

Utiliser un buzzer actif (KY-012)

Dans cet article, nous allons apprendre à utiliser un **buzzer actif KY-012** avec une Raspberry Pi Pico. L'objectif est de comprendre comment produire des bips sonores, gérer le rythme et utiliser le temps comme information, jusqu'à construire un **code Morse simplifié**.

4GMS

5GMS

4TTR



Découverte

Contrairement au buzzer passif, ici **la fréquence n'est pas contrôlable** : le buzzer actif ne sait produire **qu'une seule note**.

Rappel : principe du buzzer actif

Un buzzer actif contient un oscillateur interne. Dès qu'il est alimenté, il émet un son à une fréquence fixe.

CONSÉQUENCE DIRECTE

- Pas de PWM
- Pas de fréquence à choisir
- Uniquement ON / OFF

Côté code, on travaille uniquement avec :

- l'état de la broche
- la durée pendant laquelle elle reste active

Connexion du KY-012

- **S** (Signal) → GPIO15
- **-** (GND) → GND
- **+** (VCC) → 3.3 V

Contrairement au buzzer passif, le buzzer actif **doit être alimenté**.

Premier son : un bip simple

On commence par le cas le plus élémentaire : un bip, puis arrêt.

```
1 | from machine import Pin
2 | import utime
3 |
4 | buzzer = Pin(15, Pin.OUT)
5 |
6 | buzzer.value(1)    # buzzer ON
7 | utime.sleep(1)    # 1 seconde
8 | buzzer.value(0)    # buzzer OFF
```

À retenir

- `value(1)` alimente le buzzer
- `value(0)` coupe le son
- la durée dépend uniquement du `sleep`

Bip + pause en boucle

On introduit maintenant la répétition.

```
1 | from machine import Pin
2 | import utime
3 |
4 | buzzer = Pin(15, Pin.OUT)
5 |
6 | while True:
7 |     buzzer.value(1)
8 |     utime.sleep(0.5)
9 |
10 |     buzzer.value(0)
11 |     utime.sleep(0.5)
```

Le buzzer émet un bip régulier, comparable à un métronome.

Bip court et bip long

La différence entre un bip court et un bip long est **uniquement temporelle**.

```
1  from machine import Pin
2  import utime
3
4  buzzer = Pin(15, Pin.OUT)
5
6  while True:
7      # bip court
8      buzzer.value(1)
9      utime.sleep(0.2)
10     buzzer.value(0)
11     utime.sleep(0.3)
12
13     # bip long
14     buzzer.value(1)
15     utime.sleep(0.6)
16     buzzer.value(0)
17     utime.sleep(0.8)
```

Introduire la notion de rythme

À ce stade, on ne change jamais le son lui-même. On ne joue que sur :

- la durée du bip
- la durée du silence

Avec un buzzer actif, **LE RYTHME EST L'UNIQUE INFORMATION SONORE**.

C'est exactement le principe du code Morse.

Principe du Morse (version pédagogique)

Le code Morse repose sur :

- un son court
- un son long

- des pauses

Pour simplifier :

- point → bip court
- trait → bip long

Nous n'utilisons pas ici les durées normalisées strictes, mais une version **compréhensible et lisible**.

Dictionnaire Morse simplifié

On associe chaque lettre à une séquence de symboles.

```
1 | morse = {
2 |     'S': '...',
3 |     'O': '---',
4 |     'I': '.. ',
5 |     'N': '-. ',
6 |     'F': '...-'
7 | }
```

Jouer "SOS"

```
1 | from machine import Pin
2 | import utime
3 |
4 | buzzer = Pin(15, Pin.OUT)
5 |
6 | morse = {
7 |     'S': '...',
8 |     'O': '---'
9 | }
10 |
11 | dot = 0.2
12 | dash = 0.6
13 | pause = 0.3
14 |
15 | while True:
16 |     for letter in "SOS":
17 |         for symbol in morse[letter]:
18 |             buzzer.value(1)
19 |             if symbol == '.':
20 |                 utime.sleep(dot)
```

```
21         else:
22             utime.sleep(dash)
23
24         buzzer.value(0)
25         utime.sleep(pause)
26
27         utime.sleep(0.6) # pause entre lettres
```

Jouer “INFO”

```
1 morse = {
2     'I': '...',
3     'N': '-.',
4     'F': '...-',
5     'O': '----'
6 }
7
8 message = "INFO"
```

Le reste du code est identique : seul le dictionnaire et le message changent.

Ce que cet article a introduit

- pilotage ON / OFF
- gestion du temps
- notion de rythme
- dictionnaire comme table de correspondance
- boucle imbriquée

Erreurs fréquentes

- Essayer de modifier la fréquence d'un buzzer actif

- Oublier la pause entre lettres
- Alimenter le buzzer actif sans GND commun

Ce qu'il faut retenir

- Un buzzer actif ne produit qu'un seul son
- Toute l'information passe par la durée
- Le Morse est une application directe de cette logique
- Simplicité matérielle ≠ pauvreté pédagogique

Dans le prochain article, nous utiliserons un **buzzer passif (KY-006)** pour introduire la fréquence, le volume et la génération de sons évolutifs, jusqu'à des mélodies complètes.