

Manuel écran CAN 2×16 caractères

Trilys

trilys.com

contact@trilys.com

1115, Rue René Descartes
13857 Aix-En-Provence
France



Table des matières

I - Hardware.....	3
I.1 - Caractéristiques de l'écran.....	3
I.2 - Boutons.....	3
I.2.1 - Version trois boutons.....	3
I.2.2 - Version six boutons.....	4
I.3 - Connecteurs.....	4
I.4 - Switch terminaison CAN.....	6
II - Communication CAN.....	7
II.1 - Généralité CANOpen Trilys.....	7
II.2 - Modification paramètres CAN.....	7
II.2.1 - Numéro identifiant CAN (CAN ID) x5801:00.....	9
II.2.2 - Vitesse CAN x5801:01.....	9
II.3 - Émission de messages.....	9
II.3.1 - Heartbeat.....	9
II.3.2 - TPDO1 0x180+ID Transmit Process Data Objects.....	9
II.3.3 - TPDO2 0x280+ID.....	10
II.4 - Réception de messages Receive Process Data Objects.....	10
II.4.1 - Modification des caractères, mode rapide.....	11
II.4.2 - Modification spéciale de caractère unique : 0x500+ID.....	12

I - Hardware

Cette section décrit les différents éléments physiques que composent l'écran 2×16 (2 lignes / 16 colonnes) de Trilys.

I.1 - Caractéristiques de l'écran

L'écran utilisé se compose de deux lignes (ou quatre lignes pour la version 4×16) et seize colonnes. Chaque caractère faisant 5×8 pixels.



Figure 1 - Caractères de l'écran

Le contraste et la luminosité sont réglables à l'aide de commandes CAN.

La vitesse de rafraîchissement des caractères est de l'ordre de 100ms. Seulement les caractères changeant sont actualisés.

La consommation totale de l'écran est de 28mA avec la luminosité au maximum et moins de 9mA avec le rétroéclairage éteint.

I.2 - Boutons

I.2.1 - Version trois boutons



Figure 2 - Écran avec trois boutons

Remarque : les noms donnés aux boutons sont fictifs, puisqu'ils correspondent à des numéros sur le bus CAN.

I.2.2 - Version six boutons



Figure 3 - Écran avec six boutons

Même remarque que précédemment : les noms des boutons sont fictifs.

I.3 - Connecteurs

À l'arrière du boîtier se trouve les deux connecteurs RJ45. Tous les fils de ces connecteurs sont reliés ensemble, ainsi il n'y a aucune différence. L'intérêt est de laisser la possibilité de brancher d'autres appareils sur le même réseau CAN.

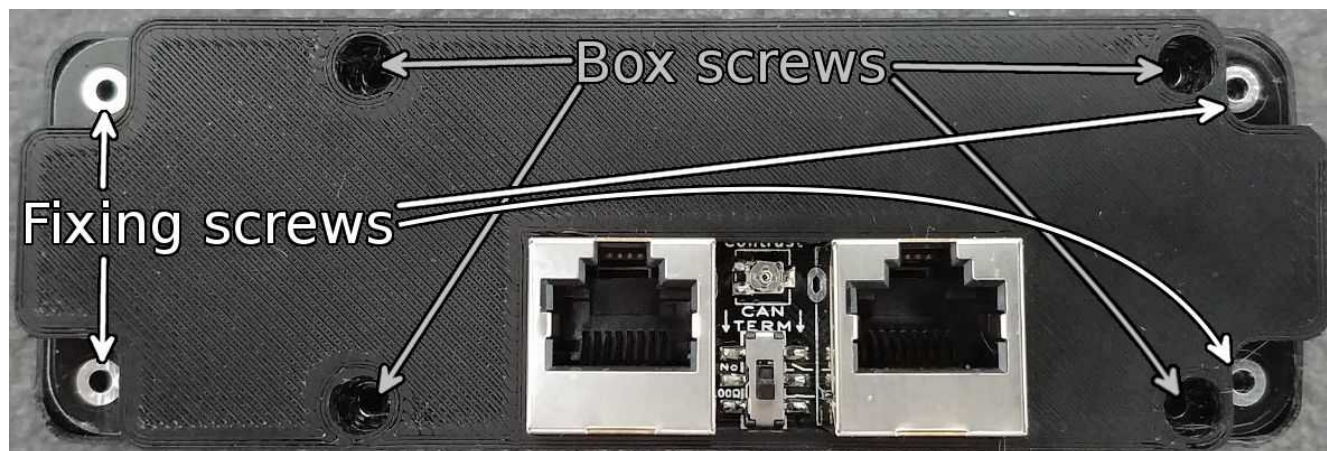


Figure 4 - Arrière de l'écran

Un câble RJ45, non croisé peut être utilisé pour communiquer avec les appareils compatibles Trilys.

