



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR40U

スマートカードリーダー(USB インターフェース)

リファレンスマニュアル V1.01



改定履歴

日付	改訂説明	バージョン
2023/09/21	<ul style="list-style-type: none">Initial version	1.00
2023/11/09	<ul style="list-style-type: none">検証テキスト付録 A の追加と 7.2 参照ミス の修正	1.01



目次

1.0. 紹介	4
1.1. 参照ファイル.....	5
1.2. 記号と略語.....	5
2.0. サポートしているスマートカード	7
2.1. MCU カード.....	7
2.2. メモリカード.....	7
3.0. システムブロック図	8
4.0. USB インターフェース	9
4.1. 通信パラメータ.....	9
4.2. エンドポイント.....	9
5.0. ユーザーインターフェース	10
5.1. LED ステータスインジケータ.....	10
6.0. スマートカードインターフェース	11
6.1. スマートカード電源 VCC (C1).....	11
6.2. プログラミング電圧 VPP (C6).....	11
6.3. カードタイプのセレクション.....	11
6.4. 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.....	11
7.0. USB 通信プロトコル	12
7.1. CCID Bulk-OUT メッセージ.....	13
7.1.1. PC_to_RDR_IccPowerOn.....	13
7.1.2. PC_to_RDR_IccPowerOff.....	15
7.1.3. PC_to_RDR_GetSlotStatus.....	15
7.1.4. PC_to_RDR_XfrBlock.....	15
7.1.5. PC_to_RDR_GetParameters.....	16
7.1.6. PC_to_RDR_ResetParameters.....	17
7.1.7. PC_to_RDR_SetParameters.....	17
7.2. CCID Bulk-IN メッセージ.....	20
7.2.1. RDR_to_PC_DataBlock.....	20
7.2.2. RDR_to_PC_SlotStatus.....	21
7.2.3. RDR_to_PC_Parameters.....	22
8.0. ホストプログラミング API	24
8.1. 周辺機器制御 (Peripherals Control).....	24
8.1.1. ファームウェアのバージョンを入手する (Get Firmware Version).....	24
8.1.2. カード電圧選択順序の取得 (Get Card Voltage Selection Sequence).....	24
8.1.3. カード電圧選択順序の設置 (Set Card Voltage Selection Sequence).....	25
8.1.4. 顧客データの書き込み (Write Customer Data).....	26



8.1.5. 顧客データの読み取り (Read Customer Data)	26
8.1.6. 顧客 PIN コード修正 (Change Customer PIN)	26
8.1.7. LED 状態読み取り (Read LED Status)	26
8.1.8. LED 状態書き込み (Write LED status)	27
8.2. メモリカードのコマンドセット.....	29
8.2.1. メモリカード – 1、2、4、8 および 16 Kb の I2C カード.....	29
8.2.2. メモリカード – 32、64、128、256、512 および 1024 Kb の I2C カード	34
8.2.3. メモリカード – ATMEL AT88SC153.....	37
8.2.4. メモリカード – ATMEL AT88C1608.....	42
8.2.5. メモリカード – SLE4418/SLE4428/SLE5518/SLE5528	46
8.2.6. メモリカード – SLE4432/SLE4442/SLE5532/SLE5542	52
附录 A. 応答エラーコード.....	58

図示一覧表

图 1 : ACR40U 構成	8
-----------------------	---

チャート一覧表

表 1 : 記号と略語.....	6
表 2 : USB インターフェース配線.....	9
表 3 : LED ステータスインジケータ	10
表 4 : 応答エラーコード	58

1.0. 紹介

ACR40U スマートカードリーダライタ（USB インタフェース）は、コンピュータとスマートカードとの間の通信媒体である。違う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/モバイルデバイス間に直接通信できません。ACR40U スマートカードリーダライタは、さまざまなスマートカードに標準化されたインタフェースを提供することで、ソフトウェア開発者が複雑なスマートカード操作から抜け出すことができるようになった。ACR40U でスマートカードの具体的な詳細を処理することにより、プログラマはスマートカードシステムの機能の実現に専念することができ、下位技術の詳細に注目する必要はありません。

1.1. 参照ファイル

下記のファイルは www.usb.org からダウンロードできる：

- 《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：スマートカード CCID 仕様 1.1（集積回路カード・インタフェース・デバイス）》、2005 年 4 月 22 日

