# RS485 Serial Communication Protocol – User

*Prepared by:*
**Empower RF Systems, Inc.**
**316 W. Florence Ave**
**Inglewood, CA, 90301**
**USA**

*Document Number: 0034-0001*
**Rev. A**

**6 September, 2012**

SIGNATURES

| Author | Signatures |
|---|---|
| Joe Storniolo | |
| | |
| | |
| | |

| Checked by | Signatures |
|---|---|
| | |
| | |
| | |
| | |

| Approved by | Signatures |
|---|---|
| | |
| | |
| | |
| | |

**Change Control**

| Doc Issue | Date | Paragraphs/Appendices/Annexes changed and reason for modification | ECO No. |
|---|---|---|---|
| 0034-0001 | 9/17/2012 | Updated user commands for: get supply voltage and get current.<br><br>Updates apply to firmware versions 00.99.40 or later. | 8054 |
| | | | |
| | | | |
| | | | |

# Table of Contents

# Scope

This document outlines the communications protocol for standard modules via a RS-485 hardware interface.

# Interface usage and general concepts

The communications protocol described in this document will be used for status and control of standard modules. The interface hardware and protocol may be extended as needed to other modules or designs with the appropriate updates to commands, etc.

## Hardware concept

The figure below provides a generic block diagram of module control hardware. The actual hardware may be different.

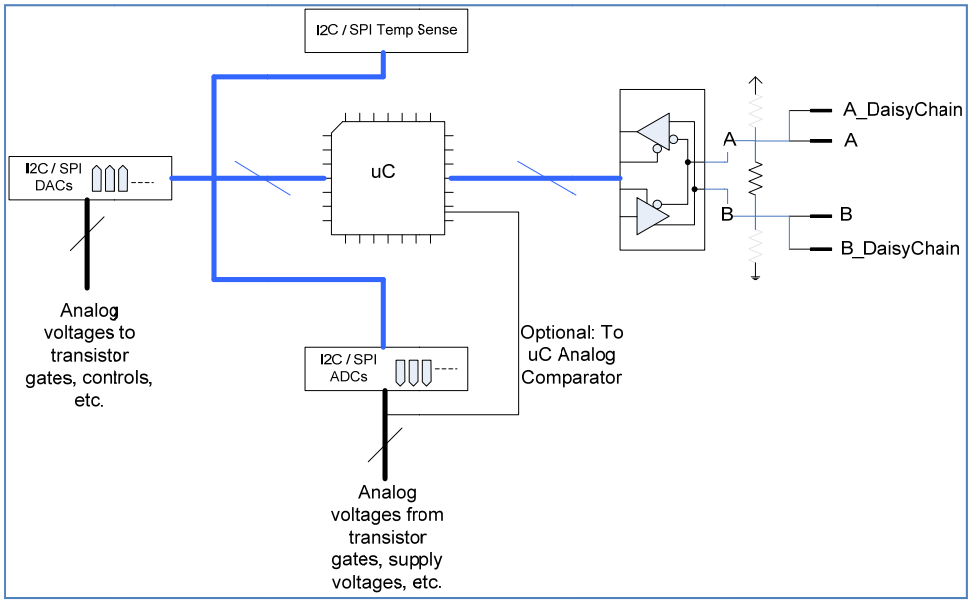


**Figure 1: Generic block diagram of control hardware.**

The following figure depicts a sample connection of masters and slaves. This particular figure depicts a star configuration. It is also possible to configure the masters and slaves in a daisy chain. This configuration allows maximum system level flexibility. Generally, the configuration will consist of a single master (or primary master) and a single set of slaves.

