User Manual

Getting Started with I-CUBE-ALIYUN IoT Cloud Software expansion for STM32Cube

Introduction

This user manual describes the content of the STM32Cube software expansion package for the Aliyun IoT platform.

The Aliyun IoT cloud software expansion package (I-CUBE-ALIYUN) for STM32Cube provides application examples that connect STMicroelectronics boards to the Aliyun IoT platform. It uses the Aliyun linkkit-embedded SDK which is compiled and running on the STM32 device.

I-CUBE-ALIYUN runs on the NUCLEO-L476RG board.

Implementation examples are included for device-to-cloud telemetry reporting, cloud-to-device messages for notifications to the connected devices.

I-CUBE-ALIYUN offers the following features:

- Ready to run firmware example using Wi-Fi® to support quick evaluation and development of device applications connect to Aliyun IoT platform.
- Board configuration interface
- Wi-Fi® connection
- Connection to the Aliyun IoT platform
- The sensor expansion board X-NUCLEO-IKS01A2 measures and report any one of the following values:
  - Temperature
  - Humidity
  - Pressure
  - 3D Accelerometer data
  - 3D Gyroscope data
  - 3D Magnetometer data
  - STM32 Temperature data

Contents

1 General information 5

2 Aliyun IoT platform 6

3 Package description 7

3.1 General description 7

3.2 Architecture 9

3.3 Folder structure 11

3.4 X-NUCLEO-IKS01A2 board sensors 11

3.5 EXT-EMW3080 Wi-Fi® expansion board 12

3.6 Reset push-button (black) 12

3.7 User push-button (blue) 12

3.8 User LED 12

4 Hardware and software environment setup 13

5 Interacting with the boards 14

6 Application example 16

6.1 Application description 16

6.2 Aliyun IoT account creation 16

6.3 Device creation on the Aliyun IoT platform 16

6.4 Application build and flash 17

6.5 Application first launch 18

6.6 Application runtime 19

7 Frequently asked questions 22

8 Revision history 23

List of tables

Table 1. List of acronyms 5
Table 2. Document revision history 23

List of figures

Figure 1. Aliyun IoT cloud ecosystem 6
Figure 2. NUCLEO-L476RG board 7
Figure 3. X-NUCLEO-IKS01A2 board 7
Figure 4. I-CUBE-ALIYUN software architecture 10
Figure 5. Project file structure 11
Figure 6. Hardware and software setup environment 13
Figure 7. Terminal Setup 14
Figure 8. Serial port setup 15
Figure 9. Aliyun IoT account creation 16
Figure 10. Device creation 17
Figure 11. Device provisioning 17
Figure 12. Device topic update 18
Figure 13. Complete assembled board 18
Figure 14. WiFi provisioning 19
Figure 15. WiFi credential update 19
Figure 16. Device connect to Aliyun IoT platform 19
Figure 17. Device to cloud message 20
Figure 18. Aliyun IoT platform console – device status 20
Figure 19. Aliyun IoT platform console – upstream data 21
Figure 20. Aliyun IoT platform console – downstream data 21
Figure 21. Pop-up when the IAR™ IDE version is not compatible with the one used for I-CUBE-ALIYUN 22

1 General information

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

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
DHCP	Dynamic host configuration protocol
DNS	Domain name server
HAL	Hardware abstraction layer
IDE	Integrated development environment
IoT	Internet of things
IP	Internet protocol
LED	Light-emitting diode
SDK	Software development kit
RTC	Real-time clock
UART	Universal asynchronous receiver/transmitter

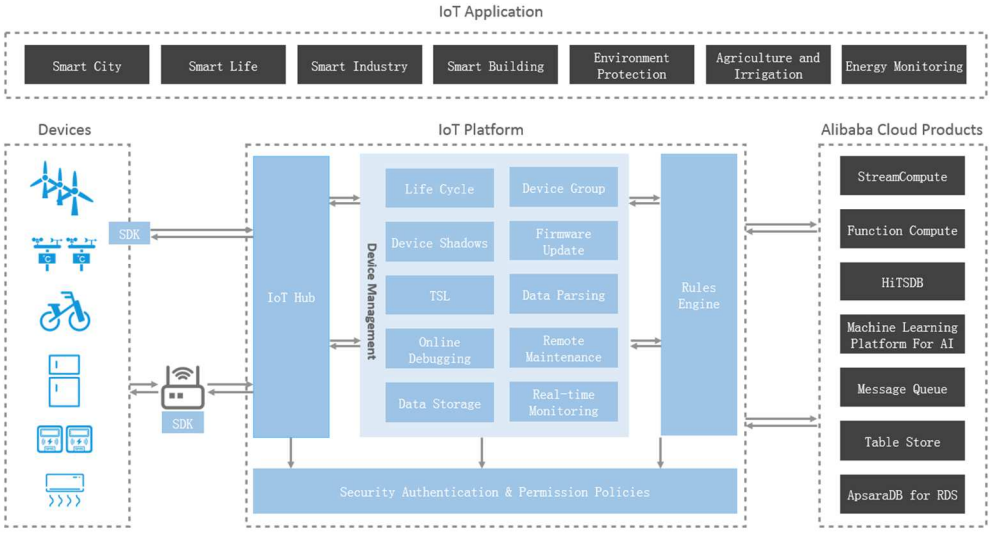
2 Aliyun IoT platform

This chapter introduces the Aliyun IoT platform.
The I-CUBE-ALIYUN package implements the Aliyun linkkit-embedded SDK which allows the board to securely connect to the Aliyun IoT platform.

