ANT BLAZE

Starter Guide

Copyright Information and Usage Notice

Revision History

Revision	Effective Date	Description
1.0	June 2017	Initial Starter Guide creation

Table of Contents

List of Figures

List of Tables

1 Overview

ANT BLAZE offers a turn-key mesh solution which can be used by an application running on the D52 module. This starter guide provides instructions for setting up a simple mesh network for evaluating and experimenting with ANT BLAZE.

ANT BLAZE consists of 2 libraries; a node library and a gateway library. An ANT BLAZE network is comprised of a single Premium D52 module configured as a gateway and several Premium D52 modules configured as mesh nodes.

Example node and gateway applications which utilize the ANT BLAZE libraries have been supplied for evaluating ANT BLAZE. The example node application (demo_node) uses the ANT BLAZE node library and automatically replies over the mesh to messages sent to it. The example network processor gateway application (np_gateway) uses the ANT BLAZE gateway library and interfaces over UART with the included PC application, "Mesh Tester". Mesh Tester takes advantage of a richer UI to enable experimenting with the gateway library by allowing the user to send manual messages or to configure tests that automatically poll nodes for data.

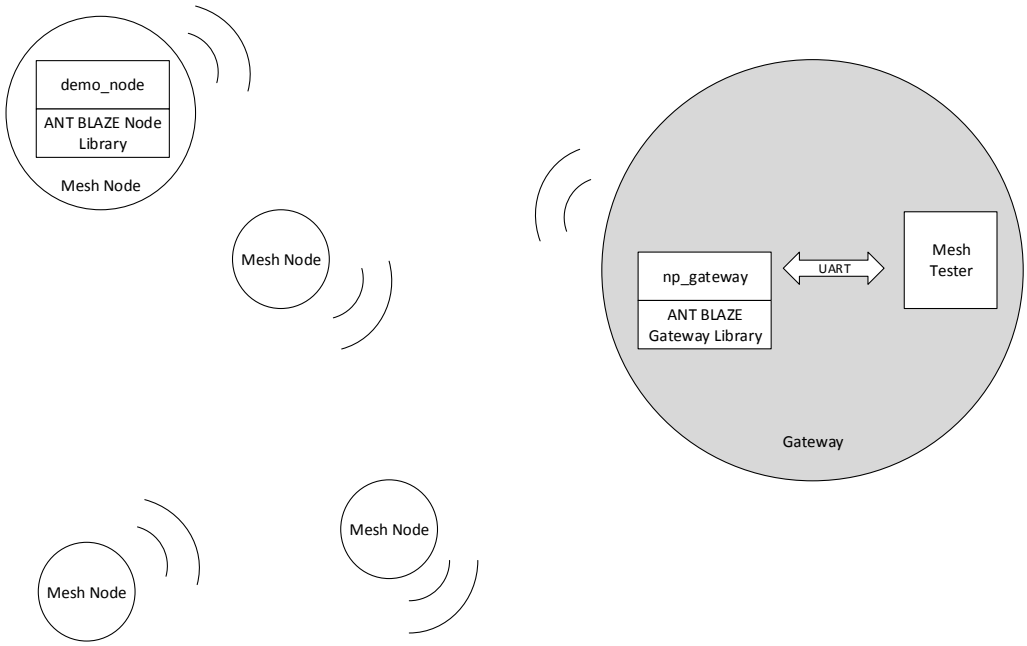

Figure 1. Typical ANT BLAZE Topology

2 Required Development Resources

2.1 Hardware

The following development kits are required to run the example code included in the ANT BLAZE SDK:

- D52 Starter Kit (D52DK2): Includes a Segger J-Link Lite Programmer and an ANT USB Interface Board (ANTUIF1) and two Premium D52Q modules (D52QSKM6IA-A). The ANT USB Interface Board is used together with a Premium D52Q module to test the gateway functionality. For more details on usage of this kit, refer to ANT SoC Starter Kit User Manual.

- D52 Extender Kit (D52EXT1): Includes 4 Premium D52Q modules (D52QSKM6IA-A) and 4 ANT AA Battery Boards (ANTBAT3). These components are used for the mesh nodes. Note the extender kit does not include a J-Link programmer or an ANT USB interface board.

