Description d'application

EDIZIOdue colore
Pussoir UNI 392x – connecteur à fiches

10.UNI3924-F.2104/210409

Tell
by Schneider Electric

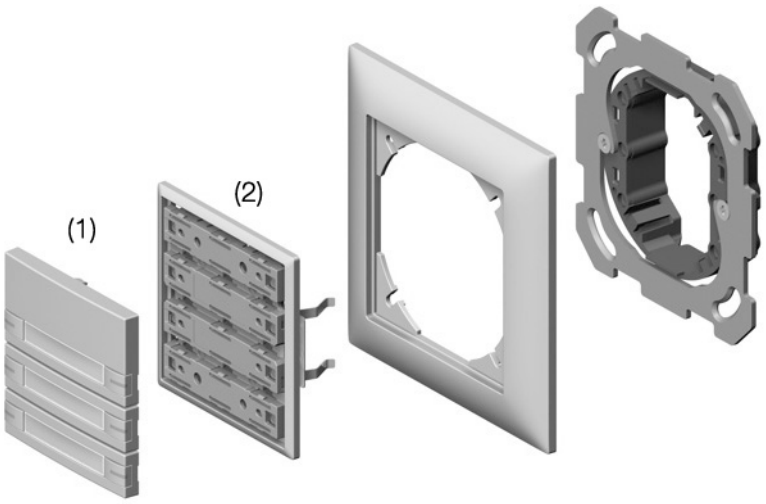
1 Introduction

Ce document sert de spécification technique pour le poussoir UNI EDIZIOdue colore 392x. Il présente l'interface et explique toutes les fonctions offertes par cet appareil.

Ce document s'applique aux produits avec les plaque frontale :
- 900-3924.FMI.... pour poussoir UNI 1-4x, touches 1/1 et 1/2, sans LED
- 900-3924.FMI.L.... pour poussoir UNI 1-4x, touches 1/1 et 1/2, avec LED
- 900-3928.FMI.... pour poussoir UNI 4-8x, touches 1/1, 1/2 et 1/4, sans LED
- 900-3928.FMI.L.... pour poussoir UNI 4-8x, touches 1/1, 1/2 et 1/4, avec LED

1.1 Conception du système

Le poussoir UNI (1) transmet des instructions par l'activation des touches et affiche des informations d'état par les LED.
Le poussoir est emboîté sur un coupleur de bus (2) via l'interface physique. Le coupleur de bus communique avec le poussoir via cette interface.
Le coupleur de bus se monte fondamentalement sur n'importe quel système ; il est monté ou acheté par le client.
La nature des instructions/fonctions du poussoir est déterminée par le coupleur de bus.



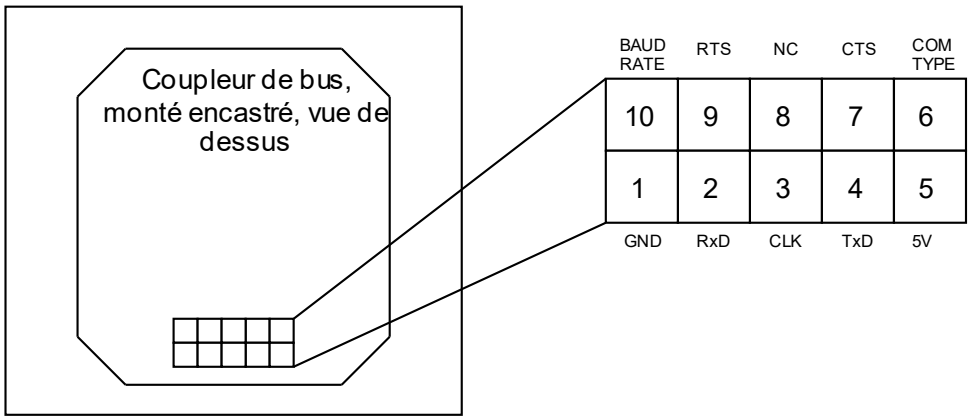
2 Interface physique

L'interface physique entre le coupleur de bus et le poussoir est assurée par une connexion enfichable 2 x 5 points au pas de 2,54 mm.

- Coupleur de bus : réglette à douilles
- Poussoir : réglette à broches

Il est recommandé d'utiliser une réglette à douilles dotée de contacts dorés pour le coupleur de bus. Il est ainsi possible d'éviter des problèmes de contact liés à la corrosion à long terme.

Le brochage de l'interface physique est défini de la manière suivante du point de vue du coupleur de bus (sens d'écoulement des données) :



Broche	Désignation	Description
1	GND	Masse
2	RxD	Caractéristiques de la ligne de réception (le coupleur de bus reçoit / le poussoir émet par cette ligne)
3	CLK	Rythme pour les lignes de données
4	TxD	Caractéristiques de la ligne d'émission (le coupleur de bus émet / le poussoir reçoit par cette ligne)
5	5V	Alimentation 5 V DC
6	COM TYPE	Mode Communication (est défini par la valeur de la tension sur cette broche, voir Mode)
7	CTS	Ligne de réception Handshake, contrôle de flux (le coupleur de bus reçoit / le poussoir émet par cette ligne)
8	NC	non raccordée (réservée pour des applications futures, laisser cette broche ouverte)
9	RTS	Ligne d'émission Handshake, contrôle de flux (le coupleur de bus émet / le poussoir reçoit par cette ligne)
10	BAUD RATE	Débit binaire (en bauds) (est défini par la valeur de la tension sur cette broche, voir Débit binaire)

2.1 Alimentation

Le poussoir doit être alimenté par le coupleur de bus Les critères de qualité de l'alimentation sont les suivants :

Désignation	mini	typique	maxi
Tension	4,75 V DC	5 V DC	5,25 V DC
Courant absorbé @ 0 LED	-	1 mA DC	-
Courant absorbé @ 4 LED rouge/vert, luminosité 50 %	-	3,4 mA DC	-
Courant absorbé @ 4 LED rouge/vert, luminosité 100 %	-	5,5 mA DC	-
Courant absorbé @ 8 LED rouge/vert, luminosité 50 %	-	5,7 mA DC	-
Courant absorbé @ 8 LED rouge/vert, luminosité 100 %	-	9,8 mA DC	-
Courant absorbé @ 4 LED bleu, luminosité 50 %	-	5,8 mA DC	-
Courant absorbé @ 4 LED bleu, luminosité 100 %	-	10,2 mA DC	-
Courant absorbé @ 8 LED bleu, luminosité 50 %	-	10,5 mA DC	-
Courant absorbé @ 8 LED bleu, luminosité 100 %	-	19,2 mA DC	-

Les niveaux des lignes de communications sont également de 5 V DC

3 Mode Communication

3.1 Préréglages

3.1.1 Mode Communication

Le mode Communication est déterminé par une résistance entre la broche 6 (COM TYPE) et la broche 5 (5V) de l'interface physique.
Le mode Communication est déterminé à chaque démarrage du programme (Hardware Power On Reset, Software Reset). Le mode Communication peut aussi être modifié par programme pendant l'exploitation.

Com Type N°	Description	Valeur de la résistance, tolérance
99	non défini, le mode Communication N° 11 est utilisé par défaut	Pas de résistance
12	réservé	82 kΩ, 1 %
11	**Protocole, sans handshake, asynchrone, par défaut**	**68 kΩ, 1 %**
10	réservé	56 kΩ, 1 %
09	Protocole, avec handshake, asynchrone	47 kΩ, 1 %
08	réservé	39 kΩ, 1 %
07	Protocole, avec handshake, synchrone	33 kΩ, 1 %
06	Echange d'octets, synchrone, couleur de LED : Rouge	27 kΩ, 1 %
05	Echange d'octets, synchrone, couleur de LED : Vert	22 kΩ, 1 %
04	Echange d'octets, synchrone, couleur de LED : Bleu	18 kΩ, 1 %
03	Echange d'octets, asynchrone, couleur de LED : Rouge	15 kΩ, 1 %
02	Echange d'octets, asynchrone, couleur de LED : Vert	12 kΩ, 1 %
01	Echange d'octets, asynchrone, couleur de LED : Bleu	10 kΩ, 1 %

3.1.2 Débit binaire (en bauds)

Le débit binaire (Baud Rate) est déterminé par une résistance entre la broche 10 (BAUD RATE) et la broche 5 (5V) de l'interface physique.
Le débit binaire est déterminé à chaque démarrage du programme (Hardware Power On Reset, Software Reset). Le débit binaire peut aussi être modifié par programme pendant l'exploitation. Le débit binaire n'est actif que pour les modes de communication asynchrones.

