Crossbow

MoteConfig User's Manual

Revision A, November 2006

PN: 7430-0112-01





www.xbow.com

Table of Contents

About This Document

The following annotations have been used to provide additional information.

NOTE

Note provides additional information about the topic.

☑ EXAMPLE

Examples are given throughout the manual to help the reader understand the terminology.

IMPORTANT

This symbol defines items that have significant meaning to the user

WARNING

The user should pay particular attention to this symbol. It means there is a chance that physical harm could happen to either the person or the equipment.

The following paragraph heading formatting is used in this manual:

1 Heading 1

1.1 Heading 2

1.1.1 Heading 3

This document also uses different body text fonts (listed in Table 0-1) to help you distinguish between names of files, commands to be typed, and output coming from the computer.

Table 0-1. Font types used in this document.

Font Type	Usage
`Courier New Normal`	Sample code and screen output
`Courier New Bold`	Commands to be typed by the user
Times New Roman Italic	TinyOS files names, directory names
Franklin Medium Condensed	Text labels in GUIs

1 Introduction

1.1 MoteConfig

MoteConfig is a Windows-based GUI utility for programming Motes. This utility provides an interface for configuring and downloading pre-compiled *XMesh*/TinyOS firmware applications onto Motes. MoteConfig allows the user to configure the Mote ID, Group ID, RF channel and RF power. The user can also enable the over-the-air-programming feature present on all *XMesh* - based firmware. High-power and low-power *XMesh* applications are available for each sensor board and platform manufactured by Crossbow as part of the MoteView install (see section 2.3).

Table 1-1. Pre-compiled MICAz XMesh applications

MICAz Mote (MPR2400 and MPR2600)	
Board Model	**Binary file name**
MTS boards	
MTS101	*XMTS101_2420_<mode>.exe*
MTS300CA	*XMTS300CA_2420_<mode>.exe*
MTS300CB	*XMTS300CB_2420_<mode>.exe*
MTS310CA	*XMTS310CA_2420_<mode>.exe*
MTS310CB	*XMTS310CB_2420_<mode>.exe*
MTS400	*XMTS400_2420_<mode>.exe*
MTS410	*XMTS410_2420_<mode>.exe*
MTS420	*XMTS420_2420_hp.exe*
MTS450	*XMTS450_2420_<mode>.exe*
MDA board	
MDA100CA	*XMDA100CA_2420_<mode>.exe*
MDA100CB	*XMDA100CB_2420_<mode>.exe*
XBW-DA100CA	*XBW-DA100CA_2420_hp.exe*
XBW-DA100CB	*XBW-DA100CB_2420_hp.exe*
MDA300	*XMDA300_2420_<mode>.exe*
MDA300 (precision)	*XMDA300p_2420_<mode>.exe*
MDA320	*XMDA320_2420_<mode>.exe*
XBW-DA325	*XBW-DA325_2420_<mode>.exe*
Base Station (common to all boards)	
XMeshBase_2420_<mode>.exe	

<mode> = hp or lp.

hp = high power mesh networking. lp = low-power mesh networking via low-power listening and time synchronized data transmissions.

Table 1-2. Pre-compiled MICA2 XMesh Applications

MICA2 Mote (MPR4x0, x = 0, 1, or 2 and MPR600)	
Board Model	**Binary file name**
MTS boards	
MTS101	*XMTS101_xxx_<mode>.exe*
MTS300CA	*XMTS300CA_xxx_<mode>.exe*
MTS300CB	*XMTS300CB_xxx_<mode>.exe*
MTS310CA	*XMTS310CA_xxx_<mode>.exe*
MTS310CB	*XMTS310CB_xxx_<mode>.exe*
MTS400	*XMTS400_xxx_<mode>.exe*
MTS410	*XMTS410_xxx_<mode>.exe*
MTS420	*XMTS420_xxx_<mode>.exe*
MTS450	*XMTS450_xxx_<mode>.exe*
MDA board	
MDA100CA	*XMDA100CA_xxx_<mode>.exe*
MDA100CB	*XMDA100CB_xxx_<mode>.exe*
MDA300	*XMDA300_xxx_<mode>.exe*
MDA300 (precision)	*XMDA300p_xxx_<mode>.exe*
MDA320	*XMDA320_xxx_<mode>.exe*
Base Station (common to all boards)	
XMeshBase_xxx_<mode>.exe	

xxx = 315, 433, or 915. <mode> = hp or lp. hp = high power mesh networking. lp = low-power mesh networking via low-power listening and time synchronized data transmissions.

Table 1-3. Pre-compiled MICA2DOT XMesh Applications

MICA2DOT Mote (MPR5x0, x = 0, 1, or 2)	
Board Model	**Binary file name**
MTS boards	
MTS510	*XMTS510_xxx_<mode>.exe*
MDA boards	
MDA500	*XMDA500_xxx_<mode>.exe*
Base Station (common to all boards)	
XMeshBase_Dot_xxx_<mode>.exe	

xxx = 315, 433, or 915. <mode> = hp or lp. hp = high power mesh networking. lp = low-power mesh networking via low-power listening and time synchronized data transmissions.

Table 1-4. Pre-compiled MSP XMesh applications

MSP410 Mote Security Package	
MSP410 Module	*XMSP410_433_hp.exe*
MSP410 Base Station	*XMeshBase_433_hp.exe*

Table 1-5. Pre-compiled MEP XMesh applications

MEP410 Mote Environmental Package
MEP410 module *MEP410_433_<mode>.exe*
MEP510 Mote Environmental Package
MEP510 module *MEP510_433_<mode>.exe*
Base Station (common to both modules)
XMeshBase_433_<mode>.exe

<mode> = hp or lp. hp = high power mesh networking. lp = low power mesh networking via low-power listening and time synchronized data transmissions.

1.2 Over – The – Air – Programming (OTAP)

The Over-The-Air-Programming (*OTAP*) feature allows users to reprogram a *Mote* over a wireless link. *OTAP* allows one or more *Motes* in the *XMesh* network to receive new firmware images from *XServe* (via the *XOtap* service).

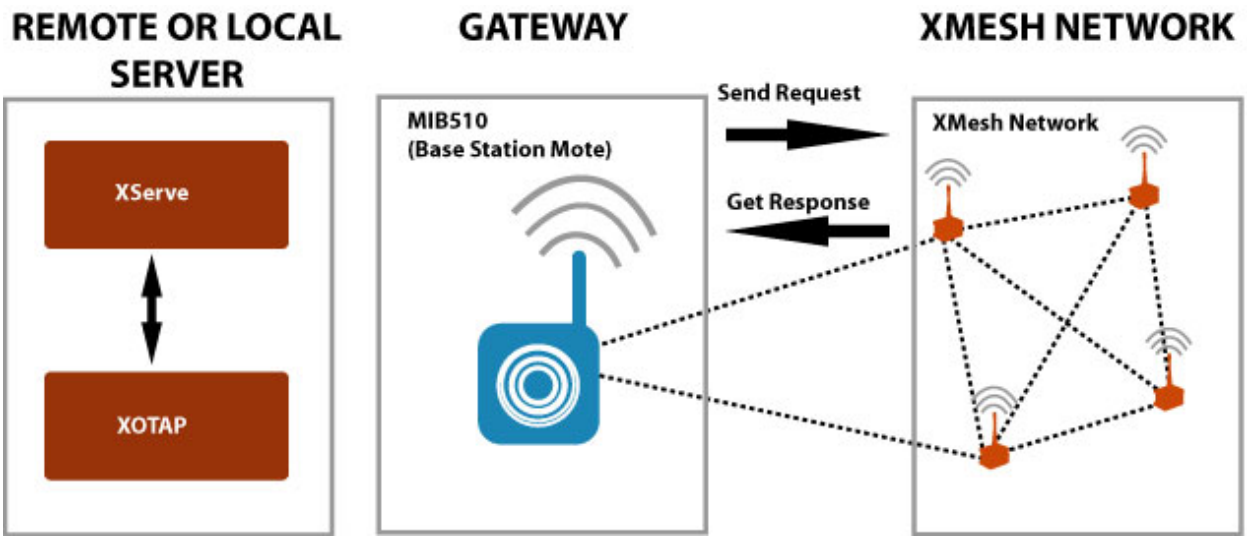

Figure 1-1: XOTAP Architecture

Each *Mote* has a 512kB external non-volatile flash divided into 4 slots. These slots have a default size of 128 kB. Slot 0 is reserved for the OTAP image. Slots 1, 2 and 3 can be used for user-specified firmware.

During the *OTAP* process, the server sends a command to the Mote to reboot into the *OTAP* image (slot 0). A user-specified firmware image is broken up into fragments and transmitted to the Mote and stored into Slot 1, 2 or 3. The server can send a message to transfer the newly uploaded firmware into the program flash and reboot the *Mote*.

