WirelessUSB™ N:1 Development Kit

User's Guide

Cypress Semiconductor Corporation • 3901 North First Street • San Jose • CA 95134 • 408-943-2600

Table of Contents

1. INTRODUCTION

The CY3635 N:1 development kit was designed to demonstrate a "many to one" protocol using the WirelessUSB™ family of radios. This DVK will support up to 512 Sensor nodes connected to a single Hub.

The N:1 protocol has been designed to operate in noisy environment and uses a combination of frequency agility, coding gain, and packet error detection and correction in order to provide reliable wireless communication.

This User's Guide provides instructions to properly assemble the WirelessUSB N:1 DVK components and to demonstrate the functionality of WirelessUSB Radios.

All of the hardware, firmware, and software files used to create the kit are included on the enclosed CD-ROM. The WirelessUSB™ N:1 Development Kit (DVK) includes the following hardware components:

Qty	Item
5	WirelessUSB N:1 node boards (PDC-9178)
5	WirelessUSB LR radio micro-modules (PDC-9192)
2	Serial adapter boards with ribbon cables (PDC-9182)
1	Prototype board with ribbon cable (PDC-9181)
2	Serial cables
1	USB to serial adapter
1	USB 5V power cable
1	5VDC power adapter
12	AAA alkaline batteries
1	Magnet
1	Set of select printed documentation
1	CD-ROM

Table 1: N:1 DVK Contents

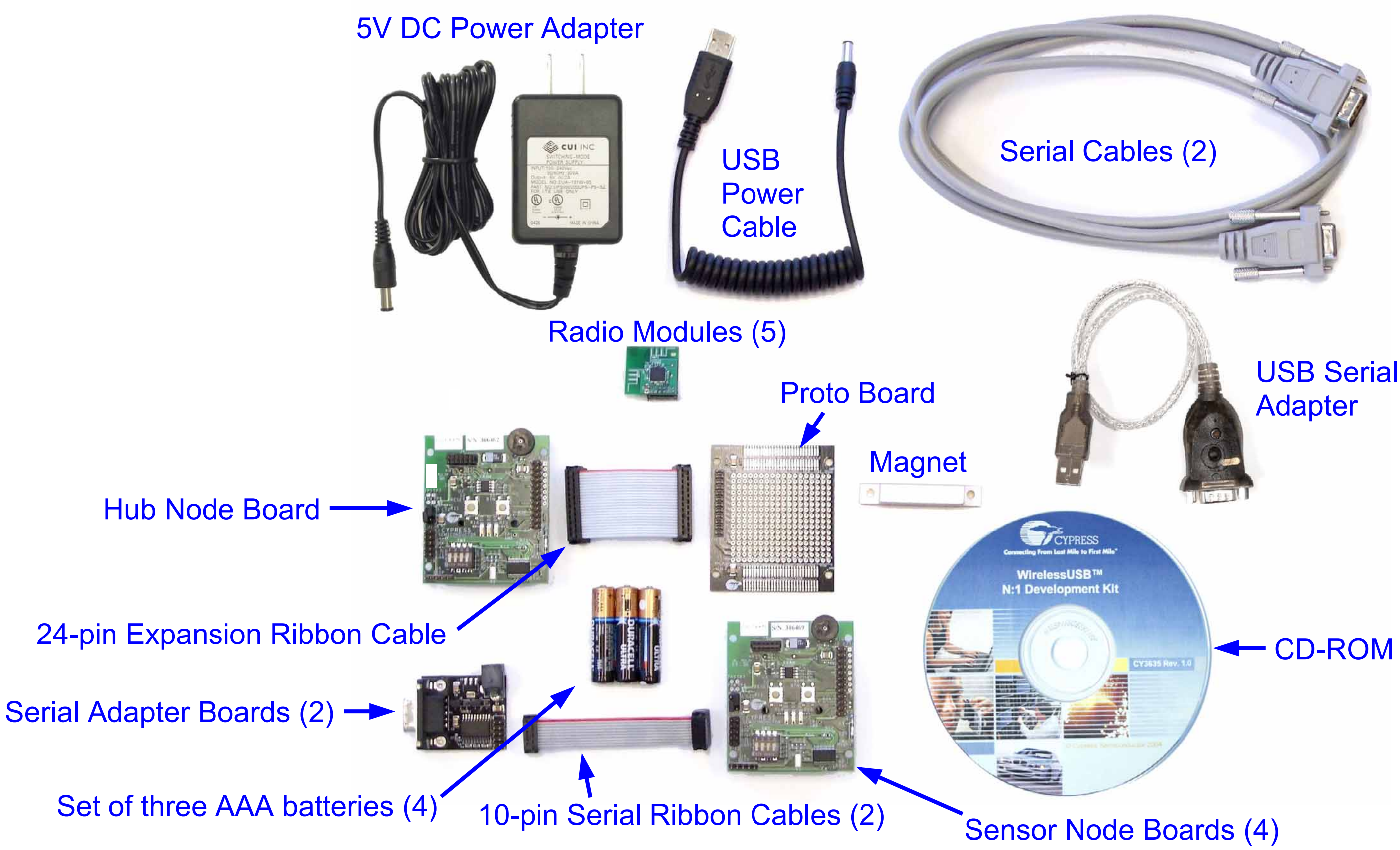

Figure 1: WirelessUSB N:1 DVK Contents

2. QUICK START

Please refer to the *CY3635 Quick Start Guide* included in the kit.

3. PRODUCT OVERVIEW

The N:1 system consists of many Sensor nodes communicating with a central Hub node. The Hub is supported by a software application running on the Host PC via a serial interface for monitoring and controlling the N:1 network.

Follow the instructions in the *CY3635 Quick Start Guide* in order to get the network up and running. Sensors in the Kit are pre-configured to automatically connect as a network when powered up.

4. N:1 SOFTWARE APPLICATION

With the network up and running, the N:1 Application screen should look similar to the following screen capture.