A user can connect to the cloud with a smartphone or personal computer and have access to the information provided by the board at any time and from any location.

Figure 1 presents the Aliyun IoT cloud ecosystem targeted by the I-CUBE-ALIYUN package.

Figure 1. Aliyun IoT cloud ecosystem



3 Package description

This chapter details the I-CUBE-ALIYUN package content and the way to use it.

3.1 General description

The I-CUBE-ALIYUN package provides an Aliyun linkkit-embedded stack middleware for STM32 microcontrollers.

It is built to run on the NUCLEO-L476RG board and connects to the Internet through the EXT-EMW3080 Wi-Fi® network interface:

- EXT-EMW3080 supports Wi-Fi® connectivity with an on-board Mxchip® WiFi module.

Figure 2. NUCLEO-L476RG board

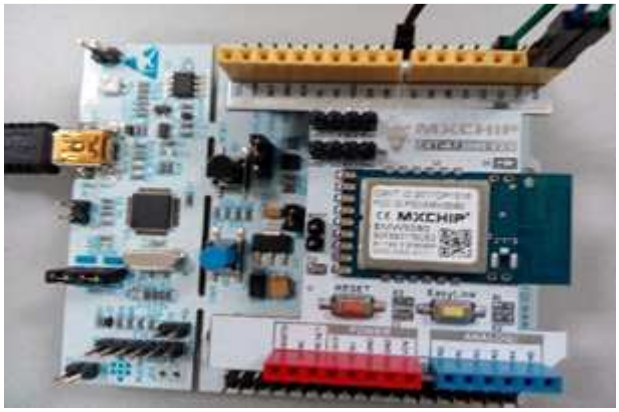
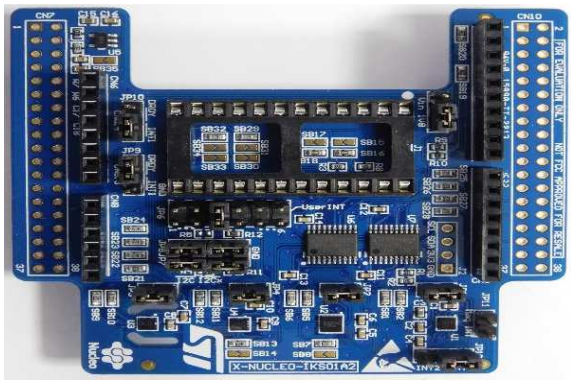


Figure 3. X-NUCLEO-IKS01A2 board



The package is split into the following software components:

- linkkit-embedded SDK for connecting to Aliyun IoT platform.
- Wi-Fi® drivers
- Sensor drivers for the X-NUCLEO-IKS01A2 board
- STM32L4 Series HAL
- Aliyun application examples

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

- IAR Embedded Workbench® for Arm® (EWARM)
- Keil® Microcontroller Development Kit (MDK-ARM)
- System Workbench for STM32

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

3.2 Architecture

This section describes the software components of the I-CUBE-ALIYUN package.

The I-CUBE-ALIYUN software is an expansion for the STM32Cube. Its main features and characteristics are:
- Fully compliant with STM32Cube architecture
- Expands STM32Cube in order to enable the development of applications accessing and using the Aliyun IoT cloud platform
- Based on the STM32CubeHAL, which is the hardware abstraction layer for STM32 microcontrollers

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

1. STM32Cube HAL

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 an easy portability onto other devices.

