WAGO I/O SYSTEM 750

Libraries for Building Automation

Function Block Description for the Library DMX_01.lib

Last change15.07.2015

WAGO INNOVATIVE CONNECTIONS

WAGO Kontakttechnik GmbH & Co. KG
Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-mail: info@wago.com

Web: http://www.wago.com

Technical Support
Phone: +49 (0) 571/8 87 – 555
Fax: +49 (0) 571/8 87 – 8555

E-mail: support@wago.com

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

We wish to point out that the software and hardware names, as well as the trademarks of companies used and/or mentioned in the present manual, are generally protected by trademark or patent.

WAGO-I/O-*PRO* V2.3 Library for the DMX Protocol

Content

Content 3

Important Notes 4

Copyright 4
Personnel Qualification 4
Intended Use 4
Scope of Validity 5

Communication 6

DMX Master Block (FbDMX_652_Master) 6

Light Effects 9

RGB Color Mixer (FbRGB_ColourMixer) 9
Save RGB Color Palette (FbRGB_SaveColourPalette) 11
Calling Up an RGB Color Palette (FbRGB_RecallColourPalette) 14
Fade Sequence (FbDMX_FadeSequence) 16
Cross Fade Sequence (FbRGB_CrossFadeSequence) 18
Chaser (FbDMX_Chaser) 20
Change Channel Values (FbDMX_ChangeChValues) 22

Important Notes

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.

Copyright

Personnel Qualification

The use of the product described in this document is exclusively geared to specialists having qualifications in SPS 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 Validity

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.

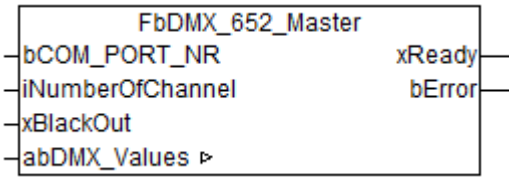
Communication

DMX Master Block (FbDMX_652_Master)

WAGO-I/O-*PRO V2.3* Library Elements		
Category:	Building Automation	
Name:	FbDMX_652_Master	
Type:	Function ☐	Funktion block ☒ Program ☐
Name of library:	DMX_01.lib	
Applicable to:	See Release Note	
Libraries used:	SerComm.lib Serial_Interface_01.lib.	
Input parameter:	**Data type:**	**Comment:**
xEnable	BOOL	Enables the function block. Default setting = TRUE
bCOM_PORT_NR	BYTE	No. of the serial interface used Default setting= 2 1 -> Internal service port 2 -> 1. connected serial module 3 -> 2. connected serial module
iNumberOfChannel	INT	Number of channels to be transmitted Value range = 1 - 254 Default setting: 45
xBlackOut	BOOL	TRUE-> Shutdown mode active
Input/output parameters:	**Data type:**	**Comment:**
abDMX_Values	ARRAY [1..DMX_MAX_CH] of BYTE	Array of DMX values. DMX_MAX_CH=512
Output parameter:	**Data type:**	**Comment:**
xReady	BOOL	Communication status TRUE = No transmission process FALSE = Transmission process is activated

bError	BYTE		Error from Sercom.lib:
		0x00	No error
		0x01	This library is not supported by the firmware.
		0x02	COM port outside of valid range.
		0x03	This function block entity is not yet assigned to a COM port.
		0x04	This function block entity is assigned to a different COM port.
		0x05	COM port already open.
		0x06	COM port already closed.
		0x07	COM port not opened.
		0x08	A write operation is still active (COM1).
		0x09	These transfer parameters are not supported by the COM port.
		0x0A	Current I/O module settings could not be read.
		0x0B	This library version does not support temporary setting of communication parameters.
		0x0C	I/O module could not be initiatlized.
		0x0D	Error during writing of data to the I/O module FIFO memory.
		0x0E	Contents of FIFO memory were not sent (continuous sending).
		0x0F	Internal error

Graphical illustration:



Function description:

The **FbDMX_652_Master** function block transmits values to a DMX line. Communication takes place via a 750-652 RS-485 interface module. This function block may be used only once per installed serial module.

