



BTDG2340 Datasheet –V1.3

Beijing Jia An Electronics Technology Co., Ltd

Contents

Contents..... 1

History..... 1

Features 2

Serial driver installation 2

Testing tool software 2

AT instruction operation 2

AT instruction description 3

Quick Start Guide 20

Apps 24

Mechanical Drawing 25

Overview..... 26

Contact Details 27

History

Date	Version	Description	Draft	Approval
2024-2-1	V1.0	Release	xbt	
2024-3-12	V1.1	Modify some information.	Xbt	
2024-10-15	V1.2	Update product model	Xbt	
2025-07-03	V1.3	Add command mode.	Xbt	

Features

- Long range wireless applications
- Programmable up to +8 dBm for RF output
- High sensitivity: down to -102dBm.
- Supports AT command
- Compact, easy to use
- Low cost and high performance
- Long communication distance
- Excellent radio robustness performance
- SMA antenna (provided according to demand)



Serial driver installation

The serial port of BLE Dongle uses the CH34x chip. Before using the module, it is necessary to install the corresponding driver on a Windows computer: CH341SER.EXE execute the program to install it.

Testing tool software

To verify and test the BLE Dongle function points, it is necessary to connect the module through a computer serial port for data transmission and reception. Our company provides a set of upper computer programs aimed at enabling users to quickly get started using BLE Dongle. please contact staff if needed.

AT instruction operation

BLE Dongle comes with a built-in serial port chip and USB interface. USB interface method, after installing the CH34x driver on the computer, Confirm the COM port, select the correct serial port (i.e. COM port), baud rate 9600-115200, Data bit 8 bits, stop bit 1 bit, no checksum.

AT instruction description

Default Baurate is 115200bps

The AT instruction is based on ASCII code, and the command format is as follows:

Request message format is: AT+<CMD>[OP]<para-1,para-2,...,para-n><CR><LF>

token	Description
AT+	Command Prefix
CMD	Instruction Name
OP code	Instruction operators can use OP codes as follows: [OP1] "=": indicates setting [OP2] "?": Indicates the current instruction description for querying [OP3] "": Indicates the execution of instructions [OP4] "=?": Refers to querying the current instruction setting content
Para-1,para-2,...,para-n	Indicates the parameters to be set currently
<CR><LF>	

The format of the response message is: <CR><LF> [+CMD:] <para 1, para 2,..., para n><CR><LF>

Or<CR><LF><Status><CR><LF>

Or both

token	Description
<CR><LF>	<CR><LF>
+CMD:	Corresponding instruction string characters
Para-1,para-2,...,para-n	Returning to the current set parameters
STATUS	Instruction execution status "OK": indicates successful execution of the instruction "ERROR": Indicates successful instruction execution

AT instruction

Basic instructions

Description	Switch Bluetooth role mode.
Syntax	AT+Role=<para-1><CR><LF>
Write cmd	AT+Role=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+Role=1 //Central mode AT+Role=0 //Peripheral mode
Query cmd	AT+Role=?<CR><LF> Return <CR><LF>+Role:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1> 0: means Peripheral role. 1: means Central role

Description	Read Firmware version number
Syntax	AT+VER=?<CR><LF>
Query	AT+VER=?<CR><LF> Return <CR><LF>+VER:<para-1> <CR><LF><CR><LF>OK<CR><LF> Para-1 is the Version number

Description	Restart module.
Syntax	AT+RESET<CR><LF> AT+RESET<CR><LF>
Execute cmd	AT+RESET<CR><LF> Return <CR><LF>OK<CR><LF>
Comment	

Description	Restore factory settings
Syntax	AT+Reset2Default<CR><LF>
Execute cmd	AT+Reset2Default<CR><LF>

	Return <CR><LF>OK<CR><LF>
Comment	After executing this command, all settings will be restored to the factory settings

Description	Set UART baudrate
Syntax	AT+BR=<para-1><CR><LF>
Write cmd	AT+BR=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+BR=9600// Set baudrate is 9600 AT+BR=921600 // Set baudrate is 115200
Query cmd	AT+BR=?<CR><LF> Return <CR><LF>+BR:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1>:UART baudrate. Value between 4800-921600 bps After setting, the module will reset and initialize.

Description	Entering the machine sleep mode
Syntax	AT+Sleep<CR><LF>
Execute cmd	AT+Sleep<CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+Sleep
Comment	Entering this command will enter deep sleep. At this point, it is necessary to set DIO13 low and then high to wake up the module.

Description	Turn on or off low-power string
Syntax	AT+LpUartMode=<para-1> <CR><LF>
Write cmd	AT+LpUartMode=<para-1> <CR><LF> Return <CR><LF>OK<CR><LF>

	For example: AT+LpUartMode=0 AT+LpUartMode=1
Query cmd	AT+LpUartMode=?<CR><LF> Return <CR><LF>+LpUartMode:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1>0 means disable. 1: means enable. When the low-power serial port is turned off, AT instructions can be executed at any time. When the low-power serial port is enabled, UART usually does not start, and GPIOXX must be set high before data can be transmitted through the serial port.