2. Board support package (BSP)

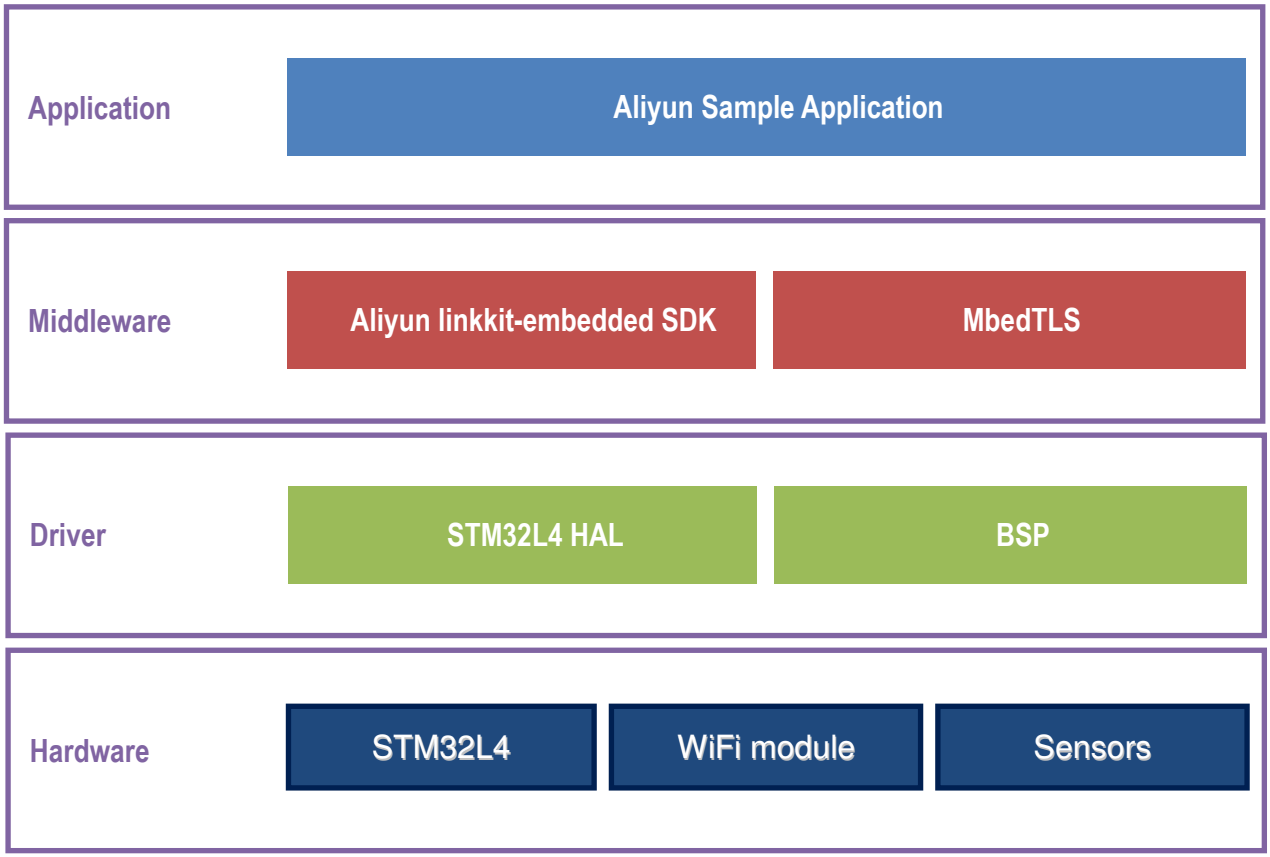
The software package needs to support the peripherals on the STM32 boards apart from the MCU. This software is included in the board support package (BSP). This is a limited set of APIs which provides a programming interface for certain board specific peripherals such as the LED and the user button.

3. Aliyun linkkit-embedded software development kit (SDK)

4. MbedTLS

5. *Figure 4* outlines I-CUBE-ALIYUN software architecture.

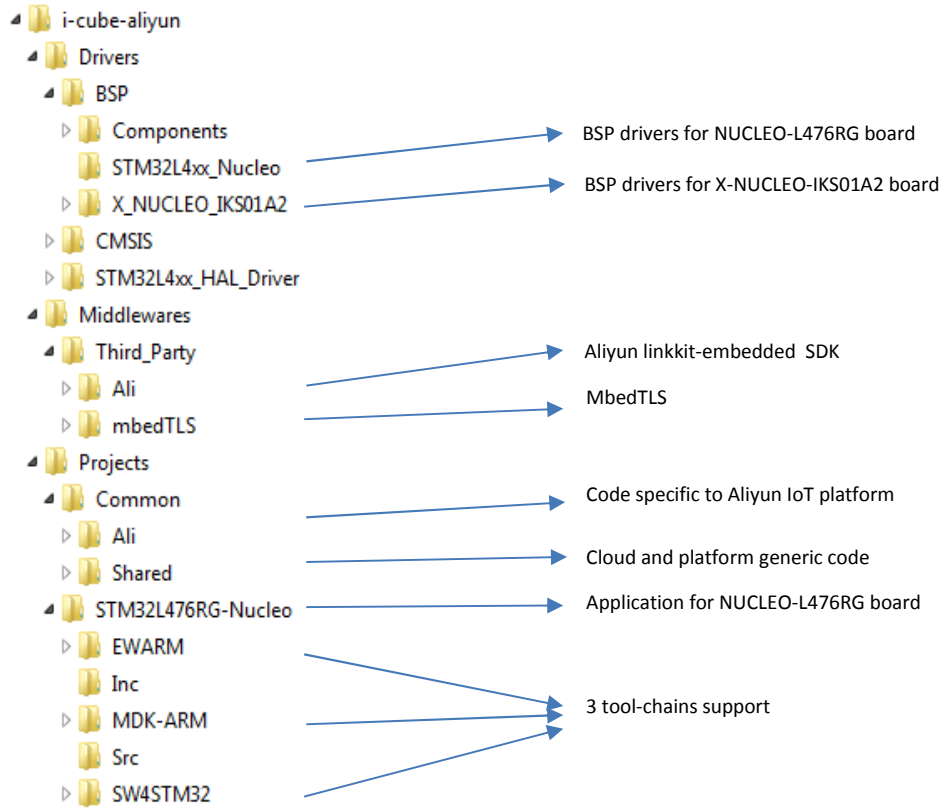
Figure 4. I-CUBE-ALIYUN software architecture



3.3 Folder structure

Figure 5 presents the folder structure of the I-CUBE-ALIYUN package.

Figure 5. Project file structure



3.4 X-NUCLEO-IKS01A2 board sensors

The sensors that are present on the board that can be used by the sample application are:

- Capacitive digital sensor for relative humidity and temperature (HTS221)
- High-performance 3-axis magnetometer (LIS303AGR)
- 3D accelerometer and 3D gyroscope (LSM6DSL)
- 260-1260 hPa absolute digital output barometer (LPS22HB)

For the application example, the internal temperature sensor data of STM32L is also sent to the cloud.

3.5 EXT-EMW3080 Wi-Fi® expansion board

The Wi-Fi® software is split over Wi-Fi abstraction, module specific high level API and module specific low level I/O operation.

