

**QuecPython DTU Software Design Guide**

**LTE Standard/LPWA Module Series**

Version: 1.0.0

Date: 2023-02-03

 Status: Preliminary

**At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:**

**Quectel Wireless Solutions Co., Ltd.**

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto:info@quectel.com)

**Or our local offices. For more information, please visit:**

<http://www.quectel.com/support/sales.htm>.

**For technical support, or to report documentation errors, please visit:**

<http://www.quectel.com/support/technical.htm>.

Or email us at: [support@quectel.com](mailto:support@quectel.com).

**Legal Notices**

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

**Use and Disclosure Restrictions**

**License Agreements**

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

**Copyright**

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

**Trademarks**

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

**Third-Party Rights**

This document may refer to hardware, software and/or documentation owned by one or more third parties (“third-party materials”). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

**Privacy Policy**

To implement module functionality, certain device data are uploaded to Quectel’s or third-party’s servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

**Disclaimer**

1. We acknowledge no liability for any injury or damage arising from the reliance upon the information.
2. We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
3. While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
4. We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

***Copyright © Quectel Wireless Solutions Co., Ltd. 2023. All rights reserved.***

# About the Document

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| - | 2023-02-03 | Elian WANG | Creation of the document |
| 1.0.0 | 2023-02-03 | Elian WANG | Preliminary |

# Contents

[About the Document 3](#_Toc126423203)

[Contents 4](#_Toc126423204)

[Table Index 5](#_Toc126423205)

[Figure Index 6](#_Toc126423206)

[1 Introduction 7](#_Toc126423207)

[1.1. Applicable Modules 7](#_Toc126423208)

[2 System Framework 8](#_Toc126423209)

[2.1. Hardware Framework 8](#_Toc126423210)

[2.2. Software Framework 9](#_Toc126423211)

[3 Key Components 10](#_Toc126423212)

[3.1. Data Downlink Business Component (DownlinkTransaction) 10](#_Toc126423213)

[3.2. OTA Business Component (OtaTransaction) 11](#_Toc126423214)

[3.3. Data Uplink Business Component (UplinkTransaction) 12](#_Toc126423215)

[3.4. Alibaba Cloud IoT Platform (aliyunIot) 13](#_Toc126423216)

[3.5. QuecCloud (quecthing) 15](#_Toc126423217)

[3.6. TCP/UDP Communication Cloud Interface (Socket) 17](#_Toc126423218)

[3.7. Historical File Function Components (history) 17](#_Toc126423219)

[4 System Initialization Process 18](#_Toc126423220)

[5 Business Process 19](#_Toc126423221)

[6 Example 21](#_Toc126423222)

[7 Appendix Terms and Abbreviations 23](#_Toc126423223)

# Table Index

[Table 1: Applicable Modules 7](#_Toc126423224)

[Table 2: Terms and Abbreviations 23](#_Toc126423225)

# Figure Index

[Figure 1: Hardware Framework 8](#_Toc126423226)

[Figure 2: Software Framework 9](#_Toc126423227)

[Figure 3: Data Downlink Business Function Implementation 11](#_Toc126423228)

[Figure 4: System Initialization Flowchart 18](#_Toc126423229)

[Figure 5: Data Downlink Business Processing Thread 19](#_Toc126423230)

[Figure 6: Data Uplink Business Processing Thread 20](#_Toc126423231)

[Figure 7: OTA Upgrade Process 20](#_Toc126423232)

1. Introduction

This document describes the design framework of QuecPython DTU, including the software and hardware framework, function descriptions of key components, system initialization process, business process and function examples. It helps users quickly understand the overall architecture and functions of DTU.

