

**CerberLink to Bluetooth Application Programming Interface
(API)
Version 1.4**

BriarTek, Inc.

Description

This document defines, for an application, the commands to and the data from CerberLink. Four mailboxes can be configured and associated with a Cerberus User ID (a 10-digit number). Only one mailbox is currently supported. A future version will support the multiple mailboxes.

Power

CerberLink is battery operated. A Mini-USB port is provided for charging the battery, and CerberLink will continue to run while charging.

Setup

All communications with CerberLink is through a Bluetooth connection. CerberLink supports SPP.

Limits

There is one shared Inbox, which has a limit of 4096 messages.
There are four user Outboxes, each with a limit of 128 messages.
There is one distress Outbox, which has a limit of 128 messages.
There are four breadcrumb boxes, each with a limit of 4096 breadcrumbs.

Command and Data formats

Commands and Data are ASCII text encoded as comma separate values (CSV). Each ends with a <CR><LF> with fields separated by commas. If these characters are used within a text message, they must be encoded during Bluetooth send and receive. All checksum calculations must be performed with the encoded values.

Sym	Decimal	Hex	<i>with</i>	Sym	Decimal	Hex
<CR>	13	0x0D		<DC1>	17	0x11
<LF>	10	0x0A		<DC2>	18	0x12
,	44	0x2C		<DC3>	19	0x13

Example:

Before encoding:

```
$INBOX,L,1,1,,P,1,8201,0,0,14,7,110203173723,17,Mike,Sue,,Hi, how are you<CR><LF>,*XX<CR><LF>
```

After encoding:

```
$INBOX,L,1,1,,P,1,8201,0,0,14,7,110203173723,17,Mike,Sue,,Hi<DC3> how are you<DC1><DC2>,*5F<CR><LF>
```

Time format

The time format will be YYMMDDHHMMSS. The year implies the 20 century. CerberLink end-of-life is the end of 2099.

Message types

The message types are **distress, distress cancel, personal, tracking, breadcrumbs (with or without text) and check-in (with and without text)**. Textcrumbs are location

messages with optional text that documents that moment and location. Breadcrumbs are location messages with no text.

The breadcrumb message can contain up to three breadcrumbs with the following format as the body (it must not exceed the 160 character limit)(see Text command \$CMD, Txt_BreadCrumbs for additional information):

C#datetime#latitude#longitude#altitude#C#datetime#latitude#longitude#altitude#C#datetime#latitude#longitude#altitude

Operation Modes

Overview

There are four ways to operate CerberLink; buttons, interactive, batch and auto.

Buttons

Several commands can be initiated by using the buttons on CerberLink; no application is necessary. The Buttonology field in the command description describes what to do. CerberLink uses its GPS time and location data to geotag messages as they are created.

Interactive

In Interactive Mode, the application is connected to CerberLink. Each command is processed as it finishes. Either CerberLink or the phone's GPS time and location data can be used to geotag messages as they are created.

Batch

In Batch Mode, the application is not connected to CerberLink, but creates and stores messages, textcrumbs and breadcrumbs to be synced with CerberLink at a later time. The application uses the phone's GPS time and location data to geotag messages as they are created.

Auto

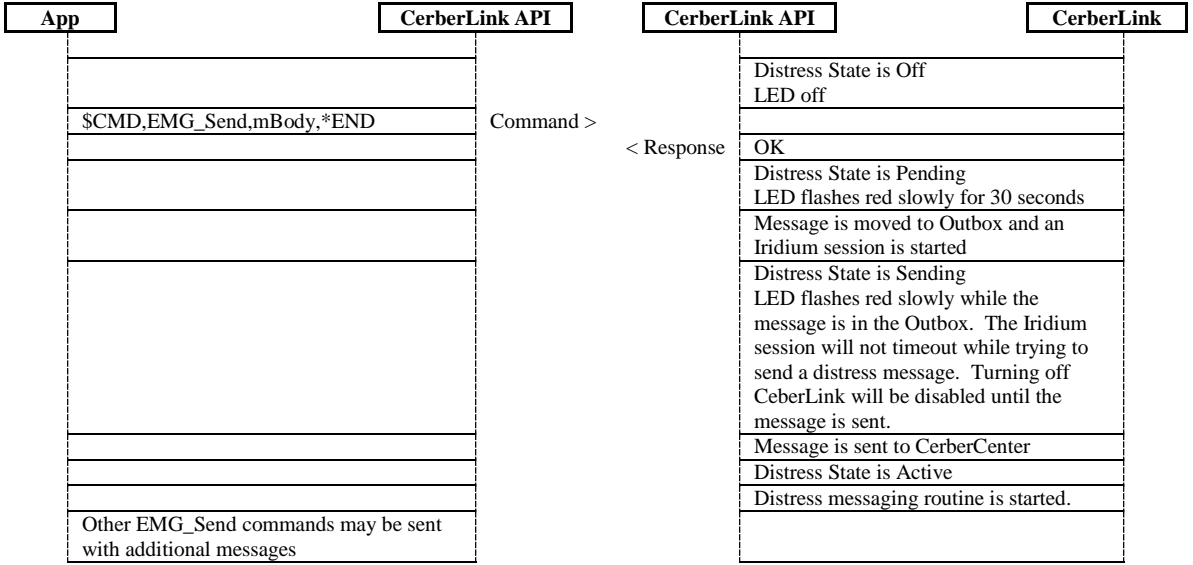
Breadcrumbs and mailbox checks can be configured to activate periodically. CerberLink uses its GPS time and location data to geotag messages as they are created.

