Modular Communication Interface Specification for Demand Response

Revision History

Version 1.1	Original	Contribution to the H2G DEWG from the EPRI / USNAP Merging Project
Version 1.2		Incorporation of responses to round 1 H2G DEWG Comments
Version 1.3		Adjustments after 7/29 meeting of the comments resolutions team

Contents

1 Introduction

This document is a specification for a modular communication interface. The specification details the mechanical, electrical, and logical characteristics of a socket interface that allows communication devices (hereafter referred-to as UCMs – universal communication modules) to be separated from end devices (hereafter referred-to as SGDs – Smart Grid Devices). Although the potential applications of this technology are wide-ranging, it is intended at a minimum to provide a means by which residential products may be able to work with any load management system through user installable plug-in communication modules. Figure 1.1 illustrates the general concept.

Figure 1.1 – Illustrations of the Modular Communication Concept on an Appliance (left) or Energy Management Console (right)

This specification identifies the physical and data link characteristics of the interface, along with certain network and application layer elements as needed to assure interoperability over a broad range of device capabilities. In addition, it defines a mechanism through which application layer messages (defined in other standards) may be passed across the interface.

The scope of this specification is limited to the socket interface between the UCM and the SGD. It does not address the technology or protocol associated with the communication system of which the UCM is part.

1.1 Acronyms & Abbreviations

SGD	Smart Grid Device – the end device that is being informed of grid conditions
UCM	Universal Communication Module – the communications device that provides communication connectivity to an SGD
SPI	Serial Peripheral Interface – data transfer standard originally defined by

	Motorola (Freescale)
0x00 – 0xFF	Two digit (8 bit) hexadecimal numbers ranging from 0 to 255 decimal
b0, b1 .. b15	Bit values within a hexadecimal number. b0 is lsb.
MS	Abbreviation for Most Significant
LS	Abbreviation for Least Significant

2 Physical/Electrical Interface

Two physical form factors are presently defined. End device manufacturers may choose either, and communication module providers who wish to cover all products may offer two module versions. For both form factors, the communication protocol across the socket interface is the same, as described herein. Also in both cases, the power for the UCM is provided by the SGD. One form factor provides a low-voltage DC supply and an SPI serial data interface. This form factor is described in detail in **Appendix A** of this document. This option might be attractive in cases where the end device has no AC power source or when smaller socket size is required.

The second form factor provides AC service voltage (120/240V) and an RS-485 based serial interface. This form factor is described in detail in **Appendix B** of this document. This option might be attractive in cases where the end device does not provide a DC power supply, where compatibility with PLC communication modules is desired, or where communication module access to line frequency is needed.

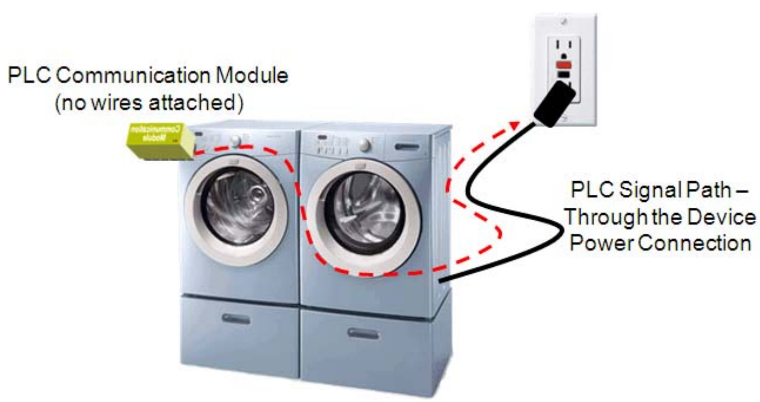


Figure 2.1 – PLC Communication Module Example

3 Serial Protocol

This specification defines an extensible serial protocol data unit that is manageable by the simplest of devices and also capable of being extended to accommodate the more complex. The general message format is as follows:

Message Type	Payload Length	Payload	Checksum
2 Bytes	**2 Bytes**	**Variable**	**2 Bytes**

Table 3-1–Protocol Data Unit Format

The idea in this design is that the “payload” part of the message can transport a range of protocols, with the “Message Type” field indicating what protocol and the checksum included so as to assure link-layer data integrity. This scheme provides a high level of flexibility and extensibility. A simple means is provided for SGDs and UCMs to discover which protocols one another support

3.1.1 Message Type Field

The “Message Type” bytes indicate the type of message, essentially indicating which communication protocol is represented in the payload. The following “Message Type” values are specified:

Message Type MS Byte	Message Type LS Byte	Description
0x00 to 0x05	0x00 to 0xFF	Reserved for vendor proprietary use
0x06	0x00 to 0xFF	Reserved to avoid confusion with link layer ACK
0x07	0x00 to 0xFF	For Future Assignment
0x08	0x01	Basic DR Application (at least partially supported by all devices)
0x08	0x02	Intermediate DR Application
0x08	0x03	Data Link Messages
0x08	0x04	Commissioning and Network Support Messages
0x08	0x05 to 0xFF	For Future Assignment
0x09	0x01	USNAP 1.0, Pass-Through
0x09	0x02	ClimateTalk, Pass-Through
0x09	0x03	Smart Energy Profile 1.0, Pass-Through
0x09	0x04	Smart Energy Profile 2.0 over IP, Pass-Through
0x09	0x05	OpenADR1.0 over IP, Pass-Through
0x09	0x06	OpenADR2.0 over IP, Pass-Through
0x09	0x07	Generic IP Pass-Through (IP packets self-identify version so both IPV4 and IPV6 are covered)
0x09	0x08 to 0xFF	For Future Assignment
0x0A to 0x14	0x00 to 0xFF	For Future Assignment
0x15	0x00 to 0xFF	Reserved to avoid confusion with link layer NAK
0x16 to 0xEF	0x00 to 0xFF	For Future Assignment
0xF0 to 0xFF	0x00 to 0xFF	Reserved for vendor proprietary use

Table 3-2 – Message Type Assignments

The "Vendor Proprietary" message types allow for device makers to make use of the serial interface for any purpose they wish. This includes manufacturing processes, field diagnostics, etc. Once a message begins with an address in these vendor-proprietary ranges, the remainder of the message may be formatted and used as desired by the manufacturer. In cases where the appliance has multiple internal subsystems sharing the serial bus, further addressing might be handled using the second byte.

New Message Types are to be assigned by the standards organization that manages this interface standard. Notionally, this would be done in coordination with the organization(s) responsible for the management of the domain-area or protocol that is to be passed through.

Chapter 5 , Data-Link Messages, describes the query used to determine if a device supports a specific message type. Support of the basic message type 0x08 0x01 is required and need not be queried.

3.1.2 Payload Length Field

The "Payload Length" is a two-byte representation of the number of bytes in the Payload field. For the "Basic DR" messages, this is always a 2. Other message types may have variable payload lengths and may also include additional length bytes somewhere in the message payload as defined by the other protocol's spec.

3.1.3 Checksum Field

The checksum is calculated starting with the first Message Type byte through to the end of the payload. Checksum calculation and encoding is according to a Fletcher checksum as detailed in Appendix D.

3.1.4 Bit and Byte Order

All bytes are transmitted most significant bit first. Multi-byte numbers, such as the Message Type and Length, are transmitted most significant byte first (Big Endian).

As an example, the 16 bit hexadecimal value 4143 (0x102F) would be transmitted as:

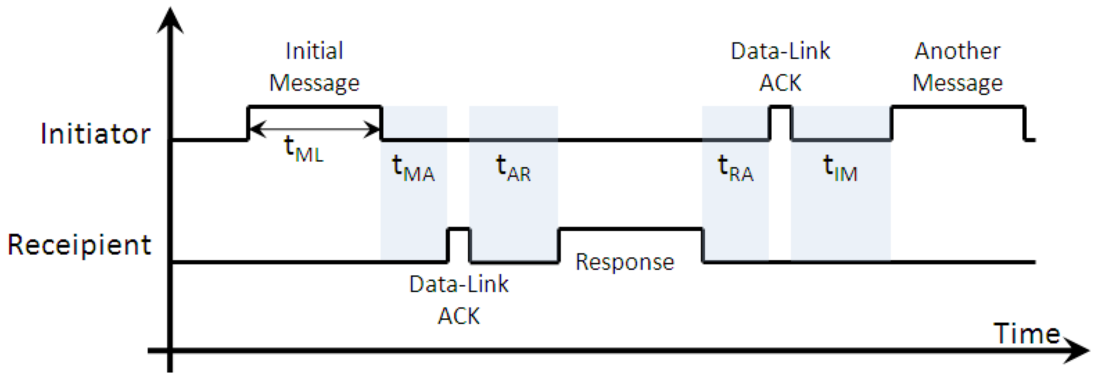
Byte	Contents	Comments
1	0x10	First byte transmitted
~~2~~	~~'0x2F~~	~~Last byte transmitted~~
2	0x2F	Last byte transmitted

3.1.5 Message Synchronization and Timing

All communication on this interface shall be half-duplex. This applies to both the low-voltage DC interface using SPI and the high voltage AC interface using RS-485. The nature of the two-wire RS-485 interface fundamentally allows only one side to transmit successfully at any time.

The specification allows either the communication module or the end device to initiate communication. As a result, bus contention is possible on the AC interface and must be electrically tolerated by the devices on both sides without damage. Recognition of contention is to be achieved by the absence of an appropriate response and recovery by the randomized data-link retry process described in section 3.1.6
.

Required message timing, including both link layer and application layer, are specified in Table 3-3~~Table 3-3~~.



Parameter	Minimum	Maximum	Description
t_{MA}	40[mS]	200[mS]	Time from the end of an initial message until the beginning of an associated data-link ACK.
t_{AR}	100[mS]	3[Seconds]	Time from the end of a data-link ACK until the beginning of an application response from the same device.
t_{RA}	40[mS]	200[mS]	Time from the end of a response message until the beginning of an associated data-link ACK.
t_{IM}	100[mS]	1[Second]	Time from the end of a final data-link ACK until the beginning of a new message. The specification of 1[Second] applies only to “grouped” Basic DR messages that are intended to be processed collectively. Otherwise there is no maximum.
~~t_{ML}~~		~~500[mS]~~	~~Maximum message duration, from end of first byte to end of last byte.~~
t_{ML}		4.5[Second]	Maximum message duration, from end of first byte to end of last byte. (Accommodates 8192 bytes at19.2KBps)

Table 3-3 Message Timing Requirements

3.1.6 Randomized Data-Link Retries

The data-link layer may determine failure from either lack of a response within the allowed time or from a data-link NAK with an error code indicating that the message was corrupted. Three retries are recommended at the data-link layer, with a randomized delay between each retry of 100 to 2000[mS].

Additional error recovery may exist at the application layer but the specific handling of such is outside the scope of this specification.

3.1.7 SGD Handling of Conflicting Messages

This specification supports multiple possible application layer protocols, including the basic and intermediate DR commands defined herein in addition to the pass-through of other industry standard protocols. Among these varied protocols are many different commands related to demand response. It is the responsibility of the UCM, and the system in which it participates, to provide SGDs with single, clear indications of the conditions at any time. In the event that an SGD is presented with conflicting commands, the last command received shall take precedence.

4 Simple Implementation

This specification identifies many message types and commands, but very few are mandatory. The messages are grouped into several types, including a set of data-link messages, basic and intermediate DR application messages, and advanced protocol pass-through messages. To be compliant with this specification, UCMs and SGDs are required to support only a few messages, each of which is limited to 8 bytes length. Devices may optionally support additional messages as desired.

The mandatory messages are:

Mandatory Message	Layer	Description
Shed	Basic DR Application	Fixed, 8 byte
End Shed	Basic DR Application	Fixed, 8 byte
Application ACK/NAK	Basic DR Application	Fixed, 8 byte
Communication Good/Bad	Basic DR Application	Fixed, 8 byte
Data Link ACK/NAK	Data Link	1 (ACK) and 2(NAK) bytes

Table 4-1 – Mandatory Message Summary

The establishment of mandatory messages is necessary in order to guarantee that any DR communication system, when connected to any end device, may still provide basic demand responsiveness. The mandatory list has been minimized in recognition that many present demand response systems provide only on/off control information and many end devices have only on/off response capabilities.

5 Data Link Messages

This specification adheres to a layered technique that distinguishes the data link layer from the application layer. The data link messages identified in this section are employed to manage data transfer across the interface and to assure its successful delivery. Support of the data-link ACK and NAK messages is mandatory, and is required in response to all messages except other data-link ACKs and NAKs (i.e. do not ACK an ACK). Use and/or support of the remaining data link commands are all optional, with a lack of support indicating that only the defaults are supported. The data-link ACK is a single byte and the data-link NAK is two bytes. The remaining data link commands follow the general protocol data unit format indicated in Table 3-1~~Table 3-1~~, with message type = (0x08, 0x03) and the payload field used as defined here in Table 5-1~~Table 5-1~~ .

			Mandatory	
Description	**Format**	**Usage**	**For UCM**	**For SGD**
Link ACK	0x06 (a single byte response)	Sent from either the UCM or the SGD to the other in response to a valid message received. Link ACK indicates: 1. Message type supported 2. Length valid and 3. CRC good	✓	✓
~~Link NAK~~	~~0x15, Link NAK Error Code~~ ~~(a two-byte response)~~	~~Sent from either the UCM or the SGD to the other in response to an invalid message received.~~ ~~See section 5.1.1 for "Link NAK Error Code" details.~~	~~✓~~	~~✓~~
Link NAK	0x15, Link NAK Error Code (a two-byte response)	Sent from either the UCM or the SGD to the other in response to an invalid message received. See section 5.1.1 for "Link NAK Error Code" details.	✓	✓

~~Message Type Support Query~~	~~Message Format Per the Serial Protocol (Table 3-1) with no Payload (6 bytes total)~~	~~To determine if a Message Type is supported, send:~~ ~~MT1, MT2, 0x00, 0x00, CS01, CS02 Where MT1 and MT2 identify the message type (see Table 3-2) and CS01 , CS02 are the checksum bytes.~~ ~~If the message type is supported, the recipient shall respond with a data-link ACK. If it is not supported, the recipient shall respond with a data-link NAK (error code 0x06)~~		
Message Type Support Query	Message Format Per the Serial Protocol (Table 3-1) with no Payload (6 bytes total) For this query, the first two bytes (MT1 and MT2) are indicative of the message type being investigated, not "0x08, 0x03".	To determine if a Message Type is supported, send: MT1, MT2, 0x00, 0x00, CS01, CS02 Where MT1 and MT2 identify the message type (see Table 3-2) and CS01 , CS02 are the checksum bytes. If the message type is supported, the recipient shall respond with a data-link ACK. If it is not supported, the recipient shall respond with a data-link NAK (error code 0x06)		
~~Request Different Power Mode~~	~~Message Format Per Basic DR Application (8 bytes total, see Table 6-1)~~ ~~Opcode1 =0x16~~ ~~Opcode2 = Power Level Indicator~~	~~Sent from the UCM to the SGD to request permission to draw higher power than the default.~~ ~~See section 5.1.2 for "Power Level Indicator" details.~~		
Request Different Power Mode	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 =0x16 Opcode2 = Power Level Indicator	Sent from the UCM to the SGD to request permission to draw higher power than the default. See section 5.1.2 for "Power Level Indicator" details.		

~~Request Different Bit Rate~~	~~Message Format Per Basic DR Application (8 bytes total, see Table 6-1)~~ ~~Opcode1 =0x17~~ ~~Opcode2 = Bit Rate Indicator~~	~~Sent from either the UCM or the SGD to the other to request a shift to a higher bit rate.~~ ~~See section 5.1.3 for "Bit Rate Indicator" details.~~		
Request Different Bit Rate	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 =0x17 Opcode2 = Bit Rate Indicator	Sent from either the UCM or the SGD to the other to request a shift to a higher bit rate. See section 5.1.3 for "Bit Rate Indicator" details.		
~~Query: Maximum payload length?~~	~~Message Format Per Basic DR Application (8 bytes total, see Table 6-1)~~ ~~Opcode1 = 0x16~~ ~~Opcode2 = 0x00~~	~~A query sent from either the UCM or the SGD to the other to ask how long message payloads can be. "Link NAK" means that only the default of 2 payload bytes are supported.~~		
Query: Maximum payload length?	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x16 Opcode2 = 0x00	A query sent from either the UCM or the SGD to the other to ask how long message payloads can be. "Link NAK" means that only the default of 2 payload bytes are supported.		

