SIEMENS

SIJECT OP15B
Menu Editor
User Manual

Technical Manual 06.2003 Edition

Manufacturer Documentation

SIEMENS

SIJECT OP15B
Menu Editor
User Manual

Technical Manual

Valid for

SIJECT OP15B

06.2003 Edition

Introduction 1

Editing Menus 2

Menu Compiler 3

Key Events and Functions 4

Abbreviations 5

SIJECT OP15B Documentation

Key to editions

The editions listed below have been published prior to the current edition.

The column headed “Note” lists the amended sections, with reference to the previous edition.

Marking of edition in the “Note” column:

A .... New documentation
B .... Unchanged reprint with new order number
C .... Revised edition of new issue
If any technical details presented on one of these pages have been changed with reference to the previous edition, it is indicated by another edition number in the header of the respective page.

Edition	Order No.	Note
06.2003	none	A

Order No. none
Printed in People’s Republic of China

Siemens-Aktiengesellschaft.

Contents

1. INTRODUCTION 1-1

1.1. What is the Menu Editor for OP15B? 1-1
1.2. Preconditions 1-1
1.3. Overview 1-2

2. EDITING MENUS 2-7

2.1. The VB Main Window 2-7
2.2. Project File, Menus and Keyboards 2-8
2.2.1. Project 2-8
2.2.2. Menus 2-10
2.2.3. Keyboards 2-23
2.2.4. Customer Specific Picture 2-25

3. MENU COMPILER (MECO) 3-26

3.1. Installation 3-26
3.2. Usage 3-27

4. KEY EVENTS AND FUNCTIONS 4-28

4.1. Predefined Key events 4-28
4.2. Communication Parameters 4-30

5. ABBREVIATIONS 5-32

5.1. List of abbreviations 5-32

1. Introduction

1.1. What is the Menu Editor for OP15B?

Menu Editor

The Menu Editor for OP15B is a Microsoft Visual Basic based, low cost Engineering Tool to allow easy editing of menu sets for OP15B. It is not supposed to replace commercial engineering tools like ProTool or ProTool Light. Both tools are of course more feature rich, but also much bigger and consume a big amount of resources which are simply not available on the low-cost operator panel OP15B.

The Menu Editor uses only one type of the possible Visual Basic targets, the standard.exe. All other VB target types are not supported.

Furthermore this Menu Editor uses only a limited set of so-called VB controls. Menu Forms contain only:

- Labels
- Text Boxes
- List Boxes
- Combo Boxes
- Images

Keyboard Forms contain only:

- Command Buttons

1.2. Preconditions

What's needed?

The OP15B itself does not run a Windows Operating System, does not run Visual Basic executables or supports real Visual Basic code in the Code Window. The Menu Editor for OP15 B uses VB to define the layout of the menus and uses only predefined parameters in the code window. To run the Menu Editor you need the following:

- Windows 95/98, Windows NT/2000 are not tested
  As the Operating System on a IBM compatible PC
- Microsoft Visual Basic 6.0
  To edit menu and keyboard forms
- A Chinese input method, e.g. Chinese Star (optional)
  To input and edit Simplified Chinese characters
- Microsoft Paint (optional)
  To draw a Customer Specific Picture
- Meco (Menu Compiler)
  To compile menus and keyboards
  To compile the customer specific picture
- SIJECT Loader and SIJECT Adapter
  To upload the Menu file to the OP15B
  To upload a CSP file to the OP15B

1.3. Overview

SW Structure

The general software structure of the OP15B is shown in the following picture. Loading Kernel and Firmware are released with each new software version, Menu Data and the Customer Specific Picture are created with the Menu Editor described in this document.

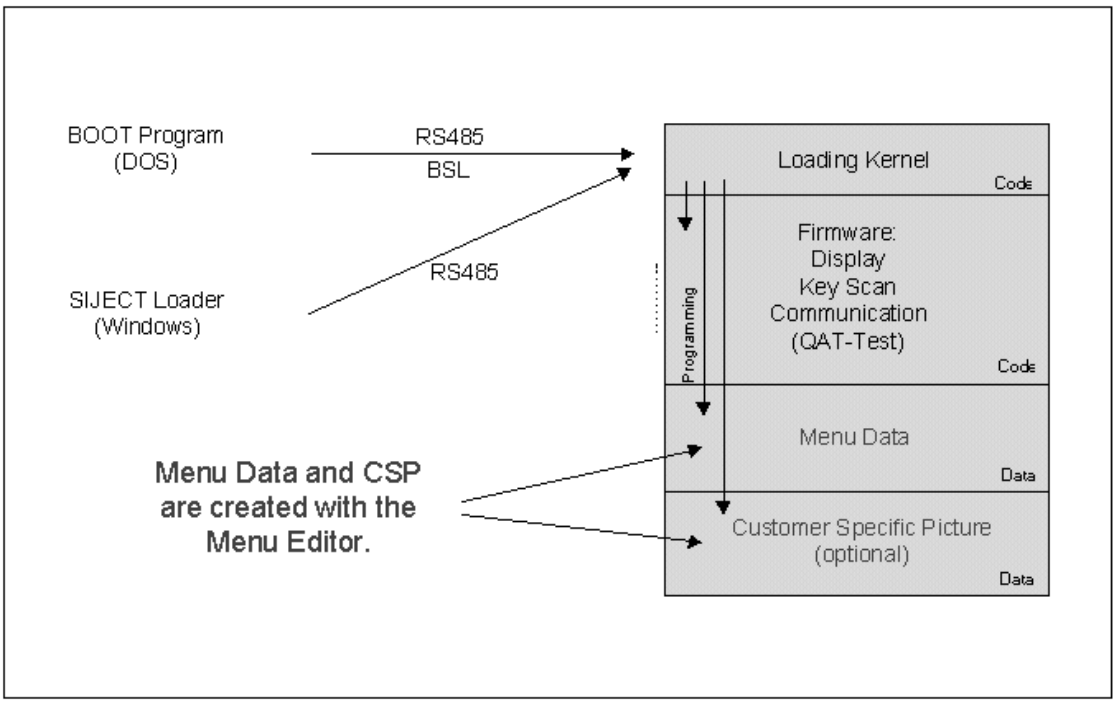

Fig.1-1, Basic software structure on the OP15B

Design Flow

The general design flow to create a new menu set for OP15B is shown in the following pictures:

First Step: Editing the Menus and Keyboards with the Menu Editor in Visual Basic. Parameters are entered in the VB code window related to each object. See Fig. 1-2

Second Step: The saved VB project files are used by the Menu Compiler to create a binary output file what can be loaded to the OP15B, finally. See Fig. 1-3 and 1-4.

Third Step: Upload the newly created binary file to the OP15B with the means of the SIJECT Loader and the SIJECT Adapter. See Fig. 1-5.

