

TCVN

TIÊU CHUẨN QUỐC GIA

**TCVN 11167-12:2015
ISO/IEC 7816-12:2005**

Xuất bản lần 1

**THẺ DANH ĐỊNH - THẺ MẠCH TÍCH HỢP -
PHẦN 12: THẺ TIẾP XÚC - THỦ TỤC VẬN HÀNH VÀ
GIAO DIỆN USB**

*Identification cards - Integrated circuit cards -
Part 12: Cards with contacts - USB electrical interface and operating procedures*

HÀ NỘI - 2015

Mục lục	Trang
Lời nói đầu	4
1 Phạm vi áp dụng	5
2 Tài liệu viện dẫn.....	6
3 Thuật ngữ và định nghĩa	6
4 Thuật ngữ viết tắt.....	6
5 Đặc tính điện của tiếp xúc	7
6 USB-ICC được vận hành bởi một thiết bị giao diện	7
7 Bộ mô tả USB	7
8 Truyền dữ liệu giữa máy chủ và USB-ICC	15
Phụ lục A (tham khảo) Kí hiệu đối với lược đồ trạng thái.....	39
Phụ lục B (tham khảo) Kịch bản đối với truyền USB.....	40
Phụ lục C (tham khảo) Thuật ngữ và định nghĩa trong đặc tả USB	50
Phụ lục D (tham khảo) Lớp thiết bị thẻ thông minh bộ mô tả cụ thể lớp	51
Thư mục tài liệu tham khảo	56

TCVN 11167-12:2015

Lời nói đầu

TCVN 11167-12:2015 hoàn toàn tương đương với ISO/IEC 7816-12:2005 và ISO/IEC 7816-12:2005/Cor.1:2007.

TCVN 11167-12:2015 do Tiểu Ban kỹ thuật tiêu chuẩn quốc gia TCVN/JTC 1/SC 17 “*Thẻ nhận dạng*” biên soạn, Tổng cục Tiêu chuẩn Đo lường Chất lượng đề nghị, Bộ Khoa học và Công nghệ công bố.

Bộ tiêu chuẩn TCVN 11167 (ISO/IEC 7816) *Thẻ định danh – Thẻ mạch tích hợp* gồm các tiêu chuẩn sau:

- Phần 1: Thẻ tiếp xúc - Đặc tính vật lý;
- Phần 2: Thẻ tiếp xúc - Kích thước và vị trí tiếp xúc;
- Phần 3: Thẻ tiếp xúc - Giao diện điện và giao thức truyền;
- Phần 4: Tổ chức, an ninh và lệnh trao đổi;
- Phần 5: Đăng ký của bên cung cấp ứng dụng;
- Phần 6: Phần tử dữ liệu liên ngành trong trao đổi;
- Phần 7: Lệnh liên ngành đối với ngôn ngữ truy vấn thẻ có cấu trúc;
- Phần 8: Lệnh đối với hoạt động an ninh;
- Phần 9: Lệnh đối với quản lý thẻ;
- Phần 10: Tín hiệu điện và trả lời để thiết lập lại cho thẻ đồng bộ;
- Phần 11: Xác minh cá nhân bằng phương pháp sinh trắc học;
- Phần 12: Thẻ tiếp xúc - Thủ tục vận hành và giao diện điện tử USB;
- Phần 13: Lệnh đối với quản lý ứng dụng trong môi trường đa ứng dụng;
- Phần 15: Ứng dụng thông tin mã hóa.

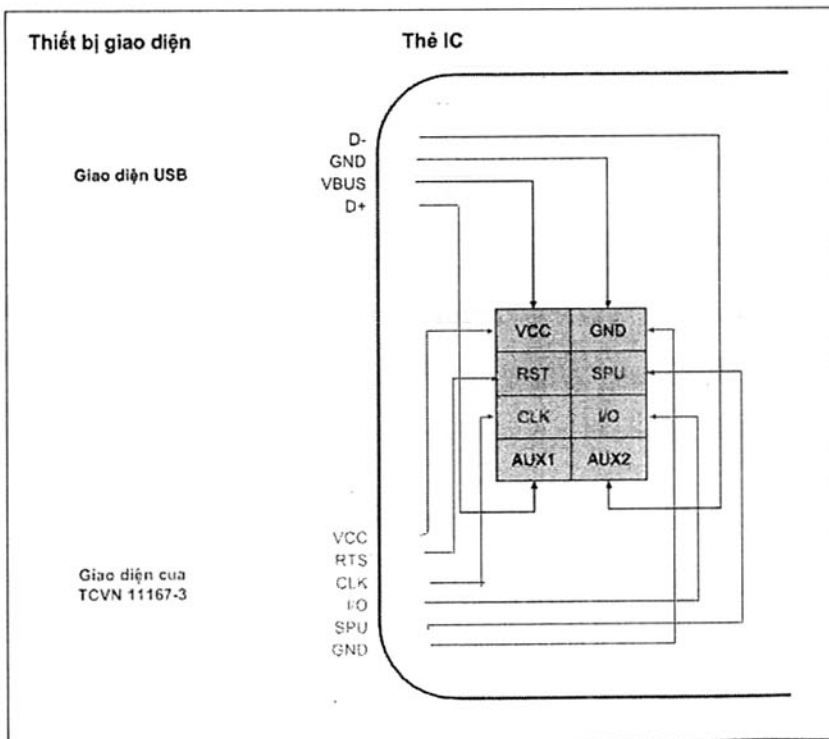
**Thẻ định danh - Thẻ mạch tích hợp -
Phần 12: Thẻ tiếp xúc - Thủ tục vận hành và giao diện điện USB**

Identification cards - Integrated circuit cards -

Part 12: Cards with contacts - USB electrical interface and operating procedures

1 Phạm vi áp dụng

Tiêu chuẩn này quy định điều kiện vận hành của thẻ mạch tích hợp nhằm cung cấp một giao diện USB. Hình 1 trình bày sự ấn định các trường tiếp xúc đối với một giao diện USB và - nhằm mô tả khả năng vận hành liên tục - sự ấn định được dùng trong TCVN 11167-3 (ISO/IEC 7816-3).



Hình 1 - Ấn định tiếp xúc đối với thẻ mạch tích hợp USB

TCVN 11167-12:2015

2 Tài liệu viện dẫn

Các tài liệu tham khảo dưới đây không thể thiếu đối với việc áp dụng tài liệu này. Đối với các tham khảo ghi năm, chỉ áp dụng bản được nêu. Đối với các tham khảo không ghi năm, bản mới nhất của tài liệu tham khảo (bao gồm cả sửa đổi) được áp dụng (nếu có).

ISO/IEC 7816-2:1999/Amd.1:2004, Identification cards - Integrated circuit cards - Part 2: Cards with contacts - Dimensions and location of the contacts - Amendment 1: Assignment of contacts C4 and C8.

TCVN 11167-3 (ISO/IEC 7816-3) Thẻ định danh - Thẻ mạch tích hợp - Phần 3: Tổ chức, an ninh và lệnh trao đổi,

Universal Serial Bus Specification Revision 2.0 (*Đặc tả USB phiên bản 2.0*)

Universal Serial Bus, Device Class Specification for USB Chip/Smart Card Interface Devices, Revision 1.00 (*USB, Đặc tả lớp thiết bị đối với thiết bị giao diện thẻ thông minh/chip USB*).

3 Thuật ngữ và định nghĩa

Tiêu chuẩn này áp dụng các thuật ngữ và định nghĩa sau.

3.1 Thiết bị

3.1.1

Thiết bị giao diện (interface device)

Thiết bị kết nối đầu cuối hay máy mà thẻ được kết nối điện trong suốt quá trình vận hành.

[TCVN 11167-3 (ISO/IEC 7816-3)]

3.1.2

Thiết bị kết nối USB (USB connection device)

Thiết bị tạo một đường kết nối điện giữa một USB-ICC và một máy chủ USB hay hub USB.

3.2 Thuật ngữ và định nghĩa dùng trong đặc tả khác

Tiêu chuẩn này áp dụng các thuật ngữ và định nghĩa tuân theo đặc tả USB và đặc tả CCID (xem Điều 4).

CHÚ THÍCH Các thuật ngữ liên quan dùng trong tiêu chuẩn này được liệt kê trong Phụ lục tham khảo C và D.

4 Thuật ngữ viết tắt

Tiêu chuẩn này áp dụng các thuật ngữ viết tắt sau.

Giao thức T = 0, Giao thức T = 1

[TCVN 11167-3 (ISO/IEC 7816-3)]

D+, D- [*Đặc tả USB, phiên bản 2.0*]

Đặc tả USB (USB specification)

Tham chiếu tới Đặc tả USB, phiên bản 2.0 (xem Điều 2)

CCID

Thiết bị giao diện thẻ Chip. Thiết kế một thiết bị giao diện được điều khiển qua USB.

Đặc tả CCID (CCID specification)

Tham chiếu đến đặc tả lớp thiết bị cho chip USB/thiết bị giao diện thẻ thông minh (xem Điều 2).

USB-ICC

Thẻ mạch tích hợp USB. Một thẻ mạch tích hợp cung cấp một giao diện USB.

5 Đặc tính điện của tiếp xúc

Sự ấn định tiếp xúc đối với các điều kiện vận hành USB được đưa ra trong TCVN 11167-2 (ISO/IEC 7816-2).

Thiết bị giao diện sẽ cung cấp một kết nối USB tới một USB-ICC thông qua VCC, GND, AUX1 và AUX2 tương ứng VBUS, GND, D+ và D- được quy định bởi đặc tả USB.

Thẻ được thiết kế theo các điều kiện vận hành của TCVN 11167-3 (ISO/IEC 7816-3) không bị hư hỏng khi được kích hoạt trong các điều kiện của USB. Ngược lại, thẻ được thiết kế cho các thao tác USB không bị hư hỏng khi kích hoạt các điều kiện vận hành của TCVN 11167-3 (ISO/IEC 7816-3) (theo định nghĩa, một thẻ bị hư hỏng không còn hoạt động theo quy định hoặc có chứa dữ liệu bị hỏng).

6 USB-ICC được vận hành bởi một thiết bị giao diện

Một USB-ICC chỉ cung cấp một giao diện USB phải được kết nối điện đến C1, C5, C4 và C8. Tất cả các vùng tiếp xúc khác phải được cách điện. Loại USB-ICC này có thể được vận hành bởi một thiết bị kết nối USB. Thiết bị kết nối USB phải thiết lập một kết nối điện đến C1, C5, C4 và chỉ đến C8, theo đặc tính điện và giao thức được đề cập trong đặc tả USB.

Một thiết bị giao diện không hỗ trợ một giao diện USB phải có AUX1 và AUX2 được cách điện hoặc đảm bảo điện áp được áp dụng ở các vùng tiếp xúc này phải duy trì trong khoảng -0,3V và Vcc + 0,3V.

7 Bộ mô tả USB**7.1 Bộ mô tả chuẩn**

Bộ mô tả chuẩn được mô tả trong đặc tả USB tạo thành một cách thức cho phần mềm máy chủ xác định một thiết bị USB mới gắn kèm và tải một hay nhiều trình điều khiển thích hợp cho thiết bị USB mới này. Bộ mô tả chuẩn được đọc bởi phần mềm máy chủ trong quá trình liệt kê. Ngoài ra, bộ mô tả cũng có thể được thu hồi bởi phần mềm máy chủ sử dụng các yêu cầu USB chuẩn.

TCVN 11167-12:2015

CHÚ THÍCH Tiêu chuẩn này sử dụng cho các giá trị thập lục phân, ký hiệu 'xxh' được sử dụng trong đặc tả USB. Điều này khác với ký hiệu 'xx' được sử dụng trong các phần khác của bộ tiêu chuẩn này. Ký hiệu 'xxh' được sử dụng ở đây để tránh nhầm lẫn khi đọc tiêu chuẩn này và các văn bản liên quan đến USB. Các bảng dưới đây của bộ mô tả chuẩn, dấu hoa thị (*) trong cột Giá trị chỉ ra rằng (các) giá trị này được quy định bởi ISO/IEC, lấy từ tập giá trị có thể được đưa nêu trong đặc tả USB. Tất cả các giá trị khác là các mục vào USB chuẩn.

Hướng truyền từ máy chủ đến USB-ICC được chỉ định là OUT. Hướng truyền từ USB-ICC tới máy chủ được chỉ định là IN.

7.1.1 Bộ mô tả thiết bị chuẩn

Bảng 1 - Bộ mô tả thiết bị chuẩn đối với một USB-ICC

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bLength</i>	1	12h	Kích cỡ của bộ mô tả theo các byte.
1	<i>bDescriptorType</i>	1	01h	Loại thiết bị DEVICE.
2	<i>bcdUSB</i>	2	0200h	Số phát hành quy định kỹ thuật USB.
4	<i>bDeviceClass</i>	1	00h*	Chỉ ra lớp thiết bị được quy định trong bộ mô tả giao diện của thiết bị.
5	<i>bDeviceSubClass</i>	1	00h	Thiết lập lại là 0 như <i>bDeviceClass</i> được thiết lập lại là 0.
6	<i>bDeviceProtocol</i>	1	00h*	Thiết bị không dùng các giao thức cụ thể-lớp theo thiết bị cơ bản. Thay vào đó thiết bị sử dụng các giao thức cụ thể-lớp trên mức giao diện.
7	<i>bMaxPacketSize0</i>	1		Kích cỡ gói lớn nhất đối với điểm cuối 0. Kích cỡ có thể là: 8, 16, 32, 64. Đối với các chức năng tốc độ thấp, giá trị phải là 8.
8	<i>idVendor</i>	2		ID của bên cung cấp (được gán bởi USB-IF)
10	<i>idProduct</i>	2		ID của sản phẩm (được gán bởi bên sản xuất) Quy định giá trị của trường này nằm ngoài phạm vi của tiêu chuẩn.
12	<i>bcdDevice</i>	2		Số phát hành thiết bị theo mã hóa nhị phân cơ số 10. Quy định giá trị của trường này nằm ngoài phạm vi của tiêu chuẩn.
14	<i>iManufacturer</i>	1		Chỉ mục của bộ mô tả chuỗi mô tả bên sản xuất. Quy định nội dung của chuỗi này nằm ngoài phạm vi của tiêu chuẩn.

