



User Guide

kcSerial v2.4

Embedded Bluetooth Serial Port Profile Firmware

20 February 2010



Table of Contents

Preface.....	4
Features.....	4
Compatibility	5
Getting Started.....	5
Modes of Operation	7
Command Mode	7
Bypass Mode.....	7
Remote Command Mode.....	7
Smart Cable Features.....	8
SmartCableSetup.....	8
SmartCableReset.....	8
DeleteSmartCable	8
Power Saving Features.....	9
Deep Sleep Mode.....	9
UART Usage With Deep Sleep	9
Deep Sleep Blocking.....	9
Auto Sniff Mode	9
GPIO Features.....	10
GPIO Defaults	10
AIO Pins	10
GPIO Pin Designated Features	10
AT Command Summary	11
AT Command Reference	12
Syntax Notes	12
AT+ZV AIORead	12
AT+ZV Bond.....	12
AT+ZV Bypass	13
AT+ZV ChangeBaud.....	13
AT+ZV ChangeDefaultBaud	13
AT+ZV DeepSleep	14
AT+ZV DeepSleepBlocking.....	14
AT+ZV DefaultPinCode	14
AT+ZV DeleteSmartCable	15
AT+ZV DisconnectNotice.....	15
AT+ZV Discovery.....	15
AT+ZV DUNConnect	16

AT+ZV DUNDisconnect.....	16
AT+ZV EraseBondTable.....	17
AT+ZV FactoryReset.....	17
AT+ZV GPIOConfig.....	17
AT+ZV GPIORead.....	18
AT+ZV GPIOWrite.....	18
AT+ZV Help.....	18
AT+ZV HCIMode.....	18
AT+ZV HostEvent.....	19
AT+ZV IndicatorActivity.....	19
AT+ZV IndicatorConnection.....	20
AT+ZV IndicatorCpu.....	20
AT+ZV LocalName.....	20
AT+ZV RemoteCommand.....	20
AT+ZV Reset.....	21
AT+ZV SaveSettings.....	21
AT+ZV Security.....	22
AT+ZV ShowSettings.....	22
AT+ZV SmartCableReset.....	23
AT+ZV SmartCableSetup.....	23
AT+ZV SPPConnect.....	23
AT+ZV SPPDisconnect.....	24
AT+ZV UpdateInquiryScan.....	24
AT+ZV UpdatePageScan.....	24
AT+ZV Verbose.....	25
AT+ZV Version.....	25
Contact Information.....	26

Preface

Our kcSerial firmware is a fully embedded RS-232 serial cable replacement application that provides point-to-point wireless communication and control between two Bluetooth devices using the Serial Port Profile (SPP), or Dial Up Networking Profile (DUN). This document describes major modes of the firmware, and a detailed reference for every command.

Our kcSerial embedded Bluetooth Serial Port Profile firmware application is designed to operate KC Wirefree Bluetooth hardware modules using an easy to operate AT style text command interface. Devices deploying kcSerial firmware are designed to operate as a standalone wireless solution, and does not require a host controller. No additional software or drivers are required if connected to a personal computer or microprocessor controller device.

KC Wirefree Bluetooth devices using kcSerial are 100% compatible with and qualified as standard Bluetooth Serial Port Profile (SPP). kcSerial can communicate with any Bluetooth qualified device that supports Bluetooth SPP. However, the interface commands, controls, and several of the features described in this document are unique to kcSerial firmware.

Features

- Command Mode – an interactive mode of operation that accepts easy to use AT commands for operation and configuration control.
- Remote Command Mode – a command mode where a kcSerial remote unit will accept kcSerial AT commands wirelessly from any other Serial Bluetooth device.
- Bypass Mode – a kcSerial mode of operation implementing a wireless cable link that transmits and receives data without interference.
- Serial Port Profile (SPP) – Bluetooth compliant profile supporting both Client and Server application.
- Dial Up Networking (DUN) – Bluetooth compliant profile supporting Client application.
- Point-to-point connection – kcSerial supports a Bluetooth connection with one device at a time.
- Power conservation – deep sleep and sniff modes to minimize power consumption.
- UART interface – a 3 wire (TX, RX, Ground) or 5 wire (TX, RX, CTS, RTS, Ground) physical interface supports data rates from 1,200 bps to 3 Mbps.

Compatibility

kcSerial firmware offers fully qualified Bluetooth Serial Port Profile (SPP) and Dial Up Networking Profile (DUN), and can communicate with any Bluetooth device that supports those profiles. The kcSerial user interface commands and several special features described in this document are unique to kcSerial firmware.

kcSerial is Bluetooth v2.1+EDR firmware for CSR chipsets designed to be compatible with legacy firmware versions for Bluetooth v1.2 Zerial firmware for Zeevo/Broadcom chipsets. Due to changes in the way Bluetooth security has changed, some commands related to security have been modified.

