Eaton EMCB
Local Networking UDP API

April 10, 2019
Revision 1.0.1

Table of Contents

Revision History

Date	Version	Author	Description
March 7, 2019	1.0.0	Austin Eldridge	Initial Document Public Release
April 10, 2019	1.0.1	Austin Eldridge	Remove Broadcast vs. Unicast discrimination limitation

1 Introduction

This document describes the proprietary communications protocol implemented by the EMCB for local area network communication. In order to leverage this protocol, a device must be properly commissioned and UDP Keys set by leveraging the EMCB's Cloud API. In order to preserve cybersecurity best practices, the UDP Keys will expire after 7 days rendering the API unusable – this means these keys must be refreshed using the Cloud interface weekly.

An example implementation of leveraging this API using established best practices written in node.js is available to accompany this document. The examples listed in Section 5.7 contain log output generated by this tool.

2 Current Implementation Limitations / Known Issues

2.1 IP Address ranges

To ensure that security is not compromised, the EMCB is limited to make connections to local/private IPv4 addresses only. In other words, the EMCB will only respond to local/private IP's in the ranges 192.168.0.0—192.168.255.255, 10.0.0.0—10.255.255.255, or 172.16.0.0—172.31.255.255.

2.2 Internet Connectivity

The current release requires a constant and reliable connection to the internet and Eaton's backend servers despite the fact that the EMCB can be communicated with directly.

This limitation will be removed in a future release.

2.3 Release Quality

This software is currently available in a very early pre-release version and has not undergone complete Quality Assurance testing at any meaningful scale. While the API is functional, it is subject to change until complete testing (including cybersecurity) has been completed.

3 Protocol Overview

The EMCB will communicate over a local area network using the custom Application Layer described in this document. The User Datagram Protocol (UDP) Transport Layer on an Internet Protocol (IP) Network Layer via the Wi-Fi Physical Layer will be used by the EMCB. EMCB's will listen to the network's broadcast address as well as their individual IP addresses (for unicast messages) on port 32866 ("Eaton" on a phone keypad) and will reply to the source address and port for received and validated datagrams.

IT IS IMPORTANT TO NOTE THAT ANY INVALID DATAGRAMS WILL FAIL SILENTLY, WITH NO RESPONSE.

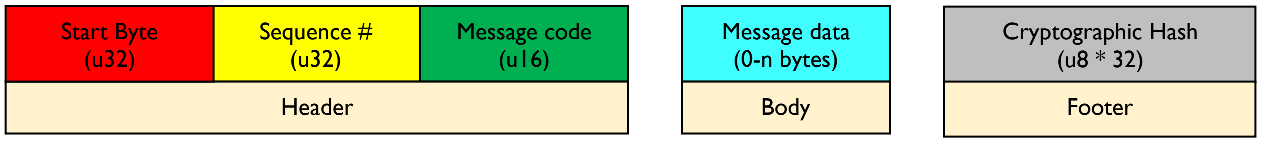
4 Application Layer

The protocol operates in a master->slave(s) mode where every transaction is initiated by the on-premises master (a local controller) and terminated by a response from the EMCB slave(s).

The master shall only retry a transaction or start a new transaction where there was no slave response 200ms after the start of the transmission of the command.

Invalid messages are NOT responded and will fail silently.

4.1 Message format



4.1.1 Start Byte

The hex code 0x45544E4D ("ETNM" in ASCII) is used to indicate the start of a new message from the Master to the Slave. The hex code 0x45544E53 ("ETNS" in ASCII) is used to indicate the start of a new message from the Slave to the Master. Note - these values may appear elsewhere in the message.

4.1.2 Sequence Number

A 32-bit unsigned counter that starts at a random number on boot and counts upwards mod 2^32 in sequence space arithmetic. This is used as a nonce to prevent replay attacks and will be the same for the request from the master and the response from the slave. See *Section 4.3 - Sequence Number in Detail* for more information.

4.1.3 Message code

Message type identifier that determines the command type and schema of the message.

4.1.4 Message data

Variable data (0-1458 bytes based on the current MTU length of 1500 bytes), depending on the Message Code and described in detail in Section 5.

4.1.5 Cryptographic Hash

A 32 byte HMAC-SHA256 hash calculated over the entire message Header and Body (including Start, Sequence Number, Message code, and Message data) using the provisioned 32-byte UDP keys (which expire weekly and are different for broadcast vs. unicast messages) provided to the master by the EMCB's Cloud API and EMCB's by their dedicated agent.

Note – there are different keys required for broadcast messages vs. unicast messages. This is to ensure that a unicast message with a valid Sequence Number, Cryptographic Hash, etc. directed to one Device ID can NOT be replayed unicast to another Device ID that didn't have a chance to see the message and therefore update its synchronized Sequence Number to prevent the replay attack.

4.1.6 Example Message

As an example, a valid message may look like the following (in hex):

```
45 54 4e 4d 00 00 00 04 00 06 ab ba 6b 00 79 03 ea a7 7f 53 81 14 6d
d5 46 29 74 5c 15 b5 b1 08 b5 ad e3 81 2d a5 07 00 86 a7 74 e4
```

This is a message from the Master to the slave with a Sequence Number of 4, message code of 0x0006, message data of 0xabba and a cryptographic hash equal to the underlined contents of the message.

4.2 Data types and Endianness

Data types are abbreviated as follows:

Type symbol	Signed	Length (bytes)
u8	No	1
u16	No	2
u24	No	3
u32	No	4
s32	Yes	4
u64	No	8
s64	Yes	8

ALL number types are little-endian, least-significant-bit first, 2’s complement integers.

4.3 Sequence Number in Detail

The current UDP API does not use any form of encryption. This means that an attacker with knowledge of the protocol and access to the local network can successfully decode any messages that they are able to monitor. One mitigation that is used to verify the authenticity of messages is the Cryptographic Hash – this allows both the master and the EMCBs to validate that at one time, a message originated from an authentic source.

Unfortunately, this alone does not protect against replay attacks. Another potential threat is for an attacker to capture packets on the network and replay them to affect device behavior. The purpose of the Sequence Number is to prevent these message replay attacks. In order to mitigate this threat, each message from the master requires the “next” Sequence Number, with the exception of the *Get Next Expected UDP Sequence Number* which can be used for device discovery and to allow synchronization between a master and slave. A *Set Next Expected UDP Sequence Number* is also available in order to synchronize and allow querying/control over groups of EMCBs, which can all respond to the same broadcast message.

This 32-bit unsigned counter will be initialized to a random number at EMCB boot and count upwards until it overflows back to zero. Values are compared in sequence space arithmetic. For a message to be accepted it must be within the range:

$$\textit{Get Next Expected UDP Sequence Number}\text{ response}$$
$$\leq \text{ Sequence Number } <$$
$$\textit{Get Next Expected UDP Sequence Number}\text{ response} + 100.$$

This ability to "fast forward" the Sequence Number by 100 provides some resiliency to the inevitable dropped UDP packets that will occur in order to keep the Sequence Number of groups of EMCB's synchronized. It also allows unicast packets (which are signed with a different UDP Key than broadcast packets) to share a common Sequence Number with broadcast packets without disturbing the synchronization of a group.

The EMCB does NOT have a secure, non-volatile monotonic counter. In order to preserve security and minimize the chance of replay attacks, the Sequence Number will be initialized to a cryptographically secure random number at every device boot. This means that the Sequence Number will be reset any time the EMCB is power cycled, however, as the EMCB is a line powered device, this should take place very rarely. Whenever a reset happens, the master must use the *Get Next Expected UDP Sequence Number* and *Set Next Expected UDP Sequence Number* commands to resynchronize the device. It is assumed that replays spanning 2^{31}-100 messages are implausible (The UDP key is only valid for a maximum of 1 week. At 25 messages/second, it would be over 284 weeks before messages have a 40ms window to be replayed, assuming the Sequence Number is never reset) and that the master and EMCBs will stay in sync.

The counter can be read or reset to any arbitrary value through both the local API and the EMCB's Cloud API to allow for syncing of groups of EMCBs. On master boot, the *Get Next Expected UDP Sequence Number* command should be sent to the broadcast address on the local network or the counter should be read/initialized through the EMCB's Cloud API. Once the Sequence Number is known, the master can either begin sending messages using this value or choose to update the EMCB's Sequence Number to its last known Sequence Number using the local API. The same process should be followed when 100+ messages go unresponded, as the EMCB could have restarted and reinitialized the Sequence Number.

It is important when using the *Set Next Expected UDP Sequence Number* command to set the Sequence Number to the (most recent value that the master has sent + 1) to minimize the probability of a successful replay attack! For example, implementations should NOT initialize the Sequence Number to a static value such as 0x00000000 on every boot. Alternatively, using a Cryptographically Secure Pseudo-Random Number Generator for the Sequence Number (as the EMCB does at boot) is also acceptable.

The Cloud API should be the preference for the *Set Next Expected UDP Sequence Number* command as this eliminates all threats of replays. However, the local API option is given for use cases where the

Cloud API may not be accessible. Refer to Eaton's example implementation for best practices associated with this protocol.

5 Messages

5.1 Get Next Expected UDP Sequence Number

Example: 6.2.1 Get Next Expected UDP Sequence Number

5.1.1 Message code 0x0000

Get the next expected UDP Sequence Number. This message also includes the Device ID and Protocol Version implemented by this particular device.

NOTE: THIS COMMAND CAN BE USED WITH SEQUENCE NUMBER 0x00000000 IN ORDER TO ALLOW SYNCRONIZATION/DISCOVERY ON THE LAN. In other words, it is susceptible to replays on purpose to facilitate discovery!

NOTE: This message is rate-limited to responding once every 2 seconds in order to prevent abuse. If the rate-limiting is hit, the slave will fail silently.

5.1.2 Command (4 bytes)

Nonce (u32)

- Nonce. 32-bit value provided by the master in order to prove the authenticity of the slave to the master. This value should be changed with every request.

5.1.3 Response (28 bytes)

Sequence # (u32)	Device ID (u8 * 16)	Protocol Version (u32)	Nonce (u32)

- Sequence Number. The expected value of the Sequence Number in the next command
- Device ID. ASCII encoded device ID.
- Protocol Version – An integer value describing revision number of the protocol. For this version, the value will always be **1**. synchronization
- Nonce. The same 32-bit value provided by the master's command. This value should be validated by the master in order to prove the integrity of the response as this message is replayable.

5.2 Get Device Status

Example: 6.2.2 Get Device Status

5.2.1 Message code 0x00FF

Get EMCB device status. See *Get Breaker Remote Handle Position* and *Get Meter Telemetry Data* for more information.

