

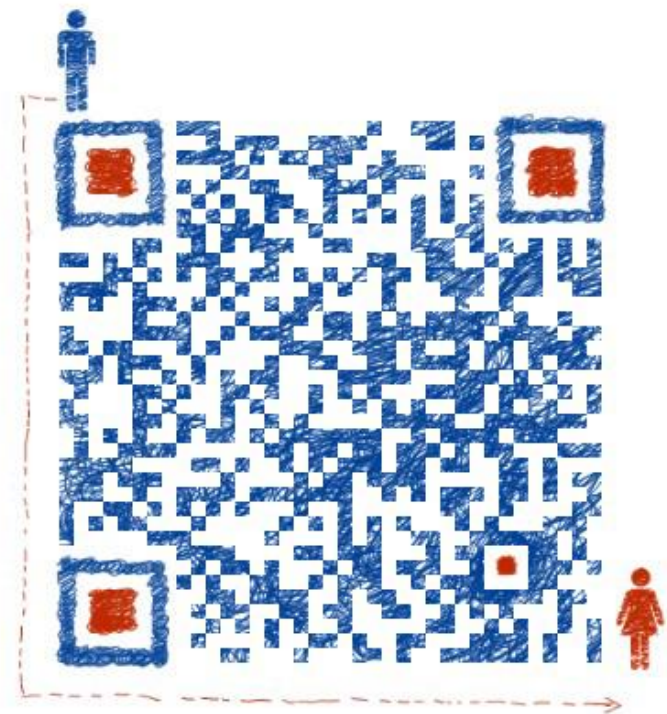
主控程序说明

—□Linux

无线传感器网项目实战



公众号:一口Linux



彭老师个人微信号



01



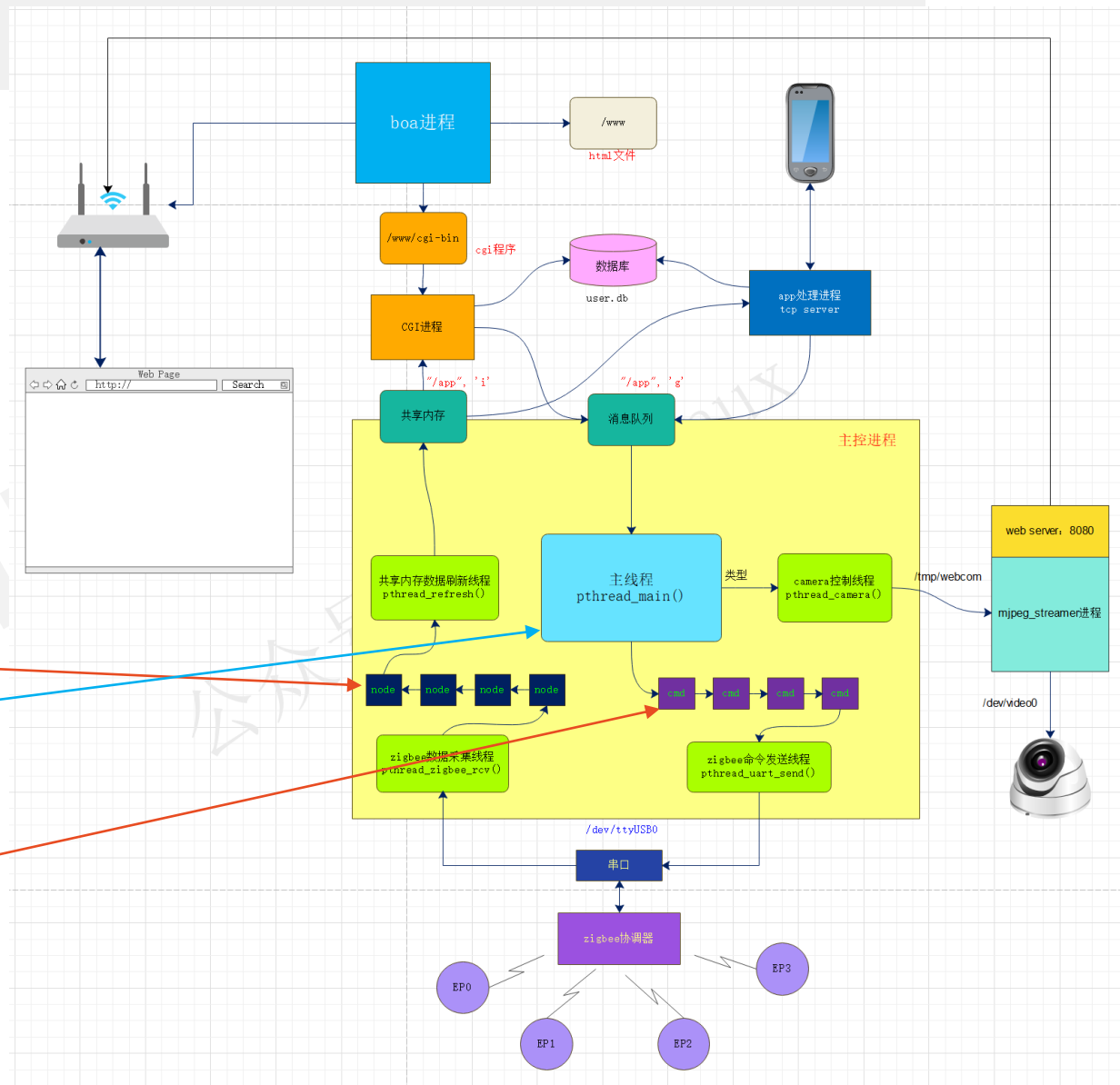
文件说明&编译

主控文件

```
peng@ubuntu:~/work/mainthread$ tree ./
```

```
./
├── bin
│   ├── sqlite3
│   ├── data_global.c
│   ├── data_global.h
│   ├── lib
│   │   ├── libsqlite3.a
│   │   ├── libsqlite3.la
│   │   ├── libsqlite3.so
│   │   ├── libsqlite3.so.0
│   │   ├── libsqlite3.so.0.8.6
│   │   ├── pkgconfig
│   │   └── sqlite3.pc
│   ├── link_list.c
│   ├── link_list.h
│   ├── main.c
│   └── Makefile
├── pthread_camera.c
├── pthread_main.c
├── pthread_refresh.c
├── pthread_uart_send.c
├── pthread_zigbee_rcv.c
├── sem.h
├── test.c
├── uart_cache.c
└── uart_cache.h
```

3 directories, 22 files



关注公众号：一口Linux

Makefile

编译脚本文件Makefile

对应工程
里所有的.c
文件

```
1 CC=gcc
2 CFLAGS1=-Wall -c -g
3 CFLAGS2= -lpthread 线程库
