Documentation of the library
WagoAppDMX
Release 1.0.1.0
1 Description

2 10 Documentation
2.1 doc01_Foreword (FB) ............................................................... 3

3 20 Program Organization Units
3.1 01 Communication / Kommunikation ........................................... 4
  3.1.1 FbDmxMaster (FB) ............................................................... 4
    00 Administration .................................................................. 5
    FbDmxMaster.Job (METH) ...................................................... 5
    01 Parameters ................................................................. 5
    FbDmxMaster.Blackout (PROP) ............................................... 5
    FbDmxMaster.NumberOfChannel (PROP) .................................. 5
    FbDmxMaster.TxTrigger (PROP) ............................................ 5
  3.1.2 FbDmxSlave (FB) ............................................................... 5
    00 Administration .................................................................. 6
    FbDmxSlave.Job (METH) ...................................................... 6
    01 Parameters ................................................................. 6
    FbDmxSlave.StartChannel (PROP) ......................................... 6
    FbDmxSlave.TxTrigger (PROP) ............................................ 6
  3.2 02 Channel values / Kanalwerte ................................................. 6
    3.2.1 FuDmxGetChannel (FUN) ............................................... 6
    3.2.2 FuDmxSetChannel (FUN) ............................................... 6
  3.3 80 Data types ......................................................................... 7
    3.3.1 typDMX (STRUCT) .......................................................... 7

4 80 Status
4.1 Status (GVL) ......................................................................... 8
4.2 eStatus (ENUM) .................................................................. 9

5 GlobalVariables (GVL) ............................................................... 10

6 ParameterList (PARAMS) ............................................................ 11

7 VersionHistory (GVL) ............................................................... 12

8 Library Reference .................................................................. 13
CHAPTER 1

Description

This document is automatically generated. Because of this, the chapter 30 Visualization is not shown in this document. If you are interested in getting to know more about visualization, we refer to the library manager of e!Cockpit.

Subject to Changes

WAGO Kontakttechnik GmbH Co. KG reserves the right to provide for any alterations or modifications. 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.

Personnel Qualification

All tasks that are carried out with libraries made for the e!COCKPIT software must only be performed by qualified electrical specialists instructed in PLC programming according to IEC 61131-3.

All tasks that have an effect on the properties or the behavior of automation hardware or software products must only be performed by qualified employees with a thorough knowledge of handling the products concerned.

Intended Use of e!COCKPIT Libraries

Libraries created for the e!COCKPIT software are used to simplify the development of application projects in the IEC 61131-3 programming languages.

For automation tasks, WAGO offers programmable logic controllers in a wide variety of performance classes. In combination with a wide range of I/O modules, the controllers can process standard types of field signals. Controllers can be implemented centrally or in decentralized configurations. The controllers offer interfaces for the most commonly used fieldbuses for use in decentralized configurations. Fieldbus independent I/O modules are then linked via fieldbus couplers. WAGO controllers offer a runtime environment for user programs called e!RUNTIME. Software projects for implementation in e!RUNTIME environments can be created in e!COCKPIT. The programming environment in e!COCKPIT is based on the established CODESYS 3 industrial standard. Users with a previous knowledge of CODESYS 3 will thus find this environment largely familiar. The following programming languages of the IEC 61131-3 standard are available:

- Structured Text (ST)
- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Instruction List (IL)
- Sequential Function Chart (SFC)
- Continuous Function Chart (CFC)

The individual programming languages can also be combined as required during the development of the software. A portfolio of prepared libraries can be accessed for many frequently used functions in order to make software development more efficient. This document provides an overview of the WagoAppDMX that WAGO offers for e!COCKPIT.
Library for DMX512 communication with Wago system

Further library information are summerized here:

- **Company**: WAGO
- **Title**: WagoAppDMX
- **Version**: 1.0.1.0
- **Categories**: WAGO BusinessView|Building Automation; WAGO LayerView|App; Application
- **Author**: WAGO / u015842
- **Placeholder**: WagoAppDMX
To ensure fast installation and start-up of the units, we strongly recommend that the following information and explanations are carefully read and adhered to.

2.1 doc01_Foreword (FB)

Copyright

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

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

Personnel Qualification

The use of the product described in this document is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards. WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

Intended Use

For each individual application, the components are supplied from the factory with a dedicated hardware and software configuration. Modifications are only admitted within the framework of the possibilities documented in this document. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG.