Getting Started

Connect the kcSerial device UART to a PC serial port physically.

NOTE: If connecting from a PC, please be aware that the UART voltage levels on a PC ARE NOT compatible with the module which is running at +3.3V, therefore the RX,TX,RTS, and CTS signals need to be properly level shifted).

Start a terminal program, such as HyperTerminal or kcToolkit™, and configure the port settings. Default UART settings in kcSerial are 115,200 bps, 8 data bits, No parity bits, 1 stop bit, with hardware flow control. We recommend downloading our kcToolkit program, because this has dropdown lists of all available commands, and user is able to open multiple comports at once, so interaction between 2 devices can be easily seen. You can download this tool from the support page at www.kcwirefree.com

Once connected, cycle the power of the kcSerial module, and you will see a startup message:

```
AT-ZV -CommandMode-
```

```
AT-ZV BDAAddress 00043e3a3111
```

With kcToolkit, you can click the VERSION button, to send a version command, or use hyperterminal to type in the command "AT+ZV Version" followed by a carriage return.

```
AT+ZV Version
```

```
AT-ZV kcSerial v2.4 (Build xx)
```

You can now discover Bluetooth devices that are in discoverable mode by clicking on the DISCOVERY button in kcToolkit.

```
AT+ZV Discovery
```

```
AT-ZV InqPending
```

```
AT-ZV DiscoveryPending 2
```

```
AT-ZV Device 00043e3861ef "KCWirefreeDevice" all COD=0x0
```

```
AT-ZV Device 00043e32a3ba "WEP200" audio COD=0x200404
```

You can now attempt to make a SPP connection to a device (provided it supports the SPP profile).

```
AT+ZV SPPConnect 00043e3861ef
```

AT-ZV BondPending 00043e3861ef

AT-ZV SPPConnectionFailed

Whoops! This indicates that you are not a trusted device of the remote device, and you need to first BOND/PAIR to the remote device using the remote devices pin code. (Note: if Security is turned off on the remote device, then this is not necessary). So issue the Bond command:

AT+ZV Bond 00043e3861ef 1234

AT-ZV BondPending

AT-ZV BondOk 00043e3861ef

OK. Now we can try to connect, since we have established a trust between the two devices (Note: once we BOND once, we should not have to BOND again).

AT+ZV SPPConnect 00043e3861ef

AT-ZV ConnectionUp

AT-ZV -BypassMode-

Now we are in Bypass Mode, and any data we send into the UART, will appear on the remote units UART, and vice versa. Bypass Mode operates as a wireless serial cable.

Once you are in Bypass Mode, you may decide that you want to issue another command (possibly to close the connection) To accomplish this, you must first send a special sequence of characters called the Escape Sequence (no carriage return or linefeed needed) to get you back into Command Mode

So send these 6 characters [Hex: 5E 23 5E 24 5E 25]:

^#^\$^%

You will remain connected, but you have now switched back into command mode. You can now issue any AT style command to the command interpreter.

If you want to close the current connection, issue the command:

AT+ZV SppDisconnect

If you want to return to Bypass Mode, issue the command:

AT+ZV Bypass

Of course there are many more commands that you can use, but the above commands are essential to establish a basic SPP connection.

Modes of Operation

Command Mode

The Command Mode is the default mode when not wirelessly connected. This interactive mode accepts all kcSerial operational, configuration, and control commands received from the local device UART port in ASCII text format. An extensive list of AT style text commands is available for execution in Command Mode. Connections can be initiated from this mode.

Command Mode can also be used while wirelessly connected. A special Escape Sequence must be sent while in Bypass Mode in order to switch into Command Mode when wirelessly connected. A simple AT command ([AT+ZV Bypass](#)) will switch the device back to Bypass Mode.

Bypass Mode

The Bypass Mode is the default mode when wirelessly connected, allowing completely transparent data traffic between Bluetooth devices. In this data transfer mode, AT Commands will not be recognized or executed. All UART input is simply transmitted as data bytes to the remote device.

In order to switch into Command Mode, the data stream is monitored for a special Escape Sequence coming from the local UART port. Also, the device monitors incoming wireless data streams from a remote device looking for a special Remote Command Mode character escape sequence.

Command Mode escape sequence: `^#^$^%` (hex: 5E 23 5E 24 5E 25)

Remote Command Mode escape sequence: `^#^$^*` (hex: 5E 23 5E 24 5E 2A)

If Command Mode sequence is detected (and the Command Mode Escape feature is Enabled), the local kcSerial device will switch out of Bypass Mode and into Command Mode. Any characters or data following the special sequence will be interpreted as commands by Command Mode.

If Remote Command Mode sequence is detected in the data stream received by a remote kcSerial device (and the Remote Command Mode Escape feature is Enabled), the remote kcSerial device will switch out of Bypass Mode and into Command Mode. The local device will stay in Bypass Mode, transparently transmitting data to the remote device. Any subsequent data received wirelessly will be interpreted and executed by the remote device as AT commands while it remains in Command Mode.

