USERS GUIDE

EK2500 Evaluation Kit

for SNAP and Portal Version 2.4

Document Revision v1.7

Wireless Technology to Control and Monitor Anything from Anywhere™

© 2008-2013 Synapse, All Rights Reserved.
All Synapse products are patent pending.
Synapse, the Synapse logo, SNAP, and Portal are all registered trademarks of Synapse Wireless, Inc.
500 Discovery Drive
Huntsville, Alabama 35806
877-982-7888

Doc# 600-0010G

1. Before Getting Started *4*
Other Documentation 4
Other Sources of Information 5
2. Getting Started *7*
Overview 7
A Portal into Your Network 9
Evaluation Kit Hardware 10
Taking the Nodes for a "Test Drive" 14
Time to push some buttons! (Seeing the demo script in action) 16
More about the Hardware 17
Device Configurations 17
3. Installing Portal *18*
Updates on the Web 18
Running Setup 18
Plug in the Bridge Device 22
Launch 25
4. Tutorial *26*
Welcome 26
Navigating within Portal 27
Pull-down Menus 27
Tool Bar 27
Tabbed Windows 27
Rearranging Windows 28
Resizing 28
Closing Tabs 28
Discovery 29
Node Info 31
Upload SNAPpy Image 33
Portal Scripting 37
Radio Range Testing 39
Battery Operation 40
Low Power Operation 41
Where To Go Next 41
License governing any code samples presented in this Guide *42*
Disclaimers *42*

1. Before Getting Started

You've come to the right place: This manual is a tutorial introduction to the Synapse SNAP product line, and you should definitely read through it and try out all of the hands-on examples it contains.

When you have completed this manual, you will have:

- gotten familiar with the included SNAP hardware
- installed the companion Portal software and any needed device drivers
- gotten familiar with the basics of using Portal and SNAP to develop wireless applications

Because it is intended to be a **tutorial**, you need to read through it *in order*, as opposed to skipping around within the document. You also need to *actually do the steps* as specified, because later sections assume the steps from previous sections have been completed.

This manual also focuses on the components actually included in the kit, rather than trying to cover all the different types of SNAP hardware that are available. Just be aware that there exist other types of SNAP-compatible hardware than what you see included in this kit.

Finally, be aware that this manual is a starting point if you will, just one piece of a much larger set of documentation.

Other Documentation

This document, the EK2500 Evaluation Kit Users Guide, is only one of several featured with this evaluation kit. (Several documents are also installed on your system when you install Portal, which you will do as part of this tutorial.) Be sure to also take a look at:

- The "SNAP Primer" (600037-01D)

  This document contains an introduction to SNAP and explanations of how mesh networking works. It also introduces the various Synapse software and hardware available, and clarifies the naming conventions used for the various SNAP items.

- The "SNAP Users Guide" (600025-01A)

  This document is where you will find an explanation of how the components in a SNAP network work together, with introductions to topics like hook handling and with sample scripts.

- The “SNAP Reference Manual” (600-0007N)

  This document is where you will find information on the built-in functions provided by Portal and SNAPpy. It also provides information specific to each platform to which SNAP has been ported.

- The “SNAP Sniffer Users Guide” (600026-01B)

  Starting with Portal version 2.2.23, a “wireless sniffer” capability is included with Portal. If you follow the instructions in this standalone manual, you will be able to actually *see* the wireless exchanges that are taking place between your SNAP nodes.

- The “Portal Reference Manual” (600024-01G)

  This document contains lots of information on how to use Portal, the software that runs on your PC and allows you to configure and manage your wireless network.

Additionally, you may wish to refer to the following hardware guides:

- The “SNAP Hardware Technical Manual” (600-101.01D)

  Every switch, button, and jumper of every SNAP board is covered in this hardware reference document.

- The “End Device Quick Start Guide” (600-0001A)
- The “SN171 Quick Start Guide” (600-0011C)

  These two documents are subsets of the “SNAP Hardware Technical Manual” and come in handy because each focuses on a single board type.

All of these documents are in Portable Document Format (PDF) files and are available on the Synapse Wireless forum website.

Other Sources of Information

There is a dedicated support forum at http://forums.synapse-wireless.com.

In the forum, you can see questions and answers posted by other users, as well as post your own questions. The forum has examples and Application Notes, waiting to be downloaded.

The forum also contains all the latest copies of the documentation included with this kit, plus the latest versions of Synapse Wireless software. Be sure to download the newest version of Portal (which includes the most recent firmware) for the latest feature set.

Also be sure to check out the Synapse website at www.synapse-wireless.com.

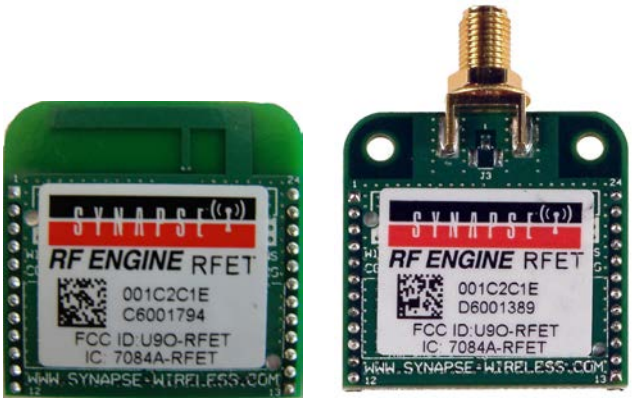
2. Getting Started

Overview

The Synapse ***SNAP*** product family provides an extremely powerful and flexible platform for developing, deploying and managing wireless applications.

SNAP is the name of our network operating system. The word is also used somewhat generically to refer to the entire product line. Often when we are talking about SNAP we are implicitly including SNAP, SNAPpy, Portal, and SNAPconnect.

A *SNAP* network consists of individual SNAP nodes. At the heart of each node included in this kit is a Synapse SNAP Engine, or some other equivalent SNAP Device.



Each SNAP Engine combines a microcontroller, a radio, and an antenna. The antenna can either be an integral "F" antenna, or a mounting point for an external antenna. The exact hardware included in your evaluation kit may vary, depending on the platform around which your kit was built. It may be based on RF100 SNAP Engines (also known as RF Engines), as pictured above, or RF200 or RF300 SNAP Engines. RF100 and RF200 SNAP Engines communicate using 802.15.4 2.4 GHz radio signals, while RF300 SNAP Engines are used for sub-gigahertz communications (900 MHz, 868 MHz, etc.). Additionally, there are EK2500 kits available that showcase the hardware of other vendors (for example Panasonic). This alternate hardware still runs SNAP.

Each SNAP Engine has an on-board microcontroller with its own internal RAM and ROM. No external components are required for operation.

Each SNAP Engine includes General Purpose I/O (GPIO) pins, which can be configured as digital inputs or outputs. Many of these same GPIO pins can also be switched to alternate functionality. The exact number pins and how they can be repurposed will vary from platform to platform, but each SNAP Engine will be able to support:

- analog inputs, providing 10-bit resolution (or better)
- serial data lines (one or two UARTs, depending on platform)
- serial handshake lines (one RTS and one CTS per UART)

Although up to 20 I/O pins are available, you only have to hook up the exact functionality required by your application. The *minimal* hookup to an SNAP Device consists of two wires:

- One wire for VCC (2.7-3.4 volts DC)
- One wire for GND

The SNAP Engines contain core code (written in C) that implements basic wireless networking functionality. This core code also implements a *virtual machine* that executes a subset of the Python programming language. Synapse has named this subset of Python **SNAPpy**.

> You can find details on the SNAPpy language, and how it compares to Python, in the "SNAP Users Guide" and the "SNAP Reference Manual." For now, the important point is to understand that SNAP nodes support a scripting language.

SNAPpy scripts can be uploaded into SNAP Engines "over the air" (OTA), or over the serial interfaces. These scripts define the personality of each node; by changing the SNAPpy script in a node you change the node's behavior.

SNAP Engines can be designed into your own products, acting as a slave device to your main microcontroller or microprocessor. In many cases the SNAP Engine can take over the functionality of the original main processor, in *addition* to adding wireless capability and SNAPpy scripting.

In addition to discrete SNAP Engine modules, Synapse also sells SNAP **demonstration boards** that extend the basic core capabilities with additional I/O hardware.

Three different kinds of SNAP demonstration boards are included in this evaluation kit:

- One SN111 End Device Demonstration Board
- One SN171 Proto Board
- Depending on the hardware platform around which the kit is based it will include ***one*** of the following:
  - One SS200 SNAP Stick USB Dongle
  - One SN132 SNAP Stick USB Carrier Module

Details about each of these boards can be found in the "SNAP Hardware Technical Manual."

A Portal into Your Network

Portal is a standalone software application that runs on any standard PC with Microsoft Windows 2000 or higher[1]. Using a USB or RS232 interface, it connects to any SNAP Engine in the SNAP Wireless Network, becoming a user interface for the entire network.



Note: most of the icons shown in the previous diagram were taken directly from the Portal user interface.

 is used (by default) within Portal to represent a SNAP Node.

 is used within Portal to represent Portal itself.

Once connected, Portal provides the capability to *interactively* build an *intelligent* wireless network. You can:

- Discover new SNAP Devices
- Upload *intelligence* to those devices over the air, using SNAPpy scripts
- Customize Portal to suit your specific application

Interactively – you do all this within Portal, observing the results immediately.

Intelligent – the network is purpose-built for your application, with the ability to monitor and conditionally control things connected to it.

Synapse's ***Portal*** administrative software is included in this evaluation kit.

[1] Portal also runs on the latest long-term release of Ubuntu Linux, and on the latest Macintosh OS. There may be minor cosmetic differences in the UI appearance, but the functionality is the same across operating systems.

For more information about using Portal, refer to the "Portal Users Guide."

Throughout this manual, the term "Portal PC" is used to refer to the PC (Windows, Linux or Macintosh) that Portal is running on. Screenshots in this document show Portal running in a Windows environment, but Portal's functionality is consistent across the platforms.

Note:
Synapse also licenses a standalone *SNAP Connect* library, giving you access into your SNAP network. Import the SNAP Connect library into your Python application to allow your backend systems to participate seamlessly in the SNAP network.

Evaluation Kit Hardware

To demonstrate many of the capabilities of Synapse ***SNAP Engines***, ***SNAP*** and ***Portal***, we've bundled them together in an evaluation kit form – the EK2500.





The hardware included in the standard kit is (clockwise, from upper left) power supply, serial cable, power supply, AA battery holder, SN111 End Device Demonstration Board, SS200 SNAP Stick USB Dongle and SN171 Proto Board. (Depending on the platform on which your kit is based, you may have a SN132 SNAP Stick USB Carrier Module, shown at right, with an additional SNAP Engine instead of the SS200 SNAP Stick USB Dongle.)

The EK2500 evaluation kit comes with software, power supplies, external battery holder, and an RS232 serial cable. Although not shown in the photo, the kit also comes with one pair of AA batteries. The kit also contains *three different kinds* of SNAP boards:

- One SS200 SNAP Stick USB Dongle or SN132 with SNAP Engine
- One SN111 End Device Demonstration Board
- One SN171 Proto Board

Each type of node in the EK2500 evaluation kit has been included to showcase a different set of SNAP traits. Each one has different strengths, and you will likely be building your final wireless network using a *mix* of the different node types. It is also possible to directly incorporate the Synapse SNAP Engines used by these "demonstration" nodes into your own designs.

Furthermore, the demonstration boards in the EK2500 kit provide different approaches to demonstrating how the nodes can be used. In your own hardware designs, it is *not* necessary that you incorporate SNAP demonstration boards in order to use SNAP Engines.

