

## ***Amp'edUP API Reference Guide***

27 June, 2011

# Amp'edUP API Reference Guide

All rights reserved. Copyright © 2011 by Amp'ed RF Technology, Inc.

No part of the contents of this document may be reproduced or transmitted in any form, by any means or for any purpose, without the prior written permission of Amp'ed RF Technology, Inc.

The information in this document and the product it describes are subject to change without notice.

The software program described in this document is provided to its authorized users pursuant to a software license agreement, and any use of this software program is strictly subject to the terms and conditions, including confidentiality obligations, set forth in the applicable software license agreement between Amp'ed RF Technology, Inc. and the licensee.

This document is intended only to assist the reader in the use of the product. Amp'ed RF Technology, Inc. shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

# TABLE OF CONTENTS

---

## Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
1.1	Conventions .....	4
1.2	Reference .....	4
<b>2</b>	<b>OVERVIEW .....</b>	<b>5</b>
2.1	Overview of Embedded Applications Using the Amp'edUP Bluetooth Interface .....	5
<b>3</b>	<b>AMP'EDUP SIGNALS .....</b>	<b>7</b>
3.1	Signals .....	7
3.1.1	Signal Format .....	7
3.1.2	Signal Binding .....	7
<b>4</b>	<b>GAP SIGNALS .....</b>	<b>9</b>
4.1	GAP Command Signals .....	9
4.1.1	SIG_AMP_GAP_DEVICE_SETUP .....	9
4.1.2	SIG_AMP_GAP_SECURITY_SETUP .....	12
4.1.3	SIG_AMP_GAP_PINCODE_REPLY .....	14
4.1.4	SIG_AMP_GAP_START_INQUIRY .....	16
4.1.5	SIG_AMP_GAP_CANCEL_INQUIRY .....	18
4.1.6	SIG_AMP_GAP_REMOTE_NAME_REQ .....	18
4.1.7	SIG_AMP_GAP_AUTHORIZE_RESPONSE .....	19
4.1.8	SIG_AMP_GAP_BOND .....	20
4.2	GAP Event Signals .....	22
4.2.1	SIG_AMP_GAP_RESULT .....	22
4.2.2	SIG_AMP_GAP_INQUIRY_RESULT .....	23
4.2.3	SIG_AMP_GAP_REMOTE_NAME_CON .....	24
4.2.4	SIG_AMP_GAP_PIN_CODE_REQUEST .....	25
4.2.5	SIG_AMP_GAP_AUTHORIZE_REQUEST .....	26
4.2.6	SIG_AMP_GAP_ABORT_SECURITY .....	27
4.2.7	SIG_AMP_GAP_AUTH_COMPLETE .....	27
4.2.8	SIG_AMP_GAP_LINK_KEY_NOTIFY .....	28
4.3	GAP API Functions .....	29
4.3.1	Setting Local Name .....	29
4.3.2	Setting the Pairable Mode .....	30
<b>5</b>	<b>SPP SIGNALS .....</b>	<b>31</b>
5.1	SPP Command Signals .....	31
5.1.1	SIG_AMP_SPP_LISTEN_REQ .....	31

- 5.1.2 SIG\_AMP\_SPP\_CONNECT\_REQ.....32
- 5.1.3 SIG\_AMP\_SPP\_SEND.....34
- 5.1.4 SIG\_AMP\_SPP\_DISCONNECT\_REQ.....35
- 5.2 SPP Event Signals .....36**
  - 5.2.1 SIG\_AMP\_SPP\_RECV.....37
  - 5.2.2 SIG\_AMP\_SPP\_LISTEN\_CON .....38
  - 5.2.3 SIG\_AMP\_SPP\_CONNECT\_CON.....39
  - 5.2.4 SIG\_AMP\_SPP\_CONNECT\_STATUS .....40
  - 5.2.5 SIG\_AMP\_SPP\_DISCONNECT\_CON.....41
- 6 SDAP SIGNALS.....43**
  - 6.1 SDAP Command Signals.....43**
    - 6.1.1 SIG\_AMP\_SDAP\_SERVICE\_SEARCH\_REQ.....43
  - 6.2 SDAP Event Signals.....44**
    - 6.2.1 SIG\_AMP\_SDAP\_SERVICE\_SEARCH\_CON .....45
- 7 MISCELLANEOUS SIGNALS.....48**
  - 7.1 Miscellaneous Command Signals.....48**
    - 7.1.1 SIG\_AMP\_CREATE\_DEVICE\_DATABASE.....48
    - 7.1.2 SIG\_AMP\_DELETE\_DEVICE\_DATABASE.....50
    - 7.1.3 SIG\_AMP\_READ\_DEVICE\_DATABASE .....51
    - 7.1.4 SIG\_AMP\_HOLD\_MODE .....51
    - 7.1.5 SIG\_AMP\_SNIFF\_MODE .....53
    - 7.1.6 SIG\_AMP\_EXIT\_SNIFF\_MODE .....54
  - 7.2 Miscellaneous Event Signals .....55**
    - 7.2.1 SIG\_AMP\_GAP\_DEVICE\_DB\_RESULT .....55
    - 7.2.2 SIG\_AMP\_MODE\_CHANGE\_RESULT.....56
    - 7.2.3 SIG\_AMP\_GAP\_LL\_CONN\_DOWN .....57

# 1 Introduction

This publication, *Amp'edUP API Reference Guide*, serves as a reference for all Amp'edUP signals and messages used to interface with the application development framework. The *Amp'edUP API Reference Guide* explains how to build signals used to interface to the lower Bluetooth profiles and layers, such as SPP, SDAP, GAP, etc. This document will not explain how to write an embedded application using the Amp'edUP interface signals. The programmer should be familiar with Bluetooth profiles and various protocol stack concepts.

## 1.1 Conventions

Document Name	<i><u>Document Names</u></i> are also italicized and they are underlined as well.
Macros, tasks, modules, and function names	Macros, tasks, modules, and function names are in bold and as they appear in the source code.
File Name	<i>File Names</i> are italicized.
Source code	Denotes source code
SIG_AMP_GAP_GET	Command signal names are bold and all capital letters (no spaces)
SIG_AMP_GAP_RESULT	Event signal names are bold, italicized, and all capital letters (no spaces)
<a href="mailto:support@ampedrf.com">support@ampedrf.com</a>	Denotes a hyperlink to website or email address



*Note: highlights important notes about previous information.*

## 1.2 Reference

The following terms are referred to in this document:

**Table 1. Definitions and Acronyms**

Term	Description/Meaning
API	Application Programmer Interface
AT	Text based command standard commonly used for modems
BD	Bluetooth Device

Term	Description/Meaning
BdAddr	Bluetooth Device Address
GAP	Generic Access Profile
HCI	Host Controller Interface
ID	Identification
Inq.	Inquiry
L2CAP	Logical Link Control and Adaptation Protocol
LAN	Local Area Network
LAP	LAN Access Point
LS	Least Significant
OBEX	Object Exchange
OS	Operating System
PIN	Personal Identification Number
PPP	Point-to-Point Protocol
RAM	Random Access Memory
Req	Request
RFCOMM	Serial Port Emulation
RTOS	Real Time Operating System
SDAP	Service Discovery Application Profile
SDP	Service Discovery Protocol
SPP	Serial Port Profile
TCP	Transport Control Protocol
ULS	Upper Layer Stack
USB	Universal Serial Bus
UUID	Universally Unique Identifier

## 2 Overview

---

### 2.1 Overview of Embedded Applications Using the Amp'edUP Bluetooth Interface

The Amp'edUP Bluetooth API is comprised of signals that invoke the functionality provided by profiles such as SPP, SDAP and GAP. They are sent to and from embedded applications.

Signals going from an embedded application to the ULS are called command signals. Signals going from the ULS to an embedded application are called event signals. Figure 1 shows this structure.

The following sections explain each command and event signal that can be sent to and received from Amp'edUP.

This document assumes the embedded application will run in its own task context (either single or multiple tasks) under the SAFE RTOS operating system.

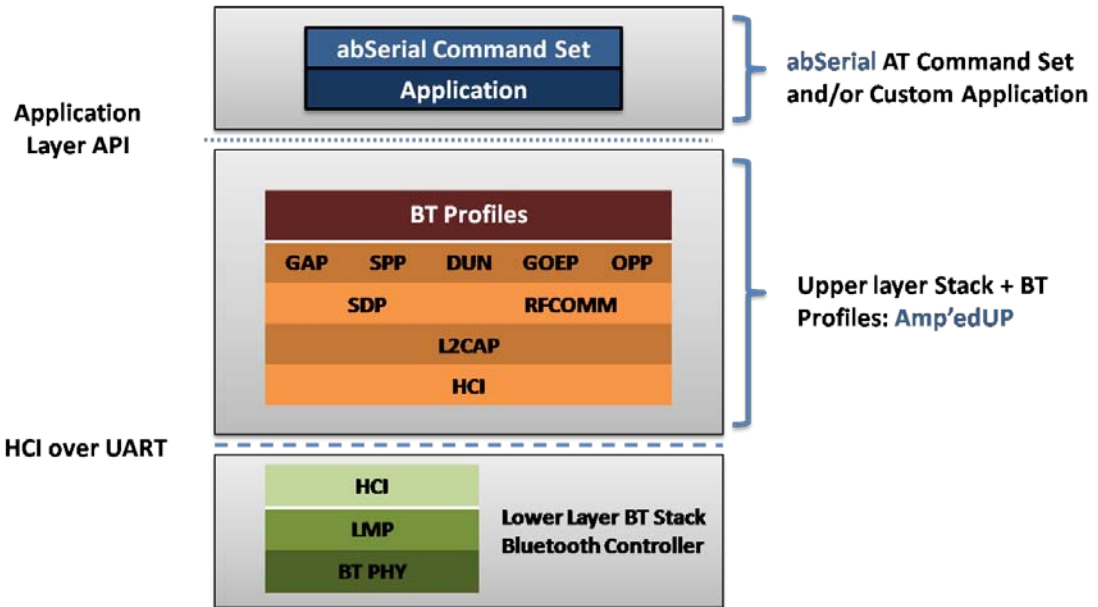


Figure 1. Embedded Bluetooth Protocol Stack Amp'edUP

## 3 Amp'edUP Signals

---

### 3.1 Signals

All Amp'edUP signals have the same general format, may have a signal number and an associated structure whether they are a command signal or event signal.

