iWRAP 2-1-0

User Guide

Version 1.2

Thursday, February 02, 2006

bluegiga TECHNOLOGIES

Contents:

List of Tables:

VERSION HISTORY

Version:	Author:	Comments:
1.0	MSa	Initial Version, which is beta so information may change
1.1	MSa	Feature updates
1.2	MSa	Build 18 updates

TERMS & ABBREVIATIONS

Term or Abbreviation:	Explanation:
Bluetooth	Set of technologies providing audio and data transfer over short-range radio connections
bps	bits per second
CD	Carrier Detect
DTR	Data Terminal Ready
HCI	Host Controller Interface
iWRAP	Interface for WRAP – a trademark registered by Bluegiga Technologies
L2CAP	The Logical Link Control and Adaptation Layer Protocol
PARK state	Bluetooth low power mode
RFCOMM	Serial cable emulation protocol; element of Bluetooth
SNIFF mode	Bluetooth low power mode
UART	Universal Asynchronous Receiver Transmitter
UUID	Universally Unique Identifier
VM	Virtual Machine

WRAP	Wireless Remote Access Platform; Bluegiga Technologies' wireless product family

1. INTRODUCTION

In figure below, the iWRAP software solution is described. In iWRAP version of the stack firmware shown, no host processor is required to run the Bluetooth protocol stack. All software layers, including application software, run on the internal RISC processor in a protected user software execution environment known as a Virtual Machine (VM).

The host processor interfaces to iWRAP software via one or more of the physical interfaces which are also shown in the figure above. The most common interfacing is done via UART interface using the ASCII commands supported by the iWRAP software. With these ASCII commands the user can access Bluetooth functionality without paying any attention to the complexity which lies in the Bluetooth protocol stack.

The user may write applications code to run on the host processor to control iWRAP software with ASCII commands and to develop Bluetooth powered applications.

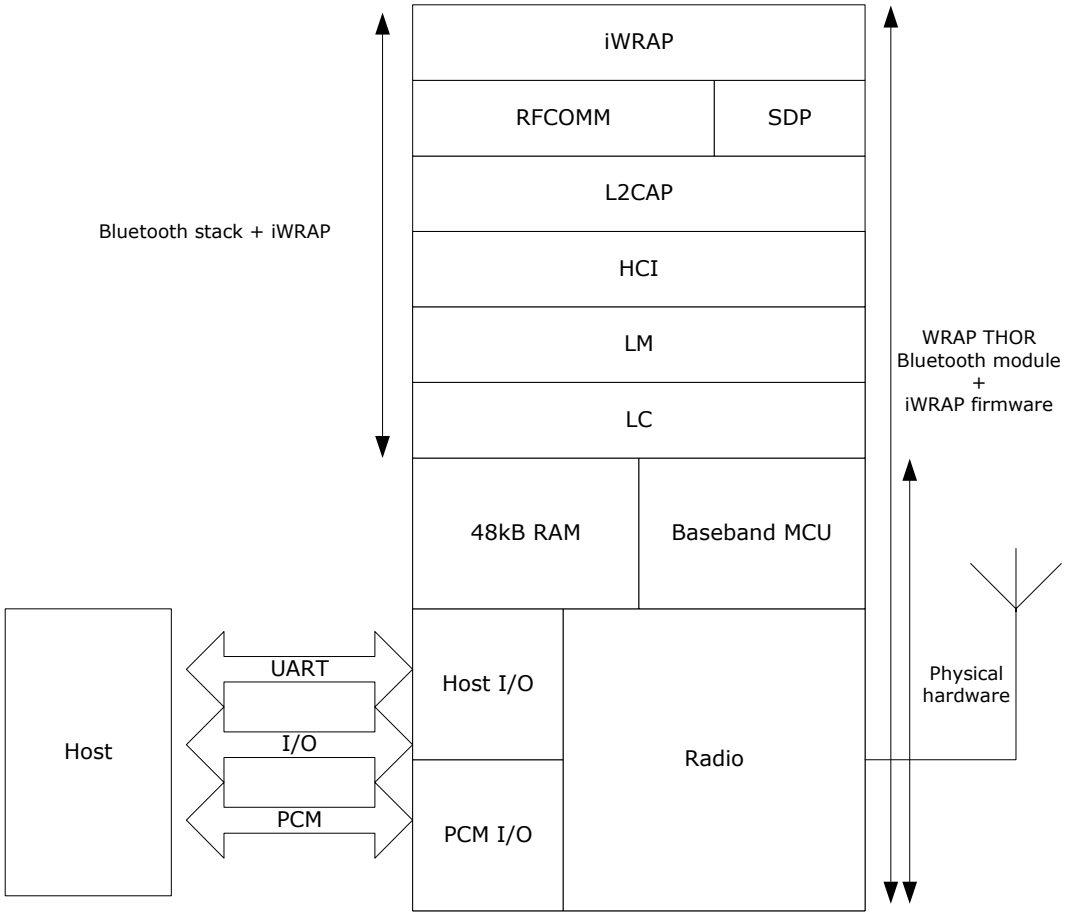


Figure 1: WRAP THOR iWRAP Stack

In the figure above a WRAP THOR Bluetooth module equipped with iWRAP firmware is connected to a host system using UART interface.

1. If host system has a processor, software can be used to control iWRAP with ASCII based commands.

2. If there is no need to control iWRAP or host system does not have a processor, iWRAP can be configured to be totally transparent only accepting connections or automatically opening them. Not all the functionality will be available with this solution.

3. GPIO lines offered by WRAP THOR 2022-1 can be also used together with iWRAP to achieve extra functionality such as DTR signaling or Carrier Detect signals.

4. PCM interface can be used to transmit audio over a Bluetooth link.

2. GETTING STARTED

To start using the iWRAP, you can use, for example, terminal software such as *HyperTerminal*. When using the terminal software, make sure the module or WRAP THOR module is connected to your PCs serial port. By default iWRAP uses following UART settings:

- Baud rate: 115200bps
- Data bits: 8
- Stop bits: 1
- Parity bit: No parity
- HW Flow Control: Enabled

When you power up your WRAP THOR module or evaluation kit you should see the command prompt appear on screen of the terminal software.

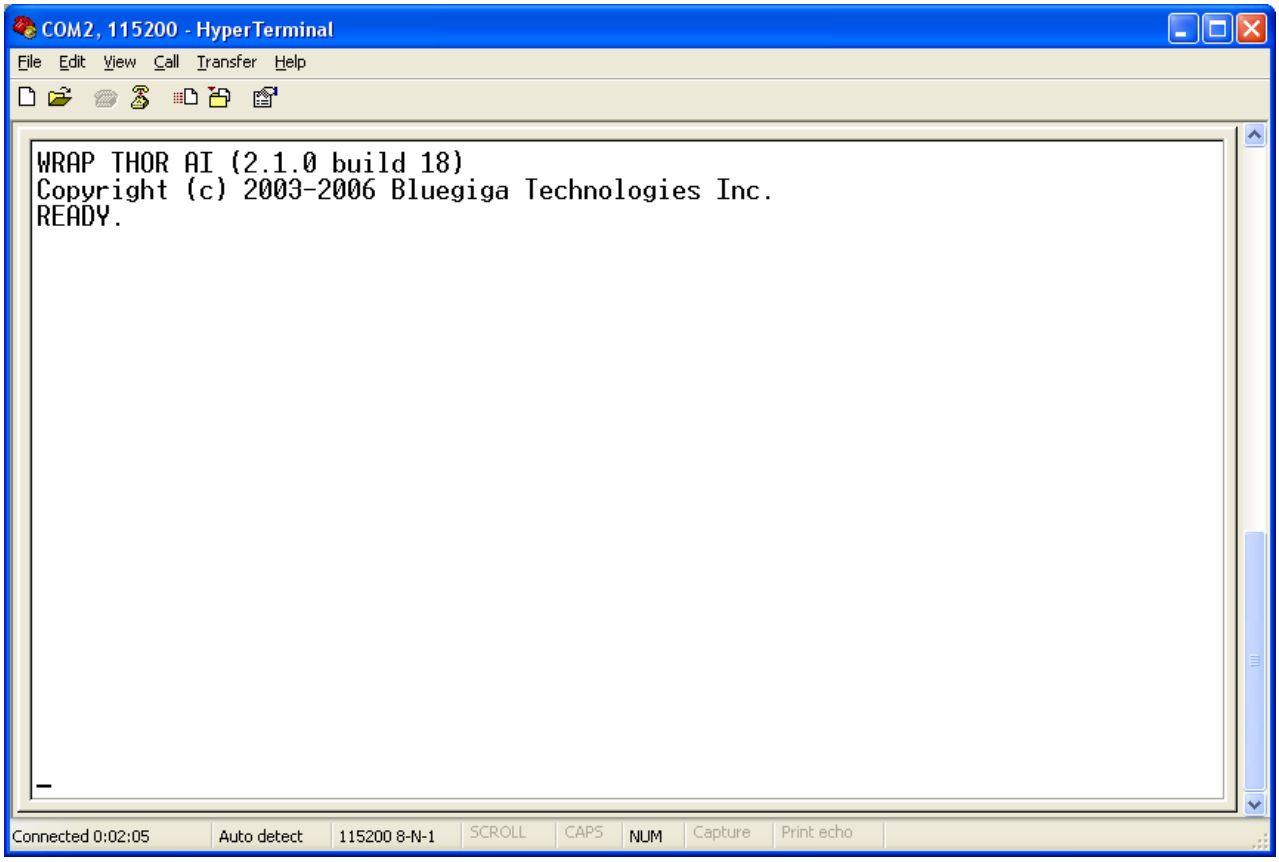


Figure 2: iWRAP boot prompt

3. IWRAP MODES

iWRAP has two operational modes, **command mode** and **data mode**. Command mode is default mode when there are no connections. It is possible to switch between modes at any time when there are one more more active connections. Data mode is not available if there are no active connections, because obviously there is no any data available, nor it can be sent anywhere.

