

# GODDARD DESIGN CO.

## V4.30 RELEASE NOTES OCTOBER 16, 2015

### Release Notes for the DMXter4 RDM and DMXter4A RDM with Responder Man Pages

Older release notes are available at [http://www.goddarddesign.com/pdf\\_doc/V4.23\\_release%20.pdf](http://www.goddarddesign.com/pdf_doc/V4.23_release%20.pdf)

General changes and bug fixes.

- Many numerical displays have been expanded to 6 digits (Packet counters, Device Hours, etc). The counter freezes at >999K.
- V4.30 properly displays one-byte responses (start "View Raw" at slot 0 instead of slot one)
- V4.30 keeps the line driver enabled until the end of the last stop bit at the end of an RDM request. Older versions disabled the line driver ~3µs before the end of the stop bits.
- This is the code that is compatible with Benjamin Electric's RDM Integrity software. RDM Integrity requires that your DMXter4A RDM be registered as licensed.

Any users upgrading to this software must have installed our V9 bootloader . Most units having the V4.25 software will have this bootloader already installed. To determine which bootloader is installed in your unit, with the DMXter in "sleeping" mode, press and hold the UP, DOWN, LEFT and RIGHT keys, then press and release the TOP key. The display should now tell you the bootloader version and your serial number. A field upgrade of the bootloader can be done just prior to upgrading your software.

#### 4.3 AUTOFADING ONE SLOT

Fixed a long hidden bug in Autofade that would corrupt random memory.

#### 5.12 Time Graph of One Slot

An ornament, check it out.

#### 9 FLICKER FINDER

V4.30 fixes a display bug in Flicker Finder, seen when display mode was set to HEX.

#### 15A RDM INSERT MODE (DMXter4A RDM only)

With software version V4.30 or later the DMXter4A gains a powerful new feature. This is the **RDM Inserter Mode**. This allow a DMXter4A to pass live DMX512 packets from an outboard controller while still giving the operator full control of both the Standard and Advanced RDM routines.

When in the Inserter mode a DMXter4A RDM interleaves at least one DMX packet between RDM request/reply packet pairs. When no RDM request is ready, the DMXter will continue to send DMX packets. The default minimum interleave is 2 DMX packets per RDM packet pair. The MiniDMXter and the DMXter4 RDM are limited to interleaving static data consisting of whatever is in the transmit buffer. The Inserter mode is available in the RDM Controller Mode, and is also available in the Advanced RDM routines(if installed) with the following exception. Routines that allow you to change and view the Null Start Code data become view only.

#### 15.6.10 Identify Device - ON/ OFF

This test allows turning on and off identify while not changing anything else .It is a debugging tool.

#### 15.6.18 Change the Language Setting

We now support the language PID.

#### 15.6.20 E1.37-2 Features

We added support for the E1.37-2 IP Configuration PIDs.

#### 15.8.5 Get NSC Message Count

We now support a sent packet count of Null Start Code DMX512 packets during RDM testing

#### 15.11.4 Get Queued Messages on ACK Timer

The software now allows setting how many times the DMXter will attempt to retrieve a queued message.

#### 15.11.5 Interleave Null Start Code

Whether this message is displayed depends on the status of the Inserter Mode. If the insert mode is enabled this display will not be seen.

#### 15.11.6 NSC Data Source. (DMXter4A only)

This menu lets you select either static null start code data for the interleave, or live data from the input connector. This is a second way of switching you in and out of Inserter mode.

#### 15.12.11 Noise Before Break

After the controller sends a request packet it watches the line for the break of the response. If the DMXter sees any formatted bytes on the line before the break, those bytes are counted as 'noise before break'.

There is also a new result code that is similar to noise before Break. That code is issued for any response that does not contain a break..

#### 15.12.13 Get NSC Message Count

This display shows how many DMX packets have been sent since the last time the DMXter4 or 4A entered the Controller or the Inserter menu.