下記のファイルは www.ansi.org から購入できる：

- 《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. 記号と略語

略語	説明
ATR	リセットへの応答 Answer To Reset
CCID	チップ/スマートカードインタフェースデバイス（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）
IFSD	プロトコル T=1 のチップ/スマートカードインタ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セレクション（Protocol and Parameters Selection）



略語	説明
RFU	将来の使用のために予約 ¹ (Reserved for Future Use)
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 : 記号と略語

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. サポートしているスマートカード

2.1. MCU カード

ACR40U は PC/SC 仕様に準拠しているスマートカードリーダーである。ISO 7816 クラス A、B、C の（5V、3V および 1.8V）カード、T = 0 または T = 1 プロトコル準拠のすべての MCU カードをサポートできる。

カードの ATR が専用の操作モードを指定した場合（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）、ただし ACR40U がこのモードをサポートできない場合、ACR40U はカードをリセットして、交渉モードに設置する。交渉モードに設置できないと、ACR40U がこのカードを拒否する。

カードの ATR が交渉のモード（TA2 が存在しない）および通信パラメータ（デフォルトパラメータではない）を指定した場合、ACR40U が PPS を実行して、その通信パラメータを使用してみる。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）。

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。

2.2. メモリカード

ACR40U が様々な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、一回で最大容量 128 バイトのメモリカード（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
- パスワードと認証によるセキュアなメモリ IC カード、以下を含めて：
 - Atmel®：AT88SC153 と AT88SC1608
- 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

3.0. システムブロック図

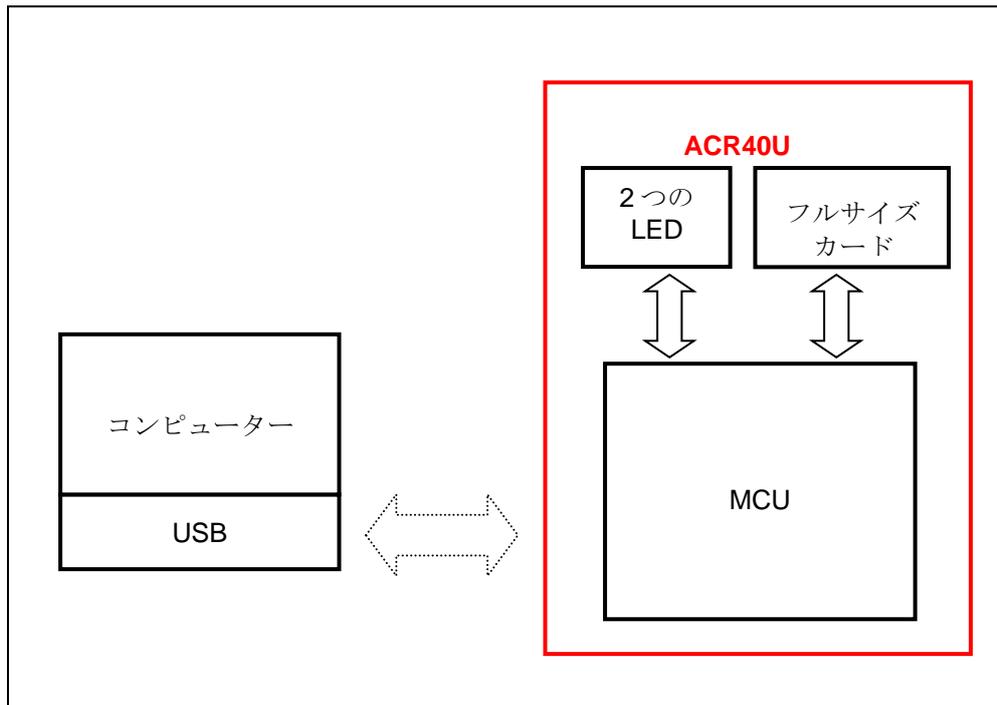


図 1 : ACR40U 構成

4.0. USB インターフェース

4.1. 通信パラメーター

ACR40U は USB 2.0 準拠の USB インターフェースを介して、PC と接続する。USB 全速モードをサポートして、速度は 12 Mbps である。

ピン	信号	機能
1	V _{BUS}	カードに+5 V の電源を供給
2	D-	ACR40U と PC は差動信号でデータを転送する。
3	D+	ACR40U と PC は差動信号でデータを転送する。
4	GND	参照用の電圧レベル

表 2： USB インターフェース配線

4.2. エンドポイント

ACR40U 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：

エンドポイント制御 (Control Endpo) 設定と制御用

バルクアウト (Bulk OUT)

