Contact Tracing Data Logging

Sentrius™ BT710 Tracker

Application Note *v1.0*

1 Introduction

This document provides some example data as published by Laird Connectivity gateway products when used in conjunction with the Sentrius™ BT710 tracker.

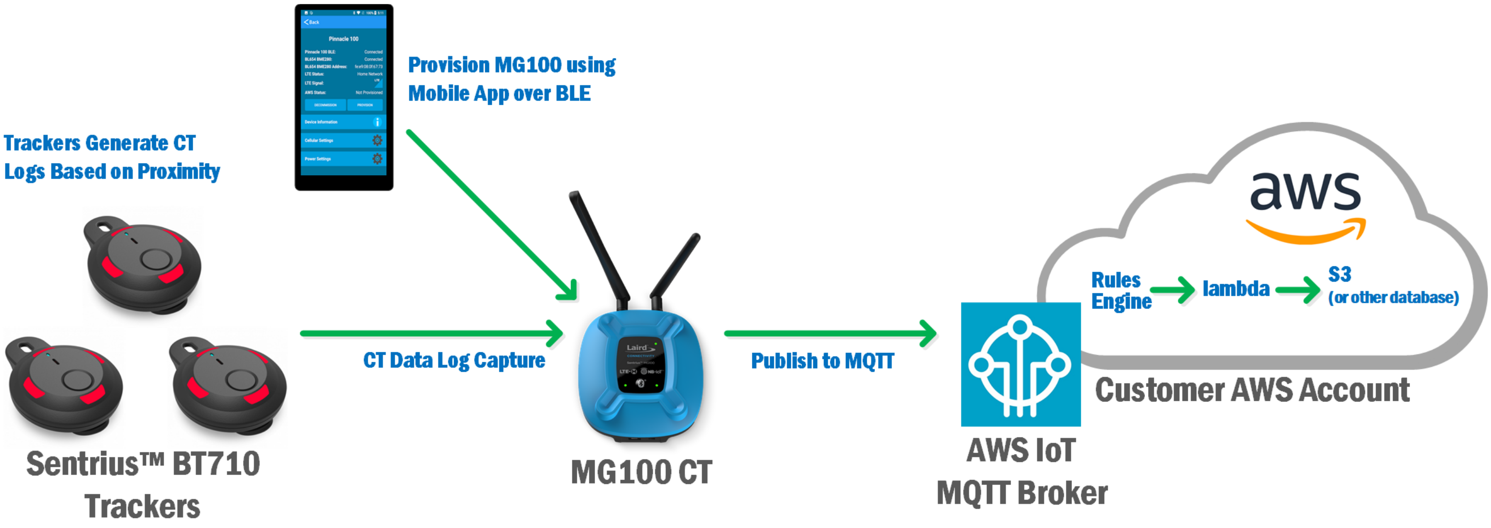
2 Overview

The Sentrius™ BT710 tracker's firmware is designed for detecting and logging when trackers are within proximity of one another, and sending and receiving data to communicate this. This firmware allows trackers to indicate their presence and sense the presence of other nearby compatible trackers. The behavior of BLE advertisement transmission and scanning/discovering other trackers is configurable based on use case via parameters (see the Sentrius™ BT710 User Guide Parameters section for details on user-configurable parameters).

3 System Requirements for Data Logging

In order to capture data logs from the BT710 Bluetooth tracking device, your system must include a Laird Connectivity gateway with contact tracing (CT) compatible firmware. The BT710 tracker will automatically interact with a properly configured gateway over Bluetooth Low Energy, allowing the gateway to download and forward CT data logs to its configured cloud-based endpoint. The Laird Connectivity MG100, IG60-BL654 and IG60-SERIAL gateways are designed for connectivity to the AWS IoT platform, however they may be configured for other platforms with support from the Laird Connectivity Design Services team, if necessary (separate business arrangement and development contract required).

The figure below depicts a typical scenario for a small deployment of BT710 trackers and an MG100 BLE-to-Cellular gateway. In this scenario, BT710 trackers will sense one another, capture CT log data locally and when nearby the MG100 will forward their logs to AWS IoT.



4 Data Log Format

The CT log data is published by the gateway to a remote MQTT broker in JSON format. This makes it easy for a web service interested in the data to subscribe to the MQTT topic and parse the data for further processing and storage. This stored data may be retrieved to aid in contact tracing operations. This section describes the JSON payload format to help implementers of such a system understand the type and format of the data captured, so it can be integrated into various web services/APIs.