Distress message connection example

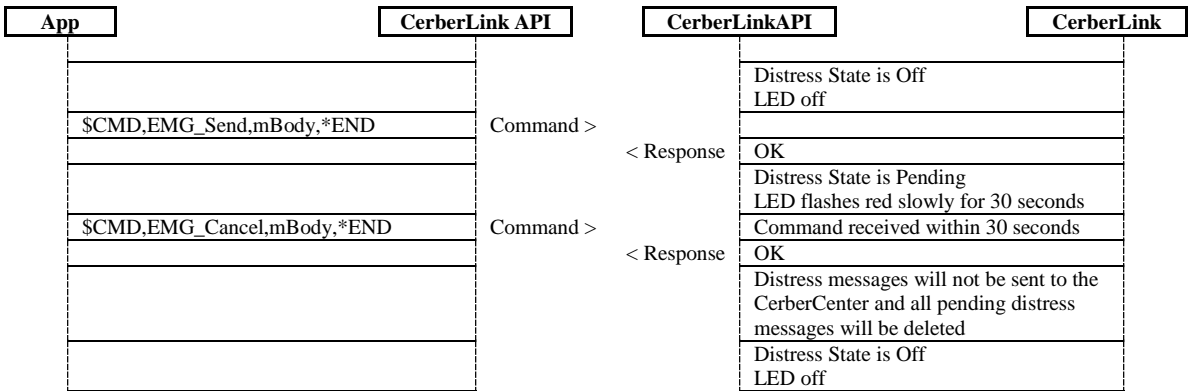
Starting a distress state can be initiated either by CerberLink buttons or by application command. An initiated distress can be terminated within 30 seconds either by CerberLink buttons or by application command. After 30 seconds, CerberLink will be in the distress state. Distress mode may be terminated by CerberCenter, CerberLink or by application. The current CerberLink distress state is updated in the \$STATUS data record. The \$STATUS data also has the current battery state. Once in distress state, further messages can be sent to the CerberCenter to update the situation.

CerberLink Distress State Name	CerberLink Distress State Code	CerberLink Distress LED	Description
Off	0	Off	
Pending	1	Flashes red slowly	Distress ON command received but not sent to the CerberCenter
Sending	2	Flashes red rapidly	Distress ON message waiting to sent
Active	3	Flashes red rapidly	Distress ON has been sent to the CerberCenter
Requesting	4	Flashes red slowly	Distress OFF message waiting to sent

