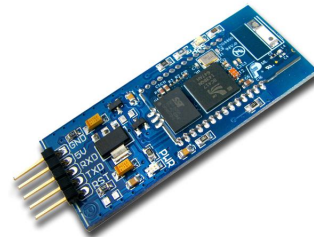


**Document** : Datasheet  
**Date** : 05-Apr-10

**Model #** : WRL - 1323

## BlueLINK Bluetooth Module - External Commands



**Rhydo Technologies (P) Ltd.**  
(An ISO 9001:2008 Certified R&D Company)  
Golden Plaza, Chitoor Road,  
Cochin – 682018, Kerala State, India  
**Phone** : 0091- 484-2370444, 2371666  
**Cell** : 0091- 99466 70444  
**Fax** : 0091 - 484-2370579  
**E-mail** : [info@rhydolabz.com](mailto:info@rhydolabz.com), [sales@rhydolabz.com](mailto:sales@rhydolabz.com)  
**WebSite** : <http://www.rhydolabz.com>

## TABLE OF CONTENTS

- 1. INTRODUCTION**
- 2. BlueLINK - DIFFERENT MODES**
- 3. BlueLINK - COMMAND SET**
  - 3.1** ENTER COMMAND MODE
  - 3.2** ENTER REMOTE COMMAND MODE
  - 3.3** ESCAPE FROM COMMAND MODE
  - 3.4** LIST ALL THE SUPPORTED COMMANDS
  - 3.5** RESET TO DEFAULT CONFIGURATION
  - 3.6** READ THE CURRENT SETTINGS
  - 3.7** REBOOT THE MODULE
  - 3.8** SET THE MODE OF THE MODULE
  - 3.9** READ THE MODE OF THE MODULE
  - 3.10** READ THE BAUDRATE SETTING
  - 3.11** SET THE BAUDRATE
  - 3.12** SET THE RF POWER
  - 3.13** READ THE RF POWER
  - 3.14** READ THE LOCAL NAME
  - 3.15** WRITE THE LOCAL NAME
  - 3.16** READ THE PIN
  - 3.17** WRITE THE PIN
  - 3.18** READ THE SECURITY SETTINGS
  - 3.19** WRITE THE SECURITY SETTINGS
  - 3.20** READ THE DISCOVERABLE MODE SETTINGS
  - 3.21** WRITE THE DISCOVERABLE MODE SETTINGS
  - 3.22** READ THE CURRENT ACTIVE PROFILE
  - 3.23** SET THE CURRENT ACTIVE PROFILE

- 3.24** READ THE CURRENT PAIRED DEVICE
- 3.25** WRITE THE PIN CODE FOR A REMOTE DEVICE
- 3.26** READ SERVICE SPECIFIC SECURITY SETTINGS
- 3.27** SET SERVICE SPECIFIC SECURITY SETTINGS
- 3.28** READ LOCAL BLUETOOTH ADDRESS
- 3.29** READ CLASS OF DEVICE
- 3.30** SET CLASS OF DEVICE
- 3.31** READ THE RSSI FOR A BLUETOOTH ADDRESS
- 3.32** START AN INQUIRY SESSION
- 3.33** CONNECT TO A REMOTE DEVICE
- 3.34** DISCONNECT A LINK WITH A REMOTE DEVICE
- 3.35** START AN OBEX SESSION
- 3.36** PUT START
- 3.37** PUT NEXT
- 3.38** OBEX SESSION REQUEST
- 3.39** GET START
- 3.40** GET NEXT

## 1. INTRODUCTION

The purpose of this document is to define the protocol between the HOST system and BlueLINK. The protocol heavily depends on the AT commands of 3GPP 27.007 and 27.005 GSM recommendations. Most of the commands are the same with the GSM standard, with several special commands.

For the exchange of the commands and unsolicited results codes, the format, syntax and procedures of 3GPP 27.007 shall be taken as reference.

The following rules specifically apply for this protocol.