Débit binaire N°	Description	Valeur de la résistance, tolérance
99	non défini, le débit binaire N° 5 est utilisé par défaut	Pas de résistance
12	réservé	82 kΩ, 1 %
11	115 200 bauds	68 kΩ, 1 %
10	réservé	56 kΩ, 1 %
09	57 600 bauds	47 kΩ, 1 %
08	38 400 bauds	39 kΩ, 1 %
07	19 200 bauds	33 kΩ, 1 %
06	réservé	27 kΩ, 1 %
05	**9600 bauds, par défaut**	**22 kΩ, 1 %**
04	4800 bauds	18 kΩ, 1 %
03	2400 bauds	15 kΩ, 1 %
02	1200 bauds	12 kΩ, 1 %
01	réservé	10 kΩ, 1 %

3.2 Echange d'octets

Le mode Communication "Echange d'octets" se compose d'un protocole simple avec des paramètres de communication définis de manière fixe.
L'utilisation de ce protocole ne permet pas d'utiliser toutes les fonctions du poussoir.

Le coupleur de bus lance la communication en envoyant l'état des LED au moyen d'un octet. Le poussoir y répond en envoyant l'état des touches au moyen d'un octet.

L'octet de LED est codé comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
L8	L7	L6	L5	L4	L3	L2	L1

Désignation	Description
L8..L1	Etats des LED L8..L1. La couleur de la LED est sélectionnée par le valeur de la résistance en mode Communication. 1 = LED allumée 0 = LED éteinte

L'octet de touche est codé comme suit :

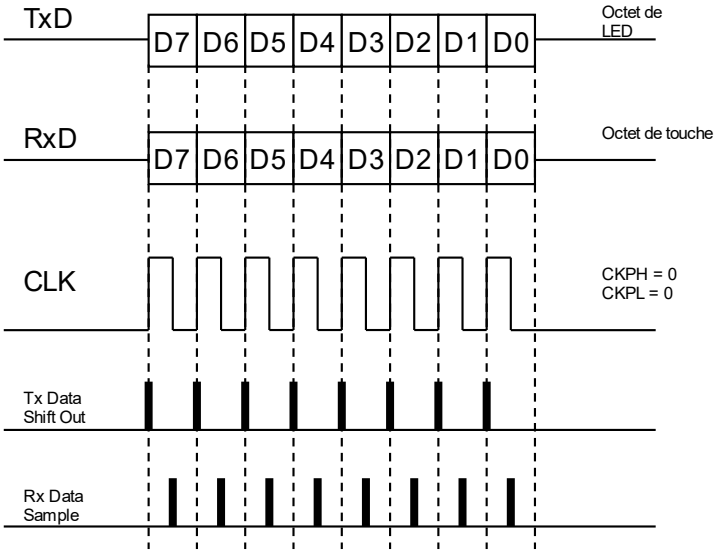
D7	D6	D5	D4	D3	D2	D1	D0
T8	T7	T6	T5	T4	T3	T2	T1

Désignation	Description
T8..T1	Etats des touches T8..T1. Les touches sont déjà sans rebondissement. 1 = Touche appuyée 0 = Touche relâchée

3.2.1 Synchrone

En mode sychrone (SPI), le coupleur de bus est supposé être le "Clock-Master" (horloge de référence). Cela veut dire que le coupleur de bus génère le rythme sur la ligne CLK et définit ainsi également le débit binaire de façon implicite. Le débit binaire maximal admissible est visible avec le service SetSystemSettings.request.

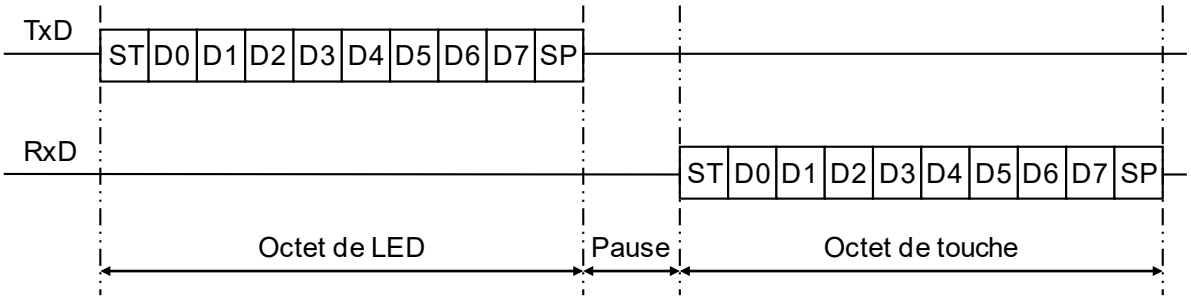
Chaque octet est envoyé et reçu selon la procédure suivante :



Désignation	Description
D7..D0	Bits de données
CKPH	Phase d'horloge normale (non retardée)
CKPL	Parité d'horloge Idle LOW

3.2.2 Asynchrone

Chaque octet est envoyé et reçu selon la procédure suivante :



Désignation	Description
ST	Bit de départ
D0..D7	Bits de données
SP	1 Bit d'arrêt
Parité	sans
Débit binaire	Selon résistance Broche 10/5, voir Interface physique
Pause	100 * (1 / Baud Rate) [s]

3.3 Protocole, avec handshake

Le mode Communication "Protocole avec handshake" se compose d'un protocole évolué avec des paramètres de communication réglables.
L'utilisation de ce protocole permet d'utiliser pleinement toutes les fonctions du poussoir.

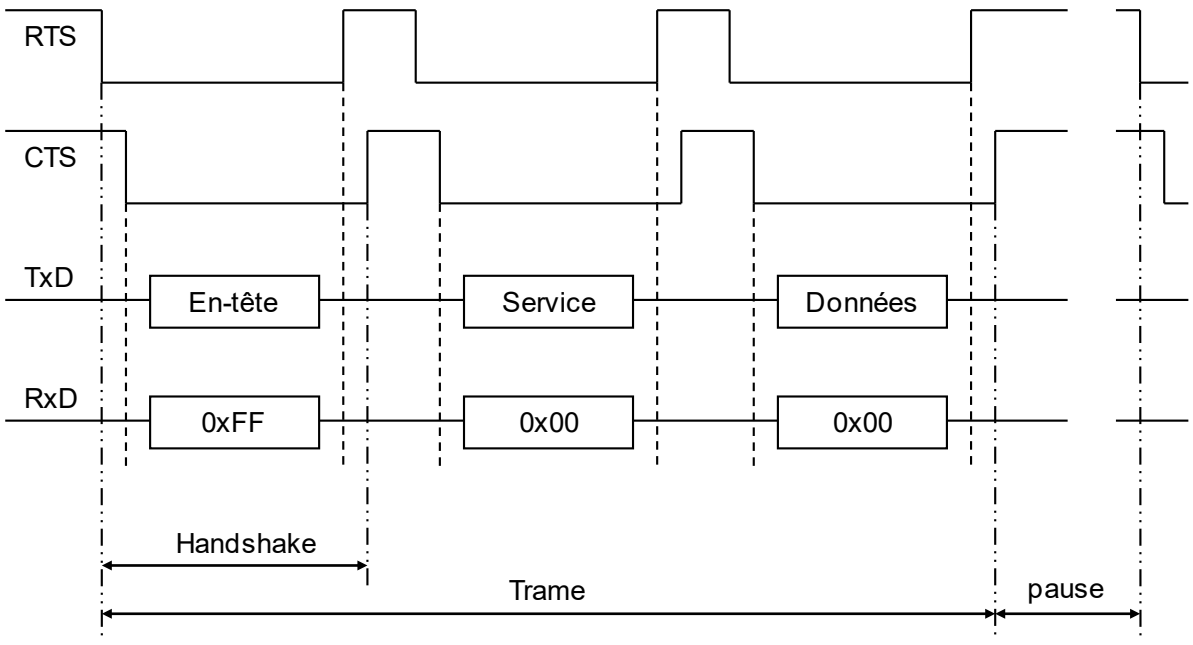
Outre le supplément de technique, le recours à un handshake (prise de conctact) offre les avantages suivants :
- Contrôle de flux au niveau des octets par le matériel Handshake
- Contrôle de flux supplémentaire du sens des données par le logiciel Handshake
- Simplification de la mise en oeuvre sur des systèmes avec une vitesse de communication moindre, des temps d'exécution du programmateur plus longs ou une absence de capacité d'interruption.

