



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR39 シリーズ スマートカードリーダー

リファレンスマニュアル V1.04



改定履歴

リリース日付	改訂説明	バージョン
2013-09-06	<ul style="list-style-type: none">初回発布	1.00
2014-03-25	<ul style="list-style-type: none">セクション 2.0 更新：特性タイトルを ACR39x から ACR39 シリーズに変更ブランド商標の属性追加USB インターフェイス説明セクションがスマートカードインターフェイスの前に移動された	1.01
2015-05-13	<ul style="list-style-type: none">セクション 2.0 更新：特性	1.02
2015-12-14	<ul style="list-style-type: none">セクション 2.0 更新：特性フォーマットの再構成セクション 9.1：PC_to_RDR_Xfrblock 経由でアクセスされるその他のコマンドの更新削除された付録 A - サポートされているカードの種類	1.03
2017-10-10	<ul style="list-style-type: none">ドキュメントのタイトル更新すべての ACR39x を ACR39 に改名セクション 2.0 更新：特性	1.04



目次

1.0.	紹介	5
1.1.	参照ファイル.....	5
1.2.	シンボルと略語.....	5
2.0.	特性	7
2.1.	ACR39U.....	7
2.2.	ACR39U PocketMate II.....	8
2.3.	ACR39T.....	9
3.0.	サポートしているスマートカード	11
3.1.	MCU カード.....	11
3.2.	メモリカード.....	11
4.0.	USB インターフェース	12
4.1.	通信パラメーター.....	12
4.2.	エンドポイント.....	12
5.0.	接触式スマートカードインターフェース	13
5.1.	スマートカード電源 VCC (C1).....	13
5.2.	プログラミング電圧 VPP (C6).....	13
5.3.	カードタイプのセレクション.....	13
5.4.	マイクロコントローラベースカードのためのインターフェース.....	13
5.5.	カード引き裂き保護.....	13
6.0.	電源パラメーター	14
6.1.	LED ステータスインジケータ.....	14
7.0.	USB 通信プロトコル	15
7.1.	CCID Bulk-OUT メッセージ.....	17
7.1.1.	PC_to_RDR_IccPowerOn.....	17
7.1.2.	PC_to_RDR_IccPowerOff.....	17
7.1.3.	PC_to_RDR_GetSlotStatus.....	18
7.1.4.	PC_to_RDR_XfrBlock.....	18
7.1.5.	PC_to_RDR_GetParameters.....	18
7.1.6.	PC_to_RDR_ResetParameters.....	19
7.1.7.	PC_to_RDR_SetParameters.....	19
7.2.	CCID Bulk-IN メッセージ.....	23
7.2.1.	RDR_to_PC_DataBlock.....	23
7.2.2.	RDR_to_PC_SlotStatus.....	23
7.2.3.	RDR_toPC_Parameters.....	24
8.0.	メモリカードのコマンドセット	26
8.1.	メモリカード – 1、2、4、8 および 16 kilobit I2C カード.....	26
8.1.1.	SELECT_CARD_TYPE.....	26



8.1.2.	SELECT_PAGE_SIZE	26
8.1.3.	READ_MEMORY_CARD.....	27
8.1.4.	WRITE_MEMORY_CARD	27
8.2.	メモリカード – 32、64、128、256、512 および 1024 kilobit I2C カード	28
8.2.1.	SELECT_CARD_TYPE	28
8.2.2.	SELECT_PAGE_SIZE	28
8.2.3.	READ_MEMORY_CARD.....	29
8.2.4.	WRITE_MEMORY_CARD	30
8.3.	メモリカード – SLE4418/SLE4428/SLE5518/SLE5528	31
8.3.1.	SELECT_CARD_TYPE	31
8.3.2.	READ_MEMORY_CARD.....	31
8.3.3.	READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE4428 と SLE5528 のみ).....	32
8.3.4.	READ_PROTECTION_BIT	32
8.3.5.	WRITE_MEMORY_CARD	33
8.3.6.	WRITE_PROTECTION_MEMORY_CARD	34
8.3.7.	PRESENT_CODE_MEMORY_CARD (SLE4428 と SLE5528 のみ).....	35
8.4.	メモリカード – SLE4432/SLE4442/SLE5532/SLE5542	35
8.4.1.	SELECT_CARD_TYPE	35
8.4.2.	READ_MEMORY_CARD.....	36
8.4.3.	READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE4442 と SLE5542 のみ).....	37
8.4.4.	READ_PROTECTION_BITS	37
8.4.5.	WRITE_MEMORY_CARD	38
8.4.6.	WRITE_PROTECTION_MEMORY_CARD	39
8.4.7.	PRESENT_CODE_MEMORY_CARD (SLE4442 と SLE5542 のみ).....	39
8.4.8.	CHANGE_CODE_MEMORY_CARD (SLE4442 と SLE5542 のみ).....	40
9.0.	PC_to_RDR_XfrBlock を介して、他のコマンドを実行する	41
9.1.	GET_READER_INFORMATION	41
付録 A.	応答エラーコード.....	42

テーブル一覧表

表 1	: シンボルと略語.....	6
表 2	: USB インターフェース配線	12
表 3	: 応答エラーコード	42

1.0. 紹介

ACR39PC リンクのスマートカードリーダーは PC（個人のパソコン）とスマートカードの間の通信インターフェースです。異なるタイプのスマートカードは異なるコマンドと通信プロトコルを採用しているため、ほとんどの場合、スマートカードとコンピュータの間に直接通信できません。ACR39 が様々のカードにコンピュータから統一されたインタフェースを確立します。スマートカードのたくさんの特性を持っているため、コンピュータ・ソフトウェア・プログラマがスマートカードの操作を詳しく了解する必要をなくなります。多くの場合、これらの操作の技術的詳細はスマートカードシステムの実装と関係がありません。

1.1. 参照ファイル

下記のファイルは www.usb.org でダウンロードできます。

- 《ユニバーサル・シリアル・バス仕様 2.0》（即ち USB 仕様）、2000 年 4 月 27 日
- 《ユニバーサル・シリアル・バス共通クラス仕様 1.0》、1997 年 12 月 16 日
- 《ユニバーサル・シリアル・バス・デバイス・クラス：集積回路（S）カード・インターフェース・デバイス用のスマートカード CCID 仕様 1.1》、2005 年 4 月 22 日