The following components are required for OTAP to work:

- *XServe* and *XOtap* running on the server
- Firmware applications that include the *XOTAPLiteM* component (this is automatically included when the firmware is built with *XMesh*)
- The Mote needs to have pre-configured with a bootloader in the program flash, and the OTAP image in slot 0 of the external flash. Both of these conditions are met by selecting **OTAP enable** during the MoteConfig download process outlined in Section 4.

2 Installation

2.1 Supported Platforms

MoteConfig is supported on the following operating systems:

- *Windows XP Home*
- *Window XP Professional*
- *Windows 2000 with SP4*

2.2 PC Interface Port Requirements

The gateway platform used in the base station determines the PC interface port required by *MoteConfig*.

1. For a **MIB510** serial gateway: an RS-232 serial port.
2. For a **MIB520** USB gateway: a USB port.
3. For a **MIB600** Ethernet gateway: A wired Ethernet or 802.11 wireless card (if the MIB600 is on a LAN with wireless access).
4. For a **Stargate** server: A wired Ethernet, an 802.11 wireless card (if the Stargate has a wireless modem or is on a LAN with wireless access), or a cellular modem for wireless Internet access.

2.3 Installation Steps

MoteConfig is shipped as a component of *MoteView* and *MoteWorks*:

1. *MoteConfig* is automatically installed with the *MoteView* installer. Refer to the *MoteView Users Manual* for additional details.
2. *MoteConfig* is an optional component in the *MoteWorks* installer. Make sure that **MoteConfig 2.0 and OTAP** item is selected as shown in Figure 2-1.

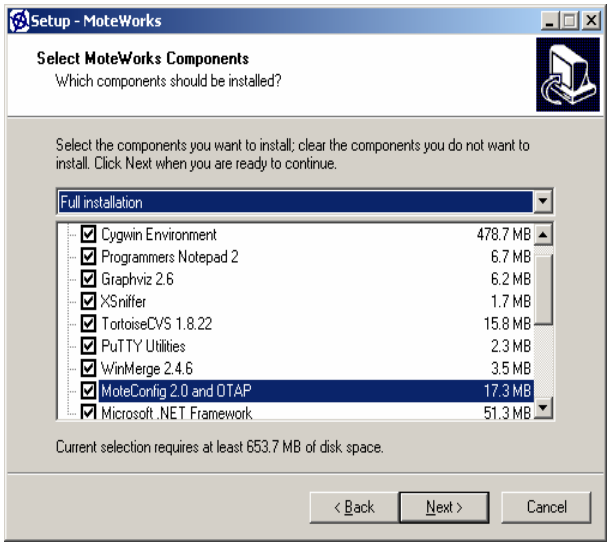

Figure 2-1: MoteConfig 2.0 and OTAP – MoteWorks Installer

3 Starting MoteConfig

If *MoteConfig* was installed using the *MoteView* installer, use the following steps:

- Open *MoteView1.4C* by either clicking on the shortcut located on the Desktop, or by going to *Start > Programs > Crossbow > MoteView 1.4C.*
- Press the **Program Mote** button () on the *MoteView* toolbar to spawn the *MoteConfig* GUI as shown in Figure 3-1

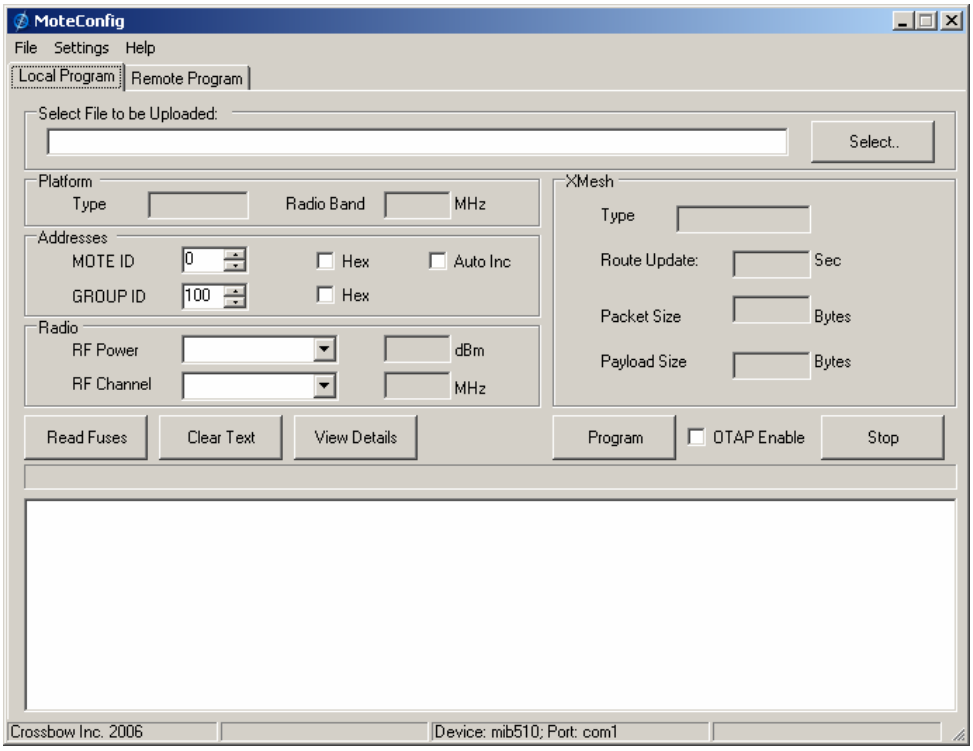

Figure 3-1: MoteConfig Application GUI

If *MoteConfig* was installed using the *MoteWorks* installer

- Click on the shortcut located on the Desktop, or select *Start > Programs > Crossbow > MoteConfig 2.0.*

4 Local Programming

The **Local Program** tab is used to upload firmware onto the Motes via a gateway.

IMPORTANT: To program motes correctly, set up the hardware as follows:

1. The gateway should be powered and connected to the PC via a serial, USB or Ethernet port.
2. If using the MIB510, the SW2 switch should be in the "OFF" position.
3. The motes should be firmly attached to the gateway.
4. The motes should be turned off before the programming.

4.1 Settings

Click on *Settings* > *Interface Board...* to select the correct gateway and port settings.

Figure 4-1 shows the **Interface Board Settings** for a **MIB510** on COM 1.

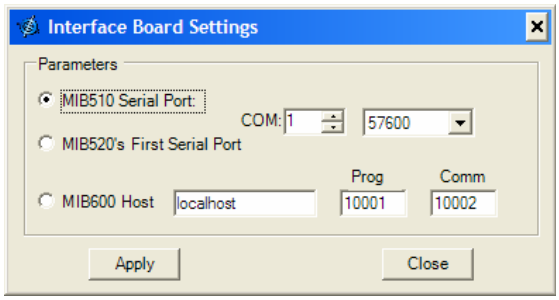

Figure 4-1: MIB510 Gateway Settings

The MIB520 virtual COM port drivers will install two sequential ports on the PC. The low-numbered port is used for programming and the high-numbered port is used for communication. Figure 4-2 shows the **Interface Board Settings** for a **MIB520** that has created COM 6 and 7 on the PC. In this example, COM 6 must be selected as the serial port.

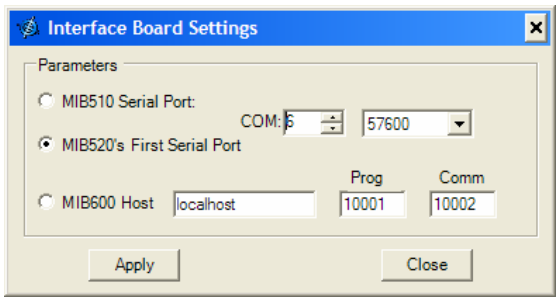

Figure 4-2: MIB520 Gateway Settings

NOTE: The **MIB520** requires the installation of the *FTDI FT2232C* drivers. Once these drivers are installed, the **Device Manager (Start > Control Panel > System > Hardware)** will display the **MIB520** as two new virtual com ports. Refer to the *MPR-MIB Series User's Manual* for details.

Figure 4-3 shows the **Interface Board Settings** for a **MIB600** on the same LAN as the PC with an assigned IP address of 10.1.1.99.