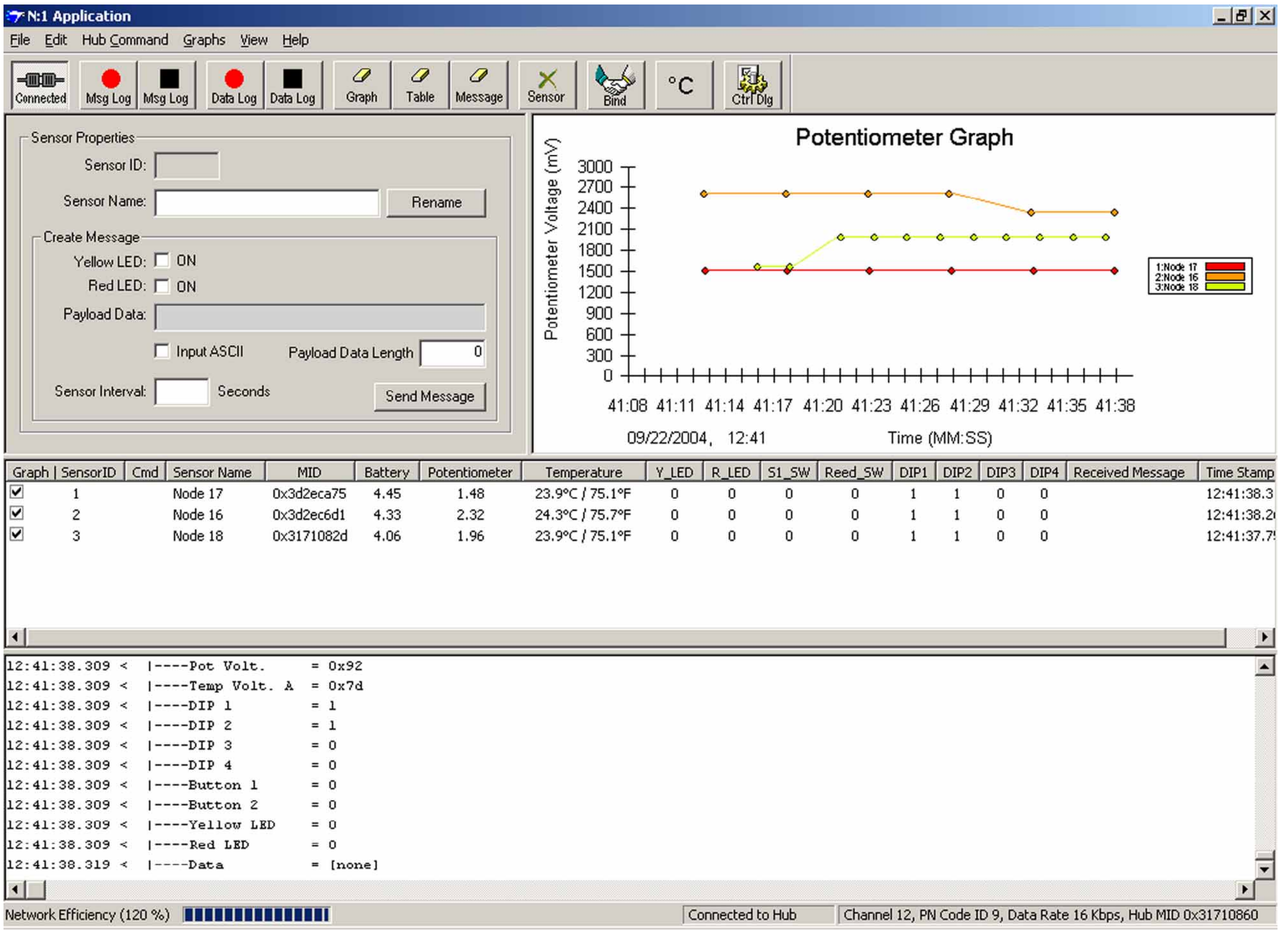

Figure 2: Software Application Screen Capture

The Sensors initiate communication with the Hub. This allows for the Sensors to optimize their power conservation. When they are not communicating, they are in a low-power state waiting for a hardware wake-up event or the sleep timer to expire. The Hub has the ability to queue a single back channel payload in non-volatile memory and will send the payload when the Sensor checks-in.

The N:1 Application screen is divided into four areas: the Detail pane in the upper-left displays details about a selected Sensor; the Graph pane in the upper-right displays data from up to 10 nodes at a time; the Sensor Status Table in the middle displays information about each of the enumerated Sensors; and the Message Log pane on the bottom shows the Host to Hub communication.

4.1 Detail Pane

Selecting a Sensor from the Sensor Status Table will cause its settings to appear in the Detail pane. Each element in the pane is described below.

4.1.1 Sensor Properties Group Box

The Sensor ID field shows the Device ID assigned to the selected Sensor. The Sensor Name edit control displays a user-definable friendly name for the Sensor (e.g. Room 401, etc.). Changing the Sensor Name and pressing the Rename button allows you to give the Sensor a different name. The Sensor Name is associated with the Manufacturing ID and is stored in the registry so that the name is retained when the application is restarted.

4.1.2 Create Message Group Box

The Create Message area allows you to generate a payload to be sent to the Sensor. Pressing the *Send Message* button will cause the payload to be queued on the Hub. When the Sensor checks-in, the Hub sends the queued payload. The following information is contained in the Sensor payload:

- **Yellow/Red LED:** Checking/un-checking this box will turn on/off the associated LED. The new state of the LED will be reflected in the Sensor Status Table under Y_LED/R_LED. Because the Sensor sends its state before accepting the payload from the Hub it takes an additional check-in before the correct state of the LEDs is reflected in the Sensor Status Table. The LED will turn on/off after the receipt of the Hub payload.

- **Report Rate** is the number of seconds between reports from the Sensor.

  Out of the box, the Sensor wakes up and connects to the network every 5 seconds. The interval is defined by the compile-time constant `DEFAULT_INTERVAL` located in `config.h`. If communication is lost with the Hub, the Sensor will use an interval set by the compile-time constant `DEFAULT_NON_CONNECTED_INTERVAL` of 60 seconds (also located in `config.h`). In this state, the Sensor will attempt several retries and channel searches in order to establish communication with the Hub. An interval of 60 seconds will conserve power without causing excessive delays. When the Sensor is able to connect with the Hub, the interval will revert back to the previous connected state interval.

Pressing S1 or activating the reed switch with the magnet will generate an immediate report.

- **Payload data**, change the *Payload Data Length* to the number of bytes to be sent (max 5) and type in a message in the *Payload Data* field. On the next Sensor update, the payload data will be sent to the Sensor. The Sensor will also output the data to the serial debug port for display on a terminal emulator (such as HyperTerminal). Payload data can also be sent from the Sensor to the Hub and will be discussed later.

4.2 Sensor Status Table

The Sensor Status Table displays information for all enumerated Sensors connected to the Hub.

