ETSI TS 102 600 V10.1.0 (2020-09)

Smart Cards;
UICC-Terminal interface;
Characteristics of the USB interface
(Release 10)

Reference

RTS/SCP-T060808va10

Keywords

interface, smart card

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Card Platform (SCP).

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC SCP modifies the contents of the present document, it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z.

where:

- x the first digit:
  - 0 early working draft;
  - 1 presented to TC SCP for information;
  - 2 presented to TC SCP for approval;
  - 3 or greater indicates TC SCP approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The present document defines the USB interface between a Terminal and an Integrated Circuit Card (ICC) that may be supported by the UICC and terminal in addition to the interface specified in ETSI TS 102 221 [1].

The USB interface may be implemented on UICCs and terminals for applications requiring a high data throughput or other features not supported by the interface defined in ETSI TS 102 221 [1].

The aim of the present document is to ensure interoperability between a UICC and a terminal independently of the respective manufacturer, card issuer or operator. The present document does not define any aspects related to the administrative management phase of the USB UICC. Any internal technical realization of either the UICC or the terminal is only specified where these are reflected over the interface.

1 Scope

The present document specifies the Inter-Chip USB interface between the USB UICC and the USB UICC-enabled terminal, subsequently referred to as the IC-USB interface. It describes:

- the characteristics of the Inter-Chip USB electrical interface between the USB UICC and the USB UICC-enabled terminal;
- the initial communication establishment and the transport protocols;
- the communication layers between the USB UICC and the USB UICC-enabled terminal.

The physical characteristics (including mechanical aspects) defined in ETSI TS 102 221 [1] apply to USB UICCs. The present document comes as an extension of ETSI TS 102 221 [1] complementing the electrical characteristics of contacts C1 and C5 and describing the behaviour of contacts C4 and C8 when the USB interface is supported.

The Inter-Chip USB interface provides access to the existing UICC resources such as the file system and security features specified in ETSI TS 102 221 [1] and to other resources and functionalities specified in the present document.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics".

[2] ETSI TS 102 223: "Smart Cards; Card Application Toolkit (CAT)".

[3] Universal Serial Bus Specification Revision 2.0, USB Implementers Forum.

NOTE: Available at http://www.usb.org/developers/docs. This is a ZIP package containing the following:

- The original USB 2.0 specification released on April 27, 2000.
- The "USB On-The-Go supplement" Revision 1.3 as of December 5, 2006.
- The "Inter-Chip USB supplement to the USB 2.0 Specification" Revision 1.0 March 13, 2006.
- Errata and Engineering Change Notices.

In the context of the present document, this reference, abbreviated as "USB 2.0", is used specifically in relation to the original USB 2.0 specification and associated errata and Engineering Change Notices, while its supplements are referred through separate references.

[4] "Inter-Chip USB supplement to the USB 2.0 Specification", Revision 1.0 March 13, 2006 published as part of the Universal Serial Bus Revision 2.0 specification package.

NOTE: Available at http://www.usb.org/developers/docs.

[5] Void.

[6] "Universal Serial Bus Mass Storage Class Bulk-Only Transport" Revision 1.0.

NOTE: Available at https://www.usb.org/documents.

[7] Universal Serial Bus: "Device Class: Smart Card ICCD Specification for USB Integrated Circuit(s) Card Devices" Revision 1.0.

NOTE: Available at https://www.usb.org/documents.

[8] "Universal Serial Bus Common Class Specification", Revision 1.0.

NOTE: Available at https://www.usb.org/documents.

[9] "Universal Serial Bus Communications Class Subclass Specification for Ethernet Emulation Model Devices", Revision 1.0, USB Implementers Forum, Device Working Group: Communication.

NOTE: Available at https://www.usb.org/documents.

[10] INCITS 405-2005: "SCSI Block Commands - 2 (SBC-2)".

NOTE: Available at https://www.t10.org/cgi-bin/ac.pl?t=d&f=99-292r1.pdf.

[11] INCITS 408-2005: "SCSI Primary Commands - 3 (SPC-3)".

NOTE: Available at https://www.t10.org/cgi-bin/ac.pl?t=d&f=99-294r1.pdf

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] IETF RFC 4122: "A Universally Unique IDentifier (UUID) URN Namespace".

NOTE: Available at http://www.ietf.org/rfc/rfc4122.txt.

3 Definition of terms, symbols, abbreviations and coding conventions

3.1 Terms

For the purposes of the present document, the following terms apply:

APDU-based application: UICC application designed to use APDUs as specified in ETSI TS 102 221 [1].

application: computer program that defines and implements a useful functionality on a smart card or in a terminal

NOTE: The term may apply to the functionality itself, to the representation of the functionality in a programming language, or to the realization of the functionality as executable code.

attachment: electrical process by which a USB peripheral, such as a USB UICC, indicates its presence to its host

Class A operating conditions: terminal or a smart card operating at 5 V ± 10 %

NOTE: See ETSI TS 102 221 [1].

Class B operating conditions: terminal or a smart card operating at 3 V ± 10 %

NOTE: See ETSI TS 102 221 [1].

Class C operating conditions: terminal or a smart card operating at 1,8 V ± 10 %

NOTE: See ETSI TS 102 221 [1].

Class C' operating conditions: terminal or smart card operating at 1,8 V ± 0,15 V

NOTE: See Inter-Chip USB [4].

configured: state reached by a USB device when its USB host may use the functionalities that it provides, i.e. after the device has correctly processed a SetConfiguration() request with a non-zero configuration value

NOTE: As defined in USB 2.0 [3].

endpoint: uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the host and device

ETSI TS 102 221 [1] interface: asynchronous serial UICC-Terminal interface defined in ETSI TS 102 221 [1], using RST on contact C2, CLK on contact C3 and I/O on contact C7

functional interface: set of USB endpoints associated with specific transfer type characteristics and described by an interface descriptor

NOTE: As specified in USB 2.0 [3].

IC-USB interface: Inter-Chip USB interface between the USB UICC and the USB UICC-enabled terminal

Inter-Chip USB: electrical interface for chip-to-chip connections over short distances, specified in a supplement to the USB 2.0 specification [3]

NOTE: As only the electrical link is affected by the Inter-Chip supplement, Inter-Chip USB products are compatible with (standard) USB compliant drivers and software.

Inter-Chip USB family: family of USB hosts and removable USB peripherals is defined as a set of hosts and peripherals having matching mechanical interfaces

NOTE: Within the family, any choice of host and peripheral are able to communicate.

