IUT ANNECY UNIVERSITÉ SAVOIE MONT BLANC

Bus de terrain

TP 2

v1

IUT d'Annecy, 9 rue de l'Arc en Ciel, 74940 Annecy

Mettre en oeuvre le protocole DMX

Dans les bâtiments tertiaires et secondaires, les techniques modernes de commande d'éclairage sont réalisées via le réseau non propriétaire DALI. Néanmoins, la dynamique de ce réseau reste trop lente par rapport aux exigences des effets de lumière que l'on souhaite faire sur les scènes de spectacle. Ainsi, dans le cadre de la commande de l'appareillage scénique (projecteurs RVB, lyres, stroboscopes, machines à fumée, gobo, ...), les standards utilisés pour piloter ces équipements via des réseaux sont : pour le plus répandu DMX512 (cf. United States Institute for Theatre Technology : DMX512-A - Asynchronous Serial Digital Data Transmission), pour le plus récent Art-Net (cf. http ://art-net.org.uk) ou bien encore les standards MIDI, Cobranet, ...

Objectif

Mettre en oeuvre le standard **DMX 512**, premier réseau à s'être imposé dans le monde du spectacle, à l'aide d'une structure de commande basée sur un automate programmable industriel.

1 Le réseau DMX

DMX est l'acronyme de Digital MultipleX qui est la définition d'une interface numérique standardisée non propriétaire pour les équipements scéniques. Le terme 512 fait référence au nombre de canaux disponibles sur un même réseau (appelé univers). La norme prévoit d'utiliser au maximum 32 systèmes de 16 canaux chacun. Un équipement DMX512 peut donc utiliser plusieurs canaux (par exemple, les ballasts RGBW que l'on utilisera nécessitent quatre canaux consécutifs).

Comme le réseau DMX512 est unidirectionnel (du maitre vers les équipements uniquement), on ne dispose pas sur ce réseau de compte rendu sur les échanges et sur la communication. Par conséquent, ce standard ne doit pas être utilisé pour des spectacles ou des équipements réclamant de la sécurité (spectacles pyrotechniques, équipements assurant le gréement de décors, leur levage, ...).

1.1 Caractéristiques techniques

- Tout équipement propose une entrée DMX IN(+, -), et une sortie DMX DMX IN DMX OUT OUT (+, -) (la sortie DMX OUT du dernier équipement sur la ligne doit être connectée à une résistance d'impédance 120 ohm)
- Topologie de type Bus avec un câblage série de type « Daisy chain »
- Bus Maitre-Esclaves unidirectionnel Terminateur
- Transmission série : 8 bits, 2 bits de stop, sans parité (RS 485) de ligne
- Vitesse de transmission : 250000 bit/s R=120 Ohm
- Longueur des câbles : 300 mètres entre équipements
- Le pilotage simultané de plusieurs appareils s'effectue en leur attribuant les mêmes canaux

Préparation 1: Le protocole RS 485

Question 1 En cherchant sur internet, expliquer le fonctionnement du protocole RS 485.

- Quelles tensions sont utilisées ?
- Quelle est la vitesse de transmission ?
- Quelle est la longueur maximale du bus ?
- Quelle est la différence avec le protocole RS 232 ?
- Expliquer la notion de bit de parité et de bit de stop.

Question 2 Comparer ce bus avec le Bus DALI déjà étudié.

Question 3 Le protocole RS485 est-il compatible avec la mise en oeuvre d'une communication DMX ?

Une trame DMX512 contient dans sa partie utile 512 octets, qui par leur position dans la trame définit la valeur de commande des 512 canaux. La norme laisse le soin à chaque constructeur de choisir l'interprétation de ces valeurs (comprises entre 0 et 255).

Préparation 2: Vitesse du DMX

Question 4 Combien de trame par seconde peut-on envoyer via le réseau DMX ?

Dans ce TP, on se propose d'utiliser un coupleur **750-652** pour mettre en oeuvre un commande DMX. Il jouera le rôle de maître DMX.

On reliera deux ballasts RGBW (Red Green Blue White). Sur ces deux ballasts, on fixe le canal de référence à l'aide de boutons poussoirs *CH+* et *CH-*. Ce canal de référence correspond à la sortie **R**, puis les canaux suivants seront respectivement **G**, **B** puis **W**.

Dans notre cas, on donnera comme canaux de référence respectivement **1 puis 7**.

Préparation 3: Architecture matérielle

Question 5 Expliquer ce que fait le module **750-652**.