More information about the development kits is available here:

https://www.dynastream.com/d52starterkit

2.2 ANT BLAZE Libraries

The contents of the ANT BLAZE Libraries package are as follows:

- ANT BLAZE static libraries: Pre-built libraries that are used by custom applications to enable ANT BLAZE mesh node or gateway functionality. The bin folder includes versions of the ANT BLAZE Node Library and ANT BLAZE Gateway Library compiled on GCC, IAR and Keil MDKv5 in the armgcc, iar and keil subdirectories.

- API Headers: These files need to be included by your application to access the ANT BLAZE static libraries, and provide documentation on the available API calls.

    - inc\ant_blaze_node_interface.h: Function definitions and API documentation for the ANT BLAZE Node Library.

    - inc\ant_blaze_gateway_interface.h: Function definitions and API documentation for the ANT BLAZE Gateway Library.

    - inc\ant_blaze_defines.h: Includes constants and data structures that are used by both libraries.

Copy the bin and inc folders of the ANT BLAZE Libraries package into the ANT_BLAZE_SDK folder.

2.3 ANT BLAZE Software Development Kit

The contents of the ANT BLAZE Software Development Kit are as follows:

- Example applications:

    - examples\demo_node: Shows how to integrate the ANT BLAZE Node Library into an application. The example implements two messages, a ping request and ping response. When the node receives a ping request from the gateway, it will create a ping response and reply over the mesh. The ping request includes application level logic that allows addressing a range of nodes (e.g. nodes 1-2) with a single message. Messages received by the nodes are output over Segger RTT for debugging. The example application also implements a custom BLE service that sends messages received over the mesh over a BLE connection as a peripheral device, demonstrating how applications can use custom connectivity side by side with ANT BLAZE.

    - examples\demo_gateway: Shows how to integrate the ANT BLAZE Gateway Library into an application. This simple example uses the buttons on the AA board to send ping requests for specific ranges of nodes. Responses received by the gateway are output over Segger RTT.

- examples\np_gateway: Enables interaction with the ANT BLAZE Gateway Library from an external microcontroller or PC application over UART. This example defines serial messaging to translate library events and configuration commands over the serial interface. The framing is compatible with the standard ANT serial messaging format described in the "ANT Messaging Protocol and Usage" document (Section 7.1), for compatibility with ANT software libraries for PC/Mac/Linux.

Project files for all example applications for GCC, IAR and Keil MDKv5 are available under the armgcc, iar and keil directories under each project directory.

- Mesh Tester: PC application that interfaces with a Premium D52 module running the np_gateway application to implement a more feature-rich gateway to evaluate ANT BLAZE technology, allowing both manual messaging and automated polling of nodes with collection of performance statistics. Mesh Tester requires .NET Framework 4.0 and the Visual C++ 2008 SP1 Redistributable Package from Microsoft.

2.4 ANT s332 SoftDevice

Refer to the release notes in the ANT BLAZE SDK package for SoftDevice version compatibility. The SoftDevice API headers (the contents of the include directory in the SoftDevice download package) will need to be copied into the components\softdevice\s332\headers directory of the Nordic nRF5 SDK. If the directory does not exist, create it.

The ANT s332 SoftDevice can be downloaded from:

https://www.thisisant.com/developer/components/nrf52832#tab_protocol_stacks_tab

2.5 ANT USB Interface Board Driver

Download the ANT USB interface board for Windows from: http://www.thisisant.com/developer/resources/downloads

Install the drivers. Note that the drivers are unsigned. On Windows 7, when installing the drivers, you may receive a warning message that Windows can't verify the publisher of the driver; you should install the driver anyway. Windows 8 and 10 require signed drivers for installation, so you will be required to boot with driver signature enforcement disabled to complete the installation process. For more detailed installation instructions, refer to Section 2.1.1 of the "ANT SoC Module Starter Kit User Manual"

2.6 Nordic nRF5 SDK.

Download version 13 the Nordic nRF5 SDK from http://developer.nordicsemi.com/

Install the SDK for the IDE/Compiler of your choice. To get the projects in the ANT BLAZE SDK to build out of the box, install the Nordic nRF5 SDK in the nrf5_sdk directory inside the ANT BLAZE SDK. The final directory structure should look like Figure 2.

