嵌入式系统实验

——STM32开发板实验

浙江大学机械工程学院 实验教学中心

实验报告

报告内容：实验1 至 实验7

报告分值：20%

上交形式：一个PDF文档，上传至“学在浙大”

截止时间：2025年6月8日 24点前

嵌入式系统综合实验考核办法

1、实验老师提供考核题目，学生上机操作。学生现场编程并展示结果。实验老师登记学生编程结果。

2、考核过程全程开卷，学生仅能带教材及实验指导书，不允许夹带其他资料，不允许U盘拷贝相关电子资料，不允许交头接耳。

3、考核过程使用西四A422实验室主机（不允许联网），不允许自带电脑（平板），不允许使用手机。

4、请各位同学确认自己的实验课程时间安排。

题目类型

基础题（40%）

考查同学们关于stm32的基础概念及基础编程能力

出题范围为实验指导书中的实验1~实验5的相关知识点

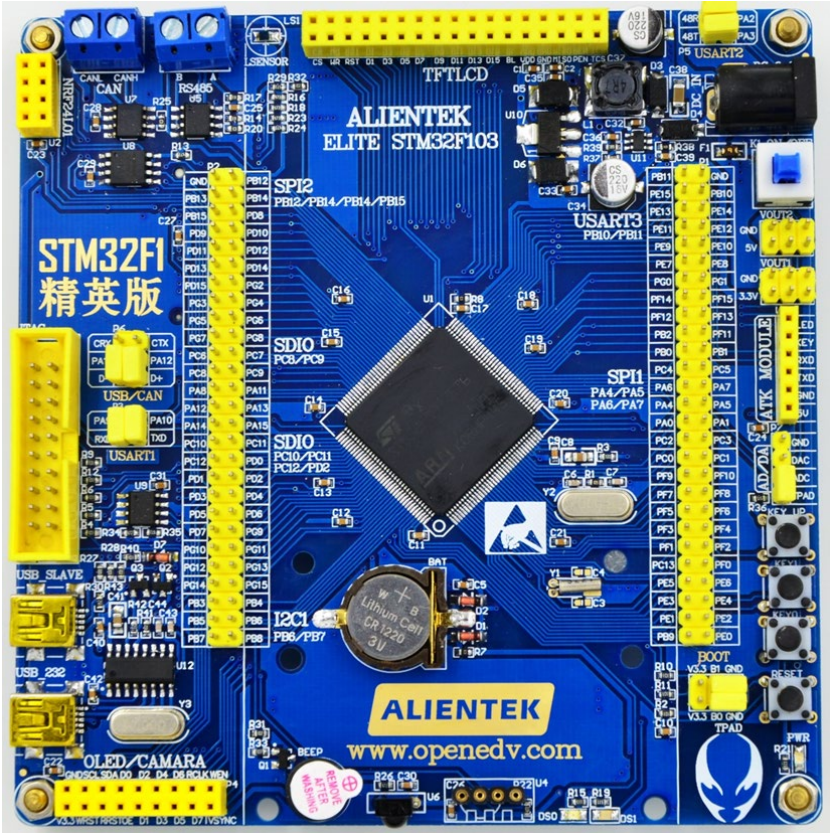
综合题（20%）

考查同学们关于stm32的综合编程能力

出题范围为实验指导书中的实验1~实验8的相关知识点

第 3 次实验

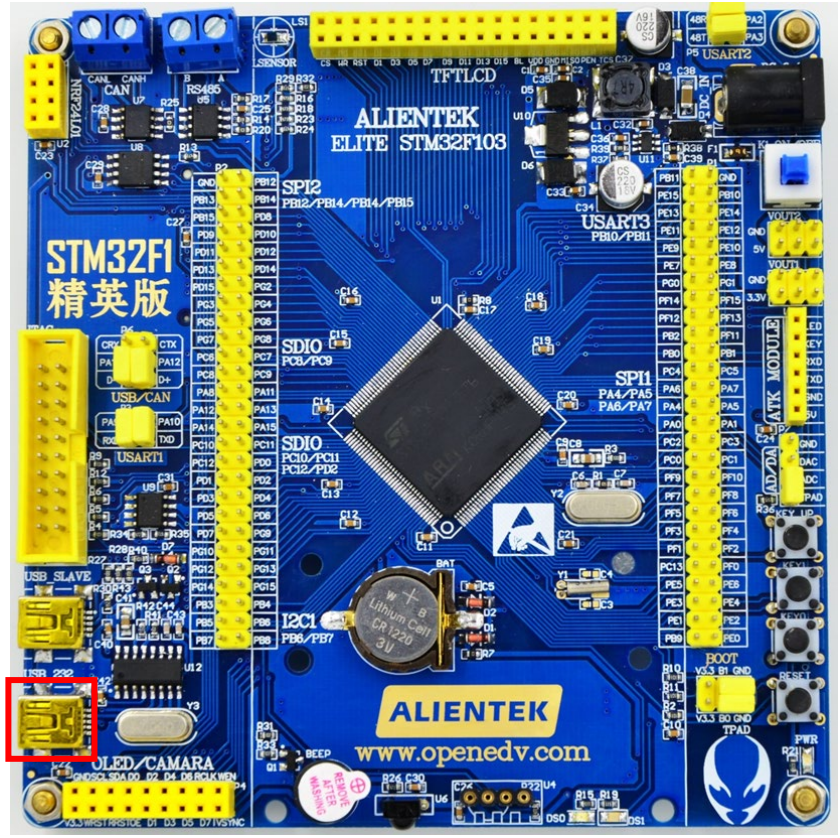
- **1. 串口通讯实验（无中断）**
- 2. 串口通讯实验（串口中断）
- 3. 定时器中断实验
- 4. 串口通讯实验（按键中断）
- 5. 定时器计时实验



