

# README – Projet HIL STM32 & Simulink

Moussa TRAORE /mtraor@insa-toulouse.fr,moussareotra021@gmail.com

August 18, 2025

## Objectif

Plateforme **Hardware-in-the-Loop (HIL)** avec deux cartes STM32 et Simulink :

- **Interface (STM32L476)** : lien principal avec Simulink (UART2), lien avec le Contrôleur (UART3), **miroir de la consigne sur DAC**, **lecture de la PWM filtrée** via ADC **synchro sur front montant** de la PWM (Input Capture), renvoi de la commande vers Simulink.
- **Contrôleur (STM32F103)** : reçoit la **consigne** (UART3 et/ou mesure analogique du DAC Interface), calcule le **PID**, génère la **PWM** vers l'Interface.

## Arborescence livrée

/Interface&Controleur.zip	# Projets STM32CubeIDE : Interface (L476) + Contrôleur (F103)
/Matlab stage.zip	# stage1.m (script) + controleur.slx (modèle Simulink)
/README.tex	# Ce fichier LaTeX (compiler en PDF si besoin)

## Versions utilisées (à compléter)

- STM32CubeIDE : version (1.19.0)
- MATLAB/Simulink : R2020a
- Cartes : NUCLEO-L476RG (Interface) & NUCLEO-F103RB (Contrôleur)
- UART : 115200 bauds, 8N1

## Flux de données (résumé)

1. **Simulink** → **Interface (UART2)** : envoie **consigne** et **measure\_plant** (float32).
2. **Interface** :
  - Sort la **mesure** sur **DAC (PA4)** ⇒ **ADC du Contrôleur** (référence analogique/validation HIL).
  - Transmet aussi la **consigne en numérique** au **Contrôleur (UART3)**.
3. **Contrôleur** : calcule **PID(consigne, mesure)** et génère la **PWM (TIM2\_CH1)**.
4. **Retour Contrôleur** → **Interface** :
  - **PWM brute** → **PA6 (TIM3\_CH1)** pour **synchro front montant** (Input Capture).
  - **PWM filtrée (RC)** → **PC1 (ADC1\_IN2)** pour mesure analogique moyenne.
5. **Interface** → **Simulink (UART2)** : renvoie la **commande** (float32) issue de l'ADC synchronisé.

## Branchements & broches (conformes aux schémas)

### Interface – STM32L476RG

Fonction	Broche	Détails
<b>UART2 (Simulink)</b> <b>UART3 (vers Contrôleur)</b>	PA2 (TX), PA3 (RX) <b>PC4 (TX), PC5 (RX)</b>	Lien PC/Simulink TX->RX(contrôleur), Rx->Tx(vcontrôleur)
<b>DAC1_OUT1 (consigne miroir)</b>	<b>PA4</b>	Vers ADC du Contrôleur (consigne analogique)
<b>TIM3_CH1 (Input Capture)</b>	<b>PA6</b>	Entrée <b>PWM brute</b> du Contrôleur (front montant)
<b>ADC1_IN2 (mesure PWM filtrée)</b>	<b>PC1</b>	Après filtre RC de la PWM
LED utilisateur	PA5	Debug (toggle)

### Contrôleur – STM32F103RB

Fonction	Broche	Détails
<b>UART3 (avec Interface)</b>	<b>PB10 (TX), PB11 (RX)</b>	D'après pinout F103 fourni
<b>PWM sortie</b>	<b>PA15(TIM2_CH1)</b>	Va vers PA6 (brut) & vers filtre RC $\Rightarrow$ PC1
<b>Entrée ADC (consigne venant du DAC Interface)</b>	<b>PA0 (ADC1_IN0)</b>	PA0 connecter à <b>PA4 Interface</b> .

**Attention** \*Il faudra lier 3.3V du controleur et l'interface et leur masse .

## Filtre analogique (PWM $\rightarrow$ tension moyenne)

Filtre passe-bas RC entre la **PWM (PA0 Contrôleur)** et l'entrée **PC1 (ADC Interface)** :

$$f_c = \frac{1}{2\pi RC}, \quad \text{avec } f_c \ll f_{\text{PWM}} \text{ (ex. } f_{\text{PWM}} \in [1, 5] \text{ kHz)}$$

Exemple :  $R = 10 \text{ k}\Omega$ ,  $C = 10 \text{ nF} \Rightarrow f_c \approx 1.6 \text{ kHz}$ .

## Configuration logicielle (extraits)

### Interface (L476)

- **ADC1** sur **PC1/IN2**, déclenché **logiciel** dans le callback TIM3 (Input Capture).
- **TIM3\_CH1 (PA6)** en **Input Capture front montant**, horloge interne, **NVIC TIM3** activée.
- **DAC1\_OUT1 (PA4)** actif pour miroiter la consigne vers le Contrôleur.
- **UART2 = Simulink**, **UART3 = Contrôleur**.

### Contrôleur (F103)

- **TIM2\_CH1 (PA0)** en PWM (ex. 1 kHz: PSC=79, ARR=999).
- **ADC1** sur **PA4/IN4** (recommandé) pour lire la **consigne analogique** issue du DAC de l'Interface (si utilisée).

- UART3 = lien numérique consigne avec l'Interface.
- PID discret avec **anti-windup** (éviter l'intégrale lorsque la sortie sature).

## Protocole d'échange

- **Simulink** → **Interface (UART2)** : 8 octets = [consigne(float32), mesure\_plant(float32)]
- **Interface** → **Contrôleur (UART3)** : 4 octets = consigne(float32)
- **Interface** → **Contrôleur (analogique)** : consigne miroir via **DAC PA4** → **ADC Contrôleur** (PA4/IN4 recommandé)
- **Contrôleur** → **Interface (analogique)** : **PWM PA0** → RC → **PC1 (ADC Interface)** ; **PA6** reçoit la PWM *brute* pour la **synchro front montant**.
- **Interface** → **Simulink (UART2)** : 4 octets = commande(float32) lue par ADC synchronisé.

Encodage binaire float32 *little-endian* (Pack/Unpack Simulink).

## Mise en route

### STM32 :

1. Décompresser **Interface&Contrôleur.zip**, ouvrir les deux projets dans STM32CubeIDE.
2. Flasher d'abord **Interface** , puis **Contrôleur(STAGELASS)**.
3. Vérifier les broches dans les fichiers **.ioc** (UART2, UART3, ADC, TIM).

### MATLAB/Simulink :

1. Décompresser Matlab **stage.zip**.
2. Lancer **stage1.m** puis ouvrir **contrôleur.slx** et démarrer la simu.