Skywire® LTE CAT-M1
Socket Dial Application Note
NimbeLink Corp
Updated: January 2020

© NimbeLink Corp. 2020. All rights reserved.

NimbeLink Corp. provides this documentation in support of its products for the internal use of its current and prospective customers. The publication of this document does not create any other right or license in any party to use any content contained in or referred to in this document and any modification or redistribution of this document is not permitted.

While efforts are made to ensure accuracy, typographical and other errors may exist in this document. NimbeLink reserves the right to modify or discontinue its products and to modify this and any other product documentation at any time.

All NimbeLink products are sold subject to its published Terms and Conditions, subject to any separate terms agreed with its customers. No warranty of any type is extended by publication of this documentation, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose and non-infringement.

Skywire and NimbeLink are registered trademarks of NimbeLink Corp. All other trademarks appearing in the document are the property of their respective owners.
# Table of Contents

## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td><strong>Introduction</strong></td>
<td>4</td>
</tr>
<tr>
<td>Orderable Part Numbers</td>
<td>4</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>4</td>
</tr>
<tr>
<td><strong>Default Socket Dial Procedure</strong></td>
<td>5</td>
</tr>
<tr>
<td>Overview</td>
<td>5</td>
</tr>
<tr>
<td>Firmware Check</td>
<td>5</td>
</tr>
<tr>
<td>Socket Configuration</td>
<td>5</td>
</tr>
<tr>
<td>Initiate Socket Dial</td>
<td>5</td>
</tr>
<tr>
<td>Send Data via HTTP</td>
<td>6</td>
</tr>
<tr>
<td>Shutdown Socket Connection</td>
<td>6</td>
</tr>
<tr>
<td><strong>Socket Dial Using Command Mode</strong></td>
<td>7</td>
</tr>
<tr>
<td>Overview</td>
<td>7</td>
</tr>
<tr>
<td>Firmware Check</td>
<td>7</td>
</tr>
<tr>
<td>Socket Configuration</td>
<td>7</td>
</tr>
<tr>
<td>Initiate Socket Dial</td>
<td>7</td>
</tr>
<tr>
<td>Send Data via HTTP</td>
<td>7</td>
</tr>
<tr>
<td>Shutdown Socket Connection</td>
<td>8</td>
</tr>
<tr>
<td><strong>Working Example 1: Dweet.io</strong></td>
<td>9</td>
</tr>
<tr>
<td>Preliminary Setup Procedure</td>
<td>9</td>
</tr>
<tr>
<td>Get IMEI of Modem</td>
<td>9</td>
</tr>
<tr>
<td>Socket Setup</td>
<td>10</td>
</tr>
<tr>
<td>Initiate Socket Dial</td>
<td>10</td>
</tr>
<tr>
<td>Send Data via HTTP</td>
<td>10</td>
</tr>
<tr>
<td>Receive Data via HTTP</td>
<td>11</td>
</tr>
<tr>
<td><strong>Working Example 2: Dweet.io Using Command Mode</strong></td>
<td>12</td>
</tr>
<tr>
<td>Preliminary Setup Procedure</td>
<td>12</td>
</tr>
<tr>
<td>Get IMEI of Modem</td>
<td>12</td>
</tr>
<tr>
<td>Socket Setup</td>
<td>13</td>
</tr>
<tr>
<td>Initiate Socket Dial</td>
<td>13</td>
</tr>
<tr>
<td>Send Data via HTTP</td>
<td>13</td>
</tr>
<tr>
<td>Receive Data via HTTP</td>
<td>14</td>
</tr>
<tr>
<td><strong>Troubleshooting</strong></td>
<td>16</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Orderable Part Numbers

<table>
<thead>
<tr>
<th>Orderable Device</th>
<th>Description</th>
<th>Firmware</th>
<th>Carrier</th>
<th>Network Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL-M1DK</td>
<td>Skywire M1 Development Kit</td>
<td>n/a</td>
<td>Any</td>
<td>Any</td>
</tr>
<tr>
<td>NL-SW-LTE-SVZM20</td>
<td>LTE CAT M1</td>
<td>32110</td>
<td>Verizon</td>
<td>LTE</td>
</tr>
<tr>
<td>NL-SW-LTE-SVZM20-B</td>
<td>LTE CAT M1</td>
<td>37120</td>
<td>Verizon</td>
<td>LTE</td>
</tr>
</tbody>
</table>

