

## 用于 STM32Cube 的 Google Cloud Platform™ 扩展包入门

### 引言

本用户手册描述 STM32Cube™ 扩展包的内容，该扩展包用于 Google Cloud Platform™（GCP）Google Cloud IoT Core 的服务。

用于 GCP 的 STM32Cube 扩展包（[X-CUBE-GCP](#)）提供了将 STMicroelectronics 板连接到 Google Cloud Platform™ 的 Google Cloud IoT Core 的应用示例。它将 MQTT 客户端库移植到相应的 STM32 器件上去实现与云平台的连接。

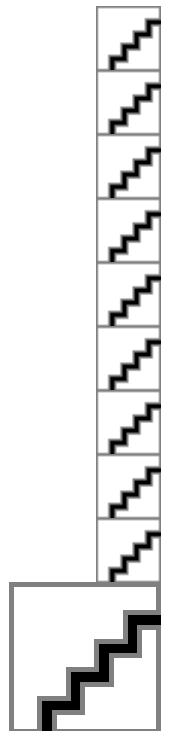
X-CUBE-GCP 可在五个平台上运行：

- 通过 Inventek ISM43362 模块支持 Wi-Fi® 连接的 [B-L475E-IOT01A](#) 和 [32F413HDISCOVERY](#) 板
- 提供原生以太网接口的 [32F769IDISCOVERY](#) 板
- [P-L496G-CELL01](#) 和 [P-L496G-CELL02](#) 探索包分别通过 2G/3G（UG96）和 LTE（BG96）Quectel 蜂窝调制解调器子板支持蜂窝连接

对于这五个平台，应用示例配置了网络连接参数，并说明了设备与 GCP 交互的各种方式。

该扩展包包含了设备到云端的遥测数据报告、用于安全连接云端的云端到设备的消息、向云端发送命令和从连接的设备接收通知的实现示例。

[B-L475E-IOT01A](#) 板报告遥测数据，例如湿度、温度和大气压力测量值。



## 1 概述

用于 Google Cloud Platform™ 的 Google Cloud IoT Core 的 X-CUBE-GCP 扩展包在基于 Arm® Cortex®-M 处理器的 STM32 32 位微控制器上运行。表 1 给出了相关缩略语的定义，帮助您更好地理解本文档。

表 1. 缩略语列表

| 术语    | 定义                      |
|-------|-------------------------|
| API   | 应用编程接口                  |
| BSP   | 板级支持包                   |
| CA    | 认证机构                    |
| DHCP  | 动态主机配置协议                |
| DNS   | 域名服务器                   |
| ECDSA | 椭圆曲线数字签名算法              |
| GCP   | Google Cloud Platform™  |
| HAL   | 硬件抽象层                   |
| HTTP  | 超文本传输协议                 |
| IDE   | 集成开发环境                  |
| IoT   | 物联网                     |
| IP    | 互联网协议                   |
| JSON  | JavaScript 对象符号         |
| JWT   | JSON 网络令牌               |
| LED   | 发光二极管                   |
| MQTT  | 消息队列遥测传输                |
| RSA   | Rivest、Shamir 和 Adleman |
| RTC   | 实时时钟                    |
| UART  | 通用异步收发器                 |

提示 Arm 是 Arm Limited（或其子公司）在美国和/或其他地区的注册商标。

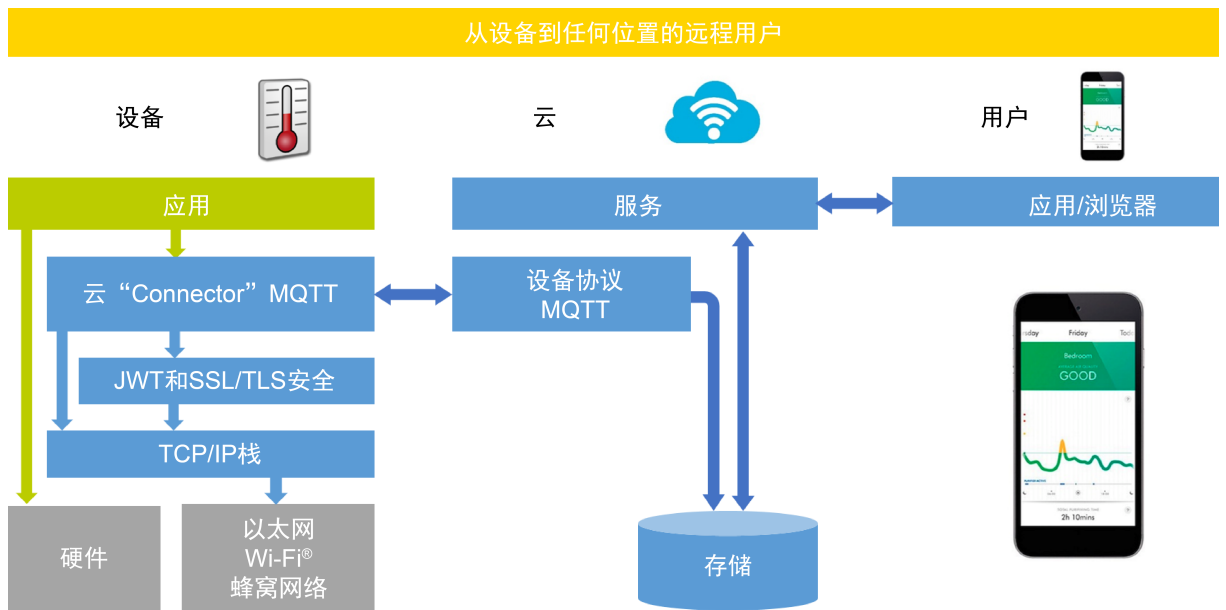


## 2 Google Cloud Platform™

本章介绍 Google Cloud Platform™ 的 Google Cloud IoT Core。与 Google Cloud Platform™（GCP）相关的详细信息可从 Google Cloud™ 专用网站 [cloud.google.com](https://cloud.google.com) 下载。

X-CUBE-GCP 实现嵌入式 C 客户端，能够将板安全地连接到 GCP 的 Google Cloud IoT Core。X-CUBE-GCP 所针对的 GCP 生态系统如图 1 所示。

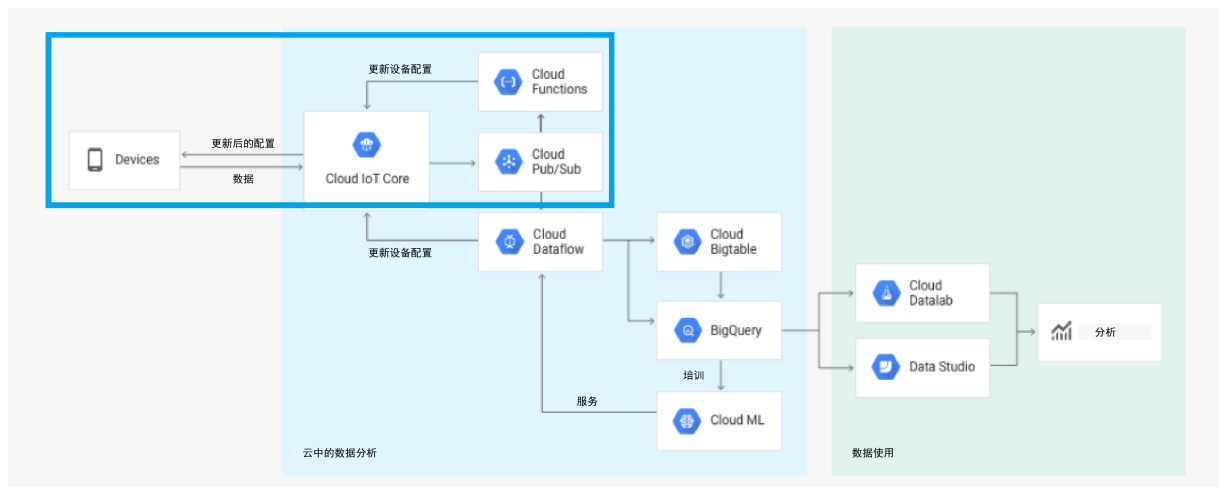
图 1. Google Cloud IoT Core 生态系统



用户可以使用智能手机或个人计算机连接到云端，并可以随时随地访问该板提供的信息。

Google Cloud™ 在 GCP 在线文档中展示的 Google Cloud IoT Core 如图 2 所示。

图 2. Google Cloud IoT Core 图



注意

©2018 Google LLC，经许可使用。Google 和 Google 徽标是 Google LLC 的注册商标

X-CUBE-GCP 实现必要的服务，将设备连接到图 2 中用蓝色框出的生态系统部分。



特别是，该扩展包演示了如何配置 GCP 的 IoT Core 服务，以便将设备安全地连接到 Google Cloud IoT Core 服务。

设备使用私钥（ECDSA 或 RSA）签署连接云平台时需要的 JWT（JSON 网络令牌）。Cloud IoT Core 通过在设备创建过程中上传到云服务的相应公钥验证设备的 JWT。

X-CUBE-GCP 还提供了示例，展示了如何使用数据格式和使用 MQTT 协议在设备与 IoT Core 元件之间交换数据，将遥测数据发送到云端，以及从云端接收配置数据。发送到 Cloud IoT Core 的数据随后可通过 Pub/Sub（发布/订阅）服务发布，并能在日后需要使用时通过网络应用进行访问。用于数据分析和使用的网络应用的开发不在 X-CUBE-GCP 的范围之内。



