TL8251-D1 Bluetooth Module User Manual

Document No.: E-32600289

Version: A.4

Compiled by			Date	
Reviewed by			Date	
			Date	
Approved by			Date	

History of File Changes			
Rev.	Record of Changes	Prepared by	Date
A.1	Newly added	Tang Chunlong	December 6, 2019
A.2	1. Multiplex the SWS pin with the DATAOUT pin. 2. Update the test data of power consumption.	Tang Chunlong	March 20, 2020
A.3	1. Modify the upper limit of the single transparent transmission data from “216 bytes” to “214 bytes”. 2. Modify the adaptive software version from “V0.2 and above” to “V0.2.3”. 3. Modify the maximum length of user-defined data of broadcast data in the setAdvData settings from “28 bytes” to “30 bytes”. 4. Add “Note: Point 3” in “4.2 Pin Function Description” for the LINK to waken master MCU.	Chen Feng	January 6, 2021
A.4	1. Add the FCC Warnings, and change the information on the sign; 2. The parameter in “setDataDir” of the AT command set is changed to “0” by default.	Li Sha	March 31, 2021

Contents

Adaptive Software Version: V0.2.3

1. Overview

The Bluetooth module is a low-power single-mode transparent transmission Bluetooth module based on Bluetooth 5.0 (downward compatible with Bluetooth 4.2) of the TLSR8251 design. This document is a direction for use of the Bluetooth module, including main functions, application scenarios, methods of use, logical structure, hardware interface and various indicators characteristics of the module.

2. Function

Communication function: The module can transparently transmit the data received by the module serial port to the APP through the BLE wireless channel; and the APP can also transparently transmit data through the module serial port, and the upper limit of single transparent transmission data is 214 bytes.

Bluetooth function: Bluetooth adopts the BLE single mode to realize the transparent transmission and interaction of data between the external device of the Bluetooth module and the APP, with the transmitting power of 0dBm.

Configuration function: The external device can set (get) the parameters of the module through AT command of the serial port.

3. Block Diagram of System

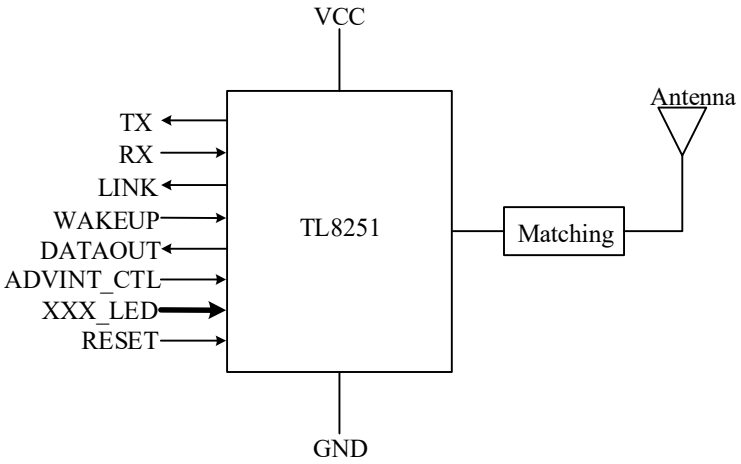


Figure 1 Block Diagram of System

4. Definition of Pin and Description of Function

4.1 Definition of Module Pin

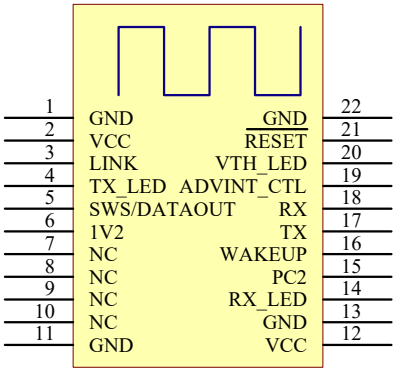


Figure 2 Definition of Module Pin

4.2 Description of Pin Function

Table 1 Description of Module Pin Function

S/N of Pin	Name of Pin	Category	Description
1,11,13,22	GND	PWR	Power ground (These pins have been short-circuited on the module.)
2,12	VCC	PWR	Positive pole of power supply (These pins have been short-circuited on the module.)
3	LINK(Note: 3)	O	Module connection status indication (can be used to waken the main control MCU): 0: The module is connected. 1: The module is not connected. The internal 10KΩ is pulled up.
4	TX_LED	I	Module RF data sending prompt port, the internal 10 KΩ is pulled up, and the initial state is at a high level. When the module sends data to mobile devices such as cellphones, the output frequency of this pin is 250 Hz, and the duty ratio is of the 50% square wave. After the data transmission is completed, stop outputting the square wave. The waveform output function of this pin is only specific to the modules that have enabled the RF data receiving/sending prompt function. For the method of starting and stopping RF data receiving/sending prompt, please refer to the AT command.
5	SWS/	I/O	The internal 1MΩ is pulled up.

	DATAOUT(Note: 3)		SWS: Programming DATAOUT: Data output indication from the module serial port (can be used to waken the main control MCU): 0: The module has data to send. 1: The module has no data to send.
6	1V2	PWR	Internally connect to the chip VDD1V2
7,8,9,10	NC	NC	There is no internal connection.
14	RX_LED	I/O	Module RF data receiving prompt port, the internal 10 KΩ is pulled up, and the initial state is at a high level. When the module received the data from mobile devices such as cellphones, the output frequency of this pin is 250 Hz, and the duty ratio is of the 50% square wave. After the data transmission is completed, stop outputting the square wave. The waveform output function of this pin is only specific to the modules that have enabled the RF data receiving/sending prompt function. For the method of starting and stopping RF data receiving/sending prompt, please refer to the AT command.
15	PC2	I/O	Internally connect to the chip PC2.
16	WAKEUP(Note: 2)	I	Module waken pin. When the internal 10KΩ is pulled up, this pin needs to be pulled down before sending serial port data to the module, and the data is sent after a delay of 5ms. During the sending period, the level needs to keep low. After sending, pull up this pin to make the module go to sleep again.
17	TX	O	Module serial port sending pin, with the internal 10KΩ being pulled up.
18	RX	I	Module serial port receiving pin, with the internal 10KΩ being pulled up.

<table>
<tr><td>19</td><td>ADVINT_CTL(Note: 1)</td><td>I</td><td>Broadcast interval time switch control port, with the internal 1MΩ being pulled up, and the initial state is of the high level. If there is no requirement for hard adjustment of the broadcast interval, connect this pin to the ground.
<table>
<tr><th>Pin Level</th><th>Broadcast Interval</th><th>Connection Interval</th></tr>
<tr><td>0</td><td>200ms</td><td>Lat is 0</td></tr>
<tr><td>1</td><td>1s</td><td>Lat is 8</td></tr>
</table></td></tr>
<tr><td>20</td><td>VTH_LED</td><td>I/O</td><td>The module low battery prompt port, pull up the internal 10 KΩ, and the initial state is at a high level.
When the module detects that the supply voltage is equal to or lower than the threshold voltage Vth, the output frequency of this pin is 2.5Hz, and the duty ratio is of 50% square wave. The waveform output function of this pin is only specific to the modules that have enabled the low battery prompt function. For the low battery prompt method, please refer to the AT command.</td></tr>
<tr><td>21</td><td>RESET</td><td>I</td><td>The main control MCU can reset the module through this pin, and it is effective at the low level.
The effective reset signal duration needs to be equal to or longer than 50ms.</td></tr>
</table>

Notes:

1. When the module is powered on, the broadcast will be turned on;
2. When WAKEUP is not pulled down, the module can receive the data sent from the mobile terminal.
3. If the Bluetooth module needs to use the uninterruptible power broadcast mode in the application, you can use the LINK pin or DATAOUT to waken the main control MCU and exit the sleep mode.

   It is recommended to use the following methods:

   1) If the main control MCU is in intermittent sleep mode, you can use the LINK pin to assist in wakening the main control MCU;
   2) If any IO port of the main control MCU can wake up the MCU, the LINK pin can be used to connect with any IO port so as to waken the main control MCU;
   3) If the main control MCU only supports external interrupt IO port to waken the MCU:
      a) You can choose to connect the LINK pin to the external interrupt IO of the main control MCU;

b) You can choose to connect the DATAOUT pin to the external interrupt IO of the main control MCU;

4) When using the LINK pin to waken the main control MCU, it can support the function of actively sending data by the MCU. If the connection time is too long, it will actively turn off or disconnect Bluetooth and other applications. However, as the main control MCU was woken up too early, the power consumption will increase accordingly. When using the DATAOUT pin to waken the main control MCU, the main control MCU will be woken up only when there is data sent to the MCU from the Bluetooth end. This mode requires the APP end to cooperate with other specific scenario developments.

5. Initial Configuration Parameters

Table 2 Initial Parameter List of Module

Name of Parameter	Initial Parameter	Remarks
Name of Module	Sinocare	
Baud Rate	9600bps Data bit: 8 Stop bit: 1 No parity check	
Broadcast Interval	200ms	The ADVINT_CTL pin needs to be pulled down.
Connection Interval	150ms	
Transmitting Power	0dbm	
TX Latency Sending Time	5ms	

6. UUID

The data transparent transmission function is realized by SPP Service/Profile. The UUID related to SPP Service is as shown in the following table.

Type	UUID	Property
SPP Service	0xFFB0	NC
SPP Data Characteristic	0xFFB2	Write without Response, Notify
SPP Command Characteristic	0xFFB1	Write, Notify

SPP Data Characteristic is used to realize the transparent transmission of data. The properties of characteristic are Write without Response and Notify. Before data transmission, the property of Notify of SPP Data Characteristic shall be enabled, that is, the value of Client Characteristic Configuration of SPP Data Characteristic shall be changed to 0x0001.

The relevant UUID for battery power acquisition is as shown in the table below.

Type	UUID	Property
Battery Service	0x180F	NC
Battery Level	0x2A19	Read, Notify

The battery level between the corresponding voltage range of 0 and 100% is 1.8 to 3.3V.

Note: The battery power acquisition function needs to turn on the low battery prompt function first. For details, see the openVth command. Otherwise, the returned power is 100%.

7. AT Command

7.1 Command Format

The command data of the module is ASCII code. It consists of four parts, as shown in the figure below.

Start mark of the command	Command code	Parameter list	End mark of the command

- Start mark of the command: AT+

These three consecutive characters are used to indicate the beginning of a command packet.

- Command code

It is the specific identification of the command, and is used to distinguish different commands.
a) Characters irrelevant to the command code are not allowed to appear between the characters of the command code;
b) Spaces or other characters irrelevant to the command code are not allowed to appear between the start mark of the command and the command code.

- Parameter list

Different commands have different parameters. The parameters of the command need to be separated by spaces. The command code and the parameter list also need to be separated by spaces.
a) Except for the command code with special requirements or instructions (for example, "=" must be used between the setName parameter list and the command code), the command code and the parameter list must be separated by a space;
b) For the multi-parameter command code, the parameters must be separated by a space (such as setConnlnt);
c) Unless there are special requirements or instructions (such as setName) for the single-parameter command code, no spaces or characters irrelevant to the parameter are allowed to appear in the parameter.

- End mark of the command: \r\n

Two characters (Enter, line break) are used to indicate the end of a command packet.
If the command code is a setting type (setXXXX) command, the command end mark must closely follow the parameter list, and no characters irrelevant to the parameter are allowed to appear between the command end mark and the parameter list;
If the command code is a getting type (getXXXX) command, the command end mark must closely follow the command code.

Example: Set the baud rate of the module to be 9600: "AT+setBR 9600\r\n"
Wherein, "AT+" is the start mark of the command;
"SetBR" is the command code, which indicates the operation of switching the baud rate;
"9600" is a parameter, which indicates that the baud rate for the serial port of the module is set to be 9600bps;
"\r\n" is the end mark of the command.

