Overview
The RN1810 is the next generation Wi-Fi module, and is very similar to the RN171. The RN1810 supports 802.11 b/g/n and has a very similar AT-style command set as the RN171. WiFly provides embedded designers with a simple data pipe through which to send data over a Wi-Fi network yet without the need to add/remove the Wi-Fi related protocol headers and footers - in effect the designer creates a simple 'wireless serial cable'. The WiFly code operates in one of two modes; data mode and control mode. Data mode is a simple data-in data-out mode; data written to the UART is sent out over Wi-Fi and data received over Wi-Fi is written out over the UART. A programmable escape sequence transitions the module into command mode where data written to the UART (or Wi-Fi if enabled) is used to configure variables such as SSID's, pass-phrases and so forth.
1 Modifications to RN1810 Commands

This section has corrections to existing RN1810 commands.

1.1 set comm idle <value>

This command closes a TCP, TLS, or HTTP client connection after <value> seconds of no transmit or receive activity. In addition, this command can also be used for TCP and TLS Servers -- if a client has not sent any data to the server for <value> seconds the server will disconnect the client socket. A value of 0 means the module does not disconnect when the connection goes idle. For the new value to take effect, the Error! Reference source not found. command should be issued after this command.

Default: 0

Example:

```plaintext
set comm idle 5  // Close TCP connection after it has been idle for 5 seconds
set comm idle 0  // Do not disconnect when TCP connection idle
```

1.2 set ip address <address>

This command sets WiFly’s static IP address, where <address> is an IP address in the form:

- For IPv4: <value>.<value>.<value>.<value> with <value> being a number between 0 and 255.
- For IPv6: xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx (x would be a hexadecimal value)

A gateway IP address cannot be set, so this command is most useful for Soft AP mode; in Soft AP mode the static IP address and gateway address are the same.

Default: 0.0.0.0 (IPv4) 0::0 (IPv6)

Example:

```plaintext
set ip address 137.57.1.1 // set IPv4 address
set ip address 2001:db9::d0ff:474a:3798:2294 // set IPv6 address
```

1.3 set ip protocol <flag>

This command sets the IP protocol, where <flag> is a bit-mapped value described below. If none of the bits are selected than no protocol is selected.

Default: 0x02 (TCP server)

Example:

```plaintext
set ip protocol 0x12  // enable HTTP client mode and TCP server
set ip protocol 0x20  // enable TLS TCP client mode
set ip protocol 0x400 // enable FTP client
```
1.4 set sys auto <value>

This command sets the HTTP client auto-connect timer, where <value> is the number of seconds. For the new value to take effect, the Error! Reference source not found. and Error! Reference source not found. commands should be issued after this command. When the module reboots or wakes from sleep it waits <value> seconds and then opens the socket. When the "close" command is issued the socket is closed and the module waits <value> seconds before reconnecting to the server.

Range: 0 (disabled), 1-64 seconds

**Note:** The token ‘auto’ cannot be abbreviated.

Default: 0

Example: set sys auto 10 // set auto-connect timer to 10 seconds
### 1.5 set sys autoconn <value>

This command causes the RN1810 to connect to a host periodically, where <value> controls how often (in seconds) to connect to the remote host, as shown below. This command only pertains to a TCP or TLS client. For the new value to take effect, the Error! Reference source not found. and Error! Reference source not found. commands should be issued after this command.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Disable auto-connect timer</td>
</tr>
<tr>
<td>1</td>
<td>Connect to remote host immediately upon power-up or awaking from sleep mode. The values set previously via the &quot;set ip host &lt;addr&gt;&quot; and &quot;set ip remote &lt;port&gt;&quot; commands are used as the connection parameters. When a &quot;close&quot; command is issued the socket closes, but the timer is not restarted. The timer runs only after the next reboot or wake from sleep.</td>
</tr>
<tr>
<td>2:254</td>
<td>Connect to remote host every N seconds. The values set previously via the &quot;set ip host &lt;addr&gt;&quot; and &quot;set ip remote &lt;port&gt;&quot; commands are used as the connection parameters. When a &quot;close&quot; command is issued the module waits &lt;value&gt; seconds and then tries to connect to server again.</td>
</tr>
<tr>
<td>255</td>
<td>Connect to remote host immediately upon power-up or when awaking from sleep mode, and go back to sleep immediately upon closing the TCP connection. The values set previously via the &quot;set ip host &lt;addr&gt;&quot; and &quot;set ip remote &lt;port&gt;&quot; commands are used as the connection parameters. When the module reboots or wakes from sleep, it opens the socket immediately. When the &quot;close&quot; command is issued the module goes to sleep immediately.</td>
</tr>
</tbody>
</table>

