Manual

WAGO Software
LON® Configurator

for the Configuration of the LON® FTT Module
753-648

Version 1.0.0
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.

E-Mail: documentation@wago.com

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 manual are generally protected by trademark or patent.
# Table of Contents

1  Notes about this Documentation ................................................................. 6
   1.1  Scope of Validity ........................................................................... 6
   1.2  Copyright ................................................................................... 6
   1.3  Symbols ..................................................................................... 7
   1.4  Number Notation ........................................................................ 9
   1.5  Font Conventions ....................................................................... 9

2  Important Notes ....................................................................................... 10
   2.1  Legal Bases ................................................................................ 10
   2.1.1  Subject to Changes ................................................................. 10
   2.1.2  Personnel Qualification ......................................................... 10
   2.2  Safety Advice (Precautions) ....................................................... 10
   2.3  Requirements ........................................................................... 12
   2.3.1  PC Hardware ......................................................................... 12
   2.3.2  PC Software .......................................................................... 12
   2.3.3  WAGO-I/O-SYSTEM .............................................................. 13

3  System Configuration and Workflow Overview ....................................... 14
   3.1  General Description ................................................................ 14
   3.2  Configuration Procedure ........................................................... 15
   3.2.1  Step 1: DEFINE LON® interface .......................................... 16
   3.2.2  Step 2: GENERATE code ...................................................... 16
   3.2.3  Step 3: INTEGRATE in user application ............................... 16
   3.2.4  Step 4: INTEGRATE in LON® network ................................. 16
   3.2.5  Step 5: CONNECT to other LON® bus nodes ......................... 17

4  LON® Configurator .................................................................................. 18
   4.1  Preparation ................................................................................ 19
   4.2  Open the LON® Configurator in WAGO-I/O-PRO ...................... 20
   4.3  User Interface Overview ............................................................ 21
   4.4  General Entries .......................................................................... 22
   4.4.1  "Configure Program ID" Dialog ........................................... 23
   4.5  Configuration Area for the LON® Network Interface ................. 25
   4.5.1  Search filter entry .................................................................. 26
   4.5.2  [x] Button ............................................................................. 26
   4.5.3  LonMark® Resources ............................................................... 26
   4.5.4  Arrow Buttons ....................................................................... 27
   4.5.5  LON® Network Interface ....................................................... 28
   4.5.5.1  Add Objects to the Tree Structure ..................................... 29
   4.5.5.2  Context Menu in the Tree Structure of the LON® Network Interface ................................................................. 29
   4.5.5.3  Actions in the Tree Structure .............................................. 31
   4.5.5.4  Symbols in the Tree Structure .......................................... 32
   4.5.5.5  Object Settings ................................................................. 33
   4.5.5.5.1  Settings of a LonMark® Object (LMO) .......................... 33
   4.5.5.5.2  Settings of a Network Variable (NV) .............................. 33
   4.5.5.5.3  Settings of a Configuration Property (CP) ....................... 35
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6</td>
<td>Function / Status Area</td>
</tr>
<tr>
<td>4.7</td>
<td>WAGO-I/O-PRO Function Blocks</td>
</tr>
<tr>
<td>4.7.1</td>
<td>Directory Structure</td>
</tr>
<tr>
<td>4.7.2</td>
<td>Call Sequence</td>
</tr>
<tr>
<td>4.7.3</td>
<td>PLC_PRG [PRG]</td>
</tr>
<tr>
<td>4.7.4</td>
<td>Cross-Module Function Blocks (Shared)</td>
</tr>
<tr>
<td>4.7.5</td>
<td>Module-specific Function Blocks (Module_X)</td>
</tr>
<tr>
<td>4.7.6</td>
<td>Module_X_PRG [PRG]</td>
</tr>
<tr>
<td>4.7.7</td>
<td>Function Blocks in the &quot;ApplicationMessage&quot; Directory</td>
</tr>
<tr>
<td>4.7.8</td>
<td>Function Blocks in the &quot;Internal&quot; Directory</td>
</tr>
<tr>
<td>4.7.9</td>
<td>Function Blocks in the &quot;Master&quot; Directory</td>
</tr>
<tr>
<td>4.7.10</td>
<td>Function Blocks in the &quot;NV_Interface&quot; Directory</td>
</tr>
<tr>
<td>4.7.11</td>
<td>Launch Behavior/Reset</td>
</tr>
<tr>
<td>4.7.12</td>
<td>Launching the PLC Application</td>
</tr>
<tr>
<td>4.7.13</td>
<td>Resetting the Neuron® Controller</td>
</tr>
<tr>
<td>4.7.14</td>
<td>LonWorks® Configuration Properties (CPs)</td>
</tr>
<tr>
<td>4.7.15</td>
<td>Synchronization</td>
</tr>
<tr>
<td>4.7.16</td>
<td>SCPTmaxSendTime / SCTPminSendTime / SCPTmaxRevTime Configuration Properties</td>
</tr>
<tr>
<td>5</td>
<td>Using a LonWorks® Network Management Tool</td>
</tr>
<tr>
<td>5.1</td>
<td>Integrate LON® FTT Module in the LON® Network</td>
</tr>
<tr>
<td>5.2</td>
<td>Integrate a Modified Interface Configuration</td>
</tr>
<tr>
<td>6</td>
<td>Planning Strategies and Restrictions</td>
</tr>
<tr>
<td>6.1</td>
<td>Planning Strategies for Project Creation</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Example with and without ARRAY</td>
</tr>
<tr>
<td>6.1.1.1</td>
<td>Example Project 1</td>
</tr>
<tr>
<td>6.1.1.2</td>
<td>Example Project 2</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Further Optimization by Arrangement</td>
</tr>
<tr>
<td>6.1.2.1</td>
<td>Example Project 3</td>
</tr>
<tr>
<td>6.1.2.2</td>
<td>Example Project 4</td>
</tr>
<tr>
<td>6.1.2.3</td>
<td>Example Project 5</td>
</tr>
<tr>
<td>6.1.2.4</td>
<td>Example Project 6</td>
</tr>
<tr>
<td>6.1.2.5</td>
<td>Example Project 7</td>
</tr>
<tr>
<td>6.2</td>
<td>Restrictions</td>
</tr>
<tr>
<td>7</td>
<td>Example Configuration</td>
</tr>
<tr>
<td>7.1</td>
<td>Example Configuration</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Step 1: DEFINE LON® interface</td>
</tr>
<tr>
<td>7.1.1.1</td>
<td>Launch LON® Configurator</td>
</tr>
<tr>
<td>7.1.1.2</td>
<td>General Settings</td>
</tr>
<tr>
<td>7.1.1.3</td>
<td>Configure Network Interface</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Step 2: GENERATE code</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Step 3: INTEGRATE in user application</td>
</tr>
<tr>
<td>7.1.4</td>
<td>Step 4: INTEGRATE in LON® network</td>
</tr>
<tr>
<td>7.1.4.1</td>
<td>Communication Test</td>
</tr>
<tr>
<td>7.1.5</td>
<td>Step 5: CONNECT to other LON® bus nodes</td>
</tr>
<tr>
<td>8</td>
<td>Glossary</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>9</td>
<td>Appendix</td>
</tr>
<tr>
<td>9.1</td>
<td>Module_X_LonMaster (LON® Master)</td>
</tr>
<tr>
<td>9.2</td>
<td>Module_X_&lt;Name of the LMO&gt;</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Transmission of network output variables</td>
</tr>
<tr>
<td>9.2.2</td>
<td>Receipt of network input variables</td>
</tr>
<tr>
<td>9.2.3</td>
<td>Use of configuration properties</td>
</tr>
<tr>
<td>9.3</td>
<td>ApplicationMessage</td>
</tr>
<tr>
<td>9.3.1</td>
<td>Module_X_ApplicationMsgSend</td>
</tr>
<tr>
<td>9.3.2</td>
<td>Module_X_ApplicationMsgSendRsp</td>
</tr>
<tr>
<td>9.4</td>
<td>FbDecode Modules</td>
</tr>
<tr>
<td>9.4.1</td>
<td>FbDecodeLevPercent</td>
</tr>
<tr>
<td>9.4.2</td>
<td>FbDecodeSnvtSetting</td>
</tr>
<tr>
<td>9.4.3</td>
<td>FbDecodeSnvtStrAsc</td>
</tr>
<tr>
<td>9.4.4</td>
<td>FbDecodeSnvtSwitch</td>
</tr>
<tr>
<td>9.4.5</td>
<td>FbDecodeSnvtTemp</td>
</tr>
<tr>
<td>9.4.6</td>
<td>FbDecodeSnvtTempP</td>
</tr>
<tr>
<td>9.5</td>
<td>FuEncode Modules</td>
</tr>
<tr>
<td>9.5.1</td>
<td>FuEncodeSnvtLevPercent</td>
</tr>
<tr>
<td>9.5.2</td>
<td>FuEncodeSnvtSetting</td>
</tr>
<tr>
<td>9.5.3</td>
<td>FuEncodeSnvtStrAsc</td>
</tr>
<tr>
<td>9.5.4</td>
<td>FuEncodeSnvtSwitch</td>
</tr>
<tr>
<td>9.5.5</td>
<td>FuEncodeSnvtTemp</td>
</tr>
<tr>
<td>9.5.6</td>
<td>FuEncodeSnvtTempP</td>
</tr>
</tbody>
</table>

List of Figures ..................................................................................................... 106

List of Tables ...................................................................................................... 108
1 Notes about this Documentation

Note
Keep this documentation!
The operating instructions are part of the product and shall be kept for the entire lifetime of the device. They shall be transferred to each subsequent owner or user of the device. Care must also be taken to ensure that any supplement to these instructions are included, if applicable.

1.1 Scope of Validity

This documentation applies to the software of the LON® configurator.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions 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 involve the right to assert damage claims.
1.3 Symbols

- **DANGER**
  Personal Injury!
  Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

- **DANGER**
  Personal Injury Caused by Electric Current!
  Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

- **WARNING**
  Personal Injury!
  Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

- **CAUTION**
  Personal Injury!
  Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

- **NOTICE**
  Damage to Property!
  Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

- **NOTICE**
  Damage to Property Caused by Electrostatic Discharge (ESD)!
  Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

- **Note**
  Important Note!
  Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.
**Information**

**Additional Information:**
Refers to additional information which is not an integral part of this documentation (e.g., the Internet).
1.4  Number Notation

Table 1: Number Notation

<table>
<thead>
<tr>
<th>Number code</th>
<th>Example</th>
<th>Note</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 quotation marks, nibble separated with dots (.)</td>
</tr>
<tr>
<td></td>
<td>'0110.0100'</td>
<td></td>
</tr>
</tbody>
</table>

1.5  Font Conventions

Table 2: Font Conventions

<table>
<thead>
<tr>
<th>Font type</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>italic</td>
<td>Names of paths and data files are marked in italic-type. e.g.: C:\Programme\WAGO-I/O-CHECK</td>
</tr>
<tr>
<td>Menu</td>
<td>Menu items are marked in bold letters. e.g.: Save</td>
</tr>
<tr>
<td>&gt;</td>
<td>A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File &gt; New</td>
</tr>
<tr>
<td>Input</td>
<td>Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range</td>
</tr>
<tr>
<td>“Value”</td>
<td>Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range.</td>
</tr>
<tr>
<td>[Button]</td>
<td>Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]</td>
</tr>
<tr>
<td>[Key]</td>
<td>Keys are marked with bold letters in square brackets. e.g.: [F5]</td>
</tr>
</tbody>
</table>
2 Important Notes

This section describes the legal principles and system requirements for using the software in compliance with intended purpose, underlying provisions and stated specifications.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of 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.

2.1.2 Personnel Qualification

Any steps related to the use of WAGO software may only be performed by qualified employees with sufficient knowledge of handling the respective PC system used.

Steps in which files are created or changed on the PC system may only be performed by qualified employees with sufficient knowledge in the administration of the PC system used in addition to the aforementioned.

Steps in which the behavior of the PC system in a network is changed may only be performed by qualified employees with sufficient knowledge in the administration of the network used in addition to the aforementioned.

The descriptions below require knowledge in the configuration of LON® network components.

2.2 Safety Advice (Precautions)

Note

Use up-to-date security software!
Secure operation of the PC system can be at risk as a result of malware such as viruses and Trojans, as well as related threats such as denial-of-service attacks. Therefore, make sure that the latest security software and definitions are always installed on the PC system.
**Note**

Disable or uninstall software that is no longer required!

The vulnerability of a PC system against malware and related threats increases with the number of installed or active software components (applications and services). Therefore, uninstall or disable software components that are not needed for the purpose at hand.
2.3 Requirements

2.3.1 PC Hardware

Table 3: PC hardware requirements

<table>
<thead>
<tr>
<th>Components</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Windows XP/Vista/Windows 7</td>
</tr>
<tr>
<td>Memory</td>
<td>Min. 128 MB</td>
</tr>
<tr>
<td>Free hard disk storage</td>
<td>Min. 1.5 MB for the LON® configurator and 280 MB (x86) or 610 MB (x64) for the .NET 4.0 Framework</td>
</tr>
<tr>
<td>Processor</td>
<td>Min. 500 MHz</td>
</tr>
<tr>
<td>Other</td>
<td>Installed network card, .NET 4.0 Framework (redistributable included in the delivery), standard Web browser with Java support</td>
</tr>
</tbody>
</table>

2.3.2 PC Software

Table 4: Required software

<table>
<thead>
<tr>
<th>Components</th>
<th>Source (item No.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAGO-I/O-PRO</td>
<td>WAGO 759-333</td>
</tr>
<tr>
<td>(includes LON® configurator</td>
<td>(manual for WAGO-I/O-PRO can be downloaded free at: <a href="http://www.wago.com">www.wago.com</a>).</td>
</tr>
<tr>
<td>Version 2.3.9.35 or higher,</td>
<td></td>
</tr>
<tr>
<td>Existing customers with older</td>
<td></td>
</tr>
<tr>
<td>versions are asked to contact</td>
<td></td>
</tr>
<tr>
<td>Support:</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:support@wago.com">support@wago.com</a></td>
<td></td>
</tr>
<tr>
<td>LonMark® resource files</td>
<td>LonMark® Organization (free download at: <a href="http://www.lonmark.org/technical_resources/resource_files/">http://www.lonmark.org/technical_resources/resource_files/</a>)</td>
</tr>
<tr>
<td>(Version 13.00)</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Recommended: Use LMRF Version 13.00!</td>
</tr>
<tr>
<td></td>
<td>We recommend that you do not use any beta, but LMRF Version 13.00 to ensure compatibility.</td>
</tr>
<tr>
<td>LON_01.lib</td>
<td>WAGO (free download at: <a href="http://www.wago.com">www.wago.com</a>)</td>
</tr>
<tr>
<td></td>
<td>(library description for LON_01.lib free download at: <a href="http://www.wago.com">www.wago.com</a>).</td>
</tr>
</tbody>
</table>

Table 5: Optional software

<table>
<thead>
<tr>
<th>Components</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAGO-I/O-CHECK</td>
<td>WAGO (759-302)</td>
</tr>
<tr>
<td>WAGO ETHERNET Settings</td>
<td>WAGO (free download at: <a href="http://www.wago.com">www.wago.com</a>)</td>
</tr>
</tbody>
</table>
### WAGO-I/O-SYSTEM

Table 6: Required Components of the WAGO-I/O-SYSTEM

<table>
<thead>
<tr>
<th>Components</th>
<th>Source (item No.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fieldbus controller/PLC</td>
<td>WAGO (750-830, example)</td>
</tr>
<tr>
<td>WAGO-I/O-SYSTEM 750 (e.g. BACnet/IP Programmable Fieldbus Controller 750-830)</td>
<td></td>
</tr>
<tr>
<td>LON® FTT module 753-648</td>
<td>WAGO (753-648)</td>
</tr>
<tr>
<td>End module 750-600</td>
<td>WAGO (750-600)</td>
</tr>
</tbody>
</table>
3 System Configuration and Workflow Overview

3.1 General Description

By configuring the LON® FTT module 753-648 using the LON® configurator, it is possible to connect and access a LonWorks® network in any fieldbus system of the WAGO-I/O-SYSTEM 750.

