Design Considerations for Self-Beaconing & Dual-Role Devices

S. Aravindhan
Synopsys, Inc.

Content also provided by: Matthew Myers, Synopsys, Inc.
Agenda

• Self-Beaconing Devices
  • Un-tethered from the host

• Dual-Role Devices
  • Host and device built-in

• Concurrent Connection Devices
  • Tethered to >1 host!

• Questions?
Self-Beaconing Devices
Out in the Wild

- Wired USB devices were slaves to host
  - Provided a service or used a service on the host
  - Wire gave a clear context of connection

- Certified Wireless USB devices share the medium with other WiMedia devices
  - May lose contact with the host
  - May support other protocols using the same radio
WiMedia – MAS and Beacon

- WiMedia Devices make reservation in each superframe (65.54mS)
- A Super frame is divided into 256 Medium Allocation Slots (MAS) of each 256uS.
  - The first 16 MAS are typically used for Beacon
  - Beacon time is used for WiMedia devices to communicate to each other about reservation and other information
  - WiMedia uses Distributed Reservation Protocol (DRP)
  - Unreserved MAS slots can be arbitraged through Prioritized Contention Access protocol (PCA)
  - CWUSB Hosts always make reservation, the CWUSB devices don’t make reservation
  - CWUSB doesn’t use PCA
Beaconing - Three Types of Certified Wireless USB Devices

- Host uses the capabilities of the beaconing devices to get information about its hidden neighbors

- Three types of Devices
  - No Beaconing
    - The device must always be very close to the host, the device and host see the same neighbor
  - Directed Beaconing
    - Not aware of WiMedia Protocols
  - Self Beaconing
    - WiMedia aware device
Survival Tools

- In order to cooperate with other WiMedia devices, Certified Wireless USB devices must beacon

- Directed-Beaconing Device (DBD) is told how to beacon by its host
  - Loses the ability to communicate when host is out of range or turned off
  - Sufficient for most USB-replacement devices

- Self-Beaconing Device (SBD) must take care of itself
Self-Beaconing Devices

- May roam away from host and still provide a useful function
- SBD Capability enables more types of products
  - Dual-Role Device
  - Concurrent Host Connections
  - Multiple PALs in one device
• Module 1:
  • No-Beaconing or Direct Beaconing Device
• Module 1 + 2:
  • Self Beaconing Device
• Module 1 + 2 + 3
  • Dual Role Device
What is an SBD?

- A Certified Wireless USB device that operates in a WiMedia channel with minimal direction from host
  - Manages beacon transmission/reception
  - BP length adjustment
  - Beacon collision detection
  - BP contraction

- Host involvement
  - Private reservation setup for WUSB Channel
  - Reservation movement after a BP Merge
Private WUSB Reservation

- WUSB Channel uses a Private Reservation
  - Provides exclusive access to the medium for the reservation owner (host) and target (devices)
  - Addresses from 0 to 255 allowed
- Host creates this reservation
  - Communicates the reservation to SBD’s via the “backdoor”
  - An SBD reflects this reservation to other WiMedia devices in its beacon
Backdoor DRP Negotiation Example

- SBD’s A, B, C, and D are connected to the host
- Host cannot see WiMedia Device X
- Device A’s Availability takes into account X’s reservations
- Host decides which MAS can be reserved
- Host directly sets the DRP IE used in A’s beacon
Backdoor DRP Negotiation Example

- Dev A notices change in Dev X’s reservations
- Host obtains new Availability and sets new DRP IE
• To free up Private Reservation, Host will send a WRELEASE_CHANNEL_IE and UDA
  • SBD should listen to IE, not UDA
  • The UDA/UDR is for the benefit of other WiMedia devices to use the remainder of the reservation

• Relinquish Request IE
  • If Host made an unsafe reservation, SBD may be the target of Relinquish Request IE
  • SBD transmits RRIE in its Beacon with Host as its target (forwarding the request)
Beacon Period Merging, Power Management

• SBD is responsible for taking part in BP Merge process
  • Either host or device could detect an Alien Beacon, and includes BP Switch IE
  • Other WiMedia devices see BP Switch IE