Offset	Trường	Kích cỡ	Giá trị	Mô tả
15	<i>iProduct</i>	1		Chỉ mục của bộ mô tả chuỗi mô tả sản phẩm. Quy định nội dung của chuỗi này nằm ngoài phạm vi của tiêu chuẩn.
16	<i>iSerialNumber</i>	1		Chỉ mục của bộ mô tả chuỗi mô tả số se-ri của thiết bị
17	<i>bNumConfigurations</i>	1		Số các cấu hình có thể

7.1.2 Bộ mô tả cấu hình chuẩn

Bảng 2 - Bộ mô tả cấu hình chuẩn đối với một USB-ICC

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bLength</i>	1	09h	Kích cỡ của bộ mô tả theo các byte.
1	<i>bDescriptorType</i>	1	02h	Loại bộ mô tả CONFIGURATION.
2	<i>wTotalLength</i>	2		Tổng độ dài dữ liệu được trả về đối với cấu hình này, bao gồm độ dài kết hợp của tất cả các bộ mô tả (cấu hình, giao diện, điểm cuối và các cụ thể-lớp) được trả về theo cấu hình này.
4	<i>bNumInterfaces</i>	1		Số giao diện được hỗ trợ bởi cấu hình này.
5	<i>bConfigurationValue</i>	1		Giá trị được dùng như một phép gán với yêu cầu SetConfiguration() nhằm lựa chọn cấu hình này. Giá trị này phải khác 0.
6	<i>iConfiguration</i>	1		Chỉ mục của bộ mô tả chuỗi mô tả cấu hình này. Quy định nội dung của chuỗi này nằm ngoài phạm vi của tiêu chuẩn.
7	<i>bmAttributes</i>	1		Các đặc tính cấu hình đối với USB-ICC: Bit 4...0: Dành riêng (thiết lập lại là 0) Bit 5: Bật từ xa Bit 6: Tự cấp nguồn Bit 7: Dành riêng (đặt là 1) Đối với một USB-ICC được cấp nguồn bus không hỗ trợ bật từ xa, <i>bmAttributes</i> phải có giá trị 80h.
8	<i>MaxPower</i>	1		Bộ tiêu thụ nguồn tối đa của USB-ICC từ các bus khi thiết bị vận hành toàn phần. Nhấn mạnh trong các đơn vị 2mA

7.1.3 Bộ mô tả giao diện chuẩn

Bảng 3 - Bộ mô tả giao diện chuẩn đối với một USB-ICC

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bLength</i>	1	09h	Kích cỡ của bộ mô tả theo các byte.
1	<i>bDescriptorType</i>	1	04h	Loại bộ mô tả INTERFACE.
2	<i>bInterfaceNumber</i>	1		Số giao diện. Giá trị theo 0 xác định chỉ mục trong mảng các giao diện đồng thời được hỗ trợ bởi cấu hình này.
3	<i>bAlternateSetting</i>	1	00h*	Giá trị được dùng nhằm lựa chọn cài đặt thay thế đối với giao diện được định danh trong trường trước đó. Các thiết lập thay thế không được hỗ trợ.
4	<i>bNumEndpoints</i>	1	00h* 01h* 02h* 03h*	Số lượng các điểm cuối đối với một USB-ICC được dùng bởi giao diện này (không bao gồm điểm cuối 0) 00h không sử dụng các điểm cuối khác 01h sử dụng điểm ngắt IN 02h sử dụng bulk-IN và bulk-OUT 03h sử dụng bulk-IN, bulk-OUT và interrupts-IN CHÚ THÍCH 01h chỉ ra rằng các điểm cuối kiểm soát được dùng đối với việc truyền dữ liệu và interrupt-IN đối với việc thông báo các sự kiện đặc trưng thẻ từ USB-ICC tới máy chủ.
5	<i>bInterfaceClass</i>	1	0Bh FFh	Mã lớp đối với lớp thiết bị Thẻ thông minh (0Bh) hoặc nhóm giao diện là đặc trưng nhà cung cấp (Ffh) CHÚ THÍCH Một sản phẩm không sử dụng một trình điều khiển cụ thể lớp có thể phù hợp với tiêu chuẩn này. Trong trường hợp này, trình điều khiển được chọn sử dụng thông tin đưa ra bởi nhà cung cấp, nhà sản xuất và ID sản phẩm (xem Bảng 1)
6	<i>bInterfaceSubClass</i>	1	00h	Mã lớp phụ

Offset	Trường	Kích cỡ	Giá trị	Mô tả
7	<i>InterfaceProtocol</i>	1	00h 01h 02h	<p>Mã giao thức.</p> <p>Lớp thiết bị thẻ thông minh cung cấp các giao thức giao diện sau đối với một USB-ICC:</p> <ul style="list-style-type: none"> - 00h Các thông điệp USB-ICC dùng số lượng lớn (tùy chọn ngắt) - 01h Các yêu cầu cụ thể USB-ICC dùng kiểm soát truyền phiên bản A (không ngắt) - 02h Các yêu cầu cụ thể USB-ICC dùng kiểm soát truyền phiên bản B (ngắt tùy chọn) <p>Giá trị đưa ra chỉ ra chế độ truyền được dùng đối với kết nối giữa host và USB-ICC.</p>
8	<i>Interface</i>	1		<p>Chỉ mục của bộ mô tả chuỗi mô tả giao diện.</p> <p>Quy định nội dung của chuỗi này nằm ngoài phạm vi của tiêu chuẩn.</p>

7.1.4 Bộ mô tả điểm cuối chuẩn

Một USB-ICC có thể giao tiếp với máy chủ bằng cách chỉ sử dụng ống điều khiển mặc định hay qua ống thông điệp bằng cách sử dụng bulk-IN và bulk-OUT. Tùy chọn, một USB-ICC có thể cung cấp một điểm cuối interrupt-IN cho phép USB-ICC để chỉ ra các sự kiện cụ thể cho máy chủ. Một USB-ICC có thể có một trong các cấu hình sau:

Bảng 4 - Cấu hình của điểm cuối đối với một USB-ICC

Điểm cuối cho việc truyền dữ liệu	Dùng bộ truyền điều khiển		Dùng bộ truyền bulk
	Phiên bản A	Phiên bản B	
Ống điều khiển mặc định	có	có	có
Bulk-IN	không	không	có
Bulk-OUT	không	không	có
Interrupt-IN	không	tùy chọn	tùy chọn

Các bảng sau đây mô tả bộ mô tả thiết bị đầu cuối:

Bảng 5 - Bộ mô tả điểm cuối bulk-OUT

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bLength</i>	1	07h	Kích cỡ của bộ mô tả theo các byte.
1	<i>bDescriptorType</i>	1	05h	Loại bộ mô tả ENDPOINT.
2	<i>bEndpointAddress</i>	1	01-0Fh	Địa chỉ của điểm cuối này trên USB-ICC. Địa chỉ là một số lượng điểm cuối nằm giữa 1 và 15 Bit 3...0 Số lượng điểm cuối Bit 6...4 Dành riêng, phải bằng 0 Bit 7 0 = OUT
3	<i>bNumInterfaces</i>	1	02h	Đây là một điểm cuối số lượng lớn
4	<i>wMaxPacketSize</i>	1	00xxh	Kích cỡ truyền dữ liệu tối đa. Có thể là: 8, 16, 32, 64.
6	<i>bInterval</i>	1	00h	Không áp dụng cho các điểm cuối số lượng lớn

Bảng 6 - Bộ mô tả điểm cuối bulk-IN

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bLength</i>	1	07h	Kích cỡ của bộ mô tả theo các byte.
1	<i>bDescriptorType</i>	1	05h	Loại bộ mô tả ENDPOINT.
2	<i>bEndpointAddress</i>	1	81-8Fh	Địa chỉ của điểm cuối này trên USB-ICC. Địa chỉ là một số lượng điểm cuối nằm giữa 1 và 15 Bit 3...0 Số lượng điểm cuối Bit 6...4 Dành riêng, phải bằng 0 Bit 7 1 = IN
3	<i>bNumInterfaces</i>	1	02h	Đây là một điểm cuối số lượng lớn
4	<i>wMaxPacketSize</i>	2	00xxh	Kích cỡ truyền dữ liệu tối đa. Có thể là: 8, 16, 32, 64.
6	<i>bInterval</i>	1	00h	Không áp dụng cho các điểm cuối số lượng lớn

Bảng 7 - Bộ mô tả điểm cuối interrupt-IN

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bLength</i>	1	07h	Kích cỡ của bộ mô tả theo các byte.
1	<i>bDescriptorType</i>	1	05h	Loại bộ mô tả ENDPOINT.
2	<i>bEndpointAddress</i>	1	81-8Fh	Địa chỉ của điểm cuối này trên USB-ICC. Địa chỉ là một số lượng điểm cuối nằm giữa 1 và 15 Bit 3...0 Số lượng điểm cuối Bit 6...4 Dành riêng, phải bằng 0 Bit 7 1 = IN
3	<i>bmAttributes</i>	1	03h	Đây là một điểm cuối ngắt.
4	<i>wMaxPacketSize</i>	1	00xyh	Kích cỡ gói đối với USB-ICC. Giá trị tối đa phải là 02h.
6	<i>bInterval</i>	1	xyh	Khoảng thời gian đối với việc chọn các bộ truyền dữ liệu điểm cuối. Tính theo mili-giây. Giá trị phải nằm trong dải từ 1 tới 255 nhằm giữ băng thông, giá trị đề xuất là 255.

7.2 Bộ mô tả cụ thể lớp

Lớp thiết bị thẻ thông minh sử dụng bộ mô tả cụ thể lớp được mô tả trong đặc tả CCID (xem Phụ lục D). Trong ngữ cảnh của một thiết bị giao diện thẻ chip, một USB-ICC thể hiện cho một cấu hình của một thiết bị giao diện có khe cắm duy nhất với một thẻ được chèn vĩnh viễn. Các giá trị có thể cho bộ mô tả cụ thể lớp thể hiện cấu hình thiết bị này. Các trường chứa *bReserved* hay *dwReserved* có các thông số mà không liên quan tới một USB-ICC. Mặc dù không liên quan, nó bắt buộc một USB-ICC sử dụng chính xác các giá trị này cho *bReserved* và *dwReserved* nhằm duy trì khả năng tương thích với đặc tả CCID.

Bảng 8 - Bộ mô tả cụ thể lớp đối với một USB-ICC

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bLength</i>	1	36h	Kích cỡ của bộ mô tả theo các byte.
1	<i>bDescriptorType</i>	1	21hICC	Loại Thiết bị Chứng năng CCID.
2	<i>bcdCCID</i>	2		Số phát hành đặc tả CCID theo hệ nhị phân mã hóa thập phân. Phiên bản hiện tại 1.0 là 0100h. Số phát hành đặc tả CCID 1.0 được cập nhật bởi thẻ thông minh USB-DWG
4	<i>bMaxSlotIndex</i>	1	00h	Chỉ mục của số khe sẵn có lớn nhất. Một USB-ICC được xem như một khe đơn lẻ.
5	<i>bReserved</i>	1	01h	Giá trị này phải là 01h.
6	<i>dwProtocols</i>	4	0000 0001h 00000002h 0000 0002h	Chỉ ra các loại giao thức được hỗ trợ: 00000001h = Giao thức T=0 00000002h = Giao thức T=1 CHÚ THÍCH USB-IVV hỗ trợ các trao đổi mức APDU đối với Y=1 hoặc các trao đổi mức .. đối với T=0. Các kết hợp khác của <i>dwProtocols</i> và <i>dwFeatures</i> không được hỗ trợ bởi USB-ICC. USB-ICC áp dụng đối với chế độ truyền bulk và đối với chế độ truyền điều khiển.
10	<i>dwReserved</i>	4	0000 0DFCh	Giá trị này phải là 0000 0DFCh.
14	<i>dwReserved</i>	4	0000 0DFCh	Giá trị này phải là 0000 0DFCh.
18	<i>bReserved</i>	1	00h	Giá trị này phải là 00h.
19	<i>dwReserved</i>	4	0000 2580h	Giá trị này phải là 0000 2580h.
23	<i>dwReserved</i>	4	0000 2580h	Giá trị này phải là 0000 2580h.
27	<i>bReserved</i>	1	00h	Giá trị này phải là 00h.
28	<i>dwMaxIFSD</i>	4		Chỉ ra IFSD lớn nhất được hỗ trợ bởi USB-ICC đối với giao thức T=1. Với T=0 bất kì giá trị nào có thể đưa ra. Với T=1: 00000FEh. Với T=0: bất kì giá trị nào.

Offset	Trường	Kích cỡ	Giá trị	Mô tả
32	<i>dwReserved</i>	4	0000 0000h	Giá trị này phải là 0000 0000h.
36	<i>dwMechanical</i>	4	0000 0000h	Chỉ ra rằng một USB-ICCC không có các đặc tính đặc biệt.
40	<i>dwFeatures</i>	4	0000 0840h 0002 0840h 0004 0840h	Giá trị của từ viết thường (-040) chỉ ra rằng host chỉ gửi các yêu cầu hiệu lực đối với USB-ICC. Giá trị của từ viết hoa là mức trao đổi dữ liệu với USB-ICC: 0000h Các trao đổi mức ký tự 0002h Các trao đổi mức APDU ngắn 0004h Các trao đổi mức APDU ngắn và APDU mở rộng CHÚ THÍCH cũng xem <i>dwProtocols</i> .
44	<i>dwMaxCCIDMessageLength</i>	4		Đối với các truyền bulk, giá trị phải nằm giữa: 261 + 10 và 65544 + 10 CHÚ THÍCH Giá trị 10 là kích cỡ của tiêu đề. Đối với các truyền điều khiển, giá trị phải nằm giữa: 261 và 65544.
48	<i>bReserved</i>	1	FFh	Giá trị này phải là FFh.
49	<i>bReserved</i>	1	FFh	Giá trị này phải là FFh.
50	<i>wRFU</i>	2	0000h	Tất cả các giá trị khác được bảo lưu với sử dụng dự kiến.
52	<i>bRFU</i>	1	00h	Tất cả các giá trị khác được bảo lưu với sử dụng dự kiến.
53	<i>bMaxCCIDBusySlots</i>	1	01h	USB-ICC được xem như một khe đơn lẻ.

8 Truyền dữ liệu giữa máy chủ và USB-ICC

Việc trao đổi dữ liệu giữa máy chủ và USB-ICC có thể được thực hiện bằng cách sử dụng truyền bulk hoặc các truyền kiểm soát. Đối với truyền kiểm soát, có hai cách triển khai khả thi. Chúng được đặt tên phiên bản A và B. Chế độ truyền bulk phù hợp với đặc tả CCID, ví dụ: nó sử dụng một tập con các thông điệp/yêu cầu theo quy định tại quy định này.

Các ký hiệu cho các lược đồ trạng thái được đưa ra trong Phụ lục A.

8.1 Truyền bulk

Nhằm truyền lệnh, hồi đáp và đáp ứng dữ liệu giữa máy chủ và USB-ICC, các thông điệp sau phải được áp dụng:

Bảng 9 - Thông điệp Bulk-IN và Bulk-OUT

Tên thông điệp Bulk-OUT	Tên thông điệp hồi đáp Bulk-IN	Mô tả
PC_to_RDR_IccPowerOn	RDR_to_PC_DataBlock	Tồn tại trạng thái ban đầu của một USB-ICC và trả lại ATR đặt lại trạng thái lạnh trong thông điệp hồi đáp.
PC_to_RDR_IccPowerOff	RDR_to_PC_SlotStatus	Đặt SUB-ICC với các điều kiện ban đầu.
PC_to_RDR_XfrBlock	RDR_to_PC_DataBlock	Thông điệp có việc truyền dữ liệu giữa máy chủ và USB-ICC.

8.1.1 Thông điệp bulk

Tất cả các thông điệp truyền qua các điểm cuối số lượng lớn bắt đầu với một tiêu đề 10 byte, tùy theo dữ liệu.

Mục đích của tiêu đề là trao đổi kiểm soát và thông tin trạng thái giữa máy chủ và USB-ICC. Ngoài ra, chuỗi đánh số gán các thông điệp lệnh với thông điệp hồi đáp tương ứng của chúng. USB-ICC trả về trạng thái và thông tin lỗi của nó trong các trường *bStatus* và *bError*.

8.1.1.1 PC_to_RDR_IccPowerOn và RDR_to_PC_DataBlock

Bảng 10 - Thông điệp PC_to_RDR_IccPowerOn

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bMessageType</i>	1	62h	Chỉ ra PC_to_RDR_IccPowerOn.
1	<i>dwLength</i>	4	00000000h	Không có các bit bổ sung của thông điệp này.
5	<i>bSlot</i>	1	00h	Số khe đối với USB-ICC.
5	<i>bSeq</i>	1	00h - FFh	Chuỗi số đối với lệnh.
7	<i>bReserved</i>	1	01h	Giá trị này phải là 01h.
8	<i>abRFU</i>	2	0000h	Tất cả các giá trị khác được dành riêng để sử dụng sau này.

