HEVs
haute école valaisanne
hochschule wallis
sciences de l'ingénieur

Filière Systèmes industriels

Orientation Infotronics

Diplôme 2006

Sébastien Farquet

Reconfiguration sécurisée

HEVs
Route du Rawyl 47
1950 Sion 2
haute école valaisanne
hochschule wallis

Professeur François Corthay

Expert Laurent Gauch

Sion, le 24 novembre 2006

Table des matières

1 PREFACE

1.1 Objectif du rapport

Ce rapport a pour but de documenter mon travail de diplôme sur la "Reconfiguration sécurisée". Il doit permettre à une personne externe de comprendre le travail effectué.

Mais comme il ne contient aucun détail sur l'implémentation des fonctions et que la majorité des fichiers implémentés ne sont pas sur le CD, il faut demander un accord à l'entreprise Amontec pour obtenir les droits de continuer le projet et obtenir les fichiers implémentés.

1.2 Conventions

Je procède différemment pour indiquer où se trouvent les informations complémentaires.

Si elles sont disponibles en annexe, j'écris l'indication : voir : *la lettre du chapitre le nom du chapitre* page n° *numéro de page*.

Sinon ce sont des documents qui se trouvent sur le CD ou sur internet.
Donc dans la majorité des cas, un lien est disponible dans la bibliographie pour indiquer où se trouve l'information. L'indication se présente sous la forme suivante : document : *le nom se retrouvant dans la bibliographie*.
De plus, certain de ces documents se retrouvent sur le CD. D'où l'indication : (lire : *le nom du fichier*).

rem : certaines abréviations et certains mots sont expliqués dans les annexes.
(Voir annexes : A Abréviations page n° 51 et B Glossaire page n°52)

1.3 Pré-requis

Quelques connaissances en informatique technique et en système numérique ainsi qu'un minimum de compréhension des termes anglais sont nécessaires à la bonne compréhension du présent document.

2 INTRODUCTION

Dans le monde actuel, la notion de sécurité des données prend de plus en plus d'importance. Pour satisfaire cette demande de sécurité, de nombreux algorithmes de cryptage ont été mis en place. Dans ce cadre de sécurité, il est aussi devenu nécessaire d'identifier la source.

C'est pourquoi, l'entreprise : Amontec souhaite fournir à ces clients un moyen d'identifier le périphérique connecté. De cette façon, l'utilisateur pourra réaliser un système de licence.

Alors, elle s'intéresse aux puces électroniques permettant d'identifier un équipement et de contrôler l'accès. Pour la réalisation de ce projet, l'entreprise : Amontec à choisi un composant de l'entreprise : MAXIM.

L'entreprise : MAXIM a mis au point des puces électroniques permettant d'être identifié par leur numéro de série qui est unique et non modifiable.
De plus, pour satisfaire le besoin de crypter l'information, certaines de ces puces électroniques sont équipées d'un système de cryptage.

Dans le cadre de ce projet, la puce électronique choisie par l'entreprise : Amontec est le DS2432. Cette puce électronique possède différentes caractéristiques permettant de nombreuses applications dans le domaine de la sécurité des données.
(Elle utilise l'algorithme de cryptage SHA-1.)

Cette puce électronique a été conçue pour permettre une utilisation diversifiée :

- identifier un équipement
- contrôler un accès
- crypter une transmission
- gérer une clef secrète

Pour pouvoir réaliser une des utilisations possibles, il faut communiquer avec la puce électronique. Etant donné que le DS2432 possède une interface 1-wire pour échanger des données, il faudra communiquer avec la technologie 1-wire ou un système compatible pour réaliser un système d'authentification.

3 CAHIER DES CHARGES

3.1 Description

Le but de ce travail de diplôme est de développer un système d'authentification basé sur le produit Amontec JTAGkey contenant une clef cryptée.

Le système sera connecté à un ordinateur par un port USB. Le logiciel doit pouvoir interroger le circuit et authentifier un utilisateur afin de pouvoir créer un système de licences.

Le logiciel comprend

- une base permettant de communiquer avec le circuit contenant la clef cryptée,
- une partie modifiable pour l'intégration d'une authentification dans un produit donné.

3.2 Objectif

- ☑ Comprendre le système de développement mis à disposition par le producteur du circuit à clef cryptée.
- ☑ Réaliser les 2 parties logicielles.
- ☑ Tester le système avec le "JTAGkey" de la société Amontec.
- ☑ Documenter le système pour des utilisateurs voulant intégrer l'authentification dans leur produit.

3.3 Remarques

Au cours de la réalisation du travail de diplôme, les objectifs ont changé.

Au commencement, l'application devait être réalisée sur un circuit reconfigurable. Mais pour les besoins du marché, l'entreprise Amontec a décidé de réorienter le projet vers une application en C.

Pour les précédents objectifs, je devais réaliser 3 parties logicielles :

- une base permettant de communiquer avec le circuit contenant la clef cryptée,
- une partie modifiable pour l'intégration d'une authentification dans un produit donné,
- une partie sécurisée effectuant le décryptage de la clef secrète.

Mais pour des raisons de sécurités, la clef doit uniquement être connue de l'entreprise : Amontec. Donc, l'application ne doit pas décrypter la clef mais, elle doit réaliser une fonction qui permet de retourner un MAC correspondant à celui réaliser par le circuit contenant la clef cryptée.
Et cette fonction sera intégré à la base permettant de communiquer avec le circuit contenant la clef cryptée.