Remote Command Mode

Our Remote Command Mode is a unique and powerful remote control mode only available from KC Wirefree. Any serial Bluetooth device can wirelessly send our AT commands to a remotely connected kcSerial device. Once a wireless connection is established, the Remote Command Mode escape sequence can be sent to a remote kcSerial device, where it will switch into Remote Command Mode (if Enabled). Remote Command Mode is similar to Command Mode, except that the AT commands are received wirelessly and executed by the remote kcSerial device. Response messages and readings from the remote kcSerial device are transmitted wirelessly to the local unit issuing the commands. A simple AT command ([AT+ZV Bypass](#)) can be sent, and the remote device will switch back into Bypass Mode. Note, the local device must be in Bypass Mode or a similar transparent data transfer mode, so that the AT command strings are sent wirelessly to the remote kcSerial device without local interpretation or execution.

Smart Cable Features

This feature provides automatic connections for cable replacement application. The following AT Commands are used for SmartCable implementation:

`AT+ZV DeleteSmartCable`

`AT+ZV SmartCableReset [e/d]`

`AT+ZV SmartCableSetup [bdaddr] [attempts] [interval]`

SmartCableSetup

Usage

- Automatically establishes a SPP link to its designated remote device.
- The designated device is paged and retried up to the retry attempt limit setting, if it is unable to connect initially.
- If a link is disconnected, the Smart Cable feature will automatically re-connect the link without user interaction.
- A wait interval is inserted between automatic page attempts.
- Only point-to-point connections are supported.
- An optional feature allows GPIO 7 to quick connect to the remote device as well as reset the current page retry attempt counter.

The SmartCableSetup AT command automatically saves settings in memory. These new settings are loaded after the next reset. The DeleteSmartCable AT command deletes these stored settings and deactivates the Smart Cable feature for the remainder of the session.

The GPIO 7 attempt reset feature resets the current connection attempt counter. If the attempt parameter is set to 0 attempts, a single attempt is initiated.

SmartCableReset

This feature enables/disables starting the SmartCable feature from a hardware GPIO pin. This is useful for providing an automatic connection button.

DeleteSmartCable

This command will disable the SmartCable feature by erasing the SmartCableSetup parameters from flash memory.

Power Saving Features

kcSerial devices support various features, which allow low power operation over a range of scenarios. This section will discuss the Deep Sleep Mode, Sniff, and Auto Sniff features and how they may be effectively used.

Deep Sleep Mode

In kcSerial, the basis for low power operation is Deep Sleep Mode, DSM. This feature temporarily halt's the chip's operation by stopping the main crystal and switching to the low power 32KHz oscillator instead. When enabled, DSM automatically enters this halt state whenever possible. Scheduled CPU activity, GPIO interrupts, and UART requests will automatically resume active mode operation.

UART Usage With Deep Sleep

When a UART is connected, the CTS line on the device's UART connector must not be asserted in order to allow DSM. The host device design must consider this when DSM is desired. In order to wake up from DSM, the host must pulse the device's CTS line and wait 10ms for the device to become active once again.

Deep Sleep Blocking

kcSerial supports a Deep Sleep Blocking feature using GPIO 5. When enabled, an active signal on GPIO 5 will temporarily prevent or block DSM. Normal DSM operation will resume when this signal is no longer asserted.

Auto Sniff Mode

This feature dynamically enables and disables sniff mode depending on a link's communication needs. Two dynamic configuration parameters control this feature: Sniff Interval and Inactivity Timeout. The Sniff Interval is the number of sniff poll interval slots that the sniff mode uses. The Inactivity Timeout is the number of seconds that the link will stay active after data is received or transmitted.

GPIO Features

Several digital General Programmable Input Output (GPIO) pins are available to support kcSerial features (when enabled), with remaining pins available for user applications. Several AT Commands are used to control operation of these pins, including AIORead, GPIOConfig, GPIORead, and GPIOWrite. Not all pins are available on all modules.

GPIO Defaults

Unassigned GPIO pins are defaulted to inputs, with weak internal pull-down.

AIO Pins

Up to 2 Analog Input Output pins may be available depending on the module. The Analog Pins are inputs by default, which digitize and read analog voltages 0 – 1.5V with 8 bit resolution.

