Manufacturing Test Guidelines for the Ember® EM250, EM260, and EM35x

(Formerly document 120-5016-000)

Most customers have standard product manufacturing test flows, but some do not incorporate RF testing. This document details the different options for integrating RF testing and characterization into your standard test flows. This application note is intended for Silicon Labs customers who are moving from the early prototype development stage to the manufacturing production environment and need assistance with manufacturing test. The specific target audience is manufacturing test engineers developing test processes for their Ember-enabled products.

This application note applies to Ember EM250, EM260, and EM35x devices and corresponding stack releases EmberZNet PRO 4.0 and later.

For a high-level overview of these test phases, refer to document AN718, *Manufacturing Test Overview*.

Device programming is not discussed in this application note. For information on the programming options available for the EM250 and EM260 devices, refer to document AN707, *Programming Options for EM2xx*. For information on the programming options available for EM35x devices, refer to document AN717, *Programming Options for EM35x*.

If after reading this document you have questions or require assistance with the procedures described, contact a support representative at www.silabs.com/zigbee-support.

New in This Revision

Added frequency offset details in section "2.6. Transmit Frequency Test."

Contents

SILICON LABS

1 Test Flow

Document AN718, *Manufacturing Test Overview*, provides a high-level overview of the product tests flow and phases – prototype testing, characterization testing, low-volume manufacturing, and high-volume manufacturing. This document's goal is to provide the finer details of these test phases and Silicon Labs' recommended best practices for manufacturing test. Due to the various options available during prototype testing, this document focuses on the characterization and manufacturing test phases.

2 Test Definitions

To communicate with the device under test (DUT) and control various device test modes by automated test software, Silicon Labs provides embedded software applications to allow for both serial communication as well as Over the Air (OTA) communication to automated test software. Both of these interfaces enable configuring an Ember device for receive or transmit modes, turning on peripherals (if applicable), reading Analog to Digital Conversion (ADC) pins on the micro (if applicable), putting the radio and/or microprocessor to sleep, and similar control functions.

The tests can be divided into different types of tests— RF testing, DC testing, and peripheral testing.

- RF testing is any test specific to the operation and functionality of the radio (for example, transmitting and receiving ZigBee packets).
- DC testing is any test related to the voltage and current characteristics of the device or board (for example, active and sleep currents).
- Peripheral testing is any test not specific to RF or DC, like a sensor or an external crystal.

The following sections describe the tests that make up the potential suite of DUT testing.

2.1 Serial Communication Test

The Serial Communication Test verifies valid serial communication with the DUT before testing. This is a basic check that the device has been programmed correctly. If no communication is present, the DUT fails this test and does not proceed with further testing. This test is important because if there is no serial communication with the DUT, there is no way to interface with the DUT to put the device into test mode.

2.2 Channel Calibration Test

When a device is powered up for the first time and selects a channel, a calibration is automatically run to properly configure the device for operation. This calibration occurs the first time each channel is selected, as well as any time there is a change in temperature large enough to trigger recalibration.

Note: The HAL in the embedded software controls the calibration.

Silicon Labs recommends that customers select each channel so that the calibration is guaranteed to run before any functional testing. Silicon Labs also recommends that the calibration data values be stored in a data log or compared against a known range of valid values as a way of initially determining if there are any problems with the device or board. A check for valid calibration data can be a good initial test that the device is soldered down properly before any other functional checks.

The channel calibration data is stored within the token system as CAL_DATA (0xd243 for the EM250 and EM260, 0xd245 for the EM35x as of EmberZNet PRO 4.5, 0xd243 for earlier releases). The calibration token is 64 bytes long, 4 bytes per channel.

AN700

Note: The channel calibration data storage has changed over time and is therefore different when comparing EmberZNet PRO stack releases. The data storage also differs between the EM2xx and EM35x devices. Please see Appendix A – Channel Calibration Data Storage Information for byte configurations and acceptable value ranges for various EmberZNet PRO stack releases for both the EM2xx and EM35x devices.

The byte configuration for EM35x devices for stack releases of EmberZNet PRO 4.5 and later is as follows:

- byte 0: LNA Temperature at Calibration time; Reset=0x7f, Minimum=0xd8 (-40ºC), Maximum=0x55 (+85ºC) (8-bit signed)
- byte 1: Modulation DAC; Reset=0x80, Minimum=0, Maximum=0x7f (7 bit)
- byte 2: Modulation DAC Temperature at Calibration Time; Reset=0x7f, Minimum=0xd8 (-40ºC), Maximum=0x55 (+85ºC) (8-bit signed)
- byte 3: Low Noise Amp; Reset=0x80, Minimum=0, Maximum=0x3e (6 bit, LSB always 0)

The byte configuration for EM2xx devices for stack releases of EmberZNet PRO 4.2 and later is as follows:

- byte 0: VCO at LNA cal; Reset=0xff, Minimum=0, Maximum=0x3f (6 bit)
- byte 1: Modulation DAC; Reset=0x80, Minimum=0, Maximum=0x3f (6 bit)
- byte 2: Unused; Reset=0x7f
- byte 3: Low Noise Amp; Reset=0x80, Minimum=0, Maximum=0x3e (6 bit, LSB always 0)

For example, for the Modulation DAC (byte 1) anything above 0x3f for EM2xx and 0x7f for EM35x is invalid with the exception of 0x80, which indicates that the channel has not yet been calibrated.

Note: For stack releases EmberZNet PRO 3.5.1 and earlier for EM2xx, byte 2 was reserved for channel filter calibration data. As of stack release EmberZNet PRO 4.2 for EM2xx, channel filter calibration data is now stored separately within the token system as CAL_FILTER (0xd244). In either case, the reset value is 0x80, minimum value is 0, and maximum value is 0x1f. The channel filter calibration is valid only for EM2xx devices, while the Modulation DAC Temperature at Calibration Time is valid only for EM35x devices.

Any values equal to 0 or greater than or equal to the maximum value indicate a problem, so the acceptable ranges are as follows:

EM35x for stack releases of EmberZNet PRO 4.5 and later:

- bytes 0,4,8,...,60: values 0xd8-0x55 are good, 0x7f is uncalibrated
- bytes 1,5,9,...,61: values 0x01-0x7e are good, 0x80 is uncalibrated
- bytes 2,6,10,...,62: values 0xd8-0x55 are good, 0x7f is uncalibrated
- bytes 3,7,11,...,63: values 0x01-0x3d are good, 0x80 is uncalibrated

EM2xx for stack releases of EmberZNet PRO 4.2 and later:

- bytes 0,4,8,...,60: values 0x01-0x3e are good, 0xff is uncalibrated
- bytes 1,5,9,...,61: values 0x01-0x3e are good, 0x80 is uncalibrated
- bytes 2,6,10,...,62: unused, reset value 0x7f
- bytes 3,7,11,...,63: values 0x01-0x3d are good, 0x80 is uncalibrated

Appendix B – Example Test Data presents an example of expected calibration data distribution across a sample of boards for the EM2xx devices.

Note: When programming a device for test or retest of a DUT, use the --erase option to ensure that all previous calibration data is erased. Silicon Labs recommends erasing the flash contents of a device before testing to ensure the calibration is executed at the time the device is tested.

SILICON LABS

2.3   Supply Current Test

The Supply Current Test verifies that current consumption is valid for each mode of operation for the DUT. The modes of operation are set through the serial interface and include transmit mode (multiple power levels, normal mode, and boost mode), receive mode (normal mode and boost mode), and sleep modes (radio sleep and deep sleep). If there is excessive current draw, the DUT fails this test and does not proceed with further testing. This test is especially important for devices that will be used in battery-operated applications, as these measurements are an effective predictor of battery life.

For measuring sleep mode current draw, a multi-meter is required with the ability to measure less than 1 µA.

2.4   Quick Verify of Transmit and Receive Test

The Quick Verify of Transmit and Receive Test quickly verifies that the DUT transmits valid packets to the Reference Node and receives valid packets from the signal generator. This can be tested on any single channel in the available frequency band. If either of these checks fails, the DUT fails this test and does not proceed with further testing. This test identifies hardware that does not require full characterization testing due to a major manufacturing defect.

2.5   Transmit Power Test

The Transmit Power Test verifies that the power level of the transmitter is at the appropriate level and within a specified range. The power output is measured with the power meter at multiple power levels and in both normal and boost modes to confirm power output accuracy at various coded settings. The serial command interface can include a function that enables continuous waveform (CW) or unmodulated tone to be transmitted for ease of measuring these power levels. Silicon Labs recommends that this test be performed over a subset of the frequency band to record trends in power output versus frequency.

The transmit power output can be measured with a spectrum analyzer or a power meter.

2.6   Transmit Frequency Test

The Transmit Frequency Test verifies the crystal accuracy and valid transmission frequency of the DUT. The CW tone is again used for this transmission. Silicon Labs recommends that this test be performed over a subset of the frequency band to record trends versus frequency.

The transmit frequency can be measured with a spectrum analyzer or a frequency counter. The measured frequency will differ depending on the channel selected. When measuring frequency offset versus channel, a saw-tooth pattern is observed, such that the frequency of every 3rd channel is repeated. This is due to the 11.71 kHz resolution of the synthesizer. 11.71 kHz correlates to 4.9 ppm, which means errors of up to ±2.4 pmm will occur from the synthesizer setting. Silicon Labs recommends measuring the true frequency error by selecting channels 0xD (2415 MHz), 0x10 (2430 MHz), 0x13 (2445 MHz), 0x16 (2460MHz), and/or 0x19 (2475 MHz), where the frequency error due to the resolution of the synth is 0.

As of EmberZNet PRO 4.6 GA, Silicon Labs released a feature to use a synthesizer frequency offset to adjust the 24 MHz oscillator frequency to accurately center IEEE-802.15.4 channels. Using nodetest command setSynOffset, an offset is applied to the radio synthesizer frequency and is used to compensate for variations in 24 MHz crystals. The MFG_SYNTH_FREQ_OFFSET token is used to permanently store this offset value. The nodetest command does not set the token, it only adjusts the synthesizer offset in RAM, so changes only have effect until the next reset. The token must be written through a separate API call or em2xx_load/em3xx_load. The manufacturing library also contains this functionality. For more information on this token and feature, refer to Table 6 and Table 7 in this document and the Manufacturing Token section of one of the following documents:

- AN702, Bringing Up Custom Devices: For the EM250 SoC Platform
- AN704, Bringing Up Custom Devices: For the EM260 Co-Processor
- AN710, Bringing Up Custom Devices: For the EM35x SoC Platform

AN700

2.7   Transmit Error Vector Magnitude Test

The Transmit Error Vector Magnitude (EVM) test verifies that the device's EVM is within specified limits. The EVM is measured with a spectrum analyzer. Either a transmit packet or transmit stream command is used for this transmission, as the spectrum itself is analyzed. Silicon Labs recommends that this test be performed over a subset of the frequency band to record trends versus frequency.

