HART Protocol Specification

HART Field Communications

HART®

COMMUNICATION FOUNDATION

# Data Link Layer Specification

**HCF_SPEC-81, Revision 8.0**

**Release Date: 18 April, 2001**

**Release Date: 18 April, 2001**

**Document Distribution / Maintenance Control / Document Approval**
To obtain information concerning document distribution control, maintenance control and document approval, please contact the HART Communication Foundation at the address shown below.

**Copyright © HART Communication Foundation**
This document contains copyrighted material and may not be reproduced in any fashion without the written permission of the HART Communication Foundation.

**Trademark Information**
HART® is a registered trademark of the HART Communication Foundation, Austin, Texas, USA. Any use of the term HART hereafter in this document, or in any document referenced by this document, implies the registered trademark. All other trademarks used in this or referenced documents are trademarks of their respective companies. For more information contact the HCF Staff at the address below.

Attention: Foundation Director
HART Communication Foundation
9390 Research Boulevard
Suite I-350
Austin, TX 78759, USA
Voice: (512) 794-0369
FAX: (512) 794-3904
http://www.hartcomm.org

# Table of Contents

# Table of Figures

# Index to Tables

# Preface

This preface is included for informational purposes only.

HART 6 represents the first major revision to the Protocol in nearly 10 years. However, developers, manufacturers and users of the HART compatible devices can be assured that fundamental HART principles are maintained. HART 6 is backward compatible with HART 5 while increasing the capabilities of 4-20mA devices and systems.

The Data Link Layer, at the core of the HART Protocol, establishes the format of HART messages and ensures the reliability of communications. The principle changes to this specifications are non-functional. In other words, the biggest changes are in the format of this document and not in the substance of its requirements. The document as a whole has been reformatted to include new sections: Preface, Introduction, Scope, References, Definitions, and Symbols/Abbreviations. In addition the requirements have been reorganized to align with ISO guidance on the specification of a protocol layer. As a result, the following major sections are now found in this specification:

- Logical Link Control (LLC) - This section describes the upper portion of the Data Link Layer. This section includes primarily Section 2.2 of the previous revision. Two additions were made to the Protocol in this section: frame expansion; and framing.

  - Frame expansion allows the addition of up to three extra bytes between the address and command fields. The function of these bytes are undefined at this time and reserved for use by the HCF.

  - Framing is the process of identifying the different fields in a message. In this revision the algorithm for successful framing of a message is explicitly stated.

- Data Link Layer Services - This section provides a "black box" specification of the Data Link Layer responsibilities and operation. This section includes primarily Section 2.3 of the previous revision. The requirements in this section are clarified, updated to reflect current HART 6 requirements and all the diagrams re-drawn. All of the diagrams in revision 7.1 of the Data Link Layer Specification were scanned images and, thus, not maintainable. The Block Transfer Services are more of a Network or Transport layer function and are moved to the *Block Data Transfer Specification*.

- Medium Access Control (MAC). This specifies the lower sub-layer of the Data Link Layer. This section specifies the bus arbitration rules to be used by HART compatible devices. This section includes primarily Sections 2.4.2 - 2.4.3 of the previous revision. In addition, the overall operation of the MAC sub-layer is specified (see Section 7). This was only implied by the previous revision.

- Physical Layer-Specific Requirements - The operation of the Data Link Layer is largely independent of the underlying Physical Layer. However, Physical Layer characteristics do impact the Data Link Layer. These impacts and the resulting requirements are now isolated in this section. These requirements include time-out values, the character (octet) transmission format and Start Of Message (SOM) and preamble requirements. Requirements for FSK, RS-485, and C8PSK are included.

In addition, Annexes have been added. These Annexes include background information that can help understand the Specifications themselves:

- Annex A explains how the Data Link Layer Services are specified and Time Sequence Diagram notation. This was formerly included in Section 2.3.1 of *Data Link Layer Specification* Revision 7.1
- Annex B provides a tutorial on the diagramming technique used by the finite state machines that specify operation of the MAC sub-layer.
- Annex C provides a derivation of the time-out values specified in Section 8.

As can be seen the functional changes (e.g., frame expansion) are few. However, the reorganization and clarifications to the *Data Link Layer Specification* will be valuable to developers and end users a like.

# Introduction

The Data Link Layer provides a reliable, transaction oriented communication path to and from field devices for digital data transfer. Communications is over twisted pair wire that may be simultaneously carrying 4-20mA signaling. The Data Link Layer will correct for errors due to noise on the communication links by using error detecting information and an Automatic Repeat Request (ARQ) protocol to request the re-transmission of data blocks that may be corrupted by line noise or other disturbances.

This specification describes the Data Link Layer in an implementation independent form so that it may be translated for any processor and any implementation programming language.

HART is a master-slave protocol and is loosely organized around the ISO/OSI 7-layer model for communications protocols (see Figure 1). The Data Link Layer supports the Application Layer above it and requires services from the Physical Layer below it. Furthermore, the Data Link Layer can be divided into two sub-layers: the Logical Link Control responsible for addressing, framing and error detection; and the Medium Access Control that controls the transmission of messages across the physical link.

| | OSI Layer | Function | HART |
|---|---|---|---|
| 7 | Application | Provides the User with Network Capable Applications | Command Oriented. Predefined Data Types and Application Procedures |
| 6 | Presentation | Converts Application Data Between Network and Local Machine Formats | |
| 5 | Session | Connection Management Services for Applications | |
| 4 | Transport | Provides Network Independent, Transparent Message Transfer | |
| 3 | Network | End to End Routing of Packets. Resolving Network Addresses | |
| 2 | Data Link | Establishes Data Packet Structure, Framing, Error Detection, Bus Arbitration | A Binary, Byte Oriented, Token Passing, Master/ Slave Protocol. |
| 1 | Physical | Mechanical / Electrical Connection. Transmits Raw Bit Stream | Simultaneous Analog & Digital Signaling. Normal 4-20mA Copper Wiring |

**Figure 1. OSI 7-Layer Model**

Note: The HART Protocol does not specify a Network, Transport or Session Layer. However, many Host Applications perform these functions. The Protocol Application Layer documents (e.g., the *Command Summary Specification*) provide information regarding the identification, management and configuration of HART networks. In addition, some Host Applications provide session-oriented interfaces that manage the details of Field Device communication (e.g., mapping of device addresses).

The Data Link Layer supports long (5 byte) "unique" and short (1 byte) "poll" addresses. Polling addresses may only be used with Command 0. This allows the HART Protocol to support both point-to-point or multi-dropped communication with field devices. If poll (short form) addresses are used, up to 64 slave devices may be multi-dropped on a single communication link (see Section 5.3.4). If unique (long form) addresses are used, the number of multi-dropped devices is essentially unlimited, and is determined based on the applications required rate of scan of the devices on the communication link (see Section 5.3.2).

The Data Link Layer will also arbitrate access to the field device between a single secondary master device such as a handheld terminal and a single primary master devices such as a control or data acquisition system (see Section 7). The Data Link Layer gives equal communication channel access to both kinds of masters when they are being simultaneously used. The Data Link Layer will not arbitrate between two secondary or two primary masters trying to talk on the same link.

To support the regular transfer of information from field device to master device, the Data Link Layer supports a mode of operation in which field devices periodically broadcast information onto the communication link. A slave device is said to be in "burst mode" when it is providing a synchronous cyclic broadcasting of data, without continuous polling by a master device. No matter how many field devices are on a communication link, only one may be in burst mode.

Information transfer between devices on the communication link is through a defined message format. The entire message is protected by a single parity check product code (sometimes also known as vertical and longitudinal parity checking). Message framing is through a combination of a start of frame delimiter and a message length field (see Section 5.1.7).

The Data Link Layer provides services for data transfer, and communication configuration (see Section 6). Other HART Protocol documents cover the interpretation of data transferred by this Data Link Layer between various field devices and the primary or secondary master.

# 1. SCOPE

This specification defines the HART Protocol Data Link Layer. The Data Link Layer is responsible for the reliable, error free communication of data between HART compatible devices. In other words, this document specifies the rules used by HART products to communicate digital information over a physical link. Figure 2 shows the scope of this specification. This document includes:

- The services provided by the Data Link Layer to the Application Layer and the services required of the underlying Physical Layer are specified. These service constitute a "black box" model of the Data Link Layer requirements. These services are specified with the assistance of time sequence diagrams.

- Logical Link Control (LLC) requirements including the format of HART frames, the structure of HART device addresses (both master and slave); and the error detection coding to be used. In effect, the LLC adds addressing and framing information to the command and data fields supplied by the Application Layer.

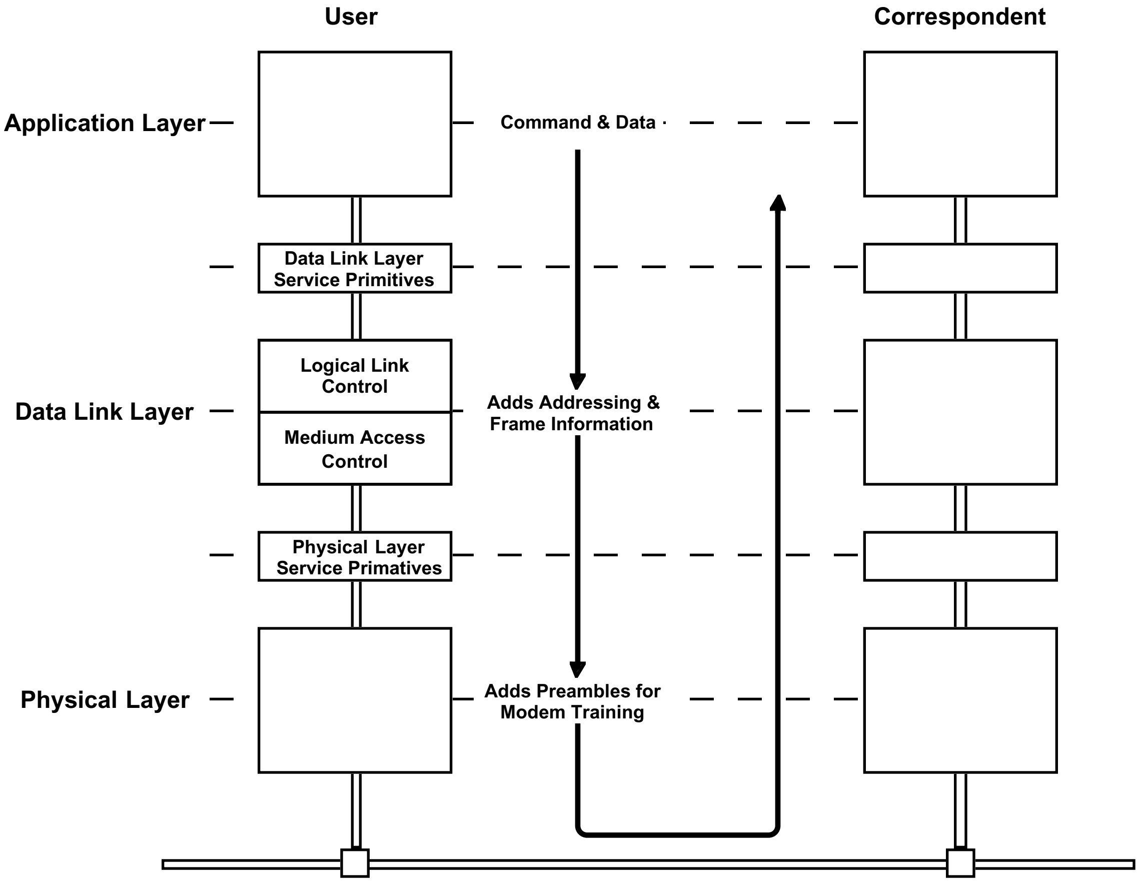
User
Correspondent
Application Layer
Command & Data
Data Link Layer Service Primitives
Logical Link Control
Data Link Layer
Medium Access Control
Adds Addressing & Frame Information
Physical Layer Service Primatives
Physical Layer
Adds Preambles for Modem Training

**Figure 2. Data Link Layer Scope**

- Medium Access Control (MAC) rules ensuring that transmissions by devices occur in an orderly fashion. In other words, the MAC specifies when a device is allowed to transmit a message. MAC specifications themselves are formulated in terms of a state transition diagrams. This notation allows an unambiguous description of the action of the MAC sub-layer.

- The actual timing values required for proper operation of the MAC sub-layer. These timing values directly correspond to Physical Layer performance characteristics (e.g., carrier turn on times). In addition, the Physical Layer requires preambles to be added to the HART Protocol Data Unit (PDU).

The segregation of requirements into these four categories is intended as a frame of reference rather than as a description of an actual implementation. While actual implementations may vary, all requirements in this specification are mandatory.

# 2. REFERENCES

## 2.1. The HART Field Communications Protocol Specifications

These documents published by the HART Communication Foundation are referenced throughout this specification:

*HART Field Communications Protocol Specification.* HCF_SPEC-12

*FSK Physical Layer Specification .* HCF_SPEC-54

*C8PSK Physical Layer Specification .* HCF_SPEC-60

*Command Summary Specification.* HCF_SPEC-99

*Common Practice Command Specification.* HCF_SPEC-151

## 2.2. Related Documents

The following patents, owned by the HCF, form the basis of the HART Protocol and, in particular the HART Data Link Layer.

Warrior. "Dual master implied token communications system" United States Patent 4988990

Warrior. "Dual master implied token communications system" United States Patent 5122794

Warrior. "Dual master implied token communications system" United States Patent 5166678

The HART Protocol uses a "single parity check product code" to ensure error-free communication. The following reference provides an analysis of the reliability of the Error Detection Coding used by the HART Data Link Layer:

> Leung, C. (1983). "Evaluation of undetected error performance of single parity check product codes" *IEEE Transactions on Communications*. Volume COM-31 No.2 pp. 250-253.

The following provides general guidelines for the specification of communication protocols.

> ISO 7498-1 *Information Processing Systems - OSI Reference Model - The Basic Model*

The following reference describes communication specification techniques used in this document including Service Access Points (SAPs) and time sequence diagrams:

> Halsall, F. *Data Communications, Computer Networks and Open Systems*. Third Edition. Addison Wesley. 1992

The following are RS-485 related documentation:

> *HART RS-485 Physical Layer Implementation Definition.* Rosemount Inc. 19 March, 1992

> *HCF General Assembly Meeting Minute*s, Bad Kreuznach, Germany 16-19 June 1997. HCF_MIN-21

> *Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems.* ANSI/EIA/TIA RS-485. Electronic Industry Alliance. 1982

The following reference describes the methods for specifying state transition diagrams used in this document .

> Hatley, D., and Pirbhai, I. *Strategies for Real-Time System Specification*. Dorset House, 1987.

# 3. DEFINITIONS

Definitions for terms can be found in *HART Field Communications Protocol Specification.* Terms used in this document include: Broadcast Address, Burst Mode, Burst Mode Device, Data Link Layer, Master, Multi-drop, Preamble, Response Message, Slave, Slave Time-Out, Software Revision Level, Time Constant, Units Code

Some of the following definitions are included in the *HART Field Communications Protocol Specification*. However, these definitions are critical to the understanding of this specification. As a result, they are included and their meaning amplified. Three types of network connections are used throughout this specification to completely define the HART Data Link Layer requirements.

**Burst-Mode Device**

This is a device which provides a digital response carrying measurement or other data, at regular intervals, without the data being specifically requested, i.e., this device normally functions as an independently broadcasting device. A Burst-Mode Device is defined to be a Slave Device with burst mode capability (hence the use of the word "mode" in describing the device type). When such a mode is enabled, the Slave Device is said to "be in burst mode".

**Field Device**

A Slave or Burst-Mode Device.

**Master Device**

A Master Device is responsible for initiating, controlling and terminating transactions with a slave device or a burst mode device. A distinction is made between master devices into Primary Masters and Secondary Masters in order to allow the simultaneous use of two Master Devices on a HART communication link. The same protocol rules are followed by a Primary Master and a Secondary Master except for customizing time-outs that differentiate between them.

**Slave Device**

This is a device which accepts or provides a digital message carrying measurement or other data, but only when specifically requested, i.e., this device always functions as a slave in a master/slave relationship.

**Time Sequence Diagram**

A diagram used to illustrate the interrelationship between the Protocol services. The protocol layer of interest and the lower, intervening layers are treated as a "black box". The internal workings of these layers are not shown on this diagram. The time sequence diagram shows the interactions between the service primitives over time.

Sometimes referred to as a Message Sequence Diagram.

