Dweet Library VSA User's Guide

Table of Contents

1. Introduction

The software included in this VSA allows Renesas Synergy developers to quickly and easily build projects that enable Synergy hardware to leverage the multi-faceted functionality of the Dweet™ IoT cloud.

Dweet™ is a secure, high-performance, data streaming and device management system that serves as the foundation for IoT application development. Dweet enables customers to build, scale, and manage bi-directional messaging infrastructure for their applications and devices. Notable features of the Dweet platform include:

- Easily connect devices – for existing products and new product designs
- Real-time messaging, alerts & storage
- Meaningful visualizations of device data and behavior
- Flexible mash-up dashboards connecting to multiple data sources & device types
- Enables faster / cheaper time-to-market with rapid prototyping, testing and evaluation of IoT Applications

2. Dweet Library VSA Description

The Dweet Library VSA includes a folder containing the DweetLib C library ("DweetLib Files") and an example implementation e2studio project ("DweetClientExample"). The DweetLib static library and header files provide other projects with the wrapper functions necessary to send and receive data to and from Dweet.io. The DweetClientExample shows application developers how to use these wrapper functions programattically and, when flashed on a Synergy development kit, provides a working demonstration of the kit connected to Dweet.io and associated cloud-based applications & tools (Freeboard.io).

3. Technical Requirements

- A Renesas Synergy development board (for example, SK-S7G2)
- A PC running Microsoft® Windows® 7 or 10 with the following Renesas software installed:
- e2studio ISDE version 5.1.0 (or above)
- Synergy Software Package (SSP) version 1.1.3
- SSP Developer Licence (obtained by signing up and logging in to the Renesas Synergy Gallery)
- Access to a wired ethernet connection

You can download the required Renesas software from the Renesas Synergy Gallery (https://synergygallery.renesas.com).

4. User Prerequisites

The example project and C library files supplied with this VSA are designed to be used with a supported Synergy hardware development kit. Users are assumed to have some experience with Renesas e2studio ISDE, SSP, and their testing hardware. It is highly recommended that any beginners to e2studio and/or SSP follow the procedures outlined in the getting started guides for their respective Synergy kit, including the "Blinky" project.

It is also recommended that users become familiar with the procedures detailed in the Synergy SSP Import Guide (https://www.renesas.com/en-us/doc/products/renesas-synergy/apn/r11an0023eu0113-synergy-ssp-import-guide.pdf) in order to understand how to install this VSA software into their ISDE and flash their Synergy development kit via the debugging functionality of e2studio.

5. Usage Instructions

## Installation and Running the Demo

The VSA software is distributed as an archive file containing one e2studio SSP project folder (the example/demo project) and a separate folder containing just the static library and header files. Carefully follow the steps in **Section 2: Import Project into e2 Studio ISDE** of the Synergy SSP Import Guide, replacing the "NETX_DNS_DK-S7G2.zip" example with this VSA's zip file. The guide will bring you through importation of the example project from the archive file, configuration within the e2 studio ISDE, and flashing of the VSA demo application to your development kit.

