User Manual

Getting Started with I-CUBE-HUAWEI
STM32Cube Expansion Package for Huawei IoT platform

Introduction

This user manual describes the content of the STM32Cube software expansion package for the HUAWEI® IoT (Internet of Things) platform.

The I-CUBE-HUAWEI STM32Cube Expansion Package for the HUAWEI IoT platform provides application examples that connect and subscribe to the HUAWEI IoT service via MQTT to receive information and publish data.

I-CUBE-HUAWEI runs on the NUCLEO-L496RG board, offering the following features:

- Ready to run firmware example using cellular connectivity to support quick evaluation and development of device applications connected to the HUAWEI IoT platform.
- Board configuration interface
- Cellular connection
- Connection to the HUAWEI IoT platform
- Firmware update Over-The-Air (FOTA)
- The sensor expansion board X-NUCLEO-IKS01A3 measures and report any one of the following values:
  - Temperature (external)
  - Humidity
  - Pressure
  - 3D Accelerometer data
  - 3D Gyroscope data
  - 3D Magnetometer data

Contents

List of tables

List of figures

1 General information

The I-CUBE-HUAWEI package for the HUAWEI IoT platform runs on STM32 32-bit microcontrollers based on the Arm® Cortex®-M processor.

1.1 Acronyms

Table 1 presents the definition of acronyms that are relevant for a better understanding of this document.

Table 1. List of acronyms

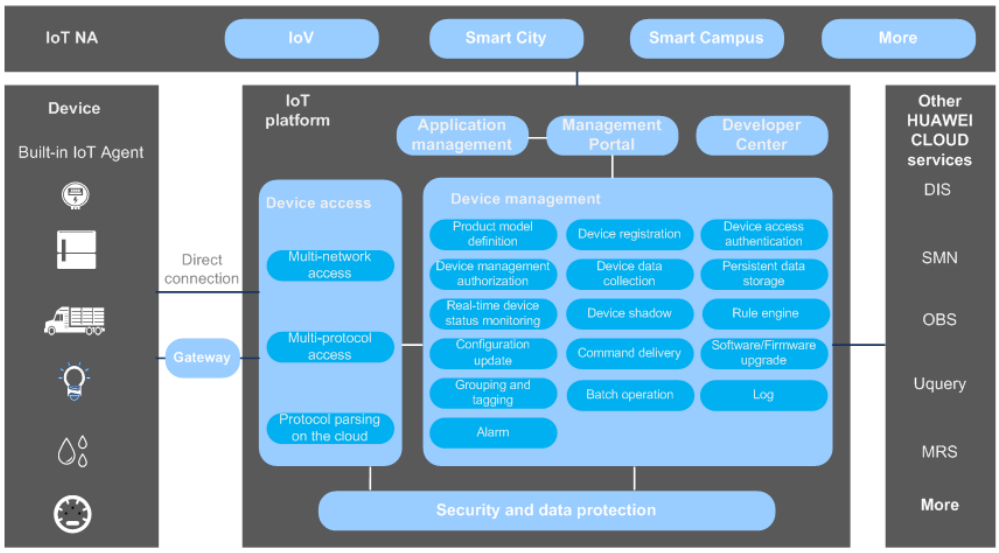
Term	Definition
API	Application programming interface
BSP	Board support package
CA	Certification authority
HAL	Hardware abstraction layer
IDE	Integrated development environment
IoT	Internet of things
IP	Internet protocol
JSON	JavaScript Object Notation
LED	Light-Emitting Diode
SDK	Software Development Kit
RTC	Real-Time Clock
UART	Universal asynchronous receiver/transmitter
Firmware	Firmware refers to the drivers in a device. These are fundamental programs that run at the bottom layer of the operating system.
Product	A product is a collection of devices with the same capabilities or features. In addition to physical devices, a product also includes product information, product models (profile files), codecs, and test reports generated during IoT capability building.
Product Model	A product model (also called profile file) is used to describe the capabilities and features of a device. Developers construct an abstract model of a device by defining a profile file on the IoT platform so that the IoT platform can understand the services, properties, and commands supported by the device.

Codec	The HUAWEI IoT platform communicates with NAS using data in JSON format. Therefore, when a device reports data in binary format, developers need to develop codecs on Developer Center to help the IoT platform convert data into different formats.
Device	A Device is a physical entity that belongs to a product. Each device has a unique ID. It can be a device directly connected to the IoT platform, or a gateway for sub devices to connect to the IoT platform.
Gateway	A Gateway is a physical entity that manages sub-devices and connects sub-devices to the IoT platform.
Sub-Device	A Sub-Device is a physical entity that connects to the IoT platform through a gateway.
Rule	A Rule is a preset condition used by the IoT platform to trigger actions. The device will report device data, which is checked against the rules. When a rule condition is met, the IoT platform will trigger corresponding actions such as delivering a command to the device or forwarding data to other HUAWEI CLOUD services for integration and utilization.
MQTT	MQTT is an IoT transmission protocol designed for lightweight release/subscription message transmission. It aims to provide reliable network services for IoT devices in low-bandwidth and unstable network environments. MQTTS refers to the combination of MQTT and SSL/TLS. The SSL and TLS protocols are used for encrypted transmission.
Constrained Application Protocol (CoAP)	CoAP is a software protocol designed to enable simple devices to perform interactive communication on the Internet. CoAPS refers to CoAP over DTLS. The DTLS protocol is used for encrypted transmission.
Lightweight Machine to Machine (LWM2M)	LWM2M is an IoT protocol defined by the Open Mobile Alliance (OMA). It mainly applies to NB-IoT devices with limited resources (such as limited storage and power supply).

2 HUAWEI IoT platform

The HUAWEI IoT platform connects to and manages a large number of devices. It provides services such as device access, device management, Security and Data Protection, Application Management, Management Portal, and Developer Center, among others.
The IoT platform works with other HUAWEI CLOUD services to quickly build IoT applications.