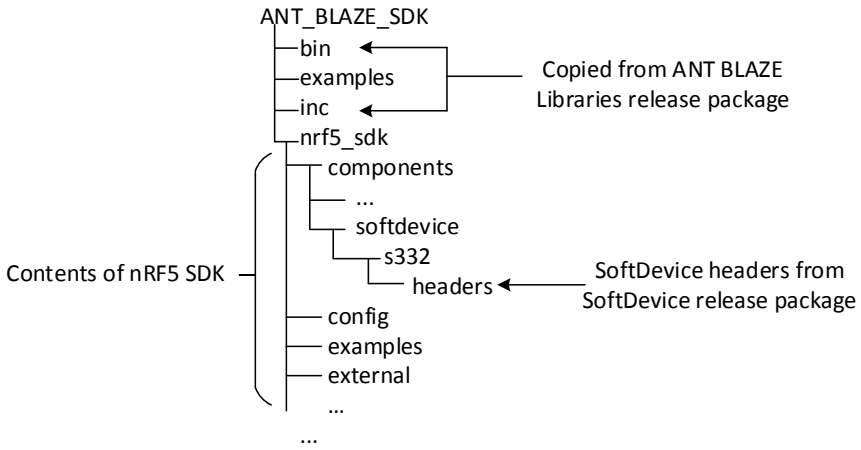

Figure 2. ANT BLAZE SDK Directory Structure

For more detailed instructions on toolchain setup using Keil, refer to the "ANT SoC Module Starter Kit User Manual"

2.7 RTT Viewer

GUI application that displays terminal output from a device connected via a Segger J-Link programmer.

https://www.segger.com/jlink-rtt-viewer.html

3 Setting up a Network

3.1 Setting up the Gateway

1. Build the np_gateway project. The first time you build the project, the compiler will point out two errors which serve as reminders that licenses must be obtained to use the S332 SoftDevice and the ANT BLAZE libraries. Follow the instructions in the error messages to enable the evaluation keys and allow the project to build.

2. Insert two AA batteries on an ANT AA battery board and mount a Premium D52Q module onto the board.

3. Connect the Segger J-Link programmer to the programming header on the module. Turn the power switch to the ON position. The setup should look as in Figure 3.

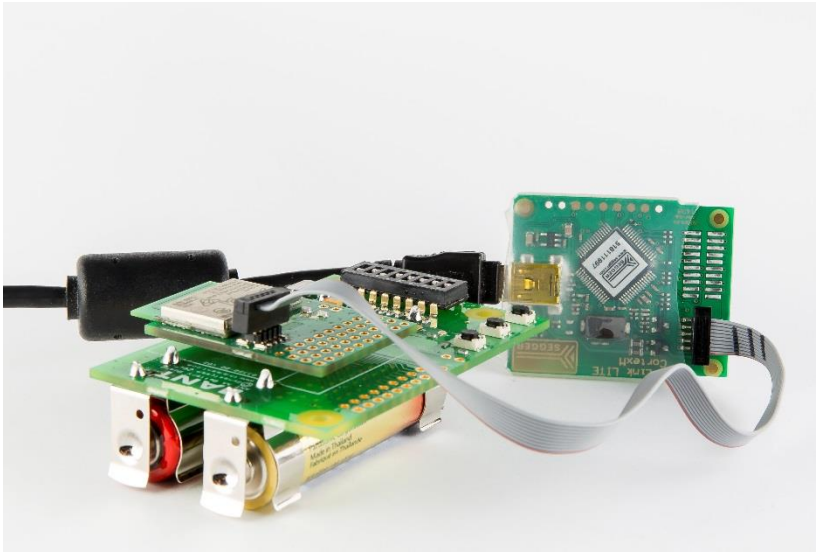

Figure 3. Module Programming Setup

4. Load the module with the application np_gateway.hex which was generated by building the np_gateway project in step 1. Note that this file will be in different places depending on which compiler was used. Check the IDE's build output settings or the Makefile to determine which directory it was placed in.