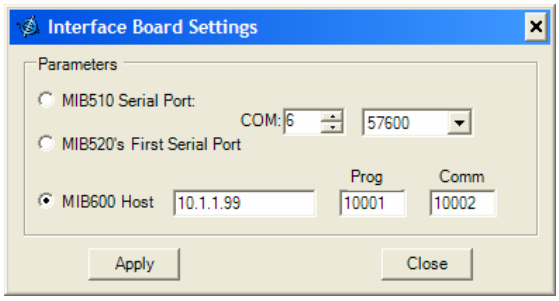

Figure 4-3: MIB600 Gateway Settings

NOTE: The IP address of the **MIB600** can be identified by using the *Lantronix DeviceInstaller* application. Refer to the *MPR-MIB Series User's Manual* for details.

4.2 Programming

The pre-compiled XMesh applications installed with MoteView are located in *C > Program Files > Crossbow > MoteView > XMesh*.

Press the **Select** button to open a file browser as shown in Figure 4-4. Navigate to the folder that corresponds to your Mote processor/radio board, radio frequency (for MICA2 and MICA2DOT) and sensor board type.

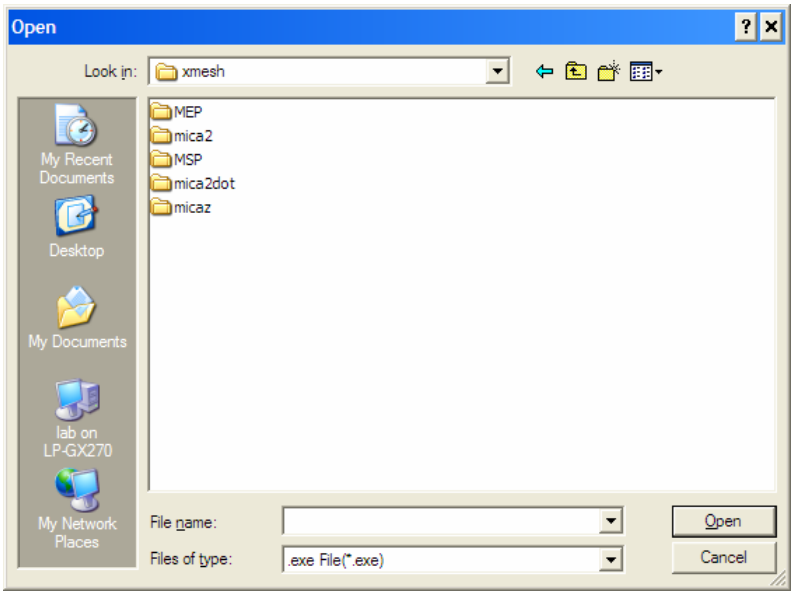

Figure 4-4: File Browser for selecting XMesh applications.

Low-power and high-power applications have been included for most sensor boards.

Note that the MEP and MSP node firmware is located in separate named folders.

NOTE: The base station Mote must be programmed with *XMeshBase_xxx_<mode>.exe* and a node ID of 0.

After an application has been selected, the binary scan feature built into *MoteConfig* will display the default parameters programmed into the application (see Figure 4-5).

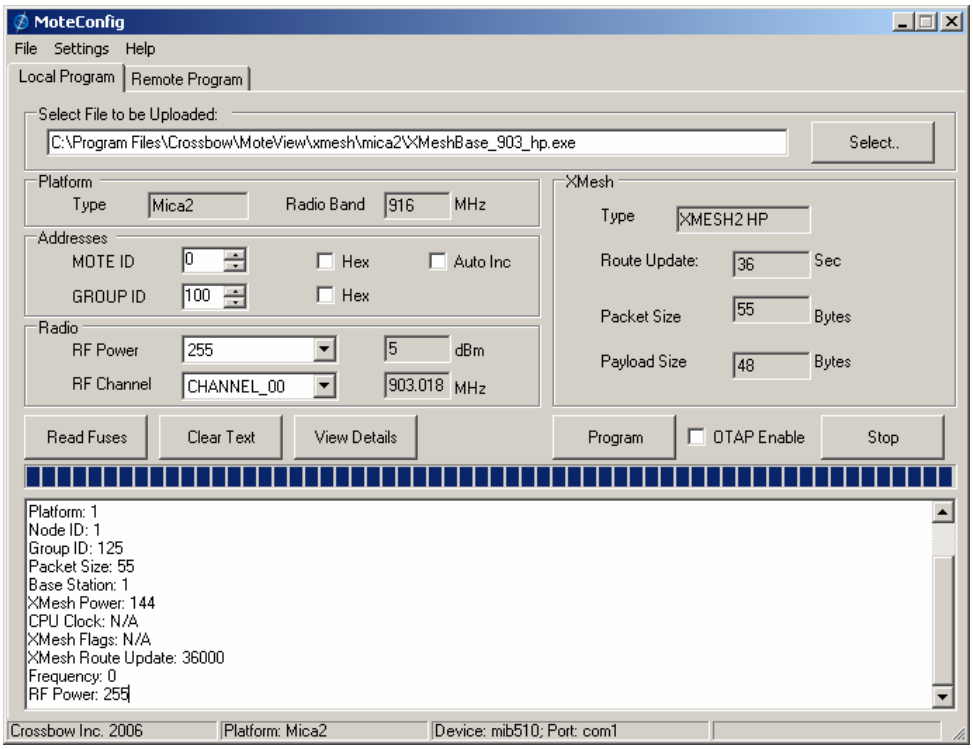

Figure 4-5: Binary Scan Result of an XMeshBase application

These default parameters can be overwritten by the user by specifying the desired **MOTE ID**, **GROUP ID**, **RF Power**, and **RF Channel.**

NOTE: Remote nodes must be programmed with a non-zero Mote ID.

Press the **Program** button to download the selected firmware and configuration into the mote, as shown in Figure 4-6.

When programming is complete, the "`Upload SUCCESSFUL!`" message is printed in the status box as shown in Figure 4-7.

The **Stop** button can be used to cancel a firmware download in progress.

