INSA TOULOUSE
INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE TOULOUSE
Université de Toulouse

ARM-ADA

Présentation du 20/09/2013

Objectif du projet

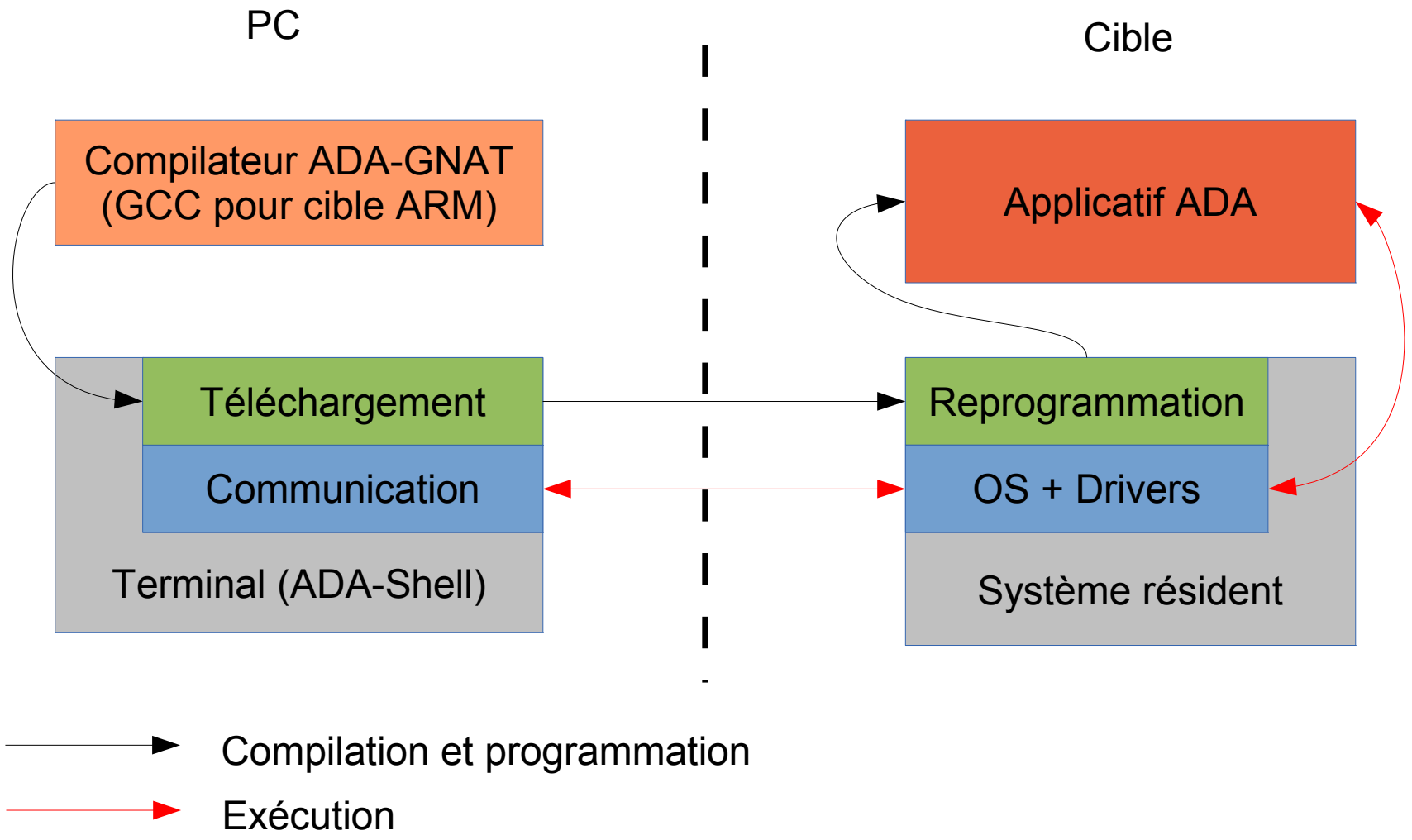
Sensibiliser les étudiants aux systèmes embarqués, dés leur 2eme année.

Les faire travailler sur un système plus contraint qu'un PC, obligeant à plus de rigueur et de réflexion lors de la programmation

