

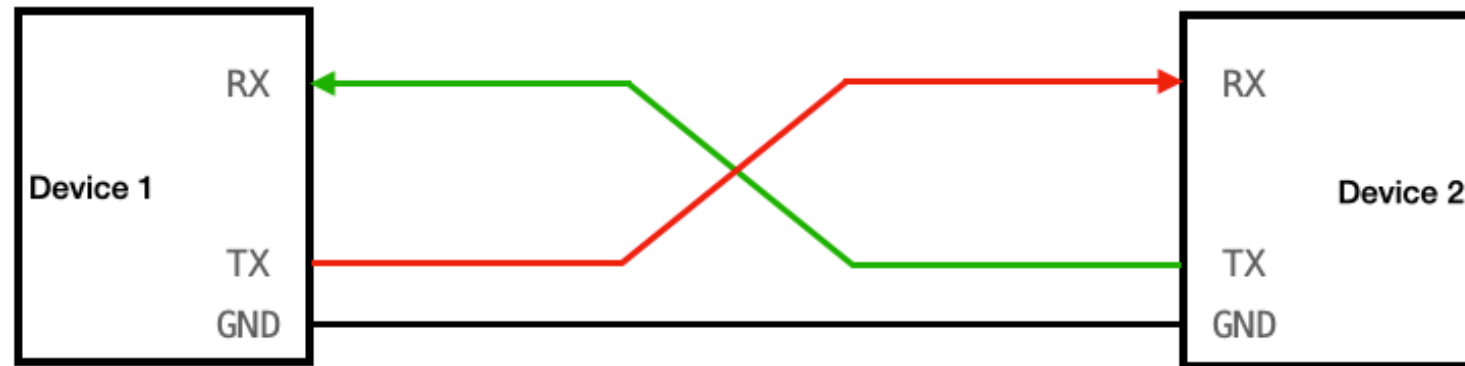
UART收发测试实验

实验目标：

- ①复习UART串口通信原理；
- ②使用串口通信软件与开发板进行UART收发测试；
- ③了解UART收发测试代码的基本结构和测试方法。

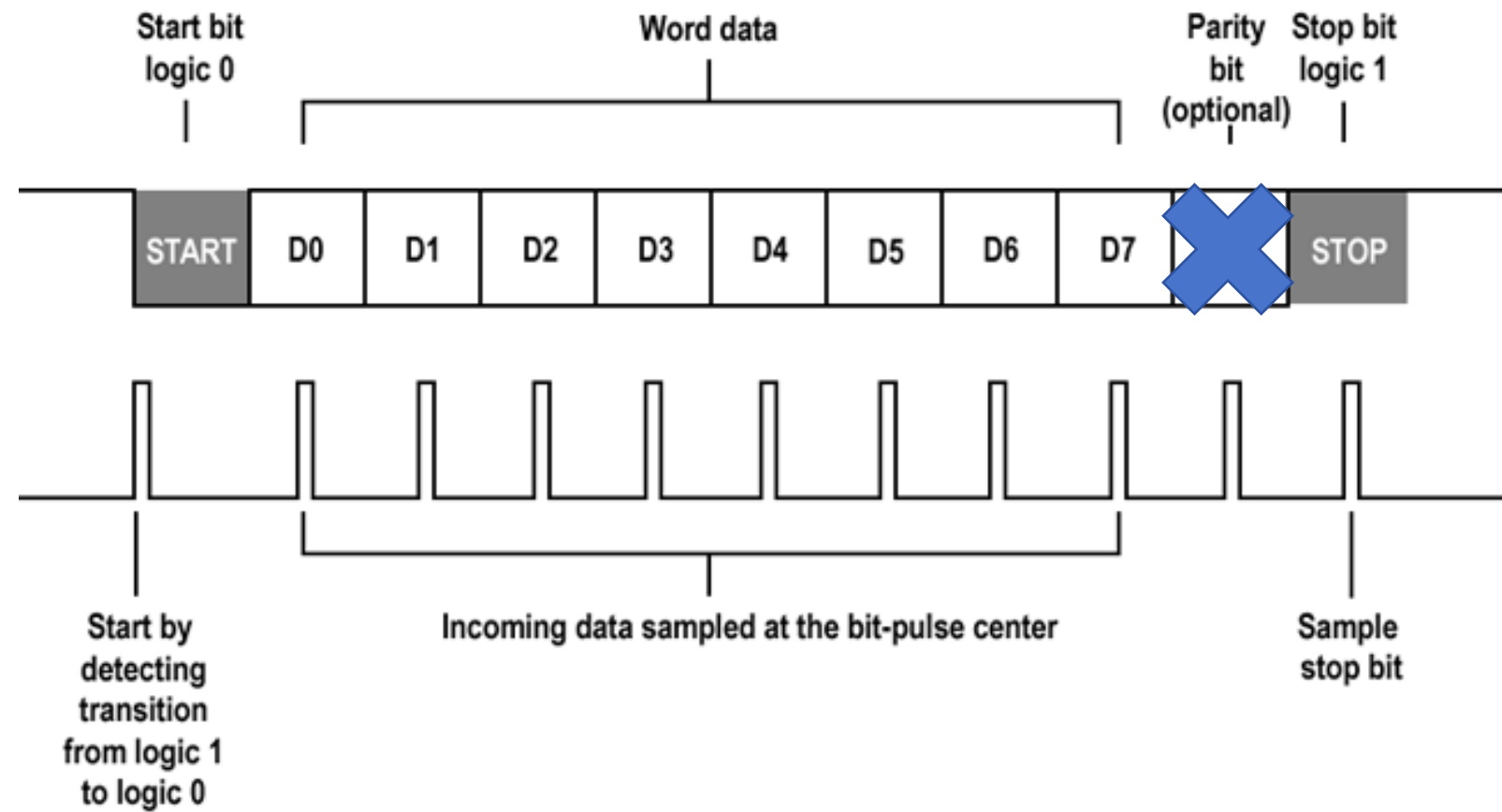
实验步骤：代码初始化，并发送hello world置串口；通过串口软件向开发板发送数据，当发送到特定的字符串时开发板会控制LED亮、灭。

UART原理及协议设定



- 串行通信：每次发送一帧数据，数据逐位传输。
- 异步通信：每两帧数据之间的时间间隔不固定（但同一个字符中的相邻bit传输间隔是固定的）
- 数据传输速率（波特率）：每秒传输的码元的位数，一般有300、1200、2400、9600、、19200、38400、115200等。
- 在串行通信中，数据以帧的形式进行传输，一帧包含了一个起始位、几个数据位、一个停止位，每次数据发送都发送一个完整的帧。

UART原理及协议设定



- 两个UART设备之间通信不需要时钟线，因此需要指定相同的传输速率、起始位、空闲位、奇偶校验位和结束位
- 协议设定：①传输速率；②起始位和空闲位的高低电平；③校验位；④停止位的长度与高低电平

代码功能实现——库、初始化、引脚复用

```
#include <stdio.h>
#include "fpioa.h"
#include <string.h>
#include "uart.h"
#include "gpiohs.h"
#include "sysctl.h"
```

```
void io_mux_init(void)
{
```

```
    fpioa_set_function(4, FUNC_UART1_RX);
    fpioa_set_function(5, FUNC_UART1_TX);
    fpioa_set_function(10, FUNC_GPIOHS3);
}
```

```
int main()
{
```

```