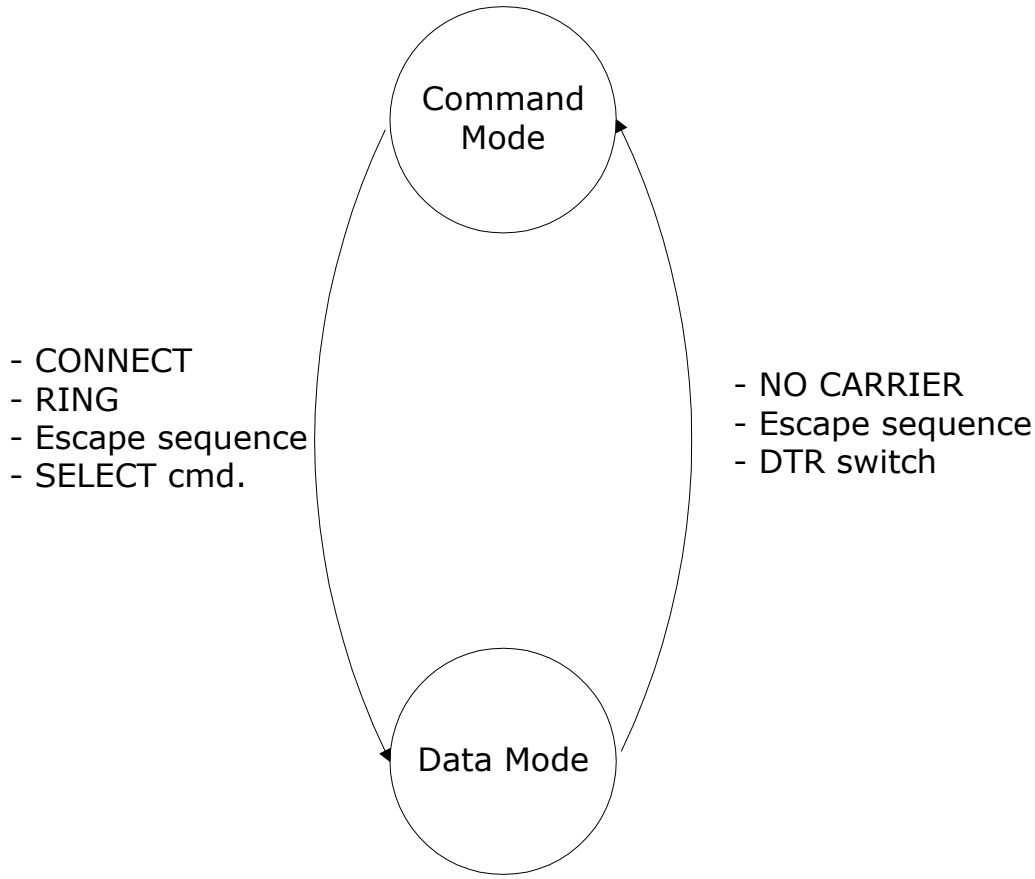


Figure 3: State Transitions

Switching from data mode to command mode is issued with the following escape sequence:

<at least 1 second sleep> +++ <at least 1 second sleep>

or it can also be done by using DTR signals.

Same sequence or **SELECT** -command may be used to return to data mode.

Note:

- When iWRAP enters to command mode **READY** event occurs

  (Unless masked away with **SET CONTROL ECHO**.)

- Escape character can be changed with 'SET CONTROL ESCAPE' –command.
- DTR mode can be enabled with 'SET CONTROL ESCAPE' –command.

3.1 Command Mode

Command mode is default mode when iWRAP is powered up. In command mode ASCII commands can be entered to iWRAP to perform various functions.

Note:

- Incoming data from remote devices is buffered when iWRAP is in command mode.
- Because of embedded nature of iWRAP buffering capabilities are low and only small amounts of data can be received to buffers. The amount of data which can be buffered depends on the firmware version and the state of iWRAP. Usually it is around 2 Kbytes, but may vary radically.

3.2 Data mode

Data mode is default mode when there are one or more connections. In data mode all data is sent transparently from UART over the Bluetooth RFCOMM link to other device and vice versa.

Initial mode	Target mode	Requirements for transition from initial mode to target mode
Command Mode (no active connection) In this mode ASCII command can be given to iWRAP.	Data Mode	Connection is successfully created using **CALL** command. (**CONNECT** event is used to notify a successful link creation). Remote device has connected us. (**RING** event is used to notify of incoming connections.)
Data Mode In this mode all data can be sent transparently from RS-232 over the Bluetooth RFCOMM link to the other device.	Command Mode	User switches mode using escape sequence <1s>**+++**<1s> or by setting DTR low. Link is terminated (closed by remote device or link loss). (**NO CARRIER** event is used to inform of link termination.)
Command Mode (active connection) In this mode ASCII command can be given to iWRAP.	Data Mode	User switches mode either using escape sequence <1s>**+++**<1s>, or using command **SELECT**.

Table 1: iWRAP modes and transitions

3.3 Multiplexing mode

In iWRAP 2.1.0 and newer there is a special mode called 'multiplexing mode'. In this mode iWRAP does not have separate command or data modes, but data, commands and events are all handled in one single mode. There is however a special protocol to separate commands and events from the actual data.

The advantage of this multiplexing mode is that several Bluetooth connections can be handled simultaneously and there is no need to do time consuming data-command-data mode switching.

To learn more about multiplexing mode, please see the description of "SET BT MUX".

4. TECHNICAL DETAILS

Feature:	Value:
MAX simultaneous ACL connections	4
MAX simultaneous SCO connections	1
MAX data rate	650Kbps (WT12/WT11 to BT2.0 USB-dongle) 570Kbps (WT12/WT11 to WT12/WT11) 450Kbps (WT12/WT11 to BT1.1-BT1.2 device)
MAX UART baud rate	921600 bps
MIN transmission delay	8-15ms
PIN code length	Configurable from 0 to 16 characters
Encryption length	Configurable from 0 to 128 bits
MAX simultaneous pairings	16
MAX Friendly name length	Configurable up to 256 characters
Packet size	Configurable from 21 to 1008

Table 2: Technical details

5. USAGE

iWRAP can be used and controlled from the host system by sending ASCII commands through UART interface.

When installed and configured the module can be commanded from the host with the following ASCII commands:

- BER
- Call
- CLose
- HELP
- Inquiry
- List
- Name
- RSsi
- RESET
- SELect
- SET
- TESTMODE
- TXPower
- HELP

Note:

These commands should end with line feed **'\r\n'** character.

5.1 Typographical conventions

The ASCII commands and their usage are described further in this chapter. Commands and output synopsis are presented as follows:

Synopsis:
COMMAND {*required parameter*} [*optional parameter*] STATIC TEXT [*2ND OPTIONAL PARAMETER*]

Command parameters on the other hand are described like this:

Description:	
parameter	Description

Responses to the command are described as in the table below:

Response:	
RESPONSE {*parameters*}	
parameter	Description

Events generated by commands or actions are described as follow:

Events:	
EVENT	Description

And finally examples shown are described like this:

EXAMPLE COMMAND RESPONSE TO COMMAND

NOTE!

- The parser is not case sensitive!
- ASCII interface 0.0.2 does not accept backspaces, but version 2.0.0 and later do.

5.2 CALL

CALL command is used to initiate connections to the remote device. Connections are closed using command **CLOSE**. Currently open connections can be viewed using command **LIST**.

5.2.1 Syntax

Synopsis:
CALL {*address*} {*target*} RFCOMM [*MTU* {*packet size*}]

Description:	
address	Bluetooth address of the remote device
target	RFCOMM target for the connection. Target may be one of the following: **channel** RFCOMM channel number Format: xx (hex) **uuid16** 16 bit UUID for searching channel Format: xxxx (hex) **uuid32** 32 bit UUID for searching channel Format: xxxxxxxx (hex) **uuid128** 128 bit UUID for searching channel Format: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx (hex)
MTU	Optional static text to indicate that packet size parameter is in use.
packet size	Packet size to use (Values from 21 to 1008 can be used).

Response:	
CALL {*link_id*}	
link_id	Numeric connection identifier

Events:	
CONNECT	Delivered if **CALL** command is successful.
NO CARRIER	Delivered if **CALL** command fails.

5.2.2 Examples

Creating successful connection to 00:07:80:80:52:27 using channel 1.

```
CALL 00:07:80:80:52:27 1 RFCOMM
CALL 0
CONNECT 0 RFCOMM 1
```

Creating successful connection to 00:07:80:80:52:27 using Serial Port Profile.

(UUID16 SPP = 1101)

```
CALL 00:07:80:80:52:27 1101 RFCOMM
CALL 0
CONNECT 0 RFCOMM 1
```

Unsuccessful connection attempt to 00:07:80:80:52:26.

```
CALL 00:07:80:80:52:26 1 RFCOMM
CALL 0
NO CARRIER 0 ERROR 406 RFC_CONNECTION_FAILED
```

Creating successful connection to 00:07:80:80:52:27 with MTU 600.

```
CALL 00:07:80:80:52:27 1101 RFCOMM MTU 600
CALL 0
CONNECT 0 RFCOMM 1
```

NOTE!

If CALL is used with CHANNEL instead of UUID it will be on average around 300ms faster, since there is no need to do service discovery. However the channel of serial port profile (SPP) must be known. Notice that the channel for a specific service may vary between different *Bluetooth* devices.

In iWRAP the channel for SPP is always 1 however.

5.3 CLOSE

Command **CLOSE** is used to terminate previously opened connection.