5.2.2 Command (0 bytes)

- No message data

5.2.3 Response (268 bytes)

Breaker State (u8)				
Meter Update Number (u8)	Line frequency (s32)	Period (u16)		
Phase 0 active energy (s64)	Phase 0 reactive energy (s64)	Phase 0 app. energy (u64)	Phase 0 voltage RMS (s32)	Phase 0 current RMS (s32)
Phase 0 active ener. QI (u64)	Phase 0 active ener. QII (u64)	Phase 0 active ener. QIII (u64)	Phase 0 active ener. QIV (u64)	
Phase 0 reactive ener. QI (u64)	Phase 0 reactive ener. QII (u64)	Phase 0 reactive ener. QIII (u64)	Phase 0 reactive ener. QIV (u64)	
Phase 0 app. ener. QI (u64)	Phase 0 app. ener. QII (u64)	Phase 0 app. ener. QIII (u64)	Phase 0 app. ener. QIV (u64)	
Phase 1 active energy (s64)	Phase 1 reactive energy (s64)	Phase 1 app. energy (u64)	Phase 1 voltage RMS (s32)	Phase 1 current RMS (s32)
Phase 1 active ener. QI (u64)	Phase 1 active ener. QII (u64)	Phase 1 active ener. QIII (u64)	Phase 1 active ener. QIV (u64)	
Phase 1 reactive ener. QI (u64)	Phase 1 reactive ener. QII (u64)	Phase 1 reactive ener. QIII (u64)	Phase 1 reactive ener. QIV (u64)	
Phase 1 app. ener. QI (u64)	Phase 1 app. ener. QII (u64)	Phase 1 app. ener. QIII (u64)	Phase 1 app. ener. QIV (u64)	
Phase-phase volt. RMS (s32)				

- Breaker State. The breaker's current Feedback State.
  - Open = 0.
  - Closed = 1.
  - Feedback Mismatch = 2. The last software commanded state does not match the hardware feedback provided by the breaker

- Meter Update Number. Starts at 0 on boot and increments when Meter periodic data is updated on the device.
- Line frequency. mHz.
- Period. The number of milliseconds over which the returned data was accumulated.
- Pole 0 active energy. mJ.
- Pole 0 reactive energy. mVARs.
- Pole 0 apparent energy. mVAs.
- Pole 0 voltage RMS. mV.
- Pole 0 current RMS. mA.
- Pole 0 quadrant values. Active, reactive, apparent cumulative energy per quadrant. Data is in mJ, mVARs and mVAs respectively.
- Pole 1 active energy. mJ.
- Pole 1 reactive energy. mVARs.
- Pole 1 apparent energy. mVAs.
- Pole 1 voltage RMS. mV.
- Pole 1 current RMS. mA.
- Pole 1 quadrant values. Active, reactive, apparent cumulative energy per quadrant. Data is in mJ, mVARs and mVAs respectively.
- Pole-Pole voltage RMS. mV.

5.3 Get Breaker Remote Handle Position

Example: 6.2.3 Get Breaker Remote Handle Position

5.3.1 Message code 0x0100

Get the feedback position of the EMCB's remotely controllable contacts.

5.3.2 Command (0 bytes)

- No message data

5.3.3 Response (1 byte)

State (u8)

- State. The breaker's current Feedback State.
  - Open = 0.
  - Closed = 1.
  - Feedback Mismatch = 2. The last software commanded state does not match the hardware feedback provided by the breaker

5.4 Get Meter Telemetry Data

Example: 6.2.4 Get Meter Telemetry Data

5.4.1 Message code 0x0200

Read periodic meter measurements. After periodic data has been read, subsequent calls will return the same data until the next low-rate period calculation completes.

5.4.2 Command (0 bytes)

- No message data

5.4.3 Response (267 bytes)

Update number (u8)	Line frequency (s32)	Period (u16)		
Phase 0 active energy (s64)	Phase 0 reactive energy (s64)	Phase 0 app. energy (u64)	Phase 0 voltage RMS (s32)	Phase 0 current RMS (s32)
Phase 0 active ener. QI (u64)	Phase 0 active ener. QII (u64)	Phase 0 active ener. QIII (u64)	Phase 0 active ener. QIV (u64)	
Phase 0 reactive ener. QI (u64)	Phase 0 reactive ener. QII (u64)	Phase 0 reactive ener. QIII (u64)	Phase 0 reactive ener. QIV (u64)	
Phase 0 app. ener. QI (u64)	Phase 0 app. ener. QII (u64)	Phase 0 app. ener. QIII (u64)	Phase 0 app. ener. QIV (u64)	
Phase 1 active energy (s64)	Phase 1 reactive energy (s64)	Phase 1 app. energy (u64)	Phase 1 voltage RMS (s32)	Phase 1 current RMS (s32)
Phase 1 active ener. QI (u64)	Phase 1 active ener. QII (u64)	Phase 1 active ener. QIII (u64)	Phase 1 active ener. QIV (u64)	
Phase 1 reactive ener. QI (u64)	Phase 1 reactive ener. QII (u64)	Phase 1 reactive ener. QIII (u64)	Phase 1 reactive ener. QIV (u64)	
Phase 1 app. ener. QI (u64)	Phase 1 app. ener. QII (u64)	Phase 1 app. ener. QIII (u64)	Phase 1 app. ener. QIV (u64)	
Phase-phase volt. RMS (s32)				

- Update Number. Starts at 0 on boot and increments when periodic data is updated on the device.
- Line frequency. mHz.
- Period. The number of milliseconds over which the returned data was accumulated.
- Pole 0 active energy. mJ.
- Pole 0 reactive energy. mVARs.
- Pole 0 apparent energy. mVAs.
- Pole 0 voltage RMS. mV.
- Pole 0 current RMS. mA.

- Pole 0 quadrant values. Active, reactive, apparent cumulative energy per quadrant. Data is in mJ, mVARs and mVAs respectively.
- Pole 1 active energy. mJ.
- Pole 1 reactive energy. mVARs.
- Pole 1 apparent energy. mVAs.
- Pole 1 voltage RMS. mV.
- Pole 1 current RMS. mA.
- Pole 1 quadrant values. Active, reactive, apparent cumulative energy per quadrant. Data is in mJ, mVARs and mVAs respectively.
- Pole-Pole voltage RMS. mV.

5.5 Set Next Expected UDP Sequence Number

Example: 6.2.5 Set Next Expected UDP Sequence Number

5.5.1 Message code 0x8000

Set the next expected UDP Sequence Number.

NOTE: THIS MESSAGE IS RATE-LIMITED TO SUCCEEDING ONCE EVERY 10 SECONDS IN ORDER TO PREVENT ABUSE.

5.5.2 Command (4 bytes)

Sequence # (u32)

- Sequence Number. The value to set on the EMCB to use as the next Sequence Number. **THIS VALUE MUST BE OUTSIDE OF THE SEQUENCE RANGE BELOW IN ORDER TO PREVENT INFINITE REPLAYS.**

$$\textit{Get Next Expected UDP Sequence Number}\text{ response} - 100$$

$$\leq \text{ Sequence Number } <$$

$$\textit{Get Next Expected UDP Sequence Number}\text{ response} + 100.$$

5.5.3 Response (1 byte)

ACK (u8)

- ACK. Positive or negative acknowledge of the command:
  - Acknowledge = 0. Command executed and Next Expected Sequence Number updated
  - Rate Limited = 1. Rate Limiting has prevented the message from succeeding. Try again later.
  - Bad Sequence Number = 2. Sequence Number is in the invalid range

5.6 Set Breaker Remote Handle Position

Example: 6.2.6 Set Breaker Remote Handle Position

5.6.1 Message code 0x8100

Control the EMCB's remotely controllable contacts.

5.6.2 Command (1 byte)

Action (u8)

- Action. The desired state of the remotely controllable contacts:
  - Open = 0.
  - Close = 1.
  - Toggle = 2. The EMCB will read the current position of the remotely controllable contacts and toggle them to the inverse state.

5.6.3 Response (2 byte)

ACK (u8)	State (u8)

- ACK. Positive or negative acknowledge of the command:
  - Acknowledge = 0. Command executed and breaker confirmed it is in desired state.
  - Negative acknowledge = any other value. The command is correct from the Application layer perspective, but some values are invalid or the command cannot be executed.
- State. The breaker's current Feedback State.
  - Open = 0.
  - Closed = 1.
  - Feedback Mismatch = 2. The last software commanded state does not match the hardware feedback provided by the breaker

5.7 Set Bargraph LED to User Defined Color

Example: 6.2.7 Set Bargraph LED to User Defined Color

5.7.1 Message code 0x8300

Control the Bargraph LED Colors. LED 0 is the LED closest to the “bump” / BlinkUp /Network Status LED on the EMCB.

5.7.2 Command (25 bytes)

Enabled (u8)	Duration (s32)		
LED 0 Red (u8)	LED 0 Green (u8)	LED 0 Blue (u8)	LED 0 Blinking (u8)
LED 1 Red (u8)	LED 1 Green (u8)	LED 1 Blue (u8)	LED 1 Blinking (u8)
LED 2 Red (u8)	LED 2 Green (u8)	LED 2 Blue (u8)	LED 2 Blinking (u8)
LED 3 Red (u8)	LED 3 Green (u8)	LED 3 Blue (u8)	LED 3 Blinking (u8)
LED 4 Red (u8)	LED 4 Green (u8)	LED 4 Blue (u8)	LED 4 Blinking (u8)

- Enabled. Determines if the User Defined Color should be shown or if the Bargraph LEDs should resume normal operation.
  - Off = 0. Note, if 0 is passed for Enabled, no other values will be parsed, however, the correct number of bytes must still be sent for Application Layer validation purposes.
  - On = 1.
- Duration. This is the amount of time in seconds that the Breaker should remain the user defined color. If the value is less than or equal to 0, it will remain the user defined color until the next command or restart of the device. The EMCB supports durations of up 10,737,418 seconds — ie. 124 days. Durations larger than this will be rejected, instead, use 0 for “infinite” time.
- LED 0 Red. LED 0 red brightness value.
- LED 0 Green. LED 0 green brightness value.
- LED 0 Blue. LED 0 blue brightness value.
- LED 0 Blinking. Controls if LED 0 blinks on for 0.5 seconds and off for 0.5 seconds or stays on continuously
  - Blinking Off = 0
  - Blinking On = 1
- LED 1 Red. LED 1 red brightness value.
- LED 1 Green. LED 1 green brightness value.

- LED 1 Blue. LED 1 blue brightness value.
- LED 1 Blinking. Controls if LED 1 blinks on for 0.5 seconds and off for 0.5 seconds or stays on continuously
    - Blinking Off = 0
    - Blinking On = 1
- LED 2 Red. LED 2 red brightness value.
- LED 2 Green. LED 2 green brightness value.
- LED 2 Blue. LED 2 blue brightness value.
- LED 2 Blinking. Controls if LED 2 blinks on for 0.5 seconds and off for 0.5 seconds or stays on continuously
    - Blinking Off = 0
    - Blinking On = 1
- LED 3 Red. LED 3 red brightness value.
- LED 3 Green. LED 3 green brightness value.
- LED 3 Blue. LED 3 blue brightness value.
- LED 3 Blinking. Controls if LED 3 blinks on for 0.5 seconds and off for 0.5 seconds or stays on continuously
    - Blinking Off = 0
    - Blinking On = 1
- LED 4 Red. LED 4 red brightness value.
- LED 4 Green. LED 4 green brightness value.
- LED 4 Blue. LED 4 blue brightness value.
- LED 4 Blinking. Controls if LED 4 blinks on for 0.5 seconds and off for 0.5 seconds or stays on continuously
    - Blinking Off = 0
    - Blinking On = 1

5.7.3 Response (1 byte)

ACK (u8)

- ACK. Positive or negative acknowledge of the command:
    - Acknowledge = 0. Command executed and breaker confirmed to be in desired state.
    - Negative acknowledge = any other value. The command is correct from the Application layer perspective, but some values are invalid or the command cannot be executed.

6 Examples

All of the examples below were generated from the emcbUDPmaster node.js module. The following examples were taken on a network with:

- an IP address space of 10.130.116.0
- a subnet mask of 255.255.255.0
- and a broadcast address of 10.130.116.255

All devices were synced using the Cloud API to use UDP Key *DD4253D8725A02A0C1FA3417D809686FE397CC8148EFF5328CE436644849A225* (presented here as hex formatted binary) for broadcast commands.

The following EMCBs were on the network, with their unicast UDP Keys presented as hex formatted binary:

- IP Address 10.130.116.28 = Device ID `30000c2a69113173`
    - UDP Key *590871BFD0D5BE4F4976A9E379B3A0ECDE4613AEC67FD4472686C2242D104D2C*
    - Logs are shown below in `blue`

- IP Address 10.130.115.50 = Device ID `30000c2a690c7652`
    - UDP Key *01C43A38DF5669F3D410602437EC2EF3DAEB12AED3C7EB3FA192D581D2AB9F20*
    - Logs are shown below in `red`

- IP Address 10.130.116.84 = Device ID `30000c2a69112b6f`
    - UDP Key *FCF618AACA30BD359311F475AA7E6B6226CBA53E86681C1416565E8C8B76A663*
    - Logs are shown below in `brown`

- IP Address 10.130.116.85 = Device ID `30000c2a6911283d`
    - UDP Key *E9227B7058B86636F41B7252CAFBFF8E3E64C4DA09A9AC806E7FC8C2D9E696DE*
    - Logs are shown below in `green`

6.1 Common Workflow

6.1.1 Discover Devices

Device discovery consists of sending the replayable *Get Next Expected UDP Sequence Number* to the network's broadcast address. It is advised to do this multiple times in order to ensure that there were no dropped UDP packets during the device discovery process (devices that have responded will ignore *Get Next Expected UDP Sequence Number* for 2 seconds and will not respond to multiple commands within this time window).

