

Le protocole TODO

1. Introduction

C'est un protocole simple qui permet l'envoi de paquets entre des clients et des serveurs via des trames I²C. Il permet à l'expéditeur de s'authentifier, et de spécifier des options indiquant par exemple la présence ou non de cryptage.

1. Pourquoi créer ce protocole ?

La technologie I²C ne permet pas à un destinataire de savoir quel composant I²C lui parle. Or une relation client-serveur impose la présence de réponse à certaines requêtes. Il a donc fallu créer une structure de communication : le paquet TODO.

2. Contexte

- Le protocole TODO forme une couche réseau se situant au-dessus de la couche liaison I²C.
- Ce protocole fonctionne en mode non connecté. En effet les communications I²C ne sont pas en mode connecté. Il n'est donc pas indispensable que le protocole TODO fonctionne en mode connecté. De plus, les cartes STM8S-Discovery ne sont pas assez puissantes pour supporter un protocole connecté.
- C'est un protocole non fiable.

2. Spécification du protocole

1. Description fonctionnelle

Le protocole TODO s'appuie sur la technologie I²C. Ainsi, l'adresse TODO permettant d'identifier une entité d'une autre est la même que l'adresse I²C. Cette adresse est donc sur 7 bits et doit être unique. De plus, elle doit respecter les classes d'adresses spéciales ou réservées de la technologie I²C telle que l'adresse de broadcast.

Lorsque le bit CR est positionné, cela indique que les données du paquet TODO sont cryptées. Le cryptage est réalisé par une opération logique XOR entre un octet de données et la clef de cryptage codée sur un octet, et ce pour chaque octet de données.

2. Entête du paquet TODO

Les données échangées posséderont l'en-tête suivant.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	ADDR_SRC							Not Used
2	LENGTH							
3	TY		CR	Not Used				
4	Not Used							
n	DATA							

- ADDR_SRC : adresse de l'expéditeur du paquet
- LENGTH : nombre d'octets de données
- TY : définit le type de l'entité
 - 00b, client
 - 01b, serveur
 - 10b, autre
 - 11b, autre
- CR : utilisation du cryptage
 - 0, cryptage des données désactivé
 - 1, cryptage des données activé
- DATA : données transportées par le paquet

Les bits non utilisés sont réservés pour les futures évolutions du protocole.

3. Cas particulier

Dans ce protocole, une communication entre un maître récepteur et un esclave émetteur n'a pas lieu d'être. Le 8^{ème} bit R/W d'une trame d'adresse I²C, servant à indiquer la direction de la transmission sera donc tout le temps configurée en mode écriture (maître émetteur vers esclave récepteur).

Cependant il reste possible pour un maître n'utilisant pas le protocole TODO d'être récepteur et de se connecter à un esclave utilisant le protocole. L'esclave ainsi émetteur n'a pas le contrôle de la transmission. Il faudra donc qu'il réponde au maître récepteur jusqu'à ce que ce dernier termine la transmission. Le protocole TODO impose donc à l'esclave émetteur de répondre au maître récepteur avec un ou plusieurs paquets TODO si besoin. Cependant, il n'est pas regardant vis-à-vis du champ DATA contenus dans ces paquets de réponse.

3. Interface utilisateur

Une interface utilisateur implémentant le protocole TODO devrait posséder les méthodes suivantes.

- Une fonction d'initialisation permettant de configurer l'adresse de l'entité, son type et si on utilise le cryptage ou non.
- Une méthode d'envoi de données, prenant en paramètre les données non cryptées à envoyer, et le nombre d'octets de données.
- Une méthode de réception des données, prenant en paramètre un buffer de réception et le nombre d'octets de données attendus.
- Une méthode de fermeture, permettant de libérer la mémoire allouée lors de l'initialisation.