5.3.1 Syntax

Synopsis:
CLOSE {*link_id*}

Description:	
link_id	Numeric connection identifier from previously used command **CALL** or from event **RING**.

Response:
No response

Events:	
NO CARRIER	This event is delivered after link is closed.

5.3.2 Examples

Closing an active connection.

```
CALL 00:60:57:a6:56:49 1103 RFC
CALL 0
CONNECT 0 RFCOMM 1

READY.
CLOSE 0
NO CARRIER 0 ERROR 0
```

5.4 INQUIRY

Command **INQUIRY** is used to find other Bluetooth devices in the area (making device discovery).

5.4.1 Syntax

Synopsis:
INQUIRY {*timeout*} [*NAME*] [*LAP* {*lap*}]

Description:	
timeout	The maximum amount of time (in units of 1.28 seconds) before the inquiry process is halted.
NAME	Optional flag to automatically request friendly name for found devices. See command **NAME** for more information about remote name request.
LAP	Optional flag for specifying that inquiry access code is in use.
lap	Value for inquiry access code. Can have following values: **0x9e8b33** General/Unlimited Inquiry Access Code (GIAC). This is the default value. **0x9e8b00** Limited Dedicated Inquiry Access Code (LIAC). **0x9e8b01-0x9e8b32-0x9e8b34-0x9e8b3f** Reserved for furure use.

Response:	
INQUIRY {*num_of_devices*} **INQUIRY {*addr*} {*class_of_device*}**	
num_of_devices	Amount of found devices
addr	Bluetooth address of a found device
class_of_device	Bluetooth Class of Device of a found device

Events:	
INQUIRY_PARTIAL	These events are delivered as devices are found.
NAME	These events are delivered after **INQUIRY** if ***NAME*** flag is present.

NOTE!

It may take up to 10.24 seconds for Bluetooth device to answer inquiry scan and thus timeout value should be at least 8 if it is necessary to find every device in the area.

*) iWRAP 2.1.0 and later support RSSI in the inquiry but this feature must be enabled with '**SET CONTROL CONFIG'** –command.

If set '**BT BT LAP'** is in use there is no need to use ***[LAP {lap}]*** in the **INQUIRY** any more.

INQUIRY_PARTIAL events can be masked off with '**SET CONTROL ECHO'** command.

5.4.2 Examples

Basic INQUIRY command:

```
INQUIRY 1
INQUIRY_PARTIAL 00:14:a4:8b:76:9e 72010c
INQUIRY_PARTIAL 00:10:c6:62:bb:9b 1e010c
INQUIRY_PARTIAL 00:10:c6:4d:62:5c 72010c
INQUIRY_PARTIAL 00:10:c6:3a:d8:b7 72010c
INQUIRY_PARTIAL 00:02:ee:d1:80:6d 520204
INQUIRY_PARTIAL 00:10:c6:62:bb:fa 1c010c
INQUIRY 6
INQUIRY 00:14:a4:8b:76:9e 72010c
INQUIRY 00:10:c6:62:bb:9b 1e010c
INQUIRY 00:10:c6:4d:62:5c 72010c
INQUIRY 00:10:c6:3a:d8:b7 72010c
INQUIRY 00:02:ee:d1:80:6d 520204
INQUIRY 00:10:c6:62:bb:fa 1c010c
```

INQUIRY command with NAME resolution:

```
INQUIRY 1 NAME
INQUIRY_PARTIAL 00:10:c6:3a:d8:b7 72010c
INQUIRY_PARTIAL 00:10:c6:62:bb:9b 1e010c
INQUIRY_PARTIAL 00:14:a4:8b:76:9e 72010c
INQUIRY 3
INQUIRY 00:10:c6:3a:d8:b7 72010c
INQUIRY 00:10:c6:62:bb:9b 1e010c
INQUIRY 00:14:a4:8b:76:9e 72010c
NAME 00:10:c6:3a:d8:b7 "TOM"
NAME 00:10:c6:62:bb:9b "CSLTJANI"
NAME 00:14:a4:8b:76:9e "SWLTMIKKO_3"
```

INQUIRY command with LAP in use:

```
INQUIRY 3 LAP 9e8b11
INQUIRY_PARTIAL 00:07:80:80:52:15 111111
INQUIRY_PARTIAL 00:07:80:80:52:27 111111
INQUIRY 2
INQUIRY 00:07:80:80:52:15 111111
INQUIRY 00:07:80:80:52:27 111111
```

INQUIRY command with RSSI enabled:

```
INQUIRY 1
INQUIRY_PARTIAL 00:10:c6:62:bb:9b 1e010c "" -71
INQUIRY_PARTIAL 00:10:c6:4d:62:5c 72010c "" -73
INQUIRY_PARTIAL 00:10:c6:3a:d8:b7 72010c "" -73
INQUIRY 5
INQUIRY 00:10:c6:62:bb:9b 1e010c
INQUIRY 00:10:c6:4d:62:5c 72010c
INQUIRY 00:10:c6:3a:d8:b7 72010c
```

5.5 IC

IC command can be used to stop on-going **INQUIRY**.

5.5.1 Syntax

Synopsis:
IC

Description:
No Description

Response:	
INQUIRY {*num_of_devices*} **INQUIRY {*addr*} {*class_of_device*}**	
num_of_devices	Amount of found devices
addr	Bluetooth address of a found device
class_of_device	Bluetooth Class of Device of a found device

Events:	
INQUIRY_PARTIAL	These events are delivered as devices are found.
NAME	These events are delivered after **INQUIRY** if ***NAME*** flag is present.

Response:	
INQUIRY {*num_of_devices*} **INQUIRY {*addr*} {*class_of_device*}** **INQUIRY {*addr*} {*class_of_device*} {*rssi*}***	
num_of_devices	Amount of found devices
addr	Bluetooth address of a found device
class_of_device	Bluetooth Class of Device of a found device
rssi	RSSI value of a found device (0 = good signal, -127 = poor signal)

Events:	
INQUIRY_PARTIAL	These events are delivered as devices are found.
NAME	These events are delivered after **INQUIRY** if ***NAME*** flag is present.

NOTE!

It may take up to 10.24 seconds for Bluetooth device to answer inquiry scan and thus timeout value should be at least 8 if it is necessary to find every device in the area.

*) iWRAP 2.1.0 and later support RSSI in the inquiry but this feature must be enabled with '**SET CONTROL CONFIG'** –command.

If set '**BT BT LAP'** is in use there is no need to use ***[LAP {lap}]*** in the **INQUIRY** any more.

INQUIRY_PARTIAL events can be masked off with '**SET CONTROL ECHO'** command.

5.6 LIST

Command **LIST** shows information about connections currently open.

Synopsis:
LIST

Response:	
LIST {num_of_links} **LIST {link_id} CONNECTED RFCOMM {blocksize} 0 0 {elapsed_time} {local_msc} {remote_msc} {addr} {channel} {direction} {powermode} {role} {crypt}***	
num_of_links	Number of currently open links
link_id	Numeric connection identifier
blocksize	Data packet size, ie. how many bytes data can be sent in one packet
elapsed_time	Link life time in seconds
local_msc & remote_msc	Serial port status bits, "8d" is normal value
addr	Bluetooth device address of the remote device
channel	RFCOMM channel number at remote device
direction	Direction of the link **"OUTGOING"** Link is initiated by local device (using command **CALL**) **"INCOMING"** Link is initiated by the remote device
powermode	Power mode for the link **"ACTIVE"** Link is in active mode

	"SNIFF" Link is in sniff mode **"HOLD"** Link is in hold mode **"PARK"** Link is in park mode
role	Role of the link **"MASTER"** iWRAP is the master device of this link **"SLAVE"** iWRAP is the slave device of this link
crypt	Encryption state of the link **"PLAIN"** Link is not encrypted **"ENCRYPTED"** Link is encrypted

Events:
No response

5.6.1 Examples

```
LIST
LIST 2
LIST 0 CONNECTED RFCOMM 669 0 0 40 8d 8d 00:07:80:80:31:e6 1
INCOMING SNIFF SLAVE ENCRYPTED
LIST 1 CONNECTED RFCOMM 669 0 0 18 8d 8d 00:07:80:80:32:0e 1
OUTGOING ACTIVE MASTER ENCRYPTED
```

5.7 NAME

Command **NAME** is used retrieve friendly name of the device.

Synopsis:
NAME {address}

Description:	
address	Address of the Bluetooth device

Response:
No response

Events:
NAME event is delivered when friendly name is known. **NAME ERROR** event is delivered if friendly name lookup fails.

5.7.1 Examples

Successful name query:

```
NAME 00:07:80:bf:bf:01
NAME 00:07:80:bf:bf:01 "AI bf:01"
```

Unsuccessfull name query:

```
NAME 00:07:80:bf:bf:bf
NAME ERROR 104 00:07:80:bf:bf:bf HCI_ERROR_PAGE_TIMEOUT
```

5.8 RESET

Command **RESET** is used to reset iWRAP.

Synopsis:
RESET

Response:
No response

Events:
None

5.9 SELECT

Command **SELECT** is used to switch to data mode.

Synopsis:
SELECT {link_id}

Description:	
link_id	Numeric connection identifier

Response:
No response. iWRAP goes to data mode with the link **link_id**.

Events:
None

5.9.1 Examples

Changing between links

```
LIST

LIST 2
LIST 0 CONNECTED RFCOMM 668 0 0 243 8d 8d 00:07:80:80:38:77 1
OUTGOING ACTIVE MASTER ENCRYPTED
LIST 1 CONNECTED RFCOMM 668 0 0 419 8d 8d 00:07:80:80:36:85 1
OUTGOING ACTIVE MASTER ENCRYPTED
SELECT 1     (Transition to DATA mode - Device:
00:07:80:80:36:85)
```

5.10 SET

SET displays or sets configuration values of iWRAP.

