

BLUETOOTH HID PROFILE

iWRAP APPLICATION NOTE

Wednesday, 14 July 2010

Version 1.4

Copyright © 2000-2010 Bluegiga Technologies

All rights reserved.

Bluegiga Technologies assumes no responsibility for any errors which may appear in this manual. Furthermore, Bluegiga Technologies reserves the right to alter the hardware, software, and/or specifications detailed here at any time without notice and does not make any commitment to update the information contained here. Bluegiga's products are not authorized for use as critical components in life support devices or systems.

The WRAP, Bluegiga Access Server, Access Point and iWRAP are registered trademarks of Bluegiga Technologies.

The *Bluetooth* trademark is owned by the Bluetooth SIG Inc., USA and is licensed to Bluegiga Technologies. All other trademarks listed herein are owned by their respective owners.

VERSION HISTORY

Version	Comment
1.0	First version
1.1	iWRAP4 updates
1.2	Minor changes
1.3	Consumer page descriptions added
1.4	Multimedia key examples added

TABLE OF CONTENTS

1	Introduction.....	5
1.1	Human Interface Device Profile.....	5
2	iWRAP firmware overview.....	6
3	Using HID with iWRAP.....	8
3.1	Profile configuration.....	8
3.2	Class-of-Device configuration.....	8
3.3	Security configuration.....	9
3.4	Service discovery.....	10
3.5	Pairing.....	11
3.6	Connection establishment.....	12
3.6.1	HID control/data channels.....	12
3.7	Connection termination.....	13
3.7.1	HID control/data channels.....	13
3.8	Mode switching.....	13
3.9	HID usage.....	14
3.10	HID raw mode.....	17
3.11	Power saving.....	19
4	Example connection diagram.....	20
5	References.....	21
6	Contact Information.....	22

1 Introduction

This application note discusses Bluetooth Human Interface Device (HID) Profile its advantages and how this profiles can be utilized. Also practical examples are given how the HID is used with the iWRAP firmware.

1.1 Human Interface Device Profile

The HID profile defines the protocols, procedures and features to be used by Bluetooth HID such as keyboards, pointing devices, gaming devices and remote monitoring devices.

The HID defines two roles, that of a Human Interface Device (HID) and a Host:

- Human Interface Device (HID) – The device providing the service of human data input and output to and from the host.
- Host – The device using or requesting the services of a Human Interface Device.

The HID profile uses the universal serial bus (USB) definition of a HID device in order to leverage the existing class drivers for USB HID devices. The HID profile describes how to use the USB HID protocol to discover a HID class device's feature set and how a Bluetooth enabled device can support HID services using the L2CAP layer. The HID profile is designed to enable initialization and control self-describing devices as well as provide a low latency link with low power requirements.

The Bluetooth HID profile is built upon the Generic Access Profile (GAP), specified in the Bluetooth Profiles Document; see Referenced Documents. In order to provide the simplest possible implementation, the HID protocol runs natively on L2CAP and does not reuse Bluetooth protocols other than the Service Discovery Protocol.

Source: [1]



Figure 1: Typical HID use case

2 iWRAP firmware overview

iWRAP is an embedded firmware running entirely on the RISC processor of WT12, WT12 and WT32 modules. It implements the full *Bluetooth* protocol stack and many *Bluetooth* profiles as well. 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 system can interface to iWRAP firmware through one or more physical interfaces, which are also shown in the figure below. The most common interfacing is done through the UART interface by using the ASCII commands that iWRAP firmware supports. With these ASCII commands, the host can access *Bluetooth* functionality without paying any attention to the complexity, which lies in the *Bluetooth* protocol stack. GPIO interface can be used for event monitoring and command execution. PCM, SPDIF, I2S or analog interfaces are available for audio. The available interfaces depend on the used hardware.

The user can write application code to the host processor to control iWRAP firmware using ASCII commands or GPIO events. In this way, it is easy to develop *Bluetooth* enabled applications.

On WT32 there is an extra DSP processor available for data/audio processing.

