

# Knihovna CanvasObjectsLib

**TXV 003 97.01  
první vydání  
duben 2016  
změny vyhrazeny**

## Historie změn

Datum	Vydání	Popis změn
Duben 2016	1	První vydání, popis odpovídá CanvasObjectsLib_v10

## OBSAH

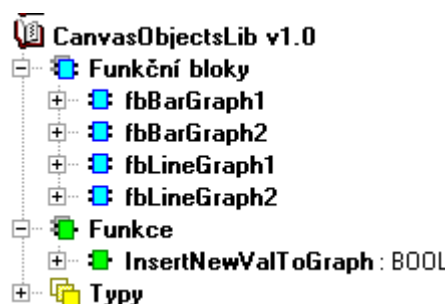
<b>1 Úvod</b>	<b>3</b>
1.1 Princip práce knihovny	4
<b>2 Grafy</b>	<b>6</b>
2.1 Čárový graf pro jeden průběh fbLineGraph1	8
2.2 Čárový graf pro dva průběhy fbLineGraph2	12
2.3 Sloupcový graf pro jeden průběh fbBarGraph1	19
2.4 Sloupcový graf pro dva průběhy fbBarGraph2	22
2.5 Funkce InsertNewValToGraph	25

## 1 ÚVOD

Knihovna CanvasObjectsLib obsahuje podporu pro práci s grafickými objekty, které jsou dostupné přes web server v PLC.

Knihovna CanvasObjectsLib je standardně dodávaná jako součást programovacího prostředí Mosaic od verze v2016.1. Pokud chceme funkce z knihovny CanvasObjectsLib použít v aplikačním programu PLC, je třeba nejprve přidat tuto knihovnu do projektu. Současně s knihovnou CanvasObjectsLib se do projektu automaticky přidají knihovny StdLib, ColorLib, CrcLib, CanvasLib a ToStringLib, protože knihovna CanvasObjectsLib využívá některé funkce z těchto knihoven.

Následující obrázek ukazuje strukturu knihovny CanvasObjectsLib v prostředí Mosaic



Knihovna CanvasObjectsLib není podporovaná na systémech TC-650, u systému TC700 nelze knihovnu použít s procesorovými moduly CP-7002, CP-7003 a CP-7005.

Funkce z knihovny CanvasObjectsLib jsou podporovány v centrálních jednotkách řady K a L (TC700 CP-7004 a CP-7007, všechny varianty systému Foxtrot) od FW verze v9.6.

Objednací číslo dokumentace ke knihovně CanvasObjectsLib je TXV 003 97.01.

## 1.1 Princip práce knihovny

Funkční bloky v knihovně CanvasObjectsLib podporují kreslení grafických objektů umístěných ve web stránce vytvořené nástrojem WebMaker v prostředí Mosaic. Grafickým objektem může být například čárový nebo sloupcový graf, posuvník (slider), apod.

Každý funkční blok z této knihovny produkuje sadu grafických příkazů, která zajistí nakreslení příslušného objektu. Příkazy jsou uloženy do zásobníku (bufferu), který je výstupem funkčního bloku. Pokud chceme objekt zobrazit ve web stránce, pak v nástroji WebMaker vložíme do stránky prvek „Kreslicí plátno“ (canvas) a jako řídící strukturu tohoto prvku uvedeme zásobník příkazů vygenerovaný příslušným funkčním blokem.

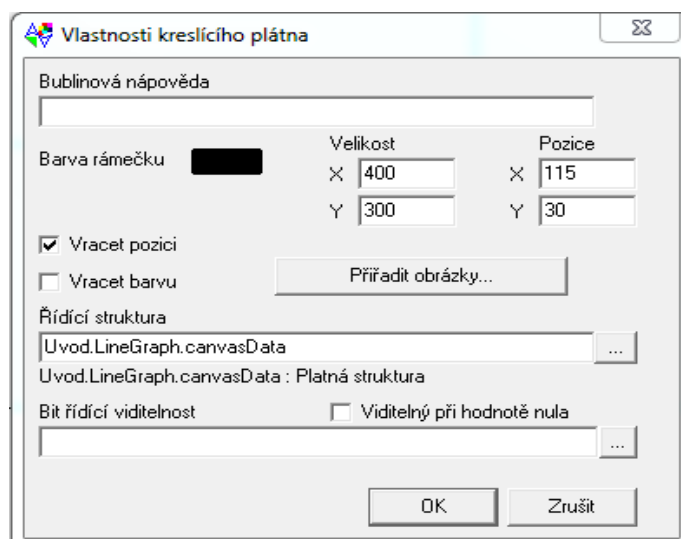
Uvedený princip ilustruje následující příklad. Předpokládejme, že chceme zobrazit čárový graf, ukazující průběh teplot během dne. Teploty budou ukládány každou hodinu do pole (*temp\_samples*), které budeme zobrazovat v grafu. Pro zobrazení grafu založíme v programu instanci funkčního bloku *fbLineGraph1*, která zpracuje pole teplot *temp\_samples* a vygeneruje potřebné příkazy pro nakreslení grafu (do svého výstupu *canvasData*).

```
PROGRAM prgUvod
VAR
  temp_samples : ARRAY[1..24] OF REAL :=
  [ 2.2, 2.2, 2.3, 2.6, 3.2, 3.3, 3.6, 3.5, 4.0, 4.3, 4.4, 4.1,
    4.1, 4.4, 4.7, 5.5, 5.4, 5.2, 4.7, 5.4, 6.3, 6.1, 5.9, 6.7 ];

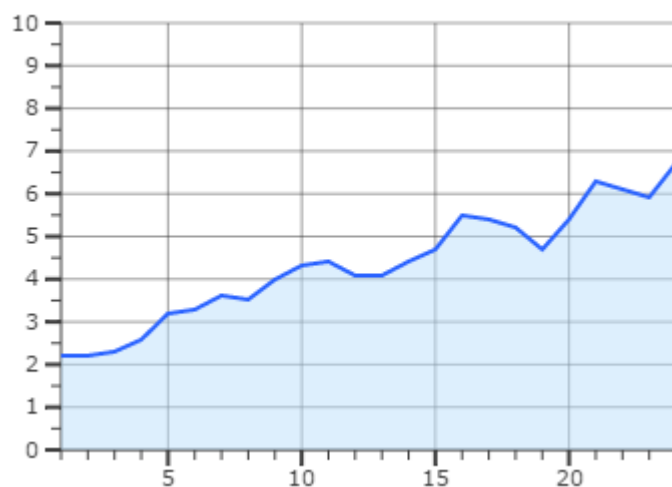
  LineGraph    : fbLineGraph1 := ( minY := 0.0, maxY := 10.0,
                                   numPoints := 24, maxPoints := 24 );
END_VAR

LineGraph ( bufY := void( temp_samples));
END_PROGRAM
```

Poté v nástroji WebMaker přidáme do web stránky prvek „Kreslicí plátno“ a ve vlastnostech tohoto prvku nastavíme výstup bloku *LineGraph.canvasData* jako řídící strukturu.



Nyní je třeba přeložit projekt a nahrát ho do PLC systému. Pak stačí zobrazit web stránku z PLC v prohlížeči. Prohlížeč musí podporovat standard HTML5.



Knihovna CanvasObjectsLib obsahuje podporu pro následující objekty:

- čárové grafy
- sloupcové grafy

## 2 GRAFY

V knihovně CanvasObjectsLib je podpora pro následující typy grafů:

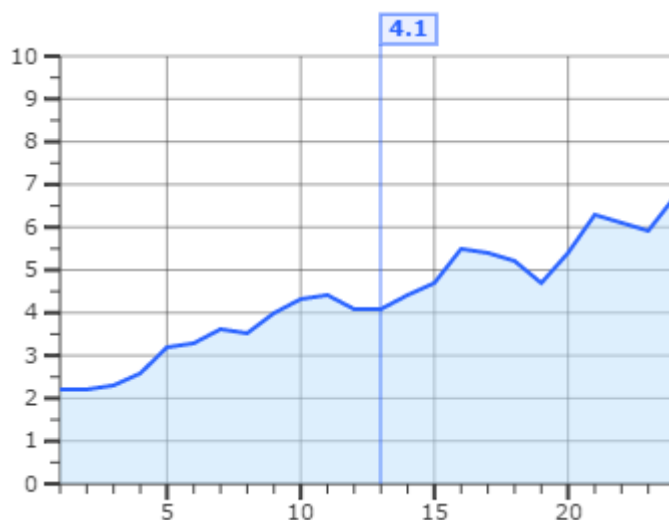
- čárové grafy
- sloupcové grafy

Grafy umožňují zobrazit jeden nebo dva průběhy. Každý průběh může mít maximálně 64 zlomových bodů resp. sloupců. Grafy jsou nakresleny automaticky tak, aby se optimálně vyplnil rozměr kreslicího plátna (zadaný v nástroji WebMaker). Velikost zobrazeného grafu lze tedy jednoduše měnit změnou velikosti kreslicího plátna.