Figure 1. HUAWEI IoT ecosystem



3 Hardware and software environment setup

To set up the hardware and software environment, the supported board must be plugged into a personal computer via a USB cable. This connection with the PC allows the user to:

- Supply power for the NUCLEO-L496ZG and L716-CN board
- Flash the board
- Interact with the board via a UART console
- Debug

3.1 Hardware setup

Figure 2. NUCLEO-L496ZG development board

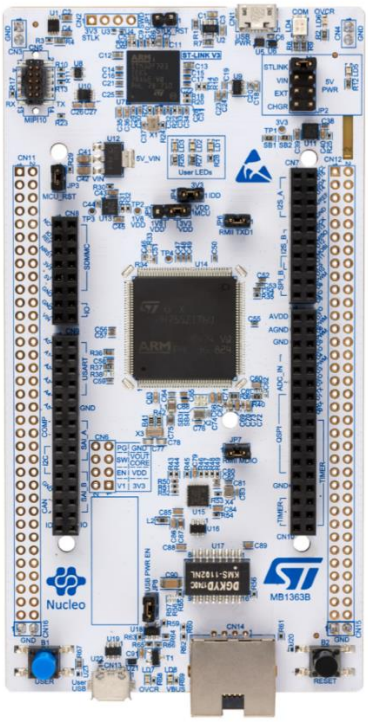


Figure 3. X-NUCLEO-IKS01A3 sensor board

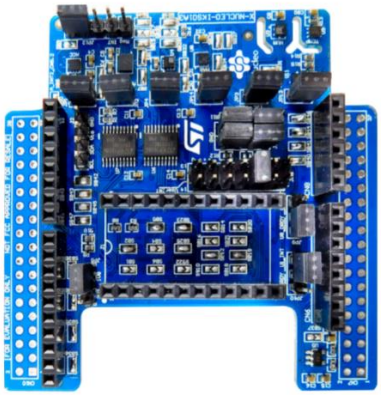
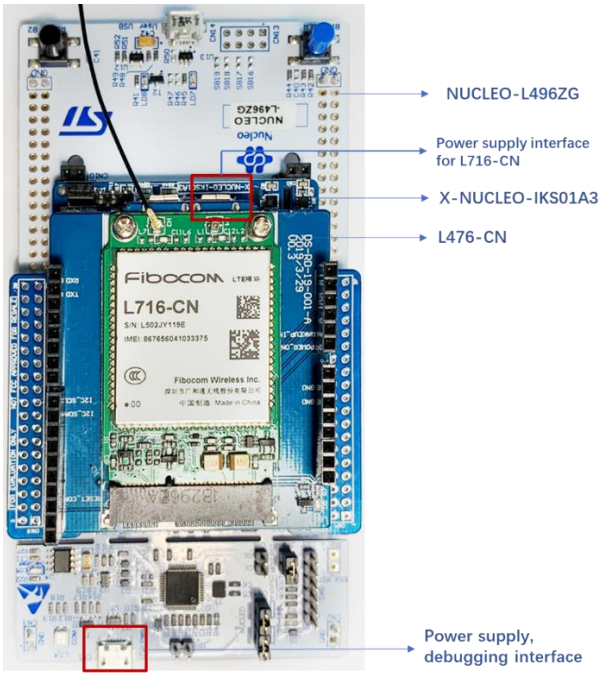


Figure 4. Fibocom L716-CN cellular board

Figure 5. Hardware setup



Important Note:

- *Need to insert a SIM Card in the L716-CN for cellular connection*
- *The power supply for L716-CN is a* ***MUST***

4 Package description

This section details the I-CUBE-HUAWEI package content and how to use it.

4.1 General description

The I-CUBE-HUAWEI package provides a HUAWEI stack middleware for the STM32 microcontrollers. The package is split into the following components:

- HUAWEI LiteOS for connecting to HUAWEI IoT platform from a device using embedded C
- STM32L4 Series HAL
- Cellular drivers for the Fibocom L716-CN board
- Sensor drivers for the X-NUCLEO-IKS01A3 board
- HUAWEI application examples

The software is provided as a zip archive containing source code.
The following integrated development environments are supported:

- SM32CubeIDE Version 1.3.0 or higher must be used
- Keil® Microcontroller Development Kit (MDK-ARM) Version 5.26 or higher must be used

Note: Refer to the release note available in the package root folder for information about the IDE versions supported.

4.2 Architecture

This section describes the software components of the I-CUBE-HUAWEI package.
The I-CUBE-HUAWEI software is an expansion for the STM32Cube. Its main features and characteristics are:

- Fully compliant with STM32Cube architecture
- Expands STM32Cube to enable the development of applications accessing and using the HUAWEI IoT platform
- Based on the STM32Cube HAL, which is the hardware abstraction layer for STM32 microcontrollers

The software components used by the application software to access and use the HUAWEI IoT platform are the following:

- STM32Cube HAL layer
  The HAL driver layer provides a generic multi-instance simple set of APIs (Application Programming Interfaces) to interact with the upper layers (application, libraries, and stacks). It is composed of generic and extension APIs. It is directly built around a generic architecture and allows the layers that are built upon, such as the middleware layer, to implement their functionalities without dependencies on the specific hardware configuration for a given microcontroller unit (MCU).
  This structure improves the library code reusability and guarantees easy portability onto other devices.

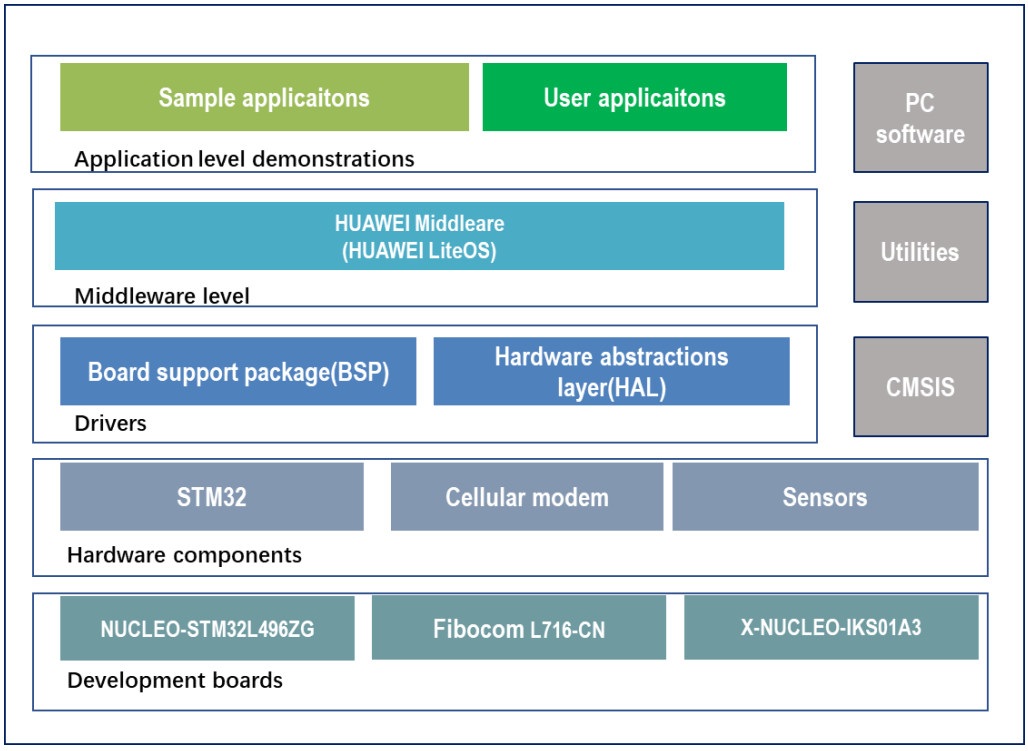
- Board Support Package (BSP) layer

The software package needs to support the peripherals on the STM32 boards apart from the MCU. This software is included in the board support package (BSP). This is a limited set of APIs which provides a programming interface for certain board-specific peripherals such as the LED and the user button.

