BitPipe™Cellular Dev-Kit

User Guide

Rev 1.2

2018-03-19

Table of Contents

Table of figures

1 Revision history

Revision	Date	Comments
0.9	2016-07-19	Pre-release
1.0	2016-09-02	First release
1.1	2017-08-23	Updated to Dev-Kit REV121 with BP-C-Tool V1.0.4
1.2	2018-03-19	Add HTML GET and POST examples Add MQTT connect, subscribe, publish and receive example. Add BitPipe mode selection Add modem USB mode Restructured the manual by mode instead of using only BC test Tool. Add figure for the antennas

2 Introduction

The BitPipe™ Development Kit provides an easy way to test the BitPipe™ cellular modules. This guide guides through configuring and using the different modes exposed by the Bitpipe cellular module. It also introduces the BitPipe™ Cellular Test Tool which is designed to help to test all the functionalities of the modem and perform cellular communication within minutes.

3 Included Material

The development kit contains the following items:

- 1x BitPipe™ Cellular modem
- 1x ADPT-BP-DEMO (demo board)
- 1x ADPT-BP-BOB (break out board)
- 1x Penta Band Antenna*
- 1x UFL to SMA antenna adapter*
- 1x AC-DC Wall Mount Adapter 9V-12W
- 1x USB to Serial communication cable 3.3V (TTL-232R-3V3-2MM)
- 1x Screw

* The contents of the development kit may vary depending on the BitPipe™ model.

Figure 1: BitPipe™ Development Kit

4 BitPipe™ Development Kit Description

In the Development Kit, there are two base boards available to connect the BitPipe™. Each one is described below.

The demo board is mainly used to do a quick demonstration of all the BitPipe™ features. Figure 2 identifies various elements on the board. Each section is listed below:

1. Thumbwheel (GPIO3)
2. BitPipe™ Connector
3. LEDs (GPIO0, GPIO1, GPIO2)
4. SETTINGS switch (SW1)
5. FTDI Connector
6. Micro-USB Connector
7. Power connector and indicator (LED D1)

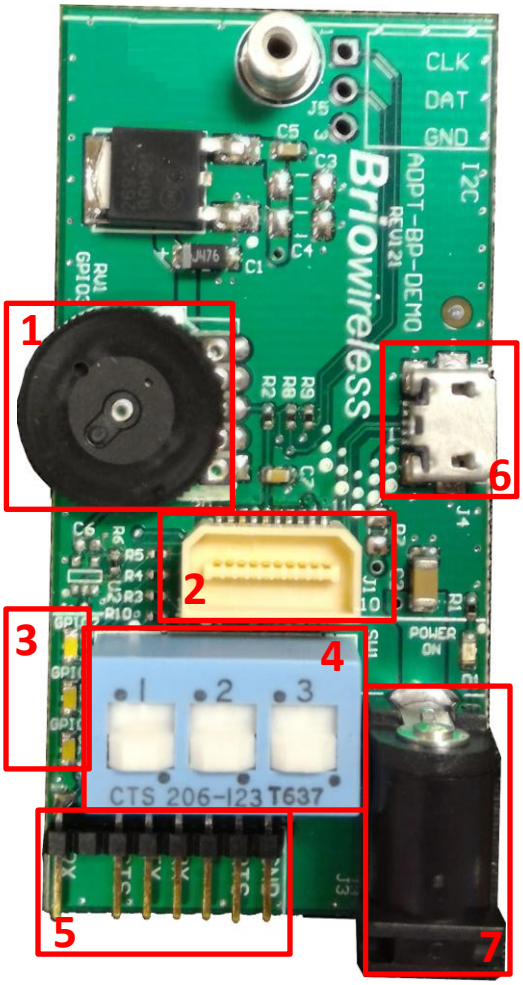

Figure 2: ADPT-BP-DEMO board

The break out board is used to provide an easy connectivity to the BitPipe™ Cellular modem connector when prototyping. Two 100 mils headers are available on the bottom side, spaced to fit most breadboards.

Figure 3 provides the pinout of these headers. Please refer to the BitPipe™ Cellular modem datasheet before connecting to any circuitry.

Figure 3: ADPT-BP-BOB board

5 Setup and installation

5.1 BitPipe™ Cellular Modem Installation

Insert the modem in the socket of the ADPT-BP-DEMO board (box 2 in Figure 2). Use the provided screw to secure the modem on the board (Figure 4). All switches (box 4 in Figure 2) should be at the bottom position to enable the default Modem mode. Please refer to the BitPipe™ Cellular modem datasheet for more information about its operating modes.

Figure 4: Modem secured with screw

5.2 SIM card insertion

Insert a Micro SIM (3FF) with its golden contacts facing towards the BitPipe™ module in the SIM card holder under the module (illustrated in Figure 5). The SIM card holder is a Push-in / Push-out type. A “click” should be heard and the SIM card should not be visible when it is fully inserted (Refer to Figure 6). Make sure the SIM card has been activated with a data plan.

Figure 5: SIM card orientation

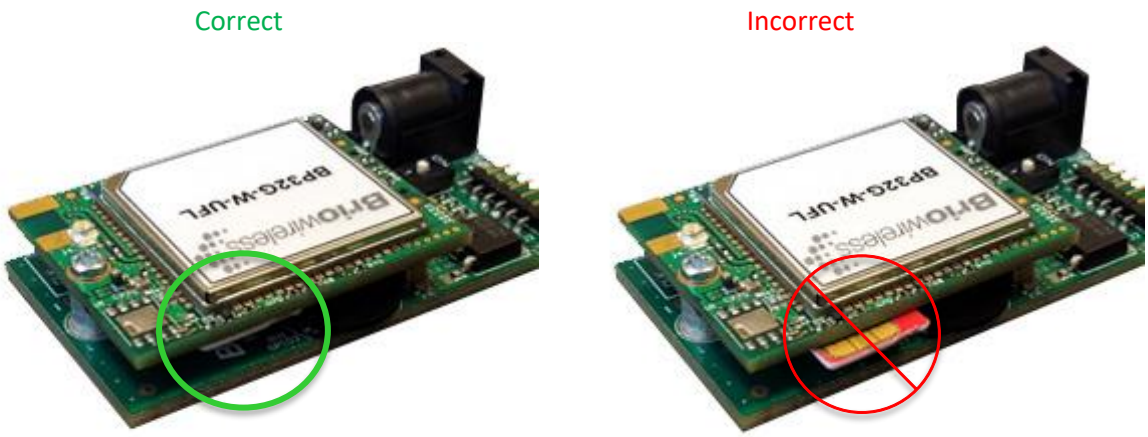

Figure 6: SIM card insertion

5.3 Antenna(e)

Install the provided antenna(e) in the modem SMA or UFL connector of the BitPipe™ module. **Note:** The number of antenna(e) will depend on the BitPipe™ model. On some models, 2 antennas are needed: The first antenna is responsible for transmission while the second antenna is used for reception diversity. See Figure 7.

Figure 7: Antenna installation

5.4 Bitpipe™ mode selection

BitPipe devices provide three different modes of operation: Autonomous, Modem UART and Modem USB. The three dipswitches on the Dev-Kit allow the configuration of the BipPipe™ mode. The table below shows the possible settings:

	SW1-1	SW1-2	SW1-3
Modem UART	ON: Power-ON Radio OFF: Power-Off Radio	**OFF**	**OFF**
Modem USB	ON: Power-ON Radio OFF: Power-Off Radio	**ON**	**OFF**
Autonomous	ON: GPIO_0 connected to LED on the Dev-Kit OFF: GPIO_0 pulled-up	ON: pull-up I2C OFF: pull-down I2C	**ON**

Please refer to the BitPipe™ datasheet for more information about the BitPipe™ operation modes.

6 Getting started with Modem in USB mode

The modem USB mode allows you to use the Bitpipe as the standard network interface to communicate over the internet. It is the recommended method if you plan to add broadband connection to your host through the USB interface. The following protocols are exposed depending on the BitPipe™ model: PPP, ECM and NCM. For each protocol, we provide you Linux scripts to easily establish connectivity to your board. Please refer to the Application Note AN0031 (Using USB Interface on BitPipe™) for information about establishing wireless connectivity through the USB.

To use the dev-kit in USB mode:

- Configure the dipswitches in modem UART mode (SW1-1 = ON, SW1-2 = ON, SW1-3 = OFF). See section 5.4 for details.
- Connect the dev-kit to your board using a USB Mini B cable,
- Connect the Wall adaptor in a power outlet. Insert the barrel plug in the demo board power connector (box 7 in Figure 2). The modem will boot automatically and the “Power ON” LED (D1 next to the power connector) will turn ON when the modem is ready to communicate to your board.
- On Linux, run the appropriate connect_script supported by your BitPipe™ model as indicated by the application note AN0031.

Please note that the BitPipe Serial API, and the associated radio abstraction layer is not used when communicating through the USB interface.

7 Getting started with Modem in UART mode

7.1 Quick Start

This section helps quickly establish cellular data transmission using the BitPipe™ Cellular in modem UART mode.

7.1.1 BitPipe™ Cellular Test Tool

The BP-C-TestTool is a software tool that allows the user to interact with the BitPipe™ modules. The tool facilitates the integration of the module and helps understand the BitPipe™ communication interfaces. Below are the main features of the tool:

- Retrieve the BitPipe™ module information
- Retrieve and configure the radio settings
- Send and receive SMS with different message encoding
- Establish IP connectivity with a remote server through HTTP, MQTT and TCP
- Test serial API commands available on the module
- Update the BitPipe™ firmware

You can find the latest version of the BitPipe™ Cellular Test Tool in following link under "Bitpipe Cellular DevKit": https://www.briowireless.com/technical-support/ . Download it and unzip it in a folder of your choice.

7.1.2 Install PC driver for the FTDI communication cable

Download and install the cable manufacturer (FTDI) driver on your PC. Drivers can be found on the FTDI website (link below):
http://ftdichip.com

Plug the FTDI communication cable (TTL-232R-3V3-2MM) in a PC USB port, and confirm that the cable is detected and the drivers are installed.
In the Device Manager (in Windows), you should see a **USB Serial Port** device listed under Ports (COM & LPT) as shown in Figure 8.