4 SCHEMATISATION DE L 'OBJECTIF

Pour répondre aux attentes du mandant, je dois réaliser 2 parties et j'ai à disposition une partie matérielle.

La partie : JTAGkey est un périphérique USB.

La partie : utilisateur est un petit logiciel qui doit permettre une communication entre la partie : hôte et la partie : JTAGkey. Cette partie est destinée à se transformer en une bibliothèque de liaisons dynamiques.

La partie : hôte est un exemple d'application. Elle doit améliorer la compréhension de l'utilisateur afin qu'il puisse réaliser sa propre application.

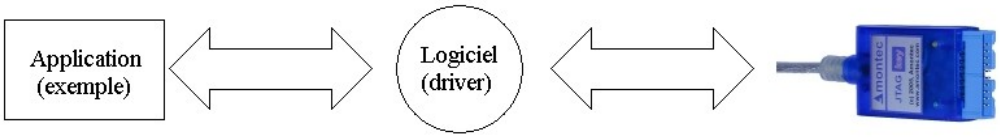


Figure 1 : Objectif imagé

L'hôte doit pouvoir :

- prendre contact avec l'utilisateur => se connecter,
- savoir qui est le périphérique => demander son MAC,
- vérifier qui est le périphérique => calculer son MAC,
- comparer les données => comparer les 2 MAC,
- congédier l'utilisateur => se déconnecter.

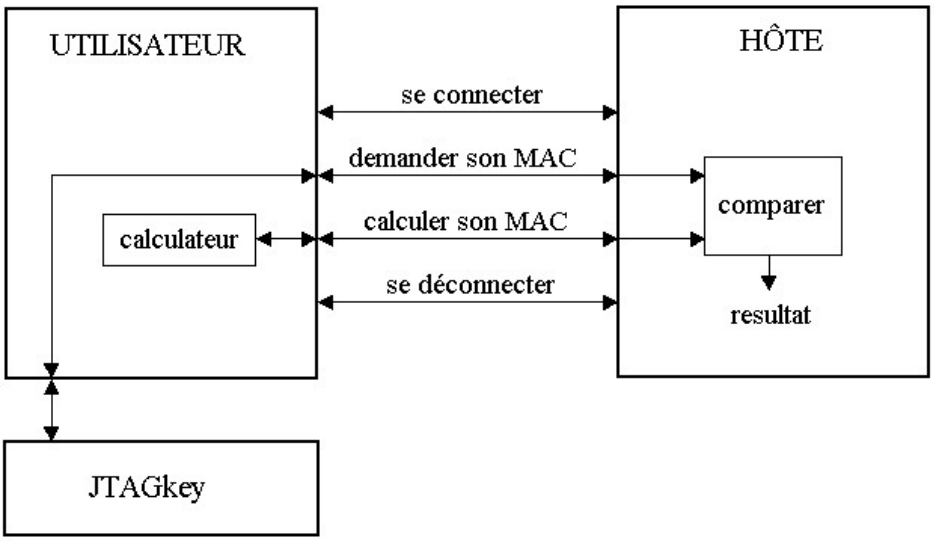


Figure 2 : Principe de fonctionnement

Au final la partie :

- JTAGkey se connecte à un port USB d'un PC
- utilisateur fournit quelques fonctions mais l'implémentation sera cachée
- hôte est une application libre de droits

Pour une meilleure compréhension, voici de façon plus détaillée ce qui est échangé.

Pour toutes les fonctions, je travaille avec un point d'entrée (jkHandle) et un statu (jksStatus). Ainsi, la partie : utilisateur connaît quelle JTAGkey est utilisée et la partie : hôte peut savoir comment c'est déroulé la fonction exécutée.

Les MACs sont obtenus en utilisant l'algorithme de cryptage SHA-1.

