
RF TO USB REFERENCE DESIGN 2 USER'S GUIDE

1. Kit Contents

The RF to USB Reference Design contains the following items:

- RF to USB Node Boards (2)
- AAA batteries (2)
- RF to USB Dongle (1)
- Tool Stick Programming Adapter (1)
- Retractable USB cord (1)

2. Introduction

This kit is intended to serve both as a simple, low-power analog data collection example and as a development platform for wireless sensing projects.

The demonstration uses one or more node boards to communicate with a USB dongle, which acts as a network coordinator and data collector. The system measures temperature and voltage on each dongle board and displays them on a PC window.

The development example uses two node boards to communicate to each other. A simple code example is provided as a jump-start for custom projects. Peripheral or low power functionality can be added to this example by referencing an available library of code examples.

This system is intended to be a simple starting point. For more advanced radio or peripheral evaluation, a full Si1000DK development kit is recommended.

This kit, RF to USB Reference Design 2, replaces the original RF to USB Reference Design.

3. Quick Start

This section guides you through the setup of a data collection example using two node boards to wirelessly transmit analog measurements to a PC-based user interface.

1. Install the RF to USB Reference Design software and firmware examples from your installer package, either downloaded from www.silabs.com/RFtoUSB or from a CD.
2. Plug the dongle into an available USB port as shown in Figure 1.

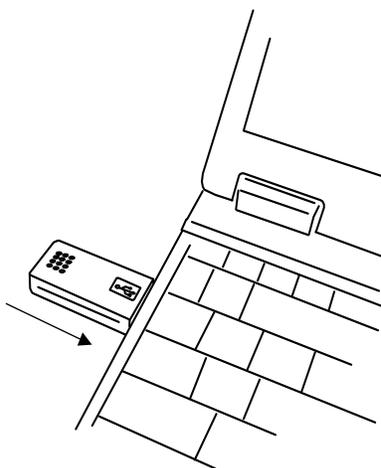


Figure 1. RF USB Dongle Connect

RF-to-USB-RD2

3. Open the PC demonstration. Default path is
C:\SiLabs\MCU\RF_to_USB_RD\PC Software, RF to USB Network Demo.exe
4. Click Connect to establish communication with the dongle as shown in Figure 2.

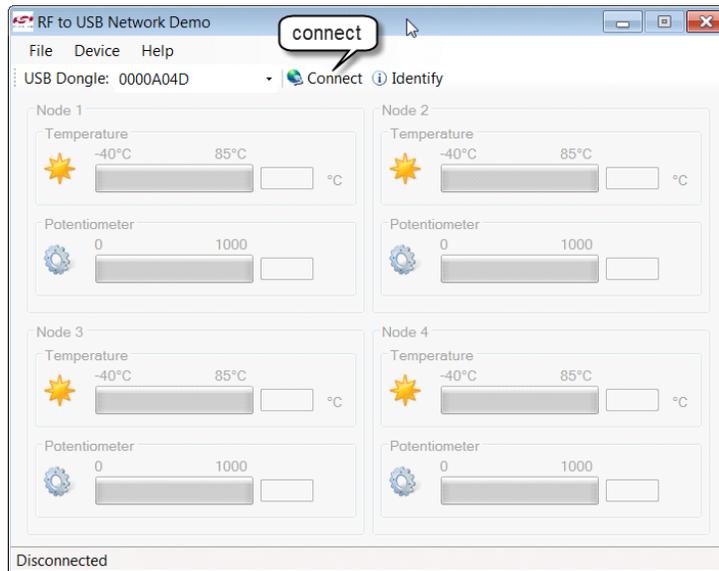


Figure 2. Click Connect

Note: If an “old firmware” warning is received, click “yes” to update as shown in Figure 3.

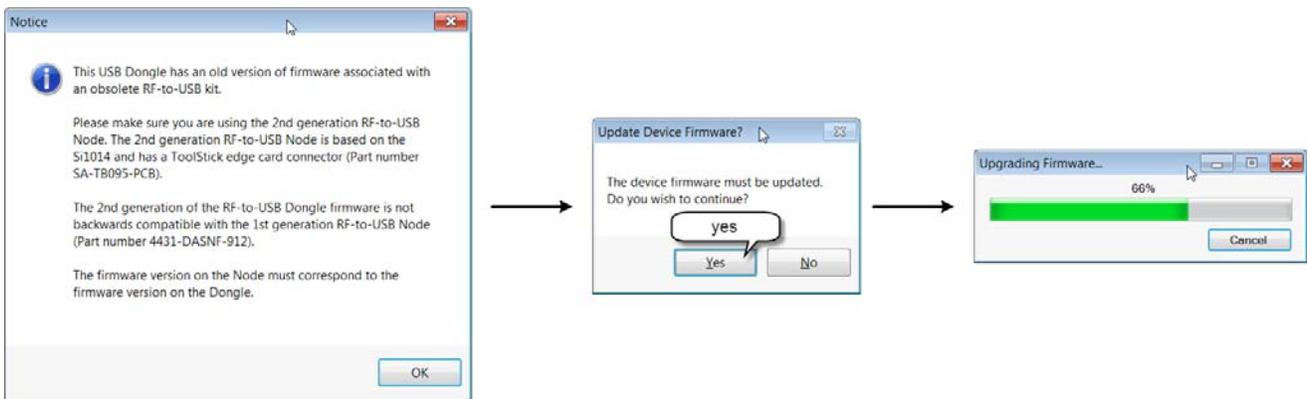


Figure 3. Old Firmware Warning

5. Insert the battery into one of the node boards as shown in Figure 4.

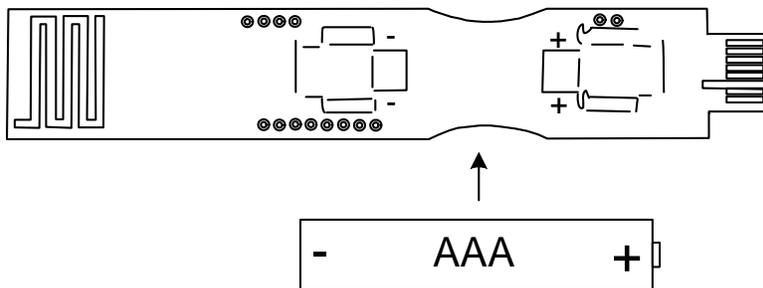


Figure 4. RF USB Node Battery In

6. Slide the power switch to BATT as shown in Figure 5.

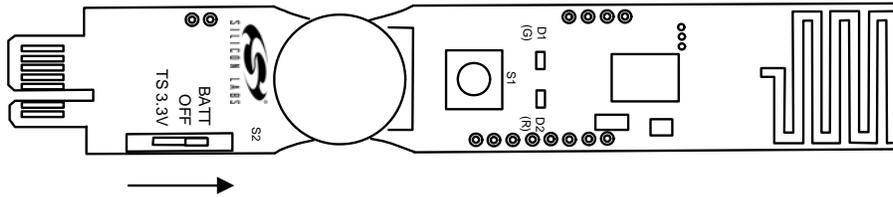


Figure 5. RF USB Node Power On

7. Associate the board with the dongle and PC application, and press the button as shown in Figure 6.
a. Press the button.

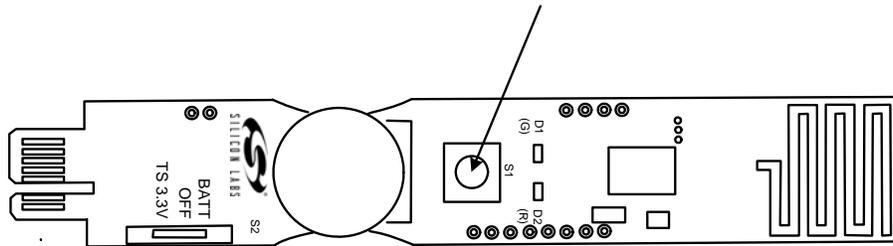


Figure 6. RF USB Node Button Press

b. Note that the first board is assigned as Node 1 (Figure 7).