The ***"abDMX_Values"*** array contains the DMX values to be transmitted. An array index is available for each DMX channel. The DMX values are transmitted in cycles as soon as the ***"xEnable"*** is set to TRUE.

The maximum number of channels to be transmitted can be limited at the ***"iNumberOfChannel"*** input.

The ***"xBlackOut"*** input activates the Blackout mode.

- *"xBlackOut"* = TRUE -> Blackout mode is activated. The values for all of the DMX channels remain at zero.
- *"xBlackOut"* = FALSE -> Blackout mode is not activated. The DMX values that have been enetered become effective.

The fieldbus controller detects and assigns the port numbers of the connected serial I/O modules independently from the left beginning with COM2. The service interface on the controller is always COM1. To address the function block to the proper serial module, the corresponding number (e.g. "2" for COM2) must be entered as a constant at the "***bCOM_PORT_NR"*** input.

The ***"xReady"*** output signals whether the module is active. As long as *"xReady"* is FALSE, no further action is taken by the function block.

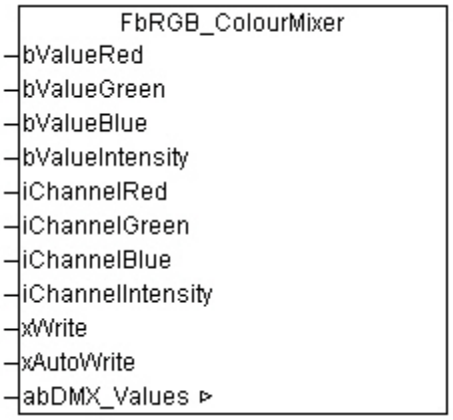
In the event of a communication error, a corresponding error code is output on the ***"bError"*** output.

Light Effects

RGB Color Mixer (FbRGB_ColourMixer)

WAGO-I/O-*PRO V2.3* Library Elements		
Category:	Building Automation	
Name:	FbRGB_ColourMixer	
Type:	Function ☐	Funktion block ☒ Program ☐
Name of library:	DMX_01.lib	
Applicable to:	See Release Note	
Library used:	SerComm.lib Serial_Interface_01.lib.	
Input parameter:	**Data type:**	**Comment:**
bValueRed	BYTE	DMX value, channel "red"
bValueGreen	BYTE	DMX value, channel "green"
bValueBlue	BYTE	DMX value, channel "blue"
bValueIntensity	BYTE	DMX value, channel "intensity"
iChannelRed	INT	Address for "red" channel
iChannelGreen	INT	Address for "green" channel
iChannelBlue	INT	Address for "blue" channel
iChannelIntensity	INT	Address for "intensity" channel
xWrite	BOOL	A rising edge writes the entered values to the corresponding channels.
xAutoWrite	BOOL	DMX values are refreshed automatically.
Input/output parameters:	**Data type:**	**Comment:**
abDMX_Values	ARRAY [1..DMX_MAX_CH] of BYTE	Array of DMX values. DMX_MAX_CH=512
Output parameter:	**Data type:**	**Comment:**
-	-	-

Graphical illustration:



Function description:

The **FbRGB_ColourMixer** function block is used for setting the color of an RGB light.

The individual color components are assigned at the ***"bValueRed"***, ***"bValueGreen"***, ***"bValueBlue"*** and ***"bValueIntensity"*** inputs.

The addresses for the corresponding DMX channels are assigned at the ***"iChannelRed"***, ***"iChannelGreen", "iChannelBlue"*** and ***"iChannelIntensity"*** inputs.

The values are transmitted to the DMX line by a rising edge at the ***"xWrite"*** input.