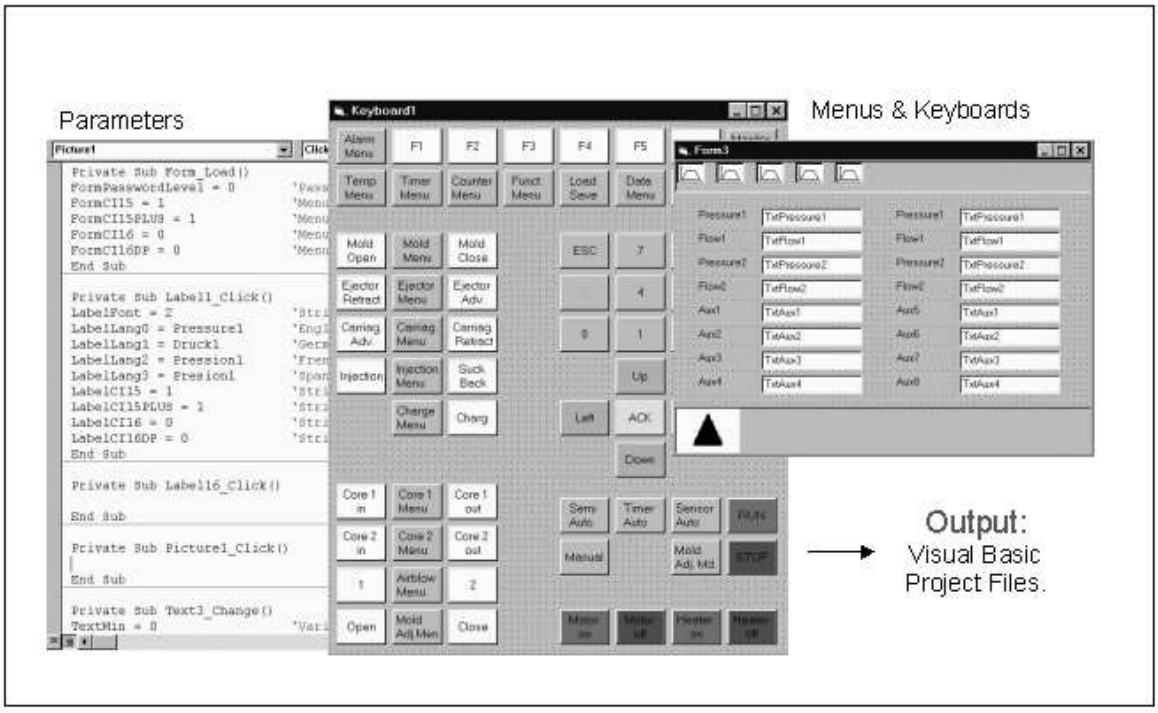

Fig. 1-2, Edit Menus, Keyboards, Parameters in VB, save the project in VB

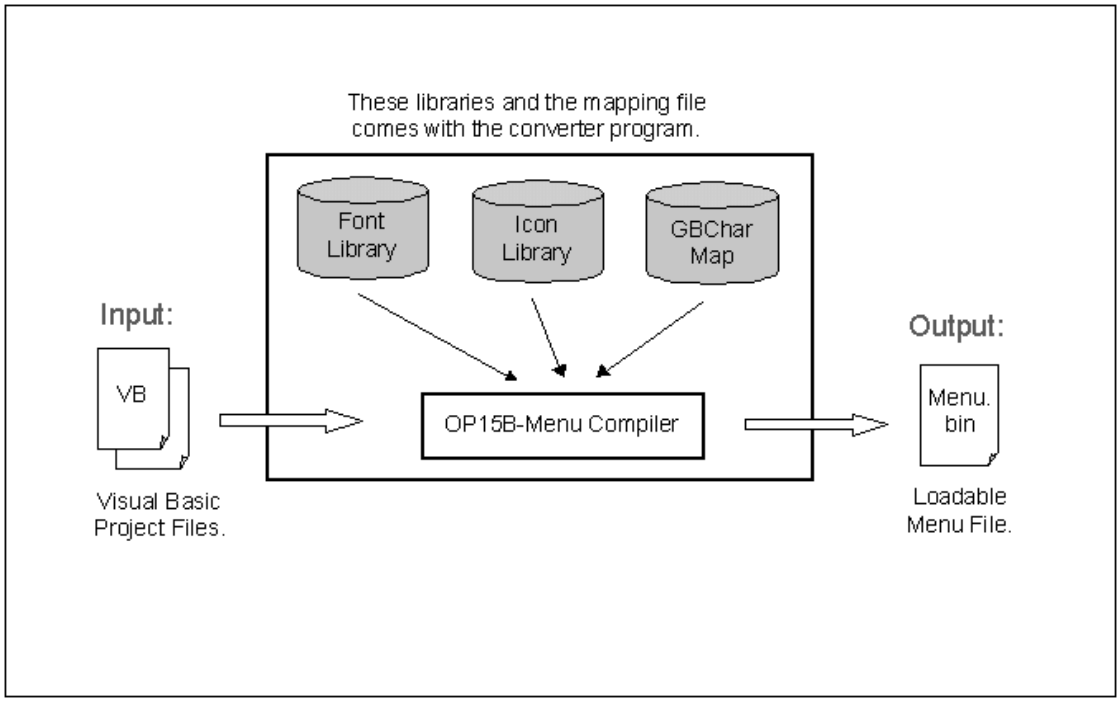

Fig. 1-3, VB project files as input, Menu File as output

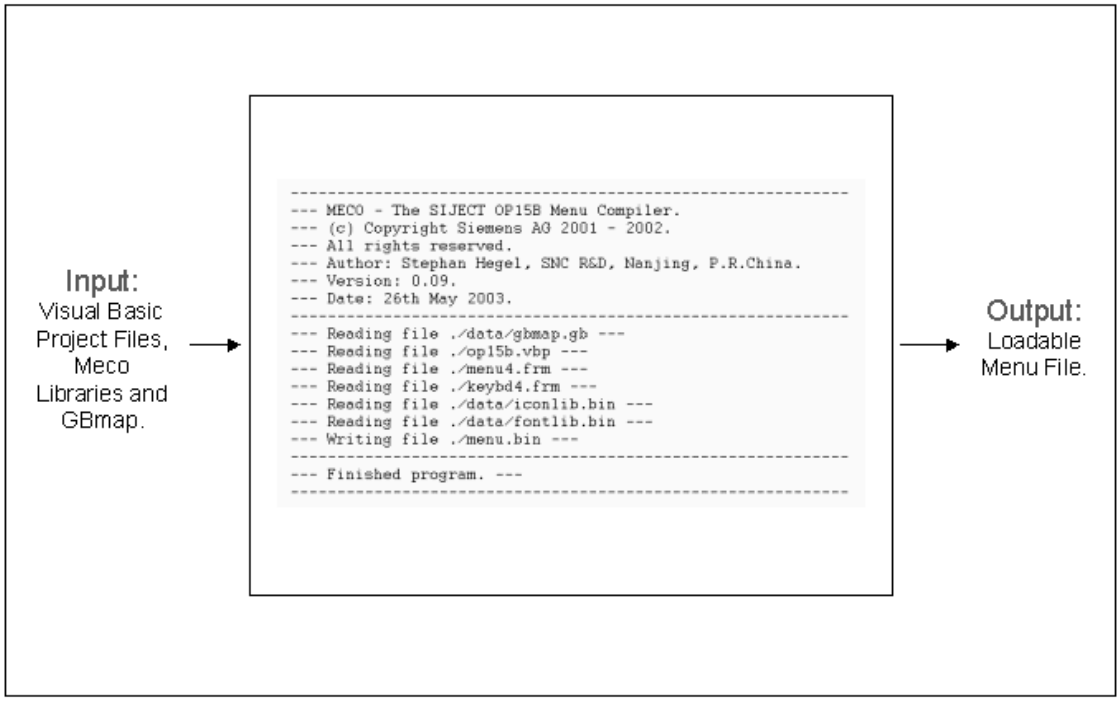

Fig. 1-4, Running the Menu Compiler

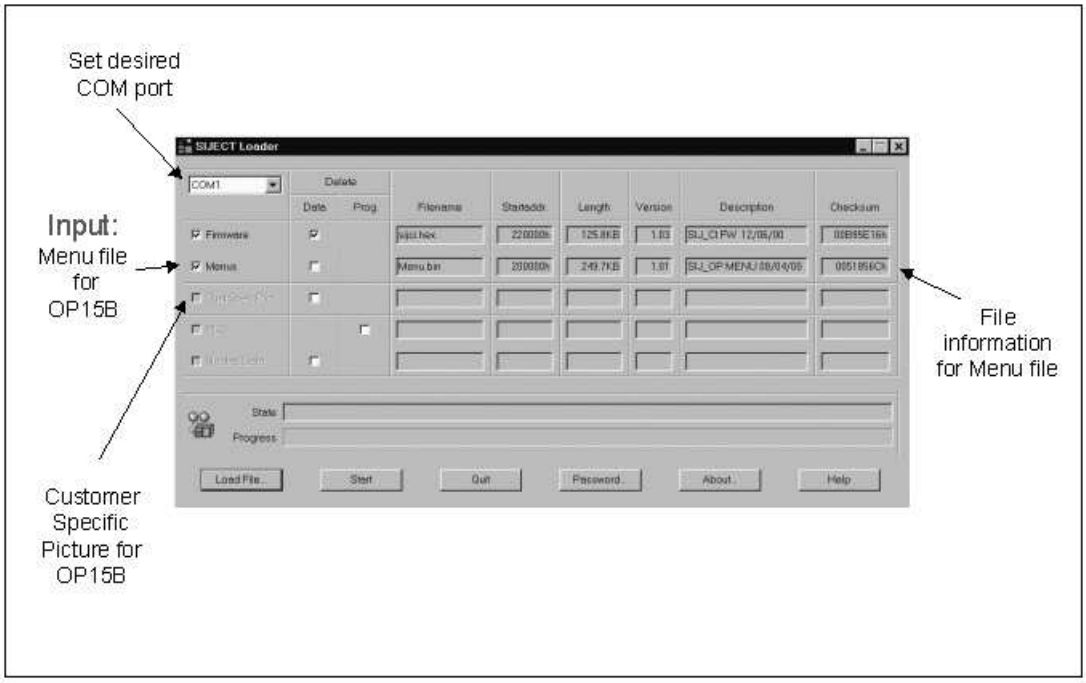

Fig 1-5, Upload the new Menu set to the OP15B

Blocks of Menu Data

The following binary blocks are located in the big binary block Menu Data as shown in Fig. 1-1.

- Icon library, predefined in advance, can be recreated
- Font library, predefined in advance, can be recreated
- String library, created by Meco

- Sub-block for Menus and Keys, created by Meco
- Sub-block with Label information, created by Meco
- Sub-block with Variable data (Text Boxes), created by Meco
- Sub-block with List Box data, created by Meco
- Sub-block with Combo Box data, created by Meco
- Sub-blocks with Icon References, created by Meco

Icon library

The icon / image library can contain up to 255 black/white icons and pictures of different size – remember OP15B is a black/white operator panel. The library is provided with the Menu Compiler but it can be extended by the end user and recreated on request. The whole compiled icon library is stored in the file "data\iconlib.bin" finally.

Standard icons are predefined in advance, they stored in the subdirectory "data\iconlib\std" and have the numbers 0 to 100. The names of these icons are defined to be "iconxx.bmp" where xx is the number between 0 and 100. There is a file “index.bmp” in this subdirectory, which gives an overview of all predefined standard icons.

User icons can be created by the OEM or end user with standard PC drawing tools like MS Paint or full-featured image processing programs like Adobe Photoshop, The Gimp and others, they are stored in the subdirectory "data\iconlib\user" and shall have the numbers 101 to 254. The names for the user icons have to be "iconxx.bmp", where xx is a number between 101 and 254.

The icon with number 255 is reserved for special use of the CSP and must not appear in the icon library itself.

Font library

The font library can contain up to 8 font sets, however, currently there are only 4 font sets supported by the OP15B software. The font library is provided but can be extended by the end user – especially the used Chinese characters and re-created on request. The whole compiled font library is stored in "data\fontlib.bin". The sources of the fonts are stored in the subdirectory "data\fontlib" and must not be changed. The font numbers 0 to 3 are predefined and refer to the following fonts:

- Number 0: English 8x8 dots
- Number 1: English 12x8 dots
- Number 2: English 16x8 dots
- Number 3: Simplified Chinese 16x16 dots

The Simplified Chinese font set contains only the Chinese characters used on the OP15B, not a complete set. The OP15B specific encoding is defined in the file "data\gbmap.gb"

Fonts and Languages

The OP15B operator panel supports up to 4 languages which can be switched on-line. The following languages are predefined, supported and tested carefully:

- Number 0: English
- Number 1: Simplified Chinese
- Number 2: not defined yet (Language using English Font only)
- Number 3: not defined yet (Language using English Font only)

Fonts and languages on the OP15B are independent, e.g. English fonts can be used also when Chinese is selected as the actual language on the OP15B and vice versa.

Best starting point is to use an existing VB project for OP15B and modify it according to your needs.

2. Editing Menus

2.1. The VB Main Window

Main Window

The file names for the VB project and the form files are fixed. The name for the project file is “op15b.vbp”, the file names “menuxx.frm” are for menu forms and the name “keybdxx.from” is supposed to be for keyboard forms. The xx refers to a number between 1 and 63 to support up to 63 menus on the OP15B.

Double-clicking on any control on the menu or keyboard form or the forms itself opens the VB code window to insert parameters.