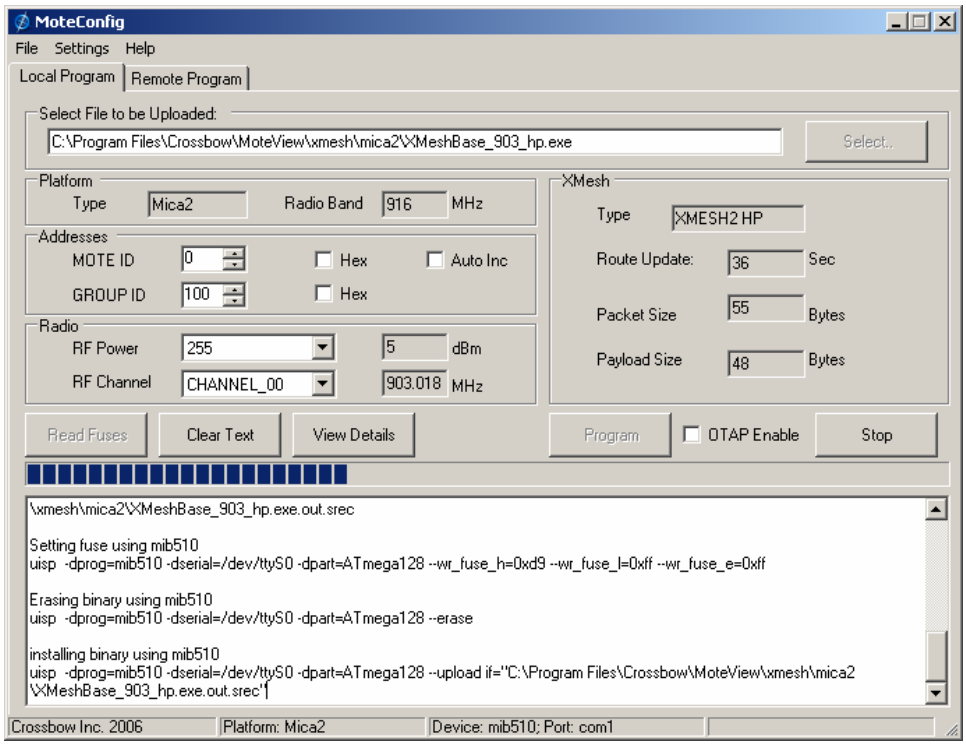

Figure 4-6: MoteConfig programming in progress

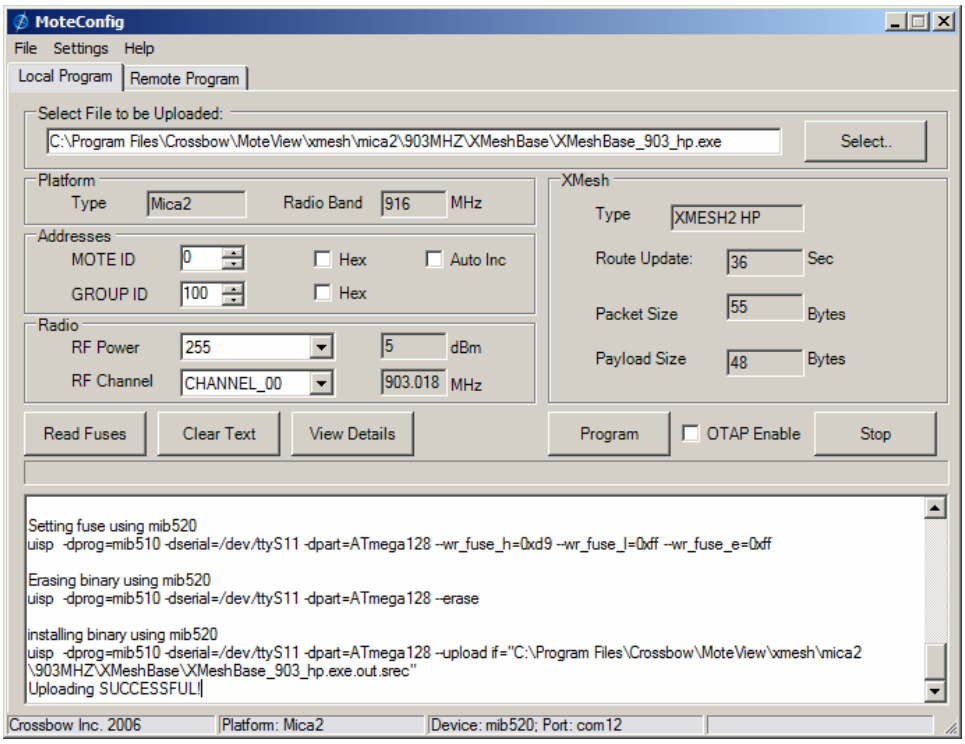

Figure 4-7: MoteConfig programming successful

The table below describes the advanced options available in the *MoteConfig* GUI.

Table 4-1. Advanced Options

Advanced Options	Description
Hex	Enables users to specify the ID as a hexadecimal value
Auto Inc	Increments the mote ID by 1 after a mote has been programmed
OTAP Enable	Allows users to enable a mote for OTAP (refer to section 5)

4.3 Fuse Settings

MoteConfig allows users to overwrite the default fuse settings of the *ATmega128* processor. The fuses are an internal set of software switches within the *ATmega128* that enable certain functions.

Select *Settings* > *Fuse Defaults…* to open the dialog box shown in Figure 4-8. Check **Override default fuse settings** to modify the available fuse options. These options are:

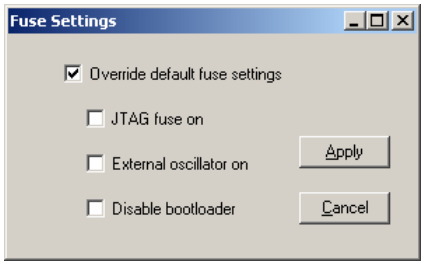

Figure 4-8: Fuse Settings Dialog

- **JTAG fuse on** activates the JTAG debug mode for the *ATmega128*. When enabled, the processor draws an additional current of 3 mA. By default, this fuse is turned off for all *XMesh* apps.
- **External oscillator on** forces the firmware app to use an external oscillator for its timer. When enabled, the processor draws more current. By default, this fuse is disabled for low power *XMesh* apps and enabled for high power *XMesh* apps.
- **Disable bootloader** will prevent the *Mote* from executing the boot loader code on reboot. By default, the bootloader is enabled for *XMesh* apps to provide *OTAP* functionality.

4.4 Address and Radio Defaults

The default behavior associated with setting the **Group ID**, **RF Power** and **RF Channel** for each node can be changed from the Address and Radio Defaults dialog shown in Figure 4-9. This can be opened be selecting *Settings* > *Address and Radio Defaults …*

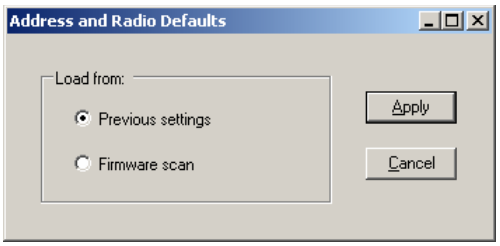

Figure 4-9: Address and Radio Defaults Dialog

When a new firmware application is **selected**, the default values for the **Group ID**, **RF Power** and **RF Channel** will**:**

- remain unchanged (from the **previous settings**)
- be replaced by values read from the firmware application (using the **firmware scan**)

5 Remote / Over-The-Air-Programming (OTAP)

The Over-The-Air-Programming (*OTAP*) feature allows users to reprogram a *Mote* over a wireless channel. *OTAP* allows one or more *Motes* in the *XMesh* network to receive new firmware images from *XServe* (via the *XOtap* service).

NOTE: *OTAP* is currently available only when the Motes are programmed with a high power firmware application(i.e. XMTS310CB_433_hp.exe).

5.1 OTAP Preparation

Before *Motes* can be programmed over the air, they must be prepared by enabling the bootloader and loading the OTAP image into slot 0. This procedure is outlined in the following steps:

1. From *Settings>Interface Board …* select the appropriate interface board and specify the correct COM port number.

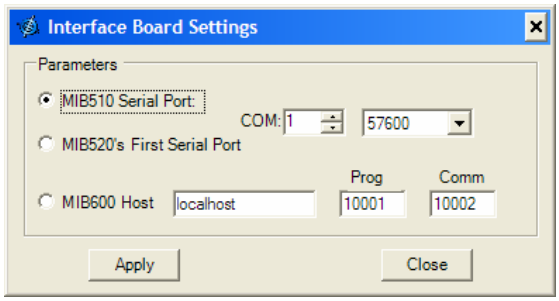

Figure 5-1: Selecting Interface Board Settings

2. Switch to the **Local Program** tab and click **Select** to browse to an *XMesh* application. Choose the appropriate **MOTE ID**, **GROUP ID**, **RF Power**, and **RF Channel**. Make sure that the **OTAP Enable** box is checked. Click on **Program**.

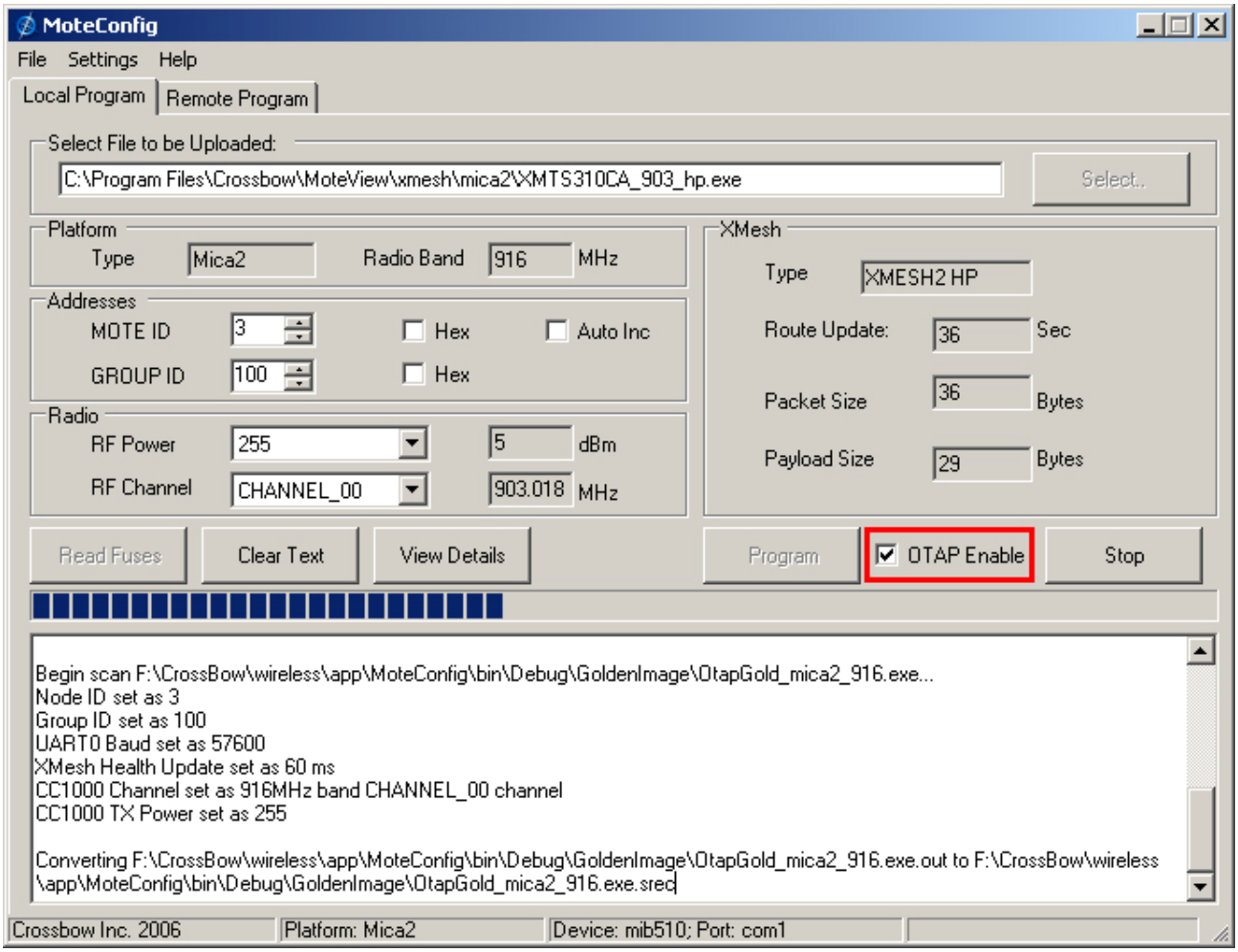

Figure 5-2: Programming the *OTAP* - enabled *XMTS310CA* application

3. Repeat Step 2 for all the nodes in the network. When the bootloader has successfully installed, the LEDs will count up twice when the node is switched "on".