The following sections detail the purpose, signal number, and associated signal structure including the type name for every command signal and the associated return event signal - if any.

The signals are grouped under categories according to the profile they belong to. They are:

- GAP
- SPP
- SDAP
- Miscellaneous

#### 3.1.1 Signal Format

The signal has been explained with the following format:

Signal Structure Type Name: Type Name

Signal Description is given next and a table of signal parameters.

Signal ID	Signal Parameters
0XXXXX	Defined separately for each of the signals below

##### 3.1.1.1 Signal Parameters

Signal parameters are then individually explained.

##### Signal ID Define

Size: 2 Bytes

Value	Parameter Description
0XXXXX (pre-defined names)	Signal ID definition that identifies the signal (specific value for each signal defined below)

#### 3.1.2 Signal Binding

For signal binding, the default destination is abSerial, but a user can change this to either "Repeat" or "Redirect". The below figure identifies how this value routes an incoming signal to abSerial and/or to the SDK Main Application.

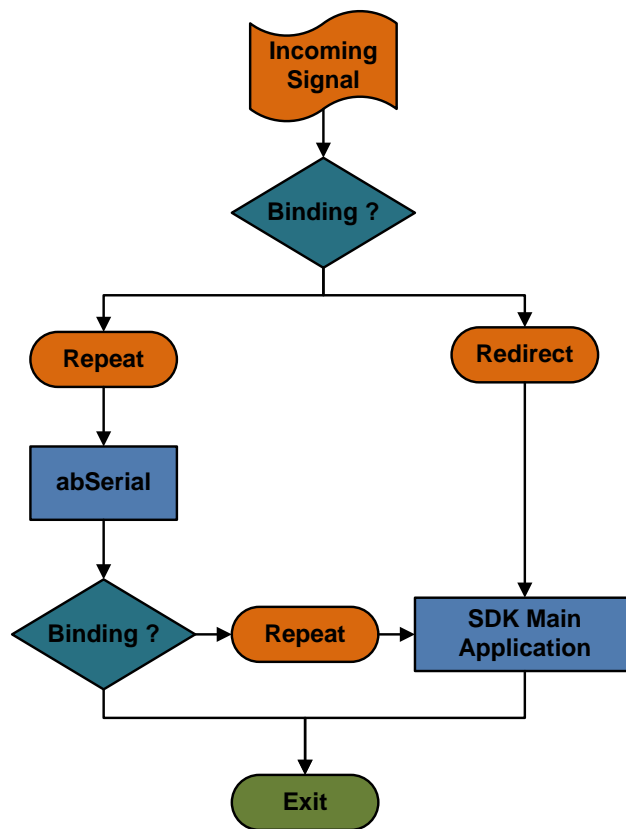


Each signal that is desired to use in the SDK Main Application, must be individually added using this function:

```
OS_BindAmpSignal(SIGNAL_REPEAT or SIGNAL_REDIRECT, Signal ID);
```

The following signals are defaulted to SIGNAL\_REPEAT in the standard SDK:

```
// Reserve AMP Signals
OS_BindAmpSignal(SIGNAL_REPEAT, SIG_AMP_GAP_RESULT);
OS_BindAmpSignal(SIGNAL_REPEAT, SIG_AMP_SPP_LISTEN_CON);
OS_BindAmpSignal(SIGNAL_REPEAT, SIG_AMP_SPP_CONNECT_CON);
OS_BindAmpSignal(SIGNAL_REPEAT, SIG_AMP_SPP_DISCONNECT_IND);
OS_BindAmpSignal(SIGNAL_REPEAT, SIG_AMP_SPP_DISCONNECT_CON);
OS_BindAmpSignal(SIGNAL_REPEAT, SIG_AMP_GAP_LL_CONN_DOWN);
```



**Figure 1:** Redirect and Repeat interaction with Incoming Signal, abSerial and SDK Main Application

## 4 GAP Signals

GAP command signals, going from an embedded application to the Amp'edUP stack, provide the following functionality:

- Initial set up of the Bluetooth device,
- Setting up security requirements,
- Device inquiry,
- Send bonding PIN code and authorization,
- Request name of remote device.

GAP event signals, going from the Amp'edUP stack to an embedded application, indicate the following:

- Results of the GAP command signals,
- GAP Inquiry results,
- Request for Bonding PIN code and authorization.

For detailed information on the terms used in this section, please refer to the [\*Bluetooth Generic Access Profile V1.1\*](#).

### 4.1 GAP Command Signals

The following list contains all GAP signals going from an embedded application to the AMP.

- SIG\_AMP\_GAP\_DEVICE\_SETUP
- SIG\_AMP\_GAP\_SECURITY\_SETUP
- SIG\_AMP\_GAP\_PINCODE\_REPLY
- SIG\_AMP\_GAP\_START\_INQUIRY
- SIG\_AMP\_GAP\_CANCEL\_INQUIRY
- SIG\_AMP\_GAP\_REMOTE\_NAME\_REQ
- SIG\_AMP\_GAP\_AUTHORIZE\_RESPONSE
- SIG\_AMP\_GAP\_BOND

#### 4.1.1 SIG\_AMP\_GAP\_DEVICE\_SETUP

The purpose of the **SIG\_AMP\_GAP\_DEVICE\_SETUP** command signal is to setup the Amp'ed RF device by specifying configuration parameters such as discoverability, connectability, and class of device.

Signal Structure Type Name: tSIG\_AMPGapDeviceSetup

Signal ID	Parameters
SIG_AMP_GAP_DEVICE_SETUP	Transac DiscoverableMode

Signal ID	Parameters
	DiscoverableDuration DiscoverableInterval ConnectableMode ConnectableDuration ConnectableInterval CodMinor CodMajor CodServices

#### 4.1.1.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Context Information From the Application

##### DiscoverableMode

Size: 1 Byte

Value	Parameter Description
0XXXXX (2 Bytes)	Used to set the discoverability of the device. AMP_GAP_NON_DISCOVERABLE_MODE (0x00); AMP_GAP_LIMITED_DISCOVERABLE_MODE (0x01) (Not Supported); AMP_GAP_GENERAL_DISCOVERABLE_MODE 0x02)

##### DiscoverableDuration

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	The duration of an inquiry scan that allows a device to be discovered. In units of 625 $\mu$ s intervals. Recommended range: 0x0012 – 0x1000 (11.25 – 2560 ms) If a value of "0" is sent, and then the default of 0x0012 is used.

##### DiscoverableInterval

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	The amount of time between the start of two inquiry scans. In units of 625 $\mu$ s intervals. Recommended Range: 0x0012 – 0x1000 (11.25 – 2560 ms) If a value of "0" is sent, then the default 0x0800 is used.

##### ConnectableMode

Size: 1 Byte

Value	Parameter Description
0XXXXX	Used to set the connectability of the device.

Value	Parameter Description
(2 bytes)	AMP_GAP_NON_CONNECTABLE_MODE (0x00); AMP_GAP_CONNECTABLE_MODE (0x01) 0x02 – 0xffff Reserved for Future Use

### ConnectableDuration

Size: 2 Bytes

Value	Parameter Description
0xXXXX (2 Bytes)	The duration of a page scan that allows the device to be connected. In units of 625 $\mu$ s intervals. Recommended Range: 0x0012 – 0x1000 (11.25 – 2560 ms) If a value of “0” is sent, then the default of 0x0012 is used.

### ConnectableInterval

Size: 2 Bytes

Value	Parameter Description
0xXXXX (2 Bytes)	The amount of time between the start of two page scans. In units of 625 $\mu$ s intervals. Recommended Range: 0x0012 – 0x1000 (11.25 – 2560 ms) If a value of “0” is sent, then the default 0x0800 is used.

### CodMinor

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Class of Device (Minor Device Class) Bit7 to bit2 of this value correspond to bit7 to bit2 of the Minor Device Class in the <i>Bluetooth Assigned Numbers</i> specification.

### CodMajor

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Class of Device (Major Device Class) Bit4 to bit0 of this value correspond to bit12 to bit8 of the Major Device Class in the <i>Bluetooth Assigned Numbers</i> specification.

### CodServices

Size: 4 Bytes

Value	Parameter Description
0xXXXX (2 Bytes)	Class of Device Service Classes Bit15 to bit5 of this value correspond to bit23 to bit13 of the Major Service Classes in the <i>Bluetooth Assigned Numbers</i> specification.