Hồi đáp cho thông điệp này là thông điệp RDR_to_PC_DataBlock.

Bảng 11 - Thông điệp RDR_to_PC_DataBlock chứa ATR

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bMessageType</i>	1	80h	Chỉ ra RDR_to_PC_DataBlock.
1	<i>dwLength</i>	4		Kích cỡ các byte đối với ATR.
5	<i>bSlot</i>	1	00h	Số khe đối với USB-ICC.
5	<i>bSeq</i>	1	Cùng giá trị với thông điệp Bulk-OUT tương ứng	Chuỗi số đối với thông điệp lệnh tương ứng.
7	<i>bStatus</i>	1		Thông tin trạng thái USB-ICC.
8	<i>bError</i>	1		Mã lỗi trong trường hợp sai sót.
9	<i>bChainParameter</i>	1	00h	Chỉ ra rằng thông điệp này chứa ATR hoàn chỉnh.
10	<i>abData</i>	1		ATR

8.1.1.2 PC_to_RDR_IccPowerOff và RDR_to_PC_SlotStatus

Bảng 12 - Thông điệp PC_to_RDR_IccPowerOff

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bMessageType</i>	1	62h	Chỉ ra PC_to_RDR_IccPowerOff.
1	<i>dwLength</i>	4	00000000h	Không có các bit bổ sung của thông điệp này.
5	<i>bSlot</i>	1	00h	Số khe đối với USB-ICC.
6	<i>bSeq</i>	1	00h - FFh	Chuỗi số đối với lệnh.
7	<i>abRFU</i>	3	0000000h	Tất cả các giá trị khác được dành riêng để sử dụng sau này.

Hỏi đáp của thông điệp này là thông điệp RDR_to_PC_SlotStatus.

Bảng 12 - Thông điệp RDR_to_PC_SlotStatus

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bMessageType</i>	1	81h	Chỉ ra RDR_to_PC_SlotStatus.
1	<i>dwLength</i>	4	00000000h	Không có các bit bổ sung của thông điệp này.
5	<i>bSlot</i>	1	00h	Số khe đối với USB-ICC.
6	<i>bSeq</i>	1	Cùng giá trị với thông điệp Bulk-OUT tương ứng	Chuỗi số đối với thông điệp lệnh tương ứng.
7	<i>bStatus</i>	1		Thông tin trạng thái USB-ICC.
8	<i>bError</i>	1		Mã lỗi trong trường hợp sai sót.
9	<i>bReserved</i>	1	00h	Giá trị này phải là 00h.

8.1.1.3 PC_to_RDR_XfrBlock và RDR_to_PC_DataBlock

Lệnh PC_to_RDR_XfrBlock được sử dụng để truyền các lệnh APDU.

Bảng 14 - Thông điệp PC_to_RDR_XfrBlock

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bMessageType</i>	1	6Fh	Chỉ ra thông điệp PC_to_RDR_XfrBlock.
1	<i>dwLength</i>			Kích cỡ của trường <i>abData</i> của thông điệp này.
5	<i>bSlot</i>	1	00h	Số khe đối với USB-ICC.
6	<i>bSeq</i>	1	00h - FFh	Chuỗi số đối với thông điệp lệnh tương ứng.
7	<i>bReserved</i>	1	00h	Phải được đặt là 00h.
8	<i>wLevelParameter</i>	2		Dựa trên mức trao đổi được báo cáo bởi bộ mô tả cụ thể lớp trong trường <i>dwFeatures</i> : <ul style="list-style-type: none"> - Mức ký tự: kích cỡ của dữ liệu được mong đợi được trả về bởi điểm cuối bulk-IN, - Mức APDU ngắn: 00h - Mức APDU mở rộng: chỉ ra nếu APDU bắt đầu hoặc kết thúc tring lệnh này: <ul style="list-style-type: none"> - 0000h: lệnh APDU bắt đầu và kết thúc với lệnh này, - 0001h: lệnh APDU bắt đầu với lệnh này, và tiếp tục trong PC_to_RDR_XfrBlock kế tiếp, - 0002h: trường <i>abData</i> này tiếp tục một lệnh APDU và kết thúc lệnh APDU,

Offset	Trường	Kích cỡ	Giá trị	Mô tả
				<ul style="list-style-type: none"> - 0003h: trường adData tiếp tục một lệnh APDU và khối khác theo nó, - 0010h: trường abData trống, APDU hồi đáp kế tiếp được mong đợi trong RDR_to_PC_DataBlock kế tiếp.
10	abData			Khối dữ liệu được gửi tới từ host tới USB-ICC.

Hồi đáp của thông điệp này là thông điệp RDR_to_PC_DataBlock.

Bảng 15 - Thông điệp RDR_to_PC_DataBlock chứa một khối dữ liệu

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	bMessageType	1	80h	Chỉ ra RDR_to_PC_DataBlock.
1	dwLength	4		Kích cỡ của các byte đối với khối dữ liệu nhận được.
5	bSlot	1	00h	Số khe đối với USB-ICC.
6	bSeq	1	Cùng giá trị với thông điệp Bulk-OUT tương ứng	Chuỗi số đối với thông điệp lệnh tương ứng.
7	bStatus	1		Thông tin trạng thái USB-ICC.
8	bError	1		Mã lỗi trong trường hợp sai sót.
9	bChainParameter	1		<p>Dựa trên mức trao đổi được báo cáo bởi bộ mô tả cụ thể lớp trong trường dwFeatures:</p> <ul style="list-style-type: none"> - Mức ký tự: 00h, - Mức APDU ngắn: 00h - Mức APDU mở rộng: chỉ ra nếu hồi đáp hoàn thiện, được tiếp tục hoặc nếu lệnh APDU có thể tiếp tục: <ul style="list-style-type: none"> • 00h: APDU hồi đáp bắt đầu và kết thúc với lệnh này, • 01h: APDU hồi đáp bắt đầu với lệnh này và tiếp tục, • 02h: trường abData này tiếp tục APDU hồi đáp và kết thúc APDU hồi đáp, • 03h: trường adData tiếp tục APDU hồi đáp và khối khác theo nó,

TCVN 11167-12:2015

Offset	Trường	Kích cỡ	Giá trị	Mô tả
				<ul style="list-style-type: none">0010h: trường abData trống, lệnh APDU kế tiếp được mong đợi trong PC_to_RDR_XfrBlock kế tiếp.
10	abData			Khởi dữ liệu được gửi tới USB-ICC tới host.

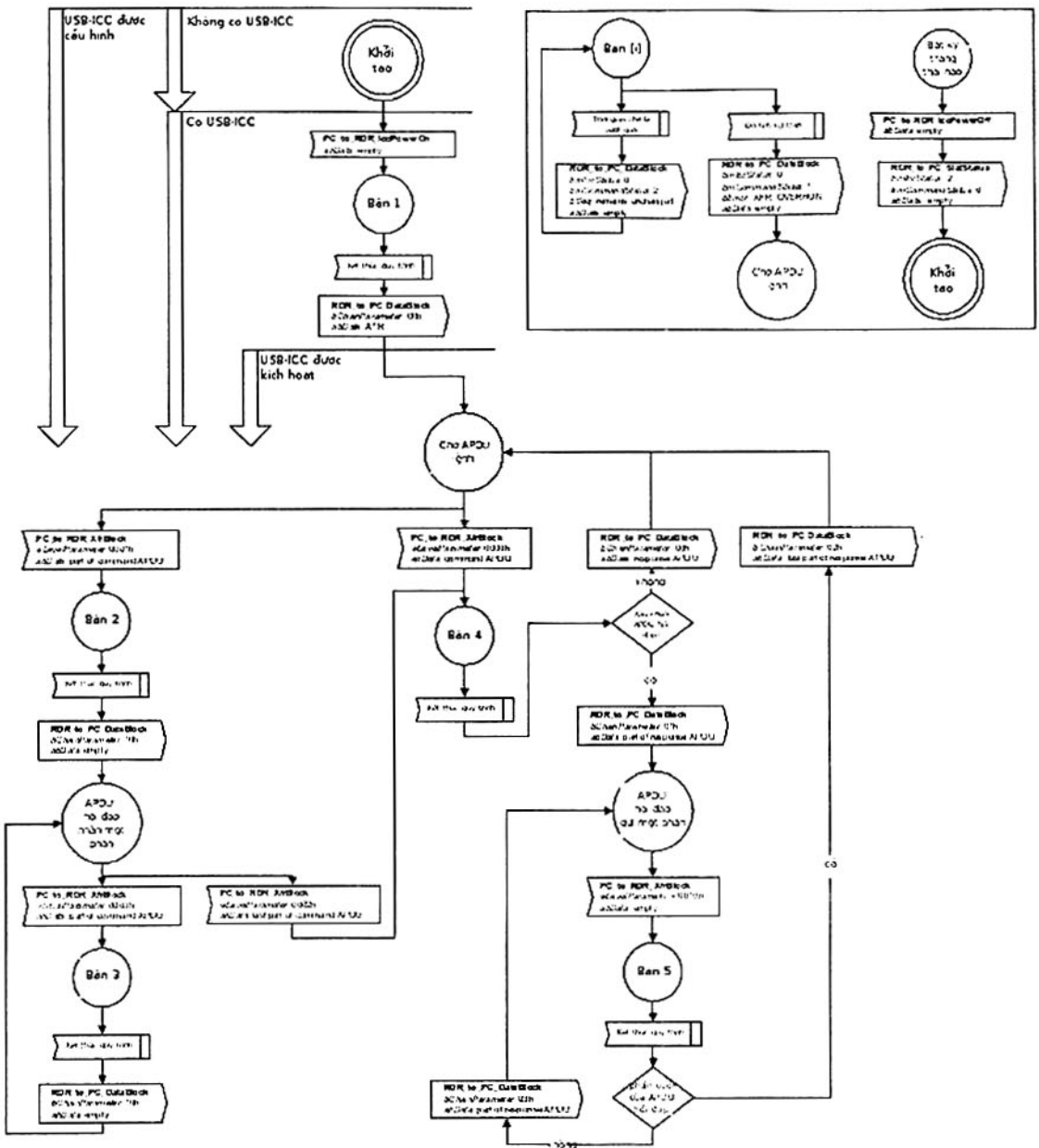
8.1.2 ATR và truyền dữ liệu

Khi một thiết bị USB được gắn vào kênh truyền và sau đó đã đạt được một trạng thái mà máy chủ có thể sử dụng các chức năng được cung cấp bởi thiết bị, thiết bị này được thiết kế là "đã cấu hình". Các thông điệp được truyền theo thứ tự để đặt USB-ICC ở trạng thái ban đầu, nhằm có được ATR và để truyền dữ liệu đã cho theo lược đồ trạng thái ở Hình 2. Việc truyền sử dụng các trao đổi mức APDU. Hình 2 bao gồm việc truyền các APDU ngắn và APDU mở rộng.

Với việc truyền chính xác dữ liệu, các quy tắc chung sau đây phải được áp dụng:

- Nếu USB-ICC nhận một PC_to_RDR_PowerOn khi nó không ở trong trạng thái "ban đầu", USB-ICC phải hồi đáp với một STALL. USB-ICC phải duy trì ở trạng thái hiện tại của nó.
- Nếu USB-ICC yêu cầu một giới hạn thời gian (xem Bảng 16), giá trị của bSeq (xem Điều 8.1.1) phải không thay đổi.
- Nếu USB-ICC trả về RDR_to_PC_DataBlock chỉ ra các lỗi ICC_MUTE hay HW_ERROR, máy chủ cần gửi một thông điệp PC_to_RDR_IccPowerOff.

QUAN TRỌNG - Trạng thái của việc thực thi hiện tại phải không ảnh hưởng bởi trạng thái của nguồn giao diện USB. Ví dụ: một bộ liệt kê kênh bus không gây ra bất kỳ chuyển đổi nào.



dwFeatures: 0002 081Ch (APDU ngắn) và
 0004 084Ch (APDU mở rộng)
 dwProtocols: 0000 0002Ch (T=1)

Hình 2 - Lược đồ trạng thái của USB-ICC dùng truyền bulk, truyền mức APDU cho APDU ngắn và APDU mở rộng

TCVN 11167-12:2015

8.1.3 Trạng thái và điều kiện lỗi

Thông điệp bulk-IN RDR_to_PC_SlotStatus và RDR_to_PC_DataBlock chứa thông tin trạng thái về USB-ICC và nếu các lệnh xử lý thành công. Trong trường hợp thất bại, một mã lỗi sẽ được trả lại.

Trường *bStatus* bao gồm hai trường bản đồ bit chứa thông tin về trạng thái USB-ICC (*bmICCStatus*) và lệnh xử lý (*bmCommandStatus*). Hai bảng dưới đây đưa ra các giá trị về tình trạng và các mã lỗi.

Bảng 16 - Bản đồ bit đối với trường bStatus

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bmIccStatus</i>	1 (2 bit)	0, 1, 2	0 = USB-ICC có và được kích hoạt. 1 = USB-ICC có nhưng không được kích hoạt 2 = USB-ICC không có 3 = RFU
(2 bit)		(4 bit)		RFU
(6 bit)	<i>bmCommandStatus</i>	(2 bit)	0, 1, 2	0 = Được xử lý không có lỗi 1 = Có sai sót, điều kiện lỗi được đưa ra bởi bError. 2 = Mở rộng thời gian được yêu cầu 3 = RFU
1	<i>bError</i>	1		Mã lỗi

Bảng 17 - Mã lỗi đối với bError

Tên lỗi	Mã lỗi	Nguyên nhân có thể
ICC_MUTE	-2 (FEh)	Các ứng dụng của USB-ICC không đáp ứng hoặc ATR có thể không được gửi bởi USB-ICC.
XFR_OVERRUN	-4 (FCh)	USB-ICC dò tìm lỗi tràn Offset khi nhận một khối dữ liệu.
HW_ERROR	-5 (FBh)	USB-ICC được dò tìm một lỗi phần cứng.
	-64 tới -127 (C0h – 81h)	Được người dùng định nghĩa.
	-3 (FDh) -8 tới -14 (F8h – F2h) -16 (F0h) -17 (EFh) -32 (E0h)	Các giá trị này phải không được dùng bởi USB-ICC.
	Tất cả các mã khác (80h và các giá trị được điền vào khoảng trống)	Được dành riêng cho sử dụng sau này.

Đối với việc sử dụng mã lỗi, các quy tắc sau được áp dụng:

- nếu giá trị của *bmCommandStatus* bằng 0 hoặc RFU, giá trị của *bError* bằng 0.
- nếu giá trị của *bmCommandStatus* bằng 1, giá trị của *bError* là:
 - mã lỗi = điều kiện lỗi như mô tả trong Bảng 17.
 - offset = nếu USB-ICC không thể phân tích một trường trong tiêu đề (10 byte) hoặc không hỗ trợ một trong những trường này, thì *bError* chứa Offset của giá trị xấu đầu tiên là một số dương (ví dụ: nếu máy chủ đặt *bSlot* là 01h, USB-ICC sẽ trả lại *bError* = 05h). Một USB-ICC nhận một lệnh không được hỗ trợ, phải đặt giá trị Offset bằng 0.

8.2 Truyền điều khiển

Chế độ truyền này có thể sử dụng cho USB-ICC cung cấp các chức năng tốc độ thấp. Ông điều khiển mặc định được dùng trao đổi dữ liệu giữa máy chủ và USB-ICC.