GPIO Pin Designated Features

This table gives a summary of the kcSerial interface's assignment of certain GPIO pins. Only the GPIO pins directly used by the kcSerial interface are considered in this table; for complete GPIO assignments see the applicable device Data Sheet.

kcSerial Feature	GPIO	Input/Output	Low	High
Indicator Connection	0	Output	Not Connected	Connected
Indicator CPU	1	Output	Deep Sleep	CPU On
Indicator Activity	4	Output	No Data Activity	Data Activity (100ms pulses)
DeepSleepBlocking	5	Input	Allow	Block
SmartCableReset	7	Input	N/A	Reset

AT Command Summary

Operation Commands

AT+ZV AIORead
AT+ZV Bond
AT+ZV Bypass
AT+ZV Discovery
AT+ZV DUNConnect
AT+ZV DUNDisconnect
AT+ZV GPIORead
AT+ZV GPIOWrite
AT+ZV Reset
AT+ZV SPPConnect
AT+ZV SPPDisconnect

Configuration Commands

AT+ZV Build
AT+ZV ChangeBaud
AT+ZV ChangeDefaultBaud
AT+ZV DeepSleep
AT+ZV DeepSleepBlocking
AT+ZV DefaultLocalName
AT+ZV DefaultPinCode
AT+ZV DeleteSmartCable
AT+ZV DisconnectNotice
AT+ZV EraseBondTable
AT+ZV FactoryReset
AT+ZV GPIOConfig
AT+ZV Help
AT+ZV HCIMode
AT+ZV HostEvent
AT+ZV IndicatorActivity
AT+ZV IndicatorConnection
AT+ZV IndicatorCpu
AT+ZV LocalName
AT+ZV RemoteCommand
AT+ZV SaveSettings
AT+ZV ShowSettings
AT+ZV Security
AT+ZV SmartCableReset
AT+ZV SmartCableSetup
AT+ZV UpdateInquiryScan
AT+ZV UpdatePageScan
AT+ZV Verbose
AT+ZV Version

▲ kcSerial v2.2 compatibility:

Previous kcSerial command prefixes are recognized: [AT+ZV](#) and [AT+KC](#).

The following kcSerial v2.2 legacy AT commands and features are not available in kcSerial v2.4:

DisableBond, EnableBond, ExitSniff, Hold, Park, RemoteCmdDisconnect, Sniff, StreamingSerial

The following AT commands are new in kcSerial v2.4:

AIORead, Build, DefaultPinCode, Help, IndicatorActivity, IndicatorConnection, IndicatorCpu, SaveSettings, ShowSettings, ShowStatus, Verbose

AT Command Reference

Syntax Notes

- Commands are accepted as simple text strings via UART
- AT Commands can be prefaced by AT+ZV or AT+KC
- System responses are in the form of AT-ZV
- Responses coming via RemoteCommand Mode are AT*ZV
- Commands and arguments are case insensitive (except for device name changes)
- Command lines must be marked as end of line by <CR> or <CR><LF>, which is [Hex 0D or 0D 0A]
- Response lines are followed by <CR><LF> which is [Hex 0D 0A]
- Arguments [e/d] are a choice of "E" to Enable Feature, or "D" to Disable Feature

AT+ZV AIORead

The AIORead command will digitize and read an analog voltage level 0 – 1.5V. There may be up to 2 AIO pins available.

Syntax

`AT+ZV AIORead [aio pin]`

Where [aio pin] is AIO pin 0 or 1

Responses

`AT-ZV AI00 [reading]mV`

AT+ZV Bond

The Bond command is used to initiate pairing with a specified device. A PIN is required with this command, and must match the PIN of the remote device. This command does not establish a connection, but obtains and saves the necessary pairing information when secured connections are required by either device. Only one successful bond is required for any particular remote device. A Bond attempt can fail for several reasons, including incorrect pin code, bonding disallowed by remote device, or bond entries have been deleted by either one of the paired devices. Please refer to the Bonding and Pairing section of this User Guide for more complete information regarding this process.

Syntax

`AT+ZV Bond [bdaddr] [pin]`

Where [bdaddr] is the BD Address of the remote device with which to bond and [pin] is the pin code to use (up to 16 characters).

Responses

`AT-ZV BondPending [bdaddr]`

If the operation is successful, the response is:

AT-ZV BondOk

If the operation fails, the response is:

AT-ZV BondFail

AT+ZV Bypass

The Bypass command is used to return the kcSerial interface to the bypass mode, if a connection is still available. The possible use for this is to change a setting after a connection has been made (such as the UART baud rate). If the kcSerial interface does not have a connection, it will respond as if the connection is down.

Syntax

AT+ZV Bypass

Responses

AT-ZV -BypassMode-

If there is currently no connection, the response is:

AT-ZV ConnectionDown

AT+ZV ChangeBaud

The host sends the ChangeBaud command in order to change the local UART speed to a new speed identified by the host. This setting will only remain in effect during the current session - until reset.

Syntax

AT+ZV ChangeBaud [rate]

Where [rate] is the new baud rate:

1,200 2,400 4,800 9,600 19,200 38,400 57,600 115,200 230,400 460,800 921,600 1,382,400

Responses

AT-ZV Baudrate Changed

The actual change will not occur until the response has been completely transmitted.

