NB Devlopment Guide

Device Development > Access Mode MCU > NB-IoT General Solution

Version: 20200319

Online Version

Contents

1 Background

The Narrow Band Internet of Things (NB-IoT) is built on cellular mobile communication network to achieve thing-to-thing communication and thing-to-person communication. NB-IoT focuses on the Low Power Wide Area (LPWA) Internet of Things (IoT) market.

The NB-IoT network consumes only about 180 kHz of bandwidth. Using the License frequency band, three deployment methods such as in-band, guard band, or independent carrier can be used to coexist with the existing network. Can be directly deployed on GSM, UMTS or LTE networks to reduce deployment costs and achieve smooth upgrades.

1.0.1 NB-IoT Features

- **Low-power**: IoT Products(such as smart meter reading, environmental monitoring, smart agriculture, etc.) have no power supply in the installation environment and require batteries. In order to meet the battery's 5 to 10-year life expectancy, the NB-IoT network introduced PSM and eDRX technology. It greatly reduces the power consumption of the terminal, which can keep the device in a very low power consumption state for most of the life cycle, thereby ensuring the battery life.
- **Low-cost**: NB-IoT Products use narrow-band technology, low baseband complexity, only use a single antenna, adopt half-duplex mode, low cost of RF module, most unnecessary functions (SRVCC, IMS, emergency call, etc.) are cut and the use of the SoC's built-in power amplifier PA reduce the requirements for terminal flash storage space, terminal size, terminal radio frequency, etc., thereby greatly reducing the cost of NB-IoT.
- **Large number of connections**: NB-IoT has 50-100 times higher uplink capacity than 2G / 3G / 4G (specific service model). NB-IoT can provide 50-100 times the number of accesses than existing wireless technologies. A single cell can support 5 10,000-level user scale.
- **Wide coverage**: 164db MCL, NB-IoT improves 20db gain over GPRS, it can cover well in places where signals such as underground garages, basements, and underground pipes are difficult to reach.

Features	Details	Specific value range
Less data	Limited Air Interface resources (180khz), suitable for small data communication	50 ~ 200 bytes is suitable, the smaller the better
Low frequency, long period	Most terminals should be in a dormant state for a long time, the frequency of reporting data is low	Report by day, 1-2 times per day is more appropriate. High frequency reporting (for example, 30 minutes), will occupy a large amount of network capacity. The higher the reporting frequency, the greater the impact on network capacity.
Low power consumption	Lowest power consumption in PSM mode	Power sensitive applications preferably
Slow mobility	Suitable for slow mobility	Movement speed is less than 30km/h
Wide coverage	Batter coverage	Can support scene coverage like basement
Low rate	The theoretical peak rate of upstream is 15.6kbps, theoretical peak rate of downstream is 21.25kbps	Bandwidth high-rate services cannot be carried by NB-IoT network

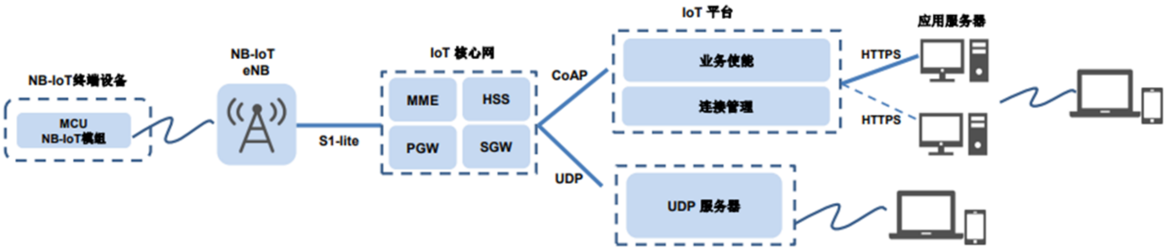


Figure 1: image.png

1.0.2 NB-IoT Network Structure

The NB-IoT network consists of NB-IoT terminals, NB-IoT base stations, NB-IoT packet core networks, IoT connection management platforms, and industry application servers. The access network architecture of NB-IoT is the same as that of LTE.

1.0.3 Power-saving Mode

> The development of NB-IoT business scale is closely related to the bearer business model, and its applicable scenarios are "small traffic, reporting-oriented, long-term sleep, power consumption sensitive, low mobility" applications. In order to realize the large number of low-power devices carried by the NB-IoT network, the most important technologies are: PSM and eDRX

NB-IoT supports 3 power-saving modes: **PSM** (Power Saving Mode), **DRX** (Discontinuous Reception), **eDRX** (Extended DRX)

1.0.4 PSM Mode

The terminal is deeply dormant during non-business periods and does not receive downlink data. Only the terminal can actively receive the downlink data buffered by the IoT platform when it actively sends uplink data (MO Data).

In this mode, the radio frequency of the terminal is turned off, which is equivalent to the shutdown state, but the core network side still retains the user context. When the user enters the idle state / connected state, there is no need to attach the PDN to establish it.

It is suitable for services with no delay requirements for downlink data; terminal equipment has low power consumption and uses battery-powered methods, such as meter reading services.