The JSON format for CT log data represents a series of events captured by a **reporting sensor** (sensor that scanned for other sensors, generated log entries and is now reporting that data in this JSON document) and a **remote sensor** (the sensor detected by the reporting sensor in a BLE scan). These JSON documents contain an array of one or more **entry** objects, each containing an array of one or more **log** objects. Each log object represents a single BLE scan performed by the reporting sensor where the remote sensor was detected in that BLE scan. The entries[n].timestamp value is the absolute timestamp in UTC time captured by the reporting sensor when the first log was added to the logs[] array. Timestamps of each log object can be calculated by multiplying the **delta** value with the **scanInterval** value and adding it to the **entries[n].timestamp** value in the parent entry object.

`{`	
`"entryProtocolVersion": 2,`	Indicates the protocol ID for this entire JSON object
`"deviceId": "01020304aaaa",`	BLE device ID for the reporting sensor
`"deviceTime": 1597936969,`	Timestamp from the gateway when it published this JSON message
`"lastUploadTime": 1597935821,`	Reporting sensor timestamp when prior log was captured by the gateway
`"fwVersion": "00000600",`	Firmware revision of reporting sensor, Major: **00**, Minor: **06**, Revision: **0600** can be interpreted as **v0.0.6**
`"batteryLevel": 3040,`	Battery voltage of reporting sensor in millivolts
`"networkId": 65535,`	Network ID of reporting sensor
`"entries": [`	Array of CT Entries
`{`	Entry object
`"entryStart": 165,`	Fixed value
`"flags": 253,`	Fixed value
`"scanInterval": 30,`	Seconds between BLE scans for the reporting sensor
`"serial": "01020304bbbb",`	Serial number of the remote sensor detected in a BLE scan
`"timestamp": 1597935576,`	UTC timestamp (seconds since epoch) of the first log in this entry
`"length": 112,`	Byte size of reported payload, only used for diagnostic purposes
`"logs": [`	Array of records associated with this entry
`{`	Record object
`"recordType": 17,`	Indicates the format for this record object
`"log": {`	Log object
`"delta": 0,`	Delta in “scan intervals” after this entry’s timestamp that this log object was captured. To calculate absolute timestamp, multiply the parent entry “scanInterval” value by this delta value and add to the parent entry “timestamp” value to get seconds since epoch in UTC.
`"rssi": -44,`	RSSI measured by reporting sensor for BLE scan of remote sensor
`"motion": 0,`	0: remote sensor not in motion, 1: remote sensor in motion
`"txPower": 0`	TX power advertised by the remote sensor during this BLE scan
`}`	
`]`	
`}`	
`]`	
`}`	

5 Example Log Data

An example JSON payload, published from a BT710 tracker with id **01020304AAAA** tracking proximity to another remote BT710 device with id **01020304BBBB,** is shown below. This log is similar to what will be posted by the Laird Connectivity gateway for parsing and processing by a web service, providing approximate timestamps, TX power the remote sensor was configured with and RSSI received at the sensing sensor (AAAA sensor) when the scan result was captured. Note that the **entries[ ]** and **entries.log[ ]** arrays may have many more values in practice.

```
{
    "entryProtocolVersion": 2,
    "deviceId": "01020304aaaa",
    "deviceTime": 1597936969,
    "lastUploadTime": 1597935821,
    "fwVersion": "00060000",
    "batteryLevel": 3040,
    "networkId": 65535,
    "entries": [
        {
            "entryStart": 165,
            "flags": 253,
            "scanInterval": 30,
            "serial": "01020304bbbb",
            "timestamp": 1597935576,
            "length": 112,
            "logs": [
                {
                    "recordType": 17,
                    "log": {
                        "delta": 0,
                        "rssi": -44,
                        "motion": 0,
                        "txPower": 0
                    }
                },
                {
                    "recordType": 17,
                    "log": {
                        "delta": 1,
                        "rssi": -50,
                        "motion": 0,
                        "txPower": 0
                    }
                },
                {
                    "recordType": 17,
                    "log": {
                        "delta": 2,
                        "rssi": -47,
                        "motion": 0,
                        "txPower": 0
                    }
                },
                {
                    "recordType": 17,
                    "log": {
                        "delta": 3,
                        "rssi": -49,
                        "motion": 0,
                        "txPower": 0
                    }
                }
            ]
        }
    ]
}
```

Revision History

Version	Date	Notes	Contributor(s)	Approver
1.0	4 Sept 2020	Initial Release	Scott Lederer	Chris Boorman