Smart Card functional interface: functional interface supporting the transfer of APDUs over Version B Control transfer or a pair of bulk pipes

NOTE: As defined in clause 9.1.

State H: high state on a signal line (Vcc)

State L: low state on a signal line (Gnd)

suspended: indicates a state where the USB interface of the USB UICC is in Suspend state

NOTE: As defined in USB 2.0 [3].

USB UICC: UICC which supports the interface using Inter-Chip USB [4] as specified in the present document

USB UICC-enabled terminal: terminal which supports the host interface using Inter-Chip USB [4] as specified in the present document

3.2 Symbols

For the purposes of the present document, the following symbols apply:

Gnd	Ground
IC_DM	Inter-Chip USB D- data line
IC_DP	Inter-Chip USB D+ data line
IC_VDD	Inter-Chip USB Power Supply Voltage
Vcc	UICC Supply Voltage
V_{IH}	Input Voltage (high)
V_{IL}	Input Voltage (low)
V_{OH}	Output Voltage (high)
V_{OL}	Output Voltage (low)

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

APDU	Application Protocol Data Unit
ATR	Answer To Reset
CAT	Card Application Toolkit
CDC	Communication Device Class
CLK	ClocK
EEM	Ethernet Emulation Model
FCP	File Control Parameters
I/O	Input/Output
ICC	Integrated Circuit Card
ICCD	Integrated Circuit Card Device
IC-USB	Inter-Chip USB
INCITS	InterNational Committee for Information Technology Standards
IP	Internet Protocol
MAC	Media Access Control
MBR	Master Boot Record
PPS	Protocol and Parameter Selection
RFU	Reserved for Future Use
RST	ReSeT
SCSI	Small Computer System Interface
SOF	Start Of Frame
USB	Universal Serial Bus

3.4 Coding conventions

For the purposes of the present document, the following coding conventions apply:

- All lengths are presented in bytes, unless otherwise stated. Each byte is represented by bits b8 to b1, where b8 is the Most Significant Bit (MSB) and b1 is the Least Significant Bit (LSB). In each representation, the leftmost bit is the MSB.
- Hexadecimal values are enclosed in single quotes ('xx').

In the UICC, all bytes specified as RFU shall be set to '00' and all bits specified as RFU shall be set to 0.

4 USB UICC system architecture

4.1 Support of the ETSI TS 102 221 interface

USB UICCs and USB UICC-enabled terminals shall remain compliant with ETSI TS 102 221 [1]. The electrical characteristics of contacts C1 and C5 are specified in the present document for a USB UICC and a USB UICC-enabled terminal. Contacts C2, C3, C6 and C7 shall behave as specified in ETSI TS 102 221 [1]. The behaviour of contacts C4 and C8 shall be as specified in the present document.

4.2 Configurations

Three new terminal/UICC configurations are possible:

- A terminal with only ETSI TS 102 221 [1] capability connected to a USB UICC. The ETSI TS 102 221 [1] interface shall be activated.
- A USB UICC-enabled terminal connected to a UICC with only ETSI TS 102 221 [1]. The ETSI TS 102 221 [1] interface shall be activated.
- A USB UICC-enabled terminal connected to a USB UICC. The interface shall be selected according to the procedures specified in the present document.

Commands and functionality specified in ETSI TS 102 221 [1] shall also be supported over the IC-USB interface. The IC-USB interface may support additional functionality not available on the ETSI TS 102 221 [1] interface.

4.3 Interworking with the ETSI TS 102 221 interface

The selection of the ETSI TS 102 221 [1] interface and IC-USB interface shall be exclusive as specified in clause 7.2.

Selection of the IC-USB interface is defined to have occurred when either a PPS procedure indicating switching to the IC-USB interface, described in clause 7.2, has been performed on the ETSI TS 102 221 [1] interface, or when the terminal has successfully configured the USB UICC on the IC-USB interface.

Selection of the ETSI TS 102 221 [1] interface is defined to have occurred when the terminal sends a PPS procedure not indicating switching to the IC-USB interface or any APDU command following an ATR on the ETSI TS 102 221 [1] interface.

Except for contacts C1 and C5, actions by an entity (terminal or UICC) on one interface shall not affect the state of the other interface.

The terminal shall always drive C2 and C3 to a defined state.

A USB UICC shall indicate the support of USB in its ATR, as described in ETSI TS 102 221 [1].

5 Physical characteristics

5.0 Introduction

The physical characteristics of the USB UICC-Terminal interface are as defined in ETSI TS 102 221 [1] except for the specific provisions specified in the present document.

5.1 Contacts

5.1.1 Provision of contacts

5.1.1.1 Terminal

A USB UICC-enabled terminal shall provide contacts C4 and C8 with the mechanical characteristics defined in ETSI TS 102 221 [1].

The IC-USB power signals, IC_VDD and GND, are respectively applied on the USB UICC Vcc (C1) and Gnd (C5) contacts.

The Inter-Chip USB lines, IC_DP and IC_DM, are respectively assigned to C4 and C8.

5.1.1.2 UICC

A USB UICC shall provide contacts C4 and C8.

The IC-USB power signals, IC_VDD and GND, are respectively assigned on the USB UICC Vcc (C1) and Gnd (C5) contacts.

The Inter-Chip USB lines, IC_DP and IC_DM, are respectively assigned to C4 and C8.

5.1.2 Contact activation and deactivation

Following power up, a USB UICC shall present a high impedance state on contacts C4 and C8 within 80µs following establishment of a stable power supply.

When the IC-USB interface is suspended or no USB UICC is attached, a USB UICC-enabled terminal may turn off Vcc without further action on C4 and C8.

5.1.3 Inactive contacts

The voltages on contacts C4 and C8 of the terminal shall be in the range 0 V ± 0,4 V referenced to Gnd (C5) when the terminal is switched off with the power source connected to the terminal, while measured with a measurement equipment having a resistance of 50 kΩ.

5.2 UICC insertion and removal

USB UICCs shall not be damaged when inserted in or removed from a slot where power is present.

6 Electrical characteristics

6.1 Operating Conditions

6.1.0 Introduction

The operating conditions defined in ETSI TS 102 221 [1] apply to USB UICCs and USB UICC-enabled terminals, except when otherwise specified in the present document.

The contacts C4 and C8 operate as specified in Inter-Chip USB [4] for IC_DP and IC_DM respectively.