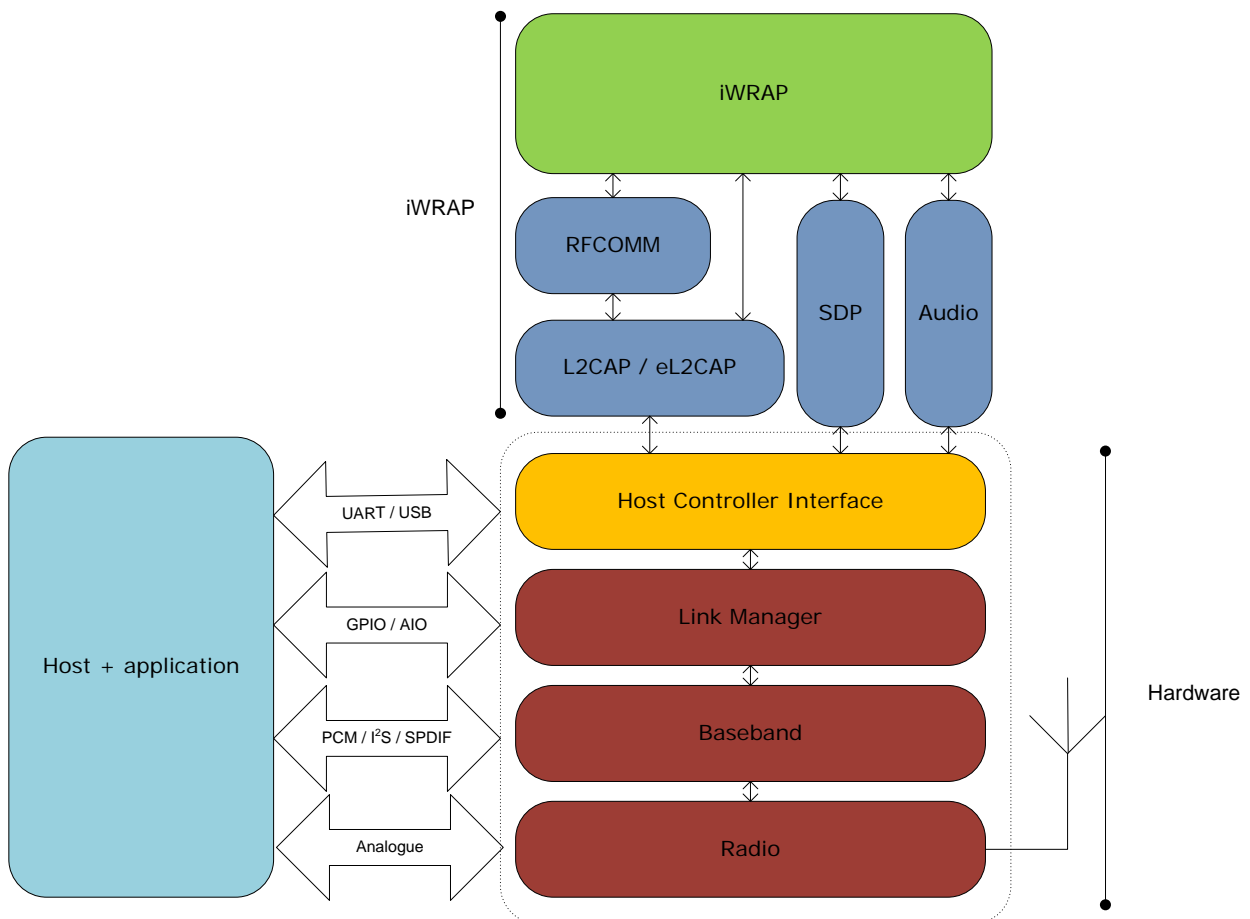


Figure 2: iWRAP Bluetooth stack

In the figure above, a Bluegiga *Bluetooth* module with iWRAP firmware could be connected to a host system for example through the UART interface. The options are:

- If the host system has a processor, software can be used to control iWRAP by using ASCII based commands or GPIO events.
- If there is no need to control iWRAP, or the host system does not need a processor, iWRAP can be configured to be totally transparent and autonomous, in which case it only accepts connections or automatically opens them.
- GPIO lines that WRAP THOR modules offer can also be used together with iWRAP to achieve additional functionality, such as Carrier Detect or DTR signaling.
- Audio interfaces can be used to transmit audio over a *Bluetooth* link.