* 1. Applicable Modules

Table 1: Applicable Modules

|  |  |
| --- | --- |
| **Module Series** | **Module** |
| LTE Standard | EC21-AUX |
| EC200U Series |
| EG915U Series |
| EC200A Series |
| NB-IoT | BC65 |
| LPWA | BG95 Series |
| BG77 |

1. System Framework
2. Hardware Framework

The hardware system framework is as follows:

1. The module supports functions such as UART, GPIO and LTE.
2. The module communicates with MCU through UART port.
3. The module can communicate with cloud platform with MQTT/TCP protocols. Currently the supported cloud platforms are QuecCloud, Alibaba Cloud, Tencent Cloud, Huawei Cloud and other private cloud platform supported by MQTT/TCP.

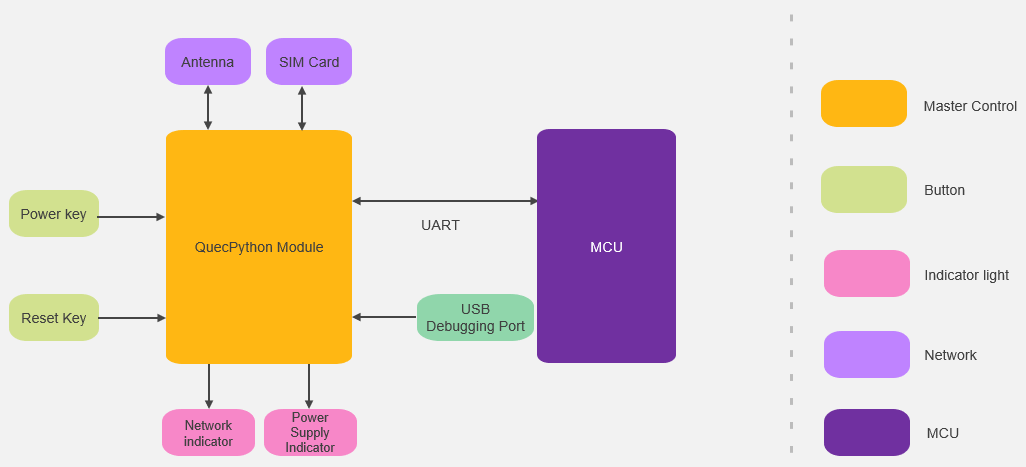


Figure 1: Hardware Framework

1. Software Framework



Figure 2: Software Framework

1. Key Components
2. Data Downlink Business Component (DownlinkTransaction)
3. Function description: obtain cloud data, and pack the data into a specific format to transmit it to external device (MCU) through the UART port.
4. Principle:
5. When the business initializes, the DownlinkTransaction module is registered as the RemoteSubscribe executor. When the QuecPython module receives the transparent transmission data from the cloud, it calls *downlink\_main()* of the DownlinkTransaction module to process the cloud data.

|  |
| --- |
| class RemoteSubscribe(CloudObserver):      """This class is for distribute cloud downlink messages"""      def \_\_init\_\_(self):          self.\_\_executor = None          self.\_\_ota\_executor = None      def \_\_raw\_data(self, \*args, \*\*kwargs):          """Handle cloud transparent data transmission."""          return self.\_\_executor.downlink\_main(\*args, \*\*kwargs) if self.\_\_executor else False |

1. *downlink\_main()* processes the data received by the module and send it out through the UART port.