![Figure 1: Principle of the LonWorks® network connection to the WAGO-I/O-System 750](image)

The interface of the LON® FTT module 753-648 to the LonWorks® network is not restricted to specific network variables. The LON® configurator can be used to configure the LON® FTT module 753-648 and create a customized LON® interface tailored specifically to the required functionality.

The LON® configurator in WAGO-I/O-PRO is used to define the LON® network interface for connecting LON® to the IEC-61131-3 user application for the WAGO-I/O-SYSTEM. The appropriate function blocks are automatically generated from the interface definition created. The WAGO-I/O-PRO can then be used to integrate the function blocks in your IEC user application.

A suitable LonWorks® network management tool is used to configure use of the defined LON® interface, i.e. the connection to the connected LonWorks® network.
Please note the limits and strategies of project planning!
Be sure to note the factors described in the "Planning Strategies" and "Restrictions" chapters before planning and executing your own project to ensure optimal function of your project with the LON® FTT module 753-648.

LON® configurator is not used to set up the LonWorks® network!
Please note that the LON® Configurator is used to configure the network interface of the LON® FTT module 753-648 and not used to set up the connected LonWorks® network or communication between the respective nodes.
A network management tool (e.g. LonMaker™ integration tool from Echelon® or similar) is used to design, set up, manage and maintain the LonWorks® network!

3.2 Configuration Procedure

The process for configuring the interface of the LON® FTT module consists of five different steps in various tools.
It is important to note the difference between the definition (in the LON® configurator and WAGO-I/O-PRO) and use of the LON® interface (in a LonWorks® network management tool).

<table>
<thead>
<tr>
<th>LON® interface</th>
<th>Tool</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td></td>
<td><strong>Step 1:</strong> DEFINE LON® interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Step 2:</strong> GENERATE code</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Step 3:</strong> INTEGRATE in user application</td>
</tr>
<tr>
<td>Use</td>
<td></td>
<td><strong>Step 4:</strong> INTEGRATE in LON® network</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Step 5:</strong> CONNECT to other LON® bus nodes</td>
</tr>
</tbody>
</table>
Figure 2: Schematic representation of the definition and use of the LON® network interface

The basic process is briefly outlined in the next chapter (see Step 1 to 5). A specific implementation of these five steps is described in the "Example Configuration" chapter.

Details and information about the software used is available in the chapters "LON® Configurator", "WAGO-I/O-PRO Function Blocks" and "Using a LonWorks® Network Management Tool".

3.2.1 **Step 1: DEFINE LON® interface**

The LON® configurator launched from the WAGO-I/O-PRO directly is used to define the LON® network interface.

The LON® configurator offers you a simple graphical interface to all LonMark® resources. From this interface, you can select the required functionality and apply it to the LON® network interface by drag-and-drop or button click. You can also specify these LonMark® resources by setting parameters.

3.2.2 **Step 2: GENERATE code**

Once you have defined the functionality that the LON® FTT module requires, clicking the [Generate code] button automatically generates the defined variables and LonMark® objects as IEC function blocks.

3.2.3 **Step 3: INTEGRATE in user application**

The generated IEC function blocks can then be integrated in WAGO-I/O-PRO in your project directly by connecting them to the inputs and outputs of your user application.

3.2.4 **Step 4: INTEGRATE in LON® network**

A LON® network management tool (LNMT or NMT) is used to integrate the LON® FTT module in the connected LonWorks® network.
The interface definition of the LON® FTT module can be imported in the NMT using an XIF file generated in WAGO-I/O-PRO or (if online) downloaded from the LON® FTT module directly.

### 3.2.5 Step 5: CONNECT to other LON® bus nodes

Then use the respective LonWorks® network management tool for binding during which you establish the connection of the network variables of the LON® FTT module with network variables of other devices for bus communication.
4 LON® Configurator

The comprehensive, easy-to-use software tool, the LON® configurator integrated in WAGO-I/O-PRO, is used to define the LON® network interface within the LON® FTT module 753-648. This forms the communication link between the WAGO-I/O-SYSTEM and the "IEC-61131-World" and the LonWorks® network.

For integration in the user program after successful definition of the LON® network interface, IEC-61131-3 program blocks are automatically created in the IEC application, which serve as representation of the LON® network interface in the IEC application.

When starting the controller, the LON® network interface and configuration data are automatically downloaded to the LON® FTT module and thus make communication possible.

To define the LON® network interface, you can select any type of network variable. In addition to standard network variable types (SNVTs) and standard configuration property types (SCPTs), user-defined types (UNVTs/UCPTs) as well as LonMark® functional profiles (FPTs) are also supported. Types and objects of the LonMark® resources installed on the PC are used for the network variable definition.

An external interface file (XIF) is created for offline configuration in a network management tool.

---

**Information**

More information about WAGO-I/O-PRO!
You can obtain the WAGO-I/O-PRO software on a CD under Item No. 759-333. This CD contains all the application program files.
You can download the WAGO-I/O-PRO software from the WAGO website at: [http://www.wago.com](http://www.wago.com).
4.1 Preparation

A prerequisite for the chapters described in the following for configuration with the LON® configurator is that the hardware of your fieldbus node and the LON® network are properly structured and their functions are error-free.

For communication with the LON® FTT module via the LON® configurator, there must already be a communication link from your PC to your fieldbus node for online configuration via the fieldbus or serially via the service interface of the programmable fieldbus controller.

In addition, the WAGO-I/O-PRO software (version with integrated LON® configurator) must be installed on your PC.

**Note**

Please note the version of WAGO-I/O-PRO with integrated LON® configurator!

Please note that the LON® configurator is integrated in a version of WAGO-I/O-PRO that is higher than V 2.3.9.35.

If you have an older version of WAGO-I/O-PRO, please contact Support by email: support@wago.com.

A current version of the "LON_01.lib" library must be saved on your PC in the following folder:

"C:\Program Files\WAGO Software\CoDeSys V2.3\Targets\WAGO\Libraries\Building".

**Note**

Important note about launching WAGO-I/O-PRO!

Please note the following when launching WAGO-I/O-PRO:

- Do not use an existing project file, e.g. double-click, to launch WAGO-I/O-PRO. Code generation is not possible otherwise.
- You can open only one instance of WAGO-I/O-PRO simultaneously.
- The "-remote" command-line argument has to be used to launch WAGO-I/O-PRO to add modules in WAGO-I/O-PRO.

*(Example: "<Installation_Path>\WAGO Software\CoDeSys V2.3\Codesys.exe" –remote)*

After creating a new project in WAGO-I/O-PRO and configuring the hardware, you can open the LON® configurator and start configuring the LON® FTT module.
4.2 Open the LON® Configurator in WAGO-I/O-PRO

1. Open the LON® configurator for the LON® FTT module 753-648 in WAGO-I/O-PRO directly. Select the "Resources" tab.

2. In the tree structure, double-click on the "PLC Configuration" entry to open the dialog to configure the controller.

3. In this dialog, expand the tree structure for the entries "Hardware configuration" and "K-Bus".

4. Then select the "0753-0648 LON[VAR]" entry.

   In the right area of the dialog, the configuration dialog for the LON® FTT module appears. It contains the following tabs:
   - "Configure LON® interface"
   - "Description".

   Figure 3: Screenshot of the WAGO-I/O-PRO PLC configuration

5. In this configuration dialog, select the "Configure LON® interface" tab.

   The user interface for defining the internal LON® network interface appears.

---

**Note**

Delay due to LMRF loading!
When selecting the entry for the LON® FTT module designation, please note that the LonMark® (LMRF) resource files are first opened in the background, which can take a few seconds.
4.3 User Interface Overview

Figure 4: Configure LON® interface user interface tab with default settings

The structure of the user interface for the "Configure LON® interface" is divided horizontally into three main areas:

1. General entries
2. Configuration area for the LON® network interface
3. Function / status area

These areas will be explained in more detail in the following chapters.
4.4 General Entries

You can enter general entries for your LON® network interface configuration in the top area of the dialog.

![General entries](image)

Figure 5: General entries

Table 8: "General entries" area

<table>
<thead>
<tr>
<th>Function</th>
<th>Value (example)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program ID:</td>
<td>90:00:70:05:01:06:04:01</td>
<td>Program ID in hexadecimal numbers (0-9 and A-F). You can manually enter the ID in this field. The input of colons is not mandatory. You can also open a help dialog for the entry by clicking the [...] button.</td>
</tr>
<tr>
<td>[...] button</td>
<td>-</td>
<td>This button is used to open the help dialog for generating a program ID (&quot;Configure Program ID&quot; dialog → see the &quot;Configure Program ID&quot; dialog in the following chapter).</td>
</tr>
<tr>
<td>Neuron firmware:</td>
<td>19</td>
<td>Neuron version number</td>
</tr>
<tr>
<td>Module number:</td>
<td>1</td>
<td>Input of any module number for the LON® FTT-module (value range 1 - 999).</td>
</tr>
<tr>
<td>Device template name:</td>
<td>NameOfMyLonInterface</td>
<td>Name of the device template file XIF (.XIF interface file). The XIF ending is automatically generated when saving.</td>
</tr>
<tr>
<td>Storage location for device template:</td>
<td>c:\StorageLocation\LonModule</td>
<td>Manual input of the storage directory for the file entered under &quot;Device template name:&quot;.</td>
</tr>
<tr>
<td>Self-documentation string comment:</td>
<td>The comment in the XIF file</td>
<td>Comment that should be attached to the end of the self-documentation string. Input in this field is not mandatory.</td>
</tr>
</tbody>
</table>

Note:
- **Module number must be unique!**
  Please note that the module number within a WAGO-I/O-PRO project must be unique!

- **Specify the storage!**
  Specify a directory as the storage location for the device template. Otherwise, an error message will appear when generating the code.
4.4.1 "Configure Program ID" Dialog

You can open the "Configure Program ID" dialog by clicking the [...] button to make setting the program ID convenient.

Doing so allows you to select the required settings from dropdown lists.

![Configure Program ID dialog](image)

Figure 6: "Configure Program ID" dialog

Depending on the selected settings, the program ID appears as a hexadecimal value in the bottom most field "Program ID [FM:MM:MM:CC:CC:UU:TT:NN]".

After clicking [OK] to confirm the settings, the specified program ID is then transferred to the user interface of the "Configure LON® interface" tab, "General Entries" area to the "Program ID:" field.
### Table 9: "Configure Program ID" dialog

<table>
<thead>
<tr>
<th>Function</th>
<th>Value (example)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>WAGO 112:</td>
<td>Selection of the manufacturer ID</td>
</tr>
<tr>
<td>Category</td>
<td>&lt;&lt;&lt;&lt;All&gt;&gt;&gt;&gt;</td>
<td>Filter for the device class selection &quot;Device class&quot;.</td>
</tr>
<tr>
<td>Device class (CC:CC):</td>
<td>Multi-I/O module (5.01)</td>
<td>Selection of the device class</td>
</tr>
<tr>
<td>Usage (UU):</td>
<td>Network Management (0)</td>
<td>Selection of the location of use</td>
</tr>
<tr>
<td>Communication channel (TT):</td>
<td>TP/FT-10(4)</td>
<td>Display of the communication channel. This information cannot be changed for the LON® FTT module.</td>
</tr>
<tr>
<td>Model number (NN):</td>
<td>1</td>
<td>Module number entry (value range: 0 - 255).</td>
</tr>
<tr>
<td>Certified interface (F)</td>
<td>☑</td>
<td>The program ID should be generated for a LonMark®-certified device template.</td>
</tr>
<tr>
<td></td>
<td>☐</td>
<td>The program ID should be generated for a non-LonMark®-certified device template (default).</td>
</tr>
<tr>
<td>Program ID (FM:MM:MM:CC:CC:UU:TT:NN)</td>
<td>90:00:70:05:01:06:04:01</td>
<td>Displays the program generated by the selection made as hexadecimal values or manual input is also possible.</td>
</tr>
<tr>
<td>[OK] button</td>
<td></td>
<td>Closes the dialog and applies the respective values for the program ID. The ID is then automatically transferred to the &quot;Configure LON® interface&quot; tab, &quot;General Entries&quot; area to the &quot;Program ID:&quot; field.</td>
</tr>
<tr>
<td>[Cancel] button</td>
<td></td>
<td>Closes the dialog, entries are discarded.</td>
</tr>
</tbody>
</table>
4.5 Configuration Area for the LON® Network Interface

The input elements for the definition of the LON® network interface is in the middle of the dialog.

![Configuration area of the LON® network interface (example view)](image)

Figure 7: Configuration area of the LON® network interface (example view)

This area of the dialog is again functionally divided into six sub-areas. A broad overview of these sub-areas is provided in the table below in which the functions of the individual areas are listed.

More details about these areas are provided in the subsequent chapters.

Table 10: Area of the dialog for creating the LON® network interface

<table>
<thead>
<tr>
<th>Sub-area</th>
<th>Description</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Search filter entry</td>
<td>The search filter entry can be used to search for a specific element of the LonMark® resources available in sub-area 3.</td>
</tr>
<tr>
<td>2.</td>
<td>[X] button</td>
<td>Clicking the [X] button resets the search filter in sub-area 1.</td>
</tr>
<tr>
<td>3.</td>
<td>LonMark® resources</td>
<td>In this sub-area, all available elements of the LonMark® resources appear in a tree structure.</td>
</tr>
<tr>
<td>4.</td>
<td>Arrow buttons (to move, add, delete objects)</td>
<td>The respective arrow buttons can be used to add an element selected from the LonMark® resources (sub-area 3) to the LON® network interface (sub-area 5) and to move an element or to delete an element from the LON® network interface.</td>
</tr>
<tr>
<td>5.</td>
<td>LON® network interface</td>
<td>This sub-area lists all objects selected for definition of the LON® network interface.</td>
</tr>
<tr>
<td>6.</td>
<td>Object settings</td>
<td>Depending on the object selected in the LON® network interface (sub-area 5), specific object settings can be made in this sub-area.</td>
</tr>
</tbody>
</table>
4.5.1 Search filter entry

A text box in the top sub-area (sub-area 1) can be used to search for a data type. When entering free form text, the designations of the LonMark® resources (sub-area 3) displayed below are filtered or listed accordingly. Case does not matter.

![Search filter entry example](image)

Figure 8: Example of a search filter entry with the string "sett"

4.5.2 [x] Button

The `x` button is used to clear the content of the search filter. All LonMark® resources are then displayed.

4.5.3 LonMark® Resources

The LonMark® resources installed on your PC appear on the left side in the sub-area for LonMark® resources (sub-area 3). Both standard and user-defined resources are supported.

The standard resources are summarized under the "Standard" entry and user-defined resources are listed in their own entry (e.g. under "echelon").

The resources are divided into data types for network variables [NVTs], configuration properties [CPTs] and function profiles [FPTs].

The number after the data type designation indicates the index (ID) of the data type.

