

C avancé

6 - Projet

6.1 - Projet

4 équipes en compétition

Dans un contexte RTOS

Programmation en C

Sur carte STM32F4 Discovery

Sur FreeRTOS à travers CMSIS RTOS v2

Conditions réelles : à vous de découvrir la documentation

6.2 - CMSIS RTOS

Abstraction de RTOS

Portabilité du code applicatif sur plusieurs OS

https://arm-software.github.io/CMSIS_5/RTOS2/html/rtos_api2.html

6.3 - STM32F4 discovery

6.3.1 - STM32F4 discovery

Carte basée sur un ARM Cortex-M4, à 168 MHz

Architecture ARMv7-M

1 Mio de mémoire Flash

192 Kio de RAM

USB ST-LINK/V2

6.3.2 - STM32F4 discovery

Dit autrement :

Un processeur à 0,168 GHz

0,000 192 Go de RAM

0,000 001 To de disque dur (Flash)

Bienvenu dans le monde de l'embarqué

6.3.3 - STM32F4 discovery

USB OTG

Accéléromètre 3 axes

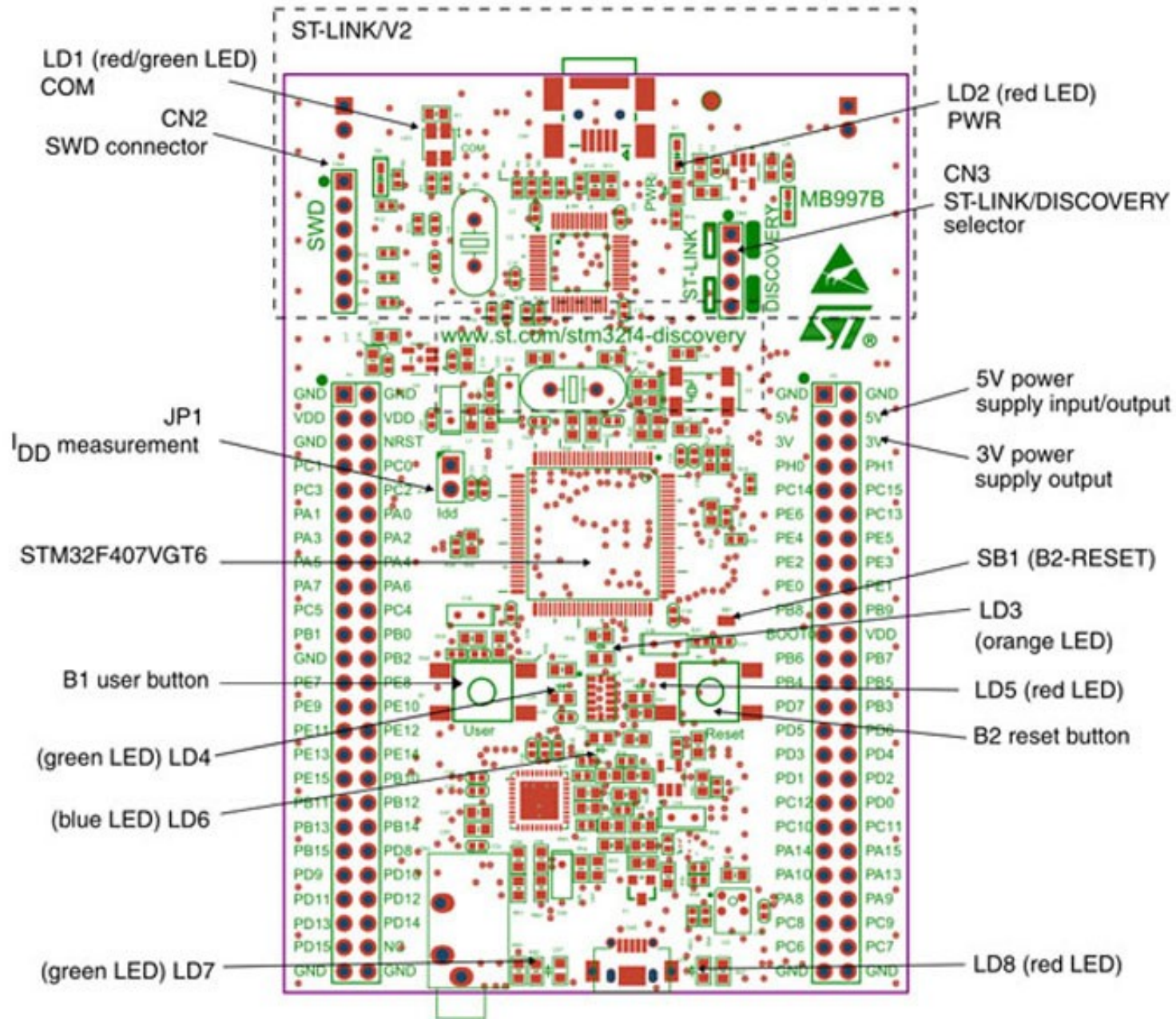
Sortie casque audio

4 LEDs

Un bouton poussoir

Des connecteurs pour les extensions

6.3.4 - STM32F4 discovery



6.3.5 - STM32F4 discovery shield

Carte porteuse

Convertisseur RS232 <-> USB

4 connecteurs Mikro BUS

Mesure météo

GPS

Rythme cardiaque

Mémoire flash

WiFi

Ethernet

Afficheur 7 segments

...

<https://www.mikroe.com/click>

6.3.6 - Outillage

IDE

Keil

AC6 System Workbench

Atollic TrueSTUDIO

STM32CubeIDE

...

Ces outils fournissent le compilateur, le débogueur, l'éditeur de code et bien plus

Nous allons utiliser PlatformIO dans Visual Studio Code

L'essentiel : STM32F4 discovery

Carte ARM Cortex-M4 à 168MHz, 1 Mio de Flash, 192 Kio de RAM

USB OTG, accéléromètre, LEDs, bouton poussoir...

STM32F4 discovery shield : carte porteuse, extensible grâce aux connecteurs Mikro BUS

6.4 - Rapport

6.4.1 - Rapport

Vous me rendrez un rapport

Allez droit au but, pas de blabla

Je suis plus intéressé par le fond que par la forme

6.4.2 - Rendu

Rendu sur
https://quiz.lecalamar.fr/quiz/upload/space_collector/

6.4.3 - Pièces jointes

Tout ce que vous avez créé ou modifié

Mais rien de ce qui vous a été fourni ou
généré

6.4.4 - Rapport

Répartition des tâches dans le groupe

6.4.5 - Rapport

Explication de votre démarche au long du projet

Étape par étape

Avec captures d'écran

6.4.6 - Rapport