For V_{IH} and V_{IL}, Vcc refers to the receiving device power supply level. For V_{OH} and V_{OL}, Vcc refers to the sending device power supply level. All voltages are referenced to Gnd. For each state (V_{OH}, V_{IH}, V_{IL} and V_{OL}), a positive current is defined as flowing into the entity (terminal or UICC).

6.1.1 Class B operating conditions

When the USB UICC-enabled terminal and the USB UICC are operating under class B operating conditions, the supply voltage on C1 and C5 shall be as defined in ETSI TS 102 221 [1]. The operation of contacts C4 and C8 shall follow the requirements specified in Inter-Chip USB [4] for the Voltage Class 3.0 Volt.

6.1.2 Class C' operating conditions

When the USB UICC-enabled terminal and the USB UICC are operating at a nominal supply voltage of 1,8 V, the supply voltage on C1 and C5 and the operation of contacts C4 and C8 shall follow the requirements specified in the Inter-Chip USB [4] for the Voltage Class 1,8 Volt. This is defined as Class C' operating conditions, as the supply voltage definition is tighter than the definition of supply voltage class C used in ETSI TS 102 221 [1].

7 Initial communication establishment procedures

7.1 Supply voltage selection

USB UICCs shall support voltage class B and C/C' operating conditions, while a USB UICC-enabled terminal shall support voltage class C' and may support voltage class B. Some USB UICCs may support enhanced capabilities when activated under Class B operating conditions. This is indicated by setting a "Class B activation preferred" indicator as part of their USB interface power negotiation (see clause 8.2 of the present document).

Any voltage class defined in the present document which is supported by the USB UICC shall be supported on both the IC-USB interface and the ETSI TS 102 221 [1] interface.

A USB UICC-enabled terminal shall perform the supply voltage selection as follows:

- The terminal shall initially select its lowest supported voltage class.
- The terminal shall power up the UICC with the selected voltage class and start the interface selection procedure defined in clause 7.2.
- If no attachment occurs and no ATR is received during the interface selection procedure, the UICC shall be deactivated and activated with the next higher class if supported by the terminal.
- In case an ATR is received on the ETSI TS 102 221 [1] interface if the voltage class used by the terminal is not indicated as supported by the UICC, the terminal shall deactivate the UICC. If an indicated voltage class is supported by the terminal, the terminal shall continue as specified in ETSI TS 102 221 [1].
- In case the voltage class used by the terminal is not indicated as supported by the UICC in the response to the Get Interface Power request, the terminal shall deactivate the UICC.
- If only the procedure using ATR is performed and a corrupted ATR is received the terminal shall perform the procedure at least 3 times using the same voltage class. In case of 3 or more consecutive failures, the terminal shall continue as specified in ETSI TS 102 221 [1].
- If under a voltage class different from class B, the data retrieved by a Get Interface Power request indicates "Class B activation preferred", a USB UICC-enabled terminal that wants to use the features only available in Class B shall power down the UICC and power it up with supply voltage class B.

USB UICCs may not always attach over IC-USB when hot plugged in a powered slot of a USB-enabled terminal.

7.2 Interface selection

The following three steps shall be performed on the IC-USB interface independently of the procedures defined hereafter:

- The USB UICC-enabled terminal shall activate its pull-down resistors on C4 and C8 from the beginning of the power up phase, as specified in Inter-Chip USB [4].
- Before attachment, the USB UICC shall present high impedance on C4 and C8 and shall monitor the signals on C4 and C8. To allow for other procedures to co-exist on the same contacts, the USB UICC shall continue with the attachment procedure only if one of the following conditions is met:
    - C4 and C8 are maintained in state L by the terminal for at least 10 ms after the supply voltage has reached a valid operation level;
    - the condition described in the procedure using ATR is met.
- The USB UICC attaches itself as a USB Full-Speed device by pulling the C4 line to state H as specified in USB 2.0 [3]. In case this action causes the C8 line to go to state H simultaneously, the USB UICC shall immediately terminate the USB attachment and activate its IC-USB pull-down resistors on contacts C4 and C8.

Two procedures are defined for the detection of the presence of a USB UICC by the USB UICC-enabled terminal. The terminal may perform only one of these procedures or perform both in parallel asynchronously.

Procedure using USB:

- USB UICC-enabled terminals do not need to provide a clock signal on contact C3 to operate a USB UICC. However, if only the procedure using USB is used, then immediately after applying power to the UICC, it is recommended that the USB UICC-enabled terminal provides a clock on contact C3 compliant with ETSI TS 102 221 [1] for at least 200 cycles while maintaining C2 in state L to allow UICCs supporting only the ETSI TS 102 221 [1] interface to assert the state of all their contacts. It is recommended that the terminal switches off this clock after that. USB UICCs shall support switching off of the clock as long as C2 is kept in state L.
- The terminal shall detect whether a USB UICC is present as described in Inter-Chip USB [4]. If no USB UICC is attached, i.e. C4 is not in state H, the USB attachment failed.
- If a USB attachment is detected, the terminal shall drive a USB Reset as specified in Inter-Chip USB [4].

Procedure using ATR:

- The terminal initiates the UICC activation procedure specified in ETSI TS 102 221 [1] until an ATR is received from the UICC.
- If a UICC not supporting IC-USB is recognized according to the ATR indication mechanism of ETSI TS 102 221 [1], the terminal shall continue as defined in ETSI TS 102 221 [1].
- If a USB UICC is recognized according to the ATR indication mechanism of ETSI TS 102 221 [1], the USB UICC-enabled terminal shall send a PPS request indicating T = 15 with PPS2 set to 'C0' in conformance with the first TBi (i > 2) of the ATR to indicate switching to the IC-USB interface.
- Upon receiving the special PPS command indicating T = 15 with PPS2 set to 'C0', the UICC shall attach on USB if this has not already happened before replying to the PPS command. A USB UICC receiving this PPS shall remain attached on USB until the terminal drives a USB Reset.
- If the USB UICC receives any other command following ATR than the special PPS command indicating T = 15 with PPS2 set to 'C0' (this would be the case when the UICC is inserted in a terminal not supporting the interface specified in the present document), it shall terminate any actions on contacts C4 and C8 and activate its pull-down resistors. The USB UICC shall not attempt to attach itself on USB again until it has been powered down and up.
- Upon receiving the PPS response, the terminal may immediately drive a USB Reset if this has not already happened.