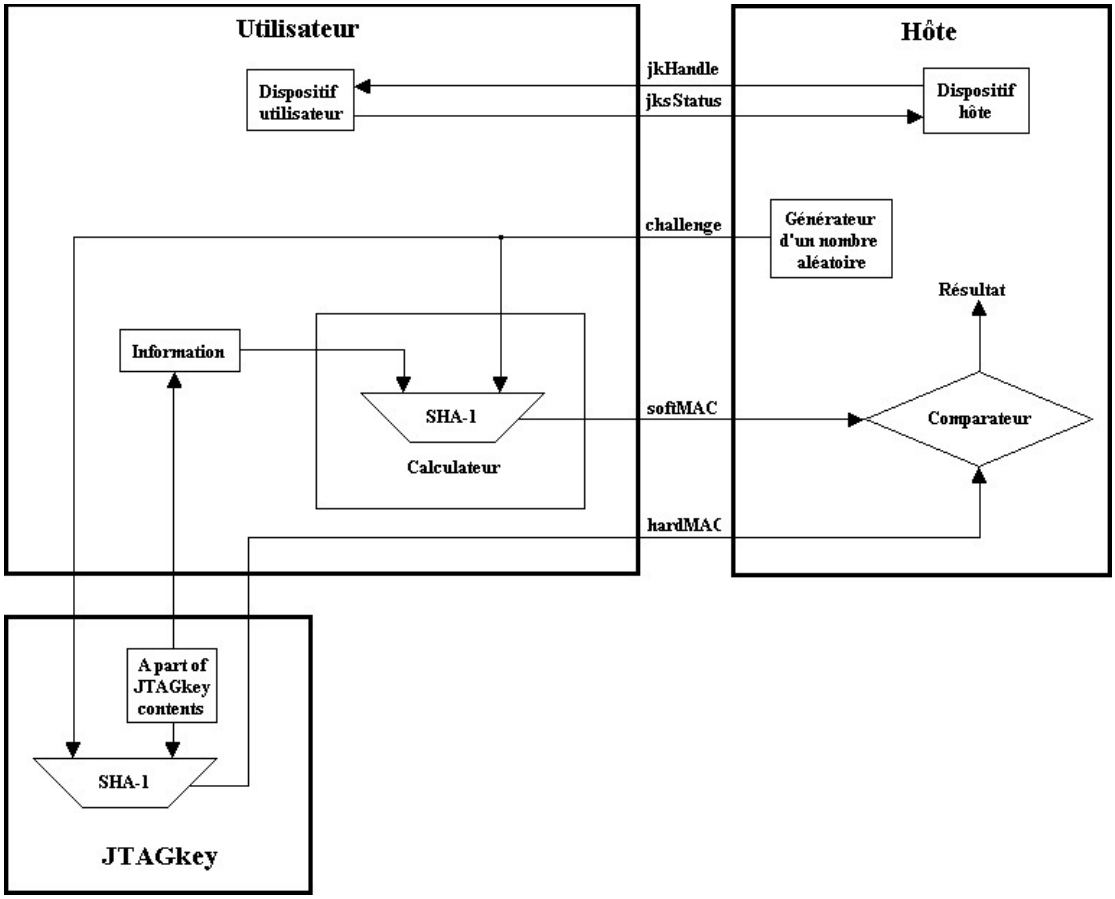
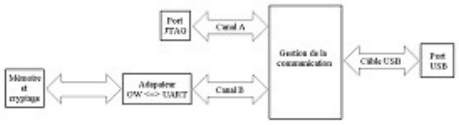


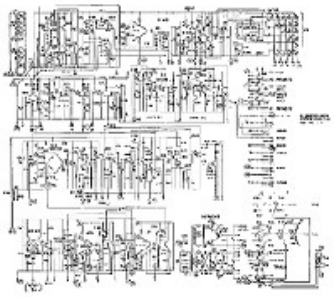
Figure 3 : Mécanisme pour JTAGkey SaferReCS

5 MATERIEL A DISPOSITION

Le périphérique
(à disposition)



L'idée générale
(nécessaire)



Le schéma
(pariellement connu)

Image 1 : Matériel à disposition

Structure du chapitre :

- Introduction
- Décomposition de la JTAGkey
- Gestion de la communication
- Mémoire et cryptage
- Adaptation UART/OW
- Complément d'information

5.1 Introduction

Le matériel à disposition est une JTAGkey. Cet élément est fournit par l'entreprise : Amontec. La partie : JTAGkey communique avec la partie : utilisateur (voir : Figure 2 : Principe de fonctionnement page n° 6).

La JTAGkey est composée d'une multitude de composants. Mais pour mon projet, j'avais besoin de savoir l'essentiel afin de réaliser l'application.

N'ayant pas reçut le schéma électronique complet de la JTAGkey, ce qui suit est uniquement une représentation permettant d'avoir les informations nécessaires à la réalisation du projet.

5.2 Décomposition de la JTAGkey

Le fonctionnement général de la JTAGkey peut être représenté à l'aide de 5 éléments.

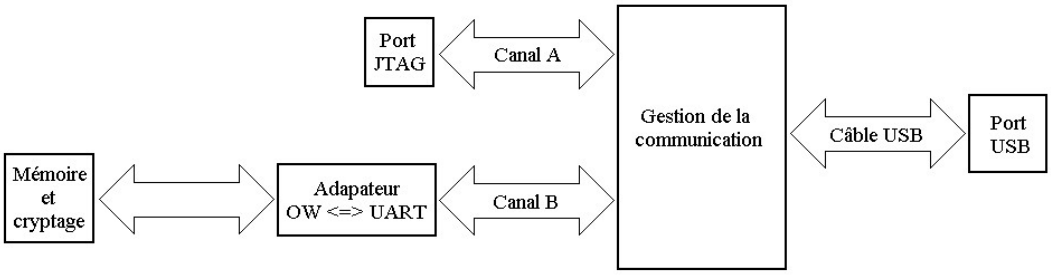


Figure 4 : Composition de la JTAGkey

Pour le projet, je me suis intéressé au 3 parties suivantes :

- Gestion de la communication
- Mémoire et cryptage
- L'adaptateur UART/OW

5.3 Gestion de la communication

La partie : gestion de la communication est réalisée par la puce électronique : FT2232L.

Cette puce électronique permet de gérer au mieux la communication nécessaire au bon fonctionnement de la JTAGkey.
D'un côté, elle communique avec un flux USB et de l'autre, elle communique sur le canal A ou le canal B sous forme d'un flux USB/UART/Bit-Bang/(autres).

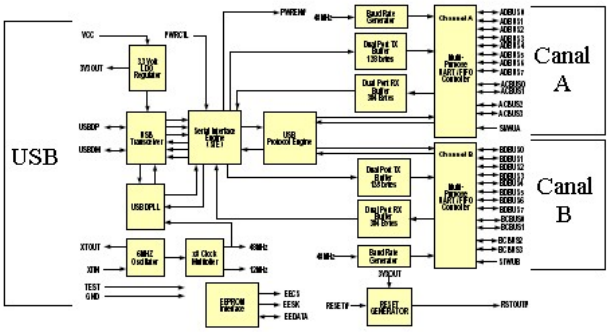


Figure 5 : Représentation du FT2232

Pour piloter cette puce électronique, il faut utiliser le driver fournit par l'entreprise : Amontec.
Ce driver permet d'appeler plusieurs fonctions pour transmettre, gérer et paramétrer la puce électronique afin de réaliser l'opération souhaitée.

