

# TP 1 Prise en main des outils de développement

Le “*Hello World*” de l’embarqué est un programme qui fait clignoter une LED. On l’appelle *blinky*. Dans ce TP nous allons nous familiariser avec la compilation, le chargement et la modification d’un programme fourni qui fait clignoter les 8 LEDs de notre carte.

## 1 La carte STM32F3-Discovery

1. Branchez la carte au port USB “ST-Link”. Ouvrez la documentation de la carte<sup>1</sup>. Identifiez les différents composants de la carte.
2. Chaque patte du MCU porte un nom, du type PA3 ou PE15. Cherchez dans la documentation :
  - à quelles pattes sont connectées les LEDs
  - à quelle patte est connecté le bouton USER
3. À quoi sert le cavalier JP3 (Idd) ?

## 2 Compilation et téléchargement

1. Téléchargez le fichier .zip associé à ce TP ; il contient les bibliothèques (`lib/`), les scripts de compilation (`makefile.inc`) et le code fourni pour cette séance (`1_blinky/`). *Attention* : sauvegardez son contenu dans un endroit dont vous vous souviendrez. Pour les prochaines séances, nous vous fournirons uniquement le code, à placer dans ce répertoire, à côté de `1_blinky/`.
2. Exécutez `make` dans le répertoire `1_blinky/` pour lancer la compilation du projet. Observez attentivement chaque commande déclenchée. Quels nouveaux fichiers ont été générés par la compilation ? Quels fichiers source prennent part à la création du binaire final ?
3. Exécutez `make flash` pour télécharger le code sur la carte. Les LEDs devraient se mettre à clignoter.
4. Modifiez le code source de façon à augmenter la fréquence de clignotement.

## 3 Un pilote de LED rudimentaire

1. Créez une nouvelle paire de fichiers `leds.c/leds.h` qui isolent les instructions d’initialisation et de changement d’état des LEDs dans des fonctions dédiées. Appelez ces fonctions dans `main.c` ; testez le résultat.
2. Compilez le fichier `main.c` à la main en un fichier assembleur `main.S` à l’aide de l’option `-S` du compilateur `arm-none-eabi-gcc`<sup>2</sup>. Pouvez-vous identifier les instructions correspondant à la boucle principale ? aux appels de fonction ?

---

1. Googlez “stm32f3-discovery user manual” et ouvrez le premier résultat (un PDF).

2. `man gcc` est votre ami.