5. Turn the power switch to the OFF position and remove the module from the board. Disconnect the J-Link programmer.

6. Mount the module onto an ANT USB Interface board and connect it to a USB port of a PC, as shown in Figure 4.

Figure 4. ANT BLAZE Gateway Board Setup

3.2 Setting up the Mesh Nodes

1. Build the demo_node project.

2. The node ID for each of the mesh nodes is derived from the configuration of the dip switches. Set up the dip switches on the 4 ANT AA battery boards to the numbers 1-4, in binary (with SW1 as the least significant bit and SW8 as the most significant bit), as described in Table 1.

Table 1. Dip Switch Configurations

Board	SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1
Board 1	0	0	0	0	0	0	0	1
Board 2	0	0	0	0	0	0	1	0
Board 3	0	0	0	0	0	0	1	1
Board 4	0	0	0	0	0	1	0	0

For example, the dip switches in board 3 should be configured as depicted in Figure 5.

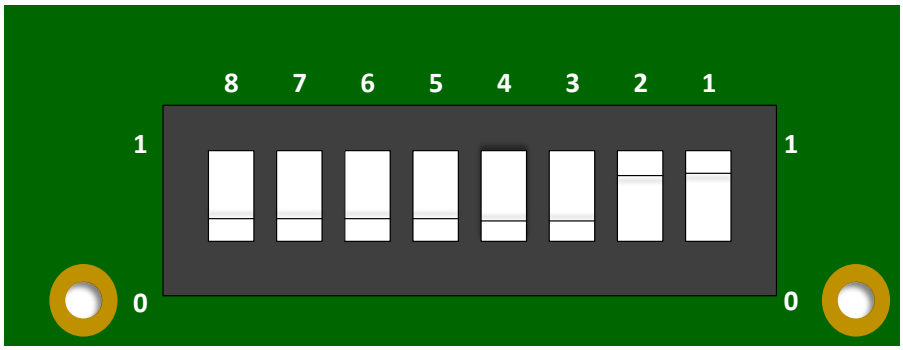

Figure 5. Dip Switch Node ID Configuration

3. Insert two AA batteries on all the battery boards, and mount a Premium D52 module on each.

4. For each of the battery board/module stacks, connect the Segger J-Link programmer to the programming header on the module, turn the power ON and download the hex file demo_node.hex which was generated by building the demo_node project in step 1.

5. Distribute the nodes around and make sure that all the battery boards are still ON. The network is now ready to be tested.

6. To see the output of one of the nodes, connect the Segger J-Link programmer to the programming header on the module and start RTT Viewer. Under "Connection to J-Link" select USB, and under "Target Interface & Speed", select SWD, and press "OK". See Figure 6.

