SAE Aerospace
An SAE International Group

AEROSPACE STANDARD

AS5394

Issued 2002-02
Reaffirmed 2008-06

Health and Usage Monitoring System Advanced Multipoint Interface Specification

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

1. SCOPE:

This document specifies a multipoint, digital, serial interface that incorporates other interface standards such as EIA-485 and the nine bit interrupt mode of many microcontrollers. Standardized interfaces are critical to the development of an "open" HUMS architecture.

2. APPLICABLE DOCUMENTS:

EIA-485 “Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems”, Electronics Industries Association, April 1983

3. PROTOCOL BASICS AND DEFINITIONS:

3.1 Electrical and Timing Characteristics:

The electrical and timing characteristics of this serial interface shall comply with those described by EIA-485.

3.2 Referencing a Device:

All HUMS Advanced Multipoint Interface Specification (HAMIS) compliant devices shall be referenced using an 8-bit addressing system. (This specification was originally developed by the Rotorcraft Industry Technology Association (RITA) and is also known as RHAMIS.) There are three types of addresses used. Each device shall respond to its own unique address (local address), as well as a global address. Devices may also respond to one or more "group" addresses. The group address, can be used by a system designer to have multiple devices respond to a common command. Two of the 256 available addresses are reserved by this specification as shown in Table 1. The Master Bus Controller should always be Address 1. The user shall assign submasters. All addresses shall be fixed. Submaster shall not change its address once it has taken control over the bus.

TABLE 1 - Reserved Addresses

Address	Device
0	Global
1	Bus Master

3.2.1 Global Address: An address recognized by all devices. The global address is used for the exchange of global data and commands. No response shall ever be given to globally addressed command. As shown in Table 1, the global address is always 0.

3.2.2 Group Address: An address recognized by multiple devices. The group address is used for the exchange of group data and commands such as group triggers. No client device shall ever respond on the bus to a group address. Groups are intended to be used for synchronized sampling of smart sensors therefore:

- Most Line Replaceable Units (LRUs) will not be part of a group
- Master will not be part of a group
- Submaster will not be part of a group
- Not all devices need to be able to be part of a group in order to be compliant with this Specification

3.2.3 Local Address: An address recognized by a single device. The local address is used for the exchange of data and commands between two specific devices. Responses are allowed and recommended for locally addressed commands.

3.3 Speed:

The bus shall operate at 1 Megabaud.

3.4 Bit Order:

The least significant bit shall always travel first through the interface.

3.5 Bus Control:

Only one device shall have control of the data interface at any given time. This controlling device sends commands out and expects a response within a designated time. If no response is given in this time it is referred to as a time-out. Time-outs and the system response to time-outs are defined in the "Time-outs" section of this document.

3.6 Error Detection and Resending:

Error detection is handled at the message level and is defined in the "Status Message" section of this document.

3.7 Device Types:

All devices shall be one of the following types: (1) Master, (2) Submaster, or (3) Slave.

3.7.1 Master Device: The functions and features of the Master Device:

- Is also referred to as the bus master controller
- Is responsible for initializing the bus at startup and is the initial bus controller at startup
- Sends the initial commands and re-synchronizes the bus in the case of a "Bus Master Time-out"
- There is only one Master Device on a bus
- Can act as (1) a bus controller or as (2) a client, if it passes control to a submaster
- Master Device shall not be required to respond, on the bus, to incoming messages

3.7.2 Submaster Device: A submaster is any device that is not a master device but may need bus control to poll other submasters or slaves. A submaster, when in control of the bus, may query the bus master at which time the master shall function as a submaster. This device can act as a bus controller or client. When a submaster is not in control of the bus it shall respond as a slave or client.

3.7.3 Slave Device: Any device that never needs to be a bus controller. This device sends and receives data on the bus when commanded by a master or submaster device. This device will always act as a bus client.

3.8 Relationships:

Any two devices that are communicating shall do so in one of two established relationships. These two relationships are the command relationship and query relationship.

3.8.1 Command Relationship: The command relationship exists for the purpose of simple data collection tasks. In the command relationship, certain devices (slaves) exist simply to supply data from the outside world to other devices such as masters or submasters. In the command relationship slave devices shall never have bus control and can only perform one task. If more than one slave task exists for a device it shall be given an address per slave task. Of the two devices in a command relationship, one is the commander and the other is the slave. The commander is either a master or a submaster.