After selecting the required data type, you can add it to the LON® network interface in one of three ways:

- Double-click the object (NVT, CPT or FPT)
- Drag & drop in the tree structure of the LON® network interface
- Click the `button`
### 4.5.4 Arrow Buttons

Table 11: Arrow buttons

<table>
<thead>
<tr>
<th>Arrow button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Right Arrow" /></td>
<td>Click the &quot;right arrow&quot; to add a selected data type from the LonMark® resources to the LON® network interface.</td>
</tr>
<tr>
<td><img src="image" alt="Up Arrow" /></td>
<td>Click the &quot;up arrow&quot; to move the data type selected in the tree structure of the LON® network interface up. NVs/CPs are moved within the same function profile.</td>
</tr>
<tr>
<td><img src="image" alt="Down Arrow" /></td>
<td>Click the &quot;down arrow&quot; to move the data type selected in the tree structure of the LON® network interface down. NVs/CPs are moved within the same function profile.</td>
</tr>
<tr>
<td><img src="image" alt="Left Arrow" /></td>
<td>Click the &quot;left arrow&quot; to remove the data type selected in the tree structure of the LON® network interface from the LON® network interface.</td>
</tr>
</tbody>
</table>
4.5.5 **LON® Network Interface**

The sub-area of the LON® network interface (sub-area 5) displays the added objects for definition of the LON® network interface in a tree structure.

The LON® network interface of the LON® FTT module is defined and specified by the communication objects implemented here. These include:

- Number, types and designations of the LonMark® objects
- Number, types, directions, connection attributes and designations of the network variables

When first opening the user interface of the LON® configurator, two objects have already been created in the tree structure of this sub-area:

- "SFPTnodeObject" ("Node Object")
- "Virtual Functional Block" (VFB)

The "SFPTNodeObject" forms the basis of the "Node Object" function profile type.

The "Virtual Functional Block" contains the NVs and CPs that can be assigned to the LON® FTT module directly and no other LonMark® object of the LON® FTT module.

---

**Note**

"**Node Object**" is a fixed object at the start!  
Please note that the "Node Object" cannot be deleted. This object is always in the first position.

---

**Note**

"**Virtual Functional Block**" (VFB) is a fixed object at the end!  
Please note that the "Virtual Functional Block" object cannot be deleted. This object is always in the last position. The designation of the "Virtual Functional Block" cannot be modified.
4.5.5.1 Add Objects to the Tree Structure

As shown in the table below, the LON® network interface objects with the selected data type can be added.

<table>
<thead>
<tr>
<th>Action</th>
<th>Element</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double-click or [↩] button</td>
<td>FPT in the LonMark® resource selection</td>
<td>The element is added to the end of the tree structure for the LON® network interface before the VFB.</td>
</tr>
<tr>
<td></td>
<td>NVT / CPT in the LonMark® resource selection</td>
<td>The element is added to the objected selected in the tree structure for the LON® network interface.</td>
</tr>
<tr>
<td>Drag &amp; Drop</td>
<td>FPT in the LonMark® resource selection on an open area of the tree structure for the LON® network interface.</td>
<td>The element is added to the end of the tree structure for the LON® network interface before the VFB.</td>
</tr>
<tr>
<td></td>
<td>FPT in the LonMark® resource selection between LMOs of the tree structure for the LON® network interface.</td>
<td>The element is inserted between the LMOs.</td>
</tr>
<tr>
<td></td>
<td>NVT / CPT in the LonMark® resource selection on an open area of the tree structure for the LON® network interface.</td>
<td>The element is added to the VFB.</td>
</tr>
<tr>
<td></td>
<td>NVT/CPT in the LonMark® resource selection on an LMO in the tree structure for the network interface.</td>
<td>The element is added to the LMO.</td>
</tr>
</tbody>
</table>

FPT: Functional Profile Template  
LMO: LonMark® Object  
VFB: Virtual Functional Block  
NV: Network Variable  
CP: Configuration Properties

4.5.5.2 Context Menu in the Tree Structure of the LON® Network Interface

For the function profile type entry (SFPT...) in the tree structure of the LON® network interface, the contained elements can be further processed. A custom context menu is available by selecting an entry and right-clicking.
Figure 9: Context menu in the tree structure of the LON® network interface

Table 13: Context menu in the tree structure of the LON® network interface

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add optional NV</td>
<td>Adds a specific NV to an LMO. NVs defined in the underlying FPT are displayed. Variables already added are grayed out.</td>
</tr>
<tr>
<td>Add optional CP</td>
<td>Adds a CP to an LMO. CPs defined in the underlying FPT are displayed. Configuration properties already added are grayed out.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the element from the LON® network interface</td>
</tr>
<tr>
<td>Open All</td>
<td>Expands all entries in the tree structure.</td>
</tr>
<tr>
<td>Close All</td>
<td>Minimizes all entries in the tree structure.</td>
</tr>
</tbody>
</table>

This menu item is only available for LMOs.
4.5.5.3 Actions in the Tree Structure

Apart from the context menu, other actions are possible in the tree structure of the LON® network interface:

Table 14: Actions for elements in the tree structure of the LON® network interface

<table>
<thead>
<tr>
<th>Action</th>
<th>Element</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>[←] button</td>
<td>LMO/NV/CP in the tree structure of the LON® network interface</td>
<td>The element is removed.</td>
</tr>
<tr>
<td>Drag &amp; Drop</td>
<td>LMO from the tree structure of the LON® network interface between two function blocks in the tree structure of the LON® network interface</td>
<td>LMO is moved.</td>
</tr>
<tr>
<td></td>
<td>NV / CP from the tree structure of the LON® network interface to another position within the same LMO</td>
<td>NV/CP is moved.</td>
</tr>
<tr>
<td>[↑], [↓] buttons</td>
<td>LMO selected in the tree structure of the LON® network interface</td>
<td>LMO is moved up or down accordingly.</td>
</tr>
<tr>
<td></td>
<td>NV / CP selected in the tree structure of the LON® network interface</td>
<td>NV/CP is moved up or down within the same LMO.</td>
</tr>
<tr>
<td>Context menu</td>
<td>On LMO in the tree structure of the LON® network interface</td>
<td>Adding an NV/CP in the tree structure of the LON® network interface</td>
</tr>
<tr>
<td></td>
<td>On NV / CP in the tree structure of the LON® network interface</td>
<td>Removing the LMO</td>
</tr>
</tbody>
</table>

LMO: LonMark® Object  
VFB: Virtual Functional Block  
NV: Network Variable  
CP: Configuration Properties

The following rules generally apply:

- The elements are created with the name of the data type of the resource when adding.

- When adding elements, a counter is added to the name if there is already an element of the same name.

- If an LMO is added to the end of the tree structure, it is inserted above the VFB.

- The required network variables and configuration properties of an LMO are immediately inserted and flagged.

- Required network variables and configuration properties of an LMO cannot be deleted.

- If an NV/CP without selected element is added to the tree structure, it is inserted in the VFB.
• If an LMO, an NV or a CP is moved within the defined LON® network interface, the respective indices of the object change. The order of the LMOs, NVs and/or CPs also change in the defined LON® network interface and the XIF file created.

4.5.5.4 Symbols in the Tree Structure

The following symbols are used to identify the elements in the tree structure:

Table 15: Symbols in the tree structure of the LON® network interface

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Symbol]</td>
<td>LonMark® Object (LMO)</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>LonMark® Object, not removable</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Network Variable (NV)</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Network variable, required variable of an FPT, not removable</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Network variable, optional variable of an FPT</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Configuration Property (CP)</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Configuration property, required configuration property of an FPT, not removable</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Configuration property, optional configuration property of an FPT</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Configuration property created as an NV</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Configuration property created as an NV, required configuration property of an FPT, not removable</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Configuration property created as an NV, optional configuration property of an FPT</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Additional symbol for NV, direction input (NVI)</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Additional symbol for NV, direction output (NVO)</td>
</tr>
</tbody>
</table>
4.5.5.5 **Object Settings**

If a LonMark® Object (LMO), Network Variable (NV) or Configuration Property (CP is selected in the tree structure of the LON® network interface, the possible settings for the respective element type appear to the right of the tree structure.

### 4.5.5.5.1 Settings of a LonMark® Object (LMO)

![Screenshot of the object settings (example)](image)

**Figure 10:** Screenshot of the object settings (example)

<table>
<thead>
<tr>
<th>Function</th>
<th>Value (example)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input text box</td>
<td>MyCLAcuator</td>
<td>Entry of the name of the LMO, max. 16 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>This option is not available for the VFB.</strong></td>
</tr>
<tr>
<td>Data type indicator</td>
<td></td>
<td>Information about the underlying FPT.</td>
</tr>
<tr>
<td>Element count</td>
<td>✓</td>
<td>Create the LMO as a field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>This option is not available for the Node Object and VFB.</strong></td>
</tr>
<tr>
<td>Element count selection list</td>
<td>1</td>
<td>Number of elements of the field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Available if &quot;Element count&quot; is activated.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>This option is not available for the Node Object and VFB.</strong></td>
</tr>
</tbody>
</table>

### 4.5.5.5.2 Settings of a Network Variable (NV)

![Screenshot of the settings of a network variable (example)](image)

**Figure 11:** Screenshot of the settings of a network variable (example)
Table 17: Settings of a network variable (NV)

<table>
<thead>
<tr>
<th>Function</th>
<th>Value (example)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input text box</td>
<td></td>
<td>Entry of the name of the NV, max. 16 characters.</td>
</tr>
<tr>
<td>Data type indicator or Selection of a data type</td>
<td></td>
<td>Information about the underlying NVTs. Selection of a data type.</td>
</tr>
<tr>
<td>Element count</td>
<td>✔</td>
<td>Create the NV as a field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Do not create the NV as a field (default). <strong>This option is not available for defined required and optional variables of an FPT.</strong></td>
</tr>
<tr>
<td>Element count selection list</td>
<td></td>
<td>Number of elements of the field. Available if &quot;Element count&quot; is activated. <strong>This option is not available for defined required and optional variables of an FPT.</strong></td>
</tr>
<tr>
<td>Direction</td>
<td></td>
<td>Direction of the network variables (network input / network output variable, NVI / NVO). <strong>This option is not available for defined required and optional variables of an FPT.</strong></td>
</tr>
<tr>
<td>Service</td>
<td></td>
<td>Selection of the variable service to be used:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Acknowledged (ACKD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Unacknowledged (UnACKD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Unacknowledged repeated (UnACKD_RPT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Only available for output variables.</strong></td>
</tr>
<tr>
<td>Modifier</td>
<td></td>
<td>Selection of the variable modifier to be used:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Polled</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Only available for output variables.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>When selecting &quot;Polled&quot;, the NVO is not independently sent through the LON® module. It must be polled instead. Network variables using the Polled modifier should therefore only be bound to NVIs that have the required function to poll the NVO.</td>
</tr>
<tr>
<td>Priority</td>
<td></td>
<td>When activated, the values of the variables are sent prioritized.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Only available for output variables.</strong></td>
</tr>
</tbody>
</table>
### 4.5.5.3 Settings of a Configuration Property (CP)

![Screenshot of the settings of a configuration property (CP) (example)](image)

#### Table 18: Settings of a configuration property (CP)

<table>
<thead>
<tr>
<th>Function</th>
<th>Value (example)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input text box</strong></td>
<td></td>
<td>Entry of the name of the CP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When selecting the &quot;Implement as NV&quot; option, max. 16 characters.</td>
</tr>
<tr>
<td><strong>Data type indicator</strong></td>
<td></td>
<td>Information about the underlying CPTs.</td>
</tr>
<tr>
<td><strong>Element count</strong></td>
<td>☑</td>
<td>Create the CP as a field.</td>
</tr>
<tr>
<td></td>
<td>☐</td>
<td>Do not create the CP as a field (default).</td>
</tr>
<tr>
<td><strong>Element count selection list</strong></td>
<td></td>
<td>Number of elements of the field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Available if &quot;Element count&quot; is activated.</td>
</tr>
<tr>
<td><strong>Implement as NV</strong></td>
<td>☑</td>
<td>The CP is created and handled like an NV.</td>
</tr>
<tr>
<td></td>
<td>☐</td>
<td>The CP is not implemented as an NV (default).</td>
</tr>
<tr>
<td><strong>Constant</strong></td>
<td>☑</td>
<td>Write-protected CP, cannot be changed using a network management tool.</td>
</tr>
<tr>
<td></td>
<td>☐</td>
<td>CP is not write-protected (default).</td>
</tr>
<tr>
<td><strong>Device-Offline</strong></td>
<td>☑</td>
<td>LON® FTT module is set to the &quot;soft-online&quot; state using a network management</td>
</tr>
<tr>
<td></td>
<td>☐</td>
<td>tool before changing the CP.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LON® FTT module is not set to the &quot;soft-online&quot; state using a network</td>
</tr>
<tr>
<td></td>
<td></td>
<td>management tool before changing the CP.</td>
</tr>
</tbody>
</table>
### Device-Specific

- Indicates that this CP should always be read from the LON® FTT module instead of using the value from the XIF file or the network database.
- This mechanism is used for CPs managed via the LON® FTT module or via a passive network management tool that has not access to the network database.
- An example of such a CP would be a setpoint value changed using the PLC application.

- Indicates that this CP should not be read from the LON® FTT module instead the value from the XIF file or from the network database should be used (default).

### Manufacturing-Only

- Indicates that this CP is read and written when creating the PLC application, normally in the field but is no longer modified.

- Indicates that this CP is not written when creating the PLC application (default).

### Reset-Required

- Indicates that a network management tool resets the LON® FTT module after the CP has been modified.
- When receiving the corresponding network management command, the ShortStack MircoServer is reset within the LON® FTT module.

- Indicates that a network management tool does not reset the LON® FTT module after the CP has been modified (default).

### Default value […] button

- Opens the dialog for setting the default value.

Another dialog is available for setting CP default values.

The dialog is either opened under Default value by clicking the [...] button or by double-clicking the CP.

In this dialog, all structural elements of a CPTs appear.

![Configuration dialog for setting CPs, example: SCPTinstalledLevel](image)

**Figure 14:** Dialog for setting CPs, example: SCPTinstalledLevel
Figure 15: Dialog for setting CPs, example: SCPTdefaultSetting

Table 19: Fields for the default values of a CP

<table>
<thead>
<tr>
<th>Function</th>
<th>Value (example)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element column</strong></td>
<td></td>
<td>Name of the data type. When the underlying data type of the CP is a structured type, the elements of the type (informative field) appear in the rows that follow.</td>
</tr>
<tr>
<td><strong>Type column</strong></td>
<td></td>
<td>Name of the base type (informative field).</td>
</tr>
<tr>
<td><strong>Value column</strong></td>
<td></td>
<td>Raw value of the elements. The &quot;Scaled value&quot; field is automatically updated when an entry is made. Cannot be modified when the &quot;Type&quot; is Struct.</td>
</tr>
<tr>
<td><strong>Scaled value column</strong></td>
<td></td>
<td>Scaled value of the element. The &quot;Value&quot; field is automatically updated when an entry is made. Cannot be modified when the &quot;Type&quot; is Struct.</td>
</tr>
<tr>
<td><strong>Unit column</strong></td>
<td></td>
<td>Character string with the unit or information about the element (informative field). If there is no character string in the definition (installed resource) of the element, then &lt;Invalid string resource&gt; appears.</td>
</tr>
<tr>
<td><strong>[OK] button</strong></td>
<td></td>
<td>Closes the dialog and applies the entries.</td>
</tr>
<tr>
<td><strong>[Cancel] button</strong></td>
<td></td>
<td>Closes the dialog, entries are discarded.</td>
</tr>
</tbody>
</table>
4.6 Function / Status Area

The function and status area is located at the bottom of the user interface.

This area contains important buttons.

- **Generate code** for creating the WAGO-I/O-PRO function block

- **Import** for importing a LON® network interface as a *.wlicf file that has already been used in another application.

- **Export** for exporting the LON® network interface as a *.wlicf file for use in another application.

The status also appears when there are incorrect settings.

