

# Software pro grafické terminály TERM10 a TERM03

## STAVEBNICE OBJEKTŮ

Příručka uživatele a programátora



**SofCon<sup>®</sup> spol. s r.o.**  
Střešovická 49  
162 00 Praha 6  
tel/fax: +420 220 180 454  
E-mail: [sofcon@sofcon.cz](mailto:sofcon@sofcon.cz)  
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 16.05.2003

Datum posledního uložení dokumentu: 16.05.2003

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

**Obsah :**

---

1.O dokumentu	4
1.1. Revize dokumentu	4
1.2. Účel dokumentu	4
1.3. Rozsah platnosti	4
1.4. Související dokumenty	4
2.Termíny a definice	4
3.Úvod	5
4.Varianty programového vybavení pro terminály	6
4.1. Lokální varianta programového vybavení	6
4.2. Vzdálená varianta programového vybavení	6
4.3. Varianta programového vybavení pro podřízenou periférii	6
5.Hierarchie objektových typů	7
5.1. Charakteristika jednotlivých objektových typů	8
6.Programové jednotky stavebnice	10
7.Hierarchie instancí objektů	12
7.1. Lokální varianta programového vybavení	12
7.2. Vzdálená varianta programového vybavení	13
7.3. Varianta programového vybavení pro podřízenou periférii	14
8.Procesy	15
8.1. Lokální varianta programového vybavení	15
8.2. Vzdálená varianta programového vybavení	15
8.3. Varianta podřízené periferie	15
9.Seznam dokumentů ke Stavebnici objektů	15
10. Seznam uživatelsky významných proměnných, procedur, funkcí, metod atd.	17

## 1. O dokumentu

---

### 1.1. Revize dokumentu

---

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00		Če		První vydání
1.10		Tu	16.05.2003	Úprava dokumentu dle ISO9000

### 1.2. Účel dokumentu

---

Tento dokument popisuje hierarchii objektů pro vytváření grafického rozhraní aplikací s terminálem Term03 a Term10.

### 1.3. Rozsah platnosti

---

Určen pro programátory a uživatele programového vybavení SofCon.

### 1.4. Související dokumenty

---

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál, ale je potřeba orientovat se v používání programového vybavení SofCon.

## 2. Termíny a definice

---

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

### 3. Úvod

---

Terminály **TERM10** a **TERM03** jsou programovatelné obecně použitelné průmyslové terminály s grafickým LCD displejem a membránovou klávesnicí. Oba terminály jsou řízeny **procesorem V40 programově kompatibilním s INTEL86**. To umožňuje využít k jejich programování vyšší programovací jazyky původně určené pro počítače IBM-PC včetně všech ladících možností, které tyto jazyky poskytují.

Pro programové vybavení terminálů je využíván programovací jazyk Turbo Pascal firmy Borland. Vzhledem k tomu, že terminály nevyužívají operační systém DOS, byly **upraveny systémové knihovny** jazyka Turbo Pascal tak, aby byly všude využívány přímo služby BIOSu terminálu.

Programové vybavení terminálů je založeno na využití **operačního systému ReTOS**, který umožňuje vytvářet a spouštět více procesů paralelně běžících v reálném čase a celou úlohu tak rozdělit do několika relativně samostatných částí. Operační systém ReTOS je implementován jako samostatná jednotka v jazyce Turbo Pascal (**Kernel**). Je inicializován a ukončován voláním příslušných procedur uvnitř programu v Turbo Pascalu. Jednotlivé paralelně běžící procesy se pak implementují jako procedury v jazyce Turbo Pascal. Popis o.s. ReTOS je obsahem samostatného dokumentu.

Programové vybavení pro terminály představuje několik jednotek obsahujících objektovou stavebnici určenou pro tvorbu komfortního grafického uživatelského rozhraní. K dispozici jsou jak objekty pro konkrétní fyzický hardware terminálů, tak pro jeho simulaci na počítači PC. Programové vybavení tak lze vyvíjet a ladit na počítači PC, přičemž displej terminálu je simulován v okénku na obrazovce monitoru a klávesnice terminálu je simulována klávesnicí počítače PC. Při přechodu k programu pro terminál pak stačí zaměnit příslušné objekty (např. podmíněným překladem) a použít systémové knihovny pro terminály.

Grafické uživatelské rozhraní je tvořeno hierarchickým systémem tzv. **menu-obrazovek**. Vzhled menu-obrazovky se skládá ze tří obrazových rovin složených operací OR: formátovaného textového výpisu s možností volby fontu a umístění jednotlivých částí výpisu, objektů vektorové grafiky (bod, úsečka, kružnice ap.) a grafického pozadí v podobě bitmapy. Systém menu je řízen objektem menu. Každá menu-obrazovka je implementována jako procedura v jazyce Turbo Pascal - tzv. **definiční procedura**. V definiční proceduře uživatel přiřadí do příslušných proměnných formální popisy jejího vzhledu v textové podobě a voláním metod objektu menu definuje její chování. Vstup dat z klávesnice a výstup na displej terminálu zajišťují objekty klávesnice a displeje, jejichž funkce je zastřešena objektem terminálu.