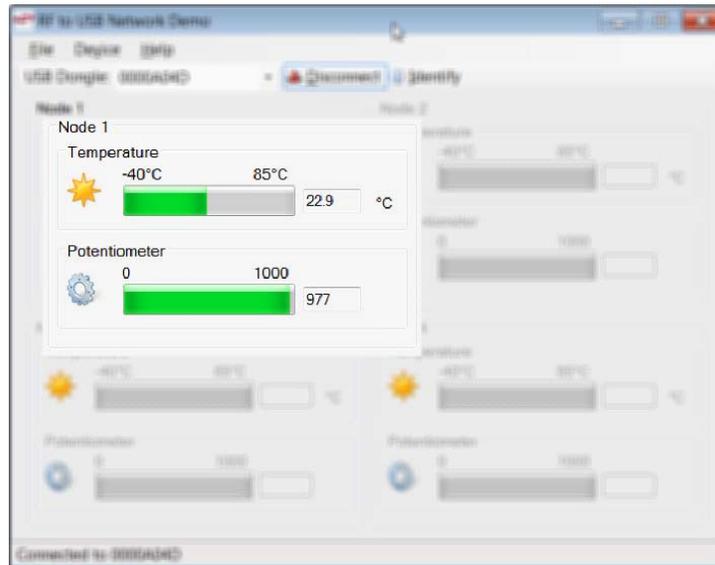


Figure 7. First Node Assigned as Node 1

RF-to-USB-RD2

3. Vary the ADC input by turning the potentiometer as shown in Figure 8.

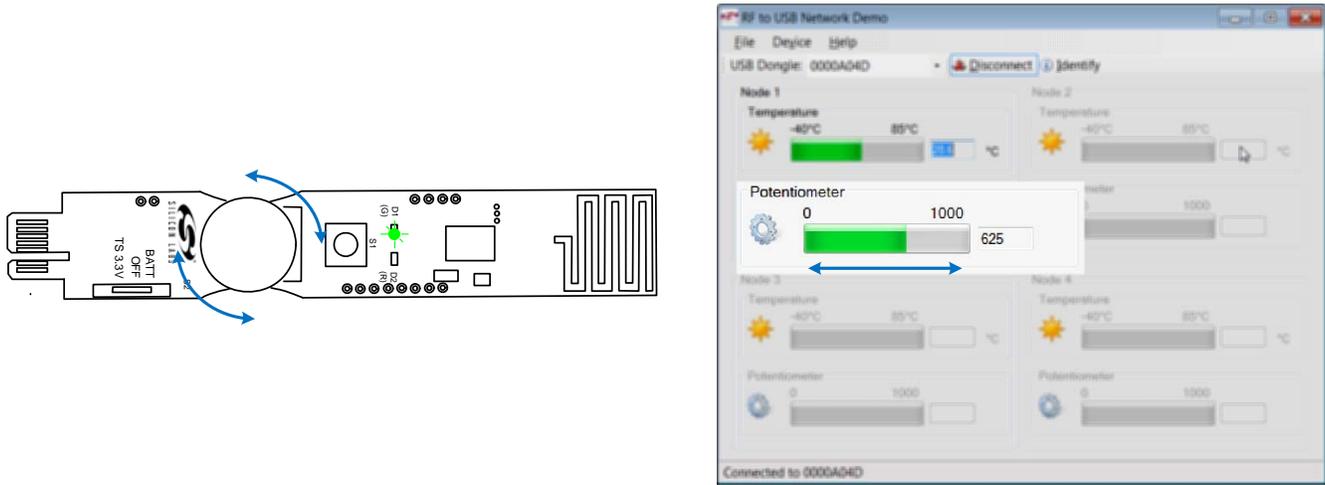


Figure 8. RF USB Node Potentiometer

4. Touch the Si1014 package to warm the internal temperature sensor as shown in Figure 9.

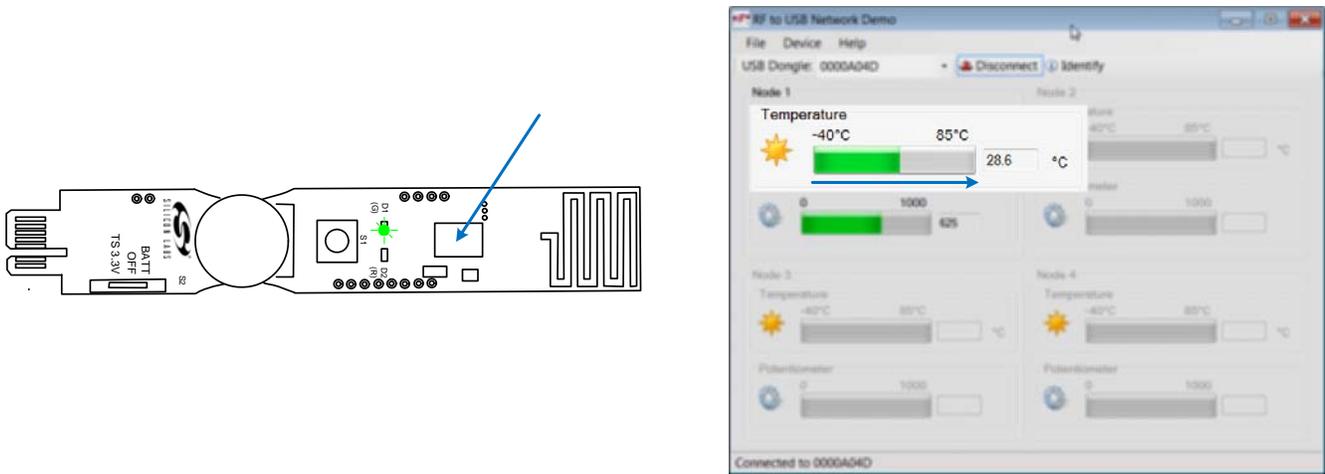


Figure 9. RF USB Node Temperature Changes

Note: The node board transmits periodically for a fixed time, then goes to sleep. Press the button again to restart.

4. RF to USB Data Collection Demo

The RF to USB kit features a wireless data sensor demonstration. This demo uses one or more node boards to transmit analog data (temperature and a potentiometer value) to a USB dongle at a fixed frequency of 919.84 MHz. Refer to "3. Quick Start" on page 1 for installation and basic use. Detailed options follow.

4.1. PC Application Features



Figure 10. RF to USB Network Demo

1. USB Dongle ID

Each dongle has a unique identifier. In the case of two or more dongles connected to the PC, this window lets you select which one you would like to connect to.

2. Connect / Disconnect

This button establishes or removes a connection between the PC application and a specific dongle.

3. Identify

In the case of two or more dongles connected to the PC, this button will flash the yellow ID on the dongle to identify. This is also useful to test communication between the PC and the dongle.

RF-to-USB-RD2

4. Node (1–4)

Up to four nodes may be connected to the PC application. The associated node window becomes enabled (or ungreyed) as each node associates.

This demonstration uses a very simple association scheme. Each node board powers up unassociated. On the first button press, the node requests a node ID from the dongle. The dongle assigns IDs in a round-robin fashion, i.e. 1-2-3-4-1-2-...

Note that there is no disassociation, and given the round-robin association scheme, it is possible to have two boards associated with the same node ID, particularly if power is cycled on the node boards. Reset is accomplished by disconnecting and reconnecting the PC application, cycling power on the node boards, then reassociating each by pressing its button.

5. Temperature Window

This displays the node's temperature, as measured by the Si1014's on-chip temp sensor. The node will transmit temperature data periodically, then enter a low-power sleep mode after several seconds. Press the node's button to wake and send again.

6. Potentiometer Window

This displays a normalized ADC reading from the node. The voltage measured by the ADC is set by a potentiometer on the board.