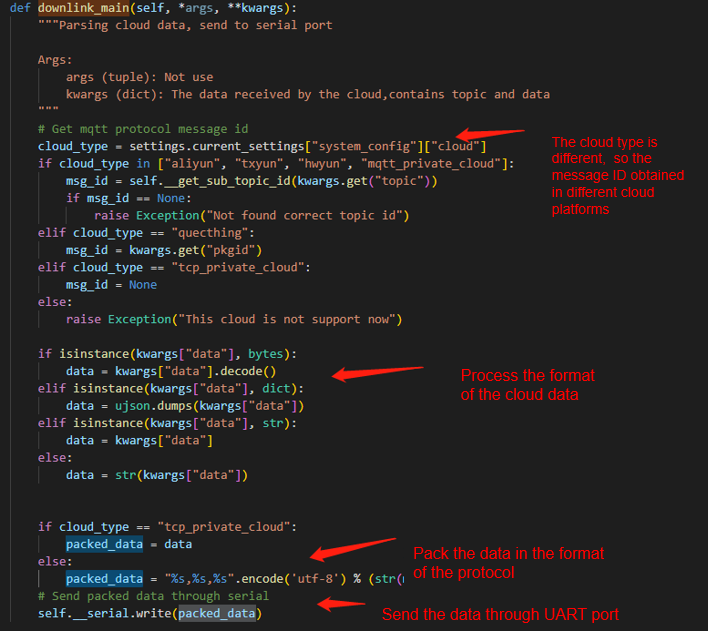
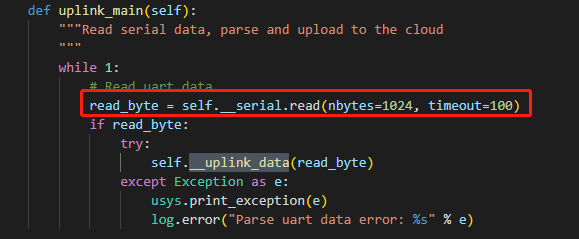


Figure 3: Data Downlink Business Function Implementation

1. OTA Business Component (OtaTransaction)
2. Function description: process the OTA upgrade plan issued by the cloud, interact with the cloud, and complete the OTA upgrade process.
3. Principle:

When the business initializes, the OtaTransaction module is registered as the RemoteSubscribe executor. When the QuecPython module receives the transmission data from the cloud, it calls *event\_ota\_plain()* of the OtaTransaction module to process the cloud data.

1. Data Uplink Business Component (UplinkTransaction)
2. Function description: parse UART port data, obtain transparent transmission data and send it to the cloud.
3. Principle:
4. Read UART port data.



1. Parse the UART port data read. Due to the different protocols used for communication between the QuecPython module and the cloud, the protocols used for communication between the QuecPython module and the external device (such as MCU) are also different. When the module and the cloud communicate with TCP protocol, the data is transmitted directly because the TCP and the serial port are in the form of data flow. At this time, no operation is required; when the module and the cloud communicate with MQTT protocol, in order to distinguish different data frames, the serial protocol of the module uses simple data frames with the flowing format:

*<topic\_id>,<msg\_len>,<msg\_data>*

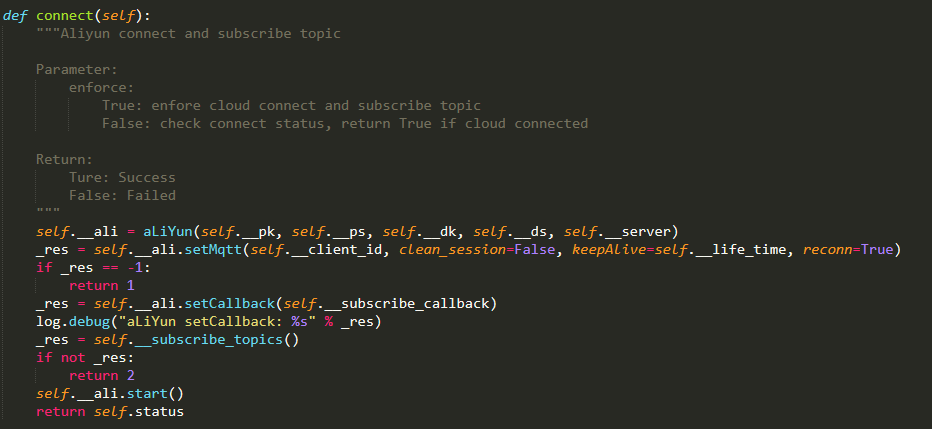
An example data frame is as follows:

|  |
| --- |
| 1,6,abcedf //Uplink data frame  1,6,ijklmn //Downlink data frame |