STM32F103ZET6 开发板

1.串口通讯实验（无中断）

芯片引脚

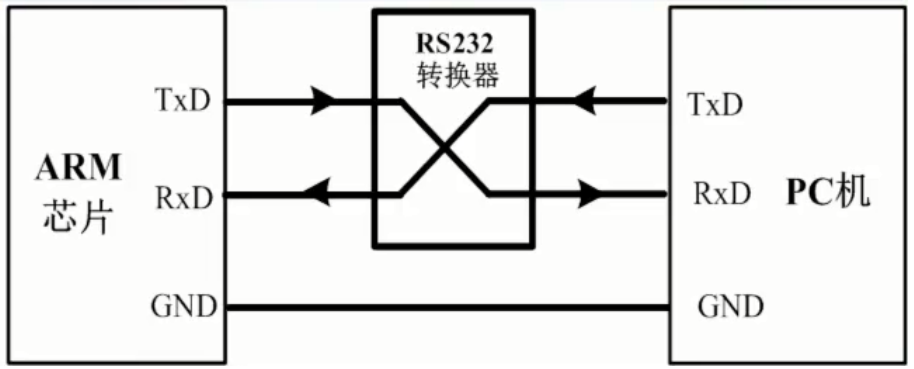


对应芯片引脚

USART1

1.串口通讯实验（无中断）

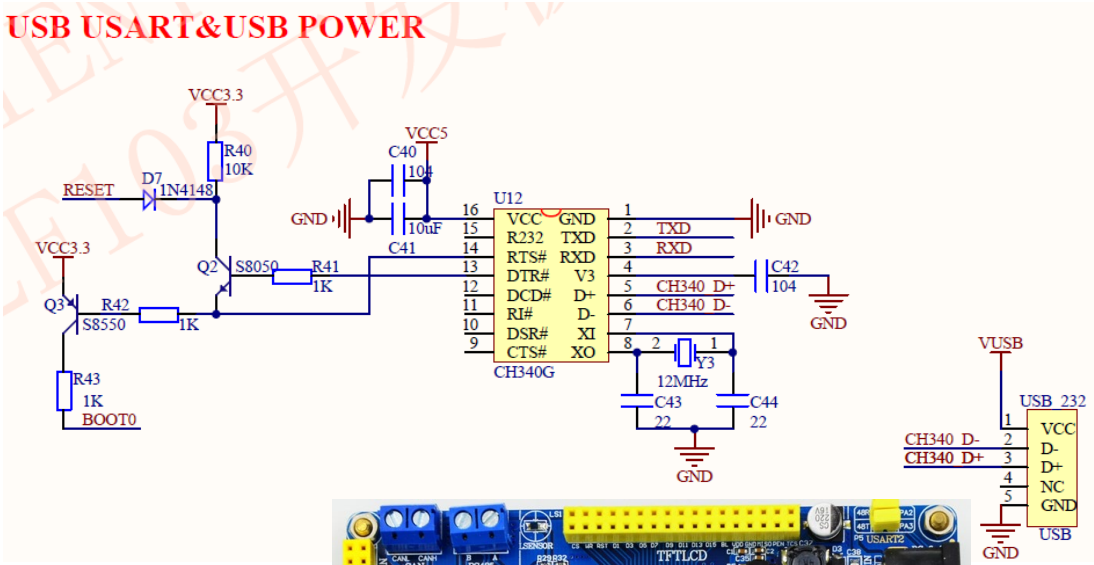
串口引脚

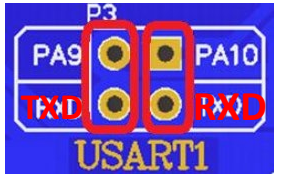


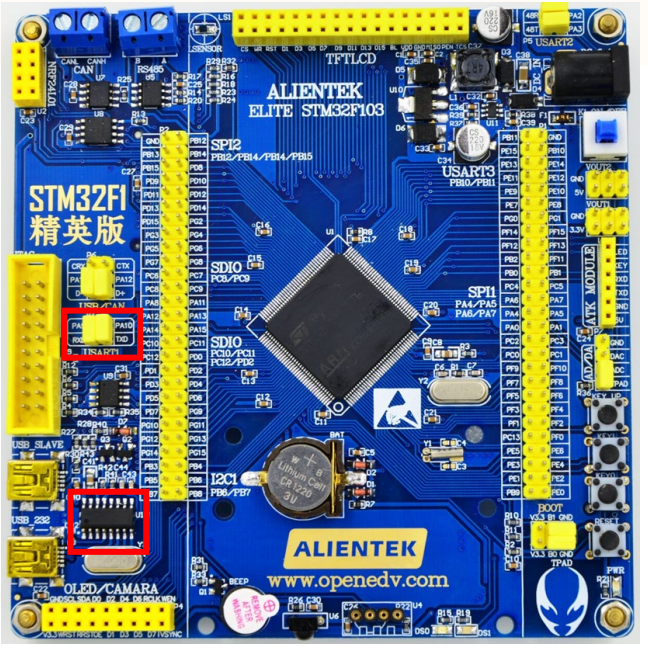
RXD：数据输入引脚，用于数据接收。

TXD：数据发送引脚，用于数据发送。

USART引脚	配置	GPIO配置
USARTx_TX	全双工模式	复用推挽输出
	半双工同步模式	复用推挽输出
USARTx_RX	全双工模式	浮空输入或带上拉输入
	半双工同步模式	未用，可作为通用I/O







任务

初始化串口通讯

接收上位机发送的字符

并将接收到的字符发送出去

1.串口通讯实验（无中断）

任务分解

1. 使能串口和GPIO时钟
   - RCC_APB2PeriphClockCmd
2. 复位串口
   - USART_DeInit
3. 初始化GPIO
   - GPIO_InitTypeDef
   - GPIO_Init

1.串口通讯实验（无中断）

任务分解

4 初始化串口

USART_InitTypeDef
USART_Init

5 使能串口

USART_Cmd

6 接收并发送字符

USART_ReceiveData
USART_SendData

1.串口通讯实验（无中断）

结构体：USART_InitTypeDef

```
typedef struct
{
