

# 📍 Étape 4a : La notion de fréquence

## Objectifs:

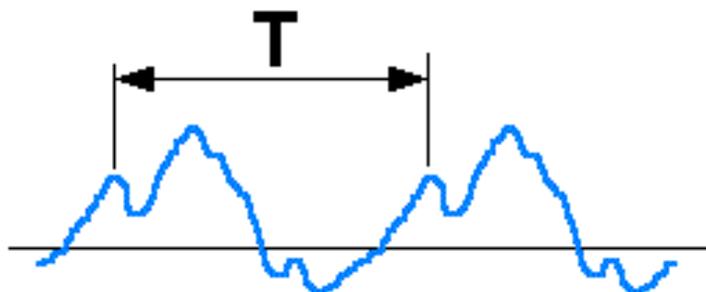
- ❑ Maîtriser la notion de fréquence

### Rappel : Notion de fréquence

On a parlé en page précédente de « fréquence » d'allumage. Il faut bien garder en tête, notamment pour ce qui suit que, dans notre cas, **plusieurs fréquences sont mises en jeu**.

Pour bien le comprendre, rappelons la définition propre de fréquence. Par définition, la fréquence (notée  $f$ ) d'un **phénomène périodique** correspond au nombre de répétitions de ce dernier pendant une seconde. La fréquence est donc associée à un **signal périodique**.

La notion de fréquence est étroitement liée à celle de **période**. En effet, on a  $f = \frac{1}{T}$ . Plus qualitativement, la période d'un signal périodique correspond à l'intervalle de temps entre deux répétitions successives d'un instant du signal.



# Étape 4b : Clignotement d'une LED

## Objectifs:

- Faire clignoter une LED
- Choisir une fréquence de clignotement

### Faire clignoter une LED

Pour faire clignoter une LED, c'est très simple : on va naturellement l'éteindre (état LOW) puis l'allumer, (état HIGH) en oubliant pas d'attendre entre les deux. Attention, la fonction delay prend un temps en ms

? Exercice pratique : écrire les 4 lignes à insérer dans le void loop pour faire clignoter la LED. On attendra **500ms** entre l'allumage et l'extinction.

```
1. const int LED = 11;
2.
3. void setup() { .....
4.     pinMode(LED, OUTPUT); .....
5.     Serial.begin(9600); .....
6. }
7. void loop() { .....
8.     ..... ←
9. }
```

? Question pratique : Si on s'intéresse au signal correspondant à l'état (allumée/éteinte) de la LED. Quelle est la **fréquence** de ce signal ?

.....  
.....

