

COMMUNICATION AVEC LES AFFICHEURS SÈRIE DM VIA PROTOCOLE DTPM.

INDEX

1. DESCRIPTION

2. PROTOCOLE DTPM

2.1. Protocole DTPM.

2.2. Commandes Protocole DTPM.

2.3. Options de Visualisation.

2.4. Variables Alphanumériques.

2.5. Test du protocole DTPM avec le Software MP Tools.

2.6. Test du protocole DTPM avec le Software Dynamic3

Annexe 1. Configuration par défaut des afficheurs

Annexe 2. Codes d'Erreur Protocole DTPM

Annexe 3. Structure de données *DEVICE_USER_SETTINGS*

1. DESCRIPTION

Il est possible de contrôler les afficheurs série DM de différentes manières suivant le modèle.

Le Protocole DTPM, est le protocole natif de ces appareils

Ce document détaille la communication avec le protocole de communication DTPM via le port série, USB ou Ethernet (TCP / IP) ainsi que les commandes et les options nécessaires pour interagir avec ces afficheurs.

2. PROTOCOLE DTPM

2.1. Protocole DTPM.

2.1.1. Description du Protocole.

Le Protocole DTPM, est le protocole natif des afficheurs DM.

Le protocole de transmission comprend un byte de synchronisation, la longueur du paquet, le numéro de dispositif, la commande envoyée à l'afficheur, les données (le cas échéant) et le checksum. Chaque paquet est envoyé via le port USB, RS232 / RS485 ou Ethernet et si il est correctement réceptionné, l'appareil retourne une réponse.

Le paquet doit être transféré de nouveau seulement si des erreurs de transmission ont été détectées (retournées par l'afficheur) ou bien si le ACK (accusé de réception) n'a pas été reçu.

La connexion entre l'afficheur (esclave) et le dispositif qui assure la fonction de maître, peut se faire par plusieurs canaux de communication, comme le montre le tableau suivant:

Port	Connexion Protocole DTPM
RS232	SI
RS485	SI
USB	USB Device – Fonction CDC (VCP)
ETHERNET (TCP)	<u>Il faut ouvrir le Socket du Port 53¹ avec l'adresse IP du Dispositif</u>

Tableau 1: Option de connexion pour le DTPM

¹Le port TCP peut se modifier dans la configuration du dispositif.

2.1.2. Structure du Protocole.

Chaque paquet du DTPM est formé par trois blocs principaux:

- Header (entête)
- Données (optionnel)
- Checksum (Somme de contrôle)

La figure ci-dessous représente la composition d'un paquet.



Figure 1: Composition du DTPM

2.1.3. Champs des paquets.

Le tableau suivant montre les champs des paquets du protocole DTPM

	# BYTE	VALEUR	NOM	DESCRIPTION
HEADER	1	0x16	SYN	Byte de synchronisme
	2	LEN1	LEN	LSB nombre bytes paquet
	3	LEN2	LEN	MSB nombre bytes paquet
	4	ID	ID	Adresse dispositif destination
	5	OD	OD	Code Commande à exécuter
DONNÉES	6	DATE[0]	Premier byte de données	Premier byte de données (le cas échéant)

	6+n ²	DATA[n-1]	Dernier byte de données	Dernier byte de données (le cas échéant)
CHECKSUM	5+n+1	MSB Checksum	LSB du Checksum	
	5+n+2	LSB Checksum	MSB du Checksum	

Figure 2: Champs du DTPM

²Où n=nombre de bytes de données. Si n=0, le LSB du Checksum restera dans le #BYTE 6 et le MSB du Checksum dans le #BYTE 7

- **LEN:** Indique le nombre d'bytes du paquet (tous les bytes entre SYN et le checksum, ceux-ci inclus).
- **ID:** Indique le numéro d'identification de l'appareil.
- **OD:** Commande (1 byte). Indique le code de la commande à exécuter

Le champ de données doit être placé seulement dans certains types de paquet. Les autres champs sont obligatoires et sont placés sur tous les types de paquets.

2.1.4. Checksum.

À la fin du paquet, il faut envoyer le **Checksum** (somme de contrôle) pour vérifier qu'il n'y ait pas eu d'erreurs lors de la transmission. Le **Checksum** est la somme de tous les bytes dans le paquet (sauf checksum), le byte de poids faible (LSB) s'envoie en premier.

Si le **Checksum** ne correspond pas, le destinataire rejette le paquet, et le dispositif d'origine, à la fin du temps d'attente pour l'ACK, renverra le même paquet.

2.1.5. Réponse du dispositif

Si l'ID du paquet est **0xFF**³, tous les dispositifs l'accepteront et exécuteront la commande correspondante, mais aucune réponse de sera retournée. Si l'ID du paquet correspond à l'adresse **LocalCast**⁴ de un ou plusieurs dispositifs, ceux ci accepteront le paquet sans répondre , mais les autres dispositifs l'ignoreront.

La réponse du dispositif se compose de 2 bytes:

	# BYTE	VALEUR	NOM	DESCRIPTION
	1	0x06	ACK	ACK
	2	ERROR_CODE	CODE D'ERREUR	

Table3: Réponse du dispositif en Protocole DTPM

Si l'ordre est arrivé correctement, une confirmation sera reçue. Elle se compose d'une validation avec un code fixe (ACK) et d'un code d'erreur du dispositif. Il sera normalement 0 (ce qui signifie que l'opération a réussi), **cependant il peut arriver qu'une commande illégale soit envoyée**. La valeur des différents codes d'erreur est indiquée dans l'Annexe 2.

La commande CHECKSUM (0x07) retourne le byte LSB du dernier Checksum correctement reçu à la place du code d'erreur.

La commande GET NUM PACKET (0x21) retourne le numéro du dernier paquet **SEND** correctement reçu par le dispositif. Le compteur est remis à 0 si l'appareil reçoit un paquet **EDIT** ou **EDITFILE** et s'incrémente a valider un paquet **SEND**.

La commande GET BAT LEVEL (0x96) retourne un byte⁵ qui indique le niveau de la batterie de l'appareil à la place du code d'erreur.

³ Adresse BROADCAST du Protocole DTPM.

⁴ Paramètre de la Configuration du Dispositif.

⁵ Voir description de la commande GET BAT LEVEL.

2.1.6. Erreur de Transmission.

En cas de transmission UNICAST (non en mode BROADCAST ou LOCALCAST), le paquet doit être retransmis que si l'afficheur a informé d'erreurs de transmission ou si le ACK n'a pas été retourné. En cas de non réception de l'ACK la procédure suivante doit être suivie pour déterminer si il est nécessaire de transmettre le paquet:

- 1) Envoyer la commande **GET NUM PACKET** suite à une erreur de communication >100ms.
- 2) L'afficheur envoie le numéro du dernier paquet reçu correctement. Attendre jusqu'à 3 secondes.
- 3) Si aucune réponse n'est reçue, répéter 3 fois la même procédure depuis 1) .

Si aucune réponse n'est reçue, on peut donner la communication comme interrompue. Si il y a réponse continuer comme suit:

- 4) Envoyer la commande **CHECKSUM** après 100ms.
- 5) L'afficheur envoie le Checksum du dernier paquet reçu correctement. Attendre jusqu'à 3 secondes.
- 6) Si aucune réponse n'est reçue, répéter 3 fois la même procédure depuis 4) .

Si aucune réponse n'est reçue, on peut donner la communication comme interrompue. Si il y a réponse continuer comme suit:

Si le Checksum et le numéro de paquet correspond à celui du dernier paquet envoyé, nous pouvons dire que l'afficheur a reçu et traité avec succès le paquet et il est donc possible de continuer la transmission.

En cas contraire:

- 7) Envoyer la commande **GET NUM PACKET** après 100ms.
- 8) L'afficheur envoie le numéro du dernier paquet reçu correctement. Attendre jusqu'à 3 secondes.
- 9) Si aucune réponse n'est reçue, répéter 3 fois la même procédure depuis 7)

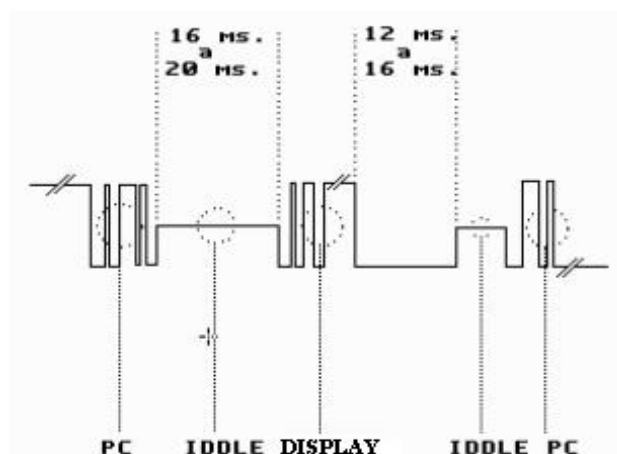
Si le Checksum et le numéro de paquet correspond à celui du dernier paquet envoyé, nous pouvons dire que le premier **GET NUM PACKET** avait reçu des données incorrectes pour la raison que ce soit mais que l'afficheur a reçu et traité avec succès le paquet et il est donc possible de continuer la transmission.