Some spectrum analyzers are able to measure EVM as a standalone instrument, while other spectrum analyzers require a PC software tool to provide EVM measurement details.

2.8   Transmit Sweep Test

The Transmit Sweep Test verifies transmission of valid packets from the DUT to the Reference Node at all channels or a subset of channels across the frequency band. The Reference Node is put into receive mode while the DUT transmits 100 packets to the Reference Node for each channel, with an attenuation between nodes that translates to a strong signal, approximately 60 dB attenuation between devices. Please refer to the specific radio chip data sheet for more information.

Packet success rate is measured at each channel. The packet success rate percentage is defined as the number of packets received divided by the number of packets transmitted and then multiplied by 100. For the Transmit Sweep Test, anything below 100% packet success rate is flagged as a failure, as this test is conducted at a signal level where all packets should be received. This test confirms that there are no frequency-dependent issues with transmit mode.

2.9   Receive Sweep Test

The Receive Sweep Test is similar to the Transmit Sweep Test. It verifies reception of valid packets at the DUT from the signal generator at all channels or a subset of channels across the frequency band and at two receiver input power levels (a strong signal level, approximately -50 dBm, and a level closer to the edge of sensitivity performance, approximately -90 dBm).

The DUT is put into receive mode while the signal generator transmits 100 packets for each channel. Packet success rate is measured at each channel/receive input level. Any packets missed at the strong signal level are considered a failure, while the failure threshold at the lower input level can be at a lower percentage, depending on the expected sensitivity of the radio. Please refer to the data sheet of the radio chip for details related to receive sensitivity. This test should be performed over the full operating band to record trends versus frequency.

Silicon Labs recommends using a signal generator to transmit packets to a DUT, as the signal generator allows for easily configuring the receive input power level at the DUT.

2.10   Receive Sensitivity Test

The Receive Sensitivity Test determines the receiver sensitivity of the DUT. The DUT is placed in receive mode with 1,000 valid packets being sent from the signal generator for each channel. The power level should begin at some level before the 1% Packet Error Rate (PER) threshold. Please refer to the data sheet for the specific radio chip for more information on the sensitivity. The PER is measured until the receiver input power level corresponding to 1% PER is determined. This test should be performed over a subset of the operating band to record trends versus frequency. During the characterization testing phase Silicon Labs recommends that the actual sensitivity level be determined, while at high volumes the receive sensitivity specification can be set as the low limit and be used as a single power level for ease of testing.

Silicon Labs recommends using a signal generator to transmit packets to a DUT, as the signal generator allows for easily configuring the receive input power level at the DUT.

SILICON LABS

2.11   Receive Drill-Down Test

The Receive Drill-Down Test determines the receiver sensitivity of the DUT by collecting data to determine the receiver roll-off curve. The DUT is placed in receive mode with 100 valid packets being sent from the signal generator for each channel. The power level should begin at some level before the 1% PER threshold. Please refer to the data sheet for the specific radio chip for more information on the sensitivity. The packet success rate is measured for all input powers selected for testing. Silicon Labs recommends that enough input power levels are selected to ensure that the data collected includes both 100% and 0% packets received. This allows for a complete roll-off curve to be observed. This test should be performed over a subset of the operating band to record trends versus frequency. Silicon Labs recommends this test for characterization testing so that the roll-off is quantified and understood but can be omitted at higher volumes.

Silicon Labs recommends using a signal generator to transmit packets to a DUT, as the signal generator allows for easily configuring the receive input power level at the DUT.

2.12   Receive Signal Strength Indicator (RSSI) Test

The RSSI Test measures the RSSI value for a single channel and known receiver input power level. The RSSI is determined by receiving a valid packet from the signal generator and reading the RSSI value through the serial command interface. The DUT is placed into receive mode while the signal generator transmits a single packet and the RSSI measurement is averaged to determine RSSI value. This single data point is measured to verify that the RSSI pin for the radio chip is connected and that RSSI is reporting a valid level. The RSSI operation of the chip itself is validated at the chip testing level and is not tested here.

Silicon Labs recommends using a signal generator to transmit packets to a DUT, as the signal generator allows for easily configuring the receive input power level at the DUT.

2.13   External 32 kHz Crystal Test

The operation of the external 32 kHz crystal (if applicable) should be verified. Nodetest includes the command `rtc`, which verifies that the crystal is operating and running properly.

2.14   Peripherals Test

Various peripherals, if applicable, can be tested through the serial port. These include anything that may be accessed through the ADC or GPIO (general purpose I/O) on the micro. For example, an LED is often tied to a GPIO pin for some status to be alerted. The state of the LED can be modified by changing the level of the GPIO pin. Another example is reading an ADC pin for a particular voltage level that corresponds to the status of a peripheral such as a temperature sensor or an accelerometer. Any peripheral accessible through GPIO or the ADC should be tested to ensure valid functionality.

3 Test Recommendations

This section outlines the various tests that can be run on the hardware product, which tests Silicon Labs recommends to be run, and the channel and power mode selection for each test in each phase.

3.1   Characterization Testing

Characterization testing is recommended for early production stages. In this phase of testing, the hardware is characterized on all 16 ZigBee channels or a subset of these channels, as well as at various transmit output power levels or receiver input power levels. This phase fully characterizes the hardware that is being developed, determines the tests to be executed in manufacturing test, determines the test limits of these tests, and flushes out any manufacturing or process issues that might be present.

AN700

Silicon Labs recommends that the tests outlined in Table 1 be conducted in the characterization phase of testing. This table, and the similar tables that follow in subsequent sections of this document, list the various tests that could be run on these devices, which tests Silicon Labs recommends be run, and the channel and power mode selection for each test in each phase.

Note: An X in these tables represents a test that is recommended for this phase of testing.

Table 1. Characterization Test Recommendations

Test	Run?	Channel				Power Mode	
		Mid	Low-Mid-High	Subset	All	Normal	Boost
Serial Communication	X						
Channel Calibration	X				X		
Supply Current	X	X				X	X
Transmit/Receive Verify	X	X				X	
Transmit Power	X		X			X	X
Transmit Frequency	X		X			X	X
Transmit EVM	X		X			X	X
Transmit Sweep	X				X	X	
Receive Sweep	X				X	X	
Receive Sensitivity	X		X			X	X
Receive Drill-Down	X		X			X	X
RSSI	X	X				X	
External 32kHz Crystal	X						
Peripherals	X						

3.2 Low Volume Manufacturing Test

Low-volume manufacturing test is usually a subset of the characterization testing. A subset of the 16 ZigBee channels or transmit output power levels can be tested to reduce the test time without compromising test coverage. For example, one channel/power level combination (likely mid-band at max power) can be measured for transmit power and frequency. Also, receive drill-down can be omitted and receive sensitivity can be run in its place, where a certain packet-success rate is expected at mid-band for a given input power level.

The results from the characterization phase of testing help determine not only what should be tested in the manufacturing phase but also the test limits to be applied to certain tests. For example, if a particular test does not fail at all during the characterization phase, it can be omitted from the manufacturing phase altogether. Also, if it is determined that a particular test will fail all channels if it fails at all, testing can be reduced from all channels to a single channel, most likely mid-band.

SILICON LABS

3.2.1 Test Recommendations

Table 2 lists the tests Silicon Labs recommends be conducted in the low-volume manufacturing phase of testing.

Table 2. Low Volume Manufacturing Test Recommendations

Test	Run?	Channel				Power Mode	
		Mid	Low-Mid-High	Subset	All	Normal	Boost
Serial Communication	X						
Channel Calibration	X				X		
Supply Current	X	X				X	X
Transmit/Receive Verify	X	X				X	
Transmit Power	X	X				X	
Transmit Frequency	X	X				X	
Transmit EVM							
Transmit Sweep	X			X		X	
Receive Sweep	X			X		X	
Receive Sensitivity	X	X				X	
Receive Drill-Down							
RSSI	X	X				X	
External 32kHz Crystal	X						
Peripherals	X						

3.2.2 Test Times

The typical test time that can be achieved in the low-volume manufacturing test phase is three minutes per board. If the devices are preprogrammed, the overall test time can be reduced to less than three minutes. Program times vary depending on the flash memory size of the microprocessor. Note that programming may be included at both the front end (test application) and back end (final application) of the process. If you do not want to perform multiple programming steps during this phase of testing, Silicon Labs recommends that you include the manufacturing library in the final application.

3.2.3 Setting Test Limits

The results of the characterization phase of testing help determine how the limits are set for low-volume manufacturing test. Other factors in setting limits are customer application and manufacturing variation. For example, if an application specifies only a certain amount of dynamic range, perhaps limits will be relaxed to allow for this. Manufacturing variation can also be a factor in setting limits. For example, if the performance of the board is sensitive to particular components, it is important to account for any performance variation that may be seen with these particular components.

3.2.4 Full Characterization Sampling

It is important to continue to fully characterize samples from each production run to ensure that nothing in the process has shifted, causing a difference in the overall performance of a production run compared to a previous run. The size of this sample can be determined by the manufacturer, but Silicon Labs recommends this full characterization sampling for additional test coverage and process control at volume testing.

3.3 High-Volume Manufacturing Test

High-volume manufacturing testing is much simpler than characterization testing or low- volume manufacturing testing. The hardware design and manufacturing process have already been proven, so the product now just requires a quick "go/no go" functional test to verify operation.

3.3.1 Test Recommendations

Silicon Labs recommends that the tests in Table 3 be conducted in the high-volume phase of testing.

Table 3. High-Volume Test Recommendations

Test	Run?	Channel				Power Mode	
		Mid	Low-Mid-High	Subset	All	Normal	Boost
Serial Communication	X						
Channel Calibration	X				X		
Supply Current	X	X				X	X
Transmit/Receive Verify	X	X				X	
Transmit Power							
Transmit Frequency							
Transmit EVM							
Transmit Sweep							
Receive Sweep							
Receive Sensitivity	X	X				X	
Receive Drill-Down							
RSSI							
External 32kHz Crystal	X						
Peripherals	X						

3.3.2 Test Times

The typical test time that can be achieved in the high-volume manufacturing test phase is less than one minute per board. This assumes that devices are preprogrammed with the customer application and that the customer application uses the manufacturing library for invoking test modes.