### 16.2 Capture Setup

Adds the ability to use two different capture rules by 'or-ing' them together.

```
|CAPTURE ON SLOT CRIT|  
| >1-8< 1-4 OR 5-8 |
```

#### 16.3.2 Sending a Custom Packet

On a good reply you will see the PLD of the response and up to four bytes of reply PD data.

```
|GOOD RESP. PLD = 2|  
|00H 04h |
```

### 16.4 Specific Tests

The specific Test menu has been reworked and expanded. The following have been added. I would check out the manual pages for details on these tests.

- \* Checksum Error (2 packets)
- \* Max Length Packet with Extra bytes at end
- \* Invalid Command Class (2 tests)
- \* Invalid Sub-Startcode
- \* Port ID Zero
- \* Fast Broadcast sequence
- \* Long and Short DUB requests
- \* Long and Short Broadcast requests

### 16.8 Autofade Null Start Code

We now warn if the user is trying to use autofade and NSC is disabled.

### 17 RDM FLAVORS

V4.30 adds a minimum break time flavor, this simulates an inline device shortening of the break.

It improves the accuracy of "RESPONSE DELAY" and "DISC. FIRST ACTIVITY" measurements - previous versions these times as 3 to 4us too high.

### 18 RDM SNIFFER

The entry menu is now the menu that repeatedly pressing the <GRAY> key returns you too. So you are less likely to exit accidentally.

The packet counters are all 6 digits and freeze at >999K.

The DMXter4 now includes a responder. We hope the following will give you a good idea of its capabilities. The responder is included in our V4.30 software.

## 19.0 GDC Test Responder

Our responder is test equipment. It is aimed at ensuring that a controller can handle common problems that sometimes are encountered with production responders. It has a rather short list of supported parameters. See 19.3 for the current list. What this responder is good at is testing the limits of the controller's protocol handling.

You can set the responder timing parameters over a wide range. This allows you to know your controller will work with all compliant and some noncompliant controllers, at least as far as the timing is concerned.

The responder UI has two different modes. One is a setup mode, the other a live responder.. Many protocol details can be changed while the unit is responding.

On entry you will be asked if the responder is "AT END OF LINE?". This messages is only seen once, however you will be given a chance to change the termination / pass thru function later in the setup menu. After the termination question the unit enters a three item menu loop.

```
START RESPONDER?
```

See Section 19.1 below

```
RESPONDER SETUP?
```

See section 19.2 below

```
BACK TO MAIN MENU?
```

Loop back to the Start Responding menu item.

### 19.1 Responding

#### 19.1.1 Request Monitor 1

```
|          MUTE          | or          |          MUTE          |
| GR NAME LAST PID     |           | (NO RDM TRAFIC)     |
```

GR = get request SR = set request - If the responder has been muted this will be shown.

#### 19.1.2 Request Monitor 2

```
|DMX ADDR 512, PERS 2|
|  DEVICE LABEL      |
```

#### 19.1.3 DMX NSC Data Monitor

```
|NSC DATA (%) 12345|
| 0 100 0 100 99 |
```

This item shows the first 5 slots of NSC data addressed to this responder. It also shows the NSC packet count. For this item **only** the <YES/Q> key cycles you thru display formats; *Yes it is not our standard interface. There were reasons. . .*

#### 19.1.4-5 Mute/ Unmute Responder

```
|          RESPONDING          | |          RESPONDING          |
|          MUTE?              | |          UNMUTE?            |
```

#### 19.1.6 Status Message Generation Sub Menu

```
|          RESPONDING          |
| GENERATE MESSAGES?         |
```

These items enable sending Status Messages when <YES/Q> is pressed. Note that this version of the software does not support Queued Messages. The controller must ask for a Status message. A Queued message can be sent by a responder whenever it is polled.

##### 19.1.6.1 Setup A Status Message

```
|STATUS MGS: STD/ERR |
|          GENERATE? |
```

When armed the bottom line will read PENDING . . .

