách. Podobným způsobem je vždy tištěn do prvního řádku i titulek grafu. Velmi zajímavé je správné nastavení parametrů YOffset a YScale pro jednotlivé grafy. Pro všechny čtyři grafy je použito nastavení **GraphMT201DispYSize = 6**, které odpovídá 48 bodům zobrazení ve vertikálním směru. Protože je použit sloupcový graf, požadujeme, aby rozsah vstupní veličiny byl přepočítán na rozsah 1 až 47 celých dílků. K dispozici je tedy 47 bodů grafu.

Graf teploty

Pro zobrazení grafu použijeme následující volby a nastavení:

- Zvolený zobrazovaný rozsah veličiny je s ohledem na možnosti EX10 stanoven od -150 do 260°C. Z toho plyne, že se hodnota síťové proměnné bude pohybovat od 1232 do 5332 a rozsah hodnot vstupní veličiny tedy vychází na $5332 - 1232 = 4100$.
- Pro zobrazení tohoto rozsahu máme k dispozici 47 bodů. Z toho plyne svislé měřítko $\text{Paramethers}[0].\text{YScale} = 4100 / 47 = 87,23$, po zaokrouhlení nahoru je

YScale = 88

- Minimální hodnotu 1232 zobrazíme jedním dílkem grafu, a protože jeden dílek grafu odpovídá právě měřítku YScale, nastavíme hodnotu posunu na:

YOffset[0] = 1232 - 88 = 1144

- Nakonec provedeme výpočet vlivu zaokrouhlení hodnoty YScale. Výpočtem určíme hodnotu vstupní veličiny odpovídající maximální výšce sloupce. Z toho plyne hodnota teploty, pro níž by bylo zobrazeno všech 48 vertikálních bodů grafu. Zobrazení odpovídá maximální teplotě:

Tep_max = (1144 + 48 * 88 - 2732) / 10 = 263,6°C

Graf napětí

Pro graf napětí vyplývají následující volby a nastavení.

- Zvolený zobrazovaný rozsah veličiny odpovídá vzhledem k možnostem EX10 rozsahu od 0 do 10V. Z tohoto rozsahu plyne rozsah hodnoty síťová proměnné od 0 do 1000 a současně rozsah vstupní veličiny grafu od 0 do 1000.
- Pro zobrazení tohoto rozsahu máme k dispozici 47 bodů a svislé měřítko grafu pak bude odpovídat výrazu $\text{Parameters}[1].\text{YScale} = 1000 / 47 = 21,27$, což po zaokrouhlení činí

$$\text{YScale} = 22$$

- Pro zobrazení minimální hodnoty 0 požadujeme zobrazení jednoho dílku grafu. Jeden dílek grafu odpovídá právě měřítku YScale, a proto nastavíme posun na:

$$\text{YOffset}[1] = 0 - 22 = - 22$$

To není bohužel přípustná hodnota a proto nám nezbývá než ukládat do datové struktury data navýšená právě o 22 (viz řádek zdrojového textu $\text{Graphs}[1][\text{secondOld}] = \text{Voltage} + 22$). Je to proto, že pokud je v datové struktuře nula, tak již není možné jakýmkoli nastavením parametrů knihovny dosáhnout tisku nenulového sloupce. Logicky pak $\text{Parameters}[1].\text{YOffset}[1] = 0$.

- Nakonec provedeme výpočet vlivu zaokrouhlení při výpočtu YScale. Výpočtem určíme hodnotu vstupní veličiny pro maximální výšku sloupce grafu. Zobrazení sloupce odpovídá maximálnímu napětí:

$$U_{\text{max}} = (48 * 22 - 22) / 100 = 10,34\text{V}.$$

To znamená, že napětí 10,33V bude ještě zobrazeno jako 47 dílků a od 10,34V výše již bude zaplněn celý sloupec.

Graf kontaktu

Na Obr. 6 je zachycen simulovaný průběh spínání kontaktu tak, jak ho ukáže v závislosti na následujících volbách a nastavení simulátor MT201 v prostředí StudioWin.

- Zvolený zobrazovaný rozsah veličiny odpovídá možnostem periferie EX10 a je od 0 do 1. Hodnota síťové proměnné odpovídá hodnotám 0 a 1 z čehož plyne pro svislé zobrazení rozsah vstupní veličiny $1 - 0 = 1$
- Pro zobrazení tohoto rozsahu máme k dispozici 47 bodů. Parametr $\text{Parameters}[2].\text{YScale}$ pak vychází $\text{Parameters}[2].\text{YScale} = 1 / 47 = 0,021$ a to je opět nepřípustná hodnota. Graf lze jen zhušťovat, roztáhnout ho nelze. Proto si musíme opět pomoci úpravou dat přímo při zápisu do datové struktury. Pro úpravu použijeme výraz:

$$\text{Graphs}[2][\text{secondOld}] = (\text{Contact} + 1) * 40.$$

Ze zápisu je vidět, že stav 0 bude ve struktuře reprezentován hodnotou 40 a stav 1 hodnotou 80. Rozsah vstupní veličiny v datové struktuře vychází nyní $80 - 40 = 40$. To je vyhovující hodnota a proto je