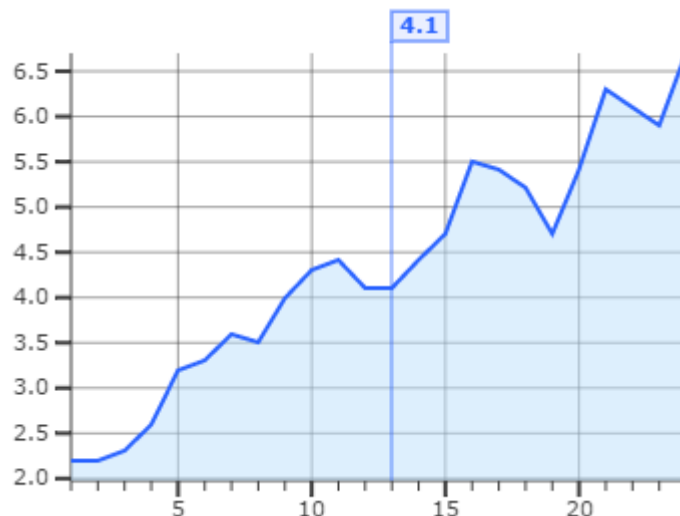
Vstupy funkčních bloků pro kreslení grafů lze rozdělit do dvou skupin:

- **ovládací vstupy**
  - vis* viditelnost grafu
  - refr* překreslení grafu
  - fit* nastavení osy Y podle min a max zobrazené hodnoty
  - zoom2x* zvětšení osy X 2x + fit osy Y (lupa)
- **konfigurační vstupy**
  - minX, maxX* rozsah osy X
  - minY, maxY* rozsah osy Y
  - ratio* poměr šířka : výška grafu
  - axisColor* barva, kterou jsou nakresleny osy grafu
  - textSize* velikost textu (popisky os)
  - maxPoints* max. počet hodnot zobrazených v grafu (šířka grafu)
  - numPoints* aktuální počet hodnot zobrazených v grafu
  - graphColor* barva grafu
  - fillColor* barva výplně grafu
  - graphStyle* styl grafu

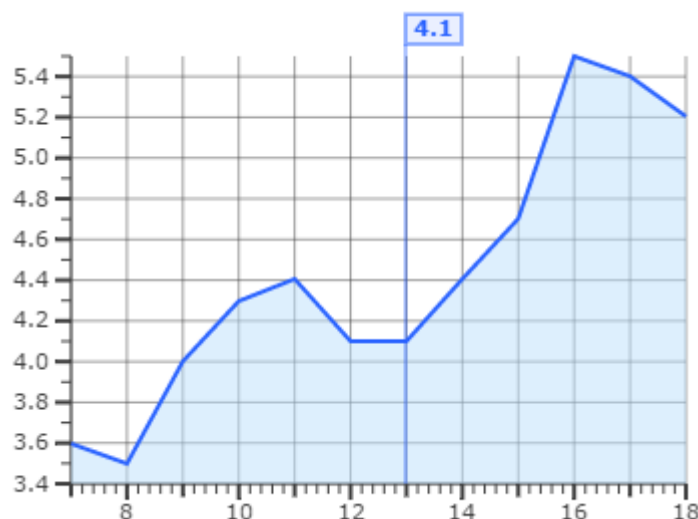
Všechny grafy umožňují položit do grafu měřicí značku (kliknutím myši do prostoru grafu), na které se zobrazí hodnota z grafu.



Pokus je nastaven vstup `fit = TRUE`, pak se změní měřítko osy Y podle minimální a maximální zobrazené hodnoty. Graf z předchozího obrázku bude vykreslen následovně:



Čárové grafy umožňují 2x zvětšit graf nastavením vstupu `zoom2x = TRUE`. Graf pak bude vypadat následovně:



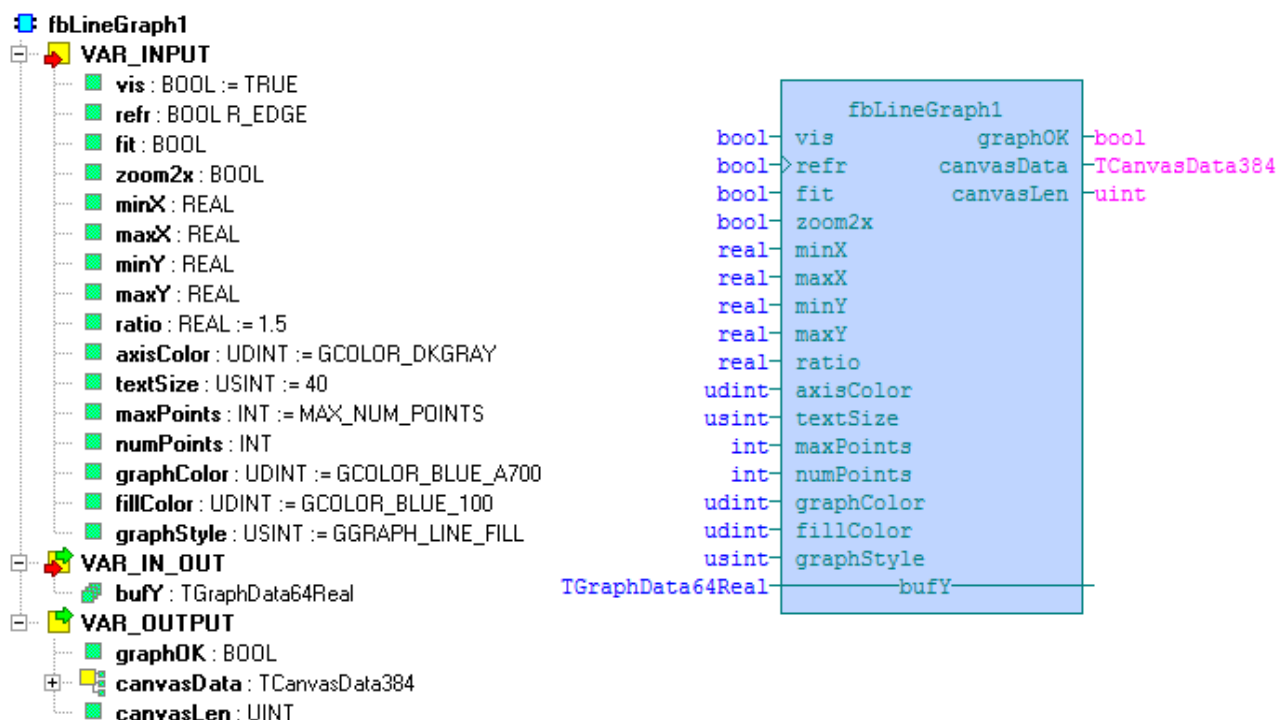
Při zvětšení se 2x zvětší měřítko osy X. Oblast zvětšení se řídí položenou měřicí značkou (snahou je vykreslit okolí položené značky). Pokud značka není položena, pak se zobrazí začátek grafu. Osa Y se přizpůsobí minimální a maximální zobrazené hodnotě.

Měřicí značku lze skrýt kliknutím mimo oblast zobrazeného grafu.  
Mřížka grafu je kreslena automaticky, pozadí grafu je průhledné.

Barvy os, popisků, čáry a výplně grafu lze měnit pomocí konfiguračních vstupů (konstanty pro barvy a styl grafu viz knihovna CanvasLib). Ukázky v této kapitole byly nakresleny blokem `fbLineGraph1` s výchozím nastavením většiny parametrů (viz program `prgUvod` v předchozí kapitole).

## 2.1 Čárový graf pro jeden průběh fbLineGraph1

Knihovna : CanvasObjectsLib



Funkční blok *fbLineGraph1* sestaví zásobník příkazů pro zobrazení čárového grafu, ve kterém může být maximálně 64 hodnot (zlomových bodů). Hodnoty zobrazené v grafu jsou očekávány v proměnné *bufY*. Příkazy, které zařídí vykreslení grafu ve web stránce, jsou uloženy ve výstupu *canvasData*.

Popis proměnných :

	Proměnná	Typ	Význam
<b>VAR_INPUT</b>			
	<i>vis</i>	BOOL	Viditelnost grafu True znamená zobrazit graf (výchozí hodnota)
	<i>refr</i>	BOOL R_EDGE	Na náběžnou hranu této proměnné dojde k překreslení grafu
	<i>fit</i>	BOOL	Umožňuje změnit rozsah osy Y podle zobrazených dat Je-li FALSE, osa Y má rozsah daný vstupy <i>minY</i> , <i>maxY</i> Je-li TRUE, minimum a maximum na ose Y odpovídá minimální a maximální hodnotě zobrazené proměnné
	<i>zoom2x</i>	BOOL	Zvětšení grafu Je-li FALSE, osa X je zobrazena podle vstupů <i>minX</i> , <i>maxX</i> Je-li TRUE, osa Y změní rozsah jako při <i>fit</i> = TRUE a osa X zmenší rozsah 2x, čímž dojde ke zvětšení grafu (lupa)
	<i>minX</i> , <i>maxX</i>	REAL	Meze udávající rozsah osy X při <i>minX</i> = <i>maxX</i> = 0 je rozsah osy X v intervalu <1, <i>maxPoints</i> >