3.3.3 Setting Test Limits

Since the test environment in the high-volume manufacturing phase is different from the test environment in the low-volume phase, setting the limits is also done differently. The test limits for the basic transmit and receive tests are dependent on the fixed attenuation between the Golden Node and the DUT, as well as the variation in over-the-air results. Silicon Labs recommends that customers run a sample size of boards through testing to determine these test limits.

3.3.4 Full Characterization Sampling

Even in high-volume testing, it makes sense to fully characterize samples from each production run to ensure that the process has not shifted in any way. The size of this sample can be determined by the manufacturer, but this full characterization sampling is recommended for additional test coverage at high volumes.

SILICON LABS

4 Test Architecture and Equipment

The following sections detail the recommended test architecture and equipment for each phase of testing, discuss test results and their dependence on test setup, and describe the typical manufacturing faults detected in manufacturing test.

4.1   Characterization Testing

The architecture and test equipment used in the characterization stage of testing is more comprehensive than that of the volume manufacturing stages.

4.1.1 Test Architecture

Figure 1 shows an example of the interfaces of the test equipment to the DUT. The test equipment can be controlled by test software through the General Purpose Interface Bus (GPIB) or Recommended Standard 232 (RS-232). The DUT may be controlled through either RS-232 or Virtual UART from a telnet connection to a Debug Adapter (ISA or ISA3). Any number of DUTs may be tested at once, but this number is dependent on serial ports available and will affect the selection of the power splitter and matrix switch hardware described in Figure 1.
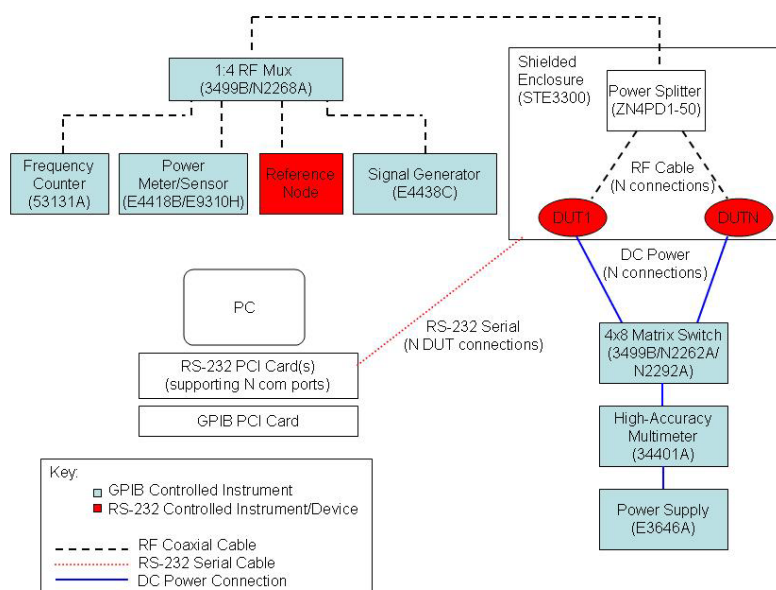
Figure 1. Architecture Example of the Characterization Test

Note: When using Virtual UART instead of RS-232, there is no need for RS-232 PCI cards in the test PC. Instead, some number of Debug Adapters (one per DUT) is required to drive the Virtual UART functionality and therefore Power over Ethernet (PoE) network hubs may be required. Please refer to the Debug Adapter technical specifications (TS1 for EM2xx and TS7 for EM35x) for more information on interfacing to the Debug Adapters.

4.1.2 Recommended Equipment

The recommended characterization test setup uses Agilent test equipment for the basic radio frequency (RF) measurements and current measurements, as well as for supplying power to the DUT and switching the RF connections from the DUT to various types of measurement equipment. A Reference Node, as mentioned in the

characterization stage of testing, is any Ember radio communication module (RCM) configured with the same radio chip as the DUT that can be used to verify transmission of packets from the DUT.

Table 4 lists the basic set of test equipment Silicon Labs has used for characterization testing. Note that some of these test equipment models are no longer available or supported by the manufacturer. Please consult the equipment manufacturer for acceptable alternative or replacement models for these particular instruments.

Table 4. Recommended Characterization Test Equipment

Manufacturer	Part Number	Description	Purpose
Agilent	E4418B	Power Meter	Used with power sensor to measure transmit power of the DUT
Agilent	E9301H	Power Sensor	Used with power meter to measure transmit power of the DUT
Agilent	53131A	Universal Counter	Measures the transmission frequency accuracy of the DUT
Agilent	53131A-030	3GHz Option	53131A universal counter option for 3GHz to measure the transmission frequency accuracy of the DUT
Agilent	E4402B	Spectrum Analyzer	Used to verify transmit power, transmit frequency accuracy, and EVM of the DUT
Agilent	89601A	Spectrum Analyzer VSA Software	Software option for the E4402B that allows for measuring EVM
Agilent	E4438C-504	Signal Generator – 250kHz to 4GHz	Verifies reception of valid packets at the DUT
Agilent	E4438C-1E5	Time Base Option	Improves frequency accuracy of E4438C
Agilent	E4438C-602	Baseband Generator	Internal baseband generator, 64MSa memory with digital bus capability
Agilent	E4438C-005	Hard Drive Option	6GB hard drive for E4438C (optional for storage beyond the 2.8MSa provided by E4438C-602)
Agilent	E3646A	Dual Output Power Supply	Powers the DUT and Reference Node
Agilent	34401A	Digital Multimeter	Verifies current consumption of the DUT in various modes of operation
Agilent	3499B	Data Acquisition Switch Unit	Supplies power to the DUT and switches RF connection
Agilent	N2262A	4x8 Matrix Module	3499B module; supplies power to the DUT
Agilent	N2268A	3.5GHz 1:4 RF Mux Module	3499B module; switches RF connection from the DUT to the Reference Node, frequency counter, power meter, or signal generator
Agilent	N2292A	Screw Terminal Block	Terminal block accessory for N2262A 3499B module
Mini-Circuits	ZN4PD1-50	1:4 Multiplexer Splitter	Splits power from test equipment to a maximum of 4 DUTs; alternative parts can be used for more or fewer DUTs
Ramsey Electronics	STE3300	Shielded RF Enclosure	Provides RF isolation for the DUT during testing
Silicon Labs	N/A	Reference Node	Known good radio module that is used as a receiver reference node in DUT transmit tests; manufacturer can be Silicon Labs, a Silicon Labs customer, or a Silicon Labs module partner

SILICON LABS

Note: Silicon Labs recommends the spectrum analyzer (E4402B or similar model number) and VSA software option (89601A or similar model number) for EVM testing and for general hardware debug. The spectrum analyzer may also be used in place of a power meter and frequency counter in characterization testing to measure transmit power and transmit frequency offset. In volume manufacturing test, however, the power meter and frequency counter are a much more cost-effective method of measuring transmit power and transmit frequency offset than the spectrum analyzer.

4.1.3 Configuring Equipment

Some test equipment needs to be configured specifically for 802.15.4 radio communications. For the signal generator, for instance, the specific packet needs to be configured. See Appendix C – Signal Generator Data Files and Configuration for more information on downloading waveform files to the signal generator and configuring the signal generator for valid 802.15.4 packets.

There are various options for triggering the signal generator to transmit packets to the device. The configuration noted in Appendix C – Signal Generator Data Files and Configuration expects a single trigger for each packet. The signal generator data file can also be configured to transmit some number of packets for each trigger event. Please consult the equipment manufacturer for more information on creating these data files and the various triggering options.

One example of triggering the signal generator is to provide a TTL-level trigger signal at the Pattern Trigger Input port on the back of the E4432B/E4438C unit. In some of Silicon Labs' test set-ups, the Transmit pin of a PC's RS-232 port serves as the trigger signal. The PC's RS-232 port is connected to a TTL-to-RS-232 converter, whose transmit pin (and ground pin) is connected to the Pattern Trigger Input port of the signal generator. The automated test sends characters (for example, the letter 'p', which in ASCII is 0x70) to provide a single pulse on the serial port. Configuring the baud rate appropriately, this serial port can be used to send characters to trigger the signal generator at various rates for testing.

4.1.4 Low-Cost Alternative to Signal Generator (Golden Node)

For some customers, adding a signal generator to manufacturing test is not desired or not possible due to cost concerns. In this case, Silicon Labs recommends using a Golden Node (a known good device that can be used in test for repeatable measurements) with TCXO (temperature controlled/compensated crystal oscillator). The TCXO allows for frequency accuracy when using the Golden Node as a known good transmitter source for DUT receive tests. Using a Golden Node without TCXO would present issues with test accuracy, as the crystal would drift over time and temperature. The Golden Node will have known transmitter performance (0 dBm +/- 0.5 dBm) across voltage and temperature, allowing for test repeatability.

4.1.5 RF Test Interface Examples

The type of interface to the DUT and the RF shielded test enclosure selected for testing can determine the accuracy and repeatability of the measurements. The Ramsey STE3300 test enclosure listed in Table 4 is recommended.

It is important to pay special attention to the RF test interface to the DUT because of the sensitivity of these signals. For example, a product with a 50-Ohm terminated Subminiature Type A (SMA) connector populated on the board can be connected directly to a coaxial cable with a known loss. Repeatability in this scenario is very good.

Another example is a product with an embedded antenna that was designed with test points for RF and ground can be connected with a pogo-pin style RF probe. The path loss from the RF test point to the cabled connections of the setup can be calibrated to determine accurate performance. The repeatability of this setup is dependent on the board layout, in the sense that the RF and ground signals should have test points in close proximity to one another. The repeatability is also more dependent on the shielded enclosure in this case, because the RF signal is exposed at the test point rather than enclosed within an SMA connector as in the first example.

AN700

As a final example, a product with an embedded antenna can be tested over the air within an enclosure. A reference antenna would be used within the enclosure to feed back the RF signal to the RF Mux. The path loss over the air from the reference antenna to the DUT antenna can be calibrated to determine accurate performance. Fixed position of the DUT and position of the DUT and reference antennas are crucial to getting repeatable results.

4.2 Low-Volume Manufacturing Testing

The architecture and test equipment used in the characterization stage of testing is more comprehensive than that of the volume manufacturing stages.

4.2.1 Test Architecture

The test architecture in the low-volume phase of testing is very similar to that of the characterization phase.