4 OBJS=main.o data_global.o pthread_zigbee_rcv.o link_list.o pth
5
6 zigbee:$(OBJS) 1. 目标zigbee依赖 $(OBJS)
7 $(CC) -o $@ $^ $(CFLAGS2) 3. 将OBJS代表的所有.o文件编译生成目标文件
8 $(OBJS):%.o:%.c 2. $(OBJS)中所有的.o文件依赖相同名字的.c文件编译生成
9 $(CC) $(CFLAGS1) $< -o $@
10
11 .PHONY:clean 4. 执行make clean目标，清除编译生成的文件
12 clean:
13 rm *.o zigbee
```

gcc

-Wall -c -g

.c文件

.o文件

\$@--目标文件，
\$^--所有的依赖文件，
\$<--第一个依赖文件

```
peng@ubuntu:~/work/mainthread$ tree ./
./
├── bin
│   └── sqlite3
├── data_global.c
├── data_global.h
├── lib
│   ├── libsqlite3.a
│   ├── libsqlite3.la
│   ├── libsqlite3.so
│   ├── libsqlite3.so.0
│   ├── libsqlite3.so.0.8.6
│   ├── pkgconfig
│   └── sqlite3.pc
├── link_list.c
├── link_list.h
├── main.c
├── Makefile
├── pthread_camera.c
├── pthread_main.c
├── pthread_refresh.c
├── pthread_uart_send.c
├── pthread_zigbee_rcv.c
├── sem.h
├── test.c
├── uart_cache.c
└── uart_cache.h

3 directories, 22 files
```

