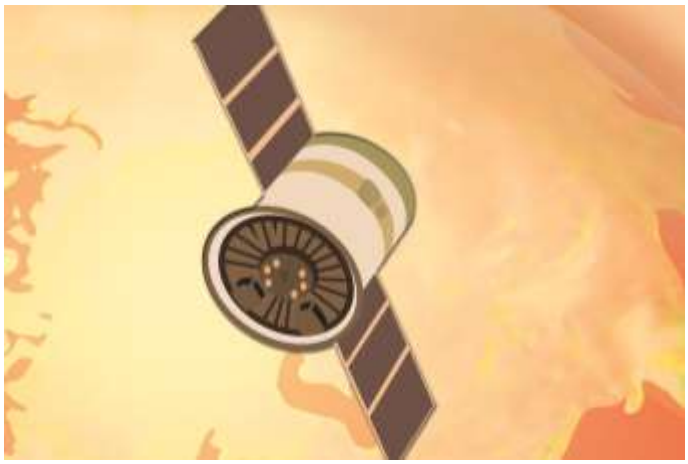
Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

Bd Pérolles 80
case postale 32
CH-1705 Fribourg
t. +41 (0)26 429 66 11
f. +41 (0)26 429 66 00
www.eia-fr.ch

Architecture et spécifications

Projet intégré



Laboratoire	Projet intégré – Systèmes embarqués 2
Professeur	D. Gachet
Etudiants	David Rossier, Loïc Gremaud, Romain Froidevaux
Classe	T-2a
Date	30.03.2014
Groupe	E

Hes·so FRIBOURG FREIBURG
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Table des matières

Architecture et spécifications

1. Introduction

1.1. Objectif

L'objectif principal de ce projet est de créer une application embarquée sur une cible APF27. Nous aurons à notre disposition une cible pour effectuer nos tests, avec un écran tactile LCD, six boutons poussoirs, un thermomètre et quatre affichages 7-segments. La communication entre le thermomètre et la FPGA se fera à l'aide d'un bus I2C.

Ce rapport a pour but de définir l'architecture globale de notre application, ainsi que les spécifications de chacun des composants et les liens entre eux.

1.2. Equipe

Afin de mener ce projet intégré à bien, nous avons formé un groupe de trois personnes : Loïc Gremaud, Romain Froidevaux et David Rossier.

1.3. Informations supplémentaires

Le code de notre application sera déposée sur un dépôt GIT, sur la forge de l'école, dans le branche groupe_e. Le tag goupe_e_incr001 a été crée pour cette étape du projet.

2. Architecture et spécifications

2.1. Structure de l'application

Ci-dessous un diagramme représentant les liens entre les composants. La direction des flêches indique le sens pour l'échange de données. Généralement ce sont les applications qui utilisent les API et les API qui appellent les drivers des entrées/sorties.

