

***Vyšší odborná škola a Střední průmyslová škola Šumperk  
Gen. Krátkého 1, 787 29 Šumperk, okres Šumperk***



**Minimum o obrazkách v DetStudios**

**Autor: Viktor Ptáček**  
**Pracoviště: VOŠ a SPŠ Šumperk**  
**Ročník: 3**  
**Rok: 2007/2008**  
**Obor: elektrotechnika**

## Obsah

<b>1 ÚVOD</b> .....	<b>3</b>
<b>1.1 OBRAZOVKY</b> .....	<b>3</b>
<b>2 OBRAZOVKY A DETSTUDIO</b> .....	<b>3</b>
<b>2.1 PROSTŘEDÍ PRO VYTVÁŘENÍ OBRAZOVEK:</b> .....	<b>4</b>
2.1.1 Postup při vytváření obrazovek:.....	4
<b>2.2 SEZNAM ASI NEJPOUŽÍVANĚJŠÍCH OBJEKTŮ:</b> .....	<b>6</b>
<b>2.3 UDÁLOSTI + SKRIPTOVACÍ JAZYK</b> .....	<b>8</b>
<b>2.4 VYTVOŘENÍ JEDNODUCHÉHO FUNKČNÍHO PROGRAMU</b> .....	<b>11</b>

## 1 ÚVOD

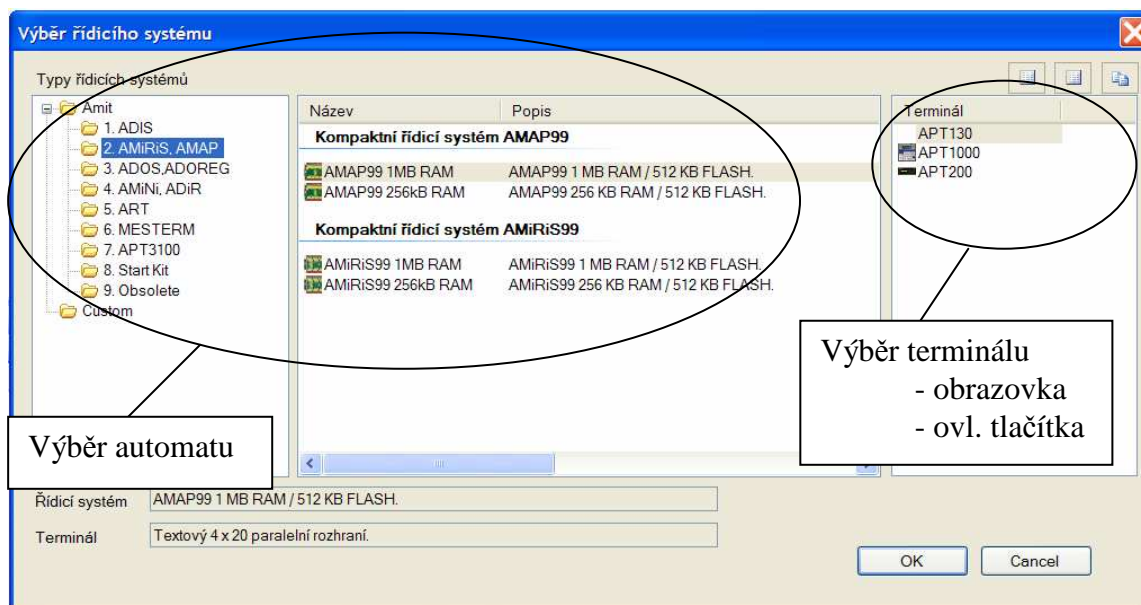
Tento text jsem začal psát po pátrání po manuálu, jak programovat obrazovky. Na naší škole jsem objevil manuál, ale byl velice stručný a ještě k tomu byl ke staršímu prostředí PSP3. Takže jediným mým vodítkem, jak se programují obrazovky, byla nápověda integrovaná přímo v programu DetStudio. Protože jsem spoluvůrcem učební pomůcky AT08, kde je vhodné použít obrazovky, bylo také nutné vytvořit text, který by navedl studenty, jak si takové obrazovky vytvořit a jak s nimi pracovat.

### 1.1 OBRAZOVKY

K čemu se vůbec obrazovky používají? – může to být například určitá nápověda pro obsluhu, kde jí řeknete co má udělat a co se stane, když to udělá. Další možností je zobrazovat určité výsledky práce, stavy jednotlivých senzorů a spoustu dalších důležitých věcí. Nebýt obrazovek, museli bychom do automatizačního systému umístit převodníky na speciální displaye a to je krajně nepohodlné, a proto nám výrobci tímto usnadňují práci. K obrazovkám jsou většinou ještě integrována tlačítka, která slouží k ovládnání a pohybu mezi obrazovkami, nebo jejich virtuální výstup můžeme využít v našem programu.