Les quatre trous extérieurs servent à fixer l'écran sur un support

Les quatre trous intérieurs servent à fermer l'électronique.

Le composant sous le mot "Contrast" est un potentiomètre utilisé pour modifier le contraste de l'écran. Depuis la version V1r1 du PCB, le contraste est géré par l'électronique et est modifiable à l'aide de commande CAN (depuis la V1r0 du logiciel).

La correspondance du faisceau du câble RJ45 est la suivante :

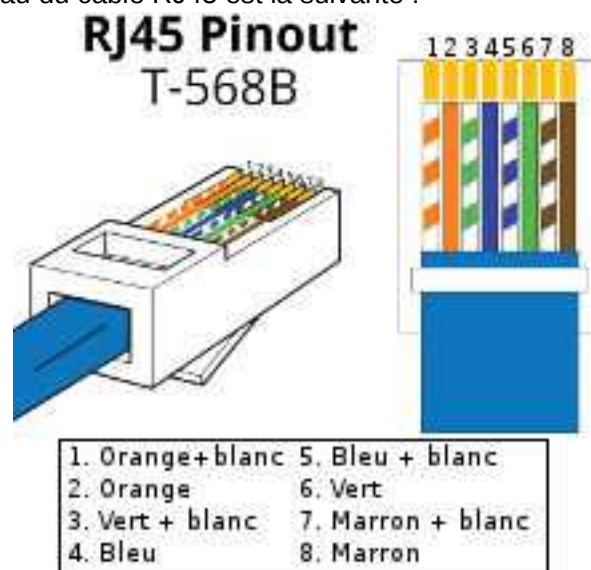


Figure 5 - RJ45

Les câbles de même couleur sont torsadés pour former une paire.

N°	Nom	Description
1	NC	Non connecté.
2	NC	Non connecté.
3	NC	Non connecté.
4	+5V	Alimentation globale.
5	GND	Masse globale.
6	NC	Non connecté.
7	CAN_L	Communication CAN, état bas.
8	CAN_H	Communication CAN, état haut.

Attention : Il ne faut pas relier ce RJ45 à un autre appareil qui n'est pas de marque Trilys. Les signaux peuvent être différents selon les constructeurs.

I.4 - Switch terminaison CAN

Tous les bus de données CAN doivent disposer de terminaison à leurs extrémités. Ainsi si l'écran est en bout de ligne, il faut lui activer la terminaison.

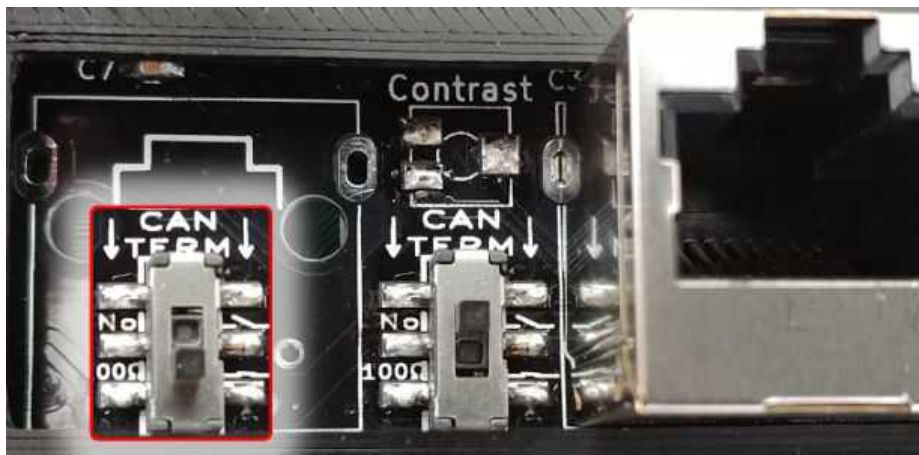


Figure 6 - Activation ou non de la terminaison CAN.

Si l'interrupteur sous les mots "CAN TERM" est orienté vers le haut (cf [Figure 6](#), non encadré en rouge), la terminaison CAN n'est pas activée. À l'inverse, si l'interrupteur est orienté vers le bas (encadré en rouge), la terminaison CAN est activée.