ホストから ACR40U に送信するコマンドを対象する
(ペイロードの大きさは 64 バイトです)

バルクイン (Bulk IN)

ACR 4 0 U からホストに返す応答を対象する
(ペイロードの大きさは 64 バイトです)

割り込み入力 (Interrupt IN)

ACR40U からホストに送信する状態メッセージを対象する
(ペイロードの大きさは 8 バイトです)

5.0. ユーザーインターフェース

5.1. LED ステータスインジケータ

ACR40U がさまざまな動作状態を表示するための 2 つの LED インジケータを備えている。その中に：

- **ブルー LED** - USB モードでカードとリーダーの状態
- **グリーン LED** - スマートカードリーダー/ライタの電源状態

色	LED ステータス	状態
ブルー	OFF	カード操作なしで、ACR40U は PC 命令を待っています
	速く点滅する	ACR40U と PC 間のデータ転送
	ON	カードが接続して、電源いれました

色	LED ステータス	状態
緑	OFF	リーダー-OFF
	ON	リーダー-ON

表 3 : LED ステータスインジケータ

6.0. スマートカードインターフェース

ACR40U 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、ACR40U の実用的な機能性を向上するために、一定の制限や機能拡張をする。

6.1. 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 よりも高くはない。

6.2. 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ありません。ACR40U が実装されたコンタクト C6 (VPP) が通常の制御信号として使われていない。

6.3. 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する前に、ACR40U に正しい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。メモリカードと MCU ベースカードも含めている。

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ぶ。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、実行する。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て、リーダーは自動的にこのプロトコルタイプを選択する。

6.4. 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する。4.8 MHz の周波数が CLK 信号 (C3) に適用する。

7.0. USB 通信プロトコル

ACR40U は、USB を介してホスト (host) 側と相互作用を確立する。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た。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る。

ACR40U 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 に準拠する必要がある。

概要を以下に要約されている：

1. **制御コマンド**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。ACR40U 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。ACR40U 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：ACR40U にとって、*bMaxCCIDBusySlots* は 01 h に相当です)。

ACR40U 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	-	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	-	ACR40U が 1 つのカードスロット持っている
5	<i>bVoltageSupport</i>	1	-	ACR40U は 1.8V、3.0V と 5V のスロットサポート。
6	<i>dwProtocols</i>	4	-	ACR40U は T=0 および T=1 プロトコルサポート
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。
19	<i>dwDataRate</i>	4	-	デフォルトの ICC I/O ボーレートは 12903 bps です。

オフセット	データフィールド	大きさ	数値	説明
23	<i>dwMaxDataRate</i>	4	-	ICC I/O サポートできる最大のボーレートは 600 kbps です。
27	<i>bNumDataRatesSupported</i>	1	-	ボーレートの手動設定をサポートできません。
28	<i>dwMaxIFSD</i>	4	-	ACR40U は T=1 プロトコルでサポートできる最大の IFSD は 247 です。
32	<i>dwSynchProtocols</i>	4	-	ACR40U は同期カードをサポートできません。
36	<i>dwMechanical</i>	4	-	ACR40U は特別な機械的な特性をサポートできません。
40	<i>dwFeatures</i>	4	-	ACR40U は下記の特性をサポートできます： <ul style="list-style-type: none"> パラメータに基づいて、自動的に ICC のクロック周波数を変更します 周波数と FI、DI パラメータに基づいて、自動的にボーレートを変更します ACR40U との TPDU レベル変更
44	<i>dwMaxCCIDMessageLength</i>	4	-	ACR40U 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	-	同じ時間に一つだけのスロットが動作状態になります。

7.1. CCID Bulk-OUT メッセージ

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。



オフセット	データフィールド	大きさ	数値	説明
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>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 03h = 1.8V
8	<i>abRFU</i>	2	-	将来の使用のために保留する。

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	-	将来の使用のために保留する。

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

7.1.3. PC_to_RDR_GetSlotStatus

スロットのステータス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	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	-	将来の使用のために保留する。

このメッセージの応答は *RDR_to_PC_Parameters* 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。

