WAGO-I/O-SYSTEM 750
KNX/EIB/TP1 Module

WagoAppKNX

Version 1.0.2
© 2019 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG
Hansastraße 27
D-32423 Minden
Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69
Email: info@wago.com
Online: http://www.wago.com

Technical Support
Phone: +49 (0) 571/8 87 – 4 45 55
Fax: +49 (0) 571/8 87 – 84 45 55
Email: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

We wish to point out that the software and hardware terms, as well as the trademarks of companies used and/or mentioned in the present document are generally protected by trademark or patent.
Information about This Documentation

Copyright

This documentation, including all figures and illustrations contained therein, is subject to copyright protection. Any use of this documentation that infringes upon the copyright provisions stipulated herein is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying), as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will entail the right of claims for damages. WAGO is a registered trademark of WAGO Verwaltungsgesellschaft mbH.

Number Notation

<table>
<thead>
<tr>
<th>Number System</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal</td>
<td>100</td>
<td>Normal notation</td>
</tr>
<tr>
<td>Hexadecimal</td>
<td>0x64</td>
<td>C notation</td>
</tr>
<tr>
<td>Binary</td>
<td>'100'</td>
<td>In single quotes, nibble separated by a period</td>
</tr>
<tr>
<td></td>
<td>'0110.0100'</td>
<td></td>
</tr>
</tbody>
</table>

Font Conventions

<table>
<thead>
<tr>
<th>Font Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>italic</td>
<td>Names of paths and files are shown in italics, e.g.: C:\Programs\WAGO-I/O-CHECK</td>
</tr>
<tr>
<td>Menu</td>
<td>Menu options are shown in bold, e.g.: Save</td>
</tr>
<tr>
<td>&gt;</td>
<td>A “greater than” symbol between two names denotes the selection of a menu option, e.g.: File &gt; New</td>
</tr>
<tr>
<td>Input</td>
<td>Names of input or selection fields are shown in bold, e.g.: Start of measurement range</td>
</tr>
<tr>
<td>“Value”</td>
<td>Input or selection values are shown in quotation marks, e.g.: Enter the value “4mA” under Start of measurement range.</td>
</tr>
<tr>
<td>[Button]</td>
<td>Button labels within the dialogs are bold and enclosed in square brackets, e.g.: [Input]</td>
</tr>
<tr>
<td>[Key]</td>
<td>Key labels on the keyboard are shown in bold and enclosed in square brackets, e.g.: [F5]</td>
</tr>
</tbody>
</table>
# Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td><strong>Warning against personal injury!</strong>&lt;br&gt;Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.</td>
</tr>
<tr>
<td>!</td>
<td><strong>Warning against personal injury!</strong>&lt;br&gt;Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.</td>
</tr>
<tr>
<td>!</td>
<td><strong>Warning against personal injury!</strong>&lt;br&gt;Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.</td>
</tr>
<tr>
<td>!</td>
<td><strong>Warning against personal injury!</strong>&lt;br&gt;Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.</td>
</tr>
<tr>
<td>!</td>
<td><strong>Warning against damage to property!</strong>&lt;br&gt;Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.</td>
</tr>
<tr>
<td>!</td>
<td><strong>Warning against damage to property caused by electrostatic discharge!</strong>&lt;br&gt;Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.</td>
</tr>
<tr>
<td>!</td>
<td><strong>Important note!</strong>&lt;br&gt;Indicates a potential malfunction, but one which will not result in damage to property if not avoided.</td>
</tr>
<tr>
<td>!</td>
<td><strong>Additional information</strong>&lt;br&gt;Refers to additional information which is not an integral part of this documentation (e.g., the Internet).</td>
</tr>
</tbody>
</table>
Legal Principles

Subject to Change

WAGO Kontakttechnik GmbH & Co. KG reserves the right to make any alterations or modifications that serve the purpose of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from granting patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

Personnel Qualification

The use of the product described in this document is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards.

Moreover, the persons named here must also be familiar with all of the products cited in this document, along with the operating instructions. They must also be capable of correctly predicting any hazards which may not arise until the products are combined.

WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

Limitation of Liability

This documentation describes the use of various hardware and software components in specific example applications. The components may represent products or parts of products from different manufacturers. The respective operating instructions from the manufacturers apply exclusively with regard to intended and safe use of the products. The manufacturers of the respective products are solely responsible for the contents of these instructions.

The sample applications described in this documentation represent concepts, that is, technically feasible applications. Whether these concepts can actually be implemented depends on various general conditions. For example, different versions of the hardware or software components may require different handling than that described here. Therefore, the descriptions contained in this documentation do not form the basis for assertion of a particular product characteristic.

Responsibility for safe use of a specific software or hardware configuration lies with the party that produces or operates the configuration. This also applies if one of the concepts described in this document was used for implementation of the configuration.

WAGO Kontakttechnik GmbH & Co. KG assumes no liability for the realization of these concepts.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>6</td>
</tr>
<tr>
<td>1  Description</td>
<td>7</td>
</tr>
<tr>
<td>2  Material Used</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Required Libraries</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Devices</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Tools</td>
<td>7</td>
</tr>
<tr>
<td>3  Setup</td>
<td>8</td>
</tr>
<tr>
<td>4  Quick Start Guide</td>
<td>9</td>
</tr>
<tr>
<td>5  Application for the KNX/EIB/TP1 Module</td>
<td>12</td>
</tr>
<tr>
<td>5.1 Tasks</td>
<td>12</td>
</tr>
<tr>
<td>5.2 e!COCKPIT Programming</td>
<td>12</td>
</tr>
<tr>
<td>5.2.1 Program</td>
<td>12</td>
</tr>
<tr>
<td>5.2.2 Fixed Memory Addressing for Base Modules</td>
<td>14</td>
</tr>
<tr>
<td>5.2.3 Configuring the Symbol File</td>
<td>15</td>
</tr>
<tr>
<td>5.3 ETS5 Configuration</td>
<td>17</td>
</tr>
<tr>
<td>5.3.1 Integrating KNX Products into the ETS5 Project</td>
<td>17</td>
</tr>
<tr>
<td>5.3.2 Defining Group Addresses</td>
<td>17</td>
</tr>
<tr>
<td>5.3.3 Starting the WAGO ETS Plug-in</td>
<td>18</td>
</tr>
<tr>
<td>5.3.4 Reading in the SYM_XML File</td>
<td>18</td>
</tr>
<tr>
<td>5.3.5 Assigning Group Addresses for the KNX/EIB/TP1 Module</td>
<td>20</td>
</tr>
<tr>
<td>5.3.6 Assigning Physical Address</td>
<td>21</td>
</tr>
<tr>
<td>5.3.7 Transferring the Application Program</td>
<td>22</td>
</tr>
<tr>
<td>6  Prefix and Address Assignment for DPT Function Blocks</td>
<td>23</td>
</tr>
<tr>
<td>6.1 Tasks</td>
<td>23</td>
</tr>
<tr>
<td>6.2 Base Modules for Single Instantiation</td>
<td>23</td>
</tr>
<tr>
<td>6.3 Base Module as Array Declaration</td>
<td>23</td>
</tr>
<tr>
<td>6.4 Base Module as Nested Block</td>
<td>24</td>
</tr>
<tr>
<td>6.5 Connecting Multiple KNX/EIB/TP1 Modules (753-646)</td>
<td>26</td>
</tr>
<tr>
<td>6.5.1 Tasks</td>
<td>26</td>
</tr>
<tr>
<td>6.5.2 e!COCKPIT Programming</td>
<td>26</td>
</tr>
<tr>
<td>6.5.3 WAGO ETS Plug-In</td>
<td>27</td>
</tr>
</tbody>
</table>
1 Description

A connection can be made to a two-wire KNX network with the 753-646 KNX/EIB/TP1 module from the WAGO-I/O-SYSTEM 750. This manual describes the basic procedure for creating a KNX network application using the e!COCKPIT programming software and the WAGO-ETS 5 plug-in.

