UM10536

DMX512/RDM getting started guide

Rev. 1 — 6 February 2012 User manual

Document information

Info	Content
Keywords	LPC1110FD20; LPC1111FDH20; LPC1111FHN33; LPC1112FD20; LPC1112FDH20; LPC1112FDH28; LPC1112FHN33; LPC1112FHI33; LPC1113FHN33; LPC1113FBD48; LPC1114FDH28; LPC1114FN28; LPC1114FHN33; LPC1114FHI33; LPC1114FBD48; LPC1115FBD48; LPC11U12FHN33; LPC11U12FBD48; LPC11U13FBD48; LPC11U14FHN33; LPC11U14FHI33; LPC11U14FBD48; LPC11U14FET48; LPC11xx, LPC11Uxx, ARM, Cortex M0, DMX, DMX512, RDM, USB, Architectural Lighting, Entertainment Lighting, USB - DMX interface
Abstract	This User Guide explains how to get started with the NXP DMX512 Master (USB - DMX interface) and DMX512 Slave (demo board for DMX fixture) and the NXP Windows .NET DMX512control application.

Revision history

Rev	Date	Description
1	20120206	Initial version.

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

This document explains how to use the DMX512/RDM demo boards described in AN11153[3] and AN11154[4] , together with the PC GUI from the DMX512/RDM development kit[5], to create an DMX512/RDM setup.

2. Required items

2.1 Required hardware

To get started the following hardware items are required:

1. NXP DMX512 Master (USB - DMX512 interface) demo board

Fig 1. NXP DMX512/RDM master demo board

2. USB cable to connect NXP DMX512 Master to a PC

Fig 2. USB cable for NXP DMX512 Master

3. NXP DMX512 Slave demo board or other DMX receiver (e.g. Lamp)

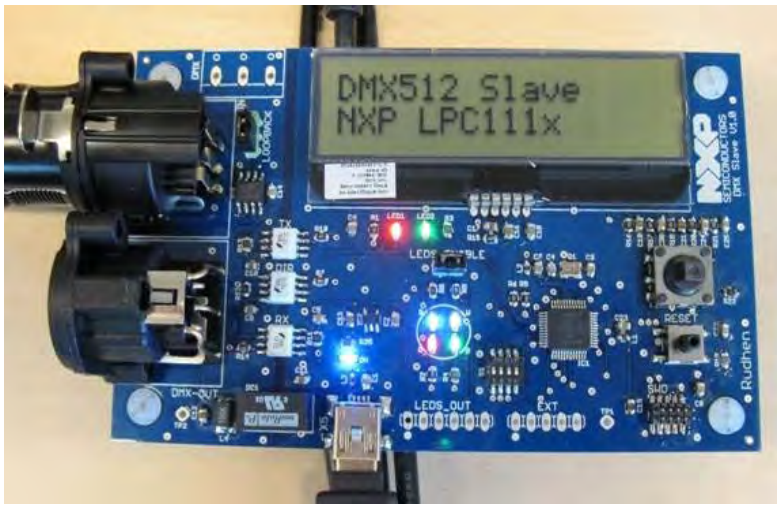

Fig 3. NXP DMX512 slave demo board

4. USB cable for NXP DMX512 Slave to connect to power adapter or Hub

Fig 4. USB cable for NXP DMX512 Slave

5. USB power adapter or USB Hub to connect to NXP DMX512 Slave

Fig 5. USB power adapter for NXP DMX512 Slave

6. DMX512 cable to connect NXP DMX512 Master to NXP DMX512 Slave

Fig 6. DMX512 cable

7. Windows7 or Windows-XP PC

Minimum requirements for concurrent DMX512 controlling and monitoring from one PC:

- Intel Core Duo CPU (e.g. E8400) @ 3.00 GHz
- 4 GB RAM
- .NET framework 4 (Windows 7 normally comes with .NET 4)

Optional hardware (only needed for concurrent DMX512 line monitoring):

8. Second NXP DMX512 Master to be used as DMX512 monitoring device (see Fig 1)
9. Second USB cable to connect NXP DMX512 Master to PC (see Fig 2)
10. Second DMX512 cable to connect DMX512 Slave to second DMX512 Master (see Fig 6)

2.2 Required software

To get started the following software items are required:

1. Windows7 or Windows-XP (Service Pack 3)
2. Microsoft .NET Framework 4 (Windows 7 normally comes with .NET 4)
3. DMX512control Windows .NET application (can be downloaded from the NXP website)

Optional software (only needed for software development):

4. Microsoft Visual Studio C# Express 2010
5. LPCXpresso v4.1 or IAR Embedded Workbench for ARM v6.20

The NXP DMX512 Master and DMX512 Slave demo boards already have DMX512 firmware stored in flash memory of the LPC11Uxx and LPC11xx microcontrollers respectively.

3. Installation and setup

3.1 Microsoft .NET Framework 4

The DMX512 demonstration requires a PC running Microsoft Windows7 or Windows-XP, and an installation of Microsoft .NET Framework 4.

The first step is to download Microsoft .NET Framework 4 from the link below, if it is not already installed on the demonstration PC.

Microsoft .NET Framework 4 Client Profile:

http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=17113

3.2 DMX512_RDM SDK 1.0

Download and install the NXP DMX512_RDM SDK 1.0 from the NXP website.

3.3 DMX512 network

Fig 7 shows the setup of a (simple) DMX512 network being used as a demonstration setup.

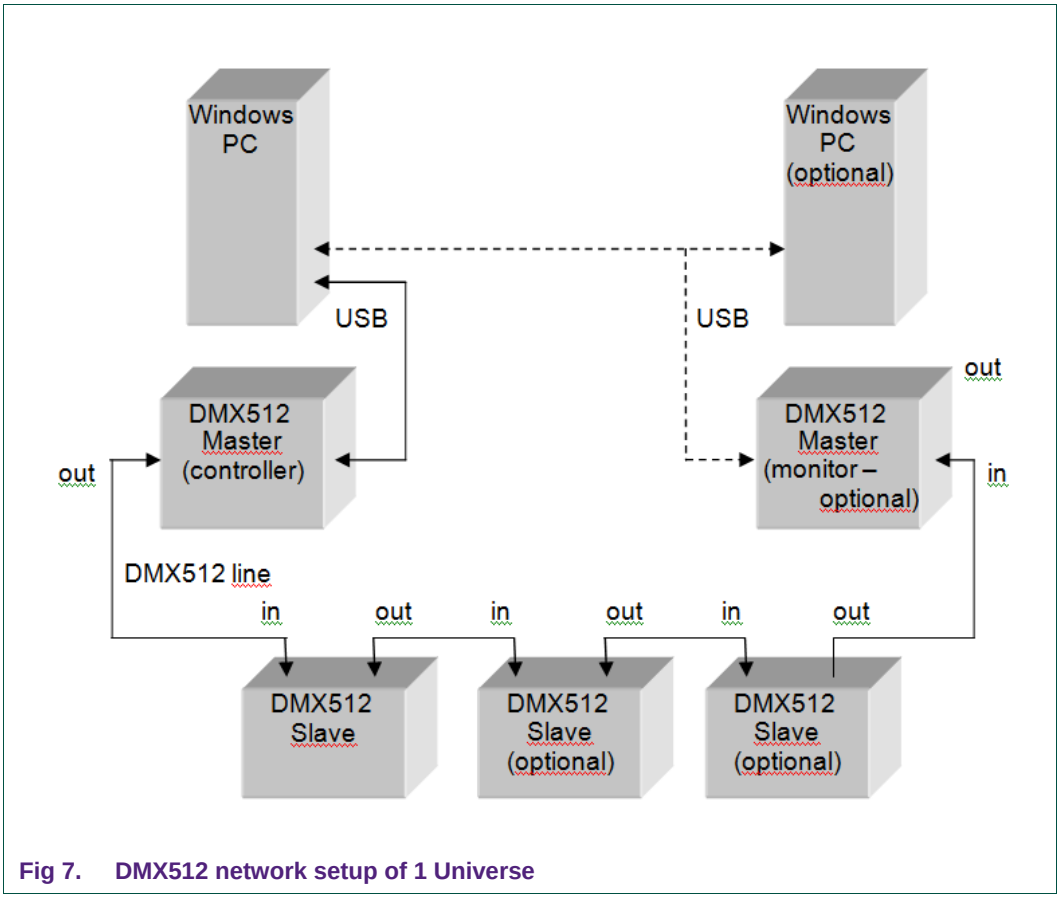

Fig 7. DMX512 network setup of 1 Universe

Fig 7 shows a DMX512 network setup of 1 Universe: one DMX512 Master (the controller) is depicted at one end of the DMX512 line, and another DMX512 Master (optional), being used as DMX512 line monitoring device, is depicted at the other end of the DMX512 line. Both masters can be placed anywhere in the DMX512 line as long as the DMX512 line is properly terminated (see clause 4.9 of [1]).