第三方库
如果是交
叉编译，
可以放在
此处

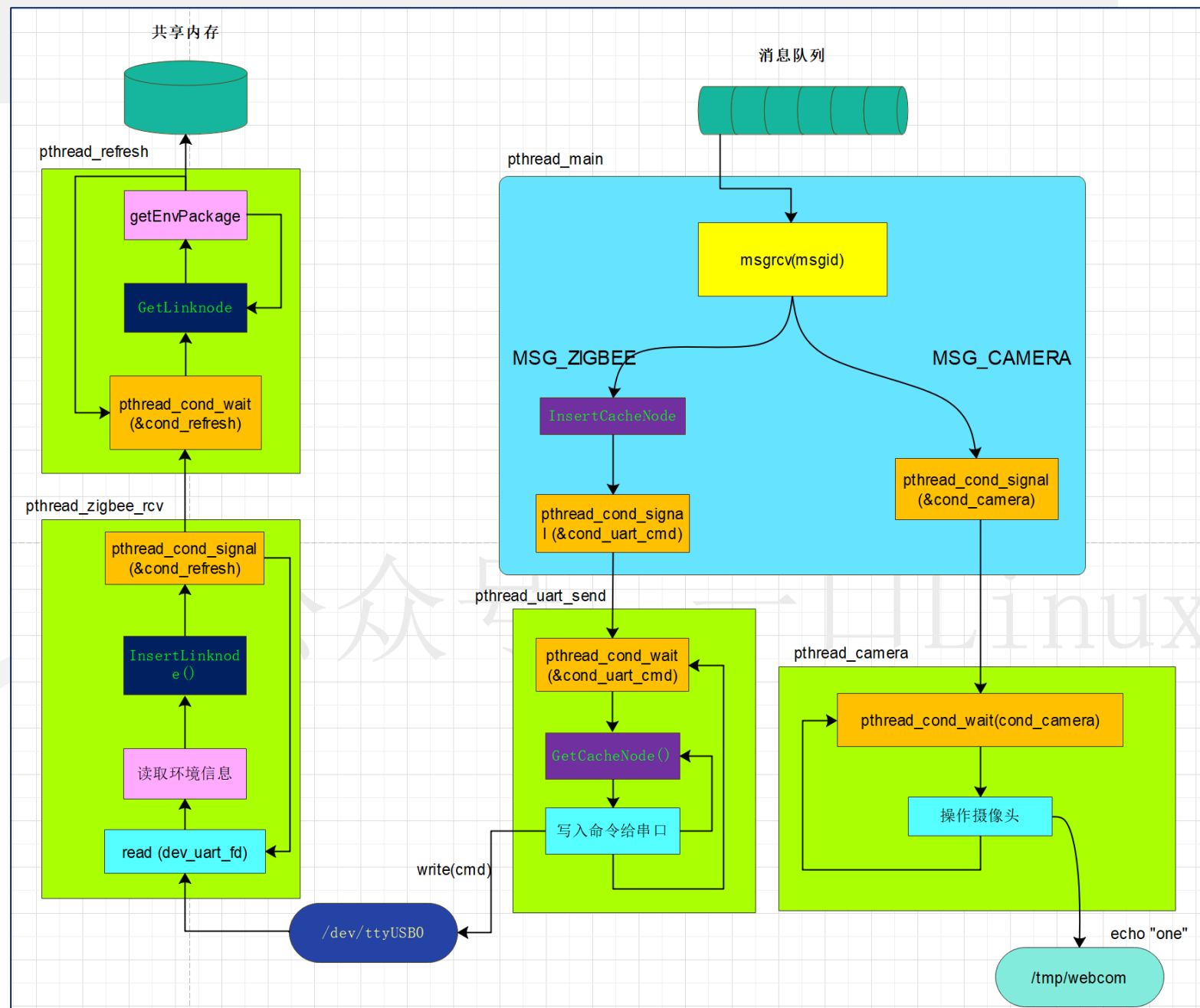
02

主控程序架构详解

初始化

- 文件
 - `main.c`
- 函数
 - `main()`
 - `releaseResource()`

函数流程图