Response: Maximum data unit length	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x19 Opcode2 = Max Length Indicator	Response to an Opcode 0x16 query. Max Length Indicator: 0x00 = 2 (default) 0x01 = 4 0x02 = 8 0x03 = 16 0x04 = 32 0x05 = 64 0x06 = 128 0x07 = 256 0x08 = 512 0x09 = 1024 0x0A = 2048 0x0B = 4096 0x0C = 8192 0x0D to 0xFF reserved		
Response: Maximum payload length	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x19 Opcode2 = Max Payload Length Indicator	Response to an Opcode 0x16 query. Max Payload Length Indicator: 0x00 = 2 (default) 0x01 = 4 0x02 = 8 0x03 = 16 0x04 = 32 0x05 = 64 0x06 = 128 0x07 = 256 0x08 = 512 0x09 = 1024 0x0A = 2048 0x0B = 4096 0x0C = 8192 0x0D to 0xFF reserved		
Query: Get SGD Slot Number	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x1A Opcode2 = 0x00	Query sent from the UCM to the SGD to determine which slot the UCM is installed in.		

Query: Get SGD Slot Number	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x1A Opcode2 = 0x00	Query sent from the UCM to the SGD to determine which slot the UCM is installed in.		
~~Response: Slot Number~~	~~Message Format Per Basic DR Application (8 bytes total, see Table 6-1)~~ ~~Opcode1 = 0x1B~~ ~~Opcode2 = Slot Number~~	~~Response to Opcode 0x1A. Slot Number = a value from 0x00 to 0x0F indicating the slot number in which the UCM is installed.~~		
Response: Slot Number	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x1B Opcode2 = Slot Number	Response to Opcode 0x1A. Slot Number = a value from 0x00 to 0x07 indicating the slot number in which the UCM is installed.		
~~Query: Get Available Slot Numbers~~	~~Message Format Per Basic DR Application (8 bytes total, see Table 6-1)~~ ~~Opcode1 = 0x1C~~ ~~Opcode2 = 0x00~~	~~UCM asking the SGD what slot numbers exist and which are used.~~		
Query: Get Available Slot Numbers	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x1C Opcode2 = 0x00	UCM asking the SGD what slot numbers exist and which are used.		
~~Response: Available Slot Numbers~~	~~Message Format Per Basic DR Application (8 bytes total, see Table 6-1)~~ ~~Opcode1 = 0x1D~~ ~~Opcode2 = Slot Number Detail~~	~~Response to Opcode 0x1C.~~		
Response: Available Slot Numbers	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x1D Opcode2 = Slot Number Detail	Response to Opcode 0x1C. Slot Number Detail = Bit-field: 0 = Slot does not exist or is not occupied 1 = Slot is occupied Bit 0 (LSbit) = Slot Number 0 … Bit 7 (MSbit) = Slot Number 7		

~~Send Next Command to Slot~~	~~Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x1E Opcode2 = Slot Number~~	~~Instructs the SGD to forward the next message to the indicated Slot Number~~		
Send Next Command to Slot	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x1E Opcode2 = Slot Number	Instructs the SGD to forward the next message to the indicated Slot Number Slot Number = a value from 0x00 to 0x07 indicating the slot number to which the message is to be sent.		

Table 5-1 Data Link Command Set

5.1.1 Link NAK Error Codes

Link NAKs are two byte messages, the second of which is an error code. These codes are enumerated, defined as indicated in Table 5-2~~Table 5-2~~.

Link NAK Error Code	Priority	Description	Usage
0x00		No Reason	Not used.
0x01	1	Invalid Byte	Indicates that a byte framing or other invalid byte error has occurred (e.g. missing stop-bit on the AC RS-485 interface)
0x02	2	Invalid Length	Used to indicate that the length indicated in the PDU length field is out of range
0x03	3	Checksum Error	The bytes in the checksum field at the end of the message did not agree with the computed checksum
0x04	4	Reserved	
~~0x05~~	~~5~~	~~Message Timeout~~	~~Indicates that more than 500[mS] elapsed between receipt of the first byte and receipt of the last byte in a message transmission. 500mS was selected to allow any combination of data rate and payload in initial draft. As additional speeds and payloads are added some combinations may be invalid. This error code is not used by the DC Form Factor of Appendix A.~~
0x05	5	Message Timeout	Indicates that more than t_{ML} (defined in Table 3-3) elapsed between receipt of the first byte and receipt of the last byte in a message transmission. t_{ML} was selected to allow any combination of data rate and payload in initial draft. As

			additional speeds and payloads are added some combinations may be invalid. This error code is not used by the DC Form Factor of Appendix A.
0x06	6	Unsupported Message Type	Indicates that the "Message Type" is not supported
~~0x07~~	~~7~~	~~Request Not Supported~~	~~Indicates that the requested setting is not supported (e.g. a requested Power Mode or Bit Rate is not supported)~~
0x07	7	Request Not Supported	Indicates that the requested setting is not supported (e.g. a requested Power Mode or Bit Rate is not supported) This error code is used only in regards to link-layer requests, not in regards to lack of support for application layer requests.

Table 5-2 Link NAK Error Codes

In the event that multiple errors are detected, the Link NAK Error Code with the lowest priority number shall be returned.

5.1.2 Interface Power Limit Negotiation

UCMs may optionally use this data link function to request that the power consumption limits be changed to the level indicated by the request. The Opcode2 field, Power Level Indicator, is an enumeration:

Power Level Indicator	DC Form Factor		AC Form Factor		
	Maximum Continuous Average	Maximum Peak [mA]	Maximum Continuous Average	Maximum Peak	Maximum Instan-taneous
0x00 (default)	50[mA]	300[mA]	50 [mA,rms]	100 [mA,rms]	10 [Amps]
0x01	50[mA]	2[Amp]	N/A	N/A	N/A
0x02 to 0xFF (Reserved)					

Table 5-3 Interface Power Level Indicator Codes

Where:

- "Maximum Peak" = allowed for a 100[msec] maximum duration with a 10% maximum duty cycle in any given second.

- “Maximum Instantaneous” = allowed for 1[mS] maximum duration, with a 10% maximum duty cycle in any given 10[mS] period and a 1% maximum duty cycle in any given 10 second period.

For the DC interface, power is supplied at low voltage DC and for the AC interface, at the AC line voltage, as described in the respective physical layer appendices.

All UCMs must operate within the default power limits until negotiating a higher power with SGD. It is recognized that wired media UCMs may draw power from their media (PoE, PLC, Telephone) if additional power is required from a separate power supply or interface dongle.

This function always results in either a “Link ACK or a “Link NAK” response. “Link ACK” means that the requested power mode is approved. “Link NAK” with Error Code 0x07 means that the requested power mode is not supported.

5.1.3 Bit Rate Negotiation

For the DC form factor, this link layer function establishes the maximum rate at which the SGD may clock the SPI interface. For the AC form factor with the asynchronous RS-485 serial interface, it establishes the bit rate that both the UCM and the SGD must use in order to communicate with one another. Either UCM or SGD may optionally use this data link function to request a different bit rate than the default or current rate.

The “Bit Rate Indicator” field is defined as:

0x00 = 19.2[Kbps] (default)
0x01 = 38.4[Kbps]
0x02 = 57.6[Kbps]
0x03 = 115.2[Kbps]
0x04 = 256 [Kbps]
0x05 = 460.8[Kbps]
0x06 = 921.6[Kbps]
0x07 = 1843.2[Kbps]
0x08 = 3686.4[kbps]
0x09 to 0xFF reserved

If the requested bit rate is supported, “Link ACK” is immediately returned at the original bit rate. Future communication is at the new requested bit rate until such time as the units revert to default or a different bit rate is requested. “Link NAK” with Error Code 0x07 means that the requested Bit Rate is not supported.
Note: If no valid communication is exchanged for more than 15 minutes, both SGD and UCM shall return to the default of 19.2[Kbps] as identified in Section 5.1.5 .

5.1.4 Message Type support Query

After power-up, communication modules and end devices shall begin communication assuming only that the data-link ACK/NAK and mandatory functions of the "Basic DR" application are supported. This requires the ability to handle only 8 byte messages, parsing of only a short list of payloads (2 required commands), and allows NAK'ing of any unsupported commands.

After power-up or upon receipt of a message through its communication network, the UCM may send the Link Layer "Message Type Support Query" to discover what types of advanced message types the appliance or console is capable of supporting.

UCMs may be developed with the ability to receive more advanced protocols, like Smart Energy Profile or OpenADR over a network, and then either pass the messages through to the SGD if it supports them, or interpret the messages and translate into "Basic DR". This interface specification makes no assumptions or requirements of the communication systems themselves.

5.1.5 Power-Up and State Reset

Upon power cycle, all operational settings shall return to defaults. Any non-default settings (e.g. bit rate, power level) must be renegotiated following a power cycle.

The "Outside Communication Connection Status" message defined in Section 6 requires that a message be sent at least once every 1-5 minutes. If no valid communication is exchanged for more than 15 minutes, both SGD and UCM shall return to defaults. This includes returning to the default bit-rate and UCM's limiting power consumption to the default levels.

6 "Basic DR" Application (Message Type = 0x08, 0x01)

This section defines a set of "Basic DR" application commands and explains how they are supported by the interface. Because these commands are not a reference into another existing specification, they are fully documented here. Understanding the "Basic DR" commands is important, because even advanced communication modules and devices that may normally use more complex protocols, are required to be able to fall back to a few required Basic DR messages in the event that the device to which they are connected is not capable of the same advanced functionality.

Each "Basic DR" message shall be formatted as follows, with the message type being a "1" and the payload being a one byte Opcode1 and a one byte Opcode2.

Message Type = 0x08, 0x01	Payload Length = 0x00, 0x02	OpCode 1	OpCode 2	Checksum

2 Bytes	2 Bytes	1 Byte	1 Byte	2 Bytes

Table 6-1 Basic Application Data Format

The Basic DR message payloads are defined as follows:

				Mandatory?	
Description	**Opcode1**	**Opcode2**	**Usage**	**for SGDs**	**for UCMs**
~~Shed~~	~~0x01~~	~~Event Duration~~	~~Sent from the UCM to the SGD to when a load shed event begins.~~ ~~If other load management commands are attempted but not accepted by the SGD, then the UCM must fall back to this Opcode.~~ ~~Event Duration: Defined in Section 6.1.2~~ ~~Note: Event Durations of 10 minutes or less relate to "spinning reserve" uses. Event Durations greater than 10 minutes relate to "shift" uses.~~	~~✓~~	~~✓~~
Shed	0x01	Event Duration	Sent from the UCM to the SGD to when a load shed event begins. If other load management commands are attempted but not accepted by the SGD, then the UCM must fall back to this Opcode. Event Duration: Defined in Section 6.1.2 Note: Event Durations of 10 minutes or less relate to "spinning reserve" uses. Event Durations greater than 10 minutes relate to "shift" uses.	✓	✓
End Shed/Run Normal	0x02	Not Used	Sent once from the UCM to the SGD when a load shed or other curtailment event ends.	✓	✓
Basic Application ACK	0x03	ACK'ed Opcode1	Acknowledge successful receipt and support of previous command. Returned for all supported Opcodes except 0x03 (do not "Application ACK" an "Application ACK") and those that are queries and have a natural response, such as 0x12.	✓	✓

~~Basic Application NAK~~	~~0x04~~	~~Reason~~	~~Reject previous command. Sent from either SGD or UCM to the other when any of the following reasons occur.~~ ~~Reason:~~ ~~0x00 = No reason given~~ ~~0x01 = Opcode1 not supported~~ ~~0x02 = Opcode2 invalid~~ ~~0x03 = Busy~~ ~~0x04 to 0xFF Reserved~~	~~✓~~	~~✓~~
Basic Application NAK	0x04	Reason	Reject previous command. Sent from either SGD or UCM to the other when any of the following reasons occur. Reason: 0x00 = No reason given 0x01 = Opcode1 not supported 0x02 = Opcode2 invalid 0x03 = Busy 0x04 = Length Invalid 0x05 to 0xFF Reserved	✓	✓
~~Request for Power Level~~	~~0x06~~	~~Percent Setting~~	~~Sent from the UCM to the SGD to request that its average power level (relative to the full rating of the device) be reduced to a level between 0 and 100% of full value on a 7bit precision scale.~~ ~~Percent Setting:~~ ~~MSbit = 0, Least significant 7 bits:~~ ~~0x00 to 0x7F = 0 to 100% power absorbed~~ ~~MSbit = 1, Least significant 7 bits:~~ ~~0x00 to 0x7F = 0 to 100% power produced~~ ~~Details regarding the use of this command are provided in Section 6.2.1~~		

Request for Power Level	0x06	Percent Setting	Sent from the UCM to the SGD to request that its average power level (relative to the full rating of the device) be reduced to a level between 0 and 100% of full value on a 7bit precision scale. Percent Setting: MSbit = 0, Least significant 7 bits: 0x00 to 0x7F = 0 to 100% power absorbed MSbit = 1, Least significant 7 bits: 0x00 to 0x7F = 0 to 100% power produced Details regarding the use of this command are provided in Section 6.2.1		
Present Relative Price	0x07	Relative Price Indicator	Sent from the UCM to the SGD when a change in relative price occurs to inform of the new relative price. Relative Price Indicator: See Section 6.2.2 for description and usage If NAK'ed, UCM must use Opcodes 0x01 and 0x02 to inform SGDs.		
Next Period Relative Price	0x08	Relative Price Indicator	Sent from the UCM to the SGD when a change in relative price occurs to inform of the relative price in the next future period. Relative Price Indicator: See Section 6.2.2 for description and usage		
~~Time Remaining in Present Price Period~~	~~0x09~~	~~Event Duration~~	~~Sent from the UCM to the SGD when a change in price occurs to inform of the duration of the present price period.~~ ~~Event Duration: Defined in Section 6.1.2~~		
Time Remaining in Present Price Period	0x09	Event Duration	Sent from the UCM to the SGD when a change in price occurs to inform of the duration of the present price period. Event Duration: Defined in Section 6.1.2		
~~Critical Peak Event~~	~~0x0A~~	~~Event Duration~~	~~Critical Peak Event is in Effect (few times a year, 5 hours max) Sent once from the UCM to the SGD when a critical peak price event goes into effect. If NAK'ed, send Opcode 0x01.~~ ~~Event Duration: Defined in Section 6.1.2~~		

Critical Peak Event	0x0A	Event Duration	Critical Peak Event is in Effect (Critical Peak Events are intended to represent events that occur only a few times per year, on system peak days, for a maximum duration determined by the terms of the program) Sent once from the UCM to the SGD when a critical peak price event goes into effect. If NAK'ed, send Opcode 0x01. Event Duration: Defined in Section 6.1.2		
~~Grid Emergency~~	~~0x0B~~	~~Event Duration~~	~~A Grid Emergency is occurring. Sent once from the UCM to the SGD when a grid emergency event goes into effect. If NAK'ed, send Opcode 0x01.~~ ~~Event Duration: Same as for Opcode 0x01~~		
Grid Emergency	0x0B	Event Duration	A Grid Emergency is occurring. Sent once from the UCM to the SGD when a grid emergency event goes into effect. If NAK'ed, send Opcode 0x01. Event Duration: Defined in Section 6.1.2		
Grid Guidance	0x0C	Guidance Indicator	Sent from the UCM to the SGD to provide an arbitrary indication of whether energy consumption is preferred or not. Guidance Indicator: 0x00 = Bad Time to Use Energy 0x01 = Neutral 0x02 = Good/ Preferred Time to Use Energy 0x03 to 0xFF = Reserved		
Outside Comm Connection Status	0x0E	Connect Status Code	Sent from the UCM to the SGD when outside communication status is gained or lost. When in the "communicating" state, this command is resent every 1 to 5 minutes so that SGD's may know that the UCM is still attached and working. Connect Status Code: 0x00 = No / Lost Connection 0x01 = Found / Good Connection 0x02 = Poor / Unreliable Connection 0x03 to 0xFF = Reserved	✓ If known it must be shared	

Customer Override	0x11	Not Used	Sent from the SGD to the UCM when a customer chooses to override any load reduction process. Also sent immediately after acknowledging receipt of any load reduction message if the customer's preference is permanently set to override.		
Query: What is your operational state?	0x12	Not Used	Sent from the UCM to the SGD.		
~~State Query Response~~	~~0x13~~	~~Operating State Code~~	~~Sent from the SGD to the UCM in response to an Opcode 0x12 query~~ ~~Operating State Codes:~~ ~~(See description in section 6.2.4)~~		
State Query Response	0x13	Operating State Code	Sent from the SGD to the UCM in response to an Opcode 0x12 query Operating State Codes: (See description in section 6.2.4)		
Sleep	0x14	0x00	Sent from the SGD to the UCM to inform it that the SGD is idle, that information from the UCM is not needed, and that the UCM may shift into a low power state, if exists. This command assumes that the UCM will be provided with a "wake" command before it will be expected to operate. Usage assumes the UCM can hear "wake" messages while in "sleep" mode.		
~~Wake / Refresh Request~~	~~0x15~~	~~0x00~~	~~Sent from the SGD to the UCM to end a "Sleep" period and to request that all messages related to currently valid connection status, price, time, and/or load curtailment be sent.~~		
Wake / Refresh Request	0x15	0x00	Sent from the SGD to the UCM to end a "Sleep" period and to request that all messages related to currently valid connection status, price, time, and/or load curtailment be sent. UCM's that previously received a "Sleep" message are required to provide up-to-date grid information within 10 seconds of receipt of a "Wake" signal. How UCMs function internally during Sleep periods in order to be able to support this capability is up to the UCM provider.		