	<b>Proměnná</b>	<b>Typ</b>	<b>Význam</b>
	<i>minY, maxY</i>	REAL	Meze udávající rozsah osy Y Platí pro <i>fit</i> = 0 a <i>zoom2x</i> = 0
	<i>ratio</i>	REAL	Poměr šířky a výšky grafu ve web stránce – měl by odpovídat zvoleným velikostem prvku „Kreslicí plátno“ Výchozí hodnota je 1.5 (např. pro 300x200 pixelů) Přípustná hodnota musí být v intervalu <1.0, 5.0>
	<i>axisColor</i>	UDINT	Barva, kterou jsou nakresleny osy a popisky Výchozí hodnota je GCOLOR_DKGRAY
	<i>textSize</i>	USINT	Velikost textu pro popisky Texty popisků se automaticky přizpůsobují zvoleným rozměrům grafu (prvku „Kreslicí plátno“) - pokud automaticky nastavená velikost písma nevyhovuje, lze ji změnit zvětšením resp. zmenšením hodnoty v tomto parametru Výchozí hodnota je 40
	<i>maxPoints</i>	INT	Max. počet hodnot zobrazitelných v grafu Výchozí hodnota 64, maximální hodnota 64. Tento parametr definuje kolik hodnot lze v grafu zobrazit
	<i>numPoints</i>	INT	Počet aktuálně zobrazených hodnot Tento parametr definuje, z kolika hodnot je nakreslena čára grafu
	<i>graphColor</i>	UDINT	Barva, kterou je nakreslena čára grafu Výchozí hodnota GCOLOR_BLUE_A700
	<i>fillColor</i>	UDINT	Barva výplně grafu Platí pro <i>graphStyle</i> = GGRAPH_LINE_FILL Výchozí hodnota GCOLOR_BLUE_100
	<i>graphStyle</i>	USINT	Styl grafu Výchozí hodnota GGRAPH_LINE_FILL (graf s výplní)
<b>VAR_IN_OUT</b>			
	<i>bufY</i>	TGraph-Data64-Real	Proměnná obsahující hodnoty, které budou zobrazeny v grafu (maximálně 64 hodnot typu REAL)
<b>VAR_OUTPUT</b>			
	<i>graphOK</i>	BOOL	TRUE znamená, že se podařilo úspěšně sestavit všechny příkazy potřebné k vykreslení grafu
	<i>canvasData</i>	TCanvas-Data384	Buffer příkazů pro vykreslení grafu
	<i>canvasLen</i>	UINT	Aktuální délka příkazů v <i>CanvasData</i>

## Příklad 1

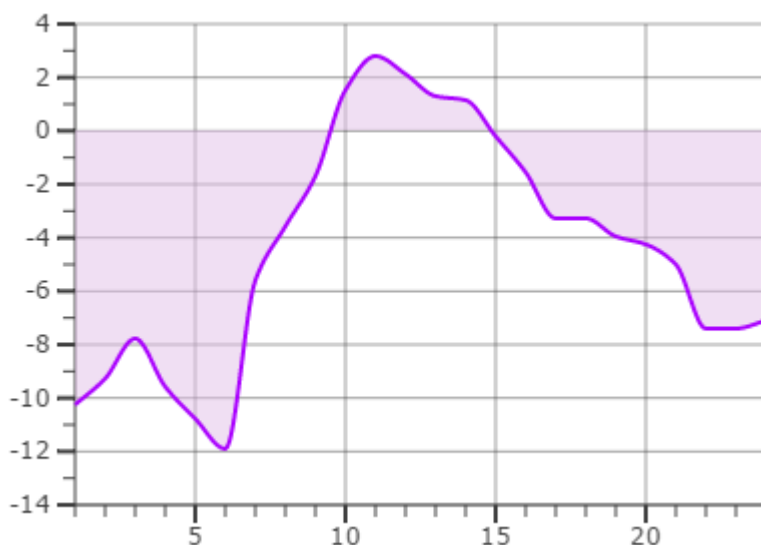
Zobrazení denních teplot čárovým grafem. Naměřené teploty jsou uloženy v poli `temp_2009_02_21`. Počet zobrazených hodnot je 24. Barva grafu je fialová `G_COLOR_PURPLE_A700`, zapnutá interpolace mezi vzorky grafu (`GGRAPH_LINE_INTERPOLATE`) a výplň grafu (`GGRAPH_LINE_FILL`). Barva výplně je `G_COLOR_PURPLE_100`. Osa X má automatické číslování (parametry `minX` a `maxX` nejsou v deklaraci `LineGraph1` vyplněny). Osy a popisky jsou vykresleny výchozí barvou a velikostí.

```
PROGRAM prgExample1
  VAR CONSTANT
    GRAPH_TEMP_SIZE : INT := 24;    // max pocet bodu grafu
  END_VAR
  VAR
    temp_2009_02_21 : ARRAY[1..GRAPH_TEMP_SIZE] OF REAL :=
      [ -10.3, -9.3, -7.8, -9.6, -10.8, -11.9, -5.6, -3.6, -1.7, 1.5, 2.8, 2.1,
        1.3, 1.1, -0.2, -1.6, -3.3, -3.3, -4.0, -4.3, -5.0, -7.4, -7.4, -7.1 ];

    LineGraph1 : fbLineGraph1 := ( minY := -14.0, maxY := 4.0,
                                   maxPoints := GRAPH_TEMP_SIZE,
                                   numPoints := GRAPH_TEMP_SIZE,
                                   graphColor := G_COLOR_PURPLE_A700,
                                   fillColor := G_COLOR_PURPLE_100,
                                   graphStyle := GGRAPH_LINE_INTERPOLATE + GGRAPH_LINE_FILL );
  END_VAR

  LineGraph1 ( bufY := void( temp_2009_02_21 ) );
END_PROGRAM
```

Prvek „Kreslicí plátno“ má nastaven rozměr 450x300 ( `ratio := 1.5` což odpovídá výchozí hodnotě) a jeho řídicí struktura je `Example1.LineGraph1.canvasData` a pole „Vrať pozici“ je zaškrtnuté (to je nutné pro pokládání měřící značky do grafu).



Nastavením proměnné `Example1.LineGraph1.zoom2x = TRUE` zvětšíme graf.

## Příklad 2

Zobrazení průměrných denních teplot v měsíci čárovým grafem. Naměřené teploty jsou uloženy v poli `temp_samples`. Počet naměřených a zobrazených hodnot je 25, graf je zkonfigurován na zobrazení max. 31 hodnot. Barva grafu je `GCOLOR_LIGHT_BLUE_A700`, interpolace mezi vzorky grafu není zapnutá (`GGRAPH_LINE_DEFAULT`), graf je bez výplně. Osa X má automatické číslování (parametry `minX` a `maxX` nejsou v deklaraci `LineGraph2` vyplněny). Osy a popisky jsou vykresleny barvou `GCOLOR_WHITE`, velikost popisků je mírně zvětšená (`textSize := 60`). Graf má zapnuté automatické přizpůsobení osyY (`fit := 1`).

```
PROGRAM prgExample2
VAR
  temp_samples : ARRAY[1..31] OF REAL :=
  [ 2.2, 2.2, 2.3, 2.6, 3.2, 3.3, 3.6, 3.5, 4.0, 4.3, 4.4, 4.1, 4.1, 4.4, 4.7,
    5.5, 5.4, 5.2, 4.7, 5.4, 6.3, 6.1, 5.9, 6.7, 7.4, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0 ];

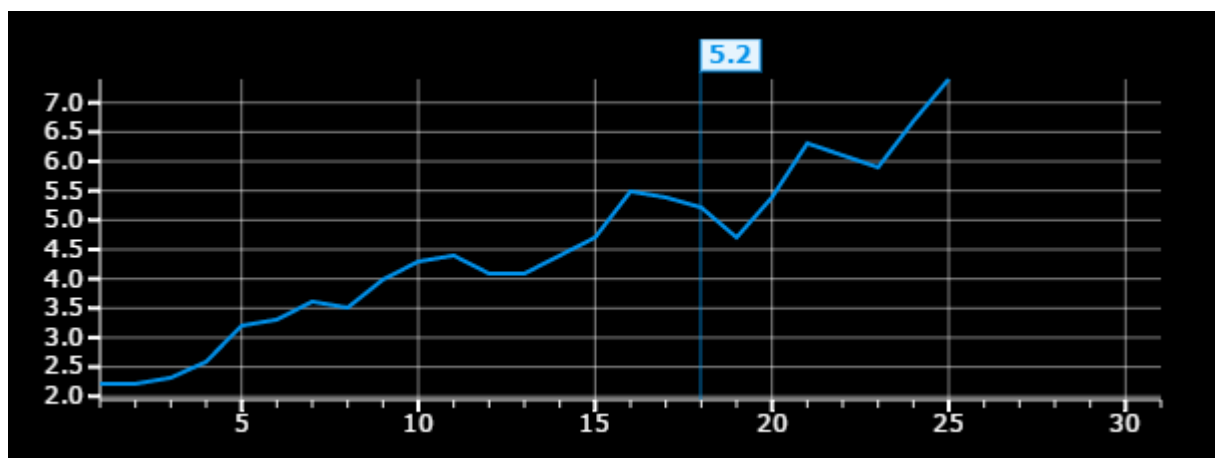
  LineGraph2 : fbLineGraph1 := ( fit := 1, minY := 0.0, maxY := 10.0,
    maxPoints := 31, ratio := 3.0,
    axisColor := GCOLOR_WHITE,
    textSize := 60,
    graphColor := GCOLOR_LIGHT_BLUE_A700,
    graphStyle := GGRAPH_LINE_DEFAULT);

END_VAR

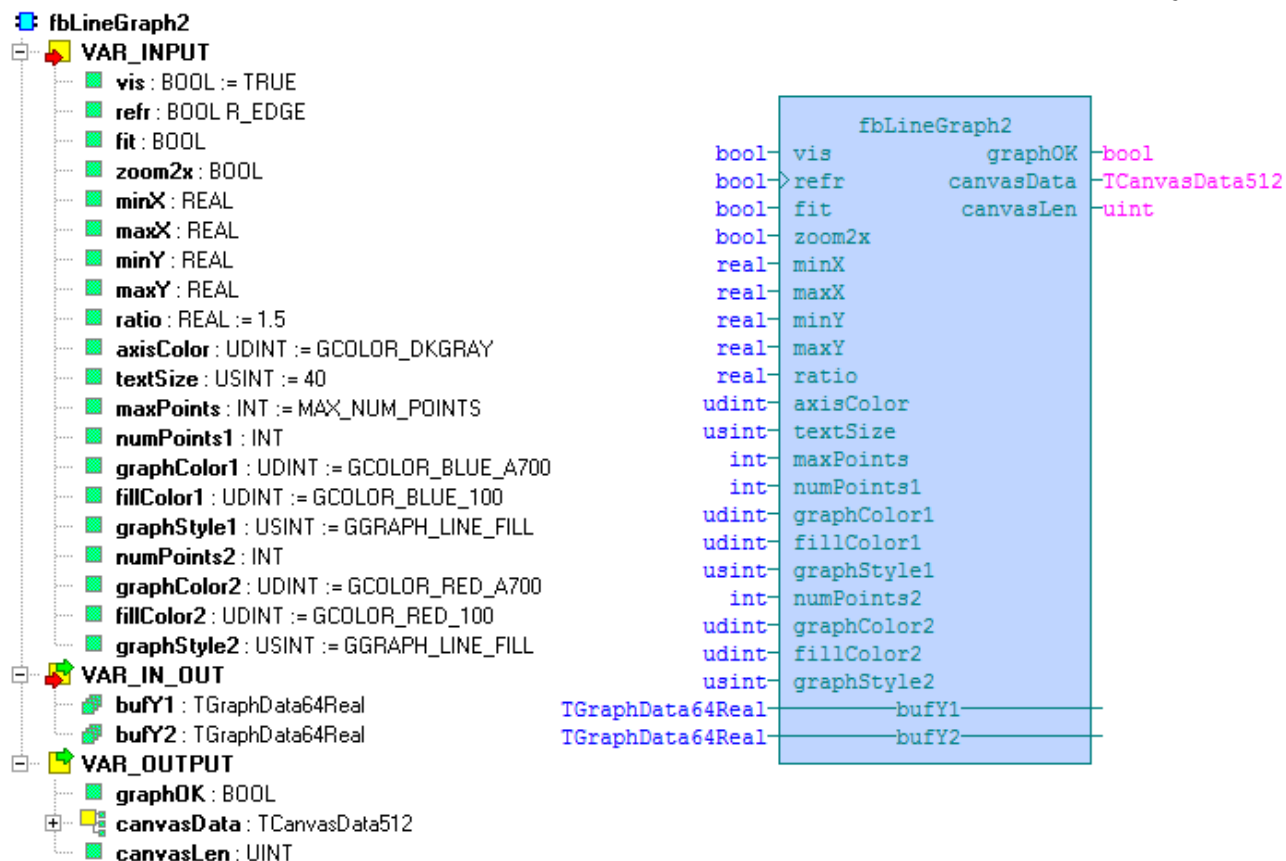
LineGraph2 ( numPoints := 25, bufY := void( temp_samples));

END_PROGRAM
```