**Note:** Be aware that when a client goes to sleep immediately after closing the connection the transaction has a higher chance of not succeeding. In which case, the server will think the client is still connected. If using an RN1810 server, which only supports a single connection, a client reconnecting will not be able to do so until the server times out and closes the previous connection.

Default: 0
Example: `set sys autoconn 1` // TCP client will connect immediately upon reset or awaking

### 1.6 set sys autosleep <value>

This command sets the UDP auto-sleep timer where <value> is a multiplier used in conjunction with Error! Reference source not found. (see Section Error! Reference source not found.). For example, the following two commands would create a UDP sleep timer of 400ms:

```
set comm time 100
set sys autosleep 4
```

The resulting timer is 100 * 4, or 400ms.

For the new value to take effect, the Error! Reference source not found. and Error! Reference source not found. commands should be issued after this command. When the client sends data to the server the timer is started. Upon timeout, the module goes to sleep.

**Note:** The token ‘autosleep’ cannot be abbreviated.

Default: 0
Example: `set sys autosleep 4`
1.7 set sys sleep <value>
This command is only applicable when the RN1810 is configured as a TCP client. The command specifies the duration, in seconds, the RN1810 waits after a TCP connection is closed before it goes to sleep. For the new value to take effect, the Error! Reference source not found. and Error! Reference source not found. commands should be issued after this command.

Default: 0 (disabled)
Example: set sys sleep 10 // Module goes to sleep 10 seconds after TCP connection closes.

1.8 set uart mode <value>
This command configures the WiFly UART special modes, where <value> is a bit-mapped number as shown below.

```
<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

Echo -
0: Echo received command characters (default)
1: Disable echoing of received command characters

TCP_Connection -
0: Do not start TCP connection (default)
1: Start TCP connection upon receiving first Rx character after boot. The Rx character will not be sent to the server.

Default: 0x0
Example: set uart mode 0x02 // start TCP connection upon receiving first Rx character after reboot

1.9 set wlan auth <value>
This command sets the authentication mode, where <value> is show below. The firmware supports the following security modes:

- WEP-64 and WEP-128 (open mode only, not shared mode)
- WPA2-PSK (AES only)
- WPA-PSK (TKIP only)