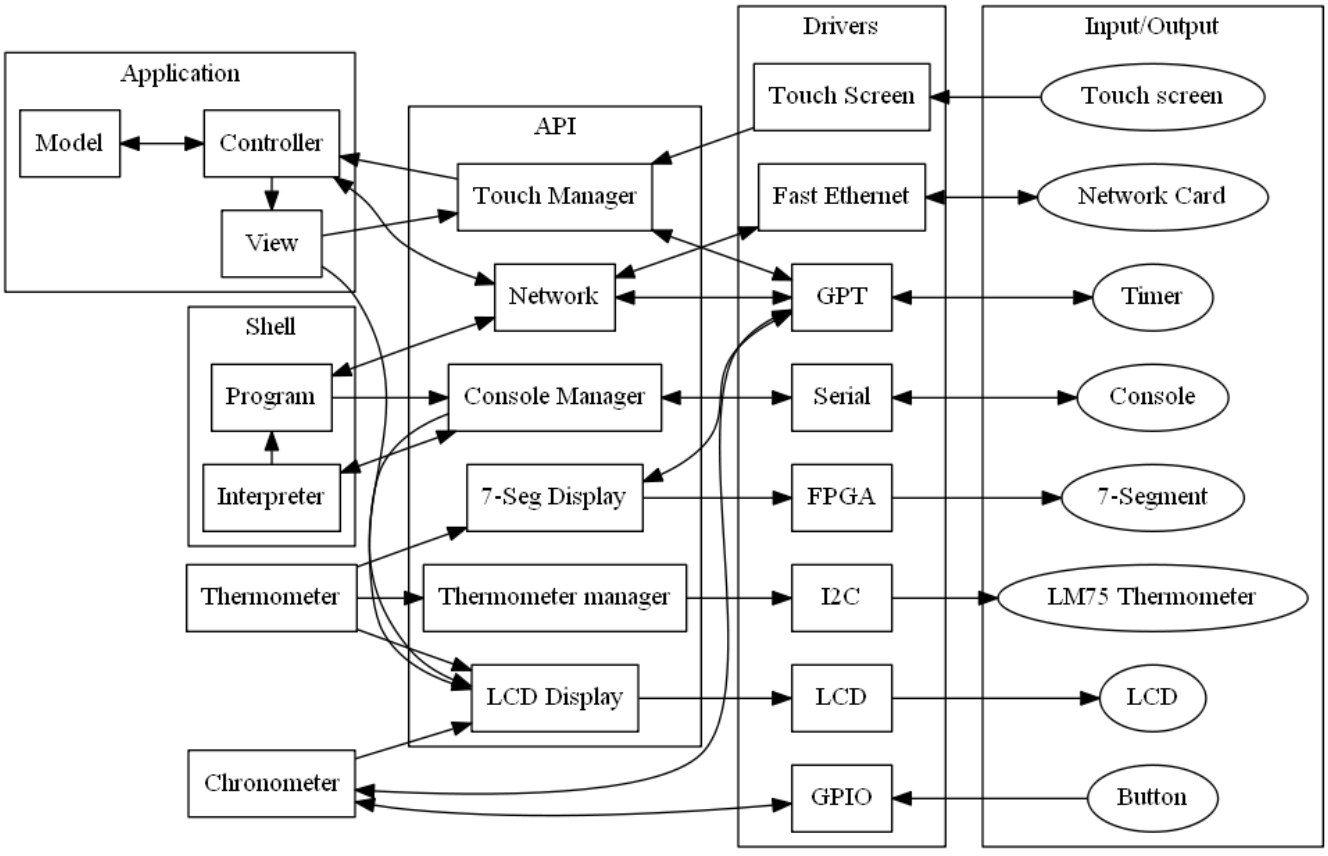


Figure 1 : Architecture générale

2.1.1. Description des drivers

Nom	Fichier	Description
Touch screen	touchscreen.h	Driver gérant les actions sur l'écran tactile
Fast Ethernet	fec.h	Driver pour le Fast Ethernet Controller
GPT	gpt.h	Driver pour gérer les horloges
Serial	serial.h	Driver pour l'interface serial
LCD	lcdc.h	Driver pour l'affichage sur l'écran LCD
FPGA	fpga.h	Driver pour gérer les boutons et les affichages 7-segments (FPGA)
I2C	i2c.h	Driver pour envoyer des données sur le bus I2C
GPIO	gpio.h	Driver pour gérer les boutons et les affichages 7-segments (GPIO)

2.1.2. Description des API

Nom	Fichier	Description
Touch Manager	touchmanager.h	API gérant les actions sur l'écran tactile
Network	network.h	API pour les connexions réseau
Console Manager	consolemanager.h	API pour gérer la console
LCD Display	lcddisplay.h	API pour gérer l'affichage sur l'écran LCD
7-Seg Display	seg7display.h	API pour gérer l'affichage 7-segments
Thermometer Manager	thermomanager.h	API pour gérer le chronomètre

2.1.3. Description des applications

Nom	Fichier	Description
GUI Applications	*_home.h *_score.h *_battleship*.h *_tictactoe.h	Écran d'accueil de la GUI Écran des scores Jeu « Bataille navale » Jeu « Tic-Tac-Toe »
Shell Program	shell_program.h	Exécution des commandes
Shell Interpreter	shell_interpreter.h	Interpréteur de la ligne de commande
Thermometer	thermometer.h	Gestion de la température
Chronometer	chronometer.h	Gestion du chronomètre

2.2. Interruption

Nous avons quatre composants qui utilisent les interruptions. Voici les différents niveaux d'interruptions que l'on gère.

