



# **kcHeadset v8.4 User Guide**

**Bluetooth v3.0**

## Introduction

Our kcHeadset firmware is primarily a speaker/headset receiver system. It can maintain a connection to a stereo transmitter (a remote A2DP Source device) and a connection to a cell phone (a remote AGHFP device) simultaneously. However, only one of the remote devices can stream audio at a time. Typically, kcHeadset is receiving stereo, until an incoming phone call is signaled. Then, if the user accepts the phone call, the stereo device is paused while the phone call is streamed. Once the phone call ends, the stereo streaming is resumed. Our kcHeadset can disable either of the audio profiles.

A rudimentary Serial Port Profile connection is available. This profile can transmit and receive data simultaneously with audio streaming. High bandwidth usage may negatively affect audio quality. Currently, the Serial Profile is not implemented with any features, and simply accepts incoming connection requests, where all data is seamlessly connected to the Uart for transmitting and receiving.

## Hardware

This User Guide covers device operations specific to this firmware edition. Hardware features and capabilities are outlined in the module Datasheets.

## Supported Bluetooth Profiles

Profile	Name	Version	Configured
A2DP	Advanced Audio Distribution Profile – Sink edition	1.2	Enabled
AVRCP	Audio Video Remote Control Profile – Controller edition	1.4	Enabled
HFP	Hands Free Profile	1.6	Enabled
HSP	Headset Profile	1.1	Enabled
MAP	Message Access Profile – Client edition	1.0	Optional
PBAP	Phonebook Access Profile – Client edition	1.2	Optional
SPP	Serial Port Profile – Client edition	--	Enabled

## Firmware Editions

Our default kcHeadset is released in class 1 and class 2 editions.

## Audio Codec Options

A2DP supports several audio codec formats. SBC (Sub-Band Coding) is the Bluetooth mandatory default. AAC is enabled by default, and subject to license fees for production products. ADPCM, APTX, and MP3 are optionally available. HFP support CVSD, aLAW, and uLaw codec formats, and are the Bluetooth mandatory defaults.

## A2DP & HFP Sampling Rates

A2DP supports 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, and 48KHz sampling rate outputs. The sampling rate is determined by the A2DP Source device. HFP supports 8 and 16KHz (wide band) sampling rates. The sampling rate is determined by the AGHFP device.

## Firmware Change Log

### V8.4 b1

- Changed customer specific features.

### V8.4 b0

- Added `AT ConnectConfig` command.
- Modified `AT Connect` to no longer drop connections while previously connected.
- Fixed A2DP reconnection bug.
- Added PTT detection upon HFP connection.
- Updated HFP commands display.
- Updated system output messages and synchronized build output to match kcGateway.

### V8.3 b3

- Added customizable HFP AT commands with the `AT HfpCmd` command.
- Changed customer specific features.

### V8.2 b1

- Fixed AT Events bug.

### V8.2 b0

- Added `AT PairingMax`, `PairingDelete`, `PinCode`, `PinConfig`, `PioRead`, `PioStatus`, `PioWrite`, `RemoteConnect` commands.
- Added remote command mode for SPP linked devices.
- Added PTT command to and SQ command compatible with AGv6.9 and later.

### V8.1 b1

- Added `AT AudioLoopback`, `OutputCallIn`, `OutputOnCall`, `ReconnectMode`, `TimeoutDisc`, `TimeoutIdle`, `TimeoutLinkloss` commands.
- Updated the At Command responses.
- Fixed audio streaming bugs.

### V8.0 b45

- Added `AT AutoAnswer`, `AuoDisc`, `AutoReconnect`, `Battery`, `Call`, `Connect`, `Disconnect`, `MonoMode`, `MicGainInit`, commands.
- Renamed `AT InputGain` -> `MicGain`, `VolumeDefault` -> `VolumeInit`.
- Updated At Command responses and system messages.

## Automatic Behaviors

Feature
Reconnect on startup
Reconnect on link loss
4 hour Idle shutoff
90 sec Discoverable timeout

## Feature Activation

PIO pins are used to activate firmware features. PIO default state is LOW (0V), and activates the assigned feature with a HIGH (3.3V) signal press, and LOW (0V) signal release. The “button presses” are debounced by 4 readings within 15ms. The following timings are configured for a “button press” to activate an assigned feature.

Press	Activation Time
Short	< 1.0 second
Double	Two Presses < 0.5 sec
Long	1.0+ seconds

Press	Activation Time
Very Long	2.5+ seconds
Very Very Long	5.0+ seconds
Hold	Repeat every 0.8 sec

## LED Event and State Indicators

When battery is low, the Red led blinks instead of the Blue led.

Event	LED Action	Timing
System On	Blue Flash	1s on
System Off	Red Flash	1s on
Reset Pairing List	Blue+Red Triple Flash	100ms on/off/on/off/on/off
Enter Test Mode	Blue+Red Triple Flash	100ms on/off/on/off/on/off
Enter DFU Mode	Blue+Red Triple Flash	100ms on/off/on/off/on/off

State	LED Action	Timing
Connectable	Blue Blinking	100ms on, 2500ms off
Connected, No Audio	Blue Double Blinking	100ms on/off/on, 2500ms off
Connected, Audio Streaming	Blue Double Blinking	100ms on/off/on, 1500ms off
Discoverable	Red/Blue Alternate Fast Blinking	100ms on/off
Reconnecting	Blue Fast Blinking	100ms on/off

## PIO Assignments

PIN Function	Name	I/O	Feature
ENABLE	POWR/BTB	Input	Power on/off. Multifunctional see below
PIO 2		Input	Enter DFU mode when HIGH during Power up
PIO 3		Input	
PIO 4	BTB	Input	Bluetooth button. Multifunctional see below
PIO 5	VOLUP	Input	Press: Volume up, Hold: Repeat
PIO 6	VOLDN	Input	Press: Volume down, Hold: Repeat
PIO 7	RR	Input	Press: Previous song, Hold: Rewind song
PIO 8	FF	Input	Press: Next song, Hold: Forward song
PIO 9	CONNECTED	Output	HIGH when device is connected
PIO 10	STREAMING	Output	HIGH when audio is streaming

### Multifunctional ENABLE / BTB

The BTB – Bluetooth button is a multi-featured input button. Most of the features are activated differently based on the current operating mode of the device. For example, if the device is not currently connected, then a short BTB press will trigger the reconnection feature. However, if a phone call is incoming, then a short BTB press will answer the call instead.

The ENABLE pin is a dual purpose pin, and kcAudioHeadset firmware can operate both power switch and power button modes.

First, power button mode is supported, where the ENABLE pin is tied to a momentary button (typically supplied directly from a li-ion battery). In this usage model, the ENABLE pin is used as the BTB. A long press of ENABLE will power up the device, and a subsequent very long press will power off the device. When the device is on, this ENABLE pin will provide the same features as the BTB.

Secondly, power switch mode is supported, where an external system power switch is used, typically to supply a DC power source. In this mode the ENABLE pin will be tied to this switched power source, and will simply turn on/off the device. In this mode, since the ENABLE pin is held HIGH when powered on, then BTB features must be operated using the BTB assigned PIO 4.

The device provides both power switch and power button operations by latching the system ENABLE internally, thus allowing the ENABLE pin to turn on/off the device with simple button presses, and additionally triggering all the features of the BTB when subsequently pressed. However, if the system is powered up, and the ENABLE pin remains HIGH for over 10 seconds, then the ENABLE button disables the internal power latch, which will allow the device to power off immediately upon release of the ENABLE pin (LOW).

## Button Controls

Note: The ENABLE button also inherits the same features as BTB when it is used as momentary power on/off button.

Feature	Button	Press	Condition
Power On	ENABLE	Very Long Press	Only when powered off
Power Off	ENABLE	Very Long Press	Only when powered on
Reconnect	BTB	Short Release	Only when connectable
Discoverable	BTB	Long Press	Only when connectable
Disconnect	BTB	Very Long Release	Only when connected
Cancel Discoverable	BTB	Short Release	Only when discoverable
Voice Command	BTB	Short Release	Only when connected
Answer Call	BTB	Short Release	Only when call incoming
Reject Call	BTB	Long Press	Only when call incoming
End Call	BTB	Short Release	Only when call active
Play/Pause	BTB	Short Press	Only when A2DP is active
Stop	BTB	Double Press	Only when A2DP is active
Volume Up	VOLUP	Short Press	Any
Volume Down	VOLDN	Short Press	Any
Previous Song	RR	Short Press	Only when connected
Rewind Song	RR	Hold	Only when streaming
Next Song	FF	Short Press	Only when connected
Fast Forward Song	FF	Hold	Only when streaming
Reset Pairing	VOLUP + VOLDN	Very Very Long	Any
Enter DFU Mode	PIO 2	High	During power up
Enter DFU Mode	PIO 2	Very Very Long	Any
Enter Test Mode	RR + FF	Very Very Long	Any
Test Audio Loopback	FF	Short	Not Implemented
Test Audio Tone	RR	Short	Not Implemented

## Output Volume

Default output volume for new connections is Level 14 = 0 dB, and can be changed with the `AT VolumeInit` command. Current volume levels are saved for each device, and separately for each profile (A2DP & HFP).