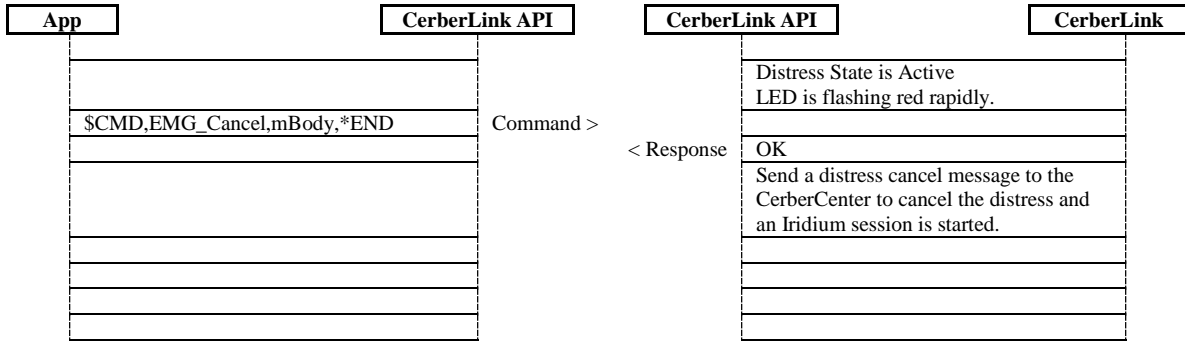
Initial the distress state



Cancel the distress message before it is sent



Cancel distress while in distress state



Commands

Overview

All commands are ASCII text encoded as comma separate values (CSV).

All commands start with “\$CMD,” and end with “,*END<CR><LF>”.

All commands will return <CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF> unless the Cerberus ID has not been entered in which case CerberLink will send the following unsolicited response: \$INFO,1,0,Enter CerberusID followed by ERROR

All commands will include all the commas.

All values must be specified unless listed as optional.

If any required values are left blank, the command response will be ERROR.

If the command response is OK, an optional data response may be sent.

If the command response is ERROR, an optional information response may be sent.

Command Details

Status Commands

Command	\$CMD,Status,mAction,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	\$STATUS or \$GPS or \$CRUMBOX or \$BUY or \$MAILBOX or \$SYSTEM
Iridium messages	None
Description	Request Status data mAction = 0: \$STATUS mAction = 1: \$GPS mAction = 2: \$CRUMBOX mAction = 3: \$BUY mAction = 4: \$MAILBOX mAction = 5: \$SYSTEM
Functional	Settings - Status
Firmware required	≥ 00.00.03.00
Buttonology	

Text Commands

All text commands can be sent with optional GPS time and location data. The mFlag is required and must be L, if optional GPS time and location data is to be used, or C, if CerberLink GPS time and location data is to be used. If mFlag is blank, the response will be ERROR. The format for mDT is YYMMDDHHMMSS. The format for mLat and mLong is degrees and fractional degrees. A maximum of 11 characters is needed for mLat and mLong. Six decimal places must be included, the leading - and zeros may be omitted, [-dd]d.ddddd. For example, mLat 38.834300 and mLong -77.059500.

Command	\$CMD,EMG_Send,mBody,mFlag,mDT,mLat,mLong,mAlt,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	Send a distress message, with mailbox check
Description	Send an emergency message with optional message. If the distress state is off, the first message will have a 30 second delay before it is sent.
Functional	Distress
Firmware required	≥ 00.00.03.00
Buttonology	1-2-3, with default message stating that the distress message was initiated from the buttons, with mailbox check.

Command	\$CMD,EMG_Cancel,mBody,mFlag,mDT,mLat,mLong,mAlt,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	Send request message, with mailbox check
Description	Request an emergency cancel with optional message. The analyst decides if the request can be canceled and only the analyst can turn off CerberLink's distress state. The user must periodically check for the response.
Functional	Distress
Firmware required	≥ 00.00.03.00
Buttonology	3-2-1, with default message stating that the cancel request was initiated from the buttons, with mailbox check.

Command	\$CMD,TXT_Send,mTo,mBody,mFlag,mDT,mLat,mLong,mAlt,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	Send a personal message, no mailbox check
Description	Send a personal text message (CerberLink will fill in the "from" based on the Bluetooth MAC)
Functional	Messaging
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,TXT_TextCrumb,mBody,mFlag,mDT,mLat,mLong,mAlt,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	Send a location message with optional text Buttons: with mailbox check Else: no mailbox check
Description	Send a GPS location message with optional message.
Functional	Messaging
Firmware required	≥ 00.00.03.00
Buttonology	2-2-2, with default message that is blank, with mailbox check.

Command	\$CMD, TXT_BreadCrumbs , mList, mFlag, mDT, mLat, mLong, mAlt, *END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	Send a list of location message without text, no mailbox check
Description	Send a list of GPS locations. mList may contain up to three breadcrumbs: C#datetime#latitude#longitude#altitude#C#datetime#latitude#longitude#altitude#C#datetime#latitude#longitude#altitude
Functional	Messaging
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD, TXT_CheckIn , mBody, mFlag, mDT, mLat, mLong, mAlt, *END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	Send a check-in message with optional text Buttons: with mailbox check Else: no mailbox check
Description	Send a message to check in at a predefined destination. The destination is configured on the server.
Functional	Messaging
Firmware required	≥ 00.00.03.00
Buttonology	1-1-1, with default message stating, "CerberLink check-in", with mailbox check.

