



COMMUNICATION AVEC LES AFFICHEURS DM PAR PROTOCOLE MODBUS

INDICE

1. [DESCRIPTION](#)

1.1. [Protocole MODBUS-RTU](#)

1.2 [Protocole MODBUS-TCP/IP](#)

2. [MODBUS MAP ADRESSES des afficheurs DM](#)

3. [OPTIONS D'EXPLOITATION EN COMMUNICATION ModBus](#)

3.1. [OPTION 1: Activation d'un programme préenregistré dans l'afficheur](#)

3.1.1. [Activation d'un programme par son numéro](#)

3.1.2. [Activation d'un programme par son nom](#)

3.2. [OPTION 2: Envoi et exécution du Script d'un programme](#)

3.2.1 [Exemples](#)

3.3. [OPTION 3: Utilisation des variables internes de l'afficheur](#)

3.3.1. [Type de variables](#)

3.3.2. [Adresses des Variables Internes](#)

3.3.3. [Exemples](#)

Annexe 1. [Paramètres par défaut des afficheurs DM](#)

Annexe 2. [Script DTPM](#) (Uniquement pour Option 2: Envoi et exécution du Script d'un programme)

Introduction

Liste et description des codes DTPM pour l'édition de programme

Structure de la trame d'un programme

Exemples de Script

Génération et test de Script DTPM avec le logiciel Dynamic3

Annexe 3. [Représentation des Variables](#) (Uniquement pour Option 2: Envoi et exécution de Script)

Exemples de représentation de Variable

Exemple de trames

1. DESCRIPTION

Les afficheurs DITEL de la série DINOS-DM, supportent le protocole ModBus sous sa forme ModBus-RTU et ModBus-TCP/IP. Ce protocole est largement utilisé dans les environnements industriels et facilement adaptable à de nombreux types d'instruments, tels que les automates programmables (PLC).

1.1. Protocole MODBUS-RTU

Le protocole Modbus RTU utilise les silences dans la ligne de transmission pour indiquer le début et la fin du message. Un silence est considéré comme égal ou supérieur au temps nécessaire pour la transmission de 3,5 octets. À chaque vitesse de transmission correspond un temps de silence précis. Après l'envoi d'un message, on ne peut pas commencer une nouvelle transmission jusqu'à ce que le temps nécessaire (3,5 fois du temps de transmission d'un octet) se soit écoulé.

Dans ce protocole, **l'afficheur fonctionne en mode esclave**. Après avoir reçu un message à son adresse, l'afficheur renvoie un message avec le résultat de la transmission.

1.2 Protocole MODBUS-TCP/IP

Le Protocole ModBus sur TCP/IP (ModBus-TCP dans ce manuel) est une variante du protocole Modbus pour les communications sur les réseaux TCP / IP. Les connexions se font via le **port TCP 502**.

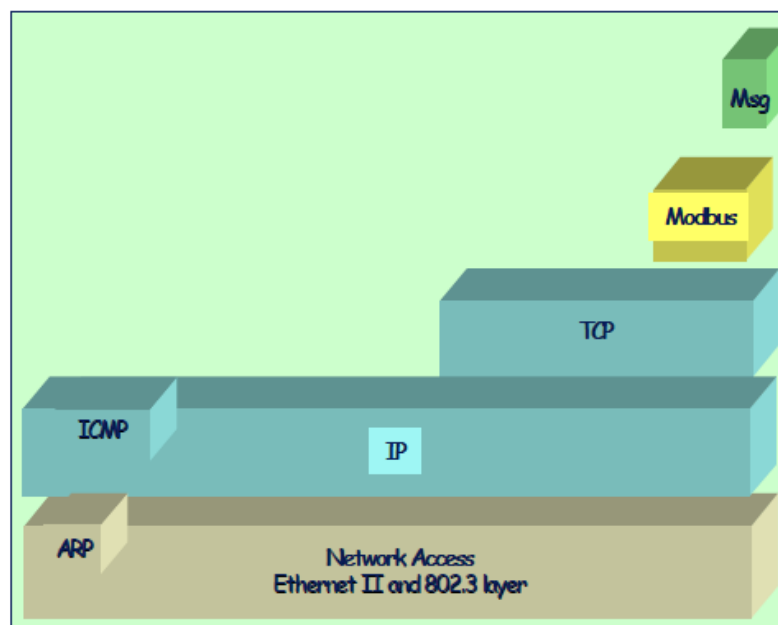
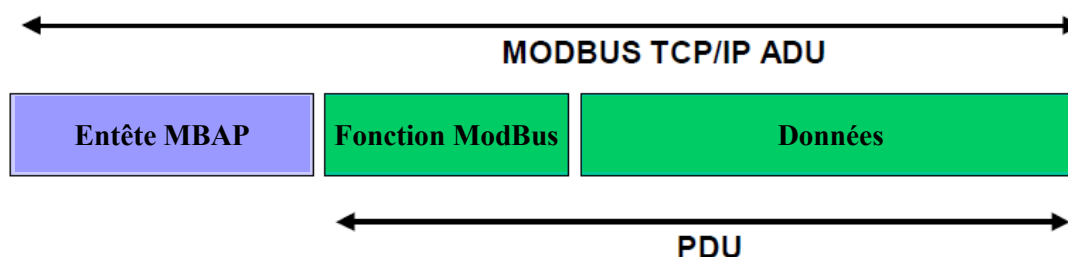


Schéma des couches de communication pour les communications ModBus-TCP

Dans ce protocole, **l'afficheur fonctionne en mode esclave**. (TCP Server). Les trames sont égales à celles du protocole Modbus RTU avec les différences suivantes:

- Le champ ID (adresse dispositif) de la trame ModBus-RTU est remplacé par un entête appelé MBAP Header, les champs de celui-ci sont détaillés dans le Tableau suivant.
- Il n'y a pas de code de vérification d'erreur CRC vu que les protocoles des couches inférieures s'occupent de cette tâche.

La structure de la trame est la suivante:



Trame du protocole ModBus-TCP

L'entête MBAP (7 bytes) contient les champs suivants:

Champ	Bytes	Description	Client (Master)	Server (Slave)
Transaction Identifier	2	Numéro de Transaction	Initialisée par le Client	Renvoyé par le Server
Protocol Identifier	2	0 = MODBUS protocol	Initialisée par le Client	Renvoyé par le Server
Length	2	Nombre de Bytes de la trame qui suit (de "Unit Identifier" jusqu'à la fin)	Initialisée par le Client	Renvoyé par le Server
Unit ID	1	Toujours 255 ou l'ID de l'afficheur	Initialisée par le Client	Renvoyé par le Server

Entête MBAP du Protocole ModBus-TCP

Après avoir reçu un message à son adresse, l'afficheur renvoie un message avec le résultat de la transmission.

