

Enseignes et afficheurs à LED

PWM : Modulation de Largeur d'Impulsion

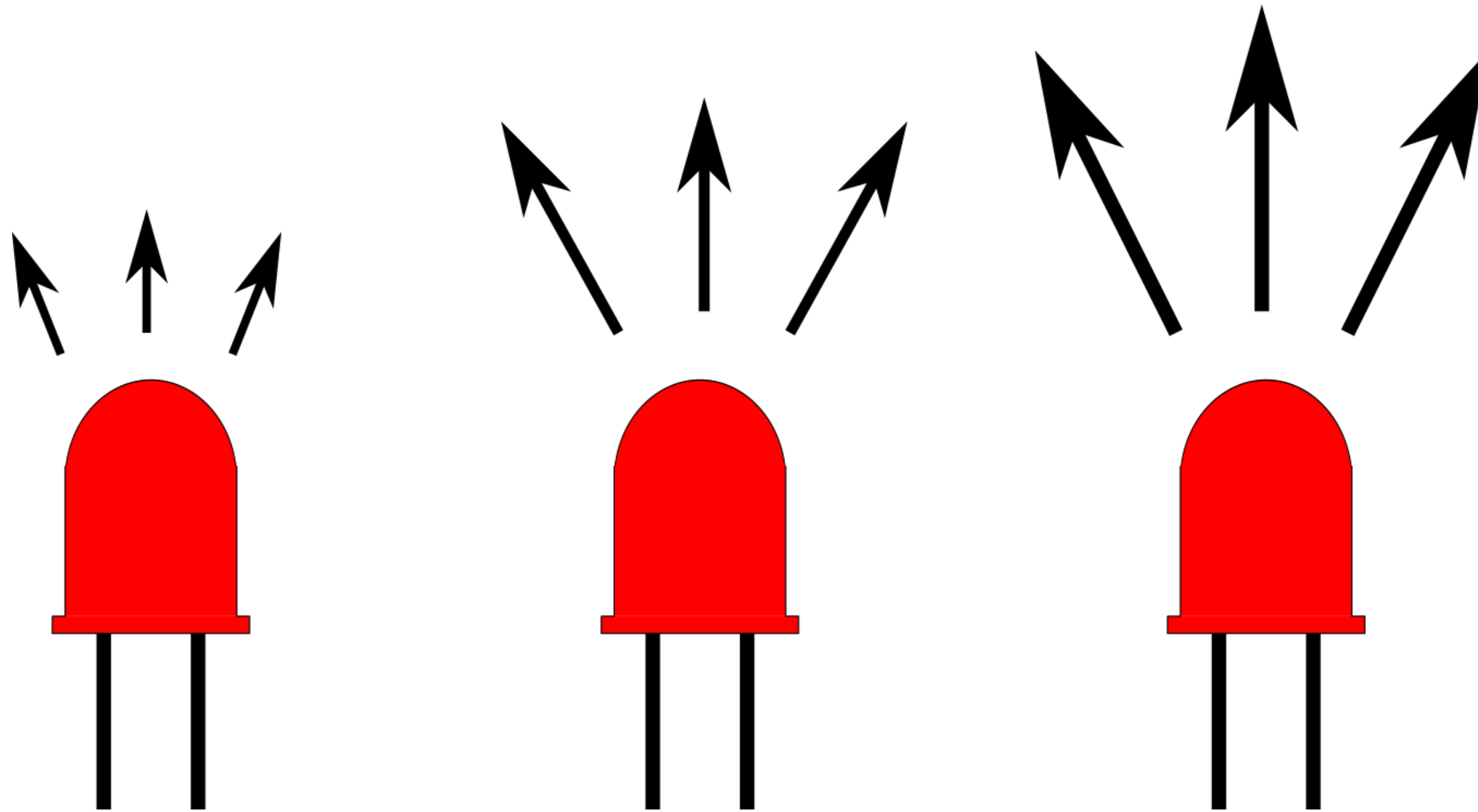
PWM : Modulation de Largeur d'Impulsion

Pierre-Yves Rochat

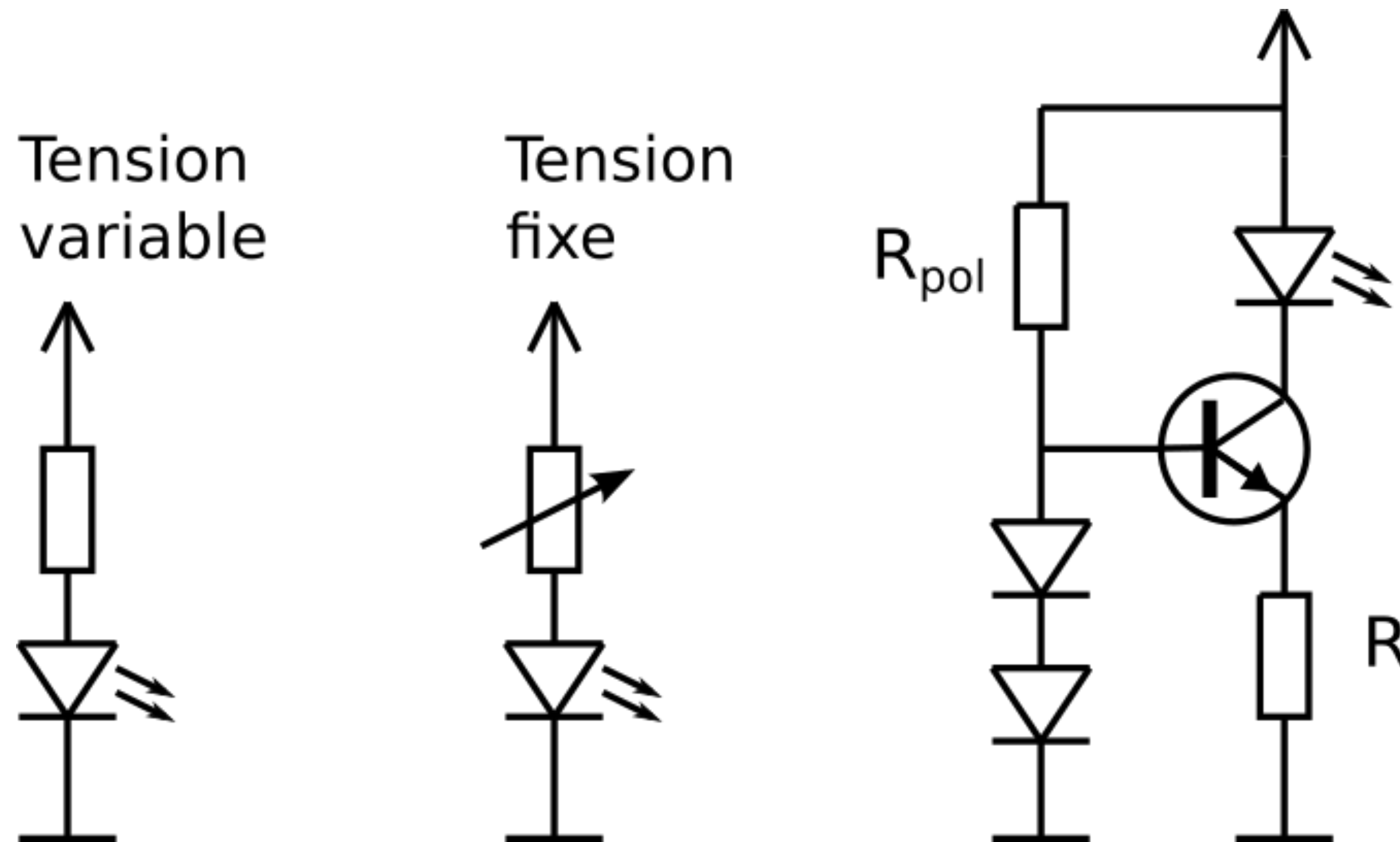
PWM : Modulation de Largeur d'Impulsion

- Principe
- Fréquence
