

Application Note 13

Tasten programmieren mit eigerScript Methoden und Beispiele

Christoph Angst

Unterägeri, 24. September 2007

© S-TEC electronics AG, CH-6300 Zug
s-tec@bluewin.ch
www.s-tec.ch
www.eigergraphics.com



Dieses Dokument können Sie von der Internetseite www.eigergraphics.com als PDF-Datei herunterladen.

Inhaltsverzeichnis

Einleitung	2
Komponenten und Schnittstellen des FOX57	3
Komponenten und Anschlüsse von FOX57	3
Schnittstelle für Keyboard Input.....	4
eigerScript-Methoden für Key-Inputs	5
HotKey-Class und HotKey-Methoden	5
Anwendungsbeispiele für HotKey-Methoden	7
eigerScript-Programm für eine Tasten-Applikation.....	8
Anordnung und Belegung der Tasten	8
Screenshots der Tasten-Applikation	9
eigerScript Programm-Code für Tasten-Applikation.....	12

Einleitung

Das eigerPanel 57 kann für reine Touch- oder Tastenanwendungen oder für eine Kombination von Touch und Tasten eingesetzt werden. In der Folge wird gezeigt, welche eigerScript Methoden für den Einsatz von Tasten zur Verfügung stehen. Ein Programmbeispiel soll Ihnen die praktische Anwendung der Tasten-Methoden vordemonstrieren. Das Programm-Skript des Beispiels können Sie ins eigerStudio kopieren oder direkt als ganzes eigerScript-Projekt „KEYS“ (inkl. den Startdateien FOXLOGOLEGI und START.FOX) von der ZIP-Datei via CompactFlash Card auf das eigerPanel 57 übertragen.

Komponenten und Schnittstellen des FOX57

Komponenten und Anschlüsse von FOX57

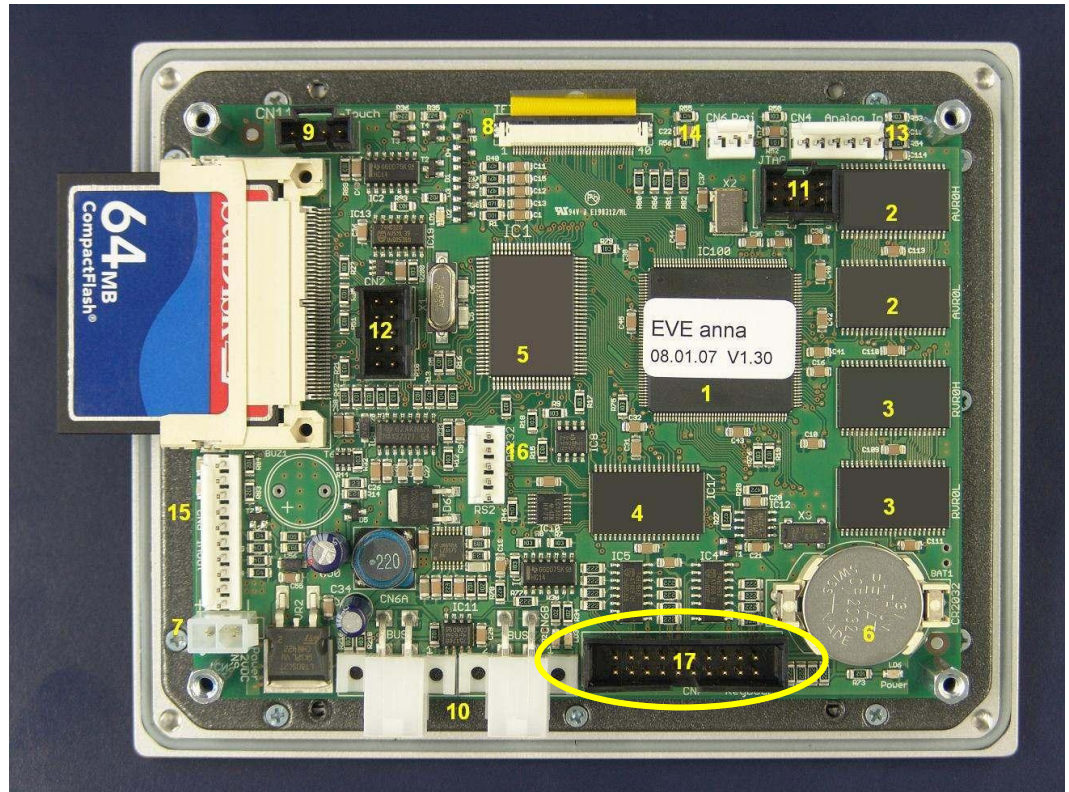


Abbildung 1: Komponenten des embedded computers für das eigerPanel 57

- | | |
|--|---|
| 1. Graphic Controller, eigerVideo Engine (EVE anna) | 9. Anschluss Touchpanel |
| 2. Videospeicher Accessible Video Ram (AVR) | 10. BUS (Serielle Schnittstelle RS485) |
| 3. Videospeicher Refresh Video Ram (RVR) oder mit Adapterkabel F4337 als FOX-COM1 (UART1) verwendbar | 11. Programmier-Schnittstelle für EVE anna (JTAG) |
| 4. Arbeitsspeicher (RAM) | 12. Programmier-Schnittstelle für Microprozessor (S-PROG10) |
| 5. CPU (Micro-Prozessor) | 13. Analog-Eingänge |
| 6. RTC-Batterie | 14. Analogeingang Poti |
| 7. Power Supply 12V (serielle Schnittstelle RS232) | 15. Externe I/O |
| 8. Anschluss Display VGA | 16. FOX-COM2 (UART2) |
| | 17. 16-KEY Keyboard Input |

Schnittstelle für Keyboard Input

Der FOX embedded computer kann über die Schnittstelle CN7 mit bis zu 16 Tasten bedient werden (vgl. Abbildung 1 und Abbildung 2). Die Tasten sind auf dem Schema mit den Grossbuchstaben **A bis P** identifiziert (vgl. Schema in Abbildung 2).

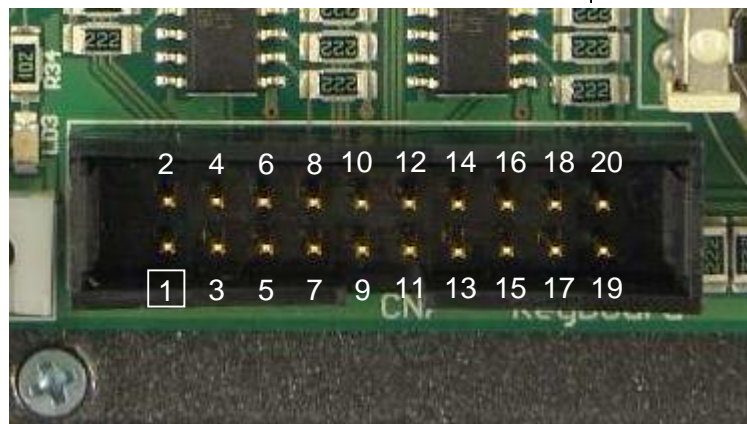
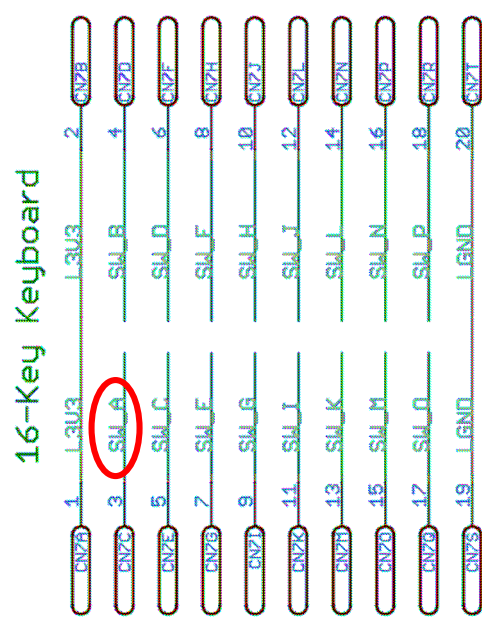


Abbildung 2: 16-Key Keyboard Input – Schema und Foto des BoxHeaders auf dem eigerPanel 57. Die Key-Pins sind mit den Grossbuchstaben A-P identifiziert (SW_A bis SW_P).

eigerScript-Methoden für Key-Inputs

Die Methoden für das Management der Key-Inputs sind in eigerScript in der HotKey-Class zusammengefasst. Die nachfolgende Beschreibung dieser Klasse und der zugehörigen Methoden sind mit einigen Anpassungen dem Befehlskatalog „Befehlsreferenz_eigerScript_eVM.pdf“ entnommen.