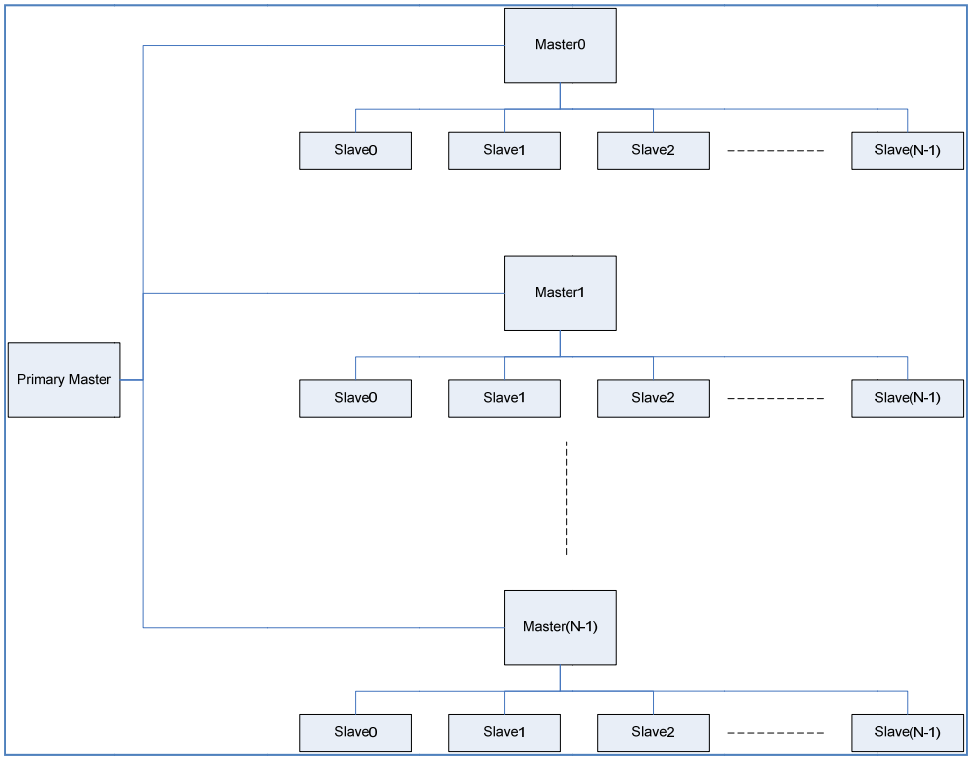


**Figure 2: Sample configuration of masters and slaves.**

## Duplexing

The RS-485 interface will support half-duplex communication. Half duplex operation is chosen to minimize the number of wires / connections required to communicate with the masters and slaves. Minimizing the number of wires is especially important when considering high power amplifiers and RF interference. Minimizing the number of wires also helps in space constrained, mixed signal (RF, DC and digital) PCB layouts.

## Serial communication settings

Any simple PC terminal application may be used for serial communication with an appropriate physical interface converter.

The following settings are used for serial communications:

- Baud rate: 115,200
- Start bits: 1
- Data bits: 8
- Stop bits: 1
- Parity: None
- Handshake: None

## Nodes and addressing

A maximum of 32 master nodes are allowed. In addition, each master node may have 32 slave nodes.

Consider a single master with some number of slaves. The slaves are selected by the master via an address frame at the beginning of the message. A slave will never transmit without first being specifically addressed and requested to do so by the master. Additional address details are provided below.

## Master-slave and Slave-master messaging

A single master controller or host computer will control the RS485 bus. The master role will never be transferred to a slave, and slave to slave direct communications are not allowed. That is, a slave will not transmit a message intended for another slave.

In the context of standard modules (or similar), modules will always behave as slaves and will only be given access to the RS485 bus when specifically addressed (as indicated above). If the slave was addressed in broadcast mode, the slave will not access the RS485 bus regardless of whether or not the slave received the message error free.

## Broadcast mode

In broadcast mode, the master transmits a message to all slaves on the bus. The broadcast bit in the message is set to logical 1 and the address bits are ignored by the slave(s). The slave(s) will not transmit after receiving a broadcast message.

# Frame and Message Structures

## Frame structure

The serial link will be asynchronous, and follows a typical RS485 frame structure. The frame structure is defined in the figure below.

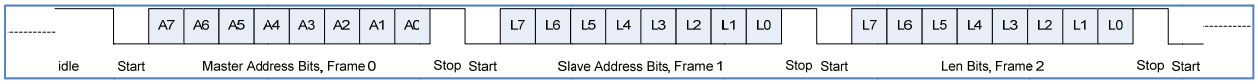


**Figure 3: Serial interface frame structure**

Each frame is 10 bits in length and will begin with a start bit, followed by 8 data bits and finally a single stop bit. Please refer to the serial communication settings section above for further information.

## Message structure

A message is comprised of at least 6 frames. The figure below provides the message structure.

