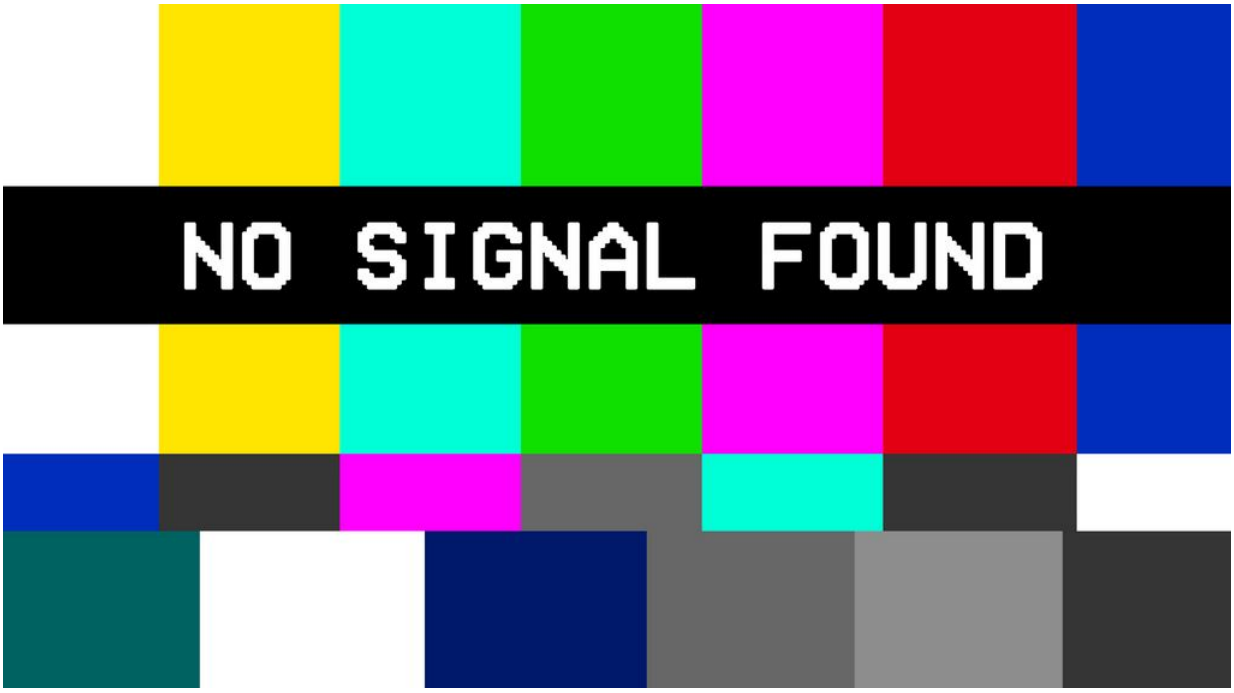
Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Systèm embarqués

Auteurs :
Marc Roten
Sven Rouvinez

Professeur :
Daniel Gachet



26 novembre 2018

Table des matières

1 Heure de travail

12 heures

2 Introduction

Le but de TP est de pouvoir comprendre le fonctionnement d'une ligne série UART, d'un bus série I^2C et d'un bus série SPI. Ensuite de développer une ligne de commande sur l'interface série UART, de mettre en oeuvre le thermomètre sur le bus I^2C ainsi qu'un écran LCD sur le bus SPI afin de pouvoir implémenter tout ça comprendre comment étudier une datasheet d'un composant électronique simple

3 Synthèse

Sven

- Comment dessiner sur l'écran LCD
- Quand utiliser les pointeurs de fonctions

Marc

- Ce travail a été compliqué. le shell n'était pas trivial. Il y avait beaucoup de différentes méthodes, une architecture complexe et tout mettre ensemble n'est pas facile.
- Acquis mais à exercer : Les méthodes pour parser des inputs au clavier. Il faudra aussi que je m'exerce sur les pointeurs qui était dans le cas du shell, pas facile à comprendre et à implémenter.
- Ce TP m'a toutefois permis de mieux comprendre le fonctionnement du beaglebone, malgré les problèmes rencontrés.
- compréhension de l'utilisation des différents périphériques tels que l'affichage LCD

4 Quelles sont les caractéristiques principales (signaux et protocole) des interfaces de communication UART, I2C et SPI ?