Graph	SensorID	Cmd	Sensor Name	MID	Battery	Potentiometer	Temperature	Y_LED	R_LED	S1_SW	Reed_SW	DIP1	DIP2	DIP3	DIP4	Received Message	Time Stamp
☑	1		Node 17	0x3d2eca75	4.45	1.48	23.9°C / 75.1°F	0	0	0	0	1	1	0	0		12:41:38.3
☑	2		Node 16	0x3d2ec6d1	4.33	2.32	24.3°C / 75.7°F	0	0	0	0	1	1	0	0		12:41:38.2
☑	3		Node 18	0x3171082d	4.06	1.96	23.9°C / 75.1°F	0	0	0	0	1	1	0	0		12:41:37.7

Figure 3: Sensor Status Table

- **Graph** – Checking this box will add this Sensor to the graph data. Note that a maximum of 10 Sensors can be selected for graphing at one time.

- **Sensor ID** – A unique 2-byte number assigned by the Hub to each Sensor during bind. It is the handle used by the software and firmware for identifying a unique Sensor.

- **Cmd** – Shows the pending command to the Hub. The following table defines the various commands

Cmd	Description
D	Application requested Deletion of Sensor
E	Error in processing command
O	Previous (pending) send command overwritten
P	Send command pending
S	Sending command to Sensor

Table 2: Pending Commands

- **Sensor Name** – User modifiable name that can provide a more logical name to a single node or a group of nodes.

- **MID** (Manufacturing ID) – Unique number stored in each Sensor's WirelessUSB radio chip. The N:1 Application stores and retrieves information about each sensor from the registry based on the MID.

It is recommended that this number is printed on each unit to assist user identification of each physical node during setup of the application.

- **Battery**, **Potentiometer**, **Temperature**, **Y_LED**, **R_LED**, **S1_SW**, **Reed_SW**, **DIP1**, **DIP2**, **DIP3**, and **DIP4** – Displays the last reading from the Sensor. Because the Sensor initiates communication to the Hub, an additional update is required before the status of an LED command is accurately reflected in this view (because the Sensor samples the LED state before processing the back channel command packet from the Hub). The current information for each of these entries is cleared when the *Clear Sensor Data* button is pressed on the toolbar.

- **Received message** – Displays user payload data sent by the Sensor. This field is only updated when new user payload information is received from the Sensor. An asterisk is prefixed to this field when a packet with no user payload is received (in order to indicate stale payload data). The information under this column is cleared when the *Clear Sensor Data* button is pressed on the toolbar.

  Data is sent from the Sensor to the Hub when the user types into a remote terminal connected to a Sensor via the Serial Adapter Board. The terminal must be configured with the following settings:

  - 115.2kbaud
  - No parity
  - 8 data bits
  - 1 stop bit
  - No flow control

  If the Sensor is sleeping it will wake up and wait for additional bytes to be sent from the terminal. The first byte typed into the terminal will be lost because it was used to wake up the Sensor. As you type, bytes will be queued in the Sensor until a carriage return is entered or seven characters have be received. The Sensor will then transmit the queued bytes to the Hub and the Application will display the received data under this field. This field is cleared when the *Clear Sensor Data* button is pressed.

- **Time stamp** – Time when the last Sensor message was received. This information is cleared when the *Clear Sensor Data* button is pressed on the toolbar.

4.3 Graph Pane

This area of the application is used to graphically display temperature or potentiometer voltage for up to 10 devices. From the *Graphs* menu, select either *Temperature* or *Potentiometer*. To change the scale, select *X Axis Scale Setting…* from the *Graphs* menu in order to change the amount of time shown on the screen at once. To change the temperature scale, select *Temperature Graph Setting…* from the *Graphs* menu.

4.4 Message Log Pane

This area displays the details of the serial communication between the Host PC and Hub. Consistent Overhead Byte Stuffing (COBS) is the encoding scheme used for serial communication and is explained further in the *CY3635 Technical Reference Manual*.

4.5 Status Bar

The status bar is shown below.

Figure 4: Status Bar

The *Network Efficiency* meter on the left of the Status Bar is a relative measure of the health of the network, and not an indication of bandwidth consumption. It is based on the number of Sensors and their individual report rate. If all of the Sensors are working properly, then *Network Efficiency* should be approximately 100%. If half of the Sensors stop reporting, the Network Efficiency will drop to 50%. It does not explicitly measure RF quality of the link or the level of interferers on the network, although it can provide a some indication of performance degradation under difficult conditions. If the user initiates asynchronous events in addition to the normal periodic Sensor events (by pressing the S1 button or activating the Magnetic Switch), then the *Network Efficiency* can exceed 100%.

The other half of the Status Bar contains the current network parameters. These are reported from the Hub in real-time. When the Hub changes channels it will be reflected in the Status Bar immediately. These parameters are useful for properly configuring the optional WirelessUSB Listener Tool to monitor the wireless traffic.

4.6 Toolbar

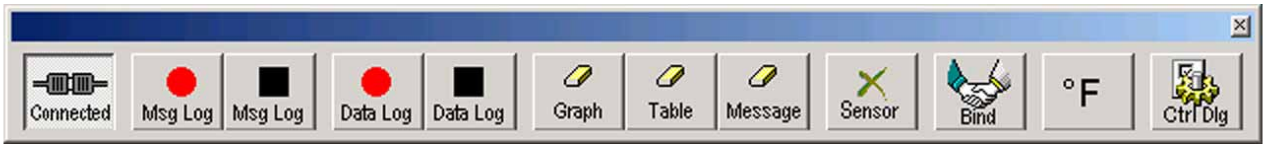

Figure 5: Toolbar

- **Connect to Hub** – Initiates communication between the Host PC and Hub via an RS-232 connection. When communication is established the button will change to the Connected state.

- **Capture Message Log to File** – When enabled, all output to the Message Log Pane is also written to a file. This button stays in the down state while the log file is open.

- **Stop Message Log Capture** – Closes the Message Log file and restores the *Capture Message Log To File* button to the up state.

- **Capture Sensor Data to File** – When enabled, all of the controls in the Control Dialog and all of the current Sensor readings are written to the Sensor Data file at the interval specified in the Control Dialog's *Storage Rate* field. This button stays in the down state while the log file is open.

- **Stop Sensor Data Capture** - Closes the Sensor Data log file and restores the *Capture Sensor Data To File* button to the up state.

- **Clear Temperature and Potentiometer Graph Pane** - All data points are removed from the graphs when this button is pressed.

- **Clear Sensor Status Table Data** – This is used to clear all of the Sensor data in the table. This is useful to monitor when each Sensor checks-in. This button does not affect any graph or logging features.

- **Clear Message Log Pane** – Deletes all of the accumulated output from the Message Log Pane. This button does not affect the Message Log File.