オフセット	データフィールド	大きさ	数値	説明
7	<i>bBWI</i>	1	-	現在転送している CCIDs ブロックの待機タイムアウトを拡張するために使用します。「この数値にブロックの待機時間を掛け」の期限が切れた後、CCID はブロックをタイムアウトにします。
8	<i>wLevelParameter</i>	2	-	短 APDU レベル、RFU = 0000h 拡張 APDU レベル： 0000h - コマンド APDU がこのコマンドで開始および終了。 0001h - コマンド APDU がこのコマンドで開始して、次のコマンド PC_to_RDR_XfrBlock で継続。 0002h - abData フィールドは APDU コマンドの転送を続け、APDU コマンドを終了する。 0003h - abData フィールドは APDU コマンドの転送を続け、後には別のデータブロックも続く。 0010h - 空の abData フィールド、次の RDR_to_PC_DataBlock はレスポンス APDU を転送し続ける
10	<i>abData</i>	バイト配列	-	CCID に送信されるデータブロック。

このメッセージの応答は *RDR_to_PC_Parameters* 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	-
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

オフセット	データフィールド	大きさ	数値	説明
7	<i>AbRFU</i>	3	-	将来の使用のために保留する。

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ニコ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	-	将来の使用のために保留する。

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
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 プロトコルの構造 01h = T=1 プロトコルの構造</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 81h = 3線プロトコルの構造 82h = I2C プロトコル構造</p>



オフセット	データフィールド	大きさ	数値	説明
8	<i>abRFU</i>	2	-	将来の使用のために保留する。
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	-	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>bmTCCCKST0</i>	1	-	B0 – 0b, B7-2 – 000000b 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>bGuardTimeT0</i>	1	-	2 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
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。

オフセット	データフィールド	大きさ	数値	説明
11	<i>BmTCKST1</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 0-9 有効 B3-0 = CWI 値 0-Fh 有効
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

7.2.1. RDR_to_PC_DataBlock

このコマンドは ACR40U によって送信されて、*PC_to_RDR_lccPowerOn* と *PC_to_RDR_XfrBlock* メッセージに対しての応答です。

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 が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。

オフセット	データフィールド	大きさ	数値	説明
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 1.1) 6.2.1 節で定義されたスロットステータスレジスタ
8	<i>bError</i>	1	-	応答エラーコード CCID 規格 (Rev 1.1) 6.2.6 節で定義されたスロットステータスレジスタ
9	<i>bChainParameter</i>	1	-	短 APDU レベル、RFU = 00h 拡張 APDU レベル： 00h - 応答 APDU がこのコマンドで開始および終了します。 01h - 応答 APDU がこのコマンドで開始および続ける。 02h - abData データフィールドは引き続き応答 APDU を渡し、APDU 応答を終了します。 03h - abData データフィールドは応答 APDU の後に続けて別のデータブロックが続きます。 10h - 空の abData フィールド、次の PC_to_RDR_XfrBlock はレスポンス APDU を転送し続ける
10	<i>abData</i>	バイト配列	-	このデータフィールドは CCID から返され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 ACR40U 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	00000000h	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。

オフセット	データフィールド	大きさ	数値	説明
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 節で定義されたスロットステータスレジスタ
8	<i>bError</i>	1	-	応答エラーコード CCID 規格 (Rev 1.0) 4.2.1 節で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 = クロック動作中 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_to_PC_Parameters

このコマンドは ACR40U 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 節で定義されたスロットステータスレジスタ
8	<i>bError</i>	1	-	応答エラーコード CCID 規格 (Rev 1.0) 4.2.1 節で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 プロトコルの構造 01h = T=1 プロトコルの構造 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は CCID 仕様（1.0バージョン） 5.2.3 節の説明を参照してください。

8.0. ホストプログラミング API

8.1. 周辺機器制御 (Peripherals Control)

リーダーの周辺機器制御コマンドは USB スモードで Escape コマンド (0x6B) を介して、PC_to_RDR_Escape コマンドを実現されます。

8.1.1. ファームウェアのバージョンを入手する (Get Firmware Version)

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。

Get Firmware Version のコマンドフォーマット (5 バイト)

コマンド	CLA	INS	P1	P2	Lc
Get Firmware Version	E0h	00h	00h	19h	00h

Get Firmware Version の応答フォーマット (5 バイト + ファームウェアメッセージの長さ)

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	受信待ちのバイト数	ファームウェアのバージョン番号

例 : 応答 = E1h 00h 00h 00h 0Bh 41h 43h 52h 34h 30h 55h 2Dh 50h 32h 30h 30h

ファームウェア (HEX) = 41h 43h 52h 34h 30h 55h 2Dh 50h 32h 30h 30h

ファームウェアバージョン (ASCII) = "ACR40U-P200 "

8.1.2. カード電圧選択順序の取得 (Get Card Voltage Selection Sequence)