Description	Obtain MAC address
Syntax	AT+MAC=?<CR><LF>
Query cmd	AT+MAC=?<CR><LF> Return <CR><LF>+MAC:%2x:%2x:%2x:%2x:%2x:%2x<CR><LF> <CR><LF>OK<CR><LF>
Comment	

Description	Set transmission power
Syntax	AT+TXPW=?<CR><LF> AT+TXPW=<para-1><CR><LF>
Query cmd	AT+TXPW =?<CR><LF> Return <CR><LF>+TXPW:<para-1><CR><LF><CR><LF>OK<CR><LF>
Write cmd	AT+TXPW =<para-1><CR><LF> Return <CR><LF>OK<CR><LF>
Comment	<para-1> #define MINUS_20_DBM 0 #define MINUS_18_DBM 1

	#define MINUS_15_DBM	2
	#define MINUS_12_DBM	3
	#define MINUS_10_DBM	4
	#define MINUS_9_DBM	5
	#define MINUS_6_DBM	6
	#define MINUS_5_DBM	7
	#define MINUS_3_DBM	8
	#define 0_DBM	9
	#define 1_DBM	10
	#define 2_DBM	11
	#define 3_DBM	12
	#define 4_DBM	13
	#define 5_DBM	14
	#define 6_DBM	15
	#define 7_DBM	16
	#define 8_DBM	17

Description	Modify Connection Interval
Syntax	AT+CONNITV=<para-1>,<para-2>,<para-3>,<para-4><CR><LF>
Write cmd	AT+CONNITV=<para-1>,<para-2>,<para-3>,<para-4><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+CONNITV=1600,1600,0,1500
Query cmd	AT+CONNITV=?<CR><LF> <CR><LF>CONNITV:<para-1>,<para-2>,<para-3>,<para-4><CR><LF> <CR><LF>OK<CR><LF> For example: AT+CONNITV=? +CONNITV:800,800,1,800 OK
Comment	This command is valid in the connection,At the current command, it will negotiate with the main node to change the connection interval <para-1>: intervalMin,value between 6-3200,per units of 1.25ms <para-2>: intervalMax, value between 6-3200,per units of 1.25ms <para-3>: connLatency,range 0-499 <para-4>: connTimeout,value between 10 to 3200 (in units of 10 ms) $LSTO \text{ (time-out)} > (1 + \text{Peripheral Latency}) * (\text{Connection Interval} * 2)$

Peripheral role instructions

Description	Reading and Setting Broadcast Names
Syntax	AT+NAME=?<CR><LF> AT+NAME=<para-1><CR><LF>
Query cmd	AT+NAME=?<CR><LF> Return <CR><LF>+NAME:<para-1><CR><LF><CR><LF>OK<CR><LF>
Write cmd	AT+NAME=<para-1><CR><LF> Return <CR><LF>OK<CR><LF>
Comment	<para-1> is the string type and the Bluetooth device name

Description	Set the module for feature UUID to 16bits or 128bits
Syntax	AT+UUIDMode=?<CR><LF> AT+UUIDMode=<para-1><CR><LF>
Query cmd	AT+UUIDMode=?<CR><LF> Return <CR><LF>+UUIDMode:<para-1><CR><LF><CR><LF>OK<CR><LF>
Write cmd	AT+UUIDMode=<para-1> <CR><LF> Return <CR><LF>OK<CR><LF>
Comment	<para-1> 1:128bits 0:16bits.

Description	128bits Service UUID settings
Syntax	AT+ServiceUUID128=?<CR><LF> AT+ServiceUUID128=<para-1><CR><LF>
Query cmd	AT+ServiceUUID128=?<CR><LF> Return <CR><LF>+ ServiceUUID128:<para-1><CR><LF><CR><LF>OK<CR><LF>
Write cmd	AT+ServiceUUID128=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+ ServiceUUID128=12345678901234567890aabbccddeeff
Comment	<para-1>is a 128bit UUID

Description	128bits NOTIFY UUID settings.
Syntax	AT+NotifyUUID128=?<CR><LF> AT+NotifyUUID128=<para-1><CR><LF>
Query cmd	AT+NotifyUUID128=?<CR><LF> Return <CR><LF>+ NotifyUUID128:<para-1> <CR><LF><CR><LF>OK<CR><LF>
Write cmd	AT+NotifyUUID128=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+AT+NotifyUUID128=12345678901234567890aabbccddeeff
Comment	<para-1> is a 128bit UUID

