QSTM32
HTTP(S) Application Note

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

Document Control Records

Revision History			
Date	Revision	Revision Description	Author
2023-12-04	0	Initial	Linkin Wang
2023-12-25	1	Modified by Copywriter Team	Linkin Wang Kelly Chen
2025-08-29	V2.0	The second generation initial version	Wells Li Linkin Wang

Contents

1 Purpose

HTTP(S) is one of the most common protocols that used in web project, it's very convenient and safe to post or get data from remote HTTP(S) server. And in the development of IoT projects, it is also widely used for sending data to remote HTTP(S) server or getting remote data to UE.

The Quectel cellular network module provides a HTTP(S) protocol stack, which can be used to guide MCU developers in quickly developing network function.

2 Scope

This document applies to products with MCU mounted with quectel module.

3 Terms and Definitions

Quectel: Quectel Wireless Solutions Co., Ltd.
HTTP: Hypertext Transfer Protocol
UE: User Equipment
SSL: Secure Socket Layer
CA: Certificate Authority

4 API Design

Quectel has designed a set of reference APIs that utilize module's AT commands to implement data transmission and reception functions for HTTP(S). The details are listed in Table 1.

Table 1: HTTP API Reference Design

API	Functionality
ql_http_init()	Initialize the HTTP(S) client instance.
ql_http_setopt()	Set HTTP(S) client options
ql_http_set_ssl()	Set SSL configuration for an HTTPS client.
ql_http_request()	Send HTTP(S) request.
ql_http_recv()	Receive HTTP(S) response data.
ql_http_deinit()	De-initialize the HTTP(S) client instance

For specific design on APIs, please refer to following appendix.
Quectel_QSTM32_SDK_API_Design_V2.0

See AT commands corresponding to API as shown below.

Table 2: AT Command Reference

API	AT Command
ql_http_setopt()	AT+QHTTPCFG
ql_http_set_ssl()	AT+QSSLCFG
ql_http_request()	AT+QHTTPPOST/AT+QHTTPGET/AT+QHTTPPUT/ AT+QHTTPREAD

In following page, we will explain how to use these APIs to send post or get request to remote server.

5 HTTP(S) Application Work Flow

5.1. HTTP(S) Communication Process

The quectel module supports HTTP(S) protocol stack, which is capable to post data to HTTP(S) server or get data from HTTP(S) server. The MCU only needs to handle data transmission, data processing, and the API defined in **Chapter 4**.