#### 4.1.1.2 Event(s) Generated

When the **SIG\_AMP\_GAP\_DEVICE\_SETUP** command is finished, a **SIG\_AMP\_GAP\_RESULT** event signal will be sent to the application indicating the result. One of the following result codes is possible:

AMP\_RESULT\_DONE (0x0000) – When the device is setup successfully.

AMP\_RESULT\_PARAMETER\_ERROR (0x0002) – When the device setup failed.

### 4.1.2 SIG\_AMP\_GAP\_SECURITY\_SETUP

The purpose of the **SIG\_AMP\_GAP\_SECURITY\_SETUP** signal is to configure the security related parameters of the device in the following ways:

- Device is pairable or not,
- Security mode to be used (none or link level or service level security),
- Device uses a variable pin code or a fixed pin code (and if it is a fixed pin code, then configures the fixed pin code and its length),
- Key type to be used (unit or combination key),
- Minimum encryption key size needed,

This signal has to be sent before initiating any security procedures like bonding.

Signal Structure Type Name: tSIG\_AMPGapSecuritySetup

Signal ID	Parameters
SIG_AMP_GAP_SECURITY_SETUP	Transac PairableMode ConnectOnlyPaired SecurityMode PinType PinLength FixedPin MinEncrKeySize KeyType SignallingTaskId



*Note: there is no security level (authorization, authentication and encryption) parameter in this structure when service level security mode is used. In that mode, it is automatically set to both authentication and encryption required (no authorization).*

#### 4.1.2.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context Information From the Application

### PairableMode

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	To Allow or disallow pairing AMP_GAP_NON_PAIRABLE (0x00); AMP_GAP_PAIRABLE (0x01) 0x02 – 0xff Reserved for Future Use

### ConnectOnlyPaired

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	If AMP_GAP_CONNECT_ONLY_PAISED (0x01), only connections to previously paired devices are allowed (this is typically set to AMP_GAP_CONNECT_ALL_DEVICES (0x00) to allow connections to all devices and for most applications)

### SecurityMode

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	To configure the security mode of the device 0x01 (None) 0x02 (Service level, not used with BT spec v2.1) 0x03 (Link level, not used with BT spec v2.1) 0x04 (SM4, required with BT spec v2.1)

### PinType

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	To configure the PIN type of the device. If it is Fixed PIN then Pin Length and Fixed PIN parameters are valid. AMP_GAP_VARIABLE_PIN (0x00); AMP_GAP_FIXED_PIN (0x01) 0x02 – 0xff Reserved For Future Use

### PinLength

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	To configure the Length of the fixed PIN of the device 0x00 – 0x10 Valid 0x11 – 0xff Invalid Length

### FixedPin

Size: 16 Bytes

Value	Parameter Description
An Array of 16 unsigned bytes	To configure the fixed PIN of the device. The PIN has to be filled from index 0 to (PinLength-1)

### MinEncrKeySize

Size: 2 Bytes

Value	Parameter Description
0xXXXX (2 Bytes)	To configure the Minimum Encryption Key size to be used (in number of bytes) 0x01 – 0x10 valid key sizes 0x11 – 0xff Invalid Key sizes

### KeyType

Size: 2 Bytes

Value	Parameter Description
0xXXXX (2 Bytes)	To configure the Key type to be used in the security procedures AMP_GAP_UNIT_KEY (0x00); (deprecated in BT spec v1.2) AMP_GAP_COMB_KEY (0x01); 0x02 – 0xff Reserved For Future Use

### SignallingTaskId

Size: 2 Bytes

Value	Parameter Description
0xXXXX (2 Bytes)	The task ID of the application task to which any PIN code request event signal should be sent in the variable PIN type. In the RTOS environment, this ID is only the ID returned when the application task has been created.

#### 4.1.2.2 Event(s) Generated

When the **SIG\_AMP\_GAP\_SECURITY\_SETUP** command is finished, a **SIG\_AMP\_GAP\_RESULT** event signal will be sent to the application indicating the result. One of the following result codes is possible:

AMP\_RESULT\_DONE (0x0000) – When the Security is setup successfully.

AMP\_RESULT\_PARAMETER\_ERROR (0x0002) – When the Security setup failed.

#### 4.1.3 SIG\_AMP\_GAP\_PINCODE\_REPLY

The purpose of the **SIG\_AMP\_GAP\_PINCODE\_REPLY** command signal is to send the PIN code (pass key) when the event signal called **SIG\_AMP\_GAP\_PIN\_CODE\_REQUEST** is received by the application.

Signal Structure Type Name: tSIG\_AMPGapPinCodeReply

Signal Number	Parameters
SIG_AMP_GAP_PINCODE_REPLY	Transac BdAddr PinLength VariablePin Response

#### 4.1.3.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context Information From the Application

##### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	The Bluetooth device address of the other device with which the security procedure is done.

##### PinLength

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	The Length of the Variable PIN of the device 0x00 – 0x10 Valid 0x11 – 0xff Invalid Length

##### VariablePin

Size: 16 Bytes

Value	Parameter Description
An Array of 16 unsigned bytes	The Variable PIN of the device. The PIN has to be filled from index 0 to (PinLength-1)



### Response

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Indication of whether the PIN Code request is honored with a PIN or not AMP_GAP_POSITIVE_REPLY (0x00) – Positive Response (PIN code & PIN length are valid) Non-Zero – Negative Response (PIN code & PIN length parameters are ignored)

#### 4.1.3.2 Event(s) Generated

None.

#### 4.1.4 SIG\_AMP\_GAP\_START\_INQUIRY

The purpose of the **SIG\_AMP\_GAP\_START\_INQUIRY** command signal is to allow the embedded application to begin inquiring for nearby devices.

Signal Structure Type Name: tSIG\_AMPGapStartInquiry

Signal Number	Parameters
SIG_AMP_GAP_START_INQUIRY	Transac Length NoOfResponses Limited FilterNewDevices FilterCod FilterCodMask FilterBdAddr

#### 4.1.4.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the Application

##### Length

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Length of the inquiry process (in the unit Of 1.28 seconds) Range: 0x01-0x30 (1.28 seconds – 61.4 seconds)

##### NoOfResponses

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX	The maximum number of responses to wait for in doing the inquiry.

Value	Parameter Description
(bool)	

### Limited

Size: 4Bytes

Value	Parameter Description
0XXXXXXXX (bool)	The device should perform limited inquiry process. (0x00: General Inquiry) (Non zero value: Limited Inquiry)

### FilterNewDevices

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Filter any new devices received during inquiry. (0x00: Filter is disabled.) (Non zero value: Filter is enabled). This is not really used; filtering is done as long as one of the next three fields is non-zero.

### FilterCod

Size: 4 Bytes

Value	Parameter Description
0XXXXXX (Lower 3 Bytes)	The class of device to be used to filter devices received during inquiry. With FilterCod, any combination of major service class, major device class, and minor device class can be set. This is used in conjunction with FilterCodMask.
0xXX (Most significant byte)	Not used.

### FilterCodMask

Size: 4 Bytes

Value	Parameter Description
0XXXXXX (Lower 3 Bytes)	Filter devices received during inquiry on their class of device and specifying this class of device mask. This specifies the bits of FilterCod on which to filter.
0xXX (Most significant byte)	Not Used.

### FilterBdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address to be used to do filter devices received during inquiry

#### 4.1.4.2 Event(s) Generated

There are two events generated for this command.

When the **SIG\_AMP\_GAP\_START\_INQUIRY** command is accepted, a **SIG\_AMP\_GAP\_RESULT** event signal will be sent to the application indicating the result. One of the following result codes is possible:

AMP\_RESULT\_PENDING (0x0001) – When the inquiry is started successfully.

AMP\_RESULT\_ILLEGAL\_OPERATION (0x0003) – When starting up inquiry failed.

When the **SIG\_AMP\_GAP\_START\_INQUIRY** command is being executed, each of the inquiry results would be reported in a **SIG\_AMP\_INQUIRY\_RESULT** event signal.

When the **SIG\_AMP\_GAP\_START\_INQUIRY** command is finished, it would generate another **SIG\_AMP\_GAP\_RESULT** with the result code AMP\_RESULT\_DONE (0x0000).

#### 4.1.5 SIG\_AMP\_GAP\_CANCEL\_INQUIRY

The purpose of the **SIG\_AMP\_GAP\_CANCEL\_INQUIRY** command signal is to allow the embedded application to cancel any pending inquiry.

Signal Structure Type Name: tSIG\_AMPGapCancelInquiry

Signal Number	Parameters
SIG_AMP_GAP_CANCEL_INQUIRY	Transac

##### 4.1.5.1 Signal Parameters

Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context Information From the Application

##### 4.1.5.2 Event(s) Generated

When the **SIG\_AMP\_GAP\_CANCEL\_INQUIRY** command is finished, a **SIG\_AMP\_GAP\_RESULT** event signal will be sent to the application with a result code indicating the result. One of the following result codes is possible.

AMP\_RESULT\_DONE (0x0000), when inquiry is cancelled successfully.

AMP\_RESULT\_ILLEGAL\_OPERATION (0x0003), when the cancel failed.

#### 4.1.6 SIG\_AMP\_GAP\_REMOTE\_NAME\_REQ

The purpose of the **SIG\_AMP\_GAP\_REMOTE\_NAME\_REQ** command signal is to allow the embedded application to find the name of the remote device.

Signal Structure Type Name: tSIG\_AMPGapRemoteNameReq

Signal Number	Parameters
SIG_AMP_GAP_REMOTE_NAME_REQ	Transac BdAddr

#### 4.1.6.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the Application

##### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the Remote device.