Il est donc nécessaire d'activer cette terminaison (interrupteur en position basse) si l'écran est en bout de ligne CAN.

II - Communication CAN

L'écran CAN 2×16 de Trilys est compatible CAN selon certaines fonctions CANOpen.

II.1 - Généralité CANOpen Trilys

Les sept derniers bits de l'adresse des messages CAN correspondent au numéro d'identifiant "*node ID*".

Par exemple, si l'adresse "0x190" (0x190=0x180+0x10) est reçu dans un message, alors l'identifiant "0x10" a envoyé un message de type "*Transmit Process Data Object #1*" (0x180). Dans notre cas, il s'agit d'un appui sur un bouton.

II.2 - Modification paramètres CAN

Il est possible de modifier certains paramètres enregistrés dans la mémoire de l'écran. **Si vous souhaitez restaurer les paramètres par défaut, il faut appuyer sur les deux boutons "Haut" et "Bas" au démarrage de l'écran. Ce dernier devrait afficher le message "Restauration effectuée.", puis redémarrer.**

La modification des données enregistrées dans la mémoire se fait via la méthode CANOpen. Il faut envoyer le message à l'adresse 0x600+NODE_ID (*par défaut 0x610*). Le message est séparé en quatre groupes : la commande, l'adresse, la sous-adresse et le message.

- Le premier octet est la commande :
 - Demande de lecture : 0x40. Réponse : 0x80 en cas d'erreur, 0x4F si le message fait 1 octet, 0x4B si le message fait 2 octets, 0x43 si le message fait 4 octets.
 - Demande d'écriture : 0x2F si le message fait 1 octet, 0x2B si le message fait 2 octets, 0x23 si le message fait 4 octets. Réponse : 0x80 en cas d'erreur, 0x60 si le message a été enregistré.
- Le second et le troisième octets sont l'adresse (au format *Little Endian* - c'est à dire que l'octet le plus faible est envoyé en premier).
- Le quatrième octet est la sous-adresse.
- Les quatre derniers octets sont le message, au format *Little Endian*.

Résumé des commandes :

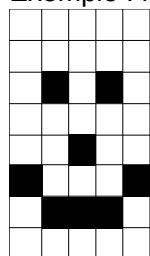
Accès aux données			
Commande	1 Octet	2 Octets	4 Octets
Demande d'écriture	0x2F	0x2B	0x23
Réponse d'écriture	0x60	0x60	0x60
Demande de lecture	0x40	0x40	0x40
Réponse de lecture	0x4F	0x4B	0x43
Réponse d'erreur	0x80	0x80	0x80

Exemple : Pour envoyer la valeur 100 (0x64, de taille 1 octet) à l'adresse 0x4600 et sous-adresse 0, du node ID 0x10, il faut envoyer 0x2F00460064 sur l'adresse CAN 0x610. L'écran doit répondre 0x6000460064 sur l'adresse 0x590 s'il accepte cette valeur, ou 0x80004600.. s'il la refuse.

Résumé des commandes et adresses :

Adresse des données			
Adresse	Sous-adresse	Description	Limites
0x1005	0	Sync ID : Émet un TPDO2 sur réception.	[0;0xff]
0x1017	0	Période du <i>heartbeat</i> en secondes.	[1; 255]
0x4600	0	Niveau de luminosité de l'écran.	[0; 100]
	1	Niveau de contraste de l'écran.	[0; 100]
	2	Délais avant d'éteindre l'écran sans activité (0=Off) en s.	[0; 65535]
	3	Utilisé pour redémarrer l'écran. (2=Forcer sans sauvegarder).	[0; 2]
0x4601	0+8*num	Caractère spécial #num, ligne 0 (en haut).	[0; 31]
	1+8*num	Caractère spécial #num, ligne 1.	[0; 31]
	2+8*num	Caractère spécial #num, ligne 2.	[0; 31]
	3+8*num	Caractère spécial #num, ligne 3.	[0; 31]
	4+8*num	Caractère spécial #num, ligne 4.	[0; 31]
	5+8*num	Caractère spécial #num, ligne 5.	[0; 31]
	6+8*num	Caractère spécial #num, ligne 6.	[0; 31]
	7+8*num	Caractère spécial #num, ligne 7 (en bas).	[0; 31]
0x4602	0	État du bouton haut (1=appuyé, 0=relâché)	[0; 1]
	..	État du bouton droit, bas, gauche, entrer (1=appuyé, 0=relâché)	[0; 1]
	5	État du bouton annuler (1=appuyé, 0=relâché)	[0; 1]
	8	Correspond au 6 bits précédent en 1 octet.	[0x0; 0x3F]
0x5800	0	Activer le TPDO1 (si =1).	[0; 1]
0x5801	0	Node ID utilisé avec CANOpen.	[1; 0x7E]
	1	Vitesse CAN :1:1kk,2:500k,3:250k,4:125k,6:50k,7:20k,8:10k.	[1; 8]