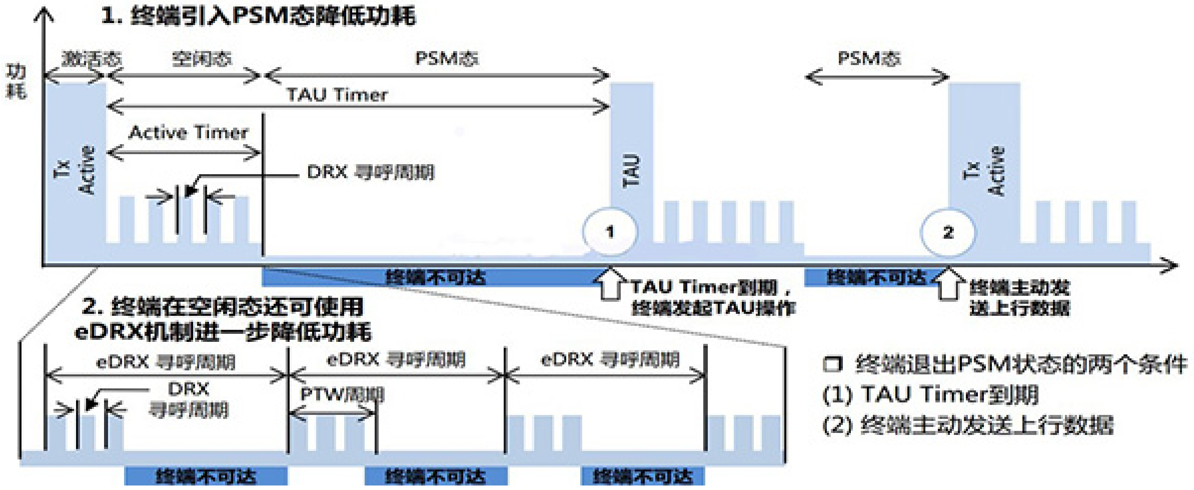


Figure 2: image.png

State:

- Active: The module is in an active state; all functions are normally available and data can be sent and received; the module can switch to Idle mode or PSM mode in this mode.
- Idle: The module is in a light sleep state; the module can receive paging messages in network connected state. Module can switch to Active or PSM mode in this mode.
- PSM: The module is in a deep sleep state, only the RTC works internally, the network is in a disconnected state and it cannot receive paging messages. When the timer expires, the module will be woken up; the module can also be woken up by pulling down the PSM_EINT pin.

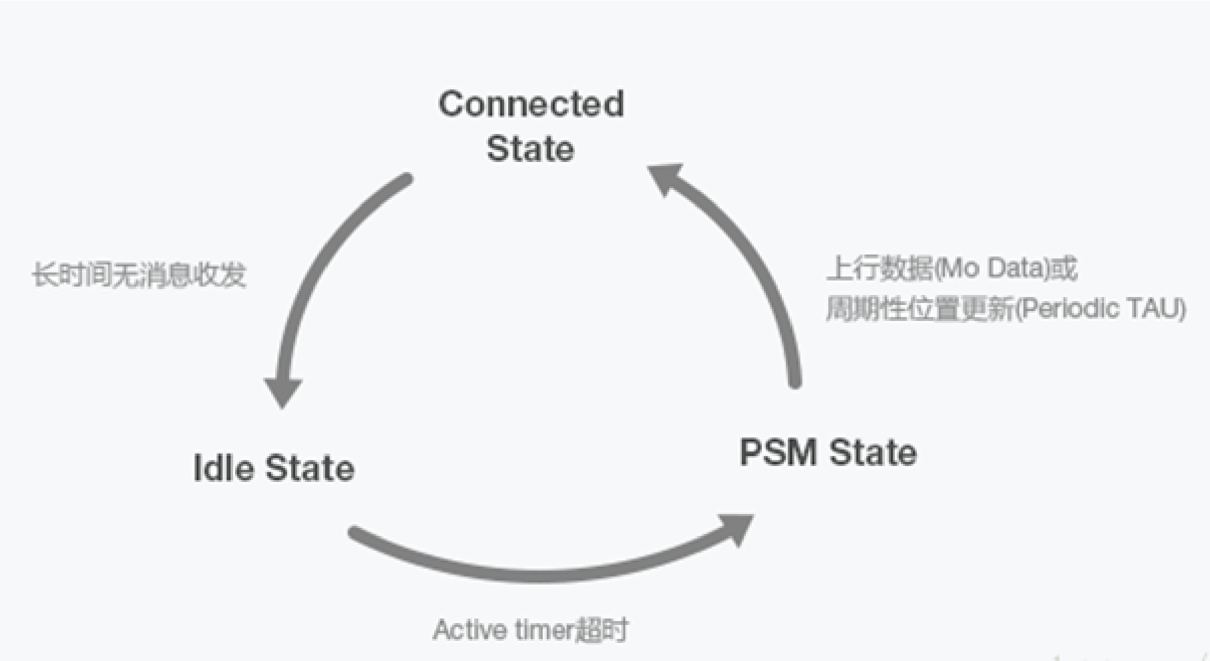


Figure 3: image.png

1.0.5 eDRX Mode

The terminal equipment takes into account both low-power consumption and services with certain delay requirements. In each eDRX cycle, the terminal can receive downlink data only within the set paging time window, the rest of the time the terminal is in a sleep state and does not receive downlink data. This mode can achieve a balance between downlink service delay and power consumption, such as remotely shutting down the gas service.

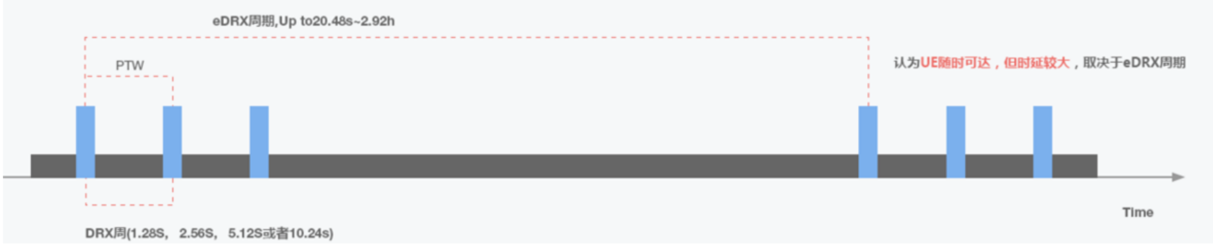

eDRX周期,Up to20.48s~2.92h
PTW
认为UE随时可达，但时延较大，取决于eDRX周期
Time
DRX周(1.28S, 2.56S, 5.12S或者10.24s)

Figure 4: image.png

Within each eDRX cycle, there is a paging time window (PTW). The terminal monitors the paging channel in the PTW according to the DRX cycle (the DRX cycle time is short, the terminal can be considered not to sleep and always reachable) in order to receive downlink Data, the rest of the time the terminal is dormant. The eDRX mode can be considered that the terminal equipment is reachable at any time, but the delay is large. The delay depends on the eDRX cycle configuration, which can achieve a balance between low power consumption and delay.