  uint32_t USART_BaudRate;            /* 串口波特率 */

  uint16_t USART_WordLength;          /* 发送的字长 */

  uint16_t USART_StopBits;            /* 串口停止位 */

  uint16_t USART_Parity;              /* 奇偶校验设置 */

  uint16_t USART_Mode;                /* 串口收发模式 */

  uint16_t USART_HardwareFlowControl;      /* 串口硬件流控制 */

} USART_InitTypeDef;
```

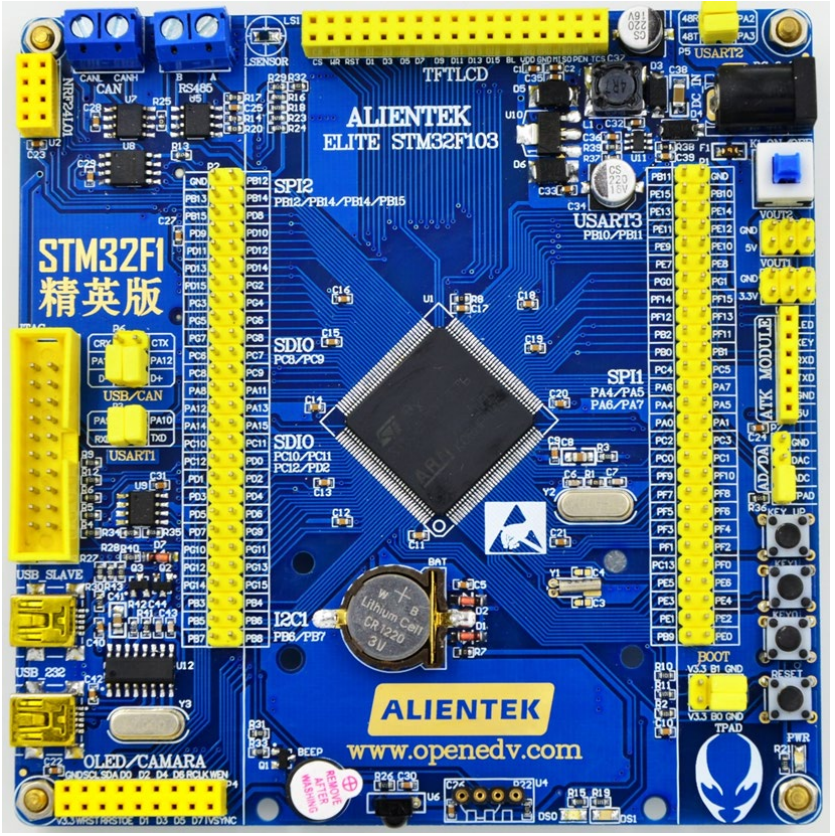
1.串口通讯实验（无中断）

初始化串口：

```
USART_InitTypeDef USART_InitStrue;
USART_InitStrue.USART_BaudRate=115200;
USART_InitStrue.USART_WordLength=USART_WordLength_8b;
USART_InitStrue.USART_StopBits=USART_StopBits_1;
USART_InitStrue.USART_Parity=USART_Parity_No;
USART_InitStrue.USART_HardwareFlowControl=USART_HardwareFlowControl_None;
USART_InitStrue.USART_Mode=USART_Mode_Tx|USART_Mode_Rx;
USART_Init(USART1,&USART_InitStrue);
```