- After a successful PPS exchange indicating T = 15 with PPS2 set to 'C0', the terminal may stop the clock on the ETSI TS 102 221 [1] interface and the USB UICC shall no longer react to events, such a new commands, on the ETSI TS 102 221 [1] interface.
- A USB UICC shall fully execute any command already initiated on the ETSI TS 102 221 [1] interface even if a USB Reset is received before completion. This shall not prevent the UICC from operating normally on the USB interface following the USB Reset.

For both procedures, after the USB Reset has occurred, the USB activation procedure shall continue as described in clause 7.3.

7.3 IC-USB interface activation

Activation of the IC-USB interface shall be performed as follows:

- USB UICC ADDRESS ASSIGNMENT: The terminal assigns a unique address to the USB UICC as specified in USB 2.0 [3].
- POWER NEGOTIATION: The USB UICC and the terminal exchange information about voltage classes and current consumption as defined in clause 8.2.
- USB UICC CONFIGURATION: The terminal configures the USB UICC for the applications it is running, as described in USB 2.0 [3]. Terminal applications using the IC-USB interface should specify the behaviour of a terminal when a USB function that it expects is not available on a USB UICC (the terminal may, e.g. inform the user of a mismatch and attempts to activate the ETSI TS 102 221 [1] interface).

If the terminal reached the configuration stage but could not successfully configure the USB UICC for at least the ICCD interface, the terminal shall power down and power up the USB UICC at the same voltage class and select the ETSI TS 102 221 [1] interface, ignoring the USB information that may be provided by the USB UICC in its ATR.

7.4 Power consumption

7.4.1 Power consumption of the USB UICC during activation

Under all operating conditions, a USB UICC-enabled terminal shall be able to supply at least 10 mA until either a power negotiation occurs on the IC-USB interface or the ETSI TS 102 221 [1] interface is selected.

The power consumption of the USB UICC shall remain within the limit that applies during ATR at maximum external clock frequency as specified in ETSI TS 102 221 [1] until it has received a USB Reset signalling from the terminal.

7.4.2 Application related electrical parameters

If the IC-USB interface is selected, after a successful power negotiation procedure, the USB UICC-enabled terminal shall be able to supply the power negotiated in the procedure and the UICC shall not exceed the negotiated power limit.

Applications based on the present document may specify a minimum power supply capability for their supporting terminals. A UICC supporting such an application shall be able to activate the USB interface and run the application with no more than the minimum current value specified for that application, and may use the power negotiation procedure to offer enhanced performance.

7.4.3 Relation with other interfaces

When its IC-USB interface is not activated or is suspended, and no other UICC interfaces are active, the USB UICC current consumption at 25°C shall not exceed the values specified in ETSI TS 102 221 [1] for a UICC in idle state.

7.5 Answer To Reset content

The ATR returned by a USB UICC activated using the USB ICCD device class on the IC-USB interface in response to an ICC_POWER_ON or a PC_to_RDR_IccPowerOn request according to the Smart Card ICCD specification [7] shall be the same as the ATR that would be returned over the ETSI TS 102 221 [1] interface after a cold reset.

NOTE: To always retrieve the full ATR, the terminal has to set the length in the request to a value that allows the UICC to return an ATR of up to 33 bytes.

ATR parameters not relevant for the IC-USB interface (e.g. clock stop mode, supply voltage class) shall be ignored by the USB UICC-enabled terminal when received as response to an ICC_POWER_ON or a PC_to_RDR_IccPowerOn request.

7.6 USB UICC as an Inter-Chip USB peripheral

The USB UICC behaves as a removable Inter-Chip USB peripheral as specified in Inter-Chip USB [4]. Interoperable USB UICCs (the peripherals) and USB UICC-enabled terminals (the hosts) constitute an Inter-Chip USB family, characterized by the following features:

1) The host and the peripheral have mechanical interfaces that interlock with each other, i.e. the form factors specified in ETSI TS 102 221 [1].

2) Any host and peripheral support a common set of electrical parameters, i.e. class C' operating conditions.

3) Any host and peripheral support at least full-speed USB operation.

To minimize power consumption, USB UICCs shall support dynamic switching of their resistors on C4 and C8 during traffic signalling as described in Inter-Chip USB [4].

7.7 Suspend, Resume and Remote Wakeup

The USB UICC shall support Suspend and Resume states as defined in USB 2.0 [3]. The USB UICC shall enter Suspend state after the bus has not transmitted any data for 3 ms, in compliance with USB 2.0 [3]. The terminal should only suspend operation of the USB UICC interface when the suspend conditions are met for all activated functional interfaces according to clause 9. If those conditions are not satisfied and a suspend occurs, the state of the USB UICC may become undefined and a USB Reset may be required to recover from this state.

Applications based on other functional interfaces should specify their conditions for entering Suspend state.

While in Suspend state, the USB UICC may support remote wakeup. The host may enable this capability using standard USB requests when desired. In order to perform a remote wakeup, the USB UICC shall perform a Resume signalling as described in USB 2.0 [3]. If the UICC supports remote wakeup signalling for minimum 10 ms, see clause 8.3, then the USB UICC shall perform a Resume signalling for at least 10 ms and up to the maximum duration of 15 ms allowed in USB 2.0 [3], to allow sufficient time for the terminal to react. The optional remote wakeup time negotiation specified in clause 8.3 also allows the terminal to select other values for the remote wakeup signalling. After a remote wakeup, the terminal shall perform the wakeup actions as defined for all configured functional interfaces.

Resuming the interface is described in USB 2.0 [3]. However, after a resume time negotiation as described in clause 8.3, the minimum duration of the resume signalling and the minimum number of SOF tokens during resume recovery are the values returned by the UICC during the resume time negotiation.

If remote wakeup is enabled, the Resume signalling time shall be longer than the remote wakeup signalling time to avoid conflicts in case of both signalling happening at the same time.

7.8 USB UICC deactivation

A USB UICC-enabled terminal should properly terminate all active applications running over USB before powering off a USB UICC. When the IC-USB interface is suspended, the terminal may remove power from the USB UICC at any time.

8 USB interface operational features

8.1 Speed support

USB Full Speed, as defined in USB 2.0 [3], shall be supported on the USB UICC and USB UICC-enabled terminal.

8.2 Power Negotiation

A USB UICC shall support the Get Interface Power request and the Set Interface Power request as defined in tables 8.1 and 8.2.

