Application Note

Supporting Custom Sensors and Embedded Devices

Rev. 1.0 – June 23, 2020

Author: Chris Rogers

Copyright © 2020 SensiML Corporation. All Rights Reserved.

The information contained in this document and the accompanying software programs is protected by copyright. All rights are reserved by SensiML Corporation. SensiML Corporation reserves the right to modify this document without any obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of this product without the prior written consent of an authorized representative of SensiML is prohibited.

SensiML is a registered trademark, and the SensiML logo is trademark of SensiML. Other trademarks are the property of their respective companies.

Contact Information

Internet: https://sensiml.com/contact

Support: https://sensiml.com/support

Phone: 503-672-7367

Mail: 8285 SW Nimbus Ave., Suite #151
Beaverton, OR 97008, USA

Notice of Disclaimer

SensiML is providing this design, product or intellectual property, 'as is'. By providing the design, product or intellectual property as one possible implementation of your desired system-level feature, application, or standard, SensiML makes no representation that this implementation is free from any claims of infringement and any implied warranties of merchantability or fitness for a particular purpose. You are responsible for obtaining any rights you may require for your system implementation. SensiML shall not be liable for any damages arising out of or in connection with the use of the design, product or intellectual property including liability for lost profit, business interruption, or any other damages whatsoever. SensiML products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use SensiML products in these types of equipment or applications.

SensiML does not assume any liability for errors which may appear in this document. However, SensiML attempts to notify customers of such errors. SensiML retains the right to make changes to either the documentation, specification, or product without notice. Verify with SensiML that you have the latest specifications before finalizing a product design.

©2020 SensiML Corporation

Revision History

Version	Date	Name	Description
1.0	June 24, 2020	C. Rogers	Initial document release

©2020 SensiML Corporation

1 Introduction

1.1 Overview

SensiML Analytics Toolkit comes preconfigured to build AI sensor models directly for a variety of third-party embedded sensor evaluation kits such as those from Nordic Semiconductor, STMicro, and Quicklogic. Such kits are a great way to explore new application concepts and build early prototypes. Inevitably though, such kitted proof-of-concepts give way to need to customize sensors and/or substitute standard eval boards with custom board designs tailored to a specific end-use application. Such modifications may include one or all of the following:

- Similar types but different models of sensor ICs from those included in off-the-shelf eval boards
- Different types of sensors not included in eval kits (ex. Flow sensors, PIR, loadcells, strain gauges, pressure sensors, etc.)
- Custom or proprietary embedded boards for unique needs and during product development (ex. Customer's own prototype of an MCU/sensor device)

Given the limitless combination of MCUs, sensors, and board layouts possible, this document describes the protocols, supporting documentation, and reference code needed for SensiML users to adapt their customized board firmware and SensiML software configuration to properly interface and work in a manner consistent with the supported evaluation kits.

1.2 Steps Required

Connecting your custom sensors and/or hardware to SensiML software involves three tasks:

1. **Creating or modifying device data collection firmware** that conforms to the SensiML Interface Specification[1] using MQTT-SN[2] to remotely control the target device to either store and forward (via device flash) or live stream data to the Data Capture Lab
2. **Editing a SensiML Data Capture Lab configuration** JSON[3] file to allow SensiML to recognize and properly handle custom data received from the user-specific device

[1] Latest SensiML Interface Specification: https://sensiml.com/documentation/mqtt-specification/introduction.html

[2] MQTT-SN is a lightweight implementation of the popular MQTT messaging protocol tailored specifically for limited resource embedded nodes as found in wireless sensor networks. More detail can be found at http://mqtt.org or by reviewing the specification at http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

[3] JSON: Javascript Object Notation, a data interchange language using a collection of name/value pairs and an ordered list of values to convey data. See https://www.json.org/json-en.html

©2020 SensiML Corporation

3. **Linking and compiling your resulting device inference model** into firmware incorporating the AutoML model from SensiML Analytics Studio with needed user-supplied sensor libraries and your own post-processing application logic

4. **(Optional) Interfacing with SensiML TestApp**, an application used to display and log real-time results from a device containing a completed SensiML Knowledge Pack inference model. Typically, TestApp is used to confirm an ML model built using the SensiML toolkit is performing as expected once flashed to the target embedded device. This step is listed as optional since users may prefer other means for displaying the output of on-device model processing that may include user-specific post-processing beyond the scope of the basic classifier output as shown in SensiML TestApp.

