Quectel BG95

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

Contents

1 Purpose

HTTP(S) is one of the most common protocols and 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 BG95 cellular network module provides a HTTP(S) protocol stack, which can be used to guide MCU developers in quickly completing network function development.

2 Scope

This document applies to products with MCU mounted with BG95 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 BG95'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_initial()	HTTP initial parameters setting.
QL_http_ssl_initial()	SSL initial parameters setting.
QL_http_request()	Send HTTP(S) request.
QL_http_read()	Read HTTP(S) server response.

For specific design on APIs, please refer to following appendix.
Quectel_BG95_HTTP(S)_API_design.docx

See AT commands corresponding to API as shown below.

Table 2: AT Command Reference

API	AT Command
QL_http_initial()	AT+QHTTPCFG
QL_http_ssl_initial()	AT+QSSLCFG
QL_http_request()	AT+QHTTPPOST/AT+QHTTPGET/AT+QHTTPPUT
QL_http_read()	AT+QHTTPREAD

In the 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 BG95 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**.

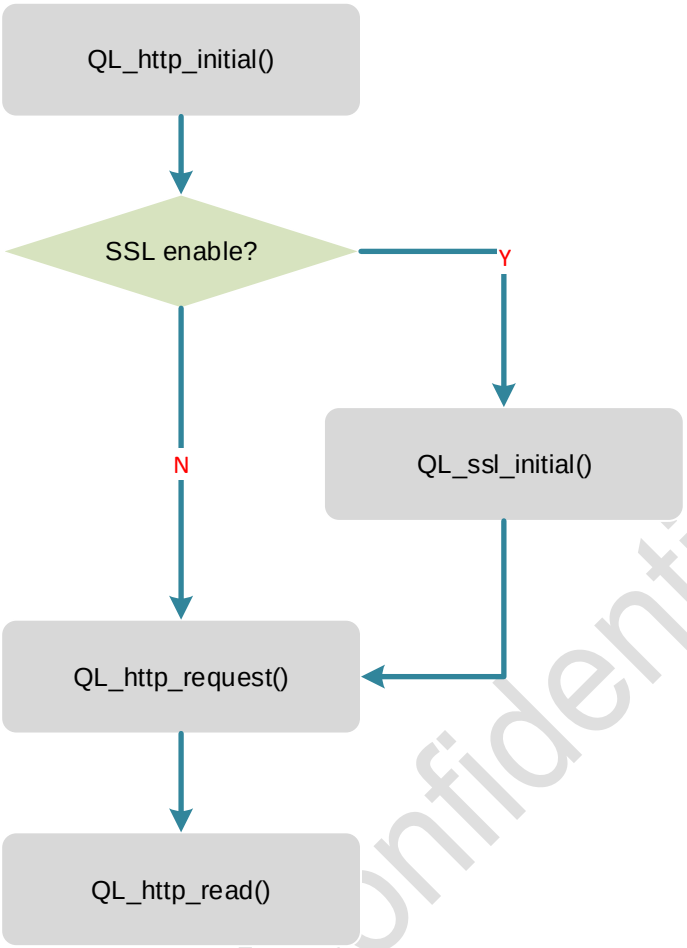

Figure 1: HTTP(S) Application Flow Overview

a) Call QL_http_initial() to configure initial HTTP parameters, including the request/response header, username, password, custom header and SNI etc.
b) As an option, please call QL_ssl_initial() 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, please call QL_http_read() to read data sent by remote server.