- HUAWEI Middleware
  The middleware is tailored from HUAWEI LiteOS, which consists of HUAWEI device-cloud interconnect components. Device-cloud interconnects components enable device-cloud synergy and integrate a full set of IoT interconnection protocol stacks, such as Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), mbed TLS, and lightweight IP (LwIP).

Figure 6 outlines I-CUBE-HUAWEI software architecture.

Figure 6. I-CUBE-HUAWEI software architecture

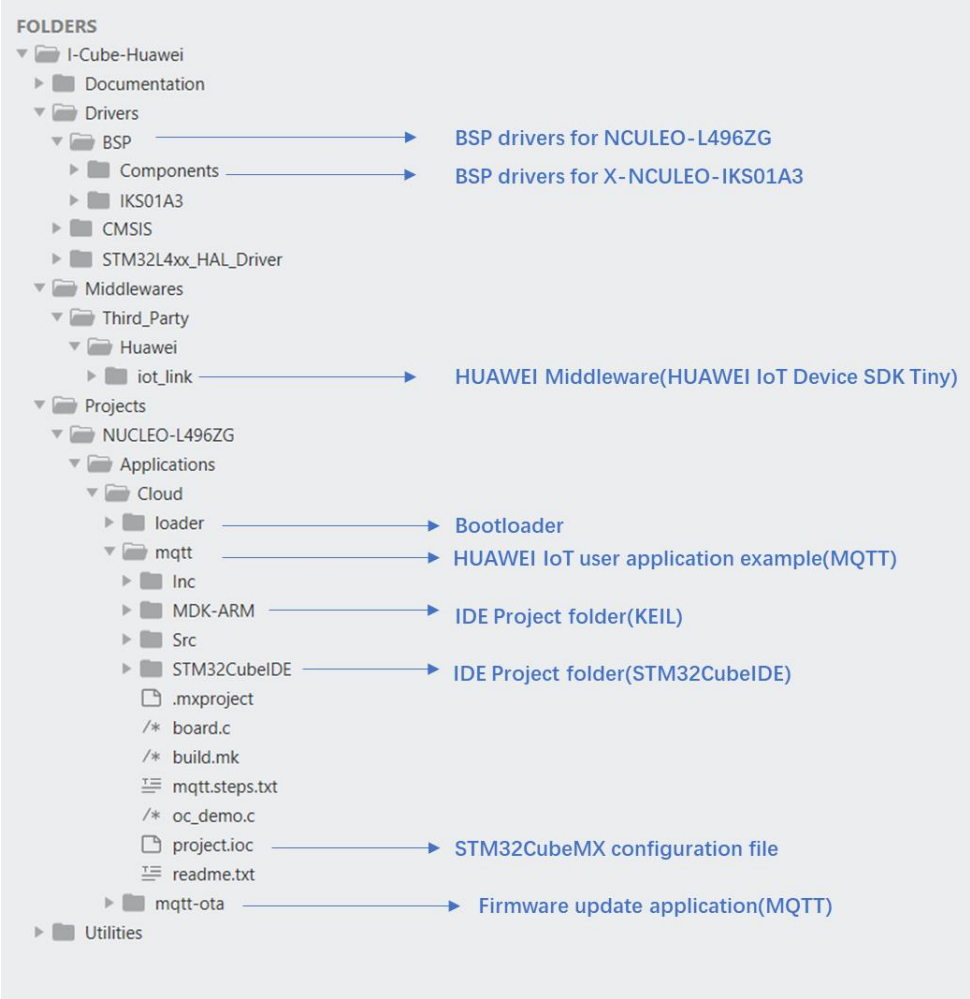


Note: Currently, the Cellular module driver (Fibocom L716-CN) is included in the HUAWEI middleware.

4.3 Folder structure

Figure 7 presents the folder structure of the I-CUBE-HUAWEI package.

Figure 7. Project Folder Structure



4.4 Application examples

This section describes how to use the application example provided at Projects\NUCLEO-L496ZG\Applications\Cloud\.

4.4.1 MQTT application example

This is an example using HUAWEI IoT service to report humidity, temperature, atmospheric pressure, 3D accelerations data. The sensor data are reported in the JSON format every 10 seconds.

4.4.1.1 Product development on HUAWEI IoT platform

When using the HUAWEI IoT platform cloud service for the first time, you must subscribe to the Developer Center. You can use the Developer Center to develop product models and codecs online.

Step 1: Register a HUAWEI CLOUD account on the [registration page](#).

Step 2: Log in to HUAWEI CLOUD and visit IoT Device Management.

Step 3: Product Creation

The product is a collection of devices with the same capabilities or features. In addition to physical devices, a product includes product information, product models (profile files), codecs, and test reports generated during IoT capability building.

To create a product, click **Create product** on the home page. In the dialog box displayed, enter relevant information and click **Create**.

Figure 8. Create a product

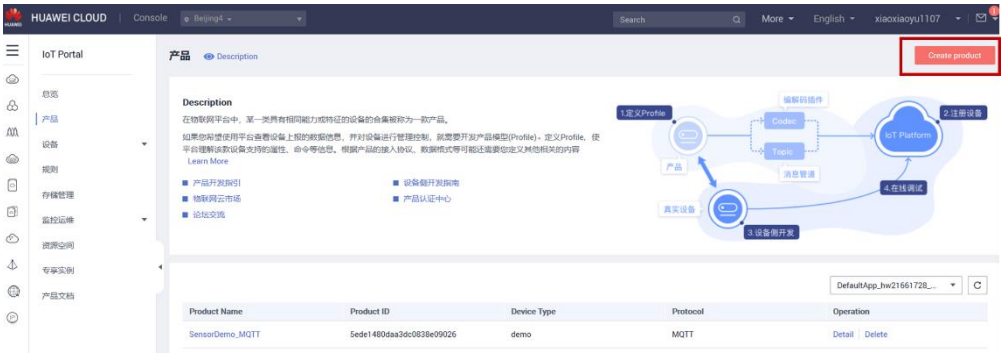
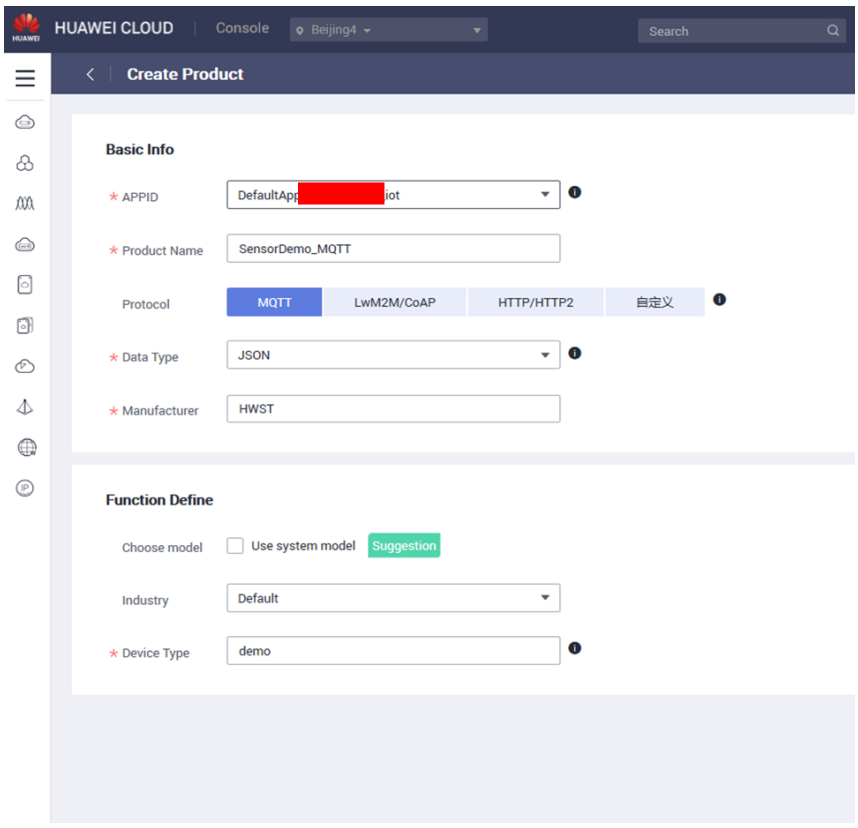