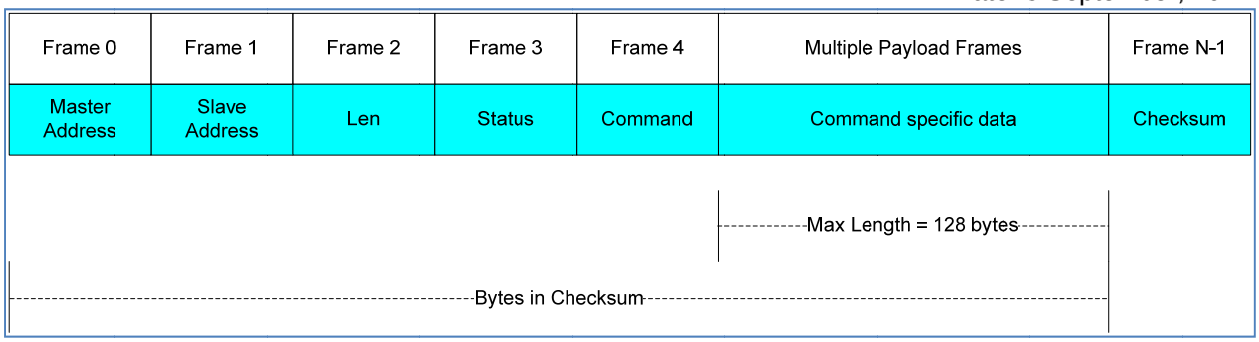


**Figure 4: Serial interface message structure**

## Address frames

The master address frame is frame 0, and the slave address is included in frame 1.

The figure below provides the address frame structure. This same structure is used for primary master to master as well as master to slave communications. For simplicity, the following assumes simple master-slave communications.

The address frame is populated by the master to communicate with a slave at address [A4…A0]. The addressed slave response includes the same address information provided by the master.

Example: Suppose the master issues a message with an address frame of 0x08h. The slave at address 8 decimal responds to the master with a message address frame of 0x08h.

Example: Suppose the master issues a message with an address of 0x88h (echo mode). The addressed slave responds to the master with a message address frame that contains 0x88h.

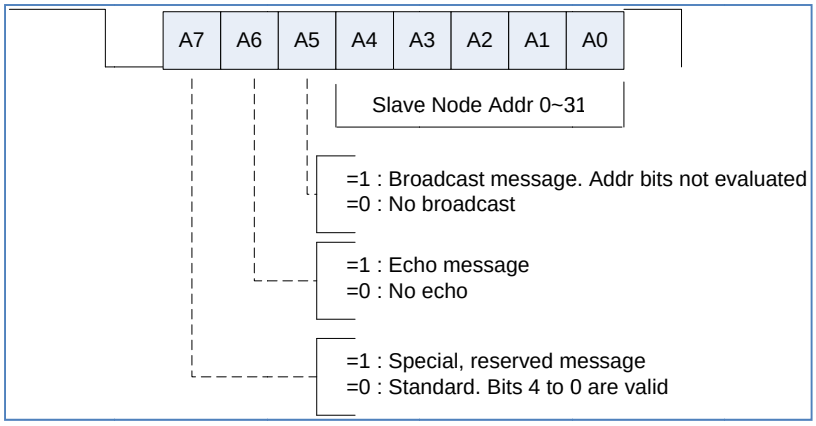


**Figure 5: Address frame definition**

| A7 | A6 | A5 | Addressing mode / operation |
|---|---|---|---|
| 0 | 0 | 0 | Standard addressing mode. Slave with the address specified in [A4…A0] will decode |

| | | | |
|---|---|---|---|
| | | | and respond to the master command(s). |
| 0 | 0 | 1 | Broadcast message. All slaves will decode and execute the message from the master. The slaves will not respond to a broadcast message; i.e. slaves will remain in high impedance state and will not place any data on the physical bus. |
| 0 | 1 | 0 | Echo message. The slave addressed in bits [A4…A0] will echo the message back to the slave after verifying the checksum is correct. The slave only echoes the message back to the master. It does not execute the command. |
| 1 | 0 | 0 | Special function - slaves ignore message |

**Figure 6: Table of address mode conditions**

### *Special, reserved messages*

If address frame bit A7 is set to a logical 1, the slave will ignore the message. This addressing / messaging mode is reserved for future use where other protocols may be required to share the same physical interface.