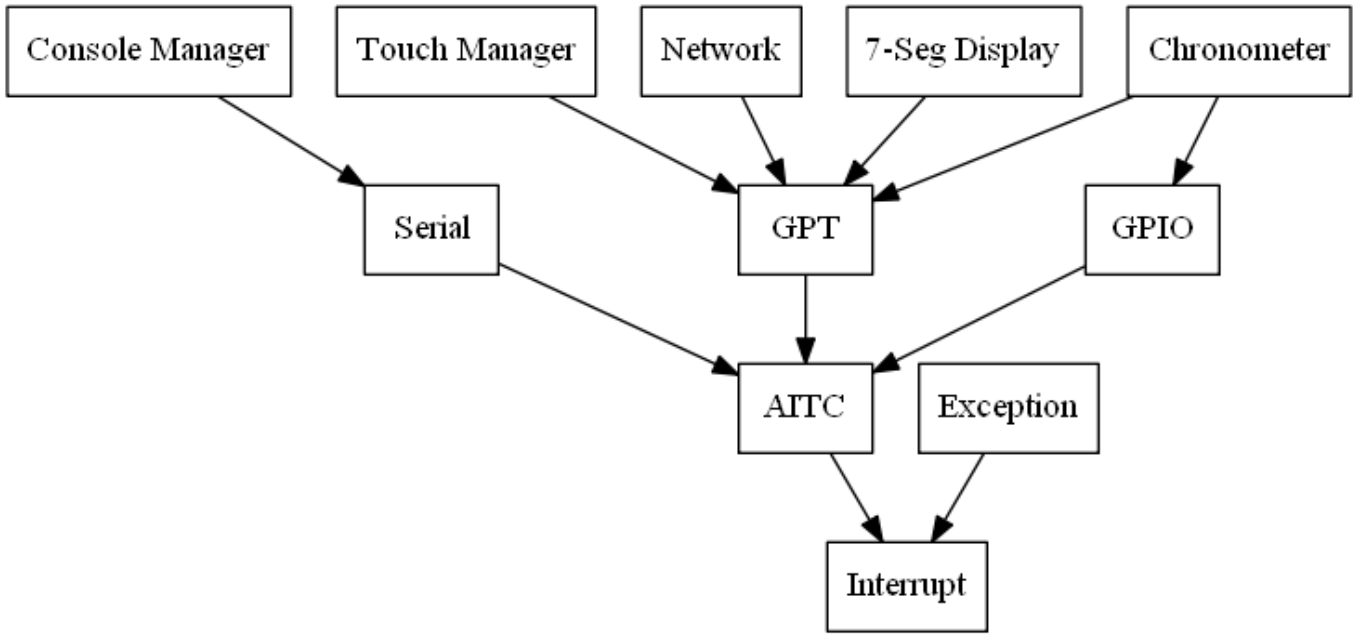


Figure 2: Interruption

Nom	Description
Interrupt	Gère les interruptions sur le microprocesseur.
AITC	Gère les interruptions matérielles.
Exception	Gère les interruptions lorsqu'une exception logicielle est levée.
Serial	Gère les interruptions lorsqu'un byte est reçu.
GPT	Gère les interruptions générées par les compteurs périodiques.
GPIO	Gère les interruptions générées par les boutons.

2.3. API « Network »

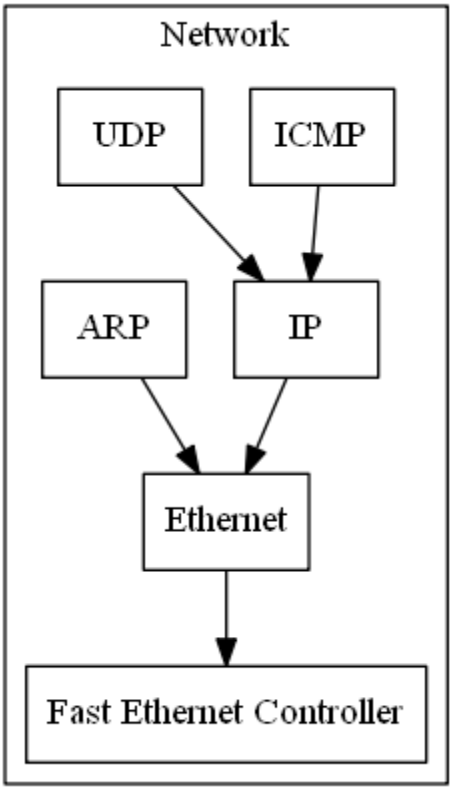


Figure 3 : Fonctionnement du réseau

Comme on peut le voir dans la figure ci-dessus, le réseau utilisera les différentes couches suivantes :

Nom	Description
UDP	Protocole de transport des données (couche 4)
ICMP	Protocole utilisé pour les PING (Couche 3 +)
IP	Protocole réseau (Couche 3)
ARP	Protocole utilisé pour la résolution MAC – IP (Couches 2-3)
Ethernet	Protocole de communication de couche 2.
FEC	Fast Ethernet Controller intégré à la cible

Chacune des couches devra être implémenter de façon à gérer l'encapsulation et la désencapsulation des en-têtes.