Figure 9. Set product information

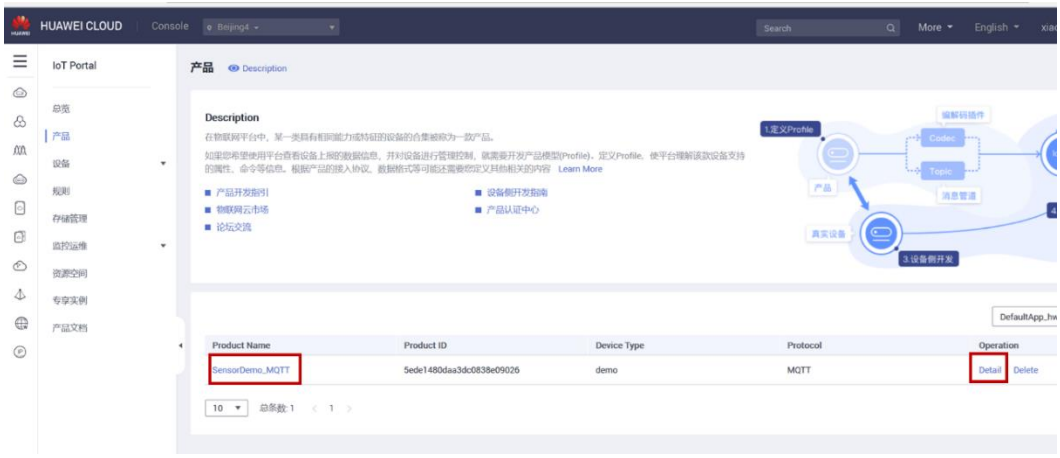


Step 4: Profile definition

A product model (also called a profile) is used to describe the capabilities and features of a device. Developers construct an abstract model of a device by defining a profile file on the IoT platform so that the IoT platform can understand the services, properties, and commands supported by the device.

On the Product Development page, select the desired product, click **Detail** on the right. The Product Details page is displayed.

Figure 10. Product details



- In the development space, click **Customize functions** and click **Add Service**.

Figure 11. Customize product functions

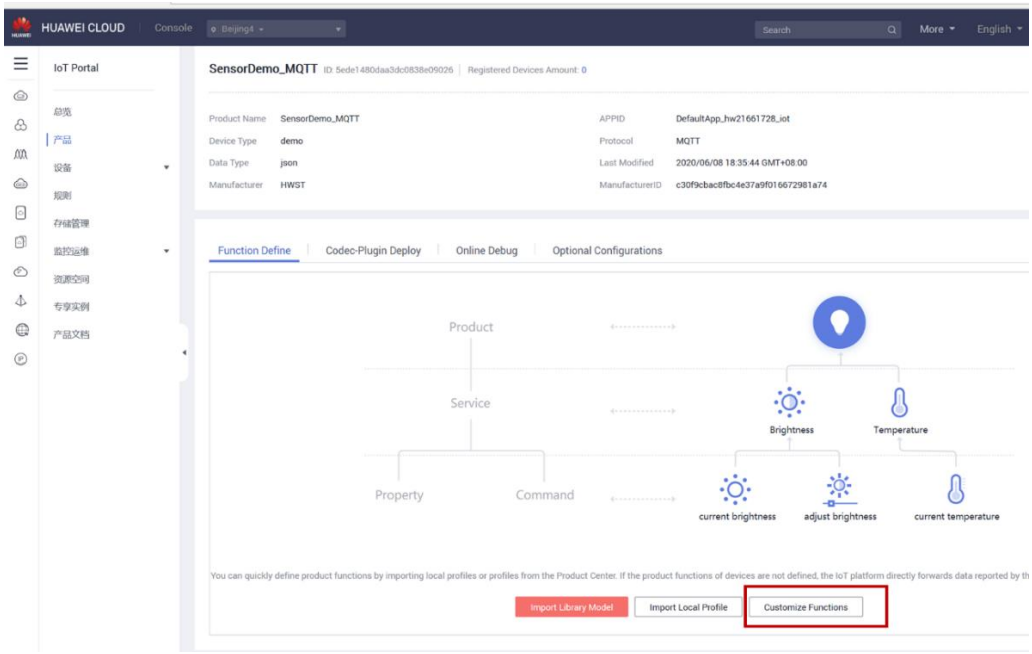
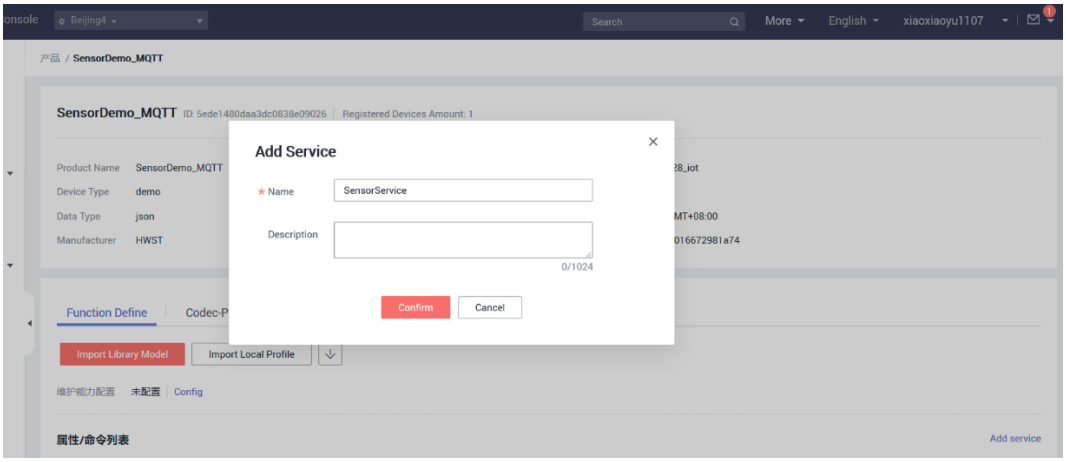


Figure 12. Add service



- In the **Add Service** area, define the service name, properties, and commands. A service can contain properties and/or commands. Configure the properties and commands based on your requirements.