#### 4.1.6.2 Event(s) Generated

There are two event signals generated for this command signal.

When the **SIG\_AMP\_GAP\_REMOTE\_NAME\_REQ** command is accepted, an event signal called **SIG\_AMP\_GAP\_RESULT** will be sent to the application indicating the result. One of the following result codes is possible:

AMP\_RESULT\_PENDING (0x0001) – When the remote name request is started successfully.

AMP\_RESULT\_ILLEGAL\_OPERATION (0x0003) – When already another remote name request is pending to be performed.

When the **SIG\_AMP\_GAP\_REMOTE\_NAME\_REQ** command is finished, an event signal called **SIG\_AMP\_GAP\_REMOTE\_NAME\_CON** will be sent to the embedded application.

#### 4.1.7 SIG\_AMP\_GAP\_AUTHORIZE\_RESPONSE

The purpose of the **SIG\_AMP\_GAP\_AUTHORIZE\_RESPONSE** command signal is to respond to an authorization request event signal. The **SIG\_AMP\_GAP\_AUTHORIZE\_RESPONSE** command signal is sent by the application when an authorization request event signal called **SIG\_AMP\_GAP\_AUTHORIZE\_REQUEST** comes to the application. The application uses this when it has turned on the authorization in the service level security mode.

Signal Structure Type Name: tSIG\_AMPGapAuthorizeResponse

Signal Number	Parameters
SIG_AMP_GAP_AUTHORIZED_RESPONSE	Transac BdAddr Response

#### 4.1.7.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the Application.

##### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the device for which authorization is sent.

##### Response

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Authorization response. AMP_GAP_AUTHORIZE (0x00) – Authorized Non Zero – Not Authorized

#### 4.1.7.2 Event(s) Generated

None.

#### 4.1.8 SIG\_AMP\_GAP\_BOND

The purpose of the **SIG\_AMP\_GAP\_BOND** command signal is to initiate the bonding procedure with a remote device. The bonding procedure bonds the local device with a remote device. Later, when the link key is created during the bonding process, they can talk securely with each other.

Signal Structure Type Name: tSIG\_AMPGapBond

Signal Number	Parameters
SIG_AMP_GAP_BOND	Transac BdAddr PinLength Pin ServiceTrustLevel

#### 4.1.8.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context Information From the Application

##### BdAddr

Size: 6 Bytes

Value	Parameter Description
0XXXXXXXXXXXXX (6 Bytes)	Value: a structure with one member called Byte that is an array of 6 bytes and the index 0 being the least significant byte of the BD-Addr; this is the bonded remote device's BD_Addr

##### PinLength

Size: 1 Byte

Value	Parameter Description
0XX (1 Byte)	Length of the provided PIN code to be used in bonding; can be any value between 1 and 16.

##### Pin

Size: 16 Bytes

Value	Parameter Description
0XXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXX (16 Bytes)	Value: an array of 16 bytes and the index 0 is the most significant byte of the PIN; this is the PIN code used in bonding; not all bytes need to be filled; it is enough to fill the array up to the PinLength bytes.

##### ServiceTrustLevel

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 bytes)	Indication of which services are trusted; it is a bitwise OR of all the trusted services.

#### 4.1.8.2 Event(s) Generated

Three event signals are generated for the **SIG\_AMP\_GAP\_BOND** command. **SIG\_AMP\_GAP\_RESULT** is returned immediately and indicates successful acceptance of the command.

**SIG\_AMP\_GAP\_LINK\_KEY\_NOTIFY** is returned after bonding is successfully completed.

**SIG\_AMP\_GAP\_AUTH\_COMPLETE** is returned if there is any failure. It is returned with an error status.

One of the following **SIG\_AMP\_GAP\_RESULT** result codes is possible:

AMP\_RESULT\_PENDING (0x0001) – When the command is successfully accepted.

AMP\_RESULT\_RESULT\_FAIL (0x0014) – When the command failed for any reason.

## 4.2 GAP Event Signals

The following list contains all GAP signals going from the Amp'edUP stack to an embedded application.

SIG\_AMP\_GAP\_RESULT  
 SIG\_AMP\_GAP\_INQUIRY\_RESULT  
 SIG\_AMP\_GAP\_REMOTE\_NAME\_CON  
 SIG\_AMP\_GAP\_PIN\_CODE\_REQUEST  
 SIG\_AMP\_GAP\_AUTHORIZE\_REQUEST  
 SIG\_AMP\_GAP\_ABORT\_SECURITY  
 SIG\_AMP\_GAP\_AUTHORIZE\_COMPLETE  
 SIG\_AMP\_GAP\_LINK\_KEY\_NOTIFY

### 4.2.1 SIG\_AMP\_GAP\_RESULT

The purpose of the **SIG\_AMP\_GAP\_RESULT** event signal is to inform the application about the status of the most of the AMP command signals sent to AMP.

Signal Structure Type Name: tSIG\_AMPGapResult

Signal Number	Parameters
SIG_AMP_GAP_RESULT	Transac Result Code

#### 4.2.1.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	Context information from the application in the command signal for which this signal is sent as a response.

##### ResultCode

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes) (only the lower 2 bytes)	Result of the operation.

Value	Parameter Description
are used)	

#### 4.2.2 SIG\_AMP\_GAP\_INQUIRY\_RESULT

The purpose of the **SIG\_AMP\_GAP\_INQUIRY\_RESULT** event signal is to inform the embedded application about the inquiry results received during the inquiry procedure.

Signal Structure Type Name: tSIG\_AMPGapInquiryResult

Signal Number	Parameters
SIG_AMP_GAP_INQUIRY_RESULT	Transac BdAddr PageScanRepMode PageScanPeriodMode PageScanMode Cod ClockOffset

##### 4.2.2.1 Signal Parameters

###### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the application SIG_AMP_GAP_INQUIRY_REQ command signal

###### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the discovered device.

###### PageScanRepMode

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Page scan repetition mode of the discovered device.

###### PageScanPeriodMode

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Page scan period mode of the discovered device.



### PageScanMode

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Page scan mode of the discovered device.

### Cod

Size: 4 Bytes

Value	Parameter Description
0XXXXXX (Least significant 3 Bytes)	Class of device of the discovered device.
0xXX (Most Significant Byte)	Not used

### ClockOffset

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Clock offset of the discovered device.

## 4.2.3 SIG\_AMP\_GAP\_REMOTE\_NAME\_CON

The purpose of the **SIG\_AMP\_GAP\_REMOTE\_NAME\_CON** event signal is to give the name of the remote device to the embedded application in response to a **SIG\_AMP\_GAP\_REMOTE\_NAME\_REQ** command signal.

Signal Structure Type Name: tSIG\_AMPGapRemoteNameCon

Signal Number	Parameters
SIG_AMP_GAP_REMOTE_NAM_CON	Transac BdAddr ResultCode Name

### 4.2.3.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Context information from the Application in SIG_AMP_GAP_REMOTE_NAME_REQ command signal

### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device that has responded with its name

### ResultCode

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	Result status AMP_RESULT_DONE (0x0000) – When the remote name is successfully received from the remote device. AMP_RESULT_NO_CONNECTION (0x0004) – when a connection could not be established with the remote device for some reason. Name should be ignored in this case.

### Name

Size: 248<sup>1</sup> Bytes

Value	Parameter Description
0xFF...FF (An array of 20 Bytes)	Name of the remote device; null terminated.

## 4.2.4 SIG\_AMP\_GAP\_PIN\_CODE\_REQUEST

The purpose of the **SIG\_AMP\_GAP\_PIN\_CODE\_REQUEST** event signal is to request the application to provide it with the PIN code to connect with a remote device with a specified BD\_Addr. Upon receiving this PIN code, the application must respond with the command signal **SIG\_AMP\_GAP\_PINCODE\_REPLY**.

Signal Structure Type Name: tSIG\_AMPGapPinCodeRequest

Signal Number	Parameters
SIG_AMP_GAP_PIN_CODE_REQUEST	Transac BdAddr

### 4.2.4.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	Context information from AMP. Usually it is set to 0.

<sup>1</sup> Currently only 100 bytes are supported.

**BdAddr**

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device address of the remote device for which the PIN code is required.

**4.2.5 SIG\_AMP\_GAP\_AUTHORIZE\_REQUEST**

The purpose of the **SIG\_AMP\_GAP\_AUTHORIZE\_REQUEST** event signal is to send a response to the application when an authorization response is needed from the application to connect to a specific service - when the service level security mode needs authorization from the application. (This is set by the application in the **SIG\_AMP\_GAP\_SECURITY\_SETUP** command). The application must respond with the command signal called **SIG\_AMP\_GAP\_AUTHORIZE\_RESPONSE**.

Signal Structure Type Name: tSIG\_AMPGapAuthorizeRequest

Signal Number	Parameters
SIG_AMP_GAP_AUTHORIZE_REQUEST	Transac ServiceName BdAddr

**4.2.5.1 Signal Parameters****Transac**

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from Amp'edUP. Usually set to 0.

**ServiceName**

Size: 26 Bytes

Value	Parameter Description
An array of 6 unsigned bytes.	The service name to which a connection is made that requires authorization. The last byte has to be NULL character.

**BdAddr**

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device that provides the service for which an authorization is required.

## 4.2.6 SIG\_AMP\_GAP\_ABORT\_SECURITY

The purpose of the **SIG\_AMP\_GAP\_ABORT\_SECURITY** event signal is to send notification that the link has gone down after issuing a pin code or authorization request event to application.