## 4. Varianty programového vybavení pro terminály

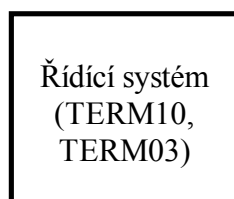
---

Objektová stavebnice umožňuje tvorbu programového vybavení ve třech základních variantách. Každou z těchto variant lze navíc simulovat na počítači PC. Použití příslušné varianty se řídí výběrem příslušných objektových typů stavebnice.

### 4.1. Lokální varianta programového vybavení

---

V lokální variantě se předpokládá použití terminálu jako samostatného řídicího systému s lokálním ovládáním a vizualizací. Jeden procesor terminálu provádí vlastní řídicí algoritmus a zároveň obsluhuje systém menu, zajišťuje příjem znaků z klávesnice a zobrazení na displeji.

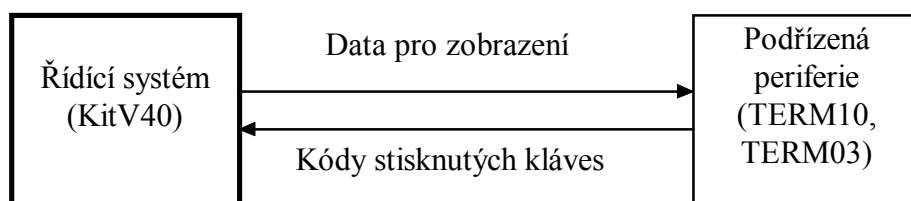


obr. A - lokální varianta

### 4.2. Vzdálená varianta programového vybavení

---

Programové vybavení ve vzdálené variantě není určeno pro hardware terminálu, ale pro průmyslovou řídicí jednotku KitV40. K této řídicí jednotce je sériovou komunikační linkou připojen terminál, který pracuje ve funkci podřízené periferie (viz níže). Řídicí jednotka KitV40 provádí řídicí algoritmus a obsluhuje systém menu. Data pro zobrazení vysílá do terminálu prostřednictvím komunikačního kanálu. Z komunikačního kanálu jsou přijímány kódy kláves stisknutých na terminálu.

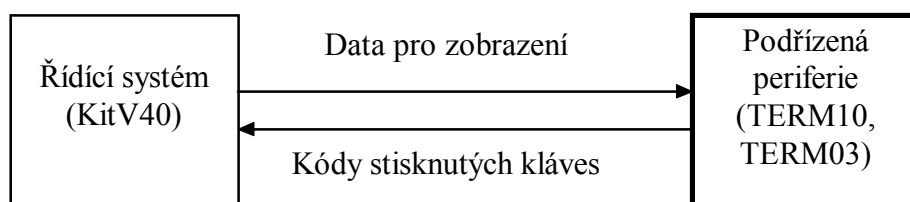


obr. B - vzdálená varianta

### 4.3. Varianta programového vybavení pro podřízenou periferii

---

Tato varianta je protějškem varianty předchozí. Programové vybavení je určeno pro terminál pracující ve funkci podřízené periferie. Data přijatá po komunikační lince zobrazuje na fyzický displej a vysílá kódy stisknutých kláves. Tato varianta programového vybavení je k dispozici jako firmware v podobě binárního obsahu paměti EPROM. Spolu s hardwarem terminálu tak tvoří samostatnou aplikaci - terminál jako podřízenou periferii s definovaným komunikačním rozhraním.

