2020 －2021 学年第 1 学期

编号：

****

### 实 验 报 告



实验课程名称 《嵌入式系统设计基础》

专 业 班 级 电子信息工程1901

学 生 姓 号 31902024、31902002

学 生 姓 名 盛楚迪、曾瑶瑶

实验指导老师 朱胜

**浙大城市学院实验报告**

课程名称 《嵌入式系统设计基础》

实验项目名称 《实验五》《实验六》《实验七》

学生姓名 盛楚迪、曾瑶瑶

专业班级 电子信息工程1901

学号 31902024、31902002

实验成绩 指导老师（签名 ） 日期

目录

浙大城市学院实验报告

实验五（定时器中断）

实验六（PWM输出）

实验七（PWM输入捕获）

# 实验五（定时器中断）

## 一. 实验目的和要求

了解使用 STM32F4 的通用定时器，STM32F4 的定时器功能十

分强大，有TIME1和TIME8等高级定时器，也有 TIME2~TIME5，TIM9~TIM14 等通用定时器，还有 TIME6 和 TIME7 等基本定时器，总共达 14 个定时器之多；

## 二．实验内容

①：定时器1实现LED1每隔1s闪烁一次

②：定时器2实验蜂鸣器每隔2s响一次

③：屏幕循环输出：

“正在进行定时器中断实验

LED1隔1s闪烁

蜂鸣器隔2s响”

## 四. 实验代码

#include "system.h"

#include "SysTick.h"

#include "led.h"

#include "time.h"

#include "beep.h"

#include "usart.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : main

\* 函数功能 : 主函数

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

SysTick\_Init(168);

LED\_Init();

BEEP\_Init();

USART1\_Init(115200);

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2); //中断优先级分组 分2组

TIM3\_Init(10000-1,8400-1);

TIM4\_Init(20000-1,8400-1);

while(1)

{

printf("正在进行定时器中断实验，LED1隔1s闪烁，蜂鸣器隔2s响\n");

delay\_ms(2000);

}

}

直接对管脚进行高低电平的设置，进行定时设置，不适用延时。

TIM3\_Init(10000-1,8400-1); //定时器时钟84MHz 分频系数8400 计10000次为1s

TIM4\_Init(20000-1,8400-1);

LED1隔1s闪烁，蜂鸣器隔2s响

printf("正在进行定时器中断实验，LED1隔1s闪烁，蜂鸣器隔2s响\n");

delay\_ms(2000);

分别每隔2S的时间进行一次打印

## 五. 实验结果与分析

①：定时器1实现LED1每隔1s闪烁一次

②：定时器2实验蜂鸣器每隔2s响一次

③：屏幕循环输出：

“正在进行定时器中断实验

LED1隔1s闪烁

蜂鸣器隔2s响”

对于中断的设置，一定要清楚晶振的大小，以确定一个机器的周期。对于一开始没有了解时间的公式乱改数据，导致实验结果有很大的偏差。每个机器的的不同导致开始设置的时间是不同的。

## 六. 讨论、心得

了解使用 STM32F4 的通用定时器，STM32F4 的定时器功能十

分强大，有TIME1和TIME8等高级定时器，也有 TIME2~TIME5，TIM9~TIM14 等通用定时器，还有 TIME6 和 TIME7 等基本定时器，总共达 14 个定时器之多；

# 实验六（PWM输出）

## 一. 实验目的和要求

了解使用 STM32F4 的通用定时器；

使用 STM32F4 的 TIM4 来产生 PWM 输出；

## 二．实验内容

呼吸灯：用两个定时器控制两个LED灯实现呼吸灯功能要求一路频率为2KHZ,另一路为3KHZ。

## 三. 实验代码

int main()

{

u16 i;

u8 fx=0;

u16 a;

u8 fy=0;

SysTick\_Init(168);

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2); //中断优先级分组 分2组

LED\_Init();

BEEP\_Init();

TIM14\_CH1\_PWM\_Init(500-1,84-1);

//时钟频率84Mhz 分频系数84 计数频率1mhz 重装载值500 pwm频率1/500=2khz

TIM13\_CH1\_PWM\_Init(333-1,84-1);

while(1)

{

if(fx==0)

{

i++;

if(i==300)

{

fx=1;

}

}

else

{

i--;

if(i==0)

{

fx=0;

}

}

if(fy==0)

{

a++;

if(a==400)

{

fy=1;

}

}

else

{

a--;

if(a==0)

{

fy=0;

}

}

TIM\_SetCompare1(TIM14,i); //i值最大可以取499，因为ARR最大值是499.