## 3 软件包说明

本章详述了 X-CUBE-GCP 扩展包的内容和使用。

### 3.1 概述

X-CUBE-GCP 扩展包由多系列的基于 STM32L4 系列, STM32F4 系列和 STM32F7 系列微控制器的库和应用示例组成。

X-CUBE-GCP 可在五个平台上运行:

- 通过 Inventek ISM43362 模块支持 Wi-Fi®连接的 B-L475E-IOT01A 和 32F413HDISCOVERY 板
- 提供原生以太网接口的 32F769IDISCOVERY 板
- P-L496G-CELL01 和 P-L496G-CELL02 包分别通过 Quectel 的 2G/3G UG96 和 LTE BG96 蜂窝调制解调器子板支持蜂窝连接。用于连接 2G/3G 蜂窝网络与云端的 P-L496G-CELL01 STM32 探索包 (STM32-C2C/2G-3G) 和用于连接 LTE 蜂窝网络与云端的 P-L496G-CELL02 STM32 探索包 (STM32-C2C/2G-LTE) 是基于蜂窝网络和云技术的解决方案的完整开发平台。探索包内含一个基于 STM32L496AGI6 的带预加载固件的低功耗探索主板和一个带天线的 STMod+蜂窝网络扩展板。

X-CUBE-GCP 扩展包包含以下组件:

- Eclipse™ Paho MQTT 嵌入式 C 客户端
- mbedTLS
- 用于 B-L475E-IOT01A 和 32F413HDISCOVERY 板的 Inventek ISM43362 Wi-Fi®驱动程序
- 用于 32F769IDISCOVERY 板的以太网驱动程序、FreeRTOS™和 LwIP
- 用于 B-L475E-IOT01A 板的传感器驱动
- 用于 P-L496G-CELL01 和 P-L496G-CELL02 套件的蜂窝网络驱动程序
- 用于所有 MCU 板的 BSP
- STM32L4 系列、STM32F4 系列, 和 STM32F7 系列 HAL
- 应用程序示例

软件以 zip 文档的形式提供, 其中包含源代码。可支持以下集成开发环境:

- IAR Embedded Workbench® 对于 Arm® (EWARM)
- Keil®微控制器开发套件 (MDK-ARM)
- System Workbench for STM32

有关所支持 IDE 版本的信息, 请参阅软件包根文件夹中的版本说明。

### 3.2 架构

本节介绍 X-CUBE-GCP 中的软件组件。X-CUBE-GCP 是 STM32Cube™的扩展软件。其主要功能和特性如下:

- 完全兼容 STM32Cube™架构
- 为了能够开发应用程序来访问和使用 Google Cloud Platform™的 Google Cloud IoT Core, 扩展了 STM32Cube™
- 基于 STM32 微控制器的硬件抽象层 STM32CubeHAL

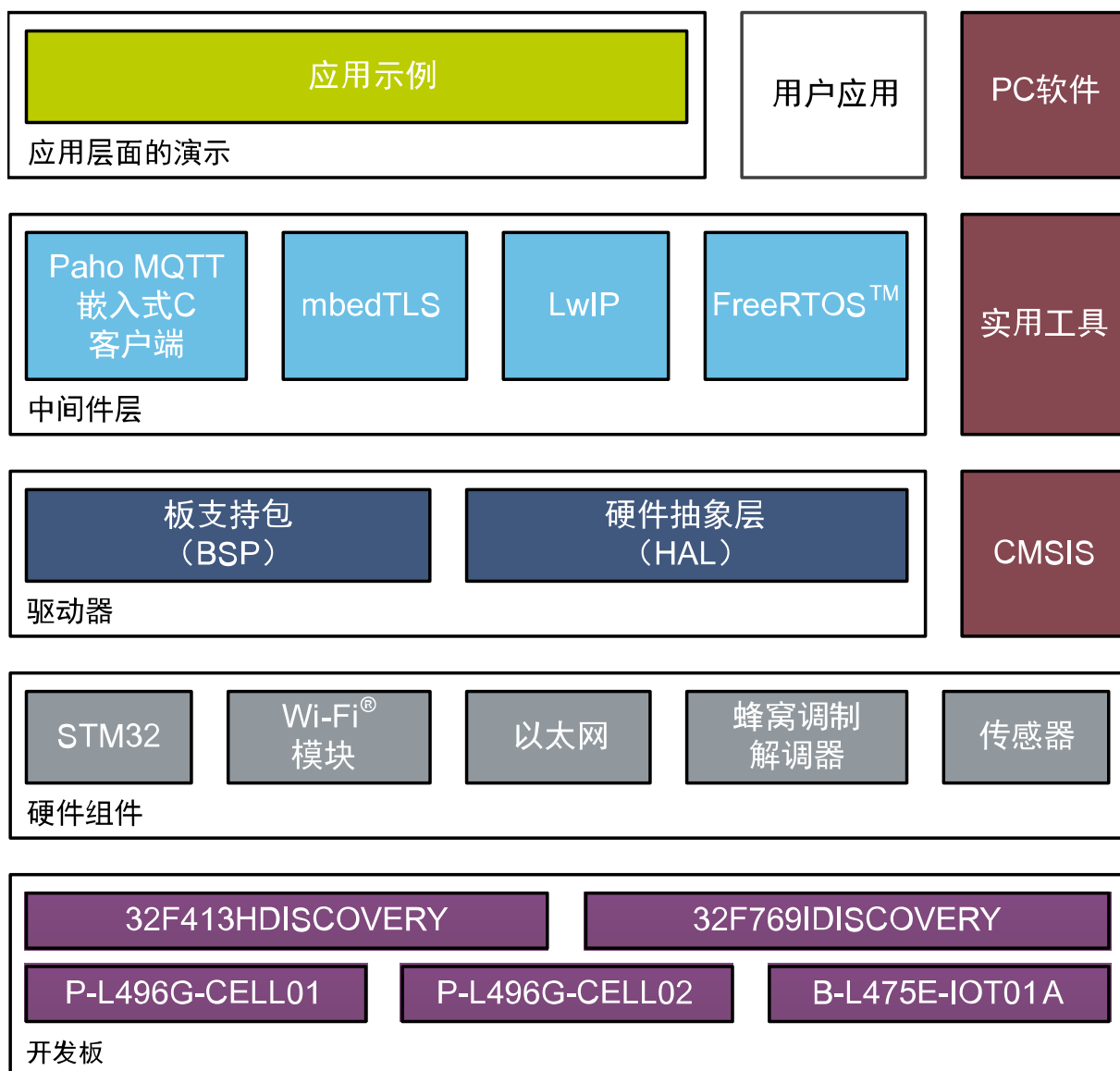
应用软件访问和使用 Google Cloud Platform™的 Google Cloud IoT Core 所用的软件组件如下:

- STM32Cube HAL  
HAL 驱动层提供通用的多实例简单 API 组 (应用程序编程接口), 以便与上层 (应用、库和协议栈) 交互。它由通用和扩展 API 构成。它直接围绕通用架构构建, 允许在其基础上的软件层, 例如中间件层, 实现了它的功能又无需依赖给定 MCU 的特定硬件配置。  
此结构可提高库代码的可复用性, 并确保可向其他设备轻松移植。
- 板级支持包 (BSP)  
除 MCU 之外, 板软件包需支持 STM32 板上的外设。板支持包中包含此软件。这是一个有限的 API 集, 为板特有的某些外设 (例如 LED 和用户按钮等) 提供编程接口。

- **mbedTLS**  
MQTT 中间件使用由 mbedTLS 库实现的 TLS 连接。
- **MQTT 客户端中间件**  
它由 Eclipse™ Paho MQTT 嵌入式 C 客户端库（被 MQTT 应用用作传输层）和 JSON 解析器构成。
- **TCP/IP**  
TCP/IP 连接可由 Wi-Fi®模块（当使用 Wi-Fi®连接时）、LwIP TCP-IP 栈（当使用以太网连接时）或通过蜂窝连接（当使用蜂窝网络扩展板时）来处理。
- **FreeRTOS™**  
它是一个实时操作系统，使用 LwIP 为用户提供基于套接字的接口时，需要用到。
- **Google Cloud™ IoT 示例应用程序**  
一个应用示例，实现了连接 GCP Google Cloud IoT Core 组件所需的设备服务。这包括连接管理、JWT 安全连接处理和与云端交互的合适消息格式。