If address bit A7 is set to a logical 0, address bits [A4…A0] are valid and the slave at the specified address will respond accordingly.

### *Echo message*

If address bit A6 is set to a logical 1, the slave with the address specified in [A4…A0] of the address frame will respond by echoing the entire message back to the master without any changes after verifying the checksum is correct. This mode of operation is useful for debug purposes; i.e. step-by-step start up operation.

If address bit A6 is set to a logical 0, the slave with the address specified in [A4…A0] of the address fame will respond in a normal fashion.

### *Broadcast message*

If address bit A5 is set to a logical 1, all slaves on the bus will decode and execute the commands issued by the master regardless of the address specified in [A4…A0].

If address bit A5 is set to a logical 0, the slave with address specified in [A4…A0] of the address fame will respond in a normal fashion.

### *Incorrect addressing mode*

If the address bits [A7…A5] are not [0,0,0], [0,1,0] or [0,0,1], slaves will ignore the message from the master and will not take control of the bus. Messages beginning with an incorrect or unsupported address mode will not be supported by slaves.

### *Slave addresses*

Address bits [A4…A0] indicate the address of the slave for communications.

By default, the slave address will be set to 0x00h. The slave address may be modified in one of two ways:

- The slave address may be modified via the serial interface. The method for changing the slave address is outlined later in this document and requires that the slave does not have a predetermined hardware address.
- Alternately, the slave may have a predetermined hardware address. This is required for systems where multiple slaves are attached to the same bus; slaves may be moved throughout the system or replaced without regard for a software programmed address. If a hardware assigned address is used, the slave address cannot be updated by the master. The slave will respond to a change of address command with an error. The format for this error is defined later in this document.

### Length frame

The length frame indicates the number of frames in a message. The maximum number of frames included in a message is 134, and the maximum value of the length frame is 131.

The length frame is a simple count of the number of frames following the length frame itself. The length frame data does include the master address, slave address or the length frame itself.

Example:

Consider the following message: 00 00 03 00 08 0B (in hex). The third byte (from the left) is the length frame and has a value of 0x03h or 3 decimal. Notice that there are 3 frames following the length frame. These 3 frames include the status frame (0x00h), a command frame (0x08h) and finally the checksum frame (0x0Bh).

### Status frame

The status frame is filled by the slave and evaluated by the master. It indicates status of the last message issued by the master.

| Value | Description |
|---|---|
| 0x00h | Message received and decoded successfully |
| 0x11h | Invalid baud rate |
| 0x12h | Message incomplete |
| 0x13h | Checksum error |
| 0x14h | Transmit overrun |
| 0x15h | Receive overrun |
| 0x16h | Receive timeout |
| 0x17h | Port closed |
| 0x27h | Invalid command code |
| 0x28h | Invalid command data |

| | |
|---|---|
| 0x29h | Invalid message data |
| 0x2Ah | Access denied |
| 0x2Bh | Command not available |
| 0x2Ch | Data not available |

**Figure 7: Status frame values and descriptions filled by the slave**

### Command frame

The command frame specifies the desired action from the slave. Details of the command frame, command specific data frames, and the expected slave results are provided in the supported commands section of this document.

### Command specific data frames

Command specific data frames will be filled by the master and/or the slave. Details of the command specific data are provided in the supported commands section of this document. Note that not all commands require command specific data, and the number of command specific data frames is variable.

### Checksum frame

The checksum frame is used for error checking. The checksum is calculated as the exclusive OR of the data bits in each frame of the message.

Example: Suppose the following message is placed on the bus: 00 00 03 00 08 0B (in hex). The checksum is calculated as 00 xor 00 xor 03 xor 00 xor 08 and results in 0B.

## Response delay time

The slave will respond to a message in less than 2 seconds.

## Slave response after reset

The slave will be ready to receive and respond to messages within 5 seconds of a reset unless the reset is followed by a software download command. If the slave is commanded to a software download, the slave will be ready to receive and respond to messages within 15 seconds after the software download is complete.

