Document Number	QW_01_0003.002

GIoT ODU Series Gateway

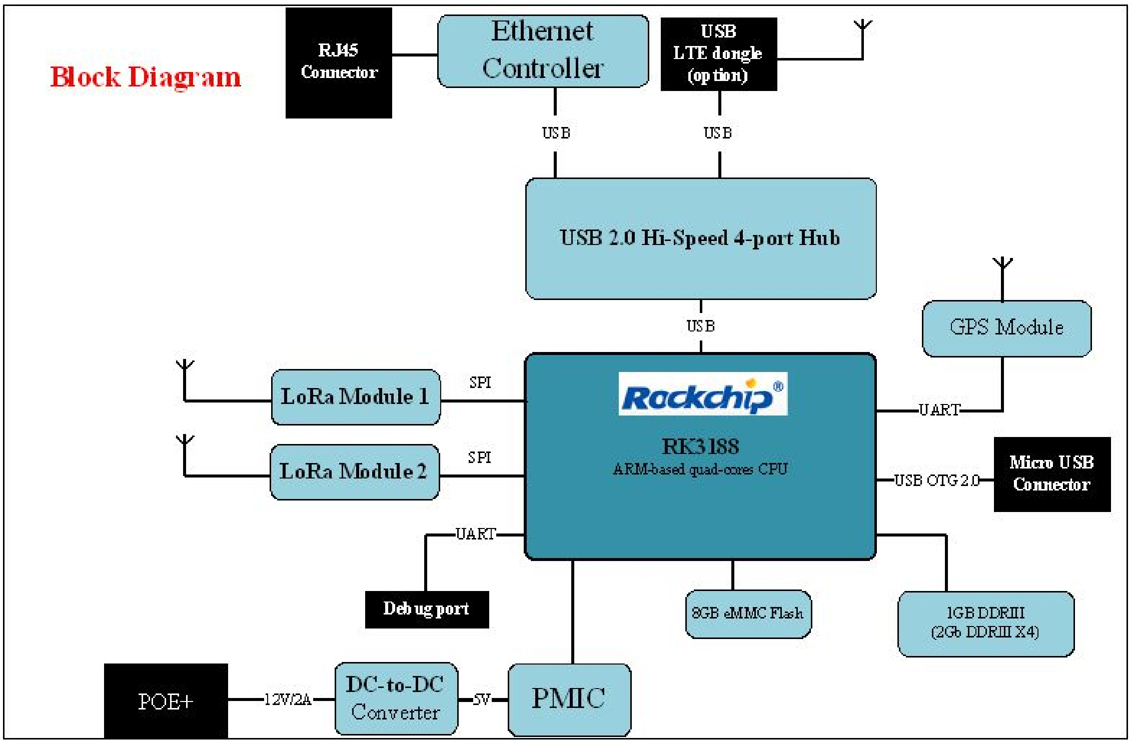
General Specification

Table of Contents

1. GIoT Gateway HW Specification

1.1 GIoT Gateway General HW Specification

GIoT 16 Channel ODU Gateway is for outdoor use. The HW specification is built with tenacious features to endure harsh environments, features as in a wider operating temperature range, waterproofed design and a surge-arrestor.



The GIoT Gateway HW Block Diagram

➢CPU :

RK3188

- ARM-based
- 4 Core
- Cortex-A9
- Frequency 1.6GHz

➢RAM:

1GB DDRIII

➢Flash:

8GB eMMC Flash

➢Power:

POE 55V, 30W

➢Ethernet:

10/100M Ethernet controller

➢GPS Module:

GlobalTop module

➢RTC:

Based on a 32.768 kHz quartz crystal

➢LoRa Module 1:

SX1301+ 2x SX1257 /SX1255

➢LoRa Module 2:

SX1301+ 2x SX1257/SX1255

➢3/4G Module (optional): Please view Appendix.

Multi-mode LTE Cat 4 Mini PCIe Module

EC20-CE	LTE BAND 1/3/8/38/39/40/41
EC25-E	LTE BAND 1/3/5/7/8/20/38/40/41
EC25-J	LTE BAND 1/3/8/18/19/26/41
EC25-AU	LTE BAND 1/2/3/4/5/7/8/28/40
EC25-A	LTE BAND 2/4/12

➢Operating Temperature:

-20℃~60℃

➢Dimensions:

L:230 x W:200 x H: 68mm

➢Certification:

TELEC

1.2 GIoT Gateway LoRa Module HW Specification

There are two LoRa modules in the GIot Gateway. Each of them supports 8 uplink channels with 125kHz bandwidth.

➢Main Chips:

- SX1301

The SX1301 digital baseband chip is a massive digital signal processing engine specifically designed to offer breakthrough gateway capabilities in the ISM bands worldwide. It integrates the LoRa concentrator IP.

The LoRa concentrator is a multi-channel high performance transmitter/receiver designed to simultaneously receive several LoRa packets using random spreading factors on random channels. Its goal is to enable robust connection between a central wireless data concentrator and a massive amount of wireless end-points spread over a very wide range of distances.