Simple Time Sync	0x16	Time Value	When supported, this command is sent from the UCM to the SGD on the hour. Time Value: Bits 7..5 = Weekday (0 = Sunday, 6 = Saturday) Bits 4..0 = Hour of Day (0 to 23)		

Table 6-2 Basic DR Application Command Set

~~After power-up, communication modules and end devices shall begin communication assuming only that the data-link ACK/NAK and mandatory functions of the "Basic DR" application are supported. This requires the ability to handle only 8 byte messages, parsing of only a short list of payloads (2 required commands), and allows NAK'ing of any unsupported commands.~~

~~After power-up or upon receipt of a message from the outside world, the UCM may send Opcode 0x15 queries to discover what types of advanced message types the appliance or console is capable of supporting. Alternatively, it could just figure it out by trying and getting ACK'ed/NAK'ed.~~

~~UCMs may be developed with the ability to receive more advanced protocols, like Smart Energy Profile or OpenADR over a network, and then either pass the messages through to the SGD if it supports them, or interpret the messages and translate into "Basic DR". This interface specification makes no assumptions or requirements of the communication systems themselves.~~

6.1.1 Basic Message Fixed Length

The "Basic DR Application" messages are fixed at 8 bytes total length, because the payload always consists of two bytes. This makes message parsing simple because processors can shift over a known number of bytes and always find the field of interest.

6.1.2 Event Duration Field

Basic DR Opcode1's 0x01, 0x09, 0x0A and 0x0B include a secondary Opcode2 that is an Event Duration indicator. This indicator is a single byte that defines the duration (or remaining duration) of the present event or price period. The value of 0x00 is reserved to indicate that the Event Duration is unknown and the value of 0xFF indicates that the duration is longer than what can be represented. For values from 0x01 to 0xFE, the indicated time is defined by a square function of the byte value:

$$\text{Time in Seconds} = 2 * (\text{Byte Value})^2$$

This results in the ability to represent a range of Event Durations between 0 and 129032 seconds (approximately 35.8 hours) as indicated in Figure 6-1~~Figure 6-1~~.

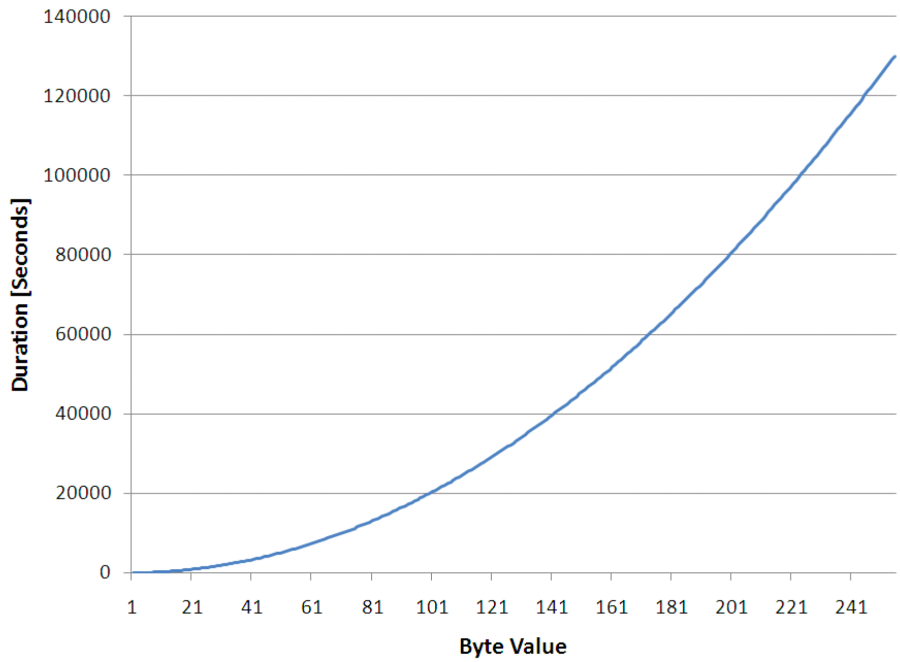


Figure 6-1 Non-Linear Event Duration Scaling

6.1.3 Grouped Messages

The Basic DR messages carry little information due to their length limit of two payload bytes. As a result, some groups of Basic DR messages may be thought-of as a grouped set, such as Opcodes 0x07, 0x08, and 0x09 (in any combination) sent from the UCM to the SGD to represent relative price information. In these cases, all the related messages in the group must be sent in rapid succession within the timeframe specified in Table 3-3~~Table 3-3~~.

6.2 Usage and Details of Basic DR Application Messages

6.2.1 Request for Power Level (Opcode 0x06)

Used by the service provider to ask suitable loads to provide ancillary service to the grid such as frequency support, in-hour load following, etc. Typical signal changes could occur as often as every 5 minutes in some scenarios or every few seconds in others. The mechanism used by the end device to respond (e.g. analog variability, adjustable regulator, or PWM duty cycling) is up to the manufacturer of the SGD and is not specified by this request. Example~~s~~: A water heater's bottom heating element, instead of operating at 4500 watts, could be managed at certain times of day to operate ~~900 watt (20%) nominal average power. But~~at 900 watt (20%). This setting could be modified rapidly in order to compensate for ~~wind power generation variation, in one 5 min~~variable generation sources ~~period the request could be for 1500 watt (33%) avg. power wind is blowing harder and in the next 5 minute period 300 watt (7%) when wind lets up. Battery chargers could change rates to accommodate this request.~~

~~Electric dryers, in some cases, could operate the heating elements at a reduced average power level. A smart pool or well pump could respond easily.~~such as wind power.

~~The~~During usage, the UCM relays the command to SGD; Application ACKs and NAKs from SGD may be conveyed upstream to Service Provider, if applicable.

For SGDs, loads that will operate at approximately the requested average power level shall ACK this command; under all other conditions a NAK should be provided. An appliance for all sorts of reasons might be able to comply at some times and not at others, e.g. exceeding the number of design relays cycles permitted per day. ACKs shall be reserved for those instances where ~~a reduction~~the target setting is actually ~~made.~~applied.

6.2.2 Relative Price Commands (Opcode 0x07 and 0x08)

The relative price commands are intended for use in variable-price systems wherein the UCM is able to provide to the SGD with an indication of the ratio of the current price to the average price. As indicated in ~~Table 4.2,~~Table 6-2, the Opcode2 field of these messages ~~(Opcodes 0x07 and~~(Opcode 1 = 0x07 or 0x08) provides the relative price indictor.

Relative_Price_Indicator = Present_Price / Average_Price

Where "Average_Price" is calculated as defined in Appendix E.

The value of 0x00 is reserved to indicate that the Relative Price is unknown and the value of 0xFF indicates that the Relative Price is higher than what can be represented. For values from 0x01 to 0xFE, the indicated Relative Price Indicator is defined by a nonlinear function of the byte value:

Relative_Price_Indicator = (Byte Value-1) * (Byte Value + 63) / 8192

This equation results in the scale indicated in Figure 6-1~~Figure 6-1~~Figure 6-2~~Figure 6-2~~.

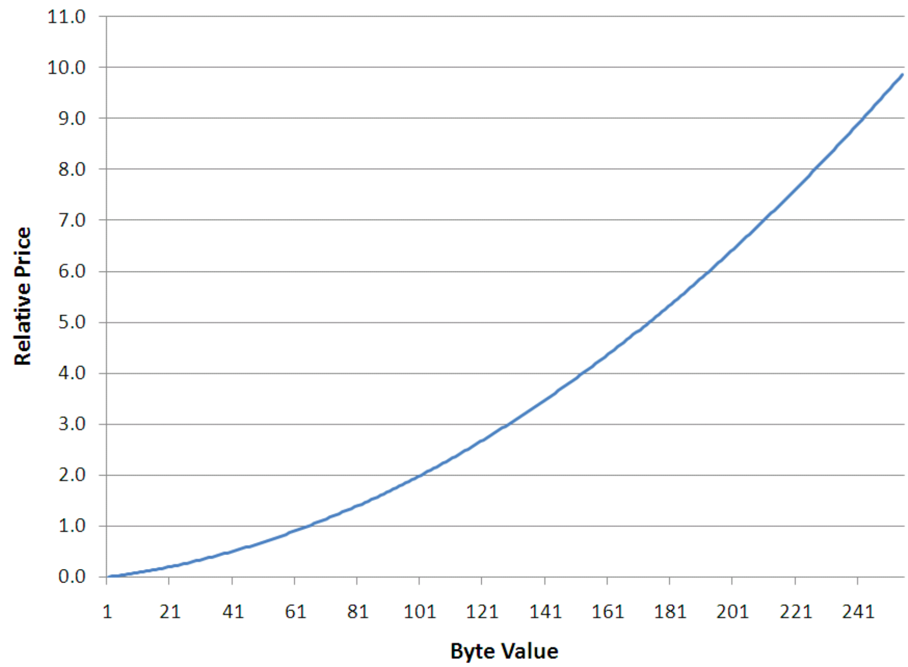


Figure 6-2 Non-Linear Relative Price Scaling

For SGD's the "Relative_Price_Indicator" may be simply used directly as an indication of how high or low the energy price for the period is relative to normal. As a simple ratio, it may be directly converted to percentages for customer presentation or preference settings.

The intent of the Opcode 0x07 is that it be sent from the UCM to the SGD at the beginning of each new price period. It reflects the price that has just become effective. The intent of the Opcode 0x08 is to provide a forward-looking indication of the relative price in the next future period. If available and supported, UCM's should attempt to provide SGDs with both the present (0x07) and next (0x08) indicators. SGD's may support neither, one, or both, at their discretion.

Note: Actual price information is supported through the use of more advanced commands as described in Section 9 –of this document.

6.2.3 Time Remaining in Present Price Period (Opcode 0x09)

This command is related to the Relative Price Messages and provides an indication of the time remaining until the next price change. When supported, this command must be sent once from the UCM to the SGD when a new relative price becomes effective or when a refresh is requested by the SGD.

This "Time Remaining in Present Period" message may be of most value when used in conjunction with the "Next Period Relative Price" message so that end devices know whether the price is increasing or decreasing at the end of this period.

6.2.4 *Operating State Monitoring (Opcodes 0x12 and 0x13)*

Opcode 0x12 requests the operational state of the SGD and Opcode 0x13 provides the response. The Opcode 0x13 response includes a single byte Opcode2 that describes the state of the device. The following may be extended in future versions. Up to 255 states may be defined.

Operating State Code	Meaning
0	Idle Normal
1	Running Normal
2	Running Curtailed Grid
3	Running Heightened Grid
4	Idle Grid
5	SGD Error Condition
6-255	Unused

~~Figure 4.5~~Table 6-3 – **Operating State Codes**

Op State Code 0 “Idle Normal” indicates a low power state, including user-operated devices that are not presently being used and automatically-operated devices that are in a standby state. Examples include a clothes dryer that is not operating, a refrigerator in idle mode (any time the compressor is not running), a water heater without a heating element energized, a TV in standby mode, etc.

Op State Code 1 “Running Normal” means that the SGD is operating in any normal mode or process. This includes user-operated devices that are presently being used (e.g. a washing machine that is in any cycle of the washing process including “soak”, an oven that is maintaining a set temperature even if the element is off at the moment, etc.) as well as automatically-operated equipment that is currently active (e.g. an HVAC unit that is presently running, a water heater with heating element energized, etc.)

Op State Code 2 “Running Curtailed Grid” the SGD is running, but has responded per some grid signal and has reduced average power relative normal or unrestricted operation. For example, a dryer may cycle the heating element in some way to reduce average power, or a refrigerator may stop the compressor or raise the temperature setting etc. An SGD, such as a stove top, may respond to an emergency curtailment for only one minute; if the state-query comes during this minute then the response would be state code 2. If the state query comes after the normal control method resumes then the state code response would be 1.

Op State Code 3 “Running Heightened Grid” means that the SGD has responded to a grid signal and has increased average power relative to normal or unrestricted operation. Common examples may be: 1) an HVAC or refrigerator consuming more average power

	(putting additional heat or cold into building, tank, or other thermal mass) in response to lower price or a more specific control command. 2) SGD operating in a grid friendly mode and using more average power. Grid friendly mode applies to SGDs that have flexibility to operate over a wide range of average power levels at times not constrained by customer demands. (e.g. a water heater maximizing water temperature at night, or an EV charging its battery at night.
Op State Code 4	“Idle Grid” means that the SGD has stopped (reduction to lowest consumption state) or is deferring consumption to a later time, in response to a grid signal. This state is different from OpState Code 1 because OpState Code 4 implies that the SGD would be operating were it not for the grid signal. Whereas OpState Code 2 is used to indicate some level of partial reduced consumption, OpState Code 4 indicates full reduction to Idle level.
Op State Code 5	SGD Inoperable: SGD is not operating because it needs maintenance support or is in some way disabled (i.e. not a response to the grid)

~~*6.2.5 Advanced Notice of Price/Curtailment Increase or Decrease (Opcodes 0x0C and 0x0D)*~~

~~These commands are used by a UCM to inform an SGD of how long the current condition is expected to continue, and whether the condition to follow is to be more or less attractive for energy consumption. UCMs that have knowledge of this information shall support this command and attempt to provide the information to SGDs. Likewise, SGDs that may take any action in regard to this information shall support this command and ACK accordingly.~~

~~When used, these advanced notice commands shall be sent once per minute, at or about the whole minute.~~

~~Only the next condition is to be be indicated. For example, if a UCM knows that a curtailment period is starting in 10 minutes, and ending in 70 minutes, it shall send only the Increase (Opcode 0x0C) command until such time as the curtailment period is in effect, after which the UCM shall begin sending the Decrease (Opcode 0x0D) command.~~

7 Intermediate DR Application (Message Type = 0x08, 0x02)

This section identifies intermediate commands to support more advanced functions. Unlike the Basic DR Application message set, which is fixed at 8 bytes total message length, Intermediate DR application messages have variable lengths. For those messages identified in this section that are longer than 8 bytes, affirmative response that the Intermediate DR message type is supported implies that the maximum Intermediate DR packet length can be accepted. In other words, this specification prohibits transfer of serial messages that would overflow buffers of the recipient.

Intermediate DR commands are all optional. Where the terms “optional” and “mandatory” are used in the tables in this section, they refer only to the requirement for those fields within the message being described. Intermediate DR commands follow the protocol data unit format indicated in Table 3-1~~Table 3-1~~, with the “Payload” field used as defined in Table 7-~~Table 7-~~ and Table 7-2.

Fields designated as Signed values use Two’s Compliment format. The signed 8-bit value -5 would be encoded as 0xFB, The signed 16 bit value -1 would be encoded as 0xFFFF.

Table 7.1 provides a list of the categories for the messages defined in this section. The Opcode1 column refers to the first byte (most significant byte) of the payload section as described in table 3.1 (protocol data unit format).

Opcode1	Usage Categories
0x00	Reserved
0x01	Device Information
0x02	Time & Date
0x03	Tier & Price
0x04	Demand Reduction
0x05	Demand Response Event Schedules
0x06	Home Consumption/Production
0xF0-0xFF	Manufacturer Specific

Table 7-1 Intermediate DR Application Command Set (Command Byte Description)

Table 7.2 provides a more granular list of the messages defined in this section.

	Payload			
Description	**OpCode1**	**OpCode2**	**Additional Payload Definitions**	**Usage**
Info Request	0x01	0x01	Defined in section 7.1.1	Request device information
Get/Set UTC Time	0x02	0x00	Defined in section 7.1.2	Set or request Time
Get/Set Energy Price	0x03	0x00	Defined in section 7.1.3	Set or request the current price of energy
Get/Set Tier	0x03	0x01	Defined in section 7.1.4	Set or request the current tier value

~~Get/Set Temperature Offset~~	~~0x03~~	~~0x02~~	~~Defined in section 7.1.5~~	~~Set or request the current temperature offset value~~
Get/Set Temperature Offset	0x03	0x02	Defined in section 7.1.5	Set or request the current temperature offset value
~~Get/Set SetPoint~~	~~0x03~~	~~0x03~~	~~Defined in section 7.1.6~~	~~Set or request the current temperature set point value(s)~~
Get/Set SetPoint	0x03	0x03	Defined in section 7.1.6	Set or request the current temperature set point value(s)
Start Autonomous Cycling	0x04	0x00	Defined in section 7.1.7	Start a Demand Reduction cycling event per the parameters passed in the command
Terminate Autonomous Cycling	0x04	0x01	Defined in section 7.1.8	Terminate a Demand Reduction cycling event
Get/Set Temperature Offset	0x05	0x00	Defined in section **Error! Reference source not found.**	Set or request the current temperature offset value
Demand Response Event Schedules	0x06	0x00	Defined in section 7.2.1	Send Scheduled Events Request

Table 7-2 Intermediate DR Application Command Set

Intermediate DR message responses include a response code byte. Table 7.3 provides a list of the defined response codes.