The Mxchip Wi-Fi module AT firmware version is “basic_at_v2.0.9”.

3.6 Reset push-button (black)

The reset push-button (black) is used to reset the board at any time. This action makes the board reboot.

3.7 User push-button (blue)

The user push-button (blue) is used in the following cases:

- To configure the Wi-Fi® access point credentials. This can be done from the time the board starts up and up to two seconds after that.

The application configures and manages the user button via the board support package (BSP) functions. The BSP functions are in the Drivers\BSP\<board name> directory.

When using the BSP button functions with the BUTTON_USER value, the application does not take into account the way this button is connected from a hardware standpoint for a given platform. The mapping is handled by the BSP.

3.8 User LED

The configuration of the user LED that is used by the applications is done via the board support package (BSP) functions.

The BSP functions are under the Drivers\BSP\<board name> directory.

Using the BSP button functions with the LED_GREEN value, the application does not take into account the way the LED is mapped for a given platform. The mapping is handled by the BSP.

The behavior On/Off of user LED has been selected to be used to indicate the desired LED state during application run time.

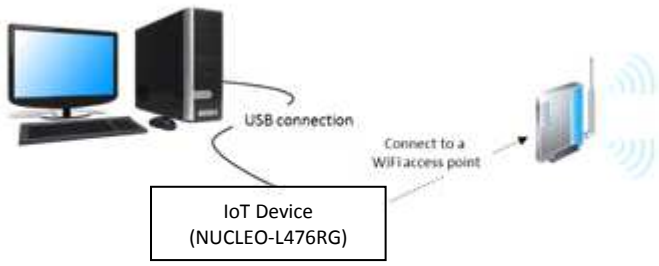
4 Hardware and software environment setup

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:

- Flash the board
- Store the Wi-Fi® AP credentials
- Interact with the board via a UART console
- Debug

The NUCLEO-L476RG must be connected to a Wi-Fi® access point as illustrated in *Figure 6*.

Figure 6. Hardware and software setup environment



The prerequisites for running the examples are:

- A Wi-Fi® access point, with a transparent Internet connectivity meaning that neither a proxy, nor a firewall are blocking the outgoing traffic. It has to run a DHCP server delivering the IP and DNS configuration to the board.
- A development PC for building the application, programming through ST-Link, and running the terminal console.
- An Aliyun account. Once registered a Product Key, Device Name and Device Secret for Aliyun IoT platform will be provided.
- To register and create an account, go to http://iot.console.aliyun.com

5 Interacting with the boards

A serial terminal is required to:
- Configure the board
- Display locally the sent/received Aliyun IoT device-to-cloud/cloud-to-device messages

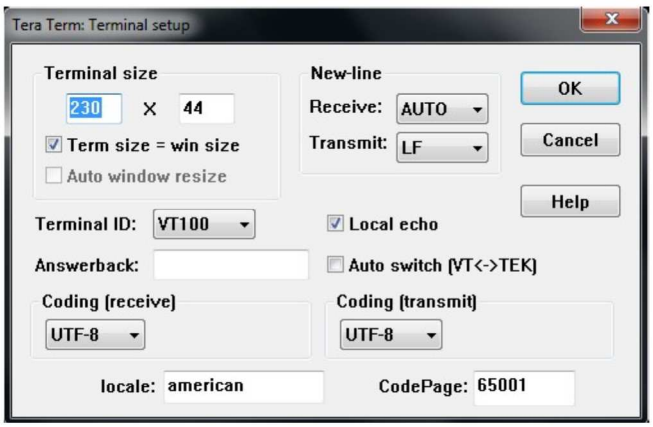
The example in this document is illustrated with the use of Tera Term. Any other similar tool can be used instead.
- Determine the STM32 ST-LINK Virtual COM port used on the PC for the Discovery board. On a Windows® PC, open the Device Manager.
- Open a virtual terminal on the PC and connect it to the 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 7*, 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) in order to allow copy-paste from UNIX type text files. The Local echo option makes copy- paste visible on the console.

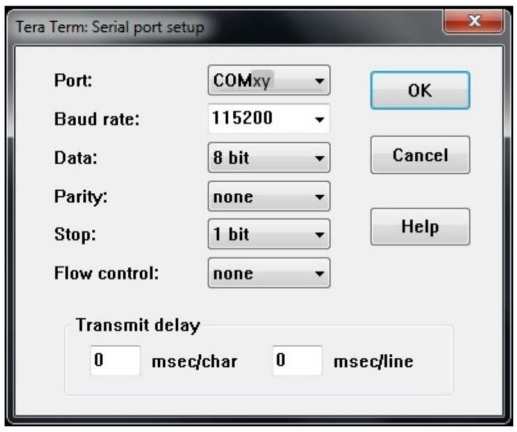
Figure 7. Terminal Setup



The serial port must be configured with:
- COM port number
- 115200 baud rate
- 8-bit data
- Parity none
- 1 stop bit
- No flow control

Serial port setup is illustrated in *Figure 8*.

Figure 8. 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.

6 Application example

This section introduces how to register and log on the Aliyun IoT platform, and how to use the Aliyun sample application from the I-CUBE-ALIYUN package.