En cas contraire, il est conclu que le colis n'a pas été reçu correctement par l'affichage et doit donc être transmis.

La transmission d'informations dans un afficheur sans appliquer correctement les règles de transmission et de contrôles adéquats peut provoquer des erreurs et des dysfonctionnements au niveau de l'affichage.

2.1.7. Communication par Bus RS485.

Ce système est basé sur la transmission de données via des tensions différentielles. Il est conçu pour supporter dans la même ligne de communication jusqu'à 32 émetteurs / récepteurs RS485. Le système se base sur le principe qu'il n'y a qu'un seul émetteur actif à un moment donné; les autres se maintiennent en mode réception, en passant en état de haute impédance. Donc, il n'y a pas de conflit entre les lignes, un récepteur peut devenir émetteur à son tour à la condition qu'aucun autre élément n'émette à ce moment. Le graphique ci-dessous montre comment doit s'établir la communication entre deux éléments d'une ligne RS485.



En premier lieu apparaissent les informations transmises par le DTE. On peut voir que quand elle se termine, le driver du DTE se met en mode réception, en passant en haute impédance ou IDDLLE LINE.

Au bout de 16 à 20 millisecondes, l'afficheur active son driver de transmission, et commence à transmettre si la réception a réussi (NOTE: dans le cas de commandes impliquant un long temps de traitement de la part de l'afficheur, telles que celles relatives à l'édition de fichiers, ce temps peut être retardé jusqu'à 3 secondes).

Une fois terminé, l'afficheur reste en mode de transmission de 12 à 16 millisecondes. Passé ce délai, il repasse en mode réception, le DTE peut commencer alors une nouvelle transmission. Si ces délais ne sont pas appliqués, il y aura probablement un conflit de ligne avec les erreurs de transmission qui en découlent.

2.2. Commandes du protocole DTPM.

A continuation les différentes commandes du Protocole sont détaillées avec une brève description et un exemple de son utilisation. Sauf indication contraire, tous les exemples sont effectués avec des afficheurs d'ID = 1. La notation est en hexadécimal, sauf indication contraire.

RESET RAM (0x01)

Données: SANS DONNÉES

Réponse: ACK.

Description: Reset du dispositif avec effacement de la mémoire non volatile. Lors du redémarrage aucun programme ne s'exécutera.

Cette commande ne supprime pas les fichiers mémorisés dans le répertoire de l'afficheur (programmes de visualisation, graphiques, polices...).

Exemple:

Envoyé à l'afficheur: 16 07 00 01 01 1F 00
Réponse de l'afficheur: 06 00

RESTART (0x02)

Données: SANS DONNÉES

Réponse: ACK.

Description: Reset du dispositif sans effacement de la mémoire non volatile.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 02 20 00
Réponse de l'afficheur: 06 00

STOP (0x03)

Données: SANS DONNÉES

Réponse: ACK.

Description: Arrêt du programme en exécution.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 03 21 00
Réponse de l'afficheur: 06 00

STOP AND CLEAR (0xA1)

Données: SANS DONNÉES

Réponse: ACK.

Description: Arrêt du programme en exécution et effacement mémoire non volatile

CHECKSUM (0x07)

Données: SANS DONNÉES

Réponse: ACK + byte de poids faible du dernier Checksum reçu correctement.

Description: L'afficheur retourne le byte de poids faible du Checksum du dernier paquet reçu correctement, à l'exception des paquets CHECKSUM et GET NUM PACKET.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 07 25 00

Réponse de l'afficheur: 06 XX

Où XX est le byte de poids faible du Checksum du dernier paquet reçu correctement.

SET TIME (0x0A)

Données: 6 Bytes pour indiquer l'heure et la date:

Année (1 Byte): 0 - 99

Mois (1 Byte): 1 - 12

Jour (1 Byte): 1 - 31

Heure (1 Byte): 0 - 23

Minutes (1 Byte): 0 - 59

Secondes (1 Byte): 0 - 59

Réponse: ACK.

Description: Envoyer la date et l'heure à l'horloge interne de l'afficheur.

Le format est 1 byte pour l'année (00-99), 1 byte pour le mois (1-12), 1 byte pour le jour (1-31), 1 byte pour l'heure (0-23), 1 byte pour les minutes (0- 59) et 1 byte pour les secondes (0-59).

Exemple:

Heure 13:40:00 Date 02/03/2014

Envoyé à l'afficheur: 16 0D 00 01 0A 0E 03 02 0D 28 00 76 00

Réponse de l'afficheur: 06 00

GET TIME (0x0B)

Données: SANS DONNÉES

Réponse: ACK + **PAQUET SEND**

Données PAQUET SEND: 6 Bytes pour indiquer l'heure et la date:

Année (1 Byte): 0 - 99

Mois (1 Byte): 1 - 12

Jour (1 Byte): 1 - 31

Heure (1 Byte): 0 - 23

Minutes (1 Byte): 0 - 59

Secondes (1 Byte): 0 - 59

Description: Pétition de la date et l'heure de l'horloge interne de l'afficheur, lequel répond en envoyant un paquet SEND au PC (ID *0xFE*).

Le format est 1 byte pour l'année (00-99), 1 byte pour le mois (1-12), 1 byte pour le jour (1-31), 1 byte pour l'heure (0-23), 1 byte pour les minutes (0-59) et 1 byte pour les secondes (0-59).

Exemple:

Envoyé à l'afficheur:	16 07 00 01 0B 29 00
Réponse de l'afficheur:	06 00
	16 0D 00 FE 0C 0E 03 02 0D 28 13 88 01

Ce qui indique que l'afficheur a l'heure *13:40:19* et la date *02/03/2014*

SEND (0x0C)

Données: *Différent pour chaque cas.*

Réponse: ACK.

Description: Cette commande est utilisée si le PC envoie quelque chose à l'afficheur et vice versa. Ce paquet peut contenir tout type de données, comme programmes, polices ou graphiques. Le contenu des données du paquet est différent pour chaque cas.

GETVER (0x12)

Données: SANS DONNÉES

Réponse: ACK + **PAQUET SEND**

Données PAQUET SEND: Structure GETVER (6 Bytes):

Version du Software du Dispositif (1 Byte): 10 - 254

Version du Hardware (1 Byte): 1 - 255

Nombre de Colonnes (2 Bytes).

NON UTILISÉ (1 Byte).

Nombre de lignes (1 Byte).

Description: Pétition d'informations sur le Hardware et Software de l'afficheur qui répond à l'ordinateur (ID 0xFE) avec un paquet SEND. Une différence de la structure GERVER EXT para rapport à GETVER est que cette dernière est de taille fixe-ci (6 bytes).

Exemple:

Envoyé à l'afficheur: 16 07 00 01 12 30 00

Réponse de l'afficheur: 06 00

16 0D 00 FE 0C 2E C4 60 00 01 06 86 02

Ce qui indique:

- *Version du Software du Dispositif (1 Byte):*

4.6

- *Version du Hardware (1 Byte):*

196

- *Nombre de Colonnes (2 Bytes):*

96

- *NON UTILISÉ (1 Byte).*

1

- *Nombre de lignes (1 Byte):*

6

NEXEC (0x1F)

Données: Le nom du programme à exécuter en ASCII (8 caractères maximum).

Réponse: ACK.

Description: Exécute le contenu d'un programme préalablement enregistré dans la mémoire de l'afficheur (répertoire).

Exemple: Si par exemple on veut exécuter un programme de nom "MPTEST" présent dans le répertoire de l'afficheur.

Envoyé à l'afficheur: 16 0D 00 01 1F 4D 50 54 45 53 54 20 02

Réponse de l'afficheur: 06 00

Si le programme n'est pas présent dans le répertoire de l'afficheur, la réponse sera: 06 01

⁷Exemple pour un cas particulier. Les valeurs varient selon le produit.

GET NUM PACKET (0x21)

Données: SANS DONNÉES

Réponse: ACK + numéro du dernier PAQUET SEND reçu correctement.

Description: Au lieu du code d'erreur, L'afficheur retourne le numéro du dernier PAQUET SEND reçu correctement. Le compteur est remis à 0 si l'appareil reçoit un paquet **EDIT** ou **EDITFILE** et s'incrémente à valider un paquet **SEND**.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 21 3F 00

Réponse de l'afficheur: 06 XX

Où XX est le numéro du dernier PAQUET SEND reçu correctement.

