Electronique Numérique

Systèmes à microprocesseur

S5 : Mbed Timer

Frédéric Giamarchi
IUT GEII Nîmes
Centre Spatial Universitaire (Montpellier Nîmes)
Université de Montpellier
frederic.giamarchi@umontpellier.fr

Mbed Timer

- Un timer est un compteur interne au µC qui s'incrémente par une horloge interne ou externe. Les STM32 possèdent plusieurs timers.

- Cela permet de maitriser le temps d'une action.
- Générations de signaux externes (1/0, MLI, SPI, I2C, UART, USB, …)
- Gestion des actions dans le temps
- Mesure de durées
- Et de rendre les actions indépendantes du temps d'exécution des instructions. On arrête d'utiliser les fonctions (wait, wait_ms).

Mbed Timer

- Cette activité des timers permet une plus grande souplesse de programmation. Ce qui engendre des gestions globales des tâches plus performantes.

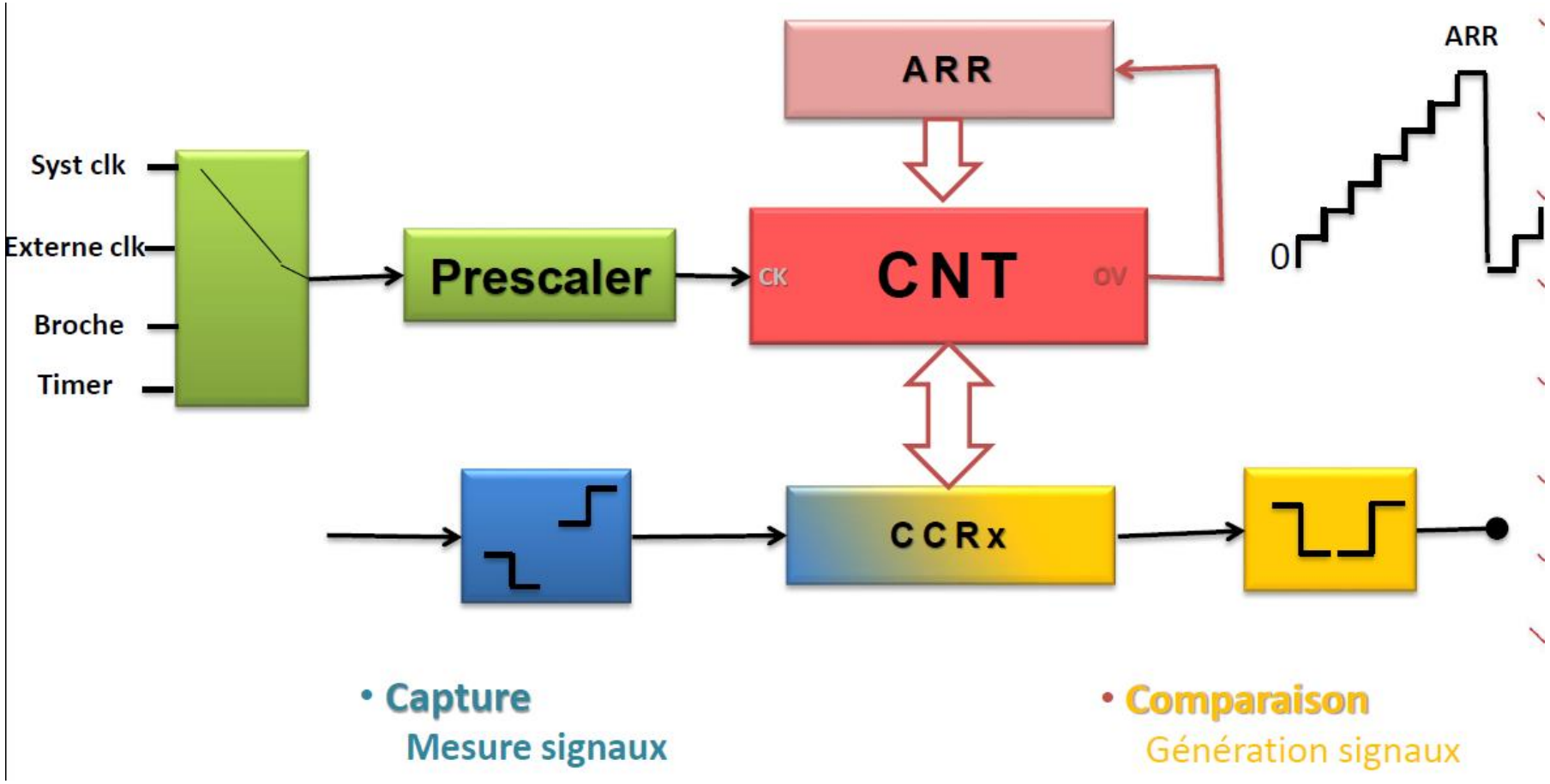
- Synchronisation par compteur (Timer)
- Approche d'un système temps réel (RTOS)
- Multitâche coopératif
- Chaque tâche (action) semble indépendante.

