NB-IoT Modules
FTP and HTTP(S)
Application Notes

Copyright © Neoway Technology Co., Ltd 2018. All rights reserved.

Notice

This document provides guide for users to use NB-IoT Modules.

This document is intended for system engineers (SEs), development engineers, and test engineers.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE DUE TO PRODUCT VERSION UPDATE OR OTHER REASONS.

EVERY EFFORT HAS BEEN MADE IN PREPARATION OF THIS DOCUMENT TO ENSURE ACCURACY OF THE CONTENTS, BUT ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT DO NOT CONSTITUTE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

Neoway provides customers complete technical support. If you have any question, please contact your account manager or email to the following email addresses:

Sales@neoway.com

Support@neoway.com

Website: http://www.neoway.com

Contents

About This Document

Scope

This document is applicable to NB-IoT Modules.

Audience

This document is intended for system engineers (SEs), development engineers, and test engineers.

Change History

Issue	Date	Change	Changed By
1.0	2018-11	Initial draft	Cenny

Conventions

Symbol	Indication
⚠️ Warning	This warning symbol means danger. You are in a situation that could cause fatal device damage or even bodily damage.
⚠️ Caution	Means reader be careful. In this situation, you might perform an action that could result in module or product damages.
⚠️ Note	Means note or tips for readers to use the module

1 Overview

This describes how to use the FTP and HTTP(S) functions and their AT command processes.

2 Initialization Process

To use the FTP and HTTP(S) functions after the modules is powered up and started, ensure that the module is initialized and registers with a network.