HotKey-Class und HotKey-Methoden

HotKeys sind Funktionstasten, die an der Seite des Displays angeordnet sind oder ein Keypad, das an den Rechner angeschlossen ist. Die Tasten können beim Drücken und beim Loslassen einen Event erzeugen. Für die Tasten, die ausgewertet werden sollen, muss der EventHandler installiert sein. Mit den HotKeys kann eine Anwendung ohne Touchpanel realisiert werden. Die HotKeys können in jedem Fall verwendet werden, egal ob ein Touchpanel vorhanden ist oder nicht. Für eine Maschinenbedienung kann eine Kombination sinnvoll sein, indem die am häufigsten gebrauchten Funktionen zusätzlich als Hardwaretasten bedient werden können. Eingaben erfolgen dann mit dem Touchpanel oder mit den HotKeys.

Hardwaremässig sind 16 Funktionstasten vorhanden. Die Tasten werden am eigerPanel 57 an den BoxHeader CN7 angeschlossen (beim FOXS an den BoxHeader CK5 oder die roten AMP microMatch-Stecker. An die roten AMP-Stecker können 2x 6 Tasten und 2 x 2 Tasten angeschlossen werden. Der BoxHeader und die roten AMP-Stecker belegen die gleichen Eingänge). Auf dem BoxHeader sind alle 16 Tastatureingänge vorhanden. Die Taste 0 (entspricht SW_A) liefert das ASCII-Zeichen 'A' beim Drücken und 'a' beim Loslassen. Die genaue Belegung ist dem Schema in Abbildung 2 (S.4) zu entnehmen.

Damit die Tastatureingänge in einer View ausgewertet werden können, muss mit **HotKey.InstallLocalKey()** bzw. zuerst ein HotKey installiert werden. Wenn die HotKey installiert sind, werden mit **HotKey.InputFlush()** allenfalls noch gespeicherte Tasten gelöscht. Danach muss die HotKeyTable aktiviert werden mit **HotKey.TableEnable()**. Die HotKeys können aber auch auf einer View dynamisch installiert, neu installiert und deinstalliert werden. Die HotKey können View local oder global installiert werden. Für die globalen Key muss unbedingt auch ein globaler EventHandler vorhanden sein!

```
HotKey.InstallLocalKey(VarInt:Key, labelRelative24:Event, VarInt:Tag)
```

Mit dem Befehl **HotKey.InstallLocalKey(Key, Event, Tag)** wird ein HotKey lokal installiert. Dabei wird der Methode der Bytewert oder ein ASCII-Zeichen und das Label des dazugehörigen EventHandlers mitgegeben. Zu beachten ist, dass der Wert für den Key im Bereich von 0x00..0x7F sein muss. Andere Werte werden mit einer Maske auf diesen Wertebereich gemappt. Das Tag ist ein beliebiger 16-Bit Wert, der im Register **eI.HK_Tag** zurückgeliefert wird, wenn der Event ausgelöst wird. Dieser Wert kann ideal dazu

verwendet werden, die vorgegebenen KeyCodes in spezifische Tastencodes umzuwandeln. Wenn der Key installiert wird, ist er enabled.

HotKey.DeInstallKey(VarInt:Key)

Mit dem Befehl **HotKey.DeInstallKey(Key)** wird ein einzelner HotKey deinstalliert. Dabei wird der Methode der KeyCode (ein ASCII-Zeichen) übergeben. Der nächste Tastendruck auf die deinstallierte Taste bewirkt nichts mehr. Der Befehl kann für lokale und globale HotKeys angewandt werden.

HotKey.DeInstallLocalKeys()

Mit dem Befehl **HotKey.DeInstallLocalKeys()** werden alle lokalen HotKeys deinstalliert. Dieser Befehl wird bei einem Viewwechsel automatisch ausgeführt, da die lokalen Events auf der neuen View nicht mehr gültig sind.

HotKey.DisableLocalKeys()

Mit dem Befehl **HotKey.DisableLocalKeys()** werden alle local Keys in der HotKeyTabelle gesperrt. Die globalen Keys können weiterhin verarbeitet werden.

HotKey.EnableLocalKeys()

Mit dem Befehl **HotKey.EnableLocalKeys()** werden alle local Keys in der HotKeyTabelle freigegeben.

HotKey.InputFlush()

Mit dem Befehl **HotKey.InputFlush()** wird der Eingangspuffer gelöscht. Dieser Befehl wird angewandt, bevor die HotKeyTable freigegeben wird, damit keine Tastendrucke gespeichert sind.

HotKey.InputSelect(i)

Mit dem Befehl **HotKey.InputSelect(i)** wird der Eingangskanal gewählt. Auf dem FOXS wird immer der Treiber aktiviert, der über die Tastaturschnittstelle die Tasten abfragt. Auf dem FOXS muss diese Methode nicht unbedingt aufgerufen werden.

HotKey.TableEnable()

Mit der Methode **HotKey.TableEnable()** wird die HotKeyTabelle freigegeben. Ab der Freigabe sind die Tasten aktiv und können von den entsprechend installierten EventHandlern ausgewertet werden.

HotKey.TableDisable()

Mit der Methode **HotKey.TableDisable()** wird die HotKeyTabelle gesperrt. Dadurch werden keine HotKeys mehr ausgewertet. Die Tabelleneinträge sind

noch vorhanden und durch den Befehl `HotKey.TableEnable()` kann die Tabelle wieder eingeschaltet werden. Dieser Befehl kann dazu verwendet werden, Tasteneingaben vorübergehend zu sperren.

HotKey.TableInit()

Mit der Methode `HotKey.TableInit()` wird die HotKeyTabelle komplett initialisiert und gelöscht. Sowol die lokalen als auch die globalen Eventhandler-Einträge werden aus der Tabelle entfernt.

Anwendungsbeispiele für HotKey-Methoden

Beispiel-Code 1: Installation der HotKeys im Hauptprogramm oder in einer Subroutine.

```
; HotKeys installieren -----
HotKey.InputFlush()
HotKey.InstallLocalKey("D",HK_Red,0)
HotKey.InstallLocalKey("E",HK_Green,0)
HotKey.InstallLocalKey("F",HK_Blue,0)

HotKey.EnableLocalKeys()
HotKey.TableEnable()
```

Erklärung zu Beispiel-Code 1: An den FOX angeschlossene Tasten sollen ausgewertet werden. Die Tasteneingänge D,E,F sollen die Events HK_Red, HK_Green und HK_Blue auslösen. Dazu werden allfällig vorhandene, alte Tasten gelöscht, anschliessend die Tasten als lokale Tasten installiert. In diesem Beispiel wird das Tag nicht gebraucht, deshalb wird 0 geladen.



Wichtig: Nach der Installation der HotKeys müssen im Hauptprogramm die Methoden `HotKey.EnableLocalKeys()` und `HotKey.TableEnable()` folgen, damit das System auf die Tasten-Events reagieren kann. Auch das `LOOP` `ENDLOOP` am Ende des Hauptprogramms darf nicht fehlen.