2. MODBUS MAP ADRESSES des afficheurs DM

Les afficheurs série DINOS-DM supportent la **fonction ModBus 16 (10^H) "Ecrire n registres"**. Le mode de fonctionnement ainsi que les variables internes sont déterminés par les adresses Modbus des registres où les données sont envoyées.

MODBUS MAP ADRESSES DES AFFICHEURS DM. (Uniquement fonction ModBus 16 (10 ^H) "Ecrire n registres")													
Nom	Adresse Interne	Registre	Description										Chapitre (page)
Non Programme	0080 ^H	40128 MW128	Activer Programme.par Nom										3.1.1 (p7)
Script	0100 ^H	40256 MW256	Envoyer Script										3.2 (p8)
N° Programme	0200 ^H	40512 MW512	Activer Programme. par Numéro										3.1.2 (p7)
Type Variable	0202 ^H	40514 MW514	Format des Variables										3.3.1 (p11)
VARIABLES	Value/Mot 1			Value/Mot 2			Value/Mot 3			Value/Mot 4			
Nom	Adresse Interne	Registre	Description	Adresse Interne	Registre	Description	Adresse Interne	Registre	Description	Adresse Interne	Registre	Description	
Variable A	0204 ^H	40516 MW516	Int16/LSB Int32 ASCII carac. 1,2	0205 ^H	40517 MW517	MSB Int32 ASCII carac. 3,4	0206 ^H	40518 MW518	Decimal point ASCII carac. 5,6	0207 ^H	40519 MW519	Color ASCII carac. 7,8	3.3.2 (p12)
Variable B	0208 ^H	40520 MW520	Int16/LSB Int32 ASCII carac. 1,2	0209 ^H	40521 MW521	MSB Int32 ASCII carac. 3,4	020A ^H	40522 MW522	Decimal point ASCII carac. 5,6	020B ^H	40523 MW523	Color ASCII carac. 7,8	3.3.2 (p12)
Variable C	020C ^H	40524 MW524	-	020D ^H	40525 MW525	-	020E ^H	40526 MW526	-	020F ^H	40527 MW527	-	-
Variable D	0210 ^H	40528 MW528	-	0211 ^H	40529 MW529	-	0212 ^H	40530 MW530	-	0213 ^H	40531 MW531	-	-
Variable E	0214 ^H	40532 MW532	-	0215 ^H	40533 MW533	-	0216 ^H	40534 MW534	-	0217 ^H	40535 MW535	-	-
Variable F	0218 ^H	40536 MW536	-	0219 ^H	40537 MW537	-	021A ^H	40538 MW538	-	021B ^H	40539 MW539	-	-
Variable G	021C ^H	40540 MW540	-	021D ^H	40541 MW541	-	021E ^H	40542 MW542	-	021F ^H	40543 MW543	-	-
Variable H	0220 ^H	40544 MW544	-	0221 ^H	40545 MW545	-	0222 ^H	40546 MW546	-	0223 ^H	40547 MW547	-	-
Variable I	0224 ^H	40548 MW548	-	0225 ^H	40549 MW549	-	0226 ^H	40550 MW550	-	0227 ^H	40551 MW551	-	-

^H → Numération hexadécimale.

Variable J	0228 ^H	40552 MW552	Int16/LSB Int32 ASCII carac. 1,2	0229 ^H	40553 MW553	MSB Int32 ASCII carac. 3,4	022A ^H	40554 MW554	Decimal point ASCII carac. 5,6	022B ^H	40555 MW555	Color ASCII carac. 7,8	3.3.2 (p12)
Variable K	022C ^H	40556 MW556	Int16/LSB Int32 ASCII carac. 1,2	022D ^H	40557 MW557	MSB Int32 ASCII carac. 3,4	022E ^H	40558 MW558	Decimal point ASCII carac. 5,6	022F ^H	40559 MW559	Color ASCII carac. 7,8	3.3.2 (p12)
Variable L	0230 ^H	40560 MW560	-	0231 ^H	40561 MW561	-	0232 ^H	40562 MW562	-	0233 ^H	40563 MW563	-	3.3.2 (p12)
Variable M	0234 ^H	40564 MW564	-	0235 ^H	40565 MW565	-	0236 ^H	40566 MW566	-	0237 ^H	40567 MW567	-	3.3.2 (p12)
Variable N	0238 ^H	40568 MW568	-	0239 ^H	40569 MW569	-	023A ^H	40570 MW570	-	023B ^H	40571 MW571	-	3.3.2 (p12)
Variable O	023C ^H	40572 MW572	-	023D ^H	40573 MW573	-	023E ^H	40574 MW574	-	023F ^H	40575 MW575	-	3.3.2 (p12)
Variable P	0240 ^H	40576 MW576	-	0241 ^H	40577 MW577	-	0242 ^H	40578 MW578	-	0243 ^H	40579 MW579	-	3.3.2 (p12)
Variable Q	0244 ^H	40580 MW580	-	0245 ^H	40581 MW581	-	0246 ^H	40582 MW582	-	0247 ^H	40583 MW583	-	3.3.2 (p12)
Variable R	0248 ^H	40584 MW584	-	0249 ^H	40585 MW585	-	024A ^H	40586 MW586	-	024B ^H	40587 MW587	-	3.3.2 (p12)
Variable S	024C ^H	40588 MW588	-	024D ^H	40589 MW589	-	024E ^H	40590 MW590	-	024F ^H	40591 MW591	-	3.3.2 (p12)
Variable T	0250 ^H	40592 MW592	-	0251 ^H	40593 MW593	-	0252 ^H	40594 MW594	-	0253 ^H	40595 MW595	-	3.3.2 (p12)
Variable U	0254 ^H	40596 MW596	-	0255 ^H	40597 MW597	-	0256 ^H	40598 MW598	-	0257 ^H	40599 MW599	-	3.3.2 (p12)
Variable V	0258 ^H	40600 MW600	-	0259 ^H	40601 MW601	-	025A ^H	40602 MW602	-	025B ^H	40603 MW603	-	3.3.2 (p12)
Variable W	025C ^H	40604 MW604	-	025D ^H	40605 MW605	-	025E ^H	40606 MW606	-	025F ^H	40607 MW607	-	3.3.2 (p12)
Variable X	0260 ^H	40608 MW608	-	0261 ^H	40609 MW609	-	0262 ^H	40610 MW610	-	0263 ^H	40611 MW611	-	3.3.2 (p12)
Variable Y	0264 ^H	40612 MW612	-	0265 ^H	40613 MW613	-	0266 ^H	40614 MW614	-	0267 ^H	40615 MW615	-	3.3.2 (p12)
Variable Z	0268 ^H	40616 MW616	-	0269 ^H	40617 MW617	-	026A ^H	40618 MW618	-	026B ^H	40619 MW619	-	3.3.2 (p12)

^H → Numération hexadécimale.