The DMX512 Master being used for monitoring should be connected to a different PC, to not disturb the PC connected to the first DMX512 Master controlling the DMX512 network. Also, the performance of the Windows PC can make it impossible to connect the monitoring DMX512 Master to the same PC as the controlling DMX512 Master.

3.4 Connecting a DMX512 Master to PC

Connect a USB cable to both the DMX512 Master and the PC. A driver must be installed when the DMX512 Master is connected to a PC for the first time. Windows will show the “Found New Hardware” window. The software tree of the DMX512 Master source code holds the driver files for Windows7 (lpc11xx-vcom_win7_64bit.inf) and Windows-XP (lpc11xx-vcom.win32.inf) as shown in Fig 8.

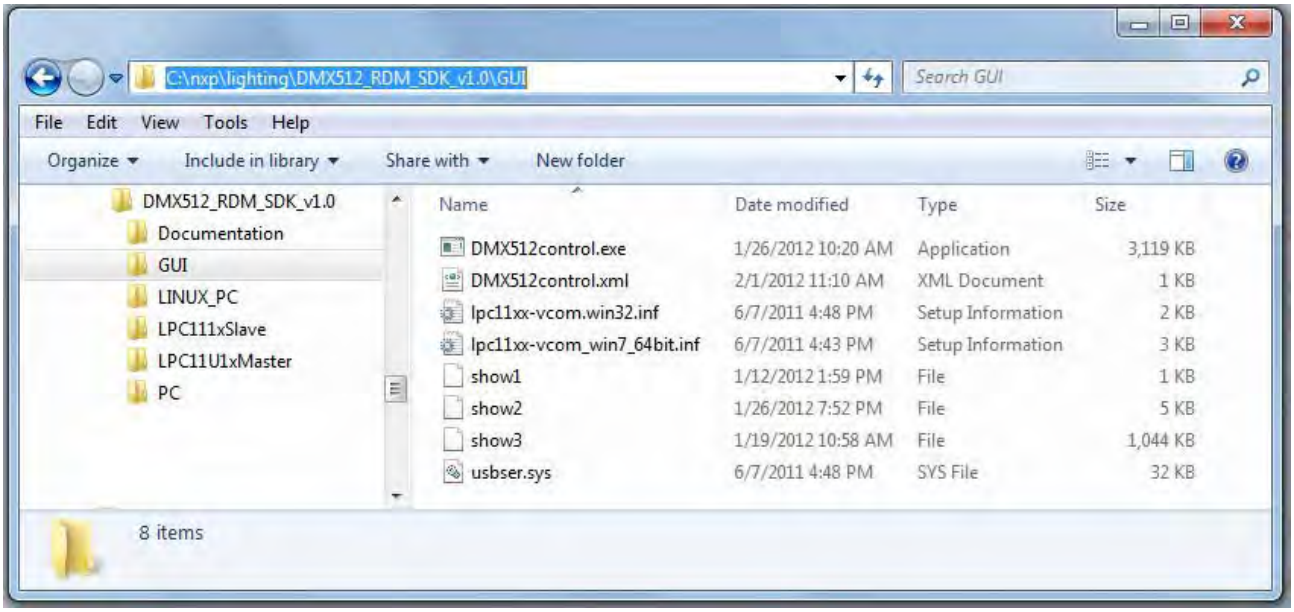

Fig 8. Software tree of DMX512 Master

From the “Found New Hardware” window browse to the directory which holds these files.

After installing the driver make sure to power cycle (disconnect and connect) the DMX512 master. Now the red LED1 of the DMX512 master must blink with a 1 Hz frequency, and the Device Manager of the PC must show the “LPC11xx USB VCom Port”, otherwise the device driver is not properly installed.

4. The DMX512control program

4.1 Startup

Double click the program file ‘DMX512control.exe’ to startup the demonstration program (Fig 9). The “Start” tab shows a picture of the NXP DMX512 Master demo board (left) and the NXP DMX512 Slave demo board (right).

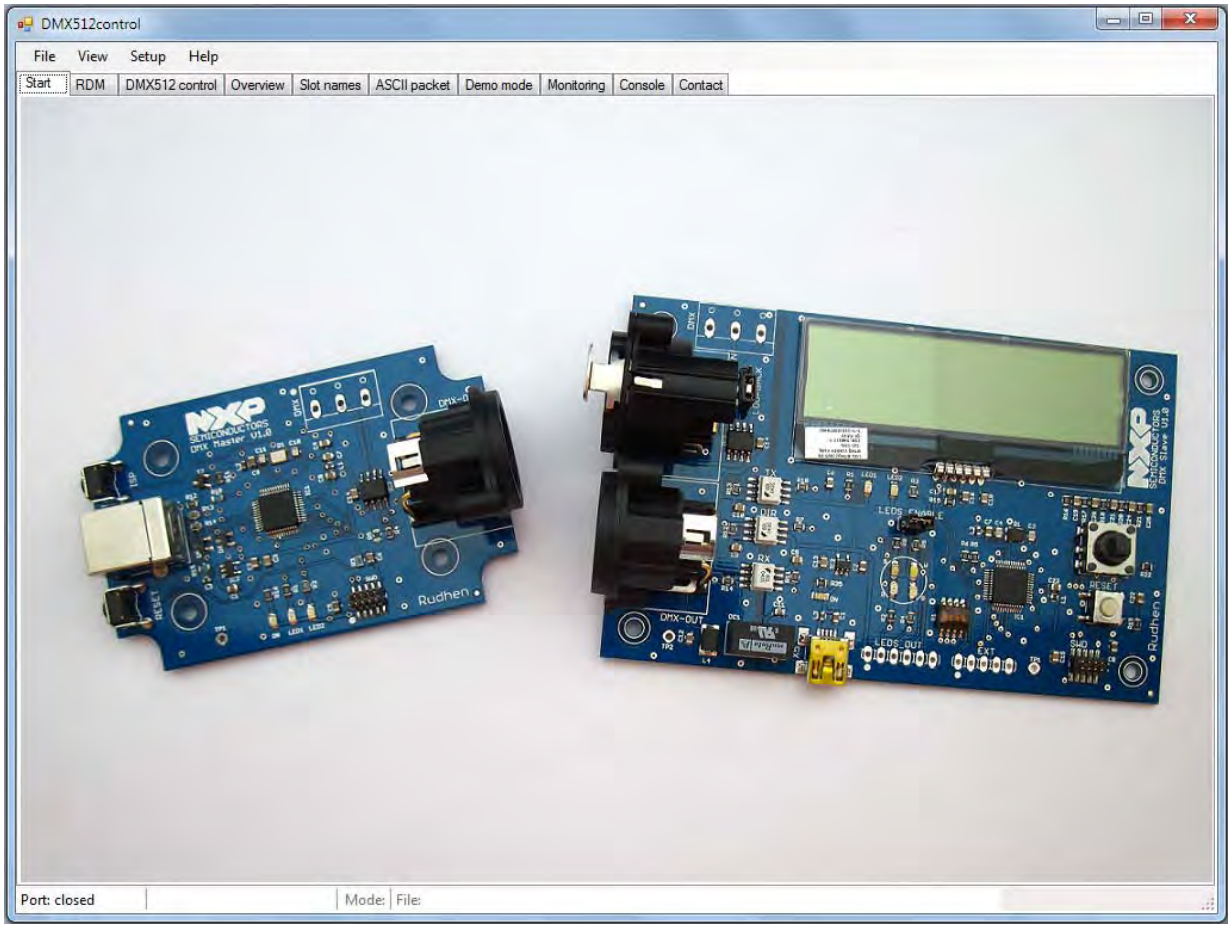

Fig 9. Startup screen of DMX512control program

The “About” window of the DMX512control program, available via the “Help” menu, shows the version number of the program described in this document (Fig 10).

Fig 10. About box of DMX512control program

The status bar at the bottom of the window shows the serial port is still closed. To open the serial port (virtual com port) select “Open com port…” from the “Setup” menu.