Beispiel-Code 2: Installation eines HotKeys

```
HotKey.InputFlush() ; Eingabepuffer leeren
HotKey.InstallLocalKey("A", HK_Test_A,blue)
HotKey.InstallLocalKey("a", HK_Test_A,eI.DisplayColor)
```

Beispiel-Code 3: Subroutine, die beim Drücken und Loslassen der Taste A aufgerufen wird.

```
SUB HK_Test_A
  Load.Geometry_XYWH(100,30,80,80)
  eI.FillColor := eI.HK_Tag
  Draw.RectangleFilled()
ENDSUB
```

Erklärung zu Beispiel-Code 2 und Beispiel-Code 3: Im Beispiel-Code 2 wird die Taste A (Pin SW_A) installiert. Bei Tastendruck wird die Subroutine `HK_Test_A` (Beispiel-Code 3) aufgerufen und das Register `eI.HK_Tag` mit der Farbkonstante „blue“ geladen. Die aufgerufene Subroutine zeichnet ein blaues Rechteck. Beim Loslassen der Taste wird die Subroutine `HK_Test_A` erneut aufgerufen, wobei ihr nun die aktuelle Hintergrundfarbe des Displays, gespeichert im Register `eI.DisplayColor`, mitgegeben wird. Damit wird das blaue Rechteck überschrieben, bzw. aus Sicht des Benutzers „gelöscht“.

eigerScript-Programm für eine Tasten-Applikation

Als konkrete Anwendung einer Tasten-Applikation ist beispielsweise die Bedienung eines Billet-Automaten denkbar. Das Programm-Beispiel zeigt, wie die Tasten in eigerScript programmiert werden.

Anordnung und Belegung der Tasten

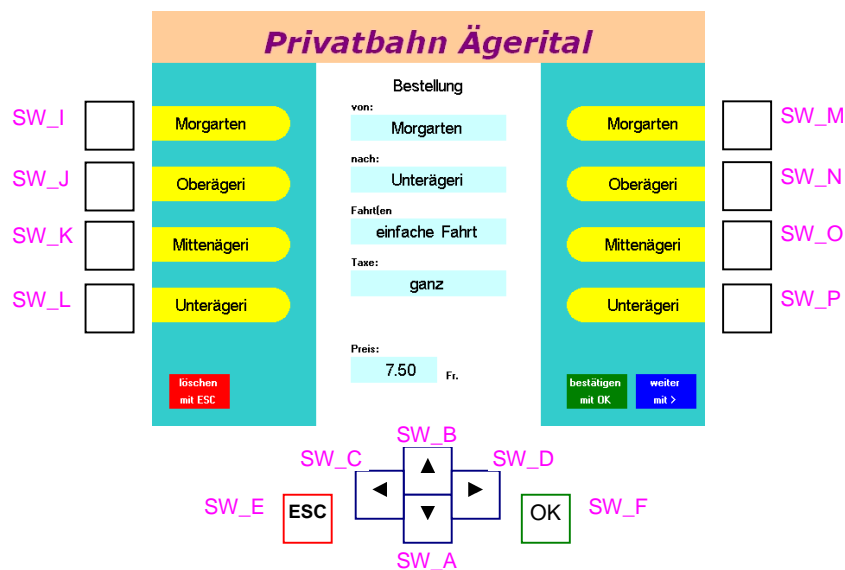


Abbildung 3: Benutzer-Interface der Beispiel-Applikation, das mit 14 Tasten bedient werden kann. Die SW #-Bezeichnungen neben den Tasten entsprechen der Pin-Identifikation von Abbildung 2.

Für unsere Beispiel-Applikation ordnen wir jeder Taste einen 16-Bit Wert zu (vgl. Tabelle 1). Bei Tastendruck wird jeweils der entsprechende Tastenwert ins Register `eI.HK_Tag` geliefert.

Tabelle 1: Werte für die Tasten

Taste	PIN	16-Bit Wert
A	SW_A	0x01
B	SW_B	0x02
C	SW_C	0x03
D	SW_D	0x04
E	SW_E	0x05
F	SW_F	0x06
I	SW_I	0x09
J	SW_J	0x0A
K	SW_K	0x0B
L	SW_L	0x0C
M	SW_M	0x0D
N	SW_N	0x0E
O	SW_O	0x0F
P	SW_P	0x11

Screenshots der Tasten-Applikation

Die folgenden Screenshots unseres Beispiels sind eine 1:1 Abbildung des eigerPanel-Displays. Sie wurden mit Hilfe der Methode `File.ScreenDump` generiert .

Beispiel-Code 4: Installation des HotSpot im Hauptprogramm. Bei Berührung der Touchscreen wird die Subroutine „ScreenDump“ aufgerufen.

```
; Hotspot rechts oben für Screenshot -----  
eI.R01 := 0  
Load.Geometry_XYWH(600,0,40,40) ; Dimension und Position des  
HotSpots  
HotSpot.Install(NIL,NIL,ScreenDump,NIL)  
  
Display.Show()  
  
HotSpot.TableEnable()
```

Beispiel-Code 5: Die Subroutine generiert ein Screenshot und legt das Bild im EGI-Format auf der CompactFlash Card ab. Die IF-Schleife dient hier dazu, gleich mehrere Screenshots aufzunehmen.

```
SUB ScreenDump ; Subroutine des HotSpots für Screenshots der  
Layouts A,B,C  
eI.R01 := eI.R01 + 1  
IF eI.R01 == 1 THEN  
File.ScreenDump('C:\\KEYS\\DUMP\\LAYOUT_A.EGI')  
ELSIF eI.R01 == 2 THEN  
File.ScreenDump('C:\\KEYS\\DUMP\\LAYOUT_B.EGI')  
ELSIF eI.R01 == 3 THEN  
File.ScreenDump('C:\\KEYS\\DUMP\\LAYOUT_C.EGI')  
eI.R01 := 0  
ENDIF  
ENDSUB
```

Privatbahn Ägerital

Morgarten

Oberägeri

Mittenägeri

Unterägeri

löschen
mit ESC

Bestellung

von: Morgarten

nach: Unterägeri

Fahrt(en) einfache Fahrt

Taxe: ganz

Preis: 7.50 Fr.

Morgarten

Oberägeri

Mittenägeri

Unterägeri

bestätigen
mit OK

weiter
mit >

Abbildung 4: Layout A der Tastenapplikation. Mit den Tasten können Start- und Zielort gewählt werden. Zudem steht eine Lösch- sowie eine Bestätigungstaste zur Verfügung. Mit der Weitertaste geht es zum Layout B.

Privatbahn Ägerital

einfache Fahrt

hin und zurück

6er Abo

löschen
mit ESC

Bestellung

von: Morgarten

nach: Unterägeri

Fahrt(en) einfache Fahrt

Taxe: 1/2

Preis: 3.75 Fr.

ganz

1/2

Velo

bestätigen
mit OK

zurück
mit <

Abbildung 5: In Layout B besteht die Möglichkeit, mittels je 3 Tasten die Billet-Art sowie die Tarifklasse zu wählen.