Đoạn này quy định các yêu cầu cụ thể cho truyền điều khiển. Các yêu cầu này cung cấp các dịch vụ tương tự cho lớp ứng dụng như đối với truyền bulk.

Có hai các thiết lập cho truyền điều khiển, đặt tên là: phiên bản A và B.

TCVN 11167-12:2015

8.2.1 Phiên bản A

8.2.1.1 Yêu cầu cụ thể

Bảng sau xác định các giá trị hợp lệ của bRequest:

Bảng 18 - Yêu cầu cụ thể lớp, phiên bản A

bRequest	Giá trị	Chiều mức dữ liệu	Mô tả
ICC_POWER_ON	62h	IN	Tồn tại trạng thái ban đầu của một USB-ICC. Trả lại ATR tái lập lạnh trong giai đoạn dữ liệu.
ICC_POWER_OFF	63h	OUT	Đặt USB-ICC với các điều kiện ban đầu.
XFR_BLOCK	65h	OUT	Truyền dữ liệu từ host tới USB-ICC.
DATA_BLOCK	6Fh	IN	Truyền dữ liệu từ USB-ICC tới máy chủ.
GET_ICC_STATUS	A0h	IN	Trả lại trạng thái của xử lý lệnh.

8.2.1.2 Giai đoạn Thiết lập

Giai đoạn thiết lập chứa các yêu cầu cụ thể lớp và các thông số tương ứng. Bảng dưới đây cung cấp các giá trị và thông số cho từng yêu cầu cụ thể lớp và mô tả dữ liệu được truyền giữa máy chủ và USB-ICC.

Đối với các thông số, các quy tắc chung sau đây phải được áp dụng:

- Giá trị của bInterface là cùng một giá trị như bInterfaceNumber được đưa ra trong Bảng 3.
- Giá trị tham số dành riêng cho các yêu cầu cụ thể lớp được sử dụng trong các trường: wValue và Windex được chỉ định là bRFU và wRFU. Giá trị của bRFU phải được đặt là 00h và giá trị của wRFU được đặt là 0000H.
- Nếu USB-ICC nhận được một yêu cầu không hợp lệ hay nếu một yêu cầu hợp lệ chứa một giá trị thông số không hợp lệ (wValue, Windex, wLength), USB-ICC phải hồi đáp với một STALL.
- Trên một yêu cầu đầu vào, các USB-ICC phải không trả về nhiều dữ liệu hơn được chỉ định bởi giá trị wLength. Nó có thể trả lại ít hơn. Trên một yêu cầu đầu ra, wLength phải luôn chỉ ra được chính xác lượng dữ liệu được gửi bởi máy chủ. Các USB-ICC phải trả về một STALL nếu máy chủ không cần gửi lượng dữ liệu nhiều hơn được quy định tại wLength.
- Đối ICC_POWER_OFF và GET_ICC_STATUS, máy chủ phải gửi các giá trị cho wLength như được quy định trong các bảng tương ứng. Nếu không, các USB-ICC sẽ hồi đáp STALL.

Bảng 19 - Yêu cầu ICC_POWER_ON, phiên bản A

bmRequestType	bRequest	wValue	wIndex	wLength	Dữ liệu
10100001B	ICC_POWER_ON	wRFU	bRFU <i>bInterface</i>	Độ dài của ATR	ATR tái lập lạnh

Trường Windex chỉ định bRFU trong byte cao và *bInterface* trong byte thấp.

Bảng 20 - Yêu cầu ICC_POWER_OFF, phiên bản A

bmRequestType	bRequest	wValue	wIndex	wLength	Dữ liệu
10100001B	ICC_POWER_OFF	wRFU	bRFU <i>bInterface</i>	0000h	Trống

Trường Windex chỉ định bRFU trong byte cao và *bInterface* trong byte thấp.

Bảng 21 - Yêu cầu XFR_BLOCK, phiên bản A

bmRequestType	bRequest	wValue	wIndex	wLength	Dữ liệu
10100001B	XFR_BLOCK	<i>bLevelParameter</i> bRFU	bRFU <i>bInterface</i>	Độ dài của dữ liệu	Lệnh APDU

Trường Windex chỉ định bRFU trong byte cao và *bInterface* trong byte thấp. Trường wValue chỉ định *bLevelParameter* trong byte cao và bRFU trong byte thấp.

Bảng 22 - Yêu cầu DATA_BLOCK, phiên bản A

bmRequestType	bRequest	wValue	wIndex	wLength	Dữ liệu
10100001B	DATA_BLOCK	wRFU	bRFU <i>bInterface</i>	Độ dài của dữ liệu. Giá trị phải là cả 2, L_e hoặc $dwMaxCCIDMessageLength$ phụ thuộc vào StatusByte trước đó	APDU hồi đáp

Trường Windex chỉ định bRFU trong byte cao và *bInterface* trong byte thấp.

Bảng 23 - Yêu cầu GET_ICC_STATUS ICC

bmRequestType	bRequest	wValue	wIndex	wLength	Dữ liệu
10100001B	GET_ICC_STATUS	wRFU	bRFU <i>bInterface</i>	0001h	Chọn trạng thái của USB-

Trường Windex chỉ định bRFU trong byte cao và *bInterface* trong byte thấp.

CHÚ THÍCH Một sản phẩm không sử dụng một trình điều khiển cụ thể lớp có thể là tuân thủ TCVN 11167-12 (ISO/IEC 7816-12). Trong trường hợp này, việc mã hóa của các bit 5,6 của bmRequestType thay đổi từ 01B đến 10B; ví dụ: bmRequestType cho GET_ICC_STATUS sẽ là 11000001B

8.2.1.3 ATR và truyền dữ liệu

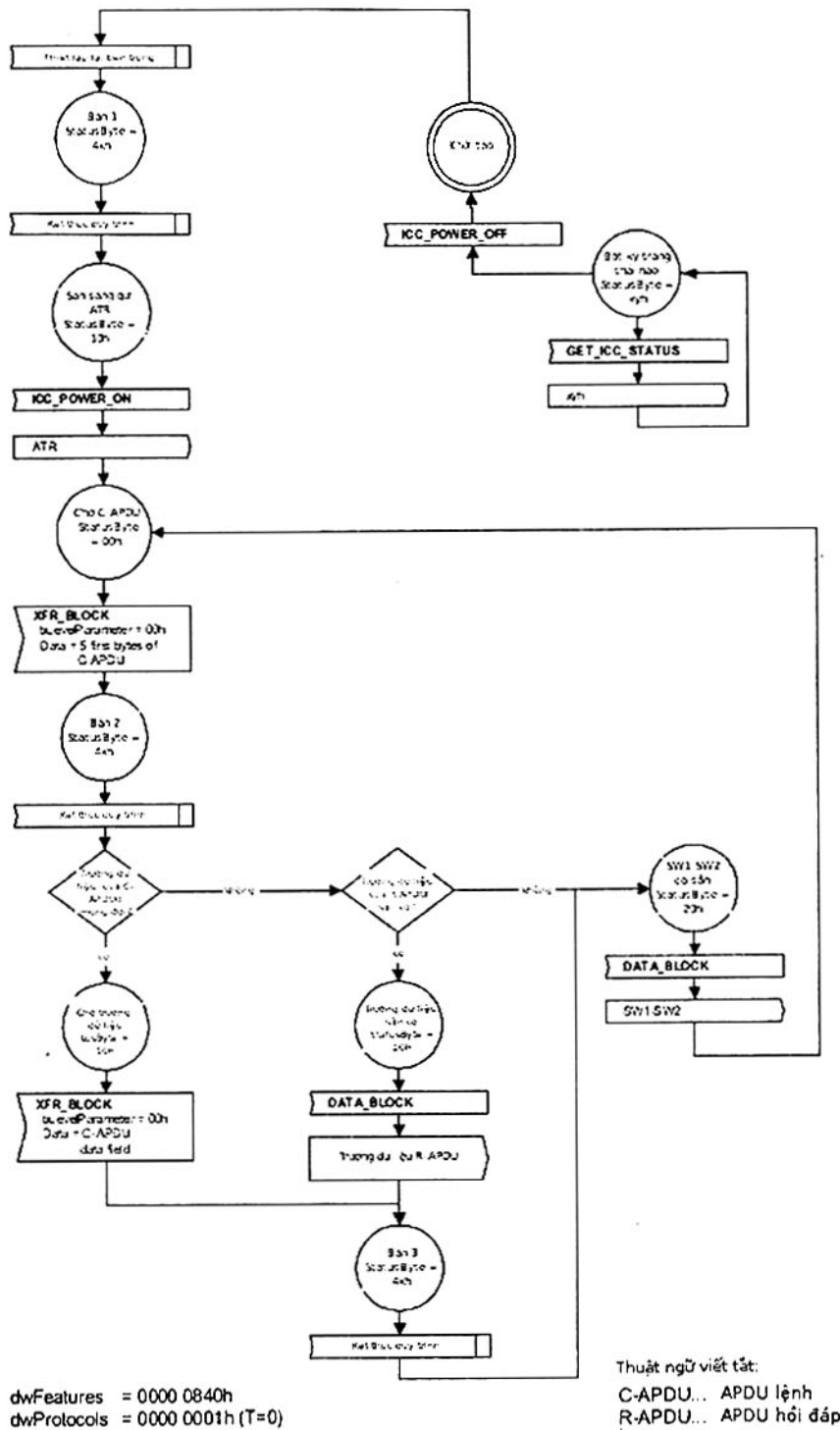
Khi một thiết bị USB được gắn vào kênh truyền và sau đó đạt được một trạng thái mà máy chủ có thể sử dụng các chức năng được cung cấp bởi thiết bị, thiết bị này được thiết kế như "được cấu hình". Thông điệp được truyền theo thứ tự để thiết lập USB-ICC ở trạng thái ban đầu, nhằm có được ATR và để truyền dữ liệu được đưa ra trong các sơ đồ trạng thái sau đây (xem Hình 3, 4 và 5).

TCVN 11167-12:2015

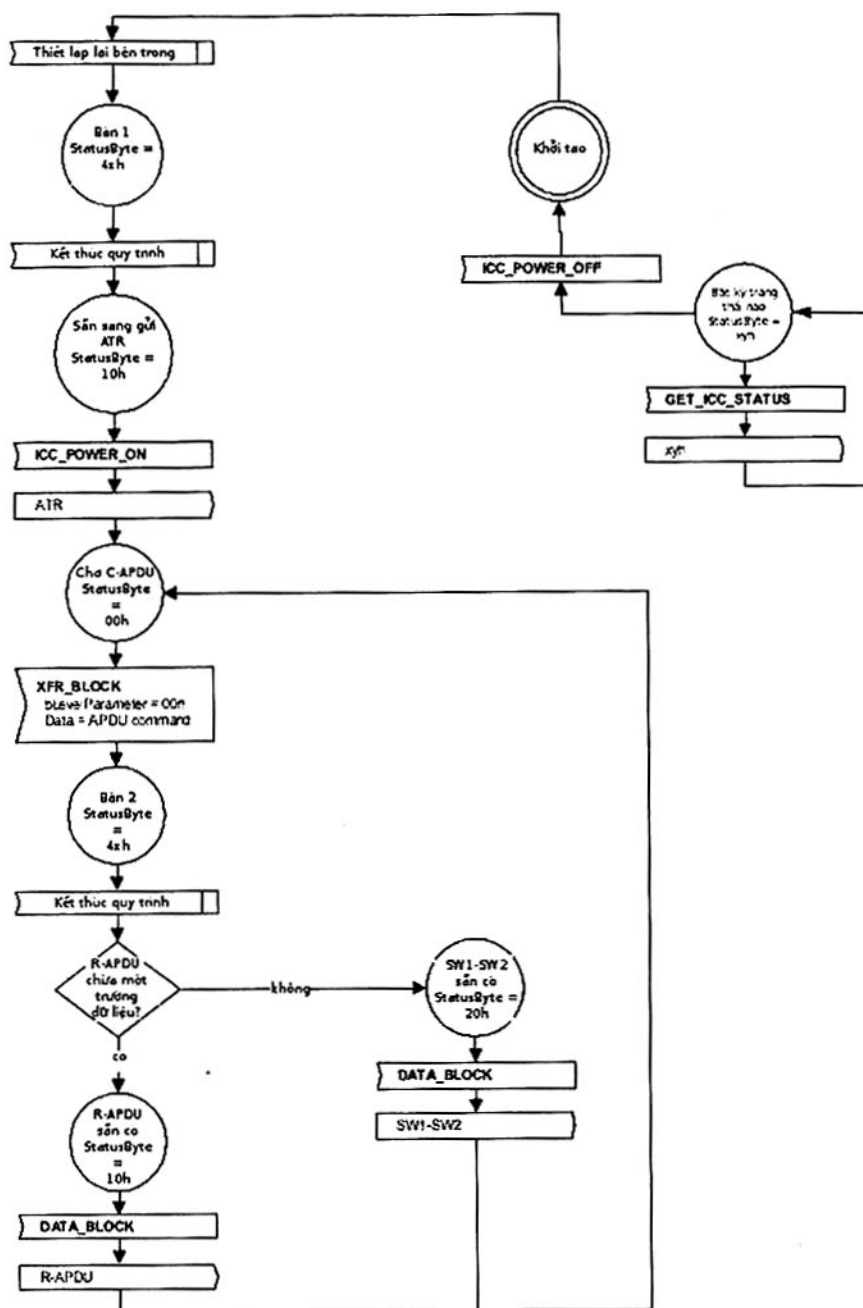
Đối với việc truyền chính xác của dữ liệu, các quy tắc chung sau đây phải được áp dụng:

- Nếu USB-ICC nhận được một yêu cầu mà không được gán cho trạng thái hiện tại trong lược đồ trạng thái, USB-ICC phải trả về một STALL và duy trì trạng thái hiện tại của nó.
- Nếu StatusByte chỉ ra rằng thẻ không đáp ứng (xem Bảng 24), máy chủ tốt nhất cần gửi ICC_POWER_OFF.

QUAN TRỌNG - Trạng thái của việc thực thi hiện tại phải không ảnh hưởng bởi trạng thái của nguồn giao diện USB. Ví dụ: một liệt kê kênh truyền phải không gây ra bất kỳ chuyển tiếp nào.