If the input variable ***"xAutoWrite"*** is set to TRUE, the inputs *"bValueRed"*, *"bValueGreen"*, *"bValueBlue"* and *"bValueIntensity"* are monitored for value shifting. As soon as a value changes this is transmitted to the DMX line.

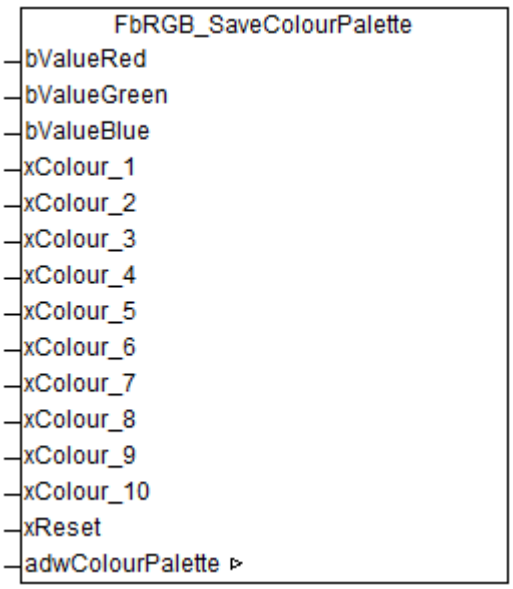
This function block is used together with the communication module (see Page 6). Synchronization of the two entities is achieved via the ***"abDMX_Values"*** array. Therefore, the communication module and function block must be linked to each other. The function block can write values to the DMX line via this link.

Save RGB Color Palette (FbRGB_SaveColourPalette)

WAGO-I/O-*PRO V2.3* Library Elements			
Category:	Building Automation		
Name:	FbRGB_SaveColourPalette		
Type:	Function ☐	Funktion block ☒	Program ☐
Name of library:	DMX_01.lib		
Applicable to:	See Release Note		
Library used:	SerComm.lib Serial_Interface_01.lib.		
Input parameter:	**Data type:**	**Comment:**	
bValueRed	BYTE	DMX value, channel "red"	
bValueGreen	BYTE	DMX value, channel "green"	
bValueBlue	BYTE	DMX value, channel "blue"	
xColour_1	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[1]	
xColour_2	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[2]	
xColour_3	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[3]	
xColour_4	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[4]	
xColour_5	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[5]	
xColour_6	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[6]	
xColour_7	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[7]	
xColour_8	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[8]	
xColour_9	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[9]	
xColour_10	BOOL	A rising edge will result in the color palette being saved in adwColourPalette[10]	
xReset	BOOL	A rising edge will result in the entire contents of the adwColourPalette array being cleared.	
Input/output parameters:	**Data type:**	**Comment:**	

WAGO-I/O-*PRO V2.3* Library Elements		
adwColourPalette	ARRAY [1..10] of DWORD	Color palette array.
Output parameter:	**Data type:**	**Comment:**
-	-	-

Graphical illustration:



Function description:

Ten (10) color palettes can be stored using the **FbRGB_SaveColourPalette** function block.

The individual color palettes can be configured via the ***"bValueRed"***, ***"bValueGreen"*** and ***"bValueBlue"*** inputs.

At a rising edge at the ***"xColour_1"*** to ***"xColour_10"*** inputs, the color palette is saved in the corresponding element of the *"adwColourPalette"* array.

The color palettes are saved in the ***"adwColourPalette"*** array. Representation is as a hexadecimal character in the order B (Blue) G (Green) R (Red). Yellow, for example, in this form has the value 16#00FFFF and white the value 16#FFFFFF.

Using a rising edge at the ***"xReset"*** input, the contents of the *"adwColourPalette"* array can be deleted.