https://www.renesas.com/en-us/doc/products/renesas-synergy/apn/r11an0023eu0113-synergy-ssp-import-guide.pdf)
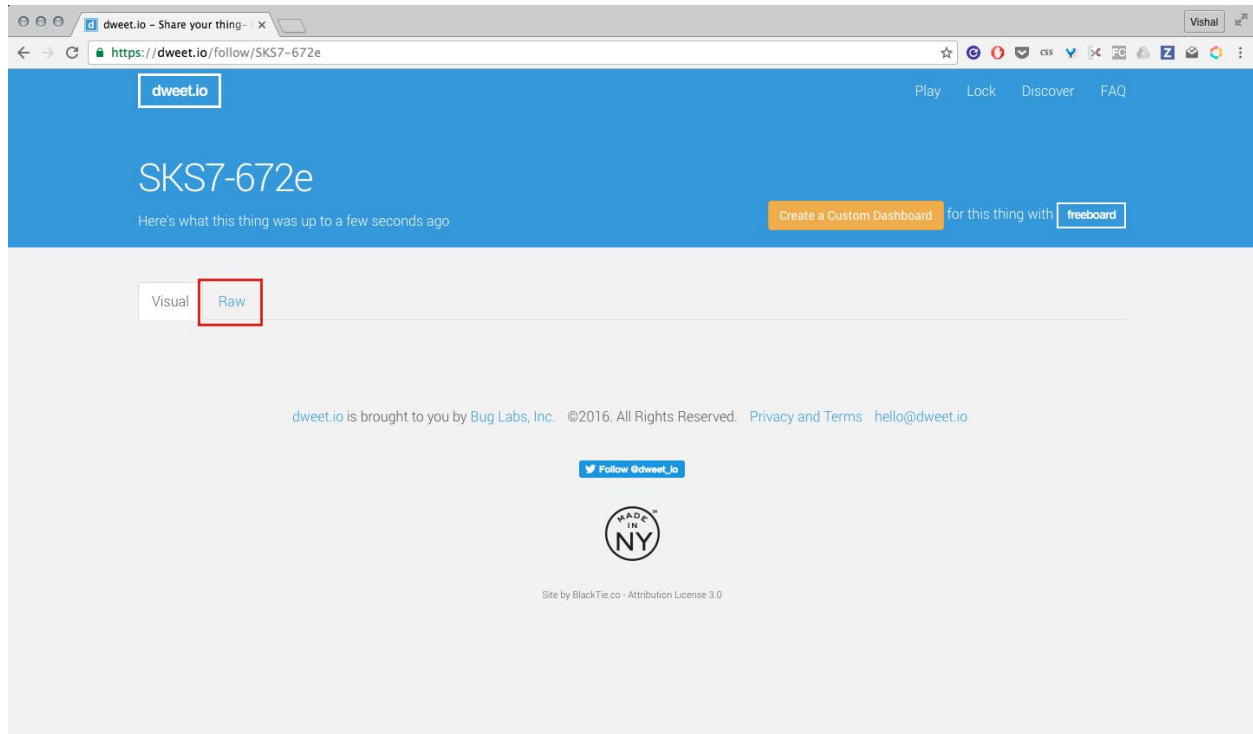
Before running the DweetClientExample application, make sure your board is connected via wired ethernet to an internet-enabled router. Once the application is successfully running on your Synergy development kit via the e2 studio debugger (Step 