UART $Universal Asynchronous Receiver/Transmitter$ **:** son but est de transmettre et recevoir les données depuis la ligne sérial. Son avantage est d'utiliser uniquement 2 fils pour

transmettre les données

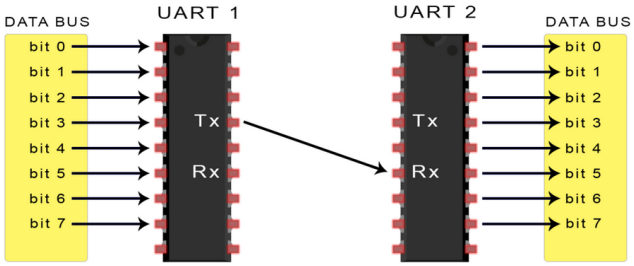


FIGURE 1 – UART

Les données sont transférées de manière asynchrone, il n'y a donc pas d'horloge, au moment de démarrer une transmission, la fréquence d'envoi est sélectionnée et exprimée en *baud rate [bits par second]* c'est l'UART qui transmet va envoyer un seul bit pour démarrer la transmission et il va ensuite envoyer les 5 ou 9 bits de message avec un possiblité d'un bit de parité et va ensuite mettre la ligne de transmission sur haut pendant au moins de bit pour indiquer la fin de la transaction

Avantages

- Utilise 2 fils
- Pas d'horloge
- bit de parité pour les erreurs
- Méthode efficace et prouvée

Désavantages

- Taille maximum des données est de 9 bits
- Différence de baud rates entre les 2 UART doit être de 10%

SPI $Serial Peripheral Interface$ **:** c'est un protocole de communication utilisé par exemple avec les lecteurs RFID et ils utilisent le SPI pour communiquer avec les microcontrollers. La transmission des données se fait avec un flux sans interruption donc les bits sont envoyés et reçus dans un flux continuz

Afin de pouvoir transmettre des données, il y a un master (microcontrolleur) et un slave (sensors) et il va recevoir les instructions du master, par exemple :

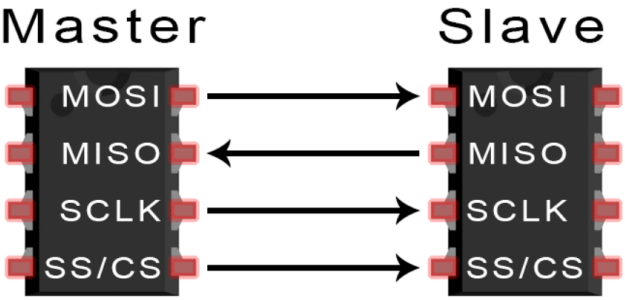


FIGURE 2 – SPI

- MOSI : ligne sur laquelle le maître envoie les données à l'esclave
- MISO : ligne sur laquelle l'esclave envoie les données au maitre
- SCLK : ligne pour l'horloge
- SS/ CS : utilisé si plusieurs slave, et permet de choisir vers qui envoyer les données

L'horloge (SCLK) permet de synchroniser l'envoi des bits au slave et c'est toujours le master qui démarre une transmission parce que c'est lui qui configure le signal d'horloge.
Ensuite il est possible de sélectionner plusieurs slave il faut donc décider avec qui il va discuter grâce à la pin SS, dans le cas d'un seul c'est slave la ligne va faire passe la ligne de bas à haut et va y rester. Dans le cas où plusieurs slave sont disponibles, il y 2 façons de les connecter : soit le master possède n SS pin qui seront connectées à chaque slave ou en utilisant un daisy-chained (les éléments sont chaînés les un aux autres) Pour envoyer les informations, c'est la

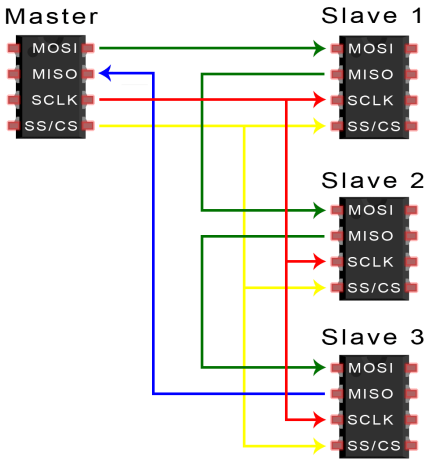


FIGURE 3 – Exemple de daisy-chained

pin MOSI qui est utilisé à travers une ligne sérial et les bits sont envoyés les uns après les autres en commençant par le MSB et le slave peut lui aussi envoyer des informations, en commançant par le LSB en utilisant la pin MISO