Level	Gain
0	-45.0 dB
1	-39.0 dB
2	-35.5 dB
3	-33.0 dB
4	-29.5 dB
5	-27.0 dB
6	-23.5 dB
7	-21.0 dB

Level	Gain
8	-18.0 dB
9	-15.0 dB
10	-12.0 dB
11	-9.0 dB
12	-6.0 dB
13	-3.0 dB
14	0 dB
15	+3.5 dB

## Microphone Gain

Default input gain is level 15 = 0.0 dB. Values are changed with the `AT MicGainInit` command.

Level	Gain
0	-45.0 dB
1	-41.5 dB
2	-39.0 dB
3	-35.5 dB
4	-33.0 dB
5	-29.5 dB
6	-27.0 dB
7	-23.5 dB

Level	Gain
8	-21.0 dB
9	-18.0 dB
10	-15.0 dB
11	-12.0 dB
12	-9.0 dB
13	-6.0 dB
14	-3.0 dB
15	0 dB

Level	Gain
16	+3.5 dB
17	+6.0 dB
18	+9.5 dB
19	+12.0 dB
20	+15.5 dB
21	+18.0 dB
22	+21.5 dB

## MicBias Settings

Default MicBias voltage is level 11, and default current is level 7. Values are changed with the `AT MicBias` command.

Mic Bias Voltage	
Level	Bias
0	1.71 V
1	1.76 V
2	1.82 V
3	1.87 V
4	1.95 V
5	2.02 V
6	2.10 V
7	2.18 V

Mic Bias Voltage	
Level	Bias
8	2.32 V
9	2.43 V
10	2.56 V
11	2.69 V
12	2.90 V
13	3.08 V
14	3.33 V
15	3.57 V

Mic Bias Current	
Level	Bias
0	0.20 mA
1	0.28 mA
2	0.34 mA
3	0.42 mA
4	0.48 mA
5	0.53 mA
6	0.61 mA
7	0.67 mA

Mic Bias Current	
Level	Bias
8	0.75 mA
9	0.81 mA
10	0.86 mA
11	0.95 mA
12	1.00 mA
13	1.09 mA
14	1.14 mA
15	1.23 mA

## Internal System Events

The following systems events are available for operations and notifications. These are listed to provide a comprehensive understanding of the built in capabilities. Most of these events are utilized in the default firmware, and all others can be triggered or utilized with customized editions of firmware.

0x01	EventPowerOn	0x3F	EventEnterPairingEmptyPDL	0x81	EventConfirmationRequest
0x02	EventPowerOff	0x40	EventReconnectFailed	0x82	EventPasskeyDisplay
0x03	EventEnterPairing	0x41	EventGasGauge0	0x83	EventPinCodeRequest
0x04	EventInitateVoiceDial	0x42	EventGasGauge1	0x84	EventEnableIIR
0x05	EventLastNumberRedial	0x43	EventGasGauge2	0x85	EventDisableIIR
0x06	EventAnswer	0x44	EventGasGauge3	0x86	EventPbapDialMch
0x07	EventReject	0x45	EventCheckForAudioTransfer	0x87	EventPbapDialICh
0x08	EventCancelEnd	0x46	EventEnterDFUMode	0x88	EventEstablishPbap
0x09	EventTransferToggle	0x47	EventGaiaAlertLEDs	0x89	EventPbapDialFail
0x0A	EventToggleMute	0x48	EventEnterServiceMode	0x8A	EventSetWbsCodecs
0x0B	EventVolumeUp	0x49	EventServiceModeEntered	0x8B	EventOverrideResponse
0x0C	EventVolumeDown	0x4E	EventEnableVoicePrompts	0x8C	EventCreateAudioConnection
0x0D	EventToggleVolume	0x4f	EventDialStoredNumber	0x8D	EventSetWbsCodecsSendBAC
0x0E	EventThreeWayReleaseAllHeld	0x50	EventDisableVoicePrompts	0x8E	EventUpdateStoredNumber
0x0F	EventThreeWayAcceptWaitingReleaseActive	0x51	EventChargeDisabled	0x92	EventEnterBootMode2
0x10	EventThreeWayAcceptWaitingHoldActive	0x52	EventRestoreDefaults	0x93	EventAvrcpPlayPause
0x11	EventThreeWayAddHeldTo3Way	0x53	EventChargerGasGauge0	0x94	EventAvrcpStop
0x12	EventThreeWayConnect2Disconnect	0x54	EventChargerGasGauge1	0x95	EventAvrcpSkipForward
0x13	EventEnableDisableLeds	0x55	EventChargerGasGauge2	0x96	EventAvrcpSkipBackward
0x14	EventResetPairedDeviceList	0x56	EventChargerGasGauge3	0x97	EventAvrcpFastForwardPress
0x15	EventEnterDutMode	0x57	EventContinueSLCConnectRequest	0x98	EventAvrcpFastForwardRelease
0x16	EventPairingFail	0x58	EventConnectableTimeout	0x99	EventAvrcpRewindPress
0x17	EventPairingSuccessful	0x59	EventLastNumberRedial_AG2	0x9A	EventAvrcpRewindRelease
0x18	EventSCOLinkOpen	0x5A	EventInitateVoiceDial_AG2	0x9B	EventPbapSetPhonebook
0x19	EventSCOLinkClose	0x5B	EventConfirmationAccept	0x9C	EventPbapBrowseEntry
0x1A	EventLowBattery	0x5C	EventConfirmationReject	0x9D	EventPbapBrowseList
0x1B	EventEndOfCall	0x5D	EventToggleDebugKeys	0x9E	EventPbapDownloadPhonebook
0x1C	EventEstablishSLC	0x5E	EventTone1	0x9F	EventPbapSelectPhonebookObject
0x1D	EventLEDEventComplete	0x5F	EventTone2	0xA0	EventPbapBrowseComplete
0x1E	EventChargeComplete	0x60	EventSelectTTSLanguageMode	0xA6	EventAvrcpToggleActive
0x1F	EventAutoSwitchOff	0x61	EventConfirmTTSLanguage	0xA7	EventAvrcpNextGroup
0x20	EventChargeInProgress	0x62	EventEnableMultipoint	0xA8	EventAvrcpPreviousGroup
0x21	EventOkBattery	0x63	EventDisableMultipoint	0xA9	EventUsbPlayPause
0x22	EventChargerConnected	0x64	EventStreamEstablish	0xAA	EventUsbStop
0x23	EventChargerDisconnected	0x65	EventSLCConnectedAfterPowerOn	0xAB	EventUsbFwd
0x24	EventSLCDisconnected	0x66	EventResetLEDTimeout	0xAC	EventUsbBck
0x25	EventBatteryLevelRequest	0x67	EventStartPagingInConnState	0xAD	EventCriticalBattery
0x26	EventLinkLoss	0x68	EventStopPagingInConnState	0xAE	EventRssiResume
0x27	EventLimboTimeout	0x69	EventMultipointCallWaiting	0xAF	EventUsbAudioDisconnect
0x28	EventMuteOn	0x6A	EventRefreshEncryption	0xB0	EventPowerOnPanic
0x29	EventMuteOff	0x6B	EventSwitchAudioMode	0xB1	EventEstablishSLCOnPanic
0x2a	EventMuteReminder	0x6C	EventButtonLockingOn	0xB2	EventTestDefrag
0x2b	EventResetComplete	0x6D	EventButtonLockingOff	0xB3	EventUsbDeadBatteryTimeout
0x2C	EventEnterTXContTestMode	0x6E	EventToggleButtonLocking	0xB4	EventUsbMute
0x2D	EventEnterTestState	0x6F	EventButtonBlockedByLock	0xB5	EventUsbLowPowerMode
0x2E	EventVolumeOrientationNormal	0x70	EventSpeechRecognitionTuningStart	0xB6	EventSpeechRecognitionStart
0x2F	EventVolumeOrientationInvert	0x71	EventSpeechRecognitionTuningYes	0xB7	EventSpeechRecognitionStop
0x30	EventNetworkOrServiceNotPresent	0x72	EventRssiPair	0xB8	EventWiredAudioConnected
0x31	EventNetworkOrServicePresent	0x73	EventRssiPairReminder	0xB9	EventWiredAudioDisconnected



