

Please note that Cypress is an Infineon Technologies Company.

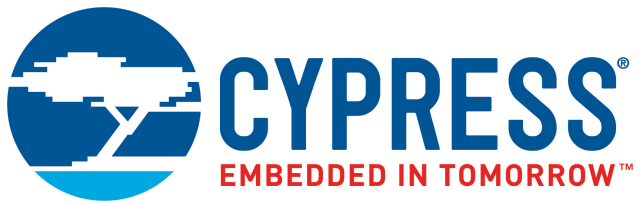
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



THIS SPEC IS OBSOLETE

Spec No: 001-64465

Spec Title: AN64465 - WEST BRIDGE(R) INTEGRATION TO ANDROID
ON OMAP ZOOM II MDP: RNDIS, CDC-ECM, AND MASS
STORAGE FUNCTIONS

Replaced by: None

West Bridge® Integration to Android on OMAP Zoom II MDP: RNDIS, CDC-ECM, and Mass Storage Functions

Author: Dhanraj Rajput

Associated Project: Yes

Associated Part Family: CYWB022X

Software Version: Astoria SDK, Android Kernel

Related Application Notes: None

To get the latest version of this application note, or the associated project file, please visit <http://www.cypress.com/go/AN64465>.

AN64465 provides guidelines to implement USB composite devices. It includes a reference implementation of the USB composite device with Remote Network Driver Interface Specification (RNDIS), Communication Device Class - Ethernet Control Model (CDC-ECM), and mass storage functions. It also discusses the known issues with the Windows operating system and USB composite device implementation, and provides workarounds.

Contents

Introduction	1
Reference Implementation	1
Ethernet over USB	1
Setup	2
West Bridge Astoria.....	2
Additional Resources	2
Example Implementation.....	2
Windows OS: Issues and Workarounds	2
Test Procedure.....	3
Linux (Ubuntu 9.10)	3
Windows OS.....	5
MAC OS	5
Summary.....	5
Appendix A.....	6
Document History.....	13
Worldwide Sales and Design Support	14

Introduction

This document assumes that reader is aware of the basics of USB communication. West Bridge® Astoria can be integrated with various embedded operating systems (OS) and processors. For more details, see the documentation available at <http://www.cypress.com/products/astoria>.

Reference Implementation

The reference implementation described is a composite device consisting of two functions:

- Ethernet over USB
- Mass storage functionality

Ethernet over USB

Ethernet over USB functionality uses the USB communications device class (USB CDC), which is a composite universal serial bus device class. It provides a single device class, but there may be more than one interface implemented such as a custom control interface, data interface, audio, mass storage or computer networking related interfaces. CDC-ECM is commonly supported protocol by Linux and MAC OS. Microsoft Windows versions prior to Windows Vista do not support the networking parts of the USB CDC (CDC-ECM). It supports the Microsoft RNDIS, a serialized version of the Microsoft NDIS.

Setup

A specific setup is used here for test purpose. However, the West Bridge Astoria can be used with any other processor without any changes to firmware and SDK APIs. See the section [Additional Resources](#), if you plan to use a different setup or OS.

The test setup used for this implementation consists of Cypress's West Bridge Astoria Development Kit (DVK) integrated with TI's OMAP34X processor running Android OS on it (OMAP Zoom II MDP). The RNDIS/ECM gadget driver source code available in Android Kernel 2.6.29 is modified to work with West Bridge Astoria Software Development Kit (SDK). The modified RNDIS/ECM gadget driver source code is provided along with this application note (WB_RNDIS_CDC-ECM.zip). The West Bridge SDK source code is not included, but is available as part of kernel release 2.6.36. For more details on the latest SDK, contact the [cypress.com support center](http://www.cypress.com/support-center). More details on OMAP34x and Zoom II MDP are available at http://omapzoom.org/wiki/Main_Page.