- Programmer un PWM
- Convertisseur numérique-analogique
- Réalisation par des circuits logiques

Comment faire varier l'intensité d'une LED ?



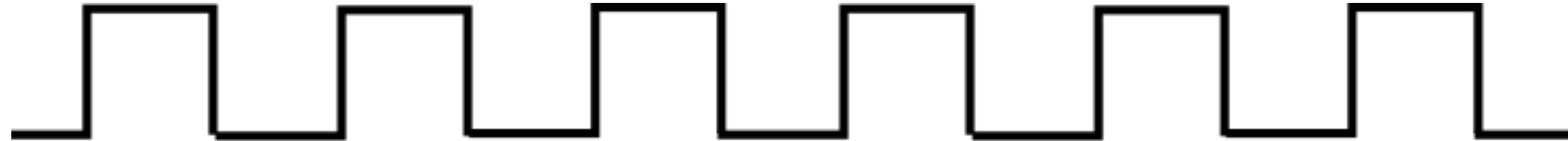
Comment faire varier l'intensité d'une LED ?



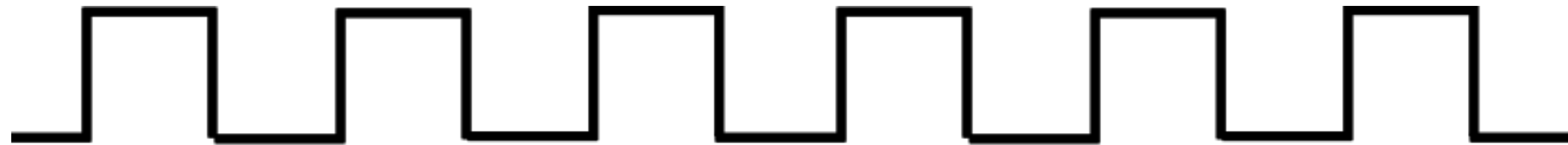
Autre solution

?

