

MuYu

MY-BT102 Commands Guide

Version 1.2

Contact Us

Shenzhen MuYu Technology Co., Ltd

Email: info@muyusmart.com

Zipcode: 518100

Web: www.muyumodule.com

Revision History

Version	Date	Notes	Author
1.0	2024/01/01	First Release	Richard

Shenzhen Muyu Technology Co., Ltd

Content

1. INTRODUCTION	5
1.1 Overview	5
1.2 Default Setting:.....	5
2. COMMAND	5
2.1 AT command format	5
2.2 General Command.....	6
2.2.1 UART Communication Test.....	6
2.2.2 AT+CMDLIST	6
2.2.3 AT+VER	7
2.2.4 AT+ADDR	8
2.2.5 AT+LEADDR	8
2.2.6 AT+NAME.....	8
2.2.7 AT+LENAME	9
2.2.8 AT+BAUD	10
2.2.9 AT+FLOWCTL	11
2.2.10 AT+TPMODE.....	12
2.2.11 AT+COD.....	12
2.2.12 AT+SSP.....	13
2.2.13 AT+LPM.....	13
2.2.14 AT+REBOOT.....	14
2.2.15 AT+RESTORE.....	15
2.2.16 AT+TXPOWER	15
2.2.17 AT+PIN.....	16
2.2.18 AT+PLIST	16
2.2.19 AT+BTEN	17
2.2.20 AT+CLOSEAC.....	17
2.2.21 AT+GPIOCFG.....	18
2.2.22 AT+DISC	19
2.3 SPP Command.....	19
2.3.1 AT+SPPSTAT.....	19
2.3.2 AT+SPPDISC.....	20
2.3.3 AT+SPPSEND.....	20
2.4 GATT Command.....	21
2.4.1 AT+GATTSTAT.....	21
2.4.2 AT+IBEAON.....	21
2.4.3 AT+ADVDATA.....	22
2.4.4 AT+ADVADDR	23
2.4.5 AT+GATTSEND	23
3. DETAILED EXPLANATION OF AIR COMMANDS.....	24

4. NOTE:24

Shenzhen Muyu Technology Co., Ltd

1. Introduction

This document provides a simple performance introduction and detailed explanation of AT commands of the MY-BT102 module, aiming to guide users to quickly become familiar with the development of the MY-BT102 module.

1.1 Overview

MY-BT102 module supports classic Bluetooth and 2.4GHz BLE (Bluetooth Low Energy) 5.2 protocols. The SPP transparent transmission rate can reach 80~90kBytes/s, and the BLE transparent transmission rate can reach 40~50kBytes/s. It supports soft flow control and hard flow control.

1.2 Default Setting:

Bluetooth Name	MY-BT102
BLE Name	MY-BT102-LE
Transmitter	0DBm
Serial Port Baud rate	115200bps/8/N/1
Service UUID	FFF0
Notify UUID	FFF1
Write or write without response UUID	FFF2

2. Command

2.1 AT command format

1. All command start with "AT", end with <CR><LF>
2. <CR> stand for "carriage return", corresponding hex is 0x0D
3. <LF> stands for "line feed", corresponding hex is 0x0A
4. If command has parameter, parameter keep behind "="
5. If command has multiple parameter, parameter must be separated by ","
6. Module will always report command's execution result using "OK" for success or "ERROR" for failure

7. In this document, << represents command input, >> represents command reply, and the content inside { } is optional.

AT+ Command {=Param1{, Param2{, Param3...}} <CR><LF>

For example:

1. Read the version number

<< AT+VER

>> +VER=1.0.0,MY-BT102

>> OK

2. Change the illegal baud rate

<< AT+BAUD=1234

>>ERROR

2.2 General Command

2.2.1 UART Communication Test

Description: UART communication testing between HOST and Module
Format: AT
Response: OK
Eg: << AT >> OK