第 3 次实验

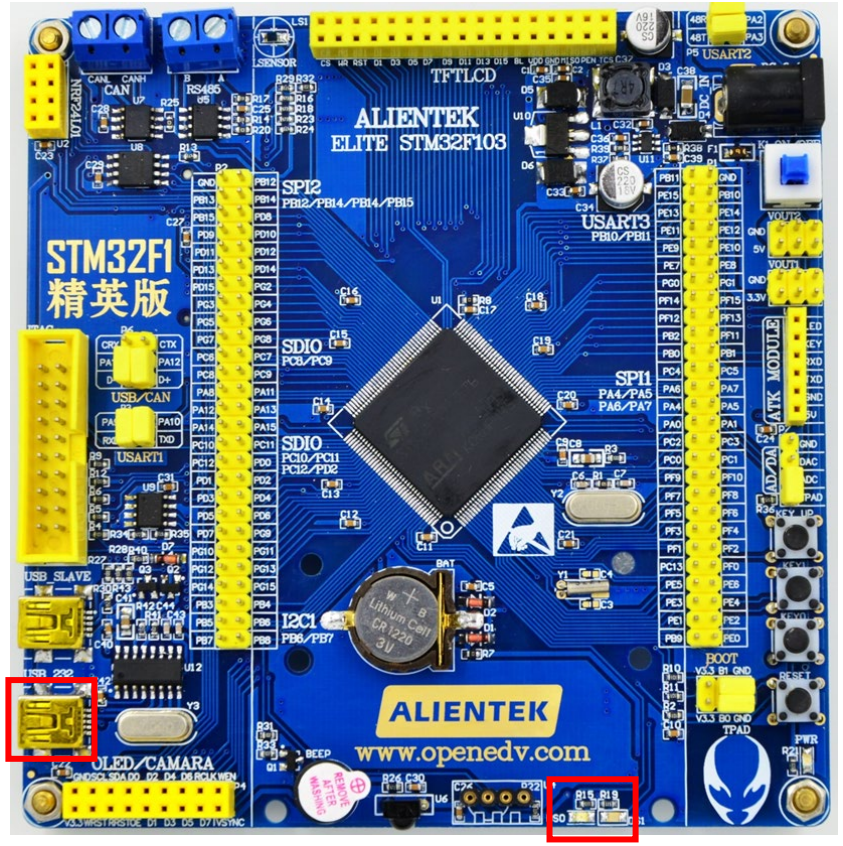
- 1. 串口通讯实验（无中断）
- **2. 串口通讯实验（串口中断）**
- 3. 定时器中断实验
- 4. 串口通讯实验（按键中断）
- 5. 定时器计时实验



STM32F103ZET6 开发板

2.串口通讯实验（串口中断）

芯片引脚



对应芯片引脚

PB5 → LED0

PE5 → LED1

USART1

任务

初始化串口**中断**

接收上位机发送的字符

并将接收到的字符发送出去

LED0间隔0.2秒闪烁一次

2.串口通讯实验（串口中断）

任务分解

1. 使能串口和GPIO时钟
   - RCC_APB2PeriphClockCmd
2. 复位串口
   - USART_DeInit
3. 初始化GPIO
   - GPIO_InitTypeDef
   - GPIO_Init

2.串口通讯实验（串口中断）

任务分解

4 初始化串口

USART_InitTypeDef
USART_Init

5 初始化中断

NVIC_InitTypeDef
NVIC_Init

6 使能中断

USART_ITConfig

2.串口通讯实验（串口中断）

任务分解

7 使能串口

USART_Cmd

8 中断服务函数

USARTx_IRQHandler

2.串口通讯实验（串口中断）

结构体：NVIC_InitTypeDef

```
typedef struct
{
  uint8_t NVIC_IRQChannel;                    /* 中断通道 */

  uint8_t NVIC_IRQChannelPreemptionPriority;  /* 抢占优先级（主优先级）*/

  uint8_t NVIC_IRQChannelSubPriority;         /* 唤醒优先级（子优先级）*/

  FunctionalState NVIC_IRQChannelCmd;         /* 中断使能*/

} NVIC_InitTypeDef;
```