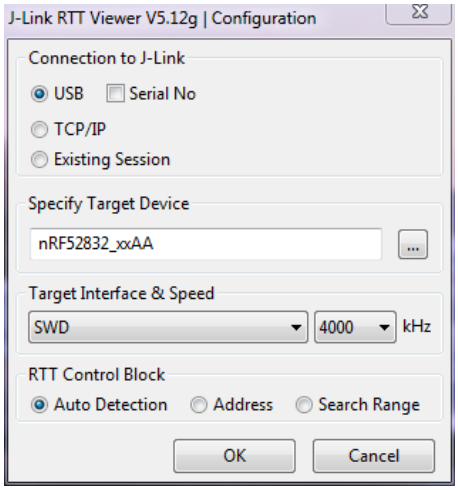

Figure 6. RTT Viewer Configuration

7. A console window, like one shown in Figure 7, with the debug output from the node will now be visible.

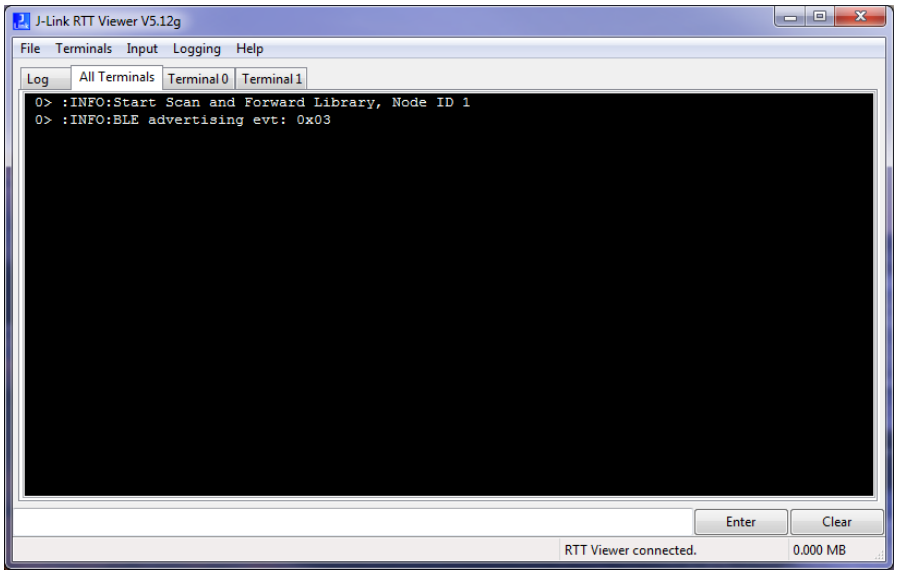

Figure 7. Example RTT Viewer Output

4 Testing Network Messaging using Mesh Tester

The Mesh Tester is a PC application that can be used to evaluate the performance and the features of the ANT BLAZE network. Ensure the network is set up as above and a gateway node has been inserted in the USB port of the PC. Follow these steps to configure Mesh Tester:

1. Launch the Mesh Tester PC tool.

2. Click the "Select a Gateway Configuration File" button and load exampleGatewayConfigFile.xml. The Gateway Configuration file specifies all the configuration parameters required to set up the gateway. For more details on the specific fields, refer to the gateway configuration structure definition in ant_sf_defines.h. All configuration parameters (except for the Node ID which must be unique) of the nodes and the gateway must match for the network to function correctly.

4.1 Manual Messaging

Messages may be sent manually into a network using the Mesh Tester.

1. To enable Manual Operation mode of the Mesh Tester, load a Gateway Configuration file as above and then click the "Start Gateway" button.

2. Populate the "Payload" text box with data up to 40 bytes long. The demo_node firmware is designed to automatically respond to special ping request messages to facilitate testing. Table 2 contains the definition of a ping request message.

Table 2. Format of a Ping Request Message

Byte	Description
0	Identifies this message as a ping request. Set to 0x01.
1	Used in automated testing. Set to 0x00.
2	Used in automated testing. Set to 0x00.
3	Bits 0-3: Set to number of packets to send in response (1 - 8).
	Bit 4: Set to 1 if bytes 5+ of the request payload are to be copied into bytes 5+ of the response. In this case, the length of the request from the gateway should match the requested response from the node. Set to 0 otherwise.
	Bits 5-7: Unused. Set to 0x00.
4	Unused. Set to 0x00.
5+	Optional. Data here will be copied into the ping response of the responding node.

3. Populate the "Address" box with one of the following:

    - A Node ID – Sends this message only to that specific node.
    - "0" – Sends this message to every node in the network.
    - A Group Address – Sends the message to every node belonging to that group. The demo_node firmware automatically assigns all nodes to groups 511 and 500. Look at the source code for more information on groups.

4. Click the "Send Message" button to send the message. Message traffic in and out of the Mesh Tester can be observed in the "Filtered Raw Messages" tab.

Example: Send the message 01-00-00-01-00 (a ping request message) to any Node ID in the network. The outgoing message will appear in the "Filtered Raw Messages" tab shortly followed by the incoming response from the node. The expected output can be seen in Figure 8.