**Unique Identifier** The concatenation of the Manufacturer ID, Device Type and Device ID. Every field device manufactured with a given Manufacturer ID and Device Type must have a different Device ID. A subset of the Unique Identifier is used in the Long Frame Address.

# 4. SYMBOLS/ABBREVIATIONS

**ACK** Acknowledge. The response of a slave device to a master message (i.e., a slave to master message).

**BACK** Burst Acknowledge. A slave message transmitted to a master without a corresponding master request (i.e., without an STX).

**BPS** **B**its **P**er **S**econd

**BT** **B**urst **T**imer. Used by a Burst-Mode Device to trigger the transmission of a BACK. When BT lapses (i.e., reaches 0) a BACK is transmitted.

**BURST** A Boolean state variable used by a device. If BURST is set (i.e., TRUE), then the network is in burst mode.

**COUNT** The master MAC is required to perform communication retries. This state variable counts the number of retries transmitted by the master MAC.

**HCF** **H**ART **C**ommunication **F**oundation

**HOLD** The amount of time allowed for a Master Device to begin a transmission once bus arbitration has granted that master access.

**HOST** A state variable used by a Burst-Mode Device to indicate the master address to be used in the next BACK.

**LLC** **L**ogical **L**ink **C**ontrol. The upper portion of the Data Link Layer responsible for the reliability of HART communications. The LLC determines the structure of the HART PDU's, addressing, and error detection requirements

**LSB** **L**east **S**ignificant **B**yte. The LSB is always the last byte transmitted over a HART data link.

**MAC** **M**edia **A**ccess **C**ontrol. The lower part of the Data Link Layer responsible for orderly access to the Physical Link by HART devices. The MAC governs when a device is allowed to transmit a message.

**MSB** **M**ost **S**ignificant **B**yte. The MSB is always the first byte transmitted over a HART data link.

**MSG_PENDING** A Boolean state variable used by a master to indicate that a message is enqueued and awaiting transmission.

**PDU** **P**rotocol **D**ata **U**nit. This refers to the format and content of a message exchanged between corresponding layers in two field devices (e.g., the Data Link Layer in one device with the Data Link Layer in another device).

RCV_MSG The Receive Message primitive.

**RT1** Link Quiet Time. The maximum interval of time that ensures that no on going transaction will be interrupted by a master which is seeking access to the communication link. If no intervening transmissions occur, the link has come free and may be accessed.

The Link Quiet Time is longer for a Secondary Master to ensure that no collisions occur when a Primary and a Secondary Master are connected to the link simultaneously.

When using RT1, Burst-Mode Slaves must always use the short Primary RT1 value.

**RT2** Link Grant Time. The time a master grants the other master network access to begin a transmission. In other words, the Link Grant Time "RT2" delays a subsequent master transmission long enough to detect a carrier/ start of message due to the other masters communication. If no transmission from the other master is heard during this interval, the granting master is free to initiate another transmission.

**SAP** **S**ervice **A**ccess **P**oint. The services supplied by one communication protocol layer and used by another protocol layer. These encapsulate the functionality supported by a protocol layer.

**SOM** **S**tart **O**f **M**essage. For the FSK Physical Layer the SOM is detected by the error-free, consecutive reception of two preambles followed with a valid delimiter.

**STO** **S**lave **T**ime-**O**ut (formerly TTO, Transmitter Time-Out). This is the maximum time allowed for a Slave Device to begin its transmission.

**STX** Start of a transaction. A master to slave message. A transaction consists of an STX and a corresponding ACK.

**XMT_MSG** The Transmit Message primitive.

# 5. LOGICAL LINK CONTROL

The HART Data Link Layer provides an acknowledged connectionless service to transfer data between HART compatible devices. This results in a simple HART LLC because the overhead associated with establishing a logical connection is not present. In fact, this characteristic of the HART Protocol carries through to the Application Layer's requirement for commands to be autonomous and stand-alone (see the *Command Summary Specification*).

The HART LLC specification consists of:

- The definition of the HART Protocol Data Unit (PDU) or message frame;
- The algorithm used to frame the PDU;
- Requirements for the addressing of devices;
- The error detection coding employed for all HART Communications.

## 5.1. The HART PDU (Frame Format)

All data transferred between entities involved in the Protocol is transferred in the form of **frames**. A frame is an encapsulation of user data in control and addressing information.

The HART PDU consist of 7 fields (see Figure 3): Delimiter, Address, Expansion Bytes, Command, Byte Count, Data, and the Check Byte. The Expansion Bytes and Data field are optional and may not be present in some messages. Synchronization (e.g., the Preamble) information is added to this PDU by the Physical Layer.

The HART PDU is delimited by the combination of a unique "start of frame" character (**Delimiter**) which identifies the frame's beginning and indicates the position of the Byte Count, and by a **Byte Count** field which determines where the frame ends (see Section 5.1.7). The source and destination of the frame is determined by an **Address** field along with the Frame Type. **Expansion Bytes** are optional and their definition controlled by the HCF. If a slave device cannot interpret the contents of the Expansion Bytes, then the Slave device must not answer the message. The **Command** field identifies the message to the Application Layer and defines the content of the Data field. The **Data** field format is defined in the *Command Summary Specification* and contains the information transferred between the host application and the field device. All portions of the frame, including the delimiter shall be protected by a combination of odd parity on each byte transmitted and the trailing **Check Byte** field. When all bytes in the frame starting with the Delimiter and including the Check Byte are exclusive or'd (i.e., XOR) together, the result must be exactly zero (0x00).

| Delimiter | Address | [Expansion Bytes] | Command | Byte Count | [Data] | Check Byte |
|---|---|---|---|---|---|---|

**Figure 3. HART Frame Format**

All fields in a frame are an integral number of bytes in length and all bytes of a frame must be transmitted in a single contiguous stream (i.e., there should be no more than a single, idle bit time gap between consecutive characters).

### 5.1.1. Delimiter Field

The delimiter is the first field in a HART message. It is used for message framing by indicating the position of the Byte Count. It also indicates the Frame Type which is used for bus arbitration. The delimiter consists of four sub-fields (see Figure 4):

- The most significant bit, bit 7, indicates whether a polling (1 byte) or unique (5 byte) address is included in this frame.
- Bits 6 and 5 indicate the number of Expansion Bytes that are in this frame. Normally this field is set to zero.
- Bits 4 and 3 indicate the Physical Layer type (see Common Table 24). For the (asynchronous) FSK Physical Layer, this field is set to 0.
- The least significant 3 bits (bits 2-0) indicate the frame type. These three bits plus the master address bit (see Section 5.2) implies the passing of a token and together they are used for Media Access Control (MAC).

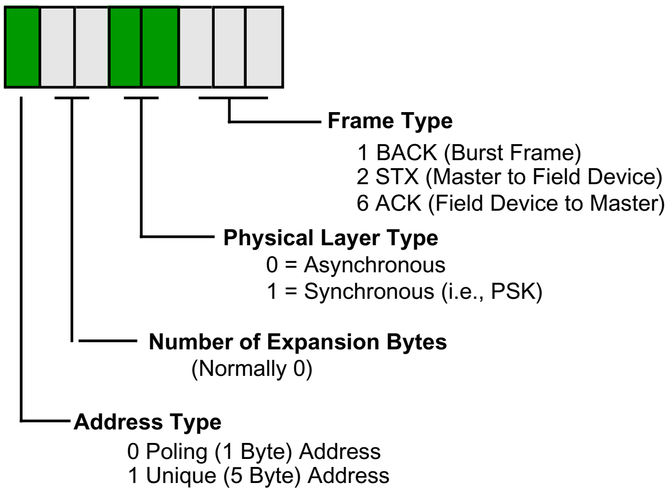
Frame Type
1 BACK (Burst Frame)
2 STX (Master to Field Device)
6 ACK (Field Device to Master)
Physical Layer Type
0 = Asynchronous
1 = Synchronous (i.e., PSK)
Number of Expansion Bytes
(Normally 0)
Address Type
0 Poling (1 Byte) Address
1 Unique (5 Byte) Address

**Figure 4. The Delimiter Field**

### Frame Types

Three frame types are supported by the HART Data Link Layer:

- An **STX** (0x02) indicates a Master to Field Device (i.e., a Slave or Burst-Mode Device) frame. An STX is the start of a transaction and is normally followed by an ACK.
- An **ACK** (0x06) is the Slave's or Burst Mode Device's response to an STX.
- An **BACK** (0x01) is a Burst Acknowledge frame periodically transmitted by a Burst-Mode Device. These frames are transmitted without a corresponding STX.

### 5.1.2. Address Field

The Protocol supports both five (5) byte "unique" addresses and one (1) byte polling addresses. The length of the Address field (1 or 5 bytes) is indicated by the Delimiter (see Section 5.1.1). The Address field always includes both the master address, the slave address, and (for ACK and BACK frames) whether the slave is in burst mode.

### 5.1.3. Expansion Bytes

This field is 0-3 bytes long and its length is indicated in the Delimiter. The definition of the Expansion Bytes are controlled by the HCF. If a Field Device does not know the meaning of all Expansion Bytes contained in the frame, then the Field Device must not answer.

### 5.1.4. Command

This field is one byte long. Command numbers are classified and allocated in the *Command Summary Specification*. Command byte values are echoed back unchanged in responses from Field Devices.

Note: Command number 254 is reserved and shall not be used in any implementation.

### 5.1.5. Byte Count

This field is one byte in length and represents the number of bytes of Application Layer data between the Byte Count and the Check Byte (both excluded from the count). All values between 0 and 255 (both inclusive) are legal in this field.

### 5.1.6. Data

The Data field is optional and consists of an integral number of bytes of Application Layer data. The Data field contains sub-fields as defined in the *Command Summary Specification* and contains the information transferred between the host application and the Field Device.

All BACK and ACK messages must contain at least two data bytes. If the most significant bit (i.e., bit 7) of first data byte is set then the byte contains Communication Error information (see the *Command Summary Specification*). Master Devices should send at least three retries (i.e., for a total of four STXs) when a Communication Error is encountered before indicating an error to the Application Layer.

No Communication Error shall be indicated in BACK (Burst Frames) frames as there is no corresponding STX about which to report a communication error.

Aside from interpreting the communication error status byte (if it is present), Data Link Layer implementations shall not make any interpretation of the Data field. This means that frame recognition is disabled from the beginning of the Data field until the frame is complete as determined by a (correct) byte-count or by the physical layer signaling the end of a message (e.g., through absence of carrier detect).

### 5.1.7. Check Byte

This field is one byte long. The Check Byte value is determined by a bitwise exclusive OR of all bytes of a message including the leading delimiter. The Check Byte is calculated and checked as described in Section 5.4.

## 5.2. Framing

Frame synchronization or "framing" is the process of identifying and receiving a message. For the HART Protocol, framing begins at the end of the preamble with the reception of the Delimiter. Four key events occur that allow the successful framing of a HART message:

- **Assertion of Carrier**. Once carrier is detected (see ENABLE.indicate in Section 6.1), the channel is considered active, and non-transmitting devices must stay silent until the end of the message (i.e., until the reception of the Check Byte). Communication theory shows that when a valid carrier is present it is impossible for another device to successfully transmit a message.

- **Reception of the Delimiter**. This allows the capture of the message header (Address, Expansion Bytes, Command and Byte Count). The Delimiter indicates the organization of the message header (see Figure 5):

  - Unique (5 Byte) or polling (1 Byte) address, and

  - The presence of any frame Expansion (0-3) Bytes.

  The Delimiter also allows the size of the frame header to be calculated.

- **Reception of the Byte Count**. The byte count is contained in the frame header and allows the data portion of the message to be framed properly.

- **Reception of the Check Byte**. This indicates the end of a message and is a key element in the error checking scheme.

Note: Bus arbitration rules allow a HART compatible device to begin transmitting its frame as soon as it receives the Check Byte from the device relinquishing control of the network. As a result, it is possible that the carrier detect signal may not be de-asserted between HART message frames.

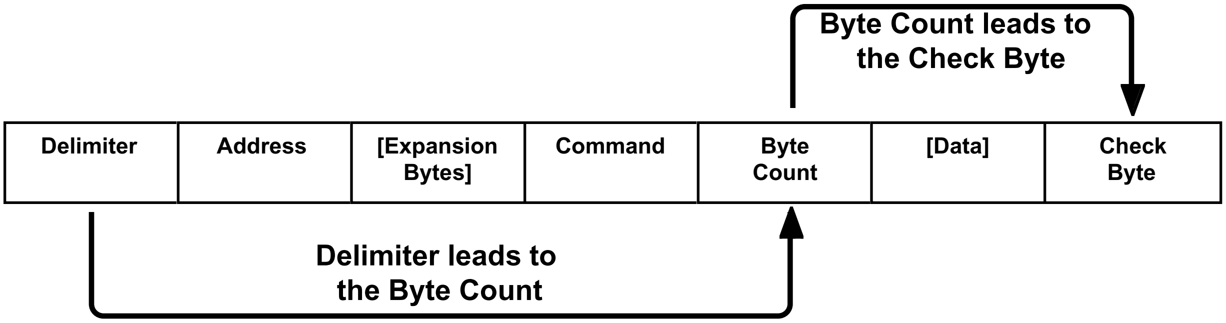
Byte Count leads to the Check Byte
Delimiter
Address
[Expansion Bytes]
Command
Byte Count
[Data]
Check Byte
Delimiter leads to the Byte Count

**Figure 5. HART Framing**

All HART devices must frame all messages. All HART timing and MAC requirements begin with the reception of the Check Byte.

While several errors are possible during message reception, only three are fatal: (1) a gap error; (2)˚loss of carrier; or (3) a communication error receiving the Delimiter, Address or Byte Count. A gap error indicates that the transmitting device malfunctioned mid-message. Loss of carrier indicates that there is insufficient signal strength to allow reliable communications. A communication error while receiving the Address means that the slave does not know if the message was for him and, as a result, the slave should not answer. An error in the Delimiter or Byte Count prevents successful framing of the message. All of these error conditions are fatal, aborts the message reception and discards any portion of the message received (see Section 5.4).

## 5.3. Device Addresses

The Protocol supports both five (5) byte "unique" addresses and one (1) byte polling addresses. The Delimiter indicates whether a unique or polling address is in the frame (see Section 5.1.1).

### 5.3.1. Master Addresses and the Burst Mode Flag

The Protocol includes the source and destination addresses in each frame. For both unique and polling addresses, the most significant bit of the address field indicates the master associated with this frame. A primary master uses the value "1" for this bit, a secondary master uses the value "0". Slave devices must echo back this field unchanged. The next bit indicates whether the slave device is in Burst-Mode. Slave devices must set this bit to a "1" to indicate that the device is in Burst-Mode or to a "0" if the device is not in Burst-Mode. Master devices must set this bit to "0" in all requests to slaves.

The master address combined with the Frame Type imply a token that indicates to the MAC the next device that should begin a message transmission (see Section 7).

### 5.3.2. Unique Addresses

Except for Command 0, all HART frames consist of a 5 byte address based on the slave device's Unique Identifier. A Unique Identifier is associated with each field instrument manufactured and consist of the Manufacturer Identifier, the Device Type and a unique Device Identifier. The Protocol normally uses the lower 38 bits of the Unique Identifier to address Data Link Layer requests to field devices on the link (see Figure 6). As a result, the unique address must consist of:

- The master address and the burst mode bit as described in Section 5.3.1.
- The least significant 6 bits of the Manufacturer ID. The valid Manufacturer ID can be found in Common Tables 8: Company (Manufacturer) Identification Codes.

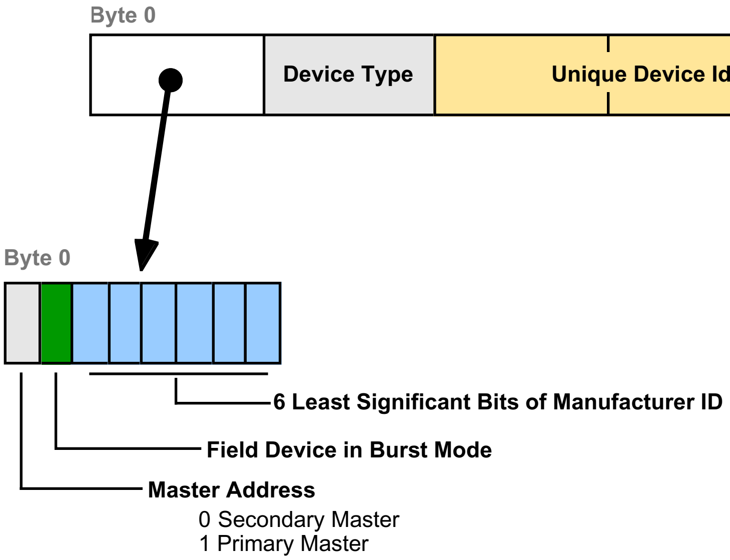

**Figure 6. Long Frame Address**

- A 1 byte Device Type Code. This code is allocated and controlled by the manufacturer. Since only the lower 6 bits of the Manufacturer ID is used in the address, the Device Type code must be allocated as indicated in Table 1 to minimize the chance of duplicate unique addresses on a single Physical Link. Further specifications regarding allocating Device Type Codes can be found in the *Command Summary Specification*.
- A 3 byte Device Identifier. This is similar to a serial number in that each device manufactured with the same Device Type Code must have a different Device Identifier.