1.0.6 DRX Mode

It can be considered that the downlink service can reach the terminal equipment at any time. In each DRX cycle, the terminal detects whether a downlink service arrives once, which is suitable for services with high requirements on delay. Terminal equipment generally adopts power supply methods, such as street light services.

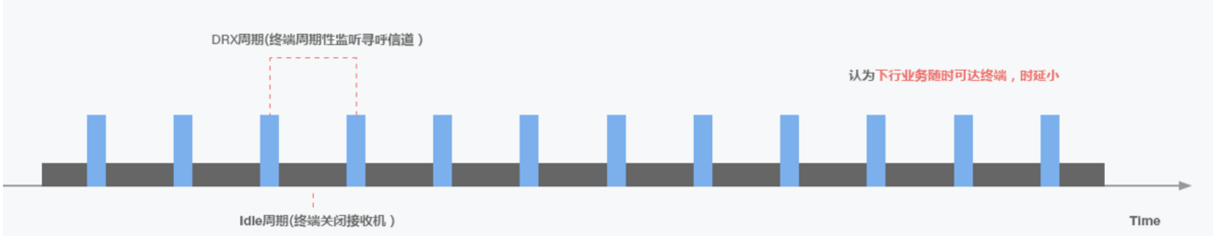

DRX周期(终端周期性监听寻呼信道）
认为下行业务随时可达终端，时延小
Idle周期(终端关闭接收机）
Time

Figure 5: image.png

Due to the short DRX cycle (1.28s, 2.56s, 5.12s, or 10.24s, which is determined by the operator's network settings and depends on the APN service handled by SIM), it can be considered that downlink services are always available and the delay is small. It is suitable for services that have high requirements on delay, but the power consumption is relatively high. Terminal equipment generally uses the mains power supply method.

1.0.7 NB-IoT Latency

First time network access latency: After the NB-IoT terminal is turned on, the terminal has a lot of message interaction with the network (authentication, channel establishment, IP address allocation, etc.). It takes 6-8s to complete network access and obtain an IP address Used for later data transmission.

Data reporting and reception latency: After the NB terminal is successfully connected, when the terminal has data transmission, the terminal will actively establish a wireless connection with the base station (no authentication, IP address allocation and other processes are required at this time). Then send the data immediately. The delay of data reporting by the terminal is closely related to the state of the terminal and the wireless network coverage.

Data reported by the terminal	Data issued by the platform(PSM)	Data issued by the platform(DRX)	Data issued by the platform(eDRX)
Air interface delay + private network to client server delay	Air interface delay + private network to platform delay + PSM maximum sleep period (maximum 310 hours)	Air interface delay (750ms) + DRX paging cycle (maximum 10.24 seconds, minimum 1.28 seconds)	Air interface delay (750ms) + eDRX paging cycle (maximum 2.92 hours, minimum 5.12 seconds)
Seconds (3 to 30 seconds)	Hour / day, depending on the reporting period of the terminal	Seconds, depending on the DRX paging cycle	Seconds and hours, depending on eDRX paging cycle

1.0.8 NB-IoT Key Parameters

The low-power consumption mechanism of the NB application is inseparable from the configuration of the SIM card operator. Therefore, when purchasing SIM cards, please be clear about the APN business you handle. Different APNs are suitable for different application scenarios.

> If you have any problem about how to choose APN business, please ask Tuya stuff.

1.0.9 NB-IoT Power Consumption

State	Power consumption	Measured results
PSM	3 uA	2.7 uA
eDRX	xxuA~2 mA	1 mA

State	Power consumption	Measured results
DRX	1~4mA	1mA
Connectted	Sending 200 mA, receiving 65 mA	Sending 189mA, receiving 161 mA

2 Tuya MCU Solution Introduction

The Narrow Band Internet of Things (NB-IoT) is built on cellular mobile communication network to achieve thing-to-thing communication and thing-to-person communication. NB-IoT focuses on the Low Power Wide Area (LPWA) Internet of Things (IoT) market.

The NB-IoT network consumes only about 180 kHz of bandwidth. Using the License frequency band, three deployment methods such as in-band, guard band, or independent carrier can be used to coexist with the existing network. Can be directly deployed on GSM, UMTS or LTE networks to reduce deployment costs and achieve smooth upgrades.

Tuya is providing developers with MCU SDK that can be quickly programmed .This is a development kit generated automatically from the product's function datapoint according to Tuya Serial Port Communication Protocol. MCU engineer can develop this procedure conveniently on this basis. This article will introduce you to the specific steps of MCU development.