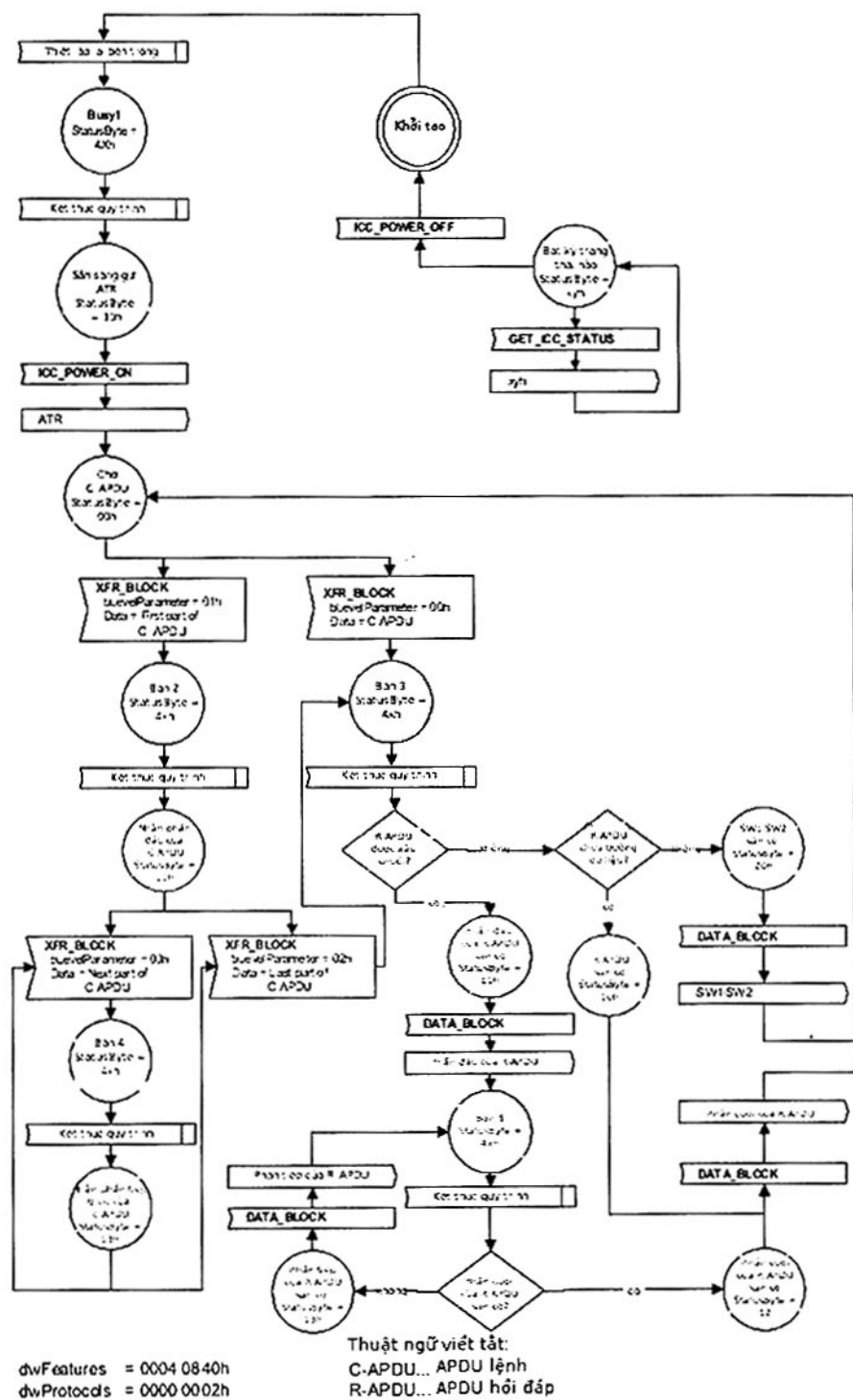
Hình 3 - Lược đồ trạng thái của USB-ICC đối với truyền điều khiển (phiên bản A) dùng truyền mức ký tự



dwFeatures = 0002 0840h
dwProtocols = 0000 0002h (T=1)

Thuật ngữ viết tắt:
C-APDU... APDU lệnh
R-APDU... APDU hồi đáp

Hình 4 - Lược đồ trạng thái của USB-ICC với truyền điều khiển (phiên bản A) dùng APDU ngắn



Hình 5 - Lược đồ trạng thái của USB-ICC với truyền điều khiển (phiên bản A) dùng APDU mở rộng

TCVN 11167-12:2015

Yêu cầu GET_ICC_STATUS thăm dò tình trạng thực thi của một lệnh APDU. Theo yêu cầu này, USB-ICC trả về StatusByte nhằm chỉ ra trạng thái thực thi. Nó có thể có các giá trị sau:

Bảng 24 - Mô tả StatusByte

StatusByte	Mô tả
4xh	<p>bận khi x phải được tăng theo chu kỳ.</p> <p>Khi nhận một chỉ báo bận, máy chủ phải hỗ trợ tuần tự GET_ICC_STATUS cho đến khi USB-ICC chỉ báo giá trị khác. Khoảng thời gian được điều chỉnh phụ thuộc. Để giữ băng thông, khoảng thời gian không được nhỏ hơn 10 mili-giây.</p> <p>CHÚ THÍCH Khi máy chủ dò tìm bốn bit có nghĩa ít nhất không thay đổi sau một khoảng nhất định, máy chủ có thể quá hạn với thiết bị này. Khoảng này được xem như quá hạn được điều chỉnh phụ thuộc và không được ít hơn 1 giây.</p>
20h	<p>trạng thái sẵn sàng gửi, chỉ các từ</p> <p>Chỉ ra rằng giai đoạn dữ liệu của DATA-BLOCK kế tiếp sẽ chỉ gồm SW1-SW2.</p>
1yh	<p>Nếu dwProtocols=00000001h và dwFeatures=0000040h</p> <p>10h: sẵn sàng gửi dữ liệu</p> <p>hay</p> <p>10h: sẵn sàng nhận dữ liệu</p> <p>Các từ trạng thái không được trả lại khi giá trị là 10.</p> <p>Khi GET_ICC-STATUS trả lại StatusByte=20h, một yêu cầu DATA_BLOCK kế tiếp phải được hỗ trợ nhằm có được các từ trạng thái.</p> <p>Nếu dwProtocols=00000002h và dwFeatures=000z0840h (với z=2 hoặc z=4), StatusByte có hai chức năng khác nhau.</p> <p>Khi bLevelParameter là 01h hoặc 03h trong yêu cầu XFR_BLOCK trước đó (lệnh APDU xâu chuỗi), StatusByte được dùng nhằm xác nhận việc xâu chuỗi các lệnh (tương ứng là 11h hoặc 13h) và chỉnh luồng dữ liệu (StatusByte = 4xh)</p> <p>Khi bLevelParameter trong yêu cầu XFR_BLOCK trước đó là 00h hoặc 02h (kết thúc lệnh APDU), StatusByte được dùng để chỉ ra việc xâu chuỗi APDU hồi đáp và tính toán luồng dữ liệu (StatusByte = 4xh):</p> <ul style="list-style-type: none"> 10h APDU hồi đáp bắt đầu và kết thúc với yêu cầu DATA_BLOCK kế tiếp 11h APDU hồi đáp bắt đầu với yêu cầu DATA_BLOCK kế tiếp và được tiếp tục 12h APDU hồi đáp bắt đầu và kết thúc với yêu cầu DATA_BLOCK kế tiếp 13h APDU hồi đáp bắt đầu và kết thúc với yêu cầu DATA_BLOCK kế tiếp và khối khác được theo 20h APDU hồi đáp chỉ chứa từ trạng thái và kết thúc với yêu cầu DATA_BLOCK kế tiếp.
80h	<p>tắt âm</p> <p>thẻ không chịu trách nhiệm</p>
00h	<p>USB-ICC sẵn sàng nhận một lệnh APDU.</p>

8.2.1.4 Trao đổi thông điệp mức APDU

Trong trường hợp độ dài của APDU hồi đáp vượt qua giá trị của wLength trong giai đoạn trạng thái của yêu cầu DATA_BLOCK, APDU hồi đáp phải truyền thành các khối liên tiếp. Trong trường hợp này, USB-ICC phải dùng cơ chế tương tự như với một APDU hồi đáp mở rộng.

8.2.1.5 Điều kiện lỗi

Điều kiện lỗi được trả lại trong StatusByte. Nếu thẻ không phản hồi, giá trị 80h sẽ trả về.

8.2.1.6 Truyền ngắt

Phiên bản A không dùng truyền ngắt.

8.2.2 Phiên bản B

Các yêu cầu kiểm soát theo phiên bản B tương tự việc trao đổi thông tin dùng ống thông điệp trong chế độ truyền bulk. Điều này đạt được bởi thực tế rằng mỗi yêu cầu OUT được theo sau bởi các yêu cầu IN. Cặp yêu cầu này được dùng phổ biến thể hiện cấu trúc của thông điệp OUT và thông điệp IN ở chế độ truyền bulk.

8.2.2.1 Yêu cầu cụ thể

Bảng sau định nghĩa các giá trị hợp lệ của bRequest:

Bảng 25 - Yêu cầu cụ thể lớp, phiên bản B

bRequest	Giá trị	Chiều mức dữ liệu	Mô tả
ICC_POWER_ON	62h	OUT	Tồn tại trạng thái ban đầu của một USB-ICC. ATR tái lập lại được trả lại trong giai đoạn dữ liệu của yêu cầu DATA_BLOCK kế tiếp
ICC_POWER_OFF	63h	OUT	Đặt USB-ICC với các điều kiện ban đầu.
XFR_BLOCK	65h	OUT	Truyền dữ liệu từ host tới USB-ICC.
DATA_BLOCK	6Fh	IN	Truyền dữ liệu từ USB-ICC tới host. Cũng trả lại thông tin được tạo ra bởi yêu cầu trước đó.
SLOT_STATUS	81h	IN	Giai đoạn dữ liệu của lệnh này bao gồm bStatus, bError và bReserved. Giá trị đối với bReversed phải là 00h.

8.2.2.2 Giai đoạn Thiết lập

Giai đoạn thiết lập gồm các yêu cầu cụ thể lớp và thông số tương ứng. Các mệnh đề sau cung cấp các giá trị và thông số cho từng yêu cầu cụ thể lớp và mô tả dữ liệu được truyền giữa máy chủ và USB-ICC.

Thông số của yêu cầu cụ thể lớp phải được thiết lập như sau:

- Giá trị của *bInterface* có cùng một giá trị với *bInterfaceNumber* được đưa ra trong Bảng 3.

TCVN 11167-12:2015

- Giá trị thông số dành riêng cho các yêu cầu cụ thể lớp được sử dụng trong các trường: *wValue* và *Windex* được chỉ định là *bRFU*, *wRFU* và *bReserved*. Giá trị của *bRFU* được đặt là 00h và giá trị của *wRFU* được đặt là 0000H. Giá trị cho *bReserved* được đưa ra trong bảng.
- Nếu USB-ICC nhận được một yêu cầu không hợp lệ hoặc nếu một yêu cầu hợp lệ chứa một giá trị tham số không hợp lệ (*wValue*, *Windex*, *wLength*), USB-ICC phải hồi đáp với một STALL.
- Trên một yêu cầu đầu vào, USB-ICC phải không trả về dữ liệu nhiều hơn được chỉ định bởi giá trị *wLength*. Nó có thể trả lại ít hơn.
- Trên một yêu cầu đầu ra, *wLength* phải luôn chỉ ra được chính xác lượng dữ liệu được gửi từ máy chủ tới USB-ICC. Khi USB-ICC nhận dữ liệu nhiều hơn trong giai đoạn thiết lập, nó phải hồi đáp với STALL. Máy chủ có thể hủy bỏ bất kỳ việc gán nào bằng cách gửi thẻ IN sớm. Trong trường hợp này, USB-ICC sẽ xác nhận thẻ IN với ACK. Nếu lượng dữ liệu nhận được không bằng *wLength*, USB-ICC phải loại bỏ các dữ liệu này.
- Đối với *ICC_POWER_ON*, *ICC_POWER_OFF* và *SLOT_STATUS*, máy chủ phải gửi các giá trị cho *wLength* như đã quy định trong các bảng tương ứng. Nếu không, USB-ICC phải hồi đáp với STALL.

Bảng 26 - Yêu cầu ICC_POWER_ON, phiên bản B

<i>bmRequestType</i>	<i>bRequest</i>	<i>wValue</i>	<i>wIndex</i>	<i>wLength</i>	Dữ liệu
00100001B	ICC_POWER_ON	<i>wRFU</i>	<i>bRFU</i> <i>bInterface</i>	0000h	Để trống

Trường *Windex* chỉ định *bRFU* trong byte cao và *bInterface* trong byte thấp. Trường *wValue* chỉ định *bRFU* trong byte cao và *bReserved* trong byte thấp.

Bảng 27 - Yêu cầu ICC_POWER_OFF, phiên bản B

<i>bmRequestType</i>	<i>bRequest</i>	<i>wValue</i>	<i>wIndex</i>	<i>wLength</i>	Dữ liệu
00100001B	ICC_POWER_OFF	<i>wRFU</i>	<i>bRFU</i> <i>bInterface</i>	0000h	Để trống

Các trường *Windex* định *bRFU* trong byte cao và *bInterface* trong byte thấp.

Bảng 28 - Yêu cầu XFR_BLOCK, phiên bản B

<i>bmRequestType</i>	<i>bRequest</i>	<i>wValue</i>	<i>wIndex</i>	<i>wLength</i>	Dữ liệu
00100001B	XFR_BLOCK	<i>BLevelParameter</i> - <i>bReserved</i> =00h	<i>bRFU</i> <i>bInterface</i>	Độ dài của dữ liệu	Lệnh APDU

Trường *Windex* chỉ định *bRFU* trong byte cao và *bInterface* trong byte thấp. Trường *wValue* chỉ định *BLevelParameter* trong byte cao và *bReserved* trong byte thấp.

Việc sử dụng *BLevelParameter* được giải thích trong Bảng 14.

Bảng 29 - Yêu cầu DATA_BLOCK, phiên bản B

bmRequestType	bRequest	wValue	wIndex	wLength	Dữ liệu
10100001B	DATA_BLOCK	wRFU	bRFU <i>bInterface</i>	Độ dài của dữ liệu+1. Giá trị của wLength phải lớn hơn hoặc bằng 4. Giá trị này cho phép USB-ICC trả lại thông tin trạng thái hoàn thành ở mức tối thiểu (xem Bảng 31). Đối với yêu cầu DATA_BLOCK đầu tiên sau một ICC_POWER_ON, wLength cần là dwMaxCCIDMessageLength	APDU hồi đáp hoặc thông tin được tạo ra bởi ICC_POWER_ON.

Trường Windex chỉ định bRFU trong byte cao và *bInterface* trong byte thấp.

Bảng 30 - Yêu cầu SLOT_STATUS

bmRequestType	bRequest	wValue	wIndex	wLength	Dữ liệu
10100001B	SLOT_STATUS	wRFU	bRFU <i>bInterface</i>	0003h	Bao gồm thông tin trạng thái/lỗi: <i>bStatus</i> , <i>bError</i> , <i>bReserved</i> . Giá trị nay đối với <i>bReserved</i> phải là 00h.

Trường Windex chỉ định bRFU trong byte cao và *bInterface* trong byte thấp.

CHÚ THÍCH Một sản phẩm không sử dụng một trình điều khiển cụ thể lớp có thể tuân thủ TCVN 11167-12. Trong trường hợp này, việc mã hóa các bit 5, 6 của *bmRequestType* thay đổi từ 01B đến 10B; ví dụ *bmRequestType* cho SLOT_STATUS sẽ là 11000001B.

