ClimateTalk 2.0
CT-485 Application Protocol Interface Reference

Document revision: 01
Release: June 12, 2013

Abstract

ClimateTalk is a universal language for innovative, cost-effective solutions that optimize performance, efficiency and home comfort. The ClimateTalk Open Standards define a set of messages and commands to enable interoperability, enhanced user interface, and machine to machine control independent of the physical layer connecting the devices.

This document maps each CT-CIM service to a Message Type Number and defines the CT-485 packet format for that Message Type. Corresponding to OSI Layer 6, the CT-485 API reference defines the packet format for CT-485-specific messages used for ClimateTalk applications implemented over CT-485.

Copyright @ 2013 by the ClimateTalk Alliance
2400 Camino Ramon
Suite 375
San Ramon, CA 94583 USA

This document and the information contained herein are provided on an “AS IS” basis and ClimateTalk Alliance DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITH-OUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT. IN NO EVENT WILL CLIMATETALK BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. All Company, brand and product names may be trademarks that are the sole property of their respective owners.

This document is subject to change without notice.

Updates

This specification may be updated at any time and may be superseded by a more recent version or amended to from time to time. Users should be certain they are using the current ClimateTalk version and the latest revision of the documents.

The released versions of all specifications are available at http://www.ClimateTalk.org

Version History

ClimateTalk Version	Document Revision	Release Date	Comments
V 0.9		2008-11-07	Pre-Release
V 1.0		2009-08-24	Initial Release
V 1.1		2011-06-23	Message Types and Control Commands for Motor Profile added.
V 1.3		2011-11-02	Additional Errata Updates, Revised Formatting
V 2.0	00	2013-01-18	Version 2.0 Release – Added exceptions to Node Discovery and Set Address messages to support CT1.0 devices.
V 2.0	01	2013-06-12	Updated 5.14 Set Network Node List – Message Type 0x14 to explicitly state the entire node list must be checked for active nodes. Added note indicating CT1.0 devices must only be sent a node list with a maximum data payload equal to 16 bytes. Added 5.15 Direct Memory Access (DMA) Read – Message Type 0x1D. This was in CT1.0 but accidentally omitted from CT2.0.

NOTE: This document was split from the Command Reference in order to complete the separation of protocol and application. Revisions to CT 1.0 relate to the Command Reference, which was the basis of this document.

Contributors

The following is a list of ClimateTalk Alliance member companies that were actively engaged in the development of this standard:

A.O. Smith Water Products Company

ecobee inc.

EDC

Emerson Electric, Co.

EWC Controls, Inc.

Microchip Technologies, Inc.

Nogginhaus, LLC.

Research Products Corp.

Rheem Manufacturing Company

Zonefirst

Table of Contents

List of Tables

1.0 Overview

1.1 Purpose

ClimateTalk is an open standard that defines a set of messages and commands to enable interoperability, enhanced user interface, and machine to machine control independent of the physical layer connecting the devices.

The messages and commands defined by ClimateTalk Information Model (CIM) are the presentation and application layers as defined by the OSI Model[1]. ClimateTalk Applications are fully defined at Layer 7 of the OSI model by a combination of a Device Specific Application Profile, the Generic Application Specification and the Command Reference.

ClimateTalk messages can be carried over any physical medium following the OSI model. The ClimateTalk Presentation Layer defines how messages are executed over the various physical mediums in use.

CT-485 and CT-LWP are wired serial physical and network layers designed to support the formation of ClimateTalk networks and transport ClimateTalk messages, but other OSI based protocols – including wireless transports - can be used as well.

1.2 Scope

CT-485 is a Physical, Data Link, and Networking set of specifications that define one of the physical media over which ClimateTalk messages are sent. CT-485 is a variant of EIA/TIA-485[2] standards with provisions against incorrect wiring and grounding requirements that meet the needs of residential systems.

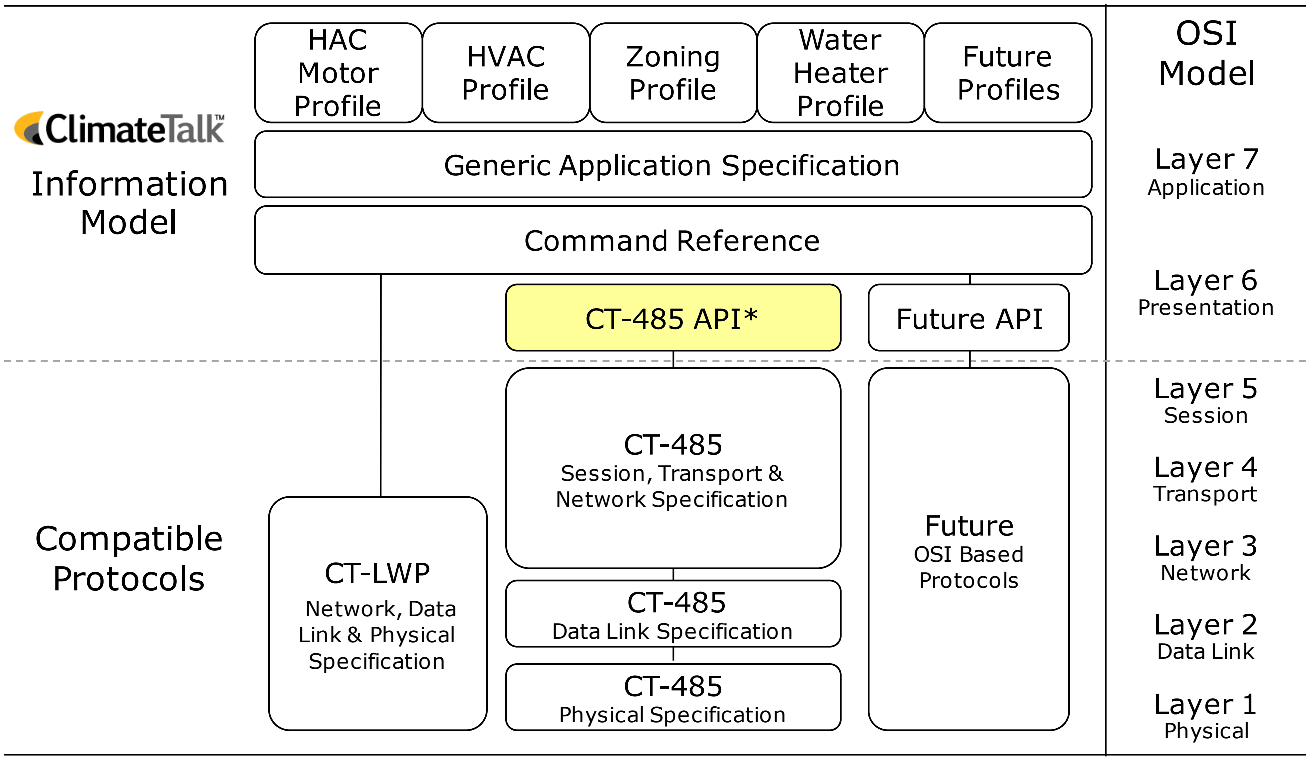
The scope of this document is twofold:

1. It maps each CT-CIM service to a Message Type Number and defines the CT-485 packet format for that Message Type
2. It details the packet format for CT-485-specific messages used to maintain and provide a communicating world to CT-CIM applications

[1] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=20269

[2] http://standardsdocuments.tiaonline.org/tia-tsb-89-a.htm

Figure 1- OSI Layers for a CT-485 Implementation



**This Document*

The ClimateTalk Open Standards package shown in Figure 1- OSI Layers for a CT-485 Implementation prescribes the requirements to ensure proper network formation of interoperable devices. Each device must comply with the mandatory requirements defined in this document as well as all other ClimateTalk standards applicable to the device functionality.

