北京大学
PEKING UNIVERSITY

K210 UART_DMA_IRQ实验

汇报人：黄森

2024/10/16

目录

- 嵌入式系统的中断
- DMA工作流程
- K210 UART DMA IRQ实验例程

中断定义及类型

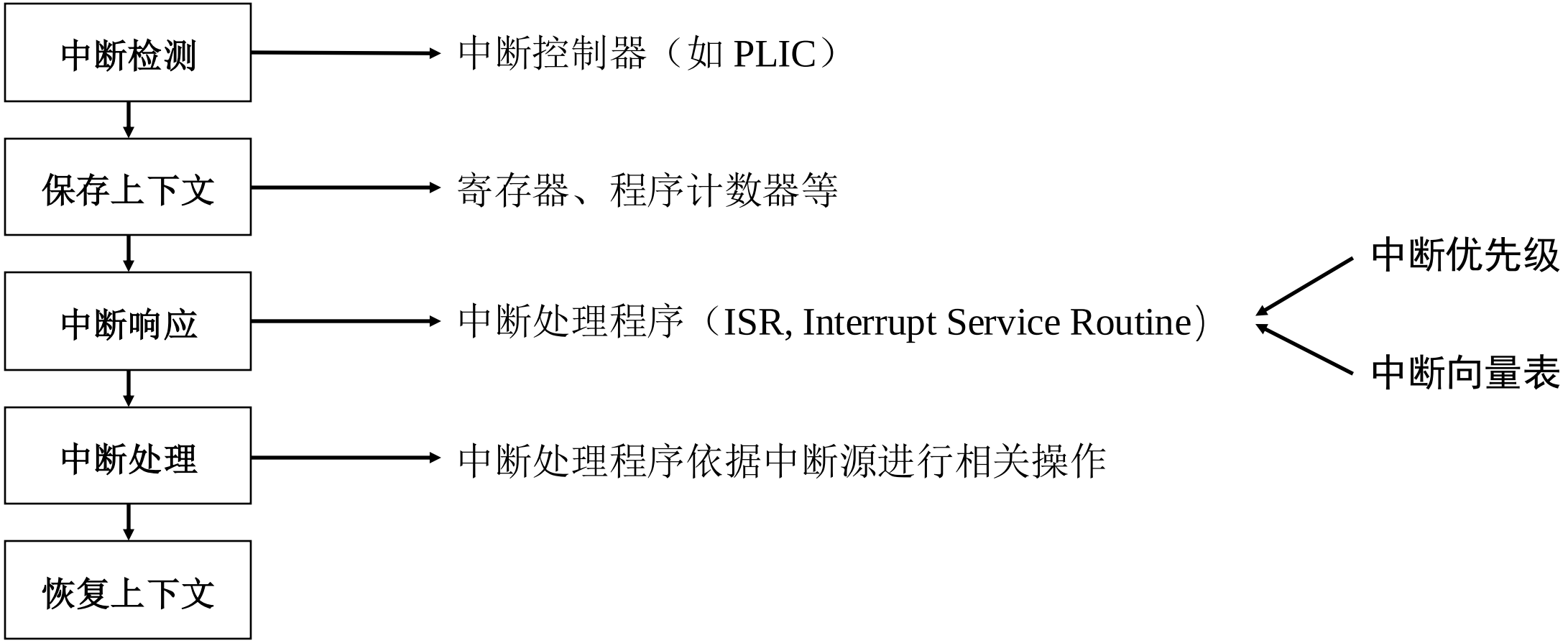
中断的定义

中断（Interrupt）是指在系统运行过程中，由于内部或外部的事件触发，使得处理器暂停当前的正常指令执行流程，转而去处理这些事件的一种机制。中断可以由外部设备（如传感器、按键等）触发，或者由处理器内部的定时器、异常等触发。

中断的类型

- 外部中断（External Interrupts）
- 内部中断（Internal Interrupts）
- 可屏蔽中断（Maskable Interrupts）
- 不可屏蔽中断（Non-maskable Interrupts, NMI）