2.2.2 AT+CMDLIST

Description: AT command list, print and display all AT commands available in the current version
Format: AT+CMDLIST
Response: +CMDLIST=params

Eg:

```
<< AT+CMDLIST
```

```
>> +CMDLIST=
```

```
{
```

```
AT+NAME
```

```
AT+VER
```

```
...
```

```
}
```

```
>>
```

```
OK
```

2.2.3 AT+VER

Description: Read Firmware version

Format: AT+VER

Response: +VER=param

Eg:

```
<< AT+VER
```

```
>> +VER=1.0.0,MY-BT102
```

```
>> OK
```

2.2.4 AT+ADDR

Description: Read Bluetooth module MAC address
Format: AT+ADDR
Response: +ADDR=param
Eg: << <i>AT+ADDR</i> >> <i>+ADDR= AB0B01020304</i> >> <i>OK</i>

2.2.5 AT+LEADDR

Description: Read BLE MAC address
Format: AT+LEADDR
Response: +LEADDR=param
Eg: << <i>AT+LEADDR</i> >> <i>+LEADDR= AB0B01020304</i> >> <i>OK</i>

2.2.6 AT+NAME

Description: Read, Setting Bluetooth Name Default name: MY-BT102
Format: AT+NAME {=param1{, param2}}

param1: Device Name (Length1~25 Bytes ASCII)

param2:(0~2)

(0) Disable MAC address suffix

(1) Device name + last 4 digits of MAC address

(2) Device name + last 6 digits of MAC address

Response: +LENAME=param

Eg1. Read Bluetooth Name

<< *AT+NAME*

>> *+NAME=MY-BT102*

>> *OK*

Eg2: Setting Bluetooth Name: AABBCCDDEEFF (Changes take effect immediately)

<< *AT+NAME=AABBCCDDEEFF*

>> *OK*

Eg3: Device name + last 4 digits of MAC address (Changes take effect immediately)

<< *AT+NAME=AABBCCDDEEFF,1*

>> *OK*

Eg4: Device name + last 6 digits of MAC address (Changes take effect immediately)

<< *AT+NAME=AABBCCDDEEFF,2*

>> *OK*

2.2.7 AT+LENAME

Description: Read, Setting Bluetooth Name Default name: **MY-BT102-LE**

Format: AT+LENAME {=param1{, param2}}

param1: Device Name (Length1~25 Bytes ASCII)
 param2:(0~2)
 (3) Disable MAC address suffix
 (4) Device name + last 4 digits of MAC address
 (5) Device name + last 6 digits of MAC address

Response: +LENAME=param

Eg1. Read Bluetooth Name

```
<< AT+LENAME
>> +NAME= MY-BT102-LE
>> OK
```

Eg2: Setting Bluetooth Name: AABBCCDDEEFF (Changes take effect immediately)

```
<< AT+LENAME=AABBCCDDEEFF
>> OK
```

Eg3: Device name + last 4 digits of MAC address (Changes take effect immediately)

```
<< AT+LENAME=AABBCCDDEEFF,1
>> OK
```

Eg4: Device name + last 6 digits of MAC address (Changes take effect immediately)

```
<< AT+LENAME=AABBCCDDEEFF,2
>> OK
```

2.2.8 AT+BAUD

Description: Read, Setting Bluetooth UART Baud rate, default: 115200

Format: AT+BAUD {=param}

<p>param: Baud rate, support: 1200/2400/4800/9600/14400/19200/38400/57600/115200/230400/460800/921600/1000000</p>
<p>Response: +BAUD=param</p>
<p>Eg1: Read Baud rate << AT+BAUD >> +BAUD=115200 >> OK</p>
<p>Eg2: Setting Baud rate: 921600 (Changes take effect immediately) << AT+BAUD=921600 >> OK</p>