## 2 OBRAZOVKY A DETSTUDIO

Využívat obrazovky můžete jen tehdy, je-li na PLC obrazovka integrovaná nebo připojíte-li vhodný terminál pro vaše PLC. To ovšem musíte „řít“ DetStudiosu již při vytváření projektu:



Z daného screenu vyplývá, že k vybranému typu automatu (AMiRiS, AMAP) je možné vybrat různé terminály. Ale třeba u AMiNi2D nemáme na výběr z různých variant, jelikož toto PLC na sobě již obsahuje obrazovku i terminálové tlačítka a zároveň nemá interface pro připojení jiného terminálu.

## 2.1 PROSTŘEDÍ PRO VYTVÁŘENÍ OBRAZOVEK:

**Pojmy:** obrazovka **global** – tato obrazovka nemá grafický výstup (je to neviditelná obrazovka), která běží na pozadí z čehož vyplývá, že do této obrazovky budeme umisťovat takový kód, který se nemá vykonávat jen když jsme v určité obrazovce, ale má se vykonávat stále.

*Pozn: Obrazovka global se dá nahradit tak, že kód v obrazovce global vyjmeme a nakopírujeme do veškerých obrazovek nacházejících se v projektu (není efektivní).*

**event** – událost nastane-li událost (třeba stisk tlačítka) vykoná se kód který, začíná koncem řádku event..... a končí end;

**metoda** – je volání funkce, která vykoná určitou činnost. Metody mají předdefinované objekty. Například metoda obrazovky show: pokus.show(). Tímto příkazem ve skriptovacím jazyku zavřete aktuální obrazovku a otevřete obrazovku pokus.

### Důležité věci, které musíme udělat před programováním obrazovek

- vybrat terminál (popsáno v předcházející stránce)
- vytvořit proces ST (většinou typu idle) a umístit do něj  
„Lcw3Idle NONE“ (některé verze DetStudia vytvářejí tento proces automaticky)
  - tento příkaz určuje - když program vstoupí do tohoto procesu, propojí informace z obrazovek a terminálu s tlačítky s kódem ostatních procesů

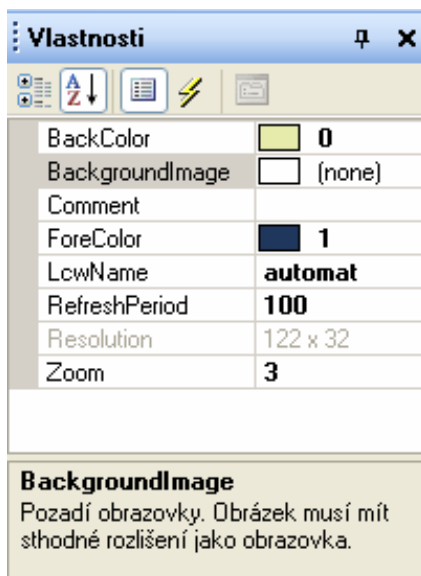
### 2.1.1 Postup při vytváření obrazovek:

#### 1) vytvořit obrazovku

- v pravém dialogu Projekt dvakrát klikneme na položku obrazovky (všimneme si, že při rozkliknutí položky obrazovky zde máme defaultně obrazovku global) -> v hlavním okně máme prázdný seznam, klikneme druhým tlačítkem a vybereme přidat obrazovku, vyskočí vám dialog, ve kterém zvolíme název a komentář.

#### 2) vlastnosti obrazovky

Po kliknutí na obrazovku se nám v pravém dolním rohu obrazovky zobrazí její vlastnosti:



**BackColor** – jaká bude barva spodní vrstvy obrazovky (máme na výběr jen ze 2 – je to monochromatický display)

**BackgroundImage** - je tu možnost zvolit obrázek pozadí, ale kvůli dvoubarevnému display a malému rozlišení si myslím, že se moc nevyužívá.

**Comment** – komentář obrazovky

**ForeColor** – barva přední vrstvy obrazovky

**LcwName** – jméno obrazovky

**RefreshPeriod** - obnovovací frekvence Obrazovky

**Resolution** - rozlišení (pevně dané)

**Zoom** - zvětšení obrazovky při návrhu (jen pro účel lepšího zobrazení obrazovky při návrhu)

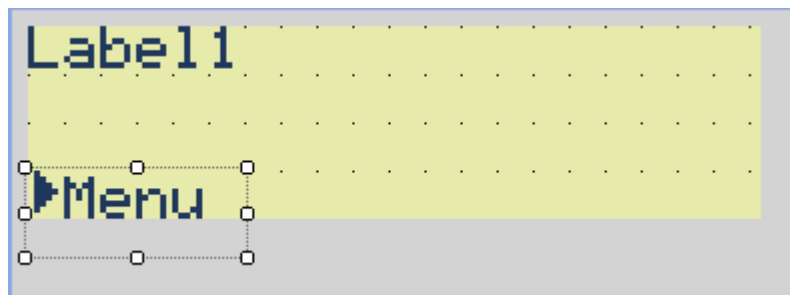
### 3) editor obrazovek

Po vytvoření obrazovky se nám tato obrazovka zobrazí v seznamu obrazovek v panelu Projekt. Pro vyvolání editoru obrazovek postačí dvojklik na tuto obrazovku v panelu Projekt.

