0011-01-16-03-000
ZM357S-USB(-LR) Modules

Document No:
0011-01-16-03-000 (Issue B)

INTRODUCTION

The MeshConnect™ ZM357S-USB(-LR) Sticks, from California Eastern Laboratories (CEL) are designed to be used along with the Silicon Labs EM35X-DEV(-IAR) Development Kit and software tools. CEL has partnered with Silicon Labs and is using their Ember ZigBee PRO stack. For more information please see the Silicon Labs document *EM35x Development Kit User Guide*.

This Quick Start Guide is provided along with other supporting documentation on CEL's website as a means to verify the ZM357S-USB(-LR) Stick is functional in the target development environment (Windows/MAC/Linux). The following quick start functions are demonstrated:

1. **Hardware Programming** – A hardware programming example is provided and a sample application called Node Test is loaded on the ZM357S-USB(-LR) Stick.

2. **Node Test** - A sample application is loaded on the ZM357S-USB(-LR) Stick, known as Node Test, and a simple functional test using freely available terminal programs is demonstrated on Windows/MAC/Linux operating systems.

3. **Network Co-Processor** - A sample application is demonstrated using Cygwin tools on a Windows machine. This example is useful when the EM357 on the ZM357S-USB(-LR) Stick is not the host processor and you wish to use another processor in your system as the host.

4. **Sniffer Example** - An example is provided showing how to use the Silicon Labs Ember Desktop, along with a ZM357S-USB(-LR) Stick, to create a ZigBee Sniffer to aid in debugging and deploying complete ZigBee solutions.

To purchase a Silicon Labs Development Kit (EM357x-DEV or EM357x-DEV-IAR) please contact Silicon Labs at www.silabs.com. For more information on MeshConnect products, visit CEL at www.cel.com/MeshConnect.

TABLE OF CONTENTS

1 – Hardware Programming ZM357S-USB(-LR) Sticks

1.1 – USB Stick Hardware Overview

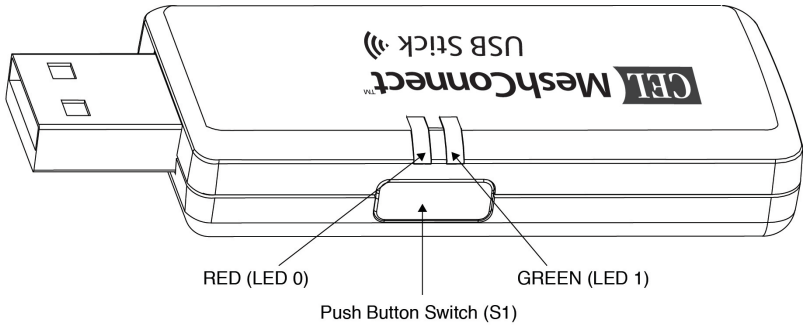

Figure 1. Profile of ZM357S-USB(-LR) showing push button switch and red and green LED

Hardware Interface	EM35x I/O
LED 0 (RED)	PA6
LED1 (GREEN)	PA7
Switch S1	PB6

Table 1. Hardware mapping to physical I/O on ZM357S-USB(-LR)

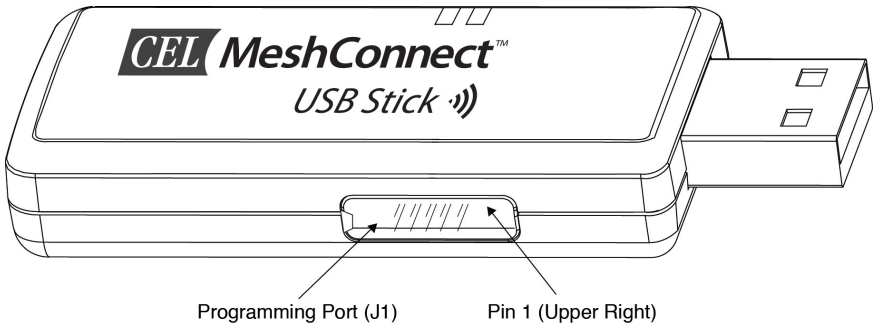

Figure 2. Profile of ZM357S-USB(-LR) showing programming port adapter and Pin 1

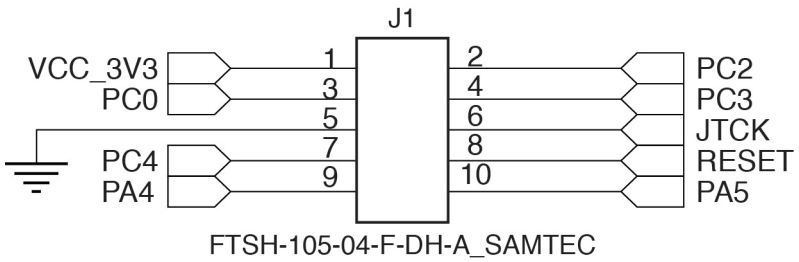

Figure 3. Shows programming pins for EM35x and internal connector connections

2 – EMBER ISA3 USED FOR PROGRAMMING THE ZM357S-USB(-LR) STICKS

Download the files using an Ember InSight Adapter (ISA3) connected via Ethernet and InSight Desktop (Ember Desktop). If you need help connecting to the ISA3 please reference the *Ember EM35xx Utilities Guide for the EM35x SoC Platform* that is provided with each EM35X-DEV Development Kit.

2.1 – Programming Using Ember InSight Desktop

The following steps presume you have Ember InSight Desktop connected and are able to see an ISA3 in your adapters view. The ISA3 InSight Port is connected to the programming port on the ZM357S-USB(-LR).

1. Right-click the adapter from the Adapters view and select Connect from the pop-up menu, as shown in Figure 5.

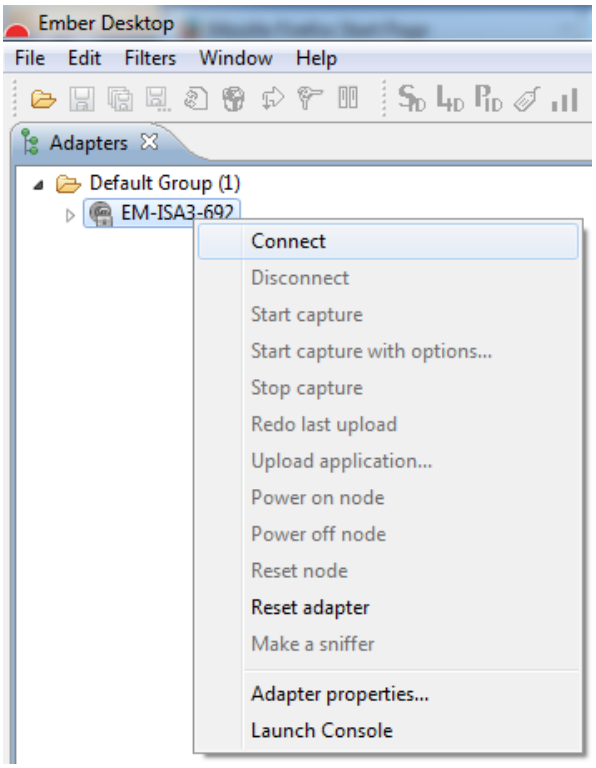

Figure 5. Connect Selection with Ember Desktop

2. Right-click the connected adapter from the Adapters view and select *Upload Application* from the pop-up menu, as shown in Figure 6.