Figure 13. Add temperature property

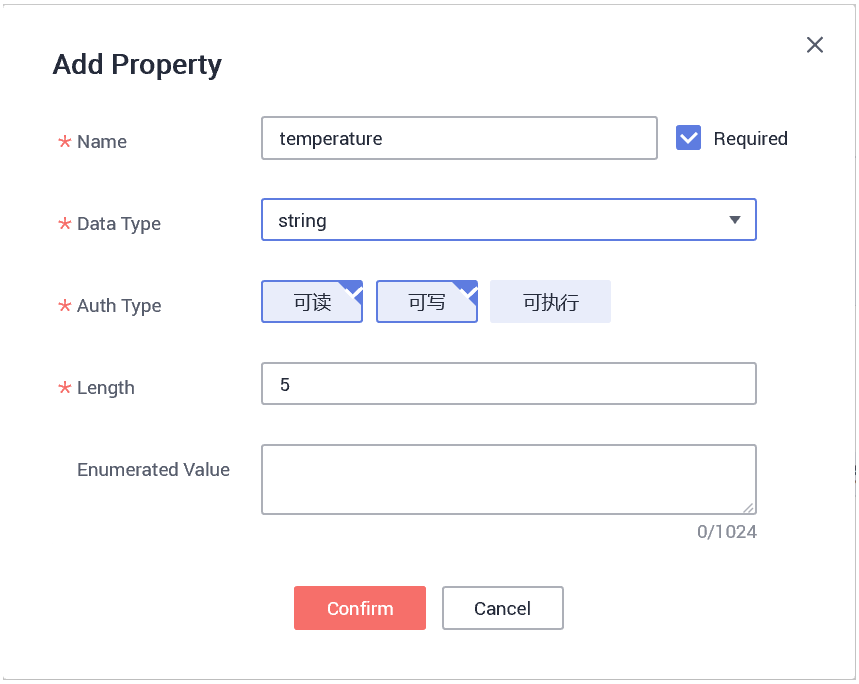


Figure 14. Add humidity property

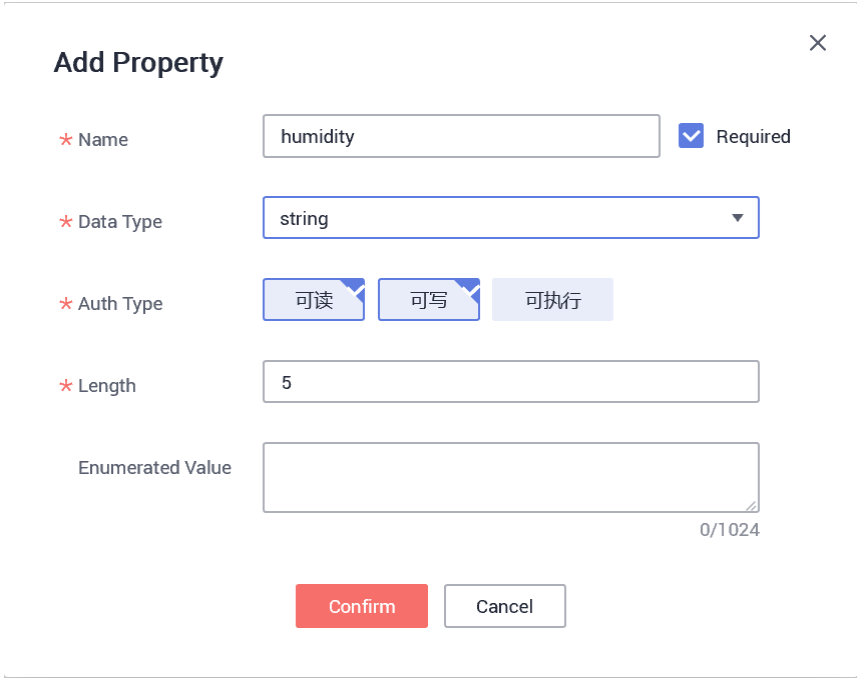


Figure 15. Add pressure property

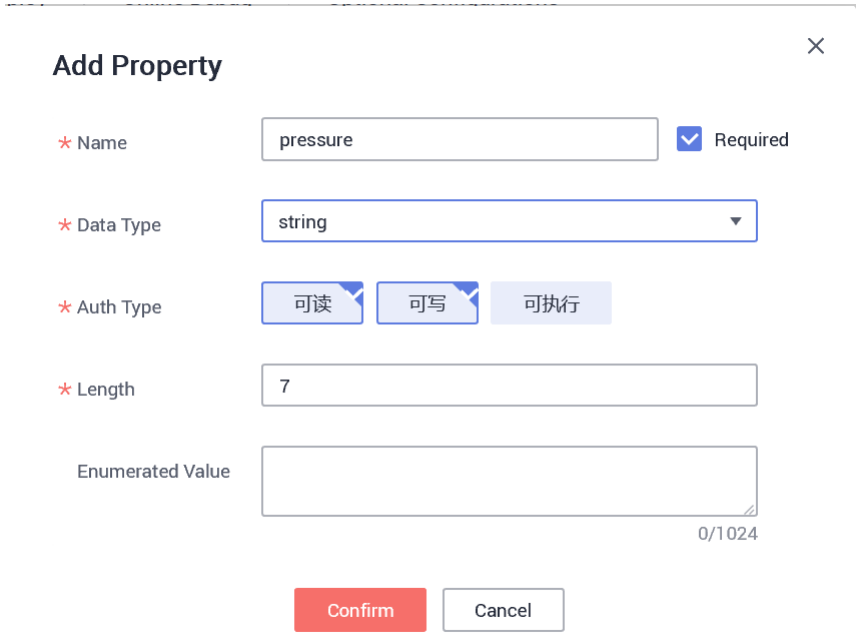


Figure 16. Add accelerometer_x property