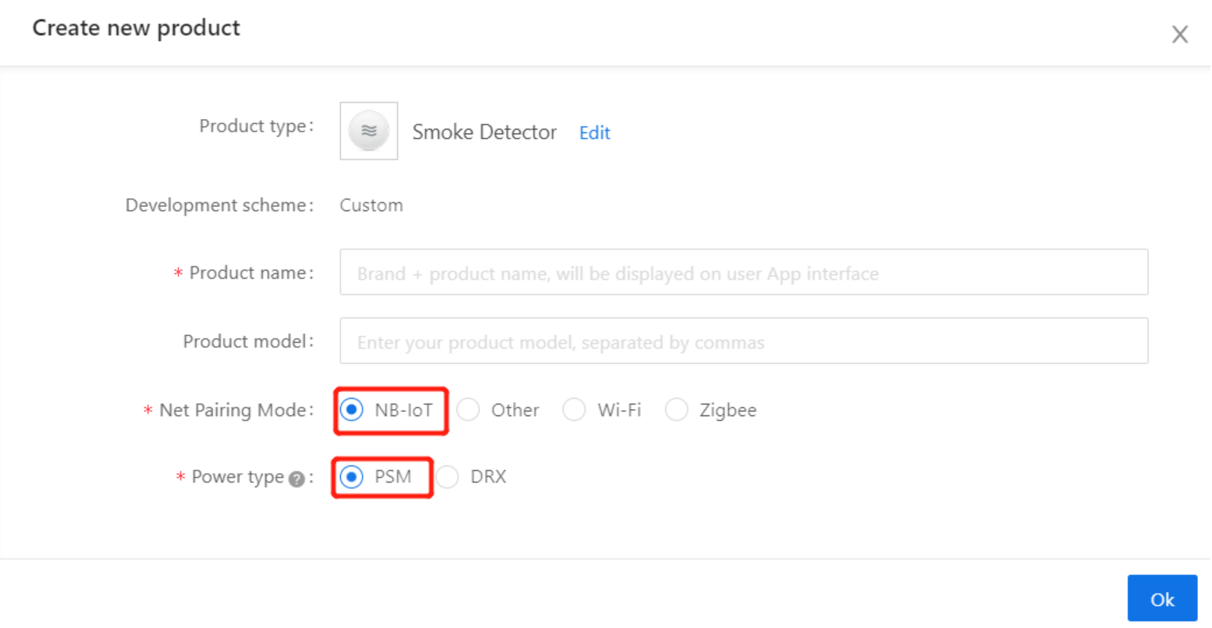
3 Development Process

The development process mainly includes: Creating products - Hardware debugging-software debugging - function debugging.

3.1 Creating products

Login Tuya Smart Developer Platform to create products. Please note that the communication type is **"NB-IoT"**, The type of power consumption is selected based on the characteristics of the product. For common battery-powered, monitoring and reporting services, please select **"PSM"**. For products that are not sensitive to power consumption and deliver controlled services, please select **"DRX"** (note that the choice of power consumption will affect the sending and production of subsequent modules).

After the product is created, users can select functions, panels, modules and firmware according to the actual needs of the product, and download the corresponding MCU development kit.


Create new product

Product type: Smoke Detector

Development scheme: Custom

* Product name: Brand + product name, will be displayed on user App interface

Product model: Enter your product model, separated by commas

* Net Pairing Mode: ● NB-IoT ○ Other ○ Wi-Fi ○ Zigbee

* Power type: ● PSM ○ DRX

Figure 6: 微信截图 _20200316175439.png

Creating a product is described in detail in the MCU Access Guide, so we won't go into details here.

3.2 Hardware Debugging

When selecting module on the platform, the platform will recommend some common module models. After selecting the module and firmware, you can purchase module samples online.

The hardware engineer can enter the drawing board stage, and the related materials for hardware development can be viewed in the developer center:

Data manual link: NB-IoT module datasheet(not available now);

Hardware design guidance: NM1 module hardware design(not available now);

PCB data: Common module package library.

Note: The peak working current of the NB-IoT universal module will reach more than 300mA. Be sure to leave a margin when designing the power supply.

3.2.1 Build Minimal Hardware System

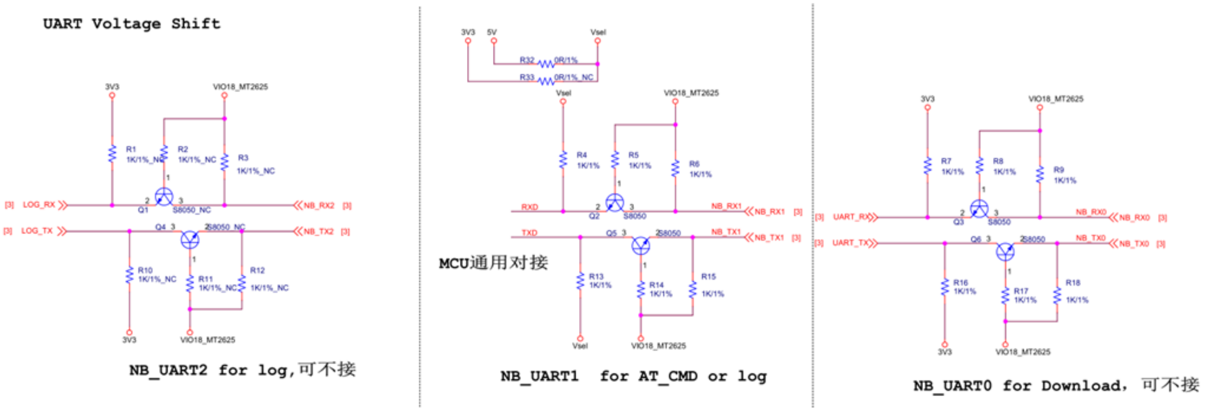


Figure 7: image.png

- Serial communication circuit-serial level 1.8V, level conversion required
  - UART0: module firmware programming pin, which is generally not used in the solution connection process, which can connect to test point debug (baud rate 115200)
  - UART1 (commonly used): Universal docking pin, used for serial communication between module and user chip (baud rate 115200 or 9600)

- UART2: Log print pin, used to find problems in the docking process. It needs to be used for collaborative judgment by grabbing logs, which can connect to test point debugging. (Baud rate 921600)