• After merge
  • SBD may send DN_MASAvailChanged to host
  • Host will deliver new DRP IE to SBD to assert existing reservations in new BP

• Power Management
  • SBD should synchronize WiMedia (Hibernation IE) with WUSB Sleep
Channel Changing

- SBD with multiple PALs communicating on same channel
- If host wants to change channel, SBD may:
  - Put Channel Change IE in beacon and move with host, or
  - Ignore channel change
    - Disconnect from host and stay with other devices
- If WiMedia device wants to change channel, SBD may:
  - Put Channel Change IE in beacon and move with the device and maybe the host, or
  - Ignore channel change
    - Disconnect from devices and stay with host
- Devices with display can ask user which connection to keep
- Devices with multiple PHYs or channel hopping can stay on both channels
Dual-Role Devices (DRD)
Dual-Role Device

• Two Types of Dual Role Devices
  • Static DRD Device
  • Dynamic DRD Device
Static Dual-Role Device

- Static DRD acts as a host or a device at a given time
- Less complex implementation
- User chooses host or device mode

![Diagram showing Host, Device, and Static DRD connections](image)

---transmits MMCs

(OR)

---receives MMCs
Dynamic Dual-Role Devices

- DRD acts as both a host and device simultaneously
- Transmits and receives MMC’s on the same channel

**Combination DRD**
- Printer connected to PC and digital camera

**Point-to-Point DRD**
- Two cellphones/MP3 players connected to each other
- Could be limited function – Allows communication only between same vendors cellphones
The DRD Trick

- Act like a device
  - Follow the thread of MMC’s from a host
- Act like a host
  - Transmit a thread of MMC’s
  - Host functionality can be limited (ala Targeted Peripheral List)
- Security Keys
  - Combo-DRD = 2 GTK’s & 1+N PTK’s
  - P2P-DRD = 1 GTK & 1 PTK
DRD Considerations

• Channel Change
  • DRD’s host may change channels
  • Combo DRD relays WCHANNEL_CHANGE_IE to its devices
  • P2P DRD just moves with the initiator of the channel change
  • With multiples PALs, same issues as SBD

• Disconnect/Sleep
  • DRD’s host may disconnect or stop channel
  • Combo DRD keeps connections to devices alive
    • Digital camera can print even though the PC is off!
  • P2P DRD can disconnect/sleep since there is only 1 link
Power Management For DRD

- As a Device - Need to Follow WUSB Sleep
- As a Host – Need to Poll for Remote Wakeup
- As a WiMedia Device - Need to Follow Hibernation
- DRD Need to Synchronize all the Three Events
Concurrent Devices
Multi-Host Definitions

• Because it is wireless, the user expects to share the device between multiple hosts

• Two Types of Multiple connection Devices
  1. Multiple Connection Contexts Device
  2. Concurrent Device
Multi-Host Definitions

- Device with Multiple Connection Contexts
  - Connected to 1 host at a time, but can switch without going through the association process
  - Need to store multiple connection context and provide user interface to select one of them
  - Application: A printer shared by two PCs, user selects the PC to connect
  - Conceptually similar to a wired USB manual share switch
  - Unlike wired USB, no additional external component; switch function is build in to the device
  - Switch function is easier to build; no Device-class specific SW/HW component

Wired USB Equivalent
Multi-Host Definitions

- Concurrent Device
  - Connected to >1 host at a time, arbitration between hosts handled by the device
  - Usage model: Shared printer between two PCs
  - Conceptually similar to a wired USB intelligent share switch
  - Similar to wired USB auto share switches, implementation is specific to a Device-class
  - Additional complexity specific to CWUSB

Wired USB Equivalent
Concurrent Device

- Device follows 2 or more threads of MMC’s
- Similar to having 2 Certified Wireless USB device PALs
  - Not the same as a compound device
Host Locations