Exemple : Nous allons enregistrer ce smiley dans l'écran pour l'afficher.



0 Chaque caractère fait 40 pixels (8 lignes par 5 colonnes).

10 Chaque ligne est codée sur 5 bits.

0 La ligne 16 8 4 2 1 correspond à la valeur que doit prendre le pixel noir selon sa colonne.

17 Les valeurs à droite correspondent à la valeur de la ligne.

14 Donc l'émission de {0, 0, 10 (0xA), 0, 4, 17 (0x11), 14 (0xE), 0} va afficher un smiley sur l'écran.

16 8 4 2 1

Il faut donc envoyer les trames suivantes :

0x2F01460000, 0x2F01460100, 0x2F0146020A, 0x2F01460300, 0x2F01460404, 0x2F01460511, 0x2F0146060E, 0x2F01460700.

Le caractère est actualisé lors de l'enregistrement de la dernière ligne (ici 0x2F01460700). Si le caractère est déjà affiché sur l'écran, il sera automatiquement mis à jour. Il n'est donc pas possible d'avoir plus de huit caractères spéciaux en même temps.

Pour afficher ce caractère en haut à gauche (du node ID 0x10) il faut envoyer 0x01010E00 à l'adresse 0x510 (plus de détails [II.4.2.Modification spéciale de caractère unique](#)).

II.2.1 - Numéro identifiant CAN (CAN ID) x5801:00

Par défaut le node ID est 0x10. Les identifiants peuvent être n'importe quelle valeur hexadécimale entre 1 et 0x7E. L'adresse de l'identifiant est 0x5801, avec la sous-adresse 00.

Pour le modifier, il faut donc envoyer le message suivant : "0x2F015800 NOUVEL_ID", à l'adresse 0x600+ANCIEN_ID.

Par exemple, l'adresse actuelle est 0x10, et nous voulons la déplacer à l'adresse 0x42, il faut envoyer : 0x2F01580042 à l'adresse 0x610. Puis redémarrer avec 0x2F00460301 à l'ancienne adresse, donc 0x610.

II.2.2 - Vitesse CAN x5801:01

Par défaut la vitesse du CAN est de 500kbps. Pour la modifier, il faut se référer au tableau suivant :

Vitesse (kbps)	1000	500	250	125	50	20	10
Valeur CAN	1	2	3	4	6	7	8

Pour la changer à 250kbps : il faut envoyer 0x2F01580103 puis redémarrer en envoyant 0x2F00460301. À partir de ce redémarrage, la fréquence du CAN de l'écran sera passée à 250kbps.

II.3 - Émission de messages

L'écran émet automatiquement des messages aux adresses 0x180+ID (TPDO1), 0x280+ID (TPDO2) et répond aux SDO à l'adresse 0x580+ID.

II.3.1 - Heartbeat

Par défaut, toutes les 3 secondes l'écran émet des *heartbeats* pour montrer son état. Ils sont à l'adresse 0x700+ID. Ils prennent trois états possibles :

- 0x00 : Démarrage en cours.
- 0x7F : Fin de démarrage. Écran opérationnel.
- 0x05 : Toutes les 3 secondes si l'écran est opérationnel.

II.3.2 - TPDO1 0x180+ID *Transmit Process Data Objects*

Lors de l'appui sur un bouton, l'écran émet des TPDO (*Transmit Process Data Objects*) à l'adresse 0x180+ID. Ces TPDO contiennent deux octets :

- Octet 1 : Valeur du bouton appuyé ou relâché. Le bit de poids fort indique l'état du bouton (1=appui, 0=lâché). Les sept bits de poids faibles indiquent le dernier bouton émettant l'action.
- Octet 2 : Valeur des boutons appuyés depuis plus de 700ms.