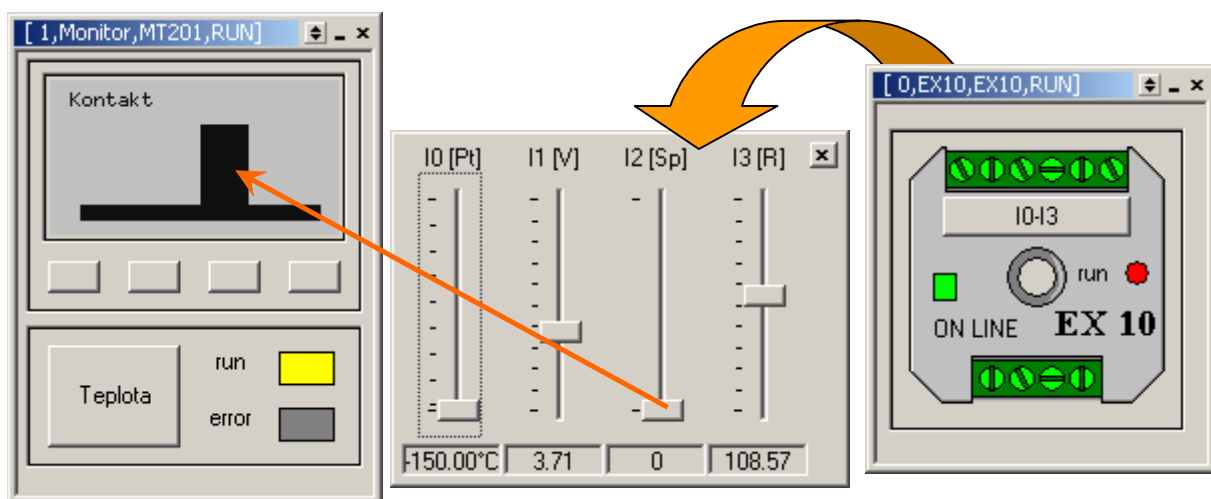
YScale = 1

správná volba.

- Z estetických důvodů požadujeme, aby pod zobrazením stavu 0 bylo v grafu stejné místo jako nad zobrazením stavu 1. Upravíme proto ještě posun grafu o polovinu zbývajících vzdálenosti do plného rozlišení, tj. použijeme výpočet $(48 - 40) / 2 = 4$. Z toho plyne výsledné nastavení:

Paramethers[2].YOffset=40-4=36,

přičemž stav 0 bude zobrazen sloupcem o velikosti $40 - \text{YOffset} = 4$ a stav 1 sloupcem velikosti $80 - \text{YOffset} = 44$.



Obr. 4 Zobrazení grafu kontaktu na simulátoru

Graf odporu

- Zvolený zobrazovaný rozsah veličiny je z demonstračních důvodů o něco menší než jsou možnosti EX10 a to v rozsahu od 11 do 190Ω . Síťová proměnná nabývá tedy hodnot v rozmezí 1100 až 19000. Tomu odpovídá hodnota rozsahu vstupní veličiny $19000 - 1100 = 17900$.
- Pro zobrazení máme k dispozici 47 bodů a z toho plyne hodnota parametru svislého měřítka vypočítaná podle vztahu $\text{Paramethers}[3].\text{YScale} = 17900 / 47 = 380,85$ a zaokrouhlená nahoru na:

YScale[3] = 381

- Pro zobrazení minimální hodnoty 1100 požadujeme jeden dílek grafu, který, jako v předchozích případech, odpovídá právě měřítku $\text{YScale}[3]$. Posun grafu tedy stanovíme na:

$$Y_{\text{offset}[3]} = 1100 - 381 = 719$$

- Nakonec provedeme výpočet vlivu zaokrouhlení hodnoty $Y_{\text{Scale}[3]}$. Výpočtem určíme maximální hodnotu odporu, která bude odpovídat plné výšce sloupce:

$$R_{\text{max}} = (719 + 48 * 381) / 100 = 190,07 \Omega.$$

Protože všechny měřené hodnoty jsou připojeny prostřednictvím síťových proměnných, je třeba ošetřit stav, kdy dojde k odpojení periférie. Nesmí se stát, že by graf vykresloval staré hodnoty síťových proměnných, které periférie již dávno nemodifikuje. To zajistíme jednoduchou konstrukcí, která vždy při posunutí času jednorázově zapíše do síťové proměnné takovou hodnotu, která bude po zápisu do datové struktury posunuta na hodnotu 0 a funkce kreslení grafu ji tak nevykreslí. Po tomto nulování má periférie právě jednu vteřinu na to, aby do síťové proměnné zapsala novou hodnotu a ta se přenesla do datové struktury. Tím je zajištěno, že po odpojení periférie se stávající graf sice posouvá, ale nový se nekreslí. To je možné vyzkoušet v prostředí StudioWin zavřením simulátoru periférie EX10.

Správné zobrazení grafu je odolné i proti výpadku napájení MT201. To platí za podmínky, že výpadek napájení nepřesáhne časový rozsah zaznamenávaných dat tj. 120 sekund. Data se nezapisují přímo na adresu danou hodnotou Second (minute) jako u příkladu MT201, ale prostřednictvím lokálně definované proměnné Time. Hodnota této proměnné je programem posouvána v závěsu za reálným časem a pokud MT201 po nějakou dobu nepracuje, začne proměnná Time za reálným časem zaostávat, neboť ten běží i při výpadku napájení. Po obnovení napájení hodnota Time rychle dožene reálný čas a při tom program vymaže stará data ve struktuře.

Tento algoritmus není funkční zcela na 100%. Mohlo by se stát, že hodnotu síťové proměnné mezi jejím nulováním a následným přepsáním do datové struktury stihne přepsat periférie. Pro účel tohoto příkladu je ale tato konstrukce dostatečně spolehlivá a jednoduchá. I tuto funkci je možné vyzkoušet na simulátoru pomocí funkcí zastavit automat a spustit automat. *Zavření simulátoru a jeho opětovné spuštění nefunguje, neboť zavřením okna simulátoru jsou ztracena veškerá data.*

4.3 Příklad „Demo“

Cílem tohoto příkladu je hlavně ukázat správné začlenění grafu do menu vytvořeného pomocí knihovny Menulib2_MT201. V dalším textu najdete upozornění na zajímavé detaily, které přiblíží jednotlivé funkce i možná úskalí.