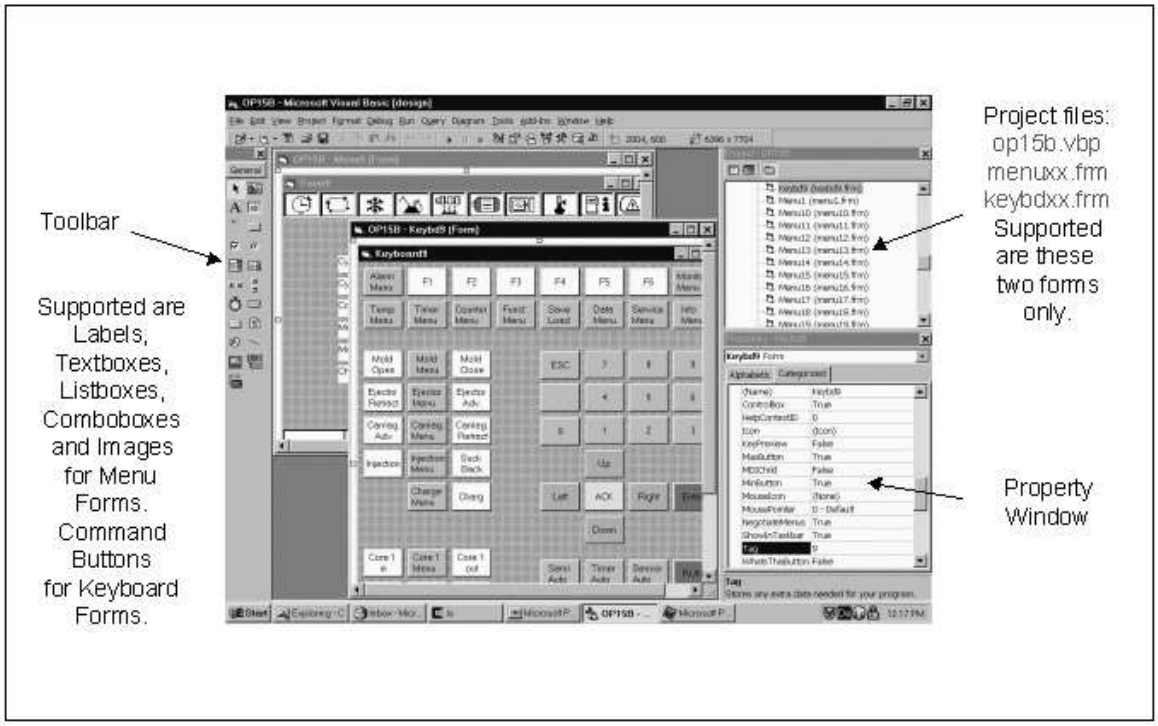

Fig. 2-1, Visual Basic Main Menu

2.2. Project File, Menus and Keyboards

2.2.1. Project

Project File

In the project file we define only the version number of a menu set, Right now supported are only a one or two digit major version number, a one or two digit minor version number but no revision number. The defined version number appears in the SIJECT Loader, finally when uploading the menu file to the OP15B.

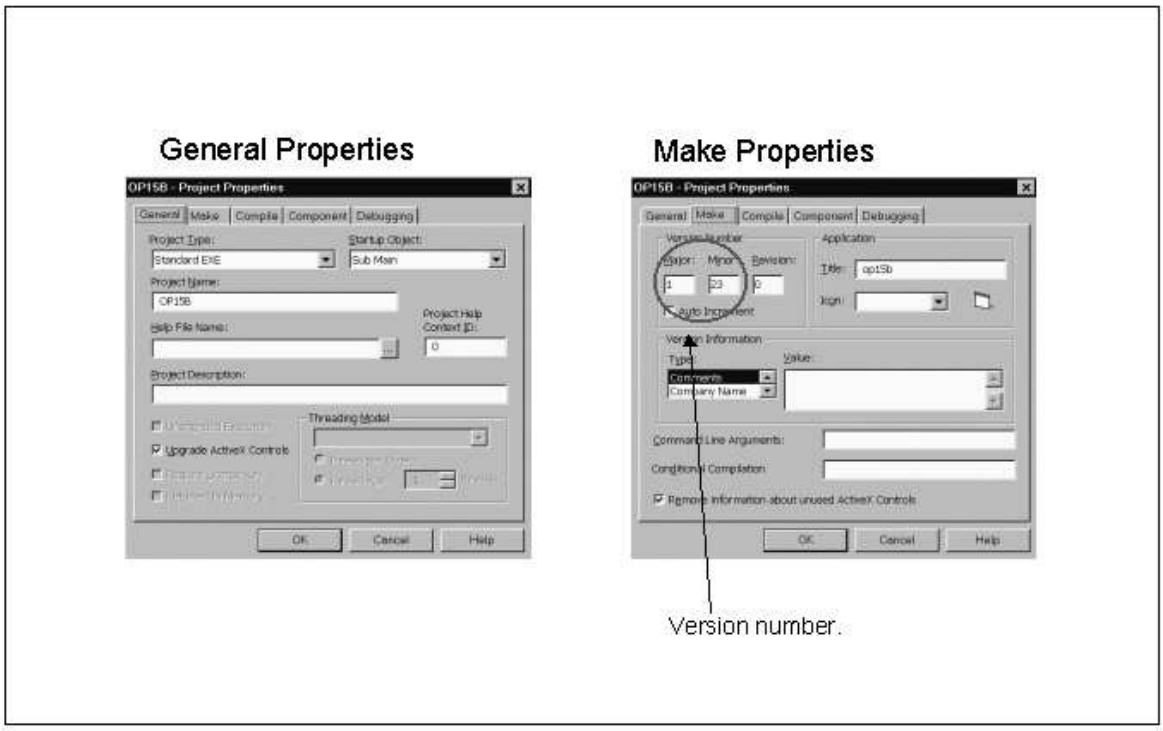

Fig. 2-2, Project properties

Forms

Menu and Keyboard Forms with the same number belong always together, in Fig. 2-3 menu and keyboard number 26.

This allows for defining the keyboard respective key events for every key depending on the actual menu. This feature is necessary for the function keys F1 to F6 anyway, however the user can do this for all other keys as well. A possible use case would be to lock a key in a certain Menu by setting the KeyEvent parameter for this key to 0.

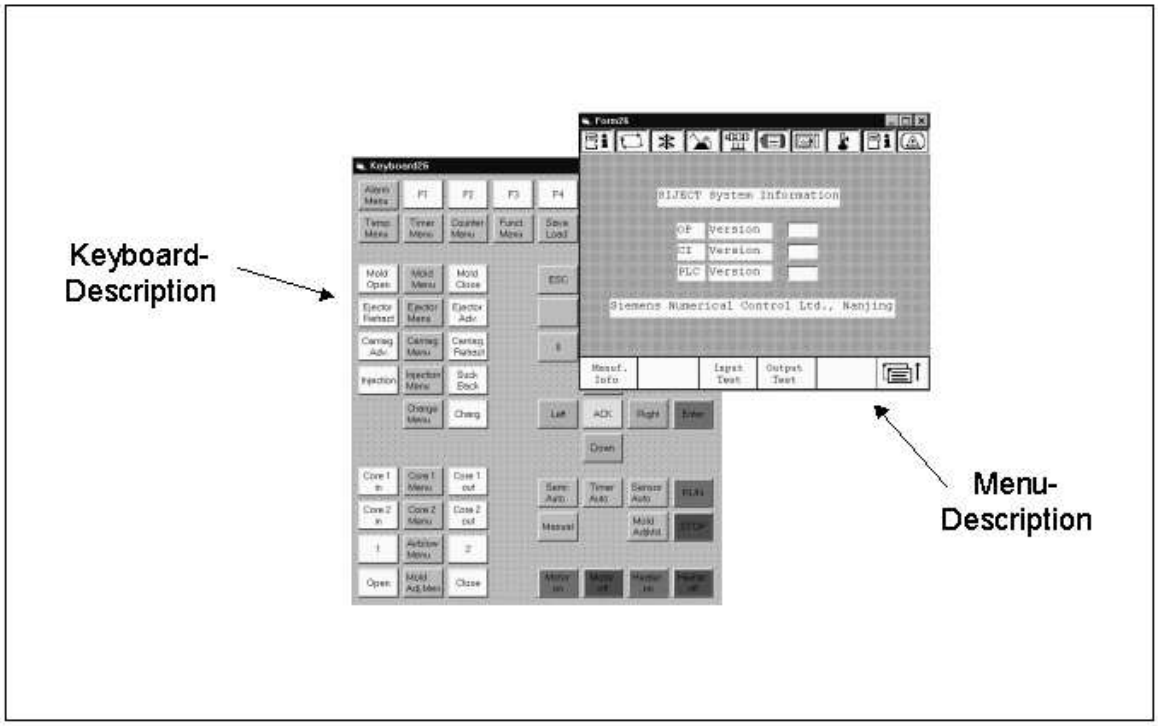

Fig. 2-3, Associated Menu and Keyboard forms

How to copy forms?

Existing Menu and Keyboard Forms can be easily copied to speedup the generation of a new pair of a menu and a keyboard form. The following example demonstrates how to copy an existing Menu27 to Menu28.

At first, copy the “menu27.frm” file to the new file “menu28.frm” in the Windows Explorer. Do the same with “keybd27.frm” to “keybd28.frm”.

Open the single “menu28.frm” file in VB by double clicking on it and change the form’s name to “Menu28”. Save the file. Do the same with “keybd28.frm” and change the form’s name to “Keybd28”. Make sure that you open only the single files in VB, don’t open the whole project.

By double clicking on the “op15b.vbp” file, open the whole project and add the two new forms to your project.

2.2.2. Menus

Menu Form

Menu Forms are used to place labels, text boxes, list boxes, combo boxes and image on it. The size of a menu form is fixed to 6400x4800 because it is used by Meco to scale down this number by the factor 20 to calculate the position of a certain object on the 320 x 240 display. Please don't change this size.