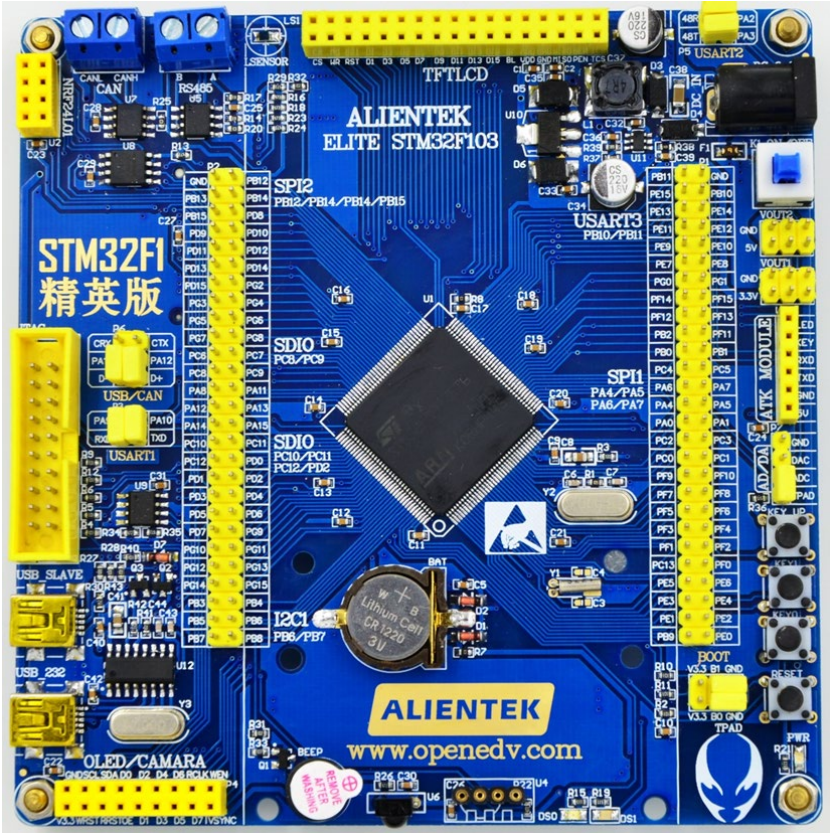
2.串口通讯实验（串口中断）

串口中断服务函数： USARTx_IRQHandler

```
void USARTx_IRQHandler(void)
{
    //检查是否触发串口接收中断
    if(USART_GetITStatus(USARTx,USART_IT_RXNE))
    {
        /* 中断执行程序 */
        USART_ClearITPendingBit(USARTx,USART_IT_RXNE); //清除USARTx更新中断标志
    }
}
```

第 3 次实验

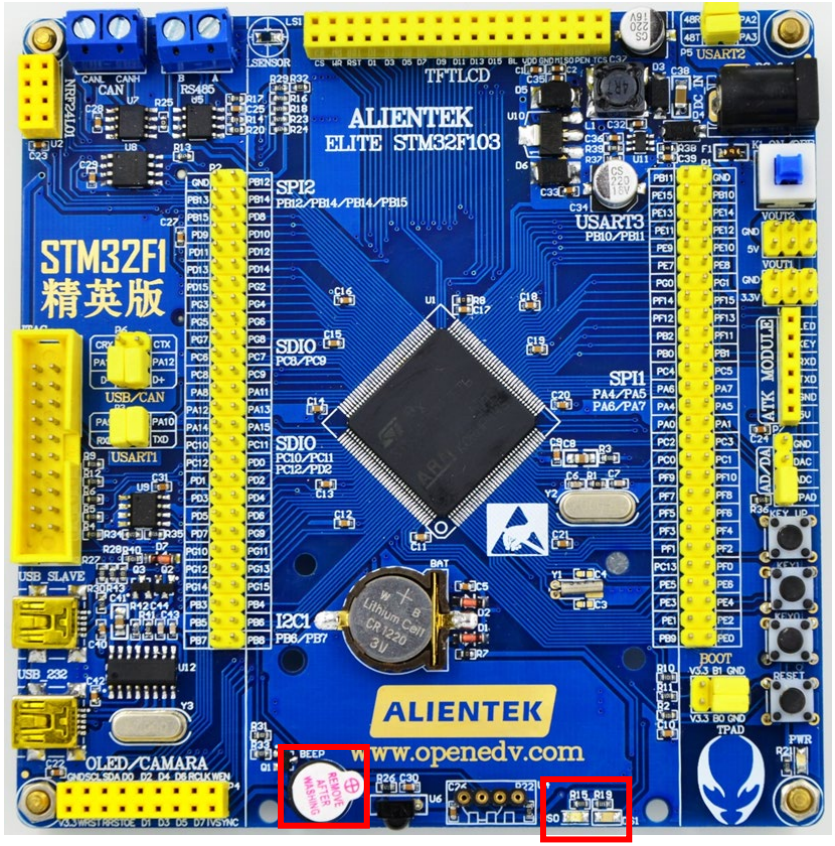
- 1. 串口通讯实验（无中断）
- 2. 串口通讯实验（串口中断）
- **3. 定时器中断实验**
- 4. 串口通讯实验（按键中断）
- 5. 定时器计时实验