```
<table>
<thead>
<tr>
<th>Value</th>
<th>Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Open (Default)</td>
</tr>
<tr>
<td>1</td>
<td>WEP-128 Open Key</td>
</tr>
<tr>
<td>2</td>
<td>WPA-PSK TKIP</td>
</tr>
<tr>
<td>4</td>
<td>WPA2-PSK AES</td>
</tr>
<tr>
<td>8</td>
<td>WEP-64 Open Key</td>
</tr>
</tbody>
</table>
```

Default: 0 (open)
Example: set wlan auth 4 // set security to WPA2-PSK authentication
1.10 set wlan join <value>
This command sets the policy for automatically associating with network access points, where <value> is one of the options shown below. The module uses this policy on power up, including waking up from the sleep timer.

<table>
<thead>
<tr>
<th>Value</th>
<th>Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Manual. Do not try to associate with a network automatically.</td>
</tr>
<tr>
<td>1</td>
<td>Try to associate with the access point that matches the stored SSID, passkey, and channel. If the channel is set to 0, the module will scan for the access point. (Default)</td>
</tr>
<tr>
<td>7</td>
<td>Create a soft AP network using the stored SSID, IP address, netmask, channel, etc. The module must be in IPv4 mode before creating a Soft AP network. (e.g. set ip v 0). Soft AP supports a maximum of 10 clients. Note that if the Soft AP module is configured as a TCP server, that server can still only support a single client.</td>
</tr>
</tbody>
</table>

Default: 0
Example: set wlan join 7 // Create soft AP network

1.11 set wlan phrase <string>
This command sets the passphrase for WPA and WPA2 security modes, where <string> is 1 to 64 characters (64 bytes). The passphrase is alphanumeric, and is used with the SSID to generate a unique 32-byte Pre-Shared Key (PSK), which is then hashed into a 256-bit number. When you change either the SSID or the passphrase, the module recalculates and stores the PSK.

Default: None
Example: set wlan phrase my_password // set passphrase to my_password

1.12 set wlan ssid <string>
This command sets the SSID with which the module associates, where <string> is 1 to 32 characters (32 bytes). Note: <string> cannot contain spaces.

Default: None
Example: set wlan ssid my_network // set SSID to my_network

1.13 set wlan tx <value>
This command sets a fixed transmit power level for the RN1810 module, where <value> is a value between 1 and 17 dBm.
Default: 16
Example: set wlan tx 8 // set transmit power to 8dBm

1.14 get sys
This command was incorrectly named “get system” in the original documentation
1.15 apmode <ssid> <channel>
This command initiates soft AP mode. Valid inputs are:

- `apmode`
- `apmode sssid`
- `apmode sssid channel`

The module must be in IPv4 mode for this command to function correctly (e.g. “set ip v 0”).

**Note:** When Soft AP mode is invoked, the IP address of the module is 192.168.1.10. The default DHCP server pool is 192.168.1.11 through 192.168.1.20. The IP address of the module can be changed via the set ip address <address> command. The DHCP server pool and lease time can be changed via the set dhcp lease command. However, the gateway address is unchangeable and remain as 192.168.1.10 which must not cause any issue in network packet distribution.

Example:
- `apmode`  // start Soft AP network using previously set SSID and channel
- `apmode my_app`  // start Soft AP network with SSID set to my_app and previously set channel
- `apmode my_ap 6`  // start Soft AP with SSID set to my_app and using channel 6

1.16 join <string>
This command instructs the WiFly module to join the network indicated by <string>. If the network has security enabled, you must first set the passphrase with the `set wlan phrase` command prior to issuing the `join` command.

**Note:** The string must not contain any spaces.

Example:
- `1) join`  // Join previously saved SSID (see set wlan ssid <string>)
- `2) join mchp`  // Join open network mchp1
- `3) set wlan pass password`  // Set the password to password
  `join mchp1`  // Join network mchp1

1.17 leave
This command instructs the WiFly module to leave the Wi-Fi network it is currently associated with. When the leave command is issued in SoftAP mode or web_app mode a reboot is required before restarting these modes.

Example: `leave`  // Leave current network
1.18 ota upgrade <bin_filename> [ip_addr|domain_name] [username] [password]

This command upgrades the module firmware using a Wi-Fi connection to an FTP server.

- **bin_filename**
  - Name of the OTA binary. This is a mandatory argument.

- **ip_addr | domain_name**
  - IP Address or Domain Name of the FTP server that will serve the OTA. This is a mandatory argument.

- **username**
  - Username credential to access the ftp server ftp://rn.microchip.com/public. If not provided, the default username “roving” will be used.

- **password**
  - Password credential to access the ftp server ftp://rn.microchip.com/public. If not provide, the default password of Pass123 will be used.

**Note**: The OTA upgrade only supports downloading the OTA firmware from the /public folder. It does not support bin files being in other directories. For example, if the domain_name is specified as example.com, then the OTA binary file must to be present in example.com/public.

**Note**: If the OTA transfer fails, the module must be reset in order to prevent memory corruption before doing any other operations.

**Example**: The following examples presume the bin_filename is wifly.bin.