下記のファイルは www.ansi.org でオーダーできます。

- 《ISO/IEC 7816-1：識別カード - 接点付きの集積回路（S）カード - パート 1：物理特性》
- 《ISO/IEC 7816-2：識別カード - 接点付きの集積回路（S）カード - パート 2：接点のサイズと場所》
- 《ISO/IEC 7816-3：識別カード - 接点付きの集積回路（S）カード - パート 3：電子信号および伝送プロトコル》

1.2. シンボルと略語

略語	説明
ATR	リセット応答 (Answer-To-Reset)
CCID	チップ/スマートカードインターフェースデバイス (Chip/Smart Card Interface Device)
ICC	集積回路カード (Integrated Circuit Cards)
IFSC	プロトコル T=1 の集積回路カードの情報フィールドのサイズ (Information Field Sized for ICC for protocol T=1)



略語	説明
IFSD	プロトコル T=1 のチップ/スマートカードインターフェースデバイスの情報フィールドのサイズ (Information Field Sized for ICC for protocol T=1)
NAD	ノードの場所 (Node Address)
PPS	プロトコルおよびパラメーターセレクション (Protocol and Parameters Selection)
RFU	将来に使用のために保留 (Reserved for Future Use ¹)
TPDU	転送プロトコルデータユニット (Application Protocol Data Unit)
USB	ユニバーサル・シリアル・バス (Universal Serial Bus)

表1 : シンボルと略語

¹ 特別の説明がなければ、0 に設置しなければなりません。



2.0. 特性

2.1. ACR39U

- USB フルスピード・インターフェース
- プラグアンドプレイ-CCID 準拠、高い柔軟性を持っている
- スマートカードリーダー：
 - 接触式インターフェース
 - ISO 7816 クラス A、B、C の（5V、3V および 1.8V）カードをサポート
 - 共通の特権カード（CAC）サポート CAC
 - SIPRNET カードサポート
 - J-LIS カードサポート
 - T = 0 または T = 1 プロトコルのマイクロプロセッサカードサポート
 - メモリカードサポート
 - PPS サポート（プロトコルとパラメータの選択）
 - 短絡保護保有
- アプリケーション プログラミング インターフェース
 - PC/SC サポート
 - （PC / SC の上のラッパー経由で）、CT-API をサポート
- Android™ 3.1 と以降のバージョンサポートしている²
- 以下の規格に準拠：
 - EN60950/IEC 60950
 - ISO 7816
 - EMV™ Level 1 (接触式)
 - PC/SC
 - CCID
 - CE
 - FCC
 - WEEE
 - RoHS 2
 - REACH
 - FIPS 201 (アメリカ)

² ACS 定義の Android ライブラリを使用

*USB タイプ A および USB タイプ C コネクタで利用可能



- TAA (アメリカ)
- J-LIS (日本)
- VCCI (日本)
- PBOC (中国)
- Microsoft® WHQL

2.2. ACR39U PocketMate II

- USB フルスピード・インターフェース*
- プラグアンドプレイーCCID 準拠、高い柔軟性を持っている
- 旋回式デザイン
- スマートカードリーダー：
 - 接触式インターフェース
 - ISO 7816 クラス A、B、C の（5V、3V および 1.8V）カードをサポート
 - 共通の特権カード（CAC）サポート CAC
 - SIPRNET カードサポート
 - J-LIS カードサポート
 - T = 0 または T = 1 プロトコルのマイクロプロセッサカードサポート
 - メモリカードサポート
 - PPS サポート（プロトコルとパラメータの選択）
 - 短絡保護保有
- アプリケーション プログラミング インターフェース
 - PC/SC サポート
 - （PC / SC の上のラッパー経由で）、CT- API をサポート
- Android™ 3.1 と以降のバージョンサポートしている³
- 以下の規格に準拠：
 - EN60950/IEC 60950
 - ISO 7816
 - EMV™ Level 1 (接触式)
 - PC/SC
 - CCID

³ ACS 定義の Android ライブラリを使用

*USB タイプ A、Micro-B および USB Type C コネクタで利用可能



- CE
- FCC
- WEEE
- RoHS 2
- REACH
- FIPS 201 (アメリカ)
- TAA (アメリカ)
- J-LIS (日本)
- VCCI (日本)
- PBOC (中国)
- Microsoft® WHQL

2.3. ACR39T

- USB フルスピード・インターフェース
- プラグアンドプレイーCCID 準拠、高い柔軟性を持っている
- 保護 USB キャップ保有
- スマートカードリーダー：
 - 接触式インターフェース
 - ISO 7816 クラス A、B、C の（5V、3V および 1.8V）カードをサポート
 - T = 0 または T = 1 プロトコルのマイクロプロセッサカードをサポート
 - メモリカードサポート
 - PPS サポート（プロトコルとパラメータの選択）
 - 短絡保護保有
 - アプリケーション プログラミング インターフェース
 - PC/SC サポート
 - （PC / SC の上のラッパー経由で）、CT- API をサポート
- Android™ 3.1 と以降のバージョンサポートしている⁴
- 以下の規格に準拠：
 - EN 60950/IEC 60950
 - ISO 7816
 - PC/SC

⁴ ACS 定義の Android ライブラリを使用

*USB Micro-B および USB タイプ C コネクタで利用可能



- CCID
- CE
- FCC
- WEEE
- RoHS 2
- REACH
- VCCI (日本)
- Microsoft® WHQL

3.0. サポートしているスマートカード

3.1. MCU カード

ACR39 は PC/SC 仕様に準拠しているため、ISO 7816 仕様の A、B と C タイプのスマートカード（5V、3 V および 1.8 V）サポートだけでなく、T=0 または T=1 プロトコル仕様の全ての MCU カードもサポートします。

カードの ATR が専用の操作モードを指定すれば（T A 2 が存在している；TA2 のビット 5 は 0 でなければなりません）、しかし ACR39 がこのモードをサポートできない場合、ACR39 はカードをリセットして、交渉モードに設置します。交渉モードを設置できないと、ACR39 がこのカードを拒否する。

カードの ATR が交渉のモード（TA2 が存在指定ない）および通信パラメータ（デフォルトパラメータじゃなくて）を指定すれば、ACR39 がその通信パラメータを使用してみ、PPS を実行します。ACR32 が PPS を拒否したら、デフォルトパラメータを使用する（F=372, D=1）。