X-CUBE-GCP 软件的架构如图 3 所示。

图 3. X-CUBE-GCP 软件架构



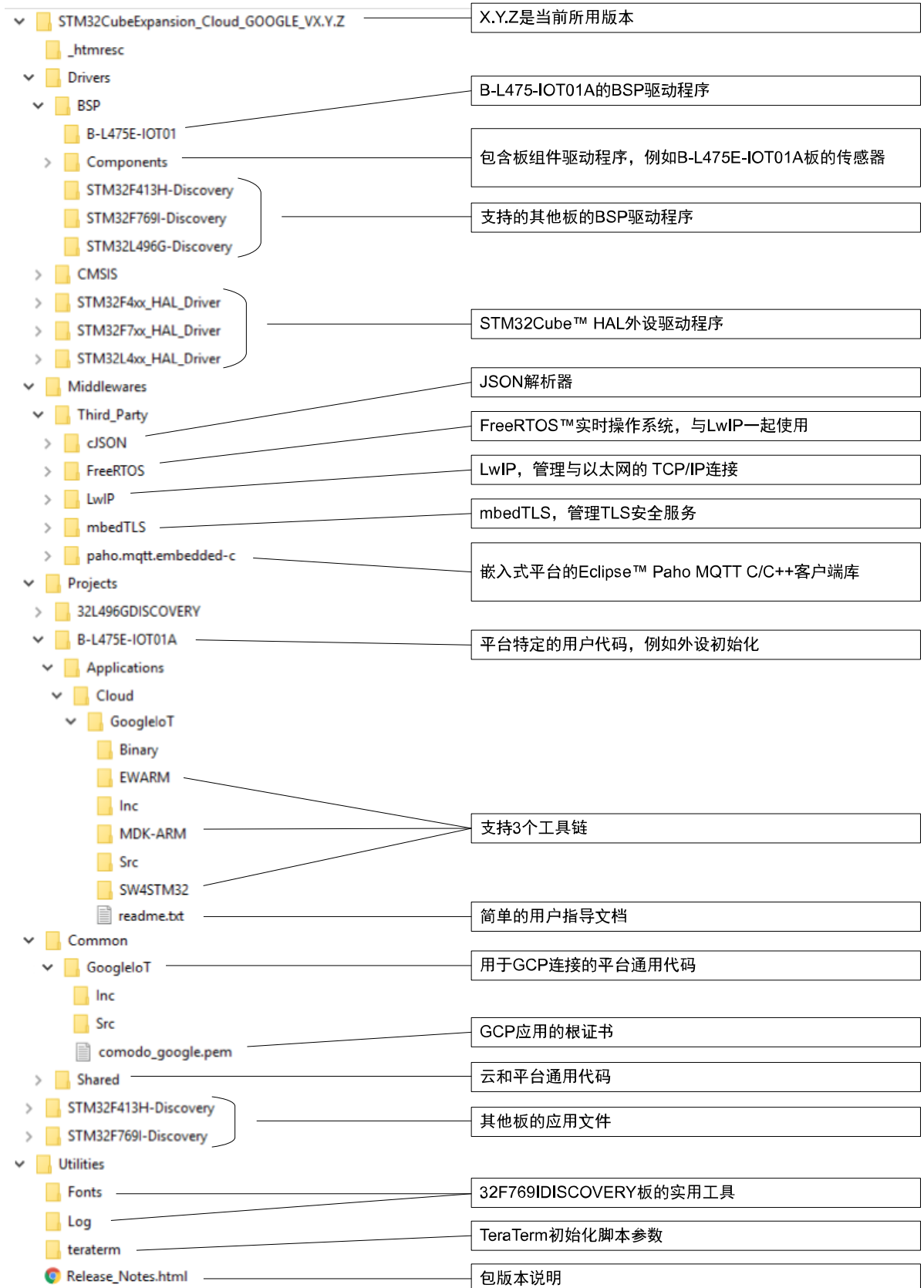


### 3.3 文件夹结构

X-CUBE-GCP 扩展包的文件夹结构如图 4 所示。



图 4. 项目文件结构



### 3.4 B-L475E-IOT01A 板传感器

应用示例使用的板载传感器包括：

- 用于相对湿度和温度测量的电容式数字传感器（HTS221）
- 高性能 3 轴磁力计（LIS3MDL）
- 3D 加速度计和 3D 陀螺仪（LSM6DSL）
- 260-1260 hPa 绝对数字输出气压计（LPS22HB）
- 接近传感器（VL53L0X）

B-L475E-IOT01A 板传感器报告值的单位如表 2 所示。

表 2. B-L475E-IOT01A 板传感器报告值的单位

| 数据  | 单位           |
|-----|--------------|
| 温度  | 摄氏度（°C）      |
| 湿度  | 相对湿度（%）      |
| 压力  | 百帕斯卡（hPa）    |
| 接近  | 毫米（mm）       |
| 加速度 | 毫重力（mgforce） |
| 角速度 | 毫度每秒（mdps）   |
| 磁感应 | 毫高斯（mG）      |

### 3.5 Wi-Fi®元件

Wi-Fi®软件分为两部分，*Drivers/BSP/Components* 下是模块专用软件，*Projects/<board>/WiFi* 下是 I/O 操作和 Wi-Fi®模块的抽象层。

### 3.6 复位按钮

复位按钮（黑色）用于随时复位板子。此操作可使板重新启动。

### 3.7 用户按钮

用户按钮（蓝色）用于以下情况：

- 要配置 Wi-Fi®和 Google Cloud™安全证书，从板启动时开始直到启动后五秒，都可以完成此操作。
- 板子初始化后，如果用户按下按钮的时间很短，则应用程序仅发布 B-L475E-IOT01A 板子的传感器值、0/1 切换值（绿色 LED 将相应切换）以及时间戳。双击按钮后，应用程序进入循环并每秒自动发布一次。再一次双击将恢复先前的模式。应用程序通过板级支持包功能来配置和管理用户按钮。BSP 功能位于 *Drivers\BSP\<board name>* 目录中。使用 BSP 按钮功能时（采用 BUTTON\_USER 值），对于给定的平台，应用不会从硬件角度考虑此按钮连接的方式；该映射由 BSP 处理。

### 3.8 用户 LED

应用程序所用的用户 LED 配置通过板级支持包功能完成。

BSP 功能位于 *Drivers\BSP\<board name>* 目录中。

使用 BSP 功能时（采用 LED\_GREEN 值），对于给定的平台，应用不会考虑 LED 映射的方式；该映射由 BSP 处理。

### 3.9 实时时钟

STM32 器件的 RTC 在启动时从 [www.gandi.net](http://www.gandi.net) 网络服务器进行更新。用户可以使用 HAL\_RTC\_GetTime() 函数得到时间值。例如，此函数可以用于时间戳消息。



### 3.10 mbedTLS 配置

利用 `#include` 配置文件，可以对 mbedTLS 中间件进行配置。可使用 `MBEDTLS_CONFIG_FILE` `#define` 指令覆盖配置文件名。X-CUBE-GCP 包使用文件 `googleiot_mbedtls_config.h` 来实现项目配置。这可以通过在 `mbedtls.c` 和 `mbedtls.h` 文件开头设置以下 `#` 指令来实现：

```
#if !defined(MBEDTLS_CONFIG_FILE)
#include "mbedtls/config.h"
#else
#include MBEDTLS_CONFIG_FILE
#endif
```

配置文件指定了要使用的相关加密参数。

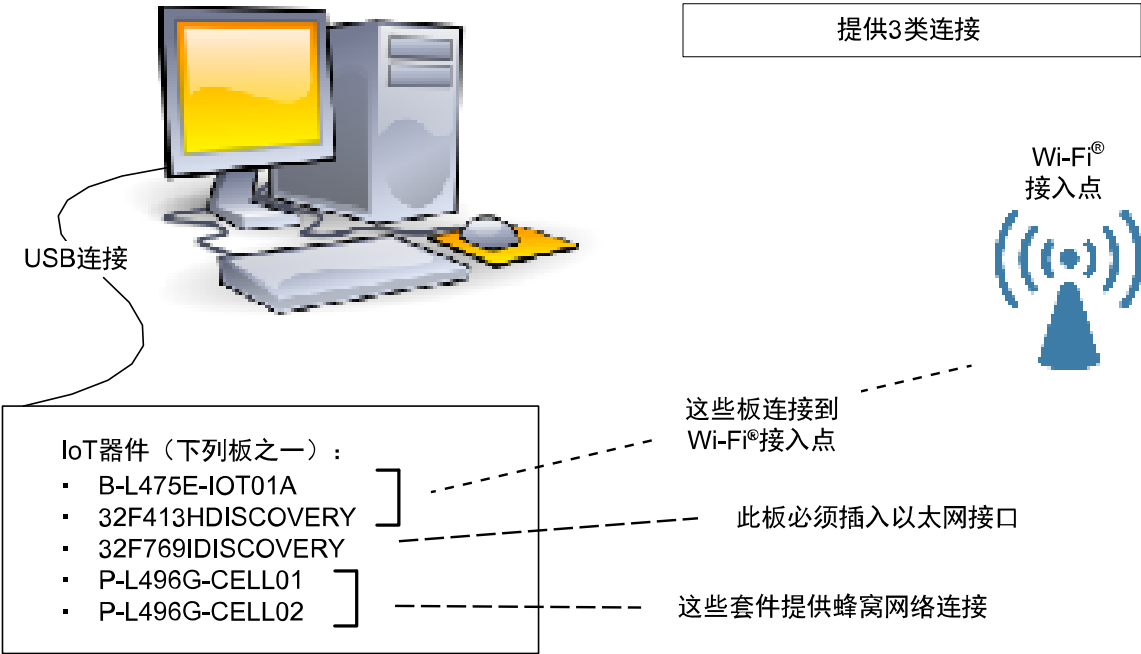
## 4 硬件和软件环境设置

为了设置硬件和软件环境，必须利用 USB 线将支持的板连接到个人计算机。连接到 PC 后，允许用户：

- 烧写板
- 存储 Wi-Fi®和 Google Cloud™安全证书
- 通过 UART 控制台与板进行交互
- 调试

B-L475E-IOT01A 和 32F413HDISCOVERY 板必须连接到 Wi-Fi®接入点，P-L496G-CELL01 和 P-L496G-CELL02 套件的天线必须插入以连接到周围的蜂窝网络，并且 32F769IDISCOVERY 板必须连接到以太网接口，如图 5 所示。