- ✓ Only one command (or unsolicited result code) per command line needs to be expected.
- ✓ The HOST, by default, shall not echo the command characters.
- ✓ The HOST shall always transmit result codes using verbose format.
- ✓ The characters below shall be used for AT commands and result codes formatting:

<cr> Corresponds to the carriage return

<lf> Corresponds to the line feed

- ✓ The format of an AT command from the HOST to the module shall be:

**<cr><lf><AT command><cr> <lf>**

- ✓ The format of the OK code from the module to the HOST shall be:

**<cr><lf>OK<cr><lf>**

- ✓ The format of the generic ERROR code from the module to the HOST shall be:

**<cr><lf>ERROR<cr> <lf>**

- ✓ The format of an unsolicited result code from the module to the HOST shall be:

**<cr><lf>RESULT CODE<cr> <lf>**

## 2. BLUELINK – DIFFERENT MODES

The module has three different working modes:

- **Data and connected mode**

In this mode, the SPP connection has been established and all the data from the UART are forwarded to the SPP connection and all the data from the SPP connection are forwarded to the UART.

- **Data and command mode**

In this mode, the SPP connection has been established. But all the data from the UART would not be forwarded to the SPP connection and all the data from the SPP connection are discarded.

- **Command mode**

In this mode, no SPP connection has been established. And all the data from the UART would be treated as AT commands. The module would enter command mode in default after powered up. In this mode the module would not be discovered by any other Bluetooth devices and would only react to the commands from the UART. The host could use commands to make the module discoverable, to find other Bluetooth devices nearby or to connect with one of them.

### Message Format

The GSM 07.07 format and syntax rules shall be taken as the reference for these commands.

### 3. BlueLINK COMMAND SET

### 3.1

#### ENTER COMMAND MODE

COMMAND	PARAMETER
LLL	

#### Description:

This command is used to force the module to enter command mode. The module would parse the stream received on the UART and looking for this command during the first 60 seconds after the module is powered up. The module would react to other commands listed in this document only after it received this command.

If the module is in Auto Connect mode and has a SPP link with a remote device, the module would send every byte received on the UART to the remote device before it finally gets this command. And it will not send any data to the remote device once it is in command mode.

The command mode would end only after the module receives Escape (==) command.

During command mode, the user could issue any commands to configure and operate the module.

After the module leaves the command mode, the module would start the 60 seconds timer again and could enter command mode again if this command is received again.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example command:

LLL

#### Example response

<cr><lf>OK <cr><lf>

## 3.2

### ENTER REMOTE COMMAND MODE

COMMAND	PARAMETER
RRR	

**Description:**

A module could force a remote device to enter command mode and could configure and operate the remote module afterwards, provided that the SPP connection has been established. The host could send any commands listed in this document, except for the 'LLL', to the remote module and all the commands have the same meaning as to the local module.

The remote device would only accept this command during the 60 seconds time frame after powered up.

**Response**

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

**Example command:**

RRR

**Example response**

<cr><lf>OK <cr><lf>



### 3.3

#### ESCAPE FROM COMMAND MODE

COMMAND	PARAMETER
===	

#### Description:

This command makes the module to escape the command mode. This command would only be valid after the module has entered command mode.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example command:

===

#### Example response

<cr><lf>OK <cr><lf>

### 3.4

#### LIST ALL THE SUPPORTED COMMANDS

COMMAND	PARAMETER
AT	?

#### Description:

This command will return a list with all the supported commands by this module.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
Commands	The name of the commands

#### Example command:

```
<cr><lf>AT?<cr><lf>
```

#### Example response

```
<cr><lf>OK<cr><lf>
```

```
<cr><lf>AT? <cr><lf>AT+RESET<cr><lf>
```

### 3.5

#### RESET TO DEFAULT CONFIGURATION

COMMAND	PARAMETER
AT+F	

#### Description:

Reset the module to its factory default configuration

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+F<cr><lf>
```

#### Response

```
<cr><lf>OK<cr><lf>
```

### 3.6

#### READ THE CURRENT SETTINGS

COMMAND	PARAMETER
AT+SETUP	

#### Description:

This command will return a list of current settings.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example command:

```
<cr><lf>AT+SETUP<cr><lf>
```

#### Example response

```
<cr><lf>BAUD:9600,1,0
<cr><lf>PIN:8888
<cr><lf>+PIO:0,0,0,0,0,0,0,0
<cr><lf>+SNIFF:0
<cr><lf>+MODE:1
<cr><lf>+CON:0
<cr><lf>Firmware ver: v1.0
<cr><lf>+NAME: BlueLINK
<cr><lf>+BLBDADDR:00:19:A4:01:00:00<cr><lf>
```

### 3.7

#### REBOOT THE MODULE

COMMAND	PARAMETER
AT+REBOOT	

**Description:** This command would reboot the module immediately

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example command:

<cr><lf>AT+REBOOT<cr><lf>

#### Example response

<cr><lf>OK<cr><lf>

### 3.8

#### SET THE MODE OF THE MODULE

COMMAND	PARAMETER
AT+MODE=	mode

#### Description:

The module has three modes: Master, Slave and Auto Connect mode. A module in Master mode could inquiry and search for devices nearby and initiate connection with them. A module in Slave mode would wait for connection request from modules in Master mode. A slave module would not accept commands for inquiry and connect. A module in Auto Connect mode would try to connect to the last connected module, no matter it is working or not. The Bluetooth address of the last connected device would be stored in a register called Connect ID in the module.

#### Parameter:

Mode

VALUE	PARAMETER DESCRIPTION
0	Slave
1	Master
2	Auto connect mode

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example command:

```
<cr><lf>AT+MODE=1<cr><lf>
```

#### Example response

```
<cr><lf>OK<cr><lf>
```

### 3.9

#### READ THE MODE OF THE MODULE

COMMAND	PARAMETER
AT+MODE=?	

**Parameter:**

**Response**

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+MODE:<mode>	

**Response Parameter:**

mode

VALUE	PARAMETER DESCRIPTION
0	Slave
1	Master
2	Auto connect mode

**Example command:**

<cr><lf>AT+MODE=?<cr><lf>

**Example response**

<cr><lf>+MODE:1<cr><lf>

### 3.10

#### READ THE BAUDRATE SETTING

COMMAND	PARAMETER
AT+BAUD=?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+BAUD: < UART baud rate>,< stop>,< parity>	

#### Response Parameter

VALUE	DESCRIPTION
UART baud rate	"0" = no change "9600" = 9600 baud "19200" = 19200 baud "38400" = 38400 baud "57600" = 57600 baud "115200" = 115200 baud "230400" = 230400 baud "460800" = 460800 baud "921600" = 921600 baud "1382400" = 1382400 baud
stop	"0" = no change "1" = one "2" = two
parity	"0" = no change "1" = none "2" = odd "3" = even

#### Example

```
<cr><lf>AT+BAUD=?<cr><lf>
```

#### Response

```
<cr><lf>OK<cr><lf><cr><lf>+BAUD:9600,1,0<cr><lf>
```



### 3.11

#### SET THE BAUDRATE

COMMAND	PARAMETER
AT+BAUD=	< UART baud rate>,< stop>,< parity>

#### Description:

Set the parameter of UART port

#### Parameter

VALUE	DESCRIPTION
UART baud rate	"0" = no change "9600"=9600 baud "19200"=19200 baud "38400"=38400 baud "57600"=57600 baud "115200"=115200 baud "230400"=230400 baud "460800"=460800 baud "921600"= 921600 baud "1382400"=1382400 baud
stop	"0"=no change "1"=one "2"=two
parity	"0"=no change "1"=none "2"=odd "3"=even

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+BAUD=57600,0,0<cr><lf>
```

#### Response

```
<cr><lf>OK <cr><lf>
```

### 3.12

#### SET THE RF POWER

COMMAND	PARAMETER
AT+POWER=	<DefaultPower>, <MaximumPower>

#### Description:

Set the maximum transmit power and default transmit power of the module, measured in dBm

#### Command Parameter

VALUE	DESCRIPTION
Default Power	format: x, “x” is from -128 to 127
Maximum Power	format: x, “x” is from -128 to 127

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+POWER: “0”, “4”<cr><lf>
```

#### Response

```
<cr><lf>OK <cr><lf>
```

### 3.13

#### READ THE RF POWER

COMMAND	PARAMETER
AT+POWER=?	

#### Description:

Read the maximum transmit power and default transmit power of the module, measured in dBm

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+POWER: <DefaultPower>,<MaxiumPower>	

#### Response Parameter

VALUE	DESCRIPTION
DefaultPower	format: x, "x" is from -128 to 127
MaximumPower	format: x, "x" is from -128 to 127

#### Example:

```
<cr><lf>AT+POWER=?<cr><lf>
```

#### Response

```
<cr><lf>+POWER:1,4<cr><lf>
```

### 3.14

#### READ THE LOCAL NAME

COMMAND	PARAMETER
AT+NAME=?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+NAME: <name>	

#### Response Parameter:

VALUE	DESCRIPTION
Name	A string contain the user's friendly name of the local module The maximum length of name is 20 characters.

#### Example:

<cr><lf>AT+NAME=?<cr><lf>

#### Response

<cr><lf>+NAME:BlueLINK<cr><lf>

### 3.15

#### WRITE THE LOCAL NAME

COMMAND	PARAMETER
AT+NAME=	<name>

#### Description:

Set the local device's "user friendly" name

#### Parameter

VALUE	DESCRIPTION
name	A string contain the user's friendly name of the local module The maxim length of name is 20 characters.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+NAME=BlueLINK<cr><lf>
```

#### Response

```
<cr><lf>OK<cr><lf>
```

### 3.16

#### READ THE PIN

COMMAND	PARAMETER
AT+PIN=?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+PIN:<PIN code>	

#### Response Parameter PIN code

VALUE	DESCRIPTION
	The pin code to return when other Bluetooth devices pair with. The maximum length of pin code is 16 digits.

#### Example:

<cr><lf>AT+PIN=?<cr><lf>

#### Response

<cr><lf>+PIN:8888<cr><lf>

### 3.17

#### WRITE THE PIN

COMMAND	PARAMETER
AT+PIN=	PIN code

#### Description:

Set Bluetooth PIN code of local module. This is the PIN code that the local machine would send when it receives an authentication request from a remote device.

#### Command Parameter

PIN code

VALUE	DESCRIPTION
	The pin code to return when other Bluetooth devices pair with. The maximum length of pin code is 16 digits.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+PIN=8888<cr><lf>
```

#### Response

```
<cr><lf>OK<cr><lf>
```

### 3.18

#### READ THE SECURITY SETTINGS

COMMAND	PARAMETER
AT+ BSECMODE =?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+BSECMODE:<mode>,<mode>	

#### Response Parameter <mode>

VALUE	DESCRIPTION
0	Bluetooth Security Mode 1 (Link Level Authentication and Encryption Disabled)
1	Application level security
2	Bluetooth Security Mode 3 (Link Level Authentication and Encryption Enabled)

#### Example:

<cr><lf>AT+BSECMODE=?<cr><lf>

#### Response

<cr><lf>+BSECMODE:1,1<cr><lf>



### 3.19

#### WRITE THE SECURITY SETTINGS

COMMAND	PARAMETER
AT+ BSECMODE =	<mode>,<mode>

#### Description:

Set the security mode. By default Security Mode is set to 1

#### Parameter

<mode>

VALUE	DESCRIPTION
0	Bluetooth Security Mode 1 (Link Level Authentication and Encryption Disabled)
1	Application level security
2	Bluetooth Security Mode 3(Link Level Authentication and Encryption Enabled)

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+MODE=0,0<cr><lf>
```

#### Response

```
<cr><lf>OK<cr><lf>
```

### 3.20

#### READ THE DISCOVERABLE MODE SETTINGS

COMMAND	PARAMETER
AT+ BDISCMODE =?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+BDISCMODE:<mode> , <timeout>	

#### Response Parameter

<mode>

VALUE	DESCRIPTION
0	Not Discoverable and Not Connectable
1	General Discoverable and Not Connectable
2	Not Discoverable and Connectable
3	General Discoverable and Connectable

<timeout>

VALUE	DESCRIPTION
0	Infinite always device is set to the 'mode' as requested in this command
<1 .. 180>	time out Value

#### Example:

<cr><lf>+BDISCMODE=?<cr><lf>

#### Response

<cr><lf>OK<cr><lf>

### 3.21

#### WRITE THE DISCOVERABLE MODE SETTINGS

COMMAND	PARAMETER
AT+ BDISCMODE =	<mode>, <timeout>

#### Description:

Set the device in specified type of discoverable mode for the duration specified in this command. At the expiration of the time period the device is not discoverable and connectable.