2 Material Used

2.1 Required Libraries

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WagoAppKNX</td>
<td>Library with function blocks for KNX communication</td>
</tr>
</tbody>
</table>

2.2 Devices

<table>
<thead>
<tr>
<th>Supplier</th>
<th>Quantity</th>
<th>Designation</th>
<th>Item No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAGO</td>
<td>1</td>
<td>PFC100 Controller; 2 x ETHERNET</td>
<td>750-8101</td>
</tr>
<tr>
<td>WAGO</td>
<td>1</td>
<td>4-Channel Digital Input; 24 VDC; 3 ms</td>
<td>750-402</td>
</tr>
<tr>
<td>WAGO</td>
<td>1</td>
<td>4-Channel Digital Output; 24 VDC; 0.5 A</td>
<td>750-504</td>
</tr>
<tr>
<td>WAGO</td>
<td>1</td>
<td>KNX/EIB/TP1 Module</td>
<td>753-646</td>
</tr>
<tr>
<td>WAGO</td>
<td>1</td>
<td>End Module</td>
<td>750-600</td>
</tr>
<tr>
<td>Any</td>
<td>1</td>
<td>KNX Touch Sensor</td>
<td>-</td>
</tr>
<tr>
<td>Any</td>
<td>1</td>
<td>KNX Switch Actuator</td>
<td>-</td>
</tr>
</tbody>
</table>

2.3 Tools

<table>
<thead>
<tr>
<th>Supplier</th>
<th>Designation</th>
<th>Item No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAGO</td>
<td>e!COCKPIT Workstation License</td>
<td>2759-101/1110-2002</td>
</tr>
<tr>
<td>WAGO</td>
<td>WAGO ETS Plug-In</td>
<td>-</td>
</tr>
<tr>
<td>KNX Association</td>
<td>ETS 5 Professional</td>
<td>-</td>
</tr>
</tbody>
</table>
3 Setup

- e.g., PFC100 Controller (750-8101)
- KNX/EIB/TP1 Module (753-646)
  - KNX Node
  - Phys. Addr. 1.1.1

Figure 1: Connection Diagram
4 Quick Start Guide

**Installation of Sample Projects for e!COCKPIT**

Sample programs can be called up from the *e!COCKPIT* Backstage view by clicking the **Updates & Add-ons** button in the navigation bar.

This quick start guide outlines the necessary steps for creating an application with the KNX/EIB/TP1 module. Within the guide, references are made to sections within this application note where a more detailed description can be found.

1. Create *e!COCKPIT* project.

2. Select or integrate the target system.
3. Create the hardware structure in the device structure.

Figure 4: Create Hardware Structure, Scan Modules – e!COCKPIT.

4. In the program structure under the Library Manager menu item, integrate the WagoAppKNX library in the project.
Figure 5: Integrate Library – e!COCKPIT.

5. Add KNX function blocks to the program and address (see Sections 5.2.1 and 5.2.2).

6. Set the configuration for the SYM_XML file (see Section 5.2.3).

7. Compile, log in and start the PLC program. Create the boot project.

8. Create the ETS project.

9. Start the WAGO ETS plug-in (see Section 5.3.3).

10. Import the SYM_XML file (see Section 5.3.4).

11. Close the WAGO ETS plug-in.

12. Drag & drop the network variables onto the group addresses.

13. Set up the communication objects and group addresses if necessary.

14. Write the application program to the KNX module.

15. After downloading, the “E” LED (top right) should light up green, indicating that the module’s device mode is activated.
5 Application for the KNX/EIB/TP1 Module

5.1 Tasks

In the application example Example1_WagoAppKNX ..., a small KNX/TP1 network consisting of a KNX switch actuator, KNX touch sensor and the WAGO KNX/EIB/TP1 module (753-646) are configured. The KNX switch actuator is switched via a digital input from the WAGO-I/O-SYSTEM 750. The KNX touch sensor drives a digital output from the WAGO-I/O-SYSTEM. The programming of this application in e!COCKPIT and the configuration in the WAGO ETS plug-in are described below.

5.2 e!COCKPIT Programming

5.2.1 Program

![KNX Program Code with Master and Two DTP Function Blocks](image)

Figure 6 shows the basic structure for programming a KNX application in e!COCKPIT.