Table 8.1: Get/Set Interface Power Request

Offset	Field	Size	Value	Description
0	*bmRequest*	1	'C0' / '40'	'C0' for Get Interface Power '40' for Set Interface Power Characteristics of request: b8: 1 = Device-to-host / 0 = Host-to-device b7...6: Type 2 = Vendor b5...1: Recipient 0 = Device
1	*bRequest*	1	'01' / '02'	Get Interface Power Request Set Interface Power Request
2	*wValue*	2	'0000'	
4	*wIndex*	2	'0000'	
6	*wLength*	2	'0002'	Number of bytes in the data stage

A USB UICC according to the present document shall accept *wLength* values greater than 2 in the Get Interface Power request, but only respond with returning 2 bytes. This allows for addition of new parameters in the future.

Table 8.2: Data Field for Get/Set Interface Power Request

Offset	Field	Size	Value	Description
0	*bVoltageClass*	1		Indicates the voltage classes supported by the UICC. If a class is supported, the corresponding bit is set to 1. B1 Class A, reserved for USB 2.0 (optional use, not specified by ETSI SCP) b2 Class B b3 Class C' b7…4 Reserved for Future Use, shall be set to 0 b8 Class B activation preferred (see clause 7.1)
1	*bMaxCurrent*	1		Maximum current that the UICC requires for best performance, expressed in 2 mA units. e.g. '0A' indicates 20 mA.

A USB UICC-enabled terminal shall perform power negotiation by sending a Get Interface Power Request followed by a Set Interface Power Request to the USB UICC. If the terminal does not send the Set Interface Power Request before retrieving the configuration descriptor, it shall support removable media as specified in the SCSI Primary Commands specification [11]. Removable media refers to storage media which can be activated and deactivated by the UICC. This allows the UICC to make the mass storage media available after enumeration of the Mass Storage class only if the negotiated power enables to activate it.

If the USB UICC supports mass storage, it shall support removable media on its mass storage endpoint and shall inform MEDIUM NOT PRESENT (see [11]) until the Get Interface Power Request followed by a Set Interface Power Request procedure has completed.

A USB UICC shall not signal MEDIUM PRESENT (see [11]) if terminal power capabilities are not sufficient.

A USB UICC-enabled terminal should stop the mass storage usage by sending a STOP UNIT command as specified in the SCSI Primary Commands specification [11] when it does not intend to use the mass storage capability of the UICC.

After evaluating the data received from the USB UICC with the Get Interface Power Request, the USB UICC-enabled terminal shall inform the USB UICC about its capabilities by sending a Set Interface Power Request. In *bVoltageClass*, the terminal shall set only the one bit of the voltage class that is provided to the UICC. In *bMaxCurrent*, it shall indicate the maximum current it can provide to the UICC, which shall be at least 10 mA (or at least the value requested by the UICC if this was lower than 10 mA).

From that point on, the USB UICC shall keep its current consumption within the limit indicated by the terminal. If power negotiation took place before the terminal retrieves the configuration descriptor, the USB UICC shall adapt its [11] USB configuration descriptors and interface descriptors according to the current provided by the terminal.

NOTE 1: Terminals wanting to make use of features like high density memories on an USB UICC should provide a maximum current of at least 64 mA measured with a capacitor in the range of 50 nF to 200 nF connected between Vcc and GND close to the contacting elements.

If a USB UICC is able to present different descriptors to the terminal, it shall use a unique identity in the device descriptor (combination of idVendor, idProduct and iSerialNumber) for each set of descriptors.

NOTE 2: The values for idVendor are centrally assigned by the USB Implementers Forum. Thus vendors have full control over the assignment of the unique identities.

NOTE 3: Terminals may keep copies of configuration descriptors and not retrieve them again when they see a device with the same identity again.

If a USB UICC which has set the "Class B activation preferred" indicator in its interface power descriptor is currently powered under a different voltage class by a terminal supporting class B, the terminal can decide to switch the UICC power supply to class B as described in clause 7.1.

8.3 Resume and remote wakeup time negotiation

A USB UICC shall be able to answer a Resume Time Request according to table 8.3 as defined in table 8.4. If the terminal sends this request to the UICC, it shall send it before suspending the interface for the first time.

Table 8.3: Resume Time Request

Offset	Field	Size	Value	Description
0	*bmRequest*	1	'C0'	Characteristics of request: b8: 1 = Device-to-host b7...6: Type 2 = Vendor b5...1: Recipient 0 = Device
1	*bRequest*	1	'03'	Resume Time Request
2	*wValue*	2	'0000'	
4	*wIndex*	2	'0000'	
6	*wLength*	2	'0003'	Number of bytes in the response data

The response data from the USB UICC shall consist of three bytes containing:

- the minimum resume signalling time required by the UICC;
- the minimum number of SOF tokens during resume recovery required by the UICC;
- information about whether the UICC supports remote wakeup time negotiation as defined below;
- in case remote wakeup time negotiation is not performed, information whether the UICC will maintain the resume signalling for remote wakeup for a minimum of 10 ms.

Table 8.4: Response to Resume Time Request

Offset	Field	Size	Value	Description
0	*bMinResTime*	1		Minimum resume signalling time required by the UICC. The unit is 0,1 ms. The minimum value is '0A' corresponding to 1 ms, and the maximum value is '1E' corresponding to 3 ms.
1	*bMinSofTokens*	1		Minimum number of SOF tokens during resume recovery required by the UICC before the terminal may access the UICC. (See note below). The minimum number is 1 and the maximum number is 5.
2	*bmRemWakeup*	1		Characteristics of request: b8...b3: RFU b2: Remote wakeup time negotiation 0 = Not supported 1 = Supported b1: Remote wake-up signalling 0 = Remote wake-up not supported or supported with signalling time of at least 10 ms not guaranteed 1 = Remote wake-up supported with signalling time of at least 10 ms b1 is irrelevant if the terminal performs remote wakeup time negotiation.
NOTE: As SOF tokens are sent periodically every 1 ms, this value directly relates to the resume recovery time.				

If the UICC indicates support for remote wakeup time negotiation, the terminal may send a Remote Wakeup Time Request to the UICC to indicate its requirements for the remote wakeup timing. This in turn allows the terminal to reduce the resume time also if remote wakeup is enabled; it simply has to be greater than *bMinResTime* × 0,1 ms and WakeupTimeMax.