FASTEXEC (0x27)

Données: Programme qui doit être exécuté par l'afficheur (1000 Bytes Maximum)

Réponse: ACK.

Description: Commande la plus souvent utilisé pour interagir avec l'écran. Le paquet FASTEXEC doit contenir un programme constitué des codes, des effets et du texte à afficher. Voir dans le chapitre OPTIONS DE VISUALISATION les Tokens et Pretokens disponibles.

El paquete FASTEXEC sólo debe usarse cuando el contenido del mismo sea inferior a **1000 bytes** porque en algunos dispositivos el tamaño del buffer de recepción es muy limitado y no sería suficiente para recibirlo sin perder bytes. En caso de suceder esto, se puede bajar la velocidad de transmisión o usar los paquetes **EDITFILE**, **SEND** y **NEXEC**, que son mucho más flexibles y permiten la transmisión de programas de grandes tamaños.

Le paquet FASTEXEC ne doit dépasser les 1000 bytes pour s'adapter à la taille du buffer de réception de certains afficheurs. Pour des paquets supérieure à 1000 bytes on peut réduire la vitesse de transmission ou utiliser les paquets **EDITFILE**, **SEND** et **NEXEC** qui sont beaucoup plus souples et permettre la transmission des programmes de grandes tailles.

Les options d'affichage sont détaillées dans la section relative aux options de visualisation.

Exemple:

Programme qui affiche le texte "MP" en mode Run (Scroll) sur la ligne 1 de l'afficheur.

Envoyé à l'afficheur: 16 10 00 01 27 03 C7 31 2C 31 04 E0 4D 50 27 03

Réponse de l'afficheur: 06 00

PUTPORT (0x28)

Données: 5 Bytes:

Numéro du Port (1 Byte): 1 - n

Action à réaliser (1 Byte): 0: Désactiver; 1: Activer; 2: Exécuter séquence

Nombre de Répétitions de la séquence ON/OFF (1 Byte)

Temps en ON (1 Byte): en 1/4 de secondes

Temps en OFF (1 Byte): en 1/4 de secondes

Réponse: ACK.

Description: Cette commande permet de contrôler les sorties numériques de l'afficheur (sorties relais) de l'écran à travers du protocole de communication.

Le champ *Action à réaliser* (Byte 2) peut prendre les valeurs suivantes:

0: DÉSACTIVER

1: ACTIVER

2: Exécuter Séquence

10: **MODE TEST** - DÉSACTIVER⁸

11: **MODE TEST** - ACTIVER⁹

Exemple:

Activer la sortie n°1 de l'afficheur.

Envoyé à l'afficheur: 16 0C 00 01 28 01 01 00 00 00 4D 00

Réponse de l'afficheur: 06 00

PUTVARS (0x2E)

Description: Modifier la valeur d'une variable Alphanumérique de la mémoire de l'afficheur.

Voir chapitre 2.4 Variables Alphanumériques.

GETVARS (0x2F)

Description: Pétition de la valeur d'une variable Alphanumérique de la mémoire de l'afficheur.

Voir chapitre 2.4 Variables Alphanumériques.

⁸ La sortie se DÉSACTIVE indépendamment de la configuration. Il reste dans cet état jusqu'à la réinitialisation de l'afficheur.

⁹ La sortie s' ACTIVE indépendamment de la configuration. Il reste dans cet état jusqu'à la réinitialisation de l'afficheur.

GETVER EXT (0x30)

Données: SANS DONNÉES

Réponse: ACK + **PAQUET SEND**

Données PAQUET SEND: Structure GETVER EXT (16 Bytes ou plus):

Version du Software du Dispositif (1 Byte): 10 - 254

Version du Hardware (1 Byte): 1 - 255

Nombre de Colonnes (2 Bytes).

NON UTILISÉ (1 Byte).

Nombre de lignes (1 Byte).

Version du FONTS (1 Byte).

Version du MPBASIC (1 Byte).

Version du PROGRAMAS (1 Byte).

... Future Ampliation ...

Description: Pétition d'informations sur le Hardware et Software de l'afficheur qui répond à l'ordinateur (ID 0xFE) avec un paquet SEND. Une différence entre la structure GERVER para rapport à GETVER EXT est que cette dernière est de taille variable et dans l'avenir sera élargi avec d'autres données de hardware et software, par conséquent, le programme qui traite l'information reçue dans le PC ne doit pas s'attendre à un paquet de 16 bytes, mais à une longueur variable.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 30 4E 00

Réponse de l'afficheur¹⁰: 06 00

16 39 00 FE 0C 2E C4 60 00 01 06 1E 14 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 C4 00 2A 43 46 47 20 31 00 00 00 00 00 45
00 00 00 00 00 00 00 00 00 00 00 00 00 00 38 05

Ce qui indique:

- *Version du Software du Dispositif (1 Byte):* **4.6**
- *Version du Hardware (1 Byte):* **196**
- *Nombre de Colonnes (2 Bytes):* **96**
- *NON UTILISÉ (1 Byte).* **1**
- *Nombre de lignes (1 Byte):* **6**
- *Version du FONTS (1 Byte):* **3.0**
- *Version du MPBASIC (1 Byte):* **2.0**
- *Version du PROGRAMAS (1 Byte):* **0**

¹⁰ Exemple pour un cas particulier. Les valeurs varient selon le produit.

TEST PIXELS (0x3C)

Données: SANS DONNÉES

Réponse: ACK.

Description: Test des Leds. Démarre le Test automatique des rangées et des colonnes de Leds. Le test s'exécute jusqu'à ce qu'à l'envoi d'un paquet STOP ou le reset de l'afficheur.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 3C 5A 00

Réponse de l'afficheur: 06 00

GET SETTINGS (0x59)

Données: SANS DONNÉES

Réponse: ACK + **PAQUET SEND**

Données PAQUET SEND: 36 Bytes: Structure de type DEVICE_USER_SETTINGS

Description: L'afficheur retourne une structure avec des paramètres de configuration de l'afficheur.

Voir détails en ANNEXE 3.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 59 77 00

Réponse de l'afficheur: 06 00

16 2B 00 FE 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 0F 05 64
00 1E 40 01 01 01 00 28 00 05 78 00 01 F0 01 32 00 00 F2 03

PUT SETTINGS (0x5A)

Données: 36 Bytes: Structure de type DEVICE_USER_SETTINGS

Réponse: ACK

Description: Envoie à l'afficheur une structure avec des paramètres de configuration.

Voir détails en ANNEXE 3.

Exemple:

Envoyé à l'afficheur: 16 2B 00 01 5A 49 4E 54 38 39 33 32 00 00 00 00 00 00 00 05 0F 05
64 00 1E 40 01 01 01 00 28 00 05 78 00 01 F0 01 32 00 00 04 05

Réponse de l'afficheur: 06 00

Après avoir reçu le ACK (et avec un retard d'environ 2 secondes), l'afficheur se réinitialisera automatiquement pour appliquer les modifications apportées à la configuration.

GET LUM INPUT (0x69)

Données: SANS DONNÉES

Réponse: ACK + **PAQUET SEND**

Données PAQUET SEND: 2 Bytes:

*Lecture de la sonde de luminosité ambiante (1 Byte): 0 à 100 %
NON UTILISÉ (1 Byte).*

Description: Pétition de la lecture de la sonde de luminosité ambiante. L'afficheur répond avec un paquet SEND de 2 bytes : la lecture de la sonde se trouve dans le premier byte(0 - 100).

Exemple:

Envoyé à l'afficheur: 16 07 00 01 69 87 00
Réponse de l'afficheur: 06 00
16 09 00 FE 0C 41 00 6A 01

Ce qui indique que la sonde détecte une luminosité ambiante de 65%.

GET TEMP INT (0x88)

Données: SANS DONNÉES

Réponse: ACK + **PAQUET SEND**

Données PAQUET SEND: 2 Bytes:

*Température interne du Dispositif (1 Byte avec signe): -128 à +127°
NON UTILISÉ (1 Byte).*

Description: Pétition de la température interne du dispositif. L'afficheur répond avec un paquet SEND de 2 bytes : la température se trouve dans le premier byte.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 88 A6 00
Réponse de l'afficheur: 06 00
16 09 00 FE 0C 1B 00 44 01

Ce qui indique que la température interne de l'afficheur est de 27°C.

GET BAT LEVEL (0x96)

Données: SANS DONNÉES

Réponse: ACK + CODE.

Description Code:

0x25: Niveau bas de la batterie.

0x30: Niveau correct de la batterie.