#### Parameter <mode>

VALUE	DESCRIPTION
0	Not Discoverable and Not Connectable
1	General Discoverable and Not Connectable
2	Not Discoverable and Connectable
3	General Discoverable and Connectable

#### <timeout>

VALUE	DESCRIPTION
0	Infinite always device is set to the 'mode' as requested in this command
<1 .. 180>	time out Value
> 180	Any Value larger than 180 is considered as 180 seconds

Default Value for Discovery Mode is 1  
i.e. always discoverable and connectable And Discovery Time is 180s

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+BDISCMODE=1, 45<cr><lf>
```

#### Response

```
<cr><lf>OK<cr><lf>
```

### 3.22

#### READ THE CURRENT ACTIVE PROFILE

COMMAND	PARAMETER
AT+ BPROFILE =?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+BPROFILE:<profile>	

#### Response Parameter

<profile>

VALUE	DESCRIPTION
0	No Active Profile
1	Serial Port Profile
2	Head-Set Profile (Future)
3	Hands-Free Profile (Future)

#### Example:

<cr><lf>AT+BPROFILE=?<cr><lf>

#### Response

<cr><lf>+BPROFILE:1<cr><lf>

### 3.23

#### SET THE CURRENT ACTIVE PROFILE

COMMAND	PARAMETER
AT+ BPROFILE=	<profile>

#### Description:

This command is used to choose the “Current Active Profile”. This command enables the user application to define the current context in which the application decides to work. If “No Active Profile” is enabled, profile specific commands mentioned in the subsequent sections will not be processed. If “No Active Profile” is enabled and peer initiated profile connections are received by the application, the user application should change the context using this command to use the features on that profile connection.

#### Parameter <profile>

VALUE	DESCRIPTION
0	No Active Profile
1	Serial Port Profile
2	Head-Set Profile (Future)
3	Hands-Free Profile (Future)

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+BPROFILE=1<cr><lf>
```

#### Response

```
<cr><lf>OK<cr><lf>
```

### 3.24

#### READ THE CURRENT PAIRED DEVICE

COMMAND	PARAMETER
AT+ BPAIR =?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+BPAIR:<BD-ADDR>	

#### Response Parameter BD-ADDR

VALUE	DESCRIPTION
	format: xx:xx:xx:xx:xx:xx, “x” is from 0 to F

#### Example:

<cr><lf>AT+BPAIR=?<cr><lf>

#### Response

<cr><lf>+BPAIR: 01:2a:12:01:00:1f <cr><lf>

### 3.25

#### WRITE THE PINCODE FOR A REMOTE DEVICE

COMMAND	PARAMETER
AT+BPAIR =	<Pin Code>,<BD_ADDR>

#### Description:

This command will store the remote device PIN Code. The stored PIN code will be used during connection setup and pairing with peer. The information of a maximum of 5 devices can be stored in the database. Note that this list consists of devices which are not paired.

**Parameter** pin code

VALUE	DESCRIPTION
	The pin code to store. The maxim length of pin code is 16 digits.

BD-ADDR

VALUE	DESCRIPTION
	format: xx:xx:xx:xx:xx:xx, “x” is from 0 to F

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

**Example:** <cr><lf>AT+BPAIR=1290,01:2a:12:01:00:1f<cr><lf>

**Response** <cr><lf>OK<cr><lf>

### 3.26

#### READ SERVICE SPECIFIC SECURITY SETTINGS

COMMAND	PARAMETER
AT+ BSERVICEATTR =?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+BSERVICEATTR:<service_type>,<authentication>,<encryption>	

#### Response Parameter

##### Service Type

VALUE	DESCRIPTION
1	Serial Port Profile
2	Head-Set Profile
3	Hands-Free Profile

##### Authentication

VALUE	DESCRIPTION
0	disable
1	enable

##### Encryption

VALUE	DESCRIPTION
0	disable
1	enable

#### Example:

```
<cr><lf>AT+BSERVICEATTR=?<cr><lf>
```

#### Response

```
<cr><lf>+BSERVICEATTR:1,0,1<cr><lf>
```



### 3.27

#### SET SERVICE SPECIFIC SECURITY SETTINGS

COMMAND	PARAMETER
AT+ BSERVICEATTR =	<service_type>,<authentication>,<encryption>

**Description:** This command is used to register security requirements for specific services. The Values registered through this command will be used only when device is operated in Security Mode 2. Enabling encryption on Bluetooth link mandates authentication. This command will not allow enabling the encryption of service types with authentication being disabled.