Eaton also recommends repeating the Device Discovery process periodically to determine if devices have been added to the network or after a series of consecutive timeouts, in order to resyncronize a device's Sequence Number (in the event of a power outage, network outage, etc.)

Refer to the *Get Next Expected UDP Sequence Number* Example.

6.1.2 Synchronize Device Sequence Numbers

Once devices are discovered, they can be interacted with using their individual Sequence Numbers. For group commands, it is necessary to synchronize Sequence Numbers to a common value as demonstrated in the *Set Next Expected UDP Sequence Number* Example.

6.1.3 Poll Device Status

Most applications will spend the bulk of their lifetime running this command. Applications may poll EMCBs as aggressively as they would like within the constraints of the *Application Layer* timing. An Example of 1 round of polling is demonstrated in the *Get Device Status* Example.

6.1.4 Control Devices

Periodically, it may also be desirable to control groups of EMCBs. This is demonstrated in the *Set Breaker Remote Handle Position* Example

6.2 Individual Command Examples

The following commands contain a mix of broadcast messages with multiple responses and unicast messages with a single response to an individual device with its own Sequence Number. Each of these commands would also work as the opposite type of message (broadcast/unicast), assuming the correct signing UDP Key was used. For broadcast messages, it is important to have synchronized both UDP keys (using the Cloud API) and Sequence Numbers.

6.2.1 Get Next Expected UDP Sequence Number

Reference: 5.1 Get Next Expected UDP Sequence Number

Note – in the following example, we only hear a reply from a single device on the network. Additional *Get Next Expected UDP Sequence Number* commands should be used to find the remaining devices.

```
[2019-03-29T14:48:05.516Z]info: Broadcast Address == 10.130.116.255
[2019-03-29T14:48:05.545Z]verbose: >>>> SENDING >>>>>>GET_NEXT_SEQUENCE_NUMBER ( 46 bytes)  to  10.130.116.255:32866 (BROADCAST)
[2019-03-29T14:48:05.546Z]debug: >>> >>> >>>
"45 54 4E 4D 00 00 00 00 00 00 24 12 69 51 E9 F9 97 C6 14 B6 CF 5C C8 92 54 2E 58 72 09 A0 08 AA 1F 78 F4 36 AC CE 6B 69 B1 E7 59 08
DE 7C"
    >>> Start        = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #   = "00 00 00 00"    = 0x0
    >>> Message Code = "00 00"          = GET_NEXT_SEQUENCE_NUMBER (0x0)
    >>> Message Data = "24 12 69 51" (4 bytes)
    >>> Crypto Hash  = "E9 F9 97 C6 14 B6 CF 5C C8 92 54 2E 58 72 09 A0 08 AA 1F 78 F4 36 AC CE 6B 69 B1 E7 59 08 DE 7C"

[2019-03-29T14:48:05.555Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 70 bytes) from 10.130.116.84:32866
[2019-03-29T14:48:05.555Z]debug: <<< <<< <<<
"45 54 4E 53 00 00 00 00 00 00 D4 B4 DF 9B 34 30 30 30 30 63 32 61 36 39 31 31 32 62 36 66 01 00 00 00 24 12 69 51 8C 4A 7B 28 3E A3
1F EC 22 D2 4A 77 6D 4C DB 31 81 3C BA D2 B6 C7 FF DF EB 5C D9 E0 BB 7C C9 4F"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "00 00 00 00"    = 0x0
    <<< Message Code = "00 00"          = GET_NEXT_SEQUENCE_NUMBER (0x0)
    <<< Message Data = "D4 B4 DF 9B 34 30 30 30 30 63 32 61 36 39 31 31 32 62 36 66 01 00 00 00 24 12 69 51" (28 bytes)
    <<< Crypto Hash  = "8C 4A 7B 28 3E A3 1F EC 22 D2 4A 77 6D 4C DB 31 81 3C BA D2 B6 C7 FF DF EB 5C D9 E0 BB 7C C9 4F"

[2019-03-29T14:48:05.556Z]verbose: <<< HANDLING       GET_NEXT_SEQUENCE_NUMBER ( 70 bytes)  in  10ms
[2019-03-29T14:48:05.560Z]debug: { nextSequenceNumber: 2615129300,
  idDevice: '40000c2a69112b6f',
  protocolRevision: 1 }
[2019-03-29T14:48:05.560Z]verbose: Creating EmcbUDPdeviceMaster Instance for 10.130.116.84
[2019-03-29T14:48:05.562Z]debug: Setting instance sequence number to 0x9BDFB4D4 = 2615129300
```

