

Cours Microcontrôleurs Avancés

CH5: Les Horloges du STM32F401RE

Par:

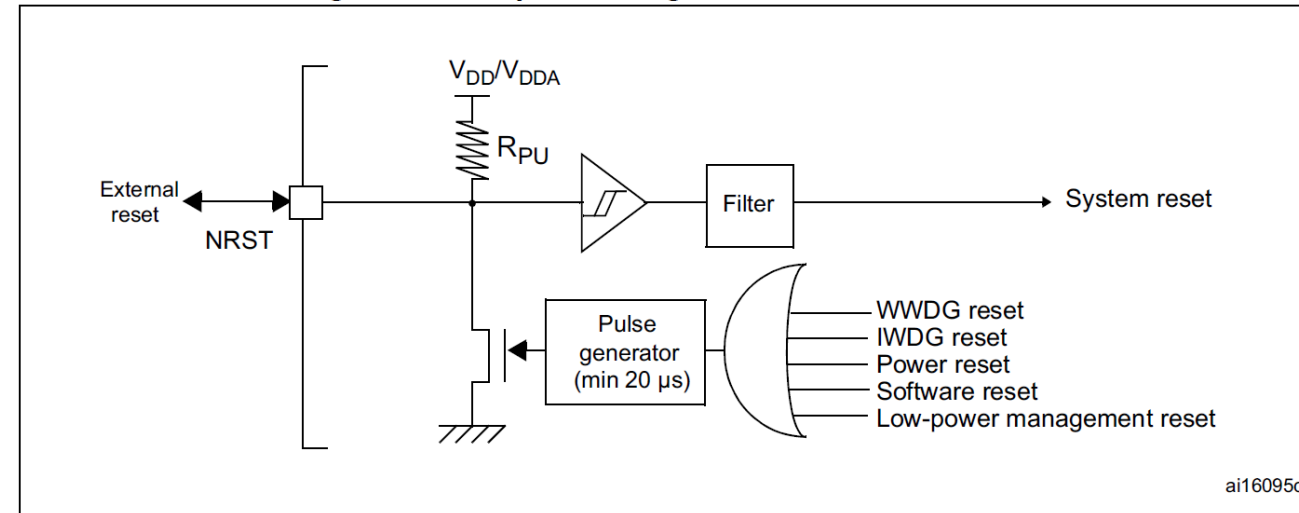
Hassan EL FADIL

elfadil.h@gmail.com

- **RCC**: Contrôle de Réinitialisation et d'Horloge: module qui gère la réinitialisation du système et la configuration de l'horloge.
- **Voici les principales fonctions:**
 - **Sélection de la Source d'Horloge** :
 - Oscillateur Interne
 - Oscillateur Externe
 - **Configuration de l'Horloge du Système**: RCC configure l'Horloge du Système (SYSCLK), qui est la principale source d'horloge pour le CPU et les périphériques.
 - **Système de Sécurité d'Horloge (CSS)** : RCC inclut un Système de Sécurité d'Horloge qui peut détecter si la source d'horloge échoue.
 - **Contrôle de Réinitialisation** : RESET
 - **Contrôle de l'Horloge des Périphériques** : RCC active et désactive les horloges de divers périphériques. Cela est crucial pour la gestion de l'énergie.
 - **Modes de Basse Consommation** : Dans les modes de basse consommation, la configuration de RCC vous permet de définir différentes configurations d'horloge pour optimiser l'utilisation de l'énergie.

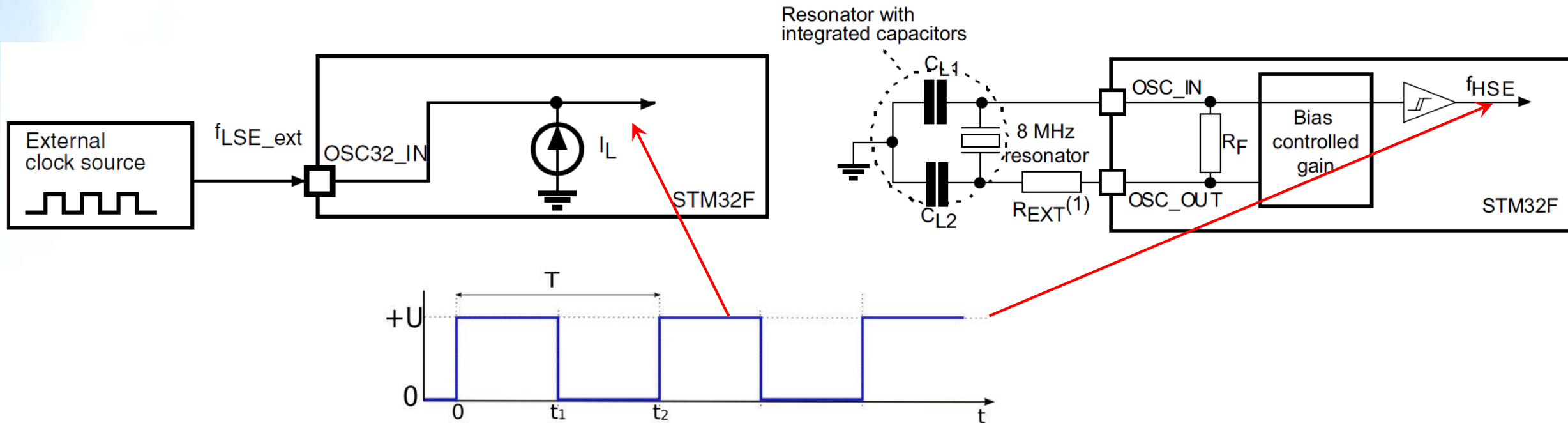
Le RCC gère différents types de réinitialisations:

- **Réinitialisation au Démarrage (POR: Power-On Reset)** : Se produit lorsque le dispositif est alimenté.
- **Réinitialisation par Défaut (BOR: Brown-Out Reset)** : Se produit lorsque la tension d'alimentation tombe en dessous d'un certain seuil.
- **Réinitialisation Externe (NRST: External Reset)** : Une broche externe peut déclencher une réinitialisation.
- **Réinitialisation par le Minuterie de Surveillance Indépendante (IWDG: Independent Watchdog Timer)** : Une minuterie interne qui déclenche une réinitialisation si elle n'est pas rafraîchie dans un certain délai.
- **Réinitialisation par le Minuterie de Surveillance à Fenêtre (WWDG: Window Watchdog Timer)** : Une autre minuterie de surveillance qui déclenche une réinitialisation si elle n'est pas rafraîchie dans une fenêtre de temps spécifiée.