4. Program the base station Mote with the *XMeshBase* application and set Node ID to 0. For the base firmware the **OTAP Enable** box should be unchecked.

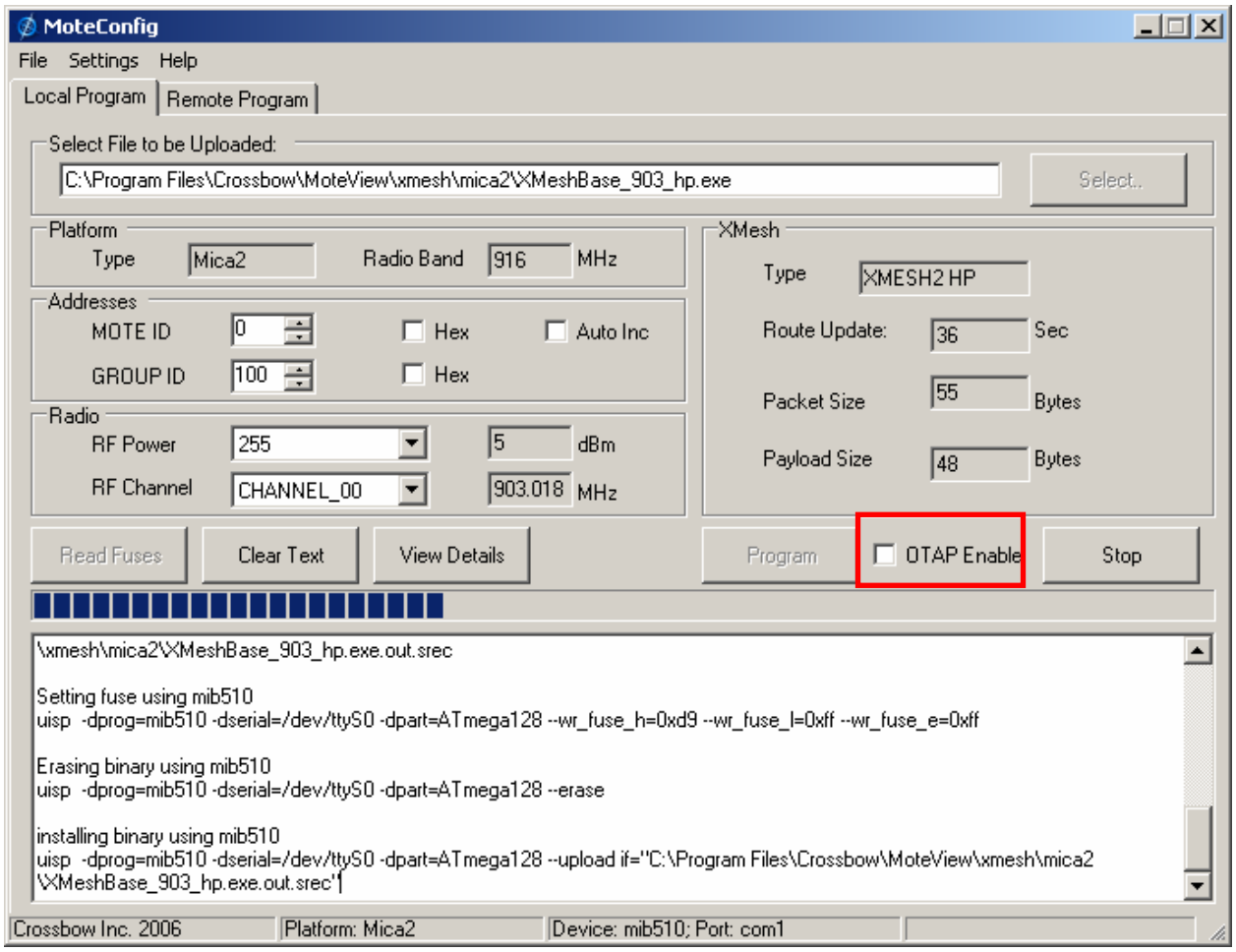

Figure 5-3: Programming the *XMeshBase* application onto the base station.

5.2 Over-the-Air-Programming

Once all the *Motes* are OTAP-enabled, use the following procedure to program them over a wireless link.

IMPORTANT: Please make sure that the Mote battery power is above 2.7V before starting the OTAP procedure.

1. Connect the base node to the PC interface board and turn on the remote nodes that were prepared as shown in section 5.1.

2. Switch to the **Remote Program** tab.

3. Click on the **Search** button to start up *XServe* and listen for remote nodes. The *Motes* found within the network will be displayed in the tree-view control as shown in Figure 5-4.

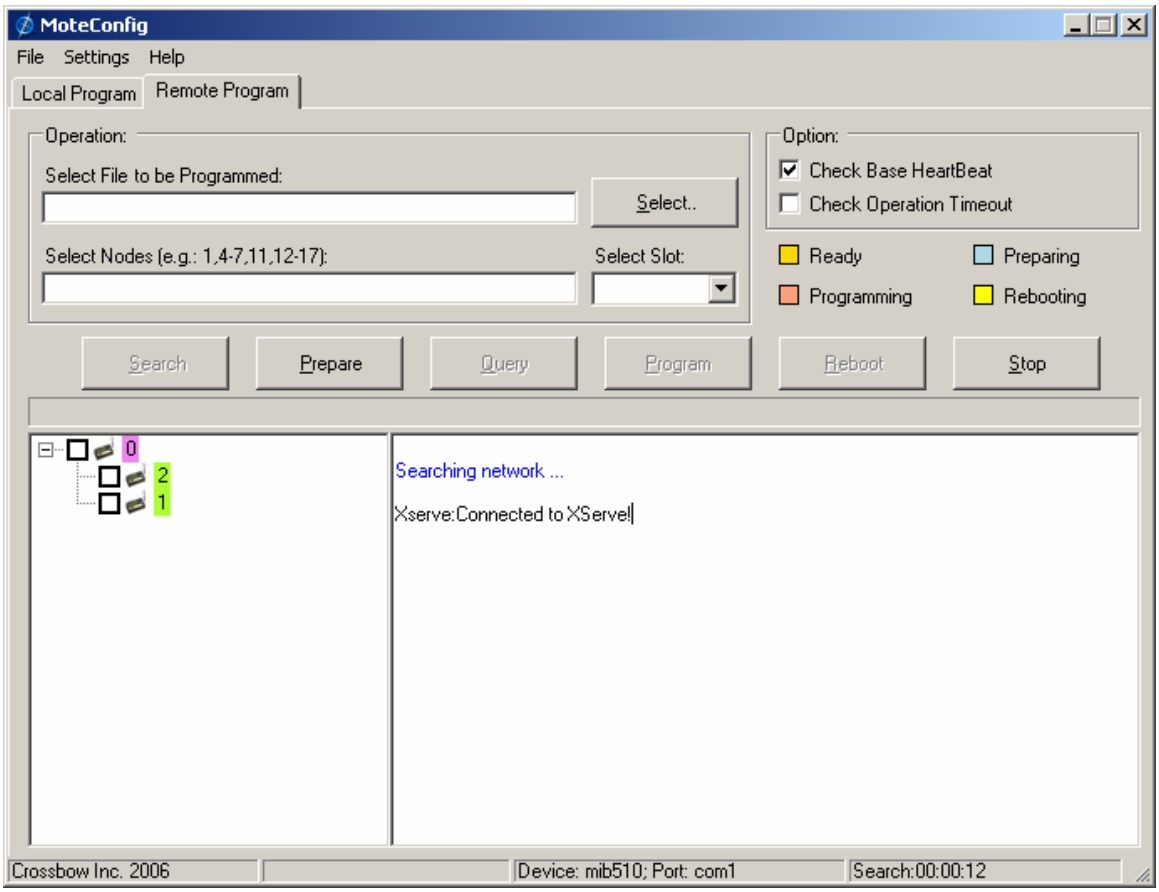

Figure 5-4: Searching for nodes within the mesh network.

