



Systèmes Embarqués 1 & 2

Classes T-2/I-2 // 2018-2019

tp.01 – Introduction

Installation de l'environnement de développement

1 Installation sur une machine personnelle Linux

- Créer l'espace de travail (workspace)
Ouvrir un terminal (une shell Linux) et créer le répertoire de travail (workspace)

```
$ mkdir -p ~/workspace
```
- Configurer GIT

```
$ git config --global user.name <Firstname Lastame>  
$ git config --global user.email user.name@edu.hefr.ch
```
- Cloner le dépôt GIT contenant les scripts d'installaton de l'environnement

```
$ cd ~/workspace  
$ git clone https://gitlab.forge.hefr.ch/embsys/baremetalenv.git
```
- Installer l'environnement et générer les bibliothèques pour la cible

```
$ ./baremetalenv/bbb-env
```
- Installer le plug-in C/C++ GDB Hardware Debugging
 - Ouvrir Eclipse
 - Choisir le menu *Help*
 - Choisir le menu *Eclipse Marketplace*
 - Choisir *GNU MCU Eclipse*
Selectionner seulement
☒ GNU MCU C/C++ J-Link Debugging

2 Installation sur une machine personnelle Windows 10

- Installer Git
 - Double-click sur "Git-2.18.0-64-bit.exe"
 - Utilisez les valeurs par défaut



- Installer Meld
 - Double-click sur "Meld-3.18.2-win32.msi"
 - Installer la chaîne d'outil GCC
 - Double-click sur "gcc-arm-none-eabi-7-2018-q2-update-win32-sha2.exe"
 - Ajouter le chemin aux variables d'environnement
 - ☒ *Add path to environment variable*
 - Installer l'outil des Makefile
 - Double-click sur "make-3.81.exe"
 - Ajouter le chemin "C:\Program Files (x86)\GnuWin32\bin" au "PATH" dans les variables d'environnement système Windows
 - Installer la chaîne d'outil JLink
 - Double-click sur "JLink_Windows_V634f.exe"
 - Installer Eclipse IDE pour développeurs C/C++
 - Double-click sur l'installeur "eclipse-inst-win64.exe"
 - Choisir "Eclipse IDE for C/C++ Developers"
 - Installer le plug-in C/C++ GDB Hardware Debugging
 - Ouvrir Eclipse
 - Choisir le menu *Help*
 - Choisir le menu *Eclipse Marketplace*
 - Choisir *GNU MCU Eclipse*
 - Selectionner seulement
 - ☒ GNU MCU C/C++ J-Link Debugging
- Ouvrir un terminal "PowerShell" et créer le répertoire de travail (workspace)
 - Configurer Git

```
$ git config --global user.name "<User Name>"
$ git config --global user.email "user.name@edu.hefr.ch"
```

 - * Compléter la configuration de Git en ajoutant le contenu du fichier "git.config" dans le fichier de configuration local "~/.gitconfig"
 - Créer l'espace de travail (workspace)

```
$ mkdir -p ~/workspace
```
 - Cloner le dépôt GIT contenant le support de cours

```
$ cd ~/workspace
$ git clone https://gitlab.forge.hefr.ch/se12-1819/se12.git
```
 - Cloner le dépôt pour les travaux pratiques

```
$ cd ~/workspace/se12
$ git clone -o upstream https://gitlab.forge.hefr.ch/se12-1819/tp.git
```
 - Générer la bibliothèque pour la cible

```
$ make clean all -C ~/workspace/se12/tp/bbb/source
```
 - Définir la variable d'environnement LMIBASE



- * Ajouter dans les variables d'environnement système de Window la variable

```
LMIBASE=~\workspace\se12\tp
```

3 Installation sur une machine personnelle Mac OS X

- Installer Git
 - Double-click sur "git-2.18.0-intel-universal-mavericks.dmg"
- Installer Meld
 - Double-click sur "meldmerge.dmg"
- Installer la chaîne d'outil GCC

```
$ mkdir -p ~/opt/gcc-arm
```

```
$ tar xvf gcc-arm-none-eabi-7-2018-q2-update-mac.tar.bz2 -C ~/opt/gcc-arm --strip 1"
```
- Ajouter le chemin dans la variable PATH en éditant le fichier ~/.bash_profile

```
export PATH=$PATH:$HOME/opt/gcc-arm/bin
```
- Installer la chaîne d'outil JLink
 - Double-click sur "JLink_MacOSX_V634f.pkg"
- Installer Eclipse IDE pour développeurs C/C++
 - Double-click sur l'installeur "Eclipse Installer"
 - Choisir "Eclipse IDE for C/C++ Developers"
- Installer le plug-in C/C++ GDB Hardware Debugging
 - Ouvrir Eclipse
 - Choisir le menu *Help*
 - Choisir le menu *Eclipse Marketplace*
 - Choisir *GNU MCU Eclipse*
Selectionner seulement
 - ☒ GNU MCU C/C++ J-Link Debugging
- Ouvrir un terminal et créer le répertoire de travail (workspace)
 - Configurer Git

```
$ git config --global user.name "<User Name>"
```

```
$ git config --global user.email "user.name@edu.hefr.ch"
```

 - * Compléter la configuration de Git en ajoutant le contenu du fichier "git.config" dans le fichier de configuration local "~/.gitconfig"
- Créer l'espace de travail (workspace)

```
$ mkdir -p ~/workspace
```
- Cloner le dépôt GIT contenant le support de cours

```
$ cd ~/workspace
```

```
$ git clone https://gitlab.forge.hefr.ch/se12-1819/se12.git
```
- Clone le dépôt pour les travaux pratiques

```
$ cd ~/workspace/se12
```



```
$ git clone -o upstream https://gitlab.forge.hefr.ch/se12-1819/tp.git
```

- Générer la bibliothèque pour la cible

```
$ make clean all -C ~/workspace/se12/tp/bbb/source
```

- Définir la variable d'environnement LMIBASE

- * Ajouter dans les variables d'environnement système de Window la variable

```
LMIBASE=~/workspace/se12/tp
```

- * Ajouter le chemin dans la variable PATH en éditant le fichier ~/.bash_profile"

```
export LMIBASE=$HOME/workspace/se12/tp
```

- * Ajouter la variable LMIBASE dans Eclipse

- Ouvrir Eclipse
- Aller dans "Preferences..."
- Aller sous "C/C++ -> Build -> Environment"
- Ajouter la variable LMIBASE

```
LMIBASE=$HOME/workspace/se12/tp
```