Système pour la vision embarquée

Objectifs

- L'objectif global du projet est la mise en œuvre d'un système de vision faible complexité destiné à une application embarquée.Une deuxième phase plus orientée vers de la robotique porte sur la réalisation d'une boussole numérique.

1.1 Description générale

La première partie du projet consiste à mettre en œuvre un système de vision simple destiné à un usage embarqué.

Le système de visualisation se basera sur un écran VGA sous le contrôle d'une IP matérielle développée sur FPGA. Cette partie sera traitée avec Fabrice Muller au cours des 5 premières semaines du projet (12 septembre au 10 octobre).

La deuxième partie du projet (16 octobre au 28novembre) s'attachera à mettre en œuvre une communication USB entre un microcontrôleur STM32 et le FPGA qui affichera les données reçues (figure 1).

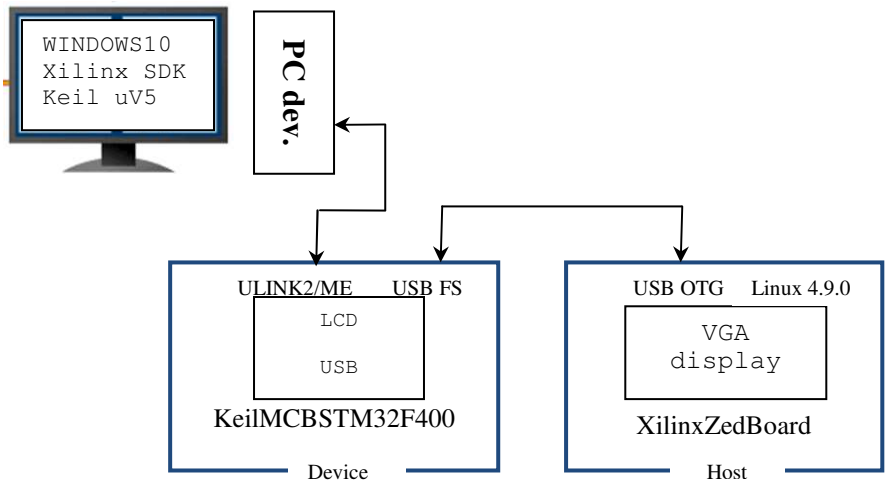

Fig1. Environnement de développement

1.2 Transmission de messages simples

Dans un premier temps, nous allons envoyer des messages simples depuis le microcontrôleur STM32 vers la carte `ZedBoard`qui affichera lesdonnées reçues sur un écran VGA. Les étapes du projet sont les suivantes :

1. Mise en œuvre d'un environnement de développement STM32 sous Linux. L'environnement KeiluVision utilisé habituellement est limité à 32Ko en taille de code exécutable. Nous risquons de dépassercette limite au cours du projet. Il est possible de contourner cette limitation en utilisant un cross compilateur GCC ARM. A partir de l'exemple décrit dans http://www.wolinlabs.com/blog/linux.stm32.discovery.gcc.html, installez un environnement Linux GCC ARM complet sur vos stations de travail qui permette de développer, compiler et exécuter du code pour la carte `MCBSTM32F400`. Vous devrez utiliser pour cela une sonde ST-LINK au lieu des sondes ULINK-ME habituelles. Mettez en œuvre cet environnement pour faire fonctionner un exemple de code sur la

carte MCBSTM32F400. Vous aurez également besoin d'étudier comment porter un projet Keil (avec ses librairies) vers GCC ARM. Appliquez cette approche pour porter le projet *USB CDC (*vu en TP Circuits et Protocoles) sous GCC ARM.

Evaluation partielle 1.2.1 Mise en œuvre d'un environnement de développement STM32 sous Linux.

2. Mise en œuvre de la communication USB CDC entre les deux cartes. La mise en œuvre d'une communication USB CDC du côté ZedBoard s'appuiera sur les pilotes Linux qui ne sont pas activés sur le noyau que vous avez utilisé en TP Linux Embarqué (vous pouvez faire le test en branchant une clé USB sur le port USB OTG). Il faut donc chercher un moyen pour avoir un noyau Linux où les pilotes USB sont activés, avec une communication USB CDC qui soit opérationnelle.

