



**Objectifs :** Mise en œuvre des TIMER

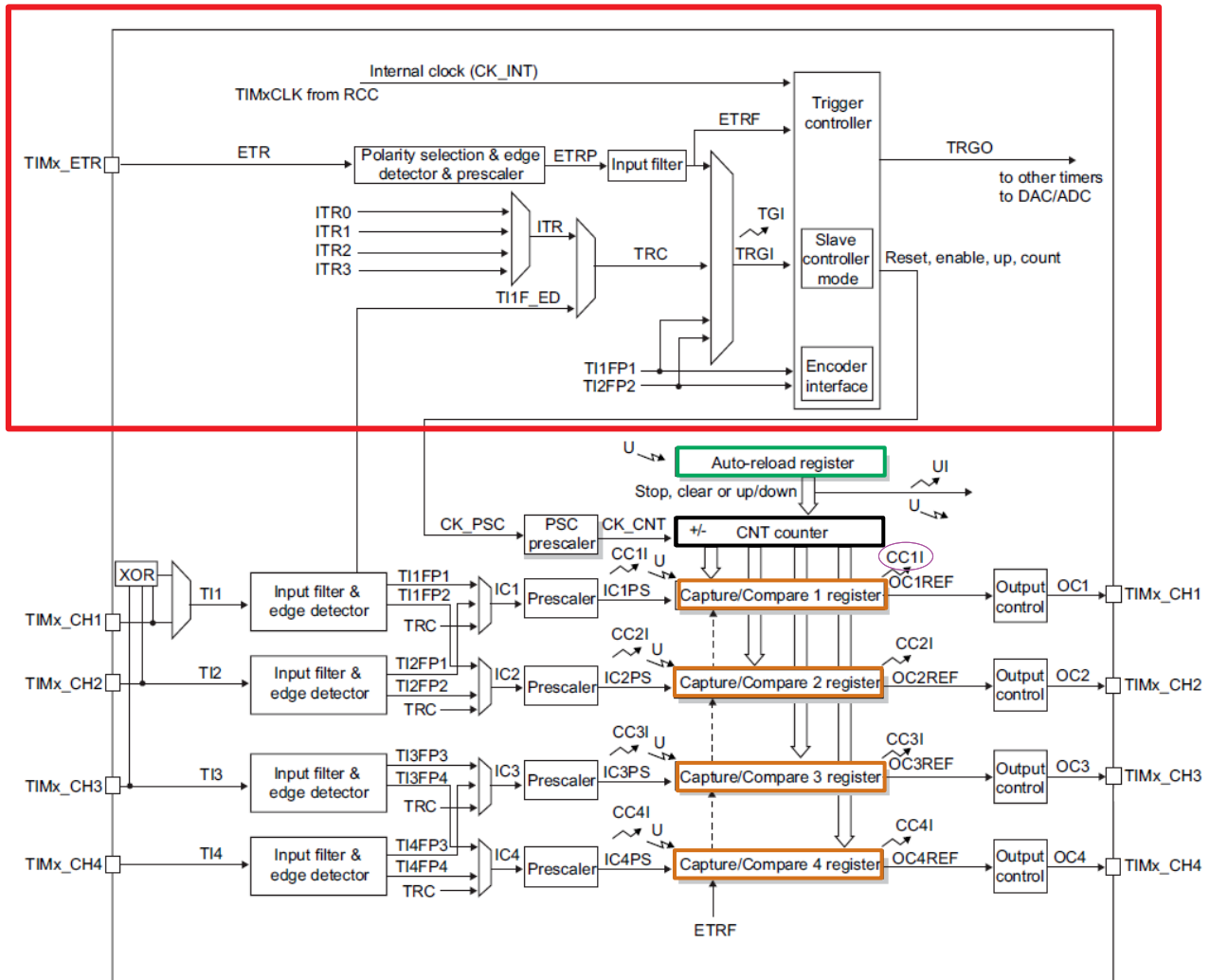
**Matériel :** Ce TP utilise une NUCLEO-F411RE, mais n'importe quelle autre carte NUCLEO convient.

**Logiciel :** MBED

Les TIMER sont un des périphériques les plus utilisés sur un microcontrôleur. Ils permettent une gestion très précise de temps.