Zobrazí se nám:

#### 1) návrh obrazovky (bez objektů, na obr. už jsou umístěny 2 objekty)

V této nabídce vlevo dole můžeme vidět tlačítka funkce, skript a návrh (nacházíme se v návrhu).



#### 2) Toolbox se seznamem použitelných objektů

Jestliže chceme použít některý z objektů, ať už je to viditelný objekt nebo Neviditelný, vždy jej musíme tahem myši umístit na nějaký volný prostor na obrazovce. (základní objekty budou popsány na další stránce)

#### 3) Vlastnosti

Rozšířené vlast	
(Name)	Menu1
BackColor	0
CursorWrap	False
Font	Tridat 7
ForeColor	1
HideSelection	False
ItemHeight	8
Location	0; 24
MenuItems	(Collection)
ShowScrollBar	False
Size	36; 14
TabIndex	0
TextAlignment	MiddleLeft

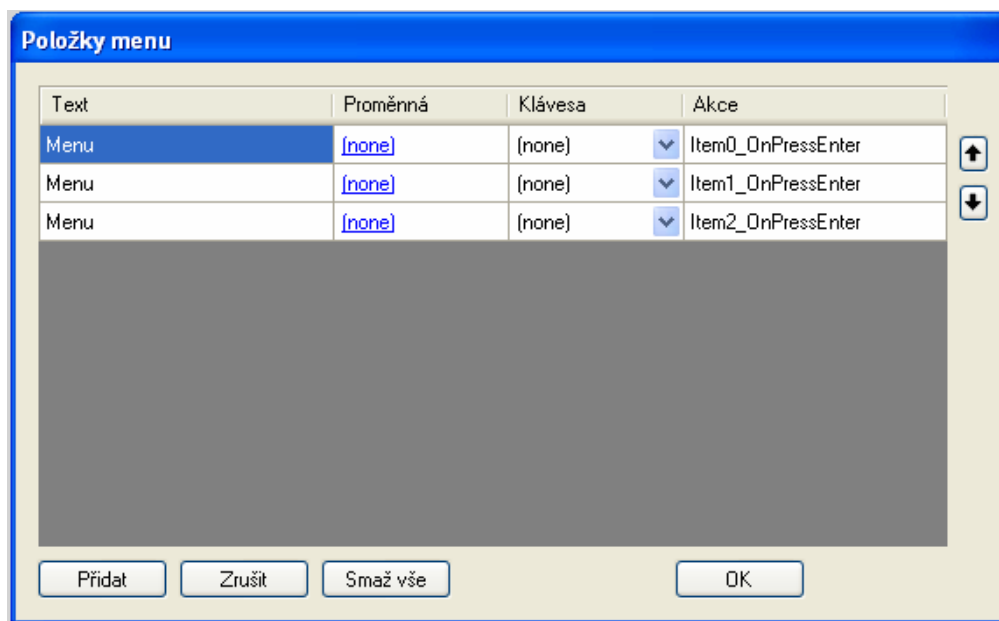
**ItemHeight**  
Výška jedné položky.

Jsou to vlastnosti obrazovky. Tento panel je velice důležitý, jelikož po vytvoření objektu na obrazovce a jeho označení se zde objeví vlastnosti právě tohoto objektu.

V tomto panelu se nemusí zobrazovat jen vlastnosti, ale dají se zde také zobrazit veškeré použitelné události (event) pro označený objekt. Tuto nabídku zobrazíte tlačítkem ve tvaru blesku.

