



Alinket Electronic Technology (Shanghai) Co.,Ltd

# **Alinket Controller Message Test Host (ACMTH) User Guide**

Version: 0.1

Date: 2016.7.2

Copyright

©Alinket Technologies Corp



This document is the property of Alinket Technologies Corp and is delivered on the express condition that it not be disclosed, reproduced in whole or in part, or used for manufacture for anyone other than Alinket Technologies Corp without its written consent, and that no right is granted to disclose or so use any information contained in said document. This restriction does not limit the right to use information obtained from other sources.

#### Trademarks

ACMTH is a registered trademark of Alinket Technologies Corp. Alinket is a registered trademark of Alinket Technologies Corp. Other products and company names mentioned herein may be the trademarks of their respective owners.



## Revision History

Date	Revision Content	Revision By	Version
2016/7/2	Initialization Version	Dechen Xu	0.1
2018/6/1	Add Set Checksum Type	Huajun Li	0.2



## List of Contents

USER GUIDE .....	1
<b>1 OVERVIEW .....</b>	<b>1</b>
1.1 SCOPE .....	1
1.2 AUDIENCE .....	1
1.3 TERMS AND ABBREVIATIONS .....	1
1.4 REFERENCE .....	2
<b>2 APPLICATION OVERVIEW .....</b>	<b>3</b>
2.1 MINIMUM SYSTEM REQUIREMENT .....	3
2.2 INSTALLATION AND EXECUTION .....	3
2.3 MAIN FEATURE .....	3
2.4 GRAPHIC USER INTERFACE .....	4
<b>3 FEATURE GUIDE .....</b>	<b>7</b>
3.1 MODULE PORT MANAGEMENT .....	7
3.1.1 Add Port .....	7



---

List of Figures

Figure 1 ACMTH work flow .....	3
Figure 2 GUI of the application.....	5
Figure 3 Pick Serial Port Dialog.....	7



List of Tables

Table 1 Terms and Abbreviations .....	1
Table 2 Minimum System Requirement .....	3

# 1 Overview

## 1.1 Scope

This documentation describes the usage of the Alinket Controller Message Test Host (ACMTH). ACMTH is a Microsoft .NET Framework based application which is designed to test and debug the Alinket IoT controller running with AiDK® ACM firmware.

## 1.2 Audience

This documentation is for the customers who is developing the Application on their host board with Alinket IoT controllers on ACM firmware.

## 1.3 Terms and Abbreviations

Table 1 Terms and Abbreviations

ACM	Alinket Controller Message
BLE	Bluetooth Low Energy
HID	Human Interface Device
HOGP	HID Over GATT Profile
GATT	Generic Attribute Profile
DUN	Dial-Up Networking
FTP	File Transfer Protocol
GOEP	Generic Object Exchange Profile
GAP	Generic Access Profile
HFP	Hands-free Profile
SAP	SIM Access Profile
SDAP	Service Discovery Application Profile
SPP	Serial Port Profile
TCP	Transmission Control Protocol
UDP	User Datagram Protocol



MIDI	Musical Instrument Digital Interface
MCU	Microprogrammed Control Unit
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface
GPIO	General Purpose Input/Output

## 1.4 Reference

[1] Alinket Host Controller Message Interface Guide




## 2 Application Overview

### 2.1 Minimum System Requirement

Table 2 Minimum System Requirement

Requirement	Version	Description
Windows OS	Win7 or Above	
.NET Framework	4.0	Integrated in windows 7.0 and above

### 2.2 Installation and Execution

This application is green software and no need to run the install at all, just need to extract the release package to any folders. Double click the file ACMTH.exe with the ICON  and the GUI interface will appear.

### 2.3 Main Feature

ACMTH is a GUI based application so user can editor the ACM request message and read the response message easily. Figure 1 shows the flow how the application works.

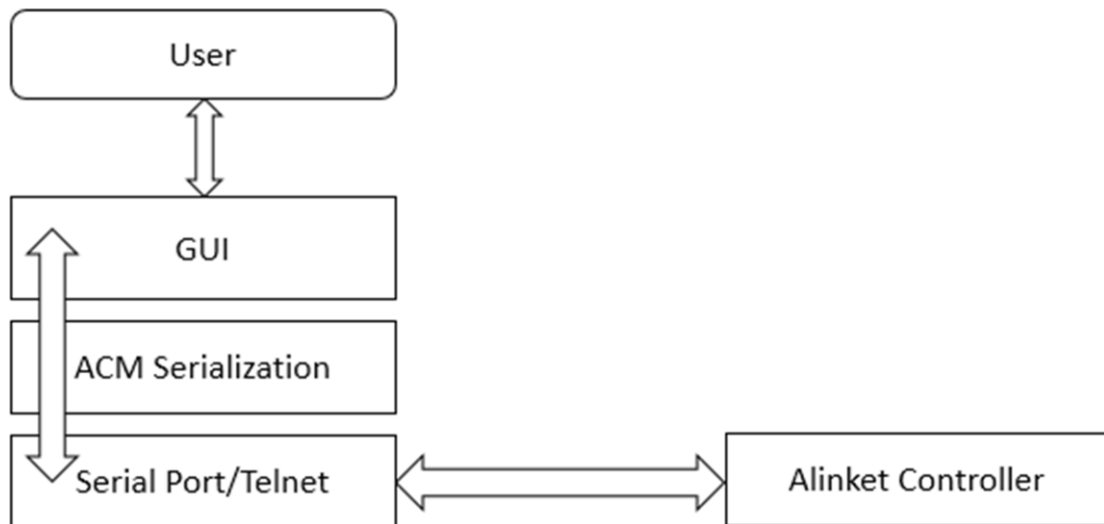


Figure 1 ACMTH work flow

User can edit the content of the ACM message and the application serializes the content to ACM message bytes sequence and sends the sequence to Controller thru PC Serial Port. Once the application receive the ACM response or indication from the controller, it can parse the messages and display the content in GUI.



---

The application also supports some function which needs the combination of the Alinket Controller Messages.

### **Traffic Simulation**

Traffic Simulation is a sub-tool which is used to test the module traffic transmitting and receiving. It contains the traffic definition (data length, payload type, transmit rate, etc...) and data statistics.

### **OTA Server**

OTA server is used to test the Alinket Controller firmware OTA upgrade feature.

### **EAP Configuration**

EAP Configuration is used to configure the WIFI module 802.1X with EAP authentication, include the TLS authentication file download.

The application also provides some useful tools.

### **Playback Editor**

Playback editor can record the Alinket Controller Messages sends by user. User can modify the messages in the recorded list or add/remove messages into/from the list and save the list to a file on the disk. The saved list can be loaded and played back again. This makes user more easier do some configuration or do some stress test.

### **Hex to String Editor**

This tool is a converter between the ASCII and HEX.

## **2.4 Graphic User Interface**

ACMTH is a GUI based application, figure 2 shows the main appearance of the application.