#### Parameter

Service\_Type

VALUE	DESCRIPTION
1	Serial Port Profile
2	Head-Set Profile
3	Hands-Free Profile

#### Authentication

VALUE	DESCRIPTION
0	disable
1	enable

#### Encryption

VALUE	DESCRIPTION
0	disable
1	enable

Default settings for all services is

Authentication – disabled

Encryption - disabled

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

**Example:** <cr><lf>AT+BSERVICEATTR=1,0,1<cr><lf>

**Response** <cr><lf>OK<cr><lf>

### 3.28

#### READ LOCAL BLUETOOTH ADDRESS

COMMAND	PARAMETER
AT+ BLBDADDR	

Description: Get the local device BD ADDR.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+ BLBDADDR:<Bluetooth address>	

#### Example:

<cr><lf>AT+BLBDADDR<cr><lf>

#### Response

<cr><lf>+BLBDADDR:00:12:34:56:78:AB<cr><lf>

### 3.29

#### READ CLASS OF DEVICE

COMMAND	PARAMETER
AT+BCOD=?	

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+BCOD:<CODbitmap>	

#### Response Parameter <cod bitmap>

VALUE	DESCRIPTION
0	Should be set depending on the end product device type as desired by the manufacturer. The Value for 'cod bitmap' should be in Hex format.

#### Response

OK  
<cod bitmap>

### 3.30

#### SET CLASS OF DEVICE

COMMAND	PARAMETER
AT+BCOD=	<cod bitmap>

#### Description:

This command is used to set the Bluetooth device type.

#### Parameter

<cod bitmap>

VALUE	DESCRIPTION
0	Should be set depending on the end product device type as desired by the manufacturer. The Value for 'cod bitmap' should be in Hex format.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed

#### Example:

```
<cr><lf>AT+BCOD=520204<cr><lf>
```

#### Response

```
<cr><lf>OK<cr><lf>
```

### 3.31

#### READ THE RSSI FOR A BLUETOOTH ADDRESS

COMMAND	PARAMETER
AT+RSSI=	<Bluetooth address>

#### Description:

This command will read the Value for the difference between the measured Received Signal Strength Indication (RSSI) and the limits of the Golden Receive Power Range for a connection to another Bluetooth device. Any positive RSSI Value returned by this command indicates how many dB the RSSI is above the upper limit and any negative Value indicates how many dB the RSSI is below the lower limit. The Value zero indicates that the RSSI is inside the Golden Receive Power Range.

The RSSI measurement compares the received signal power with two threshold levels, which define the Golden Receive Power Range. The lower threshold level corresponds to a received power between -56 dBm and 6 dB above the actual sensitivity of the receiver. The upper threshold level is 20 dB above the lower threshold level to an accuracy of +/- 6 dB.

**Parameter** <Bluetooth address>

VALUE	DESCRIPTION
	format: xx:xx:xx:xx:xx:xx, "x" is from 0 to F

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+RSSI:<Bluetooth address>,<RSSI>	

#### Response Parameter:

<Bluetooth address>

VALUE	DESCRIPTION
	format: xx:xx:xx:xx:xx:xx, "x" is from 0 to F

< RSSI >

VALUE	DESCRIPTION
-128 ≤ N ≤ 127	Size: 1 Octet (signed integer) Units: dB

**Example:** <cr><lf>AT+RSSI=00:12:39:00:34:35 <cr><lf>

**Response** <cr><lf>+RSSI:00:12:39:00:34:35 , 110<cr><lf>

### 3.32

#### START AN INQUIRY SESSION

COMMAND	PARAMETER
AT+INQ	

#### Description:

This command will cause the module to enter Inquiry Mode. Inquiry Mode is used to discover other nearby Bluetooth devices. An “OK” is sent from the module to the Host when this command has been started by the module. When the Inquiry process is completed, the module will send an +INQ event to the Host indicating that the Inquiry has finished. The event parameters of +INQ event will have a result from the Inquiry process, which reports the address and name of Bluetooth devices that responded.