中断的处理流程



中断处理流程图

中断与轮询

中断与轮询

◆**中断**：处理器只有在需要时才会暂停当前任务去处理事件，响应速度快，处理器资源利用率高。

◆**轮询**：处理器不断地检查外设状态，这会浪费处理器资源，并且响应速度较慢，特别是当外设事件发生频率低时。

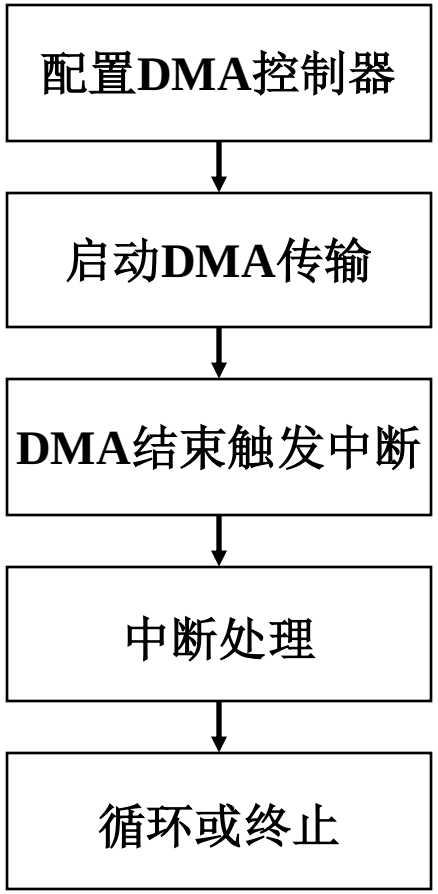
中断的优势：

●提高实时性

●减少CPU资源浪费

●灵活处理多任务

DMA工作流程



DMA工作流程图

- DMA控制器组成：
  - 源地址寄存器
  - 目标地址寄存器
  - 数据计数器
  - 控制寄存器
  - 中断系统

K210 UART DMA IRQ实验例程

北京大学 PEKING UNIVERSITY

- 初始化系统配置
  - IO端口模式配置
  - DMA初始化
  - PLIC初始化
  - 系统使能中断
  - UART初始化和配置

```c
int main(void)
{
    io_mux_init();
    dmac_init();
    plic_init();
    sysctl_enable_irq();

    gpiohs_set_drive_mode(3, GPIO_DM_OUTPUT);
    gpio_pin_value_t value = GPIO_PV_HIGH;
    gpiohs_set_pin(3, value);

    uart_init(UART_NUM);
    uart_configure(UART_NUM, 115200, 8, UART_STOP_1, UART_PARITY_NONE);
```

K210 UART DMA IRQ实验例程

北京大学 PEKING UNIVERSITY

- DMA配置（以发送端为例）
  - 设置发送内容
  - 申请内存区域并赋值
  - 初始化uart_date_t结构体
  - 初始化plic_interrupt_t结构体
  - 使用uart_handle_data_dma函数启动

```
uint8_t *hel = "hello!\n";

uint32_t *v_tx_buf = malloc(strlen(hel) * sizeof(uint32_t));
for(uint32_t i = 0; i < strlen(hel); i++)
{
    v_tx_buf[i] = hel[i];
}

uart_data_t data = (uart_data_t)
{
    .tx_channel = DMAC_CHANNEL0,
    .tx_buf = v_tx_buf,
    .tx_len = strlen(hel),
    .transfer_mode = UART_SEND,
};

plic_interrupt_t irq = (plic_interrupt_t)
{
    .callback = uart_send_done,
    .ctx = NULL,
    .priority = 1,
};

uart_handle_data_dma(UART_NUM, data, &irq);
```

K210 UART DMA IRQ实验例程

发送端流程

- 每秒发送一次
- DMA中断函数修改flag值

```
while(1)
{
    sleep(1);
    uart_handle_data_dma(UART_NUM, data, &irq);
    g_uart_send_flag = 2;
}
```

```
int uart_send_done(void *ctx)
{
    g_uart_send_flag = 1;
    return 0;
}
```