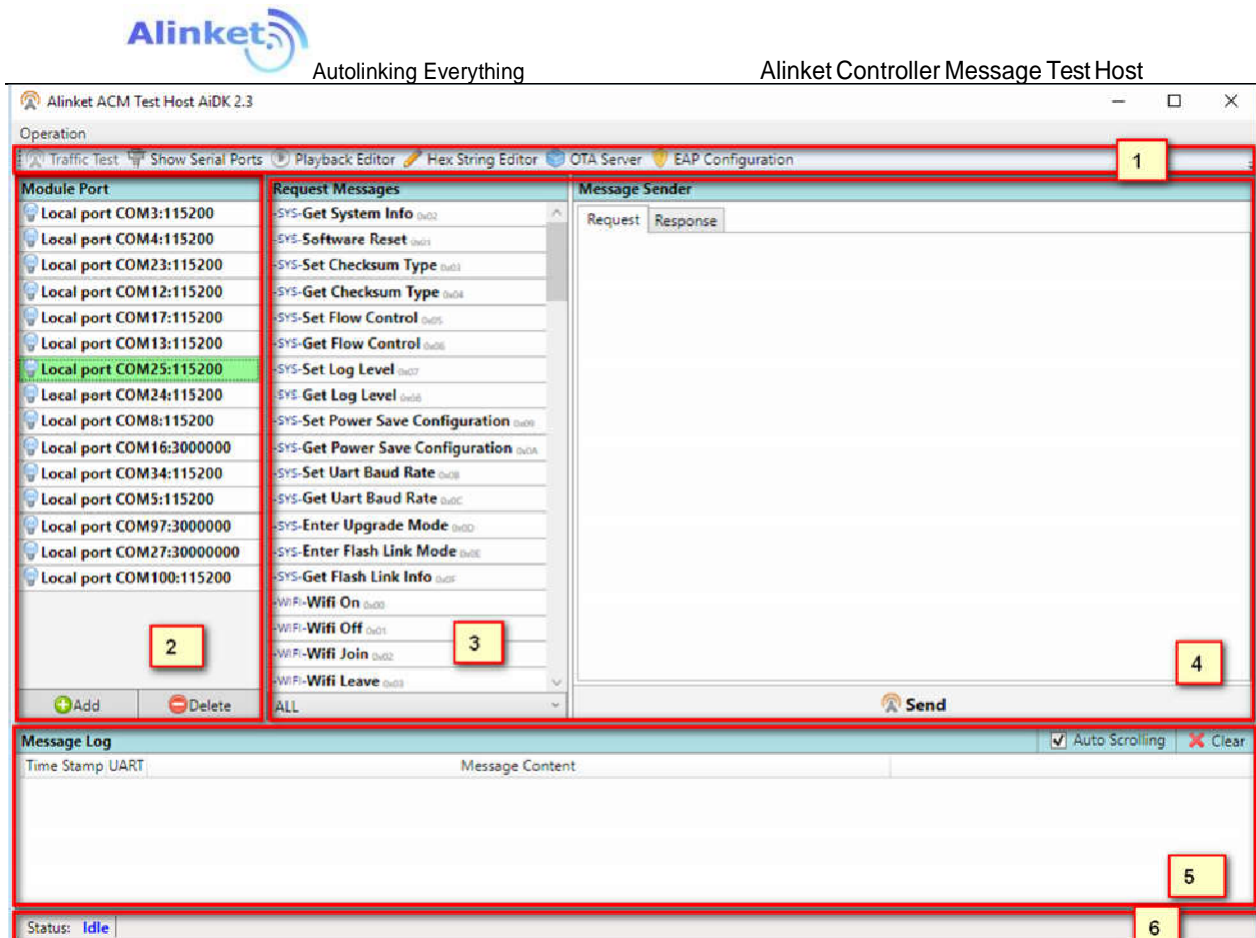


Figure 2 GUI of the application

(1) **Utilities Toolbar.**

The toolbar includes the utilities user can launch.

(2) **Module Port List.**

User can manage the ports which connected to Alinket Controller Module and select the port to send the message.

(3) **Message List.**

The supported messages list.

(4) **Messages Request and Response**

User configure the request message in "Request" Tab and get the response message in "Response" Tab.

(5) **Messages Log**

All the messages are logged in the message log box.



**(6) Status Bar.**

Indicate the message sending status.

## 3 Feature Guide

### 3.1 Module Port Management

#### 3.1.1 Add/Remove Port

ACMTH can manage multiple modules in parallel, user need to add the management port in the module port list first.

##### How to Add Port:

**Step-1** Click the  Button below the Module Port list, there will be a dialog appeared for user to fill in the serial port parameters.

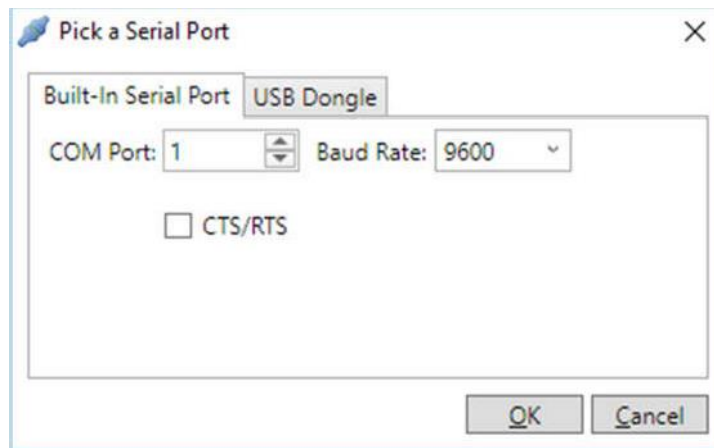


Figure 3 Pick Serial Port Dialog

**Step-2** Select the COM Port number, Baud Rate and check the CTS/RTS if the module is connected as Hardware flow control mode\*. If the user uses some special baud rate which is not listed, the special value can be inputted directly into the combo box. Once the parameters are all configured, press "OK" button. The port will be added into the module port list.

All the added ports will be saved to "%system driver%\ProgramData\Alinket\ACMTH\connections.xml" \*.


##### How to remove Port

**Step-1** Select a Port which will be removed in Module Port List and Click the  Button below the Module Port List.

\* %system driver%\ProgramData folder is a hidden folder in default.

### 3.1.2 Port Connection

There are 2 ways to connect/disconnect the port.

- (1) Double Click the Port, if the port connection established successfully, the bulb icon  will turn to light. If the connection established failed, the error message will be displayed. (eg, the comm port doesn't exist). Double click the Port again, the port will be disconnected and the light bulb icon turns to off.
- (2) Right click on the port and there will be a context menu displayed, then select the "Connect" or "Disconnect".

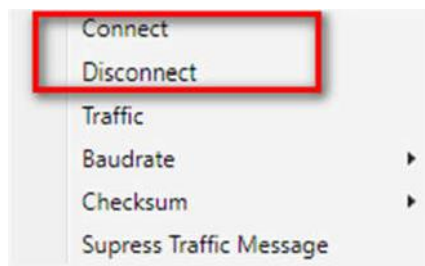


Figure 4 Context Menu of the Module Port

### 3.1.3 Change Baud Rate on Fly

ACMTH support changing the Baud Rate "on fly" which means user only need to right click on the port and select a new baud rate even the connection is established. This is useful and easy for user after change the module UART baud rate.

Figure 5 shows the context menu to modify the baud rate on fly.