A prerequisite for communicating with KNX/TPI devices is the KNX/EIB/TP1 module (753-646) and the FbKNX_Master function block.
The designation of the KNX/EIB/TP1 module assigned under the device structure must also be indicated at the “I_Port” input of the master module (see Figure 7).

![Gerätestruktur](image)

Figure 7: Hardware Configuration, Module Designations – e!COCKPIT

Both Digital_Output_1 and Digital_Input_1 instances represent the interface for communication with the KNX network. Each module represents a so-called KNX DPT (Data Point Type) and appears in the ETS later as a communication object. All DPTs defined in the KNX standard are available in a WagoAppKNX library as function blocks. Depending on the application, different DPT function blocks are used in programming. In this example, the two 1-bit DPT_Switch and DPT_UpDown types are selected.

All DPT modules have a “dwIndexDPT” input, by which the modules have to be addressed. In addition to addressing, the instance names are given a specific prefix. This prefix for simple module calls is described in Section 5.2.2 below.

If the base modules are declared as an array, nested in a macro or when using multiple KNX/EIB/TP1 modules, special address and prefix assignments must be observed. These assignments are described in Section 6 or in the library description of the WagoAppKNX library.

---

**Note**

**Advanced DTP Modules**

If the advanced FbDPT_xxx_pro modules are used, other settings can be made on the function blocks.

In this case, the “typDPT” module input must be configured. This input allows the RETAIN declaration of the object variables. This makes it possible to initialize individual DPT values to the last value before a supply failure or a reset. If this option is not required, then all inputs of this type can be connected to the same variables without identifying them with the keyword RETAIN.

**Note**

**Data processing!**

The data in the KNX telegrams to the 753-646 module are processed serially. As a result, sending or receiving of telegrams may be delayed when there is a high volume of data traffic. The delay is dependent upon the PLC cycle time and the number of send/receive requests. Data can be exchanged in both directions at the same time (bidirectional).
5.2.2 Fixed Memory Addressing for Base Modules

The DPT modules are addressed by numbering at the “dwIndex_DPT” input. By assigning the address, fixed memory addresses are assigned to the DTP function blocks. In addition to addressing at the “dwIndex_DPT” input, the instance names of the modules are given a specific prefix that is structured as follows:

Note

**Fixed Memory Addressing**

Depending on the instantiation of the base modules, the prefix and address assignment of the modules may need to be adjusted. If the base modules are declared as an array, nested in a macro or when using multiple KNX/EIB/TP1 modules, Section 6 must be observed.

Alternatively, you can also view the prefix and address assignment rules for DTP function blocks in the description of the WagoAppKNX library.

Prefix structure:

Mx_yyy_FBName

Mx – Module index of the KNX/EIB/TP1 module (753-646)

The module index corresponds to the assignment of the “bKNX_Master” input and indicates which module should be linked to the module (see Section 6.5).

yyy – FB index of the DPT module

The FB index numbers for the DPT blocks of the respective module are consecutive and have three digits.

*Example:*

M1_002_UpDown is the second DPT module programmed for the first module (see Figure 8).

![Figure 8: DPT Module with Fixed Memory Addressing](image-url)
5.2.3 Configuring the Symbol File

The WAGO ETS plug-in requires the so-called symbol file to import KNX DTP variables from the e!COCKPIT programming software. All necessary information for forming an assignment between the names of the PLC variables and the associated memory address is located in this file. To write the variables in the symbol file when compiling the PLC program, the file must be created in e!COCKPIT.

To create the symbol configuration, please proceed as follows: Right-click on the required controller application and select the Symbol Configuration menu item (see Figure 9).

![Figure 9: Creating the Symbol Configuration – e!COCKPIT](image)
A settings dialog then appears. At this point, no settings need to be changed; press the [Add] button to close the dialog and to create the symbol configuration.

**Note**

Error-free compilation run is required!
To select variables in the symbol configuration, there can be no errors when compiling. Otherwise, no variables are displayed in the configuration.

Figure 10: Setting Up the Symbol Configuration

After opening the symbol configuration, the following settings can be made (see Figure 10):

1. Press the [Create] button to start the compilation run. Compilation must be free of errors for symbol entries to be displayed.