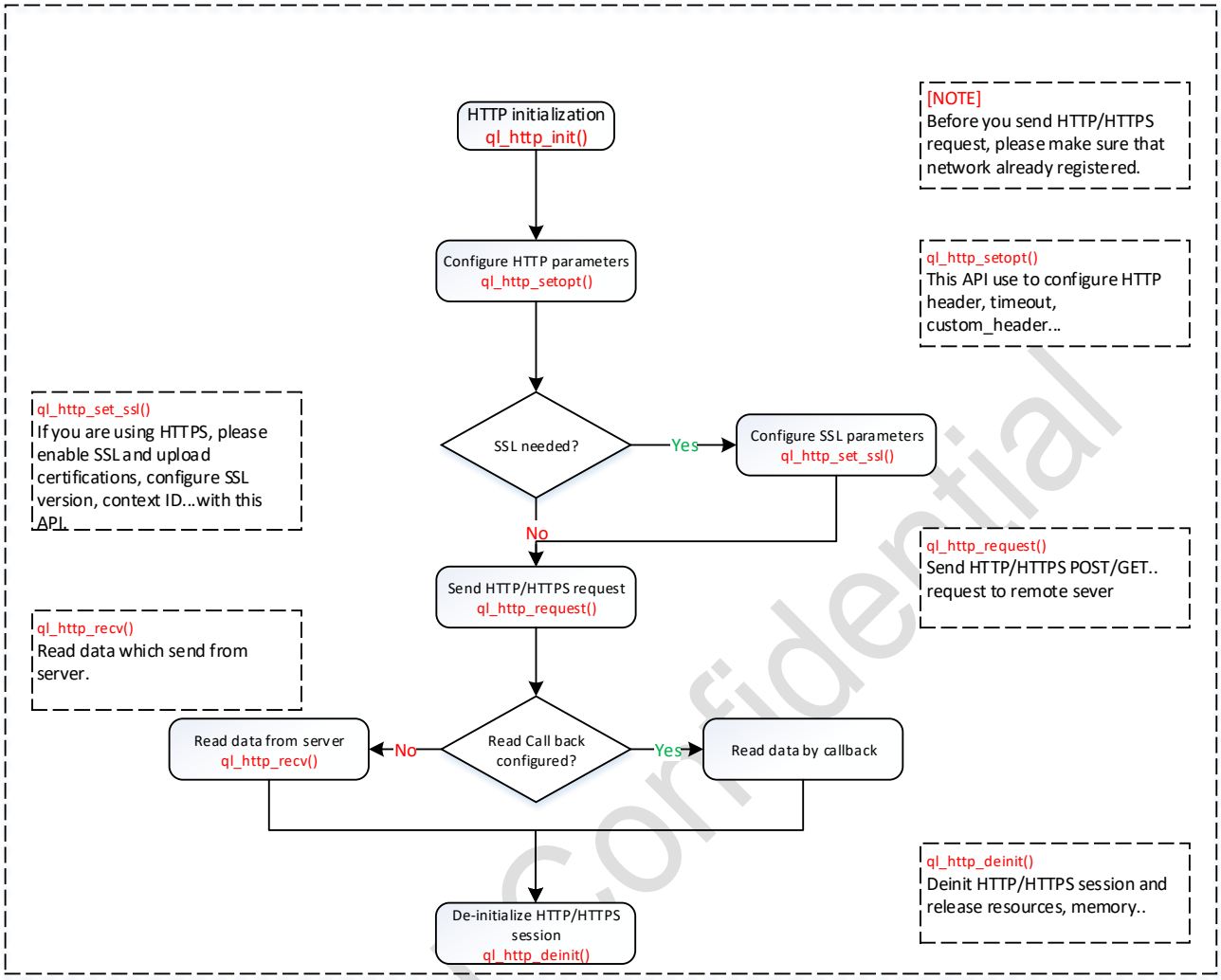

Figure 1: HTTP(S) Application Flow Overview

a) Call *ql_http_setopt()* to configure HTTP parameters, including the request/response header, timeout, read callback, custom header and so on.
b) As an option, please call *ql_http_set_ssl()* to set SSL-related parameters such as encryption mode (unilateral or bilateral), certifications, certification path, and SSL context ID if it is needed to enable it.
c) Call *ql_http_request()* to send HTTP(s) request to remote server.
d) Once it is a success to get response from remote server, if you set read callback function by *ql_http_setopt()*, you can read response data by callback function, otherwise please use *ql_http_recv()* to read response data.