Le matériel Handshake est exécuté à chaque octet.
- L'émetteur met sa ligne RTS à LOW et attend que le récepteur signale l'état "Prêt" en mettant sa ligne RTS (ligne CTS de l'émetteur) également à LOW.
- Les données (1 octet) sont transmises.
- L'émetteur met sa ligne RTS à HIGH et attend que le récepteur signale la fin de la communication en mettant sa ligne RTS (ligne CTS de l'émetteur) également à HIGH.

Le logiciel Handshake est exécuté à chaque octet.
- Au premier octet, l'émetteur envoie l'en-tête de trame (valeur différente de 0xFF). Le récepteur acquitte le premier octet avec la valeur 0xFF.
- A chaque nouvel octet, l'émetteur envoie les données. Le récepteur acquitte chaque octet avec la valeur 0x00.

Entre chaque trame, il y a une pause minimale avant qu'une trame suivante puisse être envoyée. Pour de plus amples informations, se reporter au service SetSystemSettings.request.

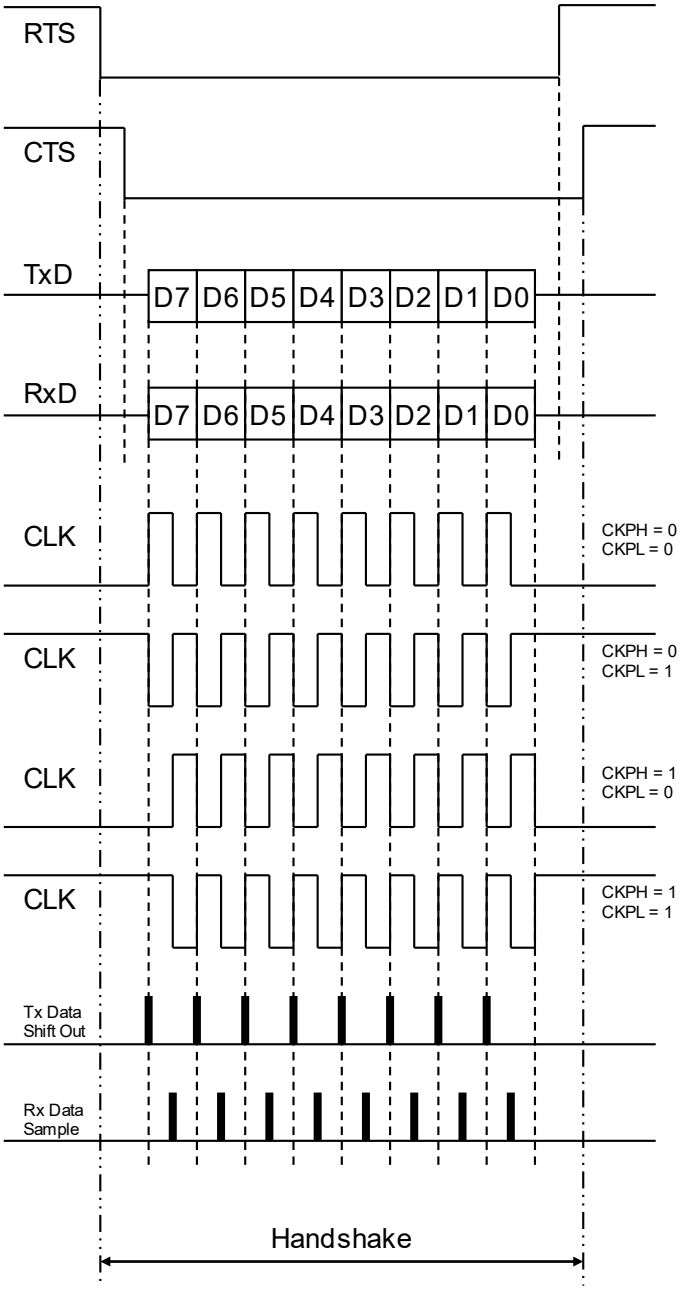
Le principe du handshake est représenté ci-après dans le temps à l'aide d'une trame de 1 octet de données :



3.3.1 Synchrone

En mode sychrone (SPI), le coupleur de bus est supposé être le "Clock-Master" (horloge de référence). Cela veut dire que le coupleur de bus génère le rythme sur la ligne CLK et définit ainsi également le débit binaire de façon implicite. Le débit binaire maximal admissible est visible sur le service SetSystemSettings.request.

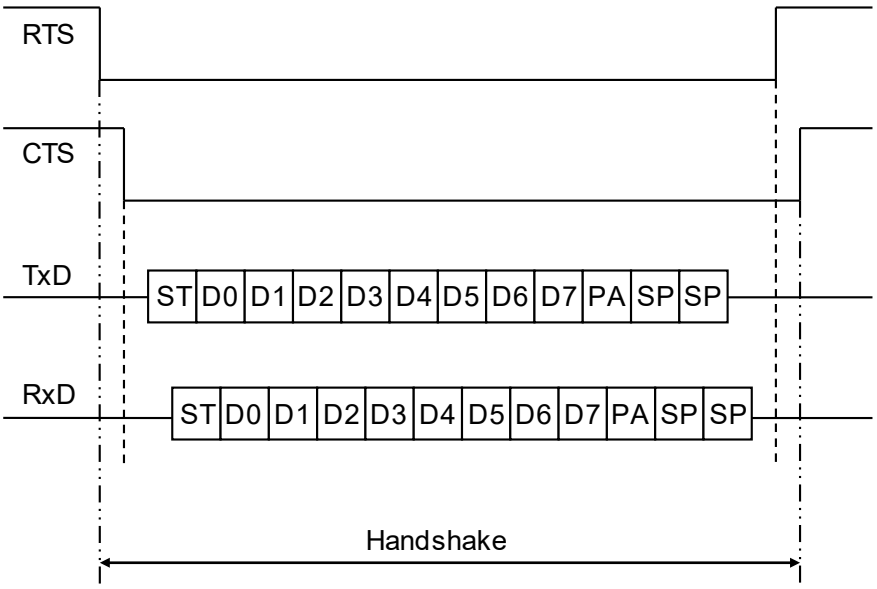
Chaque octet est envoyé et reçu selon la procédure suivante :



Désignation	Description
D7..D0	Bits de données
CKPH	Phase d'horloge configurable par le service SetSystemSettings.request
CKPL	Parité d'horloge configurable par le service SetSystemSettings.request

3.3.2 Asynchrone

Chaque octet est envoyé et reçu selon la procédure suivante :



Désignation	Description
ST	Bit de départ
D0..D7	Bits de données
PA	Bit de parité configurable par le service SetSystemSettings.request
SP	1..2 Bits d'arrêt configurable par le service SetSystemSettings.request

3.3.3 Traitement des erreurs

Pour la transmission d'un octet et la terminaison correcte du handshake, il est prévu un délai "Timeout" selon le service SetSystemSettings.request. Si ce délai est dépassé, il se produit une erreur de communication.

Après une erreur de protocole ou de communication, la liaison en cours est interrompue et l'éventuelle trame incomplète est effacée. Le poussoir met sa ligne RTS à HIGH avant de démarrer une éventuelle liaison suivante. Si le coupleur de bus ne met pas sa ligne RTS aussi à HIGH, une nouvelle tentative de communication est effectuée et acceptée du côté du coupleur de bus.

3.3.4 Master protocole

Le coupleur de bus est défini comme "master protocole".