Clignotement



Clignotement

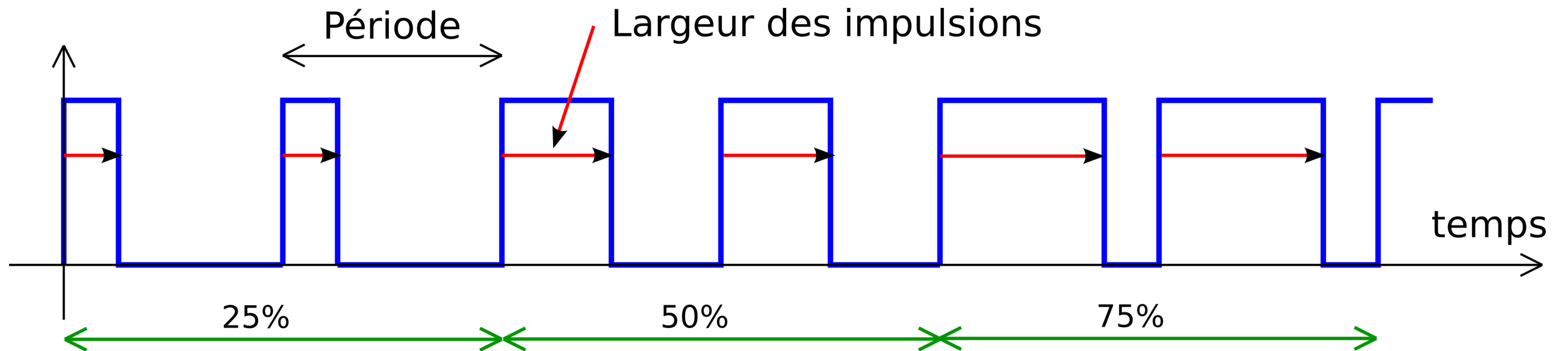


... plus rapide

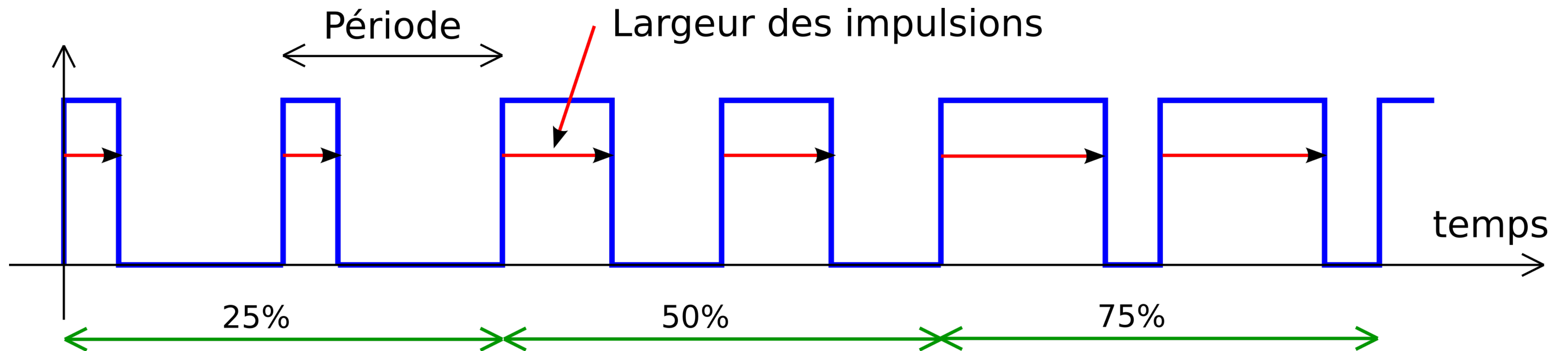


< 20ms

PWM : principe



PWM : principe



Pulse Width Modulation = Modulation de Largeur d'Impulsion

Quelle fréquence



- Selon les applications du PWM, les fréquences sont très différentes, de quelques Hz à des dizaines de MHz.

Quelle fréquence



- Selon les applications du PWM, les fréquences sont très différentes, de quelques Hz à des dizaines de MHz.

Pour des applications visuelles :

Quelle fréquence



- Selon les applications du PWM, les fréquences sont très différentes, de quelques Hz à des dizaines de MHz.

Pour des applications visuelles :

- L'œil a une fréquence limite de perception du clignotement

Quelle fréquence



- Selon les applications du PWM, les fréquences sont très différentes, de quelques Hz à des dizaines de MHz.

Pour des applications visuelles :

- L'œil a une fréquence limite de perception du clignotement
- On ne voit pas clignoter un tube fluorescent, à 100 Hz (2 x 50 Hz)

Quelle fréquence



- Selon les applications du PWM, les fréquences sont très différentes, de quelques Hz à des dizaines de MHz.

Pour des applications visuelles :

- L'œil a une fréquence limite de perception du clignotement
- On ne voit pas clignoter un tube fluorescent, à 100 Hz (2 x 50 Hz)
- Les cônes et les bâtonnets n'ont pas la même fréquence limite

Programmation d'un signal PWM

Comment programmer des signaux PWM avec un microcontrôleur ?

Comment programmer des signaux PWM avec un microcontrôleur ?

- Allumer – attendre

Comment programmer des signaux PWM avec un microcontrôleur ?

- Allumer – attendre
- éteindre – attendre

Comment programmer des signaux PWM avec un microcontrôleur ?

- Allumer – attendre
- éteindre – attendre
- et répéter !

Programmation par période

```
1 #define LedOn digitalWrite(P1_0, 1)
2 #define LedOff digitalWrite(P1_0, 0)
3 uint16_t
4   pwmLed;
5 // valeur du PWM, 0 à 100
6
7 void setup() {
8 // Initialisations
9   pinMode(P1_0, OUTPUT);
10 // LED en sortie
11   pwmLed = 25;
12 // valeur du PWM.
13 }
14
15 void loop() {
16 // Boucle infinie, durée 10ms => un cycle du PWM à 100 Hz
17   LedOn;
18   delayMicrosecond(
```

Comment programmer plusieurs signaux PWM en même temps ?

Comment programmer plusieurs signaux PWM en même temps ?

- Difficile si la boucle principale dure une période complète du PWM

Comment programmer plusieurs signaux PWM en même temps ?

- Difficile si la boucle principale dure une période complète du PWM
- Plus facile si la boucle principale dure le temps de la plus courte impulsion possible du PWM