Table 8.5: Remote Wakeup Time Request

Offset	Field	Size	Value	Description
0	*bmRequest*	1	'40'	Characteristics of request: b8: 0 = Host-to-device b7...6: Type 2 = Vendor b5...1: Recipient 0 = Device
1	*bRequest*	1	'04'	Remote Wakeup Time Request
2	*wValue*	2	'0000'	
4	*wIndex*	2	'0000'	
6	*wLength*	2	'0001'	Number of bytes in the data stage

Table 8.6: Data Field for Remote Wakeup Time Request

Offset	Field	Size	Value	Description
0	*bWakeupTime*	1		Wakeup time limits. The minimum value is '02' and the maximum value is '14'.

The UICC shall keep the wake up signal within the following limits:

WakeupTimeMin <= wakeup signal <= WakeupTimeMax

with

WakeupTimeMin = *bWakeupTime* × 0,5 ms

WakeupTimeMax = *bWakeupTime* × 0,72 ms + 0,6 ms

8.4 Pipes, endpoints and configurations

A USB UICC may communicate with the terminal using any variety of pipes defined in USB 2.0 [3] in addition to the default control pipe.

USB UICCs and USB UICC-enabled terminals shall provide at least 2 bulk endpoints (one in and one out) in addition to the default endpoint 0. It is recommended to support at least 4 bulk endpoints (two in and two out).

A USB UICC may contain several configurations for its different functional interfaces. The terminal may then switch between the different configurations while remaining in configured state as described in USB 2.0 [3]. Switching may only occur after the currently configured USB functional interface(s) is (are) in a state where the bus could be suspended. The application(s) related to the functional interfaces shall keep their internal state (e.g. file and security context or dynamically assigned IP address) when configurations are switched.

A USB UICC supporting multiple functional interfaces shall be a composite USB device, having a single USB device address.

8.5 Enumeration using standard descriptors

The standard descriptors described in USB 2.0 [3] and the Common Device Class Specification [8] provide a way for the terminal to identify a newly attached USB device, such as a USB UICC, and to activate support for this USB UICC. The standard descriptors are read by the terminal during the enumeration process as specified in USB 2.0 [3]. In addition, the descriptors can also be retrieved at a later point in time by the terminal using standard USB requests. They include configuration related information common to any USB device, as well as a description of the specific USB features of the UICC.

The standard descriptors may be complemented by class-specific descriptors depending on the USB device class(es) supported by the USB UICC. Additional vendor-specific descriptors may complement the standard descriptors to provide further information.

To uniquely identify USB UICCs as a USB device according to the present document, they shall include a descriptor of the vendor-specific category, named UICC specific descriptor as described in clause A.6. The UICC specific descriptor shall immediately follow the device descriptor.

9 Protocol stacks for USB UICC applications

9.1 Support of APDU-based UICC applications over the IC-USB Interface

9.1.0 Introduction

In order to support applications based on ETSI TS 102 221 [1] on the IC-USB interface, all USB UICC-enabled terminals shall support short APDU-level exchanges over Version B Control transfer with no Interrupt pipe, as defined in the Smart Card ICCD specification [7]. All USB UICCs shall present at least one USB configuration descriptor with short APDU-level exchanges over Version B Control transfer with no Interrupt pipe as defined in the Smart Card ICCD specification [7].

Applications relying on APDU communication to exchange large amount of data may specify one or several additional configurations using short APDU-level exchange over a dedicated pair of bulk pipes and no interrupt pipe for USB UICC-enabled terminals and USB UICC. Switching between configurations having bulk and control B interfaces shall be transparent at the application layer.

In either case, this is referred to as "Smart Card functional interface".

NOTE: Command and response APDUs (C-APDUs and R-APDUs) are transferred via the Smart Card functional interface. Only the USB protocol mechanisms are used for the transfer; no translation into TPDUs takes place and no protocol elements as defined in ETSI TS 102 221 [1] for T = 0 or T = 1 are added.

After the transition of a USB UICC from the USB device state "address" to the USB device state "configured" (whether following power up, USB reset or device deconfiguration), the terminal shall send an ICC_POWER_OFF or a PC_to_RDR_IccPowerOff request for the USB UICC to enter the "initial state" defined in the state diagrams in the Smart Card ICCD specification [7]. Contrary to what is stated in the Smart Card ICCD specification [7], when a USB UICC enters this initial state, the UICC with all its applications shall enter the same state as after a cold reset on the interface according to ETSI TS 102 221 [1].

Even though only short APDUs are currently defined a terminal shall also accept a USB UICC indicating support for short and extended APDU level exchanges in its class specific descriptor.

All applications and features based on ETSI TS 102 221 [1], such as the Card Application Toolkit defined in ETSI TS 102 223 [2], may be used in the context of APDU communication over USB. The PPS procedure does not apply when transferring APDUs over USB. Cold reset is logically performed by USB commands; an ICC_POWER_ON / PC_to_RDR_IccPowerOn request shall not be sent to the USB UICC without a preceding ICC_POWER_OFF / PC_to_RDR_IccPowerOff request. Processing time extension may be requested as defined in the Smart Card ICCD specification [7].

Specific provisions for using applications based on ETSI TS 102 221 [1] over the IC-USB interface are indicated in ETSI TS 102 221 [1]. When the USB UICC-enabled terminal uses the present interface, the content of the FCP data objects for UICC characteristics, application power consumption and minimum application clock frequency, which are specific to the ETSI TS 102 221 [1] physical interface, shall be ignored.

The suspend conditions for this functional interface is that all commands have had a complete response.

Suspend shall have no effect on the internal state of the UICC (file context, security status, etc.).

If CAT is supported, then after a remote wakeup, the terminal shall send a STATUS command on this functional interface to allow the UICC to start a proactive session.

9.1.1 Proactive Polling

All USB UICC-enabled terminals supporting CAT shall support the POLL INTERVAL and POLLING OFF proactive commands specified in ETSI TS 102 223 [2]. The default period for proactive polling using periodical STATUS commands is set to 300 seconds for USB UICC-enabled terminals to avoid a negative impact on power consumption. USB UICCs requiring a different polling frequency while using APDU communication over USB shall set it accordingly by means of a POLL INTERVAL command. When a USB UICC using APDU communication over USB has no need for proactive polling, it shall indicate it to the terminal by using the POLLING OFF command.

9.2 Support of IP applications over the IC-USB Interface

If applications require the support of the Ethernet Emulation Model subclass of the USB communication device class defined in CDC EEM [9], the requirements of this clause apply.

Support of the SuspendHint, ResponseHint and ResponseCompleteHint commands as described in CDC EEM [9] is mandatory for the USB UICC and the USB UICC-enabled terminal. The USB UICC shall send SuspendHints whenever it completes internal processing. The suspend condition for this interface is that a SuspendHint was the last EEM packet sent by the USB UICC.