11), make sure you are still in the Debugging Perspective, and open the Renesas Debug Console:
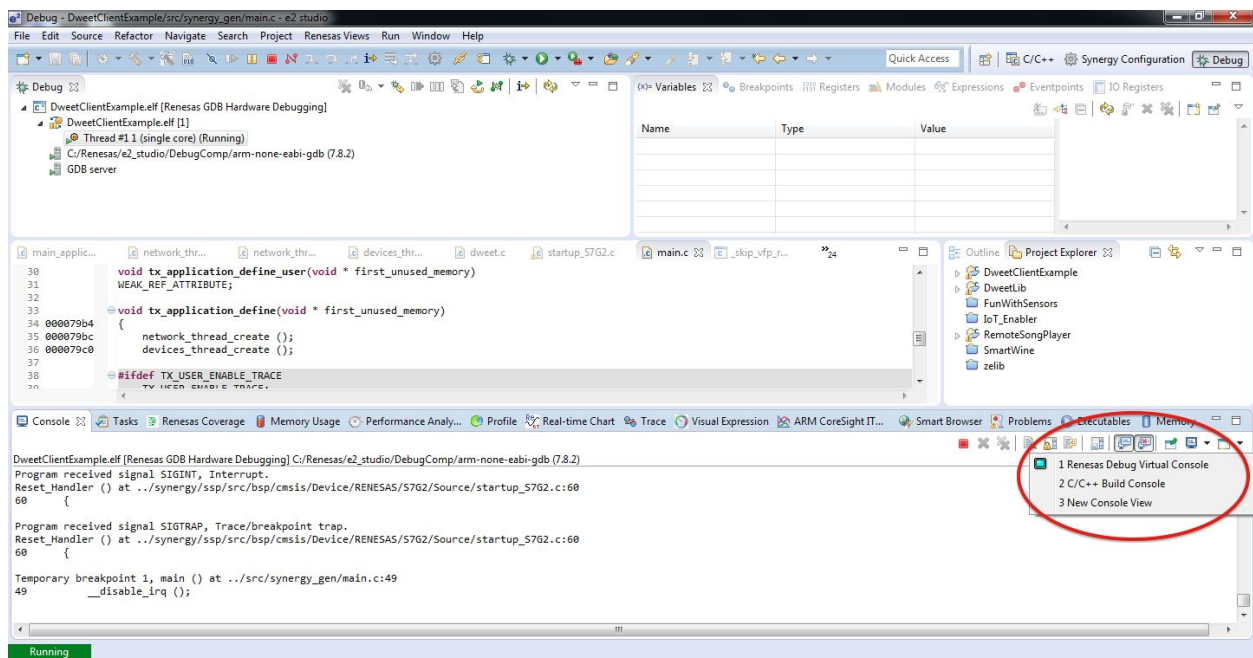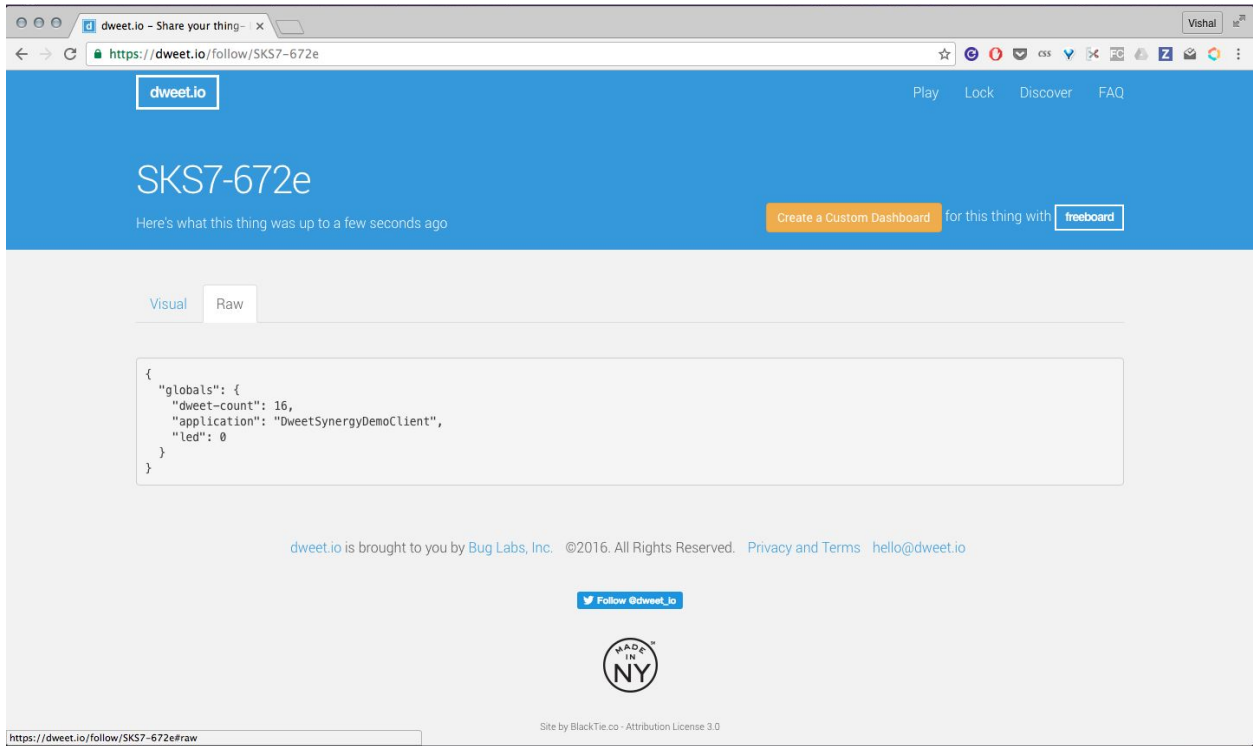
The console will show the unique board ID ('thing name') and the url that can be used to see live data coming from the board to Dweet.io
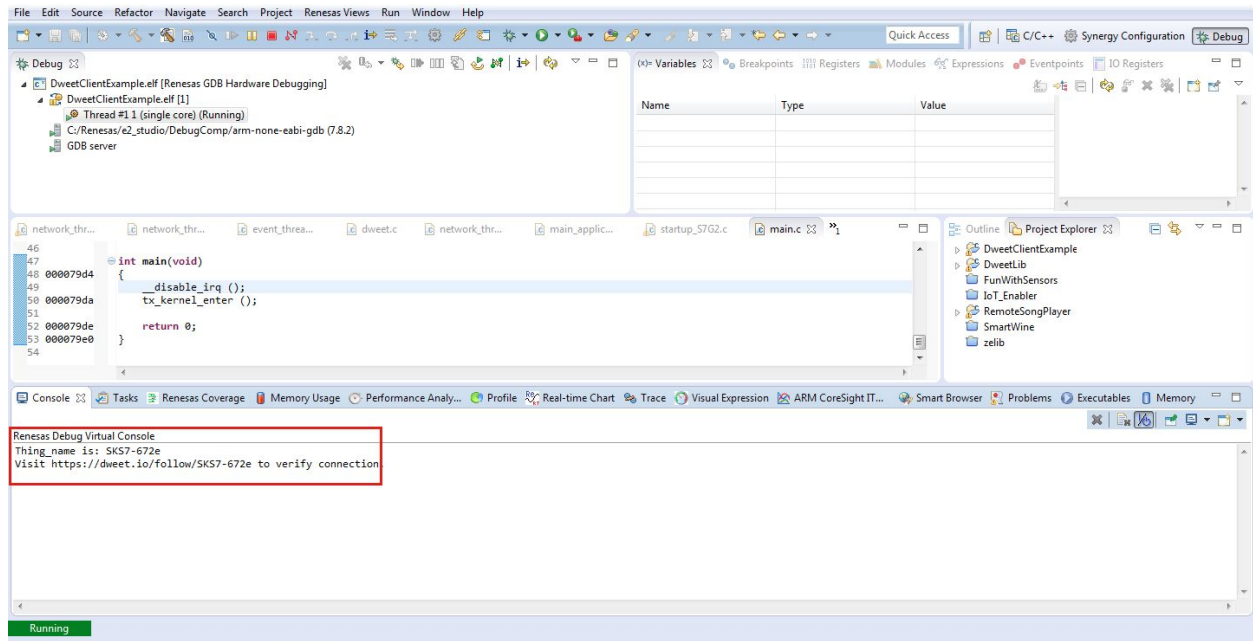