![Example views of the function / status area](image)

Figure 16: Example views of the function / status area
Table 20: Elements of the function / status area

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Generate code]</td>
<td>Clicking the button generates the WAGO-I/O-PRO function blocks for implementing the LON® FTT module in a PLC program. At the same time, the XIF file is generated in the specified directory with the general settings under &quot;Storage location for device template:&quot;.</td>
</tr>
<tr>
<td>[Import]</td>
<td>Clicking this button imports a LON® network interface as a *.wlicf file that has already been used in another application.</td>
</tr>
<tr>
<td>[Export]</td>
<td>Clicking this button exports a LON® network interface as a *.wlicf file for use in another application.</td>
</tr>
<tr>
<td>&quot;Warning triangle&quot; symbol</td>
<td>Clicking this symbol opens or closes the message list. This symbol appears when errors are detected during the configuration.</td>
</tr>
<tr>
<td>Version</td>
<td>Display of the software version of the user interface.</td>
</tr>
<tr>
<td>Message list</td>
<td>Display of errors that have occurred with a description and the element concerned in the LON® network interface. Double-clicking on an entry selects the element concerned in the tree structure of the LON® network interface.</td>
</tr>
</tbody>
</table>
4.7 **WAGO-I/O-PRO Function Blocks**

After generating the function blocks, the modules for integrating the LON® FTT module in a PLC application are available in your WAGO-I/O-PRO project.

![Figure 17: Generated function blocks in WAGO-I/O-PRO](image)

**4.7.1 Directory Structure**

![Figure 18: Directory structure of the function blocks (overview)](image)
In the tree structure view, the main program entry "PLC_PRG [PRG]" always appears automatically when creating a project in WAGO-I/O-PRO. The function blocks created in the LON® Configurator are always located under a common "LON" directory.

All function blocks belonging to a LON® FTT module are summarized in a common module-specific subdirectory "LON→Module_X" where "X" stands for the "Module number" specified in the user interface (e.g. module number = 1, "LON→Module_1"). The module-specific function blocks receive the prefix "Module_X_..." as an entry accordingly with "X" as the module number.

In parallel, a cross-module subdirectory "LON→Shared" is created in the "LON" directory in which function blocks are summarized for variable conversion.

The subdirectories and program / function blocks are briefly described in the subsequent chapters based on the directory structure. Further details are also available in the appendix of the respective chapters.

For integration in the PLC application program, it is important to note the sequence of the generated program / function blocks. This is listed below before the module descriptions.

### 4.7.2 Call Sequence

To enable access to the LON® FTT module from the WAGO-I/O-PRO application via the generated function blocks, the following call sequence is required at a minimum when creating the WAGO-I/O-PRO project:

Table 21: Program call sequence (minimum)

<table>
<thead>
<tr>
<th>Main Program</th>
<th>Module Main Program</th>
<th>LON® Master Module</th>
<th>Object Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLC_PRG [PRG]</td>
<td>Module_X_PRG [PRG]</td>
<td>Module_X_LonMaster</td>
<td>Module_X &lt;Name des LMO&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.7.3 PLC_PRG [PRG]

The PLC_PRG [PRG] is a module of the "Program" type and is automatically generated by WAGO-I/O-PRO when creating a new project.

This main program block must be contained in each PLC program for project execution if there is no task configuration for the execution sequence. This module may not be deleted or renamed.

PLC_PRG [PRG] is called up exactly once per control cycle.
4.7.4 Cross-Module Function Blocks (Shared)

Function blocks for value conversion are summarized in the cross-module directory "LON→Shared". These function blocks can be used across all created modules.

Table 22: Cross-module function blocks in the directory "LON→Shared"

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FbDecode&lt;Name of the NVT&gt;</td>
<td>Function block for calculating scaled values of an NV.</td>
</tr>
<tr>
<td>FuEncode&lt;Name of the NVT&gt;</td>
<td>Function for calculating raw values of an NV.</td>
</tr>
</tbody>
</table>

General "Decode" and "Encode" function blocks are always created when generating the code.

The function blocks for specific calculations are only generated for a few selected data types:
- SNVT_switch
- SNVT_setting
- SNVT_str_asc
- SNVT_lev_percent
- SNVT_temp
- SNVT_temp_p

More details about the individual function blocks is available in the appendix.

4.7.5 Module-specific Function Blocks (Module_X)

In addition to the entry for the "Module_X_PRG [PRG]" program block required for data exchange with the LON® FTT module, the module-specific "LON→Module_X" directory contains other subfolders.
Table 23: Generated function blocks in the module-specific "LON\(\rightarrow\)Module_X" directory

<table>
<thead>
<tr>
<th>Module/Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module_X_PRG [PRG]</td>
<td>Program for data exchange between the PLC main program and LON® FTT module.</td>
</tr>
<tr>
<td></td>
<td>(See chapter: &quot;Module_X_PRG [PRG]&quot;)</td>
</tr>
<tr>
<td>ApplicationMessage</td>
<td>Directory that contains function blocks for sending and acknowledging LonWorks application messages (explicit messages).</td>
</tr>
<tr>
<td></td>
<td>(See chapter: &quot;Function Blocks in LON(\rightarrow)Module_X(\rightarrow)ApplicationMessage&quot;.)</td>
</tr>
<tr>
<td>Internal</td>
<td>Directory that contains function blocks for internal processing.</td>
</tr>
<tr>
<td></td>
<td>(See chapter: &quot;Function Blocks in LON(\rightarrow)Module_X(\rightarrow)Internal&quot;.)</td>
</tr>
<tr>
<td>Master</td>
<td>Directory that contains function blocks for communication between the PLC application and LON® FTT module.</td>
</tr>
<tr>
<td></td>
<td>(See chapter: &quot;Function Blocks in LON(\rightarrow)Module_X(\rightarrow)Master&quot;.)</td>
</tr>
<tr>
<td>NV_Interface</td>
<td>Directory that contains all function blocks of the LMOs as Module_X_&lt;Name of the LMO&gt; with the NVs and CPs configured in the user interface.</td>
</tr>
<tr>
<td></td>
<td>(See chapter: &quot;Function Blocks in LON(\rightarrow)Module_X(\rightarrow)NV_Interface&quot;.)</td>
</tr>
</tbody>
</table>

The individual function blocks are briefly described in the following chapters and described in detail in the appendix.

### 4.7.6 Module_X_PRG [PRG]

For each module created, a module-specific program block is generated that enables data exchange between the PLC main program and the respective LON® FTT module. These program blocks must be called up cyclically from the PLC main program PLC_PRG [PRG].

The most important task of these program blocks is to call the embedded LON® master function block Module_X_LonMaster.

### 4.7.7 Function Blocks in the "ApplicationMessage" Directory

**Note**

Only use "ApplicationMessage" function blocks selectively! Use the function blocks for sending and acknowledging LonWorks® application messages (explicit messages) if you use them explicitly. These function blocks affect the LON® network directly and thus represent a high risk of collision between the network commands to the bus nodes and the processing of the application.
Table 24: Module-specific function blocks in "LON→Module_X→ApplicationMessage"  

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module_X_ApplicationMsgSend</td>
<td>Function block for sending LonWorks® application messages (explicit messages)</td>
</tr>
<tr>
<td>Module_X_ApplicationMsgSendRsp</td>
<td>Function block for acknowledging LonWorks® application messages (explicit messages) received as a REQUEST (ServiceType = LON_SRVC_REQ).</td>
</tr>
</tbody>
</table>

The structure of these function blocks is described in the appendix in detail.

4.7.8 Function Blocks in the "Internal" Directory

This directory contains automatically generated function blocks required for internal processing for updating and value polling of the LON® FTT module.

4.7.9 Function Blocks in the "Master" Directory

This directory contains the LON® master function block Module_X_LonMaster [FB]. This LON® master module is required for the functionality.

An instances if this module is automatically created and integrated in the first network of the program when creating the WAGO-I/O-PRO modules in "Module_X_PRG" because the "Module_X_PRG" program block contains the variable definition of the LON® master function block "Module_X_LonMaster".

4.7.10 Function Blocks in the "NV_Interface" Directory

In this directory, instances of the WAGO-I/O-PRO function blocks "Module_X_<Name of the LMO>" are defined that can be used to represent the LMOs created in the LON® configurator plug-in.

The specified name is used as the instance name.

Figure 1: Correlation between the user interface and the function blocks created
The structure of the modules for LonMark® objects is variable. More details about these function blocks are available in the appendix using two examples.

Programmatically, the instances are created by the code generator in the module main program "Module_X_PRG [PRG]" (see figure below) within the program section shown below. The variables that establish the connection to the process image are also declared automatically.

```c
(* Automatic added code, do not change! *)
...
(***** End of automatic added code. *****)
```

![Figure 2: Module_X_PRG with variables (example)](image)

Changes in the LON® network interface and generating the WAGO-I/O-PRO code only changes the code within the section above. User code or networks generated outside this section are not changed.

---

**Note**

A modified LON® interface configuration can disrupt binding!
When you have already integrated the LON® FTT module in the LON® network and created the binding in the LNMT, please note that loading a modified interface configuration of the LON® FTT module can disrupt the binding (see chapters "Using a LonWorks® Network Management Tool" → "Integrating a Modified Interface Configuration").

Other variables should be added after the line:

```c
(* Add your code here. *)
```
The Neuron® status is staged on the "typNeuronState" output parameter. Status polling must be explicitly initiated (rising edge on input "xRequestNeuronState").

The Neuron® status can be reset via a rising edge to the input "xResetNeuronState".

**Note**

The structures uses are defined in the LON_01.lib library! Please note that the "NeuronState", "DomainData" and "ApplicationMessageData" data types used are defined in the LON_01.lib library. The LON_01.lib library is available to download free at: www.wago.com.

---

![Figure 19: Defined data types in the integrated LON_01.lib library](image-url)
4.7.11 Launch Behavior/Reset

4.7.12 Launching the PLC Application

After launching the PLC application, communication between the LON® master and the LON® FTT module is first synchronized. The LonWorks® network interface configuration is then synchronized with the LON® FTT module by the LON® master. The configuration is then transferred if there is a different configuration in the LON® FTT module.

After successful synchronization, the LonWorks® network interface configuration is then activated on the LON® FTT module by the LON® master. Only then is the Neuron® controller contained in the LON® FTT module released from the Reset state to be configured as shown below. Until this point, the LON® FTT module does not respond to incoming LonTalk® telegrams.

When activated, the values of the configuration properties are transferred from the persistent memory (EEPROM) of the LON® FTT module to the LON® master. The values are transferred independent of the other initialization processes and depending on the number and size of the configuration properties, can take longer than the start-up process of the PLC.

After successful activation, the values of all NVIs from the LON® FTT module are polled by the "Module_X_<Name of the LMO>" modules and the initial values of all NOVs transferred to the LON® FTT module. However, the NVOs are not sent, but only saved in the LON® FTT module, so that polling on the fieldbus side can be answered with the correct value of the NVO.

The PLC application is then able to send and receive NVs and application messages via the LON® FTT module.

The initialization process described above is controlled and monitored by a state device contained in the LON® master function block. The function block must be called cyclically for correct function.

The PLC application can use the "wModuleState" output parameter of the LON® master (see chapter "1.5.4 Function Blocks in LON®→Module_X→Master") to obtain information about the current state of the LON® FTT module.

The PLC application should only begin sending network variables and application messages after initialization is complete. Bits 5 and 6 (bit mask 16#0060) must be checked by "wModuleState". If both bits are set, initialization has been completed.

When using CPs, the PLC application may also need to observe bit 3 (bit mask 16#0008) (see chapter "Synchronization").
4.7.13 Resetting the Neuron® Controller

The Neuron® controller can be reset using a LonWork® network management command. However, such a process is handled within the LON® FTT module itself and does not lead to a reset of the host controller within the LON® FTT module. Resetting the Neuron® controller has no effect on process data communication.

4.7.14 LonWorks® Configuration Properties (CPs)

The LonWorks® configuration properties are kept in the persistent memory of the LON® FTT module. They are also available to the PLC application, which can read/write them. The "Module_X_<Name des LMO>" module can also manage the internal cache.

4.7.15 Synchronization

During initialization, the CP values from the LON® FTT module are transferred to the cache of the function blocks (see also chapters "Launch Behavior/Reset", "Launching the PLC Application").

The values are also transferred when a CP value has been modified by a LonWorks® network management tool (e.g. part of a startup process of the LON® network).

Both synchronization processes run asynchronously to the rest of the communication between the LON® FTT module and PLC application.

Whether the data is current in the cache of the function block is indicated in bit 3 (bit mask 16#0008) of the "wModuleState" output parameter of the "Module_X_LonMaster" function block:

- TRUE: Values in the cache are not current.
  - LED "E" on the LON® FTT module also flashes.
- FALSE: Values in the cache are current.

4.7.16 SCPTmaxSendTime / SCPTminSendTime / SCPTmaxRcvTime Configuration Properties

If only the "SCPTmaxSendTime" or "SCPTminSendTime" configuration properties are defined for network variables, they are managed within the LON® FTT module directly. The associated functionality ("Heartbeat" and "SendThrottle") is implemented by the LON® FTT module.

The same applies to the "SCPTmaxRcvTime" configuration property for a network input variable.

The LonMark® SCPT master list contains more information about the function associated with the aforementioned configuration properties.
### Table 25: Generic configuration properties

<table>
<thead>
<tr>
<th>SCPT</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCPTmaxSendTime</td>
<td>Send interval</td>
</tr>
<tr>
<td></td>
<td>This SCPT can be assigned to an NVO. It defines the send interval at which the values are sent by the LON® FTT module even if the value itself has not changed. The interval timer is restarted when the PLC application initiates a transmission (e.g. by a change in the value).</td>
</tr>
<tr>
<td></td>
<td>If the CP is assigned the value 0, cyclic transmission is deactivated.</td>
</tr>
<tr>
<td>SCPTminSendTime</td>
<td>Minimum send interval</td>
</tr>
<tr>
<td></td>
<td>This SCPT can be assigned to an NVO. It defines the minimum time between two successive transmissions of the NVO.</td>
</tr>
<tr>
<td></td>
<td>If the value of the NVO changes before the timer has elapsed, the transmission is suppressed until the timer has elapsed.</td>
</tr>
<tr>
<td></td>
<td>When several successive changes in value occur within the minimum send interval, only the last value is transmitted after the timer expires.</td>
</tr>
<tr>
<td></td>
<td>If the CP is assigned the value 0, transmission suppression is deactivated.</td>
</tr>
<tr>
<td>SCPTmaxRcvTime</td>
<td>Receipt monitoring</td>
</tr>
<tr>
<td></td>
<td>This SCPT can be assigned to an NVI to implement monitoring for regularly incoming updates to the NV.</td>
</tr>
<tr>
<td></td>
<td>If no NV update is received within the configured time interval, the timeout is signaled by setting bit 4 of the &quot;bStatusInformation&quot; element variables of the NV representatives (see chapter &quot;Module__&lt;Name of the LMO&gt;&quot; in the appendix).</td>
</tr>
<tr>
<td></td>
<td>The bit is reset by the next NV update.</td>
</tr>
<tr>
<td></td>
<td>If the CP is assigned the value 0, receipt monitoring is deactivated.</td>
</tr>
</tbody>
</table>

**Note**

Assign SCPTmax / minSendTime and maxRcvTime to an NV!  
Please note that the SCPTs "SCPTmaxSendTime", "SCPTminSendTime" or "SCPTmaxRcvTime" must be assigned to individual network variables.  
If one of the listed SCPTs is assigned to an LMO, the automatism described here is not supported. The required function must then be implemented in the PLC application!
5 Using a LonWorks® Network Management Tool

5.1 Integrate LON® FTT Module in the LON® Network

A LON® network management tool or "LNMT" (e.g. LonMaker®) is used to integrate the LON® FTT module in the connected LonWorks® network.

For the basic process, perform the following steps. A detailed description of these steps is available in the example configuration (chapter "Example Configuration").

More information about the LNMT used!
For a detailed description of the software and software operation, please refer to the respective manual for your LonWorks® network management tool software.