Description	128bits WRITE UUID settings.
Syntax	AT+WriteUUID128=?<CR><LF> AT+WriteUUID128=<para-1><CR><LF>
Query cmd	AT+WriteUUID128=?<CR><LF> Return <CR><LF>+ WriteUUID128:<para-1><CR><LF><CR><LF>OK<CR><LF>
Write cmd	AT+WriteUUID128= <para-1> <CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+WriteUUID128=12345678901234567890aabbccddeeff
Comment	<para-1> is a 128bit UUID

Description	16bits Service UUID settings.
Syntax	AT+ServiceUUID=?<CR><LF> AT+ServiceUUID=<para-1><CR><LF>
Query cmd	AT+ServiceUUID=?<CR><LF> Return <CR><LF>+ServiceUUID:<para-1><CR><LF><CR><LF>OK<CR><LF>
Write cmd	AT+ServiceUUID= <para-1> <CR><LF> Return

	<code><CR><LF>OK<CR><LF></code> For example: <code>AT+ServiceUUID128=1234</code>
Comment	[para 1] is a 16bit UUID

Description	16bits NOTIFY UUID settings.
Syntax	<code>AT+NotifyUUID=?<CR><LF></code> <code>AT+NotifyUUID=<para-1><CR><LF></code>
Query cmd	<code>AT+NotifyUUID=?<CR><LF></code> Return <code><CR><LF>+NotifyUUID:<para-1><CR><LF><CR><LF>OK<CR><LF></code>
Write cmd	<code>AT+NotifyUUID=<para-1><CR><LF></code> Return <code><CR><LF>OK<CR><LF></code> For example: <code>AT+AT+NotifyUUID=1234</code>
Comment	<para-1> is a 16bit UUID

Description	16bits WRITE UUID settings.
Syntax	<code>AT+WriteUUID=?<CR><LF></code> <code>AT+WriteUUID=<para-1><CR><LF></code>
Query cmd	<code>AT+WriteUUID=?<CR><LF></code> Return <code><CR><LF>+WriteUUID:<para-1><CR><LF><CR><LF>OK<CR><LF></code>
Write cmd	<code>AT+WriteUUID=<para-1><CR><LF></code> Return <code><CR><LF>OK<CR><LF></code> For example: <code>AT+AT+WriteUUID=1234</code>
Comment	<para-1> is a 16bit UUID

Description	Set broadcast interval.
Syntax	<code>AT+ADITV=?<CR><LF></code> <code>AT+ADITV=<para-1><CR><LF></code>
Query cmd	<code>AT+ADITV=?<CR><LF></code> Return

	<CR><LF>+ADITV:<para-1>,<para-2><CR><LF><CR><LF>OK<CR><LF>
Write cmd	AT+ADITV=<para-1>,<para-2><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+ADITV =80,160 80*0.625 = 50ms 160*0.625 = 100ms The interval is between 50 and 100
Comment	<para-1>Adv interval min value. <para-2>Adv interval max value. <para-1> and <Para-2> *0.625 ms

Description	Turning on and off broadcasting
Syntax	AT+ADV=<para-1>,<para-2><CR><LF>
Write cmd	AT+ADV=<para-1>,<para-2><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+ADV=1 /*Turn on broadcast*/ AT+ADV=0 /*Turn off broadcasting*/
Comment	<para-1>:1 means enable. 0 means disable.

Description	Custom Adv mode selection
Syntax	AT+BeaconMode=<para-1><CR><LF>
Write cmd	AT+BeaconMode=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+ BeaconMode =0// Set the built-in connectable broadcast mode AT+BeaconMode =1// Set non connectable beacon mode AT+BeaconMode =2// Set custom connectable ADV mode
Query cmd	AT+BeaconMode=?<CR><LF>

	Return <CR><LF>+BeaconMode:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1>: Select ADV mode. 0: Represents the built-in Adv+ScanRSP mode in the system, which is connectable and has the following packet format: Adv packet:02,01,06,03,02,f0,ff ScanRsp is the device name written using AT+Name 1: Represented as unconnectable Beacon mode, At this point, the ADV packet can use AT+AdvInd=[para 1] to write data. If you want to display ScanRSP simultaneously, you can use AT+ScanRspEn=1 to turn it on, and use AT+ScanRsp=[para 1] to set the content of the ScanRsp package. 2: Represented as a connectable ADV mode, If the built-in format of AT+BeaconMode=0 used cannot meet the user's needs, this module can be used to customize the connectable broadcast mode. At this point, the ADV packet can use AT+AdvInd=[para 1] to write data. If you want to display ScanRSP simultaneously, you can use AT+ScanRspEn=1 to turn it on, and use AT+ScanRsp=[para 1] to set the content of the ScanRsp package.

Description	Turn on and off ScanRSP packet format Only valid when AT+BeaconMode=1 or=2
Syntax	AT+ScanRspEn=<para-1><CR><LF>
Write cmd	AT+ScanRspEn=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+ScanRspEn=0// Close ScanRSP AT+ScanRspEn=1// Enable ScanRSP
Query cmd	AT+ScanRspEn=?<CR><LF> Return <CR><LF>+ScanRspEn:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	This instruction is only valid when AT+BeaconMode=1 or=2.