6.2.2 Get Device Status

Reference: 5.2 Get Device Status

```
[2019-03-15T18:16:32.604Z]verbose: >>>> SENDING >>>>>>>>>>>>>>GET_DEVICE_STATUS ( 42 bytes)  to  10.130.116.255:32866 (BROADCAST)
[2019-03-15T18:16:32.605Z]debug: >>> >>> >>>
"45 54 4E 4D 61 61 B3 7E FF 00 6B 60 53 E4 F4 52 F6 28 09 FE D6 AD 8A 65 F4 EB DA BD 11 EB A6 80 79 7D 45 A7 E3 6D 76 46 C7 F5"
    >>> Start        = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #   = "61 61 B3 7E"    = 0x7EB36161
    >>> Message Code = "FF 00"          = GET_DEVICE_STATUS (0xFF)
    >>> Message Data = ""
    >>> Crypto Hash  = "6B 60 53 E4 F4 52 F6 28 09 FE D6 AD 8A 65 F4 EB DA BD 11 EB A6 80 79 7D 45 A7 E3 6D 76 46 C7 F5"

[2019-03-15T18:16:32.634Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (310 bytes) from 10.130.116.50:32866 [30000c2a690c7652]
[2019-03-15T18:16:32.634Z]debug: <<< <<< <<<
"45 54 4E 53 61 61 B3 7E FF 00 01 9D 00 00 00 00 C8 00 E0 FB 33 C9 FF FF FF FF 8D 01 6C FB FF FF FF FF 11 E5 89 37 00 00 00 00 00 00
00 00 0D 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 1E 04 CC 36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 02 00 00 00 00 00 00 00 75 FE 93 04 00 00 00 00 00 00 00 00 00 00 00 00 6A 75 1B 00 00 00 00 00 39 2E 29 00 00 00 00 00 74 E5 36 37
00 00 00 00 FA 5B 0E 00 00 00 00 00 FE FF FF FF FF FF FF FF AE FB FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 52 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 7B 00 00 00 F6 4D 1C 1E 6D 4A 63 6D 26 F5 53 74 50 8D 81 C9 42 BE 88 07 3F 9B CC BB 6A 38 FB AD 02 95 E3 A5"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "61 61 B3 7E"    = 0x7EB36161
    <<< Message Code = "FF 00"          = GET_DEVICE_STATUS (0xFF)
    <<< Message Data = "01 9D 00 00 00 00 C8 00 E0 FB 33 C9 FF FF FF FF 8D 01 6C FB FF FF FF FF 11 E5 89 37 00 00 00 00 00 00 00 00 0D
00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 1E 04 CC 36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00
00 00 00 00 00 00 75 FE 93 04 00 00 00 00 00 00 00 00 00 00 00 00 6A 75 1B 00 00 00 00 00 39 2E 29 00 00 00 00 00 74 E5 36 37 00 00 00
00 FA 5B 0E 00 00 00 00 00 FE FF FF FF FF FF FF FF AE FB FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 52
04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 7B 00 00 00"
    <<< Crypto Hash  = "F6 4D 1C 1E 6D 4A 63 6D 26 F5 53 74 50 8D 81 C9 42 BE 88 07 3F 9B CC BB 6A 38 FB AD 02 95 E3 A5"

[2019-03-15T18:16:32.635Z]verbose: <<< HANDLING              GET_DEVICE_STATUS (310 bytes)  in  30ms
[2019-03-15T18:16:32.636Z]debug: {
          "breaker": {
                    "state": 1,
                    "stateString": "Closed"
          },
          "meter": {
                    "sequence": 157,
                    "frequency": 0,
                    "period": 200,
                    "mJp0": 0,
                    "mVARsp0": 0,
                    "mVAsp0": "931783953",
                    "LNmVp0": 0,
                    "mAp0": 13,
                    "q1mJp0": "0",
                    "q2mJp0": "2",
                    "q3mJp0": "919340062",
                    "q4mJp0": "0",
                    "q1mVARsp0": "0",
                    "q2mVARsp0": "2",
                    "q3mVARsp0": "76807797",
                    "q4mVARsp0": "0",
                    "q1mVAsp0": "1799530",
                    "q2mVAsp0": "2698809",
                    "q3mVAsp0": "926344564",
                    "q4mVAsp0": "941050",
                    "mJp1": 0,
                    "mVARsp1": 0,
                    "mVAsp1": "0",
                    "LNmVp1": 0,
                    "mAp1": 8,
                    "q1mJp1": "0",
                    "q2mJp1": "0",
                    "q3mJp1": "2",
                    "q4mJp1": "0",
                    "q1mVARsp1": "0",
                    "q2mVARsp1": "0",
                    "q3mVARsp1": "1106",
                    "q4mVARsp1": "0",
                    "q1mVAsp1": "0",
                    "q2mVAsp1": "0",
                    "q3mVAsp1": "0",
                    "q4mVAsp1": "0",
                    "LLp01mV": 123
          }
}
[2019-03-15T18:16:32.636Z]info: Breaker Feedback Position changed from undefined to 1
[2019-03-15T18:16:32.636Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (310 bytes) from 10.130.116.28:32866 [40000c2a69113173]
[2019-03-15T18:16:32.636Z]debug: <<< <<< <<<
"45 54 4E 53 61 61 B3 7E FF 00 01 E4 60 EA 00 00 C8 00 F7 E7 3B EF F5 FF FF FF 4F 68 7D EA FF FF FF FF 47 44 C3 3C 0A 00 00 00 5B E7
01 00 29 00 00 00 91 2D 00 00 00 00 00 00 11 D4 45 08 00 00 00 00 5D 94 7E 08 0A 00 00 00 D4 22 00 00 00 00 00 00 52 C0 01 00 00 00 00
00 2A AF 04 01 00 00 00 00 B1 FF 87 16 00 00 00 00 7C 07 01 00 00 00 00 00 93 4B 06 00 00 00 00 00 EB AD 4E 0E 00 00 00 00 3B 06 6A 2E
0A 00 00 00 8E 44 04 00 00 00 00 00 AD 89 4C 50 F3 FF FF FF 60 18 76 1D 00 00 00 00 23 55 4F 05 0D 00 00 00 5B E7 01 00 C1 04 00 00 50
0C 00 00 00 00 00 00 E0 BA 9E AF 0C 00 00 00 36 D0 14 00 00 00 00 00 73 08 00 00 00 00 00 00 45 D4 00 00 00 00 00 00 53 30 76 1D 00 00
00 00 47 81 00 00 00 00 00 00 F1 6A 00 00 00 00 00 00 BB 32 03 00 00 00 00 00 06 CD 71 00 0D 00 00 00 18 C7 D7 04 00 00 00 00 4A 8E 02
00 00 00 00 00 72 00 00 00 79 B5 24 04 E5 12 F3 63 D1 66 B9 FC 1E FF D2 A1 6B FD B5 CB C2 1B 7C 48 6E CC E1 83 5E DB CB DF"
    <<< Start          = "45 54 4E 53"    = "ETNS"
    <<< Sequence #     = "61 61 B3 7E"    = 0x7EB36161
    <<< Message Code = "FF 00"            = GET_DEVICE_STATUS (0xFF)
    <<< Message Data = "01 E4 60 EA 00 00 C8 00 F7 E7 3B EF F5 FF FF FF 4F 68 7D EA FF FF FF FF 47 44 C3 3C 0A 00 00 00 5B E7 01 00 29
00 00 00 91 2D 00 00 00 00 00 00 11 D4 45 08 00 00 00 00 5D 94 7E 08 0A 00 00 00 D4 22 00 00 00 00 00 00 52 C0 01 00 00 00 00 00 2A AF
04 01 00 00 00 00 B1 FF 87 16 00 00 00 00 7C 07 01 00 00 00 00 00 93 4B 06 00 00 00 00 00 EB AD 4E 0E 00 00 00 00 3B 06 6A 2E 0A 00 00
00 8E 44 04 00 00 00 00 00 AD 89 4C 50 F3 FF FF FF 60 18 76 1D 00 00 00 00 23 55 4F 05 0D 00 00 00 5B E7 01 00 C1 04 00 00 50 0C 00 00
00 00 00 00 E0 BA 9E AF 0C 00 00 00 36 D0 14 00 00 00 00 00 73 08 00 00 00 00 00 00 45 D4 00 00 00 00 00 00 53 30 76 1D 00 00 00 00 47
81 00 00 00 00 00 00 F1 6A 00 00 00 00 00 00 BB 32 03 00 00 00 00 00 06 CD 71 00 0D 00 00 00 18 C7 D7 04 00 00 00 00 4A 8E 02 00 00 00
00 00 72 00 00 00"
    <<< Crypto Hash  = "79 B5 24 04 E5 12 F3 63 D1 66 B9 FC 1E FF D2 A1 6B FD B5 CB C2 1B 7C 48 6E CC E1 83 5E DB CB DF"

[2019-03-15T18:16:32.637Z]verbose: <<< HANDLING              GET_DEVICE_STATUS (310 bytes)  in  32ms
[2019-03-15T18:16:32.637Z]debug: {
         "breaker": {
                  "state": 1,
                  "stateString": "Closed"
         },
         "meter": {
                  "sequence": 228,
                  "frequency": 60000,
                  "period": 200,
                  "mJp0": 0,
                  "mVARsp0": 0,
                  "mVAsp0": "43969102919",
                  "LNmVp0": 124763,
                  "mAp0": 41,
                  "q1mJp0": "11665",
                  "q2mJp0": "138794001",
                  "q3mJp0": "43092186205",
                  "q4mJp0": "8916",
                  "q1mVARsp0": "114770",
                  "q2mVARsp0": "17084202",
                  "q3mVARsp0": "378011569",
                  "q4mVARsp0": "67452",
                  "q1mVAsp0": "412563",
                  "q2mVAsp0": "240037355",
                  "q3mVAsp0": "43728373307",
                  "q4mVAsp0": "279694",
                  "mJp1": 0,
                  "mVARsp1": 0,
                  "mVAsp1": "55923660067",
                  "LNmVp1": 124763,
                  "mAp1": 1217,
                  "q1mJp1": "3152",
                  "q2mJp1": "54486022880",
                  "q3mJp1": "1364022",
                  "q4mJp1": "2163",
                  "q1mVARsp1": "54341",
                  "q2mVARsp1": "494284883",
                  "q3mVARsp1": "33095",
                  "q4mVARsp1": "27377",
                   "q1mVAsp1": "209595",
                   "q2mVAsp1": "55842032902",
                   "q3mVAsp1": "81250072",
                   "q4mVAsp1": "167498",
                   "LLp01mV": 114
         }
}
[2019-03-15T18:16:32.637Z]info: Breaker Feedback Position changed from undefined to 1
[2019-03-15T18:16:32.637Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (310 bytes) from 10.130.116.84:32866 [40000c2a69112b6f]
[2019-03-15T18:16:32.638Z]debug: <<< <<< <<<
"45 54 4E 53 61 61 B3 7E FF 00 01 F0 6F EA 00 00 C8 00 E6 54 8E FE FF FF FF FF 6D 6F EC FF FF FF FF FF BE 59 AB 01 00 00 00 00 65 E7
01 00 09 00 00 00 1A 0C 00 00 00 00 00 00 17 58 00 00 00 00 00 00 F2 6C 71 01 00 00 00 00 D5 0D 00 00 00 00 00 00 02 C0 02 00 00 00 00
00 A3 60 05 00 00 00 00 00 1A 6B 10 00 00 00 00 00 1E 46 0B 00 00 00 00 00 66 76 07 00 00 00 00 00 8C F0 0D 00 00 00 00 00 6D 3E 80 01
00 00 00 00 5F B4 15 00 00 00 00 00 0D 8B E4 9D DB FF FF FF 3A F5 34 CD FF FF FF FF 19 30 29 6C 24 00 00 00 65 E7 01 00 FC 04 00 00 31
02 00 00 00 00 00 00 8E F1 3C 06 0D 00 00 00 3A 96 DE 5B 17 00 00 00 A4 10 00 00 00 00 00 00 8C 73 00 00 00 00 00 00 9E BF 55 00 00 00
00 00 2C DD 12 33 00 00 00 00 C4 60 0E 00 00 00 00 00 CC A2 01 00 00 00 00 00 F0 08 84 02 0D 00 00 00 AE D8 8C 69 17 00 00 00 AF AB 16
00 00 00 00 00 72 00 00 00 45 BF FE B0 30 2F 2B 90 67 16 97 56 89 BB BB EA 71 DF 6A 26 33 CC 64 F8 1C D6 1C F2 FF F7 B6 D1"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "61 61 B3 7E"    = 0x7EB36161
    <<< Message Code = "FF 00"          = GET_DEVICE_STATUS (0xFF)
    <<< Message Data = "01 F0 6F EA 00 00 C8 00 E6 54 8E FE FF FF FF FF 6D 6F EC FF FF FF FF FF BE 59 AB 01 00 00 00 00 65 E7 01 00 09
00 00 00 1A 0C 00 00 00 00 00 00 17 58 00 00 00 00 00 00 F2 6C 71 01 00 00 00 00 D5 0D 00 00 00 00 00 00 02 C0 02 00 00 00 00 00 A3 60
05 00 00 00 00 00 1A 6B 10 00 00 00 00 00 1E 46 0B 00 00 00 00 00 66 76 07 00 00 00 00 00 8C F0 0D 00 00 00 00 00 6D 3E 80 01 00 00 00
00 5F B4 15 00 00 00 00 00 0D 8B E4 9D DB FF FF FF 3A F5 34 CD FF FF FF FF 19 30 29 6C 24 00 00 00 65 E7 01 00 FC 04 00 00 31 02 00 00
00 00 00 00 8E F1 3C 06 0D 00 00 00 3A 96 DE 5B 17 00 00 00 A4 10 00 00 00 00 00 00 8C 73 00 00 00 00 00 00 9E BF 55 00 00 00 00 00 2C
DD 12 33 00 00 00 00 C4 60 0E 00 00 00 00 00 CC A2 01 00 00 00 00 00 F0 08 84 02 0D 00 00 00 AE D8 8C 69 17 00 00 00 AF AB 16 00 00 00
00 00 72 00 00 00"
    <<< Crypto Hash  = "45 BF FE B0 30 2F 2B 90 67 16 97 56 89 BB BB EA 71 DF 6A 26 33 CC 64 F8 1C D6 1C F2 FF F7 B6 D1"

[2019-03-15T18:16:32.638Z]verbose: <<< HANDLING              GET_DEVICE_STATUS (310 bytes)  in  33ms
[2019-03-15T18:16:32.638Z]debug: {
         "breaker": {
                   "state": 1,
                   "stateString": "Closed"
         },
         "meter": {
                   "sequence": 240,
                   "frequency": 60015,
                   "period": 200,
                   "mJp0": 0,
                   "mVARsp0": 0,
                   "mVAsp0": "28006846",
                   "LNmVp0": 124773,
                   "mAp0": 9,
                   "q1mJp0": "3098",
                   "q2mJp0": "22551",
                   "q3mJp0": "24210674",
                   "q4mJp0": "3541",
                   "q1mVARsp0": "180226",
                   "q2mVARsp0": "352419",
                   "q3mVARsp0": "1075994",
                   "q4mVARsp0": "738846",
                   "q1mVAsp0": "489062",
                   "q2mVAsp0": "913548",
                   "q3mVAsp0": "25181805",
                   "q4mVAsp0": "1422431",
                   "mJp1": 0,
                   "mVARsp1": 0,
                   "mVAsp1": "156433461273",
                   "LNmVp1": 124773,
                   "mAp1": 1276,
                   "q1mJp1": "561",
                   "q2mJp1": "55939232142",
                   "q3mJp1": "100325561914",
                   "q4mJp1": "4260",
                   "q1mVARsp1": "29580",
                   "q2mVARsp1": "5619614",
                   "q3mVARsp1": "856874284",
                   "q4mVARsp1": "942276",
                   "q1mVAsp1": "107212",
                   "q2mVAsp1": "55876782320",
                   "q3mVAsp1": "100555085998",
                   "q4mVAsp1": "1485743",
                   "LLp01mV": 114
          }
}
[2019-03-15T18:16:32.638Z]info: Breaker Feedback Position changed from undefined to 1
[2019-03-15T18:16:32.639Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (310 bytes) from 10.130.116.85:32866 [40000c2a6911283d]
[2019-03-15T18:16:32.639Z]debug: <<< <<< <<<
"45 54 4E 53 61 61 B3 7E FF 00 01 F4 60 EA 00 00 C8 00 08 13 87 E0 FE FF FF FF 85 78 FD 50 00 00 00 00 CB DD 25 67 01 00 00 00 F2 E7
01 00 09 00 00 00 00 00 00 00 00 00 00 34 E2 78 1F 01 00 00 00 C4 0A 00 00 00 00 00 00 00 00 00 00 00 00 17 0B 00 00 00 00 00
00 6E 6D FD 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 34 00 00 00 00 00 00 E8 25 25 67 01 00 00 00 88 6D 00 00
00 00 00 00 19 16 00 00 00 00 00 00 50 4C E2 95 F5 FF FF FF 70 B1 63 FC FF FF FF FF 03 AF C8 79 0A 00 00 00 F2 E7 01 00 08 00 00 00 00
00 00 00 00 00 00 00 53 75 D3 53 00 00 00 00 5D 3E 4A 16 0A 00 00 00 00 00 00 00 00 00 00 35 3E 02 00 00 00 00 00 04 9E B2 13 00 00
00 00 BA 24 51 17 00 00 00 00 0F 06 00 00 00 00 00 00 E8 84 03 00 00 00 00 00 15 53 29 5D 00 00 00 00 A9 AA 99 1C 0A 00 00 00 5D 2C 02
00 00 00 00 00 72 00 00 00 FC FA 69 52 6E 0B 5F F0 E7 B1 C6 8C 2F DB AA 69 62 7C 30 98 6D 61 4F 8E F2 52 AC E0 23 25 49 3C"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "61 61 B3 7E"    = 0x7EB36161
    <<< Message Code = "FF 00"          = GET_DEVICE_STATUS (0xFF)
    <<< Message Data = "01 F4 60 EA 00 00 C8 00 08 13 87 E0 FE FF FF FF 85 78 FD 50 00 00 00 00 CB DD 25 67 01 00 00 00 F2 E7 01 00 09
00 00 00 00 00 00 00 00 00 00 00 34 E2 78 1F 01 00 00 00 C4 0A 00 00 00 00 00 00 00 00 00 00 00 00 17 0B 00 00 00 00 00 00 6E 6D
FD 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 34 00 00 00 00 00 00 E8 25 25 67 01 00 00 00 88 6D 00 00 00 00 00
00 19 16 00 00 00 00 00 00 50 4C E2 95 F5 FF FF FF 70 B1 63 FC FF FF FF FF 03 AF C8 79 0A 00 00 00 F2 E7 01 00 08 00 00 00 00 00 00
00 00 00 00 53 75 D3 53 00 00 00 00 5D 3E 4A 16 0A 00 00 00 00 00 00 00 00 00 00 35 3E 02 00 00 00 00 00 04 9E B2 13 00 00 00 00 BA
24 51 17 00 00 00 00 0F 06 00 00 00 00 00 00 E8 84 03 00 00 00 00 00 15 53 29 5D 00 00 00 00 A9 AA 99 1C 0A 00 00 00 5D 2C 02 00 00 00
00 00 72 00 00 00"
    <<< Crypto Hash  = "FC FA 69 52 6E 0B 5F F0 E7 B1 C6 8C 2F DB AA 69 62 7C 30 98 6D 61 4F 8E F2 52 AC E0 23 25 49 3C"
```

```
[2019-03-15T18:16:32.639Z]verbose: <<< HANDLING              GET_DEVICE_STATUS (310 bytes)  in  34ms
[2019-03-15T18:16:32.639Z]debug: {
         "breaker": {
                  "state": 1,
                  "stateString": "Closed"
         },
         "meter": {
                  "sequence": 244,
                  "frequency": 60000,
                  "period": 200,
                  "mJp0": 0,
                  "mVARsp0": 0,
                  "mVAsp0": "6025502155",
                  "LNmVp0": 124914,
                  "mAp0": 9,
                  "q1mJp0": "0",
                  "q2mJp0": "4822983220",
                  "q3mJp0": "2756",
                  "q4mJp0": "0",
                  "q1mVARsp0": "2839",
                  "q2mVARsp0": "1358785902",
                  "q3mVARsp0": "0",
                  "q4mVARsp0": "0",
                  "q1mVAsp0": "13378",
                  "q2mVAsp0": "6025455080",
                  "q3mVAsp0": "28040",
                  "q4mVAsp0": "5657",
                  "mJp1": 0,
                  "mVARsp1": 0,
                  "mVAsp1": "44992868099",
                  "LNmVp1": 124914,
                  "mAp1": 8,
                  "q1mJp1": "0",
                  "q2mJp1": "1406367059",
                  "q3mJp1": "43323637341",
                  "q4mJp1": "0",
                  "q1mVARsp1": "146997",
                  "q2mVARsp1": "330472964",
                  "q3mVARsp1": "391193786",
                  "q4mVARsp1": "1551",
                  "q1mVAsp1": "230632",
                  "q2mVAsp1": "1562989333",
                  "q3mVAsp1": "43429505705",
                  "q4mVAsp1": "142429",
                  "LLp01mV": 114
         }
}
[2019-03-15T18:16:32.639Z]info: Breaker Feedback Position changed from undefined to 1
```