Synopsis:
SET [{category} {option} {value}]

Description:	
Without any parameters **SET** displays current configuration.	
category	Category of setting **BT** Changes different Bluetooth related settings. See **SET BT** for more information about options. **CONTROL** Changes different iWRAP settings. See **SET CONTROL** for more information about options. **{link_id}** This command is used to control the various settings related to Bluetooth links in iWRAP. These are for example master, slave and power save modes (SNIFF, PARK, and ACTIVE).
option	Option name, depends on category. See following sections for more information.
value	Value for option. See following sections for more information.

Response:

If issued without parameters:

```
SET {category} {option} [value]*
```

```
SET
```

If issued with parameters:

None.

Events:

None

5.11 SET BT

Under this section all commands related to 'SET BT' are described.

5.11.1 SET BT BDADDR

Shows the local devices Bluetooth address.

SET BT BDADDR
List format: **SET BT BDADDR {*addr*}** ***addr*** Bluetooth device address of local device **Note** This value is read-only.

5.11.2 SET BT NAME

Shows or sets the local devices Friendly name.

SET BT NAME
List format: **SET BT NAME {*friendly_name*}** ***friendly_name*** Friendly name of local device Set format: **SET BT NAME [*friendly_name*]** ***friendly_name*** Friendly name of local device **Warning** If ***friendly_name*** is left empty some device may have problems showing device.

Note:

Maximum length of friendly name is 16 characters in iWRAP 2.0.2 and older. In newer versions the maximum length is 256 characters.

5.11.3 SET BT CLASS

Shows or sets the local devices Class-of-Device (CoD).

SET BT CLASS

List format :

SET BT CLASS {*class_of_device*}

Set format :

SET BT CLASS {*class_of_device*}

class_of_device

Bluetooth Class of Device of local device

5.11.4 SET BT AUTH

Shows or sets the local devices PIN code.

SET BT AUTH
List format: **SET BT AUTH * {*pin_code*}** Set format: **SET BT AUTH * [*pin_code*]** ***pin_code*** Pin code for authorized connections. Authorization is required if this option is present. **Note:** If command '***SET BT AUTH ****' is given, PIN code will be disabled and no encryption will be used. If '***SET BT AUTH'*** is not visible when '***SET***' command is given, PIN code is disabled.

5.11.5 SET BT PAIR

Shows or sets the local devices Pairing information.

SET BT PAIR

List format:

SET BT PAIR {*addr*} {*link_key*}

Set format:

SET BT PAIR {*addr*} [*link_key*]

addr

Bluetooth device address of the paired device

link_key

Link key for authenticated connection

To remove device from list of known devices left link_key parameter empty.

Tip

To remove every paired device use * as ***addr***

'SET BT PAIR *'

Note

SET BT PAIR is not visible if there are no paired devices.

5.11.6 SET BT PAGEMODE

Shows or sets the local devices Page mode. With page mode visibility if iWRAP can be controlled in different modes. Also connection related timeouts can be configured.

SET BT PAGEMODE

List format

SET BT PAGEMODE {*page_mode*} {*page_timeout*} {*page_scan_mode*}

Set format

SET BT PAGEMODE {*page_mode*} [*page_timeout*] [*page_scan_mode*]

page_mode

0

iWRAP is NOT visible in the inquiry and does NOT answers calls

1

iWRAP IS visible in the inquiry but does NOT answers calls

2

iWRAP IS NOT visible in the inquiry but answers calls

3

iWRAP IS visible in the inquiry and answers calls (Default mode)

4

Just like mode 3 if there are NO connections. If there are connections it's like mode 0

page_timeout

Page timeout defines how long connection establishment can take before an error occurs. Page timeout is denoted as a hexadecimal number (HEX) and calculated as in the example below:

2000 (HEX) is 8192 (DEC). Multiply it by 0.625 and you'll get the page timeout in milliseconds. In this case it's 5120 ms (8192 * 0,625ms).

So connection establishment can take around 5 seconds until you'll receive an error.

page_scan_mode

0

Is mode R0 and it means that iWRAP IS connectable all the time, but NOT visible in inquiry.

1	Is mode R1 and it means that iWRAP is connectable every 1.28 sec (Default value)
2	Is mode R2 and it means device is connectable every 2.56 sec (Lowest power consumption)

5.11.7 SET BT ROLE

Shows or sets the local devices Role mode. With Role command iWRAPs master-slave behaviour can be altered. Also supervision timeout and link policy can be configured.

SET BT ROLE

List format

SET BT ROLE {*ms_policy*} {*link_policy*} {*supervision_timeout*}

Set format

SET BT ROLE {*ms_policy*} [*link_policy*] [*supervision_timeout*]

ms_policy

This value defines how Master-Slave policy works. Values are described below:

0

Allow master-slave switch when calling but do not request it when answering. (Default value)

1

Allow master-slave switch when calling, request it when answering.

2

Don't allow master-slave switch when calling, request it when answering.

link_policy

This bitmap controls the link policy mode. Bits are described below:

0

This value disables all link policy modes.

1

This value enables role switch

2

This value enables Hold mode

3

This value enables Sniff mode

4

This value enables Park state

F

This value enables all of the above modes (Default)

supervision_timeout

Supervision timeout controls how long a Bluetooth link is kept open if remote end does not answer. Supervision timeout is denoted as a hexadecimal number (HEX) and calculated as in the example below:

12C0 (HEX) is 4800 (DEC). Multiply it by 0.625 and you'll get the supervision timeout in milliseconds. In this case it's 3000 ms (4800 * 0,625ms).

So remote end can be silent for around 3 seconds until connection is closed.

The default value is 7d00 (20 seconds).

5.12 SET CONTROL

Under this section all commands related to 'SET CONTROL' are described.

5.12.1 SET CONTROL AUTOCALL

Enables or disables the autocall functionality in iWRAP (SPP master like).

When autocall feature is enabled iWRAP tries to call a **paired** device continously until a connection is established. If connection is lost or closed same sequence starts again.

If there are several paired devices in iWRAP, an inquiry (not visible to the user) is made and first paired device found is connected.

SET CONTROL ECHO
List format: **`SET CONTROL AUTOCALL [`*`target`*`]`** Set format: **`SET CONTROL AUTOCALL [`*`target`*`]`** ***`target`*** RFCOMM target for automatic connection. Target may be: ***`channel`*** RFCOMM channel number. Format xxxx (HEX) ***`uuid16`*** 16 bit UUID for searching the channel. Format xxxx (HEX) ***`uuid32`*** 32 bit UUID for searching the channel. Format xxxxxxxx (HEX) ***`Uuid128`*** 128 bit UUID for searching the channel. Format xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx (HEX)

5.12.1.1 Examples

Enabling and disabling autocall

```
"Enable Autocall for channel 1"
SET CONTROL AUTOCALL 1
"Or for SPP"
SET CONTROL AUTOCALL 1101
"Or for DUN"
SET CONTROL AUTOCALL 1103
"Or for Audio Gateway"
SET CONTROL AUTOCALL 111F
"Disabling autocall"
SET CONTROL AUTOCALL
```

5.12.2 SET CONTROL BAUD

Changes the UART related settings.

SET CONTROL BAUD

List format:

SET CONTROL BAUD {*baud_rate*},8{*parity*}{*stop_bits*}

Set format:

SET CONTROL BAUD {*baud_rate*},8 {*parity*} {*stop_bits*}

baud_rate

UART baud rate in bps

",8"

Static string indicating UART uses 8 data bits

parity

UART parity setting

"n"

None parity

"e"

Even parity

"o"

Odd parity

stop_bits

Number of stop bits in UART communications

"1"

One stop bit

"2"

Two stop bits

Important

Parameters in **SET CONTROL BAUD** must be typed together (no spaces

between)!

See data sheet for supported baud rates.

5.12.3 SET CONTROL CD

Enables or disables the Carrier Detect (CD) mode in iWRAP.

SET CONTROL CD

List format:

```
SET CONTROL CD {cd_mask} [datamode]
```

Set format:

```
SET CONTROL CD {cd_mask} [datamode]
```

cd_mask

This is bit mask for selection the GPIO pins used for CD signal.

For example for PIO2 bit mask is **00000100** and **cd_mask** is **4** (HEX).

datamode

0

CD signal is driven high if there are any connections.

1

CD signal is driven high only in data mode.

Note:

Only pins form 2 to 7 are available.

Notice that CD and DTR signals can not be configured to use same IO.

5.12.4 SET CONTROL CONFIG

This command enables or disables various functional features in iWRAP. These features are described below.

SET CONTROL CONFIG

List format

SET CONTROL CONFIG {*configuration_value*}

Set format

SET CONTROL CONFIG {*configuration_value*}

configuration_value

This value is a bitfield (represented as a hexadecimal value) which is used to control various features in iWRAP. These features are desribed below:

Bit:

0 *"RSSI visible in the inquiry"*

If this bit is set RSSI value will be visible in the inquiry results

1 *"Not used"*

Must be set to 0.

2 *"Interlaced inquiry scan"*

If this bit is set interlaced inquiry will be used. Generally interlaced inquiry is a little bit faster than regular inquiry.

3 *"Interlaced page scan"*

If this bit is set interlaced page (call) will be used. Generally interlaced page is a little bit faster than regular page.

4 *"'Deep sleep' enabled"*

If this bit is set 'Deep sleep' power saving mode will be used. Deep sleep is an aggressive power saving mode used when there are no connections.

5 *"Bluetooth address in CONNECT"*

If this bit is set Bluetooth address of remote end will be displayed on CONNECT event.

Note:

Default configuration: 'SET CONTROL CONFIG 0'

5.12.5 SET CONTROL ECHO

Lists or changes the Echo-modes in iWRAP.

