

Université d'Ottawa  
Faculté de génie

École de science informatique  
et de génie électrique



University of Ottawa  
Faculty of Engineering

School of Electrical Engineering  
and Computer Science

**CEG4566/CSI4541/SEG4545**

**Conception de systèmes informatiques en temps réel**

**Hiver 2024**

**Professeur : Mohamed Ali Ibrahim, ing., Ph.D.**

# Minuteries (*Timers*) logicielles

# Minuteries (*Timers*) logicielles: Introduction

- Les minuteries logicielles sont utilisées pour programmer l'exécution d'une fonction à un moment donné dans le futur, ou périodiquement à une fréquence donnée.
- La fonction exécutée par la minuterie logicielle est appelée fonction de rappel (*callback*) de la minuterie logicielle.
- Elles sont mises en œuvre sous le contrôle du noyau RTOS.
- Elles ne nécessitent pas de support matériel et ne sont pas liées aux minuteries matérielles.

# Minuteries logicielles : Activation

## Enable

<code>configSUPPORT_DYNAMIC_ALLOCATION</code>	<code>1</code>
<code>configUSE_TIMERS</code>	<code>1</code>

in FreeRTOSConfig.h

# Minuteries logicielles : Types

- **Minuterie de recharge automatique**
  - Une fois démarré, elle se remettra en marche à chaque fois qu'il expirera, ce qui entraînera l'exécution périodique de sa fonction de rappel.
- **Minuterie à action unique**
  - Une fois démarré, elle exécutera sa fonction de rappel une seule fois.
  - Elle peut être redémarrée manuellement, mais ne se redémarrera pas d'elle-même.

# Minuteries logicielles : Paramètres clés

## **Période**

- Le temps qui s'écoule entre le démarrage de la minuterie logicielle et l'exécution du rappel de la minuterie logicielle.