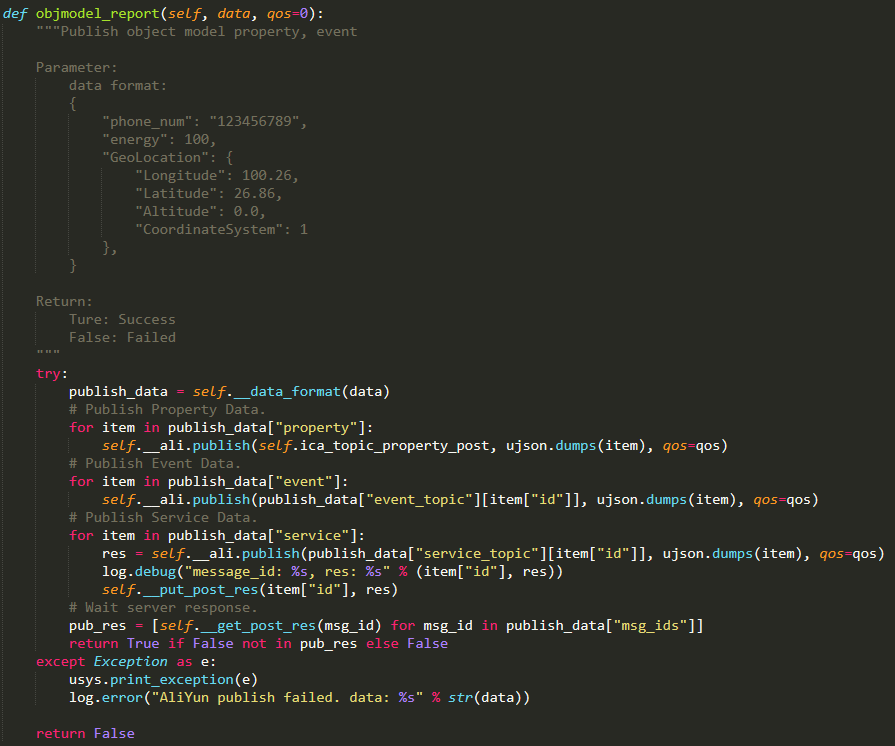
电脑屏幕的截图

描述已自动生成

1. Alibaba Cloud IoT Platform (aliyunIot)
2. Function description: interact with Alibaba Cloud IoT platform through MQTT protocol.
3. Register on Alibaba Cloud IoT platform and log in to the platform.
4. Send TSL data to Alibaba Cloud IoT platform.
5. Receive commands sent from Alibaba Cloud IoT platform.
6. Perform OTA upgrade.
7. Principle: The QuecPython module logs in to Alibaba Cloud IoT platform and interacts with the platform according to the platform communication rules through MQTT.
8. Register on Alibaba Cloud IoT platform and log in to the platform.



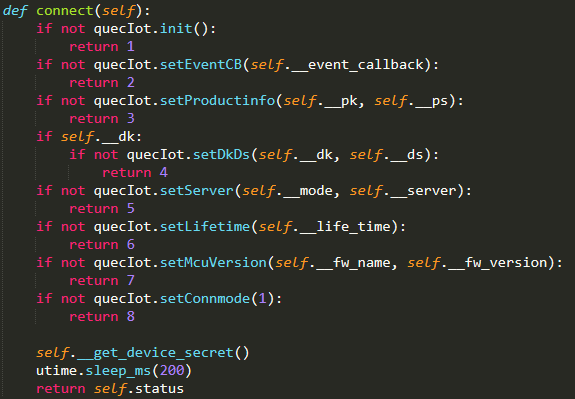
1. Send data to Alibaba Cloud IoT platform.