2.2.9 AT+FLOWCTL

<p>Description: Read/Setting flow control, default: 0</p>
<p>Response: AT+FLOWCTL{=param}</p>
<p>Param (0~1) 0: Turn off flow control 1: Turn on flow control (CTS,RTS cannot be left floating)</p>
<p>Response: +FLOWCTL=param</p>
<p>Eg1: Read Flow control << AT+FLOWCTL >> +FLOWCTL=0 >> OK</p>

2.2.10 AT+TPMODE

Description: Read / Setting Command and Throughput mode; Default: 1 (GATT Throughput Mode)
Format: AT+TPMODE {=param}
<p>param:(0~1)</p> <p>(0) It is command mode in any state, including Bluetooth connection and Bluetooth disconnection. In command mode, any data received by UART will be analyzed for commands. In connection state, data can only be sent to the remote device through the AT+GATTSEND command.</p> <p>(1) GATT transparent transmission. When Bluetooth is disconnected, it is in command mode. You can use AT commands to change relevant parameters. When Bluetooth is connected, it is in GATT transparent transmission mode. AT commands are disabled. Any data received by UART will be sent to remote device</p>
Response: +TPMODE=param
<p>Eg1: Read Mode</p> <pre><< AT+TPMODE >> +TPMODE=1 >> OK</pre>
<p>Eg2: Setting Command Mode (Changes take effect immediately)</p> <pre><< AT+TPMODE=0 >> OK</pre>

2.2.11 AT+COD

Description: Read / Setting device type; default: 240404
Format: AT+COD{=param}
param(6 bytes ASCII)

Response: +COD=param

Eg: Read device type

<< *AT+COD*

>> *+COD=240404*

>> *OK*

2.2.12 AT+SSP

Description: Read/Setting simple pairing mode, default: 1

Format: AT+SSP{=param}

param(0~1)

0: Turn off simple pairing mode

1: Turn on simple pairing mode

Response: +SSP=param

Eg: Read Simple pairing mode

<< *AT+SSP*

>> *+SSP=1*

>> *OK*

2.2.13 AT+LPM

Description: Read/Setting Low Energy Mode;
Default: 0

Format: AT+LPM {=param}

param:(0~1)
 (0) Turn off low energy mode
 (1) Turn on low energy mode. After entering low-power mode, Bluetooth can broadcast and be connected. There are two ways to wake up. 1. The serial port sends the first packet of data to wake up. After waking up, the serial port starts to work. If no serial port data or APP data is received within 10 seconds, it will automatically Close the serial port and enter low power consumption again. 2. The APP sends data to wake up. Under low power consumption conditions, receiving APP data will wake up the serial port and output data. If no serial port or APP data is received within 10s, the serial port will automatically close and enter low power consumption mode. The time (10s) can be changed through the **AT+LPDLY** command

Response: +LPM=param

Eg1: Read Bluetooth status

```
<< AT+LPM
>> +LPM=0
>> OK
```

Eg2: Setting Low Energy Mode (**Changes take effect immediately**)

```
<< AT+LPM=1
>> OK
```

2.2.14 AT+REBOOT

Description: Software reset, system restart

Format: AT+REBOOT

Response: OK

Eg:

```
<< AT+REBOOT
>> OK
```

2.2.15 AT+RESTORE

Description: All Bluetooth parameters are restored to factory settings and the system is restarted.
Format: AT+RESTORE
Response: OK
Eg: << <i>AT+RESTORE</i> >> <i>OK</i>

2.2.16 AT+TXPOWER

Description: Read/ Setting Bluetooth module Transmitter Power, default: 00DBm)
Format: AT+TXPOWER{=param}
Param (0~4)
Response: +TXPOWER=param
Eg1: Read << <i>AT+TXPOWER</i> >> <i>+TXPOWER=0</i> >> <i>OK</i>
Eg2: Setting 5DBm transmitter power (Changes take effect immediately) << <i>AT+TXPOWER=4</i> >> <i>OK</i>