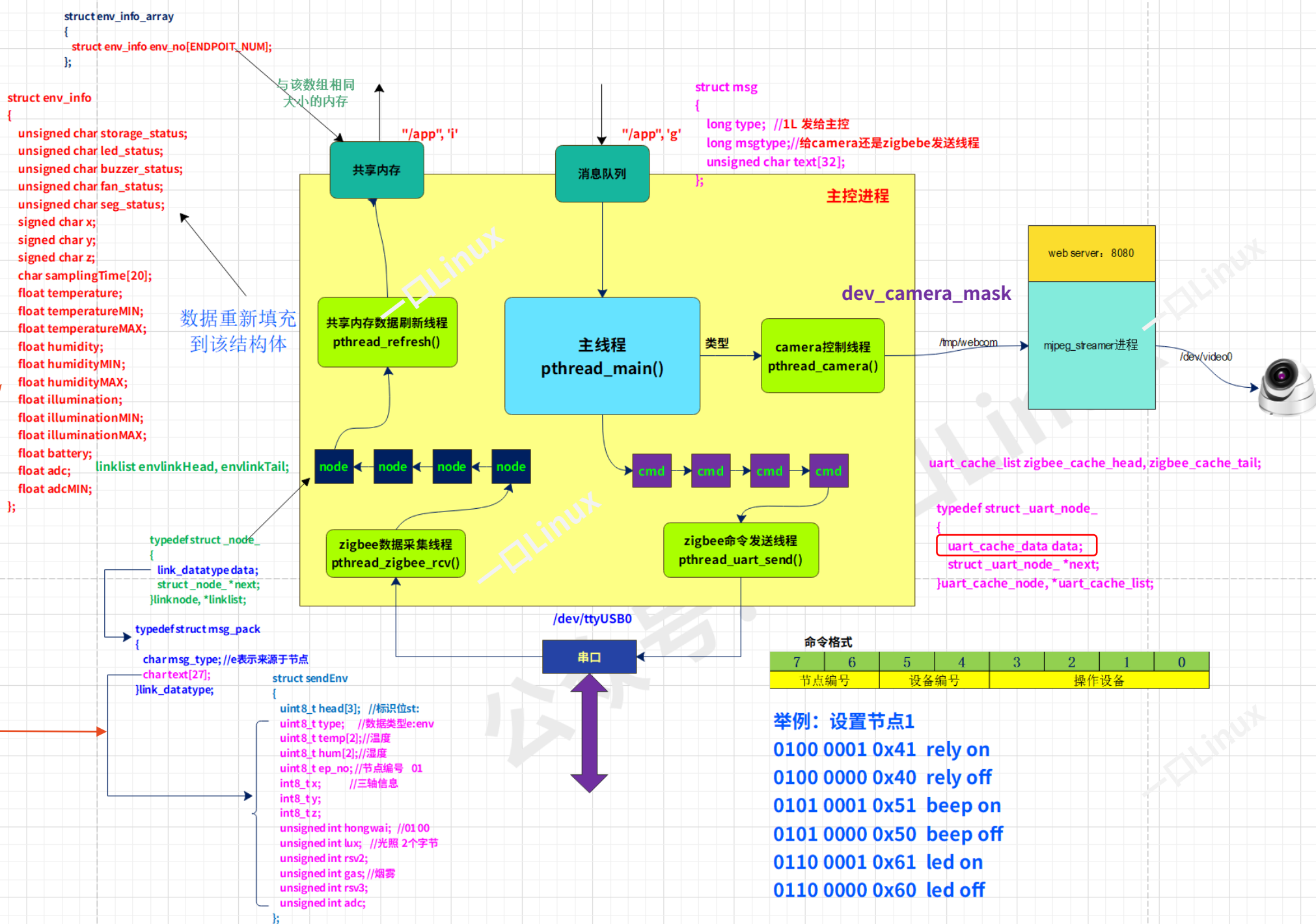
03

数据结构及数据流 详解

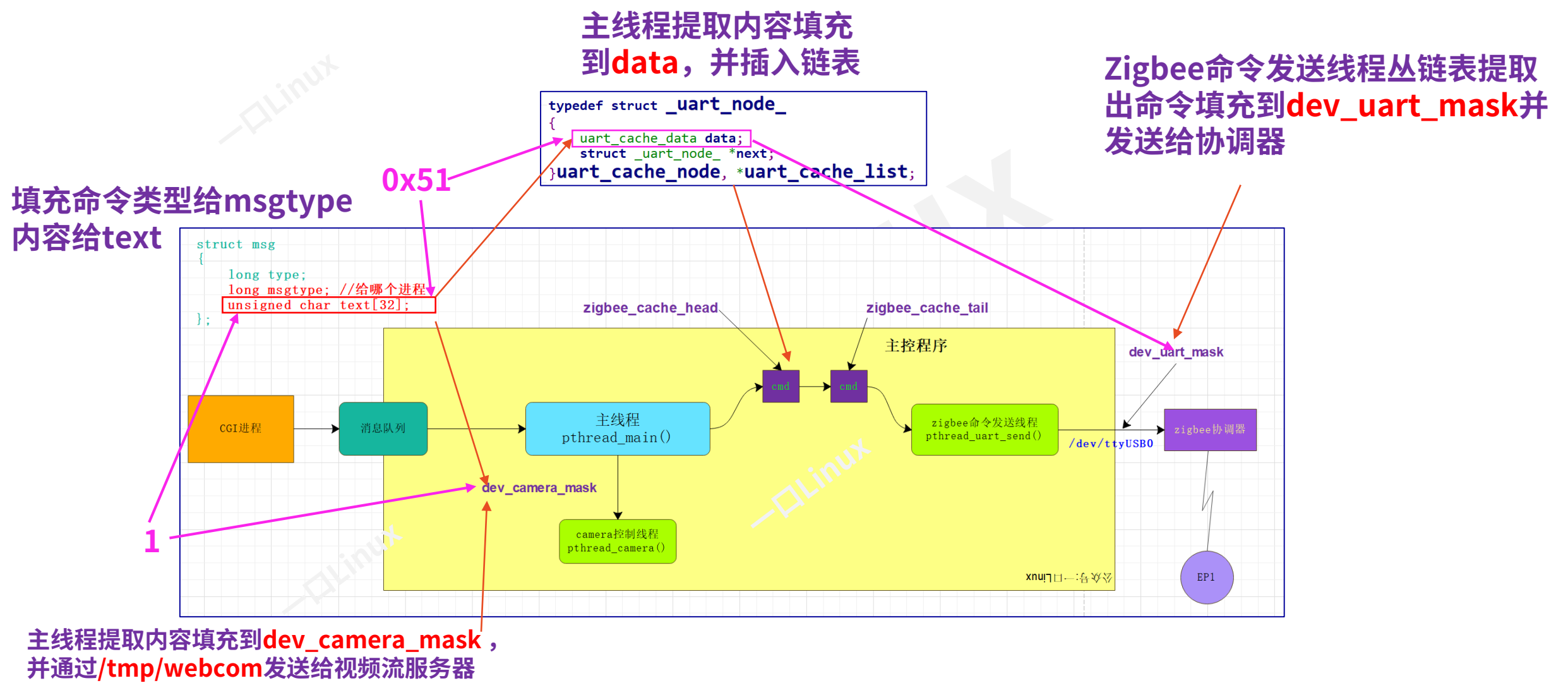
数据结构

```
struct getEnvMsg
{
    unsigned char tem[2];