Membership in the ClimateTalk Alliance as well as successful completion of conformance testing is required for listing a product as a ClimateTalk Certified Device.

2.0 Normative References

A good understanding of the most recent version of the following documents is required to apply the contents of this specification correctly.

ClimateTalk Generic Application Specification

ClimateTalk Command Reference

ClimateTalk CT-485 Networking Specification

ClimateTalk CT-485 Data Link Specification

ClimateTalk CT-485 Physical Standard

3.0 Terminology

3.1 Definitions

ACK
Message to reflect proper receipt and ability to handle request in the data payload

Alpha-numeric ASCII
Characters 0x20 to 0x7e of the ASCII character set

Control ID
Unique 2-byte Identifier assigned to each device by the ClimateTalk Alliance that is used to identify the control application

HVAC Subsystem Nodes
Any subsystem involved in the control of HVAC operation is considered a HVAC Subsystem Node. Examples: Thermostat, Furnace, Air Handler, Air Conditioner, and Heat Pump.

Manufacturer ID
Unique 2-byte Identifier assigned to each manufacturer by the ClimateTalk Alliance that is used to identify the subsystem

NAK
Message to reflect a misunderstanding or failure to handle the request in the data payload

Profile
Set of rules governing the implementation of certain aspects of the protocol, which includes timings and communication rules to function properly

3.2 Acronyms

CT	ClimateTalk
DBID	Database Identification
IEEE	Institute of Electrical and Electronics Engineers
MDI	Message Data Interface
RPM	Revolutions Per Minute

3.3 Number Notation

Hexadecimal values shall be preceded by "0x," e.g. 0x11, 0x0f.
Numbers written with no precedent shall be assumed to be decimal, e.g. 17, 9.

3.4 Word Usage

The conventions used in this document are modeled after the definitions of the *2009 IEEE Standards Style Manual*. The *IEEE Standards Style Manual* can be downloaded from https://development.standards.ieee.org/myproject/Public/mytools/draft/styleman.pdf.

can Equivalent to *is able to* or *is capable of*.

may Equivalent to *is permitted to* or *is allowed to*. The use of *may* means that something is optional and does not imply a requirement.

must Used to describe situations where no other course of action is possible.

shall Equivalent to *is required to*. Use of the word *shall* means that the specification shall be implemented exactly as described in order to ensure correct operation and interoperability with other devices.

should Equivalent to *is recommended that*. This is used in situations where there are several possible options, but one option is preferable to the others.

4.0 Message Rules

4.1 Frame Format

The Packet Header and Packet Payload (shown in blue in Table 1 below) are discussed in this document. The Message Header and Message Footer vary depending on the type of protocol carrying the ClimateTalk messages. The *Generic Application Specification* discusses transmission rules and defines the Packet Header more fully. The *Command Reference* defines the structure of the payload for each Message Type.

Table 1 – CT-485 Message Structure Example

Element	Segment	Size (in bytes)
Addressing and Routing	Message Header	Variable
Message Type	Packet Header	1
Packet Number		1
Payload Length		1
Packet Payload	Packet Payload	0-240
Message Checksum	Message Footer	Variable

4.2 Rules for Message Structure

This section defines the structure of messages. Future messages shall follow these rules to maintain backwards compatibility:

1. Message Types that query devices for a block of data (e.g. Get Status) have the following rule:
    a. Responses shall have payloads of at least two bytes. Responses with payloads of less than two bytes are not allowed. This rule does not apply if DB IDs are utilized.
2. “Set” Message Types or Message Types that are similar to Control Commands have the following rules:
    a. Request data payload must be less than or equal to two hundred and forty bytes.
    b. Responses either ECHO back the request data payload or send a confirmation code. The appropriate response is defined for each command.
3. The above rules do not apply to the R2R Message Type.
4. All response messages shall use the same Send Method and parameters as the original request message.

4.3 Packet Number

The packet number is defined as follows:

Bit 7	Bit 6	Bit 5	Bits 4 - 0
Dataflow	Reserved	Version	Chunk number
1/0	0	1/0	0

Bit 7, the Dataflow bit, is a '1' (one) on all Dataflow packets i.e. all R2Rs and ACKs.

Bit 5, the Version bit, was a '1' in CT-485 V 1.0 to indicate all CT-485 1.0 devices. CT-485 V 2.0 shall use a value of '0' (zero) for the Version. A CT2.0 coordinator shall set the Version bit to a '1' when sending Node Discovery Request messages. A CT2.0 subordinate shall reply with its version bit clear, '0', indicating it is a CT2.0 device. A CT1.0 device shall reply with its version bit set indicating it is a CT1.0 device. The setting of the version bit by the coordinator is required to obtain the expected node discovery response from some older CT1.0 devices which echoed the version bit during the addressing sequence. A coordinator shall ignore this bit except for CT version determination during a node discovery response.

Bits 4-0, chunk number: CT-485 V 2.0 does not provide streaming or chunking as a service, so the chunk number shall also be '0' (zero) on all packets.

5.0 CT-CIM Mapped Message Reference

The following subsections define the Message Types mapped to CT-CIM services.

5.1 Get Configuration – Message Type 0x01

Table 2 – Get Configuration Request Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**
Field Length	1 Byte	1 Byte	1 Byte
Value	0x01	Refer to Section 4.3 above	0x00

Table 3 – Get Configuration Response Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Data**
Field Length	1 Byte	1 Byte	1 Byte	0 – 242 Bytes
Value	0x81	Refer to Section 4.3 above	0 - 242	Configuration MDI

The Coordinator sends a Get Configuration message to a Subordinate to tell it to send its configuration data. The format of the configuration data returned in the response is device-specific and is defined in the Configuration Message Data Interface (MDI) found in the profile specification for the corresponding device type.