注：上記のパラメータの意味について、ISO 7816-3 仕様を参照してください。

3.2. メモリカード

ACR39 が様々のメモリカードをサポートしている、例：

- I2C バスプロトコルに準拠し、一回で 128 バイト/ページを書くことができるメモリカード（フリーメモリカード）、以下を含めて：
 - Atmel®：AT24C01/02/04/08/16/32/64/128/256/512/1024
 - SGS-Thomson：ST14C02C、ST14C04C
 - Gemplus：GFM1K、GFM2K、GFM4K、GFM8K
- 書き込み保護機能付インテリジェント 256 バイトの EEPROM カード、以下を含めて：
 - Infineon®：SLE4418、SLE4428、SLE5518 および SLE5528
- インテリジェント 256 バイトの EEPROM、書き込みのカードプロテクト機能付カード、以下を含めて：
 - Infineon®：SLE4432、SLE4442、SLE5532 および SLE5542

4.0. USB インターフェース

4.1. 通信パラメーター

ACR39 は USB 2.0 準拠の USB インターフェースを介して、PC と接続します。USB 全速モードをサポートして、速度は 12 Mbps です。

ピン	信号	機能
1	V _{BUS}	カードに+5 V の電源を提供します。
2	D-	ACR39 と PC は差動信号でデータを転送します。
3	D+	ACR39 と PC は差動信号でデータを転送します。
4	GND	電源用の参照電圧レベル

表2 : USB インターフェース配線

4.2. エンドポイント

ACR39 が下記のエンドポイントを介して、ホストの PC と通信します：

Control Endpoint 設置と制御のため

Bulk-OUT ホストから ACR39 (CCID) に送信するコマンドに対して
(データパケットサイズは 64 バイト)

Bulk-IN ACR39 からホストに返す応答に対して
(ペイロードの大きさは 64 バイトです)

Interrupt-IN ACR39 からホストに送信する状態メッセージに対して
(ペイロードの大きさは 8 バイトです)

5.0. 接触式スマートカードインターフェース

ACR39 と挿入されたカードの間のインターフェースが ISO 7816-3 仕様プロトコルに準拠しています。ACR39 の実用的な機能性を高めるために一定の制限や機能拡張をします。

5.1. スマートカード電源 VCC (C1)

挿入されたカードの消費電流は 50mA よりも高くはありません。

5.2. プログラミング電圧 VPP (C6)

ISO7816-3 仕様によると、スマートカードコンタクト C6 (VPP) がスマートカードにプログラミング電圧を供給する。市場内のすべてのスマートカードが EEPROM ベースであり、外部プログラミング電圧の供給の必要がないです。ACR39 のコンタクト C6 (VPP) が通常の制御信号として実装されました。このコンタクトの電気仕様は信号 RST (コンタクト C2) の電気仕様と同じです。

5.3. カードタイプのセレクション

制御 PC は、挿入されたカードをアクティブする前に、ACR39 に適切なコマンドを送信してカードタイプを選択する必要があります。メモリーカードと MCU ベースカードも含めています。

MCU ベースカードに対して、リーダーが T=0 または T=1 中から望ましいプロトコルを選定します。しかしながら、挿入されたカードは両方のプロトコルタイプをサポートできる場合は、リーダーが PPS を通じて、このセレクションを受け入れられて、実行します。プログラマベースのカードは、1 つだけのプロトコルタイプ (T=0 または T=1) をサポートする時に、アプリケーションがどのプロトコルを選ぶことと関係なく、リーダーは自動的にこのプロトコルタイプを選択します。

5.4. マイクロコントローラベースカードのためのインターフェース

マイクロコントローラベースカードは C1 (VCC) 、C2 (RST)、C3 (CLK) 、C5 (GND) および C7 (I/O) これらのコンタクトだけ使用します。4.8 MHz の周波数が CLK 信号 (C3) に適用します。

5.5. カード引き裂き保護

電気入れる状態で、急に引き出されたカードを保護するために、ACR39 がメカニズムを提供している。カードが取り外されている時、ACR39 とカード間の信号線への電力供給がすぐに非アクティブ化されます。原則として、電氣的な損傷を回避するために、パワーダウンしてから、カードをリーダーから除去されるべきです。

注： ACR39 は決してそれ自体で挿入されたカードへの電源供給に切り替わりません。この操作は制御 PC がリーダーに適切なコマンドを送信しか実行できません。



6.0. 電源パラメーター

ACR39 は 5 、 100 mA の安定化電源の電圧が必要で、PC で電源を供給します（各タイプのリーダーと共に供給されたケーブルを介して）。

6.1. LED ステータスインジケータ

LED はスマートカードインタフェースの活性化状態を示しています：

- **ゆっくり点滅する（2 秒ごとに 200ms にオンする）**

ACR39 は電気入れられて、スタンバイ状態に入ります。いずれかのスマートカードが挿入されていないか、または（それが挿入されている場合）、スマートカードは、電源が投入されていません。

- **点灯**

スマートカードへの電源がオンにされている、即ちカードが活性化状態に入ります。

- **速く点滅する**

ACR39 とカードが通信しています。

7.0. USB 通信プロトコル

ACR39 は USB を介して、ホストとのインターフェースを確立します。業界の規範 - CCID 標準は、USB チップ - スマートカードインタフェース装置に関わっているプロトコルを定義します。CCID 仕様はスマートカードを動作させるために必要な全てのプロトコルをカバーしています。

ACR39 の USB エンドポイントの装置と使用は CCID 標準の Rev 1.0 のパート 3 に準拠するはずですが、

概要を以下に要約されています：

1. 制御コマンドは制御パイプ（デフォルトのパイプ）で送信されます。特定な請求と USB 規格の請求を含めます。デフォルトのパイプで送信されたコマンドはデフォルトのパイプでホストにレポート情報を返します。
2. CCID イベント割り込みパイプで送信されます。
3. CCID コマンド BULK-OUT エンドポイントで送信されます。ACR39 に送信された全てのコマンドは自分の関係エンディング応答を持っています。いくつかのコマンドは中間応答も持っています。
4. CCID 応答 BULK-IN エンドポイントで送信されます。ACR39 に送信された全てのコマンドは必ず同期に送信されます。（例：ACR39 にとって、bMaxCCIDBusySlots は 01 h に相当です）。

ACR39 がサポートしている CCID 特性は下記のクラス記述子を参照します：

オフセット	データフィールド	大きさ	数値	説明
0	<i>bLength</i>	1		記述子のバイト。
1	<i>bDescriptorType</i>	1		CCID 機能記述子のタイプ。
2	<i>bcdCCID</i>	2		進化した 10 進数での CCID 仕様のリリース番号。
4	<i>bMaxSlotIndex</i>	1		ACR39 は一つのスロットを持っています。
5	<i>bVoltageSupport</i>	1		ACR39 が 1.8 V、3 V と 5 V のスロット電圧サポート。
6	<i>dwProtocols</i>	4		ACR39 は T=0 および T=1 プロトコルをサポートできる。
10	<i>dwDefaultClock</i>	4		デフォルトの ICC クロック周波数は 4.8 MHz です。
14	<i>dwMaximumClock</i>	4		ICC サポートできるクロック周波数は 4.8 MHz です。
18	<i>bNumClockSupported</i>	1		クロック周波数の手動設定をサポートしていません。