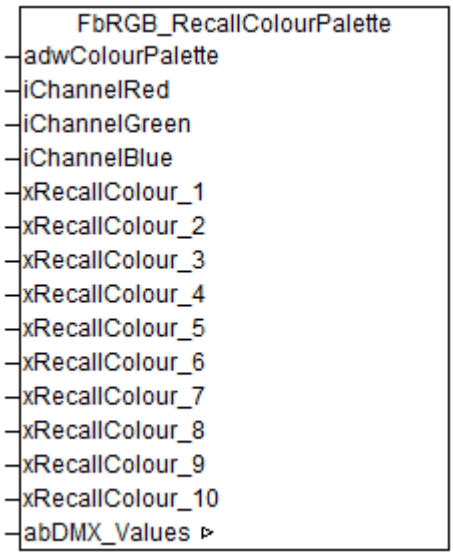
Note
The variables at the *"adwColourPalette"* input should be declared as RETAIN PERSISTENT so that the list of color palettes is retained after a controller reset and after a download.

Calling Up an RGB Color Palette (FbRGB_RecallColourPalette)

WAGO-I/O-*PRO V2.3* Library Elements			
Category:	Building Automation		
Name:	FbRGB_RecallColourPalette		
Type:	Function ☐	Funktion block ☒	Program ☐
Name of library:	DMX_01.lib		
Applicable to:	See Release Note		
Library used:	SerComm.lib Serial_Interface_01.lib.		
Input parameter:	**Data type:**	**Comment:**	
adwColourPalette	ARRAY [1..10] of DWORD	Color palette array.	
iChannelRed	INT	Address for "red" channel	
iChannelGreen	INT	Address for "green" channel	
iChannelBlue	INT	Address for "blue" channel	
xRecallColour_1	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[1] auf.	
xRecallColour_2	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[2] auf.	
xRecallColour_3	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[3] auf.	
xRecallColour_4	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[4] auf.	
xRecallColour_5	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[5] auf.	
xRecallColour_6	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[6] auf.	
xRecallColour_7	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[7] auf.	
xRecallColour_8	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[8] auf.	
xRecallColour_9	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[9] auf.	
xRecallColour_10	BOOL	A rising edge will result in the color palette being called up from adwColourPalette[10] auf.	

Input/output parameters:	Data type:	Comment:
abDMX_Values	ARRAY [1..DMX_MAX_CH] of BYTE	Array of DMX values. DMX_MAX_CH=512
Output parameter:	**Data type:**	**Comment:**
-	-	-

Graphical illustration:



Function description:

Using the **FbRGB_RecallColourPalette** function block, stored color palettes can be called from the ***"adwColourPalette"*** array.

The *"adwColourPalette"* input can be linked with the variables of the same name of the **FbRGB_SaveColourPalette** function block and contains the stored color palettes. Representation is as a hexadecimal character in the order B (Blue) G (Green) R (Red). Yellow, for example, in this form has the value 16#00FFFF and white the value 16#FFFFFF.

The addresses for the corresponding DMX channels are assigned at the ***"iChannelRed"***, ***"iChannelGreen"*** and ***"iChannelBlue"*** inputs.

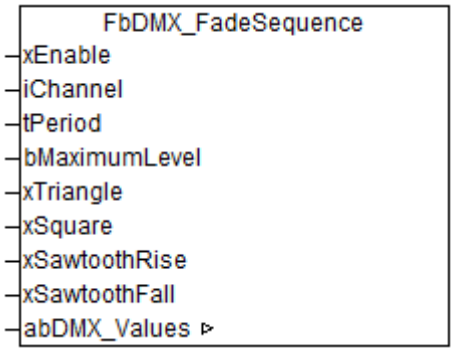
At a rising edge at the ***"xRecallColour_1"*** to ***"xRecallColour_10"*** inputs, the color palettes will be called up from the corresponding element of the *"adwColourPalette"* array.

This function block is used together with the communication module (see Page 6). Synchronization of the two entities is achieved via the ***"abDMX_Values"*** array. Therefore, the communication module and function block must be linked to each other. The function block can write values to the DMX line via this link.