Evaluation partielle 1.2.2 Mise en œuvre de la communication USB CDC entre les deux cartes

3. Mise en œuvre de l'affichage sur le système ZedBoard/Linux. Il faut ici écrire un programme Linux qui récupère les messagesde la communication USB CDCet les affiche à l'aide de l'IP VGA. Cela implique i) de développer un noyau Linux Xilinxqui intègre l'IP VGA et ii) de développer le pilote Linux correspondant. On utilisera ici un pilote caractère étant donné qu'on utilise une communication série relativement lente (115200 bauds).

Evaluation partielle 1.2.3 Mise en œuvre d'un noyau Linux qui intègre l'IP VGA et transmission/affichage d'un message texte simple.

1.3 Matrix embedded

Dans un deuxième temps, nous allons utiliser le système développé pour réaliser l'affichage d'un écran de veille de type *Matrix*, à la fois sur l'écran de la carte MCBSTM32F400 et sur la carte ZedBoard avec l'IP VGA (figure 2) :

4. Le code réalisant l'affichage *Matrix* sera extrait de la librairie *libcaca*. La première étape sera donc d'installer et de tester l'affichage *Matrix*, d'abord sur les stations de travail Ubutnu, puis d'extraire de cette librairie un projet C qui ne réalise que l'affichage *Matrix* (la librairie *libcaca* étant une librairie plus générale de conversion d'image vers une représentation ASCII). On modifiera la fonction d'affichage pour visualiser la matrice dans un terminal avec des *printf*.

   Evaluation partielle 1.2.4 Extraction de l'affichage *Matrix* de la librairie libcaca. Test sur station de travail.

5. Porter le code extrait sur la carte MCBSTM32F400 et réaliser l'affichage sur l'écran LCD. Il faudra procéder ici de manière progressive et méthodique, notamment pour adapter les librairies *gcc* du code initial vers des librairies *Keil* équivalentes.

   Evaluation partielle 1.2.5 Portage de l'affichage *Matrix* sur STM32. Test sur l'écran LCD.

6. Intégrer le code dans le projet principal de communication USB CDC avec le systèmeZedBoard/Linux/IP VGA. Modifier le code pour transmettre l'affichage*Matrix* ligne par ligne vers le port USB connecté àla carteZedBoard.

   Evaluation partielle 1.2.6 Communication USB CDC ligne par ligne vers la carte ZedBoard. On pourra utiliser picocom.

7. Ecrire un code C pour la ZedBoard qui récupèrera les données ligne par ligne et les affichera à l'aide du driver de l'IP VGA. On pourra s'aider ici dela librairie TERMIOS pour faciliter la communication avec le port USB.

Evaluation partielle 1.2.7 Validation finale du système.