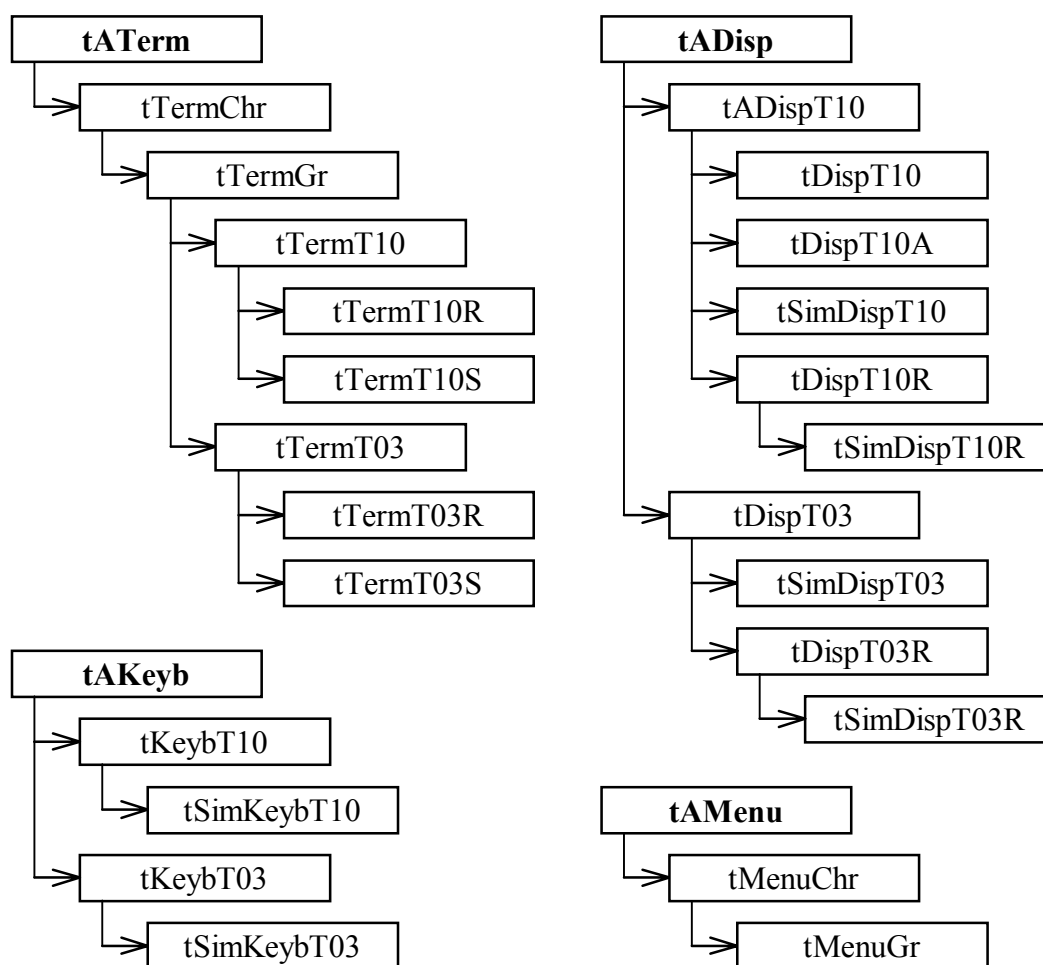


obr. C - varianta podřízené periferie

## 5. Hierarchie objektových typů

Z hlediska dědičnosti objektových typů je pro implementaci funkcí terminálu k dispozici následující hierarchická struktura, jejíž kořeny tvoří tyto abstraktní datové typy:

- tAMenu - obsluha obrazovek s menu
- tATerm - objekt, který spojuje displej a klávesnici do funkce terminálu
- tADisp - objekt implementující obecné vlastnosti libovolného displeje
- tAKeyb - objekt implementující obecné vlastnosti libovolné klávesnice
- tChnVirt - objekt implementující obecné vlastnosti komunikačního kanálu



obr. D - Hierarchie objektových typů

## 5.1. Charakteristika jednotlivých objektových typů

---

- ***tAMenu***  
Implementuje základní funkce ovládání pomocí menu-obrazovek.
- ***tMenuChr***  
Rozšiřuje objekt tAMenu pro práci se znakovým terminálem.
- ***tMenuGr***  
Rozšiřuje objekt tMenuChr pro práci s grafickým terminálem. Zavádí definici textových stylů (umístění a font), definici grafických objektů (kružnice, obdélník, přímka ap.) a definici pozadí obrazovky (BMP).
- ***tATerm***  
Typ tATerm je kořenem hierarchie objektových typů terminálů. Definiuje odkaz na instanci objektového typu tADisp a odkaz na instanci objektového typu tAKeyb. Typ tATerm slouží především jako virtuální rozhraní pro ovládání terminálu.
- ***tTermChr***  
Typ tTermChr implementuje v abstraktním terminálu práci s maticí znaků. Maticе reprezentuje řádky a sloupce znaků pro zobrazení na displej. Je implementována práce se znakovým kurzorem a editace ve vybrané části (podokénku) znakové matice. Implementována je podpora pro zobrazení obrazovek s textem nápovědy.
- ***tTermGr***  
Typ tTermGr rozšiřuje znakový terminál o možnost zobrazení textu v různých fontech a o definici grafických objektů. Editační znaková matice zůstává, avšak jsou-li definovány textové styly, je zobrazení řízeno těmito styly a nikoliv uspořádáním ve znakové matici. Nově implementovanými funkcemi jsou funkce pro dekódování příkazů pro zadávání stylů, grafických objektů a pozadí obrazovky. Tyto příkazy se zadávají v textové podobě pomocí ESC sekvencí.
- ***tTermT10***  
Typ tTermT10 rozšiřuje obecný grafický terminál o funkce vázané na konkrétní hardware terminálu TERM10.
- ***tTermT10R***  
Typ tTermT10R je modifikací typu tTermT10 určenou pro ovládání podřízeného terminálu TERM10 prostřednictvím komunikačního kanálu. Nově jsou implementovány komunikační funkce a základní obsluha terminálu.
- ***tTermT10S***  
Typ tTermT10S je implementací terminálu TERM10 jako podřízené periferie. Nově jsou implementovány především funkce příjmu dat popisujících obsah grafické obrazovky z komunikačního kanálu.  
  
Implementace je orientována na zobrazení celých obrazovek, které jsou popsány definovanou sadou dat. Tato sada obsahuje text určený k zobrazení, definici stylů pro zobrazovaný text, definici grafických objektů a definici pozadí. Obrazovka není zobrazena, dokud není přijata úplná sada dat popisujících její obsah.



