TP de mise en œuvre du bus de terrain MODBUS

# Généralités

Rappeler les conditions d'utilisation des bus de terrain

Un bus de terrain est un système de communication entre plusieurs ensembles communicants dans une zone géographique limitée (calculateurs, automates, actionneurs,capteurs,…)

Donner les principales caractéristiques du protocole MODBUS

Choix de la parité.

Choix de la vitesse de transmission de 50 bits/s à 19200 bits/s.

Le protocole MOBUS ne définit que la structure des messages et leur mode d’échange.

On peut utiliser n’importe quel support de transmission RS232, RS422 ou RS485, mais la liaison RS RS485 est la plus répandue car elle autorise le « multipoints ».

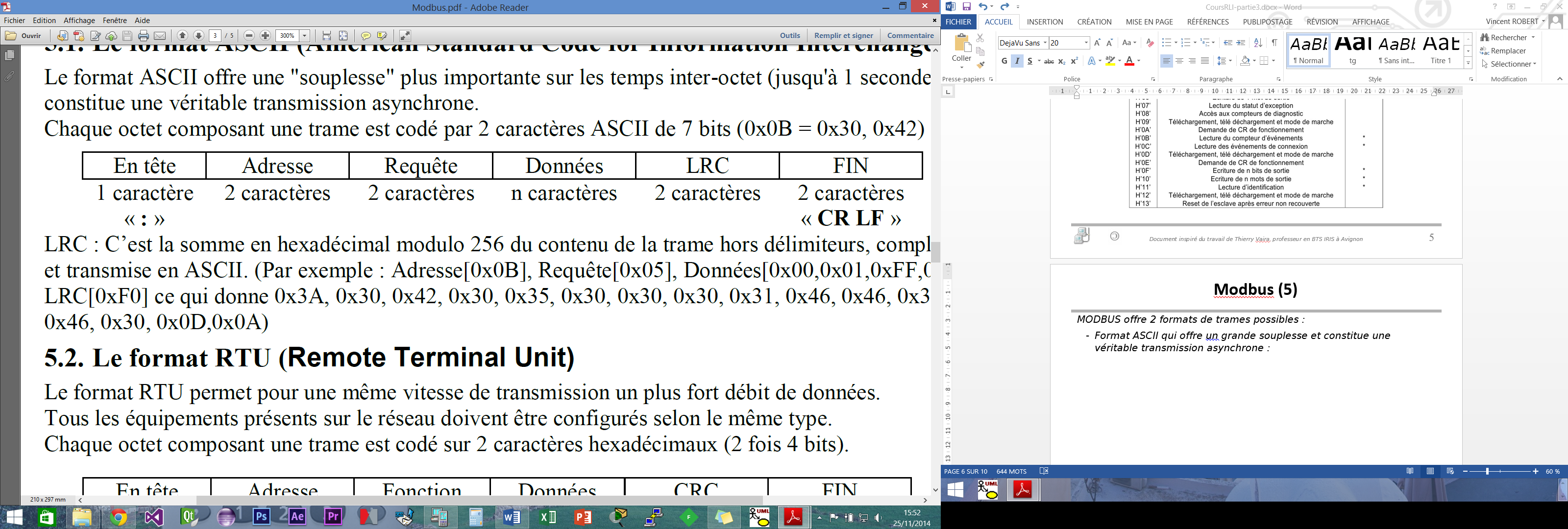
Quels sont les deux formats de trames possibles sous Modbus ?

Les deux formats de trams sont ASCII ou RTU.

# Calcul des mots de contrôle d'une trame MODBUS

## Cas du mode ASCII

Dans ce mode,



le LRC (Longitudinal Redundancy Check) permet de vérifier la validité de la trame. C’est la somme en hexadécimal modulo 256 (0x100) du contenu de la trame(avant codage en ASCII) hors délimiteurs, complémentée à 2. Après calcul, le LRC est codé par 2 caractères ASCII

Ex : Adresse[0x0B], Requête[0x05], Données[0x00,0x01,0xFF,0x00]

% signifie en binaire

0x signifie en hexa

On calcule la somme S=0x0B+0x05+0x00+0x01+0xFF+0x00 = 0x110

S modulo 0x100 = 0x16 = %0001 0000

Complément à 1 = %1110 1111 = 0xEF

Complément à 2 = % 1111 0000 = 0xF0

En ASCII, la trame est donc :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entête | Adresse | | Requête | | Données | | | | | | | | LRC | | FIN | |
| 0x3A | *0x30* | *0x42* | *0x30* | *0x35* | *0x30* | *0x30* | *0x30* | *0x31* | *0x46* | *0x46* | *0x30* | *0x30* | *0x46* | *0x30* | 0x0D | 0x0A |

* Codez une fonction C++ byte calculLRC(byte \*trame , int n ) qui calcule le LRC d'une trame de n octets. A ce stade, la trame et le LRC ne sont pas en ASCII. Testez cette fonction.