Response Code	Description
0x00	Success
0x01	Command not implemented
0x02	Bad Value – one or more values in the message are invalid

0x03	Command Length Error – command is too long
0x04	Response Length Error – response is too long

Table 7-3 Response Code Values

7.1 Usage and Details of Intermediate DR Application Messages

7.1.1 Info Request

This command may be optionally used by the UCM to determine information about the SGD and by the SGD to determine information about the UCM.

7.1.1.1 Format GetInformation() – Request

Byte	Hex value	Comments	Mandatory/ Optional
1	0x01	OpCode1	M
2	0x01	OpCode2	M

7.1.1.2 Format GetInformation()- Reply

Byte	Hex value	Comments	Mandatory/ Optional
1	0x01	OpCode1	M
2	0x81	OpCode2 (Reply always has bit 7 high)	M
3	0x00	Response Code	M
4	0x02	Interface Specification Major Version	M
5	0x00	Interface Specification Minor Version	M
6-7		Vendor ID	M
8-9		Device Type	M
10-11		Device Revision	M
12-15		Capability Bitmap	M
16		Reserved	M
17-32		Model Number – UTF-8	O
33-48		Serial Number – UTF-8	O
49		Firmware Year – 20YY	O
50		Firmware Month	O
51		Fimware Day	O
52		Firmware Major	O
53		Firmware Minor	O

Device Information included here is read-only and will not change after the device has been powered on (i.e. this value may be read once at power-on with confidence that there is no need to read it again until a subsequent reset or power cycle). Typical values included in the device information command include the interface specification version that the unit was designed for, firmware information, Serial Number and Model Number.

Vendor ID

Vendors who support this command must request a unique vendor ID provided by the standard development organization or users alliance.

Device Type

SGD Device Types		UCM Device Types (Phy/MAC)	
Device Type	**Description**	**Device Type**	**Description**
0x0000	Unspecified Type	0x4000	Wireless (other, non-standard)
0x0001	Water Heater - Gas	0x4001	PLC (other, non-standard)
0x0002	Water Heater - Electric	0x4002	Wired (other, non-standard)
0x0003	Water Heater – Heat Pump	0x4003	IEEE 802.15.4
0x0004	Central AC – Heat Pump	0x4004	IEEE 802.11 (e.g. Wi-Fi)
0x0005	Central AC – Fossil Fuel Heat	0x4005	IEEE 802.16 (e.g. WiMAX)
0x0006	Central AC – Resistance Heat	0x4006	VHF/UHF Pager
0x0007	Central AC (only)	0x4007	FM (RDS / RBDS)
0x0008	Evaporative Cooler	0x4008	Wired Ethernet
0x0009	Baseboard Electric Heat	0x4009	Coaxial Networking
0x000A	Window AC	0x400A	Telephone Line
0x000B	Portable Electric Heater	0x400B	IEEE 1901 (BPL)
0x000C	Clothes Washer	0x400C	IEEE 1901.2 (Narrowband-PLC)
0x000D	Clothes Dryer - Gas	0x400D	ITU-T G.hn
0x000E	Clothes Dryer - Electric	0x400E	ITU-T G.hnem (Narrowband-PLC)
0x000F	Refrigerator/Freezer	0x400F	Cellular (Mobile, any)
0x0010	Freezer	0x4010	Utility AMI, Wireless
0x0011	Dishwasher	0x4011	Utility AMI, PLC
0x0012	Microwave Oven	0x5000	Gateway Device
0x0013	Oven – Electric		
0x0014	Oven – Gas	All others	Available for Assignment
0x0015	Cook Top – Electric		
0x0016	Cook Top - Gas		
0x0017	Stove – Electric		
0x0018	Stove - Gas		
0x0019	Dehumidifier		

SGD Device Types		UCM Device Types (Phy/MAC)	
Device Type	**Description**	**Device Type**	**Description**
0x0020	Fan		
0x0030	Pool Pump – Single Speed		
0x0031	Pool Pump – Variable Speed		
0x0032	Electric Hot Tub		
0x0040	Irrigation Pump		
0x1000	Electric Vehicle		
0x1001	Hybrid Vehicle		
0x2000	In Premises Display		
0x8000 – 0xFFFF	Manufacturer Defined Device Types		

Capability Bitmap

Bit (2^n)	Description
0	Cycling supported
1	Tier mode supported
2	Price mode supported
3	Temperature Offset supported
4-15	Reserved

Model Number
Device model number, all zeros = not supported

Serial Number
Device serial number, all zeros = not supported

Firmware Year
Year – 2000 (e.g. Firmware Year = 11 (0x0B) for 2011).

Firmware Month
0 (0x00) = January, 11 (0x0B) = December.

Firmware Day
1 - 31

7.1.2 Get/Set UTC Time

Set the time on the device.

7.1.2.1 Format GetUTCTime() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x02	OpCode1	M
2	0x00	OpCode2	M

7.1.2.2 Format GetUTCTime()- Example reply

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x02	OpCode1	M
2	0x80	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M
4-7		UTC Seconds	M
8		Time zone offset in ¼ hours (e.g. EST = -20)	M
9		DST Offset in ¼ hours	M

7.1.2.3 Format SetUTCTime() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x02	OpCode1	M
~~2~~	~~0x01~~	~~OpCode2~~	~~M~~
2	0x00	OpCode2	M
3-6		UTC Seconds	M
7		Time zone offset in ¼ hours (e.g. EST = -20)	M
8		DST Offset in ¼ hours	M

7.1.2.4 Format SetUTCTime()- Example reply

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x02	OpCode1	M
~~2~~	~~0x81~~	~~OpCode2 (Reply always has bit 7 high)~~	~~M~~
2	0x80	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M

UTC Seconds

Unsigned 32 bit value of seconds since 1/1/2000 00:00:00 UTC

Time Zone Offset

Signed 8 bit value, offset from UTC in 15 minute intervals (e.g. EST would be -20 (0xEC))

DST Offset

Unsigned, if non-zero, add value in 15 minute intervals to UTC seconds for local time conversion

7.1.3 Get/Set Energy Price

7.1.3.1 Format GetEnergyPrice() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x00	OpCode2	M

7.1.3.2 Format GetEnergyPrice()- Example reply from UCM

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x80	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M
4-5		Current Price	M
6-7		Currency Code	M
8		Digits After Decimal Point	M
9-12		Expiration Time/Date in UTC seconds	O
13-16		Next Price	O

7.1.3.3 Format SetEnergyPrice() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x00	OpCode2	M
3-4		Current Price	M
5-6		Currency Code	M
7		Digits After Decimal Point	M
8-11		Expiration Time/Date in UTC seconds	O
12-15		Next Price	O

7.1.3.4 Format SetEnergyPrice()- Example reply from SGD

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x80	OpCode2 (Reply always has bit 7 high)	M

3		Response Code	M

Current Price

Unsigned 32 bit value.

Currency Code

Unsigned 32 bit value, see ISO 4271, US Dollar = 840, Euro = 978, Mexican Peso = 484, Canadian Dollar = 124. The codes can be found on the Web site of the ISO 4217 Maintenance agency, SNV - SIX Interbank Clearing (http://www.currency-iso.org/iso_index/iso_tables/iso_tables_a1.htm).

Digits After Decimal Point

Unsigned, the number of digits after the decimal point (e.g 22¢ = 0.22 dollars so the Digits after decimal point value would be 2).

Expiration Time/Date

Unsigned 32 bit value of seconds since 1/1/2000 00:00:00 UTC

Next Price

The price that takes affect when the current time reaches the Expiration Time/Date

7.1.4 Get/Set Tier

7.1.4.1 Format GetTier() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x01	OpCode2	M

7.1.4.2 Format GetTier()- Example reply from SGD

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x81	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M
4		Current Tier (0 – 6, 255 = no tier)	M
5-8		Expiration Time/Date in UTC seconds	O
9		Next Tier	O

7.1.4.3 Format SetTier() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x01	OpCode2	M
3-4		Current Tier (0 – 6, 255 = no tier)	M

5-8		Expiration Time/Date in UTC seconds	O
9-10		Next Tier	O

7.1.4.4 Format SetTier()- Example reply from UCM

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x81	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M

Current Tier
0 to 6, 255 = no active tier

Expiration Time/Date
Unsigned 32 bit value of seconds since 1/1/2000 00:00:00 UTC

Next Tier
Tier that takes effect when the expiration time/date is reached

7.1.5 Get/Set Temperature Offset

7.1.5.1 Format GetTemperatureOffset() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x02	OpCode2	M

7.1.5.2 Format GetTemperatureOffset()- Example reply from SGD

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x82	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M
4		Current Offset	M
5		Units	M

7.1.5.3 Format SetTemperatureOffset() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x02	OpCode2	M
3		Current Offset	M
4		Units	O

7.1.5.4 Format SetTemperatureOffset()- Example reply from UCM

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x81	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M

Current Offset

Unsigned 8 bit value. Offset to apply to the normal operating temperature in degrees

Units

0 = degrees F, 1 = degrees C

7.1.6 Get/Set Set Point

7.1.6.1 Format GetSetPoint() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x03	OpCode2	M

7.1.6.2 Format GetSetPoint()- Example reply

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x83	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M
4-5		Device Type	M
6		Units	M
7-8		Set Point 1	M
9-10		Set Point 2	O

7.1.6.3 Format SetSetPoint() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x03	OpCode2	M
3-4		Device Type	M
5		Units	M
6-7		Set Point 1	M
8-9		Set Point 2	O

7.1.6.4 Format SetSetPoint()- Example reply

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x03	OpCode1	M
2	0x83	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M

Device Type

See Info Request (Section 7.1.1) for Device Type table. For Set command, packet is ignored if the device type doesn't match the SGD's device type.

Units

0 = degrees F, 1 = degrees C

Set Point 1

Signed 16 bit value. First temperature value, 0x8000 (-32768) = don't change (set)/not supported (get). For Water Heaters, Top Element set point. For Thermostats, Heat set point. For Refrigerator/Freezer, Refrigerator set point.

Set Point 2

Signed 16 bit value. Second temperature value, 0x8000 = don't change (set)/not supported (get). For Water Heaters, Bottom Element set point. For Thermostats, Cool set point. For Refrigerator/Freezer, Freezer set point.

7.1.7 Automomous Cycling

7.1.7.1 Format StartCycling() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x04	OpCode1	M
2	0x00	OpCode2	M
3-6		Event ID	M
7-10		Start Time UTC seconds since 1/1/2000	M
11-12		Duration in minutes	M
13		Duty Cycle	M
14		Start Randomization in minutes	O
15		End Randomization in minutes	O
16		Criticality	O

7.1.7.2 Format StartCycling()- Example reply from SGD

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x04	OpCode1	M
2	0x80	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M

Event ID

Unsigned 32 bit value control event identifier

Start Time

Unsigned 32 bit value of seconds since 1/1/2000 00:00:00 UTC, 0 = Now

Duration

Duration of the control event in minutes

Duty Cycle

% reduction of the load (e.g. 75 means that the device will be off ¾ of the time)

Start Randomization

The start of the control will be delayed by this randomized value in minutes. The start randomization does not change the duration of the event.

End Randomization

The event duration will be lengthened by this random value.

Criticality

Reserved for future use.

7.1.8 Demand Reduction – Terminate Cycling

7.1.8.1 Format TerminateCycling() - Request

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x04	OpCode1	M
2	0x01	OpCode2	M
3-6		Event ID	M
4		End Randomization in minutes	O

7.1.8.2 Format TerminateCycling()- Example reply from SGD

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x04	OpCode1	M
2	0x81	OpCode2 (Reply always has bit 7 high)	M
3		Response Code	M

Event ID
Unsigned 32 bit value control event identifier

End Randomization
Continue the control for random value time to prevent large groups from turning on at the same time.

7.2 Demand Response Event Schedules

Allows for displaying information about demand response events duration, %shed, etc

7.2.1 Send Scheduled Events Request

7.2.1.1 Format SendScheduledEvents Request (from SGD)

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x50	OpCode1	M
2	0x01	OpCode2	M
4		MSB Start Time (UTC)	M
5		…	M
6		…	M
7		LSB Start Time (UTC)	M
8		UINT8 Number of Events	M

7.2.1.2 Format SendScheduledEvents Reply (from UCM)

Payload Byte	Hex value	Comments	Mandatory/ Optional
1	0x50	OpCode1	M
2	0x81	OpCode2 (Reply bit always has bit 7 high)	M
4		Response Code	M

8 Commissioning and Network Messages (Message Type = 0x08, 0x04)

These commands are all optional. The format for these messages is as identified in Table 3-1~~Table 3-1~~, with the payload as identified in Table 8-1~~Table 8-1~~. For those messages identified in this section that are longer than 8 bytes, it is required that the "Maximum Data Unit Length" query defined in the data-link section be used first. In other words, this specification prohibits transfer of serial messages that could overflow buffers of the recipient. Unless indicated otherwise, when referenced in this section, "ACK" and "NAK" refer to the Basic DR Application ACK and NAK.

	Payload			
Description	OpCode1	Opcode2	Additional Payload	Usage

Set Network ID	0x01	0x00 = LAN 0x01 = WAN	Network ID String	Sent to the UCM, from either the SGD of any other configuration/commissioning tool to which it is connected Network ID String: 1 to 120 Bytes, ASCII encoded NAK means that the UCM does not accept Network IDs.
Set User ID	0x02	0x00 = LAN 0x01 = WAN	User ID String	Sent to the UCM, from either the SGD of any other configuration/commissioning tool to which it is connected. User ID String: 1 to 120 Bytes, ASCII encoded NAK means that the UCM does not accept User IDs.
Set Password	0x03	0x00 = LAN 0x01 = WAN	Password String	Sent to the UCM, from either the SGD of any other configuration/commissioning tool to which it is connected. Password String: 1 to 120 Bytes, ASCII encode NAK means that the UCM does not accept Passwords.
Join Network	0x04	0x00	None	Set to the UCM to instruct it to join the network for which it is configured
Leave Network	0x05	0x00	None	Set to the UCM to instruct it to leave the network to which it is currently connected

Table 8-1 Commissioning and Network Messages

9 Pass-Through of Standard Protocols

In its simplest mode of operation, this modular communication interface provides for physical layer diversity and allows application layer (and network layer) protocols that are used in the communication system to pass through directly to the end device. In such a mode of operation, the UCM need not understand the content of the messages or parse them in any way. In order for this to work, the end device must be capable of accepting and understanding the protocol that is passed through.

This specification provides support for any number of such pass-through protocols through these mechanisms:

Full Encapsulation in the Message Payload - As illustrated in Figure 9-1~~Figure 9-1~~, other protocols are inserted in the message payload without any modification. As described below, the organizations that own and manage each protocol will define how their messaging is placed into the payload field.

Message Type	Payload Length	Payload = Pass-Through Message	Checksum
2 Bytes	2 Bytes	Variable	2 Bytes

Figure 9-1 Pass-Through Message

Message Type Field – As described in Table 3-2~~Table 3-2~~, each pass-through protocol is assigned a "Message Type" code. This code is placed in the "Message Type" field indicated in Figure 9-1~~Figure 9-1~~ whenever pass-through of that protocol is occurring. This field allows end devices that might support multiple protocols to recognize which is being used and to parse accordingly.

Message Type Support Query – As indicated in Table 5-1~~Table 5-1~~, a link layer query must be used to determine if the other device (UCM or SGD) supports the pass-through of a particular protocol before a pass-through is attempted. In this way, the support of the protocol to be passed through is known.

Maximum Message Length Negotiation - Devices must assure that the maximum message length associated with the protocol to be passed-through is supportable by the device on the other side of the interface. This is to be achieved by using the Link-Layer maximum message length query described in Table 5-1~~Table 5-1~~.

Pass-Through Handling Instructions by the SDO – For each protocol to be passed through in this way, the standards organization that owns and manages the protocol shall produce an application note describing how that protocol is to be mapped into the payload field. Such an application note will define, for example, the byte or field to begin with and the byte or field to end with when passing through the protocol.

9.1 Example Pass-Through Handling Instructions

9.1.1 USNAP 1.0 Protocol Pass-Through

This section shows how the USNAP1.0 protocol is supported by the interface. The messages will be formatted as follows, with the message type being a 0x09, 0x01, and the payload being defined in the USNAP1.0 specification.

Message Type = 0x09, 0x01	Payload Length	USNAP1.0 Message	Checksum

2 Bytes	2 Bytes	Variable	2 Bytes

Figure 9-2 USNAP1.0 over Serial

The USNAP1.0 Message in the payload shall begin with the first byte of the USNAP message, called "CommandByte1", and shall end with the last byte before the "Checksum" as identified in the USNAP1.0 specification.