    io_mux_init(); //设置管脚复用 (函数)
    plic_init(); //初始化外部中断
    sysctl_enable_irq(); //使能系统中断
```

```
    //将引脚10 (连接LED) 设置为输出驱动, 并且设置为高电平
    gpiohs_set_drive_mode(3, GPIO_DM_OUTPUT);
    gpio_pin_value_t value = GPIO_PV_HIGH;
    gpiohs_set_pin(3, value);
```

```
64      | UART1_RX      | UART1 Receiver
65      | UART1_TX      | UART1 Transmitter
66      | UART2_RX      | UART2 Receiver
67      | UART2_TX      | UART2 Transmitter
68      | UART3_RX      | UART3 Receiver
69      | UART3_TX      | UART3 Transmitter
```

```
typedef enum _uart_device_number
```

```
{
    UART_DEVICE_1,
    UART_DEVICE_2,
    UART_DEVICE_3,
    UART_DEVICE_MAX,
} uart_device_number_t;
```

```
typedef enum _uart_stopbit
```

```
{
    UART_STOP_1,
    UART_STOP_1_5,
    UART_STOP_2
} uart_stopbit_t;
```

```
typedef enum _uart_parity
```

```
{
    UART_PARITY_NONE,
    UART_PARITY_ODD,
    UART_PARITY_EVEN
} uart_parity_t;
```