Copier ci-dessous le code de votre fonction et de votre programme principal de test et faire valider

|  |
| --- |
| Byte calculLRC(byte \*trame, int n)  {  byte LRC=0;  for (int i = 0; i < n; i++)  {  LRC += ~trame[i]+1;  }  return LRC;  }  int main()  {  byte trame[] = { 0x0B,0x05,0x00,0x01,0xFF,0x00 };  byte LRC=calculLRC(trame, 6);  cout <<"La valeur du LRC est de : " << hex << (int)LRC << endl;  \_getch();  return 0;  Validation enseignant 🞏  } |

* Codez une fonction C++ void toModBusAscii(byte \* trame, char \* trameASCII, int n) qui convertit au format Modbus ASCII une trame de n octets et ajoute le LRC et les délimiteurs. Testez cette fonction.

**Exemple :**

octets en entrée : 0B 05 00 01 FF 00

octets en sortie : 3A 30 42 30 35 30 30 30 31 46 46 30 30 46 30 0D 0A

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| En hexa | 3A | 30 | 42 | 30 | 35 | 30 | 30 | 30 | 31 | 46 | 46 | 30 | 30 | 46 | 30 | 0D | 0A |
| En ASCII | : | 0 | B | 0 | 5 | 0 | 0 | 0 | 1 | F | F | 0 | 0 | F | 0 | \r | \n |

Copier ci-dessous le code de votre fonction et de votre programme principal de test et faire valider

void toModBusAscii(byte\* trame, char\* trameASCII, int n)

{

int result=0;

for (int i = 0; i < n; i++)

{

result += trame[i];

}

result += ~result;

cout << sprintf(trameASCII, "%.2X", result);

}

int main()

{

SetConsoleOutputCP(1252);

byte trame[] = { 0x0B,0x05,0x00,0x01,0xFF,0x00 };

char trameAscii[100 + 1];

byte LRC=calculLRC(trame, 6);

cout <<"La valeur du LRC est de : " << hex << (int)LRC << endl;

cout << "La conversion : ";

toModBusAscii(trame, trameAscii, 6);

//\_getch();

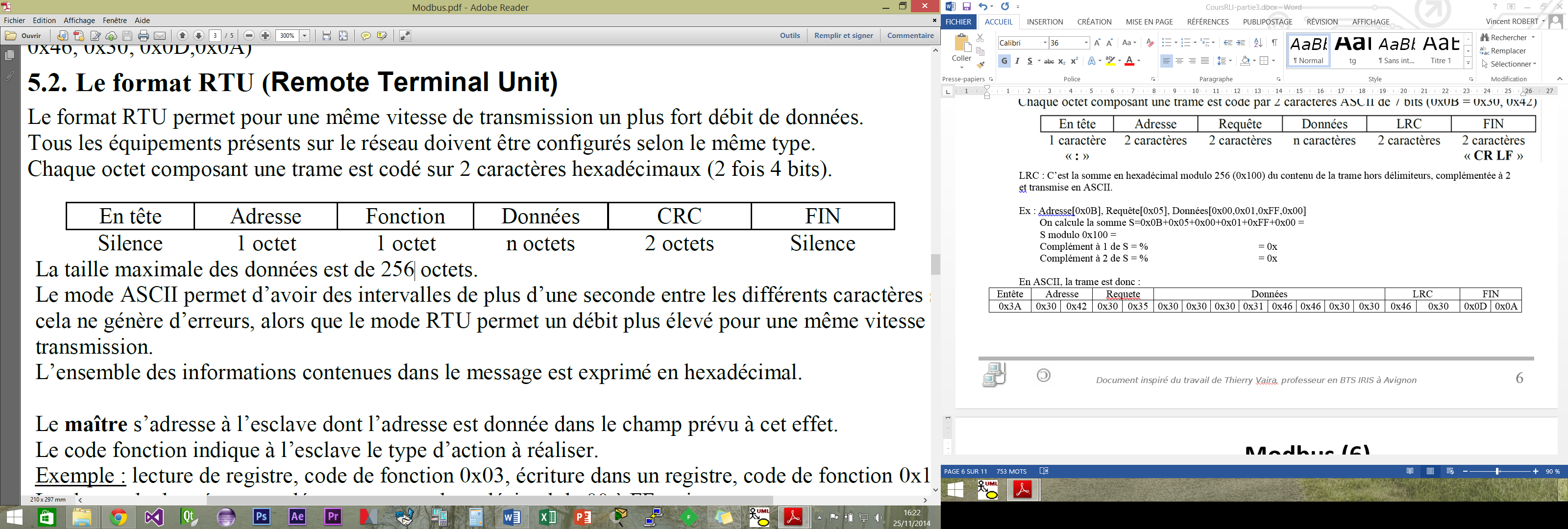
return 0;

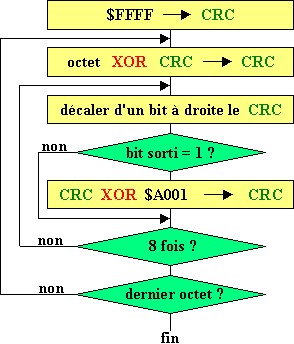
}

Validation enseignant 🞏

## Cas du mode RTU

Dans le mode RTU, la trame est construite de la manière suivante :





Créer une fonction C++

… calculCRC(… tab, … n)

Qui renvoie le CRC du tableau de n octets passé en argument. Le constructeur donne l'algorigramme ci-contre pour le calcul du CRC.