图 5. 硬件和软件设置环境



运行示例的先决条件是：

- 以下连接解决方案之一：
  - 一个 Wi-Fi® 接入点，具有透明的互联网连接，也就是说不能有代理服务器或者防火墙阻止开发板与外网的通信。它必须运行一个 DHCP 服务器，将 IP 和 DNS 配置提供给主板。
  - 以太网连接，具有透明的互联网连接，也就是说不能有代理服务器或者防火墙阻止开发板与外网的通信。
  - 有效的蜂窝网络订阅，激活了无线通信数据服务，来自集成了 MVNO 配置文件的嵌入式 SIM 或外部 Micro-SIM
- 一台开发 PC，用于构建应用、利用 ST-LINK 进行编程以及运行虚拟控制台。
- 一台计算机，用于运行连接 Google Cloud™ 控制台的网络浏览器或管理 Google Cloud Platform™ 的 GCLOUD SDK 命令行工具。计算机防火墙（如果有）必须允许 MQTT 连接进入（通常是端口 1883 和 8883）。  
*例如，它可以是开发 PC、虚拟专用服务器或单板计算机。*
- OpenSSL 工具套件，用于构建新的设备公钥/私钥对。

## 5 应用程序示例

本章介绍如何注册和登录 Google Cloud Platform™ 的 Google Cloud IoT Core，以及如何使用 X-CUBE-GCP 扩展包中的 GCP IoT Core 应用。此外，还展示了如何从 Google Cloud™ 控制台配置 GCP，但也可以使用 Google Cloud™ SDK 中的命令行工具应用相同配置。请参考 Google Cloud™ SDK 文档了解关于命令行工具的更多详细信息。

### 5.1 应用描述

应用提供下列服务：

- 连接管理  
应用首先配置板连接（Wi-Fi®、以太网或蜂窝网络），并确保连接正常且正常工作，以便提供有效的 IP 地址和 IP 连接。
- 通过 JWT 提供安全连接  
一旦有 IP 连接可用，应用将通过 JSON 网络令牌（JWT）创建安全地连接到 GCP IoT Core 服务。GCP IoT core 支持多种密钥格式。该应用支持两种密钥格式：ES256 和 RS256。
- 发送/获取设备状态  
一旦安全连接，应用将向 GCP IoT Core 发送更新后的设备状态。这将允许从 GCP 控制台获取设备信息。设备状态是设备的内部表达。  
在 X-CUBE-GCP 应用中，设备状态是一个简单的 JSON 字符串，包含板 LED 状态、遥测间隔、本地时间戳、MAC 地址和当前固件版本。下面是发布的设备状态示例：

```
{
  "LedOn": false,
  "TelemetryInterval": 60,
  "ts": 1530170497,
  "mac": "C47F5104341",
  "FW version": "V1.1.0"
}
```

- 配置设备  
GCP 提供修改设备配置的功能。根据 GCP 文档中的定义，设备配置是一种任意的用户定义的数据 BLOB。X-CUBE-GCP 应用使用配置从云端控制 LED 状态，更改遥测间隔并强制重启。应用接受简单 JSON 格式的配置，如下所示：

```
{
  "LedOn": true,
  "TelemetryInterval": 60,
  "Reboot": false
}
```

X-CUBE-GCP 在每次连接时 对配置进行注册。每次重新注册时，GCP IoT Core 服务总是发送最新配置。因此，如果在配置中发送重启 命令 "Reboot": true，用户必须将 配置重新更新为 "Reboot": false，以免在每次重新连接时发生无限重启。

- 通过 MQTT Bridge 发送遥测数据

应用只支持 GCP IoT Core 的 MQTT Bridge 服务。它不支持 HTTP Bridge。

一旦用户按一次用户按钮，应用就发送遥测数据。如果用户按两次用户按钮，设备将进入发布循环，并定期发送遥测数据。在最初的设备配置步骤中使用参数 `TelemetryInterval` 配置此循环的周期。

如 GCP 在线文档中所述，设备可以通过发出 **PUBLISH** 消息（通过 MQTT 连接）来发布遥测事件。应用示例中使用的 **Quality of Service** 是 **QOS1**（至少传输一次）。必须用以下格式将消息发布到 MQTT 主题： /

`devices/{device-id}/events`

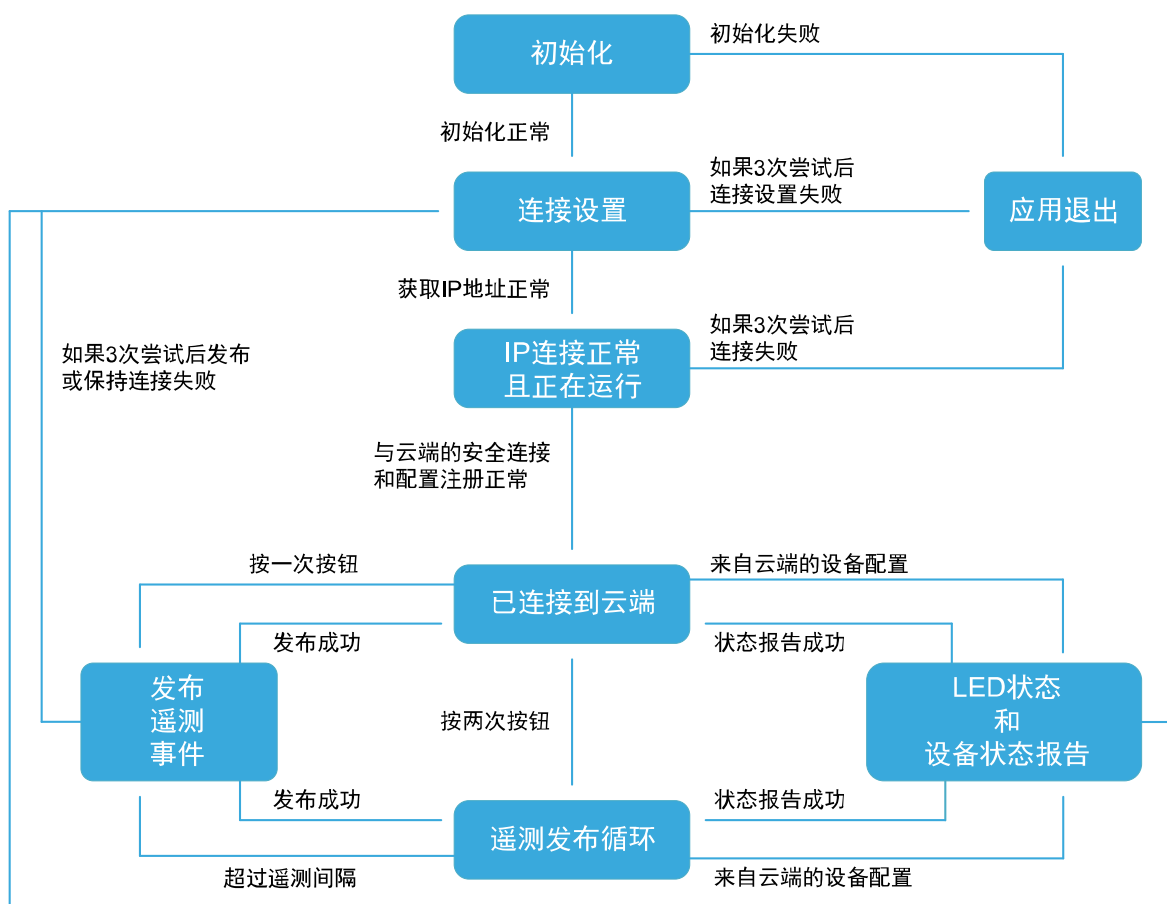
下面是带传感器的板的消息发布示例：

```
{
  "data":
  [{"dataType": "temp", "value": "27.91"},
   {"dataType": "hum", "value": "48.72"},
   {"dataType": "pres", "value": "1015.17"}]
}
```

应用行为 如图 6 所示。



图 6. 应用状态机



## 5.2 GCP 和 IoT Core 帐户设置

为了使用 Google Cloud IoT Core 服务，需创建 Google Cloud Platform™ 帐户。用户必须先创建帐户，然后才能继续操作并创建或选择 GCP 项目。