- SX1257/SX1255

The SX125X is a highly integrated RF front-end to digital I and Q modulator/demodulator Multi-PHY mode transceiver capable of supporting multiple constant and non-constant envelope modulation schemes. It is designed to operate over the 862 - 960 (optional 470~510) MHz European, North American and Japanese ISM (Industrial, Scientific and Medical) license-exempt frequency bands. Its highly integrated architecture allows for a minimum of external components whilst maintaining maximum design flexibility. All major RF communication parameters are programmable and most of them can be dynamically set. The SX125X offers support for both narrow-band and wide-band communication modes without the need to modify external components.

The SX125X is optimized for low power consumption while offering the provision for high RF output power and channelized operation. TrueRF™ technology enables a low-cost external component count whilst still satisfying ETSI, FCC and ARIB regulations.

➢Standards:

IEEE802.15.4g (Semtech LoRa SX1257/SX1255)

➢Frequency Band:

922~928 /902~928 / 862~870 / 470~510MHz (with different SKU)

➢Modulation Method:

LoRa® Technology Modulation

➢Number of Channel:

Up to 16 Concurrent channels (125kHz Bandwidth)

➢Antenna:

N-Type RF Connector

➢Security:

AES 128

- ➢Transmitted Power:

  0.5W(up to 27 dBm) , based on each country’s regulatory standards

- ➢Receive Sensitivity:

  -142 dBm

1.3 SKU

ODU Gateway has 13 SKUs via different HW/SW combination per country. Find the table as follows for SKU combination

SKU	Country	Channels	Frequency Band (MHz)	3G/4G Support
AU-16	Australia	16	AU920 (915~928)	N
AU-16-M	Australia	16	AU920 (915~928)	Y, EC25-AU
AU-8	Australia	8	AU920 (915~928)	N
AU-8-M	Australia	8	AU920 (915~928)	Y, EC25-AUY
CN-470-8	China	8	CN470 (470~510)	N
CN-470-16	China	16	CN470 (470~510)	N
CN-470-16-M	China	16	CN470 (470~510)	Y, EC20-CE
EU-8-M	Europe	8	EU868 (862~870)	Y, EC25-E
JP-16	Japan	16	920~928	N
JP-16-M	Japan	16	920~928	Y, EC25-J
MY-16	Malaysia	16	919~923	N

MY-16-M	Malaysia	16	919~923	Y, EC25-E
TW-16	Taiwan	16	920~925	N
TW-16-M	Taiwan	16	920~925	Y, EC25-E
US-16	USA	16	US915 (902~928)	N
US-16-M	USA	16	US915 (902~928)	Y, EC25-A

2.GIoT Gateway SW Specification

2.1 Linux Based

Operation system is Linux based. It can provide more extension packages for future requirements.

➢OS:

Ubuntu-14.04 desktop-amd64

➢Linux Kernel:

3.6.2

➢Database:

MySQL

➢Preinstalled Packages/Applications:

· OpenSSL
· MQTT
· CURL

2.2 Supported LoRaWAN Specification

2.2.1 LoRaWAN Basic Introduction

LoRa™ is a wireless modulation for long-range low-power low-data-rate applications developed by Semtech. LoRaWAN™ network protocol which is optimized for battery-powered end-devices that may be either mobile or mounted at a fixed location. LoRaWAN networks typically are laid out in a star-of-stars topology in which gateways relay messages between end-devices and a central network server at the backend. Gateways are connected to the network server via standard IP connections while end-devices use single-hop LoRa™ or FSK communication to one or many gateways. All communication is generally bi-directional, although uplink communication from an end-device to the network server is expected to be the predominant traffic.

Communication between end-devices and gateways is spread out on different frequency channels and data rates. LoRa data rates range from 0.3 kbps to 50 kbps . The LoRa network infrastructure can manage the data rate and RF output for each end-device individually by means of an adaptive data rate (ADR) scheme.

End-devices may transmit on any channel available at any time, using any available data rate, as long as the following rules are respected:

- The end-device changes channel in a pseudo-random fashion for every transmission. The resulting frequency diversity makes the system more robust to interferences.
- The end-device respects the maximum transmit duty cycle relative to the sub-band used and local regulations.
- The end-device respects the maximum transmit duration (or dwell time) relative to the sub-band used and local regulations.

LoRa network allows the end-devices to individually use any of the possible data rates. This feature is used by the LoRaWAN to adapt and optimize the data rate of static end-devices. This is referred to as Adaptive Data Rate (ADR) and when this is enabled the network will be optimized to use the fastest data rate possible.
There are three classes:

- **Bi-directional end-devices (Class A):**
  End-devices of Class A allow for bi-directional communications whereby each end-device’s uplink transmission is followed by two short downlink receive windows. The transmission slot scheduled by the end-device is based on its own communication needs with a small variation based on a random time basis (ALOHA-type of protocol). This Class A operation is the lowest power end-device system for applications that only require downlink communication from the server shortly after the end-device has sent an uplink transmission. Downlink communications from the server at any other time will have to wait until the next scheduled uplink.
- **Bi-directional end-devices with scheduled receive slots (Class B):**
  End-devices of Class B allow for more receive slots. In addition to the Class A random receive windows, Class B devices open extra receive windows at scheduled times. In order for the End-device to open it receive window at the scheduled time it receives a time synchronized Beacon from the gateway. This allows the server to know when the end-device is listening.
- **Bi-directional end-devices with maximal receive slots (Class C):**
  End-devices of Class C have nearly continuously open receive windows, only closed when transmitting. Class C end-device will use more power to operate than Class A or Class B but they offer the lowest latency for server to end-device communication.

