

Setting Up Host Hardware and Software to Support Wideband Speech

Associated Part Family: **CYW4330**

This Application Note describes how to configure the hardware and software on the host side to support wideband speech in conjunction with the Cypress® CYW4330, a single-chip IEEE 802.11™ a/b/g/n MAC/baseband/radio with integrated Bluetooth 4.0 + HS and FM transceiver. It is intended for engineers and designers who are incorporating support for wideband speech in designs that include the CYW4330.

1 About This Document

1.1 Cypress Part Numbering Scheme

Cypress is converting the acquired IoT part numbers from Broadcom to the Cypress part numbering scheme. Due to this conversion, there is no change in form, fit, or function as a result of offering the device with Cypress part number marking. The table provides Cypress ordering part number that matches an existing IoT part number.

Table 1. Mapping Table for Part Number between Broadcom and Cypress

Broadcom Part Number	Cypress Part Number
BCM4330	CYW4330

1.2 Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use. For a more complete list of acronyms and other terms used in Cypress documents, go to: <http://www.cypress.com/glossary>.

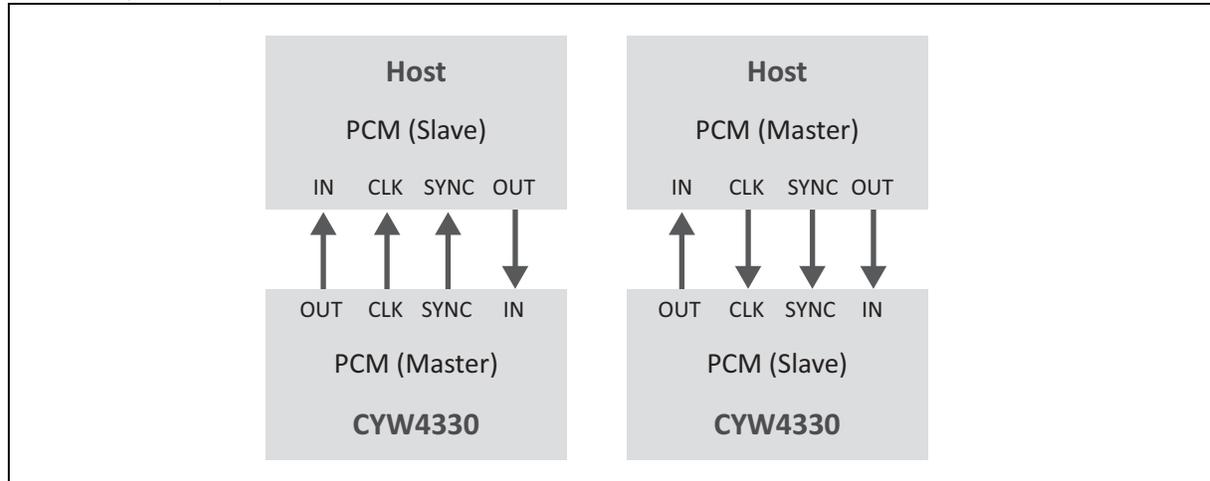
2 CYW4330/Host Transport Connection

The CYW4330 supports a variety of transports for wideband speech applications, including PCM, I²S, UART, and others. In this application note, we use the PCM transport as an example to describe how to configure the CYW4330 and the corresponding host hardware and software for wideband speech applications. [Figure 1](#) shows two typical PCM transport connections between the host and CYW4330, depending on who provides the PCM clock and SYNC signals.

3 IoT Resources

Cypress provides a wealth of data at <http://www.cypress.com/internet-things-iot> to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (<http://community.cypress.com/>).

Figure 1. Typical PCM Transport Connections Between the Host and the CYW4330



4 Host Stack Requirements For Wideband Speech

Wideband speech requires a greater audio frequency bandwidth (50–7000 Hz) compared to narrowband speech (200–3300 Hz). According to the Nyquist-Shannon sampling theorem, wideband speech also needs a higher audio sample rate (16 kHz) than narrowband speech (8 kHz). Consequently, wideband speech has minimum requirements for transport data rate and Bluetooth packet types. If the audio sample is 16-bit, the PCM transport rate has to be at least 256 Kbps, and the CYW4330 must use EV3 and 2-EV3 packets for wideband speech applications (see [Reference \[2\] on page 4](#) for more information on these packets). In addition to the settings above, the host stack will also need to send Host Control Interface (HCI) commands to configure CYW4330 audio and communication settings for wideband speech applications.