Les bits sont répartis de la manière suivantes (bit de poids faible vers fort) : 1 : Haut, 2 : Droit, 3 : Bas, 4 : Gauche, 5 : Entrer, 6 : Annuler et 8 : État de transition (0 : descendant, 1: montant).

- 0x8100 : Appui sur bouton haut.
- 0x0100 : Relâchement du bouton haut.
- 0x8100 : Appui sur bouton haut.
- 0x8101 : Appui long sur le bouton haut (700ms plus tard).
- 0x8101 : Toujours appui long sur le bouton haut (200ms plus tard).
- 0x0100 : Relâchement du bouton haut.
- 0x8400 : Appui sur bouton bas.
- 0x0400 : Relâchement du bouton bas.

Les touches peuvent également être pressées en même temps.

II.3.3 - TPDO2 0x280+ID

Sur réception d'un *Sync ID* (0x1005), l'écran émet l'état de ses boutons, de la même façon que pour le TPDO1, les bits sont répartis : 1 : Haut, 2 : Droit, 3 : Bas, 4 : Gauche, 5 : Entrer, 6 : Annuler :

- 0x01 : Haut cliqué.
- 0x05 : Haut et bas cliqués.
- 0x04 : Bas cliqué.

L'écran attend 5×NODE_ID avant de répondre.

II.4 - Réception de messages *Receive Process Data Objects*

L'écran implémente certaines fonctions ressemblant à CANOpen. Cependant les adresses 0x100 correspondantes au *timestamp* sont remplacées par un RPDO.

En effet, les messages aux adresses 0x100+ID, 0x200+ID, 0x300+ID, 0x400+ID, 0x500+ID sont des RPDO.

Pour afficher un caractère, il faut se référer à la table ASCII suivante :

HEX	Char	HEX	Char	HEX	Char	HEX	Char	HEX	Char	HEX	Char	HEX	Char
0	Spec0	28	(38	8	48	H	58	X	68	h	78	x
1	Spec1	29)	39	9	49	I	59	Y	69	i	79	y
2	Spec2	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
3	Spec3	2B	+	3B	;	4B	K	5B	[6B	k	7B	{
4	Spec4	2C	,	3C	<	4C	L	5C	¥	6C	l	7C	
5	Spec5	2D	-	3D	=	4D	M	5D]	6D	m	7D	}
6	Spec6	2E	.	3E	>	4E	N	5E	^	6E	n	7E	→
7	Spec7	2F	/	3F	?	4F	O	5F	_	6F	o	7F	←
20	Espace	30	0	40	@	50	P	60	`	70	p		
21	!	31	1	41	A	51	Q	61	a	71	q		
22	"	32	2	42	B	52	R	62	b	72	r		
23	#	33	3	43	C	53	S	63	c	73	s		
24	\$	34	4	44	D	54	T	64	d	74	t		
25	%	35	5	45	E	55	U	65	e	75	u		
26	&	36	6	46	F	56	V	66	f	76	v		
27	'	37	7	47	G	57	W	67	g	77	w		

Figure 7 - Table ASCII de l'écran

Les huit premiers caractères (noté Spec0 → 7) sont les caractères pré-enregistrés [à l'adresse 0x4601](#).

Le nombre d'octets envoyés dans un message CAN étant limités à huit, il existe cinq adresses permettant d'afficher n'importe quel caractère, n'importe où sur l'écran.

Pour effacer un caractère, il suffit d'envoyer un espace (0x20).

II.4.1 - Modification des caractères, mode rapide

L'écran est séparé en quatre sections correspondantes à des adresses différentes.

- 0x100+ID : Écran haut gauche
- 0x200+ID : Écran haut droit
- 0x300+ID : Écran bas gauche
- 0x400+ID : Écran bas droit