Open a browser, and navigate to this URL. Click the Raw tab to see the live example data coming from your Synergy hardware:
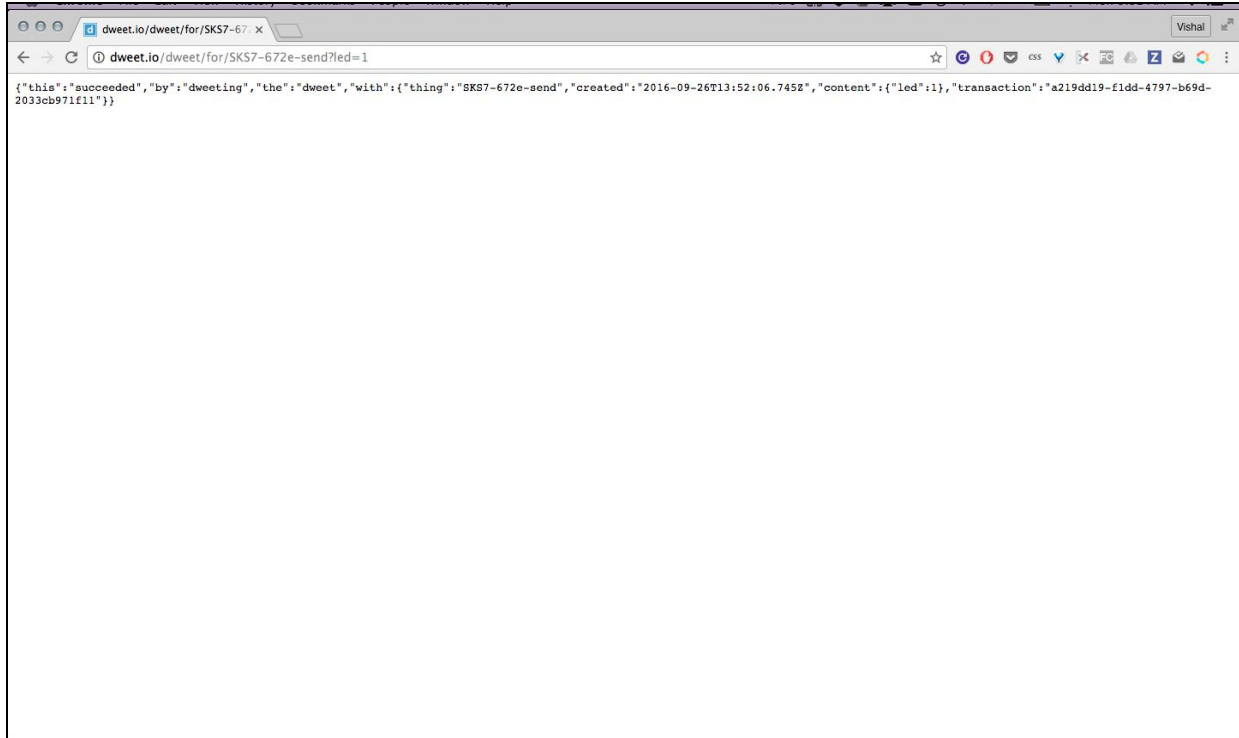
To demo the example command functionality to the board, toggle the red user LED by sending an HTTP GET to the following URL: https://dweet.io/dweet/for/<BoardID>-send?led=1
You can use cURL, a popular command-line software tool, or simply enter the URL into a browser as follows:

{"this":"succeeded","by":"dweeting","the":"dweet","with":{"thing":"SKS7-672e-send","created":"2016-09-26T13:52:06.745Z","content":{"led":1},"transaction":"a219dd19-f1dd-4797-b69d-2033cb971f11"}}

The red LED should illuminate within seconds, and the live data payload's LED status coming from the board should reflect accordingly:
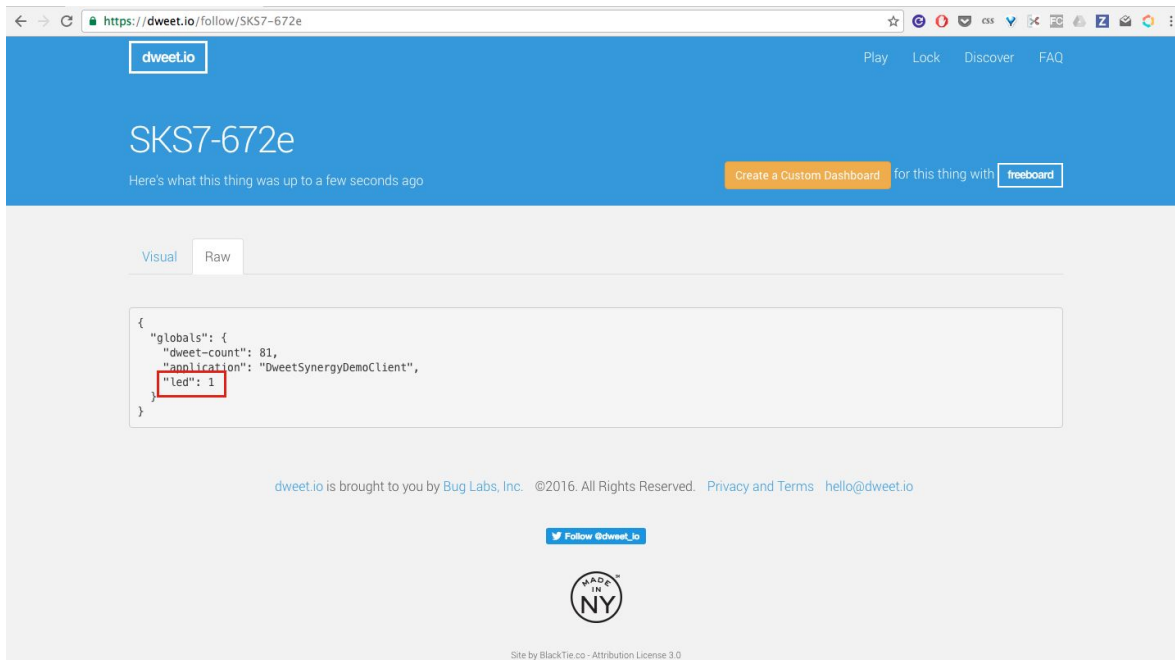
6. Demo Dashboard