## 2.2 SEZNAM ASI NEJPOUŽÍVANĚJŠÍCH OBJEKTŮ:

- Label** - je to text, který má ryze oznamující funkci (popisek)  
- nejpoužívanější vlastnost tohoto objektu je asi `visible` (volá se „`jmenolabelu.visible = true`“ - viditelný, „`jmenolabelu.visible=false`“ - neviditelný)  
- bez událostí
- Menu** - objekt pro rozskok na jednotlivé další obrazovky, nebo pro výběr činnosti  
- po umístění tohoto objektu na obrazovku je nutné na něj dvakrát kliknout pro editaci jeho položek



- vyžadujeme-li smazání (zrušit) nebo vytvoření (přidat) či přehození (šipka nahoru a dolů) jednotlivých položek, využijeme k tomu vhodná tlačítka.
- zde upravíme jména položek dvojklikem na kolonku text jednotlivých položek
- ostatní sloupce necháme tak, jak jsou
- po odklepnutí (OK) se nám zobrazí část skriptovacího jazyka, kde musíme Nadefinovat, co se stane když uživatel vybere jednotlivé nabídky (ale o tom v části události)
- často používaná událost: `Menu1_ItemX_OnPressEnter`

- Keybit** - pro nastavení hodnoty bitu určité proměnné
- je to neviditelný objekt a po umístění na obrazovku se nám zobrazí v dolní části pracovní plochy
  - v jeho vlastnostech je nejdůležitější `KeyCode` – kterou klávesou chcete tento bit měnit, `mode` – set (po nahrání 1 do bitu) reset (pro nahrání 0 do bitu), toggle (když je v bitu 1 nahraje 0 a když 0 nahraje 1)
  - objekt bez událostí

**Keyscreen** - objekt, který nám zavře aktuální obrazovku a otevře jinou

- tento přechod se vyvolá pomocí tlačítka nadefinovaného ve vlastnostech parametrem KeyCode
- pro vyjádření, která obrazovka se má otevřít, je ve vlastnostech parametr GoToScreen
- objekt je bez událostí

**Key** - je možnost psaní kódu, co se má stát, když se stiskne tlačítko

- jeho nejdůležitější parametrem je KeyCode – kterým tlačítkem se bude kód vyvolávat
- obsahuje 3 události: OnKeyPress – generuje puls, při držení tlačítka generuje sled stejných pulzů – při každém pulzu se vykoná kód, OnKeyDown – kód se vykoná, když stiskneme tlačítko, OnKeyUp – kód se vykoná při uvolnění tlačítka.

**NumericEdit** – možnost měnění proměnné po řádech – můžeme se přesunovat na jednotlivé řády čísla a ty měnit

- důležité parametry- Format – určujete kolik bude mít zobrazovaná hodnota míst, kde bude desetinná čárka...., Minimum – minimální hodnota proměnné, maximum – maximální hodnota proměnné, StartEdit – jakým způsobem začneme editovat, buďto jen najetím na tento objekt pomocí šipek - auto, nebo manuál, když najedeme na tento objekt šipkami musíme ještě použít klávesu potvrzující start editace, podobná vlastnost ale pro ukončení editace je AfterEdit
- událost - OnEditComplete – kód se vyvolá po ukončení editace.

**NumericUpDn** – možnost měnění hodnoty proměnné, pomocí tlačítek dolů a nahoru

- důležité parametry – IncrementMajor – o kolik se změní hodnota při krátkém stisku tlačítek, IncrementMinor – o kolik se změní hodnota proměnné při delším stisku tlačítek, Format – určujeme, kolik bude mít zobrazovaná hodnota míst, kde bude desetinná čárka...., Minimum – minimální hodnota proměnné, maximum – maximální hodnota proměnné
- událost - OnEditComplete – kód se vyvolá po ukončení editace.

**NumericView** – možnost zobrazení aktuální hodnoty proměnné

- po dvojkliku na tento objekt se nám zobrazí dialog, ve kterém vybereme proměnnou, jejíž hodnotu chceme zobrazit (tento dialog také vyvoláme pomocí dvojkliku na vlastnost variable v panelu vlastnosti)
- dalším parametrem format, kde si určíme, kde má být desetinná čárka, kolik pozic před čárkou a za čárkou se má zobrazit...
- objekt je bez událostí

## 2.3 UDÁLOSTI + SKRIPTOVACÍ JAZYK

Většina názvů událostí je napsaná anglicky tak, aby byla všem srozumitelná. Jestliže je to událost OnOpen (událost obrazovky), tak je to událost, která se vyvolá, když bude určitý screen otevřen (načte se obrazovka).

Vše si vysvětlíme na příkladu - vytvoříme si libovolnou obrazovku, pomocí druhého tlačítka na ní a stisknutím myši na položku zobrazit, se nám zobrazí návrhové prostředí. Klikneme na obrazovku a tím si zobrazíme vlastnosti obrazovky. Překlikneme se pomocí tlačítka s bleskem v panelu vlastností do seznamu událostí. Zde si vybereme požadovanou událost. Tak třeba OnOpen. Na tuto událost klikneme a v pravé části nabídky se nám ukáže tlačítko s „...“. Na toto tlačítko s tečkami klikneme. Tím se hlavní panel přemění z návrhu na skript. A zároveň se nám zde zobrazí kód volání události:

Nezapomeňme na () při zapisování událostí z hlavy

*Pozn: to co je na řádce za // je komentář a kompilátor programu je přeskočí, je to jen informace pro programátora, aby věděl, co kódem zamýšlel. Je dobré komentáře psát, aby třeba po vrácení se k programu po nějaké době, se v něm programátor neztratil.*

```
event pokus_OnOpen()  
    //tělo události  
end;
```

Je to událost OnOpen a je přiřazena obrazovce jménem pokus.  
Do těla události napíšeme, co se má stát po otevření obrazovky pokus.  
Základní příkazy události pro skript:

```
If (Výraz)Then  
Else;  
endIf;
```

- je to skoro stejné jako structure text jen o něco jiná syntaxe.