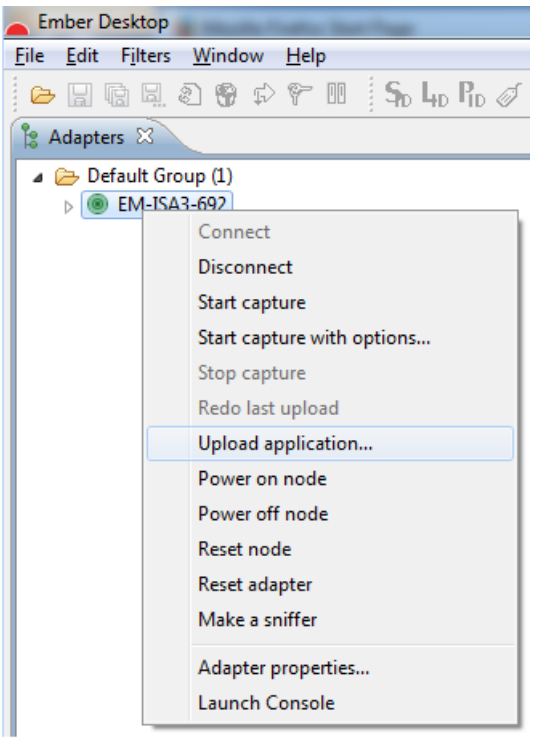

Figure 6. Upload Application Selection with Ember Desktop

3. Browse to select an application image from the dialog's list as shown in Figure 7.

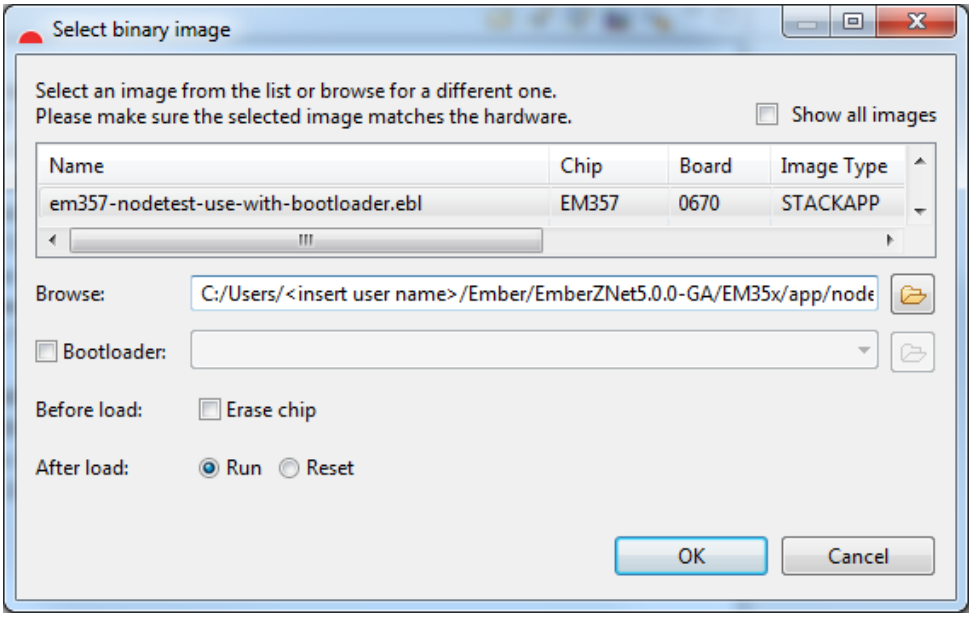

Figure 7. Select Binary Image and Bootloader with Ember Desktop

4. Your ZM357S-USB should have a bootloader installed so you may skip this option. If for some reason this is erased you can download the replacement bootloader; the link can be found in the Reference Section.

5. Select whether you would like the Target to Reset or Run after loading.

6. Click OK.

2.2 – Programming using EM35xx-Utilities

Alternatively you may use the EM3xx-Utilities provided by Silicon Labs. Figure 8 shows an example of connecting EM35xx-Utilities using a USB cable in a command prompt in Windows. Please reference *Ember EM35xx Utilities Guide for the EM35x SoC Platform* in the Reference Section for more information.

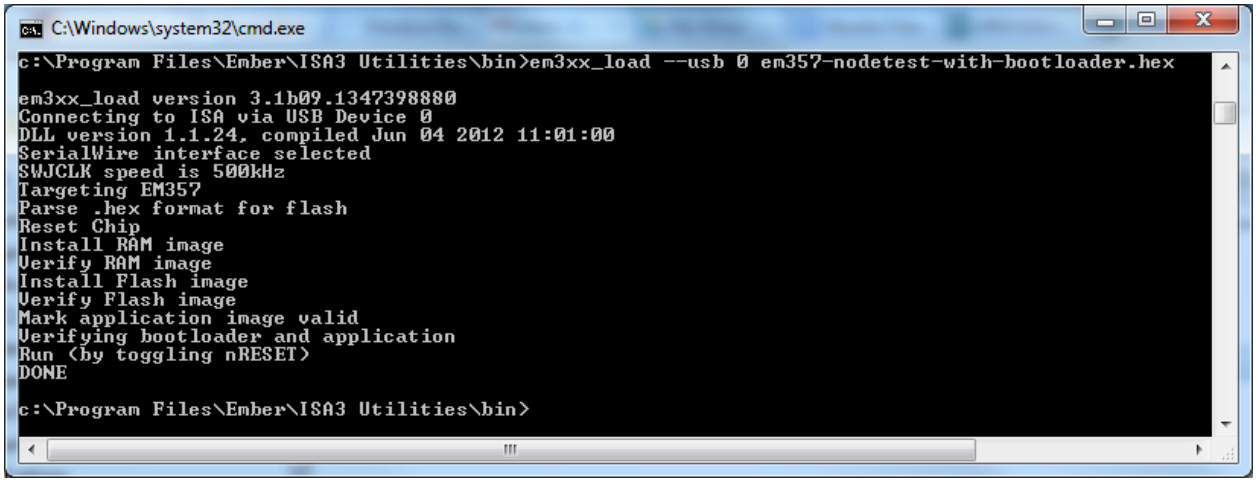

Figure 8. Shows an example of programming using `em3xx_load.exe` provided with the EM35xxUtilities from Silicon Labs.

The following is an example of using EM35xx-Utilties to load Node Test.
Move `em357-nodetest-with-bootloader.hex`
located in `.\Ember\<stack version>\EM35x-EZSP\app\nodetest`
to `c:\Program Files\Ember\ISA3 Utilities\bin`

Programming Node Test

```
c:\Program Files\Ember\ISA3 Utilities\bin>em3xx_load --usb 0 em357-nodetest-with-bootloader.hex

em3xx_load version 3.1b09.1347398880
Connecting to ISA via USB Device 0
DLL version 1.1.24, compiled Jun 04 2012 11:01:00
SerialWire interface selected
SWJCLK speed is 500kHz
Targeting EM357
Parse .hex format for flash
Reset Chip
Install RAM image
Verify RAM image
Install Flash image
Verify Flash image
Mark application image valid
Verifying bootloader and application
Run (by toggling nRESET)
DONE
```

2.3 – Terminal Example for Windows

Tera Term is shown for Windows 7, but any popular terminal program should work such as Putty, and HyperTerminal for Windows.

1. Open Tera Term and select Setup -> Terminal, check Local Echo and Transmit to CR+LF, as shown in Figure 9.

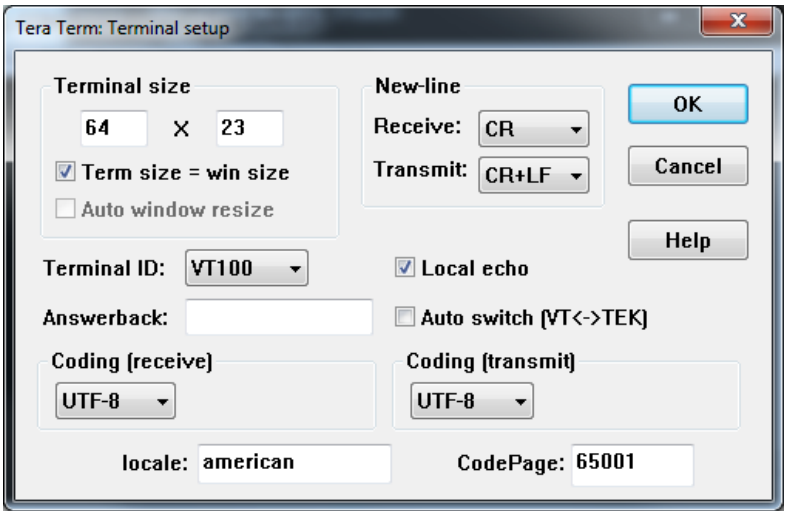

Figure 9. Terminal Setup showing Tera Term with Local Echo on and Transmit: CR+LF

2. Select Setup -> Serial, Setup Port Settings. Baud rate 115200, Data = 8, Parity= None, Stop = 1, Flow Control =none, as shown in Figure 10.