NOTE! Kit contents might be slightly different!

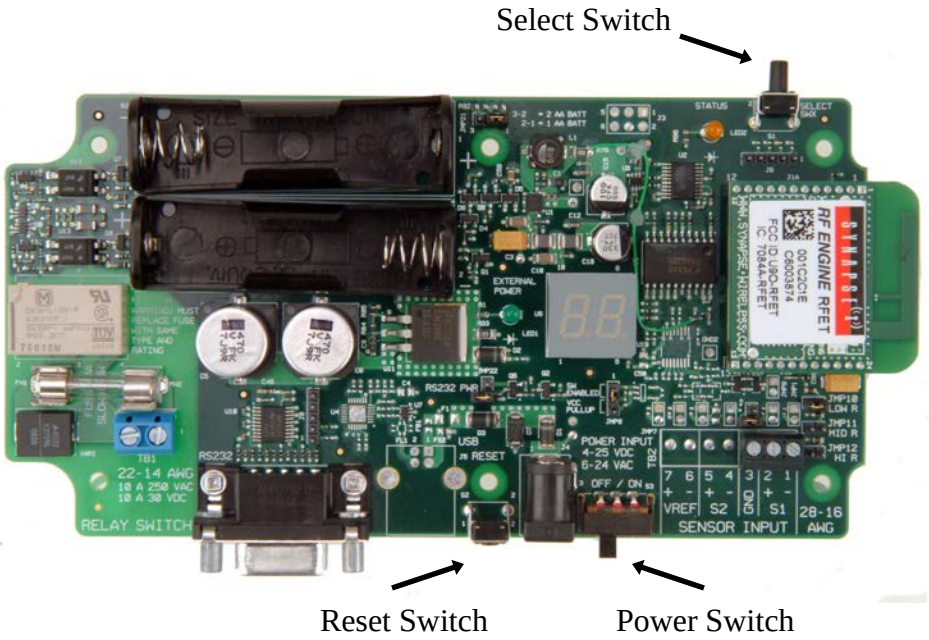
Some older versions of the EK2500 kit included the following hardware:

- One SN163 Bridge Demonstration Board
- One SN111 End Device Demonstration Board
- One SN171 Proto Board

For the demonstrations contain in this guide, if you have an older kit simply use the SN163 bridge board where references are made to the "SNAP Stick".



SN111 End Device Demonstration Board



The Synapse SN111 End Device Demonstration Board features a push-button switch, LED, seven-segment display (two seven-segment digits, 14 segments total), and a generic sensor input.

The sensor input comes with a photocell pre-installed, but other analog input devices can be connected instead.

The SN111 can be powered by an external power supply (4-25 VDC, 6-24 VAC). The SN111 can also be powered by one or two AA batteries.

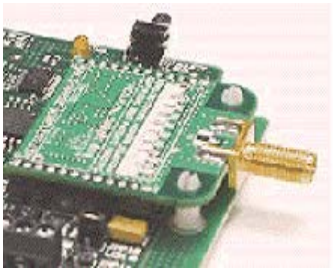
The SN111 does not provide ready access to all GPIO pins. Some are used by the on-board hardware, and other GPIO pins simply are not brought out to any user-accessible connector.

Since we will be talking wirelessly to this node, there is no need to connect an RS232 cable at this time. Do be aware that the RS232 port on this node is functional, and can be used to talk to other RS232 devices (including the Portal PC and even other SNAP nodes).

Refer to the "SNAP Hardware Technical Manual" for more details on this board.

Antenna Connection

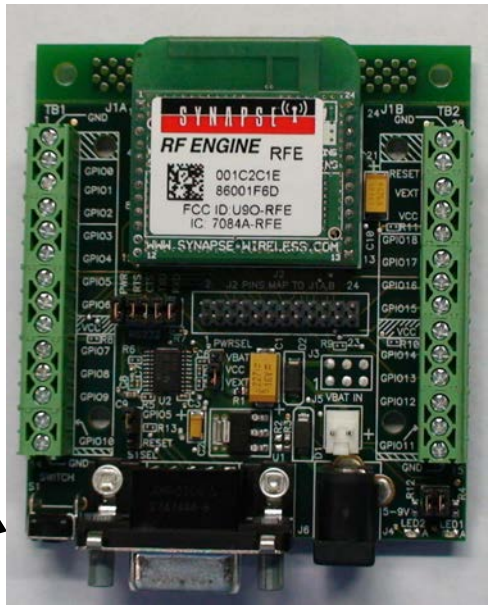
At this time, you'll need to attach the supplied antenna to the SNAP Engine with the Reverse Polarity (RP) SMA type connector (shown below). SNAP Engines *without* the connector come equipped with an integrated "F" type antenna, and don't require an external antenna.





SN171 Proto Board

Select or Reset Switch



The Synapse SN171 Proto (prototyping) Board omits the dedicated seven-segment display and sensor input hardware of the SN111. Instead, it provides easy access to all General Purpose Input/Output (GPIO) pins, including all analog inputs. It comes with a small piezo buzzer attached to GPIO pin 9, but you can remove it if you need that pin for some other purpose.

All of the onboard SNAP Engine signals are made available at easy-access screw terminal blocks located on either side of the node. These same signals are also made available in a more compact form in the center of the board. A dual-row header provides a connection point for a ribbon-cable (or some other form of wiring harness) from some other circuit board or testing apparatus.

The SN171 can be powered by an external power supply (5-9 VDC).

The power supplies included in the EK2500 kit can be used with either of the included nodes, but be aware of the different input voltage ranges if you ***substitute*** *your own* external power supplies.

The SN171 can also be powered by two AA batteries; an external battery holder for this purpose is also included in the EK2500 kit.

The third SNAP Device in this evaluation kit is either an SN132 SNAP Stick USB Carrier Module with a SNAP Engine, or an SS200 SNAP Stick. Either of these plugs into a USB port on your PC to act as the bridge node between Portal (or SNAP Connect) and the rest of your wireless network. (For the remainder of this document, this third SNAP Device will be referred to as your "bridge device" or "bridge node".)

Taking the Nodes for a "Test Drive"

All three nodes come from the factory preloaded with a demo SNAPpy script. Although this demo script showcases only a tiny portion of the capabilities of a SNAP node, it provides a quick way to get familiar with the nodes without having to install any software on your PC. (We'll install the Portal software in a later step.)

If you haven't already done so, **power up** the bridge device by connecting it to a USB port on your PC. (If the "Found New Hardware" wizard appears, just click "Cancel" – we don't want to install that software yet).

Power up the SN111 End Device Demonstration Board by connecting it to one of the two included "wall transformer" power supplies provided in the kit, and sliding the on/off switch to the "on" position.

Power up the SN171 Proto Board node by connecting it to the other "wall transformer" power supply.

> **Note: The SN171 Proto Board does not have a dedicated on/off switch. You simply disconnect the external power supply to turn the unit "off".**

The yellow LEDs on the SN111 and SN171 boards should now be blinking once every second. This shows that the preinstalled demo SNAPpy scripts in all three nodes are running. No LED will blink on the bridge device (though a green power LED will light if you have the SN132 board).

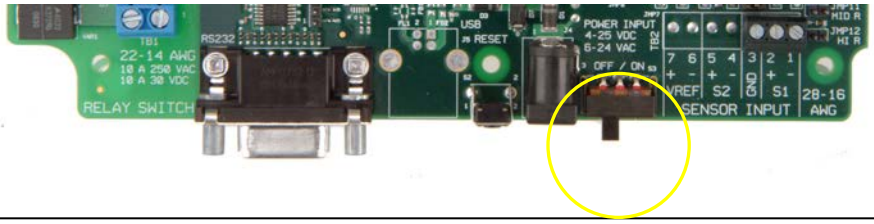
In addition, the green "external power" LED on the larger board (the SN111 node) should be on, indicating an external power supply is connected and providing power.

On the node with the seven-segment display, "00" should be displayed. This shows that the pre-loaded SNAPpy script is running. (It put the "00" pattern on the display.)

If all of your nodes are on and the yellow LEDs are blinking, proceed to the next section. Otherwise, here are some trouble-shooting tips that may apply.

SN111 Troubleshooting

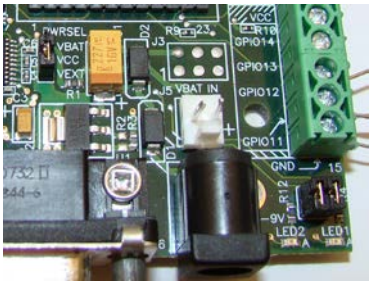
Tip: "Lit" does not always mean "on"
On the SN111 node, the "external power" LED indicates just that – that external power is currently being provided to the board – but it (by itself) does not indicate that the SNAP Engine has actually been powered. You must manually slide the on/off switch (located next to the "barrel" power connectors) to the "on" position.



SN171 Proto Board Troubleshooting

Tip #1: Verify the power-related jumpers:
Your SN171 "proto-board" node should have come from the factory preconfigured to work with an external DC power supply. Still, as a double-check, verify that the PWRSEL jumper is in the VEXT position (connecting pins 2-3), not the VBAT position (connecting pins 1-2).

Tip #2: Verify the LED related jumpers:
The SN171 "proto-board" node should come from the factory preconfigured to enable its two on-board LEDs (one yellow, one green). If the unit is not blinking its yellow LED, it is worth verifying that the LED1 and LED2 jumpers are both installed.



Time to push some buttons! (Seeing the demo script in action)

Using our "multicast" capability, here we use the preloaded SNAPpy script "McastCounter.py" to show that all three nodes can talk to (and hear) each other.

Press the select switch on the SN111 End Device Demonstration Board or the SN171 Proto Board. The seven-segment display on the SN111 should now be showing "01." *The SN171 Proto Board and the bridge device receive the count as well, but have no seven-segment display to display the change. Instead you can watch the lower two bits of the count displayed on LED2 and LED1 on the SN171, and as a cycling tri-color LED on the SNAP Stick.*

Each additional button press should advance the displayed count by one (wrapping around to 00 after the maximum value of 99), *regardless of which node's button you use.*

Press the button on the SN171 Proto Board node to see it advance the count as well. (If this does not function as expected, check the S1SEL jumper on the SN171 to be sure it is connecting the GPIO5 pin rather than the RESET pin.) **Turn off** the SN111 End Device board, and then turn it back **on.** Notice that it is now displaying 00 on its seven-segment display. **Press** the button on the SN171, and the SN111 will now display the new count.

Press <u>and hold for two seconds</u> the push-button on either of the nodes to **reset the count** to zero. The node with the seven-segment display should now show "00".

Unplug the SNAP Stick from the USB port. **Press** the buttons on the remaining nodes. The displayed count on the remaining nodes should still increment with every press of the Select button.

*The "bridge" node can provide the eventual connection to the Portal PC, but it is **not** a "coordinator" of the SNAP network. (there is no central coordinator with SNAP)*

Plug the SNAP Stick back in if you want to see all three nodes interoperating some more.

Some things to notice:

The nodes are immediately able to communicate with each other:
There is no such thing as a "network join time" with SNAP.

Any node can talk directly to any other node:
There is no central "coordinator" node with SNAP.

Using SNAPpy scripts, SNAP nodes can autonomously respond to changes in their environment:

Here you have a user pushing a button, but it could be any digital input, for example, “water detected”, or even an analog input such as “furnace temperature.”

Turn off the nodes when you are ready to begin installing the Portal software.

More about the Hardware

Since the Portal PC (the PC that the Portal software will be running on) has no radio of its own, one of the SNAP nodes must act as a “bridge” for it. Portal will connect *directly* to this bridge node, using either a USB or RS232 connection. Portal will then be able to communicate to the rest of the SNAP nodes *indirectly*, by sending packets *across* the directly connected bridge node.

