MCPC

モバイルコンピューティング推進コンソーシアム
Mobile Computing Promotion Consortium

MCPC TR-013

Applying USB in Car Electronics

Version 1.00(e)

2010-3-11

Mobile Computing Promotion Consortium
Technical Committee

Revision History

Date	Version	Summary of Changes
2010年03月11日	1.00(e)	Base version initial release.

Published and copyrights owned by:

Mobile Computing Promotion Consortium (MCPC)
Shiba Koen Sanada Bldg.
5-12 Shiba-Koen 3-chome, Minato-ku, Tokyo, 105-0011 Japan

TEL：+81-3-5401-1935
FAX: +81-3-5401-1937
EMAIL: office@mcpc-jp.org
WEB SITE: http://www.mcpc-jp.org

Confidentiality:

The MCPC Rules and the MCPC IPR Policy Shall Apply

Disclaimer:

This document is intended to provide standard specifications, recommended specifications, etc. on mobile computing. Mobile Computing Promotion Consortium (hereafter MCPC) shall not be liable for any damages or infringements of patents or other rights of third parties arising out of the use of this document. This document shall not be construed to grant (a) license(s) under any rights held by MCPC or third parties.

How we distinguish binary, decimal and hexadecimal numbers：

- For binary numbers, we append small letter "b" (e.g. 10b))
- For binary numbers, we insert a space after every four bits. (e.g.：1000 0101 0010b)
- For hexadecimal numbers, we append small letter "h" (e.g.：FFFFh and 80h)
- All other numbers shall be considered to be written in decimal

Key Words

- The Japanese interpretation of "may", "should" and "shall" in this document is expressed as follows:
- **することができる してもよい** (may) **推奨または要求に自由な選択肢を示す。**
- **すべきである** (should) **必須ではないが強い推奨を示す。実施の際、必須ではないが考慮すべき。**
- **しなければならない** (shall) **必須要求を示す。接続性、仕様準拠のために必ず実施しなければならない。**

In English, we translate these expressions as follows:

- "May" means that something is recommended or optional at the free discretion of the vendor.
- "Should" means that although something is not essential, it is strongly recommended. When implementing, the vendor shall take this requirement into consideration and determine whether this is essential or not.
- "Shall" means that something is an essential requirement. For connectivity and specification compliance, the feature must be implemented, and is mandatory.

Trademark Notes

- Company names, product names, service names and the like described in this document are generally trademarked or are registered trademarks of their respective owners.
- In general, the usage of ™ marks, © marks, and ® marks has been omitted in this document.
- In general "Inc.," "Corporation," "Co., Ltd.," have been excluded from company names. Japanese corporate names described in this document exclude the **株式会社**. Therefore, all corporate names in this document are described in the abbreviated form ending with "**社**."

Table of Contents

1. Overview

The user-facing electronics in cars is getting ever more sophisticated. The electronics in the dashboard may include a car stereo, a car navigation system, an advanced entertainment system, or some combination of the above. (For simplicity, in the rest of this document we refer to the car electronics as the Car Navi.)

This document provides recommendations for implementing Car Navis USB functions and other products so as to enable a variety of use cases. The use cases given here don't describe all the things users can do, but they give a good feeling for the intended applications.

- UC-01: The user attaches a USB flash drive to the Car Navi. The Car Navi presents a menu showing the kinds of media stored on the device, and the user can play content from the device, transfer media into the Car Navi, backup information from the Car Navi, backup information from the device to the Car Navi, and so forth.
- UC-02: The user attaches a personal media player (PMP) to the Car Navi, and plays media back through the A/V system of the car.
- UC-03: The user buys Car A and learns how to perform various simple tasks with the Car Navi. The user then buys a second car, Car B. When the user moves between the two cars, he or she does not have to re-learn basic concepts of how to connect USB accessories
- UC-04: The user goes to a store or website to buy an accessory for the car. The user has a simple way to judge whether a given accessory will work with the car.
- UC-05: The user wants to wirelessly connect their cell-phone to their Car Navi. The user buys a Bluetooth adapter, and attaches it to the Car Navi, then connects the Car Navi to their cell-phone for voice calls.
- UC-06: The user wants to easily call their contacts by initiating a call directly from the Car Navi user interface. The user connects their cell-phone to the Car Navi to synchronize contact information with the phone, and can select their contacts directly.
- UC-07: The user wants to carry out a voice call using the Car Navi hands-free calling interface. The user connects their cell-phone to the Car Navi, chooses a contact to dial, then uses the Car Navi's hands-free functions to hold their conversation.
- UC-08: The user wants to transfer photos from a Digital Camera to the Car Navi. The user connects the camera to the Car Navi, and can use the on-screen menus to browse the photos, use them for point data information, etc.
- UC-09: The user wants to transfer photos from a memory card to the Car Navi. The user connects a memory card reader to the Navi and can browse the card.
- UC-10: The user connects the PND or cell-phone to the Car Navi. They want to transfer map data or point info around car.
- UC-11: The user wants to make a plan for driving, using the computer in their house. The user creates the plan, and saves it on a USB flash drive. The user connects the USB flash drive to the Car Navi. The Navi reads the information and guides the user.