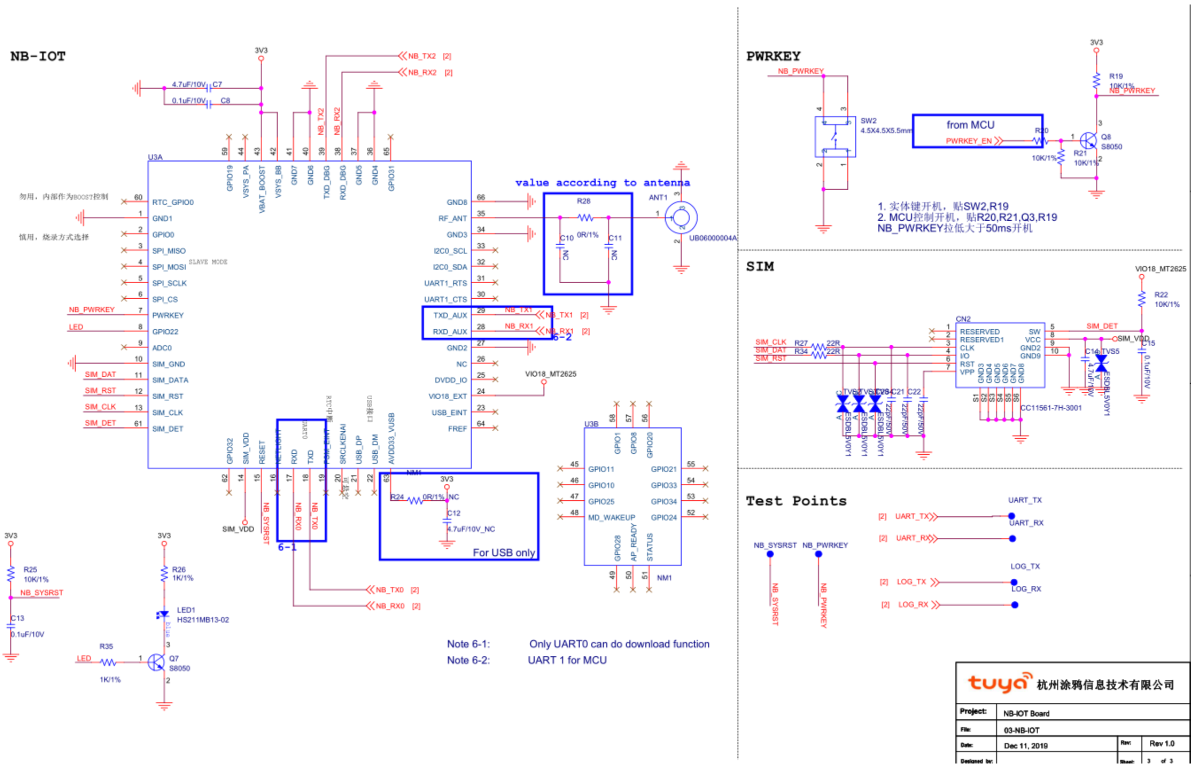


Figure 8: image.png

- Module wake-up circuit:
  - Power-on wake-up pin: PWRKEY
  - Low-power consumption mode wake-up pin: PSM_EINT
- Sim card circuit: Determine SIM Encapsulation useing USIM or eSIM before designing (Will purchase SIM card according to product features)
- Power supply circuit: The peak working current of the NB-IoT universal module will reach more than 300mA. Be sure to leave a margin when designing the power supply
- Antenna circuit: Antenna RF circuit can refer to the hardware design manual (The antenna needs to communicate to the antenna manufacturer for matching.)

3.2.2 Network verification

After getting the module, you don't need to write code first, first determine whether the module is working properly. Using the module debugging assistant (MCU simulation mode) provided by Tuya to cooperate with the module can perform network config operation, verify the module and be familiar with the protocol interaction process at the same time, and the development and debugging efficiency will be greatly improved later.

Tuya MCU simulation debugging assistant can simulate MCU data transmission and reception. Set up the minimum system of the module and connect the serial port to the computer. The assistant has the following functions: 1. Pair with a networked module and verify that the module is working properly. 2. Before the MCU completes development, debug the App panel display. 3. When the MCU developer does not know how to send and reply data to the module, the data of the simulation assistant can be used as a reference.

For more information, please refer to the Module Debugging Assistant Instruction

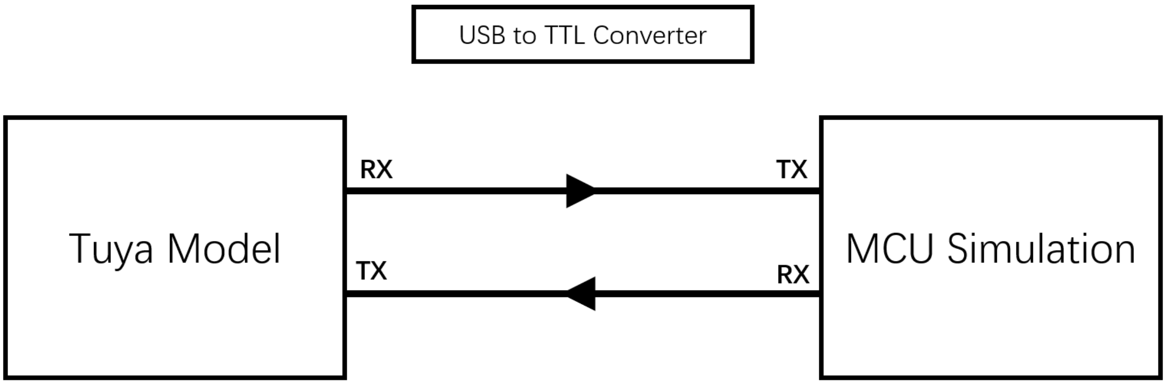


Figure 9: 微信截图 _20200316152400.png

> Module Debugging Assistant (MCU simulation mode)

step1. According to the minimum system schematic diagram, build the module peripheral circuit, simple test can fly wire directly.

step2. Open the Tuya module debugging assistant in the development kit and import the debugging file. Protocol selection NB-IoT general protocol, MCU simulation mode.

step3. Connect the module serial port to the computer through the USB to TTL tool. The assistant selects the corresponding serial port and baud rate. Open the serial port and click Start. You will see that the module and the host computer automatically perform the initialization process protocol interaction.