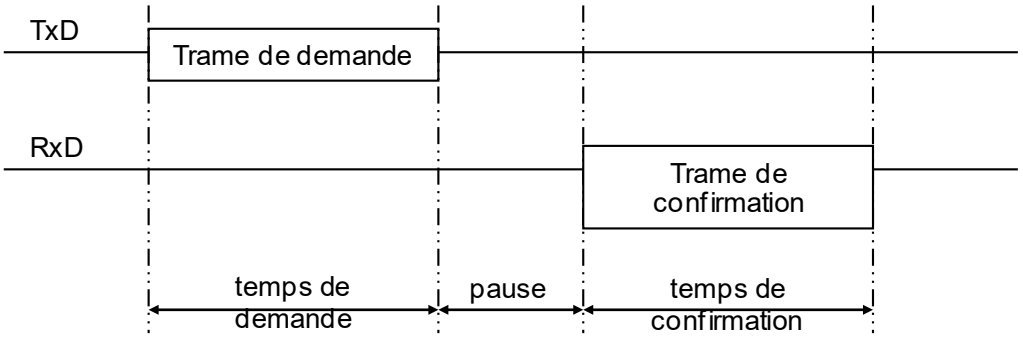
Il est possible que le coupleur de bus et le poussoir veillent établir simultanément une communication. Ceci se manifeste par le fait que ces deux entités envoient un en-tête de trame valide (valeur différente de 0xFF) dans le premier octet.
Dans ce cas, le coupleur de bus ("master protocole") peut envoyer sa trame comme voulu et le poussoir confirme chaque octet selon le logiciel Handshake. Cela signifie que le poussoir rejette sa demande d'émission.

3.4 Protocole, sans handshake

Le mode Communication "Protocole sans handshake" se compose d'un protocole évolué avec des paramètres de communication réglables.
L'utilisation de ce protocole permet d'utiliser pleinement toutes les fonctions du poussoir.

Outre l'absence de contrôle du flux, l'emploi du protocole sans handshake offre les avantages suivants :

- Possibilité d'une définition du logiciel entièrement orientée événement.
- Simplification de la mise en oeuvre sur des systèmes avec une vitesse de communication supérieure, des temps d'exécution du programmateur plus courts ou une capacité d'interruption.

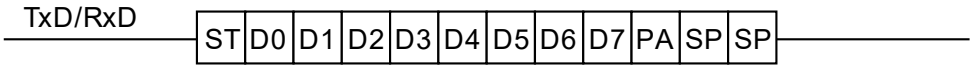


Entre chaque trame, il y a une pause minimale avant qu'une trame suivante puisse être envoyée. Pour de plus amples informations, se reporter au service SetSystemSettings.request.

3.4.1 Asynchrone

En cas de protocole sans handshake, seule une communication asynnchrone est acceptée (pas de SPI).

Chaque octet est envoyé et reçu selon la procédure suivante :



Désignation	Description
ST	Bit de départ
D0..D7	Bits de données
PA	Bit de parité configurable par le service SetSystemSettings.request
SP	1..2 Bits d'arrêt configurable par le service SetSystemSettings.request

3.4.2 Traitement des erreurs

Pour le temps maximal entre 2 octets d'une trame, il est prévu un délai "Timeout" selon le service SetSystemSettings.request. Si ce délai est dépassé et si la trame n'est pas encore achevée, il se produit une erreur de communication.
Après une erreur de protocole ou de communication, la liaison en cours est interrompue et l'éventuelle trame incomplète est effacée. Une nouvelle communication est possible uniquement après l'écoulement du "timeout".

4 Protocole

4.1 Trame

Le protocole se compose fondamentalement de paires de trames de demande et de confirmation. Cela signifie qu'une trame de demande est toujours suivie d'une trame de confirmation lorsque la communication et le traitement se sont déroulés correctement.
Une trame de demande est toujours envoyée par le coupleur de bus. Une trame de confirmation est toujours envoyée par le poussoir.
Il y a en outre des trames d'indication automatiques qui sont émises par le poussoir en cas d'événement déterminé. Les trames d'indication ne sont pas confirmées.

Une trame de longueur n se compose de l'en-tête de trame, du service et des données :

Octet 0	Octet 1	Octet 2 .. (n-1)
En-tête de trame	Service	Données

Désignation	Description
En-tête de trame	L'en-tête de trame est toujours le premier octet d'un service. Il sert à sécuriser les données et la transmission de la trame.
Service	Le service définit le type des données qui suivent et leur utilisation.
Données	Les données contiennent les indications détaillées appartenant au service.

4.1.1 En-tête de trame

L'en-tête de trame est codé comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
P	0	1	L	L	L	L	L

Désignation	Description
P	Le bit de parité D7 est sélectionné de manière à avoir une parité paire sur l'ensemble de l'octet.
01	Les bits D6..D5 forment une partie constante dans l'en-tête de trame.
LLLLL	Les bits D4..D0 définissent la longueur de la trame. Le nombre d'octets est ici compté sans l'octet d'en-tête de trame même. Il s'agit donc du nombre d'octets suivant l'en-tête de trame.

Valeurs spéciales de l'en-tête de trame :

0xA0: Le poussoir effectue une réinitialisation logicielle lorsqu'il reçoit 0xA0 comme en-tête de trame.
Le poussoir envoie 0xA0 comme en-tête de trame lorsqu'il est redémarré.

0xFF: Le récepteur d'une trame envoie 0xFF comme en-tête de trame lorsque le handshake est activé.

4.1.2 Service

Les services sont décrits en détail ci-après. En voici un aperçu :

Valeur	Description
0x10	Service SetSystemSettings.request
0x11	Service SetSystemSettings.confirm
0x12	Service GetSystemSettings.request
0x13	Service GetSystemSettings.confirm
0x18	Service GetSystemState.request
0x19	Service GetSystemState.confirm
0x1A	Service SystemState.indication
0x1C	Service GetSystemInfo.request
0x1D	Service GetSystemInfo.confirm
0x30	Service SetLedState.request
0x31	Service SetLedState.confirm
0x32	Service GetLedState.request
0x33	Service GetLedState.confirm
0x38	Service SetLedBrightness.request
0x39	Service SetLedBrightness.confirm
0x3A	Service GetLedBrightness.request
0x3B	Service GetLedBrightness.confirm
0x40	Service GetButtonState.request
0x41	Service GetButtonState.confirm
0x42	Service ButtonState.indication

4.2 Service SetSystemSettings.request

Ce service demande au poussoir de modifier les paramètres du système.
Après une modification des paramètres du système, une communication correcte doit avoir lieu avec les nouveaux paramètres dans un délai de 10 s. Si ce n'est pas le cas, les anciens paramètres du système sont rétablis.
Si ce service n'est jamais utilisé ou si des valeurs en dehors de la plage définie sont transmises, la valeur par défaut est automatiquement utilisée.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2	Octet 3	Octet 4
En-tête de trame	Service	Données[0]	Données[1]	Données[2]
0xA6	0x10	Communication	Débit binaire	Délai d'attente
		Octet 5	Octet 6	
		Données[3]	Données[4]	
		Ind. Touches	Ind. Etat syst.	