5.2 Get Status – Message Type 0x02

Table 4 – Get Status Request Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**
Field Length	1 Byte	1 Byte	1 Byte
Value	0x02	Refer to Section 4.3 above	0x00

Table 5 – Get Status Response Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Data**
Field Length	1 Byte	1 Byte	1 Byte	0 – 242 Bytes
Value	0x82	Refer to Section 4.3 above	0 - 242	Status MDI

Status messages allow for detailed internal operation information to be sent out of the device and used by other devices on the bus. The Coordinator sends a Get Status message to a Subordinate to tell it to send its status data. The format of the status data returned in the response is device-specific and is defined in the Status Message Data Interface (MDI) found in the profile specification for the device type.

5.3 Set Control Command - Message Type 0x03

Table 6 – Control Command Request Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Command Code**	**Command Data**
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes	n-2 Bytes
Value	0x03	Refer to Section 4.3 above	n	See Profile Specifications	Varies depending on command

Table 7 – Control Command Response Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Command Code**	**Command Data**
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes	n-2
Value	0x83	Refer to Section 4.3 above	n	Echo of Request Message	Echo of Request Message OR Not Present

Control Commands provide for control of a subsystem over the network. Only one Control Command shall be transmitted in each Control Command packet.

Control Commands may require additional data to fulfill the command. Command data is defined with each Control Command. If more data is required than is allowed in a single ClimateTalk packet, multiple packets may be used. The Command Code shall be included with each packet.

The Command Data field in the Control Command response packet may be either an echo of the request data or left empty. In the latter case, the payload of the response consists solely of the Command Code. Devices shall be prepared to accept either response as legitimate.

The Command Code and Command Data shall be transmitted in Little Endian byte order.

The Control Commands for each profile are defined in that profile's specification. A list of all control command codes is found in the *Command Reference* document.

5.4 Set Display Message - Message Type 0x04

Table 8 – Set Display Request Format

Field Name	Message Type	Packet Number	Payload Length	Node Type	Message Length	Message
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	0-30 Bytes
Value	0x04	Refer to Section 4.3 above	2-32	See Command Reference	0-30	

Table 9 – Set Display Response Format

Field Name	Message Type	Packet Number	Payload Length	Confirmation Code
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes
Value	0x84	Refer to Section 4.3 above	2	0xAC 0x06

Display devices such as thermostats or PC applications may allow any subsystem to display generic data and generic subsystem information. The display device does not know what is being displayed, so it is up to the requesting subsystem to control the message and format.

Set Display Message is used to give display devices the message to display.

The maximum message length is thirty alpha-numeric ASCII characters.

5.5 Set Diagnostics – Message Type 0x05

Table 10 – Set Diagnostics Request Format

Field Name	Message Type	Packet Number	Payload Length	Node Type	Major Fault Code	Minor Fault Code	Message Length	Fault Message
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	0-15 Bytes
Value	0x05	Refer to Section 4.3 above	4-19	See Command Reference	Manufacturer Defined	Manufacturer Defined	0-15	

Table 11 – Set Diagnostics Response Format

Field Name	Message Type	Packet Number	Payload Length	Confirmation Code
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes
Value	0x85	Refer to Section 4.3 above	2	0xAC 0x06

Devices on the network are not queried periodically to determine if there is a failure. It is up to the subsystem to inform the thermostat of its inability to perform its required function.

The Diagnostic Packet provides two types of fault codes:

1. Major Fault - used for situations where the subsystem is inoperable
2. Minor Fault – used for situations where there is a problem or when maintenance is required, but the subsystem remains operable

A message may be included in the packet for display on a user-interface device. The maximum message length is fifteen alpha-numeric ASCII characters.

5.5.1 Clearing a Diagnostics Set Message

The subsystem is responsible for clearing the Diagnostic Message if the fault condition goes away. To clear a Diagnostic Message, the subsystem shall send a Set Diagnostic message as defined in Table 12 – Clearing a Set Diagnostics Message.

Table 12 – Clearing a Set Diagnostics Message

Field Name	Message Type	Packet Number	Payload Length	Node Type	Major Fault Code	Minor Fault Code	Message Length
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
Value	0x05	Refer to Section 4.3 above	4	See Command Reference	0x00	0x00	0x00

5.6 Get Diagnostics – Message Type 0x06

Table 13 – Get Diagnostics Request Format

Field Name	Message Type	Packet Number	Payload Length	Fault Type	Fault Index
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
Value	0x06	Refer to Section 4.3 above	2	0 = Minor 1 = Major	0 = All Faults, otherwise value is fault index

Table 14 – Get Diagnostics Response Format

Field Name	Message Type	Packet Number	Payload Length	Fault Data			
Field Length	1 Byte	1 Byte	1 Byte	n Bytes	n Bytes	…	n Bytes
Value	0x86	Refer to Section 4.3 above	n * number of faults	Fault 1 Data	Fault 2 Data		Fault n Data

Get Diagnostics requests information on faults in the subsystem. The specific fault is identified by the Major or Minor Fault Code and the Fault Index. A Fault Index value of 0x00 indicates that the subsystem should return information on all faults.

If multiple faults are sent in a single packet, then a null (0x00) character shall separate each fault data field.

5.6.1 Get Diagnostics Response

The Fault Data for each fault active in the system is defined in Table 15 – Fault Data Detail.

Table 15 – Fault Data Detail

Major Fault Code	Minor Fault Code	Fault Msg Length	Fault Message
1 Byte	1 Byte	1 Byte	n Bytes
Manufacturer Defined	Manufacturer Defined	n	0-15 Bytes

As defined in the Set Diagnostic message, the Fault Message is limited to a maximum of 15 characters. Thus, the fault data field for each fault is limited to a maximum of 18 bytes.

The following are examples of responses to requests made to an application with a single and a double fault.

<u>Example Payloads</u>

Single Fault Response:

Payload Length = 3 + FaultMsgLength(15)= 18
Major Fault Code = 0x12
Minor Fault Code = 0x00
Fault Msg Length = 15
Fault Message = "123456789012345"

Double Fault Response:

Payload Length = (3 + FaultMsgLength1(15) + 1 + 3 + FaultMsgLength2(5) = 27
Major Fault Code = 0x12
Minor Fault Code = 0x00
Fault Msg Length = 15
Fault Message = "123456789012345"
Null separator = 0
Major Fault Code = 0x34
Minor Fault Code = 0x00
Fault Msg Length = 5
Fault Message = "12345"

If an application receives a request for all faults or at a specific index when no fault is active, the application shall respond with the following data payload:

Payload Length = 4
Major Fault Code = 0
Minor Fault Code = 0
Fault Msg Length = 0

5.7 Get Sensor Data – Message Type 0x07

Table 16 – Get Sensor Data Request Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**
Field Length	1 Byte	1 Byte	1 Byte
Value	0x07	Refer to Section 4.3 above	0

Table 17 – Get Sensor Data Response Format

Field Name	Message Type	Packet Number	Payload Length	Sensor Data MDI
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x87	Refer to Section 4.3 above	n	Sensor MDI, per Profile Spec

The Get Sensor message requests sensor data from a sensor. The data payload is defined by the MDI for the sensor found in the profile specification for the sensor.