Note: After the NB-IoT module is powered on, it will continuously send heartbeat packets. After receiving the correct response, it will perform subsequent initialization protocol interactions. If no data is sent after power on, please check if the peripheral circuit of the module is correct.

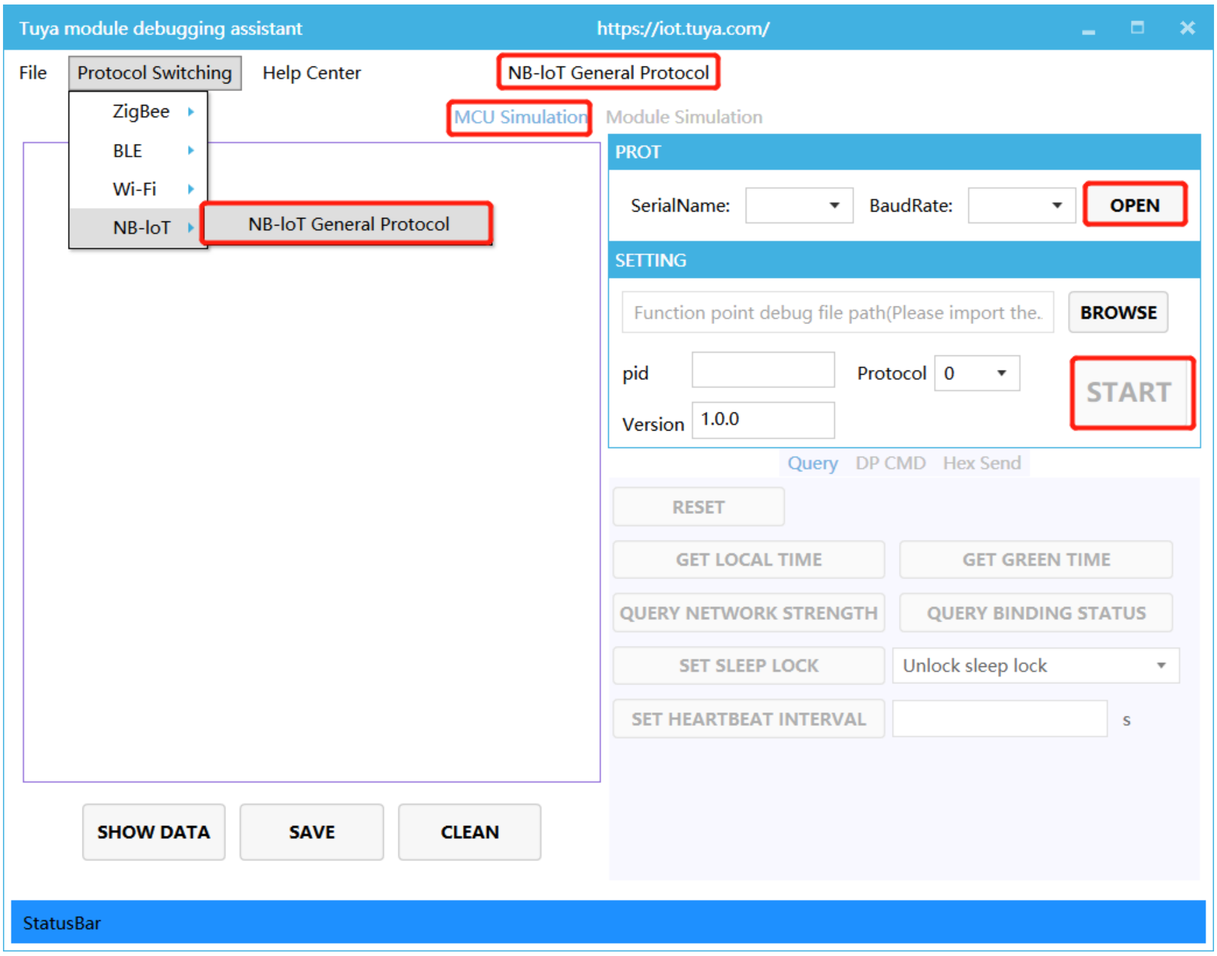


Figure 10: 微信截图 _20200316180030.png

3.2.3 App Scan Code Binding

- Registration: After the factory authorization of NB-IoT module, Tuya Cloud will pull the module information and register the module on the telecommunication IoT platform.
- Activation: After the NB-IoT module is powered on for the first time and initialized, it will connect to the telecommunications IoT platform for activation, subject to the network status issued by the module.
- Use: Only modules that have been activated and not bound by other accounts can be successfully bound by the app.

