

Special PIDs

| PID | Value | GET | SET | Comment |
|-----------------------------|--------|-----|-----|----------------------------|
| ESTA_RDM_DEV_SET_UID | 0xFFE0 | --- | Y | |
| ESTA_RDM_DEV_NACKTEST | 0xFFE4 | Y | --- | |
| ESTA_RDM_DEV_RESPONSETIME | 0xFFE5 | Y | Y | Valid argument 100 - 13100 |
| ESTA_RDM_DEV_BREAKTIME | 0xFFE6 | Y | Y | Valid argument 40 -13100 |
| ESTA_RDM_DEV_PREAMBLE | 0xFFE7 | Y | Y | Valid argument 0-7 |
| ESTA_RDM_DEV_MUTEDISABLE | 0xFFE8 | --- | Y | |
| ESTA_RDM_DEV_QM_ENABLE | 0xFFE9 | --- | Y | (default is disabled) |
| ESTA_RDM_DEV_RESPONSEDITHER | 0xFFEA | Y | Y | (default is no dither) |
| PID_HELL_PWRUPTTEST | 0xC857 | Y | Y | |
| PID_HELL_INTERNALSTATS | 0xC862 | Y | Y | |
| | | | | |
| | | | | |
| | | | | |

Special PID Commands

Support is provided for a number of special ESTA development PIDS. These are for use in development and testing of products and systems. They are NOT intended to be supported or implemented in standard product.

ESTA_RDM_DEV_SET_UID

Argument : 6 bytes of Hex Data 0xMMMMDDDDDDDD as new UID

Where MMMM is 16bit Manufacturer ID and DDDDDDDD is 32bit Device ID.

Only available as SET command.

This command writes a user defined UID (Manufacture/Device ID) into EEprom, which overrides the inherent UID. The modified UID can be defeated by issuance of a SET: PID_FACTORY_DEFAULTS command.

ESTA_RDM_DEV_NACKTEST

This GET command takes a single byte argument.

Base on the argument, the command returns a NACK and NACK Reason code OR simulates a ACK_TIMER response.

This allows a controller to check its handling of various error and boundary situations.

NACK reason codes defined in the RDM Standard Table A-17

```

NR_UNKNOWN_PID           ((u16)0x00)
NR_FORMAT_ERROR          ((u16)0x01)
NR_HARDWARE_FAULT        ((u16)0x02)
NR_PROXY_REJECT          ((u16)0x03)
NR_WRITE_PROTECT          ((u16)0x04)
NR_UNSUPPORTED_CMD_CLASS ((u16)0x05)
NR_DATA_OUT_OF_RANGE     ((u16)0x06)
NR_BUFFER_FULL           ((u16)0x07)
NR_PACKET_SIZE_UNSUPPORTED ((u16)0x08)
NR_SUBDEVICE_OUT_OF_RANG ((u16)0x09)
    
```

Argument 0x00 : replies with NACK, Reason Code 0
Thru

Argument 0x09 : replies with NACK, Reason Code 9

Argument 0xFD: Reply_ACKTimer(0x0000); // boundary condition

Argument 0xFE: Reply_ACKTimer(0x0001); // 100ms Act Timer

Argument 0xFF: Reply_ACKTimer(0xFFFF); // boundary condition

ESTA_RDM_DEV_RESPONSETIME

The GET command responds with the current setting for the nominal turnaround time of the responder, and requires no argument.

The SET command requires a two byte argument and adjusts the turnaround time in units of approx 1us.

Default setting for this revision of hardware/software is 300. Other inherent processing delays will affect the actual response time. This time therefore represents a *minimum* time. Please use an oscilloscope if you need to determine precise timings.

WARNING : It is possible, using this command, to alter the operation of the responder and set timings outside those allowed by the RDM Standard. This may result in a loss of communications or failure to respond to subsequent commands. For example, setting the Response time to 150 WILL result in discovery responses before the allowed time.

The setting is volatile – default operation of the responder may be restored by power off/on, or issuing a PID_RESET_DEVICE or PID_FACTORY_DEFAULTS command.

ESTA_RDM_DEV_BREAKTIME

The GET command responds with the current setting for the nominal Break time of the responder replies and requires no argument.

The SET command requires a two byte argument and adjusts the response break time in units of approx 1us.

Default setting for this revision of hardware/software is 176. Other inherent processing delays will affect the actual response time. This time therefore represents a *minimum* time. Please use an oscilloscope if you need to determine precise timings.