All LoRaWAN devices implement at least the Class A functionality. In addition they may implement options named Class B, Class C. In all cases, they must remain compatible with Class A.

2.2.2 GIoT Gateway Supported LoRaWAN Classes

GIoT Gateway currently supports Class A and Class C. These two classes can meet most of IoT Node application cases. Class B needs precise clock timing for each time slot.

2.2.3 MAC Commands

For network administration, a set of MAC commands may be exchanged exclusively between the network server and the MAC layer on an end-device. MAC layer commands are never visible to the application or the application server or the application running on the end-device.

A MAC command consists of a command identifier (**CID**) of 1 octet followed by a possibly empty command-specific sequence of octets 15

CID	Command	Transmitted by		Short Description
		End-device	Gateway	
0x02	LinkCheckReq	x		Used by an end-device to validate its connectivity to a network.
0x02	LinkCheckAns		x	Answer to LinkCheckReq command. Contains the received signal power estimation indicating to the end-device the quality of reception (link margin).
0x03	LinkADRReq		x	Requests the end-device to change data rate, transmit power, repetition rate or channel.
0x03	LinkADRAns	x		Acknowledges the LinkRateReq.
0x04	DutyCycleReq		x	Sets the maximum aggregated transmit duty-cycle of a device
0x04	DutyCycleAns	x		Acknowledges a DutyCycleReq command
0x05	RXParamSetupReq		x	Sets the reception slots parameters
0x05	RXParamSetupAns	x		Acknowledges a RXSetupReq command
0x06	DevStatusReq		x	Requests the status of the end-device
0x06	DevStatusAns	x		Returns the status of the end-device, namely its battery level and its demodulation margin
0x07	NewChannelReq		x	Creates or modifies the definition of a radio channel
0x07	NewChannelAns	x		Acknowledges a NewChannelReq command
0x08	RXTimingSetupReq		x	Sets the timing of the of the reception slots
0x08	RXTimingSetupAns	x		Acknowledges RXTimingSetupReq command

2.2.4 Adaptive data rate control

LoRa network allows the end-devices to individually use any of the possible data rates. This feature is used by the LoRaWAN to adapt and optimize the data rate of static end-devices. This is referred to as Adaptive Data Rate (ADR) and when this is enabled the network will be optimized to use the fastest data rate possible.

Adaptive Data Rate control may not be possible when the radio channel attenuation changes fast and constantly. When the network is unable to control the data rate of a device , the device's application layer should control it. It is recommended to use a variety of different data rates in this case. The application layer should always try to minimize the aggregated air time used given the network conditions.

If the ADR bit is set, the network will control the data rate of the end-device through the appropriate MAC commands. If the ADR bit is not set, the network will not attempt to control the data rate of the end-device regardless of the received signal quality. The ADR bit may be set and unset by the end-device or the Network on demand. However, whenever possible, the ADR scheme should be enabled to increase the battery life of the end-device and maximize the network capacity.

2.2.5 End-Device Activation

To participate in a LoRaWAN network, each end-device has to be personalized and activated.

After activation, the following information is stored in the end-device: a device address (DevAddr), an application identifier (AppEUI), a network session key (NwkSKey), and an application session key (AppSKey).

2.2.5.1 Over-the-Air Activation

For over-the-air activation, end-devices must follow a join procedure prior to participating in data exchanges with the network server. An end-device has to go through a new join procedure every time it has lost the session context information.

The join procedure requires the end-device to be personalized with the following information before its starts the join procedure: a globally unique end-device identifier (DevEUI), the application identifier (AppEUI), and an AES-128 key (AppKey).

From an end-device's point of view, the join procedure consists of two MAC messages exchanged with the server, namely a join request and a join accept.

2.2.5.2 Activation by Personalization

Under certain circumstances, end-devices can be activated by personalization. Activation by personalization directly ties an end-device to a specific network by-passing the join request - join accept procedure.

Activating an end-device by personalization means that the DevAddr and the two session keys NwkSKey and AppSKey are directly stored into the end-device instead of the DevEUI, AppEUI and the AppKey. The end-device is equipped with the required information for participating in a specific LoRa network when started.

Each device should have a unique set of NwkSKey and AppSKey. Compromising the keys of one device shouldn't compromise the security of the communications of other devices. The process to build those keys should be such that the keys cannot be derived in any way from publicly available information (like the node address for example).

2.2.6 Physical Layer ISM Band

LoRaWAN 1.0 defines several ISB Bands for different countries and regions.

- EU 863-870MHz ISM Band
- US 902-928MHz ISM Band
- China 779-787MHz ISM Band
- EU 433MHz ISM Band
- Australia 915-928MHz ISM Band
- CN 470-510MHz Band

The main difference is among the frequency range. Only the US and Australia have explicit channel definitions. The other countries and regions only define the reserved channels for the OTAA method. The data rate of the downlink table is also different based on the country and region. However, all ISM Bands support MAC Commands to facilitate network administration and elevate the overall performance of the LoRa Network.