For wideband speech applications, the host stack is responsible for ensuring that the following settings are in effect:

- The host PCM hardware is set in the correct mode (master or slave).
- The host PCM SYNC clock rate is 16 kHz
- The PCM CLK is set to 256 kHz or above.

The host stack must send a set of HCI commands to configure the CYW4330 for Wideband Speech mode (see [Enabling Wideband Speech on the CYW4330 on page 2](#)).

Note: The host stack must keep track of the audio application mode (narrowband or wideband speech) before the SCO/eSCO connection is set up, because for different audio applications, the host stack needs to configure the CYW4330 using various HCI commands. This is easy to do if the SCO/eSCO connection is initialized from the CYW4330 host stack. However, if the SCO/eSCO connection is initiated from the peer device, the CYW4330 host stack must be able to detect the audio application mode on the peer device before it will allowed the CYW4330 to accept the SCO/eSCO connection.

Note: Since narrowband and wideband speech usually share the same audio transport between the host and CYW4330, and since wideband speech and narrowband speech require different SYNC clock rates (16 kHz for wideband; 8 kHz for narrowband), the host stack must configure the SYNC clock rate on its transport as well as the CYW4330 transport when the speech mode changes. The CYW4330 cannot automatically detect the SYNC clock rate, even if its transport is configured as a slave.

5 Enabling Wideband Speech on the CYW4330

Once the host PCM hardware is set up correctly, the host stack can issue the HCI commands required to initialize the CYW4330 for wideband speech audio applications with a connected peer device (for example, a Bluetooth headset that supports wideband speech). These HCI commands must be entered in the following order:

1. Enable_WBS ([page 3](#))
2. Write_I2SPCM_Interface_Param ([page 3](#))
3. Write_SCO_PCM_Int_Param ([page 4](#))

4. Setup_Synchronous_Connection (page 6)

The first three commands are Cypress vendor-specific commands. The fourth command is a standard Bluetooth command that is defined in the Bluetooth Core Specification (see [Reference \[2\] on page 4](#)).

Note: Some of the settings in commands 2 and 3 above are overlapped, such as `CLock_Rate` and the role of the PCM interface (`Is_Master`, `Sync_Mode`, and `Clk_mode`). The BT controller only takes into account the settings in the later command (command 3). It is highly recommended to use the same settings for commands 2 and 3 (such that the command order is not a concern), and that they have the same mode (master or slave).

The commands defined on [page 3](#) through [page 6](#) assume the following:

- The ACL connection handler is 0xB.
- The PCM clock rate is 2048 kHz.
- The CYW4330 is the PCM master.

5.1 Enable_WBS

OCF 0x1

This command is used to enable or disable the wideband speech application and to set the eSCO packet type. Code elements in the prototype that are important for setting up wideband speech are rendered in bold.

Prototype:

```
Enable_WBS          Enable_WBS: 0x1
[7E FC 03]: 01 02 00
opcode = 0xFC7E (64638, "Enable_WBS")
Enable_WBS = 0x1 (1)
UUID_WBS = 2
```

Command Parameters:

Enable_WBS

Type: Boolean
Values: 0 = Disable wideband speech.
1 = Enable wideband speech.
Size: 1 byte

UUID_WBS

Type: int16
2 = EV3/2EV3
Size: 2 bytes

Return Parameters:

Status

Type: uint8
Error codes are defined in the Bluetooth Core Specification (see [Reference \[2\] on page 4](#)).

5.2 Write_I2SPCM_Interface_Param

OCF 0x06D

This command is used to configure the I²S/PCM interface. Code elements in the prototype that are important for setting up wideband speech are rendered in bold.