4.2.2 Recommended Equipment

The equipment Silicon Labs recommends for the low-volume manufacturing phase of testing is very similar to that for the characterization phase. Some of the equipment may be removed depending on the tests selected for this phase.

4.2.3 Interference

There are many RF devices in a test environment, such as wireless networks, microwave ovens, and mobile phones. For this reason, it is important to maintain RF isolation of the DUT from these sources of interference. It is also important to maintain RF isolation between multiple stations. For example, the equipment for test station A should be able to communicate only with a DUT from test station A and not with a DUT from test station B. Likewise, the equipment for test station B should be able to communicate only with a DUT from test station B and not a DUT from test station A. If these stations are not isolated from each other, the DUT will not be uniquely configured and multiple boards could share unique configuration information.

4.3 High-volume Manufacturing Testing

The architecture and test equipment used in the characterization stage of testing is more comprehensive than that of the volume manufacturing phases.

4.3.1 Test Architecture

The test architecture in the high-volume phase of testing can be structured any number of ways depending on customer preference. One approach is to develop this test with a Golden Node programmed with an application that allows the DUT to obtain unique configuration information from the Golden Node through packets transmitted and received. This test may be limited to configuring one DUT at a time. The Golden Node initiates communication with the unconfigured DUT and sends unique configuration information to the DUT. The Golden Node can then interface with the PC application to configure the DUT uniquely. Then the DUT can reboot itself and be ready for testing or application operation. A basic transmit/receive test can then be run. The test architecture for this phase of testing is shown in Figure 2.

SILICON LABS
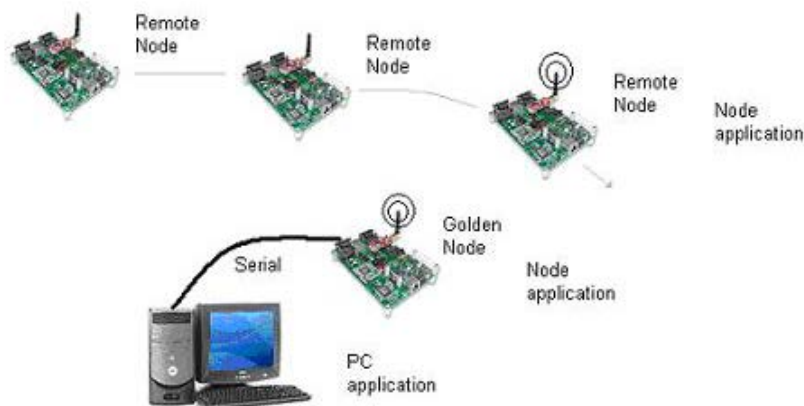
Figure 2. Architecture Example of a High-Volume Manufacturing Test

Another approach is to develop a test with a serial interface to both the Golden Node and the DUT. This may be more straightforward to develop but requires a more extensive set up and use of a UART (either serial or virtual UART through a Debug Adapter (ISA or ISA3)).

4.3.2 Recommended Equipment

The equipment is based on the tests selected for this phase of testing, as well as the preference of the customer. No matter the approach, it is important that, at the very least, the Golden Node (and in some cases the DUT) have an interface to a PC or server so that it can log test information like calibration data, serial numbers, EUI, test results, and so on.

4.3.3 Interference

A facility running multiple test stations maintains RF isolation among the stations, as previously detailed. In this test phase, the only interface to the DUT is power, as all communication between the Golden Node and the DUT is done through the radio. All of the test equipment used in the characterization testing can be removed from the process, unless a customer decides to continue to test certain functionality on the board that requires test equipment.

4.4 Manufacturing Coverage

Manufacturing testing not only determines the tests that should be included in the manufacturing test phase, but it also helps detect manufacturing failures. The following sections provide some examples of typical manufacturing faults and how they are detected in testing.

4.4.1 Excess Solder on Ember IC

An excess amount of solder will result in any number of failures in test. Excess solder will cause coplanarity issues and prevent some pins on the device from making contact with the pads on the board, as well as cause shorting of pins. This will likely result in programming failures, serial communication failures, and RF performance problems, all of which are detected in manufacturing test.

4.4.2 Insufficient Solder on Ember IC

An insufficient amount of solder will result in any number of failures in test. Insufficient solder will prevent some pins on the device from making contact with the pads on the board. This will result in programming failures, serial communication failures, and RF performance problems, all of which are detected in manufacturing test. Insufficient solder on the ground pad underneath the device will result in degraded RF performance, most likely in the receiver. This type of issue will be detected in either the Receive Drill-Down Test or Receive Sensitivity Test. In some drastic

AN700

cases where there is no solder on the ground pad and thus no ground connection from the device to the board, the device will not function and would either fail to program or fail to calibrate properly.

4.4.3 Wrong Component

An incorrect component or component value will alter the performance of the board. For example, if a crystal tuning cap is the incorrect value, the Transmit Frequency Test will detect this failure because the frequency will be outside the specified limits. If a component in the matching network is incorrect, this will affect the transmit power and will be detected in the Transmit Power Test. If a decoupling capacitor is the wrong value, it will affect the receiver performance of the device and will be detected in either the Receive Drill-Down Test or the Receive Sensitivity Test.

4.4.4 Missing Component

A missing component will also alter the performance of the board. For example, if there is a component missing from the RF path, there will be a major degradation in the transmit output power and overall receive sensitivity of the board. Also, if a component is missing that affects power distributed to the microprocessor or radio chips, the board will not be able to communicate serially in the case of the micro; the microprocessor will not configure the radio to transmit or receive in the case of the radio. The Component Removal Study section presents details for the effects of missing components on board performance as it pertains to manufacturing test.

4.4.5 Solder Shorts or Opens

A solder short or open on any component or device will cause any number of failures in manufacturing test. Any short or open in the RF circuitry of the board will cause degradation in performance and will be detected in various RF tests. A short or open in the DC circuitry or programming circuitry will prevent the device from programming properly and/or communicate properly with the test interface. For all of these cases mentioned, at least one test will detect a failure and flag this device as defective.

4.5   Component Removal Study

Results from a component removal study are presented below to help point out the manufacturing test failures that would be present if a particular component is missing or if there is an open connection for that component. This experiment was conducted on two (2) EM250 radio communication modules (130-0452-000 Revision B0). As shown in Figure 3, this design contains a Lattice RC Balun and Ceramic Antenna. The components selected were those in the RF circuitry and VREG_OUT circuitry. Decoupling capacitors were omitted in this study and remained populated for all tests, but they will in fact affect device performance if missing.

Table 5. Component Removal Study

Part	Component Description	Effect on Performance
C25	1 pF Harmonic Filter Shunt Capacitor	Change in transmit power versus frequency (2405MHz was 0.5-1 dB lower than 2480 MHz, now 1.5-2.5 dB lower); drop of 1-3 dB in transmit power output
L6	2.7 nH Harmonic Filter Series Inductor	Drop of 45-50 dB for both transmit power and receive sensitivity
C24	1 pF Harmonic Filter Shunt Capacitor	Change in transmit power versus frequency (2405 MHz was 0.5-1 dB lower than 2480 MHz, now 0-1 dB higher); no noticeable reduction in transmit power output
C2	1.8 pF Balun DC Blocking Capacitor	Drop of 45-50 dB for both transmit power and receive sensitivity
L2	5.1 nH Balun Inductor	Drop of 5-10 dB for both transmit power and receive sensitivity
C3	0.75 pF Balun Capacitor	Drop of 1-2 dB in boost mode for both transmit power and receive sensitivity
C5	0.75 pF Balun Shunt Capacitor	Drop of 1-3 dB for transmit power and no more than 1 dB for receive sensitivity
L3	5.8 nH Impedance Matching Inductor	Drop of 12-15 dB for transmit power and 4-7 dB for receive sensitivity

SILICON LABS

Part	Component Description	Effect on Performance
L1	5.1 nH 1.8 V Balun DC Feed Inductor	Drop of 30-40 dB for transmit power and no more than 2 dB for receive sensitivity
R1	0 ohm 1.8 V Balun DC Jumper Resistor	Drop of 30-40 dB for transmit power and no more than 2 dB for receive sensitivity
C1	8 pF 1.8 V Balun DC Decoupling Capacitor	Drop of 2-3 dB for transmit power and 4-7 dB for receive sensitivity
C7	27 pF Crystal Loading Capacitor	Frequency moved +1 MHz (+400 PPM); no longer able to transmit or receive packets
C8	27 pF Crystal Loading Capacitor	Frequency moved + 500 KHz (+200 PPM); no longer able to transmit or receive packets
R2	1 Ω VREG_OUT ESR Resistor for C9	No performance change; increased noise on 1.8 V plane
C9	10 µF VREG_OUT Stability Cap	No performance change; increased noise on 1.8 V plane
R6	1 Ω VCO Resistor	Device won't run, calibration not functional
R3	169 KΩ BIAS Resistor	Device won't run, calibration not functional
R4	0 Ω 1.8 V Digital Voltage Drop Resistor	Device won't run, digital circuitry not powered