0x32	EventEnableLEDS	0x74	EventRssiPairTimeout	0xBC	EventAudioTestMode
0x33	EventDisableLEDS	0x75	EventBassBoostOn	0xBD	EventToneTestMode
0x34	EventSLCConnected	0x76	EventCheckRole	0xBE	EventKeyTestMode
0x35	EventError	0x77	EventMissedCall	0xBF	EventSpeechRecognitionTuningNo
0x36	EventLongTimer	0x78	EventBassBoostOff	0xC8	EventUpdateAttributes
0x37	EventVLongTimer	0x79	EventA2dpConnected	0xC9	EventCancelPairing
0x38	EventEnablePowerOff	0x7a	EventA2dpDisconnected	0xCE	kcEventHfpPttOn
0x39	EventBassBoostEnableDisableToggle	0x7b	Event3DEnhancementEnableDisableToggle	0xCF	kcEventHfpPttOff
0x3A	EventPlacelIncomingCallOnHold	0x7c	Event3DEnhancementOn	0xD1	EventSwapMediaChannel
0x3B	EventAcceptHeldIncomingCall	0x7d	Event3DEnhancementOff	0xD2	EventCheckAudioAmpDrive
0x3C	EventRejectHeldIncomingCall	0x7e	EventVolumeMax	0xD3	EventExternalMicConnected
0x3D	EventCancelLedIndication	0x7f	EventVolumeMin	0xD4	EventExternalMicDisconnected
0x3E	EventCallAnswered	0xBA	EventPrimaryDeviceConnected	0xD5	EventDisconnect
0x74	EventRssiPairTimeout	0xBB	EventSecondaryDeviceConnected		

## Operational Messages

The device will send many different uart messages during operations, included device state changes, connection and disconnection notices, audio channel usage, and more.

For example, it will display multiple state changes in between adding a connection, dropping it, and returning to a discoverable idle mode.

```

-> State [Connectable]
-> ConnectionUp Hfp 00:26:E2:27:F5:75
-> State [Connected]
-> VolumeHfp 15
-> AutoAnswer
-> ConnectionDown Hfp 00:26:E2:27:F5:75
-> State [Connectable]
-> State [Discoverable]

```

Incoming and outgoing call activity will also generate system messages regarding changes in state and settings.

```

-> AudioChannelDn
-> State [CallOutgoing]
-> AudioChannelUp Hfp Cvsd
-> VolumeHfp 13
-> State [CallActive]
-> VolumeHfp 14
-> VolumeHfp 15
-> State [Connected]
-> Call End
-> AudioChannelDn

```

## Device State Messages

State Message	Description
A2dpStreaming	The headset is streaming A2DP audio.
CallActive	The headset has a current call, and the audio channel is utilized.
CallActiveNoAudio	The headset has a current call, but not using the audio channel (i.e. speaker phone mode).
CallIncoming	An incoming call is in progress on the remote device (phone).
CallOutgoing	An outgoing call is in progress on the remote device (phone).
Connectable	The headset is connectable, in idle mode.
Connected	The headset is connected to a remote device, in standby mode.
Discoverable	The headset is discoverable and connectable, in idle mode.
Limbo	The headset is logically off but physically on – limbo.
TestMode	The headset is in test mode.

## Tutorial

### Startup

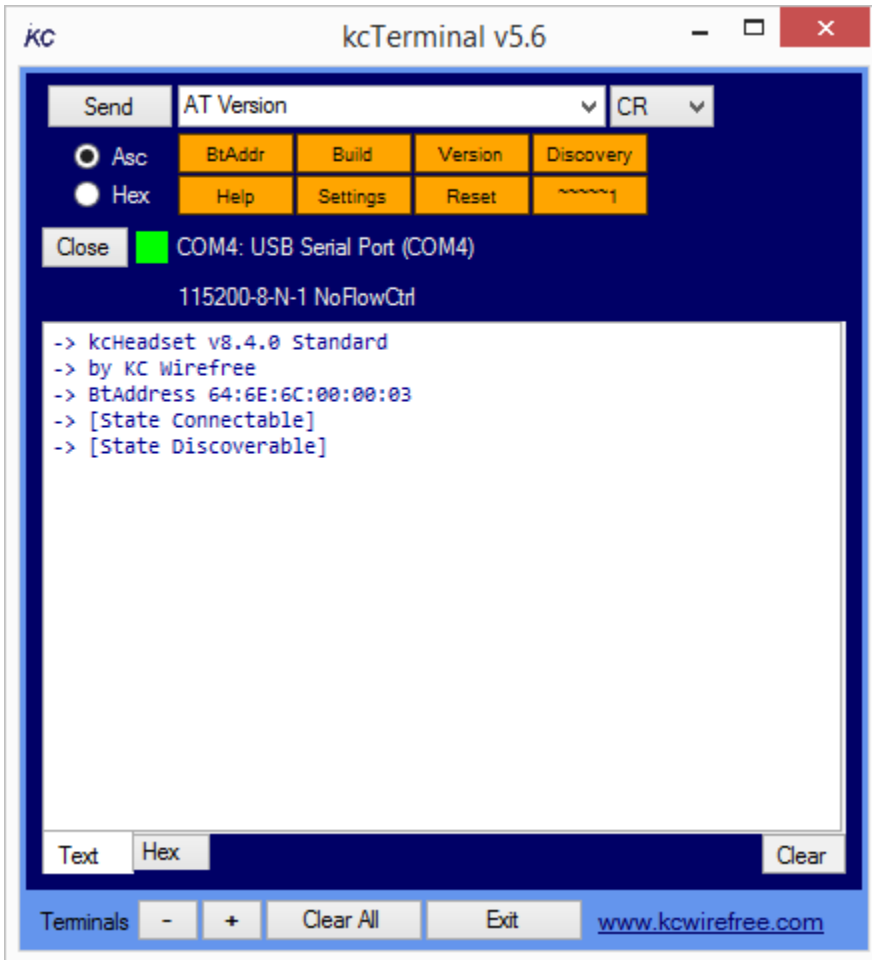
Make sure the Bluetooth demo board is either supplied power from the USB port or from a battery connection, or both. Having the USB supply power while the battery is plugged in will charge the battery while the device is in operational and limbo mode. When charging with USB, the device cannot fully shut off. Instead, it will enter a deep sleep (limbo) mode in which only the Enable button is monitored.

To turn the device on, press and hold the Enable button until the LED's flash. If the Enable button is held high for ten seconds after startup, the device can be turned off simply by cutting power from the Enable button. If however the Enable button was released after startup, turning off the device can be accomplished by pressing the button again.

On the side of the demo board are three micro USB ports. The first is a USB port and can be used for charging and downloading DFU firmware updates in DFU mode. Second is the UART port which will be the main use for device outputs and commands to and from the user. Lastly is the SPI port, which will require a SPI adaptor for computer connection. SPI is used for installing .xuv type files as well as reading and writing the device flash memory.

For UART operation, connect your computer to the micro USB port labeled UART. You will then need to open a UART text box terminal. You can download the kcTerminal v5.6 at [www.kcwirefree.com/downloads.html](http://www.kcwirefree.com/downloads.html). Select the proper COM port after making sure the UART settings are the same as our device default settings (see AT Command Syntax).

After properly powered and connected to UART, a successful startup should look similar to below.