- **Delete Selected Sensor(s)** – This will delete all of the Sensors selected in the Sensor Status Table. A command is sent to the Hub that removes them from the Hub's non-volatile memory. The next time a deleted Sensor checks-in, the Hub will inform the Sensor that it has been removed from the network. The Sensor will then sleep until a hardware reset occurs, or until the user wakes it by pressing the S1 button. The Sensor will need to be re-bound to a Hub before it can successfully transmit data again. In the N:1 DVK, a Factory Reset will need to be performed.

- Bind On/Off – When Bind is on, the Hub spends half of its time servicing the Bind channels which removes bandwidth from servicing normal network traffic. For this reason, Bind should be turned off when not needed. The Hub turns on the Yellow LED while Bind is activated. The Hub's S1 button can also be used to turn Bind on/off.
    - Toggle between Fahrenheit and Celsius operation – This affects graphing and logging operations.
    - Open Control Dialog – Invokes the Control Dialog (described later). This dialog is only available after successfully connecting to the Hub.

4.7 Control Dialog

In addition to the main application window, a special Control Dialog is provided that is designed to monitor temperature Sensors and turn an actuator on and off. We will use an HVAC control system as our example when describing the Control Dialog.

Air conditioners, heaters, and electronic dampers will be controlled based on the wireless temperature information collected by the N:1 Application. The Control Dialog in the N:1 Application provides the mechanism to control the HVAC system when specified activation thresholds are met.

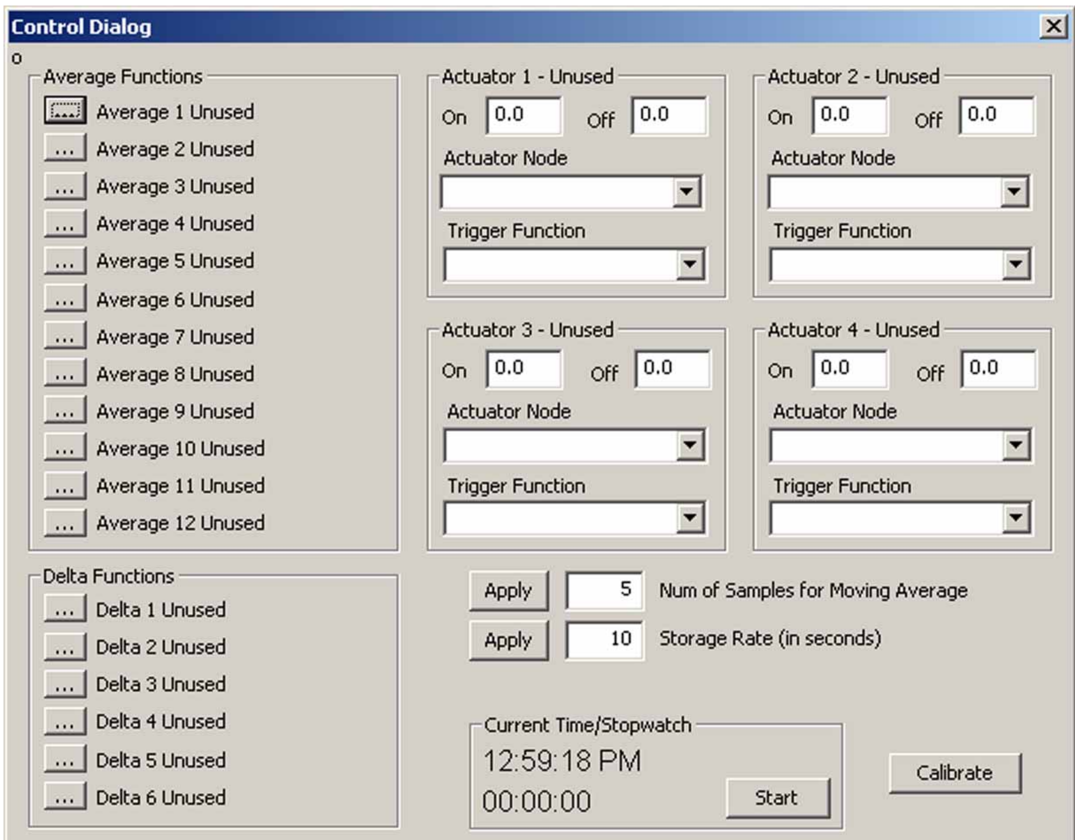

Figure 6: Control Dialog

4.7.1 Average Functions

There are 12 averaging functions available. Click on the button to the left of the average to select which Sensors to include in the average and to assign an appropriate name for the average function.

4.7.2 Delta Functions

There are six delta functions. When the button to the left of the delta function is pressed, the user can select the inputs used to calculate the delta function. An appropriate name can also be assigned for each delta function.

4.7.3 Actuator Functions

The average and delta functions are then used to create up to four actuator functions. The user specifies the *On* temperature and the *Off* temperature. The desired actuator node is specified along with the desired delta or average function to use as the trigger function. In our example, the user selects an *On* temperature of 80 degrees and an *Off* temperature of 75 degrees. This is based on the Average 1 function and will control Actuator 1. When the average temperature reaches 80 degrees or higher, the *On* command will be sent to the actuator and the Red LED will turn on. When the temperature drops to 75 degrees or below, the *Off* command is sent to the actuator and the Red LED will go off. The *On* value must be greater than the *Off* value.

Every time a Sensor report is received, the functions are recalculated.

After entering a temperature *On* or *Off* threshold, press the tab key (change the edit control's focus) for the new value to be accepted.

If you increase the value of the *On* threshold temperature after the *On* command has already been sent to the actuator node, an *Off* command will not be sent (since it hasn't passed through the *Off* temperature threshold yet). In this case, the actuator can be turned off manually through the Detail Pane in the main application window.

4.7.4 Control Dialog Settings

Number of Samples for Moving Average – If this value is greater than 1 the temperature Sensors are averaged over the number of readings specified. This is to help smooth out any undesirable variance in temperature readings.

Storage Rate – When a Sensor Data log file is recorded, a second file is also created with "ControlDlg" added to the name specified. This will capture all of the additional data in the Control Dialog to this second file at the interval specified here. This value should be a multiple of the Sensor update rate (which is defaulted to 5 seconds) so that each recording in the file will have fresh data. In order to view this log file while recording is in progress, make a copy of the file and then open the copy.