6.1 Application description

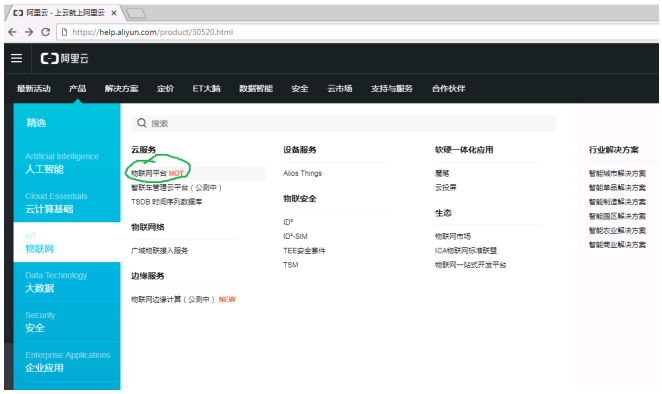
The Aliyun sample application illustrates the various way for a device to interact with the Aliyun IoT platform, using the Aliyun linkkit-embedded SDK on the device.

The application connects to the Aliyun IoT platform thanks to the credential provided by the user in the code.

6.2 Aliyun IoT account creation

To register and create an account, go to: http://iot.console.aliyun.com

Figure 9. Aliyun IoT account creation

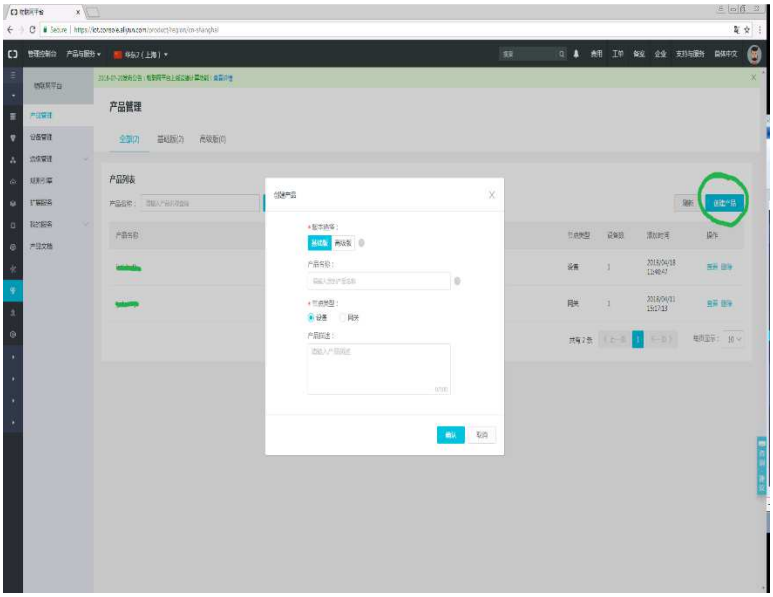


6.3 Device creation on the Aliyun IoT platform

Once an account has been created:

1. Create a new product, get Product Key.
2. Create a new device, get Device Name and Device Secret.
3. Create a new topic $(PRODECT_KEY)/$(DEVICE_NAME)/data, set the permission as: device is capable of publishing and subscribing.
4. Save the Product Key, Device Name, Device Secret for future device provisioning use.

Figure 10. Device creation



6.4 Application build and flash

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

Enter the Product Key, Device Name and Device Secret at the specified position in mqtt_example.c which is located at “\Projects\STM32L476RG-Nucleo\Src”

Figure 11. Device provisioning

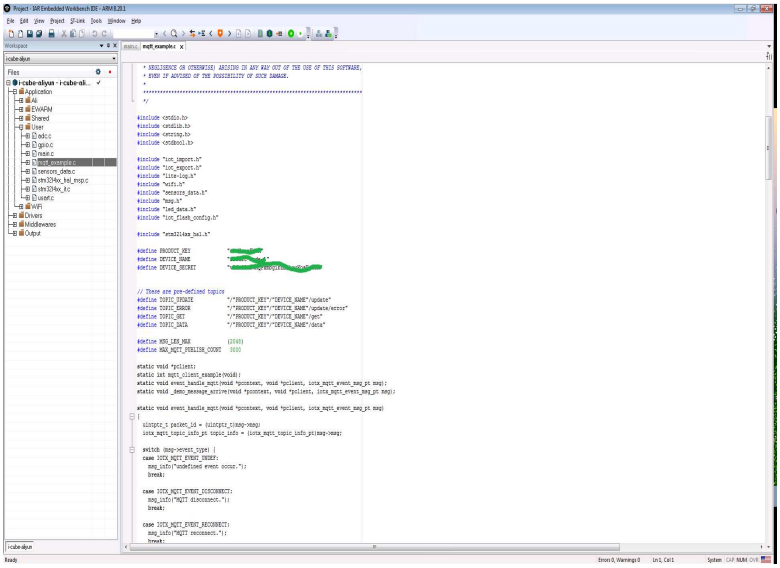
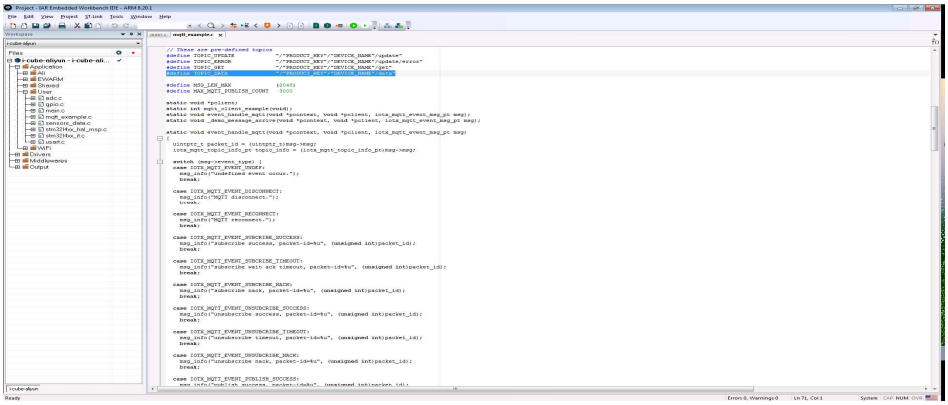


