

Enseignes et afficheurs à LED

# Horloges temps réel

# Horloges temps réel

**Pierre-Yves Rochat**

- Principe d'une horloge électronique
- Alimentation permanente
- Programmation avec un microcontrôleur
- Circuits spécialisés
- Horloges internes

- Les montres mécaniques utilisent un **mouvement oscillant** aussi stable que possible

- Les montres mécaniques utilisent un **mouvement oscillant** aussi stable que possible
- Des **engrenages** permettent ensuite d'obtenir les secondes, les minutes et les heures


- Les montres mécaniques utilisent un **mouvement oscillant** aussi stable que possible
- Des **engrenages** permettent ensuite d'obtenir les secondes, les minutes et les heures
- Une montre électronique utilise un oscillateur à **quartz** comme base de temps

- Les montres mécaniques utilisent un **mouvement oscillant** aussi stable que possible
- Des **engrenages** permettent ensuite d'obtenir les secondes, les minutes et les heures
- Une montre électronique utilise un oscillateur à **quartz** comme base de temps
- Généralement à 32'768 Hz

- Les montres mécaniques utilisent un **mouvement oscillant** aussi stable que possible
- Des **engrenages** permettent ensuite d'obtenir les secondes, les minutes et les heures
- Une montre électronique utilise un oscillateur à **quartz** comme base de temps
- Généralement à 32'768 Hz



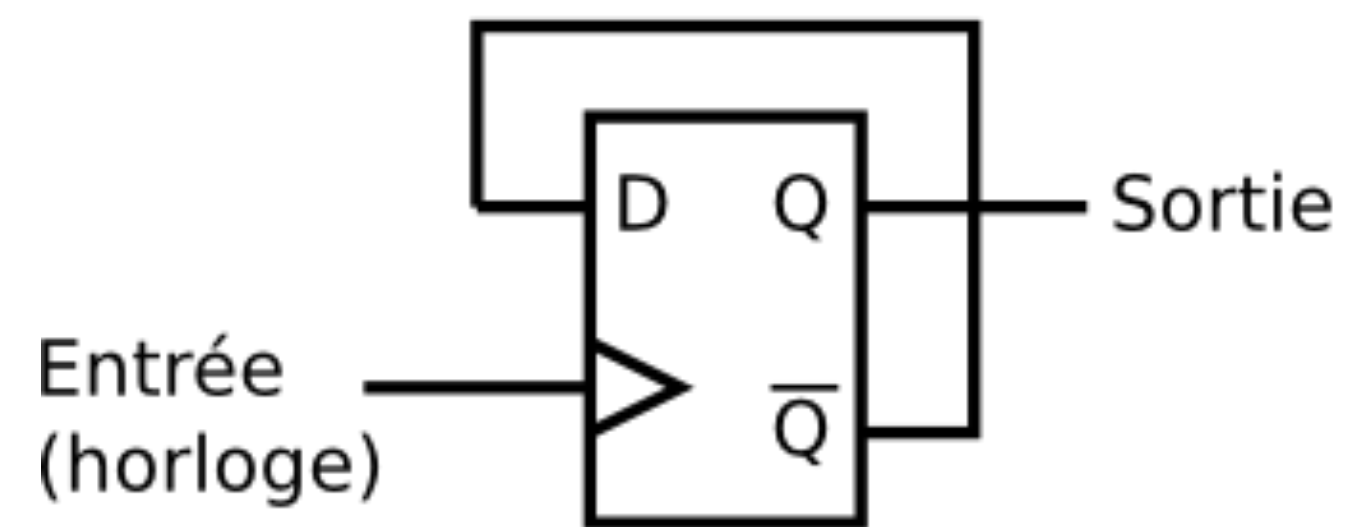


- Les montres mécaniques utilisent un **mouvement oscillant** aussi stable que possible
  - Des **engrenages** permettent ensuite d'obtenir les secondes, les minutes et les heures
  - Une montre électronique utilise un oscillateur à **quartz** comme base de temps
  - Généralement à 32'768 Hz
- 
- 15 diviseurs par 2 vont permettre d'obtenir un signal de 1 Hz