#### Response

VALUE	DESCRIPTION
OK	The command is successful
ERROR	The command is failed
+INQ:<Bluetooth address>,<name> [,<Bluetooth address>,<name>]	

#### Response Parameter

<Bluetooth address>

VALUE	DESCRIPTION
<name>	format: xx:xx:xx:xx:xx:xx, “x” is from 0 to F

<name>

VALUE	DESCRIPTION
name	A string contain the user’s friendly name of the local module

#### Example:

```
<cr><lf>AT+INQ<cr><lf>
```

#### Response

```
<cr><lf>+INQ:00:12:39:00:34:35,”RLMobile”,00:13:45:46:99:23,”Jeny”<cr><lf>
```

### 3.33

#### CONNECT TO A REMOTE DEVICE

COMMAND	PARAMETER
AT+CON	

#### Description:

This command will cause the module to establish a connection with the paired device.

#### Response

VALUE	DESCRIPTION
OK	The command is sent successfully
ERROR	The command is failed
+CON:<n>	Return the result of the request for connection

#### Parameter <n>

VALUE	DESCRIPTION
0	The connection has not been established
1	The connection has been established

**Example:** <cr><lf>AT+CON<cr><lf>

**Response :** <cr><lf>+CON:1<cr><lf>

### 3.34

#### DISCONNECT A LINK WITH A REMOTE DEVICE

COMMAND	PARAMETER
AT+DCON=	<profile>,<Bluetooth address>

#### Description:

This command will cause the module to disconnect with the Bluetooth device with the Bluetooth Address specified by the command parameters. The command also tells the module which profile to disconnect.

#### Parameter

<Bluetooth address>

VALUE	DESCRIPTION
	format: xx:xx:xx:xx:xx:xx, “x” is from 0 to F

<profile>

VALUE	DESCRIPTION
0	hands free
1	headset
2	spp
3	opp
4	A2DP

#### Response

VALUE	DESCRIPTION
OK	The command is sent successfully
ERROR	The command has failed
+DCON:<n>	Return the result of the disconnect request

#### Parameter <n>

VALUE	DESCRIPTION
0	The request has failed
1	The connection has been disconnected successfully

**Example:** <cr><lf>AT+DCON=1,00:12:39:00:34:35 <cr><lf>

**Response** <cr><lf>+DCON:1<cr><lf>



### 3.35

#### START AN OBEX SESSION

COMMAND	PARAMETER
AT+OPPCON=	<Bluetooth address>,<packSize>

#### Description:

Host uses this command to start a session with an OBEX server. This command would also negotiate with the server about the maximum packet size. The final packet size would be smaller one of the client and the server.

#### Parameter:

<Bluetooth Address>:

VALUE	DESCRIPTION
	format: xx:xx:xx:xx:xx:xx, “x” is from 0 to F

<packlength>:

VALUE	DESCRIPTION
	The length of each packet required by the client.

#### Response

VALUE	DESCRIPTION
OK	The command is correct
ERROR	The command is incorrect
+OPPCON:	< packetSize>

#### Response Parameter:

<packetSize>

VALUE	DESCRIPTION
	the packetSize negotiated by the OBEX server and client. The host would disassemble the object into packet of <packetSize> length.

### 3.36

#### PUT START

COMMAND	PARAMETER
AT+PUSHF=	<name>,<Totallength>,<length>,<end>

#### Description:

Host uses this command to start to push an object to an OBEX server. The host should start to send the binary code of the object after getting the OK message from the module.

#### Parameter:

<name>:

VALUE	DESCRIPTION
	The name of the object

<Totallength>:

VALUE	DESCRIPTION
	The total length of the object in bytes.

<length>:

VALUE	DESCRIPTION
	The amount of bytes which will be sent after this command. Note: This may be smaller than the packet size if this is the last packet

<end>:

VALUE	DESCRIPTION
0 or 1	0: There is still more to send 1: This is the last packet

#### Response

VALUE	DESCRIPTION
OK	The command is correct
ERROR	The command is incorrect

### 3.37

#### PUT NEXT

COMMAND	PARAMETER
AT+PUSHN=	<length>,<end>

#### Description:

Host uses this command to send the rest part of the object.

#### Parameter:

<length>:

VALUE	DESCRIPTION
	The amount of bytes which will be sent after this command.