5.8 Set Identification – Message Type 0x0D

Table 18 – Set Identification Data Request Format

Field Name	Message Type	Packet Number	Payload Length	Data
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x0D	Refer to Section 4.3 above	n	Identification MDI, per Profile Spec

Table 19 – Set Identification Data Response Format

Field Name	Message Type	Packet Number	Payload Length	Confirmation Code
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes
Value	0x8D	Refer to Section 4.3 above	2	0xAC 0x06

The Identification MDI describes a device and its identity. The information contained in it is unique for each device type. The Identification MDI is defined in the Profile Specification for each device.

Set Identification Data allows the optional part of the Identification MDI to be written to over the network, permitting on-site configuration of devices.

MDIs have a maximum size of two hundred and forty bytes. The Identification MDI has restricted fields that are hardcoded into each device, followed by optional fields. The total size of the restricted fields varies by device because they contain the serial number and software version numbers used by the device and this will vary by manufacturer. The payload of the Set Identification message cannot be greater than 240 – X, where X is the total length in bytes of the restricted fields in the Identification MDI.

To ensure that the Set Identification request payload is not too large, a device intending to set identification information in another device should first send a Get Identification request to the other device. The requesting device shall use the information in the Get request to determine the maximum allowable payload size for the Set Identification request.

5.9 Get Identification – Message Type 0x0E

Table 20 – Get Identification Data Request Format

Field Name	Message Type	Packet Number	Payload Length
Field Length	1 Byte	1 Byte	1 Byte
Value	0x0E	Refer to Section 4.3 above	0

Table 21 – Get Identification Data Response Format

Field Name	Message Type	Packet Number	Payload Length	Data
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x8E	Refer to Section 4.3 above	N	Identification MDI, per Profile Spec

Get Identification Data retrieves the Identification MDI data from a device. Part of this data is reprogrammable and can be edited through the Set Identification Data message. The rest of the data is fixed for the control and cannot be edited via the Set Identification message.

The Identification MDI format is specific to each device type, and is defined in the Profile Specification for the corresponding device.

5.10 Set Application Shared Data to Network – Message Type 0x10

Table 22 – Application Set Network Shared Data to Network Request Format

Field Name	Message Type	Packet Number	Payload Length	Sector Node Type	Shared Data Length	Control ID	Manufacturer ID	App Node Type	Application Data
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	0-194 Bytes
Value	0x10	Refer to Section 4.3 above	n	See Command Reference	6-200	Assigned by the ClimateTalk Alliance	Assigned by the ClimateTalk Alliance		

Table 23 – Application Set Network Shared Data to Network Response Format

Field Name	Message Type	Packet Number	Payload Length	Data
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x90	Refer to Section 4.3 above	n	Echo of original Application Set Network Shared Data Request packet

Any subsystem using shared data and requiring network replication of shared data for future population in replacement controls shall use this message to send its own shared data to the network for storage. Shared data shall be sent out to the network when one of the following events occurs:

1. The application validates new Shared Data from any source, either due to a forced factory push as mentioned in section 5.20 or by voluntarily seeking and accepting data from any of the available sources.
2. A network Node List is received.
3. The Shared Data of a device changes due to internal changes in configuration or user interaction.

The first byte after the payload length is the called the Sector Node Type. The Sector Node Type is sent to the Coordinator to properly store and access the Application Shared Data, which follows this byte. The application's Node Type is used as this index when the data needs to be retrieved.

The bytes following the sector are known as the Application Shared Data. The Application Shared Data shall have the Shared Data Length, Control ID, Manufacturer ID, and Application Node Type.

The first byte in the Application Shared Data is the Shared Data Length. The maximum Application Shared Data size is two hundred bytes.

The next two bytes are the Control ID, which is provided by the CT Alliance. This ID is a two-byte integer, which is transmitted little endian. Following these bytes, the Manufacturer ID is transmitted, which is another two-byte CT Alliance-provided integer that is little endian. Control ID and Manufacturer ID shall be represented as follows: If company A provides a circuit board to company B for the use is a company B subsystem, the Control ID would be the ClimateTalk OEM ID for company A and the Manufacturer ID would be company B.

Following these bytes, the Application Node Type is sent to be used for validating the data at a later time.

Finally, the application-specific data is sent that includes the unique data to be stored on the network and later retrieved.

5.11 Get Application Shared Data from Network – Message Type 0x11

Table 24 – Get Application Shared Data from Network Request Format

Field Name	Message Type	Packet Number	Payload Length	Sector Node Type
Field Length	1 Byte	1 Byte	1 Byte	1 Byte
Value	0x11	Refer to Section 4.3 above	1	See Command Reference

Table 25 – Get Application Shared Data from Network Response Format

Field Name	Message Type	Packet Number	Payload Length	Sector Node Type	Shared Data Length	Control ID	Manufacturer ID	App Node Type	Application Data
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	0-194 Bytes
Value	0x91	Refer to Section 4.3 above	n	Echoed from request packet	6-200	Assigned by the ClimateTalk Alliance	Assigned by the ClimateTalk Alliance		

Get Application Shared Data is sent by the subsystem to the Coordinator to request that Shared Data previously stored on the network is transmitted to the subsystem. The Coordinator uses the Sector Node Type in the request as an index to locate the correct set of Shared Data to transmit to the subsystem.

During a Get Application Shared Data from Network request, if no shared data is available the response shall return with a Payload length of 0x00.

5.12 Set Manufacturer Device Data – Message Type 0x12

Table 26 – Set Manufacturer Device Data Request Format

Field Name	Message Type	Packet Number	Payload Length	Data
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x12	Refer 4.3	n	Manufacturer Device Data

Table 27 – Set Manufacturer Device Data Response Format

Field Name	Message Type	Packet Number	Payload Length	Data
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x92	Refer to Section 4.3 above	n	Echo of Request Data

This Message Type transmits the manufacturer-specific configuration data.

The Set Manufacturer Device Data is responded to with a data payload echoing the data payload from the original Set Request.

5.13 Get Manufacturer Device Data – Message Type 0x13

This message allows the retrieval of Manufacturing Device Data sent to the device by the Set Manufacturer Device Data message.

Table 28 – Get Manufacturer Device Data Request Format

Field Name	Message Type	Packet Number	Payload Length
Field Length	1 Byte	1 Byte	1 Byte
Value	0x13	Refer to Section 4.3 above	0

Table 29 – Get Manufacturer Device Data Response Format

Field Name	Message Type	Packet Number	Payload Length	Data
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x93	Refer to Section 4.3 above	n	Manufacturer Device Data

The payload of the response is in the same format used by the Set Manufacturer Device Data message.