Canned Message Commands

Command	<code>\$CMD,TRI_Set,2[123][123],Spare,Body,*END<CR><LF></code>
Command response	<code><CR><LF>OK<CR><LF></code> or <code><CR><LF>ERROR<CR><LF></code> followed by the command Example: <code>\$CMD,TRI_S,211,,This is a test.,*END<CR><LF></code>
Data response	
Iridium messages	None
Description	Canned message commands allow users to draft text messages in the application and map these messages to CerberLink button combinations. CerberLink supports setting values for any combination beginning with 2.
Functional	Canned message
Firmware required	≥ 00.00.03.08
Buttonology	211,212,213,221,223,231,232,233 are check-in messages 222 is a textcrumb message All canned message button trios are followed with a mailbox check.

Command	<code>\$CMD,TRI_Erase,*END<CR><LF></code>
Command response	<code><CR><LF>OK<CR><LF></code> or <code><CR><LF>ERROR<CR><LF></code>
Data response	
Iridium messages	None
Description	Erase all trio text commands. Erase individual canned message by mapping empty message to button trio.
Functional	Canned message
Firmware required	≥ 00.00.03.08
Buttonology	

Command	<code>\$CMD,TRI_List,*END<CR><LF></code>
Command response	<code><CR><LF>OK<CR><LF></code> or <code><CR><LF>ERROR<CR><LF></code>
Data response	
Iridium messages	None
Description	Initiates a listing of trio messages
Functional	Canned message
Firmware required	≥ 00.00.03.08
Buttonology	

Command	<code>\$CMD,TRI_Next,*END<CR><LF></code>
Command response	<code><CR><LF>OK<CR><LF></code> or <code><CR><LF>ERROR<CR><LF></code>
Data response	<code>\$TRIO</code>
Iridium messages	None
Description	Request the next trio map command. All trios are sent sequentially.
Functional	Canned messages
Firmware required	≥ 00.00.03.08
Buttonology	

Example of retrieving canned messages from the application to CerberLink:

Note--There is no 'ACK' for the tri-list.

```
APP: $CMD,TRI_List,*END
CL: OK

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,M,9,0,211,,On the road again,*5A

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,M,9,1,212,,At the hotel,*6E

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,M,9,2,213,,Time to eat,*04

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,M,9,3,221,,, *5A

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,M,9,4,222,,, *5E

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,M,9,5,223,,, *5E

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,M,9,6,231,,, *5E

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,M,9,7,232,,, *5C

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,L,9,8,233,,, *53

APP: $CMD,TRI_Next,*END
CL: OK
CL: $TRIO,D,9,9,,,, *68

APP: $CMD,TRI_Next,*END
CL: ERROR
```

*ERROR--If you send an extra TRI_Next message, CL will return an error.
*\$CMD,TRI_List lists all mapped and empty trio messages.

Breadcrumb Commands

Command	\$CMD,GPS_Auto,mAction,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Select whether GPS unsent breadcrumbs in flash memory are sent with every mailbox check, 0-Off, 1-On, 2-Realtime tracking (record waypoint and send by satellite).
Functional	Breadcrumb
Firmware required	\$CMD,GPS_Auto,0 must be ≥ 00.00.03.00 \$CMD,GPS_Auto,1 must be ≥ 00.00.03.00 \$CMD,GPS_Auto,2 must be ≥ 00.00.03.08
Buttonology	311-Real-time tracking OFF 312-Real-time tracking 10min interval 313-Real-time tracking 20min interval 322-Real-time tracking 1hr interval 323-Real-time tracking 20min interval

Command	\$CMD,GPS_Crumb,mInterval,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Collect GPS data every mInterval minutes and store in flash memory, 0-Off or 1- 1440.
Functional	Breadcrumb
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,GPS_EraseCrumbs,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Erase all breadcrumbs in flash memory.
Functional	Breadcrumb
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,GPS_List,mAction,mNumber,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Initiate a listing of stored GPS breadcrumbs. Use the GPS_Next command to receive the data. Any command, other than GPS_Next, will terminate the listing. The latest mNumber of breadcrumbs will be sent. If mNumber = 'A', then all the messages will be sent. mAction = 0: New breadcrumbs to server mAction = 2: New breadcrumbs to application
Functional	Breadcrumb
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,GPS_Next,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	\$CRUMBLIST (if requested)
Iridium messages	Creates TXT_BreadCrums messages, no mailbox check
Description	After the GPS_List command is sent, this command will request the next packet of GPS breadcrumbs. After all packets have been received, the next packet will be empty and the "Done" flag set. If this command is sent before GPS_List, the command response will be ERROR.
Functional	Breadcrumb
Firmware required	≥ 00.00.03.00
Buttonology	

