
Modbus

Objectifs

- Identifier le protocole de communication modbus
- Comprendre le Modbus TCP/IP
- Comprendre le Modbus RTU

Prérequis:

- Auto1, Auto2
- CMSE2 et CMSE3 2

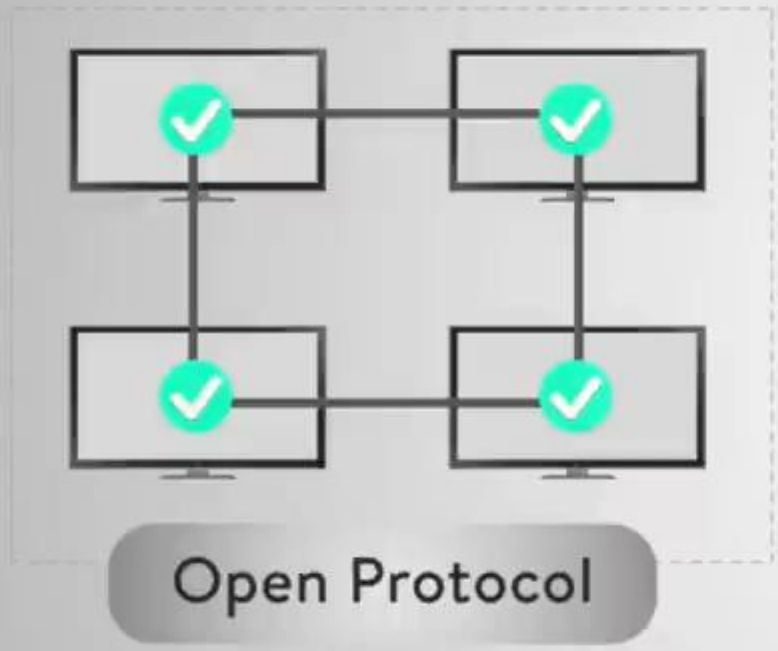
public cible:

- 3ième année Génie industriel et maintenance

Introduction

- Modbus (**MODicon BUS**) est un protocole de communication série initialement publié par Modicon (aujourd'hui Schneider Electric) en 1979 pour une utilisation avec ses automates programmables industriels . Modbus est devenu un protocole de communication standard et désormais un moyen couramment disponible pour connecter des appareils électroniques industriels.
- Modbus est maintenant un protocole ouvert et largement accepté du domaine public (**open, public-domain Protocol**). Il est souvent utilisé pour connecter un ordinateur de supervision à une unité terminale distante (Remote terminal unit RTU) dans les systèmes de contrôle de supervision et d'acquisition de données (**SCADA**).
- MODBUS est un protocole de messagerie de **couche application**, positionné au **niveau 7** du modèle OSI, qui assure la communication entre les appareils connectés sur différents types de bus ou réseaux.

Automation Communication Protocols



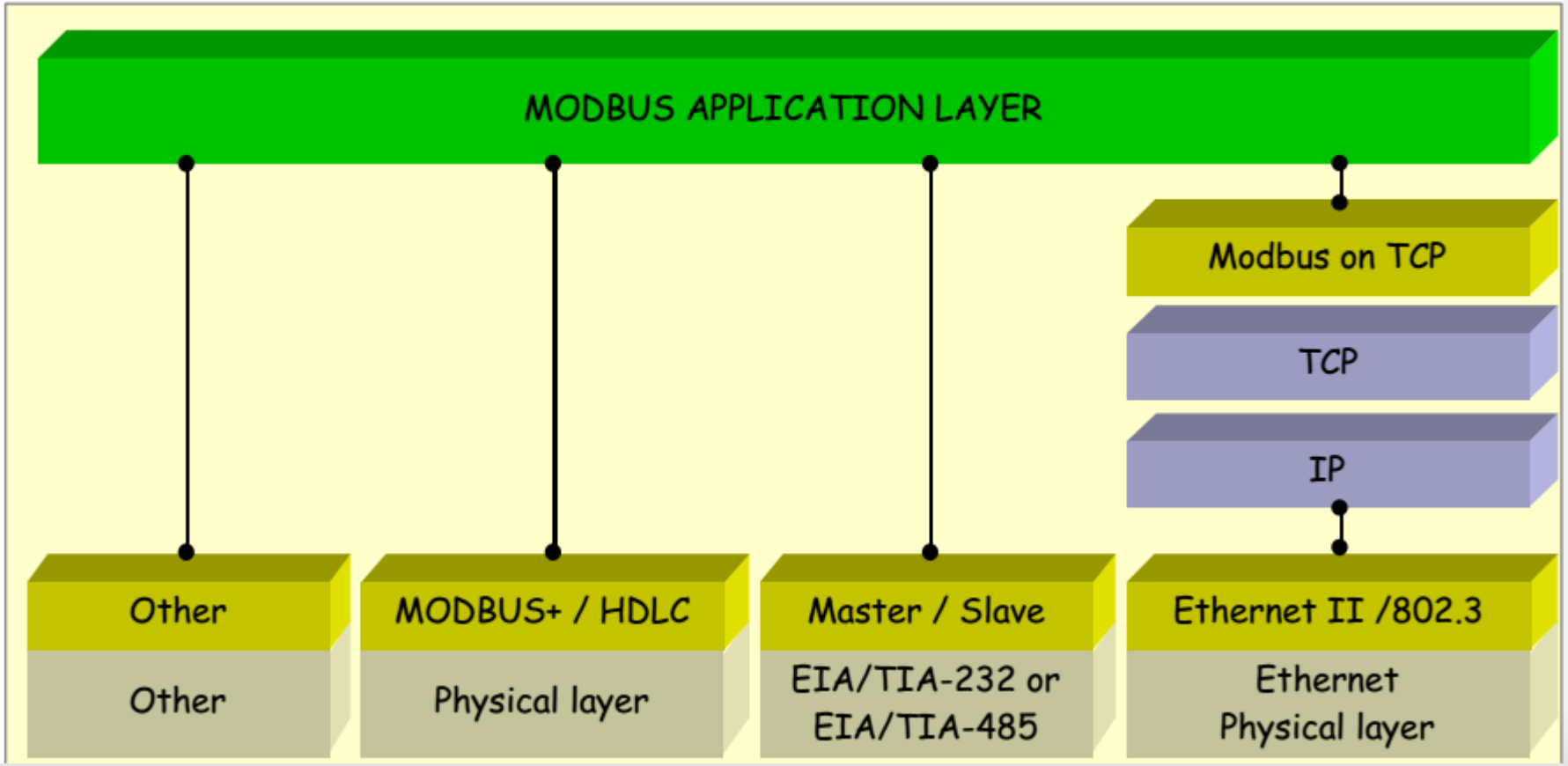
Modbus

https://www.youtube.com/watch?v=txi2p5_OjKU&t=75s

REALPARS

MODBUS est un protocole de messagerie de **couche application** pour la communication client / serveur entre périphériques connectés sur différents types de bus ou de réseaux.
Il est actuellement implémenté en utilisant:

- **TCP/IP sur Ethernet**.
- **Transmission asynchrone série** sur une variété de supports (fil: EIA / TIA-232-E, EIA-422,EIA / TIA-485-A; fibre, radio, etc.)
- **MODBUS PLUS**, un réseau de passage de jetons à grande vitesse.



MODBUS communication stack

❑ Limitations

- Étant donné que Modbus a été conçu à la fin des années 1970 pour communiquer avec les contrôleurs logiques programmables, le nombre de types de données est limité à ceux compris par les automates programmables à l'époque. Les gros objets binaires ne sont pas pris en charge.
- Il n'existe aucun moyen standard pour un nœud de trouver la description d'un objet de données, par exemple, pour déterminer si une valeur de registre représente une température comprise entre 30 et 175 degrés.
- Étant donné que Modbus est un protocole **maître / esclave**, il n'y a aucun moyen pour un appareil de terrain de "signaler une exception" (sauf sur Ethernet TCP / IP, appelé open-mbus) - le nœud maître doit régulièrement interroger chaque appareil de terrain et rechercher les changements dans les données. Cela consomme de la bande passante et du temps réseau dans les applications où la bande passante peut être coûteuse, comme sur une liaison radio à faible débit.
- Modbus est limité à l'adressage de 254 appareils sur une seule liaison de données, ce qui limite le nombre d'appareils de terrain pouvant être connectés à une station maître (encore une fois, Ethernet TCP / IP est une exception).
- Les transmissions Modbus doivent être contiguës, ce qui limite les types d'appareils de communication à distance à ceux qui peuvent mettre en mémoire tampon les données pour éviter les lacunes dans la transmission.
- Le protocole Modbus lui-même n'offre aucune sécurité contre les commandes non autorisées ou l'interception de données.

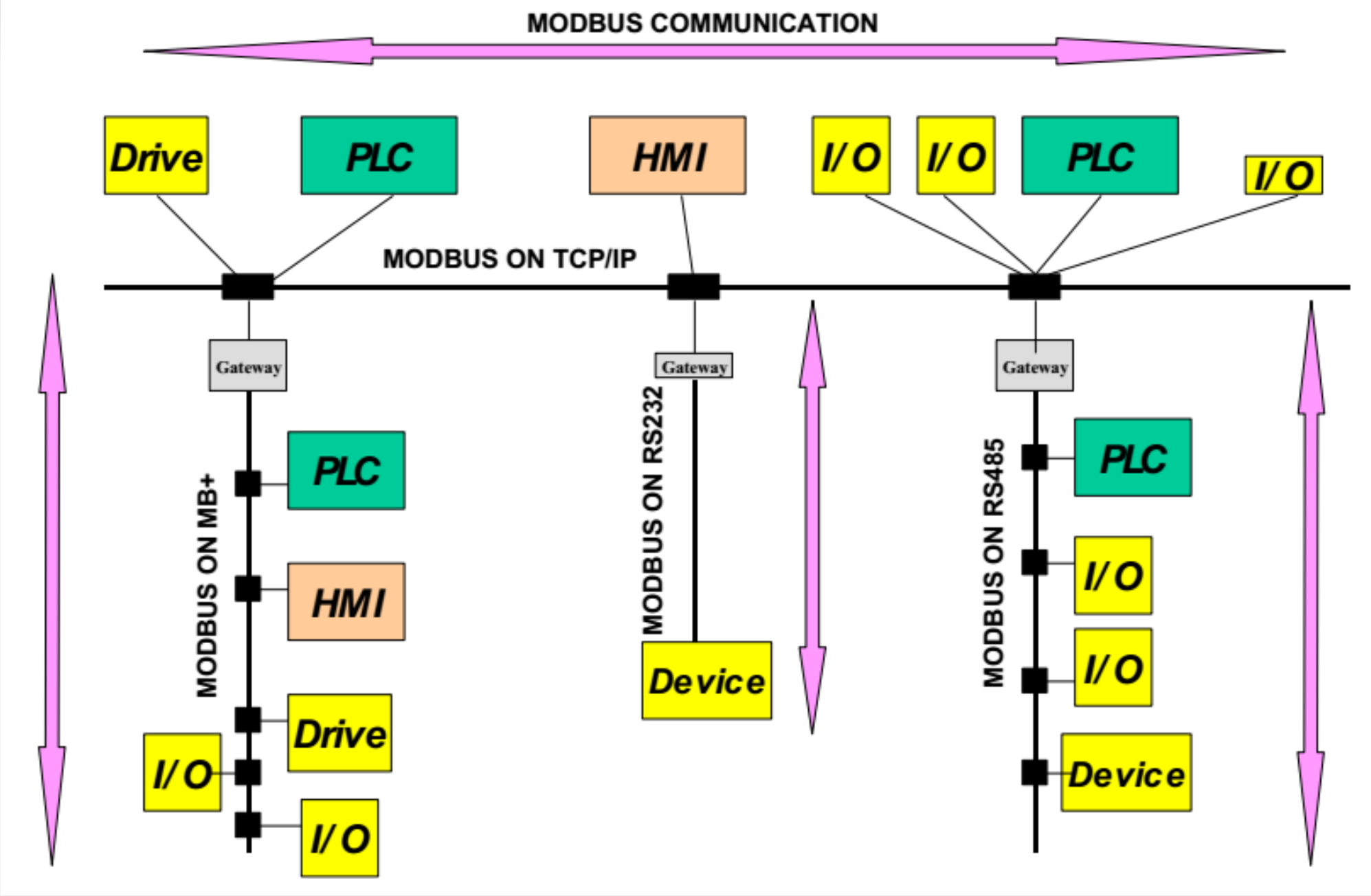
Versions de protocole

Il existe des versions du protocole Modbus pour **le port série** et pour **Ethernet** et d'autres protocoles qui prennent en charge la suite de **protocoles Internet** . Il existe de nombreuses variantes de protocoles Modbus:

1. **Modbus RTU** *(Remote Terminal Unit)* - Il est utilisé dans **la communication série** et utilise une représentation binaire compacte des données pour la communication par protocole. Le format RTU suit les **commandes/données** avec une somme de contrôle de contrôle de **redondance cyclique.** comme mécanisme de contrôle d'erreur pour garantir la fiabilité des données. Modbus RTU est l'implémentation la plus courante disponible pour Modbus. Un message Modbus RTU doit être transmis en continu sans hésitation entre les caractères. Les messages Modbus sont encadrés (séparés) par des périodes d'inactivité (silencieuses).
2. **Modbus ASCII** (*American Standard Code for Information Interchange*)- Il est utilisé dans la communication série et utilise des caractères ASCII pour la communication par protocole. Le format ASCII utilise une somme de contrôle de contrôle de **redondance longitudinale** . Les messages Modbus ASCII sont encadrés par deux points (":") et un saut de ligne (CR / LF).
3. **Modbus TCP / IP ou Modbus TCP** - Il s'agit d'une variante Modbus utilisée pour les communications sur les réseaux TCP / IP , se connectant sur le **port 502**. Elle ne nécessite pas de calcul de somme de contrôle (*checksum calculation*), car les couches inférieures offrent déjà une protection de somme de contrôle.

4. **Modbus over TCP / IP** ou **Modbus RTU / IP** - Il s'agit d'une variante Modbus qui diffère de Modbus TCP en ce qu'une somme de contrôle est incluse dans la charge utile comme avec Modbus RTU.
5. **Modbus sur UDP** - L'utilisation de Modbus sur UDP sur les réseaux IP, élimine les conditions requises pour TCP . [sept]
6. **Modbus Plus (Modbus +, MB + ou MBP)** - Modbus Plus est propriétaire de **Schneider Electric** et, contrairement aux autres variantes, il prend en charge les communications **peer-to-peer** entre plusieurs maîtres. Il nécessite un coprocesseur dédié pour gérer la rotation rapide des jetons. Il utilise une paire torsadée à **1 Mbit/s**. Un matériel spécial est requis pour connecter Modbus Plus à un ordinateur, généralement une carte conçue pour le bus ISA (*Industry Standard Architecture*), PCI (*Peripheral Component Interconnect*).
7. **Pemex Modbus** - Il s'agit d'une extension du Modbus standard avec prise en charge des données historiques et de flux. Il a été conçu pour la **société pétrolière et gazière Pemex** pour une utilisation dans le contrôle des processus et n'a jamais été largement adopté.
8. **Enron Modbus** - Il s'agit d'une autre extension du Modbus standard développé par **Enron Corporation** Les principales différences entre les deux protocoles sont la numérotation des adresses de registre, la prise en charge des registres 32 bits et 16 bits et la capacité de transmettre des journaux d'événements et des données historiques.

- Le modèle de données et les appels de fonction sont **identiques pour les 4 premières variantes** de protocoles; seule l'encapsulation est différente. Cependant, les variantes ne sont pas interopérables, pas plus que les formats de trame.



Exemple d'une Architecture d'un réseau MODBUS

Communications et appareils

- Chaque appareil communiquant (transférant des données) sur un Modbus reçoit une adresse unique.
- Sur **Modbus RTU**, **Modbus ASCII** et **Modbus Plus**, qui sont tous des réseaux multipoints à câble unique **Rs-485**, seul le nœud affecté en tant que maître peut lancer une commande. Tous les autres appareils sont esclaves et répondent aux demandes et commandes.
- Pour les protocoles utilisant **Ethernet** tels que **Modbus TCP**, n'importe quel appareil peut envoyer une commande Modbus, donc tous peuvent agir en tant que maître, bien que normalement, un seul appareil agisse en tant que maître.
- Il existe de nombreux **modems** et **passerelles** qui prennent en charge Modbus, car il s'agit d'un protocole très simple et très répondu surtout pour le SCADA. Certains d'entre eux ont été spécialement conçus pour ce protocole. Différentes implémentations utilisent la communication filaire, sans fil, comme dans la bande **ISM** , et même le service de messages courts (**SMS**) ou le service général de radiocommunication par paquets (**GPRS**). L'une des conceptions les plus courantes des réseaux sans fil utilise la mise en réseau maillée . Les problèmes typiques que les concepteurs doivent surmonter comprennent des problèmes de latence élevée et de synchronisation.

Exemple de Modems Modbus RTU

Serial Modbus Modem Gsm Modem Vending Machine Gprs Industrial Gprs Modem With Io Rs232 Rs485 For Scada

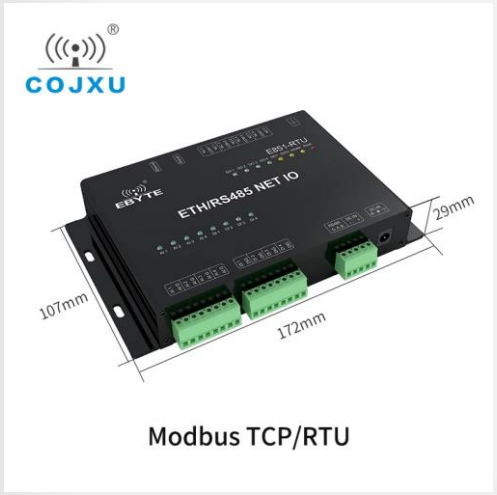
Gprs 3g Modem Gsm Modbus I/s Avec Rs232/rs485

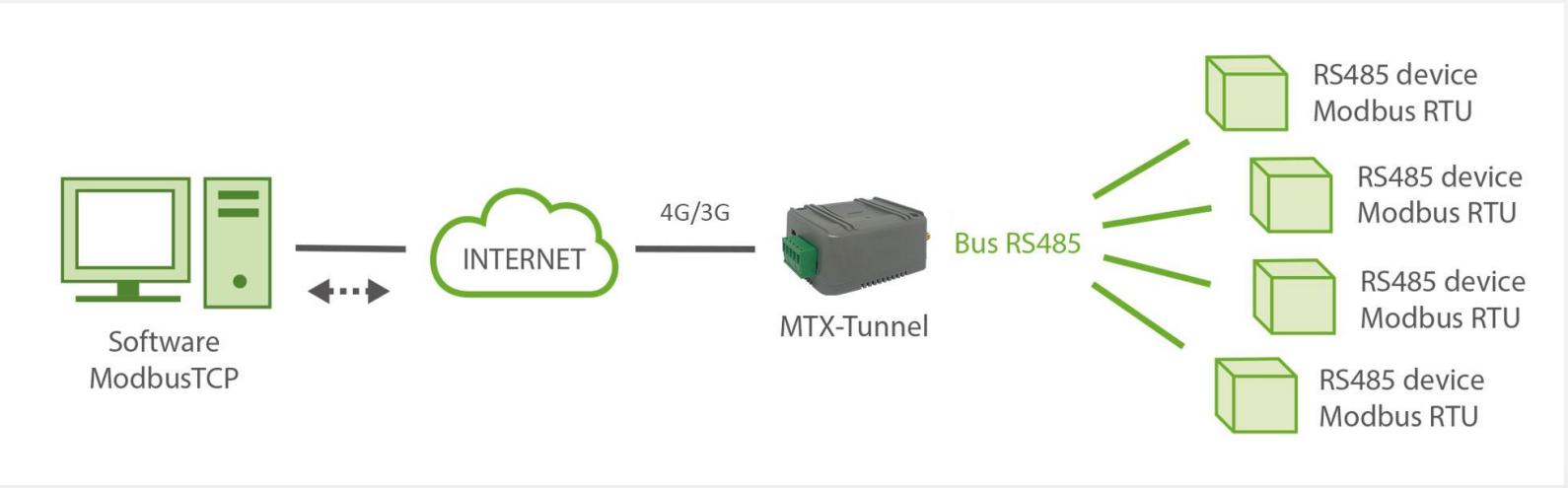
Modem Modbus RTU de Port GSM GPRS de service industriel RS485 de passerelle IOT cellulaire 4G pour le serveur SCADA OPC S275

Exemple de Modems Modbus TCP

Ethernet/ Modbus Tcp Ip Gprs Gateway

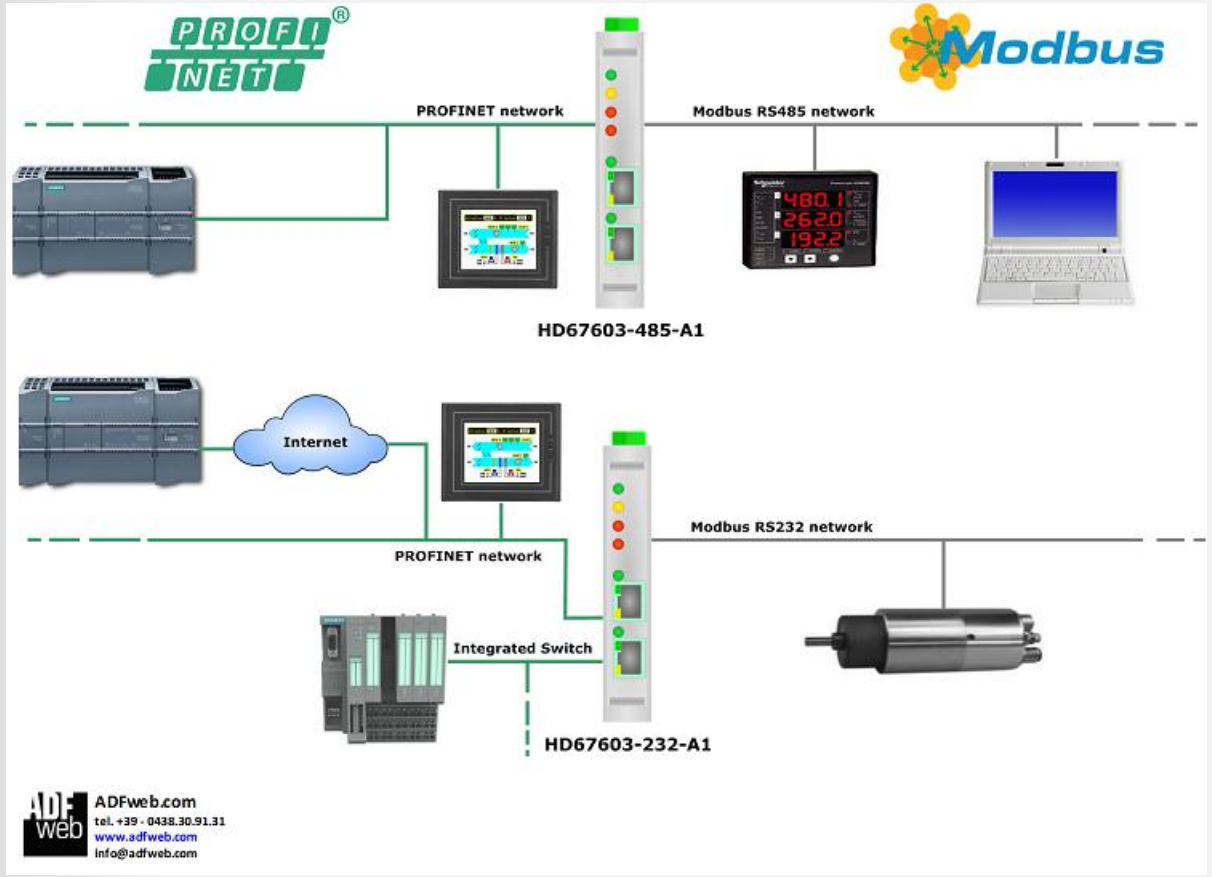
Cellular Gateway Modem RTOS System Modbus TCP protocol, Can Integrate to SCADA,HMI,DSC directly S473

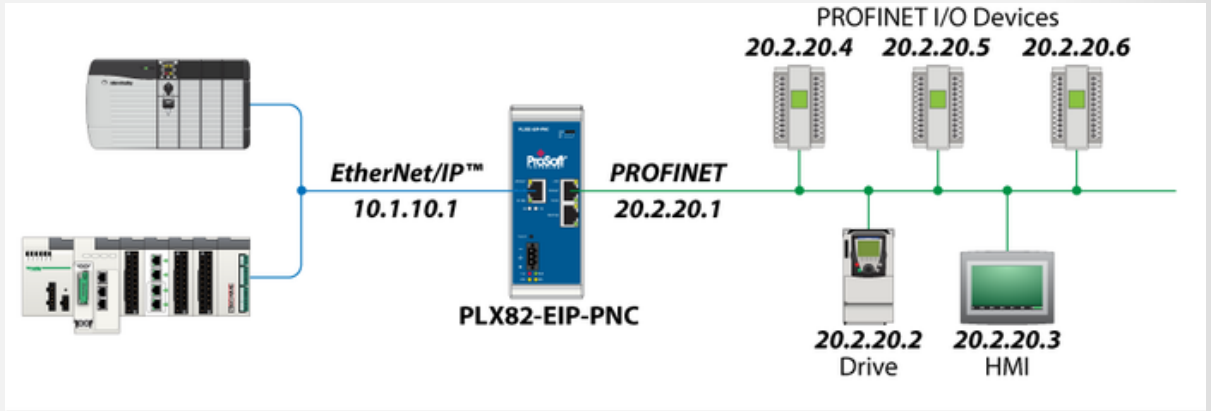


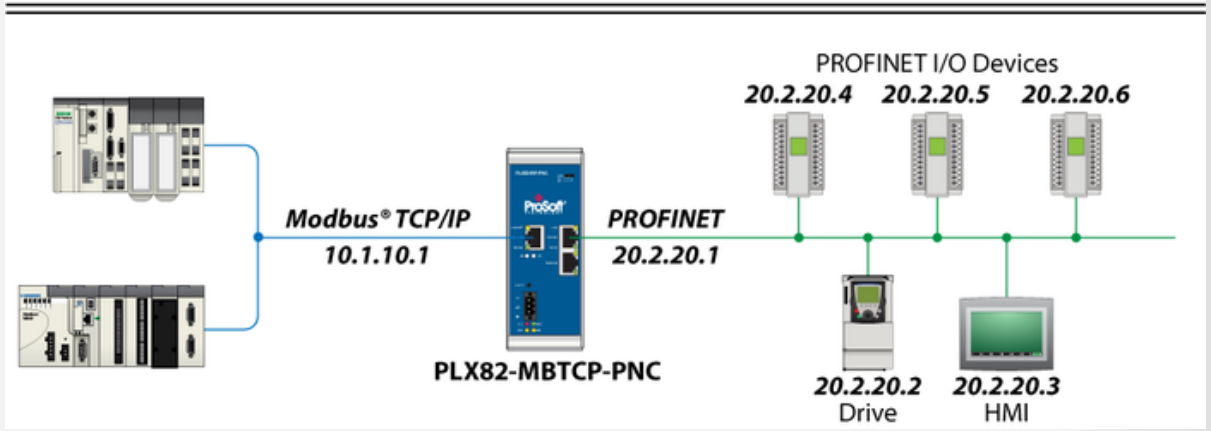


Passerelle Modbus TCP/RTU (Ethernet/RS-485)

Exemple de passerelles (Bridges) Modbus







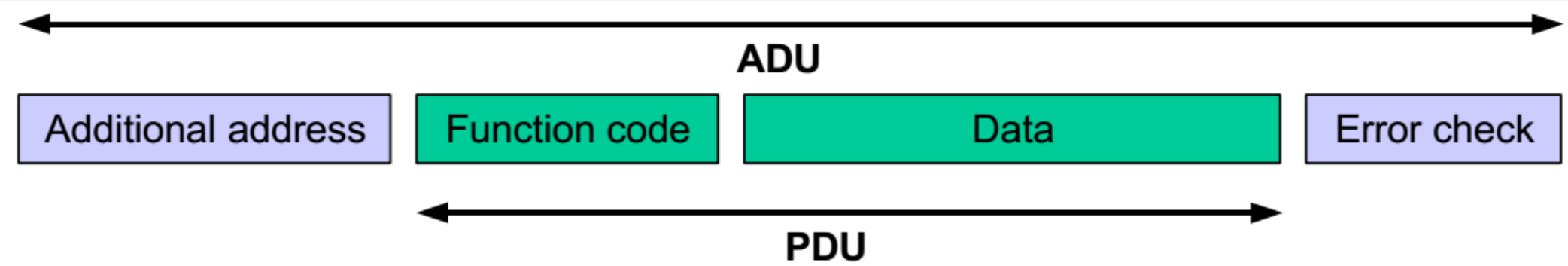
Protocole Modbus

Le protocole MODBUS permet une communication facile dans tous les types d'architectures de réseaux.

Tous les types d'appareils (API, IHM, panneau de commande, pilote, Motion control, périphérique d'E/S…) peuvent utiliser le protocole MODBUS pour lancer une opération à distance.

La même communication peut se faire aussi bien sur une ligne série que sur un réseau Ethernet TCP / IP. Les passerelles permettent une communication entre plusieurs types de bus ou de réseaux en utilisant le protocole MODBUS.

Le protocole MODBUS définit une unité de données de protocole simple *Protocol Data Unit* (**PDU**) indépendante des couches de communication sous-jacentes. Le mappage du protocole MODBUS sur des bus ou réseaux spécifiques peut introduire des champs supplémentaires sur l'unité de données d'application *Application Data Unit* **(ADU).**



Trame Modbus générale

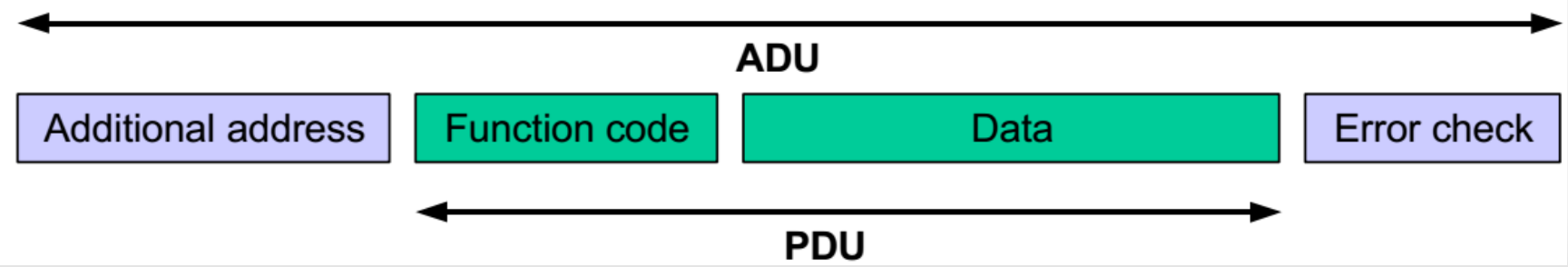
L'unité de données d'application MODBUS est créée par **le client qui initie une transaction MODBUS**.

La fonction indique au serveur le type d'action à effectuer. Le protocole d'application MODBUS établit le format d'une demande initiée par un client.

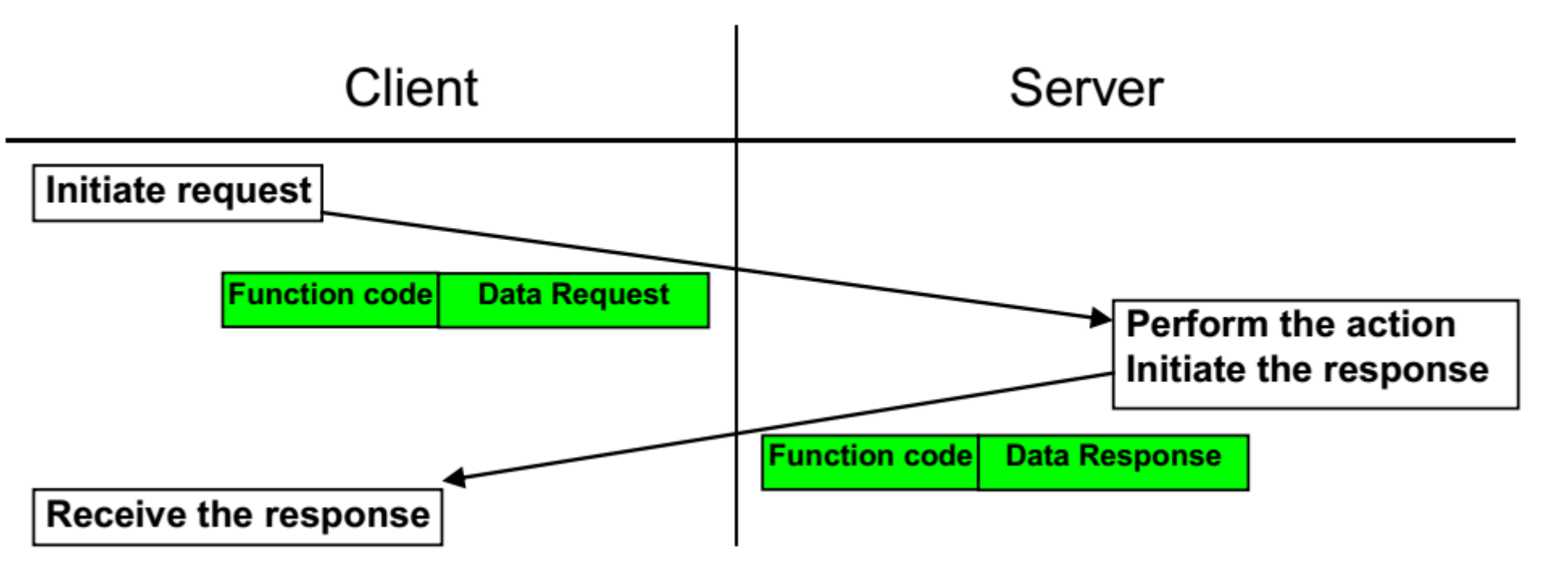
Le champ de **code de fonction** (*function code*) d'une unité de données MODBUS est codé sur **un octet**. Les codes valides sont compris entre **1 et 255** décimaux (la plage 128 à 255 est réservée et utilisée pour les réponses d'exception). Lorsqu'un message est envoyé d'un client vers un périphérique serveur, le champ de code de fonction indique au serveur le type d'action à effectuer. **Le code de fonction "0" n'est pas valide**.

Des **codes de sous-fonction** (*Sub-function codes)* sont ajoutés à certains codes de fonction pour définir plusieurs actions.

Le champ de données (DATA) des messages envoyés d'un client aux périphériques du serveur contient des informations supplémentaires que le serveur utilise pour effectuer l'action définie par le code de fonction. Cela peut inclure des éléments tels que les adresses discrètes et de registre, la quantité d'éléments à gérer et le nombre d'octets de données réels dans le champ. Le champ de données peut être inexistant (**de longueur nulle**) dans certains types de demandes, dans ce cas, le serveur ne nécessite aucune information supplémentaire. Le code de fonction spécifie à lui seul l'action.

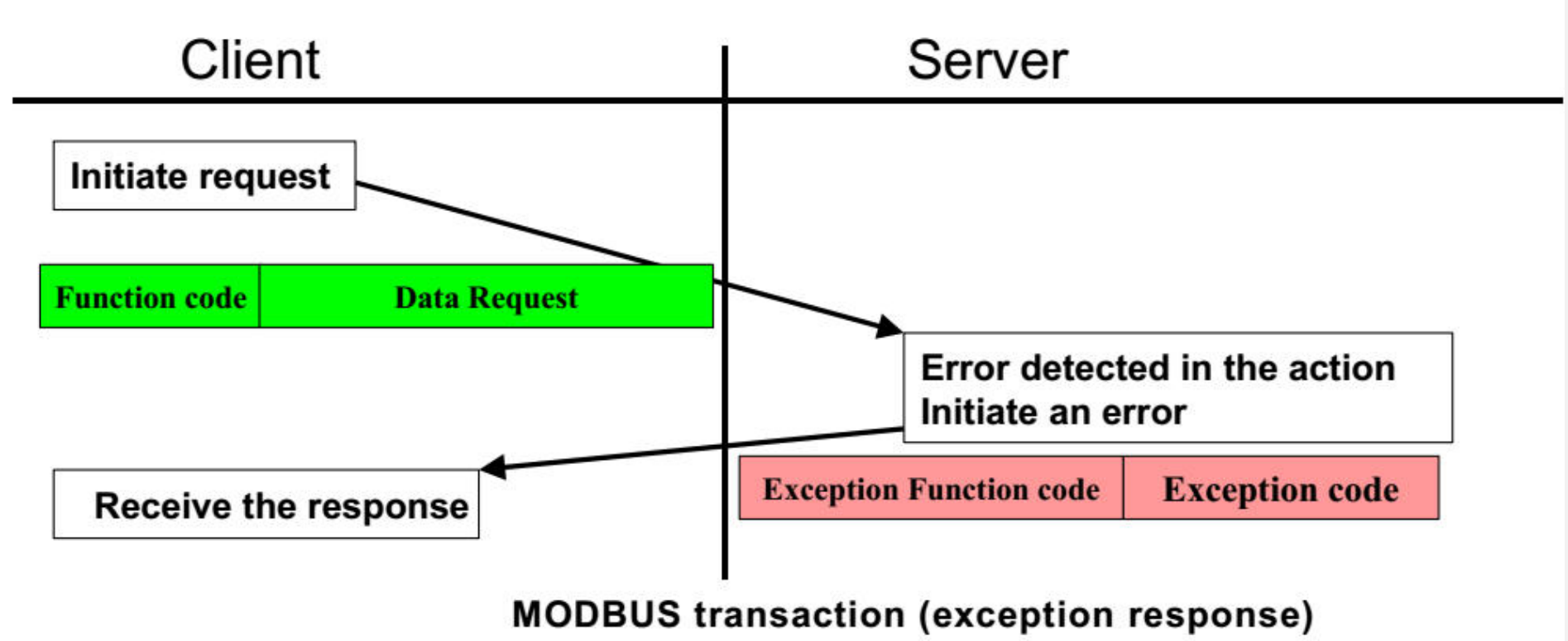


Si aucune erreur ne se produit concernant la fonction MODBUS demandée dans une ADU MODBUS correctement reçue, le champ de données d'une réponse d'un serveur à un client contient les données demandées. Par exemple, un client peut lire les états ON / OFF d'un groupe de sorties ou d'entrées discrètes ou il peut lire / écrire le contenu des données d'un groupe de registres.



Si une erreur liée à la fonction MODBUS demandée se produit, le champ contient un code d'exception que l'application serveur peut utiliser pour déterminer la prochaine action à entreprendre.

Lorsque le serveur répond au client, il utilise le champ de code de fonction pour indiquer une réponse normale (sans erreur) ou qu'une sorte d'erreur s'est produite (appelée **réponse d'exception**). Pour une réponse normale, le serveur fait simplement écho à la requête le code de fonction d'origine.



- La taille de la PDU MODBUS est limitée par la contrainte de taille héritée de la première Implémentation MODBUS sur le réseau Serial Line (max. ADU RS485= 256 octets).

  Par conséquent, PDU MODBUS pour ligne série = 256 - Adresse du serveur (1 octet) - CRC (2 octets) = 253 octets.

  Par conséquent:

  - ADU RS232 / RS485 = 253 octets + adresse du serveur (1 octet) + CRC (2 octets) = 256 octets.
  - TCP MODBUS ADU = 253 octets + MBAP (7 octets) = 260 octets.

- Le protocole MODBUS définit trois PDU. Elles sont :

  - PDU de demande MODBUS, **mb_req_pdu**

mb_req_pdu = {function_code, request_data}, where
function_code = [1 byte] MODBUS function code,
request_data = [n bytes] This field is function code dependent and usually contains information such as variable references, variable counts, data offsets, sub-function codes etc.

  - PDU de réponse MODBUS, **mb_rsp_pdu**

mb_rsp_pdu = {function_code, response_data}, where
function_code = [1 byte] MODBUS function code
response_data = [n bytes] This field is function code dependent and usually contains information such as variable references, variable counts, data offsets, sub-function codes, etc.

  - PDU de réponse d'exception MODBUS, **mb_excep_rsp_pdu**

mb_excep_rsp_pdu = {exception-function_code, request_data}, where
exception-function_code = [1 byte] MODBUS function code + 0x80
exception_code = [1 byte] MODBUS Exception Code Defined in table "MODBUS Exception Codes" (see section 7).

Modèle de données MODBUS

❑ Encodage des données

MODBUS utilise une représentation «**Gros-boutiste**» (big-endian) pour les adresses et les éléments de données. Cela signifie que lorsqu'une quantité numérique supérieure à un seul octet est transmise, **l'octet le plus significatif est envoyé en premier**. Donc par exemple:

Taille du Registre	**Valeur (Hex)**	**Le premier octet envoyé est 0x12 puis 0x34**
16 – bits	**0 x 12 34**	

❑ Modèle de données MODBUS

MODBUS base son modèle de données sur une série de tableaux qui ont des caractéristiques distinctives. Les quatre tables principales sont:

❑ Modbus Data Model

Primary tables	Object type	Type of access	Comments
Discretes Input	Single bit	Read-Only	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system
Holding Registers	16-bit word	Read-Write	This type of data can be alterable by an application program.

Primary tables	Object type	Type of access	Comments
Entrée Discrète	1 bit	Lecture-seule	Ce type de données peut être fourni par un système d'E / S
Bobine	1 bit	Lire-écrire	Ce type de données peut être modifié par un programme d'application.
Registres d'entrée	Mot 16 bits	Lecture-seule	Ce type de données peut être fourni par un système d'E / S
Registres Généraux	Mot 16 bits	Lire-écrire	Ce type de données peut être modifié par un programme d'application.

In this video

- How Modbus communication protocol works between devices.

https://www.youtube.com/watch?v=JBGaInI-TG4&t=5s

REALPARS

Modèle d’adressage MODBUS

Le protocole d'application MODBUS définit précisément les règles d'adressage des PDU.

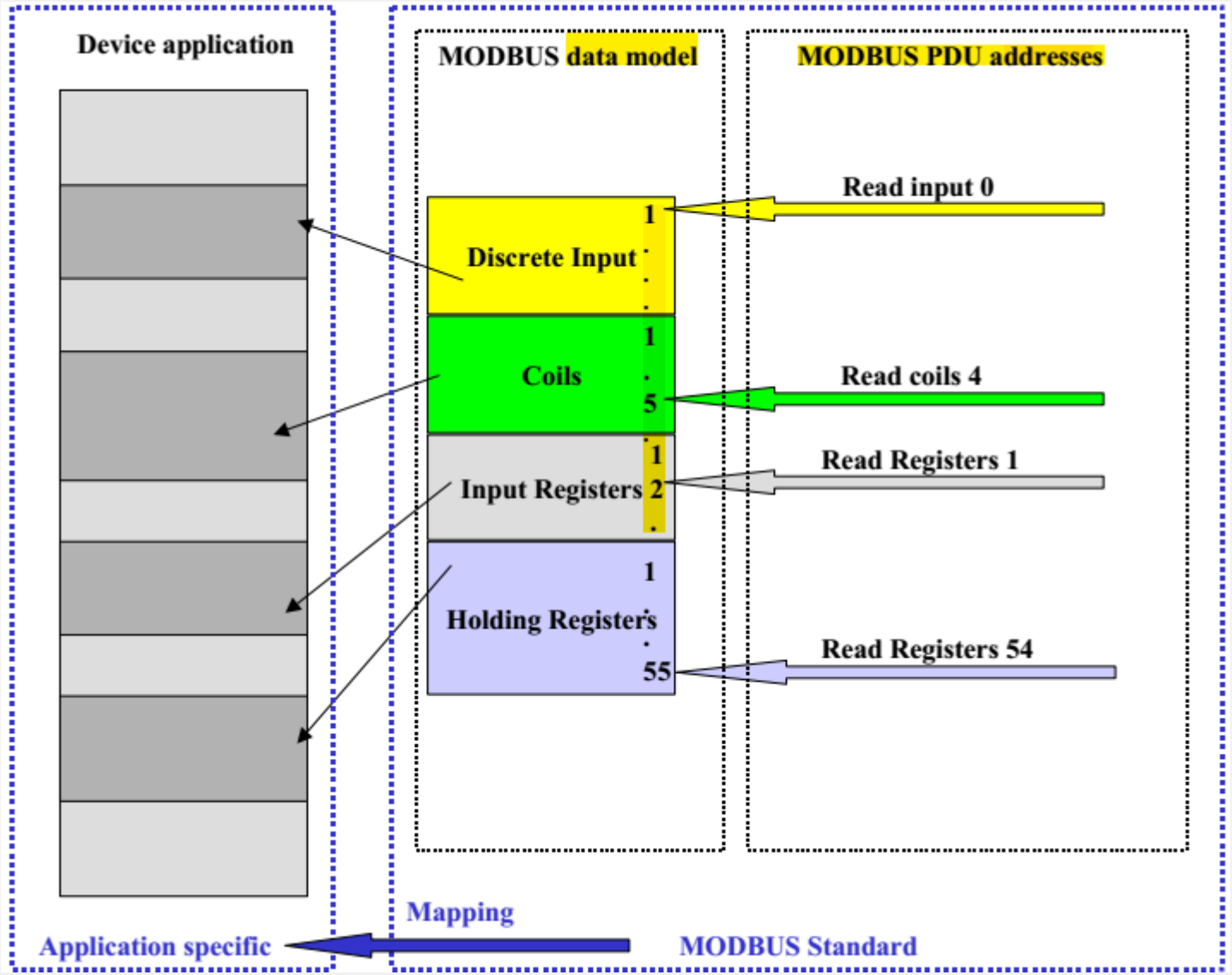
Dans une PDU MODBUS, chaque donnée est adressée de 0 à 65535.

Il définit également clairement un modèle de données MODBUS composé de 4 blocs comprenant plusieurs éléments numérotés de 1 à n.

Dans le modèle de données MODBUS, chaque élément d'un bloc de données est numéroté de 1 à n.

Ensuite, le modèle de données MODBUS doit être lié à l'application de l'appareil (IEC-61131)

La pré-cartographie entre le modèle de données MODBUS et l'application de l'appareil est totalement spécifique à l'appareil du fournisseur



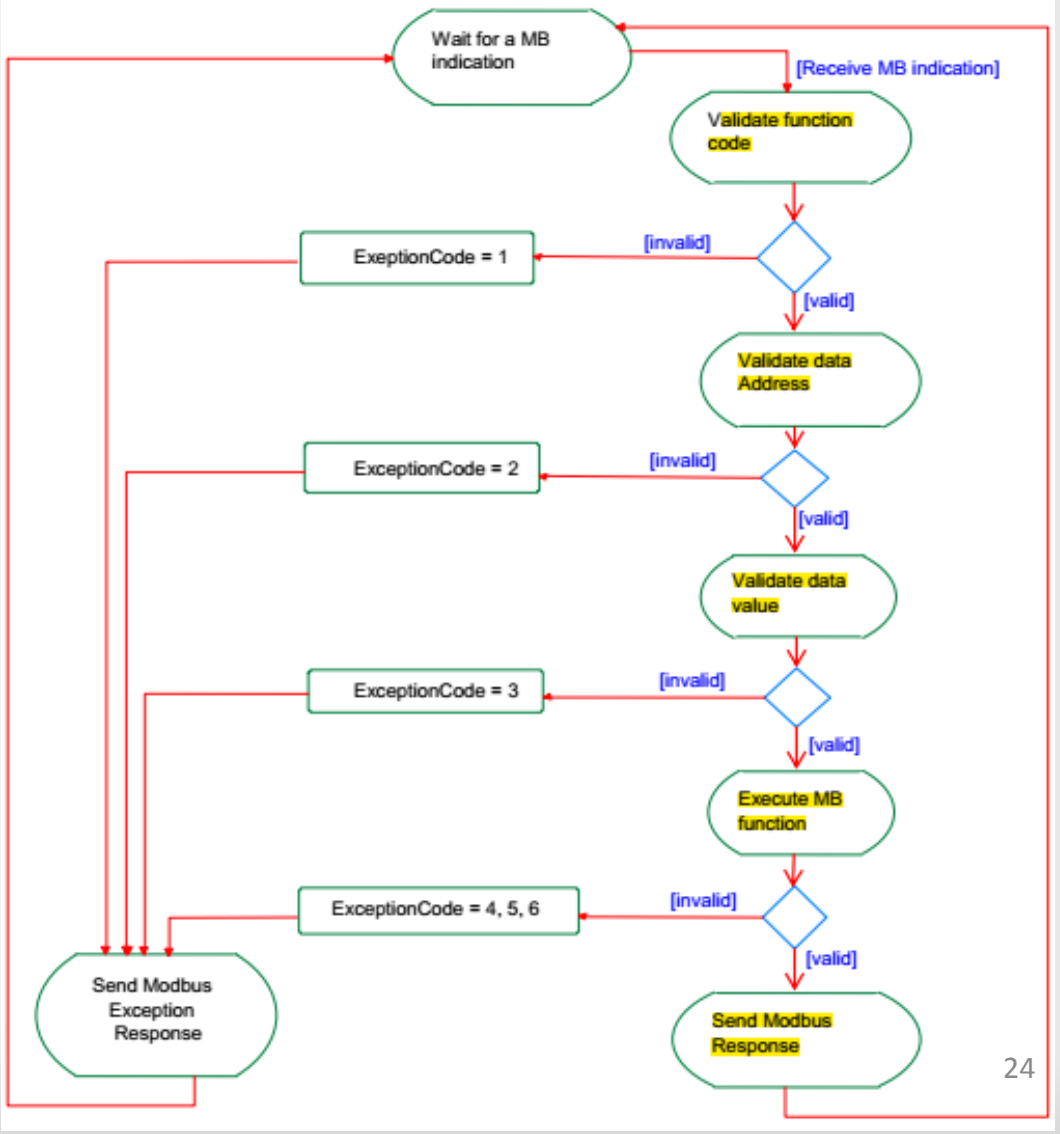
Modèle d'adressage Modbus

Transaction MODBUS

Une fois la demande traitée par un serveur, une réponse MODBUS utilisant une transaction de serveur MODBUS adéquate est créée.

En fonction du résultat du traitement, deux types de réponse sont construits:

- Une réponse MODBUS positive:
  - ✓ le code de fonction de réponse = le code de fonction de demande
- Une réponse d'exception MODBUS:
  - ✓ l'objectif est de fournir au client des informations pertinentes concernant le erreur détectée lors du traitement;
  - ✓ le code de fonction d'exception = le code de fonction de demande + 0x80;
  - ✓ un code d'exception est fourni pour indiquer la raison de l'erreur.



Catégorie de codes de fonction

Il existe trois catégories de codes de fonctions MODBUS. Elles sont :

Codes de fonction publique

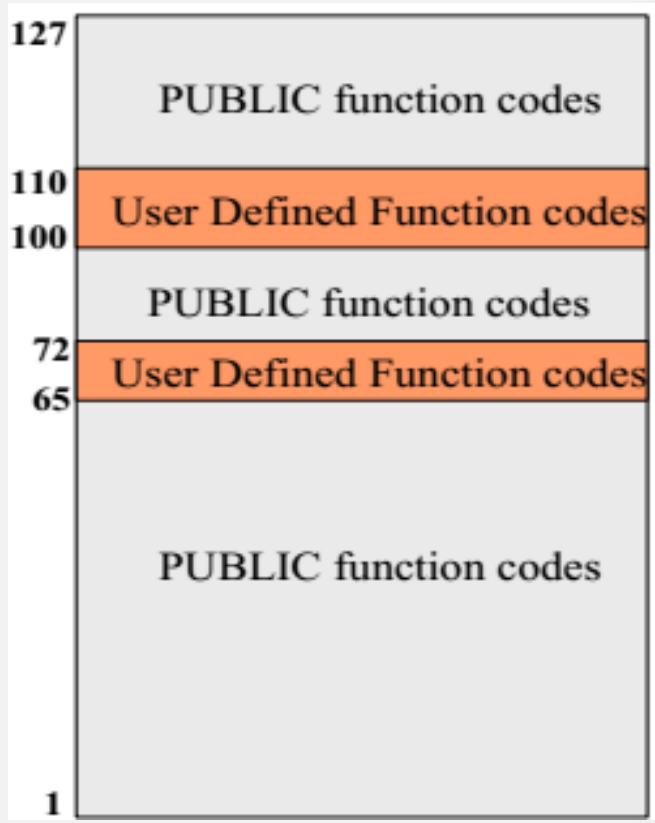
- sont des codes de fonction bien définis,
- garantie d'être unique,
- validée par la communauté MODBUS.org,
- codes réservés pour une utilisation future.

Codes de fonction définis par l'utilisateur

- Il existe deux plages de codes de fonction définis par l'utilisateur, à savoir 65 à 72 et de 100 à 110 décimal.
- L'utilisateur peut sélectionner et implémenter un code de fonction qui n'est pas pris en charge par le spécification.

Codes de fonction réservés

- Codes de fonction actuellement utilisés par certaines entreprises pour les produits hérités et ne sont pas disponibles pour un usage public.



Codes de fonction publique

				Function Codes			
				code	*Sub code*	*(hex)*	*Section*
Data Access	**Bit access**	Physical Discrete Inputs	Read Discrete Inputs	**02**		**02**	6.2
		Internal Bits Or Physical coils	Read Coils	**01**		**01**	6.1
			Write Single Coil	**05**		**05**	6.5
			Write Multiple Coils	**15**		**0F**	6.11
	16 bits access	Physical Input Registers	Read Input Register	**04**		**04**	6.4
		Internal Registers Or Physical Output Registers	Read Holding Registers	**03**		**03**	6.3
			Write Single Register	**06**		**06**	6.6
			Write Multiple Registers	**16**		**10**	6.12
			Read/Write Multiple Registers	**23**		**17**	6.17
			Mask Write Register	**22**		**16**	6.16
			Read FIFO queue	**24**		**18**	6.18
	File record access		Read File record	**20**		**14**	6.14
			Write File record	**21**		**15**	6.15
Diagnostics			Read Exception status	**07**		**07**	6.7
			Diagnostic	**08**	**00-18,20**	**08**	6.8
			Get Com event counter	**11**		**OB**	6.9
			Get Com Event Log	**12**		**0C**	6.10
			Report Server ID	**17**		**11**	6.13
			Read device Identification	**43**	**14**	**2B**	6.21
Other			Encapsulated Interface Transport	**43**	**13,14**	**2B**	6.19
			CANopen General Reference	**43**	**13**	**2B**	6.20

Références bibliographiques

- https://www.youtube.com/user/ParsicAutomation/featured
- Communication avec SIMATIC, Manuel système, 09/2006, EWA 4NEB 7106075-03 03.
- SIEMENS TIA PORTAL13_help
- MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3
- www.modbus.org