Spíše se kód podobá kódu programovacího jazyka C++. Výhoda třeba u příkazu If je, že do podmínky můžeme umístit třeba logický součin, porovnání... ne jak v ST jen binární proměnnou.

Další věcí, kterou k programování budeme potřebovat, je přiřazovací operátor. Tím u skriptovacího jazyka už není let ale jen „=“.

Takže pokud budeme chtít při otevření obrazovky pokus přiřadit do proměnné pokus typu float 50, pak bude kód skriptu vypadat takto:

```
event pokus_OnOpen()  
    pokus = 40;  
end;
```

- za každým příkazem musí být ukončení a to v podobě ; , to ovšem neplatí pro volání události. A dále pro příkaz If, který má syntaxi přesně takovou, jak je zde napsáno. Často se na středníky zapomíná a dosavadní kompilátor nahlásí jen chybu, ale nevypíše jakou, proto si na to dávejme pozor.

Na dalším příkladu zadáme znění úkolu, napíšeme vám zdrojový kód - budou tam dvě chyby, které se pokuste najít, a pak okomentujte kód.

Mám alias @motor pomocí objektu key1, který jsme umístili na obrazovku a nakonfigurovali mu příslušné vlastnosti.

Podívali jsme se do vlastností přepnuli jsme se do událostí a zjistili, že tento objekt má události OnKeyUp, OnKeyDown, OnKeyPress.

A teď potřebujeme, aby při držení nakonfigurovaného tlačítka, byl motor v 1 a když ho pustíme, aby byl v 0.

Dále chceme nezávisle na stavu tlačítka monitorovat chod motorku a to na aliasu @signalizace. K tomuto použijeme událost obrazovky a to OnRefresh.

Kód

```
event pokus_OnRefresh() _____  
    @signalizace = @motor _____  
end;  
  
event Key1_KeyUp() _____  
    @motor = 1; _____  
end;  
  
event Key1_OnKeyDown() _____  
    @motor = 0 _____  
end;
```

Přepište tento kód do jednodušší verze: nepoužívejte objekt key ale prvek keybit.

Vypište 2 základní vlastnosti, které nastavujete a jak je nastavíte.

1) \_\_\_\_\_  
2) \_\_\_\_\_

```
event _____ _____  
end;
```

## Skript a objekt Menu

Často používaná událost je událost objektu menu, tady si také ukážeme jednu z metod a způsob jejího zápisu.

Jestliže na obrazovku umístíme položku menu, jsme vyzváni, abychom zadali jména a počet položek a to pomocí dialogu, nebo jej můžeme po nějaké době vyvolat dvojklikem na tento objekt. Po potvrzení se nám spustí skriptovací jazyk a to s kódem:

```
event Menu1_Item0_OnPressEnter() //vykoná se po stisku enter na //první
                                položku
end;

event Menu1_Item1_OnPressEnter() //vykoná se po stisku enter na //druhé
                                položku
end;

event Menu1_Item2_OnPressEnter() //vykoná se po stisku enter na //třetí
                                položku
end;
```

Tyto události se vyvolají, když obsluha vybere z menu jednotlivé položky. Do těchto událostí můžeme napsat jakýkoliv kód skriptovacího jazyka, ale většinou chceme po výběru natáhnout předem specifikovanou obrazovku místo té, co zrovna vidíme.

To učiníme voláním metody: `obrazovka.Show()`

Přepsání kódu tak, aby vždy po vybrání položky z menu byla otevřena jiná obrazovka:

Kód:

```
event Menu1_Item0_OnPressEnter()
    obrazovka1.show() //volání metody
end;

event Menu1_Item1_OnPressEnter()
    obrazovka2.show() //volání metody
end;

event Menu1_Item2_OnPressEnter()
    obrazovka3.show() //volání metody
end;
```

## 2.4 VYTVOŘENÍ JEDNODUCHÉHO FUNKČNÍHO PROGRAMU

Půjde o jednoduchý program, který bude obsahovat 2 motory, které budeme zvlášť ovládat a k ovládání nás navedou.

