Skywire® LTE CAT 1
Socket Dial Example

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.

NimbeLink and Skywire are registered trademarks of NimbeLink Corp. All trademarks, service marks and similar designations referenced in this document are the property of their respective owners.
# Table of Contents

**1. Introduction**
- 1.1 Applies to the Following Part Numbers 4
- 1.2 Prerequisites 4

**2. Socket Dial**
- 2.1 Overview 5
- 2.2 Socket Dial Procedure - Rev 1
  - 2.2.1 Configure PDP Context 5
  - 2.2.2 Activate Context 5
  - 2.2.3 Initiate Socket Dial 6
  - 2.2.4 Send Data via HTTP 6
  - 2.2.5 Shutdown the Socket Connection 6
- 2.3 Socket Dial Procedure - Rev 2
  - 2.3.1 Configure PDP Context 7
  - 2.3.2 Configure Internet Service Setup Profile 7
  - 2.3.3 Activate Context 7
  - 2.3.4 Open the Socket 8
  - 2.3.5 Send Data via HTTP 8
  - 2.3.6 Read HTTP Response 9
  - 2.3.7 Shutdown the Socket Connection 9

**3. Working Example 1: dweet.io**
- 3.1 Overview 9
- 3.2 Rev 1 Firmware Procedure
  - 3.2.1 Get IMEI of Modem 10
  - 3.2.2 Initiate Socket Dial 10
  - 3.2.3 Send Data via HTTP 10
  - 3.2.4 Receive Data via HTTP 11
- 3.3 Rev 2 Firmware Procedure
  - 3.3.1 Get IMEI of Modem 11
  - 3.3.2 Configure Socket Service Profile 12
  - 3.3.3 Activate Context 12
  - 3.3.4 Open the Socket 12
  - 3.3.5 Send Data via HTTP 12
  - 3.3.6 Read HTTP Response 14
4. Troubleshooting

4.1 Serial Clients

4.2 AT^SISR Dropping Characters
1. Introduction

1.1 Applies to the Following Part Numbers

<table>
<thead>
<tr>
<th>Orderable Device</th>
<th>Description</th>
<th>Firmware Version</th>
<th>Manufacturer</th>
<th>Carrier</th>
<th>Network Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL-SWDK</td>
<td>Skywire Development Kit</td>
<td>n/a</td>
<td>NimbeLink</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NL-SW-LTE-GELS3</td>
<td>Skywire LTE CAT1 Modem</td>
<td>4.3.1.0</td>
<td>NimbeLink</td>
<td>Verizon</td>
<td>4G LTE</td>
</tr>
<tr>
<td>NL-SW-LTE-GELS3-B</td>
<td>Skywire LTE CAT1 Modem</td>
<td>4.3.2.0-25421</td>
<td>NimbeLink</td>
<td>Verizon</td>
<td>4G LTE</td>
</tr>
<tr>
<td>NL-SW-LTE-GELS3-C</td>
<td>Skywire LTE CAT1 Modem</td>
<td>4.3.3.0-29979</td>
<td>NimbeLink</td>
<td>Verizon</td>
<td>4G LTE</td>
</tr>
<tr>
<td>NL-SW-LTE-GELS3-D</td>
<td>Skywire LTE CAT1 Modem</td>
<td>4.3.3.0-36343</td>
<td>NimbeLink</td>
<td>Verizon</td>
<td>4G LTE</td>
</tr>
<tr>
<td>TG.30.8113</td>
<td>Cellular Antenna</td>
<td></td>
<td>Taoglas</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.2 Prerequisites

This document assumes you have completed the initial setup of your modem and development kit. If you have not completed those steps, refer to the Skywire® Development Kit User Manual and complete the modem setup before proceeding.
2. Socket Dial

2.1 Overview

Using socket dials is useful for uploading information to or downloading information from a website or database via HTTP commands. Below is an example for connecting to a server, uploading data, downloading data, and disconnecting from the server.

This section explains the steps required to initiate a socket dial using two different versions of firmware. To find what version of firmware your modem is running, send the following command:

`ATI`

The modem should respond with several lines of information, including a line that says:

`REVISION 4.3.x.0`

Where "x" is either 1, 2, or 3, indicating the firmware revision.

If your revision lists **4.3.1.0c**, this is referred to as "Rev 1" firmware throughout the document. Use the procedure outlined in Section 2.2.