Below is an example of syncing breadcrumbs from CerberLink to an application:

Note--There is no 'ACK' for the crumblist.

```

App:  $CMD,GPS_List,2,A,*END
CL:   OK

App:  $CMD,GPS_Next,*END
CL:   OK
CL:   $CRUMBLIST,M,5,0,,A,,110801223144,,38.834363,,-77.058536,,146.0,*44

App:  $CMD,GPS_Next,*END
CL:   OK
CL:   $CRUMBLIST,M,5,1,,A,,110801223653,,38.834184,,-77.059296,,52.8,*7F

App:  $CMD,GPS_Next,*END
CL:   OK
CL:   $CRUMBLIST,M,5,2,,A,,110801224152,,38.834261,,-77.059063,,62.4,*72

App:  $CMD,GPS_Next,*END
CL:   OK
CL:   $CRUMBLIST,M,5,3,,A,,110801224650,,38.834201,,-77.059144,,81.7,*7A

App:  $CMD,GPS_Next,*END
CL:   OK
CL:   $CRUMBLIST,L,5,4,,A,,110801225147,,38.834166,,-77.059168,,76.3,*7C

App:  $CMD,GPS_Next,*END
CL:   OK
CL:   $CRUMBLIST,D,5,5,,A,,,,,,,,,*4C

App:  $CMD,GPS_Next,*END
*CL:  ERROR

```

*ERROR--If you send an extra GPS_Next message, CL will return an error.

Iridium Commands

Command	\$CMD,IRD_Auto,mInterval,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	Send and Receive all available messages
Description	Start a mailbox check every mInterval minutes. If mInterval is 0, the auto mailbox check is turned off. The minimum value for mInterval is 1 minute. The maximum value for mInterval is 4320 minutes.
Functional	Iridium
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,IRD_Check,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	Send and Receive all available messages
Description	Start a mailbox check. The status command must be sent to watch the status. If a satellite is not in view, the session will timeout after 10 minutes.
Functional	Iridium
Firmware required	≥ 00.00.03.00
Buttonology	Mailbox Check Button, starts a mailbox session

Command	\$CMD,IRD_Reset,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Stop a mailbox check and power down the modem.
Functional	Iridium
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,IRD_Strength,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	\$INFO,33
Iridium messages	None
Description	<p>If CerberLink is not in session:</p> <ul style="list-style-type: none"> power up the modem read the RSSI value once and update the RSSI LED. This could take up to 60 seconds. power down the modem <p>The RSSI value is also located in the status data.</p>
Functional	Iridium
Firmware required	≥ 00.00.03.00
Buttonology	RSSI Button

Mailbox Commands

Future Use – a mailbox number will be associated with a Bluetooth MAC, which will be associated with a Cerberus ID. Only messages for that user will be affected.

All mailbox commands have a mMB parameter, the values are:

mMB = 0: Inbox

mMB = 1: Outbox

Command	\$CMD,MBX_Erase,mMB,mAction,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	mAction = 3: Erase all messages
Functional	Mailbox
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,MBX_List,mMB,mAction,mBodyLen,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	This command initiates listing a mailbox. After this command responds OK, messages will be received in response to the MBX_Next command. Only one message will be received at a time. After all messages are received, the "Done" flag will be set. A message with the "Done" flag set will be empty. Any command, other than MBX_Next, will terminate the listing. mAction = 0: List all new messages Include mBodyLen characters of the body (0 for none [default] and -1 for the whole message)
Functional	Mailbox
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,MBX_Next,mMB,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	\$INBOX or \$OUTBOX
Iridium messages	None
Description	After the MBX_List command is sent, this command will request the next message. After all messages have been received, the next message will be empty and the "Done" flag set. If this command is sent before MBX_List, the command response will be ERROR.
Functional	Mailbox
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,MBX_Ack,mMB,mAction,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	\$INBOX or \$OUTBOX when resending a message
Iridium messages	None
Description	mAction 0: Message received, flag as read mAction 1: Message received, do not mark as read mAction 2: Message received, resend same message mAction 3: Message received, flag as deleted, there is a problem
Functional	Mailbox
Firmware required	≥ 00.00.03.00
Buttonology	