Get Card Voltage Selection Sequence コマンドは、カード電圧の電源オン順序を取得するために使用される。

Get Card Voltage Selection Sequence コマンドのフォーマット (5 バイト)

コマンド	CLA	INS	P1	P2	Lc
Get Card Voltage Selection Sequence	E0h	00h	00h	0Bh	00h

Get Card Voltage Selection Sequence 応答のフォーマット (5 バイト + カード電圧選択順序)

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	受信待ちのバイト数	カード電圧オン順序

その中：

カード電圧オン順序

カード電圧オン順序（1バイト）
 00h = Class C => Class B => Class A
 01h = Class A のみ
 02h = Class B のみ
 03h = Class C のみ
 04h = Class A => Class B => Class C

8.1.3. カード電圧選択順序の設置（Set Card Voltage Selection Sequence）

Set Card Voltage SelectionSequence コマンドは、カード電圧の電源オン順序を設置するために使用される。

Set Card Voltage Selection Sequence コマンドのフォーマット（5バイト）

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Set Card Voltage Selection Sequence	E0h	00h	00h	0Bh	01h	カード電圧オン順序

Set Card Voltage Selection Sequence 応答のフォーマット（5バイト+カード電圧選択順序）

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	受信待ちのバイト数	カード電圧オン順序

その中：

カード電圧オン順序

カード電圧オン順序（1バイト）
 00h = Class C => Class B => Class A
 01h = Class A のみ
 02h = Class B のみ
 03h = Class C のみ
 04h = Class A => Class B => Class C

8.1.4. 顧客データの書き込み (Write Customer Data)

このコマンドは、ユーザー定義データを書き込むために使用します (顧客データの最大長 : 64 バイト)

注 : このコマンドの詳細については、龍傑スマートカード株式会社の営業担当者に Eメールを送信するか info@acs.com.hk まで連絡してください。

8.1.5. 顧客データの読み取り (Read Customer Data)

顧客データを読み取る時にこのコマンドを使用します。

注 : このコマンドの詳細については、龍傑スマートカード株式会社の営業担当者に Eメールを送信するか info@acs.com.hk まで連絡してください。

8.1.6. 顧客 PIN コード修正 (Change Customer PIN)

このコマンドは、顧客 PIN コードの読み取りと PIN コード修正のみに使用します。

注 : このコマンドの詳細については、龍傑スマートカード株式会社の営業担当者に Eメールを送信するか info@acs.com.hk まで連絡してください。

8.1.7. LED 状態読み取り (Read LED Status)

このコマンドは LED の状態を読み取る際に使用されます。

Read the Power LED's status コマンドのフォーマット (5 バイト)

コマンド	CLA	INS	P1	P2	Lc
Read button status	E0h	00h	00h	29h	00h

Read power LED status 応答フォーマット (5 バイト + 電源 LED 状態 + 2 バイトリターンコード)

応答	CLA	INS	P1	P2	Le	データ出力	
結果	E1h	00h	00h	00h	受信待ちの バイト数	電源 LED 状態 (1 バイト)	リターンコード (2 バイト)

その中 :

電源 LED 状態 (1 バイト) :

LED 状態	説明	説明
Bit 0	現時点電源 LED 状態	1 = ON ; 0 = OFF

Bit 1	デフォルト電源 LED 設定（起動時にアクティブ）	1 = ON ; 0 = OFF
Bit 2 - 7	RFU	RFU

リターンコード

結果	SW1 SW2	意味
成功	90 00h	コマンドが成功に実行された
失敗	6D 00h	コマンド実行失敗

8.1.8. LED 状態書き込み（Write LED status）

このコマンドは電源 LED の状態を書き込む際に使用されます。

Write LED status コマンドのフォーマット（5 バイト）

コマンド	CLA	INS	P1	P2	Lc	データ	
Write LED status	E0h	00h	00h	29h	02h	ストレージ	LED 状態

Write LED status 応答フォーマット（5 バイト + 電源 LED 状態 + 2 バイトリターンコード）

応答	CLA	INS	P1	P2	Le	データ出力	
結果	E1h	00h	00h	00h	受信待ちの バイト数	電源 LED 状態	リターンコード

その中：

ストレージ（1 バイト）：

ストレージ	説明
1	設定を EEPROM に保存する
0	設定を EEPROM に保存しない

電源 LED 状態（1 バイト）：

LED 状態	説明	説明
--------	----	----



Bit 0	現時点電源 LED 状態	1 = ON ; 0 = OFF
Bit 1	デフォルト電源 LED 設定 (起動時にアクティブ)	1 = ON ; 0 = OFF
Bit 2 - 7	RFU	RFU