Prvek „Kreslicí plátno“ má nastaven rozměr 600x200 (`ratio := 3.0` což odpovídá nastavenému rozměru) a jeho řídicí struktura je `Example2.LineGraph2.canvasData` a pole „Vracet pozici“ je zaškrtnuté (to je nutné pro pokládání měřící značky do grafu). Pozadí stránky je černé.



## 2.2 Čárový graf pro dva průběhy fbLineGraph2


Knihovna : *CanvasObjectsLib*

Funkční blok *fbLineGraph2* sestaví zásobník příkazů pro zobrazení čárového grafu se dvěma průběhy, z nichž každý může mít maximálně 64 hodnot (zlomových bodů). Hodnoty zobrazené v grafu jsou očekávány v proměnných *bufY1* a *bufY2*. Příkazy, které zařídí vykreslení grafu ve web stránce, jsou uloženy ve výstupu *canvasData*.

Popis proměnných :

	Proměnná	Typ	Význam
<b>VAR_INPUT</b>			
	<i>vis</i>	BOOL	Viditelnost grafu True znamená zobrazit graf (výchozí hodnota)
	<i>refr</i>	BOOL R_EDGE	Na náběžnou hranu této proměnné dojde k překreslení grafu
	<i>fit</i>	BOOL	Umožňuje změnit rozsah osy Y podle zobrazených dat Je-li FALSE, osa Y má rozsah daný vstupy <i>minY</i> , <i>maxY</i> Je-li TRUE, minimum a maximum na ose Y odpovídá minimální a maximální hodnotě zobrazené proměnné
	<i>zoom2x</i>	BOOL	Zvětšení grafu Je-li FALSE, osa X je zobrazena podle vstupů <i>minX</i> , <i>maxX</i> Je-li TRUE, osa Y změní rozsah jako při <i>fit</i> = TRUE a osa X zmenší rozsah 2x, čímž dojde ke zvětšení grafu (lupa)

	Proměnná	Typ	Význam
	<i>minX, maxX</i>	REAL	Meze udávající rozsah osy X při <i>minX = maxX = 0</i> je rozsah osy X v intervalu $<1, \text{maxPoints}>$
	<i>minY, maxY</i>	REAL	Meze udávající rozsah osy Y Platí pro <i>fit = 0</i> a <i>zoom2x = 0</i>
	<i>ratio</i>	REAL	Poměr šířky a výšky grafu ve web stránce – měl by odpovídat zvoleným velikostem prvku „Kreslící plátno“ Výchozí hodnota je 1.5 (např. pro 300x200 pixelů) Přípustná hodnota musí být v intervalu $<1.0, 5.0>$
	<i>axisColor</i>	UDINT	Barva, kterou jsou nakresleny osy a popisky Výchozí hodnota je GCOLOR_DKGRAY
	<i>textSize</i>	USINT	Velikost textu pro popisky Texty popisků se automaticky přizpůsobují zvoleným rozměrům grafu (prvku „Kreslící plátno“) - pokud automaticky nastavená velikost písma nevyhovuje, lze ji změnit zvětšením resp. zmenšením hodnoty v tomto parametru Výchozí hodnota je 40
	<i>maxPoints</i>	INT	Max. počet hodnot zobrazitelných v grafu Výchozí hodnota 64, maximální hodnota 64. Tento parametr definuje kolik hodnot lze v grafu zobrazit
	<i>numPoints1</i>	INT	Počet aktuálně zobrazených hodnot pro první graf Tento parametr definuje, z kolika hodnot je nakreslena čára grafu
	<i>graphColor1</i>	UDINT	Barva, kterou je nakreslena čára prvního grafu Výchozí hodnota GCOLOR_BLUE_A700
	<i>fillColor1</i>	UDINT	Barva výplně prvního grafu Platí pro <i>graphStyle = GGRAPH_LINE_FILL</i> Výchozí hodnota GCOLOR_BLUE_100
	<i>graphStyle1</i>	USINT	Styl prvního grafu Výchozí hodnota GGRAPH_LINE_FILL (graf s výplní)
	<i>numPoints2</i>	INT	Počet aktuálně zobrazených hodnot pro druhý graf Tento parametr definuje, z kolika hodnot je nakreslena čára grafu
	<i>graphColor2</i>	UDINT	Barva, kterou je nakreslena čára druhého grafu Výchozí hodnota GCOLOR_RED_A700
	<i>fillColor2</i>	UDINT	Barva výplně druhého grafu Platí pro <i>graphStyle = GGRAPH_LINE_FILL</i> Výchozí hodnota GCOLOR_RED_100
	<i>graphStyle2</i>	USINT	Styl druhého grafu Výchozí hodnota GGRAPH_LINE_FILL (graf s výplní)
<b>VAR_IN_OUT</b>			
	<i>bufY1</i>	TGraph-Data64-	Proměnná obsahující hodnoty, které budou zobrazeny v prvním grafu (maximálně 64 hodnot typu REAL)

	Proměnná	Typ	Význam
		Real	
	bufY2	TGraph-Data64-Real	Proměnná obsahující hodnoty, které budou zobrazeny v druhém grafu (maximálně 64 hodnot typu REAL)
<b>VAR_OUTPUT</b>			
	graphOK	BOOL	TRUE znamená, že se podařilo úspěšně sestavit všechny příkazy potřebné k vykreslení grafu
	canvasData	TCanvas-Data512	Buffer příkazů pro vykreslení grafu
	canvasLen	UINT	Aktuální délka příkazů v <i>CanvasData</i>