3. OPTIONS D'EXPLOITATION EN COMMUNICATION ModBus

3.1. OPTION 1: Activation d'un programme préenregistré dans l'afficheur

Le logiciel **Dynamic3**, qui peut se télécharger gratuitement sur notre site Web, permet d'éditer des programmes avec ou sans variables et les mémoriser dans l'afficheur pour un contrôle via communication.

La commande pour activer un programme préenregistré dans l'afficheur s'effectue depuis deux adresses de registre différentes selon si on utilise le numéro du programme ou son nom.

3.1.1. Activation d'un programme par son numéro

Dans ce mode, les programmes doivent avoir été enregistrés dans l'afficheur sous le nom **PRGM** suivi du numéro (sans zéros à gauche). Programme 1 = "PRGM1" Programme 3 = "PRGM3"...Programme 999="PRGM999".

IMPORTANT: Le "PRGM0" réalise un STOP de l'afficheur (avec ou sans effacement selon configuration interne) et ne peut pas être utilisé pour afficher un programme.

Adresse de départ	0200 ^H (0x200)
Nombre de registres (Nbr. de Mots)	1
Données	Numéro du programme (0 à 999) (STOP avec "PRGM0")

Par exemple, pour exécuter le programme "PRGM1" préenregistrée dans l'afficheur, le Modbus hexadecimal PDU à envoyer est:

Fonc.	Start Adress		Num.of words		Num. Bytes	Data Byte 1	Data Byte 2
10	02	00	00	01	02	00	01

Examen en détail du bloque de données:

Data Byte 1	Data Byte 2
00	01
'0'	'1'

3.1.2. Activation d'un programme par son nom

Dans ce mode, le bloc de données contient le nom du programme à exécuter au **format ASCII avec de 3 à 8 caractères** et une valeur NULL qui marque la fin.

Adresse de départ	0080 ^H (0x80)
Nombre de registres (Nbr. de Mots)	1 à 4
Données	Nom du programme en format ASCII + <u>un caractère NULL (0x00) pour indiquer la fin du bloc.</u> ATTENTION: Le nom du programme doit avoir de 3 à 8 caractères maximum.

^H, 0x → Numération hexadécimale.

Par exemple, pour exécuter le programme "MPTEST" préenregistrée dans l'afficheur, le Modbus hexadecimal PDU à envoyer est:

Fonc.	Start Adress		Num.of words		Num. Bytes	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8
10	00	80	00	04	08	4D	50	54	45	53	54	00	00

Examen en détail du bloque de données:

Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8
4D	50	54	45	53	54	00	00
'M'	'P'	'T'	'E'	'S'	'T'	NULL	NULL

3.2. OPTION 2: Envoi et exécution du Script d'un programme

Cette option permet d'envoyer le **Script** d'un programme pour son exécution immédiate sur l'afficheur. **Voir Annexe 2 Script DTPM.**

Adresse de départ	0100 ^H (0x100)
Nombre de registres (Nbr. de Mots)	1 à 124 (Modbus Standard)
Données	Script du programme + <u>un caractère NULL (0x00) pour indiquer la fin du bloc</u>

Envoi et exécution du Script d'un programme

3.2.1. Exemples.

Le Modbus hexadecimal PDU envoyé est détaillé pour chaque exemple suivant:

- Exemple 1: Afficher "Hola" en mode immédiat :

Fonc.	Start Adress		Num.of words		Num. Bytes	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8
10	01	00	00	04	08	04	F0	48	6F	6C	61	00	00

^H, 0x → Numération hexadécimale.

Examen en détail du bloque de données:

Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7
04	F0	48	6F	6C	61	00
Pretoken	<i>Immediate Mode</i>	'H'	'o'	'l'	'a'	NULL

- Exemple 2: Afficher "V:" suivit de la variable interne A.

Fonc.	Start Adress		Num.of words		Num. Bytes	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8
10	01	00	00	04	08	04	F0	56	3A	20	AB	41	00

Examen en détail du bloque de données:

Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8	Data Byte 9
04	F0	56	3A	20	AB	41	00	00
Pretoken	<i>Immediate Mode</i>	'V'	':'	''	VAR	'A'	NULL	NULL

- Exemple 3: Afficher "Hola a todos" en mode immédiat et clignotement du caractère "a":

Fonc.	Start Adress		Num.of words		Num. Bytes	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8
10	00	20	00	0A	14	04	F0	48	6F	6C	61	20	03
Data Byte 9	Data Byte 10	Data Byte 11	Data Byte 12	Data Byte 13	Data Byte 14	Data Byte 15	Data Byte 16	Data Byte 17	Data Byte 18	Data Byte 19	Data Byte 20		
A0	61	03	A0	20	74	6F	64	6F	73	00	00		

Examen en détail du bloque de données:

Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8	Data Byte 9	Data Byte 10
03	F0	48	6F	6C	61	20	03	A0	61
Pre-token	Inmediate Mode	'H'	'o'	'l'	'a'	''	Pre-token	<i>Blink</i>	'a'

Data Byte 11	Data Byte 11	Data Byte 12	Data Byte 13	Data Byte 14	Data Byte 15	Data Byte 16	Data Byte 17	Data Byte 18	Data Byte 19
03	A0	20	74	6F	64	6F	73	00	00
Pre-token	<i>Blink</i>	''	't'	'o'	'd'	'o'	's'	<i>NULL</i>	<i>NULL</i>

3.3. OPTION 3: Utilisation des variables internes de l'afficheur

3.3.1. Type de variables

Tous les afficheurs disposent de 26 Variables internes qui peuvent représenter des nombres entiers signés ou non signés mais aussi des valeurs alphanumériques de jusqu'à 8 caractères.

Le type de Variable (numériques ou alphanumérique) se configure dans le registre 202^H et est commun à toutes les variables.

Les variables sont identifiées de la lettre A à Z. et numérotées de 0 à 25 (A = 0, B = 1, ... Z = 25). Les variables sont initialisées dès que l'afficheur est mis sous tension. En cas de déconnexion, les variables sont conservées grâce à la batterie interne.

Les programmes préenregistrés dans l'afficheur peuvent inclure des variables. Quand un programme est activé (80^H) les variables apparaissent avec leur dernière valeur.

Il est aussi possible envoyer le **Script** d'un programme (100^H) avec des variables en utilisant le code 0x03 + 0xAB + nom de variable(A-Z) + 1F. (Voir Annexe 2 et 3)

Valeurs numériques:

3 mots sont utilisés pour les valeurs numériques. 2 mots pour la valeur, plus 1 mot pour la position du point décimal.

Cela signifie que nous pouvons envoyer des valeurs signés ou non signés de 16 ou 32 bits. Le registre 0202^H détermine comment seront interprétées les valeurs numériques.