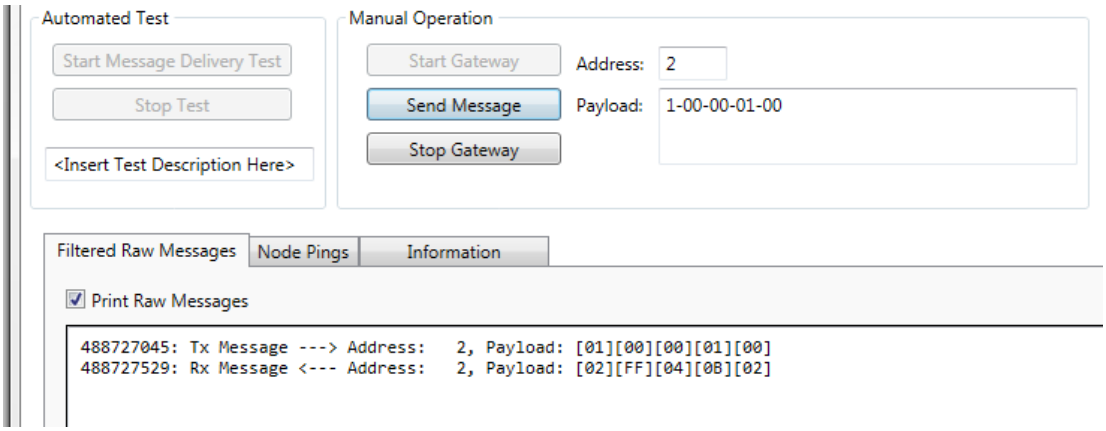

Figure 8. Example Output from Manual Messaging

4.2 Automated Testing

Mesh Tester can be used for automated testing. The Mesh Tester rotates through a list of all the nodes in the network and sends ping request messages. It then measures the amount of time it takes for each node to respond. In addition to the Gateway Config file mentioned earlier, the automated test relies on 2 more configuration files:

- Test Configuration (XML) – Specifies the test parameters
    - NumberOfTestCycles – Number of times to loop through the list of nodes in the network.
    - MaxPingWaitTimeInMs – Maximum amount of time to wait for a response before moving on to the next request.
    - StartTimeDelayInMs – Test start delay. Useful to let things settle before starting the test.
    - NumberOfPacketsPerResponse – Number of 5-byte packets each node should send in its response.
    - NumberOfNodesToPingPerRequest – Number of nodes to request a response from at a time. Note: must be a factor of the total number of nodes in the network.
    - SplitLogFiles – The test log files can be split after a certain number of cycles. This is useful for long term testing so the log files do not grow too large.
    - NumberOfCyclesPerLogFile – If SplitLogFiles is enabled, the log files will be split after this many cycles have been completed.
    - EchoPayload – Enable to make the nodes respond with specific data.
    - PayloadBytesToEcho – If EchoPayload is enabled, this data will be echoed back in each node's response.
- Device List (CSV) – A list of all Node IDs in the ANT BLAZE Network.

The example configuration files will work if the nodes in the network are running the demo_node firmware.

To run an automated test:

1. Set up an ANT BLAZE network with the demo_node firmware as explained earlier in this document.
2. Use the buttons in "Gateway and Test Configurations" to load each of the configuration files. Example files, exampleTestConfigFile.xml and exampleDeviceList.csv, have been provided which will work with a network running the demo_node firmware.

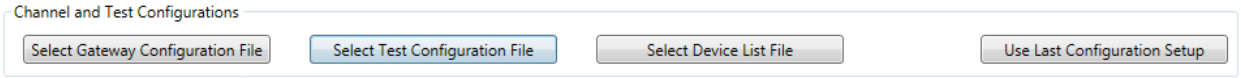

Figure 9. Loading Test Configuration and Device List Files for Automated Testing

3. Click "Start Message Delivery Test" button to begin the test.
4. Observe the test progress in the "Node Pings" tab. Raw messages to and from the gateway can be observed in the "Filtered Raw Messages" tab.

The Mesh Tester generates 4 types of log files which can be found in a timestamped folder in the executable directory. These files are:

- Summary File – Results of the test per node, per ping request, and overall metrics of the test.
- Pings File – Log of everything printed to the "Node Pings" tab.
- Messages File - Log of everything printed to the "Filtered Raw Messages" tab.
- Debug Info – Information regarding the connection to the gateway device.