### Příklad 3

V tomto příkladu jsou zobrazeny průběhy pro funkce sin a cos v plovoucím grafu. Hodnoty pro zobrazení v grafu jsou uloženy v proměnných `dataSin` a `dataCos`. Počet zobrazených hodnot je 64, což je maximální hodnota pro `fbLineGraph2`. Barvy grafu jsou výchozí, interpolace mezi vzorky grafu není zapnutá (`GGRAPH_LINE_DEFAULT`), grafy jsou s výplní. Hodnoty argumentů funkcí, které budou zobrazeny na ose X se připravují do proměnných `minX` a `maxX`. Osy a popisky jsou vykresleny výchozí barvou `G_COLOR_DKGRAY`. Proměnná `run` slouží pro zapnutí resp. vypnutí posunu grafu.

```

PROGRAM prgExample3
VAR CONSTANT
  NUM_SAMPLES : INT := 64;
  ARG_STEP    : REAL := 0.1570796;
END_VAR
VAR
  // blok pro vykreslení grafu
  LineGraph3 : fbLineGraph2 := ( minY := -1.5, maxY := 1.5,
                                maxPoints := NUM_SAMPLES,
                                numPoints1 := NUM_SAMPLES,
                                numPoints2 := NUM_SAMPLES);

  run          : BOOL := 1;
  dataSin      : ARRAY[1..NUM_SAMPLES] OF REAL;
  dataCos      : ARRAY[1..NUM_SAMPLES] OF REAL;
  arg          : REAL := 0.0;
  minX, maxX  : REAL;
END_VAR
VAR_EXTERNAL
  AT %S20.1    : BOOL;
END_VAR

// vypocet hodnot pro graf
IF %S20.1 AND run THEN
  InsertNewValToGraph(save := TRUE, newVal := sin( arg),
    buf := void( dataSin), sizeBuf := sizeof( dataSin));
  InsertNewValToGraph(save := TRUE, newVal := cos( arg),
    buf := void( dataCos), sizeBuf := sizeof( dataCos));

  maxX := arg;
  minX := arg - INT_TO_REAL(NUM_SAMPLES-1)*ARG_STEP; // osa X

```

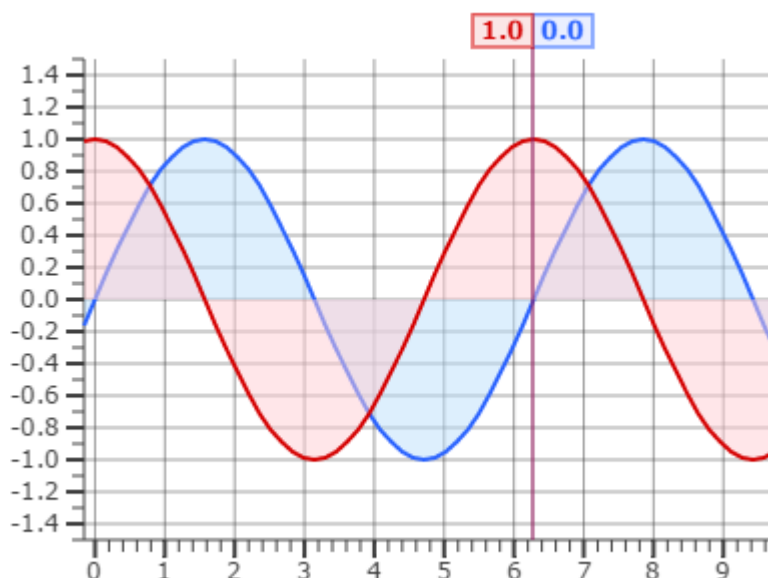
```

arg := arg + ARG_STEP; // argument pro pristi krok
IF arg > 6.*3.14159 THEN arg := arg - 4.*3.14159; END_IF;
END_IF;

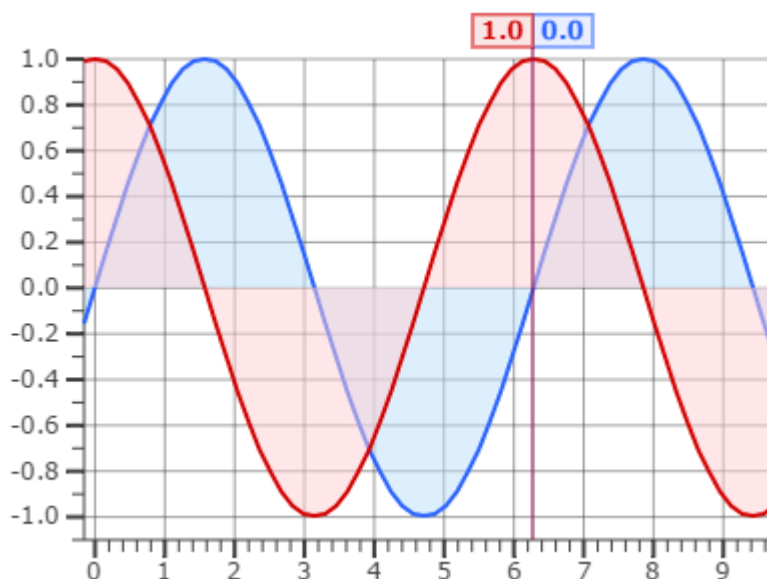
// vykreslit graf
LineGraph3( refr := %S20.1, minX := minX, maxX := maxX,
            bufY1 := void(dataSin), bufY2 := void(dataCos));
END_PROGRAM

```

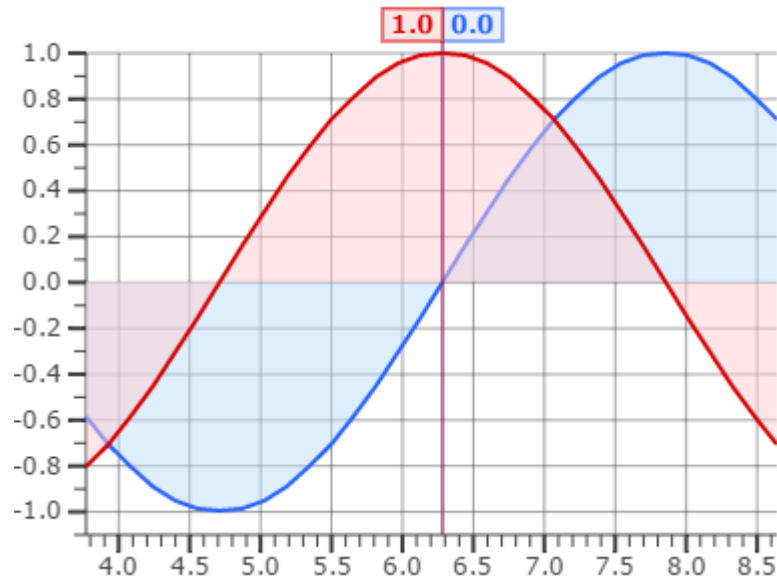
Prvek „Kreslicí plátno“ má nastaven rozměr 450x300 ( ratio := 1.5 což odpovídá výchozímu nastavení) a jeho řídicí struktura je `Example3.LineGraph3.canvasData` a pole „Vracet pozici“ je zaškrtnuté (to je nutné pro pokládání měřící značky do grafu).



Nastavením proměnné `Example3.LineGraph3.fit := 1` se automaticky přizpůsobí osa Y (podle minimální a maximální zobrazené hodnoty).



Nastavením proměnné `Example3.LineGraph3.zoom2x := 1` se automaticky přizpůsobí osa Y a osa X se 2x zvětší.