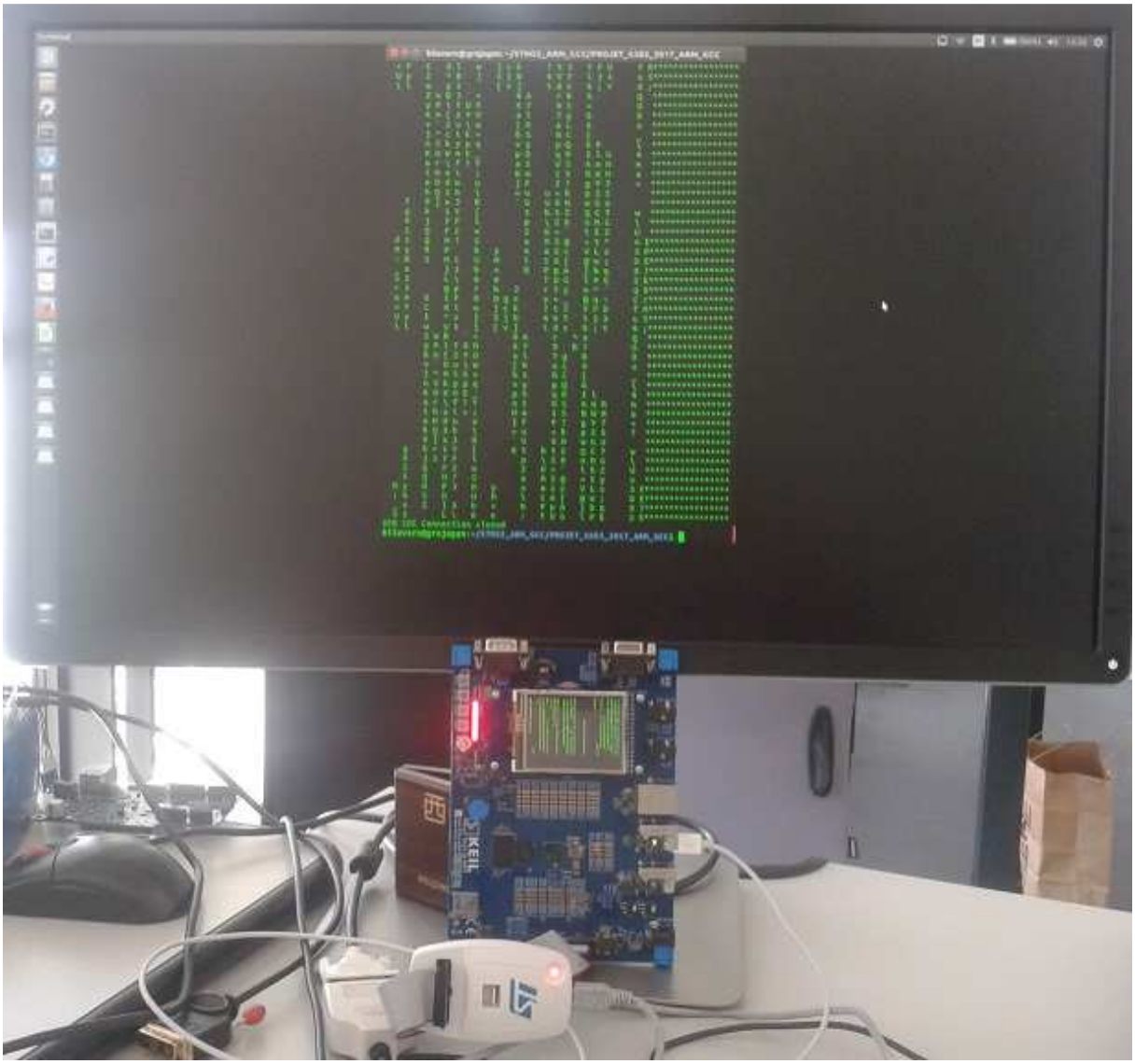

Fig2. Matrix embedded.

Déroulement, assiduité et évaluations :

1. Chaque binôme dispose d'une carte `Keil MCBSTM32F400` et d'une carte`ZedBoard`. Chaque carte est numérotée, chaque binôme est responsable de ses cartes. En cas de panne le binôme sera pénalisé sur sa note.
2. La promotion travaille en binôme, chaque binôme travaille indépendamment. Toute similitude excessive entre 2 projets sera sanctionnée sur la note.
3. Cette partie du projet (Système pour la vision embarquée) donne lieu à un suivi avec des évaluations partielles, une validation finale et un rapport. Le rapport fera 5 pages maximum (hors annexes / code) avec un plan du type: Introduction / Spécification / System overview, High-Level design / Hardware / Software, Test and results, Conclusion (critique objective, améliorations), Références (article, liens, sites web), Annexes (code).
4. L'assiduité aux séances de projet est obligatoire et sera contrôlée. Les absencesaux séances non encadrées seront pénalisées dans l'évaluation des résultats.