代码功能实现——设置UART相关参数、hello world传输

```
#define UART_NUM      UART_DEVICE_1

//设置UART相关参数
uart_init(UART_NUM);
uart_configure(UART_NUM, 115200, 8, UART_STOP_1, UART_PARITY_NONE); //设置UART相关参数

//初始化发送的数据
char *hel = {"hello world!\n"};
uart_send_data(UART_NUM, hel, strlen(hel)); //通过UART发送数据到串口
```

11.3.1 uart_init

11.3.1.1 描述
初始化 uart。

11.3.1.2 函数原型

```
void uart_init(uart_device_number_t channel)
```

11.3.3 uart_configure

11.3.3.1 描述
设置 UART 相关参数。

11.3.3.2 函数原型

```
void uart_configure(uart_device_number_t channel, uint32_t baud_rate, uart_bitwidth_t data_width, uart_stopbit_t stopbit, uart_parity_t parity)
```

11.3.3.3 参数

参数名称	描述	输入输出
channel	UART 编号	输入
baud_rate	波特率	输入
data_width	数据位 (5-8)	输入
stopbit	停止位	输入
parity	校验位	输入

11.3.4 uart_send_data

11.3.4.1 描述
通过 UART 发送数据。

11.3.4.2 函数原型

```
int uart_send_data(uart_device_number_t channel, const char *buffer, size_t buf_len)
```

11.3.4.3 参数

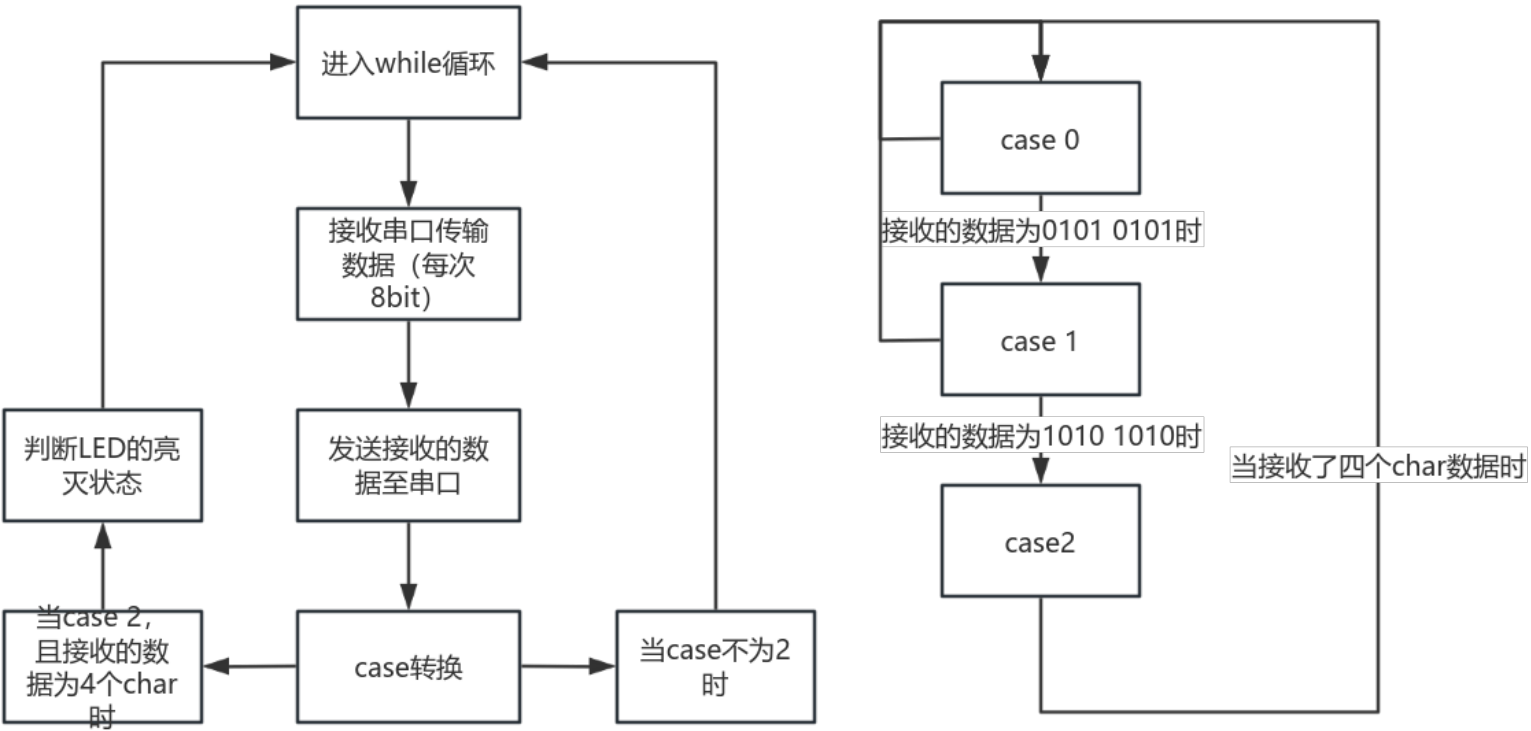
参数名称	描述	输入输出
channel	UART 编号	输入
buffer	待发送数据	输入
buf_len	待发送数据的长度	输入