	When [para 1]: 0 indicates that ScanRSP is turned off When [para 1]: 1 indicates that ScanRSP is enabled When enabled, users can set the ScanRSP packet content through AT+ScanRsp=[para 1]
--	--

Description	Set Adv or beacon packet Only valid when AT+BeaconMode=1 or=2
Syntax	AT+AdvInd=<para-1><CR><LF>
Write cmd	AT+AdvInd=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+AdvInd=0bff00010203040506070809
Query cmd	AT+AdvInd=?<CR><LF> Return <CR><LF>+AdvInd:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	This instruction is only valid when AT+BeaconMode=1 or=2

Description	Set ScanRsp packet Only valid when AT+BeaconMode=1 or=2 and ScanRspEn=1
Syntax	AT+ScanRsp=<para-1><CR><LF>
Write cmd	AT+ScanRsp=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+ScanRsp=0bff00010203040506070809
Query cmd	AT+ScanRsp=?<CR><LF> Return <CR><LF>+ScanRsp:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	This instruction is only valid when AT+BeaconMode=1 or=2 and ScanRspEn=1

Description	Turn pairing mode on and off
Syntax	AT+PairingMode=<para-1><CR><LF>
Write cmd	AT+PairingMode=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+PairingMode=1 //enable pairing mode AT+PairingMode=0 //disable pairing mode
Query cmd	AT+PairingMode=?<CR><LF> Return <CR><LF>+ScanRsp:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1> 0:means disable. 1:means enable

Description	Set pairing password
Syntax	AT+PairingKey=<para-1><CR><LF>
Write cmd	AT+PairingKey=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+PairingKey=123456 AT+PairingKey=2468
Query cmd	AT+PairingKey=?<CR><LF> Return <CR><LF>+PairingKey:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1> Pairing password Paired password with a maximum length of six digits When pairing on Android or iOS, assuming that the password you set does not exceed 6 addresses, please fill in 0 to meet 6 digits during pairing For example, when set to AT+PairingKey=2468 In actual pairing, 002468 should be entered

Central role instructions

Description	Setting scan window timeout.
Syntax	AT+ScanWindow=<para-1><CR><LF>
Write cmd	AT+ScanWindow=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+ScanWindow=5 //5*1000s = 5 seconds AT+ScanWindow=10 //10*1000 = 10 seconds
Query cmd	AT+ScanWindow=?<CR><LF> Return <CR><LF>+ScanWindow:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1> *1000ms scan window time. 0 means continues scan <para-1> value between 0-127. Please note that if you execute peripheral instructions in the Central role or execute Central instructions in the Peripheral role, ERROR_INCORRECT_MODE will be returned.

Description	Start and stop scanning.
Syntax	AT+SCAN=<para-1><CR><LF>
Write cmd	AT+SCAN=<para-1><CR><LF> For example: AT+SCAN=0 //means Stop scan AT+SCAN=1 //means Start scan
Comment	When executing AT+SCAN=0, <CR><LF>+EVT:SCAN-STOP<CR><LF><CR><LF>OK<CR><LF> will be returned to indicate that SCAN stops. When executing AT+SCAN=1, <CR><LF>+EVT will be returned: SCANNING<CR><LF><CR><LF>OK<CR><LF> indicating that scanning is in progress. At this time, scanning will be performed based on the ScanWindow time. If ScanWindow is not equal to 0, the return format is: <mac address> <rssi> <device name> e.g. 0x4FC67FD27ADB -57 Echo-ORM 0xA4C138A29F27 -53 LYWSD03MMC

	If ScanWindow is equal to 0, it means continuous scan, or Filter_MAC is turned on. At this time, the return format is: +EVT_RAW_PACKET:<mac address>,<rssi>,<raw data packet> e.g. +EVT_RAW_PACKET:4fc67fd27adb,-53,09094563686f2d30524d020a01 +EVT_RAW_PACKET:a4c138a29f27,-60,0201060f1695fe30585b05ab279fa238c1a408 +EVT_RAW_PACKET:a4c138a29f27,-60,0b094c5957534430334d4d43
--	--

Description	Set and get RSSI filters.
Syntax	AT+RSSI=<para-1><CR><LF>
Write cmd	AT+RSSI=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+RSSI=-60 AT+RSSI=-101
Query cmd	AT+RSSI=?<CR><LF> Return <CR><LF>+RSSI:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	The <para-1> between -120 ~ -30dBm, When <para-1> less than -100, the RSSI filtering will turn off. When the RSSI filter is turned on, the module will determine whether the RSSI of each advertising packet is within the set range. If so, it will output it to the UART.

Description	Turn on and off MAC filter.
Syntax	AT+EN_MAC_Filter =<para-1><CR><LF>
Write cmd	AT+EN_MAC_Filter=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+EN_MAC_Filter =1 //enable MAC filter. AT+EN_MAC_Filter=0 // disable MAC filter.