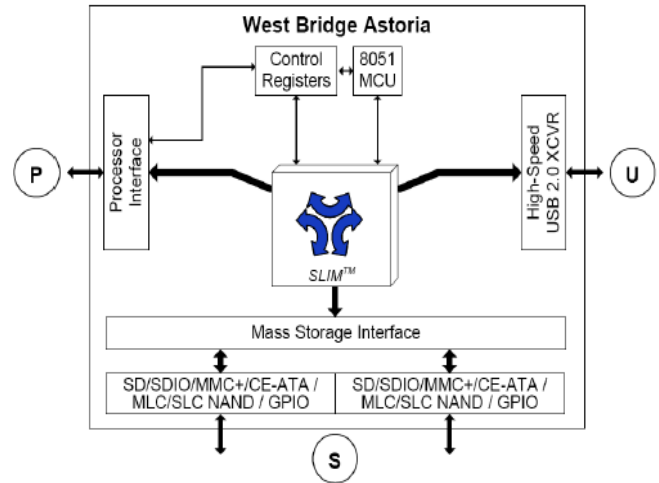
Figure 1. West Bridge Astoria DVK Connected to Zoom II MDP



West Bridge Astoria

Astoria is a flexible bridge device that uses the robust SLIM™ architecture, which allows simultaneous and independent data transfer between the processor and USB, USB and mass storage/SDIO, and processor and mass storage/SDIO. For a full description of Astoria's features, refer to the West Bridge Astoria data sheet.

Figure 2. West Bridge Astoria Block Diagram



The P-Port (processor interface) can operate in one of the multiple selectable interface types such as CRAM, SRAM, address-data multiplexed RAM, SPI, or PNAND. The U-Port is a Hi-Speed (HS) USB peripheral that is capable of independent U-to-P and U-to-S data transfers with optimal performance. The S-Port can be configured to simultaneously interface with multiple mass storage devices such as 8- or 16-bit SLC NAND, managed NAND, SD/MMC/SDIO, and CE-ATA devices.

Additional Resources

- [Integrating West Bridge® Astoria™ with Android](#)
- [Integrating West Bridge® Astoria™ with Symbian](#)
- [Integrating West Bridge® Astoria™ with WinCE 6.0](#)
- [West Bridge® Astoria™ Datasheet](#)
- [West Bridge® Antioch™ Datasheet](#)

Example Implementation

The example implementation has two configurations with three interfaces in each configuration. Configuration #1 comes up as CDC-ECM and Mass Storage Device (MSD) functions; Configuration #2 comes up as RNDIS and MSD functions. Configuration #1 is targeted for Linux and Mac OS and configuration #2 for Windows. The Appendix A lists the USB descriptors used for this.

Windows OS: Issues and Workarounds

The descriptors given in the Appendix A section work well with the Linux and MAC host operating systems, but Windows has some known issues. There are workarounds and hot fixes available from Microsoft for these issues.

- A multifunction, composite USB device that includes a network interface does not work in Microsoft Windows XP with Service Pack 2. A hot fix is available at <http://support.microsoft.com/kb/901122>.
- Composite USB devices whose interfaces are not sequentially numbered do not work in Windows XP. This issue is also observed in Linux host system. The solution is to keep interfaces in a sequence. Microsoft has a hot fix for this and it is available at <http://support.microsoft.com/kb/814560>.
- A composite device with multiple configurations does not work with Windows. For this issue, the composite device driver USBCCGP.SYS is responsible for handling composite device at first stage. It supports multiple device configurations, with a few caveats:
 - USBCCGP does not load on a multi-configuration device by default because the hub driver does not create a 'USB\COMPOSITE' PNP ID for a composite device if it has multiple configurations. However, you can write your own INF that matches a device-specific PNP ID to get USBCCGP to load as your device function driver.
 - Drivers that are clients of USBCCGP cannot change the device configuration value.
 - To select a configuration other than index 0, set registry settings as specified in <http://msdn.microsoft.com/en-us/library/ff538059%28VS.85%29.aspx>. During enumeration, USBCCGP first attempts to select the configuration whose descriptor is found at the specified 'original' index. If the attempt fails, normally due to the configuration requiring more than 100 mA while the upstream hub of the device has only bus power, then USBCCGP attempts to select the configuration found at the specified 'alternate' index instead.