SET CONTROL ECHO

List format:

SET CONTROL ECHO {*echo_mask*}

Set format:

SET CONTROL ECHO [*echo_mask*]

echo_mask

Bit mask for controlling echo and events displaying

Bit 0

If this bit is set start-up banner is visible

Bit 1

If this bit is set characters are echoed back to client in command mode

Bit 2

If set events are displayed when in command mode

Default value for **SET CONTROL ECHO** is 7 (bits 0, 1 and 2 set).

Warning!

If every bit is set off (value 0) it is quite impossible to know the status of iWRAP.

If Bit 2 is set off it is very hard to detect whether iWRAP is in command mode or in data mode. This can however be solved if one IO is used to induicate that iWRAP is in data mode (CD signal).

5.12.6 SET CONTROL ESCAPE

This command can be used to change the escape charater used to change between command and data mode. This command also enables and disables DTR signalling.

SET CONTROL CONFIG

List format

SET CONTROL ESCAPE {*esc_char*} [*dtr_mask*] [*dtr_mode*]

Set format

SET CONTROL ESCAPE {*esc_char*} [*dtr_mask*] [*dtr_mode*]

esc_char

Decimal ASCII character to define escape character used in escape sequence. Use hyphen '-' to disable escape sequence.

dtr_mask

Bit mask for selecting I/O pins used for DTR.

For example for IO5 bit mask is **00100000** and **dtr_mask** is **20** (HEX).

dtr_mode

0

Disabled

1

Return to 'command' mode when DTR is dropped

2

Close active connection when DTR is dropped

3

Reset iWRAP when DTR is dropped

Note:

Only I/O pins form 2 to 7 are available for DTR.

Notice that CD and DTR signals can not be configured to use same IO.

5.12.7 SET CONTROL INIT

Lists or changes the Initialization command in iWRAP. This command is run when iWRAP is started or rebooted.

SET CONTROL INIT
List format **SET CONTROL INIT {*command*}** Set format **SET CONTROL INIT [*command*]** ***command*** Any iWRAP command string. This command is automatically executed every time iWRAP starts (after power-on, **RESET** or watchdog event) **Tip:** Type '**SET CONTROL INIT SET BT PAIR ***' to remove all pairings on reboot.

5.13 SET {link_id}

Under this section all commands related to 'SET {link_id}' are described.

This section for example covers the Power saving mode settings and therefore below is a brief explanation to Bluetooth power saving modes supported by iWRAP:

SNIFF mode:

Once a Bluetooth device is connected to a Piconet, it can enter one of three power saving modes. In SNIFF mode the activity of a Bluetooth devise is lowered, enabling it to listen at a reduced rate to the Piconet. The interval or period between SNIFF is configurable.

SNIFF mode is the least power efficient of all three power saving modes.

PARK state:

The Park state can be used when a Bluetooth device is connected to the Piconet but does not participate in traffic transfer.

The Park mode conserves the most power compared with other power saving modes.

General information about power saving:

On the SNIFF mode and on the PARK state, the devices have a reduced participation on the traffic of messages and packets. On the SNIFF mode this occurs only at 'SNIFF intervals' and at the PARK state at the beacons (at this mode the device also listens to broadcast messages).

The main advantage for using PARK mode over SNIFF mode is that it leads to reduced power consumption and gives more time for the parked slave to participate on different Piconet(s).

5.13.1 SET {link_id} ACTIVE

This command disables all the power save modes for the defined, active Bluetooth link and sets it into an active mode.

SET {link_id} ACTIVE
List format **None.** Set format **SET {link_ID} ACTIVE** ***link_ID*** Numeric connection identifier

5.13.2 SET {link_id} MASTER

Switches link to Piconet master.

SET {link_id} MASTER
List format **None.** Set format **SET {link_ID} MASTER** ***link_ID*** Numeric connection identifier

5.13.3 SET {link_id} PARK

This command enables the PARK mode for the defined Bluetooth link.

SET {link_id} PARK

List format

None

Set format

SET {link_id} PARK {max} {min}

SET {link_id} PARK {avg}

link_id

Numeric connection identifier

max

Maximum acceptaple interval

min

Maximum acceptaple interval

avg

Shortcut for easier sniff setting

Note:

Refer to Bluetooth specification for more information.

When iWRAP is in park state it's power saving state can not be changed from the remote end.

5.13.4 SET {link_id} SLAVE

Switch to Piconet slave.

SET {link_id} SLAVE
List format **None** Set format **SET {link_id} SLAVE** ***link_id*** Numeric connection identifier

5.13.5 SET {link_id} SNIFF

This command enables the SNIFF mode for the defined, active Bluetooth link.

SET {link_id} SNIFF

List format

None.

Set format

SET {*link_ID*} SNIFF {max} {min} [{attempt} {timeout}]

link_id

Numeric connection identifier

max

Maximum acceptaple interval

min

Maximum acceptaple interval

avg

Shortcut for easier sniff setting

attempt

Number of SNIFF attempts. Default value 1.

timeout

Number of SNIFF timeout. Default value 8.

Note:

Refer to Bluetooth specification for more information.

When iWRAP is in sniff mode it's power saving state can not be changed from the remote end.

5.14 TESTMODE

Command **TESTMODE** enables Bluetooth Test Mode in which Bluetooth Testers may be used to test radio environment.

Synopsis:
TESTMODE

Response:
TEST 0

Events:
None

6. IWRAP EVENTS

Events are mechanism that iWRAP uses to notify the User for completed commands, incoming connections, etc. If iWRAP is in data mode only possible event is **NO CARRIER** event for corresponding link.

Note:

- iWRAP is designed so that unwanted events can be safely ignored. Events **CONNECT**, **NO CARRIER** and **RING** change the mode of operation and therefore they cannot be ignored.
- Events may be masked away by removing Bit 2 on command **SET CONTROL ECHO**.

6.1 CONNECT

CONNECT event is used to notify for successful link establishment.

Synopsis:
`CONNECT {link_id} RFCOMM {channel} [address]`

Description:	
`link_id`	Numeric connection identifier
`channel`	Connected RFCOMM channel number
`address`	Address of the remote end

See also: **CALL**, **LIST**

Note:

iWRAP automatically goes into data mode after CONNECT event.

6.2 INQUIRY_PARTIAL

INQUIRY_PARTIAL event is used to notify found Bluetooth device. This event precedes response for **INQUIRY** command.

Synopsis:
`INQUIRY_PARTIAL {address} {class_of_device} [{cached_name} {rssi}]`

Description:	
`address`	Bluetooth device address of found device
`class_of_device`	Bluetooth Class of Device of found device
`cached_name`	User friendly name of found device if already known
`rssi`	Received Signal Strength of found device in dBm

Note:

- **cached_name** and **rssi** are only visible if "Inquiry with RSSI" is enabled with **SET CONTROL CONFIG**

See also: **INQUIRY**

6.3 NO CARRIER

NO CARRIER event is used to notify for link loss or alternatively failure in link establishment.

Synopsis:
NO CARRIER {link_id} RFCOMM {error_code} [*message*]

Description:	
link_id	Numeric connection identifier
error_code	Code describing error
message	Optional verbose error message

See also: **CALL, CLOSE, LIST, RING**

READY

READY event is used to notify for switching to command mode.

Synopsis:
READY.

6.4 NAME

NAME event is used to notify for successful lookup for Bluetooth friendly name of the remote device.

Synopsis:
NAME {address} {"friendly_name"}

Description:	
address	Bluetooth device address of the device.
friendly_name	Friendly name of the device.

See also: **INQUIRY, NAME**

6.5 NAME ERROR

NAME ERROR event is used to notify for Bluetooth friendly name lookup failure.

Synopsis:
NAME ERROR {error_code} {address} [*message*]

Description:	
error_code	Code describing error
address	Bluetooth device address of the device
message	Optional verbose error message

See also: **INQUIRY, NAME**

6.6 RING

RING event is used to notify for incoming connection. Incoming connections are accepted only if there is no existing links.

Synopsis:
RING {link_id} {address} {channel} RFCOMM

Description:	
link_id	Numeric connection identifier
address	Bluetooth device address of the device
channel	Local RFCOMM channel

See also: **INQUIRY, NAME**

6.7 SYNTAX ERROR

SYNTAX ERROR is not an actual event but error message describing faulty typed command or error in command parameters.

Synopsis:
`SYNTAX ERROR`

7. IWRAP ERROR MESSAGES

This chapter briefly presents the iWRAP's error messages.