オフセット	データフィールド	大きさ	数値	説明
19	<i>dwDataRate</i>	4		デフォルトの ICC I/O ボーレートは 12918 bps です。
23	<i>dwMaxDataRate</i>	4		ICC I/O サポートできる最大ボーレートは 826 kbps です。
27	<i>bNumDataRatesSupported</i>	1		ボーレートの手動設定をサポートしていません。
28	<i>dwMaxIFSD</i>	4		ACR39 T1 サポートできる最大 IFSD は 247 です。
32	<i>dwSynchProtocols</i>	4		ACR39 は同期カードをサポートできません。
36	<i>dwMechanical</i>	4		ACR39 は特別な機械的特性をサポートできません。
40	<i>dwFeatures</i>	4		ACR39 は下記の特性をサポートできます： <ul style="list-style-type: none">• パラメーターに応じて、自動的に ICC のクロック周波数を変更します• 周波数に応じて、自動的 FI、DI パラメーターとボーレートを変更します• ACR39 との TPDU レベル変更
44	<i>dwMaxCCIDMessageLength</i>	4		ACR39 が受け入れられる最大なメッセージの長さは 271 バイトです。
48	<i>bClassGetResponse</i>	1		TPDU レベル交換に影響しません。
49	<i>bClassEnvelope</i>	1		TPDU レベル交換に影響しません。
50	<i>wLCDLayout</i>	2		LCD なし。
52	<i>bPINSupport</i>	1		PIN 認証サポート。
53	<i>bMaxCCIDBusySlots</i>	1		唯一の 1 スロットを同時に忙しくすることができます。

7.1. CCID Bulk-OUT メッセージ

ACR39 は CCID プロトコルのパート 4.1 が定義している Bulk-OUT メッセージを準拠するはずですが、また、この仕様はいくつかの追加機能用の拡張コマンドを定義しています。

本節は ACR39 がサポートしている CCID タイプの Bulk-OUT メッセージをリストします。

7.1.1. PC_to_RDR_IccPowerOn

このコマンドはスロットを活性化して、カードから ATR を返すために使われます。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	62h	
1	<i>dwLength</i>	4	00000000h	メッセージの余分なバイトのサイズ。
2	<i>bSlot</i>	1		このコマンドのスロット番号を識別します。
5	<i>bSeq</i>	1		コマンドのシーケンス番号。
6	<i>bPowerSelect</i>	1		ICC に印加される電圧： 00h - 自動電圧選択 01h - 5 V 02h - 3 V
7	<i>abRFU</i>	2		保留して将来使います。

このコマンドメッセージの応答は *RDR_to_PC_DataBlock* 応答メッセージです。返したデータはリセット応答 (ATR) です。

7.1.2. PC_to_RDR_IccPowerOff

スロットの活性化をキャンセルする時、このコマンドを使います。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	63h	
1	<i>dwLength</i>	4	00000000h	メッセージの余分なバイトのサイズ。
5	<i>bSlot</i>	1		このコマンドのスロット番号を識別します。
6	<i>bSeq</i>	1		コマンドのシーケンス番号。
7	<i>abRFU</i>	3		保留して将来使います。

このメッセージの応答は *RDR_to_PC_SlotStatus* メッセージです。

7.1.3. PC_to_RDR_GetSlotStatus

現在のスロットの状態情報を取得する時に、このコマンドを使います。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	65h	
1	<i>dwLength</i>	4	00000000h	メッセージの余分なバイトのサイズ。
5	<i>bSlot</i>	1		このコマンドのスロット番号を識別します。
6	<i>bSeq</i>	1		コマンドのシーケンス番号。
7	<i>abRFU</i>	3		保留して将来使います。

このメッセージの応答は *RDR_to_PC_SlotStatus* メッセージです。

7.1.4. PC_to_RDR_XfrBlock

ICC にデータブロックを転送する時にこのコマンドを使用します。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	6Fh	
1	<i>dwLength</i>	4		このメッセージの <i>abData</i> データフィールドのサイズ
5	<i>bSlot</i>	1		このコマンドのスロット番号を識別します。
6	<i>bSeq</i>	1		コマンドのシーケンス番号。
7	<i>bBWI</i>	1		CCID ブロックを転送するタイムアウト時間を拡張する時に使用します。「この数にブロック待機時間を掛け」の期限が切れた後、CCID はブロックをタイムアウトします。
8	<i>wLevelParameter</i>	2	0000h	RFU (TPDU 交換レベル)。
10	<i>abData</i>	バイト配列		CCID に送信されるデータブロック。データは ICC に「そのまま」送信されます。

このメッセージの応答は *RDR_to_PC_DataBlock* メッセージです。

7.1.5. PC_to_RDR_GetParameters

スロットのパラメーターを取得する時にこのコマンドを使用します。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	6Ch	

オフセット	データフィールド	大きさ	数値	説明
1	<i>dwLength</i>	4	00000000h	メッセージの余分なバイトのサイズ。
5	<i>bSlot</i>	1		このコマンドのスロット番号を識別します。
6	<i>bSeq</i>	1		コマンドのシーケンス番号。
7	<i>abRFU</i>	3		保留して将来使います。

このメッセージの応答は *RDR_to_PC_Parameters* メッセージです。

7.1.6. PC_to_RDR_ResetParameters

スロットのパラメーターをデフォルト値に戻す時このコマンドを使用します。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	6Dh	
1	<i>dwLength</i>	4	00000000h	メッセージの余分なバイトのサイズ。
5	<i>bSlot</i>	1		このコマンドのスロット番号を識別します。
6	<i>bSeq</i>	1		コマンドのシーケンス番号。
7	<i>abRFU</i>	3		保留して将来使います。

このメッセージの応答は *RDR_to_PC_Parameters* メッセージです。

7.1.7. PC_to_RDR_SetParameters

スロットのパラメーターを設置する時にこのコマンドを使用します。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	61h	
1	<i>dwLength</i>	4		メッセージの余分なバイトのサイズ。
5	<i>bSlot</i>	1		このコマンドのスロット番号を識別します。
6	<i>bSeq</i>	1		コマンドのシーケンス番号。