WARNING : It is possible, using this command, to alter the operation of the responder and set timings outside those allowed by the RDM Standard. This may result in a loss of communications or failure to respond to subsequent commands. For example, setting the Break time to 100 WILL result in RDM responses having a break time duration less than that permitted by the standard.

The setting is volatile – default operation of the responder may be restored by power off/on, or issuing a PID_RESET_DEVICE or PID_FACTORY_DEFAULTS command.

Default setting for this revision of hardware/software is 176 (Nominal BREAK time of 176us)

ESTA_RDM_DEV_PREAMBLE

The GET command reports the current preamble count setting, and requires no argument.

The SET command requires a single byte argument in the range 0-7 and adjusts the number of preamble bytes returned as part of responses to DISCOVERY commands.

ESTA_RDM_DEV_MUTEDISABLE

This command takes a single byte argument and allows defeating the responder's ability to be muted.

Valid arguments are 0 (normal operation) and 1 (defeat MUTE operation).

Preventing a responder from Muting simulates a serious ERROR on the part of a responder, but is an aid to evaluating Controller discovery algorithms. In an ideal world, a controller should be able to cope with a responder that refuses (for whatever reason) to MUTE as part of the discovery process.

ESTA_RDM_DEV_QM_ENABLE

This command takes a single byte argument and allows enable/disable of the responder's ability to assert the existence of a QUEUED MESSAGE. The responder implements an internal timeout on IDENTIFY, so that IDENTIFY is automatically disabled after the (60second) timeout. If support for QUEUED MESSAGE is enabled, the message count field will be incremented by one when the timeout occurs. If a subsequent GET: QUEUED MESSAGE is received, the responder will return a GET: IDENTIFY_DEVICE showing that the unit is no longer in IDENTIFY mode.

By default, support for QUEUED MESSAGE is disabled.

ESTA_RDM_DEV_RESPONSEDITHER

This command takes a single byte argument and allows enable/disable of the responder's ability to "dither" the response turn-round applied to DISCOVERY_UNIQUE_BRANCH messages. Response "dither" to the discovery request is a pre-determined extension to the turn-round time, derived from the devices' UID. This ensures that responders with consecutive UID's do NOT respond together during discovery, increasing the prospect that collisions will occur. It has been noted that it is

possible for responders with certain UIDs to reply to discovery in such a manner that an apparently “clean” response will be spoofed. If this “spoofed” UID belongs to a real responder, that responder may be discovered out of turn, however controller discovery algorithms must not be broken by such events. Turn-round will be extended in the range 0 – 1.632ms.

By default, support for response dither is disabled (enabled in v2.7d)

PID_HELL_PWRUPTTEST

This is a Howard Eaton Lighting manufacturer specific command to allow set/reset of a responders inherent visual “Power On Self Test” (POST) operation. For the RDMLabpack responder, the POST operation is to set each LED on/off in turn. Not all Howard Eaton Lighting responder products support this command.

The GET command reports the current POST setting and the maximum allowed value of POST setting. No argument required for a GET command.

The valid SET command arguments are 0 (disable POST operation) and 1 (enable POST operation). This may change in future software releases.

PID_HELL_INTERNALSTATS

This is a Howard Eaton Lighting Manufacturer specific command to allow various internal counters to be reported.

GET :

| | |
|----------------------|--|
| DMX_NSC_PacketCount | : 16bit counter of Null Start Code packets |
| DMX_ASC_PacketCount | : 16bit counter of Alternate Start Code packets (not including RDM packets) |
| RDM_ASC_PacketCount | : 16bit counter of packets with RDM Start code |
| UartOverrunError | : 8bit counter internal Uart errors – should stay at 0x00 |
| eaddr_DeviceMinutes | : 8bit counter of minutes (on time) in increments of 2minutes |
| eaddr_BrownOut_Count | : 8bit counter of powerups resulting from Brownout |
| eaddr_WDT_ResetCount | : 8bit counter of powerups resulting from WatchDog Timer |
| eaddr_SWReset_Count | : 8bit counter of powerups resulting from Software Resets |
| ResponseDitherAdjust | : 16bit value used to extend discovery turn-round. |
| Record_SensorCount | ; 8bit counter of number of successful SET:RECORD_SENSORS commands received. |

All counters will rollover at 0xFF/0xFFFF.

SET :

Resets the following counters to 0.

UartOverrunError/DMX_NSC_PacketCount/RDM_ASC_PacketCount/DMX_ASC_PacketCount