7.2 Restrictions

(1) A complete command shall be sent to the module continuously. If the command cannot be sent continuously and becomes multiple packets to be sent to the module, the data interval between packets must be less than 80ms and the total frame reception time shall be less than 500ms. If the total frame reception time expires, the received data is transparently transmitted;

(2) AT+XXX\r\n, wherein XXX needs to satisfy the characters specified in AT commands such as {0~9 A~Z a~z space = : .}. If other characters appear, directly perform transparent transmission of the module;
(3) AT+ nesting rules: The module can identify the closest AT command. For example: For AT+XXXAT+YYY\r\n, AT+YYY\r\n shall be identified, and AT+XXX shall be transparently transmitted;
(4) \r\n nesting rules: The module can identify the closest \r\n. For example: For AT+XXX \r\nYYY\r\n, AT+XXX \r\n shall be identified, and YYY\r\n shall be transparently transmitted;
(5) The data between "AT+" and "\r\n" cannot exceed 50 bytes, otherwise directly perform transparent transmission;
(6) The command code of the AT command needs to have the same character size before it can be parsed. For example: The "AT+setbr 115200\r\n" data packet may be interpreted as an error command, because setbr (br is not capitalized) is not a correct command code;
(7) When the MCU sends multiple AT commands to the module at the same time, the module only identifies the last command.

7.3 Command Set Sent by the External MCU to the Module

Table 3 Command Set of Module

Command Code	Function
setName	Set the name of module
getName	Get the name of module
setBR	Set the baud rate
getBR	Get the baud rate
setTxDly	Set the serial port output latency
getTxDly	Get the serial port output latency
setDBM	Set the transmitting power
getDBM	Get the transmitting power
setAdvInt	Set the broadcast interval
getAdvInt	Get the broadcast interval
setAdvData	Set the broadcast data
getAdvData	Get the broadcast data
setConnInt	Set the connection parameters
getConnInt	Get the connection parameters
getAddr	Get the module address
disconnect	Disconnect
getStatus	Inquire the current status of the module
saveConfigure	Save the current configuration
clearConfigure	Clear the saved configuration
getVersion	Get the firmware version number of the module
getRemoteAddr	Get the address of the other party's device to be connected

setConnIntEx	Offline configure the connection parameters
getConnIntEx	Get the currently connected parameter configuration
setAuthMode	Set the WeChat authentication method
getAuthMode	Get the WeChat authentication method
setMD5	Set the MD5 code
getMD5	Get the MD5 code
setDevID	Set the device ID
getDevID	Get the device ID
setAesKey	Set the AES128 key
getAesKey	Get the AES128 key
setDataDir	Set the data direction
getDataDir	Get the data direction
closePC	Turn off the pairing code function
setPC	Set and turn on the pairing code function
getPC	Inquire the status and the function of the pairing code
getCalInfo	Get the module calibration information
setVth	Set the low battery prompt threshold voltage of the module
getVth	Get the status of low battery prompt function and threshold voltage of the module
closeVth	Turn off the low battery prompt function of the module
openVth	Turn on the low battery prompt function of the module
closeLED	Turn off the RF data sending/receiving prompt function of the module
openLED	Turn on the RF data sending/receiving prompt function of the module
getLED	Get the RF data sending/receiving prompt function of the module

Note: For the setting commands of the module, the parameters are all valid for the corresponding time, and there is no need to save or restart for the second time.

7.4 Description for Command Sent from External MCU to Module

- setName

Purpose: Set the name of module
Number of parameters: 1
Parameter value:

Parameter Value (Name)	Meaning
=string	string: It is a specific name string

Notes:

The parameter of this command is the content between "=" and "\r\n". Spaces are only allowed to appear in the middle of the name, they are not allowed to appear at the beginning and the ending, and the maximum length of the name cannot exceed 20 bytes;

If the end of the set command contains multiple "\r\n", the module will use the first "\r\n" as the end mark of the command, and the data following that will be transparently transmitted;

Example:

Set the name of module to be "Sinocare": "AT+setName=Sinocare\r\n"

Response:

"AT+ok\r\n" (operate successfully)

"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getName

Purpose: Get the name of module

Number of parameters: None

Example:

"AT+ getName\r\n"

Response:

"AT+ok=Sinocare\r\n" (operate successfully, and the device name is "Sinocare")

"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setBR

Purpose: Set the baud rate (bps) of the module

Number of parameters: 1

Parameter value:

Parameter Value (bps)	Meaning
9600	Set the baud rate of the module to be 9600
19200	Set the baud rate of the module to be 19200
38400	Set the baud rate of the module to be 38400
57600	Set the baud rate of the module to be 57600
115200	Set the baud rate of the module to be 115200

Notes:

The module only supports the 5 baud rates listed in the above table, and the default baud rate is 9600. After the MCU sends this command to the module, the module returns the "AT+ok" command first, and then changes its own baud rate. After the MCU receives the "AT+ok" command, it should also change its own baud rate to keep consistent with the module to avoid error codes.

Notes: If you forget the baud rate you have set, you can try to send other "get commands" at different baud rates until you get the correct reply to determine the current baud rate of the module.

Example:

"AT+setBR 19200\r\n" (Set the baud rate to be 19200)

Response:

"AT+ok\r\n" (operate successfully)

"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getBR

Purpose: Get the baud rate (bps) of the module
Number of parameters: None
Example:
"AT+getBR\r\n"
Response:
"AT+ok 9600\r\n" (operate successfully, and the baud rate of device is 9600)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setTxDly

Purpose: Set the serial port output latency (ms) of the module
Number of Parameters: 1
Parameter Value:

Parameter Value (Latency)	Meaning
n	Delay the serial port output for n ms

Notes:

The serial port output latency is used in conjunction with the DATAOUT pin, for the purpose of allowing the MCU to have enough time to be woken up from the sleep state, so as to correctly receive the serial port data sent by the module to the MCU. When the module has serial port data to send to the MCU, it will first pull down the DATAOUT pin and delay the specified time before sending the serial port data. When the serial port data transmission is completed, the module sets the DATAOUT pin to a high level again. The setting range of latency is 0~255, and the default is 5ms. This value should not be set too large, so as to avoid overflow of the serial port buffer caused by the failure of sending out the serial port data from the module in time. The relationship diagram between the DATAOUT pin and the serial port data output is as follows:

Example:
"AT+setTxDly 10\r\n" (Set the serial port output latency of the module to be 10ms)
Response:
"AT+ok\r\n" (operate successfully)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getTxDly

Purpose: Get the serial port output latency (ms) of the module
Number of parameters: None
Example:
"AT+ getTxDly\r\n"
Response:
"AT+ok n\r\n" (n is Latency value, with the range of 0~255)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setDBM

Purpose: Set the transmitting power (dbm) of the module
Number of Parameters: 1
Parameter Value:

Parameter Value (Transmitting Power)	Meaning
0	Set the transmitting power of the module to be 0dbm
4	Set the transmitting power of the module to be 4dbm
-6	Set the transmitting power of the module to be -6dbm
-23	Set the transmitting power of the module to be -23dbm

Notes:
The default value of transmitting power for the module is 0dbm. The module only supports the four power values in the above table.
Example:
"AT+setDBM 4\r\n" (Set the transmitting power of the module to be 4dbm)
Response:
"AT+ok\r\n" (operate successfully)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getDBM

Purpose: Get the transmitting power (dbm) of the module
Number of Parameters: None
Example:
"AT+getDBM\r\n"
Response:
"AT+ok n\r\n" (n is the specific power value, only one of 0/4/-6/-23)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setAdvInt

Purpose: Set the module broadcast interval (625μs)
Number of Parameters: 2
Parameter Value:

Parameter Value:	Value:	Meaning
Parameter 1 (Minimum broadcast interval)	Min	Set the minimum broadcast interval of the module to be Min*625μs
Parameter 2 (Maximum broadcast interval)	Max	Set the maximum broadcast interval of the module to be Max*625μs

Notes:
(1) The minimum broadcast interval of the module is 320 by default, and the maximum broadcast interval is 328 by default (unit: 625μs);
(2) The effective range of the broadcast interval is 10ms~10.24s (the valid value range of the broadcast interval parameter: 16~16384). The maximum broadcast interval set shall not be less than the minimum broadcast interval;

(3) The larger the broadcast interval, the lower the power consumption of the module during broadcast. After setting the broadcast interval successfully, you need to restart the broadcast.
Example:
The command to set the minimum broadcast interval of the module to be 80*625μs and the maximum broadcast interval to be 100*625μs is: "AT+setAdvInt 80 100\r\n"
Response:
"AT+ok\r\n" (operate successfully)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getAdvInt

Purpose: Get the module broadcast interval (625μs)
Number of Parameters: None
Example:
"AT+ getAdvInt\r\n"
Response:
"AT+ok Min Max\r\n" (Min is the minimum broadcast interval, and Max is the maximum broadcast interval, unit: 625μs)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setAdvData

Purpose: Set the broadcast data
Number of Parameters: 1
Parameter Value:

Parameter Value (self-defined broadcast data)	Meaning
=data	Set broadcast data of the module is data.

Notes:
(1) Data between "=" and "\r\n" is valid data. If the end of the set command contains multiple "\r\n", the module will use the first "\r\n" as the end mark of the command, and the data following that will be transparently transmitted;
(2) The maximum length of the user-defined data is 30 bytes. It adopts the Hex format to express, and the broadcast data is blank by default;
(3) If the broadcast data is not an even number of bytes, add 0 to the upper 4 bits of the broadcast data.
Example:
"AT+setAdvData=123456789\r\n"
Set the broadcast data of the module to be:
[0x01, 0x23, 0x45, 0x67, 0x89], the 0 marked in red is automatically added by software.
If you want to remove the broadcast data, just send a null parameter: "AT+setAdvData=\r\n".
Response:
"AT+ok\r\n" (operate successfully)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getAdvData

Purpose: Get the current broadcast data
Number of Parameters: None
Notes:
(1) If you set broadcast data to non-even-number of bytes, add 0 to the upper 4 bits of the set broadcast data for the got broadcast data.
Example:
"AT+getAdvData\r\n" (Get the current broadcast data of the module)
Response:
"AT+ok=data\r\n" (data is the current broadcast data, if the broadcast data is blank, return "AT+ok=NULL\r\n")
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setConnInt

Purpose: Set the connection parameters
Number of Parameters: 4
Parameter Value:

Parameters	Value	Meaning
Parameter 1 (Minimum connection interval)	Min	Set the minimum connection interval to be Min*1.25ms
Parameter 2 (Maximum connection interval)	Max	Set the maximum connection interval to be Max*1.25ms
Parameter 3 (Latency)	Lat	Set Latency to be lat
Parameter 4 (Connection timeout)	Timeout	Set the connection timeout to be timeout*10ms

Notes:
(1) Minimum connection interval and maximum connection interval: The valid value range is: 0x0006~0x0C80, and the maximum connection interval shall not be less than the minimum connection interval;
(2) The greater the connection interval, the lower the power consumption and the lower the transmission rate.
(3) The value range of Latency is: 0x0000~0x01F3.
(4) The value range of connection timeout is: 0x000A~0x0C80.
The above parameter values must satisfy:
[Max*1.25*(1+Latency)]<10*Timeout.

When the module updates the connection parameters, it needs to wait for the other end device to participate in the response. The response time is related to the connection interval. The shorter the connection interval, the shorter the response time, and vice versa. Therefore, sometimes there is a phenomenon that the response is delayed for a few seconds. After the connection parameter is updated successfully, the module responds with the "ok command"; and the update fails or the update times out, the module responds with the "err command", and the timeout period is 10s. Since the module executes the command in a single step, it will return err when sending other commands to the module before getting a response. Therefore, after sending an updated connection parameter command to the module, you should perform other operations after receiving the response from the module.

If the module is currently in a disconnected state, using the setConnInt command will return an error:
AT+err notConn\r\n.
Example:
“AT+setConnInt 120 150 0 400\r\n” (The minimum connection interval of the module is set to be 150ms, the maximum connection interval is set to be 187.5ms, Latency is 0, and the connection timeout is 4s)
Response:
“AT+ok\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getConnInt

Purpose: Get the connection parameters
Number of Parameters: None
If the module is currently in a disconnected state, using the getConnInt command will return an error:
AT+err notConn\r\n.
Example:
“AT+getConnInt\r\n” (Get the connection parameters)
Response:
“AT+ok Interval Latency Timeout\r\n” (Interval is the actual connection interval, and Timeout is the connection timeout)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getAddr

Purpose: Get the module connection address
Number of Parameters: None
Example:
“AT+getAddr\r\n” (Get the module connection address)
Response:
“AT+ok A4:C1:38:1B:BC:48\r\n”
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- disconnect

Purpose: The module actively disconnects the connection
Number of Parameters: None
Notes:

After successfully disconnecting, the module will return to the idle state. When the command is successfully sent, the module will first return an ok response, and then return a disconn response after the connection is really disconnected.