Autre valeur: Commande non supportée par ce modèle

Exemple:

Envoyé à l'afficheur: 16 07 00 01 96 B4 00

Réponse de l'afficheur: 06 25

Ce qui indique que la batterie est déchargée.

FILE SYSTEM RESET (0x9E)

Données: SANS DONNÉES

Réponse: ACK.

Description: Formater le répertoire du dispositif en supprimant tous les fichiers. Cette tâche peut prendre jusqu'à 1 minute en fonction du modèle.

ATTENTION: Une fois réalisé les fichiers ne pourront pas se récupérer (y compris les polices de caractère et les fichiers graphiques). Il devront être chargés de nouveau dans l'afficheur.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 9E BC 00

Réponse de l'afficheur: 06 00

La réponse de l'afficheur peut tarder jusqu'à une minute. Ensuite, l'afficheur se réinitialise.

RESET CONFIG (0xA0)

Données: SANS DONNÉES

Réponse: ACK.

Description: Restaurer la Configuration d'usine.

ATTENTION: Une fois restauré la configuration d'usine les paramètres du client sont perdus , de sorte que l'appareil devra être reconfiguré.

Exemple:

Envoyé à l'afficheur: 16 07 00 01 A0 BE 00

Réponse de l'afficheur: 06 00

Après envoi du ACK (et avec un retard d'environ 2 secondes), l'afficheur se réinitialise pour appliquer les modifications apportées aux paramètres.

SET TEMP SETTINGS (0xC1)

Données: 1 Byte:

Offset Température Extérieure (1 Byte avec signe): -12.0° à +12.0°

Réponse: ACK.

Description: Ajoute un Offset à la Température Extérieure mesurée par l'afficheur (avec signe, en dixième de degré).

Exemple:

On souhaite corriger la lecture de la sonde de température extérieure avec un décalage de -1.5°.

Envoyé à l'afficheur: 16 08 00 01 C1 F1 D1 01

Réponse de l'afficheur: 06 00

N GET TEMP (0xC2)

Données: SANS DONNÉES

Réponse: ACK + **PAQUET SEND**

Données PAQUET SEND: 4 Bytes:

Température externe (2 Bytes avec signe): -20.0 à +120.0°

Offset (1 Byte avec signe): -12.0° à +12.0°

NON UTILISÉ (1 Byte).

Description: Pétition à l'afficheur de la température extérieure + offset. Il retourne un paquet de 3 bytes en utilisant la commande SEND: la température se trouve dans les 2 premiers bytes (la valeur représente des dixièmes de degré avec signe), et l'offset dans le troisième byte (la valeur représente des dixièmes de degré avec signe).

Exemple:

Envoyé à l'afficheur: 16 07 00 01 C2 E0 00

Réponse de l'afficheur: 06 00

16 0B 00 FE 0C 0B 01 F1 00 28 02

Ce qui indique une température extérieure de 26.7° et un offset de -1.5°

2.3. Options de Visualisation.

Pour envoyer et exécuter immédiatement des programmes de visualisation sur l'afficheur, on peut utiliser le paquet FASTEXEC. Cette commande ajoute dans le champ de données un programme également appelé Sript qui est une suite codes+textes...que l'afficheur exécutera dès sa réception. L'afficheur interprète chaque code composé de 2 bytes (1 byte Pretoken et 1 byte Token) et le texte à montrer. Chaque code à un effet direct sur l'affichage.

Chaque programme/Script peut contenir de nombreux codes qui sont exécutés séquentiellement, et peuvent être de 4 types différents: DATA (paramètres généraux d'exécution), MODE (mode d'apparition du texte à l'affichage), TEMPS (variables de temps et de température) et EFFET (ajoutent un effet au texte déjà affiché).

Les codes doivent être placés dans l'ordre DONNÉES, MODES, TEMPS, et EFFET. Le texte à afficher se situe généralement à la fin du script après les codes.

Les programmes doivent se terminer par un byte nul (0x00).

Les codes de caractères vont de 0 à 255 et correspondent à la codification **Windows-1252** (extension de la norme ISO-8859-1) et sont imprimables à partir du caractère espace (0x20).

DATA	DESCRIPTION	<i>n</i>	Pretoken	Token
Page	Nouvelle page de Script/programme		0x03	0x20
Blink	Clignotement du texte qui se trouve entre 2 BLINK		0x03	0xA0
Couleur < n >	Change la couleur du Texte depuis ce point	0 - 7 (Sans, rouge,vert,ambre, bleu,magenta,cian ,blanc)	0x03	0xA1
Graphique < n >	Faire apparaître un graphique mémorisé dans l'afficheur	0 - 99	0x03	0xA4
Variable < n >	Affiche une variable interne est défini sont format de représentation ¹¹	A - Z	0x03	0xAB
Grosueur < n >	Multiplie la grosueur du caractère de la police choisie par <n> colonnes	1 - 8	0x03	0xC0
Police Caratère < n >	Police de caractère (le nombre de polices disponibles varie selon le modèle)	0 - 99	0x03	0xC1
Vltesse Mode < n >	Vitesse de défilement du mode d'apparition (sauf Immédiat)	1 - 99	0x03	0xC4
Attende Mode < n >	Temps d'attente en 1/4 secondes entre ligne (mode indépendant) ou page (mode synchronisé)	4 - 99	0x03	0xC5
Ligne < n , h >	Positionnement du texte sur l'afficheur. Le champ <i>n</i> indique le numéro de ligne est le champ <i>h</i> le nombre de lignes (hauteur) occupées par le texte ¹² .	1 - Lignes Totales	0x03	0xC7
Synchronisme	Exécution en parallèle de plusieurs lignes. Se place au début.		0x03	0xC9
Fin de Synchronisme	Indique la fin de synchronisme entre plusieurs lignes. Se place à la fin d'un groupe de lignes.		0x03	0xCA
Langues < n >	Change la langues des Tokens de Temps		0x03	0xCB
Jour événement < x >	Date de l'événement calcul différences et restes.		0x03	0xCC
Alignement < n >	Alignement du texte dans la ligne ou fenêtre	0: Centre 1 : Gauche 2 : Droite	0x03	0xCD

¹¹ Voir chapitre 2.4

¹² Selon le modèle une ligne vaut un nombre différent de pixels de hauteur.

DATA	DESCRIPTION	<i>n</i>	Pretoken	Token
Windows < n >	Définit une fenêtre d'affichage. Contrairement à la commande Ligne qui occupe toute la largeur de l'afficheur la commande Windows permet définir une zone concrète.	n= <ID window,x1,y1,x2,y2> ID window: de A à N x1: Limite gauche: Numéro de la colonne y1: Limite supérieure Numéro ligne. x2: Limite droite: Numéro de la colonne y2: Limite inférieure Numéro ligne.	0x03	0xD3

MODE	DESCRIPTION	Pretoken	Token
Défiler Gauche	Le texte défile de gauche à droite	0x04	0xD0
Défiler Droite	Le texte défile de droite à gauche	0x04	0xE0
Monter	Le texte Monte	0x04	0xE5
Descendre	Le texte Descend	0x04	0xE6
Immédiat	Le texte apparaît complet	0x04	0xF0

TEMPS	DESCRIPTION	Pretoken	Token
Année	2 chiffres qui indiquent l'année	0x01	0x96
Mois	2 chiffres qui indiquent le mois	0x01	0x97
Nom du Mois	Nom Mois dans langue configurée dans l'afficheur	0x01	0x98
Jours	2 chiffres qui indiquent le jour	0x01	0x99
Nom du Jour	Nom Jour dans langue configurée dans l'afficheur	0x01	0x9A
Heures	2 chiffres qui indiquent l'heure	0x01	0x9B
Minutes	2 chiffres qui indiquent les minutes	0x01	0x9C
Secondes	2 chiffres qui indiquent les secondes	0x01	0x9D
Température	Température ambiante (afficheurs avec sonde)	0x01	0x9F
Différence: Jours	Jours manquants pour date événement	0x01	0xA4
Différence: Semaines	Semaines manquantes pour date événement	0x01	0xA5
Différence: Mois	Mois manquants pour date événement	0x01	0xA6
Heure (HH:MM)	5 caractères pour indiquer Heure et Minutes	0x01	0xA7
Température (°C)	4 caracteres Temperatura y los símbolos '°' y 'C'	0x01	0xA8
Nom Jour Abrégé	Nom jour abrégé langue configurée dans l'afficheur	0x01	0xA9
Nom Mois Abrégé	Nom Mois abrégé dans langue configurée dans l'afficheur	0x01	0xAA
Différence: Minutes	Minutes manquantes pour date événement	0x01	0xAC
Différence: Secondes	Secondes manquantes pour date événement	0x01	0xAD

