

Laboratoire de transmission de données :
Fonctionnement du projet

Dessandé Alexandre, Detournay Jérôme, Pecriaux Thomas, Wery Michael

4 janvier 2014

Table des matières

1	Introduction	2
1.1	Participants	2
1.2	Description du projet	2
2	Explication des logiciel	3
2.1	Description du logiciel C#	3
2.1.1	Introduction	3
2.1.2	Connexion	3
2.1.3	Structure du logiciel	3
2.1.4	Différentes fonctionnalités	4
	Données Ethernet :	4
	Température :	4
	Luminosité :	4
	Client :	4
	Création de carte :	4
	Ping :	4
	Erreur socket :	4
2.2	Description du logiciel C	4
2.2.1	Introduction	4
2.2.2	Ethernet	4
2.2.3	Usart-RFID	4
2.2.4	tios.c	5

Chapitre 1

Introduction

1.1 Participants

- Alexandre Dessandé
- Jérôme Detournay
- Thomas Pecriaux
- Michael Wery

1.2 Description du projet

Pour notre projet nous avons décidé de faire une application de monitoring pour une salle. Le projet se décompose en deux grandes parties :

- Une application en **C#** : Le logiciel reçoit des données du système embarqué et les affiche en temps réel. Nous pourrions trouver :
 - ◊ L'ip de la carte
 - ◊ La température courante, maximum, moyenne.
 - ◊ La luminosité courante, maximum.
 - ◊ La personne connecté.Le logiciel pourra aussi faire des actions sur la carte :
 - ◊ Ajouter des données sur une carte RFID
 - ◊ Allumer ou éteindre une led
- Une application en **C** pour système embarqué : Le carte communique avec l'ordinateur via deux méthodes :
 - ◊ Ethernet
 - ◊ UsartElle possède aussi une communication avec un RFID pour l'authentification sur le logiciel en **C#**.

Chapitre 2

Explication des logiciel

2.1 Description du logiciel C#

2.1.1 Introduction

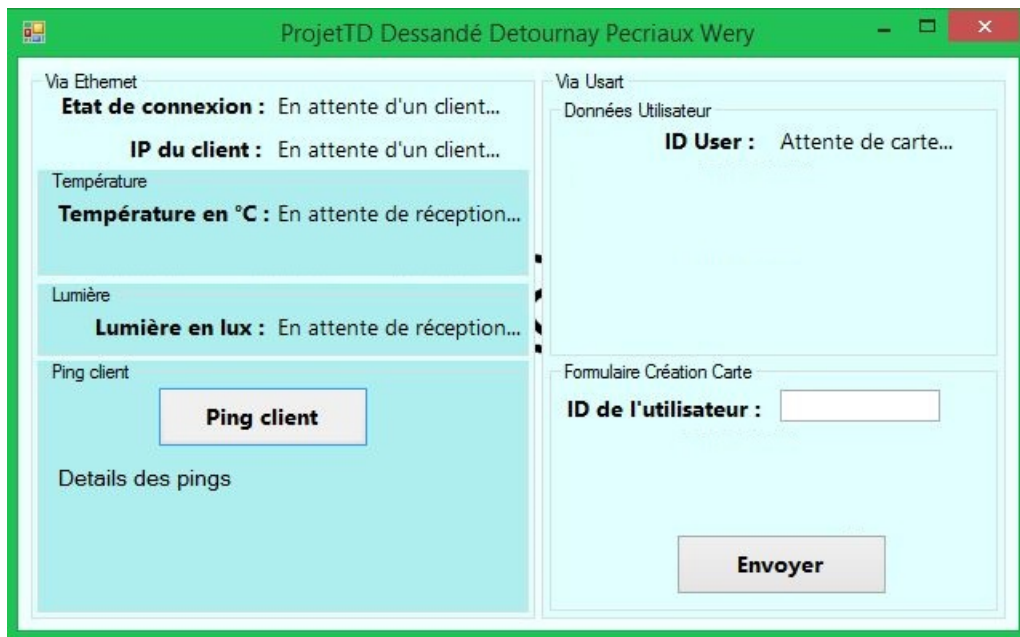
Nous allons ici détailler notre application C# pour y parvenir nous décriront en différents points notre logiciel :

- Connexion : Nous détailleront la connexion au logiciel
- Structure du logiciel : Nous parleront de l'agencement graphique
- Différentes fonctionnalités : Nous expliqueront les différentes parties du logiciel

2.1.2 Connexion

Lors de l'ouverture du logiciel une fenêtre d'authentification s'ouvre, celle-ci vous demande de passer votre carte RFID pour vous logger au programme. Ceci fait vous arrivez sur la page principale du logiciel.

2.1.3 Structure du logiciel



2.1.4 Différentes fonctionnalités

Données Ethernet : S'il n'y a pas de client connecté il nous affiche "En attente d'un client" dans les deux parties sinon :

- pour l'état de connexion il affiche "Client connecté"
- pour l'IP il affichera l'IP du système embarqué

Température : Nous pouvons trouver dans cette partie la température actuelle si aucune donnée n'a encore été récupérée il affiche : "En attente de données".

Luminosité : Nous pouvons trouver dans cette partie la luminosité dans la pièce si aucune donnée n'a encore été récupérée il affiche : "En attente de données".

Client : Cette partie nous donne les informations concernant la personne qui s'est connecté au logiciel via une carte RFID.

Création de carte : A partir d'ici nous pouvons remplir un petit formulaire pour écrire sur une carte RFID.

Ping : La section ping dispose d'un bouton où lorsque l'on clique dessus nous pourrions voir s'afficher les différentes données du ping :

- Si réussit ou non
- L'ip du client
- le TTL
- La fragmentation des données sur plusieurs
- La taille du buffer

Si la carte est éteinte le bouton renverra un "Time Out"

Erreur socket : Lorsque l'on le câble Ethernet est retiré le programme nous sort une erreur.

2.2 Description du logiciel C

2.2.1 Introduction

Nous parlons ici notre application C mis en place sur notre système embarqué.

2.2.2 Ethernet

Nous utilisons ici la carte comme un client TCP pour envoyer les données de température et luminosité en permanence au pc.

2.2.3 Usart-RFID

L'usart nous permettra d'envoyer et récupérer des données sur le logiciel C#. Nous avons décidé de coupler cette fonctionnalité avec le RFID. Nous pouvons ainsi récupérer les données d'une carte RFID pour authentifier le client sur le programme. Nous pouvons aussi

2.2.4 tios.c

Dans le fichier TIOS s'y trouve toute la partie gestion des interruptions et appel des fonctions en conséquence.

Certaines fonction sont appelé periodiquement grace au tableau MaCB qui est parcourut et appel les fonctions quand leurs temsp est arrivée.

D'autres fonctions ne necessitent pas d'être appelée periodiquement, en conséquence, nous avons besoin d'autres variables pour enregistrer les pointeurs de fonction, cerataines de ces fonctions ont des arguments ou des retours donc nous devons adapter les variables pointant vers le fonction pour gérer ces arguments et retours lors de leurs appel.

Il n'est pas possible d'utiliser un tableau de pointeurs pour ces variables car leurs retour ou arguments varient d'un a l'autre et donc automatiser les appel n'est pas possible.

Les interruptions initialisent des drapeaux pour savoir qu'elle appel non periodique doivent être appelé. Elles sont donc appeller en fonction des drapeaux dans la boucle principale : `while(1)`

Toutes ses fonctions appelé se trouvent dans le fichier "Main.c" et sont donc accessible a la modification par l'utilisateur voulant créer un programme utilisant notre TIOS.c