**Table 1. Device Type Code Allocation**

| Manufacturer ID | Device Type Code Allocation |
|---|---|
| 0-63 (0x00-0x3F) | Begin allocating at 0 |
| 64-127 (0x40-0x7F) | Begin at 239 (0xEF) and assign succeedingly smaller valued codes |
| 128-191 (0x80-0BF) | Begin at 127 (0x7F) and assign succeedingly smaller valued codes |
| 192-255 (0xC0-0xFF) | Begin at 128 (0x80) and assign succeedingly larger valued codes |

### 5.3.3. Broadcast Address

In addition to unique addresses based on the device's Unique Identifier, the Protocol supports a Broadcast Address. A Broadcast Address is a 5 byte address with 38 bits of zeros in place of the Unique Identifier. Devices shall treat a frame with this address as though the frame was addressed directly to them. Frames with Broadcast Addresses shall only be used for services or commands that use other parameters in the broadcast message to ensure that zero or exactly one device generates a field device response message to the broadcast (e.g., see Command 11).

### 5.3.4. Polling Addresses

Polling addresses allow the Protocol to dynamically associate a short frame address with each Field Device on the link. The polling addresses may be used during a Master Device's network initialization to rapidly scan and automatically identify the field devices (see the *Command Summary Specification*). The Protocol provides the capability necessary to manipulate these addresses on initialization as well as during normal operations. It also provides the capability to obtain the Unique Identifier associated with a particular short frame address when the Master Device connects to the network or when necessary.

> Note: Previous revisions of the Protocol utilized only polling addresses. In order to maintain backwards compatibility, only Universal Command 0 is allowed to use polling addressing. Field devices must ignore all frames using a polling address except Command 0.

Figure 7 shows the required format of the one byte short frame address consisting of:

- The master address and the burst mode bit as described in Section 5.3.1.
- The least significant 6 bits specify the polling address. Polling addresses 16-63 should not be assigned when multi-dropping with earlier Protocol revision (i.e., HART revision 3 through 5) field devices.

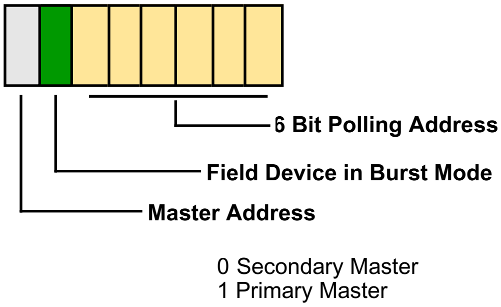

**Figure 7. Short Frame Address**

## 5.4. Error Detection Coding

To perform error detection, HART utilizes a single parity check product coding scheme [Leung]. This allows HART to use parity checking in two dimensions: across the bits in a single byte; and across the bit positions in the transmitted message. The two dimensions of parity checking can be seen in Figure 8. Each byte consists of 8 bits plus odd parity ("Vertical Parity"). In other words 9 bits are transmitted for each byte. Figure 8 shows the organization of the individual bytes into a HART long address frame.

The second dimension of error detection is generated by "Exclusive OR"-ing all bytes from the Delimiter up to and including the last Data byte. The result is transmitted as the last byte (i.e., the Check Byte) of a message. In other words, if all bytes in a frame are exclusive or'd together starting with the Delimiter and including the Check byte, the result must be 0x00. The Vertical Parity on the Check Byte is odd as well.

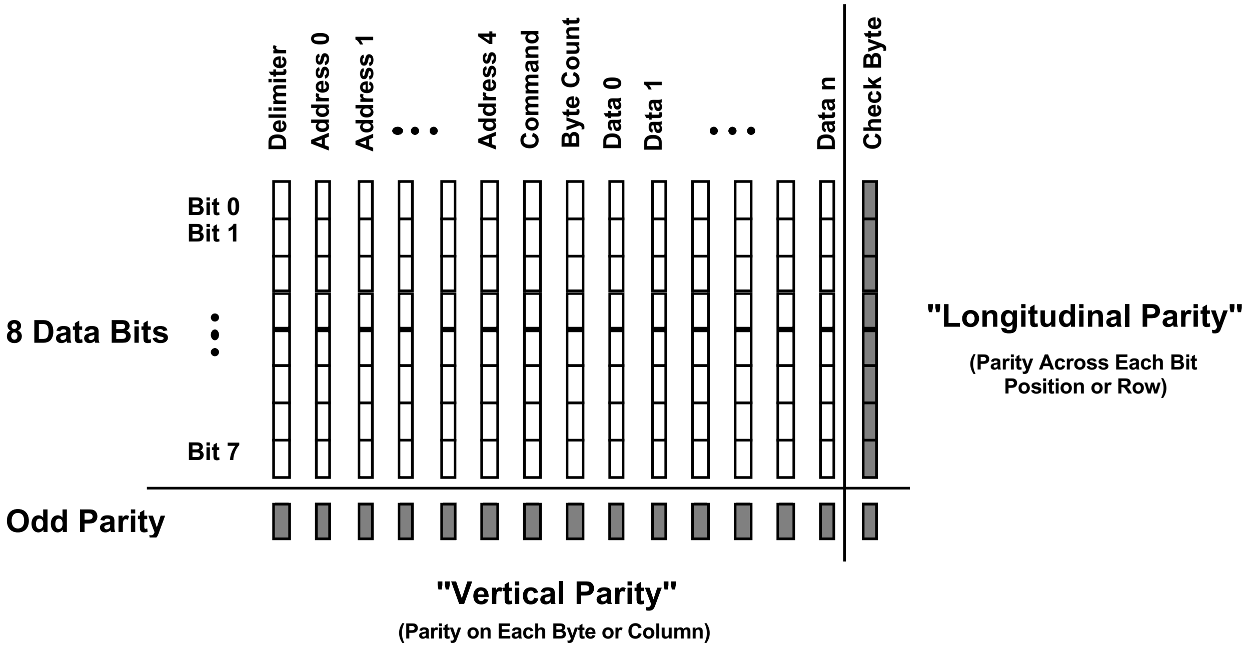

**Figure 8. HART Error Detection Scheme**

### 5.4.1. Slave Response to Communication Errors

The operation of a slave device in the presence of communication errors must be consistent. Communication errors could occur in any portion of the HART frame. As a result, the response of the slave device depends on the field in the frame where the error is detected. Table 2 summarizes the required slave action upon detection of a communication error in particular fields.

### Table 2. Slave Action by Frame Field and Error Detected

| Field | Field Device Response |
|---|---|
| Delimiter | Framing of the message must be aborted. No reply shall be generated. If the slave is in Burst Mode, then the burst wait timer should be set to RT1(Primary). |
| Expansion Bytes | Varies depending on the definition of the Expansion Byte. If the Expansion Byte affects addressing then the Field Device must not answer. If the slave is in Burst Mode then the burst wait timer should be set to RT1(Primary).<br>Note: *If a Field Device does not know the meaning of all Expansion Bytes then the Field Device must not answer.* |
| Address | The message should be treated as if addressed to another device. No reply shall be generated.<br>Note: *If a Broadcast Address is used then this action applies to an error in the additional identification field (e.g., the Tag in Universal Command 11).* |
| Command | The device must reply. The response frame shall consist of the command that the slave detected, and the communication error status byte shall indicate the appropriate error. No data shall be returned. |
| Byte Count | Framing of the message must be aborted. No reply shall be generated. If the slave is in Burst Mode, then the burst wait timer should be set to RT1(Primary). |
| Data | The response shall indicate the appropriate error in the communication error status byte. No data shall be returned. |
| Check Byte | The response shall indicate the appropriate error in the communication error status byte. No data shall be returned. |

# 6. DATA LINK LAYER SERVICES

Section 5 specified the format of a HART PDU and how error detection is implemented in the Protocol. In this section the operation of the Data Link layer is specified from a "black box" point of view. This section specifies the Service Access Points (SAPs) supplied by the Physical Layer to the Data Link Layer and available from the Data Link Layer to the Application Layer. In addition to specifying the individual SAPs, time sequence diagrams (see [Halsall]) are included to indicate the order in which the SAPs should be used and the order of event occurrence at the protocol layer boundaries. See ANNEX A for more information on the service specification methodology.

The Services described in this section are used to obtain:

- Access to Physical Layer services;
- A reliable, "at least once", transaction service between peer entities. The service is not designed to provide duplicate detection.
- Management services for Data Link Layer configuration.

The SAPs required from the Physical Layer are specified first followed by the Data Link Layer SAPs. All SAPs described here must be supported by the device unless otherwise stated. The mapping of these SAPs into an implementation is entirely a local matter, and is in no way restricted by this specification.

In the definition of the SAPs, parameters are defined. Some parameters are optional and may not be present in all invocations of the SAP. Optional parameters are distinguished by enclosing them within square brackets ("[","]") in the SAP definitions.

## 6.1. Physical Layer SAPs

The HART Physical Layers do not specify the services they provide. However, the Protocol Physical Layers must provide SAPs, and these are specified in this section. The following SAPs specified by the time sequence diagrams in Figure 9 and the following descriptions must be supported by the underlying Physical Layer.

The Physical Layer interface is determined by a set of transmit and receive SAPs. At any given instant the interface may undertake either a transmit or receive SAP sequence, but not both simultaneously (i.e., the interface is half duplex). To make the specification more clear a suggested mapping of the interface SAPs into typical modem control signals is supplied.

**`RESET.request()`**

This SAP is passed to the Physical Layer to request it to undergo a physical reset. Some implementations may optionally return a `RESET.confirm`.

**RESET.confirm()**

This optional SAP is passed from the Physical Layer to indicate that it has completed a physical reset in response to a RESET.request.

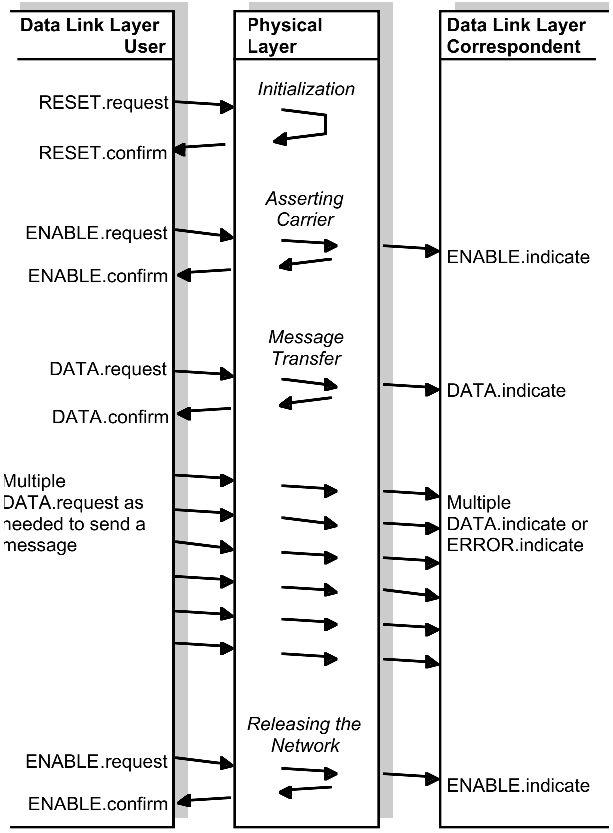

**Figure 9. Physical Layer SAPs**

**ENABLE.request(state)**

This SAP is passed to the Physical Layer to request it to prepare for data transmission or to terminate a data transmission, depending on whether the parameter state is "high" or "low". This signal is usually implemented as the request to send (RTS) signal in a modem. An ENABLE.request may only be issued if the interface is in an idle state (i.e., no pending TRANSMIT.indicate or TRANSMIT.confirm).

**ENABLE.indicate(state)**

A receive SAP passed by the Physical Layer to indicate the establishment of communication with a peer Physical Layer entity, or to indicate the termination of a currently established connection, depending on whether state is "high" or "low". This SAP is usually implemented as a **Carrier Detect** (CD) signal from a modem. Typically a CD turn-on and turn-off delay together with hysteresis on the receive carrier levels that trigger this transition is implemented to prevent oscillation and provide protection against noise bursts. See the appropriate Physical Layer specification for details.

**`ENABLE.confirm(state)`**

This SAP is passed back by the Physical Layer to confirm its ability to accept data for transmission, or to indicate that it will accept no further data. In a modem implementation this SAP is usually mapped into the clear to send (CTS) signal from a modem. The Physical Layer may implement RTS/CTS turn-on/off delays in order to give its corresponding physical layer entity (the other modem) time to establish communication.

**`DATA.request(data)`**

This is the SAP used to actually transfer a unit of data (i.e., a byte) to the Physical Layer for transmission. This would typically be implemented as a write to a transmit data register or some other such operation.

**`DATA.confirm(data)`**

This SAP is passed back from the Physical Layer to acceptance of the data from a previous `DATA.request`, and its readiness to process more of such requests. Implementation of this SAP may be setting a status bit in a register for a simple polled implementation; or by initiating an interrupt on the transmit data register becoming empty in a more sophisticated version.

**`DATA.indicate(data)`**

This SAP is passed back from the Physical Layer to indicate the availability of data unit (i.e., a byte). In implementation, this SAP may be mapped into a "receive data register full" status bit or a "receive data" interrupt.

Note: Since there is no flow control between the transmitting and receiving devices, all devices must respond to incoming data at least as fast as the underlying Physical Layer can transfer data.

**`ERROR.indicate(status, data)`**

Returned by the physical interface (e.g., the UART) to indicate an error in received data or its handling. Examples of errors that may be returned by the physical interface are parity errors, framing errors and receive data overruns.

## 6.2. Data Link Layer SAPs

The Protocol supports two sets of Data Link Layer SAPs.

- **Message** SAPs specify the basic capability to send a message and receive a reply. In addition, SAPs are specified, allowing the capture of unsolicited message traffic (e.g., to capture Burst-Mode frames)

- **Management** SAPs specify services that allow the Data Link Layer itself to be configured. SAPs are specified for both Master and Field Devices.

These SAPs represent the minimum set of services that must be provided by a Data Link Layer implementation. This section includes time sequence diagrams to define overall operation, and each individual SAP is specified.

### 6.2.1. Message SAPs

Message SAPs provide services supporting the basic transfer of data between a Master Device and a Field Device. The time sequence diagram for these SAPs is shown in Figure 10. There are three sets of SAPs:

- **Transmit**. These SAPs transfer frames between the Master Device and the Field Device. The first sequence in Figure 10 shows simple, error-free transaction. This transaction is always initiated by the master. The Protocol also supports automatic retries to ensure accurate data exchange (see Section 7.2.4). A transaction, including retries is shown in the second sequence of Figure 10.

- **Receive**. The third and fourth sequences show the optional, receive only SAPs. For a master this is a promiscuous mode where communications from the other master is captured. This can be a useful optimization if data the Application Layer desires is accessed by the other master. The `RECEIVE.indicate` SAP is used by Field Devices that support Commands 113 and 114 (see the *Common Practice Command Specification*).

- **Cyclic**. The final sequence shows the reception of Burst-Mode frames. Burst-Mode devices are managed using the Transmit services to send the appropriate commands (see the *Common Practice Command Specification*). While Burst-Mode is optional for Field Devices, masters must support Burst-Mode operation as part of Media Access Control (see Section 7.2.4). Configuration of the Field Device Data Link Layer for Burst-Mode operation is specified in Section 6.2.2.