2.2.17 AT+PIN

Description: Read/Setting Connection PIN Code, default: 0000
Format: AT+PIN{=param}
param: Pin code (4~15 Bytes ASCII)
Response: +PIN=param
<p>Eg1: Read</p> <pre><< AT+PIN >> +PIN=0000 >> OK</pre>
<p>Eg2: Setting PIN Code: 1234 (Changes take effect immediately)</p> <pre><< AT+PIN=1234 >> OK</pre>

2.2.18 AT+PLIST

Description: Read/ Clear pairing records
Format: AT+PLIST{=param}
Param(0/1~5) 0: Clear all pairing records 1~5: Clear the pairing records of the specified index
Response: +PLIST=param1,param2 param1: index number param2: Bluetooth pairing address
Eg: Read pairing records


```
<< AT+PLIST
>> +LIST=1,000000000000
>> OK
```

2.2.19 AT+BTEN

Description: Read/Setting Bluetooth module work mode, default: 1

Format: AT+BTEN{=param}

Param(0~1)

0: Disconnect all connections and disable discoverable/connectable mode

1: Enable discoverable/connectable mode

Response: +BTEN=param

Eg1: Read Bluetooth module work mode

```
<< AT+BTEN
```

```
>> +BTEN=1
```

```
>> OK
```

Eg2: Disable discoverable/connectable mode (**changes take effect immediately**)

```
<< AT+BTEN=0
```

```
>> OK
```

2.2.20 AT+CLOSEAC

Description: Close AirCommand Mode (**Only for AirCommand Mode**)

Format: AT+CLOSEAC

Response: AirCommandClosed
Eg: << <i>AT+CLOSEAC</i> >> <i>AirCommandClosed</i>

2.2.21 AT+GPIOCFG

Description: Control the input function switch of two IOs, default (0,0)
Format: AT+GPIOCFG{=param1{,param2}}
param1:(0~1) (0) Disable command/transparent transmission mode switching function (1) Enable command/transparent transmission mode switching function param2:(0~1) (0) Disable Bluetooth disconnect function (1) Enable Bluetooth disconnect function
Response: +GPIOCFG=param
Eg1: Read IO Status << <i>AT+GPIOCFG</i> >> <i>+GPIOCFG=0,0</i> >> <i>OK</i>
Eg2: Enable command/transparent transmission mode switching function; disable Bluetooth disconnect function (Changes take effect after restart) >> <i>AT+GPIOCFG=1,0</i> >> <i>OK</i>
Eg3: Disable the command/transparent transmission mode switching function and enable the Bluetooth disconnection function. (Changes take effect after restart) >> <i>AT+GPIOCFG=0,1</i> >> <i>OK</i>

2.2.22 AT+DISC

Description: Disconnect Bluetooth (Only be used in command mode)
Format: AT+DISC{=param}
Response: OK
param: Without parameters, disconnect all connections; with parameters, disconnect the specified channel.
<p>Eg1: Disconnect all connections</p> <pre><< AT+DISC >> OK</pre> <p>Eg2: Disconnect devices of Channel 0</p> <pre><< AT+DISC=0 >> OK</pre>

2.3 SPP Command

2.3.1 AT+SPPSTAT

Description: Get the status of the SPP connection (in command mode)
Format: AT+SPPSTAT
Response: +SPPSTAT=param
param(1~3): 1 Not connected, 2 Connecting, 3 Successfully connected

Eg: Read SPP connection status

```
<< AT+SPPSTAT
>> +SPPSTAT=3
>> OK
```

2.3.2 AT+SPPDISC

Description: Disconnect SPP connection

Format: AT+SPPDISC

Eg:

```
<< AT+SPPDISC
>> OK
```

2.3.3 AT+SPPSEND

Description: Send data to the remote device (**only in command mode**)

Format: AT+SPPSEND =param1, param2

Param1: Data Length (1~127)

Param2: Data (1~127Bytes)

Response: OK

Eg: Send data "0123456789" to remote device

```
<< AT+SPPSEND=10,0123456789
>> OK
```