Typy zobrazení grafů v příkladu

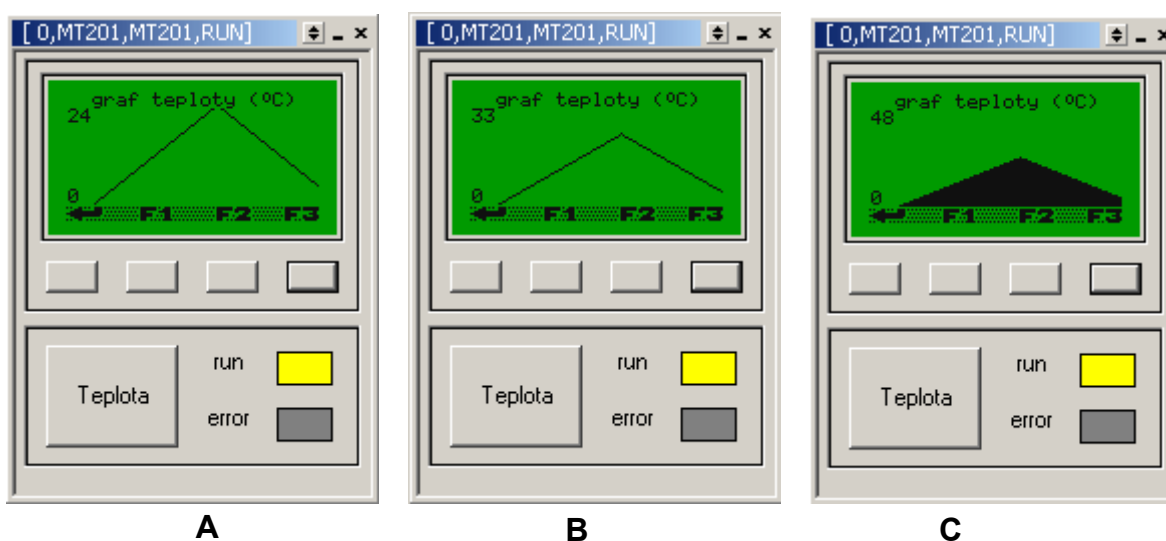
V menu jsou dvě ukázky zobrazení týchž dat. Položka menu „**Graph1-a**“ spouští zobrazení grafu, návrat z grafu do menu je realizován stiskem libovolné klávesy. Zobrazený graf se posouvá vlevo a demonstruje tak zobrazení typu „aktuální časový úsek“.

Položka „**Graph1-b**“ obsahuje zobrazení statického grafu. Zobrazení využívá šesti řádků displeje a to umožňuje horní řádek displeje využít pro libovolný komentář, spodní řádek pak pro uživatelský popis ovládacích tlačítek (viz. Obr. 7).

Funkce kláves jsou:

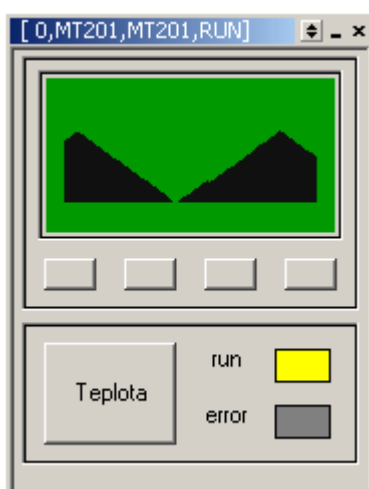
- šipka vlevo - návratová klávesa
- F1 - posun grafu v horizontálním směru
- F2 - cyklická změna horizontálního měřítka
- F3 - cyklická změna vertikálního měřítka

Všimněte si, že při změně vertikálního měřítka (F3) se mění rozsah svislé osy a tím i její popis, který využívá prvních dvou znaků řádku. Graf se tedy tiskne jen na šířku 19 znaků a. šířka grafu tak odpovídá 114 obrazovým bodům přičemž varianty A, B a C na Obr. 7 ukazují důsledky změny měřítka.



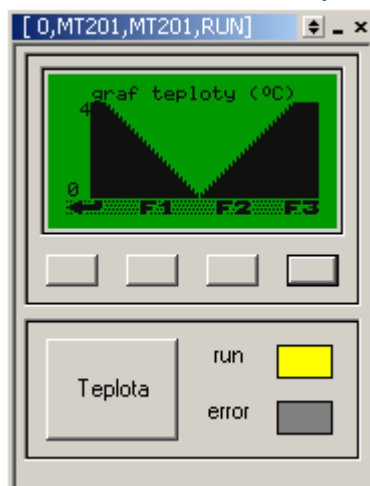
Obr. 5 Dynamická změna svislého měřítka

Jako zdroj dat slouží struktura Graph1, která je naplněna daty ve tvaru jednoho velkého zubu. Data představují nárůst a pokles teploty v rozsahu od 0°C do +25,2°C. Zub dat je navíc modulován signálem typu pila s rozkmitem $\pm 0,2^\circ\text{C}$. Díky tomu je zajímavé měnit měřítka zobrazení a sledovat, jak se modulace projeví na vykresleném grafu (viz.Obr. 7).



Obr. 6 Demonstrace přetečení indexu zobrazování

Zobrazovaná struktura dat obsahuje 504 wordů. Tento počet položek byl zvolen proto, aby byl vidět při zobrazení právě celý graf v případě, že je zobrazován na 126 bodů s vodorovným měřítkem 4, tj. $.XScale=4$ a počet záznamů vychází $4 * 126 = 504$. Zde je zajímavé si všimnout důsledku nevhodného nastavení vodorovného měřítka na hodnotu $.XScale=5$. Dosáhneme nesmyslné situace, kdy data zobrazená na začátku displeje jsou shodná s těmi na konci. Na displeji totiž zobrazujeme 126 hodnot tak, že vybíráme ze struktury vždy každou pátou, čímž dosáhneme přetečení indexu dat přes rozsah pole a to se projeví opětovným zobrazením asi pětiny hodnot. Situaci demonstruje Obr. 8.



Obr. 7 Zubatý graf modulovaných dat pilovým signálem

Detailní zobrazení zubatých dat je vidět pod položkou „**Graph1-b**“ při nastavení měřítka 1 v ose X i v ose Y a při nastavení počátku x na hodnotu 450. V režimu zobrazení „**sloupcový graf**“ je na Obr. 9 patrný detail zubaté hrany dat po modulaci pilou.

