

Introduction

Ce dossier propose à des étudiants de BTS Électrotechnique de découvrir les techniques d'échanges de données numériques mises en œuvre entre des systèmes et produits communicants à travers les liaisons séries asynchrones.

Les manipulations permettent aux étudiants d'identifier clairement le mode d'échange de données entre de multiples partenaires sur un unique support physique bifilaire.

Présentation

Titre de la séquence

Découverte des réseaux de communication asynchrones, application au réseau Modbus

Format

Essais de systèmes sur 3 séances, travaux pratiques incluant préparation

Plan de formation

Séance 1 (3h)

Objectif : Être capable de mettre en œuvre une communication série asynchrone

Contenus :

- Mettre en œuvre le câble de connexion RS232 ou RS485
- Configurer un terminal ASCII (= émission/réception de texte) sur un ordinateur selon les connexions mises en œuvre
- Afficher une trame de communication série asynchrone sur un oscilloscope numérique (mode single)
- Identifier le caractère ASCII échangé sur une trame
- Échanger des caractères entre 2 ordinateurs

Séance 2 (3h)

Objectif : Être capable d'interroger un appareil Modbus esclave à partir d'un PC

Contenus :

- Connecter le poste maître et l'appareil esclave
- Installer un programme d'interrogation Modbus
- Afficher les trames échangées
- Décoder les trames pour identifier la requête échangée

Séance 3 (3h)

Objectif : Être capable de programmer un automate en Modbus Maître pour interroger des appareils esclaves

Contenus :

- Analyser la documentation de l'API
- Installer un programme exemple pour une trame
- Préparer les trames nécessaires pour une requête donnée
- Programmer l'enchaînement de plusieurs trames

Droits

Ce document est destiné à la diffusion sur le « Réseau National de Ressources en électrotechnique » Réselec : <http://www.iufmrese.cict.fr/> . Son usage est réservé à l'enseignement public.

Toute publication ou édition à but commercial est interdite quel que soit le support.

Évolution

Ce document peut être amendé. Pour le suivi de la qualité de son contenu, toutes les corrections et propositions d'évolution peuvent être envoyées à son auteur :

Stéphane Gautreau – Lycée Bernard Palissy – 17100 Saintes – stephane.gautreau@ac-poitiers.fr

Contenu du dossier

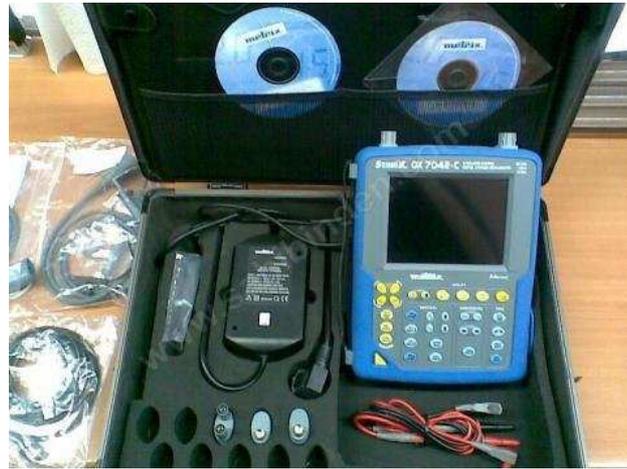
<i>Désignation</i>	<i>Type</i>	<i>Index document</i>	<i>Nombre de pages</i>
Présentation du dossier, plan de formation, matériel nécessaire	Introduction	2600	4
<i>Les sujets de manipulations pourront être reproduits de préférence sur papier BLEU</i>			
TP 1 : Communications asynchrones RS232/RS485	Manipulations	2625	4
TP 2 : Dialogue Modbus entre PC maître et dispositif esclave	Manipulations	2635	3
TP 3 : Communication ModBus Maître depuis un API	Manipulations	2645	4
TP 4 : Gestion d'une imprimante au fil de l'eau - Complément	Manipulations	2655	2
<i>Les fiches méthodes pourront être reproduits de préférence sur papier JAUNE</i>			
Réglage oscilloscope pour capture d'événement non périodique	Méthode	2603	1
<i>Les documents-ressources pourront être reproduits de préférence sur papier VERT</i>			
Le codage des caractères : code ASCII	Ressource	2602	2
Communications asynchrones RS232/RS485	Ressource	2612	8
Configuration de l'HyperTerminal	Ressource	2622	2
Protocole de communication ModBus	Ressource	2632	2
Communications asynchrones sur Twido	Ressource	2642	3
<i>Documentations-constructeurs des dispositifs communicants MODBUS</i>			
Description des systèmes étudiés	Ressource	2698	4
Doc Analyseur d'énergie Merlin-Gerin PM820			6
Doc. Analyseur d'énergie ARDETEM PECA30E			11
Doc. Variateur de vitesse Leroy-Somer Digidrive			4
Doc. Régulateur de température Chauvin-Arnoux STATOP48			6
Doc. Communications ModBus sur Schneider Twido			8 x2

Matériels nécessaires

	S1	S2	S3	Coût indicatif
Matériel standard				
Ordinateur type PC sous Windows avec port série 9 broches	X	X	X	900,00 €
Oscilloscope numérique + accessoires	X	X	X	1 000,00 €
Automate programmable avec port RS485 Sud-D et option Modbus Maître : Schneider-Electric TWIDO			X	300,00 €
Outil de programmation TwidoSoft V3.5 Téléchargeable gratuitement			X	
Option 2 ^{ème} port COM RS485 Adaptateur			X	
Matériels spécifique à acquérir				
Adaptateur RS232/RS485 avec adaptateur secteur 230V~/9V= Réf. Roline IC-485S Réf. Radiospares 523-7342 / 560-634	X	X	X	53,00 €
Adaptateur RS232 PC → AT – DB9F/DB25F – Long. 30cm Réf. Radiospares 202-644	X	X	X	2,60 €
Câble blindé multi-conducteurs pour communications RS232 Réf. BZZ4825516V025 - 7 brins de 0,2 mm ² / 25 mètres Réf. Radiospares 660-0454	X			26,43 €
Câble pour paire RS485 blindée		X	X	
DAQFactory (Téléchargement de la version Démo)		X		0,00 €
Logiciel TwidoSoft			X	
Systèmes communicants – Selon lieu de formation		X	X	
Sous-systèmes communicants – Selon lieu de formation	X	X	X	
Matériels à préparer				
Câble multi-conducteurs (7) dont 3 à souder sur prise canon 9 broches au format RS232 (Gnd {noir}, Tx {rouge}, Rx {vert})	X			
Câble RS485 3 conducteurs à câbler sur convertisseur Roline	X	X	X	
Installer HyperTerminal (gratuit)	X			
Installer DAQFactory (Demo)		X		
Installer application de programmation API TwidoSoft avec licences			X	



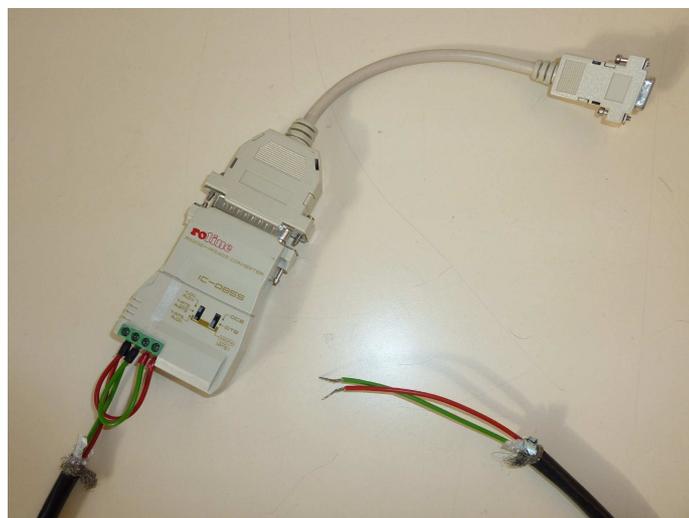
Connectique ordinateur PC avec port série 9 broches



Oscilloscope numérique Scopix 7042C



Adaptateur RS232/RS485
+ son bloc adaptateur d'alimentation secteur
+ réducteur PC-AT



Adaptateur + câble monté
(Le bloc adaptateur d'alimentation secteur
n'apparaît pas ici, mais ne l'oubliez pas !)

Sommaire

Objectifs.....	1
Contexte.....	1
Mise en situation.....	1
Liaison RS 232.....	2
Préparation (50 minutes).....	2
Manipulations.....	3
Liaison RS 485.....	4
Préparation, d'après la documentation mise à votre disposition :.....	4
Manipulations.....	4

Objectifs

Vous reviendrez sur ces objectifs en fin de séance pour savoir si ceux-ci ont été atteints.

- Être capable de configurer un logiciel terminal pour émettre des caractères sur le port série RS232 ;
- Être capable de capturer sur oscilloscope une trame de caractère saisi au clavier ;
- Être capable de déterminer le caractère émis à partir de l'oscillogramme et d'une table de code ASCII ;
- Être capable de mettre en place un dialogue en ASCII en RS232 entre 2 terminaux ;
- Être capable de mettre en œuvre un adaptateur RS232/RS485.

Contexte

Prendre connaissance des systèmes communicants disponibles dans la salle d'essais de systèmes, et affecter un binôme d'apprenants sur chaque poste.

Mise en situation

Vous disposez du matériel suivant :

- Micro-ordinateur sous Windows XP
 - disposant d'une liaison série Sub-D Canon 9 broches
- Cordon 9 conducteurs colorés, long. 1 m :
 - 1 extrémité prise Canon SUB-D 9 broches femelle
 - 1 extrémité fils volants
- Adaptateur RS232-RS485 Roline
 - + son adaptateur d'alimentation secteur
 - + 1 câble RS485 2 conducteurs Rouge/Vert
- Oscilloscope numérique portatif *non relié à la terre*
- Documentation sur les communications série asynchrones, en particulier les liaisons RS232 et RS485
- Documentation sur le code ASCII
- Pour une meilleure compréhension, il est recommandé d'effectuer les calculs demandés *sans calculatrice*.

Liaison RS 232

Préparation (50 minutes)

D'après le document-ressource « Communications séries asynchrones RS232/RS485 » disponible, répondre aux questions ci-dessous.

Lorsqu'aucun caractère n'est émis sur une ligne de communication asynchrone, quel est l'état <i>logique</i> de la ligne ?	
Combien de conducteurs sont nécessaires pour échanger 2 signaux (émission + réception) en RS232 ?	
A quelles tensions correspondent les niveaux logiques 0 et 1 sur une ligne RS232 ?	
Quelles est l'unité de vitesse de transmission de données sur une ligne série (débit binaire) ?	
Quelle est la durée d'un bit de donnée à une vitesse de 4800 bps ?	
Combien de bits sont nécessaires pour transmettre un mot en protocole « 8 bits de données / sans parité / 1 stop » ?	
Quelle est la durée de transmission d'un mot à ce même protocole ?	
Quel est le sens du front de signal qui apparaît au début d'un caractère en RS232 : montant ou descendant ?	
Le connecteur RS232 du châssis de l'ordinateur est-il Mâle ou Femelle ?	

Manipulations

- Configurer une application de terminal de données :
 - Démarrer le micro-ordinateur et votre session Windows
 - Lancer l'application Hyper Terminal
 - | Si non disponible sur le poste, voir les instructions spécifiques à l'établissement
 - Configurer le port COM1 en 9600 bps / 8 bits / sans parité / 1 stop
 - Configurer la communication avec « Aucun contrôle de flux »
 - Configurer l'écho local actif pour afficher les caractères ASCII saisis au clavier
- Préparer un câble de mesure RS232 :
 - Choisir un connecteur de cordon RS 232 9 broches adapté (M/F)
 - Repérer les bornes associées aux signaux TxD, RxD, Gnd
 - Y connecter (souder) les conducteurs du câble, respectivement Rouge, Vert, Noir
 - Brancher ce cordon sur l'ordinateur
- Capturer les signaux RS 232 :
 - Relever la tension présente sur la ligne en l'absence de transmission.
 - Vérifier la conformité de cette tension avec le standard utilisé.
 - Paramétrer l'oscilloscope pour capturer un événement non périodique selon les caractéristiques RS232
 - | Cf. la fiche méthode « Réglage oscilloscope pour capture d'événement non périodique »
 - Émettre des caractères à partir du terminal
 - Relever sur l'oscilloscope le signal TxD pour 1 caractère.
 - Imprimer ces oscillogrammes
- Analyser les signaux RS 232 :
 - Relever la durée d'un bit d'information
 - Identifier les différents bits de la trame (Start, b0..b7, Parité, Stop) et relever les valeurs des différents bits de données.
 - Reconstituer le mot binaire (octet) transmis en alignant les bits b0..b7, poids faible à droite.
 - Transformer cette valeur binaire en hexadécimal en séparant les bits en quartets (groupes de 4 bits) à partir de la droite et en convertissant chaque quartet en hexadécimal (0..9, A..F)
 - Retrouver le caractère émis d'après la table ASCII
- Jeu
 - Faites saisir un caractère par un camarade sur le terminal sans regarder, et devinez quel caractère a été frappé par l'analyse de la trame à l'oscilloscope.
- Connecter les ordinateurs 2 à 2 par la liaison RS232.
 - Vérifier le bon fonctionnement de la communication
 - Lancer un *chat* (« t'chat' ») rudimentaire entre les 2 postes à travers les 2 applications de terminal.