1.2 Prerequisites

This document assumes that the initial setup of the requisite modem and development kit has been completed using the Skywire® M1 Development Kit User Manual:

http://nimbelink.com/skywire-m1dk/user-manual/

If these steps are incomplete, please refer to the link above and complete the modem setup before proceeding.
2. Default Socket Dial Procedure

2.1 Overview

Socket dials are a useful process for uploading or downloading information from a website or database using HTTP commands. This document will cover two ways to complete a socket dial, one using the default online data mode and the other using command mode. This section will cover online data mode, and Section 3 will cover command mode. Finally, three examples of a complete socket dial procedure will be covered in Section 4, Section 5 and Section 6.

2.2 Firmware Check

Socket dial functionality is available on firmware 29318 and newer. To check firmware version, type this command, followed by the enter key:

ATI1

The modem should respond with something similar to:

UE5.0.0.0-xxxxx
LR5.0.0.0-xxxxx
OK

Where xxxx is the firmware version number. If the current firmware version is older than 29318, please consult the firmware update application note:


2.3 Socket Configuration

First, the socket connection must be configured. Issue this command into the terminal, followed by the enter key:

AT+SQNSCFG=3,3,300,90,600,50

The terminal should respond with:

OK

2.4 Initiate Socket Dial

In the terminal program, type the following command:

AT+SQNSD=3,0,80,"example.com"

In this case, 3 is the socket being used, 0 specifies TCP transmission protocol, 80 is the TCP port being used (TCP port 80 is used for HTTP), and example.com is the hostname.

The modem should respond with:
2.5 Send Data via HTTP

In order to send data to the web server, use the HTTP POST command. The syntax of the POST command is as follows:

```
POST /test/demo_form.asp HTTP/1.1
```

POST is the HTTP command being issued, /test/demo_form.asp is the endpoint of the server, and HTTP/1.1 is the HTTP version being used.

In the terminal program, enter the desired POST data, making sure that it is properly formatted according to the syntax listed above. Additionally, please note that the text being entered into the terminal will not be echoed, making manual data entry difficult. Accordingly, it is recommended that the command be pasted into the terminal, as opposed to typed.

Once the appropriate command has been entered, press the sequence `CTRL+M, CTRL+J, CTRL+M, CTRL+J`. This sequence will enter two new line characters, which are necessary to signal to the server that data transmission is complete. After a short delay, the terminal program should display:

```
HTTP/1.1 2xx OK
[text response from server]
OK
```

where 2xx is the successful response code of the HTTP server. 200 is the general HTTP/1.1 response of OK, meaning that the response was received properly. There are derivations of the 2xx response, however. If an unexpected value is received, please consult the HTTP/1.1 Protocol Documentation at the following URL:

http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

2.6 Shutdown Socket Connection

In the terminal program, type the following command, followed by the enter key:

```
+++ 
```

The output from the terminal should be:

```
OK
```

The +++ input is the escape command used to suspend the data connection in the socket. To shutdown the socket connection entirely, type the following command:

```
AT+SQNSH=3
```

followed by the enter key, and the terminal should respond with:

```
OK
```
3. Socket Dial Using Command Mode

3.1 Overview

The following section will lay out an alternate procedure for completing a socket dial using command mode.

3.2 Firmware Check

Please refer to Section 2.2 for information regarding firmware version verification.

3.3 Socket Configuration

To configure the socket, type the following command into the terminal, followed by the enter key:

AT+SQNSCFGEXT=3,1,0,0,0,0

Where: 3 is the socket being configured and 1 enables 'data amount mode'. Data Amount Mode instructs the modem to provide the number of bytes received for any HTTP response transmission. The other four arguments aren't important for this document, but a description of their usage can be found in the AT commands manual.