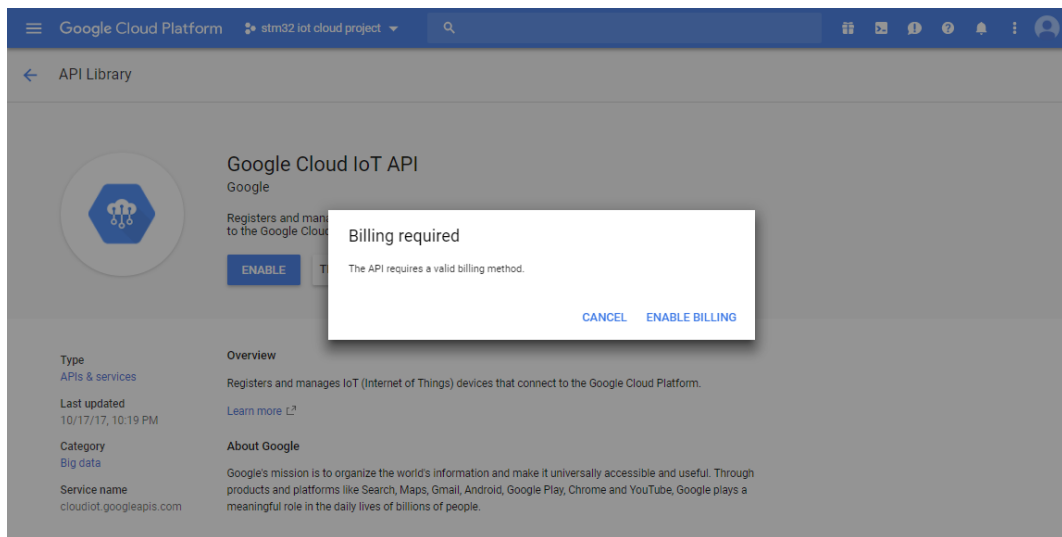
提示

关于注册、项目创建和 API 使能，请访问 [cloud.google.com](https://cloud.google.com)。

创建用户帐户的步骤如下：

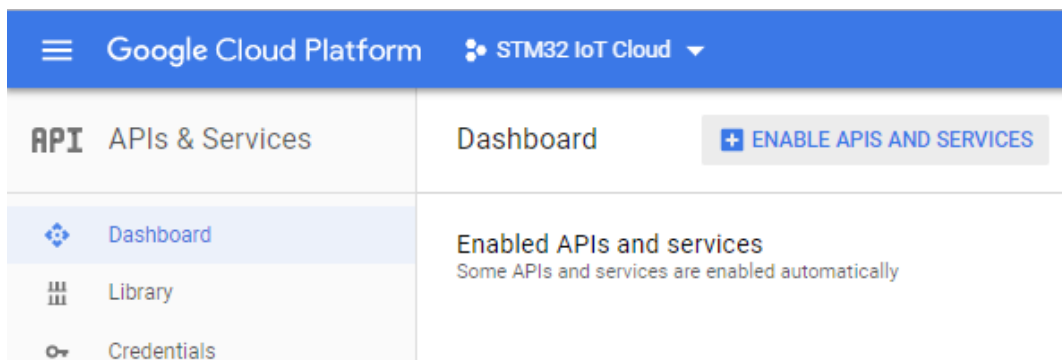
1. 使能项目计费，如图 7 所示。如不使能项目计费，用户将无法继续执行帐户创建的剩余步骤。

图 7. 需要计费



2. 使能应用使用的“Google Cloud IoT Core”和“Pub/Sub” API。在 GCP 控制台中，进入“API 和服务仪表板”，如图 8 所示。

图 8. 使能 API 菜单



3. 使用“使能 API 和服务”并至少使能 Google Cloud IoT API 和 Cloud Pub/Sub API，如图 9. 使能 Google Cloud IoT API 和图 10. 使能 Cloud Pub/Sub API 所示。

图 9. 使能 Google Cloud IoT API

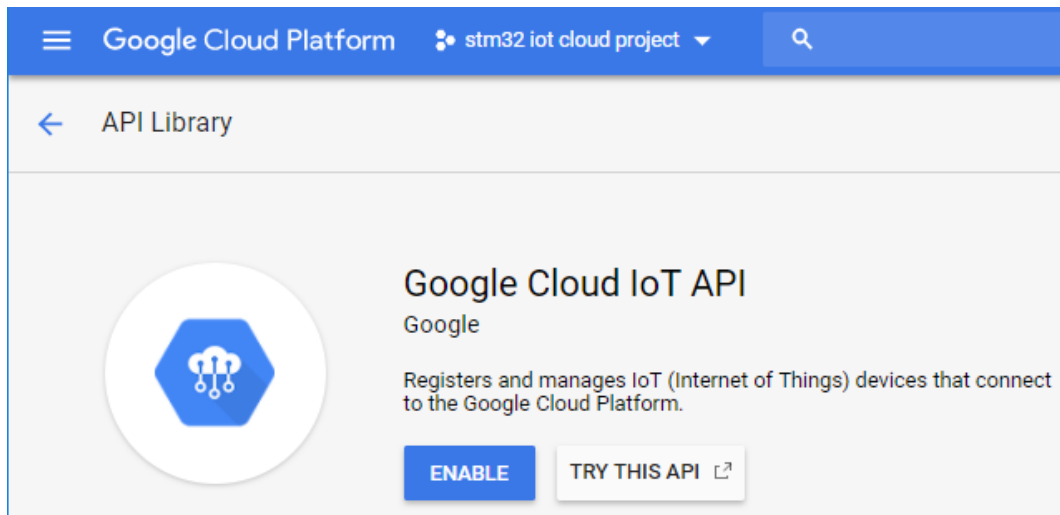
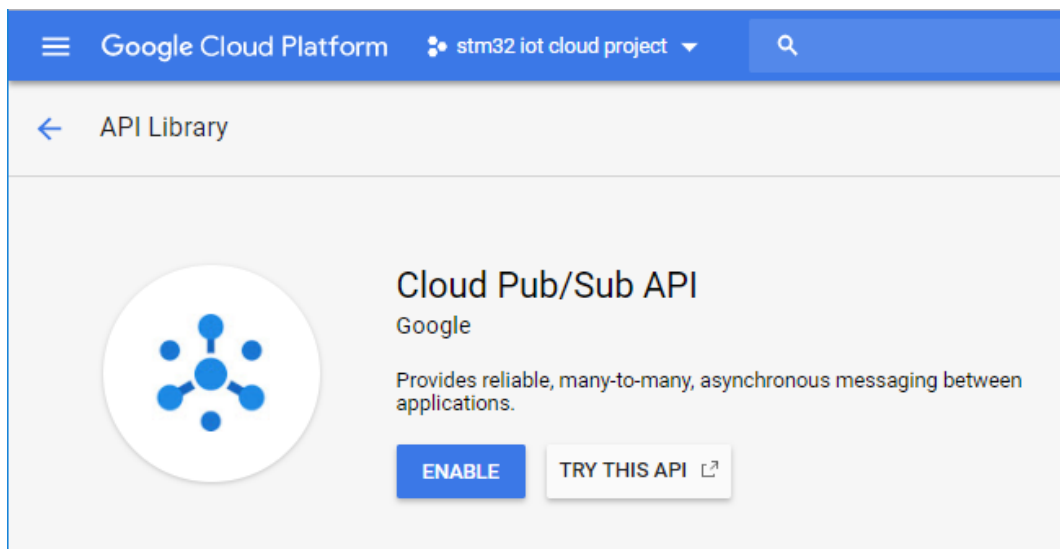


图 10. 使能 Cloud Pub/Sub API



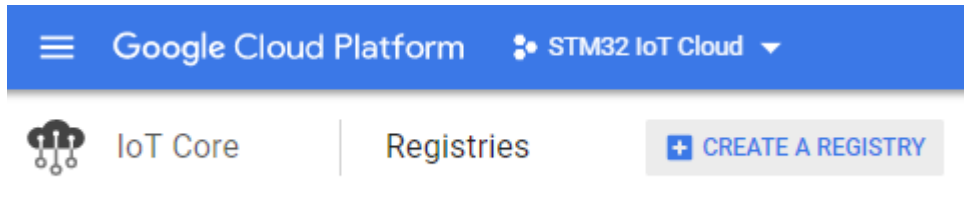
### 5.3 在 GCP IoT core 上创建设备

关于 Google Cloud IoT Core 的完整文档和如何创建注册表、设备和遥测主题，请访问 [cloud.google.com/iot/docs](https://cloud.google.com/iot/docs)。