6.2.3 Get Breaker Remote Handle Position

Reference: 5.3 Get Breaker Remote Handle Position

```
[2019-03-15T18:16:36.485Z]verbose: >>>> SENDING GET_BREAKER_REMOTE_HANDLE_POSITION ( 42 bytes)  to  10.130.116.255:32866 (BROADCAST)
[2019-03-15T18:16:36.485Z]debug: >>> >>> >>>
"45 54 4E 4D 62 61 B3 7E 00 01 91 54 BD 23 F3 E4 7C CE CE 04 E1 AE BD 00 33 20 45 D3 C3 84 B5 7A 79 41 FB 38 80 8F ED 89 C0 07"
    >>> Start        = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #   = "62 61 B3 7E"    = 0x7EB36162
    >>> Message Code = "00 01"          = GET_BREAKER_REMOTE_HANDLE_POSITION (0x100)
    >>> Message Data = ""
    >>> Crypto Hash  = "91 54 BD 23 F3 E4 7C CE CE 04 E1 AE BD 00 33 20 45 D3 C3 84 B5 7A 79 41 FB 38 80 8F ED 89 C0 07"

[2019-03-15T18:16:36.490Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.50:32866 [30000c2a690c7652]
[2019-03-15T18:16:36.490Z]debug: <<< <<< <<<
"45 54 4E 53 62 61 B3 7E 00 01 01 CA 18 DC B7 ED F6 7E 10 E4 0A 54 07 89 77 C8 40 AB D7 73 A1 45 55 D2 1D 41 03 5A 8E 57 90 AF 0A"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "62 61 B3 7E"    = 0x7EB36162
    <<< Message Code = "00 01"          = GET_BREAKER_REMOTE_HANDLE_POSITION (0x100)
    <<< Message Data = "01"
    <<< Crypto Hash  = "CA 18 DC B7 ED F6 7E 10 E4 0A 54 07 89 77 C8 40 AB D7 73 A1 45 55 D2 1D 41 03 5A 8E 57 90 AF 0A"

[2019-03-15T18:16:36.490Z]verbose: <<< HANDLING GET_BREAKER_REMOTE_HANDLE_POSITION ( 43 bytes)  in  5ms
[2019-03-15T18:16:36.491Z]debug: {
         "state": 1,
         "stateString": "Closed"
}
[2019-03-15T18:16:36.491Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.85:32866 [40000c2a6911283d]
[2019-03-15T18:16:36.491Z]debug: <<< <<< <<<
"45 54 4E 53 62 61 B3 7E 00 01 01 CA 18 DC B7 ED F6 7E 10 E4 0A 54 07 89 77 C8 40 AB D7 73 A1 45 55 D2 1D 41 03 5A 8E 57 90 AF 0A"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "62 61 B3 7E"    = 0x7EB36162
    <<< Message Code = "00 01"          = GET_BREAKER_REMOTE_HANDLE_POSITION (0x100)
    <<< Message Data = "01"
    <<< Crypto Hash  = "CA 18 DC B7 ED F6 7E 10 E4 0A 54 07 89 77 C8 40 AB D7 73 A1 45 55 D2 1D 41 03 5A 8E 57 90 AF 0A"

[2019-03-15T18:16:36.492Z]verbose: <<< HANDLING GET_BREAKER_REMOTE_HANDLE_POSITION ( 43 bytes)  in  6ms
[2019-03-15T18:16:36.492Z]debug: {
         "state": 1,
         "stateString": "Closed"
}
[2019-03-15T18:16:36.492Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.84:32866 [40000c2a69112b6f]
[2019-03-15T18:16:36.492Z]debug: <<< <<< <<<
"45 54 4E 53 62 61 B3 7E 00 01 01 CA 18 DC B7 ED F6 7E 10 E4 0A 54 07 89 77 C8 40 AB D7 73 A1 45 55 D2 1D 41 03 5A 8E 57 90 AF 0A"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "62 61 B3 7E"    = 0x7EB36162
    <<< Message Code = "00 01"          = GET_BREAKER_REMOTE_HANDLE_POSITION (0x100)
    <<< Message Data = "01"
    <<< Crypto Hash  = "CA 18 DC B7 ED F6 7E 10 E4 0A 54 07 89 77 C8 40 AB D7 73 A1 45 55 D2 1D 41 03 5A 8E 57 90 AF 0A"

[2019-03-15T18:16:36.492Z]verbose: <<< HANDLING GET_BREAKER_REMOTE_HANDLE_POSITION ( 43 bytes)  in  7ms
[2019-03-15T18:16:36.492Z]debug: {
         "state": 1,
         "stateString": "Closed"
}
[2019-03-15T18:16:36.521Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.28:32866 [40000c2a69113173]
[2019-03-15T18:16:36.521Z]debug: <<< <<< <<<
"45 54 4E 53 62 61 B3 7E 00 01 01 CA 18 DC B7 ED F6 7E 10 E4 0A 54 07 89 77 C8 40 AB D7 73 A1 45 55 D2 1D 41 03 5A 8E 57 90 AF 0A"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "62 61 B3 7E"    = 0x7EB36162
    <<< Message Code = "00 01"          = GET_BREAKER_REMOTE_HANDLE_POSITION (0x100)
    <<< Message Data = "01"
    <<< Crypto Hash  = "CA 18 DC B7 ED F6 7E 10 E4 0A 54 07 89 77 C8 40 AB D7 73 A1 45 55 D2 1D 41 03 5A 8E 57 90 AF 0A"

[2019-03-15T18:16:36.521Z]verbose: <<< HANDLING GET_BREAKER_REMOTE_HANDLE_POSITION ( 43 bytes)  in  36ms
[2019-03-15T18:16:36.521Z]debug: {
         "state": 1,
         "stateString": "Closed"
}
```

