Dweet.io Client VSA Design Document

What is this VSA?

The software included in this VSA enables a developer to easily send data and control between Synergy hardware devices and 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 (publish/subscribe) infrastructure for their applications and devices.

Dweet™ uses the familiar HTTP/HTTPs protocol and JSON formatted responses via RESTful APIs. This standard approach helps the platform conform to the highest security requirements that are, in most cases, enforced by private organizations' firewall policies. Other protocols, for example MQTT, are also supported, based on specific customer needs.

What functionality will be implemented?

Bug Labs intends to deliver a client library that will enable bidirectional communication between the Synergy Development Kit and the dweet™ server. Users can either view the sensor data by consuming a RESTful API, or by visualizing the data in the freeboard™ platform, or send commands to the device to trigger an operation.

For this VSA, we are providing source code and documentation that enables a SSP application developer, using the SKS7 board and the e2studio IDE, to build applications that communicate with Dweet™. The Dweet client library will expose APIs to the application development layer that abstract the configuration, sending and receiving of data via Dweet.

System Requirements

SSP Version: 1.1.3
ISDE Version: e2 studio Version: 5.1.0.022
Currently Supported Hardware: SK-S7G2-REV3.0

SSP Interoperability

SSP Resources Used

ThreadX, NetX, NetX DHCP, NetX DNS

RTOS requirements

ThreadX (and ThreadX components, noted above)

SSP APIs used

s7g2_sk (BSP), nx, nx_dhcp, nx_dns, tx, r_dtc, r_elc, r_flash_hp, r_fmi, r_glcd, r_gpt, r_riic, r_rtc, r_sci_common, r_sci_i2c

Note: Certain basic HAL libraries, though not implemented in the Dweet Client, are included by default for user convenience in the application development layer.

Architecture block diagram
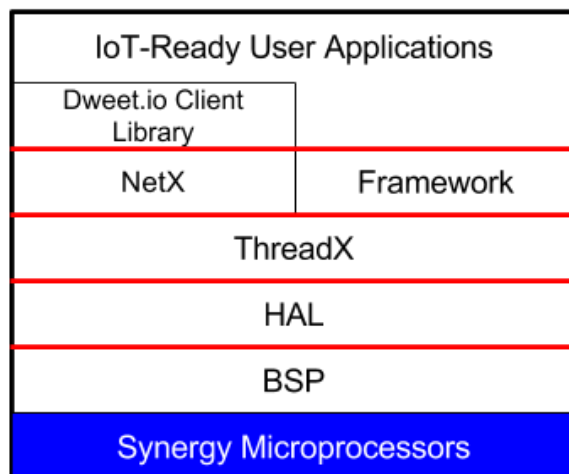
Software Details

The intended VSA delivery format is an archive file containing two projects – the Dweet Client library project ("DweetLib"), and an implementation example ("DweetClientExample").

DweetLib

This project includes source code that provides external applications with the wrapper functions necessary to send and receive data to and from Dweet.
Key source files:
- /src/dweet.c.
  - main source file implementing and exposing external hooks to configure/send/and receive dweet data
- /src/zelib.c
  - Helper library for properly encapsulating application-level sensor data into a common format to be used to send dweets.
- /src/json.c
  - Helper library for creating and parsing JSON-formatted objects

Other included files:
- /src/zelib-gpio.c; /src/zelib-i2c.c; /src/zelib-uart.c
  - These source files can be used as reference when including support for sensor hardware attached to any of these interfaces

DweetClientExample

Key Source Files:
- /src/Main_application.c
  - main file – initializes Dweet Client and creates the networking thread that uses the Dweet client
- /src/network_thread_entry.c
  - handles all Ethernet networking functionality, creates thread that periodically sends any data added via the DweetClient's data queuing interface (zelib) 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 status of user LED and updates Dweet Client's data queuing interface accordingly.