DANTE Module Interface

Advanced Topic

This topic is best suited to advanced users – most users will not need to access the module interface (MOD). The MOD is used to:

- update plug-in module firmware (through RS232 or cellular modem)
- manually configure plug-in modules (without a config file from DANTE Config software)
- manually modify sensor driver scripts (without a config file from DANTE Config software)

The MOD is not used to communicate with sensors connected to RS232 modules!

Communicating with Modules

All modules share a single serial bus with the DANTE controller. This bus is called the MOD. Connect to the MOD from DANTE with OPEN MOD. Direct communication with modules is possible using a simple addressing system based on the Module ID (MID) or Local ID (LID) of installed modules. MID's 001, 002, 003 and 004 are built-in to the backplane. For these built-in modules the MID and LID are the same and cannot be changed. Each plug-in module is labeled in the factory with a 3-digit MID on a blue sticker.

MOD commands may depend on the module type being addressed. Many commands are common to all modules.

Module Command List

LID

Sets the Local ID (LID) of a module.
This command is for plug-in modules only. Attempts to change the LID of a built-in module result in an error.

MID's 001, 002, 003 and 004 are built in to the DANTE backplane. In these modules the MID and LID are the same. These LID's are reserved. The first available LID is 005.
In this example Module ID (MID) 019 is set to LID 005:

Module Identifiers

MID = Module ID, assigned at the factory
LID = Local ID, user-settable

All three digits are required when sending commands to modules!

A Typical MOD Command

+>001GETCD
'+' tells modules to clear receive buffers
'>' signals start of command
'001' is the module ID (MID) or local ID (LID) of the target module.

A Global MOD Command

+>*DISC
'*' in place of the MID tells all modules to listen to the command.

```
+>019LID=005
```

SLEEP

```
+>001SLEEP
```

Puts a module to sleep. There is no way to wake modules from sleep without ending and re-starting the pass through mode.

```
+>*SLEEP
```

Puts all modules to sleep.

NOTE: modules are automatically put to sleep at the end of program A or B and when the MOD pass through ends.

SAMPLE

```
+>001SAMPLE
```

Tells a module to start its instrument driver script.

DATA

```
+>001DATA
```

Retrieves the sample data buffer from a module. The data is wrapped in XML tags like this:

```
<SampleData mid='001' lid='001' pt='NOT SET'>
Data in here…
</SampleData>
```

WRITEFILE

Starts a file write mode to write the driver script to a module. Use END to exit the command and complete the file write. Use READFILE to verify the write.

```
MOD>+>001WRITEFILE
SERIAL ON 9600
12V ON
…more commands…
SERIAL OFF
12V OFF
END
001:OK; 0 Events
```

READFILE

```
MOD>+>001READFILE
<FILE mid='001' lid='001'>
SERIAL ON 9600
12V ON
…more commands…
SERIAL OFF
12V OFF
</FILE>
001:OK; 0 Events
```

RESETEC

Resets a modules event counters

```
+>*RESETEC
```

Resets all modules event counters (unless they are in sleep mode)

GETEC

Reads a module's event counters.

```
MOD>+>002GETEC
<EventData>
numEvents = 0
nextAddr = 8020
</EventData>
002:OK; 0 Events
```

DONE

Tests if a module driver script is running. Responds with OK if the script is finished or not running, responds with an error if the script is running:

```
MOD>+>002DONE
002:OK; 0 Events
+>002SAMPLE
002:OK; 0 Events
+>002DONE
002:ERROR: SAMPLE IN PROCESS; 0 Events
```

DISC

Starts a module self-discovery process. All modules that are awake and not busy running a driver script or another command should respond. There is a small chance that a module might not respond. The response order is random.

```
MOD>+>*DISC
[001,001,S9M RS232 V1.3F]
[004,004,S9M RS232 V1.3F]
[003,003,S9M RS232 V1.3F]
[002,002,S9M RS232 V1.3F]
```

The response format is:
[MID, LID, FIRMWARE VERSION]

12V ON / OFF / PERM

Controls the modules 12V power output.

```
+>00112V ON
```

Enables the module's 12V switched power output immediately if it is not already on.

```
+>00112V OFF
```

Disables the module's 12V switched power output immediately. This overrides and cancels the 12V PERM command.

```
+>00112VPERM
```

Enables the module's 12V switched power output immediately and tells the module to leave the output powered at the end of the driver script. This requires DANTE's MODPOWER to be set to PERM, otherwise DANTE disconnects the 12V power to the modules at the end of the sample A or B program.

NOTE: This setting can be changed by the driver script!

VB ON / OFF /PERM

Same as 12V ON/OFF/PERM, but controls the modules VB (5V) switched power output.

RESET

Resets a module. After reset modules are awake and ready to receive commands.

Firmware Update Commands

All components in the DANTE system allow field firmware updates, including updates through a cellular modem connection. Firmware files are text files with a few script commands and 50kB to 200kB of ASCII hex data. They are intended to be written to the F file in either DANTE or the COM unit, then either run with the RUN F command or sent to the modules with the SEND F MOD command. Please refer to the DANTE Field Firmware Updates document for more information.

FIRM

Starts a firmware update process. Use the END command or press escape the exit the firmware update mode. This command is usually used only in the F file (see RUN F).

CONFIRM

Verifies the integrity of the firmware loaded with the FIRM command prior to programming. If the test fails COM responds with:

```
ERROR: FAILED - repeat firm command; 0 Events
```

PROGRAM

Programs the firmware previously loaded with the FIRM command. The device resets after this command and enters sleep mode after reset.

VER

Displays details of the hardware and firmware, including compilation date:

```
COM>firm
HTYPE 13502-1-1
CD 29440960, 0
CODE TYPE S9COM
FIRM S9COM V1.1I
CDATE Nov  7 2013 12:15:58

OK; 0 Events
```