The terminal should respond with:
OK

3.4 Initiate Socket Dial

Type the following command into the terminal to open the socket, followed by the enter key:

AT+SQNSD=3,0,80,"example.com",0,0,1

Where: 3 is the socket being used, 0 is the transmission protocol (0 - TCP, 1 - UDP), 80 is the port number being connected to, example.com is the hostname or IP address being connected to, and 1 is the connection mode (0 - default, 1 - command mode)

After a short delay, the terminal program should respond with:
OK

3.5 Send Data via HTTP

To send data through the socket, type the following command, followed by the enter key:

AT+SQNSSEND=3

The terminal should respond with:
>

© NimbeLink Corp. 2020. All rights reserved.
Now, type or paste in the correct POST command, but do not hit the enter key. For information on the POST command or syntax, please refer to Section 2.5.

Once the appropriate command has been entered, press the sequence CTRL+M, CTRL+J, CTRL+M, CTRL+J to enter two new line characters at the end of the POST command. Finally, press CTRL+Z to indicate that data transmission is complete, as per the AT commands manual.

After a short while, the terminal should respond with something similar to:

OK
+SQNSRING: 3,368

The +SQNSRING output is the socket activity notification, and signifies that there is incoming data to be read. The value of 3 corresponds to the socket being used, and the value of 368 corresponds to the number of bytes received from the server.

To read the incoming data from the server, issue this command:

AT+SQNRECV=3,1500

Where 3 corresponds to the socket being used, and 1500 corresponds to the number of bytes to read. Please note that the second argument can be no larger than 1500 bytes.

The terminal will now output the data received from the server. It should look something like this:

HTTP/1.1 2xx OK
[text response from server]
OK

3.6 Shutdown Socket Connection

To suspend the data connection to the socket, enter the following characters after issuing the AT+SQNSSEND=3 command.

+++

followed by CTRL+M, CTRL+J, CTRL+M, CTRL+J, CTRL+Z. The terminal should respond with:

OK

To close the socket, type the following command, followed by the enter key.

AT+SQNSH=3

The terminal will respond with:

OK
4. Working Example 1: Dweet.io

This section will provide a working example of sending and receiving data via an HTTP transfer using a Skywire NL-SW-LTE-SVZM20 modem using a 4G Verizon LTE CAT M1 SIM.

dweet.io is a lightweight messaging service specifically designed for IoT (Internet of Things) devices. In addition to being lightweight, dweet.io does not require an account to get up and running. At [www.dweet.io](http://www.dweet.io), they have an excellent “hello world” example that this example is based on.

4.1 Preliminary Setup Procedure

Before beginning the socket dial procedure, it is crucial to verify that cellular functionality is enabled. Type the following command into the terminal:

AT+CFUN=1

This command will enable cellular functionality, which is initially off for NL-SW-LTE-SVZM20 CAT M1 modems. Next, check for cellular connectivity using the following command:

AT+CEREG?

The terminal should respond with:

+CEREG 0,1

If the terminal responds with:

+CEREG 0,2

wait a few seconds, and then try the AT+CEREG? command again. A value of 0,2 indicates that the modem is in the process of connecting to the network.

Once network connectivity has been established, proceed to the next section.

4.2 Get IMEI of Modem

dweet.io requires a unique device name in order to send and receive data. It is recommended to use a device’s IMEI as this unique indicator. To display the IMEI, type the following command into the terminal, followed by the enter key:

AT+CGSN

The terminal will respond with something similar to:

353238060023699

OK

This number should be identical to the IMEI printed on the modem label.
4.3 Socket Setup

Type the following command into the terminal, followed by the enter key:

AT+SQNSCFG=3,3,300,90,600,50

The terminal should respond with:

OK

4.4 Initiate Socket Dial

Using the syntax described in Section 2.4, type the following command into the terminal program, followed by the enter key:

AT+SQNSD=3,0,80,"dweet.io"

In this case, 3 is the socket being used, 0 specifies TCP transmission protocol, 80 is the TCP port being used (TCP port 80 is used for HTTP), and dweet.io is the hostname.