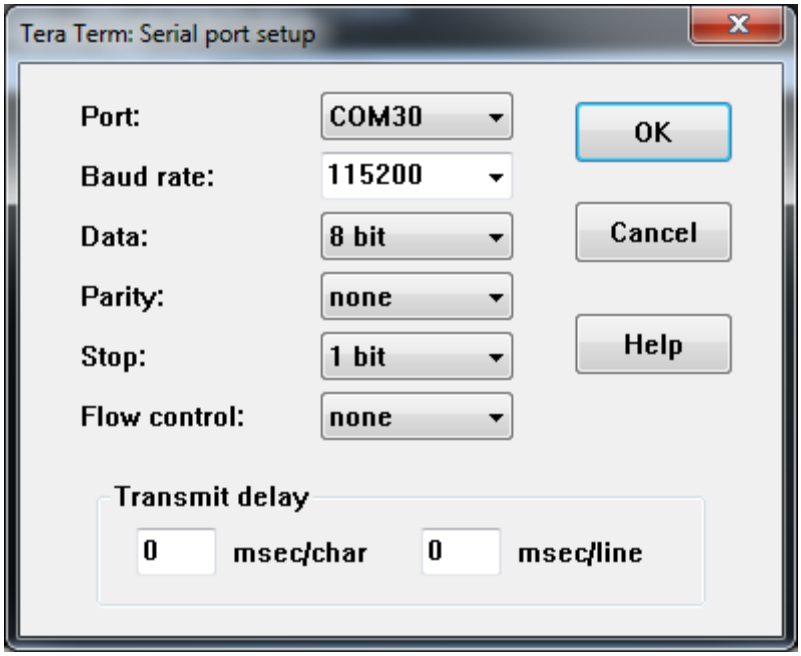

Figure 10. Tera Term Port Settings

Baud rate=115200, Data = 8, Parity= None, Stop = 1, Flow Control =none.

3. Press Enter or OK

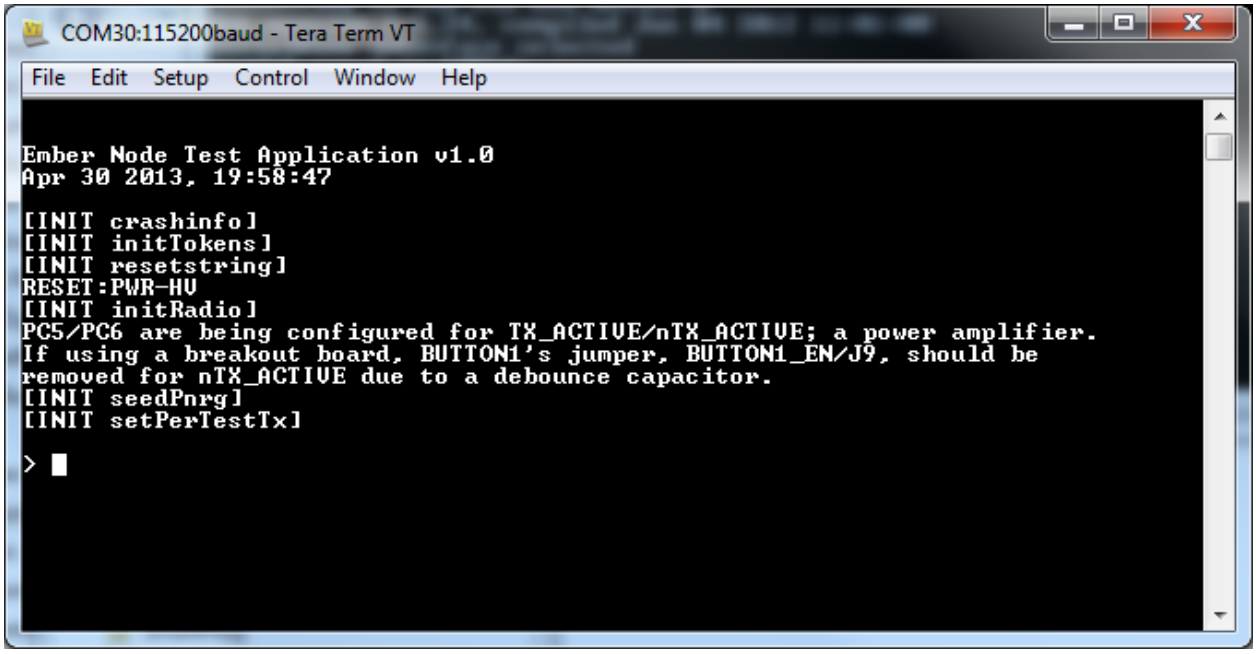

Figure 11. Example Tera Term session with Node Test running

Once you are connected to Node Test on the terminal, typing `help` will show a list of valid commands.

Note: *Caution, it is possible to erase the USB Stick parameters and transmit at power levels not allowed with this hardware.*

2.4 – ZM357S-USB(-LR) Simple LED Test

To verify that the USB drivers are installed and the sample node application is running, turn on/off the LED's provided as feedback for the end users ZigBee application. Figure 12 below shows the `ledon` and `ledoff` commands used to turn on/off the red and green LEDs on the ZM357S-USB(-LR) Stick.

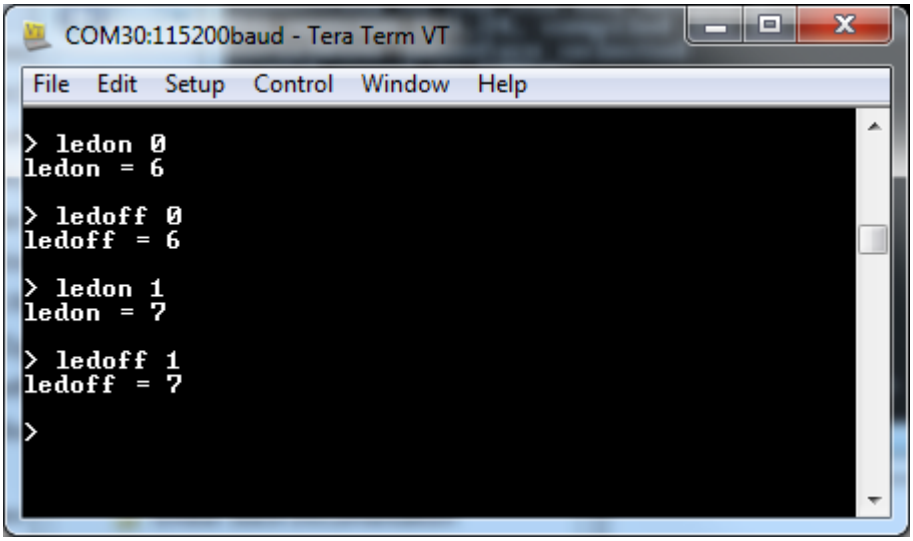

Figure 12. Tera Term showing LED test with Node Test Running on EM3567USB Stick

2.5 – ZM357S-USB(-LR) Simple GPIO Test

The following is an example of a GPIO read test.

```
> gpioread
GPIO_PAIN    = 0xA9
GPIO_PBIN    = 0x6F
GPIO_PCIN    = 0xDD
GPIO_PAOUT   = 0xA8
GPIO_PBOUT   = 0x4F
GPIO_PCOUT   = 0xC5
GPIO_PACFGH/L = 0x11991949
GPIO_PBCFGH/L = 0x98098891
GPIO_PCCFGH/L = 0x19944919
SWJ Enabled
```

Push Down on SW1 and perform another "`gpioread`".

```
> gpioread
GPIO_PAIN    = 0xA9
GPIO_PBIN    = 0x2F
GPIO_PCIN    = 0xDD
GPIO_PAOUT   = 0xA8
GPIO_PBOUT   = 0x4F
GPIO_PCOUT   = 0xC5
GPIO_PACFGH/L = 0x11991949
GPIO_PBCFGH/L = 0x98098891
GPIO_PCCFGH/L = 0x19944919
SWJ Enabled
```

As shown in Figure 13, the I/O should change from GPIO_PBIN = 0x6F to 0x2F corresponding to PB6 going active low.