Pour avoir plus d'information sur le FT2232L, lire le document : feuille d'information du FT2232L (lire : FT2232L.pdf).
Et pour avoir des informations sur les fonctions disponibles, lire le document : Guide de programmation pour le FT2232L (lire : D2XXPG33.pdf).

5.4 Mémoire et cryptage

La partie : mémoire et cryptage est réalisée par la puce électronique : DS2432.

Cette puce électronique possède une alimentation parasite, un identifiant unique, un système permettant de réaliser l'algorithme SHA-1, un système pour sécuriser la communication, de la mémoire et une zone spéciale pour écrire une clé secrète.

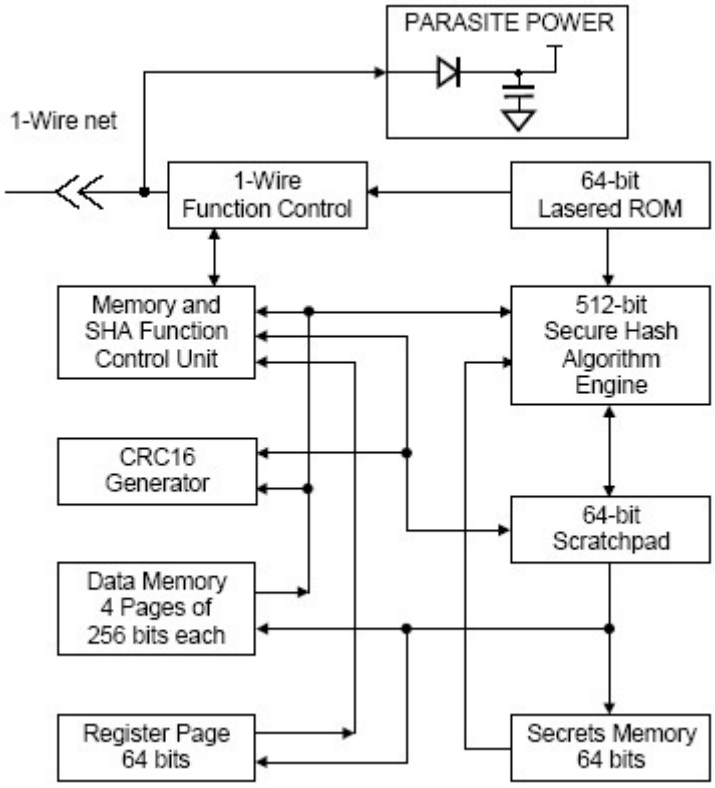


Figure 6 : DS2432

Pour communiquer avec le DS2432, il faut utiliser l'interface OW.
Et pour effectuer une opération disponible, il faut suivre une séquence afin de sélectionner la bonne opération.

Pour plus d'information sur le DS2432 et savoir qu'elles sont les fonctions réalisables avec le DS2432, voir annexes : E DS2432 page n° 60.
Et pour avoir des informations complètes sur le DS2432, lire le document : feuille d'information du DS2432 (lire : DS2432.pdf).

5.5 Adaptation UART/OW

Pour que la communication soit effective, il faut que la partie : mémoire et cryptage comprenne la partie : gestion de la communication.

Comme la partie : mémoire et cryptage utilise une interface OW et que la partie : gestion de la communication ne peut pas paramétrer le canal B pour obtenir un flux OW, il faut un adaptateur.

L'entreprise : Amontec a choisit de réalisé un montage permettant d'adapter une trame UART en un signal OW.

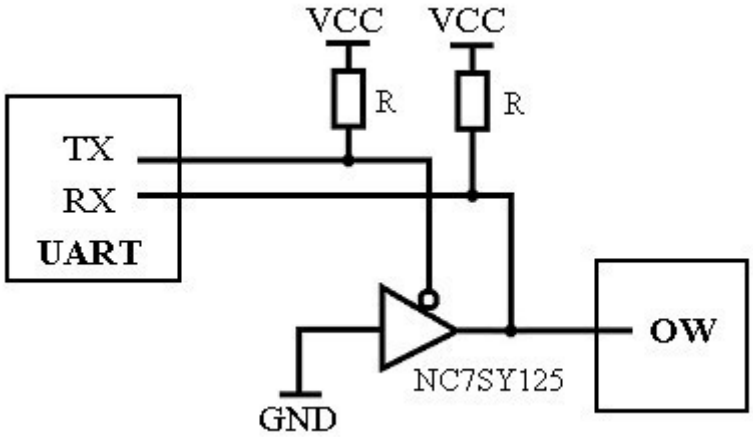


Figure 7 : Adaptateur UART/OW

Pour avoir des informations sur le NC7SZ125, lire la feuille d'information : NC7SZ125.pdf.

Pour avoir plus d'information sur la façon d'adapter une trame UART en un signal OW, voir annexes : F Adaptation UART/OW page n° 70.

5.6 Complément d' information

5.6.1 Flux OW

La communication OW se passe sur un seul fil et de manière asynchrone.
Les données sont transmises en série.
La vitesse de communication est limitée par la durée des signaux.