7.1 HCI errors

HCI errors start with code: ***0x100***

ERROR MESSAGE	CODE
HCI_SUCCESS	0x00
HCI_ERROR_ILLEGAL_COMMAND	0x01
HCI_ERROR_NO_CONNECTION	0x02
HCI_ERROR_HARDWARE_FAIL	0x03
HCI_ERROR_PAGE_TIMEOUT	0x04
HCI_ERROR_AUTH_FAIL	0x05
HCI_ERROR_KEY_MISSING	0x06
HCI_ERROR_MEMORY_FULL	0x07
HCI_ERROR_CONN_TIMEOUT	0x08
HCI_ERROR_MAX_NR_OF_CONNS	0x09
HCI_ERROR_MAX_NR_OF_SCO	0x0a
HCI_ERROR_MAX_NR_OF_ACL	0x0b
HCI_ERROR_COMMAND_DISALLOWED	0x0c
HCI_ERROR_REJ_BY_REMOTE_NO_RES	0x0d
HCI_ERROR_REJ_BY_REMOTE_SEC	0x0e
HCI_ERROR_REJ_BY_REMOTE_PERS	0x0f
HCI_ERROR_HOST_TIMEOUT	0x10
HCI_ERROR_UNSUPPORTED_FEATURE	0x11
HCI_ERROR_ILLEGAL_FORMAT	0x12

HCI_ERROR_OETC_USER	0x13
HCI_ERROR_OETC_LOW_RESOURCE	0x14
HCI_ERROR_OETC_POWERING_OFF	0x15
HCI_ERROR_CONN_TERM_LOCAL_HOST	0x16
HCI_ERROR_AUTH_REPEATED	0x17
HCI_ERROR_PAIRING_NOT_ALLOWED	0x18
HCI_ERROR_UNKNOWN_LMP_PDU	0x19
HCI_ERROR_UNSUPPORTED_REM_FEATURE	0x1a
HCI_ERROR_SCO_OFFSET_REJECTED	0x1b
HCI_ERROR_SCO_INTERVAL_REJECTED	0x1c
HCI_ERROR_SCO_AIR_MODE_REJECTED	0x1d
HCI_ERROR_INVALID_LMP_PARAMETERS	0x1e
HCI_ERROR_UNSPECIFIED	0x1f
HCI_ERROR_UNSUPP_LMP_PARAM	0x20
HCI_ERROR_ROLE_CHANGE_NOT_ALLOWED	0x21
HCI_ERROR_LMP_RESPONSE_TIMEOUT	0x22
HCI_ERROR_LMP_TRANSACTION_COLLISION	0x23
HCI_ERROR_LMP_PDU_NOT_ALLOWED	0x24
HCI_ERROR_ENC_MODE_NOT_ACCEPTABLE	0x25
HCI_ERROR_UNIT_KEY_USED	0x26
HCI_ERROR_QOS_NOT_SUPPORTED	0x27
HCI_ERROR_INSTANT_PASSED	0x28
HCI_ERROR_PAIR_UNIT_KEY_NO_SUPPORT	0x29

HCI_ERROR_CHANNEL_CLASS_NO_SUPPORT	0x2e

Table 3: HCI errors

7.2 L2CAP errors

L2CAP errors start with code: ***0x200***

7.3 SDP errors

SDP errors start with code: ***0x300***

ERROR MESSAGE	CODE
SDC_OK	0x00
SDC_OPEN_SEARCH_BUSY	0x01
SDC_OPEN_SEARCH_FAILED	0x02
SDC_OPEN_SEARCH_OPEN	0x03
SDC_OPEN_DISCONNECTED	0x04
SDC_NO_RESPONSE_DATA	0x11
SDC_ERROR_RESPONSE_PDU	0x10
SDC_CON_DISCONNECTED	0x12
SDC_CONNECTION_ERROR	0x13
SDC_CONFIGURE_ERROR	0x14
SDC_SEARCH_DATA_ERROR	0x15
SDC_DATA_CFM_ERROR	0x16
SDC_SEARCH_BUSY	0x17
SDC_RESPONSE_PDU_HEADER_ERROR	0x18
SDC_RESPONSE_PDU_SIZE_ERROR	0x19
SDC_RESPONSE_TIMEOUT_ERROR	0x1a
SDC_SEARCH_SIZE_TOO_BIG	0x1b
SDC_RESPONSE_OUT_OF_MEMORY	0x1c
SDC_RESPONSE_TERMINATED	0x1d

Table 4: SDP errors

7.4 RFCOMM errors

RFCOMM errors start with code: ***0x400***

ERROR MESSAGE	CODE
RFC_OK	0x00
RFC_CONNECTION_PENDING	0x01
RFC_CONNECTION_REJ_PSM	0x02
RFC_CONNECTION_REJ_SECURITY	0x03
RFC_CONNECTION_REJ_RESOURCES	0x04
RFC_CONNECTION_REJ_NOT_READY	0x05
RFC_CONNECTION_FAILED	0x06
RFC_CONNECTION_TIMEOUT	0x07
RFC_NORMAL_DISCONNECT	0x08
RFC_ABNORMAL_DISCONNECT	0x09
RFC_CONFIG_UNACCEPTABLE	0x0a
RFC_CONFIG_REJECTED	0x0b
RFC_CONFIG_INVALID_CID	0x0c
RFC_CONFIG_UNKNOWN	0x0d
RFC_CONFIG_REJECTED_LOCALLY	0x0e
RFC_CONFIG_TIMEOUT	0x0f
RFC_REMOTE_REFUSAL	0x11
RFC_RACE_CONDITION_DETECTED	0x12
RFC_INSUFFICIENT_RESOURCES	0x13
RFC_CANNOT_CHANGE_FLOW_CONTROL_MECHANISM	0x14

RFC_DLC_ALREADY_EXISTS	0x15
RFC_DLC_REJ_SECURITY	0x16
RFC_GENERIC_REFUSAL	0x1f
RFC_UNEXPECTED_PRIMITIVE	0x20
RFC_INVALID_SERVER_CHANNEL	0x21
RFC_UNKNOWN_MUX_ID	0x22
RFC_LOCAL_ENTITY_TERMINATED_CONNECTION	0x23
RFC_UNKNOWN_PRIMITIVE	0x24
RFC_MAX_PAYLOAD_EXCEEDED	0x25
RFC_INCONSISTENT_PARAMETERS	0x26
RFC_INSUFFICIENT_CREDITS	0x27
RFC_CREDIT_FLOW_CONTROL_PROTOCOL_VIOLATION	0x28
RFC_RES_ACK_TIMEOUT	0x30

Table 5: RFCOMM errors

8. CHANGING PARAMETERS OVER RS232 WITH PSTOOL

With iWRAP 2.1.0 changing the parameters is quite simple because of the autoBCSP feature. The actions needed to access the parameters are the following:

1. Connect RS2323 between the WRAP THOR and your PC
2. Connect power to WRAP THOR
3. Open PSTool
4. Settings are: **BCSP**, **COMn** and **115200** (Use correct COM-port)*
5. Change the settings you want
6. Close PSTool and reset WRAP THOR. iWRAP is activated automatically.

NOTE:

*) Baud rate is not dependent on the SET CONTROL BAUD configuration, but it dependent on the PSKEY_UART_BAUD RATE configuration. By default this is 115200 bps.

AutoBCSP functionality works only if PSKEY_UART_BAUD and SET CONTROL BAUD have same values!

WARNING!

If you change the parameters described after this paragraph it is highly probable that iWRAP can not be used any more after reset. Also parameter access over RS232 interface will not probably be possible. After this only access over SPI interface will be possible.

The parameters are:

1. **PSKEY_UART_CONFIG_USR = 08A8** or **0080**

   08A8 = HW flow control enabled

   08A0 = HW flow control disabled

2. **PSKEY_HOST_INTERFACE = VM access to UART**
3. **PSKEY_VM_DISABLE** = **False**

9. USING BLUETEST OVER RS232

With iWRAP 2.1.0 changing using the BlueTest application is quite simple because of the autoBCSP feature. The actions needed to use BlueTest are the following:

7. Connect RS2323 between the WRAP THOR and your PC
8. Connect power to WRAP THOR
9. Open BlueTest
10. Settings are: **BCSP**, **COMn** and **115200** (Use correct COM-port)*
11. Use the test function you need
12. Close BlueTest and reset WRAP THOR. iWRAP is activated automatically.

NOTE:

*) Baud rate is not dependent on the SET CONTROL BAUD configuration, but it dependent on the PSKEY_UART_BAUD_RATE configuration. By default this is 115200 bps.

AutoBCSP functionality works only if PSKEY_UART_BAUD and SET CONTROL BAUD have same values!

10. CHANGING TO HCI

The new firmware version (version 18.2) is called '*Unified firmware*'. This means that it contains both HCI and RFCOMM firmware when old version had only RFCOMM. To change between these two modes is nowadays done with parameters and this chapter instructs how to do the switch.

To change from iWRAP to HCI, following parameters need to be changed:

1. PSKEY_VM_DISABLE Value: **TRUE**
2. PSKEY_ONCHIP_HCI_CLIENT Value: **0000**
3. PSKEY_HOST_INTERFACE
    - **UART link running BCSP**
    - **USB link**
    - **UART link running H4**
    - **UART link running H5**
    - **UART link running H4 with Deep sleep**

Only one of the PSKEY_HOST_INTERFACE values needs to be selected. Choose of course the one that will fit your design and application.

Also if you select BCSP, H4 or H5 options, please make sure that **PSKEY_UART_BAUDRATE** and **PSKEY_UART_CONFIG_xxxx** have suitable values.

xxxx = protocol selection (BCSP, H4 etc.)

Tip:

The switch between iWRAP and HCI can be easily done using the canned operations in PSTool. These can be found form '**Entry**'->'**Canned operations**' menu. For example if you want to convert iWRAP into HCI USB module you need to select the following canned operations:

- **@usb**
- **@vm_disable**
- **@hci_external**

Note: PSTool 1.21 is needed to switch the parameters mentioned in this chapter.

11. TROUBLESHOOTING

11.1 I get no response from the iWRAP?

Make sure your terminal settings are correct. Use ***PSTool*** to check the UART settings from the WRAP THOR Bluetooth module and make similar settings into your terminal software.

Check also your ECHO MODE settings. If you have set ECHO MODE to 0, you should not be able to see any responses.

11.2 I can connect only to two devices at a time?

Only two connections at a time are supported.

11.3 I changed 'UART Baud rate' key, but it didn't seem to work?

UART baud rate is stored now into user keys instead of '***UART baud rate'*** key. Delete '***User configuration data 26***' in order to return back to default settings ***115200,8n1***.

Notice also that if you change baud rate with SET CONTROL BAUD it does not affect the baud rate you need to use with PSTool, if you want to access parameters. This baud rate is defined by the ***'UART baud rate'*** key.

11.4 Data coming form the UART is corrupted

If you are using 'Deep sleep' the minimum baud rate that can be used is 19200. Lower baud rates will corrupt the data.