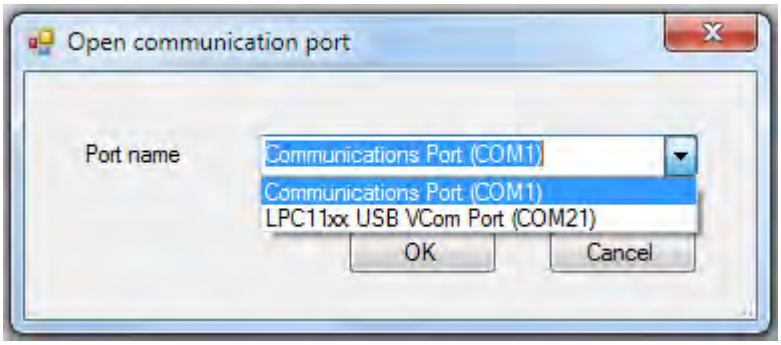

Fig 11. Open communication port window

From this “Open communication port” window select the port with name “LPC11xx USB VCom Port” and click on the “OK” button; the status bar will appear as shown in Fig 12.

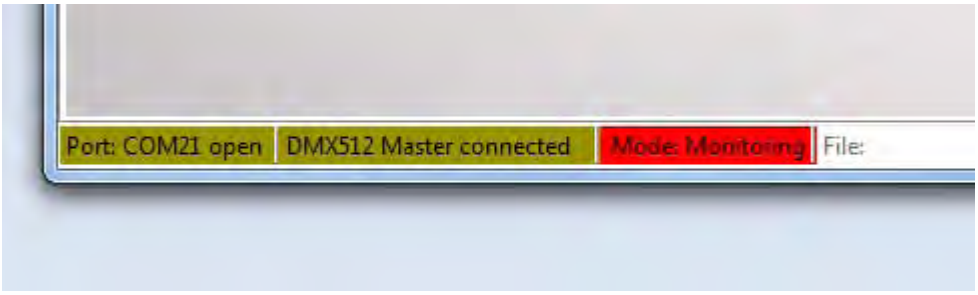

Fig 12. Status bar after Open com port

The selected COM port will be stored in the XML file '**DMX512control.xml**' that is in the same directory as the program file '**DMX512control.exe**'. The next time the .NET GUI is started, it will try to open the COM port stored in the XML file.

The "Mode:" field in the status bar turns red and shows the text "Monitoring", which is the normal startup behavior. When the communication port is opened, the DMX512 Master is put in monitoring mode, and the GUI can receive DMX512 line data as monitored by the DMX512 Master. This will only happen when there is another DMX512 Master connected to the same DMX512 line, and actively putting data on the DMX512 line.

Warning: There should only be one DMX512 Master in controller mode on the same DMX512 line (Universe)!!!

To put the DMX512 Master in controlling mode, deselect "Monitoring mode" from the "Setup" menu. Fig 13 shows the status bar after switching the DMX512 Master to Controlling mode.

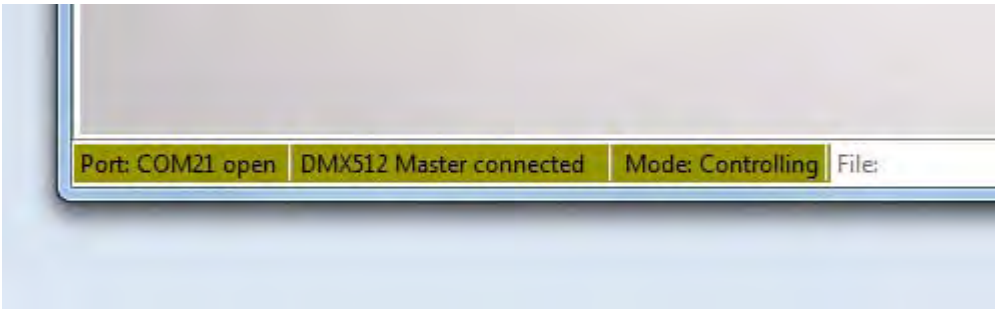

Fig 13. Status bar after switching to Controlling mode

Now the DMX512 Master can be used to control the DMX512 devices that are connected to this DMX Universe.

4.2 Remote Device Management

Remote Device Management (RDM) is supported by the NXP DMX512 Master and Slave demo boards, and is also supported by the .NET GUI. Via the tab “RDM”, press the “Discover” button to find connected DMX devices that support RDM.

Checking the “Retries” checkbox will enable the discovery retry mechanism; this allows more reliable discovery results (depending on the network), but will also result in a more time consuming discovery process.

The “Clear” button can be used to clear the list, which is only necessary to start a full discovery. The total available (discovered) RDM enabled DMX devices is shown by the enumeration below the RDM device list on the “RDM” tab. The RDM device list shows the UID or Unique ID of the RDM device, the DMX start address of the first channel (slot) of the RDM device, the footprint showing how many channels are used by the RDM device, and additional info such as software ID and label.

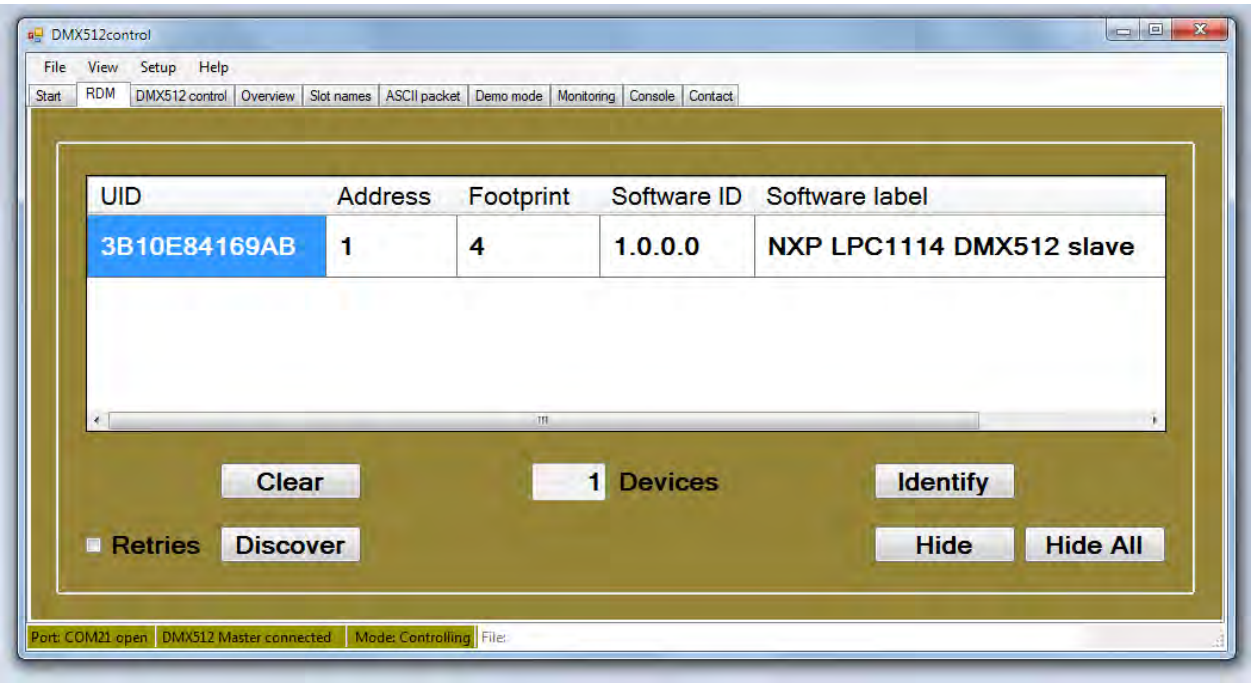

Fig 14. RDM tab in DMX512control program

The first four characters of the UID give the manufacturer ID of an RDM device, which is 3B10 for the NXP RDM enabled DMX devices (demo boards). The advantage of RDM is that the DMX start addresses of RDM enabled devices can be changed from the PC, taking away the dangerous job of changing these addresses by using ladders (a real life saver). The DMX start address of an RDM device can be changed by selecting the appropriate Address field in the RDM device list and assigning it a new value (Fig 15).