Programmer plusieurs PW

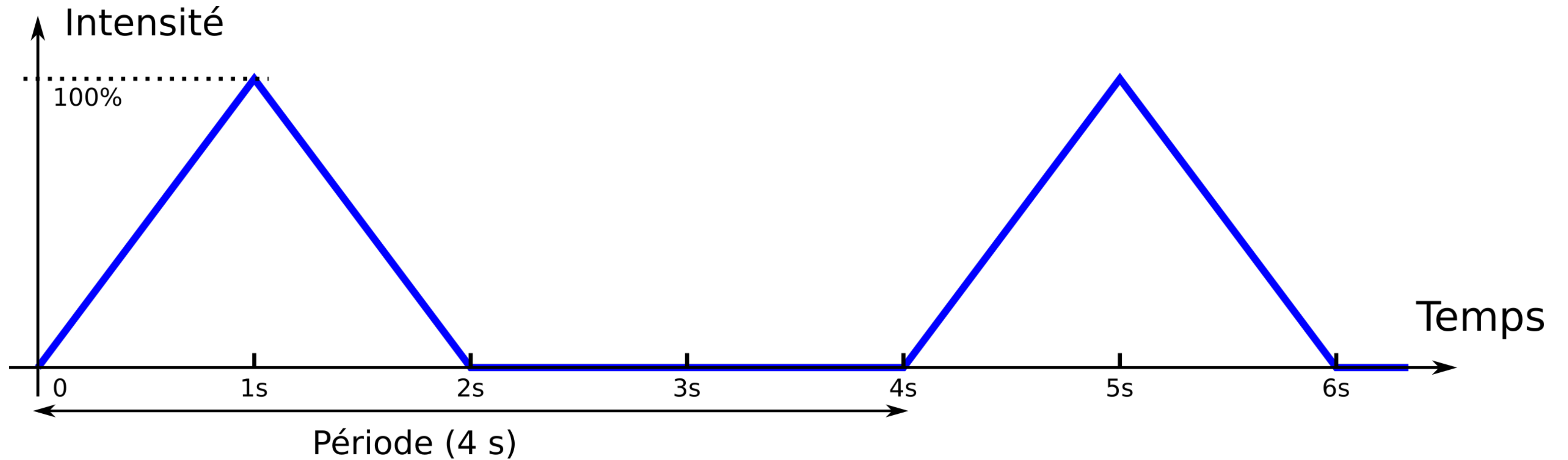
```
1 uint8_t pwmLed;
2 // valeur du PWM, 0 à 255 (8 bits)
3 uint8_t cptPwm;
4 // compteur du PWM
5
6 void setup() {
7 // Initialisations
8   pinMode(P1_0, OUTPUT);
9 // LED en sortie
10  pwmLed = 64;
11 // valeur du PWM. Elle est ici fixe, mais pourrait changer
12
13 // à tout moment en complétant le programme.
14  cptPwm = 0;
15 // compteur du PWM
16 }
17
18 void loop() {
```


- Comment utiliser ce PWM ?

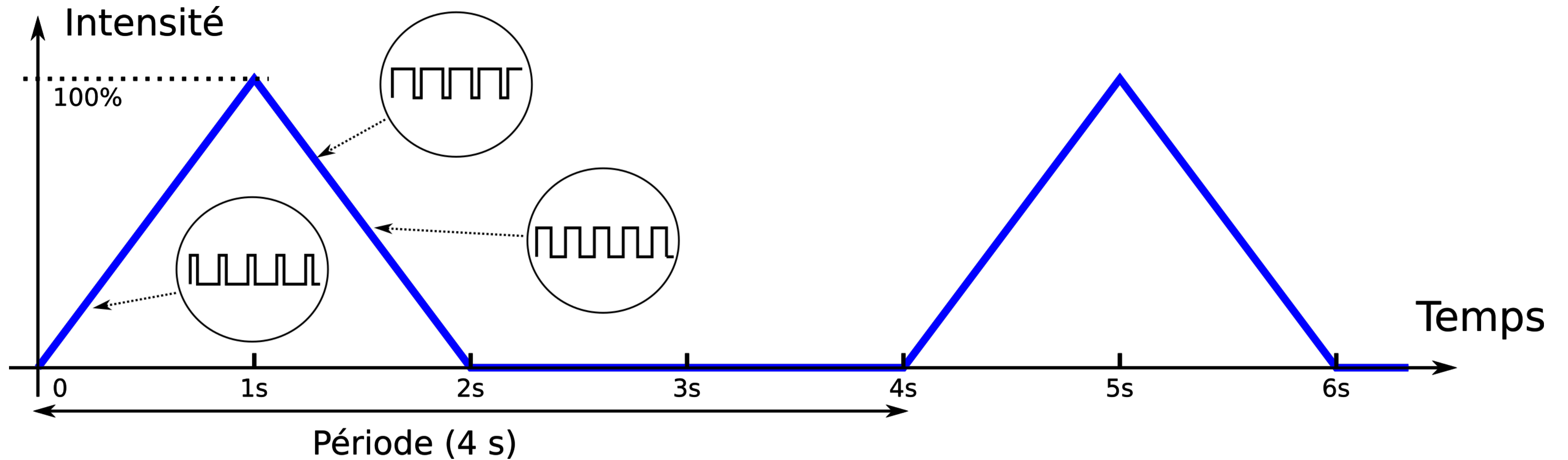
- Comment utiliser ce PWM ?
- Sur des enseignes et afficheurs, on peut créer des séquences.

- Comment utiliser ce PWM ?
- Sur des enseignes et afficheurs, on peut créer des séquences.
- Exemple : LED imitant le repos.