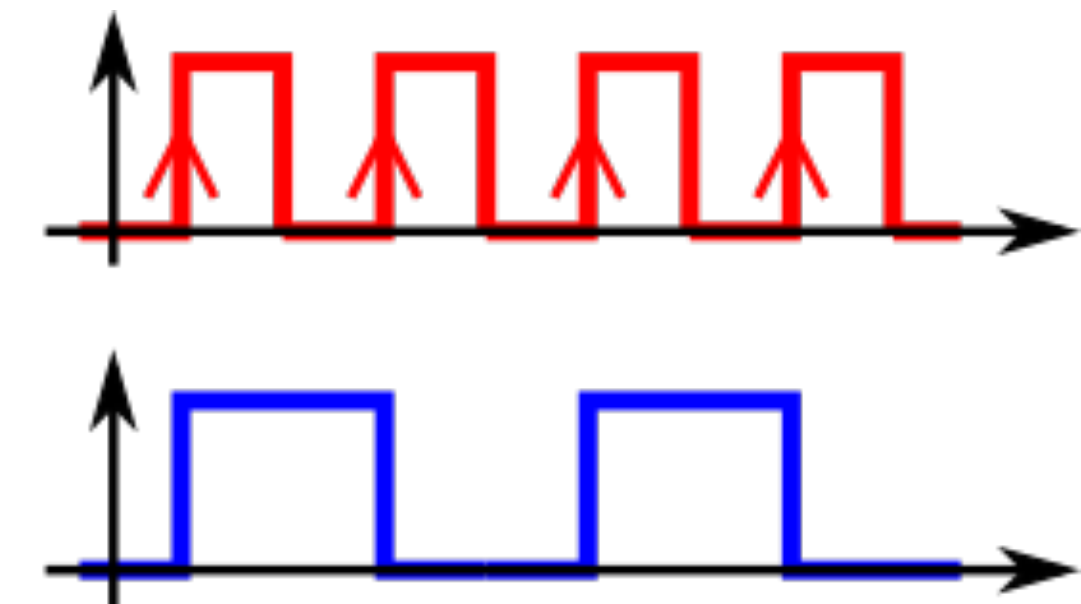
# Base de temps et division



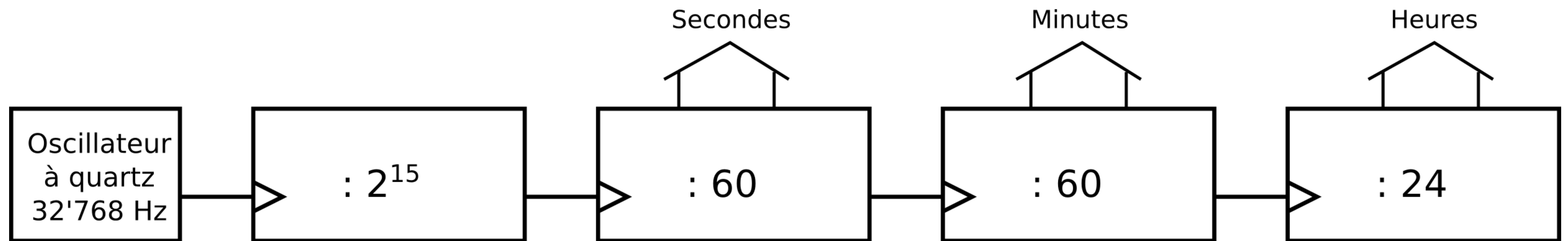
# Base de temps et division



$$\left. \begin{array}{l} Q^+ = D \\ D = \bar{Q} \end{array} \right\} \Rightarrow Q^+ = \bar{Q}$$



