

Introduction

The Howard Eaton Lighting RDM Led Responder provides a convenient means of verifying operation of RDM controllers and integrity of data passing through splitters and distribution equipment. The responder is capable of operating as a DMX device in its own right, and has support for many of the commands (PIDs) outlined in the RDM Standard.

In addition, a number of special PIDS are implemented that allow variation of response timings, break timings etc to allow some degree of testing under simulated error conditions.

Hardware

The software is intended for use with Andera Ltd CB06008B and currently utilises a PIC16F876 operating at 20 MHz.

The RDMLabPack is fitted with 6 RDM Responders and comes complete with DMX IN out DMX LOOP XLR connections. The RDM Labpack operates at 12V DC. DC Power input socket is Centre Pin +ve. A 500mA supply would allow you to have all the led mimics on at once !

Software version

These release notes apply to software version 2.9d 22--Oct-08 15:59:15 c3.249

Mimics

Each responder is fitted with five LEDs. As a default, each device has a DMX footprint of 4 slots. Four leds (Red/Green/Blue/White) can be controlled (intensity) from DMX512 data – null start code packets. The centre (Amber) Led is used as a status indicator for RDM activity. At power up, this led will flash at 1s intervals to indicate that the responder has NOT yet been muted. This normally corresponds to not having been discovered as an RDM responder.

Once the responder has been discovered and muted, the Amber led will revert to flashing on briefly to indicate receipt of a valid RDM message for this UID. The Amber Led is never controlled by slots of a Null Start code DMX packet.

By default, the responder runs a PowerUpSelfTest (POST) routine that lights each of the DMX controlled LEDs in turn, then ALL ON, then ALL OFF. Once all Leds are OFF, the unit is ready for discovery.

DMX Personalities

The responder supports two DMX personalities. Personality #1 allows simple DMX control of the individual R/G/B/W LEDs. Personality #2 allows this OR Preset Playback, as determined by the receipt of the appropriate RDM commands.

Default PIDS

PID	Value	GET	SET	Comment
PID_DISC_UNIQUE_BRANCH				
PID_DISC_MUTE				
PID_DISC_UNMUTE				
PID_DEVICE_INFO		Y	----	
PID_SUPPORTED_PARAMETERS		Y	----	Uses ACK_OVERFLOW and two packets
PID_PARAMETER_DESCRIPTION		Y	----	See notes – restricted use PID
PID_SOFTWARE_VERSION_LABEL		Y	----	
PID_DMX_START_ADDRESS		Y	Y	SET response uses ACK_TIMER
PID_IDENTIFY_DEVICE		Y	Y	Identify Flashes Red/Green/Blue/White LED

The GET:PID_SUPPORTED_PARAMETERS command is supported. The number of supported PIDs is more than can be reported in a single response packet (by design). Thus the responder issues an ACK_OVERFLOW to the first GET: and the balance of the list must be obtained by issuance of a second GET:PID_SUPPORTED_PARAMETERS. This is in accordance with the RDM standard.

If a controller does not handle ACK_OVERFLOW correctly it will not build a complete list of supported PIDs. If you do not see support for the “Special PIDs” described elsewhere in this document, it probably means you are not handling ACK_OVERFLOW correctly.

The other symptom of a controller not handling ACK_OVERFLOW correctly on this version of code is that alternate discoveries will get DIFFERENT lists. This will be trapped as an error on future versions.

Every command sent to the responder is checked for format errors. For example, sending a GET: with an argument when no argument is expected will result in the responder sending back a NACK, with a NACK reason code of FORMAT_ERROR.

Sending a command with data out of bounds (as far as the responder is concerned) will result in a NACK, and a NACK reason code of DATA_OUT_OF_RANGE.

A number of SET commands requiring saving data to non-volatile storage, which takes a finite time that is longer than the allowed response time for an RDM message. For this reason, many SET commands return an ACK_TIMER rather than an ACK response.