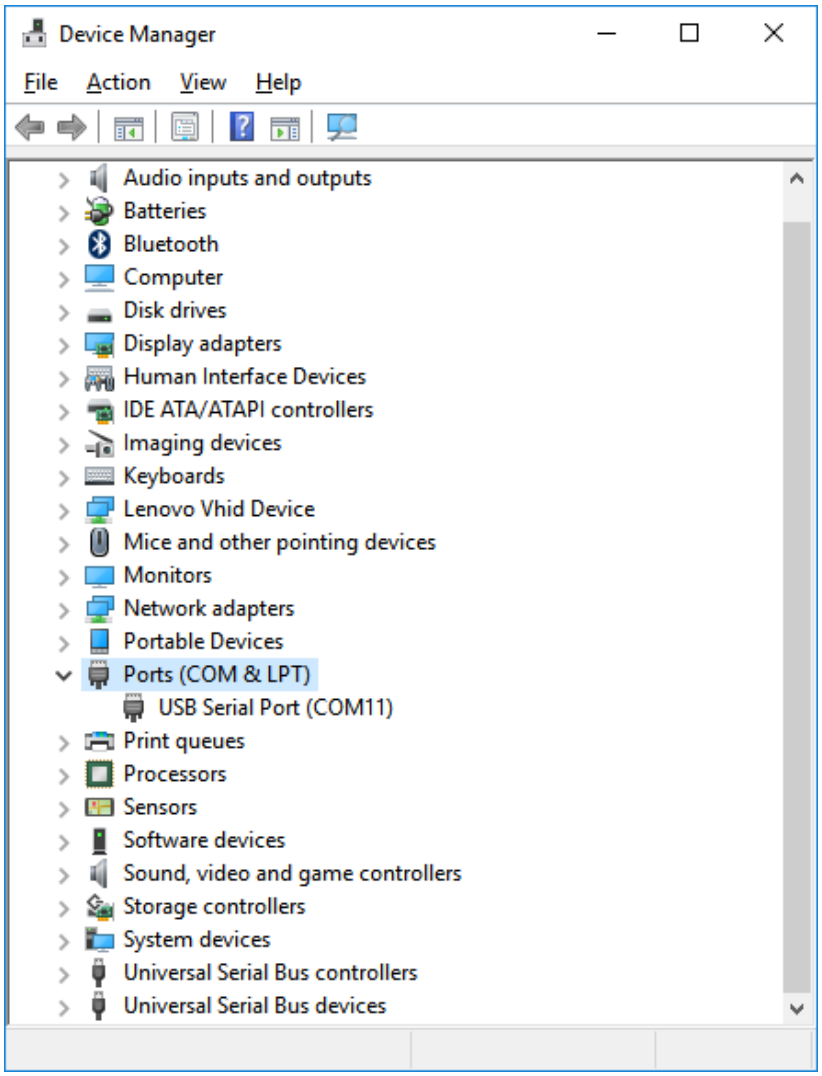

Figure 8 USB serial port listed for the FTDI communication cable

7.1.3 Dev-kit connections

To use the dev-kit in modem UART mode:

- Connect the communication cable (TTL-232R-3V3-2MM) in the FTDI Connector (box 5 in Figure 2).
- Configure the dipswitches in modem UART mode (SW1-1 = OFF, SW1-2 = OFF, SW1-3 = OFF). See section 5.4 for details.
- Connect the Wall adaptor in a power outlet. Insert the barrel plug in the demo board power connector (box 7 in Figure 2). The modem will boot automatically and the “Power ON” LED (D1 next to the power connector) will turn ON when the modem is ready to establish a communication with the PC.
-

7.1.4 Connecting using the BitPipe™ Cellular Test Tool

The following section provides the steps to connect to the modem and establish a mobile connection with the modem.

Make sure to have followed the setup and installation steps in section 5 before proceeding.

1. Launch the "BP-C-TestTool.exe" application (Refer to section 7.1.1 if it has not been installed yet). Figure 9 contains a screenshot of the application when it is first open.

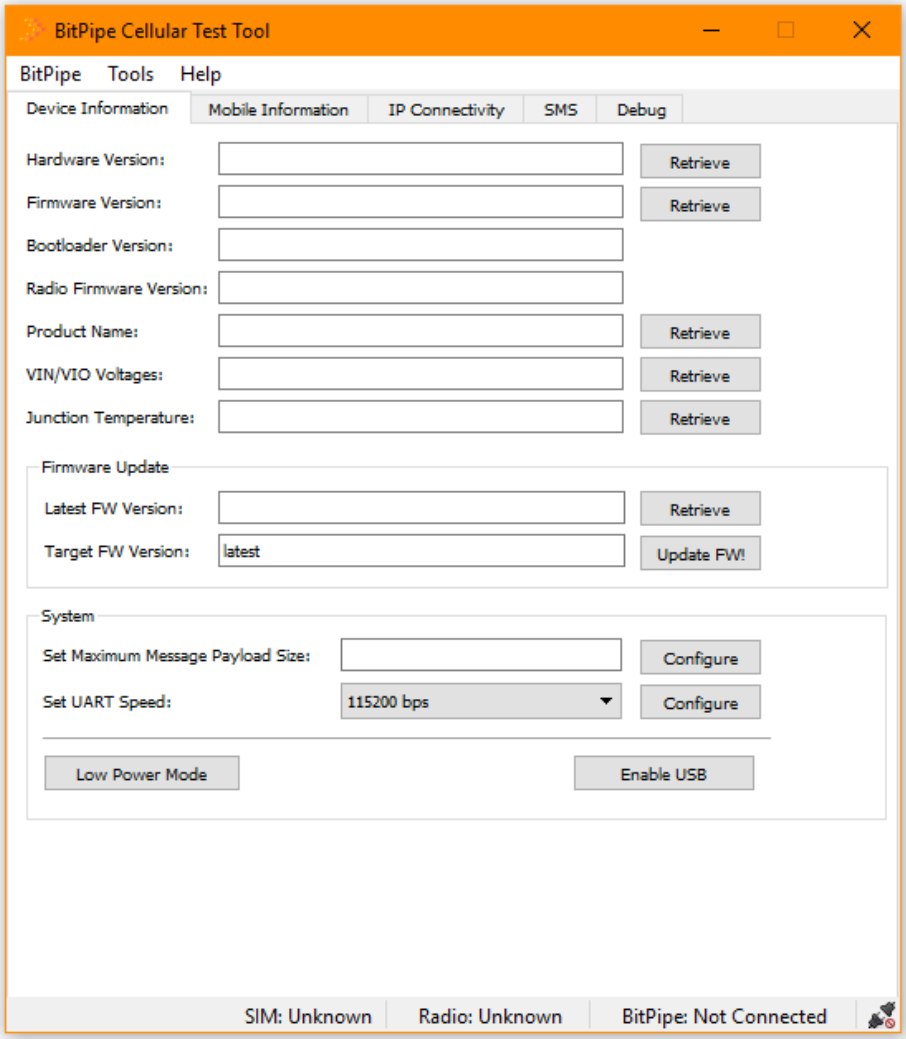

Figure 9: BitPipe™ Cellular Test Tool

2. Select the "BitPipe" menu and click on "Connect to BitPipe…" (See Figure 10).
3. In the window that popped up, select the correct serial port to communicate in the pop-up window. The default baudrate for the BitPipe™ is 115200 bps.

4. Click "OK" to start the communication. When connected, the status bar at the bottom will display the COM port and the baud rate used (see Figure 11).

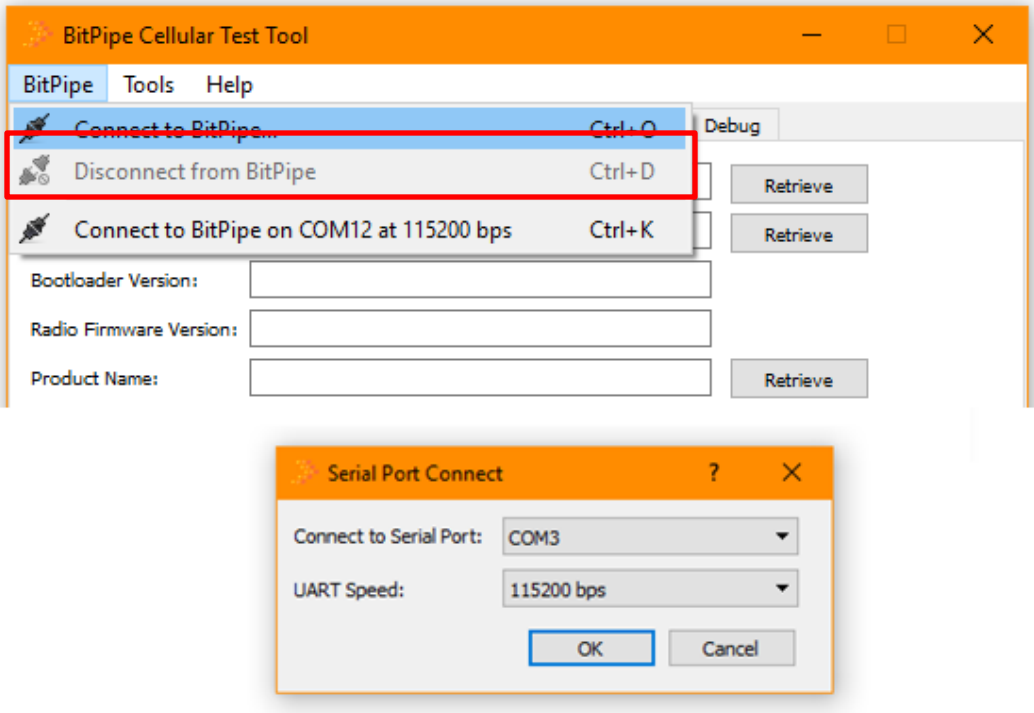

Figure 10: Communication port selection

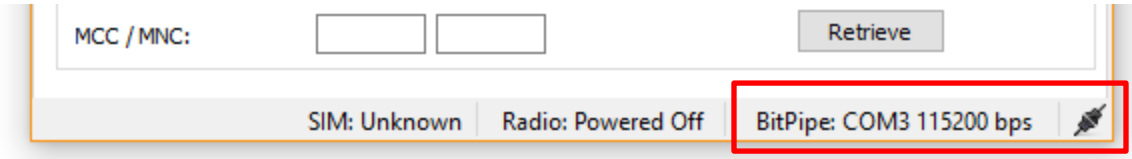

Figure 11: Communication successfully established

5. Select the "Mobile Information" tab to power up the modem (see Figure 12).
6. Make sure "DC-DC" radio button is selected.
7. Click on the "Power-On Radio" button.