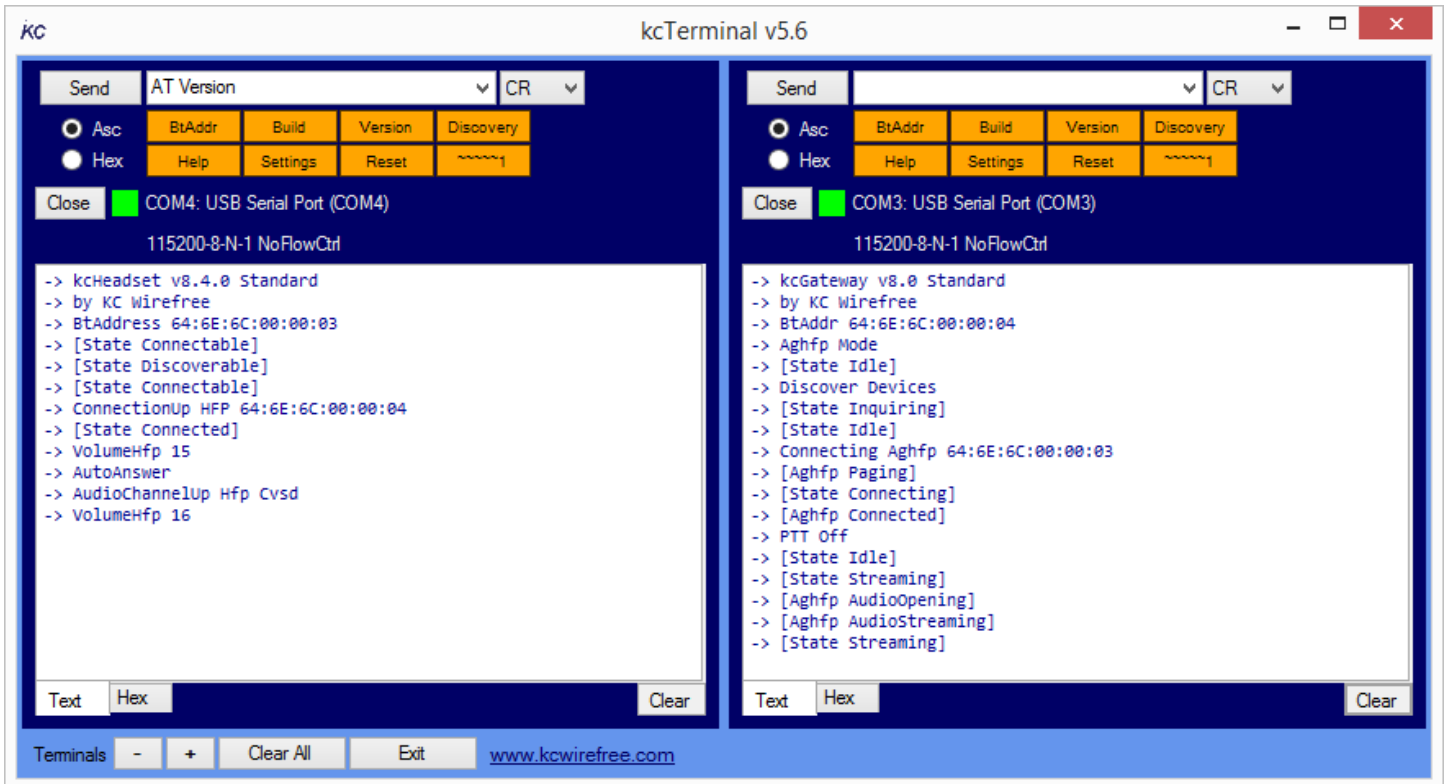
## Getting Connected

Device operations can be initiated by either button pressing or using the AT commands from a UART input. Changing device settings and customizing timeouts and features however can only be done from the AT commands.

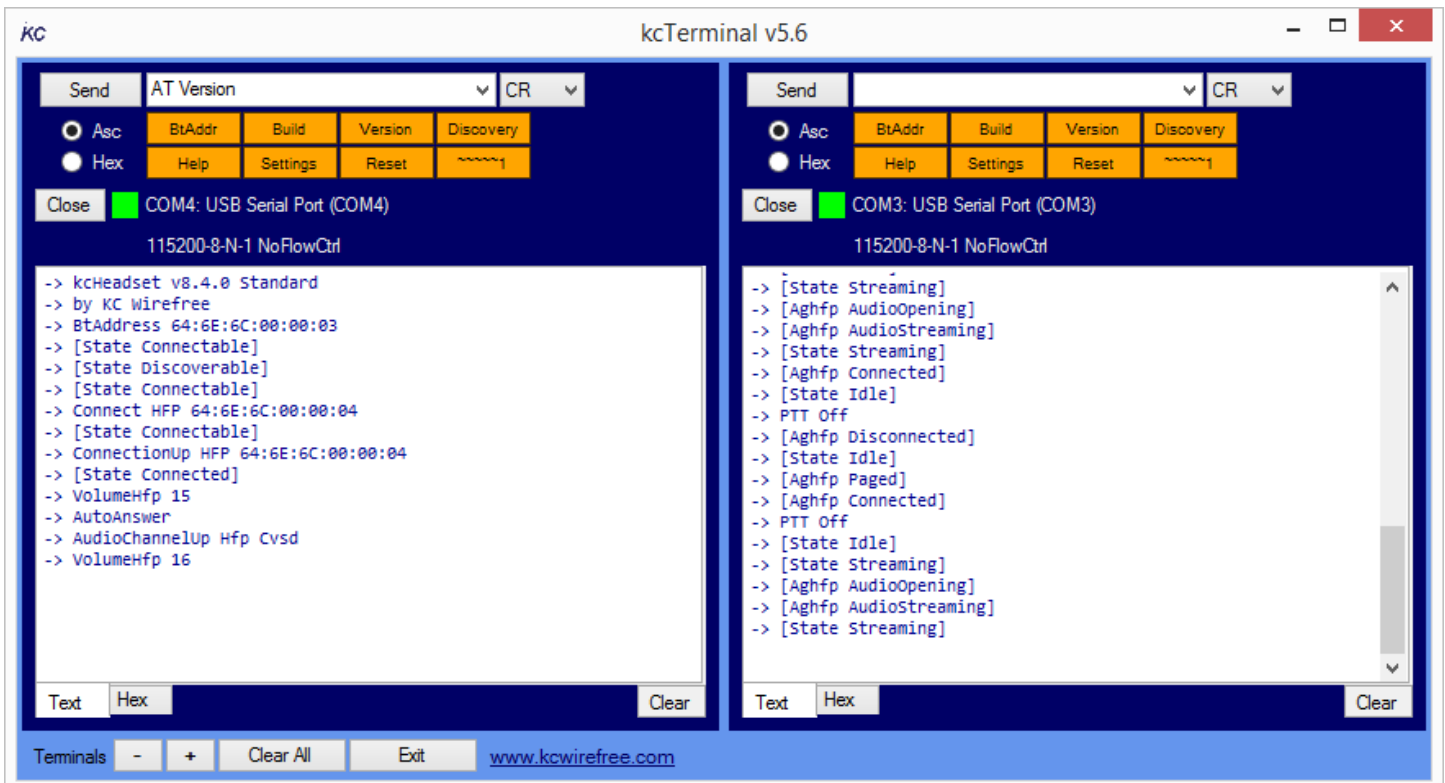
In order to get connected to a Gateway source device, the headset device must be put into the Discoverable Bluetooth state. The Gateway device will then need to start a device inquiry, where it will find the Headset and begin the pairing process.

Press and hold the BTB (PIO 4) for a couple seconds until the state becomes Discoverable. Do the same on the Gateway device to begin its Inquiry. If the signal is strong enough and they have the same profiles enabled, they should automatically pair up.

Below is an example of a successful connection using only button presses. This could have also been accomplished by using the “AT Discoverable” command instead of the button presses.



To reconnect to a device after a connection is dropped, quickly press and release the BTB. This issues a reconnection process which, depending on your settings, sends connection requests to all paired devices one at a time or just the last device repeatedly. Below is a successful reconnection after the Headset was turned off



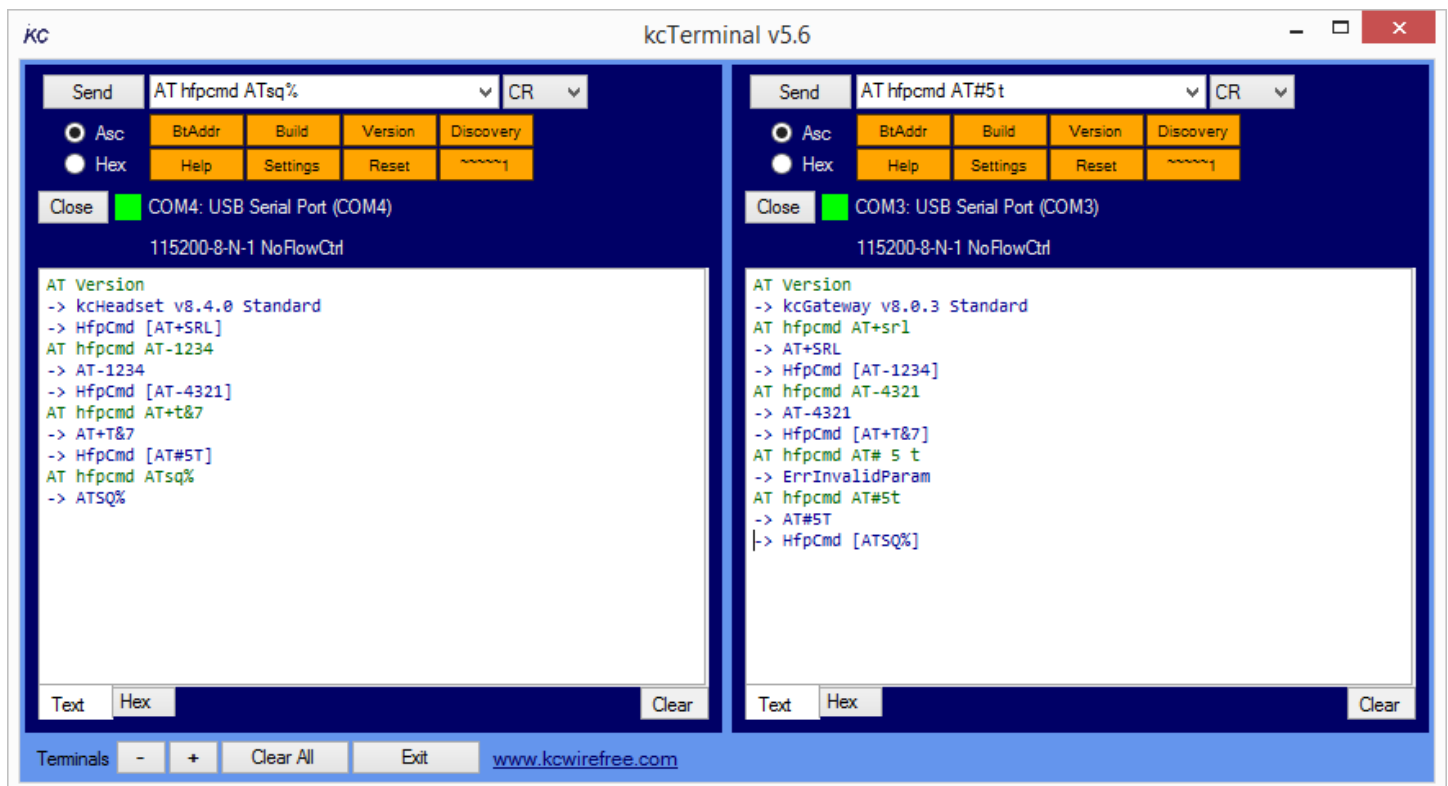
If the reconnection is unsuccessful, you may need to change the profiles back what they were when connected. Deleting pairing information and then going through the pairing process again is often a very effective tool for connecting two devices.