3 Using HID with iWRAP

This chapter instructs the HID usage and configuration with the iWRAP firmware.

3.1 Profile configuration

HID is enabled with command “**SET PROFILE HID ON**”

Finally a reset is needed to for the HID profile to become active.

Below is an example how to enable HID mode.

```
SET PROFILE HID ON  
RESET
```

3.2 Class-of-Device configuration

The Bluetooth class-of-device needs to also be configured properly. This can be done with iWRAP command “**SET BT CLASS**” The HID class of devices are presented below.

SET BT CLASS 00540	: A keyboard device
SET BT CLASS 00580	: A pointing device
SET BT CLASS 005C0	: A combined keyboard/pointing device
SET BT CLASS 00500	: Not keyboard / not pointing device

Below is an example how to configure the class-of-device.

```
SET BT CLASS 005C0
```


3.3 Security configuration

The third configuration is related to pairing and security. iWRAP4 supports Bluetooth Secure Simple Pairing (SSP) and it is strongly recommended that it is used. iWRAP3 and older do not support SSP and legacy pairing needs to be used.

To enable SSP two possible configurations can be used depending on the device type. The configuration is done with iWRAP command “**SET BT SSP**”.

For a keyboard one should use setting:

SET BT SSP 2 0 : Enables SSP pairing for keyboard device and Man-in-the-middle protection

For a mouse or any other device without a keyboard one should use setting:

SET BT SSP 3 0 : Enables SSP just works mode

To support pairing with older devices that do not implement SSP. Two other configurations should also be made. The Bluetooth PIN code should be enabled with “**SET BT AUTH * <pin>**” command and a so called interactive pairing mode should also be enabled with “**SET CONTROL CONFIG**” command.

Below is an example how to configure the security for a keyboard device supporting SSP.

```
SET BT AUTH * 0000
SET BT SSP 2 0
SET CONTROL CONFIG 800
RESET
```

Below is an example how to configure the security for a keyboard device without SSP support.

```
SET BT AUTH * 0000
SET CONTROL CONFIG 800
RESET
```

3.4 Service discovery

Bluetooth technology enables wireless service discovery, so you can find out the capabilities the remote device supports. Wireless service discovery uses the Bluetooth Service Discovery Profile (SDP).

With iWRAP the service discovery is performed with command: “**SDP {bd_addr} {uuid}**”.

bd_addr	Bluetooth device address of the remote device.
uuid	Universally unique identifier. Refers to the Bluetooth profile to be discovered. For HID the uuid is 1124.

Below is an example how to perform a service discovery for HID device.

```
SDP 00:07:80:FF:FF:FF 1124
```

```
SDP 00:07:80:ff:ff:ff < I SERVICENAME S "HID" > < I PROTOCOLDESCRIPTORLIST < < U L2CAP I 11 >  
< U 0011 > > >
```

```
SDP
```

HID	= Service name
11	= L2CAP psm for HID profile

3.5 Pairing

Usually the pairing is initiated by the HID host device such as a PC or a mobile phone. The pairing may need actions on iWRAP, depending on which pairing mode is used. If SSP pairing is used, no actions are needed on iWRAP, but with the legacy Bluetooth pairing also the interactive pairing mode needs to be enabled and this requires user interaction.

When a device like a PC starts pairing with a keyboard it usually automatically displays a pin code that the user needs to type with a keyboard. This is the reason why interactive pairing is needed in iWRAP. the following example shows how the pairing procedure is made.

Interactive pairing example:

AUTH 00:21:86:35:c9:c8?	(Pairing is initiated from a PC and iWRAP shows AUTH event)
AUTH 00:21:86:35:c9:c8 12476505	(This is responded with AUTH command)

Pairing can also be initiated from iWRAP using the iWRAP command "**PAIR {bd_addr}**".

3.6 Connection establishment

3.6.1 HID control/data channels

Usually the HID connection is opened by the PC right after pairing. This can be seen by an incoming **RING** event generated by iWRAP.

Below is an example how a HID connection is received

```
RING 0 00:21:86:35:c9:c8 11 HID
```

However if a HID connection needs to be opened from iWRAP it can be done with a **CALL** command:

“CALL {*bd_addr*} 11 HID”

bd_addr Bluetooth device address of the remote device.