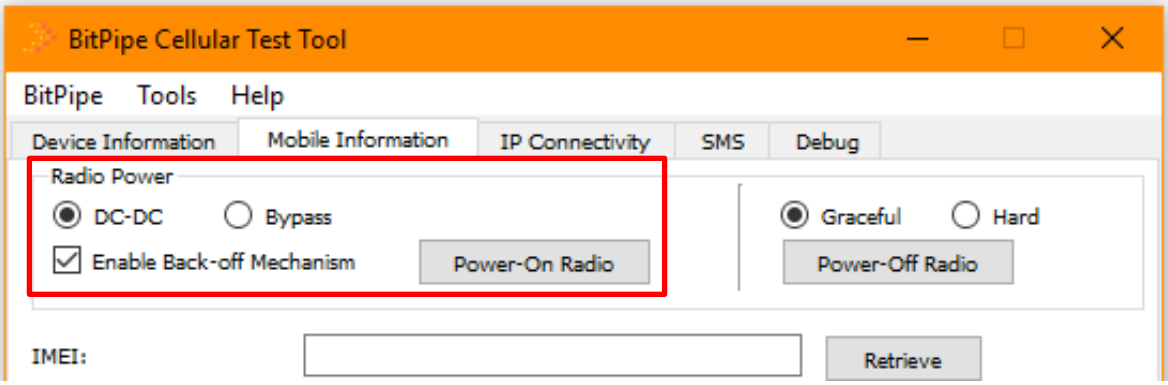

Figure 12: Power-On Radio

8. In the DC-DC warning window, click on "Yes" (see Figure 13). The dev-kit provides safe voltages for the BitPipe™.
   * Refer to the BitPipe™ Cellular modem datasheet for more information on the power modes.

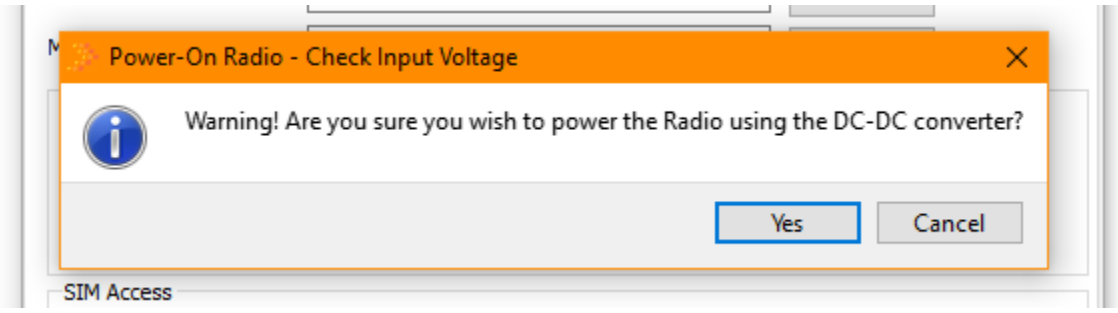

Figure 13: DC-DC Warning

9. The status bar will show when the radio is ready (see Figure 14). The radio will automatically connect to the mobile network upon power up and the "Mobile Connection Status" box will contain the mobile network information (see Figure 15).

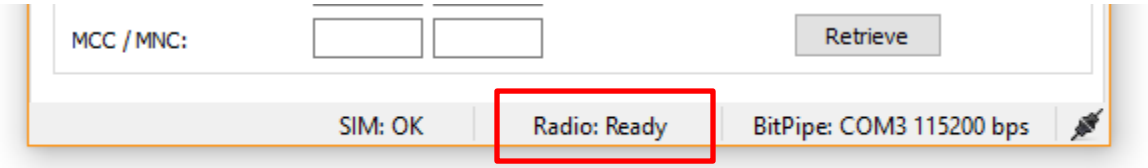

Figure 14: Radio status

10. To retrieve the radio information (IMEI, ICCID IMSI, phone number and mobile date time), press the "Retrieve" button next to corresponding text field (see Figure 15).

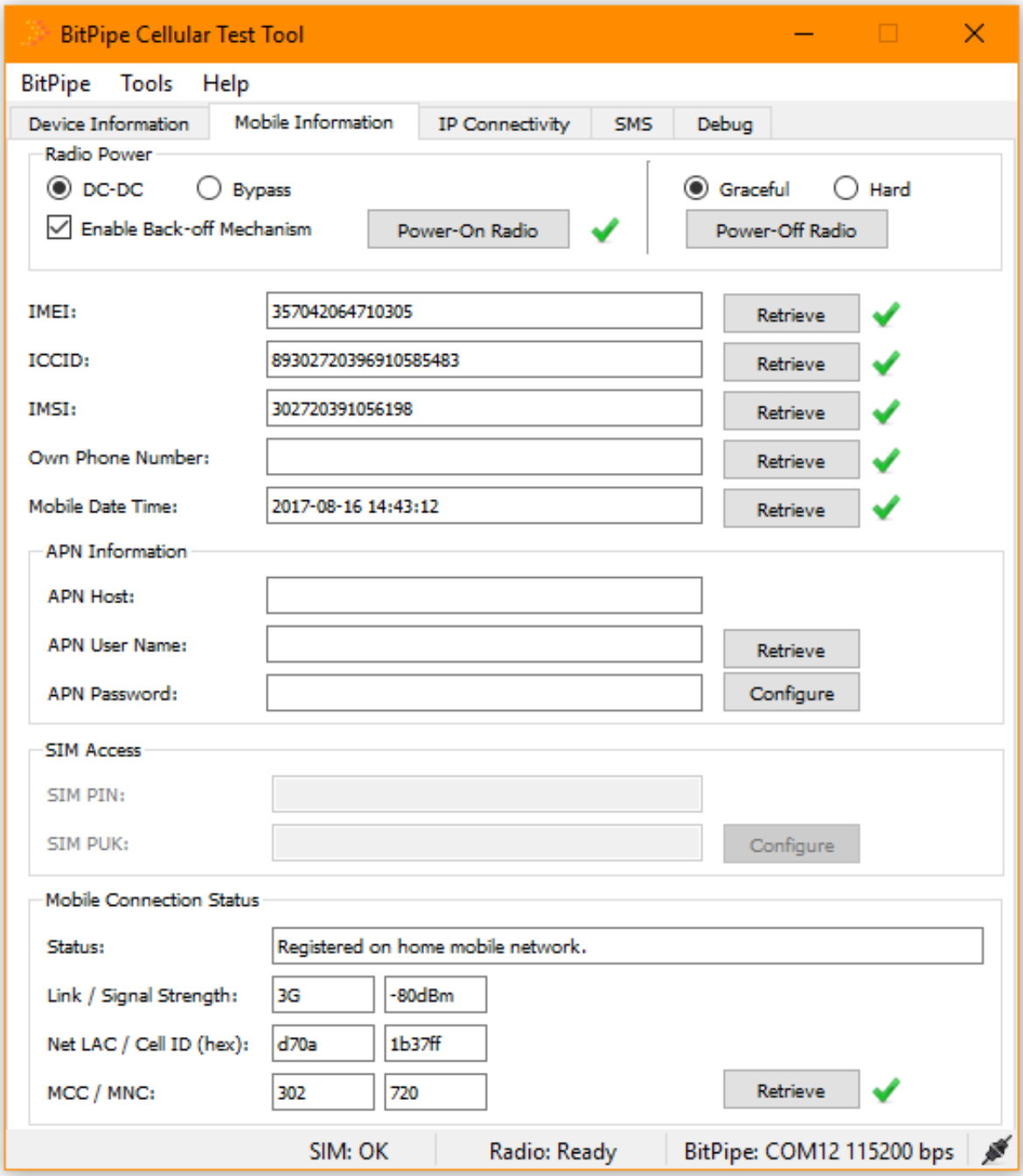

Figure 15: Mobile Information tab

11. To enable the data service, configure the APN by entering the APN Host, user name and password (see Figure 16). If the APN does not require a user name or password, leave the "APN User Name" and "APN Password" text fields blank. Then click on "Configure".

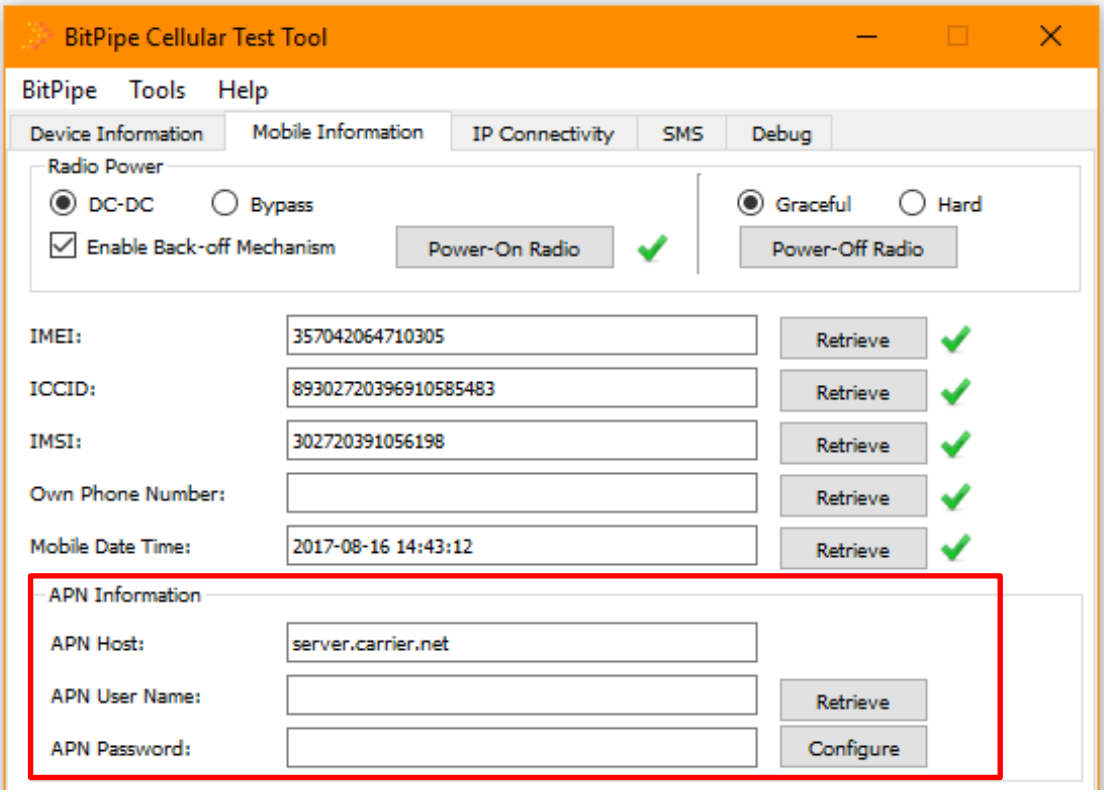

Figure 16: Configure APN

7.2 TCP/IP Protocol

The BitPipe™ modem can open sockets using the TCP/IP protocol.