1. Receive data sent from Alibaba Cloud IoT platform and process business.



1. QuecCloud (quecthing)
2. Function description: The device interacts with QuecCloud IoT platform through MQTT protocol.
3. Register on QuecCloud IoT platform and log in to the platform.
4. Send TSL data to Alibaba Cloud IoT platform.
5. Receive commands sent from Alibaba Cloud IoT platform.
6. Perform OTA upgrade.
7. Principle: Encapsulate the QuecThing functions again to use the functions such as login, message sending and downlink command callback notification.
8. Register on QuecCloud IoT platform and log in to the platform.



1. Report TSL data and positioning information.

屏幕上有字

描述已自动生成

1. Receive data sent from QuecCloud IoT platform and process business.

文本

描述已自动生成

1. TCP/UDP Communication Cloud Interface (Socket)
2. Function description: Interact with private cloud through TCP/UDP
3. Establish TCP connection
4. Monitor TCP/UDP data
5. Send TCP/UDP messages
6. Principle: Encapsulate Socket function interface of QuecPython again to use functions such as initialization, message sending, and downlink command callback notification.
7. Historical File Function Components (history)
8. Function description: read, write and store the module data that fails to be reported.
9. Principle: If the module business data fails to be reported to the server, it is stored in the device. When the device reports data to the server the next time, it reads the stored data and try to report it to the server again.
10. System Initialization Process

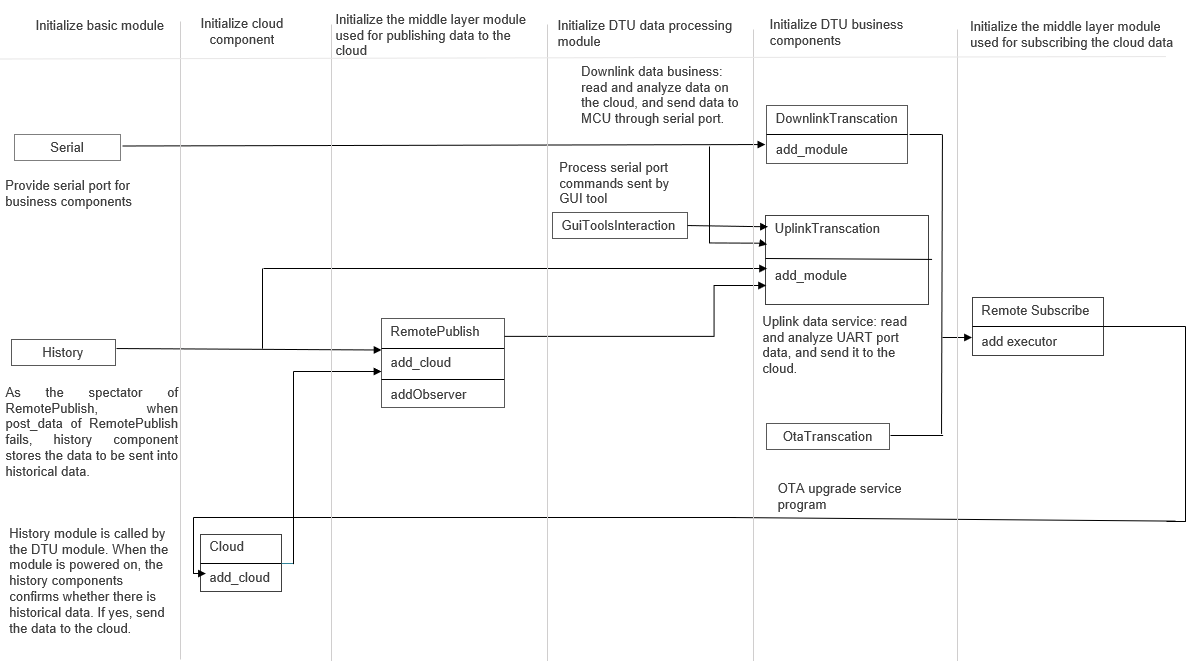


Figure 4: System Initialization Flowchart

1. Basic module initialization includes initializing the UART port, historical data storage and IoT platform.
2. Initialize the middle layer module (RemotePublish) used for publishing data to the cloud, and add IoT platform(s) for the middle layer module. When the data is sent to the cloud through RemotePublish, RemotePublish calls the registered cloud component (aliyunnIot, quecthing, Socket) to send the data.
3. Initialize DTU business components (OtaTransaction, UplinkTransaction, DownlinkTransaction), and add sub-modules for the business component.
4. Initialize the middle layer module (RemoteSubscribe) used for subscribing the cloud data and add the executors DownlinkTransaction and OtaTransaction. When the cloud data is received, the middle layer module parses the data and call the executor to complete the cloud data processing.
6. Business Process