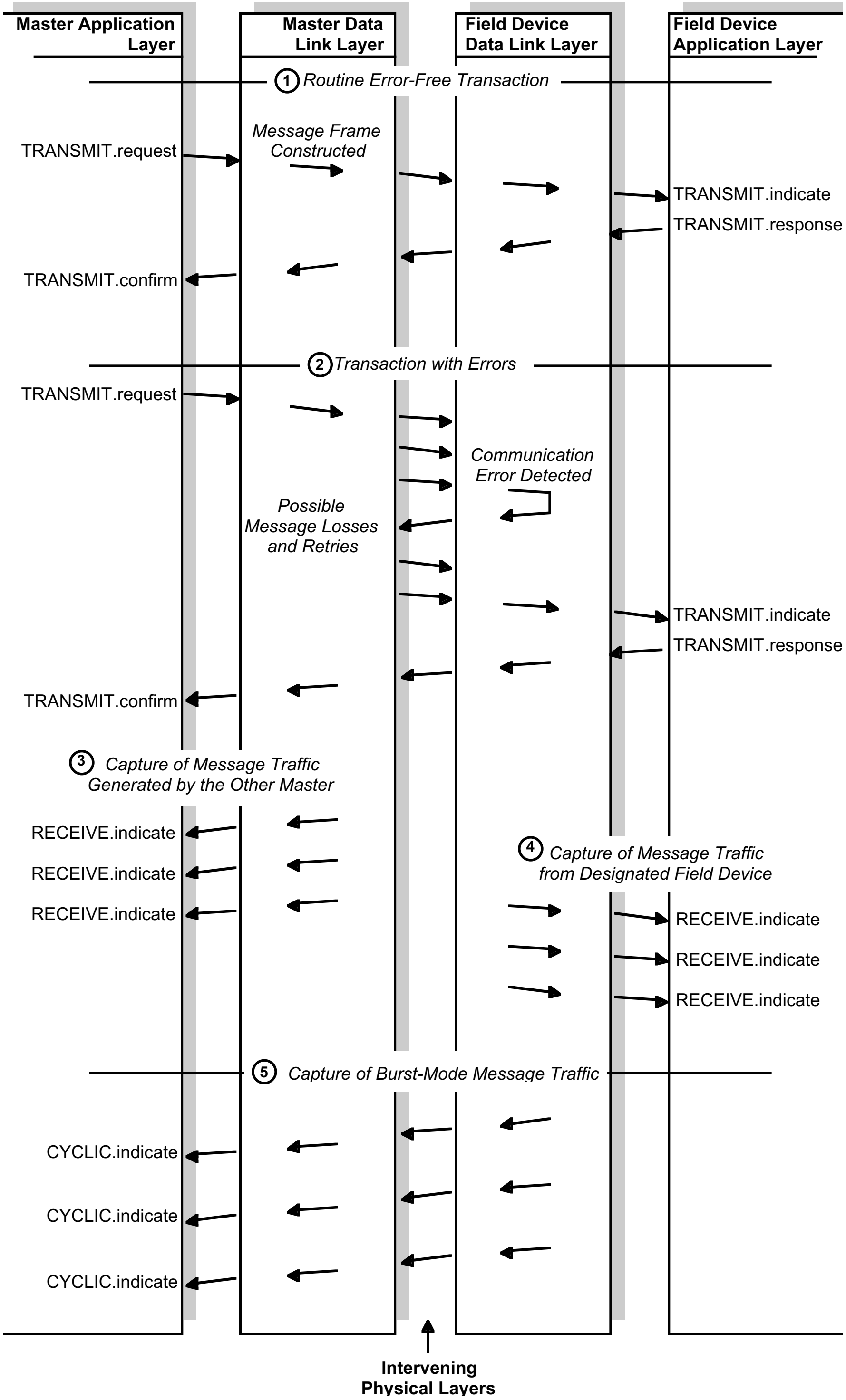
Master Application Layer
Master Data Link Layer
Field Device Data Link Layer
Field Device Application Layer
1 Routine Error-Free Transaction
Message Frame Constructed
TRANSMIT.request
TRANSMIT.indicate
TRANSMIT.response
TRANSMIT.confirm
2 Transaction with Errors
TRANSMIT.request
Communication Error Detected
Possible Message Losses and Retries
TRANSMIT.indicate
TRANSMIT.response
TRANSMIT.confirm
3 Capture of Message Traffic Generated by the Other Master
RECEIVE.indicate
RECEIVE.indicate
RECEIVE.indicate
4 Capture of Message Traffic from Designated Field Device
RECEIVE.indicate
RECEIVE.indicate
RECEIVE.indicate
5 Capture of Burst-Mode Message Traffic
CYCLIC.indicate
CYCLIC.indicate
CYCLIC.indicate
Intervening Physical Layers

**Figure 10. Transmit Service Sequence**

**TRANSMIT.request(preamble_length, address, cmd, [data])**
This SAP is used by a Master Device to request the transfer of information to a Field Device at the given address. This SAP shall be encoded as a request message (i.e., STX) addressed to a Field Device.

The `preamble_length` parameter indicates to the master Data Link Layer how long the preamble should be and can vary by Field Device. This information is returned by Field Devices in Identity Commands (see the Command Summary Specification). The `address` is the Unique or the Polling Address of the destination Field Device. The `address` may also be a Broadcast Address.

The `cmd` and the (optional) `data` parameters carry the Application Layer information being transferred.

> Note: All information must be supplied by the Master Application Layer and it is the Master Application Layer's responsibility to determine validity of the parameters. The Protocol implementation shall carry out a limited validation of the parameters (e.g., address out of range) to the extent of its capability.

**TRANSMIT.indicate(cmd, [data])**
This SAP is invoked by the Data Link Layer to indicate to a Field Device Application Layer that a valid message has been received from a Master Device. The Field Device Application Layer receives the contents of the command byte and optional data if any.

Encoding of this SAP is a local matter and is not limited by the specification.

**TRANSMIT.response(cmd, data)**
This SAP is executed by the Field Device to respond to an incoming `TRANSMIT.indicate`. It is the mechanism used to return the results of executing on information provided in the immediately previous indication.

> Note: The `data` parameter is mandatory in all Field Device responses (i.e., ACK or BACK). All Field Device Responses contain at least 2 data bytes containing communication error or command completion status and Device Status (see *Command Summary Specification*).

This SAP shall be encoded as a response message addressed to a Master Device.

**TRANSMIT.confirm(local_status, address, cmd, data)**
This SAP is returned to the Master Device Application Layer to communicate the results of a previously executed `TRANSMIT.request`. The slave response (if any) is returned. The `local_status` parameter carries the status of the communication routines in the master.

Encoding of this SAP is a local matter and is not limited by the specification.

**`CYCLIC.indicate (local_status, address, cmd, data)`**
This SAP indicates that a cyclic broadcast of data (i.e., a BACK) from a Burst-Mode Device has been received at this Master Device. BACKs addressed to either Master Device are returned by this SAP. The local status byte carries the status of the communication routines in the Master Device.

Encoding of this SAP is a local matter and is not limited by the specification.

**`RECEIVE.indicate (local_status, address, cmd, [data])`**
This optional SAP indicates that a frame, not addressed to this device, has been received. The local status byte carries the status of the communication routines in this device.

For a Master Device, the reception of communications by the other master can be a useful optimization. Sometimes this is called a "promiscuous operating mode" and can be used for network trouble shooting. The `data` parameter may not be present if the frame is an STX.

For a Field Device, this SAP can capture specific frames from another, multi-dropped Field Device (see Table 4 and Commands 113 and 114 in the *Common Practice Command Specification*). Since the Field Device may only receive an ACK or BACK the `data` parameter always contains at least two bytes.

### 6.2.2. Management SAPs

Management SAPs support both the Master and Field Device Data Link Layer. The fundamental SAP is a `LOCAL_MANAGEMENT` sequence (see Figure 11). This set of SAPs must be supported in both the Master Device and Field Device. This SAP offers several different services as specified in Table 3 and Table 4.

> Note: None of the SAPs in this section require any data to be transmitted over the communication link. Remote management of the Field Device's Data Link Layer configuration is possible using standard HART commands and the TRANSMIT SAPs in Section 6.2.1. Network and Field Device management by Master Devices are not included in this specification. These topics are addressed in the Application Layer Specifications (e.g., in the *Command Summary Specification*).

The Master Device management SAPs are specified in Table 3. These SAPs configure the master Data Link Layer properties including the master address and the number of retries. All SAPs in Table 3 must be supported.

Both mandatory and optional Field Device management services are specified in Table 4. These SAPs allow the Field Device Data Link Layer to be configured on power up by the Field Device Application Layer. In addition, this allows management of the Field Device's non-volatile and programmable non-volatile memory to be isolated from the Data Link Layer implementation. In addition, some of the SAPs may be accessed long after the Field Device has been on-line. For example, the Application Layer may receive a command from a Master Device that changes the polling address.

The final sequence in Figure 11 provides support for Burst-Mode operation. These CYCLIC_UPDATE SAPs allow the Burst-mode command and data to be updated by the Field Device while the Data Link Layer remains responsible for transmitting the Burst-Mode frame itself. CYCLIC_UPDATE may be accessed either as the result of a process update or a configuration change caused by the Application Layer's reception of the Master Device command.

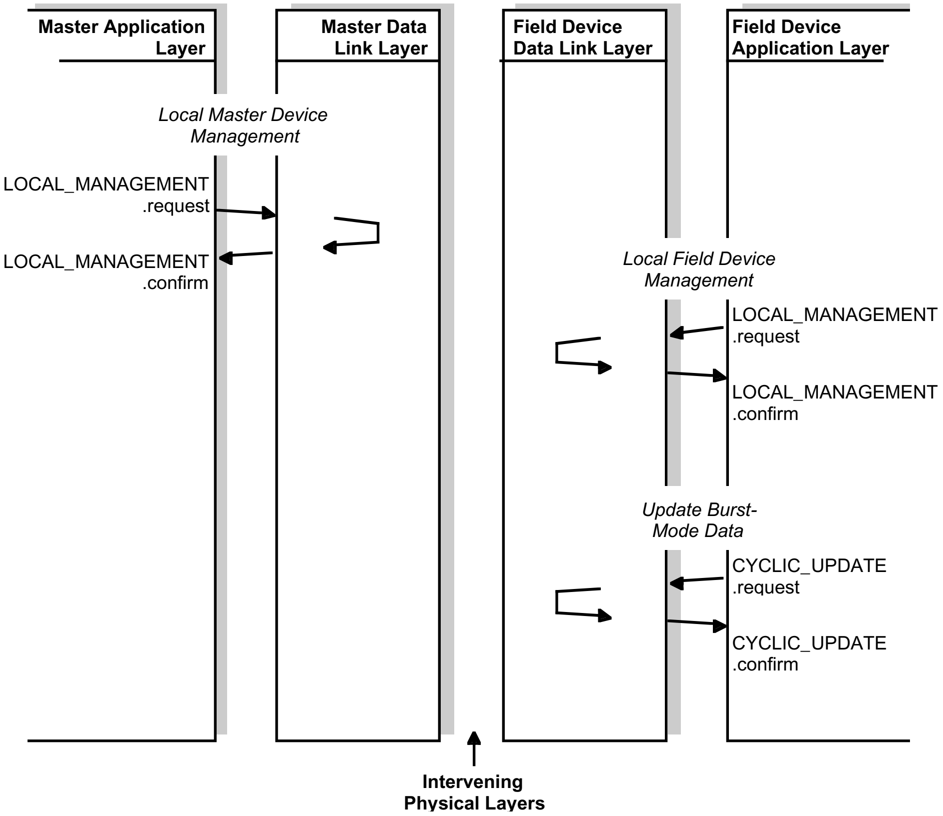
Master Application Layer
Master Data Link Layer
Field Device Data Link Layer
Field Device Application Layer
Local Master Device Management
LOCAL_MANAGEMENT .request
LOCAL_MANAGEMENT .confirm
Local Field Device Management
LOCAL_MANAGEMENT .request
LOCAL_MANAGEMENT .confirm
Update Burst-Mode Data
CYCLIC_UPDATE .request
CYCLIC_UPDATE .confirm
Intervening Physical Layers

**Figure 11. Data Link Layer Management Service Sequence**

**`LOCAL_MANAGEMENT.request(service, [data])`**
This SAP is used to configure Data Link Layer properties in Master Devices or Field Devices. Services and properties supported in Master Devices are specified in Table 3. Services and properties that may be supported in Field Devices are specified in Table 4.

**`LOCAL_MANAGEMENT.confirm(service, status, [data])`**
This SAP is used to return the results of a corresponding LOCAL_MANAGEMENT.request. The status shall return the results of executing the request.

**Table 3. Master Device Management Commands**

| Service | Data | Description |
|---|---|---|
| SET_RE-TRY_LIMIT | re-try limit | A master Data Link Layer must automatically perform retries when communication errors occur. A master Data Link Layer should retry at least three times (i.e., the original transmission plus 3 retries) before returning an error to the Application Layer. |
| SET_MASTER_ ADDRESS | master address | This configures the Data Link Layer to operate either as a Primary or Secondary master. |
| CAPTURE_BURST_ FRAMES | | This configures the Data Link Layer to begin generating `CYCLIC.indicate` SAPs when a Burst-Mode (i.e., BACK) frame is received. |
| IGNORE_BURST_ FRAMES | | This disables generation of `CYCLIC.indicate` SAPs when a Burst-Mode (i.e., BACK) frame is received. |
| CAPTURE_OTHER_ MASTER_FRAMES | | This configures the Data Link Layer to begin generating `RECEIVE.indicate` SAPs when a messages containing the other master's address are received. |
| IGNORE_OTHER_ MASTER_FRAMES | | This disables generation of `RECEIVE.indicate` SAPs when messages containing the other master's address are received. |

**`CYCLIC_UPDATE.request (cmd, data)`**

This SAP is only applicable to Burst-Mode capable Field Devices. This SAP allows the Field Device Application Layer to update the data to be included in the Burst-Mode frame. Burst-Mode frames are required to be transmit immediately after an ACK (see Section 7.2.3). As a result, this SAP allows the contents of the frame to be resident in the Data Link Layer and continuously available for immediate transmission.

Actual Field Device implementation of this SAP is a local matter and is not limited by this specification.

**CYCLIC_UPDATE.confirm (cmd, status, data)**
This SAP is used to return the results of a corresponding CYCLIC_UPDATE.request. The status shall return the results of executing the request.

Actual Field Device implementation of this SAP is a local matter and is not limited by this specification.

**Table 4. Field Device Management SAPs**

| Service | Data | Description |
|---|---|---|
| SET_UNIQUE_ ADDRESS | unique address | This service sets the Unique Address to use in identifying frames transmitted to this Field Device. |
| SET_POLLING_ ADDRESS | polling address | This service sets the Polling Address to use in identifying Command 0 frames transmitted to this Field Device. |
| SET_PREAMBLE | response preamble length | This service sets the length of the preamble (e.g., the number of 0xFF's) sent before all frames. |
| SET_CAPTURE_ ADDRESS | handle, unique address, command number | This (optional) SAP supports the capture of ACK and BACK frames transmitted by other Field Devices. If the handle is not defined, the Data Link Layer must assign one. Some Field Devices support a single capture address (see Commands 113 and 114 in the *Command Summary Specification*). |
| DISABLE_ CAPTURE | handle | This service uses the handle from the previous SET_CAPTURE_ADDRESS service to disable Field Device frame capture. |
| ENABLE_BURST_ MODE | master | This (optional) service configures the Data Link Layer to begin transmitting Burst-Mode frames. Burst-Mode command and data are cached using the CYCLIC_UPDATE SAPs. |
| DISABLE_BURST_ MODE | | This disables generation of Burst-Mode frames. |

# 7. MEDIUM ACCESS CONTROL

The HART Media Access Control (MAC) allows the coexistence of a single active (i.e., initiating transactions) primary master and a single active secondary master together with a single active burst-mode device. It is not designed to handle more than one each of these device types being simultaneously active. Slave devices are passive (i.e., they do not initiate transactions) and several may be on a given network. Furthermore, the MAC is designed to allow equal access to the channel by the primary master, secondary master and a burst mode device. Access to the network is controlled by passing an (implied) token[1] and the use of Link Grant and Link Quiet timeouts. The Link Quiet timeouts are of different durations for each master to ensure the primary master has first access to the network. The passing of the token is indicated by the type of the message and the master's address. This means that proper MAC operation depends on identifying:

- Activity on the network;
- The frame type and master address to determine token passing; and
- The end of a message to know when to begin a message transmission.