*Make sure to follow the steps in section 7.1 to power on the radio and connect to a mobile network.

7.2.1 Downloading Your First Web Page using TCP/IP

Below are the steps required to download the content of a web page by sending an "HTTP GET" request to a server using the TCP interface.

1- Select the “IP Connectivity” tab and then, select the “TCP” tab (see Figure 17).

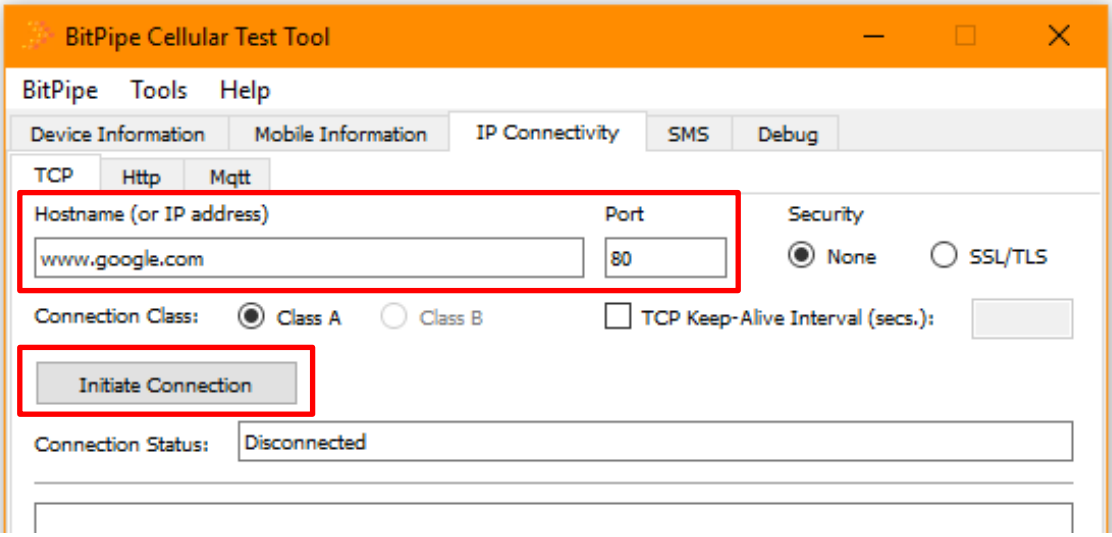

Figure 17: Establish TCP/IP connection

2- Enter the desired URL or IP address in the “Hostname (or IP address)” text box (ex: www.google.com).
3- Enter the port number to access the contents of the web page (most web pages use port 80).
4- Click on “Initiate Connection” to open a socket with the server information entered in the last two steps. If a connection was previously established with another server, it will first have to be closed by clicking on “Close connection”.

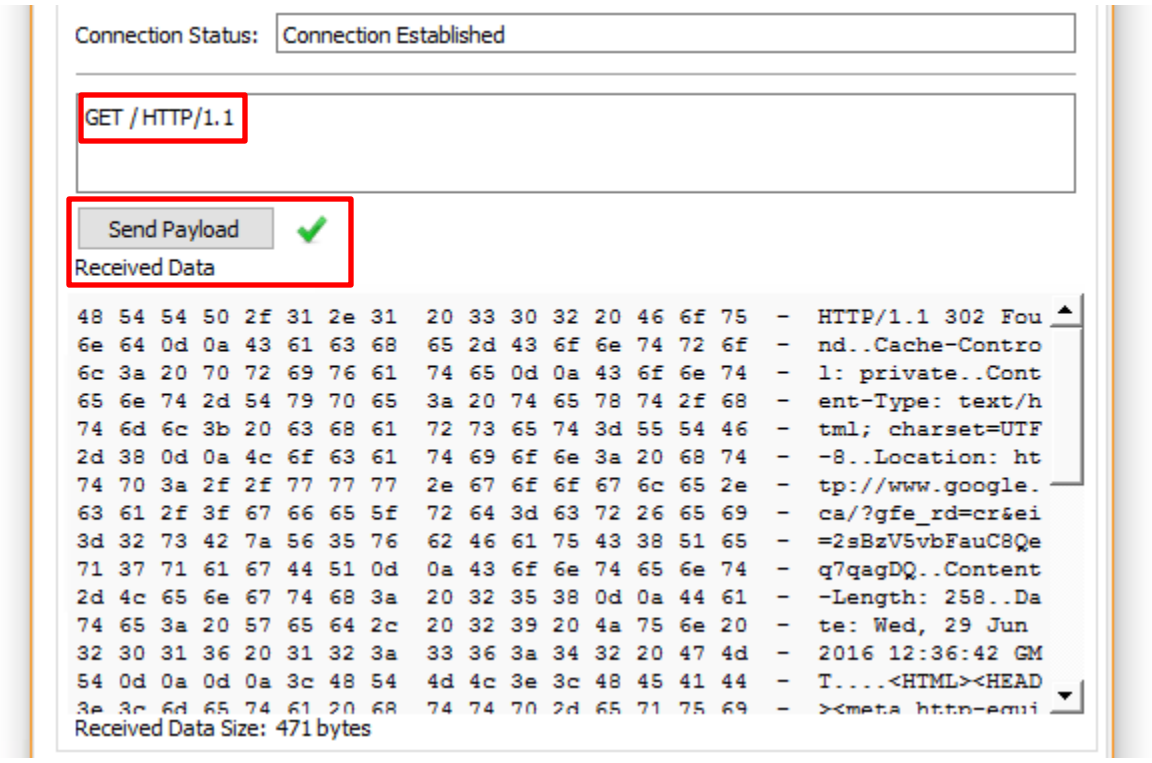

Figure 18 GET Request

5- To fetch the contents of a web page, enter a GET request in the text box above the "Send Payload" button (see Figure 18).
    a. For example, **GET / HTTP/1.1**
    b. Add a carriage return and line feed twice (Press "Enter" twice)
6- Click the "Send Payload" button
7- The Received Data text box in Figure 18 will contain the result of the HTTP GET request.

7.2.2 Sending and Receiving data using TCP/IP

Below is an example of the steps used to exchange data with a server over TCP/IP. The example uses a public TCP echo server on the internet.

1- Select the "IP Connectivity" tab and then, select the "TCP" tab.
2- Enter the desired URL or IP address in the "Hostname (or IP address)" text box (u-blox provides a public TCP echo server at dragon.u-blox.com: 195.34.89.241).
3- Enter the port number to exchange data (on the server above, the port number is 7).
4- Click on "Initiate Connection" to open a TCP socket with the server information entered in the last two steps. If a connection was previously established with another server, it will first have to be closed by clicking on "Close connection".
5- Enter a message in the text box above the "Send Payload" button.
6- Click the "Send Payload" button
7- The response content will be displayed in the "Received Data" section (see Figure 19).

You can use your own TCP echo server; if you do not have one, contact us, we can provide you access to our TCP Echo server.

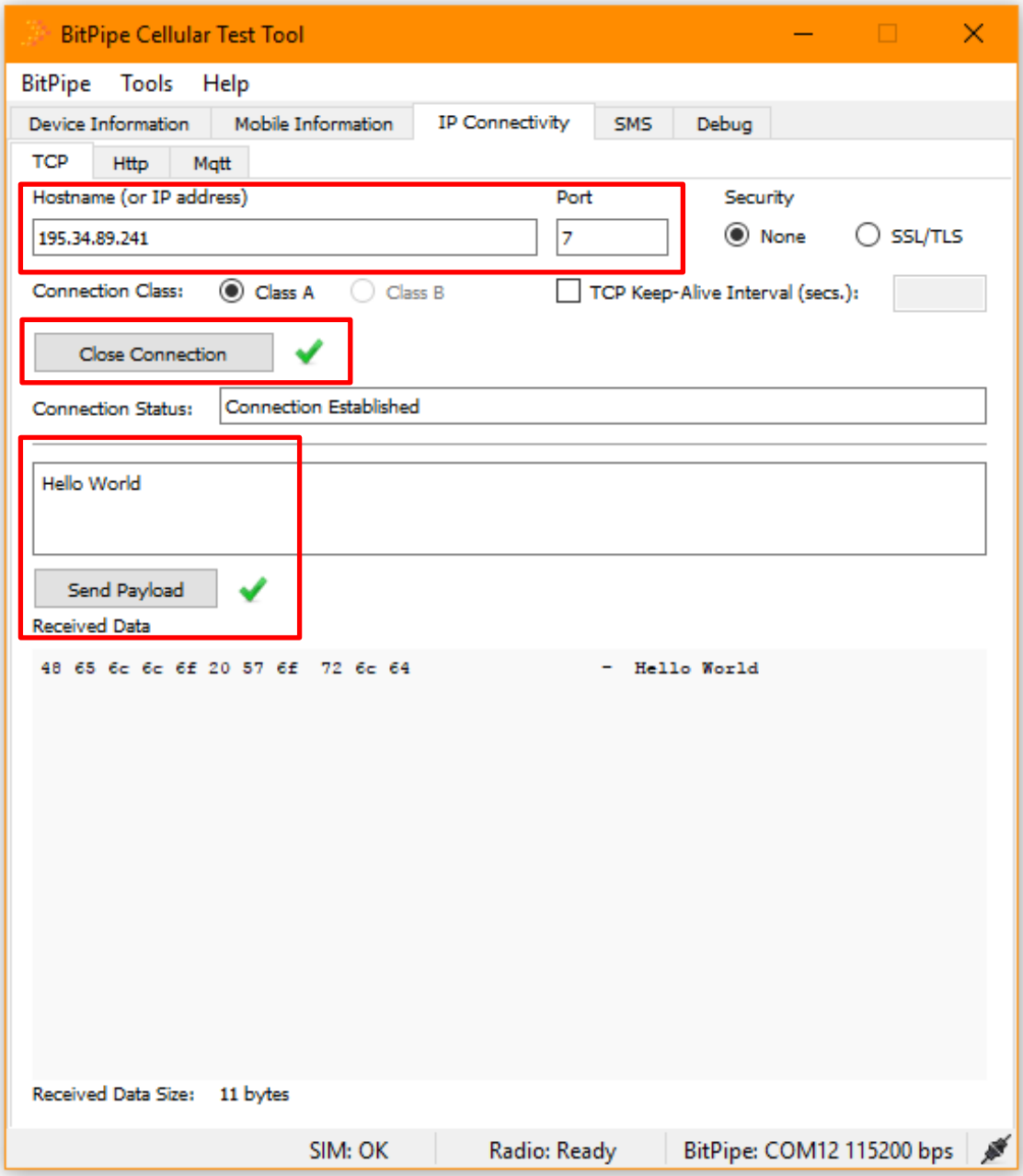

Figure 19: TCP echo

7.3 HTTP protocol

The BitPipe™ modem can also perform HTTP GET and POST requests.