2.4. Variables Alphanumériques.

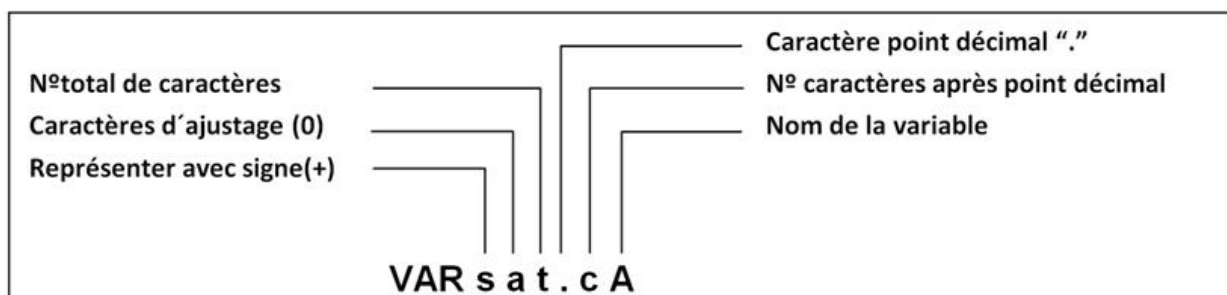
Les afficheurs sauvegardent dans leur mémoire non volatile un ensemble de 26 variables alphanumériques. Chacune de ces variables adopte le format de:

- Une chaîne de caractères *string* de 8 caractères de longueur maximum.
- Un numéro flottant en format IEEE de 64 bits.

Pour lire ou modifier la valeur des variables il faut utiliser les commandes GETVARS et PUTVARS du protocole. Ces commandes sont détaillées dans les pages suivantes.

Pour représenter les variables il faut insérer dans le Script le token **DATO VAR** puis le nom de l'une des 26 variables [**A,Z**] (en majuscules) sans espace entre les deux. Après coupure de l'alimentation, les variables conservent leur valeur tant que dure la batterie interne.

Les variables ont une précision de 16 chiffres, c'est à-dire que l'on peut afficher 16 digits (en ajoutant ceux d'avant et après la virgule) sans perte de précision, dans le cas où soit affiché plus de 16 digits, les chiffres les moins significatifs sont arrondis. Comme le format par défaut d'une variable est composé de 6 chiffres après la virgule, il peut être inconfortable représenter des nombres entiers avec autant de décimales. On peut formater la variable en additionnant le nombre total de chiffres et le nombre de chiffres après la virgule, de la forme suivante:



Exemples de représentation pour la variable A de valeur 1:

VAR6.2A Le résultat sera: __1.00 (Les 2 underscores représentent les espaces)

VAR9.0A _____1

VAR09.0A Si on ajoute un 0 à gauche On remplace les espaces par des zéros : 00000001

VAR+9.0A Si on ajoute un + on représente le signe + : _____+1

VAR-9.0A Avec le signe – on ajuste à gauche: 1_____

VAR9A Ajustage avant le point décimal: _____1

VAR.9A Ajustage après le point décimal : 1.00000000

Les chiffres sont arrondis ; par exemple si la variable B a une valeur interne de 3.141592 il s'affichera 3,1416 si on choisi une représentation á l'affichage de 4 chiffres après la virgule (**VAR.4B**),

Une variable qui a été formaté pour représenter un numéro, peut également représenté un string à travers un paquet PUTVARS.

Par exemple: Le numéro 456 342 à été attribué à la variable Z par l'envoi d'un paquet PUTVARS. **IMMÉDIAT VAR8.0Z** montrera le nombre __456342 (avec deux espaces à gauche totalisant 8 caractères). Si on envoie un autre paquet PUTVARS avec la chaîne de caractères «PARO» à la variable Z, ____PARO est affiché (avec 4 espaces à gauche totalisant 8 caractères) de sorte que à la fois le nombre et la la chaîne sont affichés au même endroit.

RESTRICTIONS

- On ne peut pas représenter des nombres de plus de 16 chiffres.
- Il ne peut pas y avoir plus de 8 caractères de formatage.
- On ne peut pas représenter des *string* de plus de 8 caractères.

CONSEILS

Les variables en format numériques sont de type flottant pour offrir une grande précision, cela implique un traitement plus long de la part de l'afficheur, ce qui peut provoquer un retard au niveau du rafraichissement de l'affichage.

Utiliser de préférence le mode d'apparition IMMÉDIAT et évitez d'utiliser le mode BLINK ou HORMIN (dasn ce cas utiliser des variables en format *string*).

ERREURS

En cas d'erreur, si l'écran ne reconnaît pas la variable à afficher, il apparaît alors à sa place trois tirets "---" (peut également indiquer qu'il y a plus de 8 caractères de formatage).

Commande PUTVARS (0x2E)

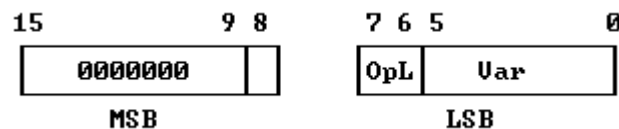
Modifie la valeur des variables dans la mémoire de l'afficheur pour être affiché par la suite avec la commande Data 'VAR'. Le paquet peut être envoyé à n'importe quel moment, que l'afficheur soit en mode exécution ou arrêt et contient 10 bytes. Bien que dans la représentation suivante les bytes sont représentés avec le MSB à gauche ils sont transmis à l'afficheur avec le LSB en premier.

bytes [0,1]

6 bits Numéro de la variable [0,25] (26 variables). 0='A', 1='B',..., 25='Z'

3 bits Type d'opération à réaliser .
0 = Assignment de *string* de caractères
1 = Assignment de valeur en type flottant
2 = Ajouter une valeur (var = var + valeur)
3 = Soustraire une valeur (var = var - valeur)

7 bits Toujours à 0



bytes [2-9]

valeur Nombres à virgule flottante (format IEEE 64 bits) ou chaîne de caractères (string) de 8 caractères maximum. Pour moins de 8 caractères il est recommandé de rajouter un 0 (zéro hexadécimal) à la fin de la trame.

RESTRICTIONS

Il est interdit de mélanger les chiffres et les *strings*. Les cas suivants peuvent provoquer des erreurs d'affichage.

- Ajouter ou soustraire un nombre à un *string*. Un string n'a pas de valeur numérique et si par exemple on augmente de 1, le résultat à l'affichage sera le numéro 1.
- Les strings ne peuvent pas dépasser 8 bytes, mais peuvent être plus court. Bien que pas obligatoire il est recommandé dans ce cas de rajouter un 0 (zéro hexadécimal) à la fin de la trame.
- Envoyer un paquet sans structure.
- Envoyer un paquet avec une variable répétée (plusieurs structures pour la même variable)

TRANSMISSION

Ne pas oublier que le byte de poids faible (LSB) doit toujours être transmis en premier.

À la fin des structures un byte est utilisé exclusivement en cas d'erreur de transmission pour éviter la possibilité que le checksum de 2 paquets consécutifs soit identique ce qui fausserait la séquence de traitements des erreurs de transmission (voir chapitre 2.1.6). Ceci n'est pas problématique en cas d'assignation d'une valeur à une variable mais peut l'être en cas d'opération (addition ou soustraction)

Pour éviter ce cas, le protocole prévoit un byte à la fin de la structure dont la valeur doit être changée à chaque nouvelle transmission. Il est recommandé d'utiliser un compteur qui augmentera sa valeur de 15H à chaque transmission.