The expected output is:

CONNECT

4.5 Send Data via HTTP

Using the syntax described in Section 2.5, enter the following command into the terminal program:

POST /dweet/for/353238060023699?hello=world HTTP/1.1

Note: 353238060023699 is the IMEI of the modem used to create this example. This IMEI should be replaced with the IMEI of the modem being used.

Press the sequence CTRL+M, CTRL+J, CTRL+M, CTRL+J, and after a short delay the terminal program should respond with something similar to:

HTTP/1.1 200 OK

Access-Control-Allow-Origin: *

Content-Type: application/json

Content-Length: 203

Date: Tue, 15 Mar 2016 21:50:59 GMT

Connection: keep-alive

{"this":"succeeded","by":"dweeting","the":"dweet","with":{"thing":"353238060023699","created":"2016-03-15T21:50:59.822Z","content":{"hello":"world"},"transaction":"debcf53b-1494-4a37-852e-c4d5278f79e6"}}

If the preceding text is generated, then the POST command has succeeded. To see the actual results on dweet.io, follow the link below:
Be sure to replace the IMEI at the end of the link with the IMEI of the modem being used to follow this example. For this particular example, the following text was taken from the dweet.io webpage:

```
```

### 4.6 Receive Data via HTTP

Repeat Section 4.4, and enter the following text into the terminal after receiving the expected CONNECT output:

```
GET /get/latest/dweet/for/353238060023699 HTTP/1.1
```

Note: 353238060023699 is the IMEI of the modem used to create this example. This IMEI should be replaced with the IMEI of the modem being used.

Press the sequence CTRL+M, CTRL+J, CTRL+M, CTRL+J, and after a short delay the terminal program should respond with something similar to:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
Content-Length: 152
Date: Tue, 15 Mar 2016 21:58:13 GMT
Connection: keep-alive
```

