THIS SPEC IS OBSOLETE

Spec No: 001-56835

Spec Title: INTERFACING WEST BRIDGE(R) ASTORIA
TO WIRELESSUSB(TM) LP - AN56835

Sunset Owner: Gayathri Vasudevan (GAYA)

Replaced by: NONE

AN56835

Interfacing West Bridge® Astoria to WirelessUSB™ LP

Author: Rama Sai Krishna Vakkantula

Associated Project: Yes
Associated Part Family: CYWB0124AB
Software Version: Astoria SDK 1.2
Related Application Notes: None

If you have a question, or need help with this application note, contact the author at rskv@cypress.com.

West Bridge® Astoria is a USB and mass storage peripheral controller that contains three main ports: processor interface (P-Port), mass storage support (S-Port), and USB interface (U-Port). This application note describes how to connect the West Bridge Astoria with WirelessUSB™ LP and also provides an example of a wireless presentation tool with storage.

Introduction

Most of the low speed USB dongles that are used in wireless mouse and keyboard applications do not have any functionality other than serving as a wireless hub. Since the USB ports can source up to 500 mA, the low speed USB dongles can also be made to perform other functions. One such implementation uses Astoria to connect with a WirelessUSB LP, so that the dongle can be made to act as mass storage controller as well as a wireless hub. This particular implementation reduces the number of components by making Astoria operate as a storage controller and a wireless hub.

West Bridge Astoria

The West Bridge Astoria (CYWB0124AB) device is a peripheral controller supporting high speed USB and mass storage access. This controller provides access from a processor interface and a high speed USB (HS-USB) interface to peripherals including SD, MMC/MMC+, SDIO, SLC, and MLC NAND, that are connected to the two storage ports (S1, S2 ports). It supports interleaving accesses between the processor interface, HS-USB, and the peripherals. This enables an external processor and an external USB host to transfer data to each other and to the mass storage peripherals simultaneously.

WirelessUSB LP

Cypress makes 2.4 GHz ISM band radios with data rates of 256 kbps/1 Mbps. Cypress radio variants are CYRF69103, CYRF69213, and CYRF6936. Among these, CYRF6936 is the WirelessUSB LP radio, which connects with an external microcontroller through a SPI interface. WirelessUSB LP is a half duplex wireless transceiver.

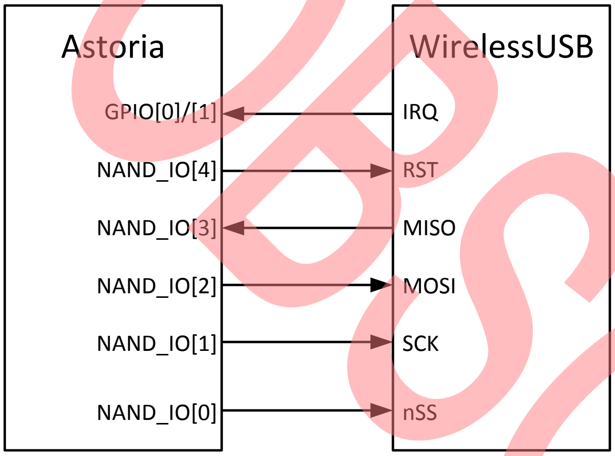
The WirelessUSB LP device is the SPI slave and the microcontroller is the SPI master. All control and data transactions occur through this 4-wire SPI interface. In addition to the SPI interface, there are two lines (IRQ, RST) that need to be connected with the microcontroller for proper communication. IRQ gives the radio interrupt line and RST is used during POR. MOSI and IRQ can be multiplexed to reduce the number of I/Os needed for interfacing.

WirelessUSB LP has an internal power amplifier which can deliver power up to 4 dBm. It supports eight power modes and can operate at any one of the 78 1 MHz bands in the range between 2,400 MHz to 2,477 MHz. A particular channel can be selected using firmware. The device can be made to send or receive in this channel by firmware (SPI commands). An interrupt is triggered whenever a packet is received or transmitted. There are other situations in which the interrupt occurs, but discussions on those are beyond the scope of this application note. WirelessUSB LP supports auto acknowledgements, which means that the receiver sends an ACK packet containing the CRC of the received packet to ensure guaranteed transfer.