Les paramétrages fondamentaux de la communication se font via l'octet de communication. Il faut tenir compte du fait que toutes les combinaisons possibles ne sont pas acceptées. Pour de plus amples informations, voir Mode communication.
L'octet de communication est codé comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
C1	C0	P1	P0	S	H	M1	M0

Désignation	Description
C1	Phase d'horloge (CKPH) 1 = retardé d'un demi-cycle 0 = normale (non retardée), par défaut (active uniquement si type de communication = synchrone)
C0	Parité d'horloge (CKPL) 1 = Idle HIGH 0 = Idle LOW, par défaut (active uniquement si type de communication = synchrone)
P1	Parité 1 = activée 0 = désactivée, par défaut (active uniquement si type de communication = asynchrone)
P0	Type de parité 1 = paire 0 = impaire, par défaut (active uniquement si type de communication = asynchrone)
S	Nombre de bits d'arrêt 1 = 2 bits d'arrêt 0 = 1 bit d'arrêt, par défaut (active uniquement si type de communication = asynchrone)
H	Handshake 1 = activé 0 = désactivé, par défaut (active uniquement si type de protocole = protocole)
M1	Type de protocole 1 = Echange d'octets 0 = Protocole, par défaut
M0	Type de communication 1 = synchrone (SPI) 0 = asynchrone (UART), par défaut

Le débit binaire est déterminant pour la vitesse de transmission uniquement lorsque le type de communication = asynchrone a été sélectionné.
Si type de communication = synchrone, le débit binaire est utilisé uniquement pour le calcul du délai d'attente. La vitesse de transmission effective peut cependant diverger du débit binaire.
L'octet de débit binaire est codé comme suit :

Valeur	Description
0x01	1200 bauds
0x02	2400 bauds
0x04	4800 bauds
0x08	**9600 bauds, par défaut**
0x10	19 200 bauds
0x20	38 400 bauds
0x30	57 600 bauds
0x60	115 200 bauds (maxi)

Le délai d'attente (Timeout) définit le temps maximal entre 2 octets d'une trame. Si les 2 octets d'une trame ont un espacement supérieur, il apparaît une erreur de protocole.
Pour Handshake = désactivé, la fin de la trame est aussi détectée sur la base de ce Timeout. Le délai d'attente est aussi le temps de pause minimal entre une trame de demande et une trame de confirmation.
L'octet de délai d'attente est codé comme suit :

Valeur	Description
0..1	**Timeout automatique, par défaut** Timeout = 10 * 10 * (1 / Baud Rate) [s]
2..255	Facteur de Timeout Timeout = Facteur * 10 * (1 / Baud Rate) [s] Le délai d'attente maxi est limité à 500 ms.

L'octet d'indication Touches est codé comme suit :

Valeur	Description
0x00	**Indication Touches désactivée, par défaut** L'état des touches est interrogé via le service GetButtonState.request.
0x01	Indication Touches activée L'état des touches est envoyé automatiquement à chaque modification via le service ButtonState.request.

L'octet d'indication Etat du système est codé comme suit :

Valeur	Description
0x00	**Indication Etat du système désactivée, par défaut** L'état du système est interrogé via le service GetSystemState.request.
0x01	Indication Etat du système activée L'état du système est envoyé automatiquement à chaque modification via le service SystemState.indication.

4.3 Service SetSystemSettings.confirm

Ce service permet de confirmer que les paramètres du système ont été modifiés.
C'est la réponse au service SetSystemSettings.request.
Ce service est encore envoyé avec les anciens paramètres du système. Les nouveaux paramètres du système sont utilisés à partir de la communication suivante.

La trame est codée comme suit :

Octet 0	Octet 1
En-tête de trame	Service
0x21	0x11

4.4 Service GetSystemSettings.request

Ce service demande au poussoir d'envoyer les paramètres du système.

La trame est codée comme suit :

Octet 0	Octet 1
En-tête de trame	Service
0x21	0x12

4.5 Service GetSystemSettings.confirm

Ce service permet au poussoir d'envoyer les paramètres du système.
C'est la réponse au service GetSystemSettings.request.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2	Octet 3	Octet 4
En-tête de trame	Service	Données[0]	Données[1]	Données[2]
0xA6	0x13	Communication	Débit binaire	Délai d'attente
		Octet 5	Octet 6	
		Données[3]	Données[4]	
		Ind. Touches	Ind. Etat syst.	

Les paramétrages fondamentaux de la communication se font via l'octet de communication. Il faut tenir compte du fait que toutes les combinaisons possibles ne sont pas acceptées. Pour de plus amples informations, voir Mode communication.
L'octet de communication est codé comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
C1	C0	P1	P0	S	H	M1	M0

Désignation	Description
C1	Phase d'horloge (CKPH) 1 = retardé d'un demi-cycle 0 = normale (non retardée), par défaut (active uniquement si type de communication = synchrone)
C0	Parité d'horloge (CKPL) 1 = Idle HIGH 0 = Idle LOW, par défaut (active uniquement si type de communication = synchrone)
P1	Parité 1 = activé 0 = désactivé, par défaut (active uniquement si type de communication = asynchrone)
P0	Type de parité 1 = paire 0 = impaire, par défaut (active uniquement si type de communication = asynchrone)
S	Nombre de bits d'arrêt 1 = 2 bits d'arrêt 0 = 1 bit d'arrêt, par défaut (active uniquement si type de communication = asynchrone)
H	Handshake 1 = activé 0 = désactivé, par défaut (active uniquement si type de protocole = protocole)
M1	Type de protocole 1 = Echange d'octets 0 = Protocole, par défaut
M0	Type de communication 1 = synchrone (SPI) 0 = asynchrone (UART), par défaut

Le débit binaire est déterminant pour la vitesse de transmission uniquement lorsque le type de communication = asynchrone a été sélectionné.
Si type de communication = synchrone, le débit binaire est utilisé uniquement pour le calcul du délai d'attente. La vitesse de transmission effective peut cependant diverger du débit binaire.
L'octet de débit binaire est codé comme suit :

Valeur	Description
0x01	1200 bauds
0x02	2400 bauds
0x04	4800 bauds
0x08	**9600 bauds, par défaut**
0x10	19 200 bauds
0x20	38 400 bauds
0x30	57 600 bauds
0x60	115 200 bauds (maxi)

Le délai d'attente (Timeout) définit le temps maximal entre 2 octets d'une trame. Si les 2 octets d'une trame ont un espacement supérieur, il apparaît une erreur de protocole.
Pour Handshake = désactivé, la fin de la trame est aussi détectée sur la base de ce Timeout. Le délai d'attente est aussi le temps de pause minimal entre une trame de demande et une trame de confirmation.
L'octet de délai d'attente est codé comme suit :

Valeur	Description
0..1	**Timeout automatique, par défaut** Timeout = 10 * 10 * (1 / Baud Rate) [s]
2..255	Facteur de Timeout Timeout = Facteur * 10 * (1 / Baud Rate) [s] Le délai d'attente maxi est limité à 500 ms.

L'octet d'indication Touches est codé comme suit :

Valeur	Description
0x00	**Indication Touches désactivée, par défaut** L'état des touches est interrogé via le service GetButtonState.request.
0x01	Indication Touches activée L'état des touches est envoyé automatiquement à chaque modification via le service ButtonState.request.

L'octet d'indication Etat du système est codé comme suit :

Valeur	Description
0x00	**Indication Etat du système désactivée, par défaut** L'état du système est interrogé via le service GetSystemState.request.
0x01	Indication Etat du système activée L'état du système est envoyé automatiquement à chaque modification via le service SystemState.indication.

4.6 Service GetSystemState.request

Ce service demande au poussoir d'envoyer l'état du système.

La trame est codée comme suit :

Octet 0	Octet 1
En-tête de trame	Service
0x21	0x18

4.7 Service GetSystemState.confirm

Ce service permet au poussoir d'envoyer l'état du système.
C'est la réponse au service GetSystemState.request.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Données[0]
0x22	0x19	Etat

L'octet d'état renferme l'erreur qui est apparue en dernier. En cas de signalisation Etat = pas d'erreur, cela signifie qu'aucune erreur n'est apparue jusqu'au moment de l'affichage.
L'octet d'état est codé comme suit :

Valeur	Description
0x00	Pas d'erreur
0x10	Service non valable
0x11	Sous-service non valable
0x12	La longueur de trame reçue ne concorde pas avec la longueur de l'en-tête de trame
0x13	La longueur de trame ne concorde pas avec le service.
0x14	En-tête de trame non valable
0x17	Logiciel Handshake non valable
0x18	Demande d'émission simultanée, le poussoir a rejeté la demande d'émission
0x20	Erreur lors de la modification des paramètres du système
0x21	Le handshake n'a pas été activé mais il est nécessaire en cas de SPI
0x22	Débit binaire non valable
0x23	Le délai d'attente (Timeout) a été limité à 500 ms au maximum
0x30, 0x32	Débordement du tampon de réception
0x31, 0x33	Débordement du tampon d'émission
0x34	Erreur pendant la réception d'un octet
0x35	Erreur pendant l'envoi d'un octet
0x36	Délai d'attente dépassé pendant la réception d'un octet
0x37	Délai d'attente dépassé pendant l'envoi d'un octet
0x40	Erreur de trame USART
0x41	Erreur de parité USART
0x42	Débordement USART
0x43	Coupure USART