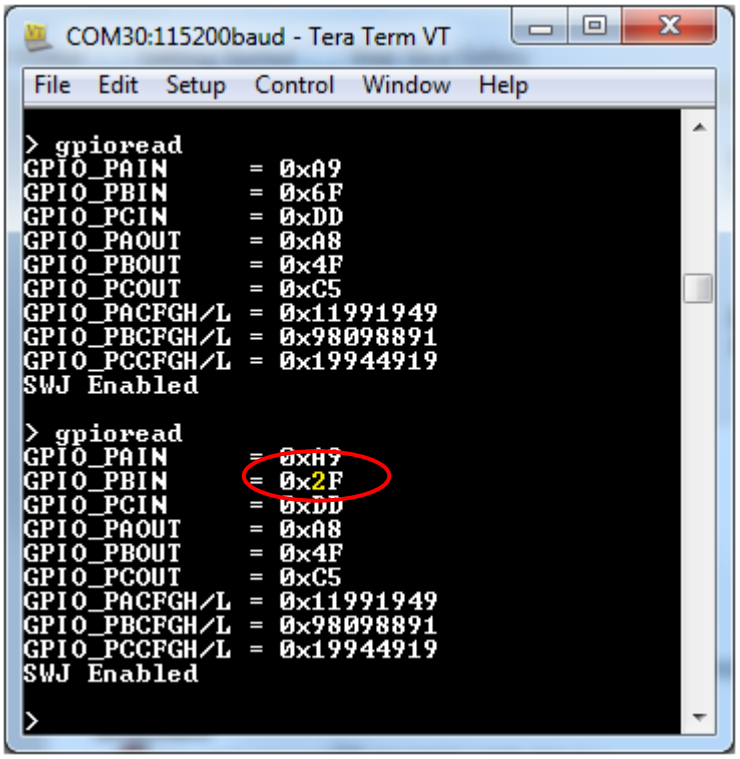

Figure 13. Screenshot of Tera Term showing the GPIO test

2.6 – Terminal Example for Mac OSX

The following is an example of how to perform the simple functional test with Node Test using Mac OSX.

Launch Terminal: Command+Space type "`Terminal`"

```
Sudo apt-get install minicom
Sudo minicom –s to setup minicom
```

Repeat the steps shown in Section 2.4 ZM357USB(-LR) Simple LED Test.

An example of using Minicom to run the simple functional test is shown in Figure 14.

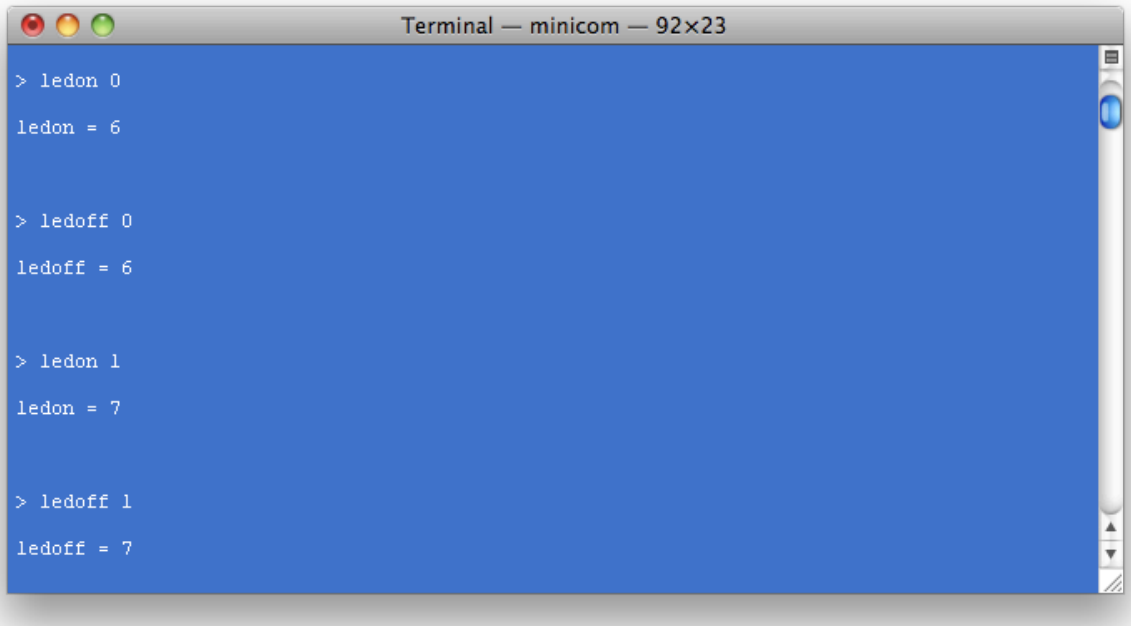

Figure 14. Screenshot of Minicom running on OSX and running the LED test

2.7 – Terminal Example for Linux (Ubuntu)

The following is an example of how to perform the simple functional test with Node Test using Linux (Ubuntu) and CuteCom. Reference the screen shots in Figures 14 and 15. Repeat the steps shown in Section 2.4 EM357USB(-LR) Simple LED Test

CuteCom is shown, but any popular terminal program should work such as Minicom, Screen, etc.

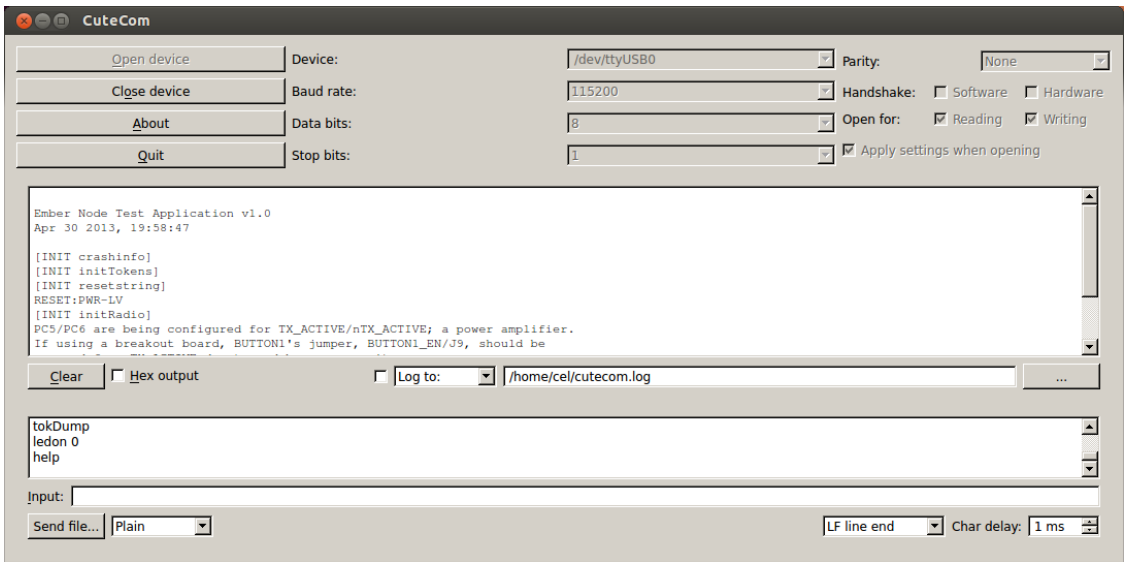

Figure 15. Screenshot of CuteCom running on Ubuntu showing the default response to CR/LF when the USB stick first boots up