Abbildung 6: Layout C erscheint, wenn die Bestellung mit der Bestätigungstaste abgeschlossen wird.

eigerScript Programm-Code für Tasten-Applikation

Beispiel-Code 6: Programmcode für die Beispielsapplikation „KEYS“.

```

;-----
; Titel      : Beispiel für Key-Application   View 1
; File       : KEYS_001.EVS
;-----
; Compiler   : eigerScript
;
; System     : eigergraphics.com; FOXS embedded computer
;
; Beschreibung: Application Note "programmieren von Tasten (HotKeys)
;             Die Applikation ist eine Benutzerführung an einem
;             Billetautomaten der fiktiven "Privatbahn Ägerital"
;
; Version    : Initialversion: 18.09.2007
;
; Autor      : Christoph Angst
;
;-----
; (c) 2005-2007      S-TEC electronics AG, CH-6300 Zug; 041 754 50 10
; www.eigergraphics.com
;-----
;
; Skizze der Benutzeroberfläche mit den Tasten (HotKeys)
;
; -----
; | I |          PRIVATBAHN ÄGERITAL          | M |
; -----
; | J |          |          |          |          | N |
; -----
; | K |          |          |          |          | O |
; -----
; | L |          |          |          |          | P |
; -----
;
; -----
; |ESC| | < | | > | |OK| |
; -----
;-----

EIGERPROJECT 'KEYS'      ; Projektbezeichnung: erster Teil des EVI-
Dateinamens
EIGERVIEW 1             ; Viewnummer: zweiter Teil des EVI-Dateinamens:
CHSS_001.EVI

IMPORT                  'DEF_eVM_OpCodes.h'      ; Token
IMPORT                  'DEF_eVM_Registers.h'    ; Register
FUNCLIB                  'DEF_eVM_Functions.lib' ; Funktionsbibliothek

INCLUDEFILE              'DEF_eiger_Colors.INC' ; Farbdefinitionen 12.03.2006
INCLUDEFILE              'DEF_eiger_Types.INC'  ; eiger Definitionen

```

```

; D E K L A R A T I O N E N -----
; Definitionen für Startview -----

STRING [60] Titel.$ = 'Key-Application, KEYS_001.EVS: Startview'
STRING [60] Titel_Right.$ = 'erstellt: 18.09.2007' ;
Titelleiste rechts

; Button-Strings -----

; für Layout_A
STRING [15] KeyButton_left1_A.$ = 'Morgarten'
STRING [15] KeyButton_left2_A.$ = 'Oberägeri'
STRING [15] KeyButton_left3_A.$ = 'Mittenägeri'
STRING [15] KeyButton_left4_A.$ = 'Unterägeri'

STRING [15] KeyButton_right1_A.$ = 'Morgarten'
STRING [15] KeyButton_right2_A.$ = 'Oberägeri'
STRING [15] KeyButton_right3_A.$ = 'Mittenägeri'
STRING [15] KeyButton_right4_A.$ = 'Unterägeri'

; für Layout_B -----
STRING [15] KeyButton_left1_B.$ = 'einfache Fahrt'
STRING [15] KeyButton_left2_B.$ = 'hin und zurück'
STRING [15] KeyButton_left3_B.$ = '6er Abo'

STRING [15] KeyButton_right1_B.$ = 'ganz'
STRING [15] KeyButton_right2_B.$ = '1/2'
STRING [15] KeyButton_right3_B.$ = 'Velo'

; für Navigier-Buttons -----
STRING [15] DelButton1.$ = 'löschen'
STRING [15] DelButton2.$ = 'mit ESC'
STRING [15] NextButton1.$ = 'weiter'
STRING [15] NextButton2.$ = 'mit >'
STRING [15] BackButton1.$ = 'zurück'
STRING [15] BackButton2.$ = 'mit <'
STRING [15] OKButton1.$ = 'bestätigen'
STRING [15] OKButton2.$ = 'mit OK'

; Überschriften für Center-Felder -----
STRING [15] CenterTitel.$ = 'Bestellung'
STRING [8] CenterField1_Titel.$ = 'von:'
STRING [8] CenterField2_Titel.$ = 'nach:'
STRING [8] CenterField3_Titel.$ = 'Fahrt(en):'
STRING [8] CenterField4_Titel.$ = 'Taxe:'
STRING [8] PriceField1_Titel.$ = 'Preis:'

; Centerfeld-Strings -----
STRING [15] Start.$ = ''
STRING [15] Ziel.$ = ''
STRING [15] Fahrten.$ = ''
STRING [16] Taxe.$ = ''
STRING [8] Preis.$ = ''

; Strings für Bestätigung -----
STRING [50] Zusammenfassung_1.$
STRING [50] Zusammenfassung_2.$
STRING [50] Zusammenfassung_3.$

; Schaltervariablen -----
INTEGER Layoutschalter.I = 1 ; 1 = Layout_A; 2 = Layout_B; 3 = Popu

```

```

; Variablen für Preisberechnung -----
INTEGER Startort.I = 0
INTEGER Zielort.I = 0
INTEGER Fahrten.I = 0
INTEGER Taxe.I = 0 ; bei Halbtax = 0.5
INTEGER Streckenlaenge.I = 0
INTEGER Hilfsvar.I = 0 ; erst wenn Hilfsvar > 0 kann der Preis
berechnet werden
INTEGER Preis.I
INTEGER Einheitspreis.I = 250 ; in Rappen

; Layout-Konstanten -----

CONST WING_W = 190 ; Breite der Seitenbalken links und rechts
CONST WING_H = 420 ; Höhe der Seitenbalken links und rechts
CONST WING_Y = 60
CONST WINGLEFT_X = 0
CONST WINGRIGHT_X = 640 - WING_W

CONST KEYBUTTON_W = 140 ; Breite der gelben Buttons
CONST KEYBUTTON_H = 40 ; Höhe der gelben Buttons
CONST KEYBUTTON_LEFT_X = 0 ; Button-Abstand vom linken Rand
CONST KEYBUTTON_RIGHT_X = 640 - KEYBUTTON_LEFT_X - KEYBUTTON_W
CONST KEYBUTTON_GRID_Y = KEYBUTTON_H + 30
CONST KEYBUTTON_ROW1_Y = 110 ; Abstand des 1.Buttons vom oberen Rand
CONST KEYBUTTON_ROW2_Y = KEYBUTTON_ROW1_Y + KEYBUTTON_GRID_Y
CONST KEYBUTTON_ROW3_Y = KEYBUTTON_ROW1_Y + 2*KEYBUTTON_GRID_Y
CONST KEYBUTTON_ROW4_Y = KEYBUTTON_ROW1_Y + 3*KEYBUTTON_GRID_Y

CONST NAVBUTTON_W = 70
CONST NAVBUTTON_H = 40
CONST NAVBUTTON_HALBE_H = 0.5*NAVBUTTON_H
CONST NAVBUTTON_LEFT_X = 0.5*KEYBUTTON_H
CONST NAVBUTTON_RIGHT1_X = KEYBUTTON_RIGHT_X - 0.5*KEYBUTTON_H
CONST NAVBUTTON_RIGHT2_X = NAVBUTTON_RIGHT1_X + 80
CONST NAVBUTTON_Y = 420
CONST ZEILE1_Y = NAVBUTTON_Y
CONST ZEILE2_Y = NAVBUTTON_Y + NAVBUTTON_HALBE_H

CONST CENTERFIELD_W = 180
CONST CENTERFIELD_H = 30
CONST CENTERFIELD_X = 230
CONST CENTERFIELD1_Y = 120
CONST CENTERFIELD2_Y = 180
CONST CENTERFIELD3_Y = 240
CONST CENTERFIELD4_Y = 300

CONST PRICEFIELD1_W = 100
CONST PRICEFIELD2_W = 70
CONST PRICEFIELD_H = 30
CONST PRICEFIELD_Y = 400
CONST PRICEFIELD1_X = CENTERFIELD_X
CONST PRICEFIELD2_X = PRICEFIELD1_X + PRICEFIELD1_W + 10

CONST TITELBALKEN_W = 640
CONST TITELBALKEN_H = 60
CONST TITELBALKEN_X = 0
CONST TITELBALKEN_Y = 0

CONST POPUP_W = 500
CONST POPUP_H = 300
CONST POPUP_X = 320 - 0.5*POPUP_W
CONST POPUP_Y = 240 - 0.5*POPUP_H