2. After completing the compilation run successfully, the variables available for selection are displayed in this area. Select the required variables in the tree structure for which symbol entries are to be generated. You can either mark project modules, which results in the associated variables being selected, or you can specifically mark individual variables. As a default setting, it is recommended that you mark all project modules.

3. Press the [Generate Code] button to create a configuration file. The XML symbol file is created in the project directory and named as follows: `<Project Name>..<Controller Name>..<Application Name>.XML`.

**Note**

Symbol configuration is mandatory!
The e/COCKPIT symbol file must be used to transfer the PLC variables to the WAGO ETS plug-in.
5.3 ETS5 Configuration

The three KNX nodes required for this example must be created in the ETS5 Professional software. The device product data are imported into the ETS database for this purpose. The WAGO ETS plug-in for the 753-646 module is included in the WAGO product database. The group addresses for communication between the KNX nodes are also defined in the ETS5 software.

Information Product Database and ETS Plug-in

The WAGO product database can be obtained at www.wago.com.

5.3.1 Integrating KNX Products into the ETS5 Project

Three devices (WAGO KNX module, switch actuator and touch sensor) are created in the topology of the ETS5 software to implement the application described in this document. You can integrate product data into the project via ETS > Catalog > Import… (see Figure 11).

Figure 11: Importing the ETS5 Device Data

5.3.2 Defining Group Addresses

Two group addresses are required for communication between the WAGO KNX/EIB/TP1 module and the switch actuator or touch sensor. The following definition has been made (see Figure 12):

- **The WAGO TP1 module sends** a switching telegram to the switch actuator via the group address 1/0/0.

- **The WAGO TP1 module receives** a switching telegram from the touch sensor via the group address 1/0/1.
5.3.3 Starting the WAGO ETS Plug-in

The plug-in is launched from the “Topology” workspace. First, select the TPI module and click the “Parameters” tab at the bottom. Then press the [Open product-specific parameter dialog] to open the plug-in (see Figure 13).

5.3.4 Reading in the SYM_XML File

As described in Section 5.2.3, the WAGO ETS plug-in requires the so-called symbol file to import KNX DTP variables from e!COCKPIT. All necessary information for forming an assignment between the names of the PLC variables and the associated memory address is located in this file. The PLC variables that are read in are designated as network variables in the plug-in.

When the WAGO ETS plug-in has been started, a dialog opens (see Figure 14), in which the path for the SYM_XML file to be imported must be entered. The setting is also required for which module index (if multiple KNX/EIB/TP1 modules are fitted) is to be imported.
If an SYM_XML file has already been imported, you can also choose whether to overwrite an existing configuration. If the existing configuration will not be overwritten, the variables/addresses are synchronized. In the default setting, the configuration is synchronized.

![Figure 14: Selecting an SYM-XML File](image)

**Note**

**Important note!**

It is recommended that the “Synchronize” function be used. Otherwise, the existing configuration settings will be deleted.

After pressing the [Import] button, the imported variables appear in the “List of network variables” field (see Figure 15).

![Figure 15: List of Network Variables](image)

After synchronizing the network variables, close the plug-in via the menu **File > End.**
5.3.5 Assigning Group Addresses for the KNX/EIB/TP1 Module

The group addresses can be linked to the network variables using the “Drag & Drop” function. The following assignments have been created in this example application:

Table 1: Linking a Group Address to Network Variables

<table>
<thead>
<tr>
<th>Group address</th>
<th>Network variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/0/0</td>
<td>Digital_Output_1</td>
</tr>
<tr>
<td>1/0/1</td>
<td>Digital_Input_1</td>
</tr>
</tbody>
</table>

Taken from ETS5, the group addresses for the switch actuator and touch sensor are assigned to the communication objects (see Figure 16).
5.3.6 Assigning Physical Address

A change in the physical address can be made in the device properties (see Figure 17). These can be accessed via the context menu (see Figure 16) of the device.

Figure 17: Call the device properties

![Figure 17: Call the device properties](image1)

Figure 18: Setting the physical address

**Programming> Physical Address** (see Figure 18) transmits the new address.

Figure 19: Programming the address

![Figure 19: Programming the address](image2)
You will be prompted by the ETS software to press the programming button of the selected device. In this case, actuate the programming button connector of the KNX/EIB/TP1 module to set it to programming mode.