<end>:

VALUE	DESCRIPTION
0 or 1	0: There is still more to send 1: This is the last packet

<length>:

VALUE	DESCRIPTION
	The amount of bytes which will be sent after this command. Note: This may be smaller than the packet size if this is the last packet

<end>:

VALUE	DESCRIPTION
0 or 1	0: There is still more to send 1: This is the last packet

#### Response

VALUE	DESCRIPTION
OK	
ERROR	

### 3.38

#### OBEX SESSION REQUEST

COMMAND	PARAMETER
AT+OPPREQ=	<Bluetooth address>,<packSize>

#### Description:

Host uses this command to notify that an OBEX client wants to start a session with the host. This command would also negotiate with the server about the maxim packet size. The final packet size would be smaller one of the client and the server.

#### Parameter:

<Bluetooth Address>:

VALUE	DESCRIPTION
<packlength>:	format: xx:xx:xx:xx:xx:xx, “x” is from 0 to F

VALUE	DESCRIPTION
	The length of each packet required by the client.

#### Response

VALUE	DESCRIPTION
OK	The command is correct
ERROR	The command is incorrect
+OPPREQ:	< packetSize>

#### Response Parameter:

<packetSize>

VALUE	DESCRIPTION
	the packetSize negotiated by the OBEX server and client. The host would disassemble the object into packet of <packetSize> length.

### 3.39

#### GET START

COMMAND	PARAMETER
AT+PULLF=	<name>,<Totallength>,<length>,<end>

#### Description:

When an OBEX client wants to push an object to the module, the module would send this message to the host. The host should return OK if it wants to accept this transfer, ERROR if not. The module would, after receiving the OK response, to send the object in binary mode. The amount of byte to be sent is specified by the parameter Totallength.

#### Parameter:

<name>:

VALUE	DESCRIPTION
	The name of the object

<Totallength>:

VALUE	DESCRIPTION
	The total length of the object in bytes.

<length>:

VALUE	DESCRIPTION
	The amount of bytes which will be sent after this command. Note: This may be smaller than the packet size if this is the last packet

<end>:

VALUE	DESCRIPTION
0 or 1	0: There is still more to send 1: This is the last packet

#### Response

VALUE	DESCRIPTION
OK	
ERROR	

### 3.40

#### GET NEXT

COMMAND	PARAMETER
AT+PULLN=	<length>,<end>

#### Description:

The host uses this command to send the next packet

#### Parameter:

<length>:

VALUE	DESCRIPTION
	The amount of bytes which will be sent after this command. Note: This may be smaller than the packet size if this is the last packet

<end>:

VALUE	DESCRIPTION
0 or 1	0: There is still more to send 1: This is the last packet

#### Response

VALUE	DESCRIPTION
OK	
ERROR	



## **TECHNICAL SUPPORT**

If you are experiencing a problem that is not described in this manual, please contact us. Our phone lines are open from 9:00 AM – 5.00 PM (*Indian Standard Time*) Monday through Saturday excluding holidays. Email can be sent to [support@rhydolabz.com](mailto:support@rhydolabz.com)

## **LIMITATIONS AND WARRANTIES**

This product is intended for personal or lab experimental purpose and in no case should be used where it harmfully effect human and nature. No liability will be accepted by the publisher for any consequence of its use. Use of the product software and or hardware is with the understanding that any outcome whatsoever is at the users own risk. All products are tested for their best performance before shipping, still rhydoLABZ is offering One year Free service warranty (Components cost + Shipping cost will be charged from Customer ).

## **DISCLAIMER**

Copyright © Rhydo Technologies (P) Ltd

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice.

### **Rhydo Technologies (P) Ltd.**

(An ISO 9001:2008 Certified R&D Company)

Golden Plaza, Chittoor Road,

Cochin – 682018, Kerala State, India

**Phone** : 0091- 484-2370444, 2371666

**Cell** : 0091- 99466 70444

**Fax** : 0091 - 484-2370579

**E-mail** : [info@rhydolabz.com](mailto:info@rhydolabz.com), [sales@rhydolabz.com](mailto:sales@rhydolabz.com)

**WebSite** : <http://www.rhydolabz.com>



**rhydolabz.com**