

Annexe 7 : Afficheur SX502

1. Caractéristiques techniques :

La version de l'appareil est codée comme suit dans la désignation du type :

SX502	-			0	/	0		/			-				/			-	M	0
2 lignes	2	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
4 lignes	4	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
6 lignes	6	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
8 lignes	8	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
20 caract./ligne*	2	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
40 caract./ligne*	4	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Hauteur de caractères 33/66/75 mm	3	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Hauteur des caract. 50/100/120 mm	5	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
LED standard	0	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
LED pour applications extérieures	2	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Couleur des LED rouge	R	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Couleur des LED commutable rouge/vert/orange	M	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Affichage monoface	1	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Affichage bi-faces	2	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Boîtier tôle d'acier, laqué	0	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Boîtier tôle d'acier, laque double couche	1	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Boîtier acier inoxydable V2A, laqué	2	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Boîtier acier inoxydable V2A, brossé	3	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Boîtier acier inoxydable V4A, brossé	5	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Indice de protection IP54	0	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Indice de protection IP65	1	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Indice de protection IP54 avec compensation climatique	2	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Indice de protection IP54 avec compensation climatique et chauffage	4	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Montage mural, entrée de câble en bas	0	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Montage mural, entrée de câble en haut	1	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Montage suspendu, entrée de câble en bas	2	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Montage suspendu, entrée de câble en haut	3	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Montage mural et suspendu, entrée de câble en bas	4	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Montage mural et suspendu, entrée de câble en haut	5	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Tension d'alimentation 230 V AC $\pm 15\%$, 50 Hz	A	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Tension d'alimentation 24 V DC $\pm 15\%$	B	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
Tension d'alimentation 115 V AC $\pm 15\%$, 60 Hz	C	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:

* concerne la fonte Acala 7

2. Affichage des caractères

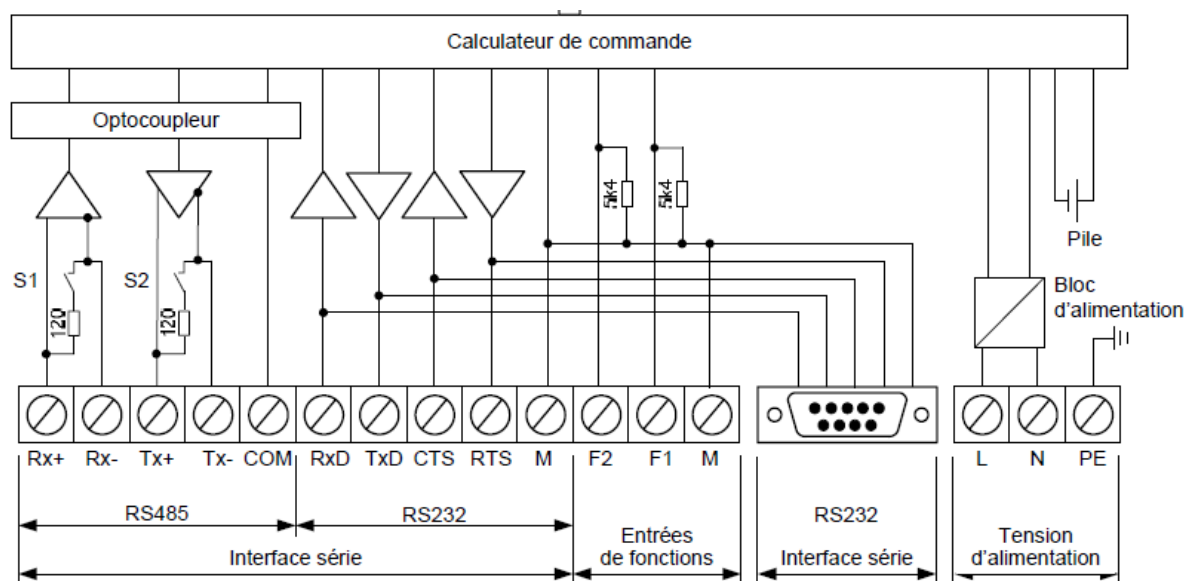
La largeur des caractères est de 5 pixels et l'espacement entre les caractères est de 1 pixel.

La hauteur des caractères est de 7 pixels.

L'interligne est de 2 Pixels.

Fonte de caractères	SX502-xxx/03/xx-xxx/xx-xx	SX502-xxx/05/xx-xxx/xx-xx
Acala 7	env. 33 mm	env. 50 mm
Acala 7 extended	env. 33 mm	env. 50 mm
Acala 14 condensed	env. 66 mm	env. 100 mm
Acala 14	env. 66 mm	env. 100 mm
Acala 14 extended	env. 66 mm	env. 100 mm
Acala 16 condensed	env. 75 mm	env. 120 mm
Acala 16	env. 75 mm	env. 120 mm
Acala 16 extended	env. 75 mm	env. 120 mm

3. Schéma de principe



4. Interface

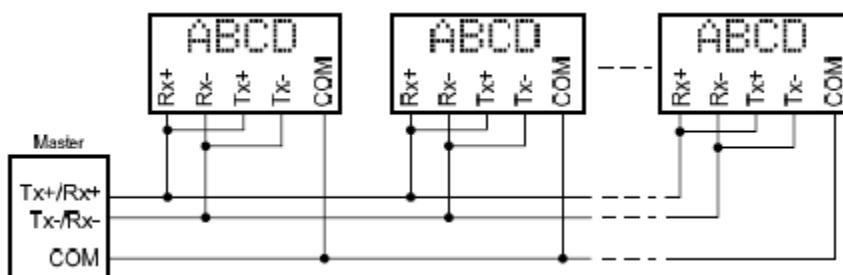
Pour la commande Modbus, utiliser l'interface RS485.

L'interface RS232 n'est pas recommandée pour la commande Modbus. Elle est prévue pour la programmation de l'appareil avec un PC, par exemple pour charger des textes statiques dans la mémoire de textes et pour installer des fontes de caractères à l'aide de l'outil de PC

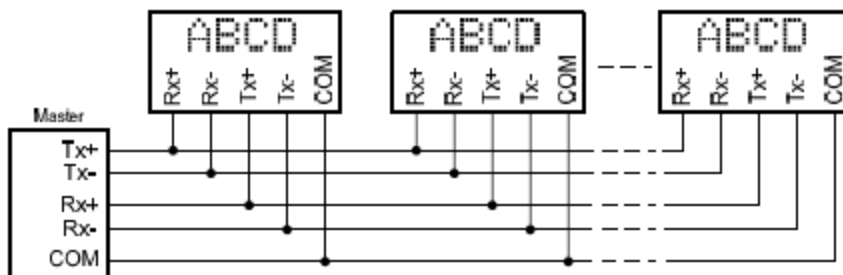
'DisplayManager', livré sur supports de données.

5. Câblage et raccordement

La documentation *Modbus over serial line specification and implementation guide* recommande le câblage de raccordement d'un bus RS485 2 fils (Two-Wire Modbus Definition) comme représenté sur le schéma page suivante.



Il est également possible de réaliser le câblage de raccordement d'un bus RS485 4 fils (Optional Four-Wire Modbus Definition) comme représenté sur le schéma ci-dessous.



Les lignes de données de l'interface RS485 doivent être équipées aux deux extrémités de résistances de terminaison pour obtenir un fonctionnement sans perturbations. Les résistances nécessaires se trouvent sur le calculateur de commande et peuvent être activées avec les commutateurs S1 pour Rx et S2 pour Tx.

6. Commande de l'afficheur

6.1. Modbus

Les appareils sont commandés en esclave Modbus RTU (Remote Terminal Unit).

Ils utilisent le code fonction 16 (0x10) : Ecriture de registres multiples (Write Multiple Registers) conformément au protocole Modbus.

6.2. Mode d'opération 'textes dynamiques et statiques'

Adresse de départ

Pour l'adresse de départ, utiliser l'adresse de registre 0000h.

Interprétation des données

Les données sont interprétées selon le tableau des commandes suivant. Dans la description des commandes, les chiffres indiqués entre [] se rapportent aux lignes correspondantes du tableau des commandes.

Tableau des commandes :

Commandes de manipulation des textes

Afficher texte dynamique	cc...	Envoyer des caractères quelconques	[1]
Afficher texte statique	\$Tn	Appeler texte statique (n = numéro de texte, 1 à 4 digits)	[2]
Afficher texte dynamique	\$Lxxcc...	Envoyer des caractères quelconques à la ligne xx	[43]

Dans les lignes [1] et [43], **cc...** représente une chaîne de caractères avec un contenu quelconque.

Ligne [43] : Les lignes sur l'afficheur sont numérotées à partir de 1.

Commandes de formatage de textes

Retour à la ligne	\$C	Retour à la ligne forcé	[6]
Clignotement	\$F1	Clignotement des caractères suivants marche	[7]
	\$F0	Clignotement des caractères suivants arrêt	[8]
Texte déroulant	\$Y	Texte déroulant à partir de la position actuelle jusqu'à la fin du texte ou \$C	[9]
Fonte de caractères	\$M1	Acala 7	[10]
	\$M2	Acala 7 extended	[11]
	\$M3	Acala 14 condensed	[12]
	\$M4	Acala 14 extended	[13]
	\$M5	Acala 7 P / Fonte de caractères personnalisée 7 pixels	[14]
	\$M6	Acala 14 / Fonte de caractères personnalisée 14/16 pixels	[15]

1. Modbus

1.1. Présentation

MODBUS est un protocole de communication couramment utilisé dans l'industrie pour faire communiquer des automates programmables.

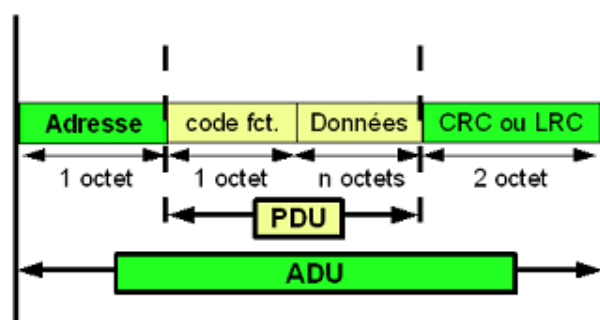
Le protocole MODBUS est un protocole de dialogue basé sur une structure hiérarchisée entre un maître et plusieurs esclaves. Les esclaves possèdent une adresse comprise entre 1 et 64.

1.2. Échange Maître/ Esclave sur Modbus

Le maître interroge un esclave de numéro unique sur le réseau et attend de la part de cet esclave une réponse.

Lorsque l'esclave envoie sa réponse, il place sa propre adresse dans le champ adresse afin que le maître puisse l'identifier.

1.3. Format d'une trame Modbus



Une trame Modbus est composée de deux parties :

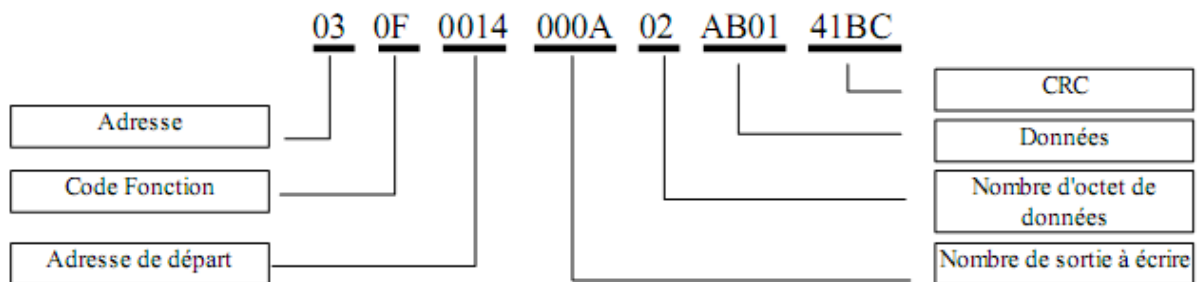
- **"ADU" Application Data Unit** : Cette partie est fonction de la couche de communication sur laquelle s'appuie Modbus.
L'exemple ci-dessus correspond à un **ADU Modbus sur liaison série**.
Pour une trame **ModbusTCP** l'**ADU** est différent (voir le paragraphe 3 de cette annexe).
- **"PDU" Protocol Data Unit** : Il est composé des champs :
 - **"Code fonction"** : Ce champ est prédéfini par le protocole Modbus (voir paragraphe 2 de cette annexe). Ce code fonction définit l'action à exécuter sur l'esclave.
Par exemple le code fonction 0x0F (write Multiple Coils) permet de modifier les sorties numériques sur un esclave.
 - **"Données"** : Dans le champ "Données" sont présentes des informations relatives au code fonction (exemple : adresse de registre pour l'écriture sur les sorties) et des données à échanger entre le maître et l'esclave (exemple : les informations relatives aux sorties à mettre à l'état haut).

Codage des trames Modbus sur liaison série :

Deux types de codage peuvent être utilisés pour communiquer sur un réseau Modbus sur liaison série : ASCII ou RTU

- Mode **ASCII** : Chaque octet composant une trame est codé avec 2 caractères ASCII (2 fois 8 bits).
Par exemple, la valeur 01 est codée avec les codes ASCII des symboles '0' et '1' (soit 0x30 et 0x31).
- Mode **RTU** : Les octets composants la trame ne sont pas codés.
Par exemple, la valeur (15)₁₀ aura pour octet 0x0F ou (00001111)₂.

Exemple d'une requête modbus RTU pour écrire une série de 10 sorties commençant à l'adresse 20 sur l'esclave à l'adresse 3:



2. Fonctions Modbus

15 (0x0F) Write Multiple Coils

This function code is used to force each coil in a sequence of coils to either ON or OFF in a remote device. The Request PDU specifies the coil references to be forced. Coils are addressed starting at zero. Therefore coil numbered 1 is addressed as 0.

The requested ON/OFF states are specified by contents of the request data field. A logical '1' in a bit position of the field requests the corresponding output to be ON. A logical '0' requests it to be OFF.

The normal response returns the function code, starting address, and quantity of coils forced.

Request PDU

Function code	1 Byte	0x0F
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Outputs	2 Bytes	0x0001 to 0x07B0
Byte Count	1 Byte	N*
Outputs Value	N* x 1 Byte	

*N = Quantity of Outputs / 8, if the remainder is different of 0 $\Rightarrow$ N = N+1

Response PDU

Function code	1 Byte	0x0F
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Outputs	2 Bytes	0x0001 to 0x07B0

Error

Error code	1 Byte	0x8F
Exception code	1 Byte	01 or 02 or 03 or 04

16 (0x10) Write Multiple registers

This function code is used to write a block of contiguous registers (1 to 123 registers) in a remote device.

The requested written values are specified in the request data field. Data is packed as two bytes per register.

The normal response returns the function code, starting address, and quantity of registers written.

Request

Function code	1 Byte	0x10
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	0x0001 to 0x007B
Byte Count	1 Byte	2 x N*
Registers Value	N* x 2 Bytes	value

*N = Quantity of Registers

Response

Function code	1 Byte	0x10
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 123 (0x7B)

Error

Error code	1 Byte	0x90
Exception code	1 Byte	01 or 02 or 03 or 04

3. ModbusTCP

3.1. Présentation

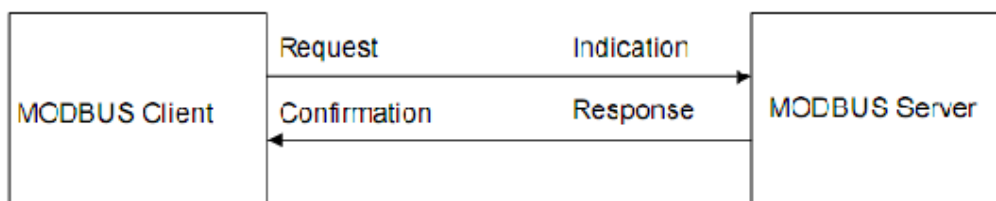
ModbusTCP est la variante "encapsulée" dans TCP/IP du protocole Modbus.

Le protocole ModbusTCP est basé sur un mode de communication client / serveur, celui-ci permet d'échanger des informations entre différents produits d'automatismes sur Ethernet TCP/IP.

Il existe plusieurs familles d'équipement ModbusTCP :

- Des clients et des serveurs Modbus TCP.
- Des matériels d'interconnexion (ponts, routeurs ou passerelles) assurant le lien entre le réseau TCP/IP et une ou plusieurs liaisons Modbus Série, sur lesquelles sont connectés différents terminaux Modbus.

3.2. Le modèle de ModbusTCP



Le modèle Modbus TCP est basé sur 4 types de message :

- Modbus request : Message envoyé par un client sur le réseau pour initier une transaction
- Modbus Indication : c'est le message "Modbus request" reçu par le serveur modbusTCP
- Modbus Response : Message de réponse envoyé par le serveur
- Modbus Confirmation : C'est le message de réponse reçu par le client.

3.3. Encapsulation d'une trame Modbus dans ModbusTCP

Le mapping du protocole Modbus sur des bus spécifiques ou des réseaux peut introduire des champs supplémentaires au niveau de l'unité de donnée d'application, ou ADU pour Application Data Unit.

Pour ModbusTCP, le champ supplémentaire ajouté dans l'ADU est appelé : MBAP (Modbus Application Protocol Header)

Description du champ MBAP :

- **Transaction Identifier** (2 octets): il s'agit en fait du numéro d'un message MODBUS. Chaque message (requête+réponse) est identifié par un numéro différent. Ce numéro est initialisé par le client et le serveur le recopie dans la réponse.
- **Protocol Identifier** (2 octets): Numéro du protocole pour les systèmes multiplexés qui en utilisent plusieurs (MODBUS= 0).
- **Length** (2 octet) : Longueur du message à suivre (Unit Identifier + Data)
- **Unit Identifier** (1 octet) : Adresse de l'esclave connecté en liaison série derrière une passerelle ModbusTCP / Modbus