If your revision lists **4.3.2.0** or **4.3.3.0**, this is referred to as "Rev 2" firmware throughout the document. Use the procedure outlined in Section 2.3.

2.2 Socket Dial Procedure - Rev 1

2.2.1 Configure PDP Context

Type the following command into your terminal program:

`AT+SQNSCFG=3,3,300,90,600,5`

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

`OK`

Next, type the following command into your terminal program:

`AT+CGDCONT=3,"IP","[APN]"`

where [APN] is your carrier’s APN, followed by the Enter key. Your terminal program should respond with:

`OK`

For example, if your APN is "vzwinternet", type:

`AT+CGDCONT=3,"IP","vzwinternet"

2.2.2 Activate Context

Type the following command into your terminal program:

`AT+CGACT=1,3`
Followed by the Enter key, and the terminal should respond with:

```
OK
```

This will get an IP address for your modem.

### 2.2.3 Initiate Socket Dial

In the terminal program, type the following command:

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

where 3 is the socket we are using, 0 is the transmission protocol to use (0 - TCP, 1 - UDP), 80 is the port number you’d like to connect to, and example is the host name or IP address you’d like to connect to, followed by the enter key. After a short delay, the terminal program should respond with:

```
CONNECT
```

Note: The TCP port (third parameter) that is typically used is port 80, which is the TCP Port for HTTP. Depending on your application, you may need a different port.

### 2.2.4 Send Data via HTTP

In order to send data to the webserver, you can 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 you will be using.

Note: For this step, you will not be able to see the text you are typing. To ensure that the data is accurate, it is recommended to copy and paste in the data. However, this is not necessary.

In the terminal program, enter the POST data you would like to send according to the above syntax. Make sure that the data is formatted to how your server is expecting it. Once you type the data, press `CTRL+J` twice. This will enter the two new line characters 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 you receive an unexpected value, please consult the HTTP/1.1 Protocol Documentation at the following URL:

[http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html)

### 2.2.5 Shutdown the Socket Connection

In the terminal program, type the following command:

```
+++`

PN 30102 rev 9 © NimbeLink Corp. 2020. All rights reserved.
followed by the Enter key, and the terminal should respond with:

   OK

This command (three plus signs) is the escape command to suspend the data connection in the socket connection.

To shut down the socket connection, type the following command:

   AT+SQNSH=3

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

   OK

2.3 **Socket Dial Procedure - Rev 2**

2.3.1 **Configure PDP Context**

Type the following command into your terminal program, followed by the Enter key:

   AT+CGDCONT=3,"IPV4V6","[APN]"

Where [APN] is your carrier's APN. The modem should respond with:

   OK

For example, if your APN is "VZWINTERNET", type:

   AT+CGDCONT=3,"IPV4V6","VZWINTERNET"

2.3.2 **Configure Internet Service Setup Profile**

Type the following commands into your terminal program, each followed by the Enter key:

   AT^SISS=0,"srvType","Socket"
   AT^SISS=0,"conId",3
   AT^SISS=0,"address","socktcp://www.example.com:[port]"

Where "www.example.com" is the website you want to connect to, and [port] is the port number the socket should use (for example, port 80 would be used for HTTP over TCP). The modem should respond to each command with the following:

   OK

This is the minimum profile configuration required to establish a connection over a socket.

2.3.3 **Activate Context**

Type the following command into your terminal program, followed by the Enter key:

   AT^SICA=1,3

The modem should respond with

   OK

At this point, the modem will begin attempting to activate the specified PDP context.
To verify that the context has been activated, type the following command followed by the Enter key:

   AT^SICA?

The modem should respond with:

   ^SICA: 1,1
   ^SICA: 2,0
   ^SICA: 3,1
   ^SICA: 4,0
   ^SICA: 8,0
   OK

If the line for context 3 shows "^SICA: 3,0", retry the context activation command at the beginning of this section.

2.3.4 Open the Socket

Type the following command into your terminal program, followed by the Enter key:

   AT^SISO=0

The modem should respond with:

   OK

   ^SISW: 0,1

The modem uses the parameters entered in Section 3.3 when opening the socket.

2.3.5 Send Data via HTTP

To initiate sending data, a POST command is sent using the following command:

   AT^SISW=0,#

Where # is the size of the message in bytes (i.e. the number of characters). It is recommended to type out the desired message in a text editor of your choice to determine the message length.

Note: The message length must include the carriage return and newline characters that are required at the end of the POST message (4 extra characters)

After entering the previous command, type the POST command using the following syntax:

   POST /test/demo_form.asp HTTP/1.1

Followed by CTRL+M, CTRL+J, CTRL+M, CTRL+J. This sends two sequences of newline and carriage return characters. In the above example, the POST message contains 33 characters, so 37 would be entered as the second parameter of the AT^SISW command (33 message characters + 4 whitespace characters).

If entered properly, the modem should respond with:
2.3.6 Read HTTP Response

If the modem responded with "^SISR: 0,1" in the previous step, that means there is incoming data on the socket that can be read. To read from the socket, type the following command followed by Enter:

```
AT^SISR=0,1000
```

Where 1000 is the maximum number of bytes to read. If there are fewer than 1000 bytes to read, all of them will be read. If more data is available, the command can be issued multiple times until all data is received.

The modem should respond with:

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

2.3.7 Shutdown the Socket Connection

Type the following command in your terminal program, followed by the Enter key:

```
AT^SISC=0
```

The modem should respond with:

```
OK
```

The socket should now be closed.

3. Working Example 1: dweet.io

3.1 Overview

This section will provide a working example of sending and receiving data via an HTTP transfer using a Skywire NL-SW-LTE-GELS3 modem using a 4G Verizon LTE 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 we will be using.

Use the procedure in Section 3.2 for modems with Rev 1 firmware and the procedure in Section 3.3 for modems with Rev 2 firmware.

3.2 Rev 1 Firmware Procedure

Use the following sequence of commands if you have the Rev 1 firmware on your modem.
3.2.1 Get IMEI of Modem

dweet.io requires a unique device name in order to send and receive data. For this, we will be using the IMEI of our modem, which is unique to our modem. To display the IMEI, type the following command into the terminal program:

```
AT+CGSN
```

followed by the Enter key, and the terminal will respond with something similar to:

```
353238060023699
```

OK

The number should be identical to the IMEI printed on the top of the modem label.

3.2.2 Initiate Socket Dial

Using the syntax described in Step 2.4, type the following command into the terminal program:

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

followed by the Enter key, and the terminal program should respond with

```
CONNECT
```

In this case, 3 is the socket we are using on the Skywire modem, 0 means we are using the TCP transmission protocol, 80 is the TCP port we are using (TCP port 80 is used for HTTP), and dweet.io is the hostname we are connecting to.

3.2.3 Send Data via HTTP

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

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

Note: 353238060023699 is the example IMEI that we used in Step 3.2.1. Replace this with your unique IMEI.

Press CTRL+J twice, 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
```
Notice that our IMEI appears next to "thing". By going to the website https://dweet.io/get/latest/dweet/for/353238060023699, and replacing 353238060023699 with your IMEI, you will see:

```json
```

This text is the same as the text at the bottom of the terminal response, indicating a successful HTTP transaction.

### 3.2.4 Receive Data via HTTP

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

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

Note: 353238060023699 is the example IMEI that we used in Step 3.2.1. Replace this with your unique IMEI.

Press CTRL+J twice, 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":"dweets","with":{"thing":"353238060023699", "created":"2016-03-15T21:51:25.173Z","content":{"hello":"world"}}}

Notice that our IMEI appears next to "thing". This indicates that we successfully received the HTTP information.

### 3.3 Rev 2 Firmware Procedure

Use the following sequence of commands if you have the Rev 2 firmware on your modem.

#### 3.3.1 Get IMEI of Modem

dweet.io requires a unique device name in order to send and receive data. For this, we will be using the IMEI of our modem, which is unique to our modem. To display the IMEI, type the following command into the terminal program:

```
AT+CGSN
```

followed by the Enter key, and the terminal will respond with something similar to:
OK

The number should be identical to the IMEI printed on the top of the modem label.

**3.3.2 Configure Socket Service Profile**

Using the commands outlined in Section 2.3.2, issue the following commands, each followed by the Enter key:

- \texttt{AT^SISS=0,"srvType","Socket"}
- \texttt{AT^SISS=0,"conId",3}
- \texttt{AT^SISS=0,"address","socktcp://dweet.io:80"}

The modem should respond to each command with:

- OK

These commands specify that we want a generic **Socket** that uses PDP context 3 to connect to dweet.io on port 80 over TCP.