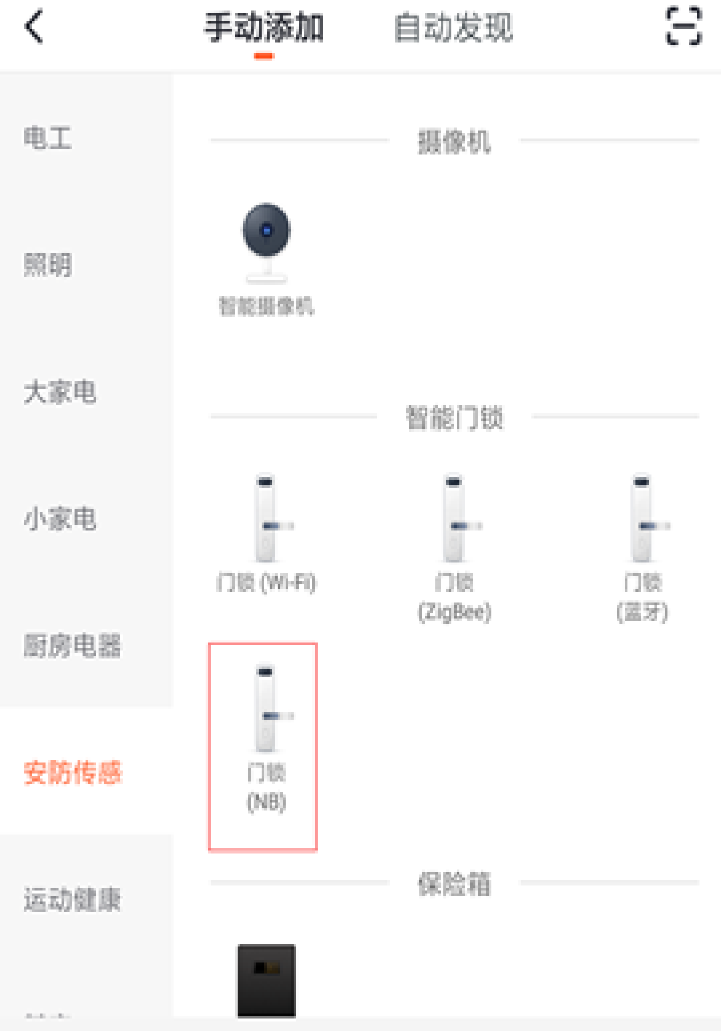


Figure 11: img

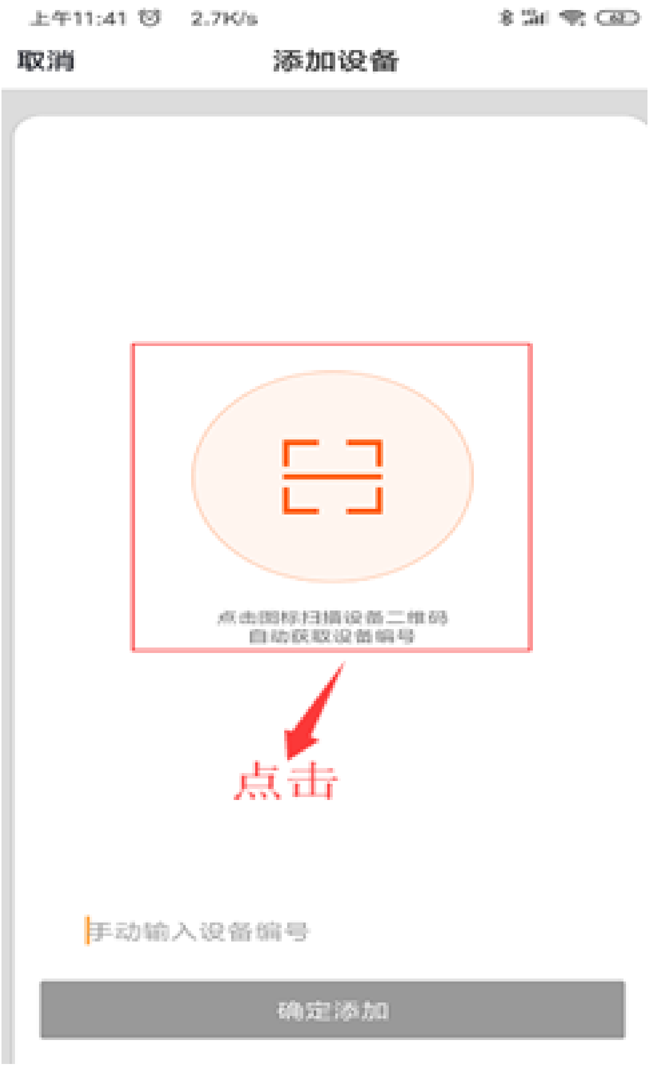


Figure 12: img

> Note: NB-IoT QR Code is generated and printed by Tuya productions Tools. You can ask staff for details.

3.3 Software Debugging

During hardware debugging, you can see that the module and the MCU have a series of serial protocol interaction data. For the data analysis part, users can refer to the protocol document in the development kit. The agreement is mainly divided into two parts: the basic protocol and the functional protocol.

The basic protocol has nothing to do with the product. It is a common protocol of the module, including module initialization instructions and some extended function instructions.

The functional protocol part is mainly based on the reporting and issuing command words of the basic protocol, and details the format of the Datapoint content. The full content of the basic protocol, the document center is kept updated in real time, you can click the link to view: Tuya Cloud NB-IoT Universal Serial Port Access Protocol.

There are two ways for the MCU to connect with the Tuya module protocol: transplant the MCU SDK or connect the protocol by yourself.

- Connect protocol by yourself:

  When the MCU resources are limited or the MCU SDK is not suitable for porting, customers can choose to connect the serial port protocol by themselves.

- Transplant MCU SDK:

  If the MCU resources are sufficient, it is generally recommended that users directly port the MCU SDK for efficient and convenient development. The MCU SDK in the development kit is a C-based protocol application code provided by Tuya, which can be directly added to the MCU project.

  The MCU SDK requires MCU hardware resources: Flash 4K bytes; RAM is related to the length of the DataPoint, about 100 bytes (if OTA function is required, it must be bigger than 260 bytes); the number of function nesting levels is 9 levels. If users with insufficient resources can connect the protocol by themselves, the functions in the SDK package can still be used as a reference.

  For details can see: Overview of migrating Tuya's MCU SDK.

3.4 Functions Debugging

After porting the MCU SDK code development, you can use the Tuya module debugging assistant - module simulation mode to verify the correctness of the MCU code. The usage method is similar to the MCU simulation mode. In the simulation module mode, the assistant will automatically send the initialization data stream to verify that the MCU response is correct and give corresponding prompts for the incorrect data. After the initial interaction is passed, you can manually click to test other extended functions.

Note： module simulation mode do not supply network functions, only used to verify the correctness of the MCU serial protocol transmission and reception. After the test is completed, the MCU can be connected to the actual module and test the network.