The *composite_rndis_mConf.inf* file, accompanying this application note, handles this issue.

Good practice is to have Interface Association Descriptors (IAD) for each function with interfaces numbered in a sequence.

Test Procedure

The driver build steps are not included because it varies depending on the platform and compiler tools used.

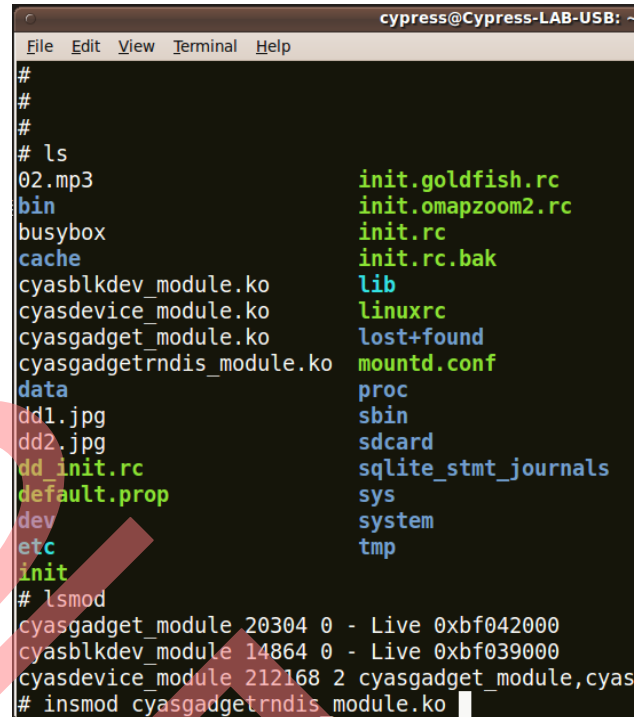
Ethernet functionality can be tested using simple ping request or TFTP file transfer.

The PC host configuration procedure varies with the OS. This section describes the test procedure for Linux, Windows, and Mac OS.

Linux (Ubuntu 9.10)

1. **Zoom II MDP:** Start the MDP and make sure that the required driver modules are initialized properly. The drivers can be part of build-in kernel or loadable module. This test uses loadable modules.

Figure 3. Android Terminal View - Required Modules and Root Directory Listing



```

cypress@Cypress-LAB-USB: ~
#
#
# ls
02.mp3                               init.goldfish.rc
bin                                   init.omapzoom2.rc
busybox                             init.rc
cache                               init.rc.bak
cyasblkdev_module.ko                lib
cyasdevice_module.ko               linuxrc
cyasgadget_module.ko               lost+found
cyasgadgetrndis_module.ko          mountd.conf
data                                proc
dd1.jpg                             sbin
dd2.jpg                             sdcard
dd_init.rc                          sqlite_stmt_journals
default.prop                        sys
dev                                 system
etc                                 tmp
init
# lsmod
cyasgadget_module 20304 0 - Live 0xbf042000
cyasblkdev_module 14864 0 - Live 0xbf039000
cyasdevice_module 212168 2 cyasgadget_module,cyas
# insmod cyasgadgetrndis module.ko
  
```


- Host PC:** Connect the West Bridge Astoria DVK to host PC via USB cable. You can confirm the enumeration by checking, log messages on host PC terminal, or list of USB devices. Host selects config#1, which is CDC-ECM and MSD device and mounts the storage disk, if connected.