- ***tTermT03***

Typ *tTermT03* rozšiřuje obecný grafický terminál o funkce vázané na konkrétní hardware terminálu TERM03.
- ***tTermT03R***

Typ *tTermT03R* je obdobou typu *tTermT10R* pro terminál TERM03.
- ***tTermT03S***

Typ *tTermT03S* je obdobou typu *tTermT10S* pro terminál TERM03.
- ***tADisp***

Abstraktní displej je implementován jako displej  $X \times Y$  znaků. Displej může zobrazovat pozici znakového kursoru v různých módech.
- ***tADispT10***

Typ *tADispT10* implementuje abstraktní grafický displej terminálu TERM10. Implementace je založena na použití stránek videopaměti. Výsledná stránka pro zobrazení se před odesláním na displej skládá ze tří dílčích stránek: stránky pro vykreslení pozadí, stránky pro vykreslení grafických objektů a stránky pro výpis textu ve fontech podle definovaných stylů. Vykreslení grafického obsahu každé stránky je vázáno na změnu dat, které obsah stránky popisují. Implementace je podřízena rozměrům rastru displeje  $240 \times 128$  pixelů. Komunikace s konkrétním hardware je implementována v dědicích tohoto abstraktního typu.
- ***tDispT10***

Typ *tDispT10* implementuje metody pro práci s konkrétním hardware displeje TERM10 Seiko (starší varianta terminálu TERM10).
- ***tDispT10A***

Typ *tDispT10A* implementuje metody pro práci s konkrétním hardware displeje TERM10 Toshiba (novější varianta terminálu TERM10).
- ***tDispT10R***

Typ *tDispT10R* používá stejný textový popis zobrazení na displeji jako *tDispT10*, avšak data nezobrazuje pomocí stránek videopaměti, ale odvysílá pomocí funkce svého terminálu (vlastníka instance *tDispT10R*) na podřízenou periférii.
- ***tSimDispT10R***

Funkce typu *tSimDispT10R* je stejná jako u *tDispT10R*, je však rozšířena o možnost kontrolního zobrazení v okénku na monitoru PC.
- ***tSimDispT10***

Typ *tSimDispT10* je určen pro ladění na počítači PC. Pro zobrazení je místo hardware terminálu TERM10 použito okénko na obrazovce monitoru PC.
- ***tDispT03***

Typ *tDispT03* implementuje ovládání grafického displeje TERM03. Funkce je obdobná jako u typu *tDispT10*, odlišný je však rozměr rastru displeje  $128 \times 64$  pixelů a některé funkce vázané na konkrétní hardware.
- ***tDispT03R***

Typ *tDispT03R* je obdobou typu *tDispT10R* pro vzdálenou variantu terminálu TERM03.

- ***tSimDispT03R***

Typ *tSimDispT03R* je obdobou typu *tSimDispT10R* pro vzdálenou variantu terminálu TERM03.

- ***tSimDispT03***

Typ *tSimDispT03* je obdobou typu *tSimDispT10* pro terminál TERM03.

- ***tAKeyb***

Typ *tAKeyb* je implementací abstraktní klávesnice. Implementovány jsou základní funkce pro práci s kruhovou vyrovnávací pamětí klávesnice.

- ***tKeybT10***

Typ *tKeybT10* implementuje ovládání klávesnice TERM10.

- ***tSimKeybT10***

Typ *tSimKeybT10* se používá k simulaci klávesnice terminálu TERM10 klávesnicí PC.

- ***tKeybT03***

Typ *tKeybT03* implementuje ovládání klávesnice TERM03.

- ***tSimKeybT03***

Typ *tSimKeybT03* se používá k simulaci klávesnice terminálu TERM03 klávesnicí PC.

- ***tChnVirt***

Typ *tChnVirt* se používá pro tvorbu konkrétních komunikačních jednotek.

- ***tAddChnVirt***

Typ *tAddChnVirt* zajišťuje dané komunikační knihovně, aby byla k aplikaci připojena a byla v ní k dispozici pro případné využití.

## 6. Programové jednotky stavebnice

---

Do stavebnice přísluší následující programové jednotky:

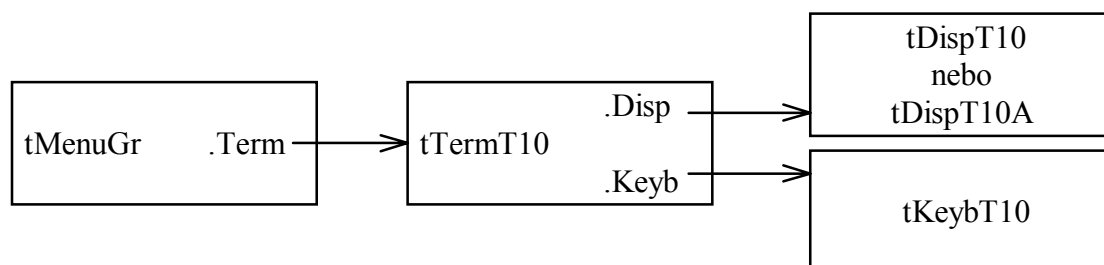
- ***uAMenu*** - Obsahuje základní deklarace typů pro práci s menu, objekt *tAMenu*.
- ***uMenuChr*** - Obsahuje objekt *tMenuChr*, proceduru *InitRunMenu*, která spouští procesy "MENU" a "TERM".
- ***uMenuGr*** - Obsahuje objekt *tMenuGr*.
- ***uATerm*** - Obsahuje základní deklarace typů pro práci s terminálem, konstanty řídicích znaků (*zESC*, *zCR*, *zLe*, *zRi* atd.), objekty *tADisp*, *tAKeyb* a *tATerm*.
- ***uTermChr*** - Obsahuje objekt *tTermChr*.
- ***uTermGr*** - Obsahuje základní deklarace typů pro práci s grafickým terminálem, objekt *tTermGr* a několik dalších objektů sloužících pro vnitřní potřebu objektů grafických terminálů a displejů.
- ***uTermT10*** - Obsahuje objekt *tTermT10*.