EXEMPLE:

Dans l'exemple suivant, les opérations suivantes sont réalisées:

1. Assigner le *string* 'PRODUCTO' à la variable A
2. Additionner 1 à la variable B
3. Assigner le nombre 2145000 à la variable C
4. Assigner le nombre 13406.25 à la variable D
5. Soustraire 1 à la variable E
6. Les autres variables sont mises à 0 (on pourrait également choisir de modifier seulement les variables A à E, car on a la possibilité d'envoyer de 1 à 26 structures,)

```
16 0C 01 01 2E 00 00 50 52 4F 44 55 43 54 4F 81 00 00 00 00 00 00 00 00 F0 3F
42 00 00 00 00 00 74 5D 40 41 43 00 00 00 00 00 20 2F CA 40 C4 00 00 00 00
00 00 00 F0 3F 45 00 00 00 00 00 00 00 00 00 46 00 00 00 00 00 00 00 00
47 00 00 00 00 00 00 00 00 00 48 00 00 00 00 00 00 00 00 00 49 00 00 00
00 00 00 00 00 4A 00 00 00 00 00 00 00 00 00 4B 00 00 00 00 00 00 00 00
4C 00 00 00 00 00 00 00 00 00 4D 00 00 00 00 00 00 00 00 00 4E 00 00 00
00 00 00 00 00 4F 00 00 00 00 00 00 00 00 00 50 00 00 00 00 00 00 00 00
51 00 00 00 00 00 00 00 00 00 52 00 00 00 00 00 00 00 00 00 53 00 00 00
00 00 00 00 00 54 00 00 00 00 00 00 00 00 00 55 00 00 00 00 00 00 00 00
56 00 00 00 00 00 00 00 00 00 57 00 00 00 00 00 00 00 00 00 58 00 00 00
00 00 00 00 00 59 00 00 00 00 00 00 00 00 00 15 25 10
```

En detail:

16	SYN
0C 01	Longueur (0x010C)
01	N° afficheur (ID)
2E	PUTVAR
00 00	Variable A string
50 52 4F 44 55 43 54 4F	'PRODUCTO'
81 00	Variable B addition
00 00 00 00 00 00 F0 3F	1
42 00	Variable C assigner
00 00 00 00 74 5D 40 41	2145000
43 00	Variable D assigner
00 00 00 00 20 2F CA 40	13406.25
C4 00	Variable E soustraire
00 00 00 00 00 00 F0 3F	1
45 00 00 00 00 00 00 00 00 00	Variable F assigner 0
46 00 00 00 00 00 00 00 00 00	Variable G assigner 0
47 00 00 00 00 00 00 00 00 00	Variable H assigner 0
48 00 00 00 00 00 00 00 00 00	Variable I assigner 0
49 00 00 00 00 00 00 00 00 00	Variable J assigner 0
4A 00 00 00 00 00 00 00 00 00	Variable K assigner 0
4B 00 00 00 00 00 00 00 00 00	Variable L assigner 0
4C 00 00 00 00 00 00 00 00 00	Variable M assigner 0
4D 00 00 00 00 00 00 00 00 00	Variable N assigner 0
4E 00 00 00 00 00 00 00 00 00	Variable O assigner 0
4F 00 00 00 00 00 00 00 00 00	Variable P assigner 0
50 00 00 00 00 00 00 00 00 00	Variable Q assigner 0
51 00 00 00 00 00 00 00 00 00	Variable R assigner 0
52 00 00 00 00 00 00 00 00 00	Variable S assigner 0
53 00 00 00 00 00 00 00 00 00	Variable T assigner 0
54 00 00 00 00 00 00 00 00 00	Variable U assigner 0
55 00 00 00 00 00 00 00 00 00	Variable V assigner 0
56 00 00 00 00 00 00 00 00 00	Variable W assigner 0
57 00 00 00 00 00 00 00 00 00	Variable X assigner 0
58 00 00 00 00 00 00 00 00 00	Variable Y assigner 0
59 00 00 00 00 00 00 00 00 00	Variable Z assigner 0
15	Contrôle transmission
25 10	Checksum 0x1025

Orden GETVARS (0x2F)

Requête de la valeur des variables de la mémoire de l'afficheur. La réponse est un PAQUET SEND qui contient 26 structures correspondantes aux variables ordonnées de 'A' à 'Z'. Chaque structure se compose de 10 bytes dans le format suivant:

bytes [0,1]

1 bit Indique si c'est un *string*.

15 bits Reservé

bytes [2-9]

Valeur Nombres à virgule flottante (format IEEE 64 bits) ou chaîne de caractères string) de 8 caractères maximum. Pour moins de 8 caractères la trame peut finir par un 0 (zéro hexadécimal).

Exemple:

Requête à l'afficheur n° 01 : 16 07 00 01 2F 4D 00

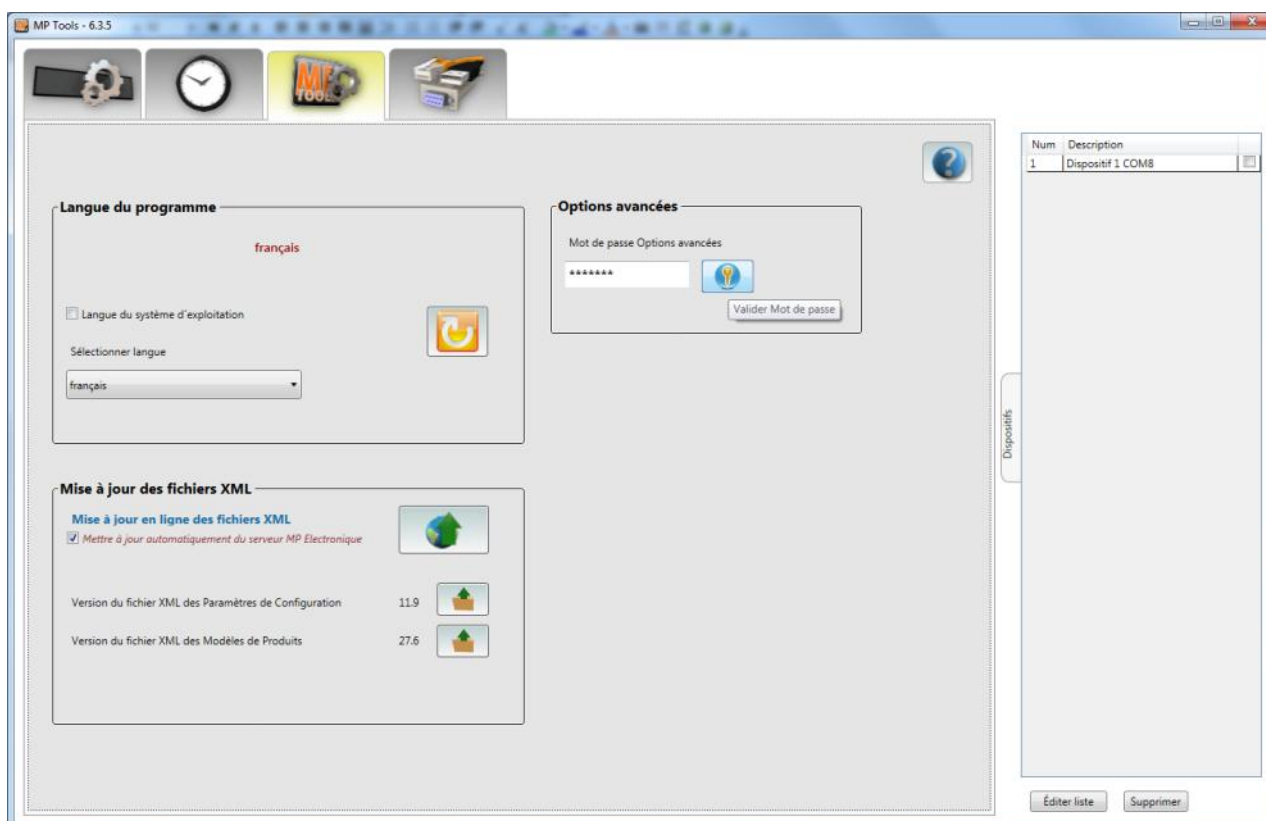
En detail:

16	SYN
00 07	Longueur (0x00 07)
01	N° afficheur (ID)
2F	PUTVAR
00 4D	Checksum 0x4D 00

2.5. Test du protocole DTPM avec le software MP Tools.

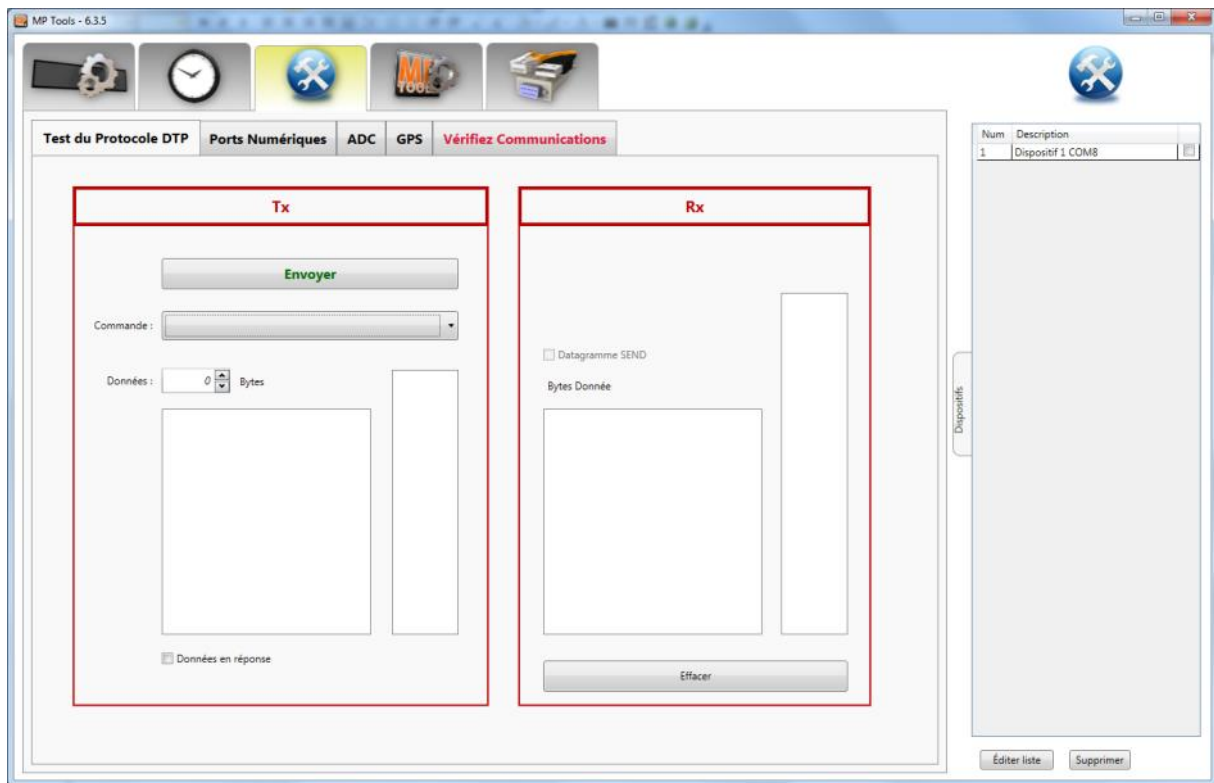
Le logiciel MP Tools, qui peut se télécharger gratuitement sur notre site Web, permet d'envoyer des commandes directement aux afficheurs connectés et d'observer en détail les trames envoyées ainsi que les réponses des afficheurs.

Pour utiliser cette fonction il faut débloquer les options avancées du logiciel. Pour ce faire aller dans l'onglet **Configuration Application** du menu principal et introduire le mot de passe **INT8932** dans le champ **Options avancées** puis valider en appuyant sur la clé.



Le logiciel mémorise la configuration de l'utilisateur, de sorte qu'il n'ai pas nécessaire refaire cette procédure à chaque utilisation.

Une fois les options avancées habilitées, un onglet du même nom apparaît dans le menu principal. Une fois dedans choisir l'onglet Test protocole DTP.



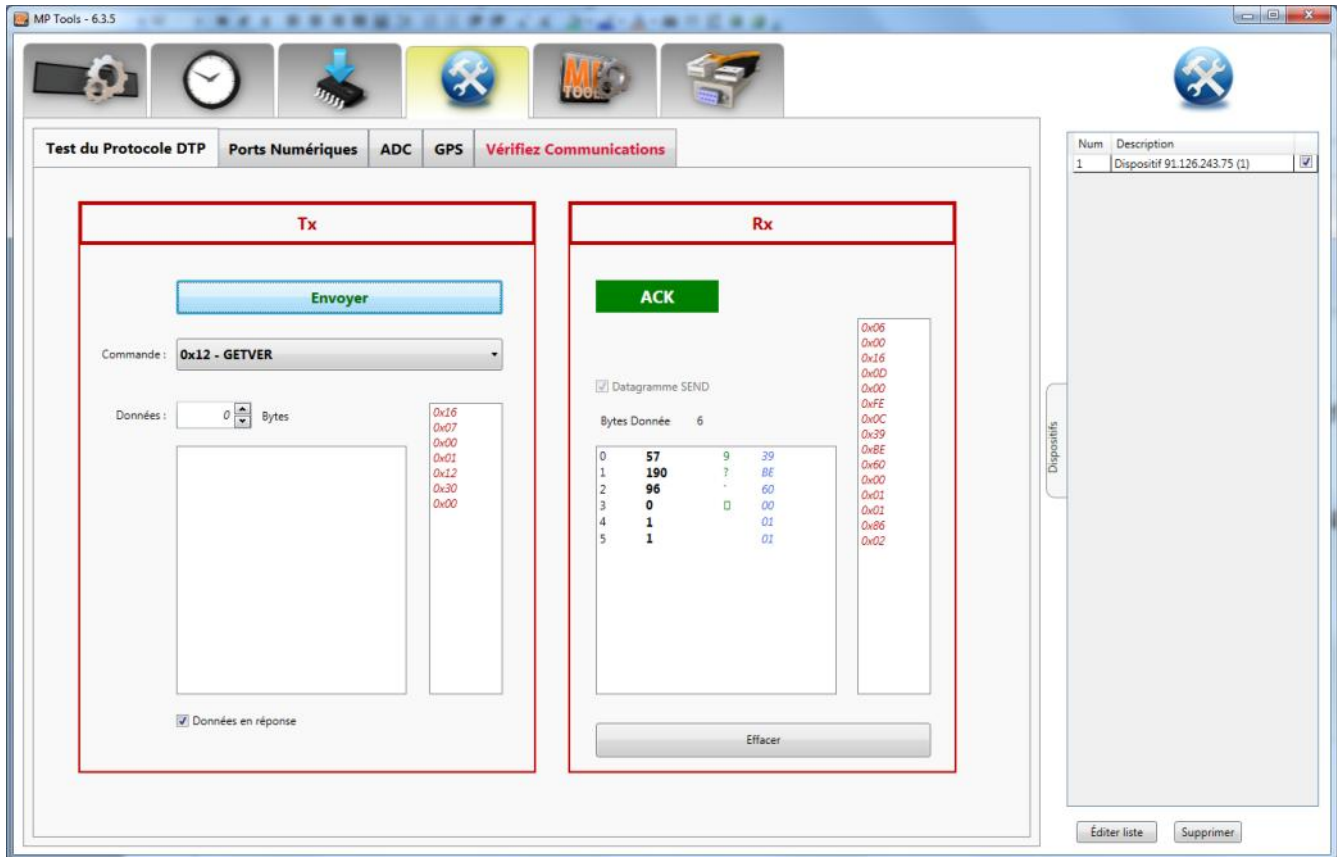
Deux fenêtres sont présentées: **TX** (Transmission) et **RX** (Réception).

Dans la fenêtre **TX** on sélectionne la commande à envoyer.

L'adresse du dispositif, la longueur du paquet ainsi que le checksum sont automatiquement ajoutés par le logiciel.

Selon la commande nécessite l'ajout de données, il faut indiquer le nombre de bytes et la valeur pour chaque byte qui peut s'introduire en format décimal, hexadécimal ou ASCII.

On doit également indiquer si on espère un PAQUET SEND en réponse après réception du ACK.



En appuyant sur le bouton **Envoyer**, le paquet est transmis et la trame complète (avec en-tête, Données et checksum). apparaît à droite dans la fenêtre **TX**.

Si il n'y a pas eu de problèmes de communication la réponse de l'appareil apparaît dans la fenêtre **RX**

L'effacement des trames dans les 2 fenêtre se fait avec le bouton **Effacer**.

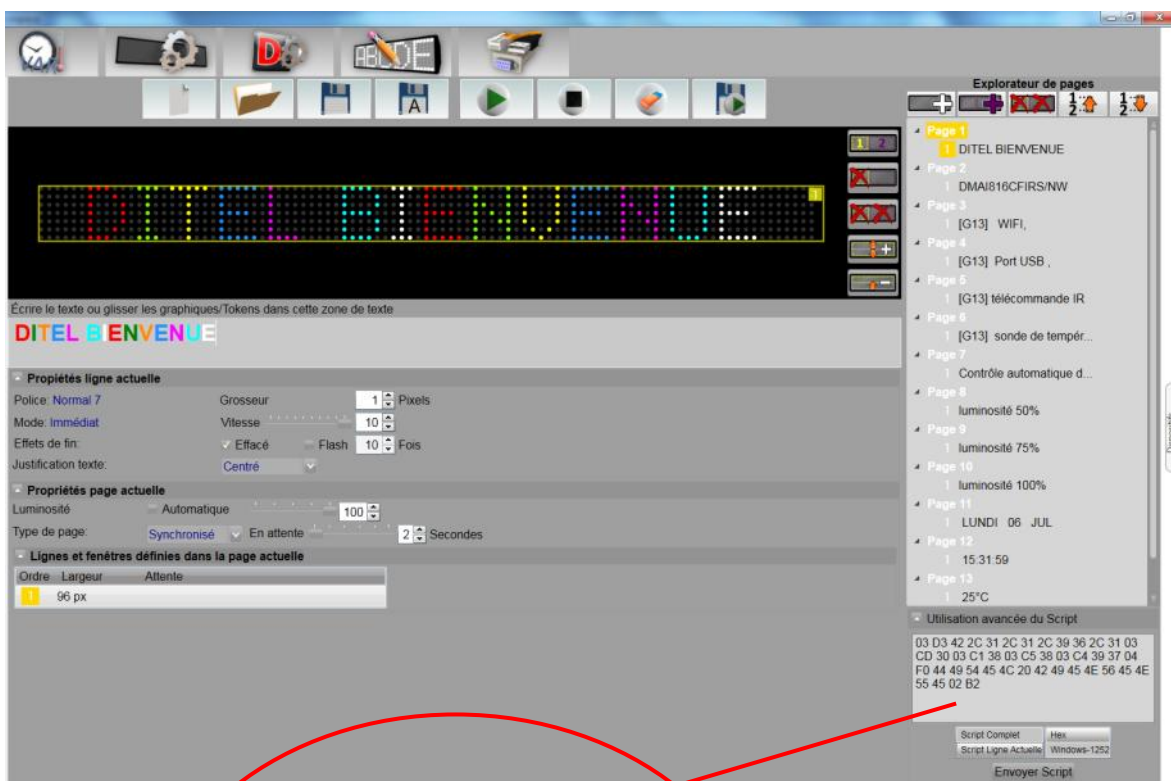
2.6. Test du protocole DTPM avec le software Dynamic3.

