

lwip移植

嵌入式工作 [关注](#)

2018.09.07 10:59:10 字数 651 阅读 2,100

转载地址: <https://www.cnblogs.com/electron/p/3546508.html>资源下载: 链接: <https://pan.baidu.com/s/1kHu7E9qpV5LUtGrK6jFVQ> 密码: hb6n

LwIP移植和使用

本手册基于lwip-1.4.x编写, 本人没有移植过1.4.0之前的版本, 更早的版本或许有差别

LwIP官网是: <http://savannah.nongnu.org/projects/lwip/>

你可以从这里获取源代码。当然也可以从Git获取源代码:

git clone <git://git.savannah.nongnu.org/lwip.git>

LwIP以BSD协议发布源代码, 我们可以自由的使用, 修改, 发布或不发布源代码。

附件中有我移植的文件, 可以用来参考。祝你移植顺利。

移植

1)新建几个头文件:

include/lwipopts.h // lwip配置文件

include/arch/cc.h // 平台相关。类型定义,大小端设置,内存对齐等

include/arch/perf.h // 平台相关的性能测量实现(没用)

include/arch/sys_arch.h // RTOS抽象层。信号量, mbox等类型定义, 函数声明

lwipopts.h // lwip配置文件, 详见附件

cc.h //类型定义,大小端设置,内存对齐等

```
1  #ifndef __CC_H__
2  #define __CC_H__
3
4  #include <stdint.h>
5
6  /* Types based on stdint.h */
7  typedef uint8_t      u8_t;
8  typedef int8_t       s8_t;
9  typedef uint16_t     u16_t;
10 typedef int16_t      s16_t;
11 typedef uint32_t     u32_t;
12 typedef int32_t      s32_t;
13 typedef uintptr_t    mem_ptr_t;
```



嵌入式工作

总资产1 (约0.15元)

[关注](#)

单片机 mcu stm32 printf自定义

阅读 22

ubuntu git ssh方式添加钥匙

阅读 13

ubuntu Git 使用教程 自测好用

阅读 13

推荐阅读

油田人捧了三代的铁饭碗, 说碎就碎了

阅读 4,263

分享 | 不要再说书评难写啦! 我写了174篇书评, 总结五个步骤

阅读 5,549

郑爽自曝想转型做网红: 她真的是一手好牌打得稀巴烂吗?

阅读 9,142

何炅一句话让人崩溃: “看你的朋友圈, 就知道你昨晚偷偷哭过”

阅读 38,707

《上海堡垒》血亏3亿, 流量鲜肉鹿晗要背这个锅吗?

阅读 16,805

写下你的评论...

评论0

赞1

...

```
20 #define S32_F "ld"
21 #define X32_F "lx"
22 #define SZT_F "uz"
23
24 /* 选择小端模式 */
25 #define BYTE_ORDER LITTLE_ENDIAN
26
27 /* Use LWIP error codes */
28 #define LWIP_PROVIDE_ERRNO
29
30 /* 内存对齐 */
31 #if defined(__arm__) && defined(__ARMCC_VERSION)
32 /* Keil uVision4 tools */
33 #define PACK_STRUCT_BEGIN __packed
34 #define PACK_STRUCT_STRUCT
35 #define PACK_STRUCT_END
36 #define PACK_STRUCT_FIELD(fld) fld
37 #define ALIGNED(n) __align(n)
38
39 #endif
```

perf.h // 两个宏定义为空即可

```
1 #ifndef __PERF_H__
2 #define __PERF_H__
3
4 #define PERF_START /* null definition */
5 #define PERF_STOP(x) /* null definition */
6
7 #endif /* END __PERF_H__ */
```

sys_arch.h

RTOS抽象层的类型定义，函数声明，详细内容见 doc/sys_arch.h

2建立RTOS抽象层文件：

port/sys_arch.c // RTOS抽象层实现

为了屏蔽不同RTOS在信号量，互斥锁，消息，任务创建等OS原语使用上的差别，lwip构造了一个RTOS的抽象层，规定了OS原语的数据类型名称和对应方法名称。我们要做的就是根据所用RTOS的api去实现这些原语。

比如移植lwip到raw-os上，信号量的移植：

类型定义，宏定义在sys_arch.h中

```
1 struct _sys_sem
2 {
3     RAW_SEMAPHORE *sem;
4 };
5
6 typedef struct _sys_sem sys_sem_t; // sys_sem_t是lwip的信号量类型名
7 #define SYS_SEM_NULL NULL
8 #define sys_sem_valid(sema) (((sema) != NULL) && ((sema)->sem != NULL))
9 #define sys_sem_set_invalid(sema) ((sema)->sem = NULL)
10 err_t sys_sem_new(sys_sem_t *sem, u8_t count)
11 {
```

写下你的评论...