4.7.5 Calibrate Button

The calibration function can be used in order to zero out any undesired variation in Sensor readings. Place all of the Sensors in a controlled environment and allow their readings to settle. Then select the Sensors to be calibrated and then click the *Calibrate* button. This process takes all of the current temperature readings for the Sensors and computes an average temperature. That average temperature is then compared to each existing Sensor reading and an offset is created and stored in the Windows registry. This offset correction is then applied to all subsequent data reports from the specified sensors.

4.7.6 Timer/Stopwatch

When collecting scientific data, it helps to have the current time prominently displayed. There is also a simple stopwatch for convenient elapsed time measurements. Press the button to start/stop. When pressing start it will reset the count to 00:00:00.

5. N:1 FIRMWARE AND HARDWARE

Security sensors, room thermostats, manufacturing monitoring, and hotel door locks are just a few examples of possible N:1 applications. The kit was designed to illustrate several different possibilities in a compact and simple design. An implementation of any one of these applications would result in a smaller and simpler board.

5.1 Node Board (Hub and Sensor)

The Hub and Sensor boards are identical from a hardware perspective. Either board can be programmed to be a Sensor or Hub board. A label is used to identify which board is pre-programmed with Hub firmware.

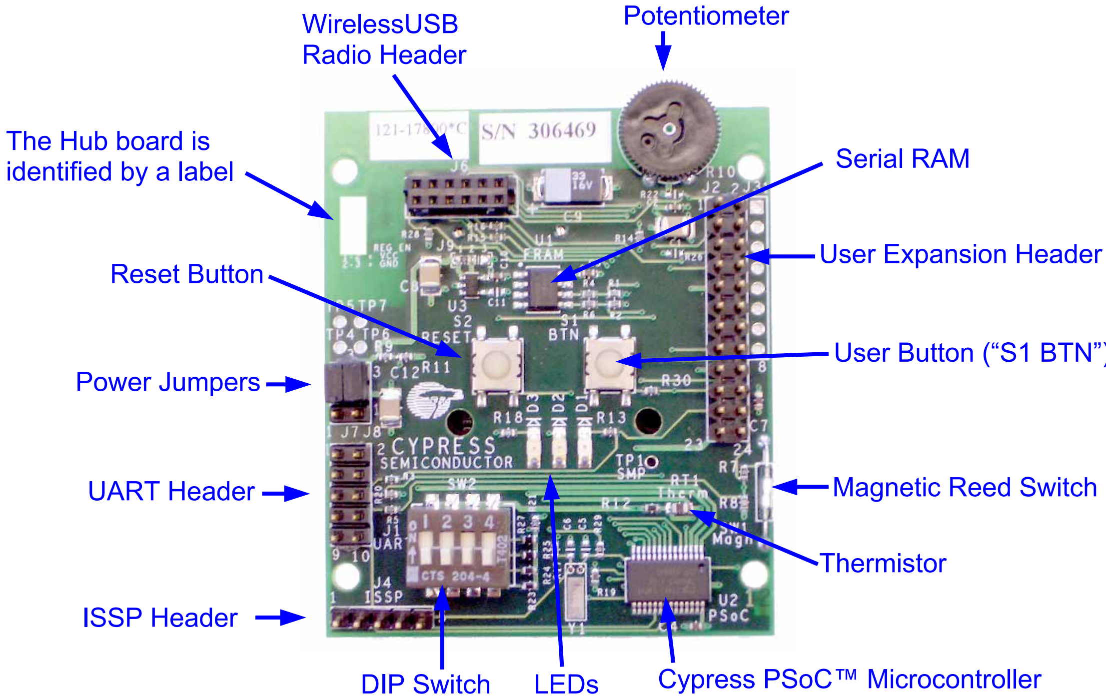

Figure 7: Node Board Overview

5.1.1 Hub and Sensor Jumper Settings

The Hub and Sensor use the same physical board assembly and differ only in the jumper settings. The Hub PSoC is powered directly from 5 volts which is required to operate the PSoC CPU at 24MHz. The Hub board is supplied with 5V from the Serial Adapter Board via the 5V DC Power Adapter or the USB 5V Power Cable.

The Sensor is typically configured to be powered by batteries, but it can also be configured to operate at 5 volts as well. The following table outlines the jumper settings for various power configurations.

Jumper J7 Selects node board power source	
Wall powered (5v)	1-2
Battery powered	2-3

Table 3: Node Board J7 Power Jumper Settings

Jumper J8 Selects PSoC supply voltage	
PSoC uses 5V supply	1-2
PSoC uses 3.3V supply	2-3

Table 4: Node Board J8 Power Jumper Settings

The following table shows the default jumper settings for the Sensor Board

Default Settings	
J7	2-3
J8	2-3

Table 5: Sensor Default Jumper Settings

The following table shows the default jumper settings for the Hub Board

Default Settings	
J7	1-2
J8	1-2

Table 6: Hub Default Jumper Settings

5.1.2 DIP Switch Configuration

The DIP Switches can be used as general purpose switches during run-time. If S1 is pressed during Reset, the DIP switches are used to access special modes as noted in the following table:

The kit is shipped with the DIP Switches set to 0000.

Hub	DIP Switch	Sensor
Select Seeded Bind network	0000-1110*	Select Seeded Bind network
Clear Non-Volatile Storage	0001	Clear Non-Volatile Storage
Unused	1001	Test Mode 1
Unused	0101	Test Mode 2
Unused	1101	Test Mode 3
Clock Out Enable	0011	Clock Out Enable
Unused	1011	Unused
Unused	0111	Unused
Unused	1111	Unused

*Note that the last bit must always be 0 (xxx0) in this mode, so only the first three bits are used to specify the Seeded Bind Network.

Table 7: DIP Switch Settings

5.1.2.1 Automatic Bind (Clear Non-Volatile Storage)

Clearing Non-Volatile Storage is used to clear all prior network parameters on the Hub and Sensor. For the Hub, this will create a brand new network. For the Sensor, this will allow it to join a different network.

Hub Automatic Bind Procedure

Note that using the Clear Non-Volatile Storage operation on the Hub will erase all prior bound Sensors and start a brand new network.

To add a new Sensor to an existing network using Automatic Bind, skip to the *Sensor Automatic Bind Procedure* below.

Use the following Hub procedure to erase all prior bound Sensors and start a brand new network.

Note: This will create a brand new network; All prior Sensors will be un-bound.		
Step	Device	Description
1	Hub	Set the DIP Switch to 0001 to select the Clear Non-Volatile Storage operation.
2	Hub	Press and hold both the S2 "RESET" button and the S1 "BTN" Button.
3	Hub	Release the Reset button
4	Hub	Release the S1 button (the red LED will now be flashing)
5	Hub	Press and release Reset. The Hub is now waiting for Automatic Bind Mode to be enabled. Refer to the next section for proceeding to bind Sensors to this Hub.