Figure 2-1 Initialization and network check process

```
                        ┌──────────────┐
                        │   Power on    │
                        └───────┬──────┘
                                │
                        ┌───────┴──────┐       ● +PBREADY is returned if a fixed baudrate is used
                        │    Delay      │- - -> ● Delay for 5 seconds to 10 seconds if baudeate is detected automatically.
                        └───────┬──────┘
                                │
          ┌────────> ┌──────────┴──────┐
          │          │    Send AT       │
        N │          └───────┬──────────┘
          │                  │
          │          ◇───────┴──────◇       ● If UART returns OK, it is ready.
          │          │  Receive "OK"? │- - -> ● If UART does not return code, keep sending AT every second for 10 times.
          │          ◇───────┬──────◇       ● If UART does not return code after 10 times, restart the module.
          │                  │
          │          ┌───────┴──────────┐    ● If the module returns +CPIN: READY, it identifies SIM card.
          │          │ Send AT+CPIN?    │    ● If the module returns +CPIN: <Code>, enter the PIN code or PUK code to unclok
          │          └───────┬──────────┘      the SIM card and then query again.
  Input PIN/PUK               │               ● If the module returns +CPIN: ERROR, +CME ERROR, or ERROR, the module
  code to unlock              │                 does not identify the SIM card. It might take long time for some cards to be
          │          ◇───────┴──────◇           identified, so send the command every second for 10 times.
          │          │ +CPIN: READY │- - ->  ● If error is returned for 10 times, check or replace the SIM card and restart the
          │          ◇───────┬──────◇           module.
          │                  │               ● Tips:
          │                  │                  A. PUK code is required after PIN code is input incorrectly for three times.
          │          ┌───────┴──────────┐       B. The SIM card will be locked forever if PUK code is input incorrectly for 10
          │          │  Send AT+CSQ     │    times.
          │          └───────┬──────────┘
        N │                  │               ● If the module returns +CSQ: 99,99 that indicates no signal, send the command
          │          ◇───────┴──────◇           every second for 40 times.
          │          │   +CSQ: X     │- - ->  ● If the module returns 0<X≤31 that indicates signal, perform next step till the
          │          ◇───────┬──────◇           value is not smaller than 12.
          │                  │               ● If no signal after querying for 40 times, check the antenna and restart the module.
          │          ┌───────┴──────────┐
          │          │ Send AT+CEREG?   │
          │          └───────┬──────────┘
        N │                  │
          │          ◇───────┴──────◇       ● If the module returns +CEREG: 0,1 or +CEREG: 0,5, it registers with PS domain.
          └────────  │  +CEREG: 0,1  │- - -> ● If the module returns +CEREG:0,<0,2,3,4>, it does not register with the PS
                     │     or        │          domain. Send the command every second for 40 times.
                     │  +CEREG: 0,5  │       ● If the module cannot register network after querying for 40 times, restart the
                     ◇───────┬──────◇           module.
                             │
                     ┌───────┴──────────┐
                     │  Register with the │
                     │ network successfully│
                     └────────────────────┘
```
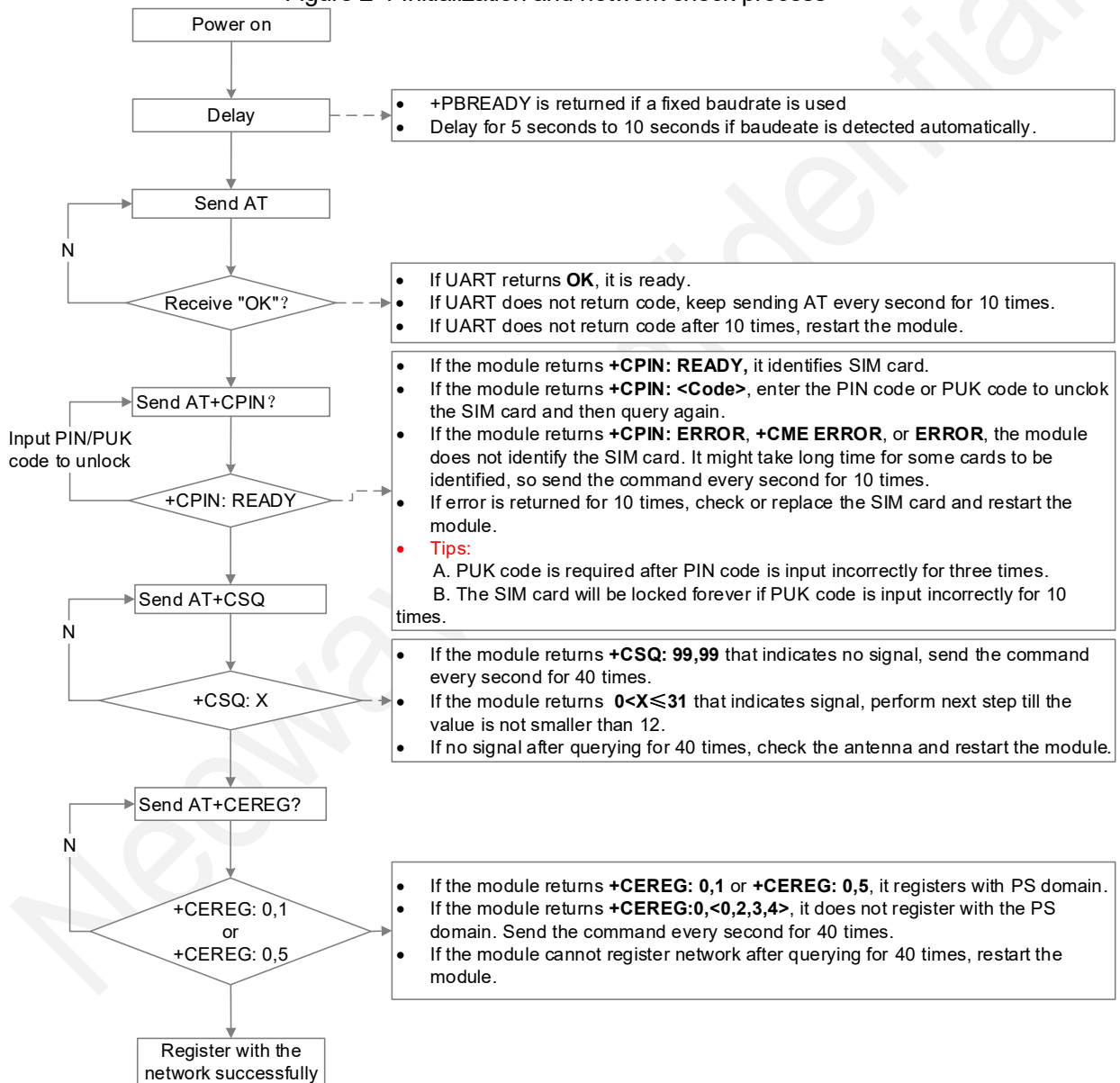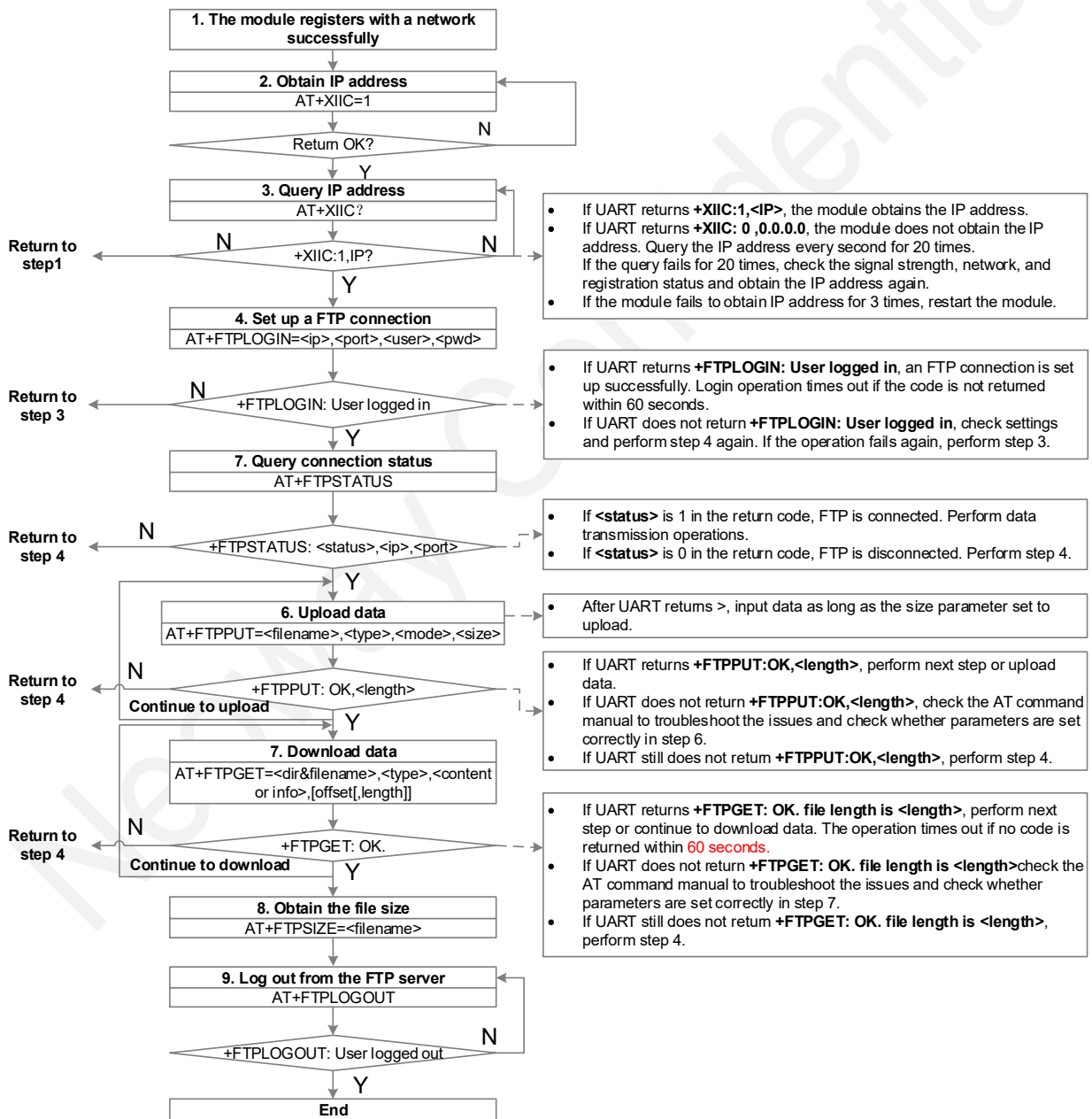
The header shows NB-IoT Modules FTP and HTTP(S) Application Notes