Signal Structure Type Name: tSIG\_AMPGapAbortSecurity

Signal Number	Parameters
SIG_AMP_GAP_ABORT_SECURITY	Transac BdAddr

### 4.2.6.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the Amp'edUP. Usually set to 0.

#### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device with which authentication procedure is being performed.

## 4.2.7 SIG\_AMP\_GAP\_AUTH\_COMPLETE

The purpose of the **SIG\_AMP\_GAP\_AUTH\_COMPLETE** event signal is to inform that a connection's authentication is completed.

Signal Structure Type Name: tSIG\_AMPGapAuthComplete

Signal Number	Parameters
SIG_AMP_GAP_AUTH_COMPLETE	Transac BdAddr Status

### 4.2.7.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the AMP. Usually set to 0.

**BdAddr**

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device with which authentication is completed.

**Status**

Size: 2 Bytes

Value	Parameter Description
0xXXXX (2 Bytes)	The result of the authentication. AMP_GAP_AUTH_SUCCESS (0x0000) – Success Other values – Indicate failure. Refer to the <i>HCI Reference Guide</i> for a list of Error codes.

**4.2.8 SIG\_AMP\_GAP\_LINK\_KEY\_NOTIFY**

The purpose of the **SIG\_AMP\_GAP\_LINK\_KEY\_NOTIFY** event signal is to notify the application that a bonding procedure (with the remote device) is finished.

Irrespective of who initiated the procedure, both the local device and the bonded remote device will send this event to the application. The event indicates what the link key is that resulted from the bonding procedure as well as the BD Address of the device that was bonded and the key type used to create the Link key.

Signal Structure Type Name: tSIG\_AMPGapLinkKeyNotify

Signal Number	Parameters
SIG_AMP_GAP_LINK_KEY_NOTIFY	Transac LinkKey BdAddr KeyType

**4.2.8.1 Signal Parameters**

**Transac**

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the application in the command signal for which this signal is sent as a response.

**LinkKey**

Size: 16 Bytes

Value	Parameter Description
0XXXXXXXXXXXXXXXXXX	Array of 16 unsigned bytes; provides the created Link Key as a result of the

Value	Parameter Description
XXXXXXXXXXXXXXXXXXXX (16 Bytes)	bonding.

### BdAddr

Size: 6 Bytes

Value	Parameter Description
0XXXXXXXXXXXX (6 Bytes)	Value: a structure with one member called Byte that is an array of 6 bytes with the index 0 being the least significant byte of the BD_Addr.

### KeyType

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Key type used in the bonding procedure: 0 – combination key 1 – local key 2 – remote unit key

## 4.3 GAP API Functions

Instead of signals, two API functions are defined in the Amp'edUP interface to set device parameters. The reason they are defined as API functions instead of signals is because no messages need to come from the remote device in order to initiate these activities.

The two API functions allow for setting the local name and for turning on the pairable mode. The following two sections detail the function calls.

### 4.3.1 Setting Local Name

```
uint32 AMP_API_Gap_Set_Local_Device_Name(char *pName)
```

#### **Description:**

Sets the local name of the device. This is the name that will be reported when Remote devices request for Remote name.

#### **Parameters:**

pName - the pointer to the device name string, up to 20 characters, that is terminated by a NULL character.

#### **Return Value:**

An unsigned 4-byte integer is returned as the result of the operation.

AMP\_RESULT\_SUCCESS - if successful

AMP\_RESULT\_PARAMETER\_ERROR - if a failure occurs because of a parameter error (pointer to the name is NULL)

AMP\_RESULT\_FAIL - if a failure occurs for any other reason

### 4.3.2 Setting the Pairable Mode

```
uint32 AMP_API_Gap_Set_Pairable_Mode(uint8 PairableMode)
```

#### **Description:**

Puts the device into pairable mode, which allows some remote devices to bond with this device.

#### **Parameters:**

Pairable mode - an unsigned byte; if it is 0, then put the device into non-pairable mode; if it is 1, then put the device in pairable mode.

#### **Return Value:**

An unsigned 4-byte integer is returned as the result of the operation.

AMP\_RESULT\_SUCCESS - if successful

AMP\_RESULT\_FAIL - if a failure occurs for any other reason

## 5 SPP Signals

The functionalities provided by the command and event signals in SPP are:

- Connection and disconnection of two devices,
- Sending serial data over the RF link,
- Receiving serial data over the RF link,
- Indicating status of the SPP connection/disconnection/data transfer.

For more information about the terms used in this section, please refer to the [Bluetooth Serial Port Profile V1.1](#).

### 5.1 SPP Command Signals

The following list contains all SPP signals going from an embedded application to the AMP.

- SIG\_AMP\_SPP\_LISTEN\_REQ
- SIG\_AMP\_SPP\_CONNECT\_REQ
- SIG\_AMP\_SPP\_SEND
- SIG\_AMP\_SPP\_DISCONNECT\_REQ

#### 5.1.1 SIG\_AMP\_SPP\_LISTEN\_REQ

The purpose of this command signal is to allow the application to start the SPP server and put it into listen mode. This would also add the SPP service information in the SDP database, so that SPP clients can perform SDP queries in order to get the service name and connect to the SPP server.

Signal Structure Type Name: tSIG\_AMP\_SppListenRequest

Signal Number	Parameters
SIG_AMP_SPP_LISTEN_REQ	Transac ServiceName PortName

#### 5.1.1.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the application

##### ServiceName

Size: 26 Bytes

Value	Parameter Description
-------	-----------------------



Value	Parameter Description
An array of 6 Bytes	Service name for the serial port service. The last character must be NULL.

**PortName**

Size: 6 Bytes

Value	Parameter Description
An array of 6 Bytes	Local port name for identification. Example: Port1. The last character has to be NULL.

**5.1.1.2 Event(s) Generated**

When the **SIG\_AMP\_SPP\_LISTEN\_REQ** command is finished, a **SIG\_AMP\_SPP\_SETUP\_RESULT** event signal will be sent to the application if there is any error. For a success, no event is generated. Additionally, **SIG\_AMP\_LISTEN\_CON** arrives for connection information.

AMP\_RESULT\_NO\_RESOURCES (0x0007) – When there are no resources (memory)

AMP\_RESULT\_PORT\_EXIST (0x0008) – When a port by that name already exists

AMP\_RESULT\_SERVICE\_EXISTS (0x0009) – When a service by that name already exists

AMP\_RESULT\_PARAMETER\_ERROR (0x0002) – In all other cases.

**5.1.2 SIG\_AMP\_SPP\_CONNECT\_REQ**

The purpose of this command signal is to inform that the application is behaving like the SPP client and is trying to connect to the SPP server. The server resides in the BD\_Addr of the device that was specified by the client under the service name.

Signal Structure Type Name: tSIG\_AMPSPpConnectRequest

Signal Number	Parameters
SIG_AMP_SPP_CONNECT_REQ	Transac ServiceName PortName BdAddr

**5.1.2.1 Signal Parameters**

**Transac**

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	Context information from the application

### ServiceName

Size: 26 Bytes

Value	Parameter Description
An array of 6 Bytes	Serial Port Service Name in the SPP server to which the client is attempting to connect. The last character must be NULL. A NULL field, will connect to the first service name found.

### PortName

Size: 6 Bytes

Value	Parameter Description
An array of 6 Bytes	Port Name for local identification of the local serial port. The last character has to be NULL.

### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote server.

#### 5.1.2.2 Event(s) Generated

When the **SIG\_AMP\_SPP\_CONNECT\_REQ** command has been accepted successfully (No errors), then no event would be generated immediately. But if there are errors in accepting the command and connecting to the server, a **SIG\_AMP\_SPP\_CONNECT\_CON** event signal will be sent to the application with an Invalid Handle and result code, which can be one of the following.

- AMP\_RESULT\_NO\_RESOURCES (0x0007) – When there are no resources (memory)
- AMP\_RESULT\_PORT\_EXISTS (0x0008) – When a port by that name already exists
- AMP\_RESULT\_SERVICE\_EXISTS (0x0009) – When a connection to that service already exists
- AMP\_RESULT\_PARAMETER\_ERROR (0x0002) – In all other cases

When the connect request command is completed, **SIG\_AMP\_SPP\_CONNECT\_CON** event signal would be sent to the application with a valid port handle (2 Bytes), if successful, otherwise with an invalid handle. Also the status would be indicated. It can be one of the following.

- AMP\_RESULT\_CONNECTION\_UP – When the SPP connection is up and running
- AMP\_RESULT\_CONNECTION\_DOWN – When the SPP connection cannot be established

### 5.1.3 SIG\_AMP\_SPP\_SEND

The purpose of this command signal is to allow the application to send data over the established SPP connection. In order to send data with zero copying, it is mandatory that the data-to-be-sent is present in a RAM buffer. There is an SPP API function that needs to be called to get the memory allocated in the RAM.

```
tBufferHeader *AllocBufferToSend (uint16 Length)
```

“Length” is the size of the data that is to be sent, with a maximum size of 321 bytes for the SPP profile. (The iOS, IAP profile maximum size is 314 bytes) The return value is a pointer to the structure tBufferHeader.

```
typedef struct tBufferHeader {
    uint8 *pData;          // Can write data here
    uint32 Transac;
}tBufferHeader;
```

The member uint8 \*pData in the structure points to the exact place where the application can start writing its data.