Verify that the SNAP Stick is correctly connected to the PC’s USB port.

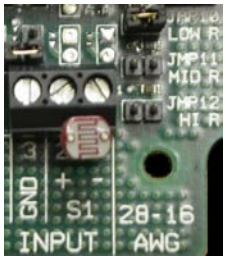
Device Configurations

The following table assumes you have not moved any of your RF Engines between the different boards and does not apply to kits with the **RF300**.

Synapse Network Evaluation Kit RF Engine Configurations:

Description	Function	Antenna	Power Amp	Receive Amp
SN200 SNAP Stick	Portal interface	Integrated	Yes	Yes
SN111 End Device Demonstration Board	Photocell	External	Yes	Yes
SN171 Proto Board	Buzzer	Integrated	Yes	Yes

It is also important to understand that SNAP Engines are not fixed to the boards on which they’re provided in the EK2500 kit. You can move the engines between boards, if you desire. For example, the SN171 Proto Board can be used with an RFET SNAP Engine that has an external antenna.



A *Photocell* component is connected to Sensor Input terminals 1&2 on the **End Device**. Note that the pullup-resistor strap is in the *LOW* range position, with jumper J10 jumped, to properly condition this sensor type.

3. Installing Portal

The Synapse Network Evaluation Kit contains a sheet of paper with instructions for where to download the Portal Installer and SNAP support documentation. This provides all the software you need to get started with SNAP and Portal. Additional updates, and the most current release of this software will always be available on the Synapse support forum.

Updates on the Web

You'll find the latest version of Portal on the Synapse Support Forum at http://forums.synapse-wireless.com (look under Software Releases -> latest Releases).

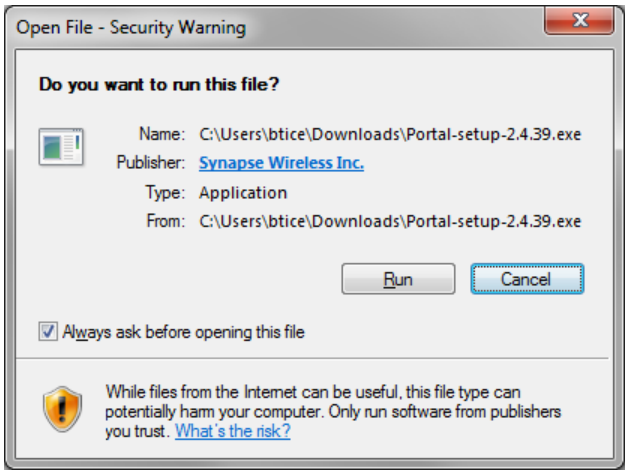
Portal comes bundled with the latest SNAP firmware and documentation.

Also be sure to check the Synapse website at http://www.synapse-wireless.com

Running Setup

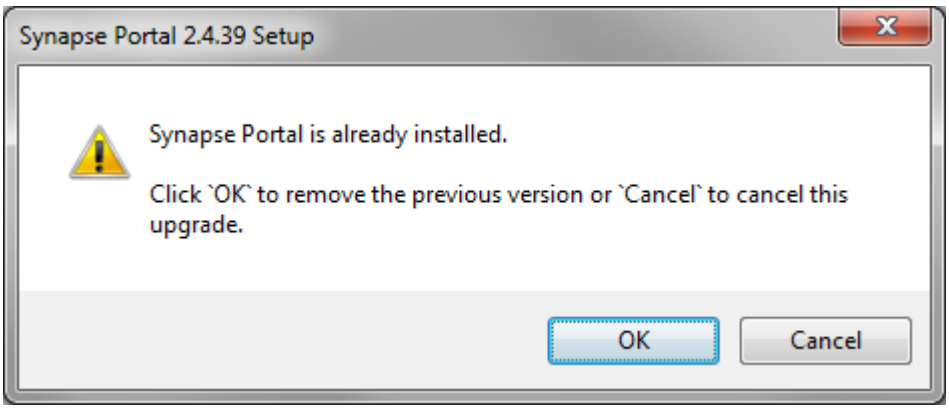
Download and run the Portal installer, Portal-setup-2.4.n.exe (where "n" indicates the latest release version).

Depending on the version of Windows running on your PC, you may get a warning dialog similar to the following somewhere in the installation process:



The warning is harmless, and you should click on ***Run*** or ***Continue Anyway*** to proceed with the installation.

NOTE – if a previous version of Portal (for example, version 2.2.23) is already installed on your computer, you will get an initial dialog box like the following:

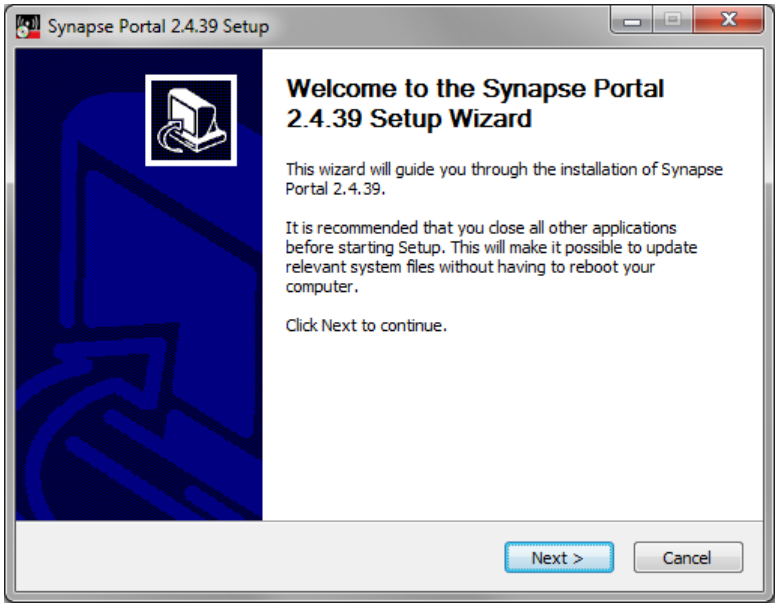


You must click the “OK” button in order to install the newer version.

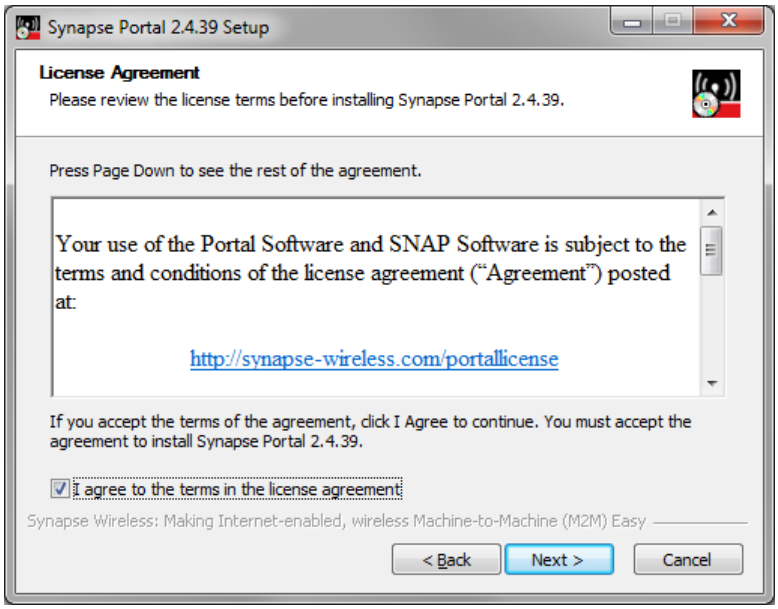
If you are using an operating system other than Windows, follow the installation directions that are provided with your OS’s installer. For the Linux installation, be sure to consult the readme file that is included with the .deb package file.

The following screenshots assume you are installing in Windows. Your precise Portal version number might be different from 2.2.39, but the process should be very similar.

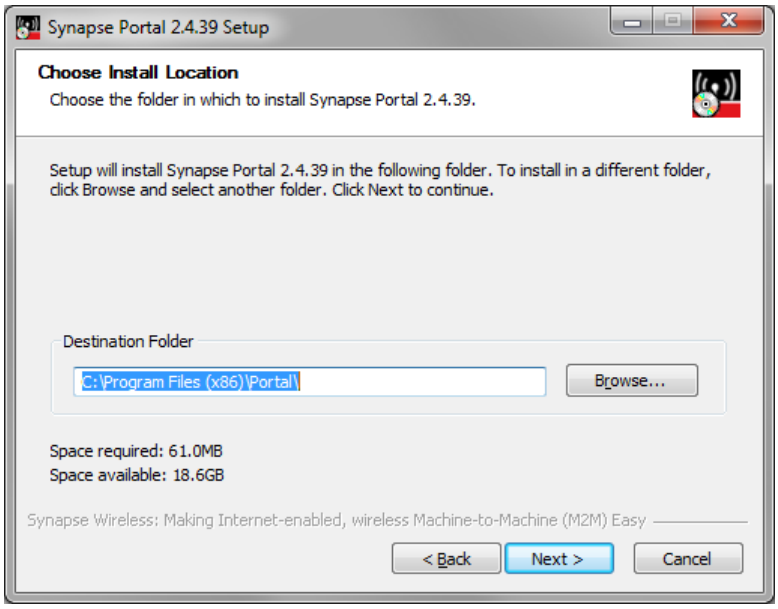
A dialog box similar to the following will appear (your version number will be higher).



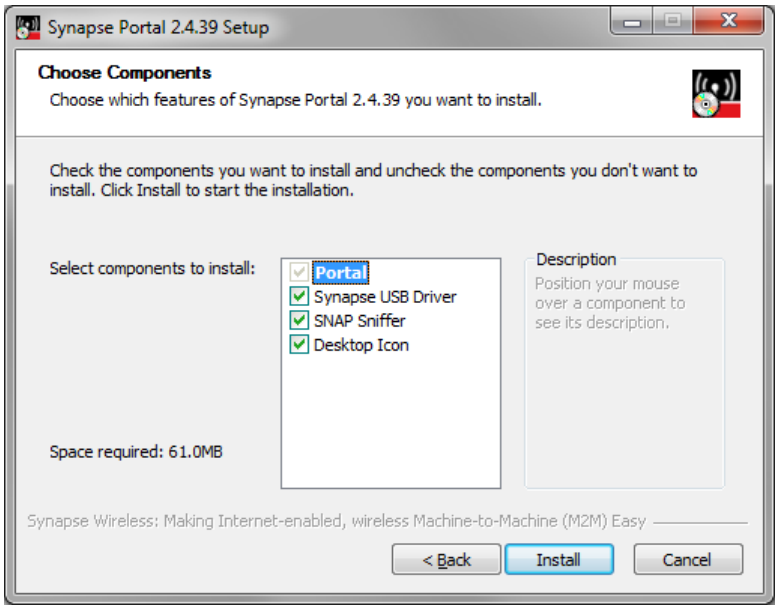
Click the **Next** button to get the following:



Read the license agreement at the specified URL, check the "***I agree***" box and then click on **Next**.

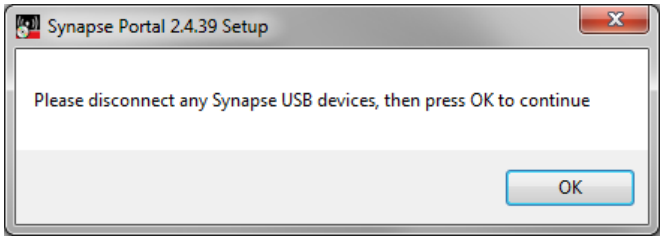


You can either enter the desired destination folder manually, browse to the desired folder, or just click on **Next** to accept the default.