AT+ZV ChangeDefaultBaud

The host sends the ChangeDefaultBaud command in order to change the default UART speed to a new speed identified by the host. This command is used to override the default baud rate from the Dynamic Configuration script so that the device does not require reprogramming to update this setting. The new baud rate is updated permanently until the device is either re-programmed or another ChangeDefaultBaud command is issued. The baud rate specified in the command will not take effect until the device is reset. To change the baud rate of the current session, use the ChangeBaud command.

Syntax

AT+ZV ChangeDefaultBaud [rate]

Where [rate] is the new baud rate:

1,200 2,400 4,800 9,600 19,200 38,400 57,600 115,200 230,400 460,800 921,600 1,382,400

Responses

AT-ZV Baudrate Changed

AT+ZV DeepSleep

The DeepSleep command enables the low power chip sleep mode for operational power savings. Please refer to the power savings section in this User Guide for additional notes regarding Deep Sleep mode. The SaveSettings command will save this setting (and others) to flash memory.

Syntax

AT+ZV DeepSleep [e/d]

Responses

AT-ZV Enabled DeepSleep

AT+ZV DeepSleepBlocking

The DeepSleepBlocking command allows GPIO control over Deep Sleep mode. Pulling the assigned GPIO high will block the device from switching into sleep mode, while pulling the GPIO low allows normal operation of sleep mode whenever possible. If this feature is enabled, then the assigned GPIO pin is unavailable for other use. The SaveSettings command will save this setting (and others) to flash memory.

Syntax

AT+ZV DeepSleepBlocking [e/d]

Responses

AT-ZV Enabled DeepSleepBlocking

AT+ZV DefaultPinCode

The DefaultPinCode command allows changing the Pin code for pairing. The Pin code is only required when a secure connection is requested by either device. The Pin code will be sent automatically to remote devices requesting a Pin code. Also, a Pin code matching this default Pin code must be received from remote devices pairing to this device when security is enabled. Upon success, the new Pin code is stored in flash memory.

The current Pin can be viewed using the AT ShowSettings command.

Syntax

AT+ZV DefaultPinCode [new pin] [old pin]

Where [pin] is a 1-16 digit alphanumeric pin code.

Responses

AT-ZV PinCodeChanged

AT+ZV DeleteSmartCable

The DeleteSmartCable command removes the current Smart Cable settings that were entered using the SmartCableSetup command, but not the setting from the dynamic configuration. The Smart Cable will then be deactivated for the remainder of this session. Upon reset, if a dynamic configuration for a Smart Cable exists, it will be activated. If there is no dynamic configuration Smart Cable setup, then this feature will remain deactivated.

Syntax

`AT+ZV DeleteSmartCable`

Responses

`AT-ZV DeleteSmartCableDone`

AT+ZV DisconnectNotice

The DisconnectNotice command sends a "###NO CARRIER" disconnection notification as soon as the connection is down. The remote device disconnects, and the local device is still in BypassMode. This command is primarily intended for backwards compatibility for previous kcSerial versions. The SaveSettings command will save this setting (and others) to flash memory.

Syntax

`AT+ZV DisconnectNotice [e/d]`

Responses

`AT-ZV Enabled DisconnectNotice`

AT+ZV Discovery

The Discovery command is used to initiate a device discovery, which has 2 stages. The number of devices listed is limited to 10 total.

An initial message, InqPending, is issued when starting the discovery process.

During the first stage, the total number of devices found will be reported (devices that have their discovery turned on). A message is issued, DiscoveryPending <num>, indicating the total number of devices found.

During the second stage, a connection is made to each discovered device, in order to obtain the device name, and class of device parameter. Messages are issued one at a time, for each device, after obtaining the information requested, and includes the BDAAddress, device name, CoD value, and CoD major class type.

If a discovered device is not connectable, then no message is printed for that device.

Syntax

`AT+ZV Discovery`

`AT+ZV Discovery [cod]`

Where devices are filtered on the [cod] class:

All (default) Misc Computer Phone LAN Peripheral Imaging Unclassified

Responses

AT-ZV InqPending

Once the initial inquiry is complete and discovery has been started, the response is:

AT-ZV DiscoveryPending [num]

Where [num] is the number of devices found, in decimal (up to 10 will be reported).

For each name or service name request that is successful, the response uses the returned names in the following format.

AT-ZV Device [bdaddr] [name] [codname] CoD=[codval]

Where [bdaddr] is the BDAddresss of a discovered device

[name] is the device name in double quotes “ ”

[codname] is a string without quotes and is the class of device category: All, Misc, Computer, Phone, LAN, Peripheral, Imaging, or Unclassified

[codval] is the hexadecimal value of the Class of Device

AT+ZV DUNConnect

The DUNConnect command is used to initiate a connection with the specified device. The remote BD address must be specified. The remote Service is optional. If not specified, the first registered DUN service will be used by default.

Syntax

AT+ZV DUNConnect [bdaddr]

Where [bdaddr] is the remote devices BD Address to page.