Figure 4. Host PC Messages Log on USB Connect

```
[ 728.244032] usb 1-3: new high speed USB device using ehci_hcd and address 4
[ 728.958677] usb 1-3: configuration #1 chosen from 2 choices
[ 729.107285] scsi5 : SCSI emulation for USB Mass Storage devices
[ 729.188456] usb-storage: device found at 4
[ 729.188459] usb-storage: waiting for device to settle before scanning
[ 729.492419] usb0: register 'cdc_ether' at usb-0000:00:1d:7-3, CDC Ethernet Device, c2:65:44:15:eb:6f
[ 729.492451] usbcore: registered new interface driver cdc_ether
[ 734.184215] usb-storage: device scan complete
[ 734.252014] scsi 5:0:0:0: Direct-Access    Cypress Astoria SD Card 0001 PQ: 0 ANSI: 0 CCS
[ 734.252743] sd 5:0:0:0: Attached scsi generic sg type 0
[ 734.259048] sd 5:0:0:0: [sdd] 60800 512-byte logical blocks: (31.1 MB/29.6 MiB)
[ 734.259674] sd 5:0:0:0: [sdd] Write Protect is off
[ 734.259679] sd 5:0:0:0: [sdd] Mode Sense: 03 00 00 00
[ 734.259683] sd 5:0:0:0: [sdd] Assuming drive cache: write through
[ 734.262576] sd 5:0:0:0: [sdd] Assuming drive cache: write through
[ 734.262903] sdd:
[ 734.274665] sd 5:0:0:0: [sdd] Assuming drive cache: write through
[ 734.274671] sd 5:0:0:0: [sdd] Attached SCSI removable disk
cypress@Cypress-LAB-USB:~$
```

Figure 5. USB Device List

```
cypress@Cypress-LAB-USB:~$ lsusb
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 002: ID 0403:6010 Future Technology Devices International, Ltd FT232RL Dual USB-UART/FIFO IC
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 004: ID 04b4:a4a2 Cypress Semiconductor Corp. West Bridge-USB Ethernet/RNDIS Gadget
Bus 001 Device 003: ID 03f0:0125 Hewlett-Packard
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
cypress@Cypress-LAB-USB:~$
```

- Zoom II MDP:** Configure the USB net device using the following command (see Figure 6).

ifconfig usb1 111.111.11.11 netmask 255.255.255.0 up

Figure 6. Network Device List before Configuration and Command for Configuration

```
# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

#
#
# ifconfig usb1 111.111.11.11 netmask 255.255.255.0 up
```

Figure 7. Network Device List after Configuration

```
# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

usb1     Link encap:Ethernet  HWaddr 2A:8D:6A:1C:D9:65
          inet addr:111.111.11.11  Bcast:111.111.11.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:732 (732.0 B)  TX bytes:0 (0.0 B)
```

- Host PC:** Configure USB network device in the host PC.

ifconfig usb1 111.111.11.12 netmask 255.255.255.0 up

Figure 8. Network Device List before Configuration

```
cypress@Cypress-LAB-USB:~$ ifconfig
eth1     Link encap:Ethernet  HWaddr 00:19:d1:3e:5b:35
          inet addr:172.23.143.92  Bcast:172.23.143.255  Mask:255.255.255.0
          inet6 addr: fe80::219:d1ff:fe3e:5b35/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9018 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2624 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3959195 (3.9 MB)  TX bytes:432670 (432.6 KB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:240 (240.0 B)  TX bytes:240 (240.0 B)

usb0     Link encap:Ethernet  HWaddr c2:65:44:15:eb:6f
          inet6 addr: fe80::c065:44ff:fe15:eb6f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:1458 (1.4 KB)
```

Figure 9. Network Device List after Configuration and Command for Configuration

```
cypress@Cypress-LAB-USB:~$ sudo ifconfig usb0 111.111.11.12 netmask 255.255.255.0 up
(sudo) password for cypress:
cypress@Cypress-LAB-USB:~$ ifconfig
eth1     Link encap:Ethernet  HWaddr 00:19:d1:3e:5b:35
          inet addr:172.23.143.92  Bcast:172.23.143.255  Mask:255.255.255.0
          inet6 addr: fe80::219:d1ff:fe3e:5b35/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9106 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2625 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3966704 (3.9 MB)  TX bytes:432736 (432.7 KB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:240 (240.0 B)  TX bytes:240 (240.0 B)

usb0     Link encap:Ethernet  HWaddr c2:65:44:15:eb:6f
          inet addr:111.111.11.12  Bcast:111.111.11.255  Mask:255.255.255.0
          inet6 addr: fe80::c065:44ff:fe15:eb6f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4263 (4.2 KB)
```