When the controller asks this responder for a Status Message, it replies with a 16 bit status type and two 16bit data values. There are both public (standard) and manufacturer Status Types. See section 10.3 of E1.20 and table xx for more information on Status Messages. The first Status Message queued up is:

Standard Error	STD/ERR 0045h - Panic Mode -	Data 1 = + 0 Data 2 = + 0
Scrolling with the <Down> key will enable:		
Standard Warning	STD/WRN 0042h - breaker trip -	Data1 = + 0 Data 2 =+ 0
Standard Advisory	STD/ADV 0041h - no dimmer -	Data 1 =+ 0 Data 2 =+ 0

Manufacturer Status Messages:

Error	MFG/ERR	8000h	“Stage manager meltdown”	Data 1= + 0 Data 2 = + 0
Warn	MFG/WARN	8001h	“performer %d min late for call”	Data 1 =+ 10 Data 2 = + 0
Advisory	MFG/ADV	8002h	‘Soprano off key Page %d line %d	Data 1 = + 22 D2=+ 6

How this data will be displayed varies controller to controller. There is an optional Status Description PID that returns a text message of up to 32 characters. The current version of the DMXter Responder dose not support the description field for the defined PIDs. It does support them for the manufacturer specific PIDs.

19.1.6.2 Clear A Status Message

```
| STS CLEARED: STD/ERR |
| GENERATE? |
Standard Error          STD/ERR
Standard Warning        STD/WRN,
Standard Advisory       STD/ADV
Manufacturer Error      MFG/ERR
Manufacturer Warn       MFG/WRN
Manufacturer Advisory   MFG/ADV
```

19.1.6.3 Manufacturer Status Message with Alternate Data

```
| MFG/ADV W/ALT DATA |
| GENERATE? |
```

A real responder may generate status messages for many conditions. The DMXter Responder is test equipment and generates the same response each time it is asked. However, for testing it is useful to have status message with differing data for testing. This item generates the following Status Message,

Advisory MFG/ADV 8002h ‘Soprano off key Page %d line %d Data 1 = + 400 Data 2=+ 4.

```
| MFG/ADV CLR ALT DATA |
| GENERATE? |
```

19.1.6.4 Generate Four Packed Statu Messages

```
| FOUR PACKED ERRORS |
| GENERATE? |
```

As well as sending four packed Status Messages this item also demonstrates the text substitution feature.

19.1.6.5 Que/Status Message Counters

```
| GET-QM GET-SM MSGCNT |
| 0 44 0 |
```

Get -SM counts how many Status Messages have been sent. The Get-QM count and the MSGCNT are for Queued Messages which this version of the code does not support.

19.1.6.4 Back to Summary Menu

19.1.7 Responder Challenges

```
| RESPONDING |
| CHALLENGES? |
```

A Challenge is modification to the ‘normal’ and ‘proper’ protocol syntaxes. These are errors that may cause a message to be NACKED, dropped as a time out. What they should not do is cause the controller to crash, or lock out further properly formatted requests and replies. The first rule is do not crash. Many of these features are self-obvious and need little added explanation.

19.1.7.1 Respond to Broadcast

```
| RESPOND TO BROADCAST |
| YES >NO< |
```

19.1.7.2 Corrupt Checksum

```
| CORRUPT CHECKSUM |
```

```
| 100% 50% 6% >OFF<|
```

#### 19.1.7.3 Drop Request

```
| DROP REQUEST |  
| 100% 50% 6% >OFF<|
```

#### 19.1.7.4 Drop Response

```
| DROP RESPONSE |  
| 100% 50% 6% >OFF<|
```

#### 19.1.7.5 Noise Before Break

```
| NOISE BEFORE BREAK |  
| YES >NO< |
```

The responder sends 3 framed FFh bytes before sending the break. It shouldn't break anything but it may with some controllers