オフセット	データフィールド	大きさ	数値	説明
7	<i>bProtocolNum</i>	1		<p>下記は指定されたプロトコルのデータ構造です。</p> <p>00h – T=0 プロトコル構造</p> <p>01h – T=1 プロトコル構造</p> <p>下記の値を保留して将来使います：</p> <p>80h – 2線プロトコルの構造</p> <p>81h – 3線プロトコルの構造</p> <p>82h = I2C プロトコル構造</p>
8	<i>abRFU</i>	2		保留して将来使います。
10	<i>abProtocolDataStructure</i>	バイト配列		プロトコルのデータ構造

T=0 プロトコルのデータ構造 (*dwLength*=00000005h)

オフセット	データフィールド	大きさ	数値	説明
10	<i>bmFindexDindex</i>	1		<p>B7-4 – FI – ISO/IEC 7816-3:1997 中のテーブル 7 をインデックスして、クロックレートの変換係数を選択します。</p> <p>B3-0 – DI - ISO/IEC 7816-3:1997 中のテーブル 8 をインデックスして、ポーレートの変換係数を選択します</p>
11	<i>bmTCCKST0</i>	1		<p>B0 – 0b, B7-2 – 000000b</p> <p>B1 – 使用している約束 (b1=0 : 直接 ; b1=1 : 逆)</p> <p>注 : <i>CCID</i> がこのビットを無視します。</p>
12	<i>bGuardTimeT0</i>	1		文字間の余分な GuardTime。正常な GuardTime (12 ETU) に 0–254 ETU を追加します。FFh と 00h が同じです。
13	<i>bWaitingIntegerT0</i>	1		T=0 の場合 WI が WWT を定義する時に使われます

オフセット	データフィールド	大きさ	数値	説明
14	<i>bClockStop</i>	1		ICC クロック停止サポート： 00h - クロックを停止することは許可されていません 01h - クロック信号が低い時に停止されます 02h - クロック信号が高い時に停止されます 03h - クロック信号が低い時または高い時に停止されます

T=1 プロトコルのデータ構造 (dwLength=00000007h)

オフセット	データフィールド	大きさ	数値	説明
10	<i>bmFindexDindex</i>	1		B7-4 – FI – ISO/IEC 7816-3:1997 中のテーブル 7 をインデックスして、クロックレートの変換係数を選択します B3-0 – DI - ISO/IEC 7816-3:1997 中のテーブル 8 をインデックスして、ポーレートの変換係数を選択します
11	<i>bmTCCKST1</i>	1		B7-2 – 000100b B0 – チェックサムタイプ (b0=0 : LRC ; b0=1 : CRC) B1 – 使用している約束 (b1=0 : 直接 ; b1=1 : 逆) 注 : CCID がこのビットを無視します。
12	<i>bGuardTimeT1</i>	1		余計の保護時間 (二つのキャラクタ間は 0–254etu) 。値は FFh である場合、保護時間が 1etu 減らします。
13	<i>bWaitingIntegerT1</i>	1		B7-4 – BWI 値 0-9h 有効 B3-0 = CWI 値 0-Fh 有効



オフセット	データフィールド	大きさ	数値	説明
14	<i>bClockStop</i>	1		ICC クロック停止サポート： 00h - クロックを停止することは許可されていません 01h - クロック信号が低い時に停止されます 02h - クロック信号が高い時に停止されます 03h - クロック信号が低い時または高い時に停止されます
15	<i>bIFSC</i>	1		交渉された IFSC の大きさ。
16	<i>bNadValue</i>	1	00h	NAD = 00h だけサポートできます。

このメッセージの応答は *RDR_to_PC_Parameters* メッセージです。

7.2. CCID Bulk-IN メッセージ

Bulk-IN メッセージは Bulk-OUT メッセージに回答をするため使われます。ACR39 は CCID プロトコルのパート 4 が定義している Bulk-IN メッセージを準拠するはずですが。

本節は ACR39 がサポートしている CCID タイプの Bulk-IN メッセージをリストします。

7.2.1. RDR_to_PC_DataBlock

このコマンドは ACR39 によって送信されて、*PC_to_RDR_IccPowerOn* と *PC_to_RDR_XfrBlock* メッセージに対する応答です。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	80h	CCID によってデータブロックを送信しています。
1	<i>dwLength</i>	4		メッセージの余分なバイトのサイズ。
5	<i>bSlot</i>	1		Bulk-OUT メッセージ中の値と同じです。
6	<i>bSeq</i>	1		Bulk-OUT メッセージ中の値と同じです。
7	<i>bStatus</i>	1		CCID 規格 (Rev 1.0) 4.2.1 節に定義されたスロットステータスレジスタ
8	<i>bError</i>	1		CCID 規格 (Rev 1.0) 4.2.1 節に定義されたスロットステータスレジスタ
9	<i>bChainParameter</i>	1	00h	RFU (TPDU 交換レベル)。
10	<i>abData</i>	バイト配列		このデータフィールドは CCID から返したデータを含めています。

7.2.2. RDR_to_PC_SlotStatus

このコマンドは ACR39 によって送信されて、*PC_to_RDR_IccPowerOff* と *PC_to_RDR_GetSlotStatus* メッセージに対する応答です。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	81h	
1	<i>dwLength</i>	4	00 00 00 00h	メッセージの余分なバイトのサイズ。
5	<i>bSlot</i>	1		Bulk-OUT メッセージ中の値と同じです。
6	<i>bSeq</i>	1		Bulk-OUT メッセージ中の値と同じです。

オフセット	データフィールド	大きさ	数値	説明
7	<i>bStatus</i>	1		CCID 規格 (Rev 1.0) 4.2.1 節に定義されたスロットステータスレジスタ
8	<i>bError</i>	1		CCID 規格 (Rev 1.0) 4.2.1 節に定義されたスロットステータスレジスタ
9	<i>bClockStatus</i>	1		数値 : 00h - 動作している 01h - L 状態に止まっている 02h - H 状態に止まっている 03h - 不明な状態に止まっている 残された値を保留して将来使います。

7.2.3. RDR_toPC_Parameters

このコマンドは ACR39 によって送信されて、*PC_to_RDR_GetParameters*、*PC_to_RDR_ResetParameters* および *PC_to_RDR_SetParameters* メッセージに対する応答です。