## Fonctionnalités des TIMER

- Production de signaux rectangulaires périodiques, d'impulsion positives, négatives.
- Génération d'interruptions périodiques ou de temporisation.
- Mesure de durées, de périodes.



Le générateur d'horloge (en rouge) dispose d'une l'horloge interne ou d'une horloge externe, il produit la base de temps (CK\_PSC) du TIMER (CNT counter, en bleu).

En mode "capture" (mesure de temps) le TIMER est recopié dans un registre (Capture/compare register en orange) lors d'un événement sur une broche (TIMx\_Chx), l'événement peut être un front montant ou descendant ou les deux.

En mode "compare" (production de durée/période) le TIMER est comparé avec la valeur d'un registre (Capture/compare register), lorsqu'il y a égalité, une action est réalisée sur une broche TIMx\_Chx. La broche peut, passer à l'état haut, à l'état bas ou changer d'état (toggle). Lors de l'égalité le TIMER peut être automatiquement rechargé avec le contenu du registre Auto-reload register (en vert).

**Les différents événements détectés peuvent déclencher des interruptions. (CcxI, UI)**



## 1 Mesurer le temps, classe Timer

```
#include "mbed.h"

Timer t;

int main() {
    printf("%c[2J",27); //efface écran
    printf("%c[0;0H",27); // curseur en 0,0
    t.start();
    wait(0.01);
    t.stop();
    printf("Le TIMER t a fonctionne %f secondes\n\r", t.read());
    printf("Le TIMER t a fonctionne %d millisecondes\n\r", t.read_ms());
    printf("Le TIMER t a fonctionne %d microsecondes\n\r", t.read_us());
    t.start();
    wait(0.1);
    t.stop();
    printf("Le TIMER t a fonctionne %f secondes\n\r", t.read());
    t.reset();
    t.start();
    wait(0.1);
    t.stop();
    printf("Le TIMER t a fonctionne %f secondes\n\r", t.read());
}
```

### Méthodes de la classe Timer :

start : démarre un TIMER  
 stop : l'arrête  
 reset : le redémarre  
 read : lecture en seconde  
 read\_ms : lecture en millisecondes entières  
 read\_us : lecture en microsecondes entières

La classe Timer compte des microsecondes sur des nombres signés de 32bits soit un maximum de  $2^{31}-1$  microsecondes (environ 30mn)

## 2 Déclencher une interruption temporisée, classe Timeout

La documentation de la classe Timeout est ici:

<https://docs.mbed.com/docs/mbed-os-api-reference/en/latest/APIs/tasks/TimeOut/>

Essayer le programme suivant

```
#include "mbed.h"

Timeout flipper;
DigitalOut led1(LED1);

double t=0.1;

void periode() {
    t=1.0;
}

int main() {
    flipper.attach(&periode, 3.0);
```



```
while(1)
{
    led1 = !led1;
    wait(t);
}
```

L'objet flipper appelle la fonction période après 3 secondes

### 3 Interruptions périodiques, la classe Ticker

La classe Ticker permet l'appel périodique d'une fonction, sa documentation est ici : <https://docs.mbed.com/docs/mbed-os-api-reference/en/latest/APIs/tasks/Ticker/>

Le programme ci dessous fera clignoter la LED verte 5 fois. Le programme principal ne fait rien (boucle sans fin).

La méthode « attach » est similaire à celle de la classe Timeout, la méthode detach détruit l'objet.

```
#include "mbed.h"

Ticker flipper;
DigitalOut led1(LED1);

void flip() {
    static int cpt=9; // compteur d'interruptions
    led1 = !led1;
    if (!cpt--) flipper.detach();
}

int main() {

    flipper.attach(&flip, 0.2);
    while(1) ;
}
```

### 4 Exercices

#### Exercice 1

Réaliser un chronomètre déclenché et arrêté par l'appui sur une touche du clavier, l'affichage se fera dans la console.

Amélioration : afficher les heures, minutes, secondes et millisecondes

#### Exercice 2

Réaliser un compte à rebours de 10s à l'aide de la classe Ticker. L'affichage se fera toutes les secondes sur la console.

#### Exercice 3

Après le reset la LED verte s'allume, vous avez alors une seconde pour enfoncer le bouton bleu, sinon la LED s'éteint. Utiliser la classe Timeout et l'interruption sur front descendant sur le bouton bleu.