The remaining sections of this Application Note describe where to find the specific documentation and reference code needed to complete each of the above steps.

2 Device-side Data Collection Firmware

SensiML Analytics Toolkit uses AI to build edge sensor inferencing models using labeled train/test data to establish features, weights, and parameters for subsequent modeling in the AutoML engine. For this process to work, it is imperative to generate a sufficient volume of labeled training and test data from devices with behavior and sensor characteristics consistent with the intended predictive device.

While it is always possible with SensiML to collect this train/test data by whatever means possible and then import it as CSV data, often the most convenient and efficient means is to acquire this data directly within the SensiML Data Capture Lab (DCL) application in a device connected capture session. Doing so requires DCL to receive properly formatted data from the device firmware which is then responsible for:

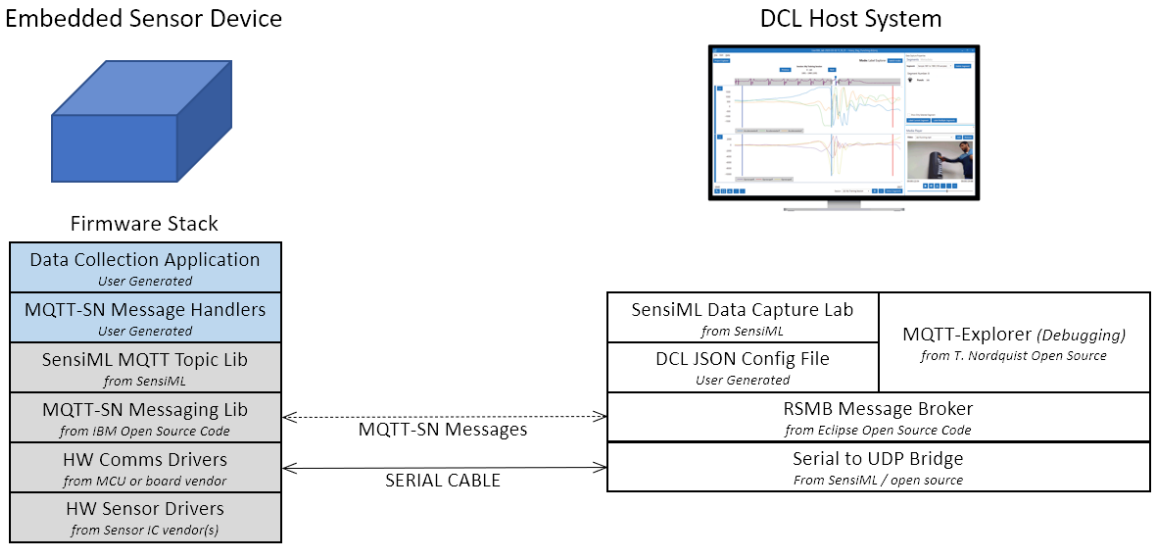
- Initializing and setting all sensor channels to desired gain, offset, calibration, and sample rate. This is done using hardware sensor drivers either written by the user and conforming to datasheets from the sensor IC vendor or using reference code supplied by the sensor IC vendor.
- Setting up recording sessions with the set of sensors to be included in the sample frame.
- Starting and stopping data acquisition from messages received from the controlling host system running the DCL application
- Reading the sensor(s) at the appropriate sample rate and buffering these sample frames into a queue for local flash datafile storage or real-time streaming over MQTT-SN to DCL
- Transmitting buffered livestream or file stored sample data to the host system.

©2020 SensiML Corporation

- Resetting the device and clearing buffers and memory to restart the process for additional capture sessions.

SensiML uses a simplified version of MQTT messaging protocol known as MQTT-SN to send and receive all command, control, and data messages between the host system and the embedded device. The full protocol for communicating with Data Capture Lab can be found in the SensiML Interface Specification[1]. We will not cover the details of the protocol in this Application Note overview, as they are discussed at length in the referenced specification.