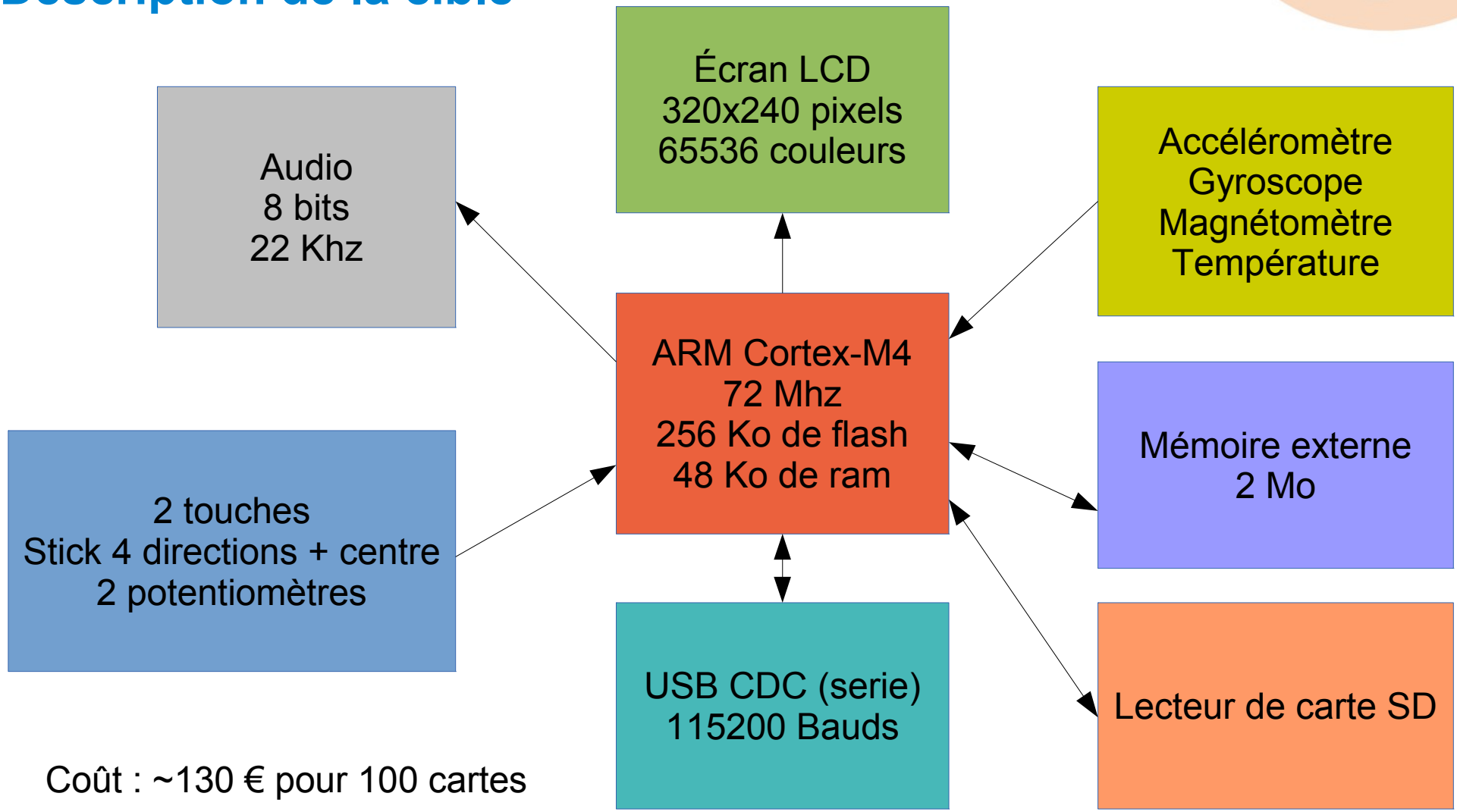
Les faire travailler sur un système ludique, utilisant différents types de capteur, présent sur des appareils courants comme tablettes ou smartphones

INSA de Toulouse • 135, Avenue de Rangueil • 31077 Toulouse
Tél. : +33(0)5.61.55.95.13 • Fax : +33(0)5.61.55.95.00 • www.insa-toulouse.fr