*Make sure to follow the steps in section 7.1 to power on the radio and connect to a mobile network.

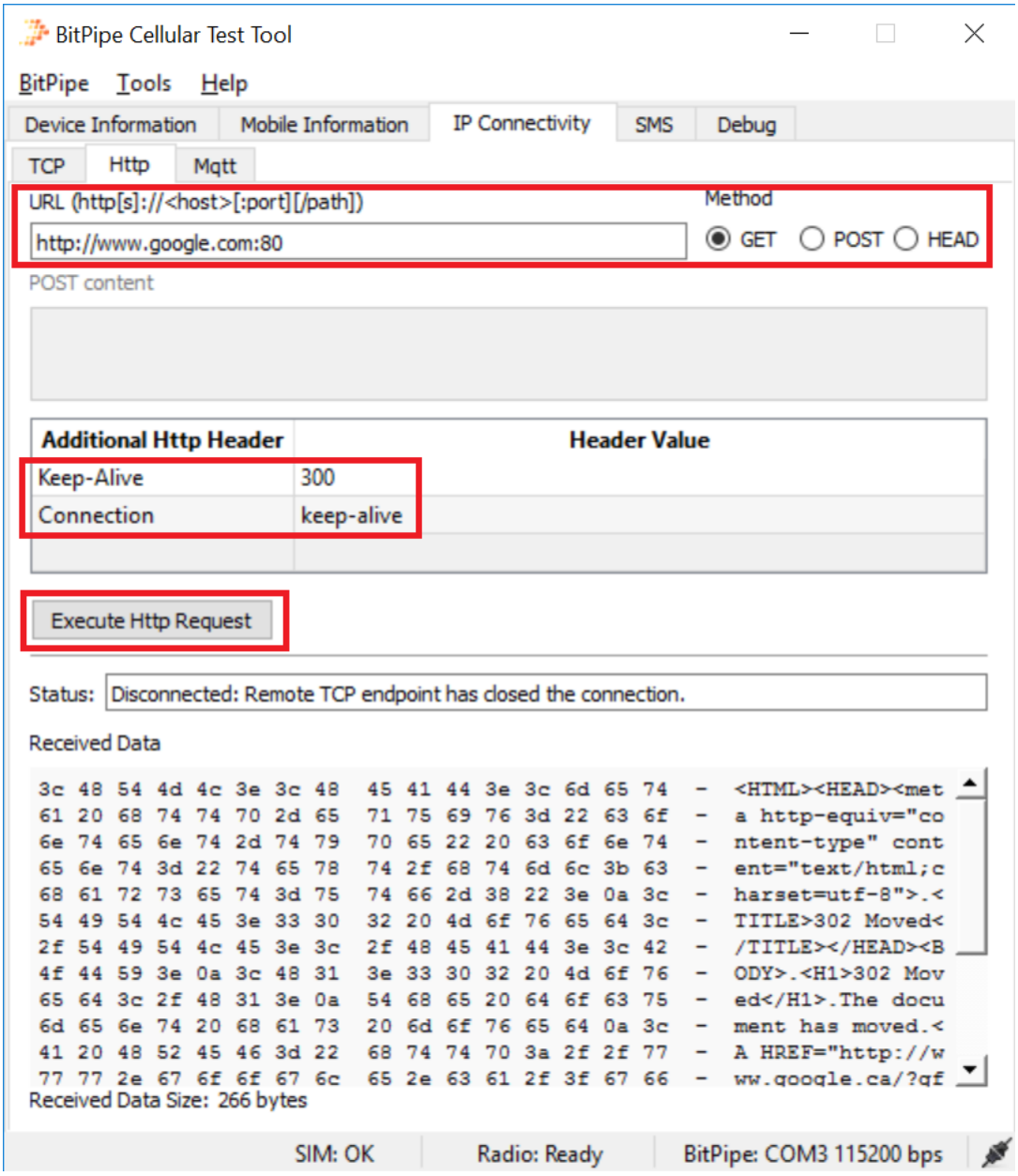

Figure 20 HTTP GET example

7.3.1 HTTP GET request

Below are the steps required to download the content of a web page by sending an “HTTP GET” request to a server using the HTTP interface.

1. Select the “IP Connectivity” tab and then, select the “Http” tab (see Figure 20).

2. Enter the desired URL or IP address in the "URL" text box.
   Use the following format: http[s]://<host>[:port][/path]
   Ex: http://www.google.com:80
3. Make sure the GET radio button to the right of the URL text box is selected.
4. Optionally, a custom HTTP header can be sent with the GET request to the server. The custom header information can be entered by specifying the header in the "Additional Http Header" column and its value in the "Header Value" column. Figure 20 shows HTTP example headers.
5. Click on the "Execute Http Request" button to initiate the connection.
6. The "Status" text field above the "Received Data" section will be updated with the status of the GET request. The following statuses can be seen during a successful request:
   a. Establishing connection ...
   b. Receiving Data …
   c. Disconnect: Remote TCP endpoint has closed the connection
7. The response content will be displayed in the "Received Data" section containing both hex values and ASCII characters (see Figure 20).

7.3.2 HTTP POST request

Below are the steps required to send data to a server using an "HTTP POST" request using the HTTP interface.
The steps below will post data to a test server: http://httpbin.org
Refer to httpbin.org for the server's expected data format.

1. Select the "IP Connectivity" tab and then, select the "Http" tab (see Figure 21).
2. Enter the desired URL or IP address in the "URL" text box.
   Use the following format: http[s]://<host>[:port][/path]
   Ex: http://httpbin.org/post
3. Make sure the "POST" radio button is selected.
4. In the "POST content" text field, enter the data to send to the server. Example: device= bitPipe & FWversion= API0.
5. Click on the "Execute Http Request" button to initiate the connection
6. The "Status" text field above the "Received Data" section will be updated with the status of the POST request. The following statuses can be seen during a successful request:
   a. Establishing connection ...
   b. Receiving Data …
   c. Disconnect: Remote TCP endpoint has closed the connection
7. The server's response content will be displayed in the "Received Data" section containing both hex values and ASCII characters (see Figure 21).

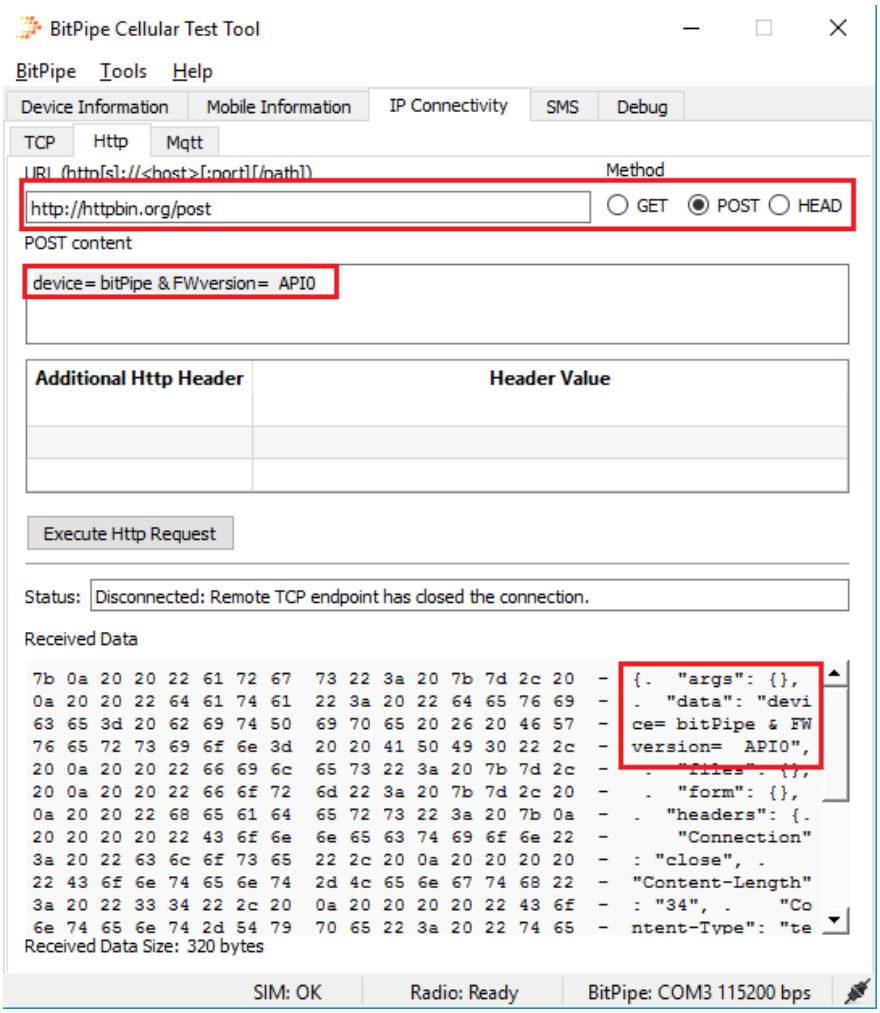

Figure 21 HTTP Post example

7.4 MQTT protocol

The BitPipe™ modem can handle the MQTT connectivity protocol. This section provides the steps to publish/subscribe messages using the MQTT protocol.

An MQTT broker and client are required for the steps in this section. The steps use the Briowireless MQTT broker. A list of available MQTT clients can be found on the official MQTT website:

http://mqtt.org/

*Make sure to follow the steps in section 7.1 to power on the radio and connect to a mobile network.

1. Select the "IP Connectivity" tab and then, select the "Mqtt" tab (see Figure 22).
2. Enter the URL or IP address of the MQTT server in the "URL" text box.
   Use the following format: mqtt[s]://<host>[:port]
   Ex: mqtt://104.196.16.233:1883

3. Check the "Keep-Alive Interval (sec.)" checkbox and enter the interval in the text box below. The interval defines the period at which the MQTT broker will be ping'ed.
4. To discard any messages from previous sessions, check the "Clean Session" checkbox. To resume a previous session (QOS1 messages published while offline will also be received), leave the checkbox blank.
5. If needed, enter the Client ID, Username and Password in their respective text boxes.
6. Click on "Initiate Connection" to connect to the MQTT broker.
7. Once the connection is successful, the "Connection Status" text box will display "Connection Established" as shown in Figure 22.