Reference the schematic in Figure 3 to determine the components that apply to your design.
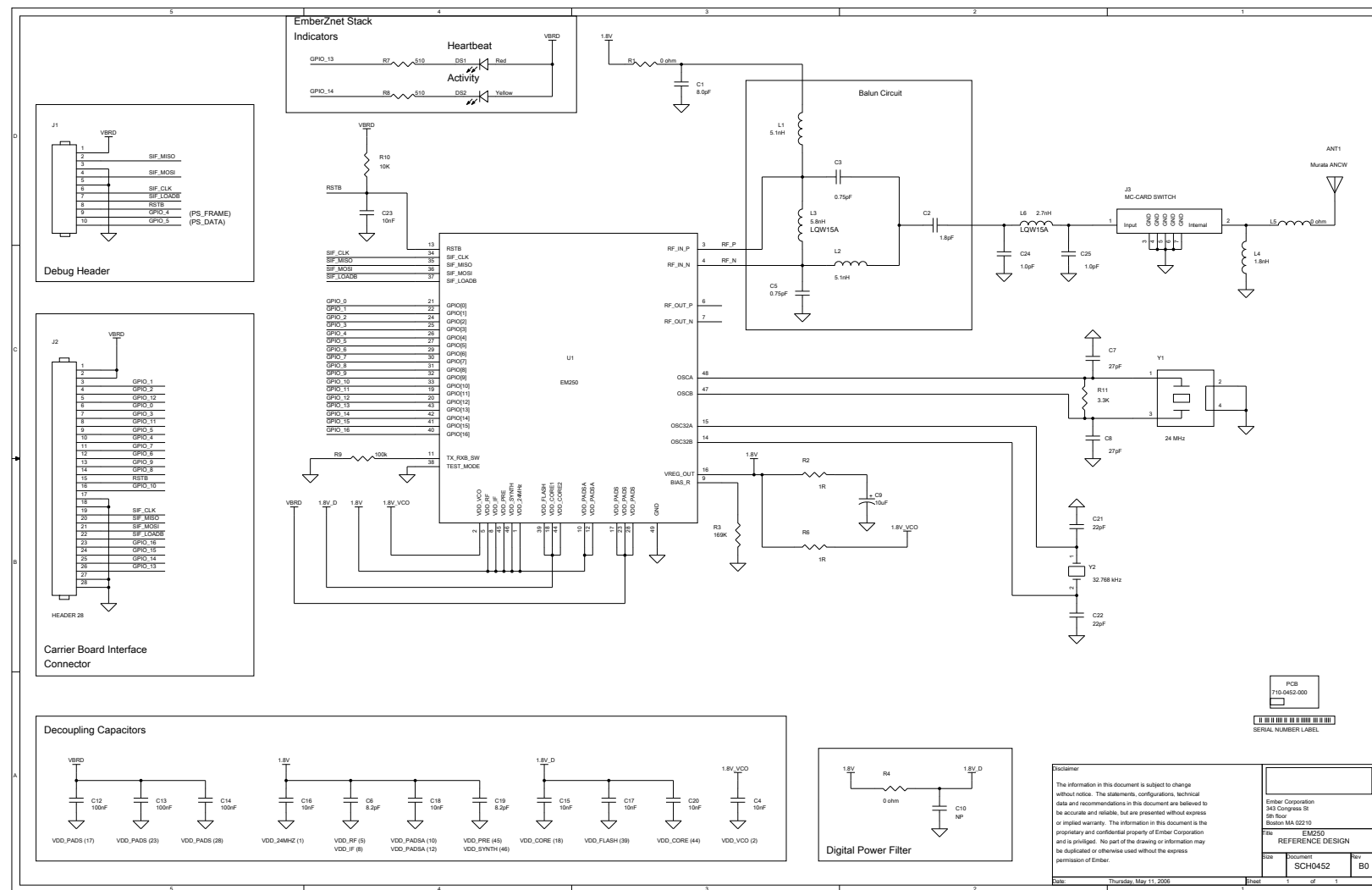
Figure 3. Schematic for EM250 Reference Design (130-0452-000 Revision B0)

SILICON LABS

5 Embedded Software Tools

Silicon Labs supports two different embedded software tools for manufacturing test: a standalone test application called nodetest, and a manufacturing test library that is used as a test mode within the customer application. The following sections describe these two embedded software tools.

Note: For stack releases EmberZNet PRO 4.0 and later, the standalone test application included in the release is nodetest. For all earlier stack releases, the standalone test application included in the release is rangetest. This document references nodetest in all standalone test application scenarios.

5.1 Standalone Test Application - nodetest

The characterization stage typically utilizes a standalone test application. Nodetest is a standalone test application that Silicon Labs recommends for any test stage in which the customer's application is not yet mature enough to include a test mode. Nodetest provides a serial command line interface to the Silicon Labs device.

Note: For the Ember SoC, the nodetest files can be found in the stack installer at /app/nodetest/. For the Ember NCP, the nodetest files can be found in the stack installer at /build/em260-images/nodetest/.

5.1.1 Nodetest Xommands

The nodetest application runs the serial port at 115200, so be sure that your baud rate has been set accordingly. The nodetest application optionally uses port 4900 of the Debug Adapter as a virtual UART for communicating with nodetest, so the port speed (baud rate) for communication is fixed (by the Debug Adapter) at 500,000 bps.

A carriage return initiates the nodetest application on reset or power-up. This displays the power-up prompt, which ends with the > (greater than) symbol. Table 6 lists the nodetest commands.

Table 6. Subset of Nodetest Commands

Command	Description
`?, help`	Prints the Help menu (which includes the full list of commands).
`calChannel`	Use `calChannel x` to switch to channel *x* and perform calibration. (Uses current channel if *x* is not specified.)
`setChannel`	Sets the channel (11 by default). For valid values, see the device's technical specification.
`getChannel`	Gets the channel). For valid values, see the device's technical specification.
`ledOff`	Use `ledOff x` to turn BOARDLEDx off.
`ledOn`	Use `ledOn x` to turn BOARDLEDx on.
`ledTest`	Cycles the application LEDs (at their default GPIOs on the applicable Breakout Board) until told to stop.
`buzzer`	Plays the piezo buzzer in the foreground (0) or background (1).
`rx`	Puts the device into receive (RX) mode on the current channel. Use `e` to exit receive mode.
`rtc`	Use `rtc` to test the operation of the external 32kHz crystal.
`shutdown`	Use `shutdown` to put the device into deep sleep mode.
`tokDump`	Dumps all known tokens and their values.
`tokRead`	`tokRead <key>` shows the contents of the token indexed by *<key>* as the stack will read it when run.
`loadToks`	Loads stack and application token defaults.
`initTokens`	Invokes top level token initialization.
`tokWrite`	`tokWrite <key>` writes a new value to the token indexed by *<key>* and prompts for each byte of data. Manufacturing tokens cannot be written with this command.
`tx`	Transmits the specified number of packets on the current channel (infinite if 0).
`setTxPower`	Sets radio transmit power to specified dBm. For valid values, see the device's technical specification.
`getTxPower`	Gets radio transmit power. For valid values, see the device's technical specification.
`setTxPowMode`	Use `setTxPowMode x y` for x=0 or 1 and y=0 or 1 to engage Boost mode (x=1) for the chip or switch to using the external PA (RF_TX_ALT_P/N) signal path (y=1).
`getTxPowMode`	Gets the transmit power mode settings for normal versus boost mode and power amplifier settings.
`txStream`	Performs a modulated carrier wave transmission on the current channel.
`txTone`	Performs an unmodulated carrier wave ("tone") transmission on the current channel.
`setSynOffset`	Use `setSynOffset x` to offset the synth frequency by x 11.7 kHz increments, where x is a signed integer. This command allows for tuning the 24 MHz crystal frequency in manufacturing (+1.397 MHz/-1.408 MHz range). This command applies the offset directly to the radio register and does not store data in the corresponding MFG_SYNTH_FREQ_OFFSET token. This token needs to be set separately to store and apply this offset.

5.2 Application Test Mode - Manufacturing Test Library

Silicon Labs recommends using the manufacturing test library for customers who are using mature applications, regardless of the testing phase. However, this library is typically used in the low-volume and high-volume phases of testing where additional programming steps add unnecessary test time. This library allows for accessibility to a test mode within the customer's application and removes the need for multiple application bootloads or multiple programming steps within the manufacturing process. There is a sample application (mfg-sample) available that demonstrates the use of this library. The following sections detail this manufacturing library.

Note: The manufacturing library files can be found in the stack installer at /app/mfglib for the Ember SoC and at /app/mfglib-host for the NCP Host.

SILICON LABS

5.2.1 Use Cases

This library can optionally be compiled into the customer's production code and run at manufacturing time. The application handles any interface to the library. This library cannot assist in hardware start up.

5.2.2 Use Overview

The manufacturing library provides APIs to relevant nodetest functionality that the customer application may call into. The APIs provide a reporting mechanism that can give the customer application whatever information is required to report the information back to the manufacturing test equipment. Unlike nodetest, each routine in the manufacturing library returns a status value of some sort, and provides for output arguments or callback functions that, if specified, will be filled in or called with relevant information. The library is not concerned with the actual reporting of the information. This is application specific, and can be handled at the application layer. Customers can access this library over many ports and protocols.

The sample application mfg-sample provides an example of how to access this library over the serial port of the Ember SoC. A host sample application, mfg-sample-host, is available for NCP host manufacturing support.

5.2.3 General Functions

The manufacturing library includes the functions that do the work and return the results and/or EmberStatus return codes. Customers can see how these functions are used or called by referencing the sample application. Customers only need to type the command over the correct port. The specifications for the manufacturing library APIs are detailed in the following documents:

- EmberZNet API Reference: For the EM250 SoC Platform (120-3016-000)
- EmberZNet API Reference: For the EM35x SoC Platform (120-3022-000)

The documents are available in HTML and PDF formats. The section that details the manufacturing library APIs is *Module Index*, subsection *EmberZNet Stack API Reference*, and subsection *Manufacturing and Functional Test Library*.

For the NCP, the manufacturing library APIs are detailed in document UG100, *EZSP Reference Guide*, in section 3.12 entitled "Mfglib frames."

5.2.4 Platform Support

For the SoC platforms, the communication to the library is dependent on the application, and the functionality could be accessible through either the SPI port or the ISA port. For the NCP platforms, the library commands are part of the EmberZNet Serial Protocol (EZSP) and can be driven through the host processor application.

5.2.5 Code Size

Silicon Labs commits to keeping the manufacturing library as small as possible. Currently it is just under 500 bytes for the SoC. The manufacturing library is included in the standard EZSP application for the NCP.

5.2.6 Sample Application Functions

The sample application mfg-sample-app enables this functionality through a serial command interpreter. The command interpreter is realized in the sample application. The command interpreter is serially driven. Table 7 lists the format of these commands related to these manufacturing library APIs.