The Protocol support three frame types: an STX, an ACK and a BACK (see Section 5.1.1). It is these frame types that imply the passing of a token between HART devices:

- STX - The token is passed to the slave addressed in the message.
- ACK - This completes a master-slave transaction and, when no burst mode device is present, passes the token to the other master (i.e., the master not addressed in the ACK).
- BACK - The message is addressed to the primary or secondary master. While a reply to the same, designated command number is always burst, the master address is toggled each burst. The token is passed to the master not addressed by the BACK.

This section defines the operation of the HART MAC sub-layer and specifies Master, Slave, and Burst Mode Device requirements that ensure reliable communications and orderly access to the HART network.

[1] U.S. Patent Number 5,166,678; 5122794, 4,998,990 Warrior et al.

## 7.1. MAC Operation

A HART network operates in one of two modes. In the first mode, communication is controlled by the master devices accessing the network (see Figure 12). In this mode, the primary master issues a request (**STX**) to a slave device who answers that master (**ACK**). While this ACK conveys data from the slave device to the primary master a token, implied ACK itself, is passed to the secondary master. In other words, the slave device's ACK to the primary master tells the secondary master it has control of the network and can issue a request message (**OSTX**). The slave device then answers the secondary master's request (i.e., sends the **OACK**) and the token passes back to the primary master.

> Note: In this document the letter O (e.g., OBACK, OSTX, OACK) is used to indicate that the message is addressing the other HART master.

In this mode the slave device must frame all messages on the network and answer any message addressed to it. In other words, the slave's only responsibility is answering requests sent to it.

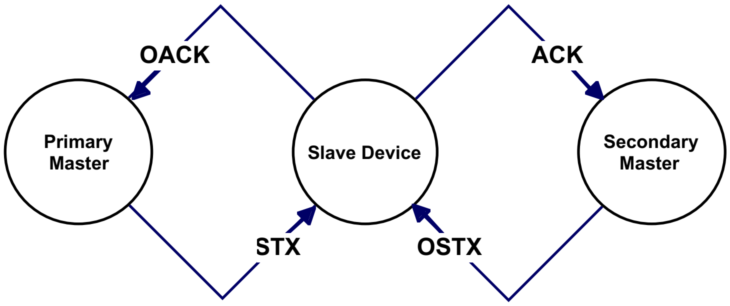

**Figure 12. Token Passing on a HART Network**

Since either a primary or a secondary master may not be present, timing rules are established to ensure network communications remain synchronized.

- When connecting to a network, or when recovering from a network error, a master is considered "unsynchronized". Before becoming "synchronized" and initiating communications, the master must wait the **Link Quiet Time** (**RT1**) or observe other network communications (see Section 7.2.4). The Link Quiet Time for a primary master, RT1(Primary) is shorter then for a secondary master RT1(Secondary), thus allowing the primary master first access to the network.

- After receiving the master's STX, the slave must begin its response transmission (i.e., start the preamble) before the **Slave Time Out** (**STO**) lapses (see Section 7.2.3).

- After the slave's ACK is received, the master waits the **Link Grant Time** (**RT2**) to allow the other master access to the network (see Section 7.2.4). This time starts with the reception of the ACK's Check Byte.

- Once the Link Grant Time expires, the master may begin another STX. However, the master must begin its transmission with the **HOLD** time (see Section 7.2.4).

These five time values (STO, HOLD, RT1(Primary), RT1(Secondary), and RT2) and the Preamble requirements are Physical Layer specific (see Section 8). There are no other Physical Layer impacts to the Data Link Layer.

In the second mode, network communication is controlled by a burst-mode device (see Figure 13). The burst mode device transmits **BACK**s continuously with the master address toggled every BACK. In this mode, the token indicating the master is to transmit an STX is now a BACK (instead of an ACK) to the other master. For example, the BACK to the primary master tells the secondary master to begin its transmission. The burst-mode device sets it Burst Timer (**BT**) to Link Grant Time to allow the master to begin its transmission. When the BT expires without a message being issued by the appropriate master, another BACK is immediately transmitted by the Burst-Mode Device (see Section 7.2.3).

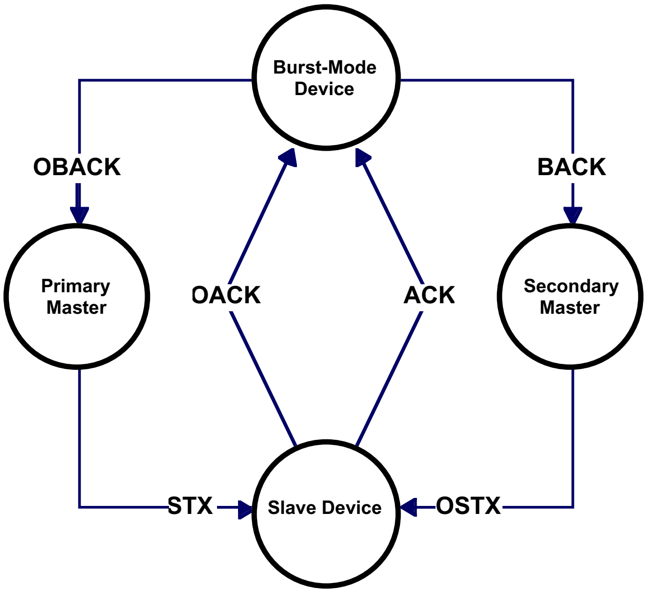

**Figure 13. Token Passing in a Burst Mode Network**

Table 5 summarizes the token passing sequences used by the HART MAC.

The operation of the master is quite different, depending on the operating mode of the network. For correct operation the master must determine: (1) if the network in Burst Mode; and (2) when the network changes operating mode.

The network can be determined to be in Burst Mode by the reception of a BACK, or the reception of an ACK from a Burst-Mode Device. All ACKs from a Burst Mode Device must have the Burst Mode flag set in the address field. There must be no more then one Burst Mode Device on a HART network.

All masters must monitor the Burst Mode flag in all ACKs whether the ACK is addressed to that master or not. When an ACK with the Burst Mode flag set is received, the state variable **BURST** must be set and the slave address noted. The network is Burst Mode and the master operates accordingly. Each ACK from the slave address is monitored and when the Burst Mode flag is reset, BURST is reset. The network is no longer in Burst Mode when this happens.

**Table 5. Token Passing**

| Mode | Message | Type | Token |
|---|---|---|---|
| No Bursting Slave | Primary Master to Slave | STX | Passes to the Slave |
| | Slave to Primary Master | ACK | Passes to Secondary Master |
| | Secondary Master to Slave | STX | Passes to the Slave |
| | Slave reply to Secondary Master | ACK | Passes to Primary Master |
| Bursting Slave | Primary Master to Slave | STX | Passes to the Slave |
| | Slave to Primary Master | ACK | Passes to Bursting Slave |
| | Burst to Primary Master | BACK | Passes to Secondary Master |
| | Secondary Master to Slave | STX | Passes to the Slave |
| | Slave reply to Secondary Master | ACK | Passes to Bursting Slave |
| | Burst to Secondary Master | BACK | Passes to Primary Master |

## 7.2. Device Requirements

This section specifies Master, Slave, and Burst Mode Device requirements to ensure proper MAC operation. Device requirements are specified in this section and state transition diagram [Hatley] are used to clarify these requirements. The state transition diagram notation is summarized in ANNEX B.

To clarify device requirements their specification is decomposed into 4 components (see Figure 14). The XMT_MSG (Section 7.2.2) and RCV_MSG (Section 7.2.1) primitives are responsible for the transmission and reception of a single message. XMT_MSG and RCV_MSG are common to Master, Slave and Burst-Mode Devices.

The Slave / Burst-Mode MAC component (see Section 7.2.3) build on the XMT_MSG and RCV_MSG requirements to specify the bus arbitration rules for field devices. Likewise, the Master MAC (see Section 7.2.4) provides the bus arbitration rules that govern Master Devices.

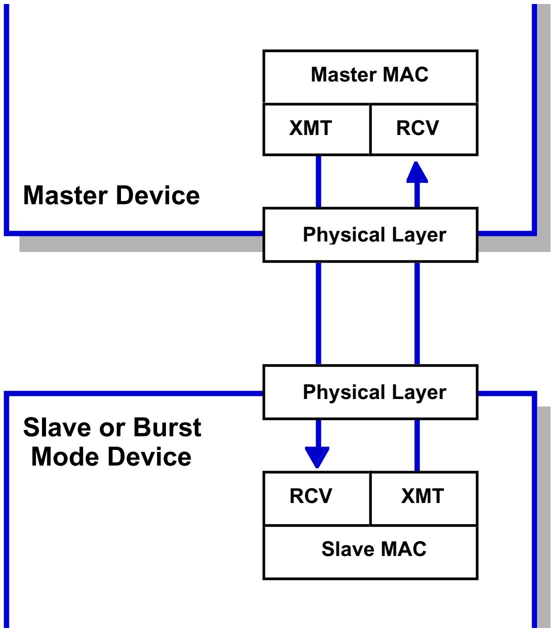

**Figure 14. HART MAC Components**

### 7.2.1. Primitive Receive Machine

The RCV_MSG state transition diagram describes a simple "I/O driver". It first watches for a valid start of a message (SOM). If the inter-character gap is too large, a parity error occurs, or a character other than a delimiter or preamble is received, RCV_MSG status is set to an error. If a valid set of preambles and delimiter are reached, RCV_MSG sets the type of message received (request, acknowledgment or burst message). It then processes the rest of the message until the end of the message is determined as shown by the byte count. At this stage, or earlier if there has been a parity error on the byte-count, such that it cannot be used (see Section 5.4.1), RCV_MSG transitions to complete handling any remaining characters in the message. RCV_MSG terminates processing when `ENABLE.indicate` is negated. In other words, when no further characters are left to be processed from the current message.

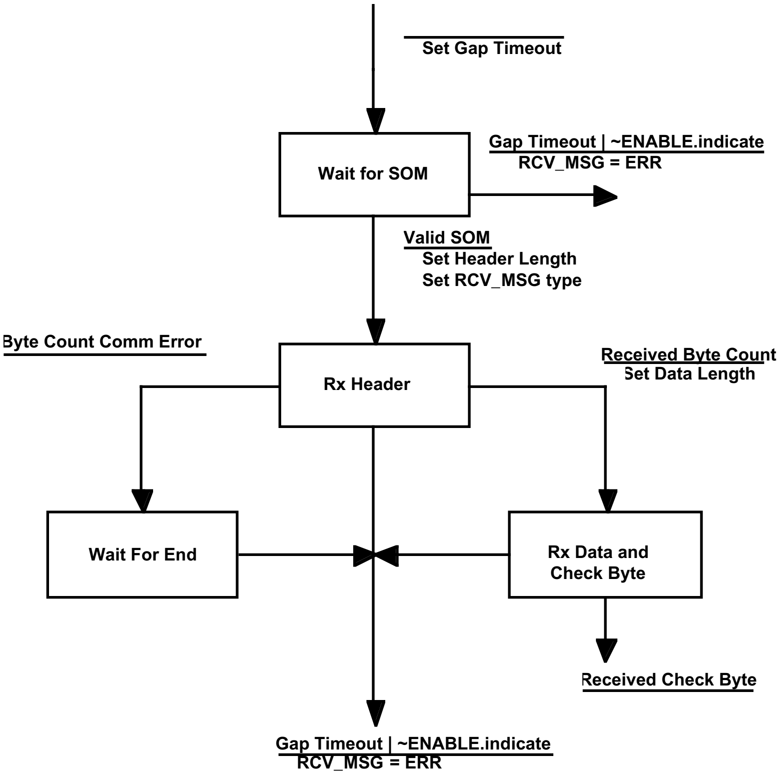

**Figure 15. RCV_MSG Receive State Machine**

Note: In the case where there is a finite amount of time involved in deciding that a response is received, it is possible for a time-out to occur during the RCV_MSG processing. In this and all other such cases, processing of the Master or field device state machine transition currently in progress must be completed before any subsequent input to the state machine (such as the time-out just mentioned) may be recognized.

### 7.2.2. Primitive Transmit Machine

The transmit machine state diagram describes a similar "component". The "INIT" state of the machine implements a start up handshake to give the physical layer time to establish communication with its peer entity at the other end of the link. Once the physical layer issues the confirm, data transfer requests are made in the "WRITE" state. As each data byte is transferred, and a confirm is received from the physical layer, another byte is written until either the end of the message to be transmitted is reached or an error occurs. The negated `ENABLE.request` to the physical layer tells it that no further transmission is required for this message. A status is returned.

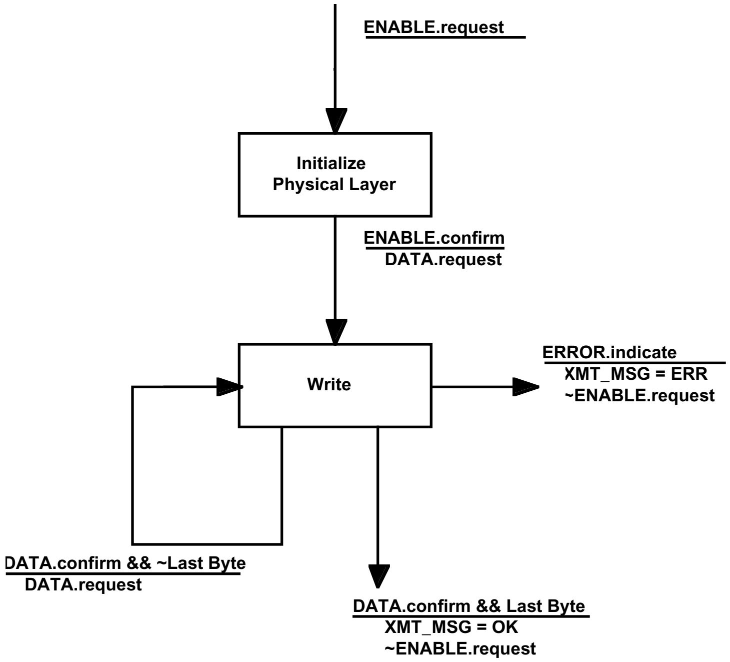

**Figure 16. XMT_MSG Transmit State Machine**

### 7.2.3. Slave and Burst-Mode MAC

Slave and Burst-Mode Device operation is described in Figure 17. Associated with all slave devices is a Slave Time Out (**STO**). It is identical for all field devices. This time out is the maximum amount of time permitted for a field device to respond to an incoming message. If a response cannot be generated within this time period, the transaction is considered to have failed by the master.

Enabling and disabling of burst mode is carried out by the device in response to a request from a master. When burst mode is enabled by a master, the first burst frame should be transmitted contiguously with the slave response. In other words, there should be a gap of less than one bit time between the slave response message to the command enabling burst mode and the first burst message. There must be only one burst mode device active on the communication link at any given time.

> Note: Enabling Burst-Mode is critical because the token passing sequence is fundamentally changed and the Master MAC must adapt its operation to ensure correct bus arbitration. Prompt transmission of the first BACK facilitates the transition of the token passing sequence (see Section 7.1)

When issuing burst messages, the device shall toggle the master address bit value in the burst frame. It does this by setting the value of the bit to the value of its internal state variable "HOST", which it updates with each burst transmission.