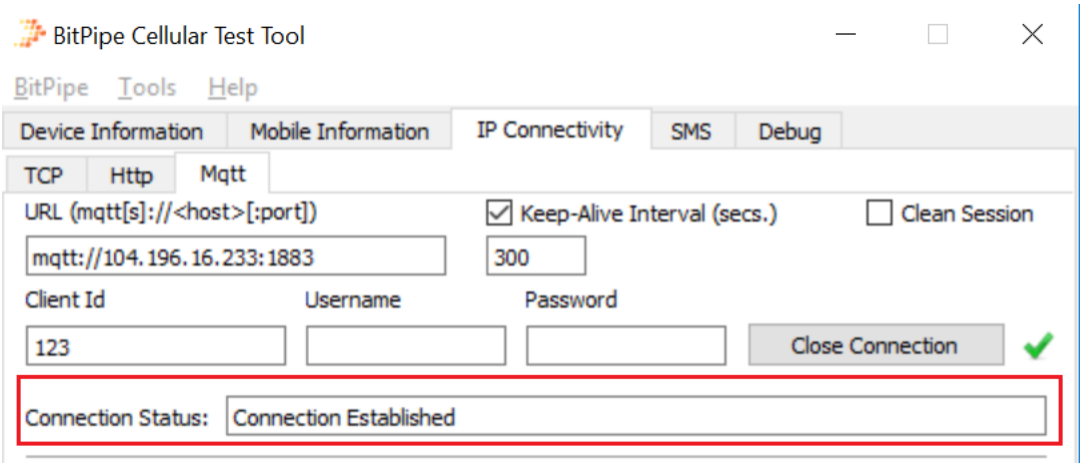

Figure 22 MQTT connection example

8. To receive publications from the MQTT broker, enter the name in the "Topic Subscriptions" text box (see Figure 23).
9. Select the Quality of Service (QOS) level from the drop-down box besides the topic name.
    a. QOS0 (at most once) defines the lowest level of QOS. The sender will send a message once and will not store the message to resend later. The receiver will not acknowledge the reception of the message.
    b. QOS1 (at least once) defines the second level of QOS. The sender will send a message to the receiver and wait for an acknowledge. Once the acknowledge is received, the sender will delete the message, otherwise, the message will be stored to be resent.
10. Click on the "Subscribe" button.
11. Once subscribed, a green check will appear next to the "Unsubscribe" button.

Figure 23 MQTT Topic subscription example

12. Any publication received from the broker can be seen by selecting the "Received Publications" tab (see Figure 24).

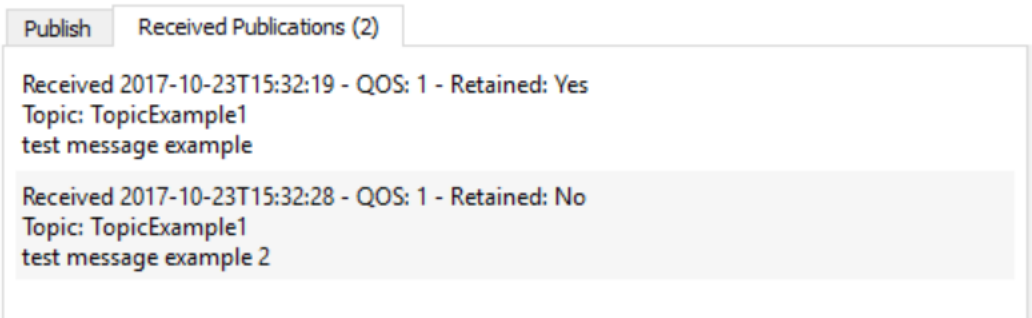

Figure 24 MQTT Received Publications

13. To publish messages to the broker, select the "Publish" tab (see Figure 24).

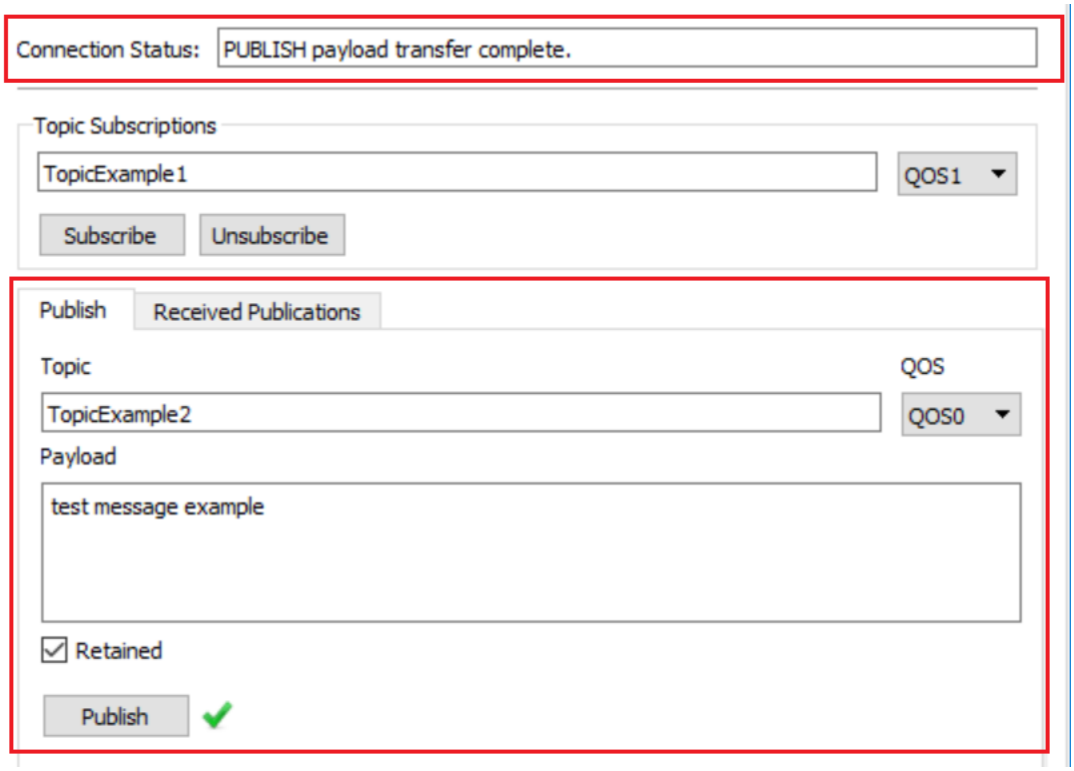

Figure 25 MQTT Publish example

14. Enter the name of the topic in the “Topic” text box.
15. Select the Quality of Service (QOS) level from the drop-down box besides the topic name.
    a. QOS0 (at most once) defines the lowest level of QOS. The sender will send a message once and will not store the message to resend later. The receiver will not acknowledge the reception of the message.
    b. QOS1 (at least once) defines the second level of QOS. The sender will send a message to the receiver and wait for an acknowledge. Once the acknowledge is received, the sender will delete the message, otherwise, the message will be stored to be resent.
16. Enter the content of the message in the “Payload” text box.
17. Check the “Retained” checkbox for the broker to send the message to newly subscribed clients upon topic subscription. Otherwise, the newly subscribed client will only receive the message after a client publishes a new message.
18. Click on the “Publish” button to send the message to the broker.
19. Once the message has been published, the “Connection Status” box should display PUBLISH payload transfer complete.

    All clients subscribed to the topic should receive the published message. Figure 26 shows the received publication by a client (MQTT.fx) subscribed to the same topic.

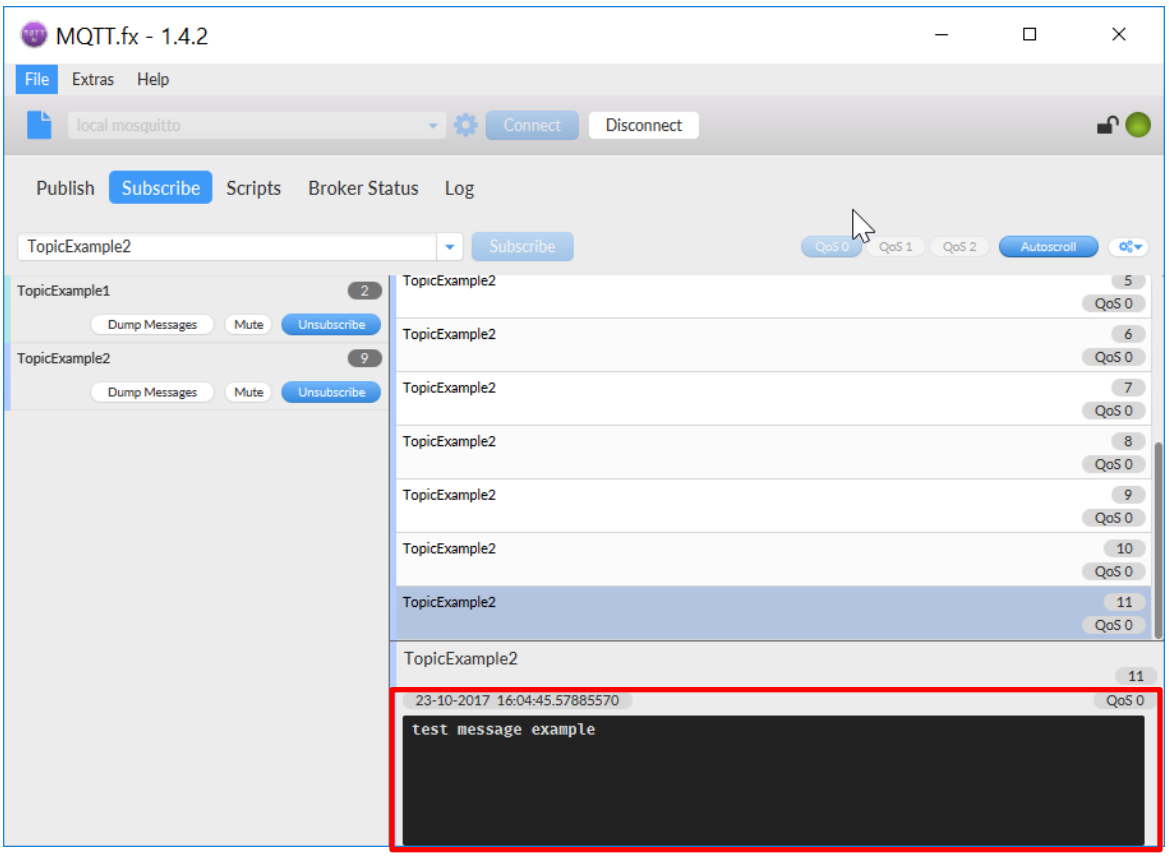

Figure 26 MQTT.fx received publications

7.5 SMS

This section provides the steps to send and receive text messages (SMS) with the BitPipe™ modem.