9.1.2 SEP1.0 Pass-Through

This section shows how the SEP1.0 protocol is supported by the interface. The messages would look as follows, with the message type being a 0x09, 0x05, and the payload being defined entirely by the Zigbee SEP organization.

Message Type = 0x09, 0x05	Payload Length	SEP1.0 Message	Checksum
2 Bytes	2 Bytes	Variable	2 Bytes

Figure 9-3 SEP1.0 over Serial

<Zigbee.org to define this section detailing how the SEP1.0 messages are to be handled>

9.1.3 ClimateTalk Pass-Through

This section shows how the ClimateTalk protocol is supported by the interface. The messages would look as follows, with the message type being a 0x09, 0x02, and the payload being defined entirely by the ClimateTalk organization.

Message Type = 0x09,0x02	Payload Length	ClimateTalk Message	Checksum
2 Bytes	2 Bytes	Variable	2 Bytes

Figure 9-4 ClimateTalk Over Serial

<ClimateTalk.org to define this section detailing how the SEP1.0 messages are to be handled>

9.1.4 General Internet Protocol Pass-Through

This section shows how the interface is used to support pass-through of applications over IP. This might be of interest for many purposes, including communication systems and SGDs that are capable of web access.

This IP pass-through mechanism supports both IPV4 and IPV6, with the self-describing Version field of the IP packet distinguishing between the two.

Using the previously defined data-link commands, either the SGD or the UCM may probe the other to determine what Message types it supports. Whenever a UCM and SGD that both support pass-through of IP are connected together, the two can recognize this fact and pass-through communication can commence. When passing an IP packet over the serial interface, the UCM or SGD shall add a leading Message Type Field (0x09,0x04) and two-byte Length field; and a trailing checksum as illustrated in Figure 9-5~~Figure 9-5~~ (IPV6 example shown, IPv4 handled in similar fashion).

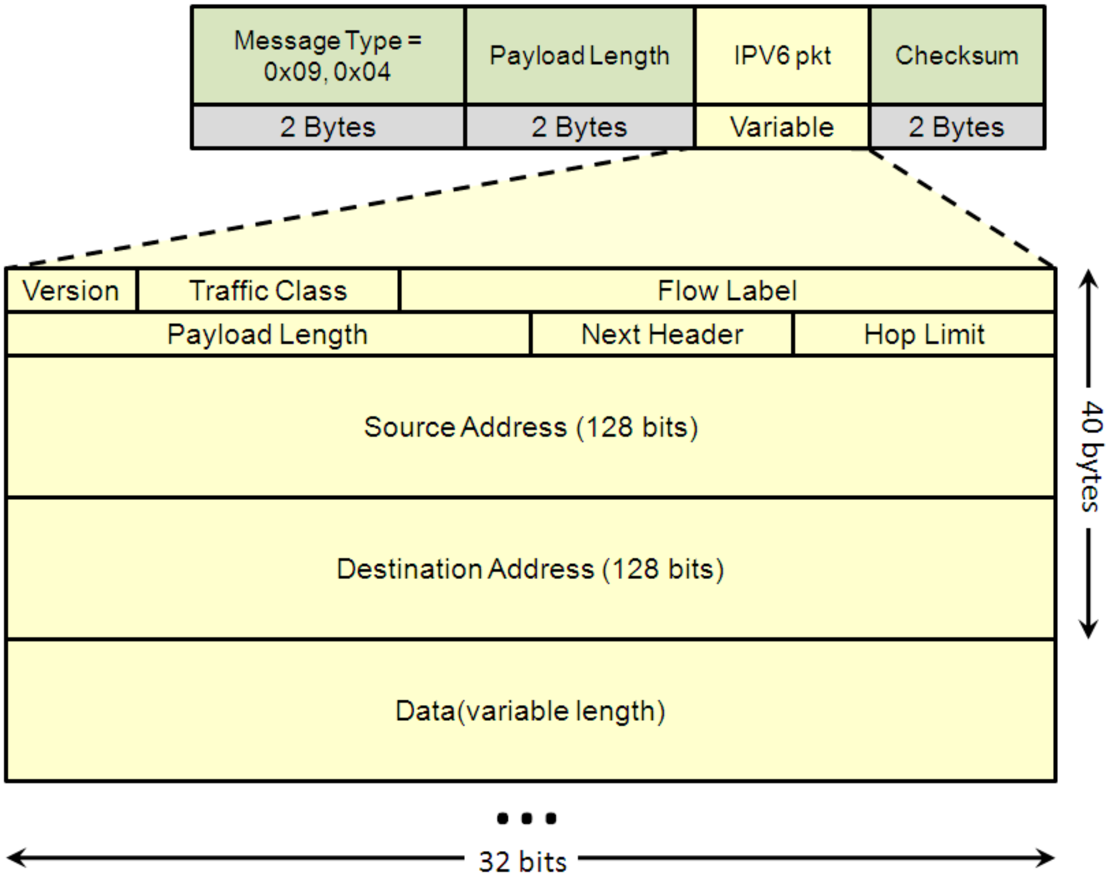


Figure 9-5 Internet Protocol Pass-Through (IPV6 Example)

Use of any IP pass-through requires that the UCM and SGD support payload lengths of 2048 bytes or greater. This length must be negotiated after power up or reset using the link layer “Maximum Payload Length” negotiation described in Section 5 . IP pass-through packets may NOT be fragmented by the UCM.

Response: Maximum payload length	Message Format Per Basic DR Application (8 bytes total, see Table 6-1) Opcode1 = 0x19 Opcode2 = Max Payload Length Indicator	Response to an Opcode 0x16 query. Max Payload Length Indicator: 0x00 = 2 (default) 0x01 = 4 0x02 = 8 0x03 = 16 0x04 = 32 0x05 = 64 0x06 = 128 0x07 = 256 0x08 = 512 0x09 = 1024 0x0A = 2048 0x0B = 4096 0x0C = 8192 0x0D to 0xFF reserved		

10 Example Communication Exchanges

Simple Serial, Request Operating State

Comm Module to End Device	→ 08 01 00 02 12 00 D8 5F	Opcode 0x12, what is your state?
End Device to Comm Module	← 06	Link Layer Acknowledge of prev. msg.
End Device to Comm Module	← 08 01 00 02 13 02 D1 63	Opcode 0x13, End device is curtailed
Comm Module to End Device	→ 06	Link Layer Acknowledge of prev. msg.

Simple Serial, Unsupported Message Followed by Shed Message

Comm Module to End Device	→ 08 01 00 02 07 40 79 89	Opcode 0x07, Relative Price
End Device to Comm Module	← 06	Link Layer Acknowledge of prev. msg.
End Device to Comm Module	← 08 01 00 02 04 01 01 44	Opcode 0x04, App NAK, Bad Opcode
Comm Module to End Device	→ 06	Link Layer Acknowledge of prev. msg.
Comm Module to End Device	→ 08 01 00 02 01 00 0C 3D	Opcode 0x01, Shed
End Device to Comm Module	← 06	Link Layer Acknowledge of prev. msg.
End Device to Comm Module	← 08 01 00 02 03 01 04 42	Opcode 0x03, App ACK of Opcode 0x01
Comm Module to End Device	→ 06	Link Layer Acknowledge of prev. msg.

Query, Then Use, of Smart Energy Profile 2.0 over IP

Comm Module to End Device	→ 09 04 00 00 CS CS	Link Query, Do you support SEP2/IP
End Device to Comm Module	← 06	Link Layer Acknowledge of prev. msg.
Comm Module to End Device	→ 09 04 01 3D <SEP msg> CS CS	SEP2/IP msg, length only an example
End Device to Comm Module	← 06	Link Layer Acknowledge of prev. msg.

* Note that the Basic DR application ACK (Opcode 0x03) is only used in response to Basic DR commands (message type 0x08, 0x01). Once SEP2/IP is used (message type 0x09, 0x04), then the application layer acknowledge, if any, is up to the SEP2 specification.

** Note that the UCM does not have to initiate communications. For example, an SGD can initiate an IP-based exchange of information.

11 General Security Principles

The serial interface between a UCM and an SGD supports end-to-end security at the application layer and/or at the IP / network layer. It is not encrypted at the link layer. For certain application protocols, such as the Basic DR, it is not encrypted. In this case, the socket interface is treated as a protected local interface like any other wired connection between circuit boards inside a product. If the communication on the communication network (PLC, wireless, etc) is encrypted, as it may be in a secured Wi-Fi or Zigbee HAN for example, the decryption may occur in the communication module or be passed through to the SGD, if supported.

In the case of more advanced protocols, like internet pass-though, encryption may exist within the IP packet embedded in the serial message. For example, if the communication network is Wi-Fi, then a Wi-Fi communication module may receive an IP packet wirelessly, strip-off any 802.11 phy/mac part, insert the IP packet as the "Payload" in the message structure shown in Table 3-1~~Table 3-1~~, and send it on through to the SGD. In this case, the communication module would be serving as a phy/mac translator and would know nothing of the packet's content, which may or may not be encrypted. The comm module would only know whether or not the SGD is accepting or NAK'ing the messages.

12 Load Management Event Randomization

It is noted that if large numbers of end devices turn on or off simultaneously there may be an undesirable sudden change in load on the power system. This unnatural synchronization could result in voltage problems on distribution systems. This type of situation could occur in a broadcast-type communication system (pager, FM, PLC) where a real-time request to shed load is sent to a large number of devices at once. Another scenario that would cause similar alignment is that of a scheduled event where a large number of devices are all responding to a common schedule, such as a high price period that begins or ends at a specific time.

Unless explicitly identified in the application layer command (e.g. the Intermediate cycling command), this interface specification does not require SGDs to perform event randomization. In fact, such behavior could prevent the device from being used for certain time-sensitive services, such as compensation for intermittent renewable generation. Rather, the utility, communication system, or UCMs, shall perform randomization, if desired. The advantage of this approach is that it allows a single SGD to be sold nationwide without assuming to know what timings are needed by the local utility. Figure 12-1~~Figure 12-1~~ illustrates this concept.

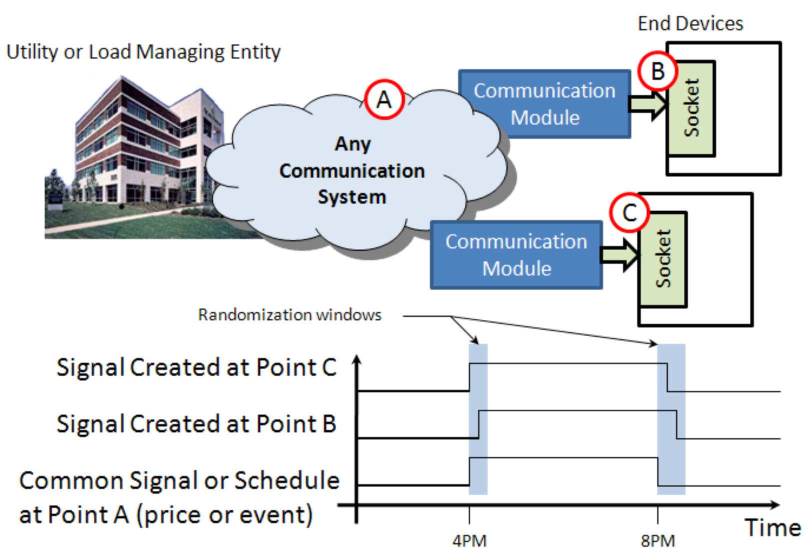


Figure 12-1 Example of Randomization of Events by Communication Modules

13 Appendix A – Low Voltage DC Form Factor

13.1 Overview

13.1.1 Limitations

- The transport speed is limited by SGD and UCM processor bandwidth. The data link defaults to a slow data rate (19.2k) and small payload(2 bytes). Data link layer commands allow negotiating the speeds and payloads for more capable devices. Process for reverting to default settings is also documented at the application layer.
- This specification does not include device reliability requirements. There are also no specifications for handling, dropping, ESD resistance, etc. It is expected that manufacturers will make appropriate design decisions, and that the market will reward those who make good decisions.

13.2 Physical Layer

13.2.1 Power for UCM

Power for UCM is provided by SGD. Power is supplied at 3.3 +/- 0.3V DC. The power consumption limits are described in section 5.1.2 .

13.2.2 Mechanical Interface

13.2.2.1 DC Form Factor Board Layout

The DC UCM device must conform to one of the physical layouts identified in Figure 13-2 and Figure 13-3. The Standard DC UCM layout is designed to fit inside the SGD while the Extended Size DC UCM may protrude outside the SGD. It includes an extended segment where larger components and external connectors may be included. The extended size DC UCM allows extension of 51mm dimension. All dimensions within the envelope of the standard size radio must be met.