```

```

; S U B R O U T I N E N -----
SUB Draw_Titel
; Titelleiste schreiben -----

Fill.LabelParameter(Titel_Style)
Label.Text(Titel.$)

Fill.LabelParameter(Titel_Right_Style)
Label.Text(Titel_Right.$)
ENDSUB

SUB Titelbalken ; mit Überschrift 'Privatbahn Ägerital'
Load.Geometry_XYWH(TITELBALKEN_X,TITELBALKEN_Y,TITELBALKEN_W,TIT
ELBALKEN_H)
Colors.Load_RGB(eI.FillColor,255,204,153)
Draw.RectangleFilled()
Load.Pos_X1Y1(123,14) ; linke obere Ecke für Privatbahn-TITEL
File.Read_EGI('C:\KEYS\PICT\TITEL.EGI'); Bildgrösse: 394x42
Pixel
ENDSUB

; _____Beginn Layout A:_____

SUB Draw_Layout_A

Layoutschalter.I := 1

;linker Seitenbalken -----
Load.Geometry_XYWH(WINGLEFT_X,WING_Y,WING_W,WING_H)
Colors.Load_RGB(eI.FillColor,51,204,204) ; Türkisfarbe
entsprechend der Excel-Vorlage
Draw.RectangleFilled()

;rechter Seitenbalken -----
Load.Geometry_XYWH(WINGRIGHT_X,WING_Y,WING_W,WING_H)
Colors.Load_RGB(eI.FillColor,51,204,204)
Draw.RectangleFilled()

;Linke Buttons -----
Load.Geometry_XYWH(KEYBUTTON_LEFT_X,KEYBUTTON_ROW1_Y,KEYBUTTON_W
,KEYBUTTON_H)
eI.FillColor := yellow
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW2_Y
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW3_Y
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW4_Y
Draw.RectangleFilled()

; Rundung Button links -----
eI.Pos_X1 := KEYBUTTON_LEFT_X + KEYBUTTON_W

eI.Pos_Y1 := KEYBUTTON_ROW1_Y + 0.5*KEYBUTTON_H
eI.Radius := 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW2_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW3_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

```

```

eI.Pos_Y1 := KEYBUTTON_ROW4_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

;rechte Buttons -----
Load.Geometry_XYWH(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW1_Y,KEYBUTTON_
W,KEYBUTTON_H)
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW2_Y
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW3_Y
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW4_Y
Draw.RectangleFilled()

; Rundung Button rechts -----
eI.Pos_X1 := KEYBUTTON_RIGHT_X

eI.Pos_Y1 := KEYBUTTON_ROW1_Y + 0.5*KEYBUTTON_H
eI.Radius := 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW2_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW3_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW4_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

;Navigier-Buttons -----
Load.Geometry_XYWH(NAVBUTTON_LEFT_X,NAVBUTTON_Y,NAVBUTTON_W,NAV
UTTON_H)
eI.FillColor := red
Draw.RectangleFilled()

Load.Geometry_XYWH(NAVBUTTON_RIGHT1_X,NAVBUTTON_Y,NAVBUTTON_W,NA
VBUTTON_H)
eI.FillColor := green
Draw.RectangleFilled()

Load.Geometry_XYWH(NAVBUTTON_RIGHT2_X,NAVBUTTON_Y,NAVBUTTON_W,NA
VBUTTON_H)
eI.FillColor := blue
Draw.RectangleFilled()

CallSubroutine(Beschriftung_Layout_A)
ENDSUB

SUB Beschriftung_Layout_A
; Start-Ortschaften -----
Fill.LabelParameter(Keybutton_Style)
Load.Pos_X1Y1(KEYBUTTON_LEFT_X,KEYBUTTON_ROW1_Y)
Label.Text(KeyButton_left1_A.$)

Load.Pos_X1Y1(KEYBUTTON_LEFT_X,KEYBUTTON_ROW2_Y)
Label.Text(KeyButton_left2_A.$)

Load.Pos_X1Y1(KEYBUTTON_LEFT_X,KEYBUTTON_ROW3_Y)
Label.Text(KeyButton_left3_A.$)

Load.Pos_X1Y1(KEYBUTTON_LEFT_X,KEYBUTTON_ROW4_Y)
Label.Text(KeyButton_left4_A.$)

```

```

; Ziel-Ortschaften -----
Fill.LabelParameter(Keybutton_Style)
Load.Pos_X1Y1(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW1_Y)
Label.Text(KeyButton_right1_A.$)

Load.Pos_X1Y1(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW2_Y)
Label.Text(KeyButton_right2_A.$)

Load.Pos_X1Y1(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW3_Y)
Label.Text(KeyButton_right3_A.$)

Load.Pos_X1Y1(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW4_Y)
Label.Text(KeyButton_right4_A.$)

; Navigier-Buttons -----
Fill.LabelParameter(Keybutton_Style)
eI.TextColor := white
eI.FontNumber := Font_System_9bd

Load.Geometry_XYWH(NAVBUTTON_LEFT_X,ZEILE1_Y,NAVBUTTON_W,NAVBU
ON_HALBE_H)
Label.Text(DelButton1.$)
Load.Geometry_XYWH(NAVBUTTON_LEFT_X,ZEILE2_Y,NAVBUTTON_W,NAVBU
ON_HALBE_H)
Label.Text(DelButton2.$)

Load.Geometry_XYWH(NAVBUTTON_RIGHT1_X,ZEILE1_Y,NAVBUTTON_W,NAVBU
TTON_HALBE_H)
Label.Text(OKButton1.$)
Load.Geometry_XYWH(NAVBUTTON_RIGHT1_X,ZEILE2_Y,NAVBUTTON_W,NAVBU
TTON_HALBE_H)
Label.Text(OKButton2.$)

Load.Geometry_XYWH(NAVBUTTON_RIGHT2_X,ZEILE1_Y,NAVBUTTON_W,NAVBU
TTON_HALBE_H)
Label.Text(NextButton1.$)
Load.Geometry_XYWH(NAVBUTTON_RIGHT2_X,ZEILE2_Y,NAVBUTTON_W,NAVBU
TTON_HALBE_H)
Label.Text(NextButton2.$)
ENDSUB
; _____ Ende Layout A _____

; _____ Beginn Layout B: _____

SUB Draw_Layout_B

Layoutschalter.I := 2

;linker Seitenbalken -----
Load.Geometry_XYWH(WINGLEFT_X,WING_Y,WING_W,WING_H)
Colors.Load_RGB(eI.FillColor,51,204,204) ; Türkisfarbe
entsprechend der Excel-Vorlage
Draw.RectangleFilled()

;rechter Seitenbalken -----
Load.Geometry_XYWH(WINGRIGHT_X,WING_Y,WING_W,WING_H)
Colors.Load_RGB(eI.FillColor,51,204,204)
Draw.RectangleFilled()

;Linke Buttons -----
Load.Geometry_XYWH(KEYBUTTON_LEFT_X,KEYBUTTON_ROW1_Y,KEYBUTTON_W
,KEYBUTTON_H)
eI.FillColor := yellow
Draw.RectangleFilled()

```

```

eI.Pos_Y1 := KEYBUTTON_ROW2_Y
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW3_Y
Draw.RectangleFilled()

; eI.Pos_Y1 := KEYBUTTON_ROW4_Y
; Draw.RectangleFilled()

; Rundung Button links -----
eI.Pos_X1 := KEYBUTTON_LEFT_X + KEYBUTTON_W

eI.Pos_Y1 := KEYBUTTON_ROW1_Y + 0.5*KEYBUTTON_H
eI.Radius := 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW2_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW3_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

;rechte Buttons -----
Load.Geometry_XYWH(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW1_Y,KEYBUTTON_
W,KEYBUTTON_H)
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW2_Y
Draw.RectangleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW3_Y
Draw.RectangleFilled()

; eI.Pos_Y1 := KEYBUTTON_ROW4_Y
; Draw.RectangleFilled()

; Rundung Button rechts -----
eI.Pos_X1 := KEYBUTTON_RIGHT_X

eI.Pos_Y1 := KEYBUTTON_ROW1_Y + 0.5*KEYBUTTON_H
eI.Radius := 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW2_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

eI.Pos_Y1 := KEYBUTTON_ROW3_Y + 0.5*KEYBUTTON_H
Draw.CircleFilled()

;Navigier-Buttons -----
Load.Geometry_XYWH(NAVBUTTON_LEFT_X,NAVBUTTON_Y,NAVBUTTON_W,NAVB
UTTON_H)
eI.FillColor := red
Draw.RectangleFilled()

Load.Geometry_XYWH(NAVBUTTON_RIGHT1_X,NAVBUTTON_Y,NAVBUTTON_W,NA
VBUTTON_H)
eI.FillColor := green
Draw.RectangleFilled()

Load.Geometry_XYWH(NAVBUTTON_RIGHT2_X,NAVBUTTON_Y,NAVBUTTON_W,NA
VBUTTON_H)
eI.FillColor := blue
Draw.RectangleFilled()

CallSubroutine(Beschriftung_Layout_B)
ENDSUB

