



# Certified Wireless USB Power Management

**Brad Hosler**  
Intel Corporation

# Agenda



- Device Power Management
- Host Power Management
- Device disconnects and loss of Certified Wireless USB channel

# Device Power Management



- Devices and their drivers own device PM
  - Just like the wired case for selective suspend
  - Hosts don't need to manage devices to save host power
- Three ways to save power
  - Conserve during normal operation
    - Shut down radio between MMCs or whatever else makes sense
    - Host is unaware of any device PM stuff
  - Device goes to sleep
    - Extended periods where device won't respond
    - Host is aware and doesn't schedule traffic
  - Device disconnects
    - Host is aware (explicit)

# Device Sleep (1)



## Mechanism so that devices can conserve power for 'longer' period of time, but still stay connected

- Host must be notified that device is sleeping
  - So that host doesn't waste bandwidth trying to talk to device
  - Device can tell host two things
    - I'm going to sleep no matter what (send 'DN\_Sleep(no matter what)' notification)
    - I'm going to sleep unless there's something pending (send 'DN\_Sleep(if no work)' notification)
  - Standard encrypted notification mechanism used
  - Host responds two ways
    - No work pending (Work IE(no work pending))
    - Work pending (Work IE(work pending))
    - Host ignores flow controlled IN endpoints and interrupt IN endpoints when deciding
  - Three retries required
  - If host doesn't respond, device will typically retry and then do Reconnect notification

# Device Sleep (2)



## Device may wakeup occasionally to tell host that it is still around

- How often is device specific – Not more often than every 100ms
  - May be very rarely, or only when the user has done something to the device
- Standard encrypted notification mechanism used
- Device can say three things
  - I'm going to sleep, no matter what (send 'DN\_Sleep(no matter what)' notification)
    - Work/No Work response from host
  - I'm going to sleep unless there's something pending (send 'DN\_Sleep(if no work)' notification)
    - Work/No Work response from host
  - I'm awake (see next slide)

# Device Sleep (3)



- When device wants to wake up
  - If last communication with host was within *TrustTimeout*, then send DN\_Alive
    - Host will resume normal activity
    - Host may do 4-way handshake
  - If last host communication was longer than *TrustTimeout*, then send DN\_Reconnect
    - Host will respond with Connect Ack IE

# Host Responses



<i>Device Notification</i>	<i>Host Response</i>	<i>Resulting Device State (Host View)</i>
Going to sleep	Work	Sleep
Going to sleep	No Work	Sleep
Want to sleep	Work	Awake
Want to sleep	No Work	Sleep
Reconnect	Connect Acknowledge	Awake
DN_Alive	Resume transactions	Awake

# Host Power Management



- Two cases
  - Case 1: WUSB channel (linked MMCs) is maintained
    - Saving power while system is operational
  - Case 2: WUSB channel is interrupted
    - System going to suspend (S3 or lower)
    - Radio is shut down for extended period
- Case 1: USB Channel maintained
  - Devices aren't consuming host power so there's no host need to get devices to conserve power
  - Only host power to conserve is radio power
    - Host specific how that happens. Slow down MMC rate. Turn off radio opportunistically
  - Devices aren't really aware, they may just see less activity

# Case 2: Radio Shutdown



- Reasons include Standby, Shutdown, more aggressive power savings, user disables radio, power loss, ...
- Host tells devices of radio shutdown (channel stop)
  - Channel Stop IE in MMCs specifies CWUSB time when MMCs will stop
    - Host must include in at least three consecutive MMCs
    - Host uses link value of zero for last MMC
  - IE also indicates if host will be checking for remote wakeup
  - If host is checking for remote wakeup
    - Host should wake up at least every 4 seconds and send at least three MMCs (with DNTS)
    - If going back to sleep, the 'Channel Stop IE' will be in these MMCs
  - If host decides not to go to sleep (while 'Channel Stop' IE is in MMC), host just removes the IE from subsequent MMCs
  - Host assumes all devices have gone to 'Sleep' state
  - If 'Channel Stop' IE is present
    - Any notification from devices are allowed

# Remote Wake



- Remote Wake
  - Devices that want to do remote wake listen for MMCs
    - Device should hear MMCs within 4 secs
    - Can wait longer if they want, at some point can decide that host has disappeared
  - Devices use 'Remote Wake' notification to tell host to wake up
    - Notification is encrypted
    - Host response is that 'Channel Stop' IE disappears from MMC
    - Device still needs to send Reconnect notification after MMCs resume
      - Allows host to re-authenticate if necessary

# Host Wakes Up



- Several cases of this
  - Coming up from standby. Booting machine
  - Host behavior (over the air) is the same in all cases
    - SW may try to retain some context (i.e., PDOs may not be deleted), but at link level all devices have to be re-authenticated
    - Devices can tell host is 'awake', not just checking for remote wake, by lack of 'Channel Stop' IEs
- Operation
  - Host starts WUSB channel
    - MMCs, DNTSs, ...

# Host Wakes Up: Device Behavior



- Devices has two choices
  - Connect notification (using unassociated address)
  - Reconnect notification (remember their previous context, avoiding a re-enumeration)
    - Should not be encrypted
  - No mandated time limit for doing the notification
    - But longer than four seconds may mean re-enumeration

# Device View: Host Sleeping



- Device can immediately save power
  - Can go to 'Sleep' state
- Detecting when host wakes up
  - Device may periodically wake up to see if host is alive
    - Run risk of disconnection if longer than 4 secs
  - Or wait until user does something (like hit a key on keyboard)
- If enabled for Remote Wakeup and host is doing remote wake
  - If external event happens and device wants to wake the host, device listens for MMCs and sends Remote Wake notification when appropriate
  - Device still needs to do Reconnect notification after remote wake

# Host View of Device Disconnect



- Explicit
  - Device sends notification to host
    - Uses DN\_Disconnect notification
      - Must try at least three times
    - Host responds with WDEV\_Disconnect IE
    - Device may not wait for final notification response
- Device is unresponsive
  - Host may decide to 'disconnect' device after some period of time of not being responsive
    - Don't disconnect because of inactivity
    - Must not be faster than 4 secs

# Device View: Loss of MMCs



- Many causes
  - Host turned off
  - User walked away with device (or host)
  - Host changed channels
  - Device behavior is the same for all cases
- Could happen in many device states
  - Normal operation: MMC stream just disappears
  - Host sleeping: Host went to sleep and seems never to wake up
  - Device awakening: Device wakes up and can't find host

# Device Behavior: Loss of MMCs



- First step: Detect/decide host gone.
  - How fast this is done depends on device
    - Device that just lost the channel may decide very quickly that host is gone
    - Other cases may take longer
- Second step: Repeat standard startup procedure
  - When this happens is device specific
    - Some devices may do this quite often (e.g., powered printer)
    - Others only do it after user stimulus (e.g., portable devices)



# Developers Conference 2006

Taipei, Taiwan