Figure 12. Device topic update



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

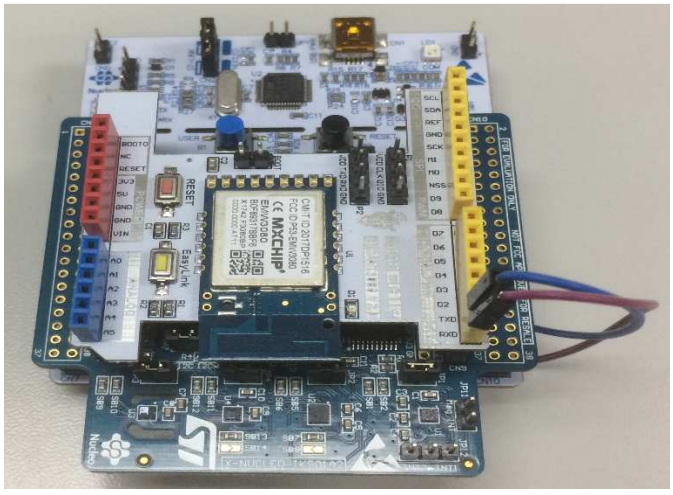
Program the firmware on the STM32 board: you can copy (or drag and drop) the generated bin file to the USB mass storage location created when you plug the STM32 board to your PC.

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

6.5 Application first launch

1. As the UART on Arduino connector for WiFi module is used by ST-Link. We need the wiring from WiFi Rx pin to Nucleo Morpho header CN10 pin 33 and WiFi Tx pin to Nucleo Morpho header CN10 pin 21.

Figure 13. Complete assembled board



2. The board must be connected to a PC through USB (ST-LINK USB port). Open the console through a serial terminal emulator (such as Tera Term), select the ST-

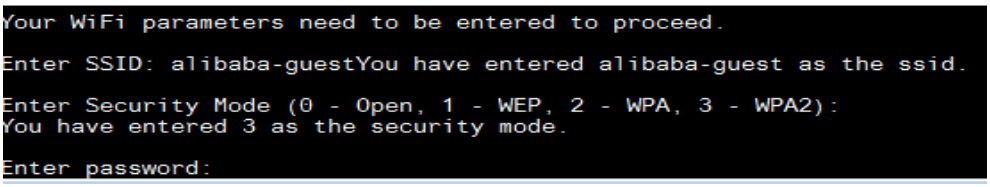
LINK COM port of the board, and configure it with:

– 8N1, 115200 bauds, no HW flow control
– Line endings set to LF

For more details, see Chapter 5: Interacting with the boards.

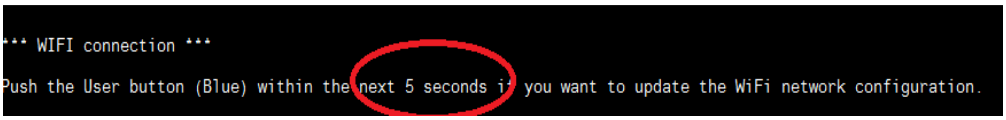
3. After the system boot up, enter the Wi-Fi configuration (SSID, encryption mode and password) via the console. If successfully connected to the WIFI access point, some information like MAC address will be showed as *Figure 12*.

Figure 14. WiFi provisioning



Note: After the parameters are configured, it is possible to change Wi-Fi network configuration by restarting the board and pressing User button (blue button) just after boot.

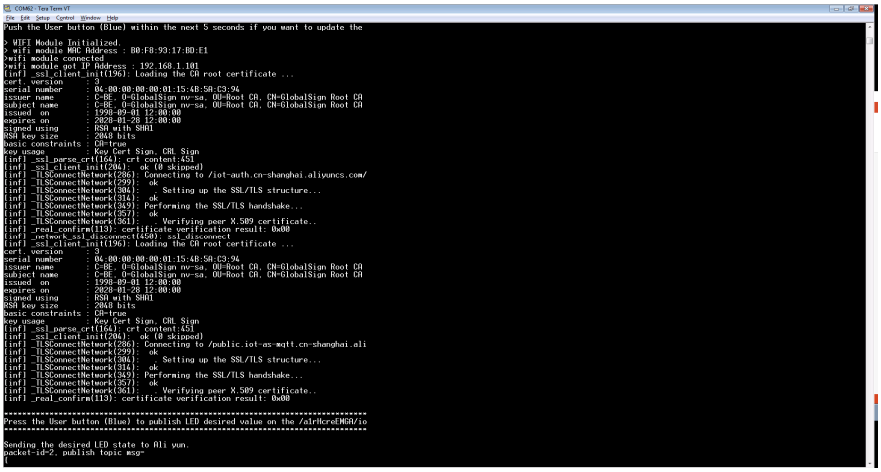
Figure 15. WiFi credential update



6.6 Application runtime

Once the WiFi connection is established, the board will connect to Aliyun IoT platform.

