

M1 - 2011

Portage d'une bibliothèque C++ standard dans MutekH

Error: Macro Include(StageContexte/MutekH) failed

```
pre-0.11 Wiki macro Include by provider <class 'includemacro.macros.IncludeMacro'> no longer sup
```

Objectif

L'objectif de ce stage est le portage d'une bibliothèque standard C++ dans MutekH. La bibliothèque standard C++ permet la compilation et l'exécution de programmes en C++ et s'appuie sur les fonctionnalités de la bibliothèque C standard déjà existante.

Le stagiaire devra compiler une des bibliothèques standards C++ libres et l'intégrer dans MutekH. L'exécution d'applications C++ sur MutekH permettra naturellement de valider le stage.

Encadrement

L'encadrement de ce stage sera effectué par Joël Porquet.

Portage de la bibliothèque 3D TinyGL dans MutekH

Error: Macro Include(StageContexte/MutekH) failed

```
pre-0.11 Wiki macro Include by provider <class 'includemacro.macros.IncludeMacro'> no longer sup
```

Objectif

L'objectif de ce stage est de porter la bibliothèque TinyGL dans MutekH.

La bibliothèque TinyGL (<http://bellard.org/TinyGL/>) est une implémentation minimaliste et légère d'OpenGL sans accélération matérielle. Elle est particulièrement bien adaptée à l'embarqué. Elle permet de rendre des scènes animées en 3D dans un frame-buffer.

Le système d'exploitation MutekH dispose de toutes les ressources pour accueillir cette nouvelle bibliothèque : des drivers de framebuffer, une bibliothèque mathématique...

Le travail de ce stage consiste à compiler TinyGL avec MutekH et à écrire le backend pour exploiter l'interface de frame buffer. Le stagiaire pourra valider son travail en exécutant les programmes de démonstration fournis avec TinyGL ou d'autres programmes 3D de son choix ou de sa création. L'expérimentation pourra être réalisée sur une plateforme SoCLib ou PC.

Encadrement

L'encadrement de ce stage sera effectué par Joël Porquet.

M1 - 2010

Injection de pannes dans les modèles de composants SoCLib

Error: Macro Include(StageContexte/Soclib) failed

```
pre-0.11 Wiki macro Include by provider <class 'includemacro.macros.IncludeMacro'> no longer sup
```

Objectif

Le simulateur SoCLib permet d'étudier le comportement des circuits. Cependant les pannes dues à l'électronique ne sont actuellement pas prises en compte. En effet, en situation réelle, certains éléments matériels peuvent cesser de fonctionner. Le projet Adam vise l'élaboration d'une plateforme capable de gérer ces situations et nécessite donc leur prise en charge dès la simulation.

L'objectif de ce stage est d'ajouter la capacité d'injection de pannes franches aux modèles SoCLib actuels. La simulation de pannes avec SoCLib permettra de valider différents protocoles de test directement depuis SystemC.

Dans un souci de modularité, l'injection de pannes devra être la moins intrusive possible au niveau de l'architecture des composants SoCLib, et facilement désactivable.

Pour commencer, le stagiaire devra se familiariser avec les modèles SoCLib et les différents types de pannes franches existantes. La validation se fera par l'exécution sur MutekH, d'une application de tests logiciels directement dans la simulation.

Encadrement

L'encadrement de ce stage sera effectué par Dimitri Refauvelet.

Développement du composant SoCLib UART 16550 et de son driver MutekH

Error: Macro Include(StageContexte/SoclibMutekH) failed

```
pre-0.11 Wiki macro Include by provider <class 'includemacro.macros.IncludeMacro'> no longer sup
```

Objectif

L'objectif de ce stage est d'obtenir un modèle de simulation SoCLib d'une UART, un modèle VHDL équivalent et le pilote du périphérique pour MutekH.

Une UART est un périphérique de communication simple permettant d'établir une liaison série entre deux machines, c'est le composant qui supporte la norme RS232 sur les PCs. L'UART 16550 dont il est question est compatible avec celle des PC.

Le modèle de composant SoCLib se rapproche beaucoup du modèle de TTY déjà existant et de son driver associé dans MutekH. Un driver pour les UART 16550 supportant à l'heure actuelle uniquement la plateforme PC existe également. Un modèle VHDL existe déjà et doit être adapté pour que son jeu de registres corresponde à la spécification choisie ici.

Tous ces éléments devront donc être développés ou adaptés pour obtenir un ensemble uniforme et cohérent.

[?doc](#)

Encadrement

L'encadrement de ce stage sera effectué par Christophe Choichillon.