Prototype:

```
Write_I2SPCM_Interface_Param I2S_Enable: Disable
[6D FC 04]: 00 01 01 04
opcode = 0xFC6D (64621, "Write_I2SPCM_Interface_Param")
I2S_Enable = 0x0 (0, "Disable")
```

```
Is_Master = 0x1 (1, "Master")  
Sample_Rate = 0x1 (1, "16 KHz")  
Clock_Rate = 0x4 (4, "2048 KHz")
```

Command Parameters:**I2S_Enable**

Type: uint8
Purpose: This parameter is used to enable or disable the I²S/PCM interface.
Values: 0x0 = Disable I²S/PCM.
0x1 = Enable I²S/PCM.
Values may not be combined bitwise.
Size: 1 byte

Is_Master

Type: uint8
Purpose: This parameter is used to set the I²S/PCM interface in master or slave mode.
Values: 0x0 = I²S/PCM slave.
0x1 = I²S/PCM master.
Values may not be combined bitwise.
Size: 1 byte

Sample_Rate

Type: uint8
Purpose: This parameter is used to set the sample rate for the I²S/PCM interface.
Values: 0x0 = 8 kHz
0x1 = 16 kHz
0x2 = 4 kHz
Values may not be combined bitwise.
Size: 1 byte

Clock_Rate

Type: uint8
Purpose: This parameter is used to set the clock rate for the I²S/PCM interface.
Values: 0x0 = 128 kHz
0x1 = 256 kHz
0x2 = 512 kHz
0x3 = 1024 kHz
0x4 = 2048 kHz
Values may not be combined bitwise.
Size: 1 byte

Return Parameters:**Status**

Type: uint8
Error codes are defined in the Bluetooth Core Specification (see [Reference \[2\] on page 4](#)).

5.3 Write_SCO_PCM_Int_Param

OFC 0x01C

This command is used to write SCO and PCM parameters. Code elements in the prototype that are important for setting up wideband speech are rendered in bold.

Prototype:

```
Write_SCO_PCM_Int_ParamSCO_Routing: PCM  
[1C FC 05]: 00 04 00 01 01
```

```
opcode = 0xFC1C (64540, "Write_SCO_PCM_Int_Param")
SCO_Routing = 0x0 (0, "PCM")
PCM_Interface_Rate = 0x4 (4, "2048 KBps")
Frame_Type = 0x0 (0, "Short")
Sync_Mode = 0x1 (1, "Master")
Clock_Mode = 0x1 (1, "Master")
```

Command Parameters:**SCO_Routing**

Type: uint8
Purpose: This parameter is used to specify whether the SCO path is through the PCM interface or transport.
Values: 0x0 = PCM
0x1 = Transport
0x2 = Codec
0x3 = I²S
Values may not be combined bitwise.
Size: 1 byte

PCM_Interface_Rate

Type: uint8
Purpose: This parameter is used to set the PCM clock frequency.
Values: 0x0 = 128 KBps
0x1 = 256 KBps
0x2 = 512 KBps
0x3 = 1024 KBps
0x4 = 2048 KBps
Values may not be combined bitwise.
Size: 1 byte

Frame_Type

Type: uint8
Purpose: This parameter is used to set the PCM frame type (short or long).
Values: 0x0 = Short frame
0x1 = Long frame
Values may not be combined bitwise.
Size: 1 byte

Sync_Mode

Type: uint8
Purpose: This parameter is used to set the Bluetooth module as either a master or a slave for the PCM_SYNC signal.
Values: 0x0 = Slave
0x1 = Master
Values may not be combined bitwise.
Size: 1 byte

Clock_Mode

Type: uint8
Purpose: This parameter is used to set the Bluetooth module as either a master or a slave for the PCM_CLK signal.
Values: 0x0 = Slave
0x1 = Master
Values may not be combined bitwise.
Size: 1 byte

Return Parameters:

Status

Type: uint8

Error codes are defined in the Bluetooth Core Specification (see [Reference \[2\] on page 4](#)).