The components needed to generate SensiML DCL compatible device firmware are shown in the following illustration highlighting (in blue) the components the user will need to create and/or modify from existing example reference code. These components leverage sourced libraries (in gray) available from either SensiML or your specific sensor and hardware platform vendor(s).



To aid in the development of writing firmware required by the user, existing reference implementations in source code form are available for the supported off-the-shelf dev boards and may be revised as needed by the user. At present writing, such reference implementations include:

- Nordic Thingy52 (an IoT eval kit from Nordic Semi based on their nRF52 MCU)[4]
- QuickLogic EOS S3 "Merced" (an Industrial IoT eval kit from Quicklogic based on their EOSS3 MCU+FPGA SoC)[5]

[4] Reference implementation based on legacy BLE protocol

[5] Recommended reference implementation based on support SensiML Interface Specification (MQTT-SN)

©2020 SensiML Corporation

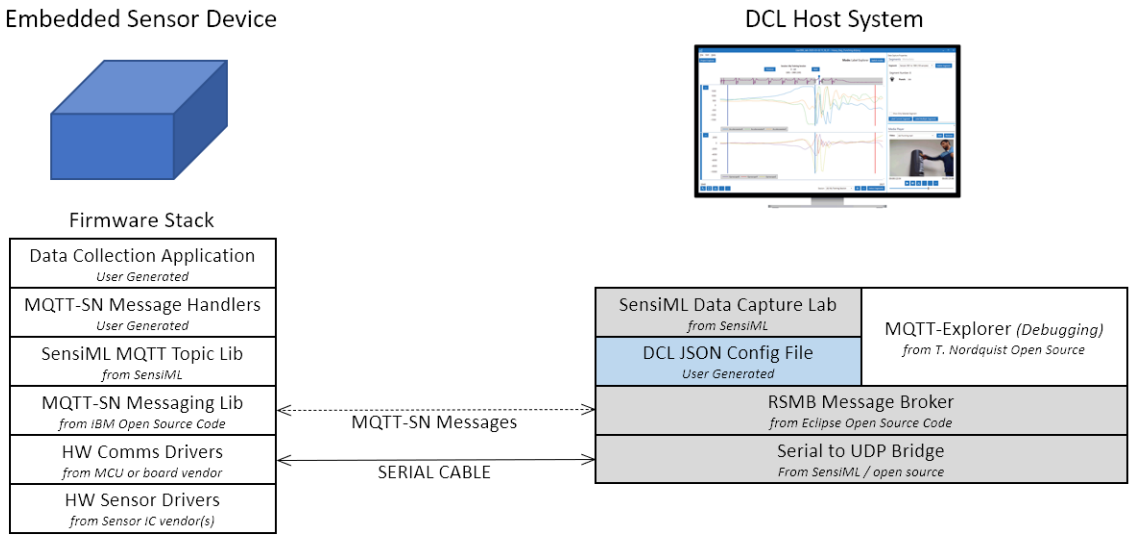
- QuickLogic EOS S3 "Chilkat" (a wearable/consumer IoT eval kit from Quicklogic based on their EOSS3 MCU+FPGA SoC)[4]
- STMicro SensorTilev1 (an IoT eval kit from STMicro based on their STM32L4 MCU)[4]
- STMicro SensorTile.Box (second generation IoT eval kit from STMicro for the STM32L4)[5]

Contact SensiML support for details on how to obtain the reference source code firmware listed above. For full details on generating compatible firmware, refer to the following online documentation:

https://sensiml.com/documentation/mqtt-specification/introduction.html

3 SensiML Data Capture Lab Configuration

Once the firmware has been created and tested to function properly in providing MQTT-SN messages, what remains is to configure DCL to recognize the specific custom hardware profile. This is accomplished by generating a JSON configuration file that can then be imported into DCL adding support for your unique board config.



This file defines sensors, logical groupings, supported sample rates and full scale sensor gains, sensor channel names, communication settings, and various other configurations necessary for DCL to initialize and communicate with the custom device.