Il existe 4 types de signaux :

- remise à zéro et l'impulsion de présence
- lire un '1' ou un '0'
- écrire un '1'
- écrire un '0'

Et ces signaux peuvent être en mode normal ou accéléré.

Pour avoir plus d'information sur la technologie OW, voir annexes : C 1-WIRE page n° 53.

5.6.2 Flux UART

La communication UART se passe sur 2 fils et de manière asynchrone.
Les données sont transmises en série.
La vitesse de communication est choisie (et se calcule en baud).

Les données sont transmises sous forme de trame :

- 1 start bit
- 5 à 8 bits de données
- 0 ou 1 bit de parité
- 0 à 2 stop bit(s)

Pour avoir plus d'information, lire la page internet: Universal asynchronous receiver/transmitter.

5.6.3 Algorithme SHA-1

L'algorithme SHA-1 est un algorithme de cryptage.
Il faut lui fournir un message de 512 bits et il retourne un code d'authentification de ce message sur 160 bits appelé MAC.

Pour avoir plus d'information sur l'algorithme SHA-1, voir annexes : D SHA-1 page n° 58.

6 DECOMPOSITION DES TACHES

6.1 SDK MAXIM

- ☑ Analyser partiellement le SDK
- ☑ Choisir une architecture fichier
- ☒ Implémenter les fichiers
  - ☑ Fichier de liaison avec l'OS
  - ☑ Fichier de liaison 1-Wire
  - ☑ Fichier de session 1-Wire
  - ☒ (Toutes les fonctions sont implémentées)
- ☑ Faire fonctionner tstfind.c
- ☑ Faire fonctionner shaap.c
- ☑ Analyser le fonctionnement du fichier shaap.c

6.2 Réaliser l' implémentation

- ☑ Réaliser la partie : utilisateur
- ☑ Réaliser la partie : hôte

6.3 Tester le système avec la "JTAGkey" de la société Amontec

- ☑ Test du bon fonctionnement du fichier tstfind.c
- ☑ Test du bon fonctionnement du fichier shaapp.c
- ☑ Test du bon fonctionnement de la partie : utilisateur
- ☑ Test du bon fonctionnement de la partie : hôte

6.4 Documentation

- ☑ Fascicule en anglais pour des utilisateurs voulant intégrer l'authentification dans leur produit.
- ☒ Commenter le code de la partie : utilisateur
- ☑ Commenter le code de la partie : hôte
- ☒ Réaliser un petit document montrant le fonctionnement de la partie : utilisateur

7 SDK MAXIM

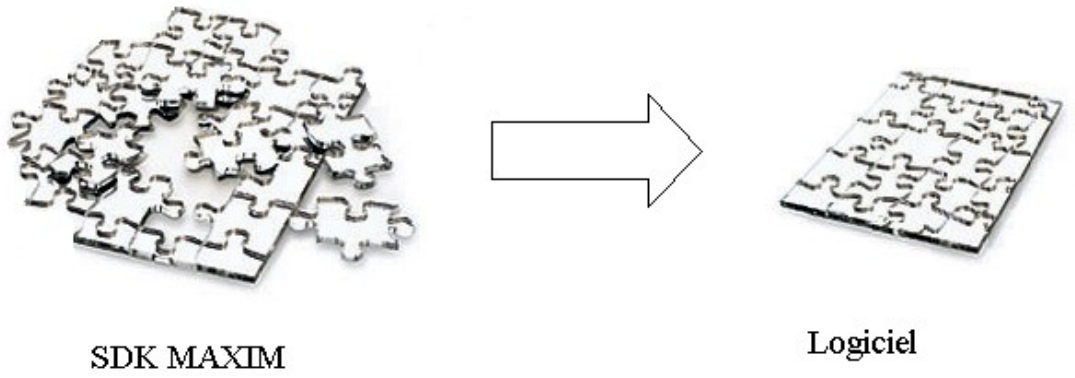


Image 2 : SDK MAXIM

Structure du chapitre :

- Introduction
- Analyse
  - Regroupement des fonctions
  - Séquence d'utilisation
  - Choix des applications
  - Fichiers à inclure
  - Architectures

7.1 Introduction

Le SDK de MAXIM doit me permettre de communiquer avec la partie : JTAGkey donc il doit me servir à la partie : utilisateur (voir : Figure 2 : Principe de fonctionnement page n° 6).

Le fabricant: MAXIM met à disposition 5 API :

- OWPD : 1-Wire Public Domain
- OWAPI : 1-Wire API for Java
- OW.NET : 1-Wire API for .NET
- OWCOM : 1-Wire COM
- TMEX : TMEX

Pour ce projet, il faut que l'application soit écrite en C, libre de droit et si possible portable sur différents systèmes d'exploitations (Windows, linux, …).

Donc après avoir parcouru le document : 1-Wire Software Resource Guide Device Description (lire : AN155.pdf), j'ai choisit de travailler avec l'interface : 1-Wire Public Domain.

7.2 Analyse

7.2.1 Regroupement des fonctions

Toutes les interfaces de programmation pour applications possèdent le même regroupement de fonctions.

SESSION
Fonctions à l'utilisation exclusive du bus 1-Wire. C'est particulièrement important sur les systèmes d'exploitations ou les environnements où plusieurs processus ou tâches pourraient accéder simultanément accéder au même bus.

LINK
Fonctions primitives de communication sur le bus 1-Wire.

NETWORK
Fonctions de réseau pour la découverte et la sélection de dispositif. Le numéro de série unique de chaque dispositif 1-Wire est employé en tant qu'adresse réseau.