# Base de temps et division



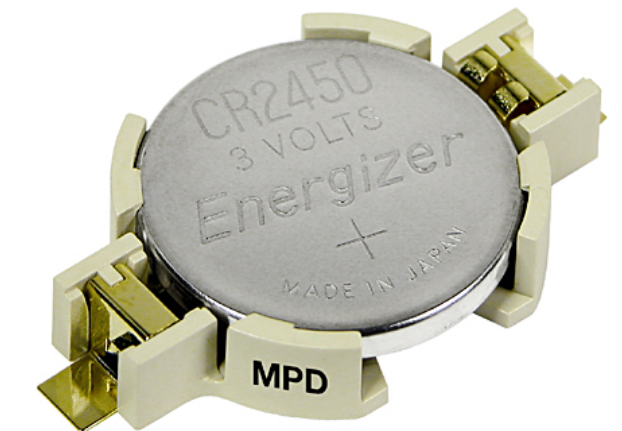
- Une alimentation doit être prévue lorsque le dispositif n'est pas sous tension

- Une alimentation doit être prévue lorsque le dispositif n'est pas sous tension
- On utilise souvent des piles au Lithium

- Une alimentation doit être prévue lorsque le dispositif n'est pas sous tension
- On utilise souvent des piles au Lithium



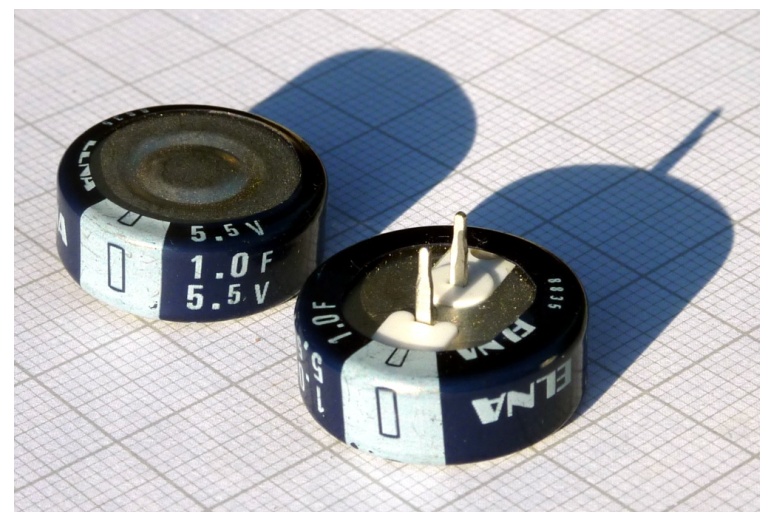
- Une alimentation doit être prévue lorsque le dispositif n'est pas sous tension
- On utilise souvent des piles au Lithium