There are three aspects in the business process: data uplink business, data downlink business and OTA upgrade business. In the data uplink business, the UART port data is periodically read, parsed and then is sent to the cloud platform. When the cloud platform receives the transparent transmission data, the module executes the data downlink business, processes the cloud data, and sends the data to the UART port.

Parse the cloud data

Process data

Event of receiving the cloud data

Data downlink business processing thread

Process OTA upgrade plan

Send data through UART port

Figure 5: Data Downlink Business Processing Thread

Receive UART port data

Parse UART port data

Process command data

Process transparent transmission data on the cloud

Send data to the cloud

Send data through UART port

Data uplink business processing thread

Figure 6: Data Uplink Business Processing Thread

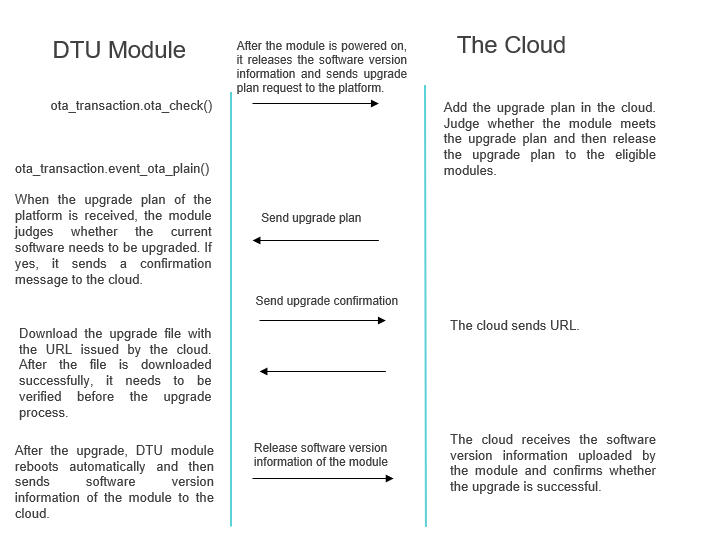


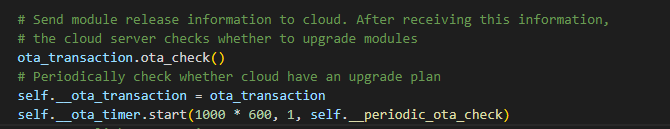
Figure 7: OTA Upgrade Process

2. Example
3. Initialize the DTU function module, and the DTU function module registers sub-module.

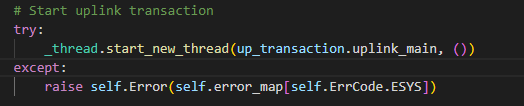
文本

描述已自动生成

1. After the module is powered on, detect the cloud OTA status once and check the OTA status periodically.



1. Create a thread to execute DTU data uplink business.



1. Appendix Terms and Abbreviations

Table 2: Terms and Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| DTU | Data Transfer Unit |
| UART | Universal Asynchronous Receiver/Transmitter |
| GPIO | General-Purpose Input/Output |
| LTE | Long-Term Evolution |
| MCU | Microcontroller Unit/Microprogrammed Control Unit |
| TCP | Transmission Control Protocol |
| MQTT | Message Queuing Telemetry Transport |
| SIM | Subscriber Identity Module |
| OTA | Over-the-air programming |
| UDP | User Datagram Protocol |