TRANSPORT
Communication par bloc et fonctions primitives lecture/écriture de mémoire.

FILE
Fonctions de la mémoire en utilisant la structure de dossier 1-Wire. (Voir: Application Note 114).

DEVICE
Fonctions spécifiques au dispositif utilisé.

Figure 8 : La façon de réunir les fonctions (propre au SDK)

7.2.2 Séquence d'utilisation

Toutes les applications du SDK suivent la même séquence de travail.

1. Acquisition du réseau OW
2. Sélection d'un dispositif OW puis réalisation d'une fonction
3. Selon l'application, revient à l'étape 2 sinon passe à l'étape 4
4. Libère le réseau OW
5. Réalisation d'autres tâches et propose de recommencer à l'étape 1 ou l'application est terminée.

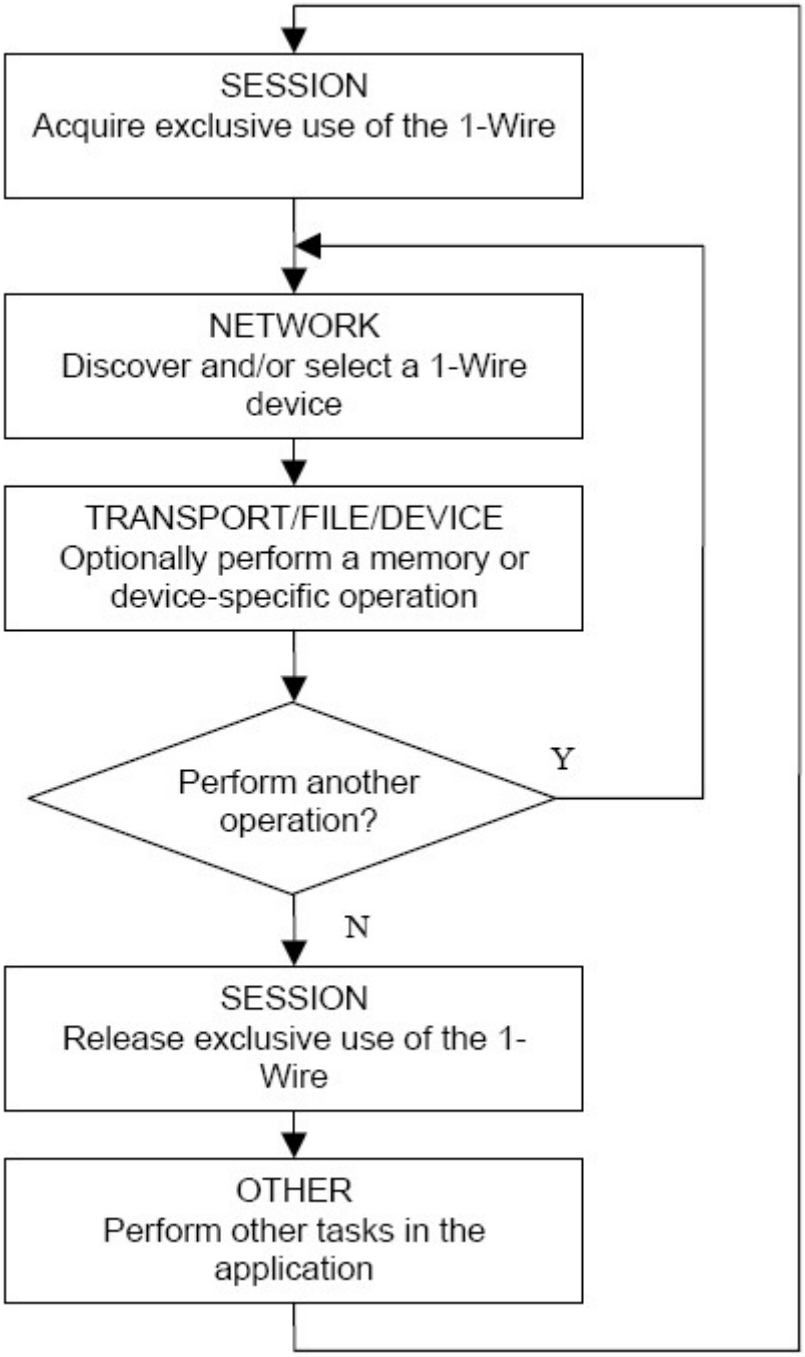


Figure 9 : Séquence pour les applications (OWPD300)

Données sensibles transmises au mandant :

nom de la compagnie	: Amontec, Gauch
rue	: La Grand Fin 123
NPA / numéro postal	: 1633
localité	: Vuippens
pays	: Switzerland
FAX	: +41 (0)26 915 22 50
e-Mail	: info@amontec.com
site Web	: http://www.amontec.com/

Sensitive data sent to the company :

company name	: Amontec, Gauch
address	: La Grand Fin 123
zip / postal code	: 1633
city	: Vuippens
country	: Switzerland
FAX	: +41 (0)26 915 22 50
e-Mail	: info@amontec.com
website	: http://www.amontec.com/

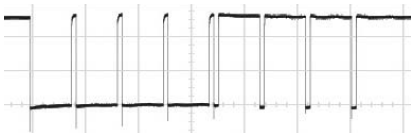
Sensible Daten an den Auftraggeber :

Firmenname	: Amontec, Gauch
Straße	: La Grand Fin 123
PLZ / Postleitzahl	: 1633
Ort	: Vuippens
Land	: Switzerland
FAX	: +41 (0)26 915 22 50
e-Mail	: info@amontec.com
Webseite	: http://www.amontec.com/

11 PROBLEMES RENCONTRES

- Installation du driver.
  S'il y a eu une précédente installation d'un périphérique d'Amontec ou d'un périphérique utilisant une puce de FTDI, il se peut que windows XP conserve des informations sur l'ancien driver.
  Donc pour en installer un nouveau, il faut effacer toutes les traces de l'ancien. (Principalement dans le répertoire de windows).

