QSTM32
Network Application Note

Confidentiality Level: (Tick the Box ■)		
Top Secret ☐	Confidential ☐	Public ☐

About the Document

Revision History

Revision	Date	Author	Description
-	2023-08-20	Foldy WANG	Initial version
-	2023-11-05	Linkin WANG	Added Network Initialization process flow chart
1.0	2023-11-25	Linkin WANG	Network initialization process modification
2.0	2025-08-29	Wells Li Linkin Wang	The second generation initial version

Contents

About the Document 2
Contents 2
1 Purpose 3
2 Scope 3
3 API Design 3
4 Network Application Workflow 5
5 Network Exception Handling 6
6 Appendix A Reference 7

1 Purpose

Cellular network modules provide connectivity capabilities such as LTE GSM,CAT-M1, and NBIOT. This document offers essential knowledge about cellular technologies and serves as a practical guide for achieving a rapid and successful network attachment.

2 Scope

This document is applicable to products with Quectel module mounted on MCU.

3 API Design

Quectel provides various APIs that implement network registration and attachment by leveraging corresponding AT commands. See **Table 1** in detail:

Table 1: Network API reference design

API	Function
ql_net_init()	Initialize the network module (e.g., initializes parameters, allocates resources, establishes a connection with the module).
ql_net_set_opt()	Configure network parameters (e.g., sets APN, selects network mode).
ql_net_attach()	Perform network registration and activates a data connection (PDP context activation).
ql_net_get_rssi()	Query the Received Signal Strength Indication (RSSI)
ql_net_get_ip()	Get the local IP address assigned by the network.
ql_net_reconnect()	Re-establish the data connection by detaching from and then re-attaching to the cellular network.
ql_module_reboot()	Perform a software reboot of the module.
ql_net_deinit()	De-initialize the network stack and release resources

For specific information of API, please refer to the document: Quectel_QSTM32_SDK_API_Design_V2.0

AT command reference table corresponding to API as below.

Table 2: AT command reference table

API	AT Command
ql_net_set_opt()	AT+QCFG=
ql_net_attach()	AT+CREG?/ AT+CEREG?/ AT+QIACT?/ AT+QIACT=
ql_net_get_rssi()	AT+CSQ
ql_net_get_ip()	AT+QIACT?
ql_net_reconnect()	AT+CFUN/ ql_net_attach()
ql_module_reboot()	AT+CFUN=”1,1”