AN700

Table 7. Sample Application Commands

Command	Description
`MFG`	Prints list of mfg commands.
`MFG START <want_callback>`	Shows how to use `mfglibStart()` to start use of `mfglib` test functionality and packet receive functionality. Receive handler callback is 0 for false, 1 for true.
`MFG END`	Shows how to use `mfglibStop()` to end use of `mfglib` test functionality.
`MFG TONE <[START][STOP]>`	Shows how to use `mfglibStartTone()` and `mfglibStopTone()` to transmit a tone over a radio.
`MFG STREAM <[START][STOP]>`	Shows how to use `mfglibStartStream()` and `mfglibStopStream()` to transmit a stream of random characters over a radio.
`MFG SEND <numPackets> test-packet <length>`	Shows how to use `mfglibSend()` to send a packet of (length) bytes of the test-packet payload (numPackets) times.
`MFG SEND <numPackets> random <length>`	Shows how to use `mfglibSend()` to send a packet of (length) bytes of a random payload (numPackets) times.
`MFG SEND <numPackets> message <first 8 bytes> <second 8 bytes>`	Shows how to use `mfglibSend()` to send packets of specified payload (numPackets) times .
`MFG CHANNEL SET <channel>`	Shows how to use `mfglibSetChannel()` to set the radio channel.
`MFG CHANNEL GET`	Shows how to use `mfglibGetChannel()` to get the radio channel.
`MFG POWER SET <power>`	Shows how to use `mfglibSetPower()` to set the radio transmit power level.
`MFG POWER GET`	Shows how to use `mfglibGetPower()` to get the radio transmit power level.
`MFG SYNOFFSET SET <steps>`	Shows how to use `mfglibSetSynOffset()` to set the radio synthesizer offset
`MFG SYNOFFSET GET`	Shows how to use `mfglibGetSynOffset()` to get the radio synthesizer offset
`MFG TOKEN WRITE SYNOFFSET`	Shows how to use `halCommonSetMfgToken()` to set the TOKEN_MFG_SYNTH_FREQ_OFFSET token using the value previously set with `mfglibSetSynOffset()`
`MFG TOKEN READ SYNOFFSET`	Shows how to use `halCommonGetMfgToken()` to read the TOKEN_MFG_SYNTH_FREQ_OFFSET token

5.2.7 Demonstration of Non-Library Features

In addition to the RF-related features, the sample application mfg-sample-app demonstrates network features. Table 8 lists and describes the features of the reference application.

Table 8. Non-Library Features

Feature	Description
`Application tokens`	A way for the sample application to set and read application tokens (mfg-sample-host only).
`Network functions`	A way for the sample application to perform network join and form functions.

The sample application includes additional non-library commands to demonstrate these features, many of which already have hal, token, or stack support which will be exploited by the sample application.

SILICON LABS

Table 9 lists and describes the non-library feature commands.

Table 9. Non-Library Feature Commands

Command	Description
HELP	Prints the Help menu list of commands.
VERSION	Prints the mfg-sample version number information.
INFO	Prints various information about the device and device status, including app version, network status, EUI-64 of device, mfg mode status, and stack status.
NETWORK	Prints the network help menu list of commands.
NETWORK FORM <channel> <power> <panid in hex>	Command for forming a network with channel and power settings for a particular panid in hexadecimal format.
NETWORK JOIN <channel> <power> <panid in hex>	Command for joining a network with channel and power settings for a particular panid in hexadecimal format.
NETWORK INIT	Command for initializing a network.
NETWORK LEAVE	Command for leaving a network.
NETWORK PJOIN <time>	Command for permitting joining a network for a period of time (in seconds).
RESET	Resets the device.

6 Conclusions and Summary

As you can see from the descriptions of each test phase within this document, the recommended tests and test flow are different when comparing characterization testing with manufacturing testing.

Table 10 lists the test recommendations by phase and Table 11 compares the test phases.

Note: In Table 10, C denotes tests recommended for characterization testing, L denotes tests recommended for low-volume manufacturing testing, and H denotes tests recommended for high-volume manufacturing.

Table 10. Test Recommendations by Phase

Test	Run?	Channel				Power Mode	
		Mid	Low-Mid-High	Subset	All	Normal	Boost
Serial Communication	CLH						
Channel Calibration	CLH				CLH		
Supply Current	CLH	CLH				CLH	CLH
Transmit/Receive Verify	CLH	CLH				CLH	
Transmit Power	CL	L	C			CL	C
Transmit Frequency	CL	L	C			CL	C
Transmit EVM	C		C			C	C
Transmit Sweep	CL			L	C	CL	
Receive Sweep	CL			L	C	CL	
Receive Sensitivity	CLH	LH	C			CLH	C
Receive Drill-Down	C		C			C	C
RSSI	CL	CL				CL	
External 32kHz Crystal	CLH						
Peripherals	CLH						

Table 11. Comparison of Test Phases

Step	Characterization	Manufacturing Low-Volume	Manufacturing High-Volume
Program bootloader (if applicable)	Functional Test	Functional Test or Preconfigured	Preconfigured
Program/load test application	Functional Test	Functional Test or Preconfigured	Manufacturing library within final application
Load stack information	Functional Test	Functional Test or Preconfigured	Preconfigured
Load manufacturing Information	Functional Test	Functional Test	Golden Node application
Load application information	Functional Test	Functional Test	Preconfigured
Verify DUT operation	Functional Test	Functional Test	Golden Node application
Program/load production application	Functional Test	Functional Test	Preconfigured

In the characterization phase of testing, all programming and configuration steps can be automated to occur within the test itself. In the low-volume manufacturing phase, some of these steps can be done before actual manufacturing. For example, the device can be preconfigured with the appropriate bootloader and/or test application. In the case of high-volume manufacturing, the test functions can be included in the production application as a test mode, as previously mentioned with the manufacturing library. The Golden Node application can be developed by the customer to configure the appropriate unique manufacturing information for each DUT.

SILICON LABS

7 Appendix A – Channel Calibration Data Storage Information

Over time, Silicon Labs has worked to improve channel calibration for both the Ember EM2xx and EM35x devices. With these improvements, the channel calibration data storage has changed at times and therefore there are differences in this data storage when comparing Ember stack releases and/or devices. This appendix provides some details regarding these differences.

7.1 EM2xx

7.1.1 EmberZNet PRO 3.5.1 and earlier
- Token creator: D243
- Channel Calibration Byte Configuration
  - byte 0: VCO at LNA cal; Reset=0xff, Minimum=0, Maximum=0x3f (6 bit)
  - byte 1: Modulation DAC; Reset=0x80, Minimum=0, Maximum=0x3f (6 bit)
  - byte 2: Channel Filter; Reset=0x80, Minimum=0, Maximum=0x1f (5 bit)
  - byte 3: Low Noise Amp; Reset=0x80, Minimum=0, Maximum=0x3e (6 bit, LSB always 0)
- Channel Calibration Valid Range of Values
  - bytes 0,4,8,...,60: values 0x01-0x3e are good, 0xff is uncalibrated
  - bytes 1,5,9,...,61: values 0x01-0x3e are good, 0x80 is uncalibrated
  - bytes 2,6,10,...,62: values 0x01-0x1e are good, 0x80 is uncalibrated
  - bytes 3,7,11,...,63: values 0x01-0x3d are good, 0x80 is uncalibrated

7.1.2 EmberZNet PRO 4.2 and Later
- Token creators: D243 (for VCO at LNA, Modulation DAC, and Low Noise Amp) and D244 (for Channel Filter)
- Channel Calibration Byte Configuration: 0xD243
  - byte 0: VCO at LNA cal; Reset=0xff, Minimum=0, Maximum=0x3f (6 bit)
  - byte 1: Modulation DAC; Reset=0x80, Minimum=0, Maximum=0x3f (6 bit)
  - byte 2: Unused; Reset=0x7f
  - byte 3: Low Noise Amp; Reset=0x80, Minimum=0, Maximum=0x3e (6 bit, LSB always 0)
- Channel Calibration Valid Range of Values: 0xD243
  - bytes 0,4,8,...,60: values 0x01-0x3e are good, 0xff is uncalibrated
  - bytes 1,5,9,...,61: values 0x01-0x3e are good, 0x80 is uncalibrated
  - bytes 2,6,10,...,62: unused, reset value 0x7f
  - bytes 3,7,11,...,63: values 0x01-0x3d are good, 0x80 is uncalibrated
- Channel Calibration Byte Configuration: 0xD244
  - byte 0: Channel Filter; Reset=0x80, Minimum=0, Maximum=0x1f (5 bit)
- Channel Calibration Valid Range of Values: 0xD244
  - byte 0: values 0x01-0x1e are good, 0x80 is uncalibrated
- Notable changes compared to previous releases
  - Channel Filter stored in separate token creator D244 and is a single value for all channels rather than one value for each channel

7.2 EM35x

7.2.1 EmberZNet PRO 4.2.0
- Token Creator: D243
- Channel Calibration Byte Configuration
  - byte 0: VCO at LNA cal; Reset=0xff, Minimum=0, Maximum=0x3f (6 bit)

AN700

- byte 1: Modulation DAC; Reset=0x80, Minimum=0, Maximum=0x7f (7 bit)
- byte 2: Modulation DAC Temperature at Calibration Time; Reset=0x7f, Minimum=0xd8 (-40ºC), Maximum=0x55 (+85ºC) (8-bit signed)
- byte 3: Low Noise Amp; Reset=0x80, Minimum=0, Maximum=0x3e (6 bit, LSB always 0)
- Channel Calibration Valid Range of Values
  - bytes 0,4,8,...,60: values 0x01-0x3e are good, 0xff is uncalibrated
  - bytes 1,5,9,...,61: values 0x01-0x7e are good, 0x80 is uncalibrated
  - bytes 2,6,10,...,62: values 0xd8-0x55 are good, 0x7f is uncalibrated
  - bytes 3,7,11,...,63: values 0x01-0x3d are good, 0x80 is uncalibrated
- Notable changes compared to previous releases
  - Byte 2 now used to store Modulation DAC Temperature at Calibration Time
  - Reset value for byte 2 changed from 0x80 to 0x7f

7.2.2 EmberZNet PRO 4.5.0 and Later

- Token Creator: D245
- Channel Calibration Byte Configuration
  - byte 0: LNA Temperature at Calibration Time; Reset=0x7f, Minimum=0xd8 (-40ºC), Maximum=0x55 (+85ºC) (8-bit signed)
  - byte 1: Modulation DAC; Reset=0x80, Minimum=0, Maximum=0x7f (7 bit)
  - byte 2: Modulation DAC Temperature at Calibration Time; Reset=0x7f, Minimum=0xd8 (-40ºC), Maximum=0x55 (+85ºC) (8-bit signed)
  - byte 3: Low Noise Amp; Reset=0x80, Minimum=0, Maximum=0x3e (6 bit, LSB always 0)
- Channel Calibration Valid Range of Values
  - bytes 0,4,8,...,60: values 0xd8-0x55 are good, 0x7f is uncalibrated
  - bytes 1,5,9,...,61: values 0x01-0x7e are good, 0x80 is uncalibrated
  - bytes 2,6,10,...,62: values 0xd8-0x55 are good, 0x7f is uncalibrated
  - bytes 3,7,11,...,63: values 0x01-0x3d are good, 0x80 is uncalibrated
- Notable changes compared to previous releases
  - Token creator changed from D243 to D245
  - Low Noise Amp Temperature at Calibration Time is tracked rather than VCO at LNA
  - Temperature at Calibration Time algorithms updated to use data from IC production test

7.3   Items of Note

- EM2xx Modulation DAC calibration data is 6 bits (0x3F max)
- EM35x Modulation DAC calibration data is 7 bits (0x7F max)

8 Appendix B – Example Test Data

Manufacturing test data from the Ember EM250 radio communication modules (130-0452-000 Revision B0) is presented here. The data presented was collected on boards tested using the Ember development kit module manufacturing test program. This collection of histograms presents the typical distribution of test results for a given sample of EM250 IC devices. Collecting data in this manner will help determine acceptable test limits in manufacturing test.