## Engineering units

The slave is expected to be a simple micro-controller that does not support floating point arithmetic. Unless otherwise indicated, data values will be 16 bit (2 frames) and expressed in terms of tenths or hundredths. Data values will be populated in frames as most significant byte followed by the least significant byte.

| Parameter | Numeric Format | Range | Example |
|---|---|---|---|
| RF Power | dB or dBm: 16 bit signed, 1/100$^{th}$ of a dB or dBm | +327.67 / -327.66 dB or dBm | 31.25 dBm = [0x0Ch] [0x35h]<br>-10.5 dBm = [0xFFh] [0x97h] |

| | | | |
|---|---|---|---|
| | Watts: 16 bit unsigned, 1/10th of a Watt | 0 ~ 6553.5 W | |
| Voltage | 16 bit signed, 1/100th of a Volt | +327.67 / -327.66 V | |
| Current | 16 bit unsigned in 1/100th of an Amp | 0 ~ 65535 in hundredths of an Amp | |
| Temperature | 16 bit signed in whole degrees | +32767 / -32766 degrees Centigrade | |

**Figure 8: Engineering units**

## Structures and Tables

This section provides details on structures and tables. These structures and tables are used as part of setting and getting device configuration and general information. The actual usage of these structures and tables will be outlined in more detail in the supported commands section of this document.

### Module identification table

The module identification table contains ASCII characters representing various manufacturing and configuration information.

| Field | Number of Bytes (characters) |
|---|---|
| Company name | 24 |
| Model number | 16 |
| SKU number | 4 |
| Option builder number | 16 |
| Manufacturing date in YYWW (year-year week-week) format | 4 |
| Serial number | 8 |
| Hardware revision | 2 |
| Software revision in MM.mm.ii format | 8 |
| Last RMA number | 8 |
| Last RMA date in YYWW format | 4 |
| RMA count | 2 |
| Test station ID | 16 |
| PVT revision | 4 |
| Spare | 1 |
| Module type (for internal use only) | 1 |

**Figure 9: Module identification table**

### Data log table

Table for data log data.

| Field | Number of Bytes (characters) | Description |
|---|---|---|
| Alarm state | 1 | |
| Current attenuator value | 1 | |
| User attenuation value | 1 | |
| Multiplexer channel | 1 | |
| Temperature value | 2 | |
| DAC values | 16 | |
| ADC values | 24 | |
| Hi threshold alarms state | 2 | |
| Low threshold alarms state | 2 | |
| Hi threshold warnings state | 2 | |
| Low threshold warnings state | 2 | |
| Current limit error count | 2 | |
| 8 Volt shutdown error count | 2 | |
| Reserved | 2 | |
| Time stamp | 4 | |

**Figure 10:**

## User commands

User commands are detailed in the sections that follow. User commands are available for internal (factory) use as well as external customer use.

### NULL

The null command responds with message status. It is used for a basic test of the communications link.

| | |
|---|---|
| Command frame value (hex) | 0x00 |
| Length frame value (hex) | 0x03 |
| Command data | none |
| Slave response | Message status (1 byte)<br>Null command data frame (1 byte) |

Example:

```
[8/30 17:58:15.3]SEND: 00 00 03 00 00 03          ......
[8/30 17:58:15.4]RECV: 00 00 03 00 00 03 FF       .......
```

### Set address

Sets the slave software address from 0 to 31.

Notes:

- The software slave address can only be changed if the physical hardware address is 0x00h.
- Software slave address will remain in memory after a power cycle.

| | |
|---|---|
| Command frame value (hex) | 0x01 |
| Length frame value (hex) | 0x05 |
| Command data | New address data (2 bytes) |
| Slave response | Message status (1 byte)<br>New module address (2 bytes) |

Example:

```
[8/30 17:56:47.3]SEND: 00 00 05 00 01 00 01 05     ........
[8/30 17:56:51.2]RECV: 00 01 03 00 01 03 FF        .......
```

## Get status

Returns a status message that may be used for "heartbeat" polling of the slave.

| | |
|---|---|
| Command frame value (hex) | 0x02 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Temperature (2 bytes)<br>Overall current (2 bytes) |
| Example | Master Send: 00 00 03 00 02 01<br>Master Receive: 00 00 07 00 02 00 1D 1F FF F8<br>This indicates that the temperature is 0x001D and the overall current is 1FFF. |

Example:

```
[8/30 17:59:39.5]SEND: 00 00 03 00 02 01           ......
[8/30 17:59:39.5]RECV: 00 00 07 00 02 00 1E 1F     ........
                 0008: FF FB FF                    ...
```

This indicates that the temperature is 0x001E and the overall current is 0x1FFF.