# Minuteries logicielles : États

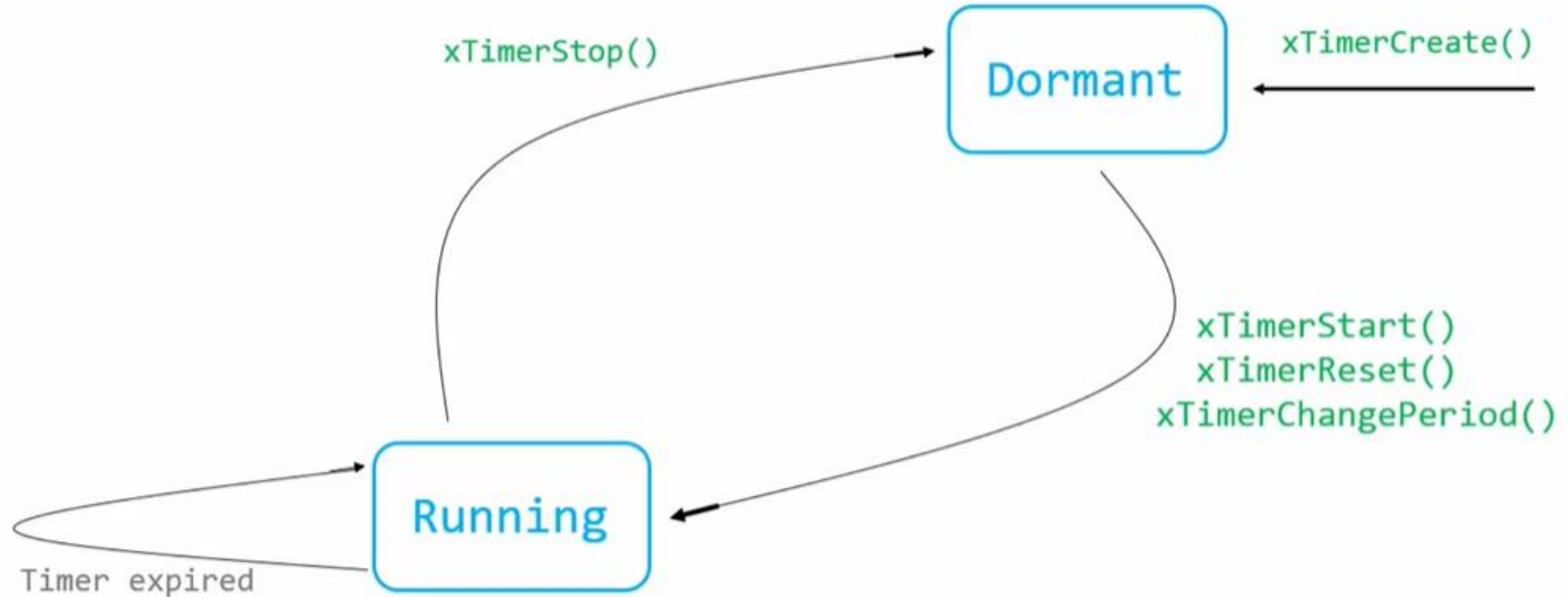
- Dormant

La fonction callback ne s'exécute pas

- Running

La fonction callback s'exécute

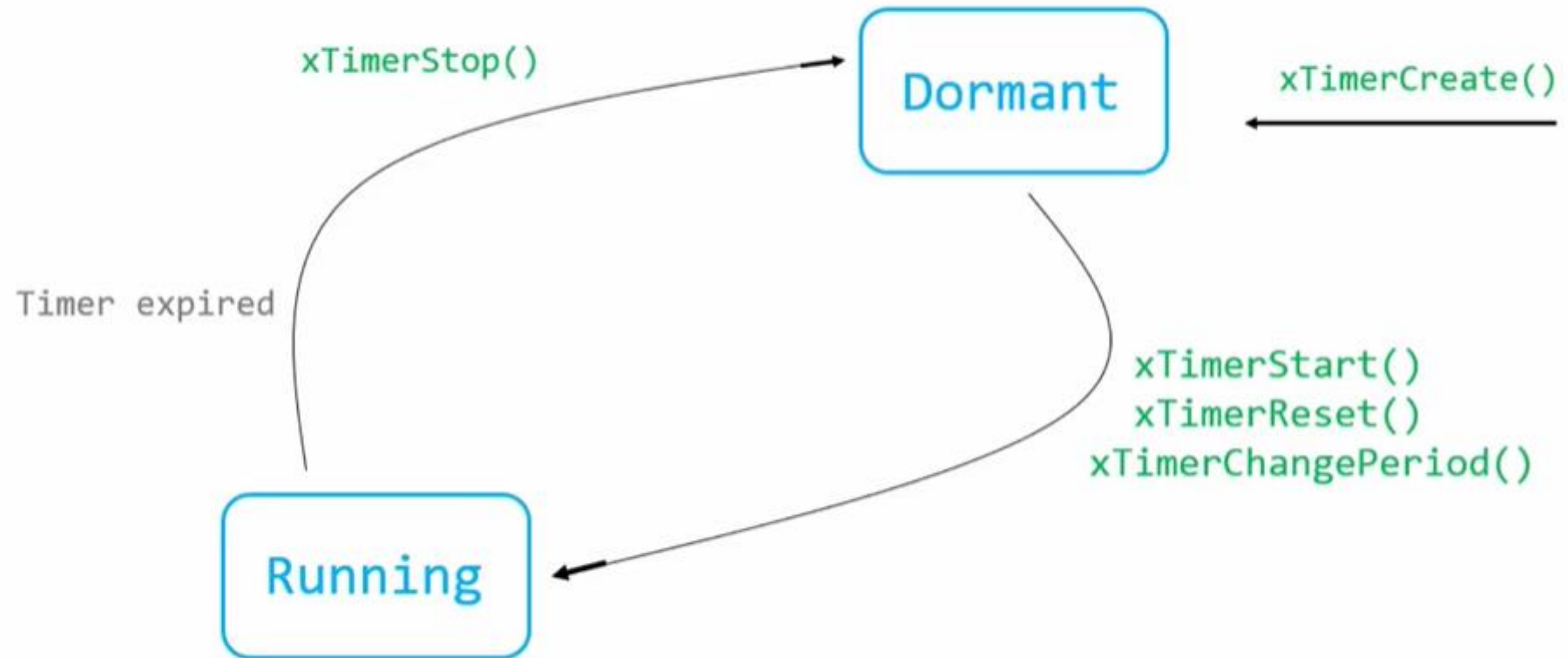
# Minuteries logicielles: États



Auto - recharger les états et les transitions de la minuterie logicielle



# Minuteries logicielles: États



Auto - états et transitions de la minuterie logicielle de prise de vue

# Minuteries logicielles: API d'usage courant

xTimerCreate()

```
TimerHandle_t xTimerCreate(  
    const char * const pcTimerName,  
    TickType_t xTimerPeriodInTicks,  
    UBaseType_t uxAutoReload,  
    void * pvTimerID,  
    TimerCallbackFunction_t pxCallbackFunction  
);
```

# Minuteries logicielles: API d'usage courant

Timer Id

```
void vTimerSetTimerID( const TimerHandle_t xTimer,  
                      void *pvNewID  
                      );
```

```
void *pvTimerGetTimerID( TimerHandle_t xTimer );
```