
Gaëtan Harter
Adrien Oliva



ÉCOLE NATIONALE SUPÉRIEUR D'INFORMATIQUE
ET DE MATHÉMATIQUES APPLIQUÉES
681, rue de la Passerelle, 38400 Saint Martin
d'Heres

Projet SLE
Écriture de driver RocketIO
Cahier des Charges



gaetan.harter@ensimag.imag.fr
adrien.oliva@ensimag.imag.fr

Table des matières

1	Contrôleur RocketIO	3
2	Driver	3
3	Planning	3

1 Contrôleur RocketIO

Le rocket IO est un contrôleur réseau couplé à un DMA. Il permet d'envoyer des paquets, d'en réceptionner comme un contrôleur classique mais aussi de recevoir et de stocker des paquets directement en mémoire, sans avoir à passer par le système d'exploitation.

2 Driver

Le but du projet est de réaliser la partie logiciel liant le contrôleur au système d'exploitation, un noyau linux 2.6.32, sur une carte Xilinx Virtex 4.

Ce driver prendra la forme d'un module réseau pour un noyau linux. Il devra supporter l'envoi de paquets TCP/IP et la réception de paquets DDP/IP. Ce dernier est un protocole qui permet la réception de paquets directement en mémoire.

La réception des paquets se fait dans un espace mémoire réservé au driver, sous la forme d'un tampon circulaire.

Pour une utilisation plus générale du driver, on implémentera également la réception TCP/IP et l'émission DDP/IP.

La communication entre le module, le matériel et le système d'exploitation s'effectue via des interruptions.

3 Planning

21 sept. 2010	Choix du sujet
28 sept. 2010	Première approche du LDD
du 4 au 19 oct. 2010	Compilation de l'environnement de cross-compilation Démonstration du driver sur machine nue et de l'environnement Xilinx
du 24 oct au 2 nov. 2010	Lecture du Linux Device Driver Lecture des anciens rapports concernant le contrôleur Rocket IO Élaboration du cahier des charges
du 2 au 9 nov. 2010	Allègement et portage du module préexistant pour noyau 2.4 jusqu'à compilation et fonctionnement sur carte (chargement et déchargement, sans réelles fonctionnalités)
du 9 nov. au 14 dec. 2010	Ajout progressif des fonctionnalités du driver avec le banc de test associé

La fin du projet est consacré au débogage et à l'écriture de la documentation même si ces deux étapes seront en partie réalisées au cours du développement.

Ajout de fonctionnalités

Les fonctionnalités suivantes seront implantées les unes à la suite des autres :

- chargement et déchargement du module
- ouverture et fermeture de l'interface réseau
- gestion du tampon circulaire
- réception et envoi via le protocole TCP/IP
- envoi de paquets DDP/IP
- réception DDP/IP