Note: The second generation RF to USB Reference Design is not backward-compatible with the earlier version because the over-the-air protocol has changed. First-generation node boards are identified by part number "4431-DASNF912 HB", whereas second-generation node boards are identified by part number "SA-TS095PCB-001". The dongle is compatible with either kit since its software is downloaded automatically by the PC application.

5. Developing an Application with the RF to USB Kit

The RF to USB Kit is intended to be a demonstration and starting point for low-power/single-cell wireless sensor applications. This section details the development features of the kit.

5.1. Firmware Development

This kit includes multiple nodes, a programming adapter, and firmware examples to help start your development.

The development environment is centered around the Silicon Laboratories IDE, using a free evaluation version of the Keil, SDCC, or Raisonance compilers. The IDE connects to the Node board through the supplied ToolStick programming adapter.

Visit the Silicon Laboratories web site (keyword search “IDE download”) to obtain a free copy of the Silicon Laboratories IDE. A trial version of a compiler must also be downloaded and installed.

This software example is a simple two-way link. Pressing the button on one node increments a counter and sends that value to the other node, which displays it on two LEDs.

The software example is based on the PreProcessor PHY (ppPhy), a free software library from Silicon Labs, which manages much of the radio configuration and interface. More information on the ppPhy is available in “AN474: Si1000 Code Examples”.

Note: The firmware source code for the node boards is included as part of the kit installation. This firmware serves as a good example for Si1014 applications, including analog measurements, onboard DC-DC converter control, and low-power sleep modes. However, the firmware for the dongle board is not included. This firmware has a certain amount of overhead needed for demo functionality and does not serve as a useful programming example. We recommend reviewing C8051F342 USB programming examples and the Si10xx PreProcessorPhy or Si10xx RunTimePhy as a starting point for your application.

RF-to-USB-RD2

5.2. Building and Downloading the Project

1. Open the Silabs IDE and load the project:
C:\Silabs\MCU\RF_to_USB_RD\Firmware\PreProcessorPHY_TxRx\ppPhyTxRx.wsp
2. Set the power switch on one of the two node boards to “TS 3.3V” (ToolStick 3.3 V)

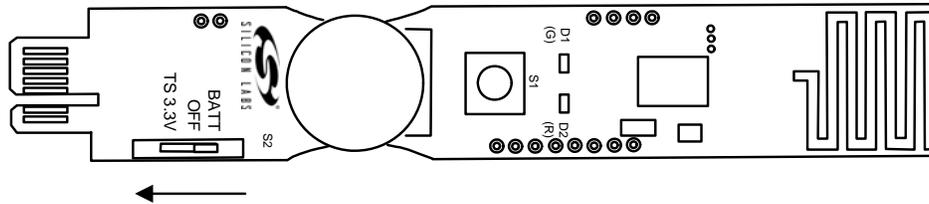


Figure 11. Power Switch to TS 3.3 V

3. Plug the node board in to the ToolStick Programming Adapter, and connect the adapter to your PC's USB port as shown in Figure 12.