**3.3.3 Activate Context**

Type the following command to activate PDP context 3:

- \texttt{AT^SICA=1,3}

Wait for the modem response as described in Section 2.3.3.

**3.3.4 Open the Socket**

Type the following command to open the socket:

- \texttt{AT^SISO=0}

This uses the socket profile settings we set in Section 3.3.2 to initiate the socket dial.

The modem should respond with:

- OK

- \texttt{^SISW: 0,1}

**3.3.5 Send Data via HTTP**

To allow data to be send over the socket, send the following command, followed by the Enter key:

- \texttt{AT^SISW=0,56}

The modem should respond with:

- \texttt{^SISW: 0,56,0}
Note: The value of 56 is specific to the message we are going to send in this example. If your message is a different length, this value must be updated accordingly.

Next, send the POST command in the following format:

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

Followed by CTRL+M, CTRL+J, CTRL+M, CTRL+J.

Note: 353238060023699 is the example IMEI that we used in Step 3.3.1. Replace this with your unique IMEI.

The modem should respond with:

    OK
    ^SISW: 0,1
    ^SISW: 0,1
    ^SISR: 0,1

The response from the POST message can be read by issuing the following command:

    AT^SISR=0,1000

The modem should respond with something similar to the following output:

    ^SISR: 0,368
    HTTP/1.1 200 OK
    Access-Control-Allow-Origin: *
    Content-Type: application/json
    Content-Length: 203
    Date: Fri, 03 Jun 2016 20:18:39 GMT
    Connection: keep-alive

    {"this":"succeeded","by":"dweeting","the":"dweet","with":{"thing":"353238060023699","created":"2016-06-03T20:18:39.914Z","content":{"hello":"world"},"transaction":"48c0b299-dc8a-484d-a587-c974043dd51f"}}

    OK

Notice that our IMEI appears next to "thing". By going to the website https://dweet.io/get/latest/dweet/for/353238060023699, and replacing 353238060023699 with your IMEI, you will see:


This text is the same as the text at the bottom of the terminal response, indicating a successful HTTP transaction.
3.3.6 Read HTTP Response

To read data via HTTP, first send the following command:

```
AT^SISW=0,54
```

Note: The value of 54 is specific to the message we are going to send in this example. If your message is a different length, this value must be updated accordingly.

Next, send a GET request using the following syntax:

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

Followed by CTRL+M, CTRL+J, CTRL+M, CTRL+J. The modem should respond with:

```
OK
^SISW: 0,1
^SISW: 0,1
^SISR: 0,1
```

To read the response to the GET request, issue the following command:

```
AT^SISR=0,1000
```

The modem should respond with:

```
^SISR: 0,317
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
Content-Length: 152
Date: Fri, 03 Jun 2016 20:31:21 GMT
Connection: keep-alive

{"this":"succeeded","by":"getting","the":"dweets","with":[{"thing":"353238060023699","created":"2016-06-03T20:18:39.914Z","content":{"hello":"world"}}]
```

```
OK
```

Notice that our IMEI appears next to "thing". This indicates that we successfully received the HTTP information.
4. Troubleshooting

4.1 Serial Clients

If you are manually inputting the above sequences 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 you are having issues sending information using a HTTP POST, try sending four "Line Feed" characters instead of the sequence of "Carriage Return" and "Line Feed".

So, in Section 2.3.5, instead of sending:

"CTRL-M" "CTRL-J" "CTRL-M" "CTRL-J"

send:

"CTRL-J" "CTRL-J" "CTRL-J" "CTRL-J"

4.2 AT^SISR Dropping Characters

If you are not getting entire HTTP responses when you issue the AT^SISR command, first query how many bytes you have available to read. For example, before you read from your socket, issue:

AT^SISR=[socket number],0

followed by the Enter key, and you'll receive the following response:

^SISR: [socket number],[number of available bytes]

OK

If [number of available bytes] is greater than 1500, you can read up to 1500 bytes at once. If that number is lower than 1500, you can read up to that number of bytes.

Here is an example using Socket 0:

AT^SISR=0,0

^SISR: 0,361

OK

In this case, there are 361 bytes available to read. You can read up to 361 bytes without error.