Hardware Interface

Figure 1 shows all the pins involved in the hardware interface. Both Astoria and WirelessUSB LP operate at 3.3 V. Hence, the interface requires no level translation. Since all the I/Os are CMOS push-pull, the I/Os can be used for other purposes by pulling up the SS. This particular interface can be expanded too if necessary.

Figure 1. Hardware Interface



Wireless USB requires power on reset; this can be given through the 0.47 µF capacitor between the reset line and VCC. This can also be implemented using the GPIO pin of Astoria, which drives a reset pulse during power on. In the above implementation, NAND_IO[4] is used to drive reset.

Astoria firmware can either poll the interrupt line of the WirelessUSB LP or the IRQ can trigger one of the Astoria interrupts if routed to GPIO[0] or GPIO[1] as only these two lines can trigger an interrupt. Astoria does not have dedicated hardware for SPI. Hence, a firmware implementation of a SPI master is done in Astoria by bit banging the GPIO pins with NAND_IO[0] as nSS, NAND_IO[1] as SCK, NAND_IO[2] as MOSI, and NAND_IO[0] as MISO. Test results have shown that the data rate can reach 440 Kbps. For HID devices, this rate is more than sufficient. This particular implementation is done with the NAND_IOs as the communication lines for SPI; if necessary, an implementation of SPI bit banging the SD_IOs, can also be done.

SPI Bit Banging

An example of the firmware implementation of SPI Master follows.

Two firmware modules are designed (one to receive a byte of data and another to send a byte of data). There are four single bit variables (MISO, MOSI, SS, and SCK) that correspond to the SPI lines and drive corresponding I/Os in the implementation. There are two bytes to buffer data, for example, READ_BUFFER and WRITE_BUFFER. Consider a temporary variable TEMP initialized to 10000000b.

SPI Byte Write

```
Store the data to be sent in WRITE_BUFFER
SS=0
SCK=1
while(TEMP)
MOSI=(WRITE_BUFFER && TEMP)
TEMP>>
SCK^=1;
end while
SS=1
```

SPI Byte Read

```
Empty READ_BUFFER
SS=0
SCK=1
Count=8
while(count)
CONV_BYTE(MISO)
READ_BUFFER|=MISO
READ_BUFFER<<
Count--
SCK^=1
end while
SS=1
```

WirelessUSB LP Driver

A generic Cypress WirelessUSB LP driver is modified and integrated into the Astoria firmware. This driver provides high level API commands to control and transfer data to WirelessUSB LP.

Some important API commands are:

- RadioStartReceive
- RadioGetChannel
- RadioSetChannel
- RadioStartTransmit

This radio driver accesses SPI bit bang calls whenever these APIs are called. For more details on WirelessUSB LP driver APIs, refer to CY4672 RDK documentation.

Application Example

Wireless Presentation Tool with Storage

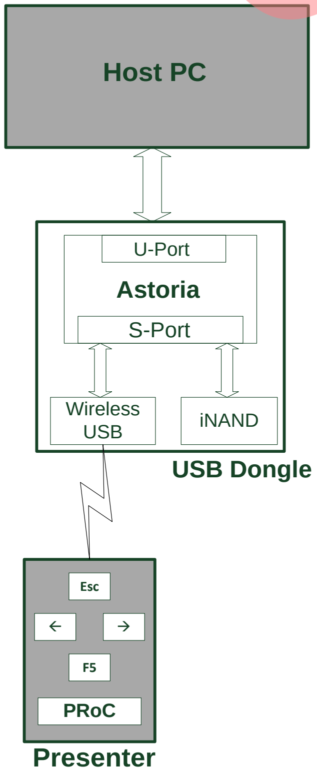
A wireless presentation tool with a storage block level diagram is shown in Figure 2. This tool consists of two subsystems – a USB dongle and a presenter. The discussion of the presenter is beyond the scope of this application note. However, a concise high level description of the presenter is given as follows.

Presenter

The presenter has a PRoC (CYRF69103) and four mechanical push button switches. For more information on PRoC, refer to CYRF69103 datasheet.