Les Horloges: Rôle de l'Horloge dans un microcontrôleur

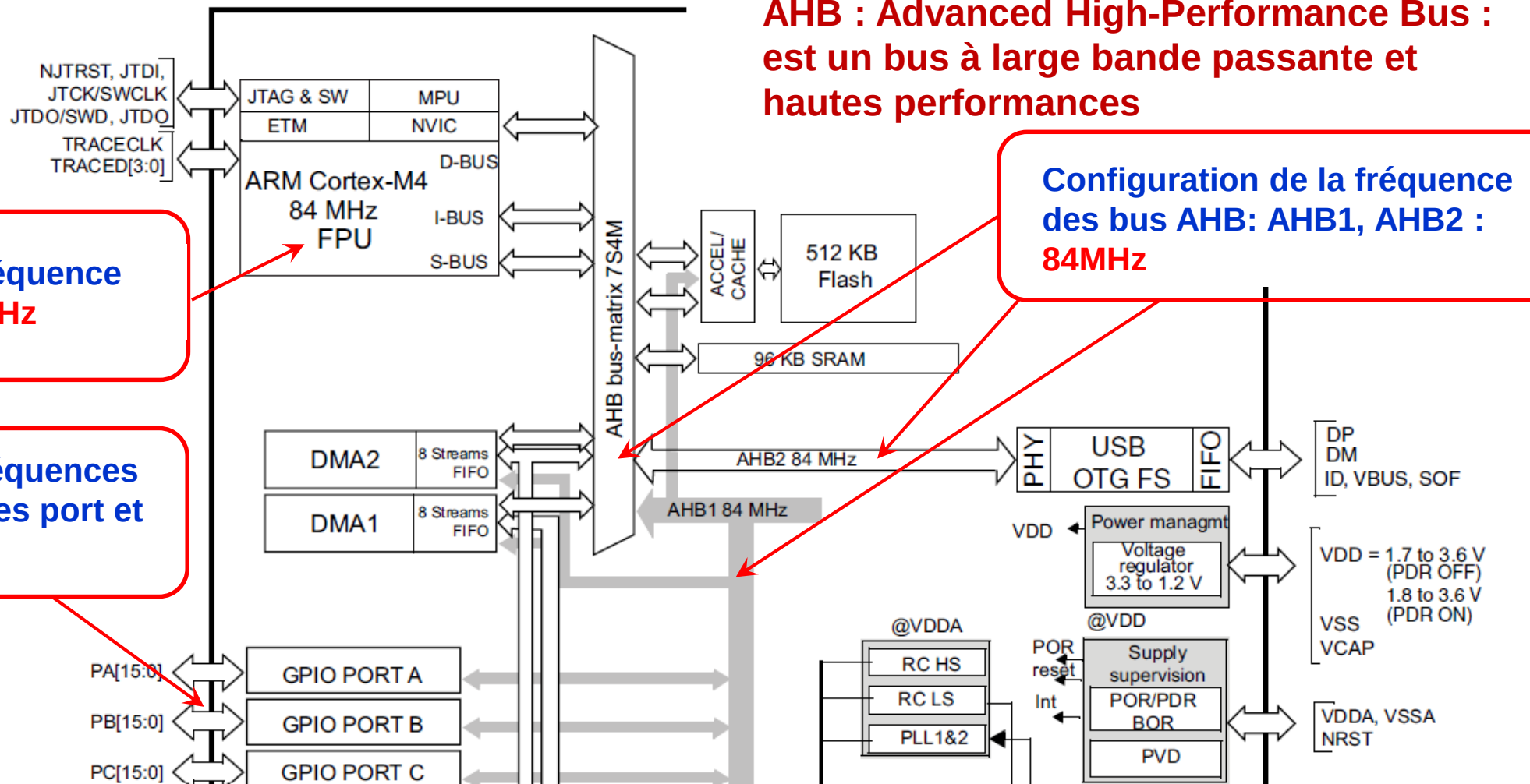
- Une horloge (**ou oscillateur**) dans un microcontrôleur est un composant crucial qui génère des signaux de synchronisation pour réguler le fonctionnement interne du microcontrôleur.
- Ces signaux de **synchronisation** sont utilisés pour **coordonner** le fonctionnement des divers composants du microcontrôleur, tels que le CPU, la mémoire, les périphériques d'entrée/sortie et d'autres modules intégrés.