按照下述步骤创建并注册新设备：

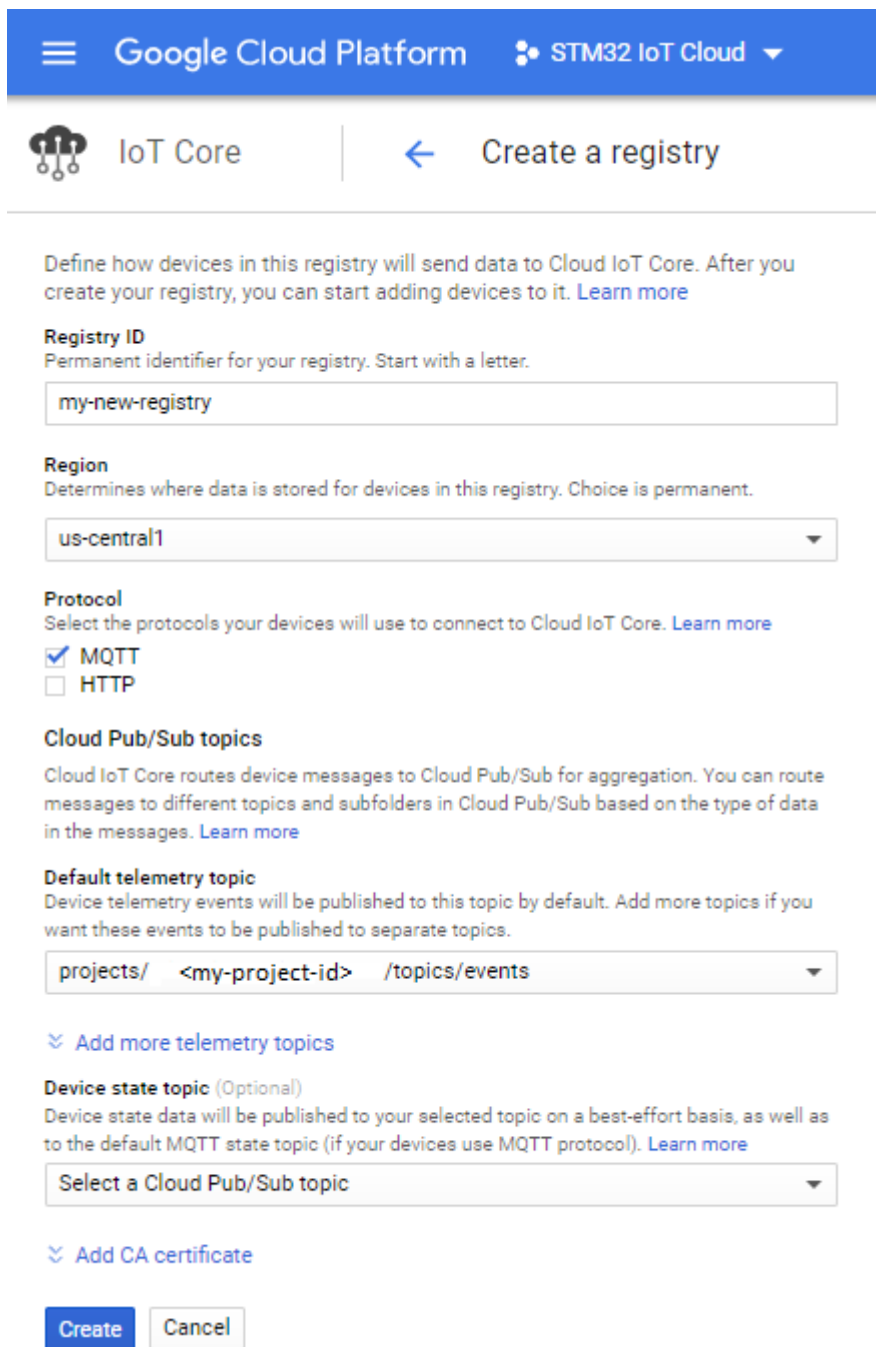
1. 在 Google Cloud IoT Core 中，创建新设备注册表（在连接字符串中定义为{registry-id}），如图 11 所示。

图 11. GCP“创建注册表”按钮



2. 在注册表创建面板上选择一个区域（默认为 us-central1），以后可以将其作为连接字符串中的{cloud-region}进行配置。如果不存在，创建名为“events”的遥测主题，并维持默认状态主题。该主题必须命名为“events”，因为这是应用使用的名称。应用只支持 MQTT 协议；不支持 HTTP 且可以取消选中 HTTP。用户无需添加证书。注册表创建设置的专用窗口如图 12 所示。

图 12. GCP“创建注册表”设置



Google Cloud Platform STM32 IoT Cloud

IoT Core Create a registry

Define how devices in this registry will send data to Cloud IoT Core. After you create your registry, you can start adding devices to it. [Learn more](#)

**Registry ID**  
Permanent identifier for your registry. Start with a letter.

my-new-registry

**Region**  
Determines where data is stored for devices in this registry. Choice is permanent.

us-central1

**Protocol**  
Select the protocols your devices will use to connect to Cloud IoT Core. [Learn more](#)

☒ MQTT  
☐ HTTP

**Cloud Pub/Sub topics**  
Cloud IoT Core routes device messages to Cloud Pub/Sub for aggregation. You can route messages to different topics and subfolders in Cloud Pub/Sub based on the type of data in the messages. [Learn more](#)

**Default telemetry topic**  
Device telemetry events will be published to this topic by default. Add more topics if you want these events to be published to separate topics.

projects/ <my-project-id> /topics/events

⌵ Add more telemetry topics

**Device state topic (Optional)**  
Device state data will be published to your selected topic on a best-effort basis, as well as to the default MQTT state topic (if your devices use MQTT protocol). [Learn more](#)

Select a Cloud Pub/Sub topic

⌵ Add CA certificate

Create Cancel

- 在新创建的注册表中添加新设备  
定义设备 ID，即连接字符串中的 {device-id}。  
按照 GCP 的“[创建公钥/私钥对](#)”文档中的描述，使用“*openssl*”创建 RS256 或 RS256 类型的公钥和私钥。
  - 对于 RS256:
 

```
> openssl genrsa -out rsa_private.pem 2048
> openssl rsa -in rsa_private.pem -pubout -out rsa_public.pem
```
  - 对于 ES256:



```
> openssl ecparam -genkey -name prime256v1 -noout -out ec_private.pem  
> openssl ec -in ec_private.pem -pubout -out ec_public.pem
```

将公钥 **pem** 文件的内容添加到设备并保存私钥，私钥将通过控制台接口传递到设备。设备的创建如图 13. GCP“创建设备”设置所示。

图 13. GCP“创建设备”设置

IoT Core | Add a device

Add a device to registry my-new-registry.

**Device ID** ?

my-new-device

**Device communication** ?

☒ Allow

☐ Block

**Authentication (Optional)** ?

**Input method**

☒ Enter manually

☐ Upload

**Public key format**

☐ RS256 ?

☒ ES256 ?

☐ RS256\_X509 ?

☐ ES256\_X509 ?

**Public key value**

```
-----BEGIN PUBLIC KEY-----
MFkwEwYIKoZIzj0DAQcDQgAEqzUYDiYyzUAs+CPaU+jhDz9Gw0NZ
iB4/+wTRDiwdQruW4AejesMe/Jlh4/a1YL1sKwlgfICikRFhlzHenhZ0oQ==
-----END PUBLIC KEY-----
```

**Public key expiration date (Optional)**

☐ Expires on:

6/29/19, 1:52 PM CEST

**Device metadata (Optional)** ?

Key must contain only letters, numbers, hyphens, and underscores, and be no longer than 128 characters

| Key             | Value |
|-----------------|-------|
| + Add attribute |       |

**Add** **Cancel**

在执行上述步骤后，云的设置完毕；可以使用设备并将其连接到 GCP。

## 5.4 应用程序编译和烧写

提示

使用任何工具链打开项目之前，请确保文件夹安装路径不太深，否则工具链可能会在编译之后报告错误。



使用所支持的某种开发工具链打开并编译项目（有关版本要求的详细信息，请参阅版本说明）。

在 STM32 板上编程固件：将 `Projects\<board name>\Applications\Cloud\GoogleIoT\Binary` 下的二进制文件复制（或拖动）到 STM32 板插入 PC 时所创建的 USB 大容量存储位置。或者，用户可以直接通过所支持的其中一个开发工具链对 STM32 板进行编程。

## 5.5 应用程序首次启动

1. 必须通过 USB（ST-LINK USB 端口）将板连接到 PC。通过串行终端仿真器（如 Tera Term）打开控制台，选择板子的 ST-LINK COM 端口，并进行以下配置：
  - 8 位数据长度，无校验，1 位停止位，115200 波特率，无硬件流控制
  - 行结束符设置为 LF
  - “echo on”选项，以便能够读取用户输入的参数
  - 此外，建议使能“自动模式”选项（如果有）

有关详细信息，请参见第 6 节 与板交互。

对于支持 Wi-Fi® 的板，通过控制台输入 Wi-Fi® SSID、加密模式和密码。对于蜂窝网络连接，输入选中操作员提供的接入点和密码。

2. 系统提示用户输入连接字符串，其中包含与使用 GCP 用户帐户创建的项目和注册设备相关的必要信息。

输入设备的 Google Cloud IoT Core 连接字符串：（模板：project-id=xxx;registry-id=xxx;device-id=xxx;cloud-region=xxx）

连接字符串的示例：

```
project-id=my-project-123456;registry-id=stm32registry;device-id=mystm32device;cloud-region=us-central1
```

3. 设置 TLS 根 CA 证书：  
复制粘贴 `Projects\Common\GoogleIoT\comodo_google.pem` 的内容。该设备使用它通过 TLS 对远程主机进行身份验证。

注意：示例应用程序要求提供 2 个 CA 证书的连结。

- a. 用于 HTTPS 服务器，启动时通过该服务器检索当前时间和日期（“Comodo”证书）
- b. 用于 GCP，以便验证云服务器。根据服务器，可能需要按照 Google Cloud™ 的支持 CA 列表更新 `comodo_google.pem`（可在 [pki.google.com/roots.pem](https://pki.google.com/roots.pem) 中找到）。

4. 设置设备私钥  
复制粘贴在 GCP 控制台上的设备创建步骤中创建的公钥所对应的私钥 `rsa_private.pem` 或 `ec_private.pem` 的内容。
5. 参数配置完成后，如果需要重新更改参数，可以根据提示通过重启板并按下用户按钮（蓝色按钮）来进行更改。