3.8.2 Query Relationship: The query relationship exists in the case where a device may need to take control of the bus in order to carry out its task. There are two devices in a query relationship a commander and a submaster. The commander is either a master or a submaster. A submaster cannot pass control to a different submaster but only back to the master.

4. PROTOCOL PROCEDURES:

Communication between any two devices shall follow one of the procedures specified below.

4.1 Command Relationship Procedures:

There are four types of command relationship procedures, they are:

- Global Acquire - Prepare, Acquire, Poll
- Group Acquire - Acquire, Poll
- Local Acquire – Acquire, Poll
- Fetch

Each of the above procedures (Global Acquire, Group Acquire, Local Acquire, and Fetch) has different overhead requirements and different levels of complexity. The devices should never respond on the bus to Global or Group commands. Command Source: not all commands will be transmitted from the Master or from the Slave, i.e. at no time should the Master be required to report an error condition to a slave or sub-master.

4.1.1 Global Acquire: The global acquire command relationship procedure is carried out for slave devices that are "smart" enough to handle a prepare statement. This method is preferred because of its ability to check a sensor before it is polled.

4.1.1.1 Prepare: The first step in this command relationship is the preparation. In the preparation step a "Prepare to Acquire Data" command shall be sent individually to each device from which the commander wants to receive data. The receiving device shall respond to this command with an error code or an acknowledgment. Errors in a device or transmission will be dealt with at the manufacture's discretion using standard or device specific error messages as described in 5.1.1 of this document.

4.1.1.2 Acquire: The second step is the "Acquire" command. The "Acquire" command shall be sent globally, by the commander, so that all prepared devices will collect data at the same time. There is no acknowledgment to this command due to its global nature. The time it takes for a device to acquire the first datum is device specific and will be referenced to the time when the stop bit, following the "To Address" in the message, is received.

4.1.1.3 Poll: The last step in the Commanding cycle is the device polling. In this step the polling command shall be sequentially sent to each slave device that contains data for collection. The device will return either the acquired data or an error code as defined by the device vendor.

4.1.2 Group Acquire: The group acquire command relationship eliminates the “Prepare” step and replaces it with a “Group Addressed Acquire” step. This procedure works by issuing an acquire command to a group address. All devices belonging to the addressed group will respond to this command by acquiring data. The acquired data is then collected by individually polling each device using their local addresses. Group addresses shall be assigned by the system designer.

4.1.2.1 Acquire: In the first step, an "Acquire" command is sent from the master or submaster device to the slave device(s). The "Acquire" command will be addressed to a specific group, as designated by the system designer. The details of the "Acquire" command are discussed later in this document. None of the devices addressed by this command shall respond to the "Acquire" command. The time it takes for a device to acquire the first datum is device specific and will be referenced to the time when the stop bit, following the "To Address" in the message, is received.

4.1.2.2 Poll: The second, and last step in the group acquire commanding cycle is the device polling. In the poll step, the polling command shall be sequentially sent to each slave device that contains data for collection. The device will return either the acquired data or an error code, as defined by the device vendor.

4.1.3 Local Acquire: The local acquire command is similar to the group acquire except the "local acquire command" commands a single device, whereas, the "group acquire command" commands a whole group of devices.

4.1.3.1 Acquire: In the first step, an "Acquire" command is sent from the master or submaster device to the slave device. The "Acquire" command will be addressed to a specific device. The details of the "Acquire" command are discussed later in this document. There is no response to an "Acquire" command. The time it takes for a device to acquire the first datum is device specific and will be referenced to the time when the stop bit, following the "To Address" in the message, is received.

4.1.3.2 Poll: The second, and last step in the local acquire commanding cycle is the device polling. In the poll step the polling command shall be sent to the slave device that contains data for collection. The device will return either the acquired data or an error code, as defined by the device vendor.

4.1.4 Fetch: The "Fetch" command relationship consists of one step and is used for interrogating very simple slave devices. The fetch command is sent to a slave device that responds with either data or an error message.

4.2 Query Relationship Procedure:

The query relationship exists for the case where a device may need to be the bus controller in order to carry out its task. There are two devices in a query relationship a commander and a submaster. The commander is either a master or a submaster.