NOTE: The base node will periodically blink with a magenta background. This indicates that heartbeat packets sent by the base firmware are being received by the PC. This verifies that the base station has been correctly configured.

4. The Motes can now be rebooted to the OTAP image (OtapGold.exe) by selecting nodes from the tree-view control and pressing the **Prepare** button. Nodes can also be selected by entering their ID's into the **Select Nodes** textbox. During this process, the **Prepare** button will be disabled and the selected node will turn blue.

NOTE: The node ID's entered in the **Select Nodes** textbox override the node selection in the tree-view control.

When the nodes have rebooted into the OTAP image, their background color will turn gold as shown in Figure 5-5.

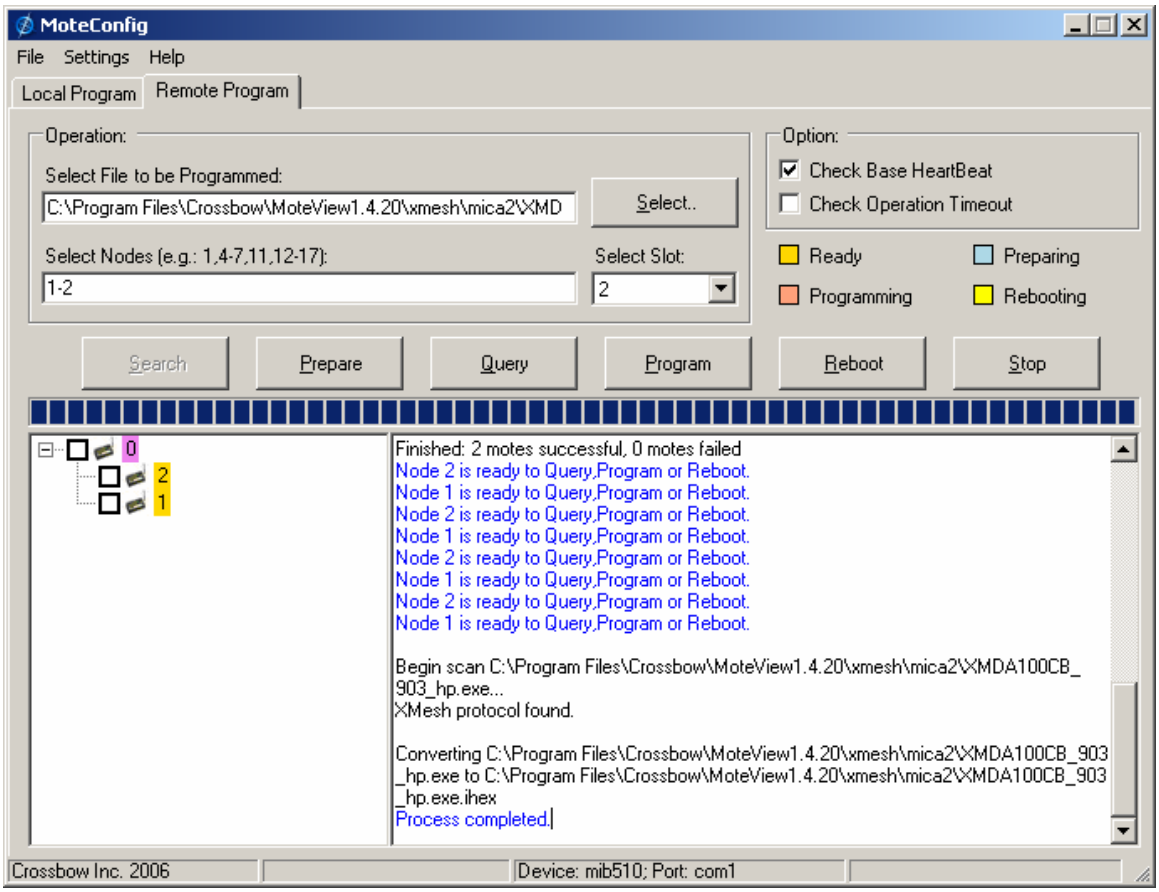

Figure 5-5: Nodes are running the OTAP image.

5. When the nodes are running the OTAP image,
   a) The **Query** button allows users to see the available slots and their contents on each Mote as shown in Figure 5-6.

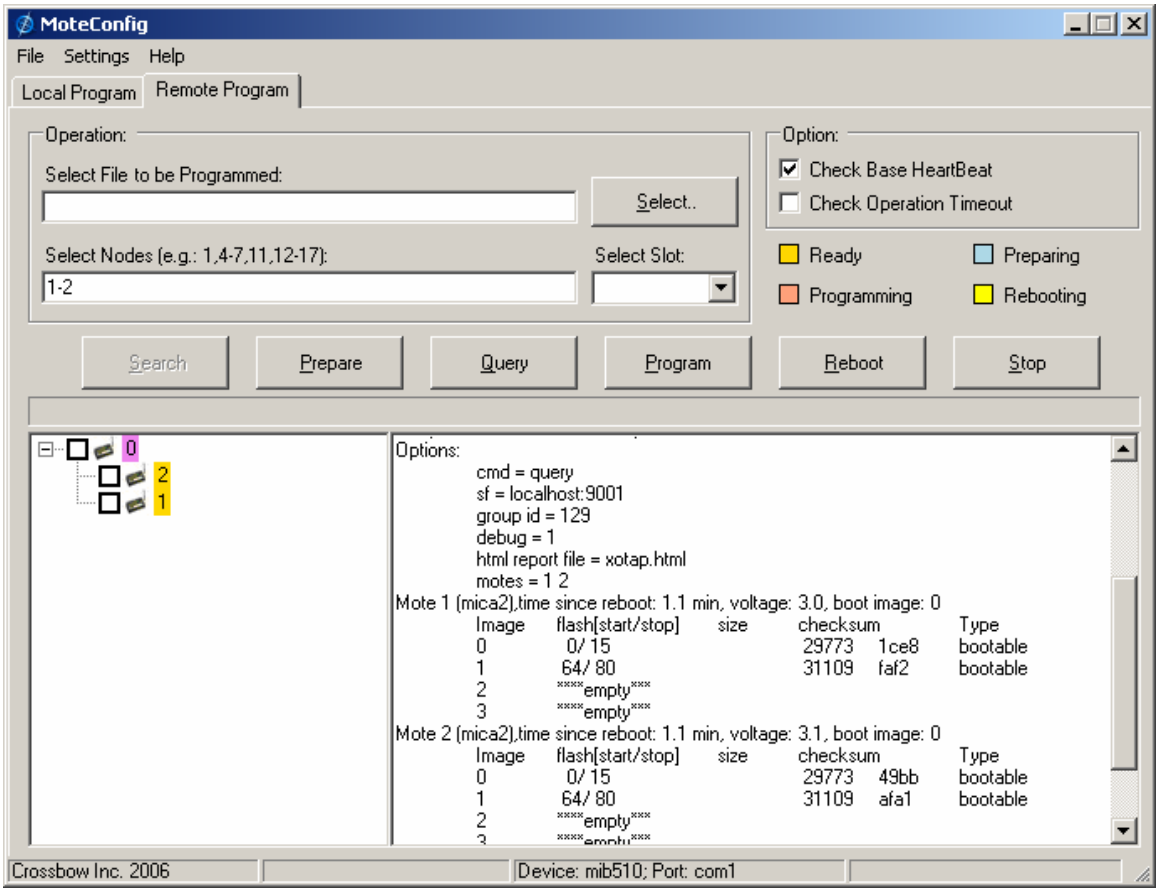

Figure 5-6: Nodes being queried for information.

b) The **Program** button can be used to load firmware images into a selected slot on one or more selected nodes using the following procedure
   1. Select the firmware application, as shown in Figure 5-7
   2. Specify the slot to store the firmware application
   3. Choose the nodes by checking the nodes in the tree-view or by entering the node ID's in the **Select Nodes** textbox
   4. Press **Program**

IMPORTANT: If nodes are specified in the **Select Nodes** textbox, the OTAP operations will only occur on these nodes; the nodes checked on the tree-view will be ignored. The **Select Nodes** textbox is an advanced feature and should be used with care.