6.2.4 Get Meter Telemetry Data

Reference: 5.4 Get Meter Telemetry Data

```
[2019-03-15T18:16:38.908Z]verbose: >>>> SENDING >>>>>>GET_METER_TELEMETRY_DATA ( 42 bytes)  to  10.130.116.255:32866 (BROADCAST)
[2019-03-15T18:16:38.908Z]debug: >>> >>> >>>
"45 54 4E 4D 63 61 B3 7E 00 02 84 AC BA 4E 51 4E 4C 19 34 7B 1F CF 75 69 31 75 3C 21 61 A6 A6 35 16 E8 61 41 CA 76 85 BA F5 C1"
    >>> Start        = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #   = "63 61 B3 7E"    = 0x7EB36163
    >>> Message Code = "00 02"          = GET_METER_TELEMETRY_DATA (0x200)
    >>> Message Data = ""
    >>> Crypto Hash  = "84 AC BA 4E 51 4E 4C 19 34 7B 1F CF 75 69 31 75 3C 21 61 A6 A6 35 16 E8 61 41 CA 76 85 BA F5 C1"

[2019-03-15T18:16:38.938Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (309 bytes) from 10.130.116.50:32866 [30000c2a690c7652]
[2019-03-15T18:16:38.938Z]debug: <<< <<< <<<
"45 54 4E 53 63 61 B3 7E 00 02 BC 00 00 00 00 C8 00 E0 FB 33 C9 FF FF FF FF 8D 01 6C FB FF FF FF FF 11 E5 89 37 00 00 00 00 00 00 00
00 0F 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 1E 04 CC 36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02 00 00 00 00 00 00 00 75 FE 93 04 00 00 00 00 00 00 00 00 00 00 00 00 6A 75 1B 00 00 00 00 00 39 2E 29 00 00 00 00 00 74 E5 36 37 00
00 00 00 FA 5B 0E 00 00 00 00 00 FE FF FF FF FF FF FF FF AE FB FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 52 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 7A 00 00 00 1F F8 25 BE BD AE 72 C3 49 31 71 65 5C EE E5 53 68 BF 16 F3 09 CA 20 DE 5B 7A 85 28 4F 81 B2 80"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "63 61 B3 7E"    = 0x7EB36163
    <<< Message Code = "00 02"          = GET_METER_TELEMETRY_DATA (0x200)
    <<< Message Data = "BC 00 00 00 00 C8 00 E0 FB 33 C9 FF FF FF FF 8D 01 6C FB FF FF FF FF 11 E5 89 37 00 00 00 00 00 00 00 00 0F 00
00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 1E 04 CC 36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00
00 00 00 00 00 75 FE 93 04 00 00 00 00 00 00 00 00 00 00 00 00 6A 75 1B 00 00 00 00 00 39 2E 29 00 00 00 00 00 74 E5 36 37 00 00 00 00
FA 5B 0E 00 00 00 00 00 FE FF FF FF FF FF FF FF AE FB FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 52 04
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7A 00 00 00"
    <<< Crypto Hash  = "1F F8 25 BE BD AE 72 C3 49 31 71 65 5C EE E5 53 68 BF 16 F3 09 CA 20 DE 5B 7A 85 28 4F 81 B2 80"

[2019-03-15T18:16:38.938Z]verbose: <<< HANDLING       GET_METER_TELEMETRY_DATA (309 bytes)  in  30ms
[2019-03-15T18:16:38.939Z]debug: {
         "sequence": 188,
         "frequency": 0,
         "period": 200,
         "mJp0": 0,
         "mVARsp0": 0,
         "mVAsp0": "931783953",
         "LNmVp0": 0,
         "mAp0": 15,
         "q1mJp0": "0",
         "q2mJp0": "2",
         "q3mJp0": "919340062",
         "q4mJp0": "0",
          "q1mVARsp0": "0",
          "q2mVARsp0": "2",
          "q3mVARsp0": "76807797",
          "q4mVARsp0": "0",
          "q1mVAsp0": "1799530",
          "q2mVAsp0": "2698809",
          "q3mVAsp0": "926344564",
          "q4mVAsp0": "941050",
          "mJp1": 0,
          "mVARsp1": 0,
          "mVAsp1": "0",
          "LNmVp1": 0,
          "mAp1": 3,
          "q1mJp1": "0",
          "q2mJp1": "0",
          "q3mJp1": "2",
          "q4mJp1": "0",
          "q1mVARsp1": "0",
          "q2mVARsp1": "0",
          "q3mVARsp1": "1106",
          "q4mVARsp1": "0",
          "q1mVAsp1": "0",
          "q2mVAsp1": "0",
          "q3mVAsp1": "0",
          "q4mVAsp1": "0",
          "LLp01mV": 122
}
[2019-03-15T18:16:38.939Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (309 bytes) from 10.130.116.28:32866 [40000c2a69113173]
[2019-03-15T18:16:38.939Z]debug: <<< <<< <<<
"45 54 4E 53 63 61 B3 7E 00 02 03 6F EA 00 00 C8 00 F7 E7 3B EF F5 FF FF FF 4F 68 7D EA FF FF FF FF 3F BE C3 3C 0A 00 00 00 F7 E6 01
00 28 00 00 00 91 2D 00 00 00 00 00 00 11 D4 45 08 00 00 00 00 5D 94 7E 08 0A 00 00 00 D4 22 00 00 00 00 00 00 52 C0 01 00 00 00 00 00
2A AF 04 01 00 00 00 00 B1 FF 87 16 00 00 00 00 7C 07 01 00 00 00 00 00 93 4B 06 00 00 00 00 00 EB AD 4E 0E 00 00 00 00 33 80 6A 2E 0A
00 00 00 8E 44 04 00 00 00 00 00 FD 25 3E 50 F3 FF FF FF 60 18 76 1D 00 00 00 00 8F B3 5D 05 0D 00 00 00 F7 E6 01 00 C1 04 00 00 50 0C
00 00 00 00 00 00 90 1E AD AF 0C 00 00 00 36 D0 14 00 00 00 00 00 73 08 00 00 00 00 00 00 45 D4 00 00 00 00 00 00 53 30 76 1D 00 00 00
00 47 81 00 00 00 00 00 00 F1 6A 00 00 00 00 00 00 BB 32 03 00 00 00 00 00 72 2B 80 00 0D 00 00 00 18 C7 D7 04 00 00 00 00 4A 8E 02 00
00 00 00 00 72 00 00 00 21 18 36 92 71 B7 18 E4 29 BC D7 5D F4 C9 FC 25 1F 64 B7 D0 75 C2 E8 90 43 E0 F3 DC D5 BB 70 4F"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "63 61 B3 7E"    = 0x7EB36163
    <<< Message Code = "00 02"          = GET_METER_TELEMETRY_DATA (0x200)
    <<< Message Data = "03 6F EA 00 00 C8 00 F7 E7 3B EF F5 FF FF FF 4F 68 7D EA FF FF FF FF 3F BE C3 3C 0A 00 00 00 F7 E6 01 00 28 00
00 00 91 2D 00 00 00 00 00 00 11 D4 45 08 00 00 00 00 5D 94 7E 08 0A 00 00 00 D4 22 00 00 00 00 00 00 52 C0 01 00 00 00 00 00 2A AF 04
01 00 00 00 00 B1 FF 87 16 00 00 00 00 7C 07 01 00 00 00 00 00 93 4B 06 00 00 00 00 00 EB AD 4E 0E 00 00 00 00 33 80 6A 2E 0A 00 00 00
8E 44 04 00 00 00 00 00 FD 25 3E 50 F3 FF FF FF 60 18 76 1D 00 00 00 00 8F B3 5D 05 0D 00 00 00 F7 E6 01 00 C1 04 00 00 50 0C 00 00 00
00 00 00 90 1E AD AF 0C 00 00 00 36 D0 14 00 00 00 00 00 73 08 00 00 00 00 00 00 45 D4 00 00 00 00 00 00 53 30 76 1D 00 00 00 00 47 81
00 00 00 00 00 00 F1 6A 00 00 00 00 00 00 BB 32 03 00 00 00 00 00 72 2B 80 00 0D 00 00 00 18 C7 D7 04 00 00 00 00 4A 8E 02 00 00 00 00
00 72 00 00 00"
    <<< Crypto Hash  = "21 18 36 92 71 B7 18 E4 29 BC D7 5D F4 C9 FC 25 1F 64 B7 D0 75 C2 E8 90 43 E0 F3 DC D5 BB 70 4F"

[2019-03-15T18:16:38.940Z]verbose: <<< HANDLING       GET_METER_TELEMETRY_DATA (309 bytes)  in  32ms
```

```
[2019-03-15T18:16:38.940Z]debug: {
         "sequence": 3,
         "frequency": 60015,
         "period": 200,
         "mJp0": 0,
         "mVARsp0": 0,
         "mVAsp0": "43969134143",
         "LNmVp0": 124663,
         "mAp0": 40,
         "q1mJp0": "11665",
         "q2mJp0": "138794001",
         "q3mJp0": "43092186205",
         "q4mJp0": "8916",
         "q1mVARsp0": "114770",
         "q2mVARsp0": "17084202",
         "q3mVARsp0": "378011569",
         "q4mVARsp0": "67452",
         "q1mVAsp0": "412563",
         "q2mVAsp0": "240037355",
         "q3mVAsp0": "43728404531",
         "q4mVAsp0": "279694",
         "mJp1": 0,
         "mVARsp1": 0,
         "mVAsp1": "55924601743",
         "LNmVp1": 124663,
         "mAp1": 1217,
         "q1mJp1": "3152",
         "q2mJp1": "54486965904",
         "q3mJp1": "1364022",
         "q4mJp1": "2163",
         "q1mVARsp1": "54341",
         "q2mVARsp1": "494284883",
         "q3mVARsp1": "33095",
         "q4mVARsp1": "27377",
         "q1mVAsp1": "209595",
         "q2mVAsp1": "55842974578",
         "q3mVAsp1": "81250072",
         "q4mVAsp1": "167498",
         "LLp01mV": 114
}
[2019-03-15T18:16:38.986Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (309 bytes) from 10.130.116.84:32866 [40000c2a69112b6f]
[2019-03-15T18:16:38.986Z]debug: <<< <<< <<<
"45 54 4E 53 63 61 B3 7E 00 02 0F 6F EA 00 00 C8 00 E6 54 8E FE FF FF FF FF 6D 6F EC FF FF FF FF FF BE 59 AB 01 00 00 00 00 6C E6 01
00 09 00 00 00 1A 0C 00 00 00 00 00 00 17 58 00 00 00 00 00 00 F2 6C 71 01 00 00 00 00 D5 0D 00 00 00 00 00 00 02 C0 02 00 00 00 00 00
A3 60 05 00 00 00 00 00 1A 6B 10 00 00 00 00 00 1E 46 0B 00 00 00 00 00 66 76 07 00 00 00 00 00 8C F0 0D 00 00 00 00 00 6D 3E 80 01 00
00 00 00 5F B4 15 00 00 00 00 00 D9 7D D5 9D DB FF FF FF 3A F5 34 CD FF FF FF FF CB 39 38 6C 24 00 00 00 6C E6 01 00 FA 04 00 00 31 02
00 00 00 00 00 00 C2 FE 4B 06 0D 00 00 00 3A 96 DE 5B 17 00 00 00 A4 10 00 00 00 00 00 00 8C 73 00 00 00 00 00 00 9E BF 55 00 00 00 00
00 2C DD 12 33 00 00 00 00 C4 60 0E 00 00 00 00 00 CC A2 01 00 00 00 00 00 A2 12 93 02 0D 00 00 00 AE D8 8C 69 17 00 00 00 AF AB 16 00
00 00 00 00 72 00 00 00 72 84 3C CD B4 F4 E6 46 DF 1E 64 1C 70 D1 43 55 63 AD 00 AA 36 1C 59 C2 4C FC A8 C1 32 92 08 1D"
```

```
    <<< Start        = "45 54 4E 53"     = "ETNS"
    <<< Sequence #   = "63 61 B3 7E"     = 0x7EB36163
    <<< Message Code = "00 02"           = GET_METER_TELEMETRY_DATA (0x200)
    <<< Message Data = "0F 6F EA 00 00 C8 00 E6 54 8E FE FF FF FF FF 6D 6F EC FF FF FF FF FF BE 59 AB 01 00 00 00 00 6C E6 01 00 09 00
00 00 1A 0C 00 00 00 00 00 00 17 58 00 00 00 00 00 00 F2 6C 71 01 00 00 00 00 D5 0D 00 00 00 00 00 00 02 C0 02 00 00 00 00 00 A3 60 05
00 00 00 00 00 1A 6B 10 00 00 00 00 00 1E 46 0B 00 00 00 00 00 66 76 07 00 00 00 00 00 8C F0 0D 00 00 00 00 00 6D 3E 80 01 00 00 00 00
5F B4 15 00 00 00 00 00 D9 7D D5 9D DB FF FF FF 3A F5 34 CD FF FF FF FF CB 39 38 6C 24 00 00 00 6C E6 01 00 FA 04 00 00 31 02 00 00 00
00 00 00 C2 FE 4B 06 0D 00 00 00 3A 96 DE 5B 17 00 00 00 A4 10 00 00 00 00 00 00 8C 73 00 00 00 00 00 00 9E BF 55 00 00 00 00 00 2C DD
12 33 00 00 00 00 C4 60 0E 00 00 00 00 00 CC A2 01 00 00 00 00 00 A2 12 93 02 0D 00 00 00 AE D8 8C 69 17 00 00 00 AF AB 16 00 00 00 00
00 72 00 00 00"
    <<< Crypto Hash  = "72 84 3C CD B4 F4 E6 46 DF 1E 64 1C 70 D1 43 55 63 AD 00 AA 36 1C 59 C2 4C FC A8 C1 32 92 08 1D"

[2019-03-15T18:16:38.987Z]verbose: <<< HANDLING       GET_METER_TELEMETRY_DATA (309 bytes)  in  79ms
[2019-03-15T18:16:38.987Z]debug: {
         "sequence": 15,
         "frequency": 60015,
         "period": 200,
         "mJp0": 0,
         "mVARsp0": 0,
         "mVAsp0": "28006846",
         "LNmVp0": 124524,
         "mAp0": 9,
         "q1mJp0": "3098",
         "q2mJp0": "22551",
         "q3mJp0": "24210674",
         "q4mJp0": "3541",
         "q1mVARsp0": "180226",
         "q2mVARsp0": "352419",
         "q3mVARsp0": "1075994",
         "q4mVARsp0": "738846",
         "q1mVAsp0": "489062",
         "q2mVAsp0": "913548",
         "q3mVAsp0": "25181805",
         "q4mVAsp0": "1422431",
         "mJp1": 0,
         "mVARsp1": 0,
         "mVAsp1": "156434446795",
         "LNmVp1": 124524,
         "mAp1": 1274,
         "q1mJp1": "561",
         "q2mJp1": "55940218562",
         "q3mJp1": "100325561914",
         "q4mJp1": "4260",
         "q1mVARsp1": "29580",
         "q2mVARsp1": "5619614",
         "q3mVARsp1": "856874284",
         "q4mVARsp1": "942276",
         "q1mVAsp1": "107212",
         "q2mVAsp1": "55877767842",
         "q3mVAsp1": "100555085998",
         "q4mVAsp1": "1485743",
         "LLp01mV": 114
}
[2019-03-15T18:16:38.999Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (309 bytes) from 10.130.116.85:32866 [40000c2a6911283d]
[2019-03-15T18:16:38.999Z]debug: <<< <<< <<<
"45 54 4E 53 63 61 B3 7E 00 02 15 6F EA 00 00 C8 00 08 13 87 E0 FE FF FF FF 85 78 FD 50 00 00 00 00 CB DD 25 67 01 00 00 00 95 E7 01
00 09 00 00 00 00 00 00 00 00 00 00 00 34 E2 78 1F 01 00 00 00 C4 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 17 0B 00 00 00 00 00 00
6E 6D FD 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 34 00 00 00 00 00 00 E8 25 25 67 01 00 00 00 88 6D 00 00 00
00 00 00 19 16 00 00 00 00 00 00 50 4C E2 95 F5 FF FF FF 70 B1 63 FC FF FF FF FF 03 AF C8 79 0A 00 00 00 95 E7 01 00 08 00 00 00 00 00
00 00 00 00 00 00 53 75 D3 53 00 00 00 00 5D 3E 4A 16 0A 00 00 00 00 00 00 00 00 00 00 35 3E 02 00 00 00 00 00 04 9E B2 13 00 00 00
00 BA 24 51 17 00 00 00 00 0F 06 00 00 00 00 00 00 E8 84 03 00 00 00 00 00 15 53 29 5D 00 00 00 00 A9 AA 99 1C 0A 00 00 00 5D 2C 02 00
00 00 00 00 72 00 00 00 44 88 34 97 DD 1A 3E D5 75 18 10 FF C1 1C B4 09 E6 63 DE 43 2D 5E 59 C3 9E 85 A3 BA EF DA 89 0B"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "63 61 B3 7E"    = 0x7EB36163
    <<< Message Code = "00 02"          = GET_METER_TELEMETRY_DATA (0x200)
    <<< Message Data = "15 6F EA 00 00 C8 00 08 13 87 E0 FE FF FF FF 85 78 FD 50 00 00 00 00 CB DD 25 67 01 00 00 00 95 E7 01 00 09 00
00 00 00 00 00 00 00 00 00 00 34 E2 78 1F 01 00 00 00 C4 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 17 0B 00 00 00 00 00 00 6E 6D FD
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 34 00 00 00 00 00 00 E8 25 25 67 01 00 00 00 88 6D 00 00 00 00 00 00
19 16 00 00 00 00 00 00 50 4C E2 95 F5 FF FF FF 70 B1 63 FC FF FF FF FF 03 AF C8 79 0A 00 00 00 95 E7 01 00 08 00 00 00 00 00 00 00
00 00 00 53 75 D3 53 00 00 00 00 5D 3E 4A 16 0A 00 00 00 00 00 00 00 00 00 00 35 3E 02 00 00 00 00 00 04 9E B2 13 00 00 00 00 BA 24
51 17 00 00 00 00 0F 06 00 00 00 00 00 00 E8 84 03 00 00 00 00 00 15 53 29 5D 00 00 00 00 A9 AA 99 1C 0A 00 00 00 5D 2C 02 00 00 00 00
00 72 00 00 00"
    <<< Crypto Hash  = "44 88 34 97 DD 1A 3E D5 75 18 10 FF C1 1C B4 09 E6 63 DE 43 2D 5E 59 C3 9E 85 A3 BA EF DA 89 0B"

[2019-03-15T18:16:38.999Z]verbose: <<< HANDLING       GET_METER_TELEMETRY_DATA (309 bytes)  in  91ms
[2019-03-15T18:16:38.999Z]debug: {
         "sequence": 21,
         "frequency": 60015,
         "period": 200,
         "mJp0": 0,
         "mVARsp0": 0,
         "mVAsp0": "6025502155",
         "LNmVp0": 124821,
         "mAp0": 9,
         "q1mJp0": "0",
         "q2mJp0": "4822983220",
         "q3mJp0": "2756",
         "q4mJp0": "0",
         "q1mVARsp0": "2839",
         "q2mVARsp0": "1358785902",
         "q3mVARsp0": "0",
         "q4mVARsp0": "0",
         "q1mVAsp0": "13378",
         "q2mVAsp0": "6025455080",
         "q3mVAsp0": "28040",
         "q4mVAsp0": "5657",
         "mJp1": 0,
         "mVARsp1": 0,
         "mVAsp1": "44992868099",
         "LNmVp1": 124821,
         "mAp1": 8,
         "q1mJp1": "0",
         "q2mJp1": "1406367059",
         "q3mJp1": "43323637341",
         "q4mJp1": "0",
         "q1mVARsp1": "146997",
         "q2mVARsp1": "330472964",
         "q3mVARsp1": "391193786",
         "q4mVARsp1": "1551",
         "q1mVAsp1": "230632",
         "q2mVAsp1": "1562989333",
         "q3mVAsp1": "43429505705",
         "q4mVAsp1": "142429",
         "LLp01mV": 114
}
```