STM32 Timers

Table 6. Timer feature comparison

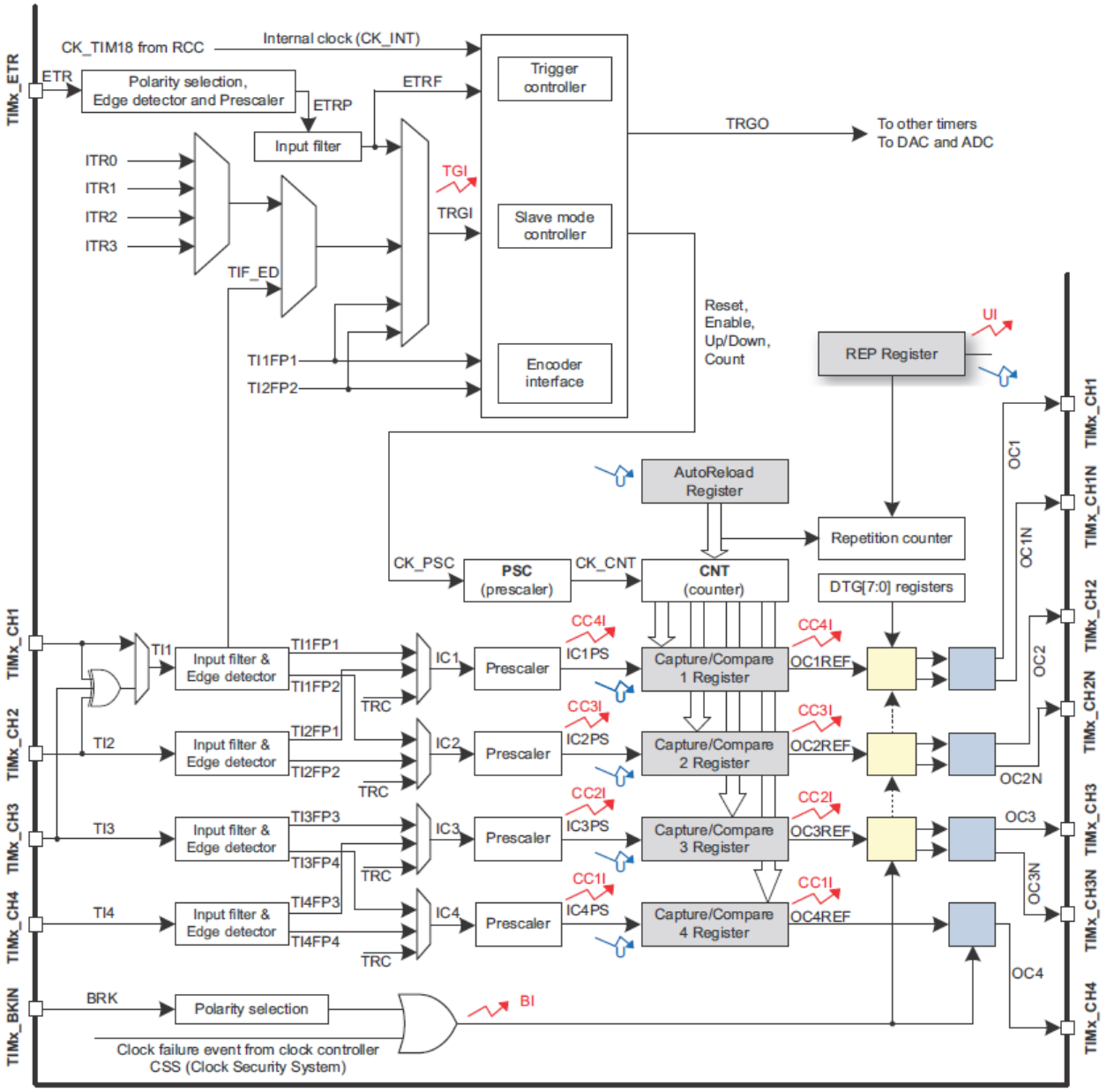
Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary outputs
TIM2, TIM3, TIM4	16-bit	Up, down, up/down	Any integer between 1 and 65536	Yes	4	No
TIM9	16-bit	Up	Any integer between 1 and 65536	No	2	No
TIM10, TIM11	16-bit	Up	Any integer between 1 and 65536	No	1	No
TIM6, TIM7	16-bit	Up	Any integer between 1 and 65536	Yes	0	No

