

工程结构

概要

讲解了串口舵机演示例程的工程结构规范，以及库文件的作用。

工程结构

以例程**舵机通讯检测**为例，讲解STM32的工程结构。

打开文件夹 *舵机通讯检测/源代码/FashionStarUartServo*

目录结构如下：

- **Project**
Keil5的工程文件，点击 `FashionStarUartServo.uvprojx` 即可通过Keil5打开此工程。
- **User**
主程序以及用户自定义库
 - `main.c` 用户主程序
 - 用户自定义的库文件，例如 *串口舵机驱动库*等
- **Libraries**
 - `CMSIS`：存放ARM公司为cm3内核专一定制的内核文件接口
 - `FWLIB`：存放STM32的外设库
- **Listings**：该目录是MDK生成信息输出目录，存放代码分布（.map和.lst）
- **Output** 该目录是MDK生成信息输出目录，存放目标文件(.o)、调试文件(.axf)、下载文件(.hex)、依赖文件（.d）等。

用户自定义库文件

User文件夹 工程结构介绍

- **sys_tick**
管理系统时间。通过配置系统定时器中断，实现了延时以及倒计时的逻辑。
- **ring_buffer**
用C语言实现了环形缓冲队列。用于存放管理串口通信的字节流。同时还具备向缓冲队列读取/写入特定数据类型的数值的功能。
- **usart**
串口通信的库。通过配置宏，可以方便的打开关闭STM32F103C8开发板上三个USART资源。
- **fashion_star_uart_servo**
封装了Fashion Star 串口舵机的通信协议，是串口舵机STM32F103版本的SDK。
- **main.c**

主程序。程序入口。

user 文件目录树

```
├─ fashion_star_uart_servo 串口舵机驱动库
│   ├── fashion_star_uart_servo.c
│   └─ fashion_star_uart_servo.h
├─ main.c 主程序
├─ ring_buffer 环形缓冲队列驱动库
│   ├── README.md
│   ├── ring_buffer.c
│   └─ ring_buffer.h
├─ stm32f10x_conf.h
├─ sys_tick 系统时钟
│   ├── sys_tick.c
│   └─ sys_tick.h
└─ usart 串口通信
    ├── README.md
    ├── usart.c
    └─ usart.h
```

Reference

1. 《STM32库开发实战指南》
2. [MDK建立STM32F103*开发模板](#)