#### 19.1.7.6 Back to Summary

```
| CHALLENGES |  
| BACK TO SUMMARY? |
```

#### 19.1.7.7 Clear Challenges

```
| CHALLENGES |  
| CLEAR CHALLENGES? |
```

**Loop to top of the challenges menu**

#### 19.1.8 Discovery Challenges

```
| RESPONDING |  
| DISC. CHALLENGES? |
```

#### 19.1.8.1 Ignore Mute/unmute

```
| IGNORE MUTE/UNMUTE |  
| YES >NO< |
```

#### 19.1.8.2 Ignore DUB Range

```
| IGNORE DUB RANGE |  
| YES >NO< |
```

#### 19.1.8.3 Bad EUID Preamble

```
| BAD EUID PREAMBLE |  
| YES >NO< |
```

#### 19.1.8.4 No EUID Separator

```
| NO EUID SEPARATOR |  
| YES >NO< |
```

#### 19.1.8.5 Noise in DUB Response

```
| NOISE IN DUB RESP |  
| BEFORE&AFTER >OFF< |
```

#### 19.1.8.6 Phantom DUB Response

```
| PHANTON DUB RESPONSE |  
| YES >NO< |
```

#### 19.1.8.7 Invalid EUID Checksum

```
| INVALID EUID CHECKSUM |  
| 100% 50% 6% >OFF< |
```

#### 19.1.8.8 DUB Response Length

```
| DUB RESPONSE LENGTH |  
| 2 EXT-256 >NORMAL< |
```

#### 19.1.8.9 Drop DUB Requests

```
| DROP DUB REQUESTS |
| 100% 50% 6% >OFF<|
```

#### 19.1.8.10 Include Binding UID

```
| INCLUDE BINDING UID |
| YES >NO< |
```

#### 19.18.11 Back to Summary

```
| CHALLENGES |
| BACK TO SUMMARY? |
```

#### 19.1.7.12 Clear Challenges

```
| CHALLENGES |
| CLEAR CHALLENGES? |
```

### Loop to top of the discovery challenges menu

#### 19.1.8 Back to the Responder Menu

#### 19.2 Responder Setup

This menu allows the user to set the responder timing parameters . The default values loaded are shown in the following list.

##### 19.2.1 Set Responder Timing

```
Response delay          200µs
Break Length            200µs
MAB                     20µs
Interslot Time          20µs
Discovery Response Delay 200µs
Discovery Timeout       5600µs
Responder 485 pre-drive 4µ
Responder 486 post-drive 4µ
```

##### 19.2.2 Adjust Preamble length

```
| DUB PREAMBLE BYTES |
| 012345>7<9 0- 15 |
```

The user can set the number of FEh preamble bytes sent as part of discovery replies. It may be set from 0 to 9, or a random number from 0 to 15.

##### 19.2.3 Start with Responder Muted / Un-muted

```
| START WITH RESPONDER |
| >MUTED< UN-MUTED |
```

##### 19.2.4 NACK Invalid Disc CMD Class

```
| NACK INVALID DISC-CC |
| YES >NO< |
```

A strict reading of the standard says your not allowed to do this. .However, wouldn't it be better . . .

##### 19.2.5 NACK invalid Command Class

##### 19.2.6 Line termination

```
| LINE TERMINATION |
| TERM&BIAS>PASSTHRU<|
```

This is another place to control the line termination. On entering this item, what the user will see is determined by the answer that was made to the 'At End of Line' question. Normally a responder should either pass the signal thru or terminate it. The option to add bias will seldom be needed but if RDM .....

Electrically selecting adding bias will also set the unit to pass thru.

##### 19.2.7 Scope Trigger

```
| SCOPE TRIG OUT >OFF<|
```

| RESP DUB BCST |

The trigger signal's output is on the female connector pins 4 and 5. The trigger is generated at the end of the selected event.

#### 19.2.8 Change Responder UID

Normally the DMXter Responder uses the DMXter's UID for the test responder. This item allow changing the UID. It will be reset to the normal UID upon leaving the Responder Menu.

#### 19.2.9 Back to Responder Menu

### Loop to top of Responder Menu

#### 19.3 Responder Supported Parameters List

In addition to the required parameters the following are supported by this version of the software.

STATUS_MESSAGES	EQU 0030h	G
STATUS_ID_DESCRIPTION	EQU 0031h	G
DEVICE_MODEL_DESCRIPTION	EQU 0080h	G
MANUFACTURER_LABEL	EQU 0081h	G
DEVICE_LABEL	EQU 0082h	G S
DMX_PERSONALITY	EQU 00E0h	G S
DMX_PERSONALITY_DESCRIPTION	EQU 00E1h	G
SENSOR_DEFINITION	EQU 0200h	G
SENSOR_VALUE	EQU 0201h	G S

file = C:\Users\RMG\Documents\wp6doc\dmx-man\v4.30\v4.30\V4.30-Release-Notes\_with\_responder.wpd