Query cmd	AT+EN_MAC_Filter=?<CR><LF> Return <CR><LF>+ EN_MAC_Filter:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1> 0 means disable. 1 means enable. When the MAC filter is turned on, we need to use the AT+MAC_Filter command to set the MAC address. Then when executing SCAN, the module will only display the MAC address that matches the filter setting on the UART. Note: When MAC_Filter is turned on, the module will automatically turn off DNAME_Filter.

Description	Turn on and off device name filter.
Syntax	AT+EN_DNAME_Filter =<para-1><CR><LF>
Write cmd	AT+EN_DNAME_Filter=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+EN_DNAME_Filter =1 //enable device name filter. AT+EN_DNAME_Filter=0 // disable device name filter.
Query cmd	AT+EN_DNAME_Filter=?<CR><LF> Return <CR><LF>+EN_DNAME_Filter:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1> 0 means disable. 1 means enable. When the device name filter is turned on, we need to use the AT+DNAME_Filter command to set the device name. Then when executing SCAN, the module will only display the device name that matches the filter setting on the UART. Note: When DNAME_Filter is turned on, the module will automatically turn off MAC_Filter.

Description	Get and set MAC filter.
Syntax	AT+MAC_Filter=<para-1><CR><LF>
Write cmd	AT+MAC_Filter=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+MAC_Filter=4FC67FD27ADB
Query cmd	AT+MAC_Filter=? <CR><LF> Return <CR><LF>+ MAC_Filter:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1> as 12-character MAC address. After we set the MAC address and turn on the MAC filter through AT+EN_MAC_Filter, the module will filter the data for this MAC address before outputting it to the UART.

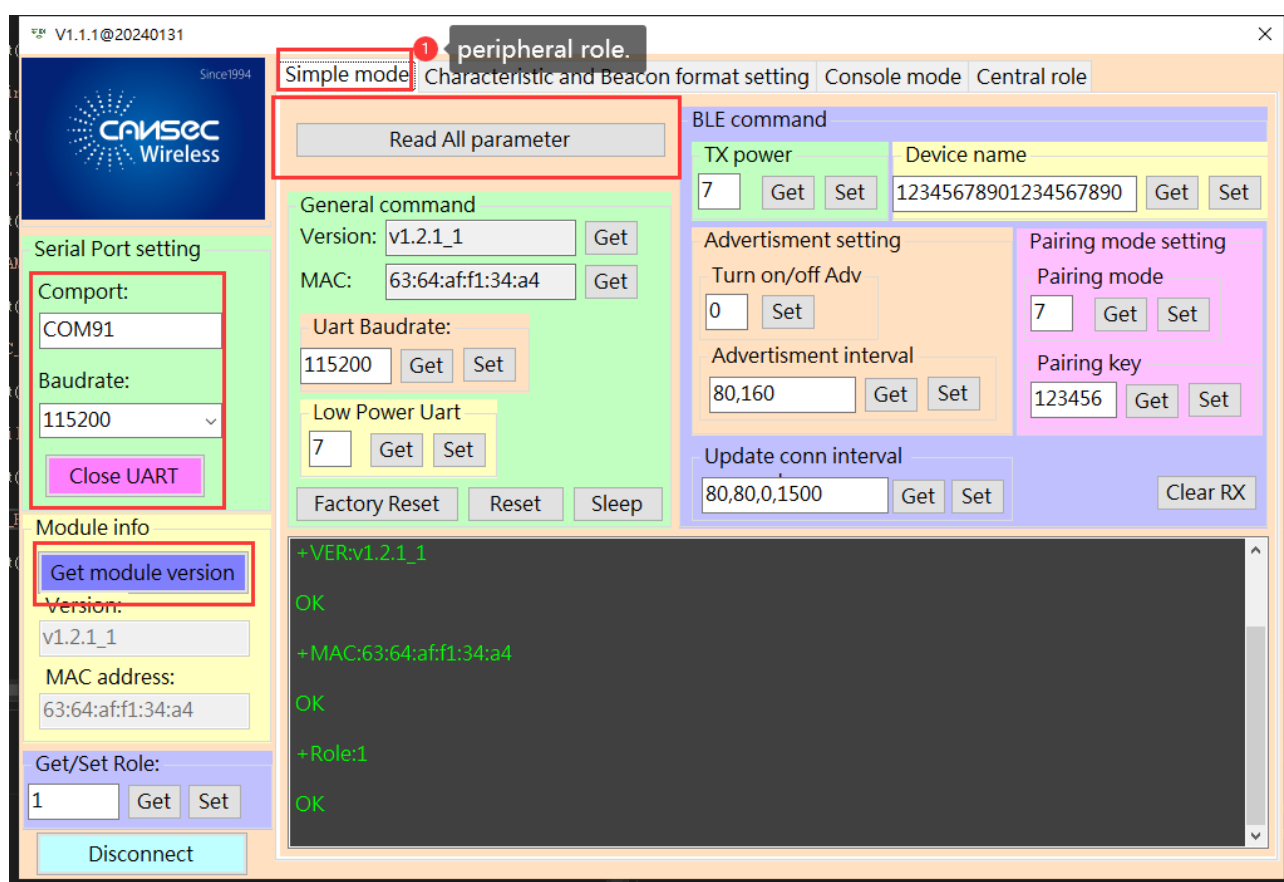
Description	Get and set Device name filter.
Syntax	AT+DNAME_Filter=<para-1><CR><LF>
Write cmd	AT+DNAME_Filter=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+DNAME_Filter=Echo-ORM
Query cmd	AT+DNAME_Filter=? <CR><LF> Return <CR><LF>+DNAME_Filter:<para-1><CR><LF> <CR><LF>OK<CR><LF>
Comment	<para-1> This is a string, with a maximum length of 30 characters, used as the name of the device to be filtered. When we set the device name to be filtered, set AT+EN_DNAME_Filte=1, and start executing SCAN, UART will only output matching device names.