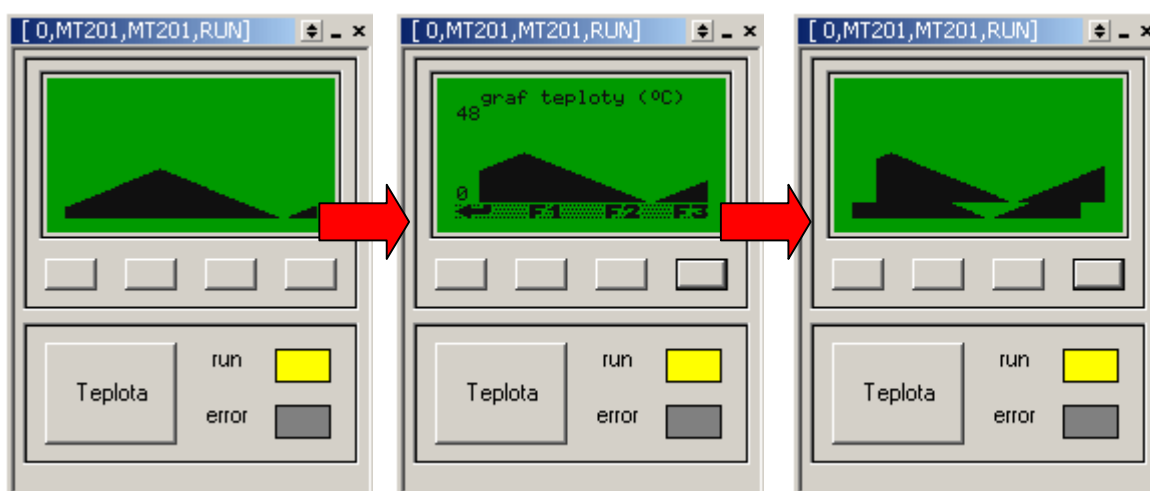
V režimu „bodový graf“ (Obr. 10) se zmíněná modulace projeví tvarem zvláštního dvojitého „V“. Obě zobrazení jsou však logickým důsledkem zobrazovaného tvaru dat, která jsou vlastně tvořena vrcholy poskakujících malých zoubků. V popisovaném případě se jedná o data uměle vytvořená mající tvar, který jen velmi těžko najdeme v běžném provozu. V praxi budou data spojitější a tím pádem vyjde graf přehlednější.



Obr. 8 Zvláštní tvar dvojitého V modulovaných dat

Další zvláštní úkaz nastane v případě, že hned po spuštění simulátoru navštívíme zobrazení grafu přes položku „Graph1-a“ a ihned potom zvolíme druhé zobrazení „Graph1-b“. Když se vrátíme zpět k zobrazení položky „Graph1-a“, zůstane ve spodním řádku pohybující se část starého grafu z položky „Graph1-b“. Situaci dokumentuje Obr. 11. Jev nastává proto, že se díky zobrazení „Graph1-b“ nastavily parametry zobrazované oblasti tak, že graf je vykreslován na displeji o jeden znak výše. Poslední řádek je tak zaplněn znaky z původního zobrazení „Graph1-a“.

Kromě části zobrazované přes položku „Graph1-a“ je všude v menu zapnuto zobrazování popisu uživatelských tlačítek a tak jsou ponechána data přepsána. To ale neplatí při uvedeném přechodu mezi jednotlivými zobrazeními a to má za následek zobrazení zbytku předchozího grafu.



Obr. 9 Pozůstatek grafu při změně počtu řádků grafu

V následujících odstavcích je uveden výpis zdrojového textu příkladu „Demo“. Komentáře jsou uvozeny středníkem a psané kurzívou a jsou vždy na konci řádku, kterého se týkají. V případě, že se do řádku nevejdou, jsou umístěny na samostatném řádku, který předchází komentovaný řádek. Jednotlivé významově odlišné celky zdrojového textu jsou od sebe odděleny komentářem uvozeným a ukončeným hvězdičkami. Deklarace a nastavení proměnných, nezbytných pro použití knihovny kreslení grafů, jsou zvýrazněna žlutým podbarvením.

Příklad:

;velikost datové struktury grafu- povinná součást kódu

```
const GraphMT201DataTypeLenght = 504
```