Voici quelques-unes des fonctions essentielles de l'horloge dans un microcontrôleur :

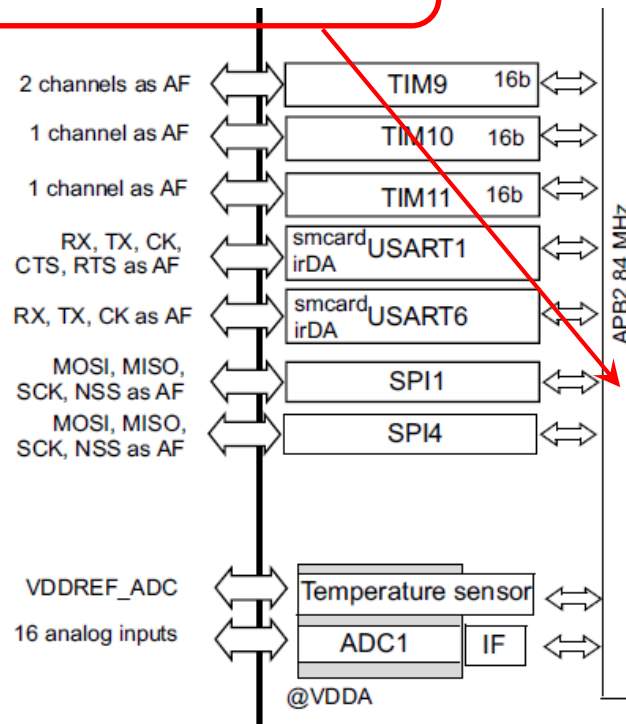
- **Synchronisation des opérations** : L'horloge assure que toutes les opérations du microcontrôleur se déroulent de manière synchronisée.
- **Contrôle de la vitesse** : L'horloge définit la vitesse à laquelle le microcontrôleur effectue ses opérations.
- **Gestion de la consommation d'énergie** : Certains microcontrôleurs peuvent ajuster la fréquence de l'horloge pour réduire la consommation d'énergie lorsqu'une performance maximale n'est pas nécessaire.
- **Synchronisation des périphériques** : Les périphériques du microcontrôleur, tels que les timers, les interfaces de communication (comme UART, SPI, I2C) et d'autres modules, utilisent également l'horloge comme base de synchronisation pour leurs opérations.
- **Timing précis** : Dans certaines applications, comme les systèmes de communication sans fil et les capteurs, un timing précis est essentiel. L'horloge garantit que les données sont échantillonnées et traitées à des moments précis, ce qui est crucial pour la fiabilité du système.

Qu'est ce qu'on configure?