    unsigned char hum[2];
    unsigned char ep_no;
    unsigned char x;
    unsigned char y;
    unsigned char z;

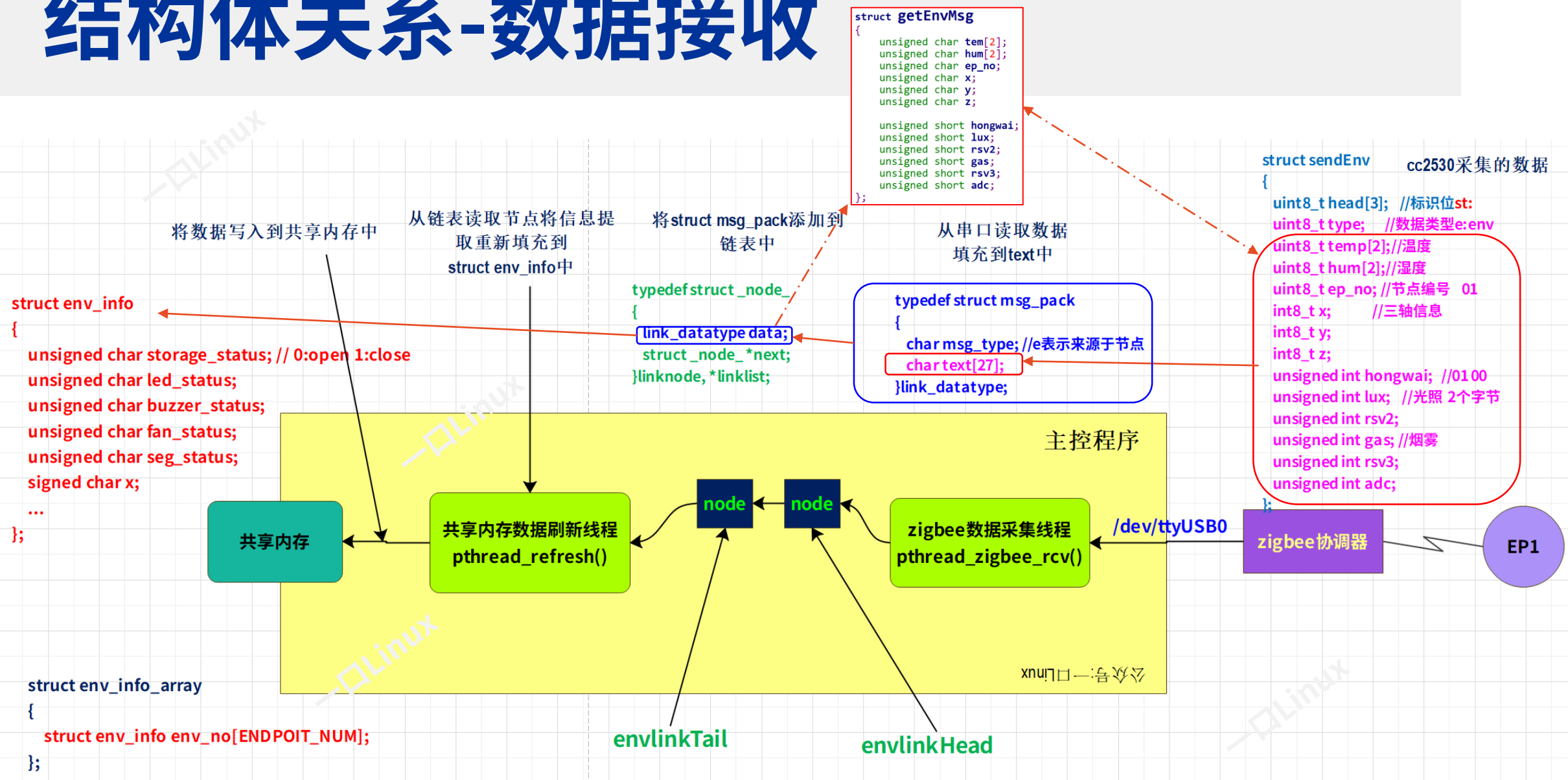
    unsigned short hongwai;
    unsigned short lux;
    unsigned short rsv2;
    unsigned short gas;
    unsigned short rsv3;
    unsigned short adc;
};
```