Device Info should return details as follows :

```
DEVICE_MODEL_ID       : 0x0001
DEVICE_CATEGORY_ID   : E120_PRODUCT_CATEGORY_TEST_EQUIPMENT
```

```
Footprint and Personality : 04/01 // as shipped but can be changed !!
```

```
DMX_START_ADDRESS    : 001 // as shipped but can be changed !!
RDM_SUBDEVICE_COUNT  : 0
SENSOR_COUNT         : 6
```

Supported PIDs

PID	Value	GET	SET	Comment
PID_MANUFACTURER_LABEL		Y	---	16 characters
PID_DEVICE_LABEL		Y	Y	16 characters
PID_DEVICE_MODEL_DESCRIPTION		Y	----	Hell RDM LED Effect
PID_PRODUCT_DETAIL_ID_LIST		Y		DETAIL_LED/DETAIL_PWM/DETAIL_TEST
PID_DMX_PERSONALITY		Y	Y	1= 4slot RGBW, 2= 4slotRGBW / RDM Preset
PID_DMX_PERSONALITY_DESCRIPTION		Y	---	4cct Led/4cct Led or Preset Playback
PID_SLOT_DESCRIPTION		Y	---	Refer known limitations
PID_DEVICE_HOURS		Y	Y	maintained
PID_LAMP_HOURS		Y	Y	Read/write only – not “actual” lamp hours
PID_LAMP_STRIKE		Y	Y	Read/write only – not “actual” lamp strikes
PID_DEVICE_POWER_CYCLES		Y	---	maintained
PID_SENSOR_DEFINITION		Y	---	refer known limitations
PID_SENSOR_VALUE		Y	---	refer known limitations
PID_RECORD_SENSORS		----	Y	Increments Record_SensorCount
PID_LANGUAGE_CAPABILITIES		Y	---	English!!
PID_LANGUAGE		Y	Y	Can set/get alpha codes, NULL resets
PID_PERFORM_SELFTEST		Y	Y	Selftest = 1 supported (Note4)
PID_SELFTEST_DESCRIPTION		Y	----	(added v2.7) Reports as “ST_1”
PID_FACTORY_DEFAULTS		Y	Y	Get : Current State of Flag
PID_RESET_DEVICE		----	Y	Restarts device, no ACK returned.
PID_CAPTURE_PRESET		----	Y	8 Presets supported
PID_PRESET_PLAYBACK		Y	Y	(Note5)
PID_REAL_TIME_CLOCK		Y	Y	Supports data format but does not have a real functional clock.
PID_LAMP_STATE		Y	Y	
PID_LAMP_ON_MODE		Y	Y	
PID_POWER_STATE		Y	Y	(Added v2.7)
PID_DISPLAY_INVERT		Y	Y	
PID_DISPLAY_LEVEL		Y	Y	
PID_COMMS_STATUS		Y	Y	Minimal support
PID_QUEUED_MESSAGE		Y	----	Minimal support
PID_STATUS_MESSAGES		Y	----	Minimal Support
PID_PAN_INVERT		Y	----	(Note1)
PID_TILT_INVERT		Y	----	(Note2)
PID_PAN_TILT_SWAP		Y	----	

NOTE1: PID_PAN_INVERT allows GET:SET of test data. Interacts with TILT_INVERT as it uses same internal test location.

NOTE2 : PID_TILT_INVERT allows GET:SET of test data. Interacts with PAN_INVERT as it uses same internal test location.

NOTE4 : PID_PERFORM_SELFTEST : Selftest 1 is implemented and should result in a short “on” long “off” flash cycle on the RGBW Leds. Set Selftest = 0 to disable.

NOTE5 : GET/SET:PID_PRESET_PLAYBACK : Preset Playback is only supported in Personality 2.

Sensors

Sensor 4 has had the Range Min/Max values adjusted to use the “Not defined” limits.
Sensor 5 has been changed since the v2.6 release and now mimics a temperature sensor.

The revised sensor assignment is as follows

Sensor Definitions

These are the values returned using the PID_SENSOR_DEFINITION command.

Sensor	Type	Text	Multiplier	Range Min	Range Max	Normal Min	Normal Max
0	Voltage DC	V_T1	10 ⁻¹	0	99.9	23.0	25.0
1	Voltage DC	V_T2	10 ⁻²	0	99.99	23.00	25.00
2	Voltage DC	V_T3	10 ⁻³	0	32.766	23.000	25.000
3	Current DC	I_T1	1	0	100	50	75
4	Current DC	I_T2	10 ⁻³	-32768	32767	0	32767
5	Temperature	Temp	1	-32767	32766	-32767	32766

Note that the Range Min and Range Max values for Sensor 4 indicate a range of “not defined” in accordance with the standard.