Fade Sequence (FbDMX_FadeSequence)

WAGO-I/O-*PRO V2.3* Library Elements		
Category:	Building Automation	
Name:	FbDMX_FadeSequence	
Type:	Function ☐	Funktion block ☒ Program ☐
Name of library:	DMX_01.lib	
Applicable to:	See Release Note	
Library used:	SerComm.lib Serial_Interface_01.lib.	
Input parameter:	**Data type:**	**Comment:**
xEnable	BOOL	Activation of the fade sequence
iChannel	INT	Address of the DMX channel
tPeriod	TIME	Cycle duration Minimum: 1s Default setting = 5 s
bMaximumLevel	BYTE	Maximum DMX channel value Default setting: 255
xTriangle	BOOL	Triangle function Default setting: TRUE
xSquare	BOOL	Pulsating signal
xSawtoothRise	BOOL	Rising sawtooth
xSawtoothFall	BOOL	Falling sawtooth
Input/output parameters:	**Data type:**	**Comment:**
abDMX_Values	ARRAY [1..DMX_MAX_CH] of BYTE	Array of DMX values. DMX_MAX_CH=512
Output parameter:	**Data type:**	**Comment:**
-	-	-

Graphical illustration:



Function description:

A fade sequence can be generated using the **FbDMX_FadeSequence** function block. The function block is activated via the ***"xEnable"*** variable.

The address of the DMX channel is assigned at the ***"iChannel"*** input.

The ***"tPeriod"*** defines the duration of the fade sequence.

The ***"bMaximumValue"*** defines the maximum achievable value for the fade sequence.

If one of the following variables is set to TRUE, the corresponding function is generated:

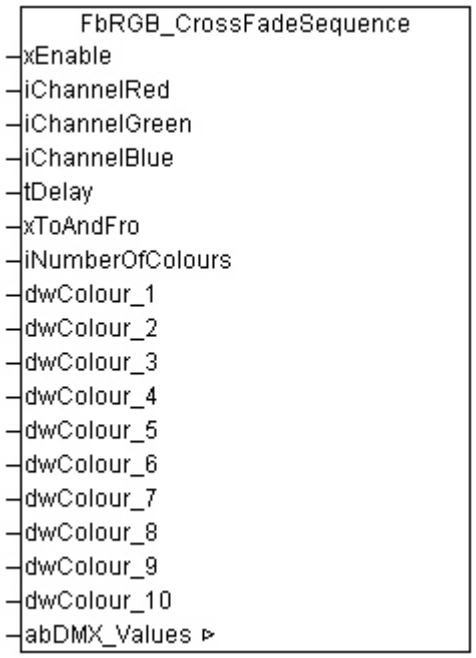
1. ***"xTriangle"***- triangle function.
2. ***"xSquare"***- pulsating signal.
3. ***"xSawtoothRise"***- Rising sawtooth.
4. ***"xSawtoothFall"***- Falling sawtooth.

This function block is used together with the communication module (see Page 6). Synchronization of the two entities is achieved via the ***"abDMX_Values"*** array. Therefore, the communication module and function block must be linked to each other. The function block can write values to the DMX line via this link.

Cross Fade Sequence (FbRGB_CrossFadeSequence)