8.2.2.3 ATR và truyền dữ liệu

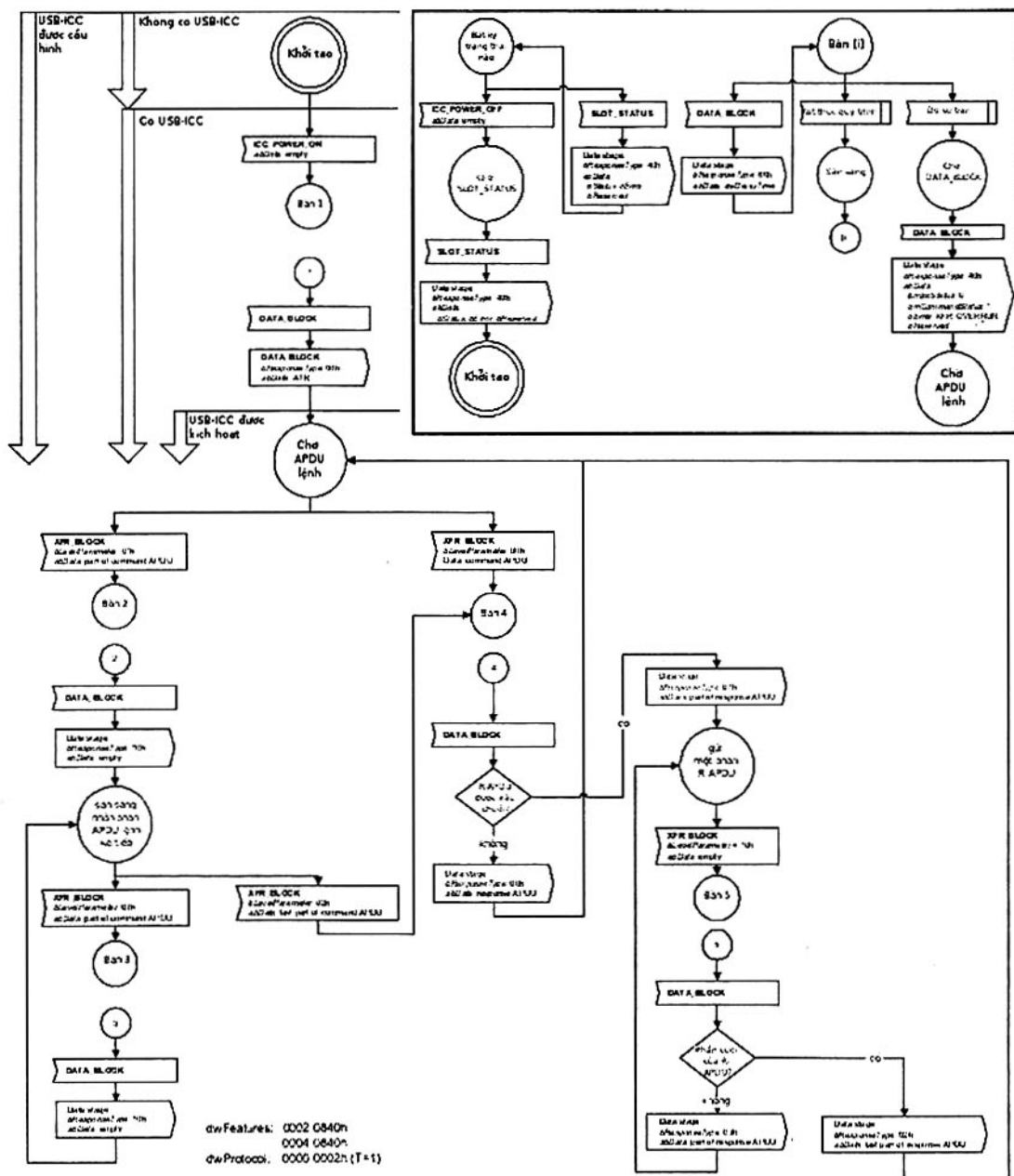
Khi một thiết bị USB được gắn vào kênh truyền và sau đó đạt được một trạng thái mà máy chủ có thể sử dụng các chức năng được cung cấp bởi thiết bị, thiết bị này được thiết kế như "đã cấu hình". Thông điệp được truyền theo thứ tự để thiết lập USB-ICC ở trạng thái ban đầu, để có được ATR và truyền dữ liệu được đưa ra trong lược đồ trạng thái Hình 6. Truyền sử dụng các trao đổi mức APDU. Hình 6 bao gồm việc truyền các APDU ngắn và APDU mở rộng.

Đối với việc truyền chính xác dữ liệu, các quy tắc chung sau phải áp dụng với các sơ đồ trạng thái:

- Nếu USB-ICC nhận một yêu cầu mà không được giao cho trạng thái hiện tại theo quy định trong lược đồ trạng thái, USB-ICC phải sẽ trả về một STALL và duy trì trạng thái hiện tại của nó.

- Nếu thiết bị giao diện gửi DATA_BLOCK và USB-ICC trả về trong giai đoạn dữ liệu giai đoạn các lỗi: ICC_MUTE hay HW_ERROR, máy chủ cần hỗ trợ ICC_POWER_OFF.

QUAN TRỌNG - Trạng thái của việc thực thi hiện tại phải không bị ảnh hưởng bởi trạng thái của nguồn giao diện USB. Ví dụ: một điều tra kênh truyền phải không gây ra bất kỳ chuyển dịch nào.



Hình 6 - Lược đồ trạng thái của USB-ICC đối với truyền điều khiển (phiên bản B)

CHÚ THÍCH Khi máy chủ gửi ICC_POWER_OFF, USB-ICC rơi vào tình trạng "hầu như không có mặt". Việc sử dụng thông điệp interrupt-IN cho trường hợp này được mô tả trong Điều 9.3

USB-ICC trả về yêu cầu DATA_BLOCK các giá trị sau trong giai đoạn dữ liệu:

Bảng 31 - Giai đoạn dữ liệu của DATA_BLOCK

Offset	Trường	Mô tả
0	bResponseType	<p>Chỉ ra loại thông tin abData bao gồm:</p> <p>00h: trường abData bao gồm thông tin được tạo ra bởi yêu cầu trước đó.</p> <p>40h: Thông tin trạng thái</p> <p>Trường abData bao gồm bStatus, bError và bReserved=00h.</p> <p>80h: Thăm dò</p> <p>Trường abData ba gồm thời gian trì hoãn (wDelayTime) cho đến khi host đợi gửi đi yêu cầu kế tiếp. Giá trị này được đưa ra theo các đơn vị của 10 mili-giây (ví dụ: 0078h = 1,2 giây). Nếu wDelayTime = 0000h, máy chủ phải đặt khoảng thời gian thăm dò tùy ý. Đối với tất cả các giá trị khác, máy chủ phải dùng giá trị đưa ra ở mức cao nhất.</p> <p>Đối với các APDU mở rộng và nếu APDU ngắn bao gồm nhiều byte hơn được chỉ định trong wLength của yêu cầu DATA_BLOCK:</p> <p>00h: APDU hồi đáp bắt đầu và kết thúc trong lệnh này.</p> <p>01h: APDU hồi đáp bắt đầu với lệnh này và được tiếp tục,</p> <p>02h: trường abData tiếp tục APDU hồi đáp và kết thúc APDU hồi đáp,</p> <p>03h: trường abData tiếp tục APDU hồi đáp và khối khác được theo dõi.</p> <p>10h: trường abData bị xóa trắng, lệnh APDU kế tiếp được mong đợi trong XFR_BLOCK kế tiếp.</p> <p>Cũng xem lược đồ trạng thái trong Hình 6.</p>
1	abData	Dữ liệu được gửi từ USB-ICC tới host.

Thông tin được truyền đi trong trường abData của DATA_BLOCK, phụ thuộc vào yêu cầu trước đó.

8.2.2.4 Mã hóa bLevelParameter đối với XFR_BLOCK

Giá trị của *bLevelParameter* chỉ định vị trí (khối đầu tiên, giữa, cuối cùng) của các khối truyền kế tiếp của lệnh APDU. Các giá trị sau đây được gán:

00h lệnh APDU bắt đầu và kết thúc bằng lệnh này

01h lệnh APDU bắt đầu với lệnh này và là tiếp tục

02h APDU lệnh và kết thúc lệnh APDU

03h lệnh APDU tiếp tục và khối khác theo sau

10h giai đoạn dữ liệu là trống rỗng, APDU hồi đáp kế tiếp được mong đợi trong yêu cầu DATA_BLOCK tiếp theo

TCVN 11167-12:2015

Xem thêm các sơ đồ trạng thái trong Hình 6.

8.2.2.5 Trao đổi thông điệp mức APDU

Trong trường hợp mà APDU hồi đáp vượt quá giá trị của *wLength* trong giai đoạn thiết lập yêu cầu *DATA_BLOCK*, APDU hồi đáp phải được truyền trong khối tiếp theo. Đối với truyền khối lớn này, USB-ICC phải sử dụng cơ chế tương tự như đối với các APDU hồi đáp mở rộng.

8.2.2.6 Điều kiện trạng thái và lỗi được báo cáo bởi yêu cầu của USB

Phiên bản B trả về trạng thái và điều kiện lỗi trong giai đoạn dữ liệu của *DATA_BLOCK*. Điều kiện này được chỉ định bởi *bResponseType* = 40h. Ngoài ra, USB-ICC sẽ hồi đáp bằng một thủ tục bắt tay STALL khi nó nhận được một yêu cầu không hợp lệ hoặc nếu một yêu cầu hợp lệ chứa một giá trị tham số không hợp lệ (*wValue*, *wIndex*, *wLength*).

Nếu *bResponseType* = 40h, trường *abData* gồm trạng thái và thông tin lỗi.

Trường *bStatus* gồm hai trường bitmap chứa thông tin về trạng thái của USB-ICC (*bmlccStatus*) và lệnh được xử lý (*bmCommandStatus*). Hai bảng dưới đây đưa ra các giá trị cho trạng thái và các mã lỗi.

Bảng 32 - Bản đồ bit đối với trường *bStatus*

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	<i>bmlccStatus</i>	1 (2 bit)	0, 1, 2	0 = USB-ICC được trình bày và kích hoạt. 1 = USB-ICC được trình bày nhưng không kích hoạt. 2 = USB-ICC không được trình bày. 3 = RFU
(2 bit)		(4 bit)		RFU
(6 bit)	<i>bmCommandStatus</i>	(2 bit)	0, 1	0 = Được xử lý không có lỗi. 1 = Bị sai sót, điều kiện lỗi được đưa ra bởi <i>bError</i> . 2 = RFU. 3 = RFU.
1	<i>bError</i>	1		Mã lỗi

Bảng 33 - Mã lỗi đối với bError

Tên lỗi	Mã lỗi	Các nguyên nhân có thể
ICC_MUTE	-2 (FEh)	Các ứng dụng của USB-ICC không hồi đáp hoặc ATR không được gửi bởi USB-ICC.
XFR_OVERRUN	-4 (FCh)	USB-ICC dò tìm một lỗi tràn Offset khi nhận một khối dữ liệu.
HW_ERROR	-5 (FBh)	USB-ICC dò tìm một lỗi phần cứng.
	-64 tới -127 (C0h – 81h)	Được người dùng định nghĩa.
	-3 (FDh) -8 tới -14 (F8h – F2h) -16 (F0h) -17 (EFh) -32 (E0h)	Các giá trị này không được dùng bởi USB-ICC.
	tất cả các mã khác (80h và các giá trị khác được điền vào khoảng trống)	Được bảo toàn cho sử dụng dự kiến.

Nếu giá trị của *bmCommandStatus* bằng 0 hoặc RFU, giá trị của *bError* bằng 0.

8.3 Truyền ngắt

Chế độ truyền bulk và chế độ truyền điều khiển (phiên bản B) đưa ra tùy chọn một điểm cuối interrupt-IN. Điểm này được dùng để nhắc báo máy chủ của sự kiện rằng có thể xảy ra bất đối xứng với trao đổi lệnh/hồi đáp giữa máy chủ và USB-ICC.

USB-ICC có thể nhắc báo máy chủ về việc chèn/gỡ bỏ nó.

8.3.1 Sự kiện chèn/gỡ bỏ thấy được

Thuật ngữ “thấy được” được dùng nhằm nhấn mạnh USB-ICC được dự định là gỡ bỏ khỏi thiết bị giao diện mặc dù nó vẫn còn được cấp nguồn.

Bảng 34 - Thông điệp Interrupt-IN

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	bMessageType	1	50h	Chỉ ra NotifySlotChange.
1	bmSlotIccState	1	000000xyB	<p>USB-ICC chỉ ra trạng thái (gỡ bỏ hay chèn vào) trong hai bit đánh dấu có ít ý nghĩa nhất. Một bit báo cáo:</p> <p>0b = USB-ICC không được trình bày. 1b = USB-ICC được trình bày.</p> <p>Bit còn lại báo cáo dù sự hiện diện của USB-ICC thay đổi khi thông điệp NotifySlotChange được gửi:</p> <p>0b = không đổi 1b = thay đổi</p> <p>Tất cả các bit khác phải đặt là 0.</p>

Đối với hoạt động chính xác của yêu cầu ngắt, các điều kiện sau được áp dụng:

- Khi USB-ICC thoát từ trạng thái "ban đầu" bởi PC_to_RDR_IccPowerOn (ICC_POWER_ON), USB-ICC phải gửi một thông điệp NotifySlotChange với *bmSlotIccState* = 00000011B.
- USB-ICC có thể nhập "hầu như không có mặt" tại bất kỳ thời điểm nào. Máy chủ sẽ nhận được thông điệp NotifySlotChange với *bmSlotIccState* = 00000010B. USB-ICC không phải gửi thông điệp NotifySlotChange sau khi nó đã nhận được PC_to_RDR_IccPowerOff (ICC_POWER_OFF).

CHÚ THÍCH Điều kiện đầu tiên cho phép máy chủ phát hiện một thẻ không phản hồi. Điều kiện thứ hai đảm bảo rằng thông điệp gián đoạn "hầu như không có mặt" là một sự kiện không đồng bộ được gây ra bởi USB-ICC. Nó không phải là kết quả của một thông điệp OUT hoặc yêu cầu nhận được từ máy chủ.

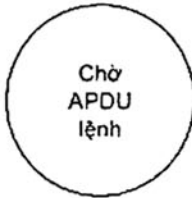
Phụ lục A

(tham khảo)

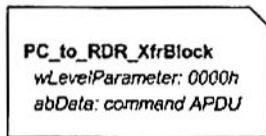
Kí hiệu đối với lược đồ trạng thái



Trạng thái Khởi tạo trong lược đồ trạng thái



Trạng thái của lược đồ trạng thái



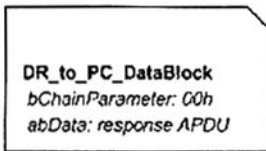
USB-ICC nhận dữ liệu.

Với truyền số lượng lớn:

Loại tin nhắn và các thông số/giá trị liên quan cho sẵn.

Với truyền kiểm soát:

Yêu cầu kiểm soát và các thông số/giá trị liên quan cho sẵn.



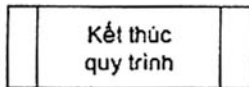
USB-ICC gửi dữ liệu

Với truyền số lượng lớn:

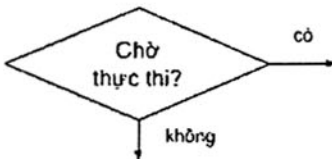
Loại tin nhắn và các thông số/giá trị liên quan cho sẵn. USB-ICC luôn khởi tạo việc truyền thông điệp này.

Với truyền kiểm soát:

Yêu cầu kiểm soát và các thông số/giá trị liên quan cho sẵn. Máy chủ luôn khởi tạo việc truyền dữ liệu này bằng cách gửi DATA_BLOCK hay SLOT_STATUS. Do vậy, hai yêu cầu này thường được thể hiện bằng hai hộp hình mũi tên: một hộp với chiều mũi tên VÀO (yêu cầu) và một hộp với chiều mũi tên RA (dữ liệu)



Trạng thái Khởi tạo trong lược đồ trạng thái



Nhánh quyết định

Phụ lục B

(tham khảo)

Kịch bản đối với truyền USB

Các chuỗi sau mô tả nhưng không đầy đủ với tất cả các trường hợp có thể tại đây. Nghĩa là các ví dụ của các chuỗi có thể của USB_ICC phải quản lý được. Các ví dụ không chỉ quản lý các trường hợp hợp lỗi (STALL trong các giao dịch USB), mà còn là các điều kiện được quản lý theo mức giao thức của ISO (ví dụ: T=0, trường hợp 2, APDU ngắn, N_a không được chấp nhận).