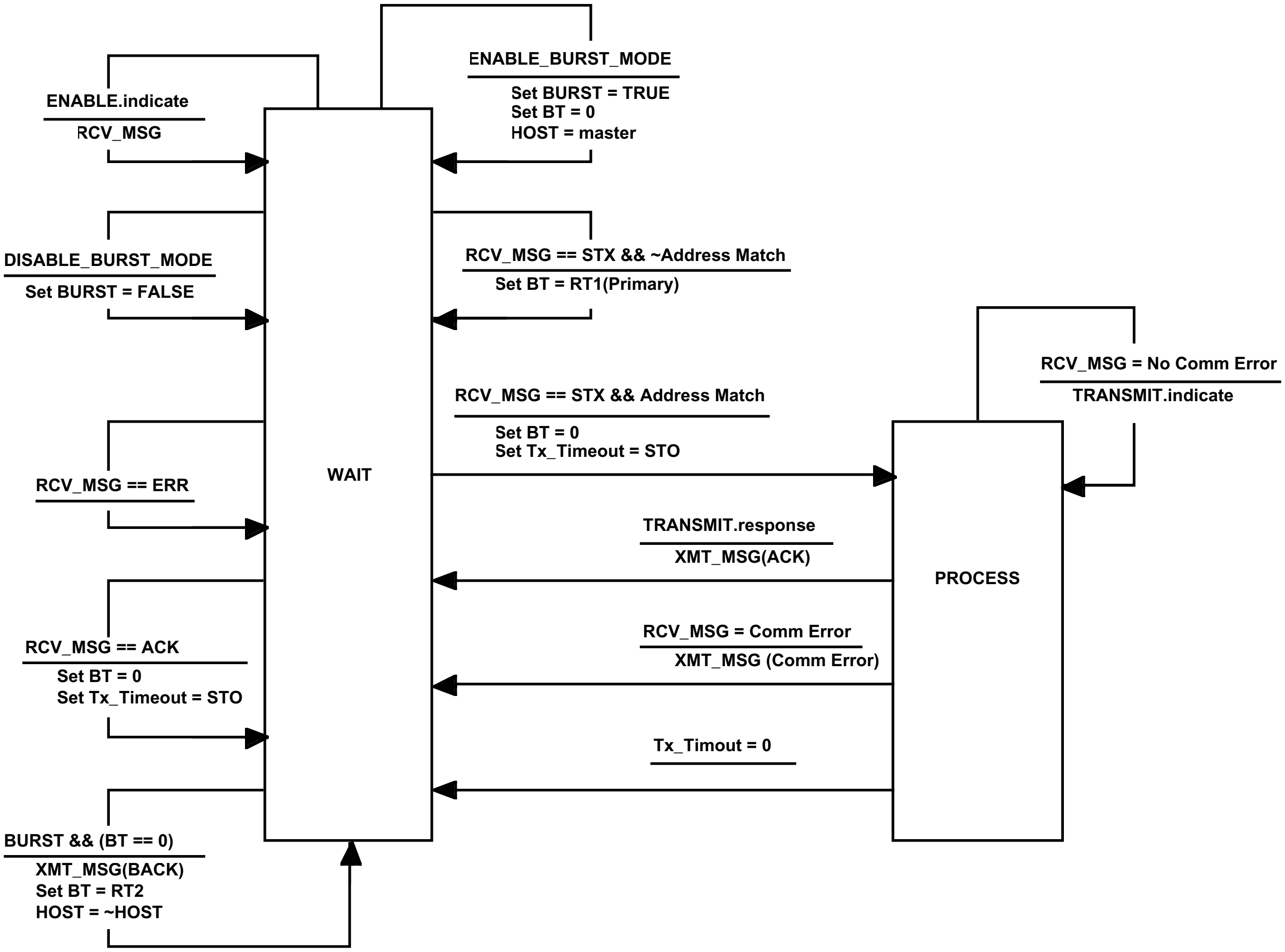

**Figure 17. Slave/ Burst-Mode Device State Machine**

An incoming message (`ENABLE.indicate` is set) triggers the RCV_MSG protocol state machine which begins operation by looking for a correctly framed message. Incorrectly framed messages are ignored. Upon return from RCV_MSG processing, the incoming message type is checked. If this is an acknowledgment, the device can burst right away. If this is a request, the address field is checked against the slave address. If there is no match, the message is ignored, and the slave device, if in burst mode, prepares to defer any burst by setting the BT to RT1(Primary). (This will allow the master to timeout). If no acknowledgment is sent by the other slave, the burst mode device should start bursting immediately after RT1 expiring. The device shall not delay its attempt to burst beyond HOLD after the end of RT1(Primary). Otherwise a validity check is made on the data portion of the message. If any of the parity check tests fail, or if the message buffer overflows, then a response with communications error information is returned, since the message was framed correctly.

If the message has been received without errors, the protocol passes the information in the message back to the user through a `TRANSMIT.indicate`. When the user executes a `TRANSMIT.response`, a response message is formulated and returned. If the device is in burst mode, a burst will follow.

A burst mode device sets "BT" whenever it has issued a burst. A subsequent burst can be issued only after the timer has expired. This permits a master an opportunity to address the device during this interval. A burst mode device shall issue bursts as soon after BT expiring as permitted. The device shall not delay its attempt to burst beyond STO.

### 7.2.4. Master MAC

Both primary and secondary masters run the identical state transition machine. The arbitration of access to the communication link is controlled by time-outs. Time-outs are measured as the interval between transitions of the protocol state machine. These define lengths of time that devices (masters and field devices) executing the communication protocol use to decide on what they do next based on whether or not they hear transmissions on the communication link.

Differentiation between the two masters is on the basis of the value of a single time-out used in two of the transitions within the protocol. This results in equal prioritization of the two masters that access the link except when there has been a failure of arbitration (due to concurrent attempts to access the link).

The master state machine has three states, "WATCHING", during which masters are monitoring the communication link for their turn to use it, "ENABLED", during which the masters can transmit on the communication link if they have a message to send, and "USING", during which they are waiting for a reply to a message they have sent. A master implementation maintains the MSG_PENDING, RCV_MSG and RT1, RT2 timer status variables as state variables for its operation.

Link arbitration is controlled by two defined intervals, the link grant time "RT2", and the link quiet/ slave time-out "RT1".

The smaller arbitration constant is "RT2". It is a constant for both masters. It is set to a sufficiently large value to allow the other master on the communication link adequate time to initiate transmission (if it desires to do so) after the master which is executing the "RT2" time-out has just completed a transaction. The link grant time "RT2" delays a subsequent transmission long enough for the master to detect a carrier/ start of message due to the other masters communication. If no transmission from the other master is heard during this interval, the granting master is free to initiate another transmission.

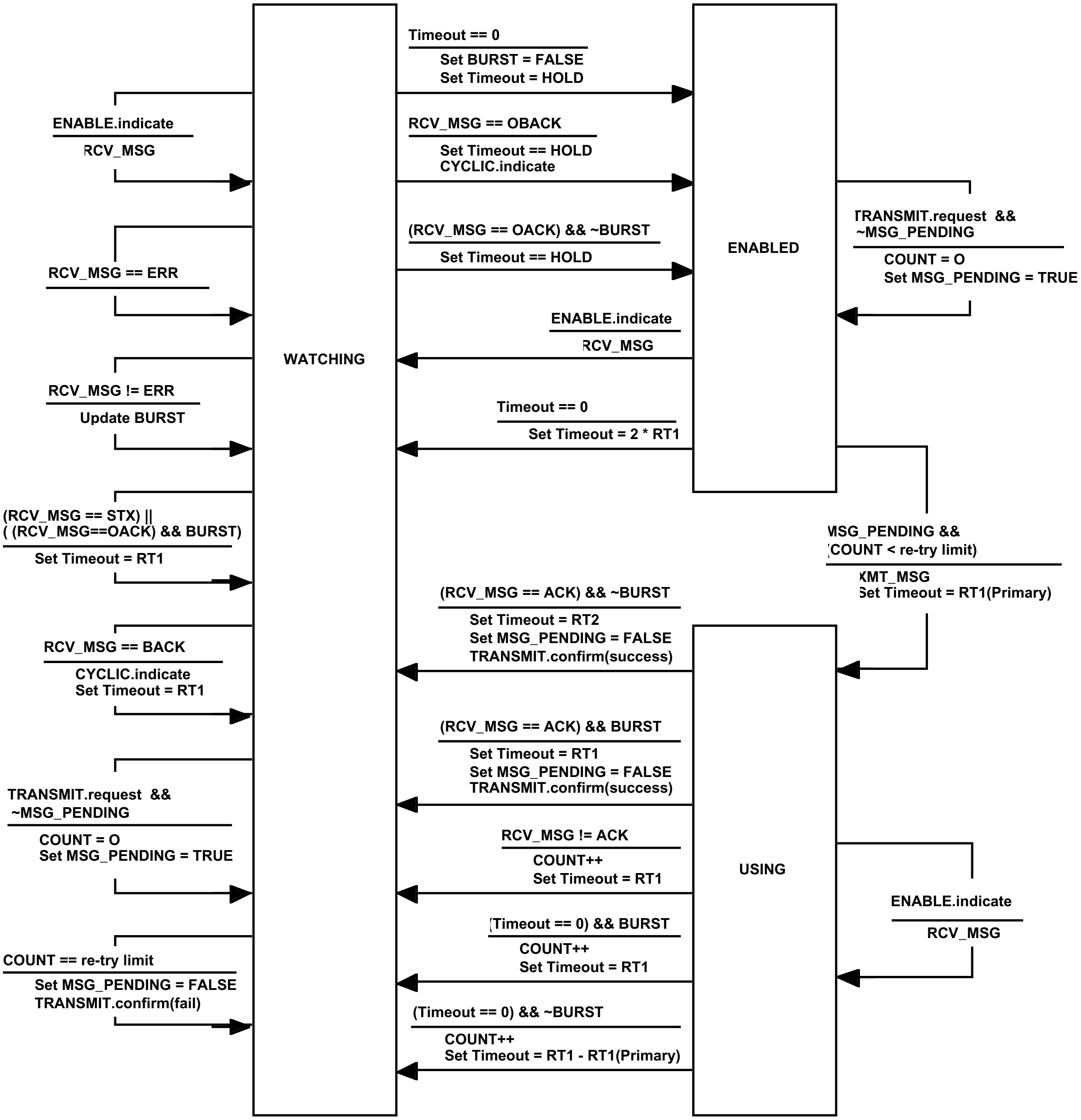

**Figure 18. Master State Machine**

The link quiet time / slave time-out defined by timer "RT1" is the larger interval. This time period must be long enough to ensure that any response from a slave would be received at a master and hence must be greater than "STO" by at least the turn-off and turn-on delays associated with carrier and message detection in the serial interface of the masters.

When used for link quiet time detection, "RT1" ensures that no ongoing transaction will be interrupted by a master which is seeking access to the communication link. It is the maximum interval of time that an unsynchronized master, which is trying to access the link, waits after seeing the end of any ongoing transmissions on the link. If no intervening transmissions occur, the link has become free and may be accessed. The arbitration and transmission process occurs as described in the next paragraph.

When an initial transmission request occurs, the state machine cycles through the arbitration process. First of all, a check is made for ongoing transmissions on the link. The link layer monitors all transmissions to extract link state information (kind of frame, and whether or not the device is in burst mode). The link is called "quiet" if no transmissions are seen for a period defined by the link quiet timer (RT1). The arbitration process is also considered complete if during this wait a response from a slave, or a burst mode message addressed to the other master is received. In either case, the master device is now in synchronization, and can immediately transmit its message. Message transmission must start during the window defined by the HOLD timer which determines the time the state machine remains in the "ENABLED" state.

Transmitting a message carries the state machine to the "USING" state. The retry count associated with the transmitted message establishes a re-try mechanism that allows the state-machine to recover from failures on the communication link. This is controlled by the link quiet/ slave time-out interval associated with timer RT1. This interval is the length of time for which a master will wait after sending a message to a slave before deciding that the transmitted message has been lost or that the slave has failed. *Note that this is the identical time period a master seeking to use the link needs to wait to ensure that no ongoing transmission is disrupted.* Different values are used for RT1 in the two masters to ensure that link synchronization can be recovered if masters make "simultaneous" (within the resolution of the carrier detect / framing mechanism) attempts to transmit.

The master will leave the USING state for the WATCHING state updating the retry count for this message if either "RT1" times out, or an illegally addressed or garbled message is received. If a correct response (ACK with status = OK or ERROR) is received, the transmission request has been completed and the master will also transition to the WATCHING state. In this case, the link grant timer ("RT2") is enabled.

At this stage the master is in synchronization in the WATCHING state. It monitors the link waiting for "RT2" to expire. This allows the other master, or burst mode device as the case may be, if any, on the link to transmit, if it needs to. Such a transmission by the other master must be initiated outside the window defined by the carrier/ message detect delays around "RT2".

When there is no device in burst mode on the link, if "RT2" expires without the master detecting any transmissions, the master is "ENABLED". It can then re-transmit a current message if an incorrect response (ACK with status = ERROR) was received and the re-try count has not been exceeded. It may alternatively transmit any new message associated with a transmit request that it receives while in this state or exit if no such request is pending.

Note If the master exits, (i.e., stops monitoring the link), it is no longer synchronized and will have to go through the longer synchronization time-out "RT1" before transmitting than the shorter "RT2" time-out if it is synchronized.

If, while waiting for "RT2" to expire, or in the USING state, the master detects a transmission on the link, then it must presume that the other master is undertaking a transmission. In this case it must now wait for as long as "RT1" or until it sees a response addressed to the other master on the link as it did when first attempting to access the link.

If there is a device in burst mode on the link, sequencing of access among masters that might be attempting to access the link is determined by burst mode messages. A master may only access the link after a burst mode message addressed to the other master. *Note that when a burst mode device exists on the link, it is the only active device permitted on that link. In other words, it is the only burst mode device on the link.* It is assumed that on system start up (from a power-off condition) a burst mode device will be up and bursting before hosts attempt to communicate with it using these arbitration rules.

# 8. PHYSICAL LAYER-SPECIFIC REQUIREMENTS

So far, all the Data Link Layer requirements have been Physical Layer independent. However, successful Data Link Layer operation requires consideration of Physical Layer characteristics. The signaling characteristics and response times affect Data Link Layer timing and Start of Message detection. Each of the following Physical Layer sections identify the following Data Link Layer requirements:

- The Character (Octet) Format;
- Gap Error detection timing;
- Start of Message (SOM) detection methodology;
- The structure of that Physical Layer's preamble; and
- The values of STO, HOLD, RT2, RT1(Primary), RT1(Secondary).

STO and HOLD are fundamental and critically control system performance. As a result, they must be made as small as possible. Once these values are specified, all other time-outs (i.e., RT2 and RT1) are based on these values and specific Physical Layer characteristics.

RT2 determines when a master can begin transmitting another message. RT2 must be selected such that:

1. The master with the token can begin a message; and
2. The master without the token can detect the start of a message transmission by the other master.

RT2 is based on the HOLD time and Physical Layer characteristics.

RT1, the link quiet timer, is used by master devices to synchronize to the HART channel. Basically, if a master connecting to the channel sees it quiet for this time then the master assumes it can safely begin a transmission. RT1(Primary) is based on STO and Physical Layer characteristics. RT1(Secondary) is based on RT1(Primary) and RT2.

While the following sections define Physical Layer-specific requirements and time-outs, Annex C provides a reference derivation of time-out values for each Physical Layer.

## 8.1. FSK Physical Layer

In the FSK Physical Layer data is transmitted as a series of asynchronous characters. Each bit in the character is then translated into an analog signal whose frequency is determined by the bit value. All data is transmitted at 1200BPS. FSK receiver training is accomplished by transmitting a preamble consisting of 5 or more 0xFF octets. Since only the asynchronous character's start bit is zero, the receiver's UART is synchronized after (possibly) loosing some 0xFFs after carrier turn-on.

**Table 6. FSK Physical Layer-Specific Requirements**

| Parameter | Requirement |
|---|---|
| Preamble | 5 each 0xFF bytes (strongly recommended) |
| Start of Message (SOM) Detect | Two consecutive 0xFF's followed by a valid Delimiter, no communication error or gap error. |
| Character Format | Asynchronous, 11 Bit characters (1 Start bit, 8 data bits, odd parity, 1 Stop bit) |
| Slave Time-Out (STO) | 28 character times |
| HOLD | 2 character times |
| Link Grant Time (RT2) | 8 character times |
| Link Quiet Time (RT1)<br>Primary<br>Secondary | <br>33 character times<br>41 character times |
| Gap Error | Must be less then 1 character time between any two characters. |

### 8.1.1. Preamble and Start of Message Detection

All frames transmitted by master, slave or burst mode devices are preceded by a specified number of hexadecimal "0xFF" characters. These characters are called the preamble to the frame. Many preamble characters (i.e., 0xFFs) may be received prior to the Delimiter. For the FSK Physical Layer, the start of message detection consists of two preamble bytes (0xFF's) and the delimiter is captured in sequence with no intercharacter gap.

No device should ever transmit less than 5 preamble characters. This allows sufficient time for asserting a carrier, training the modem circuit and for the listening device to begin receiving the message. Devices should send no more than 20 preamble characters.

> Note: While the maximum number of preamble bytes is not specified, the number actually transmitted directly affects system performance. In data acquisition applications, transmission of long preamble sequences can be very detrimental. As a result, transmission of excessive preamble characters is strongly discouraged.

In its response (ACK or BACK) message, a slave device should default to transmitting 5 preamble characters. If Common Practice Command 59 is supported then the master device may specify the number of preamble characters to be transmitted by the slave.

### 8.1.2. Character formats

Individual characters are transmitted asynchronously and consist of eleven bits. These are, in order of transmission, a start bit, eight data bits (least significant bit first), an odd parity bit, and a stop bit (see Figure 19). Characters forming a message must be transmitted in a single contiguous stream. There should not be a single idle bit (i.e., additional stop bits) between consecutive characters. Furthermore, consecutive characters in a frame must have intervening inter-character gap of less than 1 character time.