## Get manufacturing information

Returns a composite data block with module ID information.

| | |
|---|---|
| Command frame value (hex) | 0x03 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Slave responds with module identification table as |

| | |
|---|---|
| | defined above. |

Example:

```
[8/30 18:05:44.0]SEND: 00 00 03 00 03 00          ......
[8/30 18:05:44.1]RECV: 00 00 79 00 03 45 4D 50    ..y..EMP
                 0008: 4F 57 45 52 20 52 46 20    OWER RF
                 0010: 53 59 53 54 45 4D 53 20    SYSTEMS
                 0018: 49 4E 43 2E 20 42 42 4D    INC. BBM
                 0020: 32 45 33 4B 4C 4F 20 20    2E3KLO
                 0028: 20 20 20 20 20 31 31 36         116
                 0030: 33 20 20 20 20 20 20 20    3
                 0038: 20 20 20 20 20 20 20 20
                 0040: 20 31 32 33 35 30 30 33     1235003
                 0048: 32 20 46 30 31 2E 30 30    2 F01.00
                 0050: 30 30 30 30 30 30 30 30    00000000
                 0058: 30 30 30 20 20 20 20 20    000
                 0060: 20 20 20 20 20 20 20 20
                 0068: 20 20 20 30 30 30 41 20       000A
                 0070: 31 00 00 00 00 00 00 00    1.......
                 0078: 00 00 00 7C FF             ...|.
```

## Soft reset

Does a restart of the system in which data is retained and configuration tables are reloaded (does not reload RAM buffers).

| | |
|---|---|
| Command frame value (hex) | 0x04 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command (1 byte) with 0x00 |

Example:

```
[8/30 18:09:39.0]SEND: 00 00 03 00 04 07          ......
[8/30 18:09:39.1]RECV: 00 00 03 00 00 03 FF       .......
```

## Set power up condition

Sets the state of the amplifier bias on application of DC or reset.

| | |
|---|---|
| Command frame value (hex) | 0x05 |
| Length frame value (hex) | 0x03 |
| Command data | 2 bytes:<br>0x0001 sets power up condition to bias enabled.<br>0x0000 sets power up condition to bias disabled. |
| Slave response | None |

Example:

```
[8/30 18:13:08.5]SEND: 00 00 05 00 05 00 01 01     ........
[8/30 18:13:12.3]RECV: 00 00 03 00 05 06 FF        .......
```

## Disable

Disables amplifier bias.

| | |
|---|---|
| Command frame value (hex) | 0x06 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte) |

Example:

```
[8/30 18:14:36.7]SEND: 00 00 03 00 06 05           ......
[8/30 18:14:36.8]RECV: 00 00 03 00 06 05 FF        .......
```

## Enable

Enables amplifier bias.

| | |
|---|---|
| Command frame value (hex) | 0x07 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte) |

Example:

```
[8/30 18:15:35.4]SEND: 00 00 03 00 07 04           ......
[8/30 18:15:35.5]RECV: 00 00 03 00 07 04 FF        .......
```

## Get temperature

Retrieves the temperature of the device.

Notes:

- This temperature may include an offset relative to the actual internal temperature sensor reading. This offset may be set as part of the temperature sensor configuration. Temperature sensor configuration is discussed later in this document.

| | |
|---|---|
| Command frame value (hex) | 0x08 |
| Length frame value (hex) | 0x03 |

| | |
|---|---|
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte)<br>Temperature in signed 16 bit (2 bytes) integer format |

Example:

```
[8/30 18:16:50.9]SEND: 00 00 03 00 08 0B          ......
[8/30 18:16:50.9]RECV: 00 00 05 00 08 00 20 2D    ...... -
```

This indicates that the temperature is 0x0020 or 32 degrees centigrade.

## Get alarms

Returns alarm status.