*;*****deklarace použitých proměnných a datové struktury******

```
var GraphMT201DataType Graph1
```

```
var GraphMT201ParType Paramethers
```

```
table string [2] SIBodTxtTbl = ( "Sloupcový", "Bodový " )
```

```
var word PomIndex, SecondOld
```

```
var byte Strings
```

```
,*****inicializace hodnot po spuštění automatu*****
```

```
if reset then begin
```

```
Parameters.XBegin = 0 ;inicializace počátku grafu ve struktuře Graph1
Parameters.XSize = 126 ;inicializace velikosti oblasti tisku v ose x (počet sloupců)
Parameters.Offset = 2732 ;inicializace posunu grafu ve směru y
Parameters.XScale = 4 ;inicializace měřítka v ose X( bude se zobrazovat každý 4. bod)
;inicializace měřítka v ose y (10 desetin= 1°C=jeden dílek grafu tzn. Rozsah 0..48C)
Parameters.Scale = 10
Parameters.Mode = 0
GraphMT201DispYSize = 6 ; inicializace počtu znakových řádek grafu
GraphMT201DispYBegin = 2 ; inicializuje první použitý řádek grafu
Y30 = 1 ; zapnout podsvícení
```

```
end
```

```
,*****generovani dat struktury Graph1*****
```

```
PomIndex = (PomIndex+1) % 504
```

```
Graph1[PomIndex] = 2732 + 2 * (PomIndex % 2) + PomIndex
```

```
if PomIndex >251 then Graph1[PomIndex] = Graph1[PomIndex] + 504 - 2 * PomIndex
```

```
,*****kód menu a vlastní volání tisku grafu *****
```

```
MeInIt( 0,7 )
```

```
MeTitle( "Demo GraphMT201.lib" )
```

```
if MeNext ( "Graph1-a" ) then begin
```

```
if SecondOld <>Second then begin
```

```
Parameters.XBegin = (Parameters.XBegin + 4) % 504
```

```
SecondOld = Second
```

```
end
```

```
GraphMT201Draw(Graph1, Parameters)
```

```
OpCtrl = 0
```

```
if KBCode > 0 then KBCode = 3
```

```
MeEnd()
```

```
end
```

```
if MeNext ("Graph1-b") then begin
```

```
Position = 0
```

```
Display( " graf teploty (°C)" )
```

```
GraphMT201DispXSize = 57 ;inicializuje počet sloupců grafu
```

```
GraphMT201DispYSize = 6 ;inicializace počtu znakových řádek grafu
```

```
GraphMT201DispYBegin = 1 ;inicializuje první použitý řádek grafu
```

```
Format = 1
```

```
,***** tisk měřítka na svislé ose *****
```

```
Position = 40
```

```
Display((word(Parameters.YScale)*48)/10)
```

```
Position = 80
```

```
Display( " " )
```

```
Position = 120
```

```
Display( " " )
```

```

    Position = 160
    Display( " " )
    Position = 200
    Display( " " )
    Position = 240
    Display( "0 " )
    GraphMT201Draw(Graph1, Parameters)
    OpCtrl = 21 ;nastavení tisku řádku s tlačítky Esc,F1,F2,F3 viz Menulib2_MT201
,*****kód funkce kláves F1, F2 a F3*****
    if KeyCode = 8 then Parameters.XBegin = (Parameters.XBegin + 30 ) % 504
    if KeyCode = 9 then Parameters.XScale = (Parameters.XScale) % 5 + 1
    if KeyCode = 10 then Parameters.YScale = (((Parameters.YScale + 1) / 2) % 5 + 1) * 2 - 1
    MeEnd()
end
,*****kód ručního nastavení parametru*****
Format = 3
if MeLine( "Mode: " ) then MeEdit(Parameters.Mode,0,4)
if MeLine( "XBegin: " ) then MeEdit (Parameters.XBegin,0,504)
if MeLine( "DispXSize : " ) then MeEdit (Parameters.XSize,0,126)
if MeLine( "DispYSize : " ) then MeEdit (GraphMT201DispYSize,2,6)
if MeLine( "DispYBegin: " ) then MeEdit (GraphMT201DispYBegin,0,5)
if MeLine( "XScale: " ) then MeEdit (Parameters.XScale,1,20)
if MeLine( "YScale: " ) then MeEdit (Parameters.YScale,1,20)
if MeLine( "YOffset: " ) then MeEdit (Parameters.YOffset,0,4000)
,*****část menu, která umožňuje prohlídku všech 256 znaků*****
if MeNext( "Prohlídka znaků" ) then begin
    Format = 2
    if MeLine( "Znak kód: " ) then Display(Strings)
    Format = 120
    if MeLine( "Tištěný znak: " ) then Display(Strings)
    OpCtrl = 21
    if KeyCode = 8 then Strings = Strings -1
    if KeyCode = 9 then Strings = Strings +1
    if KeyCode = 10 then Strings = Strings +10
    MeEnd()
end
MeEnd()
Reset = 0
end

```

5 Závěr

Základní část tohoto manuálu ukázala nejtypičtější možnosti použití knihovny GraphMT201 v příkladech, kde nejsou zapotřebí detailní znalosti funkce knihovny. Použití knihovny je velmi snadné a kód z jednotlivých příkladů je možné po minimálních úpravách použít v libovolných aplikacích. Při realizaci jakéhokoli grafu je důležité zejména určení správného vertikálního měřítka, které představuje rozsah zobrazovaných hodnot a stejně tak i správnou volbu zobrazovaného časového úseku představovaného měřítkem horizontálním. Významný je v praxi také způsob pořizování dat. Je možné zaznamenávat vzorky v pravidelných intervalech, střední hodnotu, maxima, minima a existuje i celá řada aproximačních metod, které lze přímo při pořizování dat implementovat. Na prvním místě by však vždy mělo být hledisko užitečnosti zobrazovaného grafu pro konkrétního uživatele.

II. Pokročilé možnosti použití knihovny

1 Úvod

Pokročilé možnosti použití knihovny se dají rozdělit do dvou skupin. Je to jednak možnost tisku více grafů současně, jak demonstruje příklad „Čtyři grafy“ a pak již následuje vyloženě nestandardní použití knihovny.

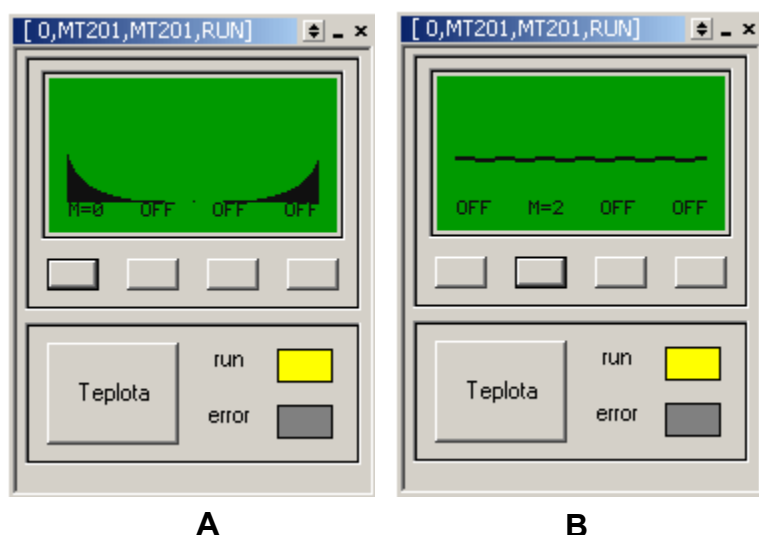
Tisk více grafů současně umožní:

- současný tisk průběhu skutečné a žádané hodnoty regulovaného okruhu – zobrazení přechodové charakteristiky
- současný tisk regulované veličiny a výstupu regulátoru
- současný tisk minima, maxima a střední hodnoty sledované veličiny, přičemž vhodná volba pro tisk minima a maxima je sloupcový graf, pro střední hodnotu je výhodný čárový graf

Nestandardní použití knihovny spočívá v tisku takových grafů a zobrazovačů, pro které není vhodné přímé použití tisku z datových struktur. Následující kapitoly textu uvádějí dva takové příklady. Ani touto částí manuálu však nejsou vyčerpány všechny možnosti použití knihovny a cílem tak spíše je ukázka alternativních postupů při práci s grafickým displejem a knihovnou GraphMT201.lib.