4.8 Service SystemState.indication

Ce service permet au poussoir d'envoyer automatiquement l'état du système en cas d'apparition d'une erreur.
Ce service peut être activé par le service SetSystemSettings.request.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Données[0]
0x22	0x1A	Etat

L'octet d'état renferme l'erreur qui est apparue en dernier. En cas de signalisation Etat = pas d'erreur, cela signifie qu'aucune erreur n'est apparue jusqu'au moment de l'affichage.
L'octet d'état est codé comme suit :

Valeur	Description
0x00	Pas d'erreur
0x10	Service non valable
0x11	Sous-service non valable
0x12	La longueur de trame reçue ne concorde pas avec la longueur de l'en-tête de trame
0x13	La longueur de trame ne concorde pas avec le service.
0x14	En-tête de trame non valable
0x17	Logiciel Handshake non valable
0x18	Demande d'émission simultanée, le poussoir a rejeté la demande d'émission
0x20	Erreur lors de la modification des paramètres du système
0x21	Le handshake n'a pas été activé mais il est nécessaire en cas de SPI
0x22	Débit binaire non valable
0x23	Le délai d'attente (Timeout) a été limité à 500 ms au maximum
0x30, 0x32	Débordement du tampon de réception
0x31, 0x33	Débordement du tampon d'émission
0x34	Erreur pendant la réception d'un octet
0x35	Erreur pendant l'envoi d'un octet
0x36	Délai d'attente dépassé pendant la réception d'un octet
0x37	Délai d'attente dépassé pendant l'envoi d'un octet
0x40	Erreur de trame USART
0x41	Erreur de parité USART
0x42	Débordement USART
0x43	Coupure USART

4.9 Service GetSystemInfo.request

Ce service demande au poussoir d'envoyer l'information Système.

La trame est codée comme suit :

Octet 0	Octet 1
En-tête de trame	Service
0x21	0x1C

4.10 Service GetSystemInfo.confirm

Ce service permet au poussoir d'envoyer l'information Système.
C'est la réponse au service GetSystemInfo.request.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2	Octet 3
En-tête de trame	Service	Données[0]	Données[1]
0xA3	0x1D	Information logicielle	Information matérielle

"Information logicielle" affiche la version logicielle du microcontrôleur sur le poussoir.
L'octet d'information logicielle est codé comme suit :

Valeur	Description
p.ex. 0x12	High Nibble: Version principale Low Nibble: Version secondaire L'exemple montré signifie : Version 1.2

"Information matérielle" affiche la variante d'équipement du poussoir.
L'octet d'information matérielle est codé comme suit :

Valeur	Description
0x00	4 poussoirs à course courte, 0 LED
0x01	8 poussoirs à course courte, 0 LED
0x02	4 poussoirs à course courte, 6 LED
0x03	8 poussoirs à course courte, 8 LED

4.11 Service SetLedState.request

Ce service demande au poussoir de modifier l'état des LED.

Le poussoir est doté de LED à trois couleurs. Cela signifie que chaque LED peut prendre une des couleurs rouge, vert ou bleu.
Par ailleurs, les LED peuvent fonctionner en mode statique (allumé en continu) ou en mode clignotant (T allumé = T éteint, f = 1 Hz).
Pour chaque LED, il est possible de sélectionner uniquement une couleur et un mode de fonctionnement. Si plusieurs couleurs/modes de fonctionnement sont sélectionnés pour une LED, la solution au conflit se déroule dans l'ordre suivant :

- rouge avant vert avant bleu
- statique avant clignotant

Les LED ne peuvent être commandées qu'en statique avec la variante de trame 1.
La trame est codée comme suit :

Octet 0	Octet 1	Octet 2		
En-tête de trame	Service	Sous-service, Données[0]		
0xA5	0x30	0x11		
		Octet 3	Octet 4	Octet 5
		Données[1]	Données[2]	Données[3]
		LED rouges, statique	LED vertes, statique	LED bleues, statique

Les LED peuvent être commandées en statique ou en clignotant avec la variante de trame 2.
La trame est codée comme suit :

Octet 0	Octet 1	Octet 2		
En-tête de trame	Service	Sous-service, Données[0]		
0x28	0x30	0x12		
		Octet 3	Octet 4	Octet 5
		Données[1]	Données[2]	Données[3]
		LED rouges, statique	LED vertes, statique	LED bleues, statique
		Octet 6	Octet 7	Octet 8
		Données[4]	Données[5]	Données[6]
		LED rouges, clignotant	LED vertes, clignotant	LED bleues, clignotant

L'octet de LED est codé comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
L8	L7	L6	L5	L4	L3	L2	L1

Désignation	Description
L8..L1	Etats des LED L8..L1. 1 = LED allumée, clignotante 0 = LED éteinte

4.12 Service SetLedState.confirm

Ce service permet de confirmer que l'état des LED a été modifié.
C'est la réponse au service SetLedState.request.

Les LED n'ont pu être commandées qu'en statique avec la variante de trame 1.
La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Sous-service, Données[0]
0x22	0x31	0x11

Les LED ont été commandées en statique ou en clignotant avec la variante de trame 2.
La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Sous-service, Données[0]
0x22	0x31	0x12

4.13 Service GetLedState.request

Ce service demande au poussoir d'envoyer l'état des LED.

Avec la variante de trame 1, seules les LED commandées en statique sont demandées.
La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Sous-service, Données[0]
0x22	0x32	0x11

Avec la variante de trame 2, les LED commandées en statique ou en clignotant sont demandées.
La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Sous-service, Données[0]
0x22	0x32	0x12

4.14 Service GetLedState.confirm

Ce service permet au poussoir d'envoyer l'état des LED.
C'est la réponse au service GetLedState.request.

Avec la variante de trame 1, seules les LED commandées en statique sont envoyées. Si une LED clignotante s'allume à cet instant, elle n'est pas considérée comme commandée en statique et n'est pas affichée ici.
La trame est codée comme suit :

Octet 0	Octet 1	Octet 2		
En-tête de trame	Service	Sous-service, Données[0]		
0xA5	0x33	0x11		
		Octet 3	Octet 4	Octet 5
		Données[1]	Données[2]	Données[3]
		LED rouges, statique	LED vertes, statique	LED bleues, statique

Avec la variante de trame 2, les LED commandées en statique ou en clignotant sont envoyées.
La trame est codée comme suit :

Octet 0	Octet 1	Octet 2		
En-tête de trame	Service	Sous-service, Données[0]		
0x28	0x33	0x12		
		Octet 3	Octet 4	Octet 5
		Données[1]	Données[2]	Données[3]
		LED rouges, statique	LED vertes, statique	LED bleues, statique
		Octet 6	Octet 7	Octet 8
		Données[4]	Données[5]	Données[6]
		LED rouges, clignotant	LED vertes, clignotant	LED bleues, clignotant

L'octet de LED est codé comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
L8	L7	L6	L5	L4	L3	L2	L1

Désignation	Description
L8..L1	Etats des LED L8..L1. 1 = LED allumée, clignotante 0 = LED éteinte

4.15 Service SetLedBrightness.request

Ce service demande au poussoir de modifier la luminosité des LED.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2	Octet 3
En-tête de trame	Service	Sous-service, Données[0]	Données[1]
0xA3	0x38	0x11	Luminosité

La luminosité des LED peut être réglée en continu. Il est ainsi possible de réaliser une réduction nocturne par exemple.
L'octet de luminosité est codé comme suit :

Valeur	Description
0	Les LED sont éteintes.
1..255	Valeur de luminosité continue des LED 1 = Luminosité mini 255 = Luminosité maxi, par défaut

4.16 Service SetLedBrightness.confirm

Ce service permet de confirmer que la luminosité des LED a été modifiée.
C'est la réponse au service SetLedBrightness.request.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Sous-service, Données[0]
0x22	0x39	0x11

4.17 Service GetLedBrightness.request

Ce service demande au poussoir d'envoyer la luminosité des LED.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Sous-service, Données[0]
0x22	0x3A	0x11

4.18 Service GetLedBrightness.confirm

Ce service permet au poussoir d'envoyer la luminosité des LED.
C'est la réponse au service GetLedBrightness.request.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2	Octet 3
En-tête de trame	Service	Sous-service, Données[0]	Données[1]
0xA3	0x3B	0x11	Luminosité

La luminosité des LED peut être réglée en continu. Il est ainsi possible de réaliser une réduction nocturne par exemple.
L'octet de luminosité est codé comme suit :

Valeur	Description
0	Les LED sont éteintes.
1..255	Valeur de luminosité continue des LED 1 = Luminosité mini 255 = Luminosité maxi, par défaut

4.19 Service GetButtonState.request

Ce service demande au poussoir d'envoyer l'état des touches.