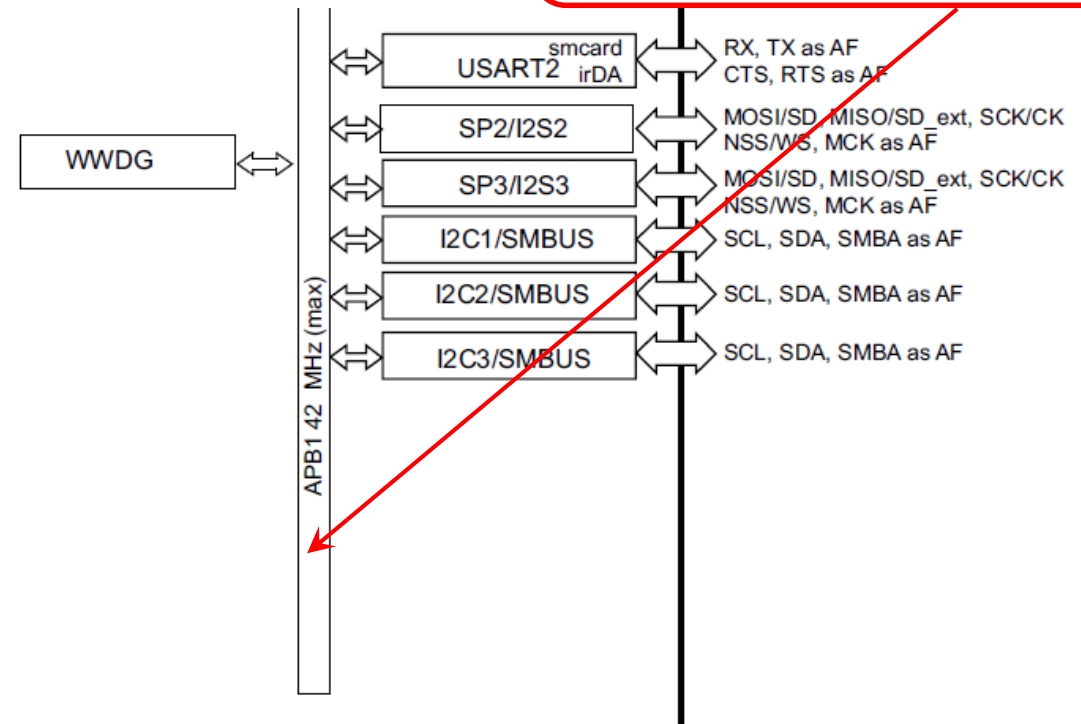


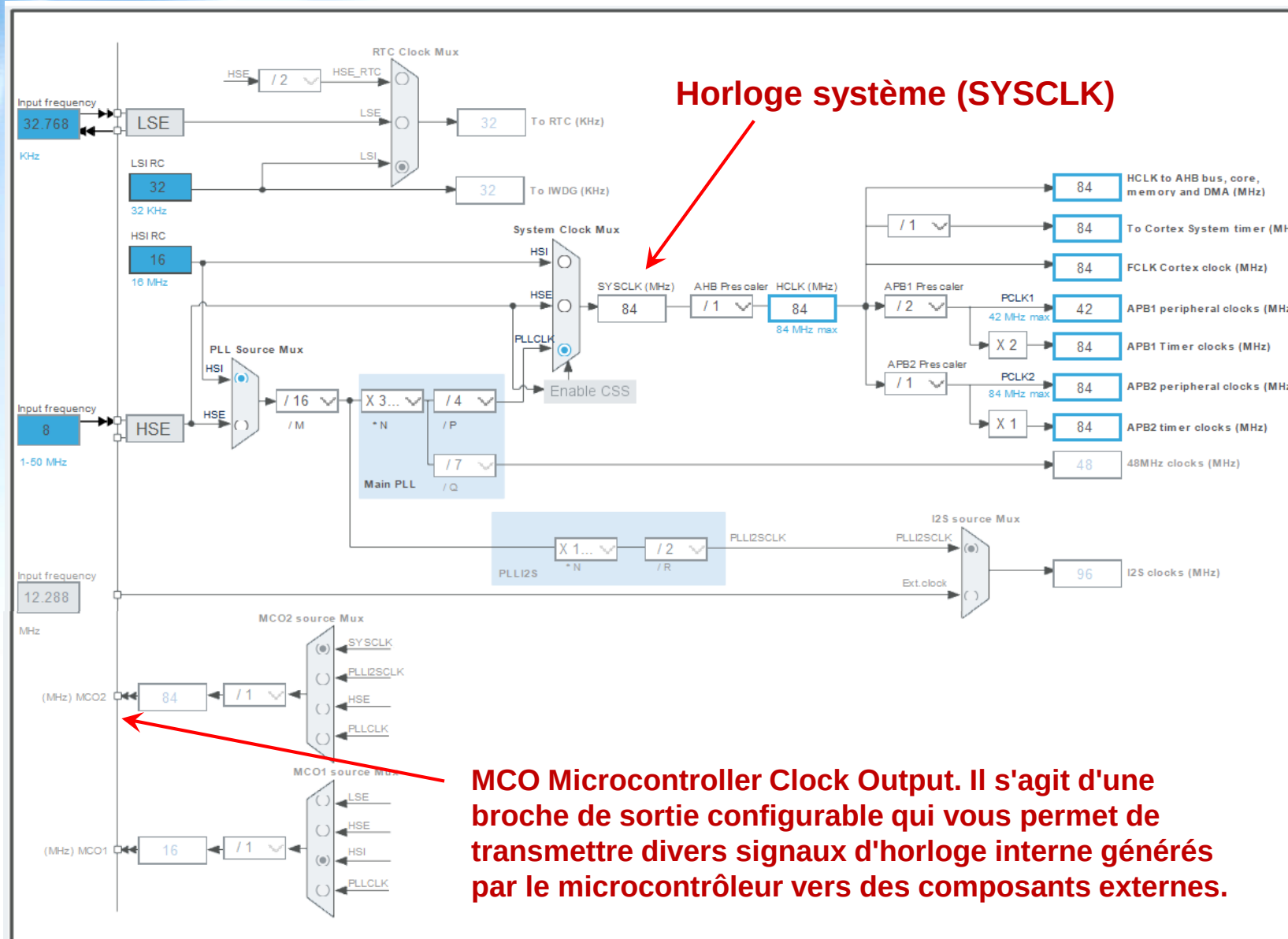
APB: Advanced Peripheral Bus est un bus périphérique à faible vitesse utilisé pour la communication entre le CPU et divers modules périphériques

**Configuration de la fréquence
du bus APB2: 84 MHz (max)**



**Configuration de la fréquence
du bus APB1: 42 MHz (max)**





Fréquence des Bus
AHB

Fréquence des Bus