## Sending HFP Commands

During HFP profile connections, wireless commands are sent between the devices in the form of HFP AT commands. These are not the same as the UART commands and have different formatting.

The UART “AT HfpCmd” command can be used to send your own custom HFP AT commands to a Hands Free Profile connected device. Details on how to format the command can be found in the AT HfpCmd section further below.

The device will also display any custom HFP commands that it receives. An example of sending and receiving proper custom HFP commands with a Gateway is shown below.



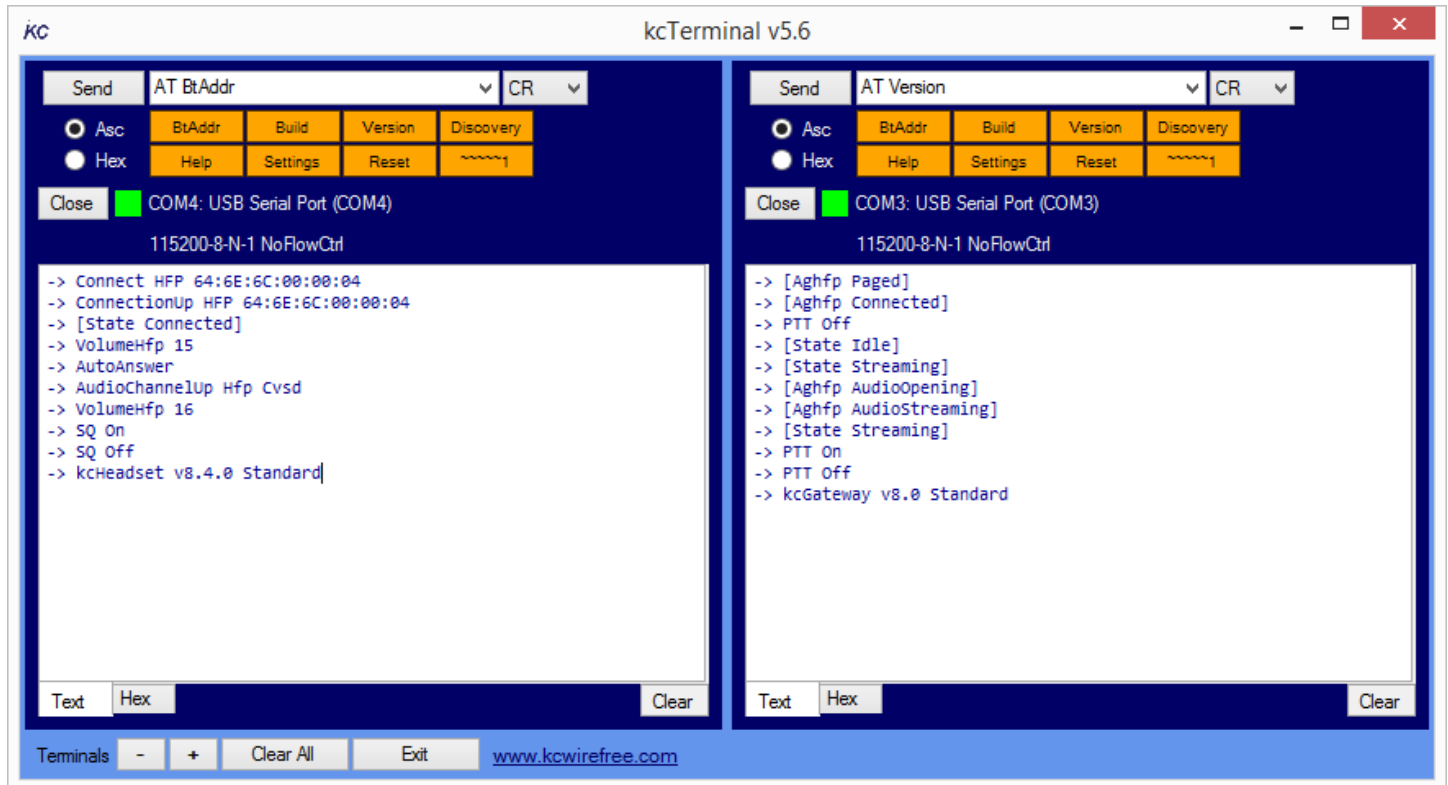
Note that most ASCII characters can be sent. Also be aware that it is best to send smaller strings.

## Push-To-Talk

The PTT feature is only available for HFP connections, just like the HFP commands.

The functionality of PTT is to suspend audio playback while the PTT PIO is high without terminating the audiostream. For our kcGateway devices, pressing the PTT button on the Headset (setting PIO 2 high) will send an HFP AT command to the gateway, suspending the audio and also setting the kcGateway equivalent PTT (PIO 2) to high. Releasing the Headset PTT button (dropping PIO 2 to low) sends another HFP AT command that will resume audio playback and drop the PTT PIO.

SQ works similarly to PTT, except that no audio is suspended and it is a PIO 3 change on the Gateway that is sent to the Headset. Below is an example of a Headset connecting to a Gateway, holding the PTT high for one second, then releasing it. Afterwards, the Gateway holds the PIO for SQ (PIO 3) high for one second before releasing it.



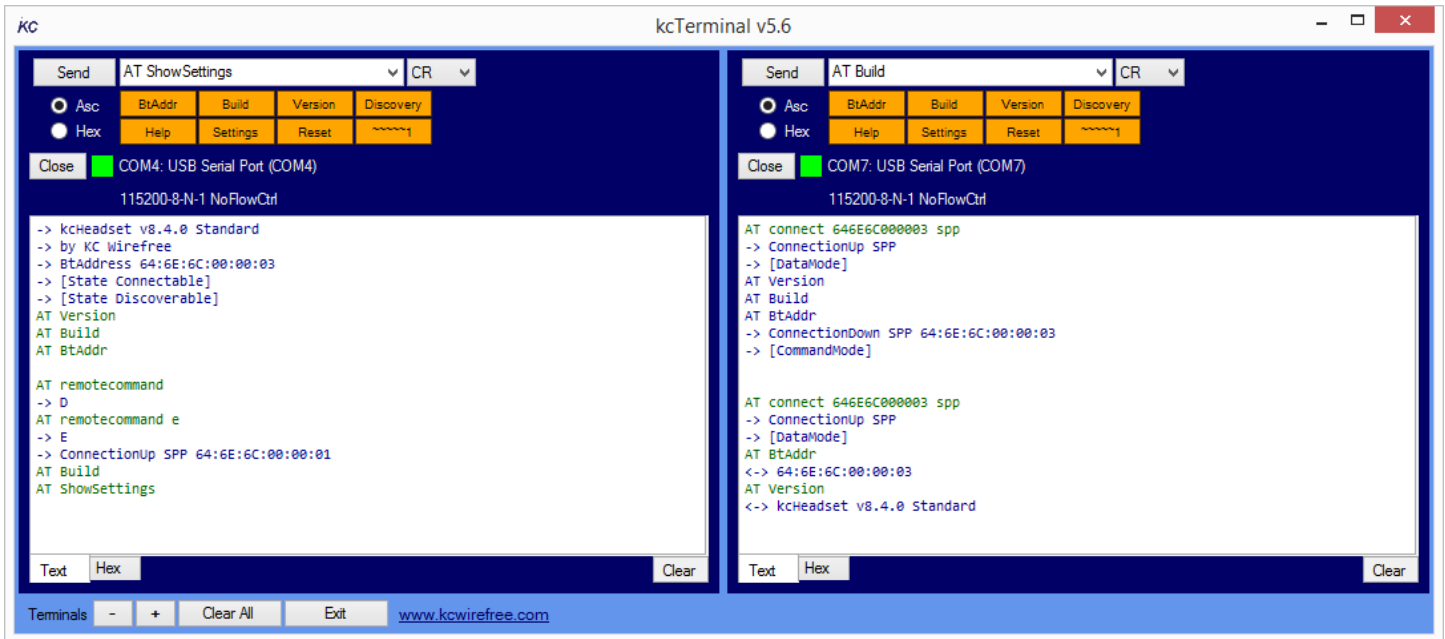
What cannot be seen is that while PTT was on, Gateway PIO 2 was high and audio was suspended. Those changed after PTT turned off. While SQ was on, Headset PIO 3 was high, but then switched to low after SQ turned off.