## 5.6 应用程序运行时

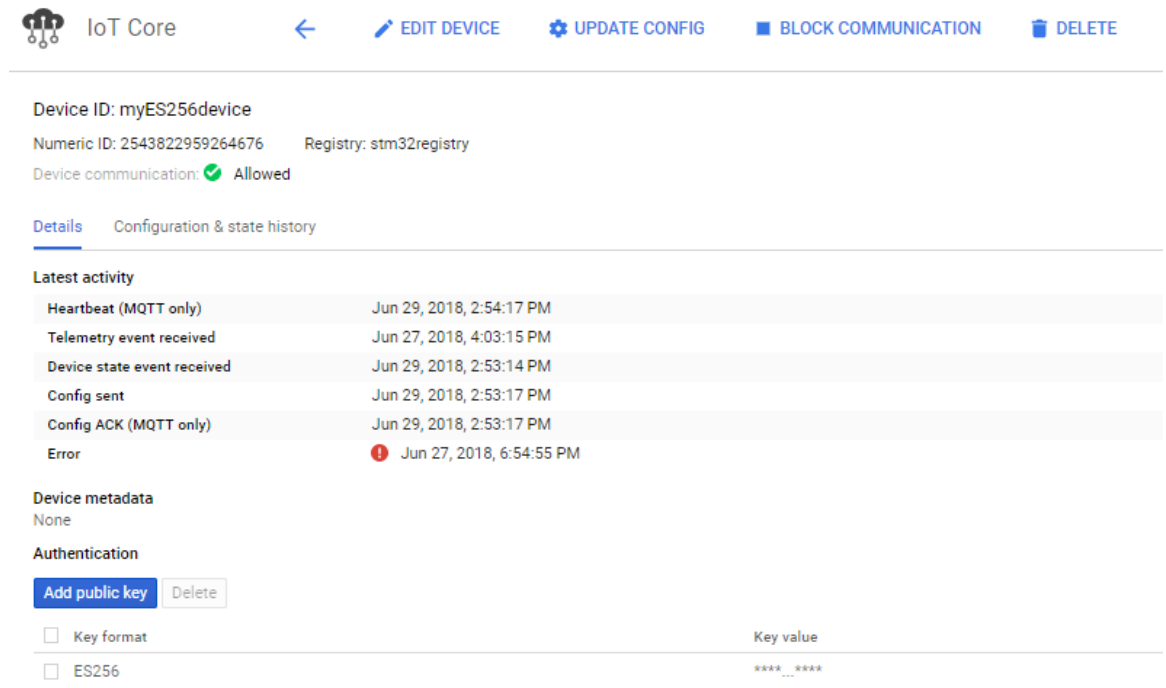
1. RTC 配置  
应用程序发出 HTTPS 请求来检索当前时间和日期，并配置 RTC。



## 2. 设备到云的连接

在正确配置设置后，设备连接到 GCP IoT core。为了核实连接是否正确，用户可以从 GCP IoT core 控制台检查设备详情，如图 14. IoT core 设备控制台所示。

图 14. IoT core 设备控制台



从同一控制台菜单，用户可以：

- 检查最新的错误状态。当设备拔出和与云端的连接中断时，一种典型的错误是“*mqtt: 服务器: MQTT 保持连接检查失败导致连接断开*”。
- 在相应菜单中检查配置和状态历史。系统提供最新发布的设备状态和配置供用户参考。

## 3. 发布的遥测数据

如果板子为 B-L475E-IOT01，则应用可发布传感器值、0/1 切换值（绿色 LED 相应切换）以及时间戳。

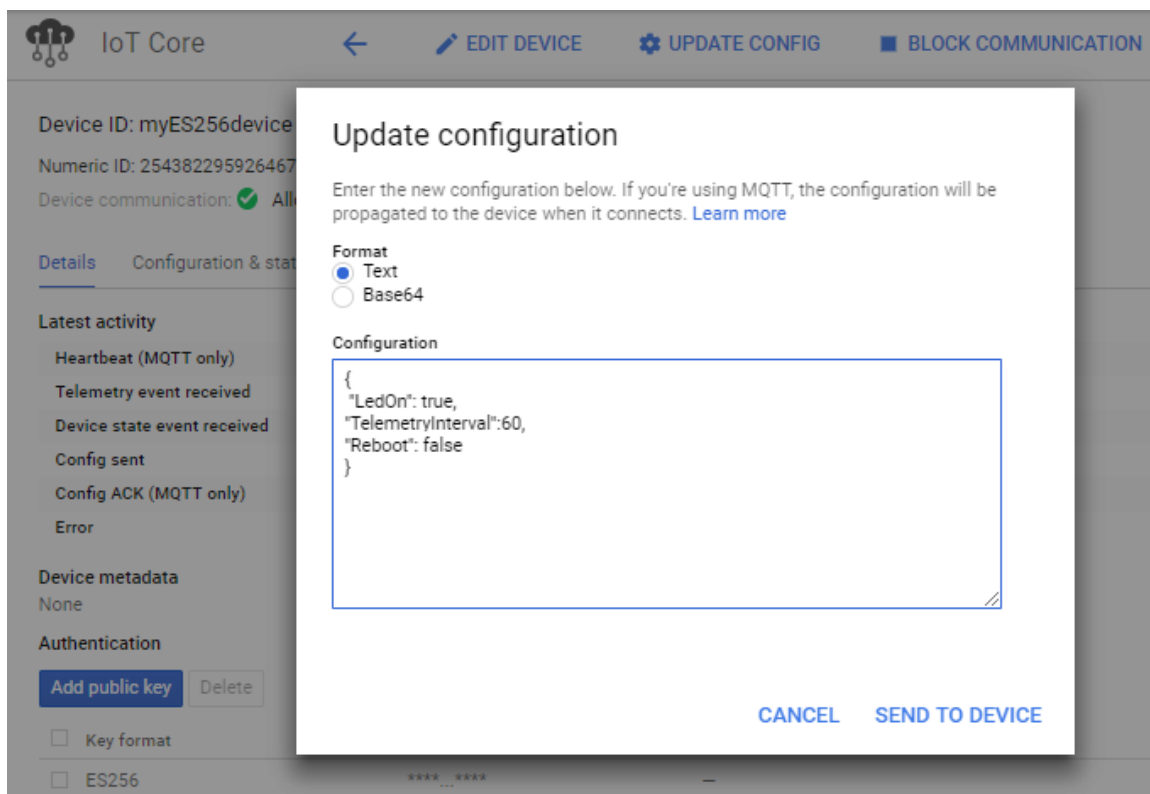
- 短暂按下用户按钮一次后
- 按用户定义的固定遥测间隔（按下用户按钮两次）。遥测间隔可通过设备配置进行修改。

上一步“设备到云的连接”中的 IoT Core Device 面板显示了最新接收的遥测数据的日期。GCP 不提供对数据的即时访问。用户必须依赖其他 GCP 服务或使用 CLOUD SDK 工具套件将数据可视化。

#### 4. 云到设备的连接

可以在 GCP 控制台上使用“更新配置”菜单进行此类交互，将以简单 JSON 格式编码的命令发送至设备（请参考图 15. IoT core 设备配置）。如果设备没有连接，将在下一次连接时接收配置。

图 15. IoT core 设备配置



注意：X-CUBE-GCP 在每次连接时对配置进行注册，GCP IoT Core 服务总是在每次重新注册时发送最新配置。因此，如果在配置中发送了重启命令“Reboot”: true，用户必须将配置重新更新为“Reboot”: false，以免在每次重新连接时发生无限重启。

## 5.7 仪表板和绘图值

GCP 中没有默认的仪表板用于数据可视化。用户需通过额外的 GCP 服务创建网络应用。或者，可以安装 Google Cloud™ 的 GCloud SDK，并使用工具套件订阅 Pub/Sub 主题。

请参考 GCloud SDK 文档和 gcloud pubsub 命令，如：\$ gcloud pubsub subscriptions pull

## 6 与板交互

需要串行终端来：

- 对板子进行配置
- 监测应用状态
- 在本地显示云到设备消息

本文档中使用 **Tera Term** 进行说明。也可以使用其他类似的工具。

在首次使用板时，必须使用连接字符串数据进行编程：

- 确定 PC 上所用的用于探索板的 **STM32 ST-LINK** 虚拟 COM 端口。在一台 Windows® PC 上，打开设备管理器。
- 打开 PC 上的串行终端并将其连接到上述虚拟 COM 端口。

在开发包实用程序目录中提供了一个 **Tera Term** 初始化脚本（参考图 4. 项目文件结构）；此脚本设置了正确的参数。要使用它，请打开 **Tera Term**，选择 **Setup**，然后选择 **Restore** 设置。