La trame est codée comme suit :

Octet 0	Octet 1
En-tête de trame	Service
0x21	0x40

4.20 Service GetButtonState.confirm

Ce service permet au poussoir d'envoyer l'état des touches.
C'est la réponse au service GetButtonState.request.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Données[0]
0x22	0x41	Touches

L'octet de touche est codé comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
T8	T7	T6	T5	T4	T3	T2	T1

Désignation	Description
T8..T1	Etats des touches T8..T1. Les touches sont déjà sans rebondissement. 1 = Touche appuyée 0 = Touche relâchée

4.21 Service ButtonState.indication

Ce service permet au poussoir d'envoyer automatiquement l'état des touches à chaque modification.
Ce service peut être activé par le service SetSystemSettings.request.

La trame est codée comme suit :

Octet 0	Octet 1	Octet 2
En-tête de trame	Service	Données[0]
0x22	0x42	Touches

L'octet de touche est codé comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
T8	T7	T6	T5	T4	T3	T2	T1

Désignation	Description
T8..T1	Etats des touches T8..T1. Les touches sont déjà sans rebondissement. 1 = Touche appuyée 0 = Touche relâchée

5 Applications

5.1 Changement de mode Communication

La façon le plus simple d'utiliser le poussoir est de régler le mode Communication et le débit binaire au moyen des résistances de préréglage et de ne pas modifier ce réglage pendant l'exploitation.

S'il n'est pas possible d'équiper les résistances de préréglage et si les paramètres par défaut ne peuvent pas être utilisés, le mode Communication peut aussi être modifié par logiciel. La procédure pour ce faire est la suivante :

- Appliquer la tension d'alimentation
- Attendre (environ 250 ms) que le poussoir envoie 0xA0
- Envoyer le service SetSystemSettings.request avec les paramètres de communication (paramètres de communication par défaut : 9600 bauds, asynchrone)
- Attendre que le poussoir envoie le service SetSystemSettings.confirm (paramètres de communication par défaut : 9600 bauds, asynchrone)
- Modifier les paramètres de communication sur le coupleur de bus
- Envoyer le service GetSystemSettings.request dans les 10 s (nouveaux paramètres de communication)
- Attendre que le poussoir envoie le service GetSystemSettings.confirm (nouveaux paramètres de communication)

5.2 Réinitialisation du poussoir

En plus de la réinitialisation matérielle par une nouvelle application de la tension d'alimentation, on a aussi la possibilité de réinitialiser le poussoir au moyen de la réinitialisation logicielle. La procédure pour ce faire est la suivante :

- Envoyer l'en-tête de trame 0xA0
- Attendre (environ 50 ms) que le poussoir envoie également 0xA0 (Indication de réinitialisation)

6 Annexe

6.1 Configuration des touches et des LED

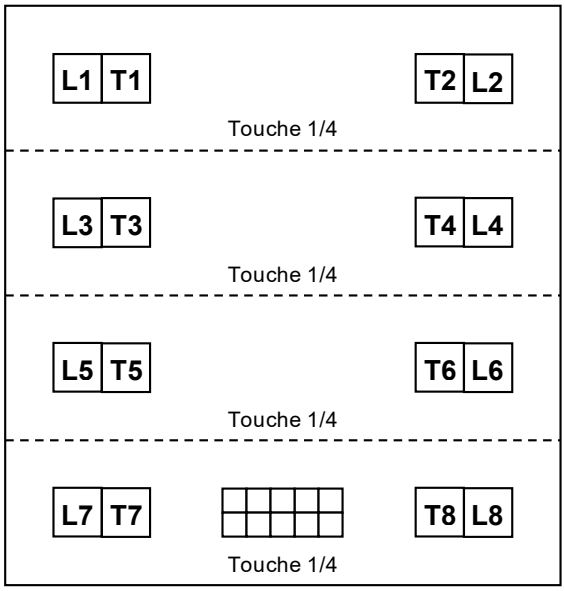
L'affectation des LED (L1..8) et des touches (T1..T8) sur le poussoir est représentée ci-après. L'affectation varie en fonction du type de touches en plastique mis en oeuvre. Les types suivants sont disponibles :

- Touche 1/4
- Touche 1/2
- Touche 1/1

La vue présentée est une vue de dessus. La réglette à bornes de l'interface est vue à partir de l'observateur.

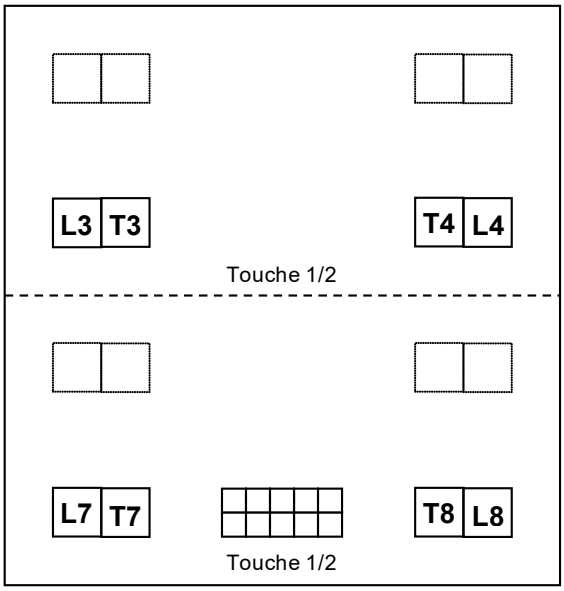
6.1.1 Touche 1/4

En cas d'utilisation de touches 1/4, les LED/touches suivantes peuvent être commandées.



6.1.2 Touche 1/2

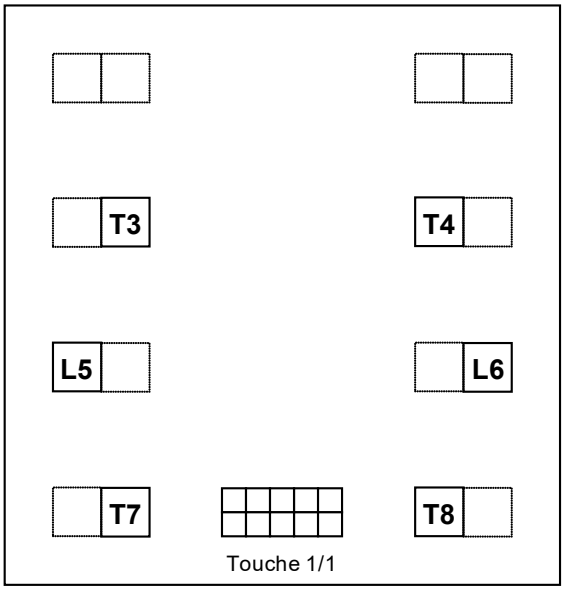
En cas d'utilisation de touches 1/2, les LED/touches suivantes peuvent être commandées.



6.1.3 Touche 1/1

En cas d'utilisation d'une touche 1/1, les LED/touches suivantes peuvent être commandées.

Nota : Pour des raisons mécaniques, la touche 1/1 possède 2 coulisseaux à gauche et à droite. Cela signifie que seule T3, seule T7 ou seules T3/T7 peuvent être commandées ensemble lors de l'activation de la touche à gauche, seule T4, seule T8 ou seules T4/T8 lors de l'activation de la touche à droite.