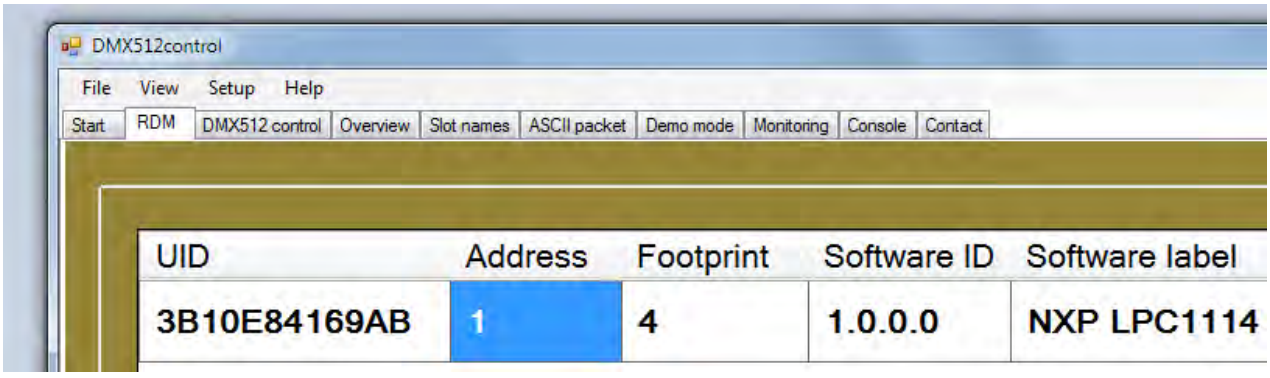

Fig 15. Changing the DMX start address

The Address field is the only editable field in the RDM device list. RDM is also used to get info on DMX device status, such as Lamp hours (burning hours). To easily find where in the DMX network this lamp is situated, RDM supports the Identify message which can be sent to a specific RDM device by selecting this device in the RDM device list and pressing the “Identify” button. For the NXP DMX512 Slave demo board, this will light up the RGBW LEDs. Pressing the “Hide” button will undo the Identify action.

4.3 DMX512 control

The “DMX512 control” tab can be used to change the data values that are sent to the DMX device channels, also called Slots. Twelve sliders are available to easily control 12 independent channels/slots, which can be of different DMX devices (Fig 16). When the thumb of a slider is moved, the accompanying value field will be updated immediately.

When a slot value is changed, the data values of all slots (restricted by the Nr of Slots setting) are downloaded to the DMX512 Master via the USB connection and repeatedly broadcast by the DMX512 Master according to the refresh rate. When the thumb of a slider is moved, many downloads take place. The data value for a slot can also be changed by entering a new value in the value field; in this case the thumb of the accompanying slider will move to the correct position. When a slot is selected by either its radio button, slider, value field or slot field, the slot name field will be updated with the name as assigned via the “Slot names” tab (Fig 18). Slot names are stored in the XML file ‘**DMX512control.xml**’ as shown:

```
<?xml version="1.0"?>
<DMX512control>
  <SERIAL_PORT>
    <PortName>COM21</PortName>
  </SERIAL_PORT>
  <SLOT_NAMES>
    <SLOT nr="1" name="slot 1" />
    <SLOT nr="2" name="2" />
  </SLOT_NAMES>
</DMX512control>
```

By editing the slot field, another slot can be chosen to be controlled by the accompanying slider. The slot field in the left corner has a different behavior (indicated by the slightly different background color); editing this field (not using the up/down spin buttons) will change all other slot fields with up running slot numbers, making it easier to select 12 adjacent slots.

The number of slots that will be addressed by the DMX512 Master can be changed between 24 and 512 by editing the “Nr of slots” field in the upper left corner of the “DMX512 control” tab.

The refresh rate is the frequency in which the DMX512 Master will put the slot values on the DMX512 line. This refresh rate can be changed between 1 Hz and 40 Hz by editing this field on the “DMX512 control” tab. By checking the “auto” checkbox, the DMX512 Master will refresh the slot values at least once per second, or as soon as the PC supplies new values. The resulting frequency as measured by the DMX512 Master will be shown in the “refresh rate” field. For 512 slots, the fastest refresh rate is 44 Hz; for 256 slots, the fastest refresh rate is 88 Hz, etc.

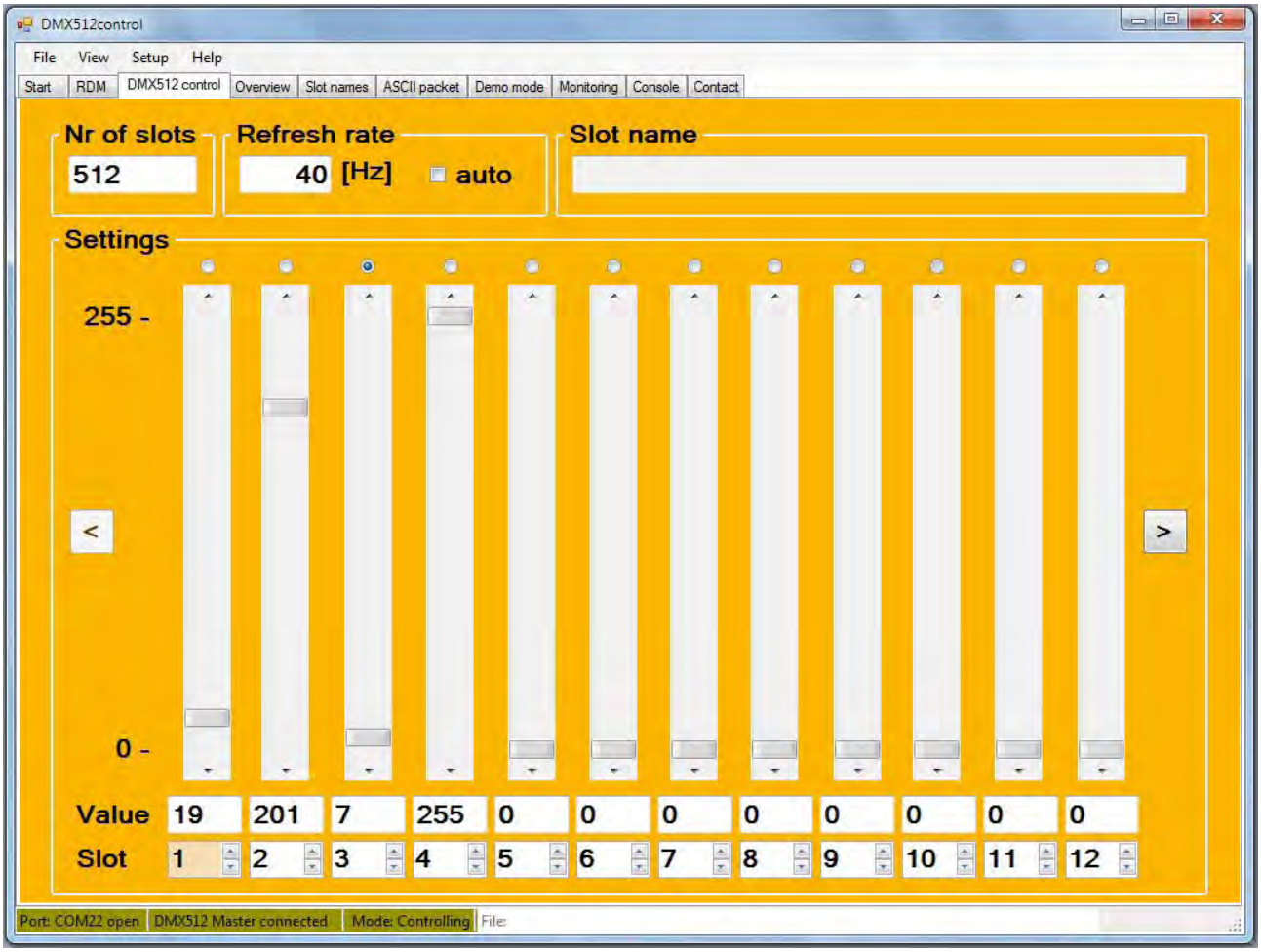

Fig 16. DMX512 control sliders

Via the “Overview” tab the actual slot values of all 512 slots can be viewed (Fig 17). During demo mode (Section 4.4), the slot values in the Overview are not updated; they are updated when the demo mode is stopped.

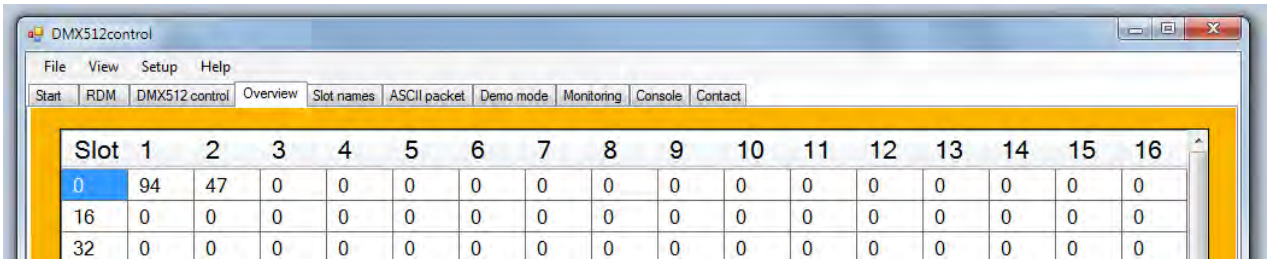

Fig 17. The overview tab

Fig 18 shows part of the “Slot names” tab.