**Note: this function only support 1 port in list.**

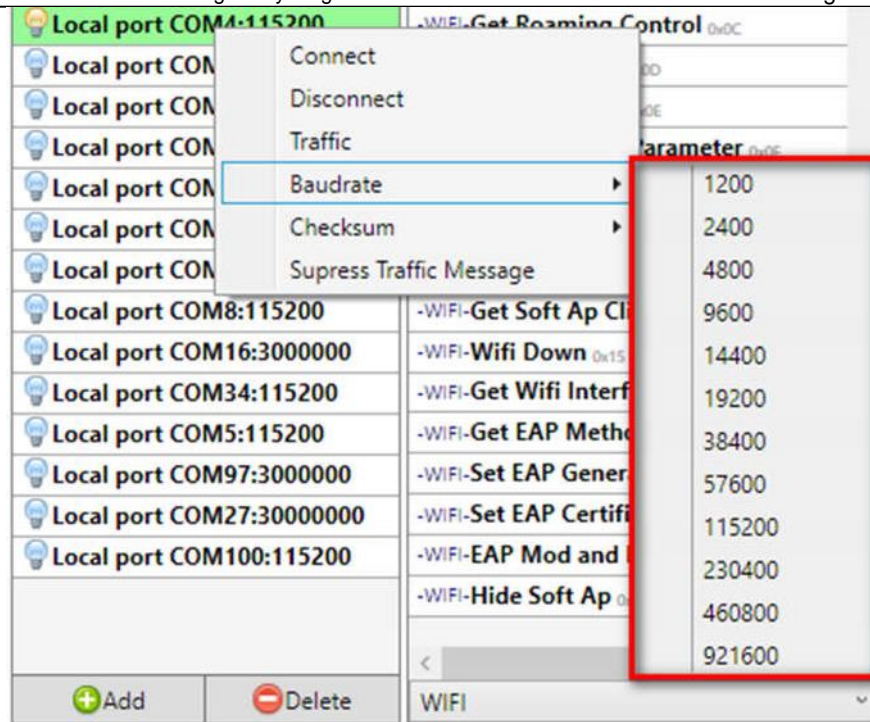


Figure 5 Change Baud Rate on Fly

### 3.1.4 Set Checksum Type on Fly

ACMTH now supports 3 types of checksum: None, Header and Full. User can change it “on fly”, which means that user only need to right click on the port and select a new checksum type even the connection is established.

Figure 6 shows the context menu to modify the checksum type on fly.

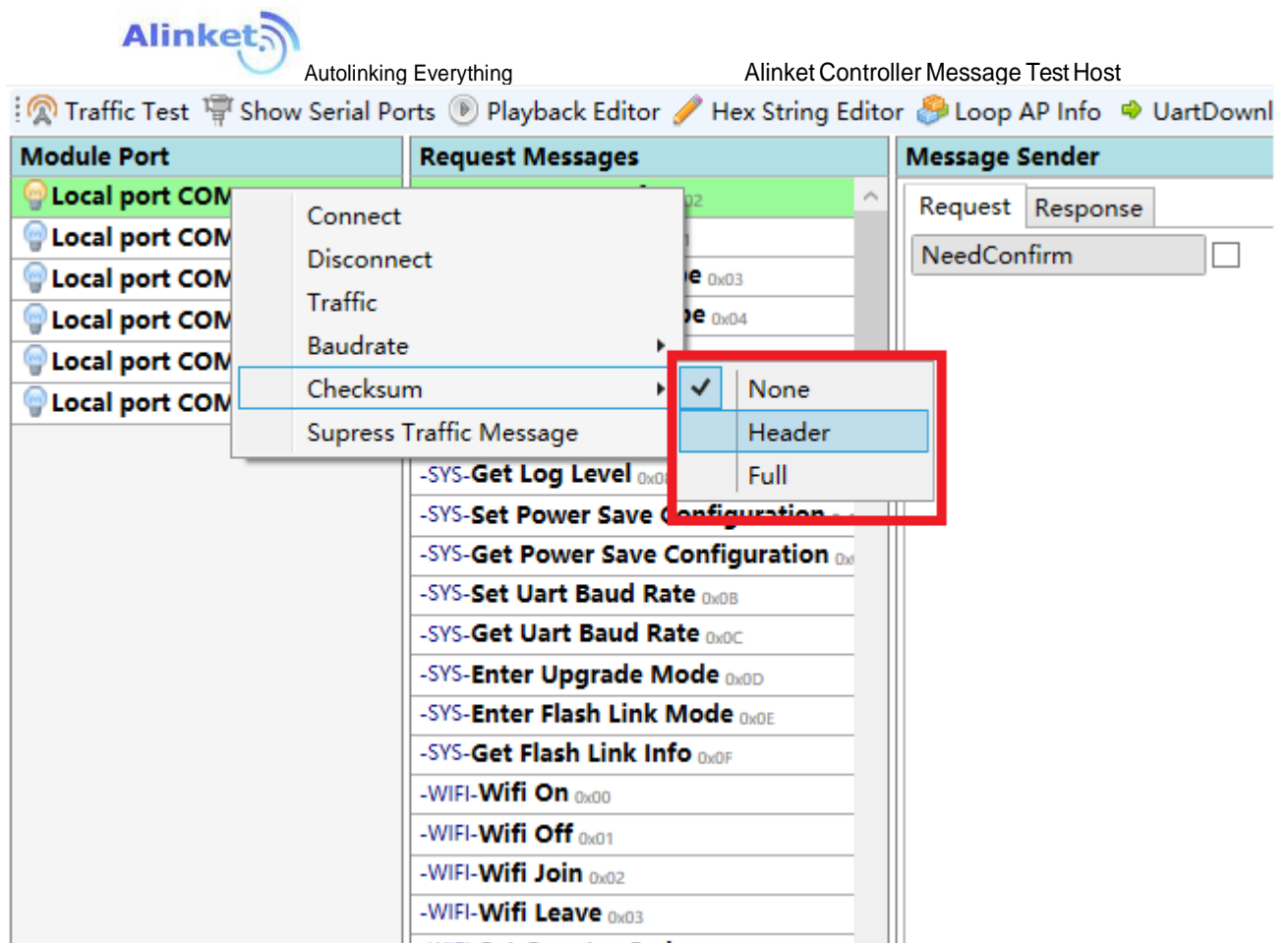


Figure 6 Change Checksum Type on fly

**Note:**

Before using this function, user must send "Set Checksum Type" message to the module and set the same checksum type parameter first.

AS Figure 7 shows :