STM32F103ZET6 开发板

3. 定时器中断实验

芯片引脚



BEEP　　LED0 / LED1

对应芯片引脚

PB5 → LED0

PE5 → LED1

PB8 → BEEP

任务

设置定时器中断，周期为1秒；

触发定时器中断后，反转LED1和BEEP的状态：当LED1熄灭时，BEEP响；当LED1点亮时，BEEP不响；

LED0间隔0.2秒闪烁一次。

3. 定时器中断实验

任务分解

1. 使能定时器时钟
   RCC_APB1PeriphClockCmd
2. 初始化定时器
   TIM_TimeBaseInitTypeDef
   TIM_TimeBaseInit
3. 开启定时器中断
   TIM_ITConfig

3. 定时器中断实验

任务分解

4 配置NVIC

NVIC_InitTypeDef
NVIC_Init

5 使能定时器

TIM_Cmd

6 编写中断服务函数

TIMx_IRQHandler

3. 定时器中断实验

结构体：TIM_TimeBaseInitTypeDef

```
typedef struct
{
  uint16_t TIM_Prescaler;          /* 分频系数 */

  uint16_t TIM_CounterMode;        /* 计数方式，向上计数或向下计数 */

  uint16_t TIM_Period;             /* 自动重载计数周期值 */

  uint16_t TIM_ClockDivision;      /* 时钟分频因子 */

  uint8_t TIM_RepetitionCounter;   /* 指定重复计数器的值 */

} TIM_TimeBaseInitTypeDef;
```

3. 定时器中断实验

初始化定时器：Tout=((arr+1)*(psc+1))/Tclk

```
TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
TIM_TimeBaseStructure.TIM_Period = arr;
TIM_TimeBaseStructure.TIM_Prescaler =psc;
TIM_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1;
TIM_TimeBaseStruc ture.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIMx, &TIM_TimeBaseStructure);
```

3. 定时器中断实验

结构体：NVIC_InitTypeDef

```
typedef struct
{
  uint8_t NVIC_IRQChannel;                    /* 中断通道 */

  uint8_t NVIC_IRQChannelPreemptionPriority;  /* 抢占优先级（主优先级）*/

  uint8_t NVIC_IRQChannelSubPriority;         /* 唤醒优先级（子优先级）*/

  FunctionalState NVIC_IRQChannelCmd;         /* 中断使能*/

} NVIC_InitTypeDef;
```

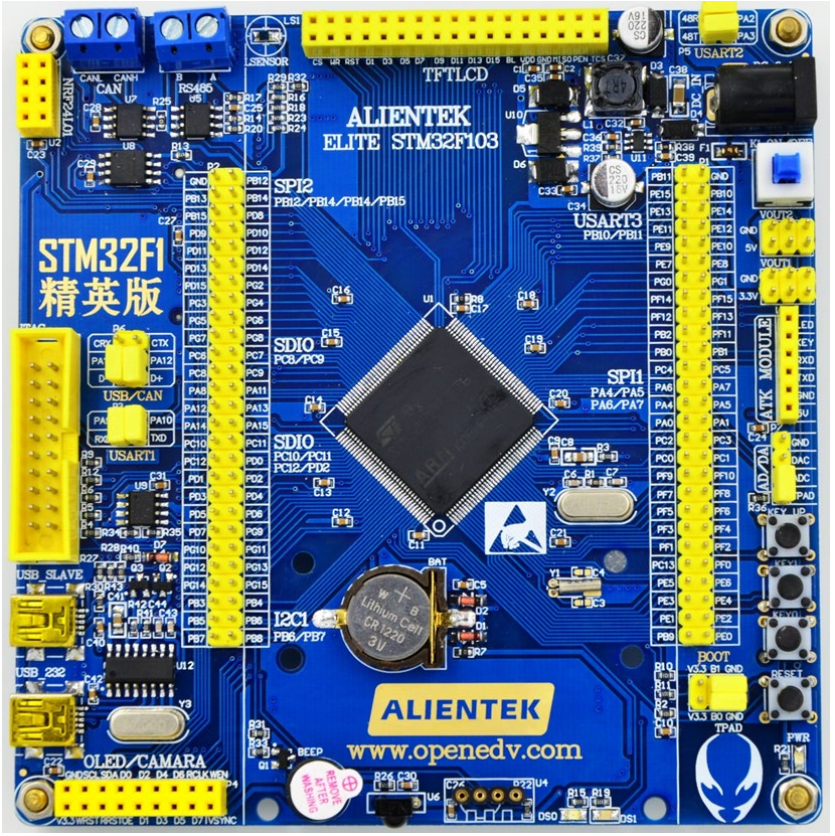
3. 定时器中断实验

定时器中断服务函数： TIMx_IRQHandler

```
void TIMx_IRQHandler(void)
{
    //检查是否发送定时器更新中断
    if (TIM_GetITStatus(TIMx, TIM_IT_Update) != RESET)
    {
        /* 中断执行程序 */
        TIM_ClearITPendingBit(TIMx, TIM_IT_Update);  //清除TIMx更新中断标志
    }
}
```