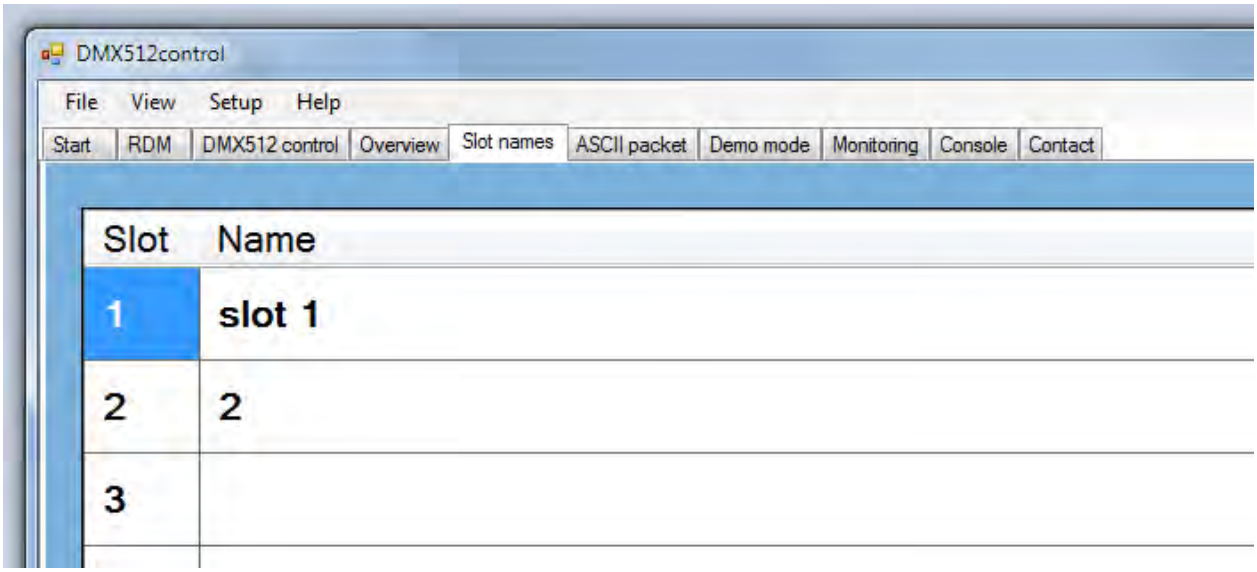

Fig 18. Assigning slot names

Via the DMX512 line, different kinds of packets can be sent, such as an ASCII Text Packet (see Annex D of [1]). Since the NXP DMX512 Slave is equipped with a 32 character LCD, these devices support the handling of an ASCII Text Packet, and the .NET GUI supports the broadcast of such a Text Packet via the “ASCII packet” tab.

The upper field shows the last broadcast message, while the lower text box can be used to input a new message that will be broadcast on pressing the Enter key on the keyboard of the PC.

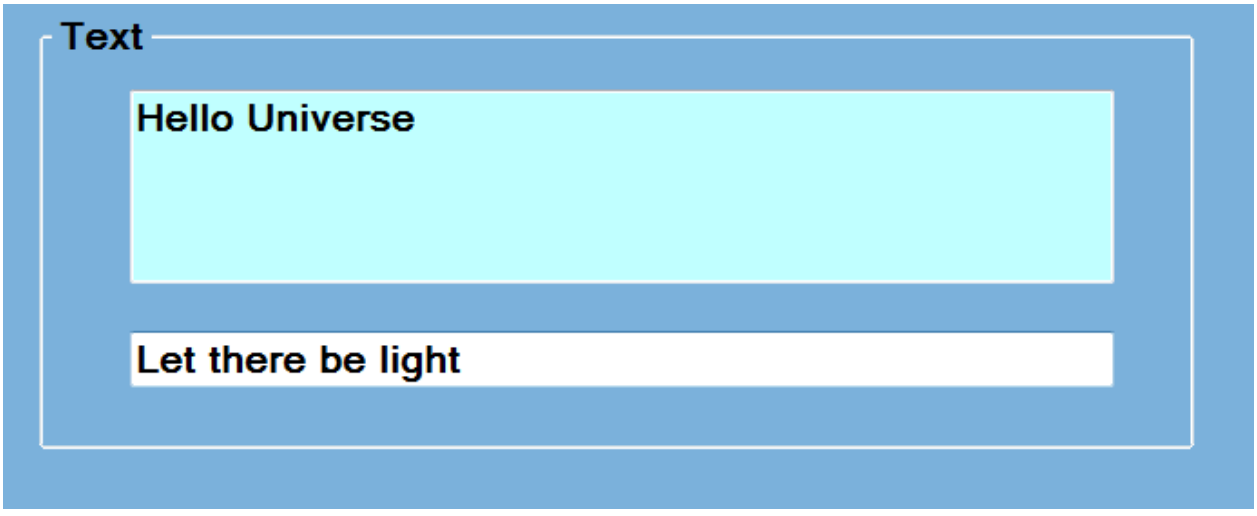

Fig 19. Sending ASCII text packets

4.4 Demo mode

The .NET GUI offers five automated demos via the “Demo mode” tab. On this tab, four channels can be chosen. Demo 1 uses the first three channels, demo 2 and 4 use all four channels, and demo 3 uses all the DMX slots from 1 till the slot number set in the “Nr of slots” field available on the “DMX512 control” tab. These channel selectors can be set to zero (meaning, skip this channel), but for demo 1 to work none of the first three channels can have the value 0, and for demo 4 to work none of the four channels can have the value 0.

The “value changes” slider sets the time interval of a software timer in the GUI that (after expiration of such a time interval) triggers an event that handles changing the slot values as required for that demo. The refresh rate on the DMX512 line is a result of the setting of the “Refresh rate” field on the “DMX512 control” tab, or by the position of the “value changes” slider when the “auto” checkbox is checked. The demos 1 through 4, on each timer event, download the data values of slots 1 until “number of slots” to the DMX512 master. Demo 5 always downloads all the 513 slot values.

Demo 1 changes the data values belonging to the first three channels according the below depicted time schedule. Channel 1 is the Red line, channel 2 is the Green line, and channel 3 is the Blue line. The increment/decrement of the data value of a channel is given by “value increment” selection (1, or 5, or 10, or 25, or 50). The minimal data value is 0, and the maximum data value is 250. This schedule repeats all the time (giving a pleasant color pattern) until the demo is stopped.

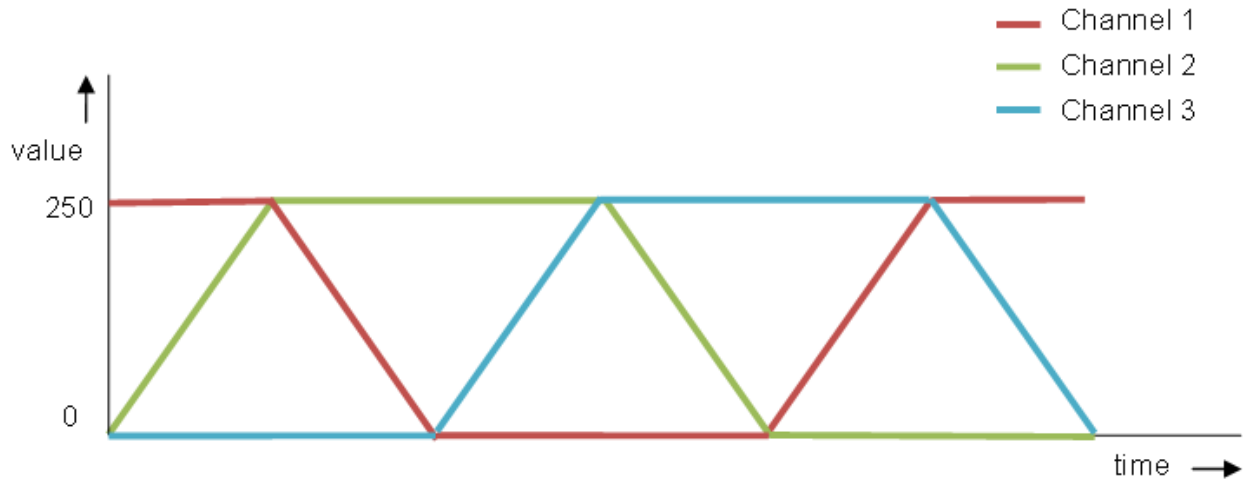

Fig 20. Demo 1 time schedule

Demo 2, for all four channels independently, increments the slot data value with 1 on every timer event until the data value reaches 255, and then decrements the data value with 1 on every timer event till it reaches zero, and then starts incrementing again (and so on). When this demo is started, the data values of the four channels are not reset to zero, so the increment starts from the current data value. Different color schedules can be created by using different start values for the selected channels.