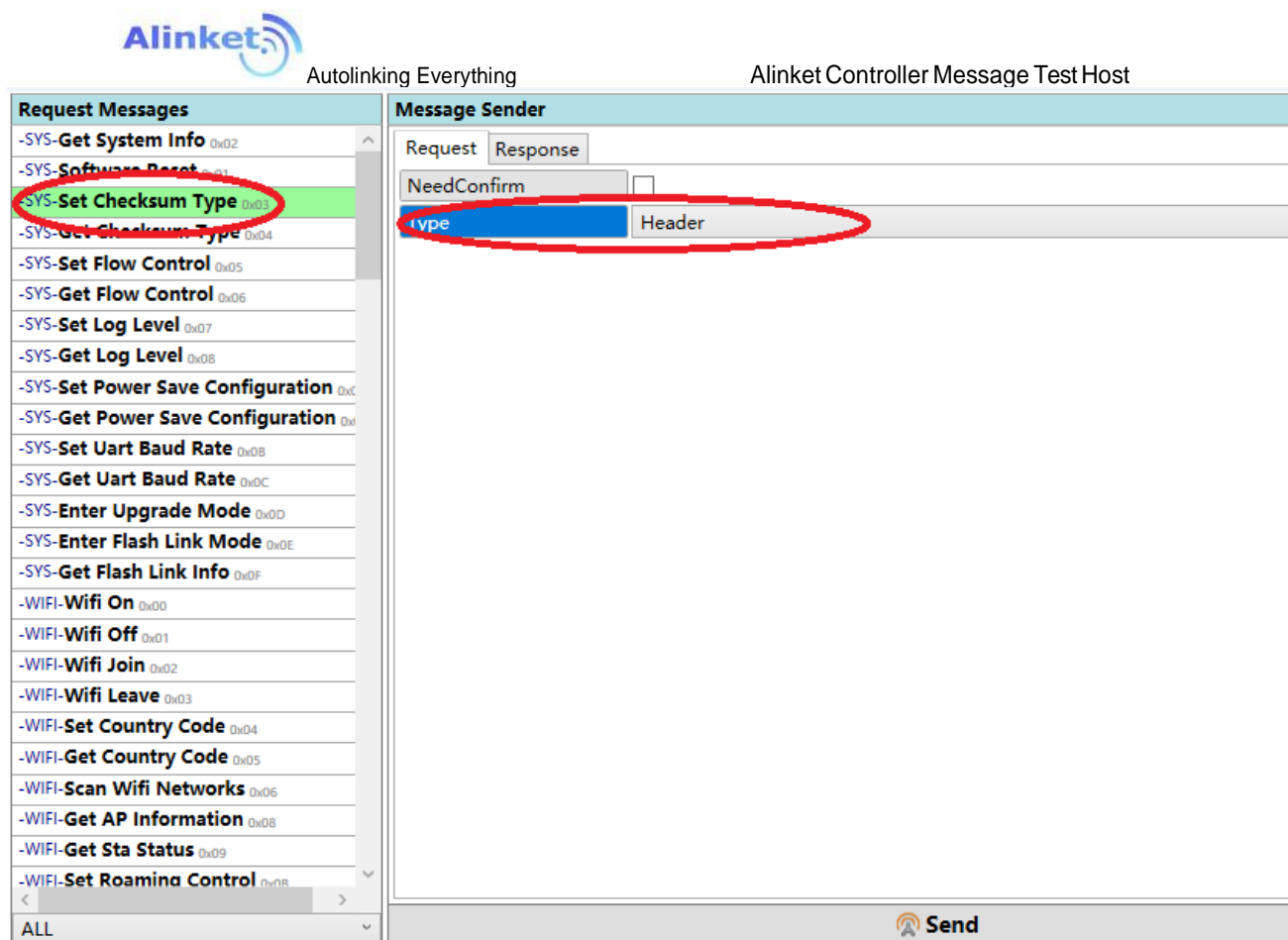


Figure 7 Change the Checksum Type

## 3.2 Alinket Controller MessageOperation

### 3.2.1 Send Message

Once the port connected, user can start send message to the module to make the configuration. The messages are grouped by the branch, user can filter the messages by the combo box below the messages list, and only the selected branch will be listed. As Figure 8 shows.

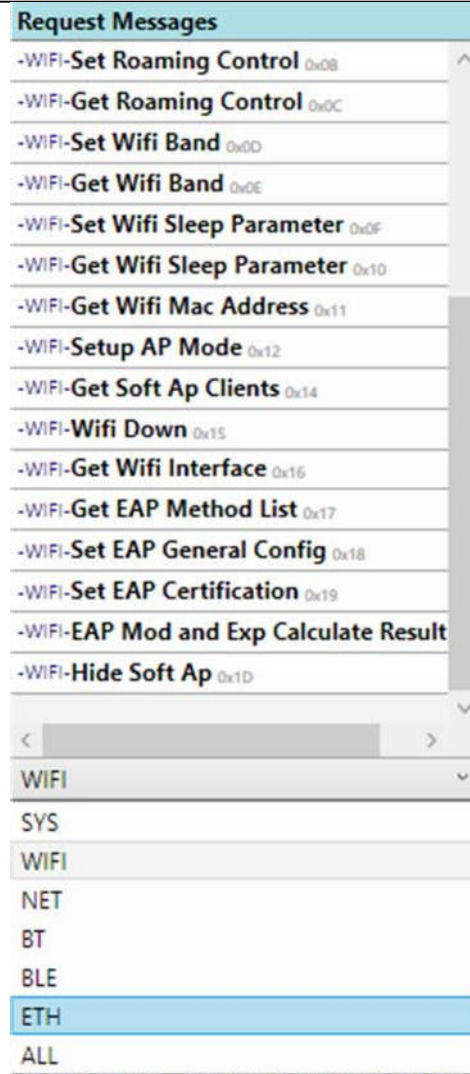


Figure 8 ACM Branch Filter

The Hex Number follows the message name is the message leaf number.

**Step-1:** Select a message, the message parameters configuration tab will be displayed in the “Request Tab” in “Message Sender”. As the Figure 9 shows, “Wifi Join” is the message which is used to make the WIFI controller module join the specific SSID, so user can set the join parameters in the “Request Tab”. Each message has a parameter named “Need Confirm” which means whether the “confirm flag” needs to be checked.

Request Messages

-WiFi-Wifi On 0x00  
-WiFi-Wifi Off 0x01  
-WiFi-Wifi Join 0x02  
-WiFi-Wifi Leave 0x03  
-WiFi-Set Country Code 0x04  
-WiFi-Get Country Code 0x05  
-WiFi-Scan Wifi Networks 0x06  
-WiFi-Get AP Information 0x08  
-WiFi-Get Sta Status 0x09  
-WiFi-Set Roaming Control 0x0B  
-WiFi-Get Roaming Control 0x0C  
-WiFi-Set Wifi Band 0x0D  
-WiFi-Get Wifi Band 0x0E  
-WiFi-Set Wifi Sleep Parameter 0x0F  
-WiFi-Get Wifi Sleep Parameter 0x10  
-WiFi-Get Wifi Mac Address 0x11  
-WiFi-Setup AP Mode 0x12  
-WiFi-Get Soft Ap Clients 0x13  
WIFI


Message Sender

Request Response

NeedConfirm ☐  
Ssid  
Code Open  
Password  
Channel 0  
BssType Default  
Band Default  
Bssid 000000000000

Send

Figure 9 WIFI join Message Configuration

**Step-2:** After the message parameters configured, press the  **Send** button below the message parameters tab, the message bytes sequence will be generated and sent to the module through the selected serial port. The sent message will be displayed in the log list and the sending status will be displayed in status bar. As Figure 10 shows.

Message Log

☒ Auto Scrolling

Time Stamp	UART	Message Content
← 06-29-2016 11:42:47.580	Local port COM100	<Response-Success>0b 20 05 30 01 05 14 00 00
→ 06-30-2016 09:57:28.111	Local port COM100	<Request>0b 20 06 20 00 01 02 b6
← 06-30-2016 09:57:28.140	Local port COM100	<Response->0b 20 06 30 a9 01 02 07 41 6c 69 6e 6b 65 74 00 00 00 00 00 00 00 00 07 41 4c 58 43 31 32 42
→ 06-30-2016 09:57:34.294	Local port COM100	<Request>0b 20 07 20 00 02 00 b6
← 06-30-2016 09:57:35.735	Local port COM100	<Response-Success>0b 20 07 30 01 02 00 00 00
→ 07-04-2016 16:46:43.780	Local port COM4	<Request>0b 20 08 20 6c 02 02 04 74 65 73 74 00

Status: Sending and Waiting for Response...

OK

Figure 10 Message logs and Sending Status

**Step-3:** Once the Module finished processing the message and send the response message back, ACMTH will parse the messages and display the parsing result into the response Tab. Also the response message raw bytes sequence will be added as log into the log box. As Figure 11 shows.