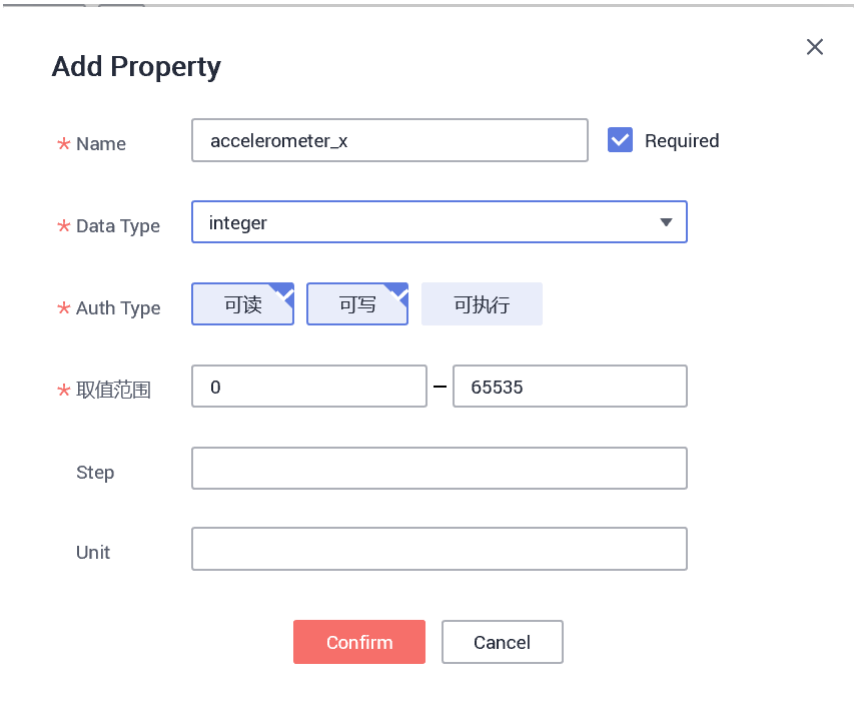


Figure 17. Add accelerometer_y property

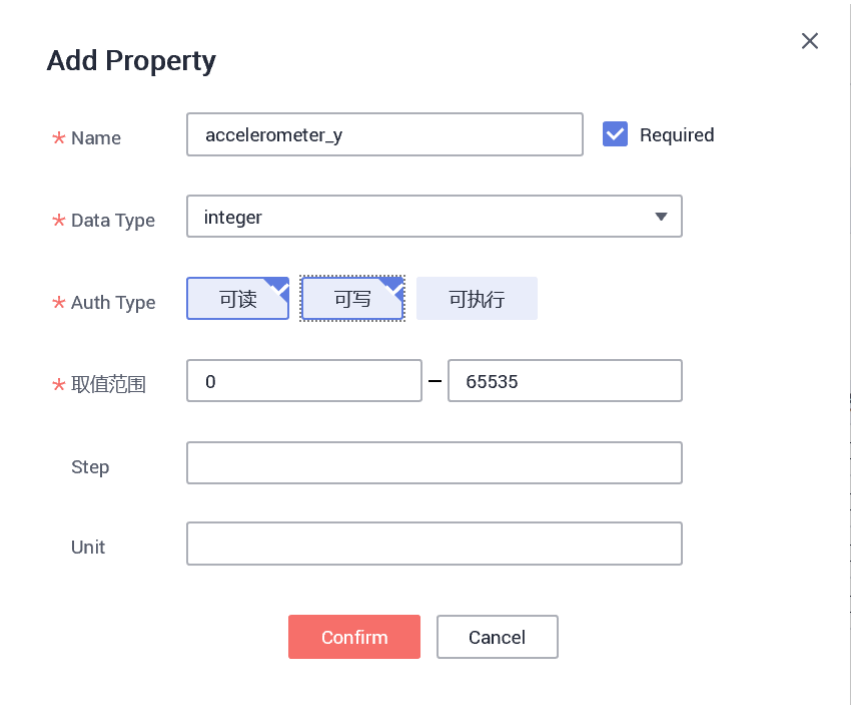


Figure 18. Add accelerometer_z property

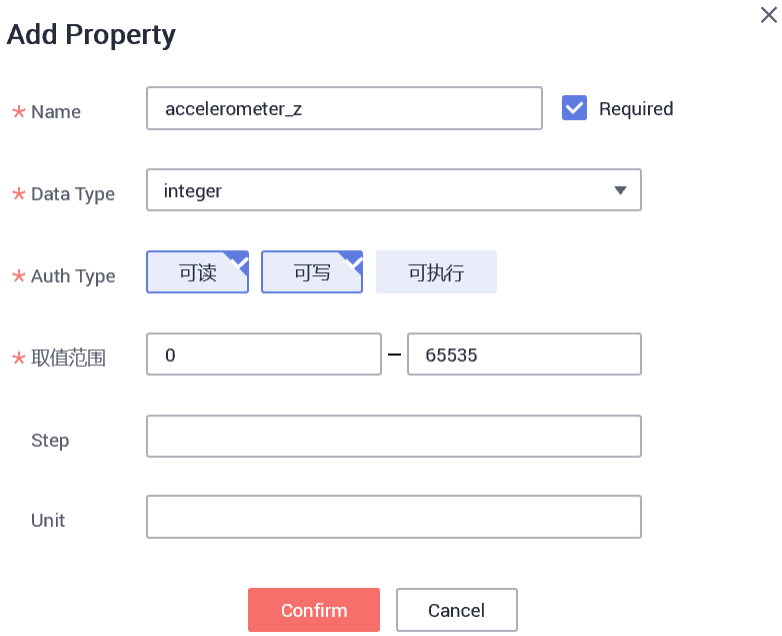
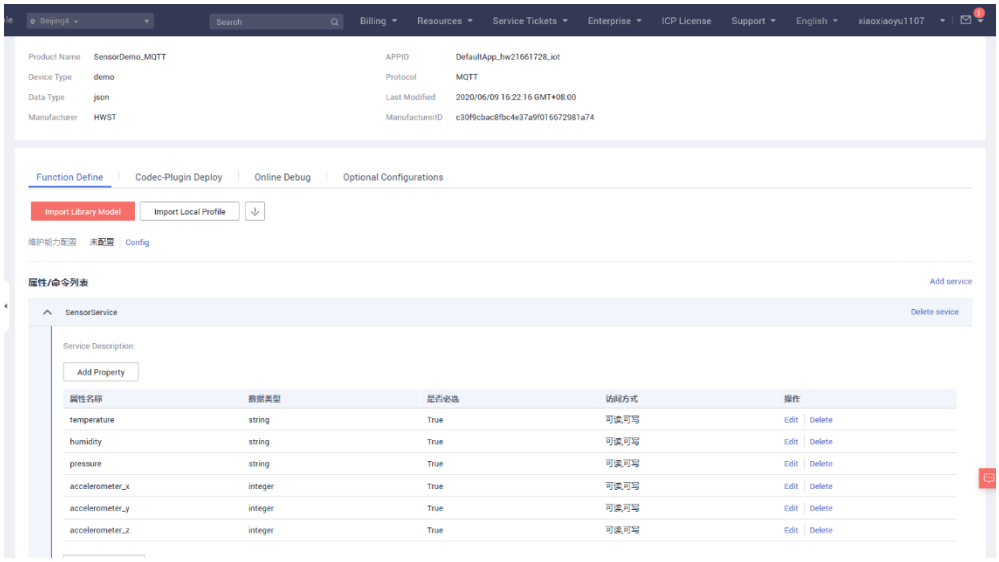