At 1200 bits per second (bps), transmitting a character takes 9.167 ms. This time period is the character time for the FSK Physical Layer.

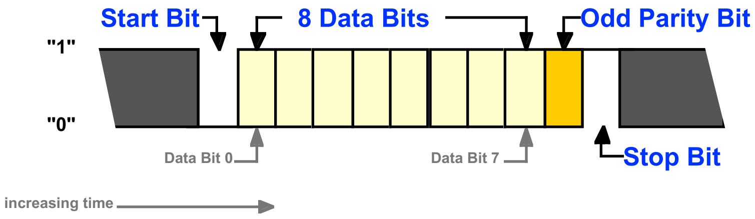

**Figure 19. Asynchronous Character Format**

## 8.2. RS-485

RS-485 is a valid Physical Layer for HART communications that has been used for many years. This section codifies the specifications specific to this Physical Layer

When using RS-485, data is transmitted as a series of asynchronous characters. Each bit in the character is then transmitted digitally using a balanced, differential voltage as specified in EIA/TIA RS-485 specifications. Receiver training is accomplished by transmitting a preamble consisting of 5 or more 0xFF octets. Since only the asynchronous character's start bit is zero, the receiver's UART is synchronized after (possibly) losing some 0xFFs at the beginning of a transmission.

A number data transmission rates (i.e., bit rates) are possible with 9600BPS; 19,200BPS; and 38,400BPS being most common. The preamble characters could be used to automatically detect the correct bit rate.

**Table 7. RS-485 Physical Layer-Specific Requirements**

| **Parameter** | **Requirement** | |
|---|---|---|
| Preamble | 5 each 0xFF characters (octets) | |
| Start of Message (SOM) Detect | Two consecutive 0xFF's followed by a valid Delimiter, no communication error or gap error. | |
| Character Format | Asynchronous, 11 Bit characters (1 Start bit, 8 data bits, odd parity, 1 Stop bit) | |
| Slave Time-Out (STO) | 28 Character Times<br>32ms | (1200bps)<br>( >1200bps) |
| HOLD | 2 Character Times<br>3ms | (1200bps)<br>( >1200bps) |
| Link Grant Time (RT2) | 8 Character Times<br>3ms + 6 Character Times | (1200bps)<br>( >1200bps) |
| Link Quiet Time (RT1) | | |
| Primary | 33 Character Times<br>32ms + 5 Character Times | (1200bps)<br>( >1200bps) |
| Secondary | 41 Character Times<br>35ms + 11 Character Times | (1200bps)<br>( >1200bps) |
| Gap Error | Must be less then 1 character time between any two characters. | |

### 8.2.1. Preamble and Start of Message Detection

All frames transmitted by master, slave or burst mode devices are preceded by a specified number of hexadecimal "0xFF" characters. These characters are called the preamble to the frame. Many preamble characters (i.e., 0xFFs) may be received prior to the Delimiter. The start of message detection consists of two preamble bytes (0xFF's) and the delimiter captured in sequence with no intercharacter gap.

No device should ever transmit less than 5 preamble characters and devices should not send more than 20 preamble characters.

In its response (ACK or BACK) message, a slave device should default to transmitting 5 preamble characters. If Common Practice Command 59 is supported then the master device may set the number of preamble characters transmitted by the slave as needed.

### 8.2.2. Character formats

Individual characters are transmitted asynchronously and consist of eleven bits (see Figure 19). Characters forming a message must be transmitted in a single contiguous stream. There should not be a single idle bit (i.e., additional stop bits) between consecutive characters. Furthermore, consecutive characters in a frame must have intervening inter-character gap of less than 1 character time.

## 8.3. C8PSK Physical Layer

The C8PSK Physical Layer groups the bit stream into symbols which are then transmitted as a series of phase shifts of the 3200Hz carrier. This technique is called Coherent 8-way Phase Shift Keying and it is similar to the signaling specified in the V.27 telecommunications standard. The 8 different symbols utilized allows each symbol to transmit three data bits. The C8PSK Physical Layer transmits at 3200 Baud and has a raw data rate of 9600 bps.

The C8PSK Physical Layer is fundamentally synchronous with a defined preamble sequence to train C8PSK receivers. One result is that each character (octet) is 9 bits long (8 data bits plus parity). Unlike asynchronous communication used in the FSK Physical Layer, no start or stop bits are included. In addition, there is no possibility for intercharacter gaps.

**Table 8. C8PSK Physical Layer-Specific Requirements**

| Parameter | Requirement |
|---|---|
| Preamble | 12.5ms (40 symbol) modem training sequence (see *C8PSK Physical Layer Specification*) |
| Start of Message (SOM) Detect | Reception of Start Flag and delimiter from the C8PSK Physical Layer |
| Character Format | Synchronous, 9 Bit characters (8 data bits plus odd parity) |
| Slave Time-Out (STO) | 32ms |
| HOLD | 3ms |
| Link Grant Time (RT2) | 14ms |
| Link Quiet Time (RT1)<br>Primary<br>Secondary | <br>42ms<br>56ms |
| Gap Error | Not Applicable |

### 8.3.1. Preamble and Start of Message Detection

A HART message transmitted over the C8PSK Physical Layer consists of four segments: the preamble, a start flag, the HART PDU, and a stop flag. The preamble, start flag, and stop flag are used to synchronize the C8PSK receiver. The preamble consists of a repeating series of symbols and synchronizes the receiver (see the *C8PSK Physical Layer Specification*).

The start flag signals the end of the preamble and the beginning of the HART PDU (see Section 5). The reception of the delimiter indicates the Start Of Message (SOM).

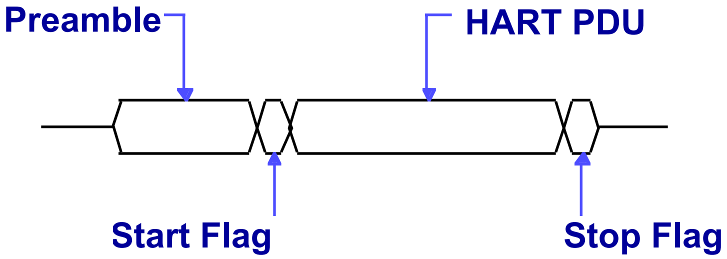

**Figure 20. C8PSK Message**

### 8.3.2. Character formats

Individual characters are transmitted synchronously and consist of nine bits which are, in turn, grouped into 3 symbols by the C8PSK Physical Layer. The HART PDU is transmitted as a single contiguous element with no inter-character gaps possible.

# ANNEX A MODEL FOR SERVICE SPECIFICATIONS

The interface between the Protocol layers is specified in terms of **Service Access Points (SAPs)**. The specification of a primitive consists of two parts, the first of which is a description of the **service**. This is a pseudo higher level language module / procedure definition describing parameters used by the service. The primitive specification's second part is the **Time Sequence Diagram (TSD)** which shows the order of events at the interface.

The SAPs together with the TSD define the responsibilities of the Protocol Layer from a "black box" perspective.

Primitives are identified using the form "NAME.function" where: the NAME, in all capital letters, indicates the service; and the function, in lower case letters, defines the action. Four functions (i.e., suffixes) are allowed:

- **.request** This initiates an sequence by the protocol layer user.
- **.indicate** This provides a indication to the protocol correspondent on the other side of the communications link.
- **.response** This is used by the protocol correspondent to reply to the previous .indicate primitive.
- **.confirm** This allows the protocol layer to confirm to the protocol layer user the completion of the previous request.

These primitives are organized into sequences (see Figure 21). There are three type of sequences in this specification:

- **Management Sequence**. In this sequence a service (indicated with the "**.request**" suffix) is invoked to access the service. The protocol performs the service requested and provides a primitive in reply (with the **.confirm** suffix) confirming its completion. The management sequence setting the number of retries to be performed by the master Data Link Layer (see Section 6.2.2) is an example of this type of sequence.
- **Unconfirmed Sequence**. This sequence consists of protocol layer user invoking a **`.request`** primitive and the correspondent (on the other side of the black box) receiving a **`.indicate`** primitive. A good example of this sequence is the transmission of a byte (see Section 6.1).
- **Confirmed Sequence**. This sequence consists of a protocol layer user invoking a **`.request`**; the correspondent receiving a **`.indicate`** and replying with a **`.response`**; and the user receiving a **`.confirm`** thus completing the sequence. The message transfer sequence in Section 6.2.1 is an example of a confirmed sequence.

Figure 21 shows the structure of a time sequence diagram. The time sequence diagram shows the protocol layer (e.g., the Physical Layer or Data Link Layer); the relationships between the primitives; and the relative timing involved. Time increases down the page. The protocol layer user (e.g., the Application Layer) is shown on the left and the correspondent at the other end of the link on the right. The diagonal arrows indicate events at each end of the communication link and, in some cases, represent transaction internal to the protocol layer. The events are labeled with the corresponding primitives.

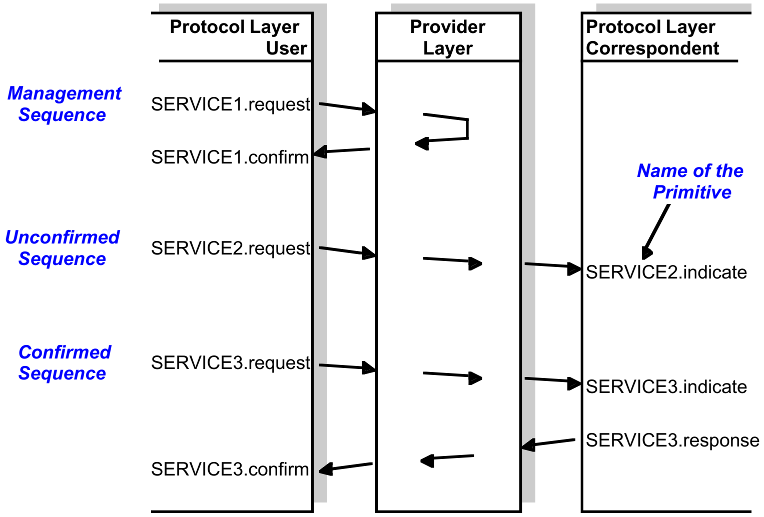

**Figure 21. Model for Protocol Service Specifications**

# ANNEX B STATE TRANSITION DIAGRAM NOTATION

The MAC requirements are specified with the assistance of finite state machines which are, in turn, illustrated using state transition diagrams. The notation is based on the real-time systems design notation described in [Hatley]. The state transition diagram notation is summarized in Figure 22. The following notation is used:

- Each MAC component (see Figure 22) is defined by a state-transition diagram.
- Labeled rectangles represent states that the component can be in.
- State transitions are directed lines. They show which state a component leaves and which state it transitions to.
- A transition is labeled with the events that caused it and the corresponding actions (this may be empty). A horizontal line separates the events (above the line) causing the transition and the resulting actions (below the line).
- Events and actions consist of: references to a SAP (e.g., ENABLE.indicate); the value of a state variable (e.g., BURST); or a call to another component (e.g., RCV_MSG).
- A tilde (" ~ ") means the ones complement or negation.

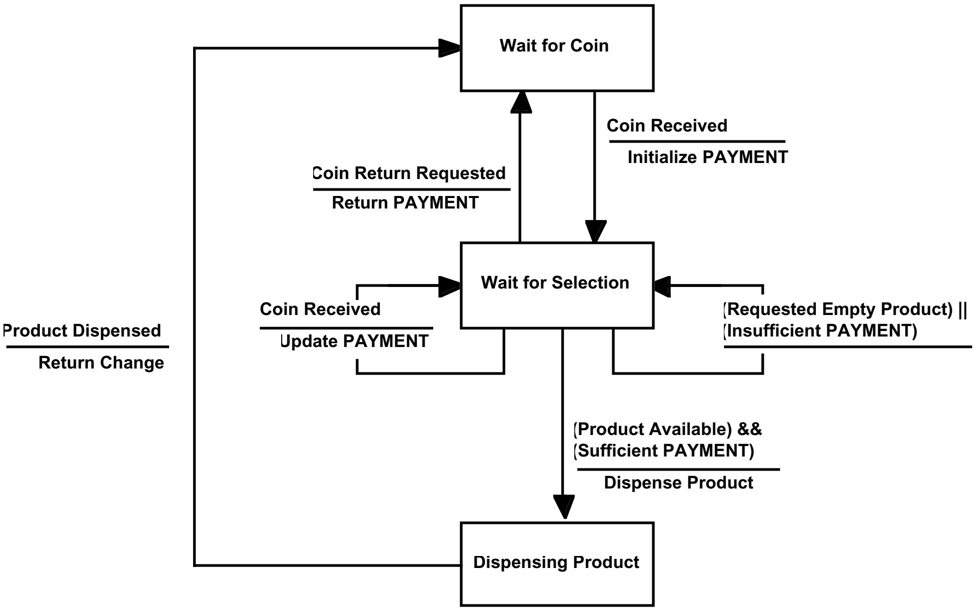

**Figure 22. Finite State Machine Model**

# ANNEX C DERIVATION OF TIME-OUT CONSTANTS

HART is an implied token passing protocol. The type of message indicates who has access to the network next. The Protocol then uses timers to determine when the token is lost and needs to be recovered. As a result, there are 4 timer settings that control token recovery and access to the channel:

- HOLD. The time allowed for a master to begin a transmission. This value is fundamental and is not directly derived from Physical Layer characteristics.
- STO. Slave Time Out. The time allowed for a slave device to begin its transmission. This value is fundamental and is not directly derived from Physical Layer characteristics.
- RT2. Link Grant Time. The time allowed by one (master or burst-mode) device to detect the start of a transmission by another master. This timer is set large enough to insure that this master can reliably detect the other master beginning its transmission. RT2 is based on the HOLD time and Physical Layer characteristics.
- RT1. Link quiet timer. This timer is different for the primary and secondary masters and represent the maximum time a master must wait to ensure there is no transmission pending on the channel. RT1(Primary) is based on STO and Physical Layer characteristics. RT1(Secondary) is based on RT1(Primary) and RT2.

This Annex will review the derivation of these timer values for the FSK Physical Layer, RS-485 and the C8PSK Physical Layer.

## C.1 FSK Physical Layer Timer Values

Figure 23 shows the link grant timing. Once a slave responds to this master then the other master has the token and must begin his transmission within the HOLD time. This master sets a timer to RT2, the link grant time. RT2 is selected to allow the other master to begin his transmission and this master to reliably detect that a message is in progress. If no message is detected by the time RT2 lapses, then this master can begin another transmission.

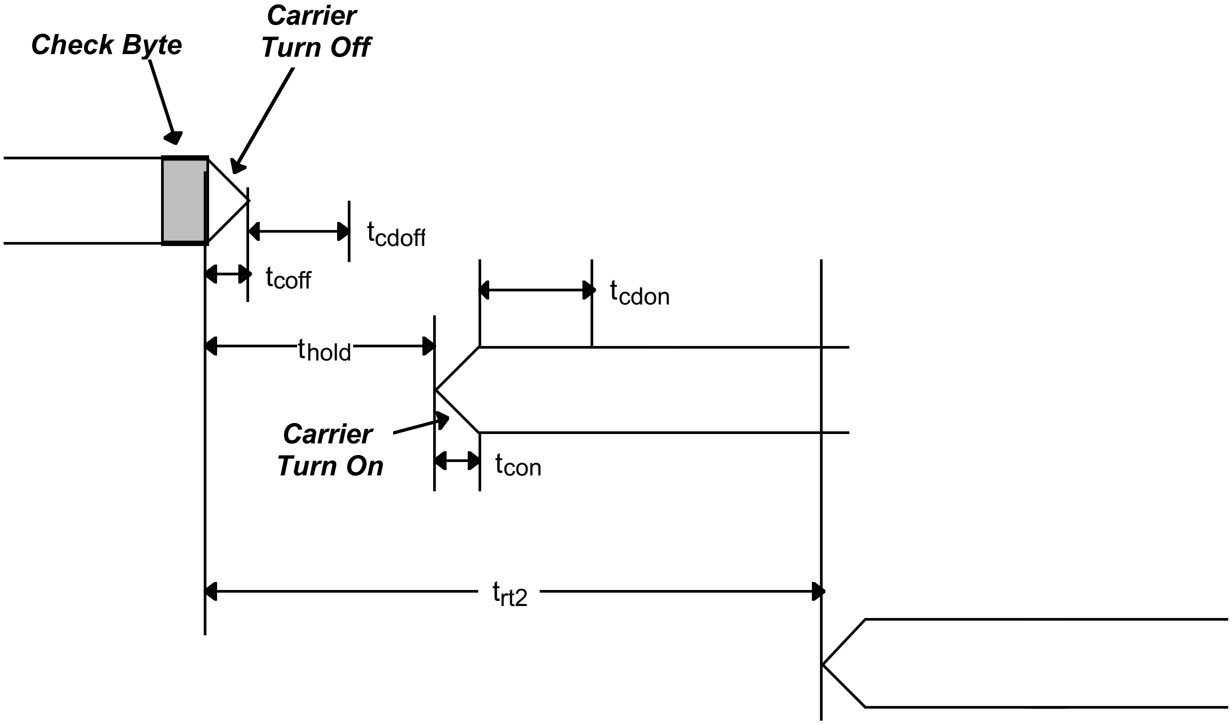

**Figure 23. FSK Physical Layer Timing**

Table 9 summarizes the timing values from Figure 23. The carrier related timing values are from the FSK Physical Layer Specification. Hold and Slave Time Out (STO) are the two fundamental timing values. These are simply selected to provide a reasonable compromise between network performance and ease of implementation. All other link layer timing values are derived from these two values.