Register 0202h	Type	Low Range	High Range
0x00	16 bits signed Integer	-32768	+32767
0x01	16 bits unsigned Integer	0	65535
0x02	32 bits signed Integer	-2147483647	+2147483647
0x03	32 unsigned Integer	0	4294967295
0x04	ASCII (Alphanumeric Value)	8 characters max length	

Register 0202^H : Type de Variable

Valeurs Alphanumériques:

Les valeurs alphanumériques sont également disponibles. Le format doit être en code ASCII (**registre 202^H = 0x04**) et la longueur est de **6 caractères maximum** si on utilise le 4ème mot pour le code couleur (voir Map Adresses) ou de **8 caractères maximum** si on utilise le 4ème mot pour les caractères. Pour la valeur alphanumérique le format de représentation affichage pas d'importance.

^H, 0x → Numération hexadécimale.

3.3.2. Adresses des Variables Internes

Le tableau suivant indique les adresses des registres pour les 26 variables.

VARIABLE		ADRESSES											
		Value/Mot 1			Value/Mot 2			Value/Mot 3			Value/Mot 4		
Nom	Nº	Adresse Interne	Registre	Description	Adresse Interne	Registre	Description	Adresse Interne	Registre	Description	Adresse Interne	Registre	Description
A	0	0204 ^H	40516 MW516	Int16/LSB Int32 ASCII carac. 1,2	0205 ^H	40517 MW517	MSB Int32 ASCII carac. 3,4	0206 ^H	40518 MW518	Decimal point ASCII carac. 5,6	0207 ^H	40519 MW519	Color ASCII carac. 7,8
B	1	0208 ^H	40520 MW520	Int16/LSB Int32 ASCII carac. 1,2	0209 ^H	40521 MW521	MSB Int32 ASCII carac. 3,4	020A ^H	40522 MW522	Decimal point ASCII carac. 5,6	020B ^H	40523 MW523	Color ASCII carac. 7,8
C	2	020C ^H	40524 MW524	-	020D ^H	40525 MW525	-	020E ^H	40526 MW526	-	020F ^H	40527 MW527	-
D	3	0210 ^H	40528 MW528	-	0211 ^H	40529 MW529	-	0212 ^H	40530 MW530	-	0213 ^H	40531 MW531	-
E	4	0214 ^H	40532 MW532	-	0215 ^H	40533 MW533	-	0216 ^H	40534 MW534	-	0217 ^H	40535 MW535	-
F	5	0218 ^H	40536 MW536	-	0219 ^H	40537 MW537	-	021A ^H	40538 MW538	-	021B ^H	40539 MW539	-
G	6	021C ^H	40540 MW540	--	021D ^H	40541 MW541	-	021E ^H	40542 MW542	-	021F ^H	40543 MW543	-
H	7	0220 ^H	40544 MW544	-	0221 ^H	40545 MW545	-	0222 ^H	40546 MW546	-	0223 ^H	40547 MW547	-
I	8	0224 ^H	40548 MW548	-	0225 ^H	40549 MW549	-	0226 ^H	40550 MW550	-	0227 ^H	40551 MW551	-
J	9	0228 ^H	40552 MW552	-	0229 ^H	40553 MW553	-	022A ^H	40554 MW554	-	022B ^H	40555 MW555	-
K	10	022C ^H	40556 MW556	-	022D ^H	40557 MW557	-	022E ^H	40558 MW558	-	022F ^H	40559 MW559	-
L	11	0230 ^H	40560 MW560	-	0231 ^H	40561 MW561	-	0232 ^H	40562 MW562	-	0233 ^H	40563 MW563	-
M	12	0234 ^H	40564 MW564	-	0235 ^H	40565 MW565	-	0236 ^H	40566 MW566	-	0237 ^H	40567 MW567	-
N	13	0238 ^H	40568 MW568	-	0239 ^H	40569 MW569	-	023A ^H	40570 MW570	-	023B ^H	40571 MW571	-
O	14	023C ^H	40572 MW572	-	023D ^H	40573 MW573	-	023E ^H	40574 MW574	-	023F ^H	40575 MW575	-

Q	16	0244 ^H	40580 MW580	Int16/LSB Int32 ASCII carac. 1,2	0245 ^H	40581 MW581	MSB Int32 ASCII carac. 3,4	0246 ^H	40582 MW582	Point Décimal ASCII carac. 5,6	0247 ^H	40583 MW583	Color ASCII carac. 7,8
R	17	0248 ^H	40584 MW584	Int16/LSB Int32 ASCII carac. 1,2	0249 ^H	40585 MW585	MSB Int32 ASCII carac. 3,4	024A ^H	40586 MW586	Point Décimal ASCII carac. 5,6	024B ^H	40587 MW587	Color ASCII carac. 7,8
S	18	024C ^H	40588 MW588	-	024D ^H	40589 MW589	-	024E ^H	40590 MW590	-	024F ^H	40591 MW591	-
T	19	0250 ^H	40592 MW592	-	0251 ^H	40593 MW593	-	0252 ^H	40594 MW594	-	0253 ^H	40595 MW595	-
U	20	0254 ^H	40596 MW596	-	0255 ^H	40597 MW597	-	0256 ^H	40598 MW598	-	0257 ^H	40599 MW599	-
V	21	0258 ^H	40600 MW600	-	0259 ^H	40601 MW601	-	025A ^H	40602 MW602	-	025B ^H	40603 MW603	-
W	22	025C ^H	40604 MW604	-	025D ^H	40605 MW605	-	025E ^H	40606 MW606	-	025F ^H	40607 MW607	-
X	23	0260 ^H	40608 MW608	-	0261 ^H	40609 MW609	-	0262 ^H	40610 MW610	-	0263 ^H	40611 MW611	-
Y	24	0264 ^H	40612 MW612	-	0265 ^H	40613 MW613	-	0266 ^H	40614 MW614	-	0267 ^H	40615 MW615	-
Z	25	0268 ^H	40616 MW616	-	0269 ^H	40617 MW617	-	026A ^H	40618 MW618	-	026B ^H	40619 MW619	-

Quatre mots sont disponibles pour chaque variable:

- **Value 1:** Pour 202^H=0x00..01..02..03 =>Entier 16 bits ou Mot poids faible Entier 32 bits.
Pour 202^H= 0x04 => Caractères ASCII 1 et 2.
- **Value 2:** Pour 202^H=0x00..01..02..03 => Mot de poids fort Entier 32 bits:
Pour 202^H= 0x04 => Caractères ASCII 3 et 4.
- **Value 3:** Pour 202^H=0x00..01..02..03 => Point décimal. (voir tableau page suivante)
Pour 202^H= 0x04 => Caractères ASCII 5 et 6.
- **Value 4:** Pour 202^H=0x00..01..02..03 => Code Couleur*
Pour 202^H= 0x04 => Caractères ASCII 7 et 8 ou Code Couleur*

*Valeur pour code couleur: 0x00 = Ne Change pas; 0x01 = Rouge; 0x02 = Vert; 0x03 = Ambre/ Jaune; 0x04 = Bleu; 0x05 = Magenta; 0x06 = Cyan; 0x07 = Blanc.