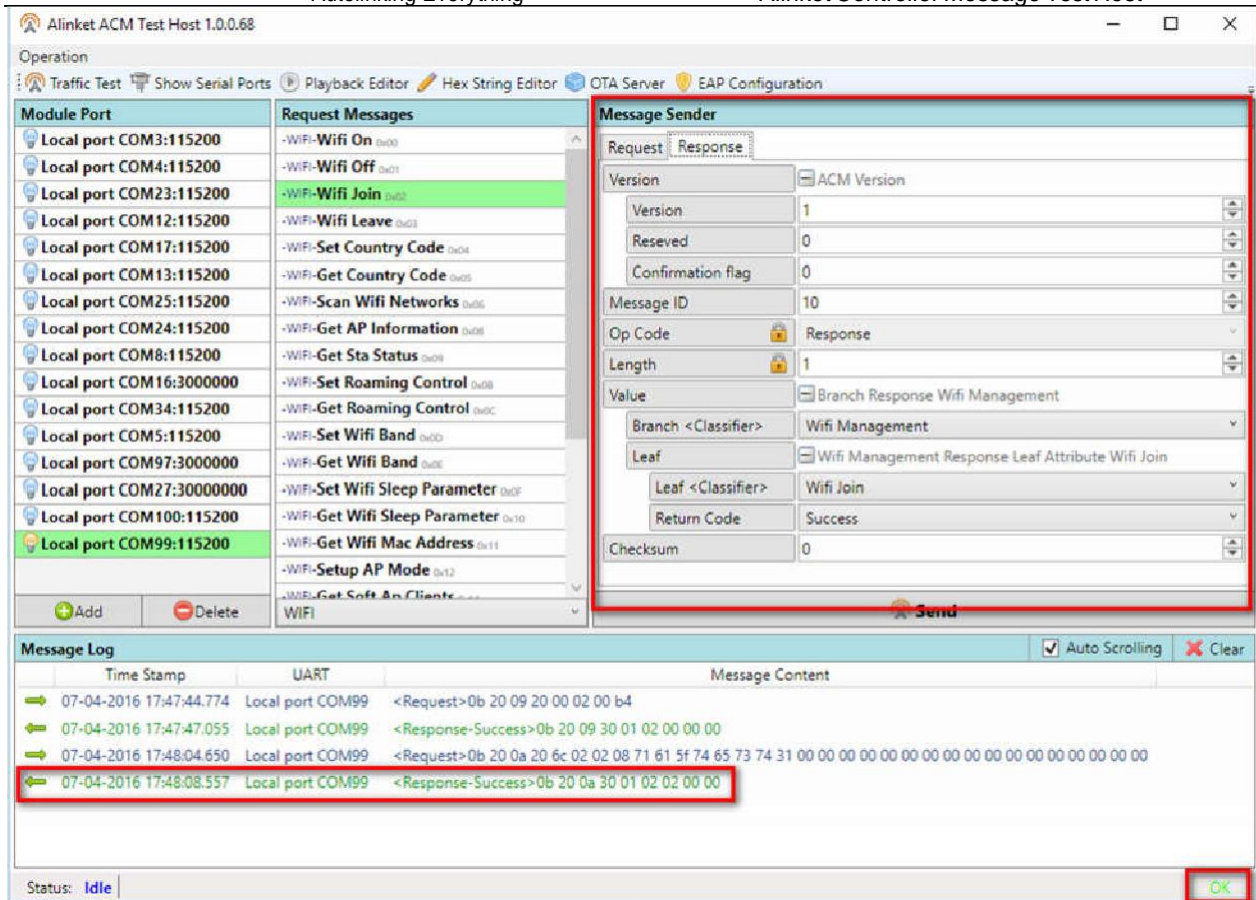


Figure 11 Response Message Parsing Result

### 3.2.2 Message Parsing

ACMTH parses the response message automatically and displays the result in the “response tab”. So once ACMTH receive the response from the module, the “response tab” will be active automatically and the response message will be displayed. As Figure 12 shows.

Message Sender

Request

Response

Version

ACM Version

Version

1

Reseved

0

Confirmation flag

0

Message ID

10

Op Code

Response

Length

1

Value

Branch Response Wifi Management

Branch <Classifier>

Wifi Management

Leaf

Wifi Management Response Leaf Attribute Wifi Join

Leaf <Classifier>

Wifi Join

Return Code

Success

Checksum

0

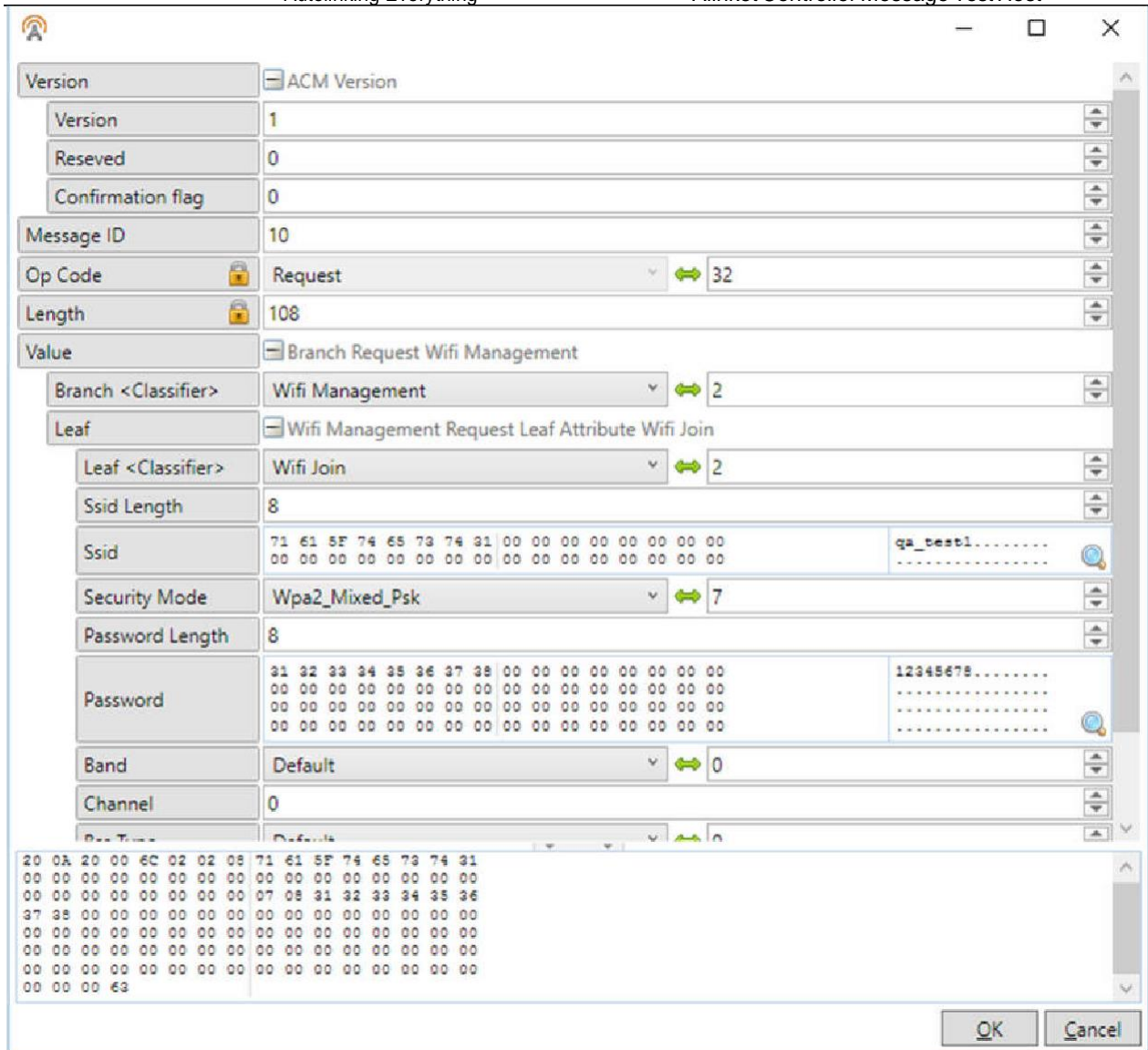
Send

Figure 12 Message Response Parsing Result

Actually for each message, no matter request, response, indication or confirmation message, you can get the parsing result for double clicking the message log in log box. There will be a new dialog window popped and shows the parsing result. As Figure 13 shows.