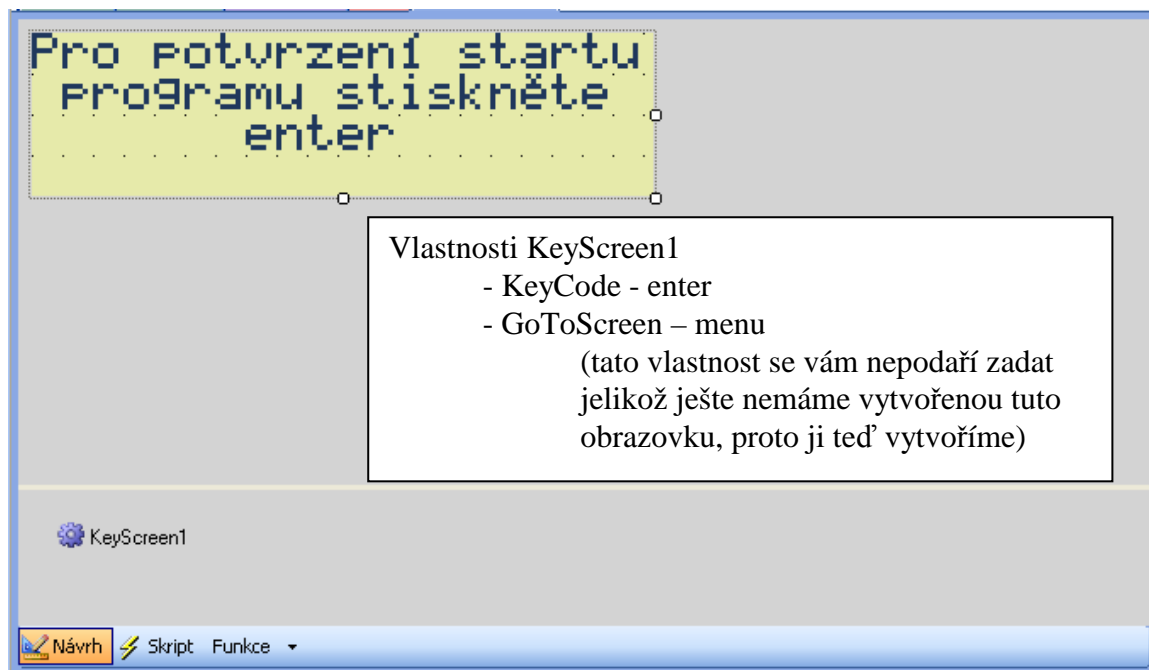
### Zadání:

Když zapneme automat uvítá nás první obrazovka a vyzve, abychom potvrdili spuštění programu jednou z kláves. Po potvrzení se nám zobrazí obrazovka, která bude říkat obsluze: „Vyberte objekt který chcete řídit“ dále tato obrazovka bude obsahovat objekt menu, který bude mít položky: Menu, Motor1, Motor2. Menu bude položka, která nebude dělat nic, je tu jen kvůli vzhledu, aby se na obrazovce hned neobjevil motor1 nebo 2, ale aby zde bylo napsáno menu. Po vybrání Motor1 nebo Motor2 bude na obrazovce vypsáno jakými tlačítky se bude motor zapínat a vypínat, a pomocí tlačítka esc se vrátíme zpět do menu.

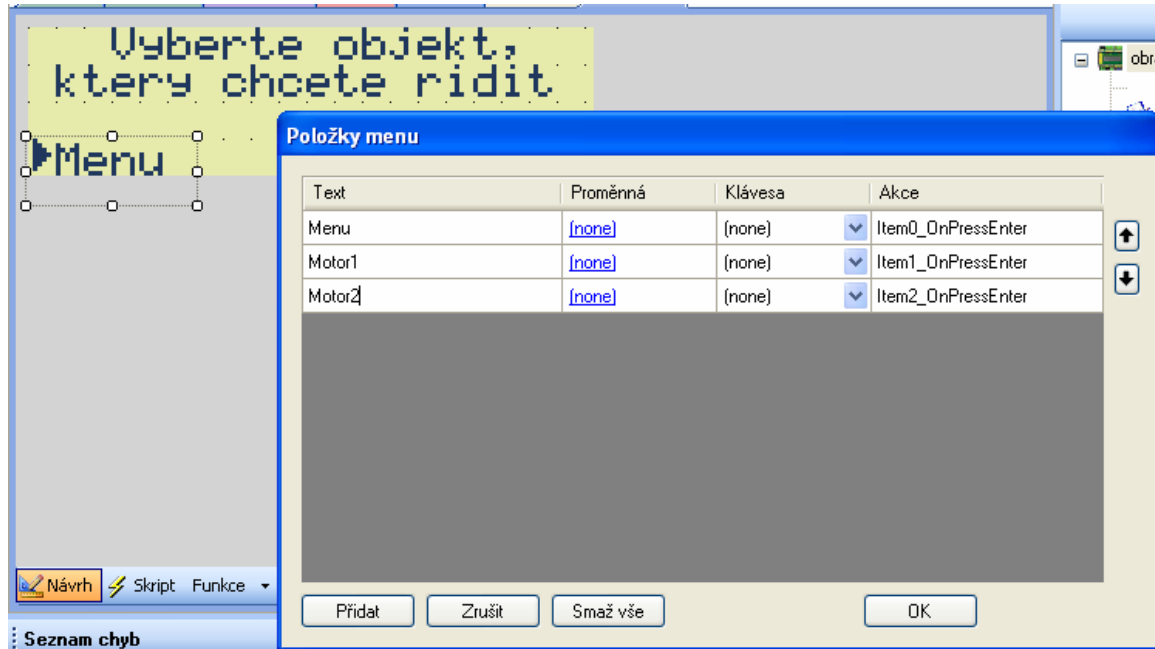
- 1) Vytvoříme projekt
- 2) Zvolíme parametry projektu
- 3) Vytvoříme proměnné DO a DI
- 4) Vytvoříme proces pro čtení a zápis proměnných na výstupy a vstupy
- 5) Vytvoříme aliasy: @motor1 a @motor2 (DO.0, DO.1)

