**Microcontroller Experiment**

The Buzzer Experiment

Name: Han Yichen 韩一尘

Student ID: 22722051

Date: 2023/11/1

**1.Experimental principle**

Buzzer is an integrated structure of electronic buzzer, mainly divided into piezoelectric buzzer and electromagnetic buzzer two types.

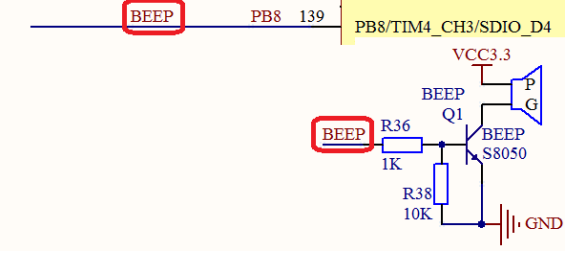
The STM32's single IO can provide a maximum of 25mA of current, while the buzzer's drive current is about 30mA, both.It is very similar, but overall consideration, the current of the STM32 entire chip, the maximum is 150mA, if the IO port is used to directly drive the buzzer, the power consumption in other places is reduced. Therefore, we do not use the IO of STM32 to drive the buzzer directly, but to drive the buzzer after expanding the current through the transistor, so that the IO of STM32 only needs to provide less than 1mA of current.

The hardware required for this chapter is:

1) Indicator DS0

2) Buzzer

The drive signal of the buzzer is connected to the PB8 of the STM32.



**2. Main program analysis**

#include "beep.h"

//Initialize PB8 as the output. And enable this mouth clock

//LED IO initialize

void BEEP\_Init(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOB, ENABLE);

//Example Enable the GPIOB port clock

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_8;

//BEEP-->GPIOB.8 configuration of port

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

// Push-pull output GPIO\_InitStructure.

GPIO\_Speed = GPIO\_Speed\_50MHz;

// The speed is 50MHz GPIO\_Init(GPIOB, &GPIO\_InitStructure);

GPIO\_ResetBits(GPIOB,GPIO\_Pin\_8); // Initialize gPIob. 8.

// If the output is 0, disable the buzzer output}

#ifndef \_\_BEEP\_H

#define \_\_BEEP\_H

#include "sys.h"

//Buzzer port definition

#define BEEP PBout(8)

// BEEP,Buzzer port

void BEEP\_Init(void);

#endif

Through the bitband operation to achieve the output control of a certain IO port, BEEP is directly substituted

The output state of PB8 is shown. We just need to set BEEP=1 to make the buzzer sound.

int main(void)

{

delay\_init();

//Delay function initialization

LED\_Init();

//Initializes the hardware interface connected to the LED

BEEP\_Init(); //Example Initialize the buzzer port

while(1)

{ LED0=0;

BEEP=0;

delay\_ms(300);

LED0=1;

BEEP=1;

delay\_ms(300);

}

}

**3.Experimental result**

In this experiment, the DSo and buzzer on the development board are controlled by code. The DSo flashes, prompting the program to run, and the buzzer makes a periodic sound of "beep" at an interval of 0.3 seconds.

**4. Improvement and perfection**

By modifying the sound interval of the buzzer, you can control the different sound frequencies of the buzzer.