The Message Parsing result is separated into 2 parts: The message structure and the raw message bytes. The message structure use a friendly way to show each fields name and values of the message and the raw message bytes are displayed by a Hex Editor control\*.





The screenshot shows a software window titled "Alinket Controller Message Test Host". Inside, there is a form for configuring a "Request Message". The form is organized into sections with expandable/collapsible headers. The "Version" section includes fields for "Version" (1), "Reserved" (0), and "Confirmation flag" (0). The "Message ID" is set to 10. The "Op Code" is set to "Request" with a value of 32. The "Length" is set to 108. The "Value" section is expanded, showing "Branch Request Wifi Management". Under this, "Branch <Classifier>" is "Wifi Management" (value 2). The "Leaf" section is expanded, showing "Wifi Management Request Leaf Attribute Wifi Join". Under this, "Leaf <Classifier>" is "Wifi Join" (value 2). The "Ssid Length" is 8, and the "Ssid" field contains the hex string "71 61 5F 74 65 73 74 31" followed by zeros, with a text input "qa\_test1" visible. The "Security Mode" is "Wpa2\_Mixed\_Psk" (value 7). The "Password Length" is 8, and the "Password" field contains the hex string "31 32 33 34 35 36 37 38" followed by zeros, with a text input "12345678" visible. The "Band" is "Default" (value 0) and the "Channel" is 0. At the bottom, there is a "Raw Data" section showing a hex dump of the message bytes, starting with 20 0A 20 00 6C 02 02 08.

Figure 13 Request Message

### IMPORTANT!!

For the raw bytes in Hex Editor, it has some difference with the bytes received from the serial port.

- (1) The SOM (Start of Message) is not displayed, so in Figure 13, the first byte of the message is 0x20.
- (2) In ACM definition, the OP code and the upper byte of the length are in 1 bytes. For example: for the request message, the Op Code is 0001b, and the length 1000 is 0x03E8, so the 3rd and 4th (exclude the SOM) bytes of the message should be 0x13E8. But because of the limitation of the Hex Editor Control, the Op code and

upper byte of the length are all displayed as 1 byte, so the op code and message length of the message raw bytes in above example will be displayed as 0x1003E8. So in Figure 13, the message header should be 200A 20 6C 0202.

### 3.2.3 Message Log


The message log is automatically scrolling in default. User can disable the auto scrolling by unchecking the ☒ Auto Scrolling check box in header bar of the log box.

For clearing the logs, click the  button in header bar of the log box.

## 3.3 Traffic Test

The traffic test utility is a tool to define a set of traffic data and the tool can convert the traffic data to a groups of traffic related ACM messages and send to module.

### 3.3.1 Start Traffic Test Tool

After the serial port connected, right click on the port and the menu item “Traffic” will become to enable. Select the “Traffic” menu item and the Traffic Test Tool Window will be launched. Also user can launch the window by clicking the  button in Tool bar. Figure 14 shows the Traffic Test Window.

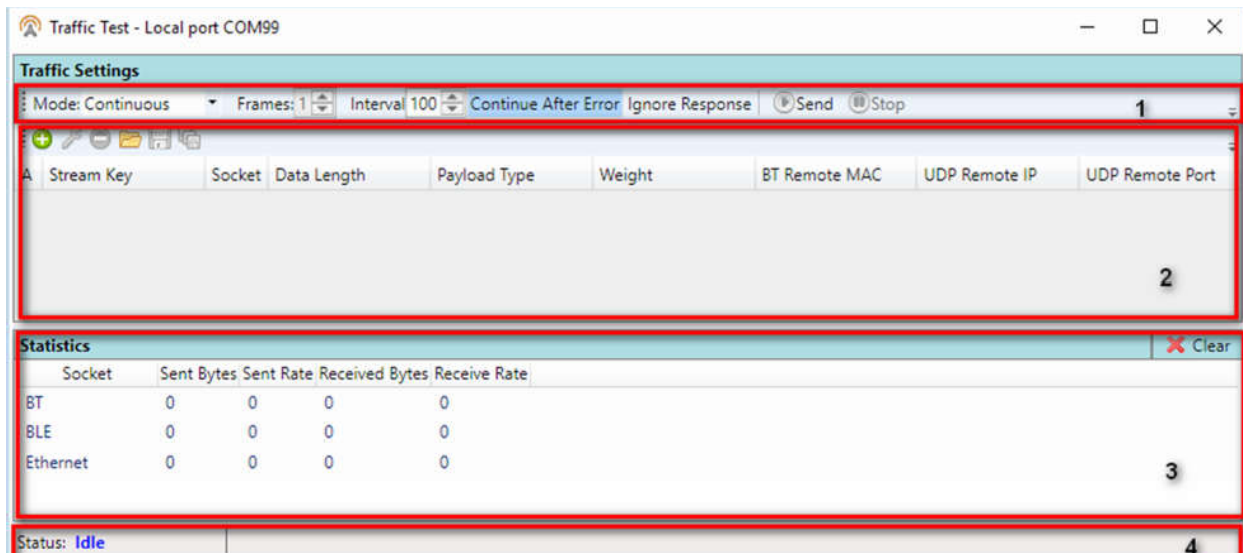


Figure 14 Traffic Test Window

The traffic test window can be launched on each module port, and the port name will be displayed on title.

- (1) Traffic Setting Bar: Set the traffic sending profile.
- (2) Stream Editor: Add, Modify and Delete the traffic streams.
- (3) Statistics: count the Sent and Received Bytes and Frames.
- (4) Status Bar: indicates the sending status.

Once the Traffic test window launched, the traffic related ACM will be suppressed in log box, this is because there will be huge amount of messages between ACMTH and module, the GUI based log might impact the performance. The "Suppress" means DO NOT display the traffic related messages in the log, that is, all messages like send data request or receive data indication WILL NOT be displayed in log any more. But in some case, for debugging purpose, user need to display these messages, user can disable the message suppressing by right clicking on module port and uncheck the "suppress traffic message" menu item. As Figure 15 shows.

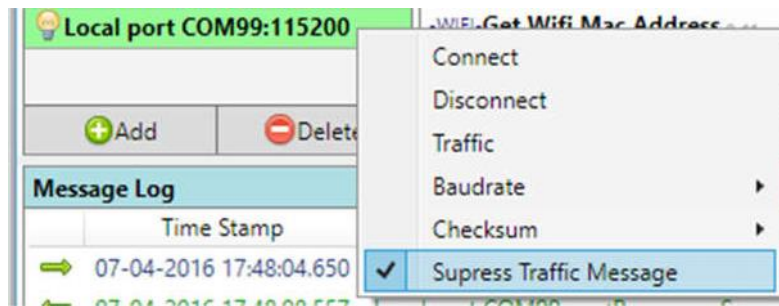


Figure 15 Message Suppressing Function

Table 3 lists the messages which is suppressed when the message suppressing function is enabled.