APB

Fréquence des ports et
des périphériques

MCO Microcontroller Clock Output. Il s'agit d'une broche de sortie configurable qui vous permet de transmettre divers signaux d'horloge interne générés par le microcontrôleur vers des composants externes.

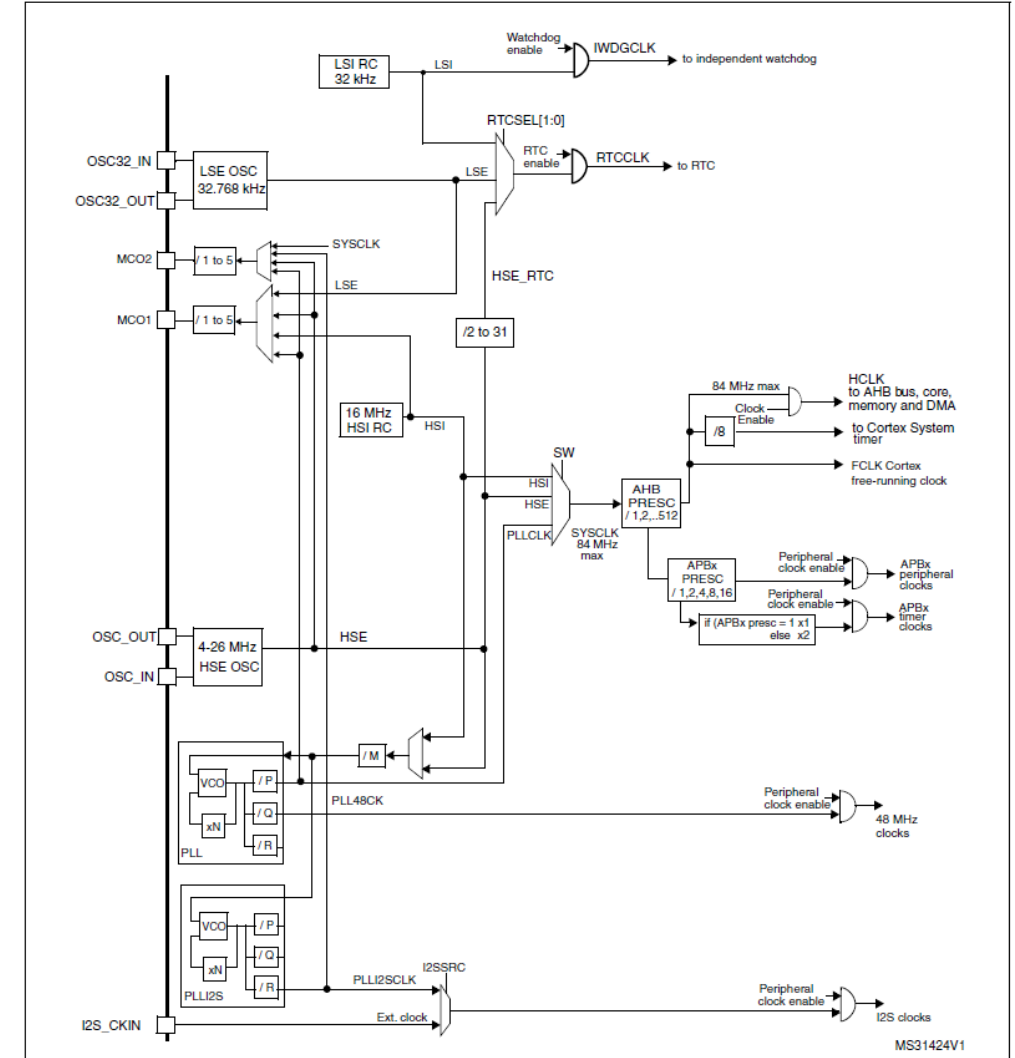
Trois sources d'horloge différentes peuvent être utilisées pour piloter l'horloge système (SYSCLK) :

