Système Numérique
Informatique et Réseaux

TP n°2 Stockage Dynamique
Pilotage des Éjecteurs et du Moteur

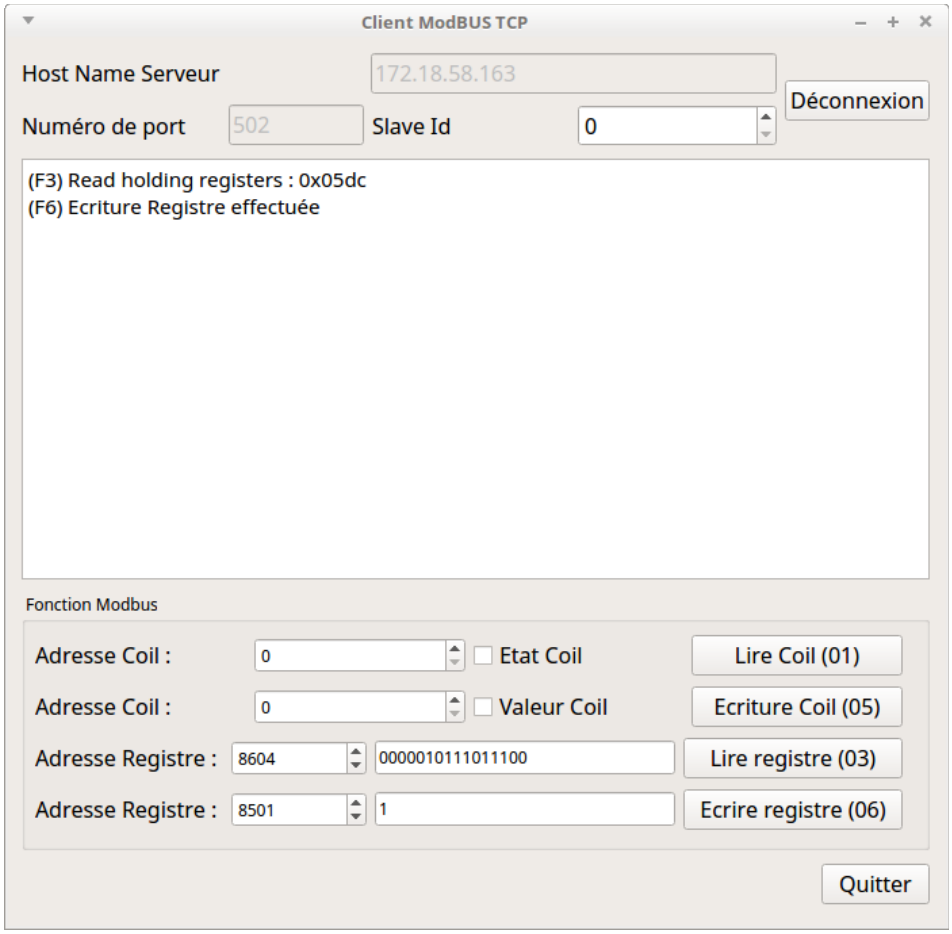
Modbus – ModBus / TCP

Ce TP est destiné à étudier les trames Modbus / TCP. Vous disposez de deux outils logiciel pour la simulation, http://www.modbustools.com/download.html un module maître et un module esclave. Ils pourront être utilisés sur le même poste dans une machine virtuelle fonctionnant sous windows ou sur 2 postes distants via Ethernet.

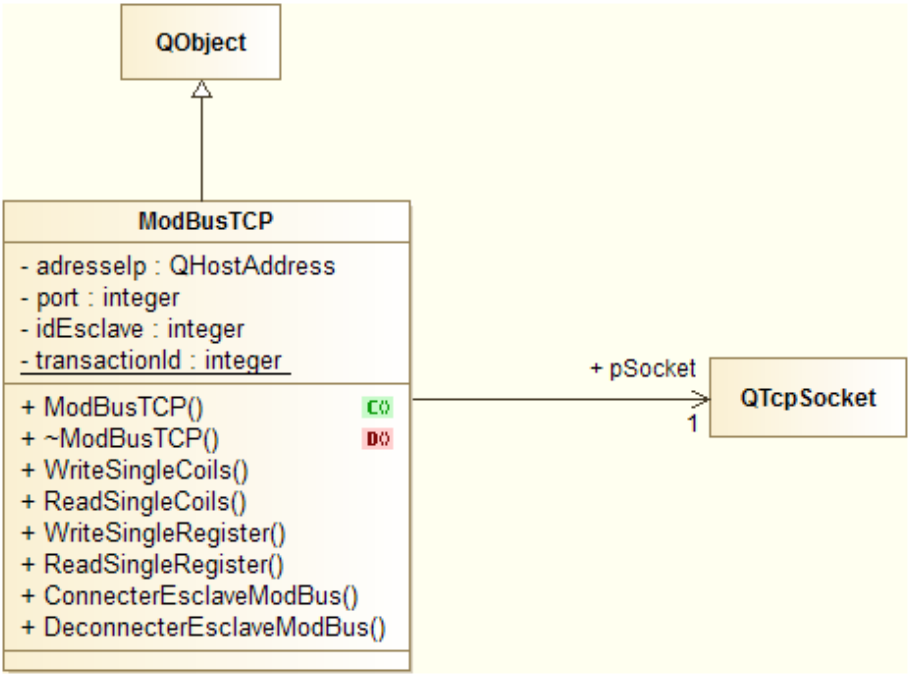
Vous pourrez également effectuer des tests sur des matériels Wago et Shneider (variateur de fréquence ATV32). Pour ce faire des fiches de synthèse sont fournies.

Codage

5. *Réaliser un projet en C++ sous QT utilisant les QSocket TCP/IP pour dialoguer avec l'esclave nommé TestModBus. Cette application doit permettre de visualiser et de commander les Entrées/Sorties sous la forme de registres et de coils comme le montre la figure ci-dessous.*



6. *Réaliser une classe ModBusTCP héritant de QObject qui contiendra le code nécessaire pour dialoguer en utilisant le protocole ModBus sous TCP.*



Remarque *: L'attribut transactionId est un attribut de classe. Reportez-vous à la documentation pour déterminer le type des paramètres des opérations et des différents attributs.*

7. *Le constructeur de la classe ModBusTcp réalise la l'initialisation de la socket,et la connexion de ces signaux : connected, disconnected et readyRead ajouter les slots correspondants dans votre classe.*

8. *Ajouter le signal* ***signalEtatConnexion*** *précisant sous la forme d'une QString l'état de la connexion à la fenêtre principale.*

Codez les méthodes : ***ConnecterEsclaveModBus*** ***DeconnecterEsclaveModBus****.*

Ansi que les slots correspondants correspondant aux signaux ***connected*** *et* ***disconnected*** *sans oublier d'émettre le signal pour indiquer l'état.*

10. *Déclarez une constante énumérée CODE_FONCTION regroupant les premiers codes fonctions du protocole modBus.*

11. *Définir les constantes suivantes : _MODBUS_TCP_HEADER_LENGTH, _CODE_EXCEPTION*

12. *Codez l'écriture d'un bit (Coils)* ***WriteSingleCoils****.Testez avec l'esclave ModBus*

13. *Réalisez le codage du slot correspondant au signal readyRead pour récupérer la réponse de l'esclave, elle va contenir par la suite les réponses pour les autres méthodes. Elle réalise également le traitement des exceptions.*

14. *Relever l'échange en utilisant Wireshark et analyser et comparer les résultats avec les trames relevées lors de la mise en œuvre.*

15. *Procédez de même pour les autres méthodes de la classe.*