Obrazovky:

- 6) Vytvoříme proces idle s příslušným příkazem pro obrazovky
- 7) Vytvoříme obrazovku, která bude mít název uvodni  
(jestliže nám automaticky vytvoří obrazovku vývojové prostředí, můžeme ji využít a zeditovat)
- 8) Vložíme na ni objekty label a keyscreen



- 9) Vytvoříme další obrazovku se jménem menu
- 10) Vrátime se na obrazovku uvodni a zadáme vlastnost GoToScreen objektu KeyScreen jako menu
- 11) Přesuneme se zpátky na obrazovku menu a vložíme sem objekty menu a label dle obrázků



- 12) Po odklepnutí položek v menu nám vyskočí skript s kódem:

```
event menu_OnOpen() //tato událost je u každé obrazovky
    menu.FocusFirstControl();
end;

event Menu1_Item0_OnPressEnter()
end;

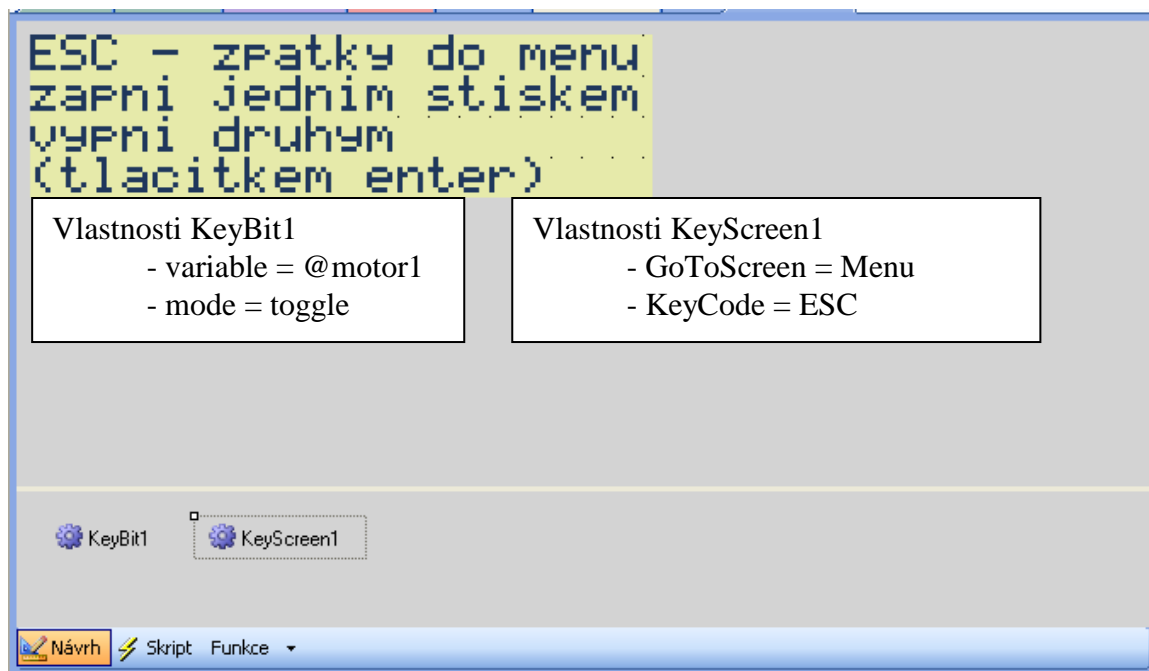
event Menu1_Item1_OnPressEnter()
    motor1.show();
end;

event Menu1_Item2_OnPressEnter()
    motor2.show();
end;
```