オフセット	データフィールド	大きさ	数値	説明
0	<i>bMessageType</i>	1	82h	
1	<i>dwLength</i>	4		メッセージの余分なバイトのサイズ。
5	<i>bSlot</i>	1		Bulk-OUT メッセージ中の値と同じです。
6	<i>bSeq</i>	1		Bulk-OUT メッセージ中の値と同じです。
7	<i>bStatus</i>	1		CCID 規格 (Rev 1.0) 4.2.1 節に定義されたスロットステータスレジスタ
8	<i>bError</i>	1		CCID 規格 (Rev 1.0) 4.2.1 節に定義されたスロットステータスレジスタ



オフセット	データフィールド	大きさ	数値	説明
9	<i>bProtocolNum</i>	1		下記は指定されたプロトコルのデータ構造です。 00h – T=0 プロトコル構造 01h – T=1 プロトコル構造 下記の値を保留して将来使います： 80h – 2線プロトコルの構造 81h – 3線プロトコルの構造 82h = I2C プロトコル構造
10	<i>abProtocolDataStructure</i>	バイト配列		プロトコルのデータ構造

8.0. メモリカードのコマンドセット

PC_to_RDR_XfrBlockコマンドでメモリカードをアクセスできます。メモリカードの全ての機能が pseudo-APDU にマップングされます。

8.1. メモリカード – 1、2、4、8 および 16 kilobit I2C カード

8.1.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択されたカードにパワーダウン/アップを実行し、同時にリセットを実行する時に使われます。

注釈： SCardConnect() API によって確立されたロジックなスマートカードリーダー通信後に使用しかできません。SCardConnect() API についての詳しい説明は PC/SC 基準を参照してください。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	01h

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

SW1	SW2

その中：

SW1 SW2 = 90 00h (エラーなしの場合)

8.1.2. SELECT_PAGE_SIZE

このコマンドはスマートカードを読み取られるページサイズを選択します。デフォルト値は8バイトの書き込みページ。カードが削除されているか、またはリーダーの電源がオフになっている時に、デフォルト値にリセットされる。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Page Size
FFh	01h	00h	00h	01h	

その中：

Page size = 03h : 8バイトの書き込みページ

- = 04h : 16 バイトの書き込みページ
- = 05h : 32 バイトの書き込みページ
- = 06h : 64 バイトの書き込みページ
- = 07h : 128 バイトの書き込みページ

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

SW1	SW2

その中 :

SW1 SW2 = 90 00h (エラーなしの場合)

8.1.3. READ_MEMORY_CARD

コマンドフォーマット (*PC_to_RDR_XfrBlock* 中の *abData* データフィールド)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B0h			

その中 :

Byte Address メモリカードのメモリアドレス位置

MEM_L メモリカードから読み出されていないデータの長さ

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

BYTE 1	...	BYTE N	SW1	SW2

その中 :

BYTE x メモリカードから読み出されたデータ

SW1 SW2 = 90 00h (エラーなしの場合)

8.1.4. WRITE_MEMORY_CARD

コマンドフォーマット (*PC_to_RDR_XfrBlock* 中の *abData* データフィールド)

Pseudo-APDU							
CLA	INS	Byte Address		MEM_L	BYTE 1	...	BYTE N
		MSB	LSB				
FFh	D0h						

その中：

Byte Address	メモ리카ードのメモリアドレス位置
MEM_L	メモ리카ードから読み出されていないデータの長さ
BYTE X	メモ리카ードに書き入っていないデータ

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

SW1	SW2

その中：

SW1 SW2 = 90 00h (エラーなしの場合)

8.2. メモ리카ード – 32、64、128、256、512 および 1024 kilobit I2C カード

8.2.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択されたカードにパワーダウン/アップを実行し、同時にリセットを実行する時に使われます。

注釈： *SCardConnect()* API によって確立されたロジックなスマートカードリーダー通信後に使用しかできません。
SCardConnect() API についての詳しい説明は *PC/SC* 基準を参照してください。

コマンドフォーマット (*PC_to_RDR_XfrBlock* 中の *abData* データフィールド)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	02h

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

SW1	SW2

その中：

SW1 SW2 = 90 00h (エラーなしの場合)

8.2.2. SELECT_PAGE_SIZE

このコマンドはスマートカードを読み取られるページサイズを選択します。デフォルト値は 8 バイトの書き込みページ。カードが削除されているか、またはリーダーの電源がオフになっている時に、デフォルト値にリセットされる。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Page Size
FFh	01h	00h	00h	01h	

その中 :

- Data** カードに送信されていない TPDU
- Page size** = 03h : 8 バイトの書き込みページ
 = 04h : 16 バイトの書き込みページ
 = 05h : 32 バイトの書き込みページ
 = 06h : 64 バイトの書き込みページ
 = 07h : 128 バイトの書き込みページ

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

SW1	SW2

その中 :

- SW1 SW2** = 90 00h (エラーなしの場合)

8.2.3. READ_MEMORY_CARD

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh				

その中 :

- INS** = B0h : 32、64、128、256 および 512 kilobit IIC カード
 = 1011 000*b : 1024 kilobit IIC カード
 その中 * はアドレッシング 17 ビットの MSB を示している。
- Byte Address** メモリカードのメモリアドレス位置
- MEM_L** メモリカードから読み出されていないデータの長さ

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

BYTE 1	...	BYTE N	SW1	SW2

その中 :

BYTE x メモリカードから読み出されたデータ

SW1 SW2 = 90 00h (エラーなしの場合)

8.2.4. **WRITE_MEMORY_CARD**

コマンドフォーマット (*PC_to_RDR_XfrBlock* 中の *abData* データフィールド)

Pseudo-APDU							
CLA	INS	Byte Address		MEM_L	BYTE 1	...	BYTE N
		MSB	LSB				
FFh							

その中 :

INS = D0h : 32、64、128、256 および 512 kilobit IIC カード

 = 1101 000*b : 1024 kilobit IIC カード

 その中 * はアドレッシング 17 ビットの MSB を示している。

Byte Address メモリカードのメモリアドレス位置

MEM_L メモリカードから読み出されていないデータの長さ

BYTE X メモリカードに書き入っていないデータ

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

SW1	SW2

その中 :

SW1 SW2 = 90 00h (エラーなしの場合)

8.3. メモリカード – SLE4418/SLE4428/SLE5518/SLE5528

8.3.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択されたカードにパワーダウン/アップを実行し、同時にリセットを実行する時に使われます。

注釈： SCardConnect() API によって確立されたロジックなスマートカードリーダー通信後に使用しかできません。SCardConnect() API についての詳しい説明は PC/SC 基準を参照してください。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	05h