结构体关系-命令发送



结构体关系-数据接收

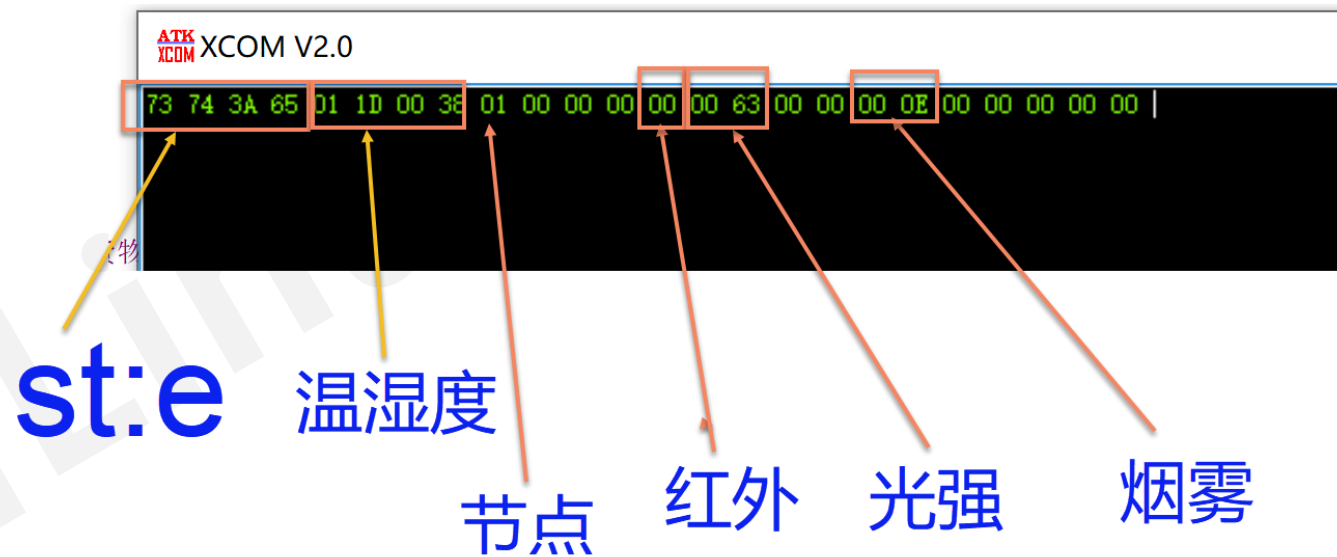


采集数据格式说明

```
struct _EnvMsg
{
    uint8_t head[3];    //标识位st:73 74 3A 65
    uint8_t type;       //数据类型

    uint8_t temp[2];    //温度
    uint8_t hum[2];     //湿度
    uint8_t ep_no;      //节点编号01

    int8_t x;           //三轴信息
    int8_t y;
    int8_t z;
    unsigned int hongwai; //01 00
    unsigned int lux;     //光照2个字节
    unsigned int rsv2;
    unsigned int gas;     //烟雾
    unsigned int rsv3;
    unsigned int adc;
};
```



04

主控测试程序

test_shm.c

- 读取共享内存中数据

一口Linux

test_cam.c

- 测试摄像头

一口Linux

test_zigbee.c

- 远程控制zigbee节点设备

一口Linux



更多嵌入式Linux知识
请关注一口君的公众号：一口Linux

公众号：一口Linux