2.4 GATT Command

2.4.1 AT+GATTSTAT

Description: Check Bluetooth connection State
Format: AT+GATTSTAT
Response: +GATTSTATE=param1
param1(0~8): Connection Channel param2(1~3):1. Not connected, 2 Connecting, 3 Connected
Eg: Check Bluetooth connection State << <i>AT+GATTSTAT</i> >> <i>+GATTSTATE=3</i> >> <i>OK</i>

2.4.2 AT+IBEACON

Description: Read/ Setting iBeacon broadcasting function, default: 1
Format: AT+IBEACON {=param}
param:(0~2) (0) Turn off iBeacon (1) Turn on iBeacon, set the iBeacon content through the command AT+ADVDATA
Response: +IBEACON=param

Eg: Read

```
<< AT+IBEACON
>> +IBEACON=1
>> OK
```

Eg2: turn off iBeacon broadcast (Changes take effect immediately)

```
<< AT+IBEACON=0
>> OK
```

2.4.3 AT+ADVDATA

Description: Read, Setting iBeacon broadcast content

Format: AT+ADVDATA{=param}

param: iBeacon data (2~56 Bytes ASCII)

Response: +ADVDATA=param

Eg1: Read iBeacon broadcast content

```
<< AT+ADVDATA
>> +ADVDATA=4C0002155B198FF269A011EE8C990242AC12000200000000B5
>> OK
```

Eg2: Setting iBeacon broadcast data:

```
0x4C 0x00 0x02 0x15 0x5B 0x19 0x8F 0xF2 0x69 0xA0 0x11 0xEE 0x8C 0x99 0x02 0x42 0xAC 0x12 0x00
0x02 0x00 0x00 0x00 0x00 0xB6(Changes take effect immediately)
<< AT+ADVDATA=4C0002155B198FF269A011EE8C990242AC12000200000000B6
>> OK
```

2.4.4 AT+ADVADDR

Description: Broadcast Bluetooth MAC address, default: 1
Format: AT+ADVADDR{=param}
param:(0~1) (0) Turn off broadcast MAC address function. (1) Turn on broadcast MAC address function
Response: +ADVADDR=param
Eg1: Read Broadcast Bluetooth MAC address << AT+ADVADDR >> +ADVADDR=1 >> OK
Eg2: Turn off the broadcasting of MAC address (Changes take effect immediately) << AT+ADVADDR=0 >> OK

2.4.5 AT+GATSEND

Description: Send data to remote device (Only on Command mode using)
Format: AT+GATSEND =param1, param2,
param1: Data length (1~127) Param2: Data (1~127Bytes)

Response: OK

Eg: Send the data "0123456789" to remote device

<< `AT+GATTSEND=10,0123456789`

>> `OK`

3. Detailed explanation of air commands

MY-BT10X supports entering AT command mode through air data. After entering air command mode, you can send commands to change related parameters.

Usage steps:

1. Use APP to establish a connection with the Bluetooth module
2. APP sends **\$OpenMysAtEngine\$** to the Bluetooth module. If APP can receive the data **\$OK,Opened\$** fed back by the module, it successfully enters the air command mode
3. The command format of air commands and UART is the same. Please refer to 2.1~2.4.5
4. After changing the parameters, you can send **AT+CLOSEAC** command to close the air command mode. If it is closed successfully, you will receive a reply **AirCommandClosed** or disconnect the Bluetooth connection to automatically exit the air command mode.

Note: In order to prevent errors in transparent data transmission, if the first data packet received by the module after Bluetooth connection is not **\$OpenMysAtEngine\$**, the module will no longer enter the air command mode unless it is disconnected and reconnected.

4. Note:

1. **Throughput Mode:** any data received by UART will be sent to the remote device through GATT or HID protocol.
2. **Command mode:** any data received by UART in any state will be parsed and processed, and will not be sent directly to the remote device.
3. **Air command mode:** any data sent by the remote device will be parsed and processed, and will not be sent out through UART.