```
{"this":"succeeded","by":"getting","the":"dweet","with":[{"thing":"353238060023699","created":"2016-03-15T21:51:25.173Z","content":{"hello":"world"}}]}
```
5. Working Example 2: Dweet.io Using Command Mode

This section will provide a working example of sending and receiving data via an HTTP transfer using a Skywire NL-SW-LTE-SVZM20 modem and a 4G Verizon LTE CAT M1 SIM.

This example is nearly identical to Section 4, except that it uses command mode instead of the default online data mode.

5.1 Preliminary Setup Procedure

Before beginning the socket dial procedure, it is crucial to verify that cellular functionality is enabled. Type the following command into the terminal:

AT+CFUN=1

This command will enable cellular functionality, which is initially off for NL-SW-LTE-SVZM20 CAT M1 modems. Next, check for cellular connectivity using the following command:

AT+CEREG?

The terminal should respond with:

+CEREG 0,1

If the terminal responds with:

+CEREG 0,2

wait a few seconds, and then try the AT+CEREG? command again. A value of 0,2 indicates that the modem is in the process of connecting to the network.

Once network connectivity has been established, proceed to the next section.

5.2 Get IMEI of Modem

dweet.io requires a unique device name in order to send and receive data. It is recommended to use a device’s IMEI as this unique indicator. To display the IMEI, type the following command into the terminal, followed by the enter key:

AT+CGSN

The terminal will respond with something similar to:

357353080053541

OK

This number should be identical to the IMEI printed on the top of the modem label.
5.3 Socket Setup

Next, configure the socket connection by issuing the following command, followed by the enter key:

AT+SQNSCFGEXT=3,1,0,0,0

Where 3 is the socket being used, and 1 enables ‘data amount mode’. Please refer to Section 3.3 for further information.

The terminal should respond with:

OK

5.4 Initiate Socket Dial

Using the syntax described in Section 3.4, type the following command into the terminal program, followed by the enter key:

AT+SQNSD=3,0,80,"dweet.io",0,0,1

Where: 3 is the socket being used, 0 is the transmission protocol (0 - TCP, 1 - UDP), 80 is the port number being connected to, example.com is the hostname or IP address being connected to, and 1 is the connection mode (0 - default, 1 - command mode)

The terminal should respond with:

OK

5.5 Send Data via HTTP

To send data through the socket, type the following command, followed by the enter key:

AT+SQNSSEND=3

The terminal should respond with:

>

Next, type or paste the following text into the terminal prompt, but do not press enter:

POST /dweet/for/357353080053541?hello=world HTTP/1.1

Note: 357353080053541 is the IMEI of the modem used to create this example. This IMEI should be replaced with the IMEI of the modem being used.

Press the sequence Ctrl+M, Ctrl+J, Ctrl+M, Ctrl+J to add two new line characters. Finally, press Ctrl+Z to signal the end of data transmission. Please refer to Section 3.5 for information regarding POST commands and related escape sequences.
The terminal should respond with something similar to this:

OK

+SQNSRING: 3,368

Where +SQNSRING: is the socket activity notification, 3 is the socket being used, and 368 is the number of bytes received in the HTTP response. If desired, read the information sent by the server through the socket by issuing the following command:

AT+SQNSRECV=3,1500

This command tells the modem to read the first 1500 bytes of incoming data on socket number 3. The output to the terminal will be something similar to this:

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
Content-Length: 203
Date: Mon, 07 May 2018 21:44:47 GMT
Connection: keep-alive

{"this":"succeeded","by":"dweeting","the":"dweet","with":{"thing":"357353080053541","created":"2018-05-07T21:44:47.635Z","content":{"hello":"world"},"transaction":"edf62c02-181d-4724-aebf-d2bd30b3b9a5"}}

If the preceding text is generated, then the POST command has succeeded. To see the actual results on dweet.io, follow the link below:

https://dweet.io/get/latest/dweet/for/357353080053541

Be sure to replace the IMEI at the end of the link with the IMEI of the modem being used to follow this example. For this particular example, the following text was taken from the dweet.io webpage:

{"this":"succeeded","by":"getting","the":"dweets","with":[]}

5.6 Receive Data via HTTP

To send data through the socket, type the following command, followed by the enter key:

AT+SQNSSEND=3

The terminal should respond with:

>

Next, type or paste the following text into the terminal prompt, but do not press enter:

GET /get/latest/dweet/for/357353080053541 HTTP/1.1
Note: 357353080053541 is the IMEI of the modem used to create this example. This IMEI should be replaced with the IMEI of the modem being used.

Press the sequence **CTRL+M, CTRL+J, CTRL+M, CTRL+J, CTRL+Z**. The terminal should respond with something similar to this:

```
OK
+SQNSRING: 3,317
```

To read the data sent by the server through the socket, issue the following command:

```
AT+SQNSRECV=3,317
```

Make sure to specify enough bytes to fully read the received text. In this case, the second argument to the following command must be greater than or equal to 317 in order to completely read all of the received data. In this example, the terminal responded with:

```
+SQNSRECV: 3,317
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
Content-Length: 152
Date: Mon, 07 May 2018 22:43:48 GMT
Connection: keep-alive

{"this":"succeeded","by":"getting","the":"dweets","with":[{"thing":"357353080053541","created":"2018-05-07T22:32:38.306Z","content":{"hello":"world"}}]}
```
6. Troubleshooting

6.1 Serial Clients

If the POST sequences are being inputted manually into a serial program such as PuTTY or TeraTerm, some serial clients may interpret the "Carriage Return" (CTRL+M) and "Line Feed" (CTRL+J) characters differently. For instance, if issues are encountered when sending information using an HTTP POST, try sending four "Line Feed" characters instead of the sequence of "Carriage Return" and "Line Feed".

For instance, instead of sending:

CTRL+M, CTRL+J, CTRL+M, CTRL+J

Send:

CTRL+J, CTRL+J, CTRL+J, CTRL+J

6.2 ERROR When Issuing AT+SQNSD

If an ERROR is received when trying to open the socket with AT+SQNSD, verify that you are on firmware version 37120 or newer. If you are not, please see the following guide to update your Skywire:


Otherwise, reset the Skywire:

AT^RESET

and try the socket dial command without the:

AT+SQNSCFG

command. The AT+SQNSCFG command is saved to non-volatile memory (NVM), so it does not need to be issued each time.