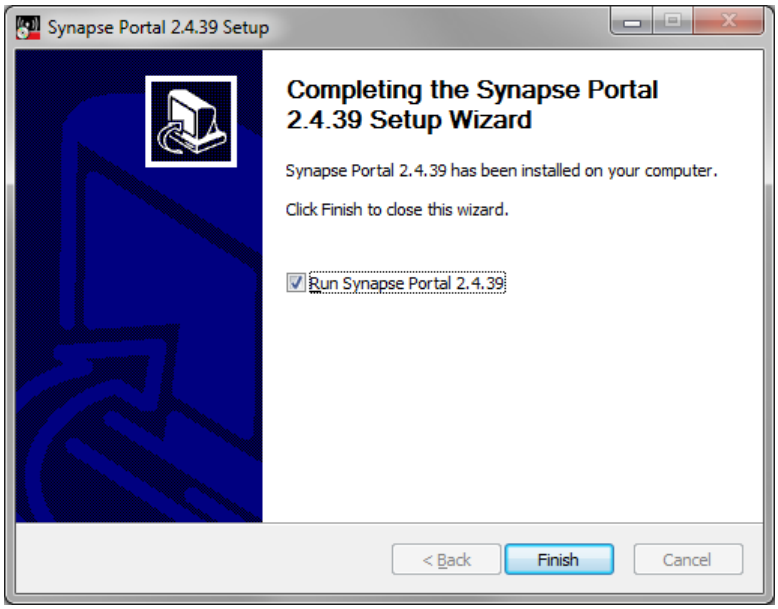


Make sure the desired components are checked, and click on **Install**.

After several files have been processed, if you specified that USB drivers should be installed you will get the following dialog box:



To ensure that the latest Synapse USB drivers can be installed, you must not be running the old versions of these drivers. Disconnect any Synapse USB devices to ensure this, and then click on the “OK” button. The installation process will continue.



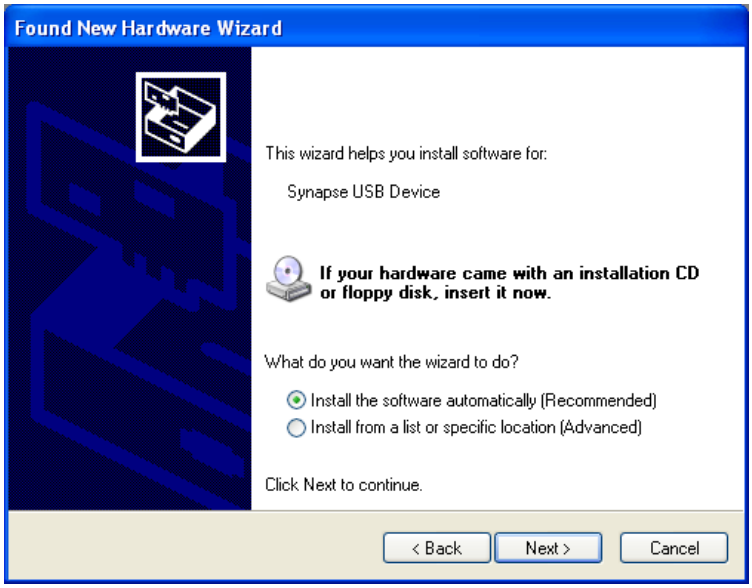
You have now successfully installed ***Portal***. There will be a ***Portal*** icon on your Windows *Desktop* (if you specified there should be), as well as in the *Start Menu*. We recommend that you do ***not*** run ***Portal*** until you have completed the bridge device driver installation through the following steps, so you should uncheck the "Run Synapse Portal" checkbox before clicking "Finish".

Plug in the Bridge Device

The SNAP Stick bridge device should now be plugged into a USB port on the PC on which you installed Portal. Depending on the drivers loaded on your system and which version of Windows you are using, it may be necessary for the Synapse USB drivers to complete installation. This will only occur the first time the bridge device is connected and powered up. If necessary, the following dialog box will appear:

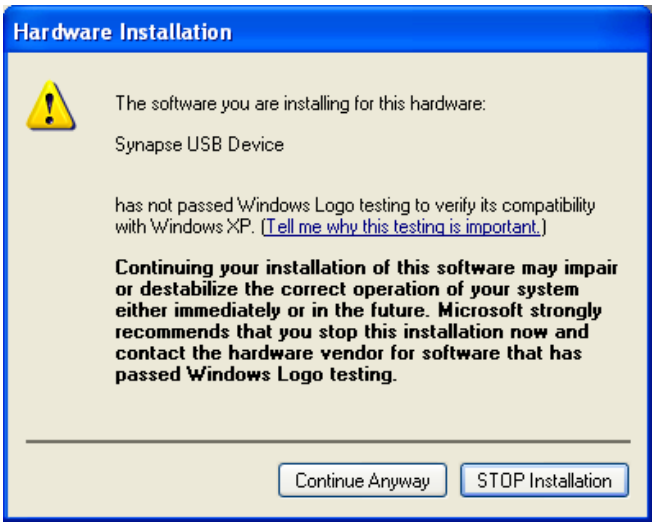


Since the correct software is already available (you just installed it along with Portal), there is no need for Windows to connect to Windows Update – just select "**No, not this time**" and then click on **Next**.

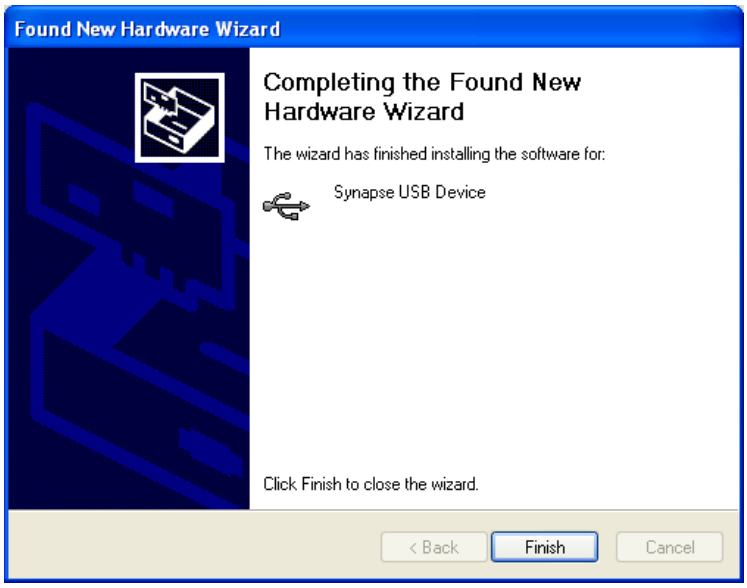


Choose "**Install the software automatically**…" and click on **Next**.

Depending on the version of Windows you are running, you may get a warning dialog similar to the following:



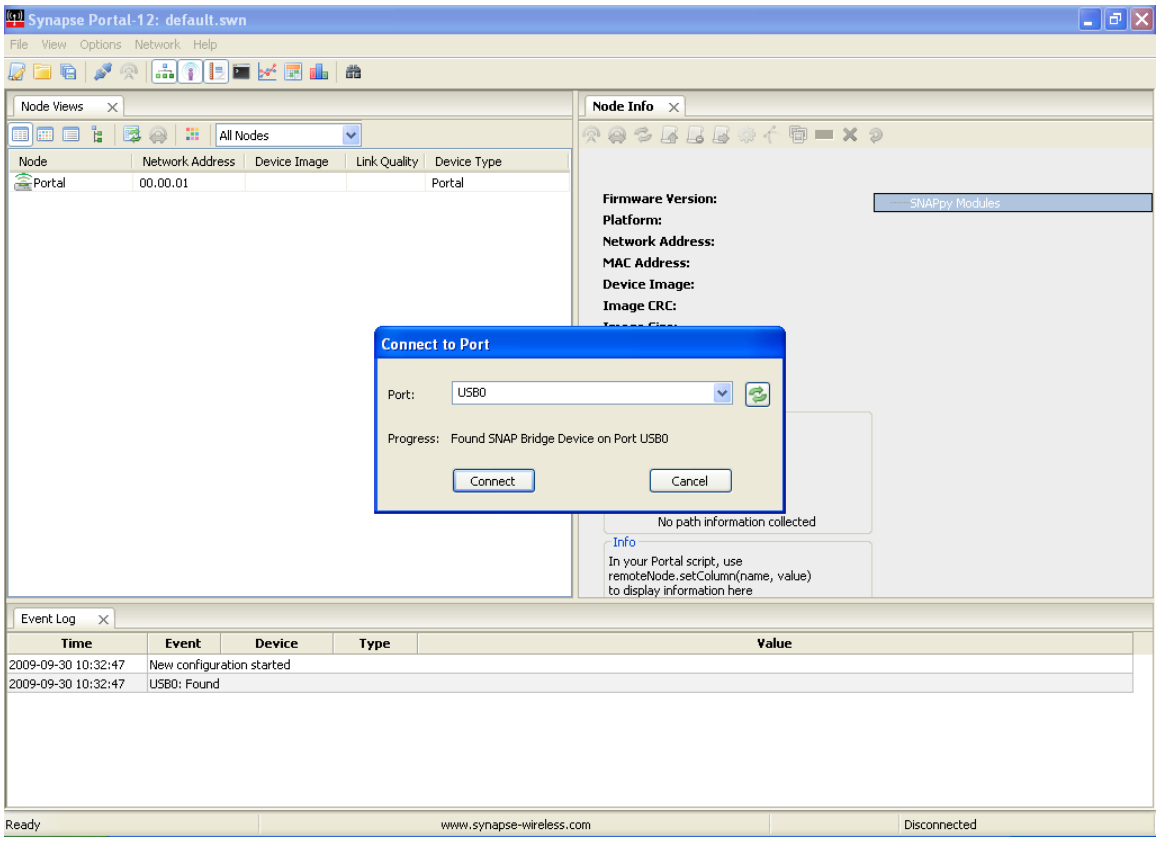
This warning is harmless, and you should click on **Continue Anyway**.



Congratulations! Your *Synapse USB Device* is installed and ready to be used by Portal.

Launch

After installation, launch the **Portal** program. You should see a screen similar to the following:



You have now successfully installed Portal, and detected a USB-connected bridge device.

Note – The "Connect" dialog box is automatically shown at Portal startup. If you click the **Cancel** button, you can *bring this dialog back* by clicking the button on the main toolbar.

Also be aware that this toolbar button doubles as a status indicator. When you *are* connected, it looks like , and functions as a *disconnect* button. When you *are not* connected, it looks like and functions as a *connect* button.