ARM-ADA

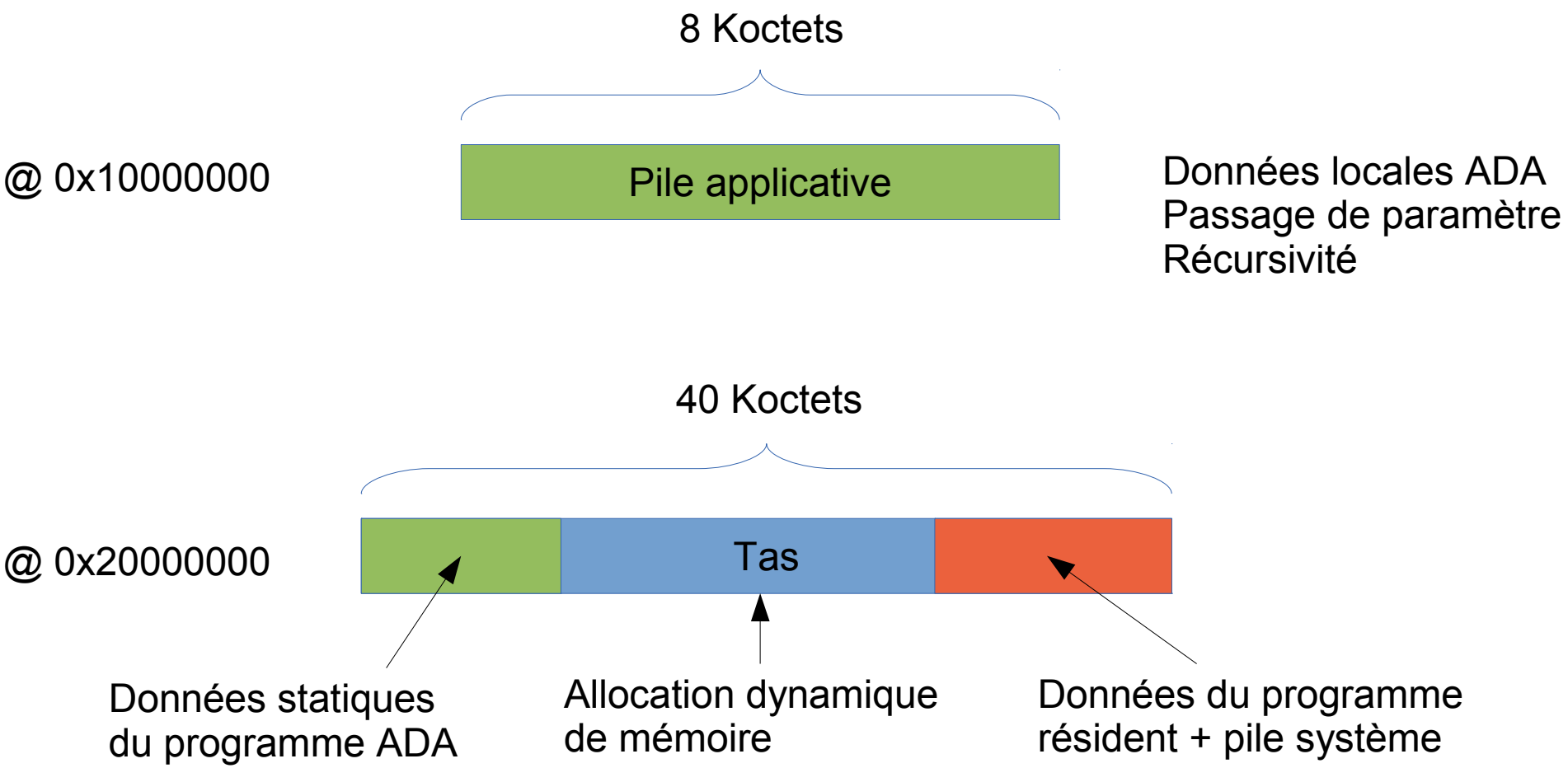

PC
Cible
Compilateur ADA-GNAT
(GCC pour cible ARM)
Applicatif ADA
Téléchargement
Communication
Terminal (ADA-Shell)
Reprogrammation
OS + Drivers
Système résident
Compilation et programmation
Exécution

Description de la cible



Coût : ~130 € pour 100 cartes

Organisation mémoire de la cible



Construction de l'exécutable ADA pour la cible

Outil `gnatmake` existant mais inutilisable pour générer le projet pour la cible → Utilisation de l'outil `make` et d'un fichier `Makefile`

L'exécutable ADA se compose :

- Du programme et paquetages écrits par l'étudiant
- Des bibliothèques (paquetage) Insa.* permettant d'accéder au drivers du logiciel résident (écran, touches, ...)
- De routines écrites en C permettant de faire le lien entre le logiciel résident et l'applicatif ADA
- De routines d'amorçage et d'initialisation écrites en ASM

Le programme est ensuite téléchargé dans la cible en utilisant le terminal ADA-Shell, qui sert aussi à la communication entre la cible et le PC (`Ada.Text_IO`)

ADAShell

Terminal graphique permettant la communication avec la cible - simulation des entrées sorties standards (écran TTY et clavier)

Permet la reprogrammation de l'exécutable ADA (fichier .hex ou .elf, selon le temps disponible)

Fonctionne sous Linux, écrit en C#/GTK#

Limitation et effets « bizarres »

La primitive/mot clef `delay` ne fonctionne pas comme prévu due à un code inadapté dans la bibliothèque ADA `Systeme.OS_Primitives`. Une fonction de remplacement (`SysDelay`) est fournie en attendant de patcher le paquetage ADA.

L'utilisation de l'optimisation (option -Ox) du GCC a pour effet de rendre les routines d'écritures (`Ada.Text_IO` et dérivées) begues ! Pour l'instant, pas de patch en vue.

Vu la quantité de mémoire très limitée sur la cible, la manipulation d'image et de son risque d'être compliquée. Une mémoire RAM externe est disponible sur la cible, mais ne peut être accédée directement (utilisation de fonctions). Pour information, une image complète occupe plus de 150 Ko (320*240*2 octets)

TODO

Finir de tester les prototypes de cible (RAM externe et carte SD)

Implémenter les routines de reprogrammation de l'exécutable (logiciel résident et ADAShell)

Finir le terminal ADAShell

Patcher la bibliothèque ADA en fonction des limitations rencontrées

Ajouter un serveur GDB pour permettre le debug des programmes ADA sur la cible (plus tard)