Table 8: Hub Automatic Bind Procedure

Sensor Automatic Bind Procedure

Follow this procedure to use Automatic Bind to add Sensors to a new or existing network.

Step	Device	Description
1	Host PC	In the N:1 Application, press the Bind button on the toolbar (or press the S1 button on the Hub board) to activate Automatic Bind Mode on the Hub. The yellow LED will turn on while Automatic Bind Mode is enabled.
2	Sensor	Set the DIP Switch to 0001 to select the "Clear Non-Volatile Storage" operation.
3	Sensor	Press and hold both the S2 "RESET" button and the S1 "BTN" Button.
4	Sensor	Release the Reset button
5	Sensor	Release the S1 button (the red LED will now be flashing)
6	Sensor	Press and release Reset. The Sensor will now proceed to use Automatic Bind Mode to bind to the Hub. Repeat steps 2-6 for each Sensor to be added to the network.
7	Host PC	In the N:1 Application, press the Bind button on the toolbar (or press the S1 button on the Hub board) to turn Automatic Bind Off. The yellow LED will then turn off. This is important so that the Hub is not distracted from servicing active Sensors.

Table 9: Sensor Automatic Bind Procedure

The “Clear Non-Volatile Storage” function is used to erase the bind parameters stored in FLASH on the Hub and Sensor. On the Hub, this function also clears the non-volatile FRAM that contains the Device ID table. When the Erase Bind parameters is complete, the firmware will flash the red LED indicating that the procedure has been completed and will need to be reset by pressing the Reset Button to resume normal operation.

After the PSoC comes out of reset on the Sensor and determines the FLASH bind parameters are invalid, it will then attempt an Automatic Bind on the Bind Channel Subset (Channel Subset 0). If a bind response is not received from the Hub, the Sensor will enter the sleep state and wait for S1 to be pressed before attempting to bind again.

After the PSoC comes out of reset on the Hub it will use the MID to compute a channel and PN Code to use on the new network.

5.1.2.2 Seeded Bind (Select Seeded Bind Network)

The range of DIP switch settings from 0000-1110 is used to select a Seeded Bind Network.

Hub Seeded Bind Procedure

Use the following procedure to re-configure a Hub for Seeded Bind Mode. This should only be done once before a network is established.

Step	Device	Description
1	Hub	Set the DIP Switches to desired network from *Table 12*
2	Hub	Press and hold both the S2 “RESET” button and the S1 “BTN” button.
3	Hub	Release the Reset button
4	Hub	Release the S1 button. The Hub is now ready to accept Seeded Bind requests from new Sensors.
5	Host PC	Make sure that Seeded Bind is enabled (there should be a check-mark next to “Seeded Bind” in the Hub Command menu). The Hub is now ready to receive Seeded Bind requests from new Sensors.

Table 10: Hub Seeded Bind Procedure

Sensor Seeded Bind Procedure

Use the following procedure to add a new Sensor to a new or existing network via the Seeded Bind method:

Step	Device	Description
1	Sensor	Set the DIP Switches to desired network from *Table 12*
2	Sensor	Press and hold both the S2 “RESET” button and the S1 “BTN” button.
3	Sensor	Release the Reset button
4	Sensor	Release the S1 button. The Sensor will then proceed to use Seeded Bind Mode to bind to the Hub. Repeat steps 1-4 for each Sensor to be added to the network.

Table 11: Sensor Seeded Bind Procedure

On the Sensor, performing the “Select Seeded Bind network” operation will force the Channel and PN Code bind parameters based on the Seeded Bind table below. If the Sensor Binds to a Hub the network parameters will be written to FLASH. If the Sensor is reset before it has bound to a Hub the prior network parameters in FLASH will be used.

On the Hub, this function will erase the bind parameters stored in FLASH and the non-volatile FRAM that contains the Device ID table. The Channel and PN Code based on the Seeded Bind table below are then written to FLASH.

Seeded Bind allows Sensors to join a network without the Hub needing to be in Automatic Bind Mode. While the Hub is in Automatic Bind Mode it splits its time between the Bind Mode channels and its own established network. Seeded Bind works by giving the Sensor the network parameters Channel and PN Code Index. The Sensor uses these parameters to connect to the Hub on the Hub's established network via a Bind Request. The Hub’s Bind Response will supply the Sensor with the remaining network parameters: Device ID and Host MID. The Hub services the seeded bind requests by default or this feature can be turned off via a command from the Host to the Hub. The downside to using Seeded Bind in your application is the coordination during manufacturing to make sure the Sensors have the correct seed values for the included Hub. Sensors connecting via the Automatic Bind method don't need to know anything about the Hub's network.

The Sensor and Hub in the N:1 DVK Kit can be forced to do a Seeded Bind on a subset of pre-configured networks. The Sensor obtains the seeding information from FLASH or from the DIP switches during startup.

The subset was chosen to fit in three DIP switches and work with both 64cpb and 32cpb. To perform a Seeded Bind, set the first three DIP switches to the desired setting based on the table below. S1 needs to be depressed when the PSoC comes out of reset in order for the firmware to perform a Seeded Bind based on the dipswitch setting.

DIP Switch	Channel Index	PN Code ID
0000	1	1
0010	2	2
0100	3	3
0110	4	4
1000	5	5
1010	1	7
1100	2	6
1110	5	5

Table 12: Channel and PN Code ID Selection

5.1.2.3 Clock Out Enable

This mode is used to enable the Radio Oscillator Output pin (X13OUT). This can be used to measure the frequency of the oscillator to ensure accuracy.

5.2 Serial Adapter Board

The Serial Adapter Board is used to convert 3.3V or 5V UART signals into RS-232 level signals. The board can also be used to provide external 5V power to the node board.

Typically, the Serial Adapter Board is connected to the Hub board for the following functions:

- Provide 5V power to the Hub Node Board (since the PSoC Firmware on the Hub is configured to run at 24MHz, which requires 5V power).
- Allow RS232 communication between the Hub and a PC Host.

The Serial Adapter Board can also be connected to a Sensor Board. This would typically be done when a simple one-to-one wireless serial link is required.

5.2.1 Serial Adapter Board Jumper Settings

The following tables describe the jumper setting options for the Serial Adapter Board.