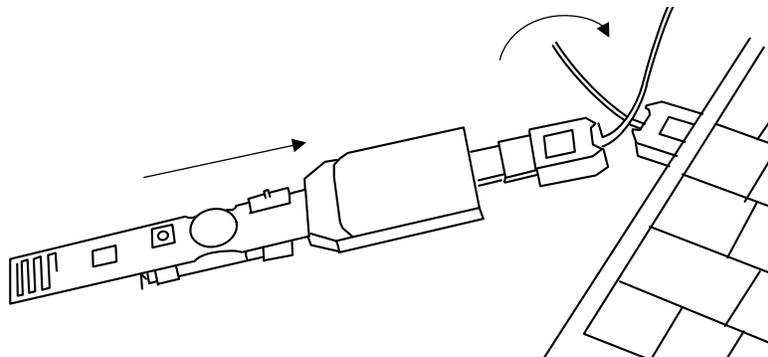


Figure 12. Connect Adapter to PC USB Port

4. Connect the IDE to the programming adapter and Node Board as shown in Figure 13.

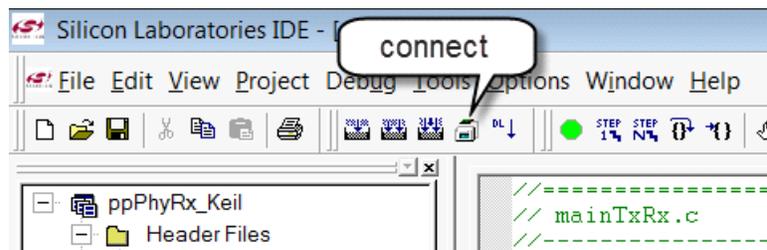


Figure 13. Connect IDE to Programming Adapter and Node Board

5. Build and download the project as shown in Figure 14.

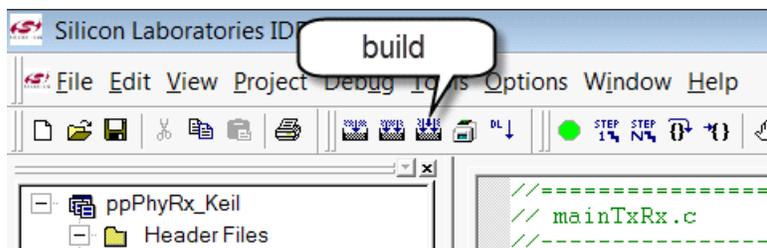


Figure 14. Build and Download Project

6. Verify successful build and download as shown in Figure 15.

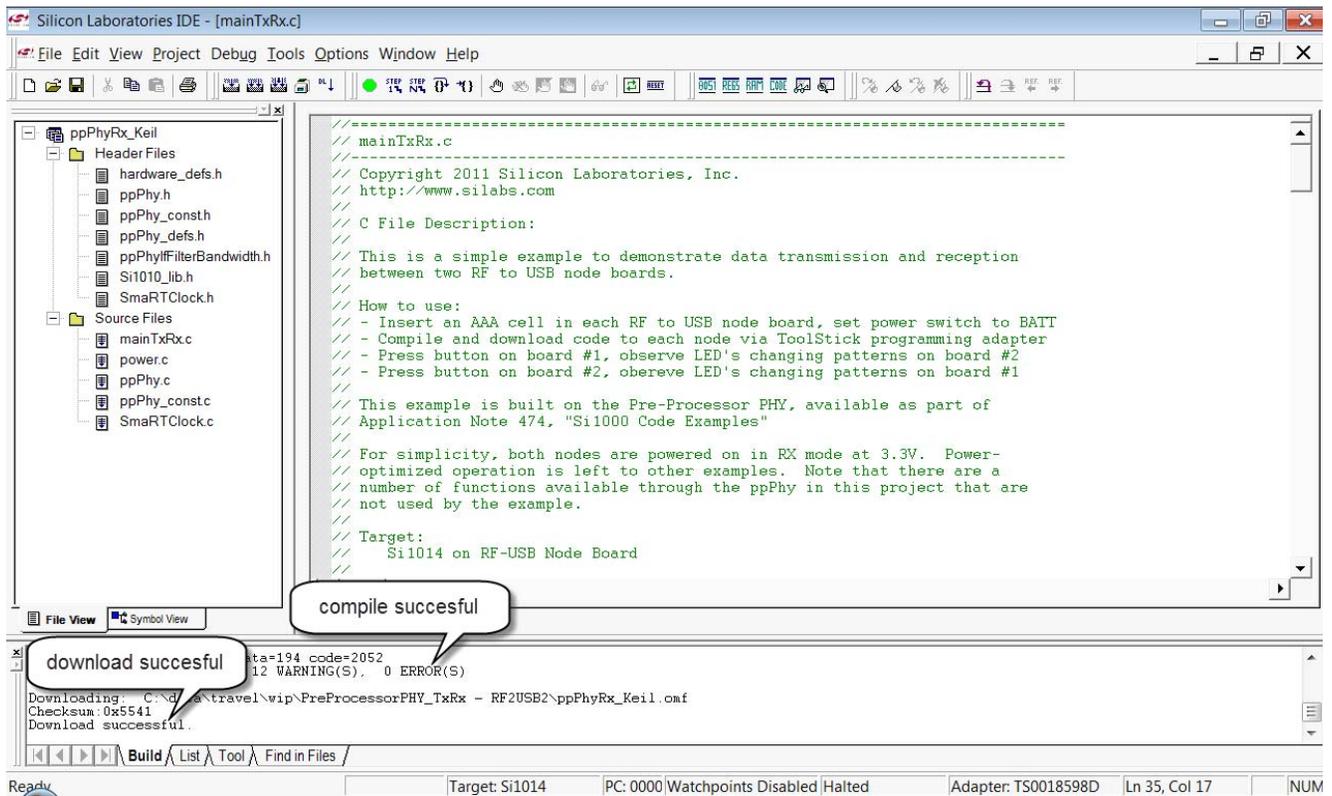


Figure 15. Verify Successful Build and Download

7. Disconnect the IDE as shown in Figure 16.

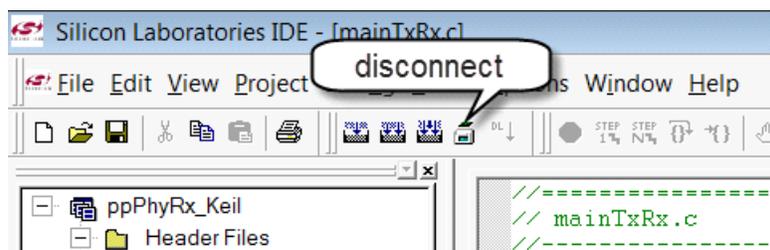


Figure 16. Disconnect IDE

RF-to-USB-RD2

8. Remove the Node board from the ToolStick Debug Adapter as shown in Figure 17.

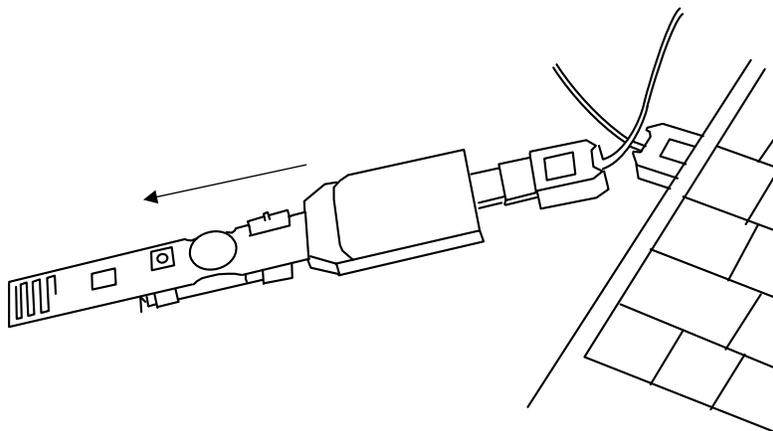


Figure 17. Remove Node Board from Toolstick Adapter

9. Repeat for the second Node board.

5.3. Running the Project

1. Insert a battery in each node board as shown in Figure 18.

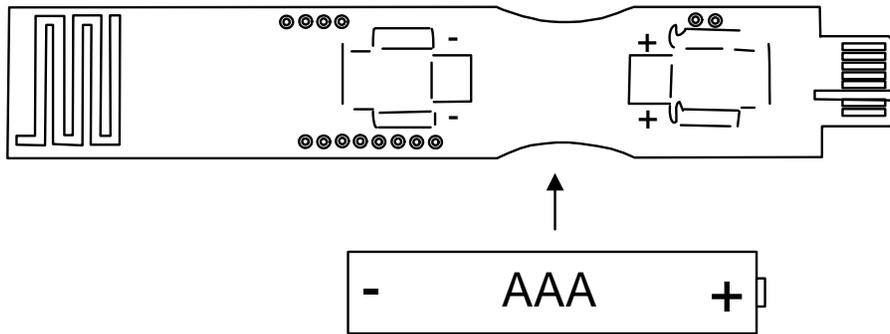


Figure 18. Insert Battery in Node Board

2. Slide the power switch to BATT.

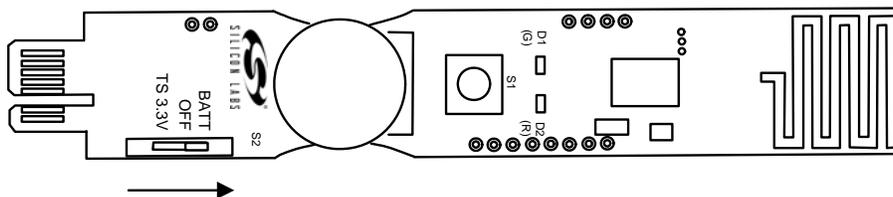


Figure 19. Slide Power Switch to BATT

3. Press the button and observe the LEDs change on the other node board with each press.

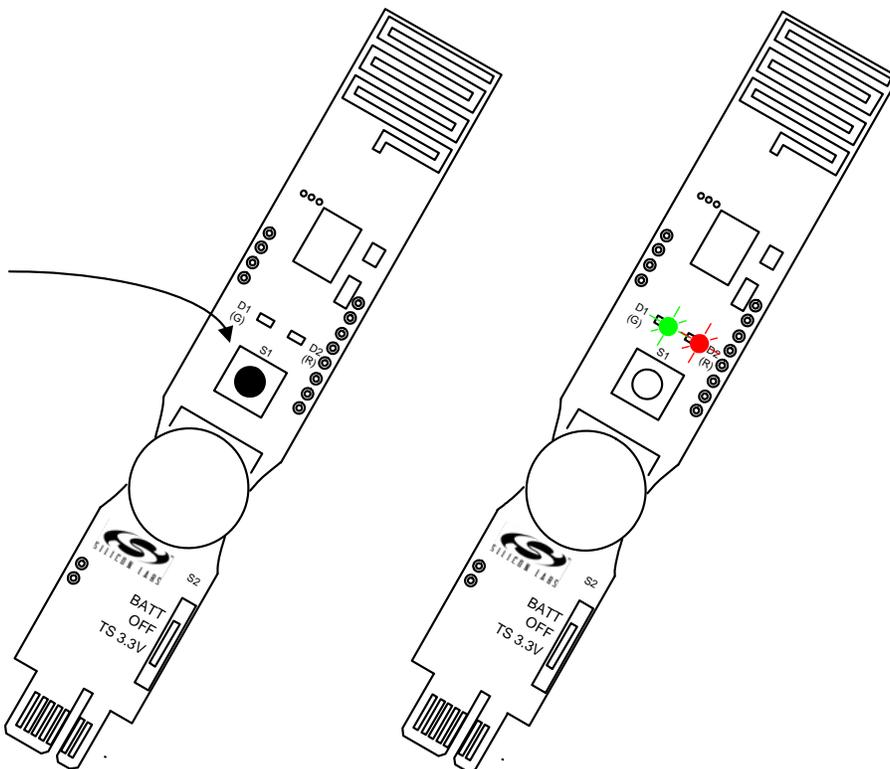


Figure 20. LEDs on Other Node Board Change with Press

RF-to-USB-RD2

5.4. Understanding the Project

Examine the user code. One file, mainTxRx.c, contains the user code. Other files contain the code modules from the ppPhy and do not need to be modified.

The first items called in main.c set the Si1014's DC-DC converter. In this example, the DC-DC converter is used to step a single-cell battery voltage up to 3.3 V.

```
void main (void)
{
    U8 status;
    U8 RxPacketLength;
    U8 transmitWord=0;
    S16 tmp;

    PCA0MD   &= ~0x40;           // disable watchdog
    // Configure DC/DC converter
    DC0CN = 0x47;                // 20ns min, big SW, 3.3V
    DC0CF = 0x04;                // Local osc, High-Peak Current
```

Next, initialization functions for the system clock, ports, and radio serial port are called:

```
    SYSCLK_Init();
    PortInit();
    SPI1_Init();
```

Then the radio is initialized by the ppPhy:

```
    status = ppPhyInit();
    status = ppPhyInitRadio();
```

These functions are included in ppPhy.c. Refer to “AN474: Si1000 Code Examples” and “AN440: Si4430/31/32 Register Descriptions” for additional detail.