Message sync: Below is an example of proper message sync sequence.

```
APP:  $CMD,MBX_List,0,0,-1,*END
CL:   OK

App:  $CMD,MBX_Next,0,*END
CL:   OK
CL:   $INBOX,M,4,0,,P,3,8201,11552,0,4,4,110801220540,160,,abc@123.com,,[#1002 1/4] Test
      message at 110801220540_ Test One,*4B

App:  $CMD,MBX_Ack,0,0,*END
CL:   OK

App:  $CMD,MBX_Next,0,*END
CL:   OK
CL:   $INBOX,M,4,1,,P,3,8201,11552,0,5,5,110801220540,160,,abc@123.com,,[#1002 2/4] Test
      message at 110801220540_ Test Two,*41

App:  $CMD,MBX_Ack,0,0,*END
CL:   OK

APP:  $CMD,MBX_Next,0,*END
CL:   OK
CL:   $INBOX,M,4,2,,P,3,8201,11552,0,6,6,110801220540,160,,abc@123.com,,[#1002 3/4] Test
      message at 110801220540_ Test Three,*41

App:  $CMD,MBX_Ack,0,0,*END
CL:   OK

App:  $CMD,MBX_Next,0,*END
CL:   OK
CL:   $INBOX,L,4,3,,P,3,8201,11552,0,7,7,110801220540,160,,abc@123.com,,[#1002 4/4] Test
      message at 110801220540_ Test Four,*26

App:  $CMD,MBX_Ack,0,0,*END
CL:   OK

App:  $CMD,MBX_Next,0,*END
CL:   OK
CL:   $INBOX,D,4,4,,0,3,0,0,0,65535,0,,0,,,,*15--message four is an empty message

App:  $CMD,MBX_Ack,0,0,*END
CL:   OK

App:  $CMD,MBX_Next,0,*END
*CL:  Error
```

*ERROR--If you send an extra MBX_Next message, CL will return an error

System Commands

Command	\$CMD, SYS_Power ,mAction,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	System Power (like the power button) Currently, only a value of 0 (Off) is supported. All current operations (e.g. Iridium session) must end before the unit will turn off. To turn CerberLink off, it must be running on battery, the USB must not be connected.
Functional	System
Firmware required	≥ 00.00.03.00
Buttonology	Power Button (toggle)

Command	\$CMD, SYS_Reboot ,mAction,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	System Reboot mAction must be 0
Functional	System
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD, SYS_Test ,mAction,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	System Self-Test mAction must be 0
Functional	System
Firmware required	≥ 00.00.03.00
Buttonology	3-3-3, if the test fails, all LEDs flash and CerberLink cannot send satellite messages.

Settings Commands

Command	\$CMD,SET_Erase,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Erase ALL system settings, user messages and breadcrumbs. It then sets the default factory settings. This could be used to reset a leased unit.
Functional	Settings - System Configuration
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,SET_Notify,mAction,mState,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Send an \$INFO packet to alert the app so the app does not have to keep checking CerberLink mState = 0 OFF and 1 ON mAction = 1 New Mail Notify - There is new mail to be read.
Functional	Settings - System Configuration
Firmware required	≥ 00.00.03.00
Buttonology	

Command	None
Command response	None
Data response	None
Iridium messages	None
Description	Hard reset of microprocessor should a fatal error occur
Functional	Settings - System Configuration
Firmware required	≥ 00.00.03.00
Buttonology	POWER - 1 - 3 for 5s

Command	\$CMD,UID_Set,mID,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Associates Cerberus User ID with application's Bluetooth Mac address. The Cerberus ID must be registered with the server. This value and the modem IMEI number will be used as the user's address in sending and receiving messages and used for billing purposes. When the pairing is removed, this association is deleted.
Functional	Settings - User
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,UID_Delete,mID,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Remove Cerberus User ID and all associated configurations and messages from the CerberLink. A new user can be assigned to the empty mailbox.
Functional	Settings - User
Firmware required	≥ 00.00.03.00
Buttonology	

Command	\$CMD,BLT_RemovePairings,mPair,*END<CR><LF>
Command response	<CR><LF>OK<CR><LF> or <CR><LF>ERROR<CR><LF>
Data response	
Iridium messages	None
Description	Remove Bluetooth pairings. If mPair is 0, then all pairing will be removed.
Functional	Settings - Bluetooth
Firmware required	≥ 00.00.03.00
Buttonology	

Data

Overview

Most data transmitted from CerberLink will be in response to a command. Operational commands may come from the server and an information response may be sent to the application. See the CerberCenter API for additional information.