- Device behavior depends on host locations
  - Hosts on same channel and coordinated
    - Beacon
    - Host 1
    - Host 2
    - MMC
    - MMC
    - MMC
    - MMC
    - MMC
  - Hosts on same channel but uncoordinated
    - Beacon
    - Host 1
    - Host 2
    - MMC
    - MMC
    - MMC
    - Beacon
    - MMC
    - MMC
  - Hosts on different channels
    - Beacon
    - Host 1
    - Host 2
    - MMC
    - MMC
    - Beacon
    - MMC
    - MMC
    - Beacon
    - MMC
    - MMC
Coordinated hosts
  - Expect SBD to beacon multiple DRP IE’s

Uncoordinated hosts
  - Expect SBD to beacon in both beacon periods
  - Should eventually result in BP Merge

Hosts on different channels
  - Expect SBD to beacon in both channels
  - Tricky if device has only one PHY (e.g. BP’s may overlap)

Concurrent devices may support one or all three configurations
Concurrent Connection Device Considerations

- Device accessed from multiple hosts
  - Behavior depends on class of device
  - Printer can stall one host while it finishes the job from the other
  - Mass storage may be able to interleave multi-host accesses
- One host asks the device to go to sleep or change PHY channels
  - Device may require all hosts to be on the same channel
- Many challenges to overcome
  - Do what it takes to make each host to think it has an exclusive connection to device
  - The size of Hardware/Software will depend upon how many hosts you want support simultaneously
## Products and Capabilities

<table>
<thead>
<tr>
<th>Products</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Host</td>
</tr>
<tr>
<td>Host</td>
<td>✓</td>
</tr>
<tr>
<td>Self-Beaconing Device</td>
<td>✓</td>
</tr>
<tr>
<td>Directed-Beaconing Device</td>
<td>✓</td>
</tr>
<tr>
<td>Dual-Role Device</td>
<td>✓</td>
</tr>
<tr>
<td>Non-Beaconing Device</td>
<td>✓</td>
</tr>
<tr>
<td>Other PAL Functions (Example: WiNet)</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Conclusion

• A Self-Beaconing Device has 2 personalities
  • As a WiMedia device, obey the MAC spec
  • As a Certified Wireless USB device, need to obey the host

• A Dual-Role Device has 3 personalities
  • WiMedia device, Certified Wireless USB host, and Certified Wireless USB device

• Concurrent Device can be connected to multiple hosts at the same time
  • Device needs to follow multiple threads of MMC’s
  • Biggest challenges are driver-related and obeying more than 1 host
Recommendation

• Since many flavors of the devices can be developed, when you are developing a Certified Wireless USB solution:
  • Choose a scalable architecture
  • Preferably a blend of HW and firmware approach where the firmware provides the customization.

Challenges:
  • Making it easier to integrate in any of your future SoC and application
  • Comparable in power, area, and performance to a HW only approach
Developers Conference 2006
Taipei, Taiwan
Backup Slides
# Addressing

<table>
<thead>
<tr>
<th>Device Type</th>
<th>48-bit MAC Addr</th>
<th>8-bit CWUSB Cluster Addr</th>
<th>8-bit CWUSB Device Addr</th>
<th>16-bit Generated DevAddr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Beaconing Device</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Combo-DRD</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>P2P-DRD</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Concurrent Device</td>
<td>1</td>
<td>2+</td>
<td>2+</td>
<td>1</td>
</tr>
<tr>
<td>Directed Beaconing Device</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Non-Beaconing Device</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
## Security Keys

<table>
<thead>
<tr>
<th></th>
<th>CWUSB PTK’s</th>
<th>CWUSB GTK’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Beaconing Device</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Directed-Beaconing Device</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Beaconing Device</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combo-DRD</td>
<td>1 + $N_{\text{devs}}$</td>
<td>2</td>
</tr>
<tr>
<td>P2P-DRD</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Concurrent Device</td>
<td>$N_{\text{hosts}}$</td>
<td>$N_{\text{hosts}}$</td>
</tr>
<tr>
<td>Host</td>
<td>$N_{\text{devs}}$</td>
<td>1</td>
</tr>
</tbody>
</table>
Class Driver Support

- Printer class driver supports knowledge of beginning/end of job
- A shared printer can block another host if a job is in progress
- What’s the difference compared to a network printer?