Table 3 Suppressed Messages


Branch	Leaf	Description
NETWORK(0x03)	0x16	Send TCP data
NETWORK(0x03)	0x1C	Send UDP data
NETWORK(0x03)	0x17	Received Data From TCP
NETWORK(0x03)	0x1D	Received Data From UDP
BT(0x04)	0x14	Send BT Data
BT(0x04)	0x15	Received Data From BT
BLE(0x05)	0x0C	Send BLE Data
BLE(0x05)	0x0D	Received Data From BLE

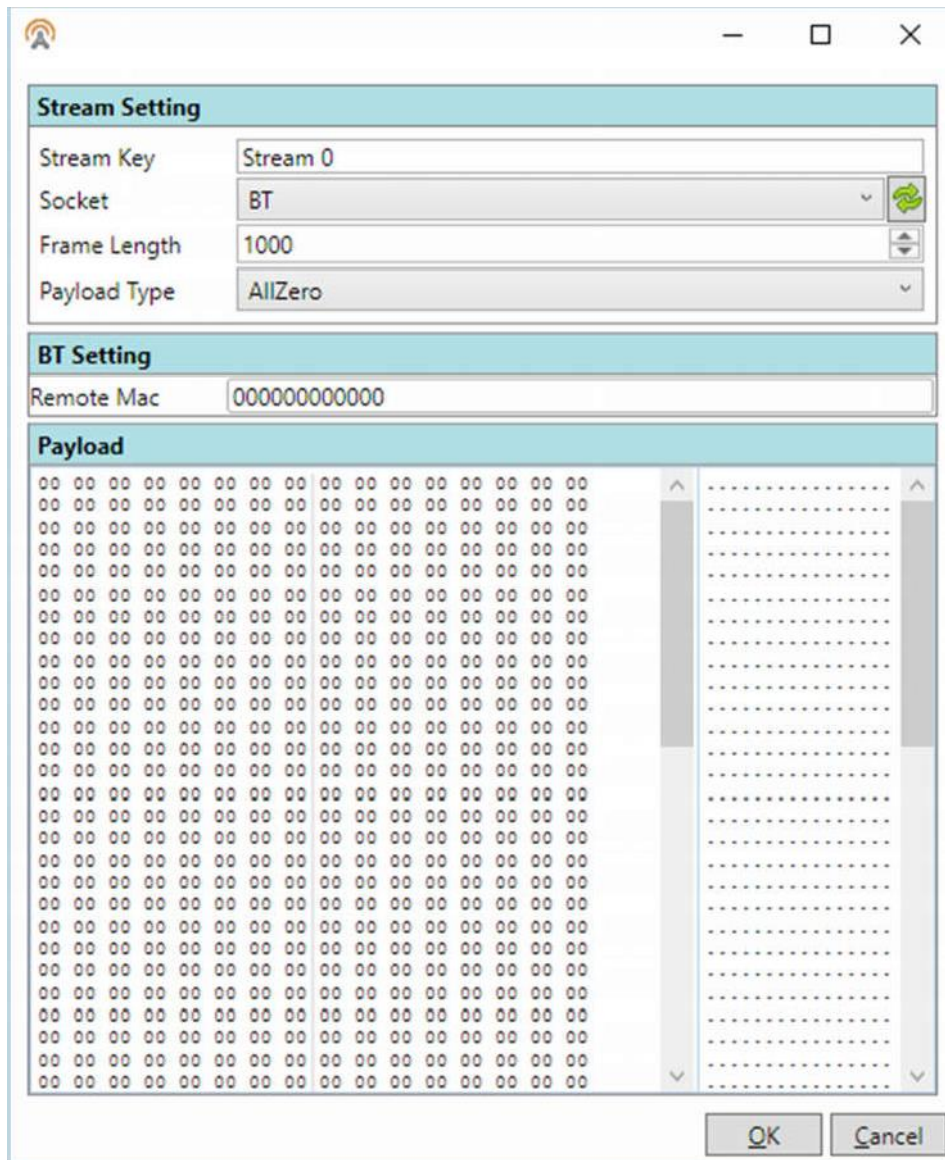


### 3.3.2 Traffic Stream

User need to define the traffic stream before sending the traffic.

#### Create Traffic Stream

**Step-1:** To create the traffic stream, click the  button in tool bar of the stream editor. The Stream Editor will be displayed, as Figure 16 shows.



The Stream Editor window is divided into three main sections:


- Stream Setting:** Contains fields for Stream Key (Stream 0), Socket (BT), Frame Length (1000), and Payload Type (AllZero).
- BT Setting:** Contains a Remote Mac field with the value 000000000000.
- Payload:** A large grid for editing the payload data, consisting of 20 columns and 20 rows of hex digits (00).

At the bottom right, there are OK and Cancel buttons.

Figure 16 Stream Editor

**Step-2:** Configure the Streams.

**Stream Key:** to Identity the stream, it is only used in ACMTH.

**Socket:** the Existing Socket on module, refreshing by clicking the  button. For WIFI module, user need to create the TCP or UDP at first, then after refreshing the socket, the created TCP or UDP socket will be displayed; For BT module, select BT; For BLE module, select BLE.

**Frame Length:** The payload data length.

**Payload Type:** There are 5 types of payload type: All Zero, Randomized Char, All 5, Randomized and Customized. The difference between Randomized Char and Randomized is the Randomized Char only generate the readable ASCII characters.

**Payload:** The preview of the payload or if the payload type is Customized, user can define the payload here.

If the socket type is BT/BLE, the BT/BLE settings need user to configure the remote MAC address.

If the socket type is UDP, the UDP settings need user to configure the remote IP and remote UDP Port.

**Step-3:** Click OK button to finish creating the new stream. As figure 17 shows.

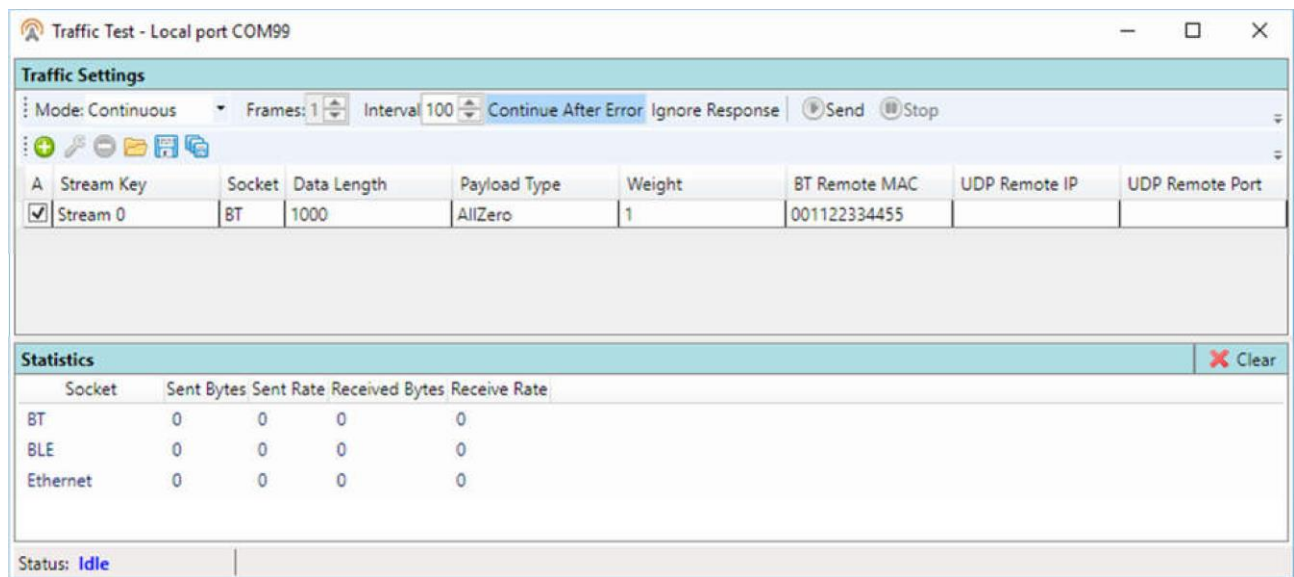




Figure 17 New Stream Added

## Edit Existed Stream

User can modify the existed streams by 2 way.