5.2. HTTP(s) Application Example

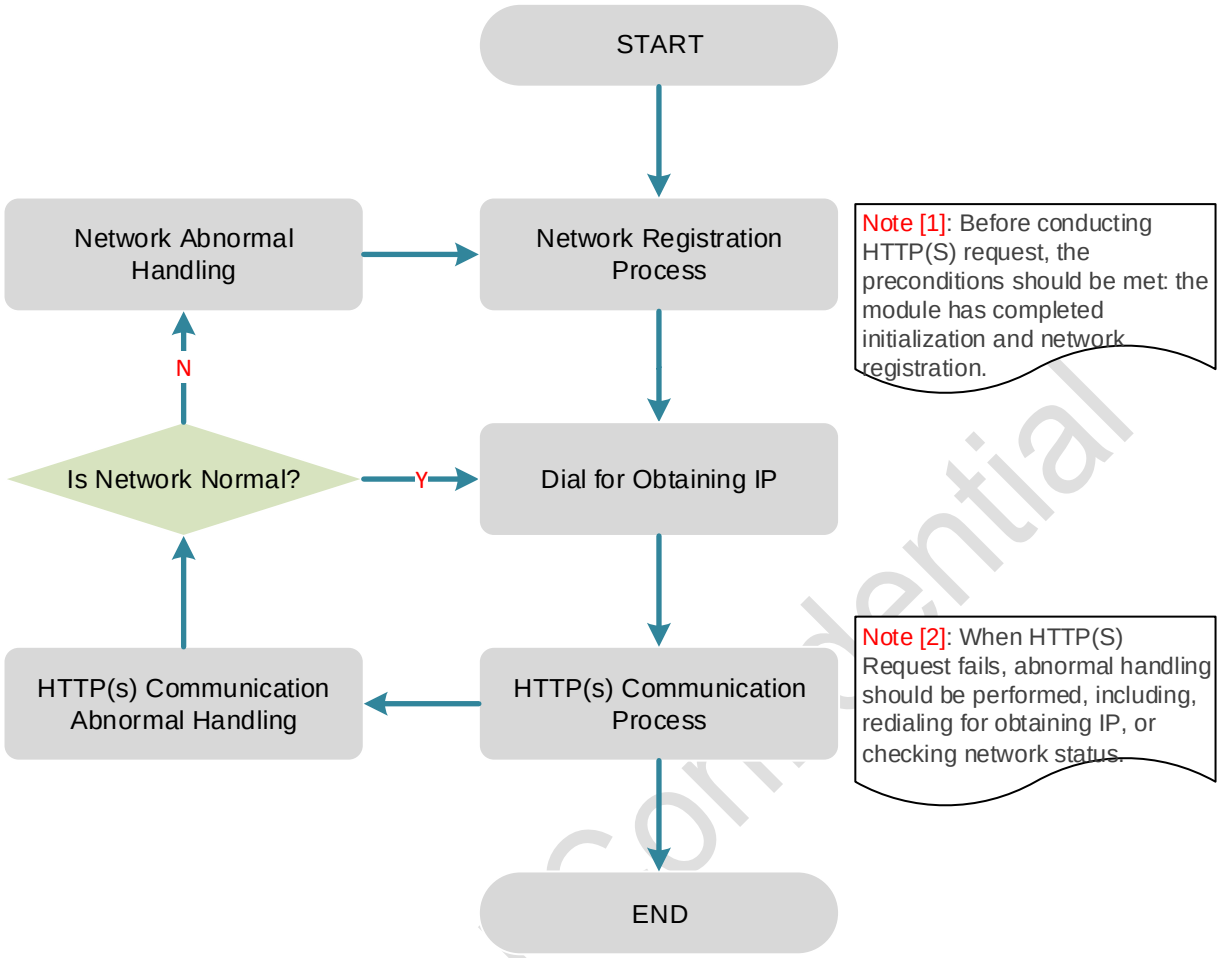

Figure 2: HTTP(S) Application Example

6 HTTP(s) Exception Handling

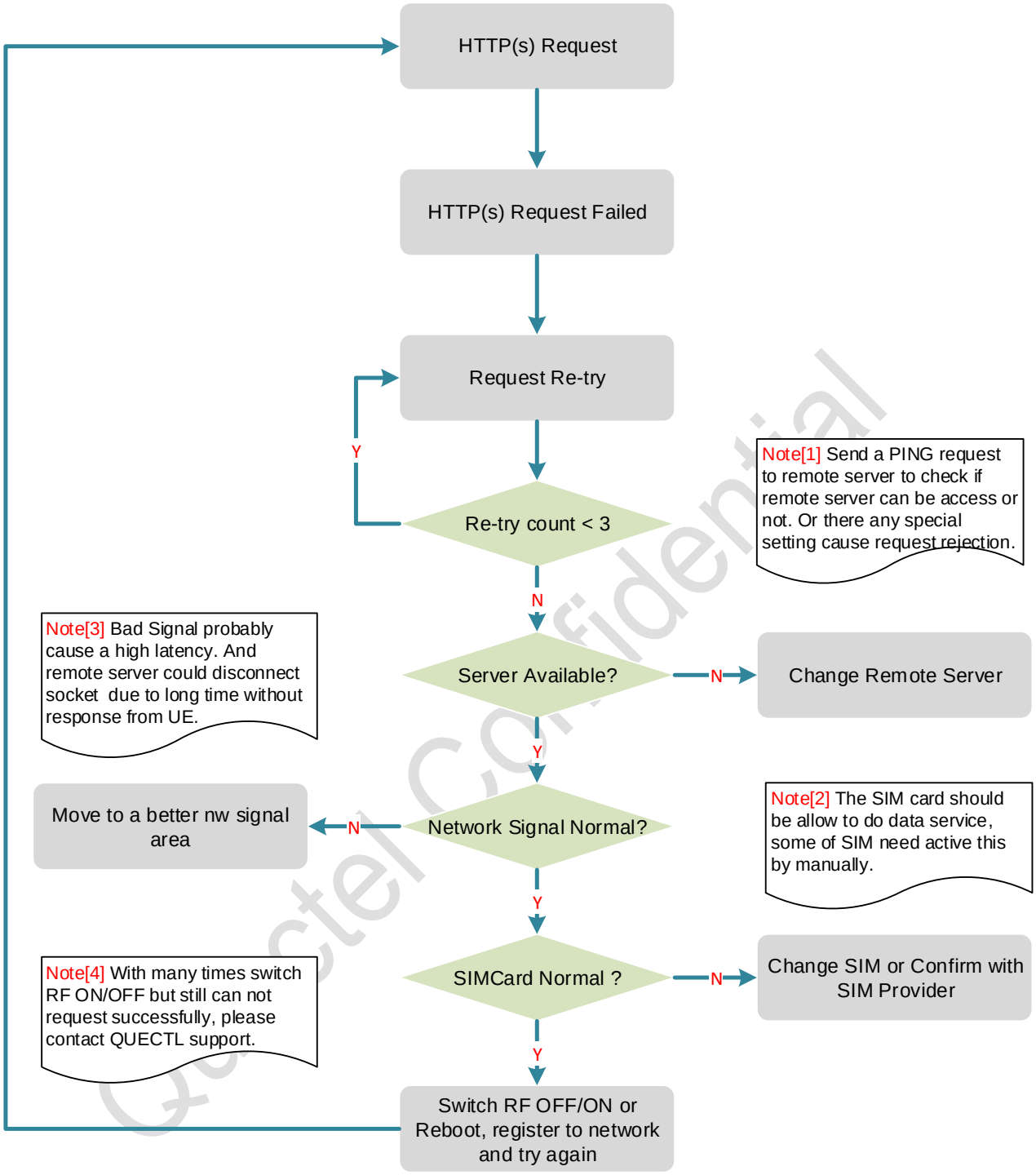

Figure 3: HTTP(S) Exception Handling

7 Appendix A Reference

Quectel_BG95&BG77&BG600L_Series_HTTP(S)_Application_Note_V1.1.pdf
Quectel_BG95_HTTP(S)_API_design.docx