5.14 Set Network Node List – Message Type 0x14

Table 30 – Set Network Node List Request

Field Name	Message Type	Packet Number	Payload Length	Data			
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	…	1 Byte
Value	0x14	Refer to Section 4.3 above	N	Coordinator Node Type	Index 1 Node Type		Index N Node Type

Table 31 – Set Network Node List Response

Field Name	Message Type	Packet Number	Payload Length	Data			
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	…	1 Byte
Value	0x94	Refer to Section 4.3 above	N	Coordinator Node Type	Index 1 Node Type		Index N Node Type

This Message Type is sent from the Coordinator to subsystems. It lists which subsystems are active on the network. When a subsystem sees itself on the list, it knows it is considered active on the network.

In CT-485 V 2.0, the Coordinator is allowed to send Node Lists of length 2 – 64. Subordinates must, however, be capable of handling Node Lists of length 2 – 240. To support future versions of the ClimateTalk standard, any node detected in a node list should be considered an active node on the network.

To ensure backwards compatibility, the Network Node List sent to a CT1.0 device shall have a maximum Payload Length of 16.

The first byte of the data payload identifies the Virtual Internal Subordinate Node Type residing within the node acting as the Coordinator. The second byte is the Node Type of the node at index one. The data payload follows this pattern to the last byte of the data payload. If any location contains a zero, this indicates that there is no active node at that index.

The response to the Network Node List is an echoed response to confirm a subsystem has a valid understanding of the network.

5.15 Direct Memory Access (DMA) Read – Message Type 0x1D

Information stored in Message Data Interfaces (MDIs) is accessed either through Message Types that retrieve the entire set of data in the MDI (e.g. Get Status), or through DMA messages that permit the retrieval of specific pieces of information from the MDI. Information in the MDI is referenced using a Database Identification (DB ID) number.

The following packet outlines how a DMA Read request would be made to retrieve a specific piece of data from a device. The DMA Read request may be one DB ID or a range of consecutive DB IDs. If non-consecutive data are required, multiple requests must be made, one for each DB ID or range of IDs.

Table 32 - DMA Request Format

Field Name	Message Type	Packet Number	Payload Length	MDI	Packet Number	Start DB ID	Range
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
Value	0x1D	0x00	n	See Table 33 – MDI Codes	0x00	As defined by the MDI	

Table 33 – MDI Codes

MDI	Value
Configuration	0x01
Status	0x02
Sensor	0x07
Identification	0x0E

Note that the MDI codes have the same value as the equivalent Get message type. For example, Get Status Data has a message type of 0x02. The Status MDI code also has a value of 0x02.

Table 34 - DMA Request Response

Field Name	Message Type	Packet Number	Payload Length	Data
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x9D	0x00	N	As defined by profile specification

Every message type that has data will be given a database identification number called the DB ID. For each packet of data per that message type, the DB ID will start over with the first byte being one. DMA Read Requests can only ask for a complete byte of data starting at that DB ID. Please refer to the associated message type's MDI for the proper DB ID number to be requested. The Start DB ID is where in the block of data the request is specifically requesting. And, the last byte tells the requesting device the range of data bytes requested starting from that DB ID number. A range of zero means to return one byte.

The response back to the requester is defined by each profile.

5.16 Set Manufacturer Generic Data – Message Type 0x1F

Set Manufacturer Generic Data is used to transmit proprietary data. This message may be used to encapsulate proprietary protocols. The data may be in any format, provided it conforms to the rules for payload length, etc.

The first field of the payload is the Manufacturer ID.

Table 35 – Set Manufacturer Generic Data Request Format

Field Name	Message Type	Packet Number	Payload Length	Payload	
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes	n Bytes
Value	0x1F	0x00	(2 + n)	Manufacturer ID	Data

5.16.1 Manufacturer Generic Set Response

The response to the Manufacturer Generic Set Message contains the Message Type 0x9F and the data payload echoes the data payload from the original Set Message.

5.17 Get Manufacturer Generic Data – Message Type 0x20

Get Manufacturer Generic Data is used to retrieve information stored in a proprietary format. Just as with the Set Manufacturer Generic Data, the first field of the payload shall be the Manufacturer ID, followed by the proprietary data.

Table 36 - Manufacturer Generic Request Packet

Field Name	Message Type	Packet Number	Payload Length	Payload	
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes	n Bytes
Value	0x20	0	2+n	Manufacturer ID	Data

5.17.1 Get Manufacturer Generic Data Response

A Get Manufacturer Generic Data Response is identified by a Message Type value of 0xA0.

The response back from this type of transaction is defined per each manufacturer's specific requirements. The first two bytes of the response message data payload are still required to contain the Manufacturer ID as in the requesting message, but the remaining part of the data payload may be defined as required, which is specific to the proprietary protocol in use.

Table 37 - Manufacturer Generic Pull Response Packet Definition

Field Name	Message Type	Packet Number	Payload Length	Payload	
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes	n Bytes
Value	0x20	Refer to Section 4.3 above	2 + n	Manufacturer ID	Data

5.18 Get User Menu - Message Type 0x41

Table 38 – Get User Menu Request Format

Field Name	Field Length	Value
Message Type	1 Byte	0x41
Packet Number	1 Byte	Refer 4.3
Payload Length	1 Byte	6 + n
Menu File	1 Byte	0x01
Main Menu	1 Byte	0x00-0x0A
Sublevel	1 Byte	0x00-0x0A
Reserved (Future Fetch Offset)	2 Bytes	0x0000
Maximum Number of Bytes to Return	1 Byte	

Table 39 – Get User Menu Response Format

Field Name	Field Length	Value
Message Type	1 Byte	0xC1
Packet Number	1 Byte	Refer 4.3
Payload Length	1 Byte	5 + n
Menu File	1 Byte	*Copied from request*
Main Menu	1 Byte	*Copied from request*
Sublevel	1 Byte	*Copied from request*
Reserved (Future Fetch Offset)	2 Bytes	0x0000
Maximum Number of Bytes to Return	1 Byte	*Copied from request*
Menu Data	n Bytes	

User menus are optional and, if the application does not allow for this ability, the response to a request would be a response of an Unknown Application Message Type as defined in the *Generic Application Specification*.

The ability for display devices like a thermostat or software application to interact with subsystems with the capability to display user configurable displays is a key concept in the development of an HVAC protocol. Over time, subsystems have become devices that require more configurations in the field due to the many variations subsystems may control.

The User Menu procedure is discussed at length in the Generic *Application Specification*. Refer to that document for details on how user menus are requested and retrieved via ClimateTalk. Below is a short description of the meaning of each of the fields in the request and response messages.