2 Příklad „Čtyři grafy“

Cílem tohoto příkladu je ukázat možnost tisku více grafů „přes sebe“ a demonstrovat kombinace různých typů tisku jednotlivých grafů. Princip tisku více grafů současně spočívá v invertování barvy společných bodů, tzn. čárový graf je tištěn černě a v místech, kde graf prochází

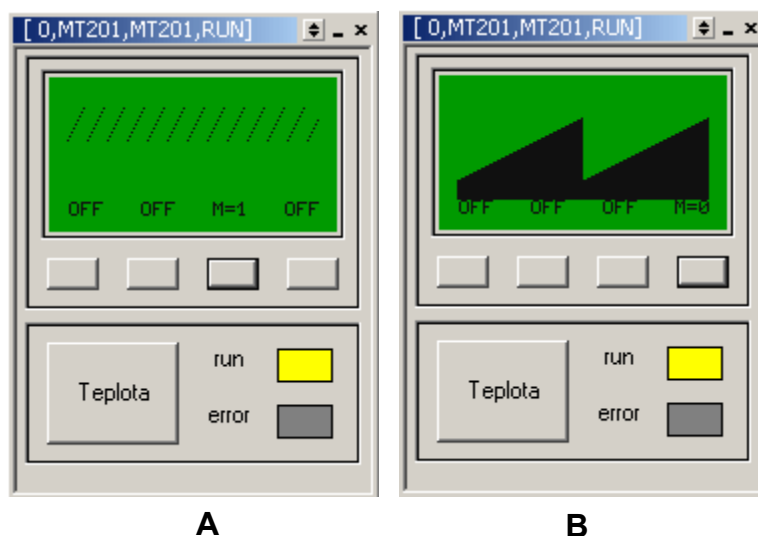


Obr. 10 Graf 0 - mód 0, sloupcový (A), Graf 1 - mód 2, čárový, šířka čáry 2 (B)

jiným např. sloupcovým černě vykresleným grafem je tištěn inverzně tj. bíle. Další možností, jak od sebe odlišit různé grafy, je změna šířky čáry jednotlivých grafů. Čtveřici zobrazovaných grafů příkladu v různých módech ukazuje Obr. 12 a Obr. 13 ve variantách A a B.

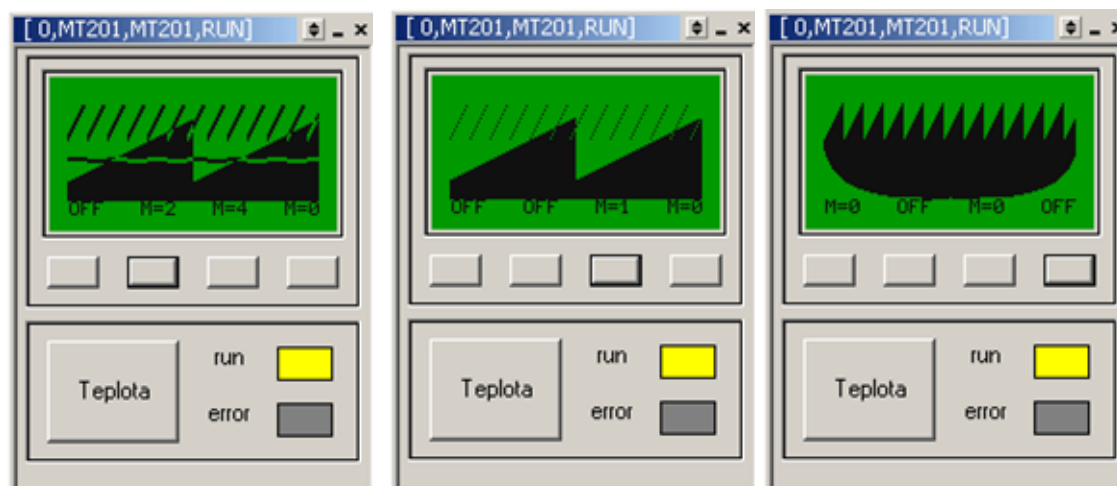
Příklad obsahuje čtyři samostatné grafy v poli struktur Graphs, datové struktury jsou vygenerovány pevně výpočtem. Ke každému grafu náleží jedna proměnná Mode_0...Mode_3 (dále Mode_x) a ta určuje režim tisku příslušného grafu.

Proměnné Mode_x mohou nabývat hodnot 0...5 ve významu 0 pro sloupcový graf, 1..4 pro šířka čáry čárového grafu a 5 pro stav, kdy graf není vykreslen. Módy zobrazení čtyř grafů lze přepínat čtyřmi tlačítky MT201, přičemž každému grafu přísluší jedno tlačítko. Cyklickým tisknutím ovládací klávesy se postupně mění mód tisku grafu za současné indikace jeho čísla na posledním



Obr. 11 Graf 2 - mód 1, čárový, šířka čáry 1 (A), Graf 3 - mód 0 sloupcový

řádku displeje. Postupným zapínáním jednotlivých grafů lze vyzkoušet různé kombinace tisku grafů. Asi nejnepřehlednější by bylo tištění všech grafů v režimu sloupcový graf. Různé varianty tisku ukazuje Obr. 14 s tím, že mód tisku jednotlivých grafů je indikován přímo na displeji MT201



Obr. 14 Různé kombinace tisku grafů „přes sebe“

Zajímavosti v kódu

Ve vlastním kódu si všimněte, že:

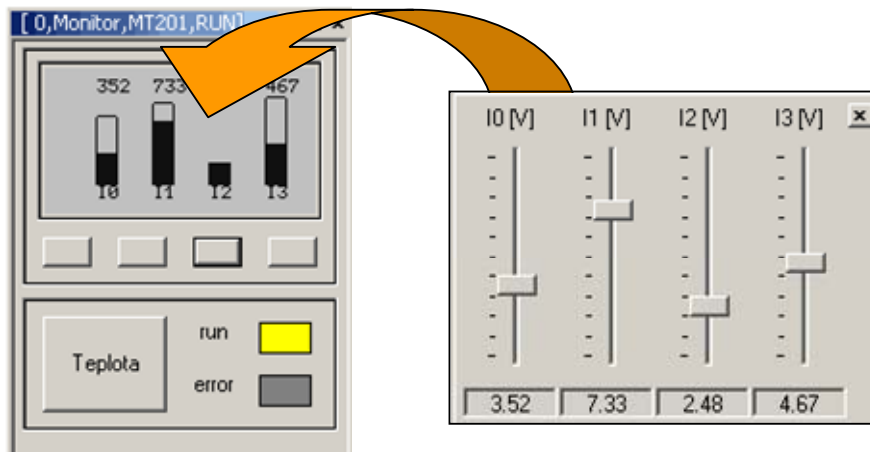
- Všechny grafy mají společné parametry
- Parametr .Mode je vždy před voláním kreslicí procedury nastaven příslušnou proměnnou Mode_x
- K parametru .Mode je v prvních třech případech přičtena hodnota 0x80 pro nastavení nejvyššího bitu bytu .Mode. Toto nastavení v knihovně způsobí potlačení inkrementace proměnné GraphMT201PointIndex, která nabývá v průběhu vykreslování hodnot od 0 do 125 a

určuje tak bod grafu, který bude zpracováván při následujícím volání funkce GraphMT201Draw. Před tiskem posledního grafu není tento bit nastaven a proměnná GraphMT201PointIndex je tudíž v závěru tisku inkrementována, čímž se posune zpracování dat k následujícímu bodu grafu.

- Je-li proměnná Mode_x = 5, pak je parametr .YOffset=1000, čímž je vlastně zajištěno vypnutí tisku grafu, neboť jsou všechny zpracovávány hodnoty menší než 1000.

3 Příklad „EX10 - monitor“

K terminálu MT201 je připojena periférie EX10, která měří čtyři napětí U0..U3 a síťovými proměnnými D32..D35 je předává terminálu MT201 k zobrazení. Síťové proměnné jsou v projektu pojmenovány přímo U0..U3, takže v programu je přístup k předávaným hodnotám přehledný. Zobrazení čtyř napětí na MT201 je formou čtyř vyplněných sloupcových grafů, průběžně vyhodnocuje maximum jednotlivých napětí a toto maximum zobrazuje do téhož sloupce formou nevyplněného sloupcového grafu. Nad každým grafem je zobrazeno aktuální napětí. Resetovat maxima lze stiskem jakéhokoli tlačítka terminálu. Vše je dobře viditelné na simulátoru a zobrazení grafu v průběhu simulace demonstruje Obr. 15.



Obr. 15 Zobrazení sloupcového grafu v simulaci příkladu

Je zřejmé, že lineární datová struktura typu GraphMT201DataType je pro tento typ úlohy nevhodná. K zobrazení grafu potřebujeme ve skutečnosti jen osm hodnot – čtyři aktuální napětí a čtyři maxima. Kdybychom chtěli použít standardní datovou strukturu, museli bychom je použít dvě a to jednu pro aktuální napětí a druhou pro maxima. Dále bychom museli poměrně složitě kopírovat zobrazované položky do datových struktur. To by bylo pomalé a datové struktury by zcela zbytečně zabíraly paměť.

Budeme tedy graf tisknout zcela odlišným principem než dosud. Klasickou, ve smyslu tohoto manuálu, datovou strukturu vlastně budeme potřebovat jen pro předávání aktuální tištěné hodnoty. Tuto hodnotu vypočítá procedura podle změřených dat a podle aktuálně tištěného bodu grafu označeného proměnnou GraphMT201PointIndex. Pro takové předávání dat stačí datová struktura „VirtualData“ velikosti jeden word. Kreslicí procedura GraphMT201Draw bude tedy vždy zobrazovat hodnotu, kterou jí pomocí této zvláštní jednopoložkové virtuální struktury předáme.

V kódu programu si všimněte:

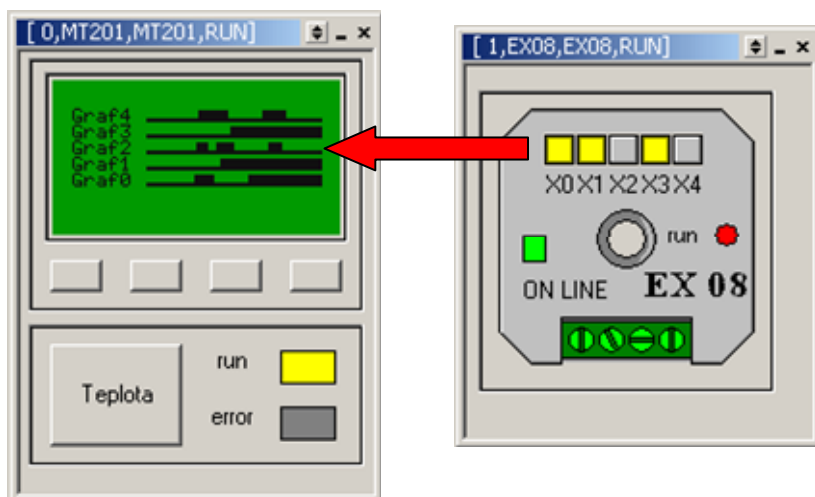
- vstupy U0...U3 jsou kopírovány do pole U[0]...U[3]. To je proto, že síťové proměnné typu word nelze indexovat, v případě jejich přímého použití bychom v programu museli psát zvláštní kód pro obsluhu každé z nich.

- Maximální hodnoty jsou průběžně ukládány do pole UMax[0]...UMax[3]. Ukládání maxim je prováděno společným kódem, kde je právě využito indexování pomocí proměnné „index“.
- Stejněho způsobu indexování je využito při tisku aktuálních napětí do prvního řádku displeje MT201.
- Způsob tisku je řízen v proceduře PrintGraph, kde je buď tištěna nulová hodnota (mezi jednotlivými sloupci) a nebo aktuální hodnota pomocí sloupcového grafu. Je-li maximum větší než aktuální hodnota, jsou okraje sloupce tištěny až do příslušné maximální hodnoty a uvnitř sloupce grafu je dotištěna maximální hodnota jako čárový graf při nastavení .Mode=0x81.
- GraphMT201DispXSize = 108, což představuje jen osmnáct znaků a proto na displeji zbývají vlevo ještě tři nevyužité znaky.

Tuto úlohu lze snadno demonstrovat na simulátoru v prostředí StudioWin.

4 Příklad „EX08“

K terminálu MT201 je připojena periférie EX08, která disponuje pěti digitálními vstupy. Stav všech vstupů vysílá po síťové proměnné D32 pojmenované EX08Inputs. Úkolem této úlohy je zobrazit všech pět průběhů digitálních vstupů na terminálu MT201 tak, aby se nepřekrývaly a graf tak znázorňoval přehledným způsobem všechny vstupy současně, jak ukazuje Obr. 16. Hodnoty vstupů budeme zaznamenávat po jedné sekundě. Aby se na obrazovku vešel popis grafů budeme zobrazovat interval 90 sekund.



Obr. 16 Zobrazení logických průběhů na pěti vstupu EX08

Na Obr. 16 je zřetelně vidět, že zapnuté vstupy X0, X1 a X3 mají odezvu v zobrazení grafu na pravé straně, která představuje současnost. Směrem vlevo od současnosti vidíme historii hodnoty jednotlivých signálů.

I pro řešení této úlohy není vhodný klasický přístup. Pro ten bychom potřebovali pět datových struktur wordů s tím, že v každém wordu by byl využit jen jeden bit. Ani vlastní tisk by nebyl zcela jednoduchý. Proto při řešení opět využijeme jednopoložkovou strukturu VirtualData. Do ní je při inicializaci napevno zapsána hodnota 100. Procedura SetParameters nastaví aktuální parametry pro tisk příslušného grafu v závislosti na datové struktuře „Data“ a aktuálním tištěným bodu grafu daném hodnotou proměnné GraphMT201PointIndex.

5 Závěr

Nadstavbová část tohoto manuálu ukázala některé další možnosti práce s knihovnou GraphMT201.lib. Tím ale zajímavé možnosti nekončí. Pro nestandardní zobrazení grafů nabízíme na závěr několik tipů. Všechny předpokládají naplněnou datovou strukturu „Data“ a správně nastavené parametry ve struktuře „Paramethers“ tak, že graf je správně zobrazen. Vyzkoušejte místo běžného volání kreslicí procedury některý z následujících kódů.

Šrafovaný graf

Graf bude tištěn jako šrafovaný tehdy, pokud Jeden bod grafu bude vždy vykreslen jako čárový graf, druhý jako bodový atd.

```
„Paramethers.Mode=0  
GraphMT201Draw (Data, Paramethers)  
Paramethers.Mode=1  
GraphMT201Draw (Data, Paramethers)“
```

Graf s proměnnou šířkou čáry

Graf bude střídavě vykreslen silnější a slabší čarou. Efekt se bude pohybovat, protože procedura GraphMT201Draw je v hlavní programové smyčce volána pětkrát a čísla 5 a 126 (počet bodů grafu) jsou nesoudělná. Proto se silnější čára začne vždy vykreslovat z jiného místa grafu.

```
„Paramethers.Mode=1  
GraphMT201Draw (Data, Paramethers)  
GraphMT201Draw (Data, Paramethers)  
Paramethers.Mode=4  
GraphMT201Draw (Data, Paramethers)  
GraphMT201Draw (Data, Paramethers)  
GraphMT201Draw (Data, Paramethers)“
```

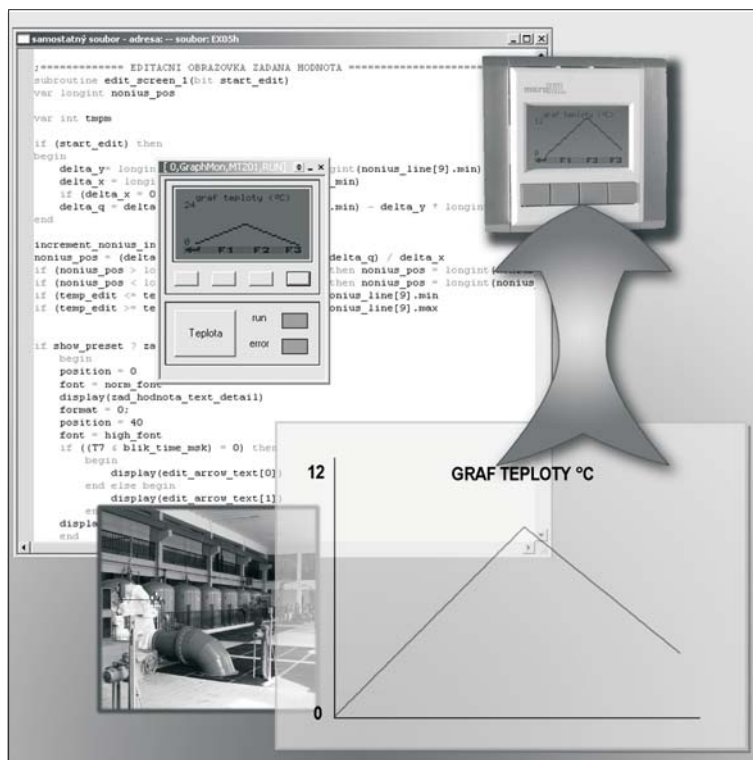
Graf s trojúhelníhovým štítem

V tomto případě bude na graf „přikreslen trojúhelníkový štít“. Efekt se nebude pohybovat, neboť je odvozen právě od tištěné pozice představované proměnnou GraphMT201PointIndex.

```
„Paramethers.Mode=GraphMT201PointIndex % 10 +1  
GraphMT201Draw (Data, Paramethers)“
```

Co dodat ?

Knihovna GraphMT201 je velmi silný nástroj, který v principu umožňuje tisk libovolné kombinace svislých úseček šíře jednoho grafického bodu. Tak je možné knihovnu využít pro tisk jakéhokoli obrázku. Zatím nenaplněnou výzvou pro programátory zůstává využití GraphMT201.lib pro 3D-animaci...



ZOBRAZENÍ GRAFŮ NA MT201

POPIS KNIHOVNY PRO TISK GRAFŮ NA MT201

edice 3.2008

1.verze dokumentu

© MICROPEL 2008, všechna práva vyhrazena
kopírování dovoleno jen bez změny textu a obsahu

<http://www.micropel.cz>