Le logiciel Dynamic3, qui peut se télécharger gratuitement sur notre site Web, permet d'éditer des programmes et les exécuter directement sur l'afficheur ou les mémoriser dans l'afficheur pour une utilisation postérieure.

Le Dynamic3 dispose également dans l'onglet **Édition** du menu principal d'un outil appelé **Utilisation avancée du Script** qui permet d'éditer en format Hexadécimal ou ASCII le script correspondant à chaque ligne ou page d'un programme. Ce script peut être modifié, copier, coller et envoyer à l'afficheur. C'est donc un outil très pratique pour la génération de trames et le débogage.

On peut ainsi récupérer tous le code d'un programme pour le réutiliser dans une notre propre application.

Pour utiliser cette fonction il faut débloquer les options avancées du logiciel. Pour ce faire aller dans l'onglet **Configuration Application** du menu principal et introduire le mot de passe **INT8932** dans le champ **Options avancées** puis valider en appuyant sur la clé.



Annexe 1. Configuration par défaut des Afficheurs

La configuration par défaut en sortie d'usine est la suivante:

PARAMÈTRE DE CONFIGURATION	Valeur par défaut
Adresse du dispositif (ID)	1
Adresse <i>Local/Cast</i> du dispositif	0
Port Série	
Port Série : Bauds	9600
Port Série : Data Bits	8
Port Série : Stop Bits	1
Port Série : Paridad	Sans Parité
TCP / IP	
Adresse IP	192.168.1.100
Masque de sous Réseau	255.255.255.0
Port TCP	53
Passerelle (Gateway)	192.168.1.1
Serveur DNS Préféré	192.168.1.100
Serveur DNS Auxiliaire	192.168.1.100
Client DHCP Habilité (Adresse IP Dynamique)	NO

Tableau 4: Configuration par défaut des afficheurs

Annexe 2. Codes d'erreur du Protocole DTPM

Code		Description
0	0x00	OK (opération effectuée correctement)
1	0x01	Programme non trouvé
2	0x02	Impossible éditer (un programme est actuellement en exécution)
3	0x03	Fichier non trouvé
5	0x05	Le programme sélectionné est actuellement en exécution
6	0x06	Opération non autorisée en mode exécution
7	0x07	Commande inconnue
8	0x08	Fichier sans contenu (ne peut pas s'exécuter)
9	0x09	Password Incorrect
10	0x0A	Nom trop long
11	0x0B	Format Heure/Date incorrect
12	0x0C	Format EDITFILE incorrect
13	0x0D	Fichier supérieur à l'espace disponible dans le répertoire
14	0x0E	Fichier supérieur à la taille maximale
25	0x19	Données non valides
28	0x1C	Dispositif occupé
49	0x31	Port Digital non disponible
62	0x3E	Impossible de créer le MBR (Système de Fichier)
63	0x3F	Impossible de formater le répertoire
64	0x40	Répertoire vide
65	0x41	Système de fichier non habilité
66	0x42	Impossible effacer le fichier
67	0x43	Impossible initier (Effacer) le système de fichier
68	0x44	Longueur sollicité trop grande
69	0x45	Erreur système de fichier (Général)
70	0x46	Taille de Fichier supérieure à l'espace réservé
73	0x49	Fichier corrompu / endommagé
74	0x4A	FASTEXEC non disponible

Tableau 5: Codes d'erreur

Annexe 3. Structure de Données *DEVICE_USER_SETTINGS*

Structure de données propre aux commandes GET_SETTINGS et PUT_SETTINGS.

BYTE	DESCRIPTION	VALEURS POSSIBLES
1	<p>PASSWORD (Mot de passe)</p> <p>Durant la réception des données de l'afficheur avec la commande GET_SETTINGS, ce champ a tous les Bytes à 0.</p> <p>Lors de l'envoi de nouveaux paramètres avec la commande PUT_SETTINGS ce champ doit prendre la valeur "INT8932" en format ASCII avec les bytes restants à NULL (0x00). En cas contraire l'afficheur retournera le code de Password Incorrect (0x09).</p>	'I' (0x49)
2		'N' (0x4E)
3		'T' (0x54)
4		'8' (0x38)
5		'9' (0x39)
6		'3' (0x33)
7		'2' (0x32)
8		0x00
9		0x00
10		0x00
11		0x00
12	Langue des textes d'aide.	0 - 10
13	Reset POWER-ON (Démarrage en mode arrêt)	0 - 1
14	Effacer après STOP (effacement affichage avant nouveau message ou superposition)	0: Effacer ; 1: Ne pas effacer
15	Temps d'attente par défaut entre chaque ligne ou page.	1 - 200 (quarts de seconde)
16	Vitesse mode d'apparition (sauf "Immédiat")	1 - 100 (pixels par seconde)
17	En cas de luminosité automatique force à un niveau minimum (%) de luminosité indépendamment de la lecture de la sonde.	1 - 100 %
18	En cas de luminosité automatique définit un niveau de luminosité ambiante % à partir duquel l'intensité des LEDs sera maximale.	1 - 100 %
19	Mode luminosité (Automatique ou Manuel) . En mode manuel indique le % de luminosité par défaut.	0 pour AUTOMATIQUE 1 à 100 % pour MANUEL
20	Filtre de la Sonda de Luminosité	0 - 255 secondes
21	RTCC: Heure d'été	0 pour DÉACTIVER 64 pour ACTIVER
22	RTCC: Type d'horaire d'été	1 : EUROPÉEN
23	RTCC: Fuseau Horaire	-12 à +14 heures UTC
24	Indication niveau batterie	0 (bas) - 1 (OK)
25	RTCC: SoftTrimm (ajustage fin Horloge interne)	-7 à +8 ppm
26	Température d'activation de la ventilation interne	-120°C à +120°C
27	Mode anti condensation	0 - 1
28	Temps d'activation de la ventilation interne (Mode anti condensation)	0 - 255 minutes
29	Période ventilation interne (Mode anti condensation)	0 - 1440 minutes
30		
31	Largeur de caractère (variable ou fixe)	0 (Variable) - 1 (fixe)
32	Mode d'apparition des textes	240 Immédiat; 224 Défiler
33	Contrôle illumination Supérieure par sonde luminosité ambiante	0 (Désactivé) - 1 (Activé)
34	Niveau activation illumination supérieure (s'active en dessous)	5 - 85 % (luminosité mesurée)
35	(Réservé)	
36	(Réservé)	

Structure *DEVICE_USER_SETTINGS*

Pour utiliser les commandes GET_SETTINGS et PUT_SETTINGS il faut suivre la procédure suivante:

1. Envoyer à l'afficheur la commande GET_SETTINGS. L'afficheur retourne une structure avec les paramètres de configuration de l'afficheur.
2. Modifier dans la structures les champs souhaités, sans modifier les autres.
3. Introduire dans les 11 premiers bytes le password en format ASCII : 'I' 'N' 'T' '8' '9' '3' '2' 'NULL' 'NULL' 'NULL' 'NULL' .
4. Envoyer à l'afficheur la commande PUT_SETTINGS la structure modifiée.
5. Après avoir reçu le ACK si l'opération à réussie, attendre environ deux secondes pour que l'afficheur redémarre et les modifications apportées aux paramètres soient appliquées.

RÉVISIONS

Révision 1.0 – Document initial.

Révision 1.1 – Variables Alphanumériques

Révision 1.2 – Commandes *STOP AND CLEAR*.

Révision 1.3 – Modification commande PUTPORT.

Révision 1.4 – Commandes GET_SETTINGS et PUT_SETTINGS.