```

```

SUB Beschriftung_Layout_B
; Fahrten-Auswahl -----
Fill.LabelParameter(Keybutton_Style)
Load.Pos_X1Y1(KEYBUTTON_LEFT_X,KEYBUTTON_ROW1_Y)
Label.Text(KeyButton_left1_B.$)

Load.Pos_X1Y1(KEYBUTTON_LEFT_X,KEYBUTTON_ROW2_Y)
Label.Text(KeyButton_left2_B.$)

Load.Pos_X1Y1(KEYBUTTON_LEFT_X,KEYBUTTON_ROW3_Y)
Label.Text(KeyButton_left3_B.$)

; Load.Pos_X1Y1(KEYBUTTON_LEFT_X,KEYBUTTON_ROW4_Y)
; Label.Text(KeyButton_left4_B.$)
;

; Taxe-Auswahl -----
Fill.LabelParameter(Keybutton_Style)
Load.Pos_X1Y1(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW1_Y)
Label.Text(KeyButton_right1_B.$)

Load.Pos_X1Y1(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW2_Y)
Label.Text(KeyButton_right2_B.$)

Load.Pos_X1Y1(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW3_Y)
Label.Text(KeyButton_right3_B.$)

; Load.Pos_X1Y1(KEYBUTTON_RIGHT_X,KEYBUTTON_ROW4_Y)
; Label.Text(KeyButton_right4_B.$)

; Navigier-Buttons -----
Fill.LabelParameter(Keybutton_Style)
eI.TextColor := white
eI.FontNumber := Font_System_9bd

Load.Geometry_XYWH(NAVBUTTON_LEFT_X,ZEILE1_Y,NAVBUTTON_W,NAVBU
ON_HALBE_H)
Label.Text(DelButton1.$)
Load.Geometry_XYWH(NAVBUTTON_LEFT_X,ZEILE2_Y,NAVBUTTON_W,NAVBU
ON_HALBE_H)
Label.Text(DelButton2.$)

Load.Geometry_XYWH(NAVBUTTON_RIGHT1_X,ZEILE1_Y,NAVBUTTON_W,NAVBU
TTON_HALBE_H)
Label.Text(OKButton1.$)
Load.Geometry_XYWH(NAVBUTTON_RIGHT1_X,ZEILE2_Y,NAVBUTTON_W,NAVBU
TTON_HALBE_H)
Label.Text(OKButton2.$)

Load.Geometry_XYWH(NAVBUTTON_RIGHT2_X,ZEILE1_Y,NAVBUTTON_W,NAVBU
TTON_HALBE_H)
Label.Text(BackButton1.$)
Load.Geometry_XYWH(NAVBUTTON_RIGHT2_X,ZEILE2_Y,NAVBUTTON_W,NAVBU
TTON_HALBE_H)
Label.Text(BackButton2.$)

ENDSUB

; _____ Ende Layout B _____

; _____ Beginn Layout Center: _____

SUB Draw_Center_Layout
; Einfüll-Felder
Load.Geometry_XYWH(CENTERFIELD_X,CENTERFIELD1_Y,CENTERFIELD_W,CE
NTERFIELD_H)
Colors.Load_RGB(eI.FillColor,204,255,255)
Draw.RectangleFilled()

```

```

eI.Pos_Y1 := CENTERFIELD2_Y
Draw.RectangleFilled()

eI.Pos_Y1 := CENTERFIELD3_Y
Draw.RectangleFilled()

eI.Pos_Y1 := CENTERFIELD4_Y
Draw.RectangleFilled()

Load.Geometry_XYWH(PRICEFIELD1_X,PRICEFIELD_Y,PRICEFIELD1_W,PRICEFIELD_H)
Draw.RectangleFilled()

; Feld-Beschriftung -----
Fill.LabelParameter(Centerfield_Titel_Style)
Load.Geometry_XYWH(CENTERFIELD_X,70,CENTERFIELD_W,CENTERFIELD_H)
eI.Position := Pos_center
eI.FontNumber := Font_Arial_14n
Label.Text(CenterTitel.$)

Fill.LabelParameter(Centerfield_Titel_Style)
eI.Pos_X1 := CENTERFIELD_X
eI.Pos_Y1 := CENTERFIELD1_Y - 20
Label.Text(CenterField1_Titel.$)

eI.Pos_Y1 := CENTERFIELD2_Y - 20
Label.Text(CenterField2_Titel.$)

eI.Pos_Y1 := CENTERFIELD3_Y - 20
Label.Text(CenterField3_Titel.$)

eI.Pos_Y1 := CENTERFIELD4_Y - 20
Label.Text(CenterField4_Titel.$)

eI.Pos_Y1 := PRICEFIELD_Y - 20
Label.Text(PriceField1_Titel.$)

Load.Geometry_XYWH(PRICEFIELD2_X,PRICEFIELD_Y,PRICEFIELD2_W,PRICEFIELD_H)
eI.Position := Pos_bottomleft
Label.Text('Fr.')
ENDSUB

SUB Felder_ausfuellen
IF Layoutschalter.I == 1 THEN
; von-Feld
IF eI.HK_Tag == 0x09 THEN
Start.$ := KeyButton_left1_A.$
Startort.I := 1
ELSIF eI.HK_Tag == 0x0A THEN
Start.$ := KeyButton_left2_A.$
Startort.I := 2
ELSIF eI.HK_Tag == 0x0B THEN
Start.$ := KeyButton_left3_A.$
Startort.I := 3
ELSIF eI.HK_Tag == 0x0C THEN
Start.$ := KeyButton_left4_A.$
Startort.I := 4
ENDIF

Fill.LabelParameter(Keybutton_Style)
Load.Geometry_XYWH(CENTERFIELD_X,CENTERFIELD1_Y,CENTERFIELD_W,CENTERFIELD_H)
Colors.Load_RGB(eI.FillColor,204,255,255)
Draw.RectangleFilled()
Label.Text(Start.$)