**Exemples à tester (en hexa)**

02 06 00 00 FF E9 🡺 le CRC vaut 0x8709

Avec cet exemple, la trame MODBUS est :

02 06 00 00 FF E9 09 87

Copier ci-dessous le code de votre fonction et de votre programme principal de test et faire valider

Validation enseignant 🞏

En lisant l'article sur le lien suivant <http://fr.wikipedia.org/wiki/Endianness> , indiquez le type d'organisation du CRC dans une trame MODBUS (justifiez votre choix)

# Simulation et analyse de trames Modbus

Nous allons utiliser avec les logiciels **ModbusPoll**( simulation du maître) et **Modbus Slave** (simulation des esclaves)

Il s'agit de simuler la supervision de 3 chambres froides dans une usine frigorifique.

Le PC de supervision fournit des températures de consigne et récupère l'état des différents capteurs de température. Les températures de consignes ou mesurées sont codées sur 16 bits en complément à 2 et exprimées en dixièmes de degrés Celsius.

**Faisons un exemple :**

Comment sera codé -18.6 °C ?

+186 [0000 0000 1011 1010]

1111 1111 0100 0101

+ 1

1111 1111 0100 0110 = 0xFF46

1. Trouver d'abord le nombre de dixièmes de degrés : -186
2. Coder en binaire le nombre positif opposé du précédent : % 0000 0000 1011 1010
3. Inverser tous les bits pour faire le complément à 1 ce qui donne : %1111 1111 0100 0101
4. Ajouter 1 pour faire le complément à 2 : %1111 1111 0100 0110
5. Ce qui donne au final en hexadécimal : 0xFF46

*C'était juste pour que vous compreniez ; vous n'aurez pas à faire ça pour la simulation car les logiciels savent coder les nombres négatifs ☺*

Nous allons utiliser les fonctions suivantes du protocole **MODBUS RTU over TCP/IP** :

Fonction 4 - lecture de n mots d'entrée de l'automate

Fonction 16 - écriture de n mots de sortie

Le procédé de communication implique cinq types de message :

|  |  |
| --- | --- |
| Du Master au slave | Du Slave au master |
| Fonction 4 : demande de lecture de n mots d'entrée de l'esclave | Fonction 4 : réponse contenant les n mots lus |
| Fonction 16 : demande d’écriture de n mots dans l'esclave | Fonction 16 : confirmation de l’écriture de n mots |
| Fonction 3 : demande de lecture de n mots de sorties de l’esclave | Fonction 3 : réponse contenant les mots lus |
|  | Réponse d’exception (en réponse aux deux fonctions, en cas d’anomalie) |

**La structure précise des trames est fournie dans le guide de référence du protocole Modbus joint au sujet de ce TP.**

Construire la simulation correspondant à la configuration suivante :

* Les 3 températures de consigne sont à envoyer à l'esclave 3 à partir de l'adresse 4
* Les 3 températures mesurées dans les chambres froides sont lues sur l'esclave 2 à partir de l'adresse 3
* La température extérieure (qui va permettre d'ajuster les températures de consigne) est lue sur l'esclave 2 à l'adresse 1.

Simuler maintenant les consignes et températures mesurées suivantes :

|  |  |
| --- | --- |
| Consigne | Mesure |
| -18.2 | -17.9 |
| -21.2 | -21.4 |
| -8.5 | -8.2 |
|  | 23.5 (température extérieure) |

Faites ici une copie d'écran des configurations du maitre et des esclaves. Vérifiez bien que tout apparaît à l'écran

Validation enseignant 🞏

Capturer les trames d'échange **côté maître** et copier les ci-dessous (Eviter de copier deux fois la même trame)

COPIE D'ECRAN DES TRAMES D'EMISSION ET RECEPTION

Une image contenant texte

Description générée automatiquement

Validation enseignant 🞏

Maintenant, en utilisant la documentation sur les fonctions Modbus expliquez le plus précisément possible les différentes parties des trames d'émission et de réception en reprenant chaque trame séparément et en remplissant pour chacune d'elle un tableau comme celui-ci-dessous.

Trame 1 : TX : Demande de lecture des 3 températures

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 02 | 04 | 00 | 03 | 00 | 03 | 40 | 38 |
| Adresse esclave | Numéro de fonction MODBUS | Adresse de début | | Nombre de registre à lire = 3 | | CRC =0x3840 | |

RX :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 02 | 04 | 06 | ff | 4d | ff | 2a | ff | ae | 9d | f2 |
| Esclave | Code fonction | 6 octets lus | Temp1 | | Temp2 | | Temp3 | | CRC=0xf29f | |