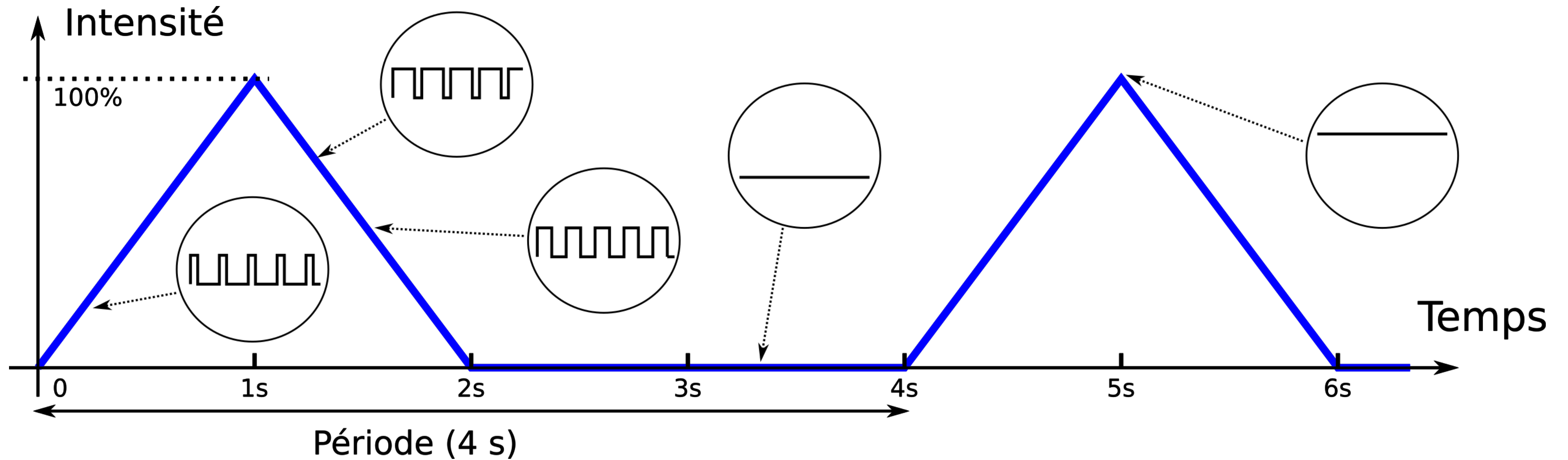
Séquences en PWM



Séquences en PWM



Séquences en PWM



Séquences en PWM

```
1 uint16_t pwmLed;
2 // valeur du PWM, 0 à 255 (8 bits, 16 bits pour les calculs)
3 uint16_t
4   cpt10ms = 0;
5 // compteur des cycles, de 0 à 400 (par 10ms, total 4s)
6 void loop() {
7   // Boucle infinie, durée 39us (256 * 39us = ~10ms)
8   if (cptPwm==0
9 ) {
10     cpt10ms++;
11     if (cpt10ms<
12 100) {
13       //première seconde
14       pwmLed = cpt10ms *
15 256 / 100;
16       // droite montante
17     } else if
18 (cpt10ms<200) {
19       // deuxième seconde
20       pwmLed =
```

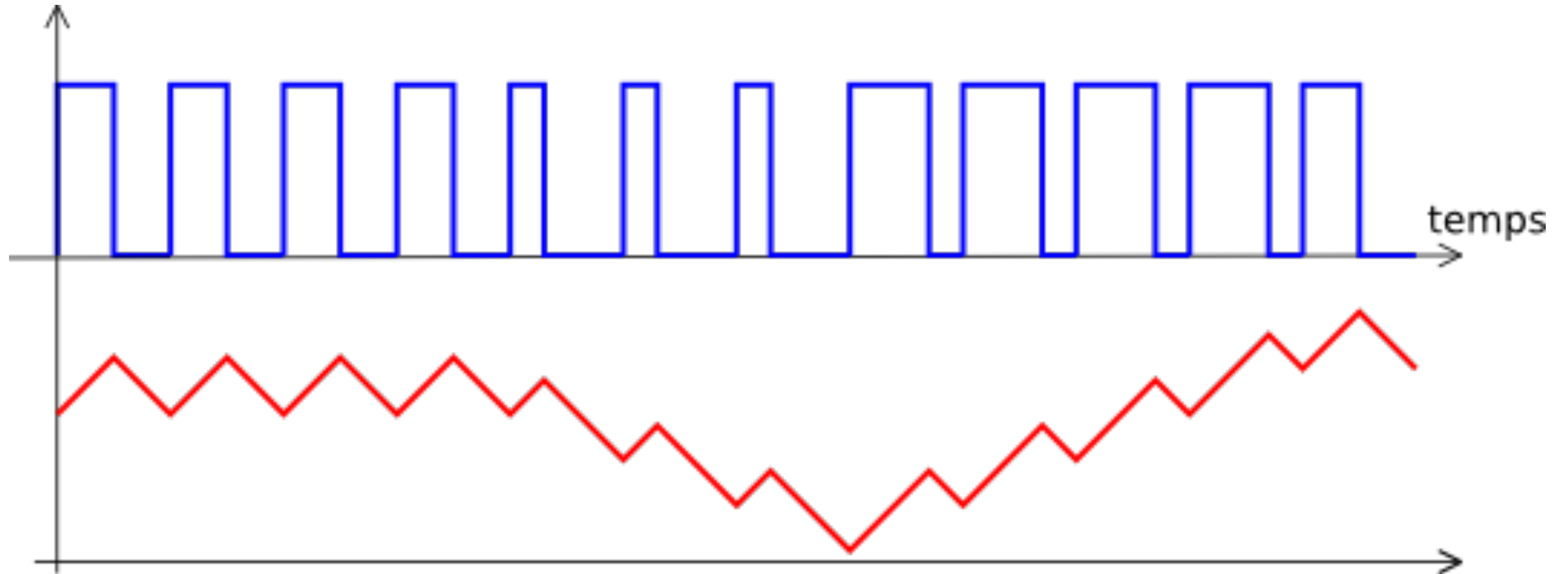
Convertisseur numérique-analogique en PWM

- Transmettre une information variable vers l'extérieur

Convertisseur numérique-analogique en PWM

- Transmettre une information variable vers l'extérieur
- Conversion Numérique-Analogique
DAC Digital to Analog Converter