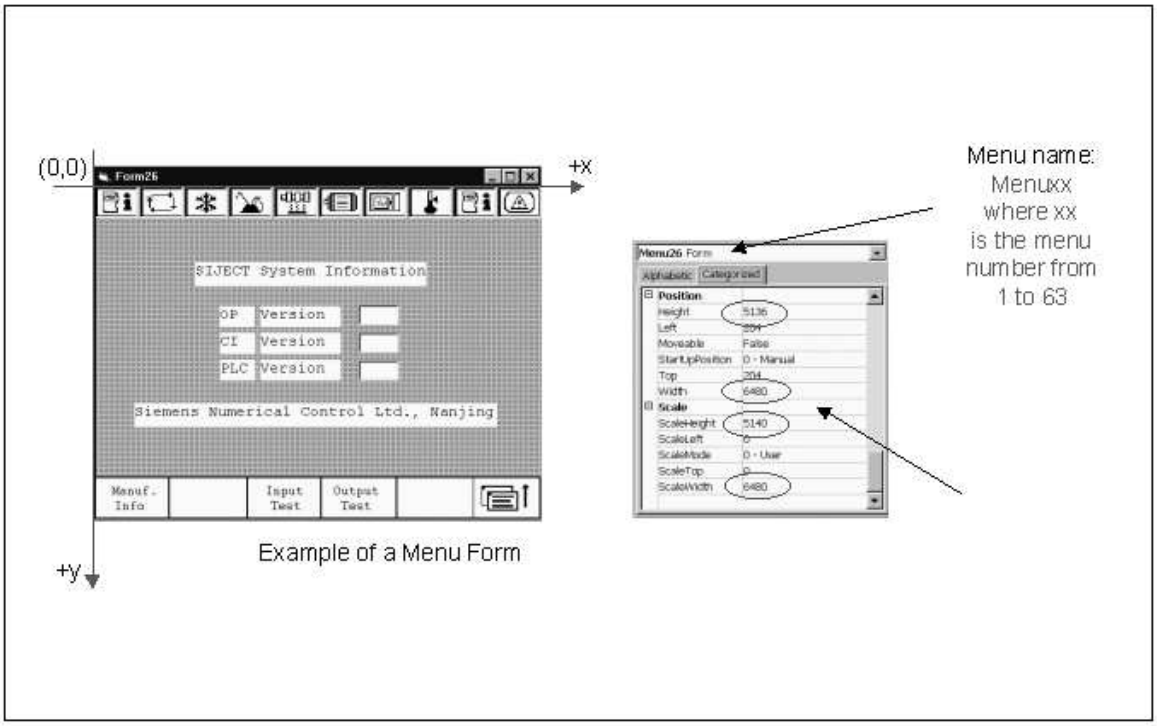

Fig. 2-4, Menu Form with coordinates and reference point

Notes

The origin for the position calculation is defined to be in the upper-left corner of the Menu Form.

Menu Form names must be Menuxx, where xx is a number between 1 and 63 to support up to 63 different menus on the OP15B.

Parameter

Menu Forms use the following predefined parameters in the VB code window, here with some example values:

Parameter	Comment	Default
FormMenuNumber = 1	Number of Menu	[0]
FormPasswordLevel = 0	Password level is 0	[0]
FormCI15 = 1	Valid for CI15	[1]
FormCI15PLUS = 1	Valid for CI15+	[1]
FormCI16 = 0	Not valid for CI16	[1]
FormCI16DP = 0	Not valid for CI16DP	[1]

Table 2-1, Parameter for Menu Form

The values in [ ] indicate the default value, when the parameter is not present.

Notes

The menus effective number is extracted from the "FormMenuNumber" parameter, from the "Tag" field in the VB property window or from the file name. The Password Levels for a menu on OP15B can be 0, 1 and 2, where 0 means no password protection.

Labels Labels are used to define fixed strings on a Menu Form.

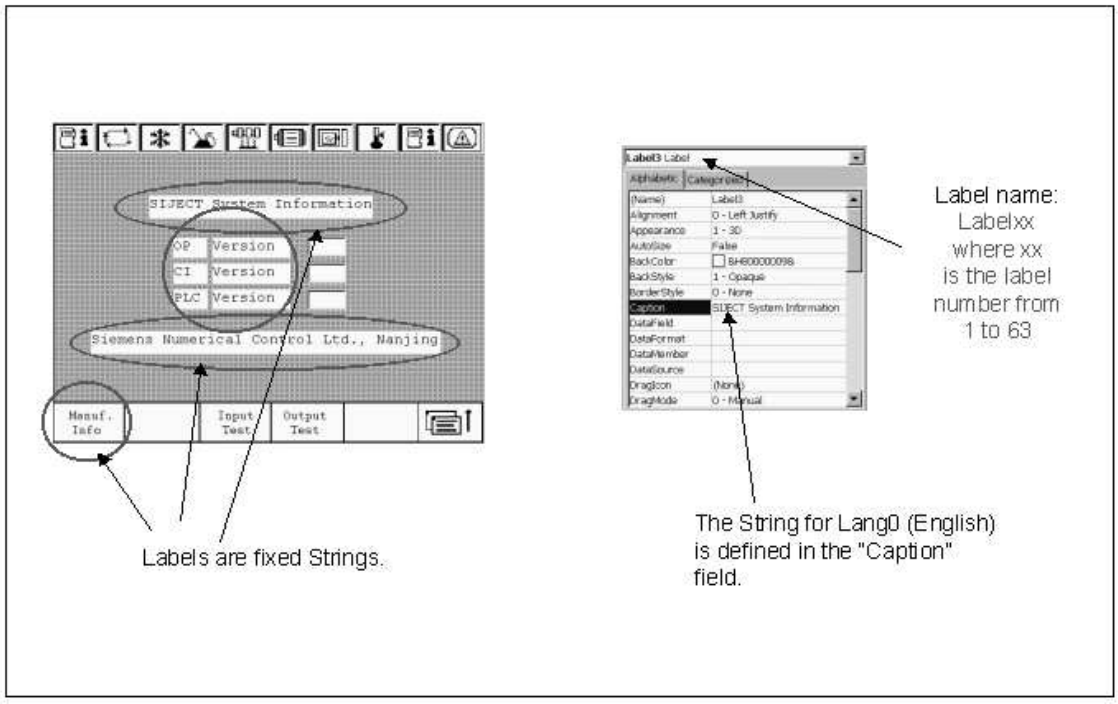

Fig. 2-5, Labels

Label names must be Labelxx, where xx is a number between 1 and 63 to support up to 63 labels per Menu.

Parameter Predefined Parameters for labels with some example values are:

Parameter	Comment	Default
LabelFont0 =1	Font for Language 0	[0]
LabelFont1 =3	Font for Language 1	[0]
LabelFont2 =1	Font for Language 2	[0]
LabelFont3 =1	Font for Language 3	[0]
LabelLang0 = “Label”	English	[“”]
LabelLang1 = “”	Chinese	[“”]
LabelLang2 = “”	t.b.d.	[“”]
LabelLang3 = “”	t.b.d.	[“”]
LabelXOffset0 = 0	Offset x for Lang. 0, in dots	[0]
LabelYOffset0 = 0	Offset y for Lang. 0, in dots	[0]
LabelXOffset1 = 0	Offset x for Lang. 1, in dots	[0]
LabelYOffset1 = -2	Offset y for Lang. 1, in dots	[0]
LabelXOffset2 = 0	Offset x for Lang. 2, in dots	[0]
LabelYOffset2 = 0	Offset y for Lang. 2, in dots	[0]
LabelXOffset3 = 0	Offset x for Lang. 3, in dots	[0]
LabelYOffset3 = 0	Offset y for Lang. 3, in dots	[0]
LabelCI15 = 1	Valid for CI15	[1]
LabelCI15PLUS = 1	Valid for CI15+	[1]
LabelCI16 = 0	Not valid for CI16	[1]
LabelCI16DP = 0	Not valid for CI16DP	[1]

Table 2-2, Parameter for Label

The values in [ ] indicate the default value, when the parameter is not present.

Notes

LabelXOffset and LabelYOffset describe an additonal offset for a particular language compared to the position of a variable calculated from the position in the menu form. The value refers to dots of the 320x240 display. Please note, that a negative x-value moves the concerned Label to the left, a negative y-value move the Label up.

The string for language 0 (English) is usually defined in the "Caption" field in the VB property window to display it on the label on the Menu Form, but can be overwritten by the "LabelLang0" parameter in the VB code window.

To get a realistic impression of the width of the fonts and strings in VB in comparison to the font on the OP15B, see the recommendations in the following picture. This does not mean that this font is then really used on the OP15B, instead we use the font defined in the parameter set for this Label and finally provided in the font library.

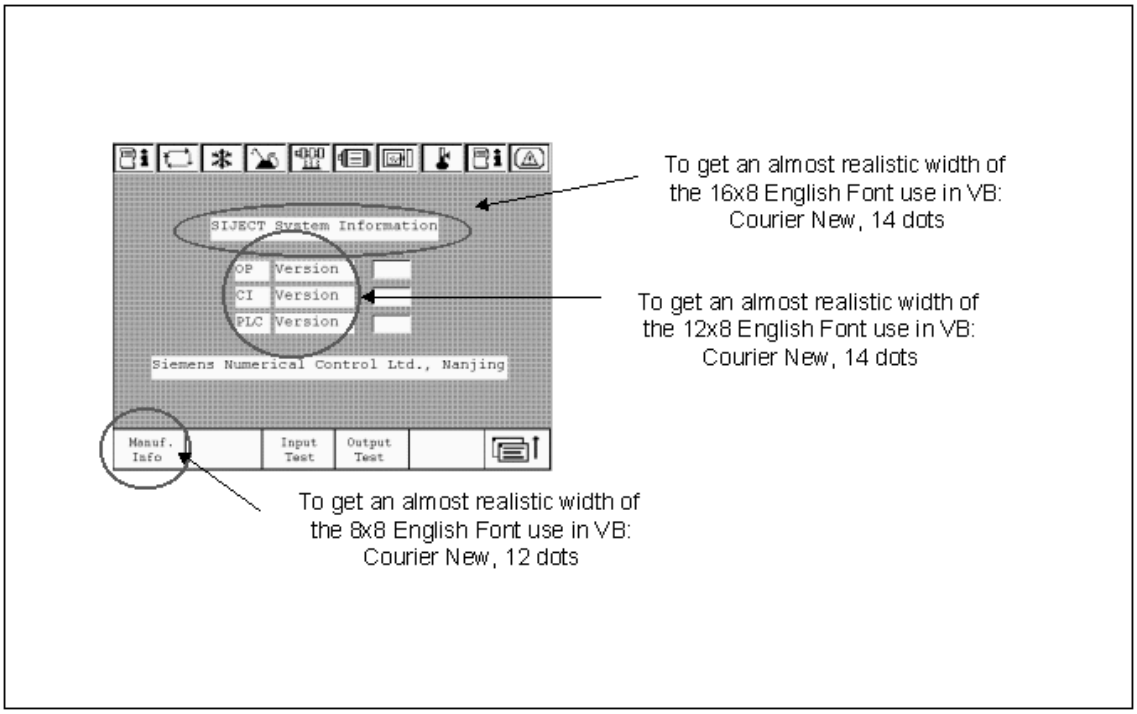

Fig 2-6, Using selected Windows font to get an almost realist picture of the Label's size.