If the module is currently in a disconnected state, using the disconnect command will return an error:
AT+err notConn\r\n.

Example:
“At+disconnect\r\n” (The module actively disconnects the connection)
Response:
“AT+ok\r\n” (operate successfully)
“At+disconn\r\n” (The connection has been disconnected)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getStatus

Purpose: Get the running status of the module
Number of Parameters: None
Notes:
There are two running status of the device: broadcast and connected (“adv” and “connected”).
Example:
“AT+getStatus\r\n” (Get the running status of the module)
Response:
“AT+ok adv\r\n” (The module is in the broadcast status)
“AT+ok connected\r\n” (The module is in the connected status)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- saveConfigure

Purpose: Save the current configuration
Number of Parameters: None
Notes:
This command is used to save the parameters and status of the current module into Flash.
Example:
“AT+saveConfigure\r\n” (Save the current configuration)
Response:
“AT+ok\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- clearConfigure

Purpose: Clear the saved configuration
Number of Parameters: None
Note: After successfully sending this command to the module, the module will immediately perform a soft reset.
Example:
“AT+ clearConfigure\r\n” (Clear the saved configuration)
Response:
“AT+ok\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getVersion

Purpose: Get the firmware version of the module
Number of Parameters: None

Example:
"AT+getVersion\r\n" (Get the firmware version of the module)
Response:
"AT+ok v0.1.0\r\n" (The current firmware version is v0.1.0)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getRemoteAddr

Purpose: Get the Bluetooth address of the other party's device currently connected
Number of Parameters: None
Example:
"AT+getRemoteAddr\r\n" (Get the Bluetooth address of the other party's device currently connected)
Response:
"AT+ok 62:FC:2E:AA:EB:80\r\n"
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setConnIntEx

Purpose: Offline configure the connection parameters
Number of Parameters: 4
Parameter Value:

Parameters	Value	Meaning
Parameter 1 (Minimum connection interval)	Min	Set the minimum connection interval to be Min*1.25ms
Parameter 2 (Maximum connection interval)	Max	Set the maximum connection interval to be Max*1.25ms
Parameter 3 (Latency)	Lat	Set Latency to be lat
Parameter 4 (Connection timeout)	Timeout	Set the connection timeout to be timeout*10ms

Notes:
(1) This command is used to configure connection parameters offline (that is, it does not need to be in a connected status). When the connection is established, the module will try to update the connection interval with the parameters set by this command. However, the update may not be successful, because not all connection parameters of the main device will be accepted.
(2) Minimum connection interval and maximum connection interval: The value range is: 0x0006~0x0C80, and the maximum connection interval shall not be less than the minimum connection interval;
(3) The greater the connection interval, the lower the power consumption and the lower the transmission rate.
(4) The value range of Latency is: 0x0000~0x01F3.
(5) The value range of connection timeout is: 0x000A~0x0C80.
The above parameter values must satisfy:
[Max*1.25*(1+Latency)]<10*Timeout.
Example:

"AT+ setConnIntEx 120 150 0 400\r\n" (The minimum connection interval of the module is set to be 150ms, the maximum connection interval is set to be 187.5ms, Latency is 0, and the connection timeout is 4s)
Response:
"AT+ok\r\n" (operate successfully)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getConnIntEx

Purpose: Get the currently connected parameter configuration
Note: This command is used to configure connection parameters offline (that is, it does not need to be in a connected status).
Number of Parameters: None
Example:
"AT+getConnIntEx\r\n" (Get the currently connected parameter configuration)
Response:
"AT+ok minInterval maxInterval latency timeout\r\n"
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setAuthMode

Purpose: Set the WeChat authentication method
Number of Parameters: 1

Parameter Value	Meaning
0	Encryption authentication with MD5 code
1	Non-encryption authentication with MD5 code
2	Non-encryption authentication with MAC address

Example:
"AT+setAuthMode 1\r\n" (Set the authentication mode to be non-encryption authentication with MD5 code)
Response:
"AT+ok\r\n"
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getAuthMode

Purpose: Get the WeChat authentication method, and it is non-encryption authentication by default
Number of Parameters: None
Example:
"AT+getAuthMode\r\n" (Get the WeChat authentication method)
Response:
"AT+ok 0\r\n"
"AT+ok 1\r\n"
"AT+ok 2\r\n"
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setMD5

Purpose: Set the MD5 code of the module
Number of Parameters: 1

Parameters	Value
MD5	The length of the MD5 code is 16 bytes. It adopts the Hex format to express, e.g.: 00112233445566778899AABBCCDDEEFF

Example:
"AT+setMD5 00112233445566778899AABBCCDDEEFF\r\n"
Set the MD5 code of the module to be:
[0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88,0x99,0xAA,0xBB,0xCC,0xDD,0xEE,0xFF]
Response:
"AT+ok\r\n" (operate successfully)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getMD5

Purpose: Get the MD5 code
Number of Parameters: None
Example:
"AT+getMD5\r\n" (Get the MD5 code)
Response:
"AT+ok 00112233445566778899AABBCCDDEEFF\r\n"
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- setDevID

Purpose: Set the Device ID of the module
Number of Parameters: 1

Parameters	Value
Device ID	An arbitrary ASCII string, the maximum length of the string is 32 characters.

Notes:
The start bit of the Device ID is required to be a non-blank character, and only a blank character is allowed to appear in the middle.
Example:
"AT+setDevID gh_01234567\r\n" (set Device ID to be gh_01234567)
"AT+setDevID 234 567\r\n" (set Device ID to be 234 567)
Response:
"AT+ok\r\n" (operate successfully)
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- getDevID

Purpose: Get the Device ID of the module
Number of Parameters: None
Example:
"AT+getDevID\r\n" (Get the Device ID of the module)

Response:
“AT+ok gh_01234567\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- setAesKey

Purpose: Set the AES128 key of the module
Number of Parameters: 1

Parameters	Value
Key	The key length is 128 bits, i.e. 16 bytes. It adopts the Hex format to express, e.g.: 00112233445566778899AABBCCDDEEFF

Example:
“AT+setAesKey 00112233445566778899AABBCCDDEEFF\r\n”
Set the key to be:
[0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88,0x99,0xAA,0xBB,0xCC,0xDD,0xE E,0xFF]
Response:
“AT+ok\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getAesKey

Purpose: Get the AES128 key of the module
Example:
“AT+getAesKey\r\n” (Get the AES128 key of the module)
Response:
“AT+ok 00112233445566778899AABBCCDDEEFF\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- setDataDir

Purpose: Set the data direction
Number of Parameters: 1

Parameter Value (data direction)	Meaning
0 (default)	Background
2	WeChat HTML page

Note: The data sent by the module can only have one direction at a time, and the direction is set by this command.
Example:
“AT+setDataDir 2\r\n” (Set the data direction to be the WeChat HTML page)
Response:
“AT+ok\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getDataDir

Purpose: Get the data direction
Number of Parameters: None

Example:
“AT+ getDataDir\r\n” (Get the data direction)
Response:
“AT+ok 0\r\n” (Background)
“AT+ok 2\r\n” (WeChat HTML page)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- closePC

Purpose: Turn off the module pairing code function, the pairing code function is closed by default.
Number of Parameters: None
Example:
“AT+closePC\r\n” (Turn off the module pairing code function)
Response:
“AT+ok\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- setPC

Purpose: Set up and turn on the pairing code function. The PIN code can only be set to be a number, the number of bits must be 6 bits, and the PIN code range is 000000~999999.
Number of Parameters: 1
Example:
“AT+setPC 123456\r\n”(Turn on and set the pairing code to be 123456)
Response:
“AT+ok\r\n” (operate successfully)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getPC

Purpose: Get the paring code status
Number of Parameters: None
Example:
“AT+getPC\r\n” (Get the paring code status)
Response:
“AT+ok 123456\r\n” (The operation is successful, if the pairing code is not set and turned on, it would return “AT+ok NULL\r\n”)
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getCalInfo

Purpose: Get the module calibration information
Number of Parameters: None
Example:
“AT+getCalInfo\r\n” (Get the module calibration information)
Response:
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

“AT+ok Status\r\n” (Status indicates the module calibration status. 0 indicates that it does not perform/pass the calibration test, and 1 indicates that it has passed the calibration test)
Example:
“AT+ok 1\r\n” (It has passed the calibration test)

- setVth

Purpose: Set the low battery prompt threshold voltage of the module
Number of Parameters: 1
Note: The valid range of the parameter is 200~300, and the actually set threshold voltage is the parameter value/100. When the parameter is set beyond the valid range, the module reports an error.
Example:
“AT+setVth 220\r\n” (Set the low battery prompt threshold voltage of the module is 2.20V)
Response:
“AT+ok\r\n”
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- getVth

Purpose: Get the status of low battery prompt function and threshold voltage of the module. The low battery prompt function of the module is in the OFF status by default, and the threshold voltage is 2.20V.
Number of Parameters: None
Example:
“AT+getVth\r\n”
Response:
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)
“AT+ok Status Vth\r\n” (Status indicates the state of low battery prompt function of the module. 0 indicates the OFF state, 1 indicates the ON state; Vth indicates the threshold voltage, ranging from 200 to 300, and this value indicates the set threshold voltage to be Vth/100)
Example:
“AT+ok 0 220\r\n” (The low battery prompt function of the module is in the OFF status, and the threshold voltage is 2.20V)

- closeVth

Purpose: Turn off the battery prompt function of the module, and at the same time, turn off the waveform output function of the VTH_LED pin. The pin level is restored to the initial state. The low battery prompt function is in the OFF status by default.
Number of Parameters: None
Example:
“AT+closeVth\r\n”
Response:
“AT+ok\r\n”
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)

- openVth

Purpose: Turn on the low battery prompt function of the module, and at the same time, turn on the battery capacity to get UUID. The low battery prompt function is in the OFF status by default.
Number of Parameters: None
Note: After the low battery prompt function of the module is turned on, the module detects the supply voltage according to the threshold set by setVth. When the module detects low battery, the VTH_LED pin outputs the corresponding waveform according to Table 3.
Example:
"AT+openVth\r\n"
Response:
"AT+ok\r\n"
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)
Note: Turning on this function will increase the power consumption of the module. If you need to control power consumption, please turn off this function.

- closeLED

Purpose: Turn off the RF data sending/receiving prompt function of the module. At the same time, turn off the waveform output functions of TX_LED and RX_LED pins. The pin level is restored to the initial state. The module data sending/receiving prompt function is in the OFF status by default.
Number of Parameters: None
Example:
"AT+closeLED\r\n"
Response:
"AT+ok\r\n"
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)

- openLED

Purpose: Turn on the RF data sending/receiving prompt function of the module. The module data sending/receiving prompt function is in the OFF status by default.
Number of Parameters: None
Note: After the data sending/receiving prompt function of the module is turned on, TX-LED and RX_LED pins of the module output the corresponding waveform according to Table 3.
Example:
"AT+openLED\r\n"
Response:
"AT+ok\r\n"
"AT+err reason\r\n" (operation failed, see Section 7.6 for the specific value of reason)
Note: Turning on this function will increase the power consumption of the module. If you need to control power consumption, please turn off this function.

- getLED

Purpose: Get the status of RF data sending/receiving prompt function of the module. The module data sending/receiving prompt function is in the OFF status by default.

Number of Parameters: None
Example:
“AT+getLED\r\n”
Response:
“AT+err reason\r\n” (operation failed, see Section 7.6 for the specific value of reason)
“AT+ok Status\r\n” (Status indicates the state of RF data sending/receiving prompt function. 0 indicates the OFF status, and 1 indicates the ON status).
Example:
“AT+ok 0\r\n” (The RF data sending/receiving prompt function of the module is in the OFF status)

7.5 Response Command Sent by Module to External MCU

Command Code	Function
ok	Operate successfully
err	Operation error
conn	Connection response
disconn	The connection response disconnected.