Convertisseur numérique-analogique en PWM



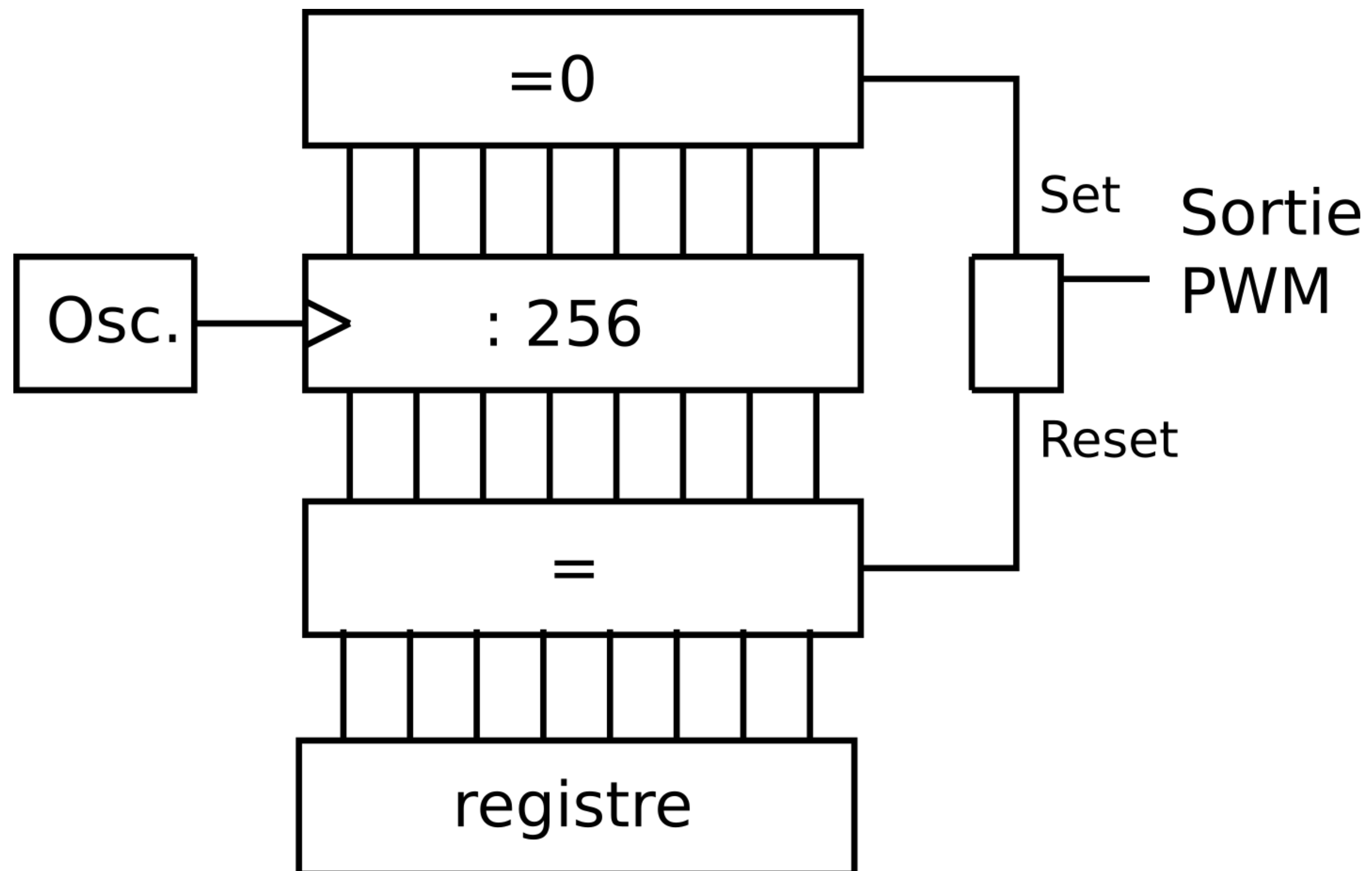
PWM réalisé avec des circuits logiques

- Comment soulager le microcontrôleur de la génération du PWM ?