If a USB UICC uses a locally administered MAC address, it is recommended to use an address of the range 82-xx-xx-xx-xx-xx.

If a USB UICC-enabled terminal uses a locally administered MAC address, it is recommended to use an address with a setting in byte 1 different from '82'.

After a remote wakeup, the terminal shall check if there is data to be transferred from the USB UICC on the bulk in pipe.

9.3 Support of mass storage applications over the IC-USB Interface

If a memory area that behaves as a storage medium not controlled by the UICC itself is supported, the requirements of this clause apply.

The USB UICC and USB UICC enabled terminal shall support the Mass Storage Bulk Only 1.0 specification [6] as explained in the Mass Storage Specification Overview [7] with the SCSI Transparent subclass '06', corresponding to support of the SCSI Primary Command set of INCITS 408-2005 [11]. The USB UICC shall support the SCSI Peripheral Device Type '00' corresponding to a direct access SCSI block device as specified in INCITS 405-2005 [10].

The first sector of the unprotected memory area shall contain an MBR with a partition table. Number, format and content of the partition(s) are beyond the scope of the present document.

The suspend condition for this interface is that no response to a command is outstanding.

9.4 Interworking of the USB functional interfaces

USB 2.0 [3] specifies that all bulk transfers shall be served based on a "fair access policy" whenever no other transfer requests are scheduled or pending. This shall imply in particular that the terminal shall check the EEM IN endpoint for data available from the UICC while the terminal is waiting for the response to an APDU currently being processed by the UICC.

NOTE: This will allow the UICC to start IP based communication after being triggered by an APDU, e.g. in the context of a connectivity event specified in ETSI TS 102 223 [2].

Annex A (normative): USB Descriptors of a USB UICC

A.0 Introduction

Within the scope of this annex, the term "interface" refers to USB functional interfaces as per USB 2.0 [3].

A.1 The Standard Device Descriptor

The Standard device descriptor for a USB UICC shall be as defined in the Smart Card ICCD specification [7].

A USB UICC may report multiple configurations to a USB UICC-enabled terminal, in which case the terminal may choose the configuration it deems appropriate.

A.2 The Standard Configuration Descriptor

The Standard device configuration for a USB UICC shall be as defined in the Smart Card ICCD specification [7].

For the purpose of the present document, the following fields shall be set as defined in table A.1.

Table A.1: Specific fields of the Standard configuration descriptor

Field	Value	Description
bmAttributes	'80' or 'A0'	'80' corresponds to a USB UICC that does not support remote wake-up. 'A0' corresponds to a USB UICC that supports remote wake-up.
bMaxPower		As the USB UICC is an Inter-Chip USB peripheral, this value is set to 4 or less. The maximum power consumption of the USB UICC from the bus when the device is fully operational is set in the Power negotiation procedure.

A USB UICCs may report configurations with multiple interfaces. The USB UICC-enabled terminal will select only configurations with interfaces that it supports.

A.3 The Standard Interface Descriptor

This descriptor is repeated for all the interfaces of the USB UICC, e.g. there may be one for APDU transfer [7], one for Ethernet Emulation [9], and another one for Mass Storage [7] and [6]. In addition, if alternate settings not specified in the present document are provided by the UICC, an interface descriptor may be repeated within a configuration.

The Standard interface descriptor for APDU transfer shall be as defined in the Smart Card ICCD specification [7].

For the purpose of the present document, the following fields shall be set as defined in table A.2.

Table A.2: Specific fields of the Standard interface descriptor for APDU transfer

Field	Value	Description
bNumEndpoints	'00' or '02'	'00' for the Smart Card Class with Control Transfer or '02' for the Smart Card class with Bulk-Only.
bInterfaceProtocol	'00' or '02'	'00' USB UICC messages using bulk. '02' USB UICC specific requests using control transfer Version B.

The Standard interface descriptor for Ethernet Emulation shall be as defined in CDC EEM [9].

The Standard interface descriptor for Mass storage shall be as defined in the Mass Storage Bulk Only 1.0 specification [6].

For the purpose of the present document, the following fields shall be set as defined in table A.3.

Table A.3: Specific fields of the Standard interface descriptor for Mass storage

Field	Value	Description
bNumEndpoints	'02'	'02' for Bulk-Only.
bInterfaceSubClass	'06'	'06' for SCSI Transparent subclass.

A USB UICCs may report additional alternate settings for the interfaces. The USB UICC-enabled terminal will select only the settings that it supports.

A.4 The Standard Endpoint Descriptors

This clause describes the endpoint descriptors that are used by the functional interfaces defined in the present document.

The Standard Endpoint Descriptors for bulk-IN and bulk-OUT shall be as defined in USB 2.0 [3].

For the purpose of the present document, the following fields shall be set as defined in table A.4.

Table A.4: Specific fields of the Endpoint descriptor for bulk-IN and bulk-OUT

Field	Value	Description
bInterval	'00'	Does not apply to bulk endpoints.

A.5 The Class Specific Descriptor

A.5.0 Introduction

This descriptor depends on the supported device class that it describes.

A.5.1 Class Descriptor for APDU transfer

The class-specific descriptor for APDU transfer shall be as defined in the Smart Card ICCD specification [7].

For the purpose of the present document , the following fields shall be set as defined in table A.5.

Table A.5: Specific fields of the Class specific descriptor for the Smart Card device class

Field	Value	Description
dwProtocols	'0000 0002'	Protocol T = 1 (APDU level exchange)
dwMaxIFSD	'0000 00FE'	For T = 1: '0000 00FE'
DwFeatures	'0002 0840' or '0004 0840'	'0002 0840' Short APDU level exchanges '0004 0840' Short and extended APDU level exchanges

A terminal compliant to the present document shall accept USB UICCs that indicate support of extended APDUs in *dwFeatures* and *dwMaxCCIDMessageLength*, even though ETSI TS 102 221 [1] currently only defines short APDUs.

NOTE: Because the Smart Card ICCD specification [7] re-uses a class descriptor already defined in a different specification, this descriptor contains references to T = 1, even though nothing of this protocol is used for the transfer of APDUs on the Smart Card functional interface.

A.5.2 Class Descriptor for Ethernet Emulation Model

The Ethernet Emulation Model subclass of the Communication device class does not use any class-specific descriptor.

A.5.3 Class Descriptor for Mass Storage

The Mass Storage Bulk Only class does not use any class-specific descriptor.