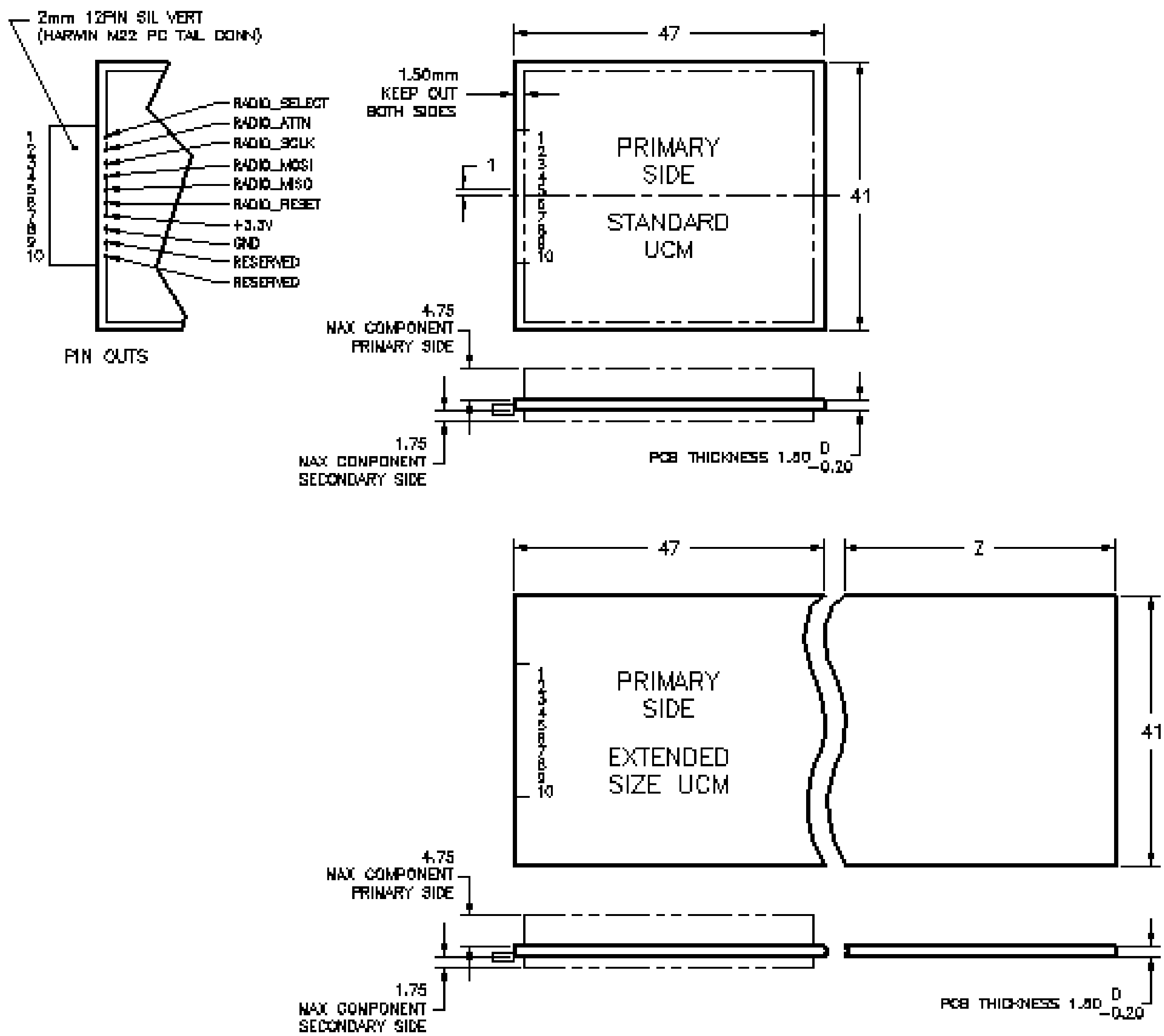


Figure 13-1 DC Form Factor PCB Dimensions

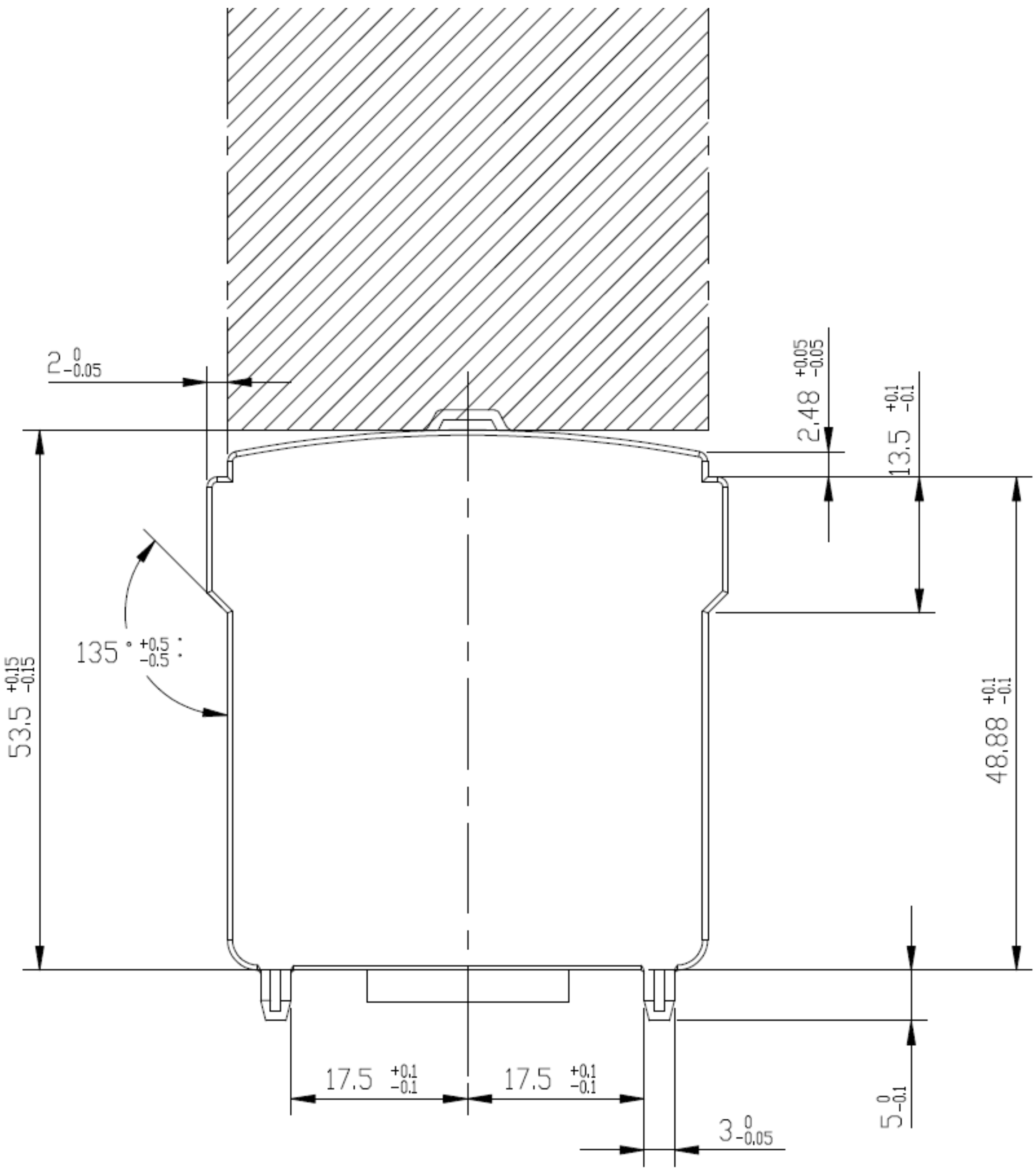
13.2.2.2 Module Configuration

The DC form factor UCM module is defined for the standard layout. Extended size UCMs will have extended housings which are required to have all the features and meet all the dimensions of the standard module with the exception of the 51mm dimension.

A special case of extended length is defined for antennas and wired media. The side of the module opposite of the connector is reserved for extending an antenna or accessing the wired media.

For example, PLC media that need access to the power line or wired media could provide a cord terminated with a Nema 1-15 plug. Another example could be a wired ethernet module with an RJ45 receptacle. Specific implementations are left to the discretion of the communication provider.

746


2 0 -0.05
2.48 +0.05 -0.05
13.5 +0.1 -0.1
135° +0.5 -0.5
53.5 +0.15 -0.15
48.88 +0.1 -0.1
17.5 +0.1 -0.1
17.5 +0.1 -0.1
5 0 -0.1
3 0 -0.05

747

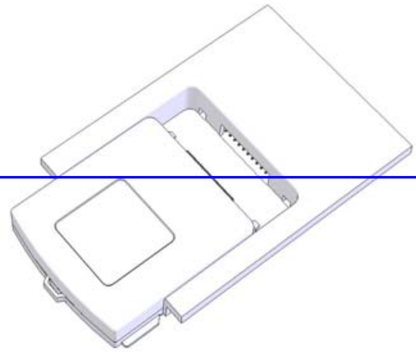


Figure 13-2 DC Form Factor Housing Dimensions – Top View

USANP_Radio_Spec.pdf

* Hatched Area Dimensions for connectors, antennas, or large components

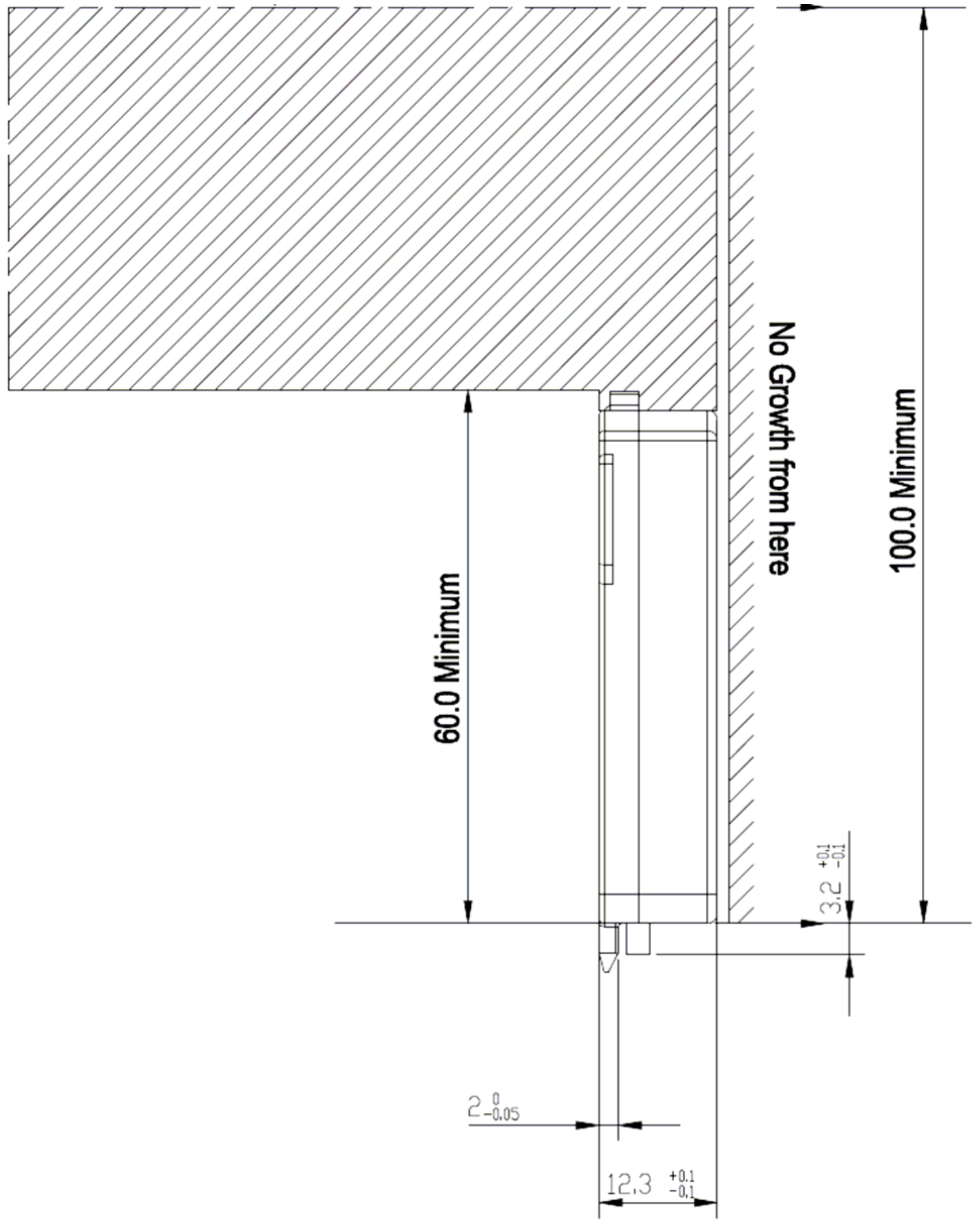


Figure 13-3 DC Form Factor Housing Dimensions – Side View

* Hatched Area Dimensions for connectors, antennas, or large components

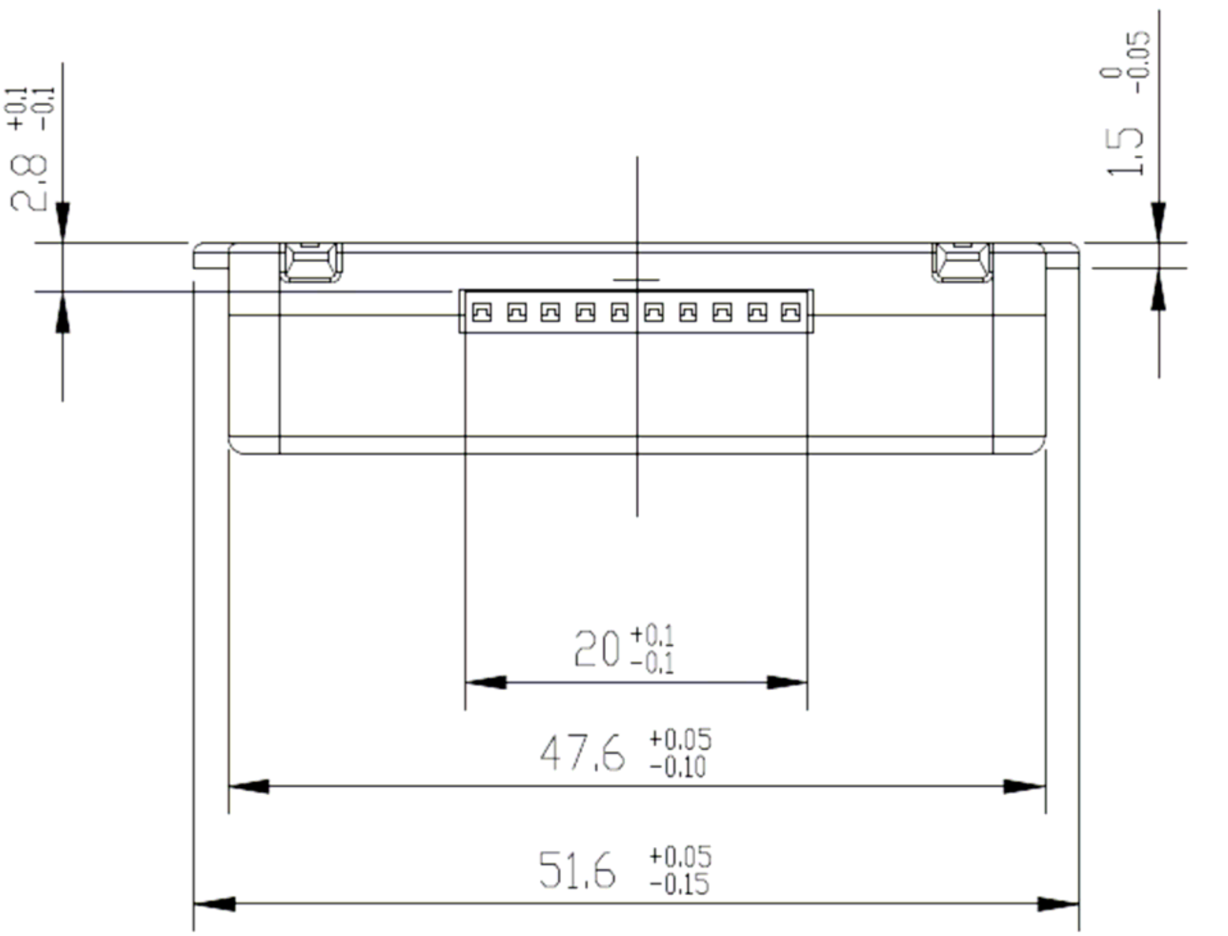


Figure 13-4 DC Form Factor Housing Dimensions – End View

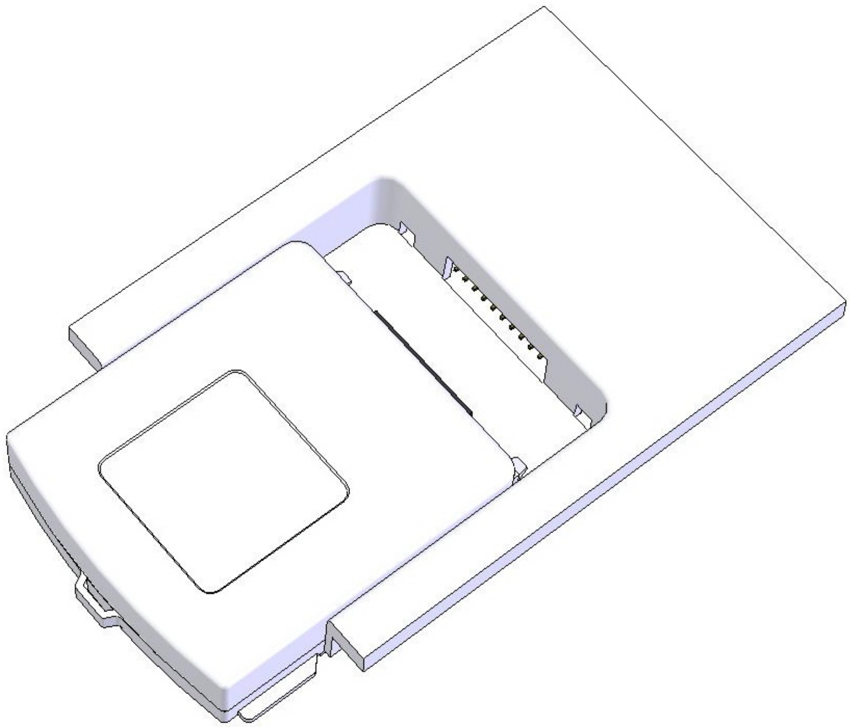


Figure 13-5 DC Form Factor Housing Visualization

13.2.2.3 Weight

The maximum weight for a DC form factor cartridge is 40 grams.

13.2.2.4 Housing Materials

UCM housing materials must be appropriate for the technology contained within and the environment where the device is expected to operate. UCMs are expected to operate in an indoor environment protected from weather. If SGD operates in a severe environment (i.e. temperatures, UV, Chemicals, etc.) it should provide necessary protection.

~~Housing~~If SGDs are UL Recognized Class 2 supply (IEC SELV), then the housing flammability rating of UCMs can be UL94-HB. Otherwise UCM housings must be UL94-V0.

~~should be a minimum of UL94 HB, unless the UCM has~~These are minimum requirements, and UCMs with internal characteristics that require a higher degree of protection may require higher flammability stadards.

13.2.2.5 Connector Type

The interface connector on the SGD is a 10-pin 2mm pin header.

Numerous parts are available including custom heights and lengths. Sample straight and right angle parts: Harwin ~~M22-2351005~~M22-2531005 Harwin M22-2511005 Samtec TMM-110-03-S-S-RA

Gold coatings are required on UCM to SGD interface for reliability.

UCMs will provide a mating 2mm receptacle, Harwin Part Number M22-7131042 or equivalent.

Alternate mating part SAMTEC SQT-110-01-L-S-RA

13.2.2.6 Pin Assignments

The pin assignments of the connector are defined as follows.

Pin	Signal "A" Device
1	SELECT*, must be held high when not communicating with UCM
2	ATTENTION*
3	SCLK
4	MOSI
5	MISO
6	RESET*
7	+3.3V Power
8	Signal Ground
9	Reserved
10	Reserved

13.2.2.7 Electrical Interface Levels

All signals are 3.3V Logic Levels. Logic '0' is defined as less than 0.8V, and Logic '1' as greater than 2.2V.

13.2.2.8 Signal Timing

Representative signal timing is shown for SPI byte transfers:

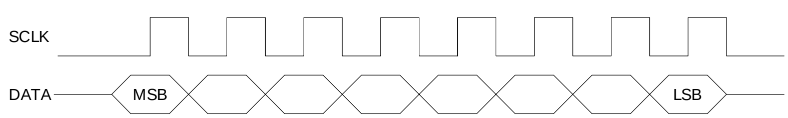


Figure 13-6~~3~~ SPI Mode 0 Bit Timing

This specification supports SPI Mode 0 only.