Next, enable the interrupts and turn on the radio to receive any incoming packets:

```
    EA = 1;
    status = ppPhyRxOn();
```

Once initialized, the firmware loops, checking for either a keypress (requesting a packet to be transmitted) or a receive flag, indicating a packet in the RX buffer.

For this example, the transmitter increments a variable, transmitWord, loads this in to the TxBuffer, and sends it. The green LED winks to show the transmission.

```
    while(1)
    {
        // transmit [1..3] value, incrementing each keypress
        if(SW==0)
        {
            while(SW==0);
            TxBuffer[0] = transmitWord++;
            if(transmitWord >3)
                transmitWord = 1;
            LED_GRN = ~LED_GRN;
            status = ppPhyTx(64,TxBuffer);
            LED_GRN = ~LED_GRN;
        } //if TX
```

Within the same loop, the firmware checks the RxPacketReceived flag (set by an interrupt service routine). If true, indicating there is data waiting, the firmware lights the red and green LED according to the value received.

```
// receive from other node, display value with LED's
if(RxPacketReceived)
{
    ppPhyGetRxPacket(&RxPacketLength, RxBuffer);

    switch (RxBuffer[0])
    {
    case 1:
        LED_GRN = ILLUMINATE;
        LED_RED = EXTINGUISH;
        break;
    case 2:
        LED_GRN = EXTINGUISH;
        LED_RED = ILLUMINATE;
        break;
    case 3:
        LED_GRN = ILLUMINATE;
        LED_RED = ILLUMINATE;
        break;
    default:
        LED_GRN = EXTINGUISH;
        LED_RED = EXTINGUISH;
        break;
    }
} //if RX
} //while (1)
```

The firmware sits in this loop indefinitely and waits for a transmit request keypress or an incoming packet.

This is a very simple code example. You may extend this example to suit your application by:

- Adding power management: See code examples in C:\SiLabs\MCU\Examples\Si101x\SleepMode to place the MCU core in low-power, waking up on keypress, or a real-time clock.
- Making a measurement with the ADC: See code examples under C:\SiLabs\MCU\Examples\Si101x\ADC.
- Adding communications via UART or SPI: See code examples under C:\SiLabs\MCU\Examples\Si101x\UART and C:\SiLabs\MCU\Examples\Si101x\SPI

Note that these code examples are installed as part of the Silicon Laboratories IDE installation.

RF-to-USB-RD2

5.5. Hardware Development

The RF to USB node design is a realistic reference design for a low-power wireless sensor node for a number of reasons:

- Low-power operation possible using a 32 kHz timer or keypress to wake
- Low-voltage operation using a single AAA cell and on-chip dc-dc boost converter
- Low-cost printed antenna
- ADC measurement capability via the Si1014's on chip ADC

PCB design files for the node board (in PADS, Gerber, and PDF formats) are available from the Silicon Laboratories website.

The RF-to-USB dongle is intended more as a fixed demonstration tool. Silicon Laboratories will gladly provide PCB source files (in Eagle) and Gerbers, but the dongle is not intended for development work since it does not have a convenient debugging interface.

Note also that there is a solar powered energy harvesting version of the node board available. Refer to the Energy Harvesting Reference Design kit for more details.

5.6. RF to USB Node Board Features

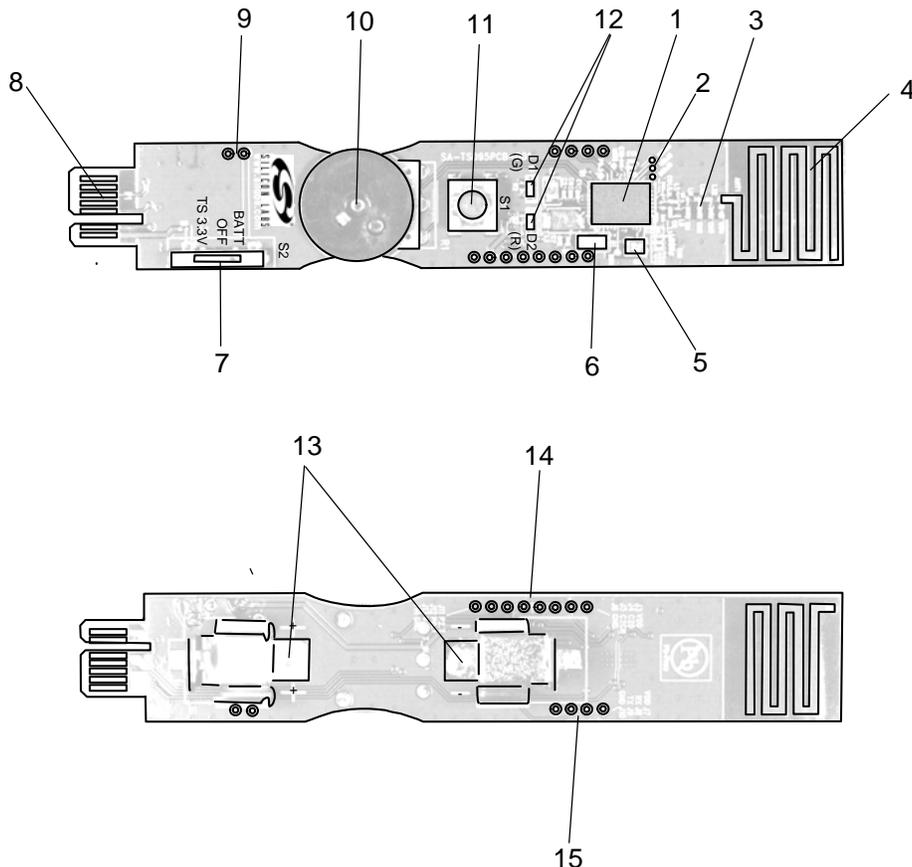


Figure 21. Node Board Features

1. **Si1014 Wireless MCU**
Includes MCU core, +13 dBm radio transceiver, and dc-dc converter.
2. **Radio GPIO Test Points**
3. **RF Matching Network**
Direct tie (refer to “AN436: Si4030/4031/4430/4431 PA Matching”) matched for 915 MHz.
4. **Printed Antenna**
5. **30 MHz Crystal**
Reference for RF PLL
6. **32.768 kHz Crystal**
Used for MCU sleep timer
7. **DC-DC Input Source Switch**
Selects between three options:
 - **TS 3.3**
3.3 V from toolstick adapter. Note that the DCDC output will be 3.3 V, regardless of setting, when the switch is in this position.
 - **OFF**
 - **BATT**
DC-DC converter connected to battery terminals. The switch should be left in this position for any low-power application since this represents the case of a battery connected directly to the MCU dc-dc converter input.
8. **ToolStick Adapter Interface**
Interface includes:
 - 3.3 V regulated power, derived from USB 5 V supply
 - C2 programming/debugging interface
 - UART interface via ToolStick Terminal
9. **Test / Prototype Points**
DC-DC converter input voltage, ground.
10. **Potentiometer**
Wiper connected to ADC input via P0.1. Potentiometer enabled by pulling port P0.0 low. P0.0 normally high impedance for power savings.
11. **Pushbutton**
Active low, P0.6
12. **LEDs**
Active low, P1.5 (green) and P1.6 (red)
13. **Battery Connectors**
Single AAA cell
14. **Test/Prototype Points**
VDD (DC-DC output), C2 connections, DCDC input, misc port.
15. **Test/Prototype Points**
UART, VDD, GND