4. Tutorial

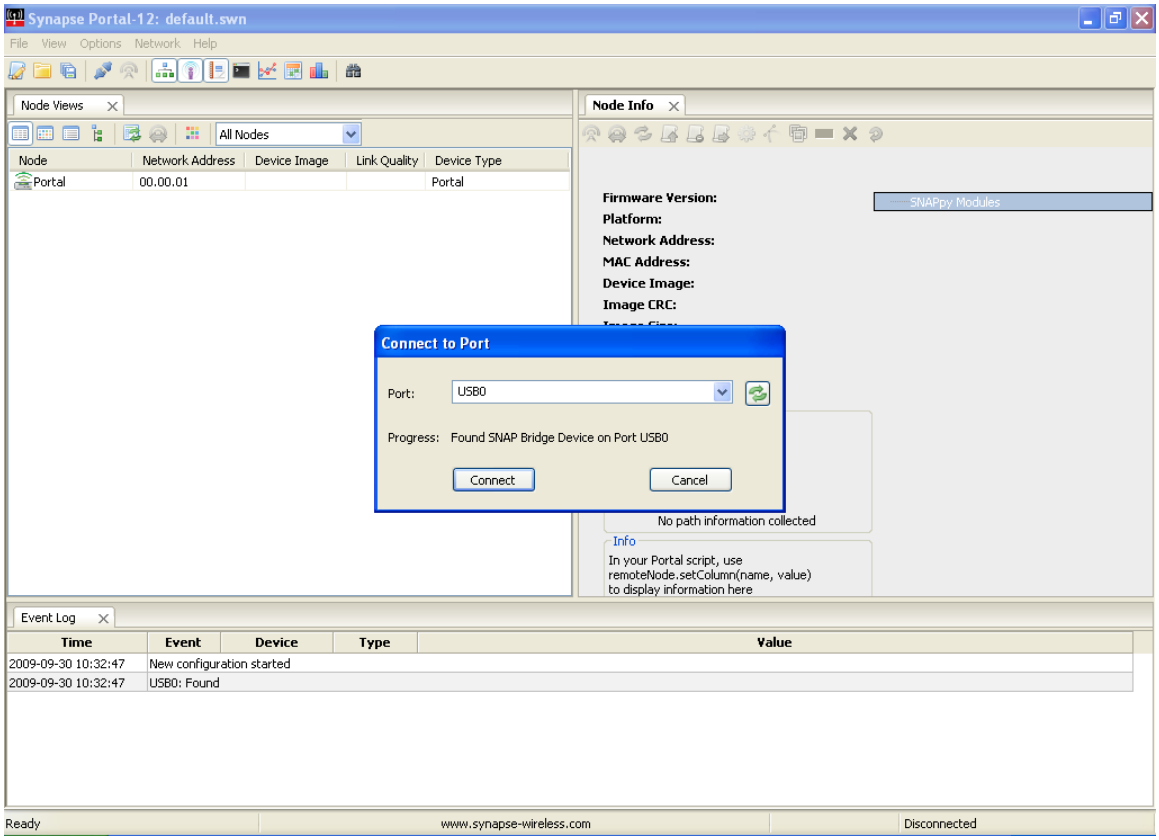
This section is a snapshot of wireless application deployment using SNAP and Portal. In the following example, we have three SNAP nodes present: a bridge device connected to the PC via USB (either the SS200 SNAP Stick or the SN132 SNAP Stick USB Carrier Module with an RF Engine), and two remote devices – the SN111 board and the SN171 board. Ensure all three nodes are powered on.

The SN111 board has a photocell connected to GPIO 18 (Analog Input 0), and its **Device Type** has been set at the factory to “Photo”.

The SN171 Proto Board has a small piezo buzzer connected to pin GPIO 9, and its **Device Type** has been factory set to “Buzz”.

Welcome

Starting Portal for the first time brings up a blank network configuration and a dialog asking what port to use to connect to the SNAP bridge device:



Make sure the SNAP Stick is plugged in, and the USB device should be found immediately. Press **Connect** once the SNAP bridge device is found.

Navigating within Portal

Portal is straightforward to use. However, since it is extremely customizable by the user, your screen layout may not always match the screen shots in this manual. For that reason it is important to understand the fundamental concepts used in navigating the Portal GUI.

Pull-down Menus

At the top of the Portal GUI are pull-down menus for **File**, **View**, **Options**, **Network**, and **Help** operations. (These menus are in their standard position at the top of the screen in Macintosh and Ubuntu versions.)

Clicking on one of these top-level menu choices will pull down a sub-menu of additional choices. For example, clicking on **Network** will present a sub-menu from which you perform actions like **Broadcast Ping**, **Find Nodes…**, or **New Configuration**. Similarly, clicking on **Help** will bring up choices for **SNAP Reference Manual**, **Portal Reference Manual**, etc.

The convention is that menu choices ending in "…" usually bring up additional menus or dialog boxes, and menu choices not ending in "…" cause immediate action to be taken, with no further prompting.

Tool Bar

Below the pull-down menus is a horizontal Tool Bar from which you can initiate several actions. Hovering the cursor over each button will display a short "tool-tip" help message, and clicking on each button will initiate the action displayed by the tool-tip.

Tabbed Windows

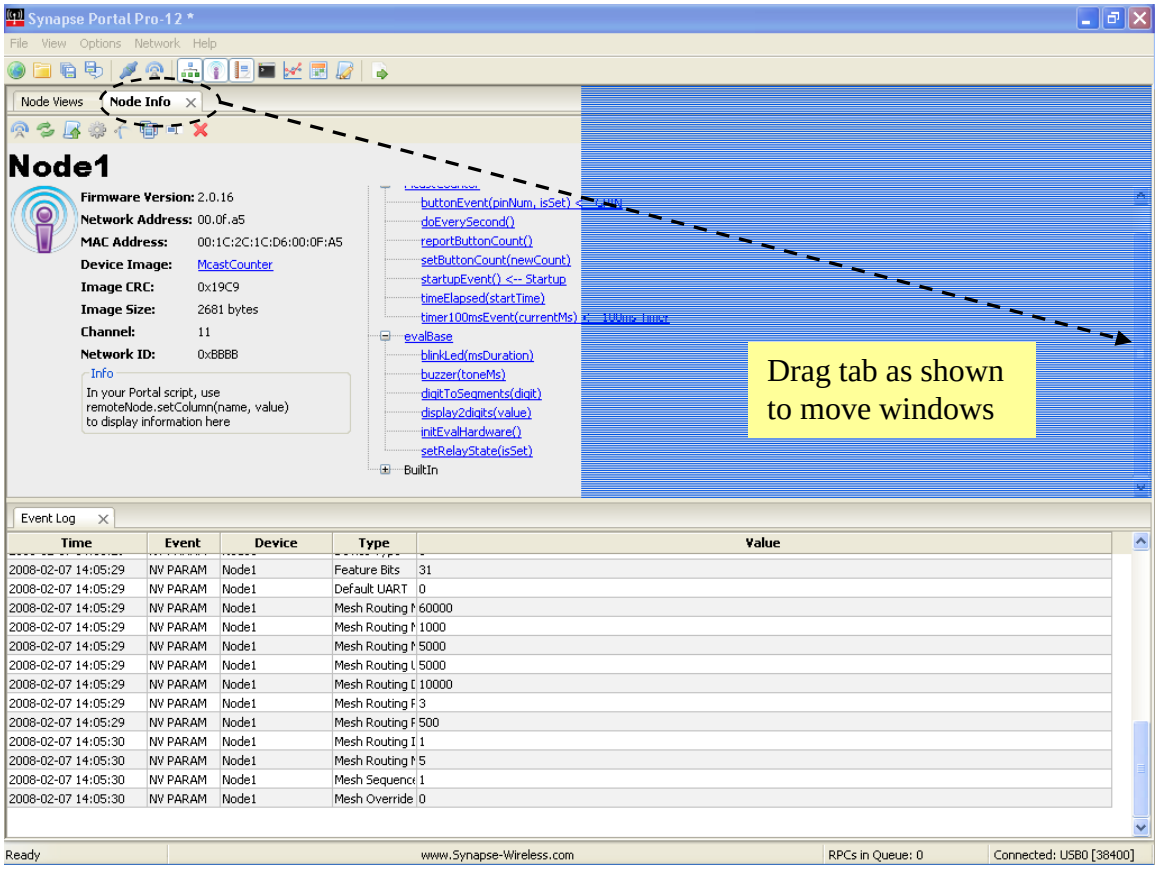
The remainder of the Portal GUI is taken up by a changeable collection of tabbed windows. Each of these windows has a name, which is displayed in the tab for that window.

Many of the tabbed windows have toolbars of their own, located in the horizontal region just below their labeled tab.

Portal starts out with an initial set of tabbed windows visible. Sometimes clicking on certain controls within one tabbed window will open and/or switch to another tabbed window. You can also open additional tabbed windows by choosing them from the **View** menu. Finally, many of the tabbed windows can be launched from the main tool bar.

Rearranging Windows

The tabbed windows in Portal can be dragged and repositioned on the screen. To do this, press and hold the left mouse button while the cursor is positioned over the tab label you want to move. While holding the button down, drag the tab until you see a light blue "shadow" indicating a possible new position for the window. When you've found a suitable new position, just release the mouse button and the move will be complete.



Resizing

Windows may be resized by clicking and dragging the horizontal and vertical borders separating them.

Closing Tabs

You can close tabbed windows that you no longer want by clicking on the small "X" located to the right of the name in the tab.

Now that you know the basics of navigating within Portal, we can continue with the tour.

Discovery

We first need to look at the **Node Views** tabbed window. If this window is not already open, you can click on **Views**, then choose **Node Views window**. Alternatively, you can click on the icon on the toolbar.

You will notice that the **Node Views** window has its own toolbar. Ignore all but the first four buttons for now.



The **Node Views** tabbed window lets you look at your nodes in four different ways:

- **Report View**
- **Icon View**
- **List View**
- **Tree View**

All four views are just that, "views" of the same network information. Click on the **Report View** button.

You should see that three Devices have been discovered. They all will have names of the form McastCounter***X***. (***X*** in this case refers to a trailing digit that will be present for any node after the first one found.) The directly connected "bridge" device should be shown in blue, while the two remote devices should be displayed in black. Because the nodes report in using a random response delay, which node gets to be McastCounter, McastCounter2, or McastCounter3 can vary.

When you double-click on a node in one of the **Node Views**, Portal displays basic information about that Node in a separate **Node Info** pane.

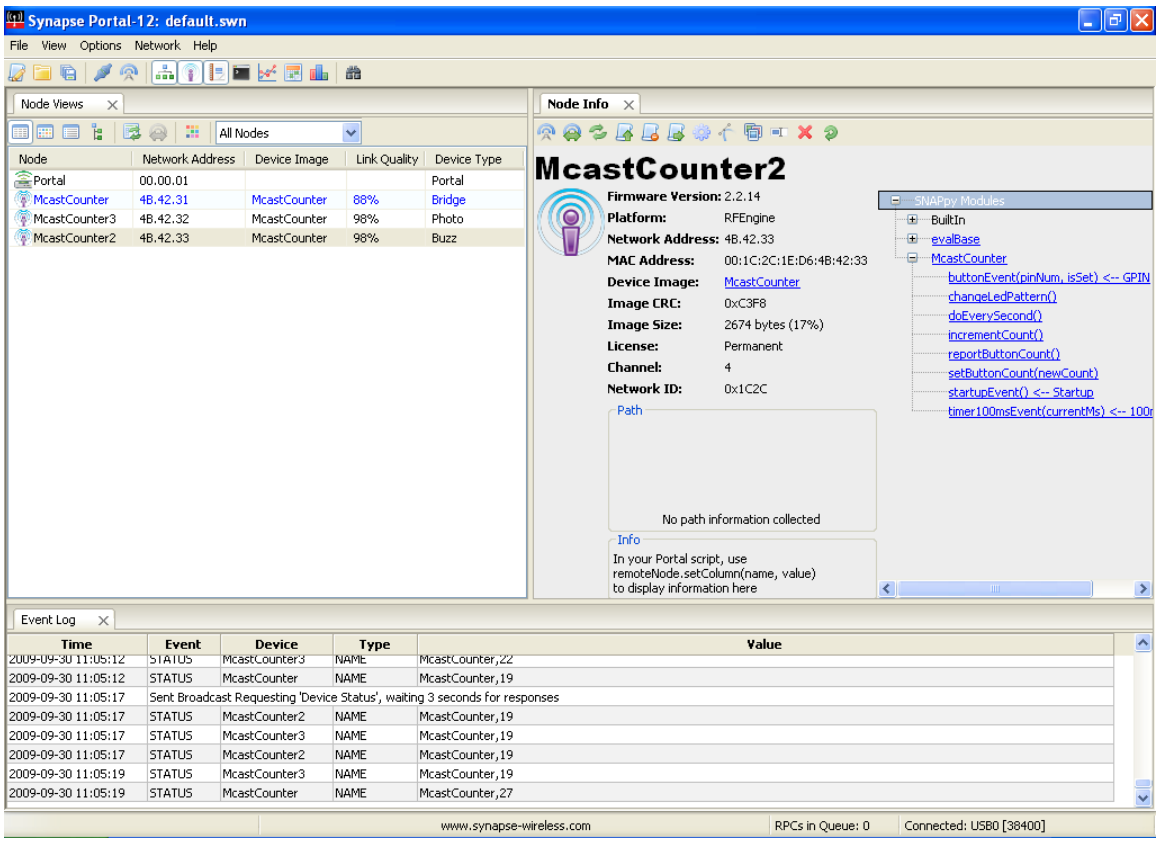
RF300 Usage Notes

Tip #1: Manually set the device type:
Most RF Engines running a demonstration script will automatically set their Device Type (NV Parameter 10) to reflect the Synapse demonstration board on which they are currently mounted. This is not the case for RF300 engines due to a slight pin-out difference.