#### Příklad 4

Porovnání průměrných denních teplot v měsíci červnu a červenci. Naměřené teploty jsou uloženy v poli `tempJun.` a `tempJul.` Pro červenec program simuluje postupné zvětšování počtu měření – druhý průběh se dynamicky mění. Graf je zkonfigurován na zobrazení max. 31 hodnot. Barva prvního průběhu je `GCOLOR_ORANGE_700`, barva druhého průběhu je `GCOLOR_DEEP_PURPLE_A700`, iinterpolace mezi vzorky grafu není zapnutá (`GGRAPH_LINE_DEFAULT`), první průběh je s výplní `GCOLOR_DEEP_ORANGE_100`. Osa X má automatické číslování (parametry `minX` a `maxX` nejsou v deklaraci `TempGraph` vyplněny). Osa Y má rozsah od 15.0 do 23.0 (`minY := 15.0`, `maxY := 23.0`). Osy a popisky jsou vykresleny výchozí barvou a velikostí.

```
PROGRAM prgExample4
VAR CONSTANT
  NUM_SAMPLES_1 : INT := 30;
  NUM_SAMPLES_2 : INT := 31;
END_VAR
VAR
  tempJun : ARRAY[1..NUM_SAMPLES_1] OF REAL :=
  [ 16.9, 16.5, 17.2, 17.5, 17.6, 17.7, 18.1, 17.4, 17.5, 17.3, 17.3, 17.6,
    18.2, 18.4, 17.8, 17.3, 18.0, 18.4, 17.9, 18.3, 18.3, 19.4, 19.1, 19.3,
    19.5, 20.2, 20.0, 19.0, 18.3, 18.5 ];
  tempJul : ARRAY[1..NUM_SAMPLES_2] OF REAL :=
  [ 18.6, 19.0, 19.6, 18.9, 19.5, 19.6, 19.3, 19.4, 19.4, 19.1, 20.0, 20.3,
    20.3, 19.9, 19.6, 20.0, 19.5, 19.4, 19.0, 19.4, 19.1, 19.5, 20.4, 20.1,
    19.7, 19.5, 19.8, 20.0, 20.6, 20.7, 21.0 ];

  TempGraph : fbLineGraph2 := ( minY := 15.0, maxY := 23.0,
                                maxPoints := NUM_SAMPLES_2,
                                graphColor1 := GCOLOR_ORANGE_700,
                                fillColor1 := GCOLOR_DEEP_ORANGE_100,
                                graphColor2 := GCOLOR_DEEP_PURPLE_A700,
                                graphStyle1 := GGRAPH_LINE_FILL,
```



```

graphStyle2 := GGRAPH_LINE_DEFAULT);

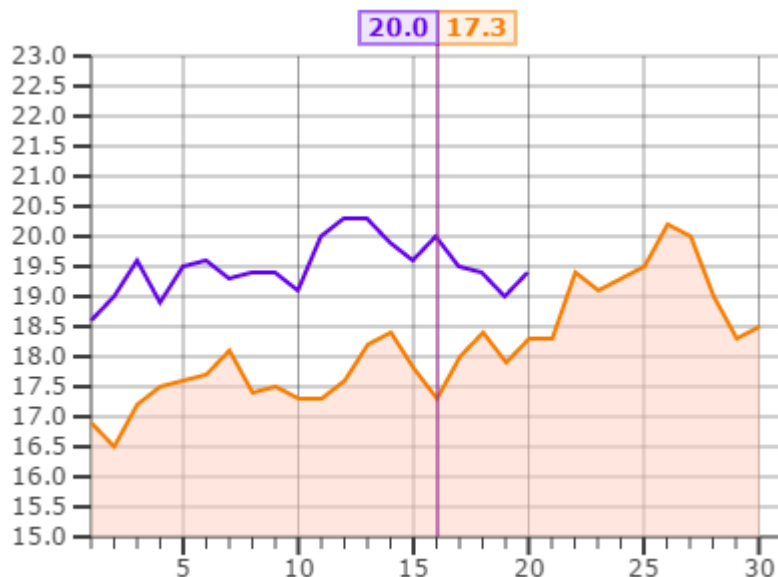
run      : BOOL := 1;
refresh  : BOOL;
step     : BOOL;           // jeden krok
index    : INT := 1;       // pocet dnu pro 2.prubeh
END_VAR
VAR_EXTERNAL
  AT %S20.1 : BOOL;       // casova zakladna
END_VAR

// priprava hodnot pro graf
IF (%S20.1 AND run) OR step THEN
  index := index + 1;      // index zobrazene hodnoty
  IF index > NUM_SAMPLES_2 THEN
    index := 1;
  END_IF;
  refresh := 1; step := 0;
END_IF;

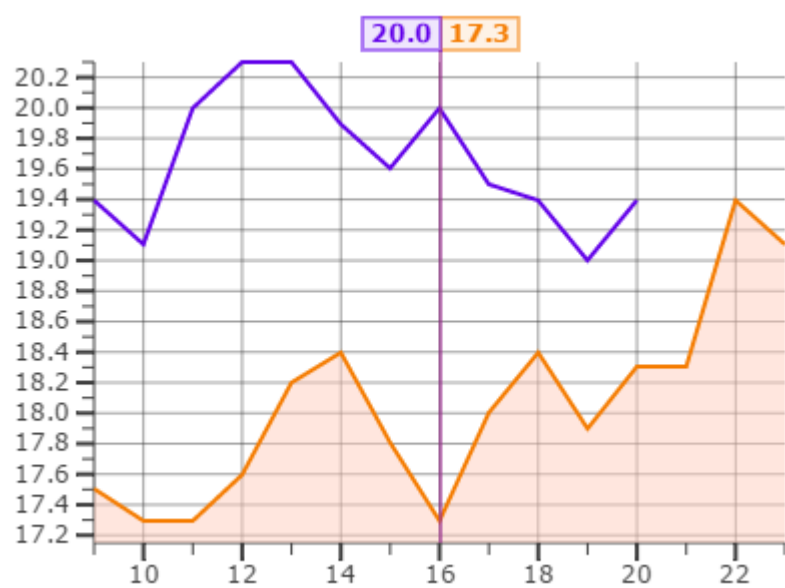
// spocitat hodnoty pro graf
TempGraph( refr      := refresh,
            numPoints1 := NUM_SAMPLES_1,
            numPoints2 := index,
            bufY1      := void(tempJun),
            bufY2      := void(tempJul));
refresh := FALSE;
END_PROGRAM

```

Prvek „Kreslicí plátno“ má nastaven rozměr 450x300 (ratio := 1.5) a jeho řídící struktura je `Example4.TempGraph.canvasData` a pole „Vracet pozici“ je zaškrtnuté (to je nutné pro pokládání měřicí značky do grafu).

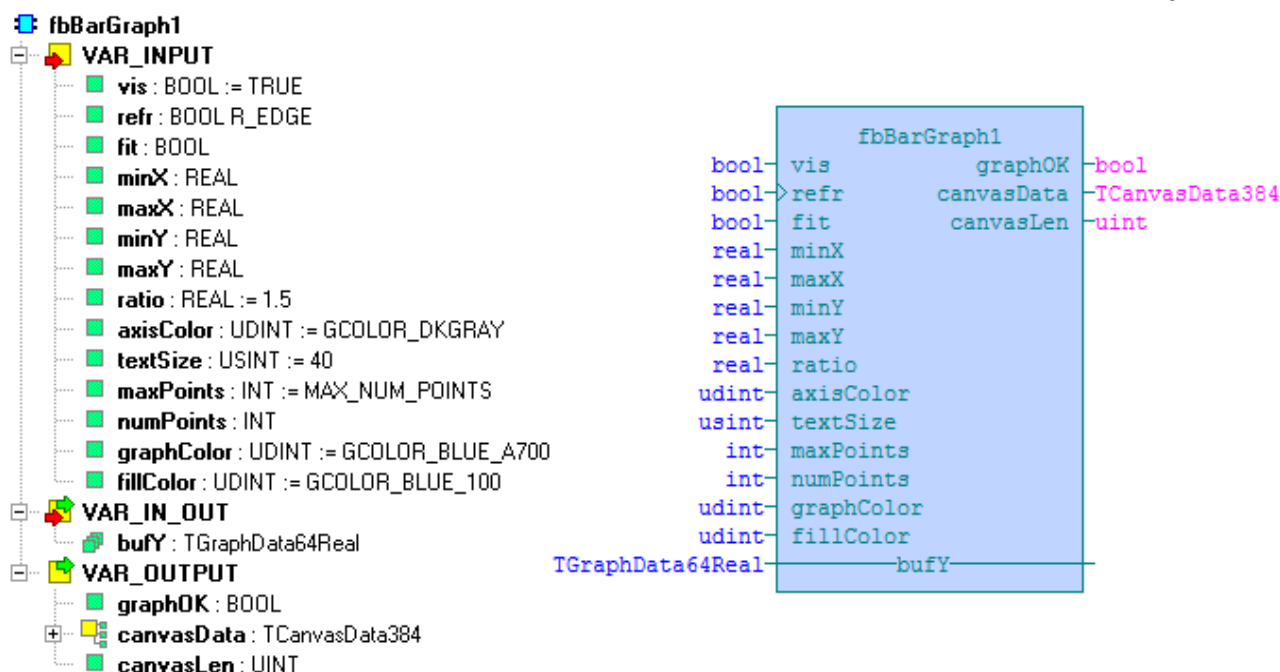


Zvětšený graf (`Example4.TempGraph.zoom2x := TRUE`) vypadá následovně:



## 2.3 Sloupcový graf pro jeden průběh fbBarGraph1








Knihovna : CanvasObjectsLib



Funkční blok *fbBarGraph1* sestaví zásobník příkazů pro zobrazení sloupcového grafu, ve kterém může být maximálně 64 hodnot (sloupců). Hodnoty zobrazené v grafu jsou očekávány v proměnné *bufY*. Příkazy, které zařídí vykreslení grafu ve web stránce, jsou uloženy ve výstupu *canvasData*.

Popis proměnných :

	Proměnná	Typ	Význam
<b>VAR_INPUT</b>			
	vis	BOOL	Viditelnost grafu True znamená zobrazit graf (výchozí hodnota)
	refr	BOOL R_EDGE	Na náběžnou hranu této proměnné dojde k překreslení grafu
	fit	BOOL	Umožňuje změnit rozsah osy Y podle zobrazených dat Je-li FALSE, osa Y má rozsah daný vstupy <i>minY</i> , <i>maxY</i> Je-li TRUE, minimum a maximum na ose Y odpovídá minimální a maximální hodnotě zobrazené proměnné
	minX, maxX	REAL	Meze udávající rozsah osy X při <i>minX</i> = <i>maxX</i> = 0 je rozsah osy X v intervalu <1, <i>maxPoints</i> >
	minY, maxY	REAL	Meze udávající rozsah osy Y Platí pro <i>fit</i> = 0 a <i>zoom2x</i> = 0
	ratio	REAL	Poměr šířky a výšky grafu ve web stránce – měl by odpovídat zvoleným velikostem prvku „Kreslící plátno“ Výchozí hodnota je 1.5 (např. pro 300x200 pixelů)

	Proměnná	Typ	Význam
			Přípustná hodnota musí být v intervalu <1.0, 5.0>
	<i>axisColor</i>	UDINT	Barva, kterou jsou nakresleny osy a popisky Výchozí hodnota je GCOLOR_DKGRAY
	<i>textSize</i>	USINT	Velikost textu pro popisky Texty popisků se automaticky přizpůsobují zvoleným rozměrům grafu (prvku „Kreslící plátno“) - pokud automaticky nastavená velikost písma nevyhovuje, lze ji změnit zvětšením resp. zmenšením hodnoty v tomto parametru Výchozí hodnota je 40
	<i>maxPoints</i>	INT	Max. počet sloupců zobrazitelných v grafu Výchozí hodnota 64, maximální hodnota 64. Tento parametr definuje kolik hodnot lze v grafu zobrazit
	<i>numPoints</i>	INT	Počet aktuálně zobrazených sloupců (hodnot)
	<i>graphColor</i>	UDINT	Barva, kterou je nakreslen obrys sloupců Výchozí hodnota GCOLOR_BLUE_A700
	<i>fillColor</i>	UDINT	Barva výplně sloupců Výchozí hodnota GCOLOR_BLUE_100
<b>VAR_IN_OUT</b>			
	<i>bufY</i>	TGraph-Data64-Real	Proměnná obsahující hodnoty, které budou zobrazeny v grafu (maximálně 64 hodnot typu REAL)
<b>VAR_OUTPUT</b>			
	<i>graphOK</i>	BOOL	TRUE znamená, že se podařilo úspěšně sestavit všechny příkazy potřebné k vykreslení grafu
	<i>canvasData</i>	TCanvas-Data384	Buffer příkazů pro vykreslení grafu
	<i>canvasLen</i>	UINT	Aktuální délka příkazů v <i>CanvasData</i>