Responses

AT-ZV ConnectionUp

AT-ZV -BypassMode-

If the connection cannot be completed, the response is:

AT-ZV DUNConnectionClosed

AT+ZV DUNDisconnect

The DUNDisconnect command is used to terminate a connection with the remote device.

Syntax

AT+ZV DUNDisconnect

Responses

AT-ZV DUNConnectionClosed

AT+ZV EraseBondTable

The EraseBondTable command is used to erase all of the bonded device entries. Single devices cannot be erased with this command

Syntax

```
AT+ZV EraseBondTable
```

Responses

```
AT-ZV BondTableErased
```

AT+ZV FactoryReset

The FactoryReset command will replace all user settings with the kcSerial default settings. The SaveSettings command must be issued to save all of the restored settings in flash memory.

Syntax

```
AT+ZV FactoryReset
```

Responses

```
AT-ZV --- Settings ---  
AT-ZV BDAAddress <bdaddr>  
AT-ZV Firmware kcSerial v2.4 (Build xx)  
AT-ZV LocalName kcSerial  
AT-ZV Bluetooth v2.1  
AT-ZV ClassOfDevice 0x1f00 unclassified  
AT-ZV Enabled HostEvent  
AT-ZV Enabled IndicatorActivity  
AT-ZV Enabled IndicatorConnection  
AT-ZV Enabled IndicatorCpu  
AT-ZV Disabled DeepSleepBlockingIO  
AT-ZV Disabled SmartCableReset  
AT-ZV Disabled DeepSleep  
AT-ZV Enabled AllowBonding  
AT-ZV Disabled DisconnectNotice  
AT-ZV Enabled RemoteCommand  
AT-ZV Disabled VerboseMode  
AT-ZV Disabled Security  
AT-ZV PinCode 1234  
AT-ZV Factory Defaults Complete
```

AT+ZV GPIOConfig

The GPIOConfig command is used to configure a GPIO pin to input or output.

Syntax

```
AT+ZV GPIOConfig [gpio pin] [configuration]
```

Where [gpio pin] is the Pin number, 0 – 15, of the desired GPIO to configure.

[Configuration] is letter “i” for input or letter “o” for output.

Responses

[AT-ZV GPIOConfigDone](#)

AT+ZV GPIORead

The GPIORead command is used to read a GPIO pin. A GPIO may be read while configured as either an input or output.

Syntax

[AT+ZV GPIORead \[gpio pin\]](#)

Where [gpio pin] is the Pin number, 0 – 15, of the desired GPIO to read.

Responses

[AT-ZV GPIOReadDone \[result\]](#)

Where [result] is either a 1 to indicate high, or 0 to indicate low.

AT+ZV GPIOWrite

The GPIOWrite command is used to set a GPIO pin to high or low. A GPIO may only be set when configured as an output.

Syntax

[AT+ZV GPIOWrite \[gpio pin\] \[Setting\]](#)

Where [gpio pin] is the Pin number, 0 – 15, of the desired GPIO to read. [Setting] is a 1 to set a pin to high and a 0 to set a pin to low.

Responses

[AT-ZV GPIOWriteDone](#)

AT+ZV Help

The Help command lists all available commands. Some commands in this list are experimental or internal debugging use only.

Syntax

[AT+ZV Help](#)

Responses

A three column list of AT commands.

AT+ZV HCIMode

The HciMode command switches the chip into raw HCI level access. The direct HCI level access is used for firmware upgrading, RF testing options, flash memory access utilities, and other chip manufacturer capabilities. External software applications are required to interface with HCI mode.

Upon power up, or reset, the default boot mode is always the UART-Serial interface.

After successfully issuing the AT HCIMode command, the device will immediately restart into the specified mode. There is no response message. But you will see HCI data coming out of the UART

Note: HCI (Host Controller Interface) is defined by Bluetooth SIG, which provides a common and compatible interface for all Bluetooth chip manufacturers. Most common usage of HCI mode is either by a USB dongle, or a built in Bluetooth module for personal computers. The computer must provide a Bluetooth "Stack" software application that provides a complex user interface to properly operate a Bluetooth device using HCI mode.

Syntax

AT+ZV HCIMode

Responses

No response, chip immediately restarts in HCI mode.

AT+ZV HostEvent

The HostEvent command is used to enable/disable the host notification strings. This will override the default setting in the dynamic configuration only for the current session; until reset.

Use SaveSettings to store new setting in flash memory.

Syntax

AT+ZV HostEvent [e/d]

Where [e/d] is "e" to enable, or "d" to disable.

Responses

AT-ZV HostEvent Enabled

If the feature is successfully disabled there is no response because the events have been disabled.

AT+ZV IndicatorActivity

The IndicatorActivity command is used to enable/disable the activity indicator feature on GPIO 4. The GPIO will pulse 100ms on and off when data is received locally from the UART, or wirelessly from a connection. Disabling this feature will make the GPIO available for general use.

Use SaveSettings to store new setting in flash memory.