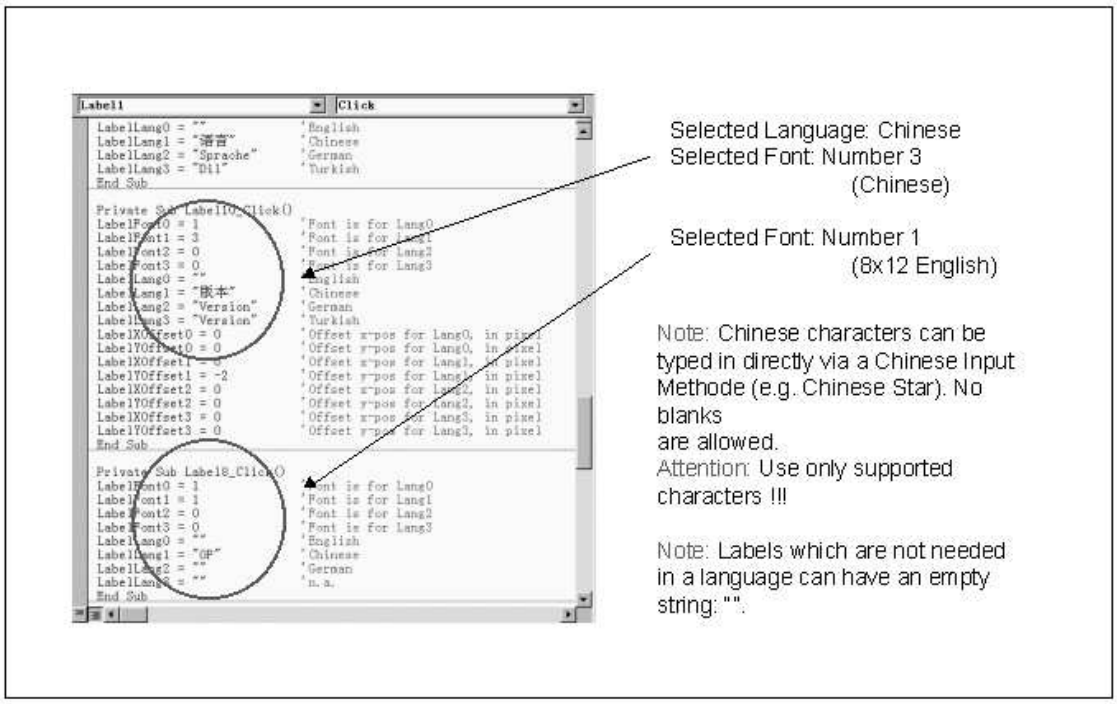

Fig 2-7, Example for a Label with different languages and fonts

Text Boxes

The Textboxes are used for displaying variables on a menu representing actual and setting values in the PLC.

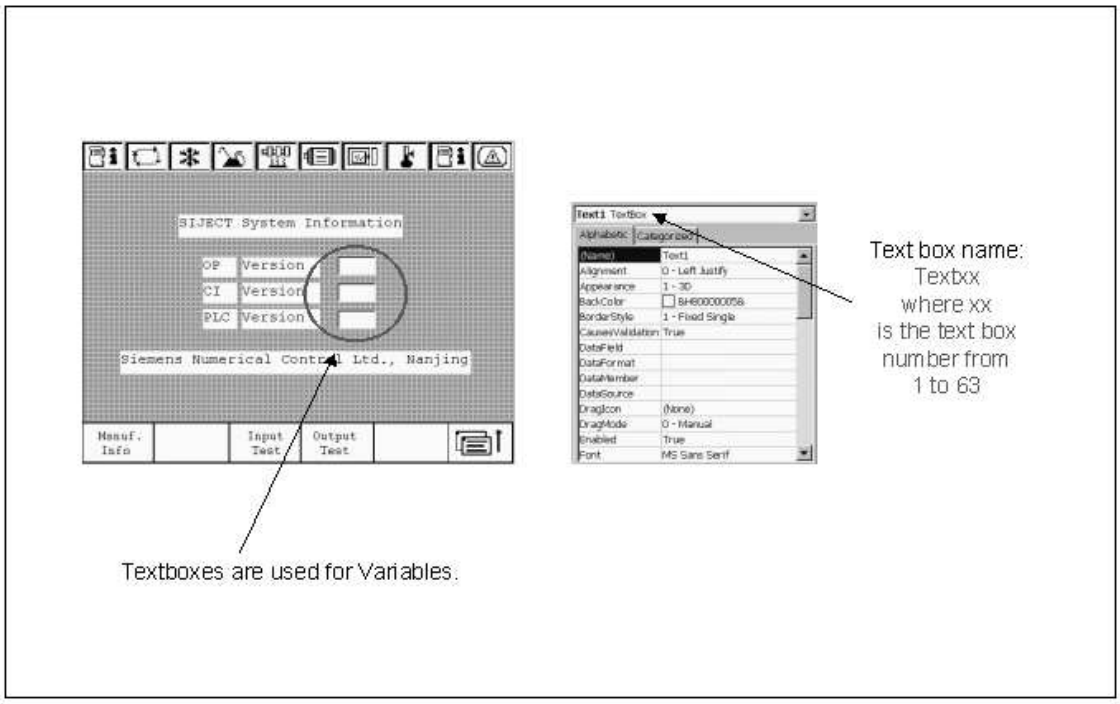

Fig 2-8, Text Boxes

They are named Textxx, where xx is a number between 1 and 63 to support up to 63 variables per Menu. Predefined Parameters in the VB code window with some example values are:

Parameter	Comment	Default
TextMin = 0	Variable's Mininum	[0]
TextMax = 10000	Variable's Maximum	[0]
TextAttribute = &H42&	Variable Attribute	[0]
TextAppearance = &H25&	Variable Appearance	[0]
TextType = &HFF&	Communication Type	[0]
TextIndex = &H1&	Communication Index	[0]
TextFont0 =1	Font for Language 0	[0]
TextFont1 =3	Font for Language 1	[0]
TextFont2 =0	Font for Language 2	[0]
TextFont3 =0	Font for Language 3	[0]
TextUp = 0	Cursor position after up	[0]
TextDown = 0	Cursor position after down	[0]
TextLeft = 0	Cursor position after left	[0]
TextRight = 0	Cursor position after right	[0]
TextEnter = 0	Cursor position after enter	[0]
TextXOffset0 = 0	Offset x for Lang. 0, in dots	[0]
TextYOffset0 = 0	Offset y for Lang. 0, in dots	[0]
TextXOffset1 = 0	Offset x for Lang. 1, in dots	[0]
TextYOffset1 = -2	Offset y for Lang. 1, in dots	[0]
TextXOffset2 = 0	Offset x for Lang. 2, in dots	[0]
TextYOffset2 = 0	Offset y for Lang. 2, in dots	[0]
TextXOffset3 = 0	Offset x for Lang. 3, in dots	[0]
TextYOffset3 = 0	Offset y for Lang. 3, in dots	[0]
TextCI15 = 1	Valid for CI15	[1]

TextCI15PLUS = 1	Valid for CI15+	[1]
TextCI16 = 0	Not valid for CI16	[1]
TextCI16DP = 0	Not valid for CI16DP	[1]

Table 2-3, Parameter for Text Boxes

The values in [ ] indicate the default value, when the parameter is not present.

Notes

TextAttribute

```
Bit 7-6: Communication Target
Bit 5-4: Read / Read-Write
Bit 3:   Update Priority
Bit 2-0: Type
Communicaton Target:
      0: PLC Variable
      1: OP variable
Read / Read-Write:
      0: Read only
      1: Read / Write
      2: Write only
Update Priority:
      0: normal priority
      1: high priority
Variable Type:
      0: integer variable
      1: unsigned integer variable
      2: float variable
```

TextAppearance

```
Bit 7-4: Point position
Bit 3-0: Length
Point Position:
      The position of the point within this
      variable. This applies only to floating
      point variables, otherwise Point Position
      should be 0.
Length:
      The Maximum length of this variable
      (excluding the point itself).
Example for TextAppearance:
      The variable 973.26 should have a
      definition of Length=5 and PointPos=2.
```

TextType and TextIndex

Describe an address for the communication with the PLC or for an OP15B specific communication. See the list of predefined communication parameters in the chapter “Key events and Functions”.

TextXOffset and TextYOffset

Describe an additonal offset for an particular language compared to the position of a variable calculated from the position in the menu form. The value refers to dots of the 320x240 display.

List Boxes

List Boxes are used for creating message windows with predefined messages, which are controlled by the PLC finally. The names for List Boxes are defined to be Listxx, where xx is a number between 1 and 16 to support up to 16 List Boxes per Menu. The number of List Box items is limited to 128.

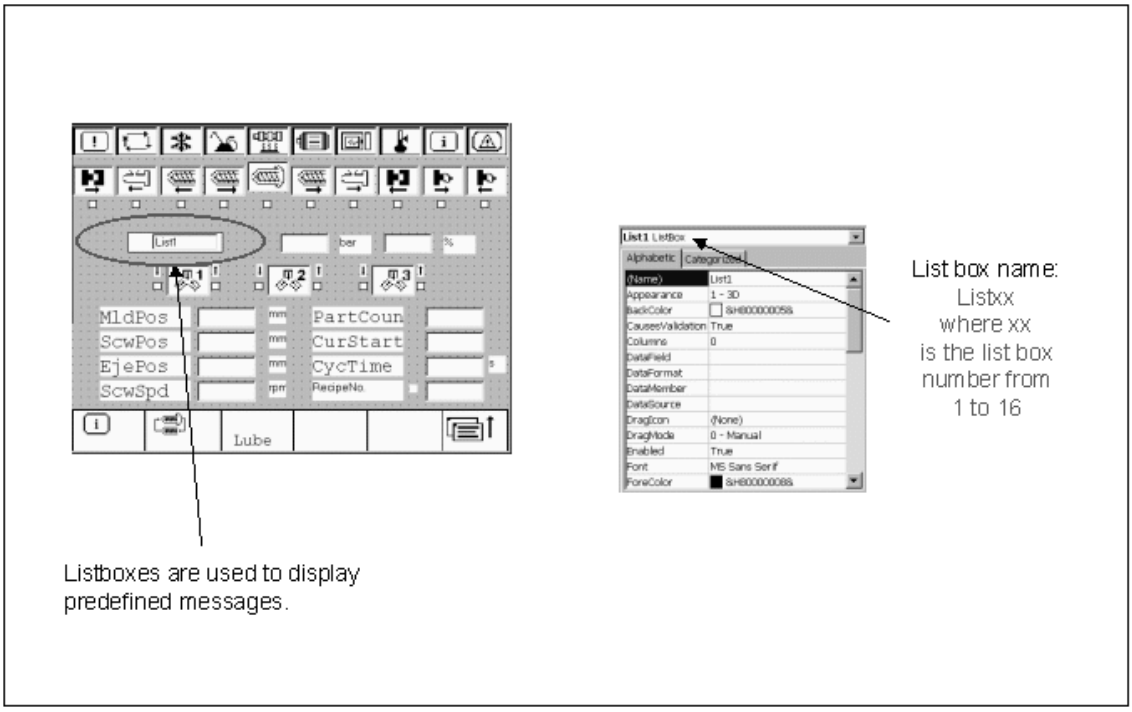

Fig 2-9, List Boxes

Parameter

Predefined Parameters for List Boxes with example values are:

Parameter	Comment	Default
ListStart = 0	Fixed.	[0]
ListAttribute = &HCC&	Variable Attribute	[0]
ListAppearance = &HEE&	Variable Appearance	[0]
ListType = &HFF&	Communication Type	[0]
ListIndex = &H1&	Communication Index	[0]
ListFont0 =0	Font for Language 0	[0]
ListFont1 =3	Font for Language 1	[0]
ListFont2 =0	Font for Language 2	[0]
ListFont3 =0	Font for Language 3	[0]
ListLang0 = “Error #1”	1st Message for Language 0	[“”]
ListLang0 = “Error #2”	2nd Message for Language 0	[“”]
ListLang1 = “”	1st Message for Language 1	[“”]
ListLang1 = “”	2nd Message for Language 1	[“”]
ListLang2 = “Fehler 1”	1st Message for Language 2	[“”]
ListLang2 = “Fehler 2”	2nd Message for Language 2	[“”]
ListLang3 = “”	1st Message for Language 3	[“”]
ListLang4 = “”	2nd Message for Language 3	[“”]
ListCI15 = 1	Valid for CI15	[1]
ListCI15PLUS = 1	Valid for CI15+	[1]
ListCI16 = 0	Not valid for CI16	[1]
ListCI16DP = 0	Not valid for CI16DP	[1]

Table 2-4, Parameter for List Boxes

The values in [ ] indicate the default value, when the parameter is not present.

ListAttribute

```
Bit 7-6: Communication Target
Bit 5-4: Not used.
Bit 3:   Update Priority
Bit 2-0: Not use.
Communicaton Target:
      0: PLC Variable
      1: OP variable
Update Priority
      0: normal priority
      1: high priority
```

ListAppearance

```
Not used.
```

ListType and ListIndex

Describe an address for the communication with the PLC or for an OP15B specific communication. See the list of predefined communication parameters in the chapter “Key events and Functions”.

Combo Boxes

Combo Boxes are used to select a setting value from a list of predefined values. The names are defined to be Comboxx, where xx is a number between 1 and 24 to support up to 24 Combo Boxes per Menu. The number of Combo Box items is limited to 8.

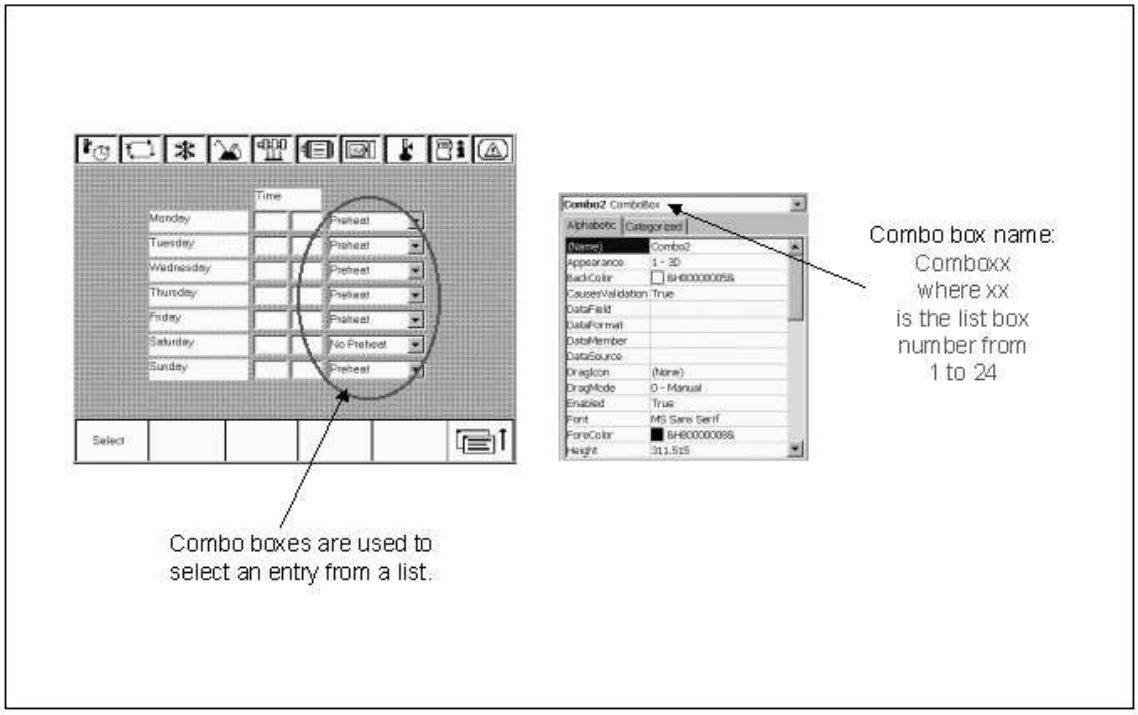

Fig. 2-10, Combo Boxes

Parameter

Predefined Parameters with example values are:

Parameter	Comment	Default
ComboStart = 0	Fixed.	[0]
ComboAttribute = &HFF&	Variable Attribute	[0]
ComboAppearance = &H2&	Variable Appearance	[0]
ComboType = &HFF&	Communication Type	[0]
ComboIndex = &H1&	Communication Index	[0]
ComboFont0 =0	Font for Language 0	[0]
ComboFont1 =3	Font for Language 1	[0]
ComboFont2 =0	Font for Language 2	[0]
ComboFont3 =0	Font for Language 3	[0]
ComboLang0 = “on”	1st Message for Language 0	[“”]
ComboLang0 = “off”	2nd Message for Language 0	[“”]
ComboLang0 = “auto”	3rd Message for Language 0	[“”]
ComboLang1 = “”	1st Message for Language 1	[“”]
ComboLang1 = “”	2nd Message for Language 1	[“”]
ComboLang1 = “”	3rd Message for Language 1	[“”]
ComboLang2 = “ein”	1st Message for Language 2	[“”]
ComboLang2 = “aus”	2nd Message for Language 2	[“”]
ComboLang2 = “auto”	3rd Message for Language 2	[“”]
ComboLang3 = “”	1st Message for Language 3	[“”]
ComboLang3 = “”	2nd Message for Language 3	[“”]
ComboLang3 = “”	3rd Message for Language 3	[“”]
ComboUp = 0	Cursor position after up	[0]
ComboDown = 0	Cursor position after down	[0]
ComboLeft = 0	Cursor position after left	[0]

ComboRight = 0	Cursor position after right	[0]
ComboEnter = 0	Cursor position after enter	[0]
ComboXOffset0 = 0	Offset x for Lang. 0, in dots	[0]
ComboYOffset0 = 0	Offset y for Lang. 0, in dots	[0]
ComboXOffset1 = 0	Offset x for Lang. 1, in dots	[0]
ComboYOffset1 = -2	Offset y for Lang. 1, in dots	[0]
ComboXOffset2 = 0	Offset x for Lang. 2, in dots	[0]
ComboYOffset2 = 0	Offset y for Lang. 2, in dots	[0]
ComboXOffset3 = 0	Offset x for Lang. 3, in dots	[0]
ComboYOffset3 = 0	Offset y for Lang. 3, in dots	[0]
ComboCI15 = 1	Valid for CI15	[1]
ComboCI15PLUS = 1	Valid for CI15+	[1]
ComboCI16 = 0	Not valid for CI16	[1]
ComboCI16DP = 0	Not valid for CI16DP	[1]

Table 2-5, Combo Boxes

The values in [ ] indicate the default value, when the parameter is not present.

Notes

ComboAttribute

```
Bit 7-6: Communication Target
Bit 5-4: Read / Read-Write
Bit 3:   Update Priority
Bit 2-0: Type
Communicaton Target:
      0: PLC Variable
      1: OP variable
Read / Read-Write
      0: Read only
      1: Read / Write
      2: Write only
Type
      0: unsigned integer variable
```

ComboAppearance

Not used.

ComboType and ComboIndex

Describe an address for the communication with the PLC or for an OP15B specific communication. See the list of predefined communication parameters in the chapter “Key events and Functions”.

ComboXOffset and ComboYOffset

Describes an additonal offset for an particular language compared to the position of a variable calculated from the position in the menu form. The value refers to dots of the 320x240 display.

Images

All kind of graphical information is displayed as an icon or image. Icon names are defined to be Imagexx, where xx is a number between 1 and 63 to support up to 63 icons or images on an OP15B menu.

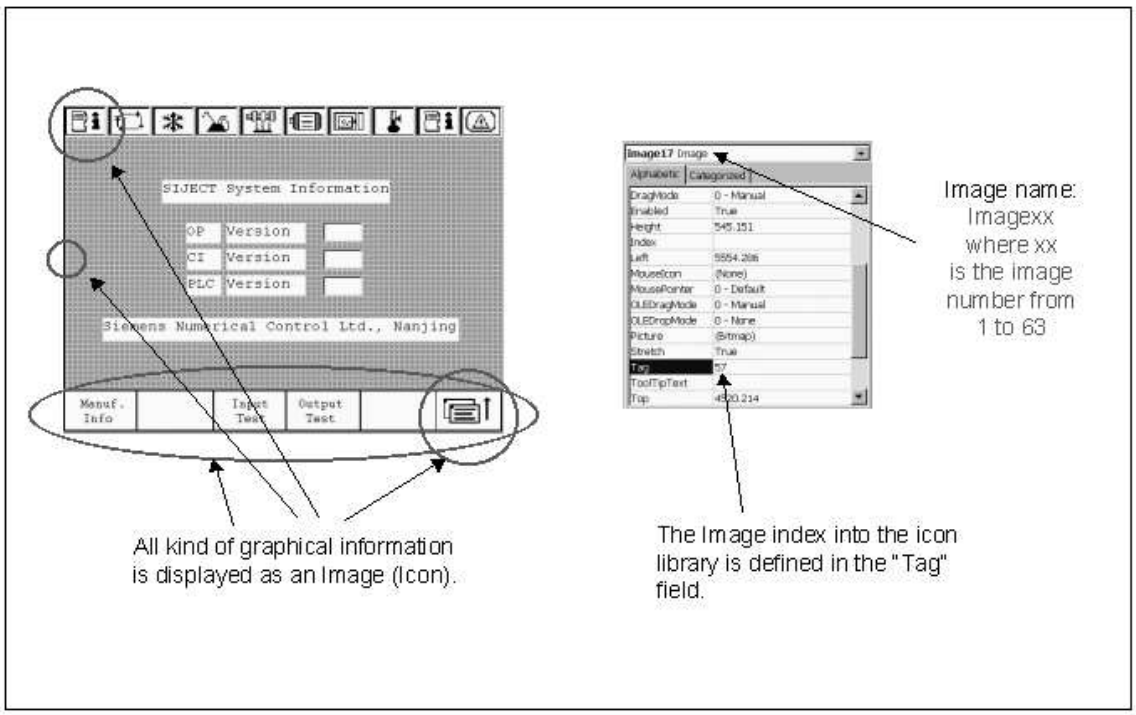

Fig. 2-11, Some samples of icons / images on a OP15B menu.