6.2.5 Set Next Expected UDP Sequence Number

Reference: 5.5 Set Next Expected UDP Sequence Number

```
[2019-03-15T14:28:18.100Z]verbose: >>>> SENDING >>>>>>SET_NEXT_SEQUENCE_NUMBER ( 46 bytes)  to  10.130.116.50:32866 [30000c2a690c7652]
[2019-03-15T14:28:18.100Z]debug: >>> >>> >>>
"45 54 4E 4D B1 81 FB 64 00 80 10 8A C1 65 27 5F 9B 97 E4 0D 3B FB 1E CA CC C4 A1 27 9E 51 F7 42 B5 42 0E 64 16 DC DE A0 90 7F 64 A1
7F 5F"
    >>> Start        = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #   = "B1 81 FB 64"    = 0x64FB81B1
    >>> Message Code = "00 80"          = SET_NEXT_SEQUENCE_NUMBER (0x8000)
    >>> Message Data = "10 8A C1 65"
    >>> Crypto Hash  = "27 5F 9B 97 E4 0D 3B FB 1E CA CC C4 A1 27 9E 51 F7 42 B5 42 0E 64 16 DC DE A0 90 7F 64 A1 7F 5F"

[2019-03-15T14:28:18.136Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.50:32866 [30000c2a690c7652]
[2019-03-15T14:28:18.136Z]debug: <<< <<< <<<
"45 54 4E 53 B1 81 FB 64 00 80 00 E5 74 BE 5C 0F 3C 27 5C 3D 2C 19 4E 5C 6B C3 F7 C5 4F CF 92 EF D1 ED 94 A7 E5 EB 54 CC 3E D7 27"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "B1 81 FB 64"    = 0x64FB81B1
    <<< Message Code = "00 80"          = SET_NEXT_SEQUENCE_NUMBER (0x8000)
    <<< Message Data = "00"
    <<< Crypto Hash  = "E5 74 BE 5C 0F 3C 27 5C 3D 2C 19 4E 5C 6B C3 F7 C5 4F CF 92 EF D1 ED 94 A7 E5 EB 54 CC 3E D7 27"

[2019-03-15T14:28:18.136Z]verbose: <<< HANDLING       SET_NEXT_SEQUENCE_NUMBER ( 43 bytes)  in  36ms
[2019-03-15T14:28:18.136Z]debug: {
         "ack": 0,
         "ackString": "Acknowledged",
         "nextSequenceNumber": 1707182608
}
[2019-03-15T14:28:18.136Z]verbose: Setting instance Sequence Number to 0x65C18A10 = 1707182608
[2019-03-15T14:28:18.137Z]verbose: ---------------------------------30000c2a690c7652 Initialized -----------------------------

[2019-03-15T14:28:18.303Z]verbose: >>>> SENDING >>>>>>SET_NEXT_SEQUENCE_NUMBER ( 46 bytes)  to  10.130.116.85:32866 [30000c2a6911283d]
[2019-03-15T14:28:18.303Z]debug: >>> >>> >>>
"45 54 4E 4D F8 24 8E C3 00 80 10 8A C1 65 5F 27 AE 93 F2 01 AD 04 36 10 24 BD 39 2F 09 85 D0 66 3E C7 29 3D F4 C3 78 FA 99 10 AA E2
2C 9E"
    >>> Start        = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #   = "F8 24 8E C3"    = 0xC38E24F8
    >>> Message Code = "00 80"          = SET_NEXT_SEQUENCE_NUMBER (0x8000)
    >>> Message Data = "10 8A C1 65"
    >>> Crypto Hash  = "5F 27 AE 93 F2 01 AD 04 36 10 24 BD 39 2F 09 85 D0 66 3E C7 29 3D F4 C3 78 FA 99 10 AA E2 2C 9E"

[2019-03-15T14:28:18.352Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.85:32866 [30000c2a6911283d]
[2019-03-15T14:28:18.353Z]debug: <<< <<< <<<
"45 54 4E 53 F8 24 8E C3 00 80 00 3C 10 AF A0 58 68 4D A5 3D 69 BE EF B5 A4 84 78 C7 61 5B 5B B7 AB FA 67 FA B4 CE 06 38 CA 1E F2"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "F8 24 8E C3"    = 0xC38E24F8
    <<< Message Code = "00 80"          = SET_NEXT_SEQUENCE_NUMBER (0x8000)
    <<< Message Data = "00"
    <<< Crypto Hash  = "3C 10 AF A0 58 68 4D A5 3D 69 BE EF B5 A4 84 78 C7 61 5B 5B B7 AB FA 67 FA B4 CE 06 38 CA 1E F2"

[2019-03-15T14:28:18.353Z]verbose: <<< HANDLING       SET_NEXT_SEQUENCE_NUMBER ( 43 bytes)  in  50ms
[2019-03-15T14:28:18.353Z]debug: {
         "ack": 0,
         "ackString": "Acknowledged",
         "nextSequenceNumber": 1707182608
}
[2019-03-15T14:28:18.353Z]verbose: Setting instance Sequence Number to 0x65C18A10 = 1707182608
[2019-03-15T14:28:18.353Z]verbose: ---------------------------------30000c2a6911283d Initialized ----------------------------

[2019-03-15T14:28:18.507Z]verbose: >>>> SENDING >>>>>>SET_NEXT_SEQUENCE_NUMBER ( 46 bytes)  to  10.130.116.84:32866 [30000c2a69112b6f]
[2019-03-15T14:28:18.507Z]debug: >>> >>> >>>
"45 54 4E 4D 8B BD 1F 0E 00 80 10 8A C1 65 D6 00 39 C2 AA 2E F3 0A A5 B2 C5 41 07 EF 08 A5 68 7E F1 17 D1 CF 9B F0 28 4B 59 41 45 E0
2C 17"
    >>> Start         = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #    = "8B BD 1F 0E"    = 0xE1FBD8B
    >>> Message Code = "00 80"           = SET_NEXT_SEQUENCE_NUMBER (0x8000)
    >>> Message Data = "10 8A C1 65"
    >>> Crypto Hash  = "D6 00 39 C2 AA 2E F3 0A A5 B2 C5 41 07 EF 08 A5 68 7E F1 17 D1 CF 9B F0 28 4B 59 41 45 E0 2C 17"

[2019-03-15T14:28:18.515Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.84:32866 [30000c2a69112b6f]
[2019-03-15T14:28:18.515Z]debug: <<< <<< <<<
"45 54 4E 53 8B BD 1F 0E 00 80 00 07 5A 82 AA 2A 50 28 1D A6 D2 70 FB 0A DF 38 75 30 1F 4C EF 2A 10 DD BC 98 7D B0 D9 C6 68 C4 C7"
    <<< Start         = "45 54 4E 53"    = "ETNS"
    <<< Sequence #    = "8B BD 1F 0E"    = 0xE1FBD8B
    <<< Message Code = "00 80"           = SET_NEXT_SEQUENCE_NUMBER (0x8000)
    <<< Message Data = "00"
    <<< Crypto Hash  = "07 5A 82 AA 2A 50 28 1D A6 D2 70 FB 0A DF 38 75 30 1F 4C EF 2A 10 DD BC 98 7D B0 D9 C6 68 C4 C7"

[2019-03-15T14:28:18.516Z]verbose: <<< HANDLING       SET_NEXT_SEQUENCE_NUMBER ( 43 bytes)  in  8ms
[2019-03-15T14:28:18.516Z]debug: {
         "ack": 0,
         "ackString": "Acknowledged",
         "nextSequenceNumber": 1707182608
}
[2019-03-15T14:28:18.516Z]verbose: Setting instance Sequence Number to 0x65C18A10 = 1707182608
[2019-03-15T14:28:18.516Z]verbose: ---------------------------------30000c2a69112b6f Initialized ----------------------------

[2019-03-15T14:28:18.711Z]verbose: >>>> SENDING >>>>>>SET_NEXT_SEQUENCE_NUMBER ( 46 bytes)  to  10.130.116.28:32866 [30000c2a69113173]
[2019-03-15T14:28:18.711Z]debug: >>> >>> >>>
"45 54 4E 4D C2 51 3B 09 00 80 10 8A C1 65 B4 CE D6 5B DE 7C 7A 7A 58 93 49 D6 60 68 F2 E3 B7 84 76 82 CA 2C DF 2A 04 54 EB B4 19 DB
96 DA"
    >>> Start         = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #    = "C2 51 3B 09"    = 0x93B51C2
    >>> Message Code = "00 80"           = SET_NEXT_SEQUENCE_NUMBER (0x8000)
    >>> Message Data = "10 8A C1 65"
    >>> Crypto Hash  = "B4 CE D6 5B DE 7C 7A 7A 58 93 49 D6 60 68 F2 E3 B7 84 76 82 CA 2C DF 2A 04 54 EB B4 19 DB 96 DA"
```

```
[2019-03-15T14:28:18.761Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.28:32866 [30000c2a69113173]
[2019-03-15T14:28:18.762Z]debug: <<< <<< <<<
"45 54 4E 53 C2 51 3B 09 00 80 00 14 E3 A7 CF 40 74 13 6B 77 CE C0 72 56 4F 15 79 9B 01 7C AA D2 45 C6 56 0F 82 88 39 37 71 BD 27"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "C2 51 3B 09"    = 0x93B51C2
    <<< Message Code = "00 80"          = SET_NEXT_SEQUENCE_NUMBER (0x8000)
    <<< Message Data = "00"
    <<< Crypto Hash  = "14 E3 A7 CF 40 74 13 6B 77 CE C0 72 56 4F 15 79 9B 01 7C AA D2 45 C6 56 0F 82 88 39 37 71 BD 27"

[2019-03-15T14:28:18.762Z]verbose: <<< HANDLING       SET_NEXT_SEQUENCE_NUMBER ( 43 bytes)  in  51ms
[2019-03-15T14:28:18.762Z]debug: {
         "ack": 0,
         "ackString": "Acknowledged",
         "nextSequenceNumber": 1707182608
}
[2019-03-15T14:28:18.762Z]verbose: Setting instance Sequence Number to 0x65C18A10 = 1707182608
[2019-03-15T14:28:18.762Z]verbose: ---------------------------------30000c2a69113173 Initialized ----------------------------
```