5.2. HTTP(s) Application Example

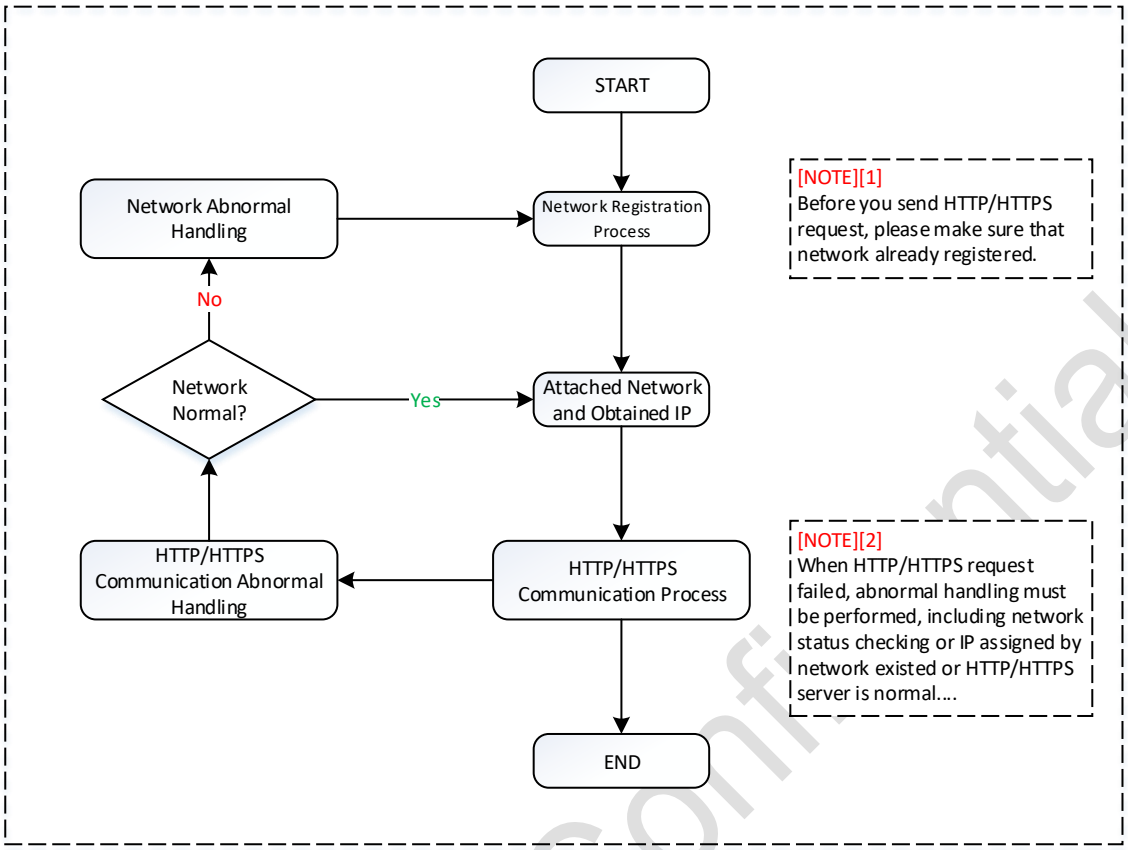

Figure 2: HTTP(S) Application Example

6 HTTP(s) Exception Handling

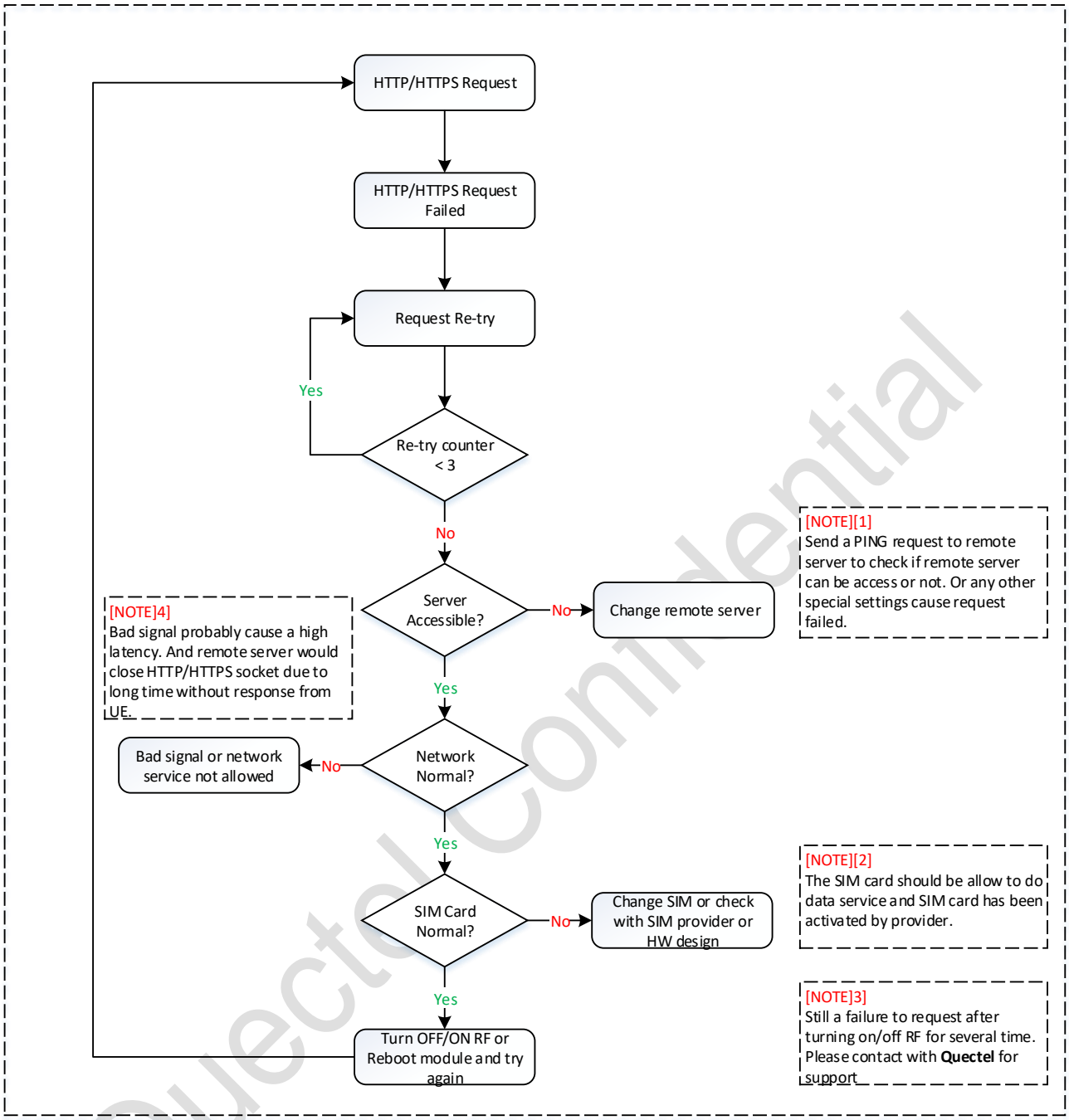

Figure 3: HTTP(S) Exception Handling

7 Appendix A Reference

Quectel_QSTM32_SDK_API_Design_V2.0