

Sciences et Techniques Industrielles

# Appareillages communicants

## Documents ressources

- ✓ Communications asynchrones RS232 et RS485
  - ✓ Configuration HyperTerminal
- ✓ Extrait Datasheet cordon de conversion USB/RS485 FTDI
- ✓ Réglage d'oscilloscope pour événement non périodique
  - ✓ Codage de caractères ASCII
- ✓ Protocole de communication ModBus

**Section de Technicien Supérieur en Électrotechnique**  
**Étudiants et Apprentis**

# Communications « série asynchrone » RS232 et RS485

## 1. Présentation

Extrait du [référentiel 2020](#) :

Obtenir le brevet de technicien supérieur (BTS) « Électrotechnique » permet d'être spécialiste des installations électriques « intelligentes », qui intègrent **les technologies numériques, communicantes** et les objets connectés au service des enjeux énergétiques.

## 2. Technologie

### 2.1. Protocole

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

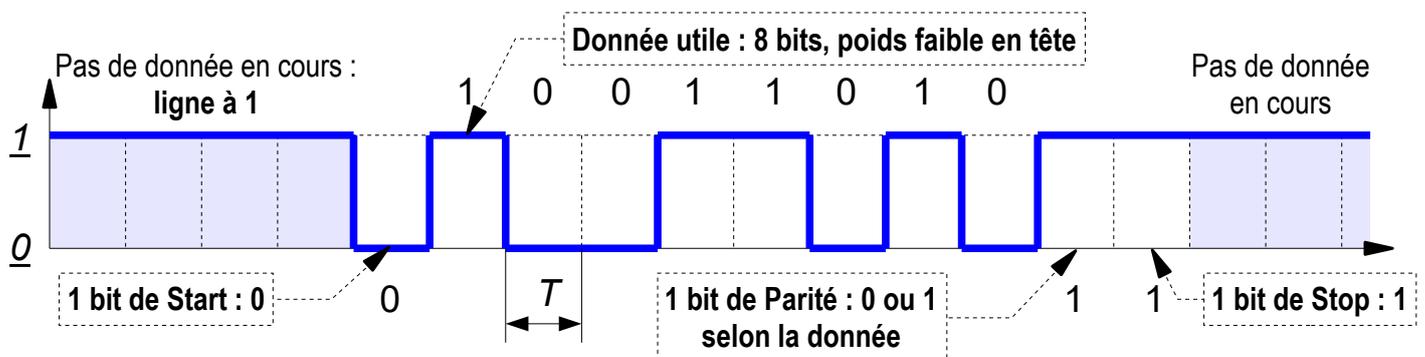
- un bit de **Start** au niveau logique « 0 »,
- les **5 à 8 bits de données**, *poids faible en tête*,
- *éventuellement*, un **bit de vérification de**

**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 à 1 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** au niveau logique « 1 ».



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

### 2.2. 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)**.

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  $104 \mu\text{s}$ .

#### Note

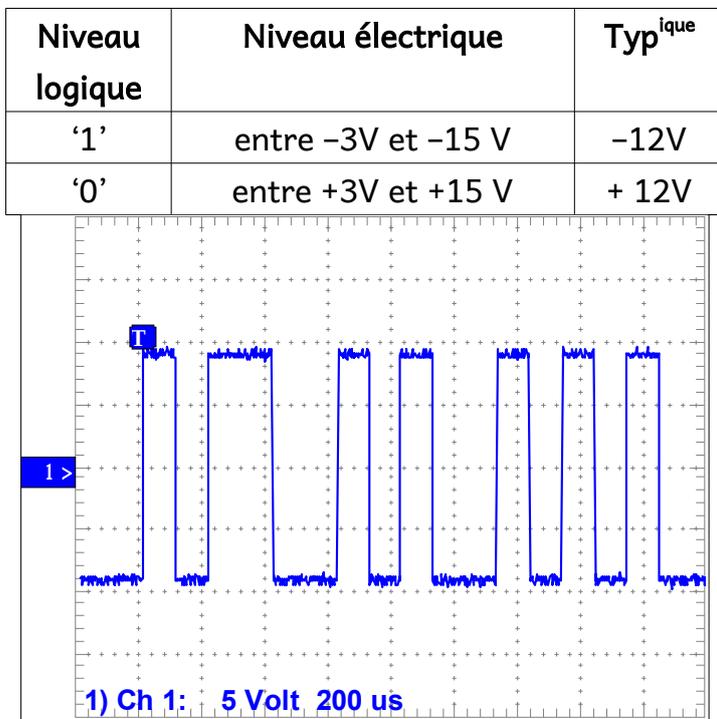
Le chronogramme ci-dessous représente l'état logique AVANT la mise en forme en tension par l'adaptation de ligne, c'est à dire indépendamment du standard RS232, RS422 ou RS485 dont les niveaux de tension sont définis plus loin.

### 3. Liaison RS232

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 de **15 mètres**.

#### 3.1. Niveaux des signaux

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



#### 3.2. Signaux de données

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

<b>Tx</b>	Transmitted Data	Conducteur d'émission des données
<b>Rx</b>	Received Data	Conducteur de réception des données
<b>Gnd</b>	Ground	Conducteur de référence du signal (« 0 V »)

#### 3.3. 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 (voir plus loin le glossaire pour DTE et DCE).

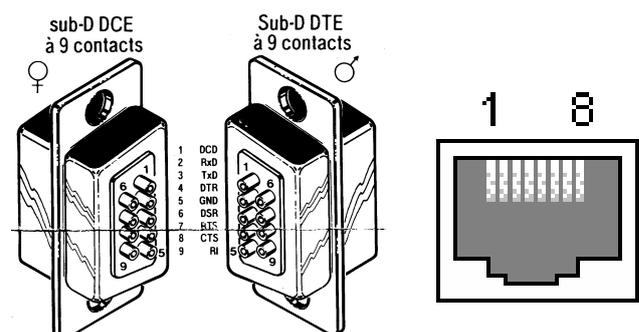
<b>RTS</b>	« Request To Send » est abaissé (« 0 ») pour préparer le DCE à accepter les données transmises.
<b>CTS</b>	« Clear To Send » est abaissé par le DCE pour informer le DTE que la transmission peut débuter.

##### Contrôle de flux logiciel

Le récepteur stoppe le flux de données en envoyant sur la ligne de données un caractère ASCII dédié nommé XOFF, et le relance en envoyant le caractère XON.

D'où le nom du protocole XON/XOFF.

#### 3.4. Connectiques



## 4. Liaisons RS422, RS485

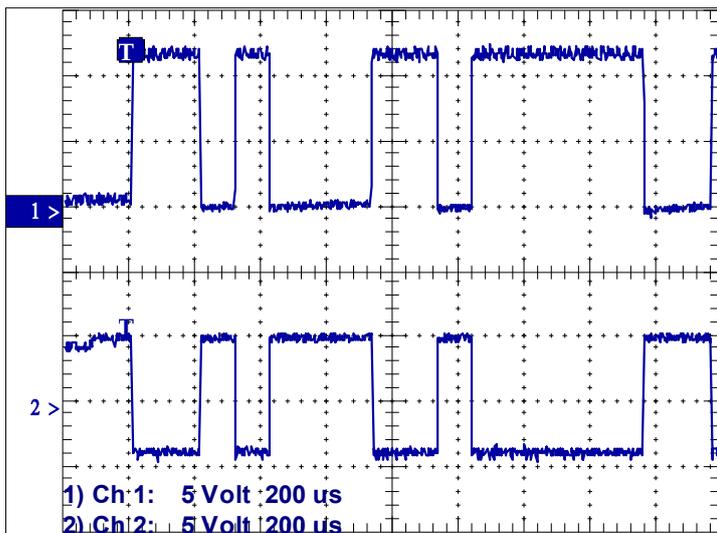
L'évolution temporelle des signaux RS422 et RS485 est identique aux signaux RS232 décrits précédemment.

### 4.1. Niveaux des signaux

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 négative de l'ordre de **- 5 V**

Contrairement au format RS232, RS485 travaille donc en logique positive.

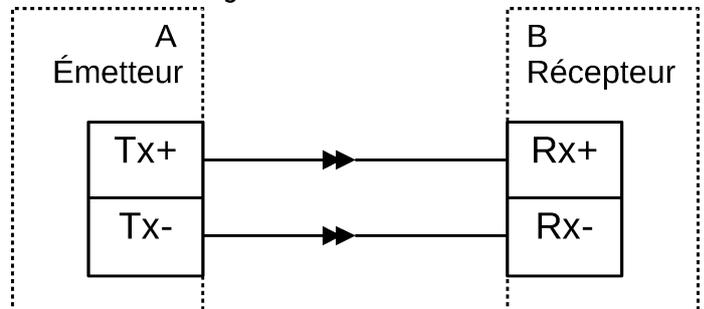


### 4.2. Nature des signaux

#### Signaux en mode **SIMPLEX** : un seul sens de communication **A → B**

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

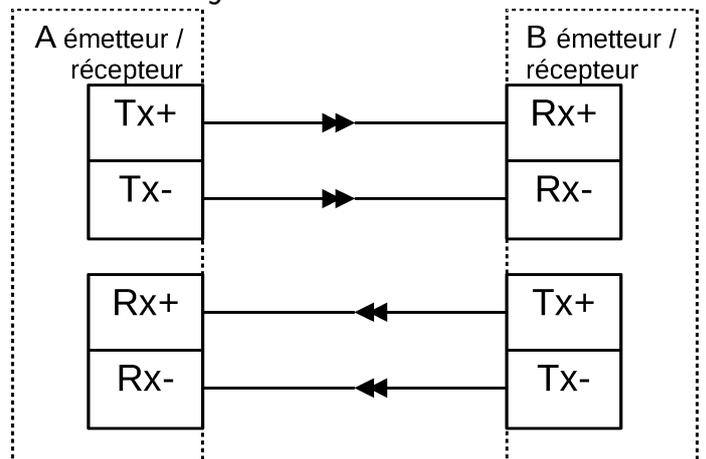
- le conducteur d'émission de données Tx+ de A qui va sur Rx+ de B : polarité positive
- le conducteur d'émission de données Tx- de A qui va sur Rx- de B : polarité négative
- le blindage



#### Signaux en mode **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+ de A → Rx+ de B) et (Tx- de A → Rx- de B)
- les 2 conducteurs de réception des données (Rx+ de A ← Tx+ de B) et (Rx- de A ← Tx- de B)
- le blindage



### Signaux en mode Half-Duplex : 2 sens de communication alternés

Sur une liaison bidirectionnelle alternée de type **2 fils** (la plus utilisée), il faudra :

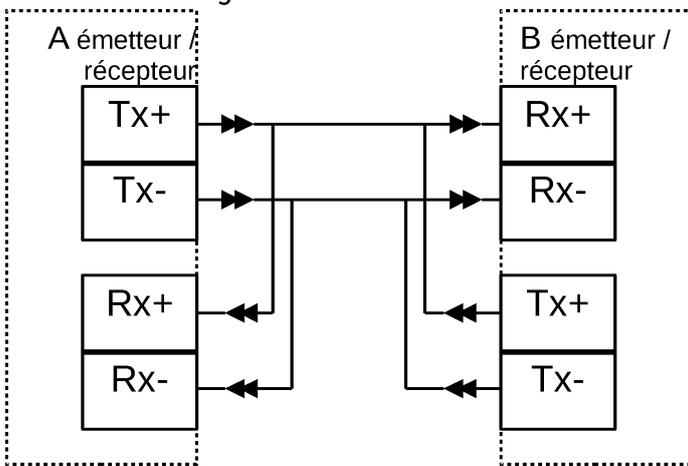
- le conducteur d'émission/ réception des données

Tx+/Rx+ de A et B

- le conducteur d'émission/ réception des données

Tx-/Rx- de A et B

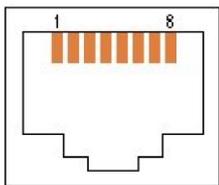
- le blindage



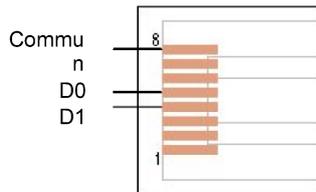
### 4.3. Connectique

#### Connexion RJ45

Vue de face



Vue de dessus



## 5. 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.

Les **terminaux de données** passifs ne possèdent pas de moyens de traitement autonomes et se contentent :

- d'accepter ce que l'utilisateur frappe au clavier, de le transmettre à un ordinateur ;
- d'imprimer les réponses de l'ordinateur ou de les afficher sur un écran (soit en mode machine à écrire, soit en mode plein écran, que ce soit du texte pur ou des images).

## 6. Ressources

Normes :

[https://www.camiresearch.com/Data\\_Com\\_Basics/RS232\\_standard.html](https://www.camiresearch.com/Data_Com_Basics/RS232_standard.html)

## 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 à partir de Windows Vista (Windows 7, 8, 10, 11, ...).

On pouvait télécharger gratuitement « HyperTerminal Private Edition » v6.3 : [htpe63.exe](http://htpe63.exe)

La version 6.4 est désormais payante : <https://www.hilgraeve.com/hyperterminal/>

## Configuration de l'HyperTerminal sous Windows

- ◆ 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 (5 pour le sud-ouest) ;
- ✓ Conserver le mode Fréquence vocale ;
- ✓ 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 COM utilisé sur votre ordinateur
  - ✓ Validez par OK



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

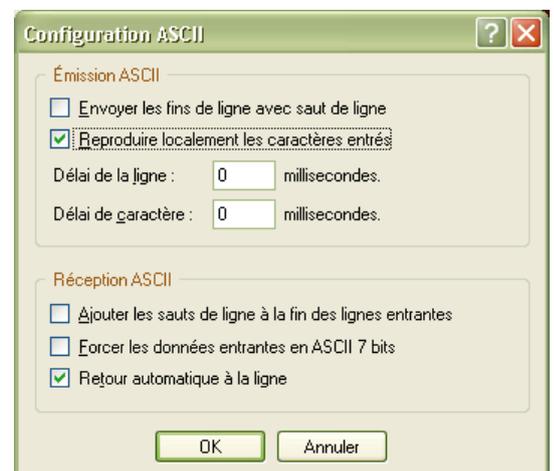
- ◆ La fenêtre de terminal apparaît. Le temps de connexion s'écoule dans la barre d'état en bas à gauche



- ✓ Ouvrir le menu Fichier | Propriétés.



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



## 1 Description

The **USB-RS485** cable is a USB to RS485 levels serial UART converter cable incorporating FTDI's FT232RQ USB to serial UART interface IC device which handles all the USB signalling and protocols. The cable provides a fast, simple way to connect devices with a RS485 interface to USB.

Each USB-RS485 cable contains a small internal electronic circuit board, utilising the FT232R, which is encapsulated into the USB connector end of the cable. The FT232R datasheet, [DS\\_FT232R](http://www.ftdichip.com), is available at <http://www.ftdichip.com>. The integrated electronics also include the RS485 transceiver plus Tx and Rx LEDs which give a visual indication of traffic on the cable (if transparent USB connector specified).

The other end of the cable is bare, tinned wire ended connections by default, but can be customised using different connectors to support various applications.

The USB-RS485 cable is designed for use at the end of an RS485 network up to approximately 10m. For longer networks or placement of the USB-RS485 cable in the middle of an RS485 network it is recommended that the attached wire is cut to 20cm or less to ensure signal integrity

Cables are FCC, CE, RoHS compliant.

The cable is USB powered and USB 2.0 full speed compatible. Each cable is 1.8m long and supports a data transfer rate up to 3 Mbaud. Each cable supports the FTDIChip-ID™, with a unique USB serial number programmed into the FT232R. This feature can be used to create a security or password protected file transfer access using the cable. Further information and examples on this feature are available at <http://www.ftdichip.com> under [FTDIChip-ID Projects](#).

The USB-RS485 cables require USB drivers, available free from <http://www.ftdichip.com>, which are used to make the FT232R in the cable appear as a virtual COM port (VCP). This then allows the user to communicate with the USB interface via a standard PC serial emulation port (for example TTY). Another FTDI USB driver, the D2XX driver, can also be used with application software to directly access the FT232R on the cable through a DLL. This is illustrated in the Figure 1.1

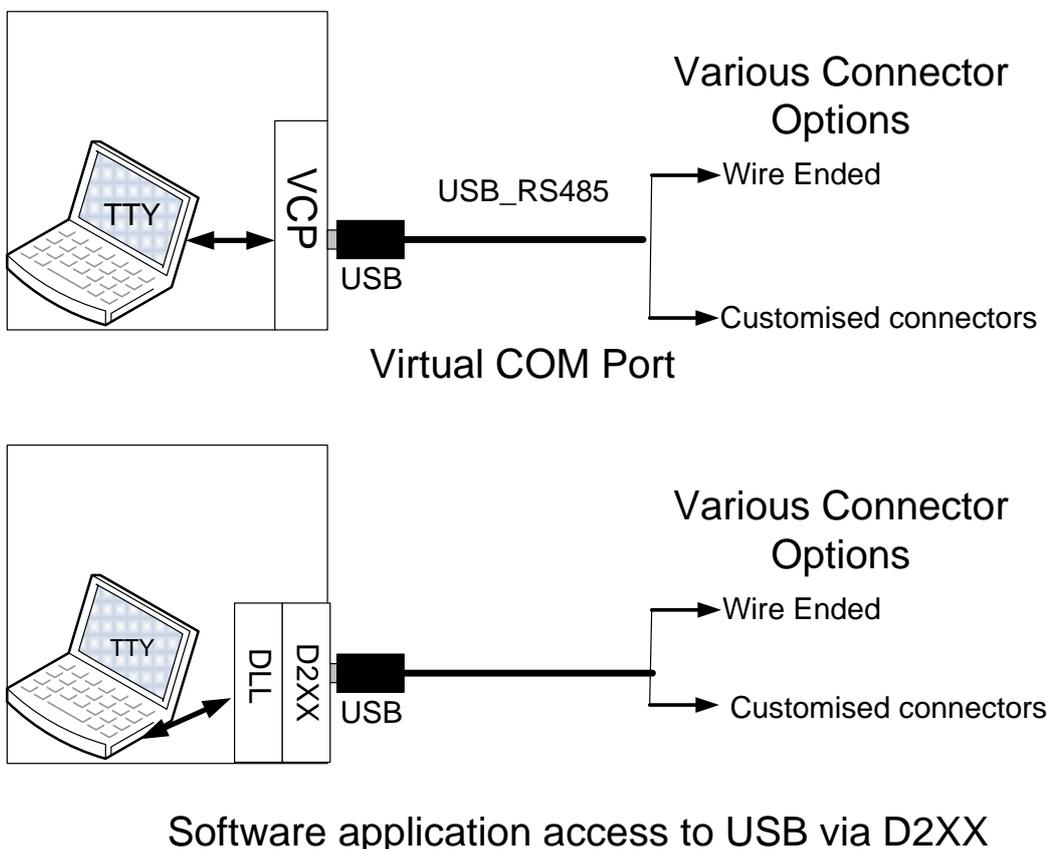


Figure 1.1 Using the USB-RS485 Cable

## 5 USB-RS485-WE-LLLL-CU-X

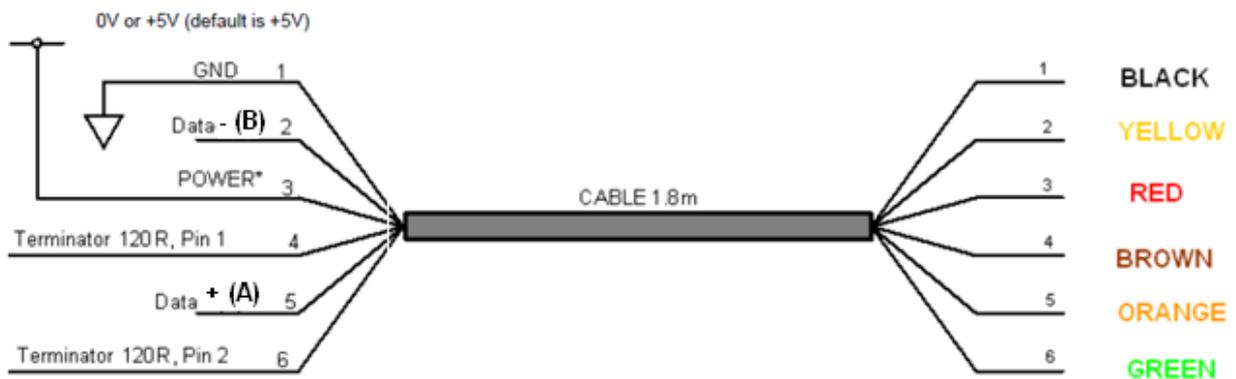
The USB-RS485-WE cable is un-terminated; it has bare and tinned wires.

The LLLL specifies the length of the cable in cm. The CU specifies the colour of the cable and the colour of the USB connector. The cable can be either Black or transparent. The USB connector comes default as transparent because of LED implemented inside the plug but can be sold in black colour as well. For simplicity, the LLLL and CU have been dropped from the following descriptions.

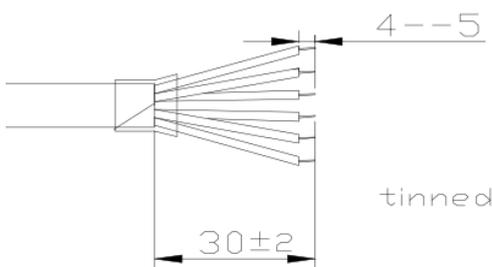
The cable outputs 5V power on the wire by default. This can be modified to output 0V power. The letter X in the part number indicates the output power selection.

### 5.1 USB-RS485-WE Connections and Mechanical Details

The following **Figure 5.1** shows the cable signals and the wire colours for the signals on the USB-RS485-WE cable.



**Figure 5.1 USB-RS485-WE Connections**



**Figure 5.2 USB-RS485-WE Mechanical Details (dimensions in mm)**



**Figure 5.3 USB-RS485-WE Cable images**

## 5.2 USB-RS485-WE Cable Signal Descriptions

Colour	Name	Type	Description
Black	GND	GND	Device ground supply pin.
Brown	Terminator 1	Input	Pin 1 of 120R Terminating Resistor. Only Required if the USB-RS485-WE cable is the first or last device in a multi-drop RS485 System, to meet RS485 Termination Requirements.
Red	POWER	Output	Default output is +5V when active, 0V during suspend.  Power output can be customized to 0V, if required, contact FTDI Sales Team (sales1@ftdichip.com). Switchable power to external device, 250mA max. OFF during suspend
Orange	Data+(A)	Bi-Direction	Data + (A) Signal
Yellow	Data-(B)	Bi-Direction	Data - (B) Signal
Green	Terminator 2	Input	Pin 2 of 120R Terminating Resistor. Only Required if the USB-RS485-WE cable is the first or last device in a multi-drop RS485 System, to meet RS485 Termination Requirements.

**Table 5.1 USB-RS485-WE Cable Signal Descriptions**

## 5.3 USB-RS485-WE Electrical Parameters

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
VCC_5V	Output Power Voltage*	4.25	5.0*	5.25	V	This only applies when cable has been configured to output 5V. The range is dependent on the USB port that the USB-RS485-WE is connected to
I <sub>o</sub>	Output Power Current***	-		250	mA	Must be less than 2.5mA during suspend.
T	Operating Temperature Range	-40		+85	°C	

**Table 5.2 USB-RS485-WE I/O Operating Parameters**

\*Default is 5V.

\*\*\*Only applies when POWER output is 5V.

## Acquisition de l'événement

	<b>Scopix 602x / 604x / 7042 / 704x</b>	<b>Votre oscillo</b>
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 (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 ou descendant selon le début d'événement	Menu Décl   Paramètres Onglet Principal, Front : + ou – ou ↑ ou ↓	
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 (ou 200 μs)	
Tester la trace	Où est le début de trace ?	
Régler le début de trace	Cliquer sur le « T » de début de déclenchement en haut de la trace et le placer à 10% de la largeur (fin de la première division)	

## Impression de la trace sur réseau Ethernet

	<b>Scopix 602x / 604x / 7042 / 704x</b>	<b>Votre oscillo</b>
Configurer l'adresse réseau de l'oscilloscope	Menu Utils   Config ports E/S   Réseau Adresse IP de l'oscilloscope : 172.20.199.22 (ou selon équipement) Adresse IP de l'imprimante : 172.20.198.250 (ou selon équipement) Valider par [OK]	
Définir le protocole de l'imprimante	Menu Utils   Copie d'écran   Protocole HP-PCL	
Définir le port d'impression	Menu Utils   Copie d'écran   Port Réseau	

## 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<sup>ème</sup> 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 <sup>1</sup>		16 <sup>0</sup>
valent		16		1
associées à	×	<b>4</b>		<b>1</b>
donnent	=	64	+	1
			=	65

#### Conversion en binaire :

Chaque digit		<b>4</b>		<b>1</b>
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>	
<b>Commandes de format</b>			
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	
<b>Extension de code</b>			
SO	Shift out		
SI	Shift in		
ESC	Escape	Début de séquence d'échappement	
<b>Commande de séparation</b>			
FS	File separator		
GS	Group separator		
RS	Record separator		
US	Unit separator		
EM	End of medium		
<b>Commandes de communication synchrone</b>			
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		
<b>Commandes de périphérique</b>			
DC1	Device control 1	DC3	Device control 3
DC2	Device control 2	DC4	Device control 4
<b>Commandes diverses</b>			
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>

## Présentation

Le protocole Modbus est un protocole Maître-Esclave qui permet à un seul et unique maître de demander, à des esclaves, des réponses ou des actions en fonction de la requête. Le maître peut s'adresser aux esclaves individuellement (« requête ») 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 au niveau caractère

- Le nombre de bits de la liaison asynchrone doit être de 8.
- Un « caractère » est constitué de 7 ou 8 bits de données :
  - le mode « RTU » (8 bits utiles par caractère) est le mode d'exploitation, le plus rapide, le plus performant ;
  - le mode « ASCII » demande la saisie au clavier de chaque caractère par son code en hexadécimal (7 bits par chiffre) , il est principalement utilisé à fins de maintenance et débogage.
- 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 <https://www.modbus.org>
  - Pour voir les requêtes ModBus supportées par le Modicon M221, voir l'aide en ligne de l'appareil/

## 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 internes (ou de sorties selon l'appareil) consécutifs :  
Param1 = Adresse, Param2 = Quantité (sur 16 bits)
  - 02 : Lecture de n bits d'entrées consécutifs :  
Param1 = Adresse, Param2 = Quantité (sur 16 bits)
  - 03 : Lecture de n mots internes (« registres ») consécutifs :  
Param1 = Adresse, Param2 = Quantité (sur 16 bits)
  - 04 : Lecture de n mots d'entrées consécutifs :  
Param1 = Adresse, Param2 = Quantité (sur 16 bits)
  - 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 :
    - x Param1 = Adresse
    - x Param2 = Nbre de bits
    - x Param3 = Nbre d'octets
    - x Params[i] = valeurs
  - 16 : Écriture de n mots internes ou registres consécutifs :
    - x Param1 = Adresse
    - x Param2 = Nbre de mots
    - x Param3 = Nbre d'octets
    - x Params[i] = valeurs
- 1<sup>er</sup> paramètre : adresse du bit ou mot adressé
- 2<sup>ème</sup> paramètre : 'quantité de mots adressés' ou 'valeur du bit ou mot écrit' selon la fonction utilisée
- Autres paramètres : données écrites dans plusieurs mots consécutifs
- Les paramètres sont codés sur 2 octets poids fort (PF) en tête, poids faible (pf) à suivre
- 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	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	CR	LF			
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