6.2.6 Set Breaker Remote Handle Position

Reference: 5.6 Set Breaker Remote Handle Position

```
[2019-03-15T14:28:19.520Z]verbose: >>>> SENDING SET_BREAKER_REMOTE_HANDLE_POSITION ( 43 bytes)  to  10.130.116.255:32866 (BROADCAST)
[2019-03-15T14:28:19.520Z]debug: >>> >>> >>>
"45 54 4E 4D 10 8A C1 65 00 81 00 4A 08 FE B7 D7 5E AD 91 CC E5 A4 B8 1D BE D7 CE BF F7 96 48 BD 11 BD 09 89 87 7E CF FC 55 FF F3"
    >>> Start         = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #    = "10 8A C1 65"    = 0x65C18A10
    >>> Message Code = "00 81"           = SET_BREAKER_REMOTE_HANDLE_POSITION (0x8100)
    >>> Message Data = "00"
    >>> Crypto Hash  = "4A 08 FE B7 D7 5E AD 91 CC E5 A4 B8 1D BE D7 CE BF F7 96 48 BD 11 BD 09 89 87 7E CF FC 55 FF F3"

[2019-03-15T14:28:19.540Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 44 bytes) from 10.130.116.50:32866 [30000c2a690c7652]
[2019-03-15T14:28:19.540Z]debug: <<< <<< <<<
"45 54 4E 53 10 8A C1 65 00 81 00 00 8A 07 C9 E4 41 C5 C1 17 D9 82 D9 81 CE 65 56 59 5B 1A 4C 26 FB 49 DF E8 4F D6 FB 93 E9 A7 55 0F"
    <<< Start         = "45 54 4E 53"    = "ETNS"
    <<< Sequence #    = "10 8A C1 65"    = 0x65C18A10
    <<< Message Code = "00 81"           = SET_BREAKER_REMOTE_HANDLE_POSITION (0x8100)
    <<< Message Data = "00 00"
    <<< Crypto Hash  = "8A 07 C9 E4 41 C5 C1 17 D9 82 D9 81 CE 65 56 59 5B 1A 4C 26 FB 49 DF E8 4F D6 FB 93 E9 A7 55 0F"

[2019-03-15T14:28:19.541Z]verbose: <<< HANDLING SET_BREAKER_REMOTE_HANDLE_POSITION ( 44 bytes)  in  21ms
[2019-03-15T14:28:19.541Z]debug: {
         "ack": 0,
         "state": 0,
         "stateString": "Open"
}
[2019-03-15T14:28:19.642Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 44 bytes) from 10.130.116.85:32866 [30000c2a6911283d]
[2019-03-15T14:28:19.642Z]debug: <<< <<< <<<
"45 54 4E 53 10 8A C1 65 00 81 00 00 8A 07 C9 E4 41 C5 C1 17 D9 82 D9 81 CE 65 56 59 5B 1A 4C 26 FB 49 DF E8 4F D6 FB 93 E9 A7 55 0F"
    <<< Start         = "45 54 4E 53"    = "ETNS"
    <<< Sequence #    = "10 8A C1 65"    = 0x65C18A10
    <<< Message Code = "00 81"           = SET_BREAKER_REMOTE_HANDLE_POSITION (0x8100)
    <<< Message Data = "00 00"
    <<< Crypto Hash  = "8A 07 C9 E4 41 C5 C1 17 D9 82 D9 81 CE 65 56 59 5B 1A 4C 26 FB 49 DF E8 4F D6 FB 93 E9 A7 55 0F"

[2019-03-15T14:28:19.643Z]verbose: <<< HANDLING SET_BREAKER_REMOTE_HANDLE_POSITION ( 44 bytes)  in  123ms
[2019-03-15T14:28:19.643Z]debug: {
         "ack": 0,
         "state": 0,
         "stateString": "Open"
}
[2019-03-15T14:28:19.662Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 44 bytes) from 10.130.116.84:32866 [30000c2a69112b6f]
[2019-03-15T14:28:19.663Z]debug: <<< <<< <<<
"45 54 4E 53 10 8A C1 65 00 81 00 00 8A 07 C9 E4 41 C5 C1 17 D9 82 D9 81 CE 65 56 59 5B 1A 4C 26 FB 49 DF E8 4F D6 FB 93 E9 A7 55 0F"
    <<< Start         = "45 54 4E 53"    = "ETNS"
    <<< Sequence #    = "10 8A C1 65"    = 0x65C18A10
    <<< Message Code = "00 81"          = SET_BREAKER_REMOTE_HANDLE_POSITION (0x8100)
    <<< Message Data = "00 00"
    <<< Crypto Hash  = "8A 07 C9 E4 41 C5 C1 17 D9 82 D9 81 CE 65 56 59 5B 1A 4C 26 FB 49 DF E8 4F D6 FB 93 E9 A7 55 0F"

[2019-03-15T14:28:19.663Z]verbose: <<< HANDLING SET_BREAKER_REMOTE_HANDLE_POSITION ( 44 bytes)  in  143ms
[2019-03-15T14:28:19.663Z]debug: {
         "ack": 0,
         "state": 0,
         "stateString": "Open"
}
[2019-03-15T14:28:19.690Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 44 bytes) from 10.130.116.28:32866 [30000c2a69113173]
[2019-03-15T14:28:19.741Z]info: Toggled all breakers successfully!
[2019-03-15T14:28:19.908Z]info: { responses:
   { '10.130.116.50': { ack: 0, state: 0, stateString: 'Open' },
     '10.130.116.85': { ack: 0, state: 0, stateString: 'Open' },
     '10.130.116.84': { ack: 0, state: 0, stateString: 'Open' },
     '10.130.116.28': { ack: 0, state: 0, stateString: 'Open' } } }
```

6.2.7 Set Bargraph LED to User Defined Color

Reference: 5.7 Set Bargraph LED to User Defined Color

```
[2019-03-21T18:02:36.254Z]debug: Setting Bargraph LED to red for 10 seconds with Blinking = true
[2019-03-21T18:02:36.457Z]verbose: >>>> SENDING SET_BARGRAPH_LED_TO_USER_DEFINED ( 67 bytes)  to  10.130.116.50:32866
[30000c2a690c7652]
[2019-03-21T18:02:36.457Z]debug: >>> >>> >>>
"45 54 4E 4D 1B C0 05 0D 00 83 01 0A 00 00 00 FF 00 00 01 FF 00 00 01 FF 00 00 01 FF 00 00 01 FF 00 00 01 69 4D E4 5C 36 0A D5 1B 2C
C5 21 7A 99 55 EB 86 14 64 C3 9B D3 21 6A 3F 86 76 D6 83 8C 0E 9E F5"
    >>> Start        = "45 54 4E 4D"    = "ETNM"
    >>> Sequence #   = "1B C0 05 0D"    = 0xD05C01B
    >>> Message Code = "00 83"          = SET_BARGRAPH_LED_TO_USER_DEFINED (0x8300)
    >>> Message Data = "01 0A 00 00 00 FF 00 00 01 FF 00 00 01 FF 00 00 01 FF 00 00 01 FF 00 00 01"
    >>> Crypto Hash  = "69 4D E4 5C 36 0A D5 1B 2C C5 21 7A 99 55 EB 86 14 64 C3 9B D3 21 6A 3F 86 76 D6 83 8C 0E 9E F5"

[2019-03-21T18:02:36.495Z]debug: <<< RECEIVED <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ( 43 bytes) from 10.130.116.50:32866 [30000c2a690c7652]
[2019-03-21T18:02:36.495Z]debug: <<< <<< <<<
"45 54 4E 53 1B C0 05 0D 00 83 00 89 8C 1C 30 F2 34 AA D9 8F 7F D8 AD 5D 2D 38 72 17 6D 02 06 EF 11 B8 5A 85 6C 7D 8A D3 1C EA E7"
    <<< Start        = "45 54 4E 53"    = "ETNS"
    <<< Sequence #   = "1B C0 05 0D"    = 0xD05C01B
    <<< Message Code = "00 83"          = SET_BARGRAPH_LED_TO_USER_DEFINED (0x8300)
    <<< Message Data = "00"
    <<< Crypto Hash  = "89 8C 1C 30 F2 34 AA D9 8F 7F D8 AD 5D 2D 38 72 17 6D 02 06 EF 11 B8 5A 85 6C 7D 8A D3 1C EA E7"

[2019-03-21T18:02:36.495Z]verbose: <<< HANDLING SET_BARGRAPH_LED_TO_USER_DEFINED ( 43 bytes)  in  38ms
[2019-03-21T18:02:36.495Z]debug: {
         "ack": 0
}
```