7.6 Operation Error Code

7.6.1 Introduction to Operation Error Code

Reason	**Meaning**
invalidParam	The command parameter is illegal.
invalidCmd	The command is illegal.
notConn	The module is not connected.
procFailure	Operation failed

7.6.2 Introduction to Part of Operation Error Codes

Start Mark of Command	Command Code	Parameter List	End Mark of Command
	invalidCmd judgment area: (1) With parameter command From the position after the start mark of the command code to the last character (inclusive) of the command code; (2) Without parameter command Between the position after the start mark of the command code and the end mark of the command code. Situation of returning invalidCmd: 1. The character irrelevant to the command code appears in the characters of the command code; 2. The character irrelevant to the command code appears between the start mark of the command and the	invalidParam judgment area: After the ending of the command code and before the end mark of the command Situation of returning invalidParam: 1. The parameter is not within the range or the parameter does not satisfy the restrictions; 2. The character irrelevant to the parameter appears in the parameter; 3. The character irrelevant to the parameter appears between the parameter list and the command code; 4. The character irrelevant to the parameter appears between the parameter list and the end mark of the command;	

	command code;	5. Lack of relevant characters between the parameter list and the command code; 6. Lack of relevant characters in the parameter list.	

(1) If there are errors in the command code and parameter list, only AT+err invalidCmd will be reported;
(2) In the event that the AT command code received by the module is correct, it will report "AT+err proFailure" when the operation cannot be executed normally according to the AT command due to reasons of the module.

8. Typical Application Circuit

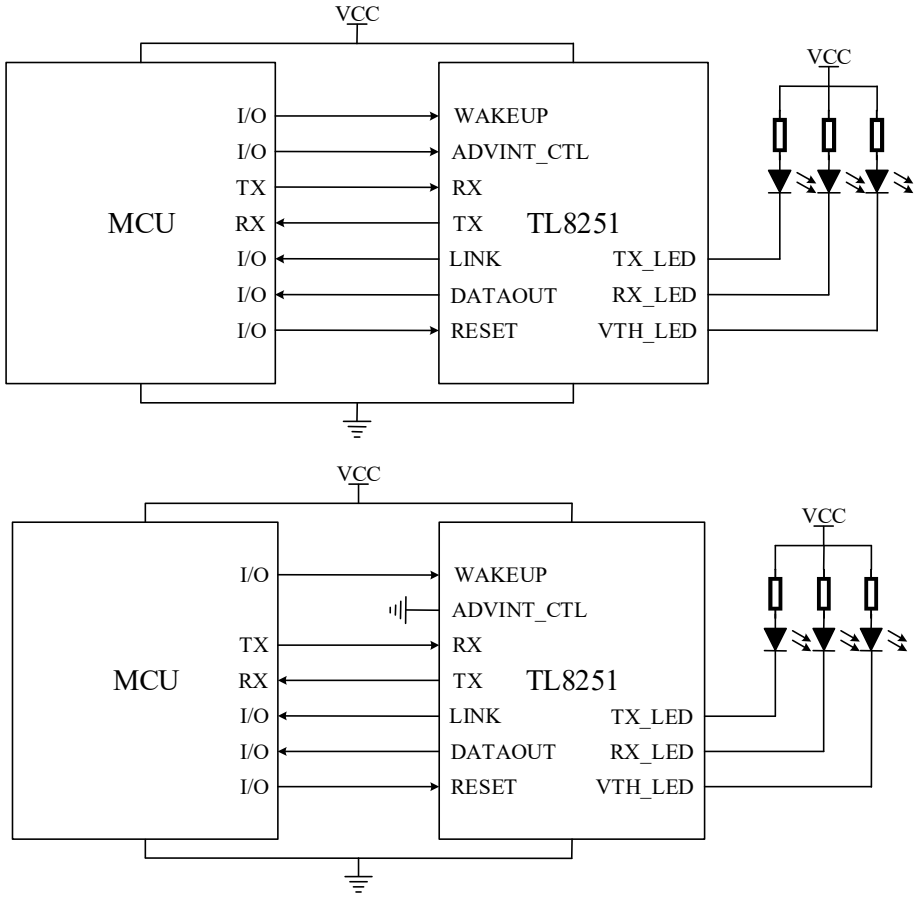


Figure 3 Block Diagram For Typical Application Circuit of Module

9. Performance Indicators

9.1 Power Consumption

Note: The following power consumption data is for reference only, and the data will fluctuate in different test conditions.

9.1.1 Power Consumption with Long Broadcast Interval

The power consumption with long broadcast interval refers to the power consumption measured when the ADVINT_CTL pin is floating.

Table 4 Power Consumption of Module in Sleep Status

Status	Average Current
Broadcast	36μA
Connected	182μA

Table 5 Power Consumption of Module in Wake-up Status

Status	Average Current
Broadcast	2.343 mA
Connected	2.925 mA

9.1.2 Power Consumption with Short Broadcast Interval

The power consumption with short broadcast interval refers to the power consumption measured when the ADVINT_CTL pin is pulled down.

Table 6 Power Consumption of Module in Sleep Status

Status	Interval (ms)	Average Current
Broadcast	50	417μA
	200	83μA
	500	44μA
	1000	27μA
Connected	50	336μA
	150	252μA
	500	210μA
	1000	157μA

Table 7 Power Consumption of Module in Wake-up Status

Status	Interval (ms)	Average Current
Broadcast	50	2.494mA
	200	2.357mA
	500	2.351mA
	1000	2.344mA
Connected	50	2.983mA
	150	2.922mA
	500	2.913mA
	1000	2.903mA

Notes:

1、See the appendix for the corresponding waveforms of power consumption data;

2、The above current test instrument is Agilent 34460A, with the sampling speed of 400μs/time;
3、Single Bluetooth module shall be used in the test and the broadcast data is blank. The pairing code function, the low battery alarm function and the RF data sending/receiving prompt function are not turned on;
4、The ambient temperature is +25℃, and the supply voltage is 3.0V;
5、The measured average current is the average value of the current lasting for 20s when the module is stable.

9.2 Electrical Specifications

Table 8 Rated Value of Maximum Electrical Parameters

Parameters	Min	Max	Unit
VCC	-0.3	3.6	V
Voltage on Input Pin	-0.3	VCC+0.3	V
Output Voltage	0	VCC	V
Storage Temperature	-65	150	℃

Table 9 Recommended Operating Conditions

Parameters	Min	Typ	Max	Unit
VCC	1.8	3.3	3.6	V
Voltage on Input Pin	-0.3		VCC+0.3	
Operating Temperature Range	-40		85	℃

Table 10 AC/DC Characteristics

Parameters	Min	Typ	Max	Unit
Input high voltage	0.7VCC		VCC	V
Input low voltage	GND		0.3VCC	V
Output high voltage	0.9VCC		VCC	V
Output low voltage	GND		0.1VCC	V
Vth error	Vth－0.1	Vth	Vth+0.1	V

10. FCC Warnings

FCC Statement
This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and
(2) this device must accept any interference received, including interference that may cause undesired operation.
Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.
The modular can be installed or integrated in mobile or fix devices only. This modular cannot be installed in any portable device.

FCC Radiation Exposure Statement

This modular complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

If the FCC identification number is not visible when the module is installed inside another device, then the outside of the device into which the module is installed must also display a label referring to the enclosed module. This exterior label can use wording such as the following:
“Contains Transmitter Module FCC ID: 2AZJTTL8251D1 Or Contains
FCC ID:2AZJTTL8251D1”

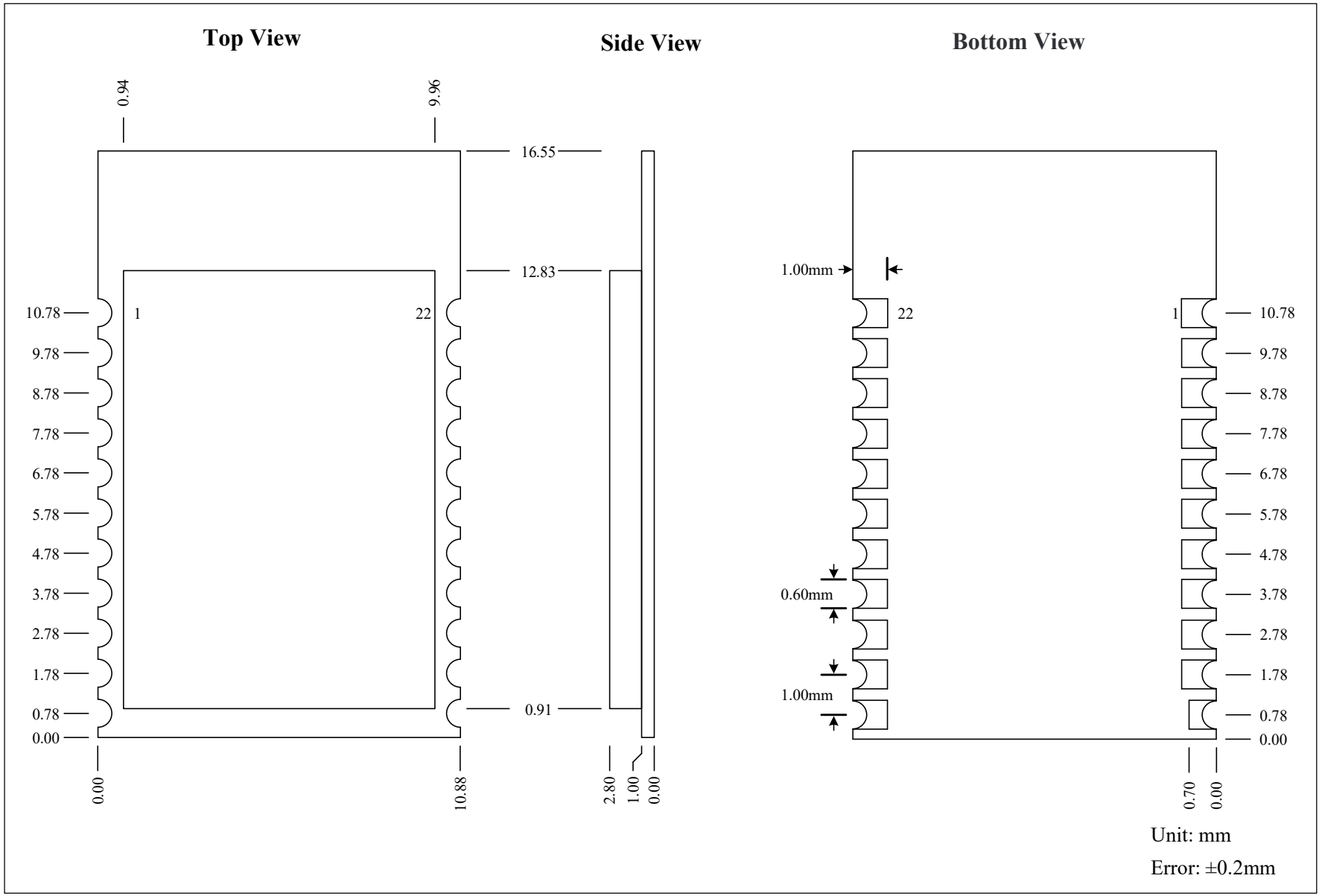
The devices must be installed and used in strict accordance with the manufacturer's instructions as described in the user documentation that comes with the product.
Any company of the host device which install this modular with Single modular approval should perform the test of radiated emission and spurious emission according to FCC part 15C : 15.247 and 15.209 requirement, Only if the test result comply with FCC part 15C : 15.247 and 15.209 requirement，then the host can be sold legally.

Antenna information

Antenna Type	Antenna Gain
PCB Antenna	2.5dBi

Trace antenna designs: Not applicable.