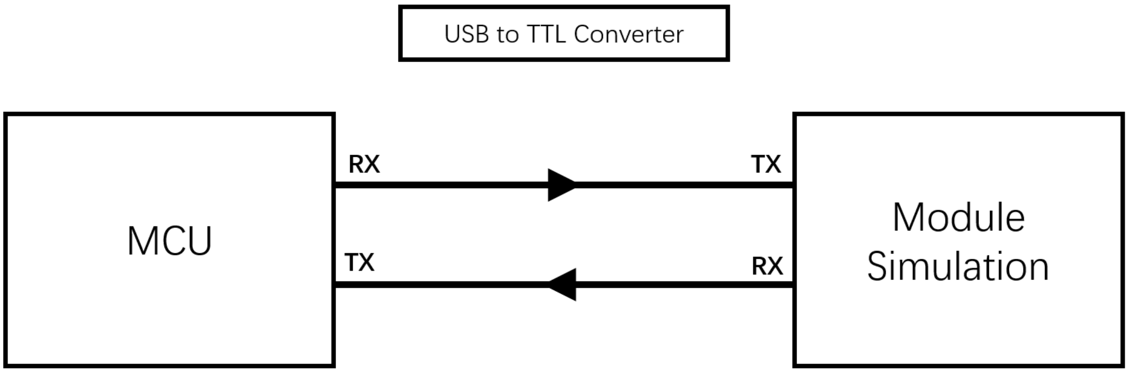
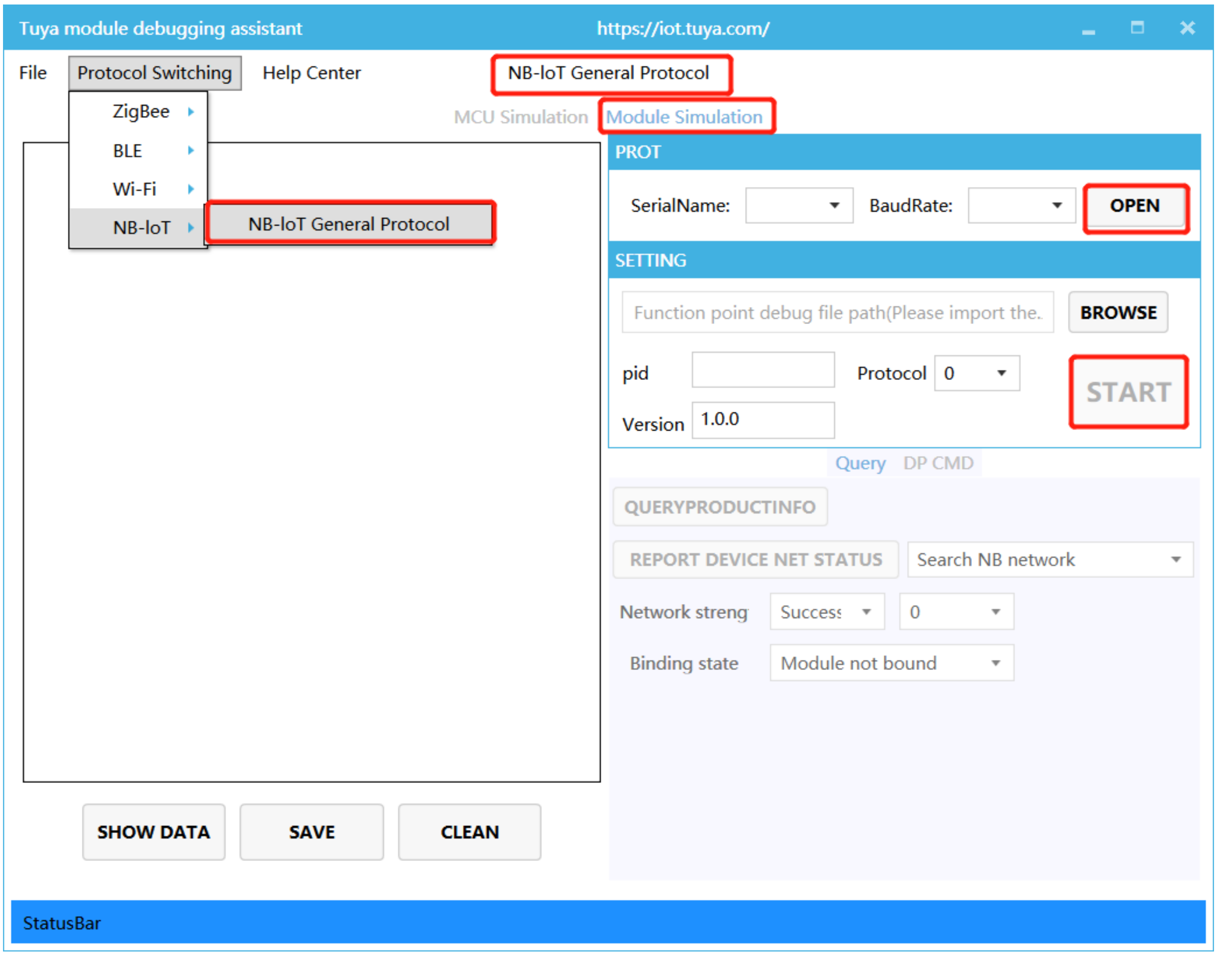


Figure 13: 微信截图 _20200316152446.png

> Module Debugging Assistant (Module simulation mode)



SerialName:
BaudRate:

Function point debug file path(Please import the.

pid

Protocol 0

Version 1.0.0

Search NB network

Network streng Success 0

Binding state Module not bound

Figure 14: 微信截图 _20200316180102.png

Other Debugging Common Tools Links：

Operations Center Guide：Tuya Smart Developer Platform - Operation. You can check device background log here according to your product id.

Tutorial for Tuya Support Center：Tuya provide online technical support, you can ask any questions like the question about documents here.

FAQ：FAQ, you can read it before developing to avoid some problem.