- **uTerm10R** - Obsahuje objekt tTermT10R.
- **uTerm10S** - Obsahuje objekt tTermT10S.
- **uTermT03** - Obsahuje objekt tTermT03.
- **uTerm03R** - Obsahuje objekt tTermT03R.
- **uTerm03S** - Obsahuje objekt tTermT03S.
- **uDispT10** - Obsahuje objekt abstraktního displeje TERM10 tADispT10, objekty tDispT10 a tDispT10A pro displeje Seiko a Toshiba a několik dalších objektů sloužících pro vnitřní potřebu tDispT10.
- **uDspT10R** - Obsahuje objekt tDispT10R.
- **uDispT03** - Obsahuje objekt tDispT03 a několik dalších objektů sloužících pro vnitřní potřebu tDispT03.
- **uDspT03R** - Obsahuje objekt tDispT03R.
- **uKeybT10** - Obsahuje objekt tKeybT10.
- **uKeybT03** - Obsahuje objekt tKeybT03.
- **uSimT10** - Obsahuje objekty tSimDispT10 a tSimKeybT10.
- **uSimT10R** - Obsahuje objekt tSimDispT10R.
- **uSimT03** - Obsahuje objekty tSimDispT03 a tSimKeybT03.
- **uSimT03R** - Obsahuje objekt tSimDispT03R.
- **ChnVirt** - Obsahuje objekty tChnVirt a tAddChnVirt.
- **uFont** - Obsahuje základní deklarace typů pro práci s fonty.
- **uBitmap** - Obsahuje základní deklarace typů pro práci s bitmapami.
- **uRLE1** - Obsahuje funkce pro kompresi a dekompresi dat metodou RLE.
- **uCharBuf** - Obsahuje několik objektů pro práci se znakovými buffery sloužícími pro vnitřní potřebu objektů terminálů.
- **GraphWin** - Obsahuje objekty a funkce pro práci s grafickými okny v os. ReTOS. Slouží pro potřeby simulátorů displejů v okně na monitoru PC.
- **NumToStr** - Obsahuje funkce pro převody čísel na řetězce.
- **uString** - Obsahuje objekt pro práci s inicializačními stringy.
- **G128x64** - Obsahuje deklaraci typu videopaměti o rozměru 128 × 64 pixelů, základní funkce pro práci s videopamětí, pole uživatelských bitmap a fontů.
- **G240x128** - Obsahuje deklaraci typu videopaměti o rozměru 240 × 128 pixelů, základní funkce pro práci s videopamětí, pole uživatelských bitmap a fontů.
- **GO128x64** - Obsahuje objekty tGraphicArea a tGrRecorder pro práci s uživatelskou videopamětí a emulaci souřadnicového zapisovače pro rozměr videopaměti 128 × 64 pixelů.
- **GO240x128** - Obsahuje objekty tGraphicArea a tGrRecorder pro práci s uživatelskou videopamětí a emulaci souřadnicového zapisovače pro rozměr videopaměti 240 × 128 pixelů.
- **SetupT10** - Obsahuje proceduru implementující základní setup terminálu TERM10. Je určena k modifikaci podle potřeb uživatele.

## 7. Hierarchie instancí objektů

Z hlediska vzájemných odkazů instancí objektů jsou pro jednotlivé varianty vytvářeny následující hierarchické struktury:

### 7.1. Lokální varianta programového vybavení



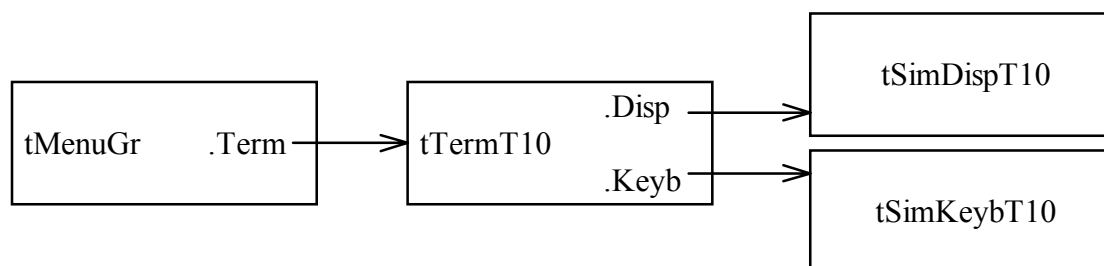
obr. E - Hierarchie instancí - lokální varianta

Příklad vytvoření instance terminálu:

```

pMyTerm:=
New(pTermT10, Init(New(PDispT10, Init(nil, 20, 10, cAdrTerm, True)),
New(PKeybT10, Init(nil, 20, cAdrTerm)), cAdrTerm, nil, nil));
  