Figure 19. Property list

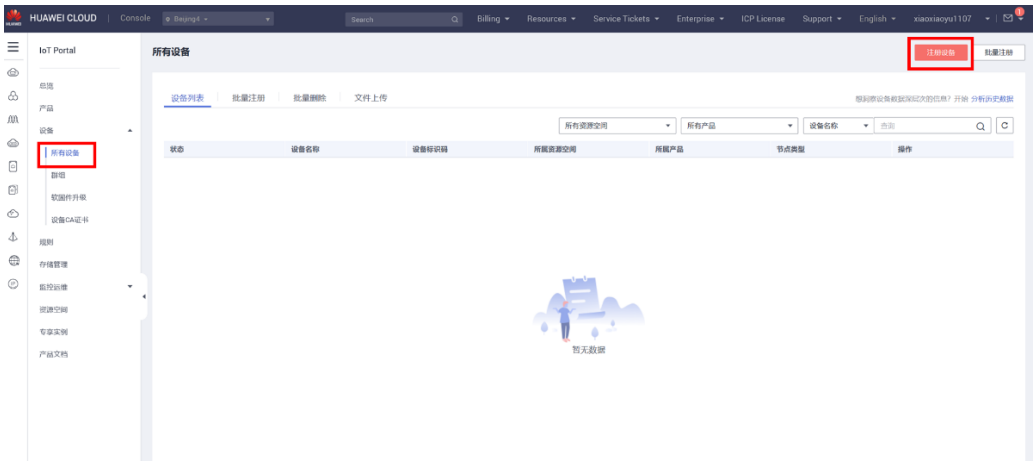


STEP 5: Registering a Device

Register a device on the IoT platform and define device parameters. Then the device can connect to the IoT platform if authentication succeeds.

- Choose Devices > Registration.

Figure 20. Register a device



In the dialog box displayed, set the parameters, and click **Confirm**.

Figure 21. Set device registration parameters

After the device is added, **Device ID** and **PSK** are returned. Keep the PSK securely as it is required when the device uses DTLS to connect to the IoT platform.

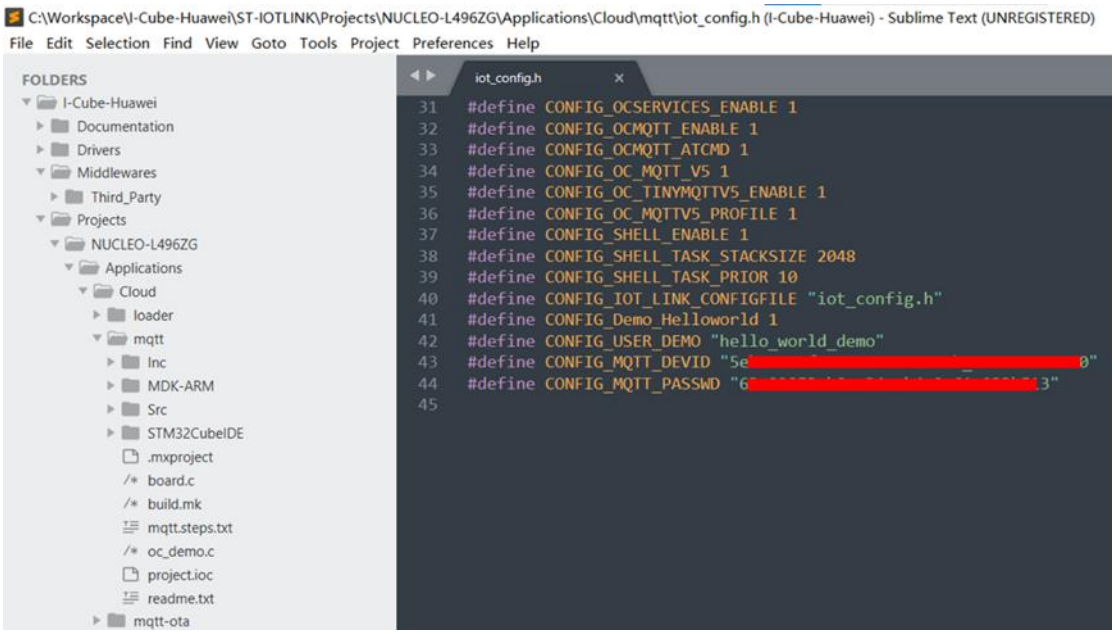
Figure 22. Device ID and PSK

4.4.1.2 Application build and first launch

Caution: Before opening the project with any toolchain, **make sure that the folder installation path is not too deep,** otherwise the toolchain may report errors after the build.

Update the device information to the corresponding macro definition in Projects\<board name>\ Applications\Cloud\mqtt\iot_config.h**.**

Figure 23. Device provisioning



Open and build the project with one of the supported development toolchains (see the release note for detailed information about the version requirements).

Program the firmware on the STM32 board: you can copy (or drag and drop) the generated bin file to the USB mass storage location created when you plug the STM32 board to your PC.

Alternatively, you can program the STM32 board directly through one of the supported development toolchains.

4.4.1.3 Interacting with the boards

A serial terminal is required to:

- Configure the board
- Display locally the sent/received HUAWEI IoT cloud device-to-cloud/cloud-to-device messages

The example in this document is illustrated with the use of Tera Term. Any other similar tool can be used instead.

- Determine the STM32 ST-LINK Virtual COM port used on the PC for the Discovery board.

On a Windows® PC, open the Device Manager.

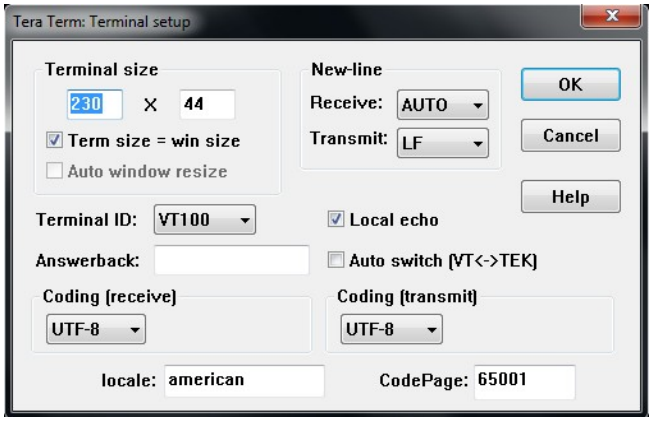
- Open a virtual terminal on the PC and connect it to the above virtual COM port.

Note: The information provided below in this chapter can be used to configure the UART terminal as an alternative to using the Tera Term initialization script.