Liaison RS 485

Préparation, d'après la documentation mise à votre disposition :

Combien de conducteurs sont nécessaires pour échanger 2 signaux en RS485 Full-duplex ?	
Combien de conducteurs sont nécessaires pour échanger 2 signaux en RS485 Half-duplex ?	
A quelles tensions correspondent les niveaux logiques 0 et 1 sur une ligne RS485 ?	
Quels moyens permettent de contrôler le flux d'informations sur une ligne de transmission RS485 ?	
Quels sont les supports d'entrée et de sortie de données sur une application de terminal ?	
Un micro-ordinateur aura-t-il une connexion de type DTE ou DCE ?	

Manipulations

- Mettre en œuvre le convertisseur RS232 / RS485 :
 - Raccorder les signaux Tx+ / Tx- / Rx+ / Rx- conformément à une liaison RS485 Half-duplex 2 fils
 - Brancher l'adaptateur d'alimentation secteur pour fournir l'énergie des signaux RS485
- Capturer les signaux RS 485 :
 - Relever et vérifier la conformité de la tension présente sur la ligne en l'absence de transmission
 - Paramétrer l'oscilloscope pour capturer un événement non périodique selon les caractéristiques RS485
 - Émettre des caractères à partir du terminal
 - Relever les signaux sur l'oscilloscope
 - Imprimer éventuellement ces oscillogrammes
- Analyser les signaux RS 485 :
 - Relever la durée d'un bit d'information
 - Relever les valeurs des différents bits de la trame
 - Reconstituer le mot binaire (octet) transmis
 - Retrouver le caractère émis d'après la table ASCII

Sommaire

Objectifs.....	1
Contexte.....	1
Mise en situation.....	1
Analyse de documentations.....	2
Étude du protocole ModBus.....	2
Préparations de trames.....	2
Manipulations.....	3
Connexion matérielle.....	3
Recherche du protocole de l'esclave à interroger.....	3
Installation d'un utilitaire ModBus (si non pré-installé).....	3
Envoi automatique de trame Modbus	3
Envoi manuel de trame ModBus.....	3

Objectifs

- Être capable d'interroger un dispositif esclave ModBus à partir d'un ordinateur de type PC et d'une application de scanner ModBus maître.
- Mettre en évidence les trames de communication afin d'être capable de diagnostiquer des défauts de communication lors d'opérations de mise en œuvre ou de maintenance

Contexte

Prendre connaissance des systèmes communicants disponibles dans la salle d'essais de systèmes, et affecter un binôme d'apprenants sur chaque poste.

Mise en situation

Vous disposez du matériel suivant :

- Micro-ordinateur sous Windows XP
 - disposant d'une liaison série Sub-D Canon 9 broches
- Adaptateur RS232-RS485 Roline
 - + son adaptateur d'alimentation secteur
 - + 1 câble RS485 2 conducteurs Rouge/Vert
- Oscilloscope numérique portatif *non relié à la terre*
- Documentation sur les communications série asynchrones, en particulier les liaisons RS232 et RS485
- Documentation sur le code ASCII
- Documentation du dispositif communicant installé sur le système étudié.

Choisissez la grandeur physique que vous désirez recueillir sur le système étudié :

- Sur compteur d'énergie : Tension V1, V2, V3, U1, U2, U3, ...
- Sur régulateur : Mesure de température, consigne, paramètres P, I, D, ...

Analyse de documentations

- ◆ Relever dans la documentation constructeur des dispositifs communicants :
 - ✓ la connectique de communication série disponible (borniers, brochage, précautions, ...),
 - ✓ le mode de configuration du protocole de communication (débit, éventuels décalages d'adresses, ...)
 - ✓ la table des grandeurs accessibles par le réseau (ou un extrait) : V1/V2/V3/U1/U2U/U3/T/...
 - ✓ l'adresse de lecture pour la grandeur désirée sur le système.

Étude du protocole ModBus

Combien de dispositifs maîtres peuvent être présents sur un réseau au protocole ModBus ?	
Combien de dispositifs esclaves peuvent être présents sur un réseau au protocole ModBus ?	
Combien de dispositifs peuvent être présents sur un réseau avec support physique de type RS485 ?	
Lorsque le maître émet une requête (question) sur le bus, combien d'esclaves peuvent répondre ?	
Quels sont les 2 modes de transmission du protocole ModBus ?	
Quels sont les différents champs d'informations transmis sur une trame de requête ModBus RTU ?	
Combien de bits sont nécessaires pour définir en binaire un numéro d'esclave comprise entre 1 et 247 ? Donc combien d'octets ?	
Combien de caractères sont nécessaires pour envoyer le numéro d'esclave sur une requête ModBus en mode RTU ?	
Combien de caractères sont nécessaires pour envoyer le numéro d'esclave sur une requête ModBus en mode ASCII ?	
Quel est le numéro de fonction ModBus permettant de lire la valeur d'un mot dans un registre interne du dispositif interrogé ?	
Combien de caractères sont nécessaires pour coder une adresse 16 bits en mode RTU ?	
Convertir l'adresse 30 004 ₍₁₀₎ en binaire, puis en hexadécimal	

Préparations de trames

Vous disposer d'un utilitaire nommé « CRC16.exe » qui calcule le code CRC16 de vérification de trame qui est placé à la fin des différents caractères envoyés selon le protocole ModBus.

- ◆ A partir de la documentation du protocole ModBus, établir les trames (liste de caractères) qui permettront d'établir les questions ModBus suivantes :
 - ✓ Lire, depuis l'esclave présent à l'adresse 1, les 2 mots des registres d'adresse 30004 ;
 - ✓ Lire, depuis l'esclave présent à l'adresse 10, les 7 bits internes d'adresse 48 ;
 - ✓ ...

Manipulations

Connexion matérielle

- ◆ Connecter l'ordinateur au système esclave à travers l'adaptateur RS232/RS485.

Recherche du protocole de l'esclave à interroger

- ◆ Accéder à la configuration du dispositif à interroger, et relever les différents paramètres de communication :
 - ✓ Débit (bps), nb de bits de données, parité, nb de bits de stop, contrôle de flux, ...
- ◆ A partir de la grandeur physique à lire dans le dispositif esclave (U1, T, ...), retrouver dans la documentation l'adresse du registre Modbus correspondant.

Installation d'un utilitaire ModBus (si non pré-installé)

- ◆ Rechercher le site internet du fournisseur de l'application de supervision DAQFactory
- ◆ Télécharger et installer DAQFactory en version d'évaluation 25 jours.
- ◆ Rechercher dans les menus Programmes l'utilitaire de scan « QuickMod ModBus Scanner », lancer cet utilitaire.
- ◆ Configurer la communication en ModBus RTU sur port série, selon le protocole relevé sur l'esclave (vitesse, parité). Le contrôle de flux sera défini à :
 - ✓ Numéro de port COM : commence à 1 sur un PC (1^{er} : COM1, 2^e : COM2, ...)
 - ✓ Type : Manuel
 - ✓ RTS Control Toggle
 - ✓ DTR : Enable

Envoi automatique de trame Modbus

- ◆ Définir les paramètres de la trame à envoyer :
 - ✓ « ID » : adresse de l'esclave à interroger en décimal
 - ✓ « Fonction » : lecture de bit internet / de bit d'entrée / de registre interne / ...
 - Si registre codé en virgule flottante (format IEEE) → « Float »
 - Si registre codé en décimal (×100 par exemple) → « Unsigned Integer »
 - ✓ « Type » : le type d'information retournée par le dispositif (selon le registre sélectionné)
 - ✓ « Tag » : l'adresse du premier registre à lire
 - ✓ « # pts » : nombre de registres successifs à lire
 - ✓ « Scan rate » : la période d'interrogation en secondes
- ◆ Relever à l'oscilloscope l'allure des trames échangées sur le réseau RS485.

Envoi manuel de trame ModBus

- ◆ Cliquer sur le bouton « Monitor » pour afficher le terminal d'émission/réception
 - ✓ Cocher la case « Display all chars as ASCII codes » afin que le moniteur ne transforme pas les codes ASCII affichables en caractères.
 - ✓ Cocher la case « Display in Hexa format » afin d'afficher les codes en hexadécimal. Chaque code Hexa est précédé des 2 caractères « \x ».
- ◆ Préparer (sur le papier) une trame de requête de votre choix
- ◆ Saisir cette requête dans le champ « Output string » et l'envoyer en cliquant sur « Send ».
 - ✓ Veillez à saisir les codes en hexadécimal, chacun précédé des caractères « \x ».
- ◆ Analyser la réponse reçue en hexadécimal et en déduire la valeur lue.
- ◆ Relever à l'oscilloscope l'allure *d'une* trame de requête échangée sur le réseau RS485. Analyser les différents caractères pour identifier la question posée
- ◆ En binôme, établir et envoyer une requête par l'un des étudiants sans la montrer au camarade. Le second étudiant devra retrouver la question posée par l'analyse du relevé à l'oscilloscope.

Sommaire

Objectifs.....	1
Mise en situation.....	1
Préparation de l'esclave.....	2
Connexion physique maître-esclave.....	2
Paramétrage de la communication.....	2
Étude des modes de communication.....	2
Programmation du Maître.....	3
Mesures électriques.....	3

Objectifs

- Être capable de paramétrer un esclave Modbus pour être interrogé par un API maître
- Être capable de configurer un API afin d'être le maître de l'esclave pré-cité
- Être capable de programmer un API Twido en Modbus Maître pour une unique requête répétée
- Être capable de programmer un API Twido en Modbus Maître pour une séquence répétée de plusieurs requêtes
- Être capable de diagnostiquer les problèmes de communications sur cette liaison Modbus maître-esclave

Mise en situation

Vous disposez du matériel suivant :

- Un dispositif industriel communicant en Modbus esclave (compteur d'énergie, régulateur, variateur de vitesse, ...)
- Un Automate Programmable Industriel de type TWIDO de Schneider-Electric
 - 1 option RS485 sur connecteur miniDin (TWDNAC485D) ou sur bornier 3 contacts (TWDNAC485T)
 - 1 cordon miniDin 9 broches (pour option RS485 à miniDin) vers fils nus pour bornier du dispositif esclave Modbus
- Micro-ordinateur sous Windows XP
 - disposant d'un cordon de programmation du Twido (Série RS232 ou USB)
- Oscilloscope numérique portatif *non relié à la terre*
- Documentation sur les communications série asynchrones, en particulier les liaisons RS232 et RS485
- Documentation sur le code ASCII
- La documentation-constructeur « Manuel de mise en œuvre logicielle TWIDO »

Préparation de l'esclave

Accédez à la documentation-constructeur « Manuel de mise en œuvre logicielle TWIDO ».

Connexion physique maître-esclave

- Vous disposez d'un cordon mini-Din 8 broches vers 8 conducteurs nus.
- Retrouver dans la documentation constructeur de l'esclave les bornes de branchement de la liaison Modbus
- Retrouver, dans le guide de mise en œuvre logicielle du TWIDO, le brochage de la prise miniDin de communications RS485.
- Noter les positions des 2 bornes de données RS485 : A / B ou RxTx+ / TxTx- ou D1 / D0
- Identifier sur le cordon les conducteurs nus (couleurs) correspondant aux 2 bornes de données du miniDin
- Connecter l'API au dispositif Modbus esclave à travers ce cordon.
Note : Attention à ce que les conducteurs dénudés non utilisés ne se touchent pas : ils mettraient l'alimentation 5V de l'API en court-circuit !

Paramétrage de la communication

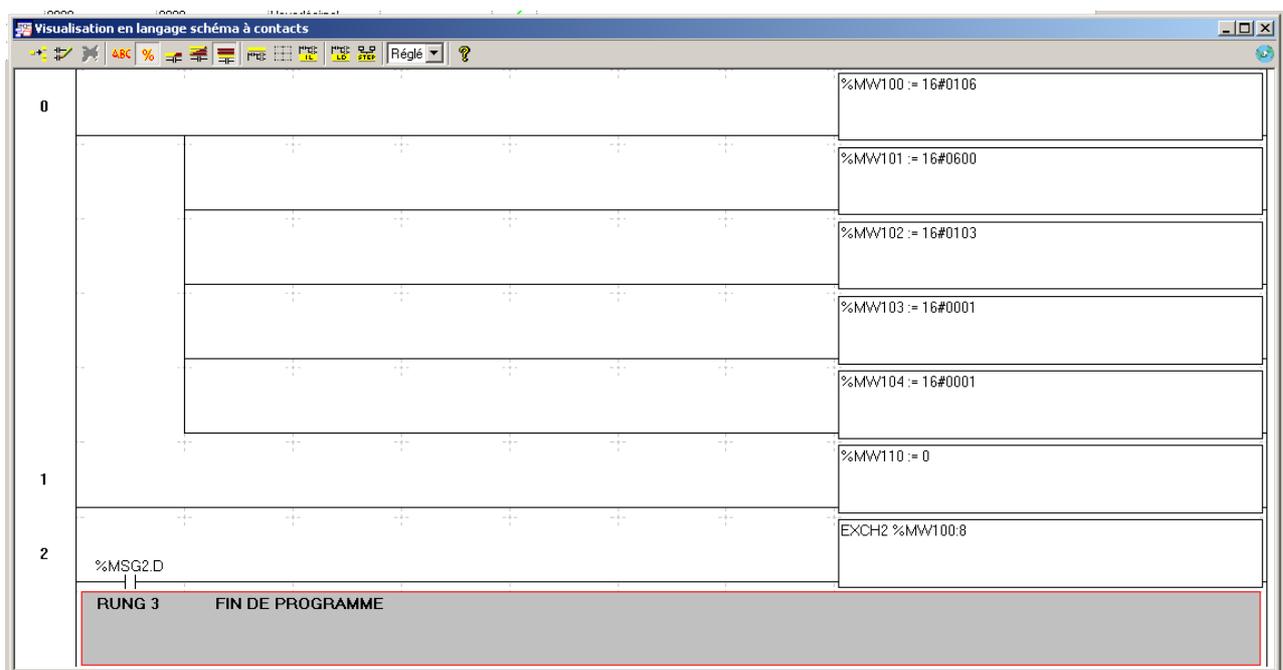
- Relever la configuration Modbus de l'esclave : vitesse, parité, stop, ...
- Déterminer l'adresse d'un registre de votre esclave dont l'information est significative dans votre esclave (tension V1, mesure de température, consigne de vitesse, ...)
- Configurer l'ensemble de l'API Twido, et notamment le port 2 pour communiquer avec l'esclave.
 Consulter les documents pédagogiques disponibles pour configurer un nouveau projet TWIDO.

Étude des modes de communication

Quels sont les 3 supports physiques de communication disponibles sur l'ensemble de la gamme Twido, options comprises ? (Consulter la documentation <u>constructeur</u> disponible ou en ligne)	
Quels sont les 3 protocoles de communication disponibles sur un port RS485 sur l'ensemble de la gamme Twido ?	
Quels sont les 2 modes de fonctionnement Modbus RS485 disponibles sur un Twido ? Quel est le plus performant en vitesse ?	
Quelle instruction du programme permet d'envoyer une requête Modbus maître ?	
Quel paramètre 'x' est fourni attaché à la fonction EXCH ?	
Quel type d'objet mémoire doit être associé à l'utilisation de cette fonction ?	
Que représente l'adresse de cet objet mémoire ?	
Quelles sont les 3 zones (tables d'informations/ tables de données) qui composent cet zone mémoire ?	

• **Programmation du Maître**

Préparer la trame Modbus correspondant à la demande de lecture du registre de votre esclave. (Typiquement 2 octets + 2 mots, CRC non compris)	
Préparer un tableau 3 colonnes (adresse, octet de poids fort, octet de poids faible), minimum 8 lignes (8 mots consécutifs), pour contenir les 3 tables associées à la fonction EXCHx.	
Remplir la table d'émission avec la trame préparée. Compléter la table de contrôle	
Choisir l'adresse de l'automate qui contiendra cette table. (Typiquement, on pourra choisir %MW100)	
Compléter/corriger le programme ci-dessous pour remplir les tables de contrôles et d'émission.	
Introduire et tester le programme pour recevoir l'information attendue.	



Mesures électriques

- Relever à l'oscilloscope l'allure des échanges de trames entre maître et esclave.
- Noter les éventuels écarts d'amplitude entre les trames requêtes maîtres et les trames réponses esclaves.
-

Objectif

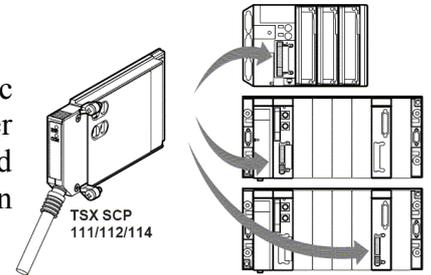
Générer une trace, un historique d'événements sur support papier, par l'envoi de messages « au fil de l'eau » vers une imprimante série ou munie d'un adaptateur parallèle / série.

Le matériel nécessaire

L'automate

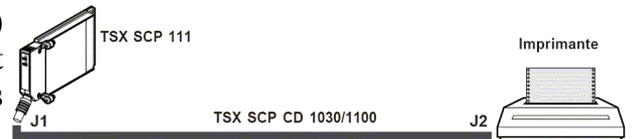
L'expérience est ici basée sur un API Schneider-Electric Telemecanique Premium TSX-57-3623, mais peut être tout API Schneider Premium TSX-57 ou Micro TSX-37 disposant d'un emplacement PC-Card (ex-PCMCIA) et programmable sous PL7-Junior/Pro, de préférence en langage structuré.

Les TSX-57 disposent pour la plupart d'un emplacement PC-Card sur le module CPU, mais on utilisera ici un module de communication additionnel TSX-SCY-21601, pouvant recevoir des cartes PCMCIA sur la voie 1.



L'imprimante

La manipulation est basée ici sur une (vieille) imprimante matricielle à impact, à port parallèle, qui fait suffisamment de bruit pour intriguer les étudiants et les inciter à envoyer des données dessus.



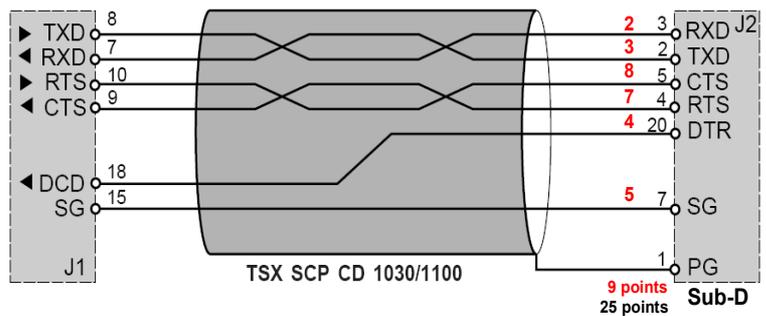
Les adaptateurs de communications

La communication entre l'API et l'imprimante série est assurée par une carte de communication RS-232 de référence TSX-SCP-111 associée à un câble TSX-SCP-CD-1030

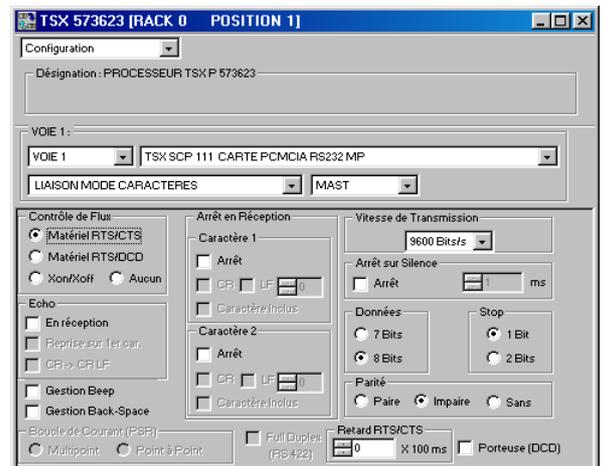
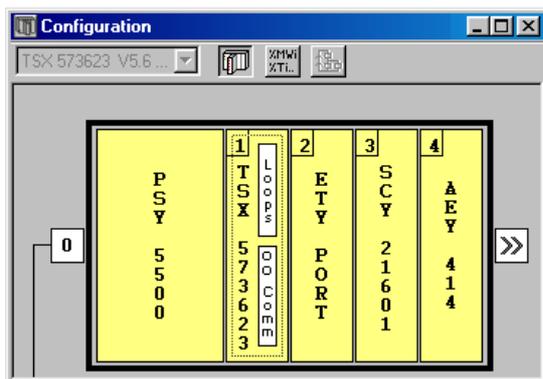
Un adaptateur série/ parallèle, auto-configurable, alimenté par pile 9V, fourni chez Radiospares sous la référence 215-167, est connecté sur le port Centronics de l'imprimante parallèle, celle-ci étant considérée ainsi comme une imprimante série RS232.



Raccordements



Configuration



Programmation

Instructions de base :

```
(*Remise a zero du tampon d'impression
%MW50 : Début du tampon d'impression
%MW98 : Index de pointage dans la table de mots
0..39 : longueur du buffer : 40 mots = 80 caractères
32 : Code ASCII du caractère [Espace] *)

FOR %MW98:=0 TO 39 DO
  %MW50[%MW98]:=32;
END_FOR;

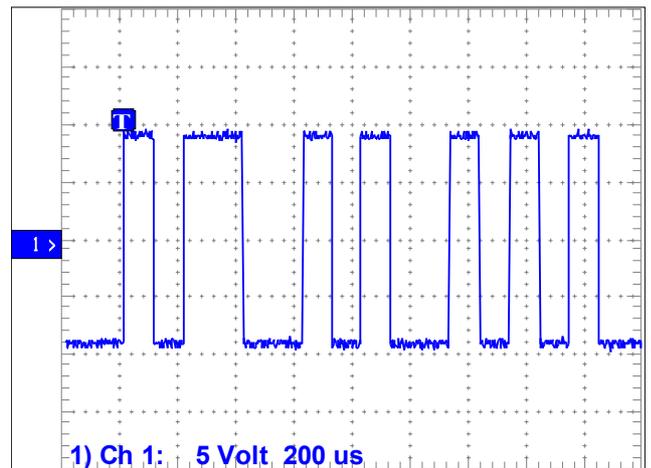
(*procedure envoi à l imprimante
%MW250 : Contrôle de communication (4 mots)
%MW253 : Dernier mot de contrôle = longueur de chaîne à émettre *)
(*)
%M250 : Copie du bit d'état de communication
%M11 : Bit de commande de l'envoi, doit être mis à 1
%MB100 : Début du tampon d'impression = %MW50
*)
%M250:=NOT %MW250:X0;
IF %M250 AND %M11 THEN
  %MW90:=16#0D0A;
  %MW253:=82;(* nbre de caractères à émettre*)
  PRINT_CHAR(ADR#{0}1.1.SYS,%MB100:82,%MW250:4);
  RESET %M11;
END_IF;
```

Mesures de signaux

Ici le caractère Y, poids faible en tête après le Start à 0 :

0100110101 → 01011001₍₂₎ → 59₍₁₆₎ → « Y »

suivi d'un autre caractère (partiel).



Présentation

A.S.C.I.I. est l'abréviation de American Standard Code for Information Exchange. Ce codage consiste à associer une valeur numérique binaire (interprétable en hexadécimal, décimal, ...) à chacun des caractères utilisables dans l'échange de données informatique : caractères alphabétiques et numériques (alphanumérique), ponctuation, codes de contrôles divers.

Différentes variantes du code ASCII sont disponibles pour différentes langues. Il existe même une version Extended de ASCII où le 8^{ème} bit de données est utilisé, ce qui permet de distinguer 2 fois plus de caractères, notamment les caractères accentués pour le français. On exprime ce sigle ASCII par le son « aski ».

Table de caractères

Codes hexadécimaux

F f	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	NP	CR	SO	SI
1.	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2.		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Del

Utilisation et conversions

Le caractère « A » a comme premier digit hexadécimal un « 4 » (colonne de gauche) et comme second digit hexadécimal un « 1 » (ligne du haut). Son code ASCII hexadécimal est donc la valeur hexa 41(h).

Ce même code ASCII en décimal donne :

Conversion en décimal :

Les puissances de 16		16 ¹		16 ⁰	
valent		16		1	
associées à	×	4		1	
donnent	=	64	+	1	= 65

Conversion en binaire :

Chaque digit		4		1	
est converti en binaire		0100		0001	

Interprétation

Lorsqu'une donnée est visualisée, comme dans une table d'animation d'automate programmable, celle-ci peut être affichée et interprétée selon différents codages, indépendamment de son utilisation : en décimal, en hexadécimal, en binaire ou en caractère ASCII.

Caractères spécifiques de contrôle

<i>Nom</i>	<i>Commande</i>	<i>Action</i>
Commandes de format		
CR	Carriage return	Retour chariot : retour en début de ligne
LF	Line feed	Avancer d'une ligne : passage à la ligne suivante
BS	Backspace	Espace arrière : suppression du caractère précédent
HT	Horizontal tabulation	Tabulation horizontale : déplacement dans la ligne pour aligner le texte qui suit
VT	Vertical tabulation	Tabulation verticale
SP	Space	Espace
FF	Form feed	Avancer d'une feuille : passer à la page suivante
Extension de code		
SO	Shift out	
SI	Shift in	
ESC	Escape	Début de séquence d'échappement
Commande de séparation		
FS	File separator	
GS	Group separator	
RS	Record separator	
US	Unit separator	
EM	End of medium	
Commandes de communication synchrone		
SOH	Start of header	ACK Positive acknowledge
STX	Start of text	NAK Negative acknowledge
ETX	End of text	SYN Synchronisation
EOT	End of transmission	DLE Data link escape
ETB	End of transmission block	NUL Null
ENQ	Enquiry	
Commandes de périphérique		
DC1	Device control 1	DC3 Device control 3
DC2	Device control 2	DC4 Device control 4
Commandes diverses		
CAN	Cancel	
SUB	Substitute	
DEL	Delete	Supprime le caractère qui suit
BEL	Bell	Émet un « bip » ou un autre avertissement sonore

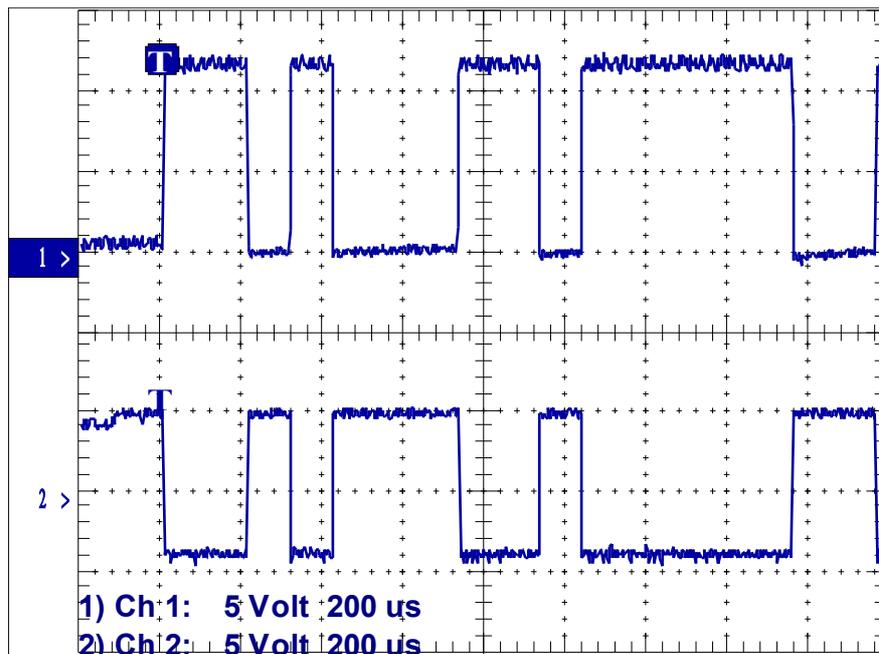
Quelques liens

<http://fr.wikipedia.org/wiki/ASCII>

http://www-clips.imag.fr/commun/bernard.cassagne/Introduction_ANSI_C/node135.html

Sommaire

- Présentation.....2
- Technologie des communications asynchrones.....3
 - Protocole.....3
 - Vitesse de transmission.....3
 - Glossaire.....4
 - Communications normalisées.....4
- Liaison RS232C.....5
 - Niveaux des signaux.....5
 - Signaux de données.....5
 - Liens.....5
 - Contrôle de flux.....5
 - Contrôle de flux matériel.....5
 - Contrôle de flux logiciel.....5
 - Signaux de contrôle et d'état de modem.....6
 - Connectique.....6
 - Raccordements.....6
- Liaisons RS422, RS485.....7
 - Nature des signaux.....7
 - Communication SIMPLEX : un seul sens de communication.....7
 - Communication Full-Duplex : 2 sens de communication simultanés.....7
 - Communication Half-Duplex : 2 sens de communication alternés.....7
 - Connectique.....8
 - Allure des signaux.....8



Présentation

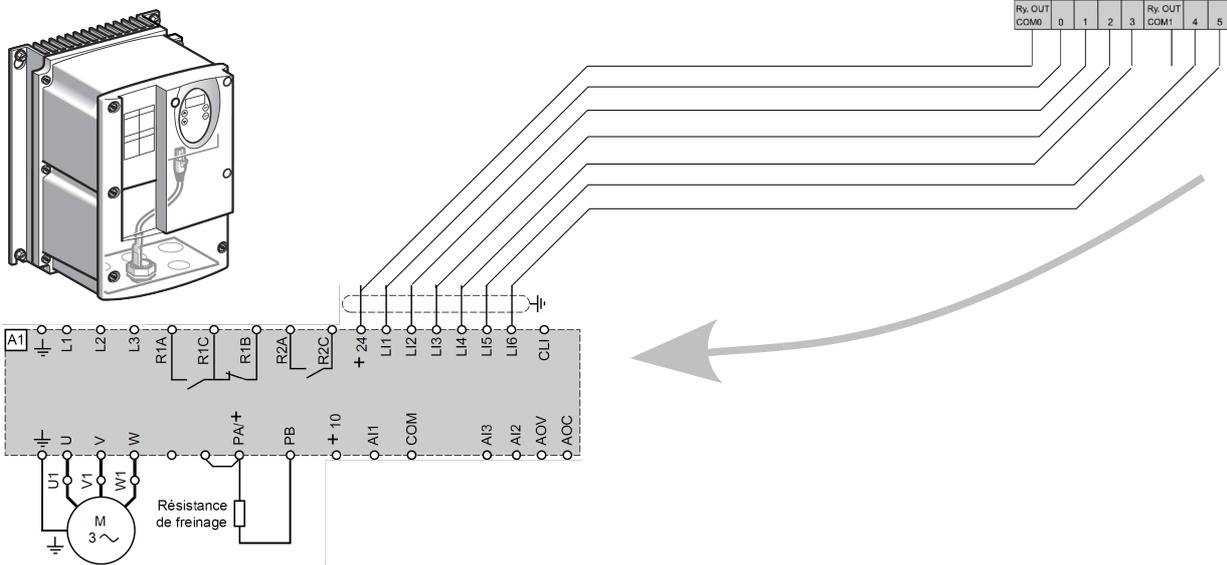
Les « liaisons séries » sont des moyens de transport d'informations (communication) entre divers dispositifs numériques industriels ou tertiaires. On les oppose aux liaisons parallèles qui nécessitent au moins un conducteur par bit à transmettre.

Les liaisons parallèles nécessitent de mettre en œuvre au moins un conducteur par information, plus un conducteur pour la référence de potentiel. Par exemple, pour qu'un automate industriel envoie à un variateur de vitesse :

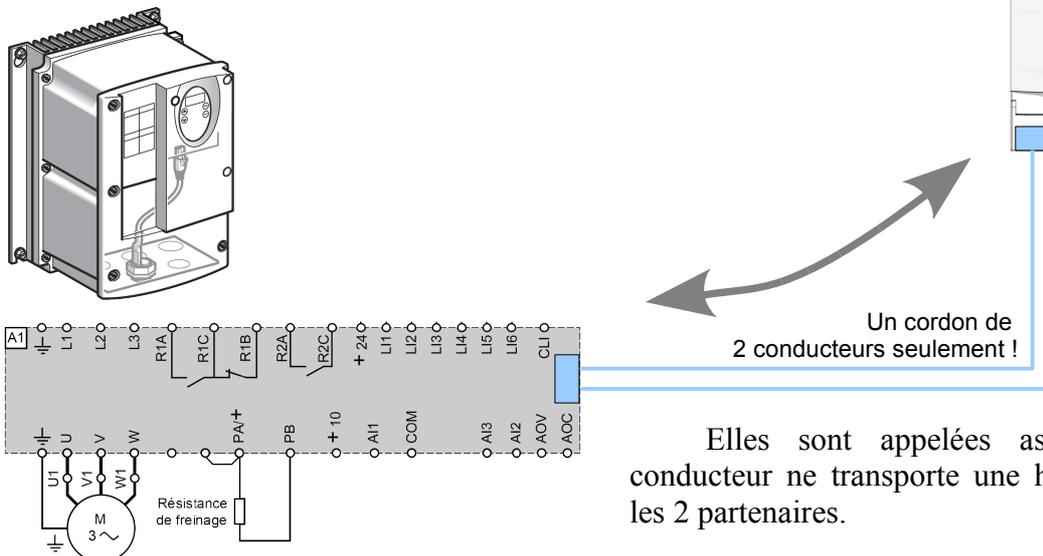
- 1 commande de marche (TOR)
 - 1 commande d'arrêt (TOR)
 - 1 commande de déverrouillage en cas de défaut (TOR)
 - 1 sélection de 8 vitesses préprogrammées (codées sur 3 bits)
 - ...
- il faudra au moins $1+1+1+3+ 1$ commun = **7 conducteurs**.

S'il existe un **autre variateur**, ce sera **7 conducteurs supplémentaires**.

Pour que l'automate récupère des informations du variateur, nouveaux conducteurs.



Dans une liaison série, les différents bits ordres TOR et valeurs numériques (8 bits, 16 bits, ...) ne sont pas échangés en parallèle mais les uns après les autres dans le temps, ce qui limite le nombre de fils de transmission à 2 conducteurs.



Elles sont appelées asynchrones car aucun conducteur ne transporte une horloge commune entre les 2 partenaires.

Technologie des communications asynchrones

Protocole

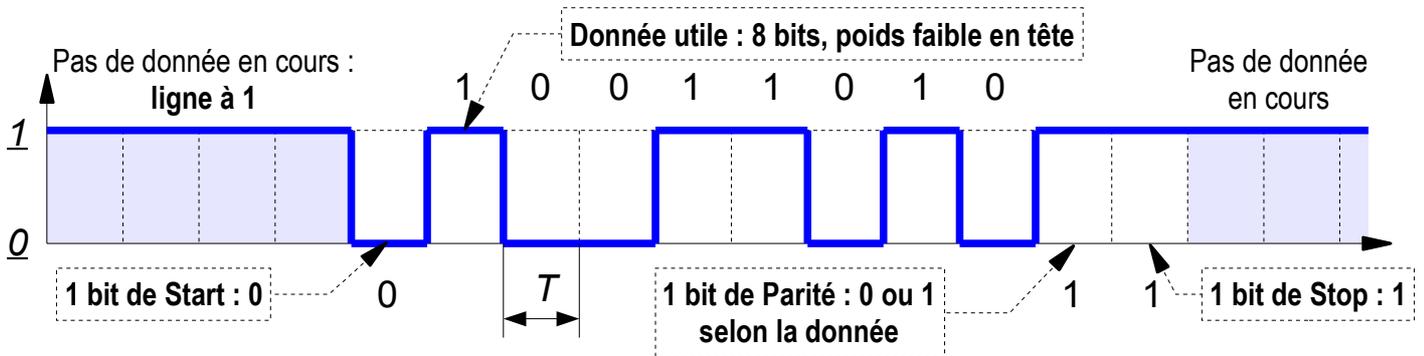
Le protocole d'échange asynchrone est défini par l'envoi, pour chaque caractère émis, de :

- un bit de **Start**,
- les 5 à 8 bits de **données**, poids faible en tête,
- éventuellement, un bit de vérification de **Parité** qui permet de détecter des erreurs de transmission des 8 bits de donnée sur la ligne,

Le bit de parité est mis à 1 si le nombre de bits de données est pair. Ainsi, si un bit de données est erroné durant la transmission, le comptage du nombre de bits à 1 à la réception mettra en évidence une non concordance avec le bit de parité, et donc une erreur de transmission. Si 2 bits sont erronés, la vérification de parité n'a plus d'effet.

- 1, 1½, ou 2 bits de **Stop** après.

Lorsqu'aucun caractère ne circule sur la ligne, celle-ci reste à l'état logique haut (« 1 »).



Note : Ce chronogramme représente l'état **logique** AVANT la mise en forme en tension par l'adaptation de ligne, c'est à dire indépendamment du standard RS232, 422 ou 485 dont les niveaux de tension sont définis plus loin.

Vitesse de transmission

La vitesse de transmission représente la quantité d'informations qui peuvent être transportées pendant un certain temps. Elle est exprimée en bits par seconde (bps).

Les vitesses de transmission peuvent être entre autres :

Vitesse en bits par seconde (bps)	Application
75	Émission Clavier Minitel → Serveur Télétex
110, 300, 600	
1200	Réception Serveur Télétex → Écran Minitel
2400	
4800, 9600, 14400, 19200	API, Modem-Fax (14400)
56000, 115200, 128000, 256000	
187,5 kbps , 1,5 Mbps, 10 Mbps	Bus de terrain : Profibus, ...

Si la vitesse est de 9600 bits par seconde, le temps de présence d'un bit dans la trame (durée du bit) est de 1/9600 seconde soit 104 µs.

Glossaire

Unipolaire	Chaque signal électrique est référencé par rapport à une masse unique. Pour 7 signaux échangés, on aurait besoin de $7 + 1 = 8$ conducteurs.
Différentiel	Chaque signal électrique est transporté entre 2 conducteurs, chacun a donc sa référence. Pour 4 signaux échangés, on aurait besoin de $2 \times 4 = 8$ conducteurs.
DTE	« Data Terminal Equipment » : un équipement terminal de données est typiquement un ordinateur qui peut envoyer des données (depuis une application ou un clavier) et recevoir des données (vers une application ou l'écran).
DCE	« Data Communication Equipment » : un équipement de communication de données ne génère aucune donnée mais convertit leur niveau électrique, typiquement c'est un Modem.
Mark	Niveau logique haut = « 1 »
Space	Niveau logique bas = « 0 »

Communications normalisées

Les différents types d'interface sont couramment désignés par le numéro de l'avis ou de la norme qui les définissent :

RS...	correspond aux normes américaines définies par l'EIA (Electronics Industries Association).
V... ou X...	correspond aux avis internationaux définis par le CCITT (Comité Consultatif International pour le Téléphone et les Télécommunications).
Boucle de courant	Particulièrement utilisée dans l'industrie, ne correspond pas à une norme.

<i>EIA CCITT</i>	<i>RS 232 V24 / V28</i>	<i>RS 423</i>	<i>RS 422 V11 / X27</i>	<i>RS 485 V11 / X27</i>	<i>Boucle de courant</i>
Type d'interface	Unipolaire	Unipolaire	Différentiel	Différentiel	0-20 mA
Sensibilité					
Distance (m)	15	1200	1200	1200	1000 à 2000
Débit max. (bps)	19200	100 K	10 M sur 100 m 100 k sur 1200 m	10 M sur 100 m 100 k sur 1200 m	19200
Multipoint	non	oui	oui	oui	oui
Nombre d'émetteurs	1	1	1	32	
Nombre récepteurs	1	10	10	32	
Niveau de sortie non chargé (V)	± 25	± 6	± 6	± 6	
Niveau de sortie pleine charge (V)	± 5 à ± 15	$\pm 3,6$	± 2	$\pm 1,5$	
Niveau de sortie typique	$\pm 12V$			$\pm 5V$	
Impédance d'entrée	3 à 7 k Ω	$\geq 4k\Omega$	$\geq 4 k\Omega$	$\geq 12 k\Omega$	
Charge émetteur	3 à 7 k Ω	$\geq 450 \Omega$	100 Ω	54 Ω	

Vocabulaire :

Non chargé	La connexion est « en l'air », le fil n'est pas branché
Chargé	Il y a connexion entre l'émetteur et le récepteur, donc impédance de charge sur la ligne

Liaison RS232C

L'évolution temporelle des signaux RS232 est conforme aux signaux de liaison asynchrone décrits précédemment. La spécificité de RS232 tient dans l'adaptation en tension des signaux afin d'être transmis sur une distance supérieure (15m).

Niveaux des signaux

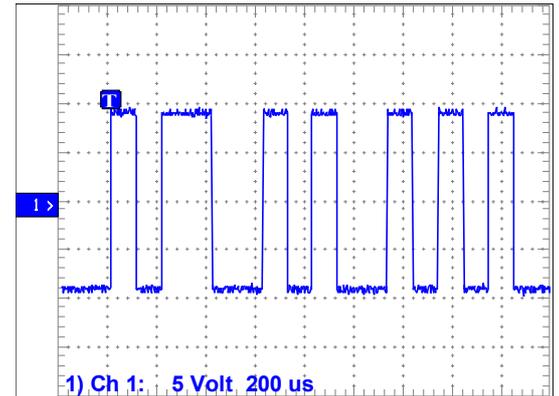
Niveau logique	Polarité	Intervalle de niveau électrique	Typique
'1'	Basse	entre -3V et -15 V	-12V
'0'	Haute	entre +3V et +15 V	+ 12V

On dit donc que l'on travaille en logique négative : « 1 » au niveau électrique bas, « 0 » au niveau électrique haut.

Signaux de données

Sur une liaison bidirectionnelle minimale sans contrôle de flux, il faudra 3 conducteurs :

Tx	Transmitted Data	Conducteur d'émission des données
Rx	Received Data	Conducteur de réception des données
Gnd	Ground	Conducteur de masse du signal



Liens

Normes : http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html

Convertisseurs : http://bb-elec.com/technical_library.asp

Schémas : http://sjeffroy.free.fr/Divers/Convertisseur_RS232-RS485/convertisseur_rs232-rs485.html

Contrôle de flux

Les équipements connectés pour un échange de données (communication) peuvent ne pas traiter les informations à la même vitesse. Si le récepteur est plus rapide que l'émetteur, aucun problème n'apparaît.

Si l'émetteur travaille plus vite que le récepteur, des données peuvent être perdues. Il faut donc mettre en place un contrôle de flux par des signaux appropriés.

Contrôle de flux matériel

Le contrôle de flux est assuré par la présence et la connexion de conducteurs supplémentaires entre le DTE et le DCE.

RTS	Request To Send	Ce signal est abaissé (« 0 ») pour préparer le DCE à accepter les données transmises. La préparation consiste à activer les circuits de réception, ou activer le canal dans les applications demi-duplex. Lorsque le DCE est prêt, il acquitte en abaissant CTS.
CTS	Clear To Send	Le signal est abaissé par le DCE pour informer le DTE que la transmission peut débuter.

Contrôle de flux logiciel

Le récepteur stoppe le flux de données en envoyant sur la ligne de données un caractère dédié nommé XOFF, et le relance en envoyant le caractère XON. D'où le nom du protocole XON/XOFF.

| Le caractère XON est le code ASCII DC1, XOFF est le code DC3, présentés plus loin dans la documentation ASCII.

Signaux de contrôle et d'état de modem

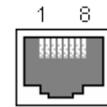
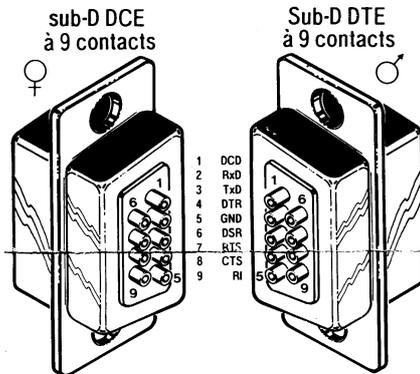
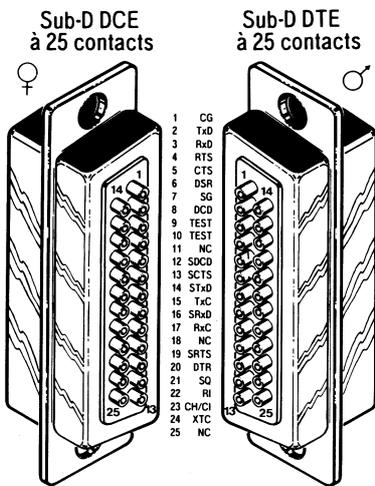
DSR	Data Set Ready	Si connecté à un modem : <ul style="list-style-type: none"> ✓ Le modem est connecté sur une ligne téléphonique saine ✓ Le modem est en mode Data, et non pas en mode voix ou numérotation ✓ Le modem est en train de générer une tonalité de réponse Si connecté à un autre dispositif : <ul style="list-style-type: none"> ✓ Le DCE est actif. Si non utilisé, doit être forcé à « 0 ».
DCD	Data Carrier Detect	Détection de porteuse de données sur la ligne
DTR	Data Terminal Ready	
RI	Ring Indicator	Signale une sonnerie d'appel téléphonique sur le modem

Connectique

Canon Sub-D 25 broches

Canon Sub-D 9 broches

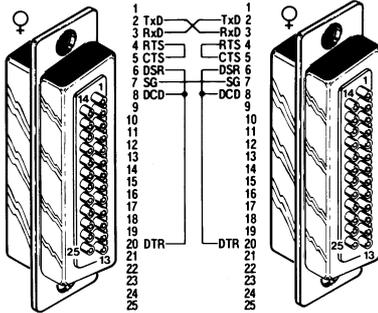
RJ45 (RS 232 D)



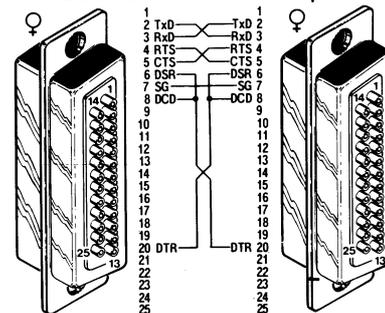
- 1 DSR/RI
- 2 CD
- 3 DTR
- 4 Gnd
- 5 RxD
- 6 TxD
- 7 CTS
- 8 RTS

Raccordements

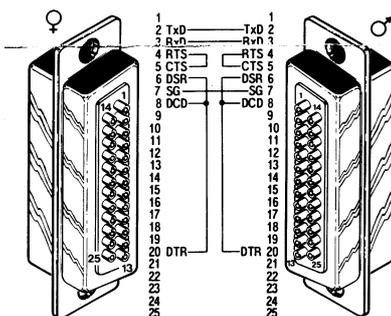
Interconnexion trifilaire DTE-DTE



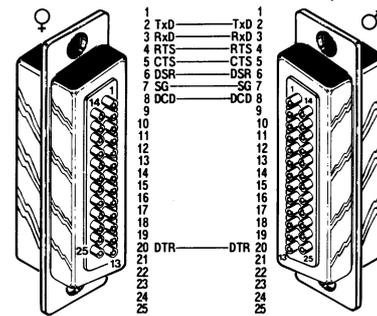
Interconnexion DTE-DTE complète



Interconnexion trifilaire DTE-DCE



Interconnexion DTE-DCE complète



Liaisons RS422, RS485

L'évolution temporelle des signaux RS422 et RS485 est identique aux signaux RS232 et liaison asynchrone décrits précédemment. La spécificité de RS422/485 tient dans l'adaptation en tension différentielle des signaux afin d'être transmis sur une distance supérieure (1200 mètres).

- Le niveau logique « 1 » est défini par une tension positive de l'ordre de + 5 V
 - Le niveau logique « 0 » est défini par une tension positive de l'ordre de – 5 V
- Contrairement au format RS232, RS485 travaille donc en logique positive.

Nature des signaux

Communication SIMPLEX : un seul sens de communication A → B

Sur une liaison unidirectionnelle (rarement utilisée), il faudra :

- le conducteur d'émission de données Tx+ de A qui va sur Rx+ de B : polarité positive
- le conducteur d'émission de données Tx – de A qui va sur Rx– de B : polarité négative
- le blindage

Communication Full-Duplex : 2 sens de communication simultanés

Sur une liaison bidirectionnelle (de type 4 fils), il faudra :

- les 2 conducteurs d'émission des données Tx+, Tx- de A → Rx+, Rx- de B
- les 2 conducteurs de réception des données Tx+, Tx- de B → Rx+, Rx- de A
- le blindage

Communication Half-Duplex : 2 sens de communication alternés : la plus utilisée

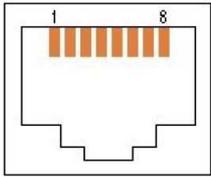
Sur une liaison bidirectionnelle alternée de type 2 fils, il faudra :

- le conducteur d'émission/ réception des données Tx+/Rx+ de A et B
- le conducteur d'émission/ réception des données Tx–/Rx– de A et B
- le blindage

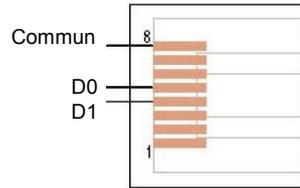
Connectique

Connecteur RJ45

Vue de face

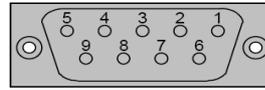


Vue de dessus

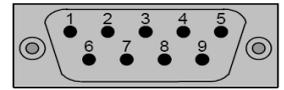


Connecteur Sub-D 9 broches

Femelle



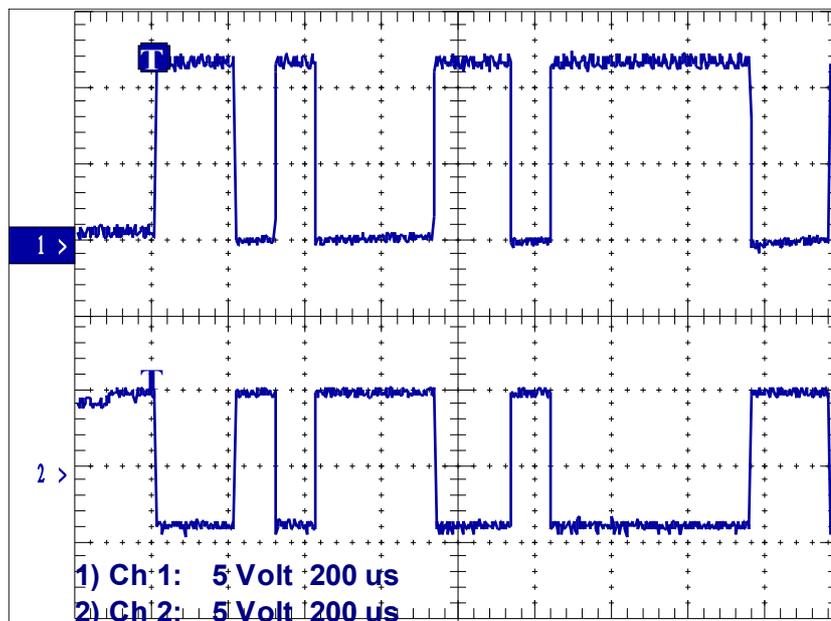
Mâle



RJ45	Sub-DB9	Exigence	2 fils		4 fils		Description
			IDv	EIA/TIA 485	IDv	EIA/TIA 485	
1	8	requis			RxD0	A'	Receiver terminal 0, potentiel Va'
2	4	requis			RxD1	B'	Receiver terminal 1, potentiel Vb'
3	3	optionnel	PMC		PMC		Port Mode Control
4	5	requis	D1	B/B'	TxD1	B	Transceiver terminal 1, potentiel V1/Vb
5	9	requis	D0	A/A'	TxD0	A	Transceiver terminal 0, potentiel V0/Va
7	2	recommandé	VP				Alimentation positive 5..24VDC
8	1	requis	Common	C/C'	Common	C/C'	Commun d'alimentation et de signal

Allure des signaux

En haut un signal RS232, logique négative



En bas la même trame convertie en RS485, logique positive

Notez bien l'origine des traces 1 et 2.

Présentation

HyperTerminal est un utilitaire livré avec Windows XP permettant de communiquer sur des liaisons distantes à travers les ports de communications série intégrés à l'ordinateur, et parfois sur de longues distances sur le réseau téléphonique commuté à travers un modem.

HyperTerminal a été retiré des distributions Windows Vista et Windows 7.

<http://windows.microsoft.com/fr-FR/windows-vista/What-happened-to-HyperTerminal>

<http://zebrablog.net/hilitec/index.php/2008/05/27/29-solution-replacer-hyperterminal-hyper-terminal-pour-windows-vista-7-alternative-tera-term-pro-professional-telecharger-download>

On peut alors **télécharger gratuitement** « HyperTerminal Private Edition » :

<ftp://ftp.hilgraeve.com/hpte/hpte63.exe>

Configuration de l'HyperTerminal sous Windows XP

- ◆ Ouvrir l'application HyperTerminal par le menu :

Démarrer | Accessoires | Communications | HyperTerminal

- ◆ Un dialogue propose d'utiliser HyperTerminal pour du réseau TCP/IP distant.

- ✓ Répondre Non



- ◆ Un dialogue demande de définir les paramètres géographiques de numérotation.

- ✓ Complétez l'indicatif régional.
- ✓ Validez par OK

- ◆ Un dialogue demande de définir un nom pour cette configuration de connexion.

- ✓ Saisir le nom selon les conventions de votre établissement. Par exemple Tp-Com1
- ✓ Validez par OK



- ◆ Un dialogue demande quel port de communication est utilisé.
 - ✓ Sélectionner le port série COM1 (sauf autre configuration de votre matériel).
 - ✓ Validez par OK



- ◆ Un dialogue demande de configurer les paramètres de communication.
 - ✓Sélectionnez les valeurs selon votre besoin.
 - ✓Validez par OK

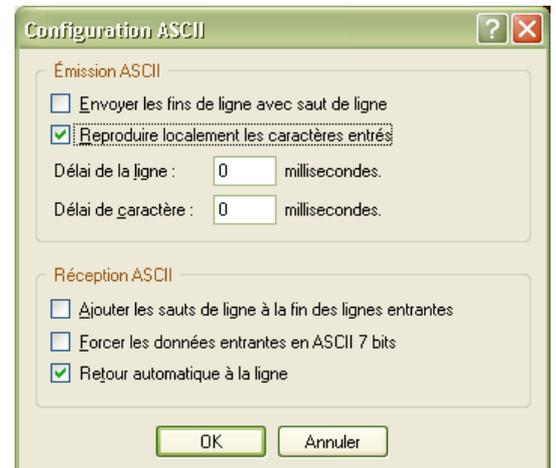
◆La fenêtre de terminal apparaît. Le temps de connexion s'écoule dans la barre d'état en bas à gauche



- ✓ Ouvrir le menu Fichier | Propriétés.



- ✓Cliquer sur le bouton Configuration ASCII...
- ✓Cocher « Reproduire localement les caractères entrés ».
- ✓Valider par OK 2 fois.



Présentation

Le protocole Modbus est un protocole Maître-Esclave qui permet à un seul et unique maître de demander des réponses à des esclaves ou des actions en fonction de la requête. Le maître peut s'adresser aux esclaves individuellement ou envoyer un message de diffusion générale à tous les esclaves. Les esclaves renvoient un message (réponse) aux requêtes qui leur sont adressées individuellement.

| Les requêtes de diffusion générale du maître n'attendent pas de réponses en retour.

Jusqu'à 32 nœuds peuvent résider sur un réseau RS485 (1 maître et jusqu'à 31 esclaves).

Protocole

- Le nombre de bits de la liaison asynchrone doit être de 8.
- La parité utilisée par défaut doit être paire (EVEN). Si l'on n'utilise aucune parité, on doit passer à 2 bits de STOP.
 - Toutes les spécifications de ModBus sont disponibles sur <http://www.modbus.org>
 - Pour voir les requêtes ModBus supportées par le TWIDO, voir l'aide en ligne de TwidoSoft et rechercher l'expression « Requêtes Modbus standard »

Requête en mode RTU

- Trame :

N° esclave	Code fonction	1er paramètre		Autres paramètres	CRC16	
1 octet	1 octet	PF : 1 octet	Pf : 1 octet	N octets	PF : 1 octet	Pf : 1 octet

- N° esclave : de 1 à 247
- N° fonction :
 - 01 : Lecture de n bits de sorties consécutifs : Param1 = Adresse, Param2 = Quantité
 - 02 : Lecture de n bits d'entrées consécutifs : Param1 = Adresse, Param2 = Quantité
 - **01 / 02 : Sur Schneider TSX-Nano et Twido : lecture de n bits internes %Mi consécutifs**
 - 03 : Lecture de n mots internes consécutifs : Param1 = Adresse, Param2 = Quantité
 - 04 : Lecture de n mots registres d'entrées consécutifs : Param1 = Adresse, Param2 = Quantité
 - **03 / 04 : Sur Schneider TSX-Nano et Twido : lecture de n mots internes %MWi consécutifs**
 - 05 : Écriture d'un bit interne ou de sortie : Param1 = Adresse, Param2 = 0000 ('0') ou FF00 ('1')
 - 06 : Écriture d'un mot interne ou registre : Param1 = Adresse, Param2 = Valeur
 - 15 : Écriture de n bits internes ou de sortie consécutifs : Param1 = Adresse, Param2 = Nbre de bits, Param3 = Nbre d'octets, Params[i] = valeurs
 - 16 : Écriture de n mots internes ou registres consécutifs : Param1 = Adresse, Param2 = Nbre de mots, Param3 = Nbre d'octets, Params[i] = valeurs
- 1^{er} paramètre : adresse du bit ou mot adressé
- 2^{ème} paramètre : 'quantité de mots adressés' ou 'valeur du bit ou mot écrit' selon la fonction utilisée
- Autres paramètres : données écrites dans plusieurs mots consécutifs
- CRC16 : code de redondance cyclique pour détecter les erreurs de transmission
- La fin de trame est détectée par une absence d'émission pendant une durée de 3 caractères minimum

Réponse en mode RTU

- Fonctions 01, 02, 03, 04

N° esclave	Code fonction	Nbre de caractères	Données reçues	CRC16	
1 octet	1 octet	PF : 1 octet	N octets	PF : 1 octet	Pf : 1 octet

- Fonctions 05, 06,

N° esclave	Code fonction	Adresse affectée		Donnée écrite		CRC16	
1 octet	1 octet	PF : 1 octet	Pf : 1 octet	PF : 1 octet	Pf : 1 octet	PF : 1 octet	Pf : 1 octet

- Fonctions 15, 16

N° esclave	Code fonction	Adresse affectée		Nbre de données écrites		CRC16	
1 octet	1 octet	PF : 1 octet	Pf : 1 octet	PF : 1 octet	Pf : 1 octet	PF : 1 octet	Pf : 1 octet

Requête en mode ASCII

- Le nombre de bits de la liaison asynchrone doit être de 7.
- La parité utilisée par défaut doit être paire (EVEN). Si l'on n'utilise aucune parité, on doit passer à 2 bits de STOP.
- Le temps par défaut entre 2 caractères peut aller jusqu'à 1 seconde, sinon configurer pour des intervalles plus longs.

Début	N° esclave	Code fonction	1er paramètre	Autres paramètres	LRC	Fin
1 caractère '!': \$3A	2 caractères '01' à 'F7'	2 caractères '01', '02', '03', '04', ...	4 caractères		2 caractères	2 caractères 'CR': \$0D, 'LF': \$0A

- Les fonctions utilisées sont identiques au mode RTU

Calcul du LRC

Somme en hexadécimal, modulo FF, du contenu de la trame, hors entêtes; complémentée à 2 et codée en ASCII.

Exemple : Écriture de la valeur \$1968 à l'adresse \$00A8 sur l'esclave N° 11

- Trame RTU

N° esclave	Code fonction	Adresse de la donnée		Valeur à écrire		CRC16	
0B	06	PF : 00	Pf : A8	PF : 19	Pf : 68	PF :	Pf :

- Trame ASCII équivalente

	Dép.	Esclave		Fonction				Adresse				Valeur				LRC		Fin de ligne	
Caractère	:	0	B	0	6	0	0	A	8	1	9	6	8	C	6	C	L		
Code ASCII	3A	30	42	30	36	30	30	41	38	31	39	36	38	43	36	0D	0A		

- Calcul du LRC

- $0B + 06 + 00 + A8 + 19 + 68 = 13A$ modulo FF = 3A = 0011 1010 b
- Complément à 1 : 1100 0101 b
- Complément à 2 : additionner 1 : 1100 0110 b
- Conversion en Hexadécimal : C 6
- Codage en ASCII : 43 36

Réponse en mode ASCII

Les types de réponses sont les mêmes qu'en mode RTU, sur des trames ASCII

L'automate

L'automate Schneider-Electric TWIDO Compact TWDLCAA24DRF, alimenté en 230V~, dispose de 14 entrées 24VDC et de 10 sorties relais.



Le port RS485 intégré (N°1) est utilisé pour le paramétrage et le développement du programme. Le TWIDO a alors été équipé d'un module d'adaptation RS485 supplémentaire sur le port N°2 de référence TWDNAC485D à sortie Mini-DIN 8 broches. Ce module n'est disponible que sur les automates 16 & 24 E/S compacts et pour le module d'expansion Afficheur.



Pour d'autres types de raccordements, on pourra utiliser le module TWDNAC485T sur bornier à vis ou TWDNAC232D en RS232 sur Mini-DIN 8 broches.

La connexion

Les nouveaux **micro-ordinateurs** ne sont plus équipés de ports série RS232. Des **adaptateurs USB → Série** sont alors proposés. Malheureusement, si ceux-ci remplissent à peu près normalement leur fonction pour des communications standard RS232, ils sont quasiment **inutilisables pour** des communications **RS485 sur 2 fils**. En effet, le RS485 sur 2 fils est un mode semi-duplex, dans lequel l'émetteur et le récepteur prennent la ligne à tour de rôle. La prise de ligne est assurée par l'activation du signal de demande d'émission RTS. Les adaptateurs USB → Série laissent le RTS en permanence à l'état actif, ce qui induit un court-circuit entre l'émetteur maître et le récepteur esclave lorsque ce dernier répond.

Connexion d'un PC à un automate TWIDO

Le raccordement entre le port RS232 du PC et le port RS485 du TWIDO est du même type pour la programmation ou pour la communication de process, à l'aide du câble adaptateur TSXPCX1031, dont les différentes fonctions du mode série sont sélectionnées selon la position du commutateur sur le convertisseur :

<i>Position de l'interrupteur Rotary</i>	<i>Fonction</i>	<i>Signal /DTP</i>	<i>Signal RTS</i>
0	TER MULTI – Connexion en mode point à point. Force le port du terminal en mode maître, protocole par défaut (remplace la câble réf. TSX PCU 1031)	1	oui
1	OTHER MULTI – Connexion en mode multipoint. Autres types de communication (remplace le câble réf. TSX PCD 1030)	0	oui
2	TER DIRECT – Connexion en mode point à point. Force le port du terminal en mode maître, protocole par défaut (remplace la câble réf. TSX PCX 1030, basculé en position maître et TSX PCU 1030)	1	non
3	OTHER DIRECT – Connexion en mode point à point. Autres types de communication définis par la configuration de l'automate (remplace le câble réf. TSX OXC 1030, basculé en position esclave)	0	non

La longueur maximale des connexions RS485 non isolées sur les automates TWIDO est de 200m.

Attention : Dommages électriques potentiels de l'automate :

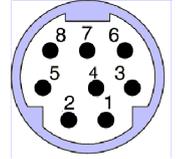
Ne connectez pas le câble de communication à l'automate avant de la connecter au PC.

Connectez toujours le câble au PC en premier.

Limitations de la couche Modbus du Twido

- Adresses des esclaves : 1 à 247
- Bits : 128 bits sur demande
- Mots : 64 mots de 16 bits sur demande

Réalisation d'un cordon de mesure des signaux RS485



Signaux utiles

La documentation en ligne de TwidoSoft fournit le brochage des connecteurs mini-DIN à 8 broches :

Pour assurer la liaison Modbus, il suffit de câbler les signaux RS485 A(+) et B(-). Cependant, le cordon convertisseur TSXPCX1031 est alimenté par le Twido, il est alors nécessaire de permettre le passage de l'alimentation par le 0V et le 5V. D'ailleurs, les seules broches câblées sur le module option RS485 sont ces 4 broches de N° 1, 2, 7 et 8.

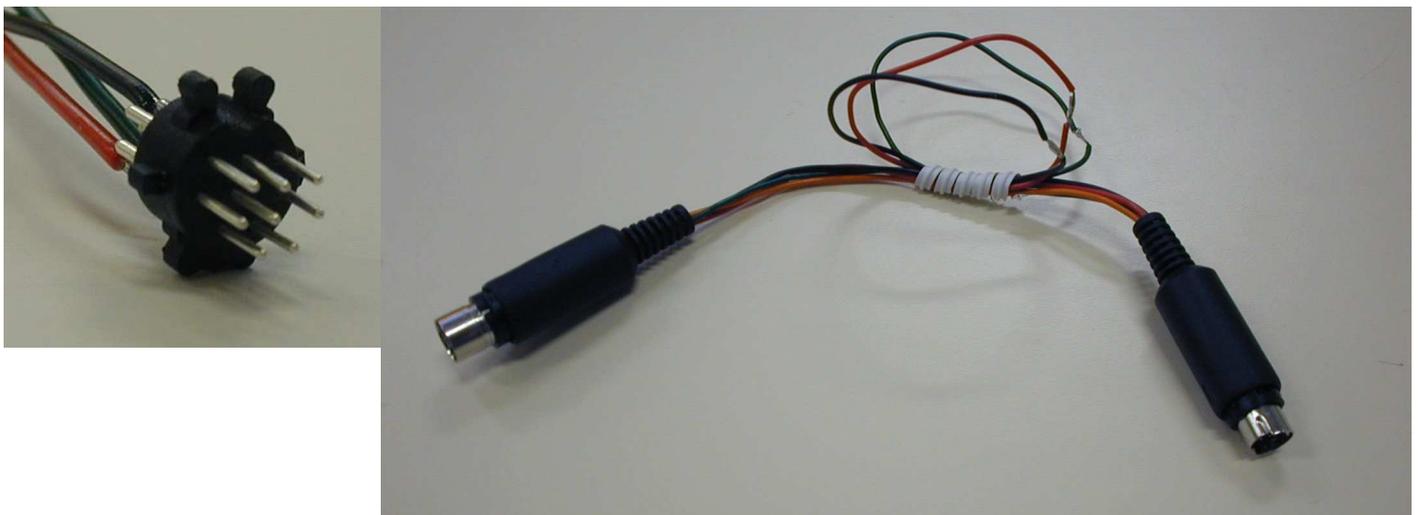
Broches	Base RS485	Option RS485	Option RS232-C
1	A (+)	A (+)	RTS
2	B (-)	B (-)	DTR
3	nc	nc	TxD
4	/DE	nc	RxD
5	/DPT	nc	DSR
6	nc	nc	Gnd
7	0 V	0 V	Gnd
8	5 V	5 V	5 V

Réalisation du cordon de mesure

Il est très difficile de trouver un fournisseur de cordons prolongateurs mini-Din 8 broches (Mâle / Femelle), desquels on aurait extrait les conducteurs 1 et 2. Seul Apple aurait utilisé ce type de câbles pour des liaisons d'imprimante série.

Il faudra donc se procurer un connecteur mâle et 1 connecteur femelle sur lesquels on soudera les 4 conducteurs. Fort heureusement, la broche N° 4 n'est pas à souder. Les broches 1, 2, 7 et 8 sont bien accessibles.

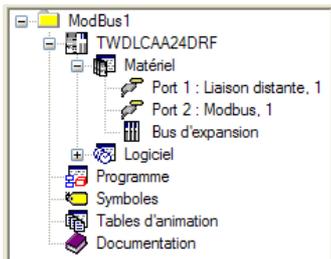
Quelques photos de la réalisation



Les conducteurs ont été dénudés en leur milieu afin d'y attacher les pinces Grip-fil de l'oscilloscope, et d'observer le signal RS485.

Configuration d'un Twido en Esclave ModBus

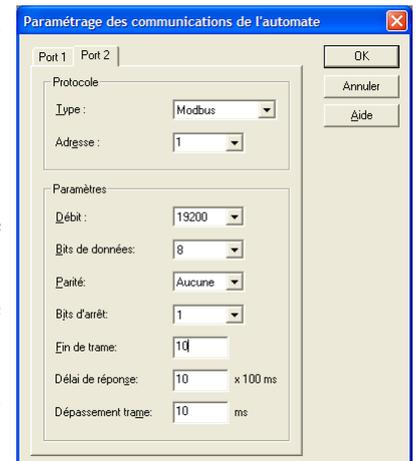
L'automate



Il existe deux types de périphériques ModBus compatibles avec TwidoSoft :

Maître : transmet une requête ModBus et demande des réponses aux périphériques esclaves. Compatible avec l'instruction EXCH.

Esclave : répond aux requêtes ModBus d'un maître ModBus.



Le TWIDO ne nécessite aucune programmation pour devenir esclave ModBus, simplement une configuration.

Les automates Twido gèrent l'utilisation des modes ModBus ASCII et ModBus RTU. Le nombre de bits de données sélectionnés dans la boîte de dialogue *Paramétrage des communications de l'automate* détermine le mode activé :

8 bits de données : protocole **ModBus RTU** 7 bits de données : protocole **ModBus ASCII**

Connecter le PC sur le port de communication TER (N°1) ;

Lancer TwidoSoft et créer une nouvelle application ;

Configurer le port N°2 en Modbus RTU (8 bits) ou ASCII (7 bits) à la vitesse de votre choix : ex. 9600 ou 19200 bps ;

Créer un programme de base qui permet d'agir sur les variables internes et sur les E/S :

- Recopier les 14 entrées %I0.0 à %I0.13 dans les bits 14 internes %M0 à %M13
- Recopier les 10 bits internes %M16 à %M25 sur les 10 sorties %Q0.0 à %Q0.9
- Placer les 4 constantes suivantes dans les mots %MW0 à %MW3 : 0x1234, 0x5678, 0x1000, 0xABCD
- Incrémenter %MW4 chaque 1/10e de seconde

```
(* IMAGES ENTRÉES / SORTIES DANS BITS INTERNES *)
LD 1
[ %M0:14 := %I0.0:14 ]
[ %Q0.0:10 := %M16:10 ]
(* AFFECTATION CONSTANTES DANS 4 MOTS *)
LD 1
[ %MW0 := 16#1234 ]
[ %MW1 := 16#5678 ]
[ %MW2 := 16#1000 ]
[ %MW3 := 16#ABCD ]
(* DÉTECTION FRONT MONTANT %S5 (CHAQUE 1/10E SECONDE) DANS %M31 *)
LD %S5
ANDN %M30
ST %M31
LDN %S5
ST %M30
(* INCRÉMENT %MW4 CHAQUE 1/10E SECONDE *)
LD %M31
[ %MW4 := %MW4 + 1 ]
(* INCRÉMENT %MW6 SUR CHAQUE MISE À 1 DE L'ENTRÉE 0 *)
LDR %I0.0.0
[ %MW6 := %MW6 + 1 ]
```

- Incrémenter %MW6 sur chaque front montant de %I0.0

Enregistrer l'application

Connecter logiquement l'API et y transférer le programme

Placer l'automate en mode RUN

Quitter TwidoSoft

Présentation

Un analyseur de réseau asynchrone permet de mettre en évidence les signaux de contrôle de flux échangés, mais aussi d'afficher en clair les trames de données. Nous avons choisi ce modèle :

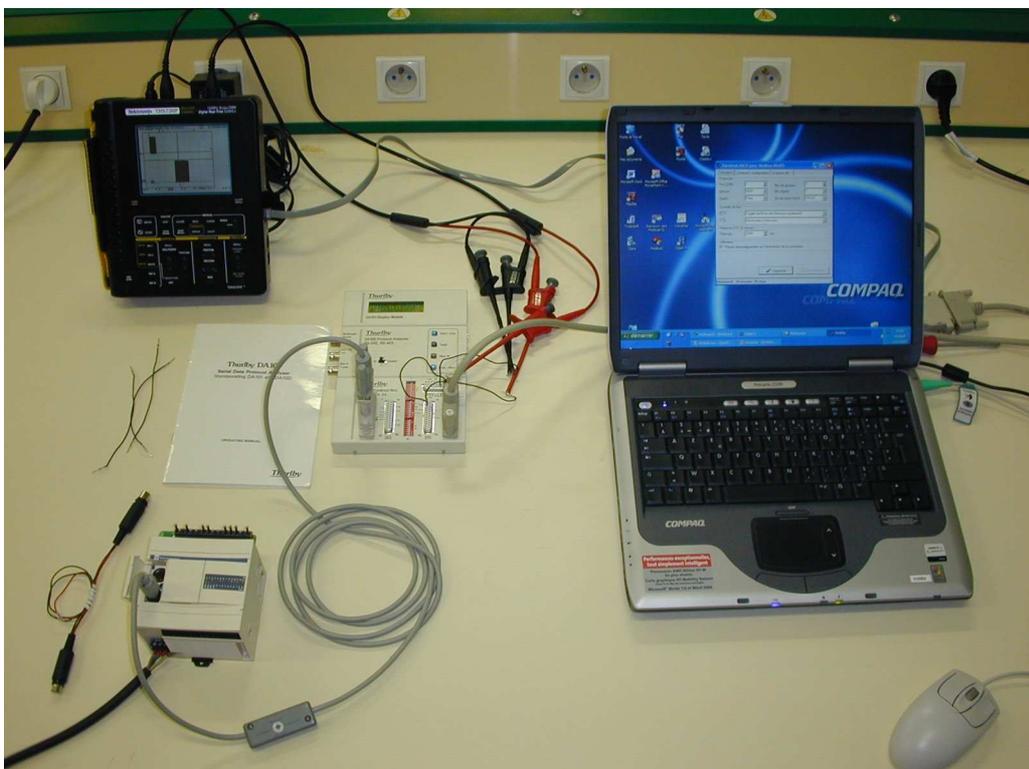
Thurlby DA100 + DA101 + DA102 : <http://www.mbelectronique.fr/html/uploads/prodfr1529.htm>

de Thurlby Thandar Instruments : <http://www.ttinst.co.uk/languages/home-french.htm>

On peut le trouver chez le distributeur Radiospares, qui propose l'ensemble DA100/101/102 pour environ 350 à 400 €.

Cet analyseur met à disposition l'ensemble des signaux de contrôle RS232 qui peuvent alors être visualisés à l'oscilloscope. Celui-ci doit être un modèle à entrées isolées afin d'avoir des masses indépendantes entre les signaux. Ainsi une voie pourra afficher la ligne RS232, tandis que la seconde affichera les données RS485. Nous disposons du modèle

Tektronix THS720P : http://www.tek.com/site/ps/0,,3M-10566-INTRO_EN,00.html



L'ensemble du banc de manipulations peut alors ressembler à l'illustration ci-dessus :

Sortie RS232 du PC

➤ Analyseur de réseau

➤ Adaptateur RS485

➤ API Twido

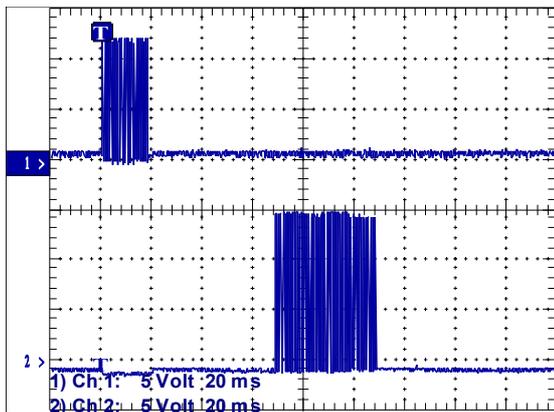
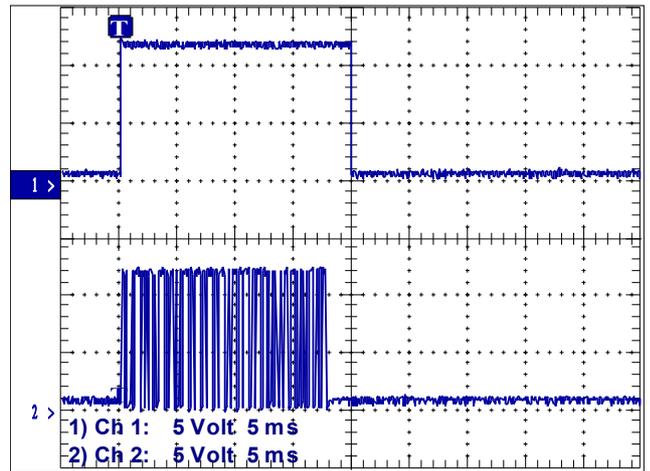
Analyseur de réseau

➤ Oscilloscope

Relevé de signaux RS232 à l'oscilloscope

Sur ce premier relevé, on trouve le signal RTS issu du PC, qui permet à l'adaptateur RS485 de prendre la ligne en charge, et d'émettre les données ici représentées par le signal TxD.

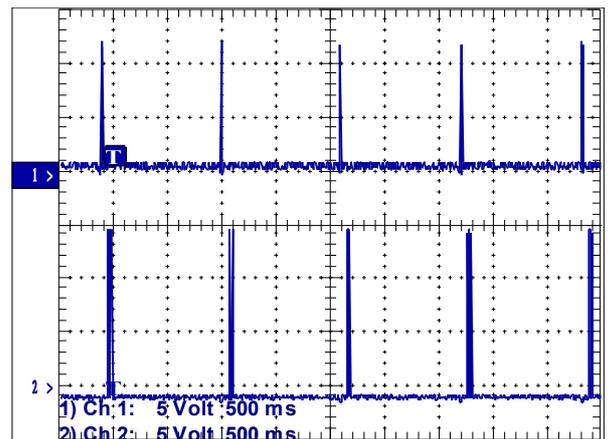
On peut noter les niveaux de tension RS232 en logique négative qui sont d'environ 0,6V pour le niveau bas (1 logique) et environ 12V pour le niveau haut (0 logique).



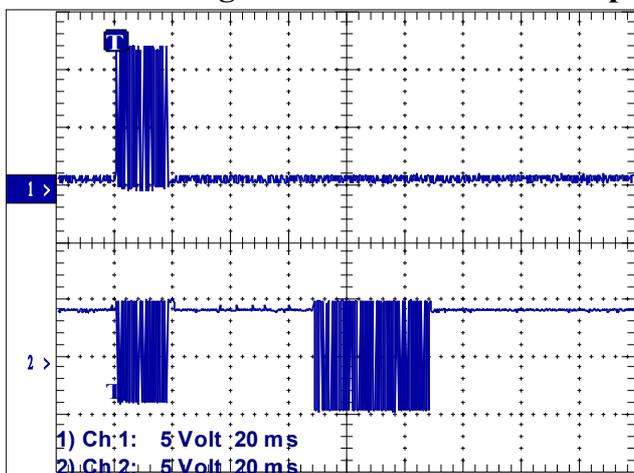
Le deuxième relevé montre l'échange question/ réponse avec en haut la trame de question du PC (TxD), en bas la trame de réponse de l'API (RxD).

On peut noter le temps écoulé entre la question et la réponse (environ 50ms), qui représente le temps de pause signifiant la fin de trame (3 caractères ~ 3 ms à 9600bps), + le temps de décodage de la question, + le temps de préparation de la réponse.

Sur ce troisième relevé, on trouve une suite d'échanges question/ réponse, à un intervalle d'environ 1 seconde entre une réponse et la nouvelle requête.



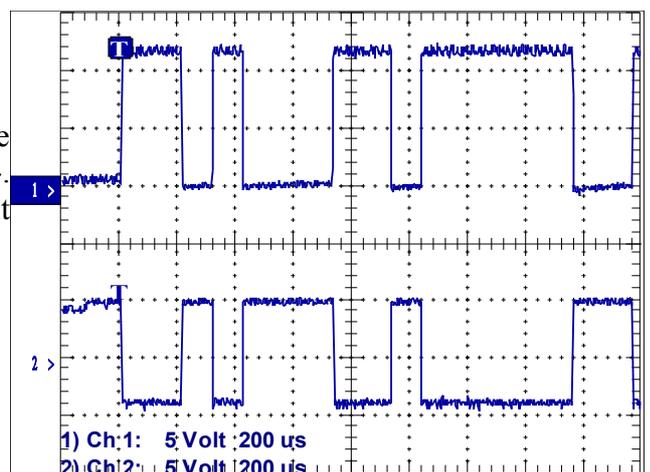
Relevé de signaux RS485 à l'oscilloscope



Ici la voie 1 représente la même émission TxD en RS232.

La voie 2 montre la ligne RS485 qui supporte l'émission TxD et la réception RxD en mode différentiel, c'est à dire en niveaux de tension +5V pour le 1 logique, -5V pour le 0 logique.

Enfin une dernière association RS232/ RS485 pour le premier caractère « : » de la trame, de code ASCII « 3A ». Le caractère a une durée d'environ 1,1 ms à compter du bit de Start qui apparaît à la première division de l'écran.



	Scopix 602x / 604x / 7042 / 704x	Votre oscillo
Activer la voie 1	Cocher ch1	
Désactiver les autres voies	Décocher ch2/math2, ch3/math3, ch4	
Configurer la voie 1 en acquisition DC (signal complet)	Menu Vert ch1 Sensibilité/Couplage Cocher Couplage DC	
Régler la voie 1 sur 5V /div	Menu Vert ch1 Sensibilité/Couplage Sensibilité voie : 5.00V	
Configurer la bande passante (Pour éliminer certains parasites)	Menu Vert ch1 Sensibilité/Couplage Limit BP 1.5 MHz	
Régler le déclenchement sur la voie 1	Menu Décl Paramètres Onglet Principal Source : 1	
Régler le couplage de déclenchement en DC (pour déclencher sur un niveau de tension)	Menu Décl Paramètres Onglet Principal Couplage : DC	
Régler le niveau de déclenchement à ~ 1 V (0,5 < V < 3V)	Menu Décl Paramètres Onglet Principal Niveau : 1.00 V	
Régler le déclenchement sur front montant	Menu Décl Paramètres Onglet Principal Front : +	
Régler le déclenchement sur apparition d'un événement type front	Menu Decl Mode déclenché	
Voir si trace ...		
Régler la base de temps horizontale	Appuyer sur les 2 boutons de /Horizontal/ jusqu'à une base de temps de 0,2 ms.	
Tester la trace	Où est le début de trace ?	
Régler le début de trace	Cliquer sur le « T » en haut de la trace et le placer à 10% de la largeur (fin de la première division)	

1. Système Éolienne didactisée

Système :

Eolienne didactisée

Objectif :

Produire de l'énergie électrique



Fournisseur

GTI Systèmes - Carcassonne

Dispositif communicant :

Compteur d'énergie
ARDETEM
PECA30E



Emplacement du dispositif

Façade

Connexion du dispositif

Esclave d'API TSX Micro 3722

Adresse de test

140

Grandeur à mesurer

Tension Phase 1

Remarques :

- Débrancher le maître ModBus par le connecteur DB9
- Configurer le format de récupération des données réelles : flottant IEEE ou Double entier



2. Système TGBT Communicant

Systeme :

Armoire TGBT

Objectif :

Distribuer l'énergie électrique et assurer les protections



Fournisseur

DEC-Industrie

Dispositif communicant :

Compteur d'énergie
ENERDIS
PM820



Emplacement du dispositif :

Façade de l'armoire TGBT



Connexion du dispositif

Esclave d'un API Wago

Adresse de test

31102

Grandeur à mesurer

Puissance Phase 2

Remarques :

- Déconnecter le maître Modbus sur le module RS485 de l'API
– paire de conducteurs blanc et marron



3. Station de pompage

Système :

Station de pompage

Objectif :

réguler la hauteur d'une colonne d'eau



Fournisseur

Réalisation Section électrotechnique

Dispositif communicant :

Variateur de vitesse
Leroy-Somer
DIGIDRIVE



Emplacement du dispositif :

Intérieur armoire électrique



Connexion du dispositif

Esclave de l'API TSX Premium

Adresse de test

Grandeur à mesurer

Vitesse moteur pompe

Remarques :

Débrancher le TSX maître par la prise SUD-D métallique sur le boîtier TSX-ACC1



4. Bain thermorégulé

Système :

Bain régulé

Objectif :

Réguler la température d'un bain de traitement thermique



Fournisseur

GTI Systèmes - Carcassonne

Dispositif communicant :

Régulateur
Chauvin Arnoux
STATOP 48



Emplacement du dispositif :

Façade de l'armoire électrique



Connexion du dispositif

Esclave de réseau externe par connecteur châssis

Adresse de test

Grandeur à mesurer

Remarques :