5.18.1 Menus

ClimateTalk menus have four levels:

1. Subsystem (corresponds to Menu File)
2. Main Menu
3. Sub Menu
4. Child/children. Children are the specific options available.

The Menu File, Main Menu, and Sublevel fields specify which menu or menu items to retrieve.

5.18.2 Menu File Field

It is possible for a subsystem to have the ability to display different types of user menu information. This byte is used to request a specific set of user menu information.

For example, a menu file of a one might correspond with a thermostat user menu while a menu file of a two might be a menu for a diagnostics device.

Currently only one subsystem level is supported, therefore, the Menu File value shall be 0x01.

5.18.3 Main Menu / Sublevel Fields

The next two bytes specify the main menu and sublevel menu data to return. A 0x00 in the main menu value is a wildcard and indicates that the subsystem should send all Level 1 main menu names.

5.18.4 Maximum Number of Bytes to Return

This byte specifies the maximum number of bytes the requesting subsystem is capable of receiving. This number shall include the size of the CTUM packet payload header including reserved bytes. As such, the maximum value identified inside Maximum Number of Bytes to Return shall be <= 240, 0xF0. If the response is larger than the value in this field, the response shall be broken up into multiple packets.

5.18.5 Fetch CT User Menu Response

The response back to the client display device is dependent on the type of request being made. The data that is being sent back is the information required for displaying via that client display.

The values for Menu File, Main Menu, Sublevel, and Fetch Offset shall be echoes of the values from the request. The Menu Data Field contains the actual User menu information to be displayed on the requesting display device.

5.19 Set User Menu Update – Message Type 0x42

A client display device has the ability to update a subsystem with user menu data depending if the data is able to be modified and has actually been altered. This section discusses the necessary requirements this modification of data is transmitted to the subsystem properly.

Table 40 – Update User Menu Request Format

Field Name	Field Length	Value
Message Type	1 Byte	0x42
Packet Number	1 Byte	Refer 4.3
Payload Length	1 Byte	7
Menu File	1 Byte	
Main Menu	1 Byte	

Field Name	Field Length	Value
Sublevel	1 Byte	
File Security Code	1 Byte	0x55
Updated Value	2 Bytes	
File Security Code	1 Byte	0xAA

Table 41 – Update User Menu Response Format

Field Name	Message Type	Packet Number	Payload Length	Payload	
Field Length	1 Byte	1 Byte	1 Byte	8 Bytes	
Value	0xC2	Refer to Section 4.3 above	8	Update Data	ACK/NAK

When a value has been modified via the client display device, the data payload is set up to transmit the modified value to that specific subsystem.

The first three bytes are defined in the previous section. The updated value is surrounded by two file security codes used to identify that a proper modification is being requested. The first file security code is a 0x55 while the ending security code is a 0xAA.

The updated value is based on the type of Main Menu/Sublevel Cluster Type, which corresponds to the appropriate value being sent back to the subsystem so it can evaluate that a proper modification has occurred.

The response back to the client display device is a sign that the data has been written successfully. The client display interface then can provide the proper feedback reflecting that the update was successful. The response echoes back the update with an appended ACK (0x06) at the end of those bytes to reflect a PASS condition. A NAK (0x15) informs the requester that the update failed.

5.20 Set Factory Shared Data to Application – Message Type 0x43

Table 42 – Set Factory Shared Data to Application Request Format

Field Name	Message Type	Packet Number	Payload Length	Shared Data Length	Control ID	Manufacturer ID	App Node Type	Application Data
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	0-194 Bytes
Value	0x43	Refer to Section 4.3 above	n	6-200	Assigned by the ClimateTalk Alliance	Assigned by the ClimateTalk Alliance		

Table 43 – Set Factory Shared Data to Application Response Format

Field Name	Message Type	Packet Number	Payload Length	Data
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0xC3	Refer to Section 4.3 above	N	Echo of original Set Application Shared Data Request packet

This message initializes a device's Shared Data at the time of manufacturing. This is not the same as the Shared Data transmission that occurs at the installed site. The data payload in this message contains the Application Shared Data.

The first byte in the data payload is the Shared Data Length. The maximum Application Shared Data size is two hundred bytes.

The next two bytes are the Control ID, which is provided by the CT Alliance. This ID is a two-byte integer which is transmitted little endian. Following these bytes, the Manufacturer ID is transmitted, which is another two-byte CT Alliance-provided integer that is little endian. Control ID and Manufacturer ID shall be represented as follows: If company A provides a circuit board to company B for the use is a company B subsystem, the Control ID would be the ClimateTalk OEM ID for company A and the Manufacturer ID would be company B.

Following these bytes, the Application Node Type is sent and used for validation of the data later.

Finally, the application-specific data is sent, which includes the unique data to be stored on the network and retrieved later.

5.21 Get Shared Data from Application – Message Type 0x44

Table 44 – Get Shared Data from Application Request Format

Field Name	Message Type	Packet Number	Payload Length
Field Length	1 Byte	1 Byte	1 Byte
Value	0x44	Refer 4.3	0x00

Table 45 – Get Shared Data from Application Response– Data Payload Example

Field Name	Message Type	Packet Number	Payload Length	App Node Type	Shared Data Length	Control ID	Manufacturer ID	App Node Type	Application Data
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	0-194 Bytes
Value	0xC4	Refer to Section 4.3 above	n	See Command Reference	6-200	Assigned by the ClimateTalk Alliance	Assigned by the ClimateTalk Alliance		

Use this command to have the application return the shared data set it is currently using.

If no shared data is present, an application shall respond with either a Payload Length of one (1) followed by the Node Type, or a Payload Length of zero. Either response shall be an indication the application does not have shared data. The preferred response when no shared data is present is a payload length of zero.

5.22 Set Echo Data – Message Type 0x5A

An Echo message is sent out to verify a subsystem can receive a message and respond with an echo of the data. Sending with the maximum data payload tests the communication lines are noise-resistant and can sustain a large data packet without errors.

Table 46 – Echo Request – Data Payload Example

Field Name	Message Type	Packet Number	Payload Length	Payload
Field Length	1 Byte	1 Byte	1 Byte	N Bytes
Value	0x5A	Refer to Section 4.3 above	N (1 … 240)	Known data to verify echo

Table 47 – Echo Response – Data Payload Example

Field Name	Message Type	Packet Number	Payload Length	Payload
Field Length	1 Byte	1 Byte	1 Byte	N Bytes
Value	0xDA	Refer to Section 4.3 above	N	Echo of Request payload

6.0 CT-485-specific Message Types

The following subsections define the Message Types used by CT-485 in maintaining and providing a connected world to the CT-CIM applications. These are messages that are not visible to the CT-CIM application and happen at a lower layer to keep communications alive.

6.1 Request to Receive (R2R) – Message Type 0x00

The R2R Message Type is used by CT-485 networks to control data flow.