- Host PC:** When the device is configured, ping the gadget device and transfer the data over TFTP. For TFTP, install a TFTP server on the host PC. TFTP installation guidelines are available at this link: http://www.wiki.kslemb.com/doku.php/uboot/tftp_setup
- Host PC:** Start TFTP server. Restart, if it fails to start.

Figure 10. TFTP Host Start Command

```
cypress@Cypress-LAB-USB:/tftpboot$ sudo /etc/init.d/xinetd start
* Starting internet superserver xinetd [fail]
cypress@Cypress-LAB-USB:/tftpboot$ sudo /etc/init.d/xinetd restart
* Stopping internet superserver xinetd [ OK ]
* Starting internet superserver xinetd [ OK ]
cypress@Cypress-LAB-USB:/tftpboot$ ls
dd.pdf dd.txt errlog hello.txt hey3.txt
cypress@Cypress-LAB-USB:/tftpboot$
```

7. **Zoom II MDP:** Transfer files by issuing TFTP commands on Zoom II terminal, see Figure 10. Make sure that the file to be received exists in tftpboot directory.

Figure 11. TFTP Client Commands to get File from Host

```
# ls
02.mp3
bin
busybox
cache
cyasblkdev_module.ko
cyasdevice_module.ko
cyasgadget_module.ko
cyasgadgetrndis_module.ko
data
dd1.jpg
dd2.jpg
dd_init.rc
default.prop
dev
etc
init
#
#
#
# tftp -g -l hello.txt -r hello.txt 111.111.11.12
```

Windows OS

Install TFTP server on the host PC.

1. Follow steps 1 and 3 described for Zoom II MDP; connect a USB cable between the host PC and the WB Astoria DVK.
2. A new device shows up in device manager. Initially, it does not come up as a composite device (Windows known issue #3). During driver installation or update, point to the attached *composite_rndis_mConf.inf*; this should make it come up as a composite device.
3. Use *linux.inf* to install driver for RNDIS function. Driver for mass storage function is installed automatically.

4. Go to Windows network connection and manually assign the IP address and net mask.
5. Start the TFTP server. Follow step 7 for Zoom II MDP given in the Linux Host section for file transfer.

MAC OS

Install TFTP server on the host PC.

1. Follow steps 1 and 3 described for Zoom II MDP; connect a USB cable between the host PC and the WB Astoria DVK.
2. When the mass storage and network device is recognized. Assign manual IP address and net-mask to the new network device.
3. Start the TFTP server. Follow step 7 for Zoom II MDP given in the Linux Host section for file transfer.

Summary

West Bridge Astoria can be effectively used as composite device with “Mass Storage” and “Ethernet over USB” functions when connected to the processor, over P-Port, running Android (Kernel 2.6.29). The feature set provided by Astoria makes it a perfect fit for the OMAP34x processor to enable new high speed USB peripheral connectivity and storage control options to latest mass storage devices. It also completely off-loads the processor from the USB mass storage access, which results in a higher performance and more efficient system design.

About the Author

Name: Dhanraj Rajput.
 Title: Applications Group Lead

Appendix A

Table 1. USB Device Descriptors

Device Descriptor	Value	Description
bLength	18	
bDescriptorType	1	
bcdUSB	2	
bDeviceClass	239	Miscellaneous device
bDeviceSubClass	2	?
bDeviceProtocol	1	Interface association
bMaxPacketSize0	64	
idVendor	0x04b4	Cypress Semiconductor Corp.
idProduct	0xa4a2	West Bridge-USB Ethernet/RNDIS gadget
bcdDevice	3.99	
iManufacturer	1	
iProduct	2	
iSerial	0	
bNumConfigurations	2	
Configuration Descriptor:		
bLength	9	
bDescriptorType	2	
wTotalLength	106	
bNumInterfaces	3	
bConfigurationValue	2	
iConfiguration	0	
bmAttributes	0xc0	
Self Powered		
MaxPower	2 mA	
Interface Association:		
bLength	8	
bDescriptorType	11	
bFirstInterface	0	
bInterfaceCount	2	
bFunctionClass	2	Communications
bFunctionSubClass	0	
bFunctionProtocol	0	
iFunction	0	
Interface Descriptor:		
bLength	9	
bDescriptorType	4	