Below is an example how to set up a HID from iWRAP to a HID host device.

```
CALL 00:07:80:aa:bb:cc 11 HID  
CALL 0  
CONNECT 0 HID 17  
CONNECT 1 HID 19
```

A typical indications of outgoing call and successful connection are received (CALL and CONNECT). A two separate CONNECT event may be seen indicating the HID control and HID data channels.

The control channel **MUST** not be used for anything and the data transmission should happen only using the 2nd connection which is the data channel.

3.7 Connection termination

3.7.1 HID control/data channels

The HID data channel and control channels should be terminated on iWRAP using the “**KILL**” command instead of the typical “**CLOSE**” command. The reason for this is that on some PC Bluetooth stacks like the Widcomm stack closing the connection properly with **CLOSE** command also removes the link keys and re-establishing the connection requires that pairing is done again. The **KILL** command is used with the following syntax:

“**KILL {*bd_addr*}**”

bd_addr Bluetooth device address of the remote device.

HID connection termination.

```
KILL A8:7B:39:C3:CA:99  
NO CARRIER 1 ERROR 0  
NO CARRIER 0 ERROR 0
```

CLOSE command can also be used to terminate the connections, but in some cases this requires that the pairing is done again.

3.8 Mode switching

The escape character is disabled when the HID profile is used and a GPIO pin needs to be used for mode switching. This is because a user might want to transmit the escape (three ‘+’ characters by default) sequence over the HID connection and this might lead to an unwanted iWRAP mode switch.

3.9 HID usage

In iWRAP the HID keyboard layout is US and the following key codes are supported. These codes are transmitted over the HID data channel to the HID host device such as a PC.

Code	Description
0	Left control + space
1	Left control + a
2	Left control + b
3	Left control + c
4	Left control + d
5	Left control + e
6	Left control + f
7	Left control + g
8	Backspace
9	Tab
10	Enter
11	Left control + k
12	Left control + l
13	Enter
14	Left control + n
15	Left control + o
16	Left control + p
17	Left control + q

18	Left control + r
19	Left control + s
20	Left control + t
21	Left control + u
22	Left control + v
23	Left control + w
24	Left control + x
25	Left control + y
26	Left control + z
0	Left control + space
28	Esc
28-31	N/A
32-126	Corresponding ASCII character
127	backspace
128	Cursor up
129	Cursor right
130	Cursor down
131	Cursor left
132	Insert
133	Delete
134	Home
135	End

136	Page up
137	Page down
138	Mouse buttons up
139	Mouse up (10px)
140	Mouse right (10px)
141	Mouse down (10px)
142	Mouse left (10px)
143	Mouse button 1 (first clear with 138)
144	Mouse button 2 (first clear with 138)
145	Mouse button 3 (first clear with 138)
146-158	-
159	raw mode*
160-255	-

Table 1: Available HID key codes and mouse events

Note that the values the table above are expressed as decimals but they needs to be sent in hexadecimal format to the host. For example, to move the mouse upwards you will have to transmit the value.

*) See next chapter for raw mode.

3.10 HID raw mode

The basic HID mode allows the transmission of most common keys but if special keys (like multimedia keys) need to be transmitted iWRAP also implements also a raw mode. The raw enables sending of raw HID reports, one report at a time.

After the raw mode byte (159) you need to give the length of the report which is in keyboard report's case 10 and in mouse report's case 5. The raw reports must use the following format.

Keyboard report:

0x9f	length	0xa1	0x01	modifier	0x00	key code 1	key code 2	key code 3	key code 4	key code 5	key code 6
------	--------	------	------	----------	------	------------	------------	------------	------------	------------	------------

Figure 3: Raw HID keyboard report

Mouse report:

0x9f	length	0xa1	0x02	buttons	x-step	y-step
------	--------	------	------	---------	--------	--------

Figure 4: Raw HID mouse report

Consumer page report:

0x9f	length	0xa1	0x03	bitfield 1	bitfield 2	bitfield 3
------	--------	------	------	------------	------------	------------

Figure 5: Raw HID consumer report (iWRAP 4.1.0 and later)

Bitfield 1:

- 0x01 Volume Increment
- 0x02 Volume Decrement
- 0x04 Mute
- 0x08 Play/Pause
- 0x10 Scan Next Track
- 0x20 Scan Previous Track
- 0x40 Stop
- 0x80 Eject

Bitfield 2:

- 0x01 Email Reader
- 0x02 Application Control Search
- 0x04 AC Bookmarks
- 0x08 AC Home
- 0x10 AC Back
- 0x20 AC Forward
- 0x40 AC Stop
- 0x80 AC Refresh

Bitfield 3:

- 0x01 Application Launch Generic Consumer Control
- 0x02 AL Internet Browser
- 0x04 AL Calculator
- 0x08 AL Terminal Lock / Screensaver
- 0x10 AL Local Machine Browser
- 0x20 AC Minimize
- 0x40 Record
- 0x80 Rewind

Full key codes can be found from document USB HID Usage Tables:

http://www.usb.org/developers/devclass_docs/Hut1_11.pdf

Example of transmitting a key press (down) for a button 'a'

```
0x9f 0x0a 0xa1 0x01 0x00 0x00 0x04 0x00 0x00 0x00 0x00 0x00
```

Example of transmitting a key press (release) for a button 'a'

```
0x9f 0x0a 0xa1 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

To indicate a user has simultaneously pressed multiple keys (up to six different keys) you can use the rest of the key codes.

0x9f is used to indicate that a raw HID report is sent. Then **0x0a** indicates the length of the data to follow (10 bytes) and then the next 10 bytes are the raw report as per the HID standard.

Example how to launch the calculator application using consumer report:

```
0x9f 0x05 0xa1 0x03 0x00 0x00 0x04 0x9f 0x05 0xa1 0x03 0x00 0x00 0x00
```

The first raw frame indicates that the calculator button is pushed down and the second frame indicates that it is released.

Example how to send play/pause report:

```
0x9f 0x05 0xa1 0x03 0x08 0x00 0x00 0x9f 0x05 0xa1 0x03 0x00 0x00 0x00
```

Example how to send next track report:

```
0x9f 0x05 0xa1 0x03 0x10 0x00 0x00 0x9f 0x05 0xa1 0x03 0x00 0x00 0x00
```

Example how to send previous track report:

```
0x9f 0x05 0xa1 0x03 0x20 0x00 0x00 0x9f 0x05 0xa1 0x03 0x00 0x00 0x00
```

3.11 Power saving

iWRAP offers two power saving options. Sniff mode, which can be used to save power for active Bluetooth connections and deep sleep mode which puts the internal processor into a reduced duty cycle mode. Please refer to iWRAP user guide for more information about sniff and deep sleep modes.

One should also know that when Bluetooth connections are in active mode i.e. no power saving in use the master device uses 3-4 times less power than a slave device. Therefore for battery powered applications it might be useful to configure the device as a master rather than a slave, eventually considering role switching.

For HID devices the Bluetooth 2.1 + EDR introduced a new sniff mode called sniff sub rating. In brief the sniff sub rate mode increases the sniff interval after a certain period of inactivity and therefore reduces the current consumption. iWRAP4 supports sniff sub rating mode. Please refer into iWRAP4 user guide for more information.

4 Example connection diagram

An example of HID configuration and a simple HID connection setup is illustrated below.

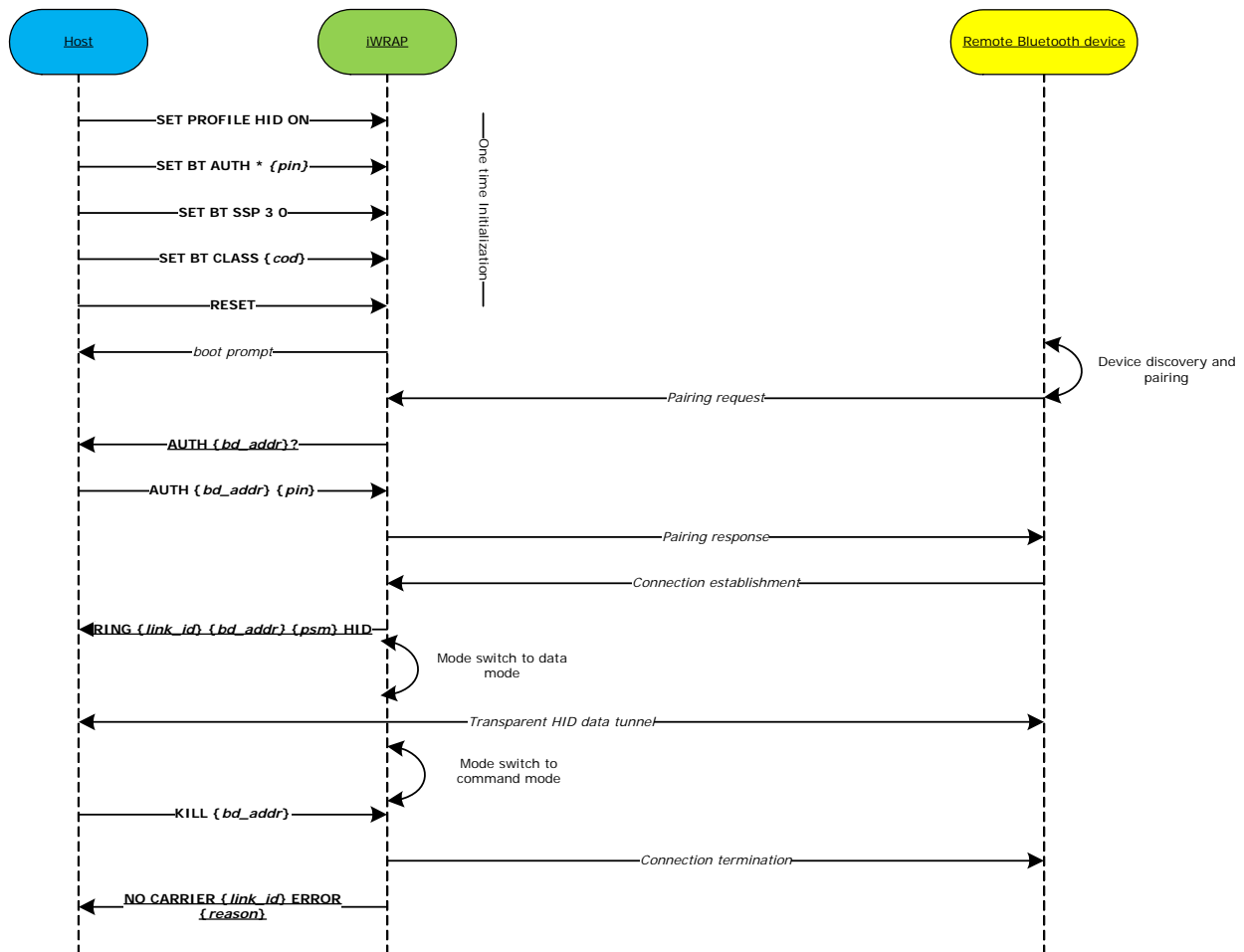


Figure 6: HID connection example

In the above example the **AUTH** event is only seen if a legacy pairing with pin code and interactive pairing is made. If the legacy pairing does not take place and SSP pairing is used instead then **AUTH** event is not seen and there is no need to reply to it with the **AUTH** command. With SSP pairing, depending on the SSP mode, however SPP events may be displayed and then need to be responded with correct SSP pairing commands.

Also depending on the remote Bluetooth device one or two **RING** events may be seen during the connection establishment.

Please refer to iWRAP user guide for more information about the iWRAP command and events.

5 References

- [1] The Bluetooth SIG, Human Interface Device Profile overview, URL: <http://www.bluetooth.com/Bluetooth/Technology/Works/HID.htm>

6 Contact Information

Sales: sales@bluegiga.com

Technical support: support@bluegiga.com
<http://www.bluegiga.com/techforum/>

Orders: orders@bluegiga.com

Head Office / Finland:

Phone: +358-9-4355 060

Fax: +358-9-4355 0660

Street Address:

Sinikalliontie 5A

02630 ESPOO

FINLAND

Postal address:

P.O. BOX 120

02631 ESPOO

FINLAND

Sales Office / USA:

Phone: (781) 556-1039

Bluegiga Technologies, Inc.

99 Derby Street, Suite 200 Hingham, MA 02043