When the Subordinate receives this message, any pending messages the Subordinate has may be transmitted to the Coordinator. If there are no pending messages queued within the Subordinate, a simple ACK message is sent back to the Coordinator and a period of time passes before the next Request to Receive packet is received by the same Subordinate again.

Table 48 – Request to Receive Message

Field Name	Message Type	Packet Number	Payload Length	R2R Code	Data	
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	8 Bytes	8 Bytes
Value	0x00	Refer 4.3	17	0x00	MAC Address of Coordinator	Session ID

If a Subordinate receives an R2R from the Network Coordinator and there is nothing that the Subordinate needs to relay to the network, the Subordinate responds with an R2R ACK.

Table 49 – Request to Receive ACK Response Message

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**R2R Code**	**Data**	
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	8 Bytes	8 Bytes
Value	0x00	Refer to Section 4.3 above	17	0x06	MAC Address of Subordinate	Session ID

6.2 Network State Request – Message Type 0x75

The Network State Request is for a new Network Coordinator to request the current network state from possibly already addressed nodes under the previous NC. The request has no payload and typically is broadcast.

Table 50 – Network State Request

Field Name	**Message Type**	**Packet Number**	**Payload Length**
Field Length	1 Byte	1 Byte	1 Byte
Value	0x75	Refer to Section 4.3 above	0

The Network State Response has the same payload as the last Network Node List observed by the responding device. The response typically is sent by one of the requesting receivers selected using Slot Delay Arbitration.

Table 51 – Network State Response

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Data**			
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	…	1 Byte
Value	0xF5	Refer to Section 4.3 above	N (0 … 240)	Coordinator's virtual internal Subordinate Node Type	Node Type at index 1		Node Type at index N

6.3 Address Confirmation – Message Type 0x76

The address confirmation holds the same payload as the current Node List and is used for reassuring current applications of the current Node List without starting a new session. The protocol may use this message periodically to indicate to all nodes that their addresses are as expected and to make sure they do not drop off of the network.

This document is subject to change without notice.

Table 52 – Address Confirmation Push Request

Field Name	Message Type	Packet Number	Payload Length	Data			
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	…	1 Byte
Value	0x76	Refer to Section 4.3 above	N (0 … 240)	Coordinator's virtual internal Subordinate Node Type	Node Type at index 1		Node Type at index N

The response to the address confirmation message is an echo from the request.

Table 53 – Address Confirmation Push Response

Field Name	Message Type	Packet Number	Payload Length	Data			
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	…	1 Byte
Value	0xF6	Refer to Section 4.3 above	N (0 … 240)	Coordinator's virtual internal Subordinate Node Type	Node Type at index 1		Node Type at index N

As in case of the Node List, a CT-485 V 2.0 Coordinator can send a Node List with a length of 2 - 64, while the Subordinate should be capable of handling 2 - 240 length Node Lists.

6.4 Token Offer – Message Type 0x77

The Token Offer is intended to offer a number of Subordinates a chance to arbitrate for a Transmission Opportunity.

Table 54 – Token Offer

Field Name	Message Type	Packet Number	Payload Length	Payload
Field Length	1 Byte	1 Byte	1 Byte	1 Byte
Value	0x77	Refer to Section 4.3 above	1	Node ID Filter

If the Node ID Filter is 0 (zero), all nodes to whom the above message is addressed and who are in need of a Transmission Opportunity start an arbitration process specific to the protocol. The winner sends out the following response:

Table 55 – Token Offer Response Format

Field Name	Field Length	Value
Message Type	1 Byte	0xF7
Packet Number	1 Byte	Refer 4.3
Payload Length	1 Byte	18
Address	1 Byte	Own address
Subnet	1 Byte	Own subnet
ClimateTalk MAC Address of addressee	8 Bytes	Own MAC ID
Session ID of addressee	8 Bytes	Own session ID

6.5 Version Announcement – Message Type 0x78

The Version Announcement is designed to be used for announcing a given subsystem's CT-485 capabilities and version.

Table 56 – Version Announcement

Field Name	Message Type	Packet Number	Payload Length	CT-485 version	CT-485 revision	CT-485 FFD (Coordinator capable)
Field Length	1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte
Value	0x78	Refer to Section 4.3 above	5	0x0001 – 0xFFFF	0x0001-0xFFFF	1 = FFD (Coordinator capable) 0 – RFD (not Coordinator capable)

The Version Announcement is generally a broadcast and the response also is a Version Announcement Broadcast subject to certain conditions specified in the associated Networking Specification.

6.6 Node Discovery – Message Type 0x79

The Node Discovery Request message is sent out periodically on broadcast address 0 and broadcast subnet 0 to discover newly added nodes to the network so that they can be addressed. The request packet will adhere to the following structure:

Table 57 – Node Discovery Request

Field Name	Message Type	Packet Number	Payload Length	Payload
Field Length	1 Byte	1 Byte	1 Byte	1 Byte
Value	0x79	Refer to Section 4.3 above	1	Node ID Filter

Table 58 – Node Discovery Response

Field Name	Message Type	Packet Number	Payload Length	Payload			
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	8 Bytes	8 Bytes
Value	0xF9	Refer to Section 4.3 above	18	Node Type	Reserved	MAC Address	Session ID

New devices that have not yet been accepted onto a network have an address of 0x00. If a new device receives this message with the appropriate Node ID Filter (either zero or its own Node Type), it shall respond with a Node Discovery Response. Note that Node Discovery Request is a non-addressable message. It is broadcast to address 0x00 and there is no acknowledgement. Upon receiving a valid Node Discovery Request, the receiving subsystem shall wait one Slot Delay and send back a Node Discovery Response to the Coordinator.

6.7 Set Address - Message Type 0x7A

Table 59 – Set Address Request Format

Field Name	Field Length	Value
Message Type	1 Byte	0x7A
Packet Number	1 Byte	Refer 4.3
Payload Length	1 Byte	19
Address	1 Byte	
Subnet	1 Byte	
ClimateTalk MAC Address of addressee	8 Bytes	
Session ID of addressee	8 Bytes	
Reserved	1 Byte	Must be set to 0x01

Table 60 – Set Address Response Format

Field Name	Field Length	Value
Message Type	1 Byte	0xFA
Packet Number	1 Byte	Refer 4.3
Payload Length	1 Byte	19
Address	1 Byte	Echoed from request
Subnet	1 Byte	Echoed from request
ClimateTalk MAC Address of addressee	8 Bytes	Echoed from request
Session ID of addressee	8 Bytes	Echoed from request
Reserved	1 Byte	Echoed from request

When a new node has been discovered through the Node Discovery message, the Coordinator sends the new node a Set Address message to inform it of its ClimateTalk network address.