WAGO-I/O-*PRO V2.3* Library Elements			
Category:	Building Automation		
Name:	FbRGB_CrossFadeSequence		
Type:	Function ☐	Funktion block ☒	Program ☐
Name of library:	DMX_01.lib		
Applicable to:	See Release Note		
Library used:	SerComm.lib Serial_Interface_01.lib.		
Input parameter:	**Data type:**	**Comment:**	
xEnable	BOOL	Activation of the fade sequence	
iChannelRed	INT	Address for "red" channel	
iChannelGreen	INT	Address for "green" channel	
iChannelBlue	INT	Address for "blue" channel	
tDelay	TIME	Delay time Minimum: 1s Default setting = 5 s	
xToAndFro	BOOL	Rising/Falling fade sequence	
iNumberOfColours	INT	Number of fade sequence colors Value range = 1 – 10 Default setting: 10	
dwColour_1	DWORD	1. Color	
dwColour_2	DWORD	2. Color	
dwColour_3	DWORD	3. Color	
dwColour_4	DWORD	4. Color	
dwColour_5	DWORD	5. Color	
dwColour_6	DWORD	6. Color	
dwColour_7	DWORD	7. Color	
dwColour_8	DWORD	8. Color	
dwColour_9	DWORD	9. Color	
dwColour_10	DWORD	10. Color	
Input/output parameters:	**Data type:**	**Comment:**	
abDMX_Values	ARRAY [1..DMX_MAX_CH] of BYTE	Array of DMX values. DMX_MAX_CH=512	
Output parameter:	**Data type:**	**Comment:**	
-	-	-	

Graphical illustration:



Function description:

A cross fade sequence can be generated using the **FbRGB_CrossFadeSequence** function block. The sequence is activated via the ***"xEnable"*** input.

The addresses for the corresponding DMX channels are assigned at the ***"iChannelRed"***, ***"iChannelGreen"*** and ***"iChannelBlue"*** inputs.

Cross fading between the sequences is defined by the delay time ***"tDelay"***.

The fade sequence colors can be configured via the ***"dwColour_1"*** to ***"dwColour_10"*** inputs. Values are entered as a hexadecimal character in the order B (Blue) G (Green) R (Red). Yellow, for example, in this form has the value 16#00FFFF and white the value 16#FFFFFF.

The number of fade sequence colors is defined at the ***"iNumberOfColours"*** input.

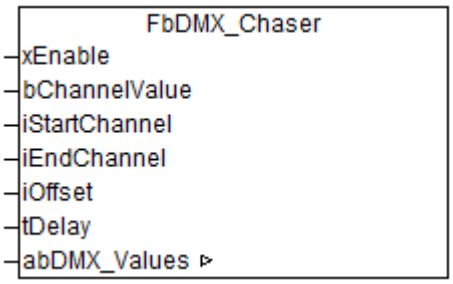
A TRUE signal at the ***"xToAndFro"*** activates a cross fade sequence that runs continuously back and forth. A FALSE must be configured at the input if the fade sequence is to start over from the beginning when a maximum number of fade sequence colors is reached.

This function block is used together with the communication module (see Page 6). Synchronization of the two entities is achieved via the ***"abDMX_Values"*** array. Therefore, the communication module and function block must be linked to each other. The function block can write values to the DMX line via this link.

Chaser (FbDMX_Chaser)

WAGO-I/O-*PRO V2.3* Library Elements			
Category:	Building Automation		
Name:	FbDMX_Chaser		
Type:	Function ☐	Funktion block ☒	Program ☐
Name of library:	DMX_01.lib		
Applicable to:	See Release Note		
Library used:	SerComm.lib Serial_Interface_01.lib.		
Input parameter:	**Data type:**	**Comment:**	
xEnable	BOOL	Enables the function block.	
bChannelValue	BYTE	DMX channel value	
iStartChannel	INT	Starting channel Minimum: 1 Default setting: 1	
iEndChannel	INT	End channel Minimum: 2 Default setting: 2	
iOffset	INT	Increment Minimum: 1 Default setting: 1	
tDelay	TIME	Delay time Minimum: 50ms Default setting = 1 s	
Input/output parameters:	**Data type:**	**Comment:**	
abDMX_Values	ARRAY [1..DMX_MAX_CH] of BYTE	Array of DMX values. DMX_MAX_CH=512	
Output parameter:	**Data type:**	**Comment:**	
-	**-**	**-**	

Graphical illustration:



Function description:

The **FbDMX_Chaser** function block copies the DMX value for a channel (A) to a different DMX channel (B) for a given number of DMX channels. The value for channel (A) is then reset to zero. A chaser effect can be created using this function.

