RAK473M UartDriver

User Manual V1.0

Shenzhen Rakwireless Technology Co., Ltd

www.rakwireless.com

info@rakwireless.com

List of Content

1. Overview

RAK473 UartDriver is a non-blocking processing drive of AT command. After MCU serial port sending a command, if which is a command (commands of query) can be timely responded by the WiFi module, you can block it and respond to it after reading the AT command; if the module needs to wait for a period of time before response, then do not block it and wait for the response, meanwhile, firstly implement other user programs, such as scanning button, liquid crystal display, etc. and then check whether the AT command has returned.

2. Command processing

The AT command can be timely responded is processed as follows:

```
Send AT command
        ↓
   Module Response? —N→ Timeout? —N→ (back to Send AT command)
        │Y              │Y
        ↓←──────────────┘
AT command over
```
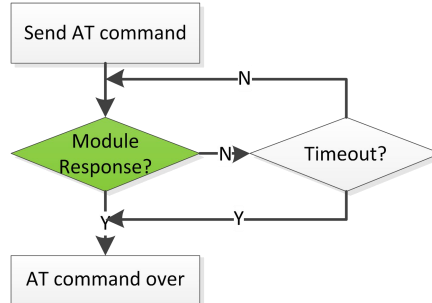
The AT command cannot be timely responded are processed as follows:

```
start
  ↓
Cmd block? —Y→ Module Response? —N→ Timeout? —N→ Ret = cmd pend
  │N              │Y                  │Y                │
Send AT command  Clr cmd block status Clr cmd block status
  ↓               ↓                    ↓                 │
Set cmd block status Ret = cmd response Ret = cmd timeout │
  ↓                                                       │
Ret = cmd pend                                            │
  └──────────────────→ return Ret ←──────────────────────┘
```
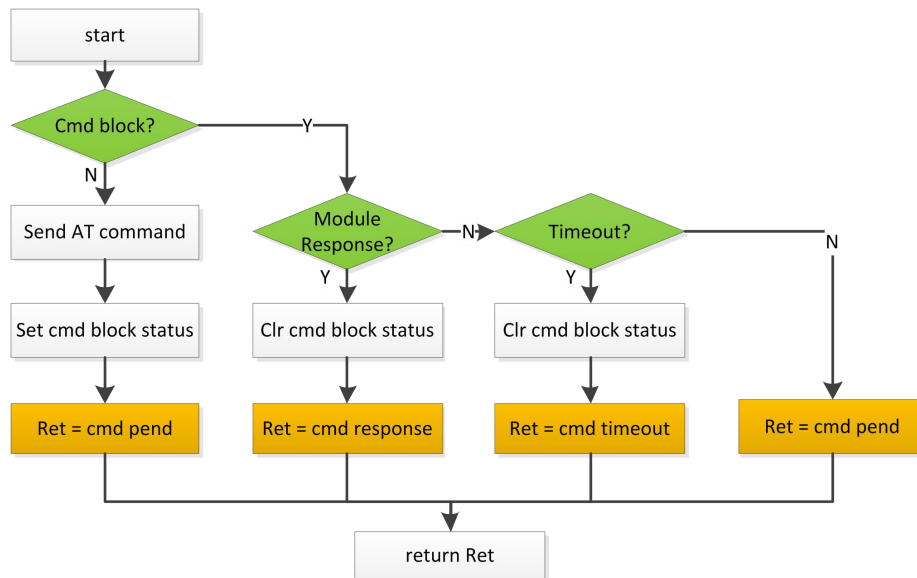
3. Function description

Function description:
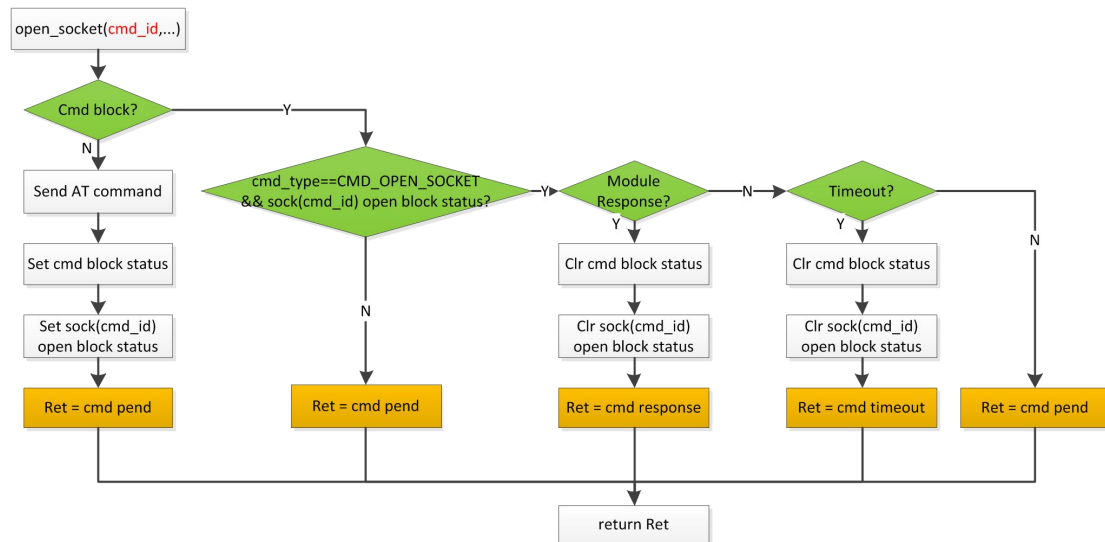
void read_data_from_module(void)

This function is used for resolving data received by the serial port, which requires circulation or regular invoking.

int open_socket(uint8_t cmd_id, …)

Non-blocking creating of socket (tcp,ltcp,udp,ludp,multicast), cmd_id increases by 1 with every one creation of socket.

The function process is as follows:

int write_socket(int fd, uint16_t dest_port, uint32_t dest_ip, uint16_t send_len, uint8_t *buf)

Non-blocking data sending, after sending the AT command, return immediately, and afterwards circularly execute this function to query the response value of AT command.

int read_socket(int fd, uint8_t* buf, uint16_t maxlen, uint32_t *remoteip, uint16_t *remoteport)

Reading data with non-blocking under interrupt mode.

Reading data with blocking under query mode.

int query_socket_status(int sock_fd)

Query of socket status with blocking.

int query_con_status(void)

Query of net-connection status with blocking.

4. Socket communication notice

▲ Each AT command cannot execute other AT commands before receiving response from the wifi module.

For TCP Server \ Client, who received disconnection event during communication process, they need to firstly determine whether the write_socket function has received the WiFi module response (the AT command is blocked or not), if response is not received, do not invalidate the socket descriptor until the write_socket function returns CMD_ERR_SOCK_INVALID; directly invalidate the socket descriptor if the write_socket function has received response from the wifi module (the AT command unblocked).

▲ Ensure each frame of data received from the serial port being correctly processed.

For TCP client, who has received disconnection event in the communication process, if the current socket descriptor has expired but the receiving buffer zone still has data unread, you need to firstly execute the read_socket function to finish reading the data and then to execute the open_socket function to reconnect the Server.

5. AT command response - interrupt mode vs query mode

The AT command response types of RAK473 module consist of interrupt mode and query mode

Interrupt mode -- network disconnection event, socket disconnection event, data receiving will all actively get MCU informed.

Query mode -- network disconnection event, socket disconnection event, data receiving will not actively inform MCU, and MCU needs to circularly or regularly sending AT command for query.

6. Modification records

Edition	Author	Date	Modification content
V1.0	harry	2016/1/27	Creating document