Jumper J2 Selects 3.3V power source	
On-board regulator	1-2
From Node Board (via J1 header)	2-3

Table 13: Serial Adapter Board J2 Jumper Settings

Jumper J3 Selects power source of RS232 Transceiver	
3.3V	1-2
5.0V	2-3

Table 14: Serial Adapter Board J2 Jumper Settings

The following table shows the default jumper settings when using the Serial Adapter Board with the Hub Node Board.

Default Settings	
J2	1-2
J3	1-2

Table 15: Default Serial Adapter Board Jumper Settings

6. ANALYSIS TOOLS

6.1 Sensor Serial Port Diagnostics

WirelessUSB N:1 Sensor firmware provides useful diagnostic information through the serial port for debugging devices under development.

To view serial diagnostics:

1. Connect the Serial Adapter Board to J1 on the Sensor node board, and connect a serial cable from the Serial Adapter Board to an available COM port on a PC.
2. Launch a terminal emulator (such as HyperTerminal) on the PC. Select the correct COM port. Change the port settings to 115,200 baud, 8 bits, no parity, and 1 stop bit, no flow control (115200, 8, N, 1, none).
3. The debug data will display on the terminal application each time a packet is transmitted

Note: Serial debug output may change the PSoC radio timing. Adding extensive debug output is not recommended as it may alter the system timing causing devices to timeout before data can be transmitted.

6.2 Network Latency Tools (Test Mode 1&2)

By default, the Sensor firmware supports two 100 packet latency tests. Both tests will transmit 100 packets to the Hub while recording the total number of packet retries and the latency from when the packet was sent to when an acknowledgement is received from the Hub. The latency for each packet is output to the debug port and at the end of the test the total number of retires and maximum and minimum latency is transmitter to the Hub in the form of a user data packet and is displayed in the application window as follows [XX XX 20 YY]. XXXX = number of retries in hex, YY = maximum latency in hex. Test Mode 1 will transmit 100 packets and stop, Test Mode 2 will repeat the 100 packet test over and over.

To access these tests you must first have the Sensor bound and connected to the Hub. With the Sensor running, set the DIP switches to desired test (see *Table 4* above). Press and hold S1 while resetting the Sensor board and then release S1. The debug out monitor should indicate that the special test mode has been accessed. Press S1 to start running the test.

6.3 Dynamic Network Quality (Test Mode 3)

This test provides dynamic link quality using the LEDs on the Sensor board and is part of the standard Sensor firmware build. This test will enable the user to quickly assess the link quality of an individual Sensor. This test can be used to help position a sensor in an optimal location and orientation for fixed installations (such as security sensors and temperature sensors). The Red, Yellow, and Green LEDs are used as follows:

LED State	Description
Green, solid	Good connection with no missing ACKs from the Hub
Green, flicker	Retry due to missed ACK from Hub
Yellow	More than 2 retries required to connect to Hub
Red	More than 4 retries required to connect to Hub

Table 16: LED Behavior for Link Quality Test

See *Table 4* above for the correct DIP switch settings.

6.4 Parameter Selection Tool

The purpose of this tool is to allow you to estimate the network quality and battery life of an application based on several basic parameters.

This Tool is located on the CD in the **\Docs** folder, and is called *CY3635 Parameter Selection Tool.xls.*

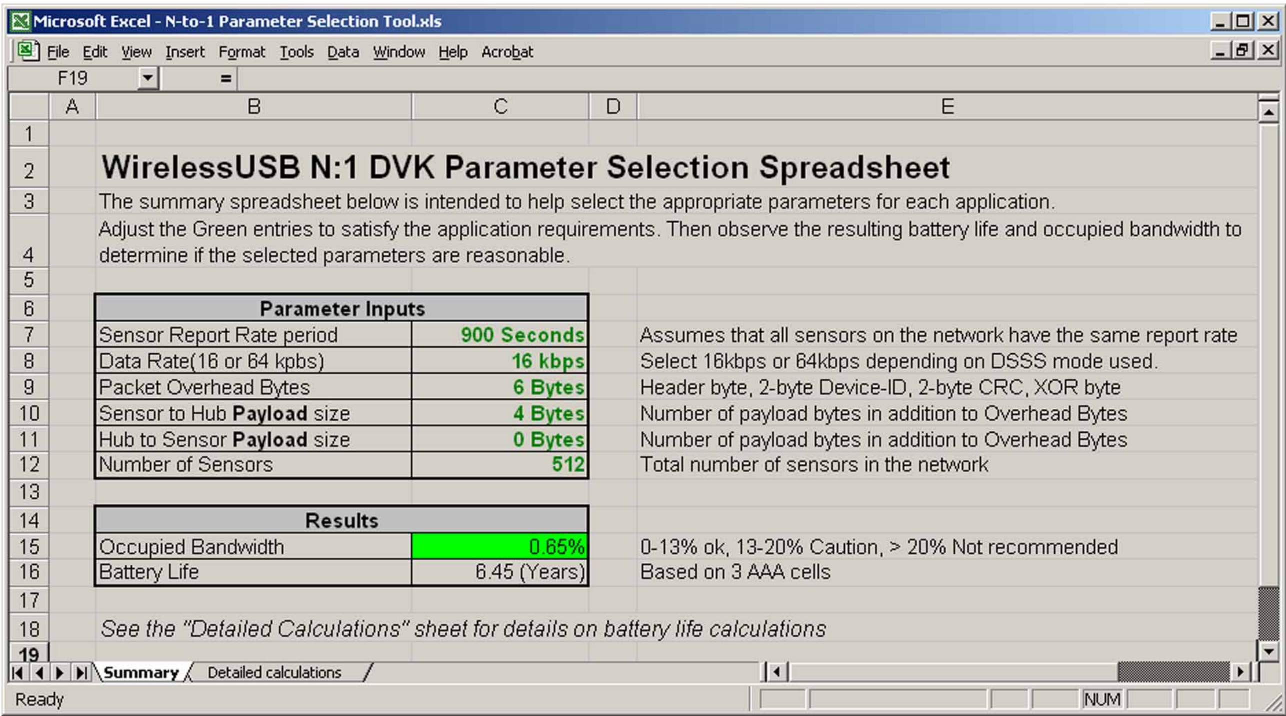

Figure 10: Parameter Selection Tool Screen Capture

The first summary spreadsheet allows you to enter the Sensor report rate, Radio data rate, payload size, and number of Sensors in the network. The expected Sensor battery life and occupied bandwidth will then be computed. The detailed calculation spreadsheet provides the calculations used for determining the battery life. Parameters on this second sheet should normally not be changed.