```
ota upgrade wifly.bin example.com
ota upgrade wifly.bin 192.168.1.10
ota upgrade wifly.bin rn.microchip.com roving
ota upgrade wifly.bin 192.168.1.10 roving
ota upgrade wifly.bin rn.microchip.com roving Pass12
ota upgrade wifly.bin 192.168.1.10 roving Pass123
```

1.19 run <string>

This command runs applications, where **<string>** is:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wps</td>
<td>Run WPS application in infrastructure mode</td>
</tr>
<tr>
<td>web_app</td>
<td>Run web application in soft AP mode</td>
</tr>
</tbody>
</table>

**Note**: A browser refresh may cause a problem when running web_app. If currently connected to an AP the "leave" command must be issued prior to running web_app.

**Example**: run web_app  // run web application server application

1.20 sleep

This command puts the module to sleep. You can wake the module by sending characters over the UART or by using the wake timer.

**Note**: This command is not allowed in Soft AP mode.

**Example**: sleep  // put the module to sleep
1.21 Soft AP Mode
Note that up to 10 clients are supported in Soft AP mode.

1.22 WPS Mode
Note that WEP security is not supported in WPS mode.

1.23 Auto-Connection and Sleep Timers
Change first paragraph to:
The RN1810 can be configured to periodically sleep and wake up. These actions are triggered by various timers as described below, along with relevant commands. Below are the timer descriptions:

Table 4-4: Timers for Various Operations

<table>
<thead>
<tr>
<th>Timer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep Timer</td>
<td>Determines in seconds how long the module must sleep. This is a 32-bit value, corresponding to a maximum value of $2^{32}/1000$, (4.29MSEC or 1191 hours). The sleep timer is set with the set sys sleep &lt;value&gt; command.</td>
</tr>
<tr>
<td>Wake Timer</td>
<td>Determines in seconds how long the module must remain in Sleep mode before waking up. This is a 22-bit number corresponding to a maximum value of 1,165 hours. The wake timer is set with the set sys wake &lt;value&gt; command.</td>
</tr>
<tr>
<td>TCP Auto-Connect Timer</td>
<td>Determines in seconds how long the module must wait after reboot before opening a TCP connection. The Auto-connect timer is set with the set sys autoconn &lt;value&gt; command.</td>
</tr>
<tr>
<td>UDP Sleep Timer</td>
<td>Determines, in seconds, how long to sleep after sending a UDP packet. See set sys autosleep and set comm timer commands.</td>
</tr>
<tr>
<td>HTTP Auto-Connect Timer</td>
<td>Determines in seconds how often an HTTP client must open a connection to a HTTP server. The timer is set with the set sys auto &lt;value&gt; command.</td>
</tr>
<tr>
<td>Idle Timer</td>
<td>Determines in seconds how long it takes to close a TCP connection that is idle. The Idle timer is set with the set comm idle &lt;value&gt; command.</td>
</tr>
</tbody>
</table>

1.24 Scan Output Security Mode Bit Mask
The Scan Output format Security Mode bit mask should be modified to:

<table>
<thead>
<tr>
<th>15:10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>8021X</td>
<td>PSK</td>
<td>N/A</td>
<td>CCMP</td>
<td>TKIP</td>
<td>WEP</td>
<td>N/A</td>
<td>N/A</td>
<td>WPA2</td>
<td>WPA</td>
</tr>
</tbody>
</table>

If no bits are set in the mask then the AP has open security.
2 New RN1810 Commands
This section describes commands that are new to RN1810 v1.1.

2.1 cert client add <ip_addr> <port>
This command downloads a client certificate from the specified TCP server, where <ip_addr> is the IP address of the server and <port> is the port number of the server. The certificate is stored in the RN1810 FLASH. Upon successful completion the string “CERTEND:\r\n” will be output.

Default: N/A

Example: cert client add 10.0.1.9 50008

2.2 cert server add <ip_addr> <port>
This command downloads a server certificate from the specified TCP server, where <ip_addr> is the IP address of the server and <port> is the port number of the server. The certificate is stored in the RN1810 FLASH. Upon successful completion the string “CERTEND:\r\n” will be output.

Default: N/A

Example: cert server add 10.0.1.9 50008

2.3 cert client del
This command deletes the previously downloaded client certificate from RN1810 FLASH.

Default: N/A

Example: cert client del

2.4 cert server del
This command deletes the previously downloaded server certificate from RN1810 FLASH.

Default: N/A

Example: cert server del

2.5 set sys power_mode <value> [ this is a new command]
This command sets the RN1810 power mode, of which there are two possible values:
   1: Use power-save mode (802.11 sleep with legacy/WMM PS-Poll)
   2: Max performance (never goes into 802.11 sleep mode)

Default: 2

Example: set sys power_mode 1 // use power save mode 
set sys power_mode 2 // max performance mode