第 3 次实验

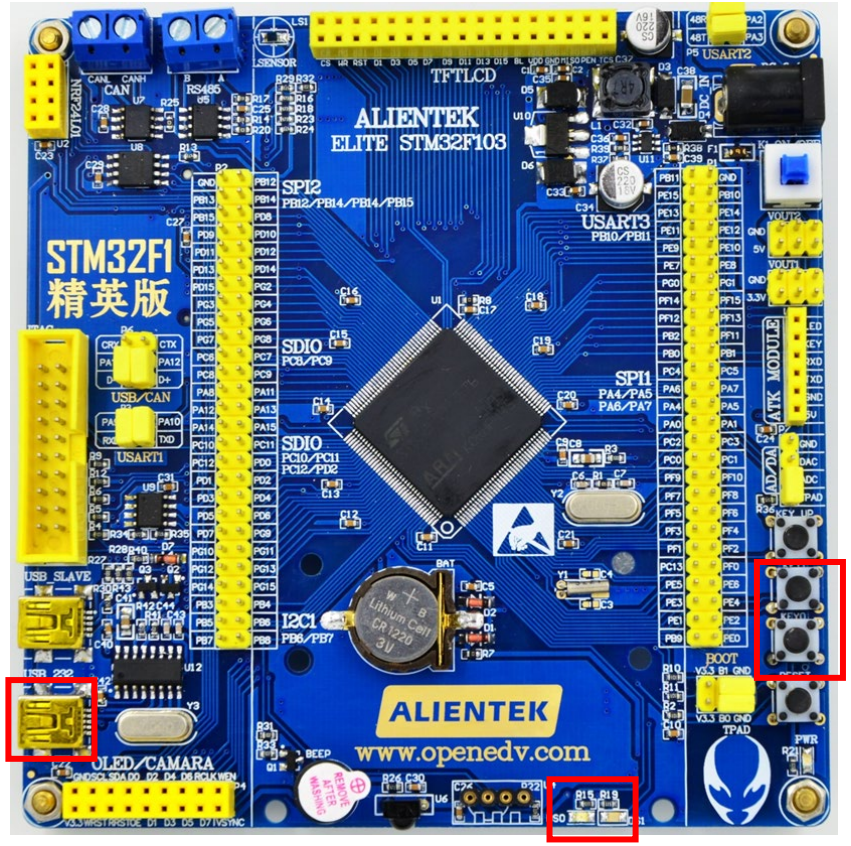
- 1. 串口通讯实验（无中断）
- 2. 串口通讯实验（串口中断）
- 3. 定时器中断实验
- **4. 串口通讯实验（按键中断）**
- 5. 定时器计时实验