11. Module Size



12. Identification Information

---------- Brand name

---------- Model name

---------- FCC ID

---------- Date code

13. Appendix

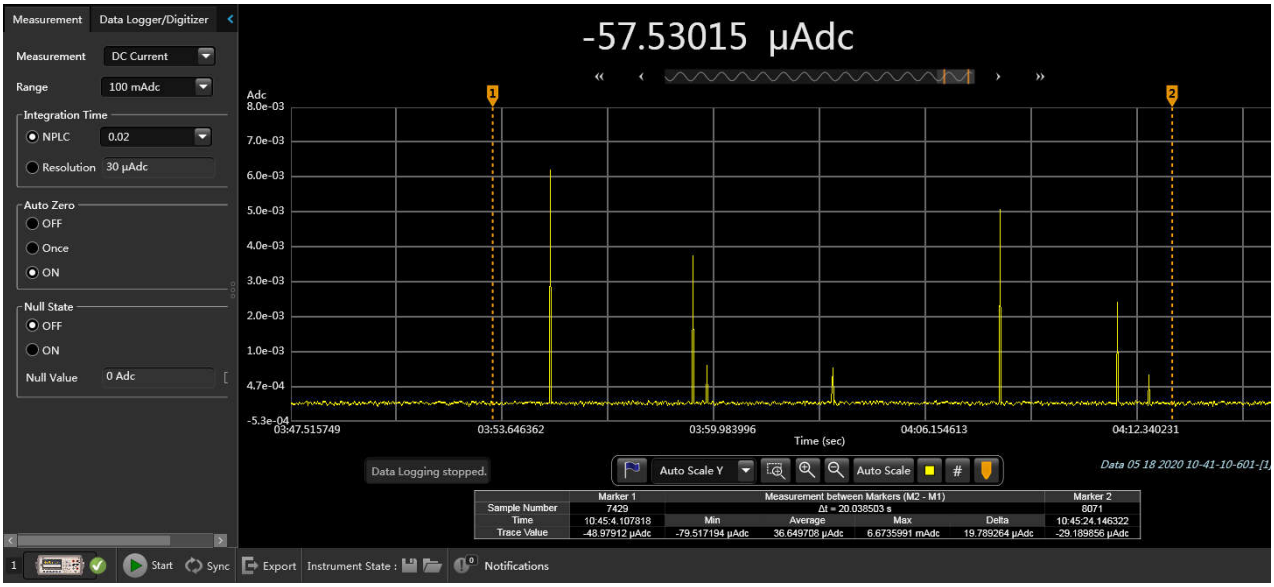


Figure 4 Broadcast Current with Long Broadcast Interval in Sleep Status

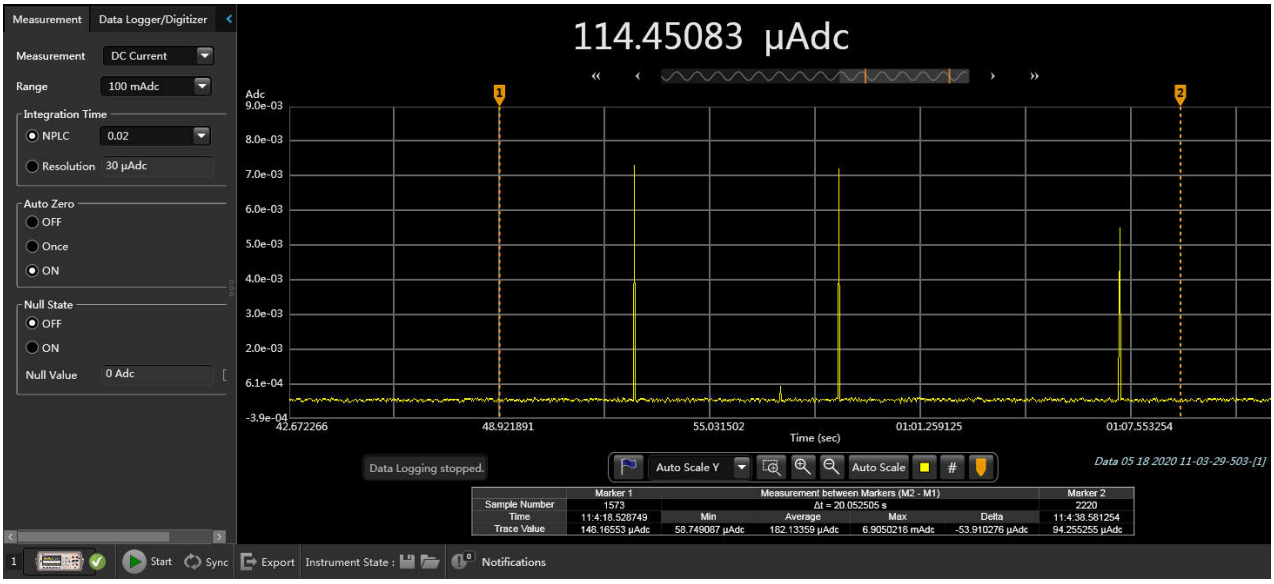


Figure 5 Connection Current with Long Broadcast Interval in Sleep Status

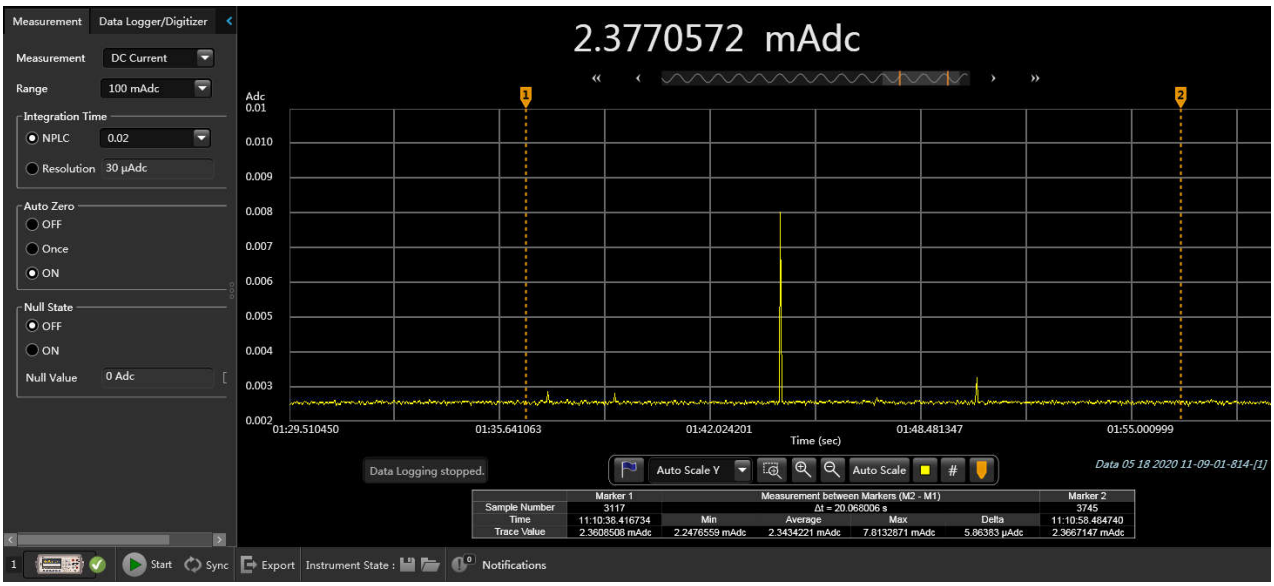


Figure 6 Broadcast Current with Long Broadcast Interval in Wake-up Status

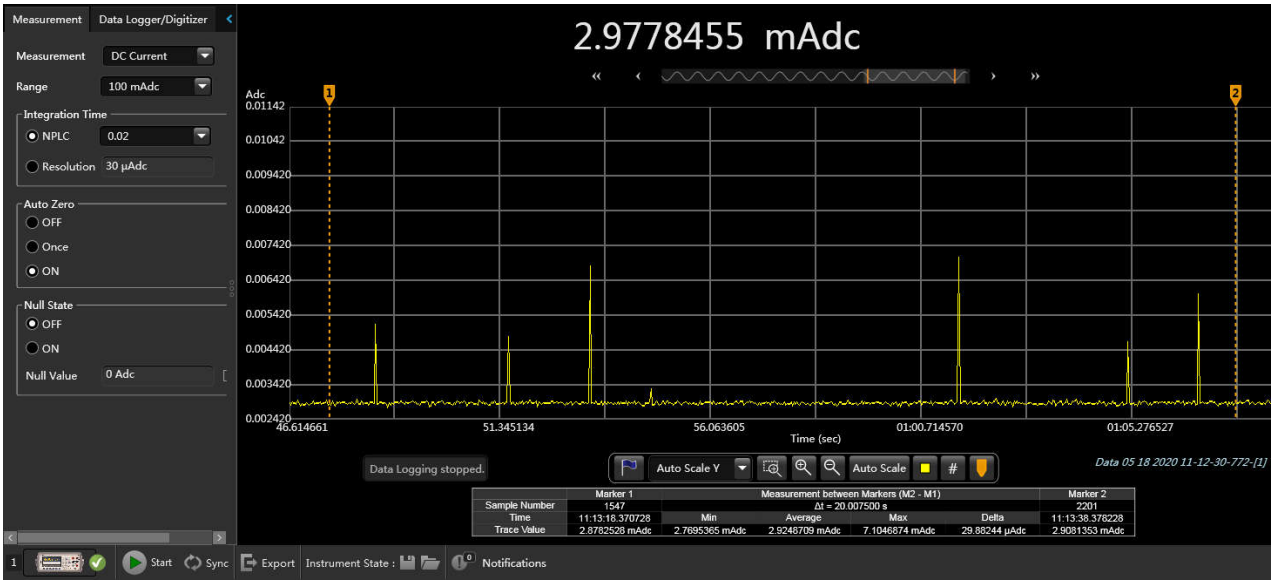


Figure 7 Connection Current with Long Broadcast Interval in Wake-up Status

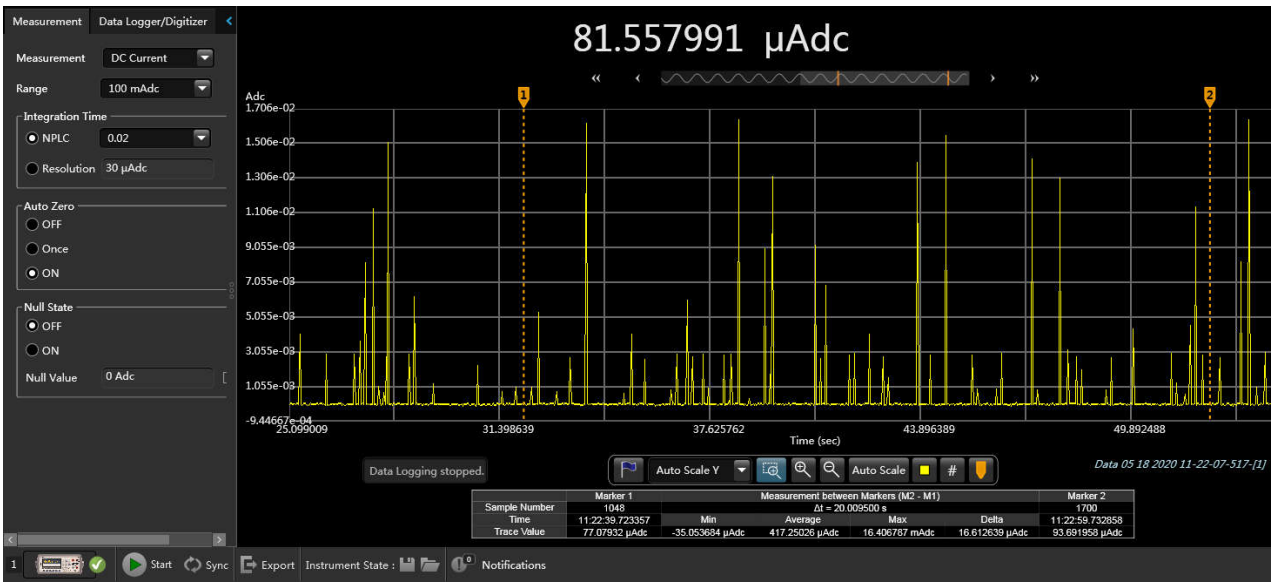


Figure 8 Current with Short Broadcast Interval in Sleep Status and with Broadcast Interval of 50ms