Full details for how to configure the JSON file and successfully import your settings can be found at the following online documentation link:

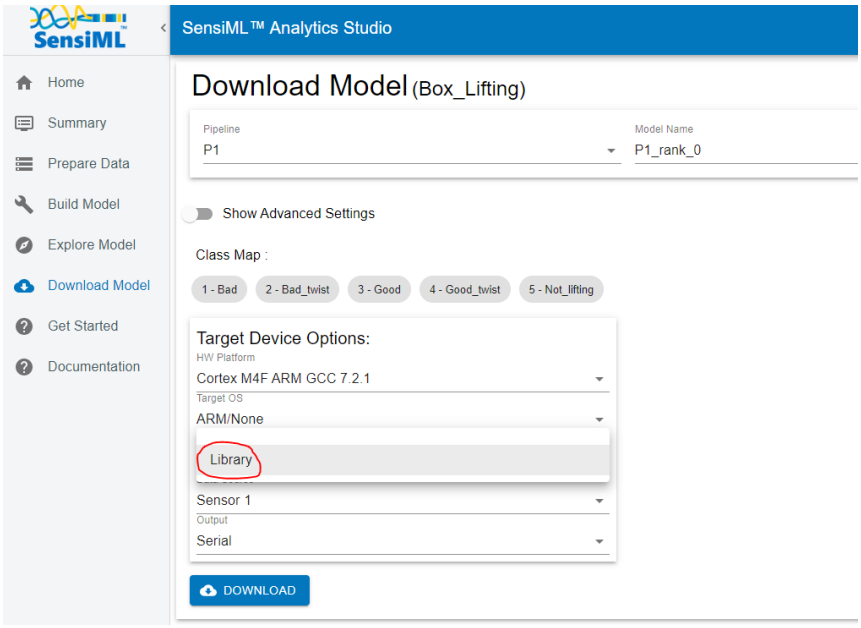
https://sensiml.com/documentation/data-capture-lab/adding-custom-device-firmware.html

©2020 SensiML Corporation

4 Device-side Inference Model Firmware

For known 3rd party hardware eval kits, it is possible to directly link and compile binary firmware ready for flashing to the device without having to utilize separate tools. During algorithm development, this direct-to-binary option is a great time-saving aid to quickly generate working code to test on device.

For customized hardware it becomes necessary to utilize library code output in Analytics Studio rather than binary. Library code output allows you to link and compile with your own firmware library additions as well any/all hardware drivers required from sensor IC vendors as created per steps outlined in section 2.



With library code generated for your particular application, the process for building a flashable binary is covered in the following online documentation link:

https://sensiml.com/documentation/knowledge-packs/building-a-knowledge-pack-library.html

5 Interfacing with SensiML TestApp

SensiML Knowledge Pack binaries built for the supported off-the-shelf IoT hardware evaluation kits include the option to send real-time edge ML model output over BLE. Combined with the available SensiML TestApp, this provides for a convenient and quick means for testing whether a given model is performing as expected on the device.

©2020 SensiML Corporation

Users seeking to continue using this mechanism and TestApp application as they develop for their own boards and custom sensors can do so by implementing BLE GATT characteristics on their device that follow the same methods used by the supported eval kit binary code.

The documentation for how to construct BLE GATT characteristics readable by TestApp can be found here:

https://sensiml.com/documentation/knowledge-packs/sensiml-ble-characteristics.html

Full source code for the Android version of TestApp is available for download here:

https://bitbucket.org/sensimldevteam/sensiml_android_open_project/src/master/

6 Conclusion

By following the steps described above, the ability to adapt the SensiML AutoML pipeline to any edge device implementation that is based on the prevailing ARM and x86 computing architectures supported by the tool is possible. Further, such customized hardware can be extended to cover a broad array of time-series sensor inputs as dictated by the needs of your application.

Should your needs require capabilities not addressed in this application note, or if you require assistance to implement customizations as outlined, refer to https://sensiml.com/support for further assistance or to inquire for additional support from the SensiML technical team.

©2020 SensiML Corporation