- Embrouille avec les bits OW et les bits UART.
  Car un bit OW est un signal : '1' ou '0' et il est fabriqué avec 10 bits UART. (Le signal reset est lui aussi composé de 10 bits UART.)
  Donc, il faut faire attention à comparer les bons éléments.
  Voir annexes : F Adaptation UART/OW page n° 70.

- Transmission OW.
  Pour transmettre le code d'une opération, il faut transmettre le bit de poids faible en premier.
  Ex, la fonction de recherche : F0 => 1111000
  Mesuré à l'oscilloscope :  => 00001111

- Protections de la puce électronique DS2432.
  Les fonctions d'écriture sont influencées par les protections. Et dès que ces protections sont activées, elles ne peuvent plus être désactivées.
  (Ceci pour des raisons de sécurité.)

- Le classement des canaux pour l'ouverture par index.

1 JTAGkey

Index	Info
0	canal A
1	canal B

2 JTAGkey

Index	Info
0	canal A
1	canal A
2	canal B
3	canal B

3 JTAGkey

Index	Info
0	canal A
1	canal A
2	canal A
3	canal B
4	canal B
5	canal B

Test avec plusieurs JTAGkeys.
Pour réaliser un test avec plusieurs JTAGkeys, j'utilisais les ports USB de l'écran.
Le problème, c'est que lorsque que l'écran est désactivé, les JTAGkeys sont aussi désactivées.
Donc quand l'écran se désactivait, j'obtenais quelques erreurs le temps que la JTAGkey connectée au PC soit utilisée par l'application.

12 AMELIORATIONS

- En ce moment, la partie : utilisateur ne peut traiter qu'une JTAGkey après l'autre. Donc, ce serait bien qu'il puisse travailler avec plusieurs JTAGkey.
  (Mémoire dynamique et pas un tableau de structure.)

- Pour améliorer la protection donnée par la clef, la partie : utilisateur pourrait être doté d'une variable interne à la structure pour permettre à la partie : utilisateur de savoir s'il peut exécuter la fonction.
  (Lors de l'authentification, toutes les fonctions doivent pouvoir s'exécuter.)

- Définir des textes d'erreur et une fonction permettant de retourner ce texte. Afin que l'utilisateur puisse accéder à un texte par défaut pour traiter les erreurs.

- Pour optimiser la taille du fichier, il faut modifier les variables temporaires dans les fonctions afin qu'elles aient la taille minimum.

- Améliorer le traitement d'erreur en analysant tous les cas, tous les retours obtenus lors d'opérations avec la puce électronique : FT2232L ou DS2432.

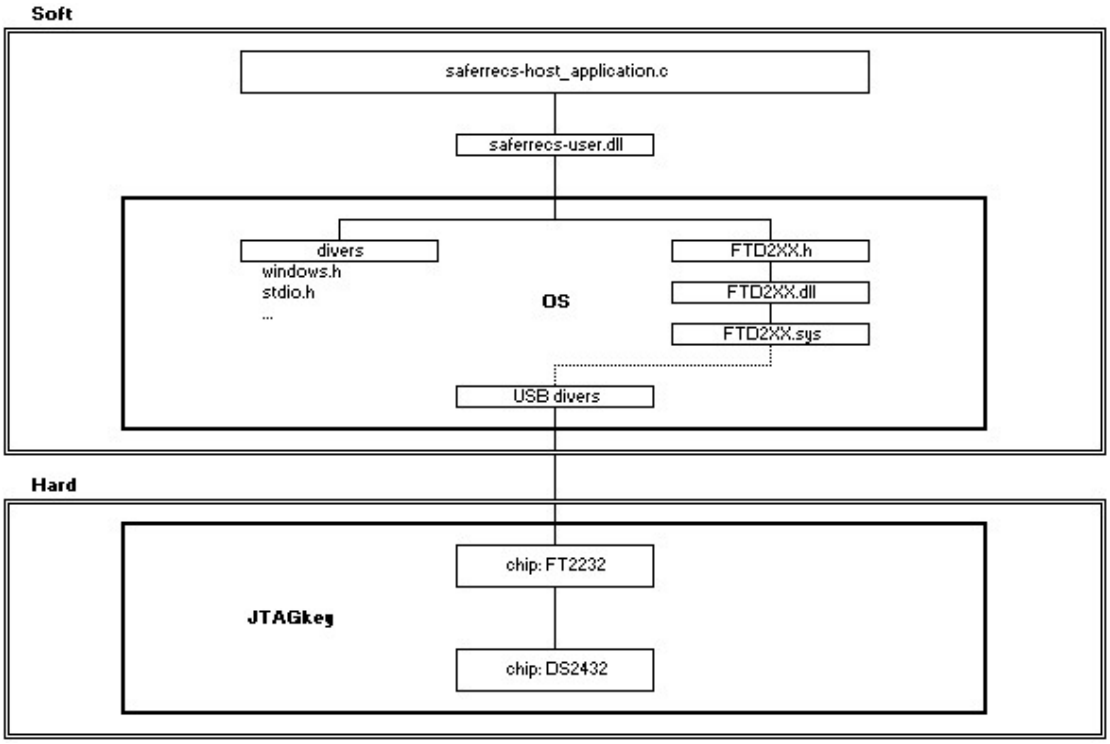
- Pour pouvoir retourner le statu de la JTAGkey à tout moment, il faut que le statu soit stocké dans la structure et qu'il soit mis à jour à chaque modification et au début de chaque fonction.