Signal Structure Type Name: tSIG\_AMP\_SppSend

Signal Number	Parameters
SIG_AMP_SPP_SEND	Transac PortHandle ByteSend DataHdr

#### 5.1.3.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the application

##### PortHandle

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Port handle returned in the events SIG_AMP_SPP_CONNECT_CON in the client or SIG_AMP_SPP_LISTEN_CON in the server.

### BytesSend

Size: 2 Bytes

Value	Parameter Description
0xFFFF (2 Bytes)	Number of bytes to send (size of the data). SPP max 321 bytes, IAP max 314 bytes

### DataHdr

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	Pointer to the tBufferHeader structure returned by AllocBufferToSend

#### 5.1.3.2 Event(s) Generated

When the **SIG\_AMP\_SPP\_SEND** command is finished successfully, no event signal will be sent to the embedded application. Only if there were any error in sending the data, **SIG\_AMP\_SPP\_CONNECT\_STATUS** event signal would be generated and sent to the application with one of the following status codes:

AMP\_RESULT\_NO\_CONNECTION (0x0004) – When the port handle is bad or if there is no connection.

AMP\_RESULT\_LINE\_ERROR (0x0006) – When there is some error with the connection

AMP\_DATA\_BUFFER\_HEADER\_CORRUPTED (0x0015) – When the Buffer header is corrupted.

AMP\_DATA\_LENGTH\_ERROR (0x0016) – When the data buffer length is greater than what the lower layers can handle (should match the size that is less than or equal to (L2CAP MTU – Header size of RFCOMM Header). These constants would be defined in the target header file that comes with the development code.

#### 5.1.4 SIG\_AMP\_SPP\_DISCONNECT\_REQ

The purpose of this command signal sent by the application is to disconnect the SPP connection on the port identified by the port name and to remove the port, both in the server and the client.

Signal Structure Type Name: tSIG\_AMP\_SppDisconnectRequest

Signal Number	Parameters
SIG_AMP_SPP_DISCONNECT_REQ	Transac PortName SppRole

##### 5.1.4.1 Signal Parameters

### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the application

### PortName

Size: 6 Bytes

Value	Parameter Description
An array of 6 Bytes	Port Name (used for local identification) of the serial port. The last character has to be NULL.

### SppRole

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Role client = 0, Role server = 1

#### 5.1.4.2 Event(s) Generated

There are two event signals possible for this command.

When the **SIG\_AMP\_SPP\_DISCONNECT\_REQ** command is finished, event signals called **SIG\_AMP\_SPP\_CONNECT\_CON** on the client side and **SIG\_AMP\_SPP\_LISTEN\_CON** on the server side will be sent to the application indicating the result that the connection is down or up with the error codes. One of the following result codes is possible.

AMP\_RESULT\_CONNECTION\_DOWN (0x000c) – When the SPP disconnect succeeded.

In the case of the command being queued the client would get **SIG\_AMP\_SPP\_DISCONNECT\_CON** with the status AMP\_RESULT\_PENDING (0x0001).

## 5.2 SPP Event Signals

The following list contains all SPP signals going from the AMP to an embedded application.

SIG\_AMP\_SPP\_RECV  
 SIG\_AMP\_SPP\_LISTEN\_CON  
 SIG\_AMP\_SPP\_CONNECT\_CON  
 SIG\_AMP\_SPP\_CONNECT\_STATUS  
 SIG\_AMP\_SPP\_DISCONNECT\_CON

### 5.2.1 SIG\_AMP\_SPP\_RECV

The purpose of this event signal is to inform the application about any data that has been received at the port from the remote device. The responsibility of releasing the data buffer memory (pDataHdr) is with the application. The application must call the following API function with the parameter pDataHdr after consuming the received data.

```
void OS_Release (tBufferHeader* pDataHdr)
```

pDataHdr is of type tBufferHeader\*. The member uint8 \*pData in the tBufferHeader structure points to the exact place where the application can start reading the received data.

```
typedef struct tBufferHeader {
    uint8 *pData;        // Read data here
    uint32 Transac;
}tBufferHeader;
```

Signal Structure Type Name: tSIG\_AMPSPpRecv

Signal ID	Parameters
SIG_AMP_SPP_RECV	Transac PortHandle, ByteRead, pDataHdr

#### 5.2.1.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	Context information. Usually set to 0.

##### PortHandle

Size: 2 Bytes

Value	Parameter Description
0xFFFF (2 Bytes)	Port handle returned in the events SIG_AMP_SPP_CONNECT_CON in the client or SIG_AMP_SPP_LISTEN_CON in the server.

##### BytesRead

Size: 2 Bytes

Value	Parameter Description
-------	-----------------------

Value	Parameter Description
0XXXXX (2 Bytes)	Number of bytes read by embedded application

### pDataHdr

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes) (of type tBufferHeader*)	Pointer to data buffer header. The data is present inside the structure pointed to by this pointer, in the member pData.

## 5.2.2 SIG\_AMP\_SPP\_LISTEN\_CON

The purpose of this event signal is to inform the SPP server application when a client successfully connects/disconnects to it.

Signal Structure Type Name: tSIG\_AMP\_SppListenCon

Signal ID	Parameters
SIG_AMP_LISTEN_CON	Transac PortName Handle Status BdAddr

### 5.2.2.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Context information from the application

#### PortName

Size: 6 Bytes

Value	Parameter Description
An array of 6 Bytes	Port Name (used for local identification) of the serial port. The last character has to be NULL character.

#### Handle

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Port handle to be used to send data from the Server. Application must save this information.

**Status**

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Result of the SPP connection AMP_RESULT_CONNECTION_UP – When the Client is connected successfully or disconnect failed. AMP_RESULT_CONNECTION_DOWN – When the connection to client drops for some reason or the successful disconnect happened AMP_RESULT_CONNECTION_ERROR When there is some error while creating the client connection.

**BdAddr**

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device.

**5.2.3 SIG\_AMP\_SPP\_CONNECT\_CON**

The purpose of this event signal is to inform the SPP client application about the result of the connection which is sent in response to connect as well as disconnect.

Signal Structure Type Name: tSIG\_AMP\_SppConnectCon

Signal ID	Parameters
SIG_AMP_SPP_CONNECT_CON	Transac PortName Handle Status BdAddr

**5.2.3.1 Signal Parameters****Transac**

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the application

**PortName**

Size: 6 Bytes

Value	Parameter Description
An array of 6 bytes	Port Name (used for local identification) of the serial port. The last character has to be NULL character



**Handle**

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Port handle to be used to send data in the Client. Application must save this information.

**Status**

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Result of the SPP connection AMP_RESULT_CONNECTION_UP – When the Client is connected successfully. Or if the disconnect failed. AMP_RESULT_CONNECTION_DOWN – When the connection to server drops for some reason or if disconnected successfully.

**BdAddr**

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device.

**5.2.4 SIG\_AMP\_SPP\_CONNECT\_STATUS**

The purpose of this event signal is to inform the application about the success/failure of data transfer.

Signal Structure Type Name: tSIG\_AMPSPpConnectStatus

Signal ID	Parameters
SIG_AMP_SPP_CONNECT_STATUS	Transac Handle Status

**5.2.4.1 Signal Parameters****Transac**

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Context information from the application

**Handle**

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Port handle used to send data.

**Status**

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Result of the Data transfer: AMP_RESULT_NO_CONNECTION – When the port handle is bad or if there is no connection. AMP_RESULT_LINE_ERROR – When there is some error with the connection AMP_DATA_BUFFER_HEADER_CORRUPTED – When the Buffer header is corrupted. AMP_DATA_LENGTH_ERROR – When the data buffer length is greater than what the lower layers can handle (should match the size that is less than or equal to [L2CAP MTU – Header size of RFCOMM Header] )

**5.2.5 SIG\_AMP\_SPP\_DISCONNECT\_CON**

The purpose of this event signal is to inform the client application that the SPP connection is going to be disconnected or there is any error in disconnect request.

Signal Structure Type Name: tSIG\_AMPSPpDisconnectCon

Signal ID	Parameters
SIG_AMP_SPP_DISCONNECT_CON	Transac PortName Status

**5.2.5.1 Signal Parameters**

**Transac**

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the application

**PortName**

Size: 6 Bytes

Value	Parameter Description
An array of 6 bytes	Port Name (used for local identification) of the serial port. The last character has to be NULL character

**Status**

Size: 2 Bytes

Value	Parameter Description
0XXXXX (Lower 2 Bytes)	Status of the SPP Disconnect AMP_RESULT_PENDING – When the disconnect is pending AMP_RESULT_PARAMETER_ERROR – When the port name given to disconnect is wrong. AMP_RESULT_DONE – If the disconnect happened already.

## 6 SDAP Signals

The command and event signals for SDAP provide the functionality to do service searches based on attribute IDs. For more information about the terms used in this section, please refer to the [Bluetooth Service Discovery Application Profile V1.1](#).

### 6.1 SDAP Command Signals

The following list contains all SDAP signals going from an embedded application to the AMP.

SIG\_AMP\_SDAP\_SERVICE\_SEARCH\_REQ

#### 6.1.1 SIG\_AMP\_SDAP\_SERVICE\_SEARCH\_REQ

The purpose of this AMP command signal is to search for a specific service available in the remote device by specifying the BDAAddr and Service class. This would return the Service Name or the Service Channel Number based on the Attribute Id.