- **Horloge d'oscillateur HSI**
- **Horloge d'oscillateur HSE**
- **Horloge PLL principale (PLL)**

Les appareils disposent des deux sources d'horloge secondaires suivantes :

- **RC interne basse fréquence de 32 kHz (LSI RC)** qui pilote le chien de garde indépendant et, en option, le RTC utilisé pour le réveil automatique à partir du mode Stop/Standby.
- **Quartz externe basse fréquence de 32,768 kHz (Quartz LSE)** qui pilote en option le RTC horloge (RTCCLK)

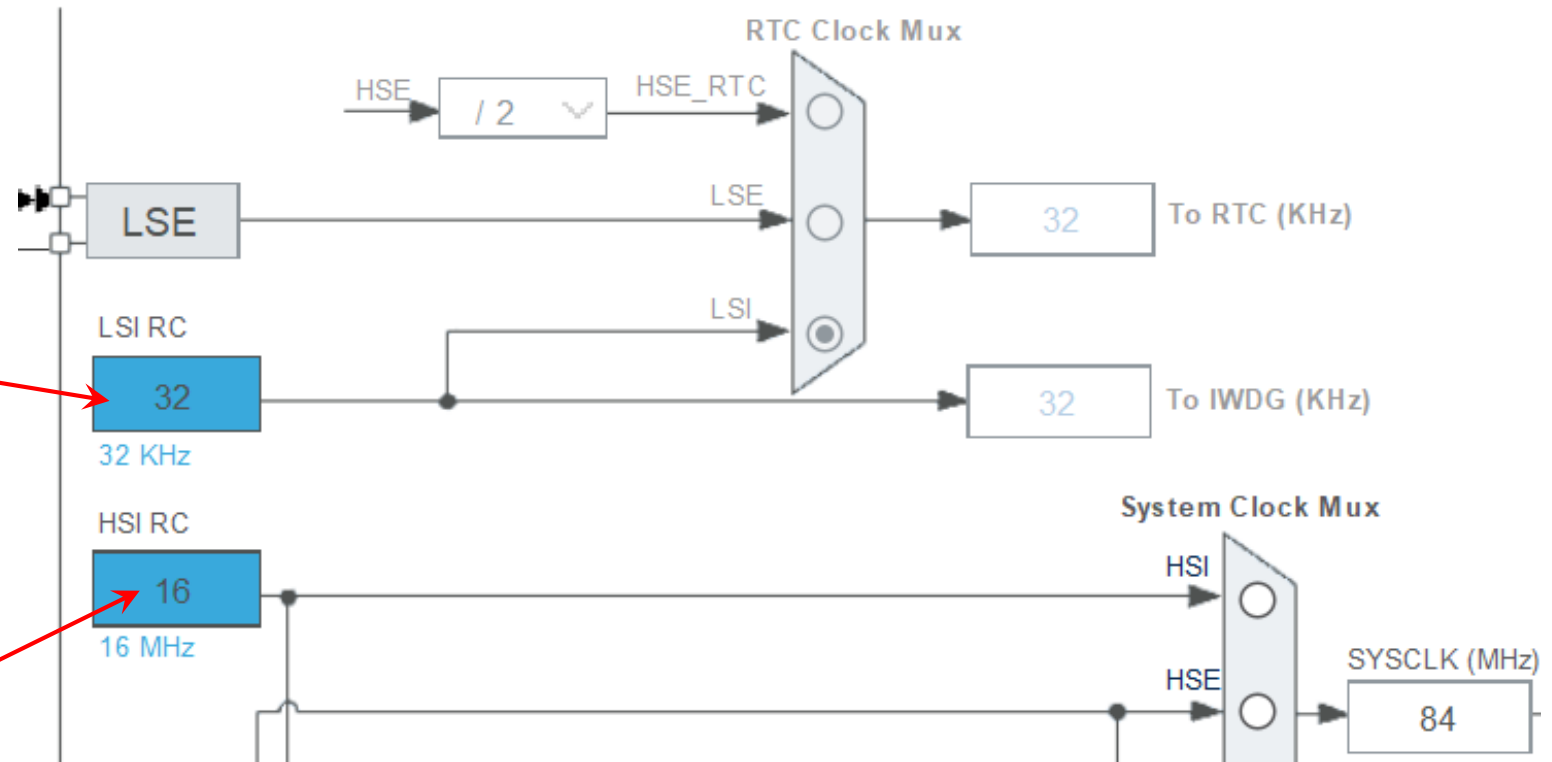
Figure 12. Clock tree



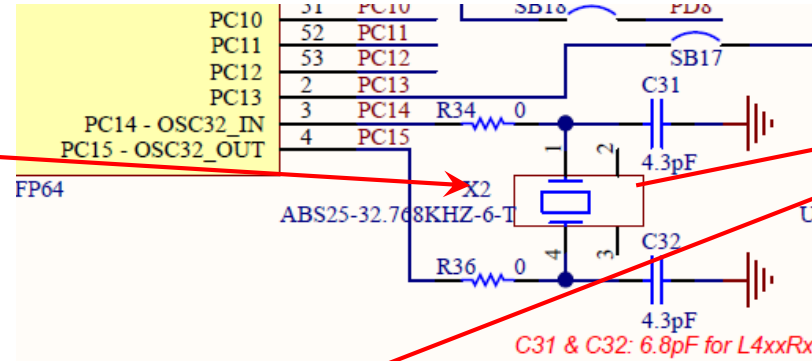
- **LSI** : Low Speed Internal clock (RC oscillator)
- **HSI** : High-Speed Internal clock (RC oscillator)

Fréquence du LSI : 32 MHz

Fréquence du HSI : 16 MHz

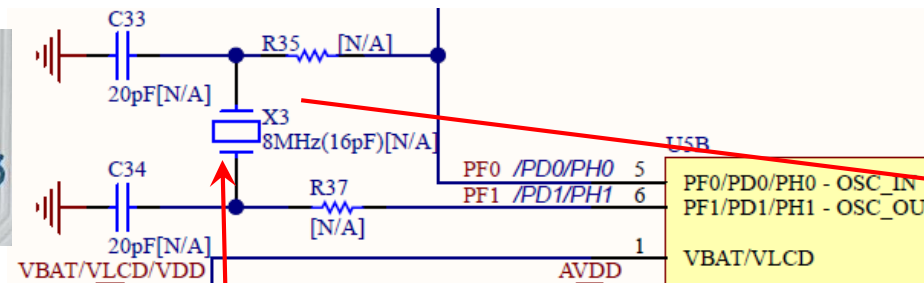


Oscillateur à
Quartz/Céramique

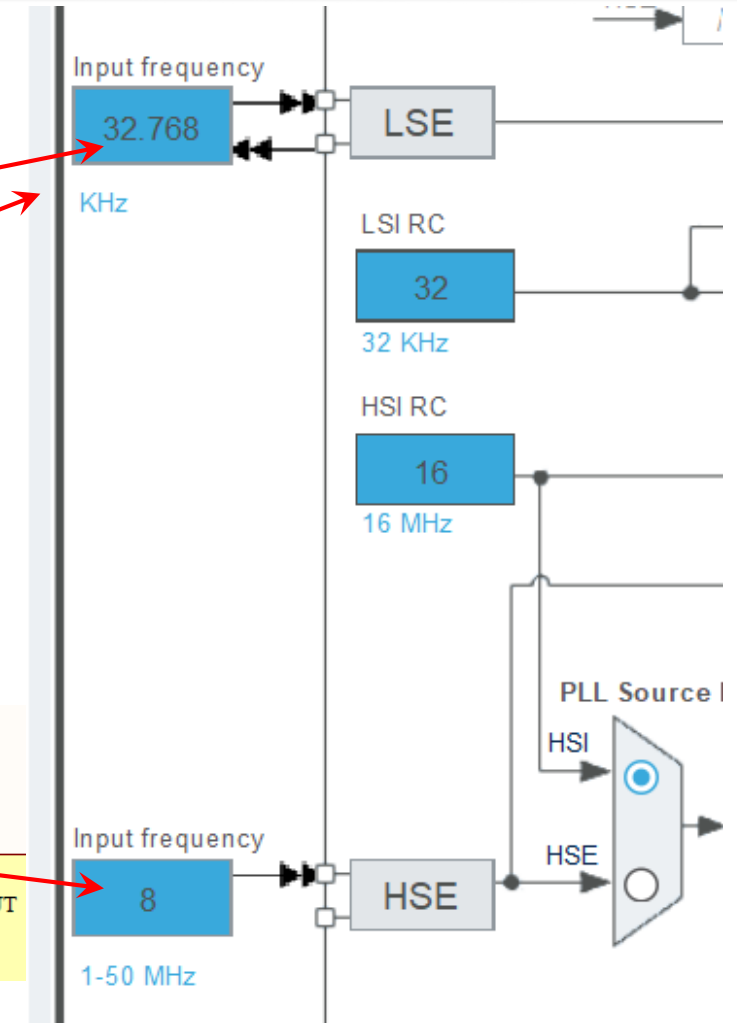


- **LSE : Low Speed External clock (32.768kHz)**
- **HSE : High-Speed External clock (Horloge utilisateur: 1 à 50MHz. Oscillateur à Quartz/Céramique : 4 à 26MHz)**

Oscillateur à
Quartz/Céramique
à souder entre ces
bornes de la carte.
Ou utiliser une
horloge externe