STM32F103ZET6 开发板

4. 串口通讯实验（按键中断）

芯片引脚



USART1

KEY1

KEY0

LED0 / LED1

对应芯片引脚

PB5 → LED0

PE5 → LED1

PE3 → KEY1

PE4 → KEY0

USART1

任务

按下KEY1后，触发中断发送当前系统日期（字符串1）

按下KEY0后，触发中断发送当前系统时间（字符串2）

LED0间隔0.2秒闪烁一次。

4. 串口通讯实验（按键中断）

任务分解

1. 初始化串口 — USART_Init
2. 初始化LED — LED_Init
3. 初始化KEY — KEY_Init

4. 串口通讯实验（按键中断）

任务分解

4 初始化按键中断

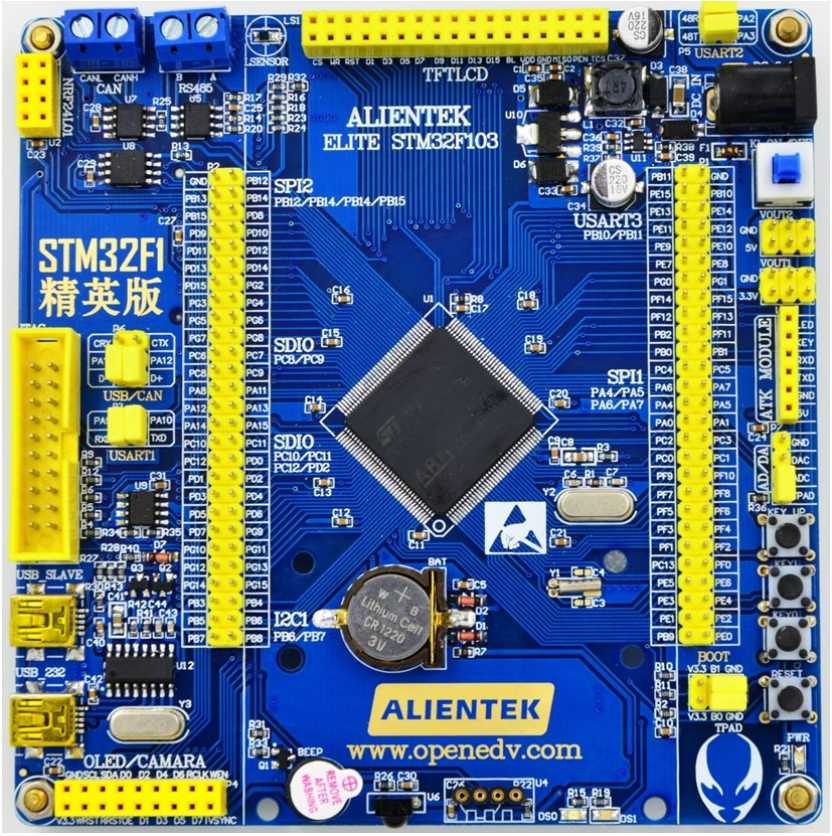
EXTIX_Init

5 编写中断服务函数

EXTIx_IRQHandler

第 3 次实验

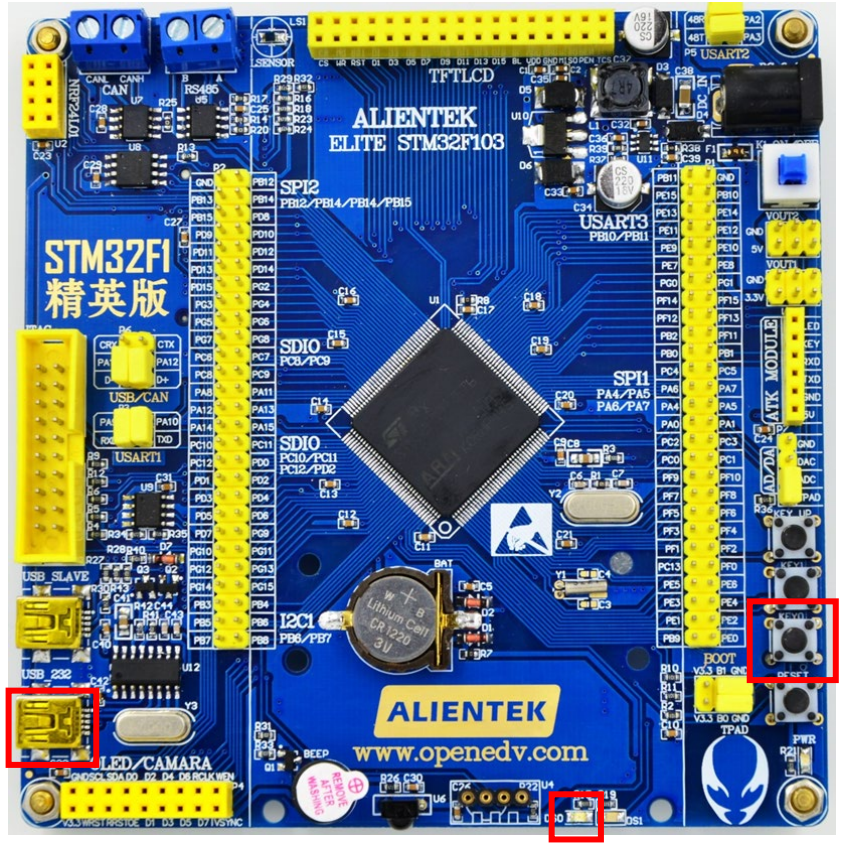
- 1. 串口通讯实验（无中断）
- 2. 串口通讯实验（串口中断）
- 3. 定时器中断实验
- 4. 串口通讯实验（按键中断）
- **5. 定时器计时实验**



STM32F103ZET6 开发板

5. 定时器计时实验

芯片引脚



USART

KEY0

LED0

对应芯片引脚

PB5 → LED0

PE4 → KEY0

任务

用定时器计算程序运行时间

按下KEY0后，用定时器计算延时程序

Delay_ms(200) 的运行时间，并发送至串口

LED0间隔0.2秒闪烁一次

5. 定时器计时实验

任务分解

1. 使能串口和PA的时钟
   RCC_APB2PeriphClockCmd
2. 复位USART
   USART_DeInit
3. 初始化PA9和PA10
   GPIO_InitTypeDef
   GPIO_Init
4. 初始化USART
   USART_InitTypeDef
   USART_Init
5. 使能USART
   USART_Cmd
6. 使能定时器时钟
   RCC_APB1PeriphClockCmd

5. 定时器计时实验

任务分解

7 初始化定时器

TIM_TimeBaseInitTypeDef

TIM_TimeBaseInit

8 使能定时器

TIM_Cmd

9 运行延时程序

Delay_ms

10 获得计时结果

TIM4->CNT

11 通过串口输出计时结果

USART_SendData

5. 定时器计时实验

串口发送字符串函数： USART_SendString

```
void USART_SendString(USART_TypeDef* USARTx, char *DataString)
{
    u8 i=0;
    // 循环发送所有字符
    while(DataString[i]!='\0')
    {
      // 发送单个字符
      USART_SendData(USARTx, DataString[i]);
      // 等待发送结束
      while(USART_GetFlagStatus(USARTx, USART_FLAG_TXE)==RESET);
      i++;
    }
}
```

嵌入式系统实验

——STM32开发板实验

浙江大学机械工程学院
实验教学中心