## Remote Command

For a connection using the Serial Port Profile (SPP), it is possible to control how data received wirelessly is handled using the "AT RemoteCommand"

When RemoteCommand is disabled, an SPP connection will simply pass data to the HS where it is taken from the RFCOMM and redirected straight to the UART output. With RemoteCommand enabled, the data from the SPP device will be processed by the handler and all HS responses will be directed to the remote device.

Below is an example of the Headset being connected to a kcSerial module with Serial Port Profile.



At first the Headset has `RemoteCommand` disabled. All of the commands we issue to the Headset are sent to the Serial device as data. Then `RemoteCommand` becomes enabled. Issuing commands on the Headset side does nothing because the Serial device is now given control. The kcSerial device issues commands for the Headset to process and all outputs are relayed back to the remote Serial device.

## AT Command Syntax

### Syntax

Default UART setting is 115200-8-N-1, without hardware flow control.

Enter AT Commands via UART as standard strings, with a CR End Of Line marker (0x0D), and optionally LF (0x0A). Output messages are terminated with CRLF (0x0D 0x0A).

Each AT Command accepts a “?” parameter, which will then display the required and optional parameters for that particular command.

Entering an AT Command without any required parameters, in most cases, will simply display the current parameter settings.

## AT Command List

AutoAnswer	Help	PioTest
AutoDisc	HfpCmd	PioWrite
AutoReconnect	LinkTest	Profiles
Avrcp	MicBias	PsRead
Battery	MicGain	PsWrite
BtAddr	MicGainInit	RemoteCommand
Build	MonoMode	Reset
Call	Name	Restrict
Codec	OutputCallIn	ShowSettings
Connect	OutputOnCall	State

ConnectConfig	PairingDelete	TimeoutDisc
DeepSleep	PairingMax	TimeoutIdle
Dfu	PinCode	TimeoutLinkloss
Disconnect	PinConfig	Version
Discoverable	PioRead	Volume
Event	PioStatus	VolumeInit

Command lines parsed and executed when the EOL carriage return is received.



## AT Commands

### AT AutoAnswer

The AutoAnswer command enables or disables the device from automatically answering calls.

Command	<code>AT AutoAnswer &lt;e/d&gt;</code>
<e/d>	Enable or Disable
Example	<code>AT AutoAnswer E</code> <code>-&gt; E</code>

### AT AutoDisc

The AutoDisc command allows the headset to automatically enter discoverable mode when unconnected.

Command	<code>AT AutoDisc &lt;e/d&gt;</code>
<e/d>	Enable or Disable
Example	<code>AT AutoDisc E</code> <code>-&gt; E</code>

### AT AutoReconnect

The AutoReconnect command allows the headset to automatically reconnect with the last known device at power on.

Command	<code>AT AutoReconnect &lt;e/d&gt;</code>
<e/d>	Enable or Disable
Example	<code>AT AutoReconnect E</code> <code>-&gt; E</code>

### AT Avrcp

The Avrcp command is purely an operational command to control the remote device's media player. The headset must be in A2DPStreaming mode, and the AVCRP profile must be enabled and connected with the remote device, otherwise, these commands have no effect.

The Avrcp FF (Fast Forward) and Avrcp RR (Rewind) commands have several options, including Press/Release/Skip/Group. When a FF or RR Press command is sent, the remote media begins playing in the fast forward or rewind direction, until the FF or RR Release command is received.

The Avrcp FF or RR Skip command simply prompts the remote media player to skip to the next (FF) or previous (RR) track

The Avrcp FF or RR Group command prompts the remote media player to skip to the next (FF) or previous (RR) playlist.

The Avrcp PP command is PlayPause, which will toggle the remote media play status.

The Avrcp SP command will issue a stop message to the remote media device.

Command	<code>AT Avrcp &lt;cmd&gt; &lt;option*&gt;</code>
<cmd>	<code>FF*, RR* (*requires an option parameter) PP, SP</code>
<option>	<code>Press, Release, Skip, Group (option parameters only used with FF or RR commands)</code>
Example	<code>AT Avrcp FF Press -&gt; Avrcp FF Press</code>
Example	<code>AT Avrcp PP -&gt; Avrcp pp</code>

## AT Battery

The Battery command provides a VDD pin voltage level reading, which is commonly a Lithium Ion battery.

Command	<code>AT Battery</code>
Example	<code>AT Battery -&gt; 3812 mV</code>

## AT BtAddr

The Build command outputs the full Bluetooth address.

Command	<code>AT BtAddr</code>
Example	<code>AT BtAddr -&gt; 64:6E:6C:00:00:03</code>

## AT Build

The Build command outputs the full firmware version information.

Command	<code>AT Build</code>
Example	<code>AT Build -&gt; [Build] -&gt; BtAddr 64:6E:6C:00:00:03 -&gt; Bluetooth: v3.0 -&gt; Hardware: KC6012 -&gt; Firmware: kcAudioHS -&gt; Version: v8.2.0 -&gt; Date: Mar 25 2015 14:28:08 -&gt; [Build End]</code>

## AT Call

The Call command allows the user to make a call by either choosing to answer an event incoming call, end a current event call, redial the last number, reject an incoming call, or initiate voice dial.

Command	<code>AT Call &lt;answer/end/redial/reject/voice&gt;</code>
<option>	<code>Answer, End, Redial, Reject, Voice</code>
Example	<code>AT Call Answer</code> <code>-&gt; Call Answer</code>

## AT Codec

The Codec command can be enabled/disabled individual available codecs. This configuration is saved in flash memory. Note: MP3, AAC, APTX, APTXLL are subject to licensing fees by their respective owners. MP3 and AAC codecs are fully functional, and are provided for evaluation purposes only, but not licensed for production products. APTX and APTXLL are not functional without an additional license configuration. Please contact KC Wirefree for demo edition licenses.

Command	<code>AT Codec &lt;eeee/dddd (Aac/Aptx/AptxLL/Fast)&gt;</code>
<e/d>	<code>Enable/Disable for each audio compression codec Aac/Aptx/AptxLL/Fast</code>
Example	<code>-&gt; AT Codec DDDE</code> <code>-&gt; Aac[D] Aptx[D] AptxLL[D] Fast[E]</code>

## AT Connect

The Connect command initiates a reconnection sequence as specified by the AT ConnectConfig command. Same operation as the Connect button press. If issued while currently connected then it will respond with the connection type.

Command	<code>AT Connect</code>
Example	<code>-&gt; AT Connect</code> <code>-&gt; Connect HFP 64:6E:6C:00:00:04</code>

## AT ConnectConfig

The ConnectConfig command will set the settings the AT Connect and Connect button press will use for reconnection. The first parameter is the type of reconnection that will be made. Either "LAST" which sends all connection attempts to the last connected device, or "LIST" which will send attempts one at a time through previously connected devices on the paired device list until the limit of attempts is reached. The second optional parameter sets the number for connection attempts. Default is LAST type and 4 attempts.

Command	<code>AT ConnectConfig &lt;Type&gt; &lt;*number&gt;</code>
<Type>	Type of reconnection. "List" or "Last"
<*number>	Amount of attempts to connect the device will make
Example	<code>AT ConnectConfig List 6</code> <code>-&gt; List 6</code>

## AT DeepSleep

The DeepSleep command is used to enable and disable DeepSleep mode. Please contact technical support on the usage of the command.

Command	<code>AT DeepSleep &lt;e/d&gt;</code>
Example	<code>AT DeepSleep E</code> <code>-&gt; E</code>

## AT Dfu

The Dfu command is used to set the device into firmware update mode. The device will immediately reboot into the Dfu mode, where the DfuWizard application can download a new firmware image into the device via USB interface. Please see Firmware Update section regarding specific procedure details.

Command	<code>AT Dfu</code>
Example	<code>AT Dfu</code> <code>-&gt; DfuMode [Reboot]</code>

## AT Disconnect

The Disconnect command will disconnect all currently connected devices, and revert to Connectable or Discoverable mode (set to Discoverable by default). Same operation as the Disconnect button press.