评论0

赞1

...

```
17
18     semaphore_ptr = port_malloc(sizeof(RAW_SEMAPHORE));
19     if(semaphore_ptr == 0)
20     {
21         RAW_ASSERT(0);
22     }
23
24     //这是raw-os的API
25     raw_semaphore_create(semaphore_ptr, (RAW_U8 *)"name_ptr", count);
26     sem->sem = semaphore_ptr;
27
28     return ERR_OK;
29 }
30
31 void sys_sem_free(sys_sem_t *sem)
32 {
33     if((sem == NULL) || (sem->sem == NULL))
34     {
35         RAW_ASSERT(0);
36     }
37
38     raw_semaphore_delete(sem->sem); //这是raw-os的API
39
40     raw_memset(sem->sem, sizeof(RAW_SEMAPHORE), 0);
41     port_free(sem->sem);
42     sem->sem = NULL;
43 }
44
45
```

还有几个函数就不一一列举了，如有疑问看doc/sys_arch.txt

3修改网卡框架文件：

netif/ethernetif.c

该文件是作者提供的网卡驱动和lwip的接口框架。

该文件中要改动的函数只有3个：

```
static void low_level_init(struct netif *netif);
```

```
static err_t low_level_output(struct netif *netif, struct pbuf *p);
```

```
static struct pbuf *low_level_input(struct netif *netif);
```

```
1  /* 你可以给网卡起个名字 */
2  /* Define those to better describe your network interface. */
3  #define IFNAME0 'e'
4  #define IFNAME1 '0'
5
6  /**
7   * Helper struct to hold private data used to operate your ethernet
8   * interface.
9   * Keeping the ethernet address of the MAC in this struct is not
10  * necessary as it is already kept in the struct netif.
11  * But this is only an example, anyway...
12  */
13  struct ethernetif
14  {
15      struct eth_addr *ethaddr;
16      // Add whatever per-interface state that is needed here.
17      // 在这里添加网卡的私有数据，比如和网卡相关的信号量，互斥锁。
```

写下你的评论...

评论0

赞1

...

函数改动

```
1 static void low_level_init(struct netif *netif)
2 {
3     struct ethernetif *ethernetif = netif->state;
4
5     /* set MAC hardware address length */
6     netif->hwaddr_len = ETHARP_HWADDR_LEN;
7
8     /* 设置MAC地址, 必须与网卡初始化的地址相同 */
9     netif->hwaddr[0] = ;
10    netif->hwaddr[1] = ;
11    netif->hwaddr[2] = ;
12    netif->hwaddr[3] = ;
13    netif->hwaddr[4] = ;
14    netif->hwaddr[5] = ;
15
16    /* maximum transfer unit */
17    netif->mtu = 1500;
18
19    /* device capabilities */
20    /* don't set NETIF_FLAG_ETHARP if this device is not an ethernet one */
21    netif->flags = NETIF_FLAG_BROADCAST | NETIF_FLAG_ETHARP | NETIF_FLAG_LINK_UP;
22
23    /* 在这里添加其他初始化代码(如真正的网卡初始化, phy初始化等) */
24 }
25
26
27
28 static err_t low_level_output(struct netif *netif, struct pbuf *p)
29 {
30     struct ethernetif *ethernetif = netif->state;
31     struct pbuf *q;
32
33     initiate_transfer();
34
35     #if ETH_PAD_SIZE
36     pbuf_header(p, -ETH_PAD_SIZE); /* drop the padding word */
37     #endif
38
39     for(q = p; q != NULL; q = q->next){
40         /* Send the data from the pbuf to the interface, one pbuf at a
41            time. The size of the data in each pbuf is kept in the ->len
42            variable. */
43         send_data_from(q->payload, q->len);
44     }
45
46     signal_that_packet_should_be_sent();
47
48     #if ETH_PAD_SIZE
49     pbuf_header(p, ETH_PAD_SIZE); /* reclaim the padding word */
50     #endif
51
52     LINK_STATS_INC(link.xmit);
53
54     return ERR_OK;
55 }
56
57 static struct pbuf * low_level_input(struct netif *netif)
58 {
59     struct ethernetif *ethernetif = netif->state;
60     struct pbuf *p, *q;
61     u16_t len;
62
63     /* Obtain the size of the packet and put it into the "len" variable. */
64     len = ; // 获取将要接收的数据长度
65
66     #if ETH_PAD_SIZE
67     struct pbuf *p = pbuf_alloc(PBUF_POOL, len + ETH_PAD_SIZE, PBUF_POOL);
68     if (p == NULL)
69         return NULL;
70     pbuf_header(p, -ETH_PAD_SIZE);
71     #endif
72
73     /* Obtain the data from the interface and put it into the pbuf. */
74     receive_data_to(p->payload, len);
75
76     return p;
77 }
```

写下你的评论...