リターンコード

結果	SW1 SW2	意味
成功	90 00h	コマンドが成功に実行された
失敗	6D 00h	コマンド実行失敗

8.2. メモリカードのコマンドセット

8.2.1. メモリカード – 1、2、4、8 および 16 Kb の I2C カード

8.2.1.1. SELECT_CARD_TYPE

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

注： 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)

疑似 APDU					
CLA	INS	P1	P2	Lc	カードタイプ
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)

8.2.1.2. 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する。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)

疑似 APDU					
CLA	INS	P1	P2	Lc	ページサイズ
FFh	01h	00h	00h	01h	

その中：

ページサイズ = 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)

8.2.1.3. 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)

疑似 APDU				
CLA	INS	バイトアドレス		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)

バイト 1	バイト 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)

8.2.1.4. 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)

疑似 APDU								
CLA	INS	バイトアドレス		MEM_L	バイト 1	バイト n
		MSB	LSB					
FFh	D0h							

その中 :

Byte Address メモリカードのメモリアドレス位置



MEM_L	メモリに書き入れているデータの長さ
Byte x	メモリカ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)

8.2.2. メモリカード – 32、64、128、256、512 および 1024 Kb の I2C カード

8.2.2.1. SELECT_CARD_TYPE

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

注： 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)

疑似 APDU					
CLA	INS	P1	P2	Lc	カードタイプ
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)

8.2.2.2. 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する。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)

疑似 APDU					
CLA	INS	P1	P2	Lc	ページサイズ
FFh	01h	00h	00h	01h	

その中：

Data カードに送信されていない TPDU。

ページサイズ = 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)

8.2.2.3. 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)

疑似 APDU				
CLA	INS	バイトアドレス		MEM_L
		MSB	LSB	
FFh				

その中 :

INS = B0h : 32 Kb、64 Kb、128 Kb、256 Kb 及び 512 Kb の IIC カード

1011 000*b : 1024 Kb IIC カード、

その中 * はアドレッシング 17 ビットの MSB を示しています。

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)

バイト 1	バイト 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)

8.2.2.4. 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)

疑似 APDU								
CLA	INS	バイトアドレス		MEM_L	バイト1	バイトn
		MSB	LSB					
FFh								

その中：

INS=D0h : 32 Kb、64 Kb、128 Kb、256 Kb 及び 512 Kb の 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 を示しています。

Byte Address メモリカードのメモリアドレス位置

MEM_L メモリに書き入れているデータの長さ

Byte x メモリカ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)

8.2.3. メモリカード – ATMEL AT88SC153

8.2.3.1. SELECT_CARD_TYPE

このコマンドは選択されて、カードリーダーに挿入されたカードにパワーダウン/アップを実行する。同時にリセットを実行する時に使われる。それはまた、8 バイトのページの書き込みページサイズを選択する。

注： 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)

疑似 APDU					
CLA	INS	P1	P2	Lc	カードタイプ
FFh	A4h	00h	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)

8.2.3.2. 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)

疑似 APDU				
CLA	INS	P1	バイトアドレス	MEM_L
FFh		00h		

その中：

INS = B0h : 00b を読み取る

= B1h : 01b ゾーンを読み取る

= B2h : 10b ゾーンを読み取る

= B3h : 11b ゾーンを読み取る

= B4h : ヒューズを読み取る

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)

バイト 1	バイト 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)

8.2.3.3. 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)

疑似 APDU								
CLA	INS	P1	バイトアドレス	MEM_L	バイト 1	バイト n
FFh		00h						

その中 :

INS = D0h : 00b ゾーンを書く

 = D1h : 01b ゾーンを書く

 = D2h : 10b ゾーンを書く

 = D3h : 11b ゾーンを書く

 = D4h : ヒューズを書き入れる

Byte Address メモリカードのメモリアドレス位置

MEM_L メモリに書き入れているデータの長さ

MEM_D これからメモリカードに書き入るデータ

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)

8.2.3.4. VERIFY_PASSWORD

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)