2.6 **get comm**
This command displays the UART and flush timer settings

Example: `get comm`

2.7 **set apmode auth <value>**
This command sets the security to be used for AP Mode. Values are 0 (open security) or 4 (WPA2_PSK_AES)

Default: 0
Example: `set apmode auth 0` // use open security in Soft AP mode

2.8 **set apmode max_clients <value>**
This command sets the maximum number of clients that can connect in Soft AP mode. By default, the RN1810 will calculate the max supported stations based on the DHCP server pool (either the default pool or the user configured pool). The default max stations is 10 for the default server pool. If the user configures the DHCP server pool for a lesser number, say 5, and then the max supported clients will automatically be set to 5. The user can then restrict this max number to, say, 3 using this command

Default: See description
Example: `set apmode max 5` // set a maximum of 5 Soft AP mode clients

2.9 **set apmode inactivity <time_in_minutes>**
This command sets the inactivity timeout for Soft AP clients that stop communicating – If a client has no activity but has not disassociated from the AP, then this timer (in minutes) will trigger the disassociation of the client. This is useful in cases where the lease time is set to 86400 (1 day), the client goes offline, and the user wants to be able to let other clients associate without resetting the module. This command does not require a save and reboot, but takes effect immediately.

Default: 0 (use DHCP lease time)
Example: `set apmode inactivity 5` // set inactivity timer to 5 minutes.
2.10 rftest <rate> <packetCount> <packetSize> <headerType> <headerType>

This command can assist in regulatory testing.

rate:
    0 = 1mbps, 1 = 2mbps, 2 = 5.5mbps, 3 = 11mbps, 4 = 6mbps, 5 = 9mbps, 6 = 12mbps,
    7 = 18mbps, 8 = 24mbps, 9 = 36mbps, 10 = 48mbps, 11 = 54mbps, 12 = 6.5mbps,
    13 = 13mbps, 14 = 19.5mbps, 15 = 26mbps, 16 = 39mbps, 17 = 52mbps, 18 = 58.5mbps

packetCount:
    Number of packets to transmit (1 through 14)

packetSize:
    payload size in bytes (0 through 1400)

channel:
    Channel number (1 through 14)

headerType:
    0 = beacon
    1 = QOS frame
    2 = address data frame
3 TCP Server

This section describes scenarios that can occur when the RN1810 is configured as a TCP server. Presume a client has connected to the server. At some point the client, for whatever reason, can lose connectivity with the server before it can inform the server that it is closing the connection. Several scenarios can occur:

a) If a client tries to connect, then the connection is denied, since only one client is allowed

b) If user enters “close” at the TCP Server – The Server disconnects, and future client connections work without any issues

c) If user enters a couple of characters at the TCP Server, they are never sent since the client’s connection is dead.
   1. If a client tries to connect at this point, then the connection is denied, since only one client is allowed.
   2. If user enters “close” now at the TCP Server – The Server disconnects, and future client connections work without any issues

d) If user keeps entering characters at the TCP Server till “ERR TCP Send Buf Full” message, then the socket is closed automatically at the RN1810 server, but if a client attempts a connection, the connection will be successful but only the client will be able to send data to the server. The server will be able to send data to the client only after about 10 minutes.
4 Creation of CA Certificate and Server Certificate

This section describes how to create CA and Server certificates that can be utilized by the RN1810 WiFly module. The creation of the certificates makes use of the Mako Server, which can be downloaded from https://makoserver.net/download/windows/. The following discussion pertains to the Microsoft Windows version of the Mako Server. Download the “make.windows.x86.exe” and install it. The server should start automatically, but if it does not, it can be started by going into the install directory and run MAKO_TUTORIAL.cmd.

4.1 Creation of CA Certificate

Open a web browser and enter the address: http://localhost/certmgr/cacert.lsp. The link will open a page to create the certificate database. Fill in the details as below:

![Create Certificate Database](image)

The ‘Base directory’ is the directory that will, after completion, contain the binary files required for the RN1810 SSL. Click ‘Submit’ when finished.
The following page appears which is used to create the key and certificate. Enter the information and click ‘Create Key & Certificate’.
The CA Root is created and following page is displayed:

The .perm and .bin files will be in the base directory defined earlier. For example:

```
...\certmg-db/RSA/ca.perm
...\certmgr-db/RSA/RSA-ca-perm.bin
```

The file `RSA-ca-perm.bin` will be downloaded to the RN1810 configured as a TLS client. The file will need to be copied to a TCP server. The RN1810 will download this file from the TCP server and copy the certificate file to its FLASH where it will be used for server validation.

### 4.2 Creation of Server Certificates

Select ‘Create Certificate’, enter the required information, and click ‘Create Key & Certificate. This will create a private key and the signer server certificate.
The next page to come will be the following, showing the key and certificate creation succeeded.

The base directory will contain the private key and server certificate files. For example:

```
...\certmg-db\RSA\keys-and-certs\RN1810.key    [private key]
...\certmg-db\RSA\keys-and-certs\RN1810.perm   [server certificate]
...\certmg-db\RSA\keys-and-certs\RSA-RN1810.bin [binary file required by RN1810]
```

The RSA-RN1810.bin file is a SHARK-SSL-specific binary file that includes both the .key file and the .perm file. The file will need to be copied to a TCP server. The RN1810 will download this file from the TCP server and copy the certificate file to its FLASH where it will be used to notify clients of its authenticity.

The issued self-signed certificate can be seen by clicking on ‘Issued Certificates’:
4.3 Serving Certificates via a TCP Server
This section describes how to create simple Windows TCP server that can server both CA Certificates and binary files to the RN1810 by running the provided cert_server.exe:

```plaintext
cert_server.exe <bin_filename> <server_port>
```

Examples:
- `cert_server.exe RN1810-ca-perm.bin 50008` [CA Certificate]
- `cert_server.exe RSA-RN1810.bin 50008` [Server Certificate]

5 Changes to Pin Descriptions
This section has corrections to RN1810 pin descriptions.

5.1 RESET Pin
Set by Host to force WiFly module reset (identical to the reboot command). To force a WiFly reset (presuming CHP_PWD_L is normally high):
1. Set RESET low
2. Delay at least 200us ns
3. Set RESET high

**Note 1:** This pin must be configured as open drain or the RN1810 cannot drive this pin low when it goes into Sleep state, and the Sleep state do not work correctly

**Note 2:** If the module is in Sleep mode it must be first taken out of Sleep mode via the WAKEUP pin. Then this RESET pin can be toggled.

5.2 WAKE Pin
Takes WiFly out of Sleep state. The pin should be set high during initialization. To wake up the RN1810 pulse the line low, then high. The WAKEUP pin can be shorted to the UART0_RX pin only if the application wakes up a sleeping RN1810 when it receives a character.

6 Known Issues

6.1 RN1810's Communicating With Each Other
This issue occurs in the following scenario:
- An RN1810 has created a Soft AP network using WPA-PSK security
- A client RN1810, using the default phy mode, attempts to communicate with the Soft AP RN1810

To work around this issue, the client RN1810 must set its phy mode to 11B or 11G. This can be done via the commands:
- `set wlan mode_phy 2` // 11B or
- `set wlan mode_phy 3` // 11G

6.2 Maximum UDP Client Packet Size
The maximum size packet a UDP client can receive is 1472 bytes.
6.3 Broadcast

If the RN1810 wants to broadcast to all the devices on the network, then it should be configured to be a UDP client, with the host address as the broadcast address of the network that the device is on. Broadcast address of a network is dependent on the subnet mask.

For e.g.: If the RN1810 has an ip address of 192.168.1.100, and has a subnet mask of 255.255.255.0, then the broadcast address is 192.168.1.255 to reach all the devices on this network. The “show ip” command will show the RN1810’s ip address and the network mask, and the broadcast address can be deduced from that.

Once the broadcast address is known then use the following command-set to setup the RN1810 for broadcast (broadcast address of 192.168.1.255):

```
set ip protocol 0x1
set ip host 192.168.1.255
set ip remote 2000
save
reboot
```

<<<Send Data --- will be broadcast on 192.168.1.255 port 2000, and any device with 192.168.1.xx ip address on this network should be able to receive this on port 2000>>>

If one of the devices that is interested in this broadcast is an RN1810 device, then it should be configured to be a UDP server, listening on port 2000 for the example given here:

```
set ip protocol 0x100
set ip local 2000
save
reboot
```