评论0

赞1

...

```
73     if (p != NULL){
74 #if ETH_PAD_SIZE
75         pbuf_header(p, -ETH_PAD_SIZE); /* drop the padding word */
76 #endif
77
78         /* We iterate over the pbuf chain until we have read the entire
79          * packet into the pbuf. */
80         for(q = p; q != NULL; q = q->next) {
81             /* Read enough bytes to fill this pbuf in the chain. The
82              * available data in the pbuf is given by the q->len
83              * variable.
84              * This does not necessarily have to be a memcpy, you can also
85              * preallocate pbufs for a DMA-enabled MAC and after receiving truncate
86              * it to the actually received size. In this case, ensure the tot_len
87              * member of the pbuf is the sum of the chained pbuf len members.
88              */
89             read_data into(q->payload, q->len);
90         }
91
92         acknowledge that packet has been read();
93
94 #if ETH_PAD_SIZE
95         pbuf_header(p, ETH_PAD_SIZE); /* reclaim the padding word */
96 #endif
97
98         LINK_STATS_INC(link.recv);
99     }
100     else
101     {
102         drop_packet();
103         LINK_STATS_INC(link.memerr);
104         LINK_STATS_INC(link.drop);
105     }
106
107     return p;
108 }
```

LwIP的使用

LwIP的初始化:

LwIP的初始化必须在RTOS启动之后才可以进行, 因为它的初始化代码使用了一些OS提供的功能!!!

初始化代码示例:

```
1 extern err_t ethernetif_init(struct netif *netif);
2 struct netif lpc1788_netif;
3 ip_addr_t e0ip, e0mask, e0gw;
4
5 /* tcpip_init使用的回调函数, 用于判断tcpip_init初始化完成 */
6 static void tcpip_init_done(void *pd)
7 {
8     *(int *)pd = 0;
9 }
10
11 void ethernetif_input(struct netif *netif);
12 // 一直调用ethernetif_input函数, 从网卡读取数据
13 static void lwip_read_task(void *netif)
14 {
15     while(1)
16     {
17         ethernetif_input(netif);
18     }
19 }
```

写下你的评论...

评论0

赞1

...

简书

首页

下载APP

搜索

Q

Aa

beta

登录

注册

```
25
26 tcpip_init(tcpip_init_done, &flag); // lwip协议栈的初始化
27 while(flag);
28
29 IP4_ADDR(&ethip, 192,168,6,188); // 设置网卡ip
30 IP4_ADDR(&ethmask, 255,255,255,0); // 设置子网掩码
31 IP4_ADDR(&ethgw, 192,168,6,1); // 设置网关
32
33 //给lwip添加网卡
34 pnetif = netif_add(&lpc1788_netif, &ethip, &ethmask, &ethgw,
35 NULL, ethernetif_init, tcpip_input);
36 netif_set_default(pnetif); // 设置该网卡为默认网卡
37 netif_set_up(&lpc1788_netif); // 启动网卡, 可以唤醒DHCP等服务
38
39 // 创建一个任务. 这个任务负责不停的调用ethernetif_input函数从网卡读取数据
40 raw_task_create(&lwip_read_obj, (RAW_U8 *) "lwip_read", &lpc1788_netif,
41 CONFIG_RAW_PRIO_MAX - 25, 0, lwip_read_stk,
42 LWIP_READ_STK_SIZE, lwip_read_task, 1);
43 }
```

1人点赞 >

嵌入式 ...

"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下

嵌入式工作 擅长物联网, gprs, mqtt, ssl, tcp, BLE蓝牙开发, wifi , mt7628网...
总资产1 (约0.15元) 共写了2.7W字 获得12个赞 共24个粉丝

关注

写下你的评论...

全部评论 0

只看作者

按时间倒序 按时间正序

推荐阅读

更多精彩内容 >

2018-06-11

mean to add the formatted="false" attribute?.[46% 47325/...

ProZoom 阅读 704 评论 0 赞 1

stm23f1+ucosii移植lwip-1.3.2到lwip-2.0.2

0.从lwip官网下载lwip-2.0.2。 1.替换旧版本的api文件夹、core文件夹。 2.复制并替换旧版本...

pureer 阅读 624 评论 0 赞 0

写下你的评论...

评论0

赞1

...

u-boot启动流程分析(2)_板级(board)部分

1. 前言 书接上文（u-boot启动流程分析(1)_平台相关部分），本文介绍u-boot启动流程中和具体版型（b...

 梓枫 阅读 328 评论 0 赞 2

0x10000000	reserve_low
Trampoline	reserve_trampoline
(TOTAL_MALLOC_LEN) < 0x10000000	reserve_high
0x10000000	gd-relocaddr
(TOTAL_MALLOC_LEN)	reserve_malloc
Board Info struct	reserve_board
(board_info->mem_start)	reserve_global_data
new global data	
new lib(device tree)	reserve_fdt
arch-special memory	reserve_arch
arch-special stacks	reserve_stacks

建群宝：原有的微信群怎么批量管理？

运营微信社群的人都知道，在没有工具的时代，自己总会人肉管理的一批微信群，少则几个，多个几十个上百个，那么现在微信群...

 运营那些事儿uc 阅读 840 评论 0 赞 2