RF-to-USB-RD2

6. PCB Schematics and Layouts

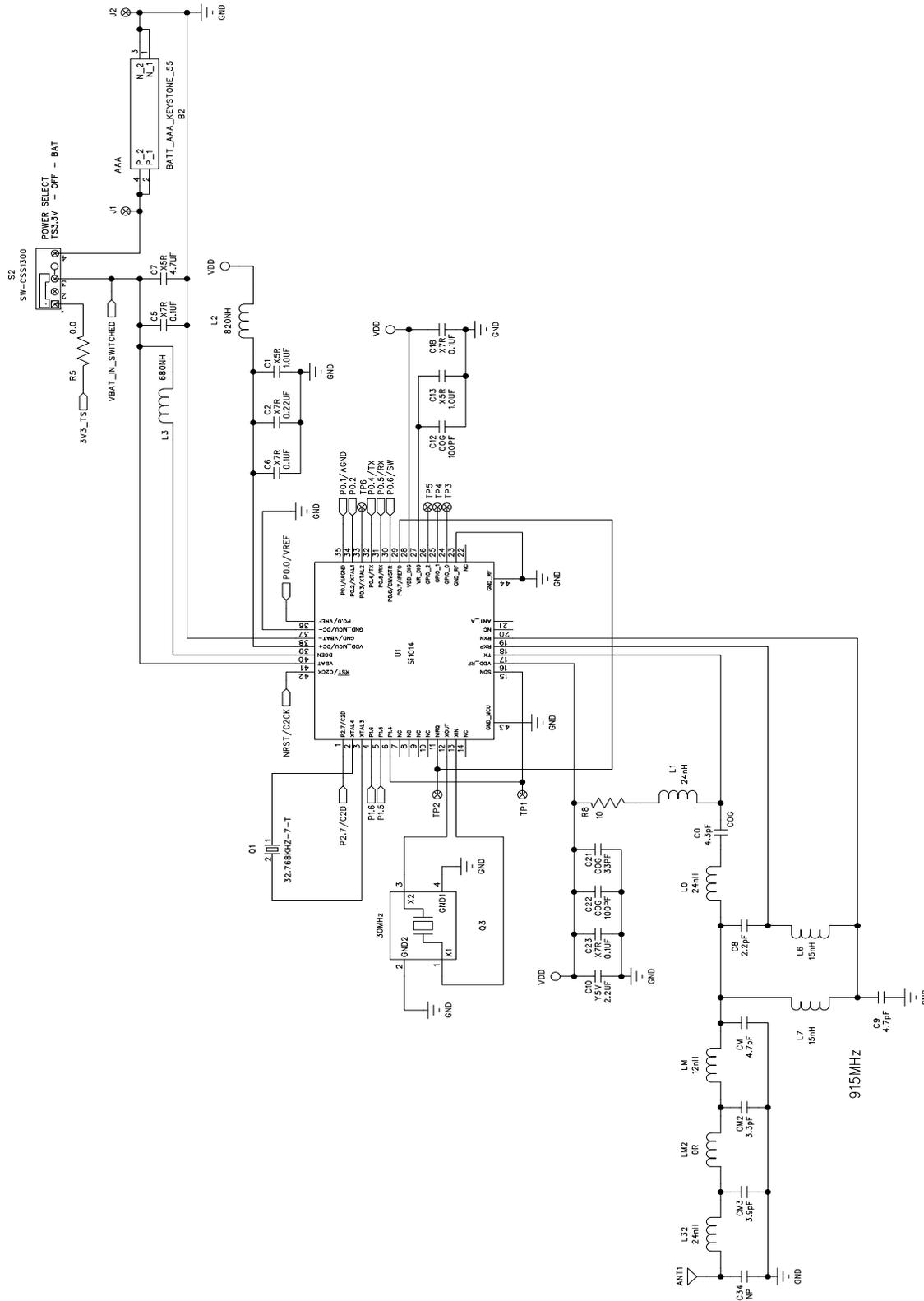


Figure 22. Node Schematic

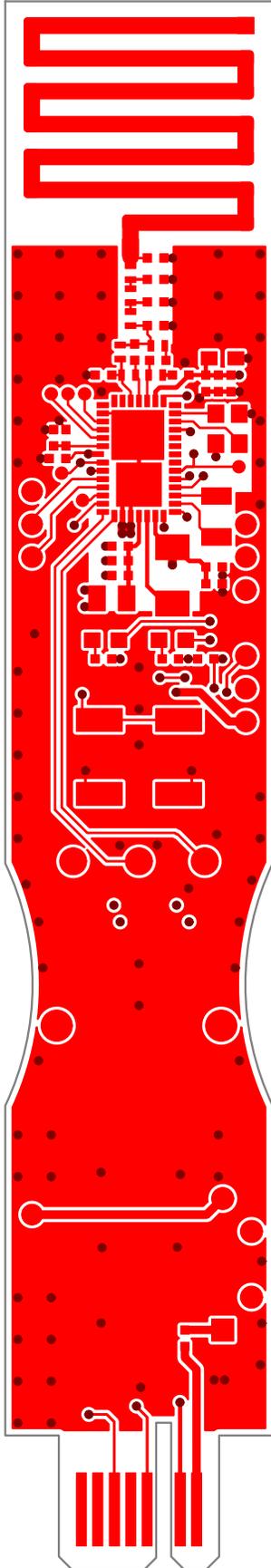


Figure 24. Node PCB Layout (L1 Top)

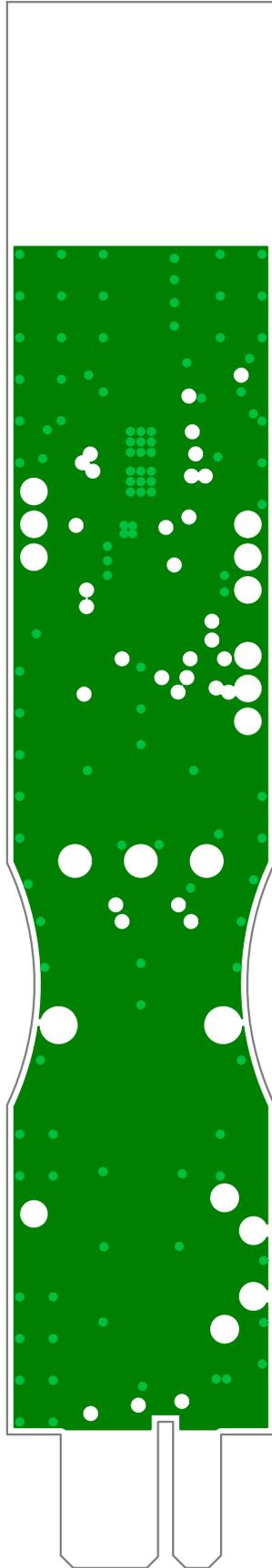


Figure 25. Node PCB Layout (L2 Gnd)

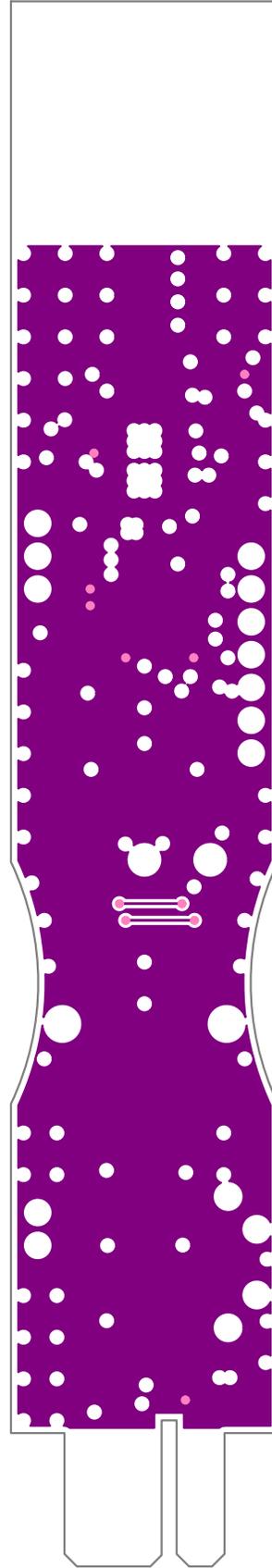


Figure 26. Node PCB Layout (L3 Gnd)