If a new device receives this message with its MAC Address and Session ID, it shall respond with a Set Address Response. Note that Set Address Request is a non-addressable message. It is broadcast to address 0x00 and there is no acknowledgement. Upon receiving a valid Set Address Request, the receiving subsystem shall send back a Set Address Response to the Coordinator.

If the subsystem matches the information in the Set Address Request and accepts the address, the Set Address Response consists of an echo of the payload from the original message.

6.8 Get Node ID – Message Type 0x7B

Table 61 – Get Node ID Request Format

Field Name	Message Type	Packet Number	Payload Length
Field Length	1 Byte	1 Byte	1 Byte
Value	0x7B	Refer 4.3	0

Table 62 – Get Node ID Response

Field Name	Message Type	Packet Number	Payload Length	Payload		
Field Length	1 Byte	1 Byte	1 Byte	1 Byte	8 Bytes	8 Bytes
Value	0xFB	Refer to Section 4.3 above	17	Node Type	CT MAC Address	Session ID

Get Node ID is used by the Coordinator to retrieve the Node Type, MAC Address, and Session ID of a node discovered using the Node Discovery Request.

The Response to the Node ID Request contains the Node Type of the responding node, its MAC Address, and its Session ID.

6.9 Network Shared Data Sector Image Read / Write Request – Message Type 0x7D

This message is used by the Coordinator to spread Shared Data Sectors to other nodes for redundant storage and also to retrieve data from other storing nodes when the Coordinator receives a request for a specific sector data and does not have it in its own storage.

Table 63 – Network Shared Data Sector Image Read / Write Request Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Payload**		
Field Length	1 Byte	1 Byte	1 Byte	1 Bit Bit 7	7 Bits Bits 6..0	n-1 Bytes
Value	0x7D	Refer to Section 4.3 above	n	0=Write 1=Read	Node Type of Shared Data in payload	Shared Data (only for Write operation)

The index for the Shared Data is the Node Type associated with the Shared Data stored in the sector. In the case of a write operation, the first byte is the index and the remaining bytes contain the actual Shared Data Sector Image.

6.9.1 Network Shared Data Sector Image Read / Write Response

Regardless of whether the Network Shared Data Sector Image operation specified a read or a write, the responding node shall perform a fresh read of the Shared Data Sector corresponding to the index specified in the message after it processes this operation. This image is included in the response and shall adhere to the format identified in Table 64.

During a Shared Data Sector Image Read Request, if no shared data is available or if the device does not provide the service to store shared data, the response shall return with a Payload length of 0x00. During a Shared Data Sector Image Write Request, if the device does not provide the service to store shared data, the response shall be returned with an indication of an Unknown Application Payload per the *ClimateTalk Generic Application Specification*.

Table 64 – Network Shared Data Sector Image Read / Write Response

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Payload**	
Field	1 Byte	1 Byte	1 Byte	1 Byte	n-1 Bytes

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Payload**	
Length					
Value	0xFD	Refer to Section 4.3 above	n	Node Type	Shared Data Records

6.10 Network Encapsulation Request – Message Type 0x7E

The Encapsulation Request message may be used to embed other protocols or encrypted data or to bridge between other networked systems.

Table 65 – Encapsulated Request Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Payload**
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0x7E	Refer to Section 4.3 above	n	OpCode, {Operand 1, Operand 2, Operand 3}, {optional supplement data}, Encapsulated Request Packet Data

Table 66 – Encapsulated Response Format

Field Name	**Message Type**	**Packet Number**	**Payload Length**	**Payload**
Field Length	1 Byte	1 Byte	1 Byte	n Bytes
Value	0xFE	Refer to Section 4.3 above	n	OpCode, {Operand 1, Operand 2, Operand 3}, {optional supplement data}, Encapsulated Request Packet Data

7.0 Message Types

Not all message types are required by every profile. Profiles define which message types are required.

7.1 CT-485 Message Types

The following are the defined message types utilized by CT-485.

Table 67 – CT-485 Message Type Ranges

Reserved Message Type Ranges				
Request Range		**Response Range**		**Purpose**
Decimal	**Hex**	**Decimal**	**Hex**	
0	0x00	128	0x80	Bus Coordinator traffic maintenance
1 – 63	0x01 – 0x3F	129 – 191	0x81 – 0xBF	General commands
64	0x40	192	0xC0	Reserved - Future network routing and bridging purposes.
65 – 84	0x41 – 0x54	193 – 212	0xC1 – 0xD4	Advanced Commands
85	0x55	213	0xD5	Reserved - Wire-line and Wireless configuration and network Usage
86 – 89	0x56 – 0x59	214 – 217	0xD6 – 0xD9	Manufacturer Specific Commands
90	0x5A	218	0xDA	Command reserved for network Echo command
91 – 119	0x5B – 0x74	219 – 244	0xDB – 0xF4	Future reserved general purpose CT Commands
120 – 126	0x75 – 0x7E	245 – 254	0xF5 – 0xFE	Subnet management and node maintenance
127	0x7F	255	0xFF	Reserved - Network bus usage and subnet arbitration usage

Table 68 – CT-485 Message Types

Message Type	Message Name
0x00	Request to Receive (R2R)
0x01	Get Configuration
0x02	Get Status
0x03	Control Command
0x04	Set Display Message
0x05	Set Diagnostics
0x06	Get Diagnostics
0x07	Get Sensor Data
0x08 - 0x0C	Reserved
0x0D	Set Identification Data
0x0E	Get Identification Data
0x0F	Reserved
0x10	Application Set Network Shared Data
0x11	Application Get Shared Device Data
0x12	Set Manufacturer Device Data
0x13	Get Manufacturer Device Data
0x14	Set Network Node List
0x15 – 0x1C	Reserved
0x1D	Direct Memory Access Read
0x1E	Direct Memory Access Write
0x1F	Set Manufacturer Generic Data
0x20	Get Manufacturer Generic Data
0x21	Manufacturer Generic Reply
0x22-0x3F	Reserved
0x40	Reserved

Message Type	Message Name
0x41	Get User Menu
0x42	Update User Menu
0x43	Factory Set Application Shared Data
0x44	Get Shared Data from Application
0x45-0x54	Reserved
0x55	Reserved
0x56-0x59	Manufacturing Specific - Reserved
0x5A	Echo
0x5B-0x74	Reserved
0x75	Network State Request
0x76	Address Confirmation
0x77	Token Offer
0x78	Version Announcement
0x79	Node Discovery
0x7A	Set Address
0x7B	Get Node ID
0x7C	Reserved
0x7D	Network Shared Data Sector Image
0x7E	Network Encapsulation Request
0x7F	Reserved

8.0 Annex A – Bibliography

“TIA-485 (Revision A), Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems” *Telecommunications Industry Association*, 1998.

Zimmermann, Hubert (April 1980). "OSI Reference Model — The ISO Model of Architecture for Open Systems Interconnection". *IEEE Transactions on Communications*