代码功能实现——UART收发与LED控制（状态机）

```
char recv = 0;
int rec_flag = 0;
char cmd[8];
int i = 0;
while (1)
{
    while(uart_receive_data(UART_NUM, &recv, 1))//通过uart读取
    {
        uart_send_data(UART_NUM, &recv, 1); //发送数据（发送读取到
        switch(rec_flag) //eg：发送数据为55AA55555555。①第一次发送
        {
            case 0:
                recv == 0x55 ? (rec_flag = 1) : rec_flag;
                break;
            case 1:
                recv == 0xAA ? (rec_flag = 2) : (rec_flag = 0);
                break;
            case 2:
                cmd[i++] = recv;
                if(i >= RECV_LENTH) //case2进入4次之后，i为4，进入到次
                {
                    i = 0;
                    release_cmd(cmd);
                    rec_flag = 0;
                }
                break;
        }
    }
}
while(1);
```

```
#define RECV_LENTH 4
#define CLOSLIGHT 0x55555555
#define OPENLIGHT 0xAAAAAAAA
```

```
int release_cmd(char *cmd)
{
    switch(*((int *)cmd)) { //((int *) cmd)
        case CLOSLIGHT:
            gpiohs_set_pin(3, GPIO_PV_LOW); //1
            break;
        case OPENLIGHT:
            gpiohs_set_pin(3, GPIO_PV_HIGH);
            break;
    }
    return 0;
}
```



串口软件介绍

SSCOM V5.13.1 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群: 52502449(最新版本)

通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者 大虾电子网

hello world!

Setup

Settings

Port

COM3

Baud rate

115200

Data bits

8

Stop bits

1

Parity

None

Flow control

None

OK

Cancel

清除窗口 打开文件 .ePi-master\standalone-demos\uart\README.md 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

☒ RTS ☒ DTR 波特率: 115200

☐ 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

关闭串口 更多串口设置

发送

为了更好地发展SSCOM软件
请您注册嘉立创结尾客户

www.daxia.com S:0 R:13 COM3 已打开 115200bps,8,1,None,None CTS=0 DSR=0 RLSI

通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者 大虾电子网

68 65 6C 6C 6F 20 77 6F 72 6C 64 21 0A 55 AA 55 55 55 55 55 AA AA AA AA AA

结果演示