Syntax

AT+ZV IndicatorActivity [e/d]

Where [e/d] is "e" to enable, or "d" to disable.

Responses

AT-ZV Enabled IndicatorActivity

AT+ZV IndicatorConnection

The IndicatorConnection command is used to enable/disable the connection indicator feature on GPIO 0. The GPIO will be high when wirelessly connected. Disabling this feature will make the GPIO available for general use.

Use SaveSettings to store new setting in flash memory.

Syntax

`AT+ZV IndicatorConnection [e/d]`

Where [e/d] is "e" to enable, or "d" to disable.

Responses

`AT-ZV Enabled IndicatorConnection`

AT+ZV IndicatorCpu

The IndicatorCpu command is used to enable/disable the connection indicator feature on GPIO 1. The GPIO will be high when device is operating normally. This pin will go low during DeepSleep. Disabling this feature will make the GPIO available for general use.

Use SaveSettings to store new setting in flash memory.

Syntax

`AT+ZV IndicatorCpu [e/d]`

Where [e/d] is "e" to enable, or "d" to disable.

Responses

`AT-ZV Enabled IndicatorCpu`

AT+ZV LocalName

The LocalName command is used to temporarily set the name of the device to the name that is reported during device discoveries. For permanent device name change, use AT+ZV DefaultLocalName.

Syntax

`AT+ZV LocalName [name]`

Where [name] is a string for the new local name (up to 50 characters). The space character is allowed; the name is assumed to be all text up to the end of the command.

Responses

`AT-ZV LocalNameOk`

AT+ZV RemoteCommand

The RemoteCommand command is used to enable/disable the Remote Command Mode. This setting is stored in persistent memory, and will be retained after each reset. Additionally, the new setting will take effect upon the next device reset.

Use SaveSettings to store new setting in flash memory.

Syntax

```
AT+ZV RemoteCommand [e/d]
```

Where [e/d] is "e" to enable, or "d" to disable.

Responses

```
AT-ZV RemoteCommand E
```

AT+ZV Reset

The Reset command is used to reset the kcSerial interface. This is provided in the event that a host application wants to perform a software reset for error recovery. There is a response prior to reset to verify the command was received by the kcSerial interface.

Syntax

```
AT+ZV Reset
```

Responses

```
AT-ZV ResetPending
```

AT+ZV SaveSettings

The SaveSettings command is used to store configuration data to the non-volatile memory. So the next time the device is restarted, all of the parameters in the non-volatile memory will be used.

Syntax

```
AT+ZV SaveSettings
```

Responses

```
AT-ZV --- Settings ---  
AT-ZV BDAAddress <bdaddr>  
AT-ZV Firmware KCSerialVer 2.4 KC21 (Build xx)  
AT-ZV LocalName KcSerial  
AT-ZV Bluetooth v2.1  
AT-ZV ClassOfDevice 0x1f00 unclassified  
AT-ZV Enabled HostEvent  
AT-ZV Enabled IndicatorActivity  
AT-ZV Enabled IndicatorConnection  
AT-ZV Enabled IndicatorCpu  
AT-ZV Disabled DeepSleepBlockingIO  
AT-ZV Disabled SmartCableIOReset  
AT-ZV Disabled DeepSleep  
AT-ZV Enabled AllowBonding  
AT-ZV Disabled DisconnectNotice  
AT-ZV Enabled RemoteCommand  
AT-ZV Disabled VerboseMode  
AT-ZV Disabled Security  
AT-ZV PinCode 1234  
AT-ZV Enabled Discoverable
```

```
AT-ZV InquiryInterval 18
AT-ZV InquiryWindow 2048
AT-ZV Enabled Connectable
AT-ZV PageInterval 18
AT-ZV PageWindow 2048
AT-ZV SaveSettings Done
```

AT+ZV Security

The Security command is used to enable security of the local device. Bluetooth v2.1 security mode 4 is used. Enabling security requires incoming and outgoing connections to be properly bonded, and disabling security allows connections without bonding. Security is disabled by default.

Use SaveSettings to store new setting in flash memory.

▲ kcSerial v2.2 compatibility: The previous arguments of “none” or “link” are also accepted.

Syntax

```
AT+ZV Security [e/d]
```

Responses

```
AT-ZV Security Enabled
```

```
AT-ZV Security Disabled
```

AT+ZV ShowSettings

This feature will show the user all of the currently configured settings.