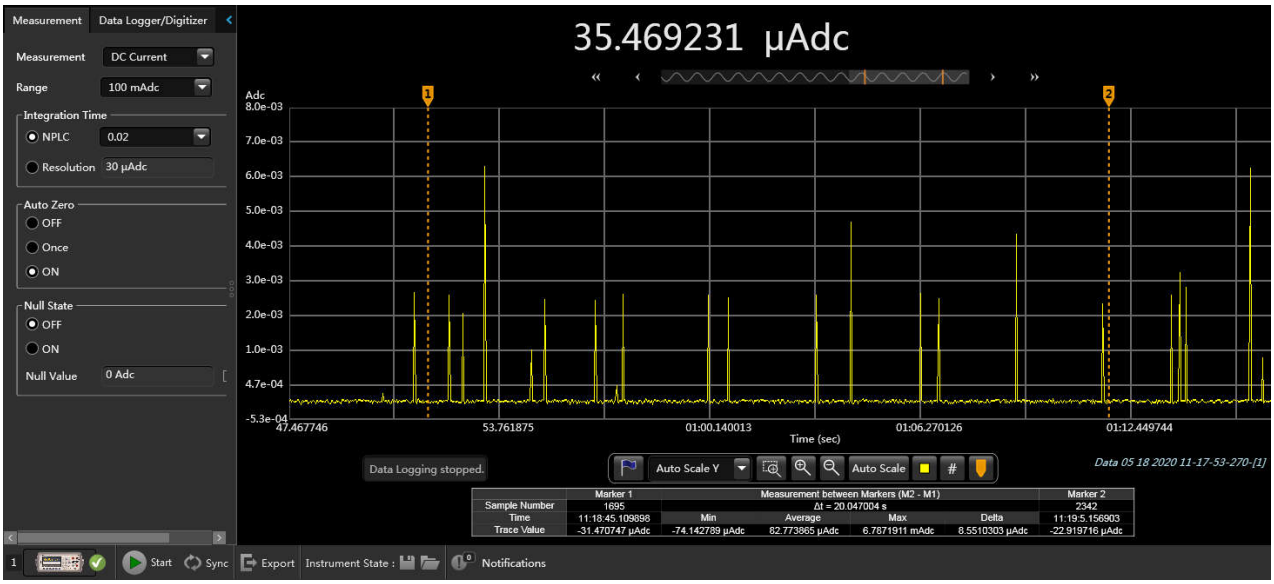


Figure 9 Current with Short Broadcast Interval in Sleep Status and with Broadcast Interval of 200ms

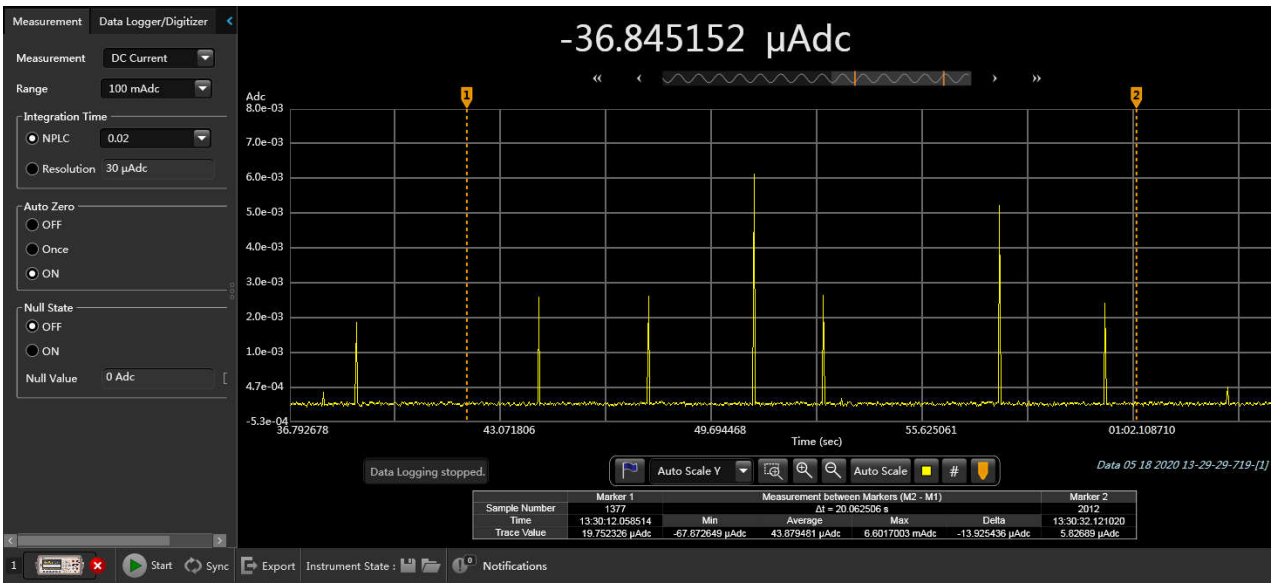


Figure 10 Current with Short Broadcast Interval in Sleep Status and with Broadcast Interval of 500ms

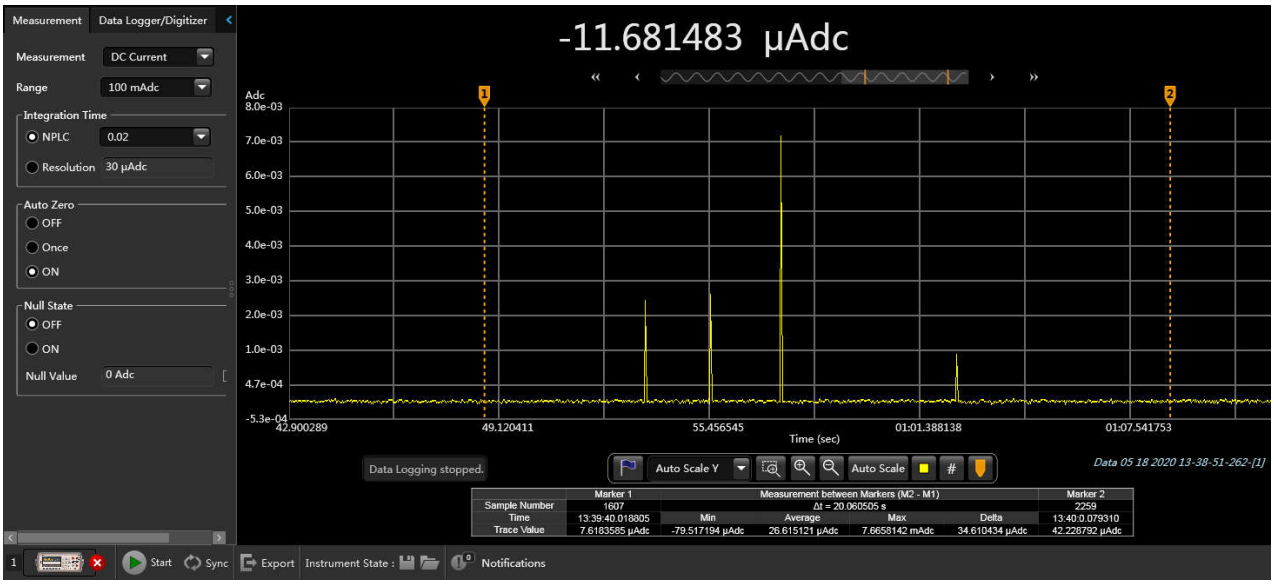


Figure 11 Current with Short Broadcast Interval in Sleep Status and with Broadcast Interval of 1000ms

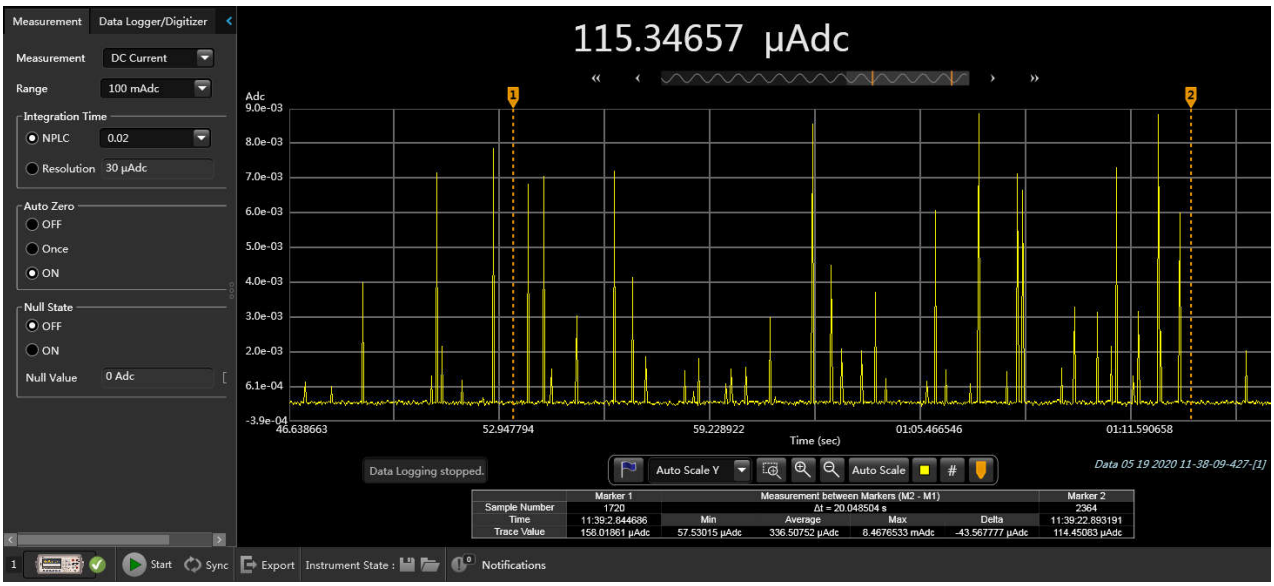


Figure 12 Current with Short Broadcast Interval in Sleep Status and with Connection Interval of 50ms

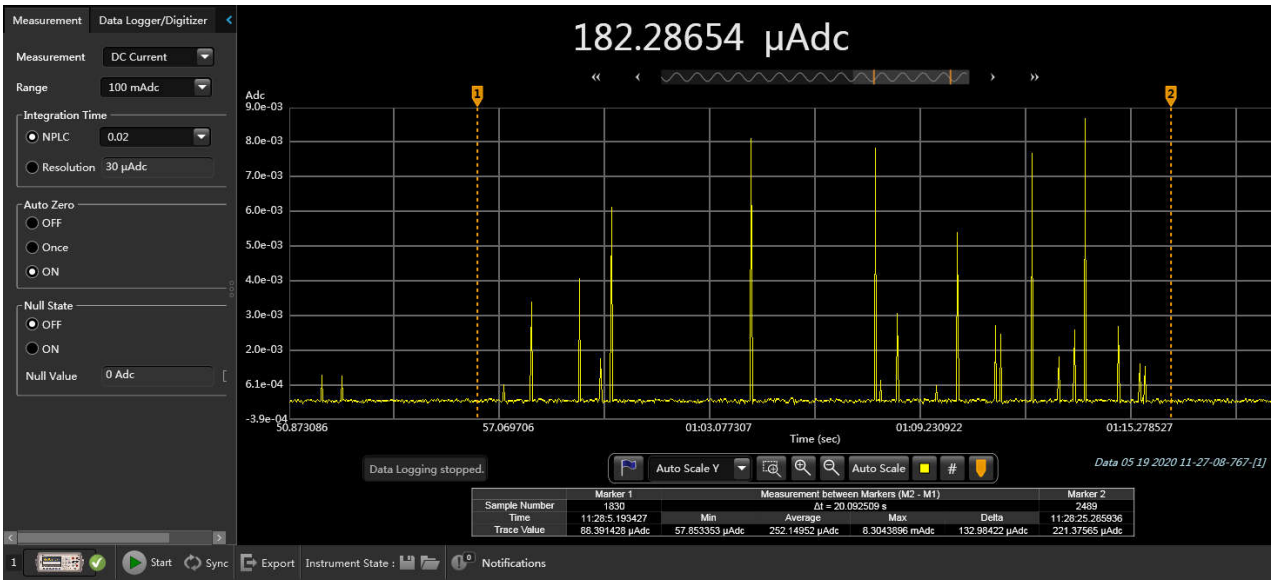


Figure 13 Current with Short Broadcast Interval in Sleep Status and with Connection Interval of 150ms