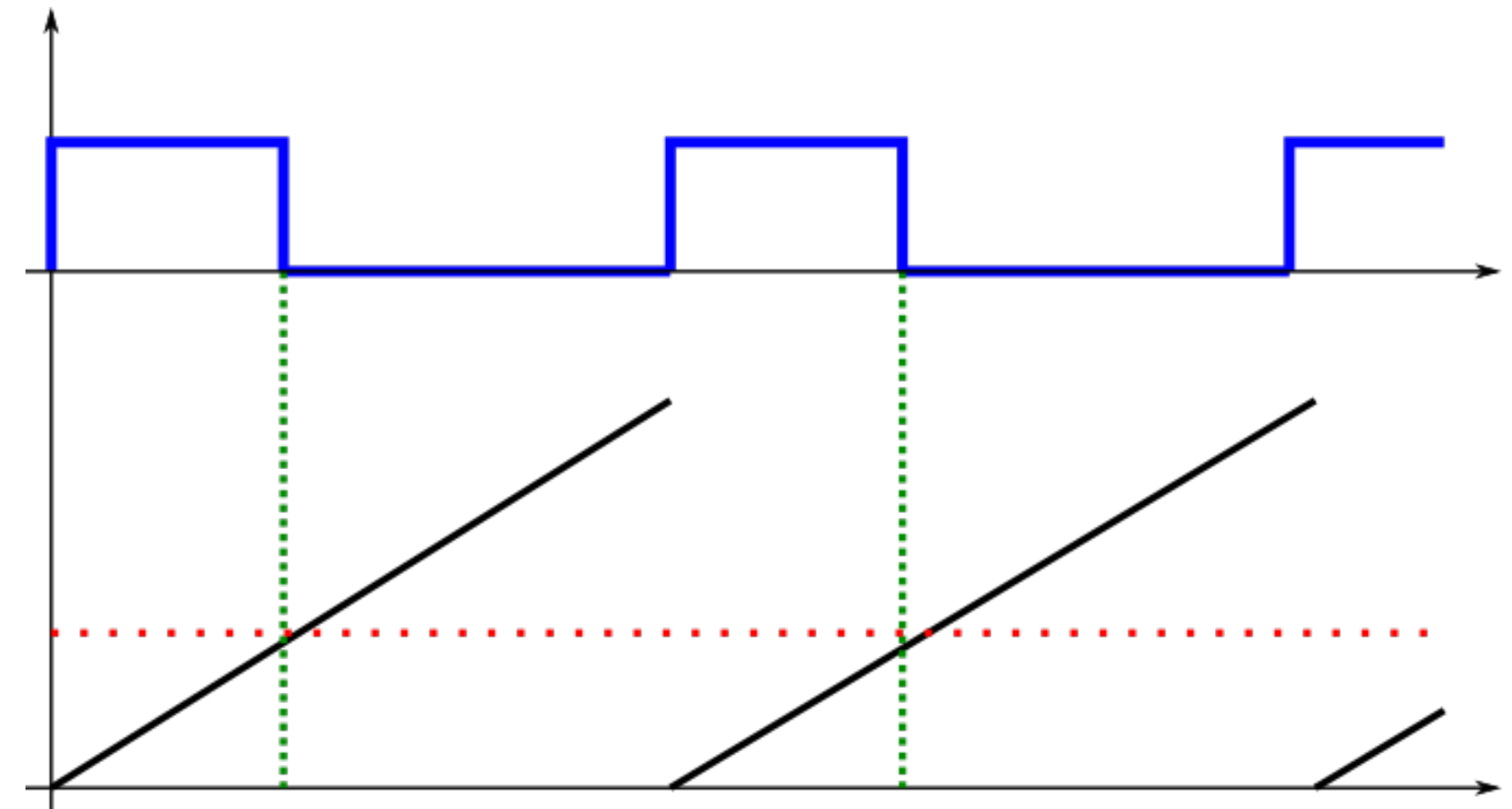
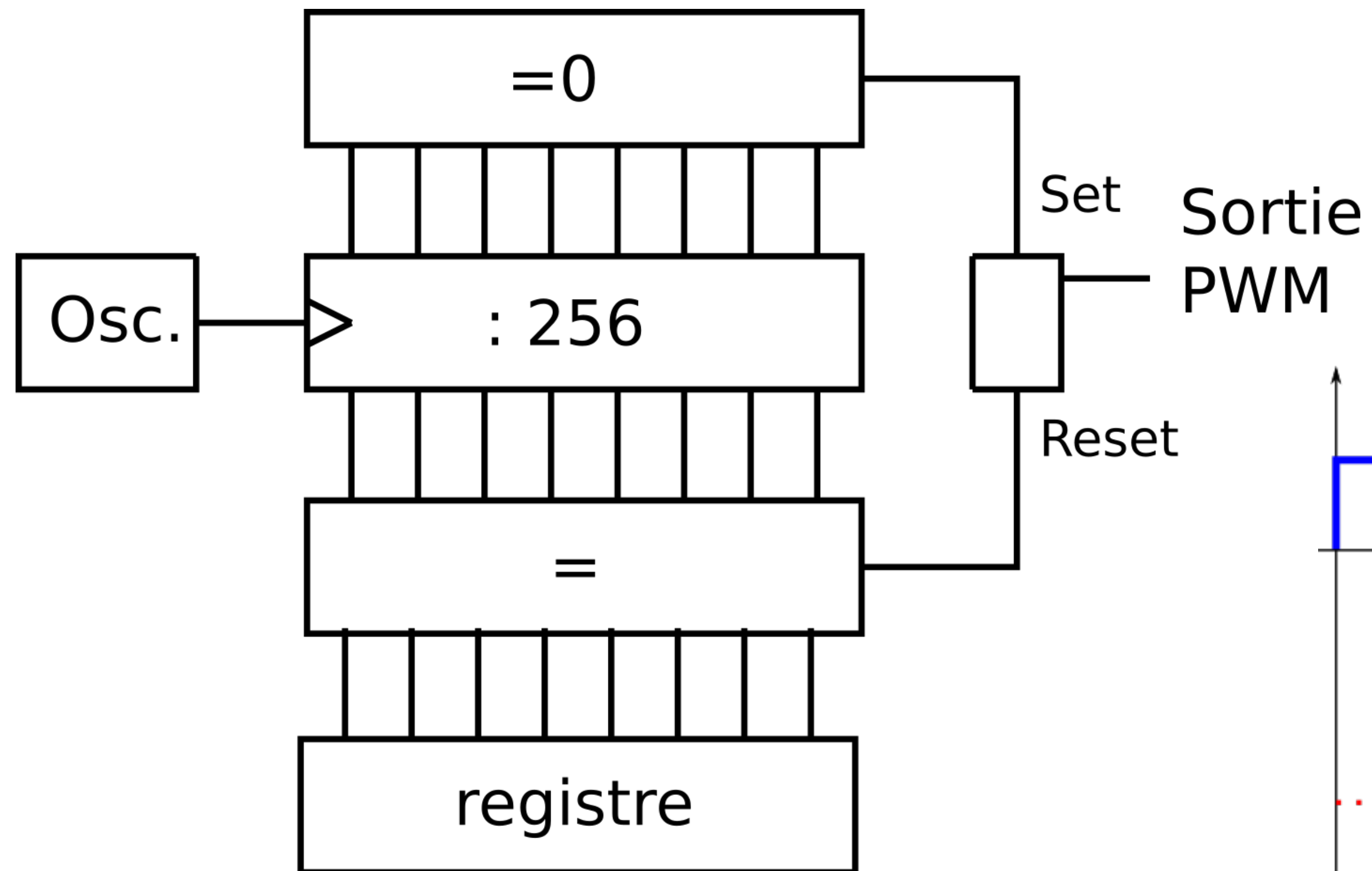
PWM réalisé avec des circuits logiques

- Comment soulager le microcontrôleur de la génération du PWM ?
- En utilisant des circuits logiques spécialisés !