The function block is activated via the ***"xEnable"*** variable. A falling edge at the "xEnable" input results in all DMX channels affected by this function block being reset to zero

The DMX value to be copied is configured at the ***"bChannelValue"*** input.

The DMX channel for which the copying process is to be started is assigned at the ***"iStartChannel"*** input. Copying of the DMX value is ended at the DMX channel ***"iEndChannel"***. The variable ***"iOffset"*** defines the increment for copying to a different DMX channel.

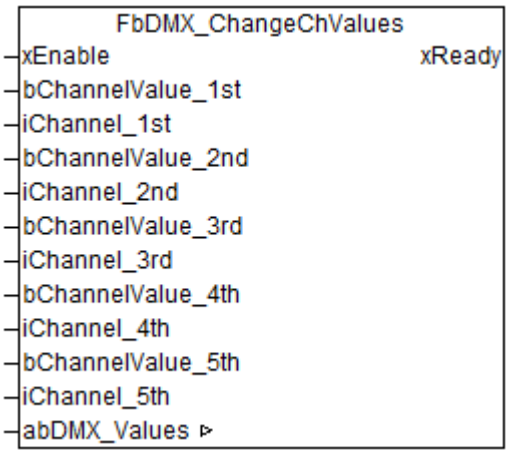
The delay period ***"tDelay"*** indicates the delay, or waiting period, between each step.

This function block is used together with the communication module (see Page 6). Synchronization of the two entities is achieved via the ***"abDMX_Values"*** array. Therefore, the communication module and function block must be linked to each other. The function block can write values to the DMX line via this link.

Change Channel Values (FbDMX_ChangeChValues)

WAGO-I/O-*PRO V2.3* Library Elements			
Category:	Building Automation		
Name:	FbDMX_ChangeChValues		
Type:	Function ☐	Funktion block ☒	Program ☐
Name of library:	DMX_01.lib		
Applicable to:	See Release Note		
Library used:	SerComm.lib Serial_Interface_01.lib.		
Input parameter:	**Data type:**	**Comment:**	
xEnable	BOOL	Enables the function block.	
bChannelValue_1st	BYTE	DMX value 1	
iChannel_1st	INT	Address of the DMX channel	
bChannelValue_2nd	BYTE	DMX value 2	
iChannel_2nd	INT	Address of the DMX channel	
bChannelValue_3rd	BYTE	DMX value 3	
iChannel_3rd	INT	Address of the DMX channel	
bChannelValue_4th	BYTE	DMX value 4	
iChannel_4th	INT	Address of the DMX channel	
bChannelValue_5th	BYTE	DMX value 5	
iChannel_5th	INT	Address of the DMX channel	
Input/output parameters:	**Data type:**	**Comment:**	
abDMX_Values	ARRAY [1..DMX_MAX_CH] of BYTE	Array of DMX values DMX_MAX_CH=512	
Output parameter:	**Data type:**	**Comment:**	
xReady	BOOL	Communication status TRUE = Function block active. FALSE = Function block busy	

Graphical illustration:



Function description:

The DMX values for up to five DMX channels can be set using the **FbDMX_ChangeChValue** function block. The function block is activated via the ***"xEnable"*** variable.

The DMX values which are to be transferred for a change in values are specified via the ***"bChannelValue_1st"*** to ***"bChannelValue_5th"*** inputs.

The addresses for the five DMX channels are assigned at the ***"bChannelValue_1st"*** to ***"bChannelValue_5th"*** inputs.

This function block is used together with the communication module (see Page 6). Synchronization of the two entities is achieved via the ***"abDMX_Values"*** array. Therefore, the communication module and function block must be linked to each other. The function block can write values to the DMX line via this link.

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 · D-32385 Minden
Hansastraße 27D-32423 Minden
Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69
E-mail: info@wago.com

Internet: http://www.wago.com