Users must set each node's "Device Type" to match the following list:

Device	SNAP Device Type Setting
SN171 Proto Board	"Buzz"
SN111 End Device	"Photo"
SN132 SNAP Stick	"Stick"

The Device Type setting can be accessed by selecting the "Change Configuration Parameters" option, then the "Device" tab, from the Node Info pane toolbar in Portal.



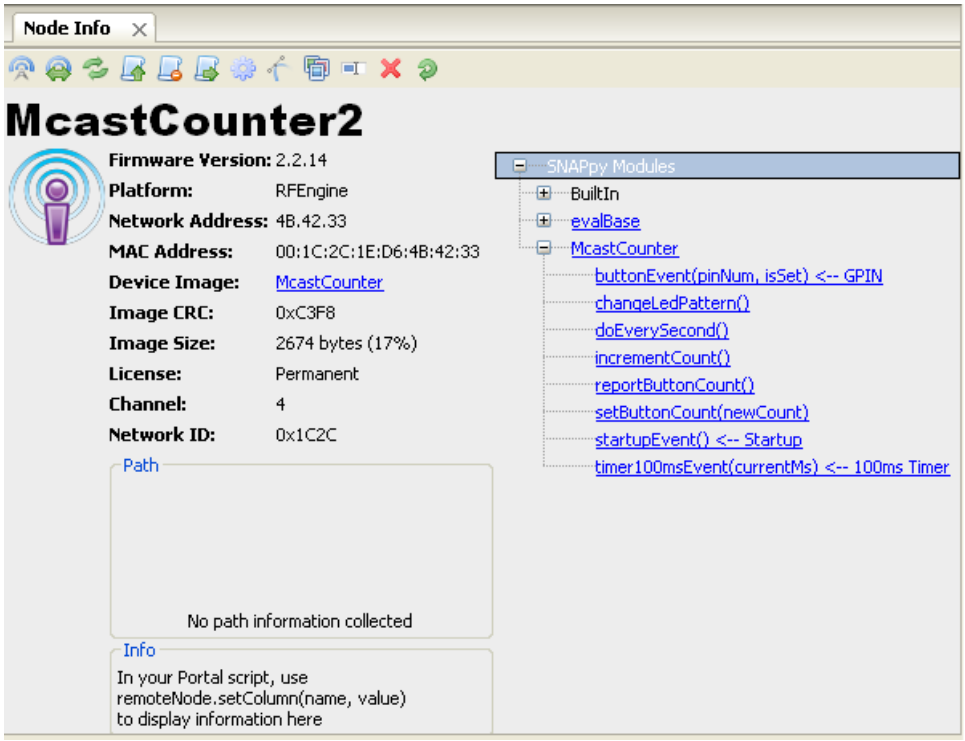
Note that the node names are based on the names of the SNAPpy scripts loaded into those same nodes *at the time of discovery*, but with additional trailing digits (2, 3, 4) added to enforce uniqueness.

Since these nodes were pre-loaded at the factory with the “McastCounter.py” script, their names are of the form McastCounterX, where “X” is replaced by a number. If the nodes had been pre-loaded with some other script, then you would have seen that script’s name in place of McastCounter.

If the three nodes had not been pre-loaded with scripts at all, then their names would have been “Node”, “Node2”, and “Node3.”

Node Info

Lots of information is shown in the **Node Info** tabbed window. However, Portal may not be set to automatically query the node for its information. (This is a configurable preference in Portal.) To be sure Portal knows everything important about your node, click the “Refresh Node Information”  icon in the toolbar that runs across the top of the Node Info tab.

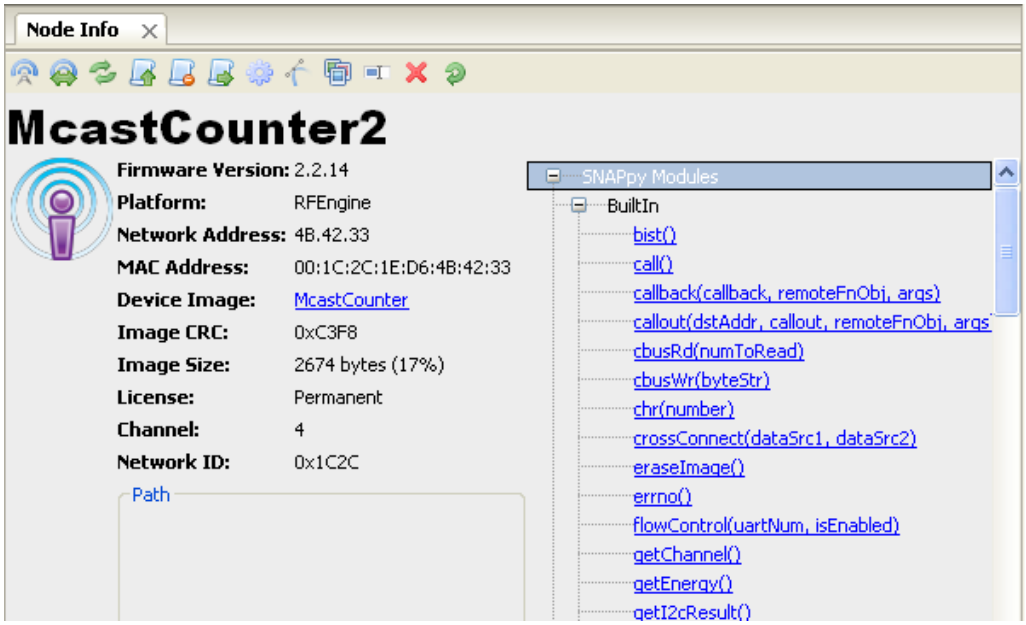


On the left-hand side, the **Firmware Version**, **Platform**, **Network Address**, **MAC Address**, **Device Image**, **Image CRC**, **Image Size, License, Channel**, and **Network Id** are shown. Below that is a block where **Path** information (the path to/from the node) can be displayed. Below that there is an **Info** field that can be controlled from Portal scripts to add your own custom field(s) to the **Node Info** panel.

Device Image refers to the SNAPpy script (also referred to as a SNAPpy image) loaded into the node. Here you can see that the script/image “McastCounter.py” has been loaded into the node. Also note that you can click on the device image name shown

(McastCounter), and automatically bring that script up in Portal's built-in source code editor.

On the right hand side, a collapsible tree of available functions is shown. In this next screenshot, you can see the BuiltIn tree (the tree of built-in functions) in expanded form.

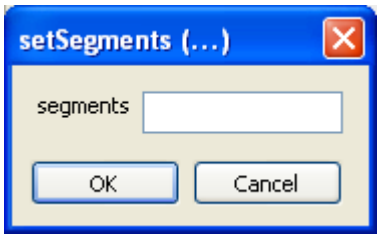


Notice that there is a scroll-bar on the right-hand side of the pane – there are too many built-in functions to fit on the screen at one time.

Hovering the cursor over a function name will display a tool-tip for that function. More importantly, you can click on any function to invoke that function **directly on the selected node**.

Functions that do not require any parameters (for example, the reboot() function) will be executed immediately.

If the function requires any parameters, Portal will automatically prompt you for them.

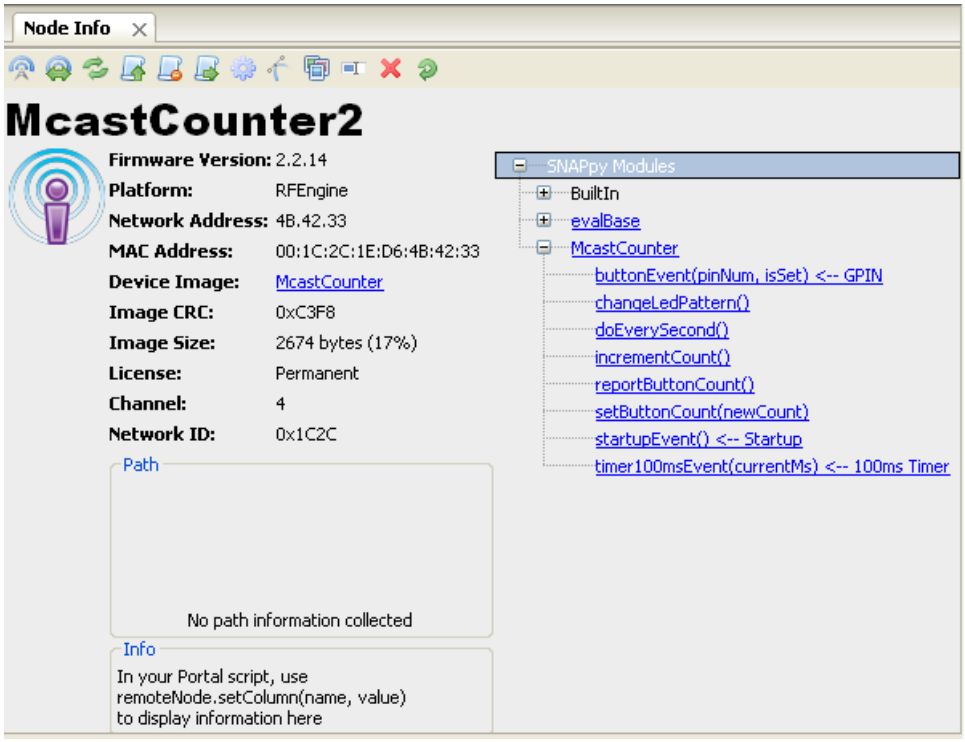


For example, clicking on the setSegments() function will prompt you to enter the actual "pattern value" to be put on the seven-segment displays.

You can either:

1) Enter a value (for example 0x4040) and press OK, or
2) You can press Cancel to abort the function invocation.

You can also expand the tree of functions defined by each module (in other words, by each SNAPpy source file).



Here you can see the various functions defined in the McastCounter.py SNAPpy script. Like the built-in functions, these can also be directly invoked by clicking on them (and entering any needed parameters).

The **Node Info** tabbed window also has its own toolbar. Most of the toolbar functions will be discussed later, but one is of particular importance to us now: "Upload SNAPpy Image."