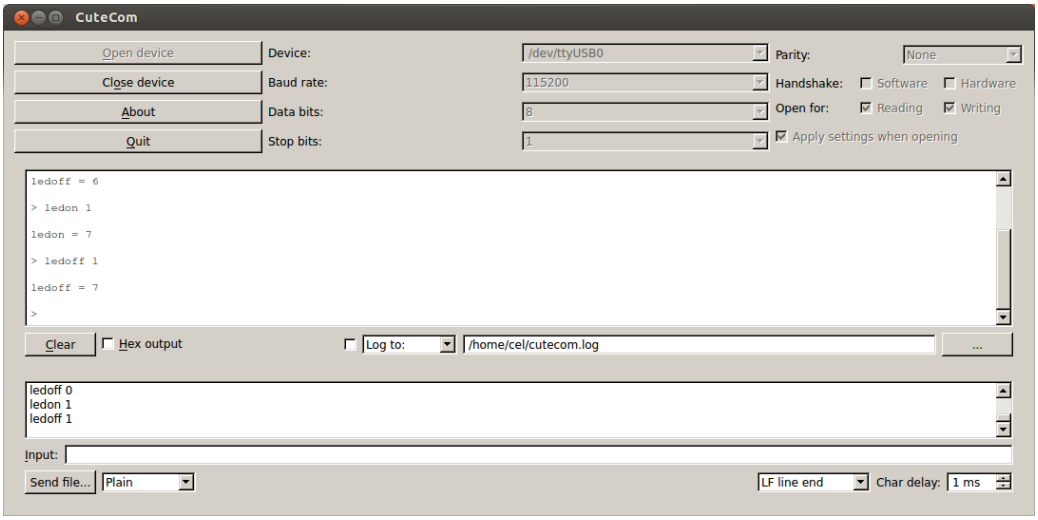

Figure 16. Screenshot of CuteCom running on Ubuntu showing the LED test with Node Test loaded on the ZM357S-USB

3 – USING ZM357-USB(-LR) STICKS AS A SNIFFER EXAMPLE

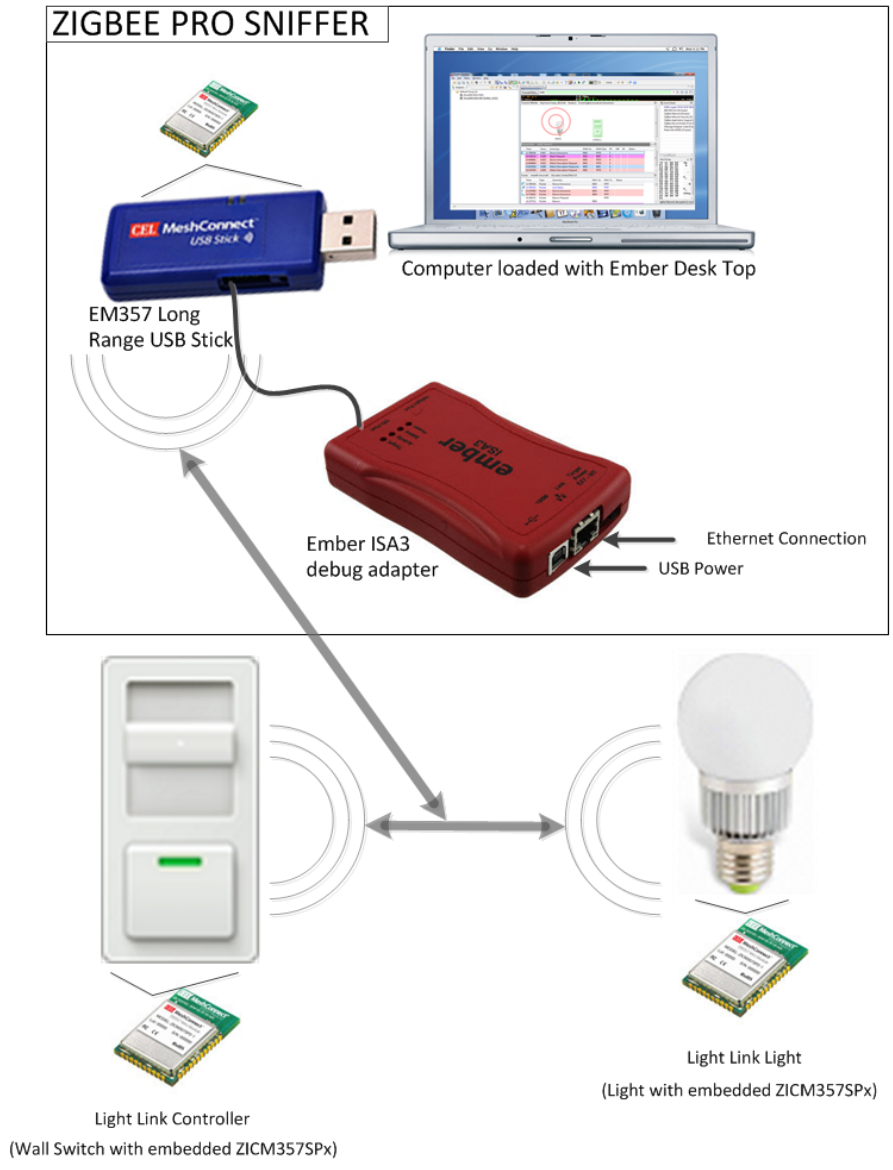

Figure 17. Setup for a ZM357S-USB(-LR) Stick being used to capture traffic that is generated between a ZigBee enabled Light Link Controller and a ZigBee enabled light

This setup presumes the Ember Desktop is configured and running, and a ZM357S-USB(-LR) has already been successfully installed and verified using NodeTest. The ISA3 InSight Port is connected to the programming port on the ZM357S-USB(-LR).