Truyền bulk:**Trao đổi thông điệp mức APDU, lệnh trường hợp 3, APDU ngắn****PC_to_RDR_XfrBlock****(1) message →**

dwLength = Lc + 00000005h

bSlot = 00h, bSeq = 00h, bReserved = 00h,

wLevelParameter = 0000h

abData = CLA, INS, P1, P2, Lc, Data(Lc)

(2) message ← RDR_to_PC_DataBlock

dwLength = 00000002h

bSlot = 00h, bSeq = 00h

bStatus = 00h, bError = 00h

bChainParameter = 00h

abData = SW1, SW2

Truyền bulk:**Trao đổi thông điệp mức APDU, lệnh trường hợp 3, APDU mở rộng****PC_to_RDR_XfrBlock****(1) message →**

dwLength = BufferLength

bSlot = 00h, bSeq = 00h, bReserved = 00h,

wLevelParameter = 0001h

abData = CLA, INS, P1, P2, Lc,
Data(BufferLength - 7)**(2) message ← RDR_to_PC_DataBlock**

dwLength = 00000000h

bSlot = 00h, bSeq = 00h

bStatus = 00h, bError = 00h

bChainParameter = 10h

abData = empty

PC_to_RDR_XfrBlock**(3) message →**

dwLength = BufferLength

bSlot = 00h, bSeq = 01h, bReserved = 00h,

wLevelParameter = 0003h

abData = Data(BufferLength)

(4) message ← RDR_to_PC_DataBlock

dwLength = 00000000h

bSlot = 00h, bSeq = 01h

bStatus = 00h, bError = 00h

bChainParameter = 10h

abData = empty

PC_to_RDR_XfrBlock**(5) message →**

dwLength = remaining part of data

bSlot = 00h, bSeq = 00h, bReserved = 00h,

wLevelParameter = 0002h

abData = Data(<buffer length)

(5) message ← RDR_to_PC_DataBlock
 dwLength = 00000002h
 bSlot = 00h, bSeq = 02h
 bStatus = 00h, bError = 00h
 bChainParameter = 00h
 abData = SW1, SW2

Truyền điều khiển, phiên bản A: Lệnh trường hợp 1

dwFeatures = 0000 0840h
 dwProtocols = 0000 0001h

XFR_BLOCK

wValue: bLevelParameter = 00h; bRFU = 00h
 wIndex: bRFU = 00h; bInterface = 00h
 wLength = 0005h
 abData = CLA, INS, P1, P2, P3=00h

(1) request →

(1) data →

GET_ICC_STATUS

wValue: wRFU = 0000h
 wIndex: bRFU = 00h; bInterface = 00h,
 wLength = 0001h

(2) request →

(2) data ← StatusByte: 4xh
 StatusByte: 20h

if (StatusByte=20h)

if (StatusByte=4xh)

Break

(2) Repeat

ready to send status words only

(3) request →

DATA_BLOCK

wValue: wRFU = 0000h
 wIndex: bRFU = 00h; bInterface = 00h
 wLength = 0002h

(3) data ← SW1, SW2
 End

Truyền điều khiển, phiên bản A: Lệnh trường hợp 2, APDU ngắn

dwFeatures = 0000 0840h
 dwProtocols = 0000 0001h

XFR_BLOCK

wValue: bLevelParameter = 00h, bRFU = 00h
 wIndex: bRFU = 00h, bInterface = 00h
 wLength = 0005h
 abData = CLA, INS, P1, P2, Le

(1) request →

(1) data →

GET_ICC_STATUS

wValue: wRFU = 0000h
 wIndex: bRFU = 00h, bInterface = 00h
 wLength = 0001h

(2) request →

(2) data ← StatusByte: 4xh
 StatusByte: 10h

if (StatusByte=10h)

if (StatusByte=4xh)

Break

(2) Repeat

ready to send data

(3) request →

DATA_BLOCK

wValue: wRFU = 0000h
 wIndex: bRFU = 00h, bInterface = 00h
 wLength = Le

(3) data ← Data(Le)

GET_ICC_STATUS

wValue: wRFU = 0000h

(4) request →

TCVN 11167-12:2015

wIndex: bRFU = 00h, bInterface = 00h
wLength = 0001h

if (StatusByte=20h)
if (StatusByte=4xh)

DATA_BLOCK

wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0002h

(4) data ← StatusByte: 4xh
StatusByte: 20h

Break
(4) Repeat

ready to send status words only

(5) request →

(5) data ← SW1, SW2

End

Truyền điều khiển, phiên bản A: Lệnh trường hợp 3, APDU ngắn

dwFeatures = 0000 0840h
dwProtocols = 0000 0001h

XFR_BLOCK

wValue: bLevelParameter = 00h, bRFU = 00h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0005h
abData = CLA, INS, P1, P2, Lc

(1) request →

(1) data →

GET_ICC_STATUS

wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0001h

(2) request →

(2) data ← StatusByte: 4xh
StatusByte: 10h

if (StatusByte=10h)
if (StatusByte=4xh)

Break
(2) Repeat

ready to receive data

(3) request →

XFR_BLOCK

wValue: bLevelParameter = 00h, bRFU = 00h
wIndex: bRFU = 00h, bInterface = 00h
wLength = Lc
abData = Data(Lc)

(3) data →

GET_ICC_STATUS

wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0001h

(4) request →

(4) data ← StatusByte: 4xh
StatusByte: 20h

if (StatusByte=20h)
if (StatusByte=4xh)

Break
(4) Repeat

ready to send status words only

(5) request →

DATA_BLOCK

wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0002h

(5) data ← SW1, SW2

End

Truyền điều khiển, phiên bản A: Trao đổi thông điệp mức APDU, lệnh trường hợp 1

dwFeatures = 0002 0840h

dwProtocols= 0000 0002h

XFR_BLOCK

(1) request →

wValue: bLevelParameter = 00h, bRFU = 00h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0004h

abData = CLA, INS, P1, P2

(1) data →

GET_ICC_STATUS

(2) request →

wValue: wRFU = 0000h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0001h

(2) data ← StatusByte: 4xh

StatusByte: 20h

if (StatusByte=20h)

Break

if (StatusByte=4xh)

(2) Repeat

DATA_BLOCK

(3) request →

wValue: wRFU = 0000h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0002h

(3) data ← SW1, SW2

End

Truyền điều khiển, phiên bản A: Trao đổi thông điệp mức APDU, lệnh trường hợp 2, APDU ngắn

dwFeatures = 0002 0840h

dwProtocols= 0000 0002h

XFR_BLOCK

(1) request →

wValue: bLevelParameter = 00h, bRFU = 00h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0005h

abData = CLA, INS, P1, P2, Le

(1) data →

GET_ICC_STATUS

(2) request →

wValue: wRFU = 0000h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0001h

(2) data ← StatusByte: 4xh

StatusByte: 10h

if (StatusByte=10h)

Break

if (StatusByte=4xh)

(2) Repeat

DATA_BLOCK

(3) request →

wValue: wRFU = 0000h

wIndex: bRFU = 00h, bInterface = 00h

wLength = Le+02h

(3) data ← Data(Le), SW1, SW2

End

Truyền điều khiển, phiên bản A: Trao đổi thông điệp mức APDU, lệnh trường hợp 3, APDU ngắn

dwFeatures = 0002 0840h

dwProtocols= 0000 0002h

XFR_BLOCK

(1) request →

wValue: bLevelParameter = 00h, bRFU = 00h

wIndex: bRFU = 00h, bInterface = 00h

TCVN 11167-12:2015

wLength = Lc + 0005h
abData = CLA, INS, P1, P2, Lc, Data(Lc) (1) →

GET_ICC_STATUS (2) request →
wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0001h (2) data ← StatusByte: 4xh
StatusByte: 20h

if (StatusByte=20h) Break
if (StatusByte=4xh) (2) Repeat

DATA_BLOCK (3) request →
wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0002h (3) data ← SW1, SW2

End

Truyền điều khiển, phiên bản A: Trao đổi thông điệp mức APDU, lệnh trường hợp 4, APDU ngắn

dwFeatures = 0002 0840h
dwProtocols = 0000 0002h

XFR_BLOCK (1) request →
wValue: bLevelParameter = 00h, bRFU = 00h
wIndex: bRFU = 00h, bInterface = 00h
wLength = Lc + 0006h
abData = CLA, INS, P1, P2, Lc, Data(Lc), Le (1) data →

GET_ICC_STATUS (2) request →
wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0001h (2) data ← StatusByte: 4xh
StatusByte: 10h

if (StatusByte=10h) Break
if (StatusByte=4xh) (2) Repeat

DATA_BLOCK (3) request →
wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = Le + 02h (3) data ← Data(Le), SW1, SW2

End

Truyền điều khiển, phiên bản A: Trao đổi thông điệp mức APDU, lệnh trường hợp 2, APDU mở rộng

dwFeatures = 0004 0840h
dwProtocols = 0000 0002h

XFR_BLOCK (1) request →
wValue: bLevelParameter = 00h, bRFU = 00h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0007h
abData = CLA, INS, P1, P2, P3=00h, Le (1) data →

GET_ICC_STATUS (2) request →
wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0001h (2) data ← StatusByte: 4xh
StatusByte: 10h

	StatusByte: 11h
if (StatusByte=10h OR StatusByte=11h)	Break
if (StatusByte=4xh)	(2) Repeat
if (10h)	Execute the next request
DATA_BLOCK	(3) request →
wValue: wRFU = 0000h	
wIndex: bRFU = 00h, bInterface = 00h	
wLength = BufferLength	(3) data ← Data(<=BufferLength-2), SW1, SW2
	End
if (11h)	Loop the next two requests
DATA_BLOCK	(n) request →
wValue: wRFU = 0000h	
wIndex: bRFU = 00h, bInterface = 00h	
wLength = BufferLength	(n) data ← Data(BufferLength)
GET_ICC_STATUS	(n+1) request →
wValue: wRFU = 0000h	
wIndex: bRFU = 00h, bInterface = 00h	
wLength = 0001h	(n+1) data ← StatusByte: 4xh StatusByte: 12h StatusByte: 13h
	Break
if (StatusByte=12h)	(n+1) Repeat
if (StatusByte=4xh)	(n) Repeat
if (StatusByte=13h)	
DATA_BLOCK	(m) request →
wValue: wRFU = 0000h	
wIndex: bRFU = 00h, bInterface = 00h	
wLength = BufferLength	(m) data ← Data(<=BufferLength-2), SW1, SW2
	End

Truyền điều khiển, phiên bản A: Trao đổi thông điệp mức APDU, lệnh trường hợp 3, APDU mở rộng

dwFeatures = 0004 0840h
dwProtocols = 0000 0002h

XFR_BLOCK	(1) request →
wValue: bLevelParameter = 00h/01h, bRFU = 00h	
wIndex: bRFU = 00h, bInterface = 00h	
wLength = BufferLength	
abData = CLA, INS, P1, P2, P3=00h, Lc, Data(BufferLength-7)	(1) data →
GET_ICC_STATUS	(2) request →
wValue: wRFU = 0000h	
wIndex: bRFU = 00h, bInterface = 00h	
wLength = 0001h	(2) data ← StatusByte: 4xh StatusByte: 20h StatusByte: 11h
	Break
if (StatusByte=20h OR StatusByte=11h)	(2) Repeat
if (StatusByte=4xh)	execute the next request
if (StatusByte=10h)	
DATA_BLOCK	(3) request →
wValue: wRFU = 0000h	
wIndex: bRFU = 00h, bInterface = 00h	

TCVN 11167-12:2015

wLength = 0002h

if (StatusByte=11h)

XFR_BLOCK

wValue: bLevelParameter = 03h/02h, bRFU = 00h

wIndex: bRFU = 00h, bInterface = 00h

wLength = BufferLength

abData = Data(BufferLength)

GET_ICC_STATUS

wValue: wRFU = 0000h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0001h

if (StatusByte=20h)

if (StatusByte=13h)

if (StatusByte=4xh)

DATA_BLOCK

wValue: wRFU = 0000h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0002h

(3) data ← SW1, SW2

End

Loop the next two requests

(n) request →

(n) data →

(n+1) request →

data ←

StatusByte: 4xh

StatusByte: 20h

StatusByte: 13h

Break

(n) Repeat

(n+1) Repeat

(m) request →

(m) data ← SW1, SW2

End

Truyền điều khiển, phiên bản B: Trao đổi thông điệp mức APDU, lệnh trường hợp 1, APDU ngắn

dwFeatures = 0002 0840h

dwProtocols= 0000 0002h

XFR_BLOCK

wValue: bLevelParameter = 00h, bReserved = 00h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0004h

abData = CLA, INS, P1, P2

(1) request →

(1) data →

DATA_BLOCK

wValue: wRFU = 0000h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0003h

(2) request →

(2) data ← 80h, wDelayTime

or

(2) data ← 00h, SW1, SW2

Break

(2) Repeat

End

if (bResponseType=00h)

if (bResponseType=80h)

Truyền điều khiển, phiên bản B: Trao đổi thông điệp mức APDU, lệnh trường hợp 2, APDU ngắn

dwFeatures = 0002 0840h

dwProtocols= 0000 0002h

XFR_BLOCK

wValue: bLevelParameter = 00h, bReserved = 00h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0005h

abData = CLA, INS, P1, P2, Le

(1) request →

(1) data →

DATA_BLOCK

wValue: wRFU = 0000h
 wIndex: bRFU = 00h, bInterface = 00h
 wLength = Le + 0003h

if (bResponseType=00h)
 if (bResponseType=80h)

(2) request →

(2) data ← 80h, wDelayTime
 or
 (2) data ← 00h, Data(Le), SW1, SW2
 Break
 (2) Repeat

End

Truyền điều khiển, phiên bản B: Trao đổi thông điệp mức APDU, lệnh trường hợp 3, APDU ngắn

dwFeatures = 0002 0840h
 dwProtocols = 0000 0002h

XFR_BLOCK

wValue: bLevelParameter = 00h, bReserved = 00h
 wIndex: bRFU = 00h, bInterface = 00h
 wLength = Lc + 0005h
 abData = CLA, INS, P1, P2, Lc, Data(Lc)

(1) request →

(1) data →

DATA_BLOCK

wValue: wRFU = 0000h
 wIndex: bRFU = 00h, bInterface = 00h
 wLength = 0003h

if (bResponseType=00h)
 if (bResponseType=80h)

(2) request →

(2) data ← 80h, wDelayTime
 or
 (2) data ← 00h, SW1, SW2
 Break
 (2) Repeat

End

Truyền điều khiển, phiên bản B: Trao đổi thông điệp mức APDU, lệnh trường hợp 4, APDU ngắn

dwFeatures = 0002 0840h
 dwProtocols = 0000 0002h

XFR_BLOCK

wValue: bLevelParameter = 00h, bReserved = 00h
 wIndex: bRFU = 00h, bInterface = 00h
 wLength = Lc + 0006h
 abData = CLA, INS, P1, P2, Lc, Data(Lc), Le

(1) request →

(1) data →

DATA_BLOCK

wValue: wRFU = 0000h
 wIndex: bRFU = 00h, bInterface = 00h
 wLength = Le + 0003h

if (bResponseType=00h)
 if (bResponseType=80h)

(2) request →

(2) data ← 80h, wDelayTime
 or
 (2) data ← 00h, Data(Le), SW1, SW2
 Break
 (2) Repeat