Command	<code>AT Disconnect</code>
Example	<code>AT Disconnect</code> <code>-&gt; Disconnecting</code>

## AT Discoverable

The Discoverable command immediately turns on or off Discoverable (Pairing) mode upon an unsuccessful connection attempt. The device remains Discoverable for 90 seconds by default. The duration time is configurable separately using `AT TimeoutDisc`.

Command	<code>AT Discoverable &lt;e/d&gt;</code>
<e/d>	Enable/Disable discoverable mode
Example	<code>AT Discoverable E</code> <code>-&gt; E</code>

## AT Event

The Event command provides a quick method to trigger system functions or responses. Most system events are highly dependent upon the current device state and many other device settings, so the actual effect can be unexpected. This command is provided as an experimental option, or perhaps used as simple backdoor method to trigger an otherwise unimplemented function or response. The table of all events and id numbers is listed in this document.

Command	<code>AT Event &lt;event&gt;</code>
<event>	Event ID number in hex (Event 0C is VolumeDown)
Example	<code>AT Event 0C</code> <code>-&gt; 0C</code>

## AT Help

The Help command will list all implemented AT Commands. Also, each command can accept an optional “?” parameter, which will output the list of command arguments.

Command	<code>AT Help</code>
Example	<code>AT Help</code> <code>-&gt; AutoAnswer</code> <code>-&gt; AutoDisc</code> <code>-&gt; Avrcp</code> <code>-&gt; Battery</code> <code>...etc</code>

## AT HfpCmd

The AT HfpCmd sends a custom HFP AT Command to a connected Gateway device. It can be used to send Bluetooth standard At Commands or custom AT commands. The formatting of the command **must begin with “AT”**. It is customary for Bluetooth AT commands to have a “+” or “-” between the “AT” and the command itself. A line return character is appended automatically. All AT commands are automatically converted to upper case. This command will only be effective in a HFP connection.

Command	<code>AT HfpCmd &lt;Command&gt;</code>
<Command>	Custom or established HFP At Command. Must begin with "AT".
Example	<code>AT HfpCmd AT+Ptton</code> <code>-&gt; AT+PTTON</code>
Example	<code>AT HfpCmd AT+PttOff</code> <code>-&gt; AT+PTTOFF</code>

## AT LinkTest

The LinkTest command can connect to a remote SPP device, and report the link quality reading.

Command	<code>AT Linktest &lt;bdaddr&gt; &lt;iterations*&gt;</code>
<bdaddr>	The remote device Bluetooth address
<iterations*>	Number of test iterations, default = 1
Example	<code>AT Linktest 646e6cffffff 5</code> <code>-&gt; [LinkTest Start]</code> <code>LinkTest 1919 Bytes BER%=0.0900</code> <code>LinkTest 1919 Bytes BER%=0.1000</code> <code>LinkTest 1919 Bytes BER%=0.2600</code> <code>LinkTest 1919 Bytes BER%=0.2600</code> <code>LinkTest 1919 Bytes BER%=0.1800</code> <code>-&gt; [LinkTest End]</code>

## AT MicBias

The MicBias command is used to configure the physical bias settings on the input port.

Command	<code>AT MicBias &lt;voltage&gt; &lt;current&gt; &lt;e/d/f&gt;</code>
<voltage>	Value 0-15
<current>	Value 0-15
<enable>	D=Disabled, E=Enabled Automatic, F=Forced On
Example	<code>AT MicBias 11 7 E</code> <code>-&gt; Voltage[11] Current[7] Enable[E]</code>

## AT MicGain

The MicGain command adjusts or sets the microphone input gain without modifying the default gain setting. The command without parameters returns the current setting. The adjusted input gain setting is not saved in memory. Use AT MicGainInit to set the default input gain.

Command	<code>AT MicGain &lt;+/-/gain*&gt;</code>
<+/-/gain>	Either + increment gain, - decrement gain, or set level 0-22
Example	<code>AT MicGain -&gt; 6</code>
Example	<code>AT MicGain + -&gt; 7</code>
Example	<code>AT MicGain 20 -&gt; 20</code>

## AT MicGainInit

The MicGainInit command sets the initial or default microphone input gain the device will have upon activation or startup.

Command	<code>AT MicGainInit &lt;gain&gt;</code>
<gain>	Set Gain level 0-22
Example	<code>AT MicGainInit 14 -&gt; 14</code>

## AT MonoMode

The MonoMode command forces the stereo signal into a mono output only on the left channel.

Command	<code>AT MonoMode &lt;e/d&gt;</code>
<e/d>	Enable/Disable Mono Mode
Example	<code>AT MonoMode E -&gt; E</code>

## AT Name

The Name command is used to set the name of this device reported when other Bluetooth devices perform discoveries. The name is saved in flash memory.

Command	<code>AT Name &lt;devicename*&gt;</code>
<key>	Up to 32 character name.
Example	<code>AT Name My Speaker -&gt; My Speaker</code>

## AT OutputCallIn

The OutputCallIn command will allow the user to set a PIO Output as an indicator for an incoming call. This command

uses a disable or a pin number as an enable.

Command	<code>AT OutputCallIn &lt;d/pio&gt;</code>
<e/d>	Disable/PIO to set as
Example	<code>AT OutputCallIn D</code> <code>-&gt; D</code>
Example	<code>AT OutputCallIn 3</code> <code>-&gt; 3</code>

## AT OutputOnCall

The OutputOnCall command will set the PIO to drive when on a call. This command uses the basic disable or enable switch <e/d>.

Command	<code>AT OutputOnCall &lt;e/d&gt;</code>
<e/d>	Disable/Enable
Example	<code>AT OutputOnCall E</code> <code>-&gt; E</code>

## AT PairingMax

The PairingMax command is used set the limit for how many previous devices the device can store in the Paired Devices List. The default limit is 8. The acceptable range is 1-8.

Command	<code>AT PairingMax &lt;max&gt;</code>
<max>	Limit of Bluetooth devices to remember in PDL. <1-8>
Example	<code>AT PairingMax 4</code> <code>-&gt; 4</code>

## AT PairingDelete

The PairingDelete command is used to erase all paired device entries.

Command	<code>AT PairingDelete</code>
Example	<code>AT PairingDelete</code> <code>-&gt; Ok</code> <code>-&gt; ResetPairedList</code>

## AT PinCode

The PinCode command allows changing the Pin code for pairing. Default Pincode for kcAudio is 1234. The Pincode is only required when a secure connection is requested by a legacy device (Bluetooth v2.0 or earlier). By default, kcAudio automatically sends this Pincode when a Pincode is requested.



Command	<code>AT PinCode &lt;pin&gt;</code>
<pin>	Valid Pincode is 1-15 alphanumeric characters.
Example	<code>AT PinCode 5565</code> <code>-&gt; 5565</code>

## AT PioConfig

The PioConfig command is used to configure one of the general PIO pins as an input or output. All PIO pins are set as inputs by default, unless they have been assigned to special Output features. A PIO Feature assigned to a pio pin must be disabled prior re-configuring the pio. The pio configurations are not saved in memory.

Command	<code>AT PioConfig &lt;pio&gt; &lt;i/o&gt; &lt;+ pull*&gt;</code>
<pio>	0-15, The PIO pin to configure
<i/o>	i=Input, o=Output
<+ pull>	+ =StrongPull up/down
Example	<code>AT PioConfig 8 0 +</code> <code>-&gt; Pio[8] = Output+</code>
Example	<code>AT PioConfig 8 I</code> <code>-&gt; Pio[8] = Input</code>

## AT PioRead

PioRead will supply the current reading of the pin, the configuration input or output, and the name of any special PIO Feature currently using this pin. If the PIO pin is configured as a strong pull up/down, a '+' sign will print following configuration.

Command	<code>AT PioRead &lt;pio&gt;</code>
<pio>	0-15, the PIO pin to read
Example	<code>AT PioRead 8</code> <code>-&gt; Pio[8] = 1 Input+</code>
Example	<code>AT PioRead 7</code> <code>-&gt; Pio [7] = 1 Output+</code>
Example	<code>AT PioRead 19</code> <code>-&gt; ErrInvalidParam</code>

## AT PioStatus

The PioStatus command provides readings for all PIO pins in a compact hexadecimal format. The Read parameter indicates the HIGH or LOW reading state, where the hexadecimal bit position [15-0] corresponds to the PIO pin number. The Dir parameter indicates the Input or Output direction of the PIO, with Input LOW and Output HIGH. The Str parameter indicates if any PIO is set to have a strong pull up/down. The mask also uses the hexadecimal bit position [15-0] to correspond with the PIO pin number, where HIGH is a present strong pull up/down and LOW is not.