Renesas Synergy projects using the Dweet Client VSA can leverage the abundant community of cloud applications that integrate with Dweet-connected devices, including Freeboard.io. Navigate to http://ssp.freeboard.io:
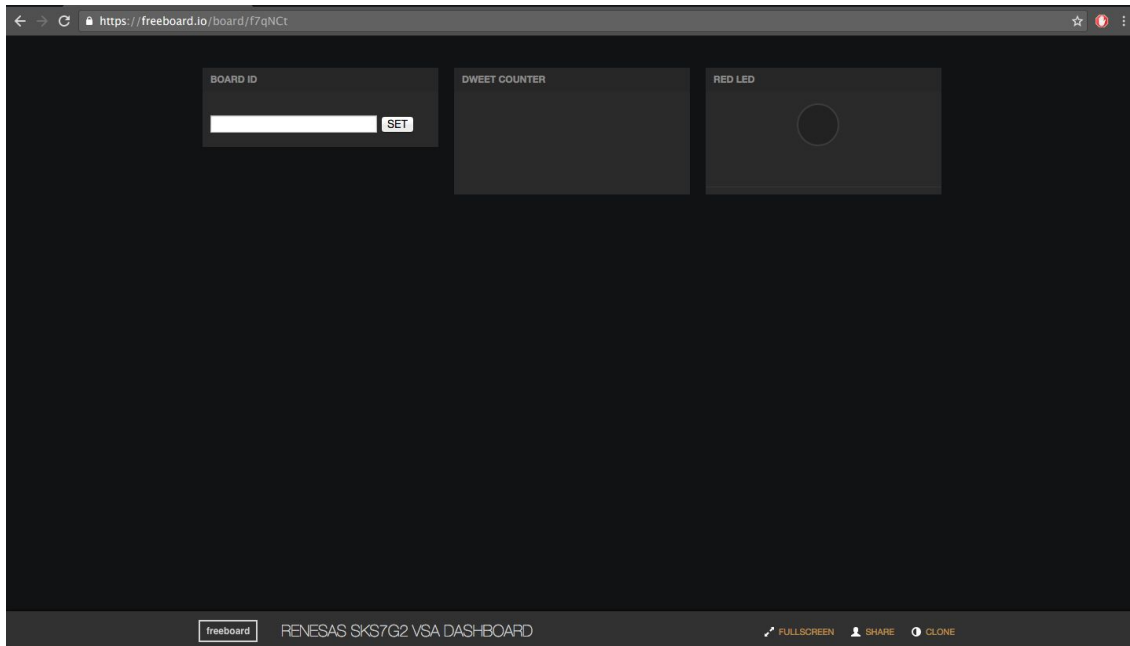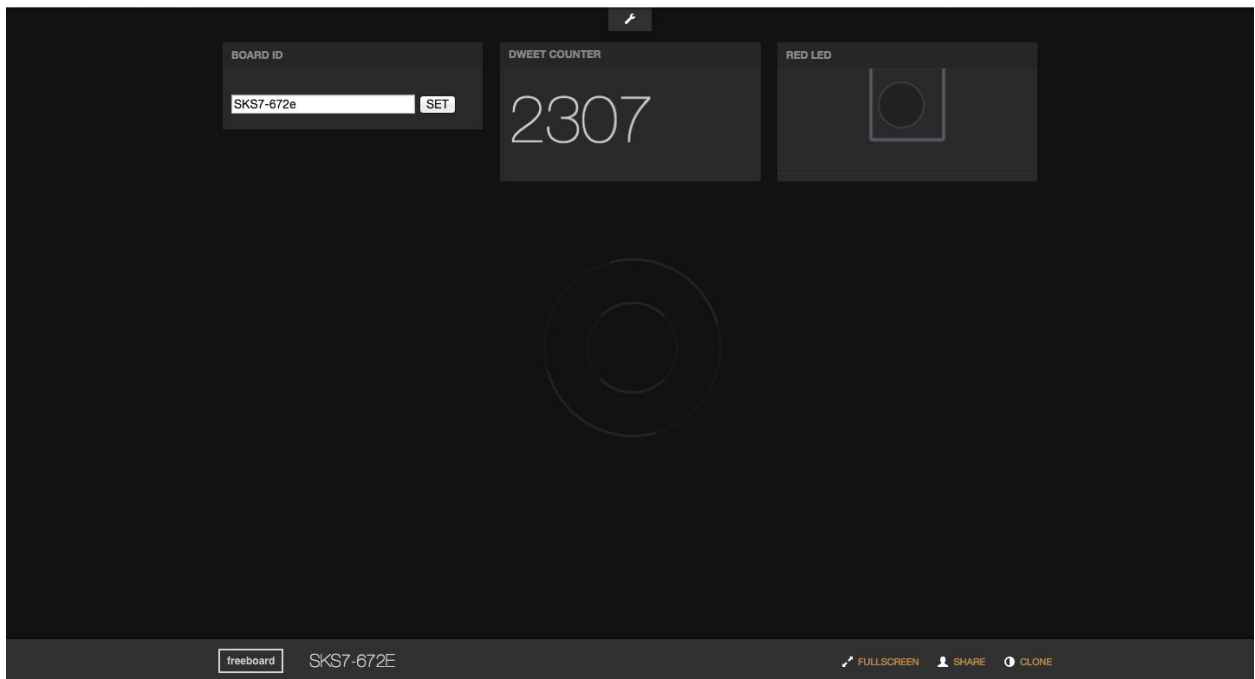