Avantages

- Flux continu sans interruption
- Adressage des slaves simple
- Taux de transfet rapide, dépend de la clk
- Informations peuvent être reçues et envoyées en même temps

Désavantages

- Utilise 4 fils
- Pas de contrôle si la donnée est correctement reçue
- Pas de bit de parité
- Uniquement un master

I2C *Inter – IntegratedCircuit* Utilisé dans les OLED display (comme dans ce TP), gyroscope, etc et la différence entre le SPI c'est qu'il peut y avoir plusieurs master et il utilise uniquement 2 fils contrairement aux 4 d'avant Les pins qui sont utilisées sont :

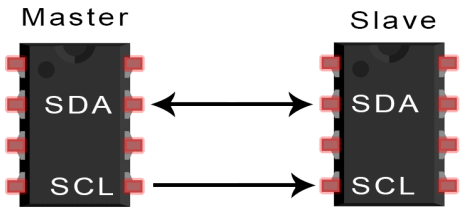


FIGURE 4 – I^2C

- SDA *Serial Data* : ligne pour envoyer et recevoir les données entre le slave et le master
- SCL *Serial Clock* : ligne qui transporte le signal de clock

Comme dans le SPI, la communication est synchrone grâce à l'horloge qui est contrôlée par le master.
Les données sont envoyées par frame, chaque message contient :

- Start Condition : la ligne SDA passe de haut à bas avant que la SCL passe de haut à bas

- Stop Condition : la ligne SDA passe de bas à haut après que la SCL passe de bas à haut
- Address Frame : Identifie avec 7 à 10 bits, le slave auquel le master veut parler
- Read/ Write bit : permet au master en d'indiquer au slave qui lui envoie des données en mettant à 0 ou il attend d'en recevoir en passant à 1
- ACK/ NACK bit : après chaque message, il est possible de savoir si la frame a bien été reçue
- data frame : contient le message codé sur 8 bits

Dans le cas où il y a un master pour n slave il y a un minimum de 128 adresses et un maximum de 1024 adresses donc slave
Par contre avec ce protocole il possible d'avoir n master pour n slave, il se peut que 2 masters reçoive ou envoie des données sur la ligne SDA qui est partagée entre tous les composants (master+slave) du système donc pour palier à cela il faut que le master qui veut émettre des données doit d'abord détecter si la ligne SDA est haute ou basse avant de transmettre. Si la ligne est à l'état bas, un autre master à le contrôle et donc si la ligne est à l'état haut, le message peut être transmis

Avantages

- Uniquement 2 fils
- Plusieurs master pour plusieurs slave
- Bit de confirmation si bien frame bien reçue
- Hardware moins compliqué que UART
- Protocole très utilisé

Désavantages

- Taux de transfert plus bas que le SPI, Ultra fast mode = 5Mbps
- Taille d'une frame limitée à 8 bits
- Plus compliqué à intégrer que SPI

5 Comment effectuer une rotation d'image avant d'être affichée sur un écran LCD ?

Il est possible d'utiliser une matrice de rotation avec de retourner l'image avant qu'elle soit affichée Dans notre cas, il faut faire une rotation de $90° = \frac{\pi}{2}$
La formule de base est :

$$x' = x\cos(\frac{\pi}{2}) - y\sin(\frac{\pi}{2})$$

$$y' = x\sin(\frac{\pi}{2}) + y\cos(\frac{\pi}{2})$$

Et en sachant que $cos(-\frac{\pi}{2}) = 0$ et $sin(-\frac{\pi}{2}) = 1$, la solution pour effectuer une rotation est

$$x = -y$$

$$y = x$$

6 Conclusion

Nous n'avons pas pu terminer ce travail et cela vient du fait que nous n'avons pas bien compris comment utliser l'afficheur LCD fournis, nous allons donc dès que les solutions seront mises à disposition, travailler sur notre code afin de trouver ce qui nous empêchait d'avancer
Nous avons récupérer du code chez certains autres groupes afin de pouvoir mieux comprendre et avancer quelque peu.