```

```

; nach-Feld -----
IF eI.HK_Tag == 0x0D THEN
  Ziel.$ := KeyButton_right1_A.$
  Zielort.I := 1
ELSIF eI.HK_Tag == 0x0E THEN
  Ziel.$ := KeyButton_right2_A.$
  Zielort.I := 2
ELSIF eI.HK_Tag == 0x0F THEN
  Ziel.$ := KeyButton_right3_A.$
  Zielort.I := 3
ELSIF eI.HK_Tag == 0x11 THEN
  Ziel.$ := KeyButton_right4_A.$
  Zielort.I := 4
ENDIF

Fill.LabelParameter(Keybutton_Style)
Load.Geometry_XYWH(CENTERFIELD_X,CENTERFIELD2_Y,CENTERFIELD_W,CEN
NTERFIELD_H)
Colors.Load_RGB(eI.FillColor,204,255,255)
Draw.RectangleFilled()
Label.Text(Ziel.$)

; Defaulteintrag für Fahrten- und Taxe-Feld
IF Startort.I > 0 THEN
  IF Zielort.I > 0 THEN
    IF Startort.I <> Zielort.I THEN
      Fahrten.I := 1
      Fahrten.$ := KeyButton_left1_B.$

      Taxe.I := 1
      Taxe.$ := KeyButton_right1_B.$

      Fill.LabelParameter(Keybutton_Style)

Load.Geometry_XYWH(CENTERFIELD_X,CENTERFIELD3_Y,CENTERFIELD_W,CENTER
FIELD_H)
      Colors.Load_RGB(eI.FillColor,204,255,255)
      Draw.RectangleFilled()
      Label.Text(Fahrten.$)

Load.Geometry_XYWH(CENTERFIELD_X,CENTERFIELD4_Y,CENTERFIELD_W,CENTER
FIELD_H)
      Colors.Load_RGB(eI.FillColor,204,255,255)
      Draw.RectangleFilled()
      Label.Text(Taxe.$)
    ENDIF
  ENDIF
ENDIF

ELSIF Layoutschalter.I == 2 THEN
; Fahrten-Feld
IF eI.HK_Tag == 0x09 THEN
  Fahrten.$ := KeyButton_left1_B.$
  Fahrten.I := 1
ELSIF eI.HK_Tag == 0x0A THEN
  Fahrten.$ := KeyButton_left2_B.$
  Fahrten.I := 2
ELSIF eI.HK_Tag == 0x0B THEN
  Fahrten.$ := KeyButton_left3_B.$
  Fahrten.I := 6
; ELSIF eI.HK_Tag == 0x0C THEN
; Fahrten.$ := KeyButton_left4_B.$
ENDIF

Fill.LabelParameter(Keybutton_Style)
Load.Geometry_XYWH(CENTERFIELD_X,CENTERFIELD3_Y,CENTERFIELD_W,CEN
NTERFIELD_H)
Colors.Load_RGB(eI.FillColor,204,255,255)

```

```

Draw.RectangleFilled()
Label.Text(Fahrten.$)

; Taxe-Feld -----
IF eI.HK_Tag == 0x0D THEN
  Taxe.$ := KeyButton_right1_B.$
  Taxe.I := 1
ELSIF eI.HK_Tag == 0x0E THEN
  Taxe.$ := KeyButton_right2_B.$
  Taxe.I := 2 ; d.h. Preis wird durch 2 dividiert
ELSIF eI.HK_Tag == 0x0F THEN
  Taxe.$ := KeyButton_right3_B.$
  Taxe.I := 2 ; d.h. Preis wird durch 2 dividiert
; ELSIF eI.HK_Tag == 0x11 THEN
;   Taxe.$ := KeyButton_right4_B.$
ENDIF

Fill.LabelParameter(Keybutton_Style)
Load.Geometry_XYWH(CENTERFIELD_X,CENTERFIELD4_Y,CENTERFIELD_W,CE
NTERFIELD_H)
Colors.Load_RGB(eI.FillColor,204,255,255)
Draw.RectangleFilled()
Label.Text(Taxe.$)

ENDIF

; Preis-Feld -----
CallSubroutine(Preisberechnung)
ENDSUB

SUB Preisberechnung
  IF Zielort.I > Startort.I THEN ; Streckenlänge darf nicht
negativ sein
    Streckenlaenge.I := Zielort.I - Startort.I
  ELSE
    Streckenlaenge.I := Startort.I - Zielort.I
  ENDIF
  ; Überprüfen, ob Preis berechnet werden kann
  Hilfsvar.I := Startort.I * Zielort.I
  Hilfsvar.I := Hilfsvar.I * Streckenlaenge.I
  Hilfsvar.I := Hilfsvar.I * Fahrten.I
  Hilfsvar.I := Hilfsvar.I * Taxe.I

  IF Hilfsvar.I == 0 THEN
    Preis.$ := ''
  ELSE
    Preis.I := Streckenlaenge.I * Fahrten.I
    Preis.I := Preis.I * Einheitspreis.I ; Preis in Rappen
    Preis.I := Preis.I / Taxe.I
    Preis.$ := ''
    Str.Cvt_IntegerDeci(Preis.$,Preis.I,6,2) ; Preis in Franken
  ENDIF

  Fill.LabelParameter(Keybutton_Style)
  Load.Geometry_XYWH(PRICEFIELD1_X,PRICEFIELD_Y,PRICEFIELD1_W,PRIC
EFIELD_H)
  Colors.Load_RGB(eI.FillColor,204,255,255)
  Draw.RectangleFilled()
  Label.Text(Preis.$)
ENDSUB

; ----- Ende Layout Center -----

; ----- Beginn BestätigungsPopup (Layout C) -----

SUB Popup ; wird eingeblendet bei Bestätigungs-Tastendruck
  IF Layoutschalter.I < 3 THEN; nur auf Bestätigungstaste
reagieren, wenn Popup noch nicht aufgerufen ist

```

```

    IF Preis.I > 0 THEN ; nur auf Bestätigungstaste reagieren, wenn
ein Preis existiert
    eI.Garbage := Layoutschalter.I
    Layoutschalter.I := 3
    Display.Direct

    Str.Concat(Zusammenfassung_1.$,'von ')
    Str.Concat(Zusammenfassung_1.$,Start.$)
    Str.Concat(Zusammenfassung_1.$,' nach ')
    Str.Concat(Zusammenfassung_1.$,Ziel.$)
    Str.Concat(Zusammenfassung_2.$,Taxe.$)
    Str.Concat(Zusammenfassung_2.$,' / ')
    Str.Concat(Zusammenfassung_2.$,Fahrten.$)
    Str.Concat(Zusammenfassung_3.$,'Preis: ')
    Str.Concat(Zusammenfassung_3.$,Preis.$)
    Str.Concat(Zusammenfassung_3.$,' Fr.')
```

```

    Load.Geometry_XYWH(POPUP_X,POPUP_Y,POPUP_W,POPUP_H)
    eI.FillColor := yellowgreen
    Colors.Darken(eI.LineColor,eI.FillColor)
    eI.BorderStyle := colorshadow_4
    Label.Box()
    Fill.LabelParameter(Keybutton_Style)

    Load.Geometry_XYWH(POPUP_X,POPUP_Y,POPUP_W,40)
    eI.Pos_Y1 := POPUP_Y + 20
    Label.Text('Ihre Bestellung:')
    eI.Pos_Y1 := POPUP_Y + 60
    eI.TextColor := white
    Label.Text(Zusammenfassung_1.$)
    eI.Pos_Y1 := POPUP_Y + 100
    Label.Text(Zusammenfassung_2.$)
    eI.Pos_Y1 := POPUP_Y + 140
    Label.Text(Zusammenfassung_3.$)
    eI.Pos_Y1 := POPUP_Y + 200
    eI.TextColor := darkslateblue
    Label.Text('Vielen Dank für Ihre Bestellung und gute Reise')
```

```

    eI.Pos_Y1 := POPUP_Y + 240
    eI.Height := 20
    eI.FillColor := red
    eI.TextColor := white
    eI.FontNumber := Font_System_9bd
    Label.Text('ändern mit ESC')
```

```

    ENDIF
    ENDIF
ENDSUB

; _____Ende BestätigungsPopup_____

SUB Esc
    IF Layoutschalter.I == 3 THEN ; Popup ausblenden
    Layoutschalter.I := eI.Garbage
    Zusammenfassung_1.$ := ''
    Zusammenfassung_2.$ := ''
    Zusammenfassung_3.$ := ''
    Display.Show()

    ELSE ; alles löschen
    Start.$ := ''
    Ziel.$ := ''
    Fahrten.$ := ''
    Taxe.$ := ''
    Startort.I := 0
    Zielort.I := 0
    Fahrten.I := 0
    Taxe.I := 0
    Streckenlaenge.I := 0