Enter the Board ID, and click SET. Within seconds, the live Dweet counter and LED status will populate in the dashboard user interface.
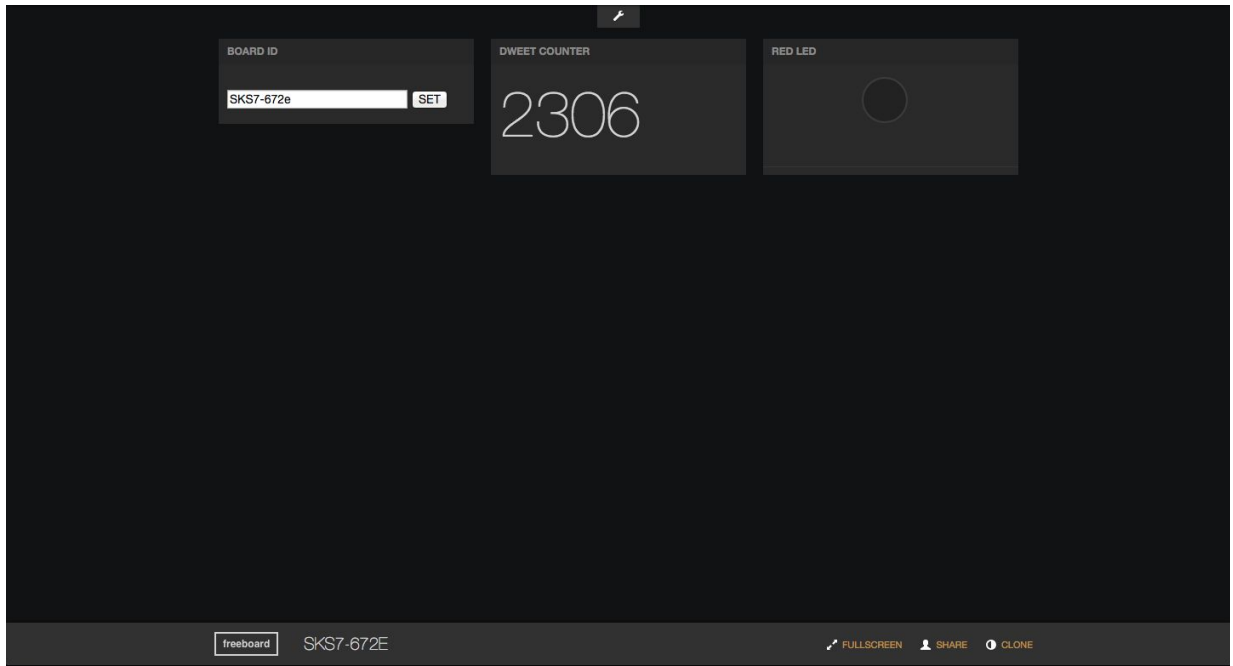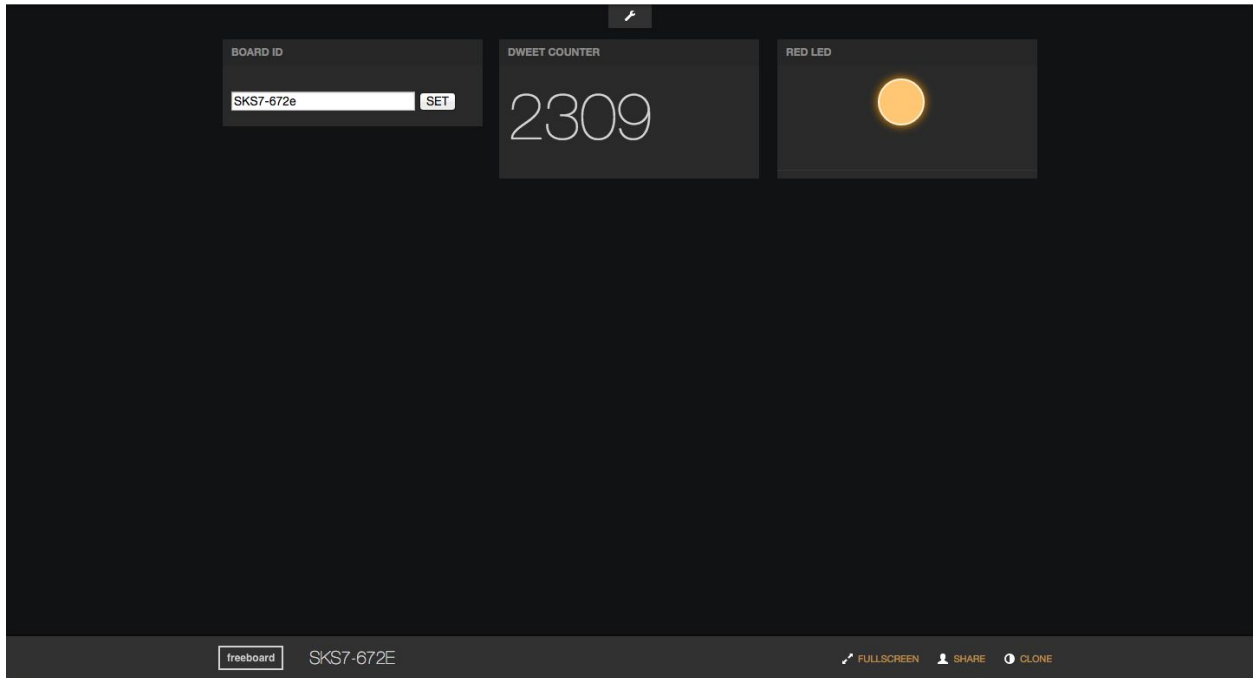
To demo control functionality to the board, click on the RED LED indicator to toggle the board's red user LED.