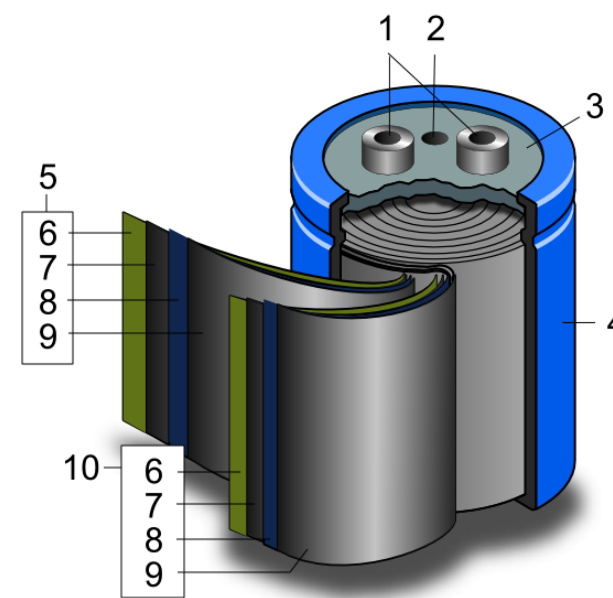
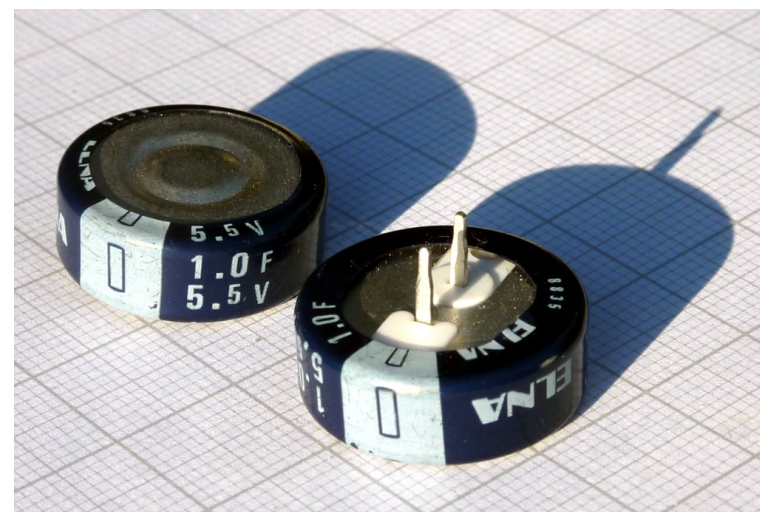


- Une alimentation doit être prévue lorsque le dispositif n'est pas sous tension
- On utilise souvent des piles au Lithium
- Des super-condensateurs *Supercap* peuvent aussi être utilisés

- Une alimentation doit être prévue lorsque le dispositif n'est pas sous tension
- On utilise souvent des piles au Lithium
- Des super-condensateurs *Supercap* peuvent aussi être utilisés



- Une alimentation doit être prévue lorsque le dispositif n'est pas sous tension
- On utilise souvent des piles au Lithium
- Des super-condensateurs *Supercap* peuvent aussi être utilisés



# Programmation sur un microcontrôleur

- Beaucoup de microcontrôleurs prévoient l'usage d'un **quartz à 32 kHz**

# Programmation sur un microcontrôleur

- Beaucoup de microcontrôleurs prévoient l'usage d'un **quartz à 32 kHz**
- Il va falloir mettre le microcontrôleur **en veille** un maximum de temps

# Programmation sur un microcontrôleur

- Beaucoup de microcontrôleurs prévoient l'usage d'un **quartz à 32 kHz**
- Il va falloir mettre le microcontrôleur **en veille** un maximum de temps
- Le code va fortement dépendre de la **famille** du microcontrôleur utilisé

# Programme pour un AVR

```
1 #include <avr/io.h>
2 #include <avr/interrupt.h>
3 #include <avr/sleep.h>
4
5 volatile uint8_t secondes;
6
7 // Il faut un quartz 32 khz sur les broches TOSC1 et TOSC2
8 ISR (TIMER2_OVF_vect) {
9     secondes++;
10    ...
11 }
12
13 int main () {
14     Temps=0;
15     ASSR=(1<<AS2); // oscillateur quartz 32 khz
16     TCCR2B=0b101; // prédivison par 128
17     TIMSK2=(1<<TOIE2); // interruption Timer2 Overflow autorisée
18     sei(); // toutes les interruptions autorisées
19     while (1) { // boucle correspondant à tous les réveils dus à l'interruption
20         set_sleep_mode(SLEEP_MODE_PWR_SAVE); // mise en veille
21         sleep_enable();
22         sleep_mode(); // mode normal après un réveil
23         sleep_disable();
24     }
25 }
```

# Programme pour un AVR

```
8    ...
9    secondes++;
10   if (secondes == 60) {
11       secondes = 0;
12       minutes++;
13       if (minutes == 60) {
14           minutes = 0;
15           heures++;
16           if (heures == 24) {
17               heures = 0;
18               ...
19           }
20       }
21   ...
```

