RAK439 SPI WiFi Module

NOS Programming Precautions V1.1

Shenzhen Rakwireless Technology Co., Ltd.

www.rakwireless.com

info@rakwireless.com

Content

1. rw_sysDriverLoop

The program needs invoking rw_sysDriverLoop circularly or regularly, so as to ensure that users' commands and received events can be processed timely. The program has the following functions:

1) Function implementation

As all APIs in RAK439 are non-blocking ones, so the specific implementation of some functions needs to be continued in the loop.

2) Events query

The loop sends a command to query abnormal events, including network connection and disconnection, TCP disconnection, data acceptance and TCP client connection.

3) Call back

2. rw_sysDriverReset

In case the module is abnormal or its network is disconnected, the function should be invoked to reset the driver.

3. Socket Return value of socket function

1) RW_ERR
Internal errors
Such errors will be returned in case of not initialization of the driver, network disconnection or memory abnormality.
Correlation function：
socket(),bind(),connect(),listen(),select(),accept(),send/sendto(),recv/recvfrom()
close()/shutdown()

2) RW_ERR_SOCKET_INVAILD
invalid socket handle
Correlation function：bind(),connect(),listen(),select(),accept(),send/sendto(),recv/recvfrom()
close()/shutdown()

3) RW_ERR_CMD_PENDING
In case the socket command or other socket commands are under implementation.
As all socket functions in RAK439 are non-blocking ones, the functions need to be invoked for many times to finish the operation. For the function which is invoked for the first time and completes the command, whether the operation is finished will be queried; if operation is not

finished, the function will return to RW_ERR_CMD_PENDING, and when it will be invoked for another time, whether operation is finished will be queried directly. The program will keep invoking the function until the operation is over (returning to RW_OK, RW_ERR, RW_ERR_SOCKET_INVAILD, RW_ERR_TIME_OUT); if the operation is not finished, the program will continue to invoke other socket functions and return to RW_ERR_CMD_PENDING.
Correlation function：bind(),connect(),listen(),select(),accept(), close()/shutdown()

4) RW_ERR_TIME_OUT
In case the socket function operation is overtime or waiting events of the select function are overtime.
Except the select function, other functions return to RW_ERR_TIME_OUT. It is suggested that the driver and memory be checked for any problem.
Correlation function：bind(),connect(),listen(),select(),accept(), close()/shutdown()

5) RW_ERR_SEND_BUFFER_FULL
In case data transmission fails, for the last datum is not sent out or data acceptance is too fast, leading to the need for data transmission for another time.
Correlation function：send/sendto()

4. Use of socket functions

1) select()
The socket functions support only one socket waiting event, and do not support FD_ZERO/FD_CLR/FD_SET/FD_ISSET. If the set time is exceeded, the functions will return to RW_ERR_TIME_OUT.

2) recv()/recvfrom()
The internal cache of the driver for acceptance is relatively small, so if the driver receives data while the upper level does not fetch the data timely, the acceptance idle queue of the driver may be empty, events and subsequent data may be blocked, and the command may not receive returned data. Thus, if data are received in the application, they should be read timely.
Acceptance can be checked by polling the functions or use select waiting data.

3) send()/sendto()
If transmission meets returning to RW_ERR_SEND_BUFFER_FULL, the reason for which may be discard of the driver layer due to fast data acceptance. If continuous data need to be accepted, do not try to send data continuously in the application, which may cause module abnormality.
If returning to 0 is sent, the parameters should be checked whether to be legal or not.

4) close()/shutdown()
When two functions are the same.
When the functions return to RW_OK, the interior of the driver is not closed actually. The driver will be closed automatically after waiting for several loops.

5) socket()

The functions will not be blocked, and will only return to RW_ERR and effective socket handle. Do not invoke the functions continuously.

Version

Version	Content modification	Date
V1.0	Create a document	2015-07-19
V1.1	Modify some of the details	2015-07-20