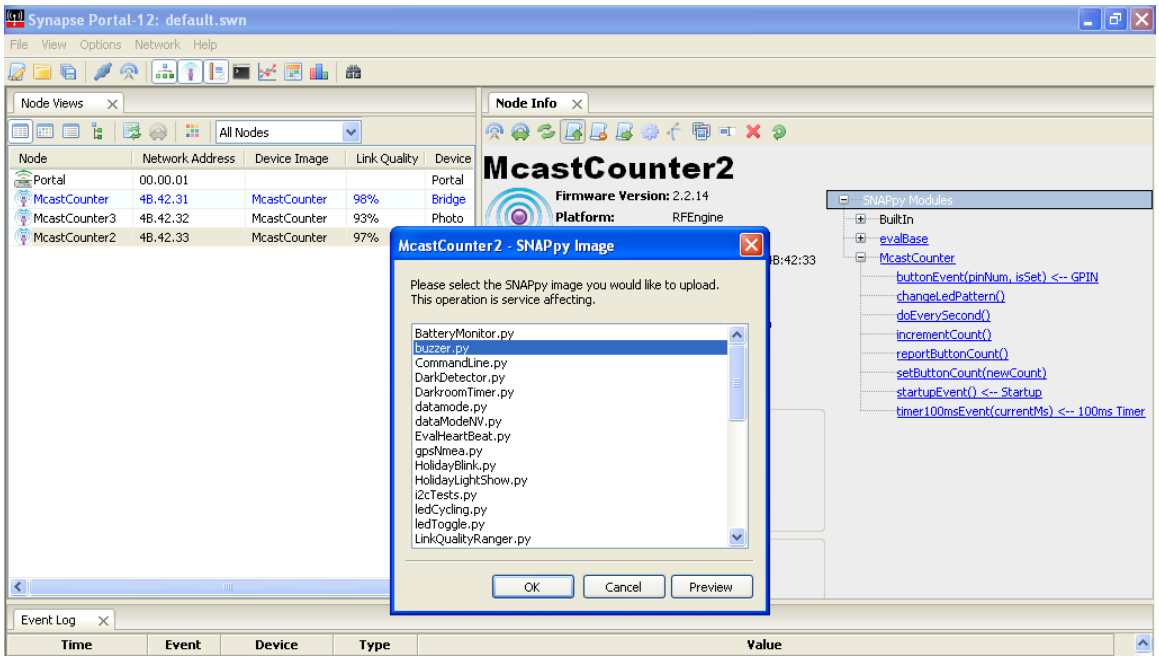
Upload SNAPpy Image

The "multicast counter" script was preloaded merely as a convenience to the user. You can overwrite these default scripts with other scripts from the set of example scripts included with Portal, or even with your own custom scripts.

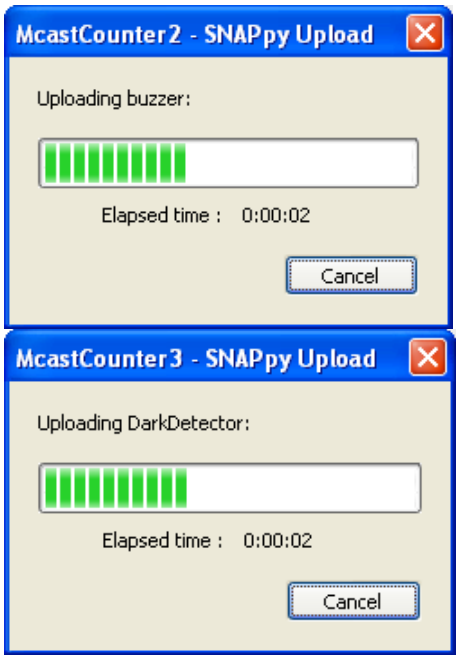
We've already tried out the "multicast counter" functionality in section 2, so now let's override that behavior with a different one. To do that, we will assign new "behavior" to two of the nodes, by giving each one a new SNAPpy Image.

Select the node with Device Type "Buzz" in the *Node Views* panel. Then, in the *Node Info* panel, select "Upload SNAPpy Image" . (If the button is disabled, it means Portal does not know enough about the node to enable it to load a script into it. Click the "Refresh Node Information" button, and after the node information refreshes the "Upload SNAPpy Image" button should be enabled.)

This will bring up a dialog box asking *which* script/image to upload.



These scripts are located in a Portal\snappyImages directory in your MyDocuments folder when using a Windows operating system. (On Macintosh and Linux, the scripts are in the user's home directory.)



Select the "buzzer.py" example, and press OK (or you could double-click on the script name). Upload of the new SNAPpy Image (over the air!) should complete in a few seconds. The node will automatically restart when the upload finishes. (Depending on the script loaded into a node, restarting may not cause any visual change. In this case, the yellow LED will begin blinking after the node restarts. This node is the SNAP Engine on the SN171 Proto Board.)

Now select the node with device type "Photo" from the *Node List* and upload the "DarkDetector.py" script into it. (This loads a new script into the SNAP Engine on the SN111 End Device Demonstration Board.)

Notice that in this example we are running different scripts on each node. *The nodes do not have to be running the same script to interact with each other.*

The buzzer.py script is pretty simple:

1) When buzzer.py first runs, it "announces itself" to any nodes that might be looking for a buzzer they can use.
2) It blinks the yellow LED once a second (just to let you know the script is running).
3) If you push the button on that node, a short beep will sound (just to let you know the buzzer is working)
4) If any node broadcasts "who has a buzzer I can use?", buzzer.py will answer
5) Other nodes can ask for a "beep" of a specified duration.

So, script buzzer.py basically provides a "buzzer service" that *other nodes* can take advantage of.

Now look at the SN111 End Device node (Device Type = "Photo"), which should now be running the DarkDetector.py script:

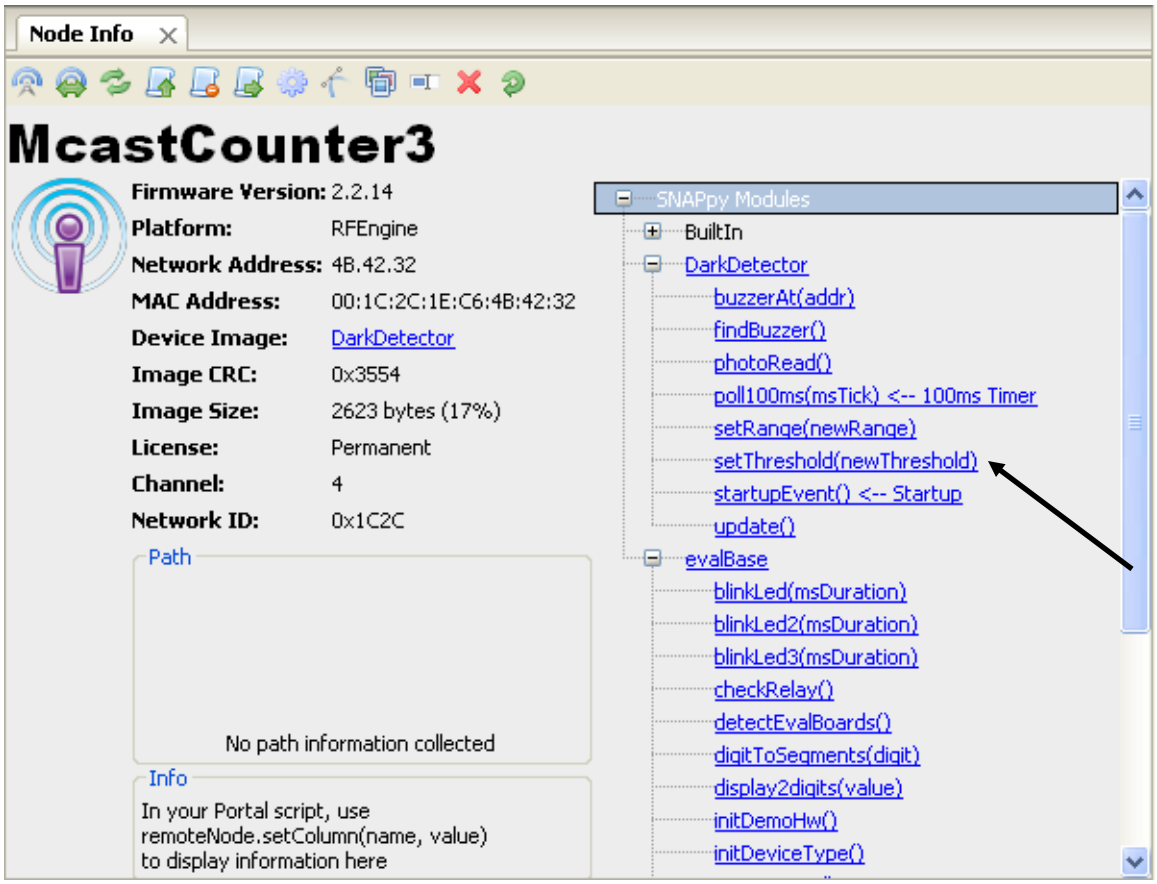
1) The yellow LED should be blinking once per second
2) Depending on the light levels and variance where the SN111 is sitting, it is either showing a value of "00" on its display, or it might be displaying various values. The displayed value represents a "percentage of darkness", 00-99, with a value of 99 representing the maximum darkness level the node has ever seen.

The Dark Detector script is *already* monitoring for changes in light levels, but the script also includes an "auto-calibration" capability. Since you have not "calibrated" the node yet, no values will be acted upon.

Cover the photocell on the node with your finger. A value of "99" will be shown on the display, and you should also hear a short beep from the *other* node (the SN171 Proto Board, which is running the buzzer.py script we uploaded into it).

Now by "hovering" your finger at various heights above the photocell, you should be able to vary the "darkness level" from 0 to 99%. Whenever you cross the 85% level, a short beep should sound from the "buzzer.py" node.

In Portal, bring up the **Node Info** panel for the Dark Detector node, and expand the tree of functions provided by the "DarkDetector" module.



Notice the setThreshold() function. The script defaults to “beeping” when the darkness level crosses 85%, but you can change this threshold “on the fly” by calling this function.

Click on setThreshold(), and answer the pop-up dialog box with a value of 50. (Don’t forget to click on OK.)



Now vary the darkness level by covering the photocell, and notice that the threshold point has indeed been changed. Now the remote beep (on the other node) will be triggered by crossing 50% darkness.

Note that the node names do not change within Portal, even though the underlying script names have. You can **rename** each node manually, or **delete** each node and then do a manual **ping** to rediscover them under new names.

Also notice that the SS200 SNAP Stick is still running the McastCounter.py script. It continues to relay messages to the other nodes from Portal even though the scripts running on those nodes do not otherwise interact with the node.

All of the nodes are independent – they can be running completely different scripts, and performing completely different functions, all at the same time and continue to communicate with and pass messages for each other.