The LED indicator will reflect the board's current LED status.

7. Source Code Details

DweetLib C Library Files

This folder includes subfolders which contain the C static library (/lib) and header files (/inc) that enable a SSP project to send and receive data to and from Dweet. When adding to an application project, we recommend creating a folder within the project's /src folder (in our DweetClientExample, this folder is called "dweet"), then copying both library subfolders into this new folder.

File Descriptions:

- /lib/libDweetLib.a
  - C Static Library file
- /inc/dweet.h
  - Provides application level functions and settings that interface with the Dweet library
- /inc/dweetlib-utils.c
  - Provides utility functions and custom data types used by the Dweet library.
- /inc/dweetlib-properties.h
  - Provides custom data types (called "properties") used to encapsulate data variables that will be sent to/from the Dweet library.

DweetClientExample e2studio Project

Key Source Files:

- /src/main_application.c
  - Main SSP application entry – initializes the Dweet client, creates the devices thread and dweet thread.
- /src/dweet_thread_entry.c
  - SSP-generated file, launches dweet_thread_main
- /src/dweet_thread_main.c
  - Handles all Ethernet networking functionality, creates thread that periodically sends any data added via the Dweet library data queuing interface and checks for any data sent to the board. An example of command-and-control functionality triggered by commands sent to the board is implemented as well, allowing a user to toggle the red user LED from the cloud.
- /src/devices_thread_entry.c
  - Example of additional thread usage for including sensor data/hardware into data queuing interface. The default example polls the status of user LED and adds that value to the Dweet client.
- src/dweet/
  - Directory holding the DweetLib static library and header files.

8. Troubleshooting Notes

- While programming your Synergy hardware via e2 studio debugging, make sure you have pressed the Resume button in e2 studio's debugging perspective enough times (twice in most cases) to surpass SSP's default breakpoints so that the application enters its main thread.
- Once the DweetClientExample application is running, the yellow user LED will illuminate as the networking connection is established. Once an IP address and DNS are successfully established, the yellow user LED turns off and green user LED illuminates. If the green user LED does not illuminate after 10-20 seconds, make sure the wired ethernet connection is secure and that the router is internet-enabled.