Description	Perform a connection to the specified MAC address.
Syntax	AT+Conn=<para-1><CR><LF>
Write cmd	AT+Conn=<para-1><CR><LF> Return <CR><LF>OK<CR><LF> For example: AT+Conn=A434F1AF6497
Comment	<para-1> as 12-character MAC address. When we execute AT+Conn=<MAC address>, the module will try to connect to this peripheral. When the connection is successful, it will return +EVT:CONNECTED:<MAC address><CR><LF>. When the connection is successful, transparent transmission can be performed with peripheral.

Quick Start Guide

Our company provides a set of upper computer programs aimed at allowing users to quickly get started using the AN2340 transparent module. After the transparent module is connected to the computer through the serial port dongle, it is turned on CC2340_Utility.exe

As shown in the following figure, input the serial port's comport, and the baudrate is preset to 115200. Then, click Open UART. Then, we click Get module version. If the publication number and MAC address are displayed, it indicates that the connection with the module has been successfully completed. Then, we click Read All parameter. At this point, the upper computer will send all AT commands to obtain all the settings of the module and display them on the upper computer. At this time, we can make the necessary settings, Click on the Set button to save the settings in the module



The following page mainly discusses the UUID module settings and characteristic settings. We can choose to use 16bit or 128bit characteristics.

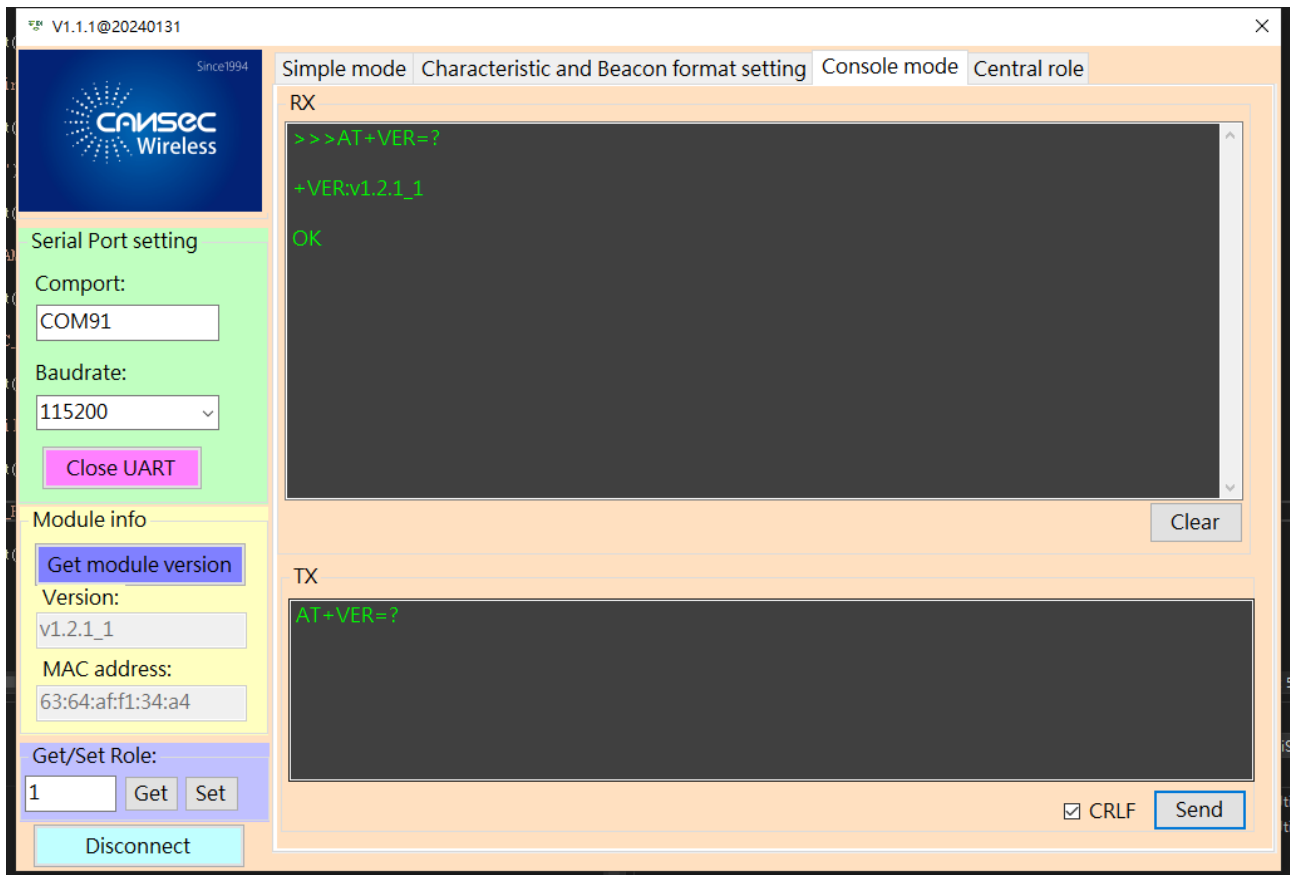
In Beacon mode, users can set the ADV packets that need to be defined and whether to enable ScanRsp and ScanRsp content after using Beacon mode.