Signal Structure Type Name: tSIG\_AMPsdapServiceSearchReq

Signal ID	Parameters
SIG_AMP_SDAP_SERVICE_SEARCH_REQ	Transac BdAddr[] ServiceClass[] Attributeld

##### 6.1.1.1 Signal Parameters

###### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context Information From the Application

###### BdAddr[]

Size: An array of 6 Bytes

Value	Parameter Description
Array of structures with one Member called Byte, which is an array of 6 bytes. The index 0 being the LS byte of the Bd Address.	Bluetooth device Address of the server to connect to find the services. Currently only one BdAddr is supported in this command, though it is an array.

###### ServiceClass[]

Size: An array of 2 Bytes

Value	Parameter Description
An array of unsigned 2	(UUID) Service class of the services that the SDAP client is interested in. Currently only one Service Class can be mentioned. Currently supported

Value	Parameter Description
byte numbers.	values are: AMP_UUID_SERVCLASS_PUBLIC_BROWSE_GROUP – 0x1002 AMP_UUID_SERVCLASS_SERIAL_PORT – 0x1101

### Attributeld

Size: 2 Bytes

Value	Parameter Description
0xXXXX (2 Bytes)	The attribute ID, like Service Name or Server Channel Number, for the service that is queried. Currently supported attributes are: AMP_SDAP_ATTR_PROTOC_DESC_LIST (0x0004) – Protocol Descriptor list that includes the Server Channel Number of the service AMP_SDAP_ATTR_NAME (0x0100) – Name of the service For more values, refer to <i>Bluetooth Specification v 1.1</i> .

#### 6.1.1.2 Event(s) Generated

There are multiple events generated for this command signal.

When the Service Search Request command is accepted, the event **SIG\_AMP\_SDAP\_RESULT** is generated. This event indicates the result about whether the service search is started or not. One of the following result codes is possible:

AMP\_RESULT\_PENDING (0x0001) – When the SDAP Service search request is sent to the other device successfully.

AMP\_RESULT\_PARAMETER\_ERROR (0x0002) – When the SDAP Service search request is not sent due to some parameter error.

When the remote device answers the service search request, the event signal called **SIG\_AMP\_SDAP\_SERVICE\_SEARCH\_CON** is returned to the application with the search results. One signal per service record retrieved is sent to the application.

When all the search results are finished and sent, the event signal called **SIG\_AMP\_GAP\_RESULT** is sent to the application with the result code AMP\_RESULT\_DONE (0x0000). When the service search fails and returns no records then the event signal **SIG\_AMP\_GAP\_RESULT** is sent to the application with the result code AMP\_RESULT\_FAIL (0x0014).

## 6.2 SDAP Event Signals

The following list contains all SDAP signals going from the AMP to an embedded application.

SIG\_AMP\_SDAP\_RESULT  
SIG\_AMP\_SDAP\_SERVICE\_SEARCH\_CON

### 6.2.1 SIG\_AMP\_SDAP\_SERVICE\_SEARCH\_CON

The purpose of this event signal is to return to the application every service record that is received by the AMP in response to the service search request sent. This signal contains all the fields of the service record.

Signal Structure Type Name: tSIG\_AMPsdapServiceSearchCon

Signal ID	Parameters
SIG_AMP_SDAP_SERVICE_SEARCH_CON	Transac Status BdAddr ServiceClass DeviceClass MajorDeviceClass MinorDeviceClass NumberOfProtocolDescriptor ProtocolDescriptor L2CapPsm RfcommScn ServiceName NumberSupportedFormats SupportedFormats

#### 6.2.1.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Context information from the application in the SIG_AMP_SDAP_SERVICE_SEARCH_REQ signal.

##### Status

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Result Status of Service Search. Only in the case of success would this signal be sent to the application, so it would always have the value of AMP_RESULT_DONE (0x0000) as the status.

##### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the device with the service that is inquired for.

##### ServiceClass

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Service Class of the Service. Not Implemented. Reserved For Future Use.

### DeviceClass

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Device Class of the device where the service resides. Not Implemented. Reserved For Future Use.

### MajorDeviceClass

Size: 2 Bytes

Value	Parameter Description
0xXX (1 byte)	Major Device Class of the device where the service resides. Not Implemented. Reserved For Future Use.

### MinorDeviceClass

Size: 1 Byte

Value	Parameter Description
0xXX (1 byte)	Minor Device Class of the device where the service resides. Not Implemented. Reserved For Future Use.

### NumberOfProtocolDescriptor

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Number of Protocol Descriptors returned in this service record if the Service search is performed for AttributeId 0x0004 (Protocol Descriptor). Otherwise undefined.

### ProtocolDescriptor

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Pointer to the Protocol Descriptors returned in this service record if the Service search is performed for AttributeId 0x0004 (Protocol Descriptor). Not to be used currently.

### L2CapPsm

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	L2CAP PSM for this service if the service search is performed for AttributeId 0x0004 (Protocol Descriptor) and the protocol descriptor lists L2Cap as one of the protocols. Otherwise undefined.

### RfcommScn

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	RFCOMM Server channel number for this service if the service search is performed for AttributeId 0x0004 (Protocol Descriptor) and the protocol descriptor lists RFCOMM as one of the protocols. Otherwise undefined.

### ServiceName

Size: An array of 6 bytes

Value	Parameter Description
An array of 6 unsigned bytes.	Name for the service, if the service search is performed for AttributeId 0x0100 (Service Name). Otherwise all bytes will have 0xFF. The name would be NULL terminated if valid.

### NumberSupportedFormats

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Number of Data formats supported by the server. If this is returned in the service search response, the number is copied onto this parameter; else, it would have 0.

### SupportedFormats

Size: An array of 8 bytes

Value	Parameter Description
An array of 8 unsigned bytes.	The supported formats if NumberSupportedFormats is not 0. Otherwise, all bytes will have 0xFF.



## 7 Miscellaneous Signals

### 7.1 Miscellaneous Command Signals

The following list contains miscellaneous command signals going from the application to the Amp'edUP stack:

SIG\_AMP\_CREATE\_DEVICE\_DATABASE  
 SIG\_AMP\_DELETE\_DEVICE\_DATABASE  
 SIG\_AMP\_READ\_DEVICE\_DATABASE  
 SIG\_AMP\_HOLD\_MODE  
 SIG\_AMP\_SNIFF\_MODE  
 SIG\_AMP\_EXIT\_SNIFF\_MODE

#### 7.1.1 SIG\_AMP\_CREATE\_DEVICE\_DATABASE

The purpose of the **SIG\_AMP\_CREATE\_DEVICE\_DATABASE** command signal is to add a known remote device's security information to the security manager's database stored in non-volatile memory.

Signal Structure Type Name: tSIG\_AMPCreateDeviceDatabase

Signal ID	Parameters
SIG_AMP_CREATE_DEVICE_DATABASE	Transac DeviceClass DeviceName TrustLevel LinkKey BdAddr Flag

#### 7.1.1.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the Application

##### DeviceClass

Size: 3 Bytes

Value	Parameter Description
An array of 3 Unsigned bytes.	Class of Device information of the device for which link key is stored.

### DeviceName

Size: 6 Bytes

Value	Parameter Description
An array of 6 Unsigned bytes.	Local name of the device for which link key is stored.

### TrustLevel

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	A bitwise OR of services that do not require authorization. If this is not used, then 0 can be sent.

### LinkKey

Size: 16 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 16 bytes with the index 0 being the LS byte of the Link Key and 15 being the MS byte	Link Key information of the device.

### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the device whose security information is added.

### Flag

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Flag for device data base table operations AMP_FLASHFLAG_DEVICEDB_NONE (0x00) – no operation AMP_FLASHFLAG_DEVICEDB_OVERWRITE (0x01) (not supported) AMP_FLASHFLAG_DEVICEDB_TRUNCATED (0x02) (not supported) AMP_FLASHFLAG_DELETE_BDADDR (0x04) – deletes the bd address from the previous parameter AMP_FLASHFLAG_DELETE_ALL (0x08) – deletes the entire table

#### 7.1.1.2 Event(s) Generated

When the **SIG\_AMP\_CREATE\_DEVICE\_DATABASE** command is completed, a **SIG\_AMP\_GAP\_RESULT** event signal will be sent to the application indicating the result. One of the following result codes is possible:

AMP\_RESULT\_DONE (0x0000) – When the device's security information is added to the database successfully.

AMP\_RESULT\_NO\_RESOURCES (0x0007) – When the device's security information cannot be added due to no resources.

### 7.1.2 SIG\_AMP\_DELETE\_DEVICE\_DATABASE

The purpose of the **SIG\_AMP\_DELETE\_DEVICE\_DATABASE** command signal is to remove the security information record associated with the specified device if found in the database.

Signal Structure Type Name: tSIG\_AMPDeleteDeviceDatabase

Signal Number	Parameters
SIG_AMP_DELETER_DEVICE_DATABASE	Transac BdAddr Flag

#### 7.1.2.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context information from the Application.

##### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the device whose security information is requested to be deleted.

##### Flag

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Flag for device data base table operations AMP_FLASHFLAG_DEVICEDB_NONE (0x00) – no operation AMP_FLASHFLAG_DEVICEDB_OVERWRITE (0x01) (not supported) AMP_FLASHFLAG_DEVICEDB_TRUNCATED (0x02) (not supported) AMP_FLASHFLAG_DELETE_BDADDR (0x04) – deletes the bd address from the previous parameter AMP_FLASHFLAG_DELETE_ALL (0x08) – deletes the entire table

#### 7.1.2.2 Event(s) Generated

When the **SIG\_AMP\_DELETE\_DEVICE\_DATABASE** command is completed, a **SIG\_AMP\_GAP\_RESULT** event signal will be sent to the application indicating the result. One of the following result codes is possible:

AMP\_RESULT\_DONE (0x0000) – When the device’s security information is deleted from the database successfully.