Data changes on the falling edge and should be read on the rising edge.

Data (MISO) from the UCM is high impedance when SELECT is high.

13.2.2.9 Interface Circuits

Circuit(s)	**Direction**	**active state**	**Purpose**
Signal Ground	*n/a*	*n/a*	Establishes a common ground potential
+3.3V Power	From SGD	n/a	Provides +3.3V power for radio use
MOSI	from SGD	High	Carries SGD SPI data to UCM
MISO	From UCM	High	Returns UCM SPI data to SGD, High Z when SELECT is high.
SCLK	From SGD	Rising Edge	SPI clock signal from SGD
SELECT*	From SGD	Low	Low during each byte of SPI data transfer. Normally high. SGD can assert (low) to request data transfer with radio.
ATTENTION*	From UCM	Low	Signal from UCM Requesting an SPI data transfer
RESET*	From SGD	Low	Signal from SGD negated after power-on. Open collector/drain output.
Reserved (Factory Test)	N/A	N/A	Two signals for SGD test use at factory only. UCM must not connect to these pins.

13.2.2.10 Data Transfer Protocol

ATTENTION*

This signal is asserted (driven low) by the UCM whenever it is ready to initiate a data byte transfer across the SPI. In general it is expected that ATTENTION* will be asserted during the entire time of a message from the UCM and the SGD response (if any).

SELECT*

This control signal is driven low before each message is transferred via SPI, and driven high after the message transfer is complete. The SGD normally parks this line high. When SELECT is high, the UCM MISO data line must be in a high impedance state. Separate SELECT signals are provided to each UCM.

In a secondary usage, the SGD may initiate data transfer to a UCM (in the course of a data transfer from one UCM to another for instance). In this type of transfer, the SGD can drive the SELECT signal low to request ATTENTION* from the UCM. The UCM can acknowledge this request by asserting (driving low) the ATTENTION* signal (initiating a data transfer).

In a tertiary usage, the SGD may use this signal to determine presence of the UCM. The SGD drives the SELECT line low. If the SGD detects the UCM pulling the ATTENTION* line low the SGD confirms the presence of the UCM, and then releases the SELECT signal to the high State.

RESET*

On power-on, the SGD will assert Reset* (drive it low) for a minimum of 100mS. This signal is open-collector (open-drain); it is the responsibility of the UCM to provide any pull-up, if required. Separate Reset* signals are provided to each UCM.

13.2.2.11 Clock and Data Rate

Data rate is controlled by the SPI SCLK. SCLK is generated by the SGD. Standard data rates are defined in section 5.1.3 .

13.2.2.12 Multiple Slots

SGD's may, but are not required to provide multiple slots.

Select, Attn, Reset must be provided to each slot.

Power, Common, Clock, MOSI, MISO are common among all slots.

SGD device manufacturers may allocate bandwidth between the slots as necessary.

13.3 Data Link

The SPI is physically capable of transferring data in both directions simultaneously. However, data is only sent in a single direction at a time. The first side to assert the signal line is the sender. The receiver will always send a 0xFF character during data transfer.

If a node is detecting data on the receive line at the same time it is transmitting it has 2 choices.

1) Ignore it, at end of transmission when the node does not reply with a <ACK> or <NAK> the sending device will try again.
2) Read the incoming message and buffer it for essentially full duplex communication. Attempt to <ACK> or <NAK> for the message when message is complete and line becomes open. This option requires a more complex state machine that also keeps track of the state of both transfer directions, and is not a requirement.

13.3.1 Messages

13.3.1.1 Frame Structure

The message frame structure is defined in section 3 . Message transfer is framed by the SELECT* and ATTENTION* lines.

13.3.1.2 Message Synchronization (Frame Delimiting)

Synchronization (detection of the start of a message) is achieved by the use of the SELECT* and ATTENTION* lines. Message transfer occurs when the UCM has asserted ATTENTION*, the SGD has asserted SELECT* and the SGD begins to clock data. The order of SELECT* and ATTENTION* depends on which device is initiating the data transfer.

13.3.1.3 Message Filling (Inter-message byte filling)

When a device is not sending a message, but SPI transactions are occurring, devices will fill using the special fill byte 0xFF.

13.3.1.4 Command/Response Encoding

Message IDs and responses are encoded as binary bytes. The application layer will define commands to support the application requirements.

13.3.1.5 Checksum Calculation

The checksum calculation is defined in Appendix “D”.

13.3.1.6 Master/Slave

13.3.1.7 The SGD is the master of the SPI bus. It generates the SPI clock and drives MOSI. Flow Control

The UCM controls data flow by using the ATTENTION* line. Flow control is achieved by the SGD through its’ control over the SPI data transfer.

When there are multiple messages queued to send, the SGD or UCM will wait until the prior message has been acknowledged, Not Acknowledged, or the Inter-Message timeout has expired before initiating the next message.

When a UCM or SGD receives a message to send from it’s application while it is in the process of receiving a message, it will Acknowledge or Not acknowledge the message it is receiving. Allow SELECT*/ATTENTION* to be de-asserted, and then initiate transfer of new message.

Note: To support separation of layers this allows multi-threading. This does not require an application to support multi-threading, but creates a mechanism to allow it if desired.

13.3.1.8 Error Detection and Recovery

13.3.1.9 SGD Error Detection and Recovery

The following errors shall be detected at the SGD device data link level:

- Check byte error;
- Invalid Length - Upon receipt of the initial 4 bytes of header, if the length is out of range( ) the SGD will de-assert SELECT*;
- Inter-message time-out. – When waiting for an ACK, if no response is received in 100 milliseconds, the SGD times out and returns to IDLE state. SGD may then attempt to transfer message again. This does not generate a <NAK>; and
- Message Initiation failure. - When SELECT* has been asserted, if no response is received in 100 milliseconds, the SGD times out, de-asserts SELECT* and returns to IDLE state. SGD may then attempt to transfer message again. This does not generate a <NAK>.

Additional error checking shall be performed at the SGD device application level, including data out of range.

13.3.1.10 UCM Error Detection and Recovery

The following receive errors may be detected at the UCM data link level:

- Invalid byte;
- Check byte error;
- Intra-message time-out error, When a message has been initiated by either device and is ready to send as indicated by both Select and Attention being asserted, and an insufficient number of bytes has been clocked by the SGD (including no bytes). If this does not occur in 500 milliseconds, the UCM times out, de-asserts ATTENTION* and returns to IDLE state. This does not generate a <NAK>
- Invalid Length - Upon receipt of the initial four bytes of header, if the length is out of range the UCM will de-assert ATTENTION*;
- Extra Bytes - If the SGD continues to clock beyond where the check-byte was expected, the UCM de-asserts ATTENTION*;
- Inter-message time-out. – When waiting for an ACK or when ATTENTION* has been asserted, if no response is received in 100 milliseconds, the UCM times out and returns to IDLE state. UCM may then attempt to transfer message again. This does not generate a <NAK>; and
- Message Initiation Failure. – When ATTENTION* has been asserted, if no response is received in 100 milliseconds, the UCM times out, de-asserts ATTENTION* and returns to IDLE state. UCM may then attempt to transfer message again. This does not generate a <NAK>.

Additional error checking may be performed at the UCM application level

13.3.2 Operation

13.3.2.1 Transaction Sequence

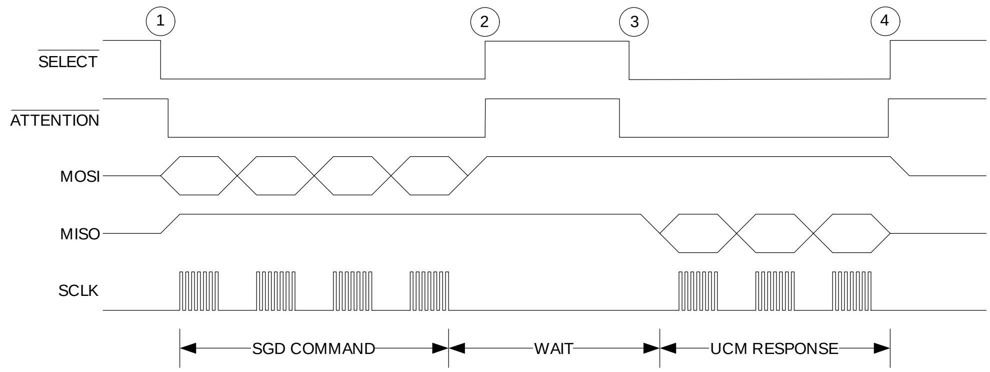


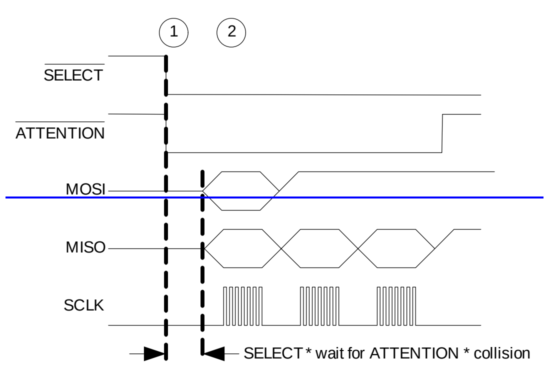
Figure 13-74 SPI Transaction Sequence

1. The SGD begins the transaction by setting SELECT* low. The UCM then sets ATTENTION* low. The SGD then starts clocking out data.
2. After clocking out the packet, the SGD sets SELECT* high. The UCM then sets ATTENTION* high.
3. When the UCM has a response ready, it sets ATTENTION* low. The SGD sets SELECT* low and then starts clocking data.
4. When the packet has completed, the UCM sets ATTENTION* high. The SGD also sets SELECT* high.

For communications initiated by the UCM, the sequence is reversed.

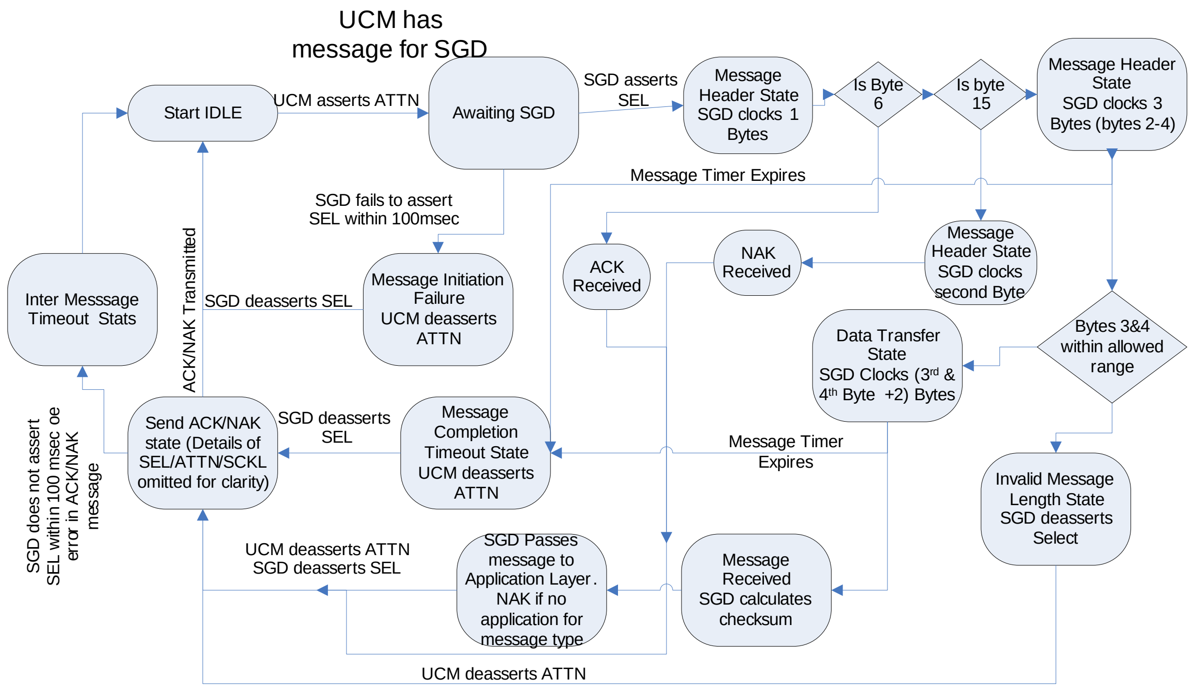
~~13.3.2.2 Collision Detection~~

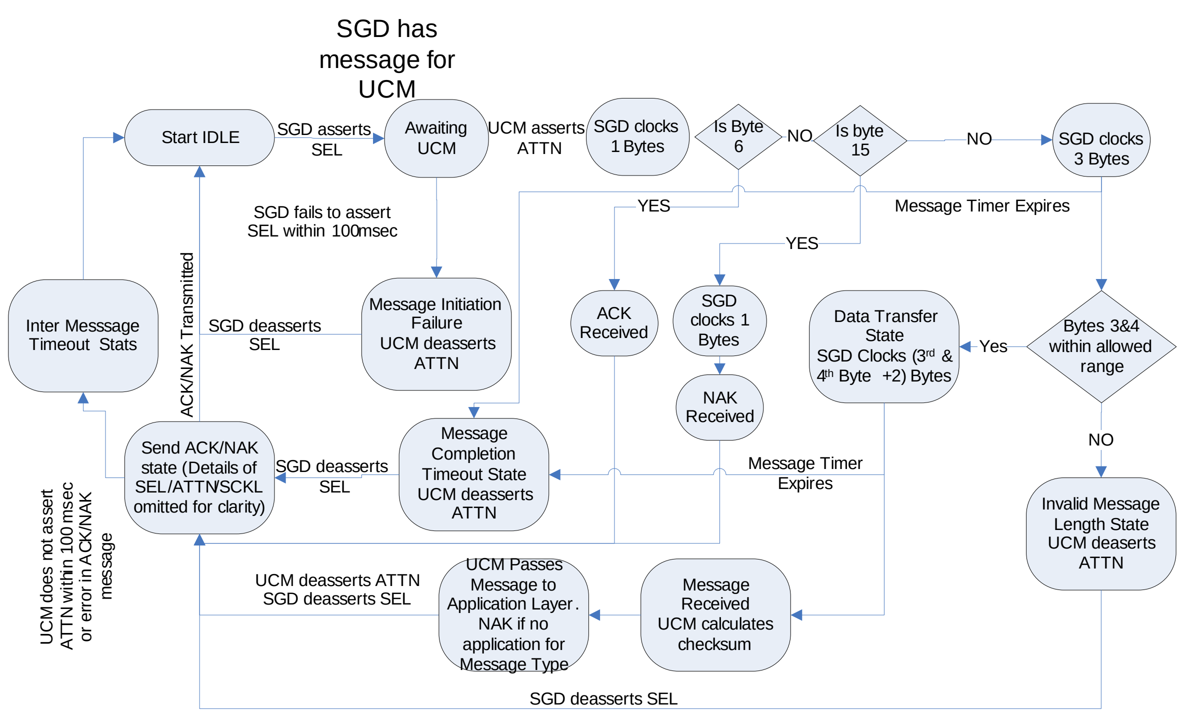
~~The SGD, as the SPI master may experience situations where SELECT* and ATTENTION* are both asserted at about the same time. Link layer retries with randomized delays shall be used to recover from such events.~~



~~Figure 13-5 SPI Collision Detection~~

13.3.2.2 SPI Data Transfer State Machine





This state machine diagram describes the behavior of both the UCM and A-SGD when transferring data over the U-SNAP interface. Data transfer can be initiated on either side by asserting the appropriate signaling line, ATTENTION for the UCM and SELECT for the SGD.

13.3.2.3 SGD Transmitter Operation

The SGD device transmitter must perform the calculation of the checksum to be transmitted with each message, encode both the message and the checksum, and queue messages for transmission.

If the SGD has a message to send it will assert SELECT.

Upon UCM asserting ATTENTION, the SGD will clock 1 bytes.

If message was ACK, SGD will de-assert SELECT.

If message was NAK, SGD will clock one additional byte and de-assert SELECT.

Otherwise

SGD will clock remainder of the message and de-assert SELECT.

SGD will then await an ATTENTION from UCM, to receive and ACK or NAK.

If ACK received,

SGD will de-assert SELECT, and return to idle.

If NAK received or a new message is received from UCM,

SGD transmitted notify application of transmit failure.

SGD may attempt to transfer again.

It is recommended that developers put an algorithm in place to handle NAKs/Timeouts. A retry counter, or an increasing delay between attempts, or evaluating the message at the application level to see if it is still valid are options.

13.3.2.4 SGD Device Receiver Operation

The SGD Device receiver must perform the following functions:

- Detect the request to transfer from UCM,
- Detect message length,

- Decode the incoming message,
- Generate ACK or NAK messages as appropriate, and
- Route messages to the appropriate application task.

The SGD Device receiver initially enters an IDLE state.