3 FTP Service

3.1 Service Process

Figure 3-1 Flowchart of FTP service

The sequence of steps 5, 6, and 7 is flexible.

3.2 Example

1. Check whether the module registers with a network.

```
AT                    //Check whether the module responds to AT commands.
OK


AT+CPIN?
+CPIN: READY          //Check whether the module identifies the SIM card.
OK


AT+CSQ
+CSQ: 31, 99          //Check the signal quality.
OK


AT+CEREG?
+CEREG: 0,1                      //Check whether the module registers with PS domain.
OK
```

2. Operate FTP data services.

```
At+XIIC=1             //Obtain the IP address.
OK


AT+XIIC?              //Query the IP address.
+XIIC:    1,10.173.119.219


AT+FTPLOGIN=58.60.184.213,21,neoway,neoway    //Set up an FTP connection. Note the IP address,
port, user name and password you use.
OK
+FTPLOGIN: User logged in


AT+FTPSTATUS     //Query the status of the FTP connection.
+FTPSTATUS: 1,58.60.184.213,21


AT+FTPPUT=text.txt,1,1,3   //Upload data. Input data after the module returns >.
>
Send →◇abc
+FTPPUT: OK,3


AT+FTPGET=text.txt,1,3    //Download data.
+FTPGET: OK.file length is 3


AT+FTPSIZE=text.txt  //Read the size of the FTP file.
+FTPSIZE: 3


AT+FTPLOGOUT   //Log out from the FTP server.
+FTPLOGOUT: User logged out
OK
```
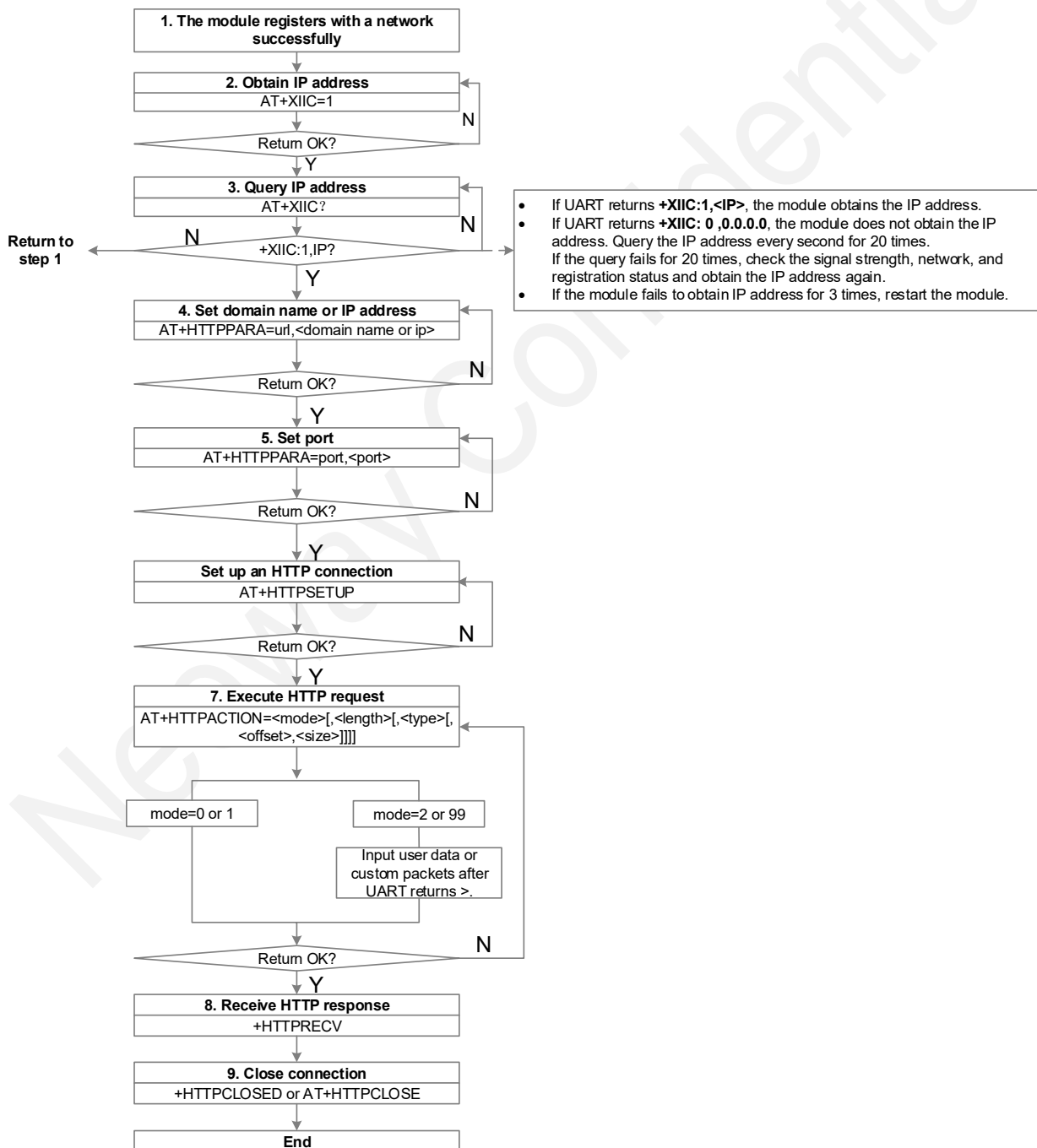
4 HTTP(S) Service

