

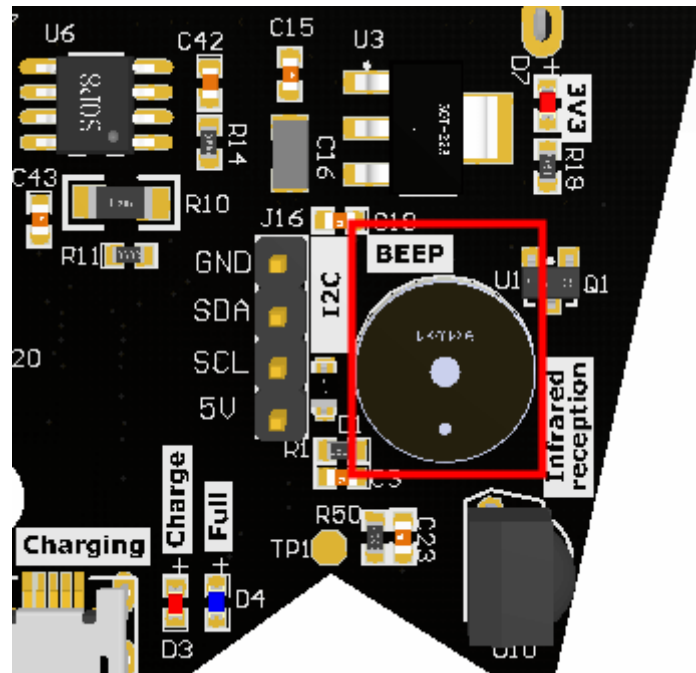
## 3.3 Passive buzzer

### I. Learning objectives

1. Learn to combine the Raspberry Pi Pico 2/Pico mainboard and the car expansion board passive buzzer for experiments.
2. Understand the use of active buzzers.

### II. Hardware usage

This course uses Pico 2/Pico mainboard and the car board passive buzzer



Passive buzzers use electromagnetic induction to attract or repel the permanent magnet formed by the voice coil after the alternating current is connected to the voice coil, and the diaphragm is pushed to make sound. When connected to direct current, it can only continuously push the diaphragm but cannot produce sound. In control, we generally use PWM to control the passive buzzer to make sound.

### 3. Program Analysis

Code path: Code -> 1.Basic course -> 3. Passive buzzer.py

```
from machine import Pin, PWM
import time
# set buzzer pin
BZ = PWM(Pin(22))
BZ.freq(1000)
# Initialize music
CM = [0, 330, 350, 393, 441, 495, 556, 624]
song =
[CM[1],CM[1],CM[5],CM[5],CM[6],CM[6],CM[5],CM[4],CM[4],CM[3],CM[3],CM[2],CM[2],C
M[1],]
beat = [ 0.5,0.5,0.5,0.5,0.5,0.5,1,0.5,0.5,0.5,0.5,0.5,0.5,1,]
# music
def music():
    print('Playing song ...')
    for i in range(len(song)):
```

```
BZ.duty_u16(500)
BZ.freq(song[i])
time.sleep(beat[i])
BZ.duty_u16(0)
time.sleep(0.01)

# play music
music()
print("Ending")
```

### **from machine import Pin, PWM**

The machine library contains all the instructions MicroPython needs to communicate with Pico and other MicroPython-compatible devices, extending the language of physical computing. Only the Pin and PWM libraries are used here.

### **import time**

The "time" library. This library handles everything related to time, from measuring it to inserting delays into the program. The unit is seconds.

### **BZ = PWM(Pin(22))**

Set IO22 to the PWM output pin for controlling the buzzer.

### **BZ.freq(1000)**

Set the PWM frequency to 1000.

### **BZ.duty\_u16(0)**

Turn off the sound when the value is 0, and turn on the sound when it is 500.

### **music()**

By calling the music() function, a for loop is used in the function to play the pre-written sounds of different frequencies one by one, so as to achieve music playback.

### **Fourth, Experimental Phenomenon**

After the program is downloaded, we can hear the buzzer playing the music Little Star, and after the playback, the Shell displays Ending.