5.3.7 Transferring the Application Program

The device configurations can then be transferred to the devices by means of an application download (see Figure 20). When this action is complete, it should be possible for data to be exchanged between the sensors and actuators.

Communication between KNX objects can then be observed via the group monitor (see Figure 21).
6 Prefix and Address Assignment for DPT Function Blocks

6.1 Tasks

In the application example Example2_WagoAppKNX..., prefix and address assignment should be explained for fixed memory addressing of DTP function blocks. The purpose of fixed addressing is to make SYM_XML synchronization with the WAGO ETS5 plug-in only required when there are changes in the KNX network interface. e!COCKPIT programming of DTP function blocks for various scenarios is described below.

Note Importing the symbol file!
The symbol file must be reimported into the WAGO ETS plug-in and the KNX application only has to be downloaded into the module when there are program changes that affect the KNX modules.

Note Inconsistent data!
When assigning fixed memory addresses, it is not verified that the prefix of the instance name (see Section 5.2.2) matches the function block assignment. If they do not match, inconsistent data is generated.

Note Program preparation!
General programming with all required steps is described in Section 5. The section assumes that you have implemented the steps as outlined.

6.2 Base Modules for Single Instantiation

See Section 5.2.2. – Fixed memory addressing for base modules

6.3 Base Module as Array Declaration

If the DPT blocks are declared as an array, the following algorithm must be followed for assigning the "dwIndex_DPT" input:

\[
dwIndex_DPT = FBindex \times 1,000 + ArrayIndex
\]

The prefix of the instance name is to be assigned as described in Section 5.2.2.

Example:
M1_003_ArraySwitch is a DPT module array. Assignment of the “dwIndex_DPT” input for the array index [0] and [1] is down in Figure 22.

Figure 22: DPT Modules with Fixed Memory Addressing

6.4 Base Module as Nested Block

If multiple DPT modules are nested in one macro, the macro’s tag must be assigned as described in Section 5.2.2. The instance name within the function block are given a specific prefix that is structured as follows:

_zz_ModuleName

zz – Internal module index of the DTP module

The internal module index for the DTP modules are consecutive and have two digits.

Within the macro, the DPT modules used must follow the algorithm below for assigning the “dwIndex_DPT” input:

\[
\text{dwIndex_DPT} = \text{index macro module} \times 1,000,000 + \text{internal module index} \times 1000 + \text{array index}
\]

Example:

The M1_004_NestedSwitch function block is a macro with DPT modules (see Figure 23). The assignment of the “dwIndex_DPT” input for some internal modules is shown in Figure 24.
Figure 23: DPT Macro Invocation in the Main Program

Figure 24: DPT Macro Internal Structure
6.5 Connecting Multiple KNX/EIB/TP1 Modules (753-646)

6.5.1 Tasks

This describes connecting multiple KNX/EIB/TP1 modules (753-646) to a fieldbus controller. This can be done, for example, to connect to independent KNX lines when using the same e!COCKPIT application.

Note

Program preparation!

General programming with all required steps is described in Section 5. The section assumes that you have implemented the steps as outlined.

6.5.2 e!COCKPIT Programming

As described in Section 5.2.1, the FbKNX_Master function block is required for communicating with KNX/TP1 devices. This module must be invoked once for each KNX/EIB/TP1 module (753-646) and assigned via the “bPortKNX” input (see Figure 25). The first KNX/EIB/TP1 module after the controller is given the number one. Subsequent modules are numbered consecutively.

DPT modules are then integrated in the program. The instance names are designated as described in Section 6. In Figure 26, each KNX/EIB/TP1 module (753-646) is assigned to a DTP function block.
6.5.3 **WAGO ETS Plug-In**

Working with the WAGO ETS plug-in is described in Section 5.3. When connecting multiple KNX/EIB/TP1 modules (753-646), ensure that the right module index is set when importing the SYM_XML file (see Figure 27).

![Figure 27: Importing the SYM-XML File](image)