These four buttons (with custom functions) are continuously polled by PRoC and their states are transmitted whenever a button is pressed.

Figure 2. Wireless Presentation Tool with Storage



The USB dongle consists of the West Bridge Astoria device (CYWB0226) which connects to an iNAND storage device and to the wireless USB LP (CYRF6936). The dongle receives commands wirelessly from the presenter, which has a PROC (CYRF69103) transmitting the key presses to the dongle. Wireless USB Agile HID Protocol version 2.2 is implemented to establish wireless communication between the dongle and the presenter. The dongle enumerates as a composite device with a mass storage interface and an HID interface. Firmware for Astoria minimizes the average power consumption of the dongle to 85 mA during normal operation and ~2 mA during PC suspend mode. This tool can also wake up the PC from standby remotely.

For the USB specifications, the attached USB devices should not draw more than 2.5 mA on average when the PC is suspended. This essentially requires the firmware to make sure that power dissipation is tightly managed. A discussion of power management when the PC is suspended and how it is implemented in this application follows.

Power Considerations

The power consumed by the dongle at any point is the sum of power consumed by Astoria, WirelessUSB LP, and storage (iNAND, SD, MMC, raw NAND). Therefore, it is mandatory to keep all or most of the devices asleep while in standby. This prevents the PC from being woken up by the remote device (the presenter in this case, but can be extended to include mice or keyboards). A work around for this issue is implemented and discussed in detail in subsequent paragraphs.

Note Powering the device through USB requires 5 V to 3.3 V converters. An LDO, though inexpensive, dissipates a lot of power and leads to inefficient designs. It is better to use a switching regulator to meet this requirement.

Power Management in Firmware

When the PC suspends itself, it sends an interrupt to all the attached USB peripherals. When the interrupt is received, it is handled in Astoria, which forces the Wireless USB LP to sleep; this means that no more than 0.8 µA of power is consumed. The firmware forces the mass storage device to sleep continuously whenever there is no data transfer. This implementation still does not meet the specifications since Astoria is in normal operation while periodically waking up the WirelessUSB LP to handle remote wakeup packets. The solution is to make Astoria suspend on its own, but this prevents Astoria from waking up the WirelessUSB LP making remote wakeup impossible. A workaround can be made in which an RC circuit is connected to the GPIO[1] which wakes up the Astoria periodically. Astoria wakes up the WirelessUSB LP and allows it to receive packets. After 25 ms, if no packet is received, Astoria suspends WirelessUSB LP and goes to the suspend mode. When a packet is received, the PC resumes and all the devices are enabled to operate in normal mode. R and C values are designed so that it wakes up the Astoria device once in a second.

Summary

West Bridge Astoria is a USB and mass storage peripheral controller that contains three main ports: processor interface (P-Port), mass storage support (S-Port), and USB interface (U-Port). This application note describes how to connect the West Bridge Astoria with WirelessUSB LP and also provides an interface example of a wireless presentation tool with storage.

Additional Resources

Firmware hex files for presenter and USB dongle are attached to this application note. In case of any difficulty in interfacing the West Bridge to the WirelessUSB LP, contact Cypress for technical support. For more information regarding West Bridge and WirelessUSB LP, visit the following web pages:

- West Bridge Resources page
- WirelessUSB LP Resources page

About the Author

Name:	Rama Sai Krishna Vakkantula
Title:	Applications Engineer Sr
Background:	Rama Sai Krishna holds an M.Tech in Systems and Control Engg. from IIT Bombay. He is currently working on Cypress USB peripherals.
Contact:	rskv@cypress.com

Document History

Document Title: Interfacing West Bridge® Astoria to WirelessUSB™ LP - AN56835

Document Number: 001-56835

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2781656	RSKV / NZAA	10/08/09	New Application note.
*A	3127501	RSKV / NZAA	01/04/2011	Added Additional Resources.
*B	3790237	RSKV	11/07/2012	Updated Abstract. Updated Summary. Updated in new template.
*C	4824465	GAYA	07/06/2015	Obsolete document.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
Optical Navigation Sensors	cypress.com/go/ons
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 5

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

xx and xx are registered trademarks of Cypress Semiconductor Corp. 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
Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.