6.5 WirelessUSB Listener Tool

To assist developers in debugging WirelessUSB N:1 systems, Cypress has created the WirelessUSB Listener Tool which is available separately (CY3632L).

The Listener Tool works by listening to WirelessUSB traffic on a given channel and PN code. It passes the captured packets to a Host computer through a full-speed USB connection. The Listener Tool software decodes the captured data into packets and displays it. For further information, please refer to the documentation included in the Listener Kit.

Figure 11: WirelessUSB Listener Tool

The listener tool has specific features for supporting the N:1 DVK. To guarantee N:1 compatibility, be sure to use the Listener application program that is included in the N:1 CD in the **\Software\Listener Tool** folder.

To enable the special N:1 features, the N:1 button must be selected in the protocol box. This will allow the Listener software to correctly interpret N:1 packets and filter traffic based on the Sensor's device ID.

The Channel and PN Code ID parameters that are displayed in the status bar of the N:1 Software Application (see *Section 4.5*) should be entered into the Channel and PN Code boxes of the Listener Application. Note that the Data Rate should be set to 16KB/s since this is the default data rate supported by the N:1 kit.

The Device ID filter can be used to listen to the traffic between a single specified Sensor and the Hub. This is useful for debugging a single node when there are many active Sensors on the network.

The following figure shows a sample screen capture of the WirelessUSB Listener Tool.

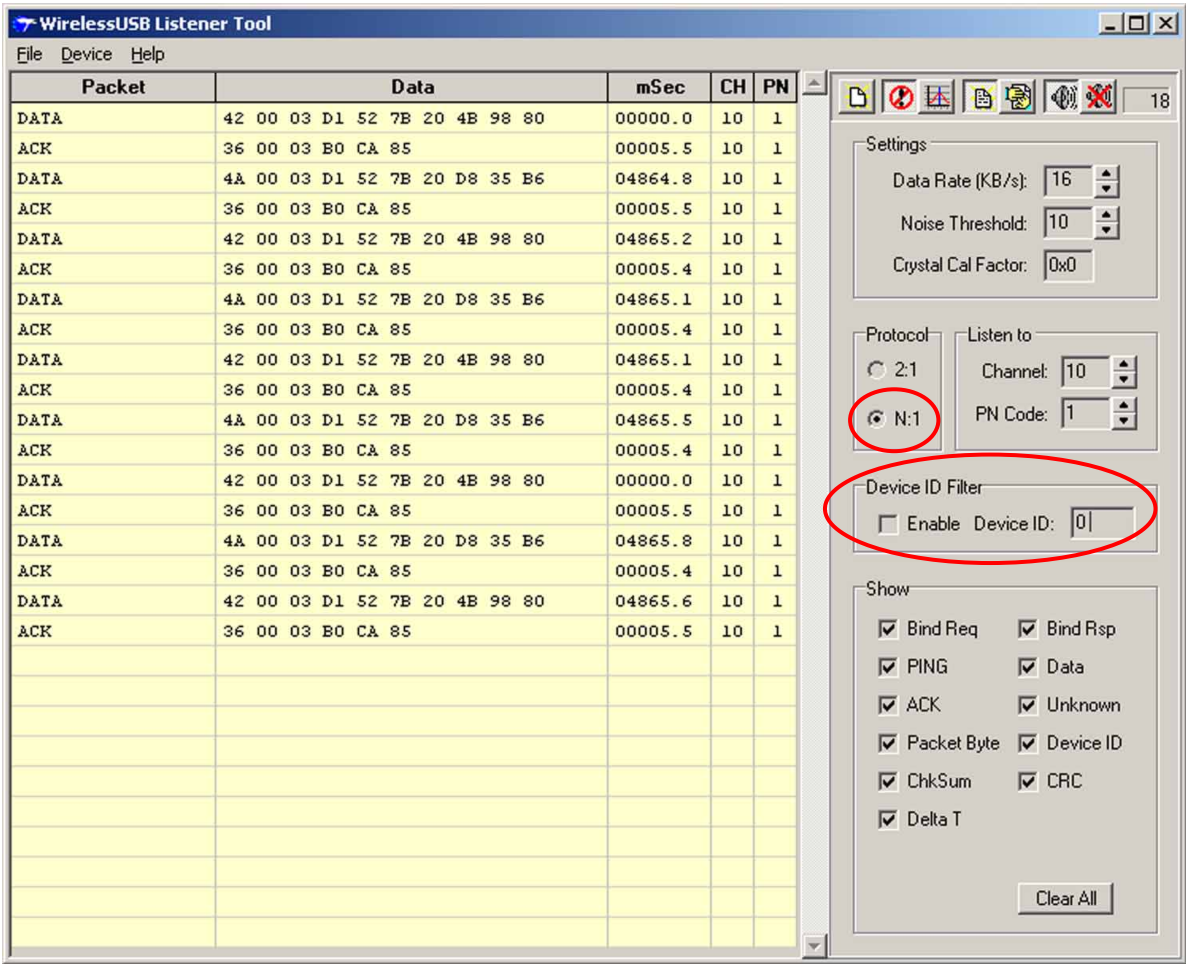

Figure 12: WirelessUSB Listener Tool Screen Capture

7. DEVELOPMENT ENVIRONMENT

7.1 Hub and Sensor Firmware Development

For firmware development of the Hub or Sensor nodes, the PSoC development kit is required. Please refer to the table below for detailed requirements. These kits are available for order from the Cypress Online Store at www.cypress.com

Development Type	DVK Part Number	DVK Description
WirelessUSB Node	CY3205-DK	PSoC™ Microcontroller Basic Development Kit
	CY3208-080*	SSOP adapter feet*
	CY3207-POD*	PSoC ICE Pod Kit for CY8C27X Device Family*
	CY3202-C	PSoC Designer™ C Compiler
	PSoC Designer v4.1 SP1	Included in this kit's CD-ROM

*These items are optional, since they are used for in-circuit emulation

Table 5: PSoC Development Kits

The minimum requirements for compiling and programming boards are:

- CY3205-DK provides the ICE and programming board
- CY3202-C Is the C compiler

In order to perform development using the In Circuit Emulator, then the following items are also required:

- CY3208-080 is the SSOP adapter foot that is soldered to the target board.
- CY3207-POD is the emulator pod that supports the CY8C27x family devices.

7.2 Software Development

Software development for the N:1 Software Application requires Microsoft Visual Studio 2003.NET.

All source code is provided on the CD-ROM.

8. INDEX

S

T

Document History Page

Revision	Date	Changes
1.0	10/19/04	Initial Release