4.1 Service Process

Figure 4-1 Flowchart of HTTP(S) service

The processes of HTTPS service and HTTP service are the same.

4.2 Example

1. Check whether the module registers with a network.

```
AT                  //Check whether the module responds to AT commands.
OK


AT+CPIN?
+CPIN: READY        //Check whether the module identifies the SIM card.
OK


AT+CSQ
+CSQ: 31, 99        //Check the signal quality.
OK


AT+CEREG?
+CEREG: 0,1                     //Check whether the module registers with a network
successfully.
OK
```

2. Obtain an IP address, set up a transparent TCP connection, and transmit and receive data.

```
At+XIIC=1                       //Obtain the IP address.
OK


AT+XIIC?          //Query the IP address.
+XIIC:   1,100,127.6,141
OK


AT+HTTPPARA=url,58.60.184.213/post/formdata   //Set the URL. Note the URL you use.
OK


AT+HTTPPARA=port,7020  //Set port.
OK


AT+HTTPSETUP   //Set up an HTTP connection.
OK


AT+HTTPACTION=2,20,0    //Execute HTTP POST request. 20 indicates the length of
id=www.neoway.com.cn.
>
Send →◇id=www.neoway.com.cn
OK


+HTTPRECV:       //Receive HTTP response.
HTTP/1.1 200 OK
Server: nginx/1.11.3
Date: Mon, 30 Jul 2018 07:21:47 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: close


b
```

```
{"id":null}
0
```

```
+HTTPCLOSED:  HTTP Link Closed   //Close the HTTP link.
```