Sensor Values

These are the values returned using the PID_SENSOR_VALUE command

Sensor	Present Value	Correct Interpretation	Lowest Value	Highest Value	Recorded Value
0	121	12.1V	0	13.0	0
1	1201	12.01V	1	1299	0
2	12000	12.000V or 12,000mV	11999	12001	0
3	80	80A	1	99	0
4	1	0.001A or 1mA	0	0,002 or 2mA	0
5	-1	-1C	-100	100	0

The purpose of the first three sensors is to help check that any controller has correctly used the declared multipliers for each sensor.

Sensor5 now mimics a Temperature Sensor. Its range is set to reflect the boundary conditions for Min/Max. If these fail to display correctly it is likely that your controller is not handling 2's compliment numbers properly.

The number of times a correctly formatted SET:PID_RECORD_SENSORS has been received since power-up can now be retrieved using the PID_HELL_INTERNALSTATS command.

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

Timing Considerations

When looking at response timings, especially when using the responder in conjunction for the timing analysis of other products such as splitters, please use the PID_IDENTIFY_DEVICE command, which has a short message construct and minimal processing overhead within the responder. This will ensure the most consistent results, and fastest turn-round. Please note that there is considerably more overhead in creating the reply to (say) a GET:SUPPORTED_PARAMETERS, and the resultant response may be delayed beyond the selected turn-round time for times less than about 500us.

Software History

Improvements in version 2.9d

GET: ESTA_RDM_DEV_QM_ENABLE now supported.

GET : ESTA_RDM_DEV_MUTEDISABLE now supported.

GET/SET : PAN_TILT_SWAP no longer interacts with PAN_INVERT and TILT_INVERT.

Known Limitations in version 2.9d

Device_Power_Cycles counter wraps around at 0xFFFF rather than limit.

Device_Hours counters wrap around at 0xFFFFFFFF rather than limit.

Use of ACK_TIMER on set commands requiring writes to EEprom do not generate any subsequent Queued Messages.

Improvements in version 2.8d

As version 2.7d but with the following changes

Discovery turn-round time set to nominal 300us.

Discovery response dither set OFF

Discovery turn-round dither adjustment no longer affects non-discovery responses. Now only affects DISCOVERY_UNIQUE_BRANCH replies.

IDENTIFY_DEVICE no longer cancelled on receipt of UN_MUTE command.

Known Limitations in version 2.8d

GET: ESTA_RDM_DEV_QM_ENABLE not supported.

GET : ESTA_RDM_DEV_MUTEDISABLE not supported.

Device_Power_Cycles counter wraps around at 0xFFFF rather than limit.

Device_Hours counters wrap around at 0xFFFFFFFF rather than limit.

Improvements in version 2.7d

Changed resolution of Turnaround Timing, allowing wider range of adjustment.

Changed resolution of Break Timing, allowing wider range of adjustment.

Improved comms error stats collection.

Discovery response “dither” added to stagger timing of responses to DISCOVERY_UNIQUE_BRANCH messages. (optional)

Support added for PID ESTA_RDM_DEV_RESPONSEDITHER to enable/disable Discovery response dither.

Support added for flagging the existence of a QUEUED_MESSAGE when the internal timeout for IDENTIFY occurs. (optional)

Support added for PID ESTA_RDM_DEV_QM_DISABLE enable/disable of

Initialisation of Device Hours Counter corrected.

Sensor4 Definition now uses the correct values to indicate Range Min/Max “not defined”, and Normal Max “not defined”

Sensor5 now imitates a Temperature sensor, and has boundary condition values set for Min/Max Range and Min/Max Normal Values. Should report present temperature as -1C.

All sensor record flags updated.

Undocumented support for PID_HELL_SELECTDEFAULTS (0xc858) removed.

Support added for PID_HELL_INTERNALSTATUS (0xc862).

Support added for PID_POWER_STATE

Support added for PID_SELFTEST_DESCRIPTION.

Now reports NACK_UNSUPPORTED_CMD_CLASS where appropriate.

Now reports NACK_UNKNOWN_PID within the Discovery Command Class

Now reports NACK_PACKET_SIZE_UNSUPPORTED should the incoming packet exceed the internal buffer size.

Device Model ID corrected to avoid conflict with Howard Eaton Lighting StarLED dimmer code. (Now 0x0001)

DEVICE_DESCRIPTION_LABEL changed to " RDMLabpack Responder "

Avoids overrun errors when internal elapsed time counter updates EEPROM.

Known Limitations in version 2.7d

Support for PID_QUEUED_MESSAGES is declared, and indeed the responder will respond to a GET:PID_QUEUED_MESSAGES with a STATUS_MESSAGE. However, the responder DOES NOT actually queue any messages in response to SET commands that it has ACK'd with an ACK_TIMER.