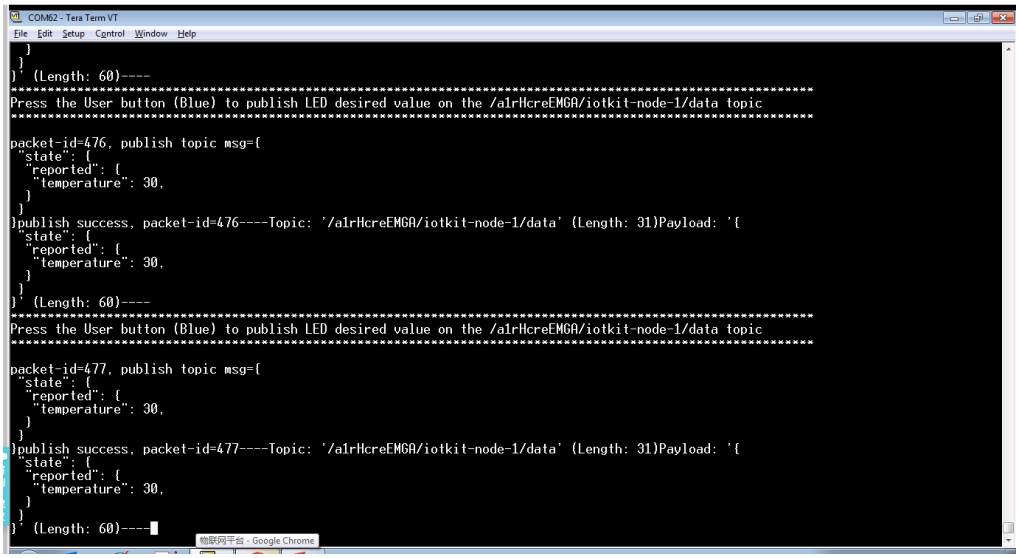
Figure 16. Device connect to Aliyun IoT platform



The application will publish and subscribe to accelerometer, gyroscope, magnetometer, pressure, humidity, temperature sensor data from X-NUCLEO-IKS01A2 sensor expansion board and STM32L476RG internal temperature sensor data.

User can push the user button (blue button) to toggle user LED (LD2) on Nucleo board, the desired LED state will be published to/subscribed from the cloud.

Figure 17. Device to cloud message



The sensor data and LED status are also visible on the Aliyun IoT platform console