```

**Simulace na počítači PC:**



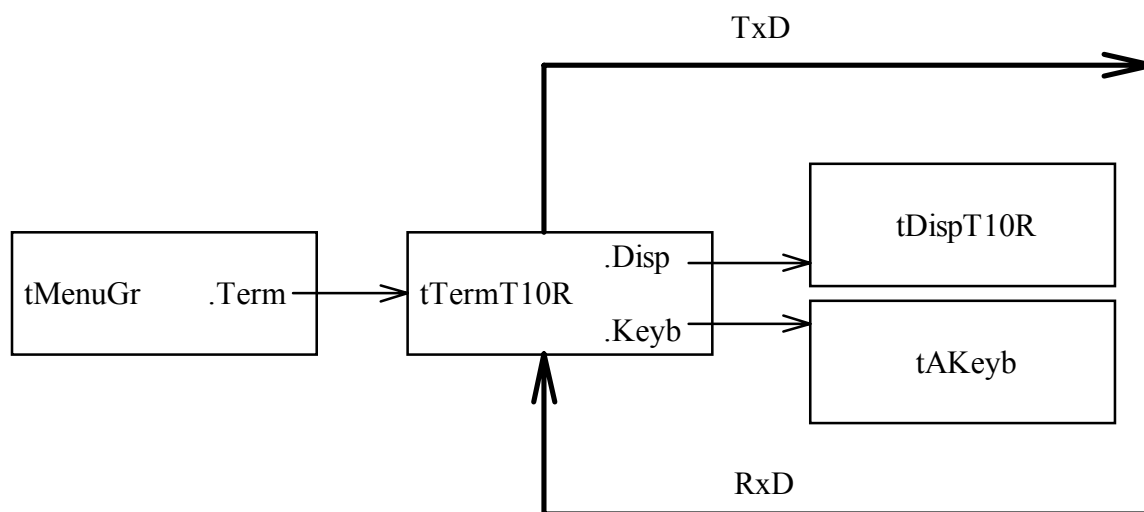
obr. F - Hierarchie instancí - lokální varianta - simulace na PC

Příklad vytvoření instance simulátoru terminálu:

```

pMyTerm:=New(pTermT10, Init(
New(PSimDispT10, Init(nil, 20, 10, cAdrTerm, true, 0, 0, 2, 2, 265, 165, false)),
New(PSimKeybT10, Init(nil, 20, cAdrTerm, FlgEnd)), cAdrTerm, nil, nil));
  
```

## 7.2. Vzdálená varianta programového vybavení



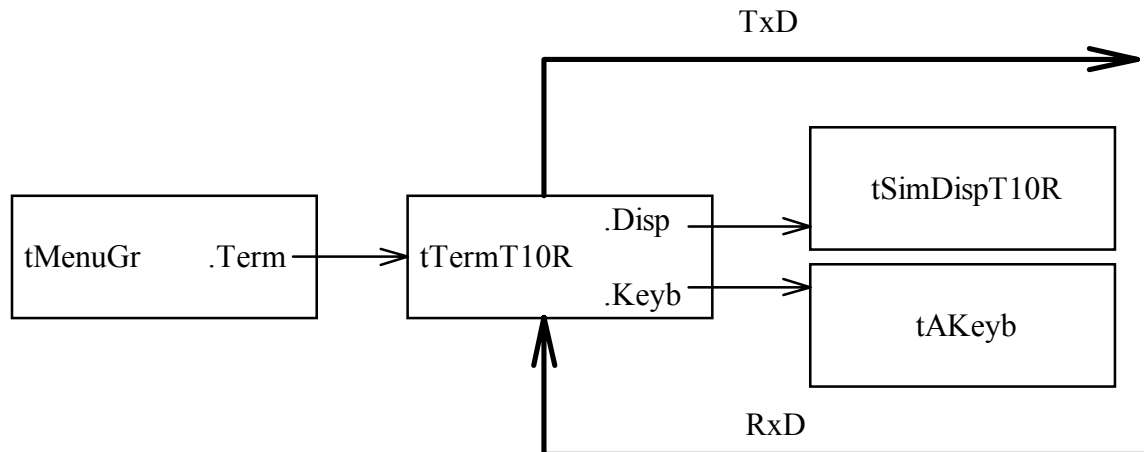
obr. G - Hierarchie instancí - vzdálená varianta

Příklad vytvoření instance terminálu:

```
pMyTerm:=New (PTermT10R, Init (New (PDispT10R, Init (nil, 20, 10, cAdrTerm, false)),
```

```
    New (PAKeyb, Init (nil, 20)),
    cAdrTerm, ChnTerm, RecBuf));
```

Simulace na počítači PC:

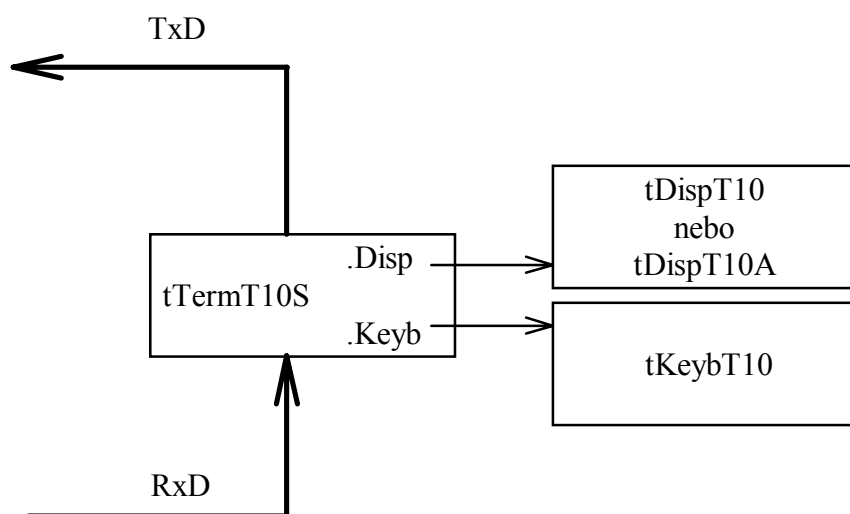


obr. H - Hierarchie instancí - vzdálená varianta - simulace na PC

Příklad vytvoření instance simulátoru terminálu:

```
pMyTerm:=Init (New (PSimDispT10R, Init (nil, 20, 10, cAdrTerm, true,
    0, 0, 2, 2, 265, 165, false {DobleSize})),
    New (PAKeyb, Init (nil, 20)),
    cAdrTerm, ChnTerm, RecBuf));
```

### 7.3. Varianta programového vybavení pro podřízenou periferii

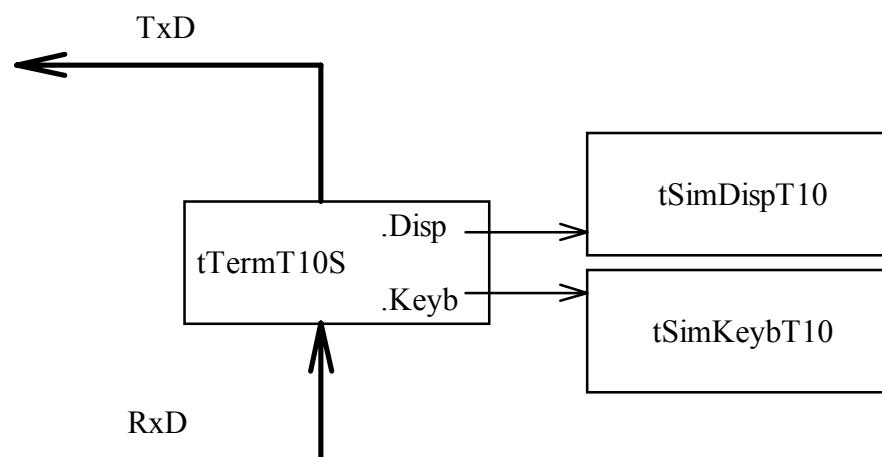


obr. I - Hierarchie instancí - podřízená periferie

Příklad vytvoření instance terminálu:

```
pMyTerm:=
  New(pTermT10S, Init (New(pDispT10, Init (nil, 20, 10, cAdrTerm, true)),
    New(pKeybT10, Init (nil, 20, cAdrTerm)), cAdrTerm, TermChnl, RecBuff));
```

Simulace na počítači PC:



obr. J - Hierarchie instancí - podřízená periferie - simulace na PC

Příklad vytvoření instance simulátoru terminálu:

```
pMyTerm:=
  New(pTermT10S, Init (New(pSimDispT10,
    Init (nil, 20, 10, cAdrTerm, true, 0, 0, 2, 2, 265, 165, false)),
    New(pSimKeybT10, Init (nil, 20, cAdrTerm, FlEnd)),
    cAdrTerm, TermChnl, RecBuff));
```

## 8. Procesy

---

### 8.1. Lokální varianta programového vybavení

---

V lokální variantě je činnost terminálu řízena dvěma procesy:

- proces "MENU" - periodicky volá metodu tMenuGr.Run, která ve svém těle volá metody pro přechod mezi jednotlivými obrazovkami Menu a uživatelské funkce jednotlivých obrazovek.
- proces "TERM" - periodicky volá metodu Tick objektu terminálu, která ve svém těle volá metody Keyb^.Tick a Disp^.Tick. Tyto metody zabezpečují vstup z klávesnice a výstup na displej terminálu.

Protože mohou být metody Tick v některých případech volány pro urychlení činnosti explicitně z kontextu procesu "MENU", jsou chráněny proti vícenásobnému vstupu z více procesů současně. Tato situace je řešena jednoduchým mechanismem, který v případě běhu procedury v jednom procesu odmítne pro druhý proces provést její tělo. Tato skutečnost nemá vzhledem k periodickému provádění činnosti sémantický důsledek.

### 8.2. Vzdálená varianta programového vybavení

---

Tato varianta programového vybavení je určena pro řídicí systém KitV40. V řídicím systému běží stejné procesy jako v terminálu u varianty předchozí. Použitím jiných objektů je však dosaženo přesměrování výstupu na displej a vstupu z klávesnice na komunikační kanál.