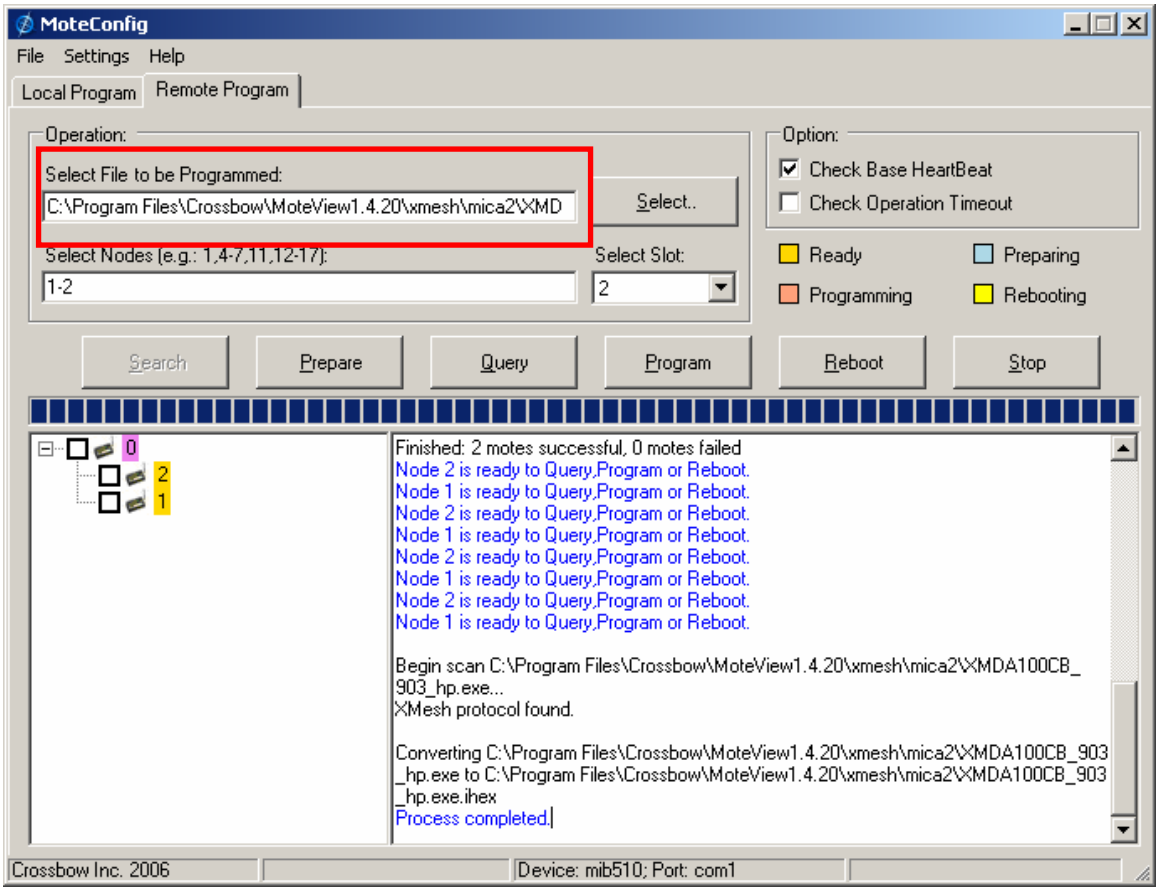

Figure 5-7: Select an application for programming

During the OTAP process, the color of the selected nodes will turn orange as shown in Figure 5-8 and the status message area will display how the number of pages downloaded into the external flash. When the selected nodes have been successfully programmed, the node will turn gold once again.

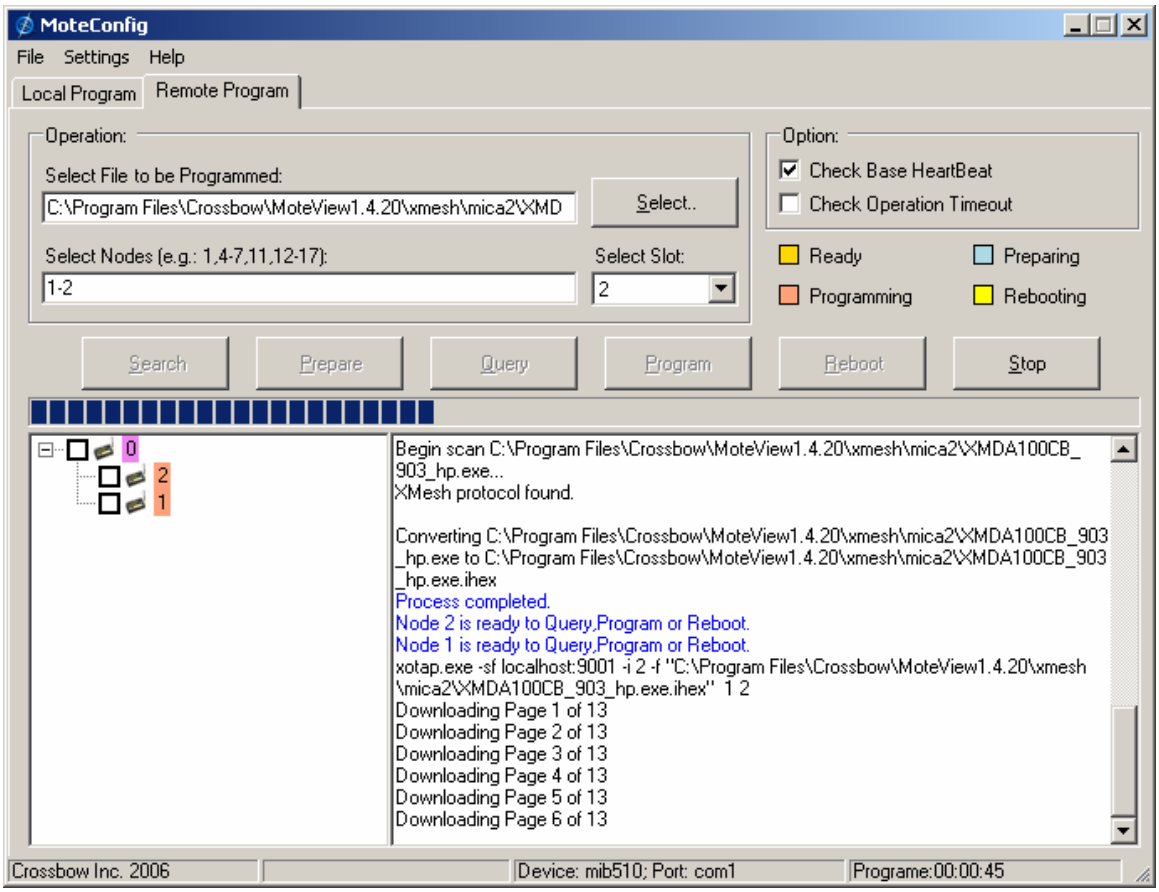

Figure 5-8: Programming the nodes in the network via XOTAP

Open the **Process Messages** window (by clicking on *File > View Process Details…*) to trace all downloading steps as shown in Figure 5-9.

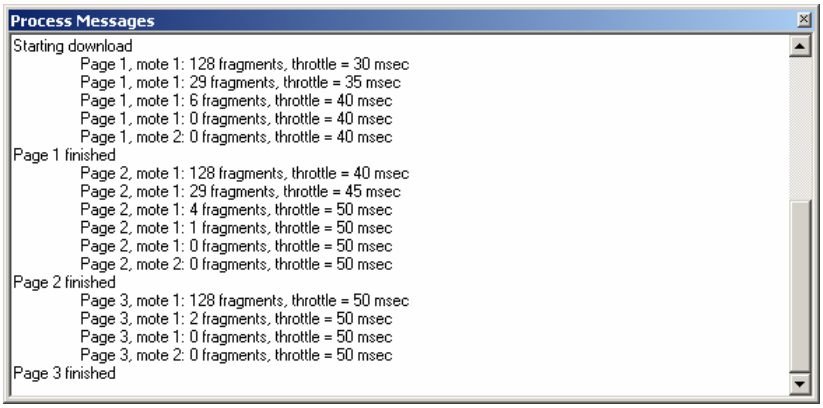

Figure 5-9: Trace all downloading steps

6. The final step is to **reboot** into a newly loaded image:
    1. Select the nodes to reboot
    2. Specify which slot to boot into
    3. Press **reboot**