2.3 GIoT Gateway SW Architecture

GIOT Gateway supports LoRaWAN 1.0 MAC commands for network administration. To participate in a LoRaWAN network, each end-device has to be personalized and activated. Activation of an end-device can be achieved in two ways, either via Over-The-Air Activation (OTAA) when an end-device is deployed or reset, or via Activation By Personalization (ABP) in which the two steps of end-device personalization and activation are done as one step. GIOT Gateway supports both ABP and OTAA end-device activation methods.

The GIoT Gateway software architecture provides various GNMS interfaces to monitor and configure the GIoT Gateways.

2.4 Hardware Abstraction Layer

There are several HAL drivers to support the HW components, which include the USB, UART, SPI and I2C. The following section introduces the details of each GIoT Gateway module.

2.4.1 GPS

The GPS module on the board is GlobalTop. The GlobalTop GPS module utilizes the MediaTek new generation GPS Chipset that achieves the industry's highest level of sensitivity (-165dBm) and instant Time-to-First Fix (TTFF) with lowest power consumption for precise GPS signal processing to give the ultra-precise positioning under low receptive, high velocity conditions The GPS is used to report the location of its modules to the GIoT GNMS system, in which would show the map location of each GIoT Gateway. Users can have a quick view of the deployment location of their GIoT Gateways.

2.4.2 Ethernet

The Ethernet component on the board is 10/100M Ethernet controller. The Realtek 10/100M Ethernet controller combines an IEEE 802.3u compliant Media Access Controller (MAC), USB bus controller, and embedded memory. The device also features USB 2.0 technology. It provides higher bandwidth and improved protocols for data exchange between the host and the device. In addition, USB 2.0 offers a more aggressive power management feature that enables selective suspend to save energy. The Uplink posting and Downlink receiving depend on the Ethernet port. The status report, alarm and configuration from GNMS also depend on the Ethernet port.

2.4.3 LoRa Module

There are two mini PCI slots for LoRa modules. The LoRa module includes one SX1301 and SX1257/SX1255. The interface between the LoRa module and the main board is the SPI interface. The LoRa concentrator library relies on libmpsse. The package must be installed in the GIoT Gateway during development.

The LoRa RF signals are sent and received by the SX1257. The LoRa RF packets are assembled/disassembled by the SX1301.

There is a libloragw for the LoRa concentrator HAL. The Semtech LoRa concentrator is a digital multi-channel multi-standard packet radio used to send and receive packets wirelessly by using LoRa or FSK modulations.
The libloragw can configure the frequency of the uplink and downlink channels, the RF signal power, and downlink bandwidth.

The libloragw is composed of 6 modules:

- loragw_hal
  This is the main module and contains the high level functions to configure and use the LoRa concentrator
- loragw_reg
  This module is used to access the LoRa concentrator registers by name other than by address
- loragw_spi
  This module contains the functions to access the LoRa concentrator register array through the SPI interface
- loragw_aux
  This module contains a single host-dependant function wait_ms to pause for a defined amount of milliseconds
- loragw_gps
  This module contains functions to synchronize the concentrator's internal counter with an absolute time reference, which in this case, a GPS satellite receiver
- loragw_radio
  This module contains functions to handle the configuration of SX125x radios

2.4.4 RTC

The RTC is used for system time clock once the GPS or NTP doesn't work.

2.5 Protocol

Several protocols support the GIoT Gateway. For the uplink and downlink, there are the HTTPs and MQTT protocols. For the GNMS notifications, there is the MQTT protocol. For GNMS configuration and monitoring, there is the TR069 protocol. For LoRa MAC commands, OTAA and ABP flow, there is the LoRaWAN 1.0 protocol. The following section introduces the above-mentioned protocols.

2.5.1 HTTPS

GIOT Gateway secures the Ethernet packets between the gateway and the Cloud server by using HTTPS. Each time the gateway POST or GET data, there must be a legal token in place to encrypt the data so that the data can be secured even when we use readable JSON format data.

HTTPS (also called HTTP over TLS, HTTP over SSL and HTTP Secure) is a protocol for secure communication over a computer network which is widely used on the Internet. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is authentication of the visited website and protection of the privacy and integrity of the exchanged data.

In its popular deployment on the internet, HTTPS provides authentication of the website and associated web server with which one is communicating, which protects against man-in-the-middle attacks. Additionally, it provides bidirectional encryption of communications between a client and server, which protects against eavesdropping and tampering with or forging the contents of the communication. In practice, this provides a reasonable guarantee that one is communicating with precisely the website that one intended to communicate with (as opposed to an impostor), as well as ensuring that the contents of communications between the user and site cannot be read or forged by any third party.

The HTTPS uniform resource identifier (URI) scheme has identical syntax to the standard HTTP scheme, aside from its scheme token. However, HTTPS signals the browser to use an added encryption layer of SSL/TLS to protect the traffic. SSL/TLS is especially suited for HTTP since it can provide some protection even if only one side of the communication is authenticated. This is the case with HTTP transactions over the Internet, where typically only the server is authenticated (by the client examining the server's certificate).