Les éléments DMX de notre maquette sont installés en *Daisy Chain*.

Question 6 Expliquer ce que signifie *Daisy Chain*.

Question 7 Dessiner l'architecture matérielle permettant de mettre en oeuvre deux bandeaux RGBW à l'aide d'un automate Wago et du protocole DMX.

Question 8 Préciser sur le schéma les numéros des canaux DMX.

1.2 Programmation

Préparation 4: DMX Master

Question 9 A l'aide de l'annexe page 9, expliquer le rôle du bloc *FbDMX_ Master*.

Question 10 De quel type et à quoi sert la variable **abDMX_ Values** ?

Question 11 Proposer une classe associée à ce bloc fonction.

Cahier des charges : Une couleur

On souhaite allumer la colonne de gauche (Canal 1) d'une couleur de votre choix.

Préparation 5: Une couleur

Question 12 Choisir une couleur mélangeant les trois couleurs primaires puis écrire son code HTML ainsi que les valeurs d'intensité des trois canaux au format *byte*.

Question 13 Écrire un programme, en langage CFC (Blocs fonctions), permettant d'allumer la colonne de gauche de la couleur choisie.

Cahier des charges : potentiomètre

La couleur du bandeau doit maintenant changer lorsque la valeur du potentiomètre *bValue* est supérieure à 100.

Préparation 6: Potentiomètre

Question 14 En ajoutant une partie de texte structuré, modifier le programme précédent pour répondre au cahier des charges.

Cahier des charges : Succession de couleurs

On souhaite effectuer une séquence de 10 couleurs différentes sur la colonne de droite (Canal 7). La séquence de couleur est définie dans le tableau suivant :

B	16#00	16#00	16#00	16#00	16#00	16#08	16#10	16#20	16#40	16#80
G	16#08	16#10	16#20	16#40	16#80	16#80	16#40	16#20	16#10	16#08
R	16#80	16#40	16#20	16#10	16#08	16#00	16#00	16#00	16#00	16#00

Préparation 7: Succession de couleurs

Pour ce cahier des charges, on propose d'utiliser le bloc fonction *FbRGB_ CrossFadeSequence* (page 14) qui permet de faire varier la couleur d'un canal de manière progressive.

Question 15 Proposer une classe associée à ce bloc fonction. On utilisera un tableau pour stocker les 10 couleurs.

Question 16 Écrire un programme, en langage ST, permettant d'allumer la colonne de gauche de la couleur choisie.

2 Partie pratique

2.1 Prise en main

Étape 1 Récupérer le projet DMX dans le dossier *Squelettes* du module, le copier dans votre dossier personnel puis l'ouvrir dans une machine virtuelle.

Cahier des charges : Une couleur de chaque côté

On souhaite allumer la colonne de gauche (Canal 1) d'une couleur de votre choix, la colonne de droite (canal 7) d'une autre couleur.

Activité 1: Premier programme en langage CFC

Dans cette activité, pour simplifier et appréhender les interactions entre les blocs, nous nous affranchissons temporairement de la notion de classe et utiliserons uniquement des blocs fonctions.

Étape 2 Mettre en oeuvre le cahier des charges ci-dessus en langage CFC à l'aide des blocs fonctions *FbDMX_ Master*, *FbRGB_ ColourMixer*.

Étape 3 Tester puis faire vérifier par l'enseignant.

2.2 Programmation en ST

Activité 2: Sucession de couleurs

Dans cette activité, nous nous proposons d'implémenter le cahier des charges *Sucession de couleurs* donné en page 6.

Étape 4 Désactiver ou supprimer le programme précédent.

Étape 5 Déclarer les classes *CDMXMaster*, et *CRGBCrossFadeSequence* dans l'onglet type de données.

Étape 6 Mettre en œuvre le cahier des charges ci-dessus en langage ST dans le module **DmxPrg** en instanciant des objets des classes créées. On pensera à appeler le bloc fonction de chaque classe.

Étape 7 Tester puis faire vérifier par l'enseignant.

Cahier des charges : Potentiomètre

La couleur du bandeau de gauche doit représenter la valeur de la tension aux bornes du potentiomètre. Cette valeur est disponible grâce à la variable *bValue* et est comprise entre 0 et 255.

Activité 3: Déclaration des classes

Étape 8 Déclarer une classe *CRGB_ ColourPalette* pour stocker les couleurs de la palette.