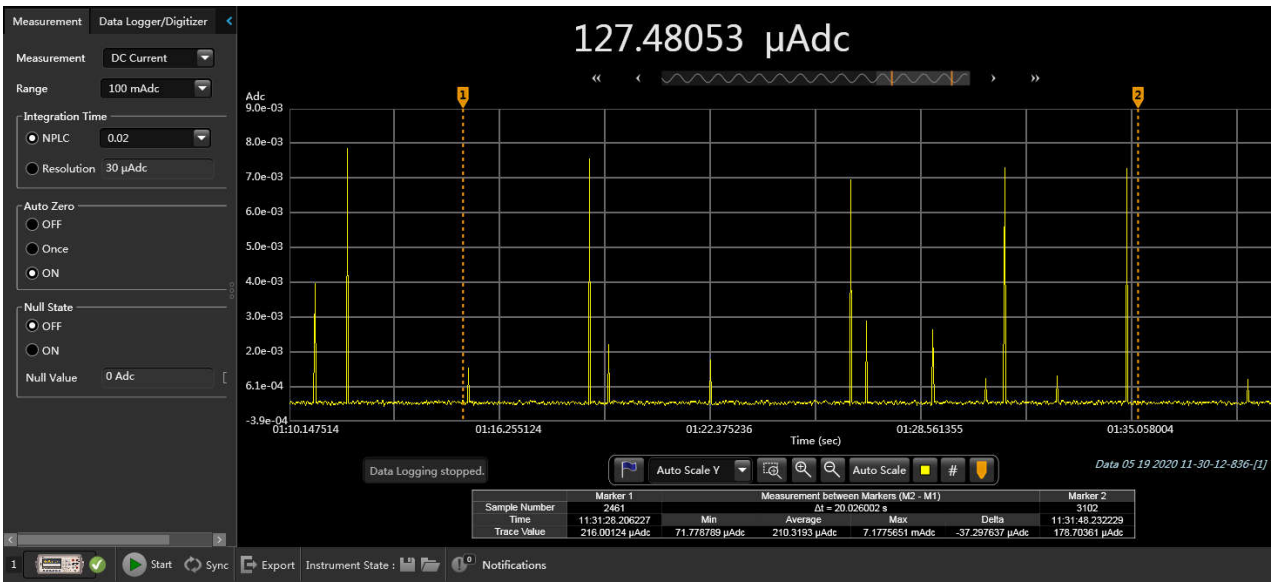


Figure 14 Current with Short Broadcast Interval in Sleep Status and with Connection Interval of 500ms

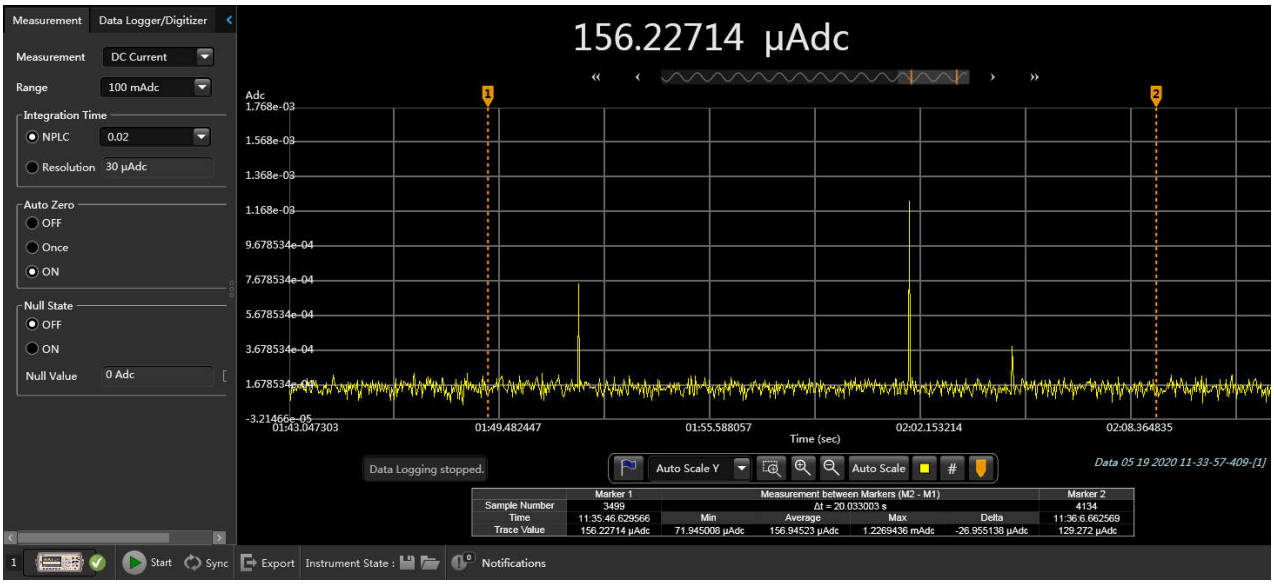


Figure 15 Current with Short Broadcast Interval in Sleep Status and with Connection Interval of 1000ms

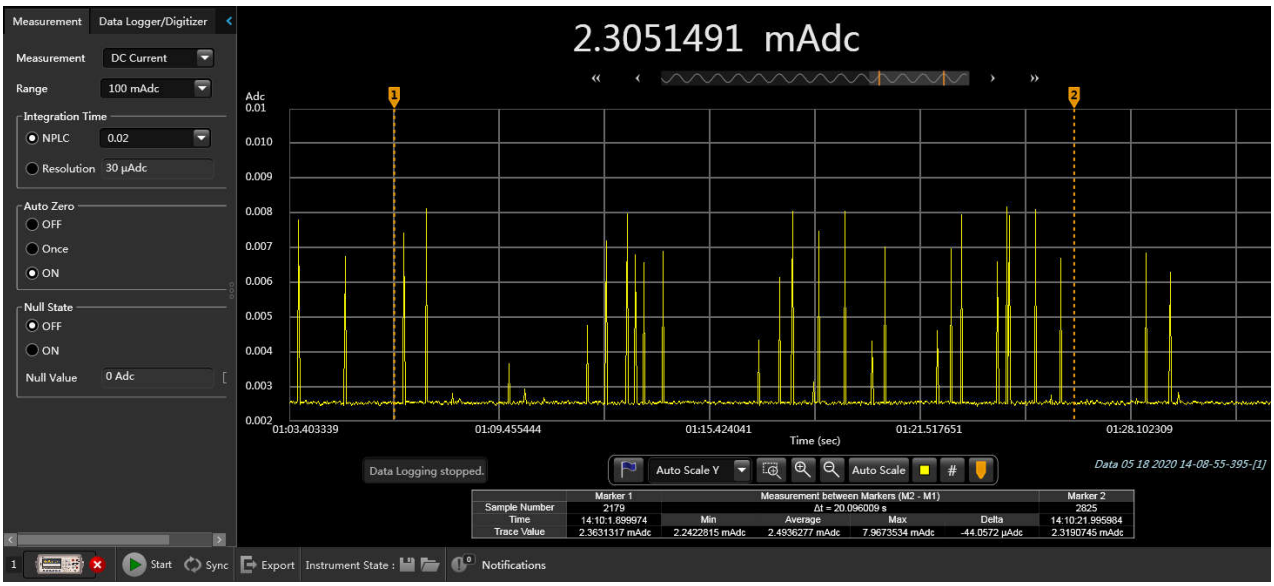


Figure 16 Current with Short Broadcast Interval in Wake-up Status and with Broadcast Interval of 50ms

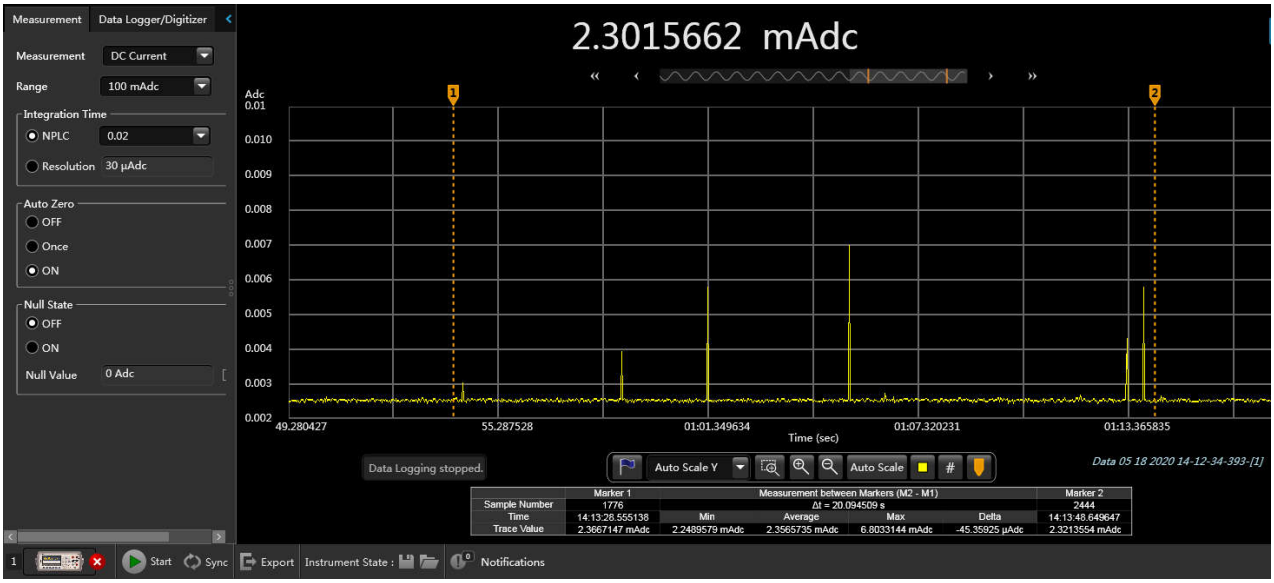


Figure 17 Current with Short Broadcast Interval in Wake-up Status and with Broadcast Interval of 200ms

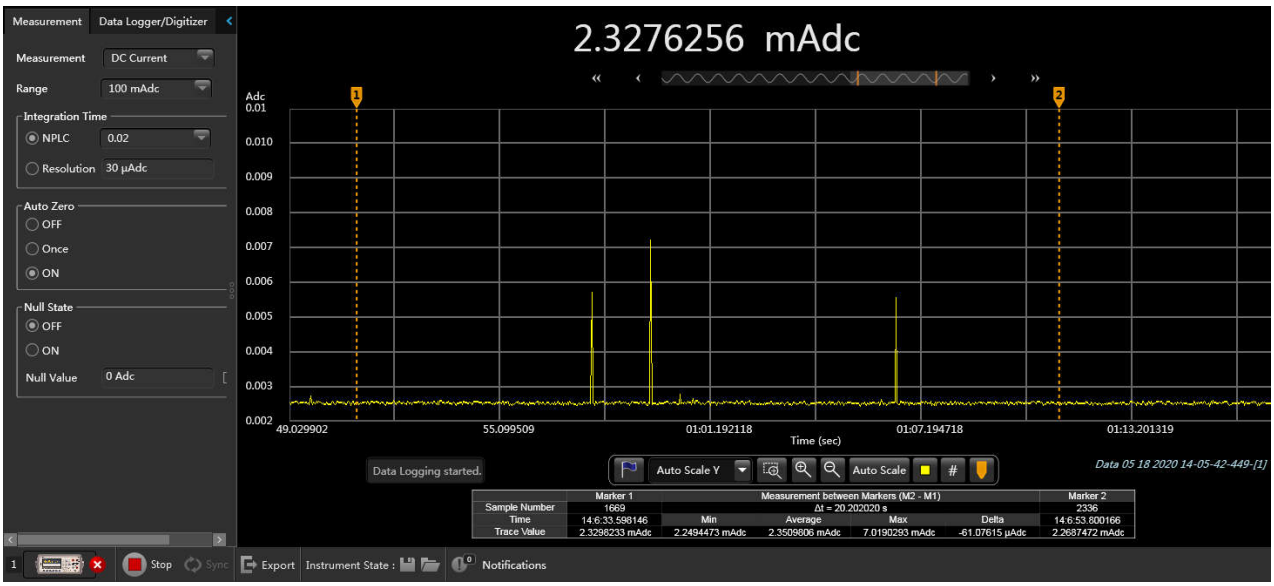


Figure 18 Current with Short Broadcast Interval in Wake-up Status and with Broadcast Interval of 500ms