The specification is organized into two basic parts. Section 2 presents the various use cases in detail and describes the components needed. Because many of the technical requirements overlap, we then present technical details and considerations about implementation details in Sections 3 through 5.

Generally speaking, the software in Car Navis is less complex than the software in traditional personal computers, and it is difficult to support the huge variety of devices that can be connected to a PC. The general characteristics of car applications lead to the following needs.

- Car Navi makers need guidance as to what features to put into the software of their products.
- Product makers need guidance as to how to implement devices that will interwork with Car Navis.
- Users need guidance as to what devices will work with their Car Navi, so they can feel confident that the accessories they purchase will work with their cars.

- Users need a simple, common experience for USB in Car Navis that is similar to what they are familiar with from working with PCs; but this experience must cater to the more limited resources available to Car Navis (as compared to PCs).

This technical report aims to meet those needs by defining a set of common vocabulary, functionality, and implementation requirements.

1.1 Categorizing Products

For purposes of discussion, it is convenient to classify USB-capable products as follows.

- **Accessories** (or **Accessory Devices**) are products that don't have any function when used alone. Examples of accessories include: Keyboard, mouse, game controller, USB audio headset, Wi-Fi adapter, USB flash disks, and so forth. When connected via USB, Accessory Devices are always USB Devices (not USB Hosts).

- **Smart Devices** are products whose primary function is stand-alone. However, Smart Devices can also usefully interact with other products. Examples of Smart Devices include: Cell phones, media players, DTVs, Digital Still Cameras, Camcorders. Smart Devices may connect to another product (especially to a PC) as USB devices; but in certain circumstances they may connect to another product as a USB host.

Car Navis are Smart Devices. They often will play the role of the USB host, in order to connect to Accessories. However, they may also need to play the role of a USB device, in order to connect to PCs or other Smart Devices.

2. Use Cases

This section investigates specific use cases in a form that can be used by product managers to determine what features will be needed in the Car Navi and in the accessory in order to achieve the desired task.

2.1 USB Flash Drive Support

Summary: UC-01: The user attaches a USB flash drive to the Car Navi. The Car Navi presents a menu showing the kinds of media stored on the device, and the user can play content from the device, transfer media into the Car Navi, backup information from the Car Navi, backup information from the device to the Car Navi, and so forth.

Car Navi USB hardware support: Shall have USB host support. Should support high speed USB. Should be able to provide 500mA.

Car Navi USB Software Support: Shall support Mass Storage Class devices as described in Section 4.2.2: “Device Class Support: MSC”

System software should recognize the standard DCF folder structure for browsing images, and may recognize the Windows Media Player folder structure for browsing music files.

Voice guidance may be enabled, so that the user can operate the device without using the display screen.

2.2 Media Player Support

Summary: UC-02: The user attaches a personal media player (PMP) to the Car Navi, and plays media back through the A/V system of the car.

Car Navi USB hardware support: shall have USB host support. If synchronization or high-bit-rate video is supported then should support high speed USB.

Class Driver requirements: shall support MTP protocol as defined in Media Transfer Protocol Enhanced V0.96. Shall operate as an Initiator.

Many media players rely on Windows-like behavior from the initiator. For example, when the library is almost full Windows Media Player will detect this and not attempt to add more media to the library. Car Navis should follow similar policies.

Car Navi USB Software Support: Shall support Mass Storage Class devices as described in Section 4.2.2: “Device Class Support: MSC”

System software should support the MTP function as part of a composite or compound device. Therefore the system should support USB composite devices, and USB hubs.

2.3 Bluetooth Add-on Support

Summary: UC-05: The user wants to wirelessly connect their cell-phone to their Car Navi. The user buys a Bluetooth adapter, and attaches it to the Car Navi, then connects the Car Navi to their cell-phone for voice calls.

Car Navi USB hardware support: shall have USB host support.