*Make sure to follow the steps in section 7.1 to power on the radio and connect to a mobile network.

1- Select the "SMS" tab (see Figure 27).
2- To send a message, enter the destination phone number in the **MSISDN (international Phone Number)** text box.
3- Select the message encoding.
4- Click on the "Setup SMS" button. A green check should appear next to the button.
5- Enter the message content in the text box below the "Setup SMS" button.
6- Click on the "Send SMS" button. A green check should appear next the message content.

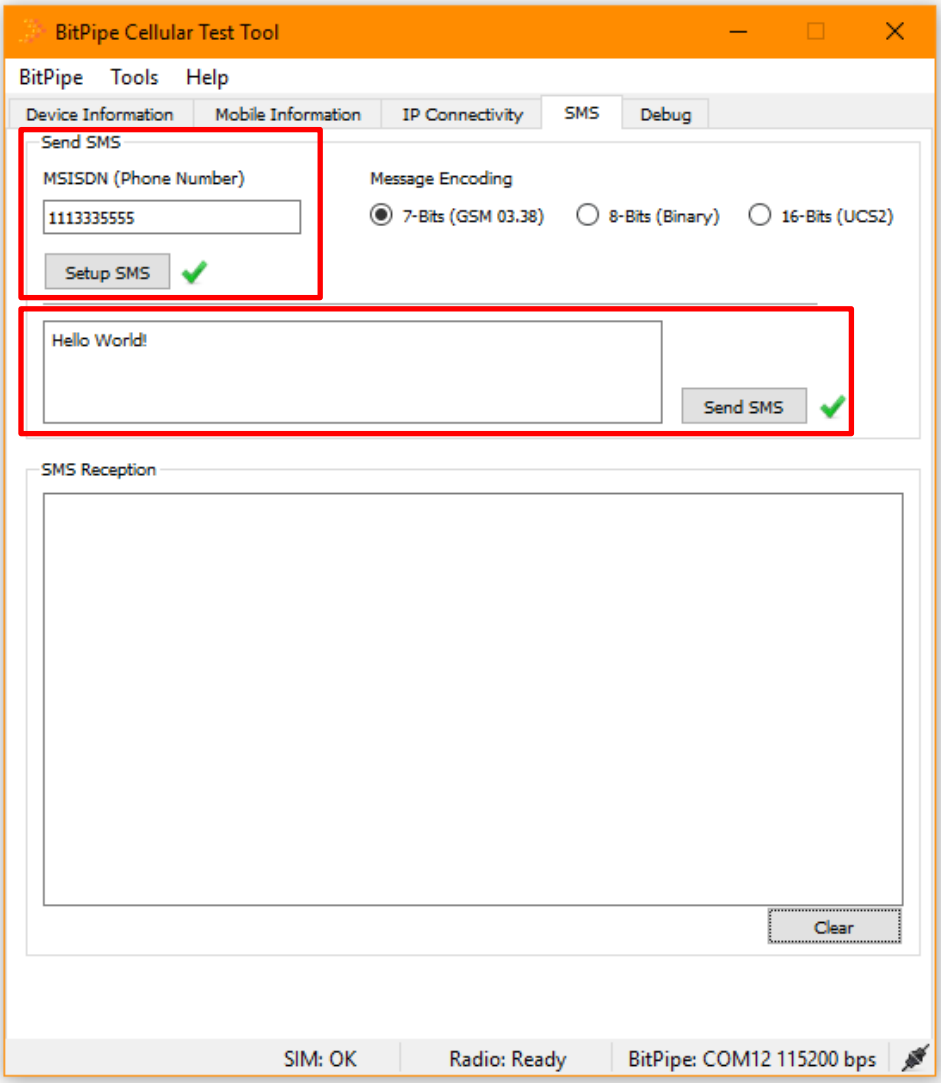

Figure 27 SMS Tab

7- The message should be received by the device specified by the phone number in step 2.
8- To test the reception of messages, send a message back to the BitPipe™ using the device that received the text message in the previous steps.
9- Once received, the contents of the SMS message will appear in the "SMS Reception" text box (see Figure 28).

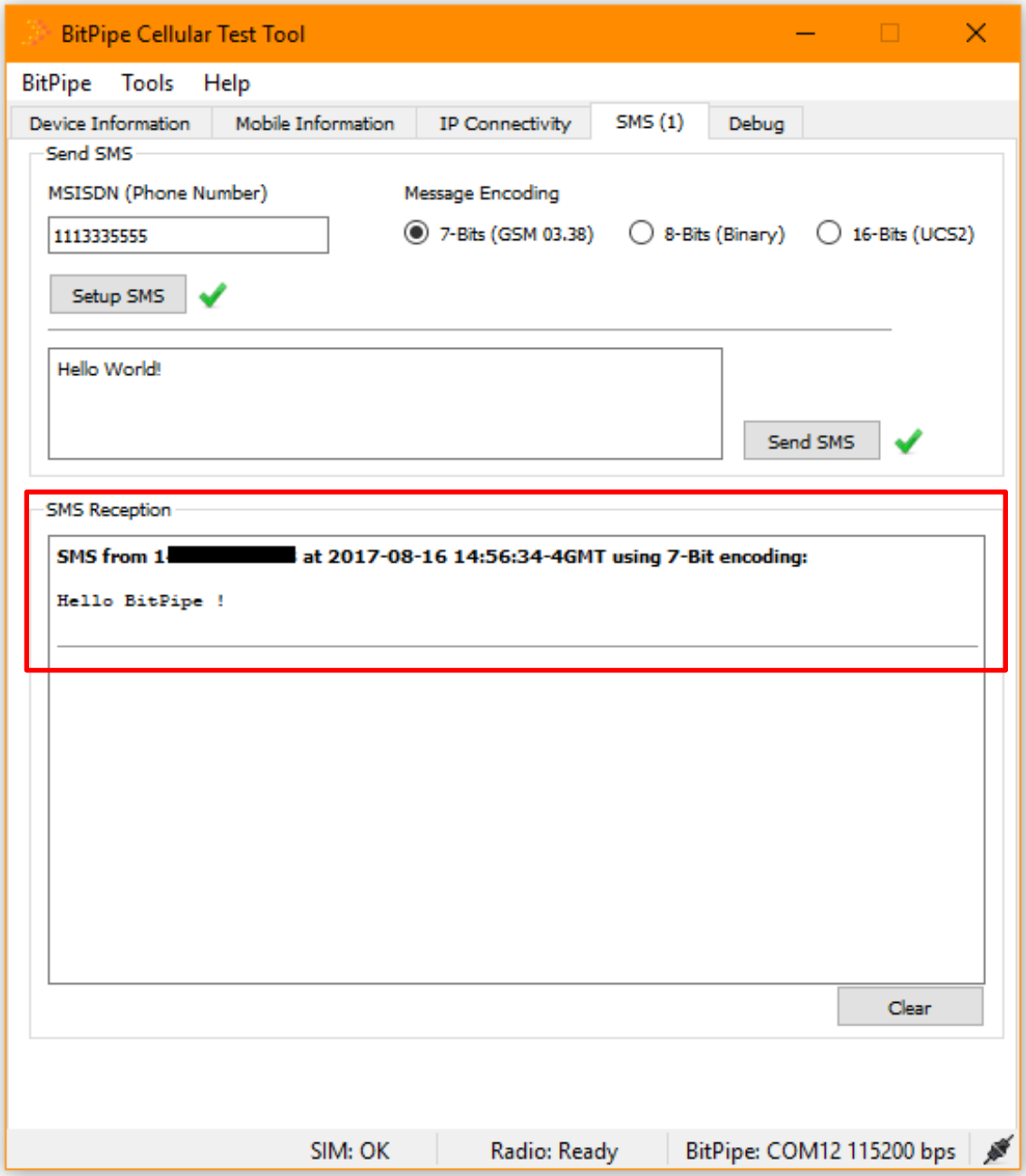

Figure 28 SMS reception

7.6 Debug Tab

The Debug tab of the BitPipe™ Cellular Test Tool is aimed at software/firmware integrators, to help them with the serial communication layer. The BitPipe™ Serial API is used to communicate with the BitPipe™ over a serial connection. This tab will help troubleshoot issues encountered while integrating the BitPipe™.

This tab allows you to see the actual message generation, transmission of a request and reception of a reply. An example is provided to retrieve the BitPipe™ firmware version. Here are the steps:

1- In the **Input Command Data** field, enter the following hexadecimal string: "11 01 00"
2- Click "Generate Message"
    a. The message is displayed in the **Generated Message** field.
3- Click "Send Message"
    a. The message is sent to the BitPipe™, a reply should be immediately visible in the **Received Reply** box below.

As shown in Figure 29, the binary data exchanged is displayed in hexadecimal notation, with the character representation on the left. This allows easy analysis of message data exchanged. The Message CRC will be generated by the Test Tool, so there's no need to enter it in the **Input Command Data**.

Refer to the BitPipe™ Serial API documentation for more information. The latest version can be found (here) in the Download section.