4.2.1 Commanding the Submaster: Any command to the submaster shall require the submaster to process and/or retrieve information before responding to the request, otherwise it shall be a slave. All commands shall be executed in the following order:

- the submaster shall receive its command and then respond with (1) an acknowledge or (2) resend,
- once the command is acknowledged the master shall relinquish bus control to the submaster by issuing the "Control Pass" command as defined in 5.8 of this document, and
- the submaster finally responds to the "control pass" command with an acknowledgment or an error.

4.2.2 Submaster Operation: The submaster operates as a commander after receiving control of the bus. It can establish other relationships, command or query, before relinquishing control of the bus back to its commander. The bus master shall monitor activity on the bus. If the bus is idle for longer than the "Bus Master Time-out" time, the bus master will regain control of the bus by issuing the "Resynchronize" command. If the submaster has nothing to send, but is not ready to relinquish control of the bus, it must generate bus activity or it will loose control of the bus. The submaster can generate bus activity, and keep control of the bus, by issuing a global "Dummy" command. The Dummy message is defined in 5.2 of this document.

4.2.2.1 Relinquishing Bus Control: Once the submaster has completed its assigned task, it shall provide a response to the query that was initiated by the master. The format of this response is based on the original command sent by the master. After sending its data the submaster shall relinquish bus control back to the master using the "Control Pass" command.

5. COMMAND MESSAGES:

These are the sixteen industry standard commands that are reserved for use by this specification.

TABLE 2 - Industry Standard Commands

Command	Command Type	Command Description
0	Bus Client	Status
1	Bus Client	Data Message without Acknowledge
2	Bus Client	Data Message with Acknowledge
3	Bus Controller	Prepare (Master/Slave Relationship)
4	Bus Controller	Acquire (Master/Slave Relationship)
5	Bus Controller	Poll (Master/Slave Relationship)
6	Bus Controller	Fetch (Master/Slave Relationship)
7	Bus Controller	Control Pass (Master/SubMaster Relationship)
8	Bus Controller	Resynchronize
10	Bus Controller	Reserved
11	Bus Controller	Reserved
12		Reserved
13		Reserved
14		Reserved
15		Reserved

5.1 Status Message:

This message is sent in response to a command and indicates the status of the destination device. A status message will have one data byte that identifies the status type by "Status Code." Of the 256 possible status codes, the first 16 are specific to this interface specification and are given in the table below. The remaining 240 status codes (16 - 255) are for device specific status and shall be defined in the particular device specification.

TABLE 3 - HAMIS Specific Status Codes

Status Code	Status Description
0	No Error
1	Resend Message
2	Resend Entire Message Set
3	Not Ready
4	Device does not Support Command
5	Device Malfunction
6	___ Reserved
7	___ Reserved
8	___ Reserved
9	___ Reserved
10	___ Reserved
11	___ Reserved
12	___ Reserved
13	___ Reserved
14	___ Reserved
15	___ Reserved

5.1 (Continued):

The command behavior:

TABLE 4

Bus Controller	Command sent to	Bus Client
Command: Status	Each slave device	Response – status Command followed with Status Word

The device shall not respond to error-laden messages or commands it does not understand. If a device does not understand the message it should be quiet.

If the message required a response the Master will time out and then issue the same command and address again.

5.1.1 No Error: This shall be sent to acknowledge that the command or data was received with no errors.

5.1.2 Resend Message: This error shall be sent if the message had some recognizable error such as a checksum or byte count error.

5.1.3 Resend Entire Message Set: This error shall be sent if there was an error with the data sequence for long data strings made up of multiple messages.

5.1.4 Not Ready: This error message shall be sent if the device is busy gathering or calculating data and cannot return data at this time.

5.1.5 Device Does Not Support Command: This error message shall be sent if the device does not support this command.

5.1.6 Device Malfunction: This error message shall be sent if the device knows that it is malfunctioning.

5.2 Data Message Without Acknowledge:

The "Data Message without Acknowledge" command shall be used when you want to transfer data without the receiving device acknowledging the transfer. This reduces overhead and increases the data throughput of the interface. This command can be used with no data as a "dummy message."

5.3 Data Message With Acknowledge:

The "Data Message with Acknowledge" command shall be used when you want to transfer data and have the receiving device acknowledge receipt of the data.

5.4 Prepare (for Master/Slave Relationship Only):

This command shall ready the slave device to buffer data. It is used to "wake" the device up and allows the master to verify the device's operability. This command shall contain no data bytes.