Class Driver requirements: shall support Bluetooth protocol as defined in the Bluetooth Core Specification Version 3.0 Volume 4 Part B.

2.4 Contact Synchronization Support

Summary: UC-06: The user wants to easily call their contacts by initiating a call directly from the Car Navi user interface. The user connects their cell-phone to the Car Navi to synchronize contact information with the phone, and can select their contacts directly.

Car Navi USB hardware support: shall have USB host support.

Class Driver requirements: shall support OBEX protocol.

System software should support the OBEX function as part of a composite or compound device. Therefore the system should support USB composite devices, and USB hubs.

There are various mobile phone protocols such as OBEX, however the exact specifications may be dependent on the vendor. For example, MCPC GL-004, MCPC GL-005, MCPC GL-007, USB communications device class (USB CDC) among others.

2.5 Voice Call Support

Summary: UC-07: The user wants to carry out a voice call using the Car Navi hands-free calling interface. The user connects their cell-phone to the Car Navi, chooses a contact to dial, then uses the Car Navi's hands-free functions to hold their conversation.

Car Navi USB hardware support: shall have USB host support.

Class Driver requirements: Calls may be initiated over an AT Command port, or using various other vendor-specific methods.

The audio channel shall be carried over either a separate USB audio interface, or over an analog channel. The host shall detect what the device can support and use the appropriate method.

System software should support the voice call function as part of a composite or compound device. Therefore the system should support USB composite devices, and USB hubs.

MCPC USB-SWG is presently working on production of guidelines for voice communications using USB interfaces and the outcome will need to be taken into consideration.

2.6 Digital Camera Support

Summary: UC-08: The user wants to transfer photos from a Digital Camera to the Car Navi. The user connects the camera to the Car Navi, and can use the on-screen menus to browse the photos, use them for point data information, etc.

Car Navi USB hardware support: shall have USB host support. Should support high speed USB.

Class Driver requirements: shall support PTP protocol.

System software should support the PTP function as part of a composite or compound device. Therefore the system should support USB composite devices, and USB hubs.

2.7 Camera Memory Card Support

Summary: UC-09: The user wants to transfer photos from a memory card to the Car Navi. The user connects a memory card reader to the Navi and can browse the card.

Car Navi USB hardware support: shall have USB host support. Should support high speed USB. Should be able to provide 500mA.

Car Navi USB Software Support: shall support Mass Storage Class devices as described in Section 4.2.2: “Device Class Support: MSC”

System software should recognize the standard DCF folder structure for browsing images, and may recognize the Windows Media Player folder structure for browsing music files.

2.8 Personal Navigation Device Support

Summary: UC-10: The user connects the PND or cell-phone to the Car Navi. They want to transfer map data or point info around car.

Car Navi USB hardware support: shall have USB host support. Should support high speed USB.

Standardization of map data and map data transfer formats are being developed by several organizations such as ISO/TC204/WG3, Kiwi-W Consortium, and iFormat Consortium. Applicable formats for map data and map data transfer require further discussion.

2.9 Route Planning Support

Summary: UC-11: The user wants to make a plan for driving, using the computer in their house. The user creates the plan, and saves it on a USB flash drive. The user connects the USB flash drive to the Car Navi. The Navi reads the information and guides the user.

Car Navi USB hardware support: shall have USB host support. Should support high speed USB. Should be able to provide 500mA.

Car Navi USB Software Support: shall support Mass Storage Class devices as described in Section 4.2.2: “Device Class Support: MSC”

Voice guidance may be enabled, so that the user can operate the device without using the display screen.

3. User Interface and System Behavior

3.1 Accessing the Connection Management Menu

Because the USB system sometimes needs to be explicitly managed by the user, complying systems shall offer the user a Connection Management menu. This menu should be accessible from the top-level menu of the system in a simple way. The menu should have both a graphic and a text representation. If a text representation is provided, it shall be the “Connection Management” button. User documentation for the system shall consistently call this the “Connection Management” button.

Justification:

- The easiest way for users to access “connection-related” control menus is to have a dedicated button or icon. This button or icon shall have a standardized label.
- For purposes of verbal communication (e.g., for telephone support), this button or icon shall have a consistent name. For example, the “Connection Management” button.
- This level of consistency will ensure that the user can move from car to car without struggling to remember how to access the key menus and perform basic tasks.

3.2 Uses of Connection Management Menu

The Connection Management menu is where all “connection-related” user interface elements are centralized.

In addition to supporting features for Wired USB interconnect, the Connection Management Menu might be used in future for Wireless USB.