The histograms were created using Galaxy Examinator (www.galaxysemi.com).
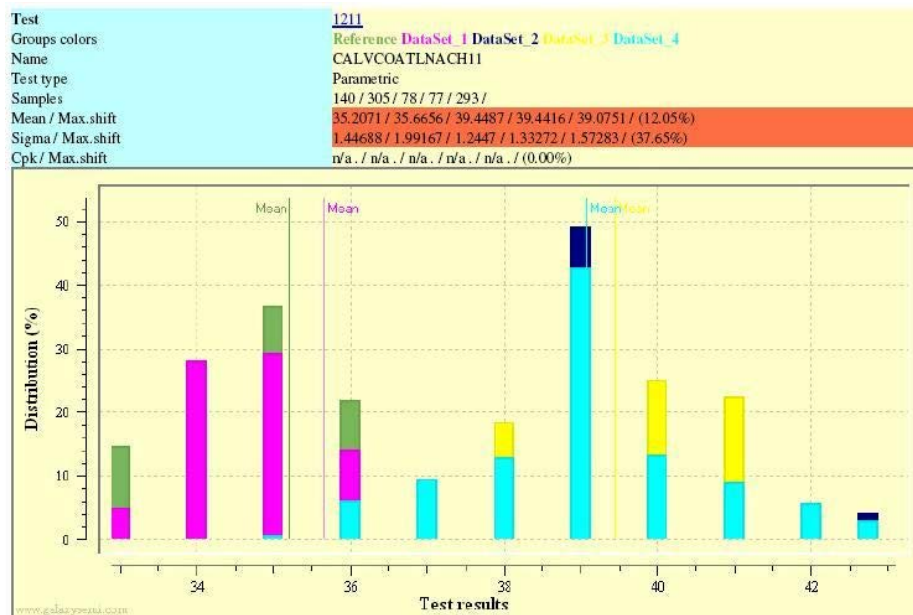
SILICON LABS

Figure 4. Comparison Histogram for VCO at LNA Calibration Channel 11, 5 Board Builds
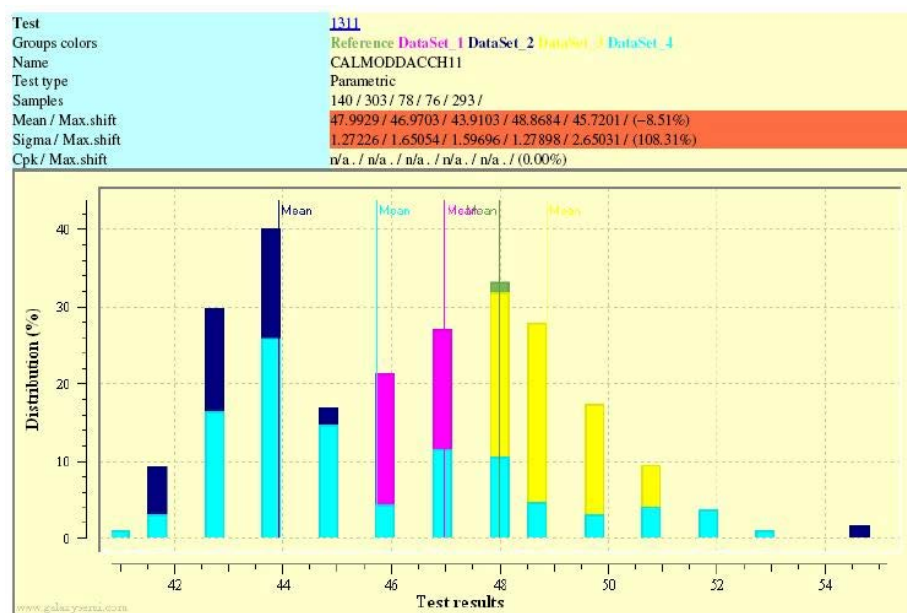

Figure 5. Comparison Histogram for Mod Dac Calibration Channel 11, 5 Board Builds
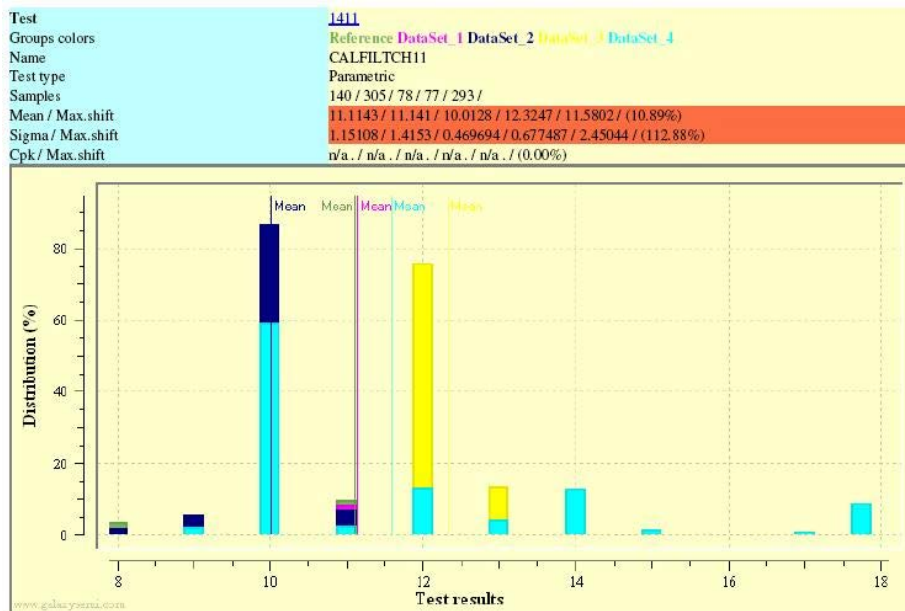
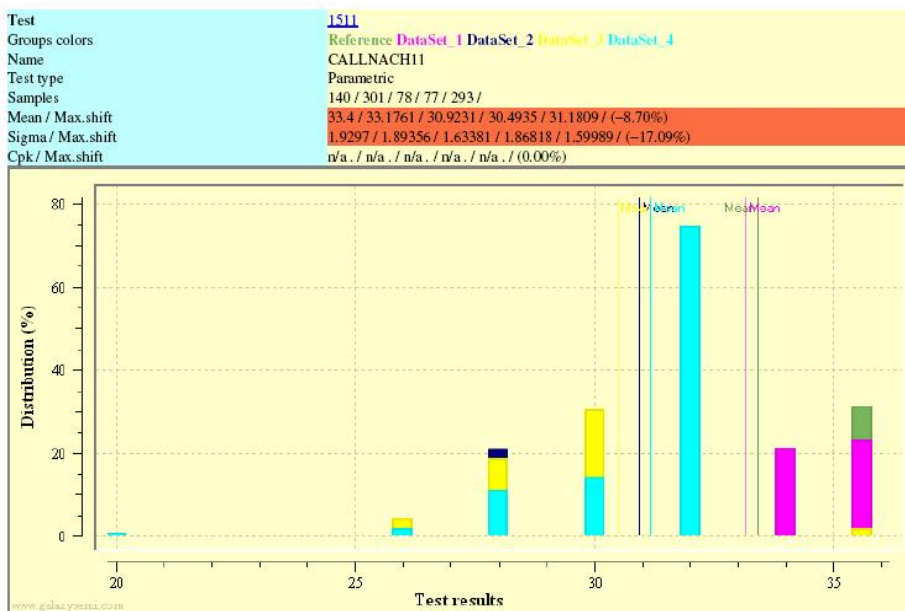**Figure 6. Comparison Histogram for Filter Calibration Channel 11, 5 Board Builds**

Figure 7. Comparison Histogram for LNA Calibration Channel 11, 5 Board Builds

SILICON LABS

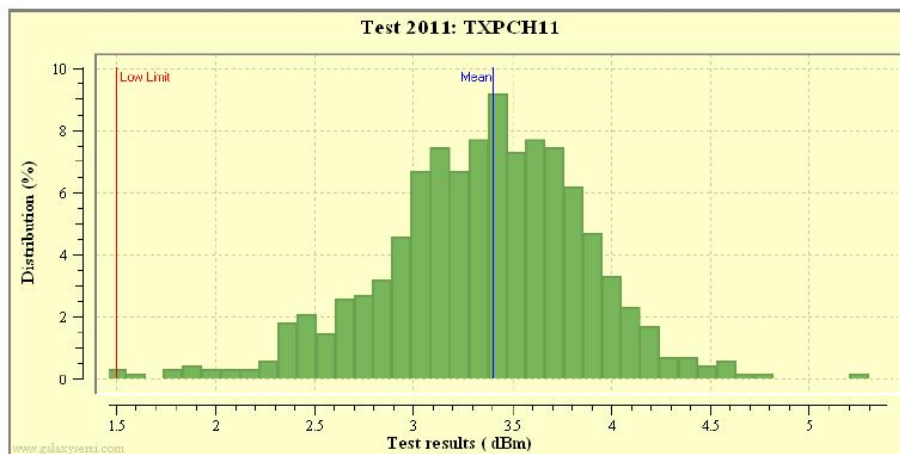| Test	2011
Name	TXPCH11
Test type	Parametric
Low L.	1.5 dBm
High L.	n/a .
Mean / Min / Max	3.40642 dBm / 1.5 dBm / 5.35 dBm from summary
Mean / Min / Max	3.40642 dBm / 1.5 dBm / 5.35 dBm from 801 samples
Sigma / Range	0.500836 dBm / 3.85 dBm
Fails / Outliers	0 / 17