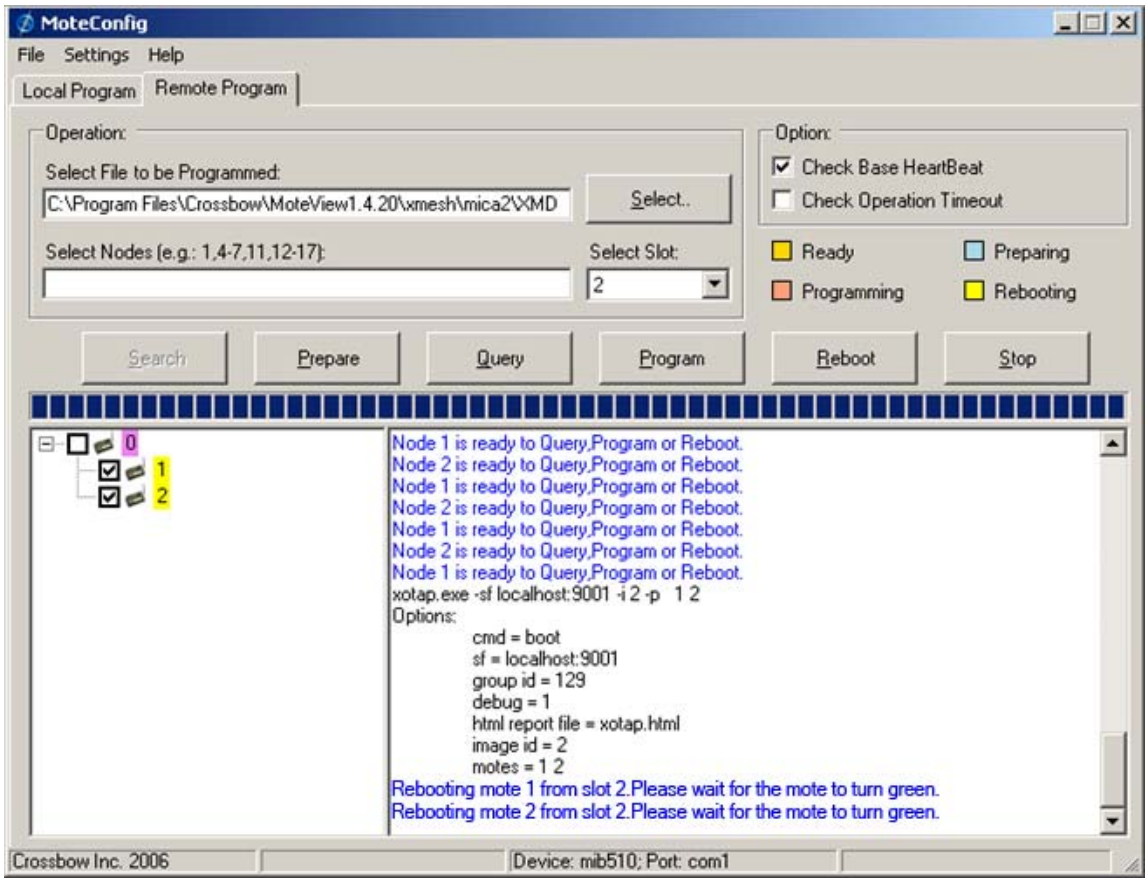

Figure 5-10: Rebooting the nodes into slot 2

The selected nodes will turn green (as shown in Figure 5-11) when they have successfully rebooted, re-joined the wireless mesh and sent health packets to *Xserve*. The nodes will now be executing the newly uploaded firmware application.

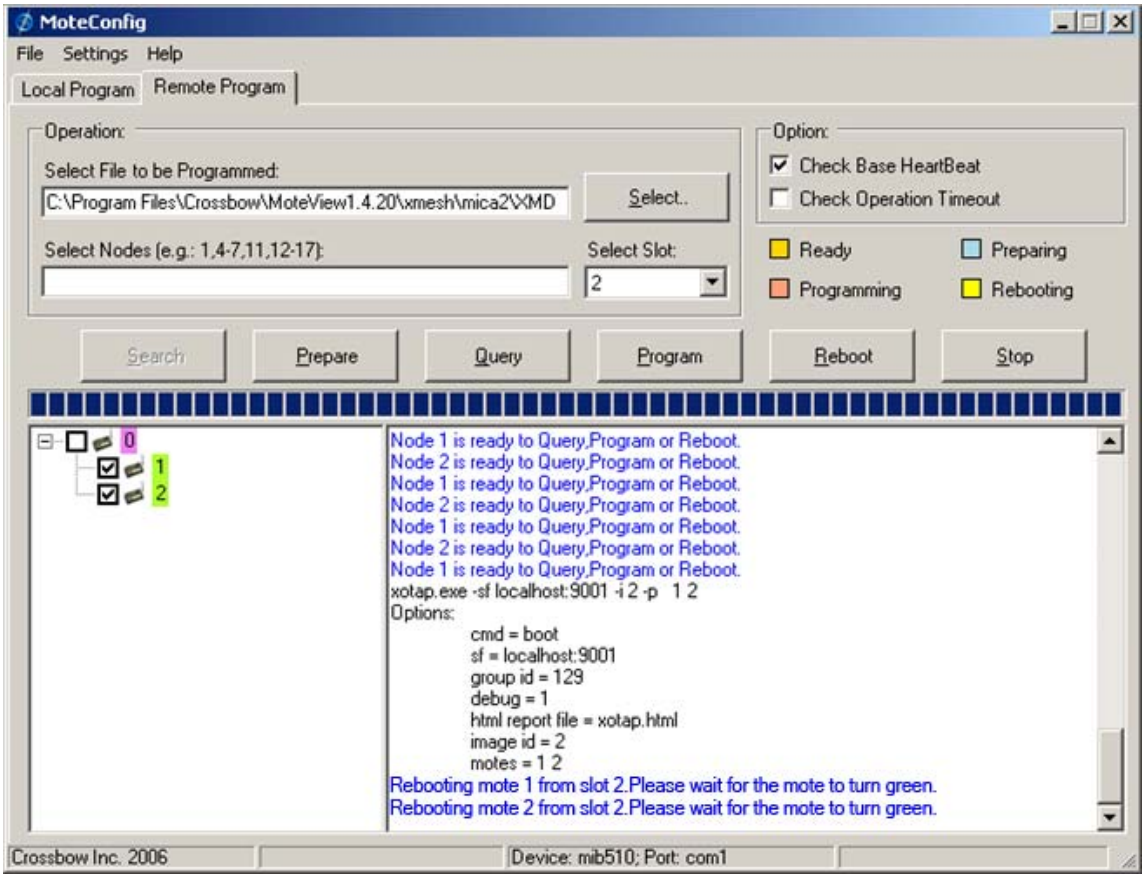

Figure 5-11: Nodes have rebooted and joined the mesh

5.3 Advanced Options

Advanced Options	Description
Check Base Heartbeat	The application will display a warning message if the *XMeshBase* heartbeat packet is not received. This notifies the user that there is a problem with the base station hardware configuration, firmware, power or connectivity to the PC. This feature is **enabled** by default. The user can specify the timeout value in *Settings* > *OTAP Timeout…* > *Base response timeout.*
Check Operation Timeout	The application will detect the time elapsed by each OTAP operation and warn the user if an operation takes too long. If a timeout occurs, Xserve will shut down. This feature is **disabled** by default. The user can specify the timeout values in *Settings* > *OTAP Timeout…*

Click *Settings* > *OTAP Timeout…* to open the **Set Timeout** dialog shown in Figure 5-12. The default values were derived by testing with a small network of about 30 nodes. If your mesh network contains more than 30 nodes, please reset the relevant values or leave the timeout check disabled.

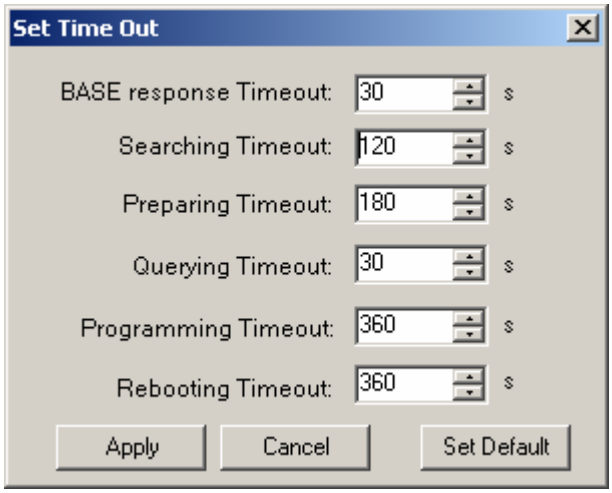

Figure 5-12: OTAP timeout settings

To enable or disable the base heartbeat timeout or OTAP operation timeout, use the checkboxes located on the **Remote Program** tab of the *MoteConfig* application.

Crossbow Technology, Inc.
4145 N. First Street
San Jose, CA 95134
Phone: 408.965.3300
Fax: 408.324.4840