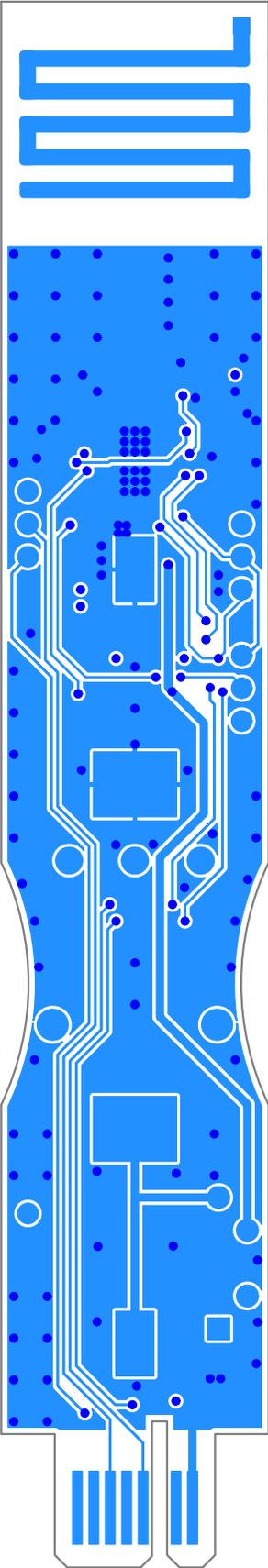


Figure 27. Node PCB Layout (L4 Bottom)

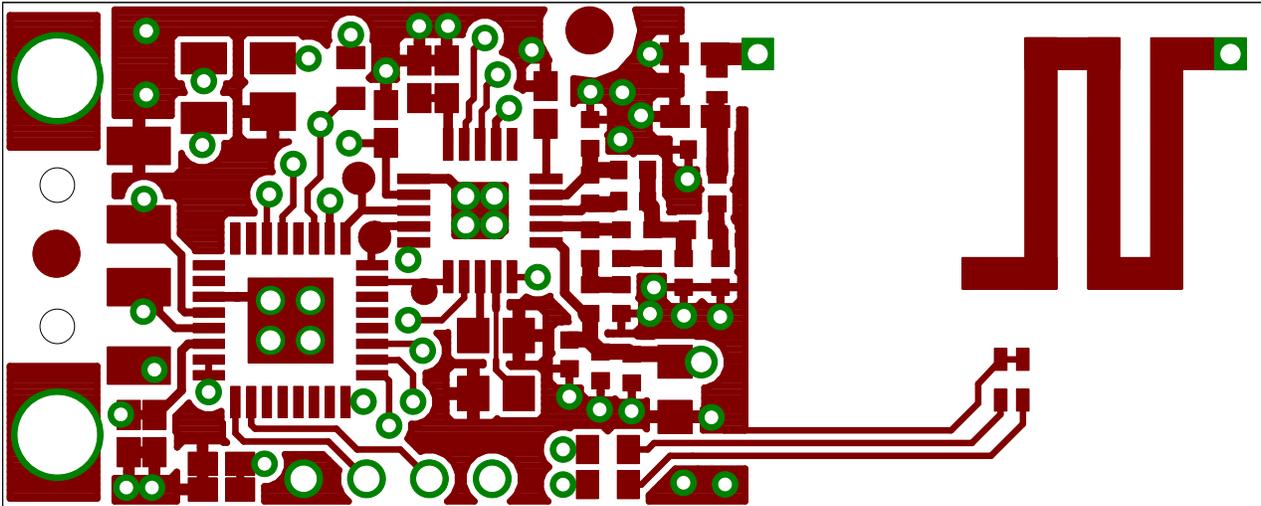


Figure 28. Dongle PCB Layout (L1 Top)

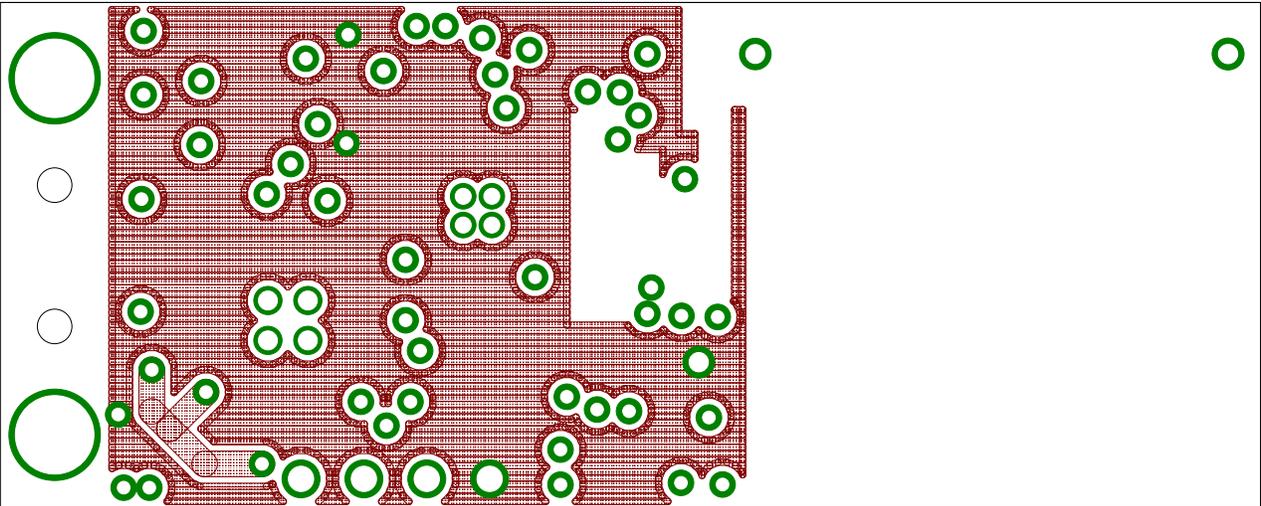


Figure 29. Dongle PCB Layout (L2)

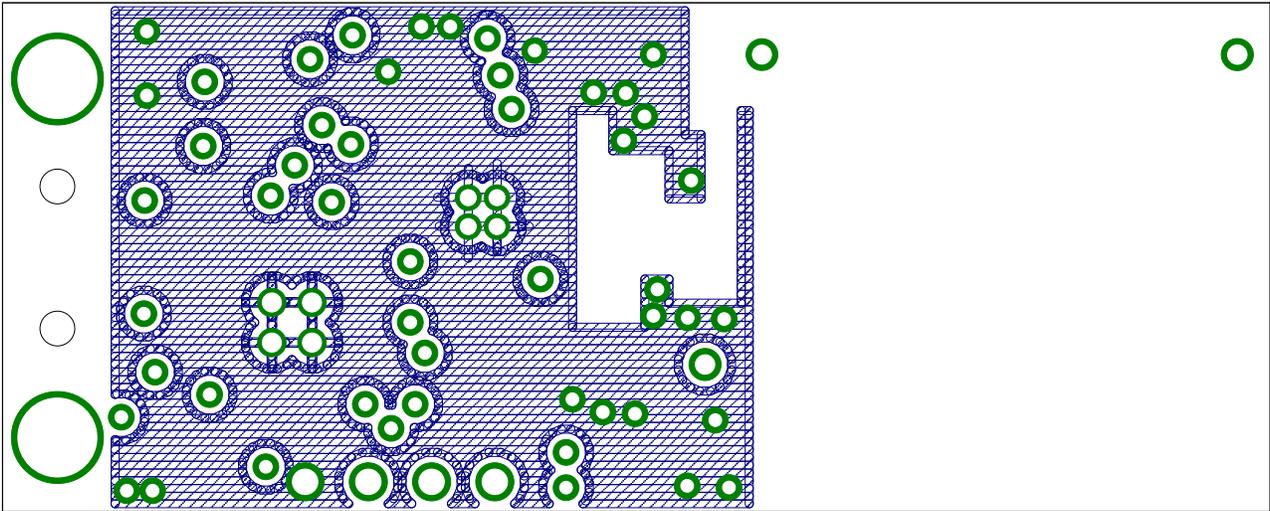


Figure 30. Dongle PCB Layout (L3)