TIM\_SetCompare1(TIM13,a);

delay\_ms(10);

}

}

## 五. 实验结果与分析

呼吸灯：用两个定时器控制两个LED灯实现呼吸灯功能要求一路频率为2KHZ,另一路为3KHZ。

直接对管脚进行高低电平的设置，进行定时设置，不适用延时。

TIM14\_CH1\_PWM\_Init(500-1,84-1);

//时钟频率84Mhz 分频系数84 计数频率1mhz 重装载值500 pwm频率1/500=2khz

TIM13\_CH1\_PWM\_Init(333-1,84-1);

//时钟频率84Mhz 分频系数84 计数频率1mhz 重装载值333 pwm频率1/333=2khz

呼吸灯和蜂鸣器的频率已知，要根据时钟频率和计数频率设置参数，使得两路不同的频率得到配置。

## 六. 讨论、心得

分强大，有TIME1和TIME8等高级定时器，也有 TIME2~TIME5，TIM9~TIM14 等通用定时器，还有 TIME6 和 TIME7 等基本定时器，总共达 14 个定时器之多；

了解使用 STM32F4 的通用定时器；使用 STM32F4 的 TIM3 来产生 PWM 输出；

# 实验七（PWM输入捕获）

## 一. 实验目的和要求

了解使用 STM32F4 的通用定时器；

通用定时器作为输入捕获的使用。在本章中，我们将用 TIM5 的通道1（PA0）来做输入捕获，捕获 PA0 上高电平的脉宽（用 KEY\_UP 按键输入高电平），通过串口打印高电平脉宽时间；

## 二．实验内容

呼吸灯：使用输入捕获测量PWM输出的信号高电平脉宽。

需要使用一根线讲PWM输出管脚与PAO输入捕获通道连接。

## 四. 实验代码

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* 普中科技

--------------------------------------------------------------------------------

\* 实 验 名 : 输入捕获实验

\* 实验说明 :

\* 连接方式 :

\* 注 意 : 输入捕获驱动在input.c内

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "system.h"

#include "SysTick.h"

#include "led.h"

#include "usart.h"

#include "input.h"

#include "pwm.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : main

\* 函数功能 : 主函数

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

u16 i=0;

u32 indata=0;

u8 fx=0;

SysTick\_Init(168);

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2); //中断优先级分组 分2组

LED\_Init();

USART1\_Init(115200);

// 输入频率

TIM5\_CH1\_Input\_Init(0xffff,83); //以1M频率计数

// 输出频率

TIM14\_CH1\_PWM\_Init(1000-1,84-1); //频率是2Kh

while(1)

{

// 输出

if(fx==0)

{

i++;

if(i==300)

{

fx=1;

}

}

else

{

i--;

if(i==0)

{

fx=0;

}

}

TIM\_SetCompare1(TIM14,i); //i值最大可以取499，因为ARR最大值是499.

// 捕获

if(TIM5\_CH1\_CAPTURE\_STA&0x80) //成功捕获

{

// 计数满的次数

indata=TIM5\_CH1\_CAPTURE\_STA&0x3f;

// 溢出大的时间总和

indata\*=0xffff; //溢出次数乘以一次的计数次数时间 us

// 加上小的时间

indata+=TIM5\_CH1\_CAPTURE\_VAL;//加上高电平捕获的时间

// 串口输出

printf("高电平持续时间: %d us\r\n",indata); //总的高电平时间

TIM5\_CH1\_CAPTURE\_STA=0; //开始下一次捕获

}

delay\_ms(10);

}

}

## 五. 实验结果与分析

使用输入捕获测量PWM输出的信号高电平脉宽。需要使用一根线讲PWM输出管脚与PAO输入捕获通道连接。

直接对管脚进行高低电平的设置，进行定时设置，不适用延时。

呼吸灯和蜂鸣器的频率已知，要根据时钟频率和计数频率设置参数，使得两路不同的频率得到配置。

## 六. 讨论、心得

了解使用 STM32F4 的通用定时器，STM32F4 的定时器功能十

分强大，有TIME1和TIME8等高级定时器，也有 TIME2~TIME5，TIM9~TIM14 等通用定时器，还有 TIME6 和 TIME7 等基本定时器，总共达 14 个定时器之多；

了解使用 STM32F4 的通用定时器；使用 STM32F4 的 TIM3 来产生 PWM 输出；

使用输入捕获测量PWM输出的信号高电平脉宽。需要使用一根线讲PWM输出管脚与PAO输入捕获通道连接。