For wired USB, the main purpose is for those device classes and use cases that require user interaction. The Car Navi should be designed so that the Connection Management menu need not be accessed in order to use simple accessories like a keyboard, a mouse and so forth.

3.3 System Issue: OTG

If a Car Navi supports USB OTG, there are three configurations:

- Car Navi connected to Accessory Device. In this case, the Car Navi will automatically become a USB host. No entries are needed in the Connection Management menu to control use of this mode of operation. However, UI elements are needed to support notifying the user about error conditions (such as failure due to insufficient power)

- Car Navi connected to PC. In this case, the Car Navi will automatically become a USB Device. No entries are needed in the Connection Management menu to control the use of this system configuration.

- Car Navi connected to another OTG “Smart Device” (DRD). In this case, the USB cable might be connected in either orientation (Car Navi might be either the OTG A-device or the OTG B-device). The Connection Management menu therefore must contain UI elements to control the operation of the transport (e.g. to cause SRP).

3.4 Device Issue: Charging

Users should be able to charge their mobile devices over USB.

Standard power is 500mA, however the USB Charging specification might allow more power to be provided. A quick charge may also be possible using USB Battery Charging 2.0.

4. Car Navi Host Support

This section describes the requirements that a Car Navi shall meet when operating as a USB Host.

4.1 Core Support

Car Navi should comply with the USB 1.1 or USB 2.0 specification. USB 1.1 should apply in the following three cases: low speed only, low speed and full speed, or full speed only. USB 2.0 should apply in the following three cases: full speed only, full speed and hi speed, or low speed and full speed and hi speed.

Car Navis should support the following device classes when operating as a USB host.

- Composite device support, including support for Interface Association Descriptors, Audio Class 1.0 Audio Control Descriptor, USB CDC Union descriptor, MCPC GL-004/005 Union descriptor, and simple one interface per function. Many devices may be composite devices, without the user being aware of this fact. Devices such as cell phones are very likely to be composite devices.
- If the Car Navi supports High-Speed USB, then the Car Navi should provide support for all the test features required by the Embedded High-Speed Host Electrical Test Procedure Version 1.01, available from USB-IF. This provides a common way for testing to be done across platforms.

When operating as a USB host, Car Navis should support the device classes required by the supported use-cases.

4.2 Details of Device Class Implementations

The details of device class implementations are given in following subsections.

4.2.1 Device Class Support: HID

HID should be supported to enable usage of devices such as a keyboard, mouse or game controller.

A HID class drive implementation can be Basic or Advanced.

A Basic HID is described as follows:

- HID keyboard supported in “boot mode” (as defined in the HID specification)
- HID mouse supported in “boot mode” (as defined in the HID specification)
- No need for HID descriptor parsing
- No need for full HID class framework

An Advanced HID is described as follows:

- Can support any HID device
- Requires a HID descriptor parser and a general framework for connecting class drivers to the parsed descriptor

4.2.2 Device Class Support: MSC

Class Driver requirements: shall support USB Mass Storage Bulk Only Transport. May support USB Mass Storage Control Bulk Interrupt.

USB Mass Storage Protocol support: Should support Transparent SCSI protocol on the mass storage device.

The Class Driver should be able to support any devices that comply with the USB Mass Storage bootability specification. There is a USBCV test that tests for compliance with the bootability specification, and this provides a baseline for devices.

To accommodate flash drives that are composite devices, system software should support mass storage as part of a composite device. To accommodate flash drives that are compound devices with an embedded hub, system software should support hubs.

Surprise removal is dangerous while devices are writing data as data loss may occur. The user interface shall warn the user not to remove the device during writes. If the user removes the device during a write operation, the user interface should warn the user not to do that again.

System software should provide FAT-32 file-system support. In addition, system software may provide UFS file-system support (for larger media).

If external DVD/CD-ROM support is needed, care must be taken about what command set is supported.

4.2.3 Device Class Support: Composite

Support for composite devices is important for newer devices. In particular the Car Navi should support the follow kinds of composite devices:

- Using Interface Association Descriptor
- Using Communications Class UNION descriptor
- Using MCPC GL-004/005 UNION descriptor
- Using Audio Class 1.0 Audio Control interface descriptor

Multifunction devices may contain some functions that Car Navi cannot support. Those functions that are not supported need to be ignored.

4.2.4 Device Class Support: Hub

Hubs should be supported to enable usage of devices with an embedded hub, or for general expansion purposes.

The USB specification allows up to 6 tiers of hubs. This allows for a user to attach 20 or 30 hubs simultaneously. Car Navis may support a reduced number of hubs to conserve memory.