Please direct any requirements pertaining to a modified and/or new hardware or software configuration directly to WAGO Kontakttechnik GmbH & Co. KG.

Scope of Applicability

This application note is based on the stated hardware and software from the specific manufacturer, as well as the associated documentation. This application note is therefore only valid for the described installation. New hardware and software versions may need to be handled differently.

Please note the detailed description in the specific manuals.
3.1 01 Communication / Kommunikation

3.1.1 FbDmxMaster (FB)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
<th>Inherited from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>I_Port</td>
<td>WagoType-sCom.I_WagoSysDmx</td>
<td></td>
<td>Serail 750-652 module</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bPortDmx</td>
<td>BYTE</td>
<td>1</td>
<td>Master assignment number. Range 1 to MAX_MODULE.</td>
<td>FbDmxSerial</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Status object. (Listed in Status) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
<td>FbDmxSerial</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Status description as string (Listed in Status)</td>
<td>FbDmxSerial</td>
</tr>
<tr>
<td>Input</td>
<td>iNumberOfChannel</td>
<td>INT</td>
<td>21</td>
<td>Number of DMX channels to send. Range 1-512.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xBlackout</td>
<td>BOOL</td>
<td>TRUE-&gt; blackout mode is active. All DMX Channel values will remain zero.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function

This function block is for sending cyclic data bytes to the DMX line.

Note: The DMX function is only possible using WAGO 750-652 Serial Module from firmware version 03. For FW 03 until FW05, maximum 21 or 45 DMX channel can be sent depends on the Process Image setting of the module. From FW 06 maximum 512 DMX channel can be sent.

Graphical Illustration

![Function Block Diagram]
00 Administration

FbDmxMaster.Job (METH)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Job</td>
<td>WagoSysErrorBase.FbResult</td>
</tr>
</tbody>
</table>

01 Parameters

FbDmxMaster.Blackout (PROP)

FbDmxMaster.NumberOfChannel (PROP)

FbDmxMaster.TxTrigger (PROP)

3.1.2 FbDmxSlave (FB)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
<th>Inherited from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>I_Port</td>
<td>WagoType-</td>
<td></td>
<td>Serail 750-652 module</td>
<td>FbDmxSerial</td>
</tr>
<tr>
<td></td>
<td>sCom.I_WagoSysDmx</td>
<td>sys</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>bPortDmx</td>
<td>BYTE</td>
<td>1</td>
<td>Master assignment number. Range 1 to MAX_MODULE.</td>
<td>FbDmxSerial</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSys-</td>
<td></td>
<td>Status object. (Listed in Status) The content of the error object could be</td>
<td>FbDmxSerial</td>
</tr>
<tr>
<td></td>
<td>sError-Base.FbResult</td>
<td>sys</td>
<td></td>
<td>displayed via the FbShowResult from the WagoSysErrorBase library.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Status description as string (Listed in Status)</td>
<td>FbDmxSerial</td>
</tr>
<tr>
<td>Input</td>
<td>uiStartChannel</td>
<td>UINT</td>
<td>1</td>
<td>Start DMX channel to read. Range 1 to 512.</td>
<td></td>
</tr>
</tbody>
</table>

Function

This function block is for reading data bytes from the DMX line.

Note: The DMX function is only possible using WAGO 750-652 Serial Module from firmware version (06). Maximum 21 DMX value can be read.

Note: The starting channel uiStartChannel must be similar with the configuration of the serial I/O module. The serial I/O module can be configured with WAGO-I/O-CHECK. Including the starting channel number, up to 21 consecutive channels can be read.

Graphical Illustration
00 Administration

FbDmxSlave.Job (METH)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Job</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
</tr>
</tbody>
</table>

01 Parameters

FbDmxSlave.StartChannel (PROP)

FbDmxSlave.TxTrigger (PROP)

3.2 02 Channel values / Kanalwerte

3.2.1 FuDmxGetChannel (FUN)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bPortDmx</td>
<td>BYTE</td>
<td>Master assignment number. Range 1 to MAX_MODULE.</td>
</tr>
<tr>
<td>Input</td>
<td>iChannel</td>
<td>INT</td>
<td>Address of the DMX channel</td>
</tr>
</tbody>
</table>

Function

This function returns DMX value from DMX universe.