4 Network Application Workflow

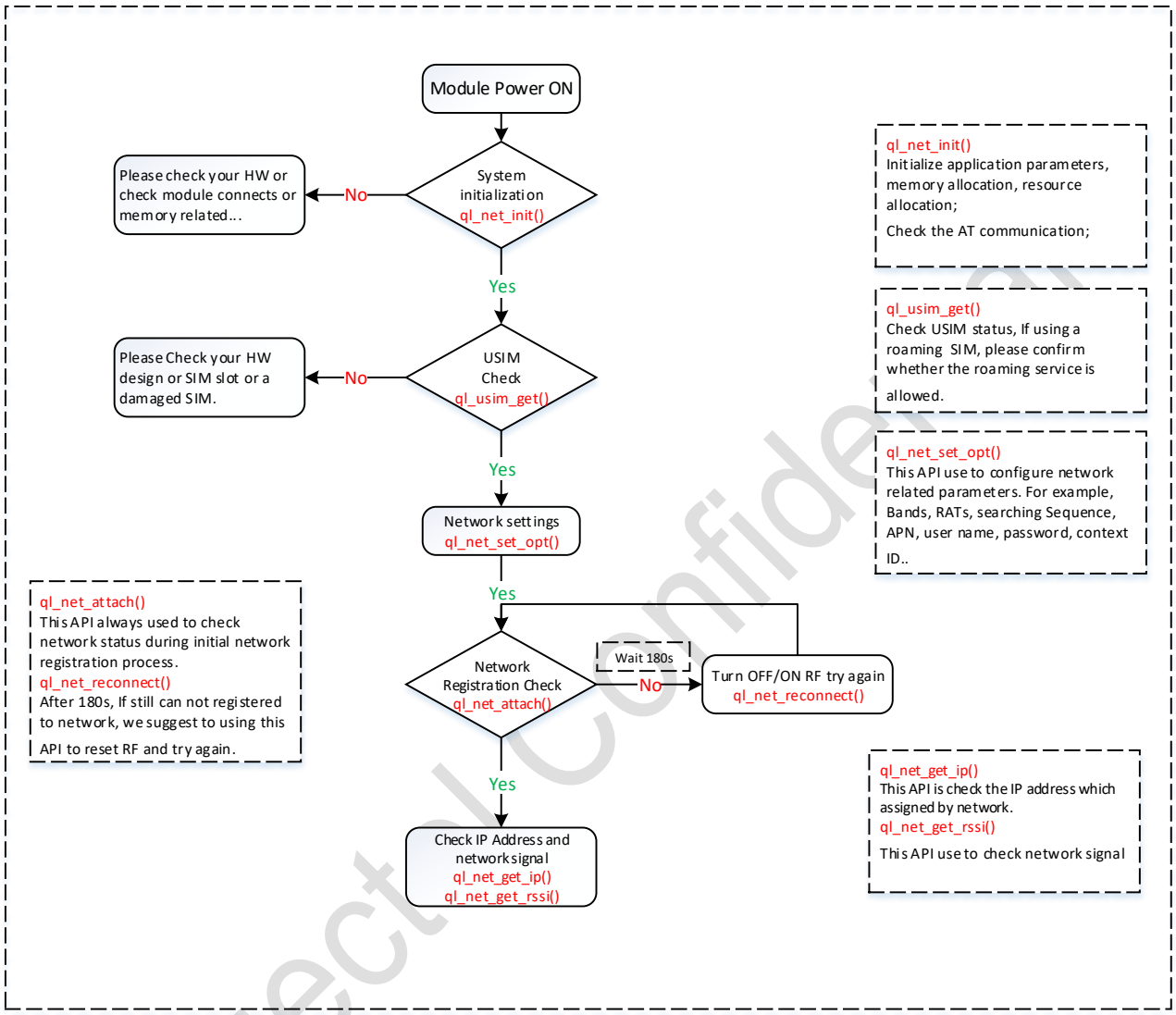

Figure 1: Network Application Workflow

a) Call ***ql_net_init()***, the application program will initialize parameters, allocate resources and check whether AT command is normal to confirm that the module has initiated normally.
b) Once all initializations are successful and completed, please check if the sim card is normal by calling ***ql_usim_get()***.
c) Following that, configure network parameters via ***ql_net_set_opt()***, including APN, username, password, network mode and so on.
d) After all network parameters are configured, network search will operate. During the network search process, you can use the ***ql_net_attach()*** API to query whether the network registration is successful and PDP is activated. If registration is still unsuccessful after 180 seconds, it is recommended to turn RF off and on again via ***ql_net_reconnect()*** and attempt later.
e) After network is attached, you can get IP address RSRP by calling ***ql_net_get_ip()*** and

ql_net_get_rsrp() respectively.

5 Network Exception Handling

You may encounter various problems during network registration. This section provides troubleshooting tips and suggestions for network registration failures.

