marvelwifi

Public

master 1 Branch 0 Tags Go to file t Go to file Add file

eYoung8475 Modify the problem... 3157a6c · 3 years ago 39 Commits

FwImage	add chip: 88w8797 firmware.	5 years ago
image	add chip: 88w8797 firmware.	5 years ago
white	modify MLAN_MAX_VER_STR_...	5 years ago
mlinux	Remove business logic code	5 years ago
mport	Defect 1:	3 years ago
.gitignore	Initial commit	6 years ago
LICENSE	Initial commit	6 years ago
README.md	Update README.md	5 years ago
SConscript	Correct the correspondence b...	5 years ago
mwifi.c	Modify the problem of memor...	3 years ago

README GPL-2.0 license

About

Marvell WiFi driver for rt-thread

Readme

GPL-2.0 license

Activity

21 stars

3 watching

15 forks

Report repository

Releases

No releases published

Packages

No packages published

Contributors 3

JianRuiqian ruiqian

Marvell WiFi

Marvell WiFi is a sdio wifi driver software running on the RT-Thread real-time operating system.

Hardware Requirements

ROM: 512KB or above
RAM: 128KB or above

WiFi Chips Support

1. 88w8782
2. 88w8801

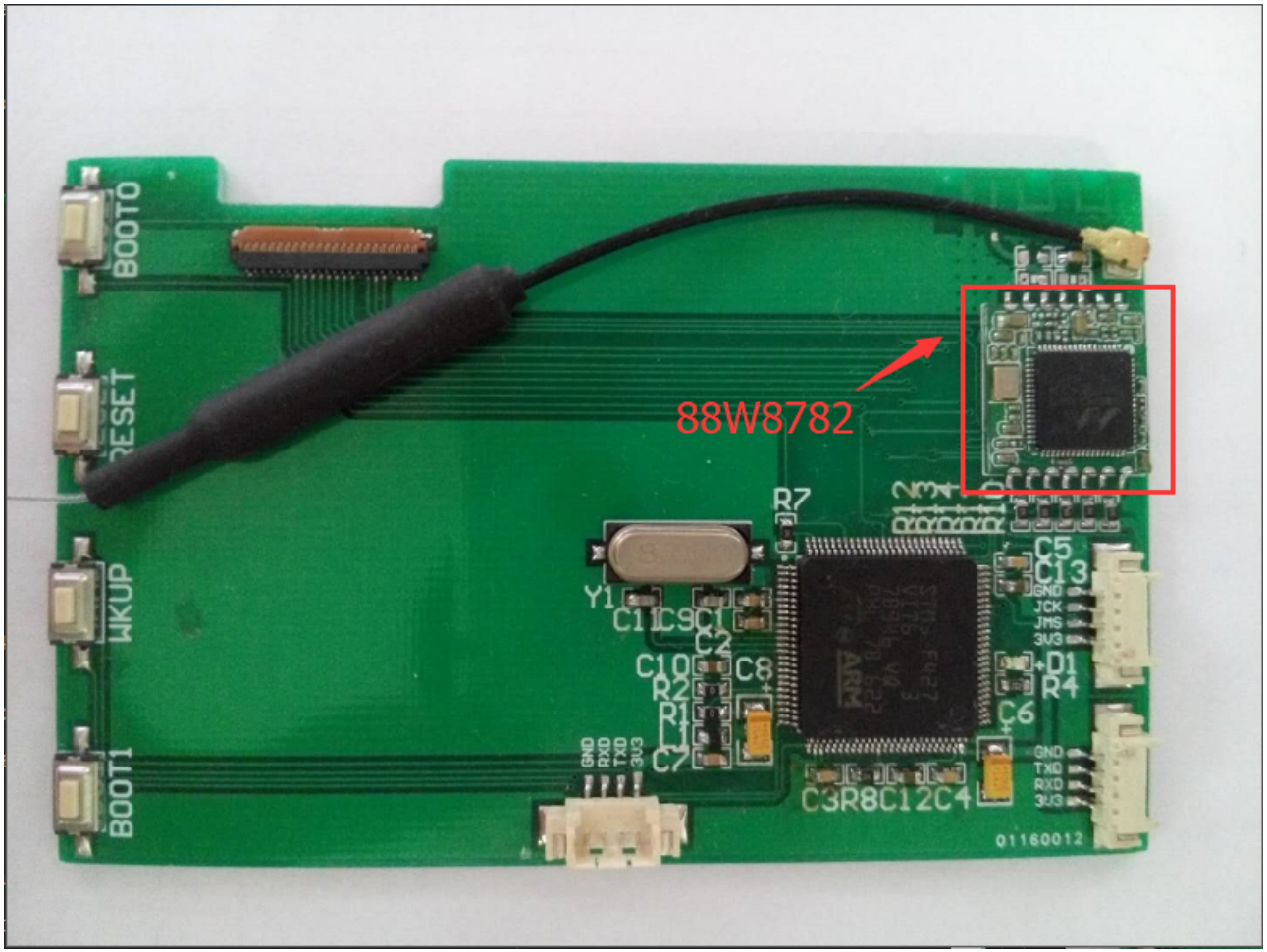
guangliangliao

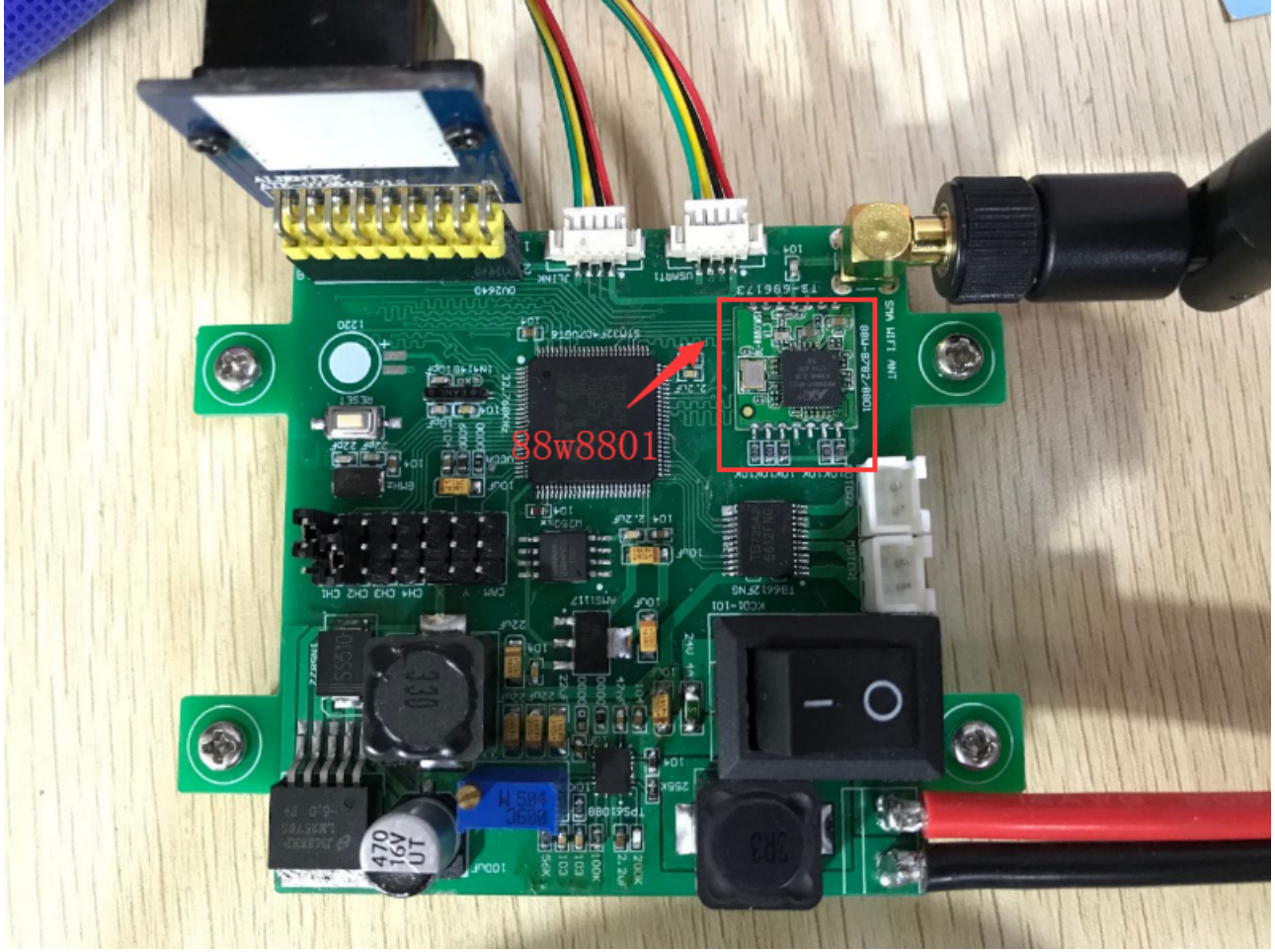
eYoung8475 linyiyang

Languages

C 99.3% Other 0.7%

3. 88w8797 (not tested yet)