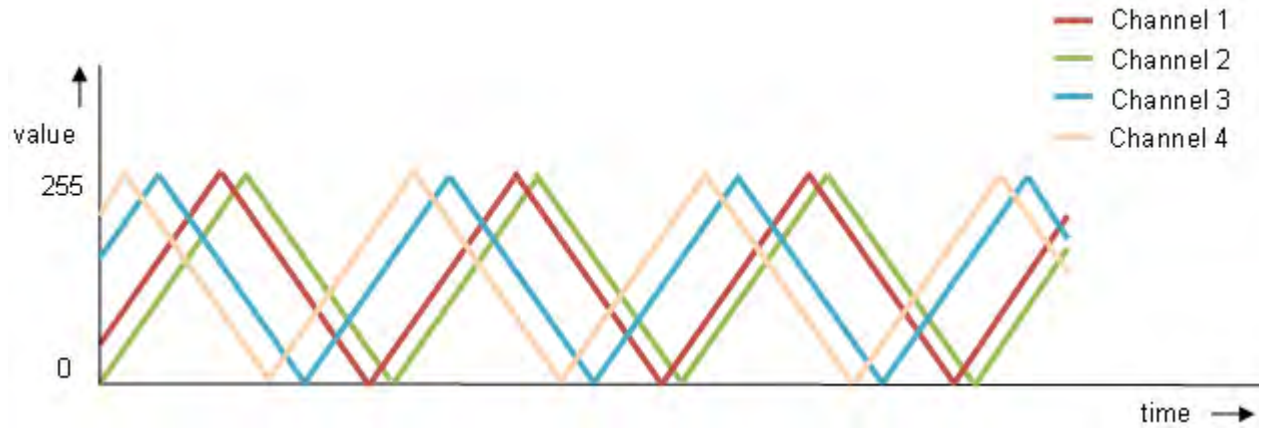

Fig 21. Demo 2 time schedule

Demo 3 just increments the data values of the slots 1 until “number of slots” with 1 on every timer event, wrapping around when data value 255 is reached. This can be a good mode to observe how frequent the DMX512 line carries new data, and whether this data is correct since all slot values should be the same.

Demo 4 uses the four selected channels from the “channel selection” group box, and assigns the respective data values from t1 (time 1) from the “channel values” group box on the first timer event. On the second timer event the data values from t2 are assigned to the selected channels. The fifth timer event will use the data values from t1 again, and so on.

Demo 5 can be used for two demos. The first one (default) plays a lightshow from a file placed in the same directory as the DMX512control application; the second one broadcasts the text that is entered in the input field (not starting with file://) next to the “demo 5” radio button, shifting the characters 1 position to the left (in a 32 characters wide text string) on every time event. This results in a scrolling text on the LCD of the DMX512 Slave. Dragging the thumb of the “value changes” slider will result in a slower/faster scrolling text on the LCD. Selecting the “ASCII packet” tab shows this in the .NET GUI.

4.5 Monitoring

A second DMX512 Master can be used to monitor the DMX512 line (Fig 7). Just start a second DMX512control GUI (preferably on a second PC because of performance issues, unless the PC fulfills the minimum requirements) and select the “DMX512 control” tab. While moving the thumb of a slider in the controlling GUI, observe that the slider of the same slot in the monitoring (second) GUI follows all the movements.

Also observe that changing the “Nr of slots” or the “Refresh rate” in the controlling GUI will be shown in the monitoring GUI. The refresh rate is measured by the second DMX512 Master over a four second time period and passed over to the monitoring GUI.

In the monitoring GUI, select the “Overview” tab and observe that the data value of a slot changes when the corresponding slot slider in the controlling GUI is moved.

The DMX512control GUI can record DMX packets, as put on the DMX512 line by the controlling DMX512 Master, or monitored on the DMX512 line by the monitoring DMX512

Master, in a file on the PC. This is accomplished by creating or opening a file via the “File” menu.

Warning: Care must be taken that the recording file is closed in time since this file can grow big very quickly!!!

In the monitoring GUI open a new file via the “File” menu. In the controlling GUI select the “RDM” tab and press the “Clear” button. When the “Discover” button is pressed, all (discovery) packets that are monitored on the DMX512 line are recorded to the file ‘untitled’ placed in the same directory as the program file ‘DMX512control.exe’. The status bar of the monitoring GUI will appear as shown in Fig 22.

Fig 22. Recording to file by monitoring GUI

Close the ‘untitled’ file via the “File” menu. The following discovery packets can be found in the recording file (with other UIDs and depending on the available RDM enabled devices).

```
12h5m43s632ms : Startcode=CC

01 18 FF FF FF FF FF FF 3B 10 10 9C 8E FE 00 01 00 00 00 10 00 03 00 09 76

12h5m43s648ms : Startcode=CC

01 24 FF FF FF FF FF FF 3B 10 10 9C 8E FE 01 01 00 00 00 10 00 01 0C 00 00 00 00 00
00 FF FF FF

FF FF FF 0F 87 FE FE FE FE FE FE FE AA BB 7F BA 55 EA FD EB 55 EB 7D AB FF AA 5D AA
D7

12h5m43s648ms : Startcode=CC

01 18 3B 10 E8 41 69 AB 3B 10 10 9C 8E FE 02 01 00 00 00 10 00 02 00 06 05

12h5m43s648ms : Startcode=CC

01 1A 3B 10 10 9C 8E FE 3B 10 E8 41 69 AB 02 00 00 00 00 11 00 02 02 00 00 06 09

12h5m43s664ms : Startcode=CC

01 18 3B 10 E8 41 69 AB 3B 10 10 9C 8E FE 03 01 00 00 00 20 00 60 00 06 74

12h5m43s664ms : Startcode=CC

01 2B 3B 10 10 9C 8E FE 3B 10 E8 41 69 AB 03 00 00 00 00 21 00 60 13 01 00 00 01 01
01 01 00

00 00 00 04 01 01 00 01 00 00 00 06 A6

12h5m43s679ms : Startcode=CC

01 18 3B 10 E8 41 69 AB 3B 10 10 9C 8E FE 04 01 00 00 00 20 00 C0 00 06 D5

12h5m43s679ms : Startcode=CC

01 30 3B 10 10 9C 8E FE 3B 10 E8 41 69 AB 04 00 00 00 00 21 00 C0 18 4E 58 50 20 4C
50 43 31
```

```
31 31 34 20 44 4D 58 35 31 32 20 73 6C 61 76 65 0D 9D
```

As can be seen from the above recording file content, the timestamp 12h5m43s648ms appears in front of three successive RDM packets, which is a problem of the .NET environment. To overcome this, the DMX512 Master can generate a relative timestamp which only shows seconds and milliseconds since power-up of the DMX512 Master. This time stamping is enabled by checking the menu option “DMX512 Master timestamps” via the “Setup” menu. The result of this is shown in the following lines:

```
6713s734ms : Startcode=CC

01 18 FF FF FF FF FF FF 3B 10 10 9C 8E FE 05 01 00 00 00 10 00 03 00 09 7B

6713s746ms : Startcode=CC

01 24 FF FF FF FF FF FF 3B 10 10 9C 8E FE 06 01 00 00 00 10 00 01 0C 00 00 00 00 00
00 FF FF FF

FF FF FF 0F 8C FE FE FE FE FE FE FE AA BB 7F BA 55 EA FD EB 55 EB 7D AB FF AA 5D AA
D7

6713s749ms : Startcode=CC

01 18 3B 10 E8 41 69 AB 3B 10 10 9C 8E FE 07 01 00 00 00 10 00 02 00 06 0A

6713s753ms : Startcode=CC

01 1A 3B 10 10 9C 8E FE 3B 10 E8 41 69 AB 07 00 00 00 00 11 00 02 02 00 00 06 0E

6713s757ms : Startcode=CC

01 18 3B 10 E8 41 69 AB 3B 10 10 9C 8E FE 08 01 00 00 00 20 00 60 00 06 79

6713s761ms : Startcode=CC

01 2B 3B 10 10 9C 8E FE 3B 10 E8 41 69 AB 08 00 00 00 00 21 00 60 13 01 00 00 01 01
01 01 00

00 00 00 04 01 01 00 01 00 00 00 06 AB

6713s766ms : Startcode=CC

01 18 3B 10 E8 41 69 AB 3B 10 10 9C 8E FE 09 01 00 00 00 20 00 C0 00 06 DA

6713s769ms : Startcode=CC

01 30 3B 10 10 9C 8E FE 3B 10 E8 41 69 AB 09 00 00 00 00 21 00 C0 18 4E 58 50 20 4C
50 43 31

31 31 34 20 44 4D 58 35 31 32 20 73 6C 61 76 65 0D A2
```