Device Descriptor	Value	Description
bInterfaceNumber	0	
bAlternateSetting	0	
bNumEndpoints	1	
bInterfaceClass	2	Communications
bInterfaceSubClass	2	Abstract (modem)
bInterfaceProtocol	255	Vendor specific (MSFT RNDIS?)
iInterface	3	
CDC Header:		
bcdCDC	1.1	
CDC Call Management:		
bmCapabilities	0x00	
bDataInterface	1	
CDC ACM:		
bmCapabilities	0x00	
CDC Union:		
bMasterInterface	0	
bSlaveInterface	1	
Endpoint Descriptor:		
bLength	7	
bDescriptorType	5	
bEndpointAddress	0x83	EP 3 IN
bmAttributes	3	
Transfer Type	Interrupt	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0008	1x 8 bytes
bInterval	9	
Interface Descriptor:		
bLength	9	
bDescriptorType	4	
bInterfaceNumber	1	
bAlternateSetting	0	
bNumEndpoints	2	
bInterfaceClass	10	CDC data
bInterfaceSubClass	0	Unused
bInterfaceProtocol	0	
iInterface	4	
Endpoint Descriptor:		
bLength	7	

Device Descriptor	Value	Description
bDescriptorType	5	
bEndpointAddress	0x85 EP 5 IN	
bmAttributes	2	
Transfer Type	Bulk	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0200	1x 512 bytes
bInterval	0	
Endpoint Descriptor:		
bLength	7	
bDescriptorType	5	
bEndpointAddress	0x07	EP 7 OUT
bmAttributes	2	
Transfer Type	Bulk	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0200	1x 512 bytes
bInterval	0	
Interface Association:		
bLength	8	
bDescriptorType	11	
bFirstInterface	2	
bInterfaceCount	1	
bFunctionClass	8	Mass storage
bFunctionSubClass	6	SCSI
bFunctionProtocol	80	Bulk (Zip)
iFunction	0	
Interface Descriptor:		
bLength	9	
bDescriptorType	4	
bInterfaceNumber	2	
bAlternateSetting	0	
bNumEndpoints	2	
bInterfaceClass	8	Mass storage
bInterfaceSubClass	6	SCSI
bInterfaceProtocol	80	Bulk (Zip)
iInterface	0	
Endpoint Descriptor:		
bLength	7	

Device Descriptor	Value	Description
bDescriptorType	5	
bEndpointAddress	0x02	EP 2 OUT
bmAttributes	2	
Transfer Type	Bulk	
Synch Type	None	
Usage Type Data		
wMaxPacketSize	0x0200	1x 512 bytes
bInterval	0	
Endpoint Descriptor:		
bLength	7	
bDescriptorType	5	
bEndpointAddress	0x86	EP 6 IN
bmAttributes	2	
Transfer Type	Bulk	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0200	1x 512 bytes
bInterval	0	
Configuration Descriptor:		
bLength	9	
bDescriptorType	2	
wTotalLength	119	
bNumInterfaces	3	
bConfigurationValue	1	
iConfiguration	0	
bmAttributes	0xc0	
Self Powered		
MaxPower	2mA	
Interface Association:		
bLength	8	
bDescriptorType	11	
bFirstInterface	0	
bInterfaceCount	2	
bFunctionClass	2	Communications
bFunctionSubClass	0	
bFunctionProtocol	0	
iFunction	0	
Interface Descriptor:		
bLength	9	