Graphical Illustration

3.2.2 FuDmxSetChannel (FUN)

Interface variables
Scope | Name          | Type | Comment                                           |
-------|---------------|------|---------------------------------------------------|
Return | FuDmxSetChannel | BOOL |                                                   |
Input  | bPortDmx             | BYTE | Master assignment number. Range 1 to MAX_MODULE. |
       | iChannel              | INT  | Address of the DMX channel                        |
       | bValue                | BYTE | DMX value                                         |

**Function**

This function is for setting value to the DMX universe.

**Graphical Illustration**

![Function Illustration]

### 3.3 80 Data types

#### 3.3.1 typDMX (STRUCT)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>abTxBuffer</td>
<td>ARRAY [0..MAX_CHANNEL] OF BYTE</td>
<td>DMX channel values</td>
</tr>
<tr>
<td>abRxBuffer</td>
<td>ARRAY [0..MAX_CHANNEL] OF BYTE</td>
<td>DMX channel values</td>
</tr>
</tbody>
</table>
4.1 Status (GVL)

<table>
<thead>
<tr>
<th>Value</th>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eStatus.Ok</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.none</td>
<td>‘OK’</td>
</tr>
<tr>
<td>eStatus.Busy</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.info</td>
<td>‘Busy’</td>
</tr>
<tr>
<td>eStatus.Timeout</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Timeout’</td>
</tr>
<tr>
<td>eStatus.FrameError</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Frame error’</td>
</tr>
<tr>
<td>eStatus.InvalidPortID</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Invalid port ID’</td>
</tr>
<tr>
<td>eStatus.InvalidParam</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Invalid parameter’</td>
</tr>
<tr>
<td>eStatus.InvalidSerialModule</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Invalid serial module’</td>
</tr>
<tr>
<td>eStatus.InvalidPort</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Invalid Port’</td>
</tr>
<tr>
<td>eStatus.InvalidFirmware</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Invalid Firmware’</td>
</tr>
<tr>
<td>eStatus_NullPointer</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Null pointer’</td>
</tr>
<tr>
<td>eStatus.InterfaceNotAvailable</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Interface not available’</td>
</tr>
<tr>
<td>eStatus.OverLimitNumberOfChannels</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘Over the limit of allowed number of send channels’</td>
</tr>
<tr>
<td>eStatus.NoModuleFound</td>
<td>WagoTypesError-Base.WagoTypes.eSeverity.error</td>
<td>‘No module found on the nodes’</td>
</tr>
</tbody>
</table>
4.2 eStatus (ENUM)

<table>
<thead>
<tr>
<th>Name</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ok</td>
<td>0</td>
</tr>
<tr>
<td>Busy</td>
<td>1</td>
</tr>
<tr>
<td>Timeout</td>
<td>2</td>
</tr>
<tr>
<td>FrameError</td>
<td>3</td>
</tr>
<tr>
<td>InvalidPortID</td>
<td>4</td>
</tr>
<tr>
<td>InvalidParam</td>
<td>5</td>
</tr>
<tr>
<td>InvalidSerialModule</td>
<td>6</td>
</tr>
<tr>
<td>InvalidPort</td>
<td>7</td>
</tr>
<tr>
<td>InvalidFirmware</td>
<td>8</td>
</tr>
<tr>
<td>NullPointer</td>
<td>9</td>
</tr>
<tr>
<td>InterfaceNotAvailable</td>
<td>10</td>
</tr>
<tr>
<td>OverLimitNumberOfChannel</td>
<td>11</td>
</tr>
<tr>
<td>NoModuleFound</td>
<td>12</td>
</tr>
</tbody>
</table>
### GlobalVariables (GVL)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_DMX_DATA</td>
<td>ARRAY [1..MAX_MODULE] OF typDMX</td>
</tr>
</tbody>
</table>
## ParameterList (PARAMS)

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>MAX_MODULE</td>
<td>DWORD</td>
<td>100</td>
<td>maximum number of master</td>
</tr>
<tr>
<td></td>
<td>RING_BUFFER_SIZE</td>
<td>INT</td>
<td>255</td>
<td>Default-Size of the RingBuffer</td>
</tr>
<tr>
<td></td>
<td>MAX_CHANNEL</td>
<td>INT</td>
<td>512</td>
<td>maximum DMX channel</td>
</tr>
<tr>
<td></td>
<td>MAX_DMX_SLAVE_RX</td>
<td>INT</td>
<td>21</td>
<td>maximum number of received slaved data</td>
</tr>
</tbody>
</table>
CHAPTER 7