提示

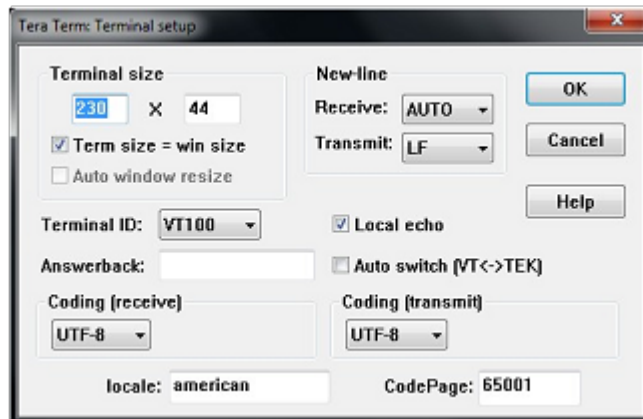
下面提供的信息可用来配置 **UART** 终端，使其成为 **Tera Term** 初始化脚本的替代选择。

### 终端设置

终端设置如图 16 所示。下面的截屏显示了终端设置和新行的推荐参数。

串行终端新行传输配置必须设置为 **LineFeed**（\n 或 LF），以便允许从 **UNIX**®类型的文本文件复制粘贴。选择 **本地回显**选项将使复制粘贴结果显示在控制台上。

图 16. 终端设置



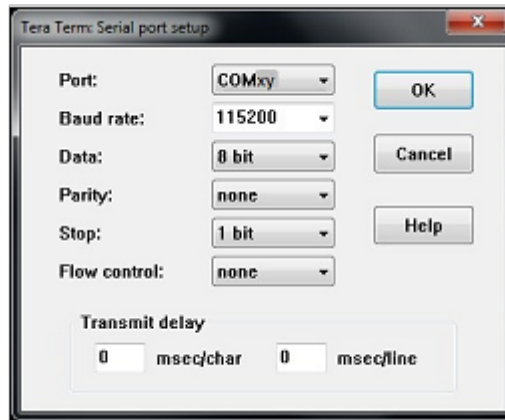
### 串口设置

串口必须配置：

- 实际的 COM 端口号
- 115200 波特率
- 8 位数据
- 无校验位
- 1 个停止位
- 无流控

串口设置如图 17 中所示。

图 17. 串口设置

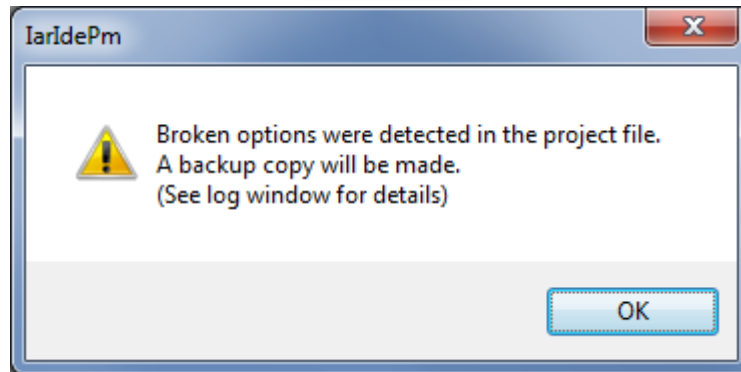


UART 终端和串口设置完毕后，按下板复位按钮（黑色）。按照 UART 终端上的指示上传 Wi-Fi®和云配置数据。这些数据保留在 Flash 存储器中，并在下次板启动时重复使用。

## 7 常见问题

问：当我用 IAR™ 打开项目时，为什么会弹出此提示框？

图 18. 当 IAR™ IDE 版本与 X-CUBE-GCP 所用的版本不兼容时会弹出此提示框



答：很可能是由于此 IAR™ IDE 版本比开发包所用版本更早（请参阅软件包根文件夹中提供的版本说明，以了解支持的 IDE 版本），因此不能保证兼容性。这种情况下，需要更新 IAR™ IDE 版本。

问：应该如何修改应用程序来发布其他消息？

答：根据是使用 B-L475E-IOT01A 还是其他板，`googleiot.c` 文件中需要分别更新函数 `GoogleIoT_publishTelemetry ()` 或 `GoogleIoT_publishDeviceState ()`。

问：我的设备不能连接到 GCP。我该如何处理？

答：需要检查的事项是：

1. 确认连接字符串中的 `project-id`、`registry-id` 和 `device-id` 均正确拼写并提交。GCP 通常创建唯一指定的 `project-id`，它包含自动生成的名称和/或编号，例如 `<google-defined-project-123456>`。因此，务必使用注册控制台上显示的 `project-id`。
2. 如果不成功，则必须再次仔细进行初始配置。
3. 如果仍然没有连接，请在 GCP 控制台上检查并确认为设备定义的设备公钥是使用设备上保存的对应的正确私钥设置的。

问：我的设备不能连接到 Wi-Fi® 接入点。我该如何处理？

答：请确认其他设备是否可以连接到 Wi-Fi® 接入点。如果可以，请在板重置后，按下用户按钮（蓝色）五秒钟，输入 Wi-Fi® 凭证。

问：在 B-L475E-IOT01A 板上，接近传感器总是报告“8190”，即使我将障碍物靠近其放置也是如此。

答：厂家交付的接近传感器贴有临时保护膜。在使用传感器前，确保已将膜去掉。保护膜为橙色，几乎不可见。在 B-L475E-IOT01A 上，接近传感器位于板前侧焊接了 MCU 的左下角附近。



版本历史

表 3. 文档版本历史

| 日期             | 版本 | 变更    |
|----------------|----|-------|
| 2018 年 9 月 6 日 | 1  | 初始版本。 |



## 目录

|             |                             |           |
|-------------|-----------------------------|-----------|
| <b>1</b>    | 概述.....                     | <b>2</b>  |
| <b>2</b>    | Google Cloud Platform ..... | <b>3</b>  |
| <b>3</b>    | 软件包说明 .....                 | <b>5</b>  |
| <b>3.1</b>  | 概述 .....                    | <b>5</b>  |
| <b>3.2</b>  | 架构 .....                    | <b>5</b>  |
| <b>3.3</b>  | 文件夹结构 .....                 | <b>8</b>  |
| <b>3.4</b>  | B-L475E-IOT01A 板传感器 .....   | <b>10</b> |
| <b>3.5</b>  | Wi-Fi 组件.....               | <b>10</b> |
| <b>3.6</b>  | 复位按钮 .....                  | <b>10</b> |
| <b>3.7</b>  | 用户按钮 .....                  | <b>10</b> |
| <b>3.8</b>  | 用户 LED.....                 | <b>10</b> |
| <b>3.9</b>  | 实时时钟 .....                  | <b>10</b> |
| <b>3.10</b> | mbedTLS 配置.....             | <b>11</b> |
| <b>4</b>    | 硬件和软件环境设置 .....             | <b>12</b> |
| <b>5</b>    | 应用程序示例 .....                | <b>15</b> |
| <b>5.1</b>  | 应用描述 .....                  | <b>15</b> |
| <b>5.2</b>  | GCP 和 IoT Core 帐户设置.....    | <b>18</b> |
| <b>5.3</b>  | 在 GCP IoT core 上创建设备.....   | <b>19</b> |
| <b>5.4</b>  | 应用程序编译和烧写.....              | <b>23</b> |
| <b>5.5</b>  | 应用程序首次启动 .....              | <b>24</b> |
| <b>5.6</b>  | 应用程序运行时 .....               | <b>24</b> |
| <b>5.7</b>  | 仪表板和绘图值 .....               | <b>26</b> |
| <b>6</b>    | 与板交互.....                   | <b>27</b> |
| <b>7</b>    | 常见问题.....                   | <b>29</b> |
|             | Revision history.....       | <b>30</b> |



表一览

|      |                                |    |
|------|--------------------------------|----|
| 表 1. | 缩略语列表 .....                    | 2  |
| 表 2. | B-L475E-IOT01A 板传感器报告值的单位..... | 10 |
| 表 3. | 文档版本历史 .....                   | 30 |



## 图一览

|       |  |    |
|-------|--|----|
| 图 1.  | Google Cloud IoT Core 生态系统 .....                 | 3  |
| 图 2.  | Google Cloud IoT Core 图 .....                    | 3  |
| 图 3.  | X-CUBE-GCP 软件架构 .....                            | 7  |
| 图 4.  | 项目文件结构 .....                                     | 9  |
| 图 5.  | 硬件和软件设置环境 .....                                  | 13 |
| 图 6.  | 应用状态机 .....                                      | 17 |
| 图 7.  | 需要计费 .....                                       | 18 |
| 图 8.  | 使能 API 菜单 .....                                  | 18 |
| 图 9.  | 使能 Google Cloud IoT API .....                    | 19 |
| 图 10. | 使能 Cloud Pub/Sub API .....                       | 19 |
| 图 11. | GCP“创建注册表”按钮 .....                               | 20 |
| 图 12. | GCP“创建注册表”设置 .....                               | 21 |
| 图 13. | GCP“创建设备”设置 .....                                | 23 |
| 图 14. | IoT core 设备控制台 .....                             | 25 |
| 图 15. | IoT core 设备配置 .....                              | 26 |
| 图 16. | 终端设置 .....                                       | 27 |
| 图 17. | 串口设置 .....                                       | 28 |
| 图 18. | 当 IAR™ IDE 版本与 X-CUBE-GCP 所用的版本不兼容时会弹出此提示框 ..... | 29 |



重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 标志是 ST 的商标。关于 ST 商标的其他信息，请访问 [www.st.com/trademarks](http://www.st.com/trademarks)。其他所有产品或服务名称是其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2019 STMicroelectronics - 保留所有权利