Parameter

Predefined Parameters for icons and images with example values are:

Parameter	Comment	Default
ImageNumber = 57	Index into the Icon Library	[0]
ImageAttribute = &H0&	Image Attribute	[0]
ImageAppearance = &H1&	Image Appearance	[0]
ImageType = &HFF&	Communication Type	[0]
ImageIndex = &H1&	Communication Index	[0]
ImageCI15 = 1	Valid for CI15	[1]
ImageCI15PLUS = 1	Valid for CI15+	[1]
ImageCI16 = 0	Not valid for CI16	[1]
ImageCI16DP = 0	Not valid for CI16DP	[1]

Table 2-6, Parameter for Icons / Images

The values in [ ] indicate the default value, when the parameter is not present.

Notes

ImageType and ImageIndex

Describe an address for the communication with the. See the list of predefined communication parameters in the chapter “Key events and Functions”.

ImageAttribute

Not used.

ImageAppearance

```
Bit 0: Icon Display Mode
Icon Display Mode:
      0: Reverse Mode
      1: Flash Mode
      See also Image Control below.
```

Image Display

Images are displayed in the order of their names. Images with a higher number overwrite images with lower numbers, e.g.: "Image17" overwrites "Image11". Images are displayed first on a menu and can be overwritten by any other VB control. The "ImageNumber" parameter overwrites the "Tag" definition in VB property window. If no "Tag" or "ImageNumber" defined we fall back to Image index "0".

Image Control

Like Text Boxes, List Boxes and Combo Boxes, Images can be controlled by the PLC. The control by the PLC takes place by sending a word value back when addressed with a certain ImageType and ImageIndex. The definitions for the Low Byte of the communication with the PLC are as follows:

For Mode 0 (ImageAppearance = &H0&):

```
0 - display icon normally
1 - display icon reversed
2 - display icon flashing slowly
3 - display icon flashing quickly
4 – remove / switch off icon
```

For Mode 1 (ImageAppearance = &H1&):

```
0 - display icon normally
1 - display icon flashing slowly
2 - display icon reversed
3 - display icon flashing quickly
4 – remove / switch off icon
```

Images (Icons) can be replaced by another image (icon) if it has exactly the same size. The definitions of the High Byte for the communication with the PLC are:

```
0 - display default icon which is defined in the
Menu Form.
1 to 254 - display icon from the icon library
with exactly this index.
```

If ImageType and ImageIndex are "&H0&", the icon is statically displayed and isn't be controlled by the PLC.

2.2.3. Keyboards

Keyboard Form

Keyboard Forms are used to place command buttons on the keyboard form to simulate the layout of an OP15B keyboard. Size is fixed to allow easy editing within VB, but it is not used on the OP15B. We use it only to extract key events from it. Names for keyboards must be Keybdxx, where xx is a number between 1 and 63 to corresponded to exactly the same number of possible Menus.

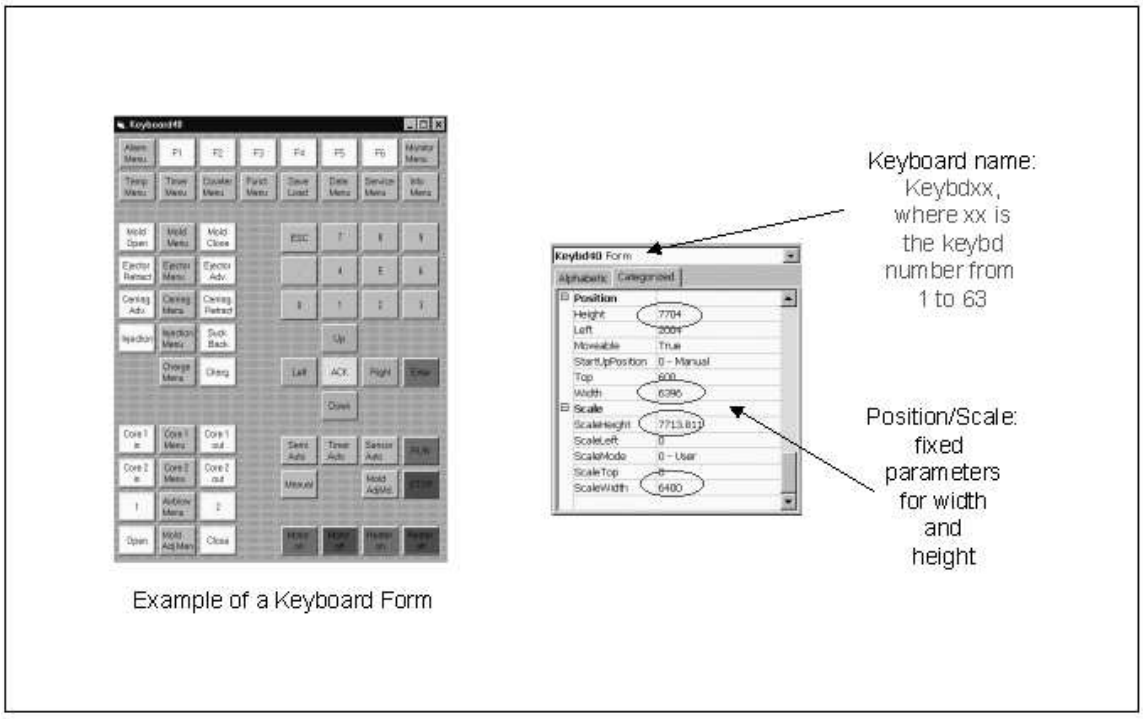

Fig. 2-12, Keyboard Form with properties

Predefined Parameters for keyboards with example values are:

Parameter	Comment	Default
FormKeyboardnumber = 1	Number of Menu	[0]

Table 2-7, Parameter Keyboard Forms

The values in [ ] indicate the default value, when the parameter is not present.

Notes

The keyboard's effective number is extracted from the "FormKeyboardNumber" parameter, the "Tag" field in the VB property window for the Form or from the file name.

Keys

Keys are placed on the keyboards and they are used to define key events. Key names must be Keyxx, where xx is a number between 0 and 70 to match the 71 keys available on the OP15B. Key names for the OP15B are predefined from Key0 to Key70, line by line, to allow the software a clear assignment of every key to a user defined key event. For a good starting point about the naming of the keys, please refer to an example project.

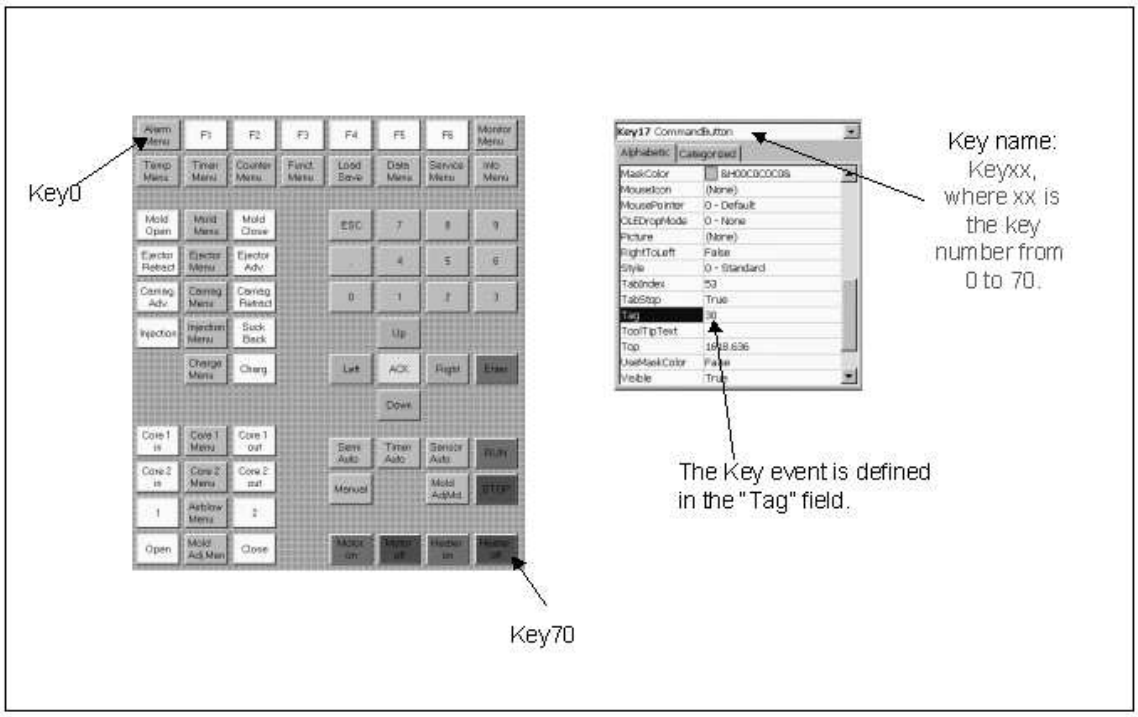

Fig. 2-13, Keyboard Form with Keys and Key properties

Parameter Predefined Parameters for keys with some example values are:

Parameter	Comment	Default
KeyEvent = 13	Event for Key	[0]
KeyCI15 = 1	Valid for CI15	[1]
KeyCI15PLUS = 1	Valid for CI15+	[1]
KeyCI16 = 0	Not valid for CI16	[1]
KeyCI16DP = 0	Not valid for CI16DP	[1]

Table 2-8, Parameter Keyboard Forms

The values in [ ] indicate the default value, when the parameter is not present.