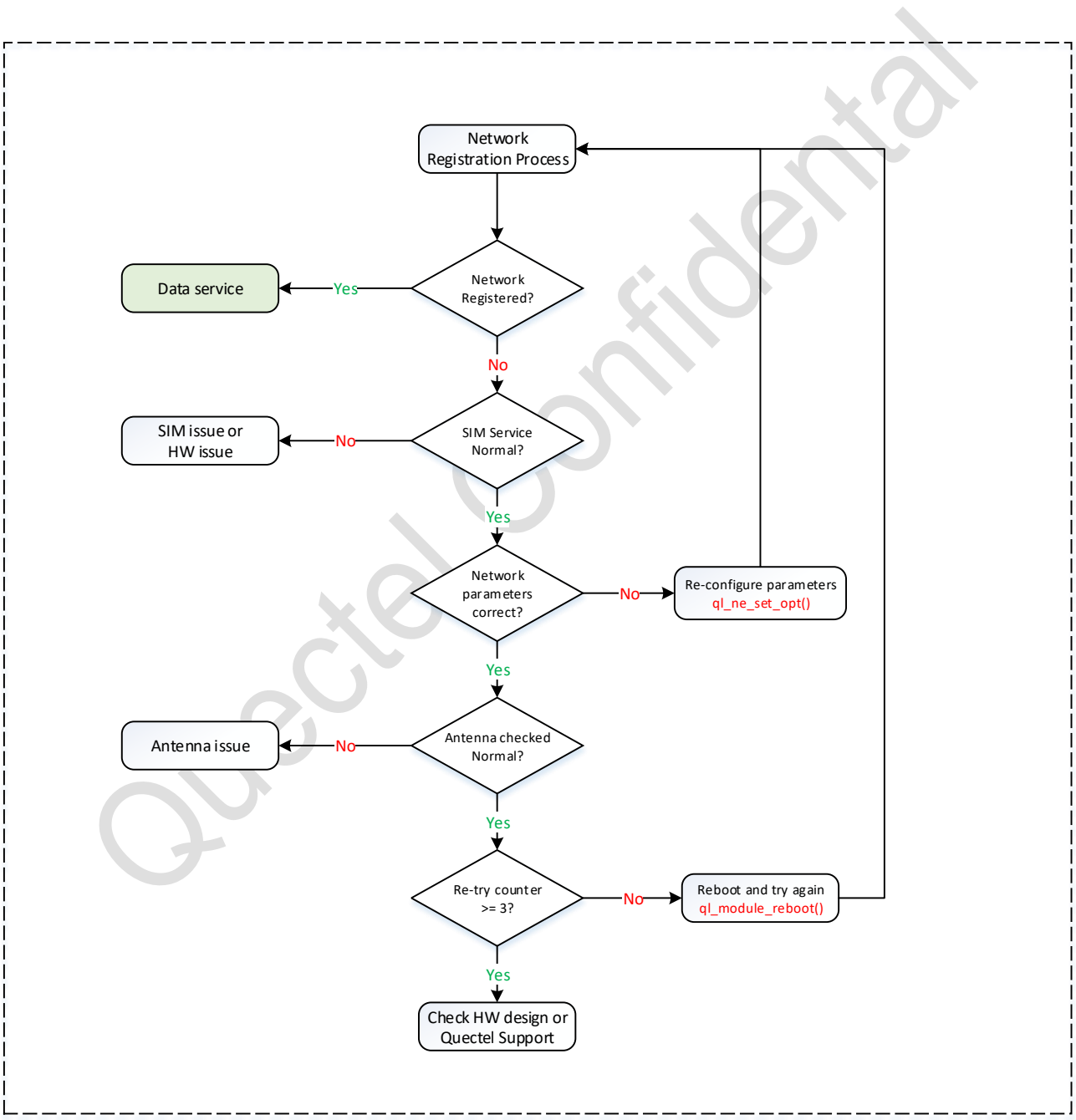

Figure 2: Network Exception Handling flowchart

6 Appendix A Reference

Table 3: Related Documents

SN	Document Name	Remark
[1]	Quectel_QSTM32_SDK_API_Design_V2.0	APIs Introduction

Table 4: Terms and Abbreviations

Abbreviation	Description
LTE	Long-Term Evolution
GSM	Global System for Mobile Communications
CAT-M1	Category M1
NBIOT	Narrow-Band Internet of Things
SIM	Subscriber Identity Module
RSRP	Reference Signal Received Power
APN	Access Point Name
MCU	Microcontroller Unit
RF	Radio Frequency
API	Application Programming Interface