Explication des **3** plus grandes difficultés

Et comment elles ont été surmontées

6.4.7 - Rapport

Liste des notions apprises / découvertes /
peaufinées pendant le projet

6.4.8 - Rapport

Un petit retour d'expérience général sur ce projet

Ce qui était améliorable

Ce qui était bien

Ce qui était original

6.5 - Et ce projet ?

Collectez vos planètes avec vos vaisseaux

Pour les ramener dans votre base

Attaquez les vaisseaux ennemis

Plus le code est performant, plus il peut
envoyer d'ordres

6.6 - Et ce projet ?

Un thread par vaisseau

Communication par liaison série

Ressource partagée par les threads

Convertisseur fourni de liaison série vers TCP/IP
pour le serveur de jeu

Template de code :

https://github.com/vpoulailleau/space_collector/raw/main/space_collector.zip

6.7 - Et ce projet ?

https://github.com/vpoulailleau/space_collector

L'essentiel : Projet

§ STM32F4 discovery

§ Rapport

https://github.com/vpoulailleau/space_collector

7 - PlatformIO

7.1 - PlatformIO

Extension de Visual Studio Code

Pour la programmation en C/C++

En embarqué (principalement)

Sur PC

Ensemble d'outils

Compilateur, débbugger...

7.2 - Installation

Installez VS Code

<https://code.visualstudio.com/Download>

Installez PlatformIO

Dans le menu extension de VS Code

`platformio.platformio-ide`

7.3 - Installation des plateformes

Dans PlatformIO: Home (icône de maison)

Dans Platforms

Installez

Embedded > ST STM32

Desktop > Windows x86

7.4 - Création d'un projet C embarqué

Dans PlatformIO: Home

Dans Projects

Create New Project

Name : en minuscules sans espace

Board : ST STM32F4DISCOVERY

Framework : STM32Cube

Use default location : coché

7.5 - Création d'un projet C embarqué

Dans `platformio.ini`, ajoutez à la fin

`platformio.ini`

```
monitor_speed = 115200
```

```
build_flags = -O0 -Wall -g3
```

```
debug_build_flags = -O0 -Wall -g3
```

7.6 - Création d'un projet C sur PC

Créez un projet C embarqué

Puis remplacez le contenu de `platformio.ini` par

`platformio.ini`

```
[env:windows_x86]
```

```
platform = windows_x86
```

```
build_flags = -O0 -Wall -g3
```

```
debug_build_flags = -O0 -Wall -g3
```