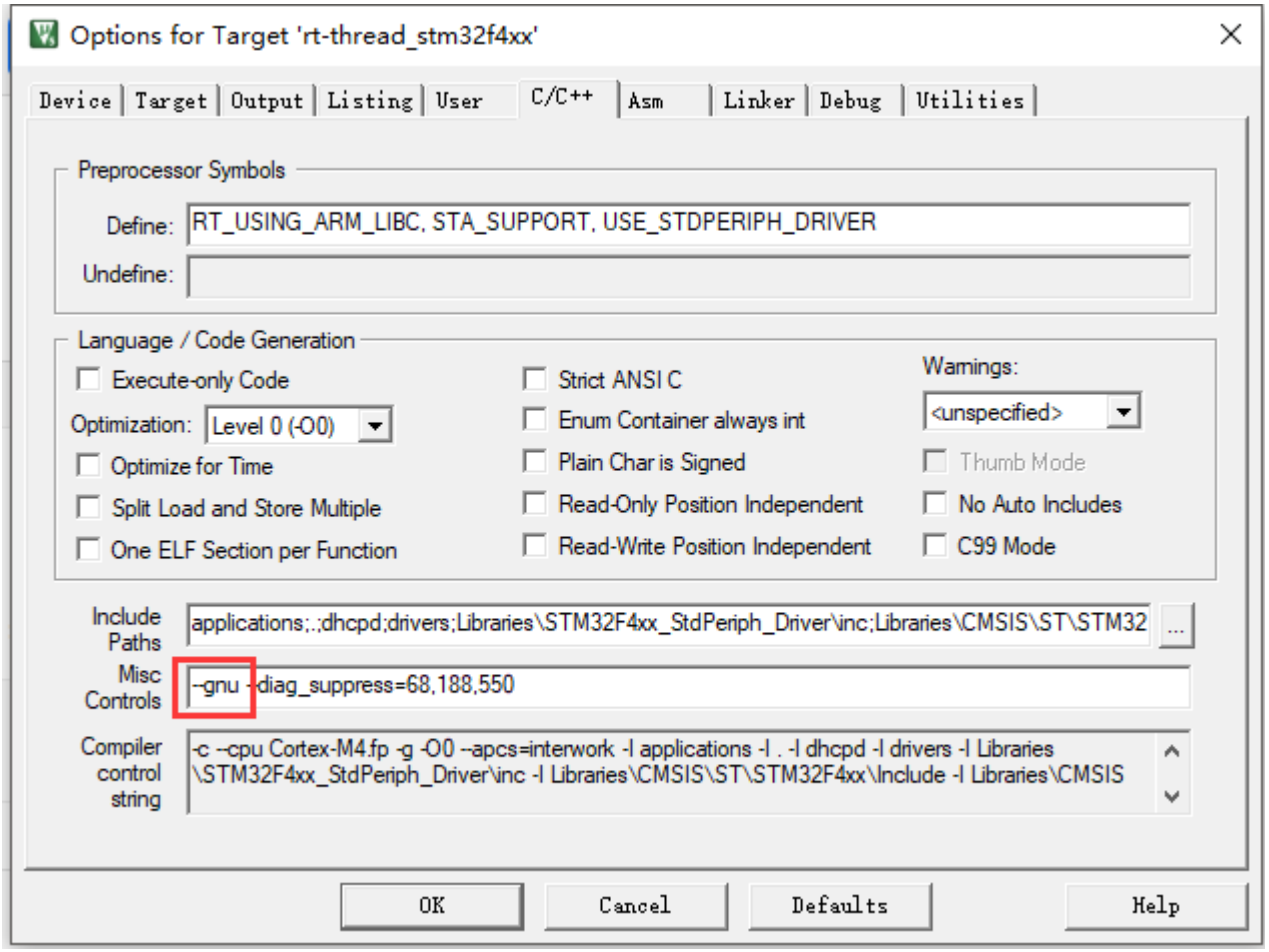


Features

1. STA, UAP mode (can coexist, but no routing)
2. Authentication method: OPEN/WPA-PSK/WPA2-PSK
3. Automatically reconnect
4. low power mode
5. High speed (stm32f407 driver can reach 2MB/s)

Compiler

1. You can use GCC to compile
2. Or add it to the MDK5 project and compile
(Note: Since the driver contains a lot of gcc features, please add --gnu to Misc Controls under the compiler C/C++ options): If you want to compile through scons, please specify toolchains as armcc in rtconfig.py and add it in CFLAGS --gnu):



```
elif PLATFORM == 'armcc':
    # toolchains
    CC = 'armcc'
    AS = 'armasm'
    AR = 'armar'
    LINK = 'armlink'
    TARGET_EXT = 'axf'

    DEVICE = ' --cpu=cortex-m4.fp'
    CFLAGS = DEVICE + ' --apcs=interwork -DUSE_STDPERIPH_DRIVER -DSTM32F40_41xxx --gnu'
    AFLAGS = DEVICE
    LFLAGS = DEVICE + ' --info sizes --info totals --info unused --info veneers --list rtthread-stm32.map --scatter stm32_rom.sct'

    CFLAGS += ' -I' + EXEC_PATH + '/ARM/RV31/INC'
    LFLAGS += ' --libpath ' + EXEC_PATH + '/ARM/RV31/LIB'

    EXEC_PATH += '/arm/bin40/'

    if BUILD == 'debug':
        CFLAGS += ' -g -O0'
        AFLAGS += ' -g'
    else:
        CFLAGS += ' -O2'

    POST_ACTION = 'fromelf --bin $TARGET --output rtthread.bin \nfromelf -z $TARGET'
```

Components Dependence

1. sdio driver framework (RT_USING_SDIO)