1. Within Ember Desktop, right click on the Adapter and select “Make a sniffer”, as shown in Figure 18.

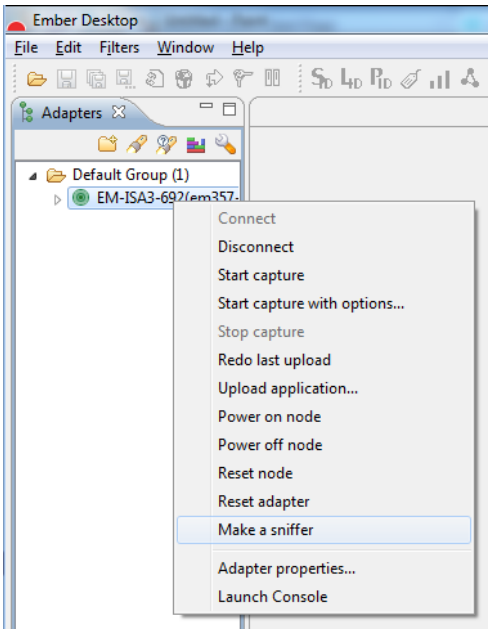

Figure 18. Make a Sniffer selection within Ember Desktop

2. Choose the EM357 Sniffer image (`internal>/images/sniffers/sniff-cortex-em357-dev0680.s37`), as shown in Figure 19.

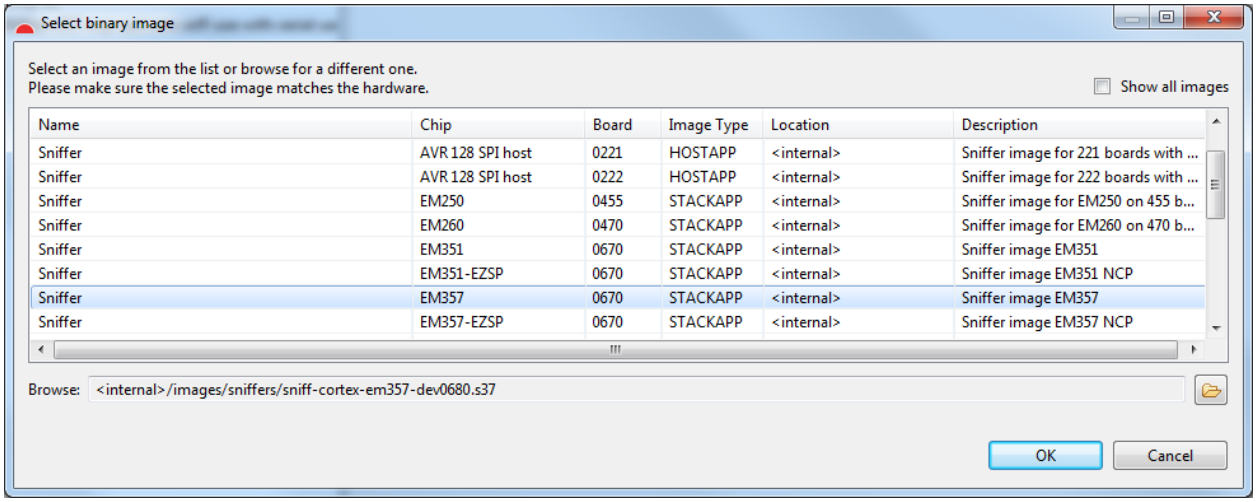

Figure 19. Ember Desktop screen shot selecting the Sniffer image for the EM357

3. Choose which Channel you would like to sniff using selection, as shown in Figure 20 and 21.

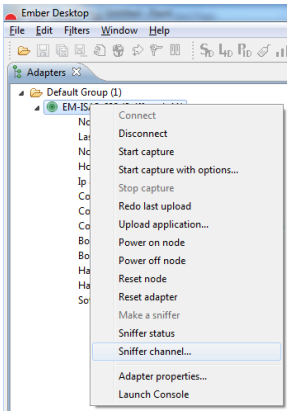

Figure 20. Selecting Sniffer Channel within Ember Desktop

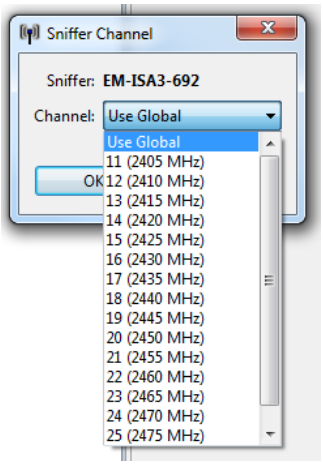

Figure 21. Selecting Sniffer Channel within Ember Desktop

4. Select “Start Capture”, as shown in Figure 22 .

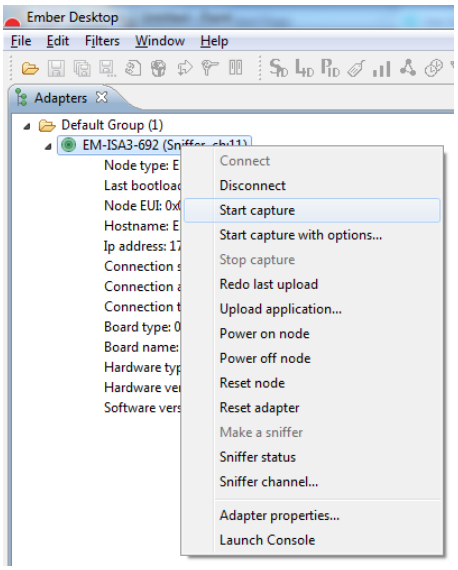

Figure 22. Selecting Start Capture in Ember Desktop

3.1 – Adding Network Keys

In order to sniff and be able to see the data inside a ZigBee packet you will need access to the ZigBee device's network key. One way to do this with the USB stick or ZigBee modules is to use a terminal program or Insight Desktop to connect to the console of a known good USB Stick or module that is able to form a network and use

the CLI(Command Line Interface). For more information about using the CLI please see EMBER® APPLICATION FRAMEWORK DEVELOPER GUIDE (UG102-AppFrameworkDevGuide.pdf).

The command line "`keys print`" can be used to get your keys on the ZigBee network. Please see the Silicon Labs *Application Framework* document in the Reference Section for more information.

As shown in Figure 23, within Ember Desktop - Select Edit -> Live Capture Security Keys and add your security key:

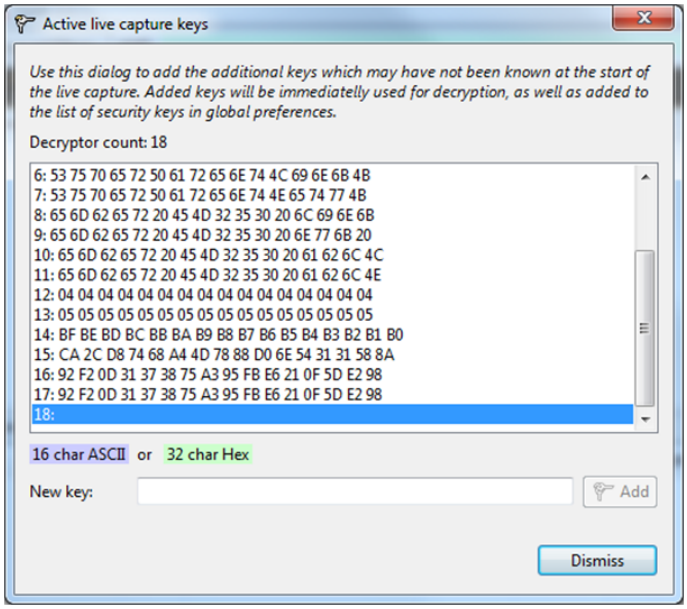

Figure 23. Ember Desktop screen shot showing how to add security keys

Once the network security keys have been added, it is possible to see network messages that are being passed back and forth through the network and look at the packets at the bit level.

3.2 – Example Messages being Sniffed by a Sniffer

Figure 24 shows an example of a device being announced.

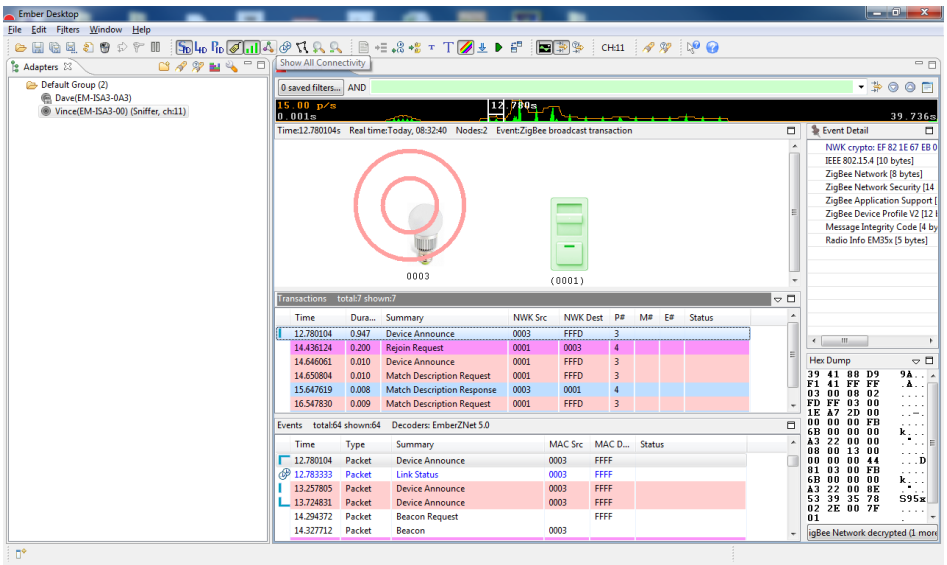

Figure 24. Example of a device announce being made from the light bulb

Figure 25 shows an example of a device sending a rejoin request.