### 8.3. Varianta podřízené periferie

---

U této varianty, na rozdíl od předchozích, schází proces "MENU". Terminál je řízen jediným procesem "TERM". V periodicky volané metodě terminálu Tick je pak zajištěn příjem a dekódování dat pro zobrazení a vysílání kódů stisknutých kláves.

## 9. Seznam dokumentů ke Stavebnici objektů

---

- **uATerm** – jednotka pro práci s abstraktním terminálem; popis objektů tATerm, tADisp a tAKeyb.
- **uTermChr** – jednotka pro práci se znakovým terminálem; popis objektu tTermChr
- **uTermGr** – jednotka pro vytváření grafických terminálů; popis objektů tGraphicPropReader, tStylePropertyReaderWriter, tGraphicDataMan a tTermGr.
- **uAMenu** – jednotka pro vytváření uživatelských menu; popis objektů tStackMenu a tAMenu.

- **uMenuChr** – jednotka pro vytváření znakových uživatelských menu; popis objektu tMenuChr.
- **uMenuGr** – jednotka pro vytváření grafických uživatelských menu; popis objektu tMenuGr.
- **uTermT10** – jednotka implementující terminál TERM10; popis objektu tTermT10.
- **uTerm10R** – jednotka implementující vzdálený terminál TERM10; popis objektu tTermT10R.
- **uTerm10S** – jednotka implementující terminál TERM10 jako podřízenou periférii; popis objektu tTermT10S.
- **uDispT10** – jednotka implementující displeje terminálu TERM10; popis objektů tADispT10, tDispT10 a tDispT10A.
- **uKeybT10** – jednotka implementující klávesnici terminálu TERM10; popis objektu tKeybT10.
- **uSimT10** – jednotka implementující objekty pro simulaci terminálu TERM10 na počítači PC; popis objektů tSimDispT10 a tSimKeybT10.
- **uDspT10R** – jednotka implementující displej vzdálené varianty terminálu TERM10; popis objektu tDispT10R.
- **uSimT10R** – jednotka simulátor vzdálené varianty terminálu TERM10 na počítači PC; popis objektu tSimDispT10.
- **uTermT03** – jednotka implementující terminál TERM03; popis objektu tTermT03.
- **uTerm03R** – jednotka implementující vzdálený terminál TERM03; popis objektu tTermT03R.
- **uTerm03S** – jednotka implementující terminál TERM03 jako podřízenou periférii; popis objektu tTermT03S.
- **uDispT03** – jednotka implementující displej terminálu TERM03; popis objektu tDispT03.
- **uKeybT03** – jednotka implementující klávesnici terminálu TERM03; popis objektu tKeybT03.
- **uSimT03** – jednotka implementující objekty pro simulaci terminálu TERM03 na počítači PC; popis objektů tSimDispT03 a tSimKeybT03.
- **uDspT03R** – jednotka implementující displej vzdálené varianty terminálu TERM10; popis objektu tDispT03R.
- **uSimT03R** – jednotka simulátor vzdálené varianty terminálu TERM03 na počítači PC; popis objektu tSimDispT03.



## 10. Seznam uživatelsky významných proměnných, procedur, funkcí, metod atd.

---

### •G240x128 a G128x64

User\_FontArray  
User\_BkBmps  
mClear  
mDrawPoint  
mDrawLine  
mDrawBox  
mFillBox  
mTextOutXY

### •GO240128 a GO128x64

tGraphicArea  
tGraphicRecorder

### •SetupT10

tTermT10SetupData  
InitSetupT10

### •uDispT10

tDispT10.Init  
tDispT10.Done  
tDispT10A.Init  
tDispT10A.Done

### •uDspT10R

tDispT10R.Init

### •uKeybT10

tKeybT10.Init

### •uSimT10

tSimKeybT10.Init  
tSimDispT10.Init

### •uSimT10R

tSimDispT10R.Init  
Cr\_SimTermT10R

### •uTerm10R

tTermT10R.Init  
tTermT10R.FILight  
tTermT10R.LedSign  
tTermT10R.DispContr  
tTermT10R.BellOn  
tTermT10R.BellOff  
Cr\_TermT10R

### •uTerm10S

tTermT10S.Init

### •uTermT10

tTermT10.Init  
tTermT10.Done  
tTermT10.FILight  
tTermT10.DispContr  
tTermT10.LedSign  
tTermT10.WriteOut  
tTermT10.ReadIn  
tTermT10.BellOn  
tTermT10.BellOff  
tTermT10.BeepKeyOn  
tTermT10.BeepKeyOff  
tTermT10.SetUserStart  
tTermT10.SetUserStop  
tTermT10.SetOnSetupTerminal  
tTermT10.EnableRefreshOut  
tTermT10.DisableRefreshOut

### •uTermGr

cNumFonts  
BkBmpsCount  
tVideoRop

### •uMenuGr

tMMenuGr  
tMenuGr.Init  
Grafický souřadný systém  
Syntaxe grafického stringu  
Syntaxe textového stringu

### •uMenuGr

tMMenuGr  
tMenuGr.Init  
Grafický souřadný systém  
Syntaxe grafického stringu  
Syntaxe textového stringu

### •uMenuChr

tMMenuChr