Syntax

```
AT+ZV ShowSettings
```

Responses

```
AT-ZV --- Settings ---
AT-ZV BDAAddress <bdaddr>
AT-ZV Firmware KCSerialVer 2.4 KC21 (Build xx)
AT-ZV LocalName KcSerial
AT-ZV Bluetooth v2.1
AT-ZV ClassOfDevice 0x1f00 unclassified
AT-ZV Enabled HostEvent
AT-ZV Enabled IndicatorActivity
AT-ZV Enabled IndicatorConnection
AT-ZV Enabled IndicatorCpu
AT-ZV Disabled DeepSleepBlockingIO
AT-ZV Disabled SmartCableIOReset
AT-ZV Disabled DeepSleep
AT-ZV Enabled AllowBonding
AT-ZV Disabled DisconnectNotice
AT-ZV Enabled RemoteCommand
AT-ZV Disabled VerboseMode
AT-ZV Disabled Security
AT-ZV PinCode 1234
```

AT-ZV Enabled Discoverable
AT-ZV InquiryInterval 18
AT-ZV InquiryWindow 2048
AT-ZV Enabled Connectable
AT-ZV PageInterval 18
AT-ZV PageWindow 2048

AT+ZV SmartCableReset

The SmartCableReset command allows a high signal on the assigned PIO 7 pin to restart the SmartCable feature. If the SmartCable feature is already running and making connection attempts, it will reset the number of attempts counter, and restart. If the SmartCableSetup <attempts> parameter is set to 0 or 1, this feature will start a single connection attempt each time the PIO receives a high input.

Use SaveSettings to store new setting in flash memory.

Syntax

AT+ZV SmartCableReset [e/d]

Responses

AT-ZV Enabled SmartCableReset

AT+ZV SmartCableSetup

The SmartCableSetup command provides automatic connections. A remote device address is specified along with the number of connection attempts, and the interval between attempts. Warning: unlimited connection attempts with a minimum wait time can be difficult to manage if the remote device is not present, because the device does not stop issuing rapid connection attempts and does not allow sufficient time to change the settings, or issue other AT commands.

A value of 0 <attempts> will simply store the connection information without automatically connecting. However, one manual connection attempt will be started when using the AT SmartCableReset feature.

Syntax

AT+ZV SmartCableSetup [bdaddr] [attempts] [interval]

Where [bdaddr] is the BD address of the remote device to page and attempt to connect.

[attempts] 0 – 999 is the number of pages that will be attempted to the specified device until a connection is successful. A value of 0 will not automatically page the remote device, however, GPIO 7 may be asserted to manually send a page. A value of 1000 will perform unlimited pages until connected.

[interval] 1-1000 is the number of 100ms intervals (0.1sec to 100 sec) between page attempts. This interval is in addition to the amount of time required by the page attempt itself.

Responses

AT-ZV SmartCableConfigDone

AT+ZV SPPConnect

The SPPConnect command is used to initiate a connection with the specified device. The remote BD address must be specified.

Syntax

`AT+ZV SPPConnect [bdaddr]`

Where [bdaddr] is the remote devices BD Address to page.

Responses

`AT-ZV ConnectionUp`

`AT-ZV -BypassMode-`

If the connection cannot be completed, the response is:

`AT-ZV SPPConnectionClosed`

AT+ZV SPPDisconnect

The SPPDisconnect command is used to terminate a connection with the remote device.

Syntax

`AT+ZV SPPDisconnect`

Responses

`AT-ZV SPPConnectionClosed`

AT+ZV UpdateInquiryScan

The UpdateInquiryScan command is used to modify the interval and window time slot parameters used to accept incoming inquiries from other devices. More power is consumed when more time is allocated scanning for incoming inquiries. A Bluetooth slot is 625 μ s.

Syntax

`AT+ZV UpdateInquiryScan [mode] [duration] [interval]`

Where [mode] is the discoverable mode: 0 = non discoverable, 2 = discoverable

[duration] is the scan length in slots; 18 to 4096. The default duration is 18 slots.

[interval] is the period between scans in slots; 18 to 4096. The default interval is 2048 slots

Responses

`AT-ZV InquiryScanUpdateDone`

AT+ZV UpdatePageScan

The UpdatePageScan command is used to modify the interval and window time slot parameters used to accept incoming connections from other devices. More power is consumed when more time is allocated scanning for incoming connections. A Bluetooth time slot is 625 μ s.

Syntax

`AT+ZV UpdatePageScan [mode] [duration] [interval]`

Where [mode] is the connectable mode: 0 = non connectable, 1 = connectable

[duration] is the scan length in slots; 18 to 4096. The default duration is 18 slots.

[interval] is the period between scans in slots; 18 to 4096. The default interval is 2048 slots

Responses

`AT-ZV PageScanUpdateDone`

AT+ZV Verbose

The Verbose command enables a lot of internal debug messages to be printed.

Syntax

`AT+ZV Verbose [e/d]`

Responses

If the operation is successful, the response is:

`AT-ZV Enabled VerboseMode`

AT+ZV Version

The Version command is used to return the current version of the kcSerial interface.

Syntax

`AT+ZV Version`

Responses

Contact Information

KC Wirefree
2640 W Medtronic Way
Tempe, Arizona 85281

(602) 386-2640 Phone

(602) 386-2642 Fax

info@kcwirefree.com

tech@kcwirefree.com