Device Descriptor	Value	Description
bDescriptorType	4	
bInterfaceNumber	0	
bAlternateSetting	0	
bNumEndpoints	1	
bInterfaceClass	2	Communications
bInterfaceSubClass	6	Ethernet networking
bInterfaceProtocol	0	
iInterface	5	
CDC Header:		
bcdCDC	1.1	
CDC Union:		
bMasterInterface	0	
bSlaveInterface	1	
CDC Ethernet:		
iMacAddress	7 (??)	
bmEthernetStatistics	0x00000000	
wMaxSegmentSize	1514	
wNumberMCFilters	0x0000	
bNumberPowerFilters	0	
Endpoint Descriptor:		
bLength	7	
bDescriptorType	5	
bEndpointAddress	0x83	EP 3 IN
bmAttributes	3	
Transfer Type	Interrupt	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0010	1x 16 bytes
bInterval	9	
Interface Descriptor:		
bLength	9	
bDescriptorType	4	
bInterfaceNumber	1	
bAlternateSetting	0	
bNumEndpoints	0	
bInterfaceClass	10	CDC data
bInterfaceSubClass	0	Unused
bInterfaceProtocol	0	
iInterface	0	

Device Descriptor	Value	Description
Interface Descriptor:		
bLength	9	
bDescriptorType	4	
bInterfaceNumber	1	
bAlternateSetting	1	
bNumEndpoints	2	
bInterfaceClass	10	CDC data
bInterfaceSubClass	0	Unused
bInterfaceProtocol	0	
iInterface	6	
Endpoint Descriptor:		
bLength	7	
bDescriptorType	5	
bEndpointAddress	0x85	EP 5 IN
bmAttributes	2	
Transfer Type	Bulk	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0200	1x 512 bytes
bInterval	0	
Endpoint Descriptor:		
bLength	7	
bDescriptorType	5	
bEndpointAddress	0x07	EP 7 OUT
bmAttributes	2	
Transfer Type	Bulk	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0200	1x 512 bytes
bInterval	0	
Interface Association:		
bLength	8	
bDescriptorType	11	
bFirstInterface	2	
bInterfaceCount	1	
bFunctionClass	8	Mass storage
bFunctionSubClass	6	SCSI
bFunctionProtocol	80	Bulk (Zip)
iFunction	0	

Device Descriptor	Value	Description
Interface Descriptor:		
bLength	9	
bDescriptorType	4	
bInterfaceNumber	2	
bAlternateSetting	0	
bNumEndpoints	2	
bInterfaceClass	8	Mass storage
bInterfaceSubClass	6	SCSI
bInterfaceProtocol	80	Bulk (Zip)
iInterface	0	
Endpoint Descriptor:		
bLength	7	
bDescriptorType	5	
bEndpointAddress	0x02	EP 2 OUT
bmAttributes	2	
Transfer Type	Bulk	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0200	1x 512 bytes
bInterval	0	
Endpoint Descriptor:		
bLength	7	
bDescriptorType	5	
bEndpointAddress	0x86	EP 6 IN
bmAttributes	2	
Transfer Type	Bulk	
Synch Type	None	
Usage Type	Data	
wMaxPacketSize	0x0200	1x 512 bytes
bInterval	0	

Document History

Document Title: AN64465 - West Bridge® Integration to Android on OMAP Zoom II MDP: RNDIS, CDC-ECM, and Mass Storage Functions

Document Number: 001-64465

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3043385	DBIR	09/30/2010	New application note.
*A	3385395	DBIR	09/27/2011	Updated Setup section and Additional Resources section . Updated template according to current Cypress standards.
*B	4536002	DBIR	10/13/2014	Updated to new template. Completing Sunset Review.
*C	5818063	AESATP12	07/19/2017	Updated logo and Copyright.
*D	5962176	RAJV	11/09/2017	Obsolete document. Completing Sunset Review.

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

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](#) | [Videos](#) | [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 Corporation, 2010-2017. 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) (1) 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.