Étape 9 Lui ajouter une méthode de type *FbRGB_ RecallColourPalette* pour sélectionner une couleur de la palette.

Étape 10 Instancier un objet de cette classe, puis définir le tableau des couleurs (on utilisera un tableau identique à celui de l'activité précédente.)

Étape 11 Pour chaque entrée *xRecallColour_x* du bloc fonction, écrire une condition fonction de la valeur du potentiomètre pour sélectionner la couleur correspondante. L'ordre devra être le même que pour la séquence de couleur.

2.3 Pour aller plus loin

Cahier des charges : Modification de la séquence

Un fois une couleur affichée sur le bandeau de gauche, on souhaite pouvoir modifier cette couleur à l'aide des boutons poussoirs et du potentiomètre. On propose la démarche suivante :

1. Lorsque le potentiomètre est tourné, la couleur varie comme pour le cahier des charges précédent.
2. Si on souhaite modifier cette couleur, on actionnera le bouton blanc.
3. La couleur en question (case du tableau) est alors sélectionnée. La couleur ne varie plus directement avec le potentiomètre, mais peut être modifiée des boutons rouge, vert et bleu avec le potentiomètre.
   (a) Lorsque le bouton rouge est actionné, la couleur rouge varie avec le potentiomètre.
   (b) Lorsque le bouton vert est actionné, la couleur verte varie avec le potentiomètre.
   (c) Lorsque le bouton bleu est actionné, la couleur bleue varie avec le potentiomètre.
4. Lorsque le bouton blanc est actionné à nouveau, la couleur est validée et la séquence reprend.

Activité 4: Pour aller plus loin

Étape 12 Proposer une démarche pour répondre au cahier des charges ci-dessus. La faire valider par l'enseignant

Étape 13 Mettre en œuvre cette démarche.

A DMX Master

Communication

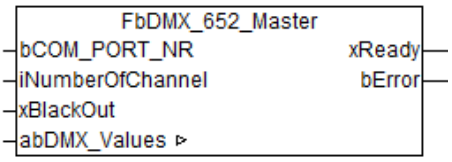
DMX Master Block (FbDMX_652_Master)

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

WAGO-I/O-*PRO V2.3* Library Elements		
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

WAGO Kontakttechnik GmbH & Co. KG
E-mail: info@wago.com
Web:http://www.wago.com
Postfach 2880 • D-32385 Minden
Hansastr. 27 • D-32423 Minden
Phone: +49 (0)5 71 / 8 87-0
Fax.: +49 (0)5 71 / 8 87-169

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.

B Colour Mixer

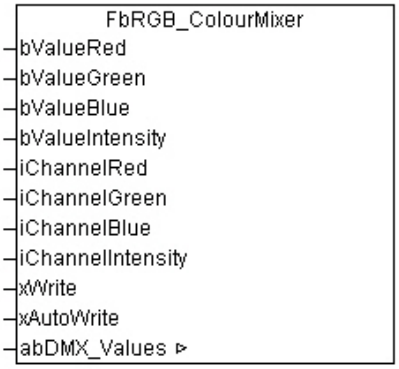
Light Effects

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:**
-	-	-

WAGO Kontakttechnik GmbH & Co. KG
E-mail: info@wago.com
Web:http://www.wago.com
Postfach 2880 • D-32385 Minden Phone: +49 (0)5 71 / 8 87-0
Hansastr. 27 • D-32423 Minden Fax.: +49 (0)5 71 / 8 87-169

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.

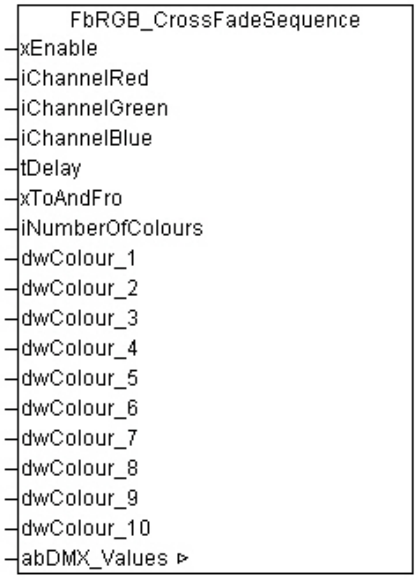
This function block is used together with the communication module (see Page 6). Synchronization of the two entities is achieved via the ***"abDMX_Values"*** array. Therefore, the communication module and function block must be linked to each other. The function block can write values to the DMX line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

C FbRGB CrossFadeSequence

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.