- (1) Some of the field can be modified directly in the Data Grid Cell by clicking the Cell.
- (2) Select the Stream and Click the  button, the Stream Editor window will be launched again.

## Remove Stream

User can remove the stream by selecting the stream and clicking the  button.

## Save and Load Stream

User can click  buttons to Load, Save and Save As the streams as a file with extension name “.asc”.

### 3.3.3 Send Traffic



After configuring the streams, user need to configure the traffic parameters before sending the traffic.

**Send Mode:** There are 2 modes can be selected, Continuous and Burst. The continuous mode allows user to send the continuous data till user stop the traffic manually (click the stop button). If user select the burst mode, the input box for Frame becomes available and user need to input the frame count.

**Interval:** the Interval between each Data ACM, the unit is millisecond. Be noticed that if the **Ignore Response** is checked, the interval is counted from each send data message to next send data message. Otherwise, the interval is counted from the last message’s response to next send data message.

**Continue After Error:** Whether the traffic goes on if the problem happens.

**Weight:** For each stream, there is a parameter named weight. This is used for multiple streams scheduler. It works just like WRR (Weighted Round Robin). For example, 3 streams have been defined and the weight for each stream is 3, 2 and 5. ACMTH will send the traffic ACM as follow sequence: 1 1 1 2 2 3 3 3 3. So in total sent bytes, data belongs to stream 1 should be 30%, data belongs to stream 2 should be 20% and 50% for stream 3.

User can click the  button to start the data sending and click the  button to stop the data sending any time.

### 3.3.4 Traffic Statistics

ACMTH collects the traffic statistics by each socket. Below statistic items are supported:

- SentBytes
- Send Rate
- Received Bytes
- Receive Rate

The statistic entries will be added when:

- (1) Traffic start sending.

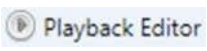
(2) Received traffic.

User can click the  button to clear the statistic counts.

## 3.4 Playback Editor

Playback Editor is used to record, save and replay the Alinket Controller Messages.

### 3.4.1 Launch Playback Editor

Select a module port and then click the  button in Toolbar, the playback editor will be launched. As Figure 18 shows.

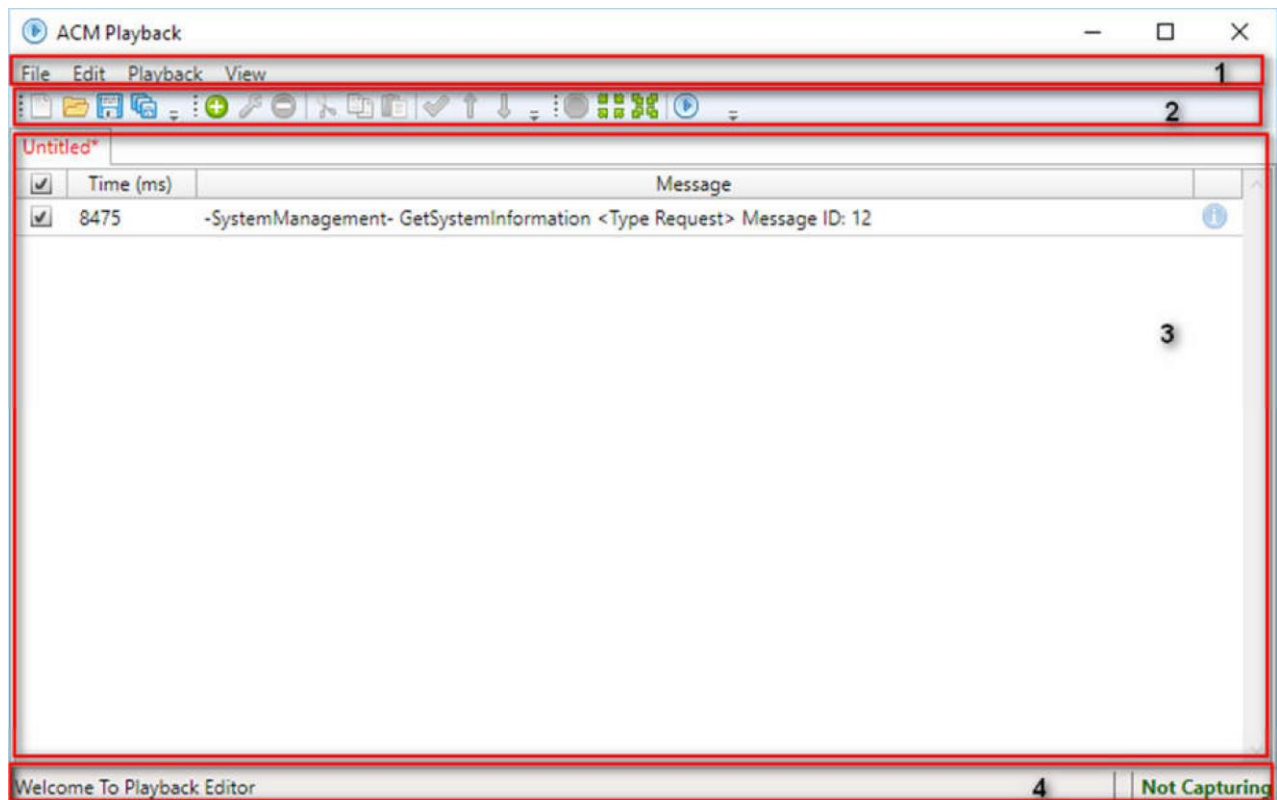





Figure 18 Playback Editor

- (1) Menu Bar
- (2) Tool Bar
- (3) Message Record List, multiple lists supported and each lists can be saved as a file.
- (4) Status bar.

### 3.4.2 Capture Messages


Click  or  button to start capture the messages. And then, any interaction between PC host and Alinket Controller Module will be recorded and displayed into the current selected message list. Click  button to stop capturing.


### 3.4.3 Message Modification

User can modify the recorded messages on:

- (1) Message content.
- (2) Message Order.
- (3) Add new messages
- (4) Remove messages.
- (5) Copy and Paste
- (6) Add Message Comments.
- (7) Save and Load.

#### Modify Message Content

Double click on the messages or select the message and click the  button, the message editor will be launched. As figure 19 shows.




Autolinking Everything      Alinket Controller Message Test Host

Version		ACM Version
Version	1	
Reserved	0	
Confirmation flag	0	
Message ID	12	
Op Code	Request	32
Length	0	
Value		Branch Request System Management
Branch <Classifier>	System Management	1
Leaf	System Management Request Leaf Attribute Get System Information	
Leaf <Classifier>	Get System Information	2
Checksum	176	


Figure 19 Message Editor

### Update Message Order

The Messages are ordered by the timestamp. By default, the message timestamp is the elapsed milliseconds from the first captured message. So for the first captured message, the timestamp is zero. User can adjust the timestamp to make the message re-ordered. Or user can select the message and click the  button to move the message up and down.

The timestamp is also used in message replay.

### Add New Message

Besides recording the messages, the playback editor also support add the message manually. User can click the  button to add a new message. After the button clicked, a message editor will be displayed and user can configure the message's branch and leaf and its parameters.


### Remove Message

User can remove the selected message by clicking the  button or by pressing the "delete" key on keyboard.

### Copy and Paste

Playback Editor also supports the copy and paste function just like in windows, Ctrl+C, Ctrl+X and Ctrl+V are supported.

### Add Comments

User can add the comment for each message by clicking the  icon at the right side of the message. After the icon is clicked, and text editor will be popped up and user can add the comment into the text box.

### Save and Load

Each Message list can be saved as a file which ext name is “.plb”.

## 3.4.4 Message Replay