The command behavior (reference 4.1.1.1):

TABLE 5

Bus Controller	Command sent to	Bus Client
Command: "prepare to acquire data"	Each slave device from which commander wants to receive data	• No response • Any error in a device or transmission – to be handled per manufacture's discretion

5.5 Acquire (for Master/Slave Relationship Only):

This command can be addressed globally, to a group, or locally to command data to be acquired. The three different uses of the acquire command are described in Section 4 of this document. There is no response to this command. Data can then be collected from the sensor(s) with the "poll" command. The "Acquire" command shall contain no data bytes. The format of the returned data will be defined in the particular device specification.

The command behavior (reference 4.1.1.2, 4.1.2.1, and 4.1.3.1):

TABLE 6

Bus Controller	Command sent to	Bus Client
Command: Acquire	all Prepared devices	No response

5.6 Poll (for Master/Slave Relationship Only):

These commands are sent to each slave device individually, commanding them to return acquired data. The format of the returned data will be defined in the particular device specification.

The command behavior (reference 4.1.1.3, 4.1.2.2, and 4.1.3.2):

TABLE 7

Bus Controller	Command sent to	Bus Client
Command: Polling	sequentially to each slave device that contains data for collection	• Responds with Data Message • If an error then it responds with the Status Command followed by the status word

5.7 Fetch (for Master/Slave Relationship Only):

This command is sent to each slave device individually, requesting them to acquire and return data. The format of the returned data will be defined in the particular device specification.

The command behavior (reference 4.1.4):

TABLE 8

Bus Controller	Command sent to	Bus Client
Command: Fetch	Slave devices	• device takes one data point and responds with Data Message w/o Ack Command • If an error then it responds with the Status Command followed by the status word

5.8 Control Pass (for Master/Submaster Relationship Only):

This command shall pass control of the bus from the master bus controller to a submaster or from a submaster back to the master bus controller. This command shall contain no data bytes.

The command behavior (reference 4.2):

TABLE 9

Bus Controller	Command sent to	Target submaster or master
Command: Control Pass	Submaster or master	Response - Acknowledge Command

5.9 Resynchronize (for Master/Slave Relationship Only):

This command is sent globally by the bus master to reestablish bus control. The bus master will issue the "Resynchronize" command in response to a "Bus Master Time-out." This command shall contain no data bytes.

The command behavior (reference 4.2.2):

TABLE 10

Bus Controller	Command sent to	Bus Client
Command: Resynchronize	Slave devices	• No response • Device will at least reset their bus interface • If slaves they will do a complete device reset

6. MESSAGE AND FRAME FORMATS:

6.1 Message Formats:

There shall only be one message format. This message format can consist of 5 to 4037 frames, with each frame containing 11 bits. The only required frames shall be the "To Address", "Command", "From Address", "Num. Of Data Frames", and "Checksum" frames in that order. Any "Data" frames would fit between the "Num. Of Data Frames " and "Checksum" frames. There can be zero to 4032 "Data" frames in a single message. If more than 4032 data frames must be sent, multiple messages can be put together to make an infinitely long string of data frames. In sending long strings the sequencing bit of the "Num. of data frames" frame shall be used as specified in 6.2.4.

TABLE 11

Byte	Frame
1	To Address
2	Command
3	From Address
4	Number of Data Frames
•	
•	Data
•	
N+5	Checksum

N = number of data bytes

It is recommended that each complete group of frames that make up a message be preceded by an 11-bit idle pattern. An idle pattern is the only way to insure that the receivers are synchronized with the beginning of a new message. Otherwise, glitches or other errors will not be resolvable until the idle pattern "resets" the receivers. The idle pattern must be 11 consecutive high bits (tri-state condition). The return message from a client back to the bus controller can immediately follow the outgoing command from the bus controller without an idle pattern.

For large data transfers, the message size should be known by the Master to ensure HUM system timing requirements are met. The slave device should never be allowed to increase the size of a data message. The master should control the data "sent" from a slave and when it is transmitted.

6.2 Frame Formats:

Messages are composed of the six 11-bit frames defined below. The first bit in every 11-bit frame shall be a start bit. The tenth bit is the address bit. The address bit of the "To Address" frame shall be set to "1". The address bit of all other frame formats shall be set to "0". The final bit of all 11-bit frames shall be the stop bit.