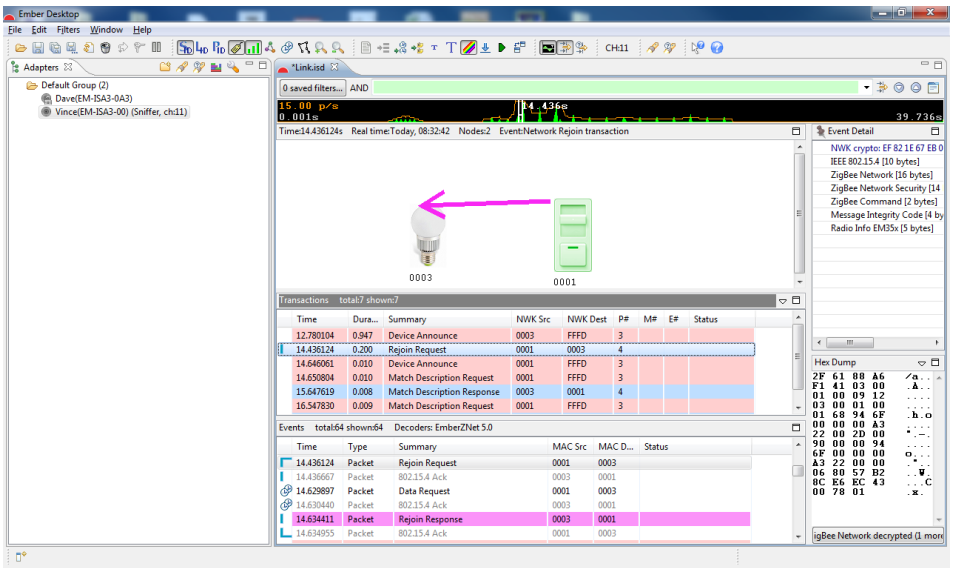

Figure 25. Example of a rejoin request being made from the light switch.

For more information please see the Silicon Labs *Testing and Debugging Applications For The Ember® Em2xx and Em35x Platforms* document in the Reference Section.

4 – EXAMPLE IMPLEMENTATION OF NCP ON USB STICK (WINDOWS CYGWIN)

The EmberZNet Serial Protocol (EZSP) is the protocol used by a host application processor to interact with the EmberZNet PRO stack running on a Network CoProcessor (NCP). EZSP messages are sent between the host and the NCP over either a SPI or a UART interface.

1. Set up Cygwin environment for running Linux-style builds under Windows. Cygwin can be downloaded from the Internet from www.cygwin.com. Cygwin is a collection of hundreds of small programs that make up a Linux distribution. Cygwin can be configured in many different ways for a particular Windows PC and so by default may not include the required tools for building an EmberZNet PRO host application. Proper installation of the required tools is necessary before actually building the EmberZNet PRO code and is outside the scope of this quick start guide. Cygwin provides a Linux-like environment for Windows. It can be used to build the EZSP-UART host code provided with Ember's NCP platform software release that can be built with Cygwin using the following packages: `gcc`, `make`, `readline`, and `ncurses`. Search and select packages to install during Cygwin install for EX: Base, Debug, Devel, Libs, Utils

   Note: This step might take a few hours depending on your download speed etc.

   The mirror used for the development of this document for the Cygwin download was:
   http://cygwin.com/mirrors.html
   United States: Illinois: mirror.team-cymru.org(http), mirror.team-cymru.org(rsync)

   Please pay special attention during download to search “Base”, “Debug”, “Devel”, “Libs”, “Utils” and make sure these components are selected to be installed.

2. Set up EmberZNet Pro installation for EM35x NCP platform by downloading and installing the corresponding stack from Silicon Labs. *Support Site is limited to current customers of Ember product.* If you are an Ember ZigBee customer, you can request an account via email: portal.logins@silabs.com

   If you are having trouble downloading the stack, please contact Silicon Labs support at silabs.com/zigbee-support

Once the stack is installed, the default location is:
`C:\Users\<username>\Ember\<stack version>\EM35x-EZSP\app\ezsp-uart-host`

Note: <username> is your computer username and <stack version> is the stack that was downloaded and installed from Silicon Labs.

For more information please see *EZSP Reference Guide (EZSPReferenceGuide.pdf) and EZSP-UART HOST INTERFACING GUIDE (AN706.pdf)*

3. Set up ZM357S-USB(-LR) using the *USB Stick Driver Installation Instruction Application Note* in the Reference Section.

4. Use the Insight Desktop to download the EZSP – NCP application onto the target ZM357S-USB. For example an EZSP – NCP application for an installed EmberZNet5.0.1-GA stack would be located here on a Windows 7 system:

   `C:\Users\<username>\Ember\<stack version>\EM35x-EZSP\build\em35x-ezsp-images\EM357\em357-ncp-uart-xon-xoff-use-with-serial-uart-bl-501.hex.`

   Where <username> is your computer username

5. Launch a Cygwin Terminal and verify that you can build Ember EZSP-UART host targets from the provided host code and makefiles. Use the "`cd`" command to change directories to the `/cygdrive/c/users/<username>/Ember/<stack version>/EM35x_EZSP/app/ezsp-uart-host` subdirectory of the EmberZNet installation for EM35x NCP platform (where <user name> and <stack version> are your local computers user name and the stack version that you downloaded and installed from Silicon Labs). You may need to use "cd /" and "cd cygdrive" if you have installed Cygwin to a different directory. Type "`make`" and ensure that the build process completes successfully. The Cygwin terminal should print "`All builds succeeded.`" at the end and leave you with 3 `uart-test-x` executables.

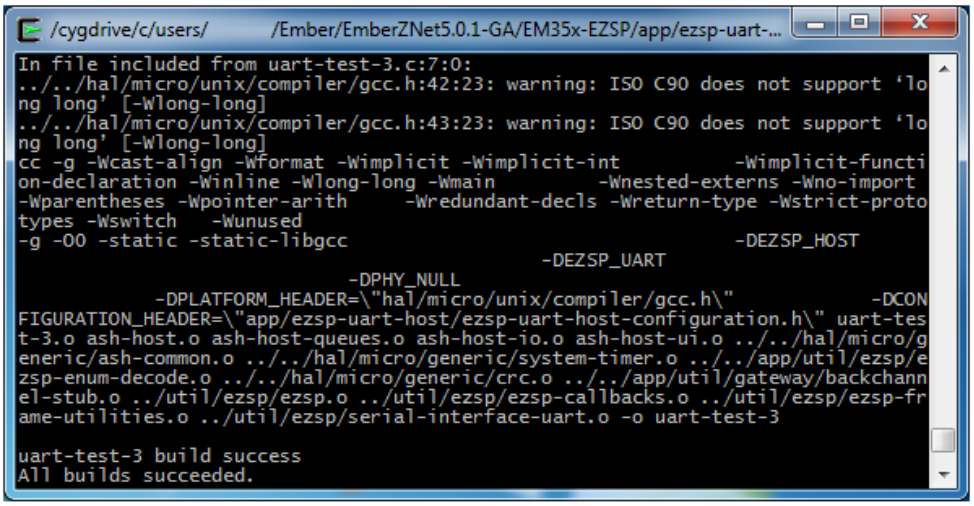

Figure 23. Cygwin build on a Windows, the message "All builds succeeded" should be seen.

6. Use Ember Desktop or the `em3xx_load.exe` tool to load the appropriate EZSP-UART Xon/Xoff firmware to your target EM35x module. For example, for an EM357 module in the EmberZNet5.0.1-GA release, you can use the `em357-ncp-uart-xon-xoff-use-with-serial-uart-bl-501.hex` file from the `.\Ember\EmberZNet5.0.1-GA\EM35x-EZSP\build\em35x-ezsp-images\EM357\` subdirectory of the EmberZNet installation. (If you load the `*.hex` file, you do not need to select a bootloader for the upload process as this is already on the ZM357S-USB module by default.)