Notes:

- The alarm status returned is independent of whether or not alarms have been enabled. In particular, the hardware alarm line may be inactive and there is potential for an alarm.
- Alarm state bit mask:
  - Bit 0 is current limit alarm
  - Bit 1 is negative supply shutdown alarm
  - Bit 5 is PA enable. (PA enable is not truly an alarm, but is routed to the microcontroller for monitoring purposes and future use.)
- Threshold alarm and warning state bitmask:
  - Bits 0 through and including 11 map to ADC 0 through and including ADC 11.
  - Bit 12 maps to temperature

| | |
|---|---|
| Command frame value (hex) | 0x09 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte)<br>Alarm state bit mask (1 byte)<br>High threshold alarm state (2 bytes)<br>High threshold warning state (2 bytes)<br>Low threshold alarm state ( 2 bytes)<br>Low threshold warning state (2 bytes) |

Example:

```
[8/30 18:18:42.6]SEND: 00 00 03 00 09 0A          ......
[8/30 18:18:42.6]RECV: 00 00 0C 00 09 03 1F FF    ........
                 0008: 1F FF 00 00 00 00 06 FF    ........
```

This example indicates that the current limit and negative supply alarms have been tripped. See byte 5 with value 0x03. In addition, all high threshold alarms and warnings have been tripped. See bytes 6 through and including byte 9 with values 0x1F, 0xFF, 0x1F, 0xFF. All low threshold alarms and warnings have not been tripped.

## Clear alarms

Clears all alarms

Notes:

- If an alarm persists, the alarm status will return shortly after the clear alarms command is issued by the master.

| | |
|---|---|
| Command frame value (hex) | 0x0A |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte) |

Example:

```
[8/30 18:21:06.8]SEND: 00 00 03 00 0A 09          ......
[8/30 18:21:06.9]RECV: 00 00 03 00 0A 09 FF       .......
```

## Get current

Returns overall current consumption.

| | |
|---|---|
| Command frame value (hex) | 0x0B |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte)<br>Current (2 bytes) in 100ths of an Amp |

Example:

```
[9/13 17:05:25.8]SEND: 00 00 03 00 0B 08          ......
[9/13 17:05:25.8]RECV: 00 00 05 00 0B 00 00 0E    ........
```

## Get supply voltage

Returns the supply voltage.

| | |
|---|---|
| Command frame value (hex) | 0x0C |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte)<br>Voltage (2 bytes) in 100ths of a Volt |

Example:

```
[9/13 17:06:52.3]SEND: 00 00 03 00 0C 0F          ......
[9/13 17:06:52.3]RECV: 00 00 05 00 0C 00 00 09    ........
```

## Get RF input power

Returns the RF input power in dBm and in accordance with the units defined above.

Notes:

- This command is currently not supported

| | |
|---|---|
| Command frame value (hex) | 0x0D |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | |

Example:

*NOT AVAILABLE*

## Get RF (forward) output power

Returns the RF (forward) output power in dBm and in accordance with the units defined above.

Notes:

- This command is currently not supported

| | |
|---|---|
| Command frame value (hex) | 0x0E |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | |

Example:

*NOT AVAILABLE*

## Get RF reflected (reverse) power at output

Returns the RF reflected (reverse) power in dBm and in accordance with the units defined above.

Notes:

- This command is currently not supported

| | |
|---|---|
| Command frame value (hex) | 0x0F |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | |

Example:

*NOT AVAILABLE*

## Get input attenuation

Retrieves the user attenuation value.

Notes:

- The gain of the amplifier is also a function of temperature compensation. Variations in gain versus temperature are expected (assuming temperature compensation is enabled) independent of the user attenuation setting.
- The attenuation returned by this function is only the user set attenuation. It does not include the factory calibration or current temperature compensated attenuation value.

| | |
|---|---|
| Command frame value (hex) | 0x10 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte)<br>User attenuation setting in dBs (2 bytes)<br>The first byte is the whole number and the second byte is the decimal portion of the attenuation setting in dBs. |

Examples:

```
[9/4 15:36:30.4]SEND: 00 00 03 00 10 13          ......
[9/4 15:36:30.4]RECV: 00 00 05 00 10 00 00 15    ........

[9/4 15:47:30.1]SEND: 00 00 03 00 10 13          ......
[9/4 15:47:30.1]RECV: 00 00 05 00 10 08 05 18    ........
```

## Set input attenuation

Sets the user attenuation level.