```

```

        CallSubroutine(Draw_Center_Layout)
    ENDIF
ENDSUB

SUB ScreenDump ; Subroutine des HotSpots für Screenshots der
Layouts A,B,C
    eI.R01 := eI.R01 + 1
    IF eI.R01 == 1 THEN
        File.ScreenDump('C:\\KEYS\\DUMP\\LAYOUT_A.EGI')
    ELSIF eI.R01 == 2 THEN
        File.ScreenDump('C:\\KEYS\\DUMP\\LAYOUT_B.EGI')
    ELSIF eI.R01 == 3 THEN
        File.ScreenDump('C:\\KEYS\\DUMP\\LAYOUT_C.EGI')
    eI.R01 := 0
    ENDIF

ENDSUB

; _____ Beginn Styles _____

SUB Styles

Titel_Style:
    INLINENWORDS (0) ; entspricht eI.Pos_Xl
    INLINENWORDS (0) ; entspricht eI.Pos_Yl
    INLINENWORDS (640) ; entspricht eI.Width
    INLINENWORDS (18) ; entspricht eI.Height
    INLINENWORDS (20) ; entspricht eI.SpaceLeft
    INLINENWORDS (8) ; entspricht eI.SpaceRight
    INLINENWORDS (0) ; entspricht eI.HorizontalAdjust
    INLINENWORDS (0) ; entspricht eI.VericalAdjust
    INLINENWORDS (transparent) ; entspricht eI.FillColor
    INLINENWORDS (transparent) ; entspricht eI.BackColor
    INLINENWORDS (transparent) ; entspricht eI.LineColor
    INLINENWORDS (lightyellow) ; entspricht eI.TextColor
    INLINENWORDS (Pos_left) ; entspricht eI.Position
    INLINENWORDS (Orientation_0deg) ; entspricht eI.Orientation
    INLINENWORDS (normal) ; entspricht eI.Appearance
    INLINENWORDS (no_border) ; entspricht eI.BorderStyle
    INLINENWORDS (Font_System_9bd) ; entspricht eI.FontNumber
    INLINENWORDS (silver) ; entspricht eI.BackgroundColor

Titel_Right_Style:
    INLINENWORDS (0) ; entspricht eI.Pos_Xl
    INLINENWORDS (0) ; entspricht eI.Pos_Yl
    INLINENWORDS (640) ; entspricht eI.Width
    INLINENWORDS (18) ; entspricht eI.Height
    INLINENWORDS (20) ; entspricht eI.SpaceLeft
    INLINENWORDS (8) ; entspricht eI.SpaceRight
    INLINENWORDS (0) ; entspricht eI.HorizontalAdjust
    INLINENWORDS (0) ; entspricht eI.VericalAdjust
    INLINENWORDS (transparent) ; entspricht eI.FillColor
    INLINENWORDS (transparent) ; entspricht eI.BackColor
    INLINENWORDS (transparent) ; entspricht eI.LineColor
    INLINENWORDS (lightyellow) ; entspricht eI.TextColor
    INLINENWORDS (Pos_right) ; entspricht eI.Position
    INLINENWORDS (Orientation_0deg) ; entspricht eI.Orientation
    INLINENWORDS (normal) ; entspricht eI.Appearance
    INLINENWORDS (no_border) ; entspricht eI.BorderStyle
    INLINENWORDS (Font_System_9bd) ; entspricht eI.FontNumber
    INLINENWORDS (silver) ; entspricht eI.BackgroundColor

Keybutton_Style:
    INLINENWORDS (no_change) ; entspricht eI.Pos_Xl
    INLINENWORDS (no_change) ; entspricht eI.Pos_Yl
    INLINENWORDS (KEYBUTTON_W) ; entspricht eI.Width

```

```

INLINENWORDS (KEYBUTTON_H) ; entspricht eI.Height
INLINENWORDS (8) ; entspricht eI.SpaceLeft
INLINENWORDS (8) ; entspricht eI.SpaceRight
INLINENWORDS (0) ; entspricht eI.HorizontalAdjust
INLINENWORDS (0) ; entspricht eI.VericalAdjust
INLINENWORDS (transparent) ; entspricht eI.FillColor
INLINENWORDS (transparent) ; entspricht eI.BackColor
INLINENWORDS (transparent) ; entspricht eI.LineColor
INLINENWORDS (black) ; entspricht eI.TextColor
INLINENWORDS (Pos_center) ; entspricht eI.Position
INLINENWORDS (Orientation_0deg) ; entspricht eI.Orientation
INLINENWORDS (normal) ; entspricht eI.Appearance
INLINENWORDS (no_border) ; entspr. eI.BorderStyle
INLINENWORDS (Font_Arial_14n) ; entspricht eI.FontNumber
INLINENWORDS (silver) ; entspricht eI.BackgroundColor

Centerfield_Titel_Style:
INLINENWORDS (no_change) ; entspricht eI.Pos_X1
INLINENWORDS (no_change) ; entspricht eI.Pos_Y1
INLINENWORDS (80) ; entspricht eI.Width
INLINENWORDS (20) ; entspricht eI.Height
INLINENWORDS (0) ; entspricht eI.SpaceLeft
INLINENWORDS (0) ; entspricht eI.SpaceRight
INLINENWORDS (1) ; entspricht eI.HorizontalAdjust
INLINENWORDS (1) ; entspricht eI.VericalAdjust
INLINENWORDS (transparent) ; entspricht eI.FillColor
INLINENWORDS (transparent) ; entspricht eI.BackColor
INLINENWORDS (transparent) ; entspricht eI.LineColor
INLINENWORDS (black) ; entspricht eI.TextColor
INLINENWORDS (Pos_left) ; entspricht eI.Position
INLINENWORDS (Orientation_0deg) ; entspricht eI.Orientation
INLINENWORDS (normal) ; entspricht eI.Appearance
INLINENWORDS (no_border) ; entspricht eI.BorderStyle
INLINENWORDS (Font_System_9bd) ; entspricht eI.FontNumber
INLINENWORDS (silver) ; entspricht eI.BackgroundColor
ENDSUB
; ----- Ende Styles -----

; H A U P T P R O G R A M M -----
BEGINVIEW
EVE.Init() ; EVE ANNA initialisieren
Display.ClearColor(white)
HotKey.InputFlush()
Display.Prepare()
CallSubroutine(Draw_Layout_A)
CallSubroutine(Draw_Titel)
CallSubroutine(Titelbalken) ; Überschrift 'Privatbahn Ägerital'
CallSubroutine(Draw_Center_Layout)

; Installieren der HotKeys
; HotKey.InstallLocalKey("A",HK_Test_A,0x01) ; Key A
; HotKey.InstallLocalKey("B",HK_Test_B,0x02) ; Key B
;
HotKey.InstallLocalKey("C",Draw_Layout_A,0x03) ; Key C
HotKey.InstallLocalKey("D",Draw_Layout_B,0x04) ; Key D
HotKey.InstallLocalKey("E",Esc,0x05) ; Key E
HotKey.InstallLocalKey("F",Popup,0x06) ; Key F
;
; HotKey.InstallLocalKey("G",HK_Test_G,0x07) ; Key G
; HotKey.InstallLocalKey("H",HK_Test_H,0x08) ; Key H
;
HotKey.InstallLocalKey("I",Felder_ausfuellen,0x09) ; Key I
HotKey.InstallLocalKey("J",Felder_ausfuellen,0x0A) ; Key J
HotKey.InstallLocalKey("K",Felder_ausfuellen,0x0B) ; Key K
HotKey.InstallLocalKey("L",Felder_ausfuellen,0x0C) ; Key L
HotKey.InstallLocalKey("M",Felder_ausfuellen,0x0D) ; Key M

```

```
HotKey.InstallLocalKey("N",Felder_ausfuellen,0x0E) ; Key N
HotKey.InstallLocalKey("O",Felder_ausfuellen,0x0F) ; Key O
HotKey.InstallLocalKey("P",Felder_ausfuellen,0x11) ; Key P

; Hotspot rechts oben für Screenshot -----
eI.R01 := 0
Load.Geometry_XYWH(600,0,40,40) ; Pos. u Dimension des HotSpots
HotSpot.Install(NIL,NIL,ScreenDump,NIL)

Display.Show()

HotSpot.TableEnable()
HotKey.EnableLocalKeys()
HotKey.TableEnable()

LOOP
ENDLOOP

ENDVIEW
```