Besides recording to file, the GUI also offers the “Monitoring” tab that can be used for fast viewing of captured DMX packets, for a maximum of 1000 packets. The “Start” button on this screen can be clicked at any time, and for each received packet the packets counter will be incremented. The captured packets will be shown after the “Stop” button is clicked.

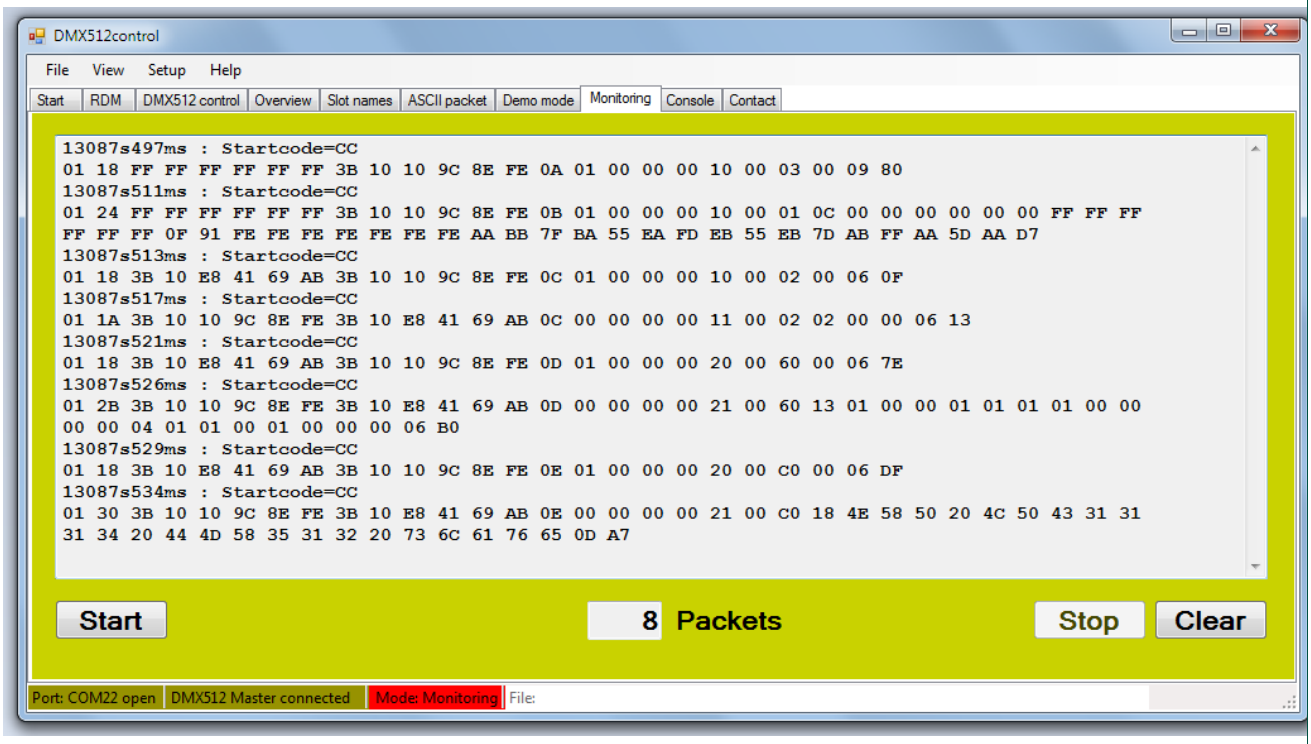

Fig 23. The monitoring tab

4.6 Console

The “Console” tab gives an example implementation of a command line interface to the DMX512 Master firmware. Only three commands are implemented, of which the **uptime** command can be used to check how long the DMX/RDM master is running.

5. Document management

5.1 Abbreviations

Table 1. Abbreviations

Acronym	Description
GUI	Graphical User Interface
Hz	Hertz
LCD	Liquid Crystal Display
LED	Light Emitting Diode
PC	Personal Computer
RDM	Remote Device Management
RGBW	Red Green Blue White
SDK	Software Development Kit
UID	Unique device ID
USB	Universal Serial Bus
XML	Extensible Markup Language

5.2 Referenced documents

Table 2. Referenced documents

Doc	Title	Version	Author	Issue Date
[1]	ANSI E1.11 - Asynchronous Serial Data Transmission Standard for Controlling Lighting Equipment and Accessories	2008	ESTA	20081204
[2]	ANSI E1.20 - Remote Device Management Over DMX512 Networks	2010	PLASA	20110104
[3]	AN11153 DMX512/RDM slave using LPC111x	V1.0	NXP Semiconductors	20120201
[4]	AN11154 DMX512/RDM master using LPC11U1x	V1.0	NXP Semiconductors	20120201
[5]	http://www.nxp.com/documents/other/DMX512_RDM_SDK1.0_Setup.exe			

6. Appendix A: ENTTEC RDM packet sniffer

The ENTTEC RDM USB PRO has been used to check whether the RDM messages generated by the NXP DMX512 Master and Slave demo boards could be recognized as valid RDM messages. The screen shots of discovery searches of different network setups with NXP RDM enabled DMX512 Slaves are shown in Fig 24 and Fig 25. These screenshots show the correctness of the captured RDM messages (RDM Collisions are normal during the discovery search).