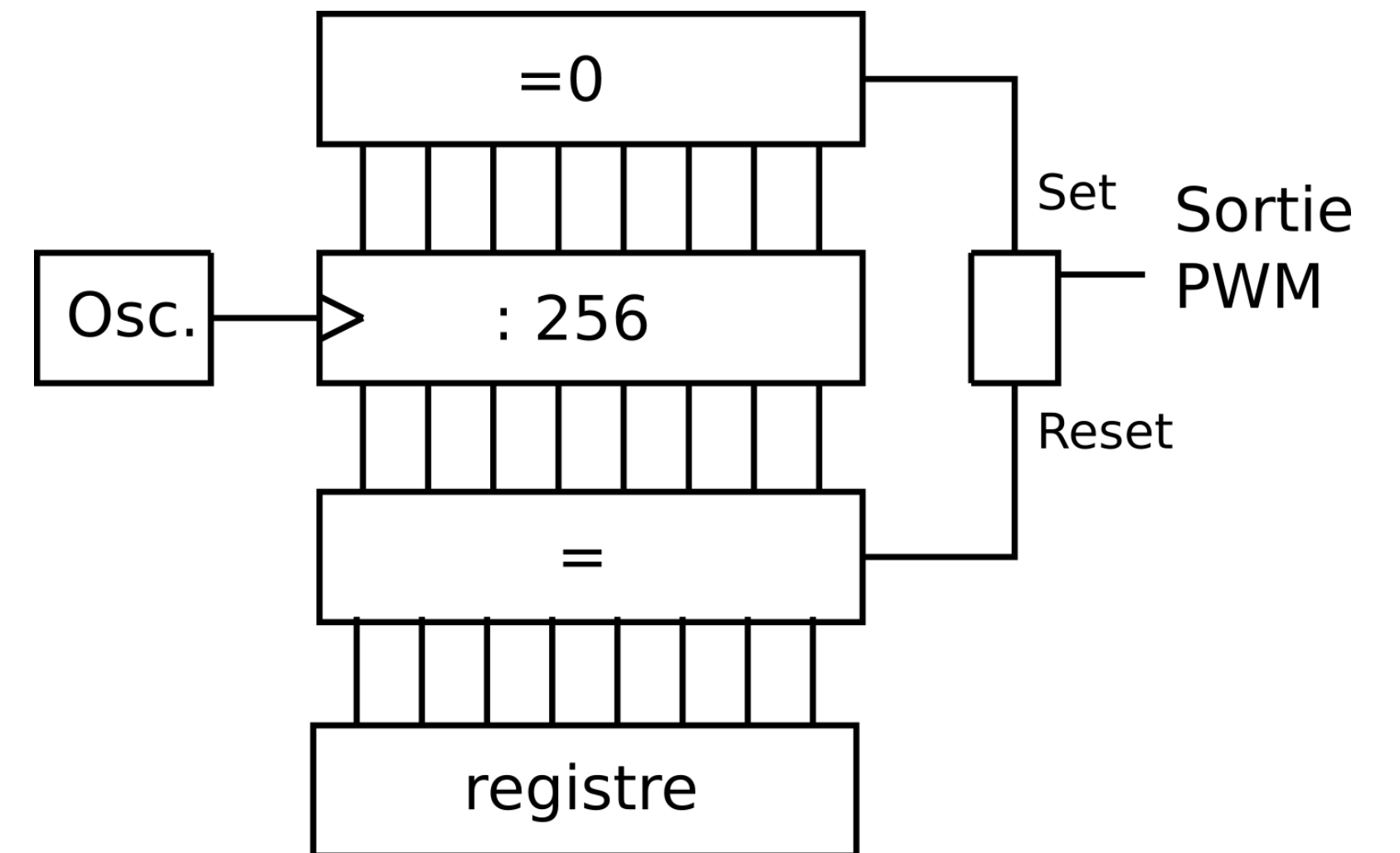
PWM réalisé avec des circuits logiques



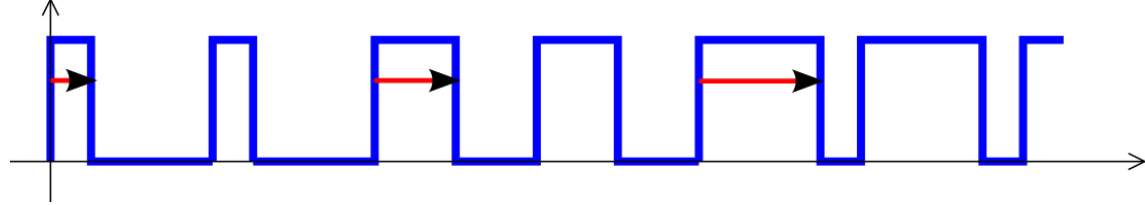
PWM réalisé avec des circuits logiques



C'est le Timer d'un microcontrôleur



PWM : Modulation de Largeur d'Impulsion

- Principe : 
- Fréquence : > 100 Hz pour l'oeil
- Programmer un PWM (occupe le proc.)
- Convertisseur DAC (+ filtre)
- Réalisation par des circuits logiques, inclus dans les microcontrôleurs