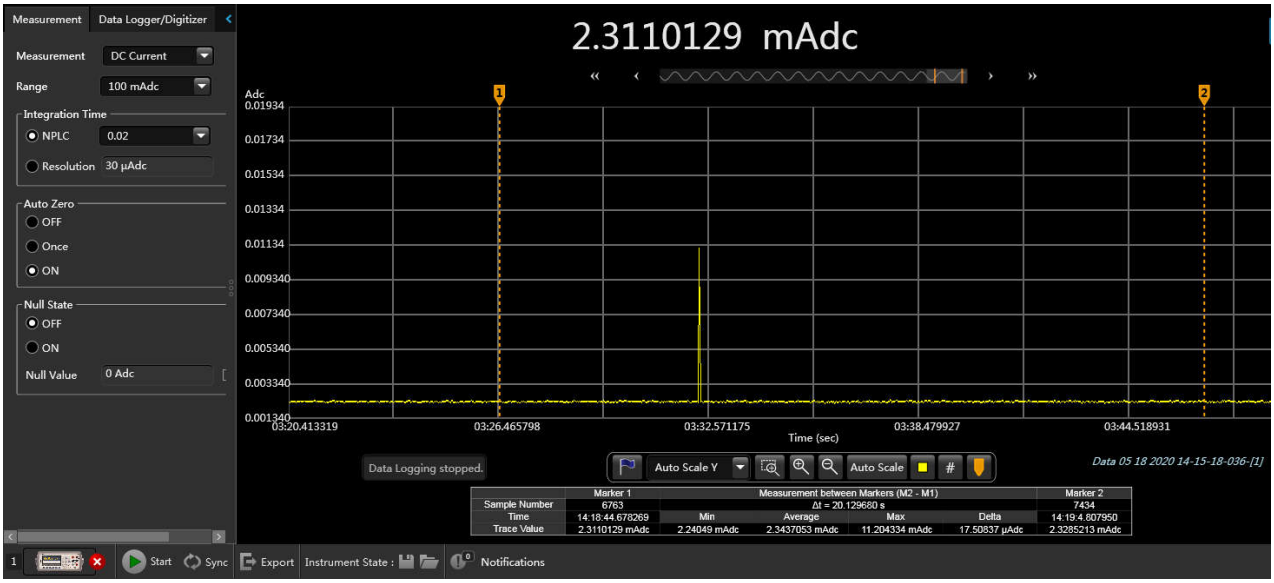


Figure 19 Current with Short Broadcast Interval in Wake-up Status and with Broadcast Interval of 1000ms

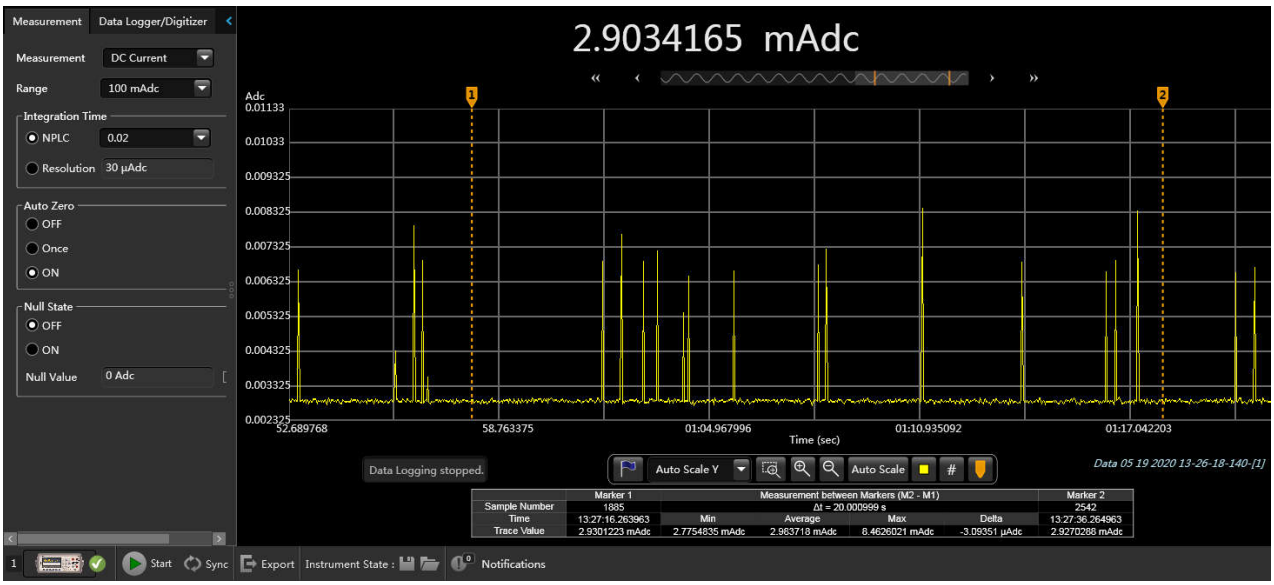


Figure 20 Current with Short Broadcast Interval in Wake-up Status and with Connection Interval of 50ms

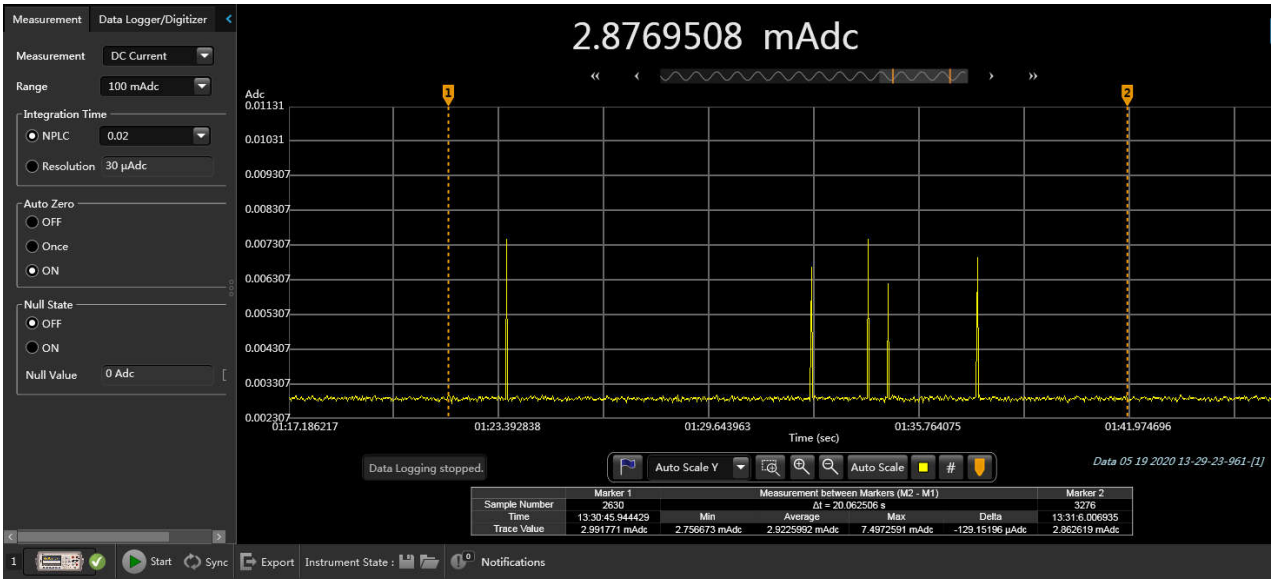


Figure 21 Current with Short Broadcast Interval in Wake-up Status and with Connection Interval of 150ms

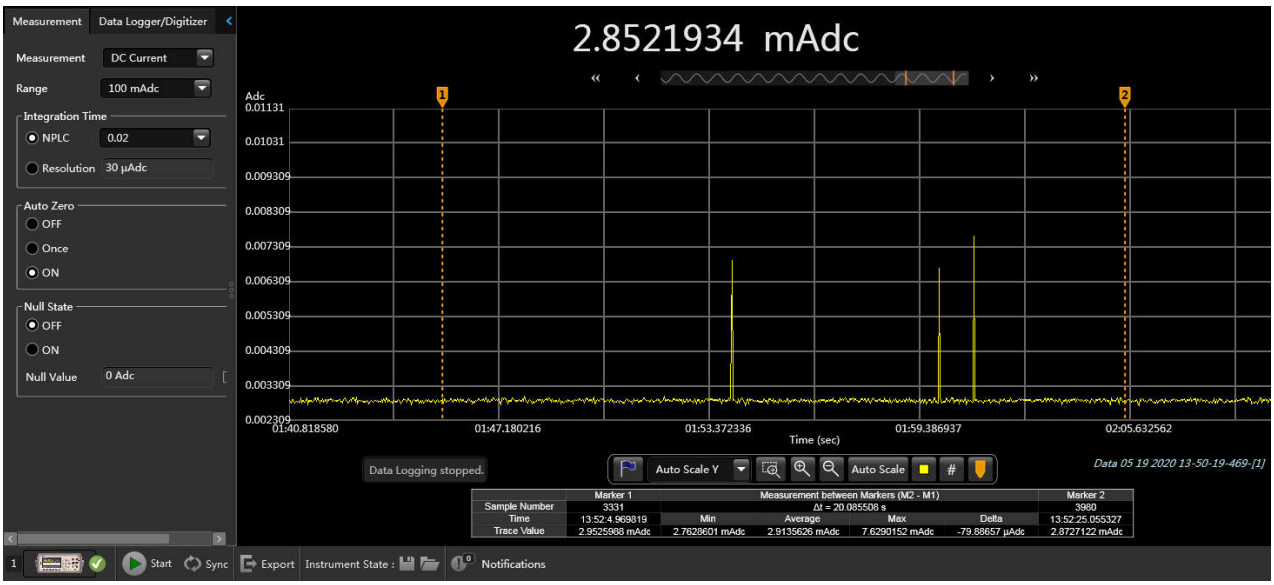


Figure 22 Current with Short Broadcast Interval in Wake-up Status and with Connection Interval of 500ms

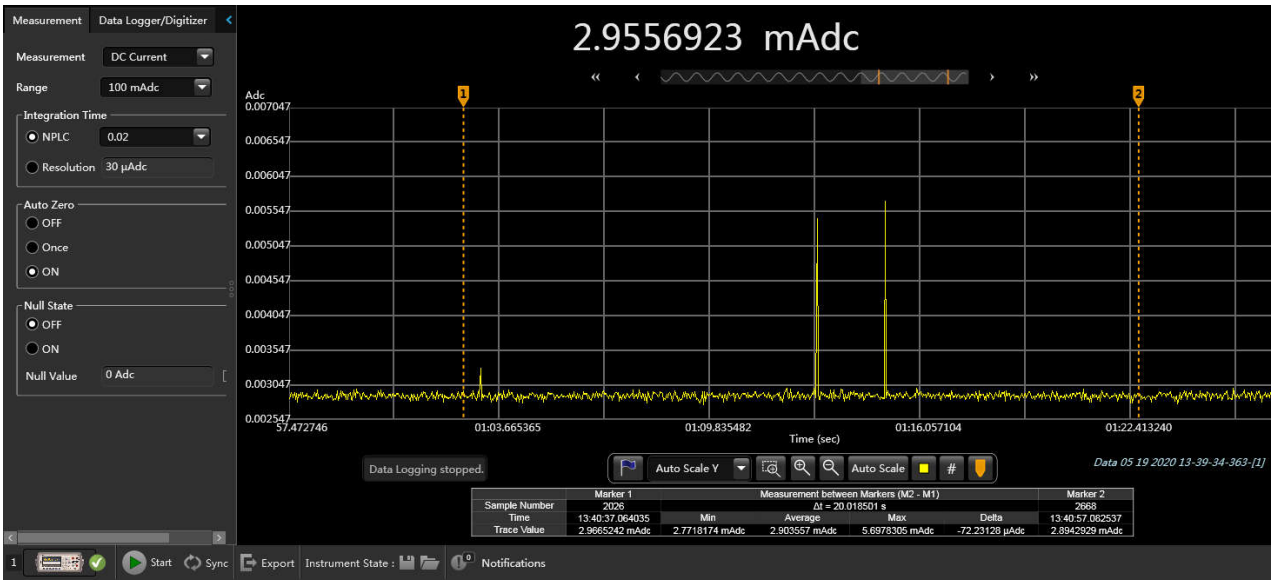


Figure 23 Current with Short Broadcast Interval in Wake-up Status and with Connection Interval of 1000ms