K210 UART DMA IRQ实验例程

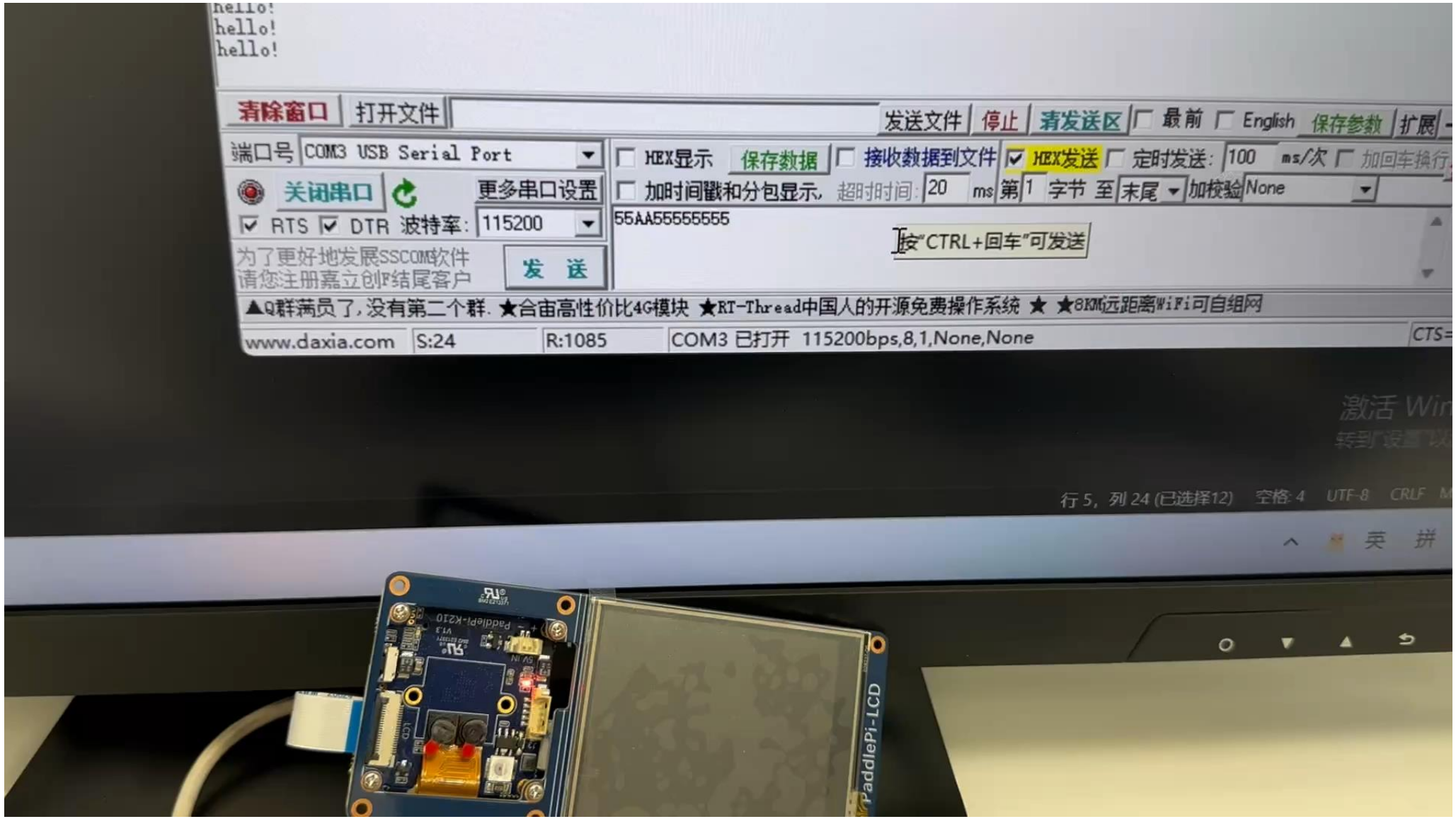
北京大学 PEKING UNIVERSITY

- 接收端流程

```c
int uart_recv_done(void *ctx)
{
    uint32_t *v_dest = ((uint32_t *)ctx) + RECV_DMA_LENTH;
    if(v_dest >= recv_buf + 48)
        v_dest = recv_buf;

    uart_data_t data = (uart_data_t)
    {
        .rx_channel = DMAC_CHANNEL1,
        .rx_buf = v_dest,
        .rx_len = RECV_DMA_LENTH,
        .transfer_mode = UART_RECEIVE,
    };

    plic_interrupt_t irq = (plic_interrupt_t)
    {
        .callback = uart_recv_done,
        .ctx = v_dest,
        .priority = 2,
    };

    uart_handle_data_dma(UART_NUM, data, &irq);
```

```c
    uint32_t *v_buf = (uint32_t *)ctx;
    for(uint32_t i = 0; i < RECV_DMA_LENTH; i++)
    {
        if(v_buf[i] == 0x55 && (recv_flag == 0 || recv_flag == 1))
        {
            recv_flag = 1;
            continue;
        }
        else if(v_buf[i] == 0xAA && recv_flag == 1)
        {
            recv_flag = 2;
            g_cmd_cnt = 0;
            continue;
        }
        else if(recv_flag == 2 && g_cmd_cnt < CMD_LENTH)
        {
            g_cmd[g_cmd_cnt++] = v_buf[i];
            if(g_cmd_cnt >= CMD_LENTH)
            {
                release_cmd(g_cmd);
                recv_flag = 0;
            }
            continue;
        }
        else
        {
            recv_flag = 0;
        }
    }

    return 0;
```

实验效果演示


hello!
hello!
hello!
清除窗口
打开文件
发送文件
停止
清发送区
最前
English
保存参数
扩展
端口号
COM3 USB Serial Port
HEX显示
保存数据
接收数据到文件
HEX发送
定时发送:
100
ms/次
加回车换行
关闭串口
更多串口设置
加时间戳和分包显示,
超时时间:
20
ms
第
1
字节
至
末尾
加校验
None
RTS
DTR
波特率:
115200
55AA55555555
为了更好地发展SSCOM软件
请您注册嘉立创P结尾客户
发 送
按"CTRL+回车"可发送
▲Q群满员了,没有第二个群. ★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自组网
www.daxia.com
S:24
R:1085
COM3 已打开 115200bps,8,1,None,None
CTS=
行 5，列 24 (已选择12)
空格: 4
UTF-8
CRLF
PaddlePi-LCD

汇报结束

感谢聆听