**Table 9. Summary of FSK Timing**

| Symbol | Time (bits) | Definition |
|---|---|---|
| $t_{coff}$ | 5 | Carrier turn off time |
| $t_{cdoff}$ | 6 | Time from carrier off to carrier detect dis-assertion |
| $t_{con}$ | 5 | Carrier turn on time |
| $t_{cdon}$ | 6 | Time from carrier on to carrier detect assertion |
| $t_{hold}$ | 22 | Hold time. Time allowed for a master to begin its transmission |
| $t_{rt2}$ | 88 | RT2. Link grant time. After the time elapses this master will begin another transmission. |

### C.1.1 Link Grant Timer (RT2) Derivation

RT2 is a key time value and it determines when a master or a burst-mode device can begin another message. For the FSK Physical Layer, the start of a transmission is considered the receipt of an FSK preamble character (0xFF) from the other master. RT2 is made large enough to ensure that this event can be captured successfully thus indicating that the channel is busy. Figure 24 shows the delay of a HART transmission through a typical FSK receiver. There are 4 values that contribute to RT2:

- $t_{hold}$: 2 character times are allowed for the other master to begin its transmission (see Figure 23).
- $t_{cdly}$: is the delay that allows for the carrier detect signal to assert ($t_{con}$ + $t_{cdon}$). In early Physical Layer specifications (i.e., before the release of *FSK Physical Layer Specification* , Revision 8.0) this was specified as 30 bit times and rounded to 3 character times. For backward compatibility 3 character times will continue to be used.
- $t_{uart}$: is the delay for reception of one character. Since the HART Protocol allows for up to a gap of one character time between characters. This time is specified as 2 character times.
- For safety one more character time is added.

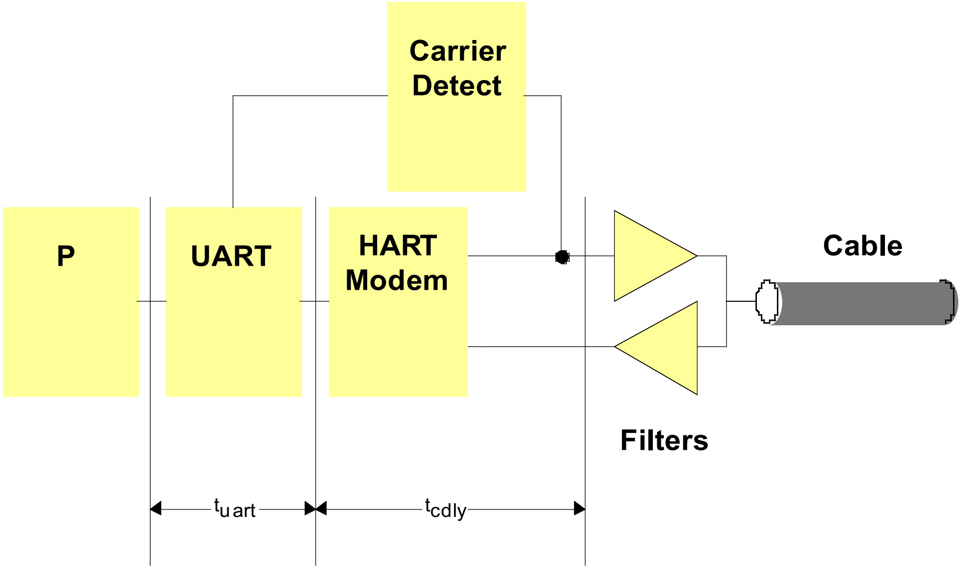

**Figure 24. Calculation of RT2**

RT2 is calculated as $t_{cdly}$ + $t_{uart}$+ $t_{hold}$ + 1 = 8 character times or 75ms.

### C.1.2 Link Quiet Timer (RT1) Derivation

RT1 is calculated in a fashion similar to that used for RT2. RT1, the Link Quiet Timer, is used by master devices to synchronize to the HART channel. Basically, if a master connecting to the channel sees it quiet for this time then the master assumes it can safely begin a transmission.

The fundamental value here is STO, Slave Time Out. For the FSK Physical Layer this value is 28 character times (256ms). A field device must begin its response within this time.

RT1(primary) is calculated as $t_{cdly}$ + $t_{uart}$ + $t_{sto}$ = 33 character times or 305ms. The secondary master must wait longer to allow the primary master to go first. This adds the RT2 time to RT1(primary), resulting in the value of RT1(secondary) as 41 character times or 380ms.

## C.2 HART over RS-485

The *HART RS-485 Physical Layer Implementation Definition* was originally a Rosemount Inc. internal document. This document was released to HCF members at the HART General Assembly in Bad Kreuznach Germany (see HCF_MIN-21). This document provides the basis for the derivation of the RS-485 timer values. Since RS-485 can operate at a variety of bit rates the timers will be scaled based on the bit rate in use.

As normal the values of the HOLD and STO are fixed:

- STO is 256ms at 1200BPS and 32ms at bit rates greater then 1200BPS
- HOLD is 20ms at 1200BPS and 3ms at bit rates greater then 1200BPS

The rest of the timers are the same as for the FSK Physical Layer when the RS-485 is operated at 1200BPS.

RT2 and RT1 can be determined by substituting the fixed STO and HOLD times into the Character time calculations from the FSK Physical Layer. This provides equations that allow the RT2 and RT1 time-out to be scaled for bit rates greater then 1200BPS. Calculating RT2:

RT2 = HOLD + 6 Character times = 3ms + 6 Character times

RT1 calculations are:

RT1(Primary) = STO + 5 Character Times = 32ms + 5 Character Times

RT1(Secondary) = RT1(Primary) + RT2 = 35ms + 11 Character Times

## C.3 PSK Physical Layer Timer Values

The PSK Physical Layer uses a classical modem training sequence as its preamble. Therefore, the Data Link Layer will not receive an 0xFF to indicate that the other master has started a transmission. In fact, if the Data Link Layer waits for message data, then the transmission will be well underway (actually, as much as half complete) before any data is received. As a result, the Data Link Layer must rely on the carrier detect signal to indicate the presence of a transmission by the other master. This is reasonable because any transmission while there is a valid carrier will not be successful and the resulting collision will not be detected by either of the transmitting devices.

### C.3.1 PSK Link Grant Timing

Figure 25 summarizes the RT2 timing for PSK and the actual timing values are shown in Table 10

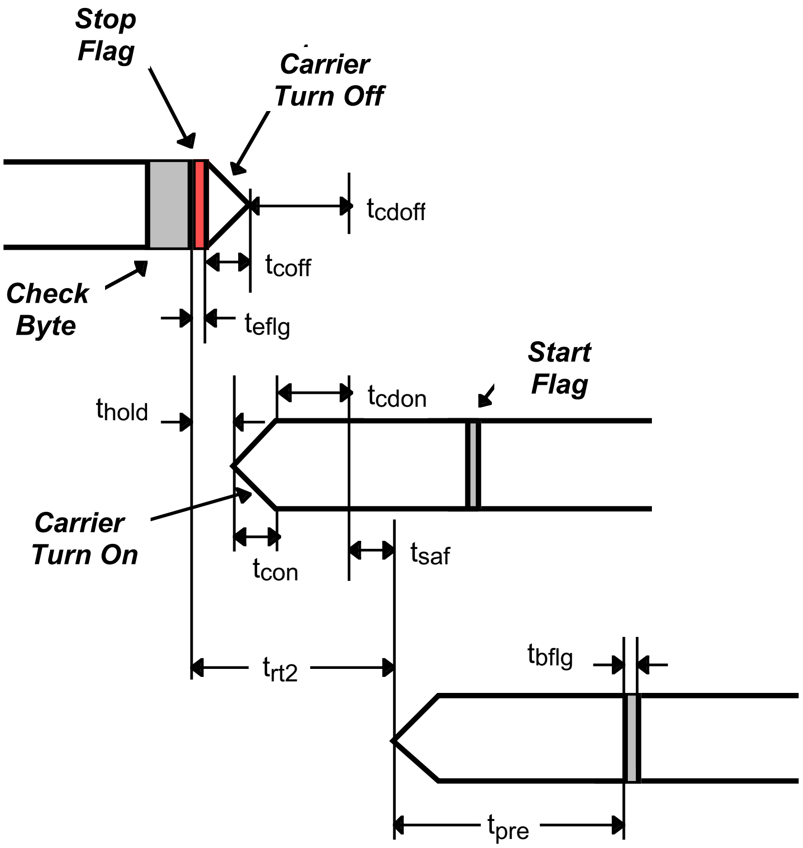

**Figure 25. PSK Physical Layer Timing**

The HOLD (3ms) and STO (32ms) values used for RS-485 are used for C8PSK. In addition, C8PSK requires the carrier detect to be used to determine the start of a message transmission.

**Table 10. Summary of PSK Timing**

| Symbol | Time (ms) | Definition |
|---|---|---|
| $t_{coff}$ | 0.94 | Carrier turn off time |
| $t_{cdoff}$ | 0.94 | Time from carrier off to carrier detect dis-assertion |
| $t_{con}$ | 1.00 | Carrier turn on time |
| $t_{cdon}$ | 3.00 | Time from carrier on to carrier detect assertion |
| $t_{eflg}$ | 0.94 | End flag |
| $t_{bflg}$ | 1.25 | Begin Flag |
| $t_{pre}$ | 12.50 | Preamble |
| $t_{saf}$ | 5.00 | Safety margin |
| $t_{hold}$ | 3.00 | Hold time. |
| $t_{rt2}$ | 13.00 | RT2. Link grant time. |

For PSK the network topology becomes more important as well. A safety margin must be included to accommodate delays through barriers and repeaters and the latency though the HART device hardware. For example, a digital repeater probably would monitor the line for a carrier detect to determine data direction. So it would need at least $t_{con} + t_{cdon}$ delay or 4.00ms. The safety margin can be reasonably set to 5.00ms.

RT2 for PSK can be calculated as $t_{eflg} + t_{hold} + t_{con} + t_{cdon} + t_{saf} = 14$ms.

### C.3.2 RT1 Derivation

RT1 calculations are:

RT1(Primary) = $t_{eflg} + t_{con} + t_{cdon} + t_{sto} + t_{saf} = 42$ms.

RT1(Secondary) = RT1(Primary) + RT2 = 56ms.

# ANNEX D REVISION HISTORY

## D.4 Changes from Revision 7.1 to Revision 8.0

Most changes to the document affected appearance only. Many sections were clarified based on frequently asked questions received at the HCF office. The principle functional change was the addition of frame expansion. Specific changes are as follows:

- General reformatting and addition of new sections: Preface, Introduction, Scope, References, Definitions, Symbols/Abbreviations, and Data Format added
- The whole document was edited to replace Transmitter Time-Out (TTO) with Slave Time-Out (STO).
- The Introduction to this specification contains material previously found in Sections 1 and 1.3.1 of Revision 7.1
- Section 5.1 includes material previously found in Sections 2.2.4 - 2.2.6
- Previous revisions contained the sentence: "Message framing is through a combination of a start of frame delimiter and a message length field." This has been expanded with the addition of Section 5.2, Framing.
- Section 5.3 includes material previously found in Section 2.2.2
- Section 5.4 includes material previously found in Section 2.2.3
- Section 6 includes material previously found in Section 2.3. In addition:
  - All TSD's were redrawn;
  - Block transfer SAPs were moved to the *Block Data Transfer Specification*;
  - Remote management SAPs were replaced with a simple Reference to the Application Layer; and
  - SAPs were added to support burst mode as well as Commands 113 and 114 (Catch Device Variable);
  - The services and attributes for the local management SAPs were updated.
- The explanation of the SAP and TSD notation was moved to Annex A and clarified.
- Section 7 includes material previously found in Section 2.4. All figures were re-drawn.
- Previous revisions contained the sentence: "The arbitration scheme is a very simple one based on an implicit token pass." Upon reviewing several sources, including the original patents, Section 7.1 was added to clarify this statement.
- The introductory remarks in Section 7.2 were added.

  - Figure 15 was corrected to show proper tracking of communication errors.
  - Two (previously missing) state transitions were added to Figure 18.
  - No other changes were made to the state machines or the requirements in Section 7.2

- All Physical Layer-specific requirements were consolidated in Section 8. RS-485 and C8PSK timing requirements were added.
- Section 8.1.1 includes material previously found in Section 2.2.1
- Section 8.1.2 includes material previously found in Section 2.4.5.1
- Annex A includes tutorial material previously found in Section 2.3.1
- Annex B includes tutorial material previously found in Section 2.4.1
- Annex C includes timing constant derivations previously found in Section 2.4.5

## D.5 Changes from Revision 7.0 to Revision 7.1

The most significant changes are confined to Section 2.2. This section is almost completely rewritten in order to clarify addressing, frame format, error detection and the slave's response when communication errors are encountered. Only small clarifications are made elsewhere in the document. The appearance of the DLL Specification was changed significantly as the result of formatting the document as per the HCF standards.

- Section 2.2.3 is renamed as "Error Detection Coding". The wording was changed to accurately reflect the required error detection coding and consistent use of "vertical parity" and "longitudinal parity". The associated Figure is reworked.
- Section 2.2.3.1 was added. This section and the accompanying table defines the slave response when a communication error is detected.
- The discussion of preambles was consolidated into Section 2.2.1: Preambles (in Revision 7.0 the preamble specification is split between Sections 2.2.4 and 2.2.1).
- The Specification was clarified to indicate that the preambles are a physical layer issue and that the preambles transmitted can vary from 5 to 20.
- A reference was added to Section 2.2.4 where the start character is defined.
- Figure 2 was decomposed. The diagram defining the start character is in 2.2.4.
- The discussion of addressing was consolidated into Section 2.2.2: Addressing (in Revision 7.0 the preamble specification is split between Sections 2.2.4 and 2.2.2). Subsections 2.2.2.1, Long Frame Addresses and 2.2.2.2, Short Frame Addresses were added. Figure 2 was decomposed. Two diagrams defining the long and short frame addresses were placed adjacent to the corresponding specifications.

- The specification was clarified to indicate that it is the slave's responsibility to set the burst mode bit correctly.
- A note was added to indicate that only Command 0 supports short frame addressing in order to maintain backward compatibility.
- The composition of the long frame address is now clearly defined. A table was added to indicate the preferred Device Type Codes for a given Manufacturer ID.
- Section 2.2 was modified to indicate that the characters in a frame are transmitted contiguously and should not have any gaps between succeeding characters.
- Since Section 2.2 is the first reference to the composition of a frame, the diagram of a frame was placed there.
- Section 2.4.5.1, Character Formats, was revised to clarify this requirement.
- The description of the Command byte in Section 2.2.4 was deleted.
- Notes were added to Table 4, referencing the Universal and Common Practice Command Specifications. The notes indicate that Command 59 is an optional Command.
- Section 1.3.1 was modified to be less restrictive, and the references to the 4-20mA loop were removed.
- Section 2.4.3 now clearly states that a burst should occur immediately after a slave response and must not start later that TT0.
- Section 2.4.3 now indicates that a burst mode device must burst within TT0 after an RT1 timeout waiting for a slave response.
- Section 2.4.3 now states that only one burst-mode device may be active on a link at a given time.
- The definitions of the use of the dash (-) and tilde (~) characters were added to Section 2.4.1. In addition a comment indicating that the actions in the state machines generally affect state variables was added.