疑似 APDU							
CLA	INS	P1	P2	Lc	Pw(0)	Pw(1)	Pw(2)
FFh	20h	00h		03h			

その中：

Pw(0),Pw(1),Pw(2) メモリカードに送信していないシークレットコード
P2 = 0000 00rp_b
 その中、2ビットの“rp”は比較されていないパスワードを示す
 r = 0 : パスワードを書く
 r = 1 : パスワードを読み取る
 p : パスワードセット番号、
 rp = 01 : セキュリティコード。

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。

8.2.3.5. INITIALIZE_AUTHENTICATION

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)

疑似 APDU								
CLA	INS	P1	P2	Lc	Q(0)	Q(1)	...	Q(7)
FFh	84h	00h	00h	08h				

その中：

Q(0),Q(1)...Q(7) ホスト挑戦、8バイト

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)

8.2.3.6. VERIFY_AUTHENTICATION



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)

疑似 APDU								
CLA	INS	P1	P2	Lc	Ch(0)	Ch(1)	...	Ch(7)
FFh	82h	00h	00h	08h				

その中 :

Ch(0),Ch(1)...Ch(7) ホスト挑戦、8 バイト

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)

8.2.4. メモリカード – ATMEL AT88C1608

8.2.4.1. SELECT_CARD_TYPE

このコマンドは選択されて、カードリーダーに挿入されたカードにパワーダウン/アップを実行する。同時にリセットを実行する時に使われる。それはまた、16 バイトのページの書き込みページサイズを選択する。

注： *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)

疑似 APDU					
CLA	INS	P1	P2	Lc	カードタイプ
FFh	A4h	00h	00h	01h	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)

SW1	SW2

その中：

SW1 SW2 = 90 00h (エラーなしの場合)

8.2.4.2. 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)

疑似 APDU				
CLA	INS	ゾーンアドレス	バイトアドレス	MEM_L
FFh				

その中：

INS = B0h : ユーザーゾーンを読み取る

= B1h : コンフィギュレーション・ゾーンまたはヒューズを読み取る

Zone Address = 0000 0A₁₀A₉A₈b、その中 A₁₀ はゾーンアドレスの MSB です

= ヒューズの書き込みと関係ない

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

= 1000 0000b : ヒューズを読み取る

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)

バイト 1	バイト 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)

8.2.4.3. 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)

疑似 APDU								
CLA	INS	ゾーンアドレス	バイトアドレス	MEM_L	バイト 1	バイト n
FFh								

その中 :

INS = D0h : ユーザーゾーンを書く

 = D1h : コンフィギュレーション・ゾーンまたはヒューズを書く

Zone Address = 0000 0A₁₀A₉A₈b, 其中 A₁₀ はゾーンアドレスの MSB です

 = ヒューズの書き込みと関係ない

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

 = 1000 0000b : ヒューズを書く

MEM_L メモリに書き入れているデータの長さ

Byte x メモリカ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)

8.2.4.4. VERIFY_PASSWORD

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)

疑似 APDU								
CLA	INS	P1	P2	Lc	データ			
FFh	20h	00h	00h	04h	RP	Pw(0)	Pw(1)	Pw(2)

その中：

Pw(0),Pw(1),Pw(2) メモリカードに送信していないシークレットコード
RP = 0000 rp₂p₁p₀b
 その中、4ビットの“rp₂p₁p₀”は比較されていないパスワードを示す
 r = 0 : パスワードを書く
 r = 1 : パスワードを読み取る
 p₂p₁p₀ : パスワードセット番号。
 (rp₂p₁p₀ = 0111 : セキュリティコード)

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。

8.2.4.5. INITIALIZE_AUTHENTICATION

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)

疑似 APDU								
CLA	INS	P1	P2	Lc	Q(0)	Q(1)	...	Q(7)
FFh	84h	00h	00h	08h				

その中：

Byte Address メモリカードのメモリアドレス位置
Q(0),Q(1)...Q(7) ホスト挑戦、8 バイト

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)

8.2.4.6. VERIFY_AUTHENTICATION

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)

疑似 APDU								
CLA	INS	P1	P2	Lc	Q1(0)	Q1(1)	...	Q1(7)
FFh	82h	00h	00h	08h				

その中：

Byte Address メモリカードのメモリアドレス位置

Q1(0),Q1(1)...Q1(7) ホスト挑戦、8 バイト

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)

8.2.5. メモリーカード – SLE4418/SLE4428/SLE5518/SLE5528

8.2.5.1. SELECT_CARD_TYPE

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

注： 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)

疑似 APDU					
CLA	INS	P1	P2	Lc	カードタイプ°
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)

8.2.5.2. 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)