5.4 Setup_Synchronous_Connection

This command is defined in the Bluetooth Core Specification (see [Reference \[2\] on page 4](#)). The code prototype is shown below. Code elements in the prototype that are important for setting up wideband speech are rendered in bold.

```
[28 04 11]: 0B 00 40 1F 00 00 40 1F 00 00 0A 00 63 00 02 80 03
opcode = 0x428 (1064, "Setup_Synchronous_Connection")
Connection_Handle = 0xB (11)
Transmit_Bandwidth = 0x1F40 (8000, bytes per second)
Receive_Bandwidth = 0x1F40 (8000, bytes per second)
Max_Latency = 0xA (10, interval and retransmission window, milliseconds, 0xFFFF=don't care)
Input_Coding = 0x0 (0, "Linear")
Input_Data_Format = 0x1 (1, "2's complement")
Sample_Size = 0x1 (1, "16-bit")
Linear_PCM_Bit_Pos = 0x0 (0)
Air_Coding_Format = 0x3 (3, "Transparent data")
Retransmission_Effort = 0x2 (2, "At least one retransmission, optimize for link quality")
eSCO_Packet_Type = 0x380 (896, "no 3-EV3 | no 2-EV5 | no 3-EV5")
```

6 Switching Between Narrowband and Wideband Speech

In most cases narrowband speech and wideband speech applications share the same transport interface, but have different interface settings. We next describe what should be done on the host side when we switch the application mode from one to another.

6.1 Switching from Narrowband to Wideband Speech

To switch the audio application from Narrowband to Wideband Speech mode, use the commands defined in [Enabling Wideband Speech on the CYW4330 on page 2](#).

6.2 Switching from Wideband to Narrowband Speech

To switch the audio application from Wideband to Narrowband Speech mode, the host stack must first disable Wideband Speech mode using the `Enable_WBS` HCI command, then change the PCM SYNC clock rate to 8 kHz using the `Write_I2SPCM_Interface_Param` HCI command. The examples below assume that the CYW4330 is the PCM master. The code elements that are important to setting up narrowband speech are rendered in bold.

1. Enable Wideband Speech mode:

```
Enable_WBSEnable_WBS: 0x0

[7E FC 03]: 01 02 00

opcode = 0xFC7E (64638, "Enable_WBS")

Enable_WBS = 0x0 (0)

UUID_WBS = 2
```

2. Change the PCM SYNC clock rate to 8 kHz:

```
Write_I2SPCM_Interface_ParamI2S_Enable: Disable
[6D FC 04]: 00 01 01 04
opcode = 0xFC6D (64621, "Write_I2SPCM_Interface_Param")
I2S_Enable = 0x0 (0, "Disable")
Is_Master = 0x1 (1, "Master")
Sample_Rate = 0x0 (0, "8 KHz")
Clock_Rate = 0x4 (4, "2048 KHz")
```

Caution! The host stack must send the HCI command to disable wideband speech when the wideband speech connection is disrupted. The host stack is responsible for keeping track of the audio application mode (narrowband or wideband) for each SCO/eSCO connection.

7 References

The references in this section may be used in conjunction with this document.

Document (or Item) Name	Number	Source
Cypress Items		
CYW4330 Single Chip IEEE 802.11™ a/b/g/n MAC/Baseband/Radio with Integrated Bluetooth® 4.0 + HS and FM Transceiver Data Sheet	4330-DS3xx-R	Cypress Developer Community
Other Items		
Bluetooth Core Specification	v4.0	www.bluetooth.org

Document History Page

Document Title: AN214963 - Setting Up Host Hardware and Software to Support Wideband Speech				
Document Number: 002-14963				
Rev.	ECN No.	Orig. of Change	Submission Date	Description of Change
**	–	–	06/10/2011	4330-AN600-R Initial release
*A	–	–	10/10/2011	4330-AN601-R Updated: <ul style="list-style-type: none"> • PCM CLK setting on page 2. • Note on page 2. • Enable_WBS on page 3. • Write_I2SPCM_Interface_Param on page 3. • Switching from Wideband to Narrowband Speech on page 6.
*B	5473356	UTSV	10/13/2016	Updated in Cypress template Added Cypress part numbering scheme
*C	5879993	AESATMP9	09/11/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturers' representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Video](#) | [Blogs](#) | [Training](#) | [Components](#)

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2011-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1s) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.