Notes:

- The user attenuation value must be zero in order for the hardware attenuation voltage to have any effect.
- The user attenuation can only be positive; i.e. the gain of the amplifier can be decreased only.
- The gain of the amplifier is also a function of temperature compensation. Variations in gain versus temperature are expected (assuming temperature compensation is enabled) independent of the user attenuation setting.

| | |
|---|---|
| Command frame value (hex) | 0x11 |
| Length frame value (hex) | 0x05 |
| Command data | 2 bytes. The first byte is the whole number and the second byte is the decimal portion of the desired user attenuation value.<br><br>Examples:<br>30dB is 0x1E 0x00<br>0dB is 0x00 0x00<br>8.5dB is 0x08 0x05 |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte) |

Example for setting 8.5dB user attenuation:

```
[9/4 15:45:49.9]SEND: 00 00 05 00 11 08 05 19    ........
[9/4 15:45:49.9]RECV: 00 00 03 00 11 12 FF       .......
```

## Get data log

Returns log table data

Notes:

- See data log table definition above.

| | |
|---|---|
| Command frame value (hex) | 0x12 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte)<br>Command data. See example below and data log table definition above |

Example:

```
[9/7 10:32:53.3]SEND: 00 00 03 00 12 11          ......
[9/7 10:32:53.4]RECV: 00 00 41 00 12 22 10 00    ..A.."..
                0008: 01 00 1F 08 A2 08 98 09    ........
                0010: BF 0A 00 00 00 00 00 0B    ........
                0018: 54 06 5C 15 B3 05 07 15    T.\.....
                0020: B3 15 B3 00 B9 00 BE 00    ........
                0028: B6 15 B3 15 B3 02 EF 15    ........
                0030: B3 15 B3 00 00 00 02 00    ........
                0038: 00 00 02 00 00 00 00 00    ........
                0040: 00 35 1F 2E FF             .5...
```

This example returns the following information. Please see data log table above.

- Alarm state: 0x022
- Current attenuator value: 0x10 or 8dB
- User attenuation value: 0x00 or 0dB
- Multiplexer channel: 0x01 or channel 1
- Temperature value: 0x001F or 31 degrees C
- DAC values: A total of 8 DACs are available. Each DAC is represented by 2 bytes of data.
    - DAC 0: 0x08A2
    - DAC 1: 0x0898
    - DAC 2: 0x09BF
    - Etc.
- ADC values: A total of 12 ADCs areavilable. Each ADC is represented by 2 bytes of data.
    - ADC 0: 0x15B3
    - ADC1: 0x0507
    - ADC2: 0x15B3
    - Etc.
- Hi threshold alarm state: 0x0000
- Low threshold alarm state: 0x0002
- Hi threshold warning state: 0x0000

- Low threshold warning state: 0x0002
- Current limit error count: 0x0000
- 8 volt shutdown error count: 0x0000
- Reserved: 0x0000
- Time stamp: 0x0000351F

## Set data log

Sets the data log table.

Notes:

- This command is not currently supported

| | |
|---|---|
| Command frame value (hex) | 0x13 |
| Length frame value (hex) | 0x03 |
| Command data | TBD |
| Slave response | |

## Clear data log

Clears the data log from memory.

Notes:

- This command is not currently supported

| | |
|---|---|
| Command frame value (hex) | 0x14 |
| Length frame value (hex) | 0x03 |
| Command data | TBD |
| Slave response | |

## Emergency override

Disables all port pin and threshold alarms and clears any current alarms.

Notes:

- This command temporarily modifies the system configuration table; i.e. sets alarm mask bits. The override is cleared when the DC voltage is cycled or a reset is applied.
- All configuration changes are inhibited while in emergency override mode.

- This mode should be used with care, as all protection has been disabled and the device may be damaged.

| | |
|---|---|
| Command frame value (hex) | 0x15 |
| Length frame value (hex) | 0x03 |
| Command data | None |
| Slave response | Message status (1 byte)<br>Command data frame (1 byte) |

Example:

```
[9/7 11:37:32.9]SEND: 00 00 03 00 15 16          ......
[9/7 11:37:32.9]RECV: 00 00 03 00 15 16 FF       .......
```