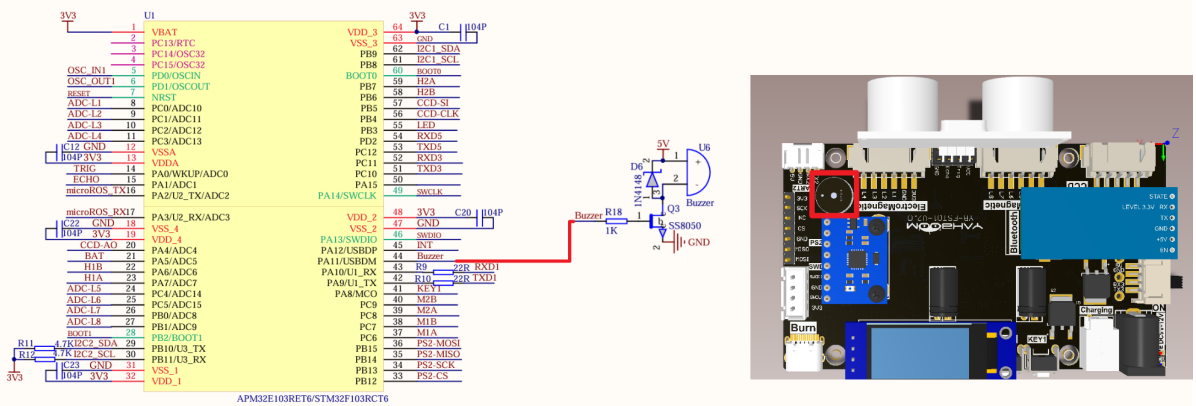
Buzzer control

Tutorial on controlling the buzzer to sound.

```
The tutorial only introduces the standard library project code
```

Hardware connection



Peripherals	Development board
Buzzer (active buzzer)	PA11

Control principle

Buzzer	Features
Active buzzer	Internal oscillation source, high and low levels can control the sound, the sound frequency is fixed
Passive buzzer	Internal oscillation source is not provided, different oscillation pulse frequencies can be provided to emit different frequency sounds

Control pin	Output level	LED effect
PA11	Output low level	Buzzer does not sound
PA11	Output high level	Buzzer sounds

Software configuration

Pin definition

Main control chip	Pin	Main function (after reset)	Default multiplexing function	Redefine function
STM32F103RCT6	PA11	PA11	USART1_CTS/USBDM/CAN_RX/TIM1_CH4	

Software code

The default function of the PA11 pin is a normal IO pin function.

```
Product supporting data source code path: Attachment → Source code summary → 1.Base_Course → 3.Beep
```

Control function

The tutorial only briefly introduces the code, you can open the project source code to read it.

> **Init_Beep**

```
void Init_Beep(void)
{
GPIO_InitTypeDef GPIO_InitStructure;
RCC_APB2PeriphClockCmd(BEEP_RCC, ENABLE);
GPIO_InitStructure.GPIO_Pin = BEEP_PIN;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz;
GPIO_Init(BEEP_PORT, &GPIO_InitStructure);
BEEP_BEEP = 0;
}
```

Experimental phenomenon

The BEEP.hex file generated by the project compilation is located in the OBJ folder of the Beep project. Find the BEEP.hex file corresponding to the project and use the FlyMcu software to download the program into the development board.

After the program is downloaded successfully, the buzzer switches its sounding state every 500ms.