A.6 The UICC specific descriptor

This clause specifies the UICC specific descriptor.

Table A.6: Fields of the UICC specific descriptor

Offset	Field	Size	Value	Description
0	bFunctionLength	1	'13'	Size of the descriptor in bytes
1	bDescriptorType	1	Constant	'51'
2	bGUID[16]	16	Byte Constant	'E0 92 05 E6 B8 4F 41 CC AD 1F 0D 95 4C 3F 89 99'
18	bVersion	1	Byte Constant	Indicates the version and shall be set to '01' for USB UICCs according to the present document

Terminals designed to support the version of the UICC specific descriptor specified above shall ignore fields that may be added in later versions of this descriptor.

NOTE: The GUID was created as specified in IETF RFC 4122 [i.1] for a version 4 UUID.

Annex B (normative): Assigned values for vendor specific USB requests

The following *bRequest* values for vendor specific USB requests are assigned in the present document.

Table B.1: Assigned *bRequest* values

Value	Description
'01'	Get Interface Power Request
'02'	Set Interface Power Request
'03'	Resume Time Request
all other values	RFU

Annex C (informative): Bibliography

- Universal Serial Bus: "Mass Storage Class Specification Overview", Revision 1.2, USB Implementers Forum, Device Working Group: Mass Storage.

NOTE: Available at https://www.usb.org/documents.

Annex C (informative): Change history

The table below indicates all changes that have been incorporated into the present document since it was placed under change control.

Change history								
Date	**Meeting**	**Plenary Doc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**
2007-10	SCP-33	SCP-040421	001	-	D	Editorial and Reference correction	7.0.0	7.1.0
2008-01	SCP-35	SCP-080022	002	-	F	Clarification of power negotiation parameters to make it future proof	7.1.0	7.2.0
		SCP-080040	003	1	F	Clarification of resume timing	7.1.0	7.2.0
2008-04	SCP-37	SCP-080235	004	1	D	Clarification on power requirements to enable usage of High Density Memory	7.1.0	7.2.0
2008-07	SCP-38	SCP-080356	005	-	F	Inter-Chip USB interface - FCP handling Note: the CR number in SCP-080356 is incorrect and is changed to 005	7.2.0	7.3.0
2008-11	SCP-39	SCP-080430	007	-	F	Interworking of the functional interfaces	7.3.0	7.4.0
2008-11	SCP-39	SCP-080430	008	-	F	Inter-Chip USB interface - FCP handling	7.3.0	7.4.0
2009-01	SCP-40	SCP-090028	006	2	F	Clarification of the interface activation procedure.	7.4.0	7.5.0
2009-05	SCP-41	SCP-090136	009	4	F	Corrections on Power Negotiation procedure.	7.5.0	7.6.0
2009-10	SCP-43	SCP-090354	010	3	F	Correction on power requirements for applications	7.6.0	7.7.0
2010-07	SCP-45	SCP(10)0133	011	-	F	Minimum value for "bMaxCurrent" after power negotiation	7.7.0	7.8.0
2010-07	SCP-45	SCP(10)0133	013	-	F	Clarifications on support of Remote Wake Up	7.7.0	7.8.0
2010-07	SCP-45	SCP(10)0133	015	-	F	Clarification of remote wakeup and Resume signalling time	7.7.0	7.8.0
Rel-8, 9 and 10 of the specification created after SCP #45 based on v7.8.0								
2010-07	SCP-45	SCP(10)0133	014	-	D	Clarifications of inconsistencies	7.8.0	10.0.0
2010-07	SCP-45	SCP(10)0133	016	-	B	Remote wakeup time negotiation	7.8.0	10.0.0
2010-10	SCP-46	SCP(10)0274	025	-	A	After power negotiation, it is mandated that "the USB UICC shall ... adapt its USB configuration descriptors and interface descriptors". However, the descriptors are retrieved by the host during the attachment process and cannot be modified and resent dynamically by the device.	10.0.0	10.1.0

Change history								
Date	Meeting	Plenary Doc	CR	Rev	Cat	Subject/Comment	Old	New
2010-10	SCP-46	SCP(10)0275 r1	017	2	B	Addition of a UICC specific descriptor enabling the use of standard USB host software stacks. The sort of standard USB host software stacks we are referring to are currently used in many cell phones. To add USB UICC support to these stacks with minimum efforts, we need this descriptor. STF 391 proposed this descriptor to be able to use these stacks without requiring a list of product IDs and vendor ID for each UICC. Without this descriptor, the list included in every single handset will have to be updated every time a new USB UICC is deployed by the manufacturers (which is unreasonable). The mechanisms used in standard USB host software stacks (as per the device framework documented in section 9 of USB 2.0 core) to discover and enable a USB device are the same for all USB devices. No socket-specific commands are used in the initial process regardless of the physical connector type or the actual electrical signalling in use. The protocol handles discovery and initialisation using the sequence: power on, enable communication, send standard requests to the device at address 0 to identify the device, and then load appropriate device-specific software. The Inter-Chip supplement to USB defines no additional USB software requests and specifically requires no changes in USB system software as all hardware difference are handled without software intervention, enabling use of standard host stacks. In addition, there is an issue with sending unknown requests to USB devices in general. Testing conducted by USB-IF Compliance determined that more than half of all existing USB devices failed catastrophically if they received an unexpected or undefined (for them) command on the default pipe. The failure mode was device lock-up and required removing all power sources from the device before it would respond to USB again: unplug the USB cable, remove any external power connection, remove any internal batteries. To avoid this, the host software stack must explicitly know that the target of the USB UICC Power Negotiation is in fact a USB UICC. This is accomplished either through the ETSI UICC Descriptor or a list of product IDs and vendor ID for each UICC. The ETSI UICC Descriptor removes the logistic and cost issues of maintaining such product IDs and vendor ID lists on every Terminal using the USB UICC connectivity. At the same time, such a descriptor allows reuse of existing standard USB host software stacks.	10.0.0	10.1.0
2010-10	SCP-46	SCP(10)0279 r1	021	1	A	Correction of inconsistencies related to the ICCD specification	10.0.0	10.1.0
2010-10	SCP-46	SCP(10)0280	022	-	D	Clarification related to the ICCD specification	10.0.0	10.1.0
2010-10	SCP-46	SCP(10)0281	026	-	C	Clarification of handling of device descriptor	10.0.0	10.1.0

History

Document history		
V10.0.0	October 2010	Publication
V10.1.0	September 2020	Publication