NOTE: Dans le cas de variables alphanumériques (202^H=0x04) si la valeur 4 s'utilise pour le code couleur la longueur maximale se réduit de 8 à 6 caractères.

^H , 0x → Numération hexadécimale.

Value 3	Point Décimal
0	0000000000
1	000000000.0
2	00000000.00
3	0000000.000
4	000000.0000
5	00000.00000
6	0000.000000
7	000.0000000
8	00.00000000
9	0.000000000
10	0.0000000000
> 10	0.0000000000

Value 3: Code pour définir la position du point décimal de chaque variable

3.3.3. Exemples

Le Modbus hexadecimal PDU envoyé est détaillé pour chaque exemple suivant:

- Exemple 1: Écrire la valeur +10489 dans la variable A

Fonc.	Start Adress		Num.of words		Num. Bytes	Value 1		Value 2		Value 3	
10	02	04	00	03	06	28	F9	00	00	00	00

- Exemple 2: Écrire la valeur -10489 dans la variable A

Fonc.	Start Adress		Num.of words		Num. Bytes	Value 1		Value 2		Value 3	
10	02	04	00	03	06	D7	07	00	00	00	00

- Exemple 3: Écrire la valeur 3,4789 dans la variable B

Fonc	Start Adress		Num.of words		Num. Bytes	Type		Data Byte 3	Data Byte 4	Variable A Data 1		Variable A Data 2	
10	02	02	00	0A	14	00	01	00	00	00	00	00	00
Variable A Value 3		Variable A Value 4		Variable B Value 1		Variable B Value 2		Variable B Value 3		Variable B Value 4			
00	00	00	00	87	E5	00	00	00	04	00	00		

- Exemple 4: Écrire la valeur 74912 dans la variable B

Fonc	Start Adress		Num.of words		Num. Bytes	Type		Data Byte 3	Data Byte 4	Variable A Value 1		Variable A Value 2	
10	02	02	00	0A	14	00	02	00	00	00	00	00	00
Variable A Value 3		Variable A Value 4		Variable B Value 1		Variable B Value 2		Variable B Value 3		Variable B Value 4			
00	00	00	00	24	A0	00	01	00	00	00	00		

Nous rappelons que le Modbus PDU est l'unité de données du protocole, et il est encapsulé dans la trame du protocole dont le format dépend de la variante de Modbus (RTU ou TCP / IP).

Par exemple, si on travaille en Modbus RTU, dans le cas de l'exemple 1 et pour l'afficheur n° 01, la trame est la suivante

ID	PDU												CRC	
ID*	Fonc.	Start Adress		Num.of words		Num. Bytes	Value 1		Value 2		Value 3		CRC	
01	10	02	04	00	03	06	28	F9	00	00	00	00	36	D1

Trame hexadécimale ModBus RTU pour écrire la valeur 10489 dans la variable A de l'afficheur n°01

...dans le cas de ModBus-TCP la trame est la suivante

MBAP Header							PDU											
TID		Protocol ID		Length		Unit ID	Func	Start Adress		Num.of words		Num. Bytes	Value 1		Value 2		Value 3	
00	00	00	00	00	0D	FF	10	02	04	00	03	06	28	F9	00	00	00	00

Trame hexadécimale ModBus TCP pour écrire la valeur 10489 dans la variable A de l'afficheur n°01

* L'ID de l'afficheur peut être configuré de 1 à 253 depuis le logiciel Dynamic Plus

ANNEXE 1. Paramètres par défaut des afficheurs DM

La configuration d'usine est la suivante:

Paramètre	Valeur par défaut
Display ID	1
LocalCast Adress	0
RS232 port: Bauds	9600
RS232 port: Data BITS	8
RS232 port: Parity	Sans Parité
RS232 port: Stop BITS	1
RS485 port: Bauds	9600
RS485 port: Data BITS	8
RS485 port: Parity	Sans Parité
RS485 port: Stop BITS	1
IP address	192.168.1.100
DHCP client	Diéshabilité
Subnet mask	255.255.255.0
Gateway	192.168.1.1
TCP port for ModBus protocol	502 (fixe)

Configuration par défaut des afficheurs DINOS-DM

Les paramètres de configuration de l'afficheur peuvent être modifiés depuis le logiciel *Dynamic3*.

ANNEXE 2. SCRIPT DTPM (Uniquement pour Option 2: Envoi et exécution du Script d'un programme)

Description

DTPM est le protocole natif des afficheurs DM. Il se compose de codes permettant le contrôle total des afficheurs. L'ensemble de ces codes plus le texte du message à afficher compose le Script DTPM ou programme.

Les codes dédiés à l'édition permettent de contrôler les paramètres d'affichage tels que le type de police et la grosseur de caractère du texte à afficher, la sélection de la ligne où écrire, le mode et la vitesse d'apparition du message, le temps d'attente, la luminosité, la couleur du texte, le clignotement du texte complet ou partie du texte, l'alignement du texte, le synchronisme des lignes, l'insertion de variables temporelles (heure, date, compte à rebours), l'insertion de variables d'affichage numériques ou alpha-numériques, l'insertion de graphiques, etc.

S'il n'y a pas de codes ajoutés au texte envoyé, celui-ci sera affiché selon la configuration par défaut de l'afficheur. Certains paramètres d'édition par défaut sont fixes et d'autres comme la vitesse d'apparition et le temps d'attente peuvent être configuré via *Dynamic3*. Cependant certains codes sont indispensables à l'exécution du programme comme le mode d'apparition et ne peuvent pas être omis dans le Script envoyé.

POUR PLUS D'INFORMATION SUR LE PROTOCOLE DTPM CONSULTER LE MANUEL "COMMUNICATION AVEC LES AFFICHEURS SÉRIE DM VIA PROTOCOLE DTPM."

Liste et description des codes DTPM pour l'édition de programmes

Les codes d'édition sont composés de "pretoken"+"token"+variables"n" (certains codes n'ont pas de variables associées) .Ils sont classés par types (données, Modes et temps).

NAME	PRETO-KEN	TOKEN	DESCRIPTION
DATA			
Blink	0x03	0xA0	Le texte compris entre 2 "BLINK" clignote. (temps nom configurable)
Text Color < n >	0x03	0xA1+n	Couleur du texte qui suit* n= 0 – Ne change pas 1 – Rouge 2 – Vert 3 – Ambre/jaune 4 – Bleu 5 – Magenta 6 – Cyan 7 – Blanc
Graphic < n >	0x03	0xA4+n	Appeller un graphique n= 0 à 49.* 50 graphiques sont disponibles pour chaque hauteur de ligne (définis par la police de caractère). Pour connaître le n° d'un graphique voir sa position dans le logiciel <i>Dynamic3</i> . <i>Exemple pour Graphique N°21 : 0x03 0XA4 0X32 0X31 0X1F</i>
Variable <n>	0x03	0xAB+n	Afficher une variable interne et définir le format de présentation n=<Sign N°characters DecPoint N°decimals NameVariable+1F> Voir ANNEX 3
Flash < n >	0x02	0xB0+n	Le texte affiché clignote n fois (1 à10)
Erase	0x02	0xB2	La ligne active sera effacé

Thickness < n >	0x03	0xC0+n	Grosseur de caractère multiplié par n (1 à 4)
Font < n >	0x03	0xC1+n	Type de police (0 à 20)* <u>Voir tableau à continuation</u>
Speed of apparition < n >	0x03	0xC4+n	Vitesse du mode d'apparition (sauf mode immédiat). Très lent à très rapide (1 à 99).
Waiting time < n >	0x03	0xC5+n	Temps d'attente de n/4 secondes avant d'afficher la ligne (sans synchronisme) ou page (avec synchronisme) suivante.
Line < n >	0x03	0xC7+n	Sélection de la ligne défini par n=< x,y > x= Numéro de la ligne Y= Hauteur de la ligne en nombre de ligne (0x31 par défaut) <i>Exemple pour ligne N°2 : 0x03 0xC7 0X32 0X2C 0X31</i>
Synchronism	0x03	0xC9	Affichage synchronisée des lignes de la page. Sans ce paramètre l'affichage se fait par défaut de forme séquentielle.
End of synchronism	0x03	0xCA	Fin synchronisme
Language < n >	0x03	0xCB+n	Langues utilisée pour les tokens. n= 0(Spanish) 1(Catalan) 2(Basque) 3(Galician) 4(French) 5(English) 6(Portuguese)
Text alignment	0x03	0xCD+n	Alignement du texte 0= Centré 1= Aligné à gauche 2= Aligné à droite.
Brightness < n >	0x03	0xD0+n	Mode de contrôle et niveau de la luminosité. n=0 pour contrôle automatique de la luminosité*. n = 1 à 100% valeur fixe de luminosité.
Window < n >	0x03	0xD3+n	Définit une fenêtre d'affichage. n=<ID window,x1,y1,x2,y2> (les caractères intermédiaires doivent également s'envoyer) ID window: Identification de la fenêtre de A à N x1: Numéro de colonne où commence la fenêtre (coté droit). y1: Numéro de ligne où commence la fenêtre (coté bas) x2: Numéro de colonne où finit la fenêtre (coté gauche). y2: Numéro de ligne où finit la fenêtre (coté haut) <i>Exemple : Voir page 24</i>

< n > → Variable en format ASCII associée au code . Par exemple 0x31 pour valeur 1.

0x → Numération hexadécimale.

* Selon modèle

NAME	PRETO-KEN	TOKEN	DESCRIPTION
MODES			
Appearing Left	0x04	0xD0	Le texte complet défile de gauche à droite de la ligne en marquant un arrêt lorsque la ligne est pleine.
Appear Right	0x04	0xD1	Le texte complet défile de droite à gauche de la ligne en marquant un arrêt lorsque la ligne est pleine avant de continuer
Scroll	0x04	0xE0	Le texte complet défile de droite à gauche de la ligne sans marquer d'arrêt.
Ascend	0x04	0xE5	Le texte apparaît de bas en haut.
Descend	0x04	0xE6	Le texte apparaît de haut en bas.
Immediate	0x04	0xF0	Le texte apparaît sur la ligne de forme immédiate.

NAME	PRETO-KEN	TOKEN	DESCRIPTION
TIME (internal clock)			
Current Date	0x01	0x95	Affiche la date actuelle en format DD/MM/YY
Current Year YY	0x01	0x96	Affiche l'année en cours en format YY
Current Month number MM	0x01	0x97	Affiche le mois en cours en format MM
Current Day number DD	0x01	0x99	Affiche le jour actuel en format DD
Current Time	0x01	0x9E	Affiche l'heure actuelle en format HH:MM:SS
Hours : Minutes	0x01	0xA7	Affiche l'heure actuelle en format HH:MM
Current Hour HH	0x01	0x9B	Affiche l'heure actuelle en format HH
Current Minutes mm	0x01	0x9C	Affiche l'heure actuelle en format MM
Current Seconds SS	0x01	0x9D	Affiche l'heure actuelle en format SS

0x → Numération hexadécimale.

Current Month long name	0x01	0x98	Affiche le nom du mois en cours
Current Month short name	0x01	0xAA	Affiche le nom en cours en abrégé (3 caractères)
Current Day long name	0x01	0x9A	Nom de la journée en cours
Current Day short name	0x01	0xA9	Affiche le nom de la journée en cours en abrégé (3 caractères)
Current Temperature °C	0x01	0xA8	Affiche la température en format xx°C*
Current Temperature	0x01	0x9F	Affiche la température en format xx*
Event Date	0x03	0xCC +n	Date de référence pour le calcul des Événements. n=<DD-MM-YY HH:MM:SS> (les caractères intermédiaires doivent également s'envoyer)
Differ Days	0x01	0xA4	Différence en jours entre la date courante et la date de l'événement (0xCC). Comptage montant (événement futur) ou descendant (événement passé)
Differ Weeks	0x01	0xA5	Différence en semaines entre la date courante et la date de l'événement (0xCC). Comptage montant (événement futur) ou descendant (événement passé)
Differ Month	0x01	0xA6	Différence en mois entre la date courante et la date de l'événement (0xCC). Comptage montant (événement futur) ou descendant (événement passé)
Differ hours	0x01	0xAB	Différence en heures entre la date courante et la date de l'événement (0xCC). Comptage montant (événement futur) ou descendant (événement passé)
Differ minutes	0x01	0xAC	Différence en minutes between current date and event date (0xCC). Comptage montant (événement futur) ou descendant (événement passé)
Differ seconds	0x01	0xAD	Différence en secondes entre la date courante et la date de l'événement (0xCC). Comptage montant (événement futur) ou descendant (événement passé)
Remaing time in Days	0x01	0xAE	Jours restants pour la date de l'événement. Comptage montant (événement futur) ou descendant (événement passé)
Remaing time in Hours	0x01	0xAF	Heures restantes pour la date de l'événement. Comptage montant (événement futur) ou descendant (événement passé)
Remaing time in Minutes	0x01	0xB0	Minutes restantes pour la date de l'événement. Comptage montant (événement futur) ou descendant (événement passé)
Remaing time in Segondes	0x01	0xB1	Secondes restantes pour la date de l'événement. Comptage montant (événement futur) ou descendant (événement passé)

Listes des codes DTPM pour l'édition de programme

< n > → Variable en format ASCII associée au code. Par exemple 0x31 pour valeur 1.

0x → Numération hexadécimale.

* Selon modèle

Nombre de pixel	Nom de la Police	CODE (Pretoken+Token+n) Numération hexadécimale
6 pixels	Pequeña	03 C1 37
7 pixels	LCD	03 C1 31 30
	Normal	03 C1 38
	Ampliada	03 C1 31 32
	Italica	03 C1 39
	West	03 C1 31 31
	Vertical	03 C1 33 32
	Vertical	03 C1 33 33
8 pixels	Normal	03 C1 32 31
12 pixels	Stand 12	03 C1 31 34
14 pixels	Normal	03 C1 30
	Broadway	03 C1 36
	Futura	03 C1 33
	Gótica	03 C1 35
	Popcorn	03 C1 32
	Vacía	03 C1 31
	Western	03 C1 34
16 pixels	Normal	03 C1 31 38
	Big	03 C1 31 39
21 pixels	Stand 21	03 C1 31 33
24 pixels	Stand 24	03 C1 31 37
28 pixels	Stand 28	03 C1 31 35
	Broad	03 C1 31 36
32 pixels	Normal	03 C1 33 34

Code DTPM pour les polices de caractères disponibles

Structure de la trame d'un programme

Les codes DTPM sont exécutés de forme séquentielle dans un ordre déterminé:

Les codes d'édition DATA vont en premiers suivis par les codes de mode d'apparition MODE qui sont indispensables puis vient le texte à afficher et éventuellement les codes correspondants aux variables temporelles TIME, aux variables internes et aux graphiques qui peuvent être insérés dans le texte.

A la fin viennent les codes d'effet qui modifient le contenu des données affichées.

Si un code n'est pas dans sa position correcte dans la trame, le programme ne s'exécutera pas ou de forme incomplète. La codification Windows-1252 (extension de la norme ISO-8859-1) est utilisée pour les codes de caractères imprimables à partir du 0x20.

Position dans la trame	Nom	CODE (Pretoken+Token+n) Numération hexadécimale
Pos1	Brighthness	03 D0
Pos2	Alignement	03 CD+n
Pos3	Language	03 CB
Pos4	Synchronism	03 C9
Pos5	Window	03 D3+n
Pos6	Line	03 C7+n
Pos7	Font	03 C1+n
Pos8	Ticness	03 C0+n
Pos9	Waiting time line or page	03 C5+n
Pos10	Speed apparition mode	03 C4+n
Pos11	Apparition Mode	04 D0 to F0
Pos12	Text Color	03 A1+n
Pos13	Message to display	Caractères imprimables en code ASCII + codes TIME, variables et graphiques.
Pos14	Blink	03 A0 (Avant et après le text sélectionné)
Pos15	Flash	02 B0+n
Pos16	Erase	02 B2
Pos17	End of synchronism	03 CA

Position des codes DTPM les plus habituels dans la trame d'un programme

Exemples de Script

Script en hexadécimal (texte en gras) envoyé au registre 0100^H (0x100) et résultat à l'affichage.

RAPPEL: Le protocole ModBus permet l'envoi de **124 mots maximum** dans chaque trame.

Immédiat Mode: Envoyer la trame " 04 F0 **48 65 6C 6C 6F 00** " pour afficher "Hello" en mode immédiat.

Scroll Mode: Envoyer la trame "04 E0 **48 65 6C 6C 6F 00**" pour afficher "Hello" en mode de défilement

Speed<n>: Envoyer la trame " 03 C4 34 35 04 E0 **48 65 6C 6C 6F 00** " pour afficher "Hello" en mode de défilement avec une vitesse de défilement de 45%.

Line<n>: Envoyer la trame " 03 C7 32 2C 31 04 F0 **48 65 6C 6C 6F 00** " pour afficher "Hello" sur la ligne 2 en mode immédiat.

Color <n>: Envoyer la trame " 04 F0 03 A1 31 **48 65 6C 6C 6F 00** " pour afficher "Hello" en rouge et mode immédiat

Left: Envoyer la trame " 03 CD 31 04 F0 03 A1 31 **48 65 6C 6C 6F 00** " pour afficher "Hello" en rouge, mode immédiat et texte aligné à gauche.

Brightness<n>: Envoyer la trame " 03 D0 35 30 04 F0 03 A1 31 **48 65 6C 6C 6F 00** " pour afficher "Hello" en rouge, mode immédiat et luminosité de 50%.

Window < n >: Envoyer la trame " 03 D3 41 2C 37 30 2C 31 2C 31 32 30 2C 32 04 F0 **48 65 6C 6C 6F 00** " pour créer dans le côté droit de l'affichage la fenêtre A de 2 lignes de hauteur qui montre "Hello". (A noter que cet exemple est prévu pour un affichage de longueur 120 pixels)

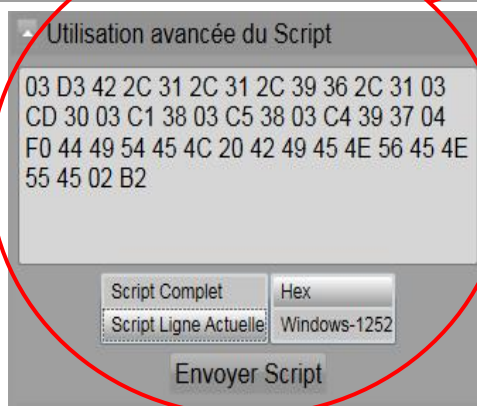
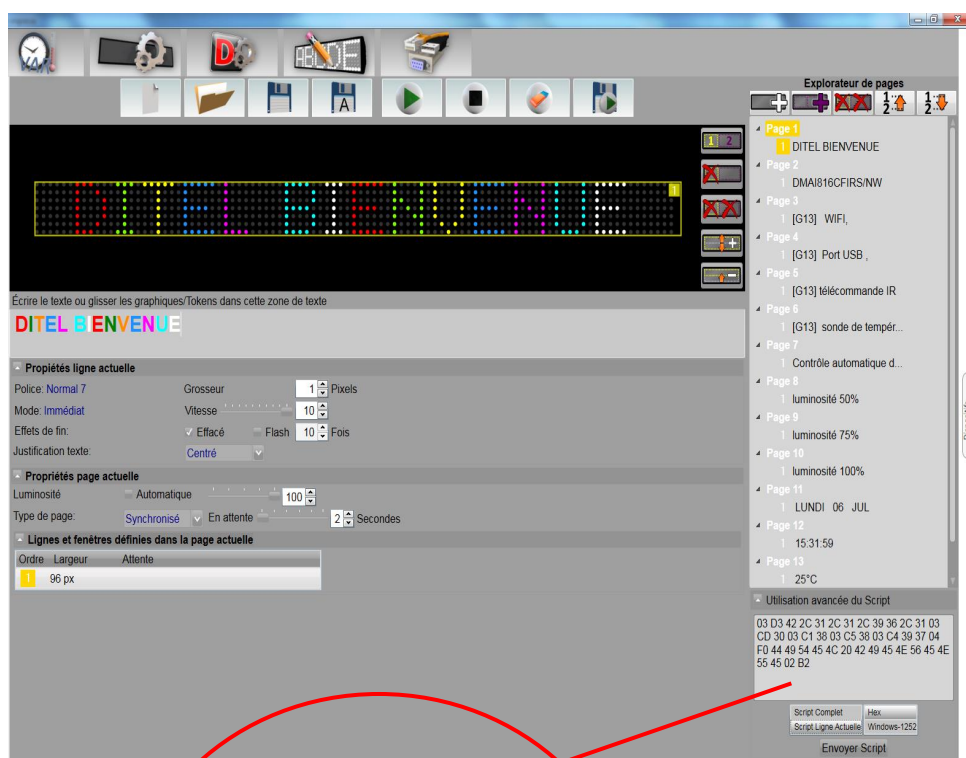
ATTENTION: Les caractères Null 0x00 ne doivent pas être utilisés dans la trame car l'afficheur l'interprétera comme un fin de trame et n'affichera pas la trame derrière ce caractère. Si nécessaire, remplacer les caractères Null 0x00 par le caractère espace 0x20.

Génération et test de Script DTPM avec le logiciel Dynamic3.

Le logiciel d'édition et configuration Dynamic3 dispose également dans l'onglet **Edition** 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 et test de trames.**

De cette forme il n'est pas nécessaire de connaître en détails les codes DTPM et leur position exacte dans la trame. On peut simplement récupérer tous le code d'un programme pour le réutiliser dans 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.



ANEXO 3. Représentation des Variables (Uniquement pour Option 2: Envoi et exécution du Script d'un programme)

Pour insérer dans un Script DTPM une des 26 variables interne de l'afficheur il faut utiliser le code 0x03+0xAB+n (Voir Liste des codes DTPM).

"n" correspond au nom de la variables [A, Z] + les paramètres pour le format de représentation de celle-ci.

La valeur des variables internes ainsi définis se modifie depuis les registres ModBus 204^H à 026Ah. Voir chapitre 3.3. "OPTION 3: Utilisation des variables internes de l'afficheur".

EXEMPLE: Var A

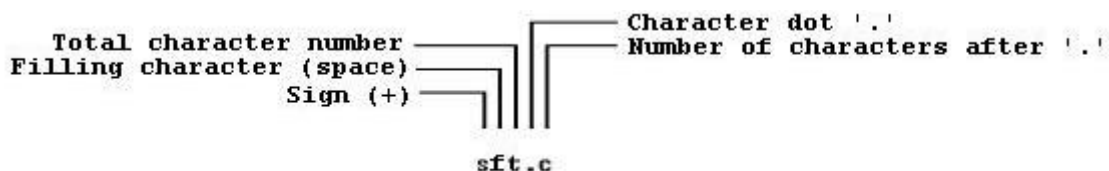
Numération hexadécimale: 03AB41

03 AB = Pretoken+Token

41 = "A" Id variable

Le format de représentation par défaut d'une variable comme dans l'exemple antérieur est composé de 6 chiffres après la virgule. Comme il peut être peu commode représenter des nombres entiers avec autant de décimales, on peut formater la variable en définissant le nombre total de chiffres et le nombre de chiffres après la virgule, de la forme suivante:

VAR <format> Variable



EXEMPLE: VAR+06.2A

Numération hexadécimale: 03 AB 2B 30 36 2E 32 41

03 AB = Pretoken+Token

2B = "+" Signe // facultatif

30 = "0" Remplir de zéros // facultatif

36 = "6" Nombre total de caractères // facultatif

2E = "." Point décimal

32 = "2" Nombre décimal // facultatif

41 = "A" Id Variable

1F = Obligatoire

Les paramètres facultatif comme signe, remplir, nombre de caractères et nombre de décimales offrent 6 formats de représentation possibles.

SIGN DISPLAY	LEFT ALIGNMENT	RIGHT ALIGNMENT	RIGHT ALIGNMENT
YES	+1.00	_+1.00	+01.00
NO	1.00	__1.00	001.00
FILL WITH CHARACTER	X	FILL WITH SPACES	FILL WITH ZERO

Formats de représentation disponibles

Exemples de représentation

ID Variable = A
Value=1

VAR.A (03 AB 2E 41 1F).....le résultat est: 1 (Nbr.carcatères et point décimal non nécessaire)
VAR+.A (03 AB 2B 2E 41 1F).....le résultat est: +1 (Nbr.carcatères et point décimal non nécessaire)

VAR03.A (03 AB 30 33 2E 41 1F).....le résultat est: 001
VAR+03.A (03 AB 30 2B 33 2E 41 1F)..le résultat est: +01

VAR+.2A (03 AB 2B 34 2E 32 41 1F).....le résultat est: +1.00
VAR+0.2A (03 AB 2B 30 36 2E 32 41 1F)le résultat est: +01.00

VAR3.A (03 AB 33 2E 41 1F).....le résultat est: __1 ("__" représentent 2 espaces)
VAR+3.A (03 AB 2B 33 2E 41 1F).....le résultat est: _+1 ("_" représentent 1 espaces)

VAR-3.A (03 AB 2D 33 2E 41 1F).....le résultat est: 1__ ("__" représentent 2 espaces)

D'autres combinaisons sont possibles.

Exemples de trames

Adresse de départ	0100 ^H (0x100)
Nombre de registres (Nbr. de Mots)	1 à 124 (Modbus Standard)
Données	Script du programme + <u>un caractère NULL (0x00)</u> pour indiquer la fin du bloc

Les exemples suivants détail le bloc de données en numération hexadécimale (texte en gras + codes).

VAR3.A: Envoyer la trame "04 F0 **56 49 54 45 53 53 45 3A** 03 AB 33 2E 41 1F **6D 2F 73**"
pour afficher "VITESSE: *VarA* m/s" en mode immédiat et la variable en format "__1"

- Affichage pour 202^H=0 204^H= 123 et 207^H=0 : "VITESSE:123m/s"
- Affichage pour 202^H=0 204^H= 123 et 207^H=1 : "VITESSE: 12m/s"

VAR3.1A: Envoyer la trame "04 F0 **56 49 54 45 53 53 45 3A** 03 AB 33 2E 31 41 1F **6D 2F 73**"
pour afficher "VITESSE: *VarA* m/s" en mode immédiat et la variable en format "_1.0"

- Affichage pour 202^H=0 204^H= 123 et 207^H=0 : "VITESSE:123.0m/s"
- Affichage pour 202^H=0 204^H= 123 et 207^H=1 : "VITESSE:12.3m/s"
- Affichage pour 202^H=0 204^H= 123 et 207^H=2 : "VITESSE:1.2m/s"

" , 0x → Numération hexadécimale.