Car Navis supporting hubs should support up to two tiers of hubs, connected such as Host→Hub→Hub→Device. At least two tiers are desirable because devices may have one or more hubs electrically embedded in them to support certain functionality. The user cannot distinguish whether a device is a standalone device or if it is a combination of two discrete devices on a single circuit board with an embedded hub. Additionally, some 7-port hubs are actually two 4-port hubs on a single circuit board. By supporting up to two tiers of hubs, a Car Navi will ensure that devices will work, regardless of any internal configuration unknown to the user.

If the Car Navi supports High-Speed USB, the Car Navi should support transaction translators. To reduce memory requirements, the Car Navi may support a reduced number of transaction translators, as compared to the number of attachable devices.

When system limitations are encountered, the Car Navi should display a notice of the limitation to the user, with a suggestion on how to enable device operation.

4.2.5 Device Class Support: Audio

Audio device class support is used for headsets and speakers. There are two version of Audio class: Audio 1.x and Audio 2.x. Audio 1.0 targets full speed devices. Audio 2.0 targets both full speed and high speed devices.

Most devices are still using Audio 1.0.

4.2.6 Device Class Support: Cell Phones

Device class support for cell phones is primarily used for connecting to mobile handsets.

While mobile handsets look alike to users, there is not much common behavior from the handsets.

- Feature phones all have different USB behaviors and capabilities as devices. Network operators use different features.
- Datacard adapters are generally vendor-specific
- Smart phone capabilities depend on the OS of mobile handset

Should target devices that follow standards:

- MCPC GL-004/005; MCPC GL-006; USB CDC WMC ACM, OBEX; NCM

There should also be application support, such as:

- Synchronization over OBEX or other transport
- DLNA over Ethernet-like transport

4.2.7 Device Class Support: Wireless LAN

As of the time of this document’s publication, there is no standard device class for Wireless LAN.

4.2.8 Device Class Support: Video

This device class is primarily used for capturing video streams and transferring video streams to a recording device.

4.2.9 Device Class Support: Still Image Class

This device class is primarily used for transferring images from digital still cameras (DSC) using Picture Transfer Protocol (PTP). It is also used for transferring media from personal media players (PMP) using Media Transfer Protocol (MTP).

For MTP, DRM may be needed but it is out of the scope of this specification.

There are special rules for how to distinguish a DSC from a PMP, this is covered by the MTP specifications.

5. Car Navi Implementation Constraints

This section deals with additional implementation constraints on the Car Navi.

5.1 Common Connection Management Menu – Implementation Constraints

The "Connection Management" menu is accessed via the "Connection Management" button. It should display a menu with common entries.

5.2 Car Navi Implementation Requirements (USB-Host)

As a USB host, a Car Navi should support the following kinds of devices:

- MCPC GL-004/005 devices (as native WUSB – but only a subset, and without support for isochronous)
- MCPC GL-007 devices (as native WUSB, similar constraints to GL-004/005)
- CDC/WMC ACM modems (essentially same as GL-007)
- Mass Storage
- Ethernet transport: e.g., CDC Ethernet Control Model, NCM, etc
- PTP/MTP responders (Still Image Class)
- HID (keyboard and mouse)
- Hub class
- Composite class
- Audio

Vendors may implement subsets (or supersets) of the above list.

5.3 Car Navi Requirements (USB-Device)

As the Car Navi may be connected to other USB hosts, the following device classes should be supported:

- PTP/MTP responder with PictBridge support – use case: printing images that are stored in the Car Navi
- Mass Storage – use case: moving media files to/from entertainment portion of Car Navi to a PC
- Ethernet transport (ECM, NCM) – use case: Car Navi contains a WAN connection module and can export it to PCs in the car
- HID class – use case: allow a PC to control the Car Navi remotely
- GPS info (NEMA-183) over CDC ACM, MCPC GL-004/005, etc.

6. Test and Verification

Conformance to the specification should be testable, by which a test COULD be run to verify conformance, but does not strictly mean that a test SHALL be run.

UI elements, vocabulary usage in manuals and verifying device class support is relatively easy to check. For device side verification, rudimentary USB tests can be checked with programs like USBCV and MCCI's WMCDVT, among others. These programs can be used to roughly check correctness of descriptors and first-order correctness of implementation for a given device.

To verify host class operation, it may be necessary to build reference "class devices" and some kind of test procedures to verify minimal functionality. This requires the creation of test devices that present "corner case" behavior to the host-under-test.