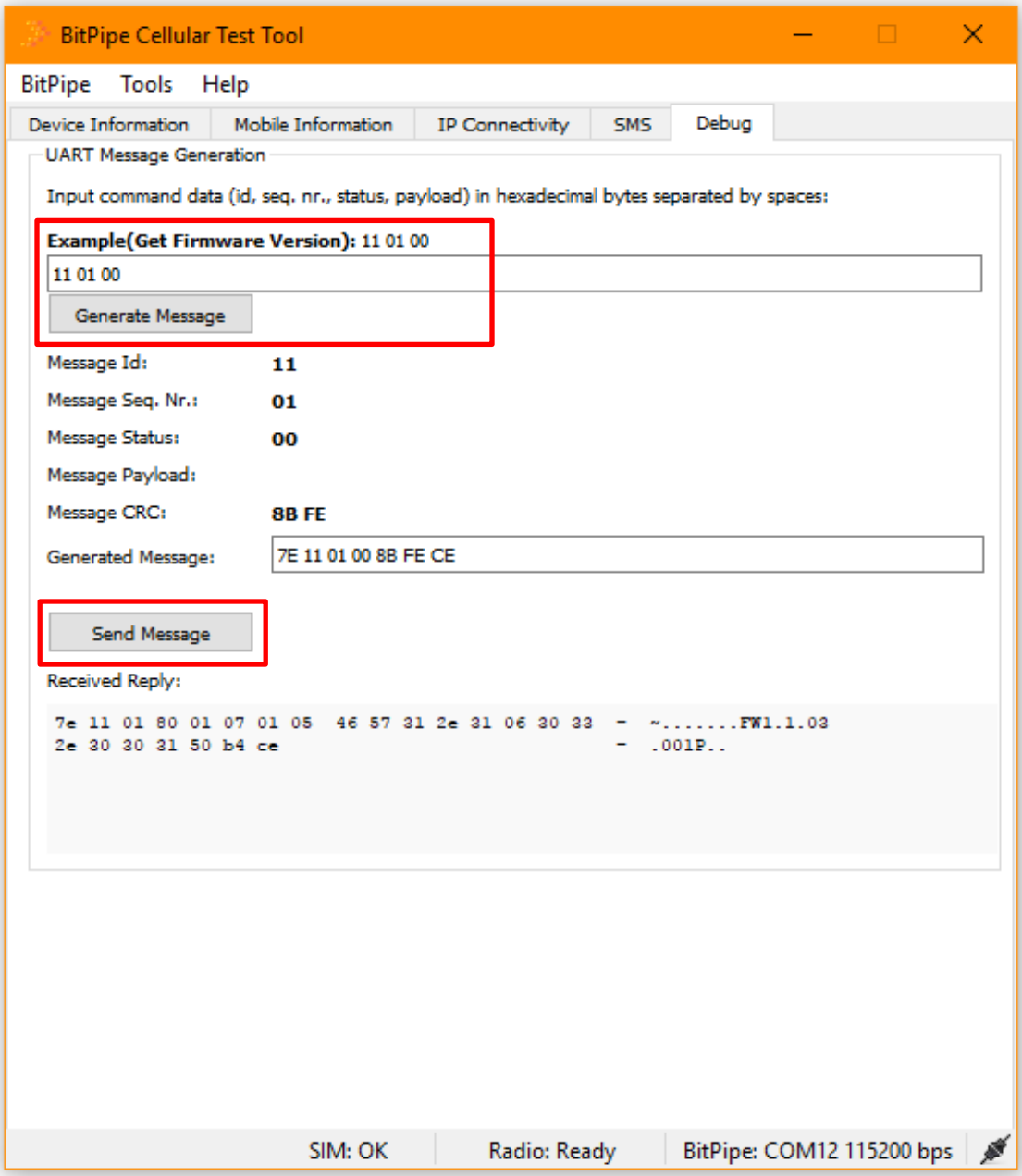

Figure 29: Debug tab

8. Autonomous Mode Demo

The **Autonomous Mode** allows the BitPipe™ to be used without a controlling Host (or application). You can read more about this operating mode in the BitPipe™ Cellular modem datasheet.

To easily demonstrate how this mode works, a demo back-end has been set up on Briowireless' servers. You may configure your BitPipe™ to use this back-end to test and see how simple it is to configure, monitor and control the BitPipe™'s IOs.

To set-up your BitPipe™ for this demo, it must be configured with Briowireless' server information. This includes setting the **MQTT Hostname**, **MQTT Port** and **Keep-Alive Interval.** For your convenience, this information can easily be configured by following these steps:

1- Open the **Autonomous Mode Configuration** dialog from the "Tools" menu
2- In the popped-up dialog, click on "Load Demo Config"
   The configuration for the demo is auto-filled in the form.
3- Click on "Configure"
4- Unplug the power supply from the BitPipe™
5- Turn the SW1-3 of the demo board to the ON position
6- Configure the dipswitches on the board as the following: **SW1-1 = OFF, SW1-2 = OFF and SW1-3 = ON**.
7- Plug back the power supply in the BitPipe™

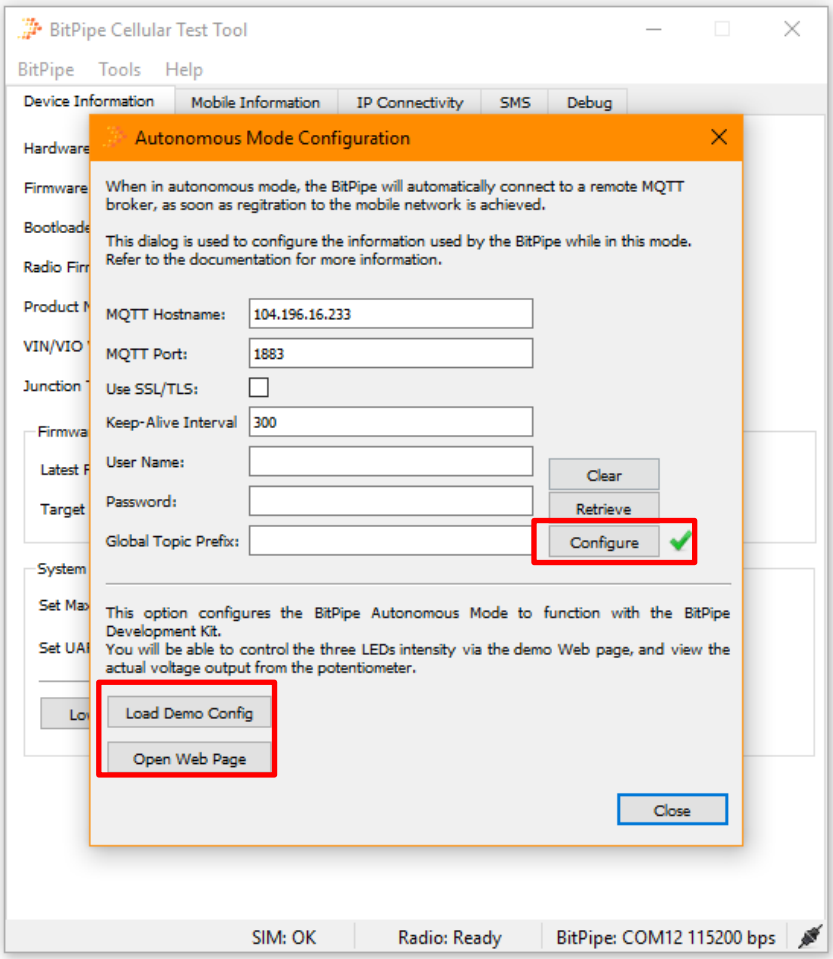

Figure 30: Autonomous Mode Configuration

After step 6, the BitPipe™ should automatically start the Radio and establish a connection to the demo servers. The APN must be configured, and an activated SIM with a data plan must be inserted in the BitPipe™. See sections 5.2 and 7.1.

From the **Autonomous Mode Configuration** dialog, press "Open Web Page" to remotely control the BitPipe™ LEDs intensity, and monitor the actual value of the thumbwheel (potentiometer), in real-time.

Note: Changing the Autonomous Mode configuration requires a power-cycling of the BitPipe™.

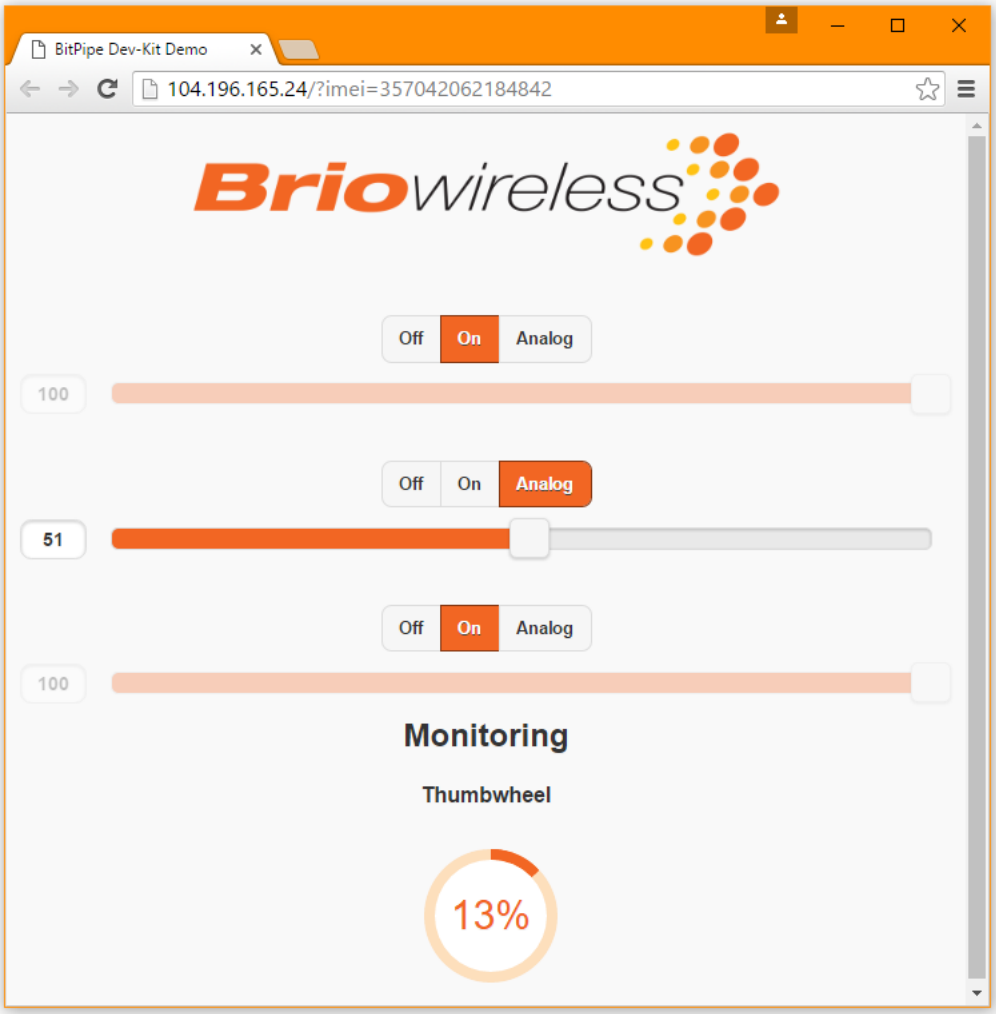

Figure 31: BitPipe™ Dev-Kit Demo Website