1. Launch your LonWorks® network management tool.
2. Create a new network.
3. Create a device.
4. Upload the configured LON® network interface definition from the LON® FTT module into the LonWorks® network management tool online or by importing the XIF file.
5. To send the Service Pin command, bridge CAGE CLAMP® connections 2 and 6 on the LON® FTT module.
6. Create a "functional block".
7. Apply the variables of the LON® interface configuration by creating the shapes automatically.
8. Repeat the process for other LON® nodes.
9. Connect the nodes (binding).
5.2 Integrate a Modified Interface Configuration

---

**Note**

A modified LON® interface configuration can disrupt binding!
When you have already integrated the LON® FTT module in the LON® network and created the binding in the LNMT, please note that loading a modified interface configuration of the LON® FTT module can disrupt the binding!

---

**Note**

Description differs from the original software description!
Please note that the following instructions for loading a modified interface definition refers to a LonMakers®. However, the steps described here differ from the original description in the "LonMaker® User's Guide". The Load command used in the original description ("Commissioning → Load") does not apply to the LON® FTT module because this I/O module has a ShortStack architecture.

---

If you load the LON® FTT module in the connected LON® network after initial integration with the modified interface configuration, make sure that the previously established connections (binding) are not disrupted. Perform the following steps (described here with the LonMaker®).

1. For the required modification to the LON® interface definition of the LON® FTT module, use the LON® Configurator to adapt it.

2. In the interface of the LON® configurator, you must change the program ID in the top area by making the general settings. For example, enter a number higher by "1" in the "Program ID" input field for the last byte.

---

**Note**

LNMT requires a unique ID for the device template!
Please note that you must change the program ID for a modified LON® interface because the template for each device must have a unique ID in the LNMT used.

---

3. Click [OK] to confirm your entry.

4. The click the [Generate code] button in the LON® configurator to generate the respective function blocks again.

5. Create your project in WAGO-I/O-PRO.

6. Load your project in the fieldbus controller.

7. Start the fieldbus node.
8. In the LonMaker® on the graphical interface, right-click on the device created for the LON® FTT module.

   The context menu appears.

9. In the context menu, select the menu item "Commissioning → Replace…".

   The "Replace Device Wizard" dialog appears.

10. In the Replace Device Wizard dialog, select the "Load XIF" option and enter the path of the storage location where the newly created XIF file is saved in the "File:" input field.

11. Click [Next>].

12. In the dialog that follows, make sure that the "Load application image" function is deactivated, i.e. the check mark is not set!

13. Click [Next>].

14. In the dialog that follows, select the settings based on the application.

15. Click [Next>].

16. Under "Device identification method", change the option to "Manual" (default: "Service pin") because LonMaker always offers the "old" Neuron® ID.

17. Click the [Finish] button to confirm the settings.

18. In the "Device Properties" dialog / "Network Variable Config" tab in the device table displayed, you can see that the previous connections remain unchanged after the upgrade.
6 Planning Strategies and Restrictions

Note
Please note the limits and strategies of project planning!
Be sure to note the strategies defined for project planning and the factors
described below that affect utilization before planning and executing your own
project to ensure optimal function of your project with the LON® FTT module
753-648.

6.1 Planning Strategies for Project Creation

Note
Number of LON® FTT modules per fieldbus node depends on the
application!
Please note the following factors that have a significant effect on utilization and
the maximum number of usable LON® FTT modules one one fieldbus node:

- No. and type of NVs
- Use of arrays for NVs of the same type (see example below)
- Optimized arrangement of the variables (see example below)
- Type of fieldbus controller and available memory

Resources can be saved by using as few different types of network variables as
possible and by creating ARRAYs for network variables or FPTs that are used
several times.

The comparison below of various example projects should illustrate the different
uses.

6.1.1 Example with and without ARRAY

In the example application, 20 SFPT_chiller are created.

In the first program, the modules are individual declared and in the second
program, as an ARRAY.

The result is clear: The program with the individual modules (see "Example
Project 1") is three times larger than the program with the ARRAY (see "Example
Project 2").
6.1.1.1 Example Project 1

The first program with the individual modules has a code size of 325494 bytes.

Figure 20: Example project 1: Modules declared individually, code size 325494 bytes

6.1.1.2 Example Project 2

The second program with the ARRAY has a code size of 101542 bytes.

---

**Note**

Declare modules as ARRAYs!

Please note that you should use ARRAYs for module declarations when possible.
6.1.2 Further Optimization by Arrangement

The arrangement of the variables can further reduce the code size of the generated modules. This is illustrated by another case example with various optimized implementations.

The functionality of all PLC programs is the same: Digital and analog inputs are mapped to NVOs and NVIs to digital and analog outputs.

The first of these examples (see "Example Project 3") is not optimized. In comparison, the code requirement of the other examples decreases by organizing the objects differently with the same arrangement (see "Example Project 4" to "Example Project 7").

6.1.2.1 Example Project 3

In this first example shown (example project 3), 20 controller objects (FPTs) are created each with one NVI and one NVO.

The program has a code size of 278530 bytes.
6.1.2.2 Example Project 4

In example project 4, all NVs are consolidated in one controller objects instead of using 20 FPTs.

The program has a code size of 126522 bytes.

**Note**

Consolidate NVs in LMOs!

Please note that you should consolidate NVs into LMOs when possible to reduce the number of FPTs used.
6.1.2.3 Example Project 5

In example project 5, the required NVIs and NVOs in the VFB are created as ARRAYS instead of using 40 individual NVs and sorted in order of inputs and outputs.

The program has a code size of 117630 bytes.

Sort NVs in relation to direction!
Please note that you should sort NVs in order by type and direction when possible.
6.1.2.4 Example Project 6

In example project 6, an FPT ("Controller [10]"") is created as an ARRAY with the required NVs instead of using 20 (or 10) FPTs.

The program has a code size of 113874 bytes.

Use as few FPTs as possible!
Please note that you should use as few FPTs as possible.

Note
6.1.2.5 **Example Project 7**

Example project 7 is used for comparison with project 4 and clearly shows how much memory requirements are affected when NVIs and NVOs are not set up closely one after the other and the index area for NVs of the same direction is not uniform.

The program has a code size of 127498 bytes and in comparison to the program in example project 4 with 126522 bytes, needs approx. 1 kB more memory.

---

**Note**

Create **NVs of the same direction one below the other**!

Please note that you should always create all NVs of the same direction one below the other, so the index area for the NVs is uniform.
Figure 26: Example project 7: with optimization, code size 127498 bytes
6.2 Restrictions

When creating a project, the following maximum values apply as restrictions:

Table 26: Restrictions

<table>
<thead>
<tr>
<th></th>
<th>Maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of NVs</td>
<td>249 (total: 254, 5 of which occupied by Node Object → 249)</td>
</tr>
<tr>
<td>No. of CPs</td>
<td>2048</td>
</tr>
<tr>
<td>Size of a CP</td>
<td>248 bytes</td>
</tr>
<tr>
<td>Size of the user data of an application message</td>
<td>48 bytes</td>
</tr>
<tr>
<td>No. of alias addresses</td>
<td>127</td>
</tr>
</tbody>
</table>

Note

Other key data acc. specification for "FT 5000 Smart Transceiver"!
Please note that other key data is based on the specifications of the "FT 5000 Smart Transceiver" from Echelon®.
The technical data for the "FT 5000 Smart Transceiver" is available on the Echelon® website at:
www.echelon.com/products/components/ic/ft5000/default.htm
7 Example Configuration

A prerequisite for the commissioning example described below is that you have correctly installed and set up the hardware for your fieldbus node and the LON® network on the LON® FTT module and that these items all function properly.

The LON® FTT modules are supplied in the fieldbus node via the system voltage.

In the example given here, the fieldbus node consists of the following WAGO-I/O-SYSTEM components:

Table 27: Example hardware for a fieldbus node setup

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>750-830</td>
<td>BACnet/IP programmable fieldbus controller</td>
</tr>
<tr>
<td>753-648</td>
<td>LON® FTT module</td>
</tr>
<tr>
<td>750-600</td>
<td>End module</td>
</tr>
</tbody>
</table>

A two-wire conductor connects the LON® bus to both LON® bus A and LON® bus B connections.

![Configuration diagram of a LON® bus to the LON® FTT module](image)

Figure 27: Configuration diagram of a LON® bus to the LON® FTT module

The PC is linked to the fieldbus node via an RJ-45 network cable. The PC's network card must be set based on the address range for the fieldbus node.

A connection can also be made via the fieldbus controller serial interface. Use the WAGO communication cable to set up a physical connection via the serial service port. This cable is supplied with the WAGO-I/O-PRO (Item No.: 759-333) programming software, or can be obtained as an accessory under Item No.: 750-920.
**NOTICE**

**Do not connect 750-920 Communication Cable when energized!**
To prevent damage to the communications interface, do not connect or disconnect 750-920 Communication Cable when energized! The fieldbus controller must be de-energized!

If all conditions listed and the principles specified in the "Preparation" chapter are met, you can launch the LON® Configurator via the successful PLC configuration in WAGO-I/O-PRO (see chapter "Open LON® Configurator in WAGO-I/O-PRO") and start configuring immediately.

### 7.1 Example Configuration

#### Note

Example using network management!
Please note that a network management tool (LNMT, here the LonMaker® from Echelon®) is used in the example described below.

#### Note

More information about the LNMT used!
For a detailed description of the software and software operation, please refer to the respective manual for your LonWorks® network management tool software.

#### 7.1.1 Step 1: **DEFINE LON® interface**

#### 7.1.1.1 Launch LON® Configurator

1. Save the LON® library "LON_01.lib" to the program directory in the folder: 
   "WAGO Software\CoDeSys V2.3\Targets\WAGO\Libraries\Building".
This library loads the Mailbox module "WagoLibMBX_01.lib".

2. Launch the program WAGO-I/O-PRO.

3. Create a new project, select your target system (in this example "WAGO_750-830") and in the "New Module" dialog, select the module language (e.g. FUP).
   Leave the default settings for the "Module type: Program" and "Module name: PLC_PRG" options.

   After clicking [OK] to confirm your settings, the PLC_PRG [PRG] program module is automatically created.

4. Switch to the "Resources" tab and double-click the "PLC Configuration" entry.

   The PLC configuration dialog appears.

5. If you have not already done so, configure the PLC according to your physical hardware.

6. Then in the tree structure of the PLC configuration, expand the "Hardware configuration" entry and then the "K-Bus (*BACNet*) [FIX]" entry.

7. Click on the "0753-0648, LON-FTT[VAR]" sub-entry for the LON FTT module to select it.

   In the PLC configuration dialog on the right, the user interface of the LON® Configurator now appears.
7.1.1.2 General Settings

1. At the top of the "Configure LON® Interface" tab, in which you carry out the general settings, first configure the program ID by clicking the [...] button behind the "Program ID" input field.

The "Configure Program ID" dialog appears.

![Configure Program ID dialog](image)

Figure 29: Configure program ID (example)

2. Select the respective parameters and click [OK] to confirm.

The dialog closes and the program ID is applied to the user interface.

3. Then enter the required name and storage location for the device template at the top for the general settings.

Enter a unique module ID for the LON® FTT module, e.g. "1" for the first LON® FTT module.

For additional LON® FTT modules, you can number sequentially. The module ID in a project with several LON® FTT modules must always be unique for each LON® FTT module.
7.1.1.3 Configure Network Interface

1. In the configuration area for the LON® network interface, select all data types that you need from the tree structure of the LonMark® resources on the left.
   In this example, select the data type "SNVT_switch".

2. To move the data types to the tree structure on the right, click the [⇒] button or double-click or drag & drop.

   Figure 31: Configure LON® interface (example "SNVT_switch")

The "SNVT_switch" data type is created under "Virtual Functional Block".
3. Then carry out the required object settings by selecting the individual data types one after the other and entering or selecting the parameters displayed on the right side.
For the selected data type "SNVT_switch", choose the "Input" direction from the selection list.

![Diagram of settings](image)

Figure 32: Carry out the required object settings (example "SNVT_switch")

### 7.1.2 Step 2: GENERATE code

**Note**

Specify the storage location in advance!
Before you generate the code, please specify a directory as the storage location for the device template at the top of the dialog. Otherwise, an error message will appear when generating the code (see chapter "General Entries").

1. After defining the LON® network interface, click the [Generate code] button.

   The program code for the defined functionality of the LON® network interface is then automatically generated and made available as WAGO-I/O-PRO function blocks.
   At the same time, the XIF file is generated in the specified directory "Storage location for device template".

2. You can then integrate these function blocks in your WAGO-I/O-PRO-project and embed them in your application.

3. If you work offline in your network management tool (e.g. LonMaker®), you can import the LON® network interface of the LON® FTT module using the XIF file.
7.1.3 **Step 3: INTEGRATE in user application**

1. In WAGO-I/O-PRO, go to the "Modules" tab.

   In the tree structure in the dialog area on the left, all generated function blocks are available under the "LON" directory.

   In the "Module_1_PRG [PRG]" program module for data exchange with the LON® FTT module, the LON® master module was automatically created in an instruction window.

   In addition, the "Node Object" and "Virtual Functional Block" have been declared in the automatically generated program text in the declaration window.

   ![Function blocks and LON® master module in WAGO-I/O-PRO (example)](image)

2. Move your cursor to the gray area on the left next to the LON® master module in the instruction window and right-click.

   The context menu appears.

3. In the context menu, select the menu item "Network (after)".

   Another instruction window appears below the LON® master module.

4. In this window, click the small box with the dotted border just to right of the three question marks and then right-click.

   The context menu appears.

5. In this context menu, select the menu item "Module".

   An "AND" module is created by default.
6. Double-click the "AND" designation and press the [F2] key.

The input assistant opens.

7. In the input assistant under "User defined Function Blocks"→"LON"→"Module_1"→"NV_Interface", select the module "Module_1_Virtualfunctionalblock [FB]" and click [OK] to confirm the selection.

8. In the graphical view, then move your cursor over the three question marks above the module and press the [F2] key.

The input assistant opens.

9. In the input assistant under "Local Variables", select the "Virtualfunctionalblock [Module_1_Virtualfunctionalblock]" variable as the name and click [OK] to confirm your selection.

Alternatively, you can also enter "Virtualfunctionalblock" directly as the name of the module just as it has already been declared in the automatically generated program section.

10. To now assign the variable, click the module output behind "SNVT_switch" to display a small box with a dotted border and right-click.

The context menu appears.
11. In this context menu, select the menu item "Assignment".

Three question marks appears as a placeholder for the output variable.

12. Double-click the three question marks and enter "nviswitch" as the variable name.

13. Press the Enter key to confirm the entry.

The dialog for variable declaration appears.

14. Click the [...] button behind the input field to select the "Type".

The input assistant opens.

15. In the input assistant under "User defined Types", select e.g. the "typSnvSwitch [STRUCT]" type and click [OK] to confirm.

16. Click [OK] to confirm the dialog for variable declaration.
The "Module_1_Virtualfunctionalblock [FB]" function block with the "SNVT_switch" output variable is now created in your project.

17. If you had also created other variables, declare all other input and output variables in a similar manner and make the variable assignments.

18. To have global access in the program to the locally declared variables, go to the instruction window and select the the rows of the variable declaration (here "nviswitch: typSnvtSwitch;").

19. Use the [Ctrl]+[X] keyboard shortcut to cut this program section.

20. Switch to the "Resources" tab → "Global Variables" → "Global Variables".

21. Then use the [Ctrl]+[V] keyboard shortcut to insert this variable declaration section between the rows "VAR_GLOBAL" and "END_VAR".

22. Then return to the "Modules" tab and double-click the "PLC_PRG [PRG]" entry.

23. Move your cursor to the empty instruction window and right-click.

The context menu appears.
24. In this context menu, select the menu item "Network (after)".

An instruction window appears below the declaration window.

25. In this window, click the small box with the dotted border just to right of the three question marks and then right-click.

The context menu appears.

26. In this context menu, select the menu item "Module".

An "AND" module is created by default.