Data packets will be ASCII text encoded as comma separate values (CSV) and viewable with a terminal emulator.

Data packets will begin with a "\$" and end with ";*XX<CR><LF>

The checksum follows ",*" and is the last two characters not including the <CR><LF>.

It consists of two hex digits representing the exclusive or (XOR) of all characters between, but not including, the \$ and * characters.

Data Summary

\$STATUS,data,*checksum<CR><LF>

\$SYSTEM,data,*checksum<CR><LF>

\$MAILBOX,data,*checksum<CR><LF>

\$INBOX,data,*checksum<CR><LF>

\$OUTBOX,data,*checksum<CR><LF>

\$TRIO,done,total,count,[123][123][123],Spare,Body,*checksum<CR><LF>

\$GPS,data,*checksum<CR><LF>

\$CRUMBLIST,data,*checksum<CR><LF>

\$CRUMBBOX,data,*checksum<CR><LF>

\$INFO,data,*checksum<CR><LF>

Data Details

\$STATUS

Response to command:

*\$CMD,Status,0,*END<CR><LF>*

Name	Example	Description
Message ID	\$STATUS	Protocol header
SYSTEM DistressState		0=off and 1=on
SYSTEM BatteryState		1=Battery Discharging; 2=Battery Low; 3&7=Battery Fault; 5=Battery Charging; 6=Battery Full
SYSTEM BatteryVoltage		XXXX in mV
SYSTEM Battery Level		0-100
SYSTEM Battery Temperature		XXXX in °C
SYSTEM PowerUpReason		empty
SYSTEM Cerberus ID valid		0
SYSTEM Locked		0=Unlocked 1=UsersLocked (Emergency and Alerts are allowed) 2=Locked (No messages will be allowed)
GPS RFPWRUP		empty
Breadcrumb AutoSend		(0 off, 1 on, 2 real-time tracking)
Breadcrumb Interval		Interval in minutes
IRIDIUM IsPowered		0=no and 1=yes
IRIDIUM NetworkAvailable		(blank, 0 or 1)
IRIDIUM RSSI		(blank, 0 - 5)
IRIDIUM InSession		(blank, 0 no, 1 waiting for satellite or 2 trying to send)
IRIDIUM InSessionStatus		Empty
IRIDIUM InSessionProgressBar		Empty
IRIDIUM InSessionTimeout		(minutes left until session ends due to no network)
IRIDIUM InSessionTimer		(minutes in session)
IRIDIUM OutSessionTimer		(minutes since last session ended)
IRIDIUM AutoCheck		(0 off, minutes) Send GPS information and check for mail
IRIDIUM FatalError		Empty
INBOX NumberOfSlots		The total number that can be received from Iridium
INBOX NumberOfOpenSlots		The number slots free to receive messages from Iridium
INBOX NumberToReceive		Reported by Iridium session, number queued at Iridium
Checksum	*XX	
<CF><LF>		

\$SYSTEM

Response to command:

*\$CMD, Status,5, *END<CF><LF>*

Name	Example	Description
Message ID	\$SYSTEM	Protocol header
Serial Number		
Firmware compile time		
Firmware ID		xx.xx.xx.xx
Hardware ID		xx.xx.xx.xx
IRIDIUM IMEI		xxxxxxxxxxxxxxxx
Cerberus ID	2025551234	10 digit number. If this value is blank or not valid, the application must request the user to enter it. This value is used as the users' address in sending and receiving messages and for billing purposes.
CerberLink's Bluetooth MAC		xx.xx.xx.xx.xx.xx
Bluetooth Passkey	1234	
Number of Bluetooth Pairs		
Bluetooth Pair 1		
Etc.		
Checksum	*XX	
<CR><LF>		

\$MAILBOX

Data depends on which user is logged in.