5 Appendix A: AA Battery Board Schematics

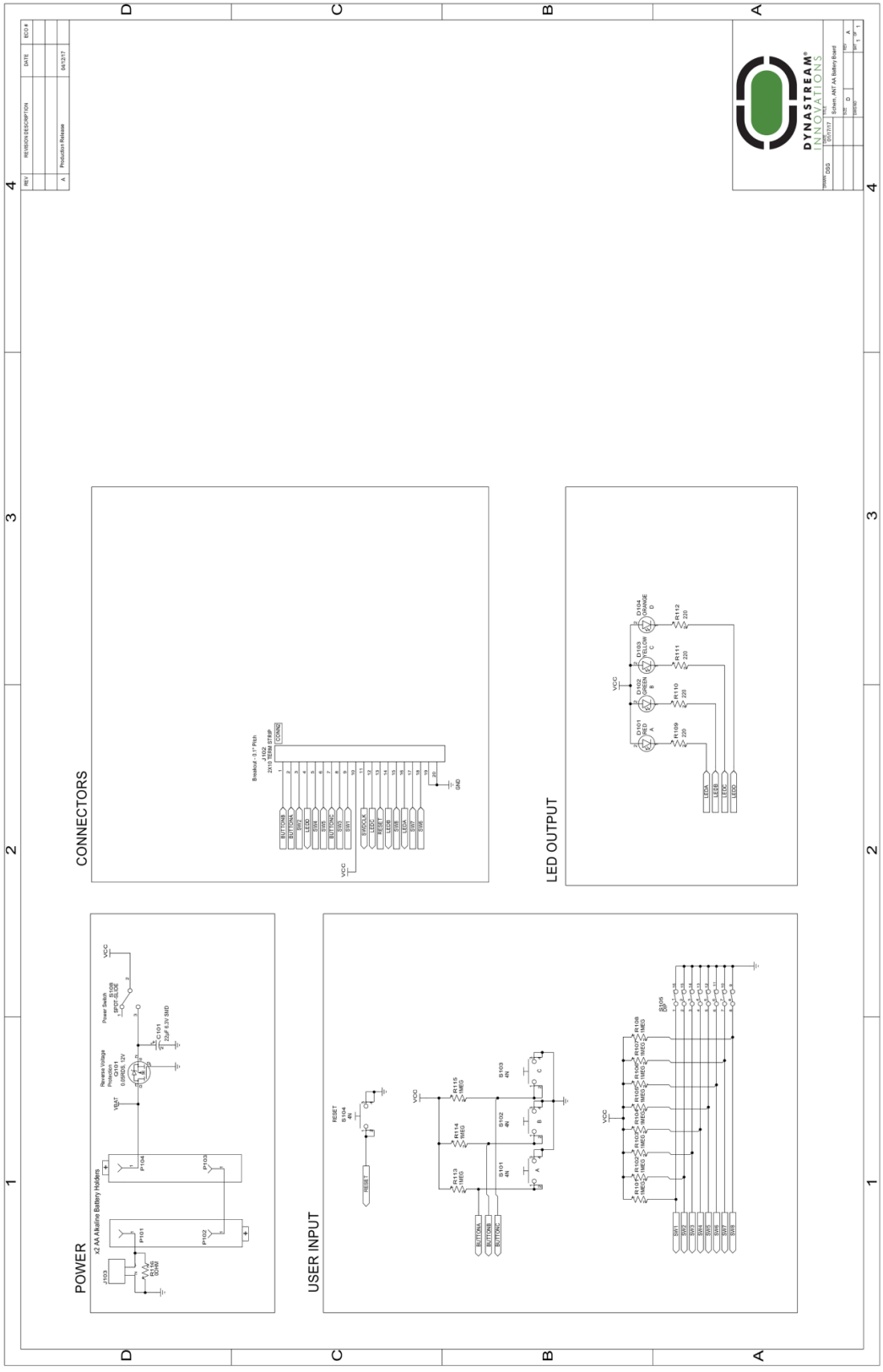

Figure 10. AA Battery Board Schematic

Table 3. AA Battery Board Pinouts

Pin Name	Molex (CONN1) Pin #	Breakout (CONN2) Pin #
VCC	1	10
SW1	2	9
BUTTONC	3	7
LEDC	4	12
SW3	5	8
SWDCLK	6	11
SW4	7	5
LEDB	8	14
SW5	9	6
RESET	10	13
SW2	11	3
LEDA	12	16
LEDD	13	4
SW8	14	15
BUTTONB	15	1
NOT CONNECTED	16	
BUTTONA	17	2
SW6	18	18
GND	19	19, 20
SW7	20	17