11.5 I'm missing characters when I type ASCII commands.

If deep sleep is used first character written to UART wakes the module from the 'Deep sleep' and that's why the character is lost. There are two ways to overcome this problem:

1. If you command iWRAP with a micro controller or processor add 'space' or 'line break' characters in front of every command.

2. In PSTool set parameter 'EXIT deep sleep on CTS line activity' to TRUE. Now 'Deep sleep' does not lose characters any more, but power consumption will be increased a little. **This is the default setting.**

12. KNOWN ISSUES

1. Only two connections at a time are supported.

2. You can not form two connections from one iWRAP into one device. This is a property of CSR's *Bluetooth* stack.

3. Giving ASCII command SET CONTROL INIT RESET will cause an infinite reset loop and should NOT be used. Also command SET CONTROL INIT SET will cause same behavior. You can however survive form this state by deleting PSKEY_USR27 with PSTool.

13. SUPPORT

Contact Bluegiga Technologies Customer Service: support@bluegiga.com

Firmware, parameters, tools and documentation: http://www.bluegiga.com/techforum/

14. RELATED DOCUMENTATION

See also following documentation:

- **iWRAP Update Client manual**
- **BlueFlash Quick Start**
- **Firmware Guide**
- **Performance Measurement Guide**
- **PSTool Quick Start**

Visit also Tech-forum (**www.bluegiga.com/techforum/**) for additional information and design references.

15. APPENDIX I: IWRAP CONFIGURATION EXAMPLES

In this chapter some examples are given how to configure iWRAP to work in different application:

Example 1: Transparent slave

In this example iWRAP is configured to be a transparent SPP slave module, which only accepts connections and transmits data. No events or any other information is displayed. The configuration is displayed in the figure below:

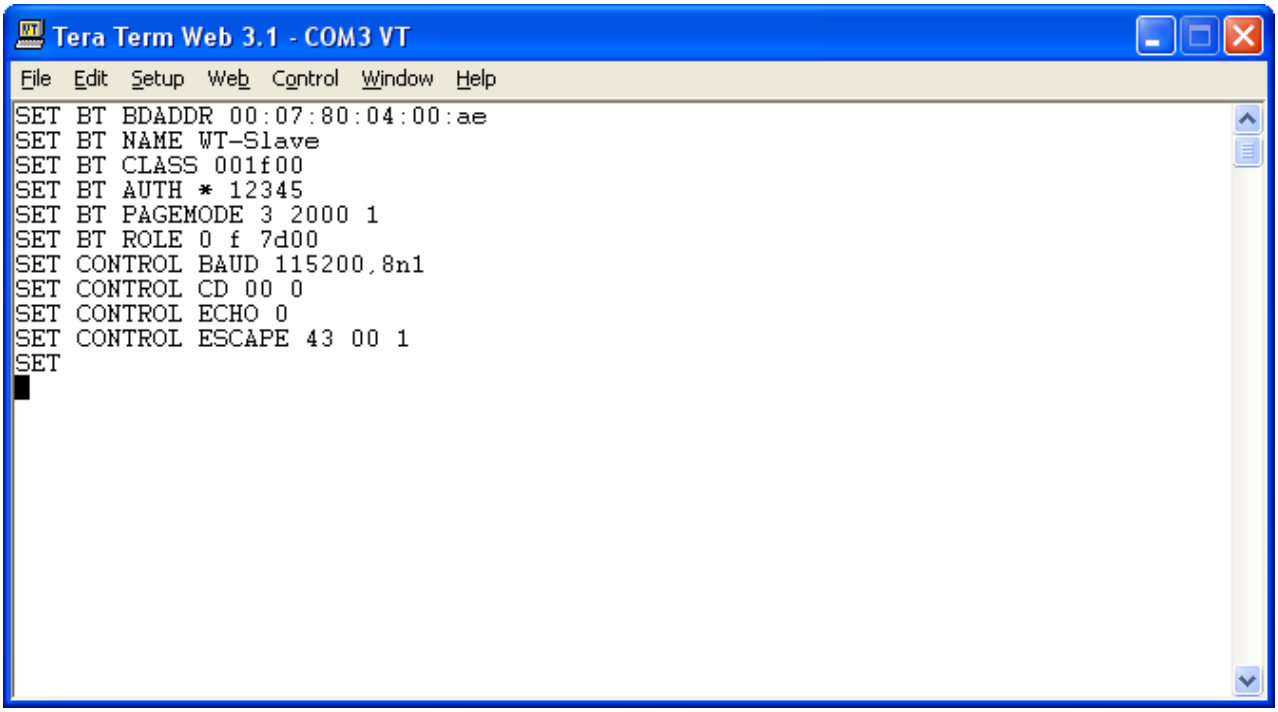


Figure 4: Slave configuration

The important settings in the figure are the following:

- **SET BT PAGEMODE 3 2000 1**

  With this setting we have configured iWRAP to be visible in the inquiry and to be connectable as a slave module should be.

  On the other hand in some cases slave mode does not need to be visible in the inquiry so our setting could be also: SET BT PAGEMODE 2 2000 1

- **SET BT ROLE 0 f 7d00**

  With this setting we have simply defined that iWRAP does not ask for master-slave switch when it's being connected. On the other hand all the link options (power saving etc.) are enabled if master wants to use them. This is the default setting.

- **SET CONTROL ECHO 0**

  This is important setting since we want the slave module to be transparent. That's why we disable all the event messages and boot banner by setting echo mode to 0.

Options:

- The timeouts for the slave module can be also configured. For the slave probably the supervision timeout is important i.e. when slave notices that connection is lost. This timeout is configured with the SET BT ROLE command.

- Sometimes the data rate is important and slave does not need to know about data and command mode switches. In these cases it might be useful to disable the escape sequence to obtain a higher data rate. This is done for example by issuing command: SET CONTROL ESCAPE – 00 1.

Example 2: Transparent master

In this example iWRAP is configured to be a transparent SPP master module, which always tries to keep a connection to a defined device and keep up transparent data mode where events or any other information are not displayed. The configuration is displayed in the figure below:

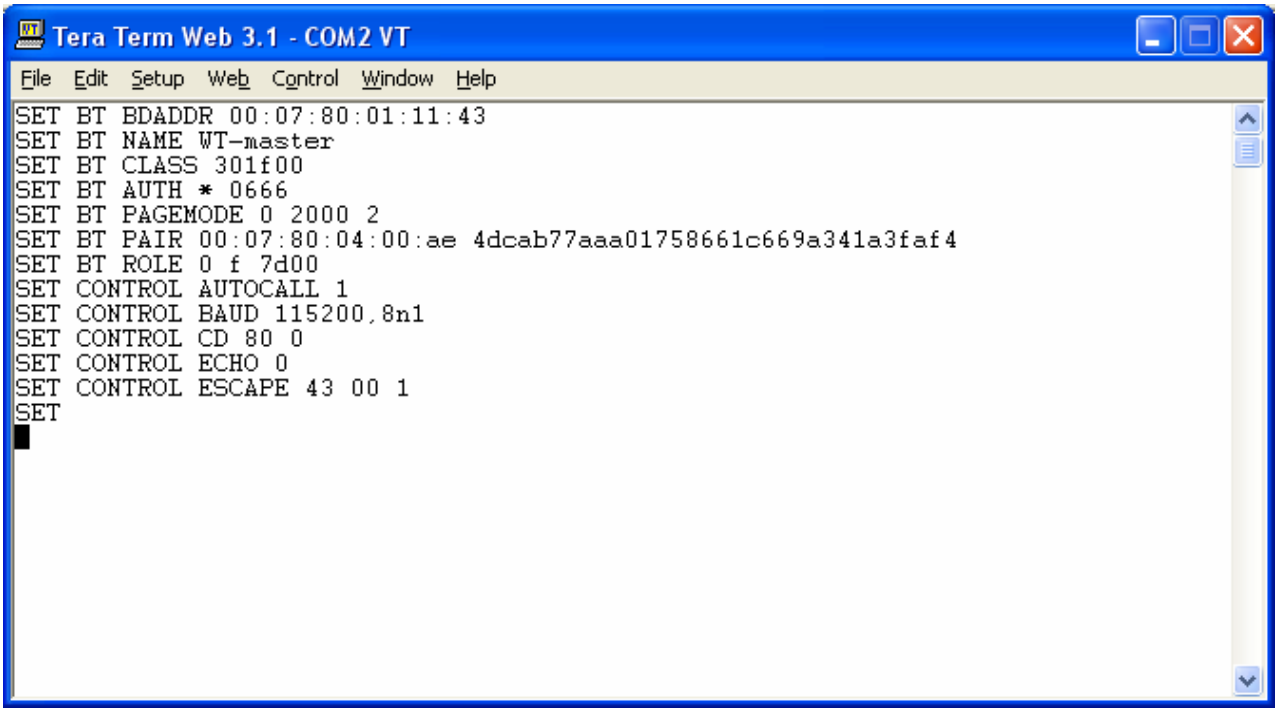


Figure 5: Transparent master

The important settings in master module are the following:

- **SET BT PAGEMODE 0 2000 1**

  Master module does not need to be visible in the inquiry nor connectable, since it only opens connection. That is why we have chosen this pagemode. This also conserves less power and makes master module fastest.

- **SET BT PAIR 00:07:80:04:00:ae 4dcab77aaa01758661c669a341a3faf4**

  Master module needs to know where is opens the connection. In iWRAP this is done based on the pairings. The slave module is the one and only paired device in master's memory. See the next example to find out how iWRAP works if there are several pairings.

- **SET BT ROLE 0 f 7d00**

  This is the default setting. Usually master module is master obviously but in some cases slaves want to do a master slave switch. That's why we allow it to be more flexible with any kind of devices.