Terminal setup is illustrated in *Figure 24*, which shows the terminal setup and the New-line recommended parameters.

The virtual terminal New-line transmit configuration must be set to Linefeed (\n or LF) to allow copy-paste from UNIX type text files. The Local echo option makes copy-paste visible on the console.

Figure 24. Terminal Setup

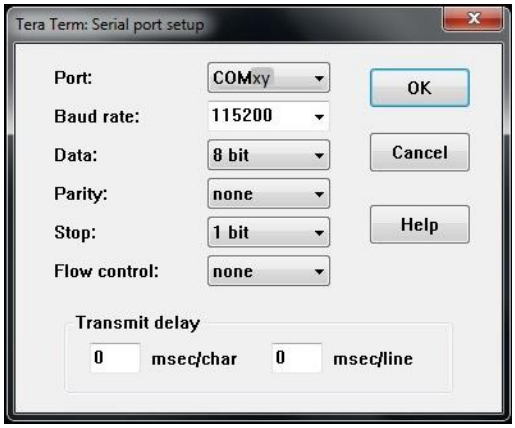


The serial port must be configured with:

- COM port number
- 115200 baud rate
- 8-bit data
- Parity none
- 1 stop bit
- No flow control

The serial port setup is illustrated in *Figure 25*.

Figure 25. Serial port setup



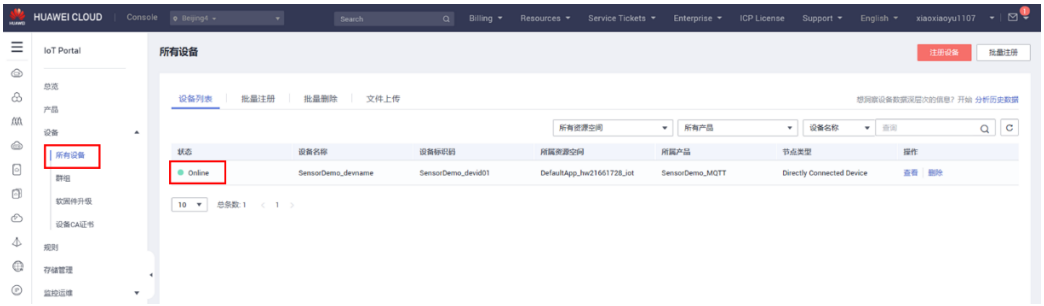
Once the UART terminal and the serial port are set up, press the board reset button (black). Follow the indications on the UART terminal to upload Wi-Fi® provisioning data. Those data remain in Flash and are reused the next time.

4.4.1.4 Connecting a Device

Connect a physical device to the IoT platform to verify that the device can report data to the IoT platform and display the data on the Management Portal.

- Log in to the Management Portal. Choose Device Management > Devices > Device List, and check the device status on the device list. If the status is **Online**, the device has been connected to the IoT platform.

Figure 26. Device list and status



- Click the device. On the details page, view the latest reported data. If the data can be properly parsed and displayed, the device reports data successfully.

Figure 27. Report sensor data

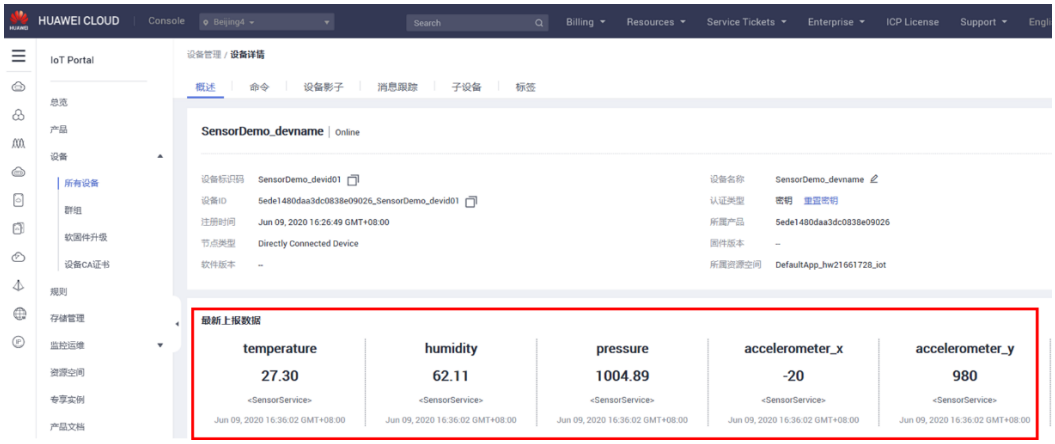
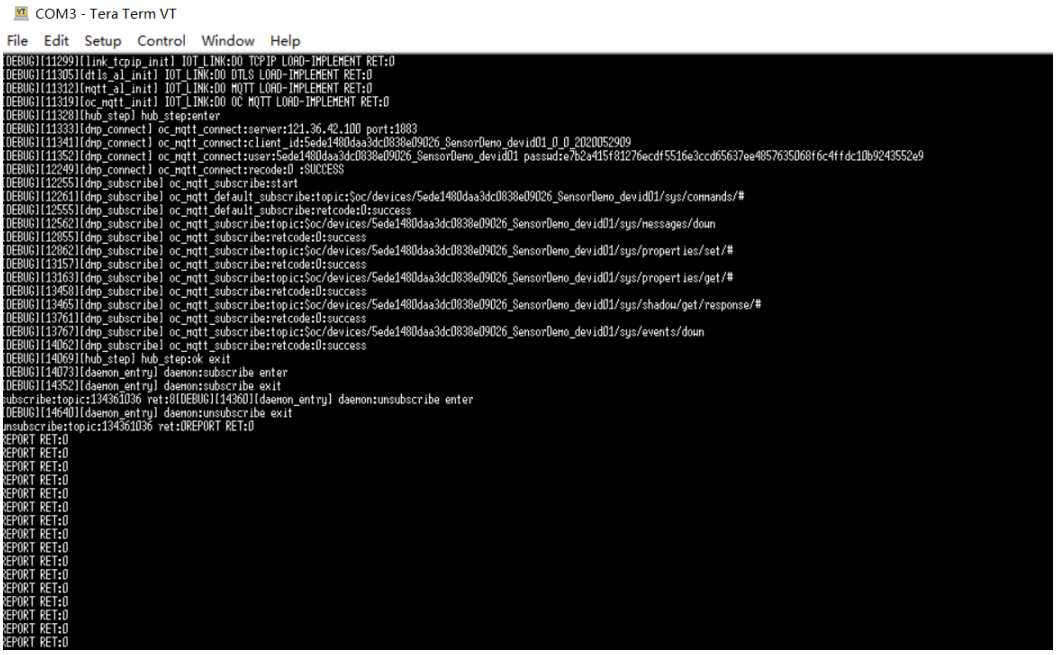


Figure 28. Log on console



5 Revision history

Table 2. Document revision history

Date	Revision	Changes
15-JUNE-2020	1.0	Initial release.