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

SW1	SW2

その中：

SW1 SW2 = 90 00h (エラーなしの場合)

8.3.2. READ_MEMORY_CARD

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B0h			

その中：

MSB Byte Address = 0000 00A9A8b はメモリカードのメモリアドレス位置である

LSB Byte Address = A7A6A5A4 A3A2A1A0b はメモリカードのメモリアドレス位置である

MEM_L = メモリカードから読み出されていないデータの長さ

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

BYTE 1	...	BYTE N	SW1	SW2

その中 :

- BYTE x** メモリカードから読み出されたデータ
- SW1 SW2** = 90 00h (エラーなしの場合)

8.3.3. **READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE4428 と SLE5528 のみ)**

このコマンドがプレゼンテーションエラーカウンタを読み取る時に使われる。

コマンドフォーマット (*PC_to_RDR_XfrBlock* 中の *abData* データフィールド)

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	B1h	00h	00h	03h

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

その中 :

- ERRCNT** プレゼンテーションエラーカウンタを消去します。FFh は最後の検証が正しいことを示している。00H はパスワードがロックされていることを示している (最大再試行回数を超過した)。他の値は最後の認証が失敗したことを示している。
- DUMMY** カードから読み取った 2 バイトのダミーデータ
- SW1 SW2** = 90 00h (エラーなしの場合)

8.3.4. **READ_PROTECTION_BIT**

コマンドフォーマット (*PC_to_RDR_XfrBlock* 中の *abData* データフィールド)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B2h			

その中 :



MSB Byte Address = 0000 00A₉A₈b はメモリカードのメモリアドレス位置である

LSB Byte Address = A₇A₆A₅A₄ A₃A₂A₁A₀b はメモリカードのメモリアドレス位置である

MEM_L カードから読み出される保護ビットの長さ (8 ビットの倍数で、最大は 32 です)
。

$$\text{MEM_L} = 1 + \text{INT} [(\text{number of bits} - 1)/8]$$

例えば、メモリ 0010H から始まりの 8 保護ビットを読み取るために、下記の pseudo-APDU を発行する必要がある :

FF B1 00 10 01h

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

PROT 1	...	PROT L	SW1	SW2

その中 :

PROT y 保護ビットが含まれているバイト

SW1 SW2 = 90 00h (エラーなしの場合)

PROT バイト中で、保護ビットは以下のように並べている :

PROT 1								PROT 2								...									
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	P18	P17

その中 :

Px は応答データの BYTE x の保護ビットです。

‘0’バイトが書き込み保護されている

‘1’バイトは書き込むことができる

8.3.5. WRITE_MEMORY_CARD

コマンドフォーマット (*PC_to_RDR_XfrBlock*中の *abData* データフィールド)

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	Byte N
		MSB	LSB					
FFh	D0h							

その中 :

MSB Byte Address = 0000 00A₉A₈b はメモリカードのメモリアドレス位置である



LSB Byte Address = $A_7A_6A_5A_4 A_3A_2A_1A_0b$ はメモリカードのメモリアドレス位置である
MEM_L メモリカードに書き入れていないデータの長さ
Byte x メモリカードに書き入れていないデータ

応答データフォーマット (*RDR_to_PC_DataBlock*中の *abData*データフィールド)

SW1	SW2

その中 :

SW1 SW2 = 90 00h (エラーなしの場合)

8.3.6. WRITE_PROTECTION_MEMORY_CARD

コマンドで指定された各バイトは、内部でカードに指定されたアドレス中のデータと比べます。一致した場合、対応している保護ビットが不可逆的に“0”にプログラムされている。

コマンドフォーマット (*PC_to_RDR_XfrBlock* 中の *abData* データフィールド)

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	Byte N
		MSB	LSB					
FFh	D1h							

その中 :

MSB Byte Address = 0000 00A₉A₈b はメモリカードのメモリアドレス位置である
LSB Byte Address = $A_7A_6A_5A_4 A_3A_2A_1A_0b$ はメモリカードのメモリアドレス位置である
MEM_L メモリカードに書き入れていないデータの長さ
Byte x バイト値がバイトアドレスから始まるカード内のデータと比較される。BYTE 1 と Byte Address 中のデータを比べます ; BYTE N と (Byte Address + N - 1) 中のデータが比べます。

応答データフォーマット (*RDR_to_PC_DataBlock*中の *abData*データフィールド)

SW1	SW2

その中 :

SW1 SW2 = 90 00h (エラーなしの場合)

8.3.7. PRESENT_CODE_MEMORY_CARD (SLE4428 と SLE5528 のみ)

SLE4428 と SLE5528 に書き込む操作を有効にするために、メモリカードにシークレットコードを提出する時に、このコマンドを使用します。以下の操作を実行する：

1. プレゼンテーションエラーカウンタにビット‘1’を検索して、‘0’に変更する。
2. 指定されたシークレットコードをカードに提出する。
3. プレゼンテーションエラーカウンタを消去する。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU						
CLA	INS	P1	P2	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	00h	00h	02h		

その中：

CODE 2 バイトのシークレットコード (PIN)

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

SW1	SW2 ErrorCnt
90h	

その中：

SW1 = 90h

SW2 (ErrorCnt) = エラー カウンター。FFh は認証が成功したことを示している。00 h はパスワードがロックされていることを示している (最大再試行回数を超過した)。他の値は現在の認証が失敗したことを示している。

8.4. メモリカード – SLE4432/SLE4442/SLE5532/SLE5542

8.4.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択されたカードにパワーダウン/アップを実行し、同時にリセットを実行する時に使われます。

注釈： SCardConnect() API によって確立されたロジックなスマートカードリーダー通信後に使用しかできません。SCardConnect() API についての詳しい説明は PC/SC 基準を参照してください。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	06h

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

SW1	SW2

その中 :

SW1 SW2 = 90 00h (エラーなしの場合)

8.4.2. READ_MEMORY_CARD

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FFh	B0h	00h		

その中 :

Byte Address = A₇A₆A₅A₄ A₃A₂A₁A₀b はメモリカードのメモリアドレス位置である

MEM_L = メモリカードから読み出されていないデータの長さ

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

BYTE 1	...	BYTE N	PROT 1	PROT 2	PROT 3	PROT 4	SW1	SW2

その中 :

BYTE x = メモリカードから読み出されたデータ

PROT y = 保護ビットが含まれているバイト

SW1 SW2 = 90 00h (エラーなしの場合)