RDM response turn-round time set to nominal 600us.

Changed to nominal 300us in v2.9d

Discovery response turn-round dither time adjustment affects all responses, not just DISCOVERY_UNIQUE_BRANCH.

Fixed in v2.9d

Discovery response dither set ON by default.

Fixed in v2.9d

GET: ESTA_RDM_DEV_QM_ENABLE not supported.

Fixed in v2.9d

GET : ESTA_RDM_DEV_MUTEDISABLE not supported.

Fixed in v2.9d

IDENTIFY_DEVICE cancelled on receipt of UN_MUTE command.

Fixed in v2.8d

Device_Power_Cycles counter wraps around at 0xFFFF rather than limit.

Device_Hours counters wrap around at 0xFFFFFFFF rather than limit.

Improvements in version 2.6d

Various bug fixes as documented.

Problem with SET:LAMP STRIKES fixed.

Problem with Sensor Text fixed.

Field size problem with Sensor Description fixed.

Sensor reporting examples improved, corrections to some sensor descriptions.

No longer responds to Development Startcode 0xF0 for RDM messages.

Known Limitations in version 2.6d

Support for PID_QUEUED_MESSAGES is declared, and indeed the responder will respond to a GET:PID_QUEUED_MESSAGES with a STATUS_MESSAGE. However, the responder DOES NOT actually queue any messages in response to SET commands that it has ACK'd with an ACK_TIMER.

Some chips programmed to v2.6d had incorrect EEPROM initialisation, giving rise to reporting of very large device hours counts.

Fixed in v2.7d

In some situations, the internal write to EEPROM that tracked elapsed time could cause a DMX or RDM packet to be discarded. This could sometimes mean that a single incoming RDM message was ignored if it coincided with the writes to EEPROM occurring at about 10minute intervals.

Fixed in v2.7d

Cannot make turn-around time shorter than valid minimum.

Fixed in v2.7d

Device Model ID conflicts with another Howard Eaton Product. (0x0102)

Fixed in v2.7d

Device_Power_Cycles counter wraps around at 0xFFFF rather than limit.
Device_Hours counters wrap around at 0xFFFFFFFF rather than limit.

Known Bugs/Limitations – v2.4d Software

PID_DEVICE_LABEL : Attempting to set a device label longer than 16 characters should give a NACK – Data Out of Range, as this device does not support 32 character device labels.

PID_DMX_PERSONALITY_DESCRIPTION : Personality 2 claims to be 3cct Led, and have a footprint of 3, but in fact the unit still responds to 4 slots. Only three slots will be supported when requesting slot descriptions.
Changed in v2.6d

When Personality 2 is selected, the *slot description* text appears as RedLED/GreenLED/BlueLED rather than RedLed/GreenLed/BlueLed. This is by design.
Maintained difference in v2.6d.

PID_CAPTURE_PRESET : Support is provided for a total of 8 Presets.
Maintained in v2.6d

PID_PRESET_PLAYBACK : is supported – but you must first set the DMX Personality to 2. This then allows you to enable/disable preset playback using PID_PRESET_PLAYBACK.
Maintained in v2.6d

Sensor Definition Text string returned at incorrect offset. Record Flags reported as 16bit when should be 8bit.
Fixed in v2.6d

Some PIDS that respond with text strings would return fixed length strings, such that there may be garbage data after the terminating null of the string. Particularly evident in Sensor Strings.
Fixed in v2.6d

PID_SENSOR_DEFINITION : A Get of sensor definitions returns a full 32 characters of “text” for each of the supported sensors. However the actual sensor string name is shorter, and thus the rest of the string buffer contains “garbage” – usually remaining characters from the software version string ! The actual sensor text is however NULL terminated.
Fixed in v2.6d

Sensor definitions for version 2.4d were not particularly useful.
Improved in version 2.6d

In some situations, the response message break times could be shorter than the required 176us.
Improved in version 2.6d
Please note the typical break time has now increased.

GET: ESTA_RDM_DEV_MUTEDISABLE not supported.
Fixed in v2.9d

Device Model ID conflicts with another Howard Eaton Product. (0x0102)
Fixed in v2.7d

Device_Power_Cycles counter wraps around at 0xFFFF rather than limit.
Device_Hours counters wrap around at 0xFFFFFFFF rather than limit.