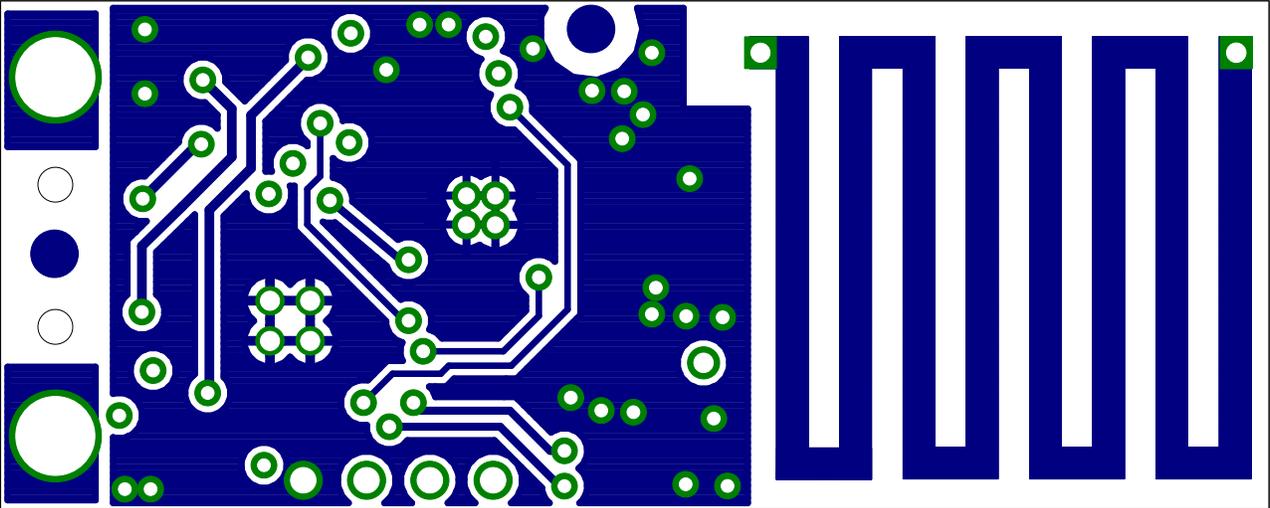


Figure 31. Dongle PCB Layout (L4 Bottom)

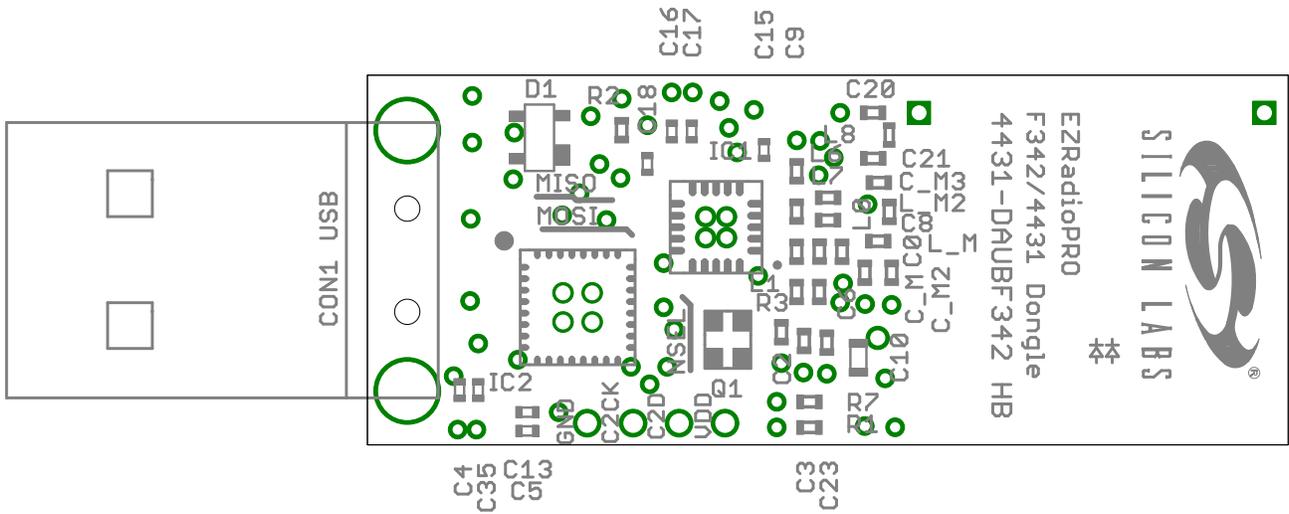


Figure 32. Dongle PCB Layout (Top Assembly)

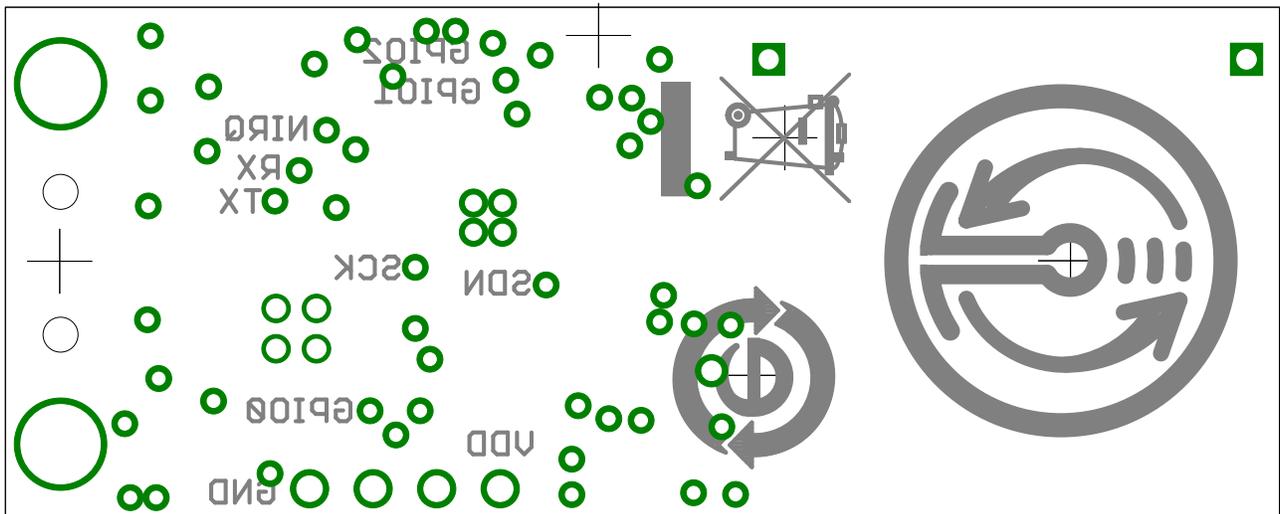


Figure 33. Dongle PCB Layout (L4 Bottom Test Points)

7. References

The RF to USB Reference Design Demo uses a 919.84 MHz RF link. The demo firmware is written on top of EZMAC[®] PRO. The RF link data rate is 128 kbps using GFSK modulation. Both demo applications use EZMAC PRO channel 19.

The output power is +13 dB for both the USB Dongle and Sensor Node. The PCB antennas on both Sensor Node and USB Dongle are designed for lowest possible cost, not maximum range.

For firmware source code or additional information, contact MCU support:

<http://www.silabs.com/support/>

RF-to-USB-RD2

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.2

- Content completely revised.

NOTES:

RF-to-USB-RD2

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:
<https://www.silabs.com/support/pages/contacttechnicalsupport.aspx>
and register to submit a technical support request.

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 and Silicon Labs are trademarks of Silicon Laboratories Inc.

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