HTTPS creates a secure channel over an insecure network. This ensures reasonable protection from eavesdroppers and man-in-the-middle attacks, provided that adequate cipher suites are used and that the server certificate is verified and trusted.

GIoT Gateway install below packages to support HTTPS

- OpenSSL
- CURL

2.5.2 MQTT

There are two ways to POST uplink and GET downlink data from the Cloud. One is via HTTPS and the other is MQTT. MQTT is more convenient to integrate for common users because an HTTPS Server is not necessarily required. In addition, an MQTT broker Server is easier to setup and operate.

MQTT (acronym for MQ Telemetry Transport) is an ISO standard (ISO/IEC PRF 20922) publish-subscribe- based “lightweight” protocol; a messaging protocol used on top of the TCP/IP protocol and port 1883 is defined at IANA as MQTT over TCP. It is designed for connections with remote locations where a “small code footprint” is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The broker is responsible for distributing messages to interested clients based on the topic of a message. Andy Stanford-Clark and Arlen Nipper of Cirrus Link Solutions authored the first version of the protocol in 1999. MQTT relies on TCP as transport protocol, which means by default the connection does not use an encrypted communication, to encrypt the whole MQTT communication, most MQTT brokers allow to use TLS instead of plain TCP, port 8883 is standardized for a secured MQTT connection.
MQTT defines methods (sometimes referred to as verbs) to indicate the desired action to be performed on the identified resource. What this resource represents, whether pre-existing data or data that is generated dynamically, depends on the implementation of the server. Often, the resource corresponds to a file or the output of an executable residing on the server.

- Connect
  Waits for a connection to be established with the server
- Disconnect

Waits for the MQTT client to finish any required work, and for the TCP/IP session to disconnect

- Subscribe
  Waits for the completion of the Subscribe or UnSubscribe method
- UnSubscribe
  Requests the server to unsubscribe the client from one or more topics
- Publish
  Returns immediately to the application thread after passing the request to the MQTT client

2.5.3 TR069

GNMS uses TR069 to monitor and config the GIoT Gateway. It is a reliable and flexible protocol to control the Internet device in general.

TR-069 (Technical Report 069) is a technical specification that defines an application layer protocol for remote management of end-user devices. It was published by the Broadband Forum and entitled CPE WAN Management Protocol (CWMP).

As a bidirectional SOAP/HTTP-based protocol, it provides the communication between customer-premises equipment (CPE) and Auto Configuration Servers (ACS). It includes both a safe auto configuration and the control of other CPE management functions within an integrated framework. The protocol addresses the growing number of different Internet access devices such as modems, routers, gateways, as well as end-user devices which connect to the Internet, such as set-top boxes, and VoIP-phones. The TR-069 standard was developed for automatic configuration and management of these devices by Auto Configuration Servers (ACS). The technical specifications are managed and published by the Broadband Forum.

CWMP is a text based protocol. Orders sent between the device (CPE) and auto configuration server (ACS) are transported over HTTP (or more frequently HTTPS). At this level (HTTP) CPE is behaving in the role of client and ACS in the role of HTTP server. This essentially means that control over the flow of the provisioning session is the sole responsibility of the device.

2.5.4 LoRaWAN 1.0

The GIoT Gateway follows the LoRaWAN 1.0 MAC commands and the OTAA and ABP flow. Additionally, three new tables are introduced to the LoRa database, namely in order, the EUI table, the OTAA(devaddr) table and the ABP table.

The GIoT Gateway provides a set of parameters ---- AppKey, AppEUI, NwkEUI, Devaddr, AppSKey, NwkSKey ,Dlsetting Rxdelay, Appnonce, aging out---- lodged its LoRaWAN 1.0 GIoT LoRa database.

2.5.4.1EUI Table:

GIoT has defined 12 parameters in the EUI table. Please see the following list:

```
EUI TABLE   (
  'eui_group' ,
  'appeui_min'
  'appeui_max'
  'deveui_min'
  'deveui_max'
  'devaddr_min'
  'devaddr_max'
  'appkey'
  'dlsetting'
  'rxdelay'
  'appnonce'
  'agingout_time'
);
```

Based on the EUI definition for appeui and deveui values, the parameter of a joining device node has to meet its distinct specifications in order to join a GIoT gateway, ie. the given appeui value has to be between appeui_min and appeui_max, the given deveui value has to be between deveui_min and deveui_max.

A eui_group within the eui_table can be determined once the values are verified. The eui_group value can be used to find the dev address of the OTAA table that shares the same eui_group value.

2.5.4.2 OTAA Table

GIoT has defined 10 parameters in the OTAA table. Please see the following list:

```
OTAA TABLE    (
  'devaddr'
  'appeui'
  'deveui'
  'appskey'
  'nwkskey'
  'eui_group'
  'dev_type' ,
  'status'
  'devnonce'
  'datetime'
    );
```

There are a number of operational functions available for the OTAA Table Operation: insert/update/delete. Please direct to the GNMS web page for the OTAA Table Operation if devaddr values are to be added to the OTAA table.