Response to command:

*\$CMD, Status,4, *END<CR><LF>*

Name	Example	Description
Message ID	\$MAILBOX	Protocol header
Mailbox ID	0	Which user_(0-3). User 0 only supported.
INBOX queue size	4096	
INBOX number to read	8	New messages
INBOX full	0	0=no or 1=yes
OUTBOX queue size	128	
OUTBOX number to send	2	Messages not sent
OUTBOX full	0	0=no or 1=yes
Checksum	*XX	
<CR><LF>		

\$INBOX and \$OUTBOX

Response to command:

*\$CMD,MBX_Next,mMB,*END<CR><LF>*

*\$CMD,MBX_Message,mMB,mIdx,*END<CR><LF>*

Example:

Before encoding:

\$INBOX,L,1,1,,P,1,8201,0,0,14,7,110203173723,17,Mike,Sue,,Hi, how are you<CR><LF>,*XX<CR><LF>

After encoding:

\$INBOX,L,1,1,,P,1,8201,0,0,14,7,110203173723,17,Mike,Sue,,Hi<DC3> how are you<DC1><DC2>,*5F<CR><LF>

Name	Example	Description
Message ID	\$INBOX	Protocol header
Done		D=done, L=last, M=more
Total messages		Number of messages requested
Message Count		Number sent
Spare		Blank - Future use, not used
Header		D=distress, A=alert, P=personal or 0=empty or deleted
Mailbox ID		Which user (1 to 4)
Flags		empty
Distress1		empty
Distress2		empty
Index		empty
Order		Order received from Iridium
Time		Time received by CerberLink - yymmddhhmmss
Length		Length of Body in bytes, control characters count as one
To		47 characters max
From		47 characters max
Subject		Blank - Future use, not used
Body		160 characters max
Checksum	*XX	
<CR><LF>		

\$TRIO

Response to command:

*\$CMD,TRI_Next,*END<CR><LF>*

Example:

\$TRIO,M,27,9,211,,This is a test.,*2A<CR><LF>

*If this is a response to the "Set" command, the done, total and count fields will be blank.

Name	Example	Description
Message ID	\$TRIO	Protocol header
Done		D=done, L=last, M=more
Total messages		Number of messages requested
Message Count		Number sent
Trio		[123][123][123]
Spare		empty
Body		Canned message body
Checksum	*XX	
<CR><LF>		

\$GPS

Response to command:

*\$CMD,Status,1,*END<CR><LF>*

Name	Example	Description
Message ID	\$GPS	Protocol header
UTC Date		ddmmyy
UTC Time		hhmmss
Latitude		dd.xxxxxx
N/S Indicator		empty
Longitude		-ddd.xxxxxx
E/W Indicator		empty
MSL Altitude		xxxxx.xx in meters
Speed over ground		knots
Course over ground		degrees
Checksum	*XX	
<CR><LF>		

\$CRUMBLIST

Response to command:

*\$CMD,GPS_Next,*END<CR><LF>*

Name	Example	Description
Message ID	\$CRUMBLIST	Protocol header
Done		D=done, L=last, M=more
Total messages		Number of messages requested
Message Count		Number sent
Spare		empty
Source		empty
Flags		empty
UTC Date		ddmmyy
UTC Time		hhmmss
Latitude		dd.xxxxxx
N/S Indicator		empty
Longitude		-ddd.xxxxxx
E/W Indicator		empty
MSL Altitude		xxxxx.xx in meters
Checksum	*XX	
<CR><LF>		

\$CRUMBBOX

Response to command:

*\$CMD,Status,2,*END<CR><LF>*

Name	Example	Description
Message ID	\$CRUMBBOX	Protocol header
Crumb queue size	4096	Number of breadcrumbs that can be stored before old breadcrumbs are over written
Number not sent to app		
Number not sent to server		
Number stored		
Checksum	*XX	
<CR><LF>		

\$INFO

Response to command:

Event driven

Name	Example	Description
Message ID	\$INFO	Protocol header
Code		Optional Numeric Code (0 through 65535)
Sub-Code		
Text		Optional Text message (string)
Checksum	*XX	
<CR><LF>		

Code	Sub-Code	Description
0	0	OK
1	0	Cerberus ID is not valid or missing
2	0	Cerberus ID has been rejected by the server
5	0	Event driven text response by CerberLink that application may display to the user.
8	0	Iridium Error
9	From modem	Iridium Error, Iridium error code will be set as the sub-code
24	response	Send to Outbox response, 0 accepted, 1 error
32	Number of new msgs	Push notify, new messages are ready to be read
33	Iridium RSSI	Push notify, the Iridium RSSI has changed