Figure 8. Histogram for Transmit Power Test Channel 11, Sample of 801 Boards

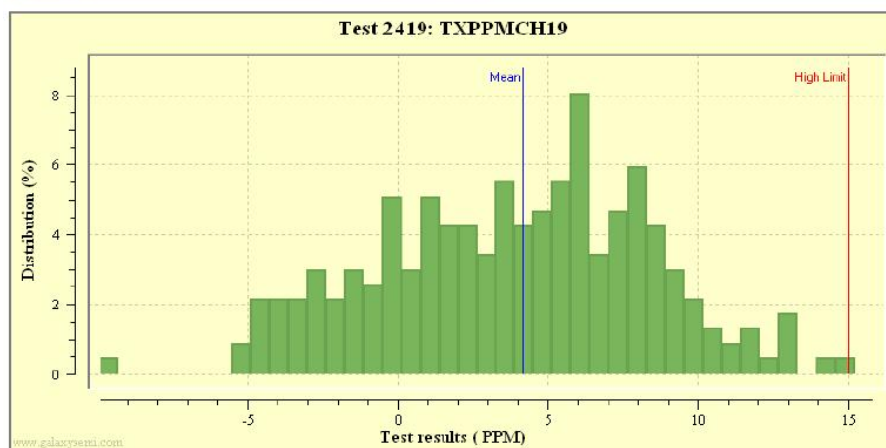| Test	2419
Name	TXPPMCH19
Test type	Parametric
Low L.	−15.0 PPM
High L.	15.0 PPM
Mean / Min / Max	4.1384 PPM / −9.65 PPM / 15.53 PPM from summary
Mean / Min / Max	4.1384 PPM / −9.65 PPM / 15.53 PPM from 238 samples
Sigma / Range	4.49499 PPM / 25.18 PPM

Figure 9. Histogram for Transmit Frequency Offset PPM Channel 19, Sample of 238 Boards

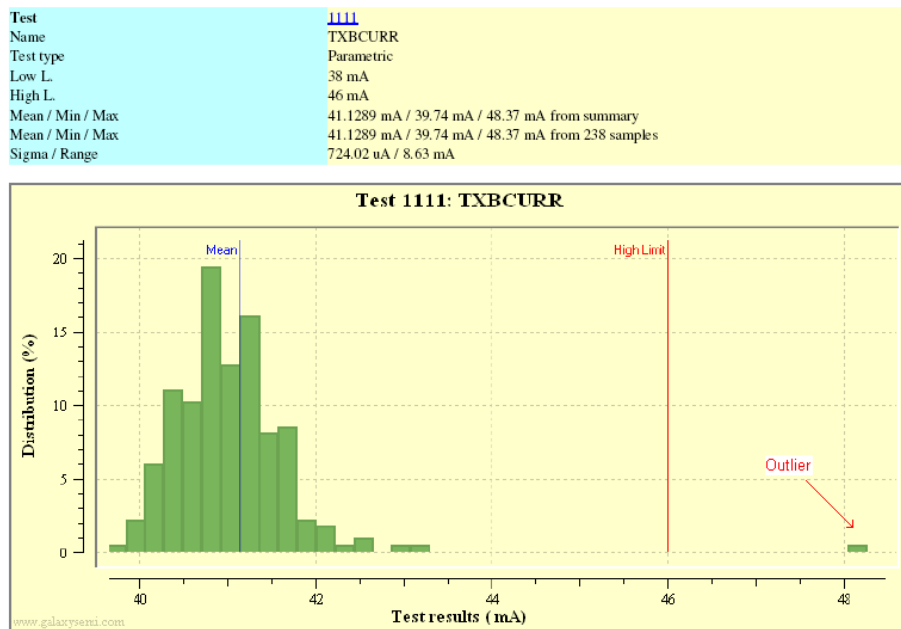| Test	
Name	TXBCURR
Test type	Parametric
Low L.	38 mA
High L.	46 mA
Mean / Min / Max	41.1289 mA / 39.74 mA / 48.37 mA from summary
Mean / Min / Max	41.1289 mA / 39.74 mA / 48.37 mA from 238 samples
Sigma / Range	724.02 uA / 8.63 mA

Figure 10. Histogram for Boost Mode Transmit Current, Sample of 238 Boards

9 Appendix C – Signal Generator Data Files and Configuration

The signal generator arbitrary waveform generator requires I and Q data files defining the waveform to be transmitted. The E4432B uses 14-bit files, while the E4438C uses 16-bit files. These data files can be downloaded from www.silabs.com/zigbee-support.

9.1 Downloading the 802.15.4 2.4 GHz arbitrary waveform files to the signal generator

The signal generator requires specific waveform files for generating the IEEE 802.15.4 type packets. This procedure describes how to download the arbitrary waveform file from the PC to the test equipment.

1. Connect the signal generator through GPIB to the PC.
2. Download and install IntuiLink from the Agilent website (www.agilent.com). Search for IntuiLink for PSG.
3. Select the download link IntuiLink for PSG/ESG Signal Generators at the bottom of the page. This is an Excel toolbar, so make sure Microsoft Excel is also installed on the PC. Follow the instructions on the web page. This link will install the toolbar that lets you download the I/Q data files to the signal generator.
4. Click Download Arb Waveform on the toolbar.
5. Select the I waveform file. Select the Auto find Q file name check box to automatically find the Q file
6. Select type Waveform and File Type ASCII if not already set.
7. Markers should be set as Marker Start 0 and Length 4. User Marker File should not be checked.
8. Select the Auto find marker – file name check box.
9. Uncheck the Auto Name check box and name the file EM250_20BYTE.
10. Set the Scale Factor to 70.
11. If using the E4438C, under Markers set Start to 0 and set Length to 4.
12. Select Download, and the signal generator should download the file appropriately.

SILICON LABS

9.2   Configuring the arbitrary waveform generator for use in test – E4432B Series

After you have downloaded the I/Q data files to waveform memory, you need to configure the signal generator with the settings detailed in the next sections: Configuring the arbitrary waveform generator for use in test – E4432B Series or Configuring the arbitrary waveform generator for use in test – E4438C Series.

Before starting this procedure, make sure the I-Q waveform file has been downloaded to the signal generator (see the section Downloading the 802.15.4 2.4 GHz arbitrary waveform files to the signal generator).

1. On the signal generator keypad, press the Mode button and then select Arb Waveform Generator | Dual Arb.
2. Choose Select Waveform and use the wheel to highlight the desired waveform file (it should be EM250_20BYTE if you followed the "Downloading the 802.15.4 2.4 GHz arbitrary waveform files to the signal generator" procedure above.
3. Choose Select Waveform again and then press Return.
4. Select Waveform Segments and use the wheel to highlight the desired waveform file EM250_20BYTE.
5. Select Store, then Store Segment to NVARB Memory and then press Return.
6. Select Arb Setup, and under Arb Sample Clock type in 16MHz with the keypad and then Enter. Press the Return button.
7. Set the following:

Set:	To:
Arb Reference	Int
Recon Filter	8MHz
Marker Pol	Neg
Mkr2 to RF Blank	On and then press the Return button.

8. Select Trigger | Single.
9. Select Trigger Set-up and set the following:

Set:	To:
Trigger Source	Ext
Ext Pol	Neg
Ext Delay	Off

10. Verify that the displayed menu in the middle of the screen shows the following before continuing:
    - Sample Clock 16 MHZ
    - Reconstruction 8 MHz
    - Ref Freq 10 MHz (Int)
    - Trig Type Single
    - Trig Source Ext
    - Polarity Neg
    - Retrigger Off
    - Delay Off
11. Turn Arb to On.
12. Press Save to save this current state so it can easily be recalled for future use. Name this state EM250_20BYTE if desired.

AN700

9.3   Configuring the arbitrary waveform generator for use in test – E4438C Series

Before starting this procedure, make sure the I-Q waveform file has been downloaded to the signal generator (see the section Downloading the 802.15.4 2.4 GHz arbitrary waveform files to the signal generator).

1. On the signal generator keypad, press the Mode button and select Arb Waveform Generator | Dual Arb.
2. Choose Select Waveform and use the wheel to highlight the desired waveform file (it should be EM250_20BYTE if you followed the preceding Downloading the 802.15.4 2.4 GHz arbitrary waveform files to the signal generator procedure.
3. Choose Select Waveform again and then press Return.
4. Select Waveform Segments and use the wheel to highlight the desired waveform file EM250_20BYTE.
5. Select Store, then Store Segment to NVWFM Memory and then press Return twice.
6. Select Arb Setup, and under Arb Sample Clock type in 16 MHz with the keypad and then Enter. Press the Return button.
7. Set Arb Reference to Int.
8. Select Waveform Utilities and set Scale Waveform Data to 100%.
9. Select Clipping, and then set the following:

Set:	To:
Clipping Type	(|I+jQ|)
Clip |I+jQ|	100%

10. Select Return twice.
11. Choose Marker Utilities | Marker Polarity and then set the following:

Set:	To:
Marker 1 Polarity	Pos
Marker 2 Polarity	Neg
Marker 3 Polarity	Pos
Marker 4 Polarity	Pos

12. Select Return and then Marker Routing and then set the following:

Set:	To:
Pulse/RF Blank	Marker 2
ALC Hold	None
Alternate Amplitude	None

13. Press Return twice, select More, and then set the following options:

Set:	To:
Waveform Runtime Scaling	70%
I/Q Mod Filter (40.0MHz)	Auto
I/Q Output Filter (40.0MHz)	Auto
Modulator Atten (10.0dB)	Auto
High Crest Mode	Off

SILICON LABS

14. To check that all of the arbitrary waveform settings are correct, choose Arb Setup | Header Utilities and verify the following settings are displayed:

This setting:	Is set to:
Sample Rate	16MHz
Runtime Sampling	70%
Marker 1 Polarity	Pos
Marker 2 Polarity	Neg
Marker 3 Polarity	Pos
Marker 4 Polarity	Pos
ALC Hold Routing	None
Alt Ampl. Routing	None
RF Blank Routing	Marker 2
High Crest Mode	Off
Mod Attenuation	Auto
I/Q Mod Filter	Auto
I/Q Output Filter	Auto

15. Select Return twice.
16. Select Trigger | Single.
17. Select Trigger Set-up and set the following:

Set:	To:
Trigger Source	Ext
Ext Pol	Neg
Ext Delay	Off

18. Verify that the displayed menu in the middle of the screen shows the following before continuing:
    - Sample Clock 16 MHZ
    - IQ Mod Filter 40 MHz
    - Ref Freq 10 MHz (Int)
    - Trig Type Single
    - Trig Source Ext
    - Polarity Neg
    - Retrigger Off
    - Delay Off
19. Turn Arb to On.
20. Press Save to save this current state so it can easily be recalled for future use. Name this state EM250_20BYTE if desired.

SILICON LABS

AN700

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Please visit the Silicon Labs Technical Support web page for ZigBee products:
www.silabs.com/zigbee-support and register to submit a technical support request

Patent Notice

Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog-intensive mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories, Silicon Labs and Ember are registered trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.

SILICON LABS