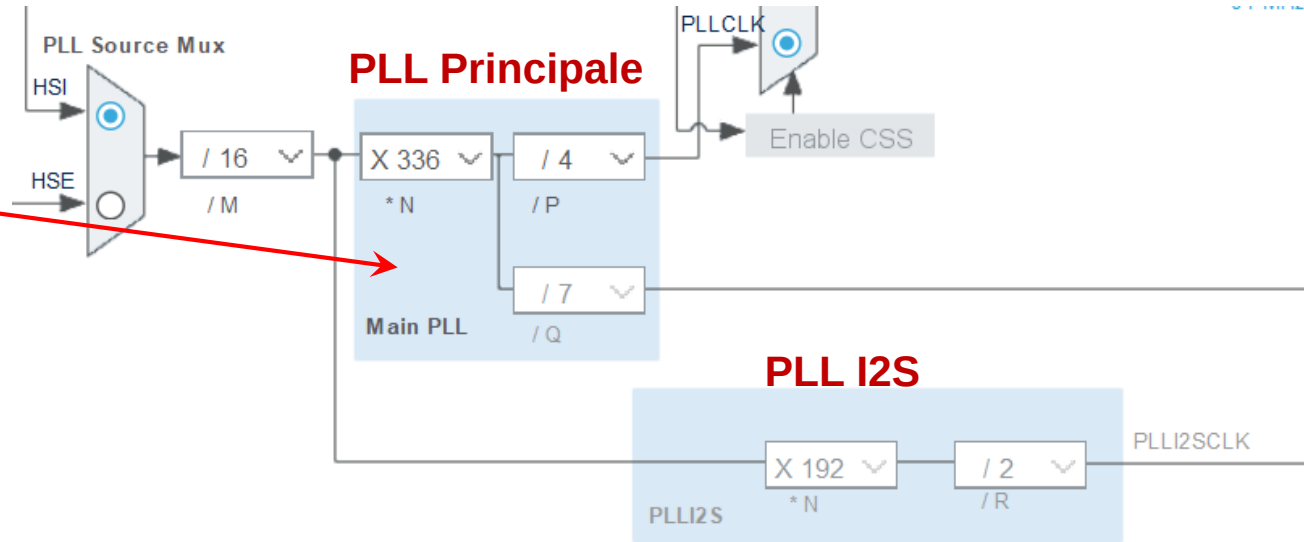
Exemple d'un oscillateur
externe à Quartz de **8MHz**



PLL : Phase-Locked Loop: Joue un rôle essentiel dans la génération de différentes fréquences internes utilisées par le microcontrôleur.

Voici les principales fonctions de la **PLL principale** :

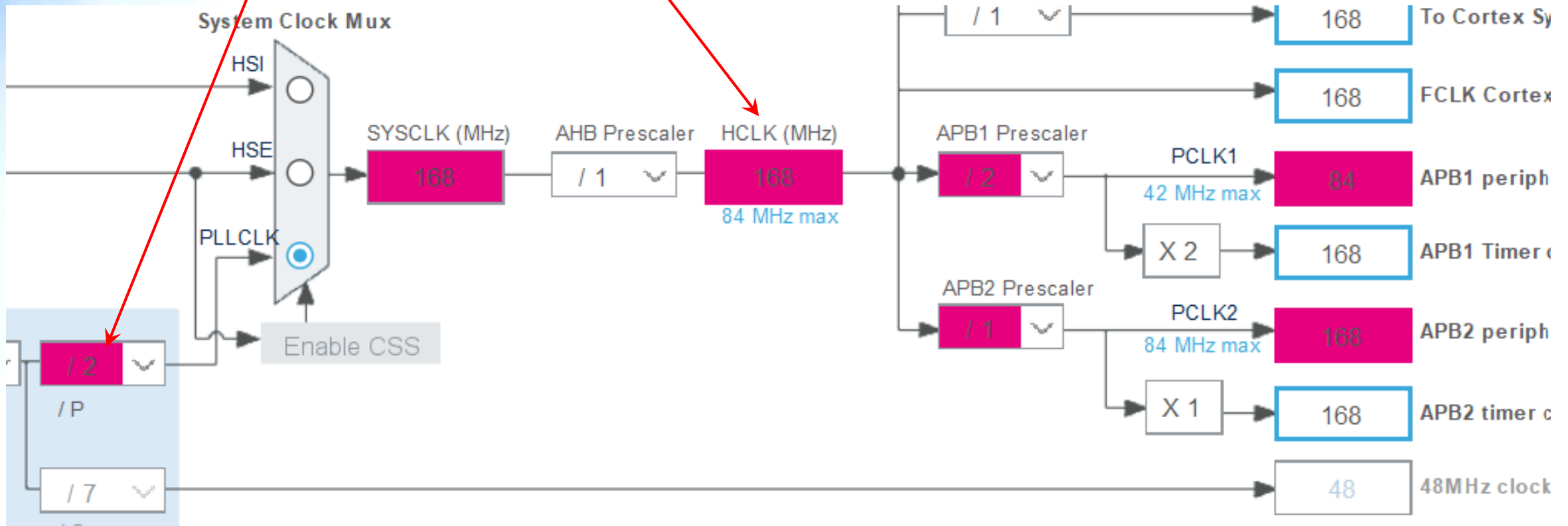
- **Stabilisation de la Fréquence**
- **Multiplier la Fréquence**: multiplier la fréquence d'entrée pour obtenir une fréquence de sortie plus élevée.
- **Fréquence de l'Horloge du Système** : La PLL est couramment utilisée pour générer la fréquence principale du système (HCLK) à partir d'une source d'horloge stable, telle qu'un oscillateur à cristal (HSE) ou un oscillateur interne (HSI). Cette fréquence est utilisée par le CPU et d'autres composants du microcontrôleur.
- **Générer des Fréquences pour les Périphériques** : Certains périphériques à l'intérieur du microcontrôleur, tels que les timers, les interfaces USART, et les bus internes, peuvent avoir leurs propres horloges. La PLL peut générer ces fréquences pour synchroniser ces périphériques.