Portal Scripting

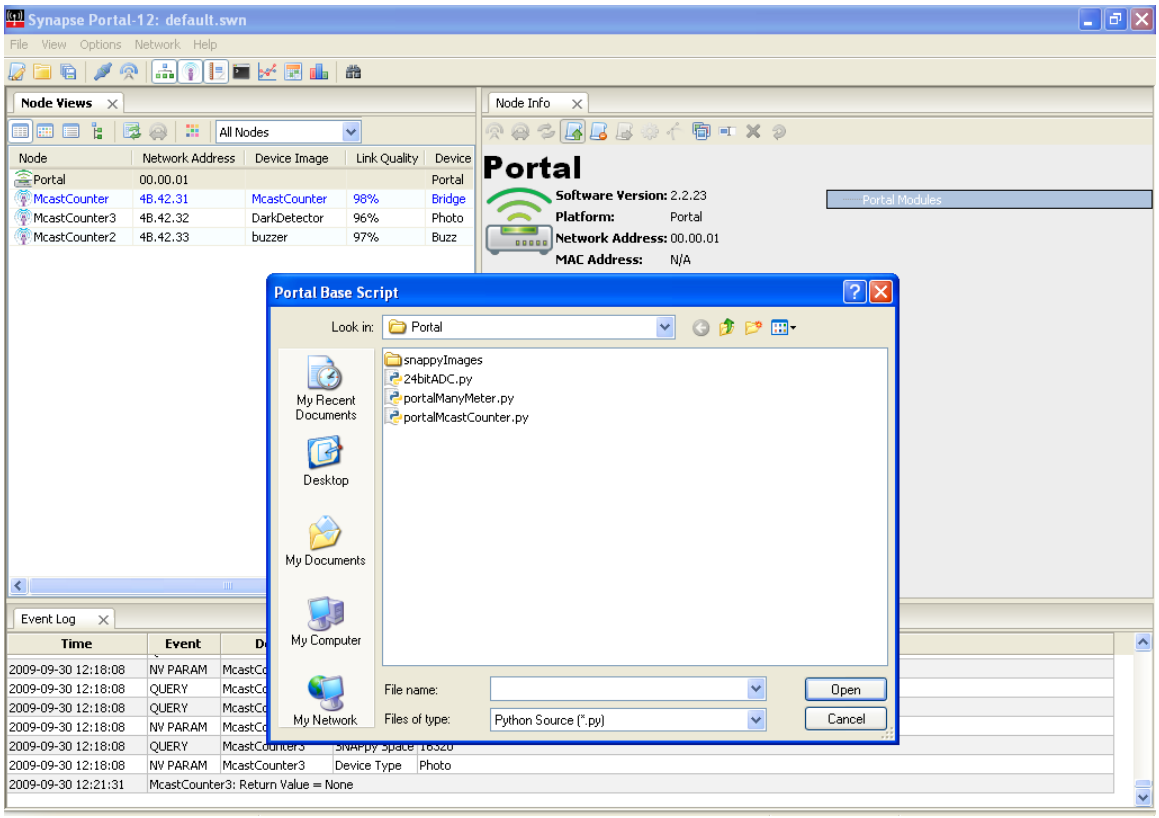
A fundamental property of SNAP is that all nodes are *peers* on the network. This provides tremendous advantages in system design and integration. With SNAP:

- All nodes are capable of routing mesh traffic
- No special node-type is required to form a network
- Any node can *bridge* mesh traffic over a UART port

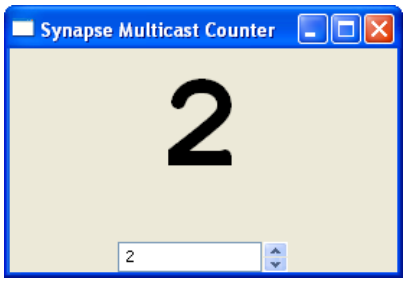
When you connect your PC to a SNAP Engine with a USB or serial port, you are using that SNAP Engine as a *bridge*. This allows Portal to participate as a peer on your SNAP network. Other SNAP devices see Portal as "just another SNAP node." Behind the scenes, the Portal GUI is calling built-in functions on SNAP nodes, and they are calling built-in functions in Portal. This is all done using SNAP's RPC (Remote Procedure Call) based protocol, which is exactly the way all SNAP nodes communicate with each other.

Since Portal is "just another SNAP node," we can load a script into it! Scripts running on Portal have all the access to the resources of the PC that the PC's user has, through libraries of the Python language. For instance, logging data to a file or database system is a simple matter using a Portal script.

A simple example of a Portal script is already installed. Select the Portal node in the **Node View** window, and use  to upload a new Portal Base File.



Select the file “portalMcastCounter.py”, and it will immediately become the device image for Portal. If you’ve been following the tutorial closely, you’ll still have the McastCounter script loaded on your Bridge node. (If not, upload it.) Use the Upload SNAPpy Image button to reload the McastCounter script into the nodes on the SN111 and SN171 boards as well. Now press the select button on an McastCounter node. You should see the following popup window (your displayed count value may differ from the “2” shown):



The Portal script you uploaded implements the “setButtonCount()” function, and it uses that to update the displayed number. Experiment with incrementing this count from all your SNAP devices. Also, try scrolling the number up/down from the PC – you can use keyboard arrow keys to do this rapidly.

It is important to note that since Portal runs full Python scripts rather than scripts written in the SNAPpy defined subset of Python, scripts that work perfectly well in Portal are not

likely to work in other nodes and vice versa. The underlying structure and grammar will be the same, but Python provides access to more data types and libraries than SNAPpy does.

Note that the GUI part of this script really has little to do with SNAP; you'll need other references if you want to explore programming Python on the PC further. For advanced applications, user-interfaces, etc. the **SNAPconnect** product provides a way for your application to participate in the SNAP network.

When you've finished playing with portalMcastCounter, you may want to "unload" this script from Portal. Removing a script is done in the same way for all SNAP nodes: just use the Erase SNAPpy Image () button on the Node Info tab's toolbar. (For Portal, the button is labeled Erase Portal Base File.)

Radio Range Testing

Now let's change the behavior of the nodes again. The SN111 board has a seven-segment display on it. We can use this display to provide real-time information on radio performance (receiver link quality).

NOTE! – The "LinkQualityRanger" script is not compatible with kits using the RF300 RF Engine

Upload script "LinkQualityRanger.py" into SN171 (Device Type = "Buzz") and SN111 (Device Type = "Photo") nodes. The two nodes will begin multicasting "test" messages to each other, at a rate of 10 messages per second (one message every 100 milliseconds).

When script "McastCounter.py" was running in these nodes, the seven-segment display showed a "current count" value. That count value could be changed by pressing a button on any of the nodes running the McastCounter.py script.

Now the seven-segment display shows radio receiver link quality. The display is updated every second with a link quality value (0-99%, where 99% is best), averaged over the last 10 received messages.

Your node on the SN111 board should be displaying a link quality value higher than 0. Depending on how close the nodes are to each other, your displayed link quality might be as high as 99%. Note that if your nodes are too close to each other, lower values may be observed due to receiver saturation.

At this point you might want to move the units around, and see how link quality is affected by both distance and intervening structures or obstacles.

This brings us to our next topic: How can we easily move the nodes around, when they have power cords and/or cables attached to them?

Battery Operation

So far, we have been running the nodes from USB power (the bridge node) and from wall power (the SN111 and SN171 nodes).

It is also possible to run most nodes from battery power.

The SN111 node comes with on-board battery holders, which can hold one or two 1.5 volt AA cells. You have the option of using one or two AA batteries.

NOTE! – The SN111 has a jumper on the right-hand side of the battery holder that configures the node to run from **one** AA cell, or **two** AA cells.

If the jumper is on posts 1 and 2 (the posts furthest away from the battery holder), then the unit only uses one AA battery, which should be installed in the battery holder closest to the center of the board. If the jumper is on posts 2 and 3 (closest to the battery holder), then the unit requires two AA batteries.

Set the jumper to match the actual number of batteries you will be using. Using two AA batteries instead of one will result in longer operation before having to replace the batteries.

Regardless of how many batteries are used, be sure to install them in the battery holder with the correct orientation. The bottom of the battery holder has the usual + and – labels molded into the plastic.

Note: There is a jumper labeled "RS232 PWR" located near the center of the board. This jumper must be installed for the RS232 port to work *when running on battery power*. To increase battery life it should **not** be installed if you are **not** using the RS232 port.

The SN171 Proto Board does not have a built-in battery holder, but a separate battery holder for this node is included in the EK2500 kit.

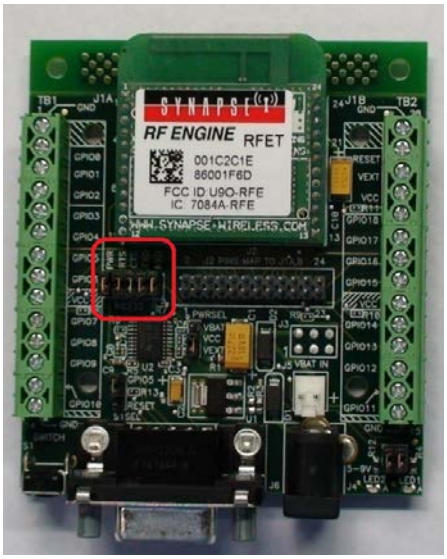
Before you can power the SN171 from the external battery pack, you must unplug the external power supply, and change the PWRSEL jumper located near the center of the circuit board. The jumper posts for this jumper are labeled "VBAT", "VCC", and "VEXT". The board comes from the factory with the jumper connected to the VEXT and VCC pins, which configures the node to run from the external power supply.

If you move the jumper so that it connects the VBAT and VCC pins, then the board will be configured to receive power from the white two-pin connector located directly behind the barrel-jack that the external power supply plugs into.

After changing the jumper to the VBAT+VCC position, install two AA batteries in the external battery holder, and connect the white connector from the external battery pack to the mating white connector on the SN171 board.

NOTE! There is an on/off switch on the external battery pack. Be sure to slide it to the "On" position when you want to power up the SN171 node.

Low Power Operation



The Proto Board is capable of achieving *years* of battery life from the included AA pack, but to do so requires running a SNAPpy script that *sleeps*, as well as removing **all** RS232 jumpers shown in the photograph (JMP2, JMP5, JMP6, JMP7 and JMP8). Unless you are running such a low-power script, be sure to turn battery power **off** when the node is not in use.

See example SNAPpy script "protoSleepCaster.py" for one example of low-power operation. This script is like McastCounter.py, but sleeps between button presses.

Where To Go Next

In this manual we have introduced the components of the EK2500 Evaluation Kit, installed Portal, and run some simple demos.

Now you will want to take advantage of some of the other SNAP documentation:

- The "SNAP Primer"
- The "SNAP Users Guide"
- The "SNAP Reference Manual"
- The "Portal Reference Manual"
- The "SNAP Hardware Technical Manual"
- The "End Device Quick Start Guide"
- The "SN171 Quick Start Guide"

These documents are in Portable Document Format (PDF) files. The first three are installed with Portal and are available from Portal's Help menu. The device guides and hardware manual are available from the Synapse website.

In addition, you can find an ever-expanding collection of useful information on the Synapse Support Forum at http://forums.synapse-wireless.com, including:

- Synapse Application Notes
- More Example Scripts

Here you can see questions and answers posted by other users, and you can post your own questions, as well.

License governing any code samples presented in this Guide

Redistribution of code and use in source and binary forms, with or without modification, are permitted provided that it retains the copyright notice, operates only on SNAP® networks, and the paragraphs below in the documentation and/or other materials are provided with the distribution:

Copyright 2008-2010, Synapse Wireless Inc., All rights Reserved.

Neither the name of Synapse nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SYNAPSE AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SYNAPSE OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SYNAPSE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Disclaimers

Information contained in this Guide is provided in connection with Synapse products and services and is intended solely to assist its customers. Synapse reserves the right to make changes at any time and without notice. Synapse assumes no liability whatsoever for the contents of this Manual or the redistribution as permitted by the foregoing Limited License. The terms and conditions governing the sale or use of Synapse products is expressly contained in the Synapse's Terms and Condition for the sale of those respective products.

Synapse retains the right to make changes to any product specification at any time without notice or liability to prior users, contributors, or recipients of redistributed versions of this Guide. Errata should be checked on any product referenced.

Synapse and the Synapse logo are registered trademarks of Synapse. All other trademarks are the property of their owners.

For further information on any Synapse product or service, contact us at:

Synapse Wireless, Inc.
500 Discovery Drive
Huntsville, Alabama 35806

256-852-7888
877-982-7888
256-852-7862 (fax)

www.synapse-wireless.com