2.4. Shell Interpreter

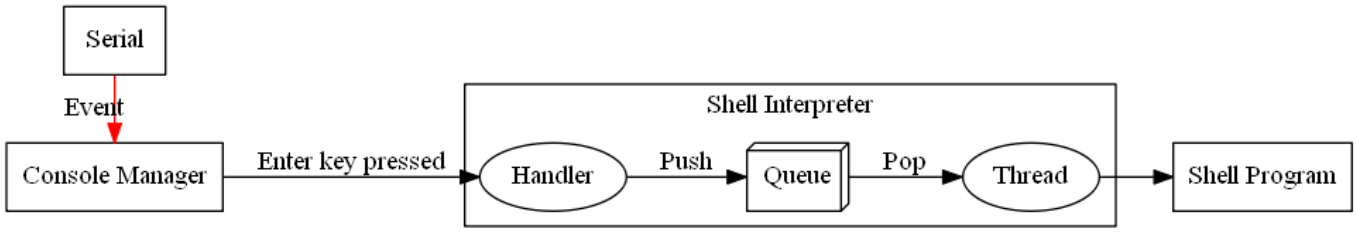


Figure 4 : Fonctionnement de l'interpréteur de commande (Shell)

Nom	Description
Serial	Passe les caractères au Console Manager.
Console Manager	Passe la chaîne de caractères au Handler, lorsque la touche entrée est pressée.
Handler	Met la chaine de caractère dans la queue pour un traitement ultérieur.
Queue	Contient la chaîne de caractère à traiter.
Thread	Traite la chaîne de caractère et appelle le bon programme.
Shell Program	Execution de la commande

2.5. Touch Manager

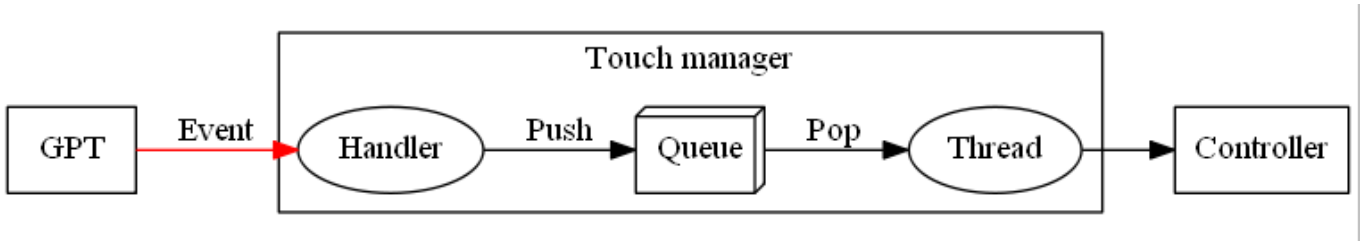


Figure 5 : Fonctionnement du TouchManager

Nom	Description
GPT	Appelle périodiquement Handle (fréquence correspond à la réactivité du tactile) et passe au Handler la position lors d'un clic.
Handler	Récupère la position et la place dans la queue pour un traitement ultérieur.
Queue	Contient les positions à traiter.
Thread	Traite les positions et appelle les fonctions du Controller correspondand à celle-ci.
Controller	Exécute l'action du bouton cliqué.

2.6. API « LCD Display »

L'API gère tous les affichages sur l'écran LCD de la cible. Elle peut être appelée par les différents éléments (contrôleurs, autres API, etc..) nécessitant une interactions avec l'interface graphique.

2.7. DoxyGen

Une documentation DoxyGen a été généré dans le but de connaître les spécifications de toutes les fonctions et les types utilisés pour communiquer en entre les différents modules. Les fichiers d'entètes utilisé pour la génération, nous permettra de démarrer plus rapidement la phase de conception.

3. Conclusion

L'établissement de l'architecture fut une aventure très intéressante. Nous avons eu l'occasion de mieux modéliser les liens entre chacunes de nos interfaces, drivers, entrées/sorties et applications.

Nous avons organisé une réunion pour discuter des composants communs à chacune de nos tâches, puis nous avons décidé de décrire les fonctions externes des composants dans les header files, avant de les représenter à l'aide du logiciel DoxyGen.

Fribourg, 30.03.2014

Rossier David, Loïc Gremaud, Romain Froidevaux

4. Annexe

1. Documentation DoxyGen