If the ATTENTION line is asserted by the UCM, the SGD Device receiver asserts the SELECT line and enters the data transfer state and clocks in one data bytes.

If ACK received,

SGD will wait for UCM to de-assert ATTENTION, then de-assert SELECT, and return to idle.

If NAK received,

SGD will clock 1 additional byte, wait for UCM to de-assert ATTENTION, then de-assert SELECT, and return to idle.

Otherwise,

The SGD will clock the third and fourth bytes is the (message count bytes). [If this is invalid an error is generated and receipt aborted]

The SGD then clocks in the balance of the message (data bytes plus the check sum byte,). The receiver then awaits the de-assertion of ATTENTION. Upon de-assertion of ATTENTION the Receiver de-asserts SELECT.

The receiver then computes the checksum, and then queues an Acknowledgement or a Not Acknowledgement for transmission. The receiver then forwards the message to the application (if correct) or discards (if incorrect) and returns to IDLE state.

13.3.2.5 UCM Operations

The UCM performs the same functions as the SGD, with the following changes:

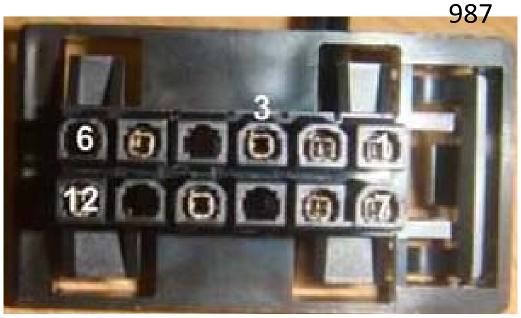
- The actions of SELECT and Attention are reversed.
- The UCM has to monitor for Message Timeout.
- The UCM relies on the SGD to clock the bytes.

14 Appendix B – AC Form Factor

14.1.1 Physical Form

The connector shown in **Error! Reference source not found.**~~Figure14-1~~ provides sufficient clearance for the 120/240Vac signals. In the case of 120V appliances, AC+ ~~is~~shall be the hot connection and AC- ~~is~~shall be the neutral connection. In the case of 208/240V three wire appliances, AC+ ~~is~~shall be phase A and AC- ~~is~~shall be phase B. The connector includes protective sleeves to cover the energized pins, has some circuits removed to increase clearance around the AC power, and is polarized so that the mating device can only be plugged-in one way. The connector part that would go on the SGD is available from Molex as 44516-1006, using pins (quantity eight) Molex 39-00-0039. The mating part, which would be used on a communication module that plugged-in at a right angle, is Molex 43459-0051.

<mechanical details, second source and connector detail drawings to be inserted here>



1	Data- (RS-485)	7	Data+ (RS-485)
2	No Connection	8	Signal Ground
3	Reserved for vendor-specific use	9	No Connection
4	No Connection	10[1]	Earth Ground
5[1]	120VAC Line 2	11	No Connection
6	No Connection	12[1]	120VAC Line 1

Note: AC power pins utilize 18AWG wires. All others utilize 22AWG

Figure14-1 AC Connector Form Factor (Appliance Side Shown) and Pin-Out

14.1.2 AC Power

UCMs must be designed to operate normally over a voltage range from 10% under the nominal service voltage to 20% over the nominal voltage. This equates to 96 to 288Vac for 120 to 240V systems. UCMs must be auto-ranging. Current consumption on the AC lines may not exceed the limits indicated in Table 5-3~~Table 5-3~~. SGDs providing this connection may optionally choose to limit the current to these levels.

14.1.3 Serial Electrical Interface

This is an EIA RS-485 2-wire connection on pins 7 (Data+) and 1 (Data-), operating at a nominal voltage of 5V as illustrated in Figure 14-2~~Figure 14-2~~.

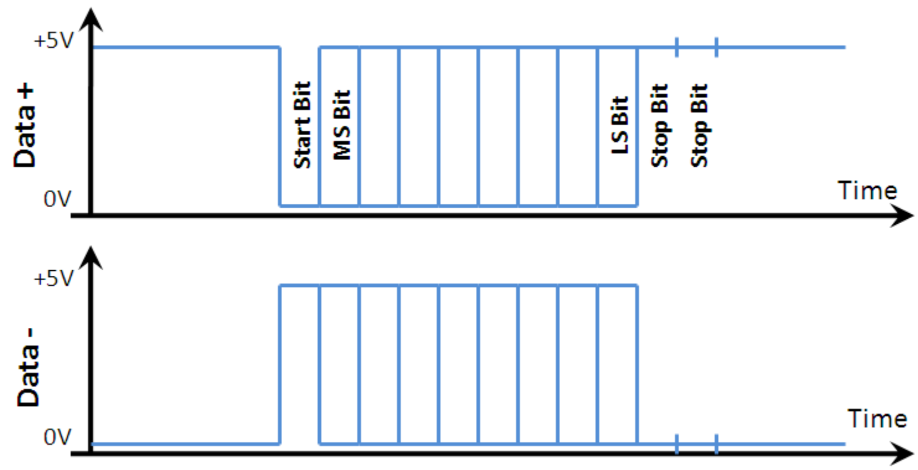


Figure 14-2 Typical RS-485 Polarity and Byte Transfer

The RS-485 interface is anticipated to employ a driver IC. The use of such a driver provides protection against ESD and other transients and potential noise problems. RS-485 is multi-drop capable, allowing the serial port going into the appliance to connect to multiple sub-systems inside the appliance if desired. The default state for this bus is a logic "1", with the bias provided by the SGD via a 1K ohm pull-up resistor on the Data+ line to +5V and a 1K pull-down resistor to GND on Data- provided by the SGD as illustrated in Figure 14-3~~Figure 14-3~~.

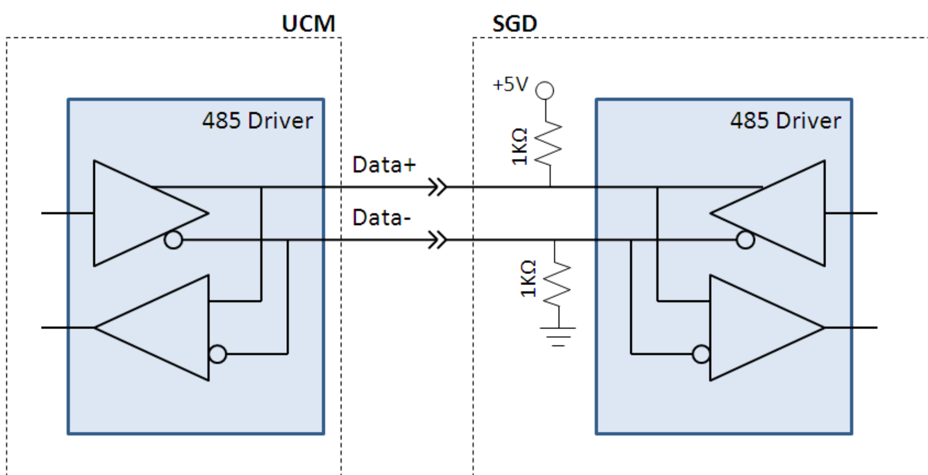


Figure 14-3 RS-485 Connections

The default baud rate on the interface is 19.2KBaud with one start bit (0), eight data bits, no parity, and two stop bits (1's). Transmitters must send two stop bits, receivers must tolerate one or more.

14.1.4 Obtaining Message Sync

Because the serial message structure does not employ a special start character, bus idle (logic "1" state) for 20[mS] or more shall indicate the start of a new message. During a packet transfer, transmitters must assure that the inter-character delay is less than this time. Receivers are only required to report time-outs at the message level, per the T_{ML} specification in Table 3-3~~Table 3-3~~. Receivers are not required to measure inter-byte delay, other than when searching for the start of a new message. There is no LINK NAK error code associated with inter-byte delay.

15 Appendix C – Fletcher Checksum

The checksum method for this protocol is a "Fletcher's Checksum". The two bytes are coded and decoded as follows:

15.1 Calculating the Checksum

Initialize check1 to 0xAA and check2 to 0x00

Use a temporary variable to read in bytes, call it "checktemp"

Loop through the message (first message type byte to end of payload) to find the check numbers. This loop excludes the checksum bytes as they are not yet created.

check1 = (check1 + checktemp) Mod 255

check2 = (check2 + check1) Mod 255

Find the bytes to append in the following manner:

mscheckbyte = 255 - ((check1 + check2) Mod 255) (the MSB of the completed checksum)

lscheckbyte = 255 - ((check1 + mscheckbyte) Mod 255) (the LSB of the completed checksum)

15.2 Decoding the Checksum

Initialize check1 to 0xAA and check2 to 0x00

Use a temporary variable to read in bytes including the checksum, call it "checktemp"

Loop through the message to find the check numbers.

check1 = (check1 + checktemp) Mod 255

check2 = (check2 + check1) Mod 255

Once the entire message including the two checksum bytes have been analyzed, check1 and check2 should equal 0x00.

15.3 Example VB Code

Creating Checksum:

```
checkbyte1 = 170 '0xAA
checkbyte2 = 0   '0x00

For I = 1 To Len(senddata) Step 1
    checktemp = Asc(Mid(senddata, I, 2))
    checkbyte1 = (checkbyte1 + checktemp) Mod 255
    checkbyte2 = (checkbyte2 + checkbyte1) Mod 255
    checktemp = 0
Next I

mscheckbyte = 255 - ((checkbyte1 + checkbyte2) Mod 255)
lscheckbyte = 255 - ((checkbyte1 + mscheckbyte) Mod 255)
```

Decoding Checksum

```
Public Function GoodChecksum(incoming As String) As Boolean
checkbyte1 = 170
checkbyte2 = 0

For I = 1 To Len(incoming) Step 1
    checktemp = Asc(Mid(incoming, I, 2))
    checkbyte1 = (checkbyte1 + checktemp) Mod 255
    checkbyte2 = (checkbyte2 + checkbyte1) Mod 255
    checktemp = 0
Next I

If checkbyte1 = 0 And checkbyte2 = 0 Then
GoodChecksum = True
Else
GoodChecksum = False
End If
```

16 Appendix D – Guideline for Computing Average Price

This appendix explains the calculation of the Average Price (AP) from which the relative price of section 4.0 is computed.

Electric rates structures, today, are very complex and vary in every jurisdiction. Common elements of rate design consist of components like fixed charges per month (including customer charge, flat tax or levy fees), a unchanging prices per kWh (often to recovery fixed costs charge on a per kWh basis), peak kW demand charges, a percent tax (or tax equivalent) adder on certain charges, and worst of all, block rates that cause a higher (or lower) rate kWh when usage during a billing period moves above a certain quantity of use. (Some rates even have multiple blocks).

Note: Average price by state and sector are here: http://www.eia.gov/cneaf/electricity/epm/table5_6_b.html. However, these average prices are inclusive of all fixed and flat rate charges. AP based on more than TVC is desirable from a customer perspective but there is no way to reconcile block rate designs with average price since two customers can have significantly different average prices from a monthly bill perspective. Also observe that averaged at the state level, the total price per kWh varies between 8 and 28 cents per kWh; this should emphasize the problem of trying to communicate actual price and have anything but a complex application make sense of it. Everyone can understand price relative to the expected monthly bill.

The relative price command only applies to the bill components that relate to price changes as a function of time during a day or hour. ("Time Varying Charges, TVC") Most commonly TVC charges apply to the cost of energy generation, but in some rate designs charges for the fixed cost of distribution may also be time varying. TVC exclude kW, demand charges if any. The purpose of command 0x07 is to convey the bundled price of all TVC which affect the retail customer's bill. Thus the service provider responsible for conveying price signals via the UCM combines all TVC even if the service provider only provides energy or distribution services. The following instructions assume that block rate price increase or decreases have no relationship to time only the usage in a billing cycle. (This is the most common outcome. It has the effect of say the flat price per kWh is $X/kWh for the first Q use and then $Y/kWh for the next block of usage. Depending on the time varying price structure the bill calculation add or subtracts dollars depending on the block rate structure, but not dependent on when in a day usage was consumed.)

The following explanation gives examples with prices quoted in mills per kWh. (This is numerically equal to $ per MWh the unit of price in wholesale markets. Retail customers are commonly quoted prices in cents per kWh; cents per kWh equals mills per kWh divided by ten.)

A utility or service provider using this command should be willing (or regulated) to provide to their customer a basis for the average price ("AP") per kWh that pertains to TVC; relative prices equals current price divided by AP. The intent of the AP is to convey the price per kWh, pertaining to TVC, that produces an average bill amount. Example let's say a bill consists of 3 components, a fixed charge per month of $11 (including flat taxes), a flat rate of 41.2 mills (40 mills per kWh plus a 3% tax) and an AP of TVC equal to 82.4 mills (80 mills plus 3% tax). Lets say the customer uses 800 kWh per month. The the intent of AP is to convey that the customer on the rate would receive a bill of $65.92 for TVC, **IF** the customer's usage pattern matches the statistical average load shape with respect to time. ($65.92 = $82.4/MWh time 0.8 MWh). Note the customer's total bill would be $109.88 = $11 + $32.96 + $65.92 Thus if the customer use more energy in higher-priced hours than the average customer they would expect a higher bill being on this time varying rate design.

Explanation for non-regulated utilities.

The above task is a straight-forward exercise today for any regulated utility (public or private); ratemaking 101 if you will. However, in a smart grid world with multiple rates and multiple providers, the best we are likely to do is set AP at a value close to the average TVC, more importantly at a level the make the relative prices (dynamic prices divided by the AP) as indicative of the correct relative price to the consumer.

For regulated utilities most customer are a flat rate price per kWh that represents the sum of fixed and TVC (to the utility). In the creation of a time varying rate the first step is to be explicit about how much of the flat rate will remain at flat price per kWh and how much will be collected through time a time varying rate. Through load research, for customers of a given class, say residential, a known load shape for the average customer is known. Load shape is the percent of total energy use per hour for each hour of the year. This shape can be based on all customers or just customers on a flat rate design. Expected revenue from a given time varying rate is calculated using this model. Consider a two price time-of-day rate. Low rate period is 10p to 6a every day and high price 6a to 10p every day. By looking at the load shape you would determine that 25% of the energy is use in the low rate period. If flat rate price is 120 mills and 80 mills are to be collected in TVC, then to achieve and AP of 80 mills in TVC and the low price is 50 mills, then the high price is 91.67 mills. Relative price when the low price is in effect is 0.563. Relative price when high price is in effect is 1.146. Note when a 3% tax is added the relative prices remain unchanged. This example is the simplest type of design. Introducing a third price period allows much more flexibility in designing the behavior desired from the customer and also the rates marketability. Critical peak pricing and real time pricing introduce a new variable in that the specific period when a price will be available is not know ahead of time. Rate design is based on expected pricing outcomes.

AP for TVC was assumed at 80 mills. This type of rate designed is called "revenue neutral" because both the flat rate and time-varying rate collect are designed to collect 120 mills per kWh for the average customer. It is expected that the customer with time varying rates will shift some loads to the low price period thus creating a net billing savings to the customer. However there are economists would argue time varying-rates should be lower than revenue neutral to reflect that customers that don't wish to deal with peak supply cost problems should pay a premium for the risk burden they put on the energy supplier relative to someone who faces time-varying pricing. Critical peak pricing and real time pricing introduce a new variable in that the specific period when a price will be available is not know ahead of time. Rate design is based on expected pricing outcomes.

The complexity of rate designs in the last paragraph raise a number of issues about how to determine AP. For example is AP based on all customers 1) in a given region, 2) of a given provider, 3) on a specific rate, before or after shifting is accounted for. If the rate is based on real time rates what index or forecast is used to determine average rates.

Summary

With regard to this standard and this command there are no absolute rules that can be workable against all the future outcomes of regulation and market forces. Imagine coming with a relative price of for a coach round trip airline seat between Washington DC and Tampa Florida in 1977 before deregulation in October 1978. Average price would have been fairly easy to determine. Relative prices for the Wednesday before thanksgiving would have been relatively easy to determine also. Today we might guess the price at $250, and feel that $200 would be a great price and $500 a very high price. We would feel fortunate if someone even gave us a relative constant metric of average price.

This analogy is only mentioned to show how pricing models to improve asset utilization can change dramatically over time. However unlike airplane flights which are expensive, relatively infrequent transactions, electricity use is like a leaky faucet-- continuous use with small cost incurred during most periods. This command only works, a machine to machine interface, if the consumer can use the signal to manage the size of their electric bill. A service provider that games the signal will eventually be found out, if a large risk of damage to market credibility.

<u>Thus the rules for determining AP can be summarized as follows</u>

-AP shall be based on TVC only

-AP price is set to indicate the price that will produce the average bill for TVC.

-In the near term, AP should be calculated in a manner similar to rate designs for regulated entities.

-In the future, AP should be set (ideally held constant for at least a quarter if not a year) at a level that provides intuitive messages via the relative price signals to customers.