27. Double-click the "AND" designation and press the [F2] key.

The input assistant opens.

28. In the input assistant under "User defined Programs"→"LON"→"Module_1", select the module "Module_1_PRG [PRG]" and click [OK] to confirm the selection.

The program block for the die LON® FTT module is then called up in the main program.

29. Move your cursor to the "Project" main menu and click the menu item "Compile all" to compile this program.

30. Move your cursor to the main menu "Online" and click the menu item "Login" to load the program to your fieldbus controller.

If no program has been uploaded to the LON® FTT module yet, the C and G LEDs flash (internal error).

31. To launch the application, click the menu item "Start" under "Online".
After launching the application, the A and C LEDs light up (user config and K-bus OK) green and the service LEDs D and H flash. You can now integrate the LON® FTT module into the LON® network.

### 7.1.4 Step 4: INTEGRATE in LON® network

1. Launch your network management tool (here the LonMaker® from Echelon®).

![LonMaker® Start Screen](image)

Figure 39: Start screen of the LNMT LonMaker® from Echelon® (example)

2. In the dialog under "Network name:", enter the required name for your network.

3. Click the [Create Network] button to create a new network.

4. Then follow the network wizard.
5. When you have online access to your network, activate the "Network attached" by placing a checkmark in the checkbox.

6. Under "Network Interface name", enter a name for your LON® network.

7. Click [Next].

8. In the subsequent dialog, select the "OnNet" option under "Management Mode".

9. Click [Finish].
10. Drag and drop to create a new device in the LNMT dialog on the graphic interface.

11. In the dialog, select the "Upload from device" option to upload the interface definition from your LON® FTT module.
Alternatively, you can select the "Load XIF" option and enter the storage location of the XIF file in the "File:" input field or search for the storage location by clicking [Browse...].

12. Click [Finish] to close the wizard.

You are then prompted to press service pin.

13. To send the Service Pin command, bridge CAGE CLAMP® connections 2 and 6 on the LON® FTT module.
The LON® FTT module is now applied on the graphic interface in the LNMT dialog as a network node. After successful integration, the service LEDs (D and H) on your LON® FTT module go out.

14. Now create a "Virtual Functional Block" in the LNMT dialog on the graphic interface.
15. Activate the "Create all network shapes" by placing a checkmark in the checkbox.

The shapes are then automatically created in the graphic interface and the variables from the LON® interface applied.
7.1.4.1 Communication Test

1. In the LonMaker® browser, manual change the variable values to test communication to and from the LON® FTT module.

   ![Test communication in the LonMaker® browser (example)]

   Figure 48: Test communication in the LonMaker® browser (example)

In communication is error-free, you can track these changes in your IEC application online.

2. Successful communication is also indicated on the LON® FTT module by the LEDs because LED E lights up and the LON® data transfer can be observed on LEDs B and F.

   ![Test communication, variable change in the IEC application (example)]

   Figure 49: Test communication, variable change in the IEC application (example)

7.1.5 Step 5: CONNECT to other LON® bus nodes

1. To integrate other nodes, repeat the described processes.

2. You can then connect the LON® network nodes through binding as required
8 Glossary

Information
More information about LON® technical terms!
Detailed information about LON®-specific terms and technologies is available on the Internet on the LonMark® organization website at: www.Lonmark.org.

C

CPs
Abbreviation for "Configuration Properties". The "Configuration Properties" are variables that specify the configuration for specific LonWorks® devices.
(see also SCPT)

CSMA
CSMA stands for "Carrier Sense Multiple Access" and is a special bus access method. With CSMA the node first "listens" to the network before becoming active. LonWorks® operates with a special CSMA method, which permits a short reaction time with a high throughput rate even in large networks.

D

Device template
Device templates provide the profile of a node in the network. They are available in the form of XIF files (External Interface Files) and contain all network-relevant data. This data as a whole determines the interface to the network. For the integration of a node in a system, the network interface (e.g. transceiver parameter and detailed information about the network variable used) must be known.

Domains
The largest addressing units are domains. These are used to build complete independent sub-systems, e.g. lighting systems, access control (in as much as these must not communicate with each other). Thus domains form a virtual network within the physical network structure. Each LON® unit can be addressed over two domain addresses. One domain can be allocated a maximum of 255 Subnets each with 27 units (together comprising of 32,385 units).

Driver
Software code that communicates with a hardware device. This communication is normally performed by internal device registers.
**E**

**Echelon**

The company Echelon® is the technology provider for LonWorks® technology.

Information about Echelon® is available on the Internet at:

**F**

**FPT (Functional Profile Template)**

FPTs consist of combined LMOs and thereby form functional units (functional profiles) that are defined across manufacturers.

FPTs are parts of LMRFs (LonMark® Resource Files).

(see also **LMO** and **LMRF**)

**Free Topology**

The Free Topology is a network topology, which was possible for the first time with the FTT-10 Transceiver. In free topology, it is possible to mix line, star or ring structures together. However, in this case the maximum transmission spacings are dependent upon the cable quality. Due to the use of routers or repeaters, the transmission spacing limits can be overcome.

**Function**

*Module* that always returns the same results (as a function value), It has no local variables that store values beyond an invoke.

**Function block**

*Module* that returns one more or more values when executed. It can be saved as a local variable ("memory").

**I**

**IEC 61131-3**

International standard for modern systems with PLC functionality created in 1993. Based on a structured software model, it defines a series of powerful programming languages to be utilized for different automation tasks.

**Interoperability**

Interoperability means that means that the cooperation between different systems is ensured by adherence to common standards. To ensure interoperability with LON®, over a hundred defined *SNVT* and *SCPT* network variables are combined into objects that precisely describe and represent the meaning, value and range of sensors, actuators and controller functions.
LAN
Abbreviation for Local Area Network.

Library
Collection of modules available to the programmer in the WAGO-I/O-PRO 32 programming tool for creating control programs in accordance with IEC 61131-3.

LMO (LonMark® Object)
Various network variables are combined into object with LON® that logically represent sensors, actuators and controller functions. LMOs combined into specific functions form a FPT (Functional Profile Template).
(see also FPT and LMRF)

LMRF (LonMark® Resource Files)
The LonMark® resource files contain all available variables from which the required functionality can be selected and compiled into LMOs. In addition to standard network variable types (SNVTs) and standard configuration property types (SCPTs), user-defined types (UNVTs/UCPTs) as well as LonMark® functional profiles (FPTs) are also supported (see also FPT and LMO).

LNO®
The LNO® – Lonmark Deutschland e.V. – is the association for companies, institutions and distributors using the LonWorks® technology in the German speaking territories. Current information concerning the LNO and the member list can be found on the Internet at: http://www.lno.de.

LNS
Abbreviation for "LonWorks® Network Service". LNS is a network operating system for LonWorks® networks.

LNS database
In the LNS database, all configuration settings are saved and administered by an LNS server.

LON®
LON® is the abbreviation for "Local Operating Network" and is a multi-master capable communication network developed for distributed industrial applications with requirements that are not time critical. LON® was specially developed for building automation. With LON®, the central tasks are subdivided into small distributed tasks to be performed, so that in each distributed intelligence (node), applications can to a large extent be processed locally without placing a burden on other bus nodes.
LonWorks®

LonWorks® is one of the uniform communication standard in building automation. This standard was developed by an independent technological institution, the US based Echelon, supported by the semi-conductor manufacturers Cypress and Toshiba. It is backed by a comprehensively documented technology open to everybody who wants to use it. It includes, for example, the Neuron® chips, bus link modules (transceivers), development tools, software packages, support. With LonWorks®, decentralized information processing structures are possible that function without centralized control (e.g. PLC). In this respect, LonWorks® differs from various other fieldbus solutions.

LPT-10

Abbreviation for "Link Power Transceiver".
This transmission medium is a twisted pair variant. Technically, it corresponds to the "Free topology FTT-10" variant, except that it has the advantage that the unit supply voltage can be jointly transmitted via the bus line. In this manner, a conductor pair is saved in the cable and the chances of a mistake when connecting is reduced. However, LPT-10 requires an additional supply voltage level, namely a special link power supply (input voltage e.g. 48 - 56 V, output voltage approx. 42 V/1.5A). There are also limitations with regard to loading capacity - only a limited number of units can be supplied with power by a link power pack (important is, for example, units with light diodes or relays, which often have higher current requirements). Installation advantages include above all buildings in which wiring of push-buttons and switches are to be found. Link power signals can also be switched on TP/FT-10 units, if these contain the corresponding block capacitors, which blocks off the supply voltage.

Note:
Consideration of the economics may be necessary prior to using LPT-10. Ensure that the power supply is of a suitable level with reserve capacity in accordance with the worst case for all units in the segment!
In addition, also check the LPT-10 compatibility of TP/FT-10 units.

M

Module

Modules in general consist of functions, function blocks and programs. Every module is made up of a declaration part and a body. The body is written in one of the IEC programming languages AWL (statement list - STL), ST (structured text), AS (process structure), FUP (function plan) or KOP (coupling plan).
Network interface
The network interface describes the interface of a node to the network. It is comprised of a number of objects where an object is defined for each individual task in a node. These objects again consist of a set of network variables (NVs) and a set of configuration properties (CPs).

Network management tool software
A network management tool is software, in which network devices (nodes) can be integrated, addressed and maintained, as well as network variables be binded. The basis of a network management tool should be LNS. Then the configuration plug-ins of different manufactures can be launched from the tool.

Network variable
A network variable (NV) is a type-related variable in the Neuron® C programming language for implementation of logical communication channels between LON® nodes. It can be associated with one or more network variables of one or more network nodes. With standardized network variable types (SNVTs), interoperable (independent of the manufacturer) communication of the LON® node in a network is possible. If data is transmitted from a node to the network, it occurs via the network output variable (nvo). If data is transmitted from the network to the node, it occurs via the network input variable (nvi).

Neuron® C
Neuron® C is a programming language based on the ANSI C standard for application programming on a Neuron® chip, the microcontroller in a LON® node.

Neuron® ID
Each microcontroller in a LON® node, Neuron® chip, has its own identification number, the Neuron® ID.

Node
A node is a unit or a device or module equipped with a Neuron® chip as a microcontroller, possibly supplemented by an external memory and I/O function. The smallest addressing units are nodes.

NVs, NVI, NVO
Abbreviation for Network Variable.
(NVI = Network Input Variable, NVO=Network Output Variable)

Power-Line
"Power-Line" is data transmission via the 230V network.
Repeater
"Repeater" are physical amplifiers without their own processing function. They refresh data without detecting damaged data and forward all signals. Repeaters are used for longer transmission distances or when the maximum number of nodes of 64 devices per twisted pair segment is exceeded. Note: In TP/FT-10 networks, only one physical repeater is permitted between two nodes. Otherwise, routers configured as repeaters have to be used. Such a router also allows a media change. The repeater always counts as a node. This means that for each segment, 63 nodes + 1 repeater can be used.

Reset
"Reset" means switching off or failure of the power supply. After switching on the power supply again, the device is restarted and initialized again.

Request
A "request" is a service request from a client that requests a service from a server.

Response
A "response" is a replay from a server to the request from a client.

Router
A router is used to connect neighboring subnets with the router operating with addresses and protocols of the third ISO/OSI layer. As this layer is hardware independent, the routers allow transition to another transmission medium. To transmit a message the router evaluates the logical address (source and destination address) and finds the best path if there are several possibilities. Routers can be operated in the Repeater or Bridge modes.

SCPT
Abbreviation for Standard Configuration Property Types. SCPTs [pronounced scipits] are predefined standardized configuration variables. There are a number of SCPTs that make interoperability easier by providing a well-defined, compact mechanism for handling large volumes of configuration data on one device. (See also Configuration variable)

Segment
Typically, a network is divided up into different physical network segments by way of routers or repeaters.

Server
Device providing services within a client/server system. The service is requested by the Client.
Service Pin
The Service-Pin is a special input/output of the *node* for service purposes. It is let to the outside on a pushbutton and an LED and sends broadcast news with the *Neuron* ID and Program ID of the *Neuron* chip when the pushbutton is actuated.

SNVT
Abbreviation for Standard Network Variable Types. SNVTs [pronounced snivits] are predefined standardized *network variables*. There are a number of SNVTs, e.g. SNVT_lux for brightness, SNVT_temp for temperatures, SNVT_switch for switching signals, etc. (See also *Network variables*)

Structured cabling
The structured cabling specifies rules for site, building and floor cabling. The maximum permissible cable lengths are defined (EIA/TIA 568, IS 11801) with recommendations for topologies.

Subnet
Subnets are partial nets and the next smallest addressing unit after the *Domain*. Due to subnet addressing, certain groups of units (e.g. a room or a production cell) are triggered. Subnets may contain a maximum of 127 units.

S-UTP
Abbreviation for "Screened Unshielded Twisted-Pair", which has only one external shield. However, the twisted pair cables are not shielded from each other.

Terminators
"Terminators" serve for the impedance based correct connection of a network on the basis of the *Twisted-Pair* technology. In accordance with the *transceivers* used and the topology used (bus or *free topology*), varying terminators are to be used per the *Echelon* specification. Terminators are also sometimes integrated in *LON* devices and can then generally be activated by switches or jumpers.

Note:
Missing or incorrect terminations of a network must not have an immediate effect, but may be the cause of irregularly occurring communication problems.

Timeout
Each network input variable, for which a timeout is specified must, must have been updated after the time set.
If the value of the *NVI* is not updated after the time set, the associated output is then set to the defined preferred position.
TP/FT-10
Abbreviation for "Transceiver Twisted Pair Free Topology" TP/FT-10, which is the most common transmission medium. The TP/FT-10 channel permits both the line bus topology, as well as free topology. As a line bus, again 64 subscribers can be connected to a segment up to a length of 2700 m. The transmission rate amounts to 78 kBit/sec. In a free topology, the network can be extended to up to 400 m with 64 units. TP/FT-10 allows the greatest degree of freedom in terms of space.

Transceiver
Transceivers are bus coupler modules between Neuron® chip and transmission medium. The most important representatives are TP/FT-10 and LPT-10. In addition, transceiver are available for radio transmission or for coupling with LWL systems.

Twisted Pair
Twisted pair cables (abbreviation: TP).

V

VFB (Virtual Functional Block)
To connect variables with a network interface, a virtual module, the "Virtual Functional Block" must be created for the interface. When first opening the LON® configurator, two objects have already been created in the tree structure for the LON® network interface that cannot be deleted or modified, namely the "Node Object" and the "Virtual Functional Block" (VFB). The "Virtual Functional Block" contains the NVs and CPs that can be assigned to the LON® FTT module directly and no other LonMark® object of the LON® FTT module.

W

Wink task
With a wink task, a user can search for a non-configured node. This makes sense when intended in the application noticeable in a defined way, e.g. by flashing STATUS LEDs, so that it is possible to create an allocation to physical nodes.

X

XIF file
A file with the "*.xif" ending is an export file that contains the description of the interface from a LON® device as a minimum variant. However, the XIF file normally contains more information such as manufacturer, version and more. This XIF file can be read from a LON® device and read into a LON® network for integration.
9 Appendix

The appendix provides further explanations about the function blocks listed below and additional information.