End

TCVN 11167-12:2015

Truyền điều khiển, phiên bản B: Trao đổi thông điệp mức APDU, lệnh trường hợp 2, APDU mở rộng

dwFeatures = 0004 0840h
dwProtocols = 0000 0002h

XFR_BLOCK

wValue: bLevelParameter = 00h, bReserved = 00h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0007h
abData = CLA, INS, P1, P2, P3=00h, Le

(1) request →

(1) data →
Loop the next request
(2) request →

DATA_BLOCK

wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = BufferLength

(2) data ← 80h, wDelayTime
or
(2) data ← 00h, Data(Le), SW1, SW2
or
(2) data ← 01h, Data(BufferLength)
or
(2) data ← 03h, Data(BufferLength),
or
(2) data ← 02h, Data(<=BufferLength-2),
SW1, SW2

if (bResponseType=00h OR bResponseType=02)
if (bResponseType=01h OR bResponseType=03)

(2) End

XFR_BLOCK

wValue: bLevelParameter = 10h, bReserved = 00h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0000h
abData = empty

(3) request →

(3) →
(2) Repeat
(2) Repeat

if (bResponseType=80h)

Truyền điều khiển, phiên bản B: Trao đổi thông điệp mức APDU, lệnh trường hợp 3, APDU mở rộng

dwFeatures = 0004 0840h
dwProtocols = 0000 0002h

XFR_BLOCK

wValue: bLevelParameter = 00h/01h, bReserved = 00h
wIndex: bRFU = 00h, bInterface = 00h
wLength = BufferLength
abData = CLA, INS, P1, P2, P3=00h, Lc,
Data(BufferLength-7)
if (bLevelParameter=00h)

(1) request →

(1) data →

DATA_BLOCK

wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h
wLength = 0003h

(2) request →

(2) data ← 80h, wDelayTime
(2) Repeat
(2) data ← 0h, SW1, SW2
End

if (bLevelParameter=01h)

DATA_BLOCK

wValue: wRFU = 0000h
wIndex: bRFU = 00h, bInterface = 00h

(3) request →

wLength = 0003h

(3) data ← 80h, wDelayTime

(3) Repeat

or

(3) data ← 10h

Loop the next two requests

(n) request →

XFR_BLOCK

wValue: bLevelParameter = 03h/02h, bReserved = 00h

wIndex: bRFU = 00h, bInterface = 00h

wLength = BufferLength

abData = Data(BufferLength)

(n) data →

DATA_BLOCK

wValue: wRFU = 0000h

wIndex: bRFU = 00h, bInterface = 00h

wLength = 0003h

(n+1) request →

(n+1) data ← 80h, wDelayTime

(n+1) Repeat

(n+1) data ← 00h, SW1, SW2

(n+1) End

(n+1) data ← 10h

(n+1) Repeat

if (bLevelParameter=02h)

if (bLevelParameter=03h)

Phụ lục C

(tham khảo)

Thuật ngữ và định nghĩa trong đặc tả USB

Các thuật ngữ và định nghĩa liệt kê bên dưới được trích từ Chương 2: Thuật ngữ và thuật ngữ viết tắt của đặc tả USB:

Truyền

Một hay nhiều trao đổi bus nhằm chuyển thông tin giữa một khách hàng phần mềm và chức năng của nó.

Loại truyền

Xác định các đặc tính của luồng dữ liệu giữa một máy khách phần mềm và chức năng của nó. Bốn loại chuẩn được xác định: kiểm soát, ngắt, bulk và đồng thời.

Truyền bulk

Một trong bốn loại truyền USB. Truyền bulk là không định kỳ, kết nối truyền loạt lớn thường được dùng cho một truyền mà có thể dùng bất kỳ băng thông sẵn có nào và cũng có thể bị trì hoãn cho đến khi băng thông có sẵn. Xem thêm loại truyền.

truyền điều khiển

Một trong bốn loại truyền USB. truyền điều khiển hỗ trợ kết nối loại cấu hình/lệnh/trạng thái giữa máy khách và chức năng. Xem thêm loại truyền.

Ống mặc định

Ống thông điệp được tạo ra bởi phần mềm hệ thống USB nhằm vượt qua thông tin kiểm soát điều khiển và thông tin trạng thái giữa máy chủ và kiểm soát điểm cuối 0 của thiết bị USB.

Điểm cuối thiết bị

Phần được chỉ định duy nhất của một thiết bị USB là nguồn hoặc phần chứa thông tin theo một luồng thông tin liên lạc giữa máy chủ và thiết bị. Xem địa chỉ điểm cuối.

Địa chỉ điểm cuối

Sự kết hợp của một số điểm cuối và một chiều điểm cuối của thiết bị USB. Mỗi địa chỉ điểm cuối hỗ trợ truyền dữ liệu theo một hướng.

Máy chủ

Hệ thống máy chủ, nơi bộ điều khiển máy chủ USB được cài đặt. Điều này gồm các nền tảng phần cứng máy chủ (máy tính, bus, .v.v.) và hệ điều hành được sử dụng.

Từ

Một phần tử dữ liệu có hai byte (16 bit) theo kích thước.

Phụ lục D

(tham khảo)

Lớp thiết bị thẻ thông minh bộ mô tả cụ thể lớp

Bảng sau mô tả bộ mô tả cụ thể lớp được đưa ra trong đặc tả CCID, Chương 3.5

Bảng 3-5 - Bộ mô tả lớp CCID

Offset	Trường	Kích cỡ	Giá trị	Mô tả
0	bLength	1	36h	Kích cỡ của bộ mô tả, tính theo byte.
1	bDescriptorType	1	??	Loại bộ mô tả tính năng CCID
2	bcdCCID	2	0100h	Số phát hành đặc tả CCID hệ thập phân mã hóa nhị phân (ví dụ: 2.10 là 0210h)
4	bMaxSlotIndex	1		Chỉ mục của khe có sẵn cao nhất trong thiết bị này. Tất cả các khe liên tiếp bắt đầu từ 00h. Ví dụ: 0Fh = 16 khe trên thiết bị được đánh số từ 00h tới 0Fh.
5	bVoltageSupport	1		Giá trị chỉ ra điện thế mà CCID có thể cung cấp cho các khe của nó. Giá trị này là một thao tác OR đảo bit được thực hiện theo các giá trị sau: <ul style="list-style-type: none"> - 01h 5.0V - 02h 3.0V - 04h 1.8V Các bit khác là RFU.
6	dwProtocols	4	RRRR PPPP	RRRR - viết hoa - là RFU = 0000h PPPP - viết thường - mã hóa các loại giao thức được hỗ trợ. Một '1' là một vị trí bit được đưa ra chỉ ra sự hỗ trợ đối với giao thức ISO tương ứng. 0001h = Giao thức T=0 0002h = Giao thức T=1 Tất cả các bit được dành riêng và phải được đặt là 0. Trường tương ứng với các định nghĩa đặc tả PCSC. Xem PCSC phần 3, Bảng 3-1 Thẻ 0x0120 Ví dụ: 00000003h hỗ trợ cho T=0 và T=1.
10	dwDefaultClock	4		Tần số khóa ICC mặc định theo kHz được mã hóa như giá trị nguyên endian

TCVN 11167-12:2015

Offset	Trường	Kích cỡ	Giá trị	Mô tả
				Ví dụ: 3.58 MHz được mã hóa như giá trị nguyên 3580 (0000DFch) Điều này được dùng trong việc tính toán ETU và WWT. Tần số đồng hồ được dùng khi đọc dữ liệu ATR.
14	dwMaximumClock	4		Tần số điện thoại ICC được hỗ trợ tối đa theo kHz được mã hóa như giá trị nguyên Ví dụ: 14.32 MHz được mã hóa như giá trị nguyên 14320 (000037F0h).
18	bNumClockSupported	1		Số tần số đồng hồ được hỗ trợ bởi CCID. Nếu CCID không cho phép thiết lập thủ công của tần số đồng hồ thì giá trị này phải là 00h. Ngược lại, nếu CCID đặt thủ công và giá trị này là 00h, tần số đồng hồ hỗ trợ được giả định bởi tần số đồng hồ mặc định được định nghĩa bởi dwDefaultClock.
19	dwDataRate	4		Tỉ lệ dữ liệu I/O ICC mặc định theo bps được mã hóa như số nguyên endian nhỏ Ví dụ: 14.32 MHz được mã hóa là giá trị nguyên 14320. (000037F0h)
23	dwMaxDataRate	4		Tỉ lệ dữ liệu I/O ICC hỗ trợ tối đa theo bps. Ví dụ: 115.2Kbps được mã hóa là giá trị nguyên 115200. (0001C200h)
27	bNumDataRatesSupported	1		Số lượng tỉ lệ dữ liệu được hỗ trợ bởi CCID. Nếu CCID không cho phép thiết lập thủ công của tỉ lệ dữ liệu thì giá trị này phải là 00h. Ngược lại, nếu bộ đọc đặt thủ công và giá trị này là 00h, tỉ lệ dữ liệu hỗ trợ được giả định là tỉ lệ dữ liệu mặc định được định nghĩa theo ISO.
28	dwMaxIFSD	4		Chỉ ra IFSD tối đa được hỗ trợ bởi CCID đối với giao thức T=1.
32	dwSynchProtocols	4	RRRR PPPP = 000000	- RRRR - viết hoa - là RFU = 0000h - PPPP - viết thường- mã hóa các loại giao thức được hỗ trợ. Một '1' trong một vị trí bit cho sẵn chỉ ra hỗ trợ với giao thức

Offset	Trường	Kích cỡ	Giá trị	Mô tả
			00h	tương ứng. 0001h chỉ ra hỗ trợ với giao thức 2 dây ¹¹ 0002h chỉ ra hỗ trợ với giao thức 3 dây 0004h chỉ ra hỗ trợ với giao thức I2C Tất cả các giá trị khác nằm ngoài đặc tả này, phải được quản lý bởi trình điều khiển hỗ trợ bởi nhà cung cấp
36	dwMechanical	4	0000 0008h	See CHÚ THÍCH bên dưới. Giá trị là một thao tác bit đảo OR được thực hiện theo các giá trị sau: <ul style="list-style-type: none"> — 00000000h Không có đặc tính đặc biệt — 00000001h Cơ chế chấp nhận thẻ ²² — 00000002h Cơ chế từ chối thẻ ² — 00000004h Cơ chế nạp thẻ ² — 00000008h Cơ chế khóa/mở khóa thẻ — 00000010h ICC được chèn vĩnh viễn
40	dwFeatures	4		Giá trị này chỉ ra đặc trưng thông minh mà CCID có. Giá trị là một thao tác bit đảo OR được thực hiện theo các giá trị sau: <ul style="list-style-type: none"> — 00000000h Không có đặc tính đặc biệt — 00000002h Cấu hình thông số tự động dựa trên dữ liệu ATR — 00000004h Kích hoạt tự động việc chèn ICC — 00000008h Chọn điện thế ICC tự động — 00000010h Thay đổi tần số đồng hồ ICC tự động theo các thông số — 00000020h Thay đổi tỉ lệ baud tự động theo tần số và thông số FI, DI — 00000040h Trao đổi thông số tự động tạo bởi CCID (dùng cho các tái thiết lập ăm,

¹ Việc phát hành đặc tả không hỗ trợ các thiết bị với 2-, 3-dây và các giao thức I2C mà PPPP = 0000h. Trường này tương thích xuôi với đặc tả PCSC.

² Các cơ chế này của thông số *dwMechanical* *dwMechanica* đã gồm tính hoàn thiện, tuy nhiên các chức năng này của các CCID cơ giới hóa không được bao trùm bởi việc phát hành đặc tả. Một phát hành dự kiến có thể chuẩn hóa giao diện cho các tính năng cơ giới này.

Offset	Trường	Kích cỡ	Giá trị	Mô tả
				<p>lệnh hay PPS theo một thuật toán độc quyền của nhà sản xuất nhằm chọn các thông số kết nối với ICC)</p> <ul style="list-style-type: none"> - 00000080h PPS tự động, được tạo bởi CCID theo các thông số hiện tại - 00000100h CCID có thể đặt ICC theo chế độ dừng đồng hồ - 00000200h Giá trị NAD khác 00 được chấp nhận (giao thức T=1 được dùng) - 00000400h Trao đổi IFSD tự động như trao đổi đầu tiên (giao thức T=1 được dùng) <p>Chỉ một trong nhiều giá trị sau có thể có được:</p> <ul style="list-style-type: none"> - 00010000h Các trao đổi mức TPDU với CCID - 00020000h Trao đổi mức APDU ngắn với CCID - 00040000h Trao đổi mức APDU ngắn và APDU mở rộng với CCID <p>Chỉ một trong nhiều giá trị 00000040h và 00000080h có thể có.</p> <p>Khi giá trị 00000040h có được, máy chủ phải không cố gắng thay đổi FI, DI và giao thức được chọn hiện tại.</p> <p>Khi một trao đổi mức APDU được chọn, một trong nhiều giá trị 00000040h hay 00000080 phải có được, như giá trị 00000002h.</p>
44	dwMaxCCIDMessageLength	4		<p>Đối với mức APDU mở rộng, giá trị phải nằm giữa 261 +10 (tiêu đề) và 65544 +10, ngược lại giá trị tối thiểu là <i>dwMaxPacketSize</i> của điểm cuối bulk-OUT.</p>
48	bClassGetResponse	1		<p>Phép gán chỉ với CCID cung cấp một trao đổi mức APDU.</p> <p>Chỉ ra giá trị lớp mặc định được dùng bởi CCID khi nó gửi một lệnh Get Response nhằm thực hiện một truyền tải của một APDU bởi giao thức T=0.</p> <p>Giá trị FFh chỉ ra rằng CCID là lặp lại ớp APDU.</p>

Offset	Trường	Kích cỡ	Giá trị	Mô tả
49	bClassEnvelope	1		<p>Phép gán chỉ với CCID cung cấp một trao đổi mức APDU mở rộng.</p> <p>Chỉ ra giá trị lớp mặc định được dùng bởi CCID khi nó gửi một lệnh Envelope nhằm thực hiện một truyền tải của một APDU mở rộng bởi giao thức T=0.</p> <p>Giá trị FFh chỉ ra rằng CCID lặp lại lớp APDU.</p>
50	wLcdLayout	2	XXYYh	<p>Số lượng các dòng và ký tự đối với hiển thị LCD được dùng để gửi thông điệp cho đầu vào PIN.</p> <p>XX: số lượng dòng</p> <p>YY: số lượng kí tự theo dòng</p> <p>XXYY=0000h không có LCD.</p>
52	bPINSupport	1	00h - 03h	<p>Giá trị này chỉ ra PIN nào hỗ trợ các đặt trưng mà CCID có.</p> <p>Giá trị đó là một thao tác bit đảo OR được thực hiện theo các giá trị sau:</p> <p>01h Xác minh PIN được hỗ trợ</p> <p>02h Thay đổi PIN được hỗ trợ</p>
53	bMaxCCIDBusySlots	1	01h - 0Fh	Số lượng khe lớn nhất mà có thể bận đồng loạt

Thư mục tài liệu tham khảo

- [1] ISO 1177:1985, Information processing - Character structure for start/stop and synchronous character oriented transmission.
 - [2] ISO/IEC 10536, Identification cards - Contactless integrated circuit cards - Close coupled cards (tất cả các phần).
 - [3] ISO/IEC 14443, Identification cards - Contactless integrated circuit cards - Proximity cards (tất cả các phần).
 - [4] ISO/IEC 15693, Identification cards - Contactless integrated circuit cards - Vicinity cards (tất cả các phần).
-