2. Lwip protocol stack (RT_USING_LWIP)
3. dfs virtual file system (RT_USING_DFS)
4. libc library (RT_USING_LIBC)
5. rt_hw_us_delay (please implement it yourself in bsp)
6. sdio host driver (please implement it yourself in bsp)
7. dhcpd protocol (LWIP_USING_DHCPD optional, used in UAP mode)

Adding Method

Use the env tool officially provided by RT-Thread to obtain the package and generate the project
Or manually download the package and add it to the existing project directory, enable the following macros in rt_config.h, and use the scons tool to regenerate the mdk project:

```
#define PKG_USING_WLANMARVELL
#define MARVELLWIFI_USING_STA
```

Initialize

Step one: Hard reset the wifi chip, which can be achieved by connecting to the MCU reset circuit or through GPIO control.
Step 2: If you use component initialization, you only need to enable the following macro:

```
#define RT_USING_COMPONENTS_INIT
```

Otherwise, please manually initialize other components on which this driver depends before calling

```
mwifi_system_init();
```

Note, before using it for the first time, please create a new directory in the target board file system: '/mrvl', and put the firmware in the FwImage folder in the package into this directory.

名称	修改日期	类型	大小
fw_version.txt	2017/6/29 15:18	文本文档	1 KB
sd8782_uapsta.bin	2014/5/7 3:00	BIN 文件	249 KB
sd8801_uapsta.bin	2017/6/29 15:14	BIN 文件	252 KB