2.5.4.3 ABP Table

GIoT has defined 10 parameters in the ABP table. Please see the following list:

```
OTAA TABLE    (
  'devaddr'
  'appeui'
  'deveui'
  'appskey'
  'nwkskey'
  'eui_group'
  'dev_type' ,
  'status'
  'devnonce'
  'datetime'
    );
```

There are a number of operational functions available for the ABP Table Operation: insert/update/delete. Please direct to the GNMS web page for ABP Table Operation if devaddr values are to be added to the ABP table.

3. Connection between GIoT Gateway and LoRaWAN End-Device

The GIoT Gateway agent supports LoRaWAN OTAA, ABP and GIoT ABP. The OTAA and ABP flow is described in 2.2.5. The GIoT ABP is another flow for a group of End-devices where users can define the first 7 bits of LoRa Devaddr as the NetID. The End-devices with the same NetID may use the same NwKSkey and AppSKey which will benefit more convenience during manufacturing.

3.1 GIoT Packet Forward

The packet forwarder is a program running on the host of a Lora gateway that forwards RF packets received by the concentrator to a server through a IP/UDP link, and emits RF packets that are sent by the server. It can also emit a network-wide GPS-synchronous beacon signal used for coordinating all nodes of the network.

Uplink: radio packets received by the gateway, with metadata added by the gateway, forwarded to the server. It might also include gateway status.

Downlink: packets generated by the server, with additional metadata, to be transmitted by the gateway on the radio channel. It might also include configuration data for the gateway.

3.2 GIoT LoRa Server

GIoT LoRa Server supports MQTT and HTTPS to POST and GET data from the Cloud. The LoRa server uses the below information to decrypt uplink and encrypt downlink packages respectively.

3.2.1 End-device address (DevAddr)

The DevAddr consists of 32 bits identifies the end-device within the current network. Its format is as follows:

Bit#	[31..25]	[24..0]
DevAddr bits	NwkID	NwkAddr

The most significant 7 bits are used as network identifier (NwkID) to separate addresses of territorially overlapping networks of different network operators and to remedy roaming issues. The least significant 25 bits, the network address (NwkAddr) of the end-device, can be arbitrarily assigned by the network manager.

3.2.2 Application identifier (AppEUI)

The AppEUI is a global application ID in IEEE EUI64 address space that uniquely identifies the entity able to process the JoinReq frame. The AppEUI is stored in the end-device before the activation procedure is executed.

3.2.3 Network session key (NwkSKey)

The NwkSKey is a network session key specific for the end-device. It is used by both the network server and the end-device to calculate and verify the MIC (message integrity code) of all data messages to ensure data integrity. It is further used to encrypt and decrypt the payload field of a MAC-only data messages.

3.2.4 Application session key (AppSKey)

The AppSKey is an application session key specific for the end-device. It is used by both the application server and the end-device to encrypt and decrypt the payload field of application-specific data messages. Application payloads are end-to-end encrypted between the end-device and the application server, but they are not integrity protected. Which means a network server may be able to alter the content of the data messages in transit. Network servers are considered to be trusted, but applications that aim to implement end-to-end confidentiality and integrity protection are recommended to use additional end-to-end security solutions, which are in general beyond the scope of this specification.

For OTAA flow, there are more DevEUI and APPKey in Join request.

3.2.5 End-device identifier (DevEUI)

The DevEUI is a global end-device ID in IEEE EUI64 address space that uniquely identifies the end-device.

3.2.6 Application key (AppKey)

The AppKey is an AES-128 root key specific to the end-device. Whenever an end-device joins a network via over-the-air activation, the AppKey is used to derive the session keys NwkSKey and AppSKey specific for that end-device to encrypt and verify network communication and application data.

3.2.7 GIoT Frame Buffer

To make the GIoT Gateway more reliable, it encompasses a buffering function that can save the uplink data for approximately 2 hours in case the Internet connection is down. Once the Internet is available, the Gateway will continue to POST the uplink data to the Cloud.

4 Connection between GIoT Gateway and Networking Server

There are several agents to support the LoRaWAN 1.0, GNMS monitor, and configuration.

4.1 GIoT CWMPC Agent

➢GIoT CWMPC is an agent that reports the configuration and status of the Gateway to the GNMS system. It provides the below information to the GNMS.

- Host Information:
  1.GPS location of gateway
  2.Provision code
  3.Gateway ID
  4.Gateway type (Macro/Micro/Pico/virtual)
    · 1 byte length
  5.Mode name
  6.Keep alive interval
  7.Watchdog's restart counter

- LoRa Chipset information:
  1.Chipset name
  2.Channel assignment
  3.Frequency of channel
  4.Bandwidth of downlink
  5.Coding rate
  6.Power(dBm)
  7.List of Spreading Factor enable
  8.Node RX queue length
  9.GIoT key Status (Only 5 bytes and CRC status to display locally)
- LoRa frame statistics:
  1.Cloud RX/TX/Drop counter(to/from cloud)
  2.Node RX/TX/Drop counter(to/from end-device)
  3.Node RX queue usage:%
  4.Node RX queue drop counter
  5.Frame lost of mac layer(FIFO status):%

➢CWMPC supports the below parameter settings from GNMS:

- Host Information:
  1.Provision
  2.Gateway type (Macro/Micro/Pico/virtual)
  3.Keep alive interval
- LoRa Chipset :
  1.Channel assignment
  2.Frequency of channel
  3.Bandwidth of downlink
  4.Coding rate
  5.Power(dBm)
  6.List of Spreading Factor enable
  7.Node RX queue length
  8.GIoT key

➢The GNMS can trigger a series of diagnostic events to the GIoT Gateway through CWMPC.