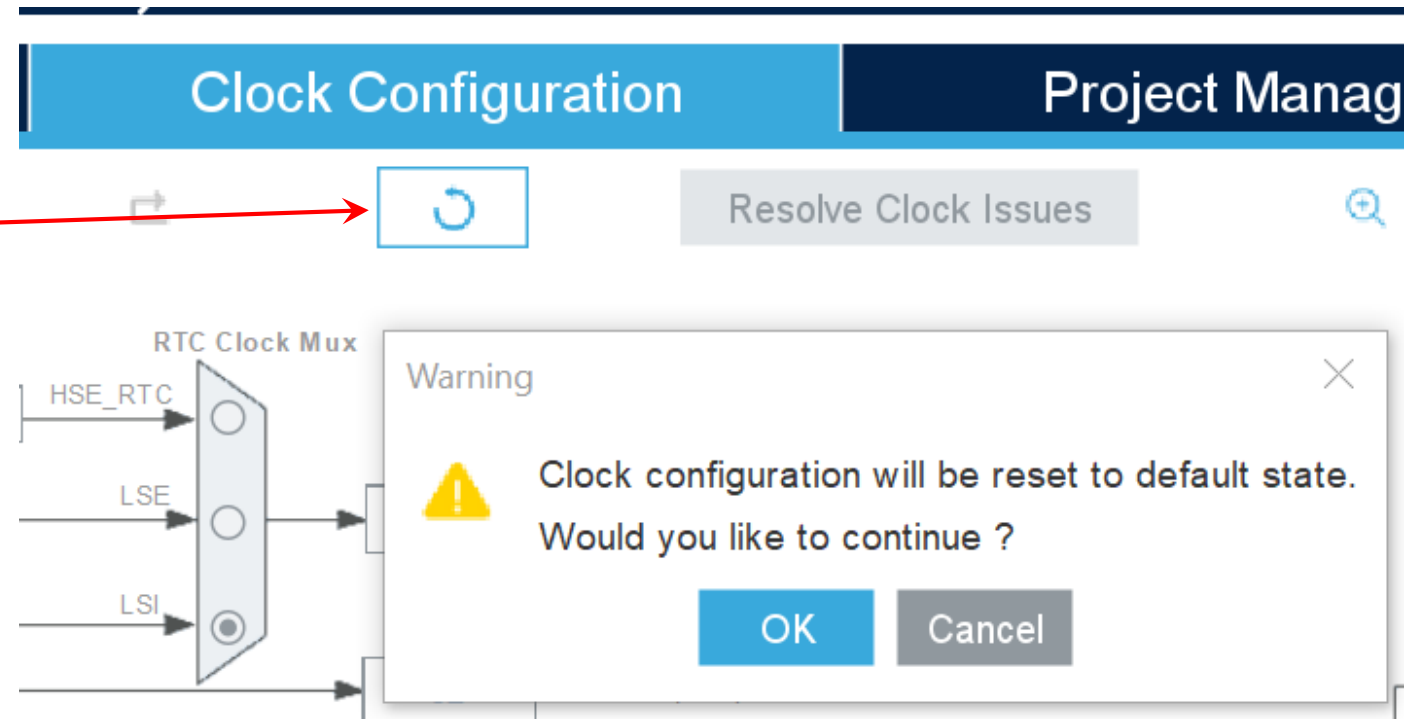
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{PLL_IN}	PLL input clock ⁽¹⁾		0.95 ⁽²⁾	1	2.10	MHz
f_{PLL_OUT}	PLL multiplier output clock		24	-	84	MHz
f_{PLL48_OUT}	48 MHz PLL multiplier output clock		-	48	75	MHz
f_{VCO_OUT}	PLL VCO output		192	-	432	MHz

PLL I2S: I2S (Inter-IC Sound) : fait référence au protocole de communication audio couramment utilisé pour la transmission des données audio numériques entre composants audio.

Remarque: Une mauvaise configuration conduit à des fréquences qui dépassent leurs limites maximales (blocs en rouge)



On peut revenir aux valeurs par défaut en cliquant sur ce bouton





La fonction de configuration
des horloges dans **main.c**

Une fois la bonne configuration
effectuée, Cliquer ensuite sur « Device
Configuration Tool Code Generation »

Ici, les valeurs de configuration
effectuée graphiquement.
Chaque changement effectué dans
l'interface graphique se traduit dans le
changement de ces valeurs.

```

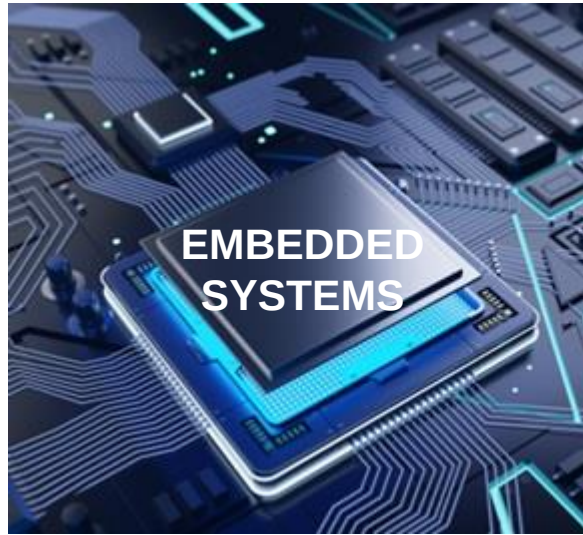
49 /* Private function prototypes -----
50 void SystemClock_Config(void);
51 static void MX_GPIO_Init(void);
52 static void MX_USART2_UART_Init(void);

```

```

138 RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
139 RCC_OscInitStruct.HSISState = RCC_HSI_ON;
140 RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
141 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
142 RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
143 RCC_OscInitStruct.PLL.PLLM = 16;
144 RCC_OscInitStruct.PLL.PLLN = 336;
145 RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV4;
146 RCC_OscInitStruct.PLL.PLLQ = 7;
147 if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
148 {
149     Error_Handler();
150 }
151
152 /** Initializes the CPU, AHB and APB buses clocks
153 */
154 RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYCLK
155                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
156 RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
157 RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
158 RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
159 RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

```



FIN !
Questions ?