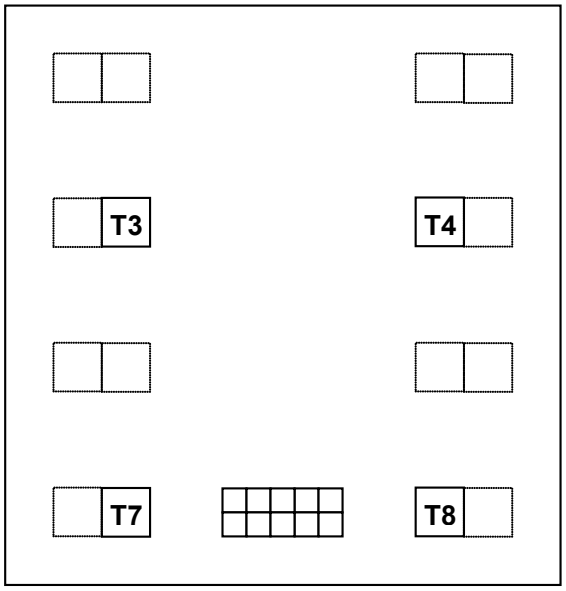
6.2 Variantes d'équipement

Toutes les variantes d'équipement disponibles pour le poussoir sont présentées ci-après. Les LED (L1..8) et les touches (T1..8) équipées sont présentées ici.

La vue présentée est une vue de dessus. La réglette à bornes de l'interface est vue à partir de l'observateur.

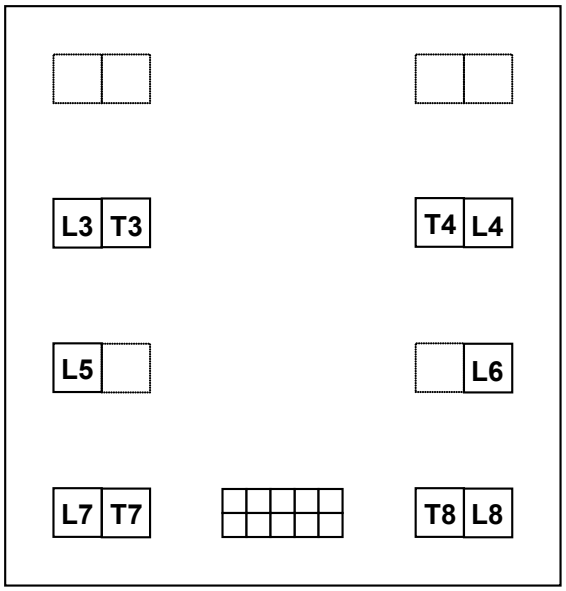
6.2.1 4 poussoirs à course courte, 0 LED

La référence 900-3924.FMI.61 convient pour les touches 1/1 et 1/2 sans LED.



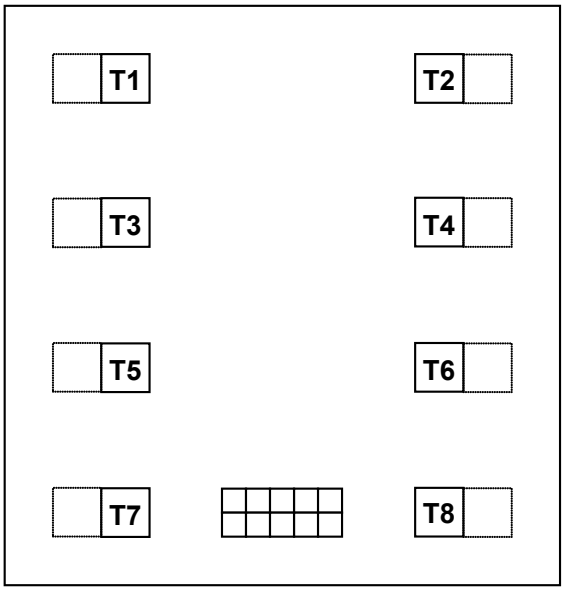
6.2.2 4 poussoirs à course courte, 6 LED

La référence 900-3924.FMI.L.61 convient pour les touches 1/1 et 1/2 avec LED.



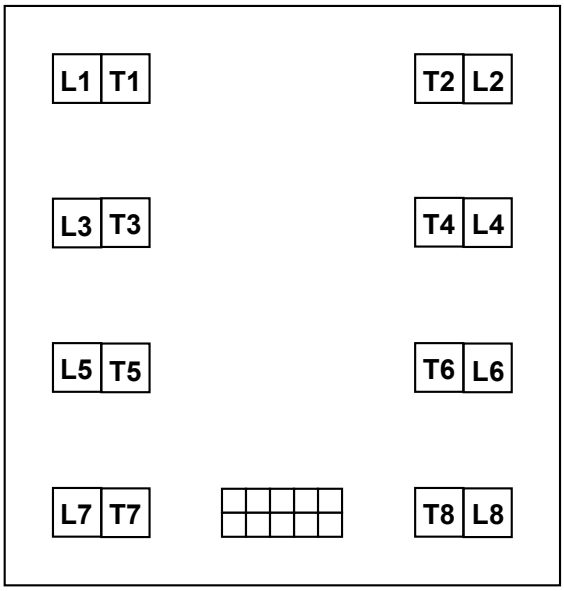
6.2.3 8 poussoirs à course courte, 0 LED

La référence 900-3928.FMI.61 convient pour les touches 1/1, 1/2 et 1/4 sans LED.



6.2.4 8 poussoirs à course courte, 8 LED

La référence 900-3928.FMI.L.61 convient pour les touches 1/1, 1/2 et 1/4 avec LED.



6.3 Logements des résistances de préréglage

Des résistances peuvent être utilisées pour prérégler le mode Communication et le débit binaire. sDans la plupart des cas, ces résistances peuvent être montées directement sur le coupleur de bus. Si cela n'est pas possible, il existe une autre solution.

Le poussoir est pourvu de 2 emplacements destinés à recevoir des résistances 0603-SMD (logements). Les résistances voulues peuvent être insérées. Les emplacements sont repérés de la façon suivante :

- CT: Mode Communication (Com Type)
- BR: Débit binaire (Baud Rate)

6.4 Résistances de rappel à la source/à la masse des lignes d'émission

Les lignes d'émission du poussoir (RxD, CTS) sont dotées de résistances de rappel à la source. Cela signifie que les niveaux des lignes d'émission s'initialisent en synchrone avec la tension d'alimentation lors du démarrage du système.
Ceci présente l'avantage que le coupleur de bus n'interprète pas un front de départ de données (RxD) ou une demande d'émission (CTS) par inadvertance tant que le microcontrôleur du poussoir n'est pas initialisé.

Il est recommandé de doter également les lignes d'émission du coupleur de bus (TxD, RTS) de résistances de rappel à la source.
En cas d'utilisation de la ligne d'horloge (CLK) avec la parité d'horloge Idle LOW (par défaut), il faut recourir à une résistance de rappel à la source et à la masse.
En cas d'utilisation de la ligne d'horloge (CLK) avec la parité d'horloge Idle HIGH, il faut recourir à une résistance de rappel à la source.

6.5 Légende

Explication de quelques expressions et abréviations utilisées dans ce document.

Désignation	Description
MSB	Most Significant Bit/Byte. Bit/Octet de poids fort. Exemple de désignation typique : D7 (valeur 8 bits)
LSB	Most Significant Bit/Byte. Bit/Octet de poids faible. Exemple de désignation typique : D0 (valeur 8 bits)
Nibble	Un demi-octet (4 bits)
0xA0	Les nombres hexadécimaux commencent toujours par "0x" High Nibble / MSB : à gauche Low Nibble / LSB : à droite
Trame	Un télégramme complet du protocole.
Demande (Request)	Demande Le coupleur de bus envoie des trames de demande.
Confirmation	Confirmation Le poussoir envoie des trames de confirmation.
Indication	Affichage Le poussoir envoie des trames d'indication.
RTS	Request to send, demande de liaison
CTS	Clear to send, confirmation de liaison

Feller AG | Postfach | 8810 Horgen | 0844 72 73 74 | customercare.feller@feller.ch | www.feller.ch

Feller SA | Caudray 2| 1020 Renens | 0844 72 73 74 | customercare.feller@feller.ch | www.feller.ch

by Schneider Electric