Améliorer le changement de mode de communication du DS2432. Pour le moment, la partie : utilisateur permet d'être soit en mode normal soit en mode accéléré. Mais, une fois l'application lancée, le mode ne peut plus être changé.

13SUITE DU PROJET

Pour l'entreprise Amontec:

- rendre la partie : utilisateur compatible pour le système d'exploitation linux
- créer une dll et son équivalent sous linux
- décrire une façon de cacher la clef dans la dll
- adapter la partie : hôte pour la rendre compatible au système d'exploitation linux



Architecture 5 : Dll

Réaliser le même projet mais faire le développement pour du VHDL afin de l'intégrer à un circuit reconfigurable.

Réaliser un projet utilisant les pages mémoires. Par exemple pour récolter des informations, augmenter le degré de sécurité, ...

14 CONCLUSION

Le système d'authentification basé sur le produit Amontec JTAGkey contenant une clef cryptée a été réalisé. On dispose d'une partie : utilisateur qui permet à la partie : hôte d'authentifier un périphérique (JTAGkey) connecté sur un port USB.

La partie : hôte contient un exemple de système d'authentification.
Et elle contient aussi un système pour entrer une nouvelle clé, un système pour écrire une ligne, un système pour afficher tout le contenu de l'EEPROM ainsi qu'un système qui pourrait permettre la réalisation d'un patch de la JTAGkey (pour autant que la partie : utilisateur soit modifiée).

Pour faciliter la compréhension des fonctions mise à disposition, une documentation en anglais a été réalisée.
(Voir document : Amontec JTAGkey : SaferReCS user interface (lire : amt_ann008.doc).

14.1 Maîtrise des objectifs

Les objectifs décrits dans le cahier des charges sont atteints à l'exception d'un test effectué par plusieurs personnes pour observer si la partie : utilisateur réagit correctement aux situations crées par les testeurs avec l'application : saferrecs-host_picked.c.

Mais selon la décomposition des tâches que je voulais réaliser. Je n'ai pas réussi à implémenter toutes les fonctions pour avoir une compatibilité complète avec les applications du SDK de MAXIM.
(Voir : 8.1 Applications du SDK page n° 21)
De plus, je n'ai pas eu le temps de réaliser toute la documentation souhaitée. La partie : utilisateur n'est quasiment pas commentée et je n'ai pas eu le temps de faire un document pour expliquer le fonctionnement de la partie : utilisateur.

Sion, le vendredi 24 novembre 2006

Sébastien Farquet

15 BIBLIOGRAPHIE

Ce chapitre permet de retrouver tous les écrits utilisés pour la réalisation de ce rapport.

Image 9 : Bibliographie

Parties du chapitre :

- Matériel à disposition
- SDK MAXIM

15.1 Matériel à disposition

Fabricant : Amontec
http://www.amontec.com

Driver pour la gestion de la communication :
http://www.amontec.com/jtagkey.shtml#drivers

Fabricant : FTDI
http://www.ftdichip.com

Feuille d'information du FT2232L :
http://www.ftdichip.com/Documents/DataSheets/ds2232l_15.pdf

Guide de programmation pour le FT2232L (D2XXPG33) :
http://www.ftdichip.com/Documents/ProgramGuides/D2XXPG33.pdf

Fabricant : MAXIM (Dallas semiconductor)
http://www.maxim-ic.com

Feuille d'information du DS2432 :
http://www.maxim-ic.com/parts.cfm/p/ds2432

Fabricant : National semiconductor
http://www.national.com

Feuille d'information sur le NC7SZ125 :
http://www.ortodoxism.ro/datasheets/nationalsemiconductor/DS012161.PDF

Universal asynchronous receiver/transmitter :
http://en.wikipedia.org/wiki/UART

15.2 SDK MAXIM

Fabricant : MAXIM (Dallas semiconductor)
http://www.maxim-ic.com

1-Wire Software Resource Guide Device Description :
http://pdfserv.maxim-ic.com/en/an/AN155.pdf

Information sur les kits de développement pour logiciel :
http://www.maxim-ic.com/products/ibutton/software/sdk/sdks.cfm

1-Wire Public Domain Kit :
http://www.maxim-ic.com/products/ibutton/software/1wire/wirekit.cfm

Le kit de développement pour logiciel: OWPD version 3.00 :
http://files.dalsemi.com/auto_id/public/owpd300.zip

Using the 1-Wire Public Domain Kit:
http://pdfserv.maxim-ic.com/en/an/wp2.pdf

Feuille d'information sur le DS2480B :
http://www.maxim-ic.com/parts.cfm/p/ds2480B

Modèle OSI :
http://sebsauvage.net/comprendre/tcpip/osi.html

Représentation du modèle OSI :
http://sebsauvage.net/comprendre/tcpip/protocols.pdf

16TABLES COMPLEMENTAIRES

16.1 Table des images

16.2 Table des figures

16.3 Table des architectures