STM32 Timers



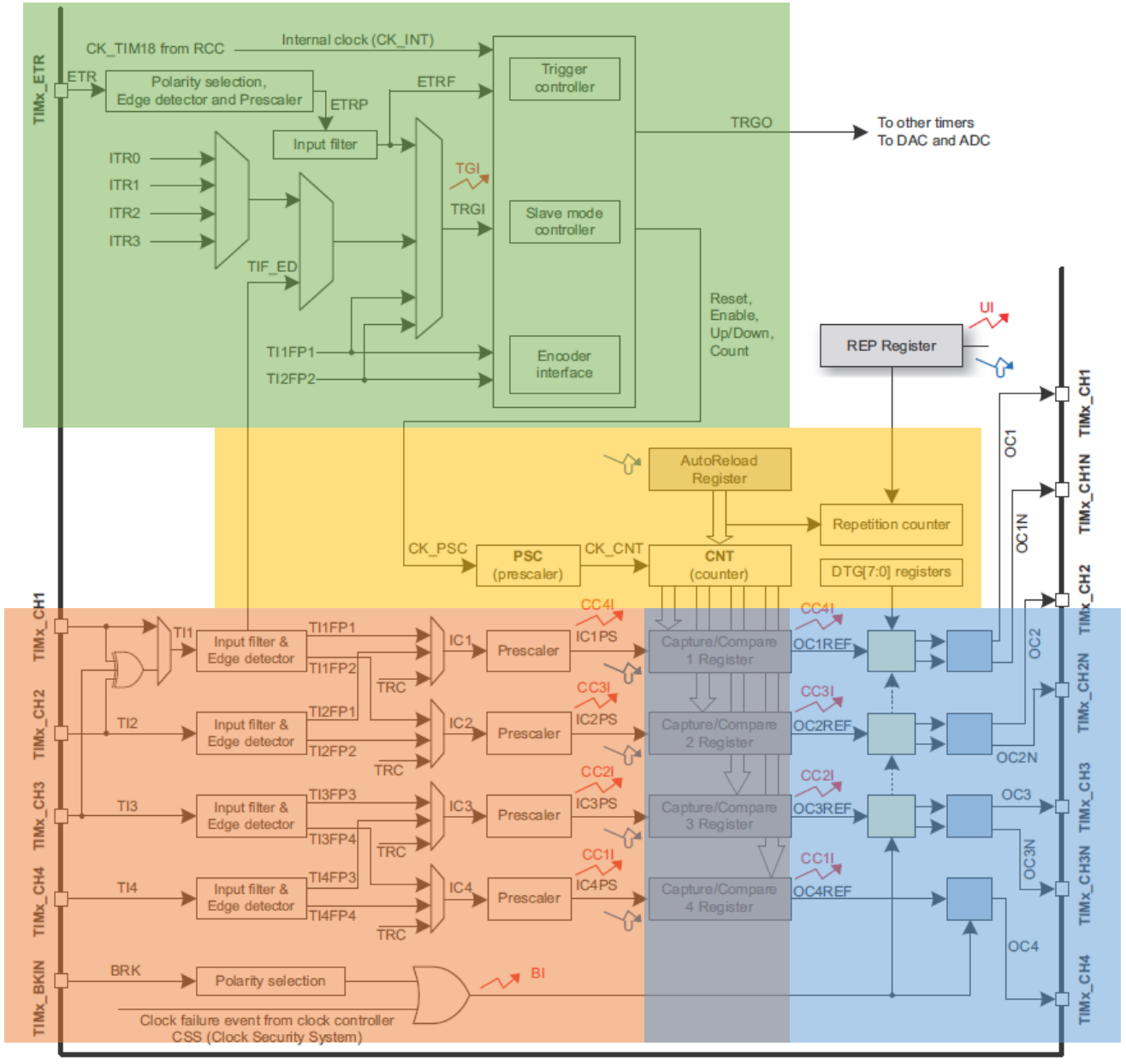
STM32 Timers

- Choix de l'horloge
- Prescaler (prédiviseur)
- Compteur (CNT)
- Capture
- Comparaison
- PWM (MLI)
- ARR (Auto reload Reg.)
- Overflow
- Interruption



STM32 Timers

- Choix de l'horloge
- Prescaler (prédiviseur)
- Compteur (CNT)
- Capture
- Comparaison
- PWM (MLI)
- ARR (Auto reload Reg.)
- Overflow
- Interruption



STM32 Timers

- Timer
  - Gestion d'un compteur
  - Liberté de paramétrage
- Ticker
  - Gestion d'un évènement périodique
- Horloge Temps réel (Real_Time Clock)
  - Compteur de gestion d'une horloge (sec, min, heu, jour, mois, année, alarme)
- PWM Gestion d'un signal paramétrable
- Timeout Gestion de la fin d'un compteur
- Interruption Gestion d'évènements asynchrones

Timer Gestion d'un compteur

- Timer Name; Déclaration d'un compteur
  - Name est le nom que vous souhaitez donner à ce compteur
- Notes
  - Compteur 32 bits
  - Mais avec 1 bit de signe
  - 0 à (2^{31}-1) µs, soit plus de 30min
- Fonctions associées
  - Name.start();
  - Name.stop();
  - Name.reset();
  - Name.read(); (read_ms et read_us)

```
#include "mbed.h“

Timer t;

int main() {
   t.start();                  // compteur actif
   pc.printf(“Bonjour !\n”);
   t.stop();                   // compteur arrété
   pc.printf(“Duree du texte : %f s\n”,t.read());
   while(1) {
                }
}
```

Horloge temps réel

Gestion d'une horloge

- Variable
  - time_t valeur; Déclaration d'une variable typée
- Fonctions associées
  - time(NULL); Lecture de l'horloge
  - set_time(chiffre); nbre de sec depuis le 1/01/1970
  - Mktime Gestion d'une structure
  - localtime
  - ctime(&valeur); Converti pour une lecture
- Notes
  - Peut avoir sa propre alimentation séparée

```
#include "mbed.h“

time_t horloge;

int main() {
// Actualise l'horloge 4 dec 2018 à
17h37 et 0sec
   set_time(1543941420);

   while(1) {
   horloge = time(NULL);
   printf("%s\n", ctime(&horloge));
   wait(1);
   }
}
```

Liens

- UTC time.is/fr/
  - Horloge universelle
- TimeStamp unixtimestamp.com
  - Convertisseur temps universel