E.g. Read[0020] indicates PIO 6 is HIGH, and remaining PIO's are LOW.

E.g. Dir[0000] indicates PIO's 10 and 3 are set as Outputs, and the remaining PIO's are Inputs.

E.g. Str[0001] indicates PIO 8 was set for a strong bias.

Command	<code>AT PioStatus</code>
Example	<code>AT PioStatus</code> <code>-&gt; Read[0020] Dir[0408] Str[0100]</code>

## AT PioTest

The PioTest command is a simple PIO high/low state test. Each PIO is set, and read back. Any discrepancy reports a failure.

Command	<code>AT PioTest</code>
Example	<code>AT PioTest</code> <code>-&gt; Pio[3] 1 0 = Error</code> <code>.....</code> <code>-&gt; [PioTest Fail]</code>
Example	<code>AT PioTest</code> <code>-&gt; [PioTest Pass]</code>

## AT PioWrite

The PioWrite command is used to set a PIO pin to high or low. A PIO pin may be set when configured as an input or output

Command	<code>AT PioWrite &lt;pio&gt; &lt;value&gt;</code>
<pio>	The PIO pin 0-15 to write
<value>	The value to write, 0 or 1
Example	<code>AT PioWrite 5 1</code> <code>-&gt; Pio[5] = 1</code>
Example	<code>AT PioWrite 18 0</code> <code>-&gt; ErrInvalidParam</code>

## AT Profiles

The Profile command enables or disables the A2DP, AVRCP, HFP, and SPP profiles available.

Command	<code>AT Profiles &lt;eeee/ddd (A2dp/Avrcp/Hfp/Spp)&gt;</code>
<e/d>	Enable/Disable for each profile A2dp/Avrcp/Hfp/Spp
Example	<code>AT Profiles EEDD -&gt; Profile A2dp[E] Avrcp[E] Hfp[D] Spp[D]</code>

## AT PsRead

The PsRead command reads the specified flash memory key. This command provides raw access to features and settings stored in flash memory. Different Keys will output different number of columns for improved formatting. Please contact technical support for additional information.

Command	<code>AT PsRead &lt;key&gt;</code>
<key>	Memory user key 0-49
Example	<code>AT PsRead 4 -&gt; key[4] words[14] -&gt; [PsRead] -&gt; [0x00FF] [0xFFFF] -&gt; [0xFFFF] [0xFF0A] -&gt; [0x1900] [0xFF00] -&gt; [0x0000] [0x0000] -&gt; [0x6085] [0x6085] -&gt; [0x0000] [0x0000] -&gt; [0x0000] [0xFFFF] -&gt; [PsRead End]</code>

## AT PsWrite

The PsWrite command writes data to the specified flash memory key. This command provides raw access to features and settings stored in flash memory. Please contact technical support for additional information.

Command	<code>AT PsWrite &lt;key&gt; &lt;data&gt;</code>
<key>	Memory user key 0-49
<data>	Hexadecimal key data
Example	<code>AT PsWrite 4 00FFFFFFFFF0A1900FF0000000000608560850000000000FFFF -&gt; Key[4] Words[14]</code>
Example	<code>AT PsWrite 4 00F -&gt; InvalidLength</code>

## AT RemoteCommand

The RemoteCommand command is used to enable command from a remote device via SPP. All commands enter from the remote device and all responses and system messages are sent back to the remote device. When disabled, an SPP connection will simply pass data to the HS where it is taken from the RFCOMM and redirected straight to the UART output. With RemoteCommand enabled, the data from the SPP device will be processed by the handler and all HS

responses will be directed to the remote device.

Note: System responses sent wirelessly to remote devices will have the “<->” prefix, indicating a remote response.

Command	<code>AT RemoteCommand</code>
Example	<code>AT RemoteCommand E</code> <code>-&gt; E</code>

## AT Reset

The Reset command will simply cold reset the device.

Command	<code>AT Reset</code>
Example	<code>AT Reset</code> <code>-&gt; [Reboot]</code>

## AT Restrict

The Restrict command limits connectivity to a single, specified device. All other device connection attempts are rejected.

If enabled, when the Restrict device address has not been specified, or is set to zero, then the next device to connect becomes the Restricted device. The specified address can be zeroed, and/or the Restrict can be Disabled.

This setting is Disabled by default, and is saved in flash memory, along with the Restricted device address.

Command	<code>AT Restrict &lt;d/address&gt;</code>
<d/address>	Accepts D to disable the feature, or an address to enable the feature.
Example	<code>AT Restrict 000000000000</code> <code>-&gt; 00:00:00:00:00:00</code>

## AT ShowSettings

The ShowSettings command indicates several of the configurable device settings.

Command	<code>AT ShowSettings</code>
Example	<pre> AT ShowSettings -&gt; [Settings] -&gt; E AutoAnswer -&gt; D AutoDiscoverable -&gt; E AutoReconnect -&gt; D Codec Aac -&gt; D Codec Aptx -&gt; D Codec AptxLL -&gt; E Codec Fast -&gt; D DeepSleep -&gt; E Profile A2dp -&gt; D Profile Avrcp -&gt; E Profile Hfp -&gt; D Profile Spp -&gt; E Restrict -&gt; MicInput DefaultGain[15] -&gt; VolumeInit A2dp[15] Hfp[15] -&gt; Discovery Timeout [90]sec -&gt; Idle Shutoff Timeout [7800]sec -&gt; LinklossReconnect [14]min -&gt; [Settings End] </pre>

## AT State

The State command allows the user to display the device’s current state of operation.

Command	<code>AT State</code>
Example	<pre> AT State -&gt; [Connected] </pre>

## AT TimeoutDisc

The TimeoutDisc command allows the user to adjust the time (in seconds) the device will last in discoverable mode. This ranges from 0 – 65535 seconds. Note that setting timeout as 0 disables discoverability.

Command	<code>AT TimeoutDisc &lt;sec&gt;</code>
<sec>	Seconds until timeout
Example	<pre> AT TimeoutDisc 1024 -&gt; 1024 seconds </pre>

## AT TimeoutIdle

The TimeoutIdle command adjusts the time in seconds in which the device will turn off if not connected. Ranges from 0 – 600 seconds. Note that 0 means it remains on indefinitely.

Command	<code>AT TimeoutIdle &lt;sec&gt;</code>
<sec>	Seconds until timeout
Example	<code>AT TimeoutIdle 1800</code> <code>-&gt; 1800 seconds</code>

## AT TimeoutLinkloss

The TimeoutLinkloss command allows the user to change the automatic timeout (in minutes) of reconnection for A2DP and HFP modes. The default is 5 minutes. Ranges from 0-250 minutes. Automatic timeout can also be enabled or disabled.

Command	<code>AT TimeoutLinkloss &lt;e/d&gt; &lt;mins&gt;*</code>
<e/d>	Enable or Disable
<mins>	Minutes until Timeout
Example	<code>AT TimeoutLinkloss E 20</code> <code>-&gt; E 20 minutes</code>

## AT Version

The Version command simply outputs the complete version. Version 8.0 followed by the specified Build version.

Command	<code>AT Version</code>
Example	<code>AT Version</code> <code>-&gt; kcAudioHS v8.2.0</code>

## AT Volume

The Volume command increments and decrements the currently volume level of the currently active profile (A2DP or HFP). Volume levels are saved in flash memory per device, as separate levels for each A2DP and HFP.

Command	<code>AT Volume &lt;+/-/vol&gt;</code>
<+/->	Either + increment volume, or - decrement volume
Example	<code>AT Volume +</code> <code>-&gt; +</code>
Example	<code>AT Volume -</code> <code>-&gt; -</code>

## AT VolumeInit

Sets the default volume and input gain levels for new A2DP and HFP connections. These levels are saved in flash memory. Using the '#' character in place of an argument will preserve its current level.

Command	<code>AT VolumeInit ault &lt;a2dp volume&gt; &lt;hfp volume&gt;</code>
<A2DP Vol>	Volume level of 0-15, or # to leave unchanged
<HFP Vol>	Volume level of 0-15, or # to leave unchanged
<Mic Gain>	Gain level of 0-22, or # to leave unchanged
Example	<code>AT VolumeInit -&gt; A2dp[14] Hfp[14]</code>
Example	<code>AT VolumeInit 10 # -&gt; A2dp[10] Hfp[14]</code>
Example	<code>AT VolumeInit # # -&gt; A2dp[10] Hfp[14]</code>

## Contact Information

KC Wirefree Corporation  
2640 W Medtronic Way  
Tempe, Arizona 85281

Phone	(602) 386-2640
Website	<a href="http://www.kcwirefree.com">www.kcwirefree.com</a>
Sales Support	<a href="mailto:info@kcwirefree.com">info@kcwirefree.com</a>
Technical Support	<a href="mailto:tech@kcwirefree.com">tech@kcwirefree.com</a>