

Le périphérique TODO

1. Introduction

Nous allons définir dans ce document le cahier des charges d'une application pouvant être utilisée comme sujet de mini-projet d'EE250 – Electronique numérique.

Cette application est utilisée dans un réseau I²C comportant des terminaux, des périphériques et des serveurs et implémentera le protocole TODO.

Taxonomie

Périphérique - Composant électronique comportant un module I²C, ajouté grâce à un STM8 ou non.

Exemples :

- Moteur contrôlé en PWM
- Matrice de LED
- Horloge temps réel ou RTC

Terminal - Périphérique d'entrée/sortie permettant une interaction entre le réseau I²C et un opérateur. Exemples :

- Clavier numérique contrôlé par un STM8
- Hyperterminal communiquant avec un STM8 via une liaison RS232

Opérateur - Personne physique agissant sur un terminal afin de provoquer une action sur un périphérique I²C.

2. Objectif et environnement de l'application

Contexte

Pour ce mini-projet, un réseau I²C a été créé. Celui-ci comporte au moins un serveur et un ou plusieurs périphériques ou terminaux. Le serveur utilise le protocole TODO pour communiquer avec les terminaux. En effet, le serveur doit connaître l'expéditeur d'une requête. Or la technologie I²C ne permet pas de savoir qui s'exprime sur le bus. L'utilisation du protocole TODO est donc justifiée.

L'adresse du serveur est considérée connue et sera indiquée lors de la remise de ce cahier des charges.

Le serveur embarqué sur un STM8 contient une CLI (Command Line Interface). Ainsi, ce STM8 est capable de recevoir une chaîne de caractères via le protocole TODO, de l'interpréter, puis d'agir en conséquence. Il peut par exemple lire l'heure sur une horloge temps réel telle que le DS1307 sur réception de la chaîne « lireHeure » et la communiquer au terminal.

Objectif

Le but de ce mini-projet est d'intégrer un périphérique à ce réseau. Le périphérique comportera 3 servomoteurs pilotés en PWM (Pulse Width Modulation) et une matrice de LED 6*8 qui permettra à l'opérateur de dessiner.

L'opérateur voulant se servir du périphérique enverra des commandes au serveur via un terminal. C'est le serveur qui interfacera le périphérique via des codes opérations qu'il faudra définir.

3. Cahier des charges

Implémentation du protocole TODO

Afin de communiquer correctement avec le serveur, nous implémenterons le protocole TODO ainsi que les méthodes suggérées par celui-ci. La spécification du protocole TODO sera jointe à ce document.

Configuration du module I²C du STM8

Il faudra veiller à ce que le périphérique utilise la même configuration I²C que le serveur. Pour cela, se référer à la documentation du serveur I²C. Les éléments auxquels il faut être attentif sont

- La vitesse de communication
- La position de l'acquittement
- Le rapport cyclique de l'horloge

L'adressage est sur 7 bits, cela est imposé par l'utilisation du protocole TODO.

Le bus I²C

Pour valider son fonctionnement, vous devrez relier le périphérique I²C au reste du réseau. Il faut pour cela le connecter au bus I²C. Le bus est composé des deux lignes SDA (Serial Data Line) et SCL (Serial Clock Line). Ces deux lignes sont tirées à la tension $V_{DD} = 5V$ via des résistances de pull-up.

Sachant que le réseau est dans la situation décrite en figure 24 du document « I2c_HowToUseIt1995.pdf », vous devrez calculer les résistances série permettant de protéger le STM8 des surtensions passagères pouvant survenir sur le bus.

De plus, il faudra vérifier que les résistances de pull-up utilisées sur le bus I²C respectent les spécifications définies dans ce même document.

Ne pas oublier que les masses des différents équipements connectés sur le bus doivent être identiques.

Liste des fonctionnalités à implémenter

Fonctionnalités basiques

1. Vous devrez imaginer un ensemble de codes opération pour piloter en position chacun des 3 servomoteurs.
2. Vous devrez configurer un timer en utilisation PWM.
3. La communication avec le serveur se fera via le protocole TODO.
4. Vous devrez imaginer un ensemble de codes opération pour permettre le déplacement d'un curseur sur la matrice de LED, ainsi que pour allumer ou éteindre la LED sélectionnée.

Fonctionnalités confortables

1. Vous devez implémenter une fonction permettant d'envoyer à l'opérateur les angles des servomoteurs.
2. Le curseur de la matrice de LED devra clignoter.

Fonctionnalités avancées

1. Vous devez utiliser l'EEPROM pour sauvegarder les dessins de l'utilisateur.