疑似 APDU				
CLA	INS	バイトアドレス		MEM_L
		MSB	LSB	
FFh	B0h			

その中：

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

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

MEM_L メモリー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)

バイト 1	バイト 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)

8.2.5.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE4428 和 SLE5528)

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)

疑似 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** エラー カウンター。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意味 (最大再試行回数を超えた)。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)

8.2.5.4. 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)

疑似 APDU				
CLA	INS	バイトアドレス		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は 8 ビットの倍数です。最大数値は 32 です。

$$MEM_L = 1 + INT((\text{ビットのナンバー} - 1)/8)$$

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 B2 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								...									
P ₈	P ₇	P ₆	P ₅	P ₄	P ₃	P ₂	P ₁	P ₁₆	P ₁₅	P ₁₄	P ₁₃	P ₁₂	P ₁₁	P ₁₀	P ₉	P ₈	P ₇

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

8.2.5.5. 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)

疑似 APDU								
CLA	INS	バイトアドレス		MEM_L	バイト 1	バイト 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 = 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)

8.2.5.6. 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いちいち照合する。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)

疑似 APDU								
CLA	INS	バイトアドレス		MEM_L	バイト 1	バイト 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 = A₇A₆A₅A₄ A₃A₂A₁A₀b はメモリカードのメモリアドレス位置である

MEM_L メモリに書き入れていないデータの長さ

Byte x カード内の**バイトアドレス (Byte Address)** から始まるデータと比較するバイト値。BYTE 1 と Byte Address 中のデータを比べる ; BYTE N と (Byte Address + N -1) 中のデータが比べる。

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)

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

8.2.6.1. SELECT_CARD_TYPE

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

注： 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)

疑似 APDU					
CLA	INS	P1	P2	Lc	カードタイプ
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)

8.2.6.2. 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)

疑似 APDU				
CLA	INS	P1	バイトアドレス	MEM_L
FFh	B0h	00h		

その中：

Byte Address = A₇A₆A₅A₄ A₃A₂A₁A₀ 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)

バイト 1	バイト 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)

8.2.6.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE 4442 と SLE 5542)

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)

疑似 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 エラー カウンター。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 从卡片读取的 3 个字节的虚拟数据

SW1 SW2 = 90 00h (エラーなしの場合)

8.2.6.4. 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)

疑似 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								...									
P ₈	P ₇	P ₆	P ₅	P ₄	P ₃	P ₂	P ₁	P ₁₆	P ₁₅	P ₁₄	P ₁₃	P ₁₂	P ₁₁	P ₁₀	P ₉	P ₁₈	P ₁₇

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

8.2.6.5. 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)

疑似 APDU								
CLA	INS	P1	バイトアドレス	MEM_L	バイト 1	バイト N
FFh	D0h	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 メモリに書き入れているデータの長さ

Byte x メモリカ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)

8.2.6.6. 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いちいち照合する。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)

疑似 APDU								
CLA	INS	P1	バイトアドレス	MEM_L	バイト 1	バイト 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** メモリに書き入れていないデータの長さ
- Byte x** カード内の**バイトアドレス** (Byte Address) から始まるデータと比較するバイト値。BYTE 1 と Byte Address 中のデータを比べる ; BYTE N と (Byte Address + N - 1) 中のデータが比べる。

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)

8.2.6.7. PRESENT_CODE_MEMORY_CARD (SLE 4442 および SLE 5542)

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)

疑似 APDU							
CLA	INS	P1	P2	MEM_L	パスワード		
					バイト 1	バイト 2	バイト 3
FFh	20h	00h	00h	03h			



その中：

CODE 3バイトのパスワード (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。

8.2.6.8. CHANGE_CODE_MEMORY_CARD (SLE 4442 と SLE 5542)

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)

疑似 APDU							
CLA	INS	P1	P2	MEM_L	パスワード		
					バイト 1	バイト 2	バイト 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)





附录 A. 応答エラーコード

次の表は、ACR40U から返す可能性のある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 4 : 応答エラーコード

Android は Google LLC の商標です。

Atmel は Atmel Corporation または子会社がアメリカとまたはほかの国の登録商標です。

BluetoothTM ワードマークおよびロゴは登録された商標で、アドバンストカードシステム株式会社はそれぞれを使用する許可が持っています。その他の商標および商品名は、それぞれの所有者のもので。

Infineon は Infineon Technologies AG の登録商標です。

Microsoft は Microsoft Corporation がアメリカとまたはほかの国の登録商標です。