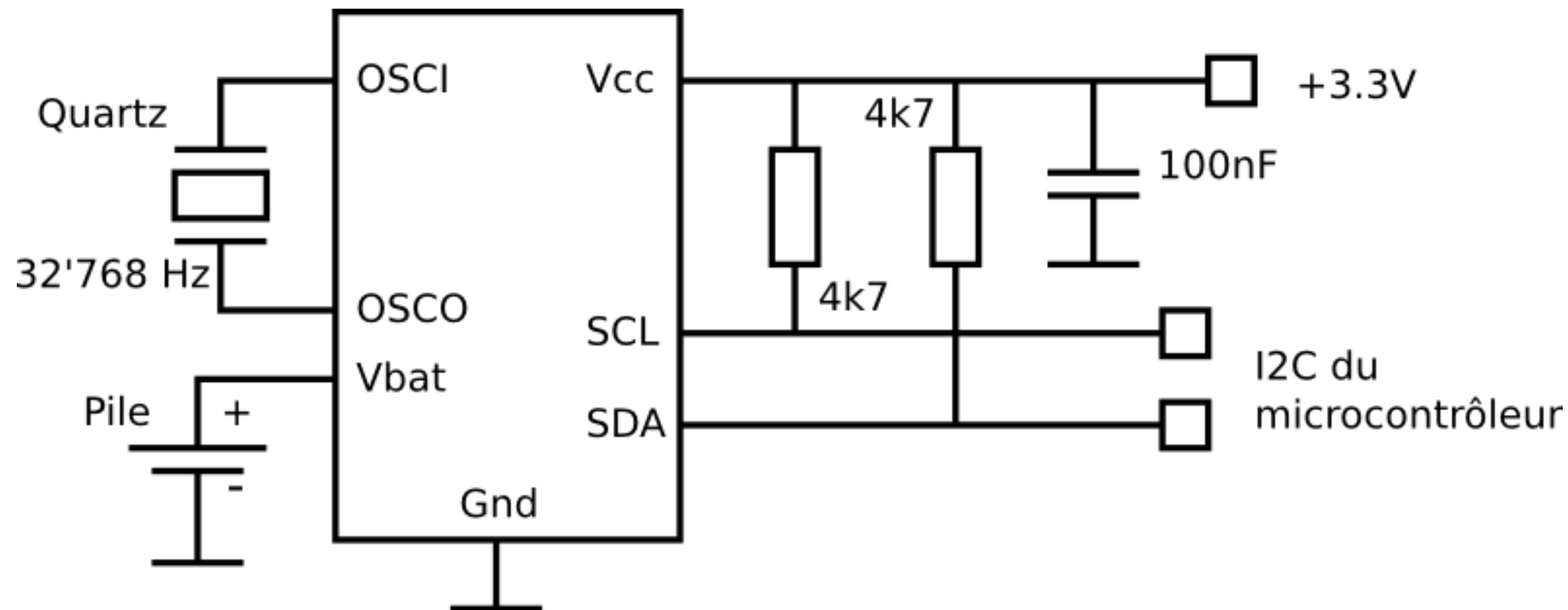


- Il existe de nombreux circuits intégrés qui réalisent une horloge temps réel

- Il existe de nombreux circuits intégrés qui réalisent une horloge temps réel
- DS1307, PCF8523, PCF8563, bq32000, ...

# Circuits intégrés spécialisés

- Il existe de nombreux circuits intégrés qui réalisent une horloge temps réel
- DS1307, PCF8523, PCF8563, bq32000, ...