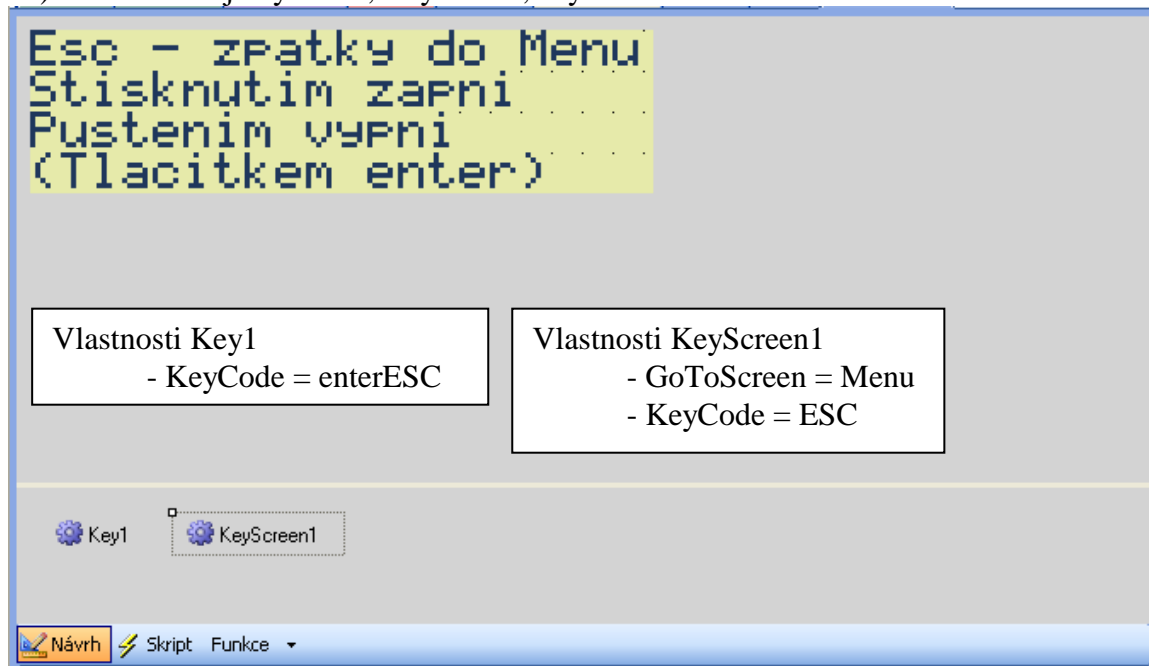
- 13) Upravíme tento kód tak, aby se nám při vybrání Motor1 objevila obrazovka motor1 a při výběru Motor2 vyskočila obrazovka motor2

*Pozn.: jestliže bychom měli vytvořené obrazovky motor1 a motor2, pak po zadání motor1. vyskočí nabídka možných zadání kódu, a stačí jen vybrat*

- 14) Vytvoříme novou obrazovku se jménem motor1  
(úkol pomocí enter budeme ovládat motor1 - stiskem zapneme, stiskem vypneme)  
15) Umístíme na ni objekty: KeyScreen, KeyBit, label



- 16) Vytvoříme novou obrazovku se jménem motor2  
(úkol pomocí enter budeme ovládat motor1 – stisknutím zapneme, puštěním vypneme)  
- použijeme události objektu key – OnKeyDown, OnKeyUp  
17) Umístíme objekty: label, KeyScreen, key



- 18) U objektu Key1 se podíváme na události (tlačítko s bleskem v panelu vlastnosti)
- 19) Vybereme událost OnKeyDown (pomocí tlačítka ...)
- 20) Vrátime se zpět do návrhu, označíme objekt Key1 a vybereme událost OnKeyUp (...)
- 21) Upravíme kód ve skriptu

*Pozn.: Pro pohyb mezi skriptem a návrhem je možné využít tlačítka ve spodní části hlavního okna*

Kód:

```
event motor2_OnOpen()  
    motor2.FocusFirstControl();  
end;  
event Key1_OnKeyDown()  
    @motor1 = 1;  
end;  
event Key1_OnKeyUp()  
    @motor1 = 0;  
end;
```

Zkompilujeme, nahrajeme do automatu, připojíme na DO.1 a DO.2 indikační zařízení a můžeme testovat.

### 3 ZÁVĚR

Doufám, že vám tento text byl dobrým průvodcem při tvorbě obrazovek. Zdrojový kód s vygenerovanou dokumentací je přiložen v textu v učitelské verzi.