PROT バイト中で、保護ビットは以下のように並べている :

PROT 1								PROT 2								...									
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	P18	P17

その中：

Px は応答データの BYTE x の保護ビットです。

‘0’バイトが書き込み保護されている

‘1’バイトは書き込むことができる

8.4.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE4442 と SLE5542 のみ)

このコマンドがプレゼンテーションエラーカウンタを読み取る時に使われる。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	B1h	00h	00h	04h

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

ERRCNT	DUMMY 1	DUMMY 2	DUMMY 3	SW1	SW2

その中：

ERRCNT プレゼンテーションエラーカウンタを消去します。07h は最後の検証が正しいことを示している。00H はパスワードがロックされていることを示している（最大再試行回数を超過した）。他の値は最後の認証が失敗したことを示している。

DUMMY 从卡片口取的 3 个字口的虚口数据

SW1 SW2 = 90 00h (エラーなしの場合)

8.4.4. READ_PROTECTION_BITS

このコマンドは最初の 32 バイトの保護ビットを読み取る時に使われる。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	B2h	00h	00h	04h

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

PROT 1	PROT 2	PROT 3	PROT 4	SW1	SW2

その中 :

- PROT y** 保護ビットが含まれているバイト
- SW1 SW2** = 90 00h (エラーなしの場合)

PROT バイト中で、保護ビットは以下のように並べている :

PROT 1								PROT 2								...									
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	P18	P17

その中 :

- Px** は応答データの BYTE x の保護ビットです。
- ‘0’バイトが書き込み保護されている
- ‘1’バイトは書き込むことができる

8.4.5. WRITE_MEMORY_CARD

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	Byte N
FFh	D0h	00h						

その中 :

- Byte Address** = A7A6A5A4 A3A2A1A0b はメモ리카ードのメモリアドレス位置である
- MEM_L** メモ리카ードに書き入れているデータの長さ
- Byte x** メモ리카ードに書き入れているデータ

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

SW1	SW2

その中 :

- SW1 SW2** = 90 00h (エラーなしの場合)

8.4.6. WRITE_PROTECTION_MEMORY_CARD

コマンドで指定された各バイトは内部でカードに指定されたアドレス中のデータと比べる。一致した場合、対応している保護ビットが不可逆的に“0”にプログラムされている。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	Byte N
FFh	D1h	00h						

その中：

- Byte Address** = 000A₄ A₃A₂A₁A₀b (00h - 1Fh) はメモリカードの保護メモリアドレス位置である
- MEM_L** メモリカードに書き入れているデータの長さ
- Byte x** バイト値がバイトアドレスから始まるカード内のデータと比較される。BYTE 1 と Byte Address 中のデータを比べます；BYTE N と (Byte Address + N -1) 中のデータが比べます。

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

SW1	SW2

その中：

- SW1 SW2** = 90 00h (エラーなしの場合)

8.4.7. PRESENT_CODE_MEMORY_CARD (SLE4442 と SLE5542 のみ)

SLE4442 と SLE5542 に書き込む操作を有効にするために、メモリカードにシークレットコードを提出する時に、このコマンドを使用します。以下の操作を実行する：

1. プレゼンテーションエラーカウンタにビット‘1’を検索して、‘0’に変更する。
2. 指定されたシークレットコードをカードに提出する。
3. プレゼンテーションエラーカウンタを消去する。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	03h			



その中：

CODE 3バイトのシークレットコード (PIN)

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

SW1	SW2 ErrorCnt
90h	

その中：

SW1 = 90h

SW2 (ErrorCnt) = エラー カウンター。07h は検証が正しいことを示している。00H はパスワードがロックされていることを示している (最大再試行回数を超過した)。他の値は現在の認証が失敗したことを示している。

8.4.8. CHANGE_CODE_MEMORY_CARD (SLE4442 と SLE5542 のみ)

指定されたデータを新しいシークレットコードとして、カードに書き入れる時に、このコマンドを使用します。PRESENT_CODE コマンドでカードに現在のシークレットコードを提出してから、このコマンドを実行します。

コマンドフォーマット (*PC_to_RDR_XfrBlock* 中の *abData* データフィールド)

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FFh	D2h	00h	01h	03h			

応答データフォーマット (*RDR_to_PC_DataBlock* 中の *abData* データフィールド)

SW1	SW2

その中：

SW1 SW2 = 90 00h (エラーなしの場合)



9.0. PC_to_RDR_XfrBlock を介して、他のコマンドを実行する

9.1. GET_READER_INFORMATION

このコマンドはリーダーのファームウェアのバージョンを取得する時に使われます。

注釈： SCardConnect() API によって確立されたロジックなスマートカードリーダー通信後に使用しかできません。
SCardConnect() API についての詳しい説明は PC/SC 基準を参照してください。

コマンドフォーマット (PC_to_RDR_XfrBlock 中の abData データフィールド)

Pseudo-APDU				
CLA	INS	P1	P2	Le
FFh	09h	00h	00h	11h

応答データフォーマット (RDR_to_PC_DataBlock 中の abData データフィールド)

FIRMWARE										

その中：

FIRMWARE 11 バイトのファームウェアバージョン番号



付録 A. 応答エラーコード

下記のテーブルは ACR39 が返す可能なエラーコードをまとめています：

エラーコード	状態
FFh	SLOTERROR_CMD_ABORTED
FEh	SLOTERROR_ICC_MUTE
FDh	SLOTERROR_XFR_PARITY_ERROR
FCh	SLOTERROR_XFR_OVERRUN
FBh	SLOTERROR_HW_ERROR
F8h	SLOTERROR_BAD_ATR_TS
F7h	SLOTERROR_BAD_ATR_TCK
F6h	SLOTERROR_ICC_PROTOCOL_NOT_SUPPORTED
F5h	SLOTERROR_ICC_CLASS_NOT_SUPPORTED
F4h	SLOTERROR_PROCEDURE_BYTE_CONFLICE
F3h	SLOTERROR_DEACTIVATED_PROTOCOL
F2h	SLOTERROR_BUSY_WITH_AUTO_SEQUENCE
E0h	SLOTERROR_CMD_SLOT_BUSY

表3 : 応答エラーコード

Android は Google LLC. の商標です。

Atmel は Atmel また子会社がアメリカとほかの国の登録商標です。

EMV は EMVCo LLC がアメリカに登録商標または商標です。

Infineon はインフィニオン テクノロジー会社の登録商標です。

Microsoft は Microsoft Corporation がアメリカおよびまたはほかの国の登録商標もしくは商標です。