The screenshot displays the CANSEC Wireless software interface, version V1.1.1, dated 20240131. The interface is divided into several sections:

- Serial Port setting:** Includes fields for Comport (COM91) and Baudrate (115200), with a 'Close UART' button.
- Module info:** Includes a 'Get module version' button, showing Version: v1.2.1_1 and MAC address: 63:64:af:f1:34:a4.
- Get/Set Role:** Includes a dropdown menu set to '1' and 'Get'/'Set' buttons.
- Disconnect:** A button at the bottom left.
- Simple mode:** The active tab, showing 'Characteristic and Beacon format setting'.
- modify UUID length:** A section with two modes:
 - 16bits mode:** Includes 'Service UUID 16bits' (1234), 'Notify UUID 16bits' (1234), and 'Write UUID 16bits' (1234), each with 'Get' and 'Set' buttons.
 - 128bits mode:** Includes 'Service UUID 128bits' (12345678901234567890aabbccddeeff), 'Notify UUID 128bits' (12345678901234567890aabbccddeeff), and 'Write UUID 128bits' (12345678901234567890aabbccddeeff), each with 'Get' and 'Set' buttons.
- Beacon mode and beacon data:** Includes 'beacon mode' (7) and 'Beacon data' (07ff52462d494448), both with 'Get' and 'Set' buttons.
- Scan RSP mode and Scan RSP data:** Includes 'ScanRSP Enable' (7) and 'Scan RSP data' (07ff52462d494448), both with 'Get' and 'Set' buttons.

A red circle with the number '1' and the text 'peripheral role.' is overlaid on the 128bits mode section.

And users can also use console mode to input AT commands themselves.



The form below is mainly related to the setting options related to Central role.

V1.1.1@20240131

Since1994

CANSEC
Wireless

Serial Port setting

Comport:
COM91

Baudrate:
115200

Close UART

Module info

Get module version

Version:
v1.2.1_1

MAC address:
63:64:af:f1:34:a4

Get/Set Role:
1

Disconnect

Simple mode

Characteristic and Beacon format setting

Console mode

Central role

OK

+EN_DNAME_Filter:1

OK

+EVT:SCANNING

OK

0x4FC67FD27ADB -60 Echo-ORM

OK

Central role AT command

ScanWindow:
5

MAC filter:
MAC: a434f1af6497

Do Scan:
Start Stop

RSSI filter:
-100

do Connection:
A434F1AF6497

Enable or Disable: 0

Device name filter:
NAME: Echo-ORM

Enable or Disable: 0

Console
Hello world

Clear Screen

Send data

Apps

We have transparent firmware and can support TI simple link apps.

TI Simple Link is a Bluetooth app developed by TI, which provides the original code and allows users to conduct secondary development.