Note: *The user must select the XON/XOFF build. RTS/CTS builds will not work with the CEL EM35x USB Sticks*

7. Move the `cygwin1.dll` to the `app/ezsp-uart-host` or register it on windows using cd to `C:\cygwin\bin` and run "`regsvr32 cygwin1.dll`". You need to be running as administrator to register a DLL. To do this manually find the Command Prompt item in the Start menu and right-click on it and select "Run As Administrator".

8. Check the COM port number of the new serial port using Windows Device Manager (under Control Panel). `Device Manager -> Ports (COM & LPT) -> CEL EM357 ZigBee USB Stick Long Range`

9. Open a CMD prompt (Windows+R type "CMD") in windows and Verify the EZSP-UART connection to the EM35x NCP by running the uart-test-2.exe with a command such as:

   `cd C:\Users\<user name>\Ember\<stack version>\EM35x-EZSP\app\ezsp-uart-host\uart-test-2 -n 1 -p 28`

   Figure 23 shows an example of `uart-test-2.exe` running in Windows 7 connected to a ZM357S-USB loaded with `Eszp-uart-host` software; there are 18 available tests that can be run with `uart-test-2.exe`. For example on a NCP connected on serial port 28, (use the COM port number found in the previous step), the command would be:

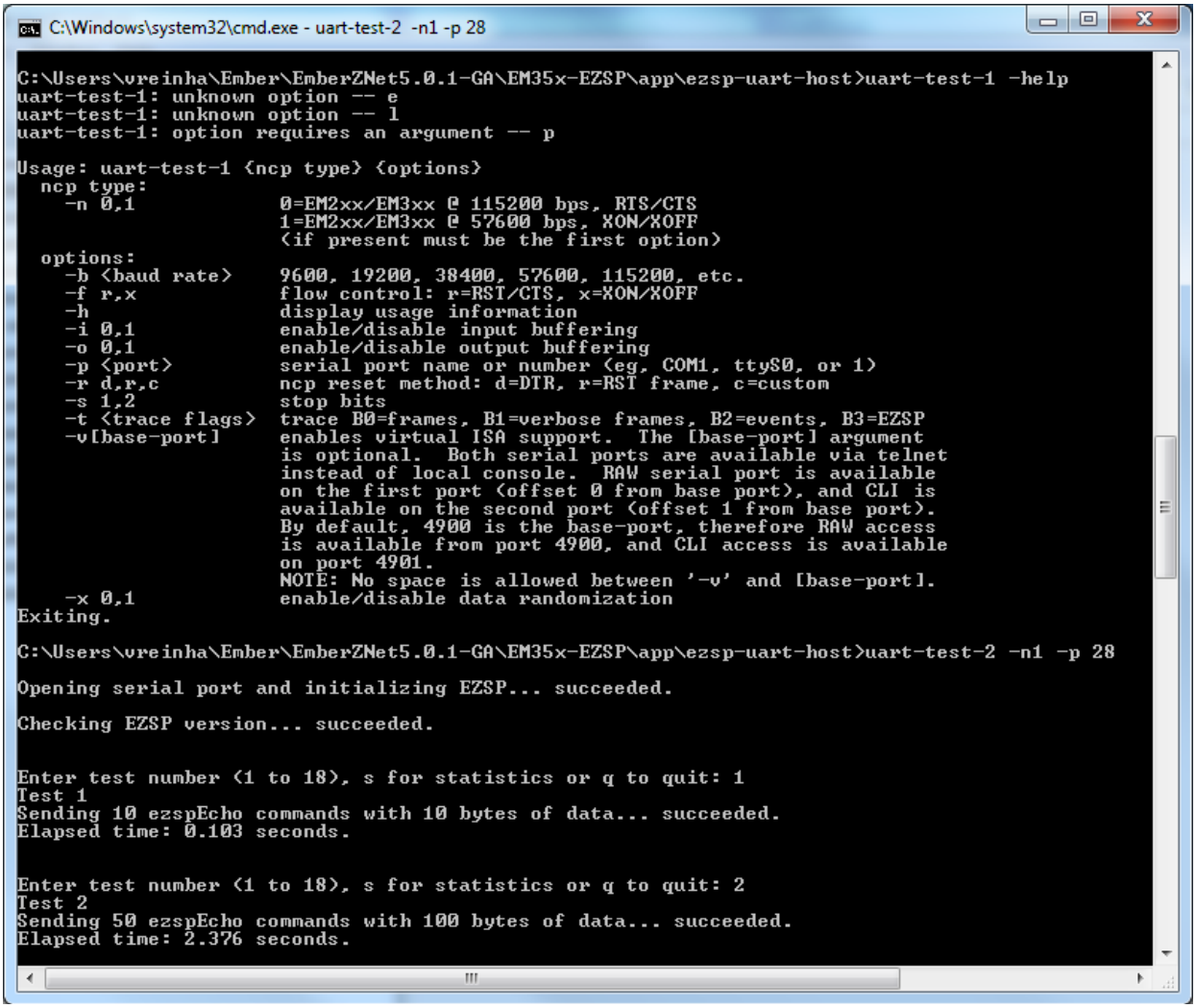

Figure 24. Example Screen shot of `uart-test-2.exe` running in Windows 7 connected to a ZM357S-USB loaded with `Ezsp-uart-host` software; there are 18 available tests that can be run with `uart-test-2.exe`

5 – CONCLUSION

The MeshConnect ZM357S-USB(-LR) Sticks provide a quick way to deploy and test ZigBee networks on Windows/MAC/Linux. Using this Quick Start Guide the user should now be familiar with programming, how to use Node Test, how to startup and NCP sample application, and how to configure a USB Sniffer.

REFERENCES

Reference Documents	Download
California Eastern Laboratories	
Download Replacement Bootloader	Link
0011-01-16-03-001 – USB Stick Driver Installation Application Note	Link
Silicon Labs	
UG110 - EM35x Development Kit User Guide)	Link
APPLICATION FRAMEWORK DEVELOPER GUIDE (UG102-AppFrameworkDevGuide.pdf)	Link
120-4032-000 - Ember EM35xx Utilities Guide for the EM35x SoC Platform	Link
120-3023-000 – Applications Framework Reference	Link
120-3030-000 – Testing and Debugging Applications for the Ember EM2xx and EM3xx Platforms	Link
EmberZet Pro Installation for EM35x NCP platform by downloading the corresponding stack from Silicon Labs. Support Site is limited to current customers of Ember product. If you are an Ember ZigBee customer, you can request an account via email at portal.logins@silabs.com	Link
Cygwin	
Cygwin Download	Link

REVISION HISTORY

Previous Versions	Changes to Current Version	Page(s)
0011-01-16-03-000 (Issue A) January 31, 2014	Initial Release	N/A
0011-01-16-03-000 (Issue B) March 24, 2014	Swapped I/Os PA6 and PA7 in Table 1	3

DISCLAIMER

The information in this document is current as of the published date. The information is subject to change without notice. For actual design-in, refer to the latest publications of CEL Data Sheets or Data Books, etc., for the most up-to-date specifications of CEL products. Not all products and/or types are available in every country. Please check with an CEL sales representative for availability and additional information.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of CEL. CEL assumes no responsibility for any errors that may appear in this document.

CEL does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of CEL products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of CEL or others.

Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. CEL assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

While CEL endeavors to enhance the quality, reliability and safety of CEL products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in CEL products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

FOR MORE INFORMATION

For more information about CEL MeshConnect products and solutions, visit our website at www.cel.com/MeshConnect.

TECHNICAL ASSISTANCE

For Technical Assistance, visit http://www.cel.com/MeshConnectHelp