1.LoRa module's physical connectivity as in hardware check: SPI/USB

2.DNS check with routing server domain

3.Ping to routing server 3 times

4.Frame Debug provides a frame log with at least 128 RX and TX data during debug mode. The log contains the following fields:

- Index
- TimeStamp
- Seq
- Channel
- SNR(dB)
- RSSI(dBm)
- Freq.
- Data rate - (ex. SF7BW125)
- C/R
- Payload
- Mac address

4.2 GIoT Alert Agent

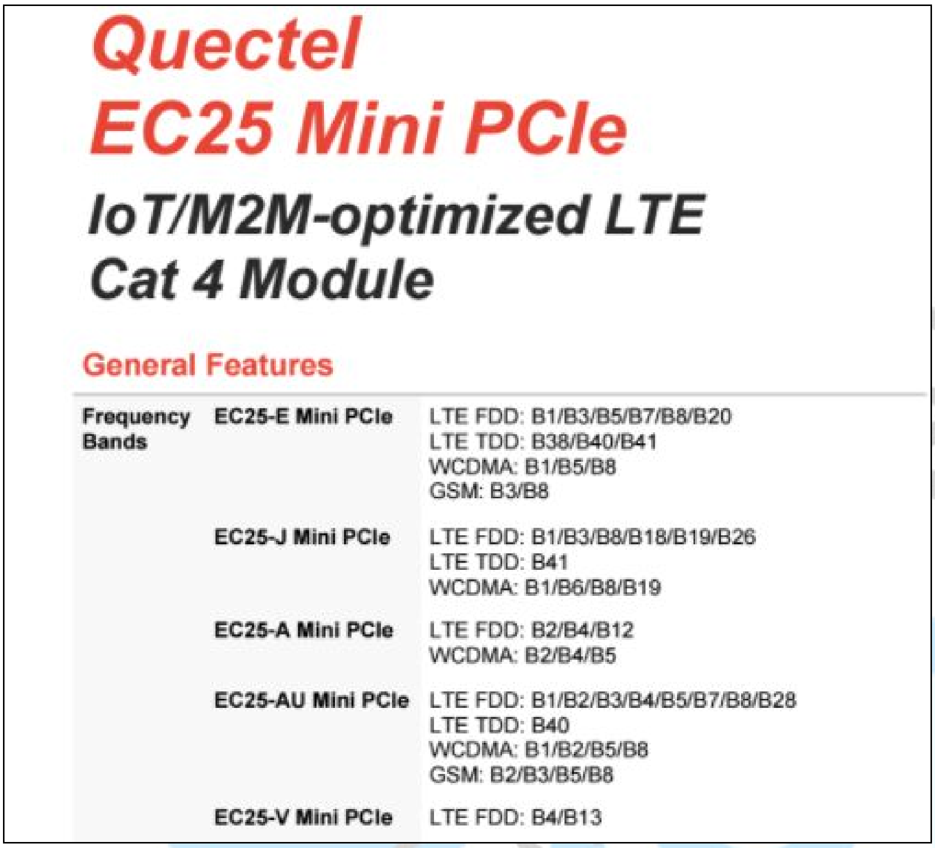
The Alert Agent monitors the system usage and loading. The Alert Agent will be triggered once the usage volume surpasses the specified threshold, which will subsequently send an alert message to the GNMS via MQTT. The alarm thresholds are described in the below table.

Features	**Description**
CPU	The CPU usage rate is considered a potentially unsustainable load. This rate is the average of all microprocessor CPUs. **Definition (0:Disabled, 1: Enabled):** -Condition: CPU load >= 80% -Detects every 10 secs -Raises flag when the condition threshold is reached and lasts over 180 secs -Turnaround conditions for lowering -Drops flag when CPU usage is contained beneath

	the threshold and lasts for over 90 secs
Memory	The Memory usage rate is considered a potentially unsustainable load. **Definition (0:Disabled, 1: Enabled):** -Condition: memory usage >= 80% -Detects every 10 secs - Raises flag when the condition threshold is reached and lasts over 180 secs -Turnaround conditions for lowering -Drops flag when CPU usage is contained beneath the threshold and lasts for over 90 secs
LoRa boot switch	Provided that the base station has a backup image whicht can induce LoRa boot switch. **Definition (0:Disabled, 1:Enabled):** -Condition: LoRa boot exchange -Detects every 20 secs -Raises flag when the condition threshold is reached -Drops flag after 180 secs
Watchdog	The watchdog will periodically check the status of drivers and processes. DMS will be advised if the watchdog fails and stops. **Definition (0:Disabled, 1:Enabled):** -Condition: software watchdog fails -Detects every 10 secs - Raises flag when the condition threshold is reached - Drops flag after 180 secs

The software watchdog monitors the CWMPC, packet forwarding, LoRa server, and alert daemons. When either of them fails, the watchdog will re-launch the items in 5 seconds.