AMP\_RESULT\_PARAMETER\_ERROR (0x0002) – When the device’s security information is missing in the Database.

### 7.1.3 SIG\_AMP\_READ\_DEVICE\_DATABASE

The purpose of the **SIG\_AMP\_READ\_DEVICE\_DATABASE** signal is to read the local device’s security manager’s database stored in non-volatile memory.

Signal Structure Type Name: tSIG\_AMPReadDeviceDatabase

Signal ID	Parameters
SIG_AMP_READ_DEVICE_DATABASE	Transac

#### 7.1.3.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context Information From the Application

#### 7.1.3.2 Event(s) Generated

When a device is read from the database, its information is sent via the event **SIG\_AMP\_GAP\_DEVICE\_DB\_RESULT**.

When the command is completed, a **SIG\_AMP\_GAP\_RESULT** event signal will be sent to the application indicating the result. One of the following result codes is possible:

AMP\_RESULT\_DONE (0x0000) – When the device database is able to be read successfully.

AMP\_RESULT\_PARAMETER\_ERROR (0x0002) – When the device database cannot be read successfully.

### 7.1.4 SIG\_AMP\_HOLD\_MODE

The purpose of the **SIG\_AMP\_HOLD\_MODE** command signal is to put the link into Hold mode. Refer to the Bluetooth specification for further details on possible parameters.

Signal Structure Type Name: tSIG\_AMPHoldMode

Signal ID	Parameters
SIG_AMP_HOLD_MODE	Transac BdAddr

Signal ID	Parameters
	HoldMinInterval HoldMaxInterval

#### 7.1.4.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Context Information From the Application

##### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device that will be put into hold mode.

##### HoldMinInterval

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	The minimum negotiable hold interval in slots.

##### HoldMaxInterval

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	The maximum negotiable hold interval in slots.

#### 7.1.4.2 Event(s) Generated

When the Hold mode request has been completed, the event **SIG\_AMP\_GAP\_RESULT** is generated. This event indicates the result of the Hold request. One of the following result codes is possible:

AMP\_RESULT\_PENDING (0x0001) – When Hold mode has successfully been requested.

AMP\_MODE\_NOT\_SUPPORTED (0x001C) – When Hold mode is not supported.

AMP\_RESULT\_NO\_RESOURCES (0x0007) – When Hold mode is not currently available.

AMP\_WRONG\_BDADDR (0x001D) – When the wrong BD Address is sent.

AMP\_RESULT\_ILLEGAL\_OPERATION (0x0003) – When the Hold mode request has failed.

### 7.1.5 SIG\_AMP\_SNIFF\_MODE

The purpose of the **SIG\_AMP\_SNIFF\_MODE** command signal is to put the link into Sniff mode. Refer to the Bluetooth specification for further details on possible parameters.

Signal Structure Type Name: tSIG\_AMPsSniffMode

Signal ID	Parameters
SIG_AMP_SNIFF_MODE	Transac BdAddr SniffMinInterval SniffMaxInterval SniffAttempt SniffTmo

#### 7.1.5.1 Signal Parameters

##### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Context Information From the Application

##### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device that will be put into Sniff mode.

##### SniffMinInterval

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	The minimum negotiable Tsniff interval slots.

##### SniffMaxInterval

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	The maximum negotiable Tsniff interval slots.

##### SniffAttempt

Size: 2 Bytes

Value	Parameter Description
0xFFFF (2 Bytes)	The number of frames that the slave will listen for a matching AM Address packets after the Tsniff interval.

### SniffTmo

Size: 2 Bytes

Value	Parameter Description
0xFFFF (2 Bytes)	The number of frames that the slave will listen after a matching AM Address packets are no longer received.

### 7.1.5.2 Event(s) Generated

When the Sniff mode request has been completed, the event **SIG\_AMP\_GAP\_RESULT** is generated. This event indicates the result of the Sniff request. One of the following result codes is possible:

AMP\_RESULT\_PENDING (0x0001) – When Sniff mode has successfully been requested.

AMP\_MODE\_NOT\_SUPPORTED (0x001C) – When Sniff mode is not supported.

AMP\_RESULT\_NO\_RESOURCES (0x0007) – When Sniff mode is not currently available.

AMP\_WRONG\_BDADDR (0x001D) – When the wrong BD Address is sent.

AMP\_WRONG\_MODE (0x001F) – When Sniff mode is invalid in the link's present state.

AMP\_RESULT\_ILLEGAL\_OPERATION (0x0003) – When the Sniff mode request has failed.

### 7.1.6 SIG\_AMP\_EXIT\_SNIFF\_MODE

The purpose of the **SIG\_AMP\_EXIT\_SNIFF\_MODE** command signal is to put a link that is currently in Sniff mode back into Active mode.

Signal Structure Type Name: tSIG\_AMPExitSniffMode

Signal ID	Parameters
SIG_AMP_EXIT_SNIFF_MODE	Transac BdAddr

### 7.1.6.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	Context Information From the Application

### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device that will be put into Active mode.

#### 7.1.6.2 Event(s) Generated

When the Exit Sniff mode request has been completed, the event **SIG\_AMP\_GAP\_RESULT** is generated. This event indicates the result of the Exit Sniff request. One of the following result codes is possible:

AMP\_RESULT\_PENDING (0x0001) – When Exit Sniff mode has successfully been requested.

AMP\_RESULT\_NO\_RESOURCES (0x0007) – When Exit Sniff mode is not currently available.

AMP\_WRONG\_BDADDR (0x001D) – When the wrong BD Address is sent.

AMP\_WRONG\_MODE (0x001F) – When Exit Sniff mode is invalid in the link's present state.

AMP\_RESULT\_ILLEGAL\_OPERATION (0x0003) – When the Exit Sniff mode request has failed.

## 7.2 Miscellaneous Event Signals

The following list contains miscellaneous event signals going from the Amp'edUP stack to an embedded application:

SIG\_AMP\_GAP\_DEVICE\_DB\_RESULT

SIG\_AMP\_MODE\_CHANGE\_RESULT

SIG\_AMP\_GAP\_LL\_CONN\_DOWN

### 7.2.1 SIG\_AMP\_GAP\_DEVICE\_DB\_RESULT

The purpose of the **SIG\_AMP\_GAP\_DEVICE\_DB\_RESULT** signal is to return the local device's security manager's database entries in response to a read database request.

Signal Structure Type Name: tSIG\_AMPDeviceDBResult

Signal ID	Parameters
SIG_AMP_GAP_DEVICE_DB_RESULT	Transac DeviceClass DeviceName TrustLevel LinkKey BdAddr



### 7.2.1.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	Context Information from the Application.

#### DeviceClass

Size: 3 Bytes

Value	Parameter Description
Array of bytes	The Class of Device.

#### DeviceName

Size: 6 Bytes

Value	Parameter Description
Array of bytes	The first 6 bytes of the device name.

#### TrustLevel

Size: 4 Bytes

Value	Parameter Description
0xFFFFFFFF (4 Bytes)	32 bit mask of trusted services of this device.

#### LinkKey

Size: 16 Bytes

Value	Parameter Description
Structure containing 16 bytes	The encryption key for this device.

#### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the device listed in the security manager's device table.

### 7.2.2 SIG\_AMP\_MODE\_CHANGE\_RESULT

The purpose of the **SIG\_AMP\_MODE\_CHANGE\_RESULT** event signal is to inform the application of a change in a link's mode.

Signal Structure Type Name: tSIG\_AMPModeChangeResult

Signal ID	Parameters
SIG_AMP_MODE_CHANGE_RESULTS	Transac ResultCode BdAddr Mode

Signal ID	Parameters
	Interval

### 7.2.2.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Context Information from the Application.

#### ResultCode

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXX (4 Bytes)	Result: AMP_RESULT_DONE (0x00) – successful operation AMP_RESULT_ILLEGAL_OPERATION (0x03) – operation failed

#### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the device.

#### Mode

Size: 1 Byte

Value	Parameter Description
0xXX (1 Byte)	Contains the current mode of the device: ACTIVE_MODE - 0x00 HOLD_MODE - 0x01 SNIFF_MODE - 0x02 PARK_MODE - 0x03 (not supported) INVALID_MODE - 0xff

#### Interval

Size: 2 Bytes

Value	Parameter Description
0XXXXX (2 Bytes)	Not used for this signal.

### 7.2.3 SIG\_AMP\_GAP\_LL\_CONN\_DOWN

The purpose of the **SIG\_AMP\_GAP\_LL\_CONN\_DOWN** event signal is to inform the application that the lower layer stack has completely disconnected from the link.

Signal Structure Type Name: tSIG\_AMPGapLLConnDown

Signal Number	Parameters
SIG_AMP_GAP_LL_CONN_DOWN	Transac BdAddr

### 7.2.3.1 Signal Parameters

#### Transac

Size: 4 Bytes

Value	Parameter Description
0XXXXXXXXX (4 Bytes)	Context Information from the Application.

#### BdAddr

Size: 6 Bytes

Value	Parameter Description
A structure with one member called Byte, that is an array of 6 bytes, with the index 0 being the LS byte of the BD_addr	Bluetooth device Address of the remote device.