Notes A Key event "0" means that this key has no effect in this menu. Key events "1" to "63" call the menu with the numbers "1" to "63" respective. Key events from "128" up call specific events which are predefined in the OP15B software, e.g. language selection, brightness, etc. See a list of predefined functions for key events below. The KeyEvent parameter overwrites the "Tag" definition in the VB property window. If no "Tag" or "KeyEvent" parameter is defined we fall back to KeyEvent "0" (no action).

2.2.4. Customer Specific Picture

What's a CSP ?

A CSP is displayed at startup of the OP15B instead of the default information from the manufacturer Siemens Numerical Control Ltd., Nanjing. The size of a CSP must 320x240 and can be created, e.g. with Microsoft Paint as a black/white picture. There are no grayscale pictures allowed.

When saved as a Monochrome Bitmap file named "csp.bmp" in the project directory, the Menu Compiler creates a new, compiled "csp.bin" file for uploading to the OP15B with the SIJECT Loader.

3. Menu Compiler (Meco)

3.1. Installation

Meco

The best starting point to design a new set of menus is to use an existing VB project for OP15B and modify it according to your needs. Beside all VB files, the Menu Compiler "meco.exe" and the shared library “cygwin1.dll” must be in the same directory. All data files must be in the subdirectory "data", optional a new Customer Specific Picture "csp.bmp" can be stored in the project directory.

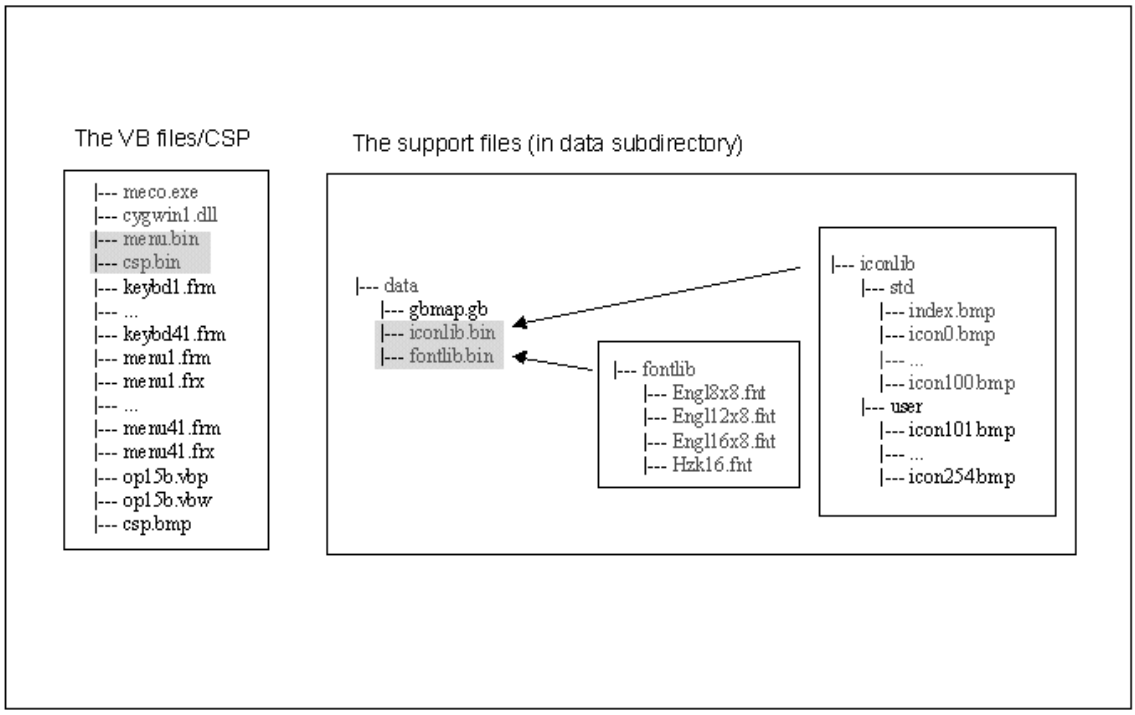

Fig. 3-1, Directory structure and file locations for a project

3.2. Usage

How to run?

The following steps launch the Menu Compiler for your project:
Open a Windows console on your PC.
Change to your project directory.
Type in “meco” and confirm with CR.

Meco Options

Optional command line options for the Menu Compiler are:

```
meco [option] ... [option]


      -h, --help        Invoke this help and exit.
      -i, --iconlib     Create a new iconlib.
      -f, --fontlib     Create a new fontlib.
      -c, --csp         Create a new CSP.
```

New Font Lib

Creating a new font library is necessary only when adding a new Simplified Chinese character to it. In this case, add the new character to the end of the "data\gbmap.gb" file (don’t forget to increment the index number in the first column), delete the "data\fontlib.bin" file and run the Menu Compiler again. It recreates a new, compiled “fontlib.bin".

Or use Meco with the command line option “-f” or “—fontlib”.

New Icon Lib

Edit an icon or add a new icon to the user library subdirectory "data\iconlib\user". Delete the "data\iconlib.bin" file and run the Menu Compiler again. It recreates a new, compiled version of the library file "iconlib.bin".

Or use Meco with the command line option “-i” or “—iconlib”.

New CSP

Delete the file "csp.bin" in the project directory and run Meco again. It uses an existing "csp.bmp" image file to create a new "csp.bin" file which can be uploaded to the OP15B with the SIJECT-Loader.

Or use Meco with the command line option “-c” or “—csp”.

4. Key events and functions

4.1. Predefined Key events

Functions

All functions predefined in the OP15B software are listed in the table below.

Event Number	Action
0	No event, key is locked
1 to 63	Switch to the menu with this particular number
64 to 127	Not used, reserved
128	Call previous menu
129	Manual lubrication
130	Start purging
131	Stop purging
132	Selection
133	Reset shot counter
134	Reset part counter
135	Increase brightness
136	Decrease brightness
137	Save button
138	Load button
139	Not used, reserved.
140	Save program
141	Load program
142	Prop valve increment
143	Prop valve decrement
144	Not used, reserved.
145	Force output
146	Core3 in
147	Core3 out
148	Auto adjust start
149	Auto adjust stop
150	Numerical 0
151	Numerical 1
152	Numerical 2
153	Numerical 3
154	Numerical 4
155	Numerical 5
156	Numerical 6
157	Numerical 7
158	Numerical 8
159	Numerical 9
160	Numerical dot
161	Escape (ESC)
162	Arrow up
163	Arrow down
164	Arrow left
165	Arrow right
166	Acknowledge (ACK)
167	Enter (CR)
168	Run
169	Stop
170	Heater on
171	Heater off
172	Motor on

173	Motor off
174	Manual Mode
175	Mold adjustment
176	Semi Auto Mode
177	Time Auto Mode
178	Sensor Auto Mode
179	Mold left
180	Mold right
181	Ejector left
182	Ejector right
183	Carriage left
184	Carriage right
185	Injection left
186	Injection right
187	Charging screw
188	Core1 left
189	Core1 right
190	Core2 left
191	Core2 right
192	Air blow 1
193	Air blow 2
194	Mold adjustment left
195	Mold adjustment right
196	View PLC version
197	Not used, reserved.
198	Update time
199 to 249	Not used, reserved.
250 to 255	For manufacturing test only, don’t use.

Tab.4-1, Possible key functions

4.2. Communication Parameters

PLC Parameters

All communication parameters predefined in the OP15B software for communication with the PLC are listed in the table below.

Type	Index	Action
&H00&	&H00&	No communication
&H00& to &HE0&	&H00& to 0HFF&	Type and Index for communication, must be agreed between OP15B and CI.
&HFC&	&H00& … &H0F&	1st to 16th packed icons in one communication
	&H10& … &H1F&	2nd batch of 16 icons.
	&H20& … &H2F& … &HF0& … &HFF&	More batches of packed icons.

Table 4-2, PLC communication

OP Parameters

All communication parameters predefined in the OP15B software for communication with the OP are listed in the tables below.

Type	Index	Action
&HFF&	&H00&	No communication
	&H01&	OP version
	&H02&	CI version
	&H03&	PLC version
	&H04&	Recipe number
	&H05&	Temperature compensation value
	&H07&	PLC version on Memory Card
	&HF0&	Password Input
	&HF1&	Password to change password for Level 2
	&HF2&	Password New, Level 2
	&HF3&	Password New Confirm, Level 2
	&HF4&	Password to change password for Level 1
	&HF5&	Password New, Level 1
	&HF6&	Password New Confirm, Level 1

Table 4-3, OP communication with Text Boxes (Variables)

To change the password for Level2, you must use the super user password. To change the password for Level1, you must use the Level2 password or the super user password.

Type	Index	Action
&HFF&	&H00&	No communication
	&H01&	Save / Load Program
	&H02&	Save / Load Recipe
	&H03&	Adjust Valve Command
	&H04&	Adjust Valve Limits
	&H05&	Adjust Valve Type
	&HF0&	Change OP15B language

	&HF1&	Screen Saver on/off

Table 4-4, OP communication with Combo Boxes

Type	**Index**	**Action**
&HFF&	&H00&	No communication
	&H01&	Status PLC switch on CI
	&H02&	Status PLC startup
	&H03&	Status PLC life counter
	&H04&	Status PLC communication
	&H05&	Status CI EEPROM
	&H06&	Status CI type

Table 4-5, OP communication with List Boxes

5. Abbreviations

5.1. List of abbreviations

Abbreviations	CI	Compact Interface
	CSP	Customer Specific Picture
	Meco	Menu Compiler
	OP	Operator Panel
	PLC	Programmable Logic Controller
	PC	Personal Computer
	VB	Visual Basic

<table>
<tr><td>SIEMENS NUMERICAL CONTROL
LTD.,NANJING,CHINA

R&D Division
No.18,Siemens Road, Jiangning Development Zone
211100 NANJING
People Republic of China

Zip code: 211100
Telephone: 025 - 2101888
Fax: 025 - 2101666</td><td>Suggestions

Corrections

For publication/Manual:

SIJECT OP15B Menu Editor
User Manual

Manufacturer Documentation</td></tr>
<tr><td>From

Name</td><td>Technical manual

Order No:

Edition: 06.2003</td></tr>
<tr><td>Company/dept.
street

_____

_____

Phone

_____

Fax

_____</td><td>Should you come across any printing errors when reading this publication, please notify us on this sheet.

Suggestions for improvement are also welcome.</td></tr>
</table>

Suggestions and / or correction: