

# Modbus



Intégration LabVIEW-Arduino (novembre 2019)

# Objectifs

- Initier au protocole Modbus en négligeant certains détails
- Focuser sur son utilisation
- Prioriser LabVIEW et Arduino
- Rejoindre les cours projets, microcontrôleurs, acquisition, positionnement, automatisation



# Qu'est-ce que Modbus

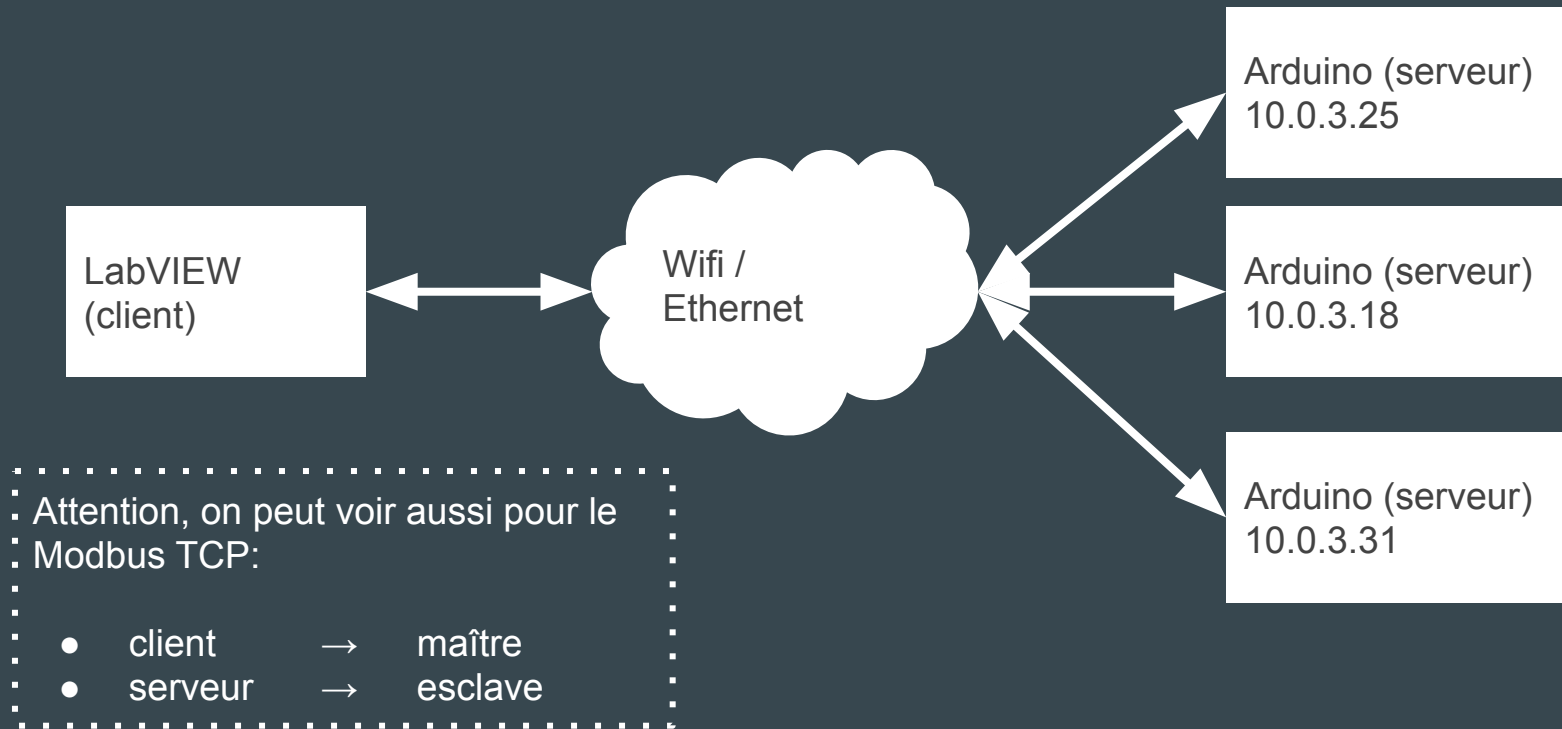
- Protocole de communication
- Vieux (1979) mais toujours pertinent et utilisé aujourd'hui
- Utilisé principalement dans le milieu industriel
- Utilisable pour l'IoT
- Protocole simple
  - Implémentation
  - Compréhension et utilisation
- Bibliothèques
  - Arduino → simple
  - LabVIEW → encore plus simple

# Trois saveurs... mais deux présentées

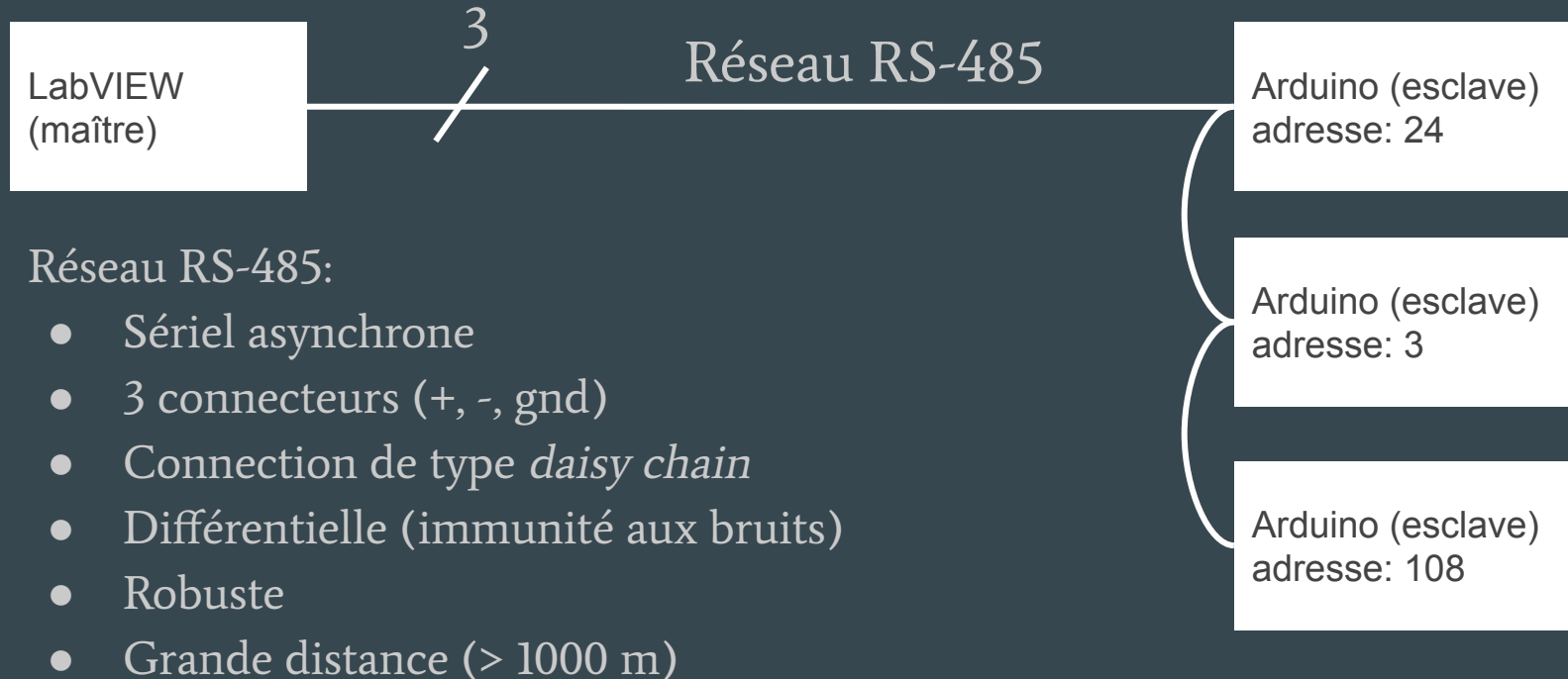
Modbus RTU	Protocole sériel asynchrone <ul style="list-style-type: none"><li>• RS-232</li><li>• RS-485</li><li>• USB *</li></ul>
Modbus TCP	Protocole “internet” <ul style="list-style-type: none"><li>• Ethernet → le fils bleu</li><li>• Wifi</li></ul>

(\*) Attention, vrai seulement pour communication COM-Virtuel comme pour l'Arduino.

# Modbus TCP

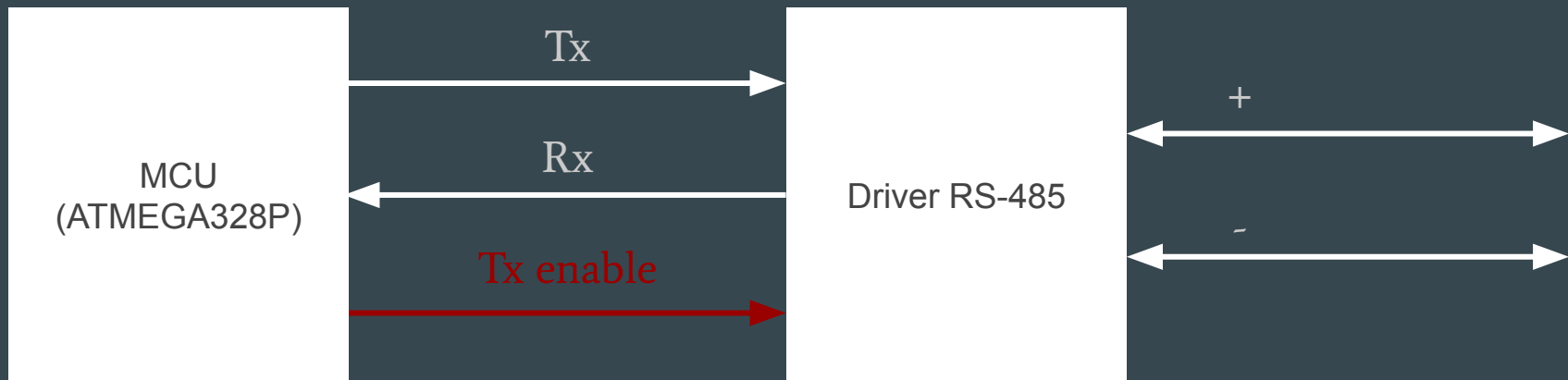


# Modbus RTU (RS-485)



# Un détail sur RS-485

Le signal supplémentaire *Tx enable* est nécessaire. Il doit être spécifié en Modbus RTU uniquement par RS-485.



Ici le commun (GND) est négligé pour simplification.

# Modbus RTU (UART, RS-232 ou USB via COM virtuel)



Connection UART, RS-232 ou USB via COM virtuel:

- Série asynchrone
- 3 connecteurs (Tx, Rx, gnd) sauf pour USB
- Une seule connection
- UART: peu robuste
- RS-232: robuste mais moins distance que RS-485
- USB: comme le USB usuel (utilisation de *hub* USB incertaine)

L'adresse peut prendre  
une valeur de 0 à 247.



# Modbus maître et esclave



Envoie des requêtes à un esclave donné par son adresse

Ne répond qu'aux requêtes (initie aucune communication)

# Fonctions

- Client/maître demande une fonction lors d'une requête.
- Chaque fonction a une description et un code.
- 21 fonctions disponibles, exemples:
  - (1) Read Coils;
  - (24) Read FIFO queue
  - etc.

Fonctions	Code
Read Holding Registers	3
Write Single Register	6
Write Multiple Registers	16

Dans notre cas, nous n'allons utiliser que les fonctions présentées dans le tableau, soient celles utilisant les *holding registers*.

# Mécanisme avec fonctions 3, 6 et 16 (*holding registers*)



- Les registres sont localisés dans l'esclave.
- Ils sont de 16 bits et ont tous une adresse spécifique.
- L'esclave peut lire/écrire dans les registres à l'interne sans communication (peu dispendieux).
- Le maître peut lire/écrire dans les registres d'un esclave donné par son adresse (exemple 10.0.3.25) à travers le réseau.

Adresses	Registres
0	0x3456
1	0x0001
2	0x2100
3	0x0400

← 16 bits →

## **Attention**

**Les esclaves ont des adresses mais  
les registres aussi → deux types  
d'adresse différentes**

# Exemples

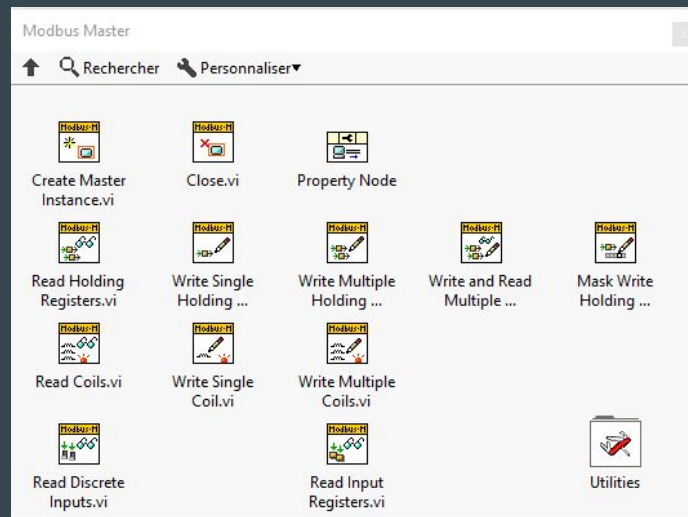
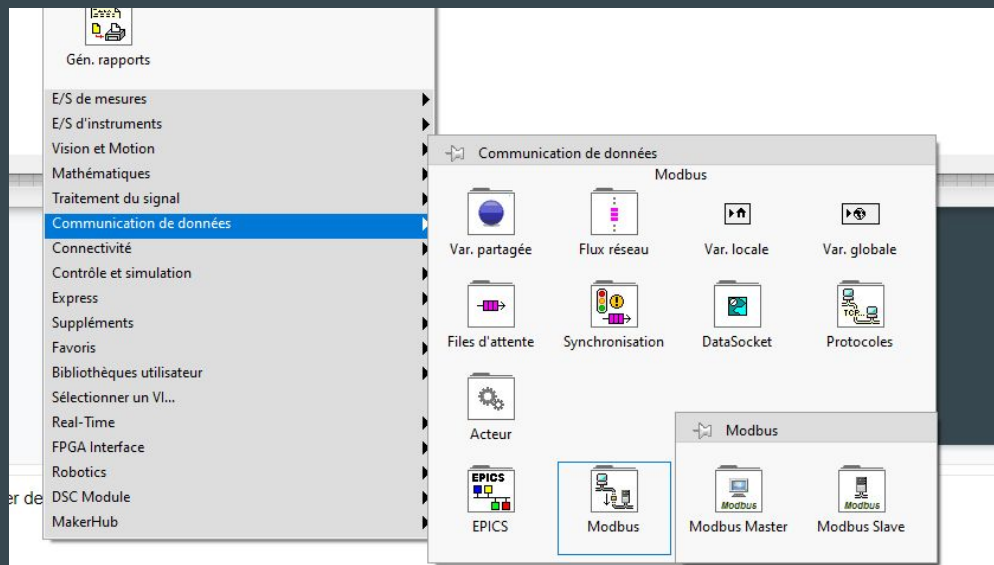


1. L'esclave (Arduino) possède un capteur de température et un relais.
2. La mesure de température → registre de l'adresse 0.
3. La commande pour actionner le relais → registre de l'adresse 1.
4. L'esclave met-à-jour la température et lit la commande du relais de façon continue.
5. Le maître (LabVIEW) peut écrire ou lire les registres de l'esclave (10.0.3.25) quand cela est nécessaire à travers le réseau afin d'obtenir la température et de commander le relais aux besoins.

Adresses	Registres
0	Température
1	Relais

← 16 bits →

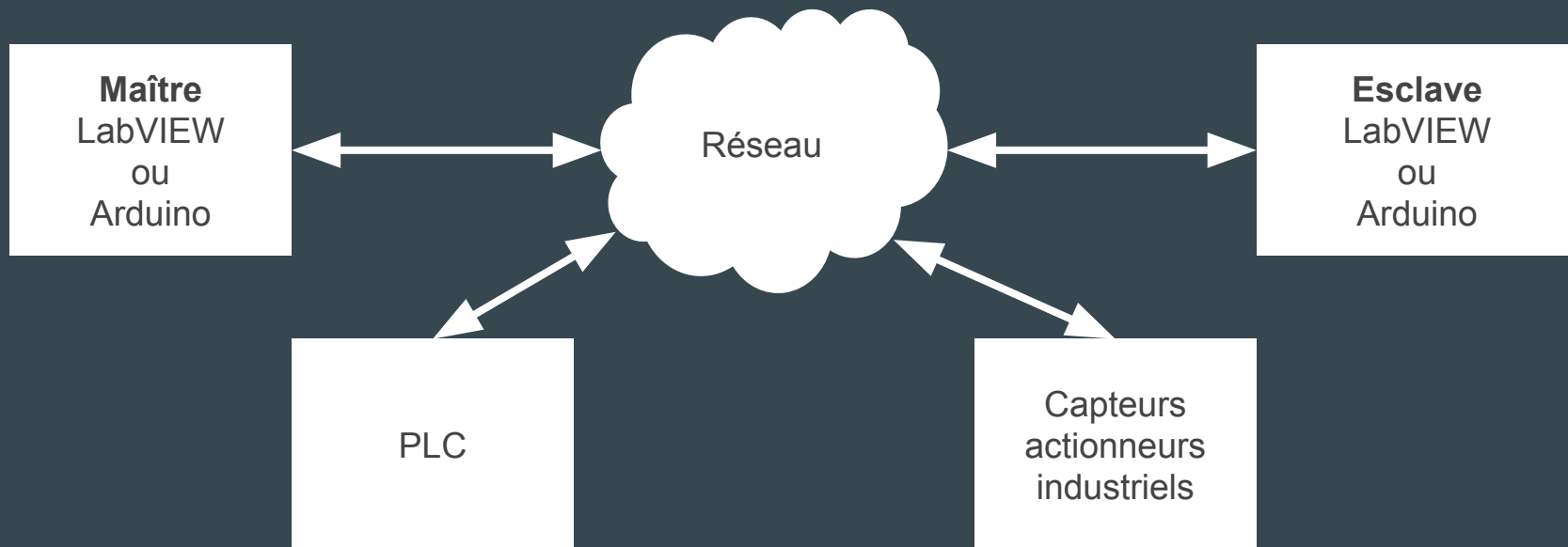
# Libraries - LabVIEW



# Librairies - Arduino

Cible	Librairie	Utilisabilité
Modbus TCP	Standard d'Arduino: <a href="https://www.arduino.cc/en/ArduinoModbus/ArduinoModbus">https://www.arduino.cc/en/ArduinoModbus/ArduinoModbus</a>	Facile
Modbus RTU (Cartes de type MKR)	Standard d'Arduino: <a href="https://www.arduino.cc/en/ArduinoModbus/ArduinoModbus">https://www.arduino.cc/en/ArduinoModbus/ArduinoModbus</a>	Facile
Modbus RTU	Basé sur simple-modbus de Bester Juan: <a href="https://github.com/jecrespo/simple-modbus">https://github.com/jecrespo/simple-modbus</a>	Très facile

# Maîtres et esclaves



- Les librairies Arduino et LabVIEW peuvent agir en tant que maître ou esclave.
- D'autres équipements (automates programmables, capteurs, actionneurs industriels) peuvent faire partis du réseau en tant qu'esclave ou maître.



# Outil de développement

- Une application PC (qModMaster) permet d'utiliser un PC comme maître Modbus en RTU ou TCP.
- Il est constitué d'une application .exe
  - Donc pas d'installateur
  - Donc pas besoin de privilèges administrateurs
- On peut donc s'en servir pour tester des esclaves présents dans un réseau Modbus facilement sans développement.

<https://sourceforge.net/projects/qmodmaster/>

# Exemples et vidéos

- Trois vidéos présentent l'intégration Modbus à l'aide de LabVIEW et d'Arduino.
- Les exemples ainsi que les liens aux vidéos sont présents sous github:

<https://github.com/jscastonguay/Integration-Arduino-LabVIEW-par-Modbus>

# Références

Le document source de la présentation est sous: *google drive* → *Recherche et projets*  
→ *Intégration Modbus LabVIEW-Arduino* → *doc*

ou au lien suivant:

[https://docs.google.com/presentation/d/1V-Ni3zOW\\_Gx-0V24tbwfoHCnR0y0QReOma5Mkps-YSo/edit?usp=sharing](https://docs.google.com/presentation/d/1V-Ni3zOW_Gx-0V24tbwfoHCnR0y0QReOma5Mkps-YSo/edit?usp=sharing)