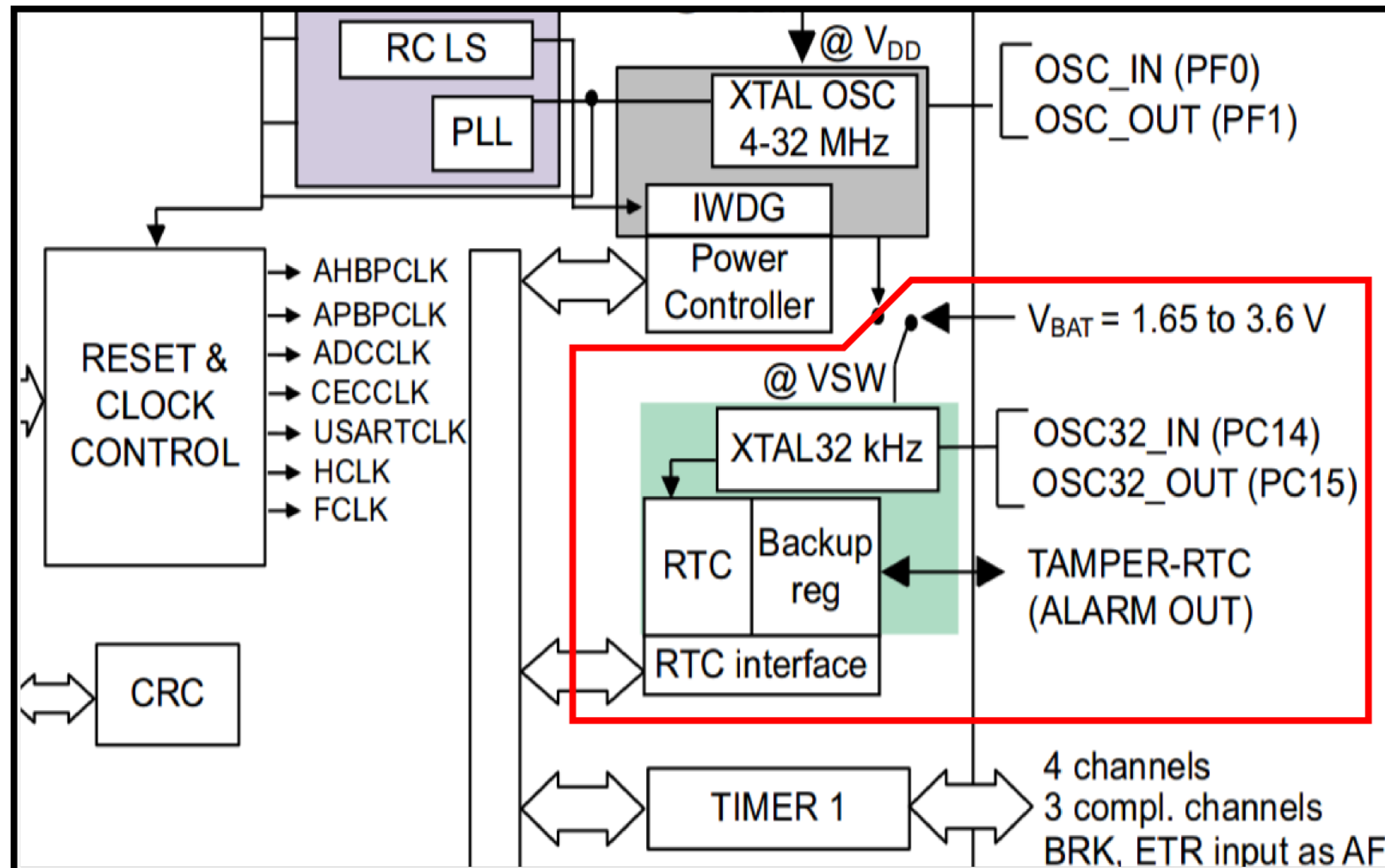


# Horloge temps réel interne

- Certains microcontrôleurs intègrent une **zone séparée** prévue pour une horloge temps réel

# Horloge temps réel interne

- Certains microcontrôleurs intègrent une **zone séparée** prévue pour une horloge temps réel



- Principe d'une horloge électronique
- Alimentation permanente
- Programmation avec un microcontrôleur
- Circuits spécialisés
- Horloges internes