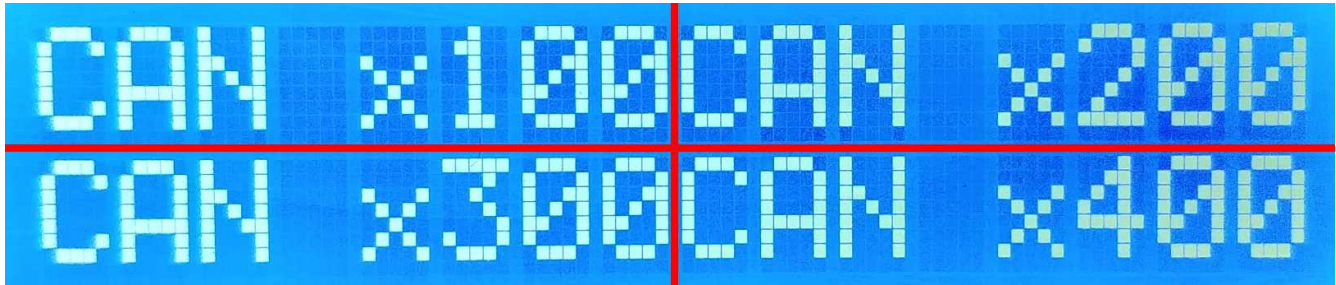


Figure 8 - Disposition des adresses CAN en fonction de la position sur l'écran

Pour cet écran il faut envoyer 0x43414e2078313030 à l'adresse 0x100+ID, puis 0x43414e2078323030 à 0x200+ID, puis 0x43414e2078333030 à 0x300+ID et 0x43414e2078343030 à 0x400+ID.

Exemple :

- Pour afficher "Ligne 1", en haut à gauche, il faut envoyer 0x4C69676E652031 ([cf Table ASCII](#)) à l'adresse 0x100+ID.
- Pour afficher "Fin de 1", en haut à droite, il faut envoyer 0x46696E2064652031 ([cf Table ASCII](#)) à l'adresse 0x200+ID.
- Pour afficher "Debut 2", en bas à gauche, il faut envoyer 0x44656275742032 ([cf Table ASCII](#)) à l'adresse 0x300+ID.
- Pour afficher "Dernier ←", en bas à droite, il faut envoyer 0x4465726E6965727F ([cf Table ASCII](#)) à l'adresse 0x400+ID.

La succession des quatre messages précédent donnent l'image suivante :



II.4.2 - Modification spéciale de caractère unique : 0x500+ID

Le dernier RPDO disponible permet d'afficher de nombreux messages sous plusieurs formes à n'importe quelle localisation.

Les huit octets du message sont définis dans le tableau suivant.

#Octet	Description	Limites
1	Numéro de ligne (1=Haut, 2=Bas)	[1; 2]
2	Numéro de colonne (1=Gauche, 16 (0x10) =Droite)	[1; 16]
3	Type de données *	[0; 14]
4, 5, 6, 7 et 8	Données	[0; 0xffffffff]

* Les types de données disponibles sont les suivants :

#Octet 3	Description
1	Donnée ASCII : 1 caractère à l'octet 4.
2	Donnée ASCII : 2 caractères aux octets 4 et 5.
3	Donnée ASCII : 3 caractères aux octets 4, 5 et 6.
4	Donnée ASCII : 4 caractères aux octets 4, 5, 6 et 7.
5	Donnée ASCII : 5 caractères aux octets 4, 5, 6, 7 et 8.
6	Donnée non signée sur 8 bits (uint8) de l'octet 4.
7	Donnée signée sur 8 bits (int8) de l'octet 4.
8	Donnée au format hexadécimal de l'octet 4.
9	Donnée au format hexadécimal (rempli de 0 pour faire 2 caractères) de l'octet 4.
10 (0x0A)	Donnée non signée sur 16 bits (uint16) des octets 4 et 5 ($[4]+256*[5]$).
11 (0x0B)	Donnée signée sur 16 bits (int16) des octets 4 et 5 ($[4]+256*[5]$).
12 (0x0C)	Donnée non signée sur 32 bits (uint32) des octets 4, 5, 6, 7 ($[4]+256*[5]+256^2*[6]+256^3*[7]$).
13 (0x0D)	Donnée signée sur 32 bits (int32) des octets 4, 5, 6 et 7 ($[4]+256*[5]+256^2*[6]+256^3*[7]$).
14 (0x0E)	Affiche le caractère spécial de 0 à 7.

Exemple :

- Pour afficher un simple caractère (une flèche) en bas à droite, il faut envoyer : 0x0210017F.
- De même pour 2, 3, 4 et 5 caractères, la commande est la même.
- Pour afficher un entier non signé sur 8 bits (123=0x7b) en haut à gauche, il faut envoyer 0x0110067B.
- Pour afficher un octet au format hexadécimal, zéro padding sur 2 caractères (0x2) en haut au milieu, il faut envoyer 0x01070942.
- Pour envoyer un entier signé sur 16 bits (-12345=0xCFC7) en bas au milieu, il faut envoyer 0x02060BC7CF.
- Pour afficher le premier caractère spécial en bas à gauche, il faut envoyer : 0x02010E00.