  Tip:

  When forming Bluetooth network with WRAP Access server is wise for example to configure access server so that it's the master for every connection even if access server does not open the connections but only listens for them. For this kind of cases it's wise to allow the master slave switch even on a master module.

- **SET CONTROL AUTOCALL 1**

  This is the key setting in a master module, since it enables the autocall feature. Parameter '1' means that master module tries to open the connection using RFCOMM channel 1. This is only a safe setting when slave is a WRAP THOR device with iWRAP software, since iWRAP has Serial Port Profile always on channel 1. With other devices instead of '1' you should use '1101' (UUID) if you want to open SPP connection.

  Tip:

  Using channel instead of UUID is faster, because with channel there is no need to do service discovery. UUID is however safer since the channel for SPP might ware between different devices.

- **SET CONTROL ECHO 0**

  This is important setting since we want the slave module to be transparent. That's why we disable all the event messages and boot banner by setting echo mode to 0.

- **SET CONTROL CD 80 0**

  When using a transparent master module, it's very hard to know if there is a connection or not, since no events are displayed. That's why we have enabled the carrier detect signal with command SET CONTROL CD 80 0. This means that when there is a connection IO7 is driven high.

  Tip:

  Using IO7 for CD signal is wise especially when using WRAP THOR Evaluation Kits, since there is a led connected to IO7.

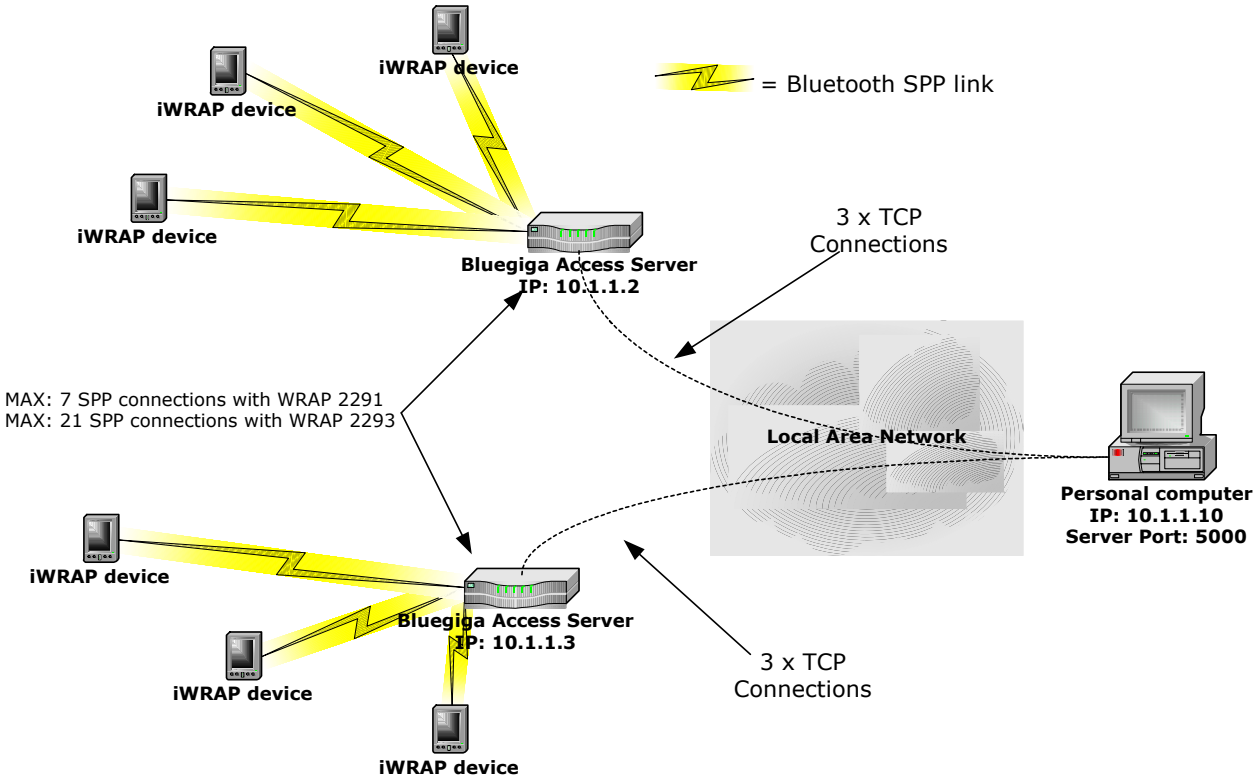
Options:

- The timeouts for the master module can be also configured. For the master probably the supervision timeout is important i.e. when it notices that connection is lost. This timeout is configured with the SET BT ROLE command. Also the time how long a connection establishment can take before error occurs might be important, at least for non transparent masters. This on the other hand can be configured with SET BT PAGEMODE command.

- Sometimes the data rate is important and there is possibility to use DTR signaling for controlling data and command mode switches. In these cases it might be useful to disable the escape sequence to obtain a higher data rate. This is done for example by issuing command: SET CONTROL ESCAPE – 80 1. Parameter '80' defines that IO 7 is used as a DTR signal. Notice however that CD and DTR signals can not be configured to use same IO.

Example 3: Transparent Bluetooth network with iWRAP and access server

In this example we build up a Bluetooth network with WRAP Access servers and iWRAP master modules. The network consists of several access servers and several master modules. The purpose is to provide a transparent 'always on' connectivity from iWRAP modules to a PC over Bluetooth and Local Area Network. The figure below illustrates this kind of a network set up:



The configuration in iWRAP is totally similar than in our second example. The only difference is that the slave device is only this time the WRAP Access Server and not a slave module.

Also WRAP Access servers have to configure correctly. The application providing the connectivity between the PC and iWRAPs is know as SPP-over-IP and it's a standard feature in WRAP access server with software version 2.0.4 and later. Please refer to SPP-over-IP documentation to see how access serves are configured.

Now if there are several WRAP access servers in our network and iWRAP devices are mobile we can do a little bit more configuration in iWRAPs. In our second example the case was more or less static and there was only one slave for the iWRAP. In a mobile situation the one defined slave is not always within a range and that causes an unwanted and probably long break in the connection. This can be avoided with the following configuration:

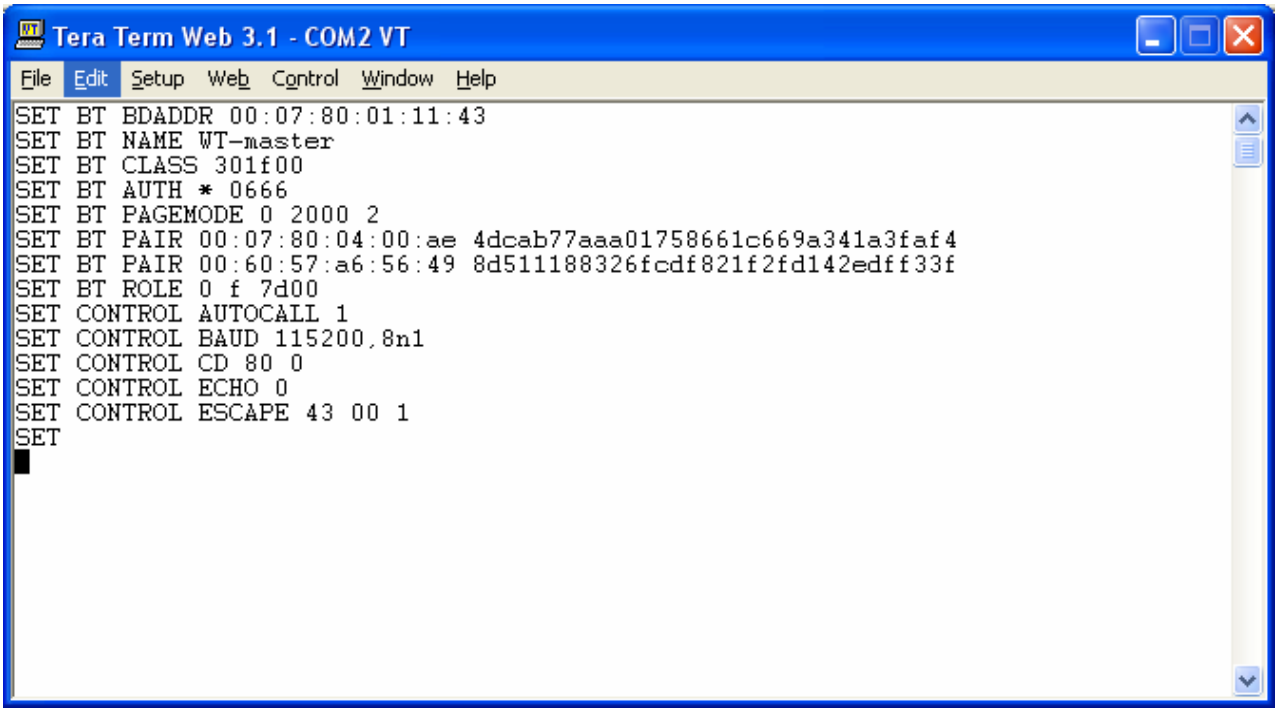


Figure 6: Configuration for multiple slaves

As you see the configuration is similar to configuration in the second example. The only difference is that there are two pairings in iWRAP.

How autocall in iWRAP works when there are several pairings? Well if there are several pairings in iWRAP first a transparent inquiry is made. When first paired device is found iWRAP ends the inquiry tries to connect this device. If the connection is successful iWRAP stays connected until the connection is being closed for one or another reason, and a new inquiry procedure begins.

So if all the access servers in this network are paired with all the iWRAPs with this configuration a transparent 'data only always on' network is possible to achieve. Of course there will be a short break in the connection when iWRAP is making in inquiry and connecting a new device. Notice also that iWRAP can only be paired with 16 other devices.