### Příklad 5

Zobrazení denních teplot sloupcovým grafem. Naměřené teploty jsou uloženy v poli *temp*. Barva sloupců a výplně je výchozí. Osa X má automatické číslování (parametry *minX* a *maxX* nejsou v deklaraci *BarGraph1* vyplněny). Osy a popisky jsou vykresleny výchozí barvou a velikostí (GCOLOR\_DKGRAY).

```

PROGRAM prgExample5
  VAR CONSTANT
    GRAPH_TEMP_SIZE : INT := 24;    // max pocet bodu grafu
  END_VAR
  VAR
    temp : ARRAY[1..GRAPH_TEMP_SIZE] OF REAL :=
      [ -10.3, -9.3, -7.8, -9.6, -10.8, -11.9, -5.6, -3.6, -1.7, 1.5, 2.8, 2.1,
        1.3, 1.1, -0.2, -1.6, -3.3, -3.3, -4.0, -4.3, -5.0, -7.4, -7.4, -7.1 ];

```

```

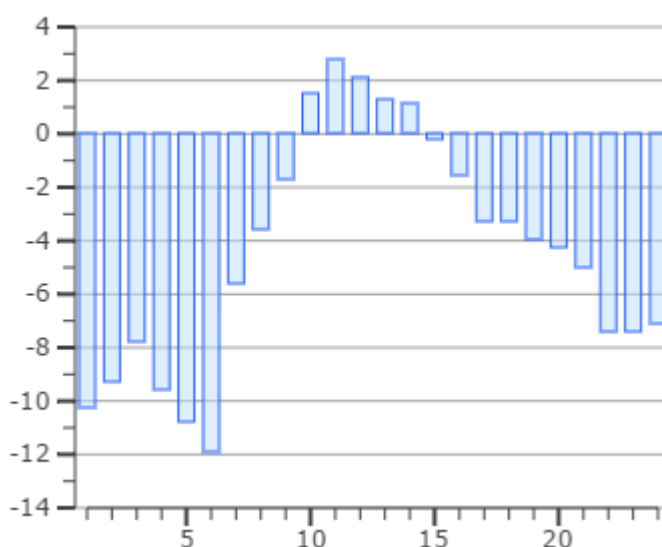
BarGraph1 : fbBarGraph1 := ( minY := -14.0, maxY := 4.0,
                             maxPoints := GRAPH_TEMP_SIZE,
                             numPoints := GRAPH_TEMP_SIZE,
                             ratio := 1.3);

END_VAR

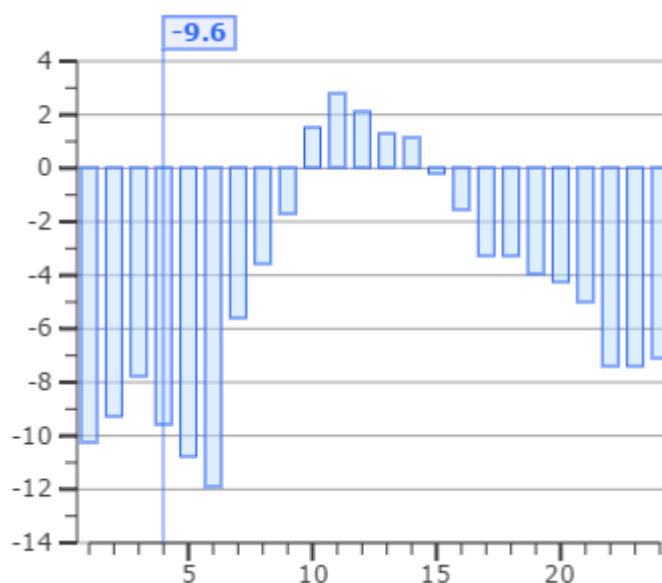
// graf v implicitním nastavení
BarGraph1 ( bufY := void( temp));
END_PROGRAM

```

Prvek „Kreslicí plátno“ má nastaven rozměr 400x300 ( $\text{ratio} := 1.3$ ) a jeho řídící struktura je `Example5.BarGraph1.canvasData` a pole „Vracet pozici“ je zaškrtnuté (to je nutné pro pokládání měřicí značky do grafu).

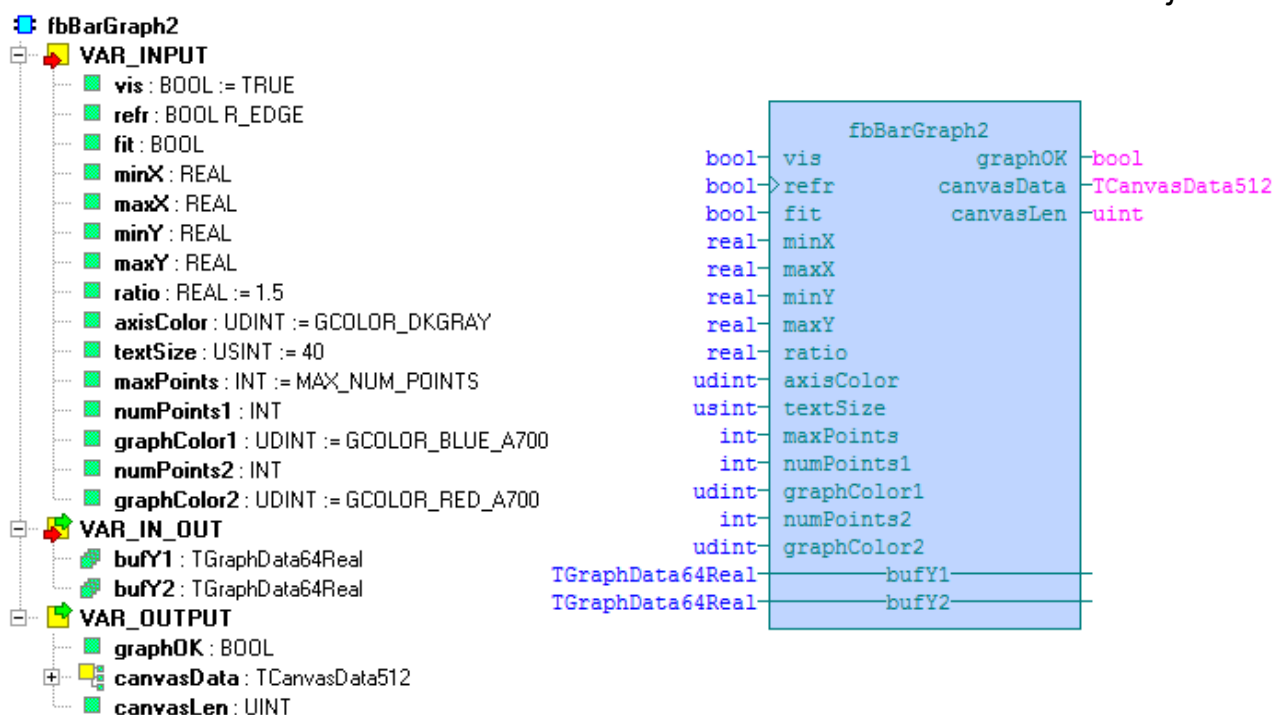


S položenou měřicí značkou vypadá graf následovně:



## 2.4 Sloupcový graf pro dva průběhy fbBarGraph2










Knihovna : CanvasObjectsLib



Funkční blok *fbBarGraph2* sestaví zásobník příkazů pro zobrazení sloupcového grafu se dvěma průběhy, z nichž každý může mít maximálně 64 hodnot (sloupců). Hodnoty zobrazené v grafu jsou očekávány v proměnných *bufY1* a *bufY2*. Příkazy, které zařídí vykreslení grafu ve web stránce, jsou uloženy ve výstupu *canvasData*.