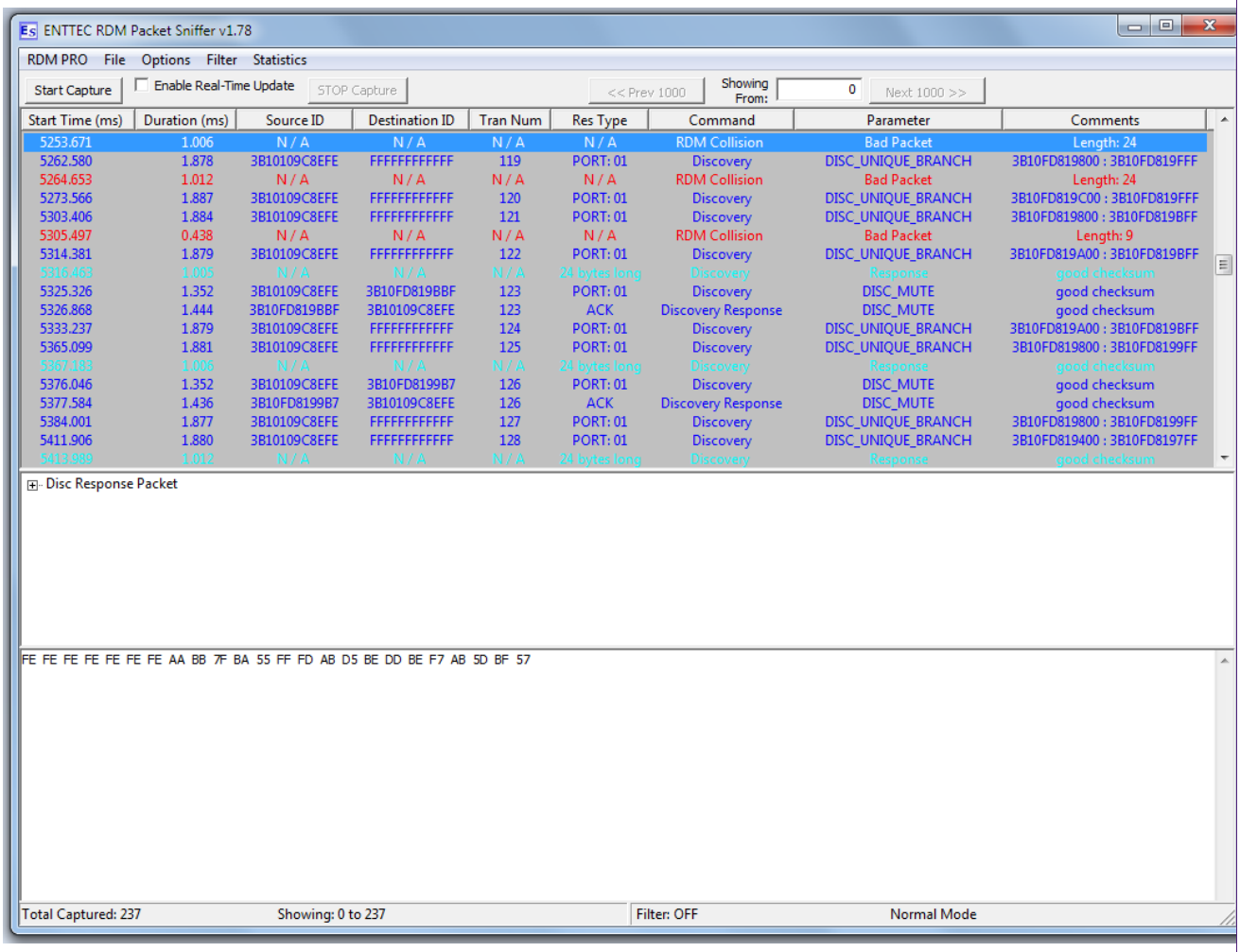

Fig 24. ENTTEC RDM message search (1 of 2)

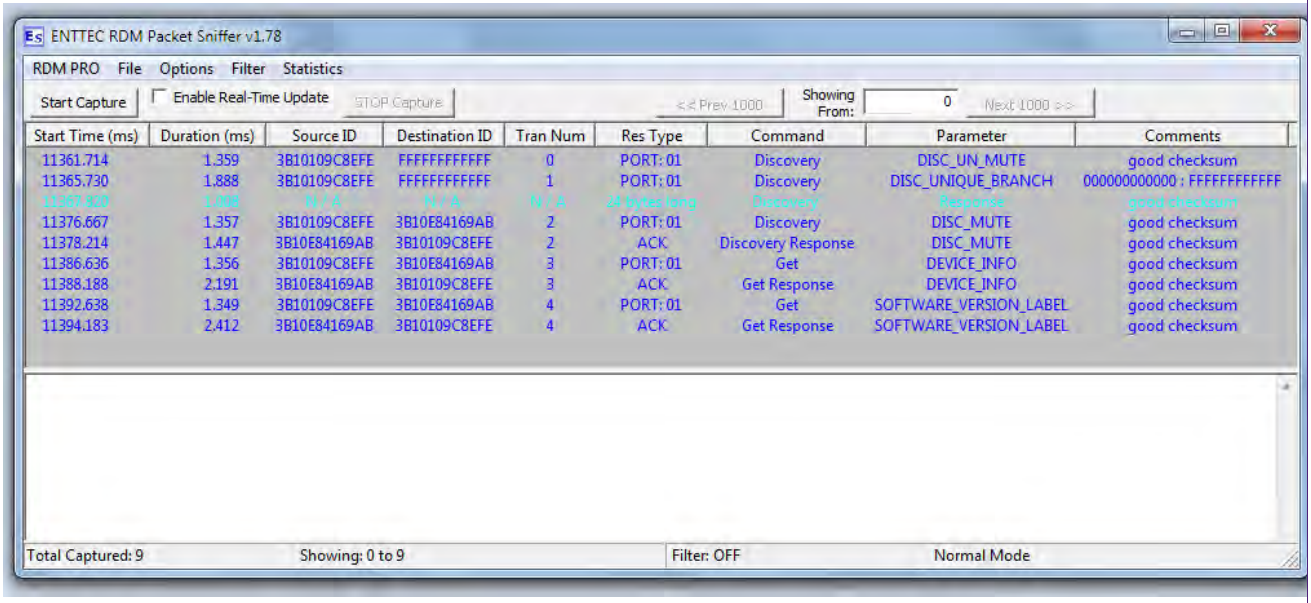

Fig 25. ENTTEC RDM message search (2 of 2)

7. Appendix B: Testing with commercial lights

Fig 26 shows the result of a discovery search on a DMX network that has a Philips LEDline2 BCS719 high-performance linear grazing light (intended for Architectural Lighting). This commercial fixture is RDM enabled and shows up neatly (found first, because of the higher UID value) in the RDM device list.

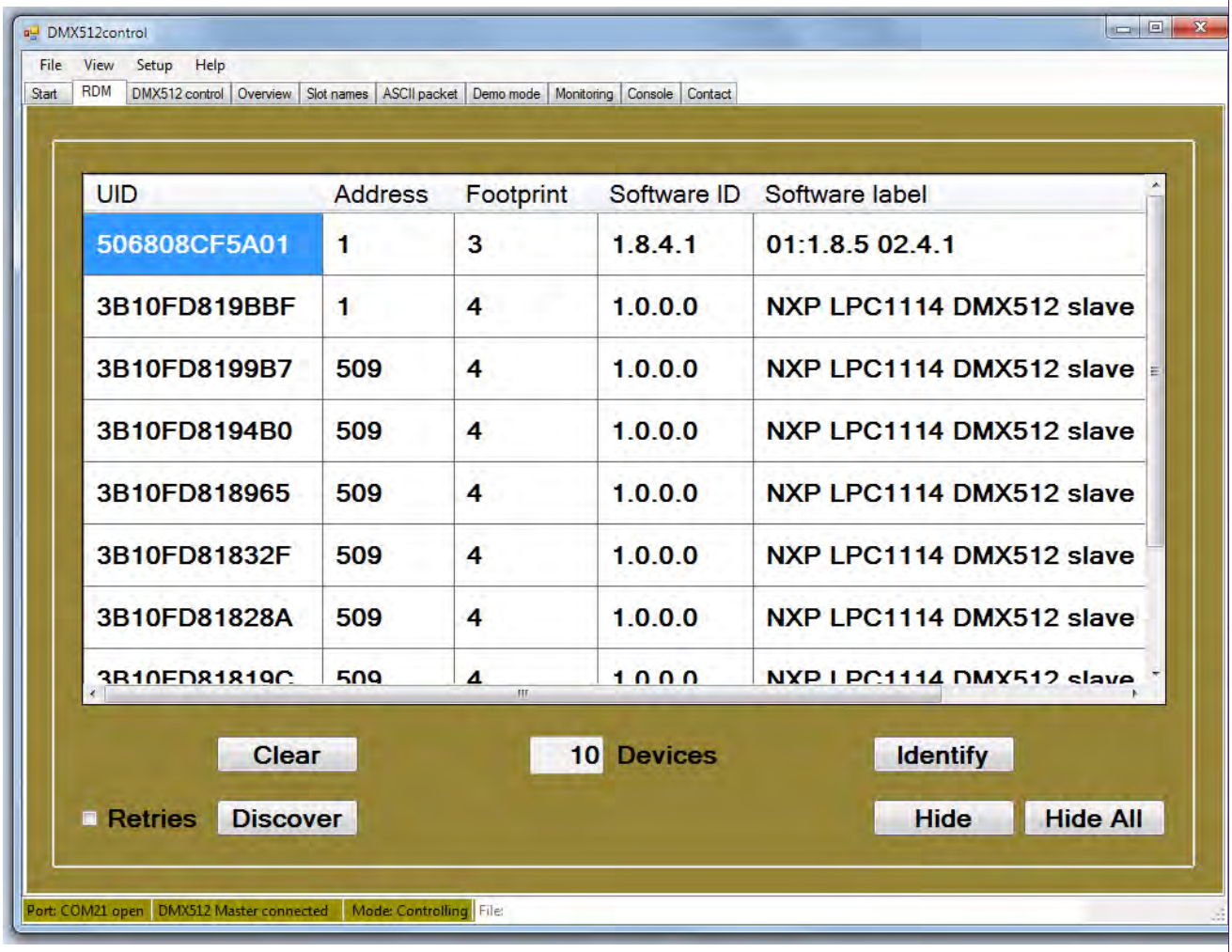

Fig 26. Discovery search on DMX network

8. Legal information

8.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

8.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

8.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

9. List of figures

10. List of tables

11. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 6 February 2012
Document identifier: UM10536