Appendix

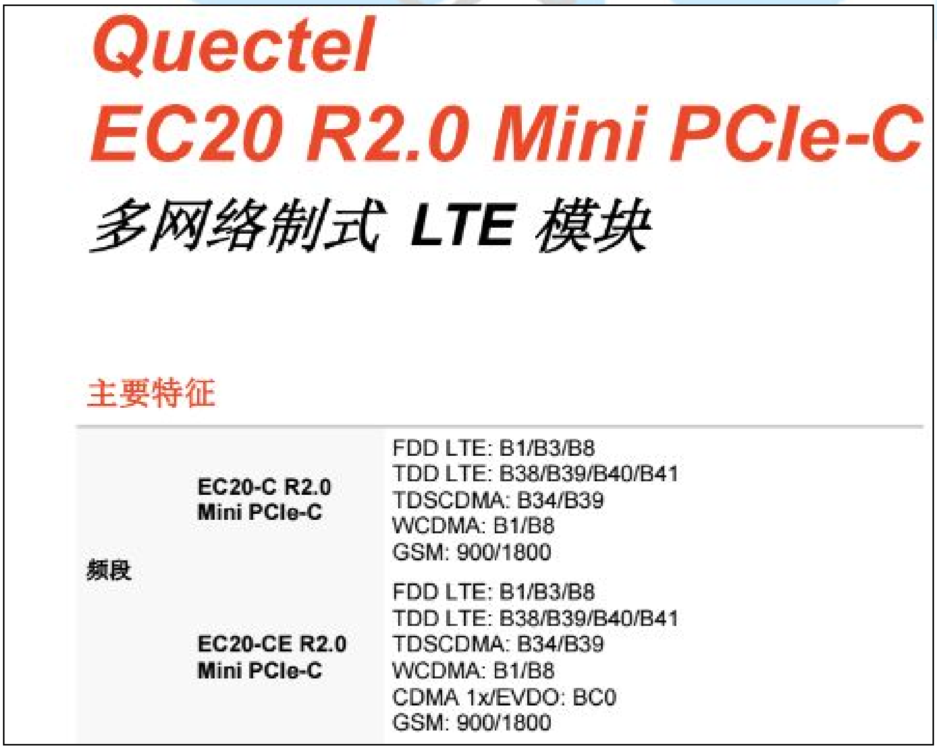


Quectel
EC25 Mini PCIe

IoT/M2M-optimized LTE Cat 4 Module

General Features

Frequency Bands	EC25-E Mini PCIe	LTE FDD: B1/B3/B5/B7/B8/B20 LTE TDD: B38/B40/B41 WCDMA: B1/B5/B8 GSM: B3/B8
	EC25-J Mini PCIe	LTE FDD: B1/B3/B8/B18/B19/B26 LTE TDD: B41 WCDMA: B1/B6/B8/B19
	EC25-A Mini PCIe	LTE FDD: B2/B4/B12 WCDMA: B2/B4/B5
	EC25-AU Mini PCIe	LTE FDD: B1/B2/B3/B4/B5/B7/B8/B28 LTE TDD: B40 WCDMA: B1/B2/B5/B8 GSM: B2/B3/B5/B8
	EC25-V Mini PCIe	LTE FDD: B4/B13



Quectel
EC20 R2.0 Mini PCIe-C

多网络制式 LTE 模块

主要特征

频段	EC20-C R2.0 Mini PCIe-C	FDD LTE: B1/B3/B8 TDD LTE: B38/B39/B40/B41 TDSCDMA: B34/B39 WCDMA: B1/B8 GSM: 900/1800
	EC20-CE R2.0 Mini PCIe-C	FDD LTE: B1/B3/B8 TDD LTE: B38/B39/B40/B41 TDSCDMA: B34/B39 WCDMA: B1/B8 CDMA 1x/EVDO: BC0 GSM: 900/1800

Module	[Frequency Bands]	LTE FDD	LTE TDD	TDSCDMA	WCDMA	GSM
EC20-CE	LTE BAND 1/3/8/38/39/40/41	B1/B3/B8	B38/B39/B40/B41	B34/B39	B1/B8	900/1800
EC25-E	LTE BAND 1/3/5/7/8/20/38/40/41	B1/B3/B5/B7/B8/B20	B38/B40/B41		B1/B5/B8	B3/B8
EC25-J	LTE BAND 1/3/8/18/19/26/41	B1/B3/B8/B18/B19/B26	B41		B1/B6/B8/B19	
EC25-A	LTE BAND 2/4/12	B2/B4/B12			B2/B4/B5	
EC25-AU	LTE BAND 1/2/3/4/5/7/8/28/40	B1/B2/B3/B4/B5/B7/B8/B28	B40		B1/B2/B5/B8	B2/B3/B5/B8

Summary Chart for 3G/4G Module Frequency Bands