VersionHistory (GVL)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>WagoAppDMX.library</td>
<td>ProjectInfo</td>
</tr>
</tbody>
</table>

Description: Library for DMX communication with WAGO I/O- System

<table>
<thead>
<tr>
<th>date</th>
<th>version</th>
<th>author</th>
<th>change</th>
</tr>
</thead>
<tbody>
<tr>
<td>08.01.2019</td>
<td>1.0.1.0</td>
<td>u015842</td>
<td>Properties: free placeholder added</td>
</tr>
<tr>
<td>11.10.2018</td>
<td>1.0.0.15</td>
<td>u015842</td>
<td>New Property TxTrigger</td>
</tr>
<tr>
<td>26.04.2018</td>
<td>1.0.0.14</td>
<td>u015842</td>
<td>Update Documentation</td>
</tr>
<tr>
<td>28.11.2017</td>
<td>1.0.0.13</td>
<td>u015842</td>
<td>Documentation note for FbDmxSlave added</td>
</tr>
<tr>
<td>05.05.2017</td>
<td>1.0.0.11</td>
<td>u015842</td>
<td>Internal interface changed</td>
</tr>
<tr>
<td>28.03.2017</td>
<td>1.0.0.10</td>
<td>u015842</td>
<td>R4 Patch 1</td>
</tr>
<tr>
<td>14.11.2016</td>
<td>1.0.0.8</td>
<td>u014521</td>
<td>R4</td>
</tr>
</tbody>
</table>
Library Reference

This is a dictionary of all referenced libraries and their name spaces.

**Standard**

*Library Identification:*
- Placeholder: Standard
- Default Resolution: Standard, 3.5.5.0 (System)
- Namespace: Standard

*Library Properties:*
- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

**WagoSysErrorBase**

*Library Identification:*
- Placeholder: WagoSysErrorBase
- Default Resolution: WagoSysErrorBase, * (WAGO)
- Namespace: WagoSysErrorBase

*Library Properties:*
- LinkAllContent: False
- Optional: False
- QualifiedOnly: False
- SystemLibrary: False
- PublishSymbolsInContainer: True

**WagoSysSerial**

*Library Identification:*
- Placeholder: WagoSysSerial
- Default Resolution: WagoSysSerial, * (WAGO)
- Namespace: WagoSysSerial
Library Properties:

- LinkAllContent: False
- QualifiedOnly: True
- SystemLibrary: False
- Optional: False

WagoSysVersion

Library Identification:
Name: WagoSysVersion
Version: 1.0.0.0
Company: WAGO
Namespace: WagoSysVersion

Library Properties:

- LinkAllContent: False
- QualifiedOnly: True
- SystemLibrary: False
- Optional: False

WagoTypesCom

Library Identification:
Placeholder: WagoTypesCom
Default Resolution: WagoTypesCom, * (WAGO)
Namespace: WagoTypesCom

Library Properties:

- LinkAllContent: False
- Optional: False
- QualifiedOnly: True
- SystemLibrary: False
- PublishSymbolsInContainer: True

WagoTypesErrorBase

Library Identification:
Placeholder: WagoTypesErrorBase
Default Resolution: WagoTypesErrorBase, * (WAGO)
Namespace: WagoTypesErrorBase

Library Properties:

- LinkAllContent: False
- Optional: False
- QualifiedOnly: True
- SystemLibrary: False
- PublishSymbolsInContainer: True
WagoTypesModuleBase

Library Identification:
Placeholder: WagoTypesModuleBase
Default Resolution: WagoTypesModuleBase, * (WAGO)
Namespace: WagoTypesModuleBase

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

© WAGO Kontakttechnik GmbH & Co. KG, Germany 2018 – All rights reserved. For the avoidance of doubt, this copyright notice does not only apply to the information above but also and primarily to the described library itself. Please note that third-party products are always mentioned without reference to intellectual property rights, including patents, utility models, designs and trademarks, accordingly the existence of such rights cannot be excluded. WAGO is a registered trademark of WAGO Verwaltungsgesellschaft mbH.