Table 28: Appendix overview

<table>
<thead>
<tr>
<th>Function block</th>
<th>see page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module X_LonMaster (LON-Master)</td>
<td>89</td>
</tr>
<tr>
<td>Module X_&lt;Name of the LMO&gt;</td>
<td>91</td>
</tr>
<tr>
<td>Transmission of network output variables</td>
<td>94</td>
</tr>
<tr>
<td>Receipt of network input variables</td>
<td>95</td>
</tr>
<tr>
<td>Use of configuration properties</td>
<td>95</td>
</tr>
<tr>
<td>ApplicationMessage</td>
<td>96</td>
</tr>
<tr>
<td>Module X_ApplicationMsgSend</td>
<td>97</td>
</tr>
<tr>
<td>Module X_ApplicationMsgSendRsp</td>
<td>98</td>
</tr>
<tr>
<td>FbDecode Modules</td>
<td>99</td>
</tr>
<tr>
<td>FbDecodeLevPercent</td>
<td>99</td>
</tr>
<tr>
<td>FbDecodeSnvtSetting</td>
<td>100</td>
</tr>
<tr>
<td>FbDecodeSnvtStrAsc</td>
<td>100</td>
</tr>
<tr>
<td>FbDecodeSwitch</td>
<td>101</td>
</tr>
<tr>
<td>FbDecodeTemp</td>
<td>101</td>
</tr>
<tr>
<td>FbDecodeTempP</td>
<td>101</td>
</tr>
<tr>
<td>FbEncode Modules</td>
<td>101</td>
</tr>
<tr>
<td>FbEncodeLevPercent</td>
<td>102</td>
</tr>
<tr>
<td>FbEncodeSnvtSetting</td>
<td>103</td>
</tr>
<tr>
<td>FbEncodeSnvtStrAsc</td>
<td>103</td>
</tr>
<tr>
<td>FbEncodeSwitch</td>
<td>104</td>
</tr>
<tr>
<td>FbEncodeTemp</td>
<td>104</td>
</tr>
<tr>
<td>FbEncodeTempP</td>
<td>105</td>
</tr>
</tbody>
</table>
9.1 Module_X_LonMaster (LON® Master)

Table 29: Description of Module_X_LonMaster

<table>
<thead>
<tr>
<th>Graphical illustration:</th>
</tr>
</thead>
</table>

Module_1_LonMaster

- ptInData
- ptOutData
- xRequestNeuronState
- xResetNeuronState

Figure 3: Graphical illustration of Module_X_LonMaster

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptInData</td>
<td>POINTER TO BYTE</td>
<td>Address of the processing input data.</td>
</tr>
<tr>
<td>ptOutData</td>
<td>POINTER TO BYTE</td>
<td>Address of the processing output data.</td>
</tr>
<tr>
<td>xRequestNeuronState</td>
<td>BOOL</td>
<td>Request of the Neuron status, response is stored in typNeuronState. Rising edge is evaluated.</td>
</tr>
<tr>
<td>xResetNeuronState</td>
<td>BOOL</td>
<td>Reset of the Neuron® status.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>wModuleState</td>
<td>WORD</td>
<td>Bit by bit OR-operated status of the module:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16#8000: Communication error between PLC program and LON® FTT module</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16#0080: Configuration of the internal LON network interface ready (the Configuration has been prepared and is waiting for the Start command from the LON® master function block, with which initialization is completed).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16#0040: Configuration of the LON® network interface fully initialized and active (the configuration was started successfully and is now active).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16#0020: LON® network interface configuration active (if this flag is not set, the LON® FTT module is only represented with a standard interface that consists of only one node object.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16#0010: Internal error in the LON® FTT module (the internal LON® network interface is not ready for communication)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16#0008: CPs are synchronized between the LON® FTT module and PLC application.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16#0004: LonWorks® device status is &quot;configured online&quot;.</td>
</tr>
</tbody>
</table>
| **typNeuronState** | **NeuronState** | Data type from LON_01.lib, TYPE NeuronState: STRUCT  
| | |  
| | | wCrcError : WORD;  
| | | wTransactionTimeouts : WORD;  
| | | wReceiveTransactionBuffersFull : WORD;  
| | | wLostMessages : WORD;  
| | | wMissedMessages : WORD;  
| | | bResetCause : BYTE;  
| | | bNodeState : BYTE;  
| | | bFirmwareVersion : BYTE;  
| | | bLastErrorLogged : BYTE;  
| | | bModelNumber : BYTE;  
| | | END_STRUCT  
| | | END_TYPE |  
| **abNeuronId** | **ARRAY[0..5] OF BYTE** | Neuron ID of the LON® FTT module. |  
| **atypDomainData** | **ARRAY[0..1] OF DomainData** | Data type from the LON_01.lib, Domain table entries 0 and 1 of the LON® FTT module  
| | | TYPE DomainData : STRUCT  
| | | abDomainId : ARRAY[0..5] OF BYTE;  
| | | bSubnet : BYTE;  
| | | xIsNonClone : BOOL;  
| | | bNodeId : BYTE;  
| | | xIsInvalid : BOOL;  
| | | bDomainIdLength : BYTE;  
| | | abAuthenticationKey : ARRAY[0..5] OF BYTE;  
| | | END_STRUCT  
| | | END_TYPE |  
| **typApplicationMessage** | **ApplicationMessageData** | Data type from the LON_01.lib, Signals receipt of a LonWorks® application message and contains the data associated with the message.  
| | | xMsgRcvd signals receipt of a new message and in this case, is TRUE for a PLC cycle.  
| | | xRspRcvd signals receipt of a new RESPONSE message and in this case, is TRUE for a PLC cycle |  
| | | TYPE ApplicationMessageData : STRUCT  
| | | xMsgRcvd : BOOL;  
| | | xRspRcvd : BOOL;  
| | | typService : ServiceType;  
| | | bSourceSubnetId : BYTE;  
| | | bSourceNodeId : BYTE;  
| | | bCorrelator : BYTE;  
| | | bCode : BYTE;  
| | | bDataLength : BYTE;  
| | | abData : ARRAY[0..47] OF BYTE;  
| | | END_STRUCT  
| | | END_TYPE |
9.2 Module_X_<Name of the LMO>

Because the structure of the modules for LonMark® objects is variable, two different examples are provided here.

Table 30: Description of Module_X_<Name of the LMO>

<table>
<thead>
<tr>
<th>Graphical illustration:</th>
</tr>
</thead>
</table>

Example 1:
This example is an object each with a network variable in input and output direct of type "SNVT_switch". In addition, a "switchMaxSendTime" and "nciLocation" CP are implemented as NVs.

![Figure 4: Graphical illustration of LMO module (example 1)](image)

Example 2
This example is an object declared as a field (see also figure "Correlation between user interface and generated modules") with a network output variable.

![Figure 5: Graphical illustration of LMO module (example 2)](image)

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bObjectIndex</td>
<td>BYTE</td>
<td>Index of the LMO if the object is declared as a field. The default setting is 0 and does not need to be assigned, for LMOs that are not a field.</td>
</tr>
</tbody>
</table>
For each configured NVO, an input is created. An NVO is always sent when a value changes. Using bit 7 of the structural element bStatusInformation also requires a dispatch even for value parity.

Example typSnvtSwitch:

```plaintext
TYPE typSnvtSwitch :
STRUCT
 bValue : BYTE;
 siState : SINT;
 bStatusInformation : BYTE; (* bit allocation of bStatusInformation: 
 bit 0: NVI update counter LSB
 bit 1: NVI update counter
 bit 2: NVI update counter
 bit 3: NVI update counter MSB
 bit 4: timeout for NVI occurs (MaxRcvTime expired)
 bit 5: reserved
 bit 6: reserved
 bit 7: force NVO update *)
 bInternalUseOnly : BYTE; (* Do not use this byte in your application! *)
END_STRUCT
END_TYPE
```

The structural elements bStatusInformation and bInternalUseOnly have all data types generated in WAGO-I/O-PRO in common. The top elements customized depending on NVT.

The element bInternalUseOnly is only used for coordination of communication between modules and may not be changed by the use PLC application!

<table>
<thead>
<tr>
<th>Input/output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
</table>
| <Name of the configured CP> | POINTER TO <configured CPT> or POINTER TO ARRAY[0..Y] OF <configured NVT> | For each configured CP, an input is created. A CP is always sent when a value changes. Using bit 7 of the structural element bStatusInformation also requires a dispatch even for value parity. A CP received from the LON® network is applied to the address of the variables attached to the input. For a description of the data type, see <Name of the configured NVO>.
<table>
<thead>
<tr>
<th>Output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Name of the configured NVI&gt;</td>
<td>&lt;configured NVT&gt; or ARRAY[0..Y] OF &lt;configured NVT&gt;</td>
<td>For each configured NVI, an output is created. When a generic CP SCPTmaxRevTime is assigned for an NVI, a timeout is displayed by bit 4 of the structural element bStatusInformation. For a description of the data type, see &lt;Name of the configured NVO&gt;.</td>
</tr>
<tr>
<td>&lt;Name of the configured CP implemented as an NV&gt;</td>
<td>POINTER TO &lt;configured CPT&gt; or POINTER TO ARRAY[0..Y] OF &lt;configured NVT&gt;</td>
<td>For each configured CP implemented as an NV, an output is created. For a description of the data type, see &lt;Name of the configured NVO&gt;.</td>
</tr>
</tbody>
</table>

This function block represents a LonMark® object with all NVs and CP set in the user interface.

The names of the inputs and outputs correspond to the names specified in the user interface.

For NV or CP data type contained in the LMO, a corresponding WAGO-I/O-PRO data type is created. The top elements of the WAGO-I/O-PRO data type correspond to the structure of the NVTs or CPTs. These contain the user data of the NVs or CPs.

The structure elements **bStatusInformation** and **bInternalUseOnly** have all WAGO-I/O-PRO data types in common.

```plaintext
TYPE typ<NV type designation> :
  STRUCT
    <Element 1> : <Base data type>; ↩ User data
    <Element 2> : <Base data type>; ↩ User data
    ...
    <Element n> : <Base data type>; ↩ User data
    bStatusInformation : BYTE; ↩ Status and control information
    bInternalUseOnly : BYTE; ↩ Status and control information
  END_STRUCT
END_TYPE
```

The element **bStatusInformation** for NVIs contains status information for network variables. For NVOs, controller information can be transmitted to the function block. For CPs, this element has no significance.

The element **bInternalUseOnly** is only used for coordination of communication between the modules.
Note

Do not change the "bInternalUseOnly" element!
Please note that the `bInternalUseOnly` element may not be changed by your PLC application program!

For further processing within a PLC main program, it is recommended that global variables for the NVs and CPs used be created.

**Example:**

```plaintext
VAR_GLOBAL
  nvoSwitch1 : typSnvtSwitch;
nviSwitch1 : typSnvtSwitch;
cpMaxSendTimeNvoSwitch1 : typScptmaxsendtime;
nciLocationUserController : typScptlocation;
etc. ...
END_VAR
```

As shown in Figure 6 below, these must be connected to the associated inputs or outputs of the function block.

CPs are readable and writable and must be linked to the ADR function as a reference.

![Figure 6: Example of linked variable](image)

CPs implemented as NVs are only available as outputs and are handled within the PLC application as NVIs.

**9.2.1 Transmission of network output variables**

The created (global) variables for NVOs must be linked to the respective inputs of the function block.

NVOs are only passed for sending to the LON® FTT module when values change during the previous PLC cycle.

To send a network variable independent of a value change, bit 7 of the `bStatusInformation` byte of the WAGO-I/O-PRO data type of the NVO must be set to TRUE.
Reset bit by application!
Please note that evaluation of the bit is not edge-controlled. You must ensure that the bit is reset in your application!

The remaining bits (0 - 6) from **bStatusInformation** have no relevance for NVOs.

### 9.2.2 Receipt of network input variables

The created (global) variables for NVIs must be linked to the respective outputs of the function block.

More information is available in the **bStatusInformation** byte for NVIs:

Table 31: Meaning of the bits for bStatusInformation

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 .. 3</td>
<td>Number of NV updates received in the current cycle for this NVI. Values for the same NV may be received several times within a PLC cycle. The messages are successively removed from the message queue in the LonMaster, so that only the last received value is displayed in the variables of the NV (no intermediate value). To mark how often the NV is received in one cycle, the bits of the LOW nibble (bit 0 - 3) is available as a counter, so that receipt of up to 15 messages can be displayed. If more than 15 messages are received, the counter remains at 15. Normally, the counter for a PLC cycle stands at 1 if the NVI has been received.</td>
</tr>
<tr>
<td>4</td>
<td>If the NVI timeout is monitored (a CP <code>SCPTmaxRevTime</code> has been assigned to the variables), bit 4 is set to TRUE if a timeout occurs (no receipt within the time set). The bit is reset when variables are received again.</td>
</tr>
<tr>
<td>5 .. 7</td>
<td>Not in use</td>
</tr>
</tbody>
</table>

CPS implemented as NVIs are handled within the PLC application as NVIs. The meaning of the bits for bStatusInformation is similar as with the NVIs.

### 9.2.3 Use of configuration properties

CPS are kept in the persistent memory of the LON® FTT module.

After launching the PLC program, the CPS are synchronized (transmission of the CP values from the LON® FTT module to the PLC application). CP synchronization can also start when CPS are modified by the LonWorks® network management tool used. The CP synchronization phase requires a certain number of PLC cycles depending on the volume of (modified) configuration data.
The values of the CPs are only valid when CP synchronization between the PLC application and the LON® FTT module is complete. The \texttt{wModuleState} variable of the "\texttt{Module\_X\_LonMaster}" module signals active CP synchronization by the bit mask 16\#0008. Only if the status message is cancelled are the CP values valid.

**Note**

\textbf{Take into account bit masks for validity of the CP values!}

Please notes that for complete assessment of the validity, it is necessary to also take into account the bit masks 16\#0040 and 16\#0020 from "\texttt{wModuleState}" (see chapter "Module\_X\_LonMaster")!

The values of the CPs can be read and written by the PLC application.

If the value of a CP is modified by the PLC application, the new value is transferred to the persistent memory of the LON® FTT module.

**Note**

\textbf{Note the maximum number of write cycles of the EEPROM!}

Please note that the CPs are stored in the nonvolatile memory of the LON® FTT module. This memory allows max. 1,000,000 write cycles.

### 9.3 ApplicationMessage

**Note**

\textbf{Only use"ApplicationMessage" function blocks selectively!}

Use the function blocks for sending and acknowledging LonWorks® application messages (explicit messages) if you use them explicitly. These function blocks affect the LON® network directly and thus represent a high risk of collision between the network commands to the bus nodes and the processing of the application.
### 9.3.1 Module_X_ApplicationMsgSend

Table 32: Description of Module_X_ApplicationMsgSend

**Graphical illustration:**

![Module_X_ApplicationMsgSend Diagram](image)

**Figure 7:** Graphical illustration of Module_X_ApplicationMsgSend

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>xSend</td>
<td>BOOL</td>
<td>Starts transmission if TRUE. The rising edge is evaluated. To send again, the input must be set to FALSE in advance.</td>
</tr>
<tr>
<td>abAddress</td>
<td>ARRAY [0..10 ] OF BYTE</td>
<td>Field for the target address of the device to which the message should be sent. To create the target address, the following functions are available from the LON_01.lib:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- FuMakeNeuronIdAddress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- FuMakeSubnetNodeAddress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- FuMakeBroadcastAddress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- FuMakeGroupAddress</td>
</tr>
<tr>
<td>typServiceSend</td>
<td>ServiceType</td>
<td>Service with which the message should be sent. TYPE ServiceType :</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LON_SRVC_ACK := 0,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LON_SRVC_UNACK := 16#10,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LON_SRVC_UNACK_RPT := 16#20,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LON_SRVC_REQ := 16#30,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LON_RESPONSE := -1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>);</td>
</tr>
<tr>
<td></td>
<td></td>
<td>END_TYPE</td>
</tr>
<tr>
<td>xPriority</td>
<td>BOOL</td>
<td>If TRUE, the message is sent prioritized.</td>
</tr>
<tr>
<td>xAuthentification</td>
<td>BOOL</td>
<td>TRUE if the message should be authenticated.</td>
</tr>
<tr>
<td>bMsgCodeSend</td>
<td>BYTE</td>
<td>Telegram code of the message. Permissible values: 0 … 61. Corresponds to the range for user-specific messages with exception of telegram code 62 that is reserved for LonTalk® FTP and that is used internally.</td>
</tr>
</tbody>
</table>

**Note:**

Do not use reserved telegram code 62!