Figure 18. Aliyun IoT platform console – device status

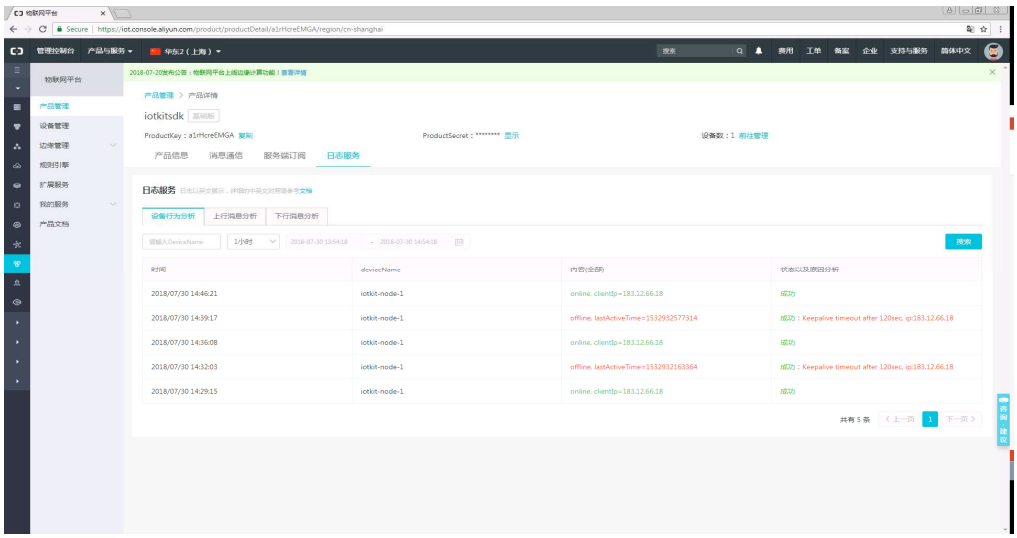


Figure 19. Aliyun IoT platform console – upstream data

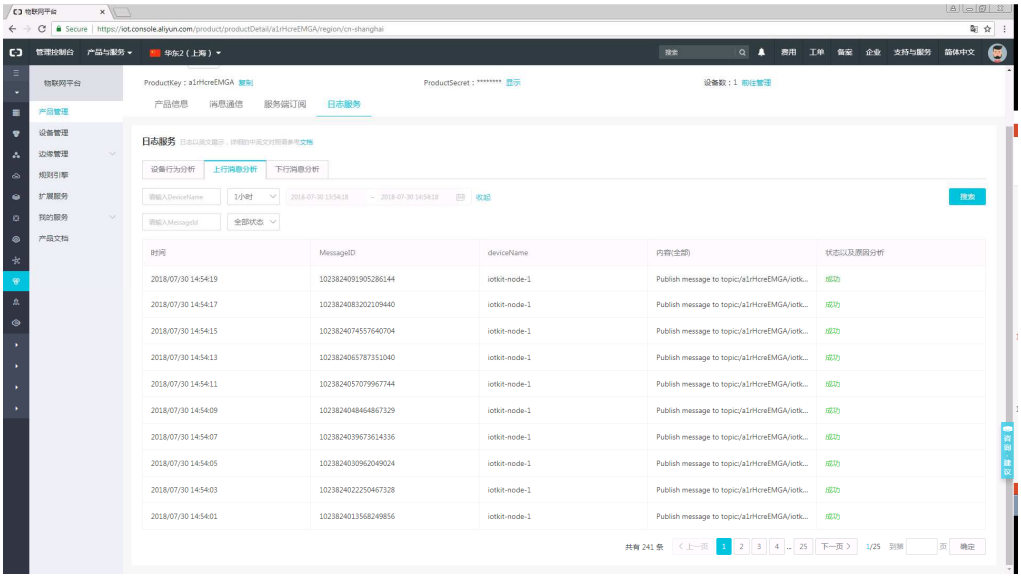
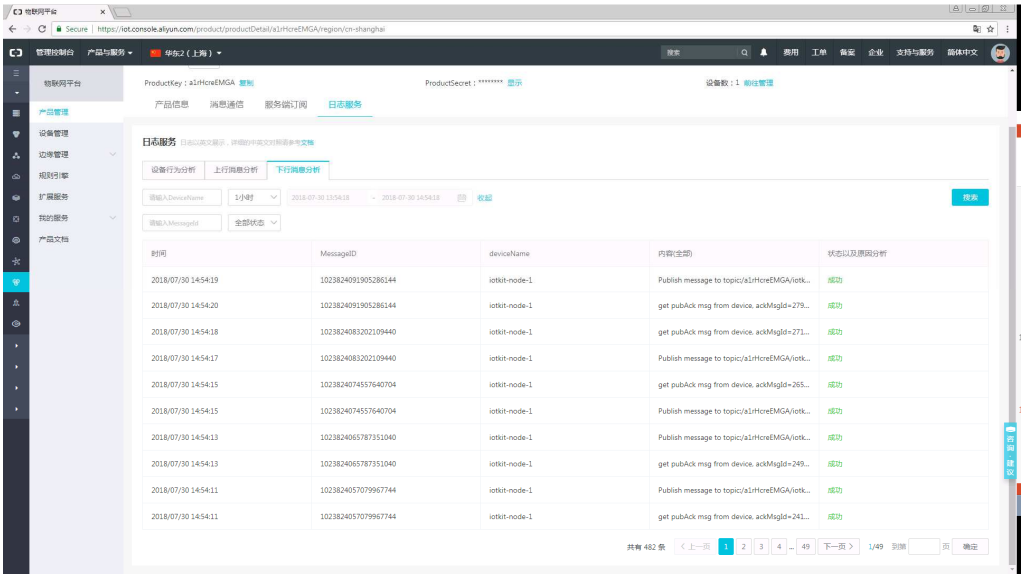


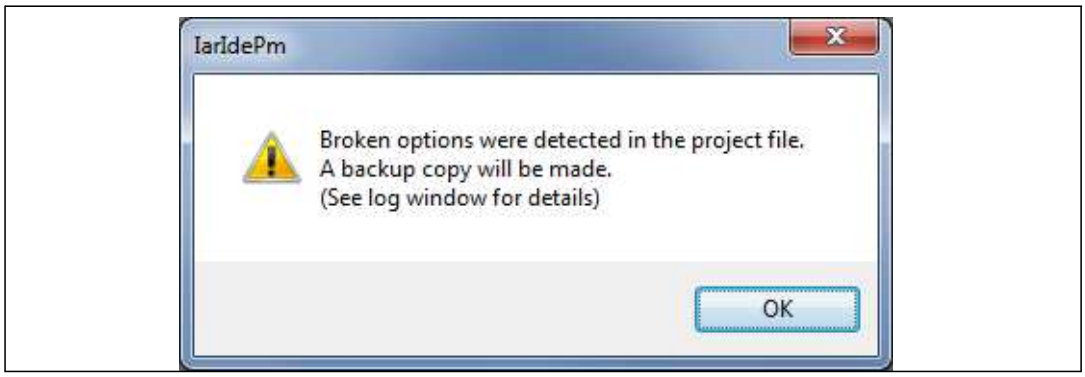
Figure 20. Aliyun IoT platform console – downstream data



7 Frequently asked questions

Q: Why do I get this pop-up (refer to *Figure 21*) when I open the project with IAR™?

Figure 21. Pop-up when the IAR™ IDE version is not compatible with the one used for I-CUBE-ALIYUN



A: It is very likely that the IAR™ IDE version is older than the one used to develop the package (refer to the release note available in the package root folder for the IDE versions supported), hence the compatibility is not ensured. In this case, the IAR™ IDE version needs to be updated.

Q: My device does not connect to the Wi-Fi® access point. How shall I proceed?

A: Make sure that another device can connect to the Wi-Fi® access point. If it can, enter the Wi-Fi® credentials by pressing the user button (blue) up to two seconds after board reset.

Q: How shall I modify my application to publish other messages?

A: Update the function prepare_mqtt_payload() is needed in the file sensor_data.c which is located at “\Projects\STM32L476RG-Nucleo\Src”.

8 Revision history

Table 2. Document revision history

Date	Revision	Changes
15-Oct-2018	1	Initial release.