TI SimpleLink Connect on Apple App Store:

<https://apps.apple.com/no/app/simplelink-connect/id6445892658>

TI SimpleLink Connect on Google App Store:

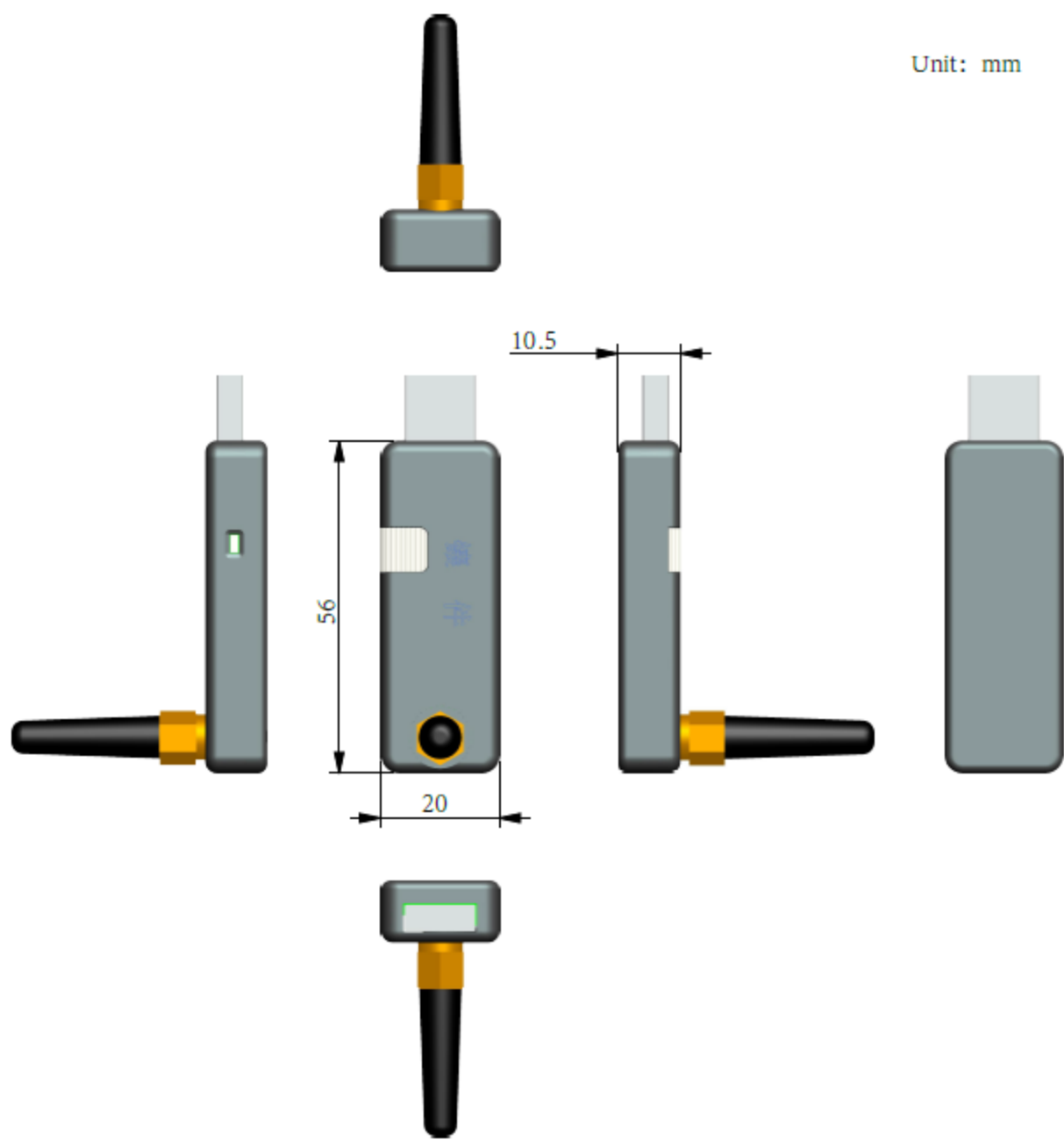
<https://play.google.com/store/apps/details?id=com.ti.connectivity.simplelinkconnect>

SimpleLink app source code download.

TI SimpleLink Connect sources:

<https://github.com/TexasInstruments/simplelink-connect>

Mechanical Drawing



Overview

The BTDG2340 is BLE USB Dongle, a cost-effective, low-power, true system-on-chip (SoC) for Bluetooth low energy applications, targeting Bluetooth® 5.3 Low Energy, Zigbee, Thread, and Proprietary 2.4GHz applications. It has two user-programmable LED, a MCU_RESET push button. or more information, visit www.rf-products.com, or contact staff if needed.

Contact Details

Tel: +86-10-6888 9971

Email: sarolyn@rf-products.com

grace@rf-products.com

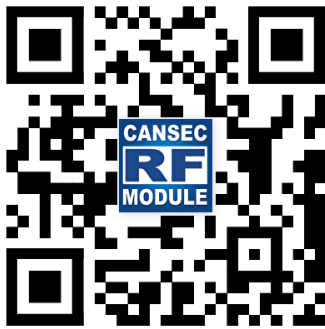
norman@rf-products.com

Technical Support: tech@rf-products.com

Addr.: Rm.1002, Block B China Railway Venture building,

No.28 Pingguoyuan Rd., Shijingshan District, Beijing, 100041, China.

Website: www.rf-products.com



<http://www.rf-products.com>



<https://cansec.en.alibaba.com/>



<http://shop144152941.taobao.com/>