Please note that telegram code 62 is reserved for transmission of configuration properties via LonTalk® FTP and it may not be used!
This function block can be used to send application messages.

Application code, user data, target address and other parameters can be sent. Sending the message is triggered with a positive edge on the "xSend" input.

### Module_X_ApplicationMsgSendRsp

This function block can be used to respond to request messages (application messages received with the "LON_SRVC_REQ" service type).

Application code, user data, target address and other parameters can be sent. Sending the message is triggered with a positive edge on the "xSend" input.

The "bCorrelator" byte should be noted that must be applied from the message received.
Messages received are displayed at the "Module_X_LonMaster" at the "typApplicationMessage" output parameter.

9.4 FbDecode Modules

The FbDecode modules allow convenient scaling of selected SNVTs in IEC base data types.

The module input can be applied from an LMO representative "Module_X_<Name of the LMO>" directly.(see chapter "Module_X_<Name of the LMO>").

Existing scaling factors acc. LonMark® SNVT master list are taken into account when converting from the modules.

Example:
In the following example, use based on an NVI of type "SNVT_switch" is demonstrated.

The member variable "bValue" is used according to the SNVT master list to map an unsigned 8-bit integer value (raw value) to an analog value (value range: 0.0 … 100.0%, resolution: 0.5%).

Table 34: Description of FbDecodeLevPercent

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value %</td>
<td>typSnvtLevPercent</td>
<td>Raw value to be converted</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>rScaledValue</td>
<td>REAL</td>
<td>Scaled value of the element</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value.iSnvtLevPercent.</td>
</tr>
</tbody>
</table>

Figure 9: Connecting a FbDecode module (example)

If in the example shown here for "nviSwitch" of the module of type "Module_1_UserController", the value "nviSwitch.bValue" = 101 and "nviSwitch.siState" = 0 is received, then the variables receive the following values:
• mySwitch1Value = 55.5
• mySwitch1State = FALSE

9.4.1 FbDecodeLevPercent

Graphical illustration: Figure 10: Graphical illustration of FbDecodeLevPercent
### 9.4.2 FbDecodeSnvtSetting

Table 35: Description of FbDecodeSnvtSetting

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value %</td>
<td>typSnvtSetting</td>
<td>Raw value to be converted</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>eFunction</td>
<td>setting_t</td>
<td>Scaled value of the element value.bFunction.</td>
</tr>
<tr>
<td>rSetting</td>
<td>REAL</td>
<td>Scaled value of the element value.bSetting.</td>
</tr>
<tr>
<td>rRotation</td>
<td>REAL</td>
<td>Scaled value of the element value.iRotation.</td>
</tr>
</tbody>
</table>

**Graphical illustration:**

[Figure 11: Graphical illustration of FbDecodeSnvtSetting]

```plaintext
TYPE setting_t :
(  SET_OFF := 0,
    SET_ON := 1,
    SET_DOWN := 2,
    SET_UP := 3,
    SET_STOP := 4,
    SET_STATE := 5,
    SET_NUL := -1 );
END_TYPE
```

### 9.4.3 FbDecodeSnvtStrAsc

Table 36: Description of FbDecodeSnvtStrAsc

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value %</td>
<td>typSnvtStrAsc</td>
<td>Raw value to be converted</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>scaledValue</td>
<td>STRING(31)</td>
<td>Resulting character string of the element value.abAscii.</td>
</tr>
</tbody>
</table>

**Graphical illustration:**

[Figure 12: Graphical illustration FbDecodeSnvtStrAsc]
9.4.4 **FbDecodeSnvtSwitch**

Table 37: Description of FbDecodeSnvtSwitch

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value %</td>
<td>typSnvtSwitch</td>
<td>Raw value to be converted</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Scaled value of the element value.bValue.</td>
</tr>
<tr>
<td>xState</td>
<td>BOOL</td>
<td>Scaled value of the element value.siState.</td>
</tr>
</tbody>
</table>

9.4.5 **FbDecodeSnvtTemp**

Table 38: Description of FbDecodeSnvtTemp

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value %</td>
<td>typSnvtTemp</td>
<td>Raw value to be converted</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>rScaledValue</td>
<td>REAL</td>
<td>Scaled value of the element value.wSnvtTemp.</td>
</tr>
</tbody>
</table>

9.4.6 **FbDecodeSnvtTempP**

Table 39: Description of FbDecodeSnvtTempP

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value %</td>
<td>typSnvtTempP</td>
<td>Raw value to be converted</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>rScaledValue</td>
<td>REAL</td>
<td>Scaled value of the element value.iSnvtTempP.</td>
</tr>
</tbody>
</table>

9.5 **FuEncode Modules**

The FuEncode modules allow convenient scaling of IEC base data types in selected SNVTs.
The return value can be applied to an LMO representative "Module_X_<Name of the LMO>" directly (see chapter "Module_X_<Name of the LMO>").

Existing scaling factors acc. LonMark® SNVT master list are taken into account when converting from the modules.

Example:
In the following example, use based on an NVO of type "SNVT_switch" is demonstrated.

The member variable "bValue" is used according to the SNVT master list to map an analog value(value range: 100.0 … 0.5%, resolution: 8%) to an unsigned 8-bit integer value (raw value).

With the input values used here in the example shown, the following values are written to the input variable "nvoSwitch" of the module of type "Module_1_UserController":
- nvoSwitch.bValue = 200
- nvoSwitch.sIState = 1

9.5.1 FuEncodeSnvtLevPercent

Table 40: Description of FuEncodeSnvtLevPercent

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Scaled value that as a raw value should be converted into the NVT structure. Value range: -163.840 - 163.830 Resolution: 0.005</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return value</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>FuEncodeSnvtLevPercent</td>
<td>typSnvtLevPercent</td>
<td>Converted value</td>
</tr>
</tbody>
</table>
9.5.2  **FuEncodeSnvtSetting**

Table 41: Description of FuEncodeSnvtSetting

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
</table>
| eFunction       | setting_t   | TYPE setting_t: :
|                 |             | ( SET_OFF := 0, 
|                 |             |   SET_ON := 1, 
|                 |             |   SET_DOWN := 2, 
|                 |             |   SET_UP := 3, 
|                 |             |   SET_STOP := 4, 
|                 |             |   SET_STATE := 5, 
|                 |             |   SET_NUL := -1 
|                 |             | ); END_TYPE                                                             |
| rSetting        | REAL        | Scaled value that as a raw value should be converted into the NVT structure. 
|                 |             | Value range: 0.0 - 100.0 
|                 |             | Resolution: 0.5                                                         |
| rRotation       | REAL        | Scaled value that as a raw value should be converted into the NVT structure. 
|                 |             | Value range: -359.98 - 360.00 
|                 |             | Resolution: 0.02                                                        |

**Return value**

<table>
<thead>
<tr>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>typSnvtSetting</td>
<td>Converted value</td>
</tr>
</tbody>
</table>

9.5.3  **FuEncodeSnvtStrAsc**

Table 42: Description of FuEncodeSnvtStrAsc

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>scaledValue</td>
<td>STRING(31)</td>
<td>Character string that should be converted into the NVT structure.</td>
</tr>
</tbody>
</table>

**Return value**

<table>
<thead>
<tr>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>typSnvtStrAsc</td>
<td>Converted value</td>
</tr>
</tbody>
</table>
### 9.5.4 FuEncodeSnvtSwitch

Table 43: Description of FuEncodeSnvtSwitch

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Scaled value that as a raw value should be converted into the NVT structure. Value range: 0.0 - 100.0 Resolution: 0.5</td>
</tr>
<tr>
<td>xState</td>
<td>BOOL</td>
<td>Boolean value that as a raw value should be converted into the NVT structure.</td>
</tr>
</tbody>
</table>

#### Graphical illustration:

![Figure 20: Graphical illustration of FuEncodeSnvtSwitch](image)

### 9.5.5 FuEncodeSnvtTemp

Table 44: Description of FuEncodeSnvtTemp

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Scaled value that as a raw value should be converted into the NVT structure. Value range: -274.0 - 6279.5 Resolution: 0.1</td>
</tr>
</tbody>
</table>

#### Graphical illustration:

![Figure 21: Graphical illustration of FuEncodeSnvtTemp](image)
9.5.6 FuEncodeSnvtTempP

Table 45: Description of FuEncodeSnvtTempP

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Scaled value that as a raw value should be converted into the NVT structure. Value range: -273.17 - 327.66, Resolution: 0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return value</th>
<th>Data type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>FuEncodeSnvtTempP</td>
<td>typSnvtTempP</td>
<td>Converted value</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1: Principle of the LonWorks® network connection to the WAGO-I/O-System 750................................................................. 14
Figure 2: Schematic representation of the definition and use of the LON® network interface.......................................................... 16
Figure 1: Screenshot of the WAGO-I/PRO PLC configuration ................................................................. 20
Figure 2: Configure LON® interface" user interface tab with default settings .... 21
Figure 5: General entries ................................................................................................................ 22
Figure 6: "Configure Program ID" dialog .......................................................................................... 23
Figure 7: Configuration area of the LON® network interface (example view).... 25
Figure 8: Example of a search filter entry with the string "sett" ........................................ 26
Figure 9: Context menu in the tree structure of the LON® network interface ........ 30
Figure 10: Screenshot of the object settings (example) ......................................................... 33
Figure 11: Screenshot of the settings of a network variable (example) ................ 33
Figure 12: Screenshot of the data type field for NVs for generic LMOs (example) .......................................................................................................................... 34
Figure 13: Screenshot of the settings of a configuration property (CP) (example) .......................................................................................................................... 35
Figure 14: Dialog for setting CPs, example: SCPTinstalledLevel.......................... 36
Figure 15: Dialog for setting CPs, example: SCPTdefaultSetting ......................... 37
Figure 16: Example views of the function / status area ........................................ 38
Figure 17: Generated function blocks in WAGO-I/O-PRO........................................ 40
Figure 18: Directory structure of the function blocks (overview)........................... 40
Figure 19: Defined data types in the integrated LON_01.lib library ..................... 46
Figure 20: Example project 1: Modules declared individually, code size 325494 bytes .................................................................................................................... 54
Figure 21: Example project 2: Modules declared as ARRAYS, code size 101542 bytes .................................................................................................................... 55
Figure 22: Example project 3: no optimization, worst case, code size 278530 bytes .................................................................................................................... 56
Figure 23: Example project 4: with optimization, code size 126522 bytes .......... 57
Figure 24: Example project 5: with optimization, code size 117630 bytes .......... 58
Figure 25: Example project 6: with optimization, code size 113874 bytes .......... 59
Figure 26: Example project 7: with optimization, code size 127498 bytes .......... 60
Figure 1: Configuration diagram of a LON® bus to the LON® FTT module........ 62
Figure 28: Integrate library "LON_01.lib" (example)............................................. 64
Figure 29: Configure program ID (example) ............................................................ 65
Figure 30: General settings (example) ................................................................. 66
Figure 31: Configure LON® interface (example "SNVT_switch").......................... 66
Figure 32: Carry out the required object settings (example "SNVT_switch")...... 67
Figure 33: Function blocks and LON® master module in WAGO-I/O-PRO (example) .......................................................................................................................... 68
Figure 34: Input assistant - User defined function blocks (example) ................. 69
Figure 35: Variable declaration (example "nviswitch") ........................................ 70
Figure 36: Input assistant - User defined types (example) .................................... 70
Figure 37: Create "Virtualfunctionalblock" function block (example) ............... 71
Figure 38: Module_1_PRG program block call for the LON® FTT module in the main program .......................................................................................................................... 72
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>Start screen of the LNMT LonMaker® from Echelon® (example)</td>
</tr>
<tr>
<td>40</td>
<td>LonMaker® network wizard (example)</td>
</tr>
<tr>
<td>41</td>
<td>Create a new device on the graphic interface (example)</td>
</tr>
<tr>
<td>42</td>
<td>Create device – Settings (example)</td>
</tr>
<tr>
<td>43</td>
<td>Upload interface definition (example)</td>
</tr>
<tr>
<td>44</td>
<td>Prompt to press the service pin</td>
</tr>
<tr>
<td>45</td>
<td>Create LON® FTT module on the graphic interface (example: Switch)</td>
</tr>
<tr>
<td>46</td>
<td>Create &quot;Virtual Functional Block&quot;</td>
</tr>
<tr>
<td>47</td>
<td>Automatically create shapes</td>
</tr>
<tr>
<td>48</td>
<td>Test communication in the LonMaker® browser (example)</td>
</tr>
<tr>
<td>49</td>
<td>Test communication, variable change in the IEC application (example)</td>
</tr>
</tbody>
</table>
## List of Tables

Table 1: Number Notation........................................................................................................ 9
Table 1: Font Conventions ...................................................................................................... 9
Table 1: PC hardware requirements ....................................................................................... 12
Table 3: Required software .................................................................................................... 12
Table 4: Optional software .................................................................................................... 12
Table 2: Required Components of the WAGO-I/O-SYSTEM ................................................ 13
Table 7: Configuration procedure overview ........................................................................ 15
Table 6: "General entries" area .......................................................................................... 22
Table 7: "Configure Program ID" dialog ............................................................................. 24
Table 8: Area of the dialog for creating the LON® network interface ................................... 25
Table 9: Arrow buttons ........................................................................................................ 27
Table 10: Adding objects from the LonMark® resource selection to the LON® network interface .......................................................... 29
Table 11: Context menu in the tree structure of the LON® network interface ...................... 30
Table 12: Actions for elements in the tree structure of the LON® network interface .......... 31
Table 13: Symbols in the tree structure of the LON® network interface ............................ 32
Table 14: Object settings of a LonMark® Object (LMO) ...................................................... 33
Table 15: Settings of a network variable (NV) ................................................................. 34
Table 16: Settings of a configuration property (CP) ......................................................... 35
Table 17: Fields for the default values of a CP ............................................................... 37
Table 18: Elements of the function / status area ............................................................... 39
Table 1: Program call sequence (minimum) ..................................................................... 41
Table 2: Cross-module function blocks in the directory "LON®→Shared" ........................... 42
Table 21: Generated function blocks in the module-specific "LON®→Module_X" directory .................................................................................................................. 43
Table 22: Module-specific function blocks in "LON®→Module_X→ApplicationMessage" .................................................................................................................. 44
Table 23: Generic configuration properties ..................................................................... 49
Table 24: Restrictions ......................................................................................................... 61
Table 1: Example hardware for a fieldbus node setup ....................................................... 62
Table 1: Appendix overview .......................................................................................... 88
Table 2: Description of Module_X_LonMaster ............................................................. 89
Table 3: Description of Module_X_<Name of the LMO> ............................................. 91
Table 4: Meaning of the bits for bStatusInformation .................................................. 95
Table 5: Description of Module_X_ApplicationMsgSend ........................................... 97
Table 6: Description of Module_X_ApplicationMsgSendRsp ...................................... 98
Table 7: Description of FbDecodeLevPercent .............................................................. 99
Table 8: Description of FbDecodeSnvtSetting ............................................................. 100
Table 9: Description of FbDecodeSnvtStrAsc ............................................................ 100
Table 10: Description of FbDecodeSnvtSwitch ............................................................ 101
Table 11: Description of FbDecodeSnvtTemp .............................................................. 101
Table 12: Description of FbDecodeSnvtTempP ............................................................ 101
Table 13: Description of FuEncodeSnvtLevPercent .................................................. 102
Table 14: Description of FuEncodeSnvtSetting .......................................................... 103
Table 15: Description of FuEncodeSnvtStrAsc .......................................................... 103
Table 16: Description of FuEncodeSnvtSwitch .......................................................... 104
Table 17: Description of FuEncodeSnvTemp.................................................... 104
Table 18: Description of FuEncodeSnvTempP.............................................. 105