Popis proměnných :

	Proměnná	Typ	Význam
<b>VAR_INPUT</b>			
	<i>vis</i>	BOOL	Viditelnost grafu True znamená zobrazit graf (výchozí hodnota)
	<i>refr</i>	BOOL R_EDGE	Na náběžnou hranu této proměnné dojde k překreslení grafu
	<i>fit</i>	BOOL	Umožňuje změnit rozsah osy Y podle zobrazených dat Je-li FALSE, osa Y má rozsah daný vstupy <i>minY</i> , <i>maxY</i> Je-li TRUE, minimum a maximum na ose Y odpovídá minimální a maximální hodnotě zobrazené proměnné
	<i>minX</i> , <i>maxX</i>	REAL	Meze udávající rozsah osy X při <i>minX</i> = <i>maxX</i> = 0 je rozsah osy X v intervalu <1, <i>maxPoints</i> >
	<i>minY</i> , <i>maxY</i>	REAL	Meze udávající rozsah osy Y Platí pro <i>fit</i> = 0 a <i>zoom2x</i> = 0
	<i>ratio</i>	REAL	Poměr šířky a výšky grafu ve web stránce – měl by odpovídat zvoleným velikostem prvku „Kreslící plátno“

	Proměnná	Typ	Význam
			Výchozí hodnota je 1.5 (např. pro 300x200 pixelů) Přípustná hodnota musí být v intervalu <1.0, 5.0>
	<i>axisColor</i>	UDINT	Barva, kterou jsou nakresleny osy a popisky Výchozí hodnota je GCOLOR_DKGRAY
	<i>textSize</i>	USINT	Velikost textu pro popisky Texty popisků se automaticky přizpůsobují zvoleným rozměrům grafu (prvku „Kreslicí plátno“) - pokud automaticky nastavená velikost písma nevyhovuje, lze ji změnit zvětšením resp. zmenšením hodnoty v tomto parametru Výchozí hodnota je 40
	<i>maxPoints</i>	INT	Max. počet sloupců zobrazitelných v grafu Výchozí hodnota 64, maximální hodnota 64. Tento parametr definuje kolik hodnot lze v grafu zobrazit
	<i>numPoints1</i>	INT	Počet aktuálně zobrazených sloupců (hodnot) prvního průběhu
	<i>graphColor1</i>	UDINT	Barva, kterou jsou nakresleny sloupce pro první průběh Výchozí hodnota GCOLOR_BLUE_A700
	<i>numPoints2</i>	INT	Počet aktuálně zobrazených sloupců (hodnot) druhého průběhu
	<i>graphColor2</i>	UDINT	Barva, kterou jsou nakresleny sloupce pro druhý průběh Výchozí hodnota GCOLOR_RED_A700
<b>VAR_IN_OUT</b>			
	<i>bufY1</i>	TGraph-Data64-Real	Proměnná obsahující hodnoty, které budou zobrazeny v prvním průběhu (maximálně 64 hodnot typu REAL)
	<i>bufY2</i>	TGraph-Data64-Real	Proměnná obsahující hodnoty, které budou zobrazeny v druhém průběhu (maximálně 64 hodnot typu REAL)
<b>VAR_OUTPUT</b>			
	<i>graphOK</i>	BOOL	TRUE znamená, že se podařilo úspěšně sestavit všechny příkazy potřebné k vykreslení grafu
	<i>canvasData</i>	TCanvas-Data512	Buffer příkazů pro vykreslení grafu
	<i>canvasLen</i>	UINT	Aktuální délka příkazů v <i>CanvasData</i>

## Příklad 6

Porovnání roční spotřeby za poslední 2 roky, jeden sloupec grafu reprezentuje spotřebu za jeden měsíc. Naměřené spotřeby jsou uloženy v poli `cons_2014` a `cons_2015`. Graf je zkonfigurován na zobrazení 2x12 sloupců. Nastavení grafu je výchozí.

```
PROGRAM prgExample6
VAR
  cons_2014 : ARRAY[1..12] OF REAL :=
    [ 1586.2, 1463.1, 998.7, 576.3, 465.2, 401.6,
      355.1, 341.4, 564.4, 784.3, 875.3, 1466.6];

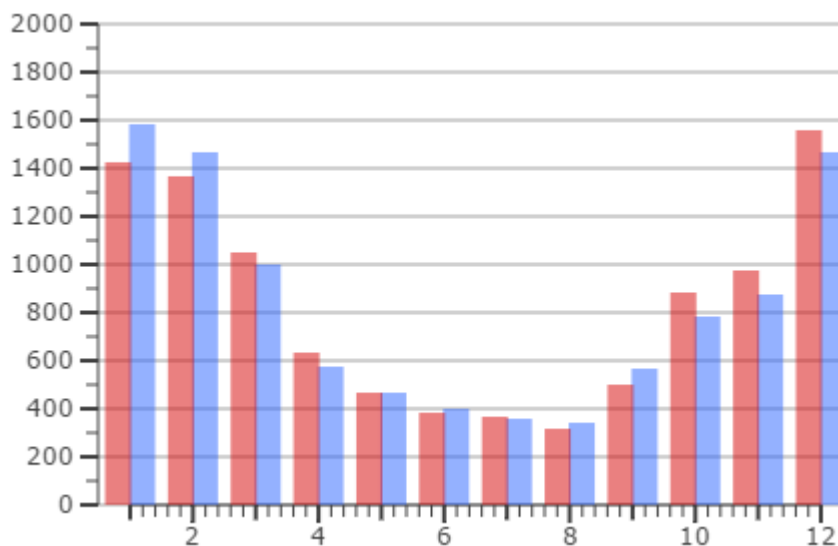
  cons_2015 : ARRAY[1..12] OF REAL :=
    [ 1422.2, 1363.1, 1050.7, 630.7, 465.4, 380.3,
      366.6, 311.7, 500.8, 884.1, 975.5, 1555.6];

  BarGraph2 : fbBarGraph2 := (minY := 0.0, maxY := 2000.0,
                               maxPoints := 12,
                               numPoints1 := 12, numPoints2 := 12);

END_VAR

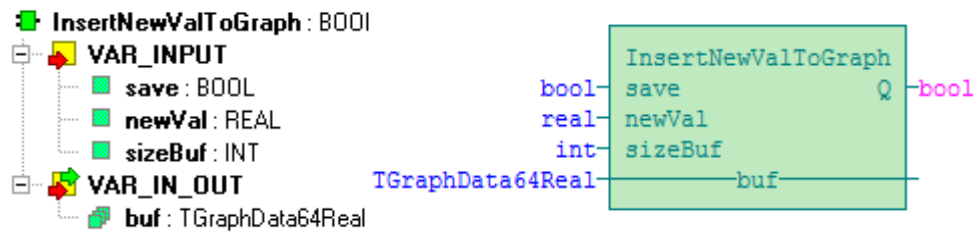
BarGraph2 ( bufY1 := void( cons_2014), bufY2 := void( cons_2015));
END_PROGRAM
```

Prvek „Kreslicí plátno“ má nastaven rozměr 450x300 (`ratio := 1.5`) a jeho řídící struktura je `Example6.BarGraph2.canvasData` a pole „Vracet pozici“ je zaškrtnuté (to je nutné pro pokládání měřicí značky do grafu).





## 2.5 Funkce InsertNewValToGraph

Knihovna : *CanvasObjectsLib*

Funkce *InsertNewValToGraph* vloží novou hodnotu na konec pole hodnot pro graf. Současně posune hodnoty v poli tak, že historicky nejstarší hodnota vypadne z pole. Vizuálně se tedy graf posune o jednu hodnotu vlevo.

Popis proměnných :

	Proměnná	Typ	Význam
<b>VAR_INPUT</b>			
	save	BOOL	Je-li TRUE, vložit novou hodnotu Je-li FALSE, nedělat nic
	newVal	REAL	Vkládaná hodnota
	sizeBuf	INT	Velikost pole hodnot (počet bytů) Typicky sizeof(TGraphData64Real)
<b>VAR_IN_OUT</b>			
	buf	TGraphData64Real	Pole hodnot pro graf
<b>FlashRead</b>			
	Návratová hodnota	BOOL	Vrací TRUE pokud je třeba překreslit graf, jinak FALSE

### Příklad 7

V následujícím příkladu je funkce *InsertNewValToGraph* použita pro plnění pole hodnot pro trapezový průběh (*dataTrapeze*). Data jsou pak zobrazena čárovým i sloupcovým grafem.

```

PROGRAM prgExample7
  VAR CONSTANT
    NUM_SAMPLES : INT := 48;
  END_VAR
  VAR
    // blok pro vykreslení grafu
    LineGraphTrapeze : fbLineGraph1 := ( minY := 0.0, maxY := 1.0,
      maxPoints := NUM_SAMPLES,
      numPoints := NUM_SAMPLES,
      graphColor := GCOLOR_GREEN_A700,
      fillColor := GCOLOR_LIME_100,
      graphStyle := GGRAPH_LINE_FILL);
  
```

```

BarGraphTrapeze : fbBarGraph1 := ( minY := 0.0, maxY := 1.0,
                                     maxPoints := NUM_SAMPLES,
                                     numPoints := NUM_SAMPLES,
                                     graphColor := GCOLOR_DEEP_ORANGE_A400,
                                     fillColor := GCOLOR_DEEP_ORANGE_100);

run          : BOOL := 1;
rfr          : BOOL;
step         : BOOL;
arg_step     : REAL := 0.05;
arg          : REAL;
dataTrapeze  : ARRAY[1..NUM_SAMPLES] OF REAL;      // trapez (osa Y)
END_VAR
VAR_EXTERNAL
  AT %S20.1   : BOOL;                               // casova zakladna
  AT %S21.1   : BOOL;                               // casova zakladna
END_VAR

// vypocet hodnot pro graf
IF ((%S20.1 OR %S21.1) AND run) OR step THEN
  step := 0;
  arg := arg + ARG_STEP;                             // argument pro tento krok
  IF (arg < 0.0) OR (arg > 1.0) THEN arg_step := arg_step * -1.0; END_IF;
  rfr := InsertNewValToGraph( save := 1,
                             newVal := LIMIT(MN := 0.2, IN := arg, MX := 0.8),
                             sizeBuf := sizeof(dataTrapeze),
                             buf := void(dataTrapeze));

END_IF;

// vykreslit carovy graf
LineGraphTrapeze( refr := rfr, bufY := void(dataTrapeze));

// vykreslit sloupcovy graf
BarGraphTrapeze ( refr := rfr, bufY := void(dataTrapeze));
rfr := 0;
END_PROGRAM

```

