Projet de recherche et développement

Protocole de communication entre FPGA et microcontrôleur sur 1 fil

Jérémy Cheynet et Yann Sionneau

Télécom SudParis

26 juin 2010

Sommaire

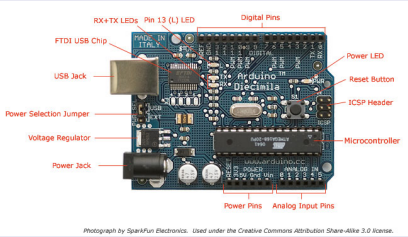
Objectifs

- Protocole de communication entre microcontrôleur et FPGA
- Protocole sur 1 fil
- Protocole asynchrone
- Resynchronisation sur chaque bit

Le matériel

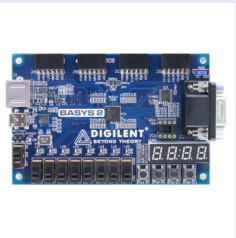
Microcontrôleur

ATmega328 sur carte arduino



FPGA

Spartan3E sur carte Basys2

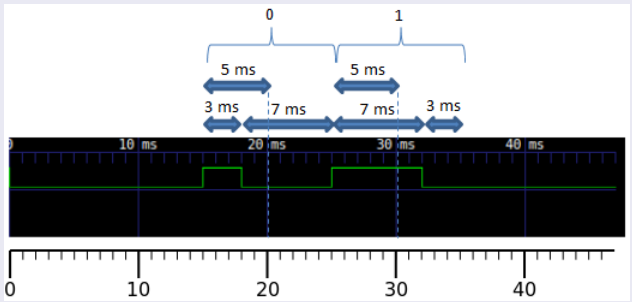


La couche physique
Les bits

Différents bits

- Le bit de start
- Le bit "haut" logique
- Le bit "bas" logique
- Le bit de stop

Définition des états logiques

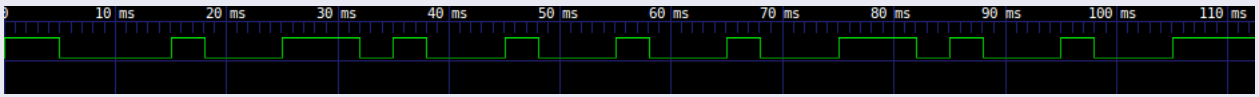


La couche physique
La trame

Trame de 10 bits

- Le bit de start
- 8 bits de data
- le bit de parité
- le bit de stop

schéma d'un paquet



La couche de transport

Au moins 4 octets

- L'adresse source du paquet (un octet)
- L'adresse de destination du paquet (un octet)
- La taille et le checksum (un octet)
  - La taille sur 4 bit
  - le checksum sur 4 bit
- Les datas (entre 1 et 16 octets)

La couche applicative

Fonction de haut niveau

- Fonction d'envoie d'un paquet prenant en paramètre :
  - L'adresse de destination
  - La taille des datas
  - Les datas
- Fonction de réception d'un paquet (aller interroger le buffer)
  - Adresse source du paquet
  - Taille des datas
  - Datas
- Fonction de traitement des erreurs
  - Problème de parité
  - Erreur du checksum
  - Buffer plein

Comment débuguer

La liaison série

- Fonction d'envoie d'un caractère
- Fonction d'envoie d'une chaine de caractères

Les test unitaires

- Tests unitaires sur chaque couche
- Codage couche par couche pour arriver à la couche applicative

Le compilateur

Pas de print dans la boucle

```
while(1){
if( reception == 1 ){
reception = 0;
print("octet recu");}}
```

Print dans la boucle

```
while(1){
print("coucou");
if( reception == 1 ){
reception = 0;
print("octet recu");}}
```

La désynchronisation

Désynchronisation des bits lors de la réception d'un octet

Solution : modifier le bit de start pour qu'il soit unique

Désynchronisation des octets lors de la réception d'un paquet

Solution : créer un octet de start de paquet (modification de l'octet d'adresse pour qu'il commence toujours par la même chose)

Ne pas bloquer le programme principal

- Interruption sur front montant qui déclenche le timer
- Interruption toute les millisecondes sur le timer (s'il est déclenché)
- Analyse, 5 millisecondes après l'interruption sur front montant, du signal
- Une fois un octet reçu entièrement, passage de l'octet pour analyse de celui-ci par la couche supérieur
- Création d'un paquet stocké dans un buffer

Pas de conflits d'envoie/reception sur la ligne

- Utilisation d'une variable globale de type "Mutex